

HC32A472 Series

32-bit ARM® Cortex®-M4 Microcontroller

Reference Manual

Rev1.01 December 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 the terms 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 flaged "®" or "™" are tradeflags 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.....	34
Figure Index	39
Document Conventions	49
Register-Related Abbreviations	49
Glossary	49
About this Manual	49
Overview.....	50
1 Memory Mapping	51
1.1 Memory Mapping.....	51
1.2 External Space Mapping	57
1.3 Bit Field Space	57
1.4 Address Remapping.....	57
1.5 Remapping Register	59
1.5.1 Access Protection Register (MMF_REMPRT).....	59
1.5.2 Remap Control Register (MMF_REMC Rx) (x=0, 1).....	60
2 Bus Architecture (BUS).....	61
2.1 Overview	61
2.2 Bus Architecture	62
2.3 Bus Function	63
3 Reset Control (RMU)	64
3.1 Introduction	64
3.2 Reset Mode and Reset Flag Bit	65
3.3 Reset Timing.....	67
3.3.1 Power-on Reset	67
3.3.2 NRST Pin Reset	67
3.3.3 Brown-out Reset	68
3.3.4 Programmable Voltage Detection 1 Reset, programmable Voltage Detection 2 Reset	69
3.3.5 Watchdog Reset, Dedicated Watchdog Reset	70
3.3.6 Power-down Wake-up Reset	71
3.3.7 Software Reset.....	72
3.3.8 MPU Error Reset	72
3.3.9 RAM Parity Reset.....	73
3.3.10 RAMECC Reset	73

3.3.11	Clock Frequency Exception Reset	74
3.3.12	External High Speed Oscillator Abnormal Stop Reset	74
3.3.13	Cortex-M4 Lockup Reset	75
3.3.14	Judgment of Reset Mode	75
3.3.15	Reset Conditions for Each Module.....	75
3.4	Function Reset Control.....	76
3.5	Register Description	77
3.5.1	Function Reset Control 0 (RMU_FRST0).....	78
3.5.2	Function Reset Control 1 (RMU_FRST1).....	79
3.5.3	Function Reset Control 2 (RMU_FRST2).....	80
3.5.4	Function Reset Control 3 (RMU_FRST3).....	81
3.5.5	Reset Control Register (RMU_PRSTCR0)	83
3.5.6	Reset Flag Register0 (RMU _ RSTF0)	84
4	Clock Controller (CMU)	86
4.1	Introduction	86
4.2	System Block Diagram	87
4.2.1	System Block Diagram.....	87
4.2.2	Clock Frequency Measurement Block Diagram	88
4.3	Timer Specification	89
4.4	Working Clock Specification	90
4.5	Crystal Oscillator Circuit	92
4.5.1	External High-speed Oscillator.....	92
4.5.2	External High-Speed Oscillator Fault Detection	94
4.5.3	External High Speed Oscillator Fractional Frequency	96
4.5.4	External Low-speed Oscillator.....	97
4.6	Internal RC Clock	98
4.6.1	HRC Clock	98
4.6.2	MRC Clock.....	98
4.6.3	LRC Clock.....	99
4.6.4	SWDTLRC Clock	99
4.7	PLL Clock	100
4.8	Clock Switching Step	100
4.8.1	Clock Source Switching.....	101
4.8.2	Clock Division Switching	102
4.9	Clock Output Function	103
4.10	Clock Frequency Measurement.....	104
4.10.1	Clock Frequency Measurement.....	104
4.10.2	Digital Filtering Function	105

4.10.3	Interrupt/Reset Function	105
4.11	Register Description	106
4.11.1	CMU XTAL Configuration Register (CMU _ XTALCFGR).....	108
4.11.2	CMU XTAL Configuration Register (CMU _ XTALSTBCR).....	108
4.11.3	CMU XTAL Fractional Divider Configuration Register (CMU_XTALDIVR)	109
4.11.4	CMU XTAL Fractional Divider Control Register (CMU_XTALDIVCR).....	109
4.11.5	CMU XTAL Control Register (CMU _ XTALCR).....	110
4.11.6	CMU XTAL Oscillatory Fault Control Register (CMU _ XTALSTDCR)	111
4.11.7	CMU XTAL Oscillatory Fault State Register (CMU _ XTALSTDSR).....	111
4.11.8	CMU XTAL32 Configuration Register (CMU_XTAL32CFGR).....	112
4.11.9	CMU XTAL32 Filter Register (CMU_XTAL32NFR).....	112
4.11.10	CMU XTAL32 Control Register (CMU_XTAL32CR).....	113
4.11.11	CMU HRC Calibration Register (CMU_HRCTRM).....	113
4.11.12	CMU HRC Control Register (CMU _ HRCCR)	114
4.11.13	CMU MRC Calibration Register (CMU_MRCTRM)	114
4.11.14	CMU MRC Control Register (CMU_MRCCR)	115
4.11.15	CMU LRC Calibration Register (CMU_LRCTRM)	115
4.11.16	CMU LRC control register (CMU _ LRCCR).....	116
4.11.17	CMU PLLH Configuration Register (CMU_PLLHCFGR)	117
4.11.18	CMU MPLL Control Register (CMU_PLLHCR)	119
4.11.19	CMU Clock Source Stabilizer (CMU _ OSCSTBSR)	119
4.11.20	CMU System Clock Source Switching Register (CMU _ CKSWR).....	120
4.11.21	CMU Clock Divider Configuration Register (CMU_SCFGR)	121
4.11.22	CMU USB Clock Configuration Register (CMU_USBCKCFGR)	123
4.11.23	CMU CAN Clock Configuration Register (CMU_CANCKCFGR).....	124
4.11.24	CMU AD/DA Clock Configuration Register (CMU_PERICKSEL).....	126
4.11.25	CMU Debug Clock Configuration Register (CMU_TPIUCKCFGR).....	126
4.11.26	CMU MCO1 Configuration Register (CMU_MCO1CFGR)	127
4.11.27	CMU MCO2 Configuration Register (CMU_MCO2CFGR)	127
4.11.28	FCM Lower Limit Compare Value Register (FCM_LVR)	128
4.11.29	FCM Upper Limit Comparison Value Register (FCM_UVR)	128
4.11.30	FCM Counter Value Register (FCM_CNTR)	128
4.11.31	FCM Start Stop Register (FCM_STR).....	129
4.11.32	FCM Measurement Object Control Register (FCM_MCCR)	129
4.11.33	FCM Measurement Reference Control Register (FCM_RCCR).....	130
4.11.34	FCM Interrupt Reset Control Register (FCM_RIER).....	131
4.11.35	FCM Flag Register (FCM_SR)	131
4.11.36	FCM Flag Clear Register (FCM_CLR).....	132

5	Power control (PWC)	133
5.1	Introduction	133
5.2	Distribution of Power	134
5.3	Power Voltage Detection Unit (PVD) Description	135
5.3.1	Description of Action of Power-on/Power-off Reset	135
5.3.2	Brown-out Reset (BOR) Description	136
5.3.3	Programmable Voltage Detection 1 (PVD1), Programmable Voltage Detection 2 (PVD2)	137
5.3.4	PVD1, PVD2 Interrupt/ reset Block Diagram	138
5.3.5	Input/Output Pin	138
5.3.6	PVD1 Interrupt and Reset	139
5.3.7	PVD2 Interrupt and Reset	141
5.3.8	Internal Voltage Sampling and Detection Function	142
5.4	Wake Timer	143
5.5	Action Mode and Low Power Mode	143
5.5.1	Sleep Mode	147
5.5.2	Stop Mode	147
5.5.3	Power Down Mode	149
5.5.4	Reduce System Clock Speed	152
5.5.5	Disable Unused Clock Sources	152
5.5.6	Functional Clock Stop	153
5.5.7	Turn off Unused RAM	153
5.6	Register Protection Function	154
5.7	Register Description	155
5.7.1	Power Mode Control Register 0 (PWC_PWRC0)	157
5.7.2	Power Mode Control Register 1 (PWC_PWRC1)	157
5.7.3	Power Mode Control Register 4 (PWC_PWRC4)	158
5.7.4	Power-down Wake-up Enable Register 0 (PWC_PDWKE0)	158
5.7.5	Power-Down Wake-up Enable Register 1 (PWC_PDWKE1)	159
5.7.6	Power-down Wake-up Enable Register 2 (PWC_PDWKE2)	159
5.7.7	Power-down Wake-up Event Edge Selection Register (PWC_PDWKES)	160
5.7.8	Power-down Wake-up Flag Register 0 (PWC_PDWKF0)	160
5.7.9	Power-down Wake-up Flag Register 1 (PWC_PDWKF1)	161
5.7.10	Power Mode Control Register 5 (PWC_PWRC5)	161
5.7.11	Power Mode Control Register 6 (PWC_PWRC6)	162
5.7.12	Function Clock Control 0 (PWC_FCG0)	163
5.7.13	Function Clock Control 1 (PWC_FCG1)	164
5.7.14	Function Clock Control 2 (PWC_FCG2)	166

5.7.15	Function Clock Control 3 (PWC_FCG3)	167
5.7.16	PWC_FCG0 Protection Control (PWC_FCG0PC)	168
5.7.17	Function Protection Control Register (PWC_FPRC).....	168
5.7.18	STOP Mode Control Register (PWC_STPMCR)	169
5.7.19	RAM Power Control Register 0 (PWC_RAMPC0)	169
5.7.20	Peripheral RAM Low Power Control Register (PWC_PRAMLPC).....	170
5.7.21	PVD Control Register 0 (PWC_PVDCR0)	170
5.7.22	PVD Control Register 1 (PWC_PVDCR1)	171
5.7.23	PVD Filter Control Register (PWC_PVDFCR)	172
5.7.24	PVD Level Control Register (PWC_PVDLCR)	173
5.7.25	PVD Interrupt Control Register (PWC_PVDICR).....	174
5.7.26	PVD Detection Status Register (PWC_PVDDSR)	174
5.7.27	Wake-up Timer Control Register (PWC_WKTCR)	175
6	Initialization Configuration (ICG).....	176
6.1	Introduction	176
6.2	Register Description	177
6.2.1	Initialization Configuration Register 0 (ICG0)	177
6.2.2	Initialization Configuration register1 (ICG1).....	179
6.2.3	Initialize Configuration Register 3 (ICG3)	179
6.2.4	Initialize Configuration Register 4 (ICG4)	180
7	Embedded Flash (EFM)	181
7.1	Introduction	181
7.2	Main Features	181
7.3	Embedded Flash	182
7.4	Read Interface	184
7.4.1	Relationship between CPU Clock and Flash read time	184
7.5	Flash Read Cache	184
7.6	Flash Programming and Erasing Operations.....	185
7.6.1	Unlock the EFM_KEY1 Register.....	186
7.6.2	Write Protection Function.....	186
7.6.3	Single Programming Without Read BackFunction	186
7.6.4	Single Program Readback Function.....	187
7.6.5	Continuous Programming Function	187
7.6.6	Erase Function	188
7.6.7	Full Erase Function	188
7.6.8	Data Security Protection	189
7.6.9	D-BUS Read Protection Function	190
7.6.10	BGO Function	190

7.6.11	Interrupt.....	191
7.7	One-Time Programmable Byte (OTP)	192
7.8	Boot Swap	194
7.9	Register Description	196
7.9.1	Flash Access Write Protection Register (EFM_FAPRT)	197
7.9.2	Flash Key 1 Register (EFM_KEY1)	197
7.9.3	Flash Key 2 Register (EFM_KEY2)	198
7.9.4	Flash Stop Register(EFM_FSTP)	198
7.9.5	Flash Read Mode Register (EFM_FRMC)	199
7.9.6	Flash Erase Mode Register (EFM_FWMC)	200
7.9.7	Flash Status Register (EFM_FSR).....	201
7.9.8	Flash Status Clear Register (EFM_FSCLR)	203
7.9.9	Flash Interrupt Enable Register (EFM_FITE)	204
7.9.10	Flash Boot Swap Status Register (EFM_FSWP)	204
7.9.11	Chip Specific Flag Register (EFM_CHIPID)	205
7.9.12	UNIQUE ID Register (EFM_UQID 0)	205
7.9.13	UNIQUE ID Register (EFM_UQID 1)	205
7.9.14	UNIQUE ID Register (EFM_UQID2)	206
7.9.15	Flash Write Protection Lock Register (EFM_WLOCK)	206
7.9.16	Flash0 Write Protection Register (EFM_F0NWPRT)	207
7.9.17	Flash1 Write Protection Register (EFM_F1NWPRT)	208
7.10	Precautions.....	209
8	Built-in SRAM (SRAM).....	210
8.1	Introduction	210
8.2	Register Description	212
8.2.1	SRAM Check Control Register (SRAM_CKCR)	213
8.2.2	SRAM Check Control Protection Register (SRAMC_CKPR)	214
8.2.3	SRAM Check Status Register (SRAM_CKSR)	214
8.2.4	SRAM0 Error Injection Enable Register (SRAM0_EIEN)	215
8.2.5	SRAM0 Error Injection Bit Enable Register 0 (SRAM0_EIBIT0)	215
8.2.6	SRAM0 Error Injection Bit Enable Register 1 (SRAM0_EIBIT1)	215
8.2.7	SRAM0 Check Error Address Register (SRAM0_ECCERRADDR)	216
8.2.8	Ret_SRAM Error Injection Enable Register (SRAMB_EIEN).....	216
8.2.9	Ret_SRAM Error Injection Bit Enable Register 0 (SRAMB_EIBIT0).....	216
8.2.10	Ret_SRAM Error Injection Bit Enable Register 1 (SRAMB_EIBIT1).....	217
8.2.11	Ret_SRAM Check Error Address Register (SRAMB_ECCERRADDR)	217
9	General IO (GPIO)	218
9.1	Introduction	218

9.2	Port Function Summary	219
9.3	Action Description	220
9.3.1	General Purpose Input/Output GPIO Function	220
9.3.2	Peripheral Functions	221
9.3.3	Dual Peripheral Function	221
9.3.4	Event Port Input and Output Functions	221
9.3.5	External Interrupt EIRQ Input Function	222
9.3.6	Analog Function	222
9.3.7	General Control.....	222
9.4	Register Description	224
9.4.1	General Input Data Register (PIDRx).....	225
9.4.2	Universal Output Data Register (PODRx)	225
9.4.3	General Output License Register (POERx)	225
9.4.4	General Output Set Register (POSRx)	226
9.4.5	General Output Reset Register (PORRx)	226
9.4.6	Universal Output Flip Register (POTRx)	226
9.4.7	Special Control Register (PSPCR)	227
9.4.8	Public Control Register (PCCR).....	227
9.4.9	Write Protection Register (PWPR).....	228
9.4.10	General Control Register (PCRxy)	228
9.4.11	Function Select Register (PFSRxy)	230
9.4.12	Event Port Direction Selection Register (PEVNTDIRRm)	231
9.4.13	Event Port Input Data Register (PEVNTIDRm)	231
9.4.14	Event Port Output Data Register (PEVNTODRm).....	232
9.4.15	Event Port Output Data Reset Register (PEVNTORRm)	232
9.4.16	Event Port Output Data Set Register (PEVNTOSRm)	233
9.4.17	Event Port Rising Edge Input Permission Register (PEVNTRISRm)	233
9.4.18	Event Port Falling Edge Input Enable Register (PEVNTFALRm)	234
9.4.19	Event Port Input Filter Control Register (PEVNTNFCR).....	235
9.5	Precautions.....	236
10	Interrupt Controller (INTC)	237
10.1	Introduction	237
10.2	System Block Diagram	238
10.3	Directional Scale.....	239
10.3.1	Interrupt Digraph	239
10.3.2	Interrupt EventList	245
10.4	Functional Description	260
10.4.1	Nomaskable Interrupt	260

10.4.2	External Interrupt	260
10.4.3	Interrupt Event Multiplexer	261
10.4.4	Software Interrupt.....	261
10.4.5	Event Input	261
10.4.6	WFE Event Management.....	262
10.4.7	Noise Filter	262
10.4.8	Low Power Mode Return.....	263
10.5	Register Description	264
10.5.1	Nomaskable Interrupt Enable Register (INTC _ NMIER)	266
10.5.2	Nomaskable Interrupt Flag Register (INTC _ NMIFR)	267
10.5.3	Non-Maskable Interrupt Flag Clear Register (INTC_NMIFCR)	268
10.5.4	External interrupt control register (INTC_EIRQCRx) (x=0~15).....	269
10.5.5	External Interrupt Flag Register (INTC_EIFR).....	270
10.5.6	External Interrupt Flag Clear Register (INTC_EIFCR)	270
10.5.7	Interrupt Event Selection Register (INTC_INTSEL0~15)	271
10.5.8	Interrupt Event Mask Register (INTC_INTEN0~15).....	271
10.5.9	Stop Mode Wake-up Event Enable Register (INTC_WKEN).....	272
10.5.10	Software Interrupt Register (INTC_SWIER)	273
10.5.11	Event Enable Register (INTC_EVTER)	273
10.5.12	Interrupt Enable Register (INTC_IER).....	273
10.6	Precautions.....	274
11	Automatic Operation System (AOS).....	275
11.1	Introduction	275
11.1.1	Function Overview	275
11.1.2	Module Block Diagram	276
11.2	Functional Description	277
11.2.1	AOS Source Events List.....	277
11.2.2	AOS target list.....	277
11.2.3	Dedicated Trigger Source.....	279
11.2.4	Common Trigger Source.....	279
11.3	Register Description	280
11.3.1	Register List.....	280
11.3.2	Peripheral Trigger Event Register (INTSFTTRG).....	281
11.3.3	DCU Trigger Source Selection Register (DCU_TRGSELx) (x=1~3).....	281
11.3.4	DMA1 Transfer Start Trigger Source Selection Register (DMA1_TRGSELx) (x=0~7) 282	
11.3.5	DMA2 Transfer Start Trigger Source Selection Register (DMA2_TRGSELx) (x=0~7) 283	

11.3.6	DMA Channel Reset Trigger Source Selection Register (DMA_RC_TRGSEL)	284
11.3.7	Timer6 Hardware Trigger Event Selection Register (TMR6_TRGSELx) (x=0~3)	285
11.3.8	Timer4 Hardware Trigger Event Selection Register (TMR4_TRGSEL)	286
11.3.9	Event Port Trigger Source Select Register (PEVNT_TRGSEL12, PEVNT_TRGSEL34)	287
11.3.10	Timer0 Hardware Trigger Event Selection Register (TMR0_TRGSEL)	288
11.3.11	Timer2 Hardware Trigger Event Selection Register (TMR2_TRGSEL)	289
11.3.12	HASH Hardware Trigger Event Select Register A (HASH_TRGSELA)	290
11.3.13	HASH Hardware Trigger Event Select Register B (HASH_TRGSELB)	291
11.3.14	TimerA Hardware Trigger Event Selection Register (TMRA_TRGSELx, x = 0 ~ 3) ...	292
11.3.15	OTS trigger source select register (OTS_TRGSEL)	293
11.3.16	ADC1 Conversion On-chip Trigger Source Selection Register (ADC1_TRGSELx, x=0,1) 294	
11.3.17	ADC2 Conversion On-chip Trigger Source Selection Register (ADC2_TRGSELx, x=0,1) 295	
11.3.18	ADC3 Conversion On-chip Trigger Source Selection Register (ADC3_TRGSELx, x=0,1) 296	
11.3.19	Common trigger source select register (AOS_COMTRGSEL1, 2)	297
12	Memory Protection Unit (MPU)	298
12.1	Introduction	298
12.2	Functional Description	299
12.2.1	Area Range Settings	299
12.2.2	Permission Settings	299
12.2.3	MPU Action Selection	299
12.2.4	Start MPU	299
12.3	Application Instances	300
12.3.1	Only Enable Partial Space Access	300
12.3.2	Only some space access is prohibited	300
12.4	Register Description	301
12.4.1	Area scope description register MPU_RGDn (n=0~15)	302
12.4.2	Status flag register MPU_SR	302
12.4.3	Flag Clear Register MPU_ECLR	303
12.4.4	Write protection register MPU_WP	303
12.4.5	IP access protection register MPU_IPPR	304
12.4.6	MSP Protection Start Address Register (MPU_MSPPBA)	306
12.4.7	MSP Protection Control Register (MPU_MSPPCTL)	306
12.4.8	PSP Protection Start Address Register (MPU_PSPPBA)	307
12.4.9	PSP Protection Control Register (MPU_PSPPCTL)	307
12.4.10	SMPU1 Region Enable Register (MPU_S1RGE)	308

12.4.11	SMPU1 Region Write Permission Register (MPU_S1RGWP)	309
12.4.12	SMPU1 Region Read Permission Register MPU_S1RGRP	310
12.4.13	SMPU1 Control Register MPU_S1CR	311
12.4.14	SMPU2 Region Enable Register (MPU_S2RGE)	312
12.4.15	SMPU2 Region Write Permission Register (MPU_S2RGWP)	313
12.4.16	SMPU2 Region Read Permission Register MPU_S2RGRP	314
12.4.17	SMPU2 Control Register MPU_S2CR	315
12.4.18	FMPU zone enable register MPU_FRGE	315
12.4.19	FMPU Zone Write Permission Register MPU_FRGWP	316
12.4.20	FMPU Zone Write Permission Register MPU_FRGRP	316
12.4.21	FMPU Control Register MPU_FCR	317
13	Keyboard Scan Control Module (KEYSCAN)	318
13.1	Introduction	318
13.2	KEYSCAN System Block Diagram	319
13.3	Pin description	319
13.4	Functional Description	320
13.4.1	Key Recognition Function	320
13.4.2	Keyboard Scan Function	320
13.4.3	Precautions for use	321
13.5	Register Description	322
13.5.1	KEYSCAN Scan Control Register (KEYSCAN_SCR)	323
13.5.2	KEYSCAN Scan Enable Register (KEYCAN_SER)	324
13.5.3	KEYSCAN Scan Status Register (KEYSCAN_SSR)	324
14	Internal Clock Calibrator (CTC)	325
14.1	Introduction	325
14.2	Structure Diagram	325
14.3	Functional Description	326
14.3.1	Reference Clock	326
14.3.2	Frequency Calibration	333
14.3.3	Error Interrupt	334
14.3.4	Programming Guide	334
14.4	Register Description	336
14.4.1	Clock Calibration Control Register1 (CTC_CR1)	337
14.4.2	Clock Calibration Control Register2 (CTC_CR2)	339
14.4.3	Clock Calibration Status Register (CTC_STR)	340
14.4.4	Clock Calibration Counter (CTC_CNT)	340
15	DMA Controller (DMA)	341
15.1	Introduction	341

15.2	Module Diagram	342
15.3	Functional Description	343
15.3.1	Enable DMA Controller	343
15.3.2	Channel Selection and Channel Priority	343
15.3.3	Start DMA	343
15.3.4	Data block.....	343
15.3.5	Transmission Address Control	344
15.3.6	Number of Transmissions	344
15.3.7	Interrupt and Event Signal Output	344
15.3.8	Chain Transmission	346
15.3.9	Discontinuous Address Transmission	348
15.3.10	Channel Reset.....	349
15.3.11	Premature Interrupt of Transmission.....	350
15.4	Application Instances	351
15.4.1	Memory-to-Memory Transmission	351
15.4.2	Memory-to-Peripheral Transmission	353
15.4.3	Memory-to-Memory Linkage	354
15.5	Register Description	357
15.5.1	Register List.....	357
15.5.2	DMA Enable Register (DMA _ EN).....	359
15.5.3	Interrupt State Register0 (DMA _ INTSTAT0)	359
15.5.4	Interrupt State Register 1 (DMA_INTSTAT1)	360
15.5.5	Interrupt Mask Register 0 (DMA_INTMASK0)	360
15.5.6	Interrupt Mask Register 1 (DMA_INTMASK1)	361
15.5.7	Interrupt Reset Register 0 (DMA_INTCLR0)	361
15.5.8	Interrupt Reset Register 1 (DMA_INTCLR1)	362
15.5.9	Channel Enable Register (DMA _ CHEN).....	362
15.5.10	Channel Enable Reset Register (DMA _ CHENCLR).....	363
15.5.11	Software Start Register (DMA_SWREQ).....	363
15.5.12	Channel Reset Control Register (DMA_RCFGCTL).....	364
15.5.13	Transfer Request Status Register (DMA_REQSTAT).....	366
15.5.14	Channel State Observation Register (DMA _ CHSTAT).....	366
15.5.15	Transmission Source Address Register (DMA _ SARx) (x = 0 ~ 7).....	367
15.5.16	Transport Destination Address Register (DMA _ DARx) (x = 0 ~ 7).....	367
15.5.17	Data Control Register (DMA_DTCTLx) (x=0~7).....	368
15.5.18	Repeat Region Size Register (DMA_RPTx) (x=0~7)	369
15.5.19	Repeat area size register B (DMA_RPTBx) (x=0~7)	370

15.5.20	Source Device Discontinuous Address Transfer Control Register (DMA_SNSEQCTLx) (x=0~7)	371
15.5.21	Source Device Discontinuous Address Transfer Control Register B (DMA_SNSEQCTLBx) (x=0~7)	372
15.5.22	Target Device Discontinuous Address Transfer Control Register (DMA_DNSEQCTLx) (x=0~7)	373
15.5.23	Target Device Discontinuous Address Transfer Control Register B (DMA_DNSEQCTLBx) (x=0~7)	374
15.5.24	Chain Pointer Register (DMA_LL Px) (x=0~7)	375
15.5.25	Channel Control Register (DMA_CHxCTL) (x = 0 ~ 7)	376
15.5.26	Channel Monitor Registers (DMA_MONSARx, DMA_MONDARx, DMA_MONDTCTLx, DMA_MONRPTx, DMA_MONSNSEQCTLx, DMA_MONDSEQCTLx) (x=0~7)	377
15.6	Precautions	378
16	Voltage Comparator (CMP)	379
16.1	Introduction	379
16.2	Functional Block Diagram	380
16.3	Functional Description	382
16.3.1	Input Selection	382
16.3.2	Normal Comparison	382
16.3.3	Window Comparison	383
16.3.4	Blank Window	385
16.3.5	Hysteresis Function	389
16.3.6	Noise filtering	389
16.3.7	Interrupt Event	389
16.3.8	External Output	389
16.4	Precautions	390
16.4.1	Module Stop Function	390
16.4.2	Power Consumption when the Module is Stopped	390
16.4.3	Power Consumption When Stopping Low Power Mode	390
16.5	Register Description	391
16.5.1	Comparator Operating Mode Register (CMPx_MDR, x=1~2)	392
16.5.2	Comparator Filter Interrupt Register (CMPx_FIR, x=1~2)	393
16.5.3	Comparator Output Control Register (CMPx_OCR, x=1~2)	394
16.5.4	Comparator Result Monitoring Register (CMPx_OMR, x=1~2)	394
16.5.5	Comparator Input Selection Register (CMPx_PMSR, x=1~2)	395
16.5.6	Comparator Output Blank Window Setting Register 1 (CMPx_BWSR1, x=1~2)	396
16.5.7	Comparator Output Blank Window Setting Register 2 (CMPx_BWSR2, x=1~2)	398
17	Analog-to-Digital Conversion Module (ADC)	399

17.1	Introduction	399
17.2	ADC System Block Diagram	400
17.3	Functional Description	402
17.3.1	ADC Clock	402
17.3.2	Channel Selection	402
17.3.3	Trigger Source Selection	403
17.3.4	Sequence A Single Scan Mode	404
17.3.5	Sequence A Continuous Scanning Mode	405
17.3.6	Double Sequence Scanning Mode	406
17.3.7	Analog Watchdog Function	408
17.3.8	Sampling Time and Conversion Time of Analog Input	409
17.3.9	ADC Data Register Automatic Clear Function	410
17.3.10	Conversion Data Average Calculation Function	411
17.3.11	Programmable Gain Amplifier PGA	411
17.3.12	Multi-ADC cooperative working mode	412
17.3.13	Interrupt and Event Signal Output	418
17.4	Register Description	419
17.4.1	Register List	419
17.4.2	ADC Start Register (ADC_STR)	420
17.4.3	ADC Control Register 0 (ADC_CR0)	421
17.4.4	ADC Control Register 1 (ADC_CR1)	422
17.4.5	ADC Conversion Start Trigger Register (ADC_TRGSR)	423
17.4.6	ADC Channel Select Register A (ADC_CHSELRA)	424
17.4.7	ADC Channel Select Register B (ADC_CHSELRB)	424
17.4.8	ADC Average Channel Select Register (ADC_AVCHSELR)	425
17.4.9	ADC Sampling Period Register ADC_SSTRx, x=0~15/ADC_SSTRL	425
17.4.10	ADC Channel Mapping Control Register (ADC_CHMUXR)	426
17.4.11	ADC Interrupt Status Register (ADC_ISR)	427
17.4.12	ADC Interrupt Enable Register (ADC_ICR)	427
17.4.13	ADC Interrupt Status Reset Register (ADC_ISCLRR)	428
17.4.14	ADC Synergy Mode Control Register (ADC_SYNCCR)	429
17.4.15	ADC data register ADC_Dry, ADC1 y=0~21, ADC2 y=0~21, ADC3 y=0~22	430
17.4.16	Analog watchdog control register (ADC _AWDCR)	431
17.4.17	Analog Watchdog Status Register (AD _AWDSR)	432
17.4.18	Analog Watchdog Status Reset Register (ADC _AWDSCLRR)	432
17.4.19	Analogwatchdog Thresholds Register (ADC _AWD0DR0, ADC _AWD0DR1, ADC _AWD1DR0, ADC _AWD1DR1)	433

17.4.20	Analog watchdog comparison channel selection register (ADC _ AWD0 CHSR, ADC _ AWD1 CHSR)	433
17.4.21	Programmable Gain Amplifier Control Register ADC_PGACR	434
17.4.22	Programmable gain amplifier reference ground select register ADC_PGAVSSSEN.	435
17.5	Precautions.....	436
17.5.1	Considerations when Reading Data Register	436
17.5.2	Scan Finish Interrupt Handling Considerations	436
17.5.3	Precautions for Module Stop and Low Power Consumption Setting	436
17.5.4	ADC Conversion Analog Channel Input Pin Setting	436
17.5.5	Noise Control	436
18	Digital to Analog Converter (DAC).....	437
18.1	Introduction	437
18.2	Functional Block Diagram.....	437
18.3	Functional Description	438
18.3.1	DAC Conversion	438
18.3.2	Synchronization Conversion.....	439
18.3.3	Output Voltage.....	439
18.3.4	ADC Conversion Priority Mode	439
18.4	Precautions.....	441
18.4.1	Module Stop Function	441
18.4.2	Stop Low Power Mode	441
18.4.3	Power-down Low Power Mode	441
18.5	Register Description	442
18.5.1	DAC Data Register (DACx_DADRy , y=1~2).....	442
18.5.2	DAC Control Register (DACx_DACR)	443
18.5.3	DAC Analog Output Control Register (DACx_DAOCR).....	444
18.5.4	DAC ADC Conversion Priority Control Register (DACx_DAADPCR).....	445
19	Temperature Sensor (OTS).....	446
19.1	Introduction	446
19.2	Usage Notes	446
19.3	Register Description	449
19.3.1	OTS Control Register (OTS_CTL)	450
19.3.2	OTS Data Register 1 (OTS_DR1)	450
19.3.3	OTS Data Register 2 (OTS_DR2)	451
19.3.4	OTS Error Compensation Register (OTS_ECR).....	451
19.3.5	OTS Preset Temperature Data Register (OTS_PDR1,2,3)	451
20	Advanced Control Timer (Timer6)	452
20.1	Introduction	452

20.2	Basic Block Diagram.....	452
20.3	Functional Description.....	454
20.3.1	Waveform Mode.....	454
20.3.2	Timer Selection.....	454
20.3.3	Counting Direction.....	455
20.3.4	Comparison Output.....	455
20.3.5	Capture Input.....	456
20.3.6	Counter Refresh.....	457
20.3.7	Software Synchronization.....	458
20.3.8	Hardware Synchronization.....	459
20.3.9	Pulse Width Measurement.....	461
20.3.10	Periodic Measurement.....	462
20.3.11	Cache Function.....	463
20.3.12	Digital Filtering.....	468
20.3.13	General PWM Output.....	469
20.3.14	Periodic Interval Response.....	476
20.3.15	Orthogonal Coding Count.....	477
20.3.16	EMB Control.....	482
20.3.17	Typical Instances.....	483
20.3.18	Function Summary Table.....	489
20.4	Interrupt and Event Description.....	490
20.4.1	Interrupt Output.....	490
20.4.2	Event Output.....	491
20.5	Register Description.....	493
20.5.1	General Counter Value Register (TMR6_CNTER).....	495
20.5.2	General Refresh Value Register (TMR6_UPDAR).....	495
20.5.3	General Period Reference Value Register (TMR6_PERmR) (m=A~C).....	495
20.5.4	General Comparison Reference Value Register (TMR6_GCMR) (no = A~F).....	496
20.5.5	Dedicated Comparison Reference Value Register (TMR6_SCMR) (No = A~F).....	496
20.5.6	Dead zone Reference Value Register (TMR6_DTmR)(m = D, U & n = A, B).....	496
20.5.7	General Control Register (TMR6_GCONR).....	497
20.5.8	Interrupt Control Register (TMR6_ICONR).....	498
20.5.9	Cache Control Register (TMR6_BCONR).....	499
20.5.10	Dead Zone Control Register (TMR6_DCONR).....	501
20.5.11	Port Control Register (TMR6_PCNR).....	502
20.5.12	Port Control Register (TMR6_PCNR).....	504
20.5.13	Filter Control Register (TMR6_FCNR).....	506
20.5.14	Filter Control Register (TMR6_FCNR).....	507

20.5.15	Valid Period Register (TMR6_VPERR)	508
20.5.16	Status Flag Register (TMR6_STFLR)	509
20.5.17	Hardware Start Event Select Register (TMR6_HSTAR).....	511
20.5.18	Hardware Stop Event Select Register (TMR6_HSTPR)	512
20.5.19	Hardware Clear Event Selection Register (TMR6_HCLRR)	513
20.5.20	Hardware Refresh Event Selection Register (TMR6_HUPDR).....	514
20.5.21	Hardware Capture Event Select Register A (TMR6_HCPAR)	515
20.5.22	Hardware Capture Event Selection Register B (TMR6_HCPBR)	517
20.5.23	Hardware Increment Event Select Register (TMR6_HCUPR).....	519
20.5.24	Hardware Decrement Event Selection Register (TMR6_HCDOR).....	521
20.5.25	Software Synchronous Start Control Register (TMR6_SSTAR)	523
20.5.26	Software Synchronous Stop Control Register (TMR6_SSTPR).....	524
20.5.27	Software Synchronous Clear Control Register (TMR6_SCLRR)	525
20.5.28	Software Synchronous Refresh Control Register (TMR6_SUPDR)	526
20.6	Precautions.....	527
21	General Control Timer (Timer4)	529
21.1	Introduction.....	529
21.2	Basic Block Diagram.....	529
21.3	Functional Description.....	531
21.3.1	Basic Action	531
21.3.2	Cache Function	534
21.3.3	General PWM Output	540
21.3.4	Periodic Interval Response.....	546
21.3.5	EMB Control	548
21.3.6	Monitoring Output.....	548
21.3.7	Hardware Startup.....	550
21.4	Interrupt and Event Description	551
21.4.1	General Count Compare Match Interruptand Event.....	551
21.4.2	Counting Period Match Interruptand Events	551
21.4.3	Reload Count Match Interruptand Event.....	551
21.4.4	Dedicated Compare Match Event	551
21.5	Register Description	553
21.5.1	Count Value Register (TMR4 _ CNTR).....	555
21.5.2	Periodic Reference Register (TMR4 _ CPSR)	555
21.5.3	Control State Register (TMR4 _ CCSR)	556
21.5.4	Valid Period Register (TMR4 _ CVPR).....	557
21.5.5	Generic Baseline Register (TMR4 _ OCCRM).....	557
21.5.6	General Control State Register (TMR4 _ OCSRn).....	558

21.5.7	Generic Extended Control Register (TMR4 _ OCERn)	559
21.5.8	General Mode Control Register (TMR4 _ OCMRm).....	561
21.5.9	Special Reference Register (TMR4 _ SCCRm).....	566
21.5.10	Special Control State Register (TMR4 _ SCSRm)	567
21.5.11	Dedicated Extended Control Register (TMR4 _ SCER).....	568
21.5.12	Special Mode Control Register (TMR4 _ SCMRm)	568
21.5.13	PWM Basic Control Register (TMR4 _ POCRn)	569
21.5.14	PWM Status Control Register (TMR4 _ PSCR)	570
21.5.15	PWM Filtering Control Register (TMR4 _ PFSRn)	573
21.5.16	PWM Dead Zone Control Register (TMR4 _ PDA < B > Rn)	573
21.5.17	Overload Control Status Register (TMR4 _ RCSR)	574
21.5.18	Dedicated Interrupt Control Register (TMR4 _ SCIR)	575
21.5.19	Dedicated Flag Control Register (TMR4 _ SCFR)	575
22	Emergency Brake Module (EMB)	576
22.1	Introduction	576
22.2	Functional Description	578
22.2.1	Overview.....	578
22.2.2	Control PWM Signal Output when External Port Input Level Changes.....	580
22.2.3	Stop PWM Signal Output when the Level of PWM Output Port is in the Same Phase (Same High or Same Low).....	581
22.2.4	Stop PWM Signal Output According to Voltage Comparator Comparison Result	582
22.2.5	Stop PWM Signal Output when System Error Occurs	582
22.2.6	Write Register Software Control PWM Signal Output.....	583
22.3	Register Description	584
22.3.1	EMB Control Register 1_0~5 (EMB_CTL1_0~5)	585
22.3.2	EMB Control Register 1_6 (EMB_CTL1_6)	586
22.3.3	EMB Control Register 2_0~5 (EMB_CTL2_0~5)	587
22.3.4	EMB Control Register 2_6 (EMB_CTL2_6)	588
22.3.5	EMB Software Output Enable Control Register (EMB_SOEx) (x=0~6).....	589
22.3.6	EMB Status Register (EMB_STATx) (x=0~6).....	590
22.3.7	EMB Status Reset Register (EMB_STATCLR _x) (x=0~6)	592
22.3.8	EMB Interrupt Permission Register (EMB_INTEN _x) (x=0~6)	593
22.3.9	EMB Control PWM Output Release Mode Select Register (EMB_RLSSEL _x) (x=0~6) 594	
23	General Timer (TimerA)	595
23.1	Introduction	595
23.2	Basic Block Diagram.....	595
23.3	Functional Description	597

23.3.1	Waveform Mode	597
23.3.2	Timer Selection	597
23.3.3	Comparison Output.....	598
23.3.4	Capture Input.....	598
23.3.5	Synchronous Startup	599
23.3.6	Digital Filtering.....	600
23.3.7	Cache Function	601
23.3.8	Cascade Count.....	602
23.3.9	PWM Output.....	602
23.3.10	Orthogonal Coding Count	604
23.3.11	Infrared Function (IFRP)	608
23.4	Interrupt and Event Description	609
23.4.1	Compare Matched Interrupt and Events	609
23.4.2	Periodic Matching Interrupt and Events	610
23.5	Register Description	611
23.5.1	Generic Count Value Register (TMRA _ CNTER).....	613
23.5.2	Periodic Reference Value Register (TMRA _ PERAR)	613
23.5.3	Comparison Reference Value Register (TMRA_CMPARm, m=1~4).....	614
23.5.4	Control Status Register L (TMRA_BCSTRL)	615
23.5.5	Control Status Register H (TMRA_BCSTRH)	616
23.5.6	Interrupt Control Register (TMRA _ ICONR)	616
23.5.7	Event control register (TMRA _ ECONR)	617
23.5.8	Filter Control Register (TMRA _ FCONR)	618
23.5.9	Status Flag Register (TMRA _ STFLR)	619
23.5.10	Cache Control Register (TMRA_BCONRm, m=1~2).....	620
23.5.11	Capture Control Register (TMRA_CCONRm) (m=1~4)	621
23.5.12	Port Control Register (TMRA_PCONRm) (m=1~4)	622
23.5.13	Hardware Trigger Event Selection Register (TMRA _ HCONR)	624
23.5.14	Hardware increment Event Selection Register (TMRA _ HCUPR).....	626
23.5.15	Hardware Decrease Event Select Register (TMRA _ HCDOR)	627
24	General Timer (Timer2)	628
24.1	Introduction	628
24.2	Basic Block Diagram.....	628
24.3	Functional Description	630
24.3.1	Timer Selection	630
24.3.2	Comparison Output.....	631
24.3.3	Hardware Triggering.....	632
24.3.4	Digital Filtering.....	634

24.4	Interrupt and Event Description	635
24.4.1	Interrupt Output.....	635
24.4.2	Event Output	635
24.5	Register Description	637
24.5.1	Count Value Register (TMR2_CNTmR) (m=A , B)	638
24.5.2	Reference Value Register (TMR2_CMPmR) (m=A , B).....	638
24.5.3	Basic Control Register (TMR2 _ BCONR).....	639
24.5.4	Interrupt Control Register (TMR2_ICONR)	641
24.5.5	Port Control Register (TMR2_PCONR).....	642
24.5.6	Hardware Control Register (TMR 2 _ HCONR)	644
24.5.7	Status Flag Register (TMR2_STFLR)	646
24.6	Precautions.....	647
25	General Timer (Timer0)	648
25.1	Introduction.....	648
25.2	Basic Block Diagram.....	648
25.3	Functional Description	649
25.3.1	Timer Selection.....	649
25.3.2	Basic Count.....	650
25.3.3	Hardware Triggering.....	650
25.4	Interrupt and Event Description	651
25.4.1	Interrupt Output.....	651
25.4.2	Event Output	651
25.5	Register Description	652
25.5.1	Count Value Register (TMR0_CNTmR) (m=A~B).....	653
25.5.2	Reference Value Register (TMR0_CMPmR) (m=A~B)	653
25.5.3	Basic Control Register (TMR0 _ BCONR).....	654
25.5.4	Status Flag Register (TMR0_STFLR)	656
25.6	Precautions.....	657
26	Real Time Clock (RTC)	658
26.1	Introduction.....	658
26.2	Basic Block Diagram.....	659
26.3	Functional Description	660
26.3.1	Power-on Setting.....	660
26.3.2	RTC Count Start Setting	660
26.3.3	System Low Power Mode Switching	660
26.3.4	Read Count Register	660
26.3.5	Write Count Register.....	661
26.3.6	Alarm Clock Setting	661

26.3.7	Clock Error Compensation	661
26.3.8	1Hz Output	661
26.4	Interrupt Description	663
26.4.1	Alarm Interrupt	663
26.4.2	Periodic Interrupt	663
26.5	Register Description	664
26.5.1	Control Register 0 (RTC_CR0)	665
26.5.2	Control Register 1 (RTC_CR1)	665
26.5.3	Control Register 2 (RTC_CR2)	666
26.5.4	Control Register 3 (RTC_CR3)	667
26.5.5	Second Count Register (RTC_SEC)	667
26.5.6	Minute Count Register (RTC_MIN)	668
26.5.7	Hour Count Register (RTC_HOUR)	668
26.5.8	Day Count Register (RTC_DAY)	670
26.5.9	Week Count Register (RTC_WEEK)	671
26.5.10	Month Count Register (RTC_MON)	671
26.5.11	Year Count Register (RTC_YEAR)	672
26.5.12	Minute Alarm Register (RTC_ALMMIN)	672
26.5.13	Hour Alarm register (RTC_ALMHOUR)	672
26.5.14	Week Alarm Register (RTC_ALM WEEK)	673
26.5.15	Clock Error Compensation Registers (RTC_ERRCRH, RTC_ERRCRL)	674
27	Watchdog Counter (WDT/SWDT)	676
27.1	Introduction	676
27.2	Functional description	677
27.2.1	Start the Watchdog	677
27.2.2	Hardware Startup Mode	677
27.2.3	Software Startup Method	678
27.2.4	Refresh Action	679
27.2.5	Flag Bit	679
27.2.6	Interrupt Reduction	679
27.2.7	Count Underflow	680
27.2.8	Refresh Error	681
27.3	Register Description	682
27.3.1	Control Registers (SWDT_CR, WDT_CR)	683
27.3.2	Status Register (SWDT_SR, WDT_SR)	685
27.3.3	Refresh Register (SWDT_RR, WDT_RR)	685
27.4	Precautions	685
28	General Synchronous Asynchronous Transceiver (USART)	686

28.1	Introduction	686
28.2	USART System Block Diagram	688
28.3	Pin Description	688
28.4	Functional Description	689
28.4.1	UART	689
28.4.2	Multiprocessor Communication.....	699
28.4.3	UART-LIN	703
28.4.4	Smart Card.....	706
28.4.5	Clock Synchronization Mode	710
28.4.6	Digital Filtering Function	715
28.4.7	Interrupt.....	716
28.5	Register Description	717
28.5.1	Status Register (USART _ SR).....	718
28.5.2	Transmit Data Register (USART_TDR).....	721
28.5.3	Receive Data Register (USART_RDR)	721
28.5.4	Baud Rate Register (USART _ BRR)	722
28.5.5	Control Register 1 (USART_CR1)	723
28.5.6	Control Register 2 (USART_CR2)	726
28.5.7	Control register 3 (USART_CR3)	728
28.5.8	Pre-Divider Register (USART _ PR).....	729
28.5.9	LIN Baud Rate Measurement Count Register (USART_LBMC)	730
28.5.10	USART1 Filter Control Register (USART1_NFC)	731
28.6	Precautions.....	732
28.6.1	UART Considerations.....	732
28.6.2	Clock Synchronization Mode Considerations.....	732
28.6.3	Other Considerations	732
29	Integrated Circuit Bus (I2C)	733
29.1	Introduction	733
29.2	I2C System Block Diagram	734
29.2.1	System Block Diagram.....	734
29.2.2	Structural Diagram	735
29.3	Action Description	736
29.3.1	I2C protocol	736
29.3.2	Address Matching	745
29.3.3	SM Bus Action	750
29.3.4	Reset.....	751
29.3.5	Interrupt and Event Signal Output	752
29.3.6	Programmable Digital Filtering	754

29.4	Application Software Setting I2C Initialization Process	754
29.5	Register Description	755
29.5.1	I2C Control Register 1 (I2C_CR1)	756
29.5.2	I2C Control Register 2 (I2C_CR2)	757
29.5.3	I2C Control Register 3 (I2C_CR3)	758
29.5.4	I2C Control Register 4 (I2C_CR4)	759
29.5.5	I2C Slave Address Register 0 (I2C _ SLR0)	760
29.5.6	I2C Slave Address Register1 (I2C _ SLR 1)	761
29.5.7	I2C SCL Level When Timeout Control Register(I2C_SLTR)	762
29.5.8	I2C State Register (I2C _ SR).....	763
29.5.9	I2C State Reset Register (I2C _ CLR).....	767
29.5.10	I2C Data Sending Register (I2C _ DTR)	768
29.5.11	I2C Data Receiving Register (I2C _ DRR).....	769
29.5.12	I2C Data Shift Register (I2C _ DSR).....	770
29.5.13	I2C Clock Control Register (I2C _ CCR).....	771
29.5.14	I2C Filtering Control Register (I2C _ FLTR)	773
29.5.15	I2CFIFO Control Register (I2C_FSTR)	773
29.5.16	I2C Slave Address Receiver Register (I2C_SLVADRR)	774
30	Serial Peripheral Interface (SPI)	775
30.1	Introduction	775
30.2	SPI System Block Diagram	776
30.3	Pin Description	776
30.4	SPI Action System Description.....	777
30.4.1	Pin Status in Host Mode	777
30.4.2	Pin State in Slave Mode	777
30.4.3	SPI System Connection Instances	778
30.5	Data Communication Description.....	779
30.5.1	Baud Rate	779
30.5.2	Data Format.....	780
30.5.3	Transfer format	783
30.5.4	Communication Mode	785
30.5.5	Communication Interval Time.....	787
30.6	Operating Instructions.....	789
30.6.1	Operational Mode Summary	789
30.6.2	Host Action in SPI Operation Mode	790
30.6.3	The Slave Action in SPI Operation Mode	791
30.6.4	Host Action in Clock Synchronous Operation Mode	792
30.6.5	Slave Maneuver in Clock Synchronous Operation Mode	793

30.6.6	Processing FLOW OF SEVERAL SPI ACTIONS	794
30.7	Self-Diagnosis of Parity	795
30.8	Error Detection	796
30.8.1	Underload error.....	796
30.8.2	Pattern Error	797
30.8.3	Overload Error	797
30.8.4	Parity Check Error	799
30.9	SPI Initialization	800
30.9.1	Clear SPE Bits for Initialization.....	800
30.9.2	System Reset Initialization.....	800
30.10	Interrupt	801
30.11	Available Event Trigger Sources.....	801
30.12	Register Description	802
30.12.1	SPI Data Register (SPI _ DR).....	802
30.12.2	SPI Control Register (SPI_CR)	803
30.12.3	SPI Communication Configuration Register1 (SPI _ CFG1)	804
30.12.4	SPI Status Register (SPI _ SR).....	806
30.12.5	SPI Communication Configuration Register2 (SPI _ CFG2)	807
31	Quad-wire Serial Peripheral Interface (QSPI)	808
31.1	Introduction	808
31.2	Memory Map.....	810
31.2.1	Internal Bus Space	810
31.2.2	ROM Space and Bus Address Width	811
31.3	QSPI Bus	812
31.3.1	SPI Protocol	812
31.3.2	SPI Mode	814
31.4	Timing Adjustment of QSPI Bus	815
31.4.1	QSPI Bus Reference Clock.....	815
31.4.2	SPI Bus Reference Clock	816
31.4.3	QSSN Signal Minimum High Level Width.....	816
31.4.4	QSSN Establishment Time	817
31.4.5	QSSN Hold Time.....	817
31.4.6	Serial Data Receive Delay.....	818
31.5	Introduction to SPI Instructions for ROM Access	819
31.5.1	Existing QSPI-ROM Instruction Reference	819
31.5.2	Standard Read Command	819
31.5.3	Fast Read Command	820
31.5.4	Two-wire Output Fast Read Command	821

31.5.5	Two-Wire Input Output Fast Read Command	823
31.5.6	Four-wire Output Fast Read Command	824
31.5.7	Four-wire Input Output Fast Read Command	825
31.5.8	Enter 4-Byte Mode Command	826
31.5.9	Exit 4-Byte Mode Command	827
31.5.10	Write Permission Command	828
31.6	Scheduling of QSPI Bus Period	828
31.6.1	Single Flash Read with Independent Conversion	828
31.6.2	Flash Read Using Pre-read Function	829
31.6.3	Termination of Prefetching	829
31.6.4	Prefetch Status Monitoring	830
31.6.5	Flash Read Using QSPI Bus Period Stretching Feature	830
31.7	XIP control	831
31.7.1	Setting of XIP mode	831
31.7.2	Exit from XIP mode	832
31.8	Pin Status of QSIO2 and QSIO3	832
31.9	Direct Communication Mode	833
31.9.1	About Direct Communication Mode	833
31.9.2	Direct Communication Mode Setting of	833
31.9.3	QSPI Bus Period Generation in Direct Communication Mode	833
31.10	Interrupt	834
31.11	Precautions for Use	834
31.11.1	Setting Sequence of QSPI Registers	834
31.11.2	Module Stop Signal Setting	834
31.12	Register Description	835
31.12.1	QSPI Control Register (QSPI_CR)	836
31.12.2	QSPI Chip Select Control Register (QSPI_CSCR)	838
31.12.3	QSPI Format Control Register QSPI_FCR	839
31.12.4	QSPI Status Register (QSPI_SR)	840
31.12.5	QSPI Command Code register (QSPI_CCMD)	841
31.12.6	QSPI Direct Communication Command Register (QSPI_DCOM)	841
31.12.7	QSPI XIP Mode Code Register (QSPI_XCMD)	842
31.12.8	QSPI Flag Clear Register (QSPI_SR2)	842
31.12.9	QSPI External Extended Address Register (QSPI_EXAR)	843
32	Programmable Logic Array (PLA)	844
32.1	Introduction	844
32.2	PLA Module Block Diagram	844
32.2.1	Module Block Diagram	844

32.2.2	Action Description.....	845
32.2.3	PLA Input Multiplexer	846
32.2.4	Look-up Table (LUT) Control	848
32.2.5	PLA Output.....	849
32.3	Register Description	850
32.3.1	Global Control Register (PLA_GCTL).....	851
32.3.2	Status Register (PLA_STAT)	852
32.3.3	Multiplexer Selection Register (PLA to _MU XS) (small = 0...15).....	853
32.3.4	Lookup Table Control Register (PLA to UTC TL) (x= 0...15)	854
32.3.5	Control register (PLAx_CTL) (x = 0...15)	854
32.4	Precautions.....	855
33	USB2.0 Full Speed Module (USBFS)	856
33.1	Introduction to USBFS.....	856
33.2	Main Features of USBFS.....	856
33.2.1	General features	856
33.2.2	Host Mode Features	857
33.2.3	Device Mode Properties	857
33.3	USBFS System Block Diagram	858
33.4	USBFS Pin Description	858
33.5	USBFS Function Description.....	859
33.5.1	USBFS Clock and Working Mode	859
33.5.2	USBFS Mode Decision	859
33.5.3	USBFS Host Function.....	859
33.5.4	USBFS Device Capabilities	864
33.5.5	USBFS SOF Pulse Pin Output Function.....	868
33.5.6	USBFS Power Control	868
33.5.7	USBFS Dynamically Updates the USBFS_HFIR Register	869
33.5.8	USBFS Data FIFO.....	869
33.5.9	USBFS Host FIFO Architecture.....	870
33.5.10	USBFS Device FIFO Architecture	872
33.5.11	USBFS FIFO RAM Allocation.....	873
33.5.12	USBFS System Performance	874
33.5.13	USBFS Interrupt and Events.....	875
33.6	USBFS Programming Model	876
33.6.1	USBFS Module Initialization.....	876
33.6.2	USBFS Host Initialization.....	876
33.6.3	USBFS Device Initialization	877

33.6.4	USBFS DMA Mode	878
33.6.5	USBFS Host Programming Model	878
33.6.6	USBFS Device Programming Model.....	879
33.6.7	USBFS Operational Model	881
33.7	Register Description	898
33.7.1	USBFS System Control Register	902
33.7.2	USBFS global registers	903
33.7.3	USBFS Host Mode Register	925
33.7.4	USBFS Device Mode Register	937
33.7.5	USBFS Clock Gating Control Register	962
34	CAN FD Controller (CAN FD).....	963
34.1	Introduction	963
34.2	CAN FD System Block Diagram	964
34.3	Pin Description	964
34.4	Functional Description	965
34.4.1	Action Mode	965
34.4.2	Baud Rate Setting	965
34.4.3	Transmit Buffer	968
34.4.4	Receive Buffer	968
34.4.5	Receive Filter Register Set	969
34.4.6	Data transmission.....	970
34.4.7	Single data transmission.....	970
34.4.8	Cancel Data Sending	970
34.4.9	Data Reception	971
34.4.10	Error handling	971
34.4.11	Node Down	972
34.4.12	Lost Arbitration Position Capture	972
34.4.13	Loop Mode	972
34.4.14	Silent Mode	973
34.4.15	Software Reset Function	974
34.4.16	Upward Compatible with CAN-FD Function	975
34.4.17	Time-triggered TTCAN.....	975
34.4.18	TDC and RDC	978
34.4.19	Interrupt.....	979
34.5	Register Description	980
34.5.1	CAN Transmit Control Register (CAN_SYCTLREG)	981
34.5.2	CAN receive BUF register (CAN_RBUF).....	982
34.5.3	CAN Transmit BUF Register (CAN_TBUF)	985

34.5.4	CAN Configuration and Status Register (CAN_CFG_STAT)	987
34.5.5	CAN Command Register (CAN_TCMD).....	988
34.5.6	CAN Transmit Control Register (CAN_TCTRL)	990
34.5.7	CAN Receive Control Register (CAN_RCTRL)	991
34.5.8	CAN Receive and Transmit Interrupt Enable Register (CAN_RTIE)	992
34.5.9	CAN Receive and Transmit Interrupt Mark Register (CAN_RTIF)	993
34.5.10	CAN Error Interrupt Enable and Flags Register (CAN_ERRINT)	995
34.5.11	CAN Slow Bit Timing Register (CAN_BT).....	996
34.5.12	CAN Fast Timing Register (CAN_FBT)	997
34.5.13	CAN Send Delay Compensation Register (CAN_TDC).....	998
34.5.14	CAN Error and Arbitration Lost Capture Register (CAN_EALCAP)	998
34.5.15	CAN Warning Limit Register (CAN_LIMIT)	999
34.5.16	CAN Receive Error Counter Register (CAN_RECNT).....	999
34.5.17	CAN Transmit Error Counter Register (CAN_TECNT)	1000
34.5.18	CAN Filter Group Control Register (CAN_ACFCTRL)	1000
34.5.19	CAN Filter Group Enable Register (CAN_ACFEN)	1001
34.5.20	CAN filter group code and mask registers (CAN_ACF).....	1002
34.5.21	TTCAN TB slot pointer register (CAN_TBSLOT)	1003
34.5.22	TTCAN Time Triggered Configuration Register (CAN_TTCFG).....	1004
34.5.23	TTCAN Reference Message Register (CAN_REF_MSG)	1005
34.5.24	TTCAN Trigger Configuration Register (CAN_TRG_CFG).....	1006
34.5.25	TTCAN Trigger Time Register (CAN_TT_TRIG)	1007
34.5.26	TTCAN Trigger Watchdog Time Register (CAN_TT_WTRIG)	1007
34.6	Precautions.....	1008
34.6.1	CAN Bus Anti-jamming Measures.....	1008
34.6.2	CAN Controller Noise Constraints.....	1008
35	MDIO Serial Bus Interface (MDIO)	1009
35.1	Introduction	1009
35.2	Module Diagram	1009
35.3	Functional Description	1009
35.3.1	MDIO Frame Structure	1009
35.4	Application Instances	1012
35.4.1	Application Block Diagram.....	1012
35.5	Register Description	1015
35.5.1	MDIO Module Control Register(MDCON)	1016
35.5.2	MDIO Module Control Register 1 (MDCON1)	1017
35.5.3	MDIO Receive Frame Control Information Register (MDFRM)	1018
35.5.4	MDIO Receive Data Register (MDRXD).....	1019

35.5.5	MDIO Receiving Address Register(MDADR)	1019
35.5.6	MDIO Transmission Data Register(MDTXD)	1020
35.5.7	MDIO Physical Address Register(MDPHY)	1021
35.5.8	MDIO Process Signal Register(MDSTA)	1022
35.5.9	MDIO Interrupt Enable Register (MDIEN)	1024
35.5.10	MDIO Read Physical Address Pin Register(MDPIN)	1025
35.5.11	Time-out Register (MDTO)	1025
35.6	Precautions.....	1026
36	External Memory Controller EXMC.....	1027
36.1	Summary.....	1027
36.2	Basic Features	1027
36.2.1	Features List	1027
36.2.2	Controller Architecture.....	1028
36.2.3	Basic Access Specification	1029
36.2.4	Address Mapping	1030
36.2.5	Protocol Interface.....	1032
36.3	Functional Description.....	1034
36.3.1	SMC-SRAM/PSRAM/NOR Flash Controller.....	1034
36.4	Register Description	1053
36.4.1	SMC-SRAM/PSRAM/NOR Flash Register	1054
36.5	Precautions.....	1062
37	Cryptographic Coprocessing Module (CPM)	1063
37.1	Introduction.....	1063
37.2	Encryption and Decryption Algorithm Processor (AES).....	1063
37.2.1	Algorithm Introduction.....	1063
37.2.2	AES Module Function Description.....	1066
37.2.3	Encryption Operation Process.....	1066
37.2.4	Decryption Operation Process.....	1066
37.2.5	Data Instance	1067
37.2.6	Runtime Instructions.....	1070
37.2.7	Operation Precautions	1070
37.2.8	Register Description.....	1071
37.3	Secure Hash Algorithm (HASH).....	1075
37.3.1	Algorithm Introduction.....	1075
37.3.2	Operating Procedures	1076
37.3.3	Message Padding	1077
37.3.4	HMAC Operation	1078
37.3.5	Interrupt Description.....	1081

37.3.6	Hardware Trigger Event Selection	1081
37.3.7	Register Description.....	1082
37.4	True Random Number Generator (TRNG)	1086
37.4.1	Module Block Diagram	1086
37.4.2	Operating Procedures	1086
37.4.3	Interrupt and Event Output.....	1086
37.4.4	Register Description.....	1087
38	CRC Operation (CRC)	1089
38.1	Introduction	1089
38.2	Functional Block Diagram.....	1089
38.3	Functional Description	1089
38.3.1	CRC16 Encoding Mode	1089
38.3.2	CRC16 Check Mode.....	1090
38.3.3	CRC32 Encoding Mode	1090
38.3.4	CRC32 Check Mode.....	1090
38.4	Register Description	1091
38.4.1	Control Register (CRC _ CR)	1091
38.4.2	Results Register (CRC _ RESLT)	1091
38.4.3	Data Register (CRC _ DAT)	1092
39	Filter Math Accelerator (FMAC)	1093
39.1	Introduction	1093
39.2	Basic Block Diagram.....	1093
39.3	Operating procedures.....	1094
39.4	Module Enable	1094
39.5	Coefficient Normalization	1094
39.6	Interrupt and Event Description	1095
39.6.1	Interrupt Output.....	1095
39.6.2	Event Output	1095
39.7	Register Description	1096
39.7.1	Module Enable Register (FMAC_ENR)	1096
39.7.2	Basic Control Register (FMAC_CTR).....	1097
39.7.3	Interrupt Control Register (FMAC_IER)	1097
39.7.4	Data Input Register (FMAC_DTR)	1098
39.7.5	Data Input Register (FMAC_DTR)....	1098
39.7.6	Result Output Register 0 (FMAC_RTR0)	1098
39.7.7	Result Output Register 1 (FMAC_RTR1)	1099
39.7.8	Operation Status Register (FMAC_STR).....	1099
40	Data Computing Unit (CRC)	1100

40.1	Summary	1100
40.2	Functional Description	1101
40.2.1	Addition Mode	1101
40.2.2	Subtraction Mode	1101
40.2.3	Hardware Triggered Boot Mode	1101
40.2.4	Comparative Mode	1102
40.2.5	Interrupt and Event Signal Output	1103
40.3	Register Description	1104
40.3.1	DCU Control Register (DCUx_CTL)(x=1~3)	1105
40.3.2	DCU Flag Register (DCUx_FLAG) (x=1~3)	1106
40.3.3	DCU Data Register (DCUx_DATAy) (x=1~3,y=0,1,2)	1107
40.3.4	DCU Flag Reset Register (DCUx_FLAGCLR) (x=1~3)	1108
40.3.5	DCU Interrupt and Event Register (DCUx_INTEVTSEL) (x=1~3)	1109
41	Mathematical Arithmetic Unit (MAU)	1110
41.1	Introduction	1110
41.2	Functional Description	1110
41.2.1	Square Root Operation	1110
41.2.2	Sine Operation	1111
41.3	Interrupt and Event Description	1112
41.3.1	Interrupt Output	1112
41.3.2	Event Output	1112
41.4	Register Description	1113
41.4.1	Control Status Register (MAU_CSR)	1113
41.4.2	Data Input Register 0 (MAU_DTR0)	1114
41.4.3	Result Output Register 0 (MAU_RTR0)	1114
41.4.4	Data Input Register 1 (MAU_DTR1)	1114
41.4.5	Result Output Register 1 (MAU_RTR1)	1115
42	Debug Controller (DBGC)	1116
42.1	Introduction	1116
42.2	DBGC System Block Diagram	1116
42.3	SWJ-DP Debug Port (SWD and JTAG)	1117
42.3.1	Switching Mechanism of JTAG-DP or SW-DP	1118
42.4	Pinout and Debug Port Pinout	1118
42.4.1	SWJ Debug Port Pin	1118
42.4.2	Flexible SWJ-DP Pin Assignment	1118
42.4.3	Internal Pull-ups on JTAG Pins	1119
42.4.4	Use the Serial Interface and Release Unused Debug Pins for GPIO	1119
42.5	Register Description	1120

42.5.1	Chip Specific Identification Register (CHIPID)	1120
42.5.2	DBG State Register (MCUDBGSTAT)	1121
42.5.3	Peripheral Debugging Pause Register (MCUSTPCTL).....	1122
42.5.4	Debug Component Configuration Register (MCUTRACECTL).....	1124
42.5.5	Peripheral Debugging Pause Register 2 (MCUSTPCTL2).....	1125
42.6	SW Debug Port	1126
42.6.1	Introduction to SW Agreement.....	1126
42.7	TPIU (Trace Port Interface Unit)	1126
42.7.1	Introduction	1126
42.7.2	TRACE Pin Allocation	1127
42.7.3	MCU internal TRACECLKIN Connection.....	1128
42.7.4	TPIU Register	1128
42.7.5	TPIU Configuration Instance.....	1128
Version Revision History		1130

Table Index

Table 1-1	Memory-Map	51
Table 1-2	QSPI Address Space Allocation	57
Table 1-3	Target Address Configuration Instance	58
Table 1-4	Register List	59
Table 3-1	Reset Methods and Generation Conditions	65
Table 3-2	Reset Methods and Reset Flags.....	66
Table 3-3	Reset Conditions for Each Module	75
Table 3-4	List of RMU Registers	77
Table 4-1	Clock Source Main Features	89
Table 4-2	Specifications of Each Internal Clock.....	90
Table 4-3	Register List	106
Table 5-1	BOR Configuration.....	136
Table 5-2	PVD1/PVD2 Features	137
Table 5-3	Operation Mode.....	143
Table 5-4	Low Power Modes	144
Table 5-5	Operating conditions of low power consumption mode and status of each module in low power consumption mode.....	144
Table 5-6	Power-down mode submodes.....	150
Table 5-7	RAM Module and RAM Power-down Control Bits	153
Table 5-8	Register Protection List	154
Table 5-9	Register Summary.....	155
Table 6-1	Register Summary.....	176
Table 7-1	CPU Clock Frequency and Flash Read Wait Cycle Comparison Table	184
Table 7-2	Flash Actual Read Cycle Number.....	185
Table 7-3	OTP Address Distribution Table.....	192
Table 7-4	Register Summary.....	196
Table 8-1	SRAM Space Allocation.....	210
Table 8-2	List of SRAM Registers.....	212
Table 9-1	PORT Register Overview 1	224
Table 9-2	PORT Register Overview 2.....	224
Table 9-3	List of PORT Registers for 32bit Access	236
Table 10-1	External Interrupt Input Pin	237
Table 10-2	Interrupt Vector Table	239
Table 10-3	Interrupt Event Table	245
Table 10-4	INTC Register List	264
Table 11-1	AOS Target List	277

Table 11-2	Register List	280
Table 12-1	Register List	301
Table 13-1	KEYSCAN Pin Description.....	319
Table 13-2	KEYSCAN Register List.....	322
Table 14-1	Measurement Error when HRC Target Frequency of 20MHz.....	326
Table 14-2	Measurement Error when HRC Target Frequency of 16MHz.....	330
Table 14-3	List of CTC Registers.....	336
Table 15-1	Channel Reset Description	349
Table 15-2	Register Summary.....	357
Table 16-1	CMP Input and Output Pins.....	381
Table 16-2	CMP Analog Input Comparison Table	382
Table 16-3	CMP Register List.....	391
Table 16-4	Timer Window PWM List	397
Table 17-1	Specifications of Each ADC Unit	401
Table 17-2	Sequences A and B of Various Competitions	406
Table 17-3	AD Conversion Time	410
Table 17-4	ADC Register List.....	419
Table 18-1	DAC Input and Output Pins.....	438
Table 18-2	DACx Register List.....	442
Table 18-3	DAC Conversion and Analog Output Control	444
Table 19-1	OTS Preset Temperature Data	448
Table 19-2	Ehrc Usage and Setting Method	448
Table 19-3	List of OTS registers	449
Table 20-1	Basic Functions and Features of Timer6	452
Table 20-2	Timer6 Port List	453
Table 20-3	Comparison Table of Functions in Different Modes	489
Table 20-4	Timer6 Register List	493
Table 20-5	Counter (CNTER) Control Priority.....	527
Table 20-6	PWMA Port Output Control Priority	527
Table 20-7	PWMB Port Output Control Priority	528
Table 21-1	Basic Functions and Features of Timer4	529
Table 21-2	Timer4 Port List	530
Table 21-3	Register List	553
Table 21-4	PWM Port Output Status and Register Setting Value	572
Table 22-1	Port Allocation Table	580
Table 22-2	EMB Group Comparison Table	581
Table 22-3	Register List	584
Table 23-1	Basic Functions and Features of TimerA.....	595

Table 23-2	TimerA Port List	596
Table 23-3	TimerA Register List	611
Table 23-4	Internal Trigger Event TMRA_TRGSEL Selection Relationship Correspondence Table.....	612
Table 24-1	Basic functions and features of Timer2	628
Table 24-2	Timer2 Port List	630
Table 24-3	Timer2 Register List	637
Table 25-1	Timer0 Register List	652
Table 26-1	Basic Specifications of RTC.....	658
Table 26-2	Register List	664
Table 27-1	Basic Characteristics of Watchdog Counter.....	676
Table 27-2	Register List	682
Table 28-1	USART Pin Descriptions	688
Table 28-2	Tolerance of the UART Receiver when DIV_Fraction is 0	694
Table 28-3	Tolerance of UART receiver when DIV_Fraction is not 0	694
Table 28-4	UART interrupt/Event Table.....	699
Table 28-5	Multi-Processor Mode Interrupt/Event	703
Table 28-6	LIN interrupt/Event Table.....	706
Table 28-7	Smart Card Mode Interrupt/Event	710
Table 28-8	Clock Synchronization Mode Interrupt/Event Table	715
Table 28-9	USART Overall Interrupt List.....	716
Table 28-10	List of USART registers	717
Table 28-11	Baud Rate Calculation Formula (Invalid Fractional Baud Rate FBME=0)	722
Table 28-12	Baud rate Calculation Formula Valid Fractional Baud Rate FBME=1)	722
Table 29-1	Input/Output Pin	735
Table 29-2	Interruption List.....	752
Table 29-3	Event Signal Output List.....	753
Table 29-4	Register Summary.....	755
Table 30-1	SPI Features Key Points	775
Table 30-2	Pin Description	776
Table 30-3	SPI Pin Status Description in Host Mode.....	777
Table 30-4	SPI Pin Status Description in Slave Mode	777
Table 30-5	Some Setting Values Bit Speed	779
Table 30-6	SPI Pattern and Register Setting Relationship	789
Table 30-7	Correspondence Table for Error Detection.....	796
Table 30-8	SPI interrupt Source Description	801
Table 30-9	Register List	802
Table 31-1	QSPI main Specifications.....	808
Table 31-2	QSPI Pin.....	809

Table 31-3	QSPI Bus Reference Clock Selection List	815
Table 31-4	List of Reference Commands.....	819
Table 31-5	Pin Status of QIO2 and QIO3	832
Table 31-6	List of QSPI Registers	835
Table 32-1	PLAxMUX0 Input Selection	846
Table 32-2	PLAxMUX1 Input Selection	847
Table 32-3	LUT Control.....	848
Table 32-4	PLA Output Port Correspondence Table	849
Table 32-5	Register List	850
Table 33-1	USBFS Pin Description	858
Table 33-2	USBFS_GLBInterrupt Event Table.....	875
Table 33-3	USBFS System Control Register List	899
Table 33-4	USBFS Clock Gating Control Register List.....	900
Table 34-1	CAN Pin Description	964
Table 34-2	CAN Bit Time Setting Rules	966
Table 34-3	Baud Rate Setting Suggestions for 20MHz Communication Clock.....	967
Table 34-4	40MHzBaud Rate Setting Suggestions For 20mhz Communication Clock	967
Table 34-5	40MHzBaud Rate Setting Suggestions For 20mhz Communication Clock	967
Table 34-6	Software Reset Range Table	974
Table 34-7	CAN Interrupt Table	979
Table 34-8	CAN Register List.....	980
Table 34-9	Standard Format CAN Receiving Mailbox Format	982
Table 34-10	Extended Format CAN Receive Mailbox Format.....	983
Table 34-11	Control Bits.....	984
Table 34-12	Extended Format CAN Sending Email Format	985
Table 34-13	Extended Format CAN Send Mailbox Format	986
Table 35-1	MDIO Different Types of Frames	1010
Table 35-2	MDIO Operation Code.....	1011
Table 35-3	MDIO Register List.....	1015
Table 36-1	Basic Functions of EXMC	1027
Table 36-2	AHB Access Width And Memory Bit Width Corresponding Access Mode Table	1029
Table 36-3	SMC Protocol Interface	1032
Table 36-4	EXMC Port Function Assignment.....	1032
Table 36-5	Basic Setting Instance Of Single Read Operation	1037
Table 36-6	Instance Of Basic Setting For Single Read Operation Using Multiplexed Address and Data Lines	1042
Table 36-7	Basic Setting Instance Of Single Write Operation.....	1043

Table 36-8 Instance Of Basic Setting For Single Write Operation Using Multiplexed Address And Data Lines	1048
Table 36-9 Basic Setting Instance Of Single Write Operation In Synchronous Mode With Multiplexed Address And Data Lines	1050
Table 36-10 EXMC Register List	1053
Table 37-1 128 -Bit Operation Register Example	1067
Table 37-2 192 -Bit Operation Register Example	1068
Table 37-3 256 -Bit Operation Register Example	1069
Table 37-4 AES Encryption And Decryption Running Time.....	1070
Table 37-5 Register List	1071
Table 37-6 Register List	1082
Table 37-7 Register List	1087
Table 38-1 CRC Register List.....	1091
Table 39-1 FMAC Register List.....	1096
Table 40-1 Register List	1104
Table 41-1 MAU Register List	1113
Table 42-1 SWJ Debug Port Pins.....	1118
Table 42-2 Flexible SWJ-DP Pinout.....	1119
Table 42-3 Register List	1120
Table 42-4 Asynchronous Mode Pin	1127
Table 42-5 Sync Mode Pin	1127
Table 42-6 Pin Disable	1128

Figure Index

Figure 2-1	Bus Architecture Diagram.....	62
Figure 3-1	Power-on Reset	67
Figure 3-2	NRST Reset Timing	68
Figure 3-3	Brownout Reset.....	68
Figure 3-4	Programmable Voltage Detection 1 Reset	69
Figure 3-5	Programmable Voltage Detection 2 Reset	70
Figure 3-6	Watchdog and Dedicated Watchdog Reset	71
Figure 3-7	Power-down Wake-up Reset	72
Figure 3-8	Software Reset.....	72
Figure 3-9	MPU Error Reset.....	73
Figure 3-10	RAM Parity Reset.....	73
Figure 3-11	RAMECC Reset	73
Figure 3-12	Clock Frequency Abnormal Reset.....	74
Figure 3-13	External High-speed Oscillation Abnormal Reset	74
Figure 3-14	Cortex-M4 Lockup Reset	75
Figure 4-1	Clock System Block Diagram	87
Figure 4-2	Clock Frequency Measurement Block Diagram.....	88
Figure 4-3	External High-speed Oscillator Connection Instance	92
Figure 4-4	Instance connection diagram of external clock input	93
Figure 4-5	Instance of External High-Speed Oscillator Fault Detection	94
Figure 4-6	XTAL is Selected as the System Clock, and XTAL Oscillation Fault is Detected.....	95
Figure 4-7	External Low-speed Oscillator Connection Instance	97
Figure 4-8	Clock Source Switching.....	101
Figure 4-9	Clock Divider Switching	102
Figure 4-10	Clock Frequency Measurement Timing Diagram.....	104
Figure 5-1	Power Supply Diagram.....	135
Figure 5-2	Power-on Reset, Power-off Reset Waveforms	136
Figure 5-3	Brownout Reset Waveform.....	136
Figure 5-4	PVD1 Interrupt/Reset Block Diagram	138
Figure 5-5	PVD2 Interrupt/Reset Block Diagram	138
Figure 5-6	Power Monitor 1 Interrupt Timing Diagram.....	139
Figure 5-7	Power Monitor 1 Reset Timing Diagram	140
Figure 5-8	Power Supply Monitoring 2 Interrupt Operation Timing Diagram.....	141
Figure 5-9	Power supply monitoring 2 reset operation sequence diagram	142
Figure 5-10	Internal Voltage Sampling Diagram	142
Figure 5-11	PTWKn Structure Block Diagram.....	152

Figure 7-1	512kbytes Product Flash Address Distribution.....	182
Figure 7-2	256kbytes Product Single Flash Address Distribution	183
Figure 7-3	256kbytes Product Dual Flash Address Distribution.....	183
Figure 7-4	When The OTP Function Of 512kbytes Product Not Enabled, Flash Address Distribution	194
Figure 7-5	When The OTP Function Of 512kbytes Product Enabled, Flash Address Distribution.....	194
Figure 7-6	256kbytes Product OTP Function Is Not Enabled, Flash Address Allocation	195
Figure 7-7	Start Boot Swap 1	195
Figure 7-8	Start Boot Swap 2	195
Figure 8-1	Schematic Diagram of Error Injection Function.....	211
Figure 9-1	Port Basic Structure Diagram.....	219
Figure 10-1	Interrupt System Block Diagram.....	238
Figure 10-2	Interrupt Request/Event Input Selection	261
Figure 10-3	Schematic Diagram of Digital Filter Operation.....	262
Figure 11-1	AOS System Block Diagram	276
Figure 13-1	KEYSCAN System Block Diagram	319
Figure 13-2	Schematic Diagram of Keyboard Scanning Function.....	320
Figure 14-1	Basic Block Diagram of CTC.....	325
Figure 14-2	Schematic Diagram of CTC Calibration	333
Figure 15-1	DMA Structure Diagram.....	342
Figure 15-2	Chain transmission diagram	347
Figure 15-3	Schematic Diagram of Discontinuous Address Transmission (Taking Source Address as an Instance).....	348
Figure 15-4	Schematic Diagram of Discontinuous Reset	350
Figure 15-5	Application Instance 1: Memory-to-Memory Transfer.....	352
Figure 15-6	Application Instance 2: Transfer from Memory to Peripheral Circuits.....	354
Figure 16-1	CMP Functional Block Diagram	380
Figure 16-2	Voltage Comparison Diagram	383
Figure 16-3	Diagram of Window Comparison	384
Figure 16-4	Window Comparison Schematic.....	385
Figure 16-5	Window Output Schematic 1	386
Figure 16-6	Window Output Schematic Diagram II	387
Figure 16-7	Window Output Schematic Diagram 3.....	388
Figure 17-1	ADC Block Diagram.....	400
Figure 17-2	Channel Mapping Diagram	403
Figure 17-3	Sequence A Single Sweep Mode.....	404
Figure 17-4	Continuous Scanning	405
Figure 17-5	Double Sequence Scan Mode (Sequence A Restarts from Interrupted Channel)	407
Figure 17-6	Double Sequence Can Mode (Sequence A Restarts from First Channel)	407

Figure 17-7	Analog Watchdog Protection Area (Compare Mode)	408
Figure 17-8	ADC Conversion Time	409
Figure 17-9	Conversion Action When Averaging Function Valid	411
Figure 17-10	AD And PGA Channel Schematic	412
Figure 17-11	Single Parallel Trigger Mode (Three ADCs)	413
Figure 17-12	Single Delay Trigger Mode (Three ADCs)	415
Figure 17-13	Cyclic Parallel Trigger Mode (Three ADCs)	416
Figure 17-14	Cyclic Delay Trigger Mode (Two ADCs)	417
Figure 17-15	Cyclic Delay Trigger Mode (Three ADCs)	417
Figure 17-16	ADC interrupt and Event Output Timing	418
Figure 18-1	DAC Conversion Channel Block Diagram	437
Figure 18-2	DAC Conversion Diagram	438
Figure 18-3	ADC Conversion Priority Mode Operation Diagram	440
Figure 19-1	OTS functional block diagram	446
Figure 20-1	Timer6 Basic Block Diagram	453
Figure 20-2	Sawtooth Waveform (Counting Up)	454
Figure 20-3	Triangle Waveform	454
Figure 20-4	Compare Output Action	456
Figure 20-5	Capturing Input Actions	457
Figure 20-6	Hardware Refresh Action	457
Figure 20-7	Software Synchronization Action	459
Figure 20-8	Hardware Synchronous Action	461
Figure 20-9	Pulse Width Measurement	462
Figure 20-10	Cycle Measurement	462
Figure 20-11	Single-buffer Mode Comparison Output Timing	464
Figure 20-12	Double Buffer Capture Input Timing	465
Figure 20-13	Counting Buffer Action in Sawtooth Wave Mode	466
Figure 20-14	Count Buffer Action in Triangle Wave Mode	467
Figure 20-15	Count Buffer Action in Triangle Wave Mode	468
Figure 20-16	Filtering Function of Capture Input Port	469
Figure 20-17	Single-edge Aligned Independent PWM	469
Figure 20-18	Bilateral Symmetrical Independent PWM	470
Figure 20-19	Software Setting of GCMBR Complementary PWM Wave Output	471
Figure 20-20	Hardware Setting GCMBR Complementary PWM Wave Output	472
Figure 20-21	Double-sided Asymmetric PWM Output	473
Figure 20-22	Phase Single Edge Aligned Independent PWM	474
Figure 20-23	3-phase Bilateral Symmetrical Complementary PWM with Dead zone	475
Figure 20-24	Cycle Interval Valid Request Signal Action	476

Figure 20-25	Position Mode - Basic Counting	478
Figure 20-26	Position Counting Mode - Phase Difference Counting (1 Time Counting)	478
Figure 20-27	Position Counting Mode - Phase Difference Counting (2 Times Counting)	479
Figure 20-28	Position Counting Mode - Phase Difference Counting (4 times Counting)	479
Figure 20-29	Position Counting Mode - Direction Counting	479
Figure 20-30	Revolution Counting Mode - Z phase Counting	480
Figure 20-31	Revolution Counting Mode - Position Overflow Counting	481
Figure 20-32	Revolution Counting Mode - Mixed Counting	481
Figure 20-33	Revolution Counting Mode - Mixed Counting Z-phase Masking Action Instance 1	482
Figure 20-34	Revolution Counting Mode - Mixed Counting Z-phase Masking Action Instance 2	482
Figure 20-35	Instance of Interrupt & Event Output in Sawtooth Wave Mode	492
Figure 21-1	Timer4 Basic Block Diagram	530
Figure 21-2	Timer4 Sawtooth Waveform	531
Figure 21-3	Timer4 Triangle Wave Waveform	531
Figure 21-4	Timer4 Sawtooth Wave Mode Counting Action	531
Figure 21-5	Timer4 triangular wave mode counting action	532
Figure 21-6	Waveform Output Instance in Sawtooth Mode	533
Figure 21-7	Triangle Wave Mode Waveform Output Instance	533
Figure 21-8	Modify the Sawtooth Count period when the Cache is Invalid	534
Figure 21-9	Modify the Sawtooth Count Period when the Cache is Enabled	535
Figure 21-10	Modify the Triangular Wave Count Period when the Cache is Enabled	535
Figure 21-11	OCCR Buffer Data Transmission (When the Cycle Interval Response Link Disabled) ...	536
Figure 21-12	OCCR Buffer Data Transmission (Period Interval Response Link Enable)	537
Figure 21-13	Output Compare Buffer Data Transfer (OCMR Buffer Enabled)	538
Figure 21-14	SCCR Buffer Transmission Operation (Period Interval Response Link Transmission Disabled)	539
Figure 21-15	SCCR Buffer Transmission Operation (When Periodic Interval Response Link Transmission is Enabled)	540
Figure 21-16	Sawtooth Wave Independent PWM Output Instance	541
Figure 21-17	Triangle Wave Independent PWM Output Instance	541
Figure 21-18	Triangle Wave Extended PWM Output	542
Figure 21-19	Software Complementary PWM Output	543
Figure 21-20	Complementary PWM Output in Dead-zone Timer Mode	544
Figure 21-21	Waveform Output in Dead Zone Timer Mode when the Pulse Width is Abnormal	544
Figure 21-22	Complementary PWM Output in Dead-band Timer Filtering Mode	545
Figure 21-23	Cycle Interval Response Timing Diagram	546
Figure 21-24	Dedicated Event Output Signal Periodic Interval Response Output	547
Figure 21-25	PWM Period Output Monitoring Instance	549

Figure 21-26	Compare Dedicated Event Output TIM4_<t>_ADSM Timing in Startup Mode.....	550
Figure 21-27	Output Timing of Dedicated Event Output Signal in Delayed Start Mode.....	552
Figure 22-1	EMB Block Diagram	577
Figure 22-2	EMB Channel Functional Block Diagram	579
Figure 23-1	TimerA Basic Block Diagram.....	596
Figure 23-2	Sawtooth Waveform (Counting Up).....	597
Figure 23-3	Triangle Waveform	597
Figure 23-4	Compare Output Action	598
Figure 23-5	Capturing Input Actions	599
Figure 23-6	Software Synchronization Action	600
Figure 23-7	Filtering Function of Clock Input Port	600
Figure 23-8	Cache Action in Sawtooth Wave Mode.....	601
Figure 23-9	32-bit Cascade Counting Action.....	602
Figure 23-10	Single-edge Aligned PWM Output Instance.....	603
Figure 23-11	Instance of Bilateral Symmetrical PWM Output	604
Figure 23-12	Position Mode - Basic Counting.....	605
Figure 23-13	Position counting mode - phase difference counting (1 times counting)	605
Figure 23-14	Position Counting Mode - Phase Difference cCounting (2 times Counting)	606
Figure 23-15	Position Counting Mode - Phase Difference Counting (4 times Counting).....	606
Figure 23-16	Position Counting Mode - Direction Counting.....	606
Figure 23-17	Revolution Counting Mode - Z phase Counting.....	607
Figure 23-18	Revolution Counting Mode - Position Overflow Counting	608
Figure 23-19	Revolution Counting Mode - Mixed Counting	608
Figure 23-20	Schematic Diagram Of Infrared Output Signal Characteristics	609
Figure 24-1	Timer2 Basic Block Diagram.....	629
Figure 24-2	Compare output action	631
Figure 24-3	Hardware Startup And Clearing Action	632
Figure 24-4	Capturing Input Actions	633
Figure 24-5	Pulse Width Measurement	633
Figure 24-6	Cycle Measurement	634
Figure 24-7	Digital Filtering At The -TRIG Input Port	635
Figure 25-1	Timer0 Basic Block Diagram	648
Figure 25-2	Timer0 Counting Timing Diagram	650
Figure 26-1	Basic block diagram of RTC.....	659
Figure 27-1	Hardware Startup Instance	677
Figure 27-2	Software Startup Instance	678
Figure 27-3	Instances of Refresh Action Timing (Action Confirmation, Refresh Request Signal Falling Edge, etc.).....	679

Figure 27-4	Counter Underflow Action Instance	680
Figure 27-5	Counter Refresh Action Instance.....	681
Figure 28-1	USART System Block Diagram	688
Figure 28-2	UART Data Format	690
Figure 28-3	UART Send Data Legend 1	692
Figure 28-4	UART Send Data Legend 2	692
Figure 28-5	UART Internal Synchronization and Sampling Timing	693
Figure 28-6	UART Received Data Legend 1.....	695
Figure 28-7	UART Received Data Legend 2.....	696
Figure 28-8	Multiprocessor Communication Diagrams	700
Figure 28-9	Multiprocessor Mode Data Format	700
Figure 28-10	Diagram of Multi-Processor Mode Send Data.....	701
Figure 28-11	Diagram of Multiprocessor Mode Receive Data 1	702
Figure 28-12	Diagram of Multiprocessor Mode Receive Data 2	703
Figure 28-13	LIN Bus Data Behavior	703
Figure 28-14	Smart card Pattern connected legend	706
Figure 28-15	Smart Card Pattern Synchronization and Sampling Timing Figure	707
Figure 28-16	Smart Card ModeData Format	707
Figure 28-17	Smart Card Pattern Sending Data Legend	709
Figure 28-18	Smart Card Pattern Receive Data Legend.....	710
Figure 28-19	Clock Synchronization Mode Data Format	711
Figure 28-20	Clock Synchronization Mode Sending Data Legend 1	713
Figure 28-21	Clock Synchronization Mode Sending Data Legend 2	713
Figure 28-22	Diagram of Clock Synchronization Mode Receive Data 1.....	714
Figure 28-23	Diagram of Clock Synchronization Mode Receive Data 2.....	714
Figure 29-1	I2C System Block Diagram	734
Figure 29-2	Instance of I2C Bus Structure	735
Figure 29-3	The Following Figure is a Sequence Diagram of the I ² C bus.	736
Figure 29-4	I2C Data Format of I2C Bus.....	737
Figure 29-5	Sequence Diagram of host Sending Data in 7-bit Address Format (Instance).....	738
Figure 29-6	Sequence Diagram of Received Data for Host in 7-bit Address Format (Instance)	739
Figure 29-7	Sequence Diagram of Slave Send Mode in 7-bit Address Format (Instance).....	740
Figure 29-8	Sequence Diagram of 7-bit Address Format Slave Receive Mode (Instance)	741
Figure 29-9	SSCL Synchronization Time Series.....	742
Figure 29-10	Sending Sequence Diagram of Slave (1)	743
Figure 29-11	Sending Sequence Diagram of Slave (2)	744
Figure 29-12	Slave Receiving Timing Diagram	744
Figure 29-13	Fast ACK/NACK Timing Diagram.....	745

Figure 29-14	Timing when selecting 7-bit address format.....	746
Figure 29-15	Timing when selecting 10-bit address format.....	747
Figure 29-16	Broadcast Address Matching Timing Diagram	748
Figure 29-17	SMBus Host Address Matching Timing Diagram	748
Figure 29-18	SMBus Alarm Response Address Matching Timing Diagram	749
Figure 29-19	SMBus Default Address Timing Diagram.....	750
Figure 29-20	Digital Filtering Circuit Block Diagram	754
Figure 30-1	System Block Diagram.....	776
Figure 30-2	Host Mode Structure	778
Figure 30-3	Three-wire Clock Synchronization Operation	779
Figure 30-4	Data Format.....	780
Figure 30-5	MSB sent First, Parity Check Invalid	781
Figure 30-6	LSB sent First, Parity Check Invalid	781
Figure 30-7	MSB sent First, when parity check valid.....	782
Figure 30-8	LSB sent First, when parity check valid.....	782
Figure 30-9	Data Transfer Format Diagram (CPHA = 0)	783
Figure 30-10	Data Transfer Format (CPHA = 1)	784
Figure 30-11	Full Duplex Synchronous Serial Communication.....	785
Figure 30-12	Sending communication only.....	786
Figure 30-13	General Communication	787
Figure 30-14	Continuous Communication.....	788
Figure 30-15	Parity Checking Process	795
Figure 30-16	Overload Error Handling	797
Figure 30-17	Schematic diagram of the Action when the Clock Automatic Stop Function is Enabled (CPHA=1).....	798
Figure 30-18	Schematic diagram of the Action when the Clock Automatic Stop Function is Enabled (CPHA=0).....	798
Figure 30-19	Parity Error.....	799
Figure 31-1	Module Composition Diagram of QSPI	809
Figure 31-2	Default Area Setting and AHB Bus Space Memory Mapping Relationship Diagram	810
Figure 31-3	QSPI-ROM Space Memory Map	811
Figure 31-4	Schematic Diagram of Extended SPI Protocol Action 1 (Fast Read Mode).....	812
Figure 31-5	Schematic Diagram of Extended SPI Protocol Action 2 (Four-Wire Input and Output Fast Read Mode).....	812
Figure 31-6	Schematic Diagram of Two-wire SPI Protocol Action (Fast Read Mode)	813
Figure 31-7	Schematic diagram wire SPI protocol action (fast read mode)	813
Figure 31-8	Basic Timing Diagram of Serial Interface.....	814

Figure 31-9 Schematic Diagram of Output Clock Duty Ratio Correction when HCLK is Divided by Three as the Reference Clock	816
Figure 31-10 Schematic Diagram of QSSL Establishment Time Configuration.....	817
Figure 31-11 QSSN Hold Time Configuration Diagram.....	817
Figure 31-12 Schematic Diagram of Data Receiving Delay	818
Figure 31-13 Schematic Diagram of a Standard Read Bus Period.....	820
Figure 31-14 Schematic Diagram of Fast Read Bus Period	820
Figure 31-15 Schematic Diagram of Selecting XIP Mode Four Fast Read Bus Period	821
Figure 31-16 Schematic Diagram of Fast Read Bus Period	822
Figure 31-17 Schematic Diagram of Two-wire Output Fast Read Bus Period with XIP Mode Selected	822
Figure 31-18 Schematic Diagram of Two-Wire Input and Output Fast Read Bus Period.....	823
Figure 31-19 Schematic Diagram of Two-wire Input/Output Fast Read Bus Period with XIP Mode Selected	824
Figure 31-20 Schematic Diagram of Four-wire Output Fast Read Bus Period	825
Figure 31-21 Schematic Diagram of Four-wire Output Fast Read Bus Period with XIP Mode Selected	825
Figure 31-22 Schematic Diagram of Four-wire Input and Output Fast Read Bus Period	826
Figure 31-23 Schematic Diagram of Four-wire Input and Output Fast Read Bus Period with XIP Mode Selected	826
Figure 31-24 Enter 4-Byte Mode Command Bus Period Schematic Diagram	827
Figure 31-25 Exit 4-Byte Mode Command Bus Period Schematic Diagram.....	827
Figure 31-26 Schematic Diagram of Write Permission Command Bus Period	828
Figure 31-27 Schematic Diagram of Single Flash Memory Data Read Operation with Independent Conversion	828
Figure 31-28 Schematic Diagram of Data Reading Operation when Pre-reading Function Valid.....	829
Figure 31-29 Schematic Diagram of Data Read Operation Using QSPI Bus Period Extension Function	831
Figure 31-30 Schematic Diagram of XIP Mode Control	831
Figure 32-1 PLA Module Block Diagram	844
Figure 32-2 PLA Unit Block Diagram	845
Figure 33-1 USBFS System Block Diagram	858
Figure 33-2 USBFS Host Mode System Construction Diagram	860
Figure 33-3 USBFS Device Mode System Construction Diagram	864
Figure 33-4 Schematic Diagram Of USBFS Dynamically Updating USBFS_HFIR Register.....	869
Figure 33-5 Schematic Diagram of FIFO Architecture in USBFS Host Mode	870
Figure 33-6 Schematic Diagram of FIFO Architecture in USBFS Device Mode.....	872
Figure 33-7 USBFS Control Status Register Memory Map	899

Figure 34-1	CANFD System Block Diagram.....	964
Figure 34-2	CAN Bit Time Definition Diagram.....	965
Figure 34-3	CANFD TBUF Register Write Transmit Buffer and Schematic Diagram.....	968
Figure 34-4	Schematic Diagram of CAN RBUF Register Read Receive Buffer	969
Figure 34-5	Schematic Diagram of CAN ACF Register Access Filter Group	969
Figure 34-6	Schematic Diagram Of CANFD LBMI And LBME	973
Figure 34-7	TDC Function Diagram.....	978
Figure 35-1	MDIO Module Structure Diagram	1009
Figure 35-2	MDIO Frame Structure	1010
Figure 35-3	MDIO Application Block Diagram	1012
Figure 35-4	MDIO Slave Communication Flow Chart.....	1013
Figure 35-5	MDIO Slave Receiving Communication Timing Diagram.....	1013
Figure 35-6	MDIO Slave Sending Communication Timing Diagram	1014
Figure 35-7	MDIO Slave Continuous Communication Timing Diagram	1014
Figure 36-1	EXMC Architecture Diagram.....	1028
Figure 36-2	External Space Address Allocation.....	1030
Figure 36-3	SMC Address Space Division.....	1031
Figure 36-4	SMC Status Diagram.....	1035
Figure 36-5	SMC Initial Setting Process.....	1036
Figure 36-6	Basic Timing Of Single Read Operation (Asynchronous Mode (RSYN=0) & 16 -Bit Width (MW=01))	1038
Figure 36-7	Basic Timing Of Single Read Operation (Asynchronous Mode (RSYN=0) & 32 -Bit Width (MW=10))	1039
Figure 36-8	Basic Timing Of Single Read Operation (Synchronous mode (RSYN=1) & 16-bit width (MW=01))	1040
Figure 36-9	Basic Timing Of Single Read Operation (Synchronous mode (RSYN=1) & 32-bit width (MW=10))	1041
Figure 36-10	Basic Timing Of Single Read Action With Multiplexing Of Address And Data Lines (Asynchronous Mode (RSYN=0) & 32 -Bit Width (MW=10))	1042
Figure 36-11	Basic Timing Of Single Read Action With Multiplexing Of Address And Data Lines (Asynchronous Mode (RSYN=0) & 32 -Bit Width (MW=10))	1043
Figure 36-12	Basic Timing Of Single Write Operation (Asynchronous Mode (WSYN=0) & 16 -Bit Width (MW=01) & BLSS=1)	1044
Figure 36-13	Basic Timing Of Single Write Operation (Asynchronous Mode (WSYN=0) & 32 -Bit Width (MW=10) & BLSS=1)	1045
Figure 36-14	Basic Timing Of Single Write Operation (Synchronize Mode (WSYN=1) & 16 -Bit Width (MW=01) & BLSS=0)	1046

Figure 36-15 Basic Timing Of Single Write Operation (Synchronize Mode (WSYN=1) & 32 -Bit Width (MW=10) & BLSS=0)	1047
Figure 36-16 Basic Timing Of Single Write Action With Multiplexing Of Address And Data Lines (Asynchronous Mode (WSYN=0) & 16 -Bit Width (MW=01))	1049
Figure 36-17 Basic Timing Of Single Write Action With Multiplexing Of Address And Data Lines (Synchronize Mode (WSYN=1) & 16 -Bit Width (MW=01))	1051
Figure 36-18 Instance Of SMC Access In Synchronous Mode	1062
Figure 37-1 AES Encryption And Decryption Process Diagram	1063
Figure 37-2 AES Encryption Flow Chart	1064
Figure 37-3 HASH Algorithm Flow Chart	1075
Figure 37-4 TRNG System Block Diagram.....	1086
Figure 38-1 CRC Application Diagram.....	1089
Figure 39-1 FIR Basic Block Diagram	1093
Figure 40-1 DCU Hardware Trigger Start.....	1102
Figure 41-1 Schematic Diagram Of Sine Operation Angle Value.....	1111
Figure 42-1 Debugging Control Systems	1116
Figure 42-2 Debugging Control Systems	1117
Figure 42-3 JTAG-DP To SW-DP Switching Timing	1118
Figure 42-4 TTPIU Block Diagram	1126

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 is invalid
RW1	Software can read and write this bit, Writing 1 is valid and writing 0 is invalid
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 "Data Sheet".

Overview

The HC32A472 product family is a high-performance MCU based on ARM® Cortex®-M4 32-bit RISC CPU with a maximum operating frequency of 120 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 complete DSP instruction set. The core integrates the MPU unit and superimposes the DMAC dedicated MPU unit at the same time to ensure the safety of system operation.

HC32A472 series integrates high-speed on-chip memory, including a maximum of 512 KB Flash, a maximum of 64 KB SRAM, and 4 KB of power-off retention SRAM. Integrated Flash access acceleration unit to achieve single period program execution of the CPU on Flash. The polling bus matrix supports multiple bus hosts to access memory and peripherals simultaneously, improving performance. The bus host includes CPU, DMA, and USB dedicated DMA. 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 HC32A472 series integrates rich peripheral functions Includes: 3 independent 12-bit 2.5MSPS ADCs; 2 gain-adjustable PGAs; 1 12 - bit DAC; 2 high-speed voltage comparators (CMP); 10 multi-function PWM Timers (Timer6) support 20 complementary PWM outputs; 1 motor PWM Timer (Timer4) supports 3 complementary PWM outputs; 4 16-bit general-purpose Timers (TimerA) and 2 32-bit general Timers (TimerA) support orthogonal encoding input and 24-way Duty independently settable PWM outputs; 13 serial communication interfaces (I2C/UART/SPI); 1 QSPI interface; 3 CANs; USBFS Controller with built-in USBFS PHY; 1 external expansion bus controller, including SMC controller; 1 mathematical operation unit (MAU) and 4 filter mathematical accelerators (FMAC).

The HC32A472 product family supports wide voltage range (1.8-3.6V), wide temperature range (-40-105 °C) and low power mode.

Typical Application

The HC32A472series provides 100pin and 48-pin LQFP packages and is mainly used in application scenarios such as automotive body, intelligent network connection, and infotainment.

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-Map

Memory Classification		Start Address	End Address	Space size	Module * 3	Protection * 4	Description
System	-	0xE0100000	0xFFFFFFFF	511MB	Reserved	-	Custom space
	Private peripheral External bus	0xE00FF000	0xE00FFFFF	4KB	ROMTABLE	-	-
		0xE0042400	0xE00FEFFF	755KB	-	-	-
		0xE0042000	0xE00423FF	1KB	DBGC	-	Debug control register zone
		0xE0041000	0xE0041FFF	4KB	BLANK	-	-
		0xE0040000	0xE0040FFF	4KB	TPIU	-	-
	Private peripheral Internal bus	0xE000F000	0xE003FFFF	196KB	-	-	-
		0xE000E000	0xE000EFFF	4KB	SCS	-	System control space NVIC/MPU, etc.
		0xE0003000	0xE000DFFF	44KB	-	-	-
		0xE0002000	0xE0002FFF	4KB	FPB	-	-
		0xE0001000	0xE0001FFF	4KB	DWT	-	-
		0xE0000000	0xE0000FFF	4KB	ITM	-	-
External equipment	-	0xA0000000	0xDFFFFFFF	1024MB	Reserved	-	-
External memory	AHB5 Clock: HCLK	0x98000000	0x9FFFFFFF	128MB	QSPI	-	-
		0x88200000	0x97FFFFFF	254MB	BLANK	-	-
	AHB5 Clock: BCLK	0x88100000	0x881FFFFF	1MB	BLANK	-	-
		0x88000800	0x880FFFFF	1022KB	BLANK	-	-
		0x88000400	0x880007FF	1KB	BLANK	-	-
		0x88000000	0x880003FF	1KB	SMCR	-	-
		0x80000000	0x87FFFFFF	128MB	BLANK	-	-
Peripheral equipment	-	0x60000000	0x7FFFFFFF	512MB	SMC	-	-
		0x44000000	0x5FFFFFFF	448MB	Reserved	-	-
		0x42000000	0x43FFFFFF	32MB	PeriBitBand	-	-
		0x40100000	0x41FFFFFF	31MB	Reserved	-	-
		0x400C0000	0x400FFFFF	256KB	BLANK	-	-

Memory Classification		Start Address	End Address	Space size	Module * 3	Protection * 4	Description
	AHB2 Clock: PCLK1	0x40080000	0x400BFFFF	256KB	USBFS	-	-
		0x40078800	0x4007FFFF	30KB	BLANK	-	-
		0x40078400	0x400787FF	1KB	BLANK	-	-
		0x40078000	0x400703FF	1KB	CAN_1	-	-
	AHB3 Clock: PCLK1	0x40071800	0x40077FFF	26KB	BLANK	-	-
		0x40071400	0x400717FF	1KB	CAN_2	-	-
		0x40071000	0x400713FF	1KB	CAN_3	-	-
		0x40060000	0x40070FFF	68KB	BLANK	-	-
	AHB1 Clock: HCLK	0x40059000	0x4005FFFF	28KB	BLANK	-	-
		0x40058C00	0x40058FFF	1KB	FMAC_4	-	-
		0x40058800	0x40058BFF	1KB	FMAC_3	-	-
		0x40058400	0x400587FF	1KB	FMAC_2	-	-
		0x40058000	0x400583FF	1KB	FMAC_1	-	-
		0x40057C00	0x40057FFF	1KB	BLANK	-	-
		0x40057800	0x40057BFF	1KB	BLANK	-	-
		0x40057400	0x400577FF	1KB	BLANK	-	-
		0x40057000	0x400573FF	1KB	BLANK	-	-
		0x40056C00	0x40056FFF	1KB	BLANK	-	-
		0x40056800	0x40056BFF	1KB	DCU_3	-	-
		0x40056400	0x400567FF	1KB	DCU_2	-	-
		0x40056000	0x400563FF	1KB	DCU_1	-	-
		0x40055800	0x40055BFF	1KB	BLANK	-	-
		0x40055800	0x40055BFF	1KB	BLANK	-	-
		0x40055400	0x400557FF	1KB	PERIC	-	Peripheral module control register
		0x40055000	0x400553FF	1KB	MAU	-	-
		0x40054400	0x40054FFF	3KB	BLANK	-	No access
		0x40054000	0x400543FF	1KB	CMU	With protection	-
		0x40053800	0x40053FFF	2KB	GPIO	-	-
		0x40053400	0x400537FF	1KB	DMA_2	-	-
		0x40053000	0x400533FF	1KB	DMA_1	-	-
		0x40052000	0x40052FFF	4KB	BLANK	-	-
		0x40051000	0x40051FFF	4KB	INTC	With protection	-
		0x40050C00	0x40050FFF	1KB	KEYSCAN	-	-
		0x40050800	0x40050BFF	1KB	RAMIF	With protection	-
		0x40050400	0x400507FF	1KB	BLANK	-	-
		0x40050000	0x400503FF	1KB	DMPU	With protection	-

Memory Classification		Start Address	End Address	Space size	Module * 3	Protection * 4	Description
	APB4 Clock: PCLK3	0x4004FC00	0x4004FFFF	1KB	PLA	-	-
		0x4004F800	0x4004FBFF	1KB	BLANK	-	-
		0x4004F400	0x4004F7FF	1KB	BLANK	-	-
		0x4004F000	0x4004F3FF	1KB	BLANK	-	-
		0x4004EC00	0x4004EFFF	1KB	BLANK	-	-
		0x4004E800	0x4004EBFF	1KB	BLANK	-	-
		0x4004E400	0x4004E7FF	1KB	BLANK	-	-
		0x4004E000	0x4004E3FF	1KB	BLANK	-	-
		0x4004D000	0x4004DFFF	1KB	BLANK	-	-
		0x4004CC00	0x4004CFFF	1KB	PWC CMU	With protection	-
		0x4004C800	0x4004CBFF	1KB	BLANK	-	-
		0x4004C400	0x4004C7FF	1KB	WKTMR	-	-
		0x4004C000	0x4004C3FF	1KB	RTC	With protection	-
		0x4004AC00	0x4004BFFF	5KB	BLANK	-	-
		0x4004A800	0x4004ABFF	1KB	OTS	-	-
		0x4004A400	0x4004A7FF	1KB	BLANK	-	-
		0x4004A000	0x4004A3FF	1KB	BLANK	-	-
		0x40049C00	0x40049FFF	1KB	CTC	-	-
		0x40049800	0x40049BFF	1KB	BLANK	-	-
		0x40049400	0x400497FF	1KB	SWDT	With protection	-
		0x40049000	0x400493FF	1KB	WDT	With protection	-
		0x40048800	0x40048FFF	2KB	BLANK	-	-
		0x40048400	0x400487FF	1KB	FCM	-	-
		0x40048000	0x400483FF	1KB	FCG	With protection	-
Peripherals	APB3 Clock: PCLK4	0x40042400	0x40047FFF	23KB	BLANK	-	-
		0x40042000	0x400423FF	1KB	TRNG	With protection	-
		0x40041C00	0x40041FFF	1KB	BLANK	-	-
		0x40041800	0x40041BFF	1KB	BLANK	-	-
		0x40041400	0x400417FF	1KB	BLANK	-	-
		0x40041000	0x400413FF	1KB	DAC_1	-	-
		0x40040C00	0x40040FFF	1KB	BLANK	-	-
		0x40040800	0x40040BFF	1KB	ADC_3	-	-
		0x40040400	0x400407FF	1KB	ADC_2	-	-
		0x40040000	0x400403FF	1KB	ADC_1	-	-
	APB5 Clock: PCLK0	0x4003C000	0x4003FFFF	16KB	TMR6	-	-
		0x4003BC00	0x4003BFFF	1KB	I2C_3	-	-

Memory Classification		Start Address	End Address	Space size	Module * 3	Protection * 4	Description
Peripherals		0x4003B800	0x4003BBFF	1KB	I2C_2	-	-
		0x4003B400	0x4003B7FF	1KB	I2C_1	-	-
		0x4003B000	0x4003B3FF	1KB	BLANK	-	-
		0x4003AC00	0x4003AFFF	1KB	TimerA_4	-	-
		0x4003A800	0x4003ABFF	1KB	TimerA_3	-	-
		0x4003A400	0x4003A7FF	1KB	TimerA_2	-	-
		0x4003A000	0x4003A3FF	1KB	TimerA_1	-	-
		0x40038800	0x40039FFF	6KB	BLANK	-	-
		0x40038400	0x400387FF	1KB	CMP_1	-	-
		0x40038000	0x400383FF	1KB	Timer4_1	-	-
	APB2 Clock: PCLK1	0x40028800	0x40027FFF	62KB	BLANK	-	-
		0x40028400	0x400287FF	1KB	MDIO	-	-
		0x40028000	0x400283FF	1KB	BLANK	-	-
		0x40027C00	0x40027FFF	1KB	BLANK	-	-
		0x40027800	0x40027BFF	1KB	BLANK	-	-
		0x40027400	0x400277FF	1KB	BLANK	-	-
		0x40027000	0x400273FF	1KB	BLANK	-	-
		0x40026C00	0x40026FFF	1KB	BLANK	-	-
		0x40026800	0x40026BFF	1KB	BLANK	-	-
		0x40026400	0x400267FF	1KB	TimerA_6	-	-
		0x40026000	0x400263FF	1KB	TimerA_5	-	-
		0x40025800	0x40025FFF	2KB	BLANK	-	-
		0x40025400	0x400257FF	1KB	Timer2_4	-	-
		0x40025000	0x400253FF	1KB	Timer2_3	-	-
		0x40024C00	0x40024FFF	1KB	Timer2_2	-	-
		0x40024800	0x40024BFF	1KB	Timer2_1	-	-
		0x40024400	0x400247FF	1KB	Timer0_2	-	-
		0x40024000	0x400243FF	1KB	Timer0_1	-	-
		0x40022800	0x40023FFF	6KB	BLANK	-	-
		0x40022400	0x400227FF	1KB	BLANK	-	-
		0x40022000	0x400223FF	1KB	BLANK	-	-
		0x40021C00	0x40021FFF	1KB	BLANK	-	-
		0x40021800	0x40021BFF	1KB	BLANK	-	-
		0x40021400	0x400217FF	1KB	USART_6	-	-
		0x40021000	0x400213FF	1KB	USART_5	-	-
		0x40020C00	0x40020FFF	1KB	USART_4	-	-
		0x40020800	0x40020BFF	1KB	BLANK	-	-
		0x40020400	0x400207FF	1KB	SPI_4	-	-

Memory Classification		Start Address	End Address	Space size	Module * 3	Protection * 4	Description
	APB1 Clock: PCLK1	0x40020000	0x400203FF	1KB	SPI_3	-	-
		0x4001E800	0x4001FFFF	6KB	BLANK	-	-
		0x4001E400	0x4001E7FF	1KB	BLANK	-	-
		0x4001E000	0x4001E3FF	1KB	BLANK	-	-
		0x4001DC00	0x4001DFFF	1KB	BLANK	-	-
		0x4001D800	0x4001DBFF	1KB	BLANK	-	-
		0x4001D400	0x4001D7FF	1KB	USART_3	-	-
		0x4001D000	0x4001D3FF	1KB	USART_2	-	-
		0x4001CC00	0x4001CFFF	1KB	USART_1	-	-
		0x4001C800	0x4001CBFF	1KB	BLANK	-	-
		0x4001C400	0x4001C7FF	1KB	SPI_2	-	-
		0x4001C000	0x4001C3FF	1KB	SPI_1	-	-
		0x40018000	0x4001BFFF	16KB	BLANK	-	-
		0x40017C00	0x40017FFF	1KB	EMB	-	-
		0x40010C00	0x40017BFF	28KB	BLANK	-	-
		0x40010800	0x40010BFF	1KB	AOS	-	Internal trigger event register zone
		0x40010400	0x400107FF	1KB	EFM	with protection	-
		0x40010000	0x400103FF	1KB	BLANK	-	-
	AHB4 Clock: PCLK1	0x40009400	0x4000FFFF	27KB	BLANK	-	-
		0x40009000	0x400093FF	1KB	BLANK	-	-
		0x40008C00	0x40008FFF	1KB	CRC	With protection	-
		0x40008800	0x40008BFF	1KB	BLANK	-	-
		0x40008400	0x400087FF	1KB	HASH	With protection	-
		0x40008000	0x400083FF	1KB	AES	With protection	-
		0x40000000	0x40007FFF	32KB	Reserved	-	-
SRAM	SRAM Clock: HCLK	0x24000000	0x3FFFFFFF	448MB	Reserved	-	-
		0x22000000	0x23FFFFFF	32MB	SRAMBitBand	-	-
		0x20100000	0x21FFFFFF	31MB	Reserved	-	-
		0x200D9000	0x200FFFFF	156KB	Reserved	-	-
		0x200F0000	0x200F0FFF	4KB	Ret_SRAM	With protection	-
		0x20060000	0x200EFFFF	576KB	Reserved	-	-
		0x20040000	0x2005FFFF	128KB	Reserved	-	-
		0x20008000	0x2003FFFF	224KB	Reserved	-	-
		0x20000000	0x20007FFF	32KB	SRAM0	-	ECC RAM

Memory Classification		Start Address	End Address	Space size	Module * 3	Protection * 4	Description
CODE	SRAM Clock: HCLK	0x1FFF8000	0x1FFFFFFF	32KB	SRAMH	-	-
		0x03006000	0x1FFF7FFF	463.94MB	BLANK	-	-
	OTP, Flash Clock: HCLK	0x0300400C	0x03005FFF	8180B	BLANK	-	-
		0x03004000	0x0300400B	12 B	Data security protection	-	Used to configure data security protection
		0x03002004	0x03003FFF	8187B	BLANK	-	-
		0x03002000	0x03002003	4 B	Boot swap	-	-
		0x03001ADC	0x03001FFF	1316B	BLANK	-	-
		0x03000000	0x03001ADB	6876B	OTP	-	-
	-	0x02100000	0x02FFFFFF	15M	BLANK	-	-
	REMAP Clock: HCLK	0x02080000	0x020FFFFF	512KB	REMAP1	-	Address remapping zone 1
		0x02000000	0x0207FFFF	512KB	REMAP0	-	Address remapping zone 0
	-	0x00080000	0x01FFFFFF	31.5M	BLANK	-	-
	Flash Clock: HCLK	0x00040000	0x0007FFFF	256KB	Embedded Flash 1	-	-
		0x00000000	0x0003FFFF	256KB	Embedded Flash 0	-	-

*1 Please refer to the ARM Cortex®-M4 Instructions Manual "Memory Systems".

*2 For bus instruction, please refer to the [Bus Architecture (BUS)]Bus chapter.

*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 instructions, refer to the [Memory Protection Unit (MPU)] chapter.

1.2 External Space Mapping

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

Table 1-2 QSPI Address Space Allocation

QSPI	0x98000000	0x9FFFFFFF	128MB
QSPI I/O registers	0x9C000000	0x9FFFFFFF	64MB
External QSPI device	0x98000000	0x9BFFFFFF	64MB

1.3 Bit Field Space

The Cortex-M4 memory map consists of two bit field regions. These regions map each word in the memory alias region to the corresponding bits in the memory bit field 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 field zone, which enables read and write operations of a single bit field. These operations are only available for Cortex-M4 access and are not valid for other bus host interfaces such as DMA.

1.4 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 1-3 Target Address Configuration Instance

Register Setting	Remapping Address (CPU Address - CPUADDR [31:0])	Source Address		
		High 3-bit Address	Medium Address	Low Address
RMSIZE [4:0]=01110 case (Remapping space: 16K)	0x02000000~0x02003FFF	All 0	RMTADDR [16:2]	CPUADDR [13:0]
RMSIZE [4:0]=01111 case (Remapping space: 32K)	0x02000000~0x02007FFF	All 0	RMTADDR [16:3]	CPUADDR [14:0]
RMSIZE [4:0]=10000 case (Remapping space: 64K)	0x02080000~0x0208FFFF	All 0	RMTADDR [16:4]	CPUADDR [15:0]
RMSIZE [4:0]=10001 case (Remapping space: 128K)	0x02080000~0x0209FFFF	All 0	RMTADDR [16:5]	CPUADDR [16:0]

For example, use the function of remapping address 0, set the source address as the main flash memory FLASH address 0x00008000, and the remapping space is 32K, and the register MMF_REMCR0 needs to be set to 0x8000800F.

Use the remapping address 1 function, set the source address to the high-speed SRAM address 0x1FFF8000, the remapping space to 16K, and set the register MMF_REMCR1 to 0x9FFF800E.

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

1.5 Remapping Register

There are three registers in the remapping module. The address space is as follows:

Register base address: 0x40010500

Table 1-4 Register List

Register Name	Symbol	Offset Address	Bit Width	Reset Value
Access Protection Register	MMF_REMPRT	0x0000	32	0x00000000
Remap Control Register 0	MMF_REMCRO	0x0004	32	0x00000000
Remap control register 1	MMF_REMCRI	0x0008	32	0x00000000

1.5.1 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
-															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MMF_REMPRT [15:0]															

Bit	Flag	Bit Name	Function	Read and write
b31~16	Reserved	-	Read as "0", write as "0"	R
b15~0	MMF_REMPRT [15:0]	Protection register	Registers MMF_REMCRO and MMF_REMCRI are write-protected: First write 0x0123 to MMF_REMPRT [15:0] and then write 0x3210 to release the protection; When the registers MMF_REMCRO and MMF_REMCRI are in the write protection state, the read register is 0 When the registers MMF_REMCRO and MMF_REMCRI are released from the write protection state, the read register is 1	R/W

1.5.2 Remap Control Register (MMF_REMC Rx) (x=0, 1)

Reset value: 0x0000_0000

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

Bit	Flag	Bit Name	Function	Read and write
b31	EN	Remapping valid bit	0: no remapping 1: remapping is valid	R/W
b30~29	Reserved	-	Read as "0", write as "0"	R
b28~12	RMTADDR [16:0]	Source Address	The effective number of digits is related to the setting of RMSIZE [4:0]. Please refer to Table 1-3 for settings.	R/W
b11~b5	Reserved	-	Read as "0", write as "0"	R
b4~b0	RMSIZE [4:0]	Remapping space	Set remapping space 00000~ 01011: Reserved, setting is prohibited	R/W
			01100: 4KB	
			01101: 8KB	
			01110: 16KB	
			01111: 32KB	
			10000: 64KB	
			10001: 128KB	
			10010: 256KB	
			10011: 512KB (256KB products cannot be set) 10100~ 11111: Reserved, setting is prohibited	

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_1 bus, system DMA_2 bus
 - USBFS_DMA bus
- Slave bus
 - Flash ICODE bus
 - Flash DCODE bus
 - Flash MCODE bus (bus for other hosts other than CPU to access Flash)
 - High-speed SRAM (SRAMH 32KB) bus
 - System SRAM (SRAM0 32KB) bus
 - System SRAM(Ret_SRAM 4KB) bus
 - APB1 peripheral bus (EMB/SPI/USART/AOS/EFM)
 - APB2 peripheral bus (Timers/SPI/USART/MDIO)
 - APB3 peripheral bus (ADC/ DAC/TRNG)
 - APB4 peripheral bus (FCM/WDT/SWDT/PWC/CTC/OTS/RTC/WKTM/PLA)
 - APB5 peripheral bus (Timers/CMP/I2C)
 - AHB1 peripheral bus (DMPU/ KEYSCAN/ INTC/ DCU/ GPIO/ DMA/ CMU/ MAU/ FMAC)
 - AHB2 peripheral bus (CAN/USBFS)
 - AHB3 peripheral bus (CAN)
 - AHB4 peripheral bus (AES/HASH/CRC)
 - AHB5 peripheral bus (SMC/ SMCR/ QSPI)

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

2.2 Bus Architecture

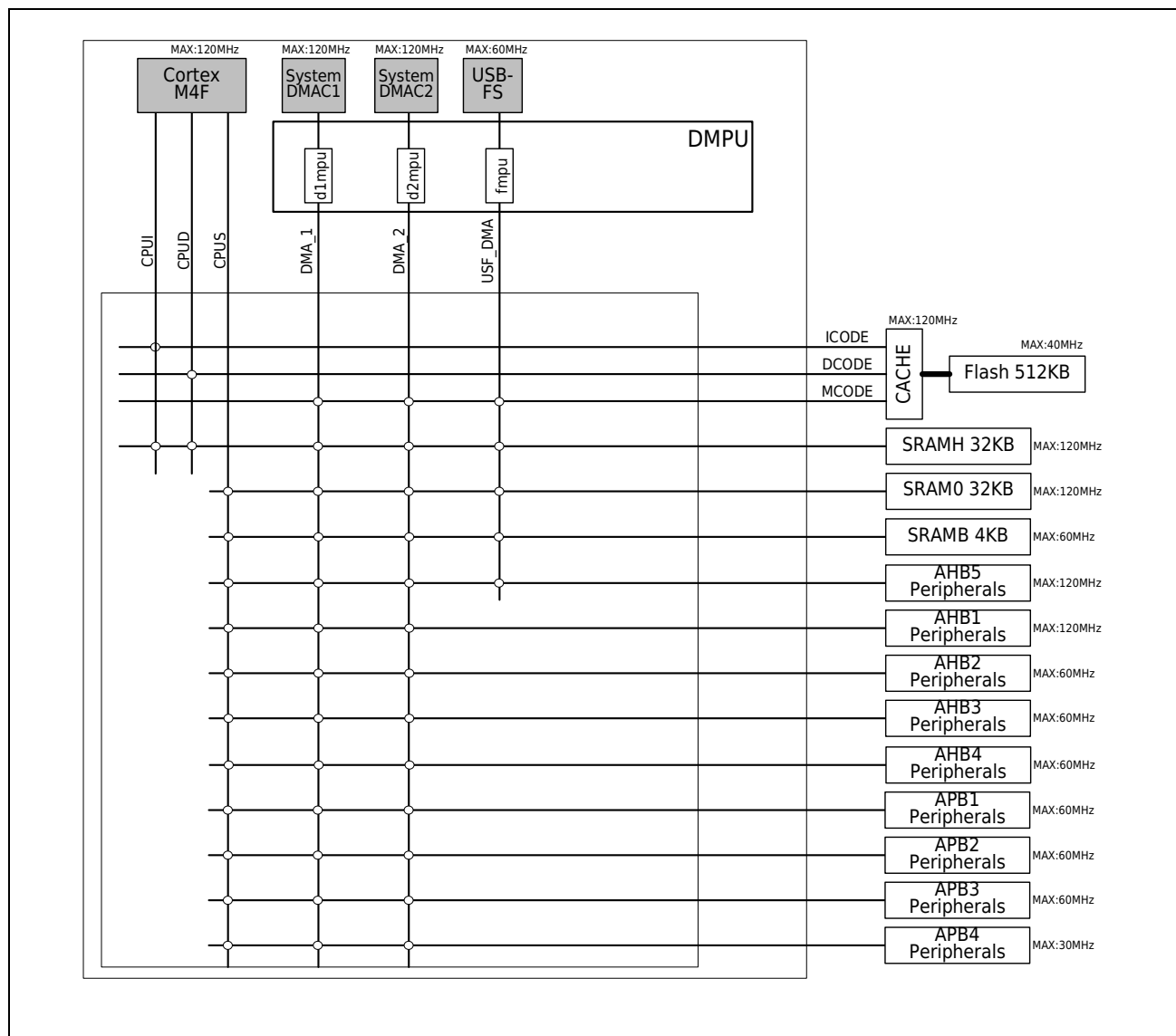


Figure 2-1 Bus Architecture Diagram

The bus matrix is used for access arbitration management among the host 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 all peripherals of SRAMH/SRAM0/Ret_SRAM and external expansion space.

- DMA_1 bus, DMA_2 bus

System DMA_1/system DMA_2 dedicated bus, DMA_1/DMA_2 accesses data memory and peripherals through this bus. The access objects are all peripherals of Flash/SRAMH/SRAM0/Ret_SRAM and external expansion space.

- USBFS-DMA bus

DMA dedicated bus for USBFS, through which USBFS accesses all memory spaces. The access objects are Flash/SRAMH/SRAM0/Ret_SRAM and external expansion space.

2.3 Bus Function

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

Through the bus matrix, when the access targets of different host buses do not conflict, each access can be carried out simultaneously. For example, CPU-I accesses Flash, CPU-D accesses SRAMH, CPUS accesses APB peripherals, DMA_1 accesses SRAM0, DMA_2 accesses Ret_SRAM, and USBFS-DMA accesses the external expansion space of AHB5. These accesses can be performed simultaneously.

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

✓: Set X: Clear -: No change

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.

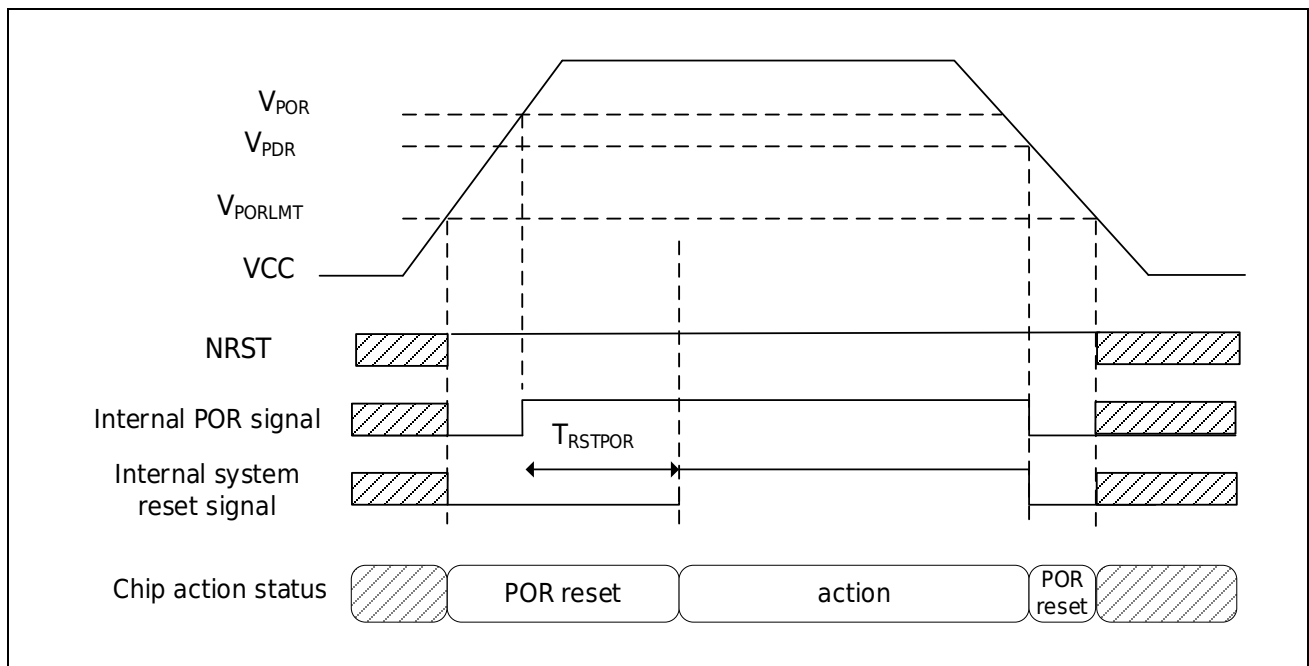


Figure 3-1 Power-on Reset

3.3.2 NRST Pin Reset

Pin reset is a reset caused by the NRST pin being driven low, and the reset timing is shown in the figure Figure 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.

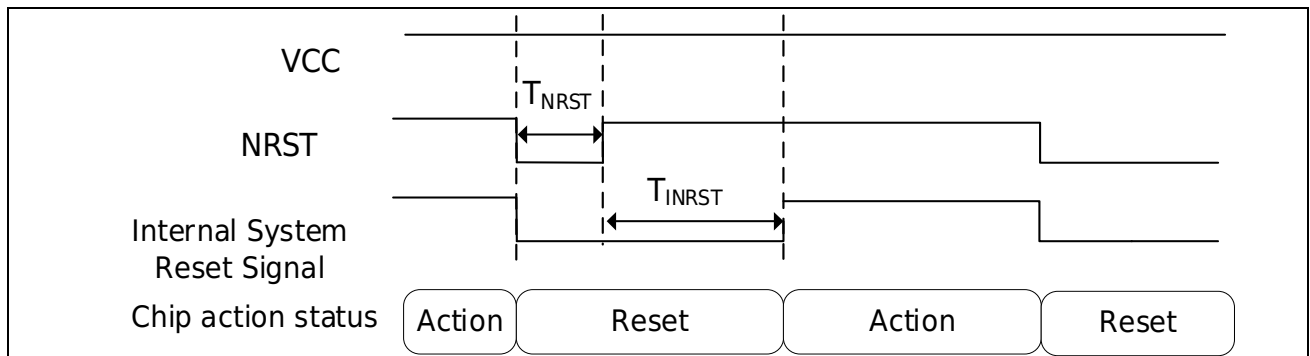


Figure 3-2 NRST Reset Timing

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 undervoltage, please refer to [Brown-out Reset (BOR) Description].

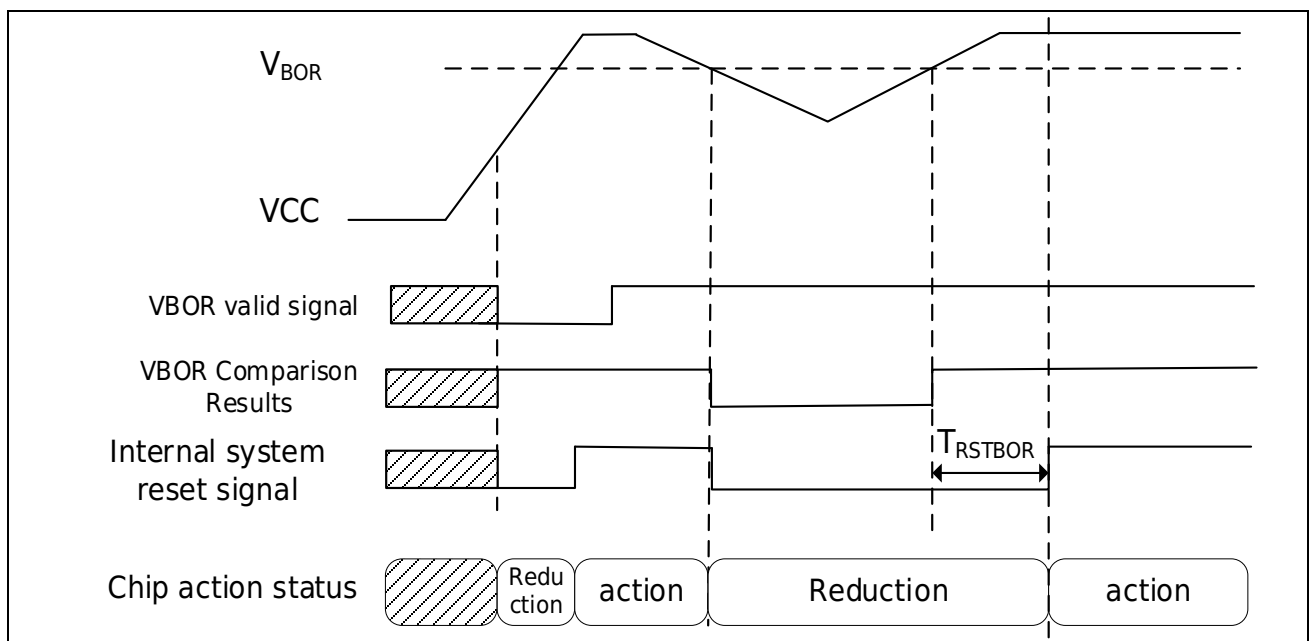


Figure 3-3 Brownout Reset

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.

After the programmable voltage detection 1 is enabled and reset is enabled, if VCC is lower than the monitoring voltage of the programmable voltage detection 1, a reset of the programmable voltage detection 1 is 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}).

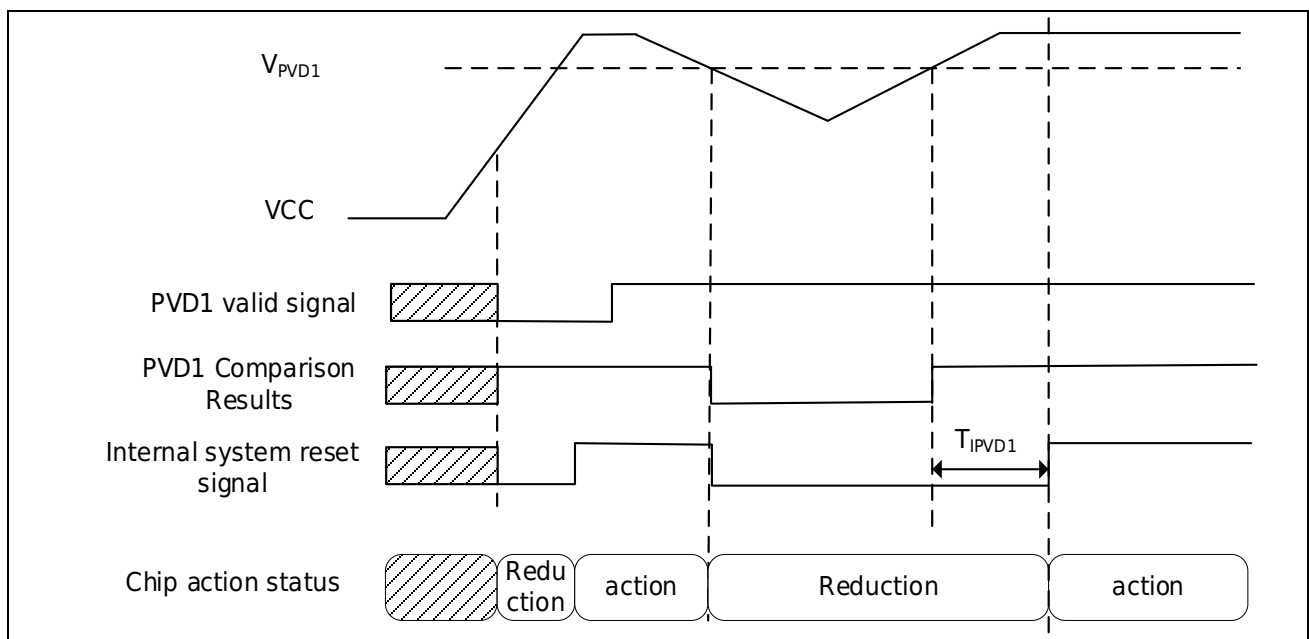


Figure 3-4 Programmable Voltage Detection 1 Reset

After the programmable voltage detection 2 is enabled and reset is enabled, if VCC is lower than the monitoring voltage of the programmable voltage detection 2, a reset of the programmable voltage detection 2 is 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}).

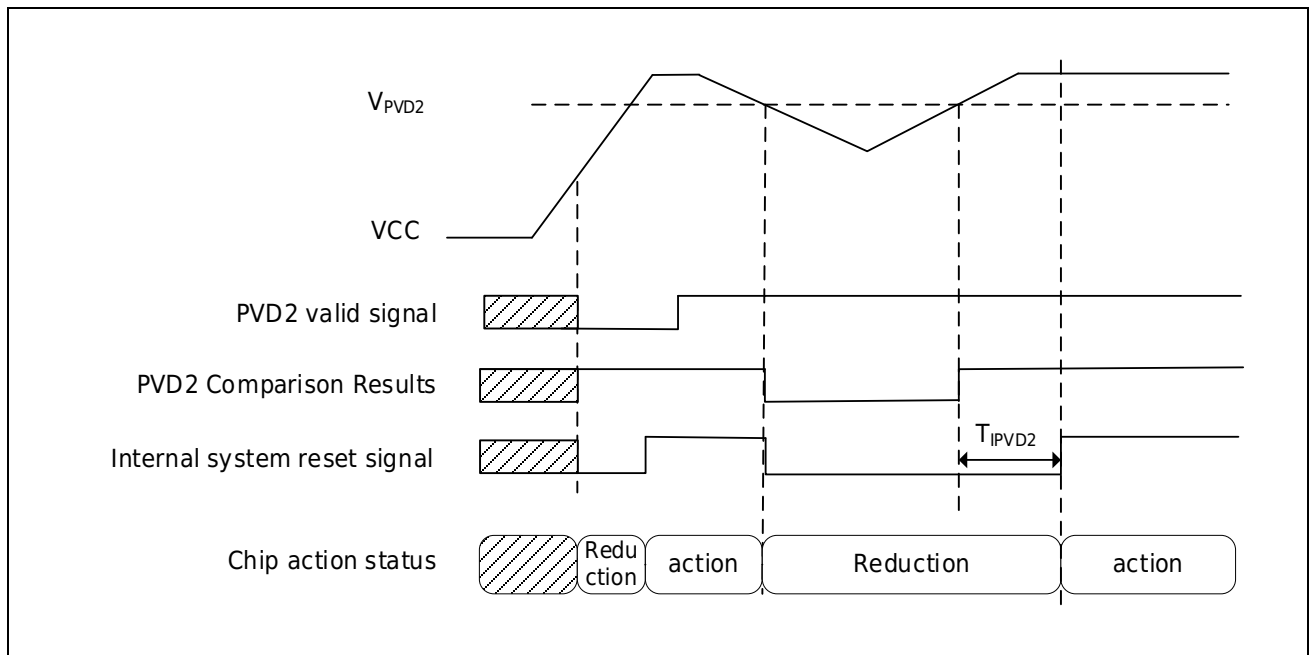


Figure 3-5 Programmable Voltage Detection 2 Reset

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, Dedicated Watchdog Reset

The watchdog reset is an internal reset caused by the watchdog timer, and the dedicated watchdog reset is an internal reset caused by the dedicated 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 dedicated watchdog reset is set to be valid, the watchdog reset will be generated when the dedicated 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.

`PWC_PWRC6.WDRTNE=1`, the port status of the chip remains the status before the watchdog reset. the port hold state can be released by setting `PWC_PWRC6.IORTNCLR`. When a watchdog reset occurs when `PWC_PWRC6.WDRTNE=0`, the port state of the chip will be initialized to a high-impedance state.

`PWC_PWRC6.WDRTNE=1`, the port status of the chip remains the status before the special watchdog reset. After a special watchdog reset occurs, the port hold state can be released by

setting PWC_PWRC6.IORTNCLR. When a special watchdog reset is generated when PWC_PWRC6.WDRTNE=0, the port state of the chip will be initialized to a high-impedance state.

PWC_PWRC6.WDRDAC=1, the DAC module in the chip maintains the state before the watchdog reset. the DAC module can be reset by setting RMU_FRST3.DAC1. PWC_PWRC6.WDRDAC=0, a watchdog reset is generated and the DAC module in the chip is reset.

PWC_PWRC6.WDRDAC=1 generates a special watchdog reset, the DAC module in the chip maintains the state before the special watchdog reset. After the special watchdog reset occurs, the DAC module can be reset by clearing RMU_FRST3.DAC1. PWC_PWRC6.WDRDAC=0, a special watchdog reset is generated and the DAC module in the chip is reset.

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

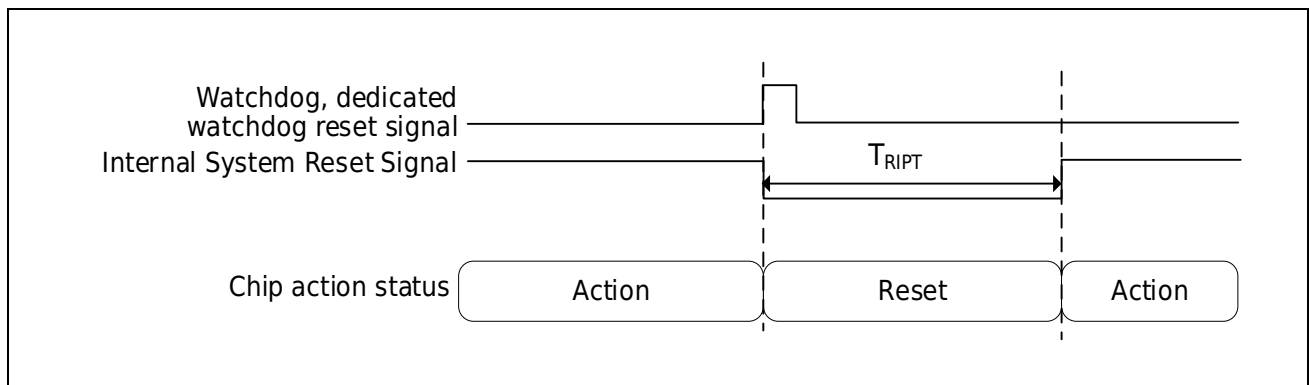


Figure 3-6 Watchdog and Dedicated Watchdog Reset

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].

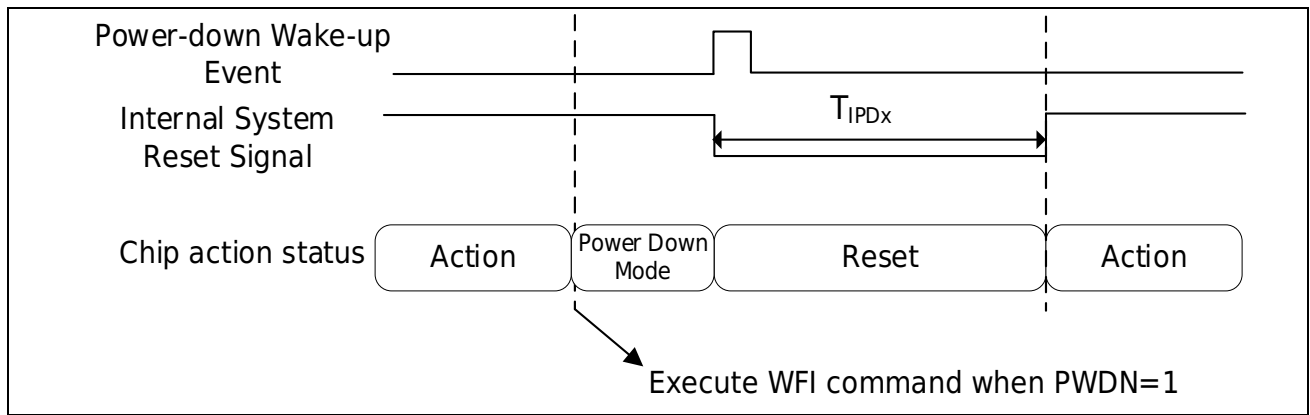


Figure 3-7 Power-down Wake-up Reset

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.

PWC_PWRC6.SWRTNE=1 , the port status of the chip remains the status before the software reset. After a software reset occurs, the port hold state can be released by setting PWC_PWRC6.IORTNCLR. When a software reset occurs when PWC_PWRC6.SWRTNE=0 , the port state of the chip will be initialized to a high-impedance state.

PWC_PWRC6.SWRDAC=1 , the DAC module in the chip maintains the state before the software reset. After a software reset occurs, the DAC module can be reset by setting RMU_FRST3.DAC1/. PWC_PWRC6.SWRDAC=0 , the DAC module in the chip is reset.

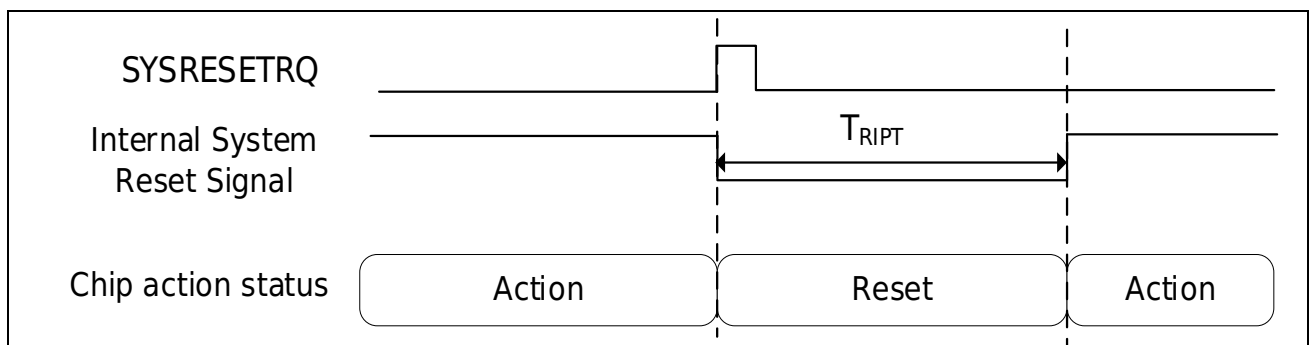


Figure 3-8 Software Reset

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)].

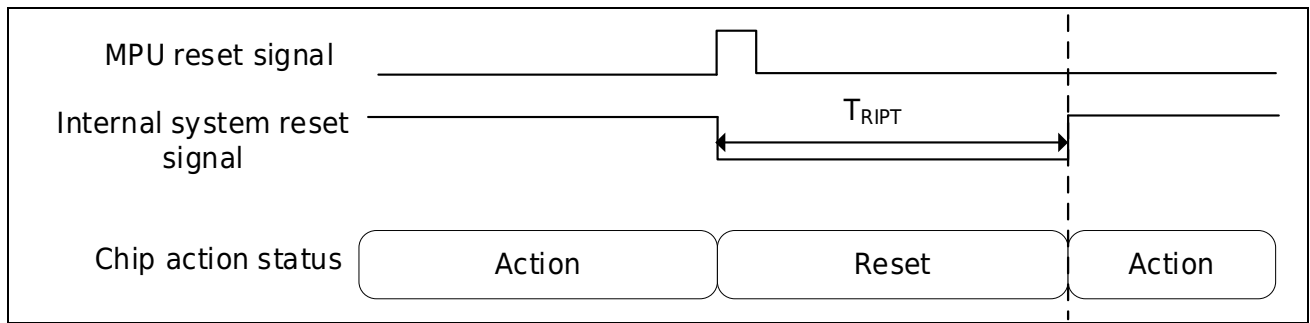


Figure 3-9 MPU Error Reset

3.3.9 RAM Parity Reset

When an error occurs in the RAM parity check, a reset of the RAM parity check is generated, and the timing is as follows 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)].

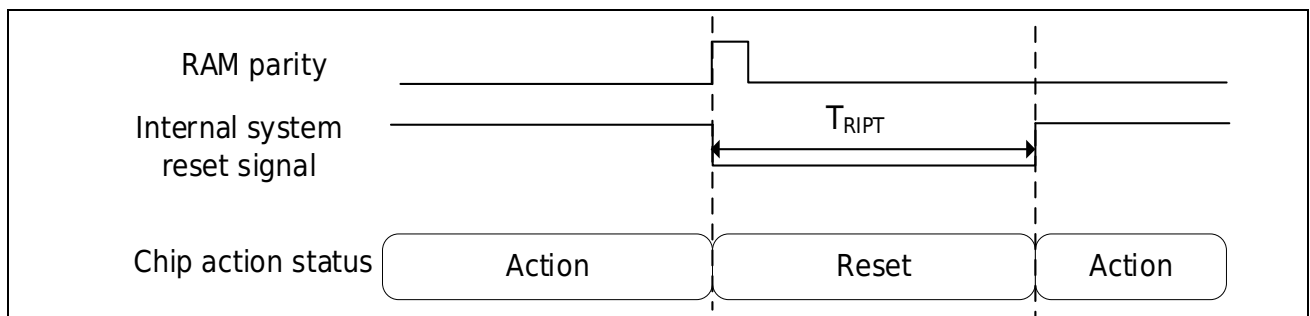


Figure 3-10 RAM Parity Reset

3.3.10 RAMECC Reset

When an error occurs in the RAMECC verification, a RAMECC reset is generated, and the timing is as follows 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)].

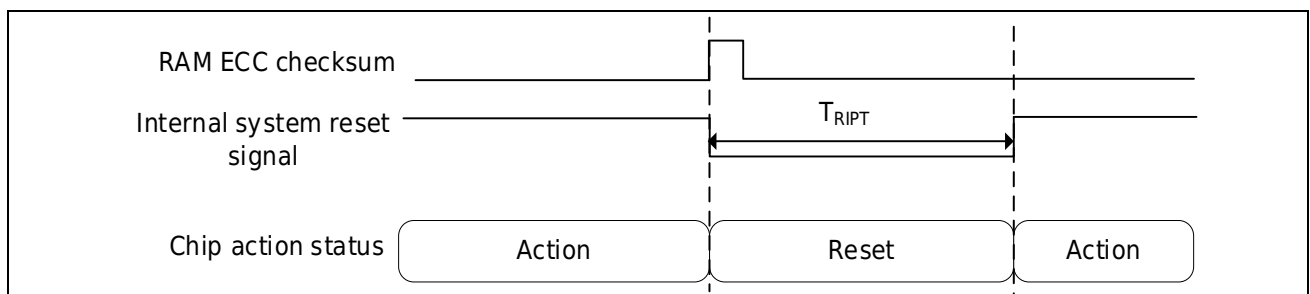


Figure 3-11 RAMECC Reset

3.3.11 Clock Frequency Exception Reset

When the built-in FCM module of the chip detects that the clock frequency is abnormal, if it is set to reset, it will generate a clock frequency abnormal reset. The timing is as follows 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.

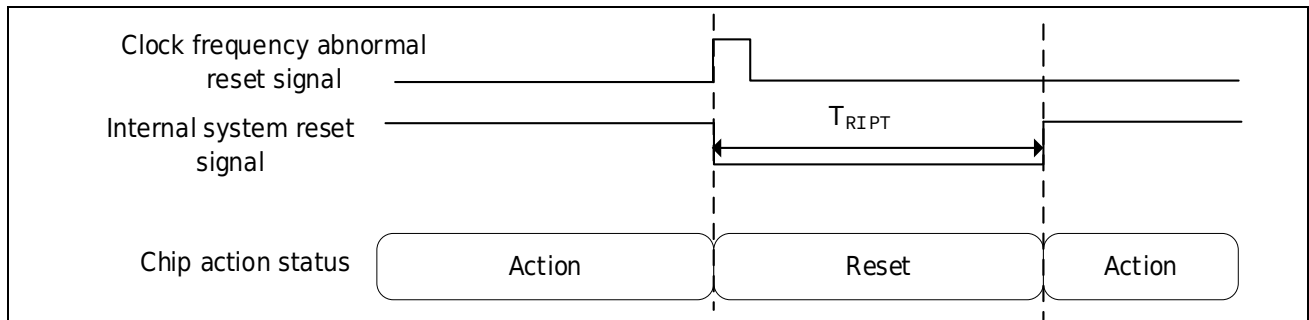


Figure 3-12 Clock Frequency Abnormal Reset

3.3.12 External High Speed Oscillator Abnormal Stop 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.

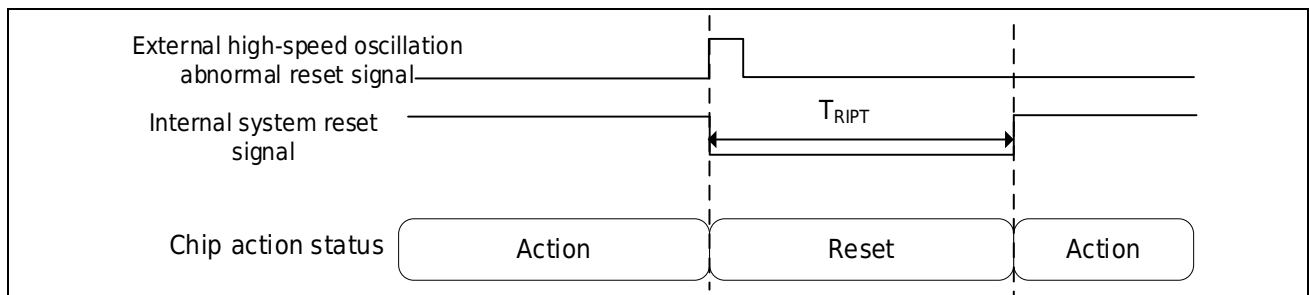


Figure 3-13 External High-speed Oscillation Abnormal Reset

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 periods delay, the timing is as follows Figure 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.

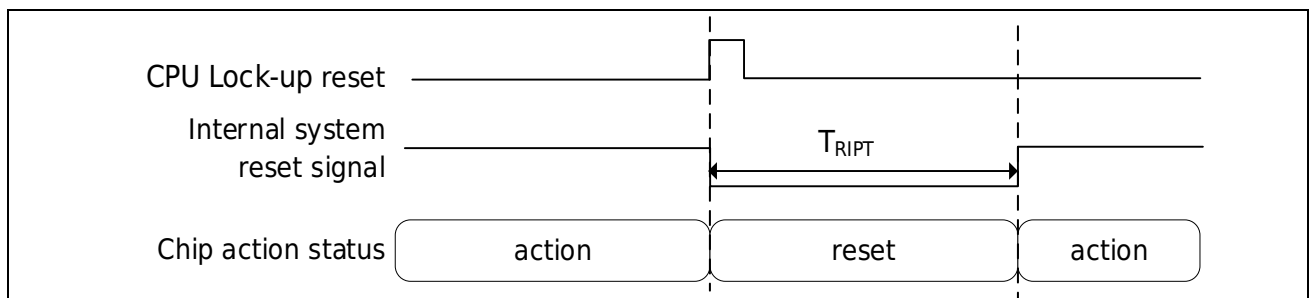


Figure 3-14 Cortex-M4 Lockup Reset

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 periods before reading the RMU_RSTF0 register again.

3.3.15 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, PWC_PWRC1.PDTS, 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_PDWKFO, PWC_PDWKF1	All reset sources except power-down wake-up reset
	PWC_PVDLCR, PWC_PVDCR1, PWC_PVDFCR, PWC_PVDCCR0, PWR_PWRC2, PWR_PWRC4	1. Power-on Reset 2. Pin Reduction 3. Brown-out Reset 4. Watchdog reset 5. Dedicated watchdog reset

Modules	Register	Reset Source
		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. Dedicated watchdog reset 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 Function Reset Control

The chip presets the function reset control register RMU_FRSTx (x=1,2,3,4) for most functions. 0 to the reset bit of the corresponding function , the function can be reset to its initial state. RMU_FRSTx , refer to the register section of this chapter.

3.5 Register Description

The list of registers is shown Table 3-4 below.

BASE ADDR: 0x4004CCE0

Table 3-4 List of RMU Registers

Register Name	Symbol	Offset Address	Bit Width	Reset Value
Function reset control register 0	RMU_FRST0	0xE0	32	0xFFFFFFFF
Function reset control register 1	RMU_FRST1	0xE4	32	0xFFFFFFFF
Function reset control register 2	RMU_FRST2	0xE8	32	0xFFFFFFFF
Function reset control register 3	RMU_FRST3	0xEC	32	0xFFFFFFFF
Reset control register	RMU_PRSTCR0	0x10	32	0x40
Reset flag register 0	RMU_RSTF0	0x14	32	Different Reset Values According to Different Reset Methods

3.5.1 Function Reset Control 0 (RMU_FRST0)

Reset value: 0xFFFFFFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	DCU3	DCU2	DCU1	CRC	TRNG	HASH	AES	MAU	CTC	AOS	FCM
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DMA2	DMA1	KEY	-	PLA	-	-	-	-	-	-	-	-	-	-	-

Bit	Flag	Bit Name	Function	Read and write
b31-b27	Reserved	-	Read at "11111", write at "11111"	R/W
b26	DCU3	DCU3 reset control	0: Digital computing unit DCU3 function reset is valid 1: Digital computing unit DCU3 function reset released	R/W
b25	DCU2	DCU2 reset control	0: Digital computing unit DCU2 function reset is valid 1: Digital computing unit DCU2 function reset released	R/W
b24	DCU1	DCU1 reset control	0: Digital computing unit DCU1 function reset is valid 1: Digital computing unit DCU1 function reset released	R/W
b23	CRC	CRC Reset control	0: CRC Function Reset Valid 1: CRC Function Reset Disable	R/W
b22	TRNG	TRNG reset control	0: The TRNG function in the encryption co-processing module CPM is reset and valid 1: The TRNG function in the cryptographic co-processing module CPM is reset and released	R/W
b21	HASH	HASH reset control	0: The HASH function of the secure hash algorithm module of the encryption co-processing module CPM is reset and valid 1: The HASH function of the secure hash algorithm module of the encryption co-processing module CPM is reset and released	R/W
b20	AES	AES reset control	0: The encryption and decryption algorithm processor AES function in the encryption co-processing module CPM is reset and valid 1: The AES function of the encryption and decryption algorithm processor in the encryption co-processing module CPM is reset and released	R/W
b19	MAU	MAU reset control	0: Mathematical operation unit MAU function reset is valid 1: Mathematical operation unit MAU function reset release	R/W
b18	CTC	CTC reset control	0: Internal clock calibrator CTC function reset valid 1: Internal clock calibrator CTC function reset release	R/W
b17	AOS	AOS reset control	0: Automatic operation system AOS function reset valid 1: Automatic operation system AOS function reset Disable	R/W
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 is reset and released	R/W
b15	DMA2	DMA2 reset control	0: DMA controller DMA2 function reset is valid 1: DMA controller DMA2 function reset release	R/W
b14	DMA1	DMA1 reset control	0: DMA controller DMA1 function reset is valid 1: DMA controller DMA1 function reset release	R/W
b13	KEY	KEYSCAN reset control	0: The KEYSCAN function of the keyboard scan control module is reset and valid 1: The KEYSCAN function of the keyboard scan control module is reset and released	R/W
b12	Reserved	-	Read at "1", write at "1"	R/W
b11	PLA	PLA reset control	0: PLA Function Reset Valid 1: PLA Function Reset Disable	R/W
b10-b0	Reserved	-	When reading, it is "111111111111", and when writing, it is "111111111111"	R/W

3.5.2 Function Reset Control 1 (RMU_FRST1)

Reset value: 0xFFFFFFFF

b31	b30	b29	b28	b27	b26	b25	b24
-	-	-	-	FMAC4	FMAC3	FMAC2	FMAC1
b23	b22	b21	b20	b19	b18	b17	b16
-	USBFS	-	-	SPI4	SPI3	SPI2	SPI1
b15	b14	b13	b12	b11	b10	b9	b8
-	-	-	-	-	-	-	MDIO
b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	QSPI	-	-	-

Bit	Flag	Bit Name	Function	Read and write
b31-b28	Reserved	-	Read at "1", write at "1"	R/W
b27	FMAC4	FMAC4 reset control	0: Filter math accelerator FMAC unit 4 function reset is valid 1: Filter math accelerator FMAC unit 4 function reset released	R/W
b26	FMAC3	FMAC3 reset control	0: Filter math accelerator FMAC unit 3 function reset is valid 1: Filter math accelerator FMAC unit 3 function reset released	R/W
b25	FMAC2	FMAC2 reset control	0: Filter math accelerator FMAC unit 2 function reset is valid 1: Filter math accelerator FMAC unit 2 function reset released	R/W
b24	FMAC1	FMAC1 reset control	0: Filter math accelerator FMAC unit 1 function reset is valid 1: Filter math accelerator FMAC unit 1 function reset released	R/W
b23	Reserved	-	Read at "1", write at "1"	R/W
b22	USBFS	USBFS reset control	0: USB2.0 full-speed module USBFS function reset is valid 1: USB2.0 full-speed module USBFS function reset release	R/W
b21~b20	Reserved	-	Read at "1", write at "1"	R/W
b19	SPI4	SPI4 reset control	0: Serial peripheral interface SPI unit 4 function reset is valid 1: SPI unit 4 function reset released	R/W
b18	SPI3	SPI3 reset control	0: Serial peripheral interface SPI unit 3 function reset is valid 1: SPI unit 3 function reset released	R/W
b17	SPI2	SPI2 reset control	0: Serial peripheral interface SPI unit 2 function reset is valid 1: SPI unit 2 function reset released	R/W
b16	SPI1	SPI1 reset control	0: Serial peripheral interface SPI unit 1 function reset is valid 1: SPI unit 1 function reset released	R/W
b15~b9	Reserved	-	Read at "1", write at "1"	R/W
b8	MDIO	MDIO reset control	0: MDIO Function Reset Valid 1: MDIO Function Reset Disable	R/W
b7-b4	Reserved	-	Read at "1", write at "1"	R/W
b3	QSPI	QSPI reset control	0: Four-wire serial peripheral interface QSPI function reset is valid 1: Four-wire serial peripheral interface QSPI function reset release	R/W
b2-b0	Reserved	-	Read at "111", write at "111"	R/W

3.5.3 Function Reset Control 2 (RMU_FRST2)

Reset value: 0xFFFFFFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	TMRA	-	-	-	TMR2
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EMB	-	-	TMR0	-	TMR4	-	-	-	-	-	-	-	-	-	TMR6

Bit	Flag	Bit Name	Function	Read and write
b31~b21	Reserved	-	Read at "1111111111", write at "1111111111"	R/W
b20	TMRA	TIMERA reset control	0: TimerA unit 1-6 Function Reset Valid 1: TimerA Unit 1-6 Function Reset Disable	R/W
b19-b17	Reserved	-	Read at "111", write at "111"	R/W
b16	TMR2	TIMER2 reset control	0: TIMER2 Unit 1- 4 Function Reset Valid 1: TIMER2 Unit 1- 1 Function Reset Disable	R/W
b15	EMB	EMB reset control	0: Emergency brake module EMB Function Reset Valid 1: Emergency brake module EMB Function Reset Disable	R/W
b14-b13	Reserved	-	Read at "11", write at "11"	R/W
b12	TMR0	TIMER0 reset control	0: TIMER0 unit 1.2 Function Reset Valid 1: TIMER0 Unit 1.2 Function Reset Disable	R/W
b11	Reserved	-	Read at "1", write at "1"	R/W
b10	TMR4	TIMER4 reset control	0: TIMER4 Function Reset Valid 1: TIMER4 Function Reset Disable	R/W
b9-b1	Reserved	-	Read at "11111111", write at "11111111"	R/W
b0	TMR6	TIMER6 reset control	0: TIMER6 unit 1 to unit 10 function reset is valid 1 : TIMER6 unit 1 to unit 10 function reset release	R/W

3.5.4 Function Reset Control 3 (RMU_FRST3)

Reset value: 0xFFFFFFFF

b31	b30	b29	b28	b27	b26	b25	b24
-	-					USART6	USART5
b23	b22	b21	b20	b19	b18	b17	b16
USART4	USART3	USART2	USART1	-			SMC
b15	b14	b13	b12	b11	b10	b9	b8
	-	-	OTS	-	-	-	CMP12
b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	DAC1	-	ADC3	ADC2	ADC1

Bit	Flag	Bit Name	Function	Read and write
b31-b26	Reserved	-	Read at “111111”, write at “111111”	R/W
b25	USART6	USART6 reset control	0: General Synchronous Asynchronous Receiver sender USART Unit 6 Function Reset Valid 1: General Synchronous Asynchronous Receiver USART Unit 6 Function Reset Disable	R/W
b24	USART5	USART5 reset control	0: General Synchronous Asynchronous Receiver sender USART Unit 5 Function Reset Valid 1: General Synchronous Asynchronous Receiver USART Unit 5 Function Reset Disable	R/W
b23	USART4	USART4 reset control	0: General Synchronous Asynchronous Receiver sender USART Unit 4 Function Reset Valid 1: General Synchronous Asynchronous Receiver USART Unit 4 Function Reset Disable	R/W
b22	USART3	USART3 reset control	0: General Synchronous Asynchronous Receiver sender USART Unit 3 Function Reset Valid 1: General Synchronous Asynchronous Receiver USART Unit 3 Function Reset Disable	R/W
b21	USART2	USART2 reset control	0: General Synchronous Asynchronous Receiver sender USART Unit 2 Function Reset Valid 1: General Synchronous Asynchronous Receiver USART Unit 2 Function Reset Disable	R/W
b20	USART1	USART1 reset control	0: General Synchronous Asynchronous Receiver sender USART Unit 1 Function Reset Valid 1: General Synchronous Asynchronous Receiver USART Unit 1 Function Reset Disable	R/W
b19-b17	Reserved	-	Read at “111”, write at “111”	R/W
b16	SMC	SMC reset control	0: The SMC controller function of the external storage controller (EXMC) is reset and enabled 1 : The SMC controller function of the external storage controller (EXMC) is reset and released	R/W
b15-b13	Reserved	-	Read at “111”, write at “111”	R/W
b12	OTS	OTS reset control	0: Temperature sensor OTS function reset is valid 1 : Temperature sensor OTS function reset release	-
b11-b9	Reserved	-	Read at “111”, write at “111”	R/W
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 Disable	R/W
b7-b5	Reserved	-	Read at “111”, write at “111”	R/W
b4	DAC1	DAC1 reset control	0: DAC unit 1 function reset is valid 1 : DAC unit 1 function reset released	R/W
b3	Reserved	-	Read at “1”, write at “1”	R/W
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 Disable	R/W

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	R/W
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	R/W

3.5.5 Reset Control Register (RMU_PRSTCR0)

Reset Value: 0x40

b7	b6	b5	b4	b3	b2	b1	b0
-	-	LKUPREN	-	-	-	-	-

Bit	Flag	Bit Name	Function	Read and write
b7	Reserved	-	Read at "0", write at "0"	R/W
b6	Reserved	-	Read at "1", write at "1"	R/W
b5	LKUPREN	LOCKUP reset enable	0: LOCKUP reset is invalid 1: LOCKUP reset enabled	R/W
b4- b0	Reserved	-	Read at "0", write at "0"	R/W

3.5.6 Reset Flag Register0 (RMU _ RSTF0)

Reset value: 0xXXXXh (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	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	LKUPRf	XTALERf	CKFERf	RAECRf	RAPERf	MPUERf	SWRf	PDRf	SWDRf	WDRf	PVD2Rf	PVD1Rf	BORf	PINRf	PORf

Bit	Flag	Bit Name	Function	Read and write
b31	CLRf	Clear reset Mark	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 flag	R/W
b30	MULTIRf	More than 2 reduction Occurrence flag bit	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	R/W
b29-b15	Reserved	-	Read at "0", write at "0"	R/W
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	R/W
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	R/W
b12	CKFERf	abnormal frequency Reset flag	clock 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	R/W
b11	RAECRf	RAMECC reset Mark	Set by hardware when a RAMECC reset occurs. Reset by setting CLRf. 0: No RAMECC reset occurred 1: RAMECC reset occurs	R/W
b10	RAPERf	RAM parity 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	R/W
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	R/W
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	R/W
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	R/W
b6	SWDRf	dedicated 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	R/W
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	R/W
b4	PVD2Rf	Programmable detection 2 Reset flag	voltage 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	R/W
b3	PVD1Rf	Programmable detection 1 Reset flag	voltage Set by hardware when a programmable voltage sense 1 reset occurs. Reset by setting. 0: Programmable voltage detection 1 reset does not occur 1: Programmable voltage detection 1 reset occurs	R/W

b2	BORF	Undervoltage 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	R/W
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	R/W
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	R/W

4 Clock Controller (CMU)

4.1 Introduction

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 system clock source can select 6 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)

The maximum operating clock frequency of the system clock can reach 120 MHz. SWDT has an independent clock source: SWDT dedicated internal low-speed oscillator (SWDTLRC). The real-time clock (RTC) uses an external low-speed oscillator or an internal low-speed oscillator as a clock source. 48MHz clock of USB-FS can select the system clock PLLH as the clock source.

For each timer, 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

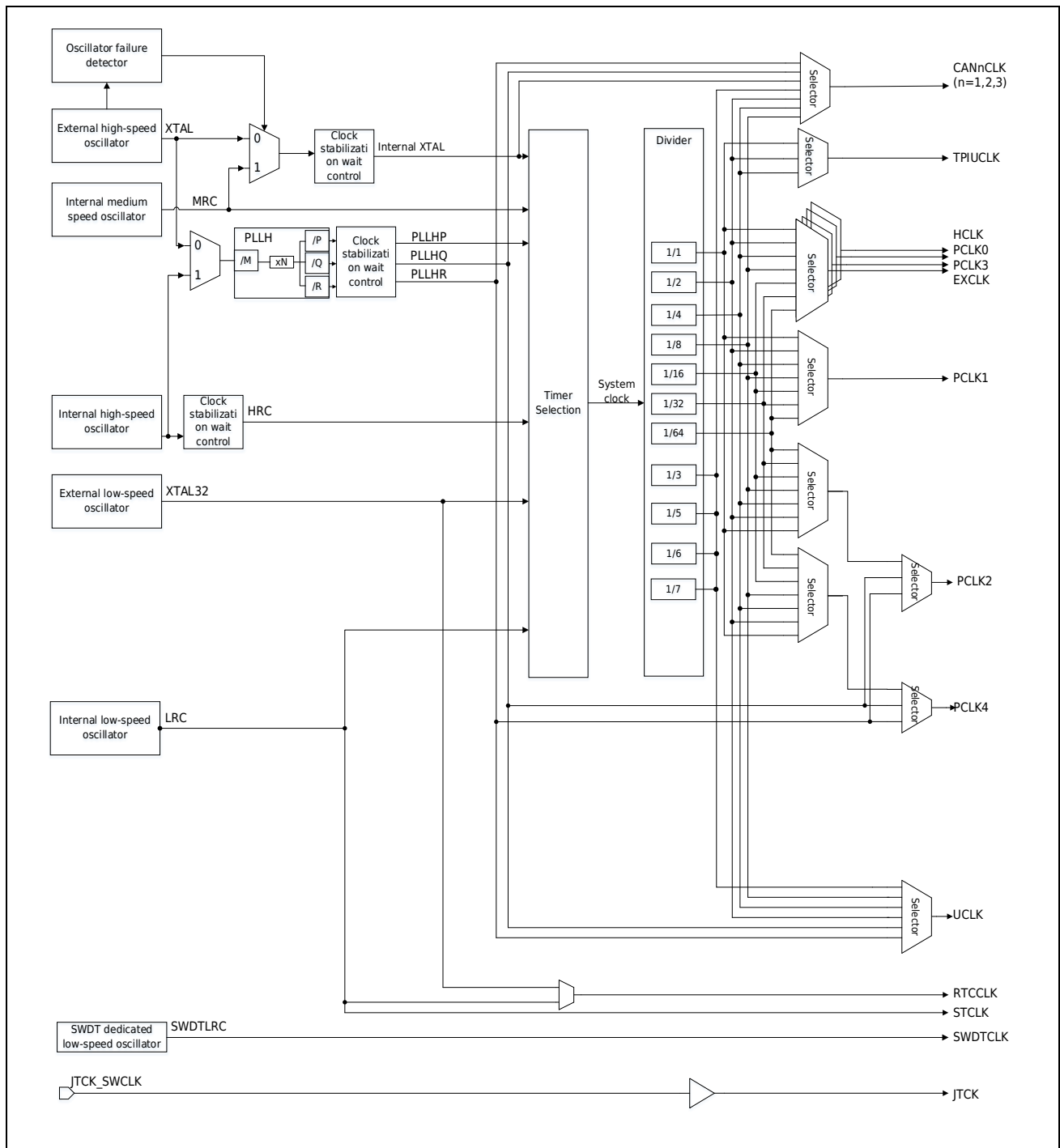


Figure 4-1 Clock System Block Diagram

4.2.2 Clock Frequency Measurement Block Diagram

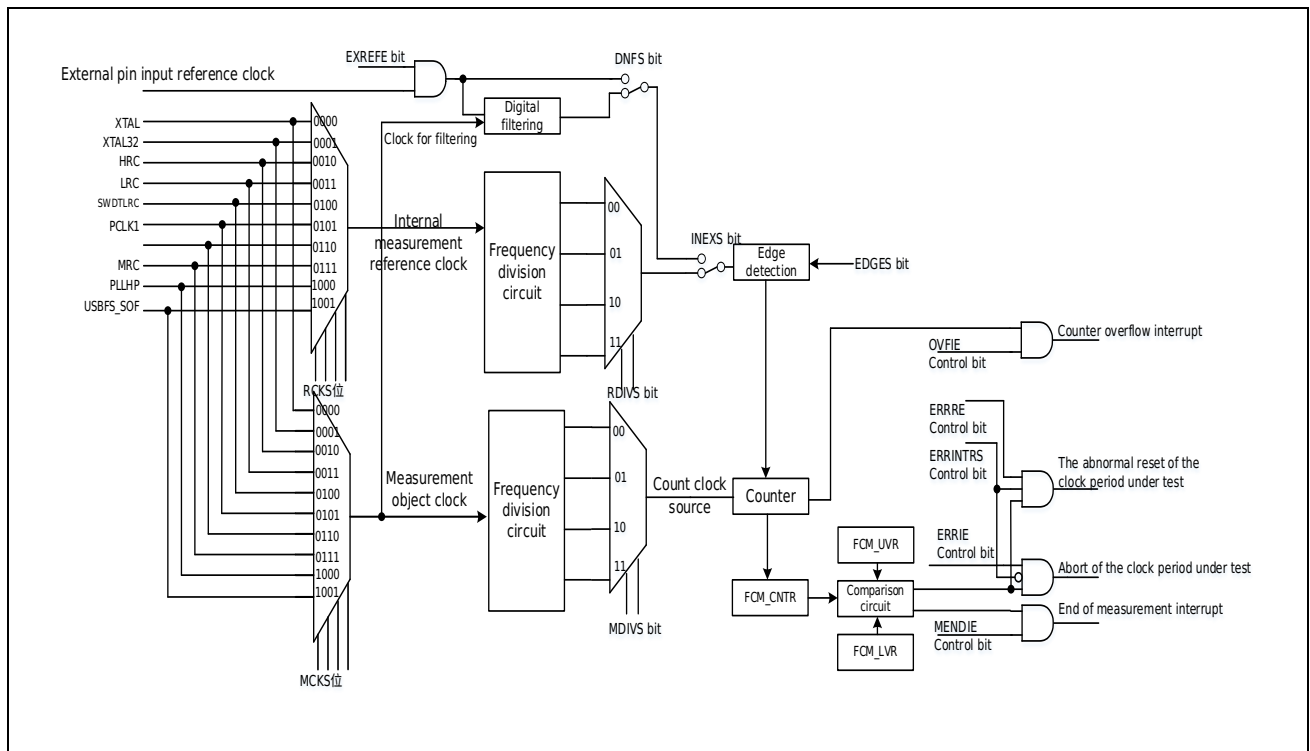


Figure 4-2 Clock Frequency Measurement Block Diagram

4.3 Timer Specification

The main characteristics of each timer 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 divider ratio: 2~16 arbitrary frequency division PLLHP output divider ratio: 2~16 arbitrary frequency division PLLHR output divider ratio: 2~16 arbitrary frequency division PLLHP output frequency = (input clock/PLLH input clock frequency division)*PLLH multiplication factor/PLLHP output frequency division ratio PLLHQ output frequency = (input clock / PLLH input clock frequency division) * PLLH 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, DMan(n=1, 2), EFM, SRAM0, SRAMH, MPU, GPIO, DCUn(n=1~3), INTC, QSPI, FIRn(n=1~4), MAU, KEYSKAN	Maximum frequency 120 MHz. Configured by the HCLKS bit in the CMU_SCFGR register. Selectable system clock division: 1, 2, 4, 8, 16, 32, 64
PCLK0	Timer6, Timer4, TimerAn(n=1~4), I2Cn(n=1~3), CMPn(n=1), CANn(n=1~3) Control logic	Maximum frequency 120 MHz Configured by the PCLK0S bit in the CMU_SCFGR register. Selectable system clock division: 1, 2, 4, 8, 16, 32, 64
PCLK1	USBFS Control Logic, USARTn(n=1~6), SPIn(n=1~4), Timer0n(n=1, 2), Timer2n(n=1~4), TimerAn(n=5~6), EMB, CRC, HASH, AES,	Maximum frequency 60 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	Maximum frequency 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, OTS, FCM, CTC, PLA, control logic	Maximum frequency 60 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), DACn(n=1) (control logic), TRNG	Maximum frequency 60 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
EXCLK	SMC	Maximum frequency 60 MHz Configured by the EXCKS bit in the CMU_SCFGR register. Selectable system clock division: 1, 2, 4, 8, 16, 32, 64
UCLK	USBFS FullSpeed Communication clock	Frequency 48 MHz Configured by the USBCKS bit in the CMU_USBCKCFGR register. The system clock division can be selected as 2, 3, 4, 5, 6, 7, or 8. Optional independent clock source: PLLHQ, PLLHR
CANnCLK	CAN1/CAN2/CAN3 Communication clock	Maximum frequency 80 MHz CAN3CKS/CAN2CKS/PLLR, XTALCMU_USB clock configuration registers by CMU_CANCKCFGR register CAN1CKS bit configuration. The system clock division can be selected as 2, 3, 4, 5, 6, 7, or 8. Optional independent clock source:
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	Maximum frequency 25 MHz
TPIUCLK	Clock for Cortex-M4 debug tracer	Maximum frequency 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 $\geq$ PCLK0, PCLK1, PCLK2, PCLK3, PCLK4 frequency

- PCLK0 frequency $\geq$ PCLK1 frequency, PCLK0 frequency $\geq$ PCLK3 frequency
- PCLK2 frequency: PCK4 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.

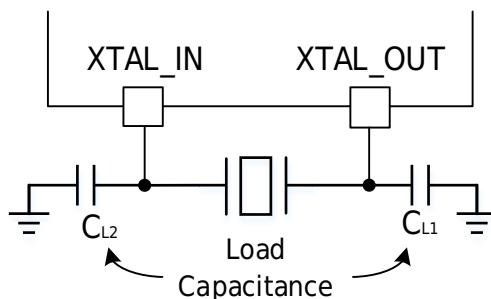


Figure 4-3 External High-speed Oscillator Connection Instance

4.5.1.2 Clock Input Mode

External timer 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 period 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 following figure.

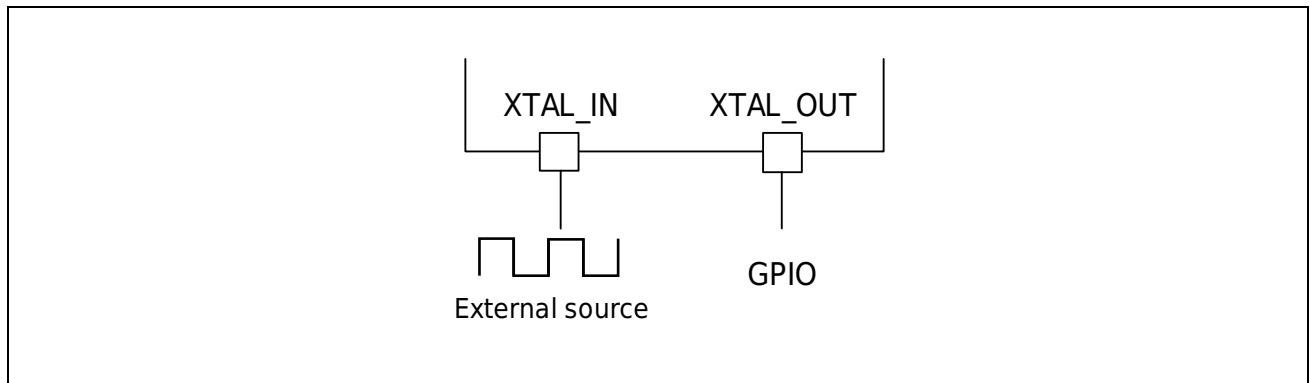


Figure 4-4 Instance connection diagram of external clock input

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 register CMU_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-off mode by software.

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

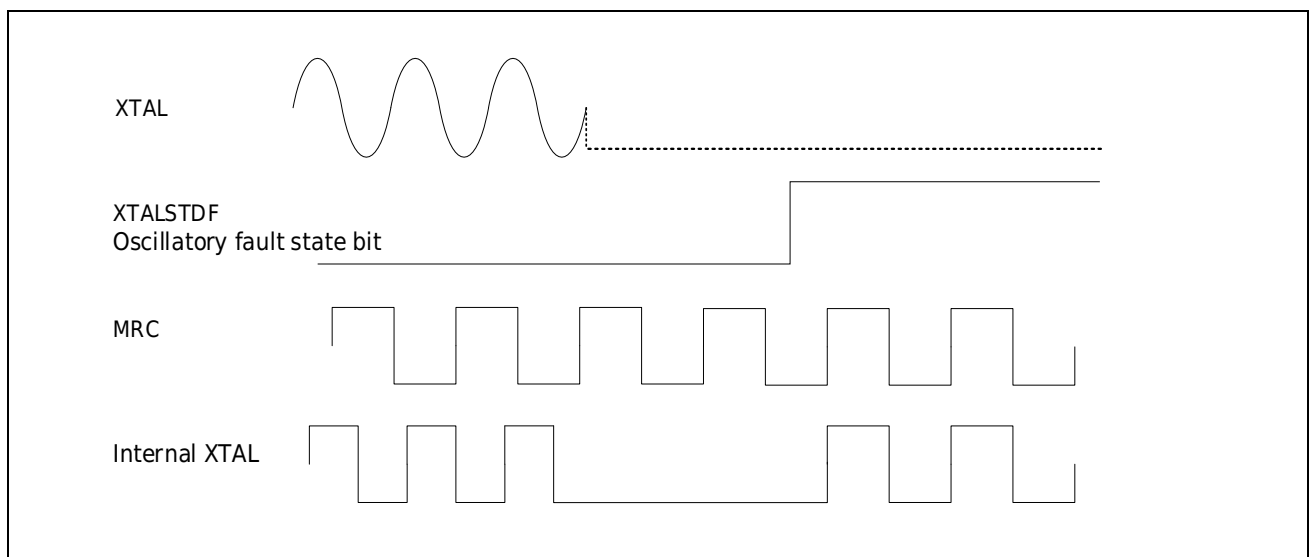


Figure 4-5 Instance of External High-Speed Oscillator Fault Detection

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 Hiz 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 following figure.

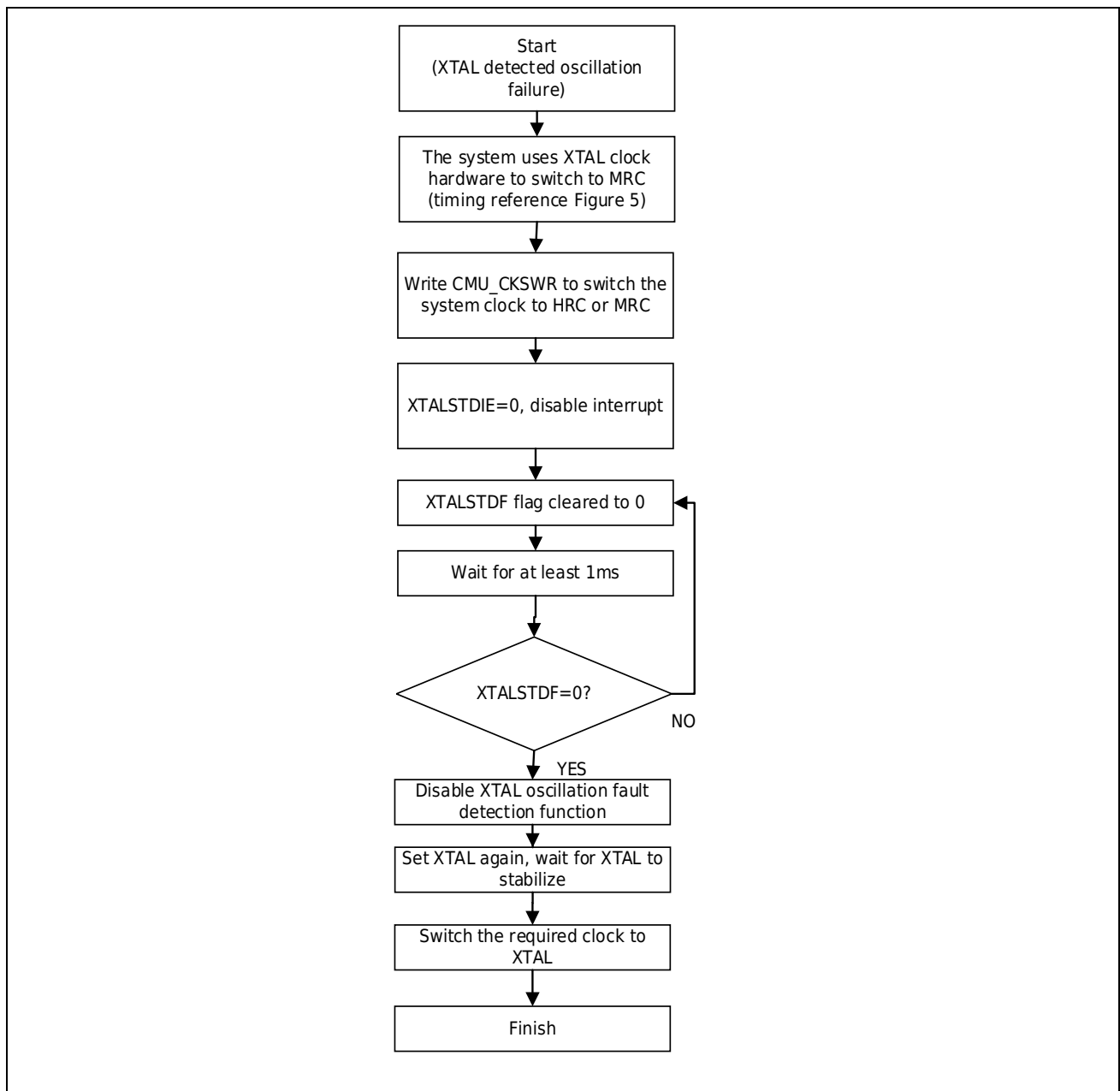


Figure 4-6 XTAL is Selected as the System Clock, and XTAL Oscillation Fault is Detected

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.

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

$$\text{Frequency division number} = \frac{10000000\text{Hz}}{32768\text{Hz}} = \frac{78125 \text{ (Decimal)}}{256 \text{ (Decimal)}} = \frac{1312\text{D (hexadecimal)}}{100 \text{ (hexadecimal)}}$$

The numerator [16:0] register is set to 0x1312D. The denominator [10:0] register is set to 0x100. That is, the register CMU_XTALDIVR is set to 0x1312D100. Then start by setting CMU_XTALDIVCR.

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

$$\text{Frequency division number} = \frac{8000000\text{Hz}}{32768\text{Hz}} = \frac{31250 \text{ (Decimal)}}{128 \text{ (Decimal)}} = \frac{7\text{A12 (hexadecimal)}}{80 \text{ (hexadecimal)}}$$

The numerator [16:0] register is set to 0x7A12. The denominator [10:0] register is set to 0x80. That is, the register CMU_XTALDIVR is set to 0x07A12080. Then start by setting CMU_XTALDIVCR.

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.

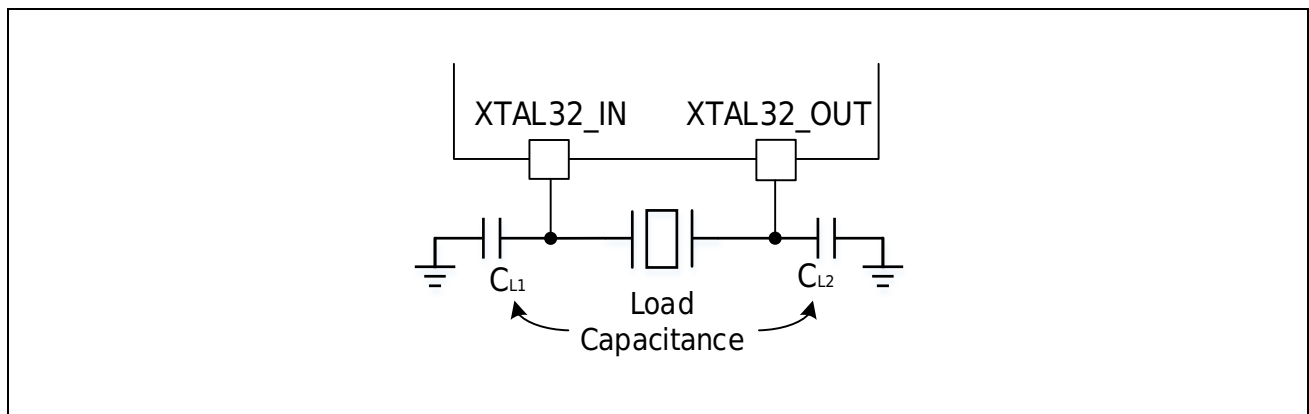


Figure 4-7 External Low-speed Oscillator Connection Instance

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 the XTAL32 to stabilize, and the stabilization time refers to the electrical characteristics chapter.

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 an 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 control HRCSTP bits in register to open or close through CMU _ HRCCR.

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 control MRCSTP bits in register to open or close through CMU _ MRCCR.

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/KEYSCAN.

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 Manual**.

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.

The LRC can control the LRCSTP bits in register to open or close through CMU _ LRCCR.

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 HC32A472 device 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 120 MHz)
 - All three outputs can be used to generate USB , ADC , and CAN 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 and stop modes.

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 Switching. You can switch from one timer to another only if the target timer is stable.

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

4.8.1 Clock Source Switching

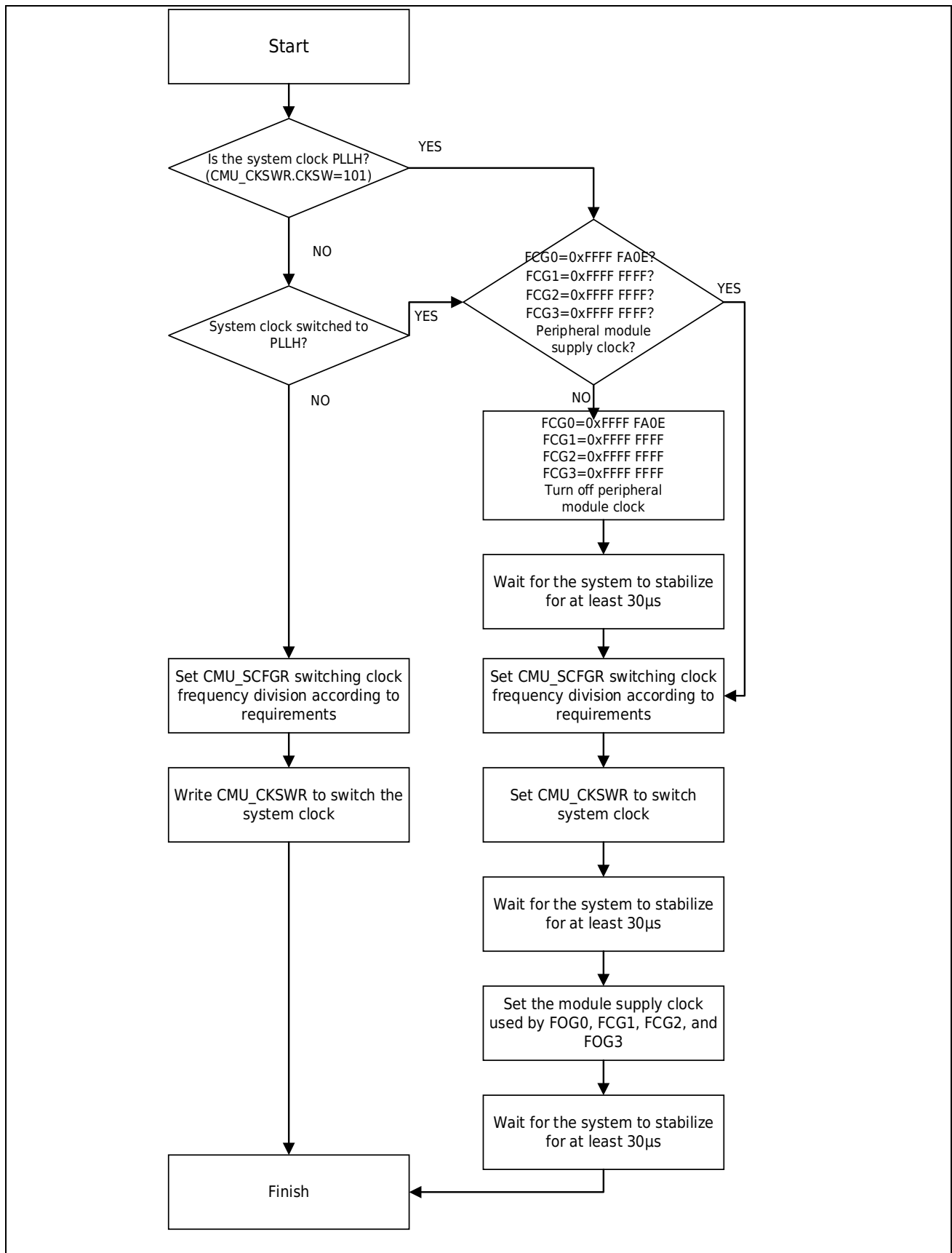


Figure 4-8 Clock Source Switching

4.8.2 Clock Division Switching

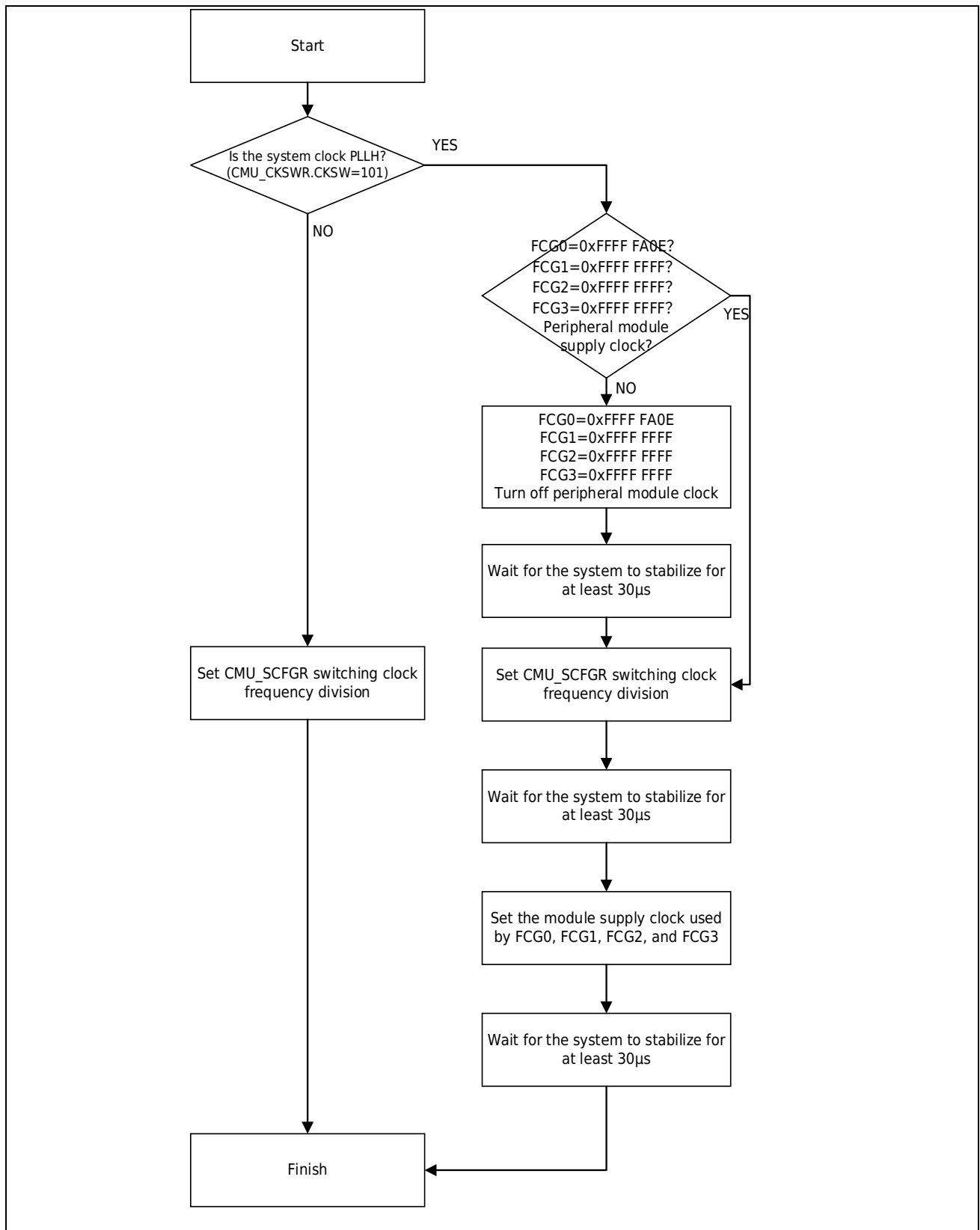


Figure 4-9 Clock Divider Switching

4.9 Clock Output Function

There are two clock outputs:

■ MCO_1

Users can output different timers to the MCO_1 pin through a configurable pre-distributor (from 1 to 128):

- 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

Users can output different timers to the MCO_2 pin through a configurable pre-distributor (from 1 to 128):

- 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).

4.10 Clock Frequency Measurement

4.10.1 Clock Frequency Measurement

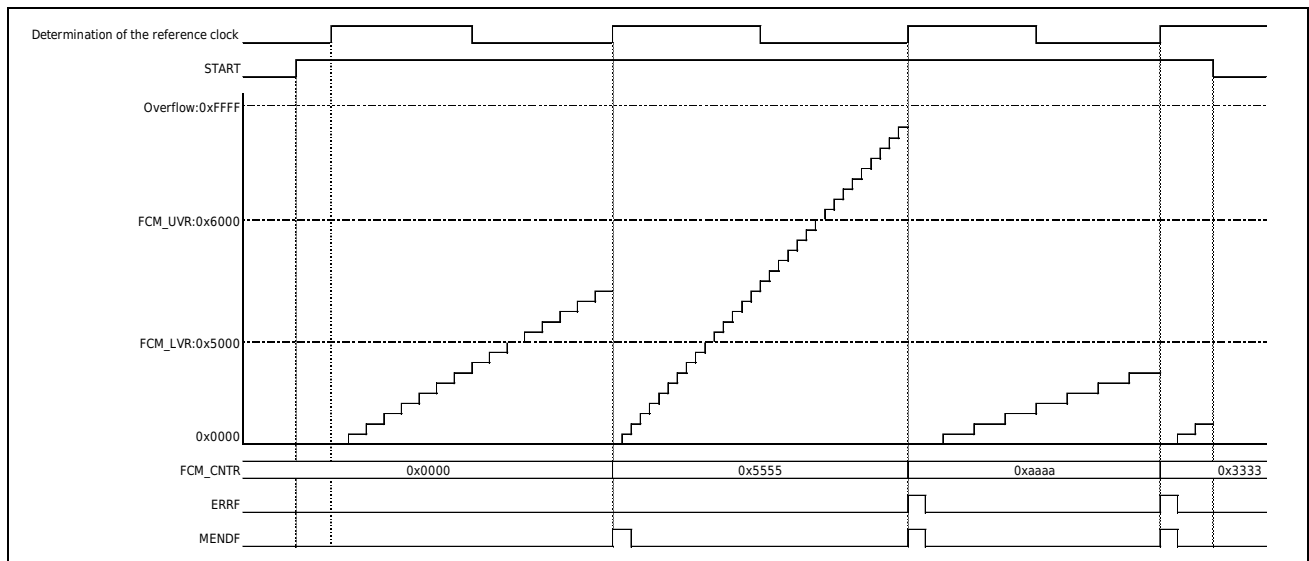


Figure 4-10 Clock Frequency Measurement Timing Diagram

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 $FCM\_LVR \leq FCM\_CNTR \leq FCM\_UVR$, 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:

- 1) Frequency abnormal reset

4.11 Register Description

Table 4-3 Register List

Base address 1: 0x40048400

Register Name	Symbol	Offset Address	Bit Width	Reset Value
FCM lower limit comparison value register	FCM_LVR	0x00	32	0x00000000
FCM upper limit comparison value register	FCM_UVR	0x04	32	0x00000000
FCM counter value register	FCM_CNTR	0x08	32	0x00000000
FCM start-stop register	FCM_STR	0x0C	32	0x00000000
FCM measurement object control register	FCM_MCCR	0x10	32	0x00000000
FCM measurement reference control register	FCM_RCCR	0x14	32	0x00000000
FCM interrupt reset control register	FCM_RIER	0x18	32	0x00000000
FCM flags register	FCM_SR	0x1C	32	0x00000000
FCM flag bit clear register	FCM_CLR	0x20	32	0x00000000

Base address 2: 0x4004CC00

CMU_XTAL configuration register	CMU_XTALCFGR	0x50	8	0x80
CMU_XTAL32 Control Register	CMU_XTAL32CR	0x54	8	0x00
CMU_XTAL32 configuration register	CMU_XTAL32CFGR	0x58	8	0x00
CMU_XTAL32 filter register	CMU_XTAL32NFR	0x68	8	0x00
CMU_LRC control register	CMU_LRCCR	0x6C	8	0x00
CMU_LRC Calibration Register	CMU_LRCTRM	0x74	8	0x00

Base address 3: 0x40048000

CMU_XTAL Fractional Frequency Configuration Register	CMU_XTALDIVR	0x18	32	0x03D09040
CMU_XTAL Fractional Frequency Control Register	CMU_XTALDIVCR	0x1C	32	0x00000000

Base address 4: 0x40054000

Register Name	Symbol	Offset Address	Bit Width	Reset Value
CMU_XTAL Stable Configuration Register	CMU_XTALSTBCR	0x0A2	8	0x05
CMU_XTAL control register	CMU_XTALCR	0x032	8	0x01
CMU_XTAL Oscillatory Fault Control Register	CMU_XTALSTDCR	0x040	8	0x00
CMU_XTAL Oscillatory Fault Control Register	CMU_XTALSTDSR	0x041	8	0x00
CMU_HRC control register	CMU_HRCCR	0x036	8	Determined by ICG1.HRCSTP value
CMU_HRC Calibration Register	CMU_HRCTRM	0x062	8	0x00
CMU_MRC Control Register	CMU_MRCCR	0x038	8	0x80
CMU_MRC Calibration Register	CMU_MRCTRM	0x061	8	0x00
CMU_PLLH Configuration Register	CMU_PLLHCFGR	0x100	32	0x11101300
CMU_PLLH Control Register	CMU_PLLHCR	0x02A	8	0x01
CMU_Clock Source Stability Status Register	CMU_OSCSTBSR	0x03C	8	0x00
CMU_System timer switchover register	CMU_CKSWR	0x026	8	0x01
CMU_Clock Divider Configuration Register	CMU_SCFGR	0x020	32	0x00000000
	CMU_USBCKCFGR	0x024	8	0x40
CMU_CAN clock configuration register	CMU_CANCKCFGR	0x018	16	0x0ddd
CMU_AD/TRNG/DA clock configuration register	CMU_PERICKSEL	0x010	16	0x0000
CMU_Debug clock configuration register	CMU_TPIUCKCFGR	0x03F	8	0x00
CMU_MCO1 configuration register	CMU_MCO1CFGR	0x03D	8	0x00
CMU_MCO2 configuration register	CMU_MCO2CFGR	0x03E	8	0x00

4.11.1 CMU XTAL Configuration Register (CMU _ XTALCFGR)

Reset value: 0x80

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

Bit	Flag	Bit Name	Function	Read and write
b7	Reserved	-	Write "1" on write	W
b6	Reserved	-	Write "0" on write	R/W
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: Small drive capability (recommended 8-16 MHz crystal oscillator) 11: Small drive capability (recommended 4-8 MHz crystal oscillator)	R/W
b3	XTALMS	XTAL mode selection bit	0: Oscillator mode 1: External clock input mode	R/W
b2~b0	Reserved	-	Read as "0", write as "0"	R/W

4.11.2 CMU XTAL Configuration Register (CMU _ XTALSTBCR)

Reset Value: 0x05

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

Bit	Flag	Bit Name	Function	Read and write
b7~b4	Reserved	-	Read as "0", write as "0"	R/W
b3~b0	XTALSTB [3:0]	XTAL Stability Time Selection	0001: Stable counter 35 periods 0010: Stable counter 67 periods 0011: Stable counter 131 periods 0100: Stable counter 259 periods 0101: Stable counter 547 periods 0110: Stable counter 1059 periods 0111: Stable counter 2147 periods 1000: Stable counter 4291 periods 1001: Stable counter 8163 periods 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.	R/W

4.11.3 CMU XTAL Fractional Divider Configuration Register (CMU_XTALDIVR)

Reset value: 0x03d09040

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

Bit	Flag	Bit Name	Function	Read and write
b31~b29	Reserved	-	Read as "0", write as "0"	R/W
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]	R/W
b11	Reserved	-	Read as "0", write as "0"	R/W
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.	R/W

4.11.4 CMU XTAL Fractional Divider Control Register (CMU_XTALDIVCR)

Reset value: 0x00000000

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

Bit	Flag	Bit Name	Function	Read and write
b31~b1	Reserved	-	Read as "0", write as "0"	R/W
b0	FRADIVEN	XTAL decimal division starts	0: XTAL fractional frequency stop 1: XTAL fractional division starts	W

4.11.5 CMU XTAL Control Register (CMU_XTALCR)

Reset value: 0x01

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	XTALSTP

Bit	Flag	Bit Name	Function	Read and write
b7~b1	Reserved	-	Read as "0", write as "0"	R/W
b0	XTALSTP	XTAL oscillator open stop bit	0: XTAL oscillator starts 1: XTAL oscillator stops	R/W

Note:

- When XTAL is selected as system clock or PLLH clock source, XTALSTP is prohibited to write 1 to stop XTAL oscillator.
- 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.
- 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	-	-	-	-	XTALSTDRIS	XTALSTDRE	XTALSTDIE

Bit	Flag	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-off mode.	R/W
b6~b3	Reserved	-	Read as "0", write as "0"	R/W
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.	R/W
b1	XTALSTDRE	XTAL Oscillatory Fault Reset Allowance	0: Disable XTAL Oscillatory Fault Reset 1: Allow XTAL Oscillation Fault Reset	R/W
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 Hiz output through EMB, and the XTALSTDIE bit needs to be set to 1.	R/W

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
-	-	-	-	-	-	-	XTALSTDF

Bit	Flag	Bit Name	Function	Read and write
b7~b1	Reserved	-	Read as "0", write as "0"	R/W
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.	R/W

4.11.8 CMU XTAL32 Configuration Register (CMU_XTAL32CFGR)

Reset value: 0xXX

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

Bit	Flag	Bit Name	Function	Read and write
b7~b3	Reserved	-	Read as "0", write as "0"	R/W
b2~b0	XTAL32DRV [2:0]	XTAL32 drive capability selection	000: Medium drive capability 001: Large drive capacity Other: prohibitions Note: For the method of use, refer to the chapter of electrical characteristics [low-speed external clock generated by crystal oscillator/ceramic resonator]	R/W

4.11.9 CMU XTAL32 Filter Register (CMU_XTAL32NFR)

Reset value: 0xXX

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

Bit	Flag	Bit Name	Function	Read and write
b7~b2	Reserved	-	Read as "0", write as "0"	R/W
b1~b0	XTAL32NF [1:0]	XTAL32 Oscillator Filter Selection	00: RUN mode/stop mode/power down mode, 3us filter of XTAL32 is valid 01: The 3us filter of XTAL32 is valid in RUN mode, and the 3us filter of XTAL32 in stop mode or power-down mode is invalid 10: Prohibit setting 11: RUN mode/stop mode/power down mode, the 3us filter of XTAL32 is invalid	R/W

4.11.10 CMU XTAL32 Control Register (CMU_XTAL32CR)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	XTAL32STP

Bit	Flag	Bit Name	Function	Read and write
b7~b1	Reserved	-	Read as "0", write as "0"	R/W
b0	XTAL32STP	XTAL32 oscillator start stop bit	0: XTAL32 oscillator oscillates 1: XTAL32 oscillator stopped	R/W

Note:

- When XTAL32 is selected as the system clock source, XTAL32STP is prohibited from writing "1" to stop the XTAL32 oscillator.
- The software sets the XTAL32 action to start, and waits for 5 XTAL32 periods before stopping the XTAL32 again.
- The software sets the XTAL32 to stop, and waits for 5 XTAL32 periods 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
HRCTRM [7:0]							

Bit	Flag	Bit Name	Function	Read and write
b7~b0	HRCTRM [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	R/W

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
-	-	-	-	-	-	-	HRCSTP

Bit	Flag	Bit Name	Function	Read and write
b31~b1	Reserved	-	Read as "0", write as "0"	R/W
b0	HRCSTP	HRCoscillator open stop bit	0: HRCoscillator Oscillation 1: HRCoscillator stops According to ICG1.HRCSTOP configuration, HRC starts to stop after reset.	R/W

Note:

- 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.
- 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.
- The software sets the HRC to stop, and the PLLH 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
MRCTRM [7:0]							

Bit	Flag	Bit Name	Function	Read and write
b7~b0	MRCTRM [7:0]	MRC Frequency Calibration Bits	10000000: -128	R/W
			10000001: -127	
			-	
			11111111: -1	
			00000000: Center Code	
			00000001: +1	
			-	
			01111110: +126	
			01111111: +127	

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
-	-	-	-	-	-	-	MRCSTP

Bit	Flag	Bit Name	Function	Read and write
b7	Reserved	-	Read as "1", write as "1"	R/W
b6~b1	Reserved	-	Read as "0", write as "0"	R/W
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.	R/W

Note:

- When MRC is selected as system timer, MRCSTP writes "1" to stop MRC clock.
- The software sets the MRC oscillation and waits for 5 MRC periods before entering stop mode, power-down mode or stopping MRC.
- The software sets the MRC to stop, and waits for 5 MRC periods before entering the stop mode, power-down mode or restarting the MRC.
- MRC is used as RTC calibration clock. Possibility of MRC oscillation when RTC is not initialized. When the RTC calibration function is enabled, the MRCSTP bit setting is invalid and the MRC may oscillate.

4.11.15 CMU LRC Calibration Register (CMU_LRCTRM)

Reset value: 0x00

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

Bit	Flag	Bit Name	Function	Read and write
b7~b0	LRCTRM [7:0]	LRC Frequency Calibration Bits	10000000: -128 10000001: -127 - 11111111: -1 00000000: Center Code 00000001: +1 - 01111110: +126 01111111: +127	R/W

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
-	-	-	-	-	-	-	LRCSTP

Bit	Flag	Bit Name	Function	Read and write
b31~b1	Reserved	-	Read as "0", write as "0"	R/W
b0	LRCSTP	LRCoscillator open stop bit	0: LRCoscillator Oscillation 1: LRCoscillator stops	R/W

Note:

- When LRC is selected as system timer, LRCSTP writes "1" to stop LRC clock.
- The software sets the start of the LRC action, and waits for 5 LRC periods before entering the stop mode, power-down mode or stopping the LRC.
- The software sets the LRC to stop, and waits for 5 LRC periods before entering the stop mode, power-down mode or restarting the LRC.
- 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: 0x11101300

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]				-	-	-	PLLH N [8]
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PLLHN [7:0]								PLLS RC	-	-	PLLHM [1:0]				

Bit	Flag	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	R/W
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	R/W
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	R/W
b19-b17	-	-	Read as "0", write as "0"	R/W
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	b16-b8
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	R/W
b6-b5	-	-	Read as "0", write as "0"	R/W
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.	R/W

00000: 1 frequency division
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
-	-	-	-	-	-	-	PLLHOFF

Bit	Flag	Bit Name	Function	Read and write
b7~b1	Reserved	-	Read as "0", write as "0"	R/W
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	R/W

Note:

- When PLLH is selected as the system clock source, it is forbidden to write 1 to PLLHOFF to stop the PLLH clock.
- 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.
- 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.
- 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 Clock Source Stabilizer (CMU _ OSCSTBSR)

Reset value: 0x00

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

Bit	Flag	Bit Name	Function	Read and write
b7-b6	Reserved	-	Read as "0", write as "0"	R
b5	PLLHSTBF	PLLH stability sign bit	0: PLLH stopped or unstabilized 1: PLLH stabilization	R
b4	Reserved	-	Read as "0", write as "0"	R
b3	XTALSTBF	XTAL stability sign bit	0: XTAL stopped or unstabilized 1: XTAL Stabilization	R
b2~b1	Reserved	-	Read as "0", write as "0"	R
b0	HRCSTBF	HRC stable sign bit	0: HRC stopped or unstabilized 1: HRC stability	R

4.11.20 CMU System Clock Source Switching Register (CMU_CKSWR)

Reset value: 0x01

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

Bit	Flag	Bit Name	Function	Read and write
b31~b3	Reserved	-	Read as "0", write as "0"	R/W
b2-b0	CKSW [2:0]	System timer switchover	0 0 0: Select HRC clock as system clock 0 0 1: Select MRC clock as system clock 0 1 0: Select LRC clock as system clock 0 1 1: Select XTAL Clock as System Clock 1 0 0: Select XTAL32 clock as system clock 1 0 1: Select PLLH as the system clock 1 1 0: Disable setting 1 1 1: Disable setting Note: 1. Switch the target timer to ensure the clock is stable. 2. Refer to the [Clock Source Switching] chapter for the process 3. When thePWC_STPMCR.CKSMRCbit is 1, after the stop mode wakes up, this register initializes and the system timer selects the MRC clock.	R/W

4.11.21 CMU Clock Divider Configuration Register (CMU_SCFGR)

Reset value: 0x00000000

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

Bit	Flag	Bit Name	Function	Read and write
b31~b27	Reserved	-	Read as "0", write as "0"	R/W
b26~24	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 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: Disable setting Note: When the PWC_STPMCR.CKSMRC bit is 1, this register is initialized after wake-up from stop mode, and HCLK is divided by 1 of the system clock.	R/W
b23	Reserved	-	Read as "0", write as "0"	R/W
b22~20	EXCKS [2:0]	ExMC clock frequency division selection bit	000: 1 frequency division of system clock 001: 2-division frequency of system clock 010: 4-division frequency of 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: Disable setting Note: When the PWC_STPMCR.CKSMRC bit is 1, this register is initialized after wake-up from stop mode, and EXCLK is divided by 1 of the system clock.	R/W
b19	Reserved	-	Read as "0", write as "0"	R/W
b18~16	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 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: Disable setting 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.	R/W
b15	Reserved	-	Read as "0", write as "0"	R/W
b14~12	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 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: Disable setting 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.	R/W
b11	Reserved	-	Read as "0", write as "0"	R/W
b10~8	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 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: Disable setting	R/W

			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.	
b7	Reserved	-	Read as "0", write as "0"	R/W
b6~4	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 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: Disable setting 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.	R/W
b3	Reserved	-	Read as "0", write as "0"	R/W
b2~0	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 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: Disable setting 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.	R/W

4.11.22 CMU USB Clock Configuration Register (CMU_USBCKCFGR)

Reset Value: 0x40

b7	b6	b5	b4	b3	b2	b1	b0
USBCKS [3:0]				-	-	-	-

Bit	Flag	Bit Name	Function	Read and write
b7~b4	USBCKS [3:0]	48MHz clock source selection for USB-FS	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 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 PWC_STPMCR.CKSMRC bit is 1, after wake-up from stop mode, this register is initialized, and USBCLK is divided by 5 of the system clock.	R/W
b3~b0	Reserved	-	Read as "0", write as "0"	R/W

4.11.23 CMU CAN Clock Configuration Register (CMU_CANCKCFGR)

Reset value: 0x0ddd

b15	b14	b13	b12	b11	b10	b9	b8
-				CAN3CKS [3:0]			
b7	b6	b5	b4	b3	b2	b1	b0
CAN2CKS [3:0]				CAN1CKS [3:0]			
Bit	Flag	Bit Name	Function				Read and write
b15~b12	-	-	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:				R/W
b11~b8	CAN3CKS [3:0]	CAN3 Communication Clock Selection	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 set USB, CAN, QSPI, SPI, general timer, FCM, ADC, and DAC to the module stop state, and then write the CMU_SCFGR register to switch the clock frequency division. After writing the CMU_CANCKCFGR register, the software waits for the system to stabilize for at least 30μs. 3. the PWC_STPMCR.CKSMRC bit is 1, this register is initialized after waking up from stop mode, and CANCLK is the XTAL clock source.				R/W
b7~b4	CAN2CKS [3:0]	CAN2 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:				R/W
b3~b0	CAN1CKS [3:0]	CAN 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				R/W

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 set USB, CAN, QSPI, SPI, general timer, FCM, ADC, and DAC to the module stop state, and then write the CMU_SCFGR register to switch the clock frequency division. After writing the CMU_CANCKCFGR register, the software waits for the system to stabilize for 30us.
 3. the PWC_STPMCR.CKSMRC bit is 1 , this register is initialized after waking up from stop mode, and CANCLK is the XTAL clock source.
-

4.11.24 CMU AD/DA Clock Configuration Register (CMU_PERICKSEL)

Reset value: 0x0000

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

Bit	Flag	Bit Name	Function	Read and write
b15~b4	Reserved	-	Read as "0", write as "0"	R/W
b3~b0	PERICKSEL [3:0]	AD/DA clock source selection	0000: PCLK2/PCLK4 set by CMU_SCFGR 1000: PCLK2/ PCLK4 configured as PLLHQ 1001: PCLK2/ PCLK4 configured as PLLHR In addition, the setting is prohibited.	R/W

Note:

- When the switching target clock source is PLLH, ensure that the PLLH clock is in a stable state.
- The TRNG clock source is not controlled by this bit. When the TRNG operates, this bit must be set to 0b0000.

4.11.25 CMU Debug Clock Configuration Register (CMU_TPIUCKCFGR)

Reset value: 0x00

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

Bit	Flag	Bit Name	Function	Read and write
b7	TPIUCKOE	TPIU clock supply enable bit	0: Disable 1: Enable	R/W
b6~b2	-	-	Read as "0", write as "0"	R/W
b1~0	TPIUCKS [1:0]	TPIU clock frequency division selection bit	00: 1 frequency division 01: 2 frequency division 10: 4 frequency division Other: prohibitions	R/W

4.11.26 CMU MCO1 Configuration Register (CMU_MCO1CFGR)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
MCO1EN	MCO1DIV [2:0]			MCO1SEL [3:0]			
Bit	Flag	Bit Name	Function				Read and write
b7	MCO1EN	MCO_1 Output License	0: Disable MCO_1 Output 1: Allow MCO_1 Output				R/W
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				R/W
b3~b0	MCO1SEL [3:0]	MCO_1 timer 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.				R/W

4.11.27 CMU MCO2 Configuration Register (CMU_MCO2CFGR)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
MCO2EN	MCO2DIV [2:0]			MCO2SEL [3:0]			
Bit	Flag	Bit Name	Function				Read and write
b7	MCO2EN	MCO_2 export permission	0: disable MCO_2 output 1: Enable MCO_2 output				R/W
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				R/W
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.				R/W

4.11.28 FCM Lower Limit Compare Value Register (FCM_LVR)

Reset value: 0x00000000

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

Bit	Flag	Bit Name	Function	Read and write
b31~b16	Reserved	-	Read as "0", write as "0"	R/W
b15~b0	LVR [15:0]	Lower limit comparison value	A 0 in the START bit configures this register.	R/W

4.11.29 FCM Upper Limit Comparison Value Register (FCM_UVR)

Reset value: 0x00000000

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

Bit	Flag	Bit Name	Function	Read and write
b31~b16	Reserved	-	Read as "0", write as "0"	R/W
b15~b0	UVR [15:0]	Upper limit comparison value	A 0 in the START bit configures this register.	R/W

4.11.30 FCM Counter Value Register (FCM_CNTR)

Reset value: 0x00000000

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

Bit	Flag	Bit Name	Function	Read and write
b31~b16	Reserved	-	Read as "0", write as "0"	R/W
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

4.11.31 FCM Start Stop Register (FCM_STR)

Reset value: 0x00000000

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

Bit	Flag	Bit Name	Function	Read and write
b31~b1	Reserved	-	Read as "0", write as "0"	R/W
b0	START	Frequency measurement start bit	0: Stop frequency measurement 1: Frequency measurement starts	R/W

4.11.32 FCM Measurement Object Control Register (FCM_MCCR)

Reset value: 0x00000000

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

Bit	Flag	Bit Name	Function	Read and write
b31~b8	Reserved	-	Read as "0", write as "0"	R/W
b7~b4	MCKS [3:0]	Measuring object clock selection bit	0000: XTAL 0001: XTAL32 0010: HRC 0011: LRC 0100: SWDTLRC 0101: PCLK1 0111: MRC 1000: PLLHP Other: prohibitions	R/W
b3~b2	Reserved	-	Read as "0", write as "0"	R/W
b1~b0	MDIVS [1:0]	Measuring object prescale selection	00: no frequency division 01: 4 frequency division 10: 8 frequency division 11: 32 frequency division	

4.11.33 FCM Measurement Reference Control Register (FCM_RCCR)

Reset value: 0x00000000

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

Bit	Flag	Bit Name	Function	Read and write
b31~b16	Reserved	-	Read as "0", write as "0"	R/W
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	R/W
b14	Reserved	-	Read as "0", write as "0"	R/W
b13~b12	EDGES [1:0]	Measurement reference edge selection bit	00: Rising edge 01: Falling edge 10: rising and falling edge 11: Setting prohibited	R/W
b11~b10	Reserved	-	Read as "0", write as "0"	R/W
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	R/W
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	R/W
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	R/W
b2	Reserved	-	Read as "0", write as "0"	R/W
b1~b0	RDIVS [1:0]	Measuring basis frequency division selection	00: 32 frequency division 01: 128 frequency division 10: 1024 frequency division 11: 8192 frequency division	

4.11.34 FCM Interrupt Reset Control Register (FCM_RIER)

Reset value: 0x00000000

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

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

4.11.35 FCM Flag Register (FCM_SR)

Reset value: 0x00000000

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

Bit	Flag	Bit Name	Function	Read and write
b31~b8	Reserved	-	Read as "0", write as "0"	R/W
b2	OVF	Counter overflow flag bit	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

4.11.36 FCM Flag Clear Register (FCM_CLR)

Reset value: 0x00000000

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

Bit	Flag	Bit Name	Function	Read and write
b31~b3	Reserved	-	Read as "0", write as "0"	R/W
b2	OVFCLR	Counter overflow flag reset 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 Introduction

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

The operating voltage (VCC) of the chip is 1.8 V to 3.6 V. A voltage regulator (LDO) supplies power to the VDD and VDDR domains, and a VDDR voltage regulator (RLDO) supplies power to the VDDR domain in power-down mode. The chip provides two operating modes: high speed and ultra-low speed, and three low power modes: sleep, stop and power-off through the power consumption control logic (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), programmable voltage detection 2 (PVD2), and reference voltage measurement path. Among them, POR, PDR, and BOR control the chip reset action by detecting the VCC voltage. PVD1 detects the VCC voltage and resets or interrupts the chip according to the register settings. PVD2 detects VCC voltage or external input detection voltage, and generates reset or interrupt according to register selection. The reference voltage measurement path is a function for measuring the reference voltage using ADC.

The VDDR area maintains power through RLDO when the chip enters power-down mode , ensuring that the real-time clock module (RTC) and wake-up timer (WKTm) can continue to operate and maintain the data of the 4KB low-power retention SRAM (Ret_SRAM). 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 domain 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, a real-time clock (RTC), and a 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, and the 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_PWRC0.VVDRSD to further reduce power consumption.

The analog power domain is mainly composed of analog-to-digital converter (ADC), digital-to-analog converter (DAC), comparator (CMP), programmable gain amplifier (PGA), and input and output pins of the analog system, and is powered by the AVCC/AVSS pins. In order to provide high-precision analog performance, the analog area is equipped with an independent power supply. In order to ensure that the ADC and DAC have higher accuracy, the reference voltage of the ADC and DAC uses dedicated pins (VREFH, VREFL).

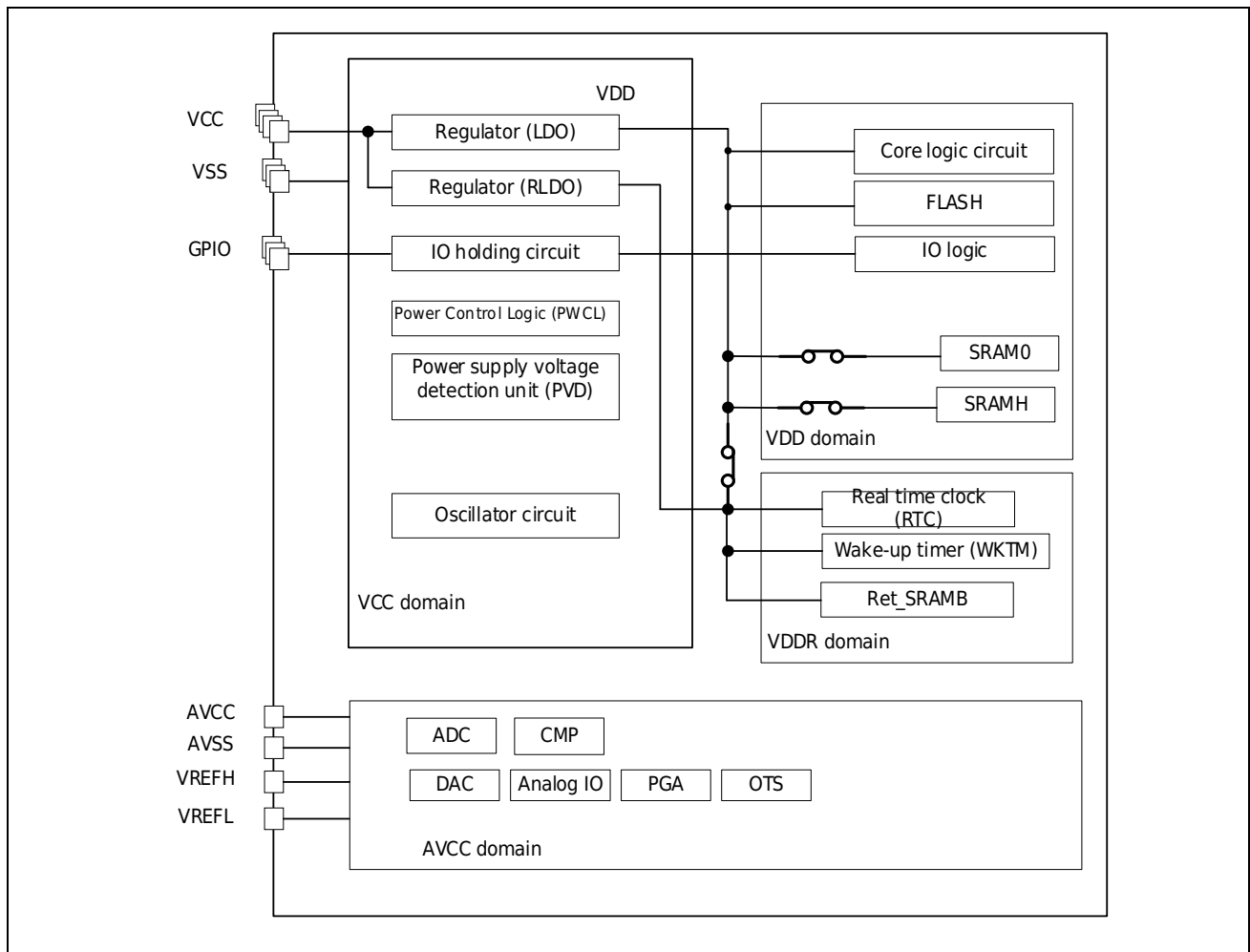


Figure 5-1 Power Supply Diagram

5.3 Power Voltage Detection Unit (PVD) Description

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

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 in 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**.

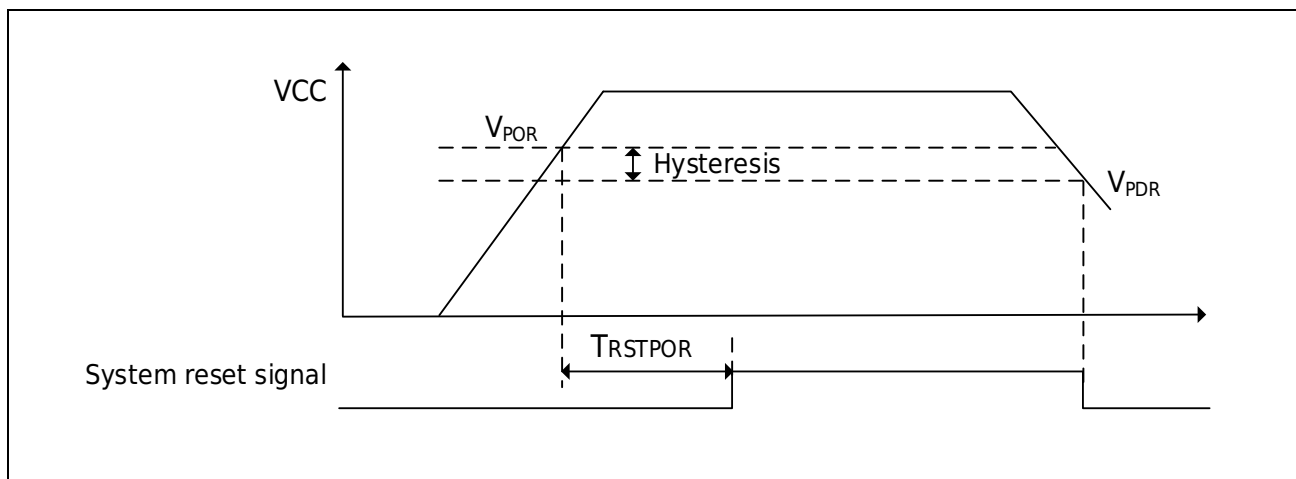


Figure 5-2 Power-on Reset, Power-off Reset Waveforms

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)

the BOR threshold, please refer to the **Electrical Characteristics** section.

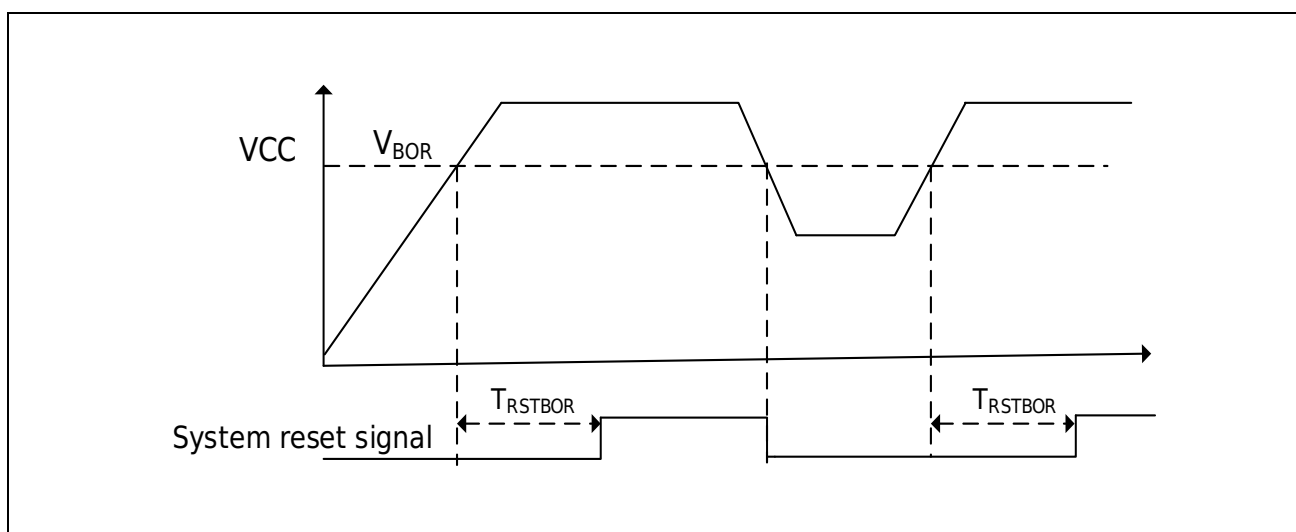


Figure 5-3 Brownout Reset Waveform

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/interrupt (maskable/non-maskable)/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]
Reset	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 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)	Configured as voltage detection 2 interrupt or non-maskable interrupt 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

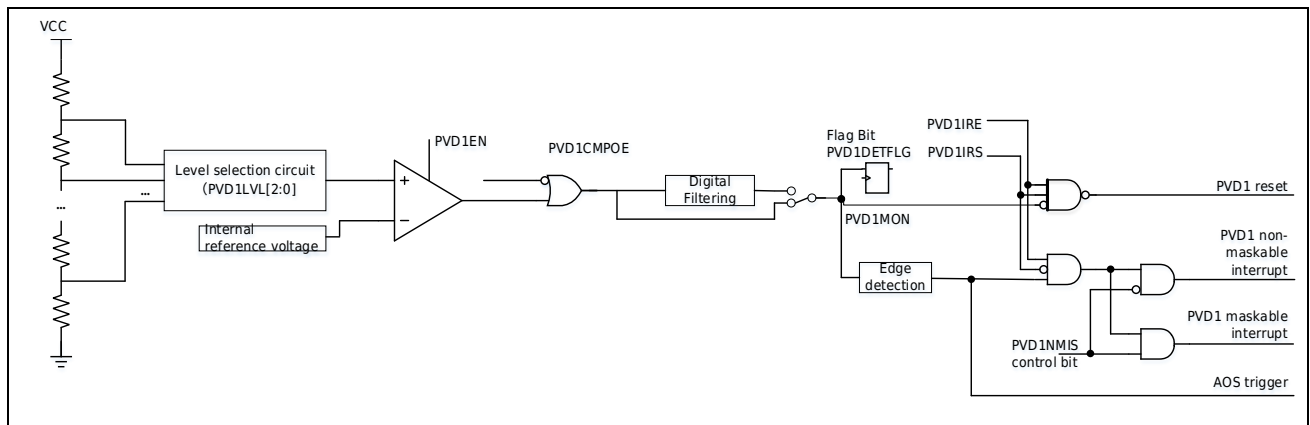


Figure 5-4 PVD1 Interrupt/Reset Block Diagram

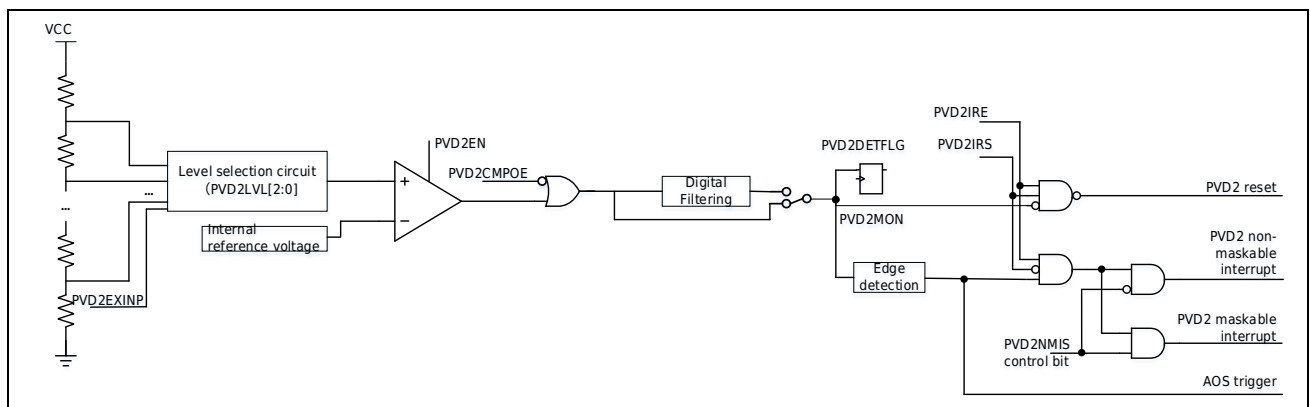


Figure 5-5 PVD2 Interrupt/Reset Block Diagram

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.

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

The above figure is the operating timing diagram of the voltage detection 1 interrupt. PVD1DETFLG needs to be cleared before the interrupt can occur again.

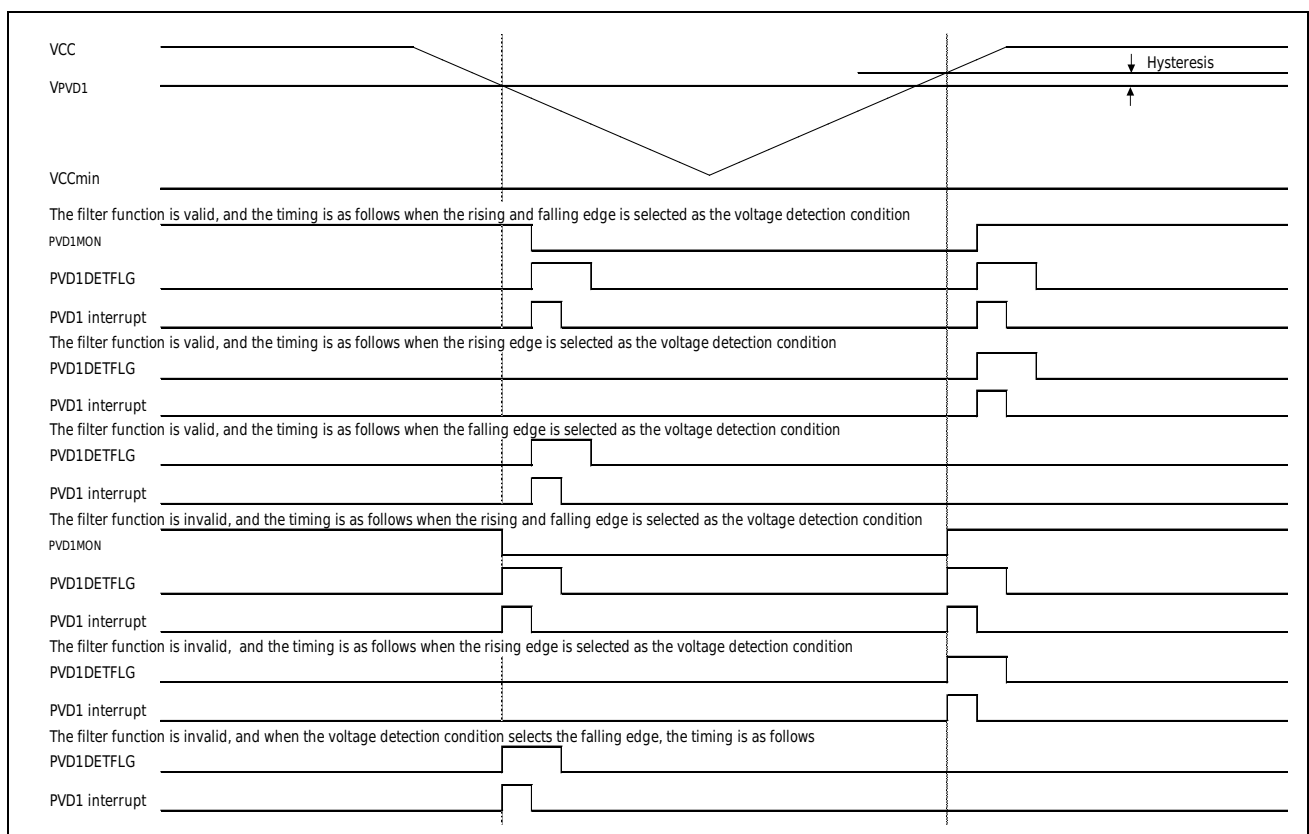


Figure 5-6 Power Monitor 1 Interrupt Timing Diagram

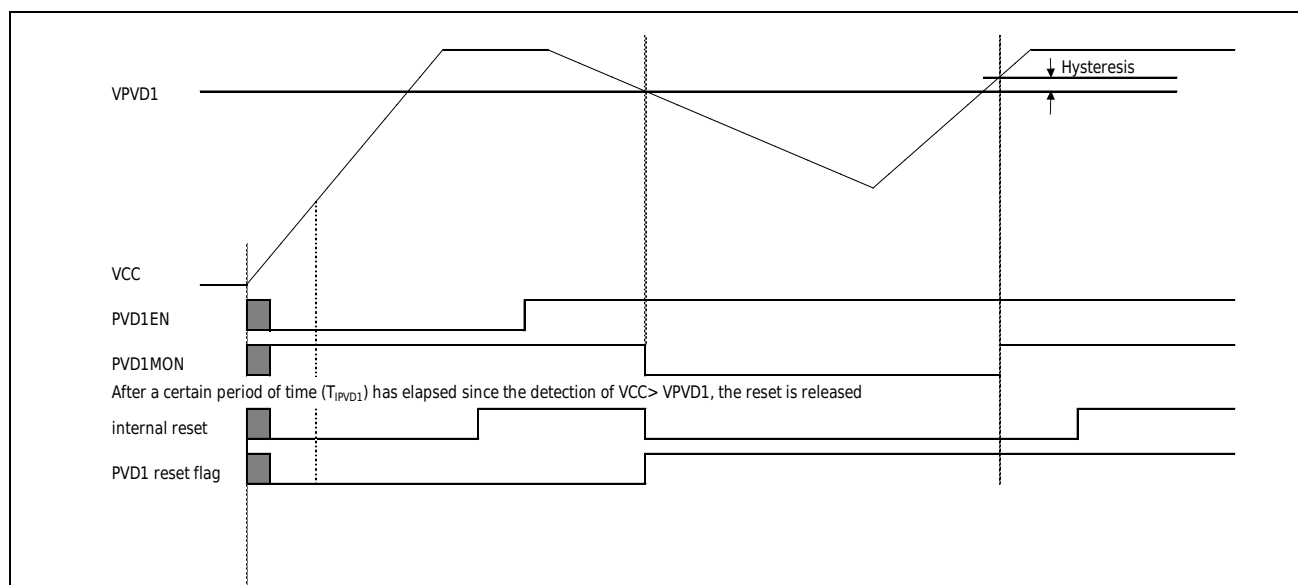


Figure 5-7 Power Monitor 1 Reset Timing Diagram

5.3.7 PVD2 Interrupt and Reset

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

- Stop mode
 - The digital filter must be disabled.
- Power Down Mode
 - The digital filter must be disabled.
 - 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.

The above figure is the operating timing diagram of the voltage detection 2 interrupt. PVD2DETFLG needs to be cleared before the interrupt can occur again.

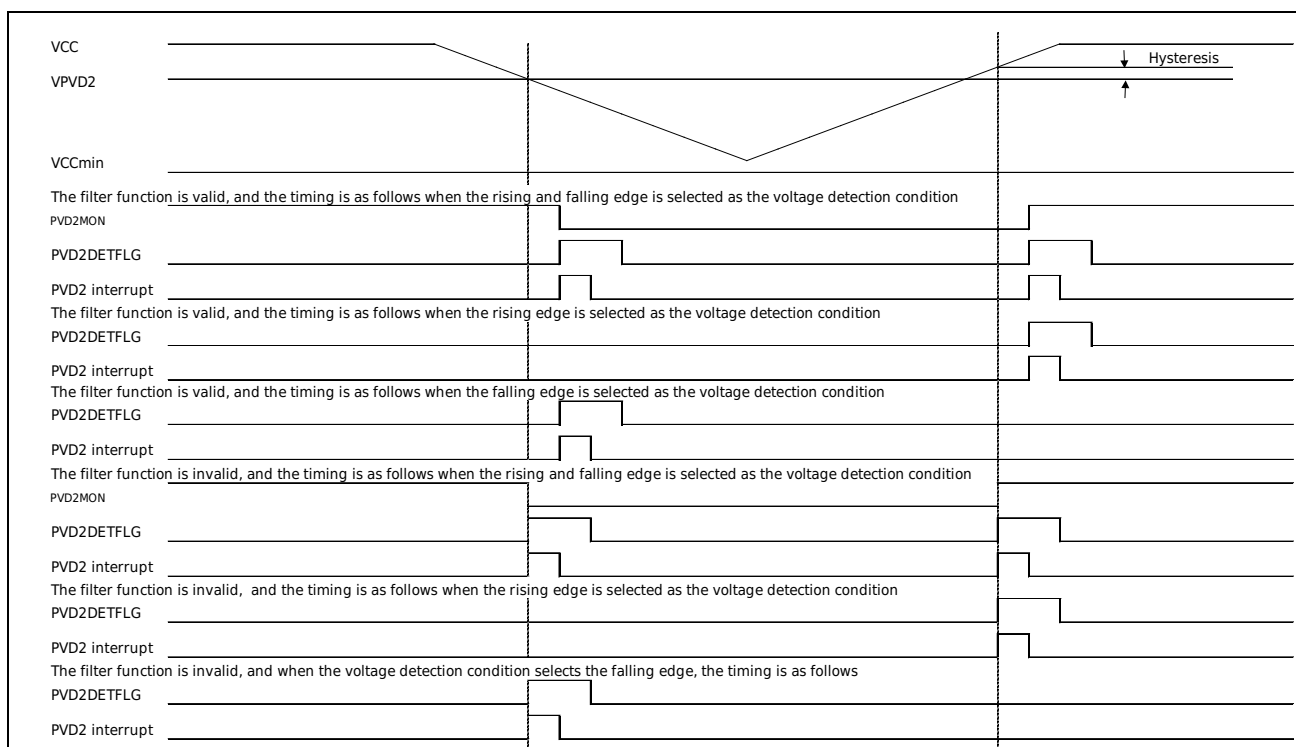


Figure 5-8 Power Supply Monitoring 2 Interrupt Operation Timing Diagram

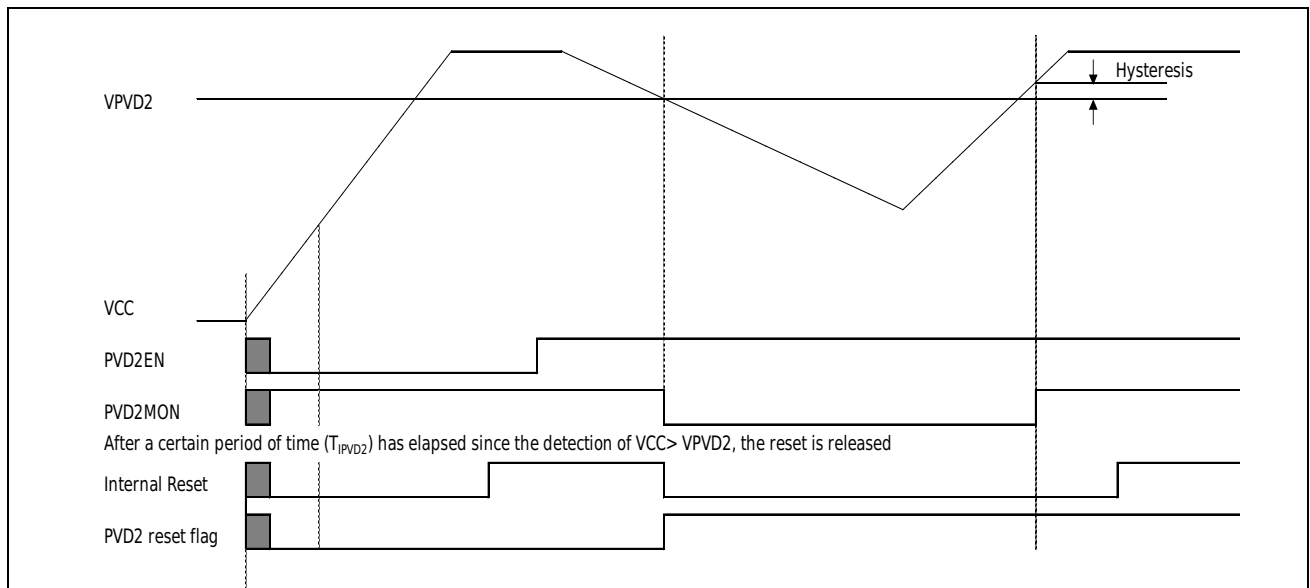


Figure 5-9 Power supply monitoring 2 reset operation sequence diagram

5.3.8 Internal Voltage Sampling and Detection Function

The internal voltage sampling and detection function of the chip refers to the reference voltage measurement path, which is the function of using the ADC to measure the reference voltage. The internal reference voltage ADVREF is approximately 1.10V.

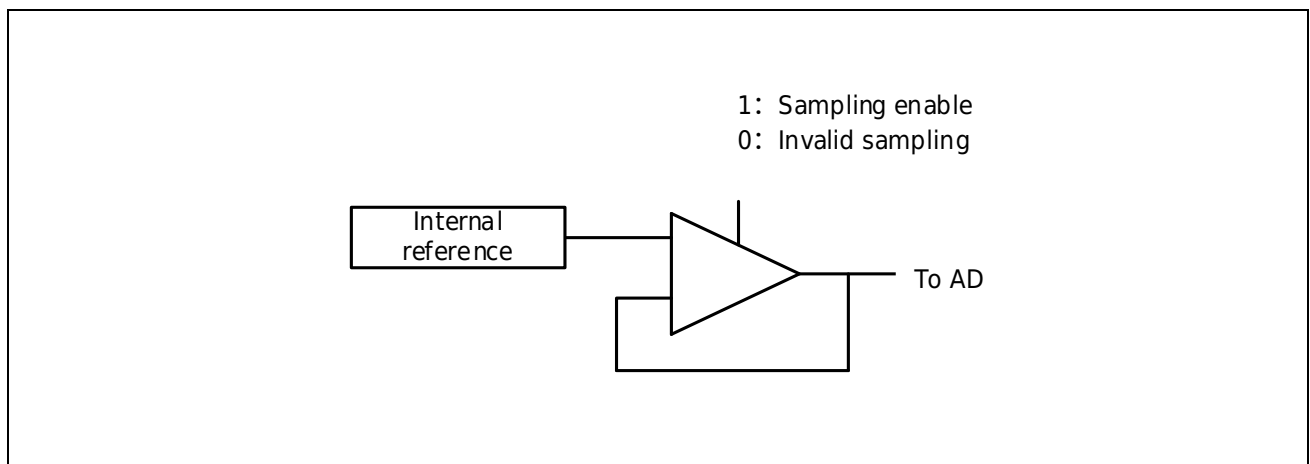


Figure 5-10 Internal Voltage Sampling Diagram

■ Reference voltage ADVREF 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. Set the ADC channel selection register and select the reference voltage ADVREF as the conversion channel
3. Use ADC to measure internal reference voltage after waiting 50uS

5.4 Wake 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 as the clock source. When the RTC uses an external low-speed oscillator as the clock action, can also select the 64Hz internal clock signal as the clock source. This counter is a count-up counter. After PWC_WKTCR.WKTCE is set, the counter starts counting. When the count value is equal to the value set by PWC_WKTCR.WKTCMP [11:0], the counter count is cleared and then counts again, generating a match consistency event. This event can be used as a normal interrupt, can be used to wake up the stop mode, and can be used to wake up the 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.

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 shown in Table 5-4. 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 Ret_SRAM can retain data. When the real-time clock, wake-up timer and Ret_SRAM in the VDDR domain are not needed, the voltage regulator RLDO of the VDDR domain can be set to be turned off, which can further reduce power consumption after entering the power-down mode.

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

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

Table 5-3 Operation Mode

Operating Mode	Description
Operating Mode (Run Mode)	Main frequency below 120MHz

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 Mode (Power 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, 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
Disable	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
PLLA	Work can be done	Stop	Power down
PLLH	Work can be done	Stop	Power down
CPU	To stop (hold).	To stop (hold).	Power down
RAM	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

Item	Sleep Mode	Stop Mode	Power Down Mode
			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 can be set Power down in power down mode 3/4
USB-FS	Work can be done	To stop (hold).	Power down
Timer0	Work can be done	Work can be done	Power down
Timer2	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
AD	Work can be done	Stop	Power down
DA	Work can be done	Work can be done	Power down
PGA	Work can be done	Work can be done	Power down
CMP	Work can be done	Work can be done	Power down
GPIO	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

Item	Sleep Mode	Stop Mode	Power Down Mode
PA11-PA12	Work can be done	Maintain; Since current will be generated when the level of this pin is pulled high, if the USB STOP mode wake-up function is not used, the pull-up is prohibited when entering STOP mode.	hold or high resistance; Since the level of this pin will generate redundant current when it is pulled high, the pull-up is prohibited when entering power-down mode.
MD	Work can be done	Maintain;	Maintain; The outside of the chip is connected to ground through a resistor
JTAG	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 dedicated watchdog does not change.

When the ICG is set to start automatically, if the WDTSLPOFF bit of ICG 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 in sleep mode does not stop counting.

When the ICG is set to start automatically, if the ICG SWDTSLPOFF bit is 1, the dedicated watchdog stops counting in sleep mode; if the SWDTSLPOFF bit is 0, the dedicated 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 clock source stops working, the voltage regulator also reduces the drive capability, so the chip power consumption will be significantly reduced.

If the SWDTSLPOFF bit of ICG is set to 1, the dedicated watchdog stops counting in stop mode. If SWDTSLPOFF bit is 0, the dedicated 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 detection 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, DAOE0, DAOE1 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, the CKSMRC and FLNWT bits of the PWC_STPMCR register are used to select the clock after waking up and whether to wait for the Flash to stabilize. CKSMRC is used to control the clock source after wake-up,

When CKSMRC=1, the wake-up system clock source is selected as MRC; when CKSMRC=0, the wake-up system clock remains the same as the clock source before entering STOP. FLNWT is used to control whether to wait for the Flash to stabilize after waking up. When FLNWT=0, it is necessary to wait for the Flash to stabilize when waking up; when FLNWT=1, it is not necessary to wait for the Flash to stabilize when waking up. FLNWT must be set when the program is running on RAM, otherwise the chip's action after waking up from STOP is not guaranteed. When the program runs on RAM and enters STOP mode, select CKSMRC = 1, FLNWT = 1 to wake up the system in the shortest time.

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

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 chip enters STOP mode after setting PWC_PWRC1.STPDAS=11 in ultra-low speed mode; if PWC_PWRC1.STPDAS=00 and enters STOP mode, the chip will consume more current in STOP mode.

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 Disabled Throu 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
- When PWC_STPMCR.STOP=1, PWC_PWRC0.PWDN=0, execute the WFI command to enter the stop mode.

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

- 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 monitoring 1/2 reset, and dedicated 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 detecting 2 interrupt, dedicated watchdog underflow interrupt, real-time clock period interrupt, alarm clock interrupt,
Wake-up timer interrupt, comparator interrupt, USART1 RX interrupt, Timer0 compare match interrupt

Timer2 compare match interrupt

When the chip interrupts the stop mode, it starts the used timer before entering the stop mode. After all of the clock source 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 SWDTSLP OFF bit of the ICG is 1, the dedicated watchdog will be the same as other modules in the VDD domain, the power will be cut off and no longer count. If the SWDTSLP OFF bit is 0, the chip will enter stop mode instead of power-down mode, and the dedicated watchdog oscillator and dedicated watchdog will continue to run if auto-start is set in ICG.

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.PDMD [1:0], the power consumption in power-down mode can be further reduced. The sub-modes of power-down mode are shown in Table 5-6. 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.

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-off mode. If the real-time clock, wake-up timer, and Ret_SRAM are not needed in power-down mode, you can set PWR_PWRC1.VVDRSD to turn off the low-power regulator to further reduce power consumption.

Table 5-6 Power-down mode submodes

Power Down Mode	PDMD [1:0]	Power Consumption	Wake-up Time	Description
Power down mode 1	00	I _{PD1}	T _{PD1}	VCC domain supply voltage detection unit is valid
Power down mode 2	01	I _{PD2}	T _{PD2}	VCC domain POR, PDR detection circuit is valid, BOR, PVD1, PVD2 are invalid
Power down mode 3	10	I _{PD3}	T _{PD3}	VCC domain POR, PDR, BOR, PVD1, PVD2 invalid
Power down mode 4	11	I _{PD4}	T _{PD4}	VCC domain POR, PDR, BOR, PVD1, PVD2 invalid

■ enter power-down mode

When PWC_STPMCR.STOP=1, PWC_PWRC0.PWDN=1, PWC_PWRC4.PD1PW=1, execute the WFI command to enter the power-down mode.

■ Disable 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=0/1/2/3) 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_PDWKFO/PWC_PDWKFI/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_PDWKFO-PWC_PDWKFI) is set. After power-down wake-up, if the power-down wake-up flag is not cleared, the chip cannot enter power-down mode again. The edge of

power-down wake-up event can be selected by PWC_PDWKES.

When waking up from power-down mode, the VDD domain will re-supply power, the system executes power-down wake-up reset, and 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_PVDPCR PWC_PVDLCR PWC_PVDICR PWC_PVDDSR
Power down mode 3	PWC_PDWKF0 PWC_PDWKF1 RMU_RSTF0

■ Pin status after releasing power-down mode

In power-down mode, the chip pins will remain in the state before entering power-down mode or in high-impedance state according to the register setting value. If PWC_PWRC0.IORTN [1:0]=10 or 11, the pin state is high-impedance in power-down mode, and the pin is initially high-impedance after the power-down mode is released. If PWC_PWRC0.IORTN [1:0]=00, 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]=01, 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, and can also select a 64Hz internal clock signal as the clock source. The counter is an up-counting counter. After WKTC0.WKTCE is set, the counter starts counting. When the count value is equal to the set value of WKTCMP [11:0], the counting stops and a wake-up event is generated to wake up the chip from power-down mode. When using WKTM again, you need to reset WKTC0.WKTCE and set it again.

■ PTWK power-down mode wake-up event

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

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

1. Configure the corresponding bit of PWC_PDWKES to select the edge
2. Configure PWC_PDWKE0 and PWC_PDWKE1 to enable the corresponding pins

3. Clear the corresponding flag in PWC_PDWKF0

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

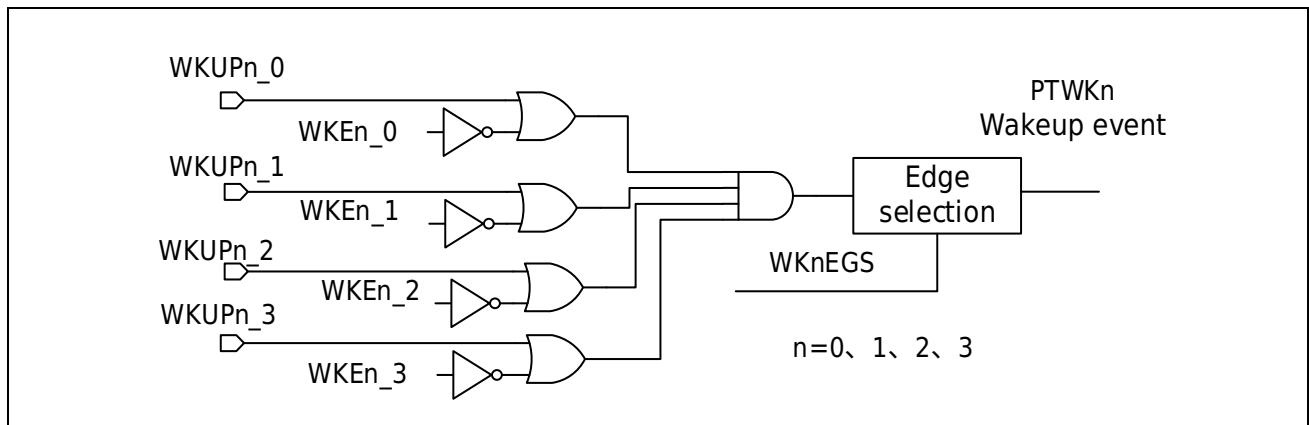


Figure 5-11 PTWKn Structure Block Diagram

■ Methods of reducing power consumption

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

1. Reduce system clock speed
2. Disable unused clock sources
3. Set the function clock control register PWC_FCGn (n=0/1/2/3) to turn off unused functions
4. Power off the RAM

5.5.4 Reduce System Clock Speed

In run mode, the speed of system clock (HCLK), external bus clock (EXCLK), 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 Disable Unused Clock Sources

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=11.

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

5.5.6 Functional Clock Stop

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	0x20000000~0x200007FFF RAM for address space	PWC_RAMPC0.RAMPDC0
SRAMH	0x1FFF8000~0x1FFFFFFF RAM for address space	PWC_RAMPC0.RAMPDC10
CAN_1 RAM	RAM for CAN unit 1	PWC_PRAMLPC.PRAMPDC0
CAN_2 RAM	RAM for CAN unit 2	PWC_PRAMLPC.PRAMPDC1
CAN_3RAM	RAM for CAN unit 3	PWC_PRAMLPC.PRAMPDC1
CACHERAM	RAM for Cache	PWC_PRAMLPC.PRAMPDC2
USBFS RAM	USBFS uses RAM	PWC_PRAMLPC.PRAMPDC3
Ret_SRAM	0x200F0000~0x200F0FFF RAM for address space	PWR_PWRC0.RETRAMSD

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_PLLHCGR, CMU_PLLHCR, CMU_PLLACFGR, CMU_PLLACR, CMU_OSCSTBSR, CMU_CKSWR, CMU_SCFGR, CMU_USBCKCFGR, CMU_CANCKCFGR, CMU_TPIUCKCFGR, CMU_MCO1CFGR, CMU_MCO2CFGR, CMU_XTAL32CR, CMU_XTALC32CFGR, CMU_XTAL32NFR, CMU_LRCCR, CMU_LRCTRM
PWC_FPRC.FPRCB1	PWC_PWRC0, PWC_PWRC1, PWC_PWRC4, PWC_PWRC5, PWC_PWRC6, PWC_PDWKE0, PWC_PDWKE1, PWC_PDWKE2, PWC_PDWKES, PWC_PDWKF0, PWC_PDWKF1, CMU_PERICKSEL, CMU_I2SCKSEL, PWC_STPMCR, PWC_RAMPC0, PWC_PRAMLPC, RMU_RSTF0, PWC_WKTCR, 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

Protected Register Bit	Protected Register
PWC_FCG0PC.B0	PWC_FCG0

5.7 Register Description

The list of registers is as Table 5-9 shown.

Table 5-9 Register Summary

BASE ADDR: 0x4004CC00

Register name	Symbol	Offset Address	Bit Width	Reset Value
Power Mode Control Register 0	PWC_PWRC0	0x00	8	0x00
Power Mode Control Register 1	PWC_PWRC1	0x04	8	0x00
Power Mode Control Register 4	PWC_PWRC4	0x10	8	0x00
PVD Control Register 0	PWC_PVDCR0	0x14	8	0x00
PVD Control Register 1	PWC_PVDCR1	0x18	8	0x00
PVD filter control register	PWC_PVDFCR	0x1C	8	0x11
PVD level control register	PWC_PVDLCR	0x20	8	0x00
Power-down wake-up enable register 0	PWC_PDWKE0	0x28	8	0x00
Power-down wake-up enable register 1	PWC_PDWKE1	0x2C	8	0x00
Power-down wake-up enable register 2	PWC_PDWKE2	0x30	8	0x00
Power-down wake-up event edge select register	PWC_PDWKES	0x34	8	0x00
Power-down wake-up flag register 0	PWC_PDWKF0	0x38	8	0x00
Power-down wake-up flag register 1	PWC_PDWKF1	0x3C	8	0x00
Power Mode Control Register 5	PWC_PWRC5	0x40	8	0x00
Power Mode Control Register 6	PWC_PWRC6	0x44	8	0x00
PVD Interrupt Control Register	PWC_PVDICR	0xC0	8	0x00
PVD Detection Status Register	PWC_PVDDSR	0xC4	8	0x11
RAM Power Control Register 0	PWC_RAMPC0	0xC8	32	0x00000000
Peripheral RAM Low Power Control Register	PWC_PRAMLPC	0xD0	32	0x00000000

BASE ADDR:0x40054000

Register Name	Symbol	Offset Address	Bit Width	Reset Value
STOP Mode Control Register	PWC_STPMCR	0x0C	16	0x0000
Function protection control register	PWC_FPRC	0x3FE	16	0x0000

BASE ADDR: 0x4004C400

Register name	Symbol	Offset Address	Bit Width	Reset Value
Wake-up Timer Control Register	PWC_WKTCR	0x00	16	0x0000

BASE ADDR:0x40048000

Register name	Symbol	Offset Address	Bit Width	Reset Value
Function Clock Control 0	PWC_FCG0	0x00	32	0xFFFFFA0E
Function Clock Control 1	PWC_FCG1	0x04	32	0xFFFFFFFF
Function Clock Control 2	PWC_FCG2	0x08	32	0xFFFFFFFF
Function Clock Control 3	PWC_FCG3	0x0C	32	0xFFFFFFFF
PWC_FCG0 protection control	PWC_FCG0PC	0x10	32	0x00000000

5.7.1 Power Mode Control Register 0 (PWC_PWRC0)

Reset value: 0x00

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

Bit	Flag	Bit Name	Function	Read and write
b7	PWDN	Power-down Mode Control Bit	0: Disable power-down mode 1: Enable power-down mode	R/W
b6	Reserved	-	Read at "0", write at "0"	R/W
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 00b to release the IO hold status. 1x: IO is high impedance in power-down mode and after power-down wake-up	R/W
b3-b2	Reserved	-	Read at "0", write at "0"	R/W
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	R/W

5.7.2 Power Mode Control Register 1 (PWC_PWRC1)

Reset value: 0xC0

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

Bit	Flag	Bit Name	Function	Read and write
b7-b6	Reserved	-	Read at "11", write at "11"	R/W
b5-b4	Reserved	-	Read at "0", write at "0"	R/W
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 25uS before starting the HRC module	R/W
b1-b0	VPLLSD [1:0]	PLL power off	00: PLL power enable 11: PLL power off 01: Disable Setting 10: Disable setting After the PLL is turned off and waits for 50us , set VPLLSD=11 and then turn off the PLL power supply to further reduce power consumption. setting VPLLSD=00 , you need to wait 25uS before starting the PLL module.	R/W

5.7.3 Power Mode Control Register 4 (PWC_PWRC4)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
ADBUFE	-	PD1PW					

Bit	Flag	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	R/W
b6	Reserved	-	Read at "0", write at "0"	R/W
b5	PD1PW	Power-down mode 1 wake-up power stabilization control	Before entering power-down Mode 1, PD1PW must be set.	R/W
b4-b0	Reserved	-	Read at "0", write at "0"	R/W

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

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

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

Reset value: 0x00

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

Bit	Flag	Bit Name	Function	Read and write
b7	WKTMWKE	WKTM wakeup event enable	0: WKTM wakeup event invalid 1: WKTM wake-up event enable	R/W
b6	Reserved	-	Read at "0", write at "0"	R/W
b5	RTCALMWKE	RTC alarm clock wake-up event enable	0: RTC alarm clock wake-up event invalid 1: RTC alarm clock wake-up event enable	R/W
b4	RTCPRDWKE	RTC period wakeup event enable	0: RTC period wake-up event invalid 1: RTC period wakeup event enable	R/W
b3-b2	Reserved	-	Read at "0", write at "0"	R/W
b1	VD2WKE	PVD2 wakeup event enable	0: PVD2 wakeup event invalid 1: PVD2 wakeup event enable	R/W
b0	VD1WKE	PVD1 wakeup event enable	0: PVD1 wakeup event invalid 1: PVD1 wakeup event enable	R/W

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

Reset value: 0x00

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

Bit	Flag	Bit Name	Function	Read and write
b7-b6	Reserved	-	Read at "0", write at "0"	R/W
b5	VD2EGS	VD2 edge selection	0: VCC < VPVD2 1: VCC > VPVD2	R/W
b4	VD1EGS	VD1 edge selection	0: VCC < VPVD1 1: VCC > VPVD1	R/W
b3	WK3EGS	PTWK3 edge selection	0: Falling edge 1: Rising edge	R/W
b2	WK2EGS	PTWK2 edge selection	0: Falling edge 1: Rising edge	R/W
b1	WK1EGS	PTWK1 edge selection	0: Falling edge 1: Rising edge	R/W
b0	WK0EGS	PTWK0 edge selection	0: Falling edge 1: Rising edge	R/W

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

Reset value: 0x00

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

Bit	Flag	Bit Name	Function	Read and write
b7-b6	Reserved	-	Read at "0", write at "0"	R/W
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.	R/W
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.	R/W
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.	R/W
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.	R/W
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.	R/W
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.	R/W

5.7.9 Power-down Wake-up Flag Register 1 (PWC_PDWK1)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
WKTMMWK	-	RTCALMWK	RTCPDWK	-	-	-	

Bit	Flag	Bit Name	Function	Read and write
b7	WKTMMWK	WKTMM wakeup flag bit	0: No WKTMM wakeup event occurs 1: A WKTMM wakeup event occurs After power-down wake-up, you need to write zero to clear this bit.	R/W
b6	Reserved	-	Read at "0", write at "0"	R/W
b5	RTCALMWK	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.	R/W
b4	RTCPDWK	RTC period wake-up flag bit	0: No RTC period wake-up event occurs 1: RTC period wake-up event occurs After power-down wake-up, you need to write zero to clear this bit.	R/W
b3~0	Reserved	-	Read at "0", write at "0"	R/W

5.7.10 Power Mode Control Register 5 (PWC_PWRC5)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
CSDIS	-	-	-	-	-	RETRMSD	VVDRSD

Bit	Flag	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, CSDIS can be set to reduce power consumption.	R/W
b6	Reserved	-	Read at "0", write at "0"	R/W
b5	Reserved	-	Read at "0", write at "0"	R/W
b4	Reserved	-	Read at "0", write at "0"	R/W
b3	Reserved	-	Read at "0", write at "0"	R/W
b2	Reserved	-	Read at "0", write at "0"	R/W
b1	RETRMSD	Maintain RAM Power Down Control	0: Ret_SRAM does not power down 1: Ret_SRAM power down	R/W
b0	VVDRSD	RLDO close control bit	0: RLDO is valid 1: RLDO invalid After VVDRSD is set and the chip enters power-down mode, the power supply of RT/WTKM/Ret_SRAM is turned off to reduce power consumption.	R/W

5.7.11 Power Mode Control Register 6 (PWC_PWRC6)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1-b0
WDRTNE	SWRTNE	WDRDAC	SWRDAC	-	IORTNCLR	RTCCKSEL [1:0]

Bit	Flag	Bit Name	Function	Read and write
b7	WDRTNE	-	0: WDT reset /SWDT reset can reset IO 1: WDT reset /SWDT reset will not reset IO. You can reset IO by writing "1" to clear this bit in IORTNCLR.	R/W
b6	SWRTNE	-	0: Software reset can reset IO 1: Software reset will not reset the IO. You can reset the IO by writing "1" to the IORTNCLR bit to clear this bit.	R/W
b5	WDRDAC	-	0: WDT reset /SWDT reset can reset DAC 1: WDT reset /SWDT reset will not reset the DAC. You can reset the DAC by writing the RMU_FRST3.DAC1 software reset bit.	R/W
b4	SWRDAC	-	0: Software reset can reset DAC 1: Software reset will not reset the DAC. You can reset the DAC by writing RMU_FRST3.DAC1 bit.	R/W
b3	Reserved	-	Read at "0", write at "0"	R/W
b2	IORTNCLR	-	When a software reset occurs when SWRTNE=1 , IO maintains the state before reset; when a WDT reset /SWDT reset occurs when WDRTNE=1 , IO maintains the state before reset. By writing 1 to this bit , the IO hold status caused by the software reset generated when SWRTNE=1 is cleared , and the IO hold status caused by the WDT reset /SWDT reset generated when WDRTNE=1 is cleared. The read value of this bit is 0.	R/W
b1-0	RTCCKSEL [1:0]	-	00: LRC 01: XTAL Fractional Frequency 10 : Terminal input (RTC_CLK1) 11 : Terminal input (RTC_CLK2)	R/W

5.7.12 Function Clock Control 0 (PWC_FCG0)

Reset value: 0xFFFFFA0E

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	DCU3	DCU2	DCU1	CRC	TRNG	HASH	AES	MAU	CTC	AOS	FCM
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DMA2	DMA1	KEY	-	PLA	SRAMRET	-	-	-	-	-	SRAM0	-	-	-	SRAMH

Bit	Flag	Bit Name	Function	Read and write
b31-b27	Reserved	-	Read at "1", write at "1"	R/W
b26	DCU3	DCU3 Functional Control	0: Digital computing unit DCU3 function enable 1: Digital Computing Unit DCU3 function invalid	R/W
b25	DCU2	DCU2 Functional Control	0: Digital computing unit DCU2 function enable 1: Digital Computing Unit DCU2 function invalid	R/W
b24	DCU1	DCU1 Functional Control	0: Digital computing unit DCU1 function enable 1: Digital Computing Unit DCU1 function invalid	R/W
b23	CRC	CRC function control	0: CRC function enable 1: Invalid CRC function	R/W
b22	TRNG	TRNG Functional Control	0: The true random generator TRNG function in the encryption co-processing module CPM is enabled 1: The true random generator TRNG function in the encryption co-processing module CPM is invalid	R/W
b21	HASH	HASH functional control	0: Enable the HASH function of the secure hash algorithm module in the encryption co-processing module CPM 1: The secure hash algorithm module HASH function in the encryption co-processing module CPM is invalid	R/W
b20	AES	AES functional control	0: Enable the AES function of the encryption and decryption algorithm processor in the encryption co-processing module CPM 1: The AES function of the encryption and decryption algorithm processor in the encryption co-processing module CPM is invalid	R/W
b19	MAU	MAU functional control	0: Mathematical operation unit (MAU) function enabled 1: Mathematical operation unit MAU function is invalid	R/W
b18	CTC	CTC Functional Control	0: Internal clock calibrator CTC function enable 1: Internal clock calibrator CTC function invalid	R/W
b17	AOS	AOS Functional Control	0: Automatic operation system AOS function enable 1: Automatic operation system AOS function invalid	R/W
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	R/W
b15	DMA2	DMA2 Functional Control	0: DMA controller DMA2 function enable 1: DMA controller DMA2 function is invalid	R/W
b14	DMA1	DMA1 Functional Control	0: DMA controller DMA1 function enable 1: DMA controller DMA1 function is invalid	R/W
b13	KEY	KEYSCAN function control	0: The KEYSCAN function of the keyboard scanning control module is enabled 1: The KEYSCAN function of the keyboard scanning control module is invalid	R/W
b12	Reserved	-	Read at "1", write at "1"	R/W
b11	PLA	PLA Functional Control	0: PLA function enable 1: Invalid PLA function	R/W
b10	SRAMRET	Ret_SRAM function control	0: Ret_SRAM function enabled 1: Ret_SRAM function disabled	R/W
b9-b5	Reserved	-	When reading, it is "5'b10000", when writing, it is written as "5'b10000"	R/W
b4	SRAM0	SRAM0 Functional Control	0: SRAM0 (0x20000000~0x20007FFF) function enabled 1: SRAM0 (0x20000000~0x20007FFF) function is invalid	R/W
b3-b1	Reserved	-	Read at "1", write at "1"	R/W
b0	SRAMH	SRAMH Functional Control	0: SRAMH (0x1FFF8000~0x1FFFFFFF) Function Enable 1: SRAMH (0x1FFF8000~0x1FFFFFFF) Function Enable	R/W

5.7.13 Function Clock Control 1 (PWC_FCG1)

Reset value: 0xFFFFFFFF

b31	b30	b29	b28	b27	b26	b25	b24
-	-	-	-	FMAC4	FMAC3	FMAC2	FMAC1
b23	b22	b21	b20	b19	b18	b17	b16
-	USBFS	-	-	SPI4	SPI3	SPI2	SPI1
b15	b14	b13	b12	b11	b10	b9	b8
-	-	-	-	-	-	-	MDIO
b7	b6	b5	b4	b3	b2	b1	b0
-	I2C3	I2C2	I2C1	QSPI	CAN3	CAN2	CAN1

Bit	Flag	Bit Name	Function	Read and write
b31-b28	Reserved	-	Read at "1", write at "1"	R/W
b27	FMAC4	FMAC4 Functional Control	0: Filter math accelerator FMAC unit 4 function enabled 1: Filter math accelerator FMAC unit 1 function is invalid	R/W
b26	FMAC3	FMAC3 Functional Control	0: Filter math accelerator FMAC unit 3 function enabled 1: Filter math accelerator FMAC unit 1 function is invalid	R/W
b25	FMAC2	FMAC2 Functional Control	0: Filter math accelerator FMAC unit 2 function enabled 1: Filter math accelerator FMAC unit 1 function is invalid	R/W
b24	FMAC1	FMAC1 Functional Control	0: Filter math accelerator FMAC unit 1 function enabled 1: Filter math accelerator FMAC unit 1 function is invalid	R/W
b23	Reserved	-	Read at "1", write at "1"	R/W
b22	USBFS	USBFS Functional Control	0: USB2.0 full speed module USBFS function enable 1: The USBFS function of the USB2.0 full-speed module is invalid	R/W
b21~b20	Reserved	-	Read at "1", write at "1"	R/W
b19	SPI4	SPI4 Functional Control	0: Serial peripheral interface SPI Unit 4 Function Enable 1: Serial peripheral Interface SPI Unit 4 function invalid	R/W
b18	SPI3	SPI3 Functional Control	0: Serial peripheral interface SPI Unit 3 Function Enable 1: Serial peripheral Interface SPI Unit 3 function invalid	R/W
b17	SPI2	SPI2 Functional Control	0: Serial peripheral interface SPI Unit 2 Function Enable 1: Serial peripheral Interface SPI Unit 2 function invalid	R/W
b16	SPI1	SPI1 Functional Control	0: Serial peripheral interface SPI Unit 1 Function Enable 1: SPI Unit 1 Function Invalid	R/W
b15~b9	Reserved	-	Read at "1", write at "1"	R/W
b8	MDIO	MDIO Functional Control	0: MDIO function enable 1: Invalid MDIO function	R/W
b7	Reserved	-	Read at "1", write at "1"	R/W
b6	I2C3	I2C3 Functional Control	0: IC bus I2C unit 3 function enabled 1: Integrated circuit bus I2C unit 3 function invalid	R/W
b5	I2C2	I2C2 Functional Control	0: IC bus I2C unit 2 function enabled 1: Integrated circuit bus I2C unit 2 function invalid	R/W
b4	I2C1	I2C1 Functional Control	0: IC bus I2C unit 1 function enabled 1: The IC bus I2C unit 1 function is invalid.	R/W

b3	QSPI	QSPI functional control	0: Four-wire serial peripheral interface QSPI function enable 1: Four-wire serial peripheral interface QSPI function is invalid	R/W
b2	CAN3	CAN3 Functional Control	0: CAN Unit 3 Function Enable 1: CAN Unit 3 Function Invalid	R/W
b1	CAN2	CAN2 Functional Control	0: CAN Unit 2 Function Enable 1: CAN Unit 2 Function Invalid	R/W
b0	CAN1	CAN1 Functional Control	0: CAN Unit 1 Function Enable 1: CAN Unit 1 Function Invalid	R/W

5.7.14 Function Clock Control 2 (PWC_FCG2)

Reset value: 0xFFFFFFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	TMRA_6	TMRA_5	TMRA_4	TMRA_3	TMRA_2	TMRA_1	TMR2_4	TMR2_3	TMR2_2	TMR2_1
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EMB	-	TMR0_2	TMR0_1	-	TMR4	TMR6_10	TMR6_9	TMR6_8	TMR6_7	TMR6_6	TMR6_5	TMR6_4	TMR6_3	TMR6_2	TMR6_1

Bit	Flag	Bit Name	Function	Read and write
b31~b26	Reserved	-	Read at "1", write at "1"	R/W
b25	TMRA_6	TMRA_6 Functional Control	0: TimerA Unit 6 Function Enable 1: TimerA Unit 6 Function Invalid	R/W
b24	TMRA_5	TMRA_5 Functional Control	0: TimerA Unit 5 Function Enable 1: TimerA Unit 5 Function Invalid	R/W
b23	TMRA_4	TMRA_4 Functional Control	0: TimerA Unit 4 Function Enable 1: TimerA Unit 4 Function Invalid	R/W
b22	TMRA_3	TMRA_3 Functional Control	0: TimerA Unit 3 Function Enable 1: TimerA Unit 3 Function Invalid	R/W
b21	TMRA_2	TMRA_2 Functional Control	0: TimerA Unit 2 Function Enable 1: TimerA Unit 2 Function Invalid	R/W
b20	TMRA_1	TMRA_1 Functional Control	0: TimerA Unit 1 Function Enable 1: TimerA Unit 1 Function Invalid	R/W
b19	TMR2_4	TMR2_4 Functional Control	0: Timer2 Unit 4 Function Enable 1: Timer2 Unit 4 Function Invalid	R/W
b18	TMR2_3	TMR2_3 Functional Control	0: Timer2 Unit 3 Function Enable 1: Timer2 Unit 3 Function Invalid	R/W
b17	TMR2_2	TMR2_2 Functional Control	0: Timer2 Unit 2 Function Enable 1: Timer2 Unit 2 Function Invalid	R/W
b16	TMR2_1	TMR2_1 Functional Control	0: Timer2 Unit 1 Function Enable 1: Timer2 Unit 1 Function Invalid	R/W
b15	EMB	EMB Functional Control	0: Enable the EMB function of the emergency brake module 1: Emergency brake module EMB function invalid	R/W
b14	Reserved	-	Read at "1", write at "1"	R/W
b13	TMR0_2	TMR0_2 Functional Control	0: Timer0_2 function enabled 1: Invalid Timer0_2 function	R/W
b12	TMR0_1	TMR0_1 Functional Control	0: Timer0_1 function enabled 1: Invalid Timer0_1 function	R/W
b11	Reserved	-	Read at "1", write at "1"	R/W
b10	TMR4	TMR4 Functional Control	0: Timer4 function enabled 1: Invalid Timer4 function	R/W
b9	TMR6_10	TMR6_10 Functional Control	0: Timer6 Unit 10 Function Enable 1: Timer6 Unit 10 Function Invalid	R/W
b8	TMR6_9	TMR6_9 Functional Control	0: Timer6 Unit 9 Function Enable 1: Timer6 Unit 9 Function Invalid	R/W
b7	TMR6_8	TMR6_8 Functional Control	0: Timer6 Unit 8 Function Enable 1: Timer6 Unit 8 Function Invalid	R/W
b6	TMR6_7	TMR6_7 Functional Control	0: Timer6 Unit 7 Function Enable 1: Timer6 Unit 7 Function Invalid	R/W
b5	TMR6_6	TMR6_6 Functional Control	0: Timer6 Unit 6 Function Enable 1: Timer6 Unit 6 Function Invalid	R/W
b4	TMR6_5	TMR6_5 Functional Control	0: Timer6 Unit 5 Function Enable 1: Timer6 Unit 5 Function Invalid	R/W
b3	TMR6_4	TMR6_4 Functional Control	0: Timer6 Unit 4 Function Enable 1: Timer6 Unit 4 Function Invalid	R/W
b2	TMR6_3	TMR6_3 Functional Control	0: Timer6 Unit 3 Function Enable 1: Timer6 Unit 3 Function Invalid	R/W
b1	TMR6_2	TMR6_2 Functional Control	0: Timer6 Unit 2 Function Enable 1: Timer6 Unit 2 Function Invalid	R/W
b0	TMR6_1	TMR6_1 Functional Control	0: Timer6 Unit 1 Function Enable 1: Timer6 Unit 1 Function Invalid	R/W

5.7.15 Function Clock Control 3 (PWC_FCG3)

Reset value: 0xFFFFFFFF

b31	b30	b29	b28	b27	b26	b25	b24
-	-	-	-	-	-	USART6	USART5
b23	b22	b21	b20	b19	b18	b17	b16
USART4	USART3	USART2	USART1	-	-	-	SMC
b15	b14	b13	b12	b11	b10	b9	b8
-	-	-	OTS	-	-	-	CMP12
b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	DAC1	CMBIAS	ADC3	ADC2	ADC1

Bit	Flag	Bit Name	Function	Read and write
b31-b30	Reserved	-	Read at "1", write at "1"	R/W
b25	USART6	USART6 Functional Control	0: General Synchronous Asynchronous Receiver/Transmitter USART Unit 6 Function Enable 1: General Synchronous Asynchronous Receiver-Transmitter USART Unit 6 Function Invalid	R/W
b24	USART5	USART5 Functional Control	0: General Synchronous Asynchronous Receiver/Transmitter USART Unit 5 Function Enable 1: General Synchronous Asynchronous Receiver-Transmitter USART Unit 5 Function Invalid	R/W
b23	USART4	USART4 Functional Control	0: General Synchronous Asynchronous Receiver/Transmitter USART Unit 4 Function Enable 1: General Synchronous Asynchronous Receiver-Transmitter USART Unit 4 Function Invalid	R/W
b22	USART3	USART3 Functional Control	0: General Synchronous Asynchronous Receiver/Transmitter USART Unit 3 Function Enable 1: General Synchronous Asynchronous Receiver-Transmitter USART Unit 3 Function Invalid	R/W
b21	USART2	USART2 Functional Control	0: General Synchronous Asynchronous Receiver/Transmitter USART Unit 2 Function Enable 1: General Synchronous Asynchronous Receiver-Transmitter USART Unit 2 Function Invalid	R/W
b20	USART1	USART1 Functional Control	0: General Synchronous Asynchronous Receiver/Transmitter USART Unit 1 Function Enable 1: General Synchronous Asynchronous Receiver-Transmitter USART Unit 1 Function Invalid	R/W
b19	Reserved	-	Read at "1", write at "1"	R/W
b16	SMC	SMC Functional Control	0: The SMC controller function of the external storage controller (EXMC) is enabled 1: The SMC controller function of the external storage controller (EXMC) is invalid	R/W
b15-b13	Reserved	-	Read at "1", write at "1"	R/W
b12	OTS	OTS Functional Control	0: The temperature sensor OTS function is valid 1: The temperature sensor OTS function is invalid	R/W
b11-b9	Reserved	-	Read at "1", write at "1"	R/W
b8	CMP12	CMP12 Functional Control	0: Voltage comparator CMP channel 1 and 2 functions are enabled 1: The voltage comparator CMP channel 1 and 2 functions are invalid.	R/W
b7-b5	Reserved	-	Read at "1", write at "1"	R/W
b4	DAC1	DAC1 Functional Control	0: DAC unit 1 Function Enable 1: DAC Unit 1 Function Invalid	R/W
b3	CMBIAS	CMP/PGA Reference Current Source Control	0: CMP/PGA reference current source is valid 1: CMP/PGA reference current source is invalid	R/W
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	R/W
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	R/W
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	R/W

5.7.16 PWC_FCG0 Protection Control (PWC_FCG0PC)

Reset value: 0x00000000

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
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PRT0

Bit	Flag	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	R/W
b15-b1	Reserved	-	Read at "0", write at "0"	R/W
b0	PRT0	Protection bit	PWC_FCG0 write enable control bit 0: PWC_FCG0 write invalid 1: PWC_FCG0 write enable	R/W

5.7.17 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	FPRCB2	FPRCB1	FPRCB0

Bit	Flag	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.	R/W
b7	FPRCB7	FPRC bit 7	Reserved, "0" when reading, "0" when writing	R/W
b6	FPRCB6	FPRC bit 6	Reserved, "0" when reading, "0" when writing	R/W
b5	FPRCB5	FPRC bit 5	Reserved, "0" when reading, "0" when writing	R/W
b4	FPRCB4	FPRC bit 4	Reserved, "0" when reading, "0" when writing	R/W
b3	FPRCB3	FPRC bit 3	Protection Register Bits, Protection Object refer to Table 5-8 0: Write protection 1: Write enable	R/W
b2	FPRCB2	FPRC bit 2	Reserved, "0" when reading, "0" when writing	R/W
b1	FPRCB1	FPRC bit 1	Protection Register Bits, Protection Object refer to Table 5-8 0: Write protection 1: Write enable	R/W
b0	FPRCB0	FPRC bit 0	Protection Register Bits, Protection Object refer to Table 5-8 0: Write protection 1: Write enable	R/W

5.7.18 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	EXBUSOE	-	-	-	-	-	-	-	-	-	-	-	-	CKSMRC	FLNWT

Bit	Flag	Bit Name	Function	Read and write
b15	STOP	STOP mode selection bit	0: STOP mode is invalid 1: STOP mode is valid	R/W
b14	EXBUSOE	EXMC address bus and output control signal output enable	State control of address bus signals and output control signals of the external bus in power-down mode and stop mode 0: High impedance 1: Keep the state before power-down mode and stop mode	R/W
b13-b2	Reserved	-	Read at "0", write at "0"	R/W
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	R/W
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	R/W

5.7.19 RAM Power Control Register 0 (PWC_RAMPC0)

Reset value: 0x00000000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	RAMPDC10	-	-
7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	RAMPDC0

Bit	Flag	Bit Name	Function	Read and write
b31-b11	Reserved	-	Read at "0", write at "0"	R/W
b10	RAMPDC10	RAM power down control bit 10	0: 0x1FFF8000~0x1FFFFFFF space RAM action 1: 0x1FFF8000~0x1FFFFFFF RAM power off	R/W
b9~b1	Reserved	-	Read at "0", write at "0"	R/W
b0	RAMPDC0	RAM power down control bit 0	0: 0x20000000~0x20007FFF space RAM action 1: 0x20000000~0x20007FFF RAM power off	R/W

5.7.20 Peripheral RAM Low Power Control Register (PWC_PRAMLPC)

Reset value: 0x00000000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-
7	6	5	4	3	2	1	0
-	-	-	-	PRAMPDC3	PRAMPDC2	PRAMPDC1	PRAMPDC0

Bit	Flag	Bit Name	Function	Read and write
b31~b4	Reserved	-	Read at "0", write at "0"	R/W
b3	PRAMPDC3	Peripheral RAM Power-down Control Bit 3	USBFS RAM power down control 0: No power down 1: power down	R/W
b2	PRAMPDC2	Peripheral RAM Power-down Control Bit 2	CACHE_RAM power-off control 0: No power down 1: power down	R/W
b1	PRAMPDC1	Peripheral RAM Power-down Control Bit 1	CAN_2 RAM/CAN_3 RAM power-off control 0: No power down 1: power down	R/W
b0	PRAMPDC0	Peripheral RAM Power-down Control Bit 0	CAN_1 RAM power-down control 0: No power down 1: power down	R/W

5.7.21 PVD Control Register 0 (PWC_PVDCR0)

Reset value: 0x00

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

Bit	Flag	Bit Name	Function	Read and write
b7	Reserved	-	Read at "0", write at "0"	R/W
b6	PVD2EN	Voltage detection 2 permissible	0: Voltage detection 2 circuit invalid 1: The voltage detection 2 circuit valid	R/W
b5	PVD1EN	Voltage detection 1 permissible	0: Voltage detection 1 circuit invalid 1: The voltage detection 1 circuit valid	R/W
b4-b1	Reserved	-	Read at "0", write at "0"	R/W
b0	EXVCCINEN	External VCC voltage input enable	0: External VCC voltage input invalid 1: External VCC voltage input valid	R/W

5.7.22 PVD Control Register 1 (PWC_PVDCR1)

Reset value: 0x00

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

Bit	Flag	Bit Name	Function	Read and write
b7	Reserved	-	Read at "0", write at "0"	R/W
b6	PVD2CMPOE	PVD2 comparison result output enable	0: Disable output of comparison result of comparator 2 1: Enable output the comparison result of comparator 2	R/W
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".	R/W
b4	PVD2IRE	PVD2 interrupt reset enable	0: Disable 1: Enable Note: Please write "1" to the PVD2IRE bit when the PVD2EN bit is "1" and the PVD2CMPOE bit is "1"	R/W
b3	Reserved	-	Read at "0", write at "0"	R/W
b2	PVD1CMPOE	PVD1 comparison result output enable	0: Disable output of comparison result of comparator 1 1: Enable output the comparison result of comparator 1	R/W
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: 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".	R/W
b0	PVD1IRE	PVD1 interrupt reset enable	0: Disable 1: Enable Note: Please write "1" to the PVD1IRE bit when the PVD1EN bit is "1" and the PVD1CMPOE bit is "1"	R/W

5.7.23 PVD Filter Control Register (PWC_PVDFCR)

Reset value: 0x11

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

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

5.7.24 PVD Level Control Register (PWC_PVDLCR)

Reset value: 0x00

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

Bit	Flag	Bit Name	Function	Read and write
b7	Reserved	-	Read at "0", write at "0"	R/W
b6~b4	PVD2LVL [2:0]	PVD2 threshold voltage selection	000: 2.1V	R/W
			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)	
b3	Reserved	-	Read at "0", write at "0"	R/W
b2~b0	PVD1LVL [2:0]	PVD1 threshold voltage selection	000: 2.0V	R/W
			001: 2.1V	
			010: 2.3V	
			011: 2.5V	
			100: 2.6V	
			101: 2.7V	
			110: 2.8V	
			111: 2.9V	

5.7.25 PVD Interrupt Control Register (PWC_PVDICR)

Reset value: 0x00

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

Bit	Flag	Bit Name	Function	Read and write
b7	Reserved	-	Read at "0", write at "0"	R/W
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: Disable setting	R/W
b4	PVD2NMIS	PVD2 interrupt type selection	0: PVD2 interrupt as non-maskable interrupt 1: PVD2 interrupt as a maskable interrupt	R/W
b3	Reserved	-	Read at "0", write at "0"	R/W
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: Disable setting	R/W
b0	PVD1NMIS	PVD1 interrupt type selection	0: PVD1 interrupt as non-maskable interrupt 1: PVD1 interrupt as a maskable interrupt	R/W

5.7.26 PVD Detection Status Register (PWC_PVDDSR)

Reset value: 0x11

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

Bit	Flag	Bit Name	Function	Read and write
b7~b6	Reserved	-	Read at "0", write at "0"	R/W
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"	R/W
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	Reserved	-	Read at "0", write at "0"	R/W
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"	R/W
b0	PVD1MON	PVD1 Monitoring Bit	0: VCC < VPVD1 1: When PVD1 is invalid or VCC > VPVD1	R

5.7.27 Wake-up Timer Control Register (PWC_WKTCR)

Reset value: 0x0000

b15	b14-b13	b12	b11-b0
WKTCE	WKCKS [1:0]	WKOVF	WKTMCMP [11:0]

Bit	Flag	Bit Name	Function	Read and write
b15	WKTCE	WKT enable	0: WKT stop 1: WKT count Clearing the WKTCE bit while WKT is counting will clear the counter and stop the counter.	R/W
b14-b13	WKCKS [1:0]	WKT clock selection	00: 64Hz clock 01: XTAL32 10: LRC 11: Reserved	R/W
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.	R/W
b11-b0	WKTMCMP [11:0]	WKT compare bit	Comparison value of WKT counter	R/W

6 Initialization Configuration (ICG)

6.1 Introduction

After the chip reset is released, the hardware circuit will read the Flash address 0x00000400~0x0000045F and load the data into the initialization configuration register. Addresses 0x0000_0408~0x0000_040B , 0x0000_0414~0x0000_041F , and 0x00000438~0x0000045F are reserved functions. Please write all 1s to ensure normal operation of the chip. Flash boot swap is valid and OTP is not enabled, this area is located in Flash block 1 sector 0; otherwise, this area exists in Flash block 0 sector 0. Users can modify the initial configuration (Initial Config) 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.

The initialization configuration register address list is as follows:

Table 6-1 Register Summary

ICG_BASE_ADDR:0x00000400

Register name	Symbol	Offset Address	Bit Width	Reset Value
Initialize Configuration Register0	ICG0	0x000	32	Indefinite
Initialize Configuration Register 1	ICG1	0x004	32	Indefinite
Initialize Configuration Register 3	ICG3	0x00C	32	Indefinite
Initialize Configuration Register 4	ICG4	0x010	32	Indefinite

6.2 Register Description

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
-	-	-	WDT SLPO FF	WDTWDPT[3:0]				WDTCKS [3:0]				WDTPERI [1:0]	WDTIT S	WDTA UTS	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	SWD TSLP OFF	SWDWDPT[3:0]				SWDTCKS [3:0]				SWDTPERI [1:0]	SWDTI TS	SWDT AUTS	

Bit	Flag	Bit Name	Function	Read and write
b31~b29	Reserved	-	Functional reservation bit	R
b28	WDTSLPOFF	WDT counting stops in sleep mode	0: WDT does not stop counting in sleep mode 1: WDT stops counting in sleep mode	R
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%				
b27~b24	WDTWDPT [3:0]	Refresh Allowed Area Count%	0111: 0%~75%	R
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%				
0010: PCLK3/4				
0110: PCLK3/64				
0111: PCLK3/128				R
1000: PCLK3/256				
b23~b20	WDTCKS [3:0]	WDT counter clock	1001: PCLK3/512	
1010: PCLK3/1024				
1011: PCLK3/2048				
1101: PCLK3/8192				
Other values: Reserved				
b19~b18	WDTPERI [1:0]	WDT count overflow period	00: 256 periods 01: 4096 periods 10: 16384 periods 11: 65536 periods	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	Reserved	-	Functional reservation bit	R

b12	SWDTSLP0FF	SWDT count stops in Sleep, Stop mode	0: SWDT counts do not stop in sleep, stop mode 1: SWDT count stopped in sleep, stop mode 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
b11~b8	SWDWDPT [3:0]	Refresh Allowed Area Count%	0000: SWDTCLK 0100: SWDTCLK/16 0101: SWDTCLK/32 0110: SWDTCLK/64 0111: SWDTCLK/128 1000: SWDTCLK/256 1011: SWDTCLK/2048 Other values: Reserved	R
b7~b4	SWDTCKS [3:0]	SWDT counter clock	00: 256 periods 01: 4096 periods 10: 16384 periods 11: 65536 periods	R
b3~b2	SWDTPERI [1:0]	SWDT count overflow period	0: Interrupt request 1: Reset request	R
b1	SWDTITS	SWDTInterrupt selection	0: SWDT starts automatically after reset (hardware starts) 1: SWDT stops after reset	R
b0	SWDTAUTS	SWDT auto start		R

6.2.2 Initialization Configuration register1 (ICG1)

Reset value: Indefinite

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

Bit	Flag	Bit Name	Function	Read and write
b31~b19	Reserved	-	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	Reserved	-	Functional reservation bit	R
b8	HRCSTOP	HRC Oscillation Stop Bit	0: HRC oscillation 1: HRC stop	R
b7~b1	Reserved	-	Read as "1", write as "1"	R
b0	HRCFREQSEL	HRC frequency selection	0: 20MHz 1: 16MHz	R

6.2.3 Initialize Configuration Register 3 (ICG3)

Reset value: Indefinite

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

Bit	Flag	Bit Name	Function	Read and write
b31~b16	Reserved	-	Functional reservation bit	R
b15~b0	DBUSPRT [15: 0]	D-BUS read protection function for 128Kbytes area	When DBUSPRT [15:0]=0x4450, Enable D-BUS read protection for the 0x0000_0000~0x0001_FFFFh area; otherwise, D-BUS read protection is invalid.	R

6.2.4 Initialize Configuration Register 4 (ICG4)

Reset value: Indefinite

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

Bit	Flag	Bit Name	Function			Read and write
			Boot source selection after system reset			
b31~b0	BTENB [31: 0]	Boot source configuration bit	MD pin	BTENB [31:0]	Boot source selector	
			0	*	Main Flash memory	R
			1	0x4844_0682	Main Flash memory	
			1	Other than the above	System Memory	

7 Embedded Flash (EFM)

7.1 Introduction

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

7.2 Main Features

- Two independent Flash memory form a dual bank, which can realize the BGO (BackGround Operation) function
- 134KBytes of OTP space
- ICODE bus 16Bytes prefetch instruction
- Two independent buffers: ICODE bus buffer space 2KBytes; DCODE bus buffer space 256Bytes
- Support boot swap function
- Support data security protection

7.3 Embedded Flash

Flash has the following main features:

- The capacity is up to 512KBytes , consisting of two 256KBytes Flash , a total of 64 sectors, each sector is 8Kbytes. Sector 0 to sector 15 in block 0 can be configured as an OTP area.
- The OTP (One Time Program) area is 134KBytes in total , of which 128KBytes are configured at block 0 address 0x0000_0000~0x0001_FFFF , and 6KBytes are configured at address 0x0300_0000~0x0300_17FF. Address 0x0300_1800~0x0300_1AD7 is the OTP data latch area.
- 128-bit wide data read and 128-bit wide read cache buffer accelerates code execution.
- The programming unit is 4Bytes, and the erasing unit is 8KBytes.

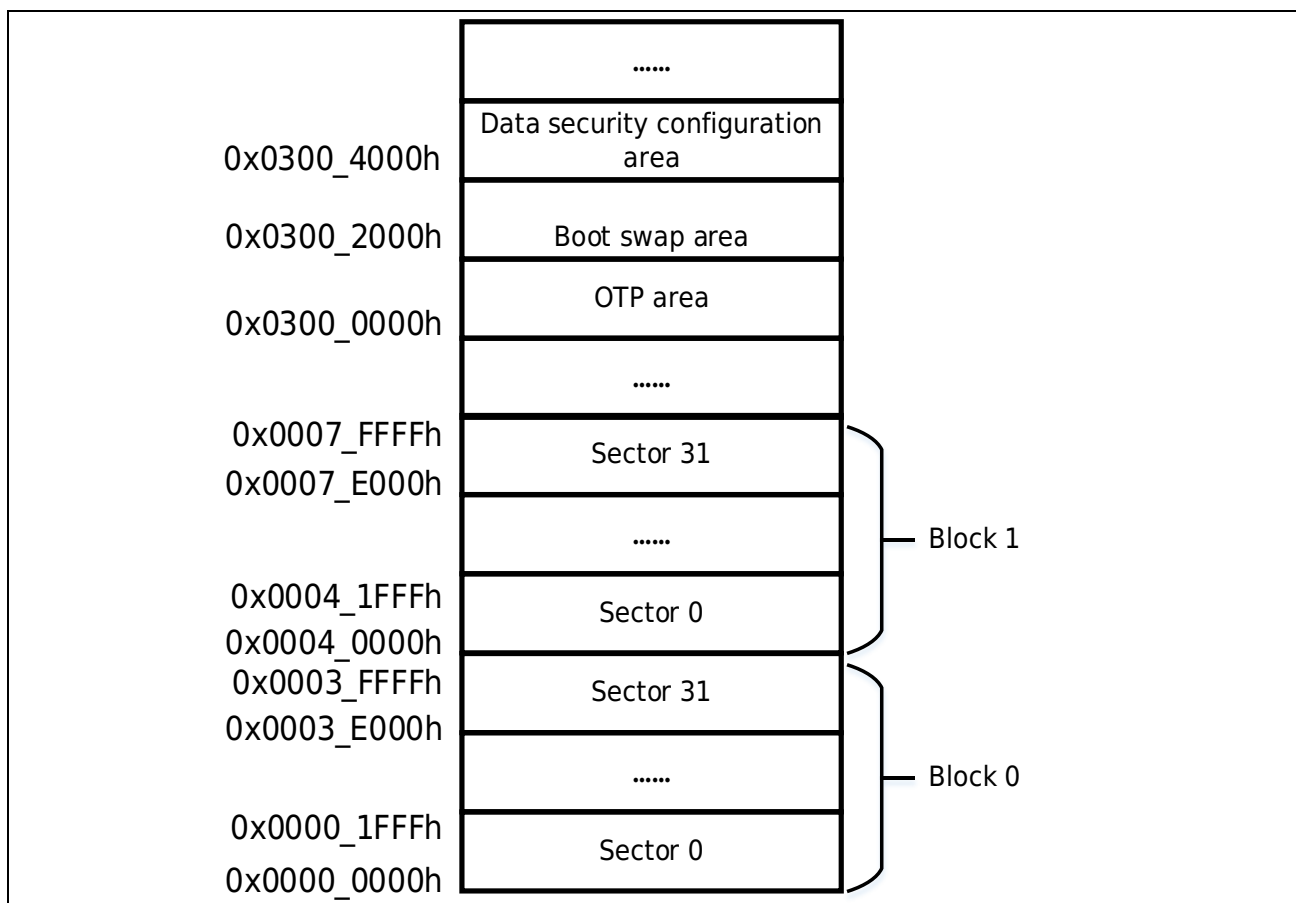


Figure 7-1 512kbytes Product Flash Address Distribution

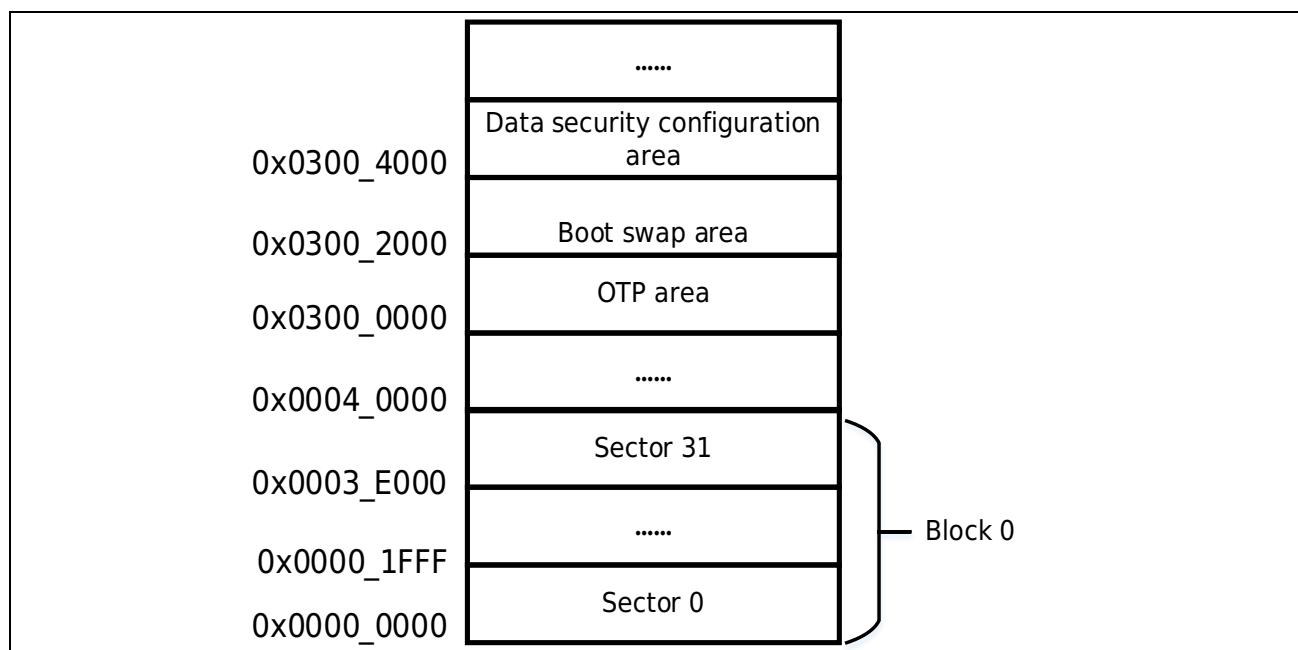


Figure 7-2 256kbytes Product Single Flash Address Distribution

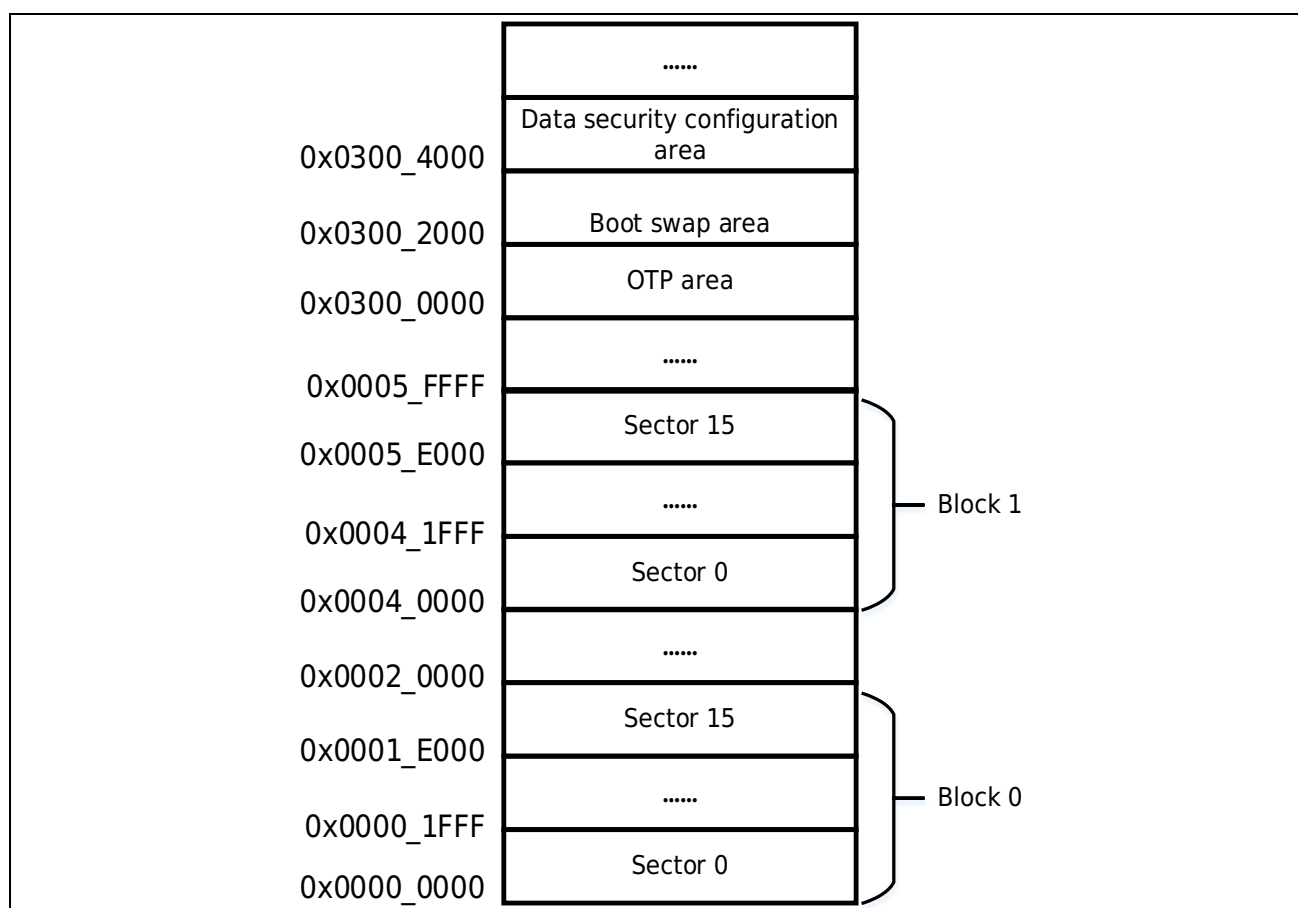


Figure 7-3 256kbytes Product Dual Flash Address Distribution

OTP area 0x0300_0000~0x0300_1ADB , the boot swap enable address 0x0300_2000~0x0300_2003 and the data security configuration area 0x0300_4000~0x0300_400B each have a separate sector, which can be programmed and erased, but full erase is prohibited;

these three sectors are physically subordinate to FLASH block 0 , and the erase flag follows FLASH block 0 , but the data of these three sectors remains unchanged if FLASH block 0 is fully 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 periods (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 period is 0. It is recommended that users modify the CPU main frequency and Flash read wait period bit according to the following steps. Please refer to the number of waiting periods Table 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 register EFM_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 register EFM_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
80MHz < Fhclk ≤ 120MHz	FLWT [3:0] = 4'b0010
40MHz < Fhclk ≤ 80MHz	FLWT [3:0] = 4'b0001
Fhclk ≤ 40MHz	FLWT [3:0] = 4'b0000

7.5 Flash Read Cache

Each Flash read operation is a 128-bit read. The data is sent to the CPU and stored in the buffer register at the same time. The 128-bit data can be 4 lines of 32-bit instructions or 8 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 period. 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 2.25Kbytes 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 (128 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 periods will change. For details, please refer to Table 7-2.

Table 7-2 Flash Actual Read Cycle Number

EFM_FRMC. FLWT [3:0] Set	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.

Flash programming, the last bit of the sector / full erase address must be aligned with 4 (the last bit of the address is: 0x0 , 0x4 , 0x8 , 0xC) , the programming unit is 4Bytes , the sector erase unit is 8Kbytes , and the full erase can be a single Flash block or two Flash blocks according to the register setting. 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.BUSHLCTL=0, and the bus will be occupied until the erasing is completed; if EFM_FWMC.BUSHLCTL=1, the bus will be released and the bus can continue to access another Flash address. 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 0x01234567 to EFM_KEY1 first , then write 0xFEDCBA98)

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_F0/1NWPRT. Each write protection register has a write protection lock bit (WLOCK1, WLOCK0). Once the write protection lock bit is set to 1, the write protection register bit can only be set from write enable 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.PRTWERR0/1=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 (EFM_FSR.PRTWERR0/1=1) occurs.

7.6.3 Single Programming Without Read BackFunction

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

- 1) Disable the Flash register write protection. (EFM_FAPRT first writes 0x0123, then 0x3210)
- 2) Unlock the EFM_KEY1 lock. (EFM_KEY1 first writes 0x01234567, then writes 0xFEDCBA98).
- 3) Set a single programming mode. (EFM_FWMC.PEMODE [2:0]=001)
- 4) Remove write protection. (The corresponding bit of EFM_F0/1NWPRT is set to 1)
- 5) Write 32-bit data to the programming address
- 6) Wait until the Flash is idle. (EFM_FSR.RDY0/1=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 end of programming flag. (EFM_FSR.OPTEND0/1)

A single programming no-return read/write operation is issued to the latched OTP address, programming fails, and the flag EFM_FSR.OTPERR0 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 EFM_FSR.MISMTCH0/1 to verify that the written data is correct.

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

- 1) Disable the Flash register write protection. (EFM_FAPRT first writes 0x0123, then 0x3210)
- 2) Unlock the EFM_KEY1 lock. (EFM_KEY1 first writes 0x01234567, then writes 0xFEDCBA98).
- 3) Set a single programming read-back mode. (EFM_FWMC.PEMODE [2:0]=010)
- 4) Remove write protection. (The corresponding bit of EFM_F0/1NWPRT is set to 1)
- 5) Write 32-bit data to the programming address
- 6) Wait until the Flash is idle. (EFM_FSR.RDY0/1=1)
- 7) Judge the programming self-read result flag bit. (EFM_FSR.MISMTCH0/1)
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 end of programming flag. (EFM_FSR.OPTEND0/1)
A single programming read/write operation is issued to the latched OTP address. If the programming fails, the flag EFM_FSR.OTPWERR0 is set.

7.6.5 Continuous Programming Function

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 16us. Continuous programming operations are set as follows:

- 1) Disable the Flash register write protection. (EFM_FAPRT first writes 0x0123, then 0x3210)
- 2) Unlock the EFM_KEY1 lock. (EFM_KEY1 first writes 0x01234567, then writes 0xFEDCBA98).
- 3) Set continuous programming mode. (EFM_FWMC.PEMOD [2:0]=011)
- 4) Remove write protection. (The corresponding bit of EFM_F0/1NWPRT is set to 1)
- 5) Write 32-bit data to the programming address. (The programming address needs to belong to a different Flash block from the execution program or the program needs to run outside the Flash area)
- 6) Wait for the operation end flag (EFM_FSR.OPTEND0/1) to be set.
- 7) Clear the operation end flag until EFM_FSR.OPTEND0/1 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]=000)
- 10) Wait until the Flash is idle. (EFM_FSR.RDY0/1=1)
- 11) Read the programming address and judge whether it is consistent with the written value.

- 12) 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.OTPWERR0 is set.

7.6.6 Erase Function

After the Flash is erased, the data at the address (8KBytes 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 the EFM_KEY1 lock. (EFM_KEY1 first writes 0x01234567, then writes 0xFEDCBA98).
- 3) Set the erase mode (EFM_FWMC.PEMOD [2: 0] = 100).
- 4) Remove write protection. (The corresponding bit of EFM_F0/1NWPRT is set to 1)
- 5) Write any 32-bit value to any address in the sector that needs to be erased (address to be aligned with 4).
- 6) Wait until the Flash is idle. (EFM_FSR.RDY0/1=1)
- 7) Clear the erase end flag bit. (EFM_FSR.OPTEND0/1)

Issue an erase operation to the latched OTP address. If the erase fails, the OTP area data is retained and the flag EFM_FSR.OTPWERR0 is set.

7.6.7 Full Erase Function

EFM provides two erasing modes: single-block Flash full erase and two-block Flash full erase at the same time. 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 the EFM_KEY1 lock. (EFM_KEY1 first writes 0x01234567, then writes 0xFEDCBA98).
- 3) When erasing a single Flash block, set EFM_FWMC.PEMOD [2:0]=10;
- 4) When both Flashes are fully erased at the same time, set EFM_FWMC.PEMOD [2:0]=110.
- 5) Remove write protection. (EFM_F0/1NWPRT all bits are set to 1)
- 6) 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).
- 7) Full erasure of two Flash blocks: Write any 32-bit value to Flash0 or Flash1 address (address must be aligned to 4). The hardware will automatically synchronize the full erase operation to another Flash.
- 8) Wait until the Flash is idle. (EFM_FSR.RDY0/1=1)
- 9) Clear the erase end flag bit. (EFM_FSR.OPTEND0/1)

OTP is enabled, a single-block full erase and write operation is issued to the Flash block 0 address. If the full erase fails, the Flash0 data is retained and the flag EFM_FSR.OTPWERR0

is set. Two full erase and write operations are issued to the Flash block 0 address. The Flash0 data is retained, the Flash1 data is fully erased, and the flag EFM_FSR.OTPWERR0 is set.

7.6.8 Data Security Protection

This product provides 4 protection levels 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

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

Protection level 1:

FLASH address 0x0000_0430-0x0000_0433 is programmed with data 0xAF180402 and the address 0x0300_4000~0x0300_400B is programmed with a 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 cannot be programmed or erased by the user program.
- The data at address 0x0300_4000~0x0300_400B 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 0xA85173AE , 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 cannot be programmed or erased by the user program.

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:

the FLASH addresses 0x0000_0420-0x0000_0423 , 0x0000_0424-0x0000_0427 , 0x0000_0428-0x0000_042B are programmed with data of 0x42545048 , 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 cannot be programmed or erased by the user program.

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

D-BUS read protection is provided for the address space 0x00000000~0x0001FFFF (128Kbytes). To enable D-BUS protection, set the ICG3 register DBUSPRT [15:0] to 0x4450. After reset and restart, the function is enabled. When the CPU-PC pointer is in this 128K space (that is, the CPU executes the program in this area), the CPU and DMAC can access it normally; when the CPU-PC pointer is outside this 128K space (that is, the CPU executes the program in the area outside 128K), the CPU and DMAC are prohibited from reading D-BUS within the 128K space and return a bus error. D-BUS is the data access bus of the CPU, including operand access, stack data access, general data access, etc. D-BUS read protection is enabled, users need to pay attention to the following matters.

- 1) A single independent program space cannot cross the 0x00020000 boundary.
- 2) When the CPU is executing a program in an area outside the 128K space, it cannot read the interrupt vector table configured in the 128K space.
- 3) When the CPU is executing a program in an area outside the 128K area, it can only return to the 128K area to execute the program through jump instructions and subroutine calls.

7.6.10 BGO Function

In Flash programming, the erase target address and the execution program address are not in the same Flash block, or the program is executed on RAM. You can set EFM_FWMC.BUSHLCTL to 1 (bus release during programming and erasing), so that Flash is programming or erasing while the program is executing. The following are the steps to set up the BGO function.

- 1) Disable the Flash register write protection. (EFM_FAPRT first writes 0x0123, then 0x3210)
- 2) Unlock the EFM_KEY1 lock. (EFM_KEY1 first writes 0x01234567, then writes 0xFEDCBA98).
- 3) Set programming and erasing modes. (EFM_FWMC.PEMODE [2:0]=001,010,011,100,101)
- 4) Remove write protection. (The corresponding bit of EFM_F0/1NWPRT is set to 1)
- 5) Set bus release during programming and erasing (EFM_FWMC.BUSHLCTL=1)
- 6) Write 32-bit data to the programming address. (The programming address and execution program belong to different blocks of Flash)
If the execution program is in RAM, two Flash memories can be programmed and (fully) erased at the same time.
- 7) Wait for the erase completion flag to be set. (EFM_FSR.OPTEND0/1=1)

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 interrupt EFM _ PEERR:

Position:

- Issue program, erase, and full erase operations (OTPWERR0=1) to the latched OTP address.
- After the OTP function is enabled, the 0x0300_0000~0x0300_1ADB space is erased, and a full erase operation is performed (OTPWERR0=1).
- Issue program, erase, and full erase operations to write-protected sectors (PRTWERR0/1=1).
- The programming address is not aligned to 4 or the data size is not 32 bits (PGSZERR0/1=1).
- In single programming read-back mode, the programming address hardware automatically reads the value that is inconsistent with the written value (MISMTCH0/1=1).

Reset:

Register EFM _ FSCLR corresponds to flag clear bit write 1 and status bit reset.

2. FLASH read and write conflict interrupt EFM _ COLERR:

Position:

- the Flash busy period (RDY0=0 or RDY1=0), read and write operations are performed on this Flash block. (Continuous programming mode, except for continuous writing to Flash)

Peset:

Register EFM _ FSCLR corresponds to purge position 1, status bit zeroed.

3. End of operation interrupt EFM _ OPTEND:

Position:

- Programming mode: Single address programming ends (OPTEND0/1=1).
- Erase mode: sector erase, full erase end (OPTEND0/1=1).

Peset:

Register EFM _ FSCLR corresponds to purge position 1, status bit zeroed.

7.7 One-Time Programmable Byte (OTP)

This MCU provides a maximum of 134KBytes of OTP area , divided into 16 8KBytes , 2 2KBytes , 4 256Bytes , 32 16Bytes , 128 4Bytes. 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_1FFF	8KB	0x0300_1800~0x0300_1803
1	0x0000_2000~0x0000_3FFF	8KB	0x0300_1804~0x0300_1807
-	-	-	-
-	-	-	-
-	-	-	-
14	0x0001_C000~0x0001_DFFF	8KB	0x0300_1838~0x0300_183B
15	0x0001_E000~0x0001_FFFF	8KB	0x0300_183C~0x0300_183F
16	0x0300_0000~0x0300_07FF	2KB	0x0300_1840~0x0300_1843
17	0x0300_0800~0x0300_0FFF	2KB	0x0300_1844~0x0300_1847
18	0x0300_1000~0x0300_10FF	256B	0x0300_1848~0x0300_184B
19	0x0300_1100~0x0300_11FF	256B	0x0300_184C~0x0300_184F
20	0x0300_1200~0x0300_12FF	256B	0x0300_1850~0x0300_1853
21	0x0300_1300~0x0300_13FF	256B	0x0300_1854~0x0300_1857
22	0x0300_1400~0x0300_140F	16B	0x0300_1858~0x0300_185B
-	-	-	-
-	-	-	-
-	-	-	-
53	0x0300_15F0~0x0300_15FF	16B	0x0300_18D4~0x0300_18D7
54	0x0300_1600~0x0300_1603	4B	0x0300_18D8~0x0300_18DB
-	-	-	-
-	-	-	-
-	-	-	-
181	0x0300_17FC~0x0300_17FF	4B	0x0300_1AD4~0x0300_1AD7

To enable the OTP function, you need to pre-program 32 bits of data containing 0 to the address 0x0300_1AD8~0x0300_1ADB (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. OTP sector 0~15 and Flash block 0 address 0x0000_0000~0x0001_FFFF share 128KBytes physical space; 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, and Flash block 0 cannot be fully erased. Programming the OTP latch address requires unlocking the EFM_KEY2 register (first write 0x10325476 to EFM_KEY2 , then write 0xEFCDAB89). 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) Disable the Flash register write protection. (EFM_FAPRT first writes 0x0123, then 0x3210)
- 2) Unlock the EFM_KEY1 lock. (EFM_KEY1 first writes 0x01234567, then writes 0xFEDCBA98).
- 3) Set programming mode. (EFM_FWMC.PEMODE [2:0]=001,010,011)
- 4) Write data to the OTP address as needed;
- 5) Unlock the EFM_KEY2 lock. (EFM_KEY2 first writes 0x10325476, then writes 0xEFCDAB89).
- 6) Write 0 to any bit of the 32-bit OTP enable address 0x0300_1AD8~0x0300_1ADB. (It is recommended to write all 32 bits 0)
- 7) Write 0 to any bit of the OTP latch address. (It is recommended to write all 32 bits 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_1AD8~0x0300_1ADB has been programmed with 0, step 6 needs to be omitted.

Programming, erasing and fully erasing the latched OTP data area address will generate the OTPWERR0 error flag bit.

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

Program or sector erase the Flash block 0 address (0x0300_2000), and then reset and restart to achieve boot swap. When the address data is 0x5A5A5A, the boot swap is valid, and when it is any other value, the boot swap is invalid. After boot swap is enabled, if the OTP function is not enabled (0x0300_1AD8~0x0300_1ADB are all 1), the entire addresses of Flash block 0 and block 1 are swapped; and the OTP enabled address cannot be programmed. If the OTP function is enabled (the data in 0x0300_1AD8~0x0300_1ADB are not all 1), the addresses of Flash block 0 and block 1 are partially swapped. The address transformation rules are shown in the figure Figure 7-4 and Figure 7-5 below.

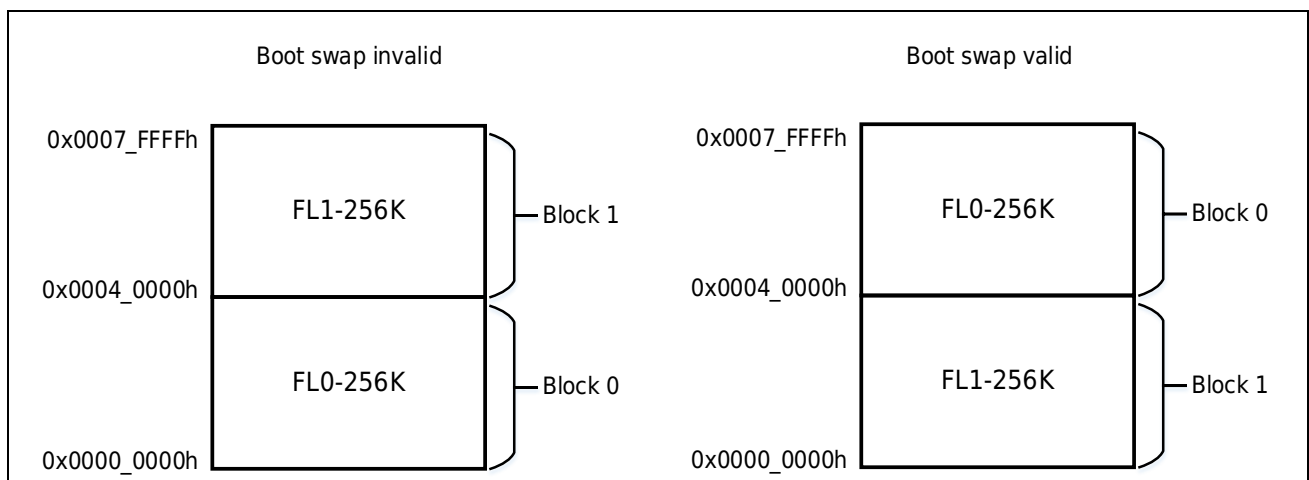


Figure 7-4 When The OTP Function Of 512kbytes Product Not Enabled, Flash Address Distribution

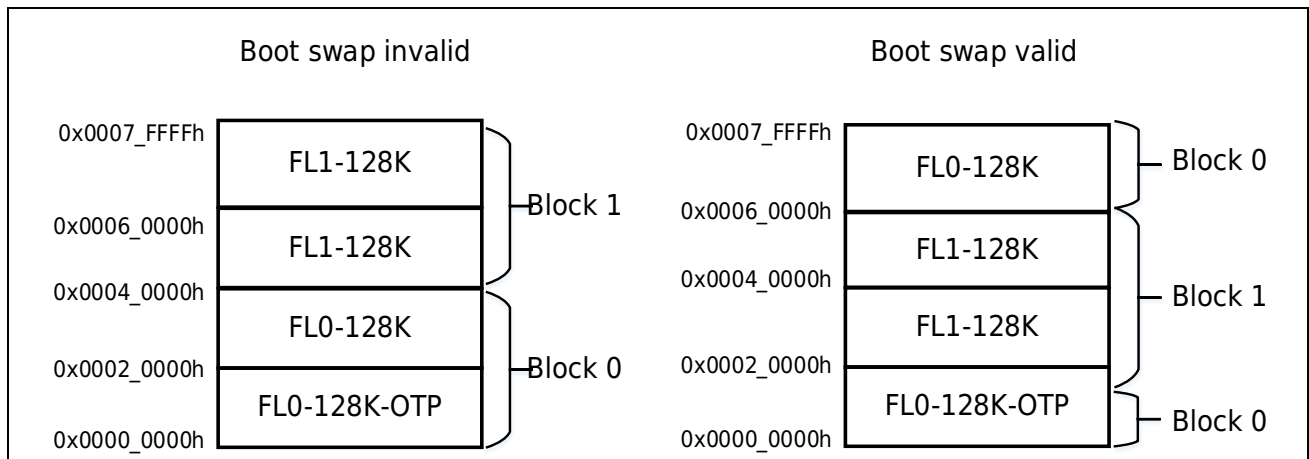


Figure 7-5 When The OTP Function Of 512kbytes Product Enabled, Flash Address Distribution

For Flash 256KBytes products, single FLASH and dual FLASH with OTP function enabled do not support boot swap. When dual FLASH and OTP function is not enabled, the address conversion rules for boot swap are as follows Figure 7-6.



Figure 7-6 256kbytes Product OTP Function Is Not Enabled, Flash Address Allocation

The following is an instance of using the boot swap function when OTP is not enabled for a 512KBytes product.

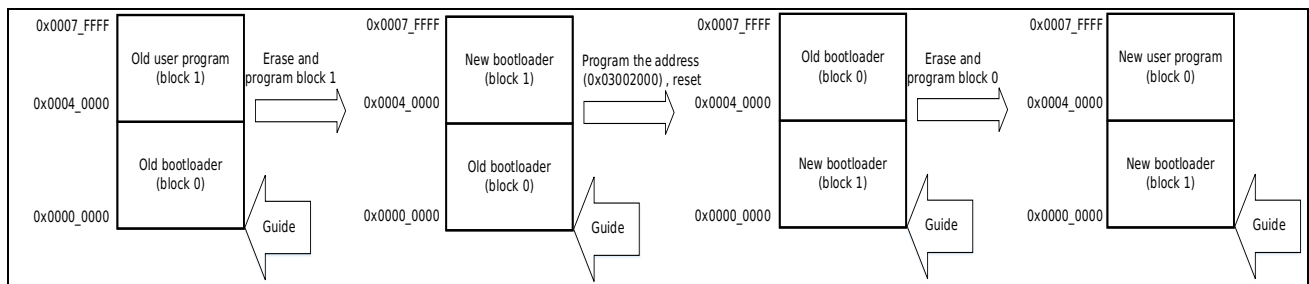


Figure 7-7 Start Boot Swap 1

When the user needs to upgrade the boot program again, since the address 0x0300_2000 has been programmed, the address 0x0300_2000 needs to be sector erased for the next upgrade. The operation process is as follows Figure 7-8.

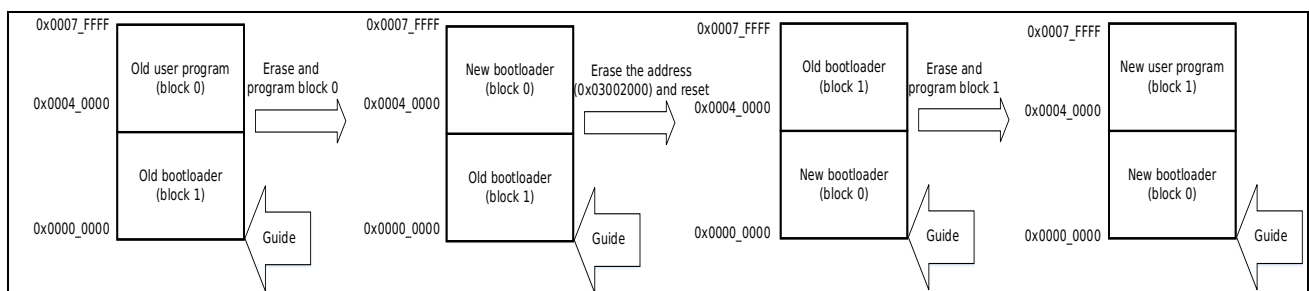


Figure 7-8 Start Boot Swap 2

7.9 Register Description

Table 7-4 Register Summary

EFM_BASE_ADDR: 0x40010400

Register Description	Register Name	Offset	Bit Width	Reset Value
Flash Access Write Protection Register	EFM_FAPRT	0x0000	32	0x00000000
Flash Key 1 Register	EFM_KEY1	0x0004	32	0x00000000
Flash Key 2 Register	EFM_KEY2	0x0008	32	0x00000000
Flash Stop Register	EFM_FSTP	0x0014	32	0x00000000
Flash Read Mode Register	EFM_FRMC	0x0018	32	0x00000000
Flash Erase Mode Register	EFM_FWMC	0x001C	32	0x00030000
Flash Status Register	EFM_FSR	0x0020	32	0x01000100
Flash Status Clear Register	EFM_FSCLR	0x0024	32	0x00000000
Flash Interrupt Enable Register	EFM_FITE	0x0028	32	0x00000000
Flash Boot Swap Status Register	EFM_FSWP	0x002C	32	Indefinite
Chip Specific Flag Register	EFM_CHIPID	0x0040	32	0x48440682
UNIQUE ID Register 0	FEM_UQID0	0x0050	32	Indefinite
UNIQUE ID Register 1	EFM_UQID1	0x0054	32	Indefinite
UNIQUE ID Register 2	EFM_UQID2	0x0058	32	Indefinite
Flash Write Protection Latch Register	EFM_WLOCK	0x0180	32	0x00000000
Flash0 Write Protection Register	EFM_F0NWPRT	0x0190	32	0x00000000
Flash1 Write Protection Register	EFM_F1NWPRT	0x01A0	32	0x00000000

7.9.1 Flash Access Write Protection Register (EFM_FAPRT)

Reset value: 0x00000000

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

Bit	Flag	Bit Name	Function	Read and write
b31-b16	Reserved	-	Read at "0", write at "0"	R/W
b15-b0	FAPRT [15:0]	EFMregister write protection	EFM Register Access 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 0x00000000. When EFMregister access protection is invalid, the register read value is 0x00000001.	R/W

7.9.2 Flash Key 1 Register (EFM_KEY1)

Reset value: 0x00000000

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	Flag	Bit Name	Function	Read and write
b31-b0	KEY1 [31:0]	Key 1 Register	EFM_FWMC Write Protection Register. First write 0x01234567 to EFM_KEY1 , then write 0xFEDCBA98 to release the write protection of EFM_FWMC. The register reads as 0x00000000.	R/W

7.9.3 Flash Key 2 Register (EFM_KEY2)

Reset value: 0x00000000

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	Flag		Bit Name		Function		Read and write								
b31-b0	KEY2 [31:0]		Key 2 Register		Write Protection Register for OTP Latch Address. First write 0x10325476 to EFM_KEY2 , then write 0xEFCDA89 to release the write protection of the OTP latch address. The register reads as 0x00000000.		R/W								

7.9.4 Flash Stop Register(EFM_FSTP)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	F1STP	F0STP
Bit	Flag		Bit Name		Function		Read and write								
b31-b2	Reserved		-		Read at "0", write at "0"		R/W								
b1	F1STP		Flash1 stop mode control		0: Flash1 active status 1: Flash1 is in stop mode When the register bit is set from 1 to 0 , please confirm that the FSR.RDY1 bit is 1 before accessing Flash1.		R/W								
b0	F0STP		Flash0 stop mode control		0: Flash0 active status 1: Flash0 is in stop mode When the register bit is set from 1 to 0 , please confirm that the FSR.RDY0 bit is 1 before accessing Flash0.		R/W								

7.9.5 Flash Read Mode Register (EFM_FRMC)

Reset value: 0x00000000

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

Bit	Flag	Bit Name	Function	Read and write
b31~b20	Reserved	-	Read at "0", write at "0"	R/W
b19	CRST	Cache Reset	0: Cache Data Reset 1: Cache data reset (buffer, cache, prefetch)	R/W
b18	PREFETE	ICODE Prefetch Permission Bit	0: Disable ICODE prefetch function 1: ICODE prefetch Function Enable	R/W
b17	DCACHE	DCODE Cache Permission Bits	0: Disable DCODE Cache Function 1: DCODE Cache Function Enable	R/W
b16	ICACHE	ICODE Cache Permission Bits	0: Disable ICODE Cache Function 1: ICODE Cache Function Enable	R/W
b15~b9	Reserved	-	Read at "0", write at "0"	R/W
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"	R/W
b7-b4	Reserved	-	Read at "0", write at "0"	R/W
b3~b0	FLWT [3:0]	Flash read insertion wait period	0000b: Do not insert a read wait period 0001b: Insert 1 wait period 0010b: Insert 2 wait period 1110b: Insert 14 wait period 1111b: Insert 15 wait period	R/W

7.9.6 Flash Erase Mode Register (EFM_FWMC)

Reset value: 0x00030000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	KEY2 LOCK	KEY1 LOCK
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	BUSH LDCTL	-	-	-	-	-	-	PEMOD [2:0]	

Bit	Flag	Bit Name	Function	Read and write
b31~b18	Reserved	-	Read at "0", write at "0"	R/W
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.	R/W
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.	R/W
b15~b9	Reserved	-	Read at "0", write at "0"	R/W
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.	R/W
b7~b3	Reserved	-	Read at "0", write at "0"	R/W
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 : Two Flash full erasure mode 111: Read-only mode	R/W

7.9.7 Flash Status Register (EFM_FSR)

Reset value: 0x01000100

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	RDY1	-	-	COLERR1	OPTEND1	MISM TCH1	PGSZ ERR1	PRTW ERR1	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	RDY0	-	-	COLERR0	OPTEND0	MISM TCH0	PGSZ ERR0	PRTW ERR0	OTPW ER0

Bit	Flag	Bit Name	Function	Read and write
b31~b25	Reserved	-	Read at "0", write at "0"	R/W
b24	RDY1	Flash block 1 busy / idle status	0: Flash block 1 busy state 1: Flash block 1 idle state	R
b23~b22	Reserved	-	Read at "0", write at "0"	R/W
b21	COLERR1	Flash block 1 read/write conflict error flag bit	Setting conditions: When RDY1 is not set, read and write operations are performed on Flash1. In continuous programming mode, Flash1 is not set. Clearing Conditions: Write 1 to the clear bit of EFM_FSCLR.	R
b20	OPTEND1	Flash block 1 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
b19	MISM TCH1	Flash block 1 space single programming readback value inconsistent flag bit	Setting conditions: In single programming read-back mode, the read data and written value after Flash1 programming are inconsistent. Clearing Conditions: EFM_FSCLR corresponding clear bit write 1	R
b18	PGSZERR1	Flash block 1 space programming address and size misalignment flag bit	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
b17	PRTWERR1	Flash block 1 write protection error flag bit	Setting conditions: Program, erase, and fully erase write-protected sectors. Clearing Conditions: EFM_FSCLR corresponding clear bit write 1	R
b16~b9	Reserved	-	Read at "0", write at "0"	R/W
b8	RDY0	Flash block 0 busy / idle status	0: Flash block 0 busy 1: Flash block 0 idle state	R
b7~b6	Reserved	-	Read at "0", write at "0"	R/W
b5	COLERR0	Flash block 0 read/write conflict error flag bit	Setting conditions: When RDY0 is not set, read and write operations are performed on Flash0. In continuous programming mode, Flash0 is not set. Clearing Conditions: Write 1 to the clear bit of EFM_FSCLR.	R
b4	OPTEND0	Flash block 0 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	MISM TCH0	Flash block 0 space single programming readback value inconsistent flag bit	Setting conditions: In single programming read-back mode, the read data and written value	R

			after Flash0 programming are inconsistent. Clearing Conditions: EFM_FSCLR corresponding clear bit write 1	
b2	PGSZERR0	Flash block 0 programming address and size misalignment flag bit	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	PRTWERR0	Flash block 0 write protection address error flag bit	Setting conditions: Program, erase, and fully erase write-protected sectors. Clearing Conditions: EFM_FSCLR corresponding clear bit write 1	R
b0	OTPWERR0	Flash block 0 OTP erase error flag bit	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: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	COLERRCL R1	OPTENDCL R1	MISM TCHC LR1	PGSZ ERRCL R1	PRTW ERRCL R1	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	COLERRCL R0	OPTENDCL R0	MISM TCHC LR0	PGSZ ERRCL R0	PRTW ERRCL R0	OTPWERRCL R0

Bit	Flag	Bit Name	Function	Read and write
b31~b22	Reserved	-	Read at "0", write at "0"	R/W
b21	COLERRCLR1	Clear read and write conflict error flag bit	0: No purge action 1: Clear FSR.COLERR1 bit When read, this bit is always 0.	R/W
b20	OPTENDCLR1	End-of-purge flag	0: No purge action 1: Clear FSR.OPTEND1 bit When read, this bit is always 0.	R/W
b19	MISM TCHCLR1	Clear programming read-back mismatch flag bit	0: No purge action 1: Clear FSR.MISM TCH1 bit When read, this bit is always 0.	R/W
b18	PGSZERRCLR1	Clear programming address and size misalignment flag	0: No purge action 1: Clear FSR.PGSZERR1 bit When read, this bit is always 0.	R/W
b17	PEPRTERRCLR1	Clear the write protection error flag	0: No purge action 1: Clear FSR.PRTWERR1 bit When read, this bit is always 0.	R/W
b16~b6	Reserved	-	Read at "0", write at "0"	R/W
b5	COLERRCLR0	Clear read and write conflict error flag bit	0: No purge action 1: Clear FSR.COLERR0 When read, this bit is always 0.	R/W
b4	OPTENDCLR0	End-of-purge flag	0: No purge action 1: Clear FSR.OPTEND0 bit When read, this bit is always 0.	R/W
b3	MISM TCHCLR0	Clear programming read-back mismatch flag bit	0: No purge action 1: Clear FSR.MISM TCH0 bit When read, this bit is always 0.	R/W
b2	PGSZERRCLR0	Clear programming address and size misalignment flag	0: No purge action 1: Clear FSR.PGSZERR0 bit When read, this bit is always 0.	R/W
b1	PEPRTERRCLR0	Clear the write protection error flag	0: No purge action 1: Clear FSR.PRTWERR0 bit When read, this bit is always 0.	R/W
b0	OTPWERRCLR0	Clear the OTP erase error flag	0: No purge action 1: Clear FSR.OTPWERR0 bit When read, this bit is always 0.	R/W

7.9.9 Flash Interrupt Enable Register (EFM_FITE)

Reset value: 0x00000000

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

Bit	Flag	Bit Name	Function	Read and write
b31~b3	Reserved	-	Read at "0", write at "0"	R/W
b2	COLERRITE	Read-write conflict error interrupt disable	0: Read and write conflict error interrupt disable 1: Read and write conflict error interrupt enable	R/W
b1	OPTENDITE	End of operation interrupt enable	0: End of operation interrupt disable 1: End of operation interrupt enable	R/W
b0	PEERRITE	Programming/Erase Error Interrupt License	0: Programming/Erase Error Interrupt Not Permitted 1: Programming/Erase Error Interrupt License	R/W

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

Bit	Flag	Bit Name	Function	Read and write
b31-b1	Reserved	-	Read at "0", write at "0"	R/W
b0	FSWP	Sector 0 and Sector 1 Address Swap Bits	0: After reset, the CPU boots from Flash block 0. 1: After reset, the CPU boots from Flash block 1. The initial value of the register is determined by the value of Flash block 0 address 0x03002000~0x03002003. When the data is 0x5A5A5A, the initial value is 1, and the rest of the values are 0.	R

7.9.11 Chip Specific Flag Register (EFM_CHIPID)

Reset value: 0x48440682

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	Flag		Bit Name		Function		Read and write								
b31-b0	CHIPID [31:0]		Chip Specific Flag		CHIPID [31:16] : ASCII code corresponding to HD CHIPID [15:0] : Product model name Register Read-only.		R								

7.9.12 UNIQUE ID Register (EFM_UQID 0)

Reset value: Indefinite

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
UQID0 [31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
UQID0 [15:0]															
Bit	Flag		Bit Name		Function		Read and write								
b31-b0	UQID0 [31:0]		Unique code		Chip unique code		R								

7.9.13 UNIQUE ID Register (EFM_UQID 1)

Reset value: Indefinite

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
UQID1 [31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
UQID1 [15:0]															
Bit	Flag		Bit Name		Function		Read and write								
b31-b0	UQID1 [31:0]		Unique code		Chip unique code		R								

7.9.14 UNIQUE ID Register (EFM_UQID2)

Reset value: Indefinite

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
UQID2 [31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
UQID2 [15:0]															
Bit	Flag		Bit Name		Function		Read and write								
b31-b0	UQID2 [31:0]		Unique code		Chip unique code		R								

7.9.15 Flash Write Protection Lock Register (EFM_WLOCK)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	WLO CK1	-	-	-	WLO CK0
Bit	Flag		Bit Name		Function		Read and write								
b31-b5	Reserved		-		Read at "0", write at "0"		R/W								
b4	WLOCK1		F1NWPRT Register Lock Bit		0: The sector write prohibition controlled by the F1NWPRT register is not locked up 1: Sector write disable lockup controlled by F1NWPRT register Once this bit is set to 1, it can only be restored to 0 by resetting it.		R/W								
b3-b2	Reserved		-		Read at "0", write at "0"		R/W								
b0	WLOCK0		F0NWPRT Register Lock Bit		0: The sector write prohibition controlled by the F0NWPRT register is not locked up 1: Sector write disable lockup controlled by F0NWPRT register Once this bit is set to 1, it can only be restored to 0 by resetting it.		R/W								

7.9.16 Flash0 Write Protection Register (EFM_FONWPRT)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
FON WPRT 31	FON WPRT 30	FON WPRT 29	FOWP RT28	FON WPRT 27	FONW PRT2 6	FON WPRT 25	FON WPRT 24	FON WPRT 23	FON WPRT 22	FON WPRT 21	FON WPRT 20	FON WPRT 19	FON WPRT 18	FON WPRT 17	FON WPRT 16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
FON WPRT 15	FON WPRT 14	FON WPRT 13	FON WPRT 12	FON WPRT 11	FONW PRT1 0	FON WPRT 9	FON WPRT 8	FON WPRT 7	FON WPRT 6	FON WPRT 5	FON WPRT 4	FON WPRT 3	FON WPRT 2	FON WPRT 1	FON WPRT 0

Bit	Flag	Bit Name	Function	Read and write
b31	FONWPRT31	Flash block 0 sector 31 write permission bit	0: Flash block 0 sector 31 write disabled 1: Flash block 0 sector 0 write enable WLOCK0=0, this bit is allowed to write 0 or 1. WLOCK0=1, this bit can only be 0, and writing 1 is invalid.	R/W
b30	FONWPRT30	Write protection bit of Flash block 0 sector 30	0: Flash block 0 sector 30 write disabled 1: Flash block 0 sector 0 write enable WLOCK0=0, this bit is allowed to write 0 or 1. WLOCK0=1, this bit can only be 0, and writing 1 is invalid.	R/W
b29~b2	-	-	-	R/W
b1	FONWPRT1	Write protection bit of Flash block 0 sector 1	0: Flash block 0 sector 1 write disabled 1: Flash block 0 sector 0 write enable WLOCK0=0, this bit is allowed to write 0 or 1. WLOCK0=1, this bit can only be 0, and writing 1 is invalid.	R/W
b0	FONWPRT0	Write protection bit of Flash block 0 sector 0	0: Flash block 0 sector 0 write disabled 1: Flash block 0 sector 0 write permission WLOCK0=0, this bit is allowed to write 0 or 1. WLOCK0=1, this bit can only be 0, and writing 1 is invalid.	R/W

7.9.17 Flash1 Write Protection Register (EFM_F1NWPRT)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
F1N WPRT 31	F1N WPRT 30	F1N WPRT 29	F1N WPRT 28	F1N WPRT 27	F1NW PRT2 6	F1N WPRT 25	F1N WPRT 24	F1N WPRT 23	F1N WPRT 22	F1N WPRT 21	F1N WPRT 20	F1N WPRT 19	F1N WPRT 18	F1N WPRT 17	F1N WPRT 16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
F1N WPRT 15	F1N WPRT 14	F1N WPRT 13	F1N WPRT 12	F1N WPRT 11	F1NW PRT1 0	F1N WPRT 9	F1N WPRT 8	F1N WPRT 7	F1N WPRT 6	F1N WPRT 5	F1N WPRT 4	F1N WPRT 3	F1N WPRT 2	F1N WPRT 1	F1N WPRT 0

Bit	Flag	Bit Name	Function	Read and write
b31	F1NWPRT31	Write protection bit of Flash block 1 sector 31	0: Flash block 1 sector 31 write disabled 1: Flash block1 sector 31 write permission WLOCK1=0, this bit is allowed to write 0 or 1. WLOCK1=1, this bit can only be 0, and writing 1 is invalid.	R/W
b30	F1NWPRT30	Write protection bit of Flash block 1 sector 30	0: Flash block 1 sector 30 write disable 1: Flash block 1 sector 30 write permission. WLOCK1=0, this bit is allowed to write 0 or 1. WLOCK1=1, this bit can only be 0, and writing 1 is invalid.	R/W
b29~b2	-	-	-	R/W
b1	F1NWPRT1	Write protection bit of Flash block 1 sector 1	0: Flash block 1 sector 1 write disabled 1: Flash block 1 sector 1 write permission WLOCK1=0, this bit is allowed to write 0 or 1. WLOCK1=1, this bit can only be 0, and writing 1 is invalid.	R/W
b0	F1NWPRT0	Write protection bit of Flash block 1 sector 0	0: Flash block 1 sector 0 write disabled 1: Flash block1 sector 0 write enable WLOCK1=0, this bit is allowed to write 0 or 1. WLOCK1=1, this bit can only be 0, and writing 1 is invalid.	R/W

7.10 Precautions

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 interrupt disable to enter low-power mode (sleep mode, stop mode, power-off mode).
5. When using read-acceleration caching, ensure that the cache RAM does not lose power. (PWC_PRAMLPC.PRAMPDC2=0)

8 Built-in SRAM (SRAM)

8.1 Introduction

This product has 4KB power-down mode retention SRAM (Ret_SRAM) and 64KB system SRAM (SRAM0/SRAMH).

Each SRAM can be accessed as a byte, half-word (16 bits), or full-word (32 bits). All SRAM (SRAMH/ SRAM0/ Ret SRAM) read and write operations can be performed at the fastest CPU speed (120 MHz).

Ret_SRAM can provide 4KB data holding space in Power down mode.

SRAM0 and Ret_SRAM have ECC (Error Checking and Correcting). ECC is a one-correction-two-checking code, which can correct one-bit error and check two-bit error. SRAMH has an Even-parity check (Even-parity check) with one parity bit per byte of data. See the table Table 8-1 for detailed definition of SRAM.

Table 8-1 SRAM Space Allocation

Name	Capacity	Address Range	Check Mode
SRAM0	32KB	0x20000000~0x20007FFF	ECC
SRAMH	32KB	0x1FFF8000~0x1FFFFFFF	Even-parity check
Ret_SRAM	4KB	0x200F0000~0x200F0FFF	ECC

Note:

- In the case where RAM parity errors are allowed to generate NMI interrupts and resets, when accessing data, the RAM space used must be initialized in word units; when executing instructions from the RAMH space, the RAM space used must be +3 The word area is initialized in units of words.
- In the case of allowing RAM ECC verification errors to generate NMI interrupts and resets, when accessing data, the used RAM space must be initialized in word units.

The ECC check of the system SRAM has an error injection function. 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.

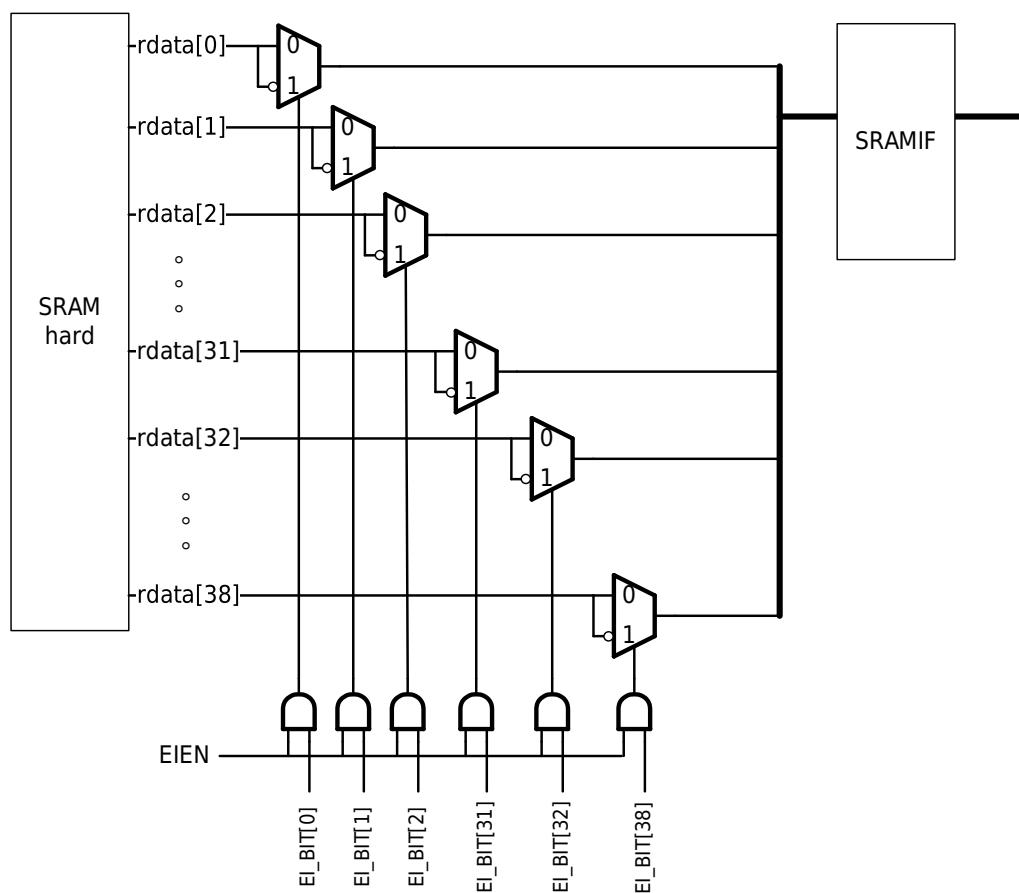


Figure 8-1 Schematic Diagram of Error Injection Function

8.2 Register Description

Table 8-2 List of SRAM Registers

Register Name	Initial Address	Reset Value
SRAM check control register (SRAM_CKCR)	0x40050808	0x00000000
SRAM Check Control Protection Register (SRAMC_CKPR)	0x4005080C	0x00000000
SRAM Check Status Register (SRAM_CKSR)	0x40050810	0x00000000
SRAM0 Error Injection Enable Register (SRAM0_EIEN)	0x40050814	0x00000000
SRAM0 Error Injection Bit Enable Register 0 (SRAM0_EIBIT0)	0x40050818	0x00000000
SRAM0 Error Injection Bit Enable Register 1 (SRAM0_EIBIT1)	0x4005081C	0x00000000
SRAM0 Check Error Address Register (SRAM0_ECCERRADDR)	0x40050820	0x00000000
Ret_SRAM Error Injection Enable Register (SRAMB_EIEN) (SRAM_CKSR)	0x40050824	0x00000000
Ret_SRAM Error Injection Bit Enable Register 0 (SRAMB_EIBIT0)	0x40050828	0x00000000
Ret_SRAM Error Injection Bit Enable Register 1 (SRAMB_EIBIT1)	0x4005082C	0x00000000
Ret_SRAM Check Error Address Register (SRAMB_ECCERRADDR)	0x40050830	0x00000000

8.2.1 SRAM Check Control Register (SRAM_CKCR)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Rev				SRAMB ECCMOD [1:0]		SRAM0ECCMOD [1:0]		Rev				SRAMBEC OAD		SRAM0ECC OAD	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Rev														PYOAD	

Bit	Flag	Bit Name	Function	Read and write
b31~b28	Reserved	-	Read as "0", write as "0"	R/W
b27~b26	SRAMBEC MOD [1:0]	ECC check enable bit of Ret_SRAM	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.	R/W
b25~b24	SRAM0ECC MOD [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.	R/W
b23~b18	Reserved	-	Read as "0", write as "0"	R/W
b17	SRAMB ECCOAD	Ret_SRAM ECC check error operation	0: Non-maskable interrupt 1: Reset	R/W
b16	SRAM0ECCOAD	Operation after SRAM0 ECC Check Error	0: Non-maskable interrupt 1: Reset	R/W
b15~b1	Reserved	-	Read as "0", write as "0"	R/W
b0	PYOAD	Parity Action after error	0: Non-maskable interrupt 1: Reset	R/W

Note:

- In the case where RAM parity errors are allowed to generate NMI interrupts and resets, when accessing data, the RAM space used must be initialized in word units; when executing instructions from the RAMH space, the RAM space used must be +3 The word area is initialized in units of words.
- In the case of allowing RAM ECC verification errors to generate NMI interrupts and resets, when accessing data, the used RAM space must be initialized in word units.

8.2.2 SRAM Check Control Protection Register (SRAMC_CKPR)

Reset value: 0x00000000

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								CKPRKW [6:0]						CKPRC	
Bit	Flag	Bit Name				Function								Read and write	
b31~b8	Reserved	-				Read as "0", write as "0"								R/W	
b7~b1	CKPRKW [6:0]	Write key				When writing to the current register, you need to write "3b" in these bits to enable the current register								R/W	
b0	CKPRC	SRAM parity control register write enable				0: SRAM verification control register writing disabled 1: SRAM verification control register write enable								R/W	

CKPRC: Controls the writing of the SRAMCKCR register. When CKPRC is set to 1, the write operation to SRAMCKCR is allowed. If it is set to 0, the write operation to SRAMCKCR cannot be performed. When writing this bit, 3bh must be written to CKPRKW [6:0] at the same time.

8.2.3 SRAM Check Status Register (SRAM_CKSR)

Reset value: 0x00000000

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							CACHERAM_ PYERR	SRAMB_ 2ERR	SRAMB_ 1ERR	SRAM0_ 2ERR	SRAM0_ 1ERR	SRAMH_ PYERR	Rev		
Bit		Flag		Bit Name				Function				Read and write			
b31~b9		Reserved		-				Read as "0", write as "0"				R/W			
b8		CACHERAM_PYERR		CACHERAM parity check error flag				0: No parity error occurs 1: A parity error occurs				R/W (Note 1)			
b7		SRAMB_2ERR		Ret_SRAM ECC 2-bit error flag bit				0: No 2-bit ECC error occurs 1: 2-bit ECC error occurs				R/W (Note 1)			
b6		SRAMB_1ERR		Ret_SRAM ECC 1-bit error flag bit				0: No 1-bit ECC error occurs 1: 1-bit ECC error occurs				R/W (Note 1)			
b5		SRAM0_2ERR		SRAM0 occurred ECC 2-bit error flag				0: No 2-bit ECC error occurs 1: 2-bit ECC error occurs				R/W (Note 1)			
b4		SRAM0_1ERR		SRAM0 occurred ECC 1-bit error flag				0: No 1-bit ECC error occurs 1: 1-bit ECC error occurs				R/W (Note 1)			
b3		SRAMH_PYERR		SRAMH Parity Parity error flag				0: No parity error occurs 1: A parity error occurs				R/W (Note 1)			
b2~b0		Reserved		-				Read as "0", write as "0"				R/W			

Note 1: Writing 1 clears 0.

8.2.4 SRAM0 Error Injection Enable Register (SRAM0_EIEN)

Reset value: 0x00000000

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	Flag	Bit Name	Function	Read and write
b31~b1	Reserved	-	Read as "0", write as "0"	R/W
b0	EIEN	Fault injection enable	0: Fault injection disabled 1: Error injection invalid	R/W

8.2.5 SRAM0 Error Injection Bit Enable Register 0 (SRAM0_EIBIT0)

Reset value: 0x00000000

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	Flag	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	R/W

8.2.6 SRAM0 Error Injection Bit Enable Register 1 (SRAM0_EIBIT1)

Reset value: 0x00000000

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								EIBIT [38:32]							

Bit	Flag	Bit Name	Function	Read and write
b31~b7	Reserved	-	Read as "0", write as "0"	R/W
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	R/W

8.2.7 SRAM0 Check Error Address Register (SRAM0_ECCERRADDR)

Reset value: 0x00000000

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	ECCERRADDR [14:0]														

Bit	Flag	Bit Name	Function	Read and write
b31~b15	Reserved	-	Read as "0", write as "0"	R/W
b14~b0	ECCERRADDR [14:0]	Checksum error address	Record the access address when a 1-bit or 2-bit ECC error occurs in SRAM0	R

8.2.8 Ret_SRAM Error Injection Enable Register (SRAMB_EIEN)

Reset value: 0x00000000

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	Flag	Bit Name	Function	Read and write
b31~b1	Reserved	-	Read as "0", write as "0"	R/W
b0	EIEN	Fault injection enable	0: Fault injection disabled 1: Error injection invalid	R/W

8.2.9 Ret_SRAM Error Injection Bit Enable Register 0 (SRAMB_EIBIT0)

Reset value: 0x00000000

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	Flag	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	R/W

8.2.10 Ret_SRAM Error Injection Bit Enable Register 1 (SRAMB_EIBIT1)

Reset value: 0x00000000

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									EIBIT [38:32]						
Bit	Flag		Bit Name				Function					Read and write			
b31~b7	Reserved		-				Read as "0", write as "0"					R/W			
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					R/W			

8.2.11 Ret_SRAM Check Error Address Register (SRAMB_ECCERRADDR)

Reset value: 0x00000000

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				ECCERRADDR [11:0]											
Bit	Flag	Bit Name		Function		Read and write									
b31~b12	Reserved	-		Read as "0", write as "0"		R/W									
b11~b0	ECCERRADDR [11:0]	Checksum error address		Record the access address when a 1-bit or 2-bit ECC error occurs in Ret_SRAM		R									

9 General IO (GPIO)

Some of the abbreviations used in this chapter are:

- Px (x=A~F) represents a group of ports, such as PA represents the 16 I/O ports of the group PA0~PA15.
- Pxy (x= A~F, y=0~15 , the same below) represents a single I/O port, such as PB10 port represents the 11th I/O in the PB group.
- General Purpose Input Output (GPIO).
- NOD/POD (Noss/Pmos Open Drain) NMOS/PMOS 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 64 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 Port Function Summary

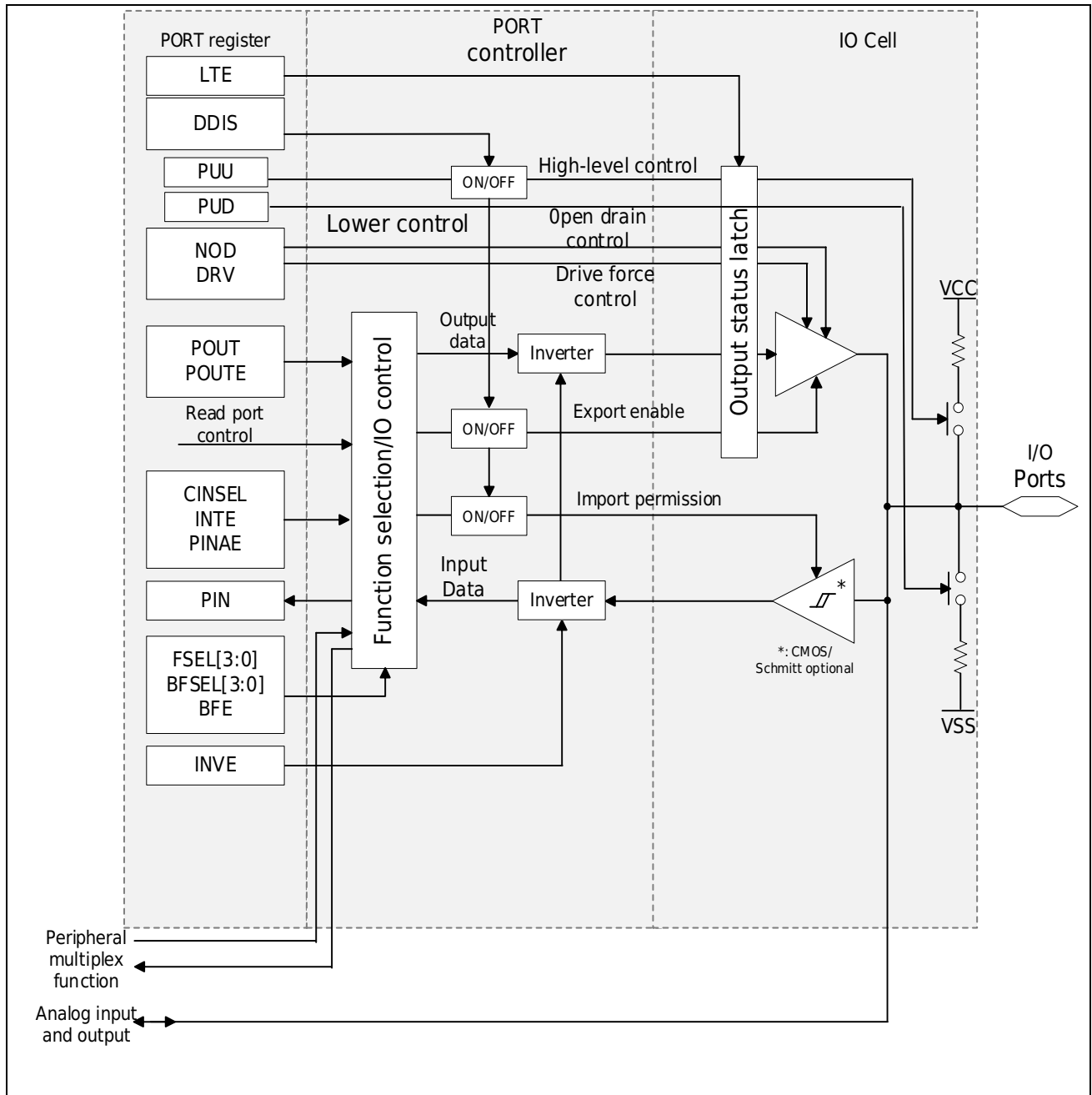


Figure 9-1 Port Basic Structure Diagram

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

9.3 Action Description

9.3.1 General Purpose Input/Output GPIO Function

General Input Function GPI:

Each I/O has a general 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.PIN [y] register bit is equivalent to the PCRxy.PIN bit.

By default, I/O input MOS is turned off to reduce power consumption. Only when PIDRx is read, PCRxyregister is opened. If needed, you can also set the register PCRxy.PINAE to 1 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 period 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 periods. 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 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.POUTE register bit is equivalent to POERx.POUTE [y]. The output value of I/O can be controlled by setting PCRxy.POUT, where the PCRxy.POUT register bit is equivalent to PODRx.POUT [y]. 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).

9.3.2 Peripheral Functions

Through the FSEL [5:0] of the function selection register PFSRxy, each port can configure up to 64 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], 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.

9.3.3 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.

9.3.4 Event Port Input and Output Functions

Support 4 groups of Event Ports, each with 16 ports. Event Port1 includes EVNTP100~EVNTP115, Event Port2 includes EVNTP200~EVNTP215, and so on. EVNTPmn (m=1~4, n=0~15) The port can be used as a trigger source to trigger other peripheral devices (such as TIMER, ADC, DMA, etc.) to start specific actions according to the event generated by the port input. It can also be used as a triggered object to accept events and automatically input or output.

As a trigger source, set PEVNTRISRm, PEVNTFALRm, PEVNTNFCR to select rising edge or falling edge detection, and digital filter function, and set the function selection register PFSRxy to select EVNTPmn function. When the selected edge is entered from the port, the event EVENT_PORTm is generated and output to other peripheral devices to trigger its start action.

As the triggered object, set PEVNTTRGSRm 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 the function clock control 0 register (PWC_FCG0) to valid.

9.3.5 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 interrupt EIRQ, 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.

9.3.6 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.

9.3.7 General Control

1. Pull Up/Pull Down Resistor

Each I/O port has an internal pull-up resistor. You can set the register PCRxy.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.

PA11/USBFS_DM, PA12/USBFS_DP pins are multiplexed with the USB port. Both pins have built-in pull-down resistors of about 400KΩ and are always valid.

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. The register can be written as 2'b11 and can be read as 2'b11, but in fact the pull-up and pull-down are not effective at the same time.

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

9.4 Register Description

Table 9-1 PORT Register Overview 1

BASE_ADDR: 0x40053800

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

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

*2: 32K secondary oscillator multiplex port PCRC14, the reset value of PCRC15 is 0x8100.

Table 9-2 PORT Register Overview 2

BASE_ADDR: 0x40010800

Register name	Symbol	Offset Address	Bit Width	Reset Value
Event port Direction Selection Register	PEVNTDIRRm	0x100+0x1C*(m-1)	32	0x00000000
Event Port input data register	PEVNTIDRm	0x104+0x1C*(m-1)	32	0x00000000
Event port output data register	PEVNTODRm	0x108+0x1C*(m-1)	32	0x00000000
Event port output data reset register	PEVNTORRm	0x10C+0x1C*(m-1)	32	0x00000000
Event port output data setting register	PEVNTOSRm	0x110+0x1C*(m-1)	32	0x00000000
Event port rising edge input permission register	PEVNTRISRm	0x114+0x1C*(m-1)	32	0x00000000
Event port falling edge input permission register	PEVNTFALRm	0x118+0x1C*(m-1)	32	0x00000000
Event port input filter control register	PEVNTNFCR	0x170	32	0x00000000

Note: m=1~4

9.4.1 General Input Data Register (PIDRx)

Reset value: 0xXXXX

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PIN [15:0]															

Bit	Flag	Bit Name	Function	Read and write
b15~b0	PIN [15:0]	Input status	0: I/O port input status is low 1: I/O port input status is high	R

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.

9.4.2 Universal Output Data Register (PODRx)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
POUT [15:0]															

Bit	Flag	Bit Name	Function	Read and write
b15~b0	POUT [15:0]	Output data	0: Output low level 1: Output high level	R/W

Override this register to change the output status of the corresponding port when the I/O port is set to GPO.

9.4.3 General Output License Register (POERx)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
POUTE [15:0]															

Bit	Flag	Bit Name	Function	Read and write
b15~b0	POUTE [15:0]	Export licence	0: Output disable 1: Export enable	R/W

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.

9.4.4 General Output Set Register (POSRx)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
POS [15:0]															

Bit	Flag	Bit Name	Function	Read and write
b15~b0	POS [15:0]	Output height	0: No change to PODRx.POUT 1: Set to 1 corresponding to PODRx.POUT	R/W

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.

9.4.5 General Output Reset Register (PORRx)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
POR [15:0]															

Bit	Flag	Bit Name	Function	Read and write
b15~b0	POR [15:0]	Output low	0: No change to PODRx.POUT 1: corresponding to clear PODRx.POUT	R/W

The read value for this register is always 0x0000.

9.4.6 Universal Output Flip Register (POTRx)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
POT [15:0]															

Bit	Flag	Bit Name	Function	Read and write
b15~b0	POT [15:0]	Output inversion	0: No change to PODRx.POUT 1: Reverse PODRx.POUT	R/W

The read value for this register is always 0x0000.

9.4.7 Special Control Register (PSPCR)

Reset value: 0x001f

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

Bit	Flag	Bit Name	Function	Read and write
b15~b5	Reserved	-	Read at 0, write at 0	R/W
b4	SPFE [4]	Special function selection	0: Invalid NJTRST feature 1: NJTRST function valid	R/W
b3	SPFE [3]	Special function selection	0: JTDI function invalid 1: JTDI function valid	R/W
b2	SPFE [2]	Special function selection	0: JTDO_TRACESWO function invalid 1: JTDO_TRACESWO function valid	R/W
b1	SPFE [1]	Special function selection	0: JTMS_SWDIO function invalid 1: JTMS_SWDIO function valid	R/W
b0	SPFE [0]	Special function selection	0: JTCK_SWCLK function invalid 1: JTCK_SWCLK function valid	R/W

Note:

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

9.4.8 Public Control Register (PCCR)

Reset value: 0x1000

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

Bit	Flag	Bit Name	Function			Read and write
b15	Reserved	-	Read at 0, write at 0			R/W
b14-b12	RDWT [2:0]	Read Port Waiting	Set the number of wait periods inserted when reading registers PIDRx, PCRxy			R/W
			Operating frequency	Minimum waiting period	Predetermined area	
			~50MHz	0 period	0~7	
			50~100MHz	1 period	1~7	
			100~120MHz	2 period	2~7	
b11~b6	Reserved	-	Read at 0, write at 0			R/W
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			R/W

9.4.9 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]								-	-	-	-	-	-	-	WE

Bit	Flag	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	Reserved	-	Read at 0, write at 0	R/W
b0	WE	Write permission	0: 0: PSPCR, PCCR, PCRxy, PFSRxy registers write disable 1: 1: PSPCR, PCCR, PCRxy, PFSRxy register write enable	R/W

9.4.10 General Control Register (PCRxy)

Reset value: b0000_000x_0000_0000 *1

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

Bit	Flag	Bit Name	Function	Read and write
b15	DDIS	Digital function disable	0: digital function valid 1: digital function disable	R/W
b14	LTE	Output state latch	0: Invalid output latch 1: Output latch effective	R/W
b13	PINAE	Input normally-open	0: Input MOS normally open invalid 1: Input MOS normally open	R/W
b12	INTE	External interrupt enable	0: External interrupt input disable 1: External interrupt input enable	R/W
b11	Reserved	-	Read at 0, write at 0	R/W
b10	CINSEL	Input mode selection	0: Schmitt input 1: CMOS input	R/W
b9	INVE	Inverse enable	0: Input/output data not inverted 1: Inverted input/output data	R/W
b8	PIN	Input status	0: I/O port input status is low 1: I/O port input status is high Consistent with the PIN [y] feature in registerPIDRx	R
b7	PUD	Pull-down enable	0: Internal pull-down resistor invalid 1: Internal pull-down resistor valid	R/W
b6	PUU	Pull-up license	0: Invalid internal pull-up resistor 1: Pullup resistance is effective	R/W
b5~b4	DRV [1:0]	Drive mode selection	b00: Low driving force mode b01: Medium driving force mode b10 , b11 : High drive force mode	R/W
b3	Reserved	-	Read at 0, write at 0	R/W
b2	NOD	NMOS Opening Leakage	0: Normal CMOS Output Mode 1: NMOS open-drain output	R/W
b1	POUTE	Export licence	0: Output disable 1: Export enable Consistent with the POUTE [y] function in registerPOERx	R/W
b0	POUT	Output data	0: Output low level 1: Output high level Consistent with the POUT [y] function in registerPODRx	R/W

When DDIS is set to 1, all digital functions of the corresponding port are forcibly invalid, including general 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.

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.

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.

CINSEL is used to set the I/O input mode and affects all digital input functions such as GPI, EIRQ and peripheral inputs.

*1: The reset value of the general control register PCR of the following ports is not b0000_000x_0000_0000, please pay attention. XTAL32_IN, XTAL32_OUT multiplex port PC15, the reset value of PCRC15 and PCRC14 registers of PC14 is 0x8100.

9.4.11 Function Select Register (PFSRxy)

Reset value: 0x0000 *1

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

Bit	Flag	Bit Name	Function	Read and write
b15~b9	Reserved	-	Read at 0, write at 0	R/W
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	R/W
b7~b6	Reserved	-	Read at 0, write at 0	R/W
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	R/W

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 Manual**, set FSEL [5:0] to b000000 to select Func0, set to b000001 to select Func1, and so on, set to b001111 to select Func15. The general output function GPO corresponding to Func0 Table 2-1 Pin Function Table.

Note:

- 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.
- After reset, the PC14 and PC15 pins are configured as external low-speed oscillator interfaces. In their initial state, digital functions are disabled. To enable digital functions, it is necessary to first set the register CMU_XTAL32CR.XTAL32STP to 1 to disable the external low-speed oscillator, and then write 0 to the DDIS bit of the corresponding PCRxy register to activate the digital functionality.

9.4.12 Event Port Direction Selection Register (PEVNTDIRRm)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PDIR [15:0]															
Bit	Flag	Bit Name	Function												Read and write
b31~b16	Reserved	-	Read at 0, write at 0												R/W
b15~b0	PDIR15:0]	Direction selection	0: Event Port is an input function 1: Event Port is an output function												R/W

9.4.13 Event Port Input Data Register (PEVNTIDRm)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PIN [15:0]															
Bit	Flag	Bit Name	Function												Read and write
b31~b16	Reserved	-	Read at 0, write at 0												R/w
b15~b0	PIN [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

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.

9.4.14 Event Port Output Data Register (PEVNTODRm)

Reset value: 0x00000000

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

Bit	Flag	Bit Name	Function	Read and write
b31~b16	Reserved	-	Read at 0, write at 0	R/W
b15~b0	POUT [15:0]	Port output value	0: Event Port output low level 1: Event Port output high level	R/W

When the direction of the Event Port is set to the output state, write this register, before the set event is triggered, the initial output value of the Event Port. When the selected event is triggered, according to the setting values of PEVNTORRm and PEVNTOSRm, the corresponding bits of PEVNTODRm.POUT are cleared to 0, set to 1, or inverted, and output to the EVNTPmn port at the same time.

9.4.15 Event Port Output Data Reset Register (PEVNTORRm)

Reset value: 0x00000000

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

Bit	Flag	Bit Name	Function	Read and write
b31~b16	Reserved	-	Read at 0, write at 0	R/W
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	R/W

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

9.4.16 Event Port Output Data Set Register (PEVNTOSRm)

Reset value: 0x00000000

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

Bit	Flag	Bit Name	Function	Read and write
b31~b16	Reserved	-	Read at 0, write at 0	R/W
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	R/W

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

9.4.17 Event Port Rising Edge Input Permission Register (PEVNTRISRm)

Reset value: 0x00000000

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

Bit	Flag	Bit Name	Function	Read and Write
b31~b16	Reserved	-	Read at 0, write at 0	R/W
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	R/W

Event Port is used as an event source. When the RIS bit is set to 1, when the corresponding EVNTPmn input rising edge, an event is output to trigger other peripheral modules. 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.

9.4.18 Event Port Falling Edge Input Enable Register (PEVNTFALRm)

Reset value: 0x00000000

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

Bit	Flag	Bit Name	Function	Read and Write
b31~b16	Reserved	-	Read at 0, write at 0	R/W
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	R/W

Event Port is used as an event source. When the FAL bit is set to 1, when the corresponding EVNTP input falls, an event is output to trigger other peripheral modules. 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.

9.4.19 Event Port Input Filter Control Register (PEVNTNFCR)

Reset value: 0x00000000

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

Bit	Flag	Bit Name	Function	Read and write
b31~b27	Reserved	-	Read at 0, write at 0	R/W
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	R/W
b24	NFEN4	Digital Filtering License	0: Event Port4 digital filter is invalid 1: Event Port4 digital filter is valid	R/W
b23~b19	Reserved	-	Read at 0, write at 0	R/W
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	R/W
b16	NFEN3	Digital Filtering License	0: Event Port3 digital filter is invalid 1: Event Port3 digital filter is valid	R/W
b15~b11	Reserved	-	Read at 0, write at 0	R/W
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	R/W
b8	NFEN2	Digital Filtering License	0: Event Port2 digital filter is invalid 1: Event Port2 digital filter is valid	R/W
b7~b3	Reserved	-	Read at 0, write at 0	R/W
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	R/W
b0	NFEN1	Digital Filtering License	0: Event Port1 digital filter is invalid 1: Event Port1 digital filter is valid	R/W

9.5 Precautions

- 1) Do not set the same feature to multiple ports.
- 2) When using the analog function, please turn off the digital function of the corresponding port (DDIS=1).
- 3) Please perform port feature switching when the output latch is active (LTE = 1) to avoid out-of-sight burrs on the port during the switchover.
- 4) The registers described in Section 9.4 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 9-3 List of PORT Registers for 32bit Access

Address	b31 ~ b16	b15 ~ b0
0x40053800+0x10*n *1	Reserved	PIDRx
0x40053804+0x10*n	POERx	PODRx
0x40053808+0x10*n	PORRx	POSRx
0x4005380C+0x10*n	Reserved	POTRx
0x40053BF4	Reserved	PSPCR
0x40053BF8	Reserved	PCCR
0x40053BFC	Reserved	PWPR
0x40053C00+0x40*n+0x04*y	PFSRxy	PCRxy

Note *1 : In the address calculation formula, x=A~F corresponds to n=0~5, y=0~15

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 358 interrupt events, which are processed and sent to NVIC as interrupt requests (IRQs). It supports 184 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 EventList] 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
EIRQ13	Input	External interrupt event 13
EIRQ14	Input	External interrupt event 14

Pin Name	I/O	Description
EIRQ15	Input	External interrupt event 15

10.2 System Block Diagram

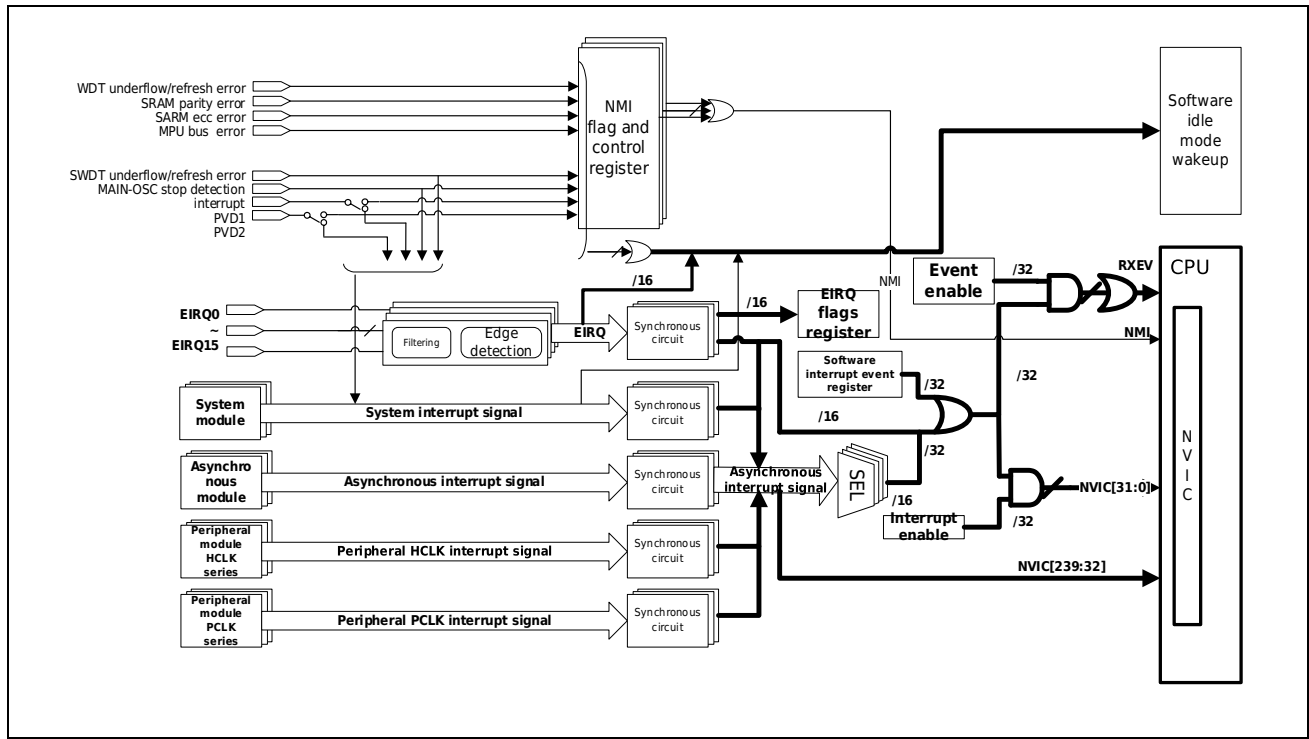


Figure 10-1 Interrupt System Block Diagram

10.3 Directional Scale

10.3.1 Interrupt Digraph

Table 10-2 Interrupt Vector Table

Vector Address	Vector Serial number	IRQ SN	Interrupt source	Description
ARM Core Interrupts				
0x00000000	0	-	ARM core	Initial stack pointer
0x00000004	1	-	ARM core	Initial Program Counter
0x00000008	2	-	ARM core	Non-maskable Interrupt (NMI)
0x0000000C	3	-	ARM core	Hard Fault
0x00000010	4	-	ARM core	MemManage Fault
0x00000014	5	-	ARM core	Bus Fault
0x00000018	6	-	ARM core	Usage Fault
0x0000001C	7	-	ARM core	Reserved
0x00000020	8	-	ARM core	Reserved
0x00000024	9	-	ARM core	Reserved
0x00000028	10	-	ARM core	Reserved
0x0000002C	11	-	ARM core	Supervisor call (SVCall)
0x00000030	12	-	ARM core	Debug Monitor
0x00000034	13	-	ARM core	Reserved
0x00000038	14	-	ARM core	Pendable request for system service(PendableSrvReq)
0x0000003C	15	-	ARM core	System tick timer (SysTick)
Non-ARM Core Interrupts				
0x00000040	16	0	INT_INTSEL0	Register INTC_INTSEL0 selects the interrupt event/software interrupt 0.
0x00000044	17	1	INT_INTSEL1	Register INTC_INTSEL1 selects interrupt event/software interrupt 1.
0x00000048	18	2	INT_INTSEL2	Register INTC_INTSEL2 selects interrupt event/software interrupt 2.
0x0000004C	19	3	INT_INTSEL3	Register INTC_INTSEL3 selects interrupt event/software interrupt 3.
0x00000050	20	4	INT_INTSEL4	Register INTC_INTSEL4 selects interrupt event/software interrupt 4.
0x00000054	21	5	INT_INTSEL5	Register INTC_INTSEL5 selects interrupt event/software interrupt 5.
0x00000058	22	6	INT_INTSEL6	Register INTC_INTSEL6 selects interrupt event/software interrupt 6.
0x0000005C	23	7	INT_INTSEL7	Register INTC_INTSEL7 selects interrupt event/software interrupt 7.
0x00000060	24	8	INT_INTSEL8	Register INTC_INTSEL8 selects interrupt event/software interrupt 8.
0x00000064	25	9	INT_INTSEL9	Register INTC_INTSEL9 selects interrupt event/software interrupt 9.
0x00000068	26	10	INT_INTSEL10	Register INTC_INTSEL10 selects interrupt event/software interrupt 10.
0x0000006C	27	11	INT_INTSEL11	Register INTC_INTSEL11 selects interrupt event/software interrupt 11.

Vector Address	Vector Serial number	IRQ SN	Interrupt source	Description
0x00000070	28	12	INT_INTSEL12	Register INTC_INTSEL12 selects interrupt event/software interrupt 12.
0x00000074	29	13	INT_INTSEL13	Register INTC_INTSEL13 selects interrupt event/software interrupt 13.
0x00000078	30	14	INT_INTSEL14	Register INTC_INTSEL14 selects interrupt event/software interrupt 14.
0x0000007C	31	15	INT_INTSEL15	Register INTC_INTSEL15 selects interrupt event/software interrupt 15.
0x00000080	32	16	PORT	External interrupt 0/software interrupt 16
0x00000084	33	17	PORT	External interrupt 1/software interrupt 17
0x00000088	34	18	PORT	External interrupt 2/software interrupt 18
0x0000008C	35	19	PORT	External interrupt 3/software interrupt 19
0x00000090	36	20	PORT	External interrupt 4/software interrupt 20
0x00000094	37	21	PORT	External interrupt 5/software interrupt 21
0x00000098	38	22	PORT	External interrupt 6/software interrupt 22
0x0000009C	39	23	PORT	External interrupt 7/software interrupt 23
0x000000A0	40	24	PORT	External interrupt 8/software interrupt 24
0x000000A4	41	25	PORT	External interrupt 9/software interrupt 25
0x000000A8	42	26	PORT	External interrupt 10/software interrupt 26
0x000000AC	43	27	PORT	External interrupt 11/software interrupt 27
0x000000B0	44	28	PORT	External interrupt 12/software interrupt 28
0x000000B4	45	29	PORT	External interrupt 13/software interrupt 29
0x000000B8	46	30	PORT	External interrupt 14/software interrupt 30
0x000000BC	47	31	PORT	External interrupt 15/software interrupt 31
0x000000C0	48	32	DMA_1	Transmission error interrupt
0x000000C4	49	33	DMA_1	Channel 0 transmission completion interrupt/channel 0 data block transmission completion interrupt
0x000000C8	50	34	DMA_1	Channel 1 transmission completion interrupt/channel 1 data block transmission completion interrupt
0x000000CC	51	35	DMA_1	Channel 2 transmission completion interrupt/channel 2 data block transmission completion interrupt
0x000000D0	52	36	DMA_1	Channel 3 transmission completion interrupt/channel 3 data block transmission completion interrupt
0x000000D4	53	37	DMA_1	Channel 4 transmission completion interrupt/channel 4 data block transmission completion interrupt
0x000000D8	54	38	DMA_1	Channel 5 transmission completion interrupt/channel 5 data block transmission completion interrupt
0x000000DC	55	39	DMA_1	Channel 6 transmission completion interrupt/channel 6 data block transmission completion interrupt
0x000000E0	56	40	DMA_1	Channel 7 transmission completion interrupt/channel 7 data block transmission completion interrupt
0x000000E4	57	41	EFM	Program (erase) error interrupt / read/write conflict error interrupt
0x000000E8	58	42	EFM	Operation end interrupt
0x000000EC	59	43	QSPI	QSPI bus error interrupt
0x000000F0	60	44	DCU_1	DCU_1 interruption
0x000000F4	61	45	DCU_2	DCU_2 interruption

Vector Address	Vector Serial number	IRQ SN	Interrupt source	Description
0x000000F8	62	46	DCU_3	DCU_3 interruption
0x000000FC	63	47	DMA_2	Transmission error interrupt
0x00000100	64	48	DMA_2	Channel 0 transmission completion interrupt/channel 0 data block transmission completion interrupt
0x00000104	65	49	DMA_2	Channel 1 transmission completion interrupt/channel 1 data block transmission completion interrupt
0x00000108	66	50	DMA_2	Channel 2 transmission completion interrupt/channel 2 data block transmission completion interrupt
0x0000010C	67	51	DMA_2	Channel 3 transmission completion interrupt/channel 3 data block transmission completion interrupt
0x00000110	68	52	DMA_2	Channel 4 transmission completion interrupt/channel 4 data block transmission completion interrupt
0x00000114	69	53	DMA_2	Channel 5 transmission completion interrupt/channel 5 data block transmission completion interrupt
0x00000118	70	54	DMA_2	Channel 6 transmission completion interrupt/channel 6 data block transmission completion interrupt
0x0000011C	71	55	DMA_2	Channel 7 transmission completion interrupt/channel 7 data block transmission completion interrupt
0x00000120	72	56	MAU	Square root operation completed interrupt
0x00000124	73	57	FMAC_1	Filter operation completed interrupt
0x00000128	74	58	FMAC_2	Filter operation completed interrupt
0x0000012C	75	59	FMAC_3	Filter operation completed interrupt
0x00000130	76	60	FMAC_4	Filter operation completed interrupt
0x00000134	77	61	Timer0_1	Channel A, B count compare match interrupt / input capture interrupt
0x00000138	78	62	Timer0_2	Channel A, B count compare match interrupt / input capture interrupt
0x0000013C	79	63	Timer2_1	Channel A, B count compare match interrupt / input capture interrupt
0x00000140	80	64	Timer2_1	Channel A, B count overflow interrupt
0x00000144	81	65	Timer2_2	Channel A, B count compare match interrupt / input capture interrupt
0x00000148	82	66	Timer2_2	Channel A, B count overflow interrupt
0x0000014C	83	67	Timer2_3	Channel A, B count compare match interrupt / input capture interrupt
0x00000150	84	68	Timer2_3	Channel A, B count overflow interrupt
0x00000154	85	69	Timer2_4	Channel A, B count compare match interrupt / input capture interrupt
0x00000158	86	70	Timer2_4	Channel A, B count overflow interrupt
0x0000015C	87	71	RTC	Alarm clock interruption, regular period interruption.
0x00000160	88	72	CMU	XTAL oscillation fault interrupt
0x00000164	89	73	PWC	WKTm compare match interrupt
0x00000168	90	74	SWDT	Count underflow/refresh error interrupt
0x0000016C	91	75	Timer6_1	General count compare match interrupt/input capture interrupt
0x00000170	92	76	Timer6_1	Count period match interrupt
0x00000174	93	77	Timer6_1	Dead zone error interrupt
0x00000178	94	78	Timer6_1	Dedicated count compare match interrupt
0x0000017C	95	79	TimerA_1	Count period match interrupt

Vector Address	Vector Serial number	IRQ SN	Interrupt source	Description
0x00000180	96	80	TimerA_1	Count comparison match interrupt
0x00000184	97	81	Timer6_2	General count compare match interrupt/input capture interrupt
0x00000188	98	82	Timer6_2	Count period match interrupt
0x0000018C	99	83	Timer6_2	Dead zone error interrupt
0x00000190	100	84	Timer6_2	Dedicated count compare match interrupt
0x00000194	101	85	TimerA_2	Count period match interrupt
0x00000198	102	86	TimerA_2	Count comparison match interrupt
0x0000019C	103	87	Timer6_3	General count compare match interrupt/input capture interrupt
0x000001A0	104	88	Timer6_3	Count period match interrupt
0x000001A4	105	89	Timer6_3	Dead zone error interrupt
0x000001A8	106	90	Timer6_3	Dedicated count compare match interrupt
0x000001AC	107	91	TimerA_3	Count period match interrupt
0x000001B0	108	92	TimerA_3	Count comparison match interrupt
0x000001B4	109	93	Timer6_4	General count compare match interrupt/input capture interrupt
0x000001B8	110	94	Timer6_4	Count period match interrupt
0x000001BC	111	95	Timer6_4	Dead zone error interrupt
0x000001C0	112	96	Timer6_4	Dedicated count compare match interrupt
0x000001C4	113	97	TimerA_4	Count period match interrupt
0x000001C8	114	98	TimerA_4	Count comparison match interrupt
0x000001CC	115	99	Timer6_5	General count compare match interrupt/input capture interrupt
0x000001D0	116	100	Timer6_5	Count period match interrupt
0x000001D4	117	101	Timer6_5	Dead zone error interrupt
0x000001D8	118	102	Timer6_5	Dedicated count compare match interrupt
0x000001DC	119	103	-	-
0x000001E0	120	104	Timer6_6	General count compare match interrupt/input capture interrupt
0x000001E4	121	105	Timer6_6	Count period match interrupt
0x000001E8	122	106	Timer6_6	Dead zone error interrupt
0x000001EC	123	107	Timer6_6	Dedicated count compare match interrupt
0x000001F0	124	108	Timer6_7	General count compare match interrupt/input capture interrupt
0x000001F4	125	109	Timer6_7	Count period match interrupt
0x000001F8	126	110	Timer6_7	Dead zone error interrupt
0x000001FC	127	111	Timer6_7	Dedicated count compare match interrupt
0x00000200	128	112	Timer6_8	General count compare match interrupt/input capture interrupt
0x00000204	129	113	Timer6_8	Count period match interrupt
0x00000208	130	114	Timer6_8	Dead zone error interrupt
0x0000020C	131	115	Timer6_8	Dedicated count compare match interrupt

Vector Address	Vector Serial number	IRQ SN	Interrupt source	Description
0x00000210	132	116	Timer6_9	General count compare match interrupt/input capture interrupt
0x00000214	133	117	Timer6_9	Count period match interrupt
0x00000218	134	118	Timer6_9	Dead zone error interrupt
0x0000021C	135	119	Timer6_9	Dedicated count compare match interrupt
0x00000220	136	120	Timer6_10	General count compare match interrupt/input capture interrupt
0x00000224	137	121	Timer6_10	Count period match interrupt
0x00000228	138	122	Timer6_10	Dead zone error interrupt
0x0000022C	139	123	Timer6_10	Dedicated count compare match interrupt
0x00000230	140	124	Timer4_1	General count compare match interrupt
0x00000234	141	125	Timer4_1	Count period match interrupt
0x00000238	142	126	Timer4_1	Heavy load count matching interrupt
0x0000023C	143	127	Timer4_1	Dedicated count compare match interrupt
0x00000240	144	128	CMP	CMP1 interruption
0x00000244	145	129	CMP	CMP2 interruption
0x00000248	146	130	-	-
0x0000024C	147	131	-	-
0x00000250	148	132	CAN_1	CAN_1 HOST interrupt
0x00000254	149	133	CAN_2	CAN_2 HOST interrupt
0x00000258	150	134	CAN_3	CAN_3 HOST interrupt
0x0000025C	151	135	USART_1	USART_1 interrupts (except RX line wake-up stop mode interrupt)
0x00000260	152	136	USART_2	USART_2 interruption
0x00000264	153	137	SPI_1	SPI_1 interruption
0x00000268	154	138	SPI_2	SPI_2 interruption
0x0000026C	155	139	TimerA_5	Count period match interrupt
0x00000270	156	140	TimerA_5	Count comparison match interrupt
0x00000274	157	141	TimerA_6	Count period match interrupt
0x00000278	158	142	TimerA_6	Count comparison match interrupt
0x0000027C	159	143	USBFS	USBFS Global Interrupt
0x00000280	160	144	PORT	Event Port1 Interrupt
0x00000284	161	145	PORT	Event Port2 Interrupt
0x00000288	162	146	PORT	Event Port3 Interrupt
0x0000028C	163	147	PORT	Event Port4 Interrupt
0x00000290	164	148	USART_3	USART_3 interruption
0x00000294	165	149	USART_4	USART_4 interruption
0x00000298	166	150	SPI_3	SPI_3 interruption
0x0000029C	167	151	SPI_4	SPI_4 interruption
0x000002A0	168	152	EMB	group0 interruption

Vector Address	Vector Serial number	IRQ SN	Interrupt source	Description
0x000002A4	169	153	EMB	group1 interruption
0x000002A8	170	154	EMB	group2 interruption
0x000002AC	171	155	EMB	group3 interruption
0x000002B0	172	156	EMB	group4 interruption
0x000002B4	173	157	EMB	group5 interruption
0x000002B8	174	158	EMB	group6 interruption
0x000002BC	175	159	CPM	HASH Interrupt
0x000002C0	176	160	USART_5	USART_5 interruption
0x000002C4	177	161	USART_6	USART_6 interruption
0x000002C8	178	162	MDIO	MDIO Interrupt
0x000002CC	179	163	-	-
0x000002D0	180	164	I2C_1	I2C_1 interruption
0x000002D4	181	165	I2C_2	I2C_2 interruption
0x000002D8	182	166	I2C_3	I2C_3 interruption
0x000002DC	183	167	USART_1	RX stop mode wake-up interrupt
0x000002E0	184	168	PWC	Programmable Voltage Detect 1 (PVD1) Interrupt
0x000002E4	185	169	PWC	Programmable Voltage Detect 1 (PVD2) Interrupt
0x000002E8	186	170	OTS	Temperature measurement completed interrupt
0x000002EC	187	171	CMU	FCM clock frequency measurement interrupt
0x000002F0	188	172	WDT	Count underflow/refresh error interrupt
0x000002F4	189	173	CTC	HRC frequency calibration value overflow interrupt
0x000002F8	190	174	ADC_1	ADC_1 interruption
0x000002FC	191	175	ADC_2	ADC_2 interruption
0x00000300	192	176	ADC_3	ADC_3 interruption
0x00000304	193	177	CPM	TRNG Interrupt
0x00000308	194	178	USART_1	USART_1_TCI interruption
0x0000030C	195	179	USART_2	USART_2_TCI interruption
0x00000310	196	180	USART_3	USART_3_TCI interruption
0x00000314	197	181	USART_4	USART_4_TCI interruption
0x00000318	198	182	USART_5	USART_5_TCI interruption
0x0000031C	199	183	USART_6	USART_6_TCI interruption

Note:

- Please refer to [Interrupt EventList] for the interrupt event number selected by the INTC_INTSEL register.
- The vector corresponding to IRQ number 84~239 is not configured with any interrupt event, so please do not use it.

10.3.2 Interrupt EventList

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 Source	Trigger source	NVIC Interrupt Request (IRQ) Number Mapping	
						Type A (INTC_INTSEL Number * 1)	Type B (Fixed)
0	0x000	PORT	EIRQ0	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [16]
1	0x001		EIRQ1	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [17]
2	0x002		EIRQ2	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [18]
3	0x003		EIRQ3	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [19]
4	0x004		EIRQ4	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [20]
5	0x005		EIRQ5	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [21]
6	0x006		EIRQ6	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [22]
7	0x007		EIRQ7	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [23]
8	0x008		EIRQ8	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [24]
9	0x009		EIRQ9	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [25]
10	0x00A		EIRQ10	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [26]
11	0x00B		EIRQ11	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [27]
12	0x00C		EIRQ12	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [28]
13	0x00D		EIRQ13	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [29]
14	0x00E		EIRQ14	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [30]
15	0x00F		EIRQ15	✓	✓	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)	-
27	0x01B	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
28	0x01C	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-

SN	Interrupt Event Number	Modules	Interrupt Name	Interrupt Source	Trigger source	NVIC Interrupt Request (IRQ) Number Mapping	
						Type A (INTC_INTSEL Number * 1)	Type B (Fixed)
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_1	DMA_1_ERR	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [32]
33	0x021		DMA_1_TC0	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [33]
34	0x022		DMA_1_BTC0	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
35	0x023		DMA_1_TC1	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [34]
36	0x024		DMA_1_BTC1	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
37	0x025		DMA_1_TC2	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [35]
38	0x026		DMA_1_BTC2	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
39	0x027		DMA_1_TC3	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [36]
40	0x028		DMA_1_BTC3	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
41	0x029		DMA_1_TC4	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [37]
42	0x02A		DMA_1_BTC4	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
43	0x02B		DMA_1_TC5	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [38]
44	0x02C		DMA_1_BTC5	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
45	0x02D		DMA_1_TC6	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [39]
46	0x02E		DMA_1_BTC6	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
47	0x02F		DMA_1_TC7	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [40]
48	0x030		DMA_1_BTC7	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
49	0x031	EFM	EFM_PEERR	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [41]
50	0x032		EFM_RDCOL	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
51	0x033		EFM_OPTEND	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [42]
52	0x034	USBFS	USBFS_SOF	-	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
53	0x035	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
54	0x036	QSPI	QSPI_INTR	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [43]
55	0x037	DCU	DCU1	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [44]
56	0x038		DCU2	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [45]
57	0x039		DCU3	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [46]
58	0x03A	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
59	0x03B	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
60	0x03C	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
61	0x03D	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
62	0x03E	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
63	0x03F	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
64	0x040	DMA_2	DMA_2_ERR	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [47]

SN	Interrupt Event Number	Modules	Interrupt Name	Interrupt Source	Trigger source	NVIC Interrupt Request (IRQ) Number Mapping	
						Type A (INTC_INTSEL Number * 1)	Type B (Fixed)
65	0x041		DMA_2_TC0	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [48]
66	0x042		DMA_2_BTC0	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
67	0x043		DMA_2_TC1	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [49]
68	0x044		DMA_2_BTC1	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
69	0x045		DMA_2_TC2	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [50]
70	0x046		DMA_2_BTC2	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
71	0x047		DMA_2_TC3	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [51]
72	0x048		DMA_2_BTC3	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
73	0x049		DMA_2_TC4	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [52]
74	0x04A		DMA_2_BTC4	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
75	0x04B		DMA_2_TC5	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [53]
76	0x04C		DMA_2_BTC5	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
77	0x04D		DMA_2_TC6	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [54]
78	0x04E		DMA_2_BTC6	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
79	0x04F		DMA_2_TC7	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [55]
80	0x050		DMA_2_BTC7	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
81	0x051	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
82	0x052	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
83	0x053	MAU	MAU_SQRT	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [56]
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	FMAC	FMAC_1_FIR	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [57]
92	0x05C		FMAC_2_FIR	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [58]
93	0x05D		FMAC_3_FIR	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [59]
94	0x05E		FMAC_4_FIR	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [60]
95	0x05F	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
96	0x060	Timer0_1	TMR0_1_CMPA	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [61]
97	0x061		TMR0_1_CMPB	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
98	0x062	Timer0_2	TMR0_2_CMPA	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [62]
99	0x063		TMR0_2_CMPB	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
100	0x064	Timer2_1	TMR2_1_CMPA	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [63]

SN	Interrupt Event Number	Modules	Interrupt Name	Interrupt Source	Trigger source	NVIC Interrupt Request (IRQ) Number Mapping	
						Type A (INTC_INTSEL Number * 1)	Type B (Fixed)
101	0x065	Timer2_2	TMR2_1_CMPB	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
102	0x066		TMR2_1_OVFA	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [64]
103	0x067		TMR2_1_OVFB	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
104	0x068		TMR2_2_CMPA	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [65]
105	0x069	Timer2_3	TMR2_2_CMPB	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
106	0x06A		TMR2_2_OVFA	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [66]
107	0x06B		TMR2_2_OVFB	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
108	0x06C		TMR2_3_CMPA	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [67]
109	0x06D	Timer2_4	TMR2_3_CMPB	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
110	0x06E		TMR2_3_OVFA	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [68]
111	0x06F		TMR2_3_OVFB	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
112	0x070		TMR2_4_CMPA	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [69]
113	0x071	Timer2_5	TMR2_4_CMPB	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
114	0x072		TMR2_4_OVFA	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [70]
115	0x073		TMR2_4_OVFB	✓	✓	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	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [71]
122	0x07A		RTC_PRD	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
123	0x07B	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
124	0x07C	XTAL32	XTAL32_STOP	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
125	0x07D	XTAL	XTAL_STOP	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [72]
126	0x07E	WKT	WKT_PRD	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [73]
127	0x07F	SWDT	SWDT_REFUDF	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [74]
128	0x080	Timer6_1	TMR6_1_GCMA	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [75]
129	0x081		TMR6_1_GCMB	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
130	0x082		TMR6_1_GCMC	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
131	0x083		TMR6_1_GCMD	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
132	0x084		TMR6_1_GCME	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
133	0x085		TMR6_1_GCMF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
134	0x086		TMR6_1_GOVF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [76]
135	0x087		TMR6_1_GUDF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
136	0x088		TMR6_1_GDTE	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [77]

SN	Interrupt Event Number	Modules	Interrupt Name	Interrupt Source	Trigger source	NVIC Interrupt Request (IRQ) Number Mapping	
						Type A (INTC_INTSEL Number * 1)	Type B (Fixed)
137	0x089	TimerA_1	TMR6_1_SCMA	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [78]
138	0x08A		TMR6_1_SCMB	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
139	0x08B		TMRA_1_OVF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [79]
140	0x08C		TMRA_1_UDF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
141	0x08D		TMRA_1_CMP	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [80]
142	0x08E	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
143	0x08F	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
144	0x090	Timer6_2	TMR6_2_GCMA	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [81]
145	0x091		TMR6_2_GCMB	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
146	0x092		TMR6_2_GCMC	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
147	0x093		TMR6_2_GCMD	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
148	0x094		TMR6_2_GCME	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
149	0x095		TMR6_2_GCMF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
150	0x096		TMR6_2_GOVF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [82]
151	0x097		TMR6_2_GUDF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
152	0x098		TMR6_2_GDTE	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [83]
153	0x099		TMR6_2_SCMA	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [84]
154	0x09A		TMR6_2_SCMB	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
155	0x09B	TimerA_2	TMRA_2_OVF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [85]
156	0x09C		TMRA_2_UDF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
157	0x09D		TMRA_2_CMP	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [86]
158	0x09E	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
159	0x09F	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
160	0x0A0	Timer6_3	TMR6_3_GCMA	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [87]
161	0x0A1		TMR6_3_GCMB	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
162	0x0A2		TMR6_3_GCMC	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
163	0x0A3		TMR6_3_GCMD	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
164	0x0A4		TMR6_3_GCME	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
165	0x0A5		TMR6_3_GCMF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
166	0x0A6		TMR6_3_GOVF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [88]
167	0x0A7		TMR6_3_GUDF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
168	0x0A8		TMR6_3_GDTE	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [89]
169	0x0A9		TMR6_3_SCMA	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [90]
170	0x0AA		TMR6_3_SCMB	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
171	0x0AB	TimerA_3	TMRA_3_OVF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [91]
172	0x0AC		TMRA_3_UDF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑

SN	Interrupt Event Number	Modules	Interrupt Name	Interrupt Source	Trigger source	NVIC Interrupt Request (IRQ) Number Mapping	
						Type A (INTC_INTSEL Number * 1)	Type B (Fixed)
173	0x0AD		TMRA_3_CMP	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [92]
174	0x0AE	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
175	0x0AF	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
176	0x0B0	Timer6_4	TMR6_4_GCMA	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [93]
177	0x0B1		TMR6_4_GCMB	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
178	0x0B2		TMR6_4_GCMC	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
179	0x0B3		TMR6_4_GCMD	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
180	0x0B4		TMR6_4_GCME	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
181	0x0B5		TMR6_4_GCMF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
182	0x0B6		TMR6_4_GOVF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [94]
183	0x0B7		TMR6_4_GUDF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
184	0x0B8		TMR6_4_GDTE	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [95]
185	0x0B9		TMR6_4_SCMA	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [96]
186	0x0BA		TMR6_4_SCMB	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
187	0x0BB	TimerA_4	TMRA_4_OVF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [97]
188	0x0BC		TMRA_4_UDF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
189	0x0BD		TMRA_4_CMP	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [98]
190	0x0BE	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
191	0x0BF	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
192	0x0C0	Timer6_5	TMR6_5_GCMA	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [99]
193	0x0C1		TMR6_5_GCMB	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
194	0x0C2		TMR6_5_GCMC	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
195	0x0C3		TMR6_5_GCMD	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
196	0x0C4		TMR6_5_GCME	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
197	0x0C5		TMR6_5_GCMF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
198	0x0C6		TMR6_5_GOVF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [100]
199	0x0C7		TMR6_5_GUDF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
200	0x0C8		TMR6_5_GDTE	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [101]
201	0x0C9		TMR6_5_SCMA	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [102]
202	0x0CA		TMR6_5_SCMB	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
203	0x0CB	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [103]
204	0x0CC		-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
205	0x0CD		-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
206	0x0CE		-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [103]
207	0x0CF	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
208	0x0D0	Timer6_6	TMR6_6_GCMA	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [104]

SN	Interrupt Event Number	Modules	Interrupt Name	Interrupt Source	Trigger source	NVIC Interrupt Request (IRQ) Number Mapping	
						Type A (INTC_INTSEL Number * 1)	Type B (Fixed)
209	0x0D1		TMR6_6_GCMB	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
210	0x0D2		TMR6_6_GCMC	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
211	0x0D3		TMR6_6_GCMD	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
212	0x0D4		TMR6_6_GCME	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
213	0x0D5		TMR6_6_GCMF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
214	0x0D6		TMR6_6_GOVF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [105]
215	0x0D7		TMR6_6_GUDF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
216	0x0D8		TMR6_6_GDTE	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [106]
217	0x0D9		TMR6_6_SCMA	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [107]
218	0x0DA		TMR6_6_SCMB	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
219	0x0DB	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
220	0x0DC	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
221	0x0DD	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
222	0x0DE	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
223	0x0DF	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
224	0x0E0	Timer6_7	TMR6_7_GCMA	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [108]
225	0x0E1		TMR6_7_GCMB	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
226	0x0E2		TMR6_7_GCMC	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
227	0x0E3		TMR6_7_GCMD	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
228	0x0E4		TMR6_7_GCME	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
229	0x0E5		TMR6_7_GCMF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
230	0x0E6		TMR6_7_GOVF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [109]
231	0x0E7		TMR6_7_GUDF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
232	0x0E8		TMR6_7_GDTE	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [110]
233	0x0E9		TMR6_7_SCMA	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [111]
234	0x0EA		TMR6_7_SCMB	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
235	0x0EB	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
236	0x0EC	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
237	0x0ED	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
238	0x0EE	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
239	0x0EF	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
240	0x0F0	Timer6_8	TMR6_8_GCMA	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [112]
241	0x0F1		TMR6_8_GCMB	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
242	0x0F2		TMR6_8_GCMC	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
243	0x0F3		TMR6_8_GCMD	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
244	0x0F4		TMR6_8_GCME	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑

SN	Interrupt Event Number	Modules	Interrupt Name	Interrupt Source	Trigger source	NVIC Interrupt Request (IRQ) Number Mapping	
						Type A (INTC_INTSEL Number * 1)	Type B (Fixed)
245	0x0F5		TMR6_8_GCMF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
246	0x0F6		TMR6_8_GOVF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [113]
247	0x0F7		TMR6_8_GUDF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
248	0x0F8		TMR6_8_GDTE	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [114]
249	0x0F9		TMR6_8_SCMA	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [115]
250	0x0FA		TMR6_8_SCMB	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
251	0x0FB	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
252	0x0FC	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
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	Timer6_9	TMR6_9_GCMA	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [116]
257	0x101		TMR6_9_GCMB	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
258	0x102		TMR6_9_GCMC	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
259	0x103		TMR6_9_GCMD	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
260	0x104		TMR6_9_GCME	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
261	0x105		TMR6_9_GCMF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
262	106		TMR6_9_GOVF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [117]
263	0x107		TMR6_9_GUDF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
264	0x108		TMR6_9_GDTE	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [118]
265	0x109		TMR6_9_SCMA	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [119]
266	0x10A		TMR6_9_SCMB	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
267	0x10B	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
268	0x10C	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
269	0x10D	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
270	0x10E	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
271	0x10F	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
272	0x110	Timer6_10	TMR6_10_GCMA	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [120]
273	0x111		TMR6_10_GCMB	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
274	0x112		TMR6_10_GCMC	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
275	0x113		TMR6_10_GCMD	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
276	0x114		TMR6_10_GCME	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
277	0x115		TMR6_10_GCMF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
278	0x116		TMR6_10_GOVF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [121]
279	0x117		TMR6_10_GUDF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
280	0x118		TMR6_10_GDTE	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [122]

SN	Interrupt Event Number	Modules	Interrupt Name	Interrupt Source	Trigger source	NVIC Interrupt Request (IRQ) Number Mapping	
						Type A (INTC_INTSEL Number * 1)	Type B (Fixed)
281	0x119		TMR6_10_SCMA	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [123]
282	0x11A		TMR6_10_SCMB	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
283	0x11B	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
284	0x11C	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
285	0x11D	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
286	0x11E	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
287	0x11F	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
288	0x120	Timer4_1	TMR4_1_GCMU H	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [124]
289	0x121		TMR4_1_GCMUL	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
290	0x122		TMR4_1_GCMVH	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
291	0x123		TMR4_1_GCMVL	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
292	0x124		TMR4_1_GCMW H	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
293	0x125		TMR4_1_GCMW L	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
294	0x126		TMR4_1_GOVF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [125]
295	0x127		TMR4_1_GUDF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
296	0x128		TMR4_1_GRLU	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [126]
297	0x129		TMR4_1_GRLV	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
298	0x12A		TMR4_1_GRLW	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
299	0x12B		TMR4_1_SCM0	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [127]
300	0x12C		TMR4_1_SCM1	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
301	0x12D		TMR4_1_SCM2	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
302	0x12E		TMR4_1_SCM3	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
303	0x12F		TMR4_1_SCM4	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
304	0x130		TMR4_1_SCM5	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
305	0x131	CMP	CMP_1_IRQ	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [128]
306	0x132		CMP_2_IRQ	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [129]
307	0x133		-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
308	0x134		-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
309	0x135	CAN	CAN_1_INT	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [132]
310	0x136		CAN_2_INT	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [133]
311	0x137		CAN_3_INT	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [134]
312	0x138	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
313	0x139	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
314	0x13A	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
315	0x13B	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-

SN	Interrupt Event Number	Modules	Interrupt Name	Interrupt Source	Trigger source	NVIC Interrupt Request (IRQ) Number Mapping	
						Type A (INTC_INTSEL Number * 1)	Type B (Fixed)
316	0x13C	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
317	0x13D	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
318	0x13E	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
319	0x13F	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
320	0x140	USART_1	USART_1_EI	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [135]
321	0x141		USART_1_RI	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
322	0x142		USART_1_TI	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
323	0x143		USART_1_TCI	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [178]
324	0x144		USART_1_RTO	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [135]
325	0x145	USART_2	USART_2_EI	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [136]
326	0x146		USART_2_RI	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
327	0x147		USART_2_TI	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
328	0x148		USART_2_TCI	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [179]
329	0x149		USART_2_RTO	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [136]
330	0x14A	SPI_1	SPI_1_SPRI	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [137]
331	0x14B		SPI_1_SPTI	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
332	0x14C		SPI_1_SPII	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
333	0x14D		SPI_1_SPEI	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
334	0x14E		SPI_1_SPEND	-	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
335	0x14F	SPI_2	SPI_2_SPRI	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [138]
336	0x150		SPI_2_SPTI	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
337	0x151		SPI_2_SPII	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
338	0x152		SPI_2_SPEI	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
339	0x153		SPI_2_STEND	-	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
340	0x154	TimerA_5	TMRA_5_OVF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [139]
341	0x155		TMRA_5_UDF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
342	0x156		TMRA_5_CMP	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [140]
343	0x157	TimerA_6	TMRA_6_OVF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [141]
344	0x158		TMRA_6_UDF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
345	0x159		TMRA_6_CMP	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [142]
346	0x15A	USBFS	USBFS_GLB	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [143]
347	0x15B	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
348	0x15C	EVENT port	EVENT_PORT1	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [144]
349	0x15D		EVENT_PORT2	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [145]
350	0x15E		EVENT_PORT3	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [146]
351	0x15F		EVENT_PORT4	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [147]

SN	Interrupt Event Number	Modules	Interrupt Name	Interrupt Source	Trigger source	NVIC Interrupt Request (IRQ) Number Mapping	
						Type A (INTC_INTSEL Number * 1)	Type B (Fixed)
352	0x160	USART_3	USART_3_BRKW KPI	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [148]
353	0x161		USART_3_EI	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
354	0x162		USART_3_RI	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
355	0x163		USART_3_TI	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
356	0x164		USART_3_TCI	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [180]
357	0x165	USART_4	USART_4_EI	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [149]
358	0x166		USART_4_RI	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
359	0x167		USART_4_TI	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
360	0x168		USART_4_TCI	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [181]
361	0x169		USART_4_RTO	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [149]
362	0x16A	SPI_3	SPI_3_SPRI	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [150]
363	0x16B		SPI_3_SPTI	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
364	0x16C		SPI_3_SPII	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
365	0x16D		SPI_3_SPEI	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
366	0x16E		SPI_3_SPEND	-	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
367	0x16F	SPI_4	SPI_4_SPRI	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [151]
368	0x170		SPI_4_SPTI	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
369	0x171		SPI_4_SPII	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
370	0x172		SPI_4_SPEI	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
371	0x173		SPI_4_SPEND	-	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
372	0x174	EMB	EMB_GR0	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [152]
373	0x175		EMB_GR1	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [153]
374	0x176		EMB_GR2	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [154]
375	0x177		EMB_GR3	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [155]
376	0x178		EMB_GR4	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [156]
377	0x179		EMB_GR5	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [157]
378	0x17A		EMB_GR6	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [158]
379	0x17B	HASH	HASH_INT	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [159]
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_5	USART_5_EI	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [160]
385	0x181		USART_5_RI	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
386	0x182		USART_5_TI	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
387	0x183		USART_5_TCI	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [182]

SN	Interrupt Event Number	Modules	Interrupt Name	Interrupt Source	Trigger source	NVIC Interrupt Request (IRQ) Number Mapping	
						Type A (INTC_INTSEL Number * 1)	Type B (Fixed)
388	0x184		USART_5_RTO	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [160]
389	0x185	USART_6	USART_6_BRKW KPI	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [161]
390	0x186		USART_6_EI	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
391	0x187		USART_6_RI	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
392	0x188		USART_6_TI	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
393	0x189		USART_6_TCI	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [183]
394	0x18A	USART_1	USART_1_TENDI	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [135]
395	0x18B	USART_2	USART_2_TENDI	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [136]
396	0x18C	USART_3	USART_3_TENDI	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [148]
397	0x18D	USART_4	USART_4_TENDI	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [149]
398	0x18E	USART_5	USART_5_TENDI	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [160]
399	0x18F	USART_6	USART_6_TENDI	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [161]
400	0x190	MDIO	MDIO_WRF	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [162]
401	0x191		MDIO_ADRF	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
402	0x192		MDIO_INCF	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
403	0x193		MDIO_RDF	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
404	0x194		MDIO_DEVM	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
405	0x195		MDIO_DEVN	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
406	0x196		MDIO_PHYM	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
407	0x197		MDIO_PHYN	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
408	0x198		MDIO_TANM	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
409	0x199		MDIO_TO	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
410	0x19A		MDIO_UDR	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
411	0x19B		MDIO_OVR	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
412	0x19C		MDIO_RBNE	✓	-	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*2	-	✓	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)	-

SN	Interrupt Event Number	Modules	Interrupt Name	Interrupt Source	Trigger source	NVIC Interrupt Request (IRQ) Number Mapping	
						Type A (INTC_INTSEL Number * 1)	Type B (Fixed)
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)	IRQ [163]
432	0x1B0		-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
433	0x1B1		-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
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	I2C_1*4	I2C_1_RXI	✓	✓*4	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [164]
449	0x1C1		I2C_1_TXI	✓	✓*4	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
450	0x1C2		I2C_1_TEI	✓	✓*4	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
451	0x1C3		I2C_1_EEI	✓	✓*4	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
452	0x1C4	I2C_2*4	I2C_2_RXI	✓	✓*4	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [165]
453	0x1C5		I2C_2_TXI	✓	✓*4	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
454	0x1C6		I2C_2_TEI	✓	✓*4	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
455	0x1C7		I2C_2_EEI	✓	✓*4	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
456	0x1C8	I2C_3*4	I2C_3_RXI	✓	✓*4	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [166]
457	0x1C9		I2C_3_TXI	✓	✓*4	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
458	0x1CA		I2C_3_TEI	✓	✓*4	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
459	0x1CB		I2C_3_EEI	✓	✓*4	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑

SN	Interrupt Event Number	Modules	Interrupt Name	Interrupt Source	Trigger source	NVIC Interrupt Request (IRQ) Number Mapping	
						Type A (INTC_INTSEL Number * 1)	Type B (Fixed)
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	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [167]
465	0x1D1	PVD	PVD_PVD1	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [168]
466	0x1D2		PVD_PVD2	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [169]
467	0x1D3	OTS	OTS	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [170]
468	0x1D4	FCM	FCMFERRI	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [171]
469	0x1D5		FCMMENDI	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
470	0x1D6		FCMCOVFI	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
471	0x1D7	WDT	WDT_REFUDF	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [172]
472	0x1D8	CTC	CTC_ERR	✓	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [173]
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	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [174]
481	0x1E1		ADC_1_EOCB	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
482	0x1E2		ADC_1_CMP0	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
483	0x1E3		ADC_1_CMP1	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
484	0x1E4	ADC_2	ADC_2_EOCA	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [175]
485	0x1E5		ADC_2_EOCB	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
486	0x1E6		ADC_2_CMP0	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
487	0x1E7		ADC_2_CMP1	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
488	0x1E8	ADC_3	ADC_3_EOCA	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [176]
489	0x1E9		ADC_3_EOCB	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
490	0x1EA		ADC_3_CMP0	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
491	0x1EB		ADC_3_CMP1	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	↑
492	0x1EC	TRNG	TRNG_END	✓	✓	IRQ [0]~ [15] (INTC_INTSEL0~15)	IRQ [177]
493	0x1ED	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
494	0x1EE	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-
495	0x1EF	-	-	-	-	IRQ [0]~ [15] (INTC_INTSEL0~15)	-

SN	Interrupt Event Number	Modules	Interrupt Name	Interrupt Source	Trigger source	NVIC Interrupt Request (IRQ) Number Mapping	
						Type A (INTC_INTSEL Number * 1)	Type B (Fixed)
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)	-
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: If the interrupt event is not configured, the number selected by the interrupt selection register invalid.

*2: AOS_STRG is generated by software setting the STRG bit in the INTSFTRG register.

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

*4: When the I2C_1/ I2C_2/ I2C_3 event is used as the AOS source, it can only trigger the DMA module action, and cannot trigger other AOS target modules.

10.4 Functional 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 NVIC 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 EventList] section. When using NVIC interrupt requests 0 to 15, the corresponding interrupt event mask position in the interrupt event mask register `INTC_INTEN` can be set to 0, which 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

NVIC 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:

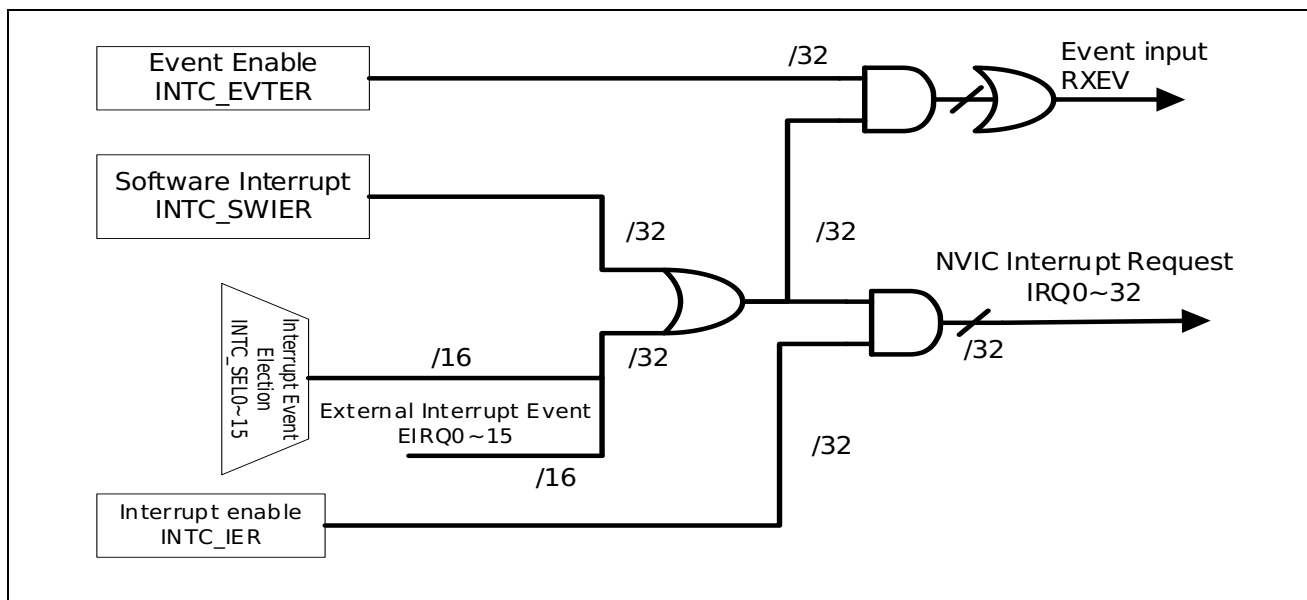


Figure 10-2 Interrupt Request/Event Input Selection

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 Core

Select an interrupt event, set `INTC_INTSEL` according to the selected interrupt request and enable the `INTC_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 wakes up the 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` ($x=0\sim15$) 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 periods will be filtered out. The sampling period is set by the `EISMPCLK` [1:0] bits in the `INTC_EIRQCRx` register.

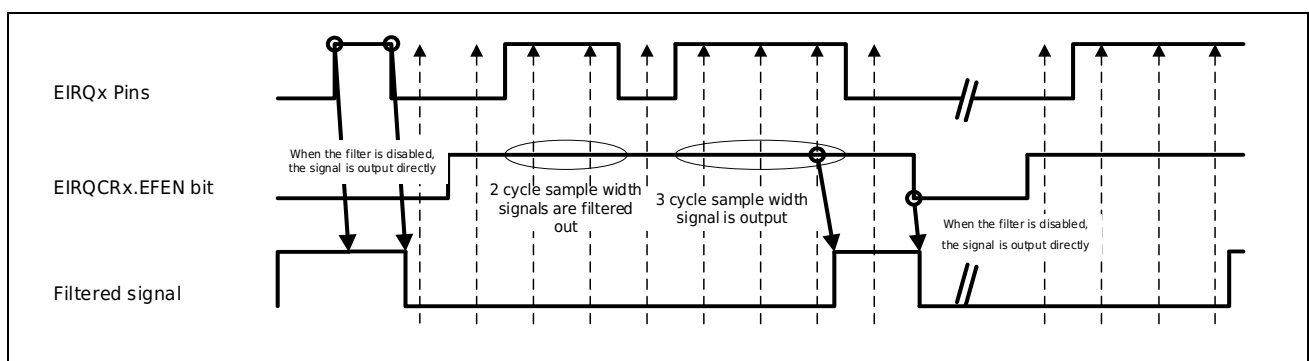


Figure 10-3 Schematic Diagram of Digital Filter Operation

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 0~32, 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 0~32, 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: main oscillator stop detection interrupt, SWDT underflow / refresh interrupt, and low voltage detection PVD1/PVD2 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 Undervoltage detection 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”.

10.5 Register Description

Table 10-4 INTC Register List

INTC Base Address: 0x40051000

Register name	Symbol	Offset Address	Bit Width	Reset Value
Nomaskable interrupt enable register	INTC_NMIER	0x0004	32	0x00000000
Nomaskable interrupt flag register	INTC_NMIFR	0x0008	32	0x00000000
Nomaskable interrupt flag clear register	INTC_NMIFCR	0x000C	32	0x00000000
External Interrupt Control Register 0	INTC_EIRQCR0	0x0010	32	0x00000000
External Interrupt Control Register 1	INTC_EIRQCR1	0x0014	32	0x00000000
External Interrupt Control Register 2	INTC_EIRQCR2	0x0018	32	0x00000000
External Interrupt Control Register 3	INTC_EIRQCR3	0x001C	32	0x00000000
External Interrupt Control Register 4	INTC_EIRQCR4	0x0020	32	0x00000000
External Interrupt Control Register 5	INTC_EIRQCR5	0x0024	32	0x00000000
External Interrupt Control Register 6	INTC_EIRQCR6	0x0028	32	0x00000000
External Interrupt Control Register 7	INTC_EIRQCR7	0x002C	32	0x00000000
External Interrupt Control Register 8	INTC_EIRQCR8	0x0030	32	0x00000000
External Interrupt Control Register 9	INTC_EIRQCR9	0x0034	32	0x00000000
External Interrupt Control Register 10	INTC_EIRQCR10	0x0038	32	0x00000000
External Interrupt Control Register 11	INTC_EIRQCR11	0x003C	32	0x00000000
External Interrupt Control Register 12	INTC_EIRQCR12	0x0040	32	0x00000000
External Interrupt Control Register 13	INTC_EIRQCR13	0x0044	32	0x00000000
External Interrupt Control Register 14	INTC_EIRQCR14	0x0048	32	0x00000000
External Interrupt Control Register 15	INTC_EIRQCR15	0x004C	32	0x00000000
Stop mode wakeup event enable register	INTC_WKEN	0x0050	32	0x00000000
External interrupt flag register	INTC_EIFR	0x0054	32	0x00000000
External interrupt flag clear register	INTC_EIFCR	0x0058	32	0x00000000
Interrupt Event Select Register 0	INTC_INTSEL0	0x005C	32	0x000001FF
Interrupt Event Select Register 1	INTC_INTSEL1	0x0060	32	0x000001FF
Interrupt Event Select Register 2	INTC_INTSEL2	0x0064	32	0x000001FF
Interrupt Event Select Register 3	INTC_INTSEL3	0x0068	32	0x000001FF
Interrupt Event Select Register 4	INTC_INTSEL4	0x006C	32	0x000001FF
Interrupt Event Select Register 5	INTC_INTSEL5	0x0070	32	0x000001FF
Interrupt Event Select Register 6	INTC_INTSEL6	0x0074	32	0x000001FF
Interrupt Event Select Register 7	INTC_INTSEL7	0x0078	32	0x000001FF
Interrupt Event Select Register 8	INTC_INTSEL8	0x007C	32	0x000001FF
Interrupt Event Select Register 9	INTC_INTSEL9	0x0080	32	0x000001FF
Interrupt Event Select Register 10	INTC_INTSEL10	0x0084	32	0x000001FF
Interrupt Event Select Register 11	INTC_INTSEL11	0x0088	32	0x000001FF

Register name	Symbol	Offset Address	Bit Width	Reset Value
Interrupt Event Select Register 12	INTC_INTSEL12	0x008C	32	0x000001FF
Interrupt Event Select Register 13	INTC_INTSEL13	0x0090	32	0x000001FF
Interrupt Event Select Register 14	INTC_INTSEL14	0x0094	32	0x000001FF
Interrupt Event Select Register 15	INTC_INTSEL15	0x0098	32	0x000001FF
Interrupt event mask register 0	INTC_INTEN0	0x025C	32	0xFFFFFFFF
Interrupt event mask register 1	INTC_INTEN1	0x0260	32	0xFFFFFFFF
Interrupt event mask register 2	INTC_INTEN2	0x0264	32	0xFFFFFFFF
Interrupt event mask register 3	INTC_INTEN3	0x0268	32	0xFFFFFFFF
Interrupt event mask register 4	INTC_INTEN4	0x026C	32	0xFFFFFFFF
Interrupt event mask register 5	INTC_INTEN5	0x0270	32	0xFFFFFFFF
Interrupt event mask register 6	INTC_INTEN6	0x0274	32	0xFFFFFFFF
Interrupt event mask register 7	INTC_INTEN7	0x0278	32	0xFFFFFFFF
Interrupt event mask register 8	INTC_INTEN8	0x027C	32	0xFFFFFFFF
Interrupt event mask register 9	INTC_INTEN9	0x0280	32	0xFFFFFFFF
Interrupt event mask register 10	INTC_INTEN10	0x0284	32	0xFFFFFFFF
Interrupt event mask register 11	INTC_INTEN11	0x0288	32	0xFFFFFFFF
Interrupt event mask register 12	INTC_INTEN12	0x028C	32	0xFFFFFFFF
Interrupt event mask register 13	INTC_INTEN13	0x0290	32	0xFFFFFFFF
Interrupt event mask register 14	INTC_INTEN14	0x0294	32	0xFFFFFFFF
Interrupt event mask register 15	INTC_INTEN15	0x0298	32	0xFFFFFFFF
Software interrupt register	INTC_SWIER	0x029C	32	0x00000000
Event enable register	INTC_EVTER	0x02A0	32	0x00000000
Interrupt enable register	INTC_IER	0x02A4	32	0xFFFFFFFF

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
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved				WDT EN	BUS ERRE N	REC CERR EN	RPAR ERRE N	Reserved		XTAL STPE N	Rese rved	PVD 2EN	PVD 1EN	SWD TEN	Rese rved

Bit	Flag	Bit Name	Function	Read and write
b31~b12	Reserved	-	Read at "0", write at "0"	R/W
b11	WDTEN	WDT underflow/refresh interrupt selection	0: Disable interrupt as a nomaskable interrupt source 1: Choose interrupt as nomaskable interrupt source	R/W
b10	BUSERREN	MPU main bus error interrupt selection	0: Disable interrupt as a nomaskable interrupt source 1: Choose interrupt as nomaskable interrupt source	R/W
b9	RECCERREN	SRAM ECC check error interrupt selection	0: Disable interrupt as a nomaskable interrupt source 1: Choose interrupt as nomaskable interrupt source	R/W
b8	RPARERREN	SRAM parity error interrupt select	0: Disable interrupt as a nomaskable interrupt source 1: Choose interrupt as nomaskable interrupt source	R/W
b7~b6	Reserved	-	Read at "0", write at "0"	R/W
b5	XATLSTPEN	Detect host Oscillator Stop Interrupt Selection	0: Disable interrupt as a nomaskable interrupt source 1: Choose interrupt as nomaskable interrupt source	R/W
b4	Reserved	-	Read at "0", write at "0"	R/W
b3	PVD2EN	Low voltage detection PVD2 interrupt selection	0: Disable interrupt as a nomaskable interrupt source 1: Choose interrupt as nomaskable interrupt source	R/W
b2	PVD1EN	Low voltage detection PVD1 interrupt selection	0: Disable interrupt as a nomaskable interrupt source 1: Choose interrupt as nomaskable interrupt source	R/W
b1	SWDTEN	SWDT underflow/refresh interrupt selection	0: Disable interrupt as a nomaskable interrupt source 1: Choose interrupt as nomaskable interrupt source	R/W
b0	Reserved	-	Read at "0", write at "0"	R/W

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
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved				WDT F	BUS ERRF	REC CERR F	RPAR ERRF	Reserved		XTAL STPF	Reserved	PVD 2F	PVD 1F	SWD TF	Reserved

Bit	Flag	Bit Name	Function	Read and write
b31~b12	Reserved	-	Read at "0", write at "0"	R/W
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 host 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	Reserved	-	Read at "0", write at "0"	R/W
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	Reserved	-	Read at "0", write at "0"	R/W
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	Reserved	-	Read at "0", write at "0"	R/W

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
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved				WDT FCLR	BUS ERRF CLR	REC CERR FCLR	RPAR ERRF CLR	Reserved		XTAL STPF CLR	Rese rved	PVD 2FCL R	PVD 1FCL R	SWD TFCL R	Rese rved
Bit	Flag	Bit Name		Function		Read and write									
b31~b12	Reserved	-		Read at "0", write at "0"		R/W [Note 1]									
b11	WDTFCLR	WDT underflow/refresh interrupt flag clear		0: Invalid 1: Clear WDT underflow/refresh flag		R/W [Note 1]									
b10	BUSERRFCLR	MPU main bus error interrupt flag cleared		0: Invalid 1: Clear the MPU main bus error flag		R/W [Note 1]									
b9	RECCERRFCLR	SRAM DED parity error interrupt flag clear		0: Invalid 1: Clear the SRAM DED verification error flag		R/W [Note 1]									
b8	RPARERRFCLR	SRAM parity error interrupt flag cleared		0: Invalid 1: Clear the SRAM parity error flag		R/W [Note 1]									
b7~b6	Reserved	-		Read at "0", write at "0"		R/W [Note 1]									
b5	XTALSTPFCLR	Detect main oscillator stop interrupt flag clear		0: Invalid 1: Clear the detection of the stop flag of the main oscillator		R/W [Note 1]									
b4	Reserved	-		Read at "0", write at "0"		R/W									
b3	PVD2FCLR	Low voltage detection PVD2 interrupt flag clear		0: Invalid 1: Clear low voltage detection PVD2 flag		R/W [Note 1]									
b2	PVD1FCLR	Low voltage detection PVD1 interrupt flag clear		0: Invalid 1: Clear low voltage detection PVD1 flag		R/W [Note 1]									
b1	SWDTFCLR	SWDT underflow/refresh interrupt flag cleared		0: Invalid 1: Clear SWDT underflow/refresh flag		R/W [Note 1]									
b0	Reserved	-		Read at "0", write at "0"		R/W [Note 1]									

[Note 1] Only "1" can be written, and it will be "0" when read.

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
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
NOC EN	Reserved	NOCSEL [1:0]						EFEN	Reserved	EISMPCLK [1:0]		Reserved		EIRQTRG [1:0]	
Bit	Flag	Bit Name	Function										Read and write		
b31~b16	Reserved	-	Read at "0", write at "0"										R/W		
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.										R/W		
b14	Reserved	-	Read at "0", write at "0"										R/W		
b13~b12	NOCSEL [1:0]	Analog filter filter width selection	0 0: Filtering width level 1 0 1: Filtering width level 2 1 0: Filtering width level 3 1 1: Filtering width level 4 For specific values of each gear, please refer to the EIRQ Filter features section in the Electrical Features of Data Sheet .										R/W		
b11~b8	Reserved	-	Read at "0", write at "0"										R/W		
b7	EFEN	EIRQ digital filter enable	0: Disable digital filter function 1: Allow digital filter function										R/W		
b6	Reserved	-	Read at "0", write at "0"										R/W		
b5~b4	EISMPCLK [1:0]	Digital filter sampling period selection	0 0: 1*PCLK3 period 0 1: 8*PCLK3 periods 1 0: 32*PCLK3 periods 1 1: 64*PCLK3 periods										R/W		
b3~b2	Reserved	-	Read at "0", write at "0"										R/W		
b1~b0	EIRQTRG [1:0]	Trigger selection	0 0: Falling edge 0 1: Rising edge 1 0: Double edge 1 1: Low level										R/W		

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
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EIF [15:0]															
Bit	Flag	Bit Name	Function										Read and write		
b31~b16	Reserved	-	Read at "0", write at "0"										R/W		
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{EIFCLR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EIFCLR [15:0]															
Bit	Flag	Bit Name	Function										Read and write		
b31~b16	Reserved	-	Read at "0", write at "0"										R/W		
b15~b0	EIFCLR	EIRQ Flag Clear Bit	0: Writing "0" is invalid 1: Write "1" to clear the EIFR register										R/W		

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
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved							INTSEL [8:0]								
Bit	Flag	Bit Name	Function												Read and write
b31~b9	Reserved	-	Read at "0", write at "0"												R/W
b8~b0	INTSEL [8:0]	Interrupt Event Request Selection	9'h000~9'h1FE: 10.3.2Events corresponding to the interrupt event table												R/W

10.5.8 Interrupt Event Mask 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	Flag	Bit Name	Function												Read and write
b31~b0	INTEN [x*32+31:x*32] x=0~15	Interrupt 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.												R/W

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
-	-	CMP 2WK EN	USF WKE N	-	RX WKE N	TMR 2OVF WKE N	TMR 2CMP WKE N	TMR 0CMP WKE N	RTC PRD WKE N	RTC ALM WKE N	WKT MWK EN	CMP 1WK EN	PVD 2WKE N	PVD 1WK EN	SWD TWKE N
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EIRQWKEN [15:0]															

Bit	Flag	Bit Name	Function	Read and write
b31~b30	Reserved	-	Read at “0”, write at “0”	R/W
b29	CMP2WKEN	CMP2 Stop mode wake-up enable	0: Arousal prohibition 1: Arousal permit	R/W
b28	USFWKEN	USBFS_WKUP stop mode wakeup enable	0: Arousal prohibition 1: Arousal permit	R/W
b27	Reserved	-	Read at “0”, write at “0”	R/W
b26	RXWKEN	USART_1 RX line stop mode wake-up enable	0: Arousal prohibition 1: Arousal permit	R/W
b25	TMR2OVFWKEN	TMR2_1_OVFA stop mode wake-up enable	0: Arousal prohibition 1: Arousal permit	R/W
b24	TMR2CMPWKEN	TMR2_1_CMPA stop mode wake-up enable	0: Arousal prohibition 1: Arousal permit	R/W
b23	TMR0CMPWKEN	TMR0_1_CMPA stop mode wake-up enable	0: Arousal prohibition 1: Arousal permit	R/W
b22	RTCPRDWKEN	RTC_PRD stop mode wakeup enable	0: Arousal prohibition 1: Arousal permit	R/W
b21	RTCALMWKEN	RTC_ALM stop mode wake-up enable	0: Arousal prohibition 1: Arousal permit	R/W
b20	WKTMWKEN	WKT_MW_KMP stop mode wake-up enable	0: Arousal prohibition 1: Arousal permit	R/W
b19	CMP1WKEN	CMP1 stop mode wake-up enable	0: Arousal prohibition 1: Arousal permit	R/W
b18	PVD2WKEN	PVD2 stop mode wake-up enable	0: Arousal prohibition 1: Arousal permit	R/W
b17	PVD1WKEN	PVD1 stop mode wake-up enable	0: Arousal prohibition 1: Arousal permit	R/W
b16	SWDTWKEN	SWDT_REFUDF stop mode wakeup enable	0: Arousal prohibition 1: Arousal permit	R/W
b15~b0	EIRQWKEN [15:0]	EIRQ stop mode wakeup enable	0: Arousal prohibition 1: Arousal permit	R/W

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
SW IE [31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SW IE [15:0]															
Bit	Flag	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.											R/W	

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	Flag	Bit Name	Function											Read and write	
b31~b0	EVTE	Event enable	0: Event selection disabled 1: Event selection permission											R/W	

10.5.12 Interrupt 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	Flag	Bit Name	Function											Read and write	
b31~b0	IEN	Interrupt 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											R/W	

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 Introduction

The automatic operation system (Automatic Operation System) is used to realize the linkage between peripheral hardware circuits without the help of the CPU. Use the events generated by the peripheral circuit as the AOS source (AOS Source), such as the comparison and matching of the timer, timing overflow, the periodic signal of the RTC, and the various states of the communication module's sending and receiving data (idle, receiving data full, sending data end, Send data empty), ADC conversion end, etc., to trigger other peripheral circuit actions. The triggered peripheral circuit action is called AOS target (AOS Target).

11.1.1 Function Overview

- There are 322 AOS sources in total. Except for special restrictions, each AOS target can select one of them as the trigger source. 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. In this way, for an AOS target, any trigger event from any of the three trigger sources can trigger the AOS target action.
- 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:
 - 3 DCU trigger targets
 - 17 DMA trigger targets
 - 4 advanced control timers (Timer6) trigger targets
 - 1 General-purpose control timer 4 (Timer4) trigger target
 - 1 general timer 0 (Timer0) trigger target
 - 1 general timer 2 (Timer2) trigger targets
 - 4 general timer A (TimerA) trigger targets
 - 2 Event Port trigger targets
 - 2 HASH trigger targets
 - 1 temperature sensor (OTS) trigger target
 - 3 groups of 2 ADC trigger targets in each group, used for ADC1~ADC3 sequence trigger

11.1.2 Module Block Diagram

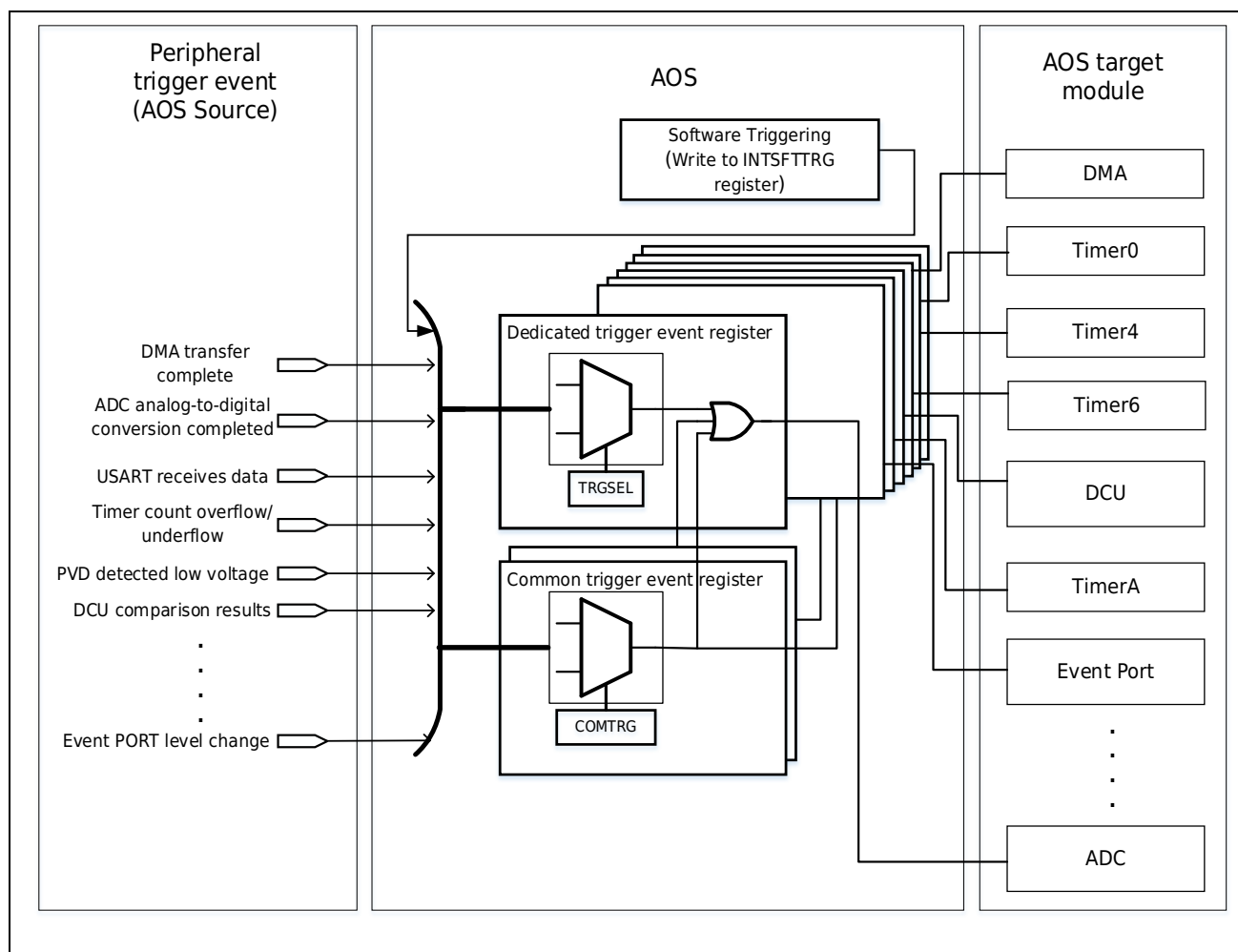


Figure 11-1 AOS System Block Diagram

11.2 Functional Description

11.2.1 AOS Source Events List

the AOS source event number, see the interrupt event table in Chapter [Interrupt Controller (INTC)].

Events flagged with “ ✓ ” in the “ Trigger Source ” column in the table can be used as AOS sources.

11.2.2 AOS target list

Table 11-1 AOS Target List

Modules	SN	Action
DCU_1	0	Trigger addition/subtraction operation
DCU_2	1	Trigger addition/subtraction operation
DCU_3	2	Trigger addition/subtraction operation
DMA_1	3	Trigger channel 0 starting transmission
	4	Trigger channel 1 starting transmission
	5	Trigger channel 2 starting transmission
	6	Trigger channel 3 starting transmission
	7	Trigger channel 4 starting transmission
	8	Trigger channel 5 starting transmission
	9	Trigger channel 6 starting transmission
	10	Trigger channel 7 starting transmission
DMA_2	11	Trigger channel 0 starting transmission
	12	Trigger channel 1 starting transmission
	13	Trigger channel 2 starting transmission
	14	Trigger channel 3 starting transmission
	15	Trigger channel 4 starting transmission
	16	Trigger channel 5 starting transmission
	17	Trigger channel 6 starting transmission
	18	Trigger channel 7 starting transmission
DMA_1&2	19	Trigger Event Triggers Channel Reset
Timer6	20	Trigger counter start/stop/clear/refresh/capture/increment/decrement
	21	Trigger counter start/stop/clear/refresh/capture/increment/decrement
	22	Trigger counter start/stop/reset/refresh/capture/increment/decrement
	23	Trigger counter start/stop/reset/refresh/capture/increment/decrement
Timer4	24	Trigger counter start/stop/clear/capture
Event Port	25	Triggering Event Port1&2 action
	26	Triggering Event Port3&4 action
Timer0	27	Trigger counter start/stop/clear/capture
Timer2	28	Trigger counter start/stop/reset/capture
HASH	29	DMA completes a data block transmission and triggers a HASH operation

Modules	SN	Action
	30	DMA completes the transmission of all data blocks and triggers the HASH operation
TimerA	31	Trigger counter unit 1/5 start/stop/clear/increment/decrement Trigger counter unit 2/6 input capture
	32	Trigger counter unit 2/6 start/stop/clear/increment/decrement Trigger counter unit 1/5 input capture
	33	Trigger counter unit 3 start/stop/clear/increment/decrement Trigger Counter Unit 4 Input Capture
	34	Trigger counter unit 4 start/stop/clear/increment/decrement Trigger Counter Unit 3 Input Capture
OTS	35	Trigger to start temperature measurement
ADC_1	36, 37	Trigger sequence A/B starting analog-to-digital conversion
ADC_2	38, 39	Trigger sequence A/B starting analog-to-digital conversion
ADC_3	40, 41	Trigger sequence A/B starting analog-to-digital conversion

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). Used to implement the function of multiple AOS sources triggering the same AOS target. When using, first enable the public 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 public 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.

11.3 Register Description

11.3.1 Register List

Table 11-2 Register List

Register base address: 0x40010800

Register Name	Symbol	Offset Address	Bit Width	Initial value
Peripheral Trigger Event Register	INTSFTRTG	0x00	32	0x00000000
DCU trigger source select register	DCU_TRGSELx (x=1~3)	0x04,0x08,0x0C	32	0x000001FF
DMA1 Transmit start trigger source select register	DMA1_TRGSELx (x=0~7)	0x10,0x14,0x18,0x1C,0x20,0x24,0x28,0x2C	32	0x000001FF
DMA2 Transmit start trigger source select register	DMA2_TRGSELx (x=0~7)	0x30,0x34,0x38,0x3C,0x40,0x44,0x48,0x4C	32	0x000001FF
DMA channel reset trigger source select register	DMA_RC_TRGSEL	0x50	32	0x000001FF
Timer6 hardware trigger event selection register	TMR6_TRGSELx (x=0~3)	0x54,0x58,0x5C,0x60	32	0x000001FF
Time4 hardware trigger event selection register	TMR4_TRGSEL	0x64	32	0x000001FF
Event Port1,2 trigger source selection register	PEVNT_TRGSEL12	0x68	32	0x000001FF
Event Port3,4 trigger source selection register	PEVNT_TRGSEL34	0x6C	32	0x000001FF
Timer0 hardware trigger event selection register	TMR0_TRGSEL	0x70	32	0x000001FF
Timer2 hardware trigger event selection register	TMR2_TRGSEL	0x74	32	0x000001FF
HASH hardware trigger event select register A	HASH_TRGSELA	0x7C	32	0x000001FF
HASH Hardware Trigger Event Selection Register B	HASH_TRGSELB	0x78	32	0x000001FF
TimerA Hardware Trigger Event Selection Register	TMRA_TRGSELx (x=0~3)	0x80,0x84,0x88,0x8C	32	0x000001FF
OTS trigger source selection register	OTS_TRGSEL	0x90	32	0x000001FF
ADC1 on-chip trigger source select register	ADC1_TRGSELx (x=0,1)	0x94,0x98	32	0x000001FF
ADC2 on-chip trigger source select register	ADC2_TRGSELx (x=0,1)	0x9C,0xA0	32	0x000001FF
ADC3 on-chip trigger source select register	ADC3_TRGSELx (x=0,1)	0xA4,0xA8	32	0x000001FF
Common trigger source selection register 1	AOS_COMTRGSEL1	0xAC	32	0x000001FF
Common trigger source selection register 2	AOS_COMTRGSEL2	0xB0	32	0x000001FF

11.3.2 Peripheral Trigger Event Register (INTSFTTRG)

Each time a “1” is written to this register, a trigger event will be generated.

Offset address: 0x00

Reset value: 0x00000000

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

Bit	Flag	Bit Name	Function	Read and write
b31~b1	Reserved	-	Read at “0”, write at “0”	R/W
b0	SFTG	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

11.3.3 DCU Trigger Source Selection Register (DCU_TRGSELx) (x=1~3)

After the DCU selects the hardware trigger start mode, please write the number of the trigger event into this register. When a peripheral event corresponding to the number occurs, the DCU will be triggered by the event to start and perform calculations.

Each DCU unit corresponds to a register.

Offset address: 0x04, 0x08, 0x0C

Reset value: 0x000001FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN [1:0]		Reserved													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved							TRGSEL [8:0]								

Bit	Flag	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	R/W
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	R/W
b29~b9	Reserved	-	Read at “0”, write at “0”	R/W
b8~b0	TRGSEL [8:0]	Trigger source selection	Write the number of the peripheral circuit event to be selected	R/W

11.3.4 DMA1 Transfer Start Trigger Source Selection Register (DMA1_TRGSELx) (x=0~7)

After DMA1 is enabled, the number of the trigger event is written into this register. When the peripheral event corresponding to the number occurs, DMA1 will be triggered by the event to start and perform transmission.

Each data transmission channel corresponds to a register.

Offset address: 0x10 , 0x14 , 0x18 , 0x1C , x20 , 0x24 , 0x28 , 0x2C

Reset value: 0x000001FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN [1:0]		Reserved													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved							TRGSEL [8:0]								
Bit	Flag	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									R/W
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									R/W
b29~b9	Reserved	-				Read at “0”, write at “0”									R/W
b8~b0	TRGSEL [8:0]	Trigger source selection				Write the number of the peripheral circuit event to be selected									R/W

11.3.5 DMA2 Transfer Start Trigger Source Selection Register (DMA2_TRGSELx) (x=0~7)

After DMA2 is enabled, the number of the trigger event is written into this register. When the peripheral event corresponding to the number occurs, DMA2 will be triggered by the event to start and perform transmission.

Each data transmission channel corresponds to a register.

Offset address: 0x30 , 0x34 , 0x38 , 0x3C , 0x40 , 0x44 , 0x48 , 0x4C

Reset value: 0x000001FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN [1:0]		Reserved													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved							TRGSEL [8:0]								

Bit	Flag	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	R/W
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	R/W
b29~b9	Reserved	-		Read at "0", write at "0"	R/W
b8~b0	TRGSEL [8:0]	Trigger selection	source	Write the number of the peripheral circuit event to be selected	R/W

11.3.6 DMA Channel Reset Trigger Source Selection Register (DMA_RC_TRGSEL)

After DMA is enabled, the number of the trigger event is written into this register. When the peripheral event corresponding to the number occurs, DMA will be triggered by the event to reset the channel.

DMA1 and DMA2 share this register.

Offset address: 0x50

Reset value: 0x000001FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN [1:0]		Reserved													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved							TRGSEL [8:0]								

Bit	Flag	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	R/W
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	R/W
b29~b9	Reserved	-		Read at "0", write at "0"	R/W
b8~b0	TRGSEL [8:0]	Trigger selection	source	Write the number of the peripheral circuit event to be selected	R/W

11.3.7 Timer6 Hardware Trigger Event Selection Register (TMR6_TRGSELx) (x=0~3)

After Timer6 selects internal trigger hardware synchronization, write the trigger event number into this register. When the peripheral circuit event corresponding to the number occurs, Timer6 will be triggered by the event to synchronize hardware (including start / stop / clear / refresh / increment / decrement / input capture).

All Timer6 units share these 4 registers.

Offset address: 0x54, 0x58, 0x5C, 0x60

Reset value: 0x000001FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN [1:0]		Reserved													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved							TRGSEL [8:0]								

Bit	Flag	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	R/W
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	R/W
b29~b9	Reserved	-	Read at “0”, write at “0”	R/W
b8~b0	TRGSEL [8:0]	Trigger selection	source Write the number of the peripheral circuit event to be selected	R/W

11.3.8 Timer4 Hardware Trigger Event Selection Register (TMR4_TRGSEL)

After Timer4 selects internal trigger hardware synchronization, write the trigger event number into this register. When the peripheral circuit event corresponding to the number occurs, Timer4 will be started by the hardware synchronization triggered by the event.

Offset address: 0x64

Reset value: 0x000001FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN [1:0]		Reserved													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved							TRGSEL [8:0]								

Bit	Flag	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	R/W
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	R/W
b29~b9	Reserved	-		Read at "0", write at "0"	R/W
b8~b0	TRGSEL [8:0]	Trigger selection	source	Write the number of the peripheral circuit event to be selected	R/W

11.3.9 Event Port Trigger Source Select Register (PEVNT_TRGSEL12, PEVNT_TRGSEL34)

Write the trigger event number into this register. When the peripheral event corresponding to the number occurs, the Event Port is triggered to output the specified level or latch the port status. Event Port1 and 2 share the PEVNT_TRGSEL12 register, and Event Port3 and 4 share the PEVNT_TRGSEL34 register.

Offset address: 0x68, 0x6C

Reset value: 0x000001FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN [1:0]		Reserved													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved							TRGSEL [8:0]								

Bit	Flag	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	R/W
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	R/W
b29~b9	Reserved	-		Read at "0", write at "0"	R/W
b8~b0	TRGSEL [8:0]	Trigger selection	source	Write the number of the peripheral circuit event to be selected	R/W

11.3.10 Timer0 Hardware Trigger Event Selection Register (TMR0_TRGSEL)

After selecting hardware trigger in Timer0, write the trigger event number into this register. When the peripheral event corresponding to the number occurs, Timer0 will be triggered by the event to take action (including start/stop/clear/input capture).

All Timer0 units and their channels share this register.

Offset address: 0x70

Reset value: 0x000001FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN [1:0]		Reserved													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved							TRGSEL [8:0]								

Bit	Flag	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	R/W
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	R/W
b29~b9	Reserved	-		Read at “0”, write at “0”	R/W
b8~b0	TRGSEL [8:0]	Trigger selection	source	Write the number of the peripheral circuit event to be selected	R/W

11.3.11 Timer2 Hardware Trigger Event Selection Register (TMR2_TRGSEL)

After Timer2 selects internal hardware trigger, write the trigger event number into this register. When the peripheral event corresponding to the number occurs, Timer2 will be triggered by the event to take action (including start/stop/clear/input capture).

All Timer2 units and their channels share this register.

Offset address: 0x74

Reset value: 0x000001FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN [1:0]		Reserved													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved							TRGSEL [8:0]								
Bit	Flag	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										R/W	
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										R/W	
b29~b9	Reserved	-		Read at “0”, write at “0”										R/W	
b8~b0	TRGSEL [8:0]	Trigger selection	source	Write the number of the peripheral circuit event to be selected										R/W	

11.3.12 HASH Hardware Trigger Event Select Register A (HASH_TRGSELA)

When data is written to HASH_DR , the hardware event trigger source is selected through this register to start the HASH operation. Note: Do not select any other trigger signal except DMA_BTCx (x=0~7).

Offset address: 0x7C

Reset value: 0x000001FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN [1:0]		Reserved													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved							TRGSEL [8:0]								

Bit	Flag	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	R/W
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	R/W
b29~b9	Reserved	-		Read at "0", write at "0"	R/W
b8~b0	TRGSEL [8:0]	Trigger selection	source	Write the number of the peripheral circuit event to be selected	R/W

11.3.13 HASH Hardware Trigger Event Select Register B (HASH_TRGSELA)

When the last set of data is written to HASH_DR , the hardware event trigger source is selected through this register to start the HASH for the last operation. Note: Do not select any other trigger signal except DMA_BTCx (x=0~7).

Offset address: 0x78

Reset value: 0x000001FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN [1:0]		Reserved													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved							TRGSEL [8:0]								

Bit	Flag	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	R/W
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	R/W
b29~b9	Reserved	-		Read at "0", write at "0"	R/W
b8~b0	TRGSEL [8:0]	Trigger selection	source	Write the number of the peripheral circuit event to be selected	R/W

11.3.14 TimerA Hardware Trigger Event Selection Register (TMRA_TRGSELx, x = 0 ~ 3)

After TimerA selects internal hardware trigger, write the trigger event number into this register. When the peripheral event corresponding to the number occurs, TimerA will be triggered by the event to take action (including start / stop / clear / input capture).

All TimerA units share these 4 registers.

Offset address: 0x80, 0x84, 0x88, 0x8C

Reset value: 0x000001FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN [1:0]		Reserved													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved							TRGSEL [8:0]								

Bit	Flag	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	R/W
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	R/W
b29~b9	Reserved	-		Read at "0", write at "0"	R/W
b8~b0	TRGSEL [8:0]	Trigger selection	source	Write the number of the peripheral circuit event to be selected	R/W

11.3.15 OTS trigger source select register (OTS_TRGSEL)

After OTS selects hardware trigger, the number of the trigger event is written into this register. When the peripheral event corresponding to the number occurs, OTS will be triggered by the event to start temperature measurement.

Offset address: 0x90

Reset value: 0x000001FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN [1:0]		Reserved													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved							TRGSEL [8:0]								

Bit	Flag	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	R/W
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	R/W
b29~b9	Reserved	-		Read at "0", write at "0"	R/W
b8~b0	TRGSEL [8:0]	Trigger selection	source	Write the number of the peripheral circuit event to be selected	R/W

11.3.16 ADC1 Conversion On-chip Trigger Source Selection Register (ADC1_TRGSELx, x=0,1)

After ADC1 selects the on-chip trigger mode, the number of the trigger event is written into this register. When the peripheral event corresponding to the number occurs, ADC1 will be triggered by the event to perform analog-to-digital conversion.

ADC1_TRGSEL0, 1 registers correspond to sequences A, B respectively.

Offset address: 0x94, 0x98

Reset value: 0x000001FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	
COMEN [1:0]		Reserved														
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
Reserved							TRGSEL [8:0]									

Bit	Flag	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	R/W
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	R/W
b29~b9	Reserved	-		Read at “0”, write at “0”	R/W
b8~b0	TRGSEL [8:0]	Trigger selection	source	Write the number of the peripheral circuit event to be selected	R/W

11.3.17 ADC2 Conversion On-chip Trigger Source Selection Register (ADC2_TRGSELx, x=0,1)

After ADC2 selects the on-chip trigger mode, the number of the trigger event is written into 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_TRGSEL0, 1 registers correspond to sequences A, B respectively.

Offset address: 0x9C, 0xA0

Reset value: 0x000001FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN [1:0]		Reserved													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved							TRGSEL [8:0]								

Bit	Flag	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	R/W
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	R/W
b29~b9	Reserved	-		Read at "0", write at "0"	R/W
b8~b0	TRGSEL [8:0]	Trigger selection	source	Write the number of the peripheral circuit event to be selected	R/W

11.3.18 ADC3 Conversion On-chip Trigger Source Selection Register (ADC3_TRGSELx, x=0,1)

After ADC3 selects the on-chip trigger mode, the number of the trigger event is written into this register. When the peripheral event corresponding to the number occurs, ADC3 will be triggered by the event to perform analog-to-digital conversion.

ADC3_TRGSEL0, 1 registers correspond to sequences A, B respectively.

Offset address: 0xA4, 0xA8

Reset value: 0x000001FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
COMEN [1:0]		Reserved													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved							TRGSEL [8:0]								

Bit	Flag	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	R/W
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	R/W
b29~b9	Reserved	-		Read at "0", write at "0"	R/W
b8~b0	TRGSEL [8:0]	Trigger selection	source	Write the number of the peripheral circuit event to be selected	R/W

11.3.19 Common trigger source select register (AOS_COMTRGSEL1, 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.

Offset address: 0xAC, 0xB0

Reset value: 0x000001FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved							COMTRG [8:0]								
Bit	Flag	Bit Name		Function		Read and write									
b31~b9	Reserved	-		Read at “0”, write at “0”		R/W									
b8~b0	COMTRG [8:0]	Trigger selection	source	Write the number of the peripheral circuit event to be selected		R/W									

12 Memory Protection Unit (MPU)

12.1 Introduction

The MPU can provide protection to the memory, and can improve the security of the system by preventing unauthorized access.

This chip has built-in 1 MPU unit for CPU, 1 MPU unit for CPU main stack pointer, 1 MPU unit for CPU thread stack pointer, 3 MPU units 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
SystemDMA_1 MPU: SMPU1	Memory protection unit of system DMA_1 16 areas, 8 areas dedicated to system DMA, 8 areas shared by all DMAs
SystemDMA_2 MPU: SMPU2	Memory protection unit of system DMA_2 16 areas, 8 areas dedicated to system DMA, 8 areas shared by all DMAs
USBFS-DMA MPU: FMPU	Storage Protection Unit of USBFS-DMA 8 regions, shared by all DMAs
IPMPU	Access protection unit for system IP and security-related IP

Among them, the ARM MPU provides the access control of the CPU to the entire 4G address space, and the introduction is omitted.

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/ SMPU2/ FMPU respectively provide system DMA_1/system DMA_2/USBFS-DMA to control the read and write access rights of 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 Area Range Settings

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 32Byte~4GByte, the size must be 2^n Byte ($n=5\sim32$), 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

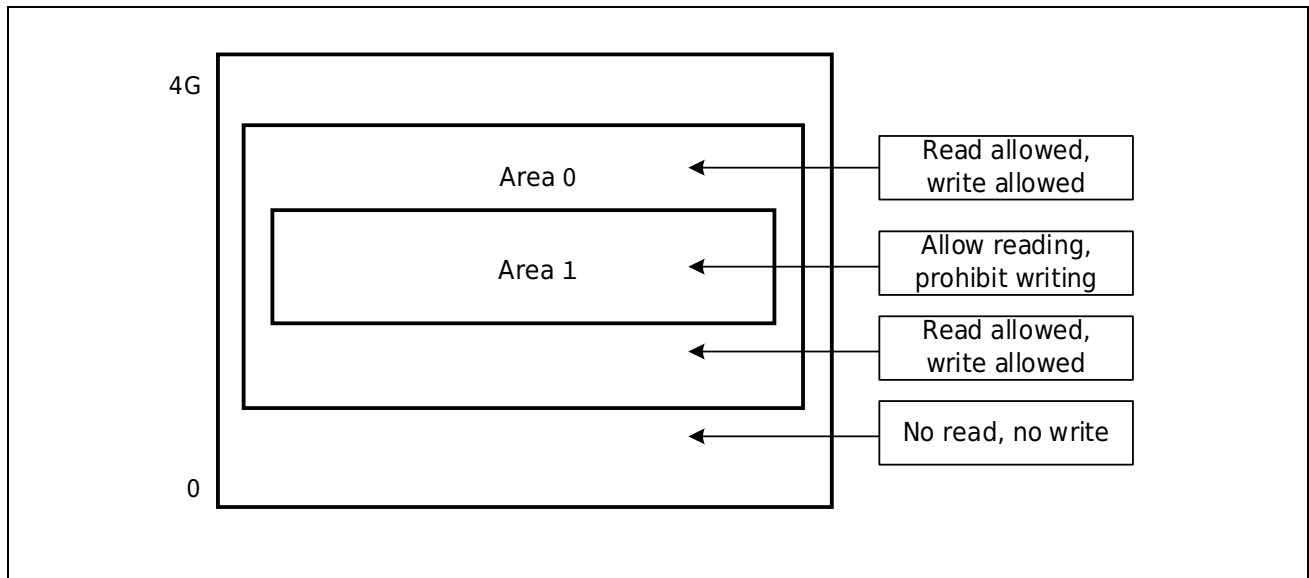
MSPMPU/ PSPMPU/ SMPU1/ SMPU2/ FMPU can be enabled independently.

It is recommended to enable the MPU after setting the area range/permission setting/action selection.

12.3 Application Instances

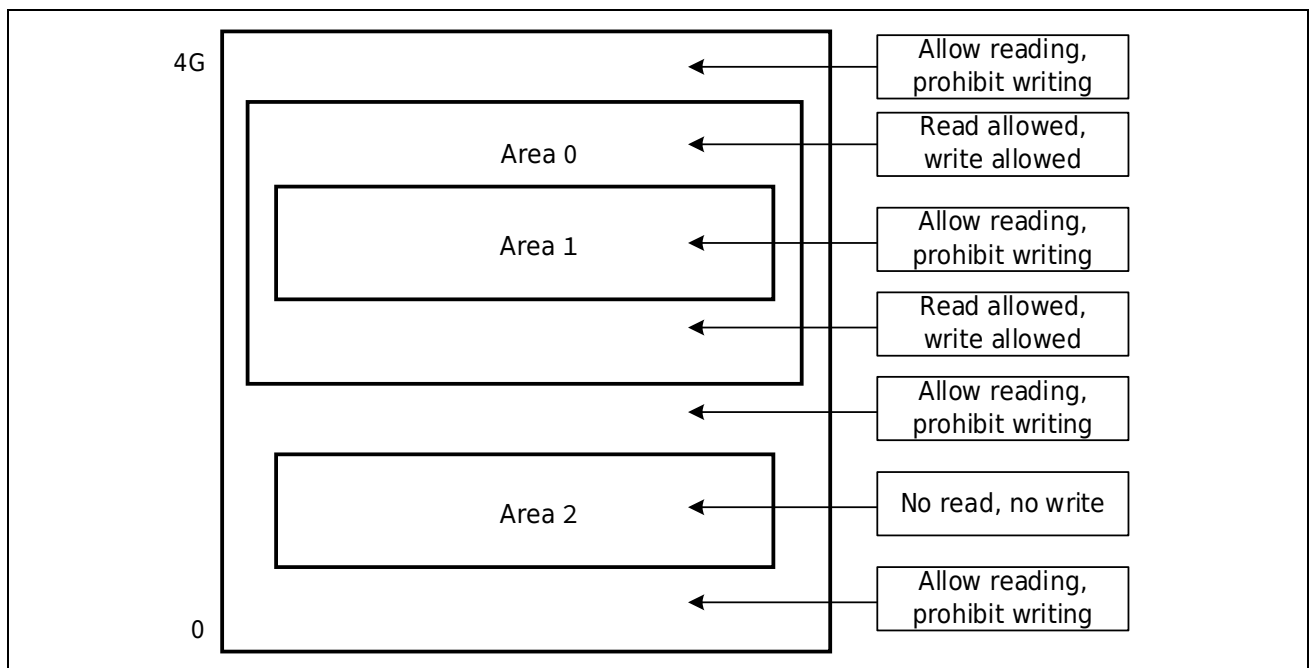
12.3.1 Only Enable Partial Space Access

Instance: 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.



12.3.2 Only some space access is prohibited

Instance: 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.



12.4 Register Description

The registers of this module can only be set by the CPU.

MPU base address: 0x40050000

Table 12-1 Register List

Offset Address	Register Name	Initial value	Name	Write protection
0x00~0x3C	MPU_RGD0~15	0x00000000	Area 0~15 range description register	MPUWE
0x40	MPU_SR	0x00000000	Status flag register	No
0x44	MPU_ECLR	0x00000000	Flag Clear Register	No
0x48	MPU_WP	0x00000000	Write protection register	WKEY
0x4C	MPU_IPPR	0x00000000	IP access protection register	MPUWE
0x50	MPU_MSPPBA	0x00000000	MSP Protection Start Address Register	MPUWE
0x54	MPU_MSPPCTL	0x00000003	MSP Protection Control Register	MPUWE
0x58	MPU_PSPPBA	0x00000000	PSP Protection Start Address Register	MPUWE
0x5C	MPU_PSPPCTL	0x00000003	PSP Protection Control Register	MPUWE
0x60	MPU_S1RGE	0x00000000	SMPU1 Region Enable Register	MPUWE
0x64	MPU_S1RGWP	0x00000000	SMPU1 Region Write Permission Register	MPUWE
0x68	MPU_S1RGRP	0x00000000	SMPU1 Region Read Permission Register	MPUWE
0x6C	MPU_S1CR	0x00000000	SMPU1 Control Register	MPUWE
0x70	MPU_S2RGE	0x00000000	SMPU2 Region Enable Register	MPUWE
0x74	MPU_S2RGWP	0x00000000	SMPU2 Region Write Permission Register	MPUWE
0x78	MPU_S2RGRP	0x00000000	SMPU2 Region Read Permission Register	MPUWE
0x7C	MPU_S2CR	0x00000000	SMPU2 Control Register	MPUWE
0x80	MPU_FRGE	0x00000000	FMPU Zone Enable Register	MPUWE
0x84	MPU_FRGWP	0x00000000	FMPU zone write authority register	MPUWE
0x88	MPU_FRGRP	0x00000000	FMPU zone read authority register	MPUWE
0x8C	MPU_FCR	0x00000000	FMPU control register	MPUWE

12.4.1 Area scope description register MPU_RGDn (n=0~15)

Reset value: 0x00000000

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	Flag	Bit Name		Function										Read and write	
b31~b5	MPURGnADDR [31:5]	Zone address	base	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										R/W	
				Set the size of the region n											
				00000~00011: Reserved, setting prohibited											
				00100: 32Byte											
b4~b0	MPURGnSIZE [4:0]	Area size		00101: 64Byte										R/W	
				-											
				11110: 2GByte											
				11111: 4GByte											

12.4.2 Status flag register MPU_SR

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved											MSPEF	PSPEF	FMPUEAF	SMPU2EAF	SMPU1EAF
Bit	Flag	Bit Name		Function										Read and write	
b31~b5	Reserved	-		Reserved bit, read as 0, write 0 when writing										R/W	
b4	MSPEF	MSP error flag		0: MSP pointer is within the set range 1: The MSP pointer is out of the setting range										R	
b3	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	
b2	FMPUEAF	FMPU error flag		0: USBFS-DMA has no error access 1: USBFS-DMA has a wrong access										R	
b1	SMPU2EAF	SMPU2 Error Flags		0: System DMA_2 has no error access 1: System DMA_2 has an error access										R	
b0	SMPU1EAF	SMPU1 error flag		0: System DMA_1 has no error access 1: System DMA_1 has an error access										R	

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: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved											MSP ECLR	PSP ECLR	FMPU ECLR	SMPU2 ECLR	SMPU1E CLR
Bit	Flag	Bit Name		Function		Read and write									
b31~b5	Reserved	-		Reserved bit, read as 0, write 0 when writing		R/W									
b4	MSPECLR	MSP error flag cleared		Writing 1 can clear MSPEF to 0		R/W									
b3	PSPECLR	PSP error flag cleared		Writing 1 can clear PSPEF to 0		R/W									
b2	FMPUECLR	FMPU error flag cleared		Writing 1 can clear FMPUEAF to 0		R/W									
b1	SMPU2ECLR	SMPU2 error flag cleared		Writing 1 can clear SMPU2EAF to 0		R/W									
b0	SMPU1ECLR	SMPU1 error flag cleared		Writing 1 can clear SMPU1EAF to 0		R/W									

The read value of this register is fixed at 0x00000000.

12.4.4 Write protection register MPU_WP

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
WKEY [15:1]															MPUWE
Bit	Flag	Bit Name		Function		Read and write									
b31~b16	Reserved	-		Reserved bit, read as 0, write 0 when writing		R/W									
b15~b1	WKEY [15:1]	Write code		When writing to MPUWE, it must be written to WKEY at the same time 15'b1001_0110_1010_010 , read as 0		R/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									

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: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
BUSERRE	-	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	-	-	EFMWWRP	EFMRDP	CRCWRP	CRCRDP	TRNGWRP	TRNGRDP	HASHWRP	HASHRDP	AESWRP	AESRDP

Bit	Flag	Bit Name	Function	Read and write
b31	BUSERRE	Bus enable error	0: Ignore access to protected IP 1: When an access to the protected IP occurs, a bus error is returned	R/W
b30	Reserved	-	Reserved bit, read as 0, write 0 when writing	R/W
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	R/W
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	R/W
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	R/W
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	R/W
b25	INTCWRP	INTC protection write	0: Allow write operation to INTC 1: Prohibit writing to INTC	R/W
b24	INTCRDP	INTC protection read	0: Allow read operation on INTC 1: Disable the read operation of INTC	R/W
b23	SRAMCWRP	SRAMC protection write	0: Enable writing to SRAMC 1: Forbid writing to SRAMC	R/W
b22	SRAMCRDP	SRAMC protection read	0: Enable read operations on SRAMC 1: Disable read operation on SRAMC	R/W
b21	DMPUWRP	DMPU protection write	0: Enable writing to DMPU 1: Disable writing to DMPU	R/W
b20	DMPURDP	DMPU protection read	0: Enable read operations on DMPU 1: Disable DMPU read operation	R/W
b19	RTCWRP	RTC protection write	0: Enable writing to RTC 1: Disable writing to RTC	R/W
b18	RTCRDP	RTC protection read	0: Enable read operations on RTC 1: Disable read operation on RTC	R/W
b17	BKSRAMWRP	BKSRAM protection write	0: Allow writing to Ret SRAM 1: Disable writing to Ret SRAM	R/W
b16	BKSRAMRDP	BKSRAM protection read	0: Allow read operations on Ret SRAM 1: Disable read operation on Ret SRAM	R/W
b15	SWDTWRP	SWDT protection write	0: Allow writing to SWDT 1: Disable writing to SWDT	R/W
b14	SWDTRDP	SWDT protection read	0: Enable read operations on SWDT 1: Disable read operation on SWDT	R/W
b13	WDTWRP	WDT protection write	0: Enable writing to WDT 1: Disable writing to WDT	R/W
b12	WDTRDP	WDT protection read	0: Enable read operations on WDT 1: Disable read operation on WDT	R/W
b11-b10	Reserved	-	Reserved bit, read as 0, write 0 when writing	R/W
b9	EFMWWRP	EFM protection write	0: Enable writing to EFM 1: Disable writing to EFM	R/W
b8	EFMRDP	EFM protection read	0: Enable read operations on EFM 1: Disable read operation on EFM	R/W
b7	CRCWRP	CRC protection write	0: Enable writing to CRC 1: Disable writing to CRC	R/W
b6	CRCRDP	CRC protection read	0: Allow reading of CRC 1: Disable CRC read operation	R/W
b5	TRNGWRP	TRNG protection write	0: Enable writing to TRNG 1: Disable writing to TRNG	R/W
b4	TRNGRDP	TRNG protection read	0: Enable read operations on TRNG 1: Disable read operation on TRNG	R/W

b3	HASHWRP	HASH protection	write	0: Enable writing to HASH 1: Forbid writing to HASH	R/W
b2	HASHRDP	HASH protection	read	0: Allow read operations on HASH 1: Disable read operation on HASH	R/W
b1	AESWRP	AES protection	write	0: Enable writing to AES 1: Disable writing to AES	R/W
b0	AESRDP	AES protection	read	0: Allow AES read operation 1: Disable AES read operation	R/W

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: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
MSPPPBA [31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MSPPPBA [15:2]														Reserved	
Bit		Flag		Bit Name		Function								Read and write	
b31~b2		MSPPPBA [31:2]		Allow range start address		Set the MSP pointer to allow the start address of the range								R/W	
b1~b0		Reserved		-		Reserved bit, read as 0, write 0 when writing								R/W	

12.4.7 MSP Protection Control Register (MPU_MSPPCTL)

Reset value: 0x00000003

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
MSPPE	MSPPACT	Reserved													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MSPPSIZE [15:2]														Reserved	
Bit		Flag		Bit Name		Function								Read and write	
b31		MSPPE		MSP protection enable		0: MSP pointer protection invalid 1: MSP pointer protection valid								R/W	
b30		MSPPACT		MSP protection action selection		0: Nomaskable interrupt 1: Reset									
b29~b16		Reserved		-		Reserved bit, read as 0, write 0 when writing								R/W	
b15~b2		MSPPSIZE [15:2]		Allowable range size		Set the allowed range of the MSP pointer, the setting value is 4Byte to 64Kbyte When MSPPSIZE=0x0, the MSP pointer allows a range of 4 bytes. When MSPPSIZE=0x3FFF, the MSP pointer allows a range of 64 KBytes.								R/W	
b1~b0		Reserved		-		Reserved bit, read as 1, write 1 when writing								R/W	

12.4.8 PSP Protection Start Address Register (MPU_PSPBPA)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
PSPBPA [31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PSPBPA [15:2]														Reserved	
Bit		Flag		Bit Name		Function								Read and write	
b31~b2		PSPBPA [31:2]		Allow range start address		Set the PSP pointer to allow the start address of the range								R/W	
b1~b0		Reserved		-		Reserved bit, read as 0, write 0 when writing								R/W	

12.4.9 PSP Protection Control Register (MPU_PSPCTL)

Reset value: 0x00000003

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
PSPPE	PSPPACT	Reserved													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PSPPSIZE [15:2]														Reserved	
Bit		Flag		Bit Name		Function								Read and write	
b31		PSPPE		PSP protection enable		0: PSP pointer protection disable 1: PSP pointer protection valid								R/W	
b30		PSPPACT		PSP protection action selection		0: Nomaskable interrupt 1: Reset									
b29~b16		Reserved		-		Reserved bit, read as 0, write 0 when writing								R/W	
b15~b2		PSPPSIZE [15:2]		Allowable range size		Set the PSP pointer allowed range, setting value 4Byte~64Kbyte When PSPPSIZE=0x0, the PSP pointer allows a range of 4 bytes. When PSPPSIZE=0x3FFF, the PSP pointer allows a range of 64 KBytes.								R/W	
b1~b0		Reserved		-		Reserved bit, read as 1, write 1 when writing								R/W	

12.4.10 SMPU1 Region Enable Register (MPU_S1RGE)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
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	Flag	Bit Name		Function		Read and write									
b31~b16	Reserved	-		Reserved bit, read as 0, write 0 when writing		R/W									
b15	S1RG15E	SMPU1 enable	region	15	0: Area 15 of SMPU1 invalid 1: Area 15 of SMPU1 valid	R/W									
b14	S1RG14E	SMPU1 enable	region	14	0: Area 14 of SMPU1 invalid 1: Area 14 of SMPU1 valid	R/W									
b13	S1RG13E	SMPU1 enable	region	13	0: Area 13 of SMPU1 invalid 1: Area 13 of SMPU1 valid	R/W									
b12	S1RG12E	SMPU1 enable	region	12	0: Area 12 of SMPU1 invalid 1: Area 12 of SMPU1 valid	R/W									
b11	S1RG11E	SMPU1 enable	region	11	0: Area 11 of SMPU1 invalid 1: Area 11 of SMPU1 valid	R/W									
b10	S1RG10E	SMPU1 enable	region	10	0: Area 10 of SMPU1 invalid 1: Area 10 of SMPU1 valid	R/W									
b9	S1RG9E	SMPU1 enable	region	9	0: Area 9 of SMPU1 invalid 1: Area 9 of SMPU1 valid	R/W									
b8	S1RG8E	SMPU1 enable	region	8	0: Area 8 of SMPU1 invalid 1: Area 8 of SMPU1 valid	R/W									
b7	S1RG7E	SMPU1 enable	region	7	0: Area 7 of SMPU1 invalid 1: Area 7 of SMPU1 valid	R/W									
b6	S1RG6E	SMPU1 enable	region	6	0: Area 6 of SMPU1 invalid 1: Area 6 of SMPU1 valid	R/W									
b5	S1RG5E	SMPU1 enable	region	5	0: Area 5 of SMPU1 invalid 1: Area 5 of SMPU1 valid	R/W									
b4	S1RG4E	SMPU1 enable	region	4	0: Area 4 of SMPU1 invalid 1: Area 4 of SMPU1 valid	R/W									
b3	S1RG3E	SMPU1 enable	region	3	0: Area 3 of SMPU1 invalid 1: Area 3 of SMPU1 valid	R/W									
b2	S1RG2E	SMPU1 enable	region	2	0: Area 2 of SMPU1 invalid 1: Area 2 of SMPU1 valid	R/W									
b1	S1RG1E	SMPU1 enable	region	1	0: Area 1 of SMPU1 invalid 1: Area 1 of SMPU1 valid	R/W									
b0	S1RG0E	SMPU1 enable	region	0	0: Area 0 of SMPU1 invalid 1: Area 0 of SMPU1 valid	R/W									

12.4.11 SMPU1 Region Write Permission Register (MPU_S1RGWP)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
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	Flag	Bit Name		Function		Read and write									
b31~b16	Reserved	-		Reserved bit, read as 0, write 0 when writing		R/W									
b15	S1RG15WP	SMPU1 area 15 write permission		0: zone 15 allows system DMA 1 write 1: Area 15 prohibits system DMA 1 writing		R/W									
b14	S1RG14WP	SMPU1 zone 14 write enable		0: zone 14 allows system DMA 1 write 1: Area 14 prohibits system DMA 1 writing		R/W									
b13	S1RG13WP	SMPU1 zone 13 write enable		0: zone 13 allows system DMA 1 write 1: Area 13 prohibits system DMA 1 writing		R/W									
b12	S1RG12WP	SMPU1 zone 12 write enable		0: zone 12 allows system DMA 1 write 1: Area 12 prohibits system DMA 1 writing		R/W									
b11	S1RG11WP	SMPU1 zone 11 write enable		0: zone 11 allows system DMA 1 write 1: Area 11 prohibits system DMA 1 writing		R/W									
b10	S1RG10WP	SMPU1 zone 10 write enable		0: zone 10 allows system DMA 1 write 1: Area 10 prohibits system DMA 1 writing		R/W									
b9	S1RG9WP	SMPU1 zone 9 write enable		0: zone 9 allows system DMA 1 write 1: Area 9 prohibits system DMA 1 writing		R/W									
b8	S1RG8WP	SMPU1 zone 8 write enable		0: zone 8 allows system DMA 1 write 1: Area 8 prohibits system DMA 1 writing		R/W									
b7	S1RG7WP	SMPU1 zone 7 write enable		0: zone 7 allows system DMA 1 write 1: Area 7 prohibits system DMA 1 writing		R/W									
b6	S1RG6WP	SMPU1 zone 6 write enable		0: zone 6 allows system DMA 1 write 1: Area 6 prohibits system DMA 1 writing		R/W									
b5	S1RG5WP	SMPU1 zone 5 write enable		0: zone 5 allows system DMA 1 write 1: Area 5 prohibits system DMA 1 writing		R/W									
b4	S1RG4WP	SMPU1 zone 4 write enable		0: zone 4 allows system DMA 1 write 1: Area 4 prohibits system DMA 1 writing		R/W									
b3	S1RG3WP	SMPU1 zone 3 write enable		0: zone 3 allows system DMA 1 write 1: Area 3 prohibits system DMA 1 writing		R/W									
b2	S1RG2WP	SMPU1 zone 2 write enable		0: zone 2 allows system DMA 1 write 1: Area 2 prohibits system DMA 1 writing		R/W									
b1	S1RG1WP	SMPU1 zone 1 write enable		0: zone 1 allows system DMA 1 write 1: Area 1 prohibits system DMA 1 writing		R/W									
b0	S1RG0WP	SMPU1 zone 0 write enable		0: zone 0 allows system DMA 1 write 1: Area 0 prohibits system DMA 1 writing		R/W									

12.4.12 SMPU1 Region Read Permission Register MPU_S1RGRP

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
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	Flag	Bit Name		Function		Read and write									
b31~b16	Reserved	-		Reserved bit, read as 0, write 0 when writing		R/W									
b15	S1RG15RP	SMPU1 area 15 read permission		0: zone 15 allows system DMA 1 read 1: Area 15 prohibits system DMA 1 from reading		R/W									
b14	S1RG14RP	SMPU1 zone 14 read permission		0: zone 14 allows system DMA 1 read 1: Area 14 prohibits system DMA 1 from reading		R/W									
b13	S1RG13RP	SMPU1 zone 13 read permission		0: zone 13 allows system DMA 1 read 1: Area 13 prohibits system DMA 1 from reading		R/W									
b12	S1RG12RP	SMPU1 zone 12 read permission		0: zone 12 allows system DMA 1 read 1: Area 12 prohibits system DMA 1 from reading		R/W									
b11	S1RG11RP	SMPU1 zone 11 read permission		0: zone 11 allows system DMA 1 read 1: Area 11 prohibits system DMA 1 from reading		R/W									
b10	S1RG10RP	SMPU1 zone 10 read permission		0: zone 10 allows system DMA 1 read 1: Area 10 prohibits system DMA 1 from reading		R/W									
b9	S1RG9RP	SMPU1 zone 9 read permission		0: zone 9 allows system DMA 1 read 1: Area 9 prohibits system DMA 1 from reading		R/W									
b8	S1RG8RP	SMPU1 zone 8 read permission		0: zone 8 allows system DMA 1 read 1: Area 8 prohibits system DMA 1 from reading		R/W									
b7	S1RG7RP	SMPU1 zone 7 read permission		0: zone 7 allows system DMA 1 read 1: Area 7 prohibits system DMA 1 from reading		R/W									
b6	S1RG6RP	SMPU1 zone 6 read permission		0: zone 6 allows system DMA 1 read 1: Area 6 prohibits system DMA 1 from reading		R/W									
b5	S1RG5RP	SMPU1 zone 5 read permission		0: zone 5 allows system DMA 1 read 1: Area 5 prohibits system DMA 1 from reading		R/W									
b4	S1RG4RP	SMPU1 zone 4 read permission		0: zone 4 allows system DMA 1 read 1: Area 4 prohibits system DMA 1 from reading		R/W									
b3	S1RG3RP	SMPU1 zone 3 read permission		0: zone 3 allows system DMA 1 read 1: Area 3 prohibits system DMA 1 from reading		R/W									
b2	S1RG2RP	SMPU1 zone 2 read permission		0: zone 2 allows system DMA 1 read 1: Area 2 prohibits system DMA 1 from reading		R/W									
b1	S1RG1RP	SMPU1 zone 1 read permission		0: zone 1 allows system DMA 1 read 1: Area 1 prohibits system DMA 1 from reading		R/W									
b0	S1RG0RP	SMPU1 zone 0 read permission		0: zone 0 allows system DMA 1 read 1: Area 0 prohibits system DMA 1 from reading		R/W									

12.4.13 SMPU1 Control Register MPU_S1CR

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								SMPU1 E	Reserved			SMPU1ACT [1:0]		SMPU1B WP	SMPU1 BRP
Bit	Flag		Bit Name		Function									Read and write	
b31~b8	Reserved		-		Reserved bit, read as 0, write 0 when writing									R/W	
b7	SMPU1E		SMPU1 enable		0: SMPU1 invalid 1: SMPU1 effective									R/W	
b6~b4	Reserved		-		Reserved bit, read as 0, write 0 when writing									R/W	
b3~b2	SMPU1ACT [1:0]		SMPU1 action selection		Set the action when the system DMA_1 is prohibited from accessing 00: ignore (read access reads 0, write access ignore) 01: Ignore + DMA bus error 10: ignore + unmaskable interrupt 11: reset									R/W	
b1	SMPU1BWP		SMPU1 write setting	background permission	0: SMPU1 background space allows system DMA_1 to write 1: SMPU1 background space prohibits system DMA_1 from writing									R/W	
b0	SMPU1BRP		SMPU1 read permission setting	background	0: SMPU1 background space allows system DMA_1 to read 1: SMPU1 background space prohibits system DMA_1 from reading									R/W	

When multiple area settings overlap, the priority is: setting prohibited > setting allowed.

12.4.14 SMPU2 Region Enable Register (MPU_S2RGE)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
S2RG15E	S2RG14E	S2RG13E	S2RG12E	S2RG11E	S2RG10E	S2RG9E	S2RG8E	S2RG7E	S2RG6E	S2RG5E	S2RG4E	S2RG3E	S2RG2E	S2RG1E	S2RG0E
Bit	Flag	Bit Name		Function		Read and write									
b31~b16	Reserved	-		Reserved bit, read as 0, write 0 when writing		R/W									
b15	S2RG15E	SMPU2 enable	region	15	0: Area 15 of SMPU2 invalid 1: Area 15 of SMPU2 valid	R/W									
b14	S2RG14E	SMPU2 enable	region	14	0: Area 14 of SMPU2 invalid 1: Area 14 of SMPU2 valid	R/W									
b13	S2RG13E	SMPU2 enable	region	13	0: Area 13 of SMPU2 invalid 1: Area 13 of SMPU2 valid	R/W									
b12	S2RG12E	SMPU2 enable	region	12	0: Area 12 of SMPU2 invalid 1: Area 12 of SMPU2 valid	R/W									
b11	S2RG11E	SMPU2 enable	region	11	0: Area 11 of SMPU2 invalid 1: Area 11 of SMPU2 valid	R/W									
b10	S2RG10E	SMPU2 enable	region	10	0: Area 10 of SMPU2 invalid 1: Area 10 of SMPU2 valid	R/W									
b9	S2RG9E	SMPU2 enable	region	9	0: Area 9 of SMPU2 invalid 1: Area 9 of SMPU2 valid	R/W									
b8	S2RG8E	SMPU2 enable	region	8	0: Area 8 of SMPU2 invalid 1: Area 8 of SMPU2 valid	R/W									
b7	S2RG7E	SMPU2 enable	region	7	0: Area 7 of SMPU2 invalid 1: Area 7 of SMPU2 valid	R/W									
b6	S2RG6E	SMPU2 enable	region	6	0: Area 6 of SMPU2 invalid 1: Area 6 of SMPU2 valid	R/W									
b5	S2RG5E	SMPU2 enable	region	5	0: Area 5 of SMPU2 invalid 1: Area 5 of SMPU2 valid	R/W									
b4	S2RG4E	SMPU2 enable	region	4	0: Area 4 of SMPU2 invalid 1: Area 4 of SMPU2 valid	R/W									
b3	S2RG3E	SMPU2 enable	region	3	0: Area 3 of SMPU2 invalid 1: Area 3 of SMPU2 valid	R/W									
b2	S2RG2E	SMPU2 enable	region	2	0: Area 2 of SMPU2 invalid 1: Area 2 of SMPU2 valid	R/W									
b1	S2RG1E	SMPU2 enable	region	1	0: Area 1 of SMPU2 invalid 1: Area 1 of SMPU2 valid	R/W									
b0	S2RG0E	SMPU2 enable	region	0	0: Area 0 of SMPU2 invalid 1: Area 0 of SMPU2 valid	R/W									

12.4.15 SMPU2 Region Write Permission Register (MPU_S2RGWP)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
S2RG15 WP	S2RG14 WP	S2RG13 WP	S2RG12 WP	S2RG11 WP	S2RG10 WP	S2RG9 WP	S2RG8 WP	S2RG7 WP	S2RG6 WP	S2RG5 WP	S2RG4 WP	S2RG3 WP	S2RG2 WP	S2RG1 WP	S2RG0 WP

Bit	Flag	Bit Name	Function	Read and write
b31~b16	Reserved	-	Reserved bit, read as 0, write 0 when writing	R/W
b15	S2RG15WP	SMPU2 area 15 write permission	0: zone 15 allows system DMA_2 write 1 : Area 15 prohibits system DMA_2 writing	R/W
b14	S2RG14WP	SMPU2 zone 14 write enable	0: zone 14 allows system DMA_2 write 1 : Area 14 prohibits system DMA_2 writing	R/W
b13	S2RG13WP	SMPU2 zone 13 write enable	0: zone 13 allows system DMA_2 write 1 : Area 13 prohibits system DMA_2 writing	R/W
b12	S2RG12WP	SMPU2 zone 12 write enable	0: zone 12 allows system DMA_2 write 1 : Area 12 prohibits system DMA_2 writing	R/W
b11	S2RG11WP	SMPU2 zone 11 write enable	0: zone 11 allows system DMA_2 write 1 : Area 11 prohibits system DMA_2 writing	R/W
b10	S2RG10WP	SMPU2 zone 10 write enable	0: zone 10 allows system DMA_2 write 1 : Area 10 prohibits system DMA_2 writing	R/W
b9	S2RG9WP	SMPU2 zone 9 write enable	0: zone 9 allows system DMA_2 write 1 : Area 9 prohibits system DMA_2 writing	R/W
b8	S2RG8WP	SMPU2 zone 8 write enable	0: zone 8 allows system DMA_2 write 1 : Area 8 prohibits system DMA_2 writing	R/W
b7	S2RG7WP	SMPU2 zone 7 write enable	0: zone 7 allows system DMA_2 write 1 : Area 7 prohibits system DMA_2 writing	R/W
b6	S2RG6WP	SMPU2 zone 6 write enable	0: zone 6 allows system DMA_2 write 1 : Area 6 prohibits system DMA_2 writing	R/W
b5	S2RG5WP	SMPU2 zone 5 write enable	0: zone 5 allows system DMA_2 write 1 : Area 5 prohibits system DMA_2 writing	R/W
b4	S2RG4WP	SMPU2 zone 4 write enable	0: zone 4 allows system DMA_2 write 1 : Area 4 prohibits system DMA_2 writing	R/W
b3	S2RG3WP	SMPU2 zone 3 write enable	0: zone 3 allows system DMA_2 write 1 : Area 3 prohibits system DMA_2 writing	R/W
b2	S2RG2WP	SMPU2 zone 2 write enable	0: zone 2 allows system DMA_2 write 1 : Area 2 prohibits system DMA_2 writing	R/W
b1	S2RG1WP	SMPU2 zone 1 write enable	0: zone 1 allows system DMA_2 write 1 : Area 1 prohibits system DMA_2 writing	R/W
b0	S2RG0WP	SMPU2 zone 0 write enable	0: zone 0 allows system DMA_2 write 1 : Area 0 prohibits system DMA_2 writing	R/W

12.4.16 SMPU2 Region Read Permission Register MPU_S2RGRP

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
S2RG15RP	S2RG14RP	S2RG13RP	S2RG12RP	S2RG11RP	S2RG10RP	S2RG9RP	S2RG8RP	S2RG7RP	S2RG6RP	S2RG5RP	S2RG4RP	S2RG3RP	S2RG2RP	S2RG1RP	S2RG0RP
Bit	Flag	Bit Name		Function		Read and write									
b31~b16	Reserved	-		Reserved bit, read as 0, write 0 when writing		R/W									
b15	S2RG15RP	SMPU2 area 15 read permission		0: zone 15 allows system DMA 2 read 1: Area 15 prohibits system DMA 2 from reading		R/W									
b14	S2RG14RP	SMPU2 zone 14 read permission		0: zone 14 allows system DMA 2 read 1: Area 14 prohibits system DMA 2 from reading		R/W									
b13	S2RG13RP	SMPU2 zone 13 read permission		0: zone 13 allows system DMA 2 read 1: Area 13 prohibits system DMA 2 from reading		R/W									
b12	S2RG12RP	SMPU2 zone 12 read permission		0: zone 12 allows system DMA 2 read 1: Area 12 prohibits system DMA 2 from reading		R/W									
b11	S2RG11RP	SMPU2 zone 11 read permission		0: zone 11 allows system DMA 2 read 1: Area 11 prohibits system DMA 2 from reading		R/W									
b10	S2RG10RP	SMPU2 zone 10 read permission		0: zone 10 allows system DMA 2 read 1: Area 10 prohibits system DMA 2 from reading		R/W									
b9	S2RG9RP	SMPU2 zone 9 read permission		0: zone 9 allows system DMA 2 read 1: Area 9 prohibits system DMA 2 from reading		R/W									
b8	S2RG8RP	SMPU2 zone 8 read permission		0: zone 8 allows system DMA 2 read 1: Area 8 prohibits system DMA 2 from reading		R/W									
b7	S2RG7RP	SMPU2 zone 7 read permission		0: zone 7 allows system DMA 2 read 1: Area 7 prohibits system DMA 2 from reading		R/W									
b6	S2RG6RP	SMPU2 zone 6 read permission		0: zone 6 allows system DMA 2 read 1: Area 6 prohibits system DMA 2 from reading		R/W									
b5	S2RG5RP	SMPU2 zone 5 read permission		0: zone 5 allows system DMA 2 read 1: Area 5 prohibits system DMA 2 from reading		R/W									
b4	S2RG4RP	SMPU2 zone 4 read permission		0: zone 4 allows system DMA 2 read 1: Area 4 prohibits system DMA 2 from reading		R/W									
b3	S2RG3RP	SMPU2 zone 3 read permission		0: zone 3 allows system DMA 2 read 1: Area 3 prohibits system DMA 2 from reading		R/W									
b2	S2RG2RP	SMPU2 zone 2 read permission		0: zone 2 allows system DMA 2 read 1: Area 2 prohibits system DMA 2 from reading		R/W									
b1	S2RG1RP	SMPU2 zone 1 read permission		0: zone 1 allows system DMA 2 read 1: Area 1 prohibits system DMA 2 from reading		R/W									
b0	S2RG0RP	SMPU2 zone 0 read permission		0: zone 0 allows system DMA 2 read 1: Area 0 prohibits system DMA 2 from reading		R/W									

12.4.17 SMPU2 Control Register MPU_S2CR

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								SMPU2E	Reserved			SMPU2ACT [1:0]		SMPU2BWP	SMPU2BRP
Bit	Flag	Bit Name		Function										Read and write	
b31~b8	Reserved	-		Reserved bit, read as 0, write 0 when writing										R/W	
b7	SMPU2E	SMPU2 enable		0: SMPU2 invalid 1: SMPU2 effective										R/W	
b6~b4	Reserved	-		Reserved bit, read as 0, write 0 when writing										R/W	
b3~b2	SMPU2ACT [1:0]	SMPU2 action selection		Set the action when the system DMA_2 is prohibited from accessing 00: ignore (read access reads 0, write access ignore) 01: Ignore + DMA bus error 10: ignore + unmaskable interrupt 11: reset										R/W	
b1	SMPU2BWP	SMPU2 write setting	background permission	0: SMPU2 background space allows system DMA_2 to write 1: SMPU2 background space prohibits system DMA_2 from writing										R/W	
b0	SMPU2BRP	SMPU2 read permission setting	background read permission	0: SMPU2 background space allows system DMA_2 to read 1: SMPU2 background space prohibits system DMA_2 from reading										R/W	

When multiple area settings overlap, the priority is: setting prohibited > setting allowed.

12.4.18 FMPU zone enable register MPU_FRGE

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								FRG7E	FRG6E	FRG5E	FRG4E	FRG3E	FRG2E	FRG1E	FRG0E
Bit	Flag	Bit Name		Function		Read and write									
b31~b8	Reserved	-		Reserved bit, read as 0, write 0 when writing		R/W									
b7	FRG7E	FMPU zone enabled	7	0: Area 7 of FMPU is invalid 1: FMPU area 7 valid		R/W									
b6	FRG6E	FMPU zone enabled	6	0: Area 6 of FMPU is invalid 1: FMPU area 6 valid		R/W									
b5	FRG5E	FMPU zone enabled	5	0: Area 5 of FMPU is invalid 1: FMPU area 5 valid		R/W									
b4	FRG4E	FMPU zone enabled	4	0: Area 4 of FMPU is invalid 1: FMPU area 4 valid		R/W									
b3	FRG3E	FMPU zone enabled	3	0: Area 3 of FMPU is invalid 1: FMPU area 3 valid		R/W									
b2	FRG2E	FMPU zone enabled	2	0: Area 2 of FMPU is invalid 1: FMPU area 2 valid		R/W									
b1	FRG1E	FMPU zone enabled	1	0: Area 1 of FMPU is invalid 1: FMPU area 1 valid		R/W									
b0	FRG0E	FMPU zone enabled	0	0: Area 0 of FMPU is invalid 1: FMPU area 0 valid		R/W									

12.4.19 FMPU Zone Write Permission Register MPU_FRGWP

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								FRG7WP	FRG6WP	FRG5WP	FRG4WP	FRG3WP	FRG2WP	FRG1WP	FRG0WP
Bit	Flag	Bit Name		Function		Read and write									
b31~b8	Reserved	-		Reserved bit, read as 0, write 0 when writing		R/W									
b7	FRG7WP	FMPU zone 7 write permission		0: zone 7 enable USBFS-DMA write 1: Area 7 disable USBFS-DMA write		R/W									
b6	FRG6WP	FMPU zone 6 write permission		0: zone 6 enable USBFS-DMA write 1: Area 6 disable USBFS-DMA write		R/W									
b5	FRG5WP	FMPU zone 5 write permission		0: zone 5 enable USBFS-DMA write 1: Area 5 disable USBFS-DMA write		R/W									
b4	FRG4WP	FMPU zone 4 write permission		0: zone 4 enable USBFS-DMA write 1: Area 4 disable USBFS-DMA write		R/W									
b3	FRG3WP	FMPU zone 3 write permission		0: zone 3 enable USBFS-DMA write 1: Area 3 disable USBFS-DMA write		R/W									
b2	FRG2WP	FMPU zone 2 write permission		0: zone 2 enable USBFS-DMA write 1: Area 2 disable USBFS-DMA write		R/W									
b1	FRG1WP	FMPU zone 1 write permission		0: zone 1 enable USBFS-DMA write 1: Area 1 disable USBFS-DMA write		R/W									
b0	FRG0WP	FMPU zone 0 write permission		0: zone 0 allows USBFS-DMA write 1: Area 0 disable USBFS-DMA write		R/W									

12.4.20 FMPU Zone Write Permission Register MPU_FRGRP

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								FRG7RP	FRG6RP	FRG5RP	FRG4RP	FRG3RP	FRG2RP	FRG1RP	FRG0RP
Bit	Flag	Bit Name		Function		Read and write									
b31~b8	Reserved	-		Reserved bit, read as 0, write 0 when writing		R/W									
b7	FRG7RP	FMPU zone 7 read permission		0: zone 7 enable USBFS-DMA read 1: zone 7 disable USBFS-DMA read		R/W									
b6	FRG6RP	FMPU zone 6 read permission		0: zone 6 enable USBFS-DMA read 1: Disable USBFS-DMA read in zone 6		R/W									
b5	FRG5RP	FMPU zone 5 read permission		0: zone 5 enable USBFS-DMA read 1: Disable USBFS-DMA read in zone 5		R/W									
b4	FRG4RP	FMPU zone 4 read permission		0: zone 4 enable USBFS-DMA read 1: Disable USBFS-DMA read in zone 4		R/W									
b3	FRG3RP	FMPU zone 3 read permission		0: zone 3 enable USBFS-DMA read 1: Disable USBFS-DMA read in zone 3		R/W									
b2	FRG2RP	FMPU zone 2 read permission		0: zone 2 enable USBFS-DMA read 1: Disable USBFS-DMA read in zone 2		R/W									
b1	FRG1RP	FMPU zone 1 read permission		0: zone 1 enable USBFS-DMA read 1: Disable USBFS-DMA read in zone 1		R/W									
b0	FRG0RP	FMPU zone 0 read permission		0: zone 0 allows USBFS-DMA read 1: Disable USBFS-DMA read in zone 0		R/W									

12.4.21 FMPU Control Register MPU_FCR

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								FMPUE	Reserved			FMPUACT [1:0]		FMPUBWP	FMPUBRP
Bit	Flag		Bit Name		Function									Read and write	
b31~b8	Reserved		-		Reserved bit, read as 0, write 0 when writing									R/W	
b7	FMPUE		FMPU enable		0: FMPU invalid 1: FMPU is valid									R/W	
b6~b4	Reserved		-		Reserved bit, read as 0, write 0 when writing									R/W	
b3~b2	FMPUACT [1:0]		FMPU action selection		Set the action when USBFS-DMA access is prohibited 00: ignore (read access reads 0, write access ignore) 01: ignore + bus error 10: ignore + unmaskable interrupt 11: reset									R/W	
b1	FMPUBWP		FMPU background write permission setting		0: FMPU background space allows USBFS-DMA write 1: FMPU background space prohibits USBFS-DMA writing									R/W	
b0	FMPUBRP		FMPU background read permission setting		0: FMPU background space allows USBFS-DMA read 1: FMPU background space prohibits USBFS-DMA reading									R/W	

13 Keyboard Scan Control Module (KEYSCAN)

13.1 Introduction

This product is equipped with a keyboard scan control module (KEYSCAN) 1 unit. The KEYSCAN module supports keyboard array (row and column) scanning, the column is driven by an independent scan output KEYOUT_m (m=0~7), and the row KEYIN_n (n=0~15) is input as EIRQ_n (n=0~15) is detected. This module realizes the key recognition function through the line scan query method.

KEYSCAN Main Features:

- EIRQ0~EIRQ15 can be independently selected as the row input of the keyboard array.
- KEYOUT can be selected by register.
- Output low level sequentially at regular intervals to scan the keyboard array.
- Scan time can be set.
- Stop scanning when the EIRQ interrupt is detected, and locate the pressed key according to the SSR.INDEX value and the IRQ interrupt flag (INT_EIFR.INT_EIFR).

13.2KEYSCAN System Block Diagram

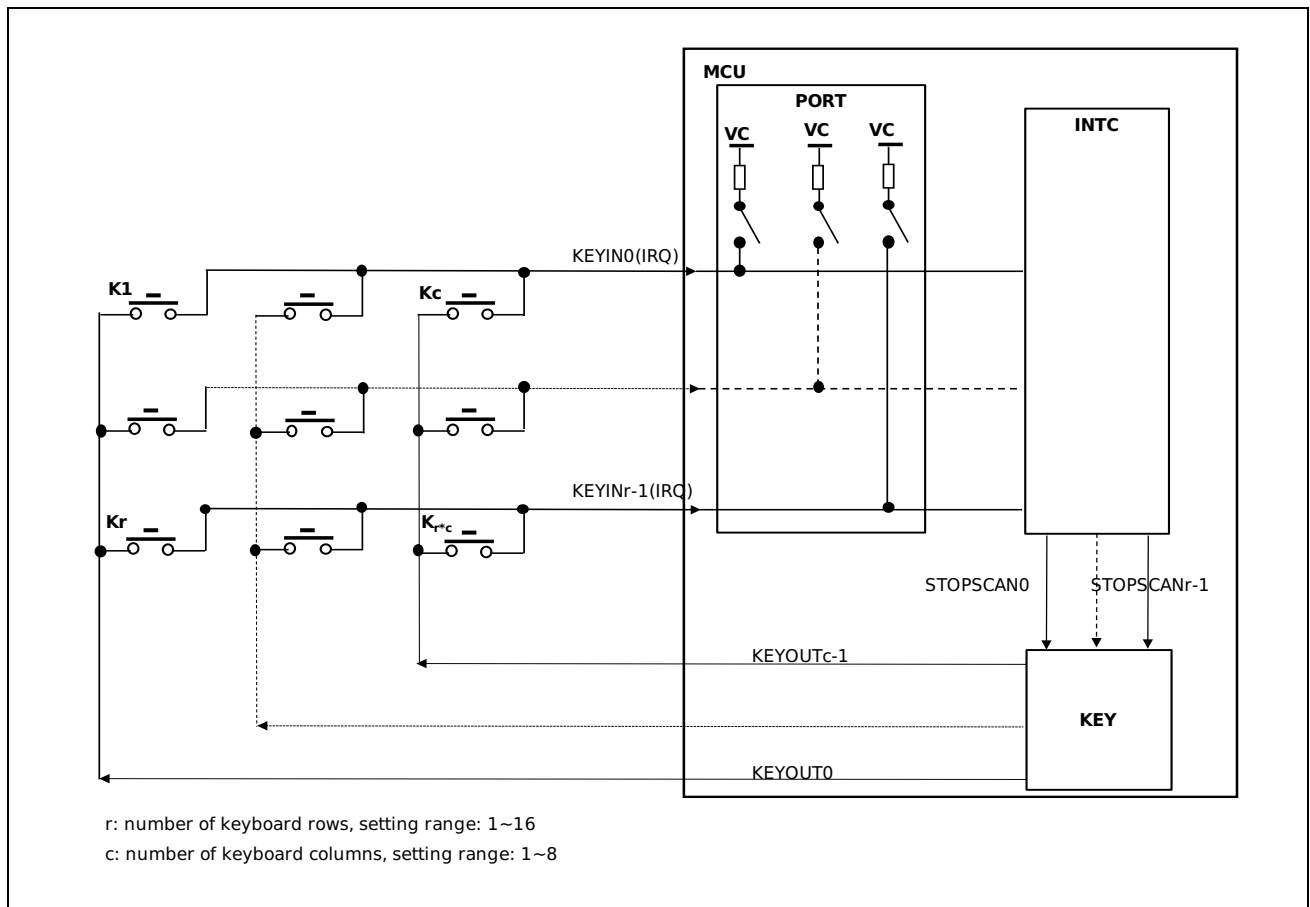


Figure 13-1 KEYSKAN System Block Diagram

13.3Pin description

Table 13-1 KEYSKAN Pin Description

Pin Name	Direction	Functional Description
KEYINn	Input	Keyboard row input signal
KEYOUTm	Output	Keyboard column output signal

n:0~15 m:0~7

13.4 Functional Description

This chapter will describe the keyboard scan function and key recognition function in detail.

13.4.1 Key Recognition Function

When a key is pressed, the row and column of the keyboard are short-circuited, and the row generates a falling edge, thereby generating an EIRQ interrupt flag, which is located by comparing the value of the interrupt flag bit (INT_EIFR.INT_EIFR) and SSR.INDEX [2:0]. The key is currently pressed.

Through the register SCR.KEYINSEL [15:0], KEYIN can be independently selected from EIRQ0~EIRQ15, and through the register SCR.KEYOUTSEL [2:0], the KEYOUT pin can be selected, so that the number of rows and columns of the keyboard can be flexibly selected. It can support a keyboard array of 16 rows*8 columns.

13.4.2 Keyboard Scan Function

The keyboard scanning function is: output low level to the columns of the keyboard array in a continuous period, so that when a key is pressed, a corresponding EIRQ interrupt flag will be generated.

When SER.SEN is set to 1, KEYOUT0 outputs low level, KEYOUT1~KEYOUTn (n is set by SCR.KEYOUTSEL [2:0]) is HIZ, after the time set by SCR.T_LLEVEL [4:0], KEYOUT0~KEYOUTn pins are all HIZ, after the time set by SCR.T_HIZ [2:0], KEYOUT1 outputs low level, other KEYOUT pins are HIZ, and so on. When a key is pressed and an EIRQ interrupt flag is generated, the keyboard scanning function stops, and the scanning automatically restarts after the corresponding interrupt flag is cleared.

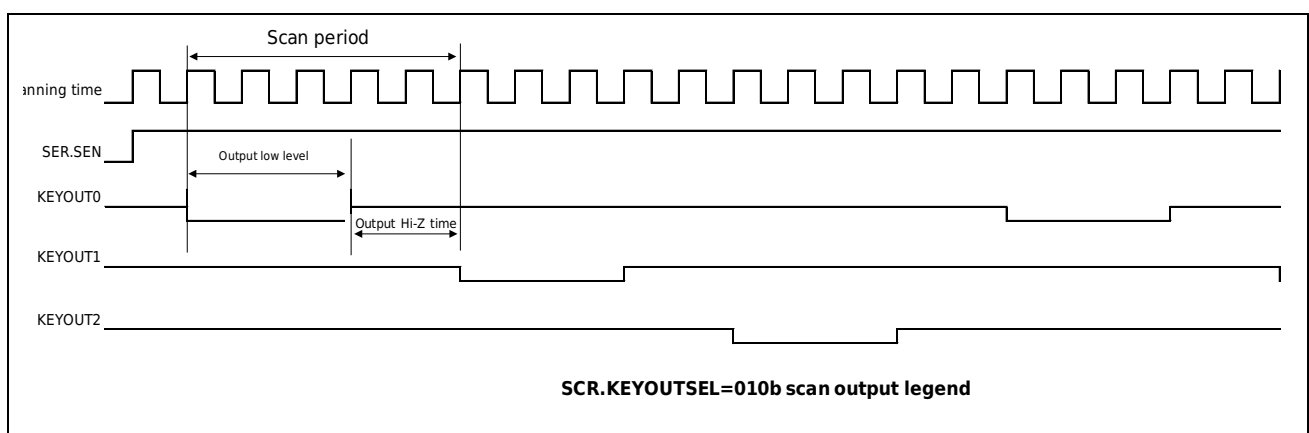


Figure 13-2 Schematic Diagram of Keyboard Scanning Function

13.4.3 Precautions for use

This module drives the keyboard column, and the keyboard row detection is realized by the external EIRQ function of the interrupt control module (INTC). EIRQ needs to select the falling edge detection, and open the digital filter function, and set the appropriate filter time.

If you use this function in STOP mode, you need to set the scan-related parameters and select the internal low-speed oscillator LRC or the external low-speed oscillator XTAL32 clock as the scan clock.

If you use the on-chip pull-up resistor, please refer to the PORT characteristics to select the appropriate scan time and filter time.

13.5 Register Description

Table 13-2 KEYSKAN Register List

KEYSCAN_BASE_ADDR: 0x40050C00

Register name	Symbol	Offset Address	Bit Width	Reset Value
KEYSCAN scan control register	KEYSCAN_SCR	0x00	32	0x0000 0000
KEYSCAN scan enable register	KEYSCAN_SER	0x04	32	0x0000 0000
KEYSCAN scan status register	KEYSCAN_SSR	0x08	32	0x0000 0000

13.5.1 KEYSCAN Scan Control Register (KEYSCAN_SCR)

KEYSCAN Scan Control Register

Offset address: 0x00

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
T_HIZ [2:0]			T_LLEVEL [4:0]				-	-	CKSEL [1:0]	-	KEYOUTSEL [2:0]				
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
KEYINSEL [15:0]															

Bit	Flag	Bit Name	Function	Read and write
b31~b29	T-HIZ [2:0]	Output HIZ time	KEYOUT output low level HIZ time (number of scan clocks) Scan period = output low level time + output HIZ time Setting value: number of HIZ periods 000b: 4 001b: 8 010b: 16 011b: 32 100b: 64 101b: 256 110b: 512 111b: 1024 Note: SCR.T-HIZ [2:0] can only be set valid when SER.SEN=0	R/W
b28~b24	T_LLEVEL [4:0]	Output low level time	KEYOUT output low level time (number of scan clocks) Scan period = output low level time + output HIZ time Output low level time = 2 T_LLEVEL power scan clocks Note: SCR.T-LLEVEL [4:0] can only be set to be effective when SER.SEN=0, and 00000b and 00001b are prohibited from being set. The maximum value that can be set is 11000b	R/W
b23~b22	Reserved	-	Read as "0", write as "0"	R/W
b21~b20	CKSEL [1:0]	Scan clock source select bit	Scan clock source select bit 00b: System clock HCLK 01b: Internal low-speed oscillator LRC 10b: External low-speed oscillator XTAL32 11b: Setting prohibited Note: SCR.CKSEL [1:0] can only be set valid when SER.SEN=0	R/W
b19	Reserved	-	Read as "0", write as "0"	R/W
b18~b16	KEYOUTSEL [2:0]	Output selection	KEYOUT output selection bit Set value: Output selection 000b: Prohibition 001b: KEYOUT0~KEYOUT1 010b: KEYOUT0~KEYOUT2 011b: KEYOUT0~KEYOUT3 100b: KEYOUT0~KEYOUT4 101b: KEYOUT0~KEYOUT5 110b: KEYOUT0~KEYOUT6 111b: KEYOUT0~KEYOUT7 Note: SCR.KEYOUTSEL [2:0] can only be set valid when SER.SEN=0	R/W
b15~b0	KEYINSEL [15:0]	Row input select bit	Row input selection bit, the selected row is used as the row of the keyboard array, and as EIRQn (n: 0~15) is detected KEYINSEL [n]=0: KEYINSEL [n] is not used as the line of the keyboard array KEYINSEL [n]=1: KEYINSEL [n] is used as the row of the keyboard array nRange : 0~15 Note: SCR. KEYINSEL [15:0] can only be set valid when SER.SEN=0	R/W

13.5.2 KEYSCAN Scan Enable Register (KEYCAN_SER)

KEYSCAN Scan Enable Register

Offset address: 0x04

Reset value: 0x00000000

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

Bit	Flag	Bit Name	Function	Read and write
b31~b1	Reserved	-	Read as "0", write as "0"	R/W
b0	SEN	scan enable bit	scan enable bit 0: scan disabled 1: scan enable	R/W

13.5.3 KEYSCAN Scan Status Register (KEYSCAN_SSR)

KEYSCAN Scan Status Register

Offset address: 0x08

Reset value: 0x00000000

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

Bit	Flag	Bit Name	Function	Read and write
b31~b3	Reserved	-	Read as "0", write as "0"	R/W
b2~b0	INDEX [2:0]	Current working SCAN pin index bit	Current working SCAN pin index bit 000: The current working SCAN pin is KEYOUT0 001: The currently working SCAN pin is KEYOUT1 010: The currently working SCAN pin is KEYOUT2 011: The currently working SCAN pin is KEYOUT3 100: The currently working SCAN pin is KEYOUT4 101: The currently working SCAN pin is KEYOUT5 110: The currently working SCAN pin is KEYOUT6 111: The currently working SCAN pin is KEYOUT7 Note: The SSR.INDEX [2:0] bits are read-only registers, and the data read only when SER.SEN=1 is meaningful	R

14 Internal Clock Calibrator (CTC)

14.1 Introduction

The clock trimming controller(CTC) is used to calibrate the frequency of the internal high-speed oscillator (HRC). HRC frequency has been calibrated before the chip leaves the factory , due to different working environments, the HRC frequency often still deviates and affects the system accuracy. Using CTC , the HRC can be further calibrated automatically based on an external high-precision reference clock , thus providing the system with a precise HRC clock.

Main Features of CTC :

- Hardware automatic calibration without software involvement
- Three external high-precision reference clock sources
- Two calibration modes: continuous calibration and single calibration
- Adjustable from -32 to +31 based on factory calibration values
- Error interrupt to prompt calibration failure

14.2 Structure Diagram

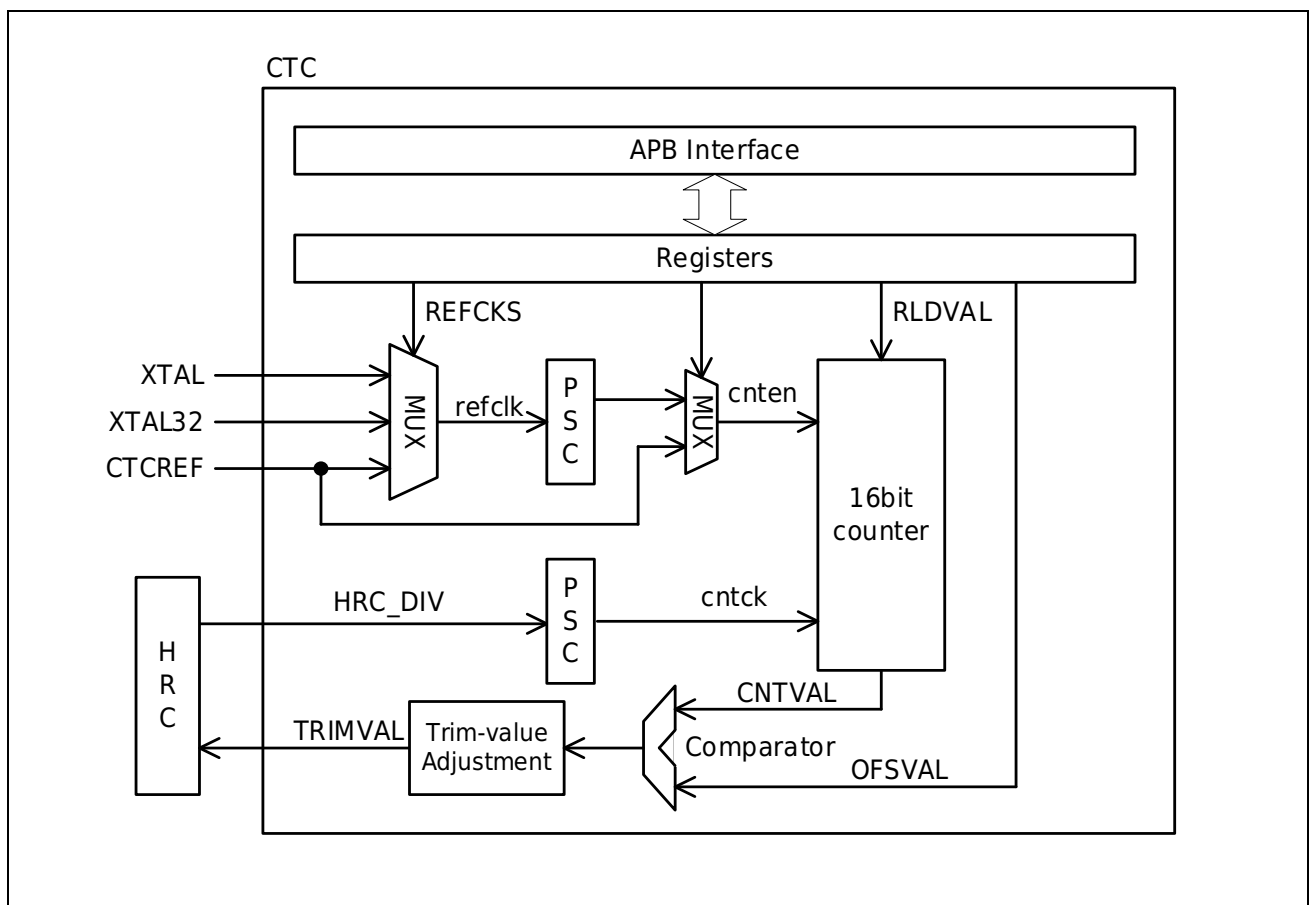


Figure 14-1 Basic Block Diagram of CTC

14.3 Functional Description

14.3.1 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 HRCPSK [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 factor; PSC_{ref} represents the reference clock frequency division factor;

Table 14-1 and Table 14-2 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 14-1 Measurement Error when HRC Target Frequency of 20MHz

Fref		1KHz							
PSCref		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%
Fref		32.768KHz							

PSCref		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%
Fref		100KHz							
PSCref		2	4	8	16	64	256	1024	4096
PSC hrc	1	Not recommended	1.607%	1.033%	0.678%	0.308%	0.146%	Irreducible	Irreducible
	2	Not recommended	Not recommended	1.283%	0.803%	0.339%	0.154%	Irreducible	Irreducible
	4	Not recommended	Not recommended	1.783%	1.053%	0.402%	0.170%	0.077%	Irreducible
	8	Not recommended	Not recommended	Not recommended	1.553%	0.527%	0.201%	0.085%	Irreducible
	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%
Fref		1MHz							
PSCref		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
Fref		8MHz							
PSCref		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
Fref		20MHz							
PSCref		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

	51 2	Not recommen ded	Not recommen ded	Not recommen ded	Not recommen ded	Not recommen ded	Not recommen ded	Not recommen ded	Not recommen ded
	20 48	Not recommen ded	Not recommen ded	Not recommen ded	Not recommen ded	Not recommen ded	Not recommen ded	Not recommen ded	Not recommen ded

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 14-2 Measurement Error when HRC Target Frequency of 16MHz

Fref		1KHz							
PSCref		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
	12 8	1.775%	0.924%	0.488%	0.262%	0.081%	0.028%	Irreducible	Irreducible
	51 2	Not recommen ded	Not recommen ded	1.688%	0.862%	0.231%	0.065%	0.020%	Irreducible
	20 48	Not recommen ded	Not recommen ded	Not recommen ded	Not recommen ded	0.831%	0.215%	0.058%	0.016%
Fref		32.768KHz							
PSCref		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 recommen ded	1.528%	0.911%	0.559%	0.228%	0.101%	Irreducible	Irreducible
	8	Not recommen ded	Not recommen ded	1.320%	0.764%	0.279%	0.114%	0.051%	Irreducible
	32	Not recommen ded	Not recommen ded	Not recommen ded	1.993%	0.587%	0.191%	0.070%	0.029%
	12 8	Not recommen ded	Not recommen ded	Not recommen ded	Not recommen ded	1.815%	0.498%	0.147%	0.048%
	51 2	Not recommen ded	Not recommen ded	Not recommen ded	Not recommen ded	Not recommen ded	1.727%	0.454%	0.125%
	20 48	Not recommen ded	Not recommen ded	Not recommen ded	Not recommen ded	Not recommen ded	Not recommen ded	1.683%	0.432%
Fref		100KHz							
PSCref		2	4	8	16	64	256	1024	4096
PSC hrc	1	Not recommen ded	1.863%	1.188%	0.775%	0.348%	0.164%	Irreducible	Irreducible
	2	Not recommen ded	Not recommen ded	1.500%	0.931%	0.388%	0.174%	Irreducible	Irreducible
	4	Not recommen ded	Not recommen ded	Not recommen ded	1.244%	0.466%	0.194%	0.087%	Irreducible
	8	Not recommen ded	Not recommen ded	Not recommen ded	1.869%	0.622%	0.233%	0.097%	Irreducible
	32	Not recommen ded	Not recommen ded	Not recommen ded	Not recommen ded	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%
Fref		1MHz							
PSCref		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
Fref		8MHz							
PSCref		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%
	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
Fref		20MHz							

PSCref		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 period). 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, in units of S; PSC_{hrc} represents the HRC division number.

14.3.2 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.

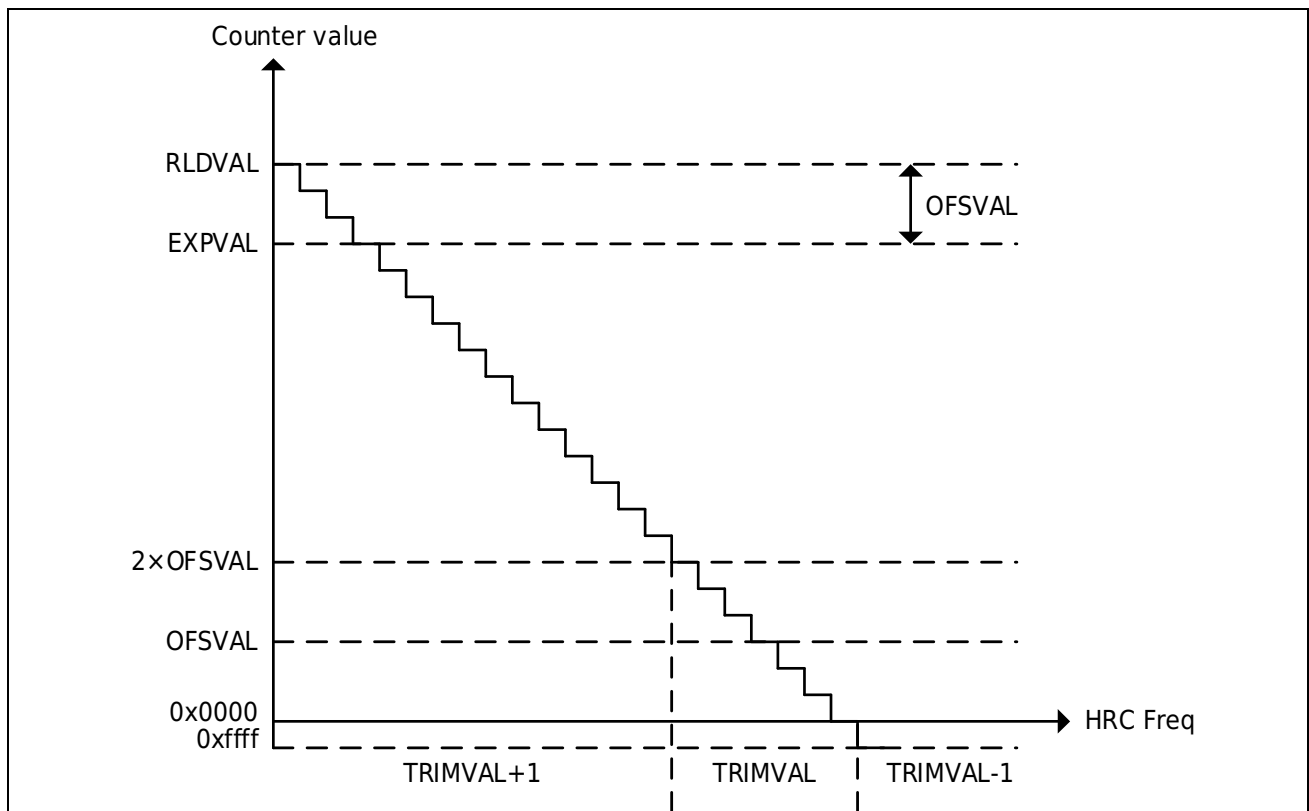


Figure 14-2 Schematic Diagram of CTC Calibration

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 $2 \times \text{OFSVAL}$ 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 $2 \times \text{OFSVAL}$, 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 > ($\text{OFSVAL} \times 2$), counting stops and the value of TRMVAL automatically increases by 1.
- Counting stops when $0 \leq \text{Counter} \leq (\text{OFSVAL} \times 2)$, 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.

14.3.3 Error 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 TRMUFD 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. 12. Interrupt Controller" section of this manual.

14.3.4 Programming Guide

Take the instance 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 looking up Table 14-2, 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.

- Set the register.
CTC_CR2 = 0x1EAB0027 , where
$$\text{OFSVAL} = (16 \div 0.032768) \times 16 \times 0.5\% \approx 39 \text{ (0x27)};$$
$$\text{RLDVAL} = (16 \div 0.032768) \times 16 + 39 \approx 7851 \text{ (0x1EAB)};$$
- Set the register.
CTC_CR1 = 0x000000e3 , where
TRMVAL = 0x00 (Calibrate from median)
HRCPSC=000b (select HRC without frequency division)
CTCEN = 1 (start CTC)
ERRIE = 1 (Allow incorrect interrupt)
REFCKS = 10b (select XTAL32 as reference clock for continuous calibration)
REFPSC = 011b (Select 8 frequency division of reference clock)
- After a while, confirm the status of the CTC_STR.TRIMOK bit.
If TRIMOK=1, write 0 to CTCEN to stop CTC;
If TRIMOK=0 , return to this step;
- 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.

14.4 Register Description

Table 14-3 List of CTC Registers

Reference address: 0x40049C00

Register Name	Symbol	Offset Address	Bit Width	Reset Value
Clock calibration control register1	CTC_CR1	0x00	32	0x00000000
Clock calibration control register2	CTC_CR2	0x04	32	0x00000000
Clock calibration status register	CTC_STR	0x08	32	0x00000000
Clock Calibration Counter	CTC_CNT	0x0c	16	0x0000

14.4.1 Clock Calibration Control Register1 (CTC _ CR1)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	TRMVAL [5:0]					
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	REFEDG [1:0]		-	HRCPSC [2:0]			CTC EN	ERR IE	REFCKS [1:0]		-	REFPSC [2:0]		

Bit	Flag	Bit Name	Function	Read and write
b31~b22	Reserved	-	Read as "0", write as "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.	R/W
b15~b14	Reserved	-	Read as "0", write as "0"	R
b13~b12	REFEDG [1:0]	Reference signal edge selection	Used to select the valid edge of CTCREF when REFCKS = 01b. 00b: Select the rising edge to the rising edge of CTCREF for calibration 01b: Select the rising edge to the falling edge of CTCREF for calibration 10b : Select the falling edge to the rising edge of CTCREF for calibration 11b : Select the falling edge to 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.	R/W
b11	Reserved	-	Read as "0", write as "0"	R
b10~b8	HRCPSC [2:0]	HRC frequency division selection	000b: No frequency division 001b: 2 frequency division 010b: 4 frequency division 011b: 8 frequency division 100b: 32 frequency division 101b: 128 frequency division 110b: 512 division 111b: 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.	R/W
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	R/W

			occurs in calibration; (3) Automatically reset to zero at the end of a single calibration;	
b6	ERRIE	Error interrupt disable	Used to select whether error interrupt occurs when a calibration overflow or underflow occurs. 0: Prevent error interrupt 1: Allow error interrupt	R/W
b5~b4	REFCKS [1:0]	Reference clock selection	Used to select a reference timer for frequency calibration. 00b: Select CTCREF as reference clock (continuous calibration) 01b: Select CTCREF as the reference signal (single calibration) 10b : Select XTAL32 as reference clock (continuous calibration) 11b : 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.	R/W
b3	Reserved	-	Read as "0", write as "0"	R
b2~b0	REFPSC [2:0]	Reference clock frequency division selection	When REFCKS=00b/10b/11b, select the reference clock division number. 000b: 2 frequency division 001b: 4 frequency division 010b: 8 frequency division 011b: 16 frequency division 100b: 64 frequency division 101b: 256 frequency division 110b: 1024 division 111b: 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.	R/W

14.4.2 Clock Calibration Control Register2 (CTC _ CR2)

Reset value: 0x00000000

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
-	-	-	-	-	-	-	-	OFSVAL [7:0]							

Bit	Flag	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, $RLDVAL = (Fhrc \div PSChrc) \div (Fref \div PSCref) + OFSVAL$ For single calibration, $RLDVAL = ((Fhrc \div PSChrc) \times Ttrim) + OFSVAL$ Fhrc indicates the HRC target frequency; Fref represents the frequency of the reference clock; Ttrim represents a single calibration period; OFSVAL represents the calibration count deviation value; PSChrc represents the HRC frequency division factor; PSCref represents the reference clock frequency division factor; 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.	R/W
b15~b8	Reserved	-	Read as "0", write as "0"	R
b7~b0	OFSVAL [7:0]	Calibration target deviation value	OFSVAL sets the permissible deviation value for HRC calibration. During continuous calibration, $OFSVAL = (Fhrc \div PSChrc) \div (Fref \div PSCref) \times TA$ During single calibration, $OFSVAL = ((Fhrc \div PSChrc) \times Ttrim) \times TA$ Fhrc indicates the HRC target frequency; Fref represents the frequency of the reference clock; Ttrim represents a single calibration period; TA represents the calibration accuracy, for example 0.5%; PSChrc represents the HRC frequency division factor; PSCref represents the reference clock frequency division factor; 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.	R/W

14.4.3 Clock Calibration Status Register (CTC_STR)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	CTC BSY	TRM UDF	TRM OVF	TRIM OK

Bit	Flag	Bit Name	Function	Read and write
b31~b4	Reserved	-	Read as "0", write as "0"	R
b3	CTCBSY	CTC status flag	0: CTC is stopped 1: CTC is working	R
b2	TRMUDF	Calibration underflow flag	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 flag	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 flag	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

14.4.4 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	Flag	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

15 DMA Controller (DMA)

15.1 Introduction

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.

- DMA bus is independent of CPU bus and transmitted according to AMBA AHB-Lite bus protocol.
- With 2 DMA control units, a total of 16 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
- Up to 65535 transmissions can be configured
- 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
- HPROT value in the AHB bus during DMA access can be set by register

15.2 Module Diagram

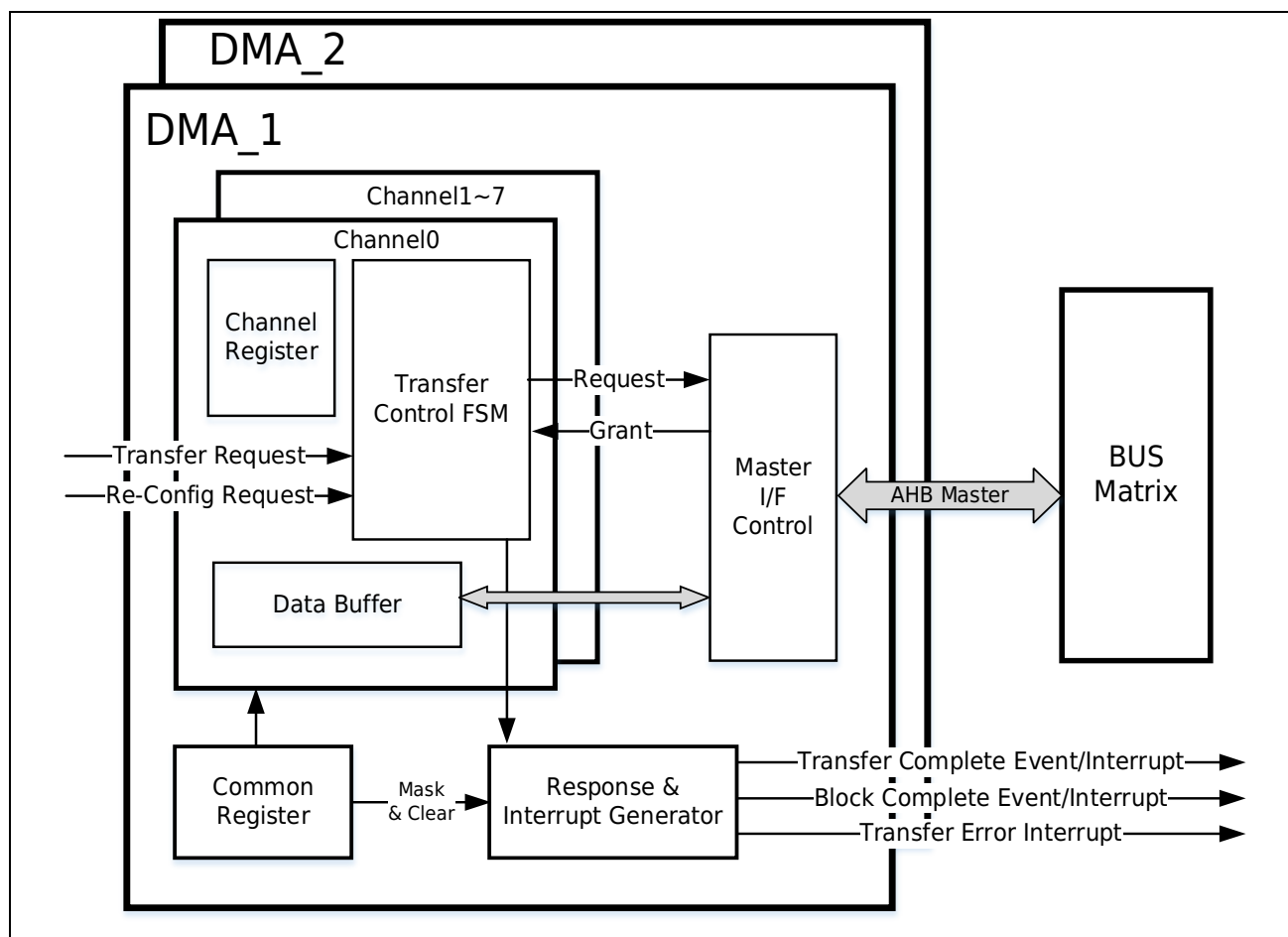


Figure 15-1 DMA Structure Diagram

15.3 Functional Description

15.3.1 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 register DMA_CHSTAT.DMAACT is 0 before writing 0 to ensure that DMA has completed all transfers.

15.3.2 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.

15.3.3 Start DMA

DMA is started by requests generated by peripheral circuits. These requests 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).

15.3.4 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.

15.3.5 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.

15.3.6 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.

15.3.7 Interrupt and Event Signal Output

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.

15.3.8 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_LL Px

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:

If LLPEN=1, LLPRUN=0, BTC and TC interrupts will be generated according to the interrupt permission configuration, and the channel permission CHEN [x] will not be automatically cleared to 0. The next descriptor specified by LLP is loaded into the channel configuration register from the memory, waits for the next transmission request input, and starts the first transmission of the new descriptor.

If LLPEN=1, LLPRUN=1, BTC and TC interrupts will not be generated, channel permission CHEN [x] will not be automatically cleared to 0. The first transfer of a new descriptor begins directly after the next descriptor specified by the LLP is loaded from memory into the channel configuration register.

If LLPEN=0, the chain transmission ends, and BTC and TC interrupts are generated according to the interrupt permission configuration, and the channel permission CHEN [x] is automatically cleared to 0.

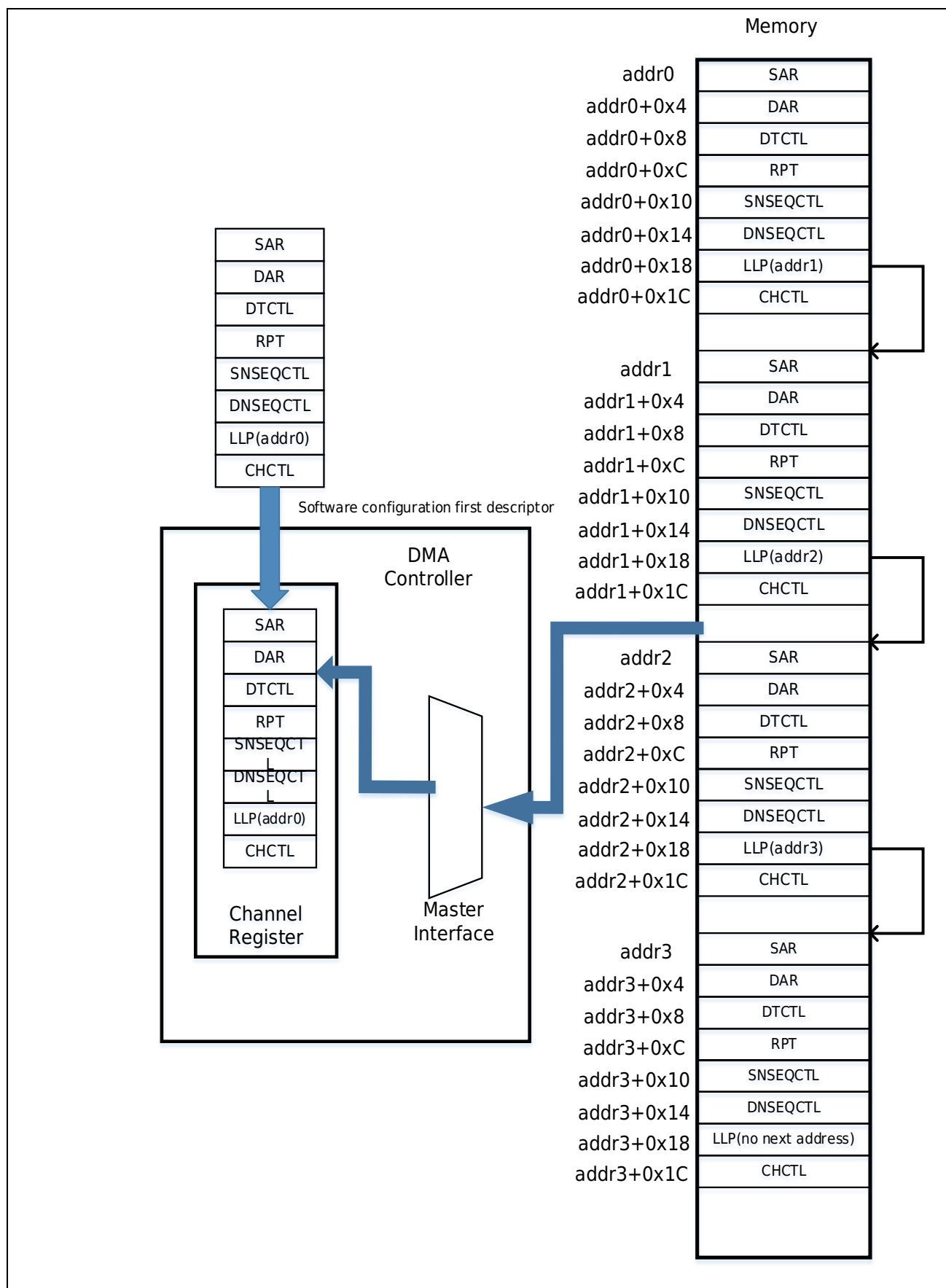


Figure 15-2 Chain transmission diagram

15.3.9 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 instance: 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.

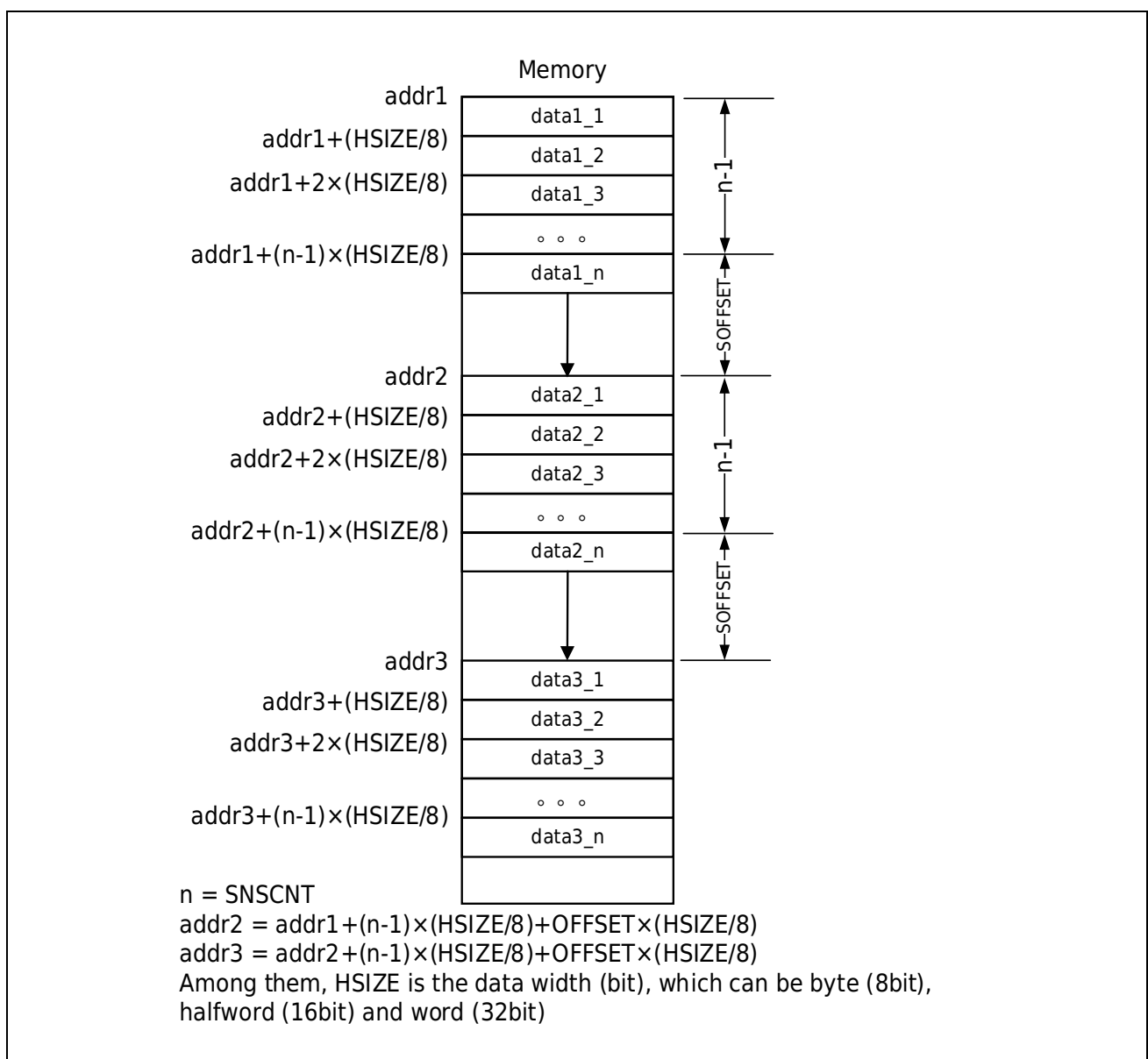


Figure 15-3 Schematic Diagram of Discontinuous Address Transmission (Taking Source Address as an Instance)

15.3.10 Channel Reset

The channel reset function refers to modifying the internal status register of the channel through the event request of the peripheral circuit, and reconfiguring the next data transmission mode. Set register DMA_RCFGCTL.RCFGGEN to 1 to allow channel reset. Select the reset request source through the 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, reload reset is selected, the internal state of the channel is updated as described in the table below.

Table 15-1 Channel Reset Description

Channel Internal State	Reset Method	
	Discontinuous	Heavy Duty
Remaining Transfers Counter	Update to the value after the next address discontinuous jump occurs in the normal state	Update to the value after the next reload occurs in the normal state
Source/destination address for next transfer	Update to the first address of the next discontinuous transfer area	Update to the initial setting value of register DMA_SARx/DARx

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 and DMA_SNSEQCTLx, DMA_DNSEQCTLx are invalid.

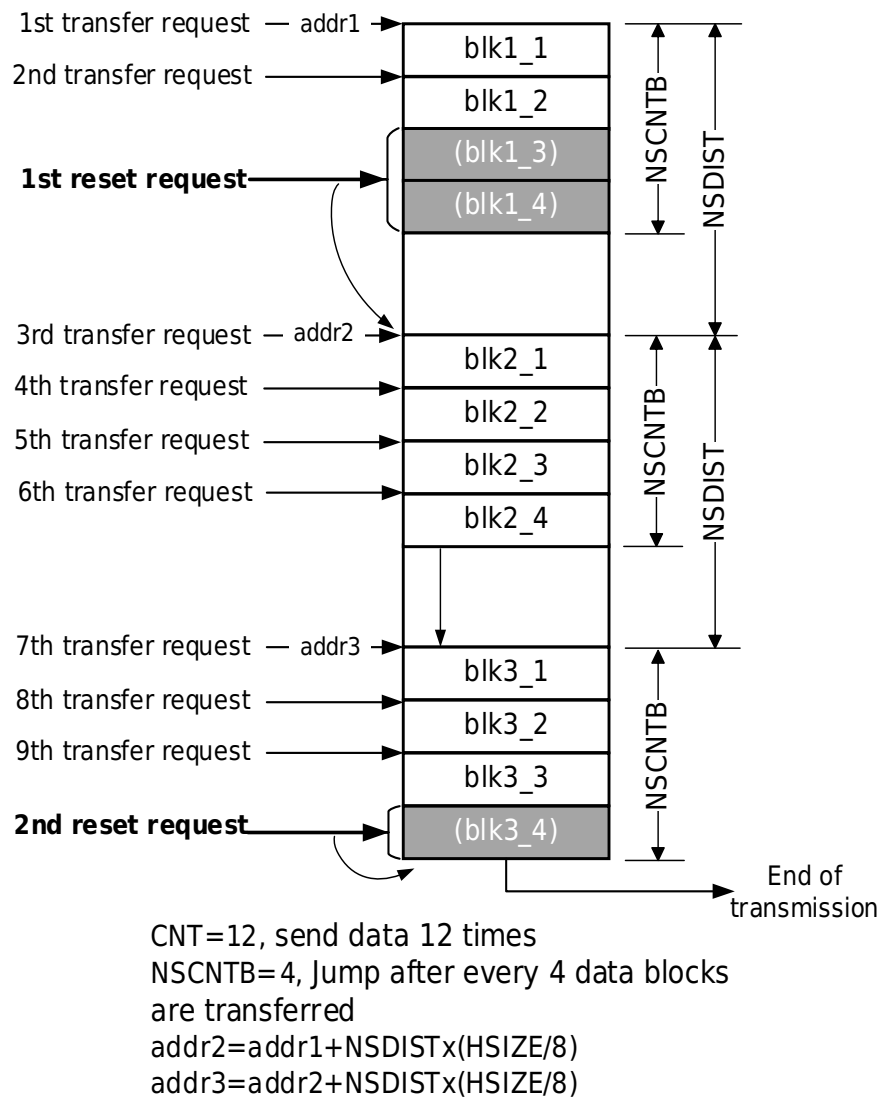


Figure 15-4 Schematic Diagram of Discontinuous Reset

Figure 15-4 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.

15.3.11 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.CHENCLR x 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 CHENCLR_x 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 CHEN_x 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.

15.4 Application Instances

15.4.1 Memory-to-Memory Transmission

Target: Transfer 12 data from the RAM address 0x20000000 to 0x20001000 with a 32-bit data width.

1. Register Setting
 - Dma _ EN.EN Write 1 Enables DMAcontroller
 - Select a channel, such as channel 0, and configure the channel register to:
 - Write DMA _ SAR0 configuration source address is SRAM area 0x20000000
 - Write DMA _ DAR0 configure target address as SRAM area 0x20001000
 - 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
 - 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
 - Configure channel control register DMA_CH0CTL to achieve:
 - * Invalid link between source and destination
 - * Source address overload is valid, and destination address update is self-increasing
 - * Data width is word (32 bits)
 - * Interrupt enable is valid
 - Configure trigger source controllerDMA _ TRGSEL0 and select software trigger as start request for DMA channel 0
 - Channel enable bit DMA_CHEN.CHEN0 write 1, enable channel 0
 - 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 INTSFTRG.STRG is 1, and the number of transmissions is DMA_DTCTL0.CNT CH0CTL0 when a data block is transmitted.CNT minus 1, and produces a block transport complete interrupt, the software can continue to write INTSFTRG.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 0x20000000 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_DTCTL0.CNT is reduced by 1, and a block transfer completion interrupt is generated, and the software can continue to write INTSFTRG.STRG in the interrupt subroutine to start the third transfer. After the third transfer is completed, the number of transfers DMA_DTCTL0.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.

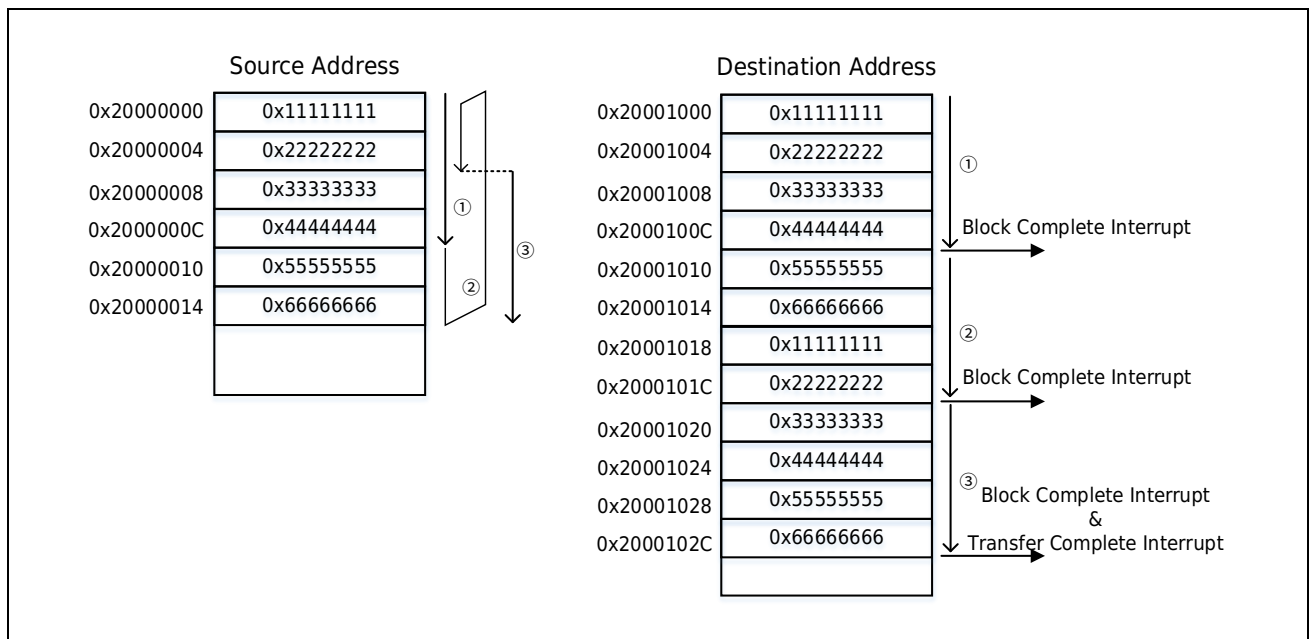


Figure 15-5 Application Instance 1: Memory-to-Memory Transfer

15.4.2 Memory-to-Peripheral Transmission

Target: Transfer 10 pieces of data with a width of half-word from the RAM address 0x20000000 to the transmission buffer register of the communication module, each time the communication module completes the transmission request. When the last data is sent, DMA generates a transfer complete interrupt.

1. Register Setting
 - Dma _ EN.EN Write 1 Enables DMAcontroller
 - Configure the DMA_INTMSK register, mask the block transfer completion interrupt, and enable the transfer completion interrupt
 - Select a channel to configure the channel register, for example, select channel 0
 - Write DMA _ SAR0 configuration source address is SRAM area 0x20000000
 - Write DMA_DAR0 to configure the target address to be the register address of the peripheral circuit 0x40000000
 - Write DMA_DTCTL0Configuration data block size is 1, number of transfers is 10, each transfer request is delivered once, one data at a time.
 - Configure channel control register DMA_CH0CTL to achieve:
 - * Invalid link between source and destination
 - * Source address update mode is self-increment, target address is fixed
 - * Source and destination address data width is half word (16 bits)
 - * Interrupt enable is valid
 - Configure the trigger source controller DMA_TRGSEL0, and select the event that the send register of the communication module is empty as the start request of DMA channel 0
 - 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 masked 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.

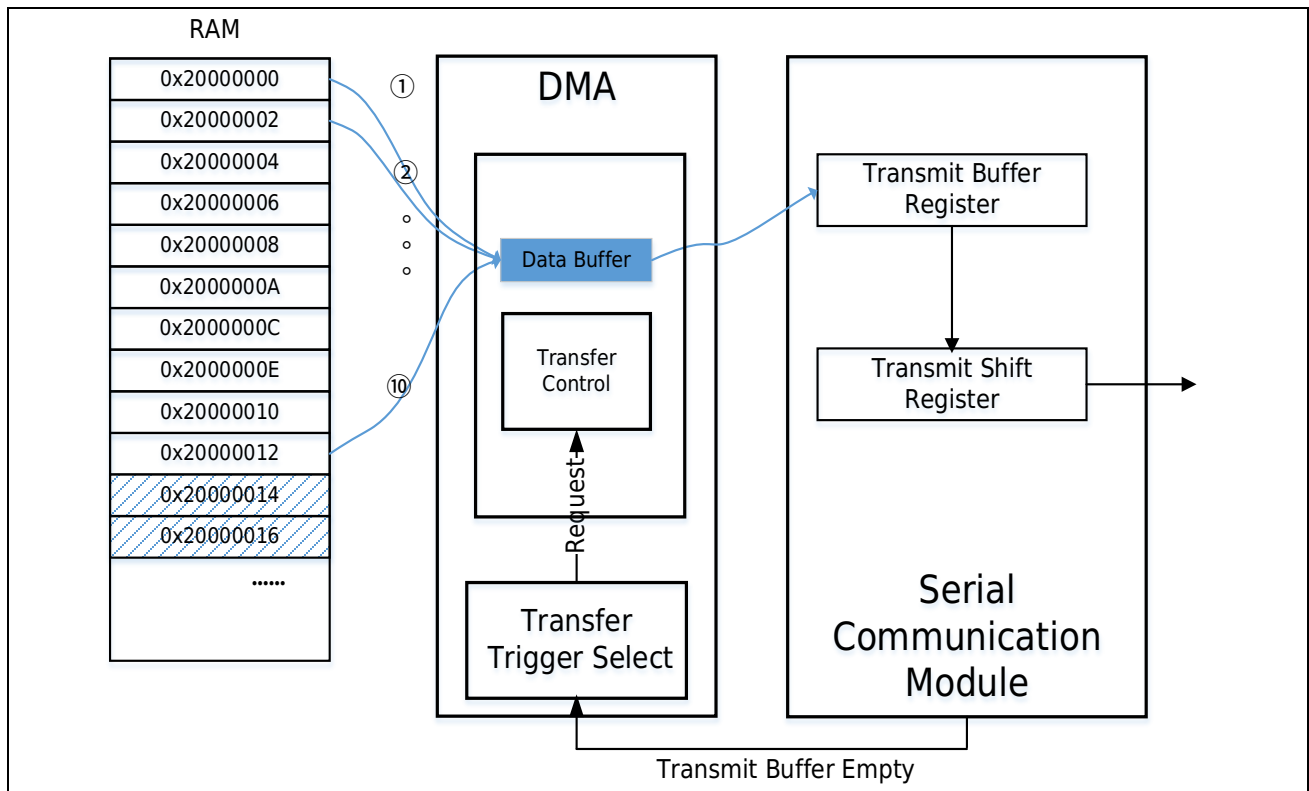


Figure 15-6 Application Instance 2: Transfer from Memory to Peripheral Circuits

15.4.3 Memory-to-Memory Linkage

1. Register Setting

- Dma _ EN.EN Write 1 Enables DMAcontroller
- Select a channel, configure the channel register, for example, select channel 0, configure the descriptor for the first transmission (descriptor 0)
 - Write DMA _ SAR0 configuration source address is SRAM area 0x20000000
 - Write DMA _ DAR0 configure target address as SRAM area 0x20001000
 - Write DMA_DTCTL0 configuration data block size is 10
 - Write the address 0x20002000 of the second descriptor (descriptor1) in the chain pointer register DMA_LLPO
 - Configure the channel control register DMA_CH0CTL, and configure the transmission parameters of the first data block to achieve:

- * Linkage transmission efficiency
- * Chain transmission mode is to start the next transmission directly
- * Source and destination address updates are self-incremented
- * Data width is word (32 bits)
- * Interrupt enable invalid
- Configure the descriptor (descriptor 1) for the second transmission in the 0x20002000 address of the RAM space, including:
 - 0x20002000 writes 32-bit data 0x20000100, which is the source address of the second transmission
 - 0x20002004 writes 32-bit data 0x20001100, which is the target address of the second transmission
 - 0x20002008 Configuration data block size is 20
 - 0x20002018 writes 32-bit data 0x20002020, which is the address of the descriptor (descriptor 2) of the third transmission
 - 0x2000201C writes control data for the second transmission, which implements:
 - * Linkage transmission efficiency
 - * Chain transmission mode is to start the next transmission directly
 - * Source and destination address updates are self-incremented
 - * Data width is half word (16 bits)
 - * Interrupt enable invalid
- Configure the descriptor (descriptor 2) for the third transmission in the 0x20002020 address of the RAM space, including:
 - 0x20002020 writes 32-bit data 0x20000200, which is the source address of the third transmission
 - 0x20002024 writes 32-bit data 0x20001200, which is the target address for the third transmission
 - 0x20002028 Configuration data block size is 40
 - 0x20002038 writes 32-bit data 0x0, representing the last transmission of the link
 - 0x2000203C write to the third transmission of the control data implementation:
 - * Chain transmission invalid
 - * Source and destination address updates are self-incremented
 - * Data width in bytes (8 bits)
 - * Interrupt enable is valid
- Configure the transfer start trigger source selection register DMA_TRGSEL0, and select the software trigger as the start request of DMA channel 0
- Channel enable bit DMA_CHEN.CHEN0 write 1, enable channel 0
- Write software triggers registerINTSFTRG.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.

15.5 Register Description

15.5.1 Register List

Table 15-2 Register Summary

Unit 1 BASE_ADDR: 0x40053000

Unit 2 BASE_ADDR: 0x40053400

Register Name	Symbol	Offset Address	Bit Width	Reset Value
DMA enable register	DMA_EN	0x00	32	0x00000000
Interrupt state register0	DMA_INTSTAT0	0x04	32	0x00000000
Interrupt state register1	DMA_INTSTAT1	0x08	32	0x00000000
Interrupt mask register 0	DMA_INTMASK0	0x0C	32	0x00000000
Interrupt mask register 1	DMA_INTMASK1	0x10	32	0x00000000
Interrupt reset register 0	DMA_INTCLR0	0x14	32	0x00000000
Interrupt reset register 1	DMA_INTCLR1	0x18	32	0x00000000
Channel enable register	DMA_CHEN	0x1c	32	0x00000000
Channel enable reset register	DMA_CHENCLR	0x34	32	0x00000000
Transmission request status register	DMA_REQSTAT	0x20	32	0x00000000
Channel status observation register	DMA_CHSTAT	0x24	32	0x00000000
Channel reset control register	DMA_RCFGCTL	0x2c	32	0x00000000
Software startup register	DMA_SWREQ	0x30	32	0x00000000
Transmission source address register	DMA_SARx *1	0x40+0x40*x	32	0x00000000
Transmission destination address register	DMA_DARx	0x44+0x40*x	32	0x00000000
Data control register	DMA_DTCTLx	0x48+0x40*x	32	0x00000001
Repeat Region Size Register	DMA_RPTx	0x4C+0x40*x	32	0x00000000
Repeat Region Size Register B	DMA_RPTBx	-	-	-
Source Device Discontinuous Address Transfer Control Register	DMA_SNSEQCTLx	0x50+0x40*x	32	0x00000000
Source Device Discontinuous Address Transfer Control Register B	DMA_SNSEQCTLBx	-	-	-
Target Device Discontinuous Address Transfer Control Register	DMA_DNSEQCTLx	0x54+0x40*x	32	0x00000000
Target Device Discontinuous Address Transfer Control Register B	DMA_DNSEQCTBx	-	-	-
Chain pointer register	DMA_LL Px	0x58+0x40*x	32	0x00000000
Channel control register	DMA_CHCTLx	0x5C+0x40*x	32	0x00001000
Transfer Source Address Monitor Register	DMA_MONSARx	0x60+0x40*x	32	0x00000000
Transfer Destination Address Monitor Register	DMA_MONDARx	0x64+0x40*x	32	0x00000000
Data Control Monitor Register	DMA_MONDTCTLx	0x68+0x40*x	32	0x00000001
Repeat Region Counter Monitor Register	DMA_MONRPTx	0x6C+0x40*x	32	0x00000000
Source Device Discontinuous Transfer Counter Monitor Register	DMA_MONSNSEQCTLx	0x70+0x40*x	32	0x00000000

Register Name	Symbol	Offset Address	Bit Width	Reset Value
Target Device Discontinuous Transfer Counter Monitor Register	DMA_MONDNSEQCTLx	0x74+0x40*x	32	0x00000000

15.5.2 DMA Enable Register (DMA_EN)

Reset value: 0x00000000

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

Bit	Flag	Bit Name	Function	Read and write
b31-b1	Reserved	-	Read as "0", write as "0"	R/W
b0	EN	DMA enable bit	0: DMA invalid 1: DMA enable	R/W

15.5.3 Interrupt State Register0 (DMA_INTSTAT0)

Reset value: 0x00000000

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

Bit	Flag	Bit Name	Function	Read and write
b31-b24	Reserved	-	Read as "0", write as "0"	R/W
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	Reserved	-	Read as "0", write as "0"	R/W
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

15.5.4 Interrupt State Register 1 (DMA_INTSTAT1)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	BTC [7:0]							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	TC [7:0]							
Bit	Flag	Bit Name	Function											Read and write	
b31-b24	Reserved	-	Read as "0", write as "0"											R/W	
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	Reserved	-	Read as "0", write as "0"											R/W	
b7-b0	TC [7:0]	Transport completion interrupt bit	The interrupt occurs after the number of transmissions set by register MA_CNTx is completed 0: The channel was not interrupted 1: The channel is interrupted											R	

15.5.5 Interrupt Mask Register 0 (DMA_INTMASK0)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	MSKREQERR [7:0]							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	MSKTRNERR [7:0]							
Bit	Flag	Bit Name	Function											Read and write	
b31-b24	Reserved	-	Read as "0", write as "0"											R/W	
b23-b16	MSKREQERR [7:0]	Transmission request overflow interrupt mask	0: Nomaskable transport request overflow interrupt 1: Shielded transport request overflow interrupt											R/W	
b15-b8	Reserved	-	Read as "0", write as "0"											R/W	
b7-b0	MSKTRNERR [7:0]	Transmission error interrupt mask	0: Nomaskable transport error interrupt 1: Mask transmission error interrupt											R/W	

15.5.6 Interrupt Mask Register 1 (DMA_INTMASK1)

Reset value: 0x00000000

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

Bit	Flag	Bit Name	Function	Read and write
b31-b24	Reserved	-	Read as "0", write as "0"	R/W
b23-b16	MSKBTC [7:0]	Block transport complete interrupt mask	0: Nomaskable block transport complete interrupt 1: Block transport complete interrupt	R/W
b15-b8	Reserved	-	Read as "0", write as "0"	R/W
b7-b0	MSKTC [7:0]	Transport complete interrupt mask	0: Nomaskable transport complete interrupt 1: Shield transport complete interrupt	R/W

15.5.7 Interrupt Reset Register 0 (DMA_INTCLR0)

Reset value: 0x00000000

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

Bit	Flag	Bit Name	Function	Read and write
b31-b24	Reserved	-	Read as "0", write as "0"	R/W
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	R/W
b15-b8	Reserved	-	Read as "0", write as "0"	R/W
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	R/W

15.5.8 Interrupt Reset Register 1 (DMA_INTCLR1)

Reset value: 0x00000000

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

Bit	Flag	Bit Name	Function	Read and write
b31-b24	Reserved	-	Read as "0", write as "0"	R/W
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	R/W
b15-b8	Reserved	-	Read as "0", write as "0"	R/W
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	R/W

15.5.9 Channel Enable Register (DMA_CHEN)

Reset value: 0x00000000

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

Bit	Flag	Bit Name	Function	Read and write
b31-b8	Reserved	-	Read as "0", write as "0"	R/W
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.	R/W

15.5.10 Channel Enable Reset Register (DMA _ CHENCLR)

Reset value: 0x00000000

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

Bit	Flag	Bit Name	Function	Read and write
b31-b8	Reserved	-	Read as "0", write as "0"	R/W
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.	R/W

15.5.11 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
SWR CFGR EQ	Res							SWREQ [7:0]							

Bit/Bitfield	Flag	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

15.5.12 Channel Reset Control Register (DMA_RCFGCTL)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	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
-	-	-	-	RCFGCHS [3:0]				-	-	-	-	-	-	RCFGLP	RCFGEN

Bit	Flag	Bit Name	Function	Read and write
b31-b22	Reserved	-	Read as "0", write as "0"	R/W
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_SNEQCTLBx.SNSCNTB; when the source address is repeated, the remaining transfer times counter 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.	R/W
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 + (DNSDIST x 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.	R/W
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 + (SNSDIST x 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.	R/W
b15-b12	Reserved	-	Read as "0", write as "0"	R/W
b11-b8	RCFGCHS [3:0]	Reset channel selection	0x0: channel 0 0x1: channel 1 And so on 0x7: channel 7 Others: Reserved , setting prohibited	R/W
b7-b2	Reserved	-	Read as "0", write as "0"	R/W
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-bit21 of this register are all invalid.	R/W
b0	RCFGEN	Channel Reset Permission	0: Disable event-triggered channel configuration register forced update 1: Enable event-triggered channel configuration register forced update	R/W

Note:

- *Please set this register when DMA_EN.EN is 0, this register must be set before the first transmission of the reset channel.*

15.5.13 Transfer Request Status Register (DMA_REQSTAT)

Reset value: 0x00000000

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

Bit	Flag	Bit Name	Function	Read and write
b31-b17	Reserved	-	Read as "0", write as "0"	R/W
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	Reserved	-	Read as "0", write as "0"	R/W
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 1 state, the transmission request of this channel is input again, and a transmission request overflow error occurs, and the second request is ignored. 0: There is no transmission request for this channel 1: The channel has a transfer request	R

15.5.14 Channel State Observation Register (DMA _ CHSTAT)

Reset value: 0x00000000

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

Bit	Flag	Bit Name	Function	Read and write
b31-b24	Reserved	-	Read as "0", write as "0"	R/W
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	Reserved	-	Read as "0", write as "0"	R/W
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

15.5.15 Transmission Source Address Register (DMA _ SARx) (x = 0 ~ 7)

Reset value: 0x00000000

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	Flag	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=01, SAR [0] is invalid. When the transmission data width is 32 bits, that is, DMA_CHCTLx.HSIZE=1x, SAR [1:0] is invalid.											R/W	

15.5.16 Transport Destination Address Register (DMA _ DARx) (x = 0 ~ 7)

Reset value: 0x00000000

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	Flag	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=01, DAR [0] is invalid. When the transmission data width is 32 bits, that is, DMA_CHCTLx.HSIZE=1x, DAR [1:0] is invalid.	R/W

15.5.17 Data Control Register (DMA_DTCTLx) (x=0~7)

Reset value: 0x00000001

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
-	-	-	-	-	-	BLKSIZE [9:0]									
Bit	Flag		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.									R/W	
b15-b10	Reserved		-		Read as "0", write as "0"									R/W	
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.									R/W	

15.5.18 Repeat Region Size Register (DMA_RPTx) (x=0~7)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	DRPT [9:0]									
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	SRPT [9:0]									

Bit	Flag	Bit Name	Function	Read and write
b31-b26	Reserved	-	Read as "0", write as "0"	R/W
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 the HSIZE bit of the DMA_CHxCTL register.	R/W
b15-b10	Reserved	-	Read as "0", write as "0"	R/W
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 the HSIZE bit of the DMA_CHxCTL register.	R/W

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_CHxCTL register to be valid, and configure the SINC/DINC bit of the DMA_CHxCTL register to make the address update mode self-increment or self-decrement. If it is fixed, the address reload function is invalid.

The two registers DMA_RPTx and DMA_RPTBx share the same address and are used to define the size of the repeated area. Which one to use depends on whether the channel has reset enabled or not. DMA_RPTx is used when reset is not enabled, and DMA_PRTBx is used when reset is enabled.

15.5.19 Repeat area size register B (DMA_RPTBx) (x=0~7)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	DRPTB [9:0]									
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	SRPTB [9:0]									

Bit	Flag	Bit Name	Function	Read and write
b31-b26	Reserved	-	Read as "0", write as "0"	R/W
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 [9:0] must be greater than 0.	R/W
b15-b10	Reserved	-	Read as "0", write as "0"	R/W
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..	R/W

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.

The two registers DMA_RPTx and DMA_RPTBx share the same address and are used to define the size of the repeated area. Which one to use depends on whether the channel has reset enabled or not. DMA_RPTx is used when reset is not enabled, and DMA_PRTBx is used when reset is enabled.

15.5.20 Source Device Discontinuous Address Transfer Control Register (DMA_SNSEQCTLx) (x=0~7)

Reset value: 0x00000000

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	Flag	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.	R/W
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 image Figure 15-3. 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_CHxCTL.HSIZE) and the value of SOFFSET. Address offset=SOFFSET×(HSIZE(bit)/8) For example, when SOFFSET is set to 10 and HSIZE is a word (32bit), the address offset is 10×4=40; if HSIZE is a halfword (16bit), the offset is 10×2=20; if HSIZE is Byte (8bit), the offset is 10×1=10. The source address of the next transfer = the source address of the current transfer ± address offset	R/W

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.

The two registers DMA_SNSEQCTLx, DMA_SNSEQCTLBx share the same address and are used to define discontinuous transmission. Which one to use depends on whether the channel has reset enabled or not. DMA_SNSEQCTLx is used when reset is not enabled, and DMA_SNSEQCTLBx is used when reset is enabled.

15.5.21 Source Device Discontinuous Address Transfer Control Register B (DMA_SNSEQCTLBx) (x=0~7)

Reset value: 0x00000000

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	Flag	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. When the reset function of this channel valid (DMA_RCFGCTL.RCFGEN=1), the setting value of SNSCNTB [11:0] must be greater than 0.	R/W
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 imageFigure 15-4. 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_CHxCTL.HSIZE) and the value of SNSDIST. Address spacing=SNSDIST×(HSIZE(bit)/8) For example, when SNSDIST is set to 10 and HSIZE is word (32bit), the address spacing is 10×4=40; if HSIZE is halfword (16bit), the spacing is 10×2=20; if HSIZE is byte (8bit), the spacing is 10×1=10. The source address of the next transmission = the first address of the current source discontinuous area ± address distance	R/W

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.

The two registers DMA_SNSEQCTLx, DMA_SNSEQCTLBx share the same address and are used to define discontinuous transmission. Which one to use depends on whether the channel has reset enabled or not. DMA_SNSEQCTLx is used when reset is not enabled, and DMA_SNSEQCTLBx is used when reset is enabled.

15.5.22 Target Device Discontinuous Address Transfer Control Register (DMA_DNSEQCTLx) (x=0~7)

Reset value: 0x00000000

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	Flag	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.	R/W
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 imageFigure 15-3. 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_CHxCTL.HSIZE) and the value of DOFFSET. Address offset=DOFFSET×(HSIZE(bit)/8) For example, when DOFFSET is set to 10 and HSIZE is word (32bit), the address offset is 10×4=40; if HSIZE is halfword (16bit), the offset is 10×2=20; if HSIZE is Byte (8bit), the offset is 10×1=10. The target address of the next transfer = the target address of the current transfer ± address offset	R/W

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.

The two registers DMA_DNSEQCTLx, DMA_DNSEQCTLBx share the same address and are used to define discontinuous transmission. Which one to use depends on whether the channel has reset enabled or not. DMA_DNSEQCTLx is used when reset is not enabled, and DMA_DNSEQCTLBx is used when reset is enabled.

15.5.23 Target Device Discontinuous Address Transfer Control Register B (DMA_DNSEQCTLBx) (x=0~7)

Reset value: 0x00000000

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	Flag	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. When the reset function of this channel valid (DMA_RCFGCTL.RCFGGEN=1), the setting value of DNSCNTB [11:0] must be greater than 0.	R/W
b19-b0	DNSDIST [19:0]	Discontinuous target area address interval	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 imageFigure 15-4. 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_CHxCTL.HSIZE) and the value of DNSDIST. Address distance=DNSDIST×(HSIZE(bit)/8) For example, when DNSDIST is set to 10 and HSIZE is word (32bit), the address spacing is 10×4=40; if HSIZE is halfword (16bit), the spacing is 10×2=20; if HSIZE is byte (8bit), the spacing is 10×1=10. The target address of the next transmission = the first address of the current target discontinuous area ± address distance	R/W

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.

The two registers DMA_DNSEQCTLx, DMA_DNSEQCTLBx share the same address and are used to define discontinuous transmission. Which one to use depends on whether the channel has reset enabled or not. DMA_DNSEQCTLx is used when reset is not enabled, and DMA_DNSEQCTLBx is used when reset is enabled.

15.5.24 Chain Pointer Register (DMA_LL Px) (x=0~7)

Reset value: 0x00000000

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]														-	-	
Bit	Flag	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									R/W	
b1-b0	Reserved	-				Read as "0", write as "0"									R/W	

15.5.25 Channel Control Register (DMA _ CHxCTL) (x = 0 ~ 7)

Reset value: 0x00001000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
HPROT [3:2]		-	IE	LLPR UN	LLPE N	HSIZE [1:0]		DNS EQEN	SNSE QEN	DRP TEN	SRPT EN	DINC [1:0]		SINC [1:0]	
Bit		Flag		Bit Name		Function								Read and write	
b31~b16		Reserved		-		Read as "0", write as "0"								R/W	
b15~b14		HPROT [3:2]		HPROT value in AHB bus		The HPORT [3:2] bits control the HPROT [3:2] of the AHB bus. When using DMA to access SMC , please set HPROT [3:2] to 2'b11; In other cases, please set HPROT [3:2] to 2'b00.								R/W	
b13		Reserved		-		Read as 0, write as 0								R/W	
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								R/W	
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								R/W	
b10		LLPEN		Chain transmission enable		0: Invalid chain transmission 1: Linkage transmission is effective								R/W	
b9~b8		HSIZE [1:0]		Width of transmitted data		00: 8bit 01: 16bit 10, 11: 32bit								R/W	
b7		DNSEQEN		Destination Address Discontinuous Transfer Enable		0: Do not allow discontinuous address transmission 1: Allow discontinuous address transmission								R/W	
b6		SNSEQEN		Source address discontinuous transfer enable		0: Do not allow discontinuous address transmission 1: Allow discontinuous address transmission								R/W	
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								R/W	
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								R/W	
b3~b2		DINC [1:0]		Update mode of destination address		00: Fixed 01: Incremental 10, 11: decrement								R/W	
b1~b0		SINC [1:0]		Source address update		00: Fixed 01: Incremental 10, 11: decrement								R/W	

15.5.26 Channel Monitor Registers (DMA_MONSARx, DMA_MONDARx, DMA_MONDTCTLx, DMA_MONRPTx, DMA_MONSNSEQCTLx, DMA_MONDSEQCTLx) (x=0~7)

These monitoring registers correspond to the corresponding channel configuration registers, and the register bit configurations are consistent, but all are read-only registers.

The channel configuration register remains unchanged before and after the DMA transfer, and the channel monitoring register will be updated after each transfer corresponding to a request is completed by the DMA, that is, after each data block transfer is completed. The update content and method are as follows:

- DMA_MONSARx.SAR [31:0], DMA_MONDARx.DAR [31:0]: Update the address of the next transmission according to the fixed/increment/decrement/reload/discontinuous jump mode set by the channel configuration register.
- DMA_MONDTCTLx.CNT [15:0]: Subtract 1, if it is already 0, keep it as 0.
- DMA_MONRPTx.SRPT [9:0] , DRPT [9:0] : When the channel reset is 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.
- DMA_MONSNSEQCTLx.SNSCNT [11:0] , DMA_MONDNSEQCTLx.DNSCNT [11:0] : When the channel reset is invalid, subtract the block size, and when it is reduced to 0 , reload the original setting value of DMA_SNSEQCTLx/DMA_DNSEQCTLx. When reset is effective, it decreases by 1, and when it decreases to 0, the original setting value is reloaded.
- Monitor register bits other than the above remain the same as the configuration register.

15.6 Precautions

- DMA registers only support 32bit read and write, and 8/16bit read and write operations are invalid.
- Prohibiting DMA access to memory regions that may cause bus errors (such as reserved address spaces, protected address spaces, etc.).

16 Voltage Comparator (CMP)

16.1 Introduction

The voltage comparator (CMP) is a peripheral module that compares the magnitudes of two analog voltages and provides a set of two comparison channels, CMP1 and CMP2.

CMP has the following main features:

- All comparison channels can perform common comparison independently
- The combination of two comparison channels in the same group can realize window comparison
- Multiple analog inputs (IO , PGA , DAC)
- Configurable hysteresis voltage
- Comparison result output is equipped with noise filter
- Comparison result output with blank window function
- The comparison result change edge can generate interrupts and peripheral trigger events
- The comparison results can wake up the system's low-power mode
- Output remains on when software reset or watchdog reset occurs
- The comparison result can be output to an external pin (CMP_OUT)
- The comparison results can be output to the emergency brake module

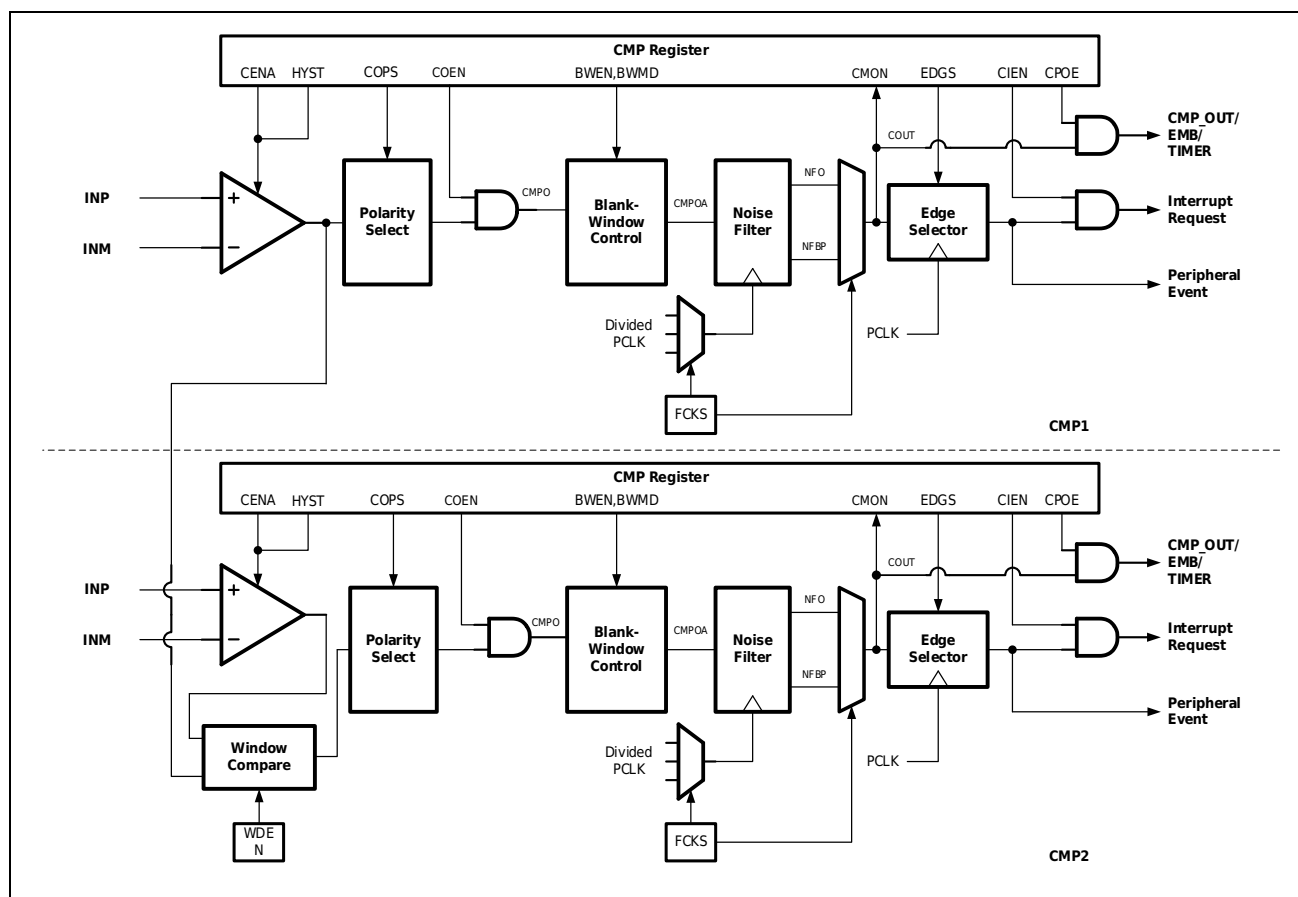


Table 16-1 CMP Input and Output Pins

Pin Name	Input/Output	Function
AVCC	I	Analog Power
AVSS	I	Analog Ground
PA0	I	CMP1_INM1
	O	CMP1_OUT
PA1	I	CMP1_INP1/CMP2_INP1
PA2	I	CMP1_INM2/CMP2_INM1
	O	CMP2_OUT
PA3	I	CMP1_INP2/CMP2_INP2
PA6	O	CMP1_OUT
PA7	O	CMP2_OUT
PA11	O	CMP1_OUT
PA12	O	CMP2_OUT
PB6	O	CMP1_OUT
PB7	O	CMP2_OUT
PB9	O	CMP2_OUT
PB15	O	CMP1_OUT
PF1	O	CMP1_OUT

16.3 Functional Description

16.3.1 Input Selection

CMP1~2 are configured according to the following table.

Table 16-2 CMP Analog Input Comparison Table

CMP1					CMP2				
S/N	Register PMSR.CVSL	Positive Input INP	Register PMSR.RVSL	Negative input INM	SN	Register PMSR.CVSL	Positive Input INP	Register PMSR.RVSL	Negative input INM
1	0001b	PA1	0000000001b	-	1	0001b	PA1	0000000001b	-
2	0010b	PA3	0000000010b	-	2	0010b	PA3	0000000010b	-
3	-	-	0000000100b	-	3	-	-	0000000100b	-
4	-	-	0000001000b	-	4	-	-	0000001000b	-
5	-	-	0000010000b	DA1O1 ^{note1}	5	-	-	0000010000b	DA1O1 ^{note1}
6	-	-	0000100000b	DA1O2 ^{note1}	6	-	-	0000100000b	DA1O2 ^{note1}
7	-	-	0001000000b	PA0	7	-	-	0001000000b	PA2
8	-	-	0010000000b	PA2	8	-	-	0010000000b	-
9	-	-	0100000000b	-	9	-	-	0100000000b	-
10	-	-	1000000000b	-	10	-	-	1000000000b	-

Note 1: When selecting the DAC internal path, turning off the analog output of the DAC channel can release the corresponding analog IO.

16.3.2 Normal Comparison

Each comparison channel can perform normal comparison independently. Taking CMP1 as an instance, the setting steps are as follows:

1. Turn off the module stop function of CMP1 (write 0 to PWR_FCG3.bit8).
2. Set the CVSL bit in the CMP1_PMSR register to select the positive terminal voltage;
Set the RVSL bit in the CMP1_PMSR register to select the negative terminal voltage.
3. Set the HYST bit in the CMP1_MDR register to select the hysteresis voltage.
Set the CENA bit in the CMP1_MDR register to 1 to enable the comparator.
4. Wait for the comparator to stabilize for tCMP. (For specific values, please refer to the product data sheet)
5. Set the FCKS bit of the CMP1_FIR register to set the number of Filtering sampling periods;
Set the EDGS and CIEN bits in the CMP1_FIR register to set the interrupt.
6. Set the COPS and CPOE bits in the CMP1_OCR register to set the external output.
7. Set the COEN bit in the CMP1_OCR register to 1 to enable the comparator output.

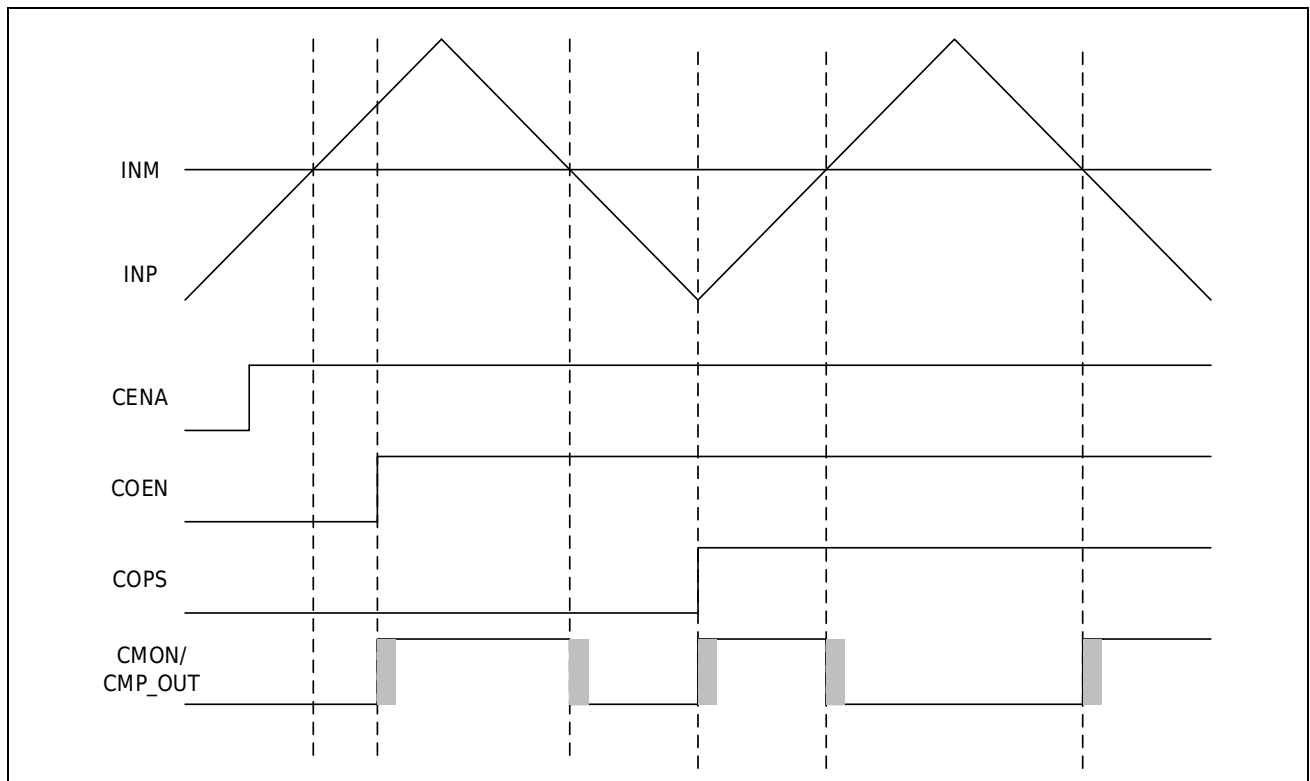


Figure 16-2 Voltage Comparison Diagram

As shown in Figure 16-2, 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 CMPx_OMR 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 change edge of CMON will be at least 6 CMPCLK periods later (shown in gray in the figure) due to synchronization and filtering sampling.

16.3.3 Window Comparison

The combination of two comparison channels in the same group can realize window comparison, and CMP2 completes the monitoring, filtering, interruption and output of the comparison results.

Taking the combination of CMP1 and CMP2 as an instance, the setting steps are as follows:

1. Disable the module stop function of CMP1 and 2 (write 0 to PWR_FCG3.bit8).
2. Set the CVSL bit of the CMP1_PMSR register to select the positive terminal voltage of CMP1;
Set the RVSL bit in the CMP1_PMSR register to select the negative terminal voltage of CMP1 as the lower limit of the window.
3. Set the CVSL bit of the CMP2_PMSR register to select the positive terminal voltage of CMP2;
(Note that the positive inputs of CMP1 and CMP2 must be consistent)
Set the RVSL bit in the CMP2_PMSR register to select the negative terminal voltage of CMP2 as the upper limit of the window.
4. Set the WDEN bit in the CMP2_MDR register to 1 to select the window comparison mode.

5. Set the HYST bit in the CMP1_MDR and CMP2_MDR registers to select the hysteresis voltage;
(Note the window change caused by hysteresis voltage)
Set the CENA bit in the CMP1_MDR and CMP2_MDR registers to 1 to enable the comparator.
6. Wait for the stabilization time tCMP. (For specific values, please refer to the product data sheet)
7. Set the FCKS bit of the CMP2_FIR register to set the number of Filtering sampling periods;
8. Set the EDGS and CIEN bits of the CMP2_FIR register to set the interrupt.
9. Set the COPS and CPOE bits of the CMP2_OCR register to set external output;
10. Set the COEN bit in the CMP1_OCR and CMP2_OCR registers to 1 to enable the comparator output.

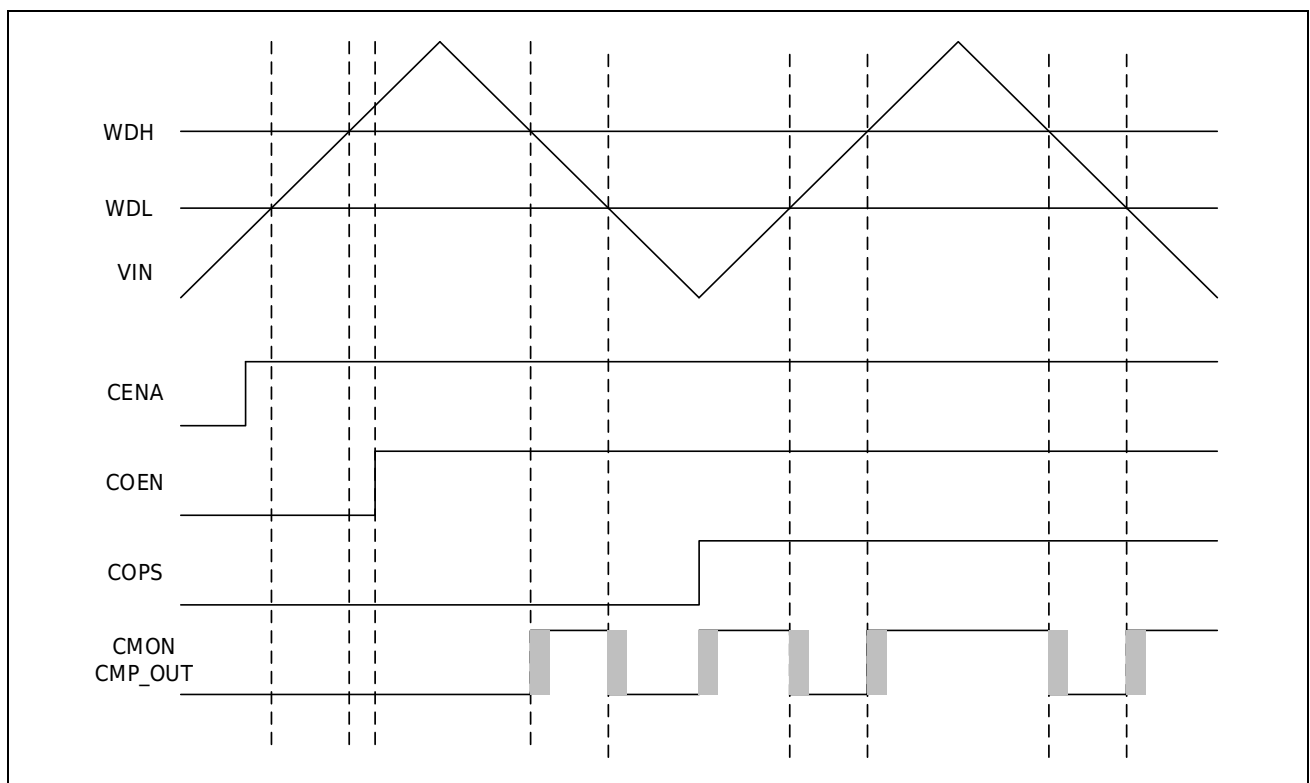


Figure 16-3 Diagram of Window Comparison

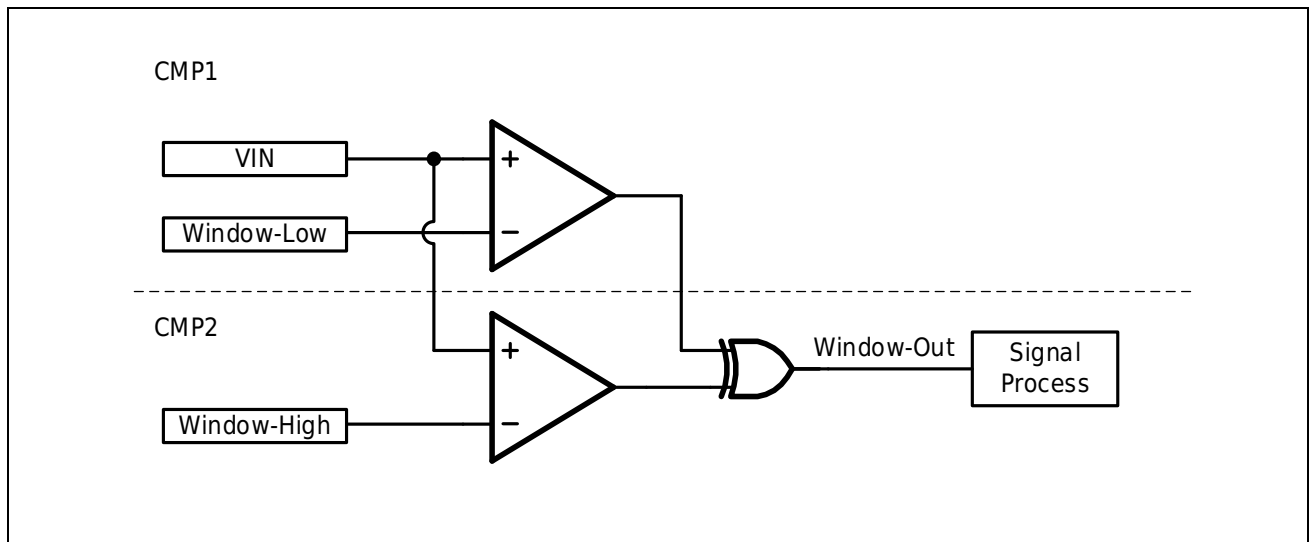


Figure 16-4 Window Comparison Schematic

As shown in Figure 16-3, in the positive phase output, after CMP is allowed to output, if the positive terminal voltage VIN is within the window composed of the two negative terminal voltages (WDH , WDL), the CMON bit of the CMP2_OMR register is 1; if VIN is outside the window, 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 periods (shown in the gray part in the figure) due to synchronization and Filtering sampling.

16.3.4 Blank Window

PWM signal output by the timer can be used as a blank window (Blank-Window , also known as timer window) signal to control the output of the comparison result CMPO , which is used in application scenarios such as timer PWM zero-crossing detection.

Window output has two modes: level and edge.

■ Level Mode

When the window output is enabled (the BWEN bit in the CMPx_OCR register is 1) and the level mode is selected (the BWMD bit in the CMPx_OCR register is 0), one or more timer PWMs are selected as Blank-Window signals through the CTWS bit in the CMPx_BWSR1 register, and the effective level is selected for each Blank-Window signal through the CTWP bit. In this way, the window output CMPOA will be fixed to the state set by the BWOL bit of the CMPx_OCR register when Blank-Window valid.

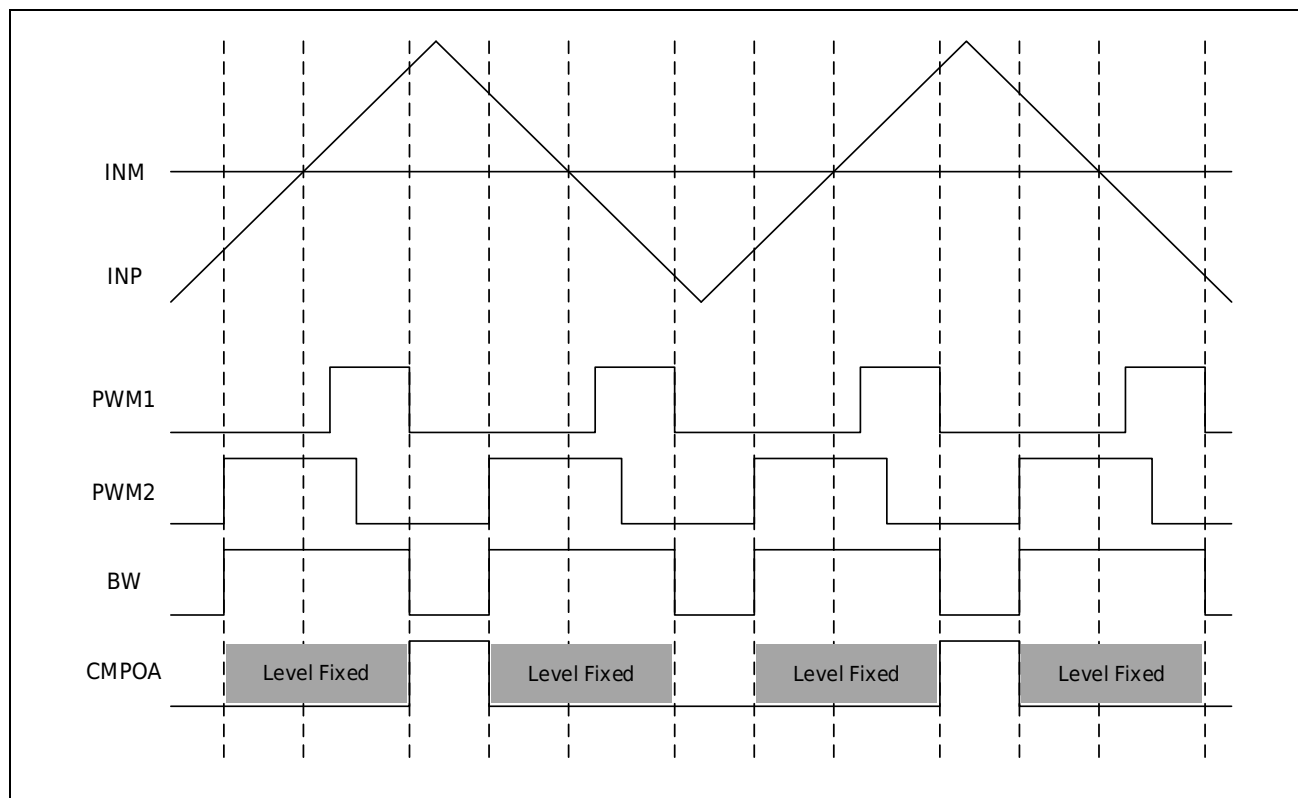


Figure 16-5 Window Output Schematic 1

As shown in Figure 16-5, select PWM1, 2 as Blank-Window and both are valid at high level. At the same time, when Blank-Window is valid, the output state of CMPOA is low level. Therefore, when one of PWM1 and 2 is at a high level, the CMPOA output is fixed to a low level, and when both PWM1 and 2 are at a low level, the CMPOA outputs the comparison result normally.

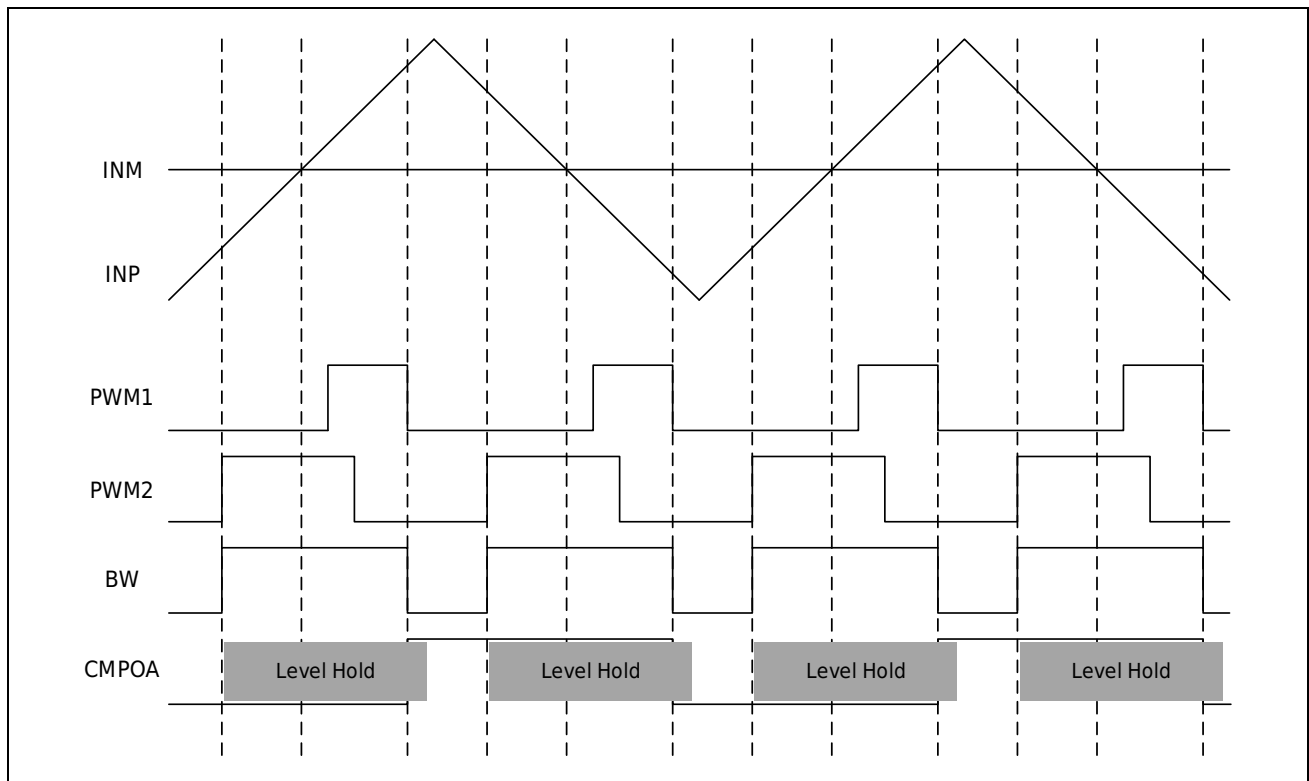


Figure 16-6 Window Output Schematic Diagram II

As shown in Figure 16-6, select PWM1, 2 as Blank-Window and both are valid at high level. At the same time, the CMPOA output state is maintained when Blank-Window is valid. Therefore, when one of PWM1 and 2 is at a high level, CMPOA maintains the previous state; when both PWM1 and 2 are at a low level, CMPOA outputs the comparison result normally, but it will be 1 to 3 CMPCLK periods late. This is caused by synchronization and sampling.

■ Edge Mode

When the window output is enabled (the BWEN bit of the CMPx_OCR register is 1) and the edge mode is selected (the BWMD bit of the CMP_OCR register is 1), one or more timer PWMs are selected as trigger sources to generate Blank-Window signals through the CTWS bit of the CMPx_BWSR1 register, and the effective edge of the trigger source is selected through the TWEG bit of the CMPx_BWSR2 register. In this way, the window output CMPOA will be fixed to the state set by the BWOL bit of the CMPx_OCR register at the effective edge of the trigger source, and maintained for N CMPCLK periods set by the MSKW bit of the CMPx_BWSR2 register.

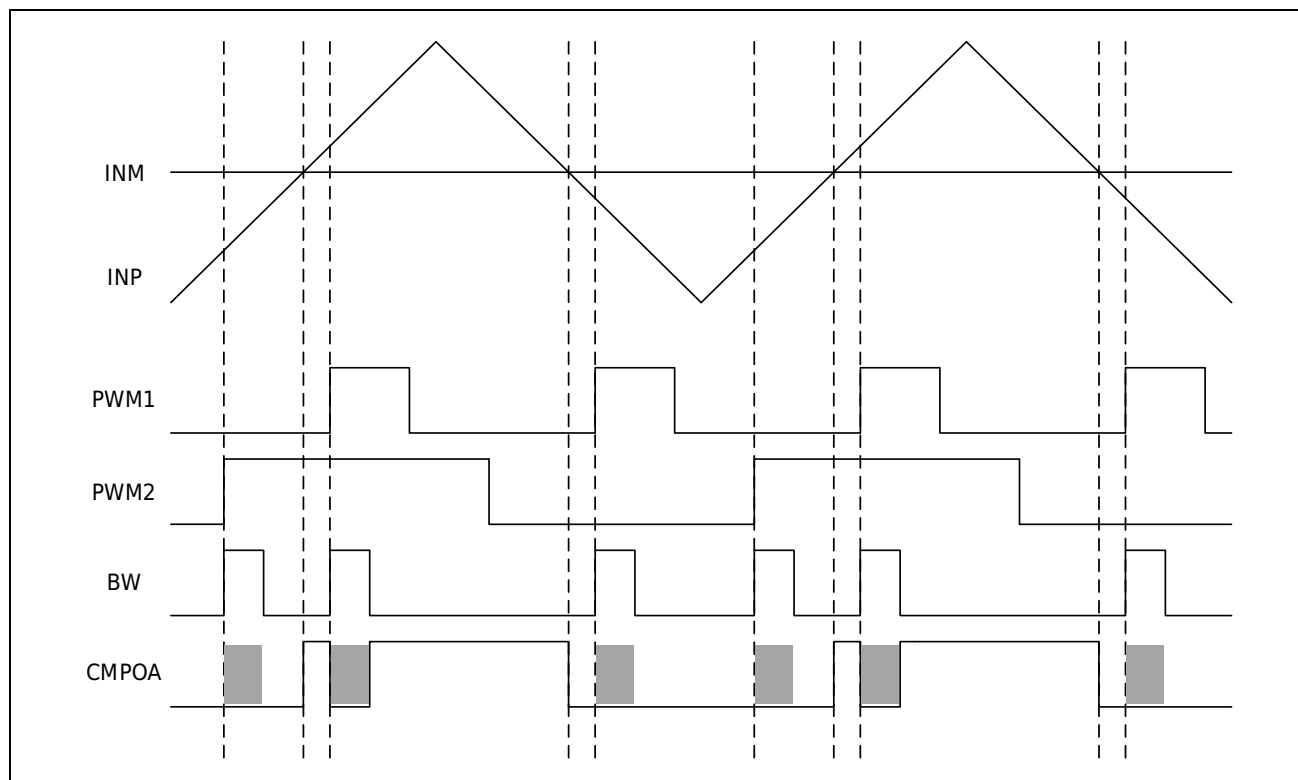


Figure 16-7 Window Output Schematic Diagram 3

As shown in Figure 16-7, if PWM1,2 is selected as the trigger source and the rising edge is valid, and Blank-Window is valid, the CMPOA output state is low level. Therefore, a Blank-Window signal with adjustable width is generated at each rising edge of PWM1,2. When Blank-Window is valid, CMPOA output is fixed to low level. When Blank-Window is invalid, CMPOA outputs comparison result normally.

16.3.5 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 000b.

16.3.6 Noise filtering

Each comparison channel includes a noise filter to filter the comparator output CMPOA. The noise filter samples the comparator output CMPOA in each sampling clock period. If the same value is sampled three times in succession, the value is output to COUT in the next sampling clock period. Otherwise, COUT remains unchanged. The sampling clock can be selected by the FCKS bit in the CMPx_FIR register. When the FCKS bit is set to 00b, the noise filter is turned off.

16.3.7 Interrupt Event

Each comparison channel can generate an interrupt request through the output COUT. the CIEN bit in the CMPx_FIR register must be set to 1 , and the edge that generates the interrupt on COUT must be set using the EDGS bit (a value other than 2'b00). Since COUT is the original comparison result output after polarity selection, blank window, and noise filtering, please pay attention to the edge where the interrupt actually occurs when using it.

COUT can also wake up and stop low power mode. Before entering the stop mode, the interrupt must be enabled (CIEN=1) and the digital filter and edge detection must be disabled. The system will be awakened and an interrupt will be generated when COUT changes from low to high. After the system returns to working mode, please set the interrupt to invalid (CIEN=0) before performing subsequent operations.

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.

16.3.8 External Output

COUT can be output to the outside of CMP , such as CMPx_OUT pin and peripherals such as EMB and TIMER. The CPOE bit in the CMPx_OCR register is used to enable the external output. Since the pins corresponding to CMPx_OUT are general ports (GPIO) by default, they need to be set to the CMPx_OUT function when used.

16.4 Precautions

16.4.1 Module Stop Function

CMP has module stop function, by setting module stop register can turn off the digital part of the module.

CMP1 and CMP2 share a module stop register bit PWR_FCG3.bit8.

All comparison channels are initially in the stopped state, and their respective registers can only be accessed when the setting module is working. Please refer to the low power consumption section for related register settings.

16.4.2 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 CMPx_MDR register to "0".

16.4.3 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 CMPx_MDR register to 0 before entering stop low-power mode.

16.5 Register Description

CMP1 reference address: 0x40038400

CMP2 reference address: 0x40038500

Table 16-3 CMP Register List

Register Name	Symbol	Offset Address	Bit Width	Reset Value
Comparator Operating Mode Register	CMPx_MDR	0x00	8	0x00
Comparator filter interrupt register	CMPx_FIR	0x01	8	0x00
Comparator output control register	CMPx_OCR	0x02	8	0x00
Comparator result monitoring register	CMPx_OMR	0x03	8	0x00
Comparator Input Select Register	CMPx_PMSR	0x04	32	0x00000000
Comparator output blank window setting register 1	CMPx_BWSR1	0x10	32	0x00000000
Comparator output blank window setting register 2	CMPx_BWSR2	0x14	16	0x0000

In the table, x=1~2.

All registers are retained across software reset and watchdog reset.

16.5.1 Comparator Operating Mode Register (CMPx_MDR, x=1~2)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Reserved			WDEN	HYST [2:0]			CENA

Bit	Flag	Bit Name	Function	Read and write
b7~b5	Reserved	-	Read at "0", write at "0"	R/W
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: General comparison mode 1: Window comparison mode	R/W
b3~b1	HYST [2:0]	Hysteresis voltage selection	The values listed below are typical values. 000: 0mv (No hysteresis) 001: 10mv 010: 20mv 011: 30mv 100: 40mv 101: 50mv 110: 60mv 111: 70mv	R/W
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.	R/W

16.5.2 Comparator Filter Interrupt Register (CMPx_FIR, x=1~2)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
CRF	CFF	EDGS [1:0]		CIEN	FCKS [2:0]		

Bit	Flag	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.	R/W
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.	R/W
b5~b4	EDGS [1:0]	Comparator output edge Detection selection	0 0: Do not detect the edge of the comparator output COUT 0 1: Detect the rising edge of the comparator output COUT 1 0: Detect the falling edge of the comparator output COUT 1 1: 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.	R/W
b3	CIEN	Comparator Interrupt Permission	0: Disable comparator interrupt 1: Allow the comparator to interrupt	R/W
b2~b0	FCKS [2:0]	Comparator Result Filtering Sample Selection	0 0 0: No noise filter 0 0 1: Use noise filter and sample through PCLK0 0 1 0: Use noise filter, sample by (PCLK0)/2 0 1 1: Use noise filter, sample by (PCLK0)/4 1 0 0: Use noise filter, sample by (PCLK0)/8 1 0 1: Use noise filter and sample by (PCLK0)/16 1 1 0: Use noise filter, sample by (PCLK0)/32 1 1 1: Use noise filter and sample by (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 '00b' 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.	R/W

16.5.3 Comparator Output Control Register (CMPx_OCR, x=1~2)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
BWOL [1:0]		BWMD	BWEN	Reserved	CPOE	COPS	COEN

Bit	Flag	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)	R/W
b5	BWMD	Blank-Window Mode	0: Level mode 1: Edge mode	R/W
b4	BWEN	Blank-Window Enable	0: Disable the BLANK-WINDOW function 1: Turn on the BLANK-WINDOW function	R/W
b3	Reserved	-	Read at "0", write at "0"	R/W
b2	CPOE	Comparator external output allows	0: Disable output to EMB/TIMER/VCOUT pin 1: Enable output to EMB/TIMER/VCOUT pin	R/W
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.	R/W
b0	COEN	Comparator output permissible	0: Disable comparator output (comparator output fixed to low level) 1: Allow comparator output	R/W

16.5.4 Comparator Result Monitoring Register (CMPx_OMR, x=1~2)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Reserved							CMON

Bit	Flag	Bit Name	Function	Read and write
b7~b1	Reserved	-	Read at "0", write at "0"	R/W
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

16.5.5 Comparator Input Selection Register (CMPx_PMSR, x=1~2)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24
Reserved							
b23	b22	b21	b20	b19	b18	b17	b16
Reserved				CVSL [3:0]			
b15	b14	b13	b12	b11	b10	b9	b8
Reserved						RVSL [9:8]	
b7	b6	b5	b4	b3	b2	b1	b0
RVSL [7:0]							

Bit	Flag	Bit Name	Function	Read and write
b31~b20	Reserved	-	Read at "0", write at "0"	R/W
b19~b16	CVSL [3:0]	Positive input selection	Select the positive input voltage of CMP. 0000: No positive voltage input 0001: Select INP1 as the positive terminal voltage (See Table 16-2 for selected input) 0010: Select INP2 as the positive terminal voltage (See Table 16-2 for selected input) Other: prohibitions Note 1	R/W
b15~b10	Reserved	-	Read at "0", write at "0"	R/W
b9~b0	RVSL [9:0]	Negative input selection	Select the negative input voltage of CMP. 0000000000: No negative voltage input 0000000001: Select INM1 as the negative terminal voltage (Selected inputs are shown in Table 16-2.) 0000000010: Select INM2 as the negative terminal voltage (Selected inputs are shown in Table 16-2.) 0000000100: Select INM3 as the negative terminal voltage (Selected inputs are shown in Table 16-2.) 0000001000: Select INM4 as the negative terminal voltage (Selected inputs are shown in Table 16-2.) 0000010000: Select INM5 as the negative terminal voltage (Selected inputs are shown in Table 16-2.) 0000100000: Select INM6 as the negative terminal voltage (Selected inputs are shown in Table 16-2.) 0001000000: Select INM7 as the negative terminal voltage (Selected inputs are shown in Table 16-2.) 0010000000: Select INM8 as the negative terminal voltage (Selected inputs are shown in Table 16-2.) 0100000000: Select INM9 as the negative terminal voltage (Selected inputs are shown in Table 16-2.) 1000000000: Select INM10 as the negative terminal voltage (Selected inputs are shown in Table 16-2.) Other: prohibitions Note 1	R/W

Note 1: Please rewrite when the comparator output is disabled (i.e. COEN bit is "0"). After rewriting each time, please wait for 300ns before enabling the comparator output (i.e., COEN bit is set to "1"). Rewriting may cause interrupts or peripheral trigger events, so please set this register when the interrupt is disabled or the peripheral trigger function is invalid. After the register is set, clear the corresponding interrupt flag.

16.5.6 Comparator Output Blank Window Setting Register 1 (CMPx_BWSR1, x=1~2)

Reset value: 0x00000000

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	Flag	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 used to turn off the comparator output. (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.	R/W
b15~b0	CTWS15~0	Window signal enable	When Blank-Window is valid (CMP_OCR.BWEN=1), select the window signal used to turn off the comparator output. (The window signal corresponding to each bit is shown in Table 16-4) 0: Window signal is invalid 1: Window signal is valid	R/W

Table 16-4 Timer Window PWM List

CTWS bit	CMP1	CMP2
15	TIM4_1_OWL	TIM4_1_OWL
14	TIM4_1_OWH	TIM4_1_OWH
13	TIM4_1_OVL	TIM4_1_OVL
12	TIM4_1_OVH	TIM4_1_OVH
11	TIM4_1_OUL	TIM4_1_OUL
10	TIM4_1_OUH	TIM4_1_OUH
9	TIM6_7_PWMA	TIM6_8_PWMA
8	TIM6_5_PWMA* ¹	TIM6_6_PWMA* ¹
7	TIM6_3_PWMA* ¹	TIM6_4_PWMA* ¹
6	TIM6_1_PWMA* ¹	TIM6_2_PWMA* ¹
5	TIMA_2_PWM3	TIMA_4_PWM3
4	TIMA_2_PWM2	TIMA_4_PWM2
3	TIMA_2_PWM1	TIMA_4_PWM1
2	TIMA_1_PWM3	TIMA_3_PWM3
1	TIMA_1_PWM2	TIMA_3_PWM2
0	TIMA_1_PWM1	TIMA_3_PWM1

Note 1: High-precision PWM mode is optional.

16.5.7 Comparator Output Blank Window Setting Register 2 (CMPx_BWSR2, x=1~2)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8
-	-	-	-	-	-	TWEG [1:0]	
b7	b6	b5	b4	b3	b2	b1	b0
MSKW [7:0]							

Bit	Flag	Bit Name	Function	Read and write
b15~b10	Reserved	-	Read at "0", write at "0"	R/W
b9~b8	TWEG [1:0]	Window Signal Edge	When Blank-Window is in edge mode (CMP_OCR.BWMD=1), set the effective edge of the window signal used to turn off the comparator output. 0 0: No valid edge 0 1: The comparator output is turned off at the rising edge of the window signal 1 0: Comparator output is turned off at the falling edge of the window signal 1 1: The comparator output is turned off at the changing edge of the window signal	
b7~b0	MSKW	BLANK-WINDOW effective width	When Blank-Window is in edge mode (CMP_OCR.BWMD=1), it sets the width of the 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 periods set by the MSKW bit.	R/W

17 Analog-to-Digital Conversion Module (ADC)

17.1 Introduction

12-bit ADC is an analog-to-digital converter with successive approximation. This MCU is equipped with 3 ADC units, each unit supports 18 channels and 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.

ADC Main Characteristics

- 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 system clock HCLK. In this case, the frequency ratio of PCLK4:
 $PCLK4:PCLK2=1:1$
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
 - Simulation watchdog, monitoring conversion results
 - The ADC module can be set to stop when not in use
- Analog input channel
 - A total of 27 external analog inputs, a single ADC unit supports up to 18 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
- 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 DMA and only carry DMA product support

17.2 ADC System Block Diagram

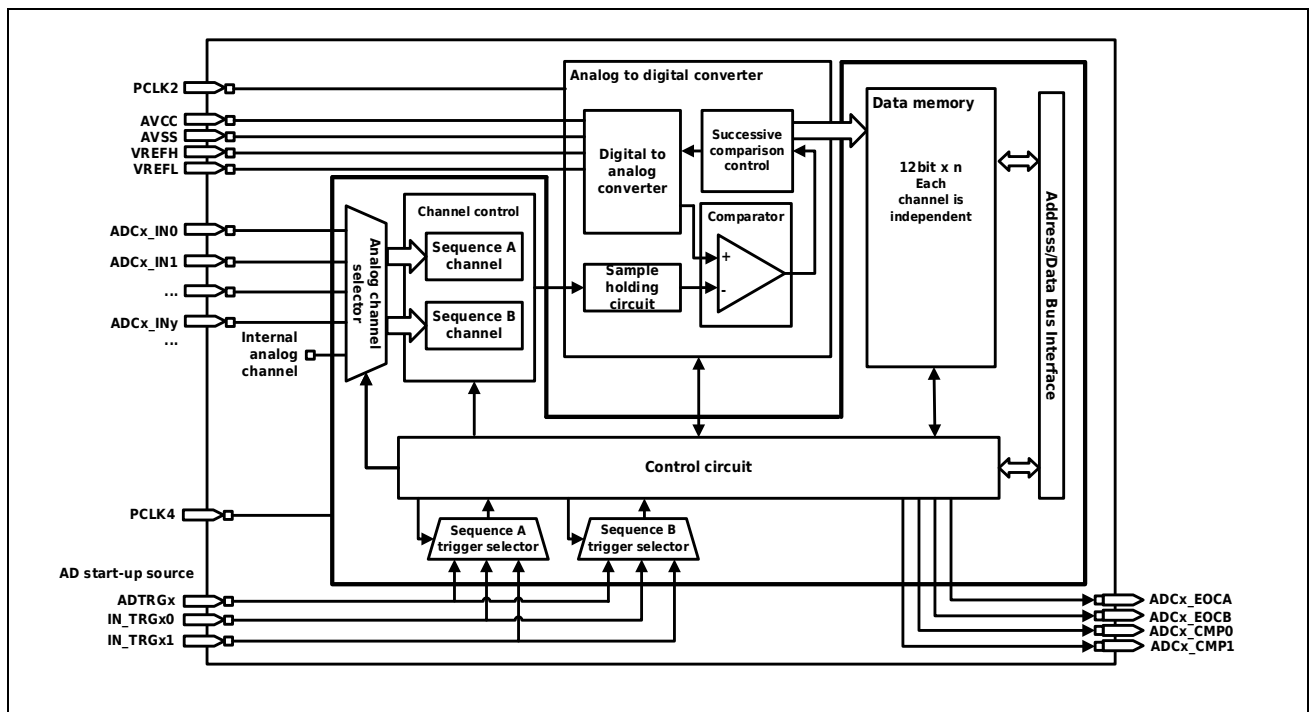


Figure 17-1 ADC Block Diagram

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 17-1 Specifications of Each ADC Unit

Item		Unit 1 (ADC1)	Unit 2 (ADC2)	Unit 3 (ADC3)
Power supply		AVCC		
		AVSS		
The reference voltage		VREFL		
		VREFH		
Analog channels *1	CH0	ADC123_IN0	ADC123_IN0	ADC123_IN0
	CH1	ADC123_IN1	ADC123_IN1	ADC123_IN1
	CH2	ADC123_IN2	ADC123_IN2	ADC123_IN2
	CH3	ADC123_IN3	ADC123_IN3	ADC123_IN3
	CH4	ADC12_IN4	ADC12_IN4	ADC3_IN4
	CH5	ADC12_IN5	ADC12_IN5	ADC3_IN5
	CH6	ADC12_IN6	ADC12_IN6	ADC3_IN6
	CH7	ADC12_IN7	ADC12_IN7	ADC3_IN7
	CH8	ADC12_IN8	ADC12_IN8	-
	CH9	ADC12_IN9	ADC12_IN9	-
	CH10	ADC123_IN10	ADC123_IN10	ADC123_IN10
	CH11	ADC123_IN11	ADC123_IN11	ADC123_IN11
	CH12	ADC123_IN12	ADC123_IN12	ADC123_IN12
	CH13	ADC123_IN13	ADC123_IN13	ADC123_IN13
	CH14	ADC12_IN14	ADC12_IN14	-
	CH15	ADC12_IN15	ADC12_IN15	-
	CH16	-	-	-
	CH17	-	-	ADC3_IN17
	CH18	-	-	ADC3_IN18
	CH19	-	-	ADC3_IN19
	CH20	ADC12_IN20	ADC12_IN20	ADC3_IN20
	CH21	Internal Reference Voltage ADVREF	Internal Reference Voltage ADVREF	ADC3_IN21
	CH22	-	-	Internal Reference Voltage ADVREF
Programmable gain Amplifier PGA	PGA	ADC12_IN7	ADC12_IN8	-
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.

17.3 Functional Description

17.3.1 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.

17.3.2 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. Please refer to the register description for details (only the first 16 channels support mapping). Unless otherwise specified in this chapter, channel n or CHn refers to a virtual channel.

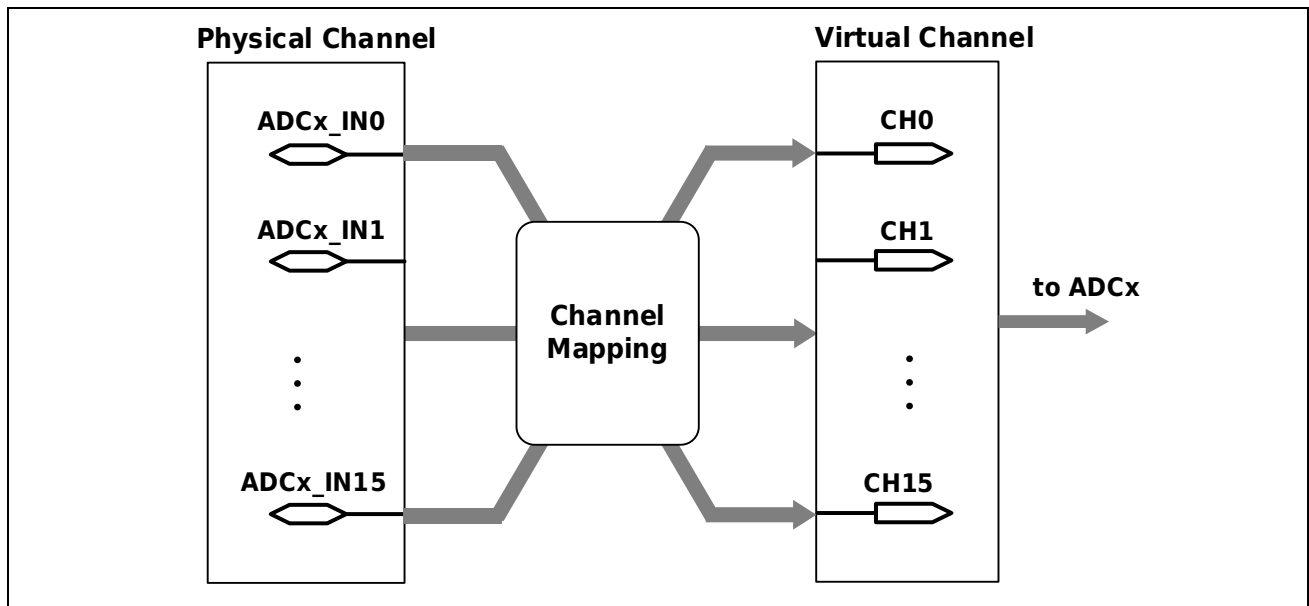


Figure 17-2 Channel Mapping Diagram

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.

In addition, most physical channels can be input to multiple ADC module units, such as ADC123_IN0 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.

17.3.3 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.

17.3.5 Sequence A Continuous Scanning Mode

The ADC control register ADC_CR0.MS [1:0] is set to 01b 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.

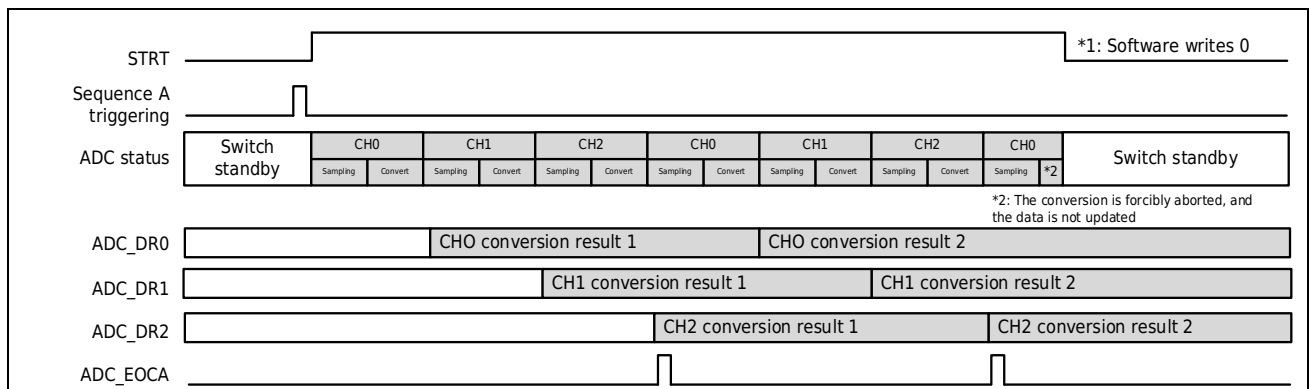


Figure 17-4 Continuous Scanning

Sequence A Continuous Scan Mode software flow:

1. Confirm that ADC_STR.STRT is 0, ADC is in conversion standby state.
2. The ADC control register ADC_CR0.MS [1:0] is set to 01b to select the sequence A continuous scan mode.
3. Set the sequence A channel select register ADC_CHSELRA.
4. Set the sampling time register ADC_SSTR.
5. ADC_STR.STRT Write 1 software trigger sequence A or set register ADC_TRGSR select sequence A trigger condition.
6. Query sequence A conversion end flag bit EOCAF.
7. Read each channel data register ADC_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.

17.3.6 Double Sequence Scanning Mode

The ADC control register ADC_CR0.MS [1:0] is set to 10b or 11b 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 [1: 0] = 10b, sequences A and B are equivalent to two independent single-scan sequences. MS [1: 0] = 11b sequence A is continuous scan 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.EOCFA 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_ISR.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 17-2 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.	

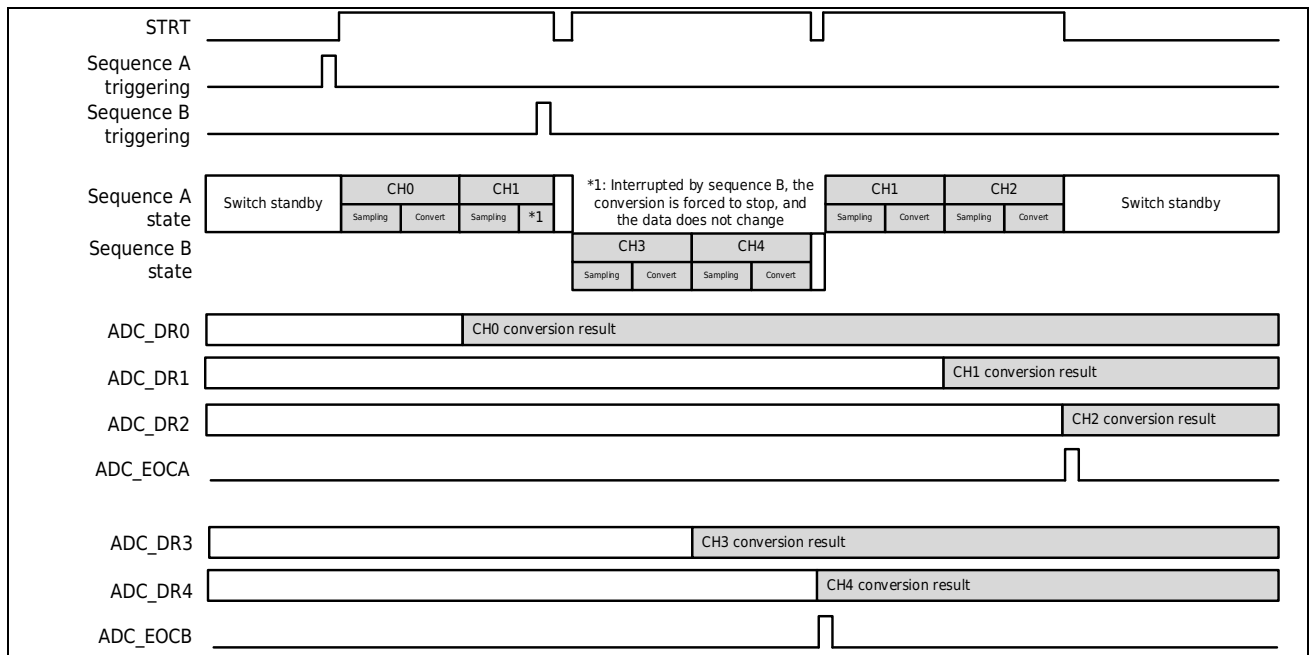


Figure 17-5 Double Sequence Scan Mode (Sequence A Restarts from Interrupted Channel)

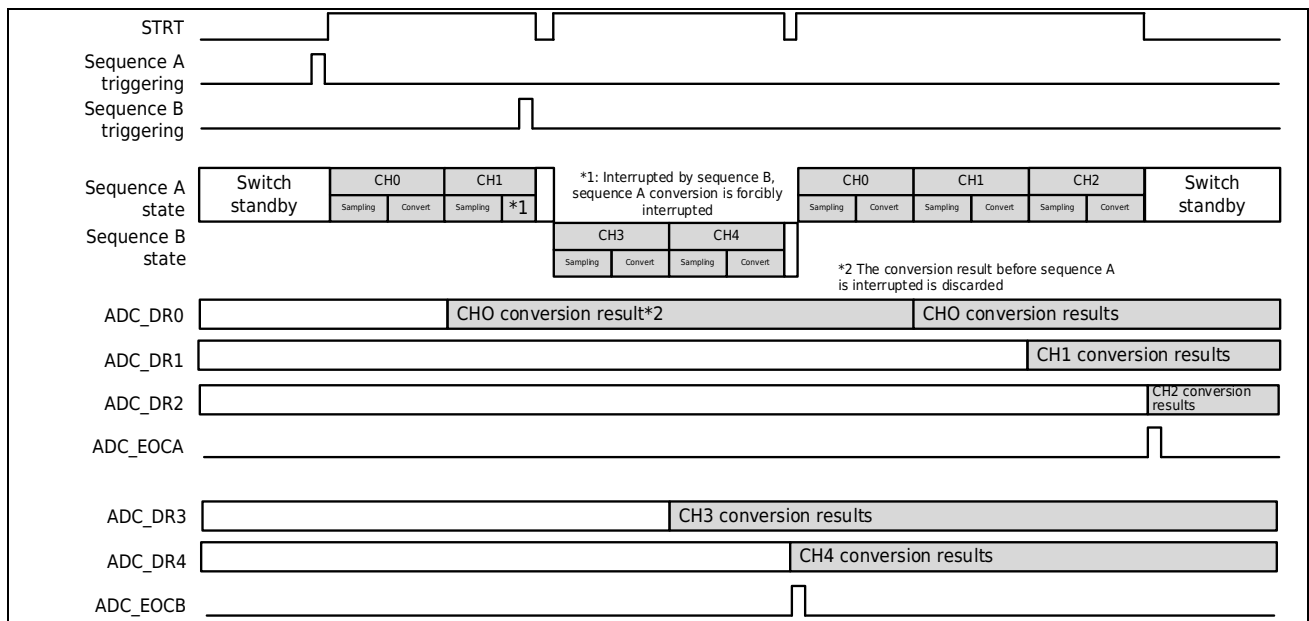


Figure 17-6 Double Sequence Can Mode (Sequence A Restarts from First Channel)

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 [1:0] is set to 10b or 11b 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 register ADC_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.

17.3.7 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 instance, 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.

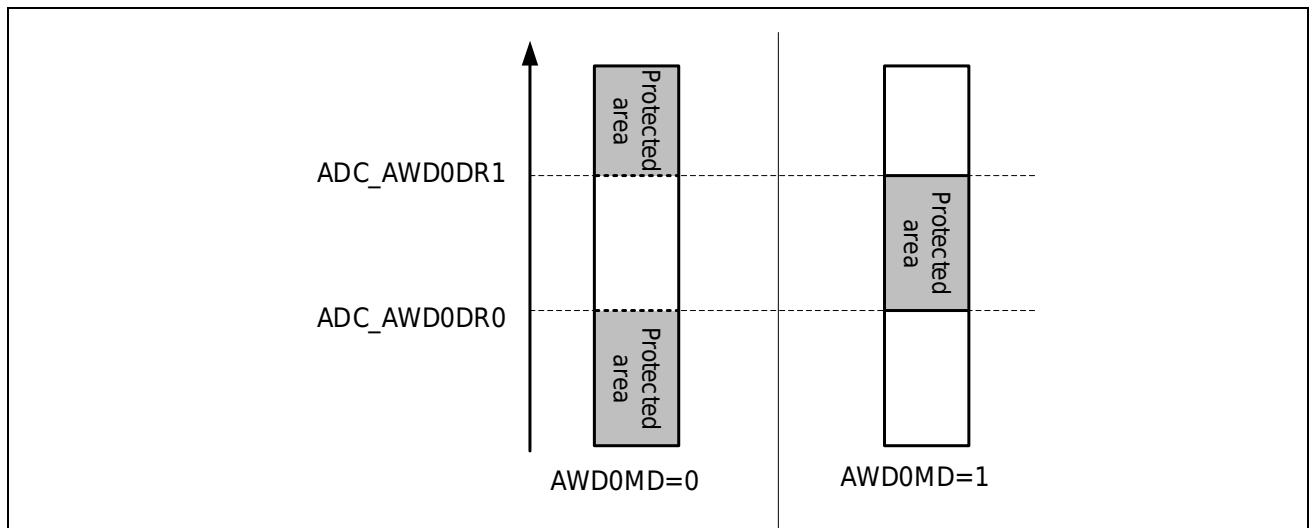


Figure 17-7 Analog Watchdog Protection Area (Compare Mode)

The software flow using the analog watchdog function is illustrated by comparison window 0:

1. Set the threshold register AD_AWD0DR0, ADC_AWD0DR1;
2. Set the comparison channel register AD _ 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 register ADC _ 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.

17.3.8 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 periods 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 time t_{SPL} can be set independently, the conversion time t_{CONV} of each channel can be different.

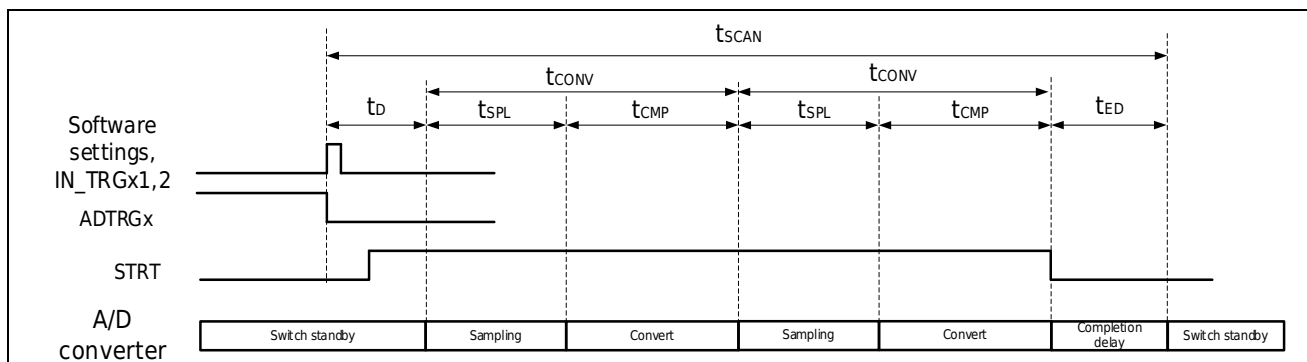


Figure 17-8 ADC Conversion Time

Table 17-3 AD Conversion Time

Flag		Description		Conditions			
				Synchronous Peripheral Trigger	Asynchronous Peripheral Trigger *Note	External Pin Trigger	Software Triggering
tD		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 conversion, starting sequence A starting sequence B conversion	5 PCLK2	4 PCLK4 + 5 PCLK2 + 1 PCLK4_SYNC	2 PCLK4 + 5 PCLK2	-
tCONV	tSPL	Sampling time		ADSSTRx.SST [7:0] x PCLK2			
	tCMP	Successful conversion time	12-bit resolution	13 PCLK2			
			10-bit resolution	11 PCLK2			
			8-bit resolution	9 PCLK2			
tED		Scan completion time		3 PCLK2			
tTD		Minimum continuous trigger interval		ΣtCONV + 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.

17.3.9 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 register ADC_DR is updated. Instances are provided below.

- When ADC_CR0.CLREN is "0" and the automatic clearing function is disabled, 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 automatic 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 in the interrupt processing, it will be easy to determine whether the ADC conversion data is stored correctly.

17.3.10 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 averaged conversion results into the data register. 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.

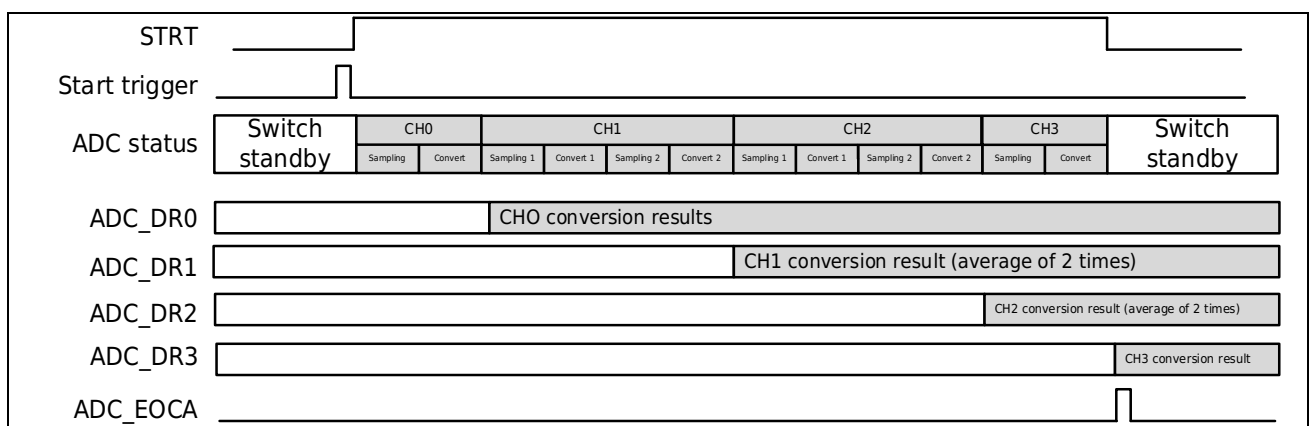


Figure 17-9 Conversion Action When Averaging Function Valid

In the Figure 17-9, 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.

17.3.11 Programmable Gain Amplifier PGA

This MCU is equipped with a programmable gain amplifier PGA. You can set the register ADC_PGACR to enable the PGA circuit and select the gain multiple. The gain range is x2~x32. At this time, the analog input is first amplified by the PGA circuit, and then input to the ADC module for conversion. using PGA, please set register PWC_FCG3.CMBIAS to enable PGA and wait for 2us startup stabilization time.

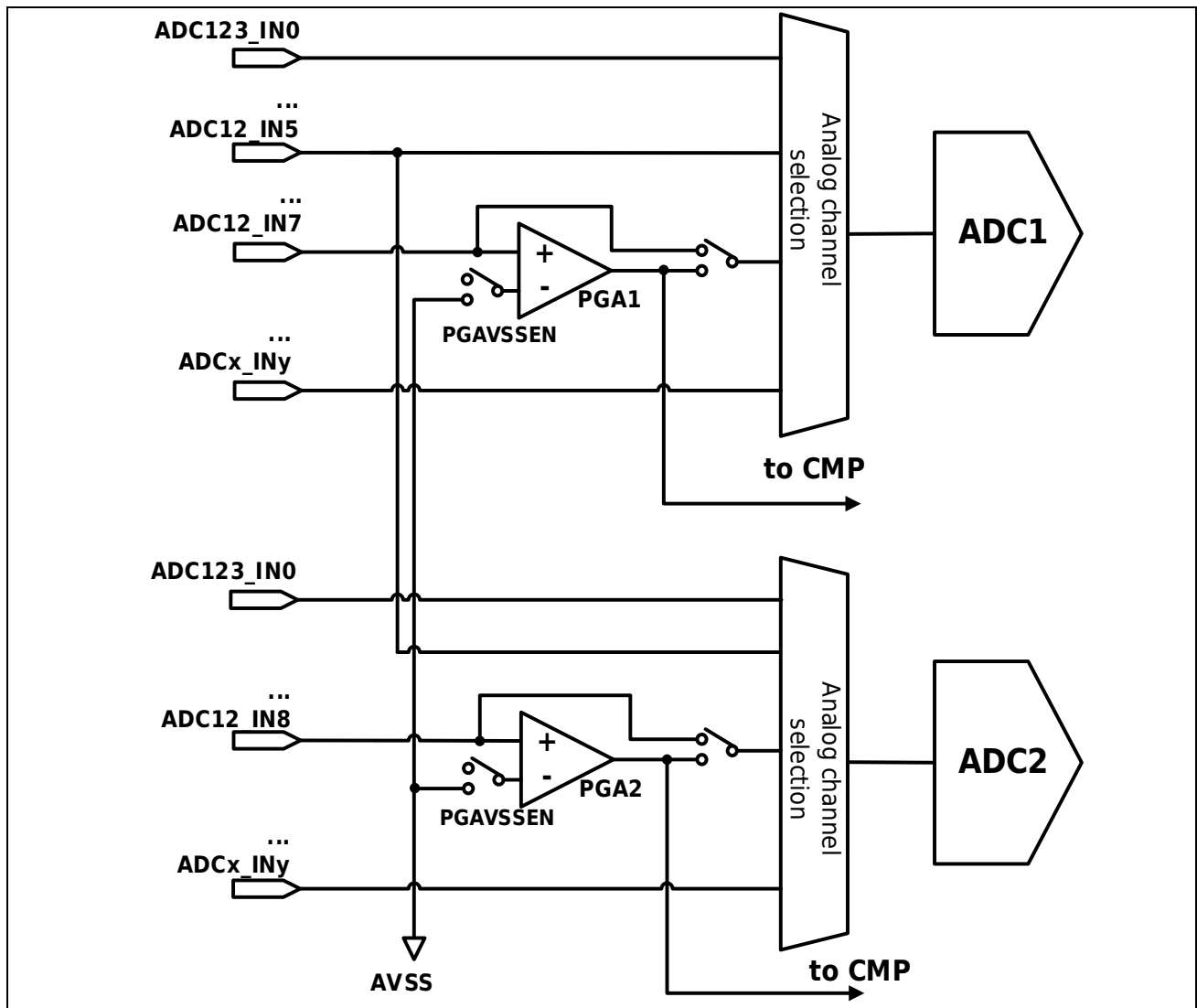


Figure 17-10 AD And PGA Channel Schematic

As shown in the figure above, ADC1 is connected to PGA1, which corresponds to the physical channel ADC12_IN7. ADC2 is connected to PGA2, corresponding to the physical channel ADC123_IN8. ADC3 does not support PGA. are 2 PGA channels in total.

17.3.12 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.

You can set the two ADC modules ADC1 and ADC2 to work together, or you can set the three ADC modules ADC1, ADC2 and ADC3 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
- Cycle delay trigger mode

Single 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.

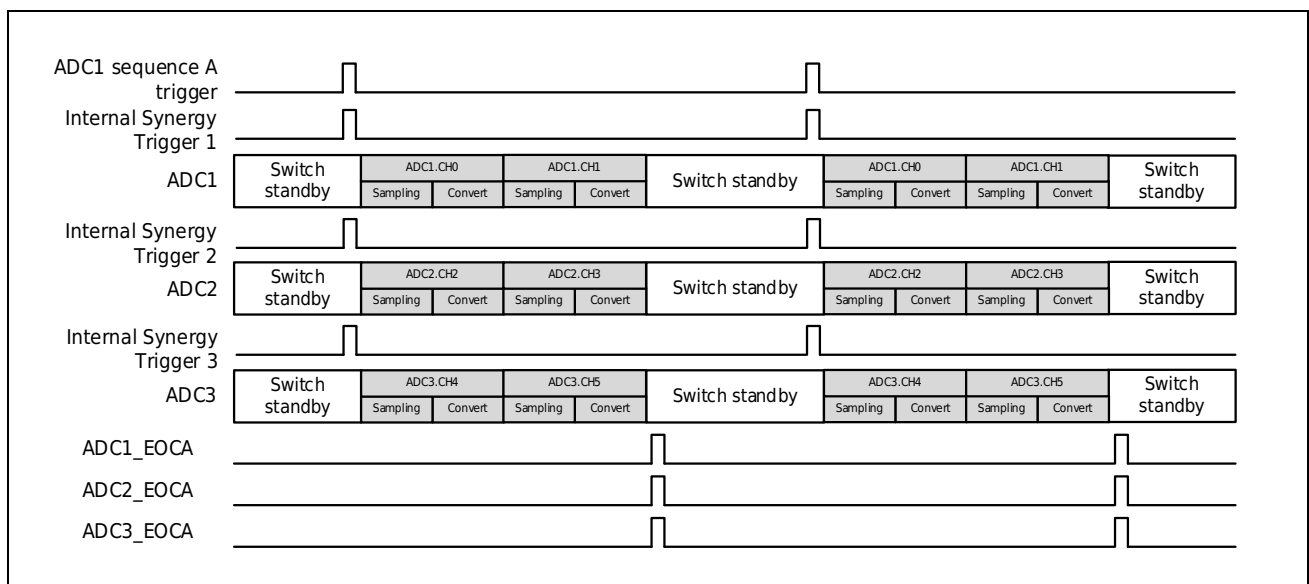


Figure 17-11 Single Parallel Trigger Mode (Three ADCs)

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_SYNCRR.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 [1:0] to 00b: sequence A single scan mode, or 01b: 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 [1:0], 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 [1:0], 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_SYNCRR.SYNCMD [2:0], write 010b: ADC1 and ADC2 work together. Or write 011b: ADC1, ADC2, ADC3 three ADCs work together
 6. Write 1 to the collaborative work permission register ADC_SYNCRR.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.

Single Delay 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.

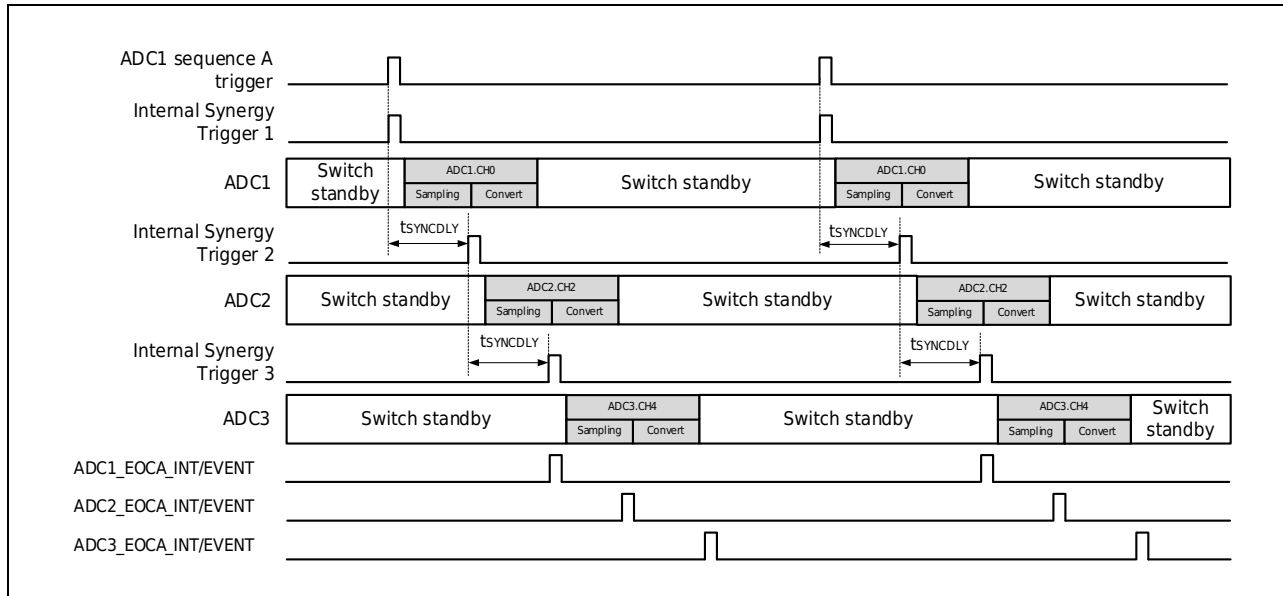


Figure 17-12 Single Delay Trigger Mode (Three ADCs)

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_SYNCCCR.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_SYNCCCR.SYNCDLY` [7:0] to set the startup delay of the two ADCs.
4. Set the cooperative mode control register `ADC_SYNCCCR.SYNCMD` [2:0], write 000b: ADC1 and ADC2 work together. Or write 001b: ADC1, ADC2, ADC3 and three ADCs work together.
5. Write 1 to the collaborative work permission register `ADC_SYNCCCR.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 and 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.

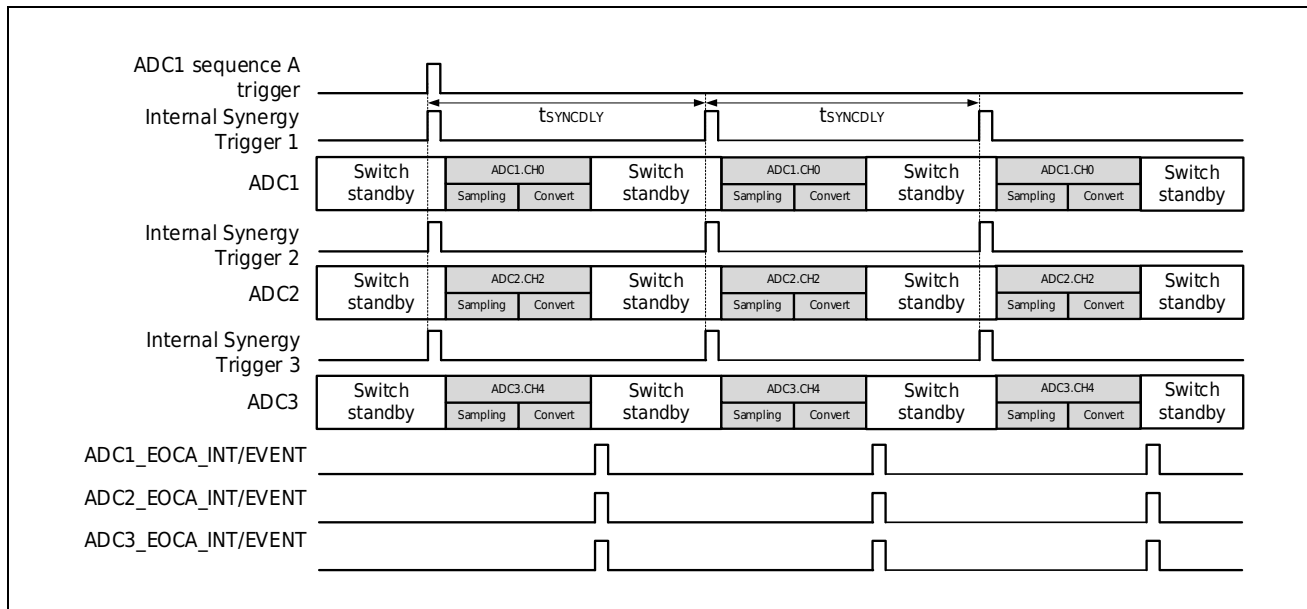


Figure 17-13 Cyclic Parallel Trigger Mode (Three ADCs)

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_SYNCRR.SYNCEN` to confirm that the collaborative work is invalid.
2. Set up ADC1, 2, 3 modules, refer to single parallel mode. `ADC_CR0.MS [1:0]` set to 00b: Sequence A single scan mode.
3. Set the cooperative mode control register `ADC_SYNCRR.SYNCDLY [7:0]` to set the delay of each parallel trigger.
4. Set the cooperative mode control register `ADC_SYNCRR.SYNCMD [2:0]`, write 110b: ADC1 and ADC2 work together. Or write 111b : The three ADCs ADC1 , ADC2 and ADC3 work together.
5. Write 1 to the collaborative work permission register `ADC_SYNCRR.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 the sequence A trigger condition of ADC1 triggers ADC1, after each set delay, it will continuously trigger ADC2, ADC3, ADC1, ADC2... until the user actively software stops the ADC1 module or disables the cooperative working mode.

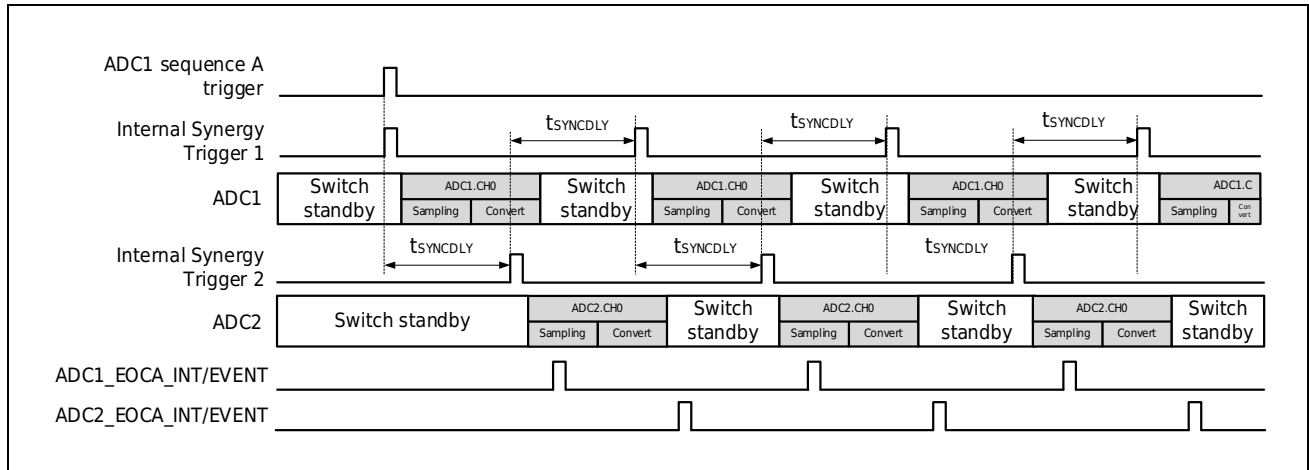


Figure 17-14 Cyclic Delay Trigger Mode (Two ADCs)

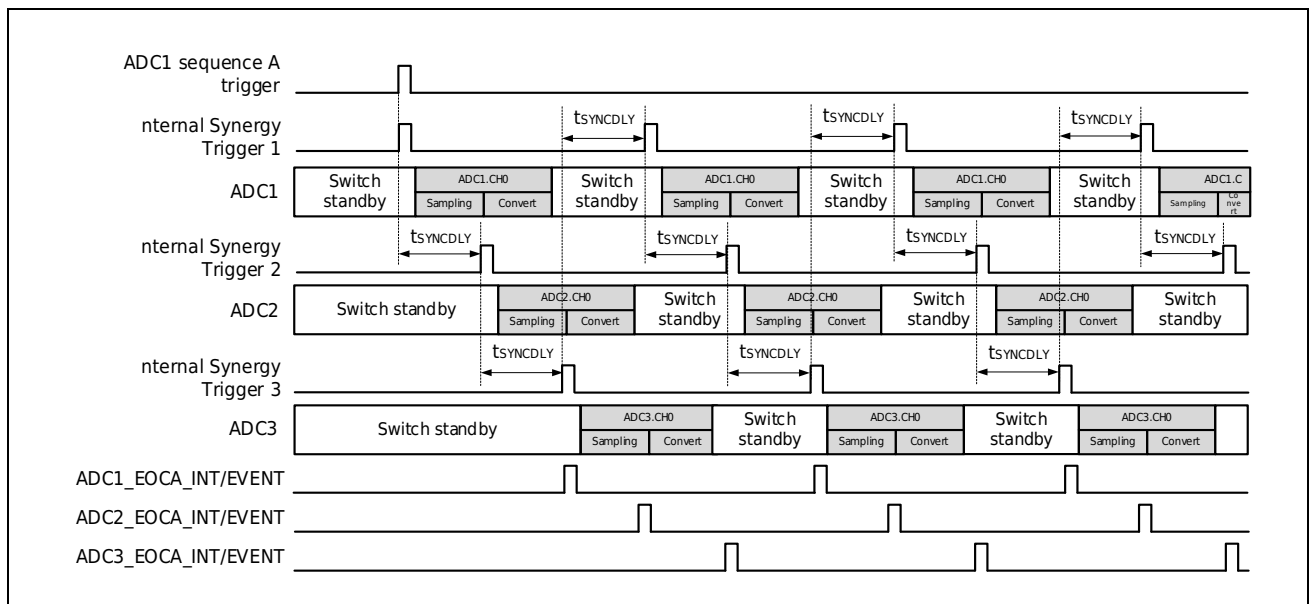


Figure 17-15 Cyclic Delay Trigger Mode (Three ADCs)

Note:

- When the two ADCs work together, the delay time $t_{SYNCDLY}$ and the time t_{SCAN} of one 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_SYNCRR.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 cooperative mode control register ADC_SYNCRR.SYNCDLY [7:0] to set the delay of each trigger.
4. Set the cooperative mode control register ADC_SYNCRR.SYNCMD [2:0], write 100b: ADC1 and ADC2 work together. Or write 101b: ADC1, ADC2, ADC3 and three ADCs work together.
5. Write 1 to the collaborative work permission register ADC_SYNCRR.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.

17.3.13 Interrupt and Event Signal Output

The ADC module can produce the following four types of event outputs. When each event occurs, if the corresponding interrupt disable register is set to valid, the interrupt application is also output.

1. Sequence A Scan End ADC _ EOCA, corresponding to interrupt enable register ADC _ ICR.EOCAIEN
2. Sequence B Scan End ADC _ EOCB, corresponding to interrupt enable register ADC _ 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, refer to the DMA description 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.

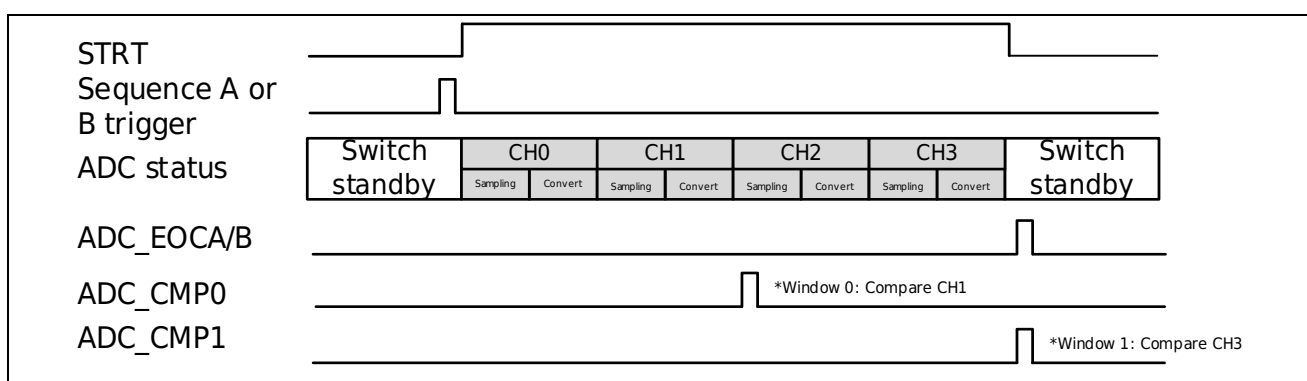


Figure 17-16 ADC interrupt and Event Output Timing

17.4 Register Description

17.4.1 Register List

Table 17-4 ADC Register List

Unit 1 BASE_ADDR: 0x40040000

Unit 2 BASE_ADDR: 0x40040400

Unit 3 BASE_ADDR: 0x40040800

Register Name	Symbol	Offset Address	Bit Width	Reset Value
ADC start register	ADC_STR	0x00	8	0x00
ADC Control Register0	ADC_CR0	0x02	16	0x0000
ADC Control Register1	ADC_CR1	0x04	16	0x0000
ADC Conversion Start Trigger Register	ADC_TRGSR	0x0a	16	0x0000
ADC Channel Selection Register A	ADC_CHSELRA	0x0c	32	0x00000000
ADC Channel Selection Register B	ADC_CHSELRB	0x10	32	0x00000000
ADC Mean Channel Selection Register	ADC_AVCHSELR	0x14	32	0x00000000
ADC Sampling Period Register	ADC_SSTRx [note]	0x20+x	8	0x0b
	ADC_SSTRL	0x30	8	0x0b
ADC Channel Mapping Control Register 0	ADC_CHMUXR0	0x38	16	0x3210
ADC Channel Mapping Control Register 1	ADC_CHMUXR1	0x3a	16	0x7654
ADC Channel Mapping Control Register 2	ADC_CHMUXR2	0x3c	16	0xba98
ADC Channel Mapping Control Register 3	ADC_CHMUXR3	0x3e	16	0xfedc
ADC Interruption Status Register	ADC_ISR	0x44	8	0x00
ADC interrupt enable register	ADC_ICR	0x45	8	0x03
ADC Interrupt Status Reset Register	ADC_ISCLRR	0x46	8	0x00
ADC Cooperative Mode Control Register	ADC_SYNCCR	0x4c	16	0x0c00
ADC Data Register	ADC_DRy [note]	0x50+2*y	16	0x0000
Analog watchdog control register	ADC_AWDCR	0xa0	16	0x0000
Analog watchdog status register	ADC_AWDSR	0xa2	8	0x00
Analog watchdog status reset register	ADC_AWDSCLRR	0xa3	8	0x00
Analog watchdog window 0 threshold register	ADC_AWD0DR0	0xa4	16	0x0000
	ADC_AWD0DR1	0xa6	16	0xffff
Analog watchdog window 0 comparison channel selection register	ADC_AWD0CHSR	0xa8	8	0x00
Analog watchdog window 1 threshold register	ADC_AWD1DR0	0xac	16	0x0000
	ADC_AWD1DR1	0xae	16	0xffff
Analog watchdog window 1 comparison channel selection register	ADC_AWD1CHSR	0xb0	8	0x00
Programmable Gain Amplifier Control Register	ADC_PGACR	0xc0	8	0x00
Programmable gain amplifier reference	ADC_PGAVSSENr	0xc4	8	0x00

Register Name	Symbol	Offset Address	Bit Width	Reset Value
ground select register				

Note:

The "x" in ADC_SSTRx represents the channel, and x=0~15 in ADC1/ADC2/ADC3.

The "y" in ADC_DRy represents the channel, y=0~21 in ADC1 , y=0~21 in ADC2 , and y=0~22 in ADC3.

17.4.2 ADC Start Register (ADC_STR)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	STRT

Bit	Flag	Bit Name	Function	R/W
b7-b1	Reserved	-	Read at 0, write at 0	R/W
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.	R/W

17.4.3 ADC Control Register 0 (ADC_CR0)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8
-	-	-	-	-	AVCNT [2:0]		
b7	b6	b5	b4	b3	b2	b1	b0
DFMT	CLREN	ACCSEL [1:0]		-	-	MS [1:0]	

Bit	Flag	Bit Name	Function	R/W
b15-b12	Reserved	-	Read at 0, write at 0	R/W
b10-b8	AVCNT [2:0]	Times selection	0 0 0: Average of 2 consecutive conversions 0 0 1: Average of 1 consecutive conversion 0 1 0: Average of 8 consecutive conversions 0 1 1: Average of 16 consecutive conversions 1 0 0: Average of 32 consecutive conversions 1 0 1: Average of 64 consecutive conversions 1 1 0: Average of 128 consecutive conversions 1 1 1: Average of 256 consecutive conversions	R/W
b7	DFMT	Data format	0: Right alignment of converted data 1: Left alignment of converted data	R/W
b6	CLREN	Data register automatic purge	0: Automatic removal of prohibitions 1: Automatic Reset Note: After the CLREN bit is set, register ADC_DRx is automatically cleared after CPU, DMA, etc. The automatic clearing function is mainly used to detect whether the data register is updated.	R/W
b5-b4	ACCSEL [1:0]	Resolution selection	0 0: 12-bit resolution 0 1: 10 bit resolution 1 0: 8 bit resolution 1 1: Prohibitions	R/W
b3-b2	Reserved	-	Read at 0, write at 0	R/W
b1-b0	MS [1:0]	Pattern selection	0 0: Sequence A single scan mode, sequence B invalid 0 1: Sequence A continuous scan mode, sequence B invalid 1 0: Sequence A single scan mode, Sequence B single scan mode 1 1: Sequence A continuous scan mode, sequence B single scan mode	R/W

Note:

- Set this register when ADC_STR. STRT is "0".

17.4.4 ADC Control Register 1 (ADC_CR1)

Reset value: 0x0000

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

Bit	Flag	Bit Name	Function	R/W
b15-b3	Reserved	-	Read at 0, write at 0	R/W
b2	RSCHSEL	Sequence A Restart Channel Selection	0: After sequence B interrupt, sequence A continues to scan from the interrupt channel when it restarts 1: Rescan from the first channel when sequence A restarts after sequence B interrupt	R/W
b1-b0	Reserved	-	Read at 0, write at 0	R/W

Note:

- Set this register when ADC_STR. STRT is "0".

17.4.5 ADC Conversion Start Trigger Register (ADC_TRGSR)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8
TRGENB	-	-	-	-	-	TRGSELB [1]	TRGSELB [0]
b7	b6	b5	b4	b3	b2	b1	b0
TRGENA	-	-	-	-	-	TRGSELA [1]	TRGSELA [0]

Bit	Flag	Bit Name	Function	R/W
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 is valid. If ADTRGx is changed from "High" to "Low" and a falling edge is detected, the scan conversion begins. Keep "Low" 1.5 * PCLK4 periods above.	R/W
b14-b10	Reserved	-	Read at 0, write at 0	R/W
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 00b: ADTRGx 01b: IN_TRGx0 10b: IN_TRGx1 11b: IN_TRGx0 + IN_TRGx1 Note: Only valid in sequence B active 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.	R/W
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 is valid. If ADTRGx is changed from "High" to "Low" and a falling edge is detected, the scan conversion begins. Keep "Low" 1.5 * PCLK4 periods above.	R/W
b6-b2	Reserved	-	Read at 0, write at 0	R/W
b1-b0	TRGSELA [1:0]	Sequence A trigger condition selection	Trigger condition for sequence A. 00b: ADTRGx (x=1~3, represents the ADC unit number) 01b: IN_TRGx0 10b: IN_TRGx1 11b: 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.	R/W

Note:

- Set this register when ADC_STR.STRT is "0".

17.4.6 ADC Channel Select Register A (ADC_CHSELRA)

Reset value: 0x00000000

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	Flag	Bit Name	Function	R/W
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 the non-existent channel are reserved bits, 0 at read and 0 at write. Note: Do not select the same channel in sequence A and sequence B.	R/W

Note:

- Set this register when ADC_STR. STRT is "0".

17.4.7 ADC Channel Select Register B (ADC_CHSELRB)

Reset value: 0x00000000

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	Flag	Bit Name	Function	R/W
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 the non-existent channel are reserved bits, 0 at read and 0 at write. Note: Do not select the same channel in sequence A and sequence B.	R/W

Note:

- Set this register when ADC_STR. STRT is "0".

17.4.8 ADC Average Channel Select Register (ADC_AVCHSELR)

Reset value: 0x00000000

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	Flag	Bit Name	Function	R/W
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 the non-existent channel are reserved bits, 0 at read and 0 at write. 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.	R/W

Note:

- Set this register when ADC_STR. STRT is "0".

17.4.9 ADC Sampling Period Register ADC_SSTRx, x=0~15/ADC_SSTRL

Reset value: 0x0B

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

Bit	Flag	Bit Name	Function	R/W
b7-b0	SST [7:0]	Number of sampling periods	The number of sampling periods can be set from 5 to 255 periods. Channels CH0~15 are set by ADC_SSTRx, x=0~15, and other channels are set by ADC_SSTRL. 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 periods. $SST \geq (R_{AIN} + R_{ADC}) * C_{ADC} * \ln(2^{N+2}) * f_{ADC} + 1$ Where: R_{AIN} represents external input impedance (Ω), R_{ADC} represents internal sampling switch resistance (Ω), C_{ADC} represents internal sampling and retention capacitance (F), N represents AD resolution (12/10/8), and f_{ADC} represents PCLK2 frequency (Hz). Refer to the description of electrical characteristics.	R/W

Note:

- Set this register when ADC_STR. STRT is "0".
- The sampling time of the internal reference voltage should not be less than 1us.

17.4.10 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]			

Bit	Flag	Bit Name	Function	R/W																																																																				
			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>ADC123 IN0</td><td>ADC123 IN0</td><td>ADC123 IN0</td></tr><tr><td>0x1</td><td>ADC123 IN1</td><td>ADC123 IN1</td><td>ADC123 IN1</td></tr><tr><td>0x2</td><td>ADC123 IN2</td><td>ADC123 IN2</td><td>ADC123 IN2</td></tr><tr><td>0x3</td><td>ADC123 IN3</td><td>ADC123 IN3</td><td>ADC123 IN3</td></tr><tr><td>0x4</td><td>ADC12 IN4</td><td>ADC12 IN4</td><td>ADC3 IN4</td></tr><tr><td>0x5</td><td>ADC12 IN5</td><td>ADC12 IN5</td><td>ADC3 IN5</td></tr><tr><td>0x6</td><td>ADC12 IN6</td><td>ADC12 IN6</td><td>ADC3 IN6</td></tr><tr><td>0x7</td><td>ADC12 IN7</td><td>ADC12 IN7</td><td>ADC3 IN7</td></tr><tr><td>0x8</td><td>ADC12 IN8</td><td>ADC12 IN8</td><td>ADC3 IN8</td></tr><tr><td>0x9</td><td>ADC12 IN9</td><td>ADC12 IN9</td><td>ADC3 IN9</td></tr><tr><td>0xa</td><td>ADC123 IN10</td><td>ADC123 IN10</td><td>ADC123 IN10</td></tr><tr><td>0xb</td><td>ADC123 IN11</td><td>ADC123 IN11</td><td>ADC123 IN11</td></tr><tr><td>0xc</td><td>ADC123 IN12</td><td>ADC123 IN12</td><td>ADC123 IN12</td></tr><tr><td>0xd</td><td>ADC123 IN13</td><td>ADC123 IN13</td><td>ADC123 IN13</td></tr><tr><td>0xe</td><td>ADC12 IN14</td><td>ADC12 IN14</td><td>ADC3 IN14</td></tr><tr><td>0xf</td><td>ADC12 IN15</td><td>ADC12 IN15</td><td>ADC3 IN15</td></tr></table>	Set value	ADC1 mapping object	ADC2 mapping object	ADC3 mapping object	0x0	ADC123 IN0	ADC123 IN0	ADC123 IN0	0x1	ADC123 IN1	ADC123 IN1	ADC123 IN1	0x2	ADC123 IN2	ADC123 IN2	ADC123 IN2	0x3	ADC123 IN3	ADC123 IN3	ADC123 IN3	0x4	ADC12 IN4	ADC12 IN4	ADC3 IN4	0x5	ADC12 IN5	ADC12 IN5	ADC3 IN5	0x6	ADC12 IN6	ADC12 IN6	ADC3 IN6	0x7	ADC12 IN7	ADC12 IN7	ADC3 IN7	0x8	ADC12 IN8	ADC12 IN8	ADC3 IN8	0x9	ADC12 IN9	ADC12 IN9	ADC3 IN9	0xa	ADC123 IN10	ADC123 IN10	ADC123 IN10	0xb	ADC123 IN11	ADC123 IN11	ADC123 IN11	0xc	ADC123 IN12	ADC123 IN12	ADC123 IN12	0xd	ADC123 IN13	ADC123 IN13	ADC123 IN13	0xe	ADC12 IN14	ADC12 IN14	ADC3 IN14	0xf	ADC12 IN15	ADC12 IN15	ADC3 IN15	
Set value	ADC1 mapping object	ADC2 mapping object	ADC3 mapping object																																																																					
0x0	ADC123 IN0	ADC123 IN0	ADC123 IN0																																																																					
0x1	ADC123 IN1	ADC123 IN1	ADC123 IN1																																																																					
0x2	ADC123 IN2	ADC123 IN2	ADC123 IN2																																																																					
0x3	ADC123 IN3	ADC123 IN3	ADC123 IN3																																																																					
0x4	ADC12 IN4	ADC12 IN4	ADC3 IN4																																																																					
0x5	ADC12 IN5	ADC12 IN5	ADC3 IN5																																																																					
0x6	ADC12 IN6	ADC12 IN6	ADC3 IN6																																																																					
0x7	ADC12 IN7	ADC12 IN7	ADC3 IN7																																																																					
0x8	ADC12 IN8	ADC12 IN8	ADC3 IN8																																																																					
0x9	ADC12 IN9	ADC12 IN9	ADC3 IN9																																																																					
0xa	ADC123 IN10	ADC123 IN10	ADC123 IN10																																																																					
0xb	ADC123 IN11	ADC123 IN11	ADC123 IN11																																																																					
0xc	ADC123 IN12	ADC123 IN12	ADC123 IN12																																																																					
0xd	ADC123 IN13	ADC123 IN13	ADC123 IN13																																																																					
0xe	ADC12 IN14	ADC12 IN14	ADC3 IN14																																																																					
0xf	ADC12 IN15	ADC12 IN15	ADC3 IN15																																																																					
-	CHxMUX [3:0]	Channel x mapping selection x=0-15		R/W																																																																				
Note: Please do not set it to an analog input that does not exist.																																																																								

Note:

- Set this register when ADC_STR.STRT is "0".

17.4.11 ADC Interrupt Status Register (ADC_ISR)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	SASTPDF	-	-	EOCBF	EOCAF

Bit	Flag	Bit Name	Function	R/W
b7-b5	Reserved	-	Read at 0, write at 0	R/W
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	Reserved	-	Read at 0, write at 0	R/W
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

17.4.12 ADC Interrupt Enable Register (ADC_ICR)

Reset value: 0x03

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	EOCBIEN	EOCAIEN

Bit	Flag	Bit Name	Function	R/W
b7-b2	Reserved	-	Read at 0, write at 0	R/W
b1	EOCBIEN	Sequence B conversion complete interrupt enable	0: Sequence B conversion complete interrupt disable 1: Sequence B conversion complete interrupt enable	R/W
b0	EOCAIEN	Sequence A conversion complete interrupt enable	0: Sequence A conversion complete interrupt disable 1: Sequence A conversion complete interrupt enable	R/W

17.4.13 ADC Interrupt Status Reset Register (ADC_ISCLRR)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	CLRSASTPDF	-	-	CLREOCBF	CLREOCAF

Bit	Flag	Bit Name	Function	R/W
b7-b5	Reserved	-	Read at 0, write at 0	R/W
b4	CLRSASTPDF	Sequence A interrupted flag reset	Writing 0 has no effect, writing 1 resets the SASTPDF status bit. Read always 0	R/W
b3-b2	Reserved	-	Read at 0, write at 0	R/W
b1	CLREOCBF	Sequence B conversion complete flag reset	Write 0 has no effect. Write 1 resets the EOCBF status bit. Read always 0	R/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.	R/W

17.4.14 ADC Synergy Mode Control Register (ADC_SYNCRR)

Reset value: 0x0c00

b15	b14	b13	b12	b11	b10	b9	b8
SYNCDLY [7:0]							
b7	b6	b5	b4	b3	b2	b1	b0
-	SYNCMD [2]	SYNCMD [1]	SYNCMD [0]	-	-	-	SYNCEN

Bit	Flag	Bit Name	Function	R/W
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 $t_{\text{SYNCDLY}} = 1 * \text{PCLK2}$, 0xff means $t_{\text{SYNCDLY}} = 255 * \text{PCLK2}$ Note: Set this register 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: Single delayed trigger mode: $t_{\text{SYNCDLY}} > t_{\text{SPL}}$ Two ADC period delay trigger modes: $t_{\text{SYNCDLY}} > t_{\text{SPL}}$, and $t_{\text{SYNCDLY}} > t_{\text{SCAN}}/2$ Three ADC period delay trigger modes: $t_{\text{SYNCDLY}} > t_{\text{SPL}}$, and $t_{\text{SYNCDLY}} > t_{\text{SCAN}}/3$ Single parallel trigger mode: the setting of this register is invalid. Loop parallel trigger mode: $t_{\text{SYNCDLY}} > t_{\text{SCAN}}$	R/W
b7	Reserved	-	Read at 0, write at 0	R/W
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: Set this register 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.	R/W
b3-b1	Reserved	-	Read at 0, write at 0	R/W
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.	R/W

Note:

- This register is only available in the master ADC (i.e. ADC1), and is not available in other ADC units.

17.4.15 ADC data register ADC_Dry, ADC1 y=0~21, ADC2 y=0~21, ADC3 y=0~22

The ADC_DR register is a read-only register used to store the ADC conversion data of each channel.
The reset value is 0x0000

Conversion results are stored differently based on data alignment and conversion resolution.

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

17.4.16 Analog watchdog control register (ADC _AWDCR)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8
-	-	-	-	-	-	AWDCM [1:0]	
b7	b6	b5	b4	b3	b2	b1	b0
-	AWD1MD	AWD1IEN	AWD1EN	-	AWD0MD	AWD0IEN	AWD0EN

Bit	Flag	Bit Name	Function	R/W
b15-b10	Reserved	-	Read at 0, write at 0	R/W
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 _CMP1 interrupt or event is output at the end of the channel conversion selected for Window 1.	R/W
b7	Reserved	-	Read at 0, write at 0	R/W
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.	R/W
b5	AWD1IEN	Watchdog window 1 interrupt enable	0: Watchdog Window 1 comparison interruptADC _CMP1 invalid 1: Watchdog Windows1 Compare the validity of interruptADC _CMP1	R/W
b4	AWD1EN	Watchdog window 1 compare enable	0: Watchdog Window 1 comparison function invalid 1: Watchdog Window 1 invalid	R/W
b3	Reserved	-	Read at 0, write at 0	R/W
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.	R/W
b1	AWD0IEN	Watchdog window 0 Interrupt enable	0: Watchdog Window 0 Compare interruptADC _CMP0 Invalid 1: Watchdog Windows0 Compared with interruptADC _CMP0	R/W
b0	AWD0EN	Watchdog window 0 comparing function enable	0: Watchdog Window 0 comparison function invalid 1: Watchdog Window 0 Compare Function Effective	R/W

17.4.17 Analog Watchdog Status Register (AD _ AWDSR)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	AWDCMF	-	-	AWD1F	AWD0F

Bit	Flag	Bit Name	Function	R/W
b7-b5	Reserved	-	Read at 0, write at 0	R/W
b4	AWDCMF	Watchdog window combination comparison status bit	When the watchdog window combination comparison function is effective (that is, AWDCM [1: 0] = 01b/10b/11b), when window 0 and window 1 compare results meet the combination condition, set 1. Invalid write to this register bit.	R
b3-b2	Reserved	-	Read at 0, write at 0	R/W
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

17.4.18 Analog Watchdog Status Reset Register (ADC _ AWDSCLR)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	CLRAWDCMF	-	-	CLRAWD1F	CLRAWD0F

Bit	Flag	Bit Name	Function	R/W
b7-b5	Reserved	-	Read at 0, write at 0	R/W
b4	CLRAWDCMF	Watchdog window combination compare status bit reset	Write 0 has no effect. Write 1 resets the AWDCMF status bit. Read always 0	R/W
b3-b2	Reserved	-	Read at 0, write at 0	R/W
b1	CLRAWD1F	Watchdog window 1 comparative state reset	Write 0 has no effect. Write 1 resets the AWD1F status bit. Read always 0	R/W
b0	CLRAWD0F	Watchdog window 0 comparative state reset	Write 0 has no effect. Write 1 resets the AWD1F status bit. Read always 0	R/W

17.4.19 Analogwatchdog Thresholds Register (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	Flag	Bit Name	Function	R/W
b15-b0	AWDDR [15:0]	Comparative data	Comparative data	R/W

AWD0DR0 Setup Window 0 Low Valve, AWD0DR1 Setup Window 0 High Valve.

AWD1DR0 Setup Window 1 Low Valve, AWD1DR1 Setup Window 1 High Valve.

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]

17.4.20 Analog watchdog comparison channel selection register (ADC _ AWD0 CHSR, ADC _ AWD1 CHSR)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	AWDCH [4]	AWDCH [3]	AWDCH [2]	AWDCH [1]	AWDCH [0]

Bit	Flag	Bit Name	Function	R/W
b7-b5	Reserved	-	Read at 0, write at 0	R/W
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	R/W

17.4.21 Programmable Gain Amplifier Control Register ADC_PGACR

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
PGAGAIN [3:0]				PGACTL [3:0]			
Bit	Flag	Bit Name	Function	R/W			
b7-b4	PGAGAIN [3:0]	Gain setting	0 0 0 0: × 2.000	-			
			0 0 0 1: × 2.133				
			0 0 1 0: × 2.286				
			0 0 1 1: × 2.667				
			0 1 0 0: × 2.909				
			0 1 0 1: × 3.2				
			0 1 1 0: × 3.556				
			0 1 1 1: × 4.000				
			1 0 0 0: × 4.571				
			1 0 0 1: × 5.333				
			1 0 1 0: × 6.4				
			1 0 1 1: × 8				
			1 1 0 0: × 10.667				
			1 1 0 1: × 16				
			1 1 1 0: × 32				
			Note: Other values are prohibited to be set.				
b3-b0	PGACTL [3:0]	Amplifier control	0000: Amplifier invalid	R/W			
			1110 : The amplifier is enabled and the signal is amplified according to the setting value of PGAGAIN [3:0].				
			Note: Setting values other than the above is prohibited.				

Note:

- Unit 1 (ADC1) supports 1 PGA channel, refer to Table 17-1. Register ADC1_PGACR corresponds to the control of PGA1.
- Unit 2 (ADC2) supports 1 PGA channel, refer to Table 17-1. Register ADC2_PGACR corresponds to the control of PGA2.
- Unit 3 (ADC3) does not support PGA and does not have this register.

17.4.22 Programmable gain amplifier reference ground select register ADC_PGAVSSEN

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	PGAVSSEN

Bit	Flag	Bit Name	Function	R/W
b7-b1	Reserved	-	Read at 0, write at 0	R/W
b0	PGAVSSEN	PGA amplifier reference ground enable	0: Disable analog ground AVSS as PGA reference ground 1 : Use analog ground AVSS as the PGA reference ground	R/W

Note:

- Unit 1 (ADC1) supports 1 PGA channel, refer to Table 17-1. Register ADC1_PGACR corresponds to the control of PGA1.
- Unit 2 (ADC2) supports 1 PGA channel, refer to Table 17-1. Register ADC2_PGACR corresponds to the control of PGA2.
- Unit 3 (ADC3) does not support PGA and does not have this register.

17.5 Precautions

17.5.1 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.

17.5.2 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.

17.5.3 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 1us 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.

For details, please refer to the Low Power Description section.

17.5.4 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 IO (GPIO)] chapter.

17.5.5 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.

18 Digital to Analog Converter (DAC)

18.1 Introduction

HC32A472 is equipped with a digital -to-analog converter DAC1. Each DAC contains two 12 -bit DAC conversion channels, which can be converted independently or synchronously.

The main features of the DAC are as follows:

- 2 independent 12 -bit DAC conversion channels
- The converted data supports left-aligned or right-aligned mode
- The two conversion channels of the same DAC can be converted synchronously
- Independent reference voltage VREFH input can improve conversion accuracy
- The output can be used as the negative input of the voltage comparator (CMP)
- ADC conversion priority mode can reduce interference during ADC conversion
- DAC output remains on when software reset / watchdog reset occurs

18.2 Functional Block Diagram

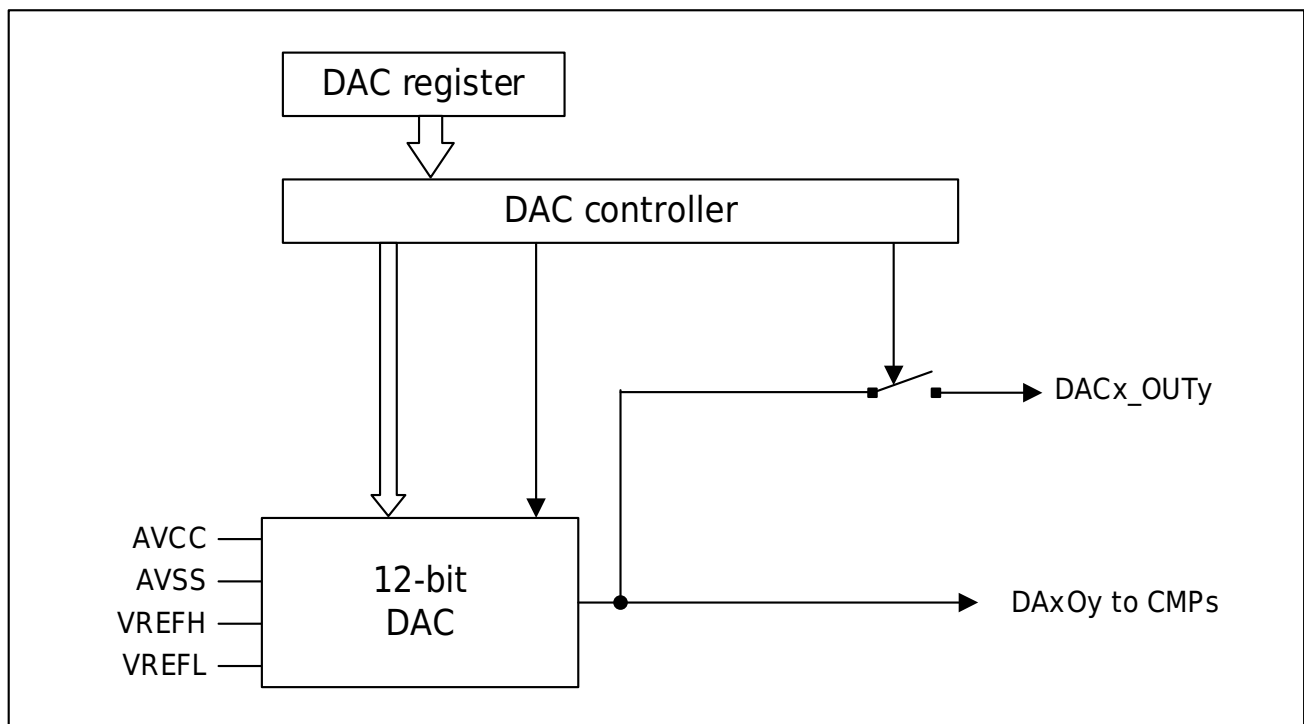


Figure 18-1 DAC Conversion Channel Block Diagram

Note: *x* represents the DAC number, and its value range is 1; *y* represents the DAC conversion channel number, and its value range is 1~2. The same applies to the following.

Table 18-1 DAC Input and Output Pins

Pin Name	Input/Output	Function
AVCC	Input	Analog Power
AVSS	Input	Analog Ground
VREFH ^{note1}	Input	Reference positive voltage
VREFL ^{note1}	Input	Reference negative voltage (external analog ground)
DACx_OUTy	Output	Analog Voltage Output

Note 1: Some packages do not have VREFH/VREFL terminals, which are internally connected to AVCC/AVSS respectively.

18.3 Functional Description

18.3.1 DAC Conversion

Each conversion channel can perform DAC conversion independently. When DACx_DACR.DA1E is set to 1, the DAC conversion of the corresponding channel starts and the analog voltage is output from the DACx_OUTy port. If the output analog voltage is only used as the input voltage of CMP, the DACx_DAOCR register can be used to turn off the DACx_OUTy output and release the external port.

The following is an instance of using DAC1_ch1 to perform single-channel conversion. The action sequence is shown in Figure 18-2.

1. Set the DAC conversion data (DAC1_DADR1) and data format (DAC1_DACR.DPSEL).
2. DAC conversion starts after setting DAC1_DACR.DA1E to 1. After the conversion time $t_{D CONV}$, the conversion result is output from port DAC1_OUT1 and held until DAC1_DADR1 is rewritten or DAC1_DACR.DA1E is set to 0.
3. When DAC1_DACR.DA1E is 1, rewriting the value of DADR will trigger a new DAC conversion. Similarly, after the conversion time $t_{D CONV}$, a new conversion result is output from port DAC1_OUT1.
4. Set the DAC1_DACR.DA1E bit to 0 to turn off DAC1_ch1 and make DAC1_OUT1 output high impedance.

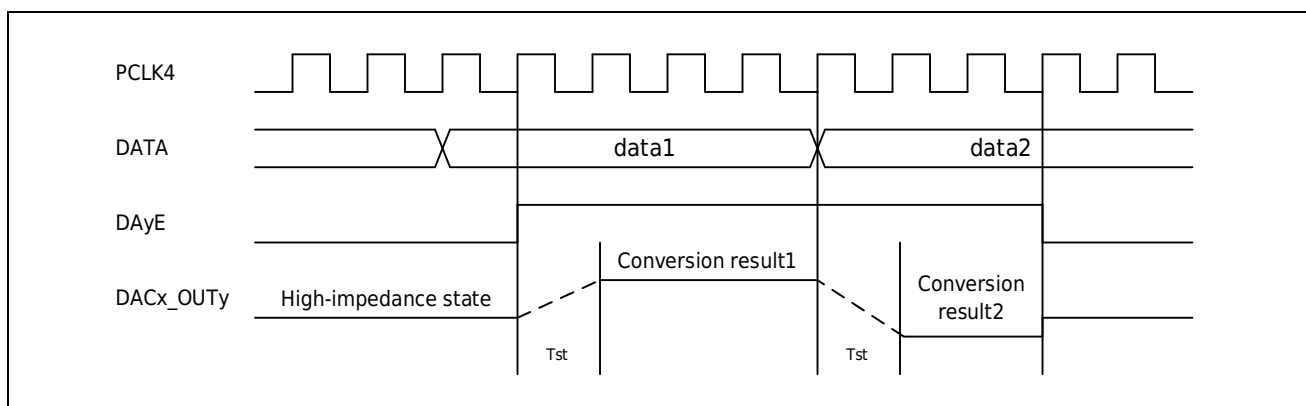


Figure 18-2 DAC Conversion Diagram

18.3.2 Synchronization Conversion

Synchronous conversion of two conversion channels of the same DAC unit can be achieved by synchronously updating the conversion data.

Below is an instance of using DAC1 to perform two-channel simultaneous conversions.

1. Set the DAC conversion data (DAC1_DADR1,2) 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 tDCONV , the conversion results are output from ports DAC1_OUT1, 2 respectively and are kept until DADR is rewritten or DAC1_DACR.DAE is set to 0.
3. When DAC1_DACR.DAE is 1, rewriting the values of DAC1_DADR1,2 at the same time will synchronously trigger the two channels to start a new DAC conversion. Similarly, after the conversion time tDCONV , the new conversion result is output from DAC1_OUT1,2.
4. Set DAC1_DACR.DAE to 0, the two channels are turned off synchronously, and DAC1_OUT1,2 outputs high impedance.

18.3.3 Output Voltage

Each DAC channel can output two voltages: DAXOy and DACx_OUTy. DAXOy is for internal use only, DACx_OUTy can be output to an external pin.

$$DAXOy = \text{ConversionData} / 4096 \times VREFH$$

$$DACx\_OUTy = \text{ConversionData} / 4096 \times VREFH$$

DAXOy always has output when the DAC channel is enabled, and DACx_OUTy is disabled when DACx_DAOOCR.DAODISy = 1. Both DAXOy and DACx_OUTy can be used for analog input of CMP.

18.3.4 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 effectively avoids this situation 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_DADRY 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_DADRY is rewritten, the DAC conversion will start after 2 ADCLK periods. DACx_DAADPCR.ADCSLn (n=1~3) is used to select the ADC channel with priority for ADC conversion.

The following takes DAC1_ch1 as an instance to explain the steps for setting the ADC conversion priority mode. The action sequence is shown in Figure 18-3.

1. Confirm that ADC is in stop state and set DAC1_DAADPCR.ADCSL;
 2. Confirm that ADC is in stop state and set DAC1_DACR.DA1E to 1;
 3. Write the conversion data to DAC1_DADR1.
- When the conversion data A is written to DAC1_DADR1, ADC_STR.STRT is 0, the ADC is in the stopped state, and the DAC conversion starts after 2 ADCLK periods.
 - When conversion data B is written to DAC1_DADR1, ADC_STR.STRT is 1, ADC is performing ADC conversion, and DAC conversion will not start until 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.

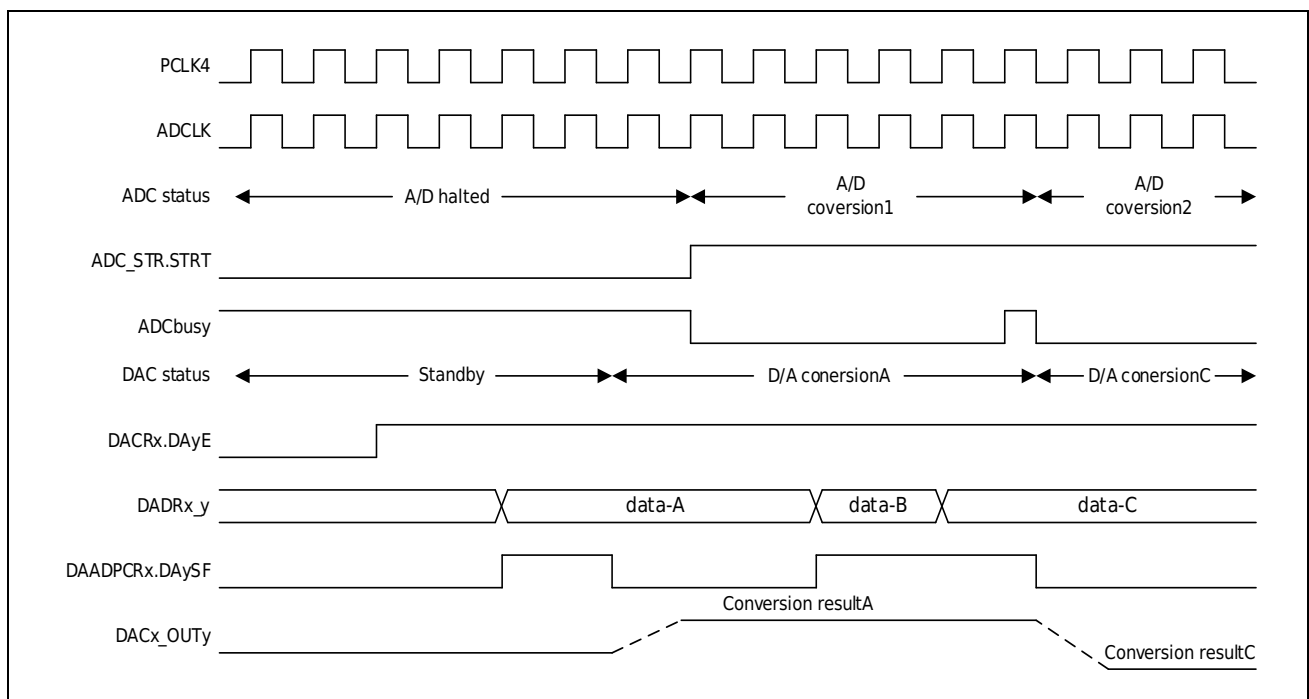


Figure 18-3 ADC Conversion Priority Mode Operation Diagram

18.4 Precautions

18.4.1 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. See the Low Power Mode section for details.

18.4.2 Stop Low Power Mode

If the system enters stop low power mode during DAC conversion, the analog output will be held and the current flowing through the analog supply will be the same as when the DAC is converting. If you need to further reduce the power consumption in stop mode, set the DAyE and DAE bits of DACx_DACR to all 0s.

18.4.3 Power-down Low Power Mode

If the system enters power-down low-power mode while the DAC is converting, the analog output will be placed in a high-impedance state.

18.5 Register Description

Table 18-2 DACx Register List

DAC1 reference address: 0x40041000

Register Name	Symbol	Offset Address	Bit Width	Reset Value
DACx Data Register 1	DACx_DADR1	0x00	16	0x0000
DACx Data Register 2	DACx_DADR2	0x02	16	0x0000
DACx Control Register	DACx_DACR	0x04	16	0x0000
DACx ADC Conversion Priority Control Register	DACx_DAADPCR	0x06	16	0x0000
DACx Analog Output Control Register	DACx_DAOCR	0x1c	16	0x0000

In the table, x=1.

All registers are retained across software reset and watchdog reset.

18.5.1 DAC Data Register (DACx_DADRx , 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
-	-	-	-	DADR [11:0]											

Bit	Flag	Bit Name	Function	Read and write
b15~b12	Reserved	-	Read 0 when reading, write 0 when writing	R/W
b11~b0	DADR [11:0]	transformation Data	Transformation data setting value	R/W

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]												-	-	-	-

Bit	Flag	Bit Name	Function	Read and write
b15~b4	DADR [11:0]	transformation Data	Transformation data setting value	R/W
b3~b0	Reserved	-	Read 0 when reading, write 0 when writing	R/W

DACx_DADRx is used to store DAC conversion data, one for each channel. As long as the DAC conversion is enable, the value in DACx_DADRx is converted to an analog voltage and output from the analog port. 32 -bit operation on DACx_DADR1 of the same unit can realize dual-channel synchronous conversion.

18.5.2 DAC Control Register (DACx_DACR)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	DP SEL	-	-	-	-	-	DA2 E	DA1 E	DAE

Bit	Flag	Bit Name	Function	Read and write
b15~b9	Reserved	-	Read at "0", write at "0"	R/W
b8	DPSEL	DADR Data Alignment Mode Selection	0: DADR data is right aligned 1: DADR data is left-aligned	R/W
b7~b3	Reserved	-	Read at "0", write at "0"	R/W
b2	DA2E	Channel 2 enable	This bit setting valid when DAE=0 0: Channel 2 disabled 1: Channel 2 enabled	R/W
b1	DA1E	Channel 1 enable	This bit setting valid when DAE=0 0: Channel 1 disabled 1: Channel 1 enabled	R/W
b0	DAE	Always Enable	0: Channel 1, 2 disabled 1: Channel 1, 2 enabled	R/W

DACR is used to control DAC conversion and DAC output. See Table 18-3 for details.

When a channel is disabled, the analog output of the channel is also disabled and the port is in a high-impedance state.

In ADC conversion priority mode, please set DA1E , DA2E and DAE bits when 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.

18.5.3 DAC Analog Output Control Register (DACx_DAOCR)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DAO DIS2	DAO DIS1	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Bit	Flag	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	R/W
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	R/W
b13~b0	Reserved	-	Read at "0", write at "0"	R/W

Table 18-3 DAC Conversion and Analog Output Control

DAE	DA1E/ DA2E	DAODIS1/ DAODIS2	DAC Conversion	DAC_OUT Output	DAO Output
0	0	All	Stop	High Impedance State	High Impedance State
		0	Start	Normal Output	Normal Output
		1	Start	High Impedance State	Normal Output
1	0	0	Start	Normal Output	Normal Output
		1	Start	High Impedance State	Normal Output
	1	0	Start	Normal Output	Normal Output
		1	Start	High Impedance State	Normal Output

18.5.4 DAC ADC Conversion Priority Control Register (DACx_DAADPCR)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ADPEN	-	-	-	-	-	DA2SF	DA1SF	-	-	-	-	-	ADCSL3	ADCSL2	ADCSL1

Bit	Flag	Bit Name	Function	Read and write
b15	ADPEN	ADC conversion priority enable	0: ADC conversion priority mode invalid 1: ADC conversion priority mode valid	R/W
b14~b10	Reserved	-	Read at "0", write at "0"	R/W
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	R
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	R
b7~b3	Reserved	-	Read at "0", write at "0"	R/W
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	R/W
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	R/W
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	R/W

DACx_DAADPCR is used to control the ADC conversion priority mode.

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.

DA1SF and DA2SF are DA conversion status flags, reflecting whether the data in the current channel data register has been converted.

19 Temperature Sensor (OTS)

19.1 Introduction

The temperature sensor (On-chip Temperature Sensor, hereinafter referred to as OTS) can obtain the temperature inside the chip to support the reliable operation of the system. OTS provides a set of temperature-related digital quantities, and the temperature value can be obtained through calculation. When not in use, it can be turned off by the module stop function to reduce system power consumption.

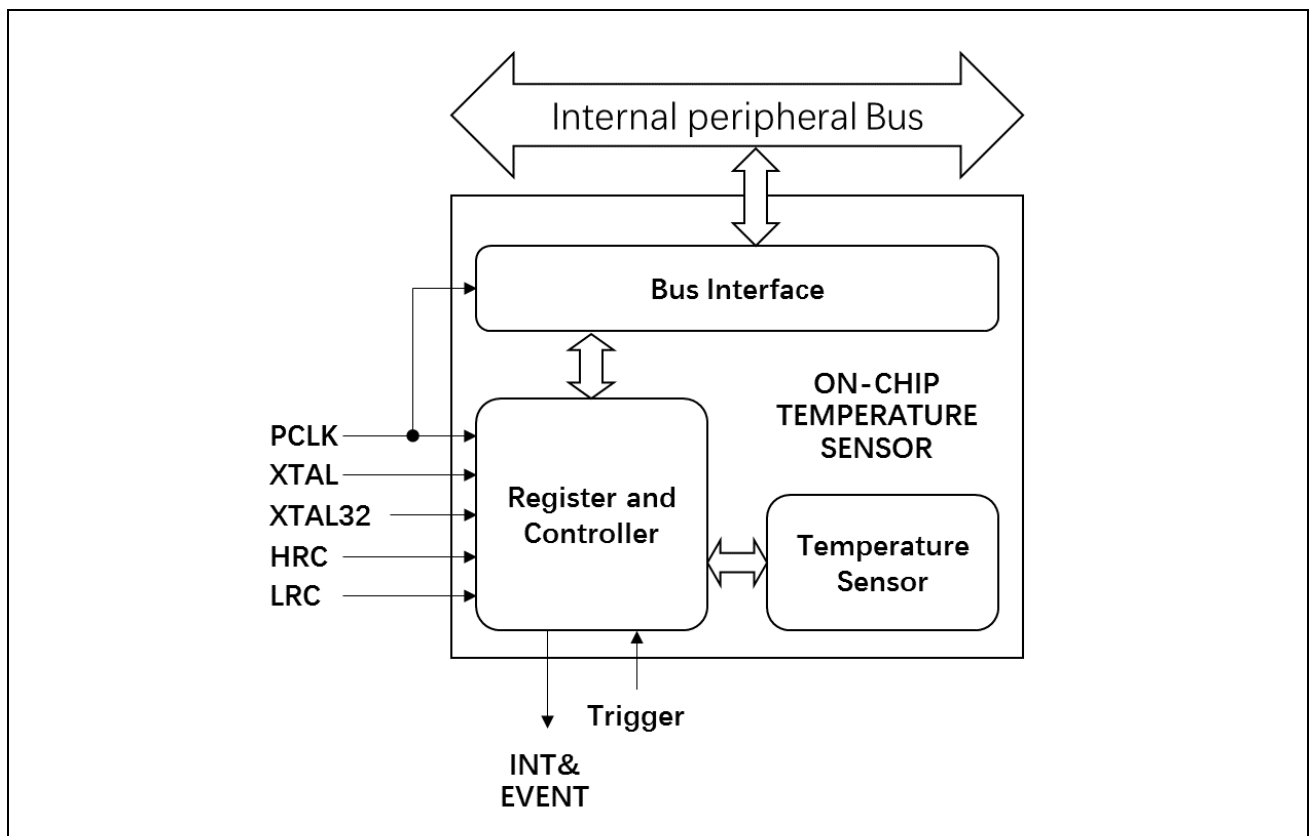


Figure 19-1 OTS functional block diagram

19.2 Usage Notes

using OTS to obtain the internal temperature of the chip, it must be calibrated first, and the measurement accuracy is related to the calibration method. Please refer to the following calibration experiment instructions for calibration. When using, please turn off the module stop function and start the internal low-speed clock LRC. At the same time, please choose to start the internal high-speed clock HRC, the external high-speed clock XTAL and the external low-speed clock XTAL32 according to the usage. After setting the register OTS_CTL.OTSST to "1", the temperature measurement starts, and the OTSST bit will be automatically cleared after the temperature measurement is completed. After confirming that OTSST is "0", read the temperature parameters

in the registers OTS_DR1, 2 and OTS_ECR, and use the following calculation formula to obtain the temperature value.

$$T = K \times (1/D1 - 1/D2) \times E_{hrc} + M$$

[Parameter Description]

T: temperature (°C)

K: temperature slope (determined by calibration experiment)

D1 : Temperature data 1 (read from register OTS_DR1)

D2 : Temperature data 2 (read from register OTS_DR2)

E_{hrc}: HRC frequency error compensation data (read from register OTS_ECR)

M: Temperature offset (determined by calibration experiment)

The register OTS_CTL.OTSCK is used to select the temperature measurement clock. When the HRC action is selected, its frequency error may affect the accuracy of the final calculated temperature. To eliminate this error, start XTAL32 before measuring the temperature and use E_{hrc} when calibrating and calculating the temperature. When selecting XTAL action, ignore E_{hrc} and show it as a constant 1.

[Calibration experiment]

Calibration experiments were carried out at two defined temperatures to calculate K and M.

$$K = (T2 - T1) / (A2 - A1)$$

$$M = T1 - K \times A1 = T2 - K \times A2$$

T1: experimental temperature 1

T2: experimental temperature 2

$$A1: (1/D1_{T1} - 1/D2_{T1}) \times E_{hrcT1}$$

D1_{T1}, D2_{T1}, E_{hrcT1} are D1, D2, E_{hrc} measured at temperature T1 respectively;

$$A2: (1/D1_{T2} - 1/D2_{T2}) \times E_{hrcT2}$$

D1_{T2} , D2_{T2} , E_{hrcT2} are D1, D2, E_{hrc} measured at temperature T2 respectively;

For reference, three sets of temperature data are preset inside the chip, and the data of two temperature points can be arbitrarily selected for calibration. Since the test environment temperature may vary, the preset data is for reference only.

Table 19-1 OTS Preset Temperature Data

Test temperature	Reading method		Test Condition
	Temperature data 1 (D1)	Temperature data 2 (D2)	
Ta= -40℃	OTS_PDR3 [15:0]	OTS_PDR3 [31:16]	XTAL=8MHz
Ta= 25℃	OTS_PDR1 [15:0]	OTS_PDR1 [31:16]	XTAL=8MHz
Ta= 125℃	OTS_PDR2 [15:0]	OTS_PDR2 [31:16]	XTAL=8MHz

E_{hrc} is a calibration value to eliminate the influence of the error of HRC itself on the measured temperature. It can be read from the register OTS_ECR and is used as follows.

Table 19-2 Ehrc Usage and Setting Method

Action Clock	Calibration experiment (calculation of K,M)		Temperature measurement
(OTSCK)	Use preset temperature data	Re-acquire temperature data	(Calculate T)
XTAL	$E_{hrc}=1$	$E_{hrc}=1$	$E_{hrc}=1$
HRC	$E_{hrc}=f_{hrc}(MHz)/0.032768$	$E_{hrc}=OTS\_ECR [15:0]$	$E_{hrc}=OTS\_ECR [15:0]$

f_{hrc} is the standard frequency of HRC , that is, 16MHz or 20MHz.

The register OTS_CTL.TSSTP is used to select whether to turn off the analog temperature sensor after the temperature measurement is completed. The initial value of TSSTP is "0", which means that the analog temperature sensor is turned on after a temperature measurement is completed, so that the stabilization time of the analog temperature sensor from off to on will be automatically skipped in the next temperature measurement. To turn off the analog temperature sensor after each temperature measurement, please set TSSTP to "1".

The temperature measurement can be triggered by other peripheral events. When using it, please set the trigger target of the trigger source to OTS. When the temperature measurement is completed, an event can also be generated to trigger the start of other peripherals. When using it, please set the register OTS_TRGSEL to select the trigger target. Please set the register OTS_CTL.OTSIE to "1" when using the temperature measurement completion interrupt.

19.3 Register Description

Table 19-3 List of OTS registers

Reference address: 0x4004A800

Register Name	Symbol	Offset Address	Bit Width	Reset Value
OTS control register	OTS_CTL	0x00	16	0x0000
OTS data register 1	OTS_DR1	0x02	16	0x0000
OTS data register 2	OTS_DR2	0x04	16	0x0000
OTS Error Compensation Register	OTS_ECR	0x06	16	0x0000

Reference address: 0x40010600

Register Name	Symbol	Offset Address	Bit Width	Reset Value
OTS preset data register 1	OTS_PDR1	0xe4	32	Undefined value
OTS preset data register 2	OTS_PDR2	0xf4	32	Undefined value
OTS preset data register 3	OTS_PDR3	0xf8	32	Undefined value

19.3.1 OTS Control Register (OTS_CTL)

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	TSST P	OTS IE	OTS CK	OTS ST

Reset value: 0x0000

Bit	Flag	Bit Name	Function	Read and write
b15~b4	Reserved	-	Read 0 when reading, write 0 when writing	R/W
b3	TSSTP	Turn off the analog temperature sensor	Select whether to automatically turn off the analog temperature sensor after the temperature measurement is completed 0: Do not turn off the analog temperature sensor 1: Turn off the analog temperature sensor	R/W
b2	OTSIE	Interrupt enable bit	0: Disable temperature measurement end interrupt request 1: Allow temperature measurement end interrupt request	R/W
b1	OTSCK	clock select bit	0: Select external high-speed clock (XTAL) action 1: Select Internal high-speed clock (HRC) action	R/W
b0	OTSST	Temperature measurement start bit	0: stop temperature measurement 1: start temperature measurement Set the "1" condition: (1) Software set "1" (2) Hardware trigger set to "1" Clear "0" condition: (1) Software "0" (2) Automatically clear "0" after temperature measurement	R/W

19.3.2 OTS Data Register 1 (OTS_DR1)

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TSDC [15:0]															

Reset value: 0x0000

Bit	Flag	Bit Name	Function	Read and write
b15~b0	TSDC [15:0]	Temperature data D1	Temperature data D1 It will be automatically updated after the temperature measurement is completed. Please confirm that OTS_CTL.OTSST is "0" before reading.	R

19.3.3 OTS Data Register 2 (OTS_DR2)

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TSDC [15:0]															
Reset value: 0x0000															

Bit	Flag	Bit Name	Function	Read and write
b15~b0	TSDC [15:0]	Temperature data D2	Temperature data D2 It will be automatically updated after the temperature measurement is completed. Please confirm that OTS_CTL.OTSST is "0" before reading.	R

19.3.4 OTS Error Compensation Register (OTS_ECR)

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TSEC [15:0]															
Reset value: 0x0000															

Bit	Flag	Bit Name	Function	Read and write
b15~b0	TSEC [15:0]	Error factor	Error coefficient Ehrc It will be automatically updated after the temperature measurement is completed. Please confirm that OTS_CTL.OTSST is "0" before reading.	R

19.3.5 OTS Preset Temperature Data Register (OTS_PDR1,2,3)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
TSPD2 [15:0]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TSPD1 [15:0]															
Reset value: Undefined															

Bit	Flag	Bit Name	Function	Read and write
b31~b16	TSPD2 [15:0]	Preset temperature data D2	Preset temperature data for calibration. This data is measured at 8MHz. Please convert it with the actual frequency of the action clock selected by OTSCK before use. $D2 = TSPD2 \times f_{OTSCK} / 8$, f_{OTSCK} is the frequency of the actual action clock, in MHz.	R
b15~b0	TSPD1 [15:0]	Preset temperature data D1	Preset temperature data for calibration. This data is measured at 8MHz. Please convert it with the actual frequency of the action clock selected by OTSCK before use. $D1 = TSPD1 \times f_{OTSCK} / 8$, f_{OTSCK} is the frequency of the actual action clock, in MHz.	R

20 Advanced Control Timer (Timer6)

20.1 Introduction

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 and 3-phase orthogonal encoding; supports EMB control. This series of products is equipped with 10 units of Timer6.

20.2 Basic Block Diagram

The basic functions and characteristics of Timer6 are shown in Table 20-1.

Table 20-1 Basic Functions and Features of Timer6

Waveform mode	Sawtooth wave (counting up and down), triangle wave (counting up and down)
Basic functions	• Capture input
	• Software synchronization
	• Hardware synchronization
	• Cache function
	• Pulse width measurement
	• Period measurement
	• Orthogonal coding count
	• General PWM output
	• EMB control
Interrupt output	Count comparison match interrupt
	Count period match interrupt
	Dead zone error interrupt
Event output	Count compare match events
	Count period match event

The basic block diagram of Timer6 is shown in Figure Figure 20-1. As shown in the block diagram, "< t >" represents the unit number, that is, "< t >" is 1~10. This section refers to "< t >" when referring to "< t >" later.

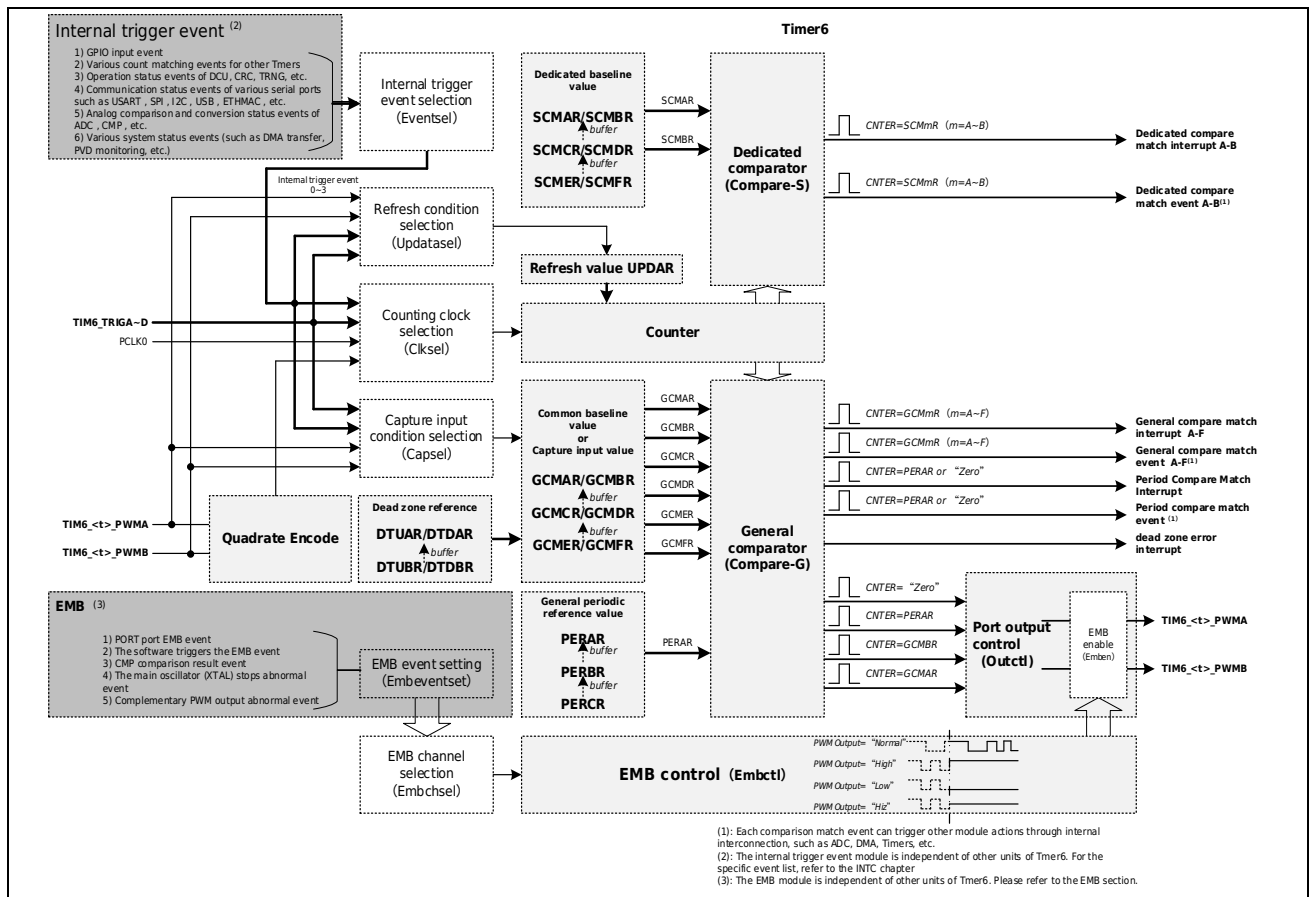


Figure 20-1 Timer6 Basic Block Diagram

Table 20-2 shows the list of input and output ports of Timer6.

Table 20-2 Timer6 Port List

Port Name	Direction	Function
TIM6_<t>_PWMA	in or out	1) Quadrature encoding count clock input port or capture input port or comparison output port 2) Hardware start, stop, clear, refresh condition input port
TIM6_<t>_PWMB		
TIM6_TRIGA	in	1) Hardware count clock input port or capture input port 2) Hardware start, stop, clear, refresh condition input port
TIM6_TRIGB		
TIM6_TRIGC		
TIM6_TRIGD		

20.3 Functional Description

20.3.1 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 20-2 and Figure 20-3.

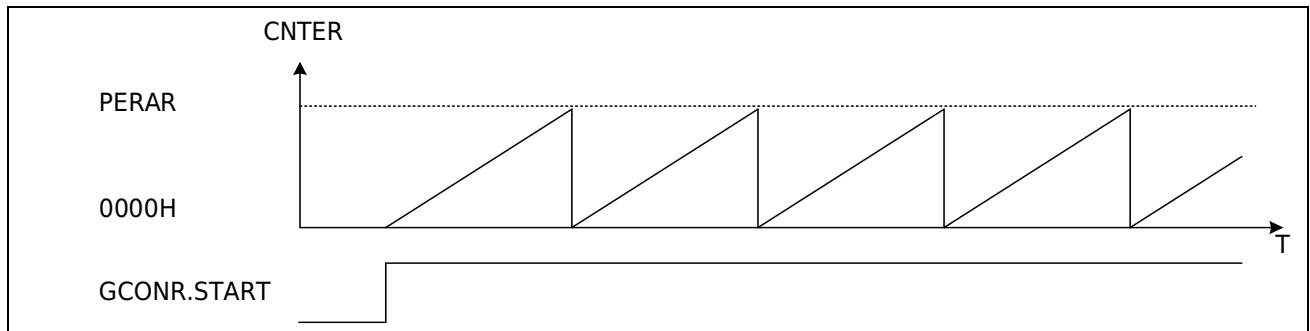


Figure 20-2 Sawtooth Waveform (Counting Up)

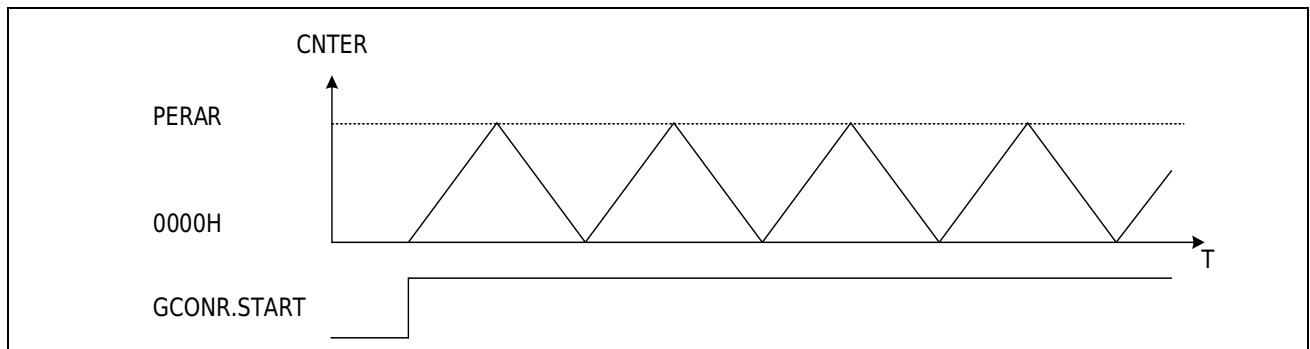


Figure 20-3 Triangle Waveform

20.3.2 Timer Selection

The counting clock of Timer6 can have the following options:

- PCLK0 and PCLK0 divided by 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 (set by GCONR.CKDIV [3:0])
- Internal trigger event input 0~3 (set by HCUPR.HCUP [11:8] or HCDOR.HCDO [11:8])
- TIM6_TRIGA~D port input (HCUPR.HCUP [23:16] or HCDOR.HCDO [23:16] setting)
- Quadrature encoding relationship input of TIM6_<t>_PWMA and TIM6_<t>_PWMB (set by HCUPR.HCUP [7:0] or HCDOR.HCDO [7:0])

The counting clock source is a software counting mode, and the counting clock source 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.

20.3.3 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.

20.3.3.1 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.

20.3.3.2 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.

20.3.4 Comparison Output

Timer6 of each unit has 2 comparison output ports (TIM6_<t>_PWMA, TIM6_<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 TIM6_<t>_PWMA and TIM6_<t>_PWMB respectively. When the timer count value is equal to GCMAR , the TIM6_<t>_PWMA or TIM6_<t>_PWMB port outputs the specified level.

The counting start level, counting stop level, counting overflow level, counting compare match level, etc. of the TIM6_<t>_PWMA port can be set by the PCNAR.STACA, PCNAR.STPCA, PCNAR.OVFCA, PCNAR.UDFCA, PCNAR.CMAU<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 TIM6_<t>_PWMB port can be set by the PCNBR.STACB, PCNBR.STPCB, PCNBR.OVFCA, PCNBR.UDFCB, PCNBR.CMAU<D>B, and PCNBR.FORCB bits of the port control register (PCNBR).

Figure 20-4 This is an instance of the operation of the comparison output.

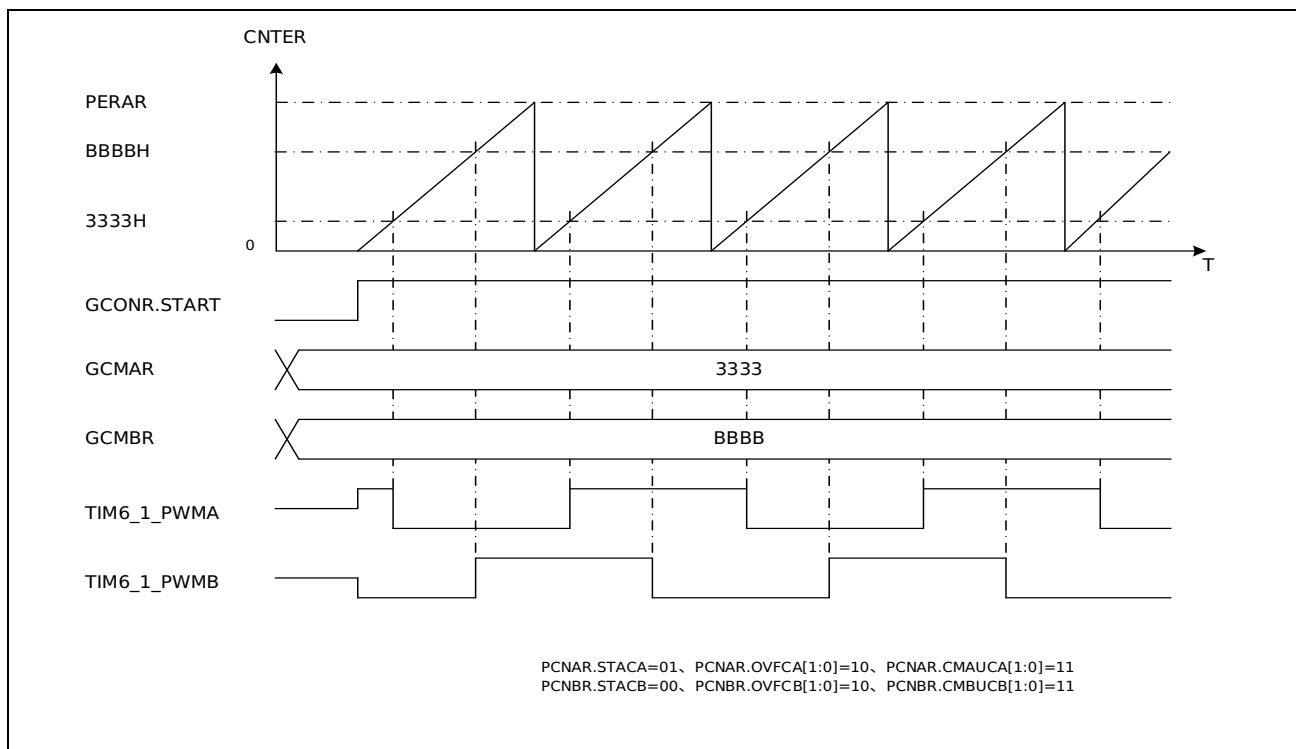


Figure 20-4 Compare Output Action

20.3.5 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 bit in the port control register (PCNAR) to 1 enables the capture input function. When the corresponding capture input condition is set and valid, the current count value is saved in the corresponding capture register (GCMAR , GCMBR), and the CMAF bit of the status flag register (STFLR) is 1.

The conditions for each capture input of each unit can be internal trigger event input 0~3, port input of TIM6_TRIGA~D, port input of TIM6_<t>_PWMA or TIM6_<t>_PWMB, etc. The specific condition selection can be set through the hardware capture event selection register (HCPAR, HCPBR).Figure 20-5 An instance of an action for capturing input.

20.3.7 Software Synchronization

20.3.7.1 Software Synchronization Start

Each unit can achieve synchronous start of the target unit counter (COUNTER) by setting the relevant bits of the software synchronous start control register (SSTAR).

20.3.7.2 Software Co-stop

Each unit can achieve the synchronous stop of the target unit counter (COUNTER) by setting the relevant bits of the software synchronous stop control register (SSTPR).

20.3.7.3 Software Synchronous Clear

Each unit can achieve synchronous clearing of the target unit counter (COUNTER) by setting the relevant bits of the software synchronous clear control register (SCLRR).

20.3.7.4 Software Synchronization Refresh

Each unit can achieve synchronous refresh of the target unit counter (COUNTER) by setting the relevant bits of the software synchronous refresh control register (SUPDR).

As shown in Figure 20-7, 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.

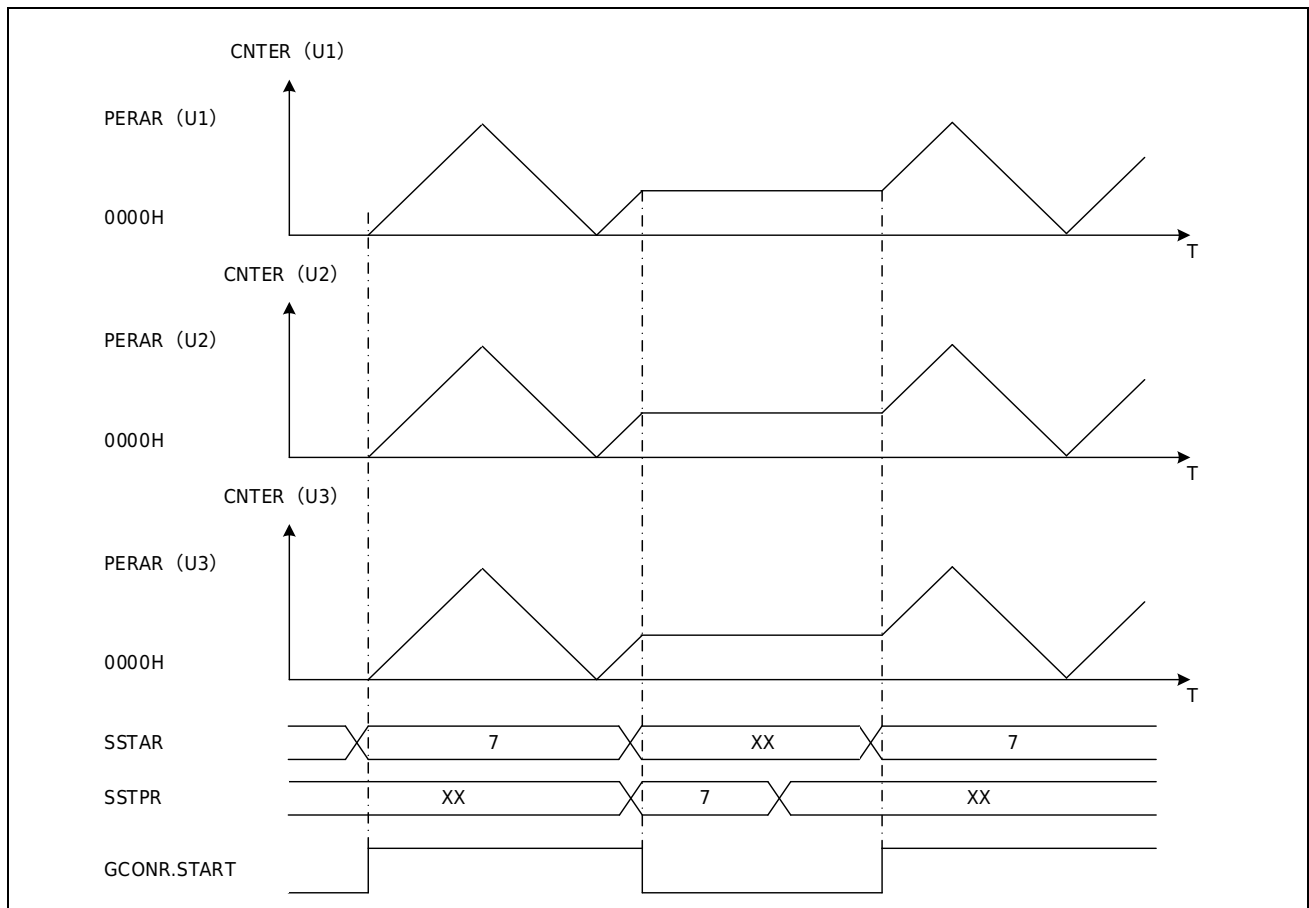


Figure 20-7 Software Synchronization Action

20.3.8 Hardware Synchronization

In addition to two general-purpose input ports (TIM6_<t>_PWMA, TIM6_<t>_PWMB) independently, each unit also has four general-purpose external trigger input ports (TIM6_TRIGA, TIM6_TRIGB, TIM6_TRIGC, TIM6_TRIGD) and four internal trigger event input conditions, which can realize hardware synchronization 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_TRGSEL0~3). For specific event correspondence, please refer to the INTC chapter. When using the internal hardware trigger function, the function clock control register (PWC_FCG0) peripheral circuit trigger function is required to enable position 0.

20.3.8.1 Hardware Synchronization 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).

20.3.8.2 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).

20.3.8.3 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 selection register (HCLRR).

20.3.8.4 Hardware Synchronized 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).

20.3.8.5 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).

20.3.8.6 Hardware Synchronized 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 20-8 shows an instance of hardware synchronization operation of units 1 and 3.

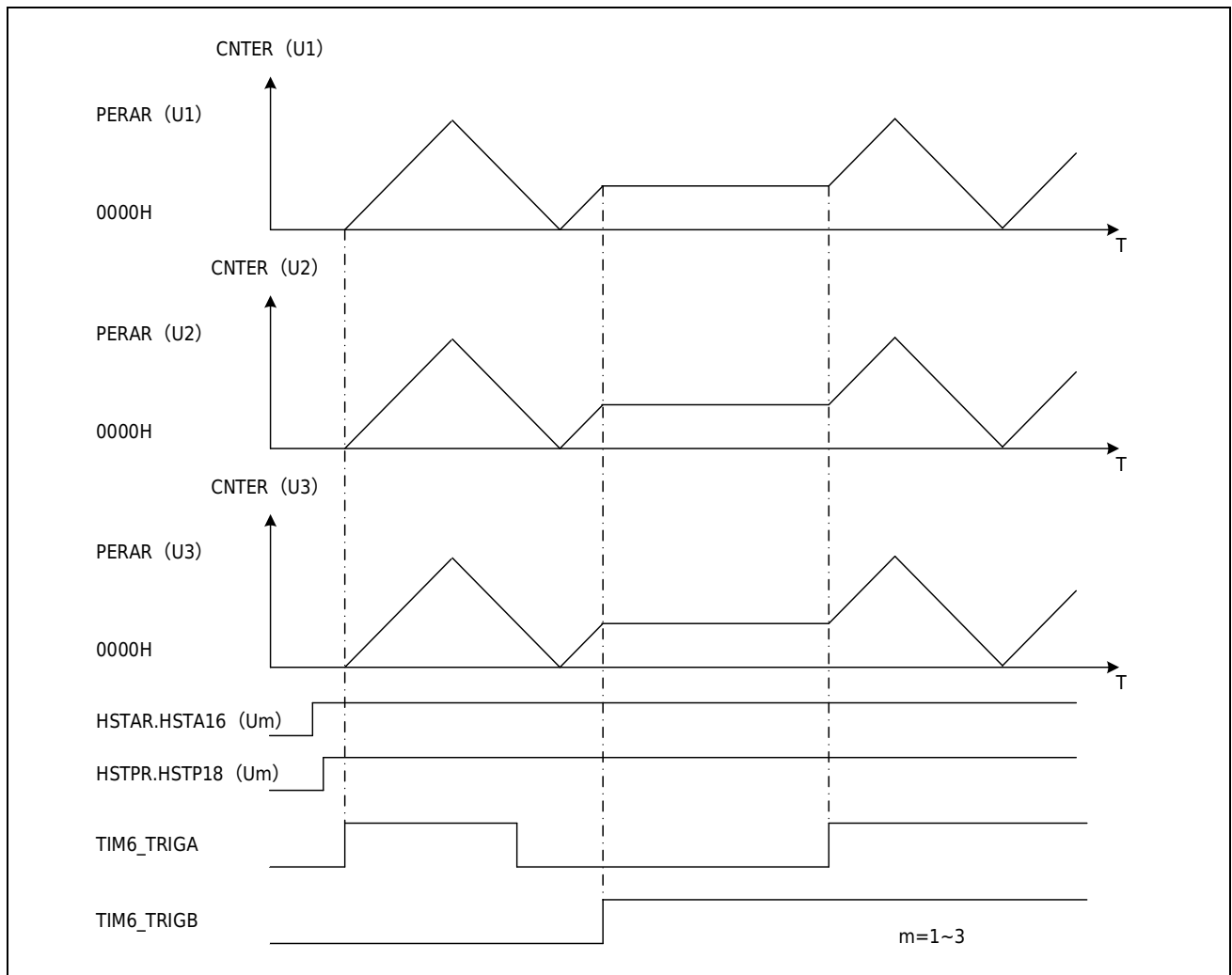


Figure 20-8 Hardware Synchronous Action

20.3.9 Pulse Width Measurement

Please refer to the Hardware Synchronization section when using the hardware trigger related functions of the TIM6_TRIGA~D port and TMR6_<t>_PWMA~B port. Each unit can realize 2 independent pulse width measurement functions.

For example, by setting the hardware start condition of the counter to the rising edge of TIM6_TRIGA, and the hardware clear condition, stop condition, and capture input condition of the GCMAR register to the falling edge of TIM6_TRIGA, continuous pulse width measurement can be achieved. The corresponding actions are shown in Figure 20-9.

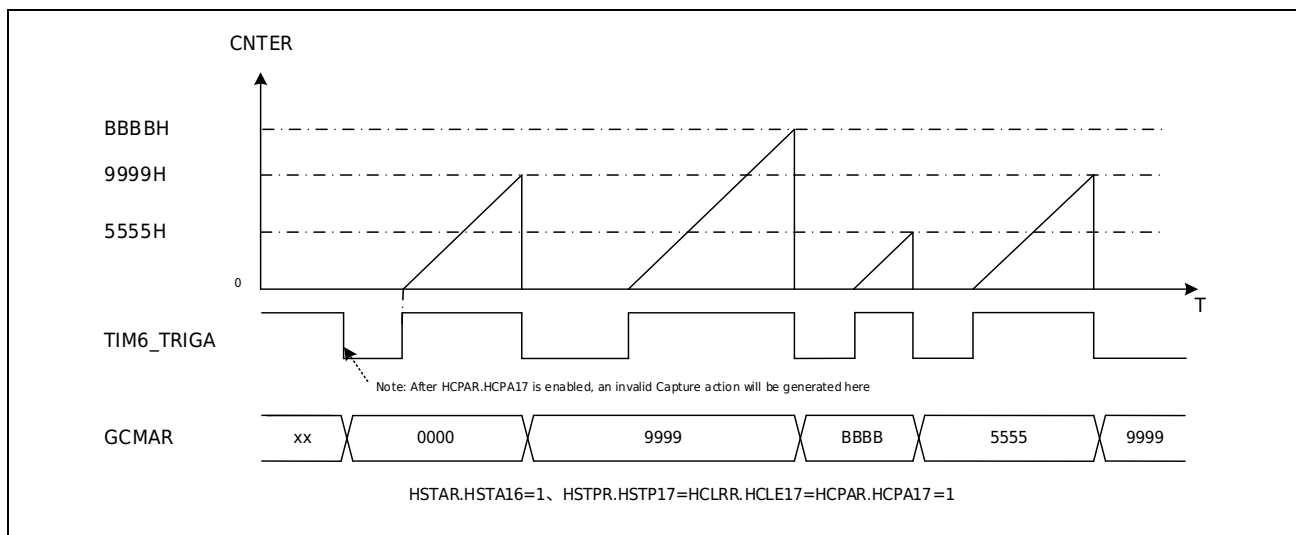


Figure 20-9 Pulse Width Measurement

20.3.10 Periodic Measurement

Please refer to the Hardware Synchronization section when using the hardware trigger related functions of the TIM6_TRIGA~D ports and TMR6_<t>_PWMA~B ports. Each unit can implement 2 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 TIM6_TRIGB, continuous period measurement can be achieved. The corresponding actions are shown in Figure 20-10.

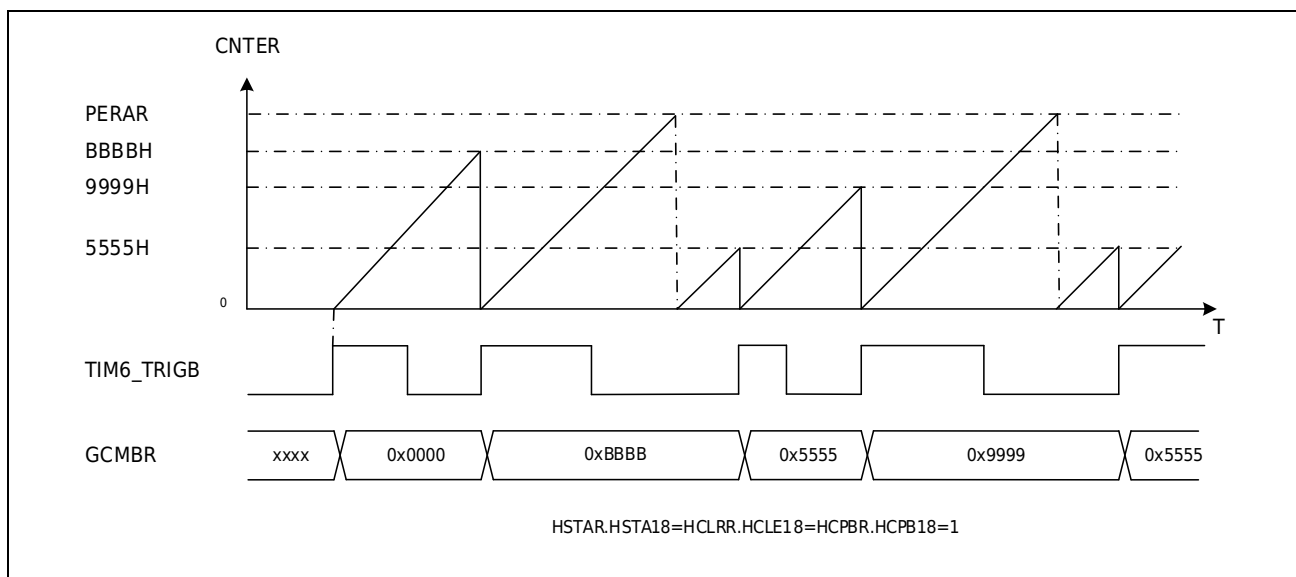


Figure 20-10 Cycle Measurement

20.3.11 Cache Function

The general period reference value, general comparison reference value, special comparison reference value, dead zone reference value, etc. of Timer6 all have a cache function, which can realize period changes, duty period changes, dead zone changes, etc. in hardware during counting. The period reference value, general comparison reference value and special comparison reference value have single buffer and double buffer functions, and the dead zone reference value has single buffer function.

20.3.11.1 Single Cache Action

Single buffer action means that by setting the buffer control register (BCONR.BENA<P><SPA><SPB>=1, BCONR.BSEA<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 general period reference buffer register (PERBR) is automatically transferred to the general period reference register (PERAR)
- b) The value of the general comparison reference value buffer register (GCMCR, GCMDR) is automatically transferred to the general comparison reference value register (GCMAR, GCMBR) (when comparing output)
- c) The value of the general comparison reference value register (GCMAR, GCMBR) is automatically transferred to the general 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 zone reference buffer register (DTUBR, DTDBR) is automatically transferred to the dead zone reference register (DTUAR, DTDAR)

As shown in Figure 20-11, this is the timing diagram of the single cache mode of the General 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 general comparison reference value register (GCMAR) during counting can adjust the output duty period, and changing the value of the general period reference value register (PERAR) can adjust the output period.

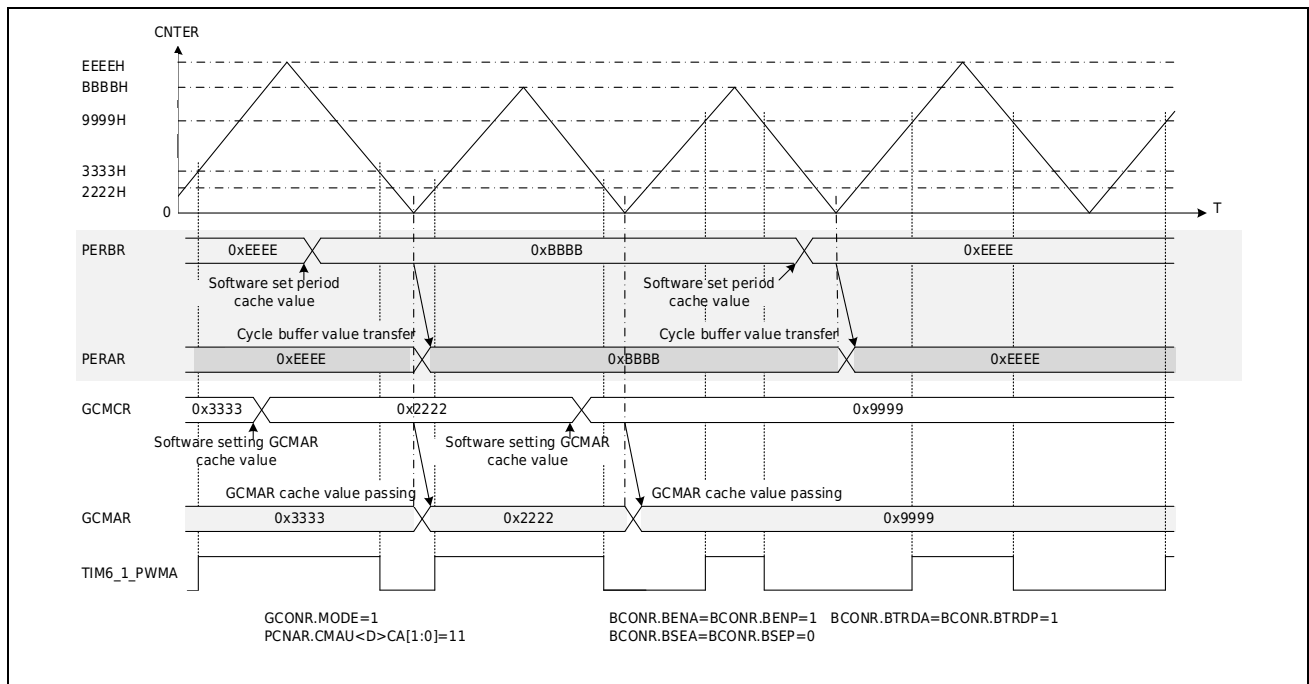


Figure 20-11 Single-buffer Mode Comparison Output Timing

20.3.11.2 Double Buffering Action

The double buffering action means that by setting the buffer control register (BCONR.BENA<P><SPA><SPB>=1, BCONR.BSEA<P><SPA><SPB>=1), the following events are selected to occur at the buffer transfer time point:

- The value of the general period reference value buffer register (PERBR) is automatically transferred to the general period reference value register (PERAR), and the value of the general period reference value double buffer register (PERCR) is automatically transferred to the general period reference value buffer register (PERBR)
- The value of the general comparison reference value buffer register (GCMCR, GCMDR) is automatically transferred to the general comparison reference value register (GCMAR, GCMBR), and the value of the general comparison reference value double buffer register (GCMER, GCMFR) is automatically transferred to the general comparison reference value buffer register (GCMCR, GCMDR) (when comparing output)
- The value of the general comparison reference value buffer register (GCMCR, GCMDR) is automatically transferred to the general comparison reference value double buffer register (GCMER, GCMFR), and the value of the general comparison reference value register (GCMAR, GCMBR) is automatically transferred to the general comparison reference value buffer In registers (GCMCR, GCMDR) (when capturing input)
- 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 20-12, it is a timing diagram of the double buffer mode when the internal trigger event 0 triggers the capture input.

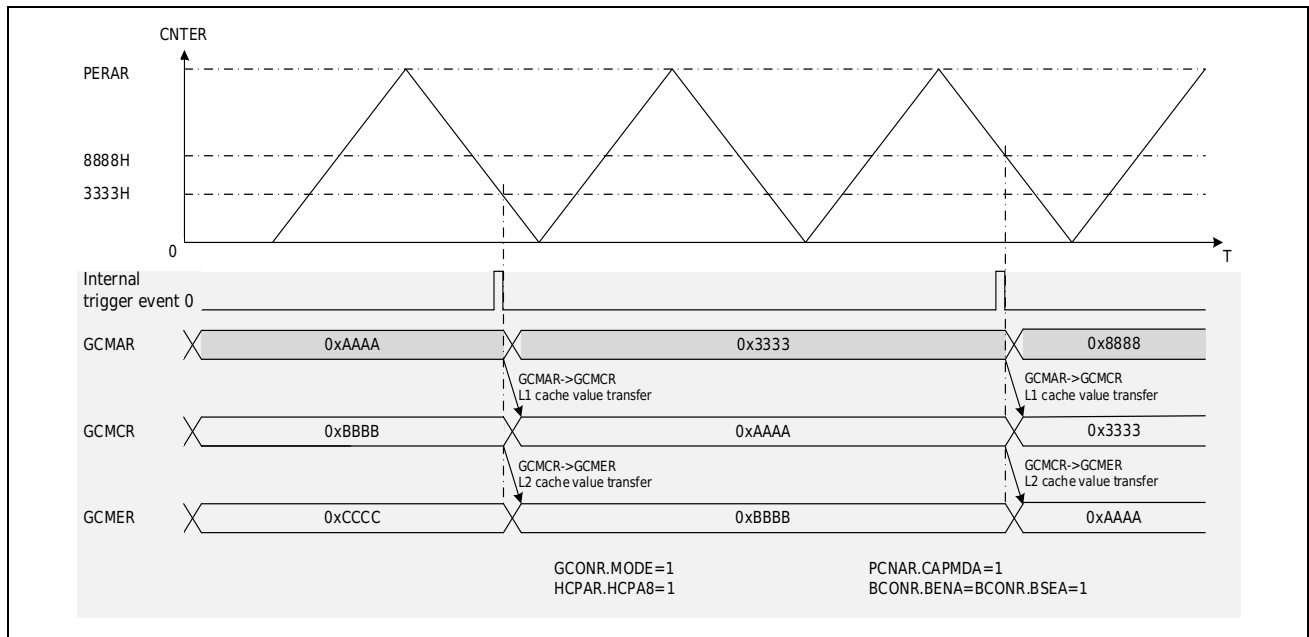


Figure 20-12 Double Buffer Capture Input Timing

20.3.11.3 Cache TransferNode

Compare Output (Sawtooth Wave) Buffer Transfer

When the cache function valid ($BCONR.BENA<P><SPA><SPB>=DCONR.DTBENU<D>=1$) && comparison output count ($PCNAR.CAPMDA=0$) && sawtooth counting mode ($GCONR.MODE=0$), the cache transfer of the General period reference value, General comparison reference value, dedicated comparison reference value, and dead zone reference value occurs at the overflow point of the up-counting or the underflow point of the down-counting.

As shown in Figure 20-13, this is the cache action when the sawtooth wave counts up.

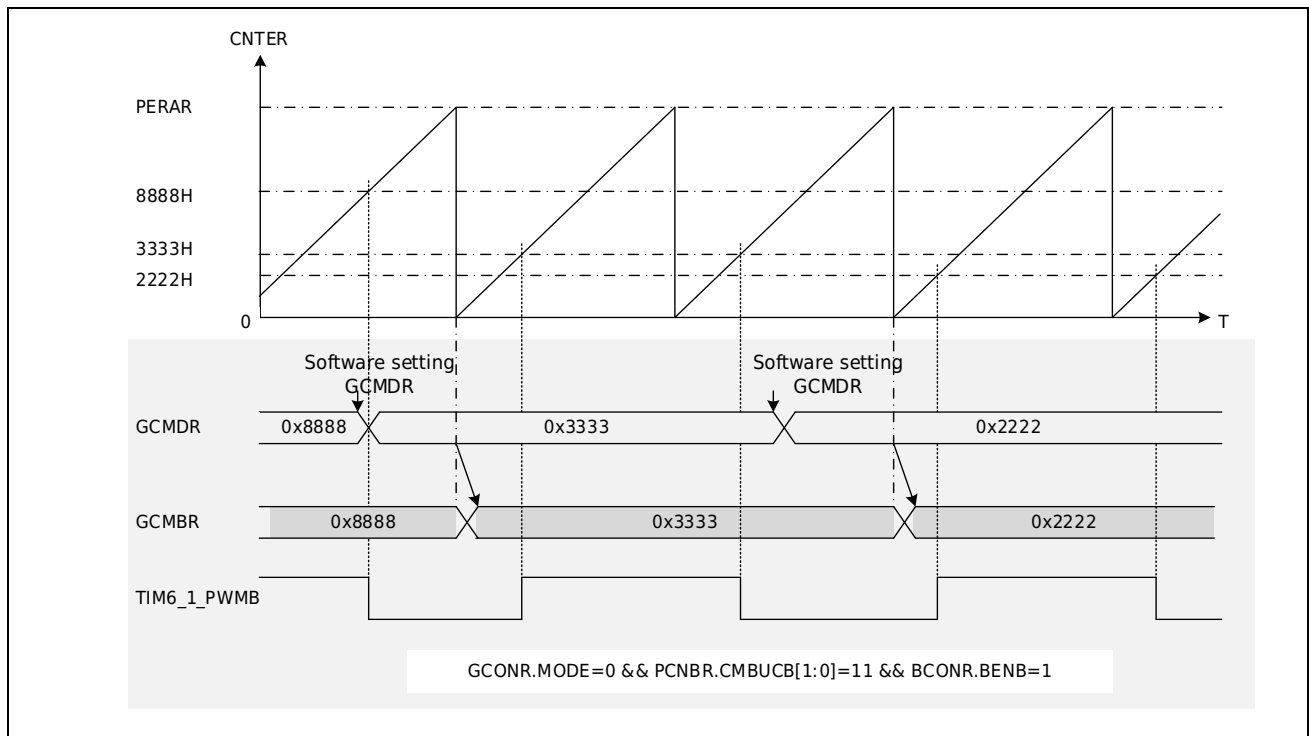


Figure 20-13 Counting Buffer Action in Sawtooth Wave Mode

Note:

- In sawtooth counting mode, if a reset action occurs, it is also regarded as a count overflow. Each benchflag 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 general period reference value and the General 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<P><SPA><SPB>=DCONR.DTBENU<D>=1) && comparison output count (PCNAR.CAPMDA=0) && triangle wave counting mode (GCONR.MODE=1), the cache transfer time point of the General period reference value, General comparison reference value, dedicated comparison reference value, and dead zone reference value is determined by the corresponding register control bits. When BCONR.BTRUA<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<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 20-14, it is a schematic diagram of the cache action when the triangle wave counts to the valley point , As shown in the Figure 20-15, it is a schematic diagram of the cache action occurring when the triangle wave counts to the peak point and the valley point.

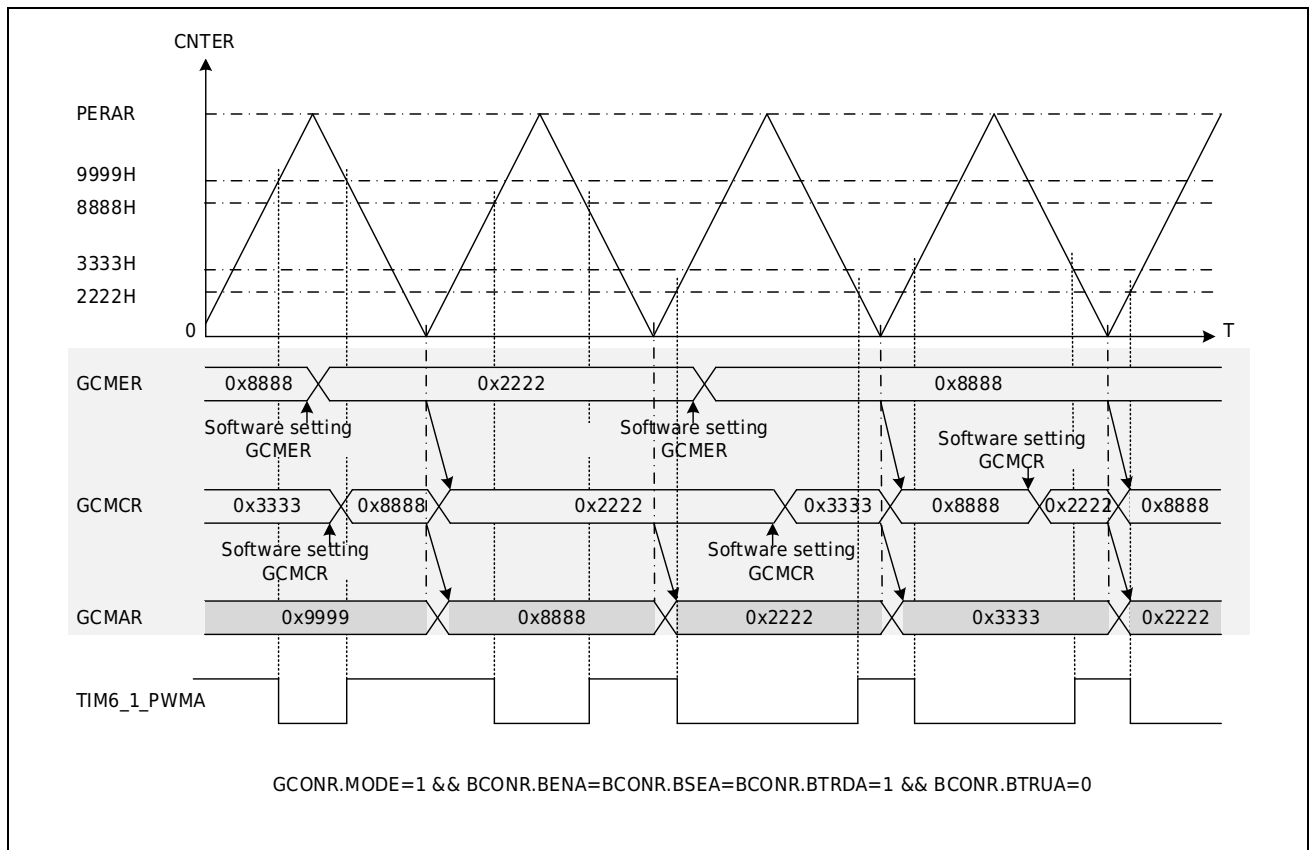


Figure 20-14 Count Buffer Action in Triangle Wave Mode

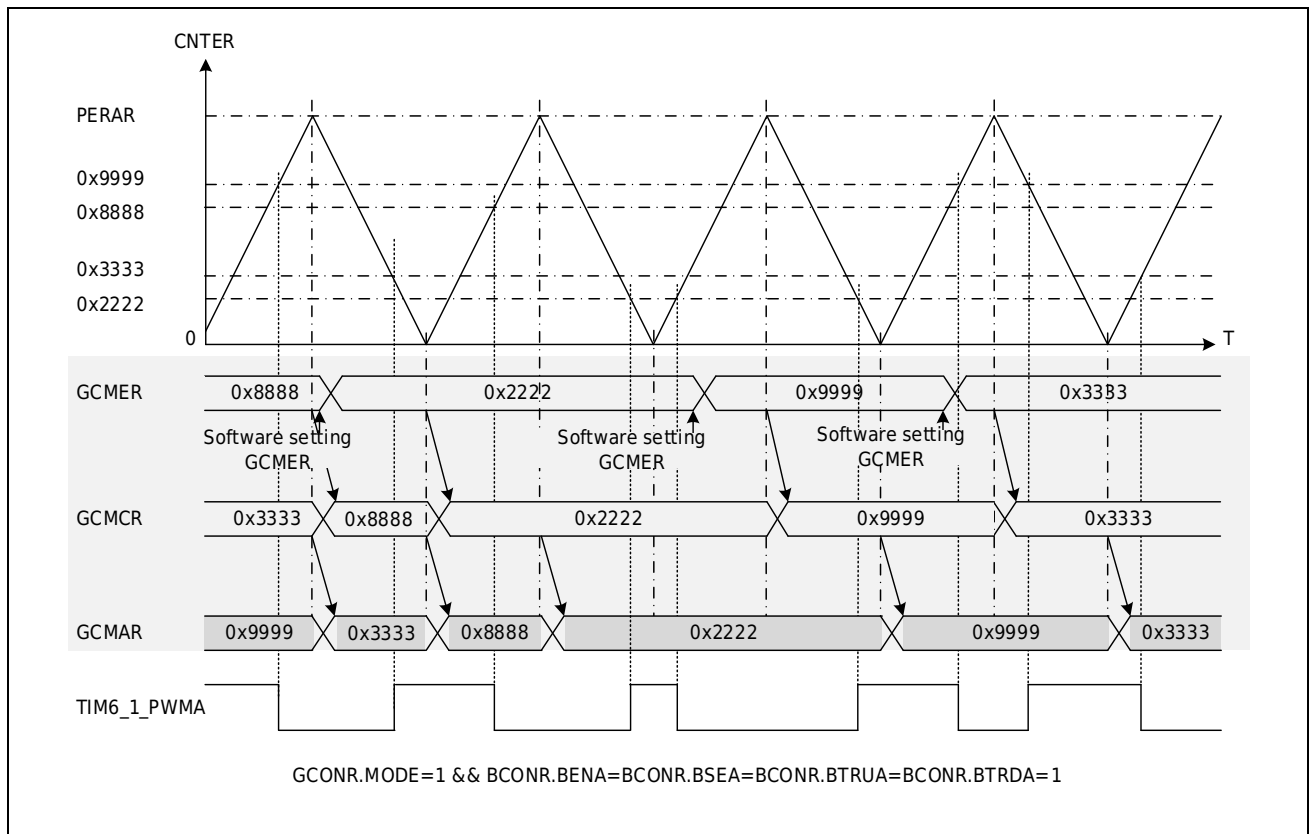


Figure 20-15 Count Buffer Action in Triangle Wave Mode

Capture Input Buffer Transfer

When the capture input action ($PCNAR.CAPMDA=1$) valid, the common comparison reference value supports the cache function. If the buffer setting valid ($BCONR.BENA=1$), 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 $BCONR.BSEA$).

20.3.12 Digital Filtering

TIM6_<t>_PWMA, TIM6_<t>_PWMB, TIM6_TRIGA~D port inputs all have digital filtering function. TIM6_<t>_PWMA and TIM6_<t>_PWMB enable the filtering function of the corresponding port by setting the relevant enable bits of the filtering control register (FCNCR). The filtering reference clock when the filtering is effective can also be set through the filtering control register (FCNCR). TIM6_TRIGA~D ports are ports shared by a group of units. The filtering function of the corresponding port is enable 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 20-16.

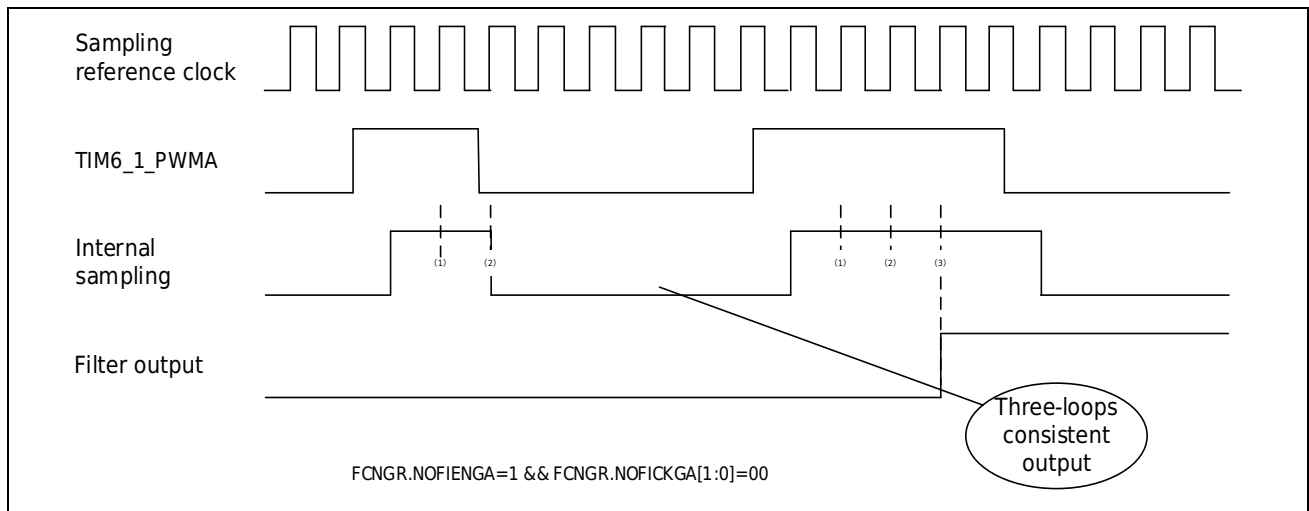


Figure 20-16 Filtering Function of Capture Input Port

20.3.13 General PWM Output

20.3.13.1 Single-edge Aligned Independent PWM Output

In sawtooth counting mode (GCONR.MODE=0), the two ports TIM6_<t>_PWMA and TIM6_<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 FigureFigure 20-17below.

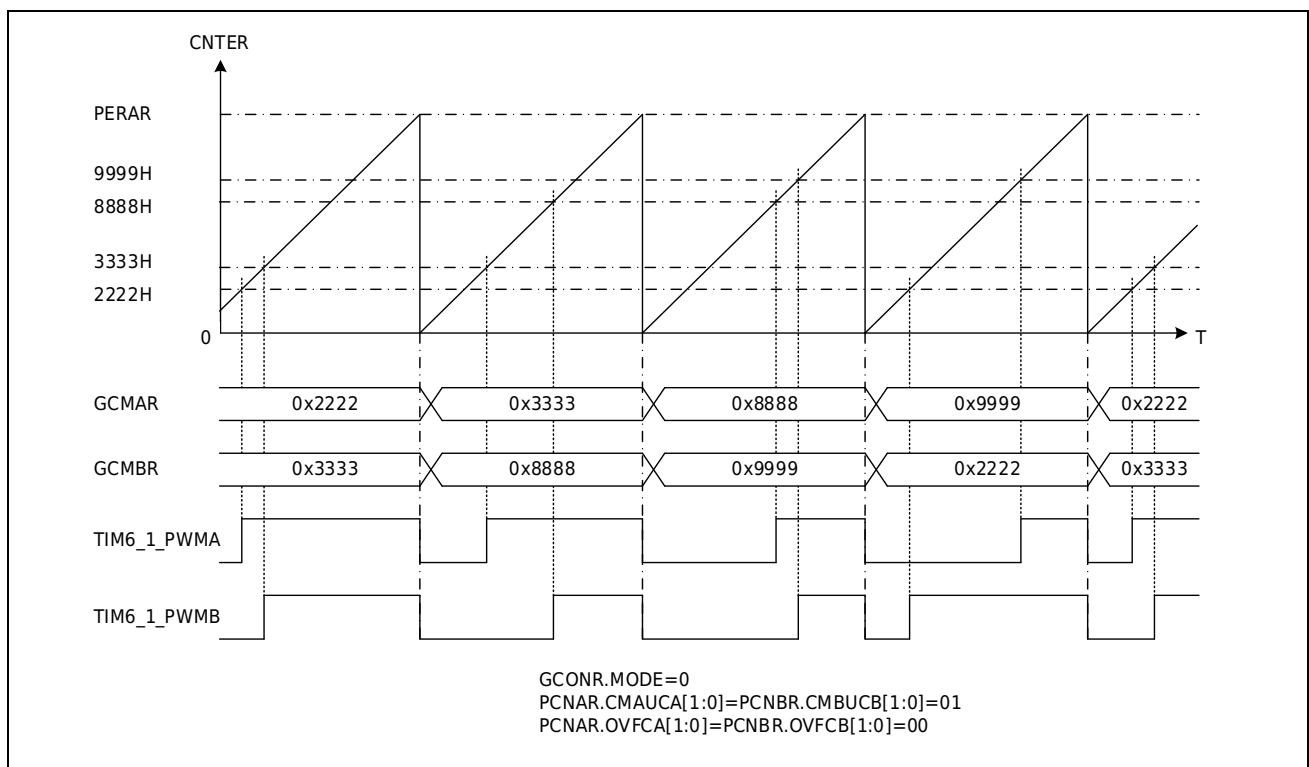


Figure 20-17 Single-edge Aligned Independent PWM

20.3.13.2 Symmetrical Independent PWM Output on Both Sides

In triangle wave counting mode (GCONR.MODE=1), the two ports TIM6_<t>_PWMA and TIM6_<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 the figure, the TIM6_<t>_PWMA port and the TIM6_<t>_PWMB port implement bilateral symmetrical independent output PWM.

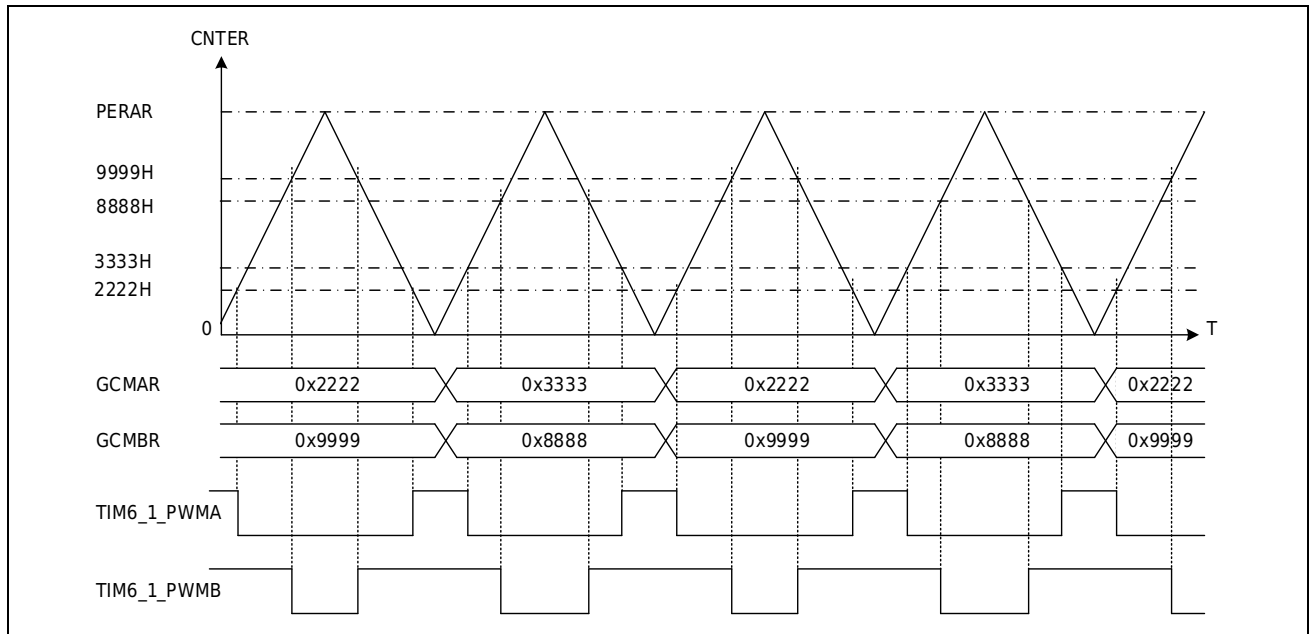


Figure 20-18 Bilateral Symmetrical Independent PWM

20.3.13.3 Bilateral Symmetrical Complementary PWM Output

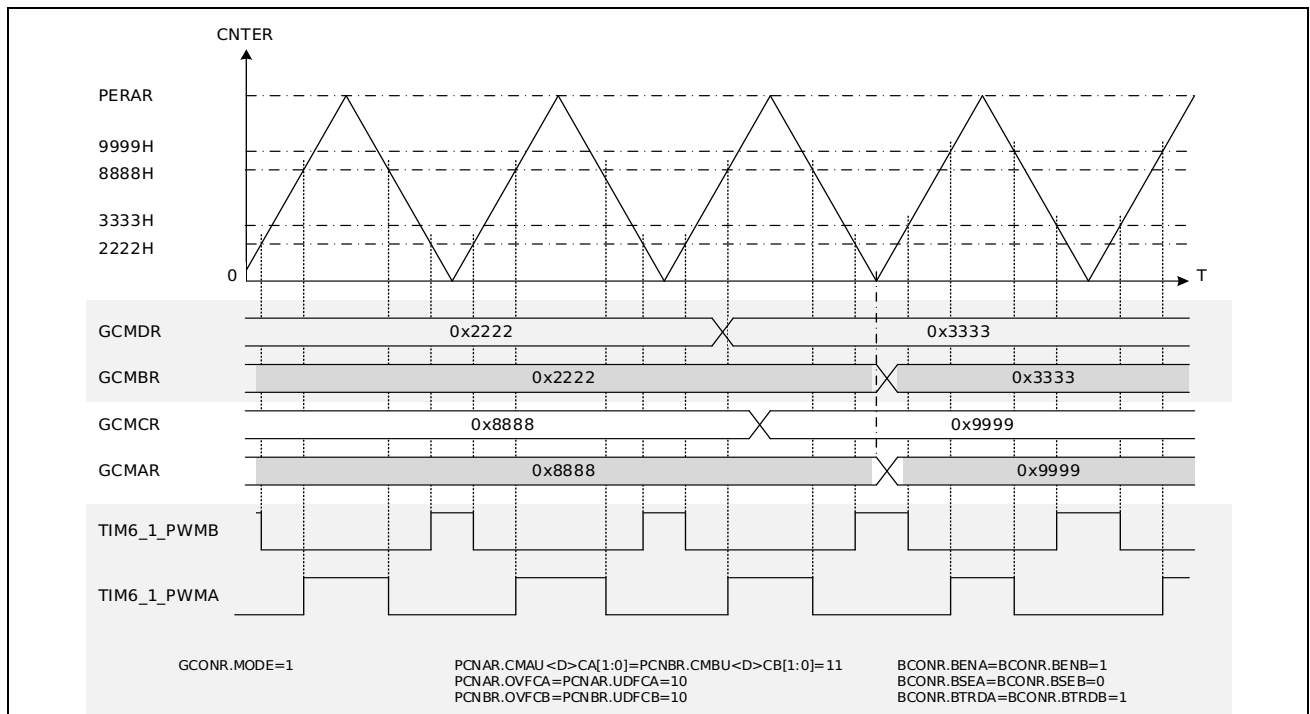
In the triangle wave counting mode (GCONR.MODE=1), by setting the level state of the port at count start, compare match, and count overflow, a pair of complementary PWM waveforms can be output on the TIM6_<t>_PWMA and TIM6_<t>_PWMB ports.

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 general comparison reference value register (GCMBR) used for the waveform output of the TIM6_<t>_PWMB port is directly written by the CPU, etc., and has no direct relationship with the value of GCMAR.

Figure 20-19 is an instance of software setting the output of GCMBR complementary PWM wave.



Hardware Setting GCMBR Complementary PWM Output

The hardware setting method means that in the triangle wave mode, the value of the general comparison reference value register (GCMBR) used for the waveform output of the TIM6_<t>_PWMB port is determined by the value operation of the General Comparison Reference Value Register (GCMAR) and the dead zone reference value register (DTU<D>AR).

Figure 20-20 is an instance of setting GCMBR complementary PWM wave output for hardware.

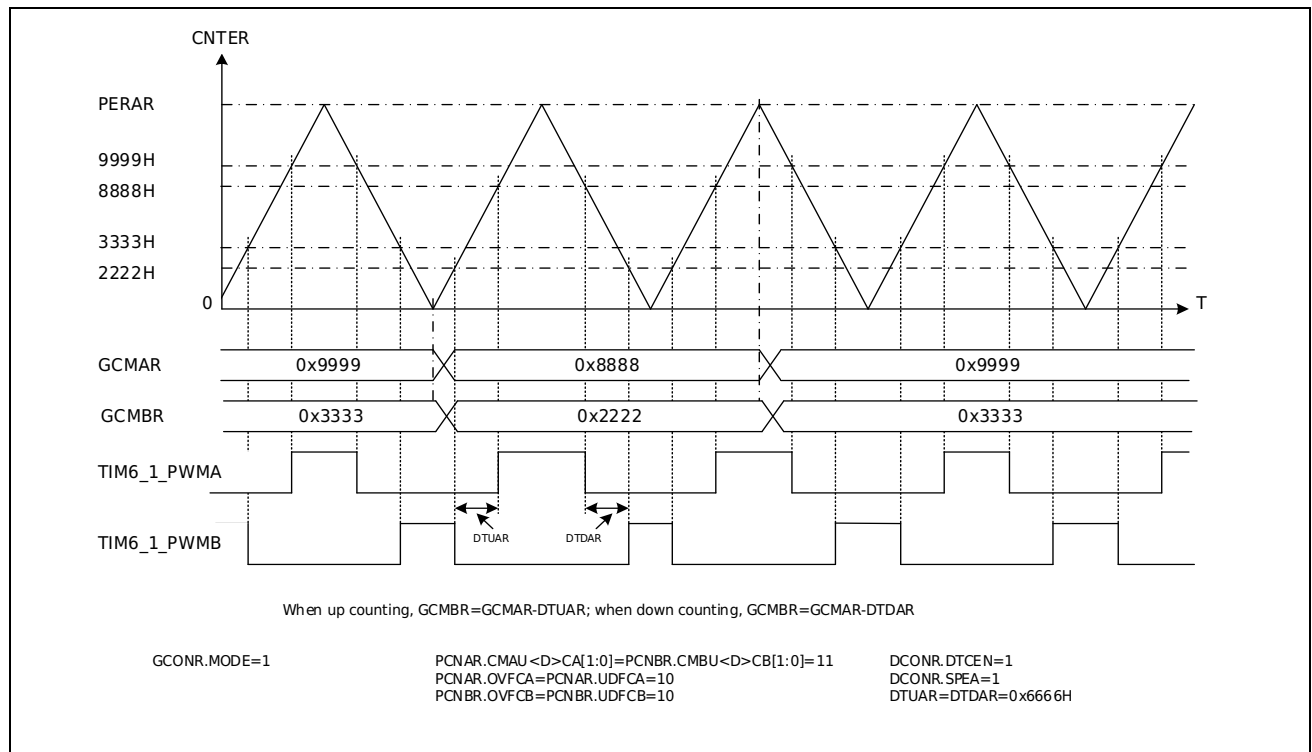


Figure 20-20 Hardware Setting GCMBR Complementary PWM Wave Output

20.3.13.4 Double-sided Asymmetric PWM Output

In the triangle wave counting mode (GCONR.MODE=1), the two ports TIM6_<t>_PWMA and TIM6_<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 general comparison reference value register (GCMAR, GCMBR). If the port level change of TIM6_<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 the Figure 20-21, it is the bilateral asymmetric PWM output of the TIM6_<t>_PWMA port.

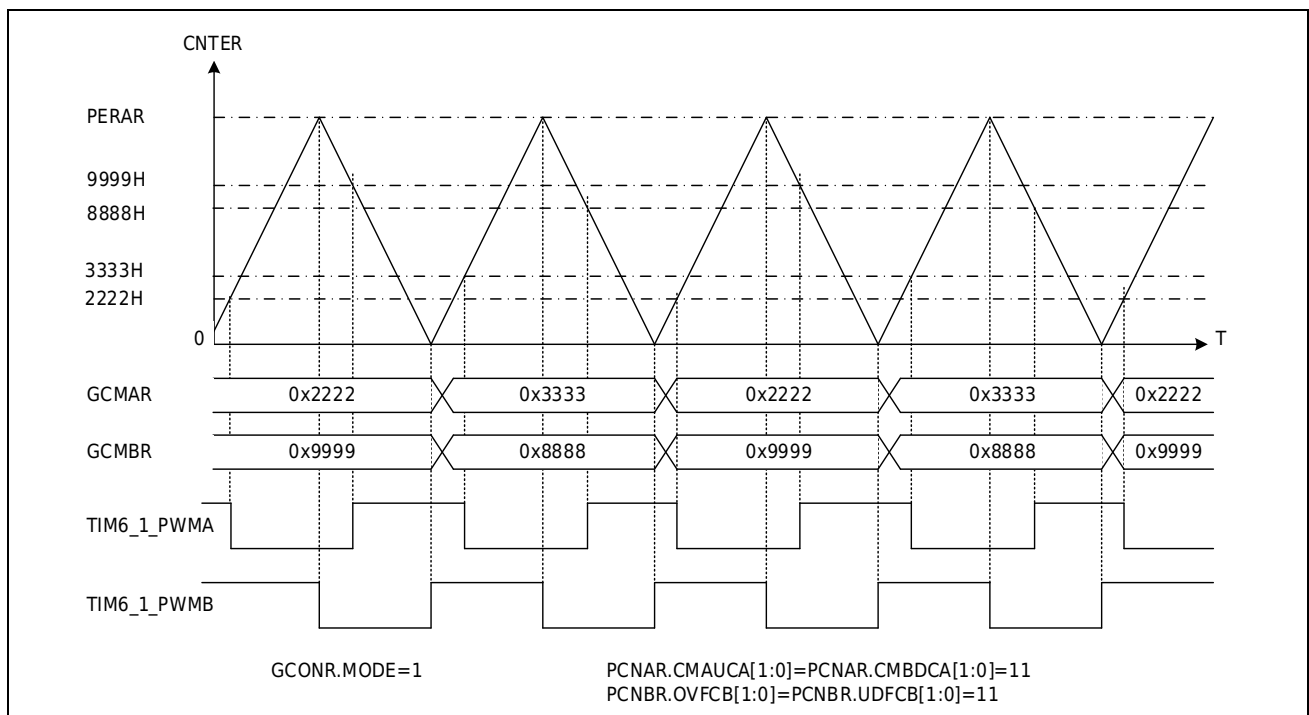


Figure 20-21 Double-sided Asymmetric PWM Output

20.3.13.5 Multi-phase PWM Output between Units

TIM6_<t>_PWMA and TIM6_<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 the Figure 20-22, unit 1 , unit 2 , and unit 3 are combined to output 6- phase single-side aligned independent PWM; As shown in the Figure 20-23, unit 1 , unit 2 , and unit 3 are combined to output 3 -phase double-side symmetrical complementary PWM.

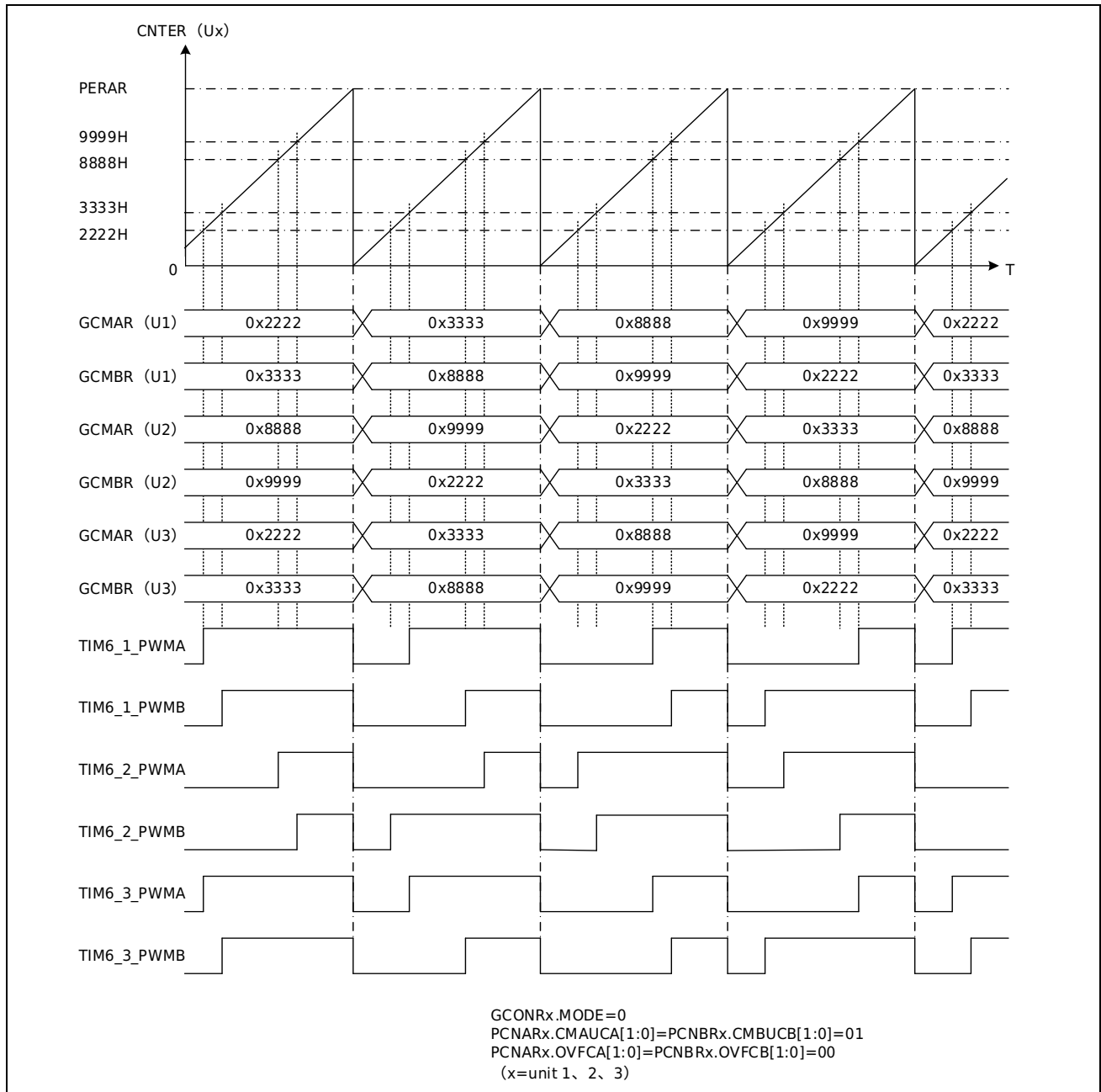


Figure 20-22 Phase Single Edge Aligned Independent PWM

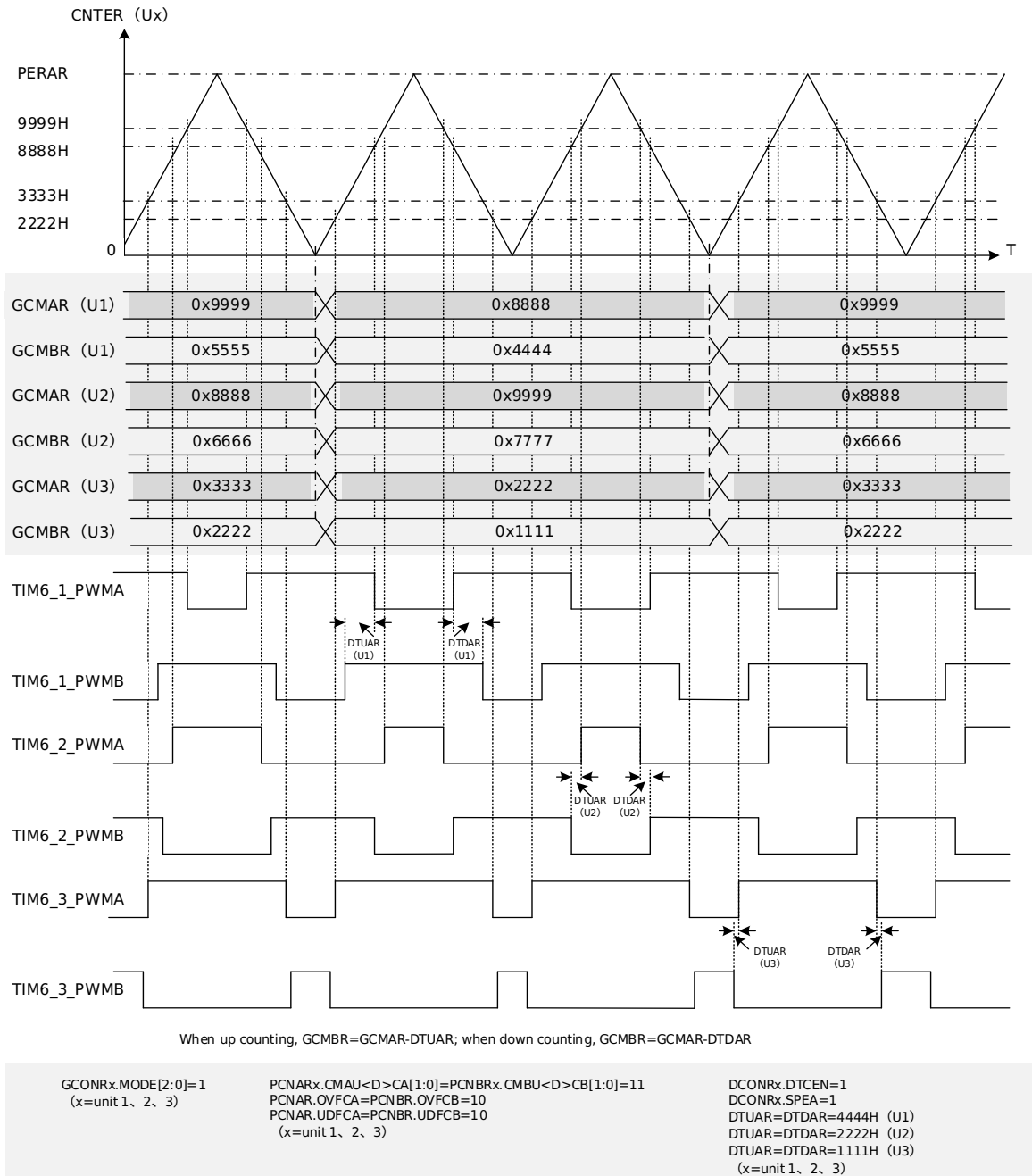


Figure 20-23 3-phase Bilateral Symmetrical Complementary PWM with Dead zone

20.3.14 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 periods, 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 periods the request signal is valid for once. In other periods, 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 20-24 Shown is an instance of the operation of the periodic interval valid request signal.

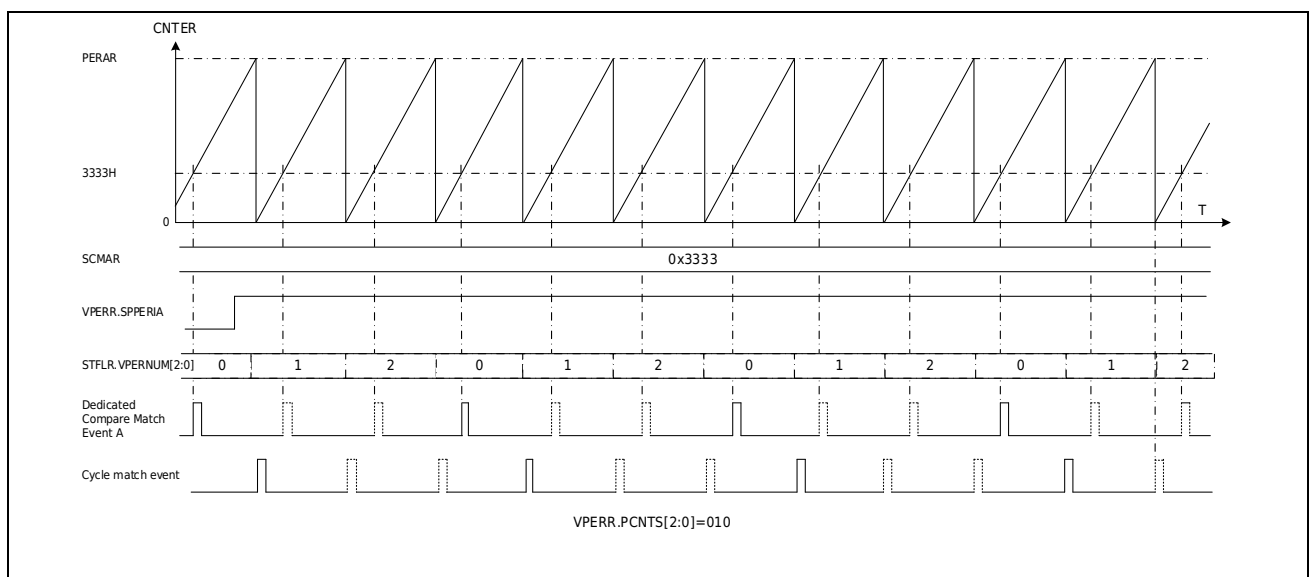


Figure 20-24 Cycle Interval Valid Request Signal Action

20.3.15 Orthogonal Coding Count

By considering the TIM6_<t>_PWMA input as AIN input, the TIM6_<t>_PWMB input as BIN input, and any one of the TIM6_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 10 units of Timer6 can be arbitrarily combined to select position counting units or revolution counting units, but the combination is fixed when the Z- phase Masking function is valid. Units 1 and 2 are combined, unit 1 is used as a position counting unit, and unit 2 is used as a revolution counting unit to realize position counting and revolution counting respectively; units 3 and 4 are combined, unit 3 is used as a position counting unit, and unit 4 is used as a revolution counting unit to realize position counting and revolution counting respectively; units 5 and 6 are combined, unit 5 is used as a position counting unit, and unit 6 is used as a revolution counting unit to realize position counting and revolution counting respectively; units 7 and 8 are combined, unit 7 is used as a position counting unit, and unit 8 is used as a revolution counting unit; units 9 and 10 are combined, unit 9 is used as a position counting unit, and unit 10 is used as a 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 TIM6_<t>_PWMA and TIM6_<t>_PWMB in the hardware increment event selection register (HCUPR) and hardware decrement event selection register (HCDOR); the input action of ZIN Clear the position timer of the position counting unit by setting the hardware clearing event selection register (HCLRR) of the position unit, and realize the revolution timer counting of the revolution counting unit by setting the hardware increment event selection register (HCUPR) of the revolution counting unit.

20.3.15.1 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 20-25 below.

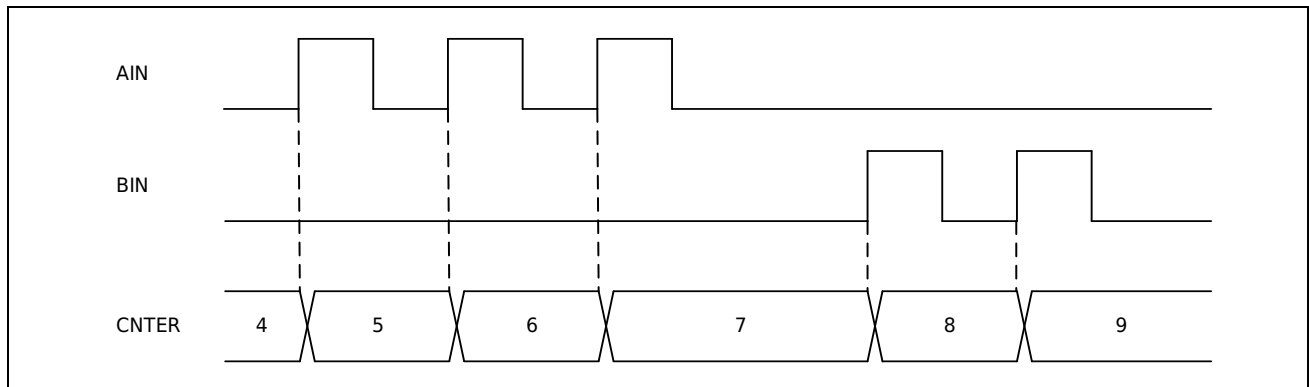


Figure 20-25 Position Mode - Basic Counting

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 Figure 20-26~Figure 20-28 below

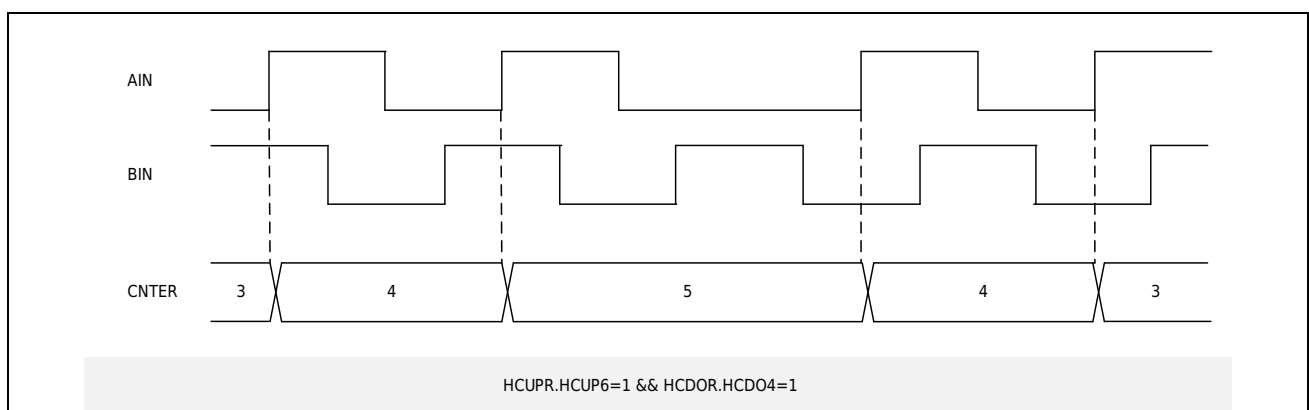


Figure 20-26 Position Counting Mode - Phase Difference Counting (1 Time Counting)

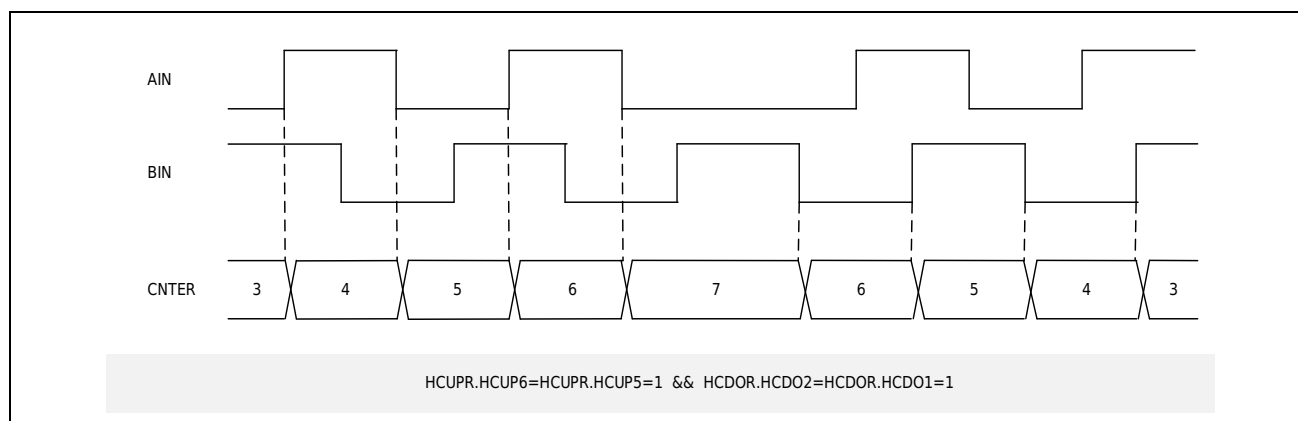


Figure 20-27 Position Counting Mode - Phase Difference Counting (2 Times Counting)

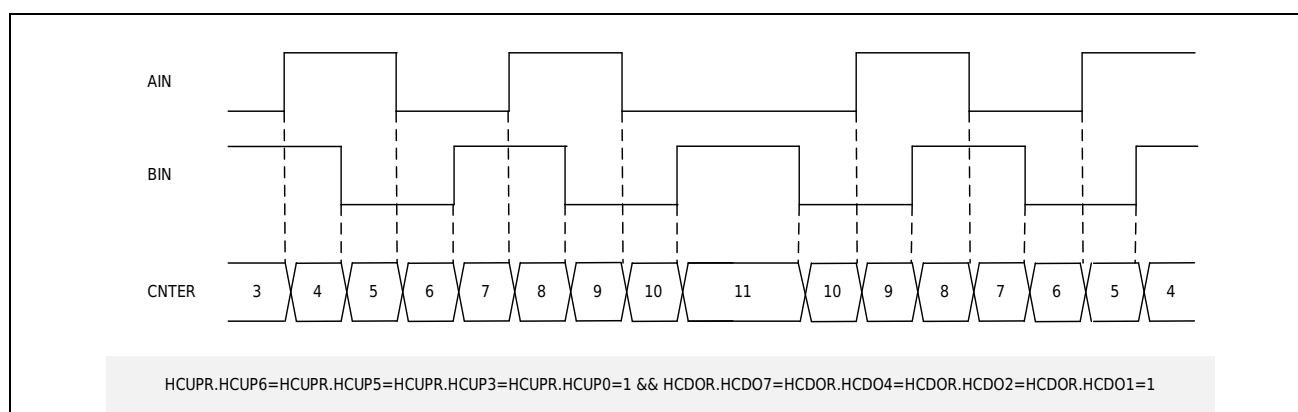


Figure 20-28 Position Counting Mode - Phase Difference Counting (4 times Counting)

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 figureFigure 20-29below.

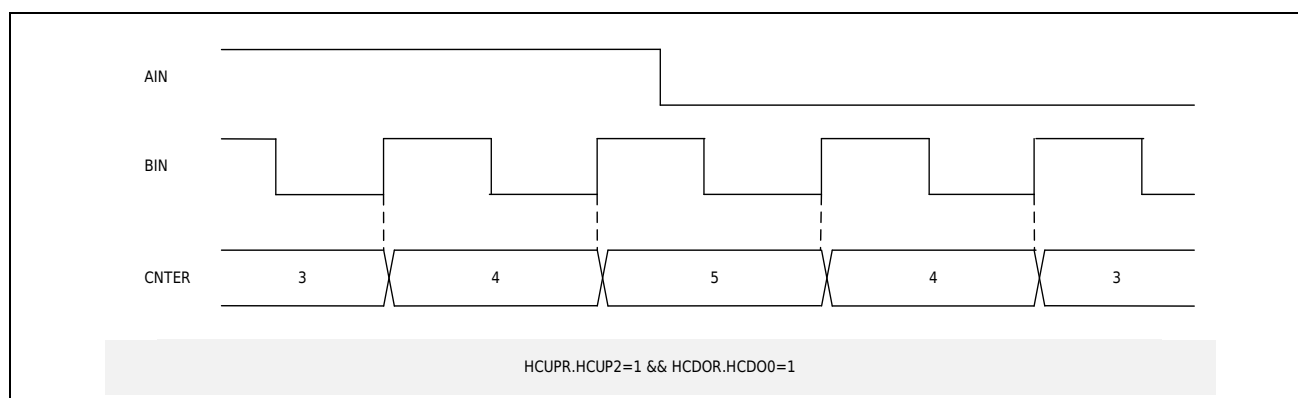


Figure 20-29 Position Counting Mode - Direction Counting

20.3.15.2 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 20-30 below.

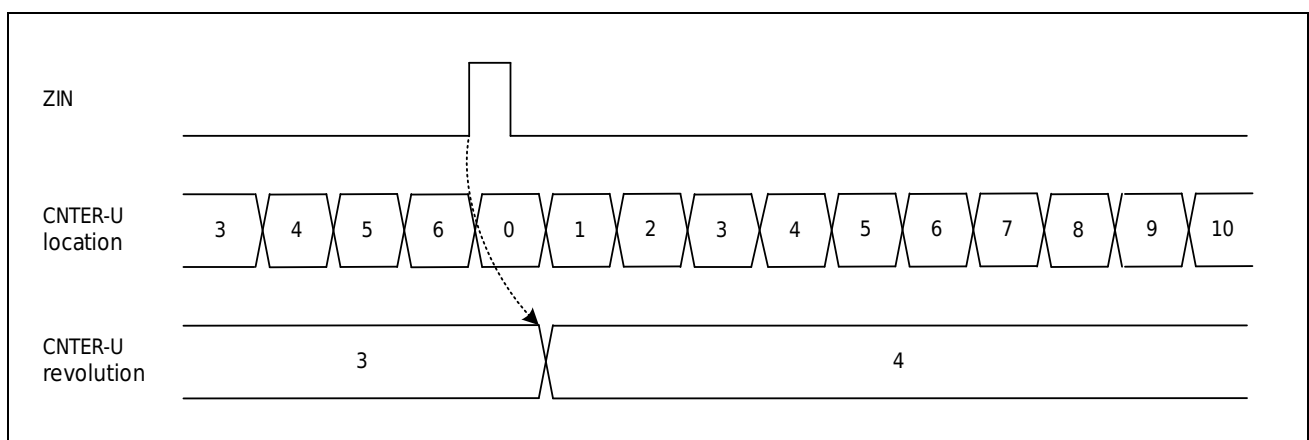


Figure 20-30 Revolution Counting Mode - Z phase Counting

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_TRGSEL0~3) to the overflow or underflow event of the position counting unit. For specific event numbers, refer to the INTC chapter. As shown in Figure 20-31 below.

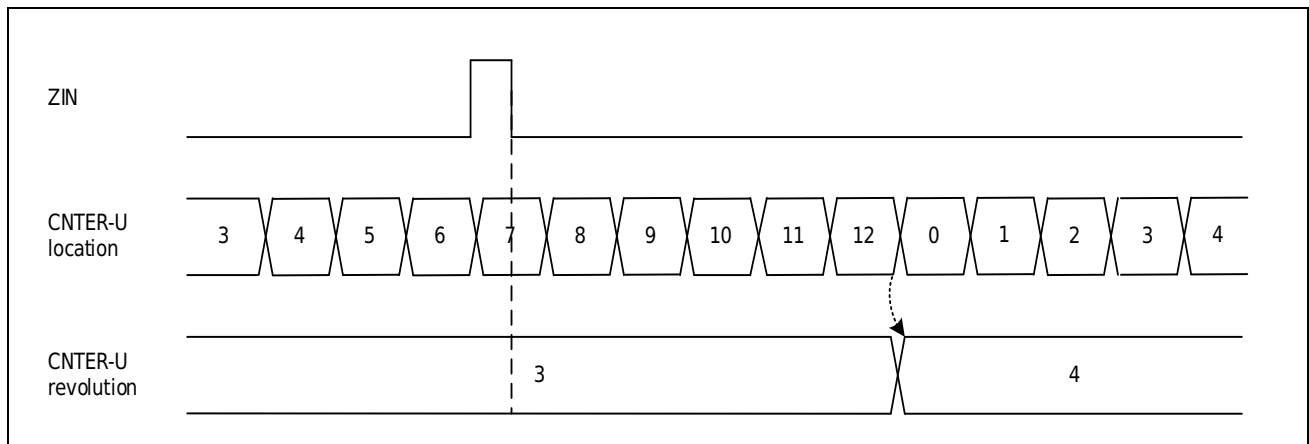


Figure 20-31 Revolution Counting Mode - Position Overflow Counting

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 FigureFigure 20-32below.

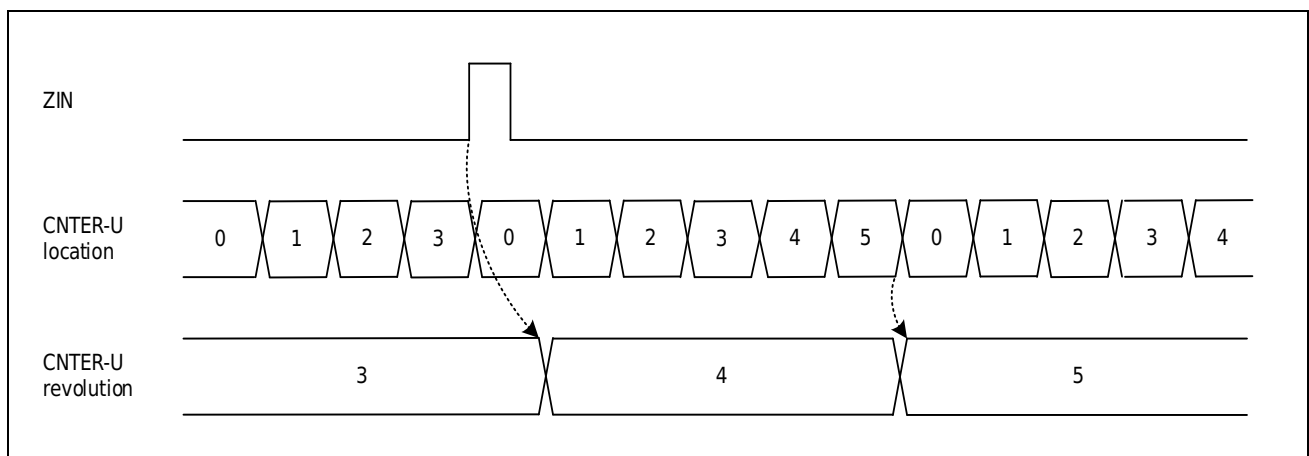


Figure 20-32 Revolution Counting Mode - Mixed Counting

20.3.15.3 Z Phase Motion Masking

In the Z-phase counting function or mixed counting function of the revolution counting mode, it can be set within a few periods 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 general control register (GCONR) of the position counting unit is 1, the Z-phase masking function of the position counting unit is enabled, and the number of periods of Z-phase masking is set by GCONR. ZMSKVAL; the General Control register of the revolution counting unit (When GCONR.ZMSKREV of GCONR) is 1, the Z-phase masking function of the revolution counting unit is enabled.

When Figure 20-33 is the mixed counting of revolution counting mode, if there is ZIN phase input within 4 counting cycles after the position counting unit overflows, the action of ZIN phase input is

invalid, that is, the revolution counting unit does not count and the position counting unit is not cleared; the subsequent ZIN phase input acts normally.

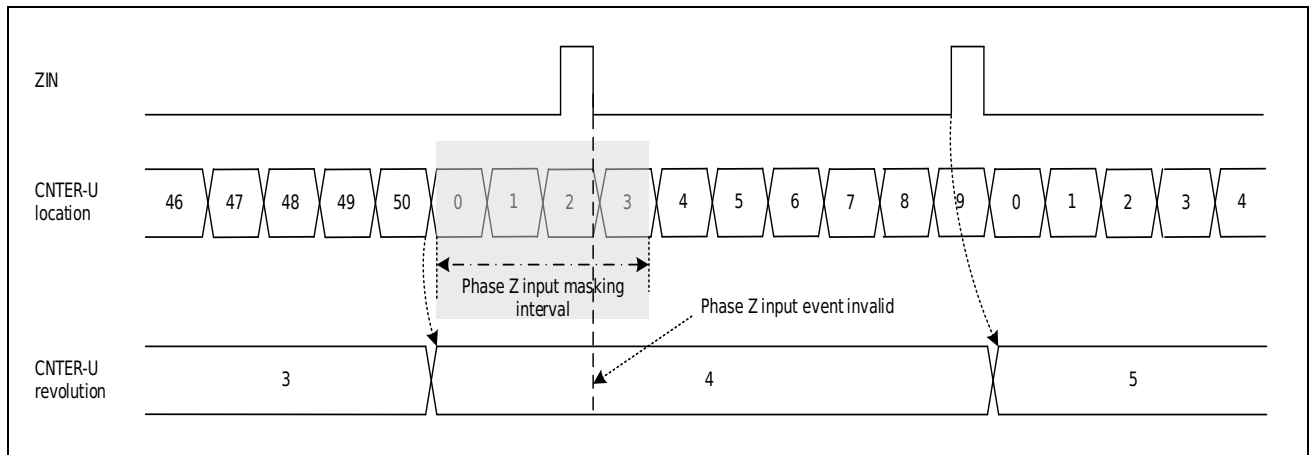


Figure 20-33 Revolution Counting Mode - Mixed Counting Z-phase Masking Action Instance 1

When Figure 20-34 is the mixed counting of revolution counting mode, the counting direction changes in the third cycle after the position counting unit overflows. At this time, the set 4-cycle masking cycle becomes invalid (the actual ZIN phase masking function is maintained for 3 cycles), and starts counting down. After the counting underflow occurs in the position counting unit, the ZIN phase masking 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.

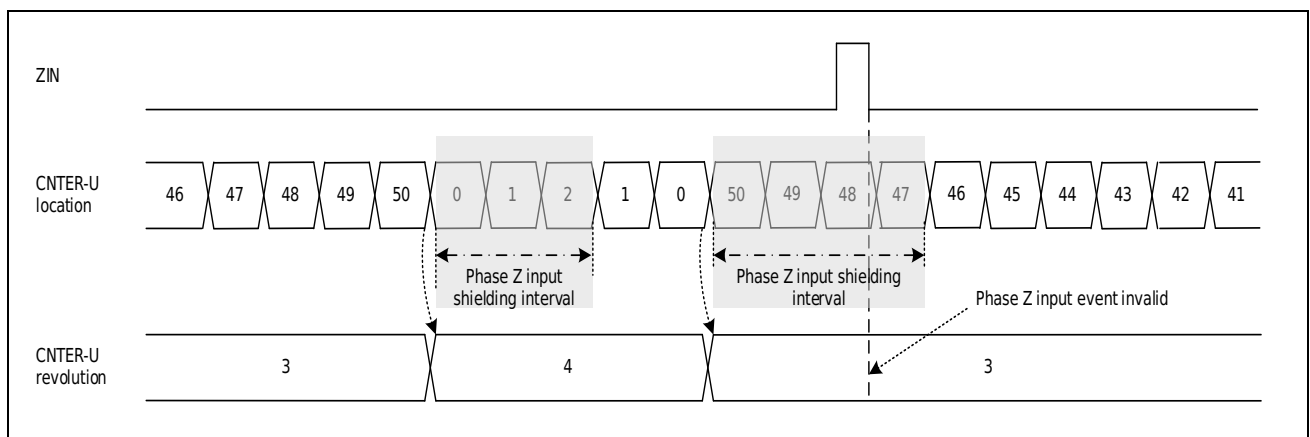


Figure 20-34 Revolution Counting Mode - Mixed Counting Z-phase Masking Action Instance 2

20.3.16 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 6 common port output control interfaces. Each unit selects the EMB event connection interface to be used by setting the port control register (PCNA.EMBSA). 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 general 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 General PWM output port can change its state to output high impedance, output low level or output high level (determined by the setting of PCNA.R.EMBCA). For example, if PCNAR.EMBCA=01 is set, then during the normal output of the TIM6_<t>_PWMA port, if an EMB event occurs, the output on the TIM6_<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.R.EMBRA=00; this method is called One Shot release) or restore the normal PWM output after counting to the next overflow point (PCNA.R.EMBRA=01, 10, 11; this method is called Cycle By Cycle release) through the setting of the port control register.

20.3.17 Typical Instances

The following describes the basic settings of Timer6 related registers in several typical application situations for user reference.

20.3.17.1 Basic Counting and Interrupt Actions

- a) Setting the general period reference value (PERAR);
- b) Set the required comparison reference value, including general comparison reference value (GCMAR~GCMFR), special comparison reference value (SCMAR~SCMBR), etc.
- c) 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.
- d) Set internal count clock frequency division (GCONR.CKDIV [3:0])
- e) Set waveform mode (GCONR.MODE)
- f) Set the counting direction (only required when sawtooth wave mode GCONR.MODE=0);
- g) Start counter (GCONR.START=1)

20.3.17.2 Comparison Output and Interrupt Action

- a) Setting the Universal Period Reference Value (PERAR)
- b) Set the comparison reference value of each channel, including general comparison reference value A (GCMAR), general comparison reference value B (GCMBR)
- c) Set the required interrupt enable bit, including count overflow interrupt (ICONR.INTENOVF), count underflow interrupt (ICONR.INTENUDF), count match interrupt (ICONR.INTENA~B), etc.
- d) Set the port output status of each channel in different counting states (refer to the related

control of bit17~bit0 of PCNAR or PCNBR);

- e) Set internal count clock frequency division (GCONR.CKDIV [3:0])
- f) Set waveform mode (GCONR.MODE)
- g) Set the counting direction (only required when sawtooth wave mode GCONR.MODE=0);
- h) Set the comparison output mode of each channel (PCNAR.CAPMDA=0, PCNBR.CAPMDB=0);
- i) Set each channel output enable (PCNAR.OUTENA=1, PCNBR.OUTENB=1)
- j) Start counter (GCONR.START=1)

20.3.17.3 Capture Input and Interrupt Actions

- a) Setting the general period reference value (PERAR);
- b) Set the required interrupt enable bit, including count overflow interrupt (ICONR.INTENOVF), count underflow interrupt (ICONR.INTENUDF), capture input interrupt (ICONR.INTENA~B), etc.
- c) 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)
- d) Set internal count clock frequency division (GCONR.CKDIV [3:0])
- e) Set waveform mode (GCONR.MODE)
- f) Set the counting direction (only required when sawtooth wave mode GCONR.MODE=0);
- g) Set capture input mode (PCNAR.CAPMDA=1, PCNBR.CAPMDB=1);
- h) Start counter (GCONR.START=1)
- i) 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

20.3.17.4 Buffer Transfer Action (Period Reference Value)

- a) Set desired common period reference values (PERAR, PERBR, PERCR)
- b) Set single and double buffer transfer mode (BCONR.BSEP)
- c) 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);
- d) Set internal count clock frequency division (GCONR.CKDIV [3:0])
- e) Set waveform mode (GCONR.MODE)
- f) Set the counting direction (only required when sawtooth wave mode GCONR.MODE=0);
- g) Set the cache function to be valid (BCONR.BENP=1)
- h) Start counter (GCONR.START=1)
- i) 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))

20.3.17.5 Cache Transfer Action (Common Benchmark Value)

- a) Set the desired common comparison base value (GCMAR, GCMCR, GCMER, GCMBR, GCMDR, GCMFR)
- b) Set the single and double buffer transmission mode of each channel (BCONR.BSEA, BCONR.BSEB)
- c) 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);
- d) Set internal count clock frequency division (GCONR.CKDIV [3:0])
- e) Set waveform mode (GCONR.MODE)
- f) Set the counting direction (only required when sawtooth wave mode GCONR.MODE=0);
- g) Set the buffer function of each channel to be valid (BCONR.BENA=1, BCONR.BENB=1)
- h) Start counter (GCONR.START=1)
- i) 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))

20.3.17.6 Cache Transmission Action (Specific Benchmark Value)

- a) Set the required specific comparison reference value (SCMAR, SCMCR, SCMER, SCMBR, SCMDR, SCMFR)
- b) Set the single and double buffer transmission mode of each channel (BCONR.BSESPA, BCONR.BSESPB)
- c) 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)
- d) Set internal count clock frequency division (GCONR.CKDIV [3:0])
- e) Set waveform mode (GCONR.MODE)
- f) Set the counting direction (only required when sawtooth wave mode GCONR.MODE=0);
- g) Set the buffer function of each channel to be valid (BCONR.BENSPA=1, BCONR.BENSPB=1)
- h) Start counter (GCONR.START=1)
- i) 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))

20.3.17.7 Buffer Transfer Action (Deadband Reference Value)

- a) Set the desired dead zone reference (DTUAR, DTUBR, DTDAR, DTDBR)
- b) 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)
- c) Set internal count clock frequency division (GCONR.CKDIV [3:0])
- d) Set waveform mode (GCONR.MODE)
- e) Set the counting direction (only required when sawtooth wave mode GCONR.MODE=0);
- f) Set the cache function to be valid (DCONR.DTBENU=1, DCONR.DTBEND=1)
- g) Set the hardware dead zone function to be valid (DCONR.DTCEN=1)
- h) Start counter (GCONR.START=1)
- i) Wait for the corresponding cache transmission time point, and the cache action occurs (DTUBR->DTUAR, DTDBR->DTDAR)

20.3.17.8 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) Start the counter synchronously (set the corresponding bit of the SSTAR register to 1, and each unit corresponds to a register bit)

20.3.17.9 Synchronous Start Action (Hardware Method)

- a) Setting the general period reference value (PERAR);
- b) Set the required comparison reference value, including general comparison reference value (GCMAR~GCMFR), special comparison reference value (SCMAR~SCMBR), etc.
- c) 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.
- d) Set the hardware startup conditions (selected by HSTAR.HSTAx, x=8~31);
- e) Set hardware startup enable (HSTAR.STAS=1);
- f) Repeat steps a~e 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)
- g) Wait for the set trigger event to occur, and confirm that the counters of each unit start synchronously

20.3.17.10 Quadrature Encoder Counting Action (2 Phases)

- a) Setting the general period reference value (PERAR);
- b) Set the required comparison reference value, including general comparison reference value

(GCMAR~GCMFR), special comparison reference value (SCMAR~SCMBR), etc.

- c) 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.
- d) Set the required hardware count-up condition (selected by HCUPR.HCUPx, x=0~7)
- e) Set the required hardware down counting conditions (selected by HCDOR.HCDOx, x=0~7)
- f) Start counter (GCONR.START=1)
- g) Wait for the set quadrature encoding counting event to be generated, and confirm that the counter counts normally

20.3.17.11 Quadrature Encoder Counting Action (3 Phases)

- a) Refer to steps a~e in the chapter [Quadrature Encoder Counting Action (2 Phase)] to set the position counting unit
- b) Set the hardware clearing condition of the position counting unit (selected by HCLRR.HCLR_x, x=16~31)
- c) Set position counter unit hardware clear enable (HCLRR.CLEAR_S=1);
- d) Set the general period reference value (PERAR) of the revolution counter unit
- e) Set the comparison reference value of revolution counting unit, including general comparison reference value (GCMAR~GCMFR), special comparison reference value (SCMAR~SCMBR), etc.
- f) Set the interrupt enable bits required by the revolution counting unit, including count overflow interrupt (ICONR.INTENOVF), count underflow interrupt (ICONR.INTENUDF), count match interrupt (ICONR.INTENA~F, ICONR.INTENSAU, ICONR.INTENSAD , ICONR.INTENSBU, ICONR.INTENSBD), etc.
- g) Set the hardware up counting condition 1 of the revolution counting unit (ZIN phase input) (selected by HCUPR.HCUP_x, x=16~31, the set event here should be consistent with the event set by the position counting unit in step b)
- h) 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)
- i) 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)
- j) Set the trigger source number in TMR6_TRGSEL0 to the count overflow event of the position counting unit (the overflow event number refers to the INTC chapter)
- k) Set the trigger source number in TMR6_TRGSEL1 to the counting underflow event of the position counting unit (for the underflow event number, refer to the INTC chapter)
- l) Start revolution counting unit counter (GCONR.START=1)

- m) Start position counting unit counter (GCONR.START=1)
- n) Wait for the set AIN, BIN, ZIN phase count event to occur, and confirm that the counter counts normally

20.3.17.12 Single PWM Output

- a) Refer to the setting of steps a~j in chapter [comparison output and interrupt action] (the output states of the two PWM channels TIM6_<t>_PWMA and TIM6_<t>_PWMB inside each unit can be set independently to form two mutual unrelated single PWM output)

20.3.17.13 Complementary PWM Output (Software Dead Zone)

- a) Setting the general period reference value (PERAR);
- b) Set general comparison reference value A (GCMAR) and general comparison reference value B (GCMBR)
- c) Set the required interrupt enable bit, including count overflow interrupt (ICONR.INTENOVF), count underflow interrupt (ICONR.INTENUDF), count match interrupt (ICONR.INTENA~B), etc.
- d) 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)
- e) Set internal count clock frequency division (GCONR.CKDIV [3:0])
- f) Set the waveform mode to triangle wave mode (GCONR.MODE=1);
- g) Set the comparison output mode (PCNAR.CAPMDA=0, PCNBR.CAPMDB=0);
- h) Set output enable (PCNAR.OUTENA=1, PCNBR.OUTENB=1)
- i) Start counter (GCONR.START=1)

20.3.17.14 Complementary PWM Output (Hardware Dead Zone)

- a) Setting the general period reference value (PERAR);
- b) Set general comparison reference value A (GCMAR), dead zone reference value (DTUAR, DTDAR)
- c) 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) wait
- d) 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)
- e) Set internal count clock frequency division (GCONR.CKDIV [3:0])
- f) Set the waveform mode to triangle wave mode (GCONR.MODE=1);
- g) Set the comparison output mode of each channel (PCNAR.CAPMDA=0, PCNBR.CAPMDB=0);
- h) Set each channel output enable (PCNAR.OUTENA=1, PCNBR.OUTENB=1)

- i) Set the hardware dead zone function to be valid (DCONR.DTCEN=1)
- j) Start counter (GCONR.START=1)

20.3.17.15 EMB Monitoring and Interrupt Action

- a) Refer to steps a~h in chapter [Complementary PWM output (software dead zone)] or steps a~i in chapter [Complementary PWM output (hardware dead zone)] to set the complementary PWM output action
- b) 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)
- c) Set the time point when the PWM port resumes normal output when the EMB event becomes invalid (PCNAR.EMBRA, PCNBR.EMBRB)
- d) Set and select the EMB event source input from the EMB module (PCNAR.EMBSA, PCNBR.EMBSB)
- e) Set the relevant registers of the EMB module (including EMB interrupt disable 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 (GCONR.START=1), the EMB module monitors the system status in real time

20.3.18 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 Table 20-3 below.

Table 20-3 Comparison Table 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	PCNAR.STACA
		When stopped		Support	Support	PCNAR.STPCA
		When overflowing		Support	Support	PCNAR.OVFCA
		When underflow		Support	Support	PCNAR.UDFCA
		When counting matches (Up Counting)		Support	Support	PCNAR.CMAUCA
		When counting matches (Down Counting)		Support	Support	PCNAR.CMADCA
	Cache Transmission	Periodic Base Value	Single cache	Support	Support	PERBR->PERAR
			Double buffer	Support	Support	PERCR->PERBR PERBR->PERAR
		Comparison Reference Value	Single cache	Support	Support	GCMDR->GCMBR GCMCR->GCMAR
			Double buffer	Support	Support	GCMFR->GCMDR GCMER->GCMCR GCMDR->GCMBR GCMCR->GCMAR

PWM Output Function				Sawtooth Wave	Triangle Wave	Related Main Registers
	EMB			Support	Support	PCNAR.EMBCA
Complementary PWM Output	Port Status Control	At startup		Support	Support	PCNAR.STACA
		When stopped		Support	Support	PCNAR.STPCA
		When overflowing		Support	Support	PCNAR.OVFCA
		When underflow		Support	Support	PCNAR.UDFCA
		When counting matches (Up Counting)		Support	Support	PCNAR.CMAUCA
		When counting matches (Down Counting)		Support	Support	PCNAR.CMADCA
	Cache Transmission	Periodic Base Value	Single cache	Support	Support	PERBR->PERAR
			Double buffer	Support	Support	PERCR->PERBR PERBR->PERAR
		Comparison Reference Value	Single cache	Support	Support	GCMDBR->GCMDBR GCMCR->GCMAR
			Double buffer	Support	Support	GCMFR->GCMDBR GCMER->GCMCR GCMDBR->GCMDBR GCMCR->GCMAR
		Dead Zone Reference Value	Single cache	Support	Support	DTUBR->DTUAR DTDBR->DTDAR
	PWM output without dead zone			Support	Support	GCMAR=GCMDBR
	PWM output with dead zone		Software method	Support	Support	GCMAR≠GCMDBR
			Hardware method	Not support	Support	GCMDBR=GCMAR-DTUAR GCNBR=GCMAR-DTDAR
	EMB			Support	Support	PCNAR.EMBCA

20.4 Interrupt and Event Description

20.4.1 Interrupt Output

Timer6 contains 6 general count compare match interrupts (including 2 capture input interrupts), 2 dedicated count compare match interrupts, 2 count cycle match interrupts, and 1 dead zone error interrupt.

20.4.1.1 Count Comparison Match Interrupt

There are six general 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>_GCMAR~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.

20.4.1.2 Count Period Match Interrupt

When the sawtooth wave counts up to the overflow point, the sawtooth wave counts down to the underflow point, the triangle wave counts to the valley point or the triangle wave counts to the peak point, the STFLR. 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 period match interrupt (TMR6_<t>_GOVF and TMR6_<t>_GUDF) can be triggered at the corresponding time point.

20.4.1.3 Dead Zone Error Interrupt

When the value of the dead zone reference register (DTU<D>AR) is loaded into the general comparison reference register (GCMBR), if it exceeds the period limit, a dead zone 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 zone error interrupt (TMR6_<t>_GDTE) will be triggered at this moment.

20.4.2 Event Output

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 general 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 instance of the general 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.

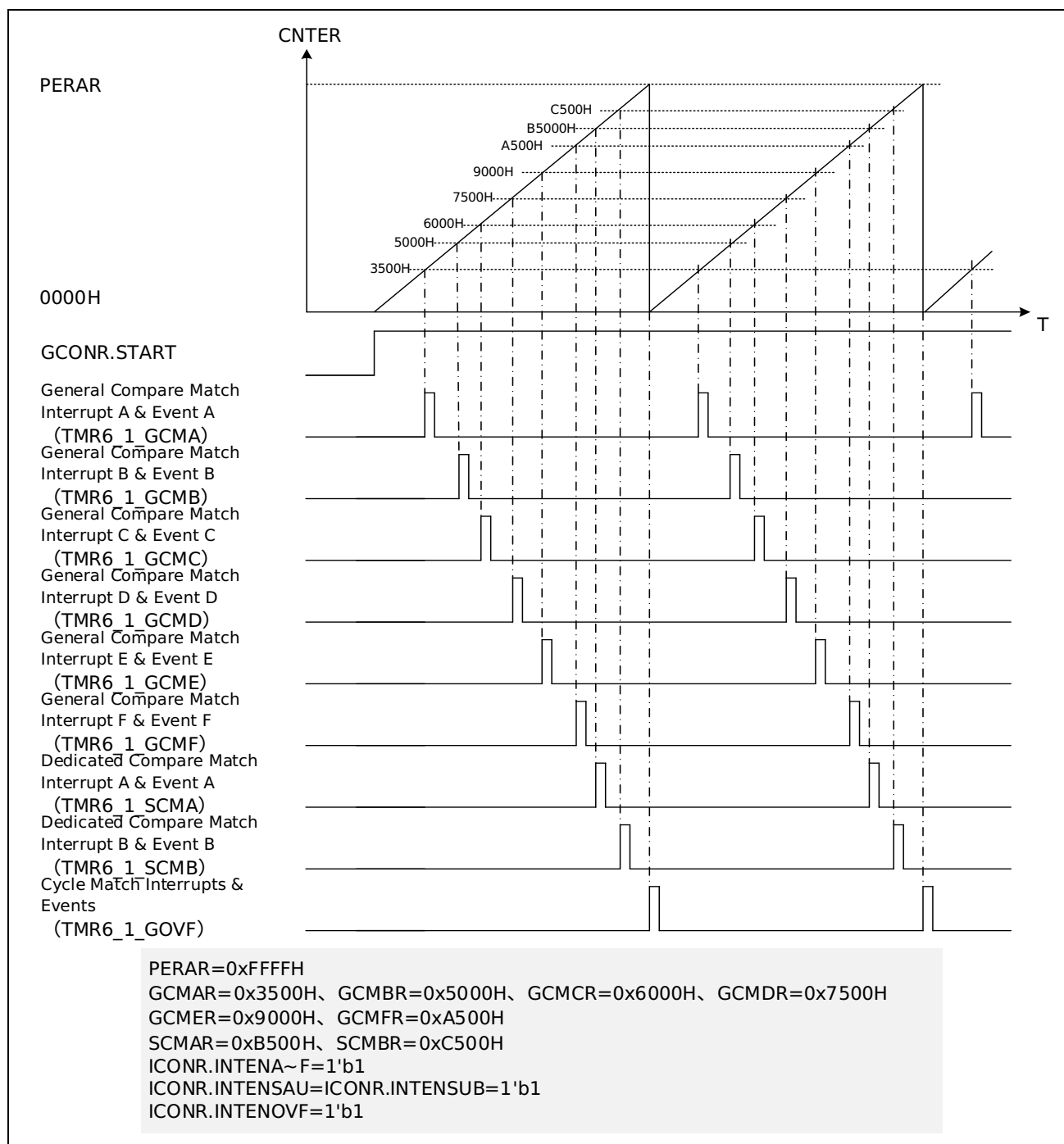


Figure 20-35 Instance of Interrupt & Event Output in Sawtooth Wave Mode

20.5 Register Description

Table 20-4 shows the register list of the Timer6 module.

Table 20-4 Timer6 Register List

BASE ADDR:

0x4003C000(U1), 0x4003C400(U2), 0x4003C800(U3), 0x4003CC00(U4)

0x4003D000(U5), 0x4003D400(U6), 0x4003D800(U7), 0x4003DC00(U8)

0x4003E000(U9), 0x4003E400(U10)

Register Name	Symbol	Offset	Bit Width	Reset Value
General count register	TMR6_CNTER	0x0000	32	0x0000 0000
General refresh value register	TMR6_UPDAR	0x0004	32	0x0000 0000
General period reference value register A	TMR6_PERAR	0x0040	32	0x0000 FFFF
General period reference value register B	TMR6_PERBR	0x0044	32	0x0000 FFFF
General period reference value register C	TMR6_PERCR	0x0048	32	0x0000 FFFF
General comparison reference value register A	TMR6_GCMAR	0x0080	32	0x0000 FFFF
General comparison reference value register B	TMR6_GCMBR	0x0084	32	0x0000 FFFF
General comparison reference value register C	TMR6_GCMCR	0x0088	32	0x0000 FFFF
General comparison reference value register D	TMR6_GCMDR	0x008C	32	0x0000 FFFF
General comparison reference value register E	TMR6_GCMER	0x0090	32	0x0000 FFFF
General comparison reference value register F	TMR6_GCMFR	0x0094	32	0x0000 FFFF
Dedicated comparison reference value register A	TMR6_SCMAR	0x00C0	32	0x0000 FFFF
Dedicated comparison reference value register B	TMR6_SCMBR	0x00C4	32	0x0000 FFFF
Dedicated comparison reference value register C	TMR6_SCMCR	0x00C8	32	0x0000 FFFF
Dedicated comparison reference value register D	TMR6_SCMDR	0x00CC	32	0x0000 FFFF
Dedicated comparison reference value register E	TMR6_SCMER	0x00D0	32	0x0000 FFFF
Dedicated comparison reference value register F	TMR6_SCMFR	0x00D4	32	0x0000 FFFF
Dead zone reference value register UA	TMR6_DTUAR	0x0100	32	0x0000 FFFF
Dead zone reference value register DA	TMR6_DTDAR	0x0104	32	0x0000 FFFF
Dead zone reference value register UB	TMR6_DTUBR	0x0108	32	0x0000 FFFF
Dead zone reference value register DB	TMR6_DTDBR	0x010C	32	0x0000 FFFF
General control register	TMR6_GCONR	0x0140	32	0x0000 0002
Interrupt control register	TMR6_ICONR	0x0144	32	0x0000 0000
Cache control register	TMR6_BCONR	0x0148	32	0x0000 0000

Register Name	Symbol	Offset	Bit Width	Reset Value
Dead zone control register	TMR6_DCONR	0x014C	32	0x0000 0000
Port control register A	TMR6_PCNAR	0x0154	32	0x0000 0000
Port control register B	TMR6_PCNBR	0x0158	32	0x0000 0000
Filter control register G	TMR6_FCNGR	0x015C	32	0x0000 0000
Valid period register	TMR6_VPERR	0x0160	32	0x0000 0000
Status flag register	TMR6_STFLR	0x0164	32	0x8000 0000
Hardware boot event selection register	TMR6_HSTAR	0x0180	32	0x0000 0000
Hardware stop event selection register	TMR6_HSTPR	0x0184	32	0x0000 0000
Hardware clear event selection register	TMR6_HCLRR	0x0188	32	0x0000 0000
Hardware refresh event selection register	TMR6_HUPDR	0x018C	32	0x0000 0000
Hardware capture event selection register A	TMR6_HCPAR	0x0190	32	0x0000 0000
Hardware capture event selection register B	TMR6_HCPBR	0x0194	32	0x0000 0000
Hardware increment event selection register	TMR6_HCUPR	0x0198	32	0x0000 0000
Hardware decrememt event selection register	TMR6_HCDOR	0x019C	32	0x0000 0000
Filtering control register T	TMR6_FCNTNTR	U1_base+0x03EC	32	0x0000 0000
Software synchronous start control register	TMR6_SSTAR	U1_base+0x03F0	32	0x0000 0000
Software synchronization stop control register	TMR6_SSTPR	U1_base+0x03F4	32	0x0000 0000
Software synchronization clear control register	TMR6_SCLRR	U1_base+0x03F8	32	0x0000 0000
Software synchronization refresh control register	TMR6_SUPDR	U1_base+0x03FC	32	0x0000 0000

Note:

- The trigger selection registers (TMR6_TRGSEL0~3) are 4 unit-independent registers shared by 10 units of Timer6.
- The software synchronization registers (TMR6_SSTAR, TMR6_SSTPR, TMR6_SCLRR, TMR6_SUPDR) are 4 unit-independent registers shared by 10 units of Timer6.
- The filter control register (TMR6_FCNTNTR) is a unit-independent register shared by all 10 units of Timer6.

20.5.1 General Counter Value Register (TMR6_CNTER)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CNT [15:0]															
Bit		Flag		Bit Name		Function		Read and write							
b31~b16		Reserved		-		Read as "0"		R							
b15~b0		CNT [15:0]		Count value		Current timer count		R/W							

20.5.2 General Refresh Value Register (TMR6_UPDAR)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
UPDA [15:0]															
Bit		Flag		Bit Name		Function		Read and write							
b31~b16		Reserved		-		Read as "0"		R							
b15~b0		UPDA [15:0]		Refresh value		Set the refresh value to be updated into the timer		R/W							

20.5.3 General Period Reference Value Register (TMR6_PERmR) (m=A~C)

Reset value: 0x0000FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PERA-C [15:0]															
Bit		Flag		Bit Name		Function		Read and write							
b31~b16		Reserved		-		Read as "0"		R							
b15~b0		PERA-C [15:0]		Count period value		Set the count period value and corresponding cache value of each round of counting		R/W							

20.5.4 General Comparison Reference Value Register (T Mr6_G cm Mr) (no = A~F)

Reset value: 0x0000FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
GCMA-F [15:0]															
Bit	Flag	Bit Name		Function		Read and write									
b31~b16	Reserved	-		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		R/W									

20.5.5 Dedicated Comparison Reference Value Register (T Mr6_SCM Mr) (No = A~F)

Reset value: 0x0000FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SCMA-F [15:0]															
Bit	Flag	Bit Name		Function		Read and write									
b31~b16	Reserved	-		Read as "0"		R									
b15~b0	SCMA-F [15:0]	Dedicated baseline value		Set comparison reference value and cache value		R/W									

20.5.6 Dead zone Reference Value Register (TMR6_DTmnR)(m = D, U &&n = A, B)

Reset value: 0x0000FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
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	Flag	Bit Name		Function		Read and write									
b31~b16	Reserved	-		Read as "0"		R									
b15~b0	DTU/DA-B [15:0]	Dead zone value		Dead zone setting value and cache value		R/W									

20.5.7 General Control Register (TMR6_GCONR)

Reset value: 0x00000002

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved												ZMSK VAL [1:0]	ZMSK POS	ZMSK REV	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved							OV STP	CKDIV [3:0]				-	MODE	DIR	START
Bit		Flag		Bit Name		Function								Read and write	
b31~b20		Reserved		-		Read as "0", write as "0"								R/W	
b19~b18		ZMSKVAL [1:0]		Phase Z input muting period number		Quadrature encoding phase Z input masked count period value 00: Phase Z input masking function is invalid 01: The Z-phase input is masked within 4 counting periods after the position count overflows or underflows 10: The Z-phase input is masked within 8 counting periods after the position count overflows or underflows 11: The Z-phase input is masked within 16 counting periods after the position count overflows or underflows								R/W	
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								R/W	
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 masking 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 masking period								R/W	
b15~b9		Reserved		-		Read as "0", write as "0"								R/W	
b8		OVSTP		Count overflow stop control		0: Counter continues counting after overflow or underflow 1: Counter stops counting after overflowing or underflowing								R/W	
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								R/W	
b3		Reserved		-		Read as "0", write as "0"								R/W	
b2		MODE		Counting mode		0: Sawtooth mode 1: Triangular wave mode								R/W	
b1		DIR		Counting direction		0: count down 1: count up								R/W	
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								R/W	

20.5.8 Interrupt Control Register (TMR6_ICONR)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved												INTEN SBD	INTEN SBU	INTEN SAD	INTEN SAU
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved							INTEN DTE	INTEN UDF	INTEN OVF	INTEN F	INTEN E	INTEN D	INTEN C	INTEN B	INTEN A
Bit	Flag	Bit Name	Function											Read and write	
b31~b20	Reserved	-	Read as "0", write as "0"											R/W	
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											R/W	
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											R/W	
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											R/W	
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											R/W	
b15~b9	Reserved	-	Read as "0", write as "0"											R/W	
b8	INTENDTE	Dead zone error interrupt enable	0: When the dead zone is wrong, the interrupt is invalid 1: When the dead zone is wrong, the interrupt is enabled											R/W	
b7	INTENUDF	Underflow interrupt enable	0: When the counter underflows, the interrupt is invalid. 1: When the counter underflows, the interrupt is enabled											R/W	
b6	INTENOVF	Overflow interrupt enable	0: When the counter overflows, the interrupt is invalid. 1: When the counter overflows, the interrupt is enabled											R/W	
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											R/W	
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											R/W	
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											R/W	
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											R/W	
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											R/W	
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											R/W	

20.5.9 Cache Control Register (TMR6_BCONR)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved								BTRD SPB	BTRU SPB	BSE SPB	BEN SPB	BTRD SPA	BTRU SPA	BSE SPA	BEN SPA
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	BTRD P	BTRU P	BSE P	BEN P	BTRD B	BTRU B	BSE B	BEN B	BTRD A	BTRU A	BSE A	BEN A

Bit	Flag	Bit Name	Function	Read and write
b31~b24	Reserved	-	Read as "0", write as "0"	R/W
b23	BTRDSPB	Dedicated comparison reference value buffer transmission 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	R/W
b22	BTRUSPB	Dedicated comparison reference value buffer transmission 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	R/W
b21	BSESPB	Dedicated baseline value buffer transmission option B	0: Single buffer transfer (SCMDR->SCMBR) 1: Double buffer transfer (SCMFR->SCMDR->SCMBR)	R/W
b20	BENSPB	Dedicated baseline cache transfer B	0: Buffer transmission invalid 1: Buffer transmission enable	R/W
b19	BTRDSPA	Dedicated comparison reference value buffer transmission 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	R/W
b18	BTRUSPA	Dedicated comparison reference value buffer transmission 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	R/W
b17	BSESPA	Dedicated baseline value buffer transmission option A	0: single buffer transfer (SCMCR->SCMAR) 1: Double buffer transfer (SCMER->SCMCR->SCMAR)	R/W
b16	BENSPA	Dedicated baseline cache transfer A	0: Buffer transmission is invalid 1: Buffer transfer enable	R/W
b15~b12	Reserved	-	Read as "0", write as "0"	R/W
b11	BTRDP	Cycle value buffer transmission 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	R/W
b10	BTRUP	Cycle value buffer transmission 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	R/W
b9	BSEP	Period value buffer transmission selection	0: Single buffer transmission (PERBR->PERAR) 1: Double buffer transmission (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	R/W
b8	BENP	Period value buffer transfer	0: Buffer transmission is invalid 1: Buffer transfer enable	R/W
b7	BTRDB	General comparison value buffer transmission 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	R/W
b6	BTRUB	General comparison value buffer transmission 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	R/W

b5	BSEB	General comparison value buffer transmission option B	When comparing output functions: 0: Single buffer transmission (GCMDR->GCMBR) 1: Double buffer transmission (GCMFR->GCMDR->GCMBR) When capturing the input function: 0: Single buffer transmission (GCMBR->GCMDR) 1: Double buffer transmission (GCMBR->GCMDR->GCMFR)	R/W
b4	BENB	General comparison value buffer transmission B	0: Buffer transmission invalid 1: Buffer transmission enable	R/W
b3	BTRDA	General comparison value buffer transmission 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	R/W
b2	BTRUA	General comparison value buffer transmission 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	R/W
b1	BSEA	General comparison value buffer transmission option A	When comparing output functions: 0: Single buffer transmission (GCMCR->GCMAR) 1: Double buffer transmission (GCMER->GCMCR->GCMAR) When capturing the input function: 0: Single buffer transmission (GCMAR->GCMCR) 1: Double buffer transmission (GCMAR->GCMCR->GCMER)	R/W
b0	BENA	General comparison value buffer transmission A	0: Buffer transmission invalid 1: Buffer transmission enable	R/W

20.5.10 Dead Zone Control Register (TMR6_DCONR)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								DTB TRD	DTB TRU	DTB END	DTB ENU	-	-	SEPA	DTC EN
Bit	Flag	Bit Name	Function											Read and write	
b31~b8	Reserved	-	Read as "0", write as "0"											R/W	
b7	DTBTRD	Dead zone value buffer transmission 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											R/W	
b6	DTBTRU	Dead zone value buffer transmission 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											R/W	
b5	DTBEND	Dead zone value cache transmission D	0: Buffer transmission invalid 1: Buffer transmission enable (DTDBR->DTDAR)											R/W	
b4	DTBENU	Dead zone value cache transfer U	0: Buffer transmission invalid 1: Buffer transmission enable (DTUBR->DTUAR)											R/W	
b3~b2	Reserved	-	Read as "0", write as "0"											R/W	
b1	SEPA	Separate settings	0: DTUAR and DTDAR are set separately 1: The value of DTDAR is automatically equal to the value of DTUAR											R/W	
b0	DTCEN	Dead zone function	0: Dead zone function is invalid 1: The dead zone function is valid											R/W	

20.5.11 Port Control Register (TMR6_PCNAR)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CAPMDA	-	-	OUTENA	-		EMBSA [2:0]		EMBRA [1:0]		EMBCA [1:0]		-	-		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	Flag	Bit Name	Function	Read and write
b31	CAPMDA	Functional mode selection A	0: Comparison output function 1: Capture input function	R/W
b30~b29	Reserved	-	Read as "0", write as "0"	R/W
b28	OUTENA	Output enable A	0: TIM6_<t>_PWMA port output is invalid when Timer6 is functioning 1: TIM6_<t>_PWMA port output is valid when Timer6 is functioning	R/W
b27	Reserved	-	Read as "0", write as "0"	R/W
b26~b24	EMBSA [2:0]	When an EMB event occurs, the port status is set to A	000: Select EMB group0 control event 001: Select EMB's group1 control event 010: Select EMB's group2 control event 011: Select EMB's group3 control event 100 : Select EMB group4 control event 101: Select EMB's group5 control event	R/W
b23~b22	EMBRA [1:0]	Select A for EMB release mode	00: When the EMB group control event selected by EMBSA [2:0] is invalid, the TIM6_<t>_PWMA port is released immediately (One Shot) 01: When the EMB group control event selected by EMBSA [2:0] is invalid, the TIM6_<t>_PWMA port is released when the counter counts to overflow (Cycle By Cycle 1) 10 : When the EMB group control event selected by EMBSA [2:0] is invalid, the TIM6_<t>_PWMA port is released when the counter counts to underflow (Cycle By Cycle 2) 11 : When the EMB group control event selected by EMBSA [2:0] is invalid, the TIM6_<t>_PWMA port is released when the counter counts to overflow or underflow (Cycle By Cycle 3)	R/W
b21~b20	EMBCA [1:0]	EMB event port status set to A	00: When the EMB group control event selected by EMBSA [2:0] occurs, the TIM6_<t>_PWMA port outputs normally 01: When the EMB group control event selected by EMBSA [2:0] occurs, the TIM6_<t>_PWMA port outputs high impedance state 10 : When the EMB group control event selected by EMBSA [2:0] occurs, the TIM6_<t>_PWMA port outputs a low level 11 : When the EMB group control event selected by EMBSA [2:0] occurs, the TIM6_<t>_PWMA port outputs a high level	R/W
b19~b18	Reserved	-	Read as "0", write as "0"	R/W
b17~b16	FORCA [1:0]	Force port status setting A	0x: Set invalid 10 : The next period starts, the TIM6_<t>_PWMA port output is set to low level 11 : The next period starts, the TIM6_<t>_PWMA port output is set to high level Note 1: The next period 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 period.	R/W
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 TIM6_<t>_PWMA port output is set to low level 01: During down counting, when the timer count value is equal to GCMBR, the TIM6_<t>_PWMA port output is set to high level 10: During down counting, when the timer count value is equal to GCMBR, the TIM6_<t>_PWMA port output maintains the previous state 11: During down counting, when the timer count value is equal to GCMBR, the TIM6_<t>_PWMA port output is set to the inverted level	R/W
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 TIM6_<t>_PWMA port output is set to low level 01: During the up counting period, when the timer count value	R/W

			is equal to GCMBR, the TIM6_<t>_PWMA port output is set to high level 10: During up counting, when the timer count value is equal to GCMBR, the TIM6_<t>_PWMA port output maintains the previous state 11: During the up counting period, when the timer count value is equal to GCMBR, the TIM6_<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 TIM6_<t>_PWMA port output is set to low level 01: During down counting, when the timer count value is equal to GCMAR, the TIM6_<t>_PWMA port output is set to high level 10: During down counting, when the timer count value is equal to GCMAR, the TIM6_<t>_PWMA port output maintains the previous state 11: During down counting, when the timer count value is equal to GCMAR, the TIM6_<t>_PWMA port output is set to the inverted level	R/W
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 TIM6_<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 TIM6_<t>_PWMA port output is set to high level 10: During up counting, when the timer count value is equal to GCMAR, the TIM6_<t>_PWMA port output maintains the previous state 11: During the up counting period, when the timer count value is equal to GCMAR, the TIM6_<t>_PWMA port output is set to the inverted level	R/W
b7~b6	UDFCA [1:0]	Count underflow port status setting A	00: When the count underflows, the TIM6_<t>_PWMA port output is set to low level 01: When the count underflows, the TIM6_<t>_PWMA port output is set to high level 10 : When the count underflows, the TIM6_<t>_PWMA port output keeps the previous state 11 : When the count overflows, the TIM6_<t>_PWMA port output is set to the inverted level	R/W
b5~b4	OVFCA [1:0]	Count overflow port status setting A	00: When the count overflows, the TIM6_<t>_PWMA port output is set to low level 01: When the count overflows, the TIM6_<t>_PWMA port output is set to high level 10 : When the count underflows, the TIM6_<t>_PWMA port output keeps the previous state 11 : When the count overflows, the TIM6_<t>_PWMA port output is set to the inverted level	R/W
b3~b2	STPCA [1:0]	Count stop port state setting A	00: When count stops, TIM6_<t>_PWMA port output is set to low 01: When count stops, TIM6_<t>_PWMA port output is set to high 10: When the count stops, the TIM6_<t>_PWMA port output remains in the previous state 11: When the count stops, the TIM6_<t>_PWMA port output remains in the previous state	R/W
b1~b0	STACA [1:0]	Count start port status setting A	00: When counting starts, the output of TIM6_<t>_PWMA port is set to low level 01: When counting starts, TIM6_<t>_PWMA port output is set to high level 10: When counting starts, the TIM6_<t>_PWMA port output maintains the previous state 11: When counting starts, the TIM6_<t>_PWMA port output maintains the previous state	R/W

20.5.12 Port Control Register (TMR6_PCNBR)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CAP MDB	-	-	OUT ENB	-		EMBS B [2:0]		EMBR B [1:0]		EMBC B [1:0]		-	-		FORC B [1:0]
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CMBDC B [1:0]		CMBUC B [1:0]		CMADC B [1:0]		CMAUC B [1:0]		UDFC B [1:0]		OVFC B [1:0]		STPC B [1:0]		STAC B [1:0]	

Bit	Flag	Bit Name	Function	Read and write
b31	CAPMDB	Function mode selection B	0: Comparison output function 1: Capture input function	R/W
b30~b29	Reserved	-	Read as "0", write as "0"	R/W
b28	OUTENB	Output enable B	0: TIM6_<t>_PWMB port output is invalid when Timer6 is functioning 1: TIM6_<t>_PWMB port output is valid when Timer6 is functioning	R/W
b27	Reserved	-	Read as "0", write as "0"	R/W
b26~b24	EMBSB [2:0]	When an EMB event occurs, the port status is set to B	000: Select EMB group0 control event 001: Select EMB's group1 control event 010: Select EMB's group2 control event 011: Select EMB's group3 control event 100 : Select EMB group4 control event 101: Select EMB's group5 control event	R/W
b23~b22	EMBRB [1:0]	EMB release mode selection B	00: When the EMB group control event selected by EMBSB [2:0] is invalid, the TIM6_<t>_PWMB port is released immediately (One Shot) 01: When the EMB group control event selected by EMBSB [2:0] is invalid, the TIM6_<t>_PWMB port is released when the counter counts to overflow (Cycle By Cycle 1) 10 : When the EMB group control event selected by EMBSB [2:0] is invalid, the TIM6_<t>_PWMB port is released when the counter counts to underflow (Cycle By Cycle 2) 11 : When the EMB group control event selected by EMBSB [2:0] is invalid, the TIM6_<t>_PWMB port is released when the counter counts to overflow or underflow (Cycle By Cycle 3)	R/W
b21~b20	EMBCB [1:0]	Port status setting B during EMB event	00: When the EMB group control event selected by EMBSB [2:0] occurs, the TIM6_<t>_PWMB port outputs normally 01: When the EMB group control event selected by EMBSB [2:0] occurs, the TIM6_<t>_PWMB port outputs high impedance state 10 : When the EMB group control event selected by EMBSB [2:0] occurs, the TIM6_<t>_PWMB port outputs a low level 11 : When the EMB group control event selected by EMBSB [2:0] occurs, the TIM6_<t>_PWMB port outputs a high level	R/W
b19~b18	Reserved	-	Read as "0", write as "0"	R/W
b17~b16	FORCB [1:0]	Forced port status setting B	0x: Set invalid 10 : The next period starts, the TIM6_<t>_PWMB port output is set to low level 11 : The next period starts, the TIM6_<t>_PWMB port output is set to high level Note 1: The next period 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 period.	R/W
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 TIM6_<t>_PWMB port output is set to low level 01: During down counting, when the timer count value is equal to GCMBR, the TIM6_<t>_PWMB Aport output is set to high level 10: During down counting, when the timer count value is equal to GCMBR, the TIM6_<t>_PWMB port output maintains the previous state 11: During down counting, when the timer count value is equal to GCMBR, the TIM6_<t>_PWMB port output is set to the inverted level	R/W
b13~b12	CMBUCB [1:0]	Count up && when comparison value B matches, port state is set to B	00: During down counting, when the timer count value is equal to GCMBR, the TIM6_<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 TIM6_<t>_PWMB port output is set to	R/W

			high level 10: During up counting, when the timer count value is equal to GCMBR, the TIM6_<t>_PWMB port output maintains the previous state 11: During the up counting period, when the timer count value is equal to GCMBR, the TIM6_<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 TIM6_<t>_PWMB port output is set to low level 01: During down counting, when the timer count value is equal to GCMAR, the TIM6_<t>_PWMB Aport output is set to high level 10: During down counting, when the timer count value is equal to GCMAR, the TIM6_<t>_PWMB port output maintains the previous state 11: During down counting, when the timer count value is equal to GCMAR, the TIM6_<t>_PWMB port output is set to the inverted level	R/W
b9~b8	CMAUCB [1:0]	Count up && 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 TIM6_<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 TIM6_<t>_PWMB port output is set to high level 10: During up counting, when the timer count value is equal to GCMAR, the TIM6_<t>_PWMB port output maintains the previous state 11: During the up counting period, when the timer count value is equal to GCMAR, the TIM6_<t>_PWMB port output is set to the inverted level	R/W
b7~b6	UDFCB [1:0]	Count underflow port status setting B	00 : When the count underflows, the TIM6_<t>_PWMB port output is set to the inverted level 01: When the count underflows, the TIM6_<t>_PWMB port output is set to high level 10 : When the count underflows, the TIM6_<t>_PWMBport output keeps the previous state 11 : When the count underflows, the TIM6_<t>_PWMB port output is set to the inverted level	R/W
b5~b4	OVFCB [1:0]	Count overflow port status setting B	00: When the count overflows, the TIM6_<t>_PWMB port output is set to low level 01: When the count overflows, the TIM6_<t>_PWMB port output is set to high level 10 : When the count overflows, the TIM6_<t>_PWMB port output maintains the previous state 11 : When the count overflows, the TIM6_<t>_PWMB port output is set to the inverted level	R/W
b3~b2	STPCB [1:0]	Count stop port status setting B	00: When counting stops, TIM6_<t>_PWMB port output is set to low level 01: When counting stops, TIM6_<t>_PWMB port output is set to high level 10: When the count stops, the TIM6_<t>_PWMB port output remains in the previous state 11: When the count stops, the TIM6_<t>_PWMB port output remains in the previous state	R/W
b1~b0	STBCB [1:0]	Count start port status setting B	00: When counting starts, TIM6_<t>_PWMB port output is set to low level 01: When counting starts, TIM6_<t>_PWMB port output is set to high level 10: When counting starts, the TIM6_<t>_PWMB port output maintains the previous state 11: When counting starts, the TIM6_<t>_PWMB port output maintains the previous state	R/W

20.5.13 Filter Control Register (TMR6_FCNGR)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved									NOFI CKGB [1:0]		NOFI ENGB	-	NOFI CKGA [1:0]		NOFI ENGA
Bit		Flag		Bit Name		Function								Read and write	
b31~b7		Reserved		-		Read as "0", write as "0"								R/W	
b6~b5		NOFICKGB [1:0]		Filter sampling reference clock selection GB		00: PCLK0 01: PCLK0/4 10: PCLK0/16 11: PCLK0/64								R/W	
b4		NOFIENGB		Capture input port filter GB		0: The filter function of the TIM6_<t>_PWMB input port of this unit is invalid 1: The unit TIM6_<t>_PWMB input port filter function is enabled								R/W	
b3		Reserved		-		Read as "0", write as "0"								R/W	
b2~b1		NOFICKGA [1:0]		Filter sampling reference clock selection GA		00: PCLK0 01: PCLK0/4 10: PCLK0/16 11: PCLK0/64								R/W	
b0		NOFIENGA		Capture input port filter GA		0: The filter function of the TIM6_<t>_PWMA input port of this unit is invalid 1: The unit TIM6_<t>_PWMA input port filtering function is enabled								R/W	

20.5.14 Filter Control Register (TMR6_FCNT)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	NOFI CKTD [1:0]	NOFI ENTD	-	NOFI CKTC [1:0]	NOFI ENTC	-	NOFI CKTB [1:0]	NOFI ENTB	-	NOFI CKTA [1:0]	NOFI ENTA				
Bit	Flag	Bit Name	Function		Read and write										
b31~b15	Reserved	-	Read as "0", write as "0"		R/W										
b14~b13	NOFICKTD [1:0]	Filter sampling reference clock selection TD	00: PCLK0 01: PCLK0/4 10: PCLK0/16 11: PCLK0/64		R/W										
b12	NOFIENTD	Capture input port filter TD	0: TIM6_TRIGD port input filter function is invalid 1: TIM6_TRIGD port input filter function is enabled		R/W										
b11	Reserved	-	Read as "0", write as "0"		R/W										
b10~b9	NOFICKTC [1:0]	Filter sampling reference clock selection TC	00: PCLK0 01: PCLK0/4 10: PCLK0/16 11: PCLK0/64		R/W										
b8	NOFIENTC	Capture input port filter TC	0: TIM6_TRIGC port input filter function is invalid 1: TIM6_TRIGC port input filter function is enabled		R/W										
b7	Reserved	-	Read as "0", write as "0"		R/W										
b6~b5	NOFICKTB [1:0]	Filter sampling reference clock selection TB	00: PCLK0 01: PCLK0/4 10: PCLK0/16 11: PCLK0/64		R/W										
b4	NOFIENTB	Capture input port filter TB	0: TIM6_TRIGB port input filter function is invalid 1: TIM6_TRIGB port input filter function is enabled		R/W										
b3	Reserved	-	Read as "0", write as "0"		R/W										
b2~b1	NOFICKTA [1:0]	Filter sampling reference clock selection TA	00: PCLK0 01: PCLK0/4 10: PCLK0/16 11: PCLK0/64		R/W										
b0	NOFIENTA	Capture input port filter TA	0: TIM6_TRIGA port input filter function is invalid 1: TIM6_TRIGA port input filter function is enabled		R/W										

20.5.15 Valid Period Register (TMR6_VPERR)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	
Reserved											PCNTS [2:0]			PCNTE [1:0]		
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
Reserved						SP PERIB	SP PERIA	Reserved								
Bit		Flag		Bit Name		Function									Read and write	
b31~b21		Reserved		-		Read as "0", write as "0"									R/W	
b20~b18		PCNTS [2:0]		Valid period selection		000: Valid period selection function is invalid 001: Valid every 1 period 010: Valid every 2 period 011: Valid every 3 period 100: Valid every 4 period 101: Valid every 5 period 110: Valid every 6 period 111: Valid every 7 period									R/W	
b17~b16		PCNTE [1:0]		Active period 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									R/W	
b15~b10		Reserved		-		Read as "0", write as "0"									R/W	
b9		SPPERIB		Dedicated signal effective period selection B		0: Valid period selection function is invalid 1: Valid period selection function is enabled									R/W	
b8		SPPERIA		Special signal effective period selection A		0: Valid period selection function is invalid 1: Valid period selection function is enabled									R/W	
b7~b0		Reserved		-		Read as "0", write as "0"									R/W	

20.5.16 Status Flag Register (TMR6_STFLR)

Reset value: 0x80000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
DIRF	-	-	-	CMBF2	CMAF2	-	-	VPERNUM [2:0]			-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	CMSBDF	CMSBUF	CMSADF	CMSAUF	DTEF	UDFF	OVFF	CMFF	CMEF	CMDF	CMCF	CMBF	CMAF

Bit	Flag	Bit Name	Function	Read and write
b31	DIRF	Counting Direction	0: count down 1: count up	R
b30-b28	Reserved	-	Read as "0", write as "0"	R
b27	CMBF2	Capture B again	0: No TIM6_<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, TIM6_<t>_PWMB capture completion action occurs again	R/W
b26	CMAF2	Capture A again	0: No TIM6_<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 TIM6_<t>_PWMA capture completion action occurs again	R/W
b25~24	Reserved	-	Read as "0", write as "0"	R
b23~b21	VPERNUM [2:0]	Number of periods	When the effective period selection function is enabled, the number of periods after counting	R
b20~b13	Reserved	-	Read as "0", write as "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	R/W
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	R/W
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	R/W
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	R/W
b8	DTEF	Dead zone error	0: no dead zone error occurred 1: A dead zone 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	R/W
b6	OVFF	Overflow match	0: No sawtooth overflow or triangular wave counting to peak 1: Sawtooth overflow occurs or triangular wave counts to peak	R/W
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	R/W
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	R/W
b3	CMDF	Count match D	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	R/W
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	R/W
b1	CMBF	Count match B	0: The value of the GCMBR register is not equal to the count value, and the TIM6_<t>_PWMB capture completion action has not occurred 1: The value of the GCMBR register is equal to the count	R/W

			value, or a TIM6_<t>_PWMB capture completion action occurs	
b0	CMAF	Count Match A	0: The value of the GCMAR register is not equal to the count value, and the TIM6_<t>_PWMA capture completion action has not occurred 1: The value of the GCMAR register is equal to the count value, or a TIM6_<t>_PWMA capture completion action occurs	R/W

20.5.17 Hardware Start Event Select Register (TMR6_HSTAR)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved								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
-	-	-	-	HSTA 11	HSTA 10	HSTA 9	HSTA 8	STAS	-	-	-	HSTA 3	HSTA 2	HSTA 1	HSTA 0

Bit	Flag	Bit Name	Function	Read and write
b31~b24	Reserved	-	Read as "0", write as "0"	R/W
b23	HSTA23	Hardware startup condition 23	Condition: TIM6_TRIGD port is sampled to the falling edge 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	R/W
b22	HSTA22	Hardware startup condition 22	Condition: TIM6_TRIGD port is sampled to the rising edge 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	R/W
b21	HSTA21	Hardware startup condition 21	Condition: TIM6_TRIGC port is sampled to the falling edge 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	R/W
b20	HSTA20	Hardware startup condition 20	Condition: TIM6_TRIGC port is sampled to the rising edge 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	R/W
b19	HSTA19	Hardware startup condition 19	Condition: TIM6_TRIGB port is sampled to the falling edge 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	R/W
b18	HSTA18	Hardware startup condition 18	Condition: TIM6_TRIGB port is sampled to the rising edge 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	R/W
b17	HSTA17	Hardware startup condition 17	Condition: TIM6_TRIGA port is sampled to the falling edge 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	R/W
b16	HSTA16	Hardware startup condition 16	Condition: TIM6_TRIGA port is sampled to the rising edge 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	R/W
b15~b12	Reserved	-	Read as "0", write as "0"	R/W
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	R/W
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	R/W
b9	HSTA9	Hardware startup condition 9	Condition: internal hardware trigger event 1 valid 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	R/W
b8	HSTA8	Hardware startup condition 8	Condition: internal hardware trigger event 0 valid 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	R/W
b7	STAS	Hardware enable	0: hardware startup invalid 1: Hardware startup valid Note: When the hardware boot is valid, the setting of SSTAR is invalid	R/W
b6~b4	Reserved	-	Read as "0", write as "0"	R/W
b3	HSTA3	Hardware startup condition 3	Condition: TIM6_<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	R/W
b2	HSTA2	Hardware startup condition 2	Condition: TIM6_<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	R/W
b1	HSTA1	Hardware startup condition 1	Condition: TIM6_<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	R/W
b0	HSTA0	Hardware startup condition 0	Condition: TIM6_<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	R/W

20.5.18 Hardware Stop Event Select Register (TMR6_HSTPR)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved								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
-	-	-	-	HSTP 11	HSTP 10	HSTP 9	HSTP 8	STPS	-	-	-	HSTP 3	HSTP 2	HSTP 1	HSTP 0

Bit	Flag	Bit Name	Function	Read and write
b31~b24	Reserved	-	Read as "0", write as "0"	R/W
b23	HSTP23	Hardware stop condition 23	Condition: TIM6_TRIGD port is sampled to the falling edge 0: Hardware stop invalid when condition matches 1: Hardware stops working when conditions match	R/W
b22	HSTP22	Hardware stop condition 22	Condition: TIM6_TRIGD port is sampled to the rising edge 0: Hardware stop invalid when condition matches 1: Hardware stops working when conditions match	R/W
b21	HSTP21	Hardware stop condition 21	Condition: TIM6_TRIGC port is sampled to the falling edge 0: Hardware stop invalid when condition matches 1: Hardware stops working when conditions match	R/W
b20	HSTP20	Hardware stop condition 20	Condition: TIM6_TRIGC port is sampled to the rising edge 0: Hardware stop invalid when condition matches 1: Hardware stops working when conditions match	R/W
b19	HSTP19	Hardware stop condition 19	Condition: TIM6_TRIGB port is sampled to the falling edge 0: Hardware stop invalid when condition matches 1: Hardware stops working when conditions match	R/W
b18	HSTP18	Hardware stop condition 18	Condition: TIM6_TRIGB port is sampled to the rising edge 0: Hardware stop invalid when condition matches 1: Hardware stops working when conditions match	R/W
b17	HSTP17	Hardware stop condition 17	Condition: TIM6_TRIGA port is sampled to the falling edge 0: Hardware stop invalid when condition matches 1: Hardware stops working when conditions match	R/W
b16	HSTP16	Hardware stop condition 16	Condition: TIM6_TRIGA port is sampled to the rising edge 0: Hardware stop invalid when condition matches 1: Hardware stops working when conditions match	R/W
b15~b12	Reserved	-	Read as "0", write as "0"	R/W
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	R/W
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	R/W
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	R/W
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	R/W
b7	STPS	Hardware disable	0: hardware stop is invalid 1: Hardware stop is valid Note: When the hardware stop is valid, the setting of SSTPR is invalid	R/W
b6~b4	Reserved	-	Read as "0", write as "0"	R/W
b3	HSTP3	Hardware stop condition 3	Condition: TIM6_<t>_PWMB port is sampled to the falling edge 0: Hardware stop invalid when condition matches 1: Hardware stops working when conditions match	R/W
b2	HSTP2	Hardware stop condition 2	Condition: TIM6_<t>_PWMB port is sampled to the rising edge 0: Hardware stop invalid when condition matches 1: Hardware stops working when conditions match	R/W
b1	HSTP1	Hardware stop condition 1	Condition: TIM6_<t>_PWMA port is sampled to the falling edge 0: Hardware stop invalid when condition matches 1: Hardware stops working when conditions match	R/W
b0	HSTP0	Hardware stop condition 0	Condition: TIM6_<t>_PWMA port is sampled to the rising edge 0: Hardware stop invalid when condition matches 1: Hardware stops working when conditions match	R/W

20.5.19 Hardware Clear Event Selection Register (TMR6_HCLRR)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved								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
-	-	-	-	HCLE 11	HCLE 10	HCLE 9	HCLE 8	CLES	-	-	-	HCLE 3	HCLE 2	HCLE 1	HCLE 0

Bit	Flag	Bit Name	Function	Read and write
b31~b24	Reserved	-	Read as "0", write as "0"	R/W
b23	HCLE23	Hardware zeroing condition 23	Condition: TIM6_TRIGD port is sampled to the falling edge 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	R/W
b22	HCLE22	Hardware zeroing condition 22	Condition: TIM6_TRIGD port is sampled to the rising edge 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	R/W
b21	HCLE21	Hardware zeroing condition 21	Condition: TIM6_TRIGC port is sampled to the falling edge 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	R/W
b20	HCLE20	Hardware zeroing condition 20	Condition: TIM6_TRIGC port is sampled to the rising edge 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	R/W
b19	HCLE19	Hardware zeroing condition 19	Condition: TIM6_TRIGB port is sampled to the falling edge 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	R/W
b18	HCLE18	Hardware zeroing condition 18	Condition: TIM6_TRIGB port is sampled to the rising edge 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	R/W
b17	HCLE17	Hardware zeroing condition 17	Condition: TIM6_TRIGA port is sampled to the falling edge 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	R/W
b16	HCLE16	Hardware zeroing condition 16	Condition: TIM6_TRIGA port is sampled to the rising edge 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	R/W
b15~b12	Reserved	-	Read as "0", write as "0"	R/W
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	R/W
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	R/W
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	R/W
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	R/W
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	R/W
b6~b4	Reserved	-	Read as "0", write as "0"	R/W
b3	HCLE3	Hardware zeroing condition 3	Condition: TIM6_<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	R/W
b2	HCLE2	Hardware zeroing condition 2	Condition: TIM6_<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	R/W
b1	HCLE1	Hardware zeroing condition 1	Condition: TIM6_<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	R/W
b0	HCLE0	Hardware zeroing condition 0	Condition: TIM6_<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	R/W

20.5.20 Hardware Refresh Event Selection Register (TMR6_HUPDR)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved								HUPD 23	HUPD 22	HUPD 21	HUPD 20	HUPD 19	HUPD 18	HUPD 17	HUPD 16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	HUPD 11	HUPD 10	HUPD 9	HUPD 8	UPDS	-	-	-	HUPD 3	HUPD 2	HUPD 1	HUPD 0

Bit	Flag	Bit Name	Function	Read and write
b31~b24	Reserved	-	Read as "0", write as "0"	R/W
b23	HUPD23	Hardware refresh condition 23	Condition: TIM6_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	R/W
b22	HUPD22	Hardware refresh condition 22	Condition: TIM6_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	R/W
b21	HUPD21	Hardware refresh condition 21	Condition: TIM6_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	R/W
b20	HUPD20	Hardware refresh condition 20	Condition: TIM6_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	R/W
b19	HUPD19	Hardware refresh condition 19	Condition: TIM6_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	R/W
b18	HUPD18	Hardware refresh condition 18	Condition: TIM6_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	R/W
b17	HUPD17	Hardware refresh condition 17	Condition: TIM6_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	R/W
b16	HUPD16	Hardware refresh condition 16	Condition: TIM6_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	R/W
b15~b12	Reserved	-	Read as "0", write as "0"	R/W
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	R/W
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	R/W
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	R/W
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	R/W
b7	UPDS	Hardware refresh enable	0: Hardware refresh invalid 1: Hardware refresh valid Note: When hardware refresh is enabled, the SUPDR setting disable.	R/W
b6~b4	Reserved	-	Read as "0", write as "0"	R/W
b3	HUPD3	Hardware refresh condition 3	Condition: TIM6_<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	R/W
b2	HUPD2	Hardware refresh condition 2	Condition: TIM6_<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	R/W
b1	HUPD1	Hardware refresh condition 1	Condition: TIM6_<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	R/W
b0	HUPD0	Hardware refresh condition 0	Condition: TIM6_<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	R/W

20.5.21 Hardware Capture Event Select Register A (TMR6_HCPAR)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved								HCP A23	HCP A22	HCP A21	HCP A20	HCP A19	HCP A18	HCP A17	HCP A16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	HCP A11	HCP A10	HCP A9	HCP A8	-	-	-	-	HCP A3	HCP A2	HCP A1	HCP A0

Bit	Flag	Bit Name	Function	Read and write
b31~b24	Reserved	-	Read as "0", write as "0"	R/W
b23	HCPA23	Hardware Capture A Condition 23	Condition: TIM6_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	R/W
b22	HCPA22	Hardware capture A condition 22	Condition: TIM6_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	R/W
b21	HCPA21	Hardware capture A condition 21	Condition: TIM6_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	R/W
b20	HCPA20	Hardware capture A condition 20	Condition: TIM6_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	R/W
b19	HCPA19	Hardware capture A condition 19	Condition: TIM6_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	R/W
b18	HCPA18	Hardware capture A condition 18	Condition: TIM6_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	R/W
b17	HCPA17	Hardware capture A condition 17	Condition: TIM6_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	R/W
b16	HCPA16	Hardware capture A condition 16	Condition: TIM6_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	R/W
b15~b12	Reserved	-	Read as "0", write as "0"	R/W
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	R/W
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	R/W
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	R/W
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	R/W

			1: When the condition is matched, the hardware capture A is valid	
b7~b4	Reserved	-	Read as "0", write as "0"	R/W
b3	HCPA3	Hardware Capture A Condition 3	Condition: TIM6_<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	R/W
b2	HCPA2	Hardware capture A condition 2	Condition: TIM6_<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	R/W
b1	HCPA1	Hardware capture A condition 1	Condition: TIM6_<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	R/W
b0	HCPA0	Hardware capture A condition 0	Condition: TIM6_<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	R/W

20.5.22 Hardware Capture Event Selection Register B (TMR6_HCPBR)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved								HCP B23	HCP B22	HCP B21	HCP B20	HCP B19	HCP B18	HCP B17	HCP B16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	HCP B11	HCP B10	HCP B9	HCP B8	-	-	-	-	HCP B3	HCP B2	HCP B1	HCP B0

Bit	Flag	Bit Name	Function	Read and write
b31~b24	Reserved	-	Read as "0", write as "0"	R/W
b23	HCPB23	Hardware Capture B Condition 23	Condition: TIM6_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	R/W
b22	HCPB22	Hardware capture B condition22	Condition: TIM6_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	R/W
b21	HCPB21	Hardware capture B condition 21	Condition: TIM6_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	R/W
b20	HCPB20	Hardware capture B condition 20	Condition: TIM6_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	R/W
b19	HCPB19	Hardware capture B condition19	Condition: TIM6_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	R/W
b18	HCPB18	Hardware capture B condition 18	Condition: TIM6_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	R/W
b17	HCPB17	Hardware capture B condition 17	Condition: TIM6_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	R/W
b16	HCPB16	Hardware capture B condition 16	Condition: TIM6_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	R/W
b15~b12	Reserved	-	Read as "0", write as "0"	R/W
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	R/W
b10	HCPB10	Hardware capture B condition 10	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	R/W
b9	HCPB9	Hardware capture B condition 9	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	R/W
b8	HCPB8	Hardware capture B condition 8	Condition: internal hardware trigger event 0 valid 0: When the condition is matched, the hardware capture	R/W

			B is invalid 1: When the condition is matched, the hardware capture B is valid	
b7~b4	Reserved	-	Read as "0", write as "0"	R/W
b3	HCPB3	Hardware Capture B Condition 3	Condition: TIM6_<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	R/W
b2	HCPB2	Hardware capture B condition 2	Condition: TIM6_<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	R/W
b1	HCPB1	Hardware capture B condition 1	Condition: TIM6_<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	R/W
b0	HCPB0	Hardware capture B condition 0	Condition: TIM6_<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	R/W

20.5.23 Hardware Increment Event Select Register (TMR6_HCUPR)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved								HC UP23	HC UP22	HC UP21	HC UP20	HC UP19	HC UP18	HC UP17	HC UP16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	HC UP11	HC UP10	HC UP9	HC UP8	HC UP7	HC UP6	HC UP5	HC UP4	HC UP3	HC UP2	HC UP1	HC UP0

Bit	Flag	Bit Name	Function	Read and write
b31~b24	Reserved	-	Read as "0", write as "0"	R/W
b23	HCUP23	Hardware increment condition 23	Condition: TIM6_TRIGD port is sampled to the falling edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	R/W
b22	HCUP22	Hardware increment condition 22	Condition: TIM6_TRIGD port is sampled to the rising edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	R/W
b21	HCUP21	Hardware Added Condition 21	Condition: TIM6_TRIGC port is sampled to the falling edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	R/W
b20	HCUP20	Hardware Added Condition 20	Condition: TIM6_TRIGC port is sampled to the rising edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	R/W
b19	HCUP19	Hardware Added Condition 19	Condition: TIM6_TRIGB port is sampled to the falling edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	R/W
b18	HCUP18	Hardware Added Condition 18	Condition: TIM6_TRIGB port is sampled to the rising edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	R/W
b17	HCUP17	Hardware Added Condition 17	Condition: TIM6_TRIGA port is sampled to the falling edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	R/W
b16	HCUP16	Hardware Added Condition 16	Condition: TIM6_TRIGA port is sampled to the rising edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	R/W
b15~b12	Reserved	-	Read as "0", write as "0"	R/W
b11	HCUP11	Hardware increment condition 11	Condition: internal hardware trigger event 3 valid 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	R/W
b10	HCUP10	Hardware increment condition 10	Condition: internal hardware trigger event 2 valid 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	R/W
b9	HCUP9	Hardware increment condition 9	Condition: Internal Hardware Trigger Event 1Valid 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	R/W
b8	HCUP8	Hardware increment condition 8	Condition: Internal Hardware Trigger Event 0Valid 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	R/W
b7	HCUP7	Hardware increment condition 7	Condition: When the TIM6_<t>_PWMB port is high level, the TIM6_<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	R/W
b6	HCUP6	Hardware increment condition 6	Condition: When the TIM6_<t>_PWMB port is high, the TIM6_<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	R/W
b5	HCUP5	Hardware increment condition 5	Condition: When the TIM6_<t>_PWMB port is low level, the TIM6_<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	R/W
b4	HCUP4	Hardware increment condition 4	Condition: When the TIM6_<t>_PWMB port is low, the TIM6_<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	R/W
b3	HCUP3	Hardware increment condition 3	Condition: When the TIM6_<t>_PWMA port is high, the TIM6_<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	R/W

b2	HCUP2	Hardware increment condition 2	Condition: When the TIM6_<t>_PWMA port is high, the TIM6_<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	R/W
b1	HCUP1	Hardware increment condition 1	Condition: When the TIM6_<t>_PWMA port is low, the TIM6_<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	R/W
b0	HCUP0	Hardware increment condition 0	Condition: When the TIM6_<t>_PWMA port is low level, the TIM6_<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	R/W

20.5.24 Hardware Decrement Event Selection Register (TMR6_HCDOR)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved								HC DO23	HC DO22	HC DO21	HC DO20	HC DO19	HC DO18	HC DO17	HC DO16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	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		Flag		Bit Name		Function		Read and write							
b31~b24		Reserved		-		Read as "0", write as "0"		R/W							
b23		HCDO23		Hardware decrement condition 23		Condition: TIM6_TRIGD port is sampled to the falling edge 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match		R/W							
b22		HCDO22		Hardware decrement condition 22		Condition: TIM6_TRIGD port is sampled to the rising edge 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match		R/W							
b21		HCDO21		Hardware decrement condition 21		Condition: TIM6_TRIGC port is sampled to the falling edge 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match		R/W							
b20		HCDO20		Hardware decrement condition 20		Condition: TIM6_TRIGC port is sampled to the rising edge 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match		R/W							
b19		HCDO19		Hardware decrement condition 19		Condition: TIM6_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: TIM6_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: TIM6_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: TIM6_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		Reserved		-		Read as "0", write as "0"		R/W							
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		R/W							
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		R/W							
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		R/W							
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		R/W							
b7		HCDO7		Hardware decrement condition 7		Condition: When the TIM6_<t>_PWMB port is high level, the TIM6_<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		R/W							
b6		HCDO6		Hardware decrement condition 6		Condition: When the TIM6_<t>_PWMB port is high, the TIM6_<t>_PWMA port is sampled to the rising edge 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions		R/W							

			match	
b5	HCDO5	Hardware decrement condition 5	Condition: When the TIM6_<t>_PWMB port is low level, the TIM6_<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	R/W
b4	HCDO4	Hardware decrement condition 4	Condition: When the TIM6_<t>_PWMB port is low, the TIM6_<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	R/W
b3	HCDO3	Hardware decrement condition 3	Condition: When the TIM6_<t>_PWMA port is high, the TIM6_<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	R/W
b2	HCDO2	Hardware decrement condition 2	Condition: When the TIM6_<t>_PWMA port is high, the TIM6_<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	R/W
b1	HCDO1	Hardware decrement condition 1	Condition: When the TIM6_<t>_PWMA port is low, the TIM6_<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	R/W
b0	HCDO0	Hardware decrement condition 0	Condition: When the TIM6_<t>_PWMA port is low level, the TIM6_<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	R/W

20.5.25 Software Synchronous Start Control Register (TMR6_SSTAR)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved						SSTA 10	SSTA 9	SSTA 8	SSTA 7	SSTA 6	SSTA 5	SSTA 4	SSTA 3	SSTA 2	SSTA 1
Bit	Flag	Bit Name		Function		Read and write									
b31~b10	Reserved	-		Read as "0", write as "0"		R/W									
b9	SSTA10	Unit 10 Software Startup		0: Software startup is invalid 1: Software startup enable		R/W									
b8	SSTA9	Unit 9 software start		0: Software startup is invalid 1: Software startup enable		R/W									
b7	SSTA8	Unit 8 software start		0: Software startup is invalid 1: Software startup enable		R/W									
b6	SSTA7	Unit 7 software start		0: Software startup is invalid 1: Software startup enable		R/W									
b5	SSTA6	Unit 6 software start		0: Software startup is invalid 1: Software startup enable		R/W									
b4	SSTA5	Unit 5 software start		0: Software startup is invalid 1: Software startup enable		R/W									
b3	SSTA4	Unit 4 software start		0: Software startup is invalid 1: Software startup enable		R/W									
b2	SSTA3	Unit 3 software start		0: Software startup is invalid 1: Software startup enable		R/W									
b1	SSTA2	Unit 2 software start		0: Software startup is invalid 1: Software startup enable		R/W									
b0	SSTA1	Unit 1 software start		0: Software startup is invalid 1: Software startup enable		R/W									

20.5.26 Software Synchronous Stop Control Register (TMR6_SSTPR)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved						SSTP 10	SSTP 9	SSTP 8	SSTP 7	SSTP 6	SSTP 5	SSTP 4	SSTP 3	SSTP 2	SSTP 1
Bit	Flag	Bit Name		Function		Read and write									
b31~b10	Reserved	-		Read as "0", write as "0"		R/W									
b9	SSTP10	Unit 10 Software Stop		0: Software stop is invalid 1: Software stop enable		R/W									
b8	SSTP9	Unit 9 software stop		0: Software stop is invalid 1: Software stop enable		R/W									
b7	SSTP8	Unit 8 software stop		0: Software stop is invalid 1: Software stop enable		R/W									
b6	SSTP7	Unit 7 software stop		0: Software stop is invalid 1: Software stop enable		R/W									
b5	SSTP6	Unit 6 software stop		0: Software stop is invalid 1: Software stop enable		R/W									
b4	SSTP5	Unit 5 software stop		0: Software stop is invalid 1: Software stop enable		R/W									
b3	SSTP4	Unit 4 software stop		0: Software stop is invalid 1: Software stop enable		R/W									
b2	SSTP3	Unit 3 software stop		0: Software stop is invalid 1: Software stop enable		R/W									
b1	SSTP2	Unit 2 software stop		0: Software stop is invalid 1: Software stop enable		R/W									
b0	SSTP1	Unit 1 software stop		0: Software stop is invalid 1: Software stop enable		R/W									

20.5.27 Software Synchronous Clear Control Register (TMR6_SCLRR)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved						SCLE 10	SCLE 9	SCLE 8	SCLE 7	SCLE 6	SCLE 5	SCLE 4	SCLE 3	SCLE 2	SCLE 1
Bit	Flag	Bit Name		Function		Read and write									
b31~b10	Reserved	-		Read as "0", write as "0"		R/W									
b9	SCLE10	Unit 10 software reset		0: Software reset is invalid 1: Software clear enable		R/W									
b8	SCLE9	Unit 9 software reset		0: Software reset is invalid 1: Software clear enable		R/W									
b7	SCLE8	Unit 8 software reset		0: Software reset is invalid 1: Software clear enable		R/W									
b6	SCLE7	Unit 7 software reset		0: Software reset is invalid 1: Software clear enable		R/W									
b5	SCLE6	Unit 6 software reset		0: Software reset is invalid 1: Software clear enable		R/W									
b4	SCLE5	Unit 5 software reset		0: Software reset is invalid 1: Software clear enable		R/W									
b3	SCLE4	Unit 4 software reset		0: Software reset is invalid 1: Software clear enable		R/W									
b2	SCLE3	Unit 3 software reset		0: Software reset is invalid 1: Software clear enable		R/W									
b1	SCLE2	Unit 2 software reset		0: Software reset is invalid 1: Software clear enable		R/W									
b0	SCLE1	Unit 1 software reset		0: Software reset is invalid 1: Software clear enable		R/W									

20.5.28 Software Synchronous Refresh Control Register (TMR6_SUPDR)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved						SUPD 10	SUPD 9	SUPD 8	SUPD 7	SUPD 6	SUPD 5	SUPD 4	SUPD 3	SUPD 2	SUPD 1
Bit	Flag	Bit Name		Function		Read and write									
b31~b10	Reserved	-		Read as "0", write as "0"		R/W									
b9	SUPD10	Unit 10 software update		0: Software refresh invalid 1: Software refresh enable		R/W									
b8	SUPD9	Unit 9 software update		0: Software refresh invalid 1: Software refresh enable		R/W									
b7	SUPD8	Unit 8 software update		0: Software refresh invalid 1: Software refresh enable		R/W									
b6	SUPD7	Unit 7 software update		0: Software refresh invalid 1: Software refresh enable		R/W									
b5	SUPD6	Unit 6 software update		0: Software refresh invalid 1: Software refresh enable		R/W									
b4	SUPD5	Unit 5 software update		0: Software refresh invalid 1: Software refresh enable		R/W									
b3	SUPD4	Unit 4 software update		0: Software refresh invalid 1: Software refresh enable		R/W									
b2	SUPD3	Unit 3 software update		0: Software refresh invalid 1: Software refresh enable		R/W									
b1	SUPD2	Unit 2 software update		0: Software refresh invalid 1: Software refresh enable		R/W									
b0	SUPD1	Unit 1 software update		0: Software refresh invalid 1: Software refresh enable		R/W									

20.6 Precautions

- 1) The CPU cannot write to the COUNTER register when the counter is counting, and can only write to the COUNTER register when the counter stops.
- 2) When the corresponding cache function of GCMCR~GCMFR valid, its interrupt output and event output are invalid.
- 3) 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.
- 4) 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 (COUNTER) is shown in Table 20-5 below.

Table 20-5 Counter (COUNTER) Control Priority

Timer Control Events		Counter (COUNTER) Value	Priority
The CPU performs a write operation on COUNTER		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 count		Normal count value	↓
Sawtooth mode count		Normal count value	Low

- 5) 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 (TIM6_<t>_PWMA and TIM6_<t>_PWMB) are shown in Table 20-6 and Table 20-7 respectively.

Table 20-6 PWMA Port Output Control Priority

PWMA Port Output Control Event		Port Status	Priority
EMB selected event occurs		PCNAR.EMBCA setting status	high
Set forced port output		PCNAR.FORCA setting status	↓
When counting starts		PCNAR.STACA setting status	↓
When the sawtooth wave counts up	COUNTER=PERAR	PCNAR.OVFCA set status	↓
	COUNTER=GCMAR	PCNAR.CMAUCA set status	↓
	COUNTER=GCMBR	PCNAR.CMBUCA setting status	↓
When the sawtooth wave counts down	COUNTER=0	PCNAR.UDFCA set status	↓
	COUNTER=GCMAR	PCNAR.CMADCA set status	↓
	COUNTER=GCMBR	PCNAR.CMBDCA set status	↓
When the triangle wave counts up	COUNTER=GCMAR	PCNAR.CMAUCA set status	↓
	COUNTER=GCMBR	PCNAR.CMBUCA setting status	↓

PWMA Port Output Control Event		Port Status	Priority
	CNTER=PERAR	PCNAR.OVFCA set status	↓
When the triangle wave counts down	CNTER=GCMAR	PCNAR.CMADCA set status	↓
	CNTER=GCMBR	PCNAR.CMBDCA set status	↓
	CNTER=0	PCNAR.UDFCA set status	↓
When counting stops		PCNAR.STPCA setting status	Low

Table 20-7 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.OVFCEB set status	↓
	CNTER=GCMBR	PCNBR.CMBUCB setting status	↓
	CNTER=GCMAR	PCNBR.CMAUCB setting status	↓
When the sawtooth wave counts down	CNTER=0	PCNBR.UDFCEB 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.OVFCEB set status	↓
When the triangle wave counts down	CNTER=GCMBR	PCNBR.CMBDCB setting status	↓
	CNTER=GCMAR	PCNBR.CMADCB set status	↓
	CNTER=0	PCNBR.UDFCEB setting status	↓
When counting stops		PCNBR.STPCB setting status	Low

21 General Control Timer (Timer4)

21.1 Introduction

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.

21.2 Basic Block Diagram

The basic functions and characteristics of Timer4 are shown in Table 21-1.

Table 21-1 Basic Functions and Features of Timer4

Waveform mode	Sawtooth Wave, Triangular Wave
Basic functions	• Direction of increments and decrements
	• Cache function
	• General PWM Output
	• Generic compare match event triggers other peripherals
	• Overflow, underflow, and reload count match events trigger other peripherals
	• Dedicated compare match event triggers other peripherals
	• Other peripheral startup count
	• EMB control
Interrupt type	General compare match interrupt
	Count period match interrupt
	Heavy load count matching interrupt
	Dedicated compare match interrupt

As shown in Figure 21-1, the basic structure of the General Control timer Timer4 is described. As shown in the block diagram, "< t >" represents the unit number, that is, "< t >" is 1. This section refers to "< t >" when referring to "< t >" later.

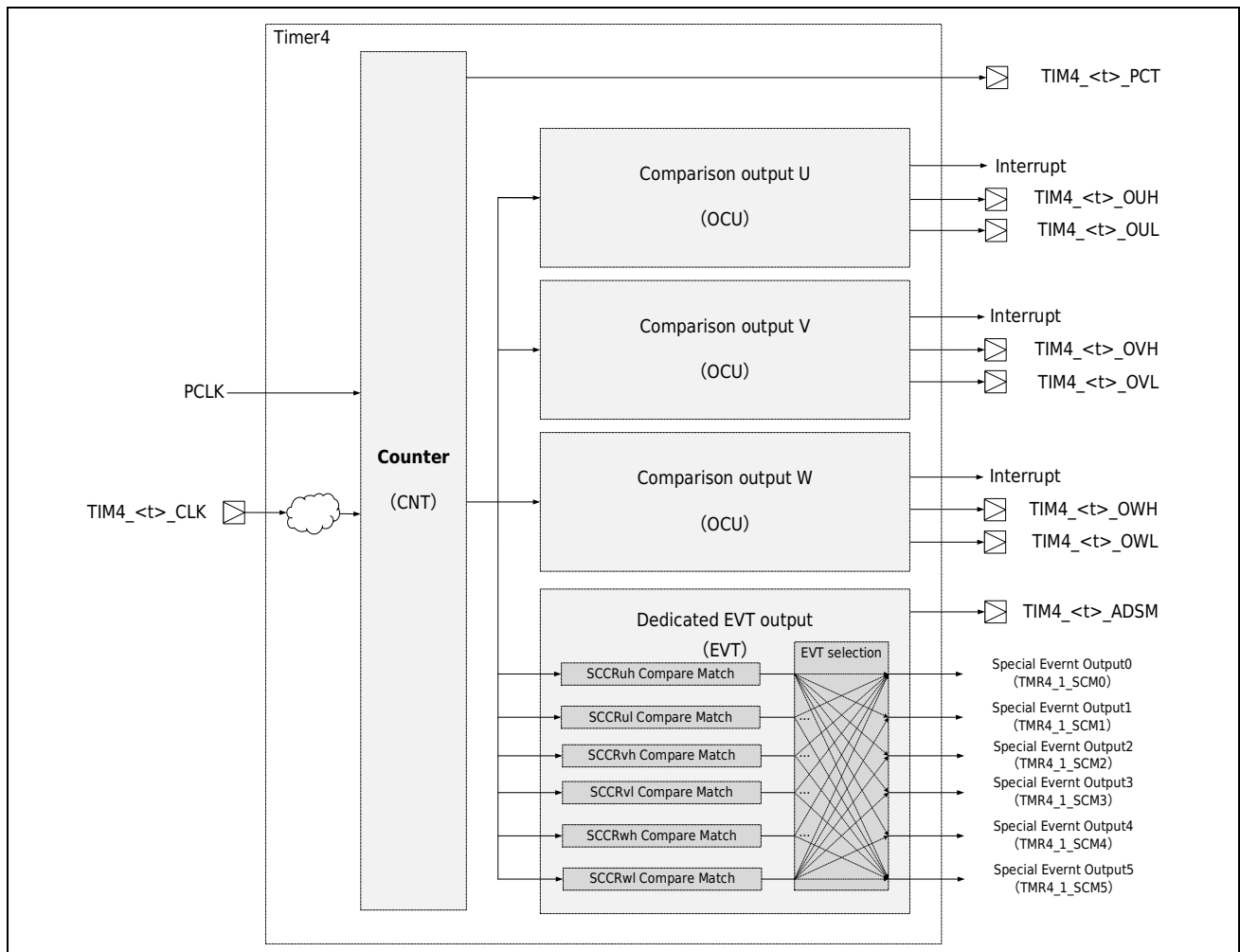


Figure 21-1 Timer4 Basic Block Diagram

Table 21-2 shows the list of input and output ports of Timer4.

Table 21-2 Timer4 Port List

Port Name	Direction	Function
TIM4_<t>_CLK	Input	Counting clock input port
TIM4_<t>_PCT	Output	PWM periodic output monitoring
TIM4_<t>_ADSM	Output	Dedicated event output monitoring
TIM4_<t>_OUH	Output	PWM output port
TIM4_<t>_OUL		
TIM4_<t>_OVH		
TIM4_<t>_OVL		
TIM4_<t>_OWH		
TIM4_<t>_OWL		

21.3 Functional Description

21.3.1 Basic Action

21.3.1.1 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 21-2 and Figure 21-3 below are the waveform diagrams of sawtooth wave and triangular wave respectively.

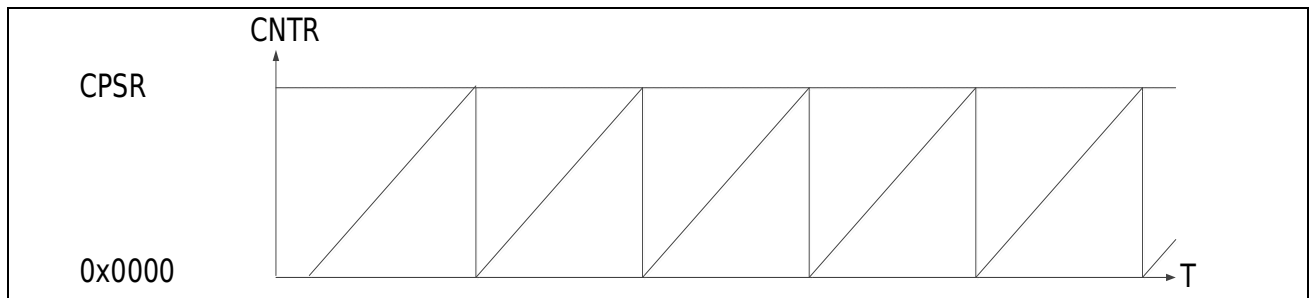


Figure 21-2 Timer4 Sawtooth Waveform

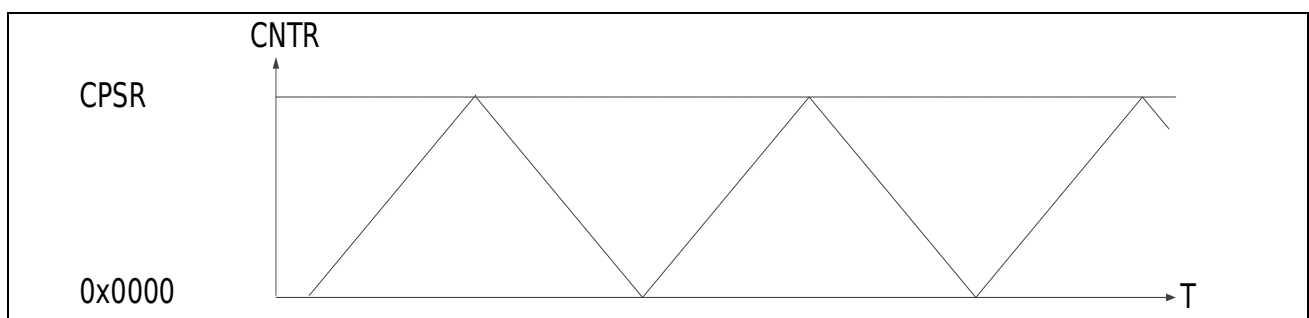


Figure 21-3 Timer4 Triangle Wave Waveform

21.3.1.2 Counting Action

1) The sawtooth wave counting operation and control flow are as Figure 21-4 shown.

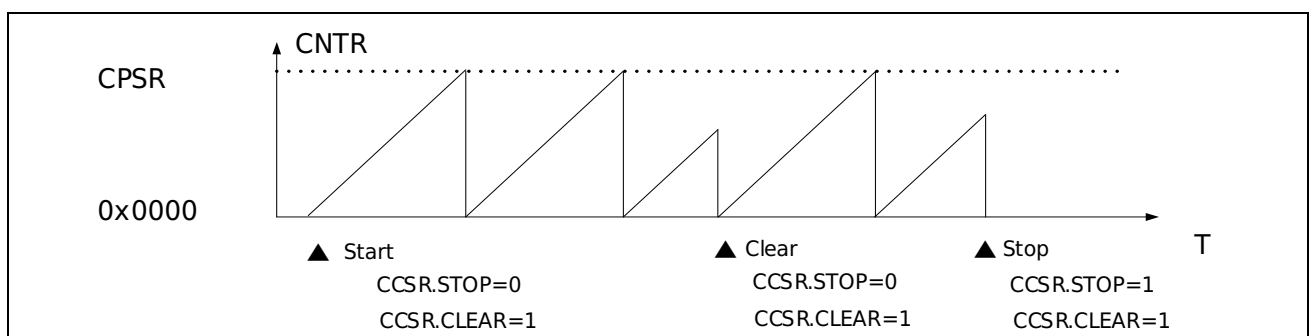


Figure 21-4 Timer4 Sawtooth Wave Mode Counting Action

- Set mode CCSR.MODE = 0.
- Sets the count peak CPSR register.
- 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.

- d) Counting period = (CPSR + 1) × Counting clock period
- e) During counting, write CCSR.STOP = 0 and CCSR.CLEAR = 1 to initialize the count value to 0x0000 and continue counting operations.
- f) 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 21-5 shown.

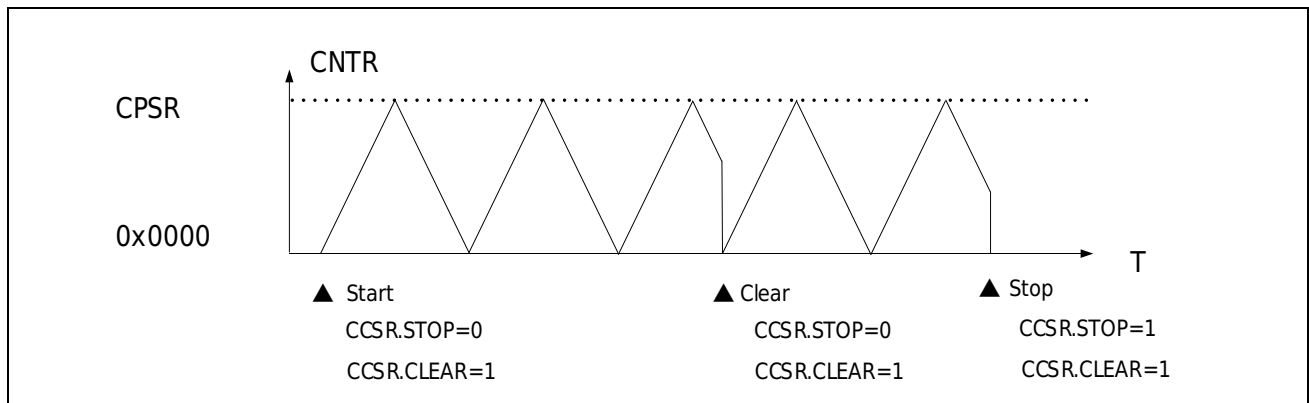


Figure 21-5 Timer4 triangular wave mode counting action

- a) Set mode CCSR.MODE = 1.
- b) Sets the count peak CPSR register.
- c) 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;
- d) Counting period = (CPSR) × 2 × Clock period
- e) During the counting process, write CCSR.STOP=0 and CCSR.CLEAR=1 to initialize the counting value to 0x0000 and restart the counting up operation, and then repeat the above operation.
- f) During counting, write CCSR.STOP = 1 and CCSR.CLEAR = 1 to initialize the count value to 0x0000 and stop the counting operation.

21.3.1.3 Comparison Output

- 1) The following Figure 21-6 is an instance of the waveform output of the comparison output module in sawtooth wave count mode.

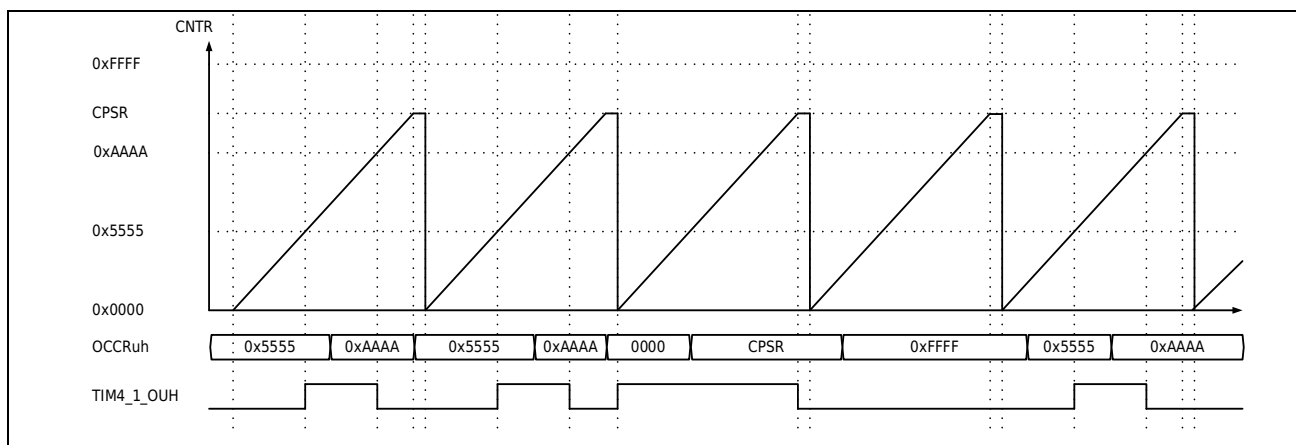


Figure 21-6 Waveform Output Instance in Sawtooth Mode

- 2) The following Figure 21-7 is an instance of the waveform output of the comparison output module in the triangle wave count mode.

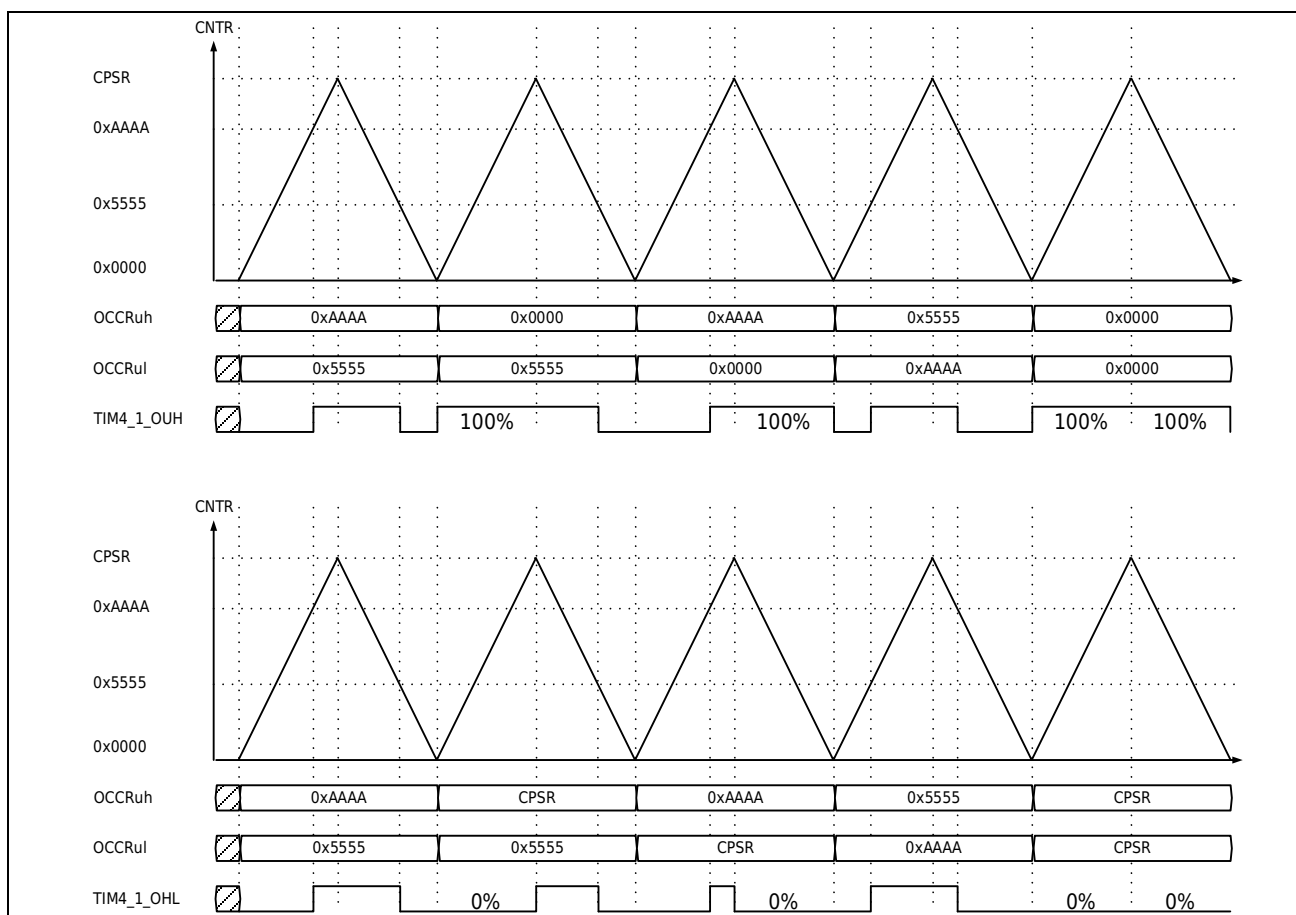


Figure 21-7 Triangle Wave Mode Waveform Output Instance

21.3.2 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.

21.3.2.1 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 CPSR register under the following conditions.

- When buffer is disabled (CCSR.BUFEN = 0), write data is immediately transferred from buffer register to CPSR register.
- Data is transferred from the buffer register to the CPSR register 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 21-8 As shown, the operation of modifying the count peak CPSR when sawtooth mode and the buffer function is disabled.

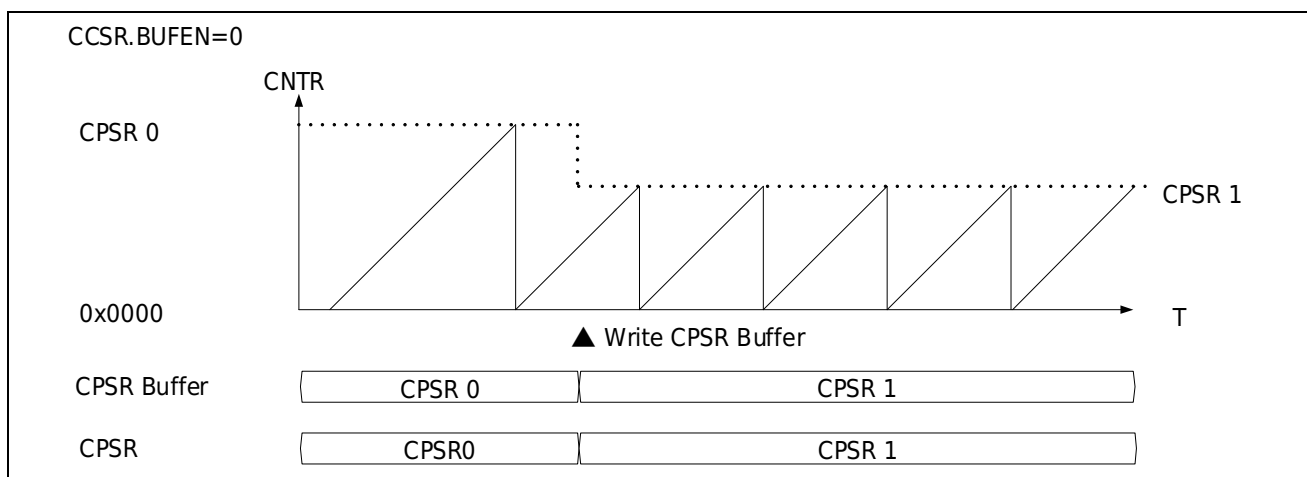


Figure 21-8 Modify the Sawtooth Count period when the Cache is Invalid

Note:

- When the buffer function is disabled, write data is immediately transferred from the buffer register to the CPSR counter, 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 21-9 As shown, the operation of modifying the count peak value CPSR when the sawtooth mode is enabled and the buffer function is enabled.

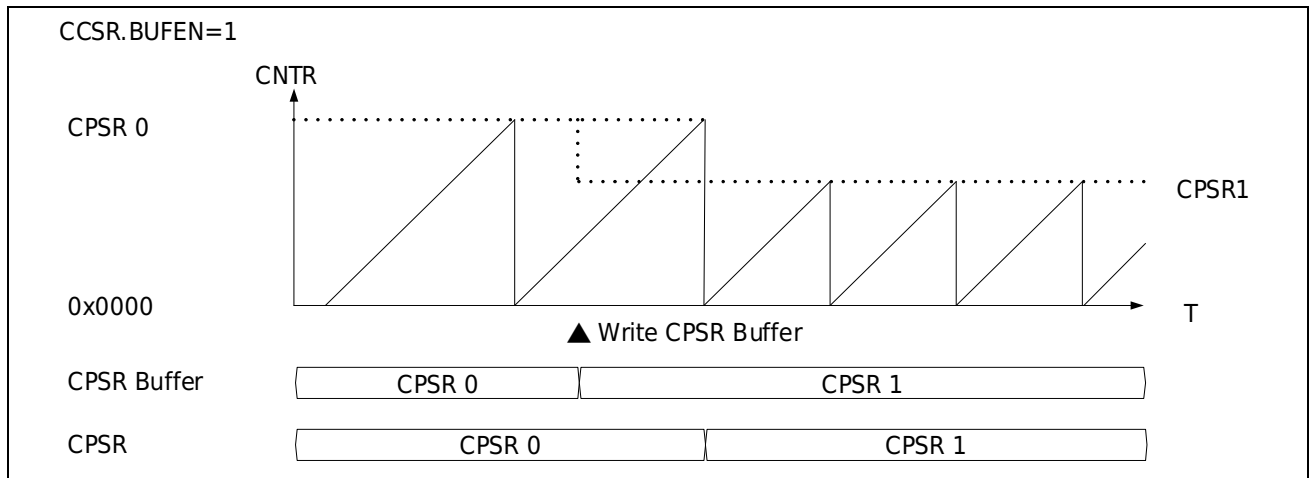


Figure 21-9 Modify the Sawtooth Count Period when the Cache is Enabled

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 21-10 As shown, the operation of modifying the count peak CPSR when the triangle wave mode and the buffer function is enabled.

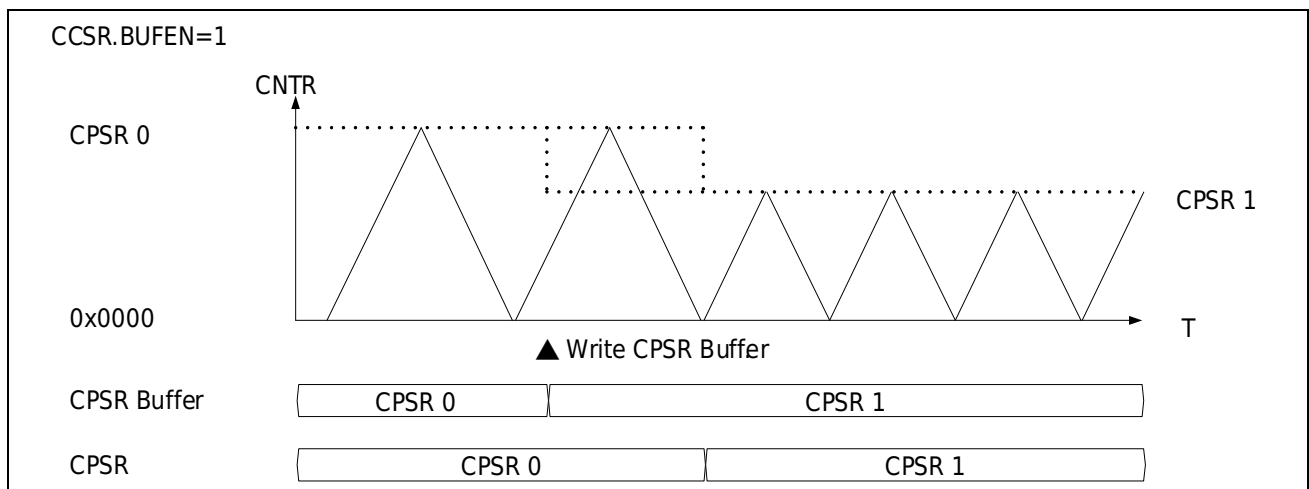


Figure 21-10 Modify the Triangular Wave Count Period when the Cache is Enabled

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 CPSR register when the counter stops or the next counter count value is "0x0000". The change in the counter period begins after the write operation completes the next counter period.

21.3.2.2 General 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 OCMR register 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 21-11 OCCR Buffered Data Transfer (Periodic Interval Response Link Disabled) shows the waveform (x = L or H) when the general output comparison OCCR buffer function is enabled, the counter is loaded at zero (OCER.CxBUFEN = 01), and the counter interval response link is disabled (OCER.LMCx = 0).

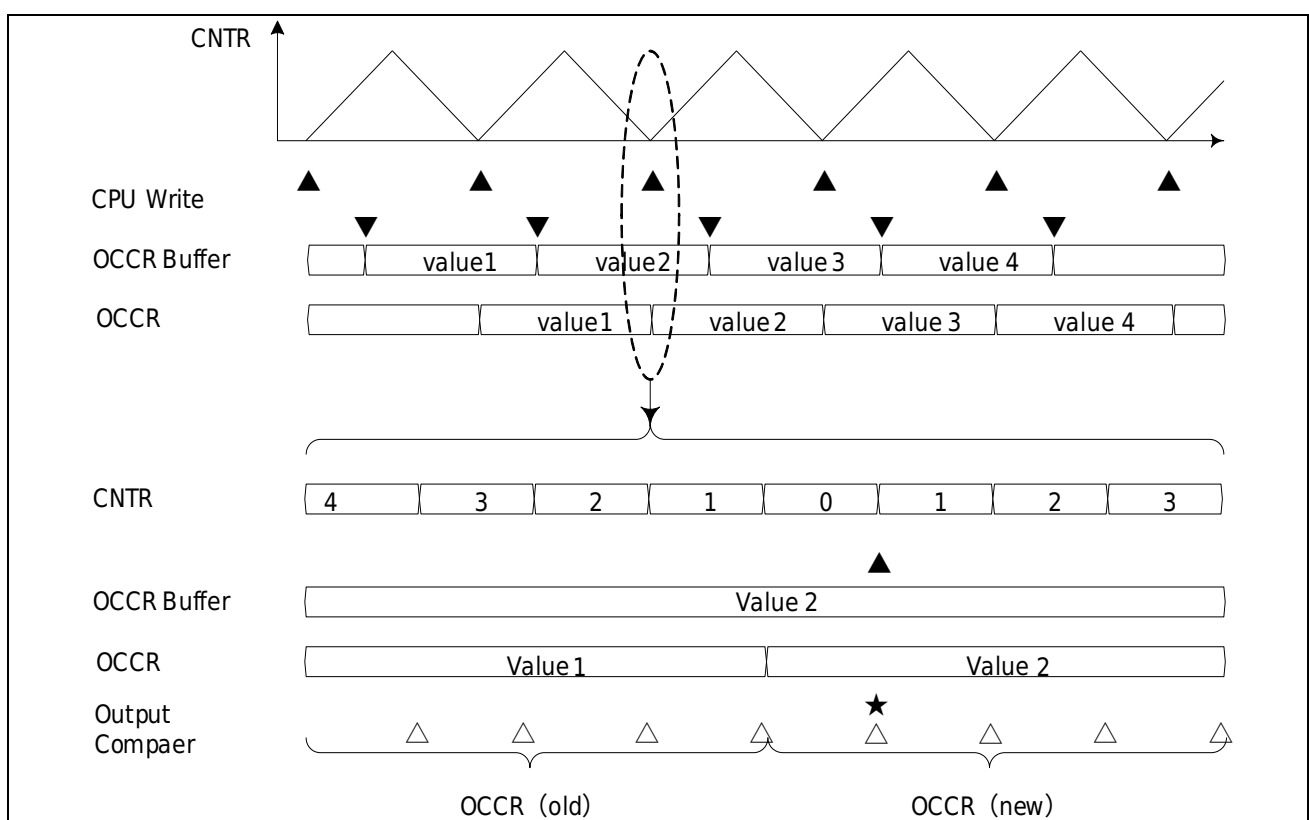


Figure 21-11 OCCR Buffer Data Transmission (When the Cycle Interval Response Link Disabled)

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 flag ▲ (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 21-12 OCCR Buffered Data Transfer (Periodic Interval Response Link Enabled) shows the waveform (x = L or H) when the general 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).

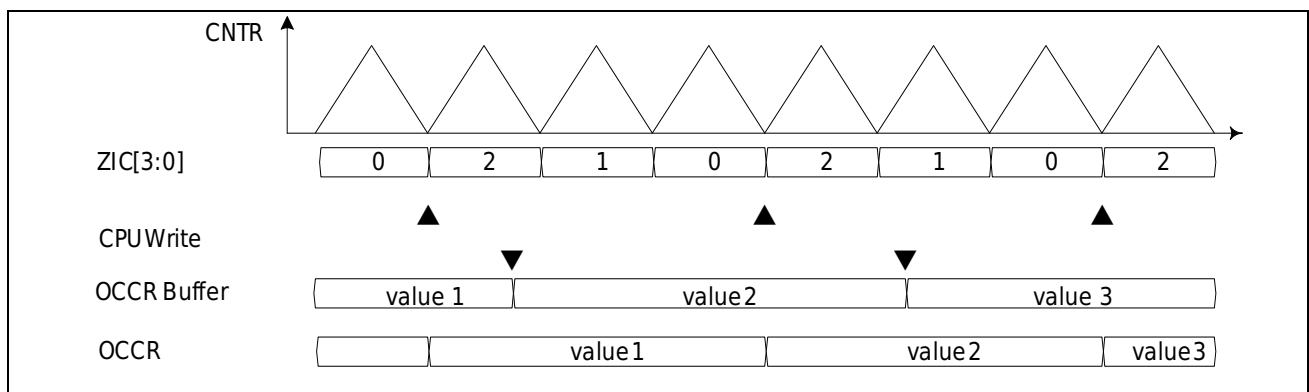


Figure 21-12 OCCR Buffer Data Transmission (Period Interval Response Link Enable)

The counter is in the triangular wave counting mode, the period interval counter (CVPR.ZIC) counts down from 2 to 0, and the underflow interrupt is generated at the moment of the flag ▲ (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 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 21-13 The case of changing the OCMR register value to produce different output waveforms, TIM4 _ < t > _ OUL, when the output comparison

registerOCCRuh and OCCRul remains unchanged is illustrated in Output comparison buffer data transfer (OCMR buffer enable).

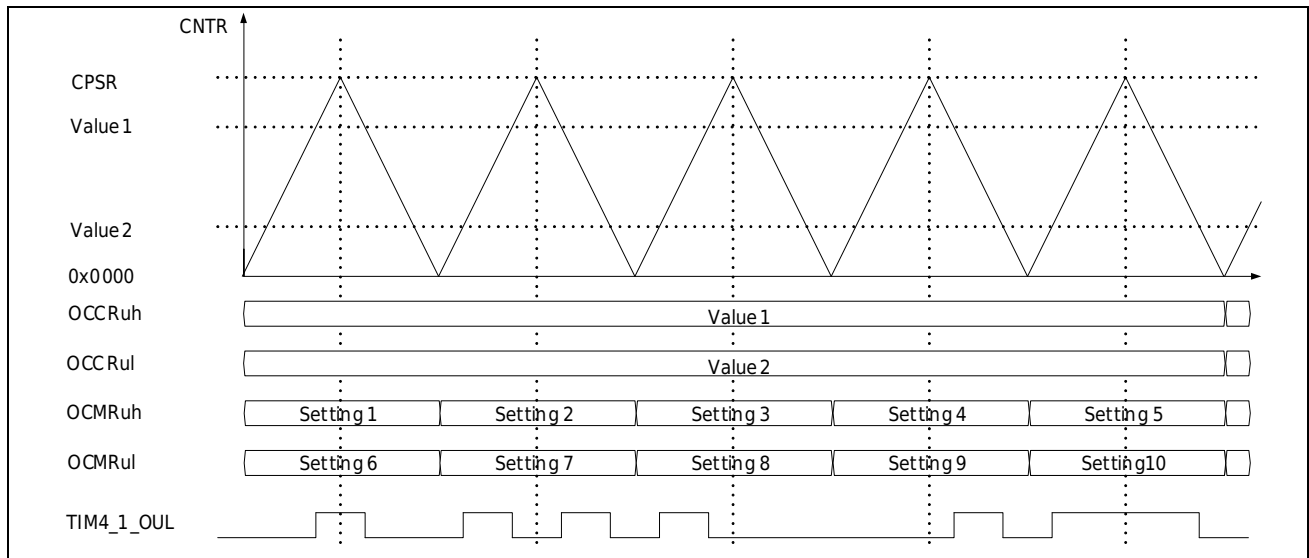


Figure 21-13 Output Compare Buffer Data Transfer (OCMR Buffer Enabled)

21.3.2.3 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 SCMR registers in the set counter state.

- a) When the counter interval response link transfer is disabled, the buffer transmission operation is only related to the counter state and is not affected by the counter interval counter.

Figure 21-14As 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.

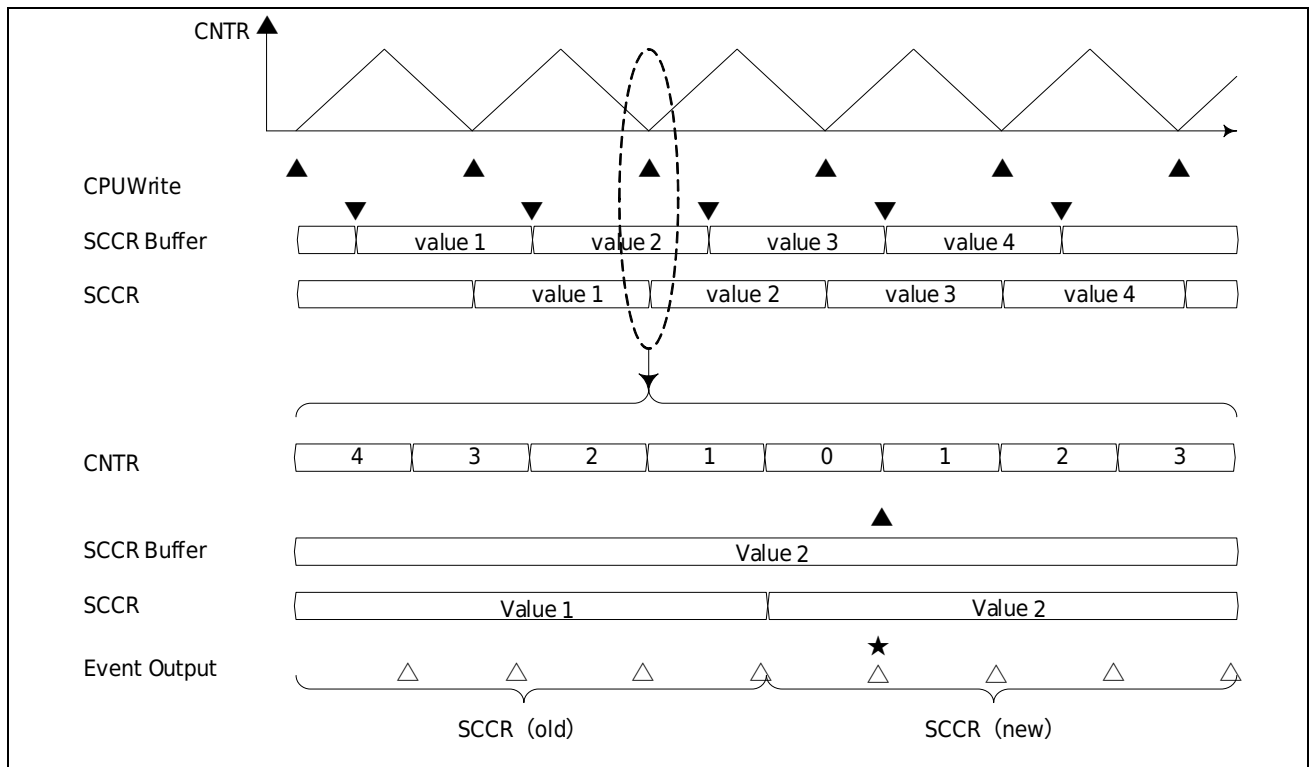


Figure 21-14 SCCR Buffer Transmission Operation (Period Interval Response Link Transmission Disabled)

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 flag ▲ (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 SCCR register, and the interrupt flag IRQZF is generated.

At △ time, the comparison operation is performed according to the set SCCR register 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 21-15, it is 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).

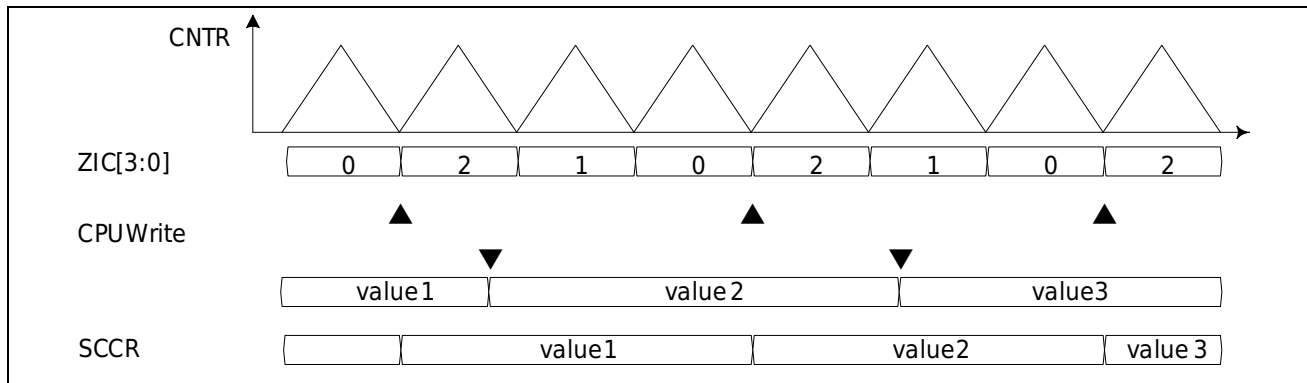


Figure 21-15 SCCR Buffer Transmission Operation (When Periodic Interval Response Link Transmission is Enabled)

The counter is in the triangular wave counting mode, the period interval counter (CVPR.ZIC) counts down from 2 to 0, and the underflow interrupt is generated at the moment of the flag ▲ (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.

21.3.3 General PWM Output

21.3.3.1 Independent PWM Output

The output states of OCCRxh, OCCRxI and OCMRxh and OCMRxI ports (x = u, v, w) can be set respectively in the pass-through mode (POCR.PWMMD = 00). At this point, the PWM output of each port is independently controlled. using the independent PWM output mode, you need to set the MOE and OExy bits (register PSCR) to enable the output of Timer4 (x=u , v , w , y=h , I). For detailed description, please refer to [PWM Status Control Register (TMR4_PSCR)]. And are instances of independent PWM outputs for unit 1's sawtooth and triangle waves, respectively.

Note:

- The pass-through mode refers to the internal output signal (in _opxh, in _opxl) generated by comparing the values of the common reference registers (OCCRxh, OCCRxI), which is directly output to the corresponding ports (TIM4 _ < t > _ OXH, TIM4 _ < t > _ OXL) (X = U, V, W, x = u, v, w).

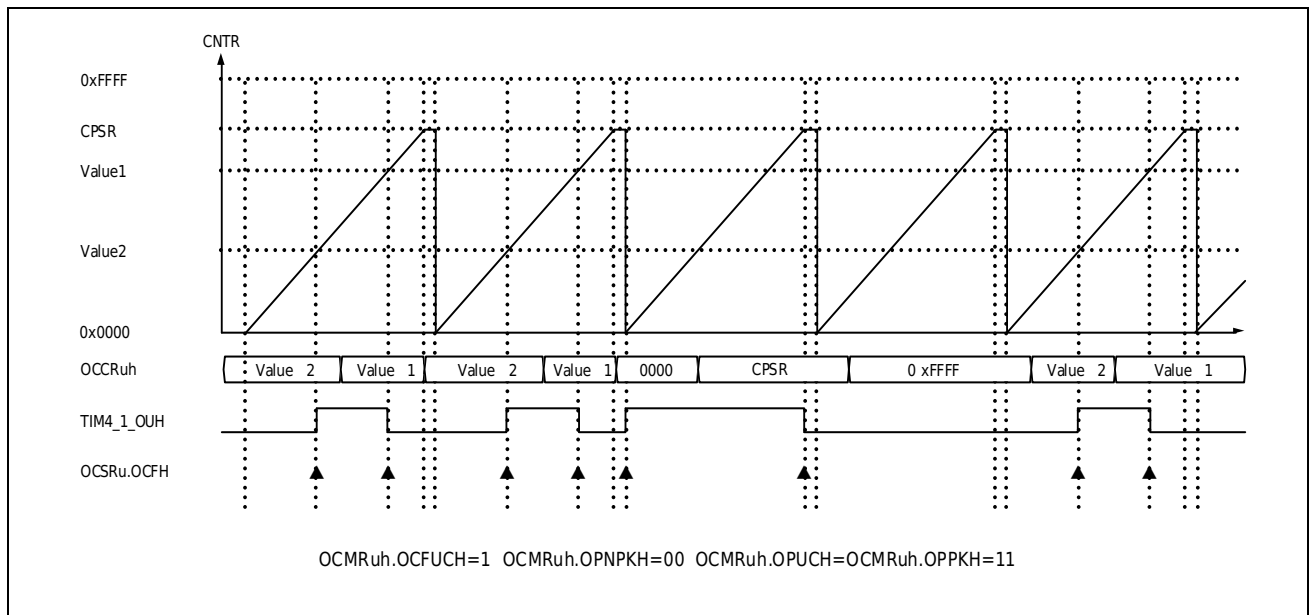


Figure 21-16 Sawtooth Wave Independent PWM Output Instance

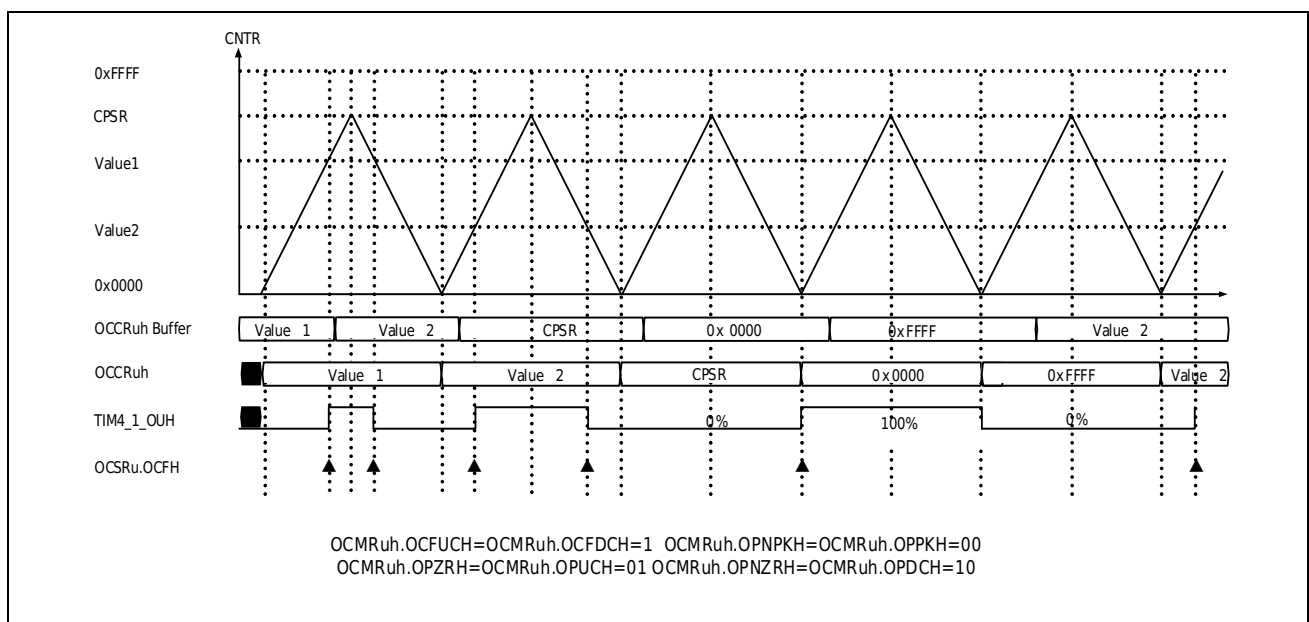


Figure 21-17 Triangle Wave Independent PWM Output Instance

21.3.3.2 Extended PWM output

The output state of TIM4_<t>_OXL in the pass-through mode (POCR.PWMMD = 00) can also be determined by the extension bit (bit32 ~ 16) in the OCMRxIregister. The extension bit is set to be correlated with the reference value of OCCRxh, thus realizing the extension PWM output (X = U, V, W, x = u, v, w) on the TIM4_<t>_OXL port. using the extended PWM output mode, you need to set the MOE and OExy bits (register PSCR) to enable the output of Timer4 (x=u , v , w , y=h , l). For detailed description, please refer to [PWM Status Control Register (TMR4_PSCR)]. Figure 21-18 Shown is the PWM output of TIM4_<t>_OUH and TIM4_<t>_OUL ports in this mode.

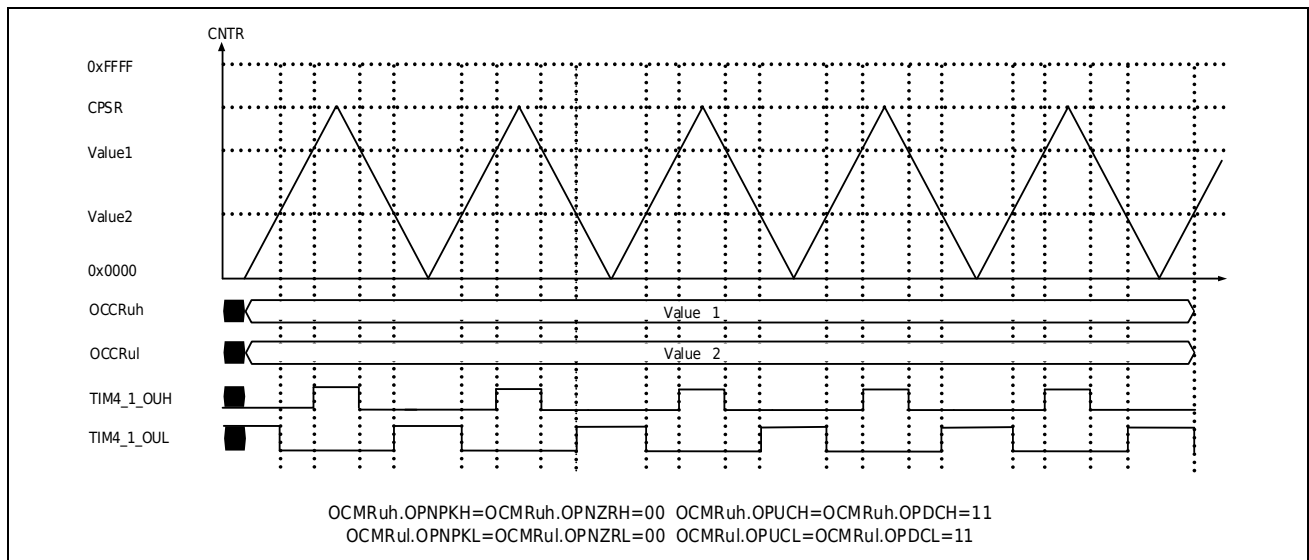


Figure 21-19 Software Complementary PWM Output

Realization of Complementary PWM Output by Hardware Setting

In the dead-zone timer mode (POCR.PWMMD = 01), the values of common reference register (OCCRxI) are matched to the internal output signal (in _opxI) and PWM dead-zone control register (PDAR/PDBR).

In this mode, the polarity of TIM4 _ < t > _ OXH port output is the same as in _ opxI, and the polarity of TIM4 _ < t > _ OXL port output is opposite to in _ opxI (X = U, V, W, x = u, v, w). Figure 21-20 Shown is an instance of complementary PWM output in dead-band timer mode.

If the rising edge of in _ opxI is detected, the output of TIM4 _ < t > _ OXL becomes low, the dead zone counter loads the set value of PDBRxregister and starts the decrement count. When the count value becomes 0x0000, the counter stops and TIM4 _ < t > _ OXH outputs high level. If an in _ opxI falling edge is detected and the TIM4 _ < t > _ OXH output becomes low, the dead-zone counter loads the set value of the PDARxregister and starts the decrement count. When the count value becomes 0x0000, the counter stops and the TIM4 _ < t > _ OXL output is high (X = U, V, W, x = u, v, w).

By setting PWM dead zone to control registerPDAR and PDBR, output rising and falling dead zone time can be set accordingly.

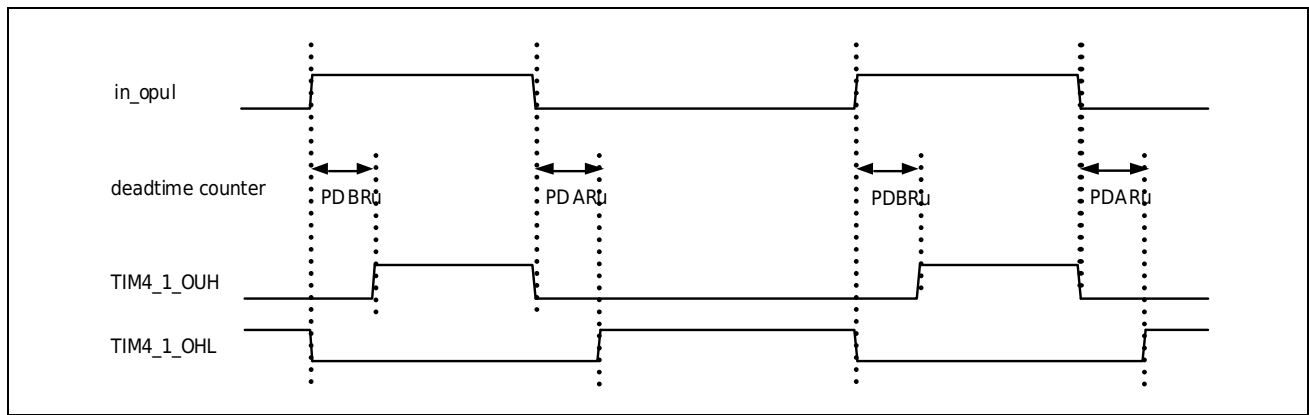


Figure 21-20 Complementary PWM Output in Dead-zone Timer Mode

When the pulse width of in_opxl is less than the dead zone set by PDBR, only the output of TIM4_< t >_ OXL becomes low. The condition for the output level of TIM4_< t >_ OXL to become high is after the dead zone set by PDARregister after the in_opxl falling edge. In this case, the TIM4_< t >_ OXH output remains low.

When the pulse width of in_opxl is less than the dead zone set by PDAR, only the output of TIM4_< t >_ OXH becomes low. TIM4_< t >_ OXH output level becomes high if it passes the dead zone set by PDBRregister after in_opxl rising edge. In this case, the TIM4_< t >_ OXL output will remain at low level (X = U, V, W, x = u, v, w). As shown in FigureFigure 21-21below.

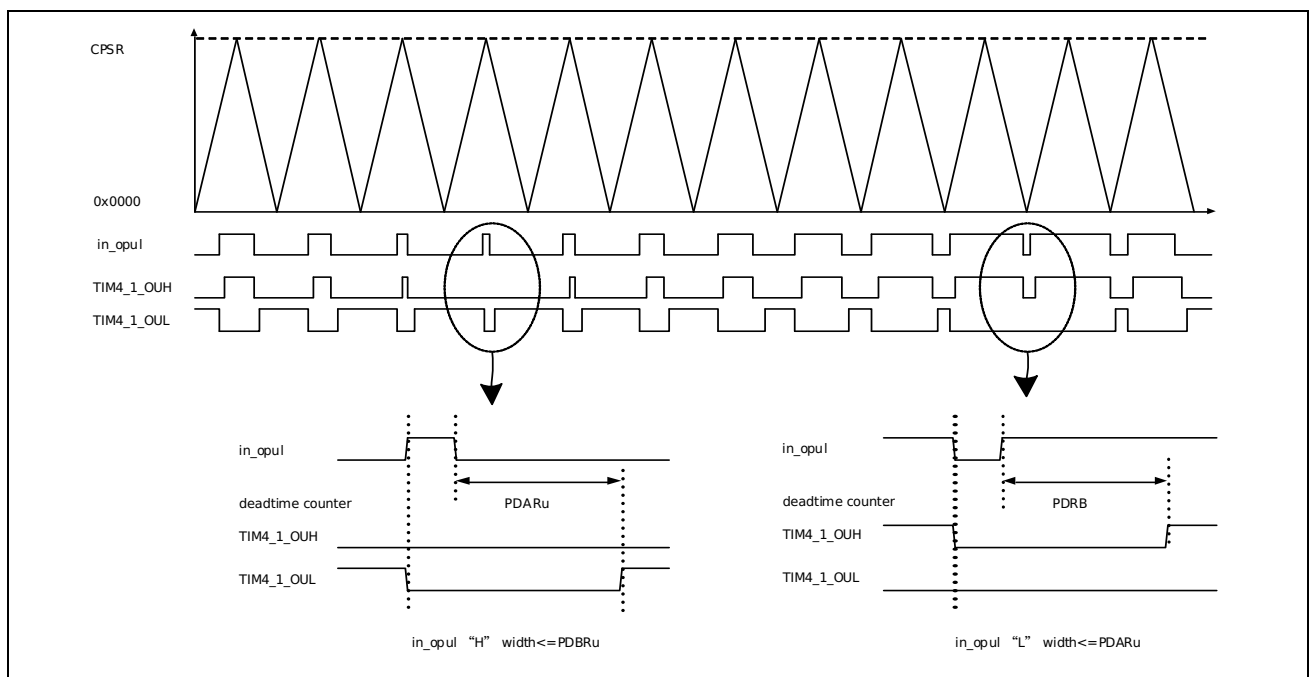


Figure 21-21 Waveform Output in Dead Zone Timer Mode when the Pulse Width is Abnormal

On the basis of the dead-zone output mode of the above hardware, the pulse width of in_opxl signal can be monitored to realize the filtering control of in_opxl signal. This dead-zone output implementation of in_opxl band pulse width filtering is called dead-zone counter filtering mode (POCR.PWMMD = 10) (x = u, v, w).

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 _ opxl is larger than the time set by registerPFSRn, the filter counter delays the output of in _ opxl signal after the time set by PFSR, and then produces the complementary PWM output (x = u, v, w) through the dead-time timer mode.

If the rising edge of the signal in _ opxl is detected, the filter counter loads the PFSRregister value and initiates the high level width of the measure in _ opxl. When the high level pulse width of in _ opxl is greater than the time set by the registerPFSR, after the time set by the PFSR, the TIM4 _ < t > _ OXL output becomes low, the dead zone counter loads the set value of the PDBRregister and initiates the decrement count, and when the count value becomes 0x0000, the counter stops and causes the TIM4 _ < t > _ OXH output to be high. If the signal in _ opxl falling edge is detected, the filter counter loads the PFSRregister value and initiates the low-level width of the measure in _ opxl. When the low-level pulse width of in _ opxl is greater than the time set by the registerPFSR, the TIM4 _ < t > _ OXH output becomes low after the time set by the PFSR, the dead zone counter loads the PDARregister setting value and initiates the decrement count. When the count value becomes 0x0000, the counter stops and causes the TIM4 _ < t > _ OXL output to be high. When the level pulse width of in _ opxl is less than the time set by registerPFSR, the output TIM4 _ < t > _ OXH and TIM4 _ < t > _ OXL will remain unchanged (X = U, V, W, x = u, v, w).

Figure 21-22 is an instance of complementary PWM output in dead-band counter filter mode.

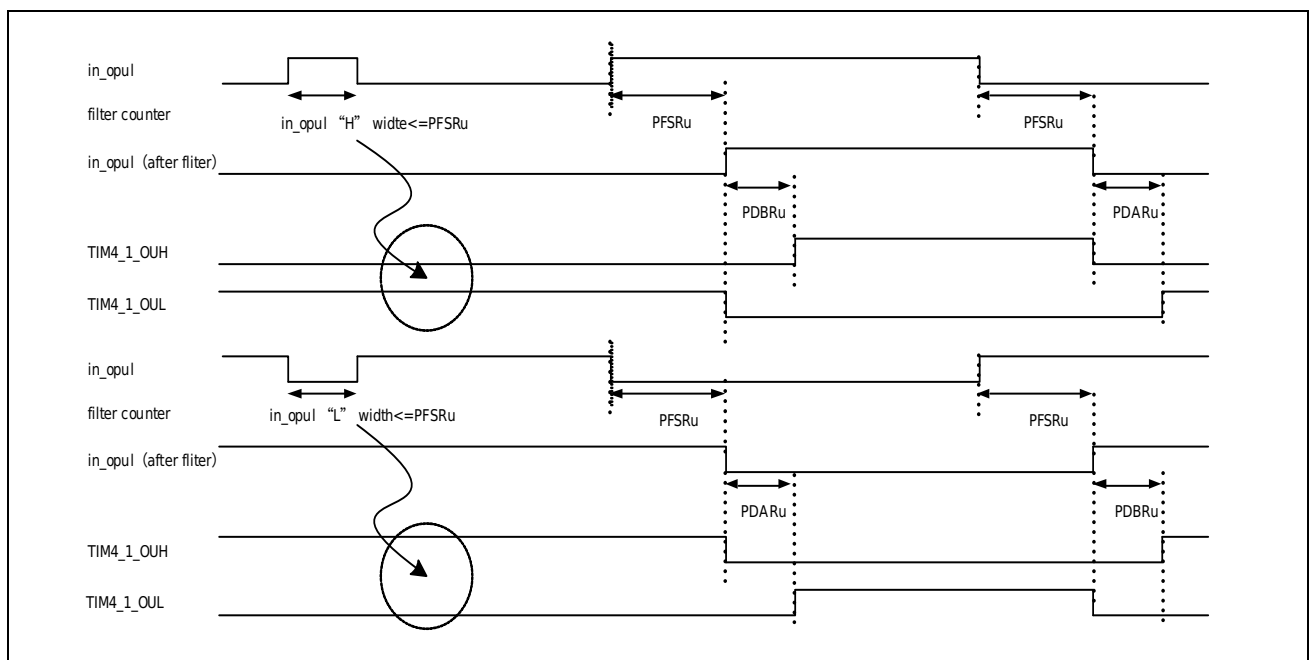


Figure 21-22 Complementary PWM Output in Dead-band Timer Filtering Mode

Note:

- using software or hardware complementary PWM output mode, you need to set the MOE and OExy bits (register PSCR) to enable the output of Timer4 (x=u , v , w , y=h , l). For detailed description, please refer to [PWM State Control Register].

21.3.4 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]="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 loads the value set by CVPR.PIM [3:0] at the beginning, when CVPR.PIC [3:0]="0". When 0", the overflow flag bit (CCSR.IRQPF) is set to "1".

As Figure 21-23 shown below, it is the timing diagram of setting IRQZF and IRQPF at the time of periodic interval response.

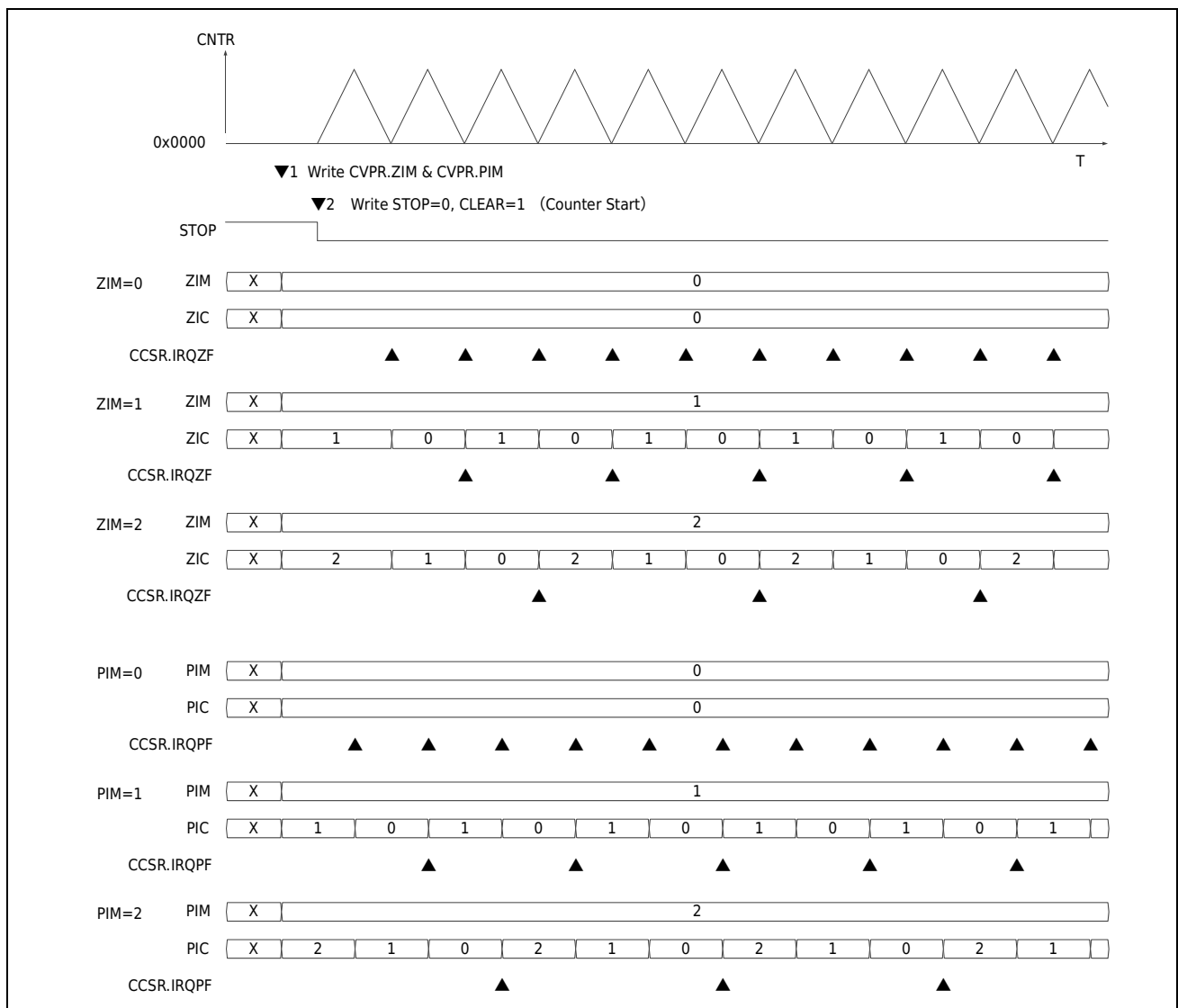


Figure 21-23 Cycle Interval Response Timing Diagram

▼1 When counter stops, it writes CVPR.ZIM and CVPR.PIM initialization values, which are immediately reflected in internal counter (CVPR.ZIC, CVPR.PIC).

▼2 Initialize and start the counter (STOP=0 and CLEAR=1). The counter starts counting up from zero after the bus is reset or CLEAR=1 is initialized from software. At this moment, the CCSR.IRQZF flag is not set immediately. After that, whenever the count value of the interrupt mask counter is 0x0000 and the count value of the counter is 0x0000 and CPSR, the flag ▲ is the moment when CCSR.IRQZF or CCSR.IRQPF is set.

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 21-24 shown below is the periodic interval response output graph of the dedicated event output signal.

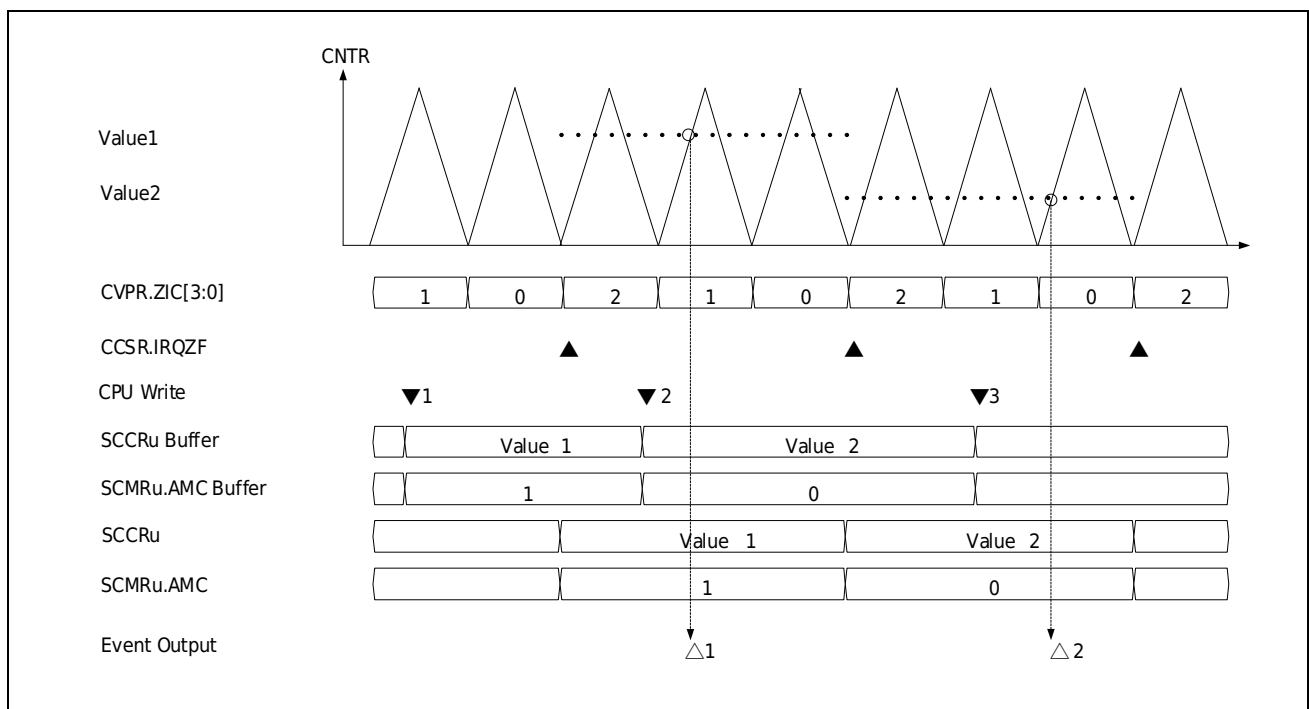


Figure 21-24 Dedicated Event Output Signal Periodic Interval Response Output

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 register SCCRu 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 $\Delta 1$.

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 $\Delta 2$.

21.3.5 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 three groups of 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 setting of PSCR.OSxy, the preset port state can be output high impedance, output low level or output high level (x=U, V, W, y=H, L).

For example, if PSCR.OSUH=01 is set, and an EMB event occurs during the normal output of the TIM4_<t>_OUH port of Timer4, the output on the TIM4_<t>_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 periods to read the actual value on the bus.

For detailed operation methods of EMB, please refer to the description in the [Emergency Brake Module (EMB)] section.

21.3.6 Monitoring Output

Each Timer4 unit has a PWM period output monitor port TIM4_<t>_PCT and a dedicated event output monitor port TIM4_<t>_ADSM. These monitoring ports are output to the outside, and users can flexibly control the application system based on the port change information.

The PWM period output monitoring port TIM4_<t>_PCT is used to monitor the current counting state of the counter. In triangle wave mode, each time the count reaches a peak or a trough, a flip occurs on the monitoring port TIM4_<t>_PCT. Figure 21-25 is an instance of a count direction signal.

The dedicated event output monitoring port TIM4_<t>_ADSM can be used to monitor the dedicated comparison events of the internal counter. 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 TIM4_<t>_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 TIM4_<t>_ADSM will be reset (the output is low).

It should be noted that when the configuration register TMR4_SCSRm enables both up-counting EVT and down-counting EVT (TMR4_SCSRm.UEN=1 and TMR4_SCSRm.DEN=1), the synchronization signal output to the external port is only valid for up-counting EVT.

Figure 21-26 is an instance of the synchronization signal output to the port in comparison start mode (SCSRm.EVTMS=0). When the upward counting EVT is valid, TIM4_<t>_ADSM is set, and when the counter underflows, TIM4_<t>_ADSM is cleared.

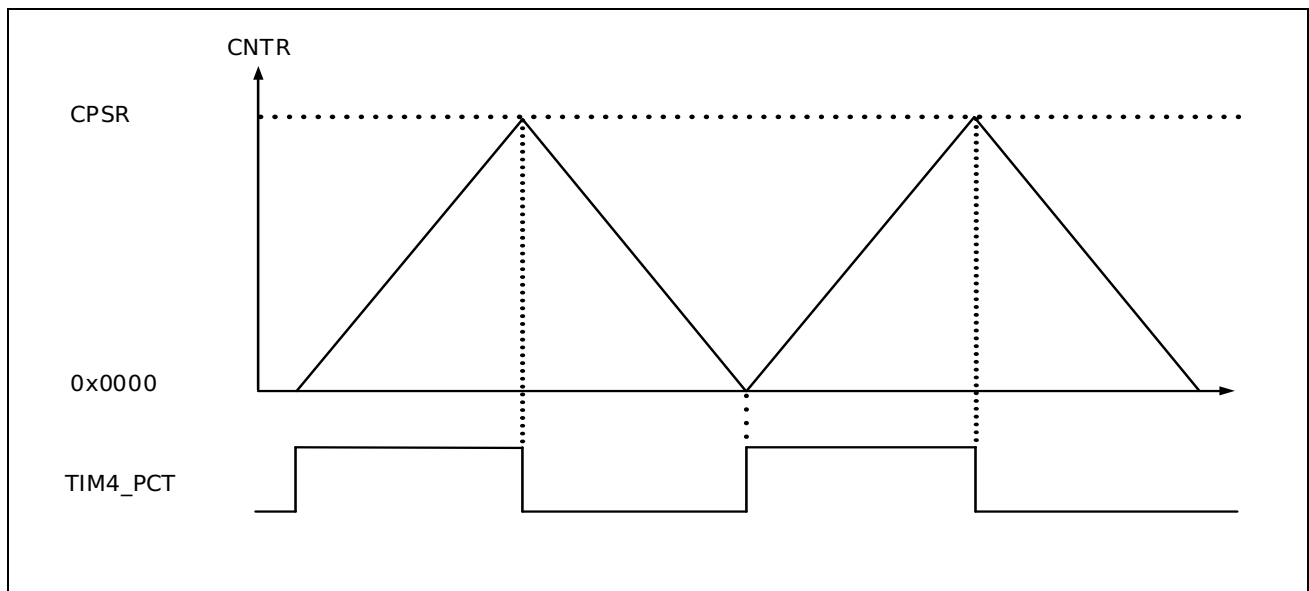


Figure 21-25 PWM Period Output Monitoring Instance

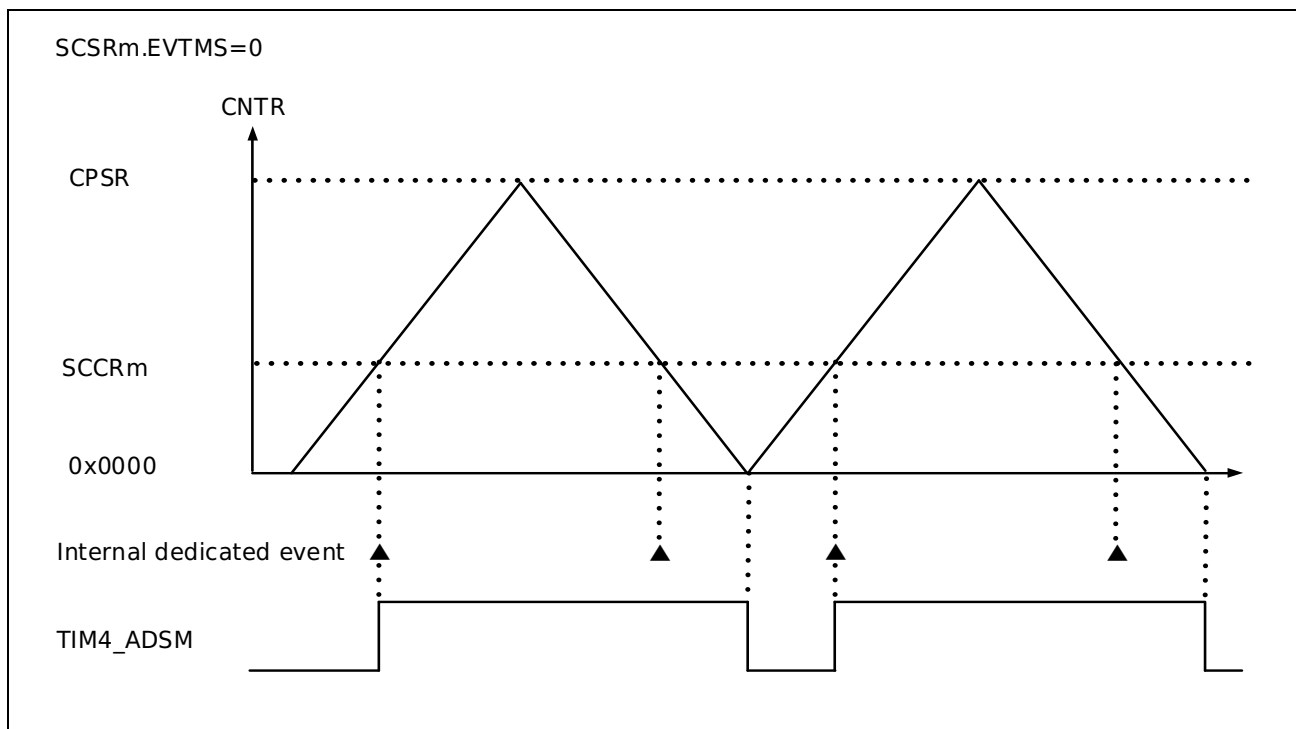


Figure 21-26 Compare Dedicated Event Output TIM4_<t>_ADSM Timing in Startup Mode

21.3.7 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 select register (TMR4_TRGSEL). For specific event correspondence, please refer to the INTC chapter. When using the internal hardware trigger function, the function clock control register (PWR_FCG0) peripheral circuit trigger function is required to enable position 1.

21.4 Interrupt and Event Description

21.4.1 General Count Compare Match Interrupt and Event

There are 6 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 interrupt, the corresponding interrupt request (TMR4_<t>_GCMmn, m=U, V, W; n=H, L) will also be triggered.

The general count comparison match event can be used to trigger other modules. For the event number, please refer to the [Interrupt Controller (INTC)] [Interrupt EventList] section.

21.4.2 Counting Period Match Interrupt and 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 period match interrupt (TMR4_<t>_GOVF and TMR4_<t>_GUDF) can be triggered at the corresponding time point.

The counting period match event can be used to trigger other modules. For the event number, please refer to the [Interrupt EventList] section.

21.4.3 Reload Count Match Interrupt and 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 comparison matches, the RCSR.RTIFU, RCSR.RTIFV, and RCSR.RTIFW bits in the overload control state register (RCSR) are set to 1, respectively. If RCSR.RTIDU, RCSR.RTIDV and RCSR.RTIDW interrupt masks are invalid, the corresponding heavy load count match interrupt request (TMR4_<t>_RLOm, m = U, V, W) 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 Controller (INTC)] [Interrupt EventList] section.

21.4.4 Dedicated Compare Match Event

The six special reference registers (SCCRm) of Timer4 produce six special event output signals, which can be used to select and trigger other modules, such as starting 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.ITEN0~5 is set to enable interrupt, the corresponding dedicated compare match interrupt request (TMR4_1_SCM0~5) will also be triggered.

The output of the event request signal can be selected to compare the start mode or delay start mode. When the start-up mode is compared (SCSR.EVTMS = 0), the specific event output signal is output directly after the SCCR count match event is generated. OCCRxh or OCCRxI (selected by the SCSR.EVTDS bit) are generated at the time of the delayed start-up mode (SCSR.EVTMS = 1); X = u, v, w). After SCCR, the output signal of the special event is output. The Figure 21-27 below shows an example of request output of the dedicated event output signal in the delayed start mode.

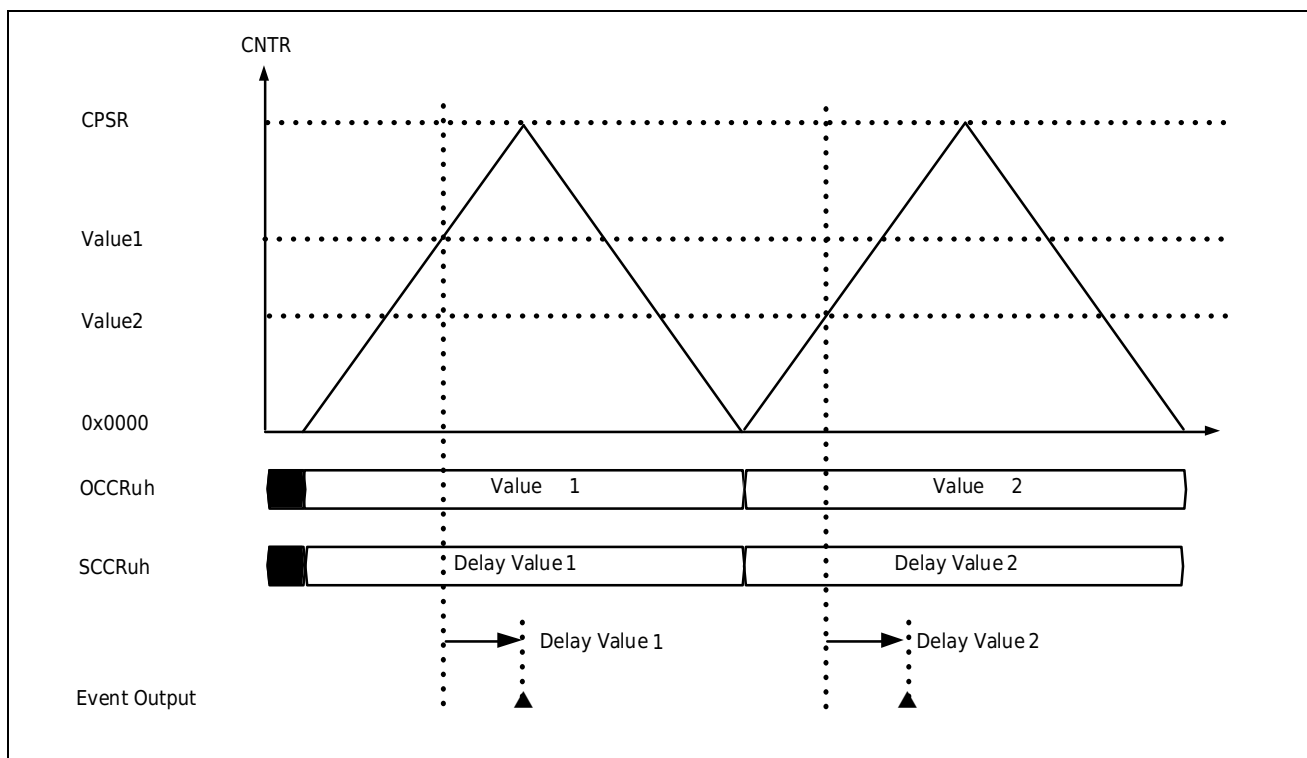


Figure 21-27 Output Timing of Dedicated Event Output Signal in Delayed Start Mode

Note:

- In a delay count run, if an OCCRx and counter match event occurs again, the delay counter reloads the count value and decrements the count again. Therefore, if the OCCRx 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.

21.5 Register Description

Table 21-3 shows the register list of the Timer4 module.

Table 21-3 Register List

BASE ADDR: 0x4003_8000 (U1)

Register name	Symbol	Offset	Bit Width	Reset Value
Register count value	TMR4_CNTR	0x0044	16	0x0000
Period reference register	TMR4_CPSR	0x0040	16	0xFFFF
Control status register	TMR4_CCSR	0x0048	16	0x0040
Valid period register	TMR4_CVPR	0x004A	16	0x0000
PWM status control register	TMR4_PSCR	0x004C	32	0x05550000
Generic baseline registerUH	TMR4_OCCRuh	0x0000	16	0x0000
Generic baseline registerUL	TMR4_OCCRuI	0x0004	16	0x0000
Generic baseline registerVH	TMR4_OCCRvh	0x0008	16	0x0000
Generic baseline registerVL	TMR4_OCCRvI	0x000C	16	0x0000
Generic baseline registerWH	TMR4_OCCRwh	0x0010	16	0x0000
Generic baseline registerWL	TMR4_OCCRwl	0x0014	16	0x0000
Generic control state registerU	TMR4_OCSRu	0x0018	16	0xFF00
Generic control state registerV	TMR4_OCSRv	0x001C	16	0xFF00
Generic control state registerW	TMR4_OCSRw	0x0020	16	0xFF00
Generic extension control registerU	TMR4_OCERu	0x001A	16	0x0000
Generic extension control registerV	TMR4_OCERv	0x001E	16	0x0000
Generic Extended Control RegisterW	TMR4_OCERw	0x0022	16	0x0000
Generic mode control registerUH	TMR4_OCMRuh	0x0024	16	0x0000
Generic mode control registerUL	TMR4_OCMRuI	0x0028	32	0x00000000
Generic mode control registerVH	TMR4_OCMRvh	0x002C	16	0x0000
Generic mode control registerVL	TMR4_OCMRvI	0x0030	32	0x00000000
Generic mode control registerWH	TMR4_OCMRwh	0x0034	16	0x0000
Generic mode control registerWL	TMR4_OCMRwl	0x0038	32	0x00000000
Dedicated baseline registerUH	TMR4_SCCRuh	0x00B0	16	0x0000
Special baseline registerUL	TMR4_SCCRuI	0x00B4	16	0x0000
Dedicated baseline registerVH	TMR4_SCCRvh	0x00B8	16	0x0000
Dedicated baseline registerVL	TMR4_SCCRvI	0x00BC	16	0x0000
Special baseline registerWH	TMR4_SCCRwh	0x00C0	16	0x0000
Dedicated baseline registerWL	TMR4_SCCRwl	0x00C4	16	0x0000
Special control state registerUH	TMR4_SCSRuh	0x00C8	16	0x0000
Special control status registerUL	TMR4_SCCRuI	0x00CC	16	0x0000

Register name	Symbol	Offset	Bit Width	Reset Value
Special control state registerVH	TMR4_SCSRvh	0x00D0	16	0x0000
Special control state registerVL	TMR4_SCSRvl	0x00D4	16	0x0000
Special control state registerWH	TMR4_SCSRwh	0x00D8	16	0x0000
Special control state registerWL	TMR4_SCSRwl	0x00DC	16	0x0000
Special mode control registerUH	TMR4_SCMRuh	0x00CA	16	0xFF00
Special mode control registerUL	TMR4_SCMRul	0x00CE	16	0xFF00
Special mode control registerVH	TMR4_SCMRvh	0x00D2	16	0xFF00
Special mode control registerVL	TMR4_SCMRvl	0x00D6	16	0xFF00
Special mode control registerWH	TMR4_SCMRwh	0x00DA	16	0xFF00
Special mode control registerWL	TMR4_SCMRwl	0x00DE	16	0xFF00
Dedicated Extended Control Register	TMR4_SCER	0x00E0	16	0xFF00
PWM basic control register U	TMR4_POCRu	0x0098	16	0xFF00
PWM basic control register V	TMR4_POCRv	0x009C	16	0xFF00
PWM Basic Control RegisterW	TMR4_POCRw	0x00A0	16	0xFF00
PWM filtering control registerU	TMR4_PFSRu	0x0082	16	0x0000
PWM Filtering Control RegisterV	TMR4_PFSRv	0x008A	16	0x0000
PWM Filtering Control RegisterW	TMR4_PFSRw	0x0092	16	0x0000
PWM dead zone control registerAU	TMR4_PDARu	0x0084	16	0x0000
PWM dead zone control registerBU	TMR4_PDBRu	0x0086	16	0x0000
PWM dead zone control registerAV	TMR4_PDARv	0x008C	16	0x0000
PWM dead zone control registerBV	TMR4_PDBRv	0x008E	16	0x0000
PWM dead zone control registerAW	TMR4_PDARw	0x0094	16	0x0000
PWM dead zone control registerBW	TMR4_PDBRw	0x0096	16	0x0000
Overload control status register	TMR4_RCSR	0x00A4	16	0x0000
Dedicated interrupt control register	TMR4_SCIR	0x00E4	16	0xFF00
Dedicated flag control register	TMR4_SCFR	0x00E8	16	0xFF00

Note:

- In the detailed description of the following registers, $m=uh, ul, vh, vl, wh, wl, n=u, v, w$. The registers pointed to by m respectively correspond to the output control of ports $TIM4\_<t>_{OUH}$, $TIM4\_<t>_{OUL}$, $TIM4\_<t>_{OVH}$, $TIM4\_<t>_{OVL}$, $TIM4\_<t>_{OWH}$, $TIM4\_<t>_{OWL}$, etc.; The registers indicated by n correspond to the output control of ports $TIM4\_<t>_{OUx}$, $TIM4\_<t>_{OVx}$, $TIM4\_<t>_{OWx}$, etc. when the function is realized, where $x=H$ or L , and the specific control of H or L is in these There is a corresponding symmetry bit in the register.

21.5.1 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	Flag	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											R/W	

21.5.2 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	Flag	Bit Name	Function											Read and write	
b15~b0	CPSR [15:0]	General periodic reference value	Counter count period value Note: When reading data from the local area, it is not the value of the buffer register, but the value of the CPSRregister											R/W	

21.5.3 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	-	HST	-	IRQ ZF	IRQ ZEN	IRQ PF	IRQ PEN	BUF EN	STOP	MODE	CLEAR	CKDIV [3:0]			

Bit	Flag	Bit Name	Function	Read and write
b15	ECKEN	Timer Selection	0: Internal PCLK1 clock 1: External TIM4 _ < t > _ CLK port input clock Note 1: The bit is set when counter stops Note 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	R/W
b14	Reserved	-	Read as "0", write as "0"	R/W
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	R/W
b12	Reserved	-	Read as "0", write as "0"	R/W
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 Note 2: When the counter is reset by the bus or written to CLEAR = '1', the IRQZF bit will not be set	R/W
B10	IRQZEN	Underflow interrupt enable	0: Disable IRQZF from producing interrupt to CPU 1: Allow IRQZF to generate interrupt to CPU	R/W
b9	IRQPF	Overflow state	0: No count overflow 1: Occurrence count overflow Note 1: When the period interval response function is used, the setting condition of this bit is set by the period interval counter set by CVPR. Note 2: When the counter is reset by the bus or written to CLEAR = '1', the IRQZF bit will not be set	R/W
b8	IRQPEN	Overflow interrupt enable	0: Disable IRQPF from producing interrupt to CPU 1: Allow IRQPF to generate interrupt to CPU	R/W
b7	BUFEN	Cache Enable	0: Disable CPSR caching 1: Enable CPSR caching	R/W
b6	STOP	Counter enable	0: Counter Startup 1: Counter stopped	R/W
b5	MODE	Waveform Mode	0: Sawtooth mode (counting up only) 1: Triangular wave mode	R/W
b4	CLEAR	Counter reset	0: No operation 1: Counter zeroing Note: This bit is always read at 0	R/W
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	R/W

21.5.4 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		Flag		Bit Name		Function						Read and write			
b15~b12		PIC [3:0]		Overflow interrupt masking state		Number of currently shielded overflow interrupts						R			
b11~b8		ZIC [3:0]		Underflow interrupt masking state		Number of currently shielded underflows interrupt						R			
b7~b4		PIM [3:0]		Overflow interrupt masking setting		Set the number of overflow interrupts for masking						R/W			
b3~b0		ZIM [3:0]		Overflow interrupt masking setting		Set the number of underflow interrupts of shielded						R/W			

21.5.5 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	Flag	Bit Name		Function		Read and write									
b15~b0	OCCR [15:0]	General comparison reference value		General comparison reference 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		R/W									

21.5.6 General Control State Register (TMR4 _ OCSRn)

Reset value: 0xFF00

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								OCFL	OCFH	OCIEL	OCIEH	OCPL	OCPL	OCEL	OCEH
Bit	Flag	Bit Name	Function											Read and write	
b15~b8	Reserved	-	Read as "1", write as "1"											R/W	
b7	OCFL	Count Matching L	0: Counter count value is not equal to OCCRxL set value 1: Counter count is equal to OCCRxL set (x=u, v, w) Note: The bit must be valid when OCEL = 1											R/W	
b6	OCFH	Count Matching H	0: Counter count value is not equal to OCCRxH set value 1: Counter count is equal to OCCRxH set (x=u, v, w) Note: The bit must be valid when OCEH = 1											R/W	
b5	OCIEL	Count Matching Interrupt Enable	0: OCFL does not interrupt when set 1: Interrupt occurs when OCFL is set											R/W	
b4	OCIEH	Count Matching Hinterrupt Enable	0: No interrupt occurs when OCFH is set 1: Interrupt occurs when OCFH is set											R/W	
b3	OCPL	Port state L when comparison output is invalid	0: Output low level on TIM4 _ < t > _ OxL when OCEL = 0 1: High Output Level on TIM4 _ < t > _ OxL at OCEL = 0 (x=U, V, W)											R/W	
b2	OCPL	Port state H when the comparison output is invalid	0: Output Low Level on TIM4 _ < t > _ OxH when OCEH = 0 1: High Output Level on TIM4 _ < t > _ OxH at OCEH = 0 (x=U, V, W)											R/W	
b1	OCEL	Port Output Selection L	0: The comparison output is invalid. The port status of TIM4 _ < t > _ OxL is determined by OCPL. 1: The output is valid. The status of TIM4 _ < t > _ OxL port is determined by the setting of OCMRyl and the status of OCFL. (x=U, V, W, y=u, v, w)											R/W	
b0	OCEH	Port Output Selection H	0: The comparison output is invalid. The status of TIM4 _ < t > _ OxH port is determined by OCPH. 1: The output is valid. The status of TIM4 _ < t > _ OxH port is determined by the setting of OCMRyh and the status of OCFH. (x=U, V, W, y=u, v, w)											R/W	

21.5.7 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
-	-	MCEC L	MCEC H	LMM L	LMM H	LMC L	LMC H	MLBUF EN [1:0]	MHBUF EN [1:0]	CLBUF EN [1:0]	CHBUF EN [1:0]				
Bit		Flag		Bit Name		Function		Read and write							
b15~b14		Reserved		-		Read as "0", write as "0"		R/W							
b13		MCECL		Extended control enable L		0: Extended control function to disable matching of count values with OCCRxL 1: Extended control function to enable the matching condition between the count value and OCCRxL (x=u, v, w) Note 1: This bit can only be modified if the output comparison function is invalid (OCSR.OCEL = 0) Note 2 : Please refer to the notes of TMR4_OCMRm register description for details.		R/W							
b12		MCECH		Extended control enable H		0: Extension control function to disable matching condition between count value and OCCRxh 1: Extended control function to enable the matching condition between the count value and OCCRxh (x=u, v, w) Note 1: This bit can only be modified if the output comparison function is invalid (OCSR.OCEH = 0) Note 2 : Please refer to the notes of TMR4_OCMRm register description for details.		R/W							
b11		LMML		Cycle Interval Response link L		0: Interval response function link is invalid, OCMRxL cache transfer is determined by MLBUFEN setting 1: The periodic interval response function link is valid, and the buffer transmission of OCMRxL 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) (x=u, v, w)		R/W							
b10		LMMH		Cycle Interval Response link H		0: Interval response function link is invalid, OCMRxh cache transfer is determined by MLBUFEN setting 1: The periodic interval response function link is valid, and the buffer transmission of OCMRxh 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) (x=u, v, w)		R/W							
b9		LMCL		Cycle Interval Response link L		0: The interval response link is invalid. The cache transfer of OCCRxL is determined by MLBUFEN 1: The periodic interval response function link is valid, and the buffer transmission of OCCRxL 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) (x=u, v, w)		R/W							
b8		LMCH		Cycle Interval Response link H		0: The interval response link is invalid. The cache transfer of OCCRxh is determined by MLBUFEN 1: The periodic interval response function link is valid, and the buffer transmission of OCCRxh 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) (x=u, v, w)		R/W							
b7~b6		MLBUFEN [1:0]		OCMRxL Cache Transport		00: OCMRxL cache register values are written directly to OCMRxL 01: OCMRxL cache register values are written to OCMRxL at count underflow 10: The value of the OCMRxL cache register is written to OCMRxL when the count overflows 11: The value of the OCMRxL cache register is written to OCMRxL on count underflow or overflow (x=u, v, w)		R/W							
b5~b4		MHBUFEN [1:0]		OCMRxh Cache Transfer		00: OCMRxh Cache register value written directly to OCMRxh 01: OCMRxh Cache register values are written to OCMRxh on count underflow 10: OCMRxh Cache register values are written to OCMRxh on count overflow		R/W							

			11: OCMRxh Cache register values are written to OCMRxh on count underflow or overflow (x=u, v, w)	
b3~b2	CLBUFEN [1:0]	OCCRxI Cache Transport	00: OCCRxI cache register values are written directly to OCCRxI 01: OCCRxI cache register values are written to OCCRxI at count underflow 10: The value of the OCCRxI cache register is written to OCCRxI when the count overflows 11: The value of the OCCRxI cache register is written to OCCRxI on count underflow or overflow (x=u, v, w)	R/W
b1~b0	CHBUFEN [1:0]	OCCRxh Cache Transport	00: OCCRxh Cache register value written directly to OCCRxh 01: OCCRxh Cache register values are written to OCCRxh on count underflow 10: OCCRxh Cache register values are written to OCCRxh on count overflow 11: OCCRxh Cache register values are written to OCCRxh on count underflow or overflow (x=u, v, w)	R/W

21.5.8 General 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						

Note: This register bit description is used for OCMRuh, OCMRvh, OCMRwh

Bit	Flag	Bit Name	Function	Read and write
b15~b14	OPNZRH [1:0]	Underflow point port state H	Condition: Count underflow & OCCRxh count mismatch (x = u, v, w) 00: The TIM4 _ < t > _ OyH port position remains unchanged when the condition is satisfied 01: TIM4 _ < t > _ OyH port bit output high level when conditions are met 10: TIM4 _ < t > _ OyH port bit output low level when condition is satisfied 11: TIM4 _ < t > _ OyH port bit output inversion when conditions are met (y=U, V, W)	R/W
b13~b12	OPNPKH [1:0]	Overflow point port state H	Condition: Count overflow & OCCRxh count mismatch (x = u, v, w) 00: The TIM4 _ < t > _ OyH port position remains unchanged when the condition is satisfied 01: TIM4 _ < t > _ OyH port bit output high level when conditions are met 10: TIM4 _ < t > _ OyH port bit output low level when condition is satisfied 11: TIM4 _ < t > _ OyH port bit output inversion when conditions are met (y=U, V, W)	R/W
b11~b10	OPZRH [1:0]	Underflow point port state H	Condition: Count underflow & OCCRxh count match (x = u, v, w) 00: The TIM4 _ < t > _ OyH port position remains unchanged when the condition is satisfied 01: TIM4 _ < t > _ OyH port bit output high level when conditions are met 10: TIM4 _ < t > _ OyH port bit output low level when condition is satisfied 11: TIM4 _ < t > _ OyH port bit output inversion when conditions are met (y=U, V, W)	R/W
b9~b8	OPUCH [1:0]	Up Count Port State H	Condition: Counter Up Count & OCCRxh Count Match (x = u, v, w) 00: The TIM4 _ < t > _ OyH port position remains unchanged when the condition is satisfied 01: TIM4 _ < t > _ OyH port bit output high level when conditions are met 10: TIM4 _ < t > _ OyH port bit output low level when condition is satisfied 11: TIM4 _ < t > _ OyH port bit output inversion when conditions are met (y=U, V, W)	R/W
b7~b6	OPPKH [1:0]	Overflow point port state H	Condition: Count Overflow & OCCRxh Count Match (x = u, v, w) 00: The TIM4 _ < t > _ OyH port position remains unchanged when the condition is satisfied 01: TIM4 _ < t > _ OyH port bit output high level when conditions are met 10: TIM4 _ < t > _ OyH port bit output low level when condition is satisfied 11: TIM4 _ < t > _ OyH port bit output inversion when conditions are met (y=U, V, W)	R/W
b5~b4	OPDCH [1:0]	Down count port state H	Condition: Counter Down Count & OCCRxh Count Match (x = u, v, w) 00: The TIM4 _ < t > _ OyH port position remains unchanged when the condition is satisfied 01: TIM4 _ < t > _ OyH port bit output high level when conditions are met	R/W

			10: TIM4_ < t > _OyH port bit output low level when condition is satisfied 11: TIM4_ < t > _OyH port bit output inversion when conditions are met (y=U, V, W)	
b3	OCFZRH	Underflow point OCFH state H	Condition: Count underflow & OCCRxh count match (x = u, v, w) 0: OCSR.OCFH bit remains unchanged when the condition is satisfied 1: OCSR.OCFH position bit when conditions are met	R/W
b2	OCFUCH	Up count OCFH state H	Condition: Counter Up Count & OCCRxh Count Match (x = u, v, w) 0: OCSR.OCFH bit remains unchanged when the condition is satisfied 1: OCSR.OCFH position bit when conditions are met	R/W
b1	OCFPKH	Overflow point OCFH state H	Condition: Count Overflow & OCCRxh Count Match (x = u, v, w) 0: OCSR.OCFH bit remains unchanged when the condition is satisfied 1: OCSR.OCFH position bit when conditions are met	R/W
b0	OCFDCH	Down count OCFH state H	Condition: Counter Down Count & OCCRxh Count Match (x = u, v, w) 0: OCSR.OCFH bit remains unchanged when the condition is satisfied 1: OCSR.OCFH position bit when conditions are met	R/W

Reset value: 0x00000000

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						

Note: This register bit description is used for OCMRul, OCMRvl, OCMRwl

Bit	Flag	Bit Name	Function	Read and write
b31~b30	EOPNZRL [1:0]	Extended underflow point port state L	Condition: Count underflow & OCCRxL count does not match & OCCRxh count matches (x=u, v, w) 00: TIM4_ < t > _OyL port position remains unchanged when the condition is satisfied 01: TIM4_ < t > _OyL port bit output high level when conditions are met 10: TIM4_ < t > _OyL port bit output low level when condition is satisfied 11: TIM4_ < t > _OyL port bit output inversion when conditions are met (y=U, V, W)	R/W
b29~b28	EOPNPKL [1:0]	Expanded overflow point port state L	Condition: Count Overflow & OCCRxL Count Mismatch & OCCRxh Count Match (x = u, v, w) 00: TIM4_ < t > _OyL port position remains unchanged when the condition is satisfied 01: TIM4_ < t > _OyL port bit output high level when conditions are met 10: TIM4_ < t > _OyL port bit output low level when condition is satisfied 11: TIM4_ < t > _OyL port bit output inversion when conditions are met (y=U, V, W)	R/W
b27~b26	EOPZRL [1:0]	Extended underflow point port state L	Condition: Count underflow & OCCRxL count match & OCCRxh count match (x = u, v, w) 00: TIM4_ < t > _OyL port position remains unchanged when the condition is satisfied 01: TIM4_ < t > _OyL port bit output high level when conditions are met 10: TIM4_ < t > _OyL port bit output low level when condition is satisfied 11: TIM4_ < t > _OyL port bit output inversion when	R/W

			conditions are met (y=U, V, W)	
b25~b24	EOPUCL [1:0]	Expansion up count port status L	Condition: Counter Up Count & OCCRxI Count Match & OCCRxh Count Match (x = u, v, w) 00: TIM4_ < t > _OyL port position remains unchanged when the condition is satisfied 01: TIM4_ < t > _OyL port bit output high level when conditions are met 10: TIM4_ < t > _OyL port bit output low level when condition is satisfied 11: TIM4_ < t > _OyL port bit output inversion when conditions are met (y=U, V, W)	R/W
b23~b22	EOPPKL [1:0]	Expanded overflow point port state L	Condition: Count Overflow & OCCRxI Count Match & OCCRxh Count Match (x = u, v, w) 00: TIM4_ < t > _OyL port position remains unchanged when the condition is satisfied 01: TIM4_ < t > _OyL port bit output high level when conditions are met 10: TIM4_ < t > _OyL port bit output low level when condition is satisfied 11: TIM4_ < t > _OyL port bit output inversion when conditions are met (y=U, V, W)	R/W
b21~b20	EOPDCL [1:0]	Expansion Down Count Port Status L	Condition: Counter Down Count & OCCRxI Count Match & OCCRxh Count Match (x = u, v, w) 00: TIM4_ < t > _OyL port position remains unchanged when the condition is satisfied 01: TIM4_ < t > _OyL port bit output high level when conditions are met 10: TIM4_ < t > _OyL port bit output low level when condition is satisfied 11: TIM4_ < t > _OyL port bit output inversion when conditions are met (y=U, V, W)	R/W
b19~b18	EOPNUCL [1:0]	Expansion up count port status L	Condition: Counter up count & OCCRxI count mismatch & OCCRxh count match (x = u, v, w) 00: TIM4_ < t > _OyL port position remains unchanged when the condition is satisfied 01: TIM4_ < t > _OyL port bit output high level when conditions are met 10: TIM4_ < t > _OyL port bit output low level when condition is satisfied 11: TIM4_ < t > _OyL port bit output inversion when conditions are met (y=U, V, W)	R/W
b17~b16	EOPNDCL [1:0]	Expansion Down Count Port Status L	Condition: Counter Down Count & OCCRxI Count Mismatch & OCCRxh Count Match (x = u, v, w) 00: TIM4_ < t > _OyL port position remains unchanged when the condition is satisfied 01: TIM4_ < t > _OyL port bit output high level when conditions are met 10: TIM4_ < t > _OyL port bit output low level when condition is satisfied 11: TIM4_ < t > _OyL port bit output inversion when conditions are met (y=U, V, W)	R/W
b15~b14	OPNZRL [1:0]	Underflow point port state L	Condition: Count underflow & OCCRxI count mismatch & OCCRxh count mismatch (x = u, v, w) 00: TIM4_ < t > _OyL port position remains unchanged when the condition is satisfied 01: TIM4_ < t > _OyL port bit output high level when conditions are met 10: TIM4_ < t > _OyL port bit output low level when condition is satisfied 11: TIM4_ < t > _OyL port bit output inversion when conditions are met (y=U, V, W)	R/W
b13~b12	OPNPKL [1:0]	Overflow point port status L	Condition: Count Overflow & OCCRxI Count Mismatch & OCCRxh Count Mismatch (x = u, v, w) 00: TIM4_ < t > _OyL port position remains unchanged when the condition is satisfied 01: TIM4_ < t > _OyL port bit output high level when conditions are met 10: TIM4_ < t > _OyL port bit output low level when	R/W

			condition is satisfied 11: TIM4_< t >_OyL port bit output inversion when conditions are met (y=U, V, W)	
b11~b10	OPZRL [1:0]	Underflow point port state L	Condition: Count underflow & OCCRxI count match & OCCRxh count mismatch (x = u, v, w) 00: TIM4_< t >_OyL port position remains unchanged when the condition is satisfied 01: TIM4_< t >_OyL port bit output high level when conditions are met 10: TIM4_< t >_OyL port bit output low level when condition is satisfied 11: TIM4_< t >_OyL port bit output inversion when conditions are met (y=U, V, W)	R/W
b9~b8	OPUCL [1:0]	Up count port status L	Condition: Counter up count & OCCRxI count match & OCCRxh count mismatch (x = u, v, w) 00: TIM4_< t >_OyL port position remains unchanged when the condition is satisfied 01: TIM4_< t >_OyL port bit output high level when conditions are met 10: TIM4_< t >_OyL port bit output low level when condition is satisfied 11: TIM4_< t >_OyL port bit output inversion when conditions are met (y=U, V, W)	R/W
b7~b6	OPPKL [1:0]	Overflow point port status L	Condition: Count Overflow & OCCRxI Count Match & OCCRxh Count Mismatch (x = u, v, w) 00: TIM4_< t >_OyL port position remains unchanged when the condition is satisfied 01: TIM4_< t >_OyL port bit output high level when conditions are met 10: TIM4_< t >_OyL port bit output low level when condition is satisfied 11: TIM4_< t >_OyL port bit output inversion when conditions are met (y=U, V, W)	R/W
b5~b4	OPDCL [1:0]	Down count port status L	Condition: Counter Down Count & OCCRxI Count Match & OCCRxh Count Mismatch (x = u, v, w) 00: TIM4_< t >_OyL port position remains unchanged when the condition is satisfied 01: TIM4_< t >_OyL port bit output high level when conditions are met 10: TIM4_< t >_OyL port bit output low level when condition is satisfied 11: TIM4_< t >_OyL port bit output inversion when conditions are met (y=U, V, W)	R/W
b3	OCFZRL	Underflow point OCFL state L	Condition: Count underflow & OCCRxI count match (x = u, v, w) 0: OCSR.OCFL bit remains unchanged when the condition is satisfied 1: OCSR.OCFL position bits when conditions are met	R/W
b2	OCFUCL	Count up OCFL status L	Condition: Counter Up Count & OCCRxI Count Match (x = u, v, w) 0: OCSR.OCFL bit remains unchanged when the condition is satisfied 1: OCSR.OCFL position bits when conditions are met	R/W
b1	OCFPKL	Overflow point OCFL state L	Condition: Count Overflow & OCCRxI Count Match (x = u, v, w) 0: OCSR.OCFL bit remains unchanged when the condition is satisfied 1: OCSR.OCFL position bits when conditions are met	R/W
b0	OCFDCL	Count down OCFL status L	Condition: Counter Down Count & OCCRxI Count Match (x = u, v, w) 0: OCSR.OCFL bit remains unchanged when the condition is satisfied 1: OCSR.OCFL position bits when conditions are met	R/W

Note:

- When reading data from the local area, it is not the value of the buffer register, but the value of the OCMR register.

- *TIM4_<t>_OyL can be determined by the count value of OCCRxI and counter (stand-alone mode of operation), or the count value of OCCRxh and counter, and the count value of OCCRxI and counter (linked mode of operation). Write the same 12-bit value to the bit [31: 20] and bit [15: 4] of register OCMRxI and write OCMRxI [19: 16] to "0000". At this time, the output of TIM4 TIM4_<t>_OyL is not affected by OCCRxh, but only determined by OCCRxI. This mode is called standalone operating mode - channel yH is determined by OCCRxh and channel yL is determined by OCCRxI. If the above conditions are not met in the stand-alone operation mode, the link operation mode-channel yL output is affected by both OCCRxh and OCCRxI.*

(X=u, v, w, y=U, V, W)

- *When the following two conditions are met, it is also considered to meet the conditions set by OCMRxh.OPPKH [7:6] (x=u , v , w):*
 - 1) *CCSR.MODE=1 && count overflow && OCCRxh=0xFFFF;*
 - 2) *OCERx.MCECH=1 && count overflow && OCCRxh≥CNTR*
- *When the following two conditions are met, it is also considered to meet the conditions set by OCMRxI.OPPKL [7:6] (x=u , v , w):*
 - 1) *CCSR.MODE=1 && count overflow && OCCRxI=0xFFFF;*
 - 2) *OCERx.MCECL=1 && count overflow && OCCRxI≥CNTR*

21.5.9 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	Flag	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 SCCRegister											R/W	

21.5.10 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	-	-	EVT DS	EVT MS	-	-	LMC	EVTOS [2:0]		BUFEN [1:0]		
Bit	Flag		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								R/W		
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								R/W		
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								R/W		
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								R/W		
b11~b10		Reserved		-		Read as "0", write as "0"								R/W	
b9	EVTDS		EVT Delay Object Selection		0: OCCRxh matches as a delay comparison object in delay start mode 1: OCCRxI matches as a delay comparison object in delay start mode (x=u, v, w) Note: The bit is invalid when EVTMS = 0								R/W		
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)								R/W		
b7~b6		Reserved		-		Read as "0", write as "0"								R/W	
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)								R/W		
b4~b2		EVTOS [2:0]		EVT Output Selection		000: EVT Output Valid for Special Event 0 001: EVT Output Valid for Special Event 1 010: EVT Output Valid for Special Event 2 011: EVT Output Valid for Special Event 3 100: EVT Output Valid for Special Event 4 101: EVT Output Valid for Special Event 5 Please do not set other values								R/W	
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								R/W	

21.5.11 Dedicated Extended Control Register (TMR4_SCER)

Reset value: 0xFF00

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved												PCTS	EVTRS [2:0]		
Bit	Flag	Bit Name	Function											Read and write	
b15~b4	Reserved	-	Read as "0", write as "0"											R/W	
b3	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 TIM4_<t> PCT port 1 : Enable the counting direction signal to be output to the TIM4_<t> PCT port											-	
b2~b0	EVTRS [2:0]	Dedicated event output selection	000: Disable dedicated event output to the port 001: EVT of Special Event 0 is output to TIM4_<t>_ADSM port 010: EVT of Special Event1 is output to TIM4_<t>_ADSM port 011: EVT of Special Event 2 is output to TIM4_<t>_ADSM port 100: EVT of Special Event 3 is output to TIM4_<t>_ADSM port 101: EVT of Special Event 4 is output to TIM4_<t>_ADSM port 110: EVT of Special Event 5 is output to TIM4_<t>_ADSM port 111: Disable setting											R/W	

21.5.12 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
Reserved								MPCE	MZCE	-	-	AMC [3:0]			
Bit	Flag	Bit Name	Function											Read and write	
b15~b8	Reserved	-	Read as "1", write as "1"											R/W	
b7	MPCE	Periodic interval response enable	0: No comparison between AMC and CVPR.PIC 1: Enable comparison between AMC and CVPR.PIC											R/W	
b6	MZCE	Periodic interval response enable	0: No comparison between AMC and CVPR.ZIC 1: Enable comparison between AMC and CVPR.ZIC											R/W	
b5~b4	Reserved	-	Read as "0", write as "0"											R/W	
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.											R/W	

Note:

- When reading data from the local area, it is not the value of the buffer register, but the value of the SCMRregister.

21.5.13 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
Reserved								LVLS [1:0]		PWMMD [1:0]		-	DIVCK [2:0]		
Bit	Flag	Bit Name	Function										Read and write		
b15~b8	Reserved	-	Read as "1", write as "1"										R/W		
b7~b6	LVLS [1:0]	PWM output polarity control	00: Neither TIM4 _ < t > _ OxH nor TIM4 _ < t > _ OxL are inverted 01: Output of TIM4 _ < t > _ OxH and TIM4 _ < t > _ OxL are inverted 10: The output of TIM4 _ < t > _ OxH is reversed, and the output of TIM4 _ < t > _ OxL is not reversed. 11: The output of TIM4 _ < t > _ OxH is not reversed, the output of TIM4 _ < t > _ OxL is reversed.										R/W		
b5~b4	PWMMD [1:0]	PWM output mode	00: Pass-through mode 01: Dead-zone timer mode 10: Dead-zone timer filtering mode 11: Disable setting										R/W		
b3	Reserved	-	Read as "0", write as "0"										R/W		
b2~b0	DIVCK [2:0]	Count 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										R/W		

21.5.14 PWM Status Control Register (TMR4_PSCR)

Reset value: 0x05550000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved				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
Reserved						AOE	MOE	ODT [1:0]		OEWL	OEWH	OEVL	OEVL	OEVL	OEVL

Bit	Flag	Bit Name	Function	Read and write
b31~b28	Reserved	-	Read as "0", write as "0"	R/W
b27~b26	OSWL [1:0]	TIM4_<t>_OWL 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 TIM4_<t>_OWL port outputs normally 01: When one of the above conditions is met, the TIM4_<t>_OWL port output is Hi-z 10 : When one of the above conditions is met, the TIM4_<t>_OWL port output is fixed at a low level. 11 : When one of the above conditions is met, the TIM4_<t>_OWL port output is fixed at a high level Note: This bit can only be modified when Timer4 is stopped (CCSR.STOP=1).	R/W
b25~b24	OSWH [1:0]	TIM4_<t>_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 TIM4_<t>_OWH port outputs normally 01: When one of the above conditions is met, the TIM4_<t>_OWH port output is Hi-z 10 : When one of the above conditions is met, the TIM4_<t>_OWH port output is fixed at a low level. 11 : When one of the above conditions is met, the TIM4_<t>_OWH port output is fixed at a high level Note: This bit can only be modified when Timer4 is stopped (CCSR.STOP=1).	R/W
b23~b22	OSVL [1:0]	TIM4_<t>_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 TIM4_<t>_OVL port outputs normally 01: When one of the above conditions is met, the TIM4_<t>_OVL port output is Hi-z 10 : When one of the above conditions is met, the TIM4_<t>_OVL port output is fixed at a low level. 11 : When one of the above conditions is met, the TIM4_<t>_OVL port output is fixed at a high level Note: This bit can only be modified when Timer4 is stopped (CCSR.STOP=1).	R/W
b21~b20	OSVH [1:0]	TIM4_<t>_OVH 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 TIM4_<t>_OVH port outputs normally 01: When one of the above conditions is met, the TIM4_<t>_OVH port output is Hi-z 10 : When one of the above conditions is met, the TIM4_<t>_OVH port output is fixed at a low level. 11 : When one of the above conditions is met, the TIM4_<t>_OVH port output is fixed at a high level Note: This bit can only be modified when Timer4 is stopped (CCSR.STOP=1).	R/W
b19~b18	OSUL [1:0]	TIM4_<t>_OUL 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 TIM4_<t>_OUL port outputs normally 01: When one of the above conditions is met, the TIM4_<t>_OUL port output is Hi-z 10 : When one of the above conditions is met, the TIM4_<t>_OUL port output is fixed at a low level. 11 : When one of the above conditions is met, the TIM4_<t>_OUL port output is fixed at a high level	R/W

			Note: This bit can only be modified when Timer4 is stopped (CCSR.STOP=1).	
b17~b16	OSUH [1:0]	TIM4_<t>_OUH output disable status	Conditions: 1. EMB event occurs; 2. Software writes MOE=0; 3. MOE=1&OEUH=0 00: When one of the above conditions is met, the TIM4_<t>_OUH port outputs normally 01: When one of the above conditions is met, the TIM4_<t>_OUH port output is Hi-z 10: When one of the above conditions is met, the TIM4_<t>_OUH port output is fixed at a low level. 11: When one of the above conditions is met, the TIM4_<t>_OUH port output is fixed at a high level Note: This bit can only be modified when Timer4 is stopped (CCSR.STOP=1).	R/W
b15~b10	Reserved	-	Read as "0", write as "0"	R/W
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 Setting EMB_STATCLR or clearing EMB_SOE can clear the EMB event.	R/W
b8	MOE	Host Output Enable	0: TIM4_<t>_Oxy port outputs the state specified by the OSxy bit (x=U, V, W, y=H, L) 1: TIM4_<t>_Oxy port output is enabled when OExy bit is set to enable (x=U, V, W, y=H, L) 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.	R/W
b7~b6	ODT	Port enable bit effective time	After the value of the port enable bit OExy is changed (x=U , V , W , y=H , L) 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.	R/W
b5	OEWL	TIM4_<t>_OWL output enable	0: TIM4_<t>_OWL port output status is set by OSWL bit 1: When MOE=1, TIM4_<t>_OWL port outputs normal PWM waveform	R/W
b4	OEWH	TIM4_<t>_OWH output enable	0: TIM4_<t>_OWH port output status is set by OSWH bit 1: When MOE=1, TIM4_<t>_OWH port outputs normal PWM waveform	R/W
b3	OEVL	TIM4_<t>_OVL output enable	0: TIM4_<t>_OVL port output status is set by OSVL bit 1: When MOE=1, TIM4_<t>_OVL port outputs normal PWM waveform	R/W
b2	OEVL	TIM4_<t>_OVH output enable	0: TIM4_<t>_OVH port output status is set by OSVH bit 1: When MOE=1, TIM4_<t>_OVH port outputs normal PWM waveform	R/W
b1	OEUL	TIM4_<t>_OUL output enable	0: TIM4_<t>_OUL port output status is set by OSUL bit 1: When MOE=1, TIM4_<t>_OUL port outputs normal PWM waveform	R/W
b0	OEUH	TIM4_<t>_OUH output enable	0: TIM4_<t>_OUH port output status is set by OSUH bit 1: When MOE=1, TIM4_<t>_OUH port outputs normal PWM waveform	R/W

The following table shows the relationship between the PWM port output status and the register setting value.

Table 21-4 PWM Port Output Status and Register Setting Value

MOE bit	OSxy Bit	OSxy bit	TIM4_<t>_Oxy output status
1	0	00	Normal Output (Timer4 driven)
		01	Hi-Z
		10	Low level
		11	High level
	1	X	Normal Output (Timer4 driven)
0	X	00	Normal Output (Timer4 driven)
		01	Hi-Z
		10	Low level
		11	High level

21.5.15 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	Flag	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.											R/W	

21.5.16 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	Flag	Bit Name	Function											Read and write	
b15~b0	PDA/BR [15:0]	Dead zone initial value	Dead zone count initial value											R/W	

21.5.17 Overload Control Status Register (TMR4 _ RCSR)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RTS W	RTE W	RTIC W	RTIF W	RTS V	RTE V	RTIC V	RTIF V	RTS U	RTE U	RTIC U	RTIF U	-	RTID W	RTID V	RTID U
Bit	Flag	Bit Name	Function											Read and write	
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											R/W	
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.											R/W	
b13	RTICW	Zero count matching state W	0: No operation 1: Clear RTIFW flag bit Note: The bit is always 0 when read											R/W	
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											R/W	
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.											R/W	
b9	RTICV	Zero count matching state V	0: No operation 1: Clear RTIFV flag bit Note: The bit is always 0 when read											R/W	
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											R/W	
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.											R/W	
b5	RTICU	Zero count matching state U	0: No operation 1: Clear RTIFU flag bit Note: The bit is always 0 when read											R/W	
b4	RTIFU	Count matching state U	0: Overloaded counter count does not match PFSRu 1: Overloaded counter counts match PFSRu											R	
b3	Reserved	-	Read as "0", write as "0"											R/W	
b2	RTIDW	Heavy-duty interrupt masking 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											R/W	
b1	RTIDV	Heavy-duty interrupt masking 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											R/W	
b0	RTIDU	Heavy-duty interrupt masking 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											R/W	

21.5.18 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
Reserved										ITEN5	ITEN4	ITEN3	ITEN2	ITEN1	ITEN0
Bit	Flag	Bit Name	Function		Read and write										
b15~b6	Reserved	-	Read as "0", write as "0"		R/W										
b5	ITEN5	Dedicated match wl interrupt enable	0: Dedicated count compare match 5 interrupt is disabled 1: Dedicated count compare match 5 interrupt enable		R/W										
b4	ITEN4	Dedicated match wh interrupt enable	0: Dedicated count compare match 4 interrupt is disabled 1: Dedicated count compare match 4 interrupt enable		R/W										
b3	ITEN3	Dedicated match vl interrupt enable	0: Dedicated count compare match 3 interrupt is disabled 1: Dedicated count compare match 3 interrupt enable		R/W										
b2	ITEN2	Dedicated match vh interrupt enable	0: Dedicated count compare match 2 interrupt is disabled 1: Dedicated count compare match 2 interrupt enable		R/W										
b1	ITEN1	Dedicated match ul interrupt enable	0: Dedicated count compare match 1 interrupt is disabled 1: Dedicated count compare match 1 interrupt enable		R/W										
b0	ITEN0	Dedicated match uh interrupt enable	0: Dedicated count compare match 0 interrupt is disabled 1: Dedicated count compare match 0 interrupt enable		R/W										

21.5.19 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
Reserved										SF5	SF4	SF3	SF2	SF1	SF0
Bit	Flag	Bit Name	Function		Read and write										
b15~b6	Reserved	-	Read as "1", write as "1"		R/W										
b5	SF5	Dedicated count match flag wl	0: No dedicated count compare match 5 event occurred 1: A dedicated count compare match 5 event occurs		R/W										
b4	SF4	Dedicated count match flag wh	0: No dedicated count compare match 4 event occurred 1: A dedicated count compare match 4 event occurs		R/W										
b3	SF3	Dedicated count match flag vl	0: No dedicated count compare match 3 event occurred 1: A dedicated count compare match 3 event occurs		R/W										
b2	SF2	Dedicated count match flag bit vh	0: No dedicated count compare match 2 event occurred 1: A dedicated count compare match 2 event occurs		R/W										
b1	SF1	Dedicated count match flag ul	0: No dedicated count compare match 1 event occurred 1: A dedicated count compare match 1 event occurs		R/W										
b0	SF0	Dedicated count match flag uh	0: No dedicated count compare match 0 event occurred 1: A dedicated count compare match 0 event occurs		R/W										

22 Emergency Brake Module (EMB)

22.1 Introduction

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 outputting PWM signals to the external motor. The following factors are 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

The EMB structure block diagram is shown in the Figure 22-1.

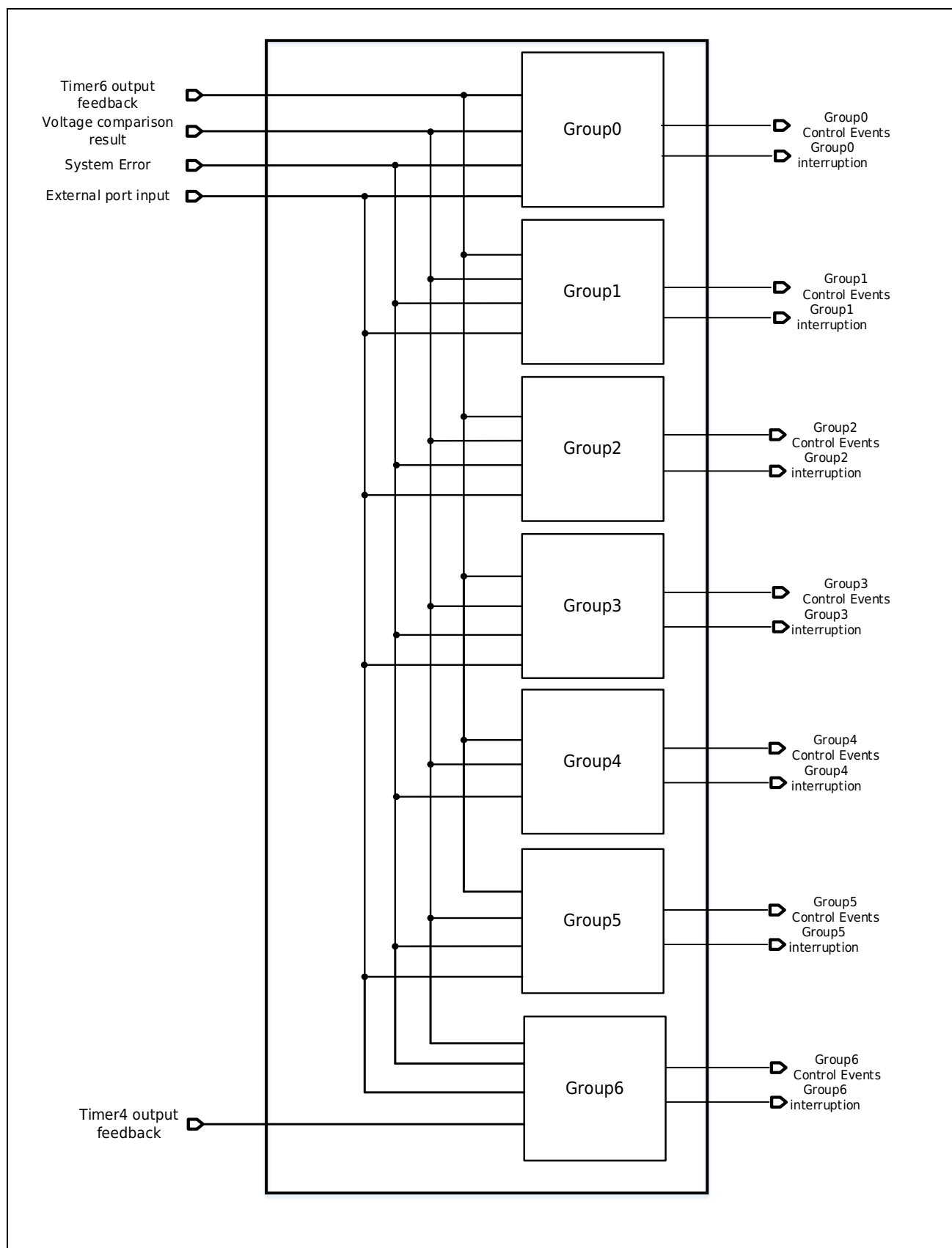


Figure 22-1 EMB Block Diagram

22.2 Functional Description

22.2.1 Overview

EMB is used to output a control event signal to the timer module with PWM function (Timer4, Timer6) when certain conditions are met, notifying the timer module to turn off the current PWM output. EMB module has 7 groups , among which group0 ~group5 are used to control Timer6 and are selected by the Timer6 register setting. Group6 is used to control Timer4.

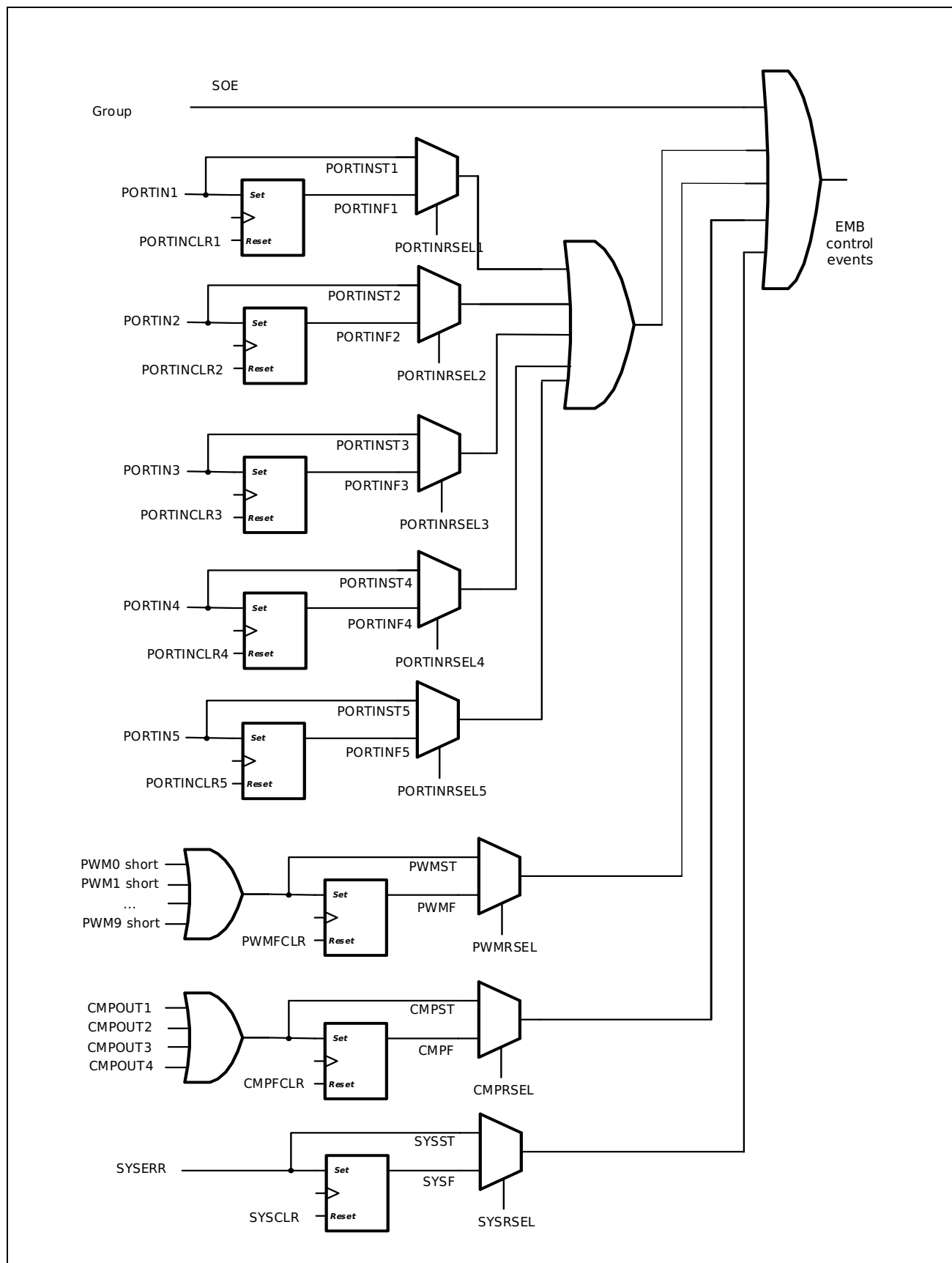


Figure 22-2 EMB Channel Functional Block Diagram

22.2.2 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 is shown in the following table.

Table 22-1 Port Allocation Table

port	PIN Name
EMB_PORT1	PA6, PA15, PA12, PE15, PB12, PA14, PB8
EMB_PORT2	PA15, PA11, PA6, PC3
EMB_PORT3	PA0, PA9, PE7, PC5
EMB_PORT4	PB11, PB5, PE8, PA1, PF2
EMB_PORT5	PC7, PA10, PD2, 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=0~6, y=1~5) to be valid, and at the same time, set EMB_CTL1_x.INVSEL y (x=0~6, 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 through EMB_CTL2_x.NFENy (x=0~6,y=1~5) and EMB_CTL2_x.NFSELy (x=0~6,y=1~5) as needed.

After the filter samples the input signal according to the filter clock, it uses three comparison consistency methods to filter. That is, when the filter clock samples the level on the port three times in a consistent manner, the level is regarded as a valid level and transmitted to the module; levels less than three times consistent will be filtered out as external interference and not transmitted to the module.

When a valid level that meets the conditions is generated on the port, the port input status EMB_STATx.PORTINSTy (x=0~6, y=1~5) of the EMB status register is set, and the port input control flag EMB_STATx.PORTINFy (x=0~6, y=1~5) is set. An interrupt will be generated when the interrupt enable register EMB_INTENx.PORTINEN (x=0~6)=1.

EMB are output to Timer6 and Timer4 through EMB release mode select register EMB_RLSSELx.PORTINRSELy (x=0~6, y=1~5) and selected by EMB_STATx.PORTINSTy (x=0~6, y=1~5) or EMB_STATx.PORTINFy (x= 0 ~6, y=1 ~ 5). After receiving a control event, Timer6 and Timer4 can set the output port to high level, low level or high impedance state according to the register settings.

When releasing control, if EMB_RLSSELx. PORTINRSELy (x=0~6, y=1~5)=1, when the port input level becomes invalid, EMB_STATx.PORTINSTy (x=0~6, y=1~5) will be automatically cleared and the control event will be released immediately. If EMB_RLSSELx.PORTINRSELy (x=0~6, y=1~5)=0, when the port input level becomes invalid, it is necessary to write the EMB status reset register

EMB_STATCLR_x.PORTINFCLR (x=0~6) to clear EMB_STAT_x.PORTINF (x=0~6) to release the control event.

22.2.3 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=0~9 or x=4~5, y=0~2) to be valid, and at the same time select the valid level to be monitored through the EMB control register EMB_CTL2_x.PWMLV_y (x=0~5, y=0~9 or x=4~5, y=0~2).

After the setup is completed, EMB monitors the complementary PWM output signals of Timer6 and Timer4. When the output signals are at the same high or low level, the PWM output status EMB_STAT_x.PWMST (x=0~6) of the EMB status register is set, and the PWM output same-phase control flag EMB_STAT_x.PWMSF (x=0~6) is set. When the interrupt enable register EMB_INTEN_x.PWMEN (x=0~6)=1, an interrupt will be generated.

The control events of EMB are output to Timer6 and Timer4 through EMB_STAT_x. PWMST (x=0~6) or EMB_STAT_x. PWMSF (x=0~6) through the EMB release mode selection register EMB_RLSSEL_x.PWMRSEL (x=0~6). 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_RLSSEL_x. PWMRSEL (x = 0 ~ 6) = 1, when the PWM output is released from the high or low state, EMB_STAT_x.PWMST (x = 0 ~ 6) will be automatically cleared and the control event will be released immediately. If EMB_RLSSEL_x. PWMST (x=0~6) = 0, when the PWM output is released from the high or low state, it is necessary to write the EMB status reset register EMB_STATCLR_x. PWMSFCLR (x=0~6) to clear EMB_STAT_xPWMSF (x=0~6) to 0 and then release the control event.

Group 0-5 can be used to monitor the complementary PWM output signal of Timer6 , and group 6 can be used to monitor the complementary PWM output signal of Timer4_1.

Table 22-2 EMB Group Comparison Table

Port Name	Function	Correspond Group	EMB_CTL1 Control Bits	EMB_CTL2 Control Bits
TIM6_m_PWMA(m=1~10)	Complementary PWM Output Signal of Timer6	group0 group1 group2 group3 group4 group5	PWMSEN [9:0]	PWMLV [9:0]
TIM6_m_PWMB(m=1~10)				
TIM4_1_OUH	Complementary PWM Output Signal of Timer4_1	group6	PWMSEN [2]	PWMLV [2]
TIM4_1_OUL			PWMSEN [1]	PWMLV [1]
TIM4_1_OVH				
TIM4_1_OVL				

Port Name	Function	Correspond Group	EMB_CTL1 Control Bits	EMB_CTL2 Control Bits
TIM4_1_OWH			PWMSSEN [0]	PWMLV [0]
TIM4_1_OWL				

22.2.4 Stop PWM Signal Output According to Voltage Comparator Comparison Result

Each group of EMB can send control event signals to Timer6 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. CMPENy (x=0~6) (y=1~2) to be valid. When the voltage comparator comparison result flag is set, the voltage comparator status EMB_STATx.CMPST (x=0~6) of the EMB status register is set, and the EMB voltage comparator control flag EMB_STATx.CMPF (x=0~6) is set. When the interrupt enable register EMB_INTENx. CMPINTEN (x=0~6)=1, an interrupt will be generated.

The control events of EMB are output to Timer6 and Timer4 through EMB_STATx.CMPST (x=0~6) or EMB_STATx.CMPF (x=0~6) via the EMB release mode selection register EMB_RLSSELx.CMPRELY (x=0~6). 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 ~ 6) = 1, when the voltage comparator result becomes invalid, EMB_STATx.CMPST (x = 0 ~ 6) will be automatically cleared and the control event will be released immediately. If EMB_RLSSELx.CMPRELY (x=0~6)=0, when the voltage comparator result becomes invalid, it is necessary to write the EMB status reset register EMB_STATCLRxCMPFCLR (x=0~6) to clear EMB_STATx.CMPF (x=0~6) to 0 and then release the control event.

22.2.5 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, SRAM parity error and 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=0~6) to be valid, and set the EMB_CTL1_x. PVDEN, LOCKUPEN, SRAMPYERREN, SRAMERREN, 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=0~6) of the EMB status register is set, and the EMB system error control flag EMB_STATx.SYSF (x=0~6) is set. An interrupt will also be generated when the interrupt enable register EMB_INTENx. SYSINTEN (x = 0 ~ 6) = 1.

The control events of EMB are output to Timer6 and Timer4 through EMB_STATx.SYSST (x=0~6) or EMB_STATx.SYSF (x=0~6) via the EMB release mode selection register EMB_RLSSELx.SYSRSELY (x=0~6). 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=0~6)=1, EMB_STATx.SYSST(x=0~6) will be automatically cleared when the system error flag bit becomes invalid, and the control event will be released immediately. If EMB_RLSSELx.SYSRSELY (x=0~6)=0, when the system error flag bit becomes invalid, it is necessary to write the EMB status reset register EMB_STATCLR x.SYSFCLR (x=0~6) to clear EMB_STATx.SYSF (x=0~6) to 0 and then release the control event.

22.2.6 Write Register Software Control PWM Signal Output

The software output enable control register (EMB_SOEx) (x=0~6) of EMB allows the user to send control signals to Timer6 and Timer4 by directly setting and resetting the software. No interrupt request will be generated when the software controls the PWM output.

22.3 Register Description

Table 22-3 Register List

Name	Abbreviation	Description	Offset Address
EMB Control Register 1	EMB_CTL1	Enable of each PWM output control event	0x0
EMB Control Register 2	EMB_CTL2	Filtering of port input control events and selection of effective level of PWM feedback signal	0x4
EMB software output enable control register	EMB_SOE	Software generates PWM output control events	0x8
EMB status register	EMB_STAT	Indicates the status of PWM output control	0xC
EMB state reset register	EMB_STATCLR	Clear the status of PWM output control	0x10
EMB interrupt enable register	EMB_INTEN	Interrupt enable	0x14
EMB control PWM output release mode selection register	EMB_RSEL	Select the release method for each PWM output control event	0x18

22.3.1 EMB Control Register 1_0~5 (EMB_CTL1_0~5)

The register is a single write register that can only be written once after a reset

Address: 0x40017C00, 0x40017C20, 0x40017C40, 0x40017C60, 0x40017C680, 0x40017C6A0

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
PVD EN	LOCKUPEN	SRAMPYERR EN	SRAMERREN	OSCSTPEN	INVS EL5	INVSE L4	INVSE L3	INVS EL2	INVS EL1	Reserved	PORT INEN 5	PORT INEN 4	PORT INEN 3	PORT INEN 2	PORT INEN 1
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved	PWM SEN 9	PWM SEN 8	PWM SEN 7	PWM SEN 6	PWM SEN5	PWM SEN4	PWM SEN3	PWM SEN2	PWM SEN1	PWM SEN0	SYSEN	Reserved	Reserved	CMP EN2	CMP EN1

Bit	Flag	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	R/W
b30	LOCKUPEN	Cortex-M4 LOCKUP control enable	0: Output control is invalid when LOCKUP occurs 1: Output control is valid when LOCKUP occurs	R/W
b29	SRAMPYERREN	SRAM parity check error control enable	0: Output control is invalid when SRAM parity error occurs 1: Output control is valid when SRAM parity error occurs	R/W
b28	SRAMERREN	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	R/W
b27	OSCSTPEN	Oscillator stop control enable	0: Output control invalid when oscillator stops 1: Output control valid when oscillator stops	R/W
b26	INVSEL5	Port 5 input active level selection	0: High level valid 1: Low level valid	R/W
b25	INVSEL4	Port 4 input active level selection	0: High level valid 1: Low level valid	R/W
b24	INVSEL3	Port 3 input active level selection	0: High level valid 1: Low level valid	R/W
b23	INVSEL2	Port 2 input active level selection	0: High level valid 1: Low level valid	R/W
b22	INVSEL1	Port 1 input active level selection	0: High level valid 1: Low level valid	R/W
b21	Reserved	-	Read "0" when reading, write "0" when writing	R/W
b20	PORTINEN5	Port 5 input control enable	0: Invalid port input control 1: Port input control is valid	R/W
b19	PORTINEN4	Port 4 input control enable	0: Invalid port input control 1: Port input control is valid	R/W
b18	PORTINEN3	Port 3 input control enable	0: Invalid port input control 1: Port input control is valid	R/W
b17	PORTINEN2	Port 2 input control enable	0: Invalid port input control 1: Port input control is valid	R/W
b16	PORTINEN1	Port 1 input control enable	0: Invalid port input control 1: Port input control is valid	R/W
b15	Reserved	-	Read "0" when reading, write "0" when writing	R/W
b14~b5	PWMSEN [9:0]	TIM6_m_PWMA/B(m=1~10) Short-circuit output control enable	Corresponding to 10 groups of PWM feedback input 0: Invalid output control when short circuit 1: Output control effective when short circuit	R/W
b4	SYSEN	System error control enable	0: Output control invalid when a system error occurs 1: Output control valid when a system error occurs	R/W
b3~b2	Reserved	-	Read "0" when reading, write "0" when writing	R/W
b1	CMPEN2	CMP2 Voltage comparator comparison result control enable	0: Invalid voltage comparator output control 1: Voltage comparator output control is effective	R/W
b0	CMPEN1	CMP1 Voltage comparator comparison result control enable	0: Invalid voltage comparator output control 1: Voltage comparator output control is effective	R/W

22.3.2 EMB Control Register 1_6 (EMB_CTL1_6)

The register is a single write register that can only be written once after a reset

Address: 0x40017CC0

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
PVDEN	LOCKUPEN	SRAMPYERREN	SRAMERREN	OSCSTPEN	INVSEL5	INVSEL4	INVSEL3	INVSEL2	INVSEL1	Reserved	PORTINEN5	PORTINEN4	PORTINEN3	PORTINEN2	PORTINEN1
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								PWMSEN2	PWMSEN1	PWMSEN0	SYSEN	Reserved	Reserved	CMPEN2	CMPEN1

Bit	Flag	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	R/W
b30	LOCKUPEN	Cortex-M4 LOCKUP control enable	0: Output control is invalid when LOCKUP occurs 1: Output control is valid when LOCKUP occurs	R/W
b29	SRAMPYERREN	SRAM parity check error control enable	0: Output control is invalid when SRAM parity error occurs 1: Output control is valid when SRAM parity error occurs	R/W
b28	SRAMERREN	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	R/W
b27	OSCSTPEN	Oscillator stop control enable	0: Output control invalid when oscillator stops 1: Output control valid when oscillator stops	R/W
b26	INVSEL5	Port 5 input active level selection	0: High level valid 1: Low level valid	R/W
b25	INVSEL4	Port 4 input active level selection	0: High level valid 1: Low level valid	R/W
b24	INVSEL3	Port 3 input active level selection	0: High level valid 1: Low level valid	R/W
b23	INVSEL2	Port 2 input active level selection	0: High level valid 1: Low level valid	R/W
b22	INVSEL1	Port 1 input active level selection	0: High level valid 1: Low level valid	R/W
b21	Reserved	-	Read "0" when reading, write "0" when writing	R/W
b20	PORTINEN5	Port 5 input control enable	0: Invalid port input control 1: Port input control is valid	R/W
b19	PORTINEN4	Port 4 input control enable	0: Invalid port input control 1: Port input control is valid	R/W
b18	PORTINEN3	Port 3 input control enable	0: Invalid port input control 1: Port input control is valid	R/W
b17	PORTINEN2	Port 2 input control enable	0: Invalid port input control 1: Port input control is valid	R/W
b16	PORTINEN1	Port 1 input control enable	0: Invalid port input control 1: Port input control is valid	R/W
b15~b8	Reserved	-	Read "0" when reading, write "0" when writing	R/W
b7	PWMSEN2	TIM4_m_OUH/L short-circuit output control enable (m=1)	0: Invalid output control when short circuit 1: Output control effective when short circuit	R/W
b6	PWMSEN1	TIM4_m_OVH/L short-circuit output control enable (m=1)	0: Invalid output control when short circuit 1: Output control effective when short circuit	R/W
b5	PWMSEN0	TIM4_m_OWH/L short-circuit output control enable (m=1)	0: Invalid output control when short circuit 1: Output control effective when short circuit	R/W
b4	SYSEN	System error control enable	0: Output control invalid when a system error occurs 1: Output control valid when a system error occurs	R/W
b3~b2	Reserved	-	Read "0" when reading, write "0" when writing	R/W
b1	CMPEN2	CMP2 Voltage comparator comparison result control enable	0: Invalid voltage comparator output control 1: Voltage comparator output control is effective	R/W
b0	CMPEN1	CMP1 Voltage comparator comparison result control enable	0: Invalid voltage comparator output control 1: Voltage comparator output control is effective	R/W

22.3.3 EMB Control Register 2_0~5 (EMB_CTL2_0~5)

The register is a single write register that can only be written once after a reset

Address: 0x40017C04, 0x40017C24, 0x40017C44, 0x40017C64, 0x40017C84, 0x40017CA4

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved	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
Reserved						PWMLV9	PWMLV8	PWMLV7	PWMLV6	PWMLV5	PWMLV4	PWMLV3	PWMLV2	PWMLV1	PWMLV0

Bit	Flag	Bit Name	Function	Read and write
b31	Reserved	-	Read "0" when reading, write "0" when writing	R/W
b30	NFEN5	Port 5 input digital filter enable	0: Invalid filter 1: Filter is valid	R/W
b29~b28	NFSEL5 [1:0]	Port 5 digital filter clock selection	00: Using bus clock filtering 01: 8 Frequency Division Filter Using Bus Clock 10: 32 Frequency Division Filter with Bus Clock 11: 128 Frequency Division Filter with Bus Clock	R/W
b27	NFEN4	Port 4 input digital filter enable	0: Invalid filter 1: Filter is valid	R/W
b26~b25	NFSEL4 [1:0]	Port 4 digital filter clock selection	00: Using bus clock filtering 01: 8 Frequency Division Filter Using Bus Clock 10: 32 Frequency Division Filter with Bus Clock 11: 128 Frequency Division Filter with Bus Clock	R/W
b24	NFEN3	Port 3 input digital filter enable	0: Invalid filter 1: Filter is valid	R/W
b23~b22	NFSEL3 [1:0]	Port 3 digital filter clock selection	00: Using bus clock filtering 01: 8 Frequency Division Filter Using Bus Clock 10: 32 Frequency Division Filter with Bus Clock 11: 128 Frequency Division Filter with Bus Clock	R/W
b21	NFEN2	Port 2 input digital filter enable	0: Invalid filter 1: Filter is valid	R/W
b20~b19	NFSEL2 [1:0]	Port 2 digital filter clock selection	00: Using bus clock filtering 01: 8 Frequency Division Filter Using Bus Clock 10: 32 Frequency Division Filter with Bus Clock 11: 128 Frequency Division Filter with Bus Clock	R/W
b18	NFEN1	Port 1 input digital filter enable	0: Invalid filter 1: Filter is valid	R/W
b17~b16	NFSEL1 [1:0]	Port 1 digital filter clock selection	00: Using bus clock filtering 01: 8 Frequency Division Filter Using Bus Clock 10: 32 Frequency Division Filter with Bus Clock 11: 128 Frequency Division Filter with Bus Clock	R/W
b15~b10	Reserved	-	Read "0" when reading, write "0" when writing	R/W
b9~b0	PWMLV [9:0]	TIM6_m_PWMA/B(m=1~10) Output effective level selection	Corresponding to 10 groups of PWM feedback input 0: Low to active 1: High to active	R/W

22.3.4 EMB Control Register 2_6 (EMB_CTL2_6)

The register is a single write register that can only be written once after a reset

Address: 0x40017CC4

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved	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
Reserved													PWMLV2	PWMLV1	PWMLV0

Bit	Flag	Bit Name	Function	Read and write
b31	Reserved	-	Read "0" when reading, write "0" when writing	R/W
b30	NFEN5	Port 5 input digital filter enable	0: Invalid filter 1: Filter is valid	R/W
b29~b28	NFSEL5 [1:0]	Port 5 digital filter clock selection	00: Using bus clock filtering 01: 8 Frequency Division Filter Using Bus Clock 10: 32 Frequency Division Filter with Bus Clock 11: 128 Frequency Division Filter with Bus Clock	R/W
b27	NFEN4	Port 4 input digital filter enable	0: Invalid filter 1: Filter is valid	R/W
b26~b25	NFSEL4 [1:0]	Port 4 digital filter clock selection	00: Using bus clock filtering 01: 8 Frequency Division Filter Using Bus Clock 10: 32 Frequency Division Filter with Bus Clock 11: 128 Frequency Division Filter with Bus Clock	R/W
b24	NFEN3	Port 3 input digital filter enable	0: Invalid filter 1: Filter is valid	R/W
b23~b22	NFSEL3 [1:0]	Port 3 digital filter clock selection	00: Using bus clock filtering 01: 8 Frequency Division Filter Using Bus Clock 10: 32 Frequency Division Filter with Bus Clock 11: 128 Frequency Division Filter with Bus Clock	R/W
b21	NFEN2	Port 2 input digital filter enable	0: Invalid filter 1: Filter is valid	R/W
b20~b19	NFSEL2 [1:0]	Port 2 digital filter clock selection	00: Using bus clock filtering 01: 8 Frequency Division Filter Using Bus Clock 10: 32 Frequency Division Filter with Bus Clock 11: 128 Frequency Division Filter with Bus Clock	R/W
b18	NFEN1	Port 1 input digital filter enable	0: Invalid filter 1: Filter is valid	R/W
b17~b16	NFSEL1 [1:0]	Port 1 digital filter clock selection	00: Using bus clock filtering 01: 8 Frequency Division Filter Using Bus Clock 10: 32 Frequency Division Filter with Bus Clock 11: 128 Frequency Division Filter with Bus Clock	R/W
b15~b3	Reserved	-	Read "0" when reading, write "0" when writing	R/W
b2	PWMLV2	TIM4_m_OUH/L output valid level selection (m=1)	0: Low to active 1: High to active	R/W
b1	PWMLV1	TIM4_m_OVH/L output valid level selection (m=1)	0: Low to active 1: High to active	R/W
b0	PWMLV0	TIM4_m_OWH/L output valid level selection (m=1)	0: Low to active 1: High to active	R/W

22.3.5 EMB Software Output Enable Control Register (EMB_SOEx) (x=0~6)

Address: 0x40017C08, 0x40017C28, 0x40017C48, 0x40017C68, 0x40017C88, 0x40017CA8, 0x40017CC8

Reset value: 0x00000000

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

Bit	Flag	Bit Name	Function	Read and write
b31~1	Reserved	-	Read "0" when reading, write "0" when writing	R/W
b0	SOE	Software control output	0: PWM normal output 1: PWM stop output	R/W

22.3.6 EMB Status Register (EMB_STATx) (x=0~6)

Address: 0x40017C0C, 0x40017C2C, 0x40017C4C, 0x40017C6C, 0x40017C8C, 0x40017CAC, 0x40017CCC

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved													PORTINST5	PORTINST4	PORTINST3
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PORTINST2	PORTINST1	Reserved	PORTINF5	PORTINF4	PORTINF3	PORTINF2	PORTINF1	SYSST	CMPST	PWMS T	Reserved	SYSF	CMPF	PWMS F	Reserved

Bit	Flag	Bit Name	Function	Read and write
b31~19	Reserved	-	Read "0" when reading, write "0" when writing	R/W
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
b13~12	Reserved	-	Read "0" when reading, write "0" when writing	R/W
b12	PORTINF5	Port 5 input control flag bit	EMB_STATx. Set when PORTINST5 is 1 EMB_STATCLR _x . PORTINFCLR5 is reset to 0 when writing 1	R
b11	PORTINF4	Port 4 input control flag bit	EMB_STATx. Set when PORTINST4 is 1 EMB_STATCLR _x . PORTINFCLR4 is reset to 0 when writing 1	R
b10	PORTINF3	Port 3 input control flag bit	EMB_STATx. Set when PORTINST3 is 1 EMB_STATCLR _x . PORTINFCLR3 is reset to 0 when writing 1	R
b9	PORTINF2	Port 2 input control flag bit	EMB_STATx. Set when PORTINST2 is 1 EMB_STATCLR _x . PORTINFCLR2 is reset to 0 when writing 1	R
b8	PORTINF1	Port 1 input control flag bit	EMB_STATx. Set when PORTINST1 is 1 EMB_STATCLR _x . PORTINFCLR1 is reset 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	Reserved	-	Read "0" when reading, write "0" when writing	R/W
b3	SYSF	System error flag bit	EMB_STATx. Set when SYSST is 1 EMB_STATx. SYSFCLR is cleared to 0 when writing 1	R
b2	CMPF	Voltage comparator control flag bit	EMB_STATx. Set when CMPST is 1 EMB_STATCLR _x . CMPFCLR is reset to 0 when writing 1	R
b1	PWMSF	PWM output same phase control flag bit	EMB_STATx. Set if PWMST is 1 EMB_STATCLR _x . PWMSCLR is cleared to 0 when writing 1	R
b0	Reserved	-	Read "0" when reading, write "0" when writing	R/W

22.3.7 EMB Status Reset Register (EMB_STATCLR_x) (x=0~6)

Address: 0x40017C10, 0x40017C30, 0x40017C50, 0x40017C70, 0x40017C90, 0x40017CB0, 0x40017CD0

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved				PORTIN FCLR5	PORTIN FCLR4	PORTIN FCLR3	PORTIN FCLR2	PORTIN FCLR1	Reserved			SYSFC LR	CMPF CLR	PWMS FCLR	Reser ved

Bit	Flag	Bit Name	Function	Read and write
b31~b13	Reserved	-	Read "0" when reading, write "0" when writing	W
b12	PORTINFCLR5	Reset 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	Reset 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	Reset 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	Reset 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	Reset 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	Reserved	-	Read "0" when reading, write "0" when writing	W
b3	SYSFCLR	Reset 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	Reset 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	Reset 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	Reserved	-	Read "0" when reading, write "0" when writing	W

22.3.8 EMB Interrupt Permission Register (EMB_INTENx) (x=0~6)

Address: 0x40017C14, 0x40017C34, 0x40017C54, 0x40017C74, 0x40017C94, 0x40017CB4, 0x40017CD4

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved				PORTIN TEN5	PORTIN TEN4	PORTIN TEN3	PORTIN TEN2	PORTIN TEN1	Reserved			SYSIN TEN	CMPIN TEN	PWMS INTEN	Reser ved

Bit	Flag	Bit Name	Function	Read and write
b31~b13	Reserved	-	Read "0" when reading, write "0" when writing	R/W
b12	PORTINTEN5	Port 5 input control interrupt enable	0: Port 5 input control does not generate interrupt 1: Port 5 input control generates interrupt	R/W
b11	PORTINTEN4	Port 4 input control interrupt enable	0: Port 4 input control does not generate interrupt 1: Port 4 input control generates interrupt	R/W
b10	PORTINTEN3	Port 3 input control interrupt enable	0: Port 3 input control does not generate interrupt 1: Port 3 input control generates interrupt	R/W
b9	PORTINTEN2	Port 2 input control interrupt enable	0: Port 2 input control does not generate interrupt 1: Port 2 input control generates interrupt	R/W
b8	PORTINTEN1	Port 1 input control interrupt enable	0: Port 1 input control does not generate interrupt 1: Port 1 input control generates interrupt	R/W
b7~b4	Reserved	-	Read "0" when reading, write "0" when writing	R/W
b3	SYSINTEN	System error control interrupt enable	0: System error control does not generate interrupt 1: System error control generates interrupt	R/W
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	R/W
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	R/W
b0	Reserved	-	Read "0" when reading, write "0" when writing	R/W

22.3.9 EMB Control PWM Output Release Mode Select Register (EMB_RLSSELx) (x=0~6)

Address: 0x40017C18, 0x40017C38, 0x40017C58, 0x40017C78, 0x40017C98, 0x40017CB8, 0x40017CD8

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved				PORTINRSEL5	PORTINRSEL4	PORTINRSEL3	PORTINRSEL2	PORTINRSEL1	Reserved			SYSRSEL	CMRSEL	PWMRSEL	Reserved

Bit	Flag	Bit Name	Function	Read and write
b31~b12	Reserved	-	Read "0" when reading, write "0" when writing	R/W
b11	PORTINRSEL5	Port input 5 control release mode selection	0: Disable PWM output control when EMB_STATx. PORTINF5 = 0 0: Disable PWM output control when EMB_STATx. PORTINST5 = 0	R/W
b11	PORTINRSEL4	Port input 4 control release mode selection	0: Disable PWM output control when EMB_STATx. PORTINF4 = 0 0: Disable PWM output control when EMB_STATx. PORTINST4 = 0	R/W
b10	PORTINRSEL3	Port input 3 control release mode selection	0: Disable PWM output control when EMB_STATx. PORTINF3 = 0 0: Disable PWM output control when EMB_STATx. PORTINST3 = 0	R/W
b9	PORTINRSEL2	Port input 2 control release mode selection	0: Disable PWM output control when EMB_STATx. PORTINF2 = 0 0: Disable PWM output control when EMB_STATx. PORTINST2 = 0	R/W
b8	PORTINRSEL1	Port input 1 control release mode selection	0: Disable PWM output control when EMB_STATx. PORTINF1 = 0 0: Disable PWM output control when EMB_STATx. PORTINST1 = 0	R/W
b7~b4	Reserved	-	Read "0" when reading, write "0" when writing	R/W
b3	SYSRSEL	System error control release mode selection	0: Disable PWM output control when EMB_STATx. SYSF = 0 1: Disable PWM output control when EMB_STATx.SYSYST=0	R/W
b2	CMRSEL	Comparator comparison result controls the release mode selection	0: Disable PWM output control when EMB_STATx. CMPF = 0 1: Disable PWM output control when EMB_STATx.SYSYST=0	R/W
b1	PWMRSEL	PWM output control release mode selection	0: Disable PWM output control when EMB_STATx. PWMSF = 0 1: Disable PWM output control when EMB_STATx.SYSYST=0	R/W
b0	Reserved	-	Read "0" when reading, write "0" when writing	R/W

23 General Timer (TimerA)

23.1 Introduction

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 6 units of TimerA (U1~2 are 32-bit timers; U3~U6 are 16-bit timers), which can achieve a maximum of 24 PWM outputs.

23.2 Basic Block Diagram

The basic functions and features of TimerA are as Table 23-1 shown.

Table 23-1 Basic Functions and Features of TimerA

Waveform Mode	Sawtooth wave, triangular wave
Basic functions	• Direction of increments and decrements
	• Synchronous start counter
	• Benchflag 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

The basic block diagram of TimerA is as Figure 23-1 shown. In the figure, "< t >" is the unit number, that is, "< t >" is 1 to 6. In this section, when "< t >" is mentioned later, it refers to the unit number and is not repeated

This series of products is equipped with 6 units of TimerA, among which the bus clock and count clock of unit 1 to unit 4 are both PCLK0 (in this case, the PCLK shown in the block diagram refers to PCLK0); the bus clock and count clock of unit 5 to unit 6 are PCLK1 (in this case, the PCLK shown in the block diagram refers to PCLK1). The counting frequency division clock source of each unit and the sampling reference clock source of the port digital filter are the same as its bus clock and counting clock.

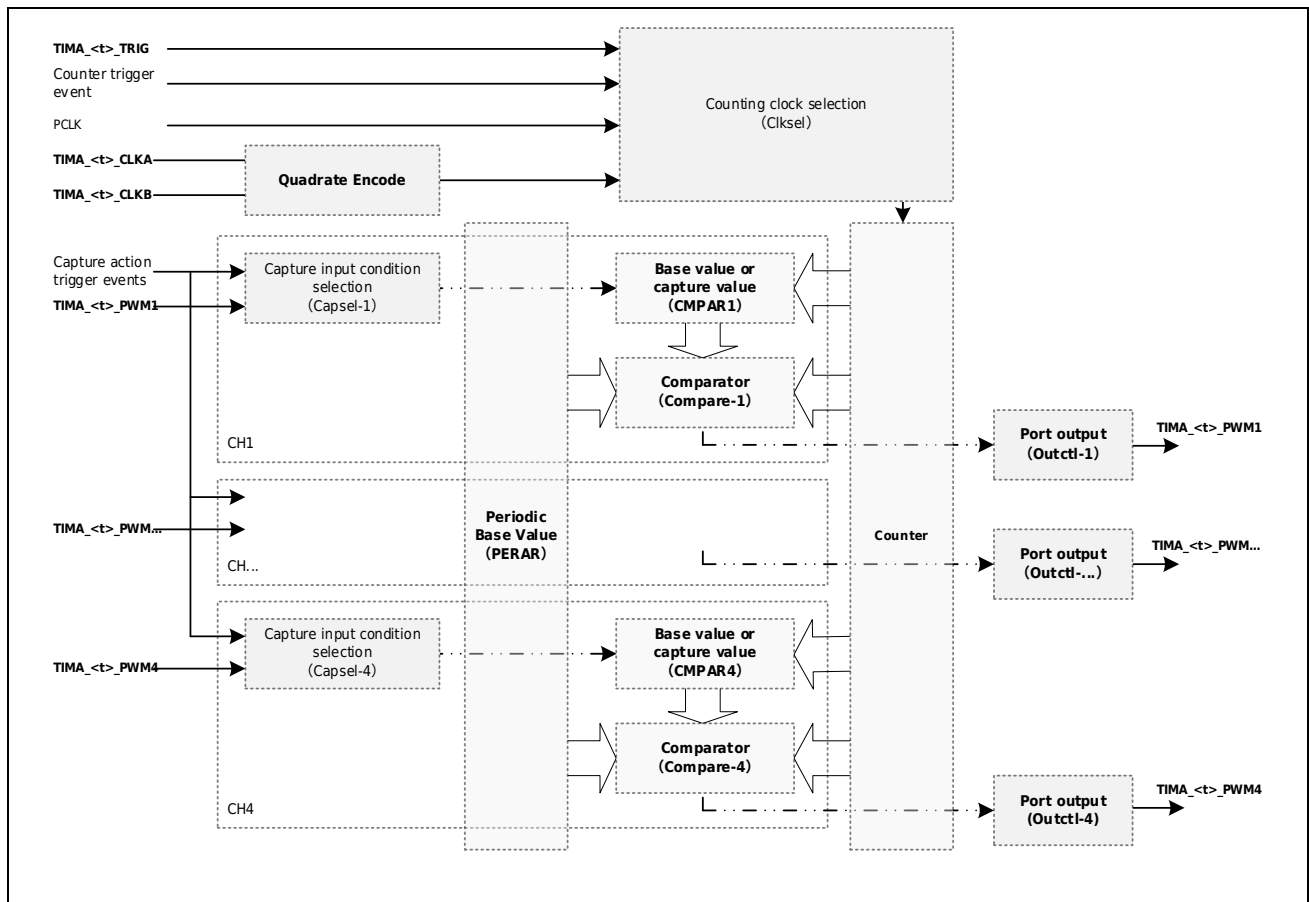


Figure 23-1 TimerA Basic Block Diagram

Table 23-2 Shown is the list of input and output ports of TimerA.

Table 23-2 TimerA Port List

Port Name	Direction	Function
TIMA_<t>_PWMMm	in or out	Capture input event port or PWM output port (m = 1 ~ 4)
TIMA_<t>_CLKA	in	Orthogonal Coded Count Event Input Port
TIMA_<t>_CLKB		
TIMA_<t>_TRIG	in	Hardware Trigger Startup, Stop, Zero Event Input Port

23.3 Functional Description

23.3.1 Waveform Mode

TimerA has two basic counting wave modes: Serrated wave mode and triangular wave mode. The basic waveforms of the two waveform modes Figure 23-2 Figure 23-3 are shown in.

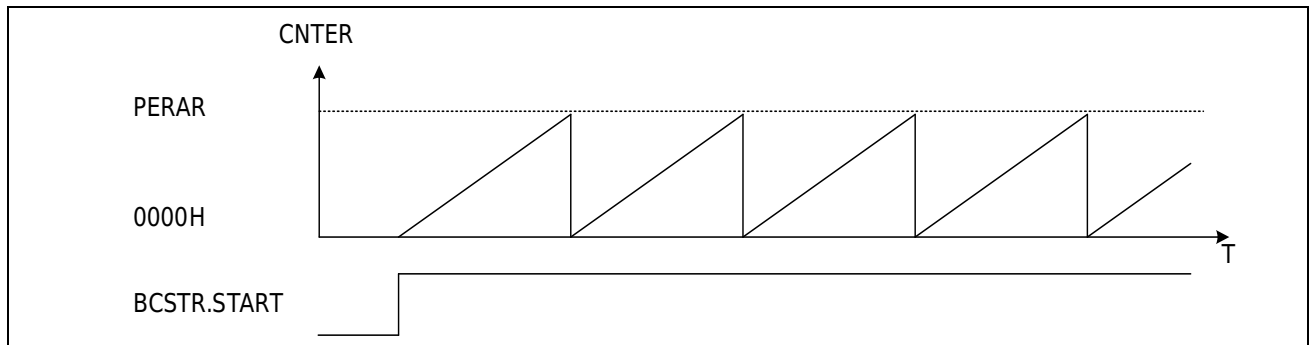


Figure 23-2 Sawtooth Waveform (Counting Up)

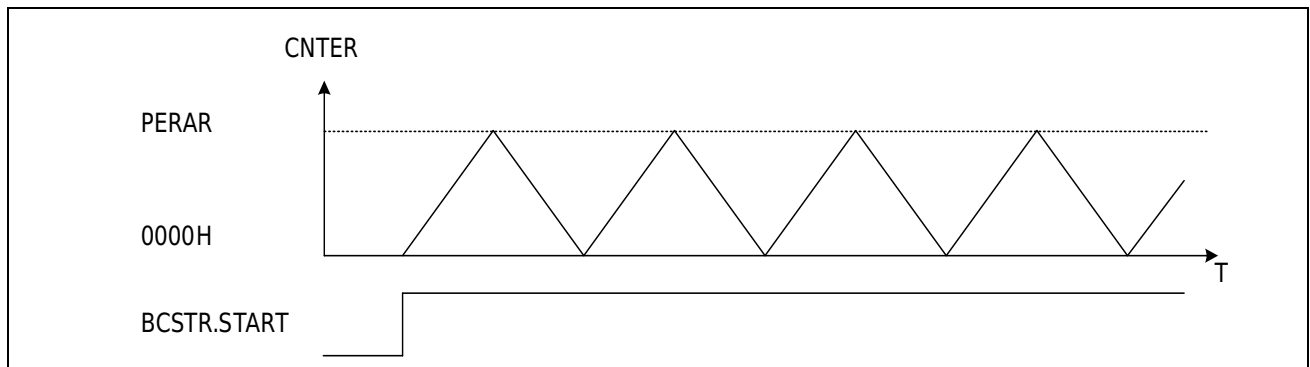


Figure 23-3 Triangle Waveform

23.3.2 Timer Selection

TimerA's counting clock has the following options:

- 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 division of PCLK (BCSTR.LCKDIV [3:0] setting)
- TIMA_ < t > _TRIG Port Event Input (HCUPR [9: 8] or HCDOR [9: 8] setting)
- Internal counter trigger event input (HCUPR [10] or HCDOR [10] settings)
- Count overflow or count underflow event input of the symmetric unit (set by HCUPR [12:11] or HCDOR [12:11])
- TIMA_ < t > _CLKA, TIMA_ < t > _CLKB port orthogonal code input (HCUPR [7: 0] or HCDOR [7: 0] setting)

The counting clock source is a software counting mode, and the counting clock source is b, c, d, e hardware counting mode. When the counting clock selects d, it is mostly used in the revolution counting mode of three-phase quadrature encoding counting (see [Position overflow counting] and [Mixed counting] chapters), and it 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.

23.3.3 Comparison Output

Each Timer A unit contains a 4-channel comparison output (TIMA_< t >_PWMn) that outputs the specified level when the count value matches the baseline value. TMRA_< t >_CMPARn register corresponds to the count baseline value of TIMA_< t >_PWMn output port, respectively. The TIMA_< t >_PWMn port outputs the specified level (n = 1 ~ 4) when the timer count is equal to TMRA_< t >_CMPARn.

TIMA_< t >_PWMn port is controlled by port control register (PCONRn) STAC, STPC, CMPC, PERC, FORC bit setting (n = 1 ~ 4). Figure 23-4 This is an instance of the comparison output operation of Unit 1.

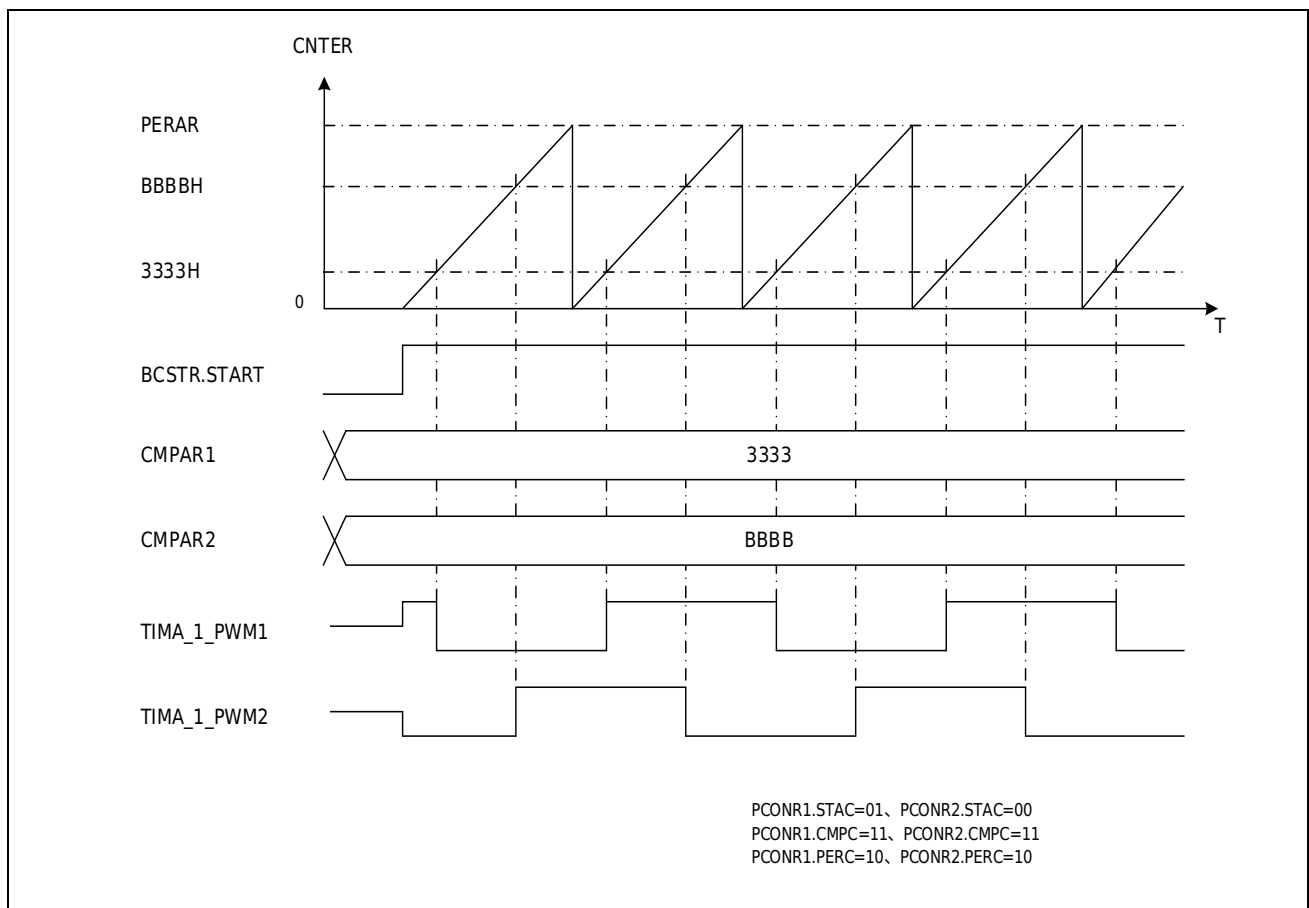


Figure 23-4 Compare Output Action

23.3.4 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 (CCONRn) 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 (CMPARn) (n = 1 ~ 4).

The capture input condition can select the internal capture action trigger event (selected by TMRA_TRGSEL0~3 registers, refer to the register description section for details), TIMA_<t>_PWMn port input, etc. The specific condition selection can be set through the HICP bit of the capture control register (CCONRn) (n=1~4). Figure 23-5 For capturing input action instance.

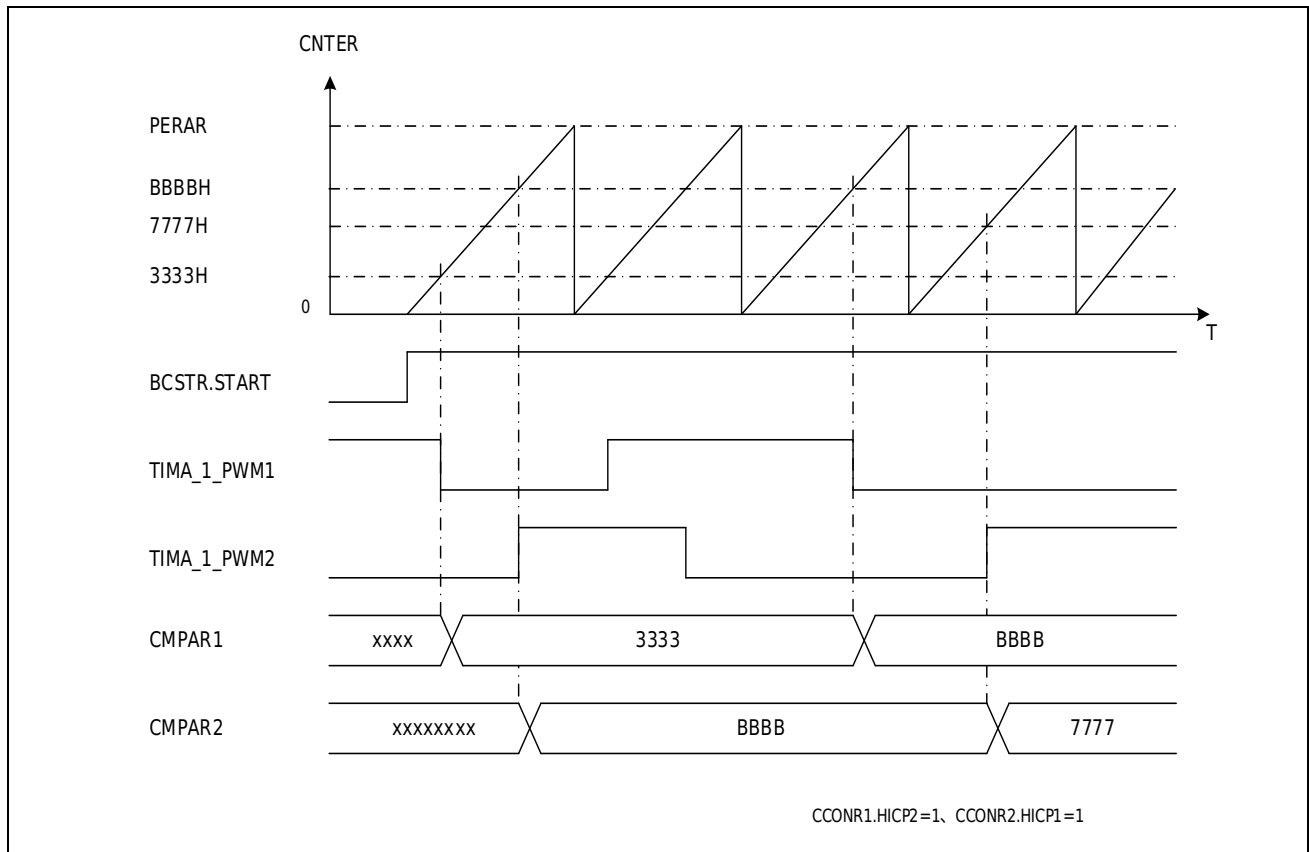


Figure 23-5 Capturing Input Actions

23.3.5 Synchronous Startup

The 6-unit TimerA on this product can realize software synchronous start or hardware synchronous start. When the current unit is unit m, unit m can choose to start synchronously with unit n (when m=2, 4, 6, n=1, 3, 5).

When the BCSTR.LSYNST bit in unit m is set to 1, the synchronous start function of unit m and unit n is valid. In this case, if the BCSTR.START bit of the unit n is 1, the counter of the unit m starts to count synchronously. If any bit in the HCONR.HSTA1 ~ 0 of the hardware setting unit n is 1, and the corresponding hardware event of the unit n occurs, the counter of the synchronization unit (unit m) starts hardware synchronization counting. When the hardware synchronous count start function is selected, the corresponding bits of HCONR.HSTA1~0 of the synchronized unit (unit m) must also be set to valid (when m=2, 4, 6, n=1, 3, 5).

Figure 23-6 is an instance of software synchronous startup when BCSTR.LSYNST = 1 in unit 4 is set.

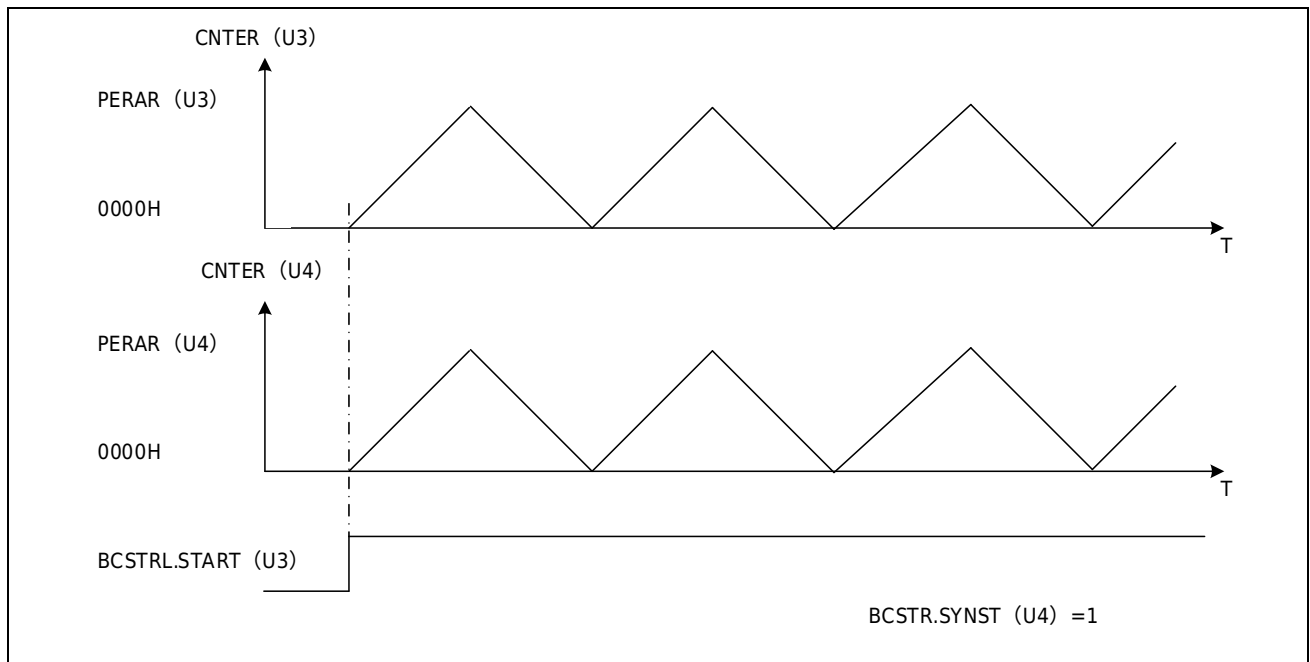


Figure 23-6 Software Synchronization Action

23.3.6 Digital Filtering

The input ports of TIMA_< t >_CLKA, TIMA_< t >_CLKB, TIMA_< t >_TRIG and TIMA_< t >_PWMn have digital filtering functions. The enable of filter function and the selection of filter clock can be realized by setting the corresponding bits of filter control register (FCONR) and capture control register (CCONRn) (n = 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 23-7 shows an instance of TIMA_1_CLKA port filtering operation.

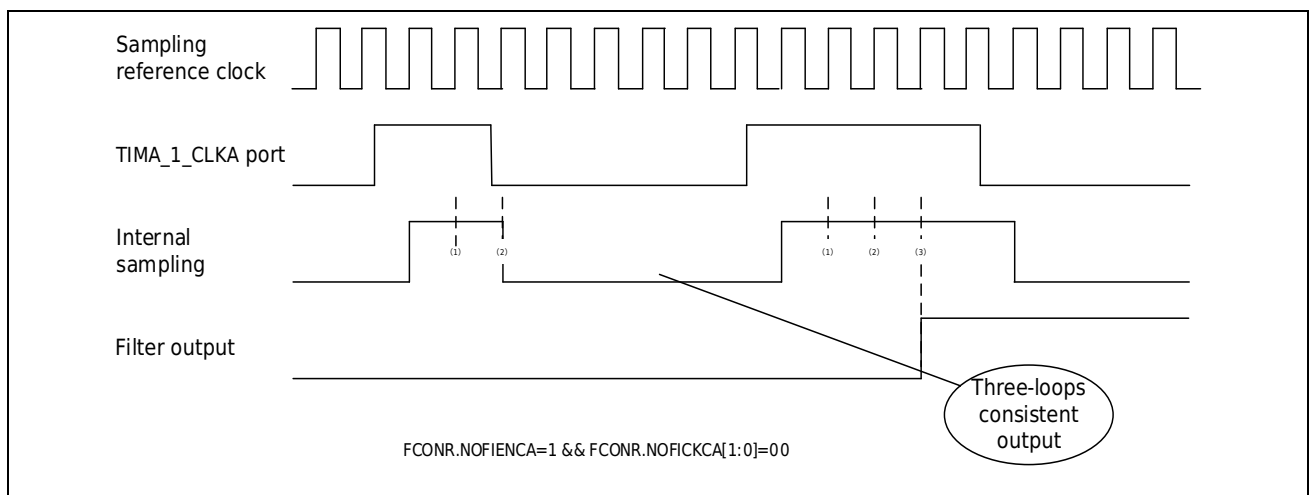


Figure 23-7 Filtering Function of Clock Input Port

23.3.7 Cache Function

Two benchflag registers (CMPARn) of TimerA can implement cache function in pairs ($n = 1 \sim 4$). That is, CMPAR2 is used as the cache reference value of CMPAR1, and CMPAR4 is used as the cache reference value of CMPAR3. The cache control register (BCONRm) controls the two sets of cache functions respectively ($m=1\sim2$).

When the BEN bit of the cache control register (BCONRm) is set, the cache function becomes valid ($m=1\sim2$). When the counter reaches a certain point in time, a cache transmission occurs (CMPAR4/2→CMPAR3/1). The "specific point in time" has the following circumstances:

- In hardware counting mode, the count reaches the overflow point (BCSTR.LDIR=1) or the underflow point (BCSTR.LDIR=0)
- In sawtooth counting mode (BCSTR.MODE=0), the counter counts to the overflow point (BCSTR.LDIR=1) or underflow point (BCSTR.LDIR=0)
- In triangle wave counting mode (BCSTR.MODE=1), count to peak point (BCSTR.LDIR=1 && BCONRn.BSE0=1) ($n=1\sim2$)
- In triangle wave counting mode (BCSTR.MODE=1), count to valley point (BCSTR.LDIR=0 && BCONRn.BSE1=1) ($n=1\sim2$)
- Zeroing occurs when hardware counting mode or sawtooth wave counting mode is used.

The following Figure 23-8 is a schematic diagram of cache transfer in sawtooth mode.

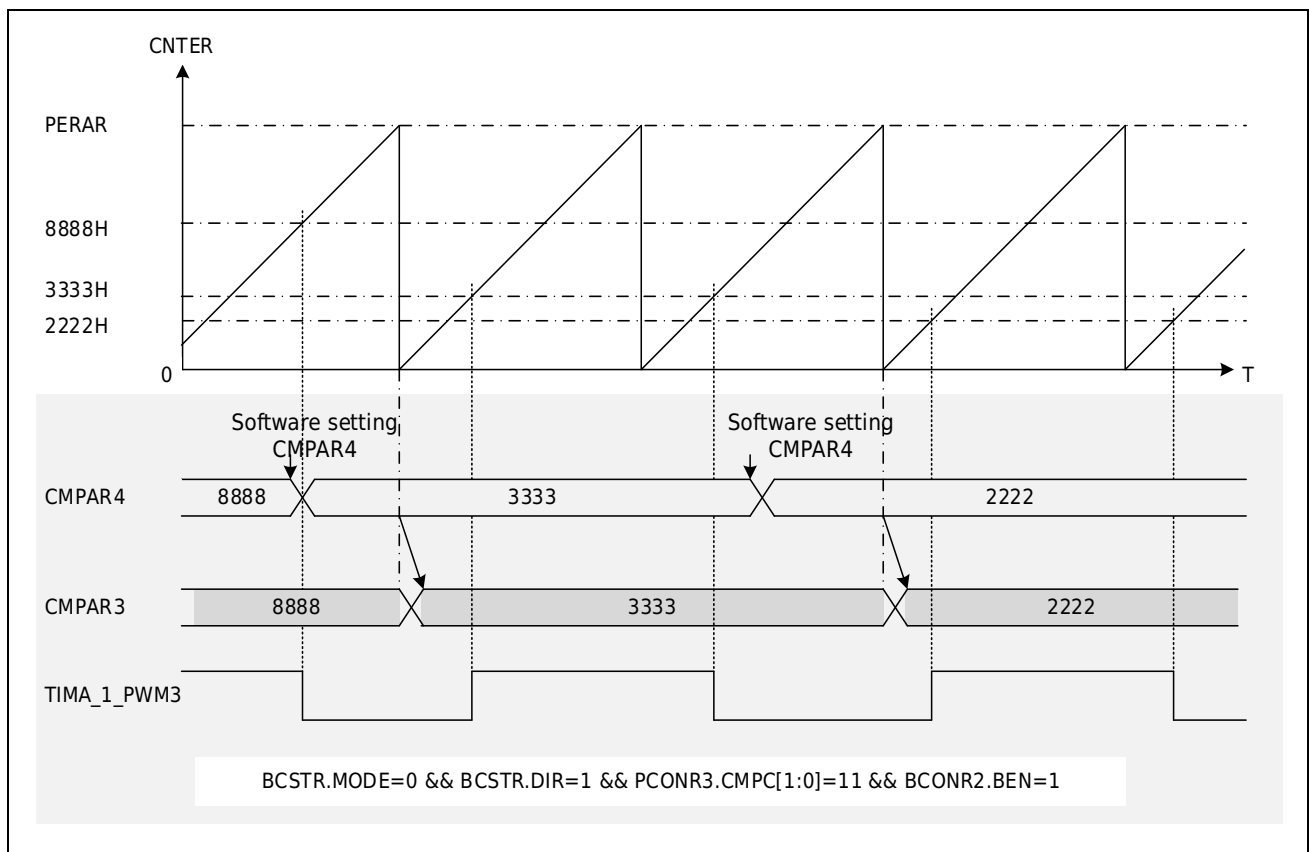


Figure 23-8 Cache Action in Sawtooth Wave Mode

23.3.8 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 1 and 2 (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 23-9 shown is a schematic diagram of unit 1, 2 cascade counting.

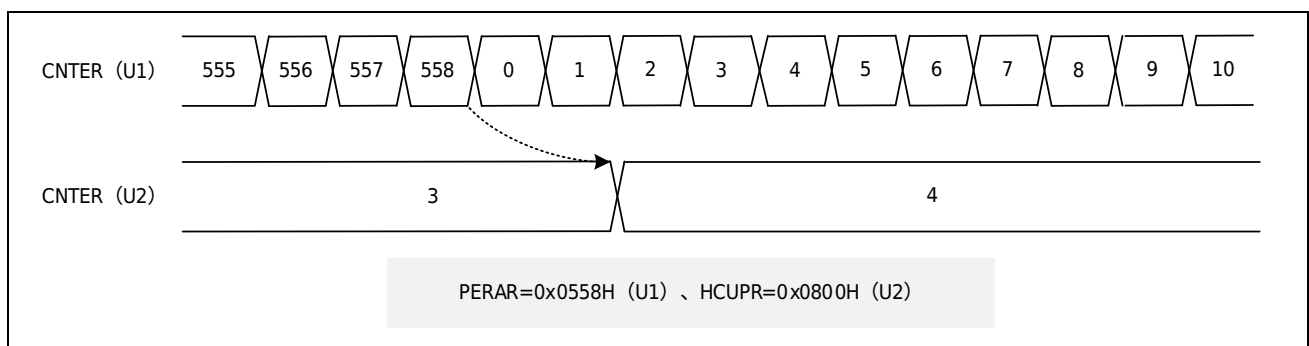


Figure 23-9 32-bit Cascade Counting Action

23.3.9 PWM Output

23.3.9.1 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 period point). Setting, for example, to flip when a baseline comparison matches (PCONR.CMPC = 11) and to flip when a period baseline comparison matches (PCONR.PERC = 11) enables a single-sided aligned PWM output to be generated in one period.

The Figure 23-10 shows an instance of a single-side aligned PWM output waveform in sawtooth wave mode.

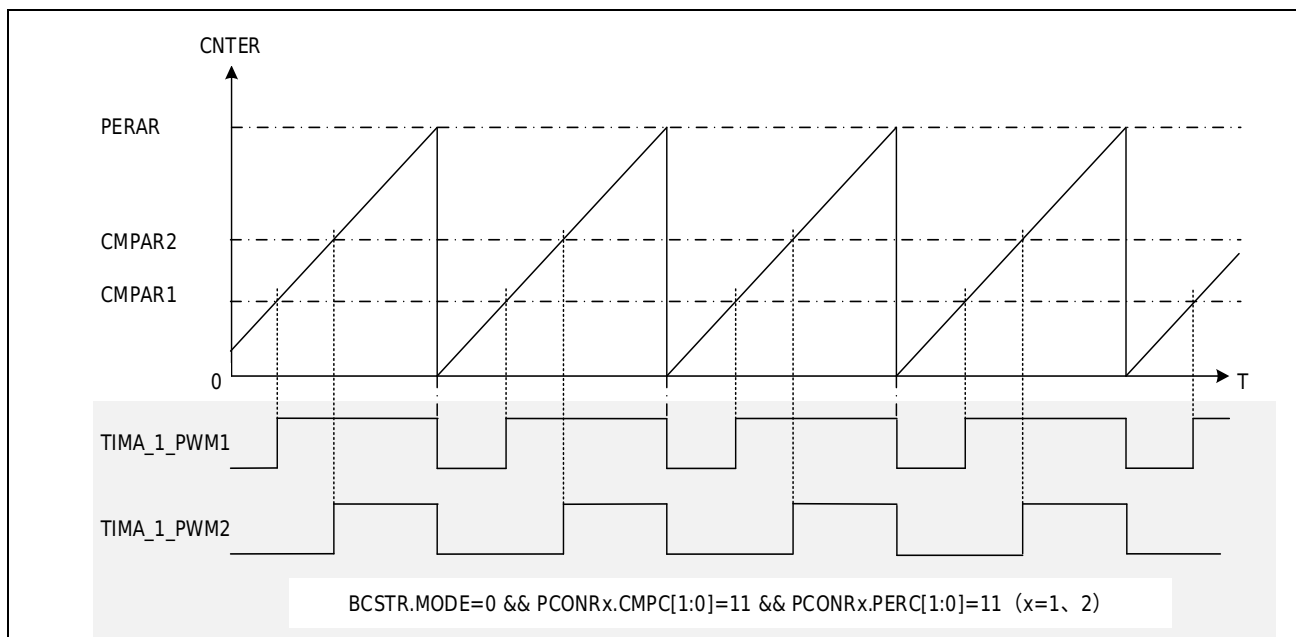


Figure 23-10 Single-edge Aligned PWM Output Instance

23.3.9.2 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 23-11 The figure shows an instance 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.

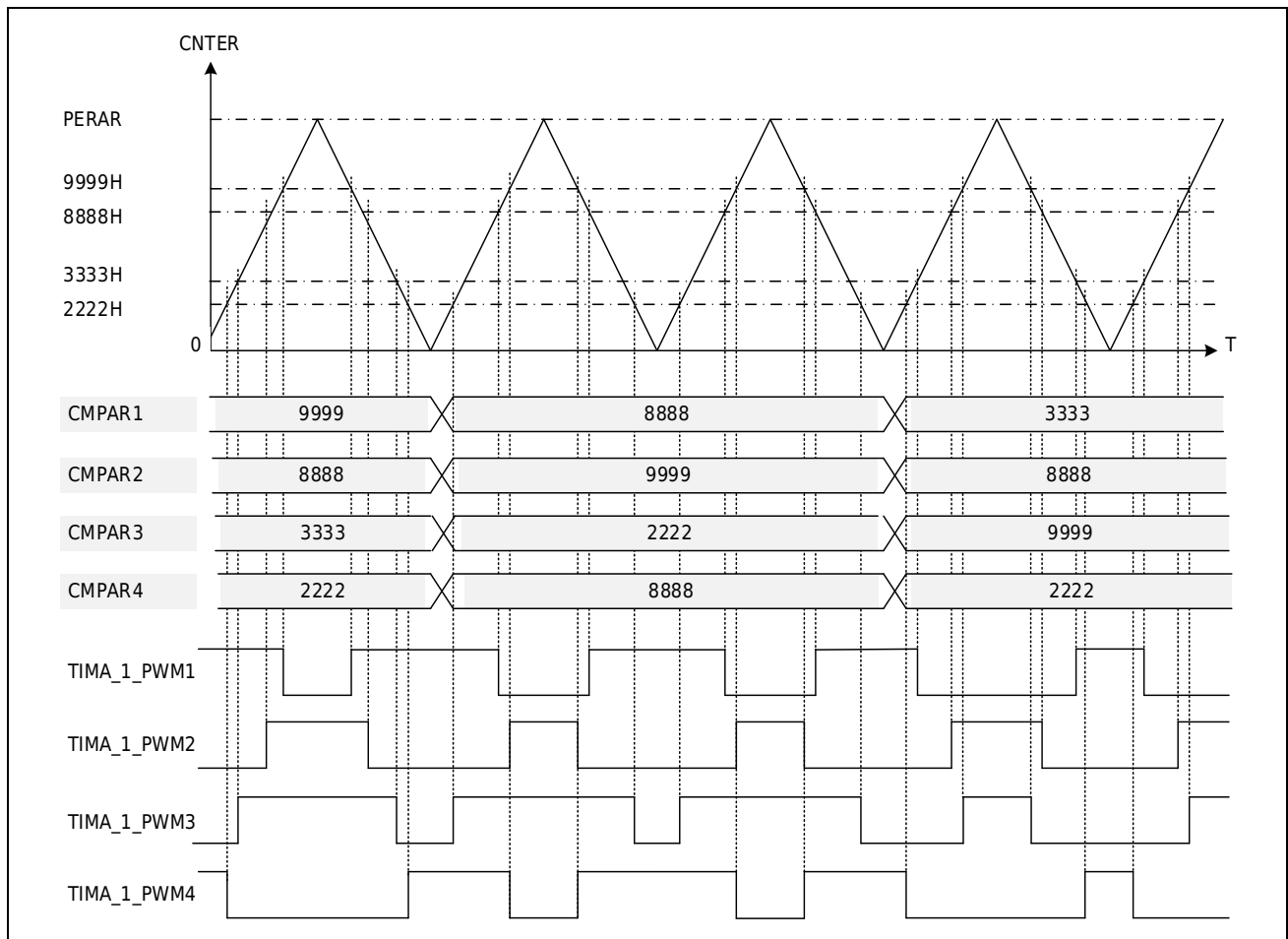


Figure 23-11 Instance of Bilateral Symmetrical PWM Output

23.3.10 Orthogonal Coding Count

The TIMA_<t>_CLKA input is regarded as the AIN input, the TIMA_<t>_CLKB input is regarded as the BIN input, and the TIMA_<t>_TRIG input is regarded as the ZIN input, and TimerA can realize the quadrature encoding count 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 revolution counting mode, every two units are combined with each other (combination of units 1 and 2; combination of units 3 and 4; combination of units 5 and 6), and the position counting unit and revolution counting unit in the combination can be specified arbitrarily.

The counting conditions of AIN and BIN are enabled by setting the orthogonal relationship between TIMA_<t>_CLKA and TIMA_<t>_CLKB in the hardware increment event selection register (HCUPR) and hardware decrement event selection register (HCDOR); ZIN's The input action realizes the clearing of the position timer by setting the clear enable bit of the hardware trigger event selection

register (HCONR) of the position counting unit, and realizes the counting of the revolution timer by setting the hardware increment event selection register (HCUPR) of the revolution unit..

23.3.10.1 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 23-12 below.

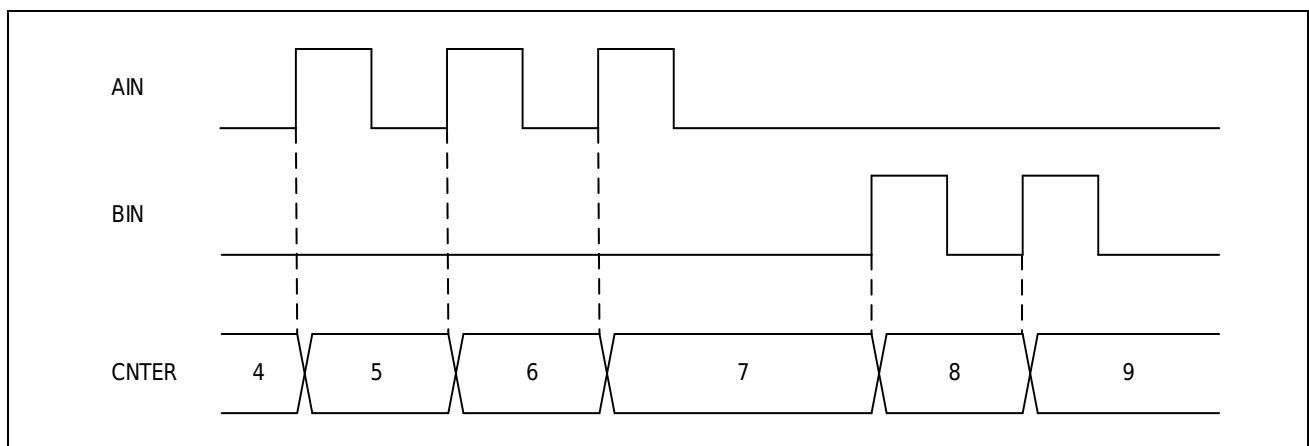


Figure 23-12 Position Mode - Basic Counting

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 Figure 23-13~Figure 23-15 below

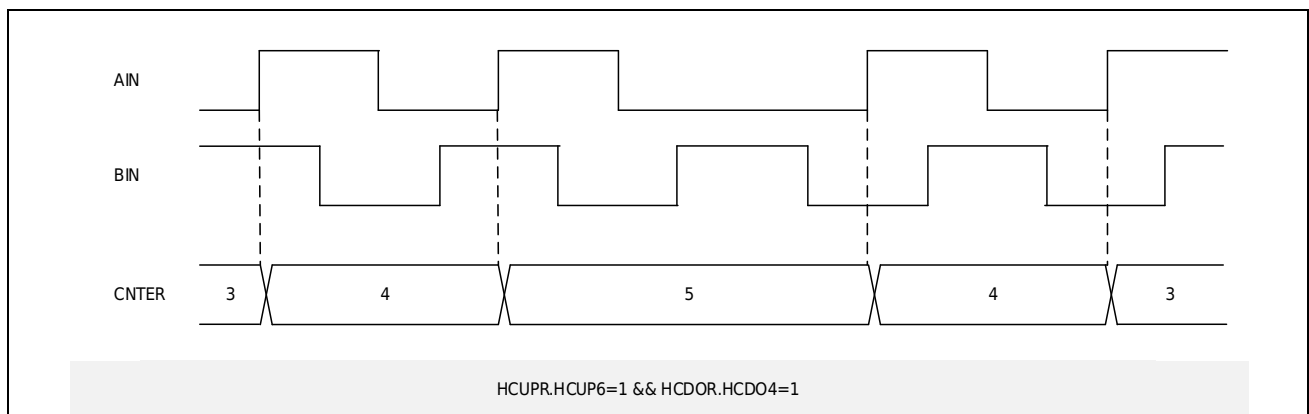


Figure 23-13 Position counting mode - phase difference counting (1 times counting)

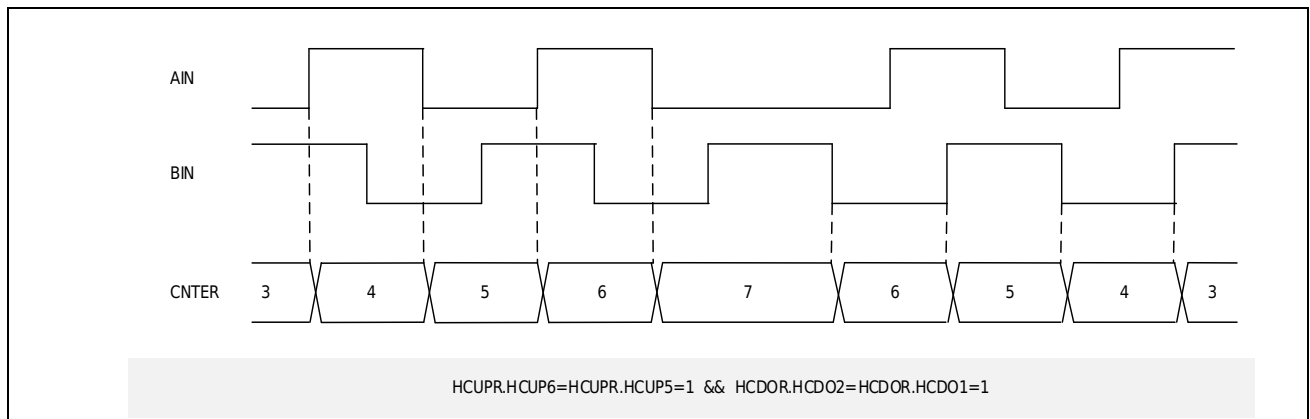


Figure 23-14 Position Counting Mode - Phase Difference cCounting (2 times Counting)

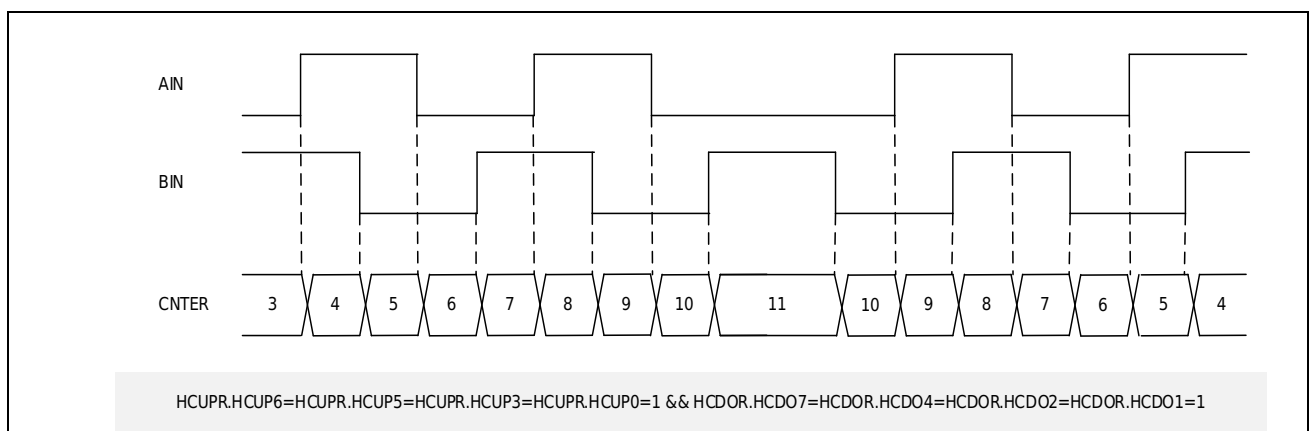


Figure 23-15 Position Counting Mode - Phase Difference Counting (4 times Counting)

Direction Counting

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 figureFigure 23-16below.

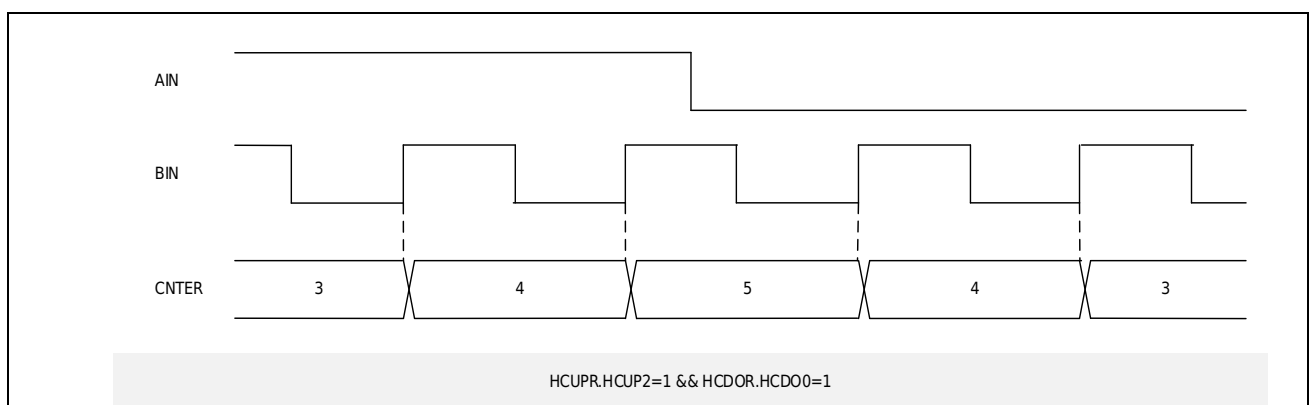


Figure 23-16 Position Counting Mode - Direction Counting

23.3.10.2 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 Counting

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 23-17 below.

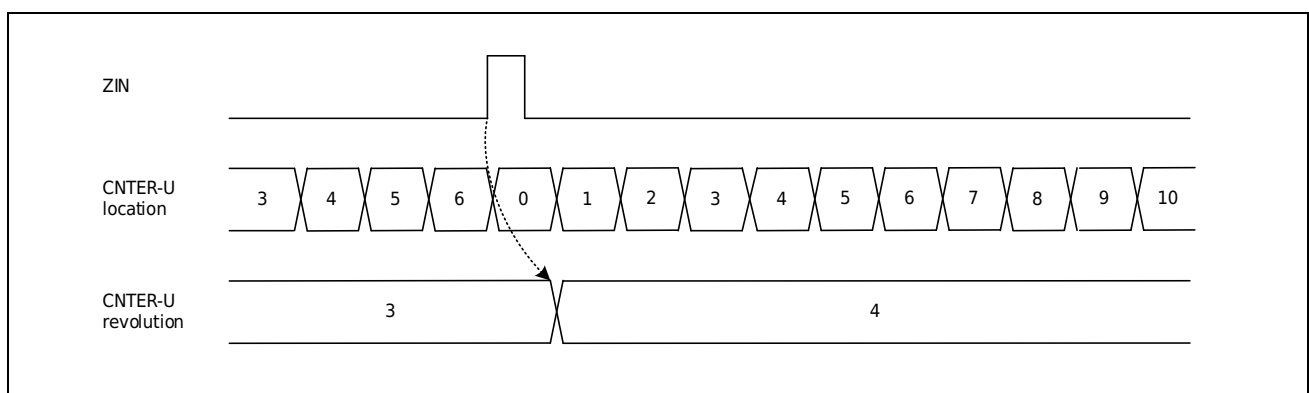


Figure 23-17 Revolution Counting Mode - Z phase Counting

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 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 23-18 below.

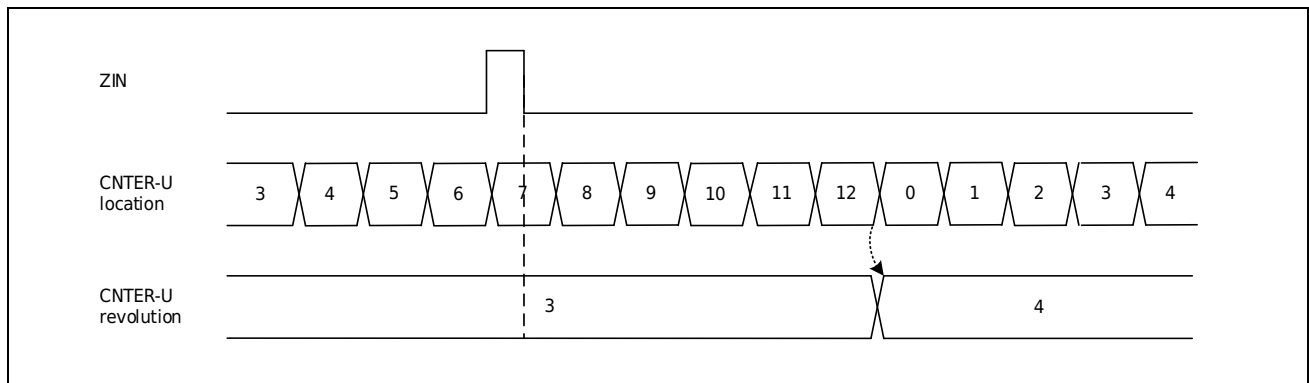


Figure 23-18 Revolution Counting Mode - Position Overflow Counting

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 FigureFigure 23-19below.

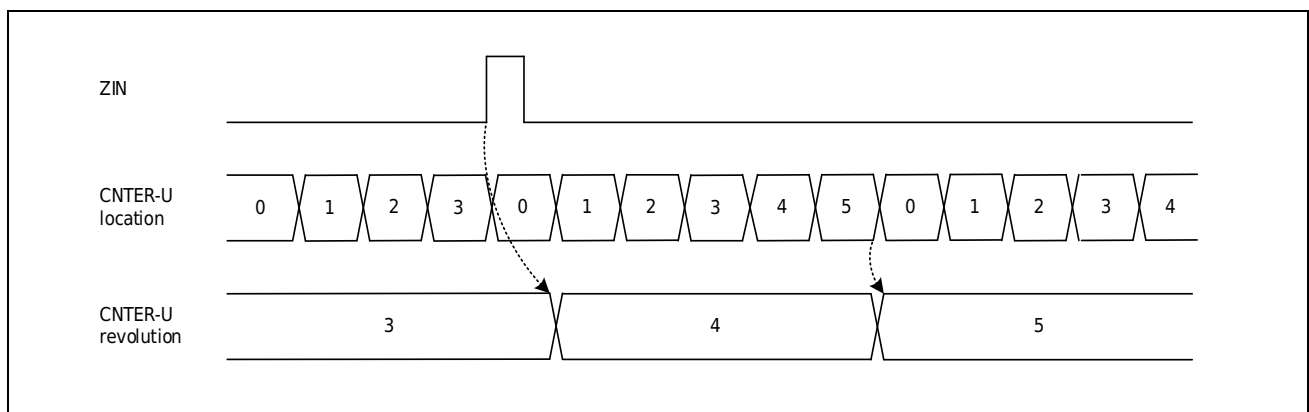


Figure 23-19 Revolution Counting Mode - Mixed Counting

23.3.11 Infrared Function (IFRP)

the LED that emits infrared light. The LED can transmit infrared data to realize infrared remote control and is controlled by U5 and U6 (unit 5 and unit 6) of TimerA. This function integrates the outputs of U5 and U6 of TimerA to generate an infrared signal. PWM1 of U5 outputs a low-frequency PWM signal to generate a modulated envelope signal. PWM1 of U6 outputs a high-frequency PWM signal to generate a carrier signal. Before generating the signal, you need to enable the PWM output of the above channels and select the IFRP function on the PORT pin. The output signal of IFRP is delayed by one PCLK1 clock compared to the input TIMA_6_PWM1 high-frequency carrier signal. The duty period of the two input signals can be changed. When the TIMA_5_PWM1 low-frequency modulation envelope signal level is high, the output signal of IFRP is inverted relative to the TIMA_6_PWM1 high-frequency carrier signal. The output signal of IFRP will maintain the integrity of the high-frequency carrier signal. Even if the TIMA_5_PWM1 low-frequency modulation envelope signal level changes from high to low, there will be no incomplete output waveform relative to the TIMA_6_PWM1 high-frequency carrier signal. As shown in Figure 23-20 below.

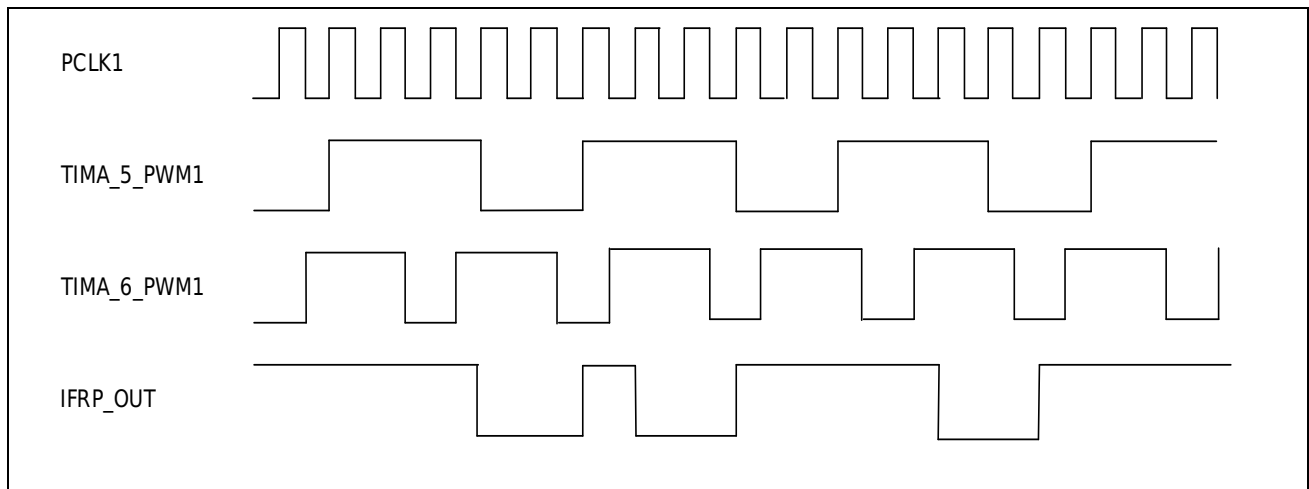


Figure 23-20 Schematic Diagram Of Infrared Output Signal Characteristics

Note:

- The output signal of IFRP is determined by U5 of TimerA and PWM1 of U6.
- In order to obtain the correct infrared signal, U5 should generate a low-frequency modulation envelope signal and U6 should generate a high-frequency carrier signal.

23.4 Interrupt and Event Description

TimerA contains three interrupt outputs and three event outputs: One comparison match interrupt and event, two period match interrupt and event.

23.4.1 Compare Matched Interrupt and Events

The corresponding bit (STFLR.CMPFn) in the status flag register (STFLR) is set to 1 when the comparison of the baseline value register (CMPARn) and the count value matches. At this time, if the corresponding bit (ICONR.ITENn) 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.ETENn) of the event control register (ECONR) is set to 1, the corresponding event request (TMRA_<t>_CMP) will be triggered (n=1~4).

When the capture control register (CCONRn) selected capture input valid condition occurs, the capture input action occurs. At this time, if the corresponding bit (ICONR.ITENn) of the interrupt control register (ICONR) is set to 1, the corresponding interrupt request (TMRA_<t>_CMP) is triggered; if the corresponding bit (ECONR.ETENn) of the event control register (ECONR) is set to 1, the corresponding event request (TMRA_<t>_CMP) will be triggered (n=1~4).

Comparison-match interrupt and comparison-match events of the four baseline values within each cell are not output independently. Comparison-match interrupt is aggregated into a interrupt module by OR logic (refer to INTC section), and comparison-match events are aggregated into an event output by OR logic to select another module to trigger.

23.4.2 Periodic Matching Interrupt and Events

The OVFF or UDF 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 period point to be output to the interrupt module (INTC); the period match event has no corresponding enable bit control, and the period match event (TMRA_<t>_OVF and TMRA_<t>_UDF) output is triggered at the corresponding counting period point to select triggering other modules.

23.5 Register Description

Table 23-3 Shown is the register list of the TimerA module.

Table 23-3 TimerA Register List

BASE ADDR:

0x4003A000(U1), 0x4003A400(U2), 0x4003A800(U3),

0x4003AC00(U4), 0x40026000(U5), 0x40026400(U6)

Register Name	Symbol	Offset	Bit Width	Reset Value
General count register	TMRA_CNTER	0x0000	16	0x0000
Period reference value register	TMRA_PERAR	0x0004	16	0xFFFF
Compare base value register 1	TMRA_CMPAR1	0x0040	16	0xFFFF
Compare base value register 2	TMRA_CMPAR2	0x0044	16	0xFFFF
Compare base value register 3	TMRA_CMPAR3	0x0048	16	0xFFFF
Compare base value register 4	TMRA_CMPAR4	0x004C	16	0xFFFF
Control status register L	TMRA_BCSTRL	0x0080	8	0x02
Control Status Register H	TMRA_BCSTRH	0x0081	8	0x00
Interrupt control register	TMRA_ICONR	0x0090	16	0x0000
Event control register	TMRA_ECONR	0x0094	16	0x0000
Filter control register	TMRA_FCONR	0x0098	16	0x0000
Status flag register	TMRA_STFLR	0x009C	16	0x0000
Cache Control Register 1	TMRA_BCONR1	0x00C0	16	0x0000
Cache Control Register 2	TMRA_BCONR2	0x00C8	16	0x0000
Capture Control Register 1	TMRA_CCONR1	0x0100	16	0x0000
Capture Control Register 2	TMRA_CCONR2	0x0104	16	0x0000
Capture Control Register 3	TMRA_CCONR3	0x0108	16	0x0000
Capture Control Register 4	TMRA_CCONR4	0x010C	16	0x0000
Port Control Register 1	TMRA_PCONR1	0x0140	16	0x0000
Port Control Register 2	TMRA_PCONR2	0x0144	16	0x0000
Port Control Register 3	TMRA_PCONR3	0x0148	16	0x0000
Port Control Register 4	TMRA_PCONR4	0x014C	16	0x0000
Hardware trigger event select register	TMRA_HCONR	0x0084	16	0x0000
Hardware increment event selection register	TMRA_HCUPR	0x0088	16	0x0000
Hardware decrememt event selection register	TMRA_HCDOR	0x008C	16	0x0000

Note:

- U1~2 units The bit width of the period reference value register and the comparison reference value register are all 32 bits, and the reset values are different from those of the U3~U6 units. The reset value of the general count value register is 0x00000000 , and

the reset value of the period reference value register and the comparison reference value register is 0xFFFFFFFF.

Table 23-4 Internal Trigger Event TMRA_TRGSEL Selection Relationship Correspondence Table

Number of units	Event Classification	Corresponding Selection Register
U1	Counter trigger event: TRGSEL selection event indicated by HCONR, HCUPR, HCDOR	TMRA_TRGSEL0
U2		TMRA_TRGSEL1
U3		TMRA_TRGSEL2
U4		TMRA_TRGSEL3
U5		TMRA_TRGSEL0
U6		TMRA_TRGSEL1
U1	Capture action trigger events: TRGSEL selection event referred to in CCONR	TMRA_TRGSEL1
U2		TMRA_TRGSEL0
U3		TMRA_TRGSEL3
U4		TMRA_TRGSEL2
U5		TMRA_TRGSEL1
U6		TMRA_TRGSEL0

23.5.1 Generic Count Value Register (TMRA _ CNTER)

U1~2: Reset value: 0x00000000

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	Flag		Bit Name		Function										Read and write
b31~b0	CNT [31:0]		Count value		Current timer count										R/W

U3~6: Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CNT [15:0]															
Bit	Flag		Bit Name		Function										Read and write
b15~b0	CNT [15:0]		Count value		Current timer count										R/W

23.5.2 Periodic Reference Value Register (TMRA _ PERAR)

U1~2: Reset value: 0xFFFFFFFF

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	Flag		Bit Name		Function										Read and write
b31~b0	PER [31:0]		Count period value		Set the count period value for each round										R/W

U3~6: Reset value: 0xFFFF

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PER [15:0]															
Bit	Flag		Bit Name		Function										Read and write
b15~b0	PER [15:0]		Count period value		Set the count period value for each round										R/W

23.5.3 Comparison Reference Value Register (TMRA_CMPARm, m=1~4)

U1~2: Reset value: 0xFFFFFFFF

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	Flag		Bit Name		Function		Read and write								
b31~b0	CMP [31:0]		Count baseline		Set baseline		R/W								

U3~6: Reset value: 0xFFFF

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CMP [15:0]															
Bit	Flag		Bit Name		Function		Read and write								
b15~b0	CMP [15:0]		Count baseline		Set baseline		R/W								

23.5.4 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	Flag	Bit Name	Function	Read and write
b7~b4	CKDIV [3:0]	Counting clock selection	0000: PCLK	RW
			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	
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
			0: Sawtooth mode 1: Triangular wave mode	
b2	MODE	Counting mode	0: Counter down 1: Counter up Note: In triangle wave mode, this bit indicates the counter direction status.	RW
b1	DIR	Counting direction	0: Timer off 1: Timer Startup Note 1: This bit will automatically change to 0 if the hardware stop condition is valid Note 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:

- In triangle wave mode, do not write to this register while the timer is in operation.

23.5.5 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	Flag	Bit Name	Function	Read and write
b7	UDFF	Underflow flag	0: Count down, no count down 1: Count down, count down	RW
b6	OVFF	Overflow flag	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

23.5.6 Interrupt Control Register (TMRA_ICONR)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved												IT EN4	IT EN3	IT EN2	IT EN1

Bit	Flag	Bit Name	Function	Read and write
b15~b4	Reserved	-	Read as "0", write as "0"	R/W
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 CMPAR1 register is equal to the count value, or when a capture input event occurs	R/W
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 CMPAR1 register is equal to the count value, or when a capture input event occurs	R/W
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 CMPAR1 register is equal to the count value, or when a capture input event occurs	R/W
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 CMPAR1 register is equal to the count value, or when a capture input event occurs	R/W

23.5.7 Event control register (TMRA _ ECONR)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved												ET EN4	ET EN3	ET EN2	ET EN1

Bit	Flag	Bit Name	Function	Read and write
b15~b4	Reserved	-	Read as "0", write as "0"	R/W
b3	ETEN4	Count Match Event Enabled 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	R/W
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	R/W
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	R/W
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	R/W

23.5.8 Filter Control Register (TMRA _ FCONR)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	NOFI CKCB [1:0]		NOFI ENCB	-	NOFI CKCA [1:0]		NOFI ENCA	-	-	-	-	-	NOFI CKTG [1:0]		NOFI ENTG

Bit	Flag	Bit Name	Function	Read and write
b15	Reserved	-	Read as "0", write as "0"	R/W
b14~b13	NOFICKCB [1:0]	Filter Sampling Reference Clock Selection CB	00: PCLK 01: PCLK/4 10: PCLK/16 11: PCLK/64	R/W
b12	NOFIENCB	Capture input port filter CB	0: Invalid input filtering function for TIMA _ < t > _ CLKB port 1: TIMA _ < t > _ CLKB port input filtering enabled	R/W
b11	Reserved	-	Read as "0", write as "0"	R/W
b10~b9	NOFICKCA [1:0]	Filter sampling reference clock selection CA	00: PCLK 01: PCLK/4 10: PCLK/16 11: PCLK/64	R/W
b8	NOFIENCA	Capture input port filter CA	0: Invalid input filtering function of TIMA _ < t > _ CLKA port 1: TIMA _ < t > _ CLKA port input filtering function enabled	R/W
b7~b3	Reserved	-	Read as "0", write as "0"	R/W
b2~b1	NOFICKTG [1:0]	Filter sampling reference clock selection TG	00: PCLK 01: PCLK/4 10: PCLK/16 11: PCLK/64	R/W
b0	NOFIENTG	Capture input port filter TG	0: TIMA _ < t > _ TRIG input port filter invalid 1: TIMA _ < t > _ TRIG input port filtering function enabled	R/W

23.5.9 Status Flag Register (TMRA _ STFLR)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved				ICP F4	ICP F3	ICP F2	ICP F1	Reserved				CMP F4	CMP F3	CMP F2	CMP F1

Bit	Flag	Bit Name	Function	Read and write
b15-b12	Reserved	-	Read as "0", write as "0"	R
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	R/W
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	R/W
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	R/W
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	R/W
b7-b4	Reserved	-	Read as "0", write as "0"	R
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	R/W
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	R/W
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	R/W
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	R/W

Note:

- The flag bit in this register can only be written to 0, not 1.

23.5.10 Cache Control Register (TMRA_BCONRm, m=1~2)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved													BSE1	BSE0	BEN

Bit	Flag	Bit Name	Function	Read and write
b15~b3	Reserved	-	Read as "0", write as "0"	R/W
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)	R/W
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)	R/W
b0	BEN	Cache Enable	0: Invalid Cache Function for CMPARn Benchflag 1: The Cache Function of CMPARn Benchflag (n = 1, 3)	R/W

23.5.11 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
-	NOFICKCP [1:0]	NOFIENCP	-	-	HICP4	HICP3	-	HICP2	HICP1	HICP0	-	-	-	-	CAPMD

Bit	Flag	Bit Name	Function	Read and write
b15	Reserved	-	Read as "0", write as "0"	R/W
b14~b13	NOFICKCP [1:0]	Filtering sampling reference clock selection CP	00: PCLK 01: PCLK/4 10: PCLK/16 11: PCLK/64	R/W
b12	NOFIENCP	Capture input port filtering CP	0: Invalid input filtering function of TIMA_<t>_PWMn port 1: TIMA_<t>_PWMn port input filtering enabled (n = 1 ~ 4)	R/W
b11~b10	Reserved	-	Read as "0", write as "0"	R/W
b9	HICP4	Capture input condition enable 4	0: When the TIMA_<t>_TRIG port input is sampled to the falling edge, channel m does not capture the input action 1: When the TIMA_<t>_TRIG port input is sampled to the falling edge, the channel m generates a capture input action 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	R/W
b8	HICP3	Capture input condition enable 3	0: When the TIMA_<t>_TRIG port input is sampled to the rising edge, the channel m does not capture the input action 1: When the TIMA_<t>_TRIG port input is sampled to a rising edge, channel m will generate a capture input action 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	R/W
b7	Reserved	-	Read as "0", write as "0"	R/W
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.	R/W
b5	HICP1	Capture input condition enable 1	0: No capture input action occurs when TIMA_<t>_PWMn port input is sampled to the falling edge 1: When the input of TIMA_<t>_PWMn port is sampled to the falling edge, the capture input action is generated. (n = 1 ~ 4)	R/W
b4	HICP0	Capture input condition enable 0	0: No capture input action occurs when TIMA_<t>_PWMn port input is sampled to the rising edge 1: When the input of TIMA_<t>_PWMn port reaches the rising edge, the capture input action is generated. (n = 1 ~ 4)	R/W
b3~b1	Reserved	-	Read as "0", write as "0"	R/W
b0	CAPMD	Functional mode selection	0: Comparison output function 1: Capture input function	R/W

23.5.12 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
-	-	-	OUTEN	-	-	FORC [1:0]	PERC [1:0]	CMPC [1:0]	STPC [1:0]	STAC [1:0]					

Bit	Flag	Bit Name	Function	Read and write
b15~b13	Reserved	-	Read as "0", write as "0"	R/W
b12	OUTEN	Output enable	0: Invalid output of TIMA_ < t > _ PWMn port when PWM output function 1: Output Validity of TIMA_ < t > _ PWMn Port with PWM Output Function (n = 1 ~ 4)	R/W
b11~b10	Reserved	-	Read as "0", write as "0"	R/W
b9~b8	FORC [1:0]	Forced port status setting	0x: Set invalid 10: Starting from next period, TIMA_ < t > _ PWMn port output is set to low 11: Starting next period, the output of TIMA_ < t > _ PWMn port is set to high (n = 1 ~ 4) Note 1: The next period 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 period.	R/W
b7~b6	PERC [1:0]	Port status setting when period values match	00: TIMA_ < t > _ PWMn port output is set to low when count is equal to PERAR 01: TIMA_ < t > _ PWMn port output is set to high when count is equal to PERAR 10: TIMA_ < t > _ PWMn port output remains in the previous state when the count value is equal to PERAR 11: TIMA_ < t > _ PWMn port output is set to inverted level when count is equal to PERAR (n = 1 ~ 4) Note: The sawtooth wave mode counts up and the period point is PERAR , the sawtooth wave mode counts down and the triangle wave mode period point is zero.	R/W
b5~b4	CMPC [1:0]	Port status setting when comparison values match	00: TIMA_ < t > _ PWMn port output is set to low when the count value is equal to CMPARn 01: When the count value is equal to CMPARn, the TIMA_ < t > _ PWMn port output is set to high 10: When the count value is equal to CMPARn, the TIMA_ < t > _ PWMn port output remains in the previous state 11: TIMA_ < t > _ PWMn port output is set to inverted level when the count value is equal to CMPARn (n = 1 ~ 4)	R/W
b3~b2	STPC [1:0]	Port status setting when count stops	00: TIMA_ < t > _ PWMn port output set to low level when count stops 01: When the count stops, the TIMA_ < t > _ PWMn port output is set to high 10: When the count stops, the TIMA_ < t > _ PWMn port output remains in the previous state 11: When the count stops, the TIMA_ < t > _ PWMn port output remains in the previous state (n = 1 ~ 4)	R/W
b1~b0	STAC [1:0]	Port status setting at start of count	00: At the start of the count, the TIMA_ < t > _ PWMn port output is set to low 01: At the beginning of the count, the TIMA_ < t > _ PWMn port output is set to high 10: At the beginning of the count, the TIMA_ < t > _ PWMn port output remains in the previous state 11: At the beginning of the count, the TIMA_ < t > _ PWMn port output remains in the previous state (n = 1 ~ 4) Note: This bit setting is only valid when there is no frequency division (BCSTR.LCKDIV=4'h0). For other frequency divisions,	R/W

please set it to 2'b10 or 2'b11

23.5.13 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	-	HCLE2	HCLE1	HCLE0	-	HSTP2	HSTP1	HSTP0	-	HSTA2	HSTA1	HSTA0

Bit	Flag	Bit Name	Function	Read and write
b15	HCLE6	Hardware zeroing condition 6	Condition: TIMA_<t>_PWM3 port input sampling to falling edge 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	R/W
b14	HCLE5	Hardware zeroing condition 5	Condition: TIMA_<t>_PWM3 port input sampled to rising edge 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	R/W
b13	HCLE4	Hardware zeroing condition 4	Condition: When this unit is unit m, the TRIG port input of unit n is sampled to the falling edge (when m=1, 3, 5, 7, 9, 11, n=2, 4, 6, 8, 10, 12; when m=2, 4, 6, 8, 10, 12, n=1, 3, 5, 7, 9, 11) 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	R/W
b12	HCLE3	Hardware zeroing condition 3	Condition: When this unit is unit m, the TRIG port input of unit n is sampled to the rising edge (when m=1, 3, 5, 7, 9, 11, n=2, 4, 6, 8, 10, 12; when m=2, 4, 6, 8, 10, 12, n=1, 3, 5, 7, 9, 11) 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	R/W
b11	Reserved	-	Read as "0", write as "0"	R/W
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.	R/W
b9	HCLE1	Hardware zeroing condition 1	Condition: TIMA_ TRIG port input to drop edge 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	R/W
b8	HCLE0	Hardware zeroing condition 0	Condition: TIMA_ TRIG port input sampled to rising edge 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	R/W
b7	Reserved	-	Read as "0", write as "0"	R/W
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.	R/W
b5	HSTP1	Hardware stop condition 1	Condition: TIMA_ < t > _TRIG port input to drop edge 0: Hardware stop invalid when condition matches 1: Hardware stops working when conditions match	R/W
b4	HSTP0	Hardware stop condition 0	Condition: TIMA_ < t > _TRIG port input sampling to rising edge 0: Hardware stop invalid when condition matches 1: Hardware stops working when conditions match	R/W
b3	Reserved	-	Read as "0", write as "0"	R/W
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.	R/W
b1	HSTA1	Hardware startup condition 1	Conditions: 1) The input of the TIMA_<t>_TRIG port of this unit is sampled to the falling edge (the synchronous start function is invalid) 2) TIMA_ TRIG port input is sampled to the falling edge (synchronous start function is valid) 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match Note: In condition 2), when the current unit is 2, 4, 6, 8, 10, 12, n=1, 3, 5, 7, 9, 11; when the current unit is 1, 3, 5, 7, 9, 11, this function is invalid	R/W
b0	HSTA0	Hardware startup condition 0	Conditions: 1) The input of the TIMA_<t>_TRIG port of this unit is sampled	R/W

to the rising edge (the synchronous start function is invalid)
2) TIMA_n_TRIG port input is sampled to the rising edge
(synchronous start function is valid)
0: Hardware startup is invalid when conditions match
1: Hardware startup is valid when conditions match
Note: In condition 2), when the current unit is 2, 4, 6, 8, 10, 12,
n=1, 3, 5, 7, 9, 11; when the current unit is 1, 3, 5, 7, 9, 11, this
function is invalid

23.5.14 Hardware increment Event Selection Register (TMRA_HCUPR)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	HC UP12	HC UP11	HC UP10	HC UP9	HC UP8	HC UP7	HC UP6	HC UP5	HC UP4	HC UP3	HC UP2	HC UP1	HC UP0

Bit	Flag	Bit Name	Function	Read and write
b15~b13	Reserved	-	Read as "0", write as "0"	R/W
b12	HCUP12	Hardware increment condition 12	Condition: When the current unit is unit m, unit n has a count underflow (when m=1, 3, 5, 7, 9, 11, n=2, 4, 6, 8, 10, 12; when m=2, 4, 6, 8, 10, 12, n=1, 3, 5, 7, 9, 11) 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	R/W
b11	HCUP11	Hardware increment condition 11	Condition: When this unit is unit m, unit n has count overflow (when m=1, 3, 5, 7, 9, 11, n=2, 4, 6, 8, 10, 12; when m=2, 4, 6, 8, 10, 12, n=1, 3, 5, 7, 9, 11) 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	R/W
b10	HCUP10	Hardware increment 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.	R/W
b9	HCUP9	Hardware increment condition 9	Condition: TIMA_ < t > _TRIG port samples up to falling edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	R/W
b8	HCUP8	Hardware increment condition 8	Condition: TIMA_ < t > _TRIG port samples up to rising edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	R/W
b7	HCUP7	Hardware increment condition 7	Condition: When the TIMA_ < t > _CLKA port is at high level, the TIMA_ < 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	R/W
b6	HCUP6	Hardware increment condition 6	Condition: When the TIMA_ < t > _CLKA port is at a high level, the TIMA_ < t > _CLKA port is sampled up to the rising edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	R/W
b5	HCUP5	Hardware increment condition 5	Condition: When the TIMA_ < t > _CLKA port is at low level, the TIMA_ < 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	R/W
b4	HCUP4	Hardware increment condition 4	Condition: When the TIMA_ < t > _CLKA port is at low level, the TIMA_ < t > _CLKA port is sampled up to the rising edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	R/W
b3	HCUP3	Hardware increment condition 3	Condition: When the TIMA_ < t > _CLKA port is at high level, the TIMA_ < t > _CLKB port is sampled up to the falling edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	R/W
b2	HCUP2	Hardware increment condition 2	Condition: When the TIMA_ < t > _CLKA port is at a high level, the TIMA_ < t > _CLKB port is sampled up to the rising edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	R/W
b1	HCUP1	Hardware increment condition 1	Condition: When the TIMA_ < t > _CLKA port is at low level, the TIMA_ < t > _CLKB port is sampled up to the falling edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	R/W
b0	HCUP0	Hardware increment condition 0	Condition: When the TIMA_ < t > _CLKA port is at low level, the TIMA_ < t > _CLKB port is sampled up to the rising edge 0: Hardware add invalid when condition matches 1: Hardware addition is valid when a condition matches	R/W

23.5.15 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
-	-	-	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	Flag	Bit Name	Function	Read and write
b15~b13	Reserved	-	Read as "0", write as "0"	R/W
b12	HCD012	Hardware decrement condition 12	Condition: When the current unit is unit m, unit n has a count underflow (when m=1, 3, 5, 7, 9, 11, n=2, 4, 6, 8, 10, 12; when m=2, 4, 6, 8, 10, 12, n=1, 3, 5, 7, 9, 11) 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	R/W
b11	HCD011	Hardware decrement condition 11	Condition: When this unit is unit m, unit n has count overflow (when m=1, 3, 5, 7, 9, 11, n=2, 4, 6, 8, 10, 12; when m=2, 4, 6, 8, 10, 12, n=1, 3, 5, 7, 9, 11) 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	R/W
b10	HCD010	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.	R/W
b9	HCD09	Hardware decrement condition 9	Condition: TIMA_ < t > _TRIG port samples up to falling edge 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	R/W
b8	HCD08	Hardware decrement condition 8	Condition: TIMA_ < t > _TRIG port samples up to rising edge 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	R/W
b7	HCD07	Hardware decrement condition 7	Condition: When the TIMA_ < t > _CLKA port is at high level, the TIMA_ < 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	R/W
b6	HCD06	Hardware decrement condition 6	Condition: When the TIMA_ < t > _CLKA port is at a high level, the TIMA_ < t > _CLKA port is sampled up to the rising edge 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	R/W
b5	HCD05	Hardware decrement condition 5	Condition: When the TIMA_ < t > _CLKA port is at low level, the TIMA_ < 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	R/W
b4	HCD04	Hardware decrement condition 4	Condition: When the TIMA_ < t > _CLKA port is at low level, the TIMA_ < t > _CLKA port is sampled up to the rising edge 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	R/W
b3	HCD03	Hardware decrement condition 3	Condition: When the TIMA_ < t > _CLKA port is at high level, the TIMA_ < t > _CLKB port is sampled up to the falling edge 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	R/W
b2	HCD02	Hardware decrement condition 2	Condition: When the TIMA_ < t > _CLKA port is at a high level, the TIMA_ < t > _CLKB port is sampled up to the rising edge 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	R/W
b1	HCD01	Hardware decrement condition 1	Condition: When the TIMA_ < t > _CLKA port is at low level, the TIMA_ < t > _CLKB port is sampled up to the falling edge 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	R/W
b0	HCD00	Hardware decrement condition 0	Condition: When the TIMA_ < t > _CLKA port is at low level, the TIMA_ < t > _CLKB port is sampled up to the rising edge 0: Hardware decrement is invalid when conditions match 1: Hardware decrements are effective when conditions match	R/W

24 General Timer (Timer2)

24.1 Introduction

General timer 2 (Timer2) is a basic timer that can implement synchronous counting and asynchronous counting. This timer contains 2 channels (CH-A and CH-B). Each channel has an output port for basic square wave output; each channel has two input ports, one is the clock input port for asynchronous port counting; the other is the trigger input port for timer start, stop, reset, counting action and count value capture input; supports pulse width measurement and period measurement. 4 unit of Timer 2 is carried in this product family.

24.2 Basic Block Diagram

The basic functions and features of Timer2 are as follows:

Table 24-1 Basic functions and features of Timer2

Basic action	Hardware triggered counter start, stop, reset, count and capture inputs
	Synchronous count
	Pulse width measurement
	Period measurement
	Square wave output
	Event signal output
Interrupt type	Count comparison match interrupt
	Count overflow interrupt

The basic block diagram of Timer2 is shown in Figure Figure 24-1. 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. The ANF on TIM2_<t>_CLKA/B port input in the block diagram is an analog filter unit with a filtering width of 40ns (typical condition).

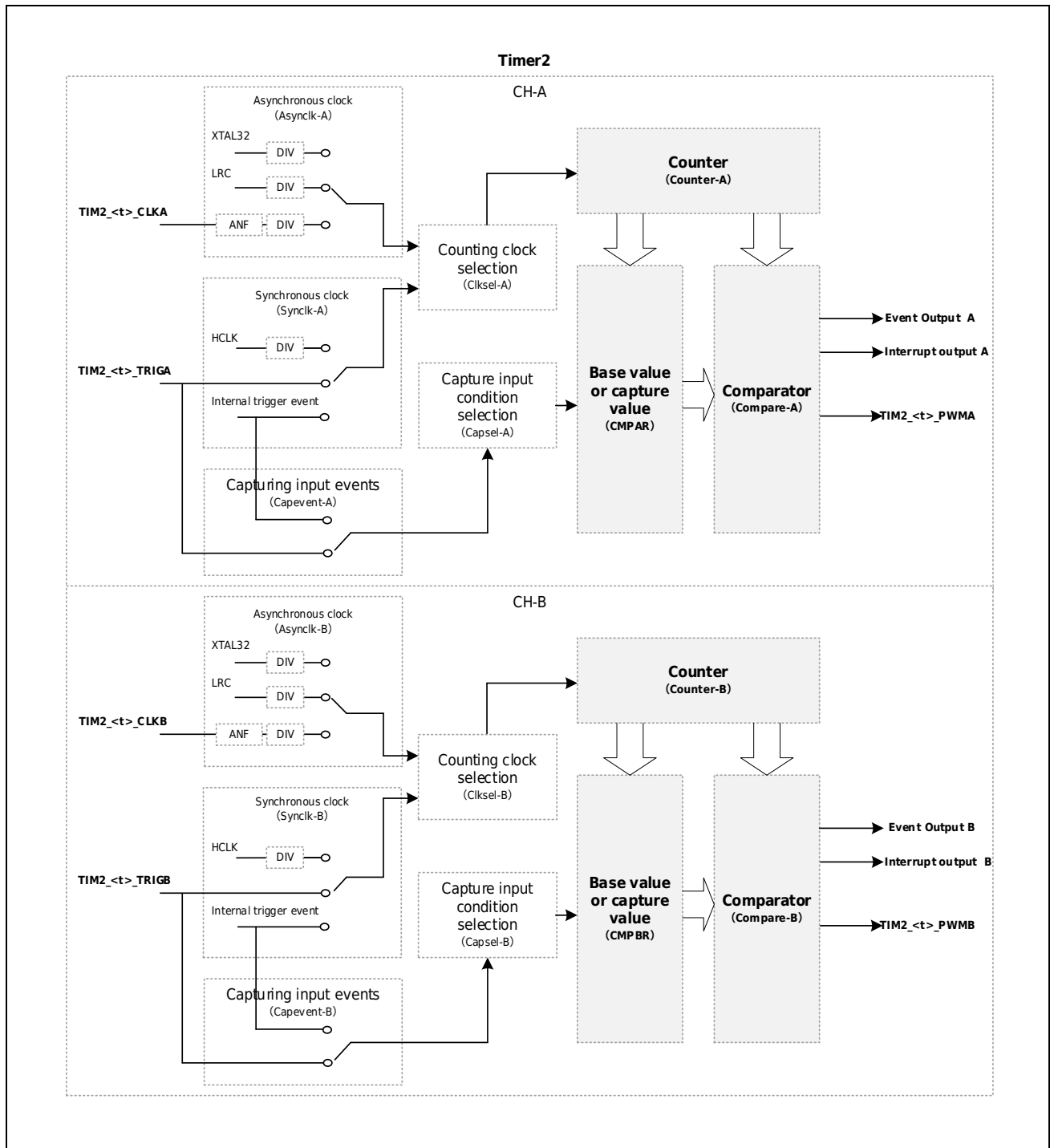


Figure 24-1 Timer2 Basic Block Diagram

Table 24-2 shows the list of input and output ports of Timer2.

Table 24-2 Timer2 Port List

Port Name	Direction	Function
TIM2_<t>_CLKA/B	I	Asynchronous clock input port
TIM2_<t>_TRIGA/B	I	1) Hardware startup, stopping, zeroing, counting input port 2) Capture input port
TIM2_<t>_PWMA/B	O	Square wave output port

24.3 Functional Description

24.3.1 Timer Selection

Timer2 can be counted either synchronously or asynchronously.

The synchronous counting mode refers to the synchronous timing relationship between the timer count clock and the bus access clock (register read/write operation clock). Asynchronous counting means that the timer count clock and bus access clock (register read/write operation clock) are non-synchronous timing relationships. The status of a timer, etc., may be changing when the register is read by asynchronous counting. Therefore, in the asynchronous counting mode, the register read operation must be implemented in the count stopped state.

24.3.1.1 Synchronous Count Clock Source

In synchronous counting mode (BCONR.SYNSA=0), the clock source can be selected from the following options (BCONR.SYNCLK A [1:0] or BCONR.SYNCLK A T [1:0] setting):

- PCLK1 and PCLK1 divided by 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 as the synchronous counting clock (BCONR.SYNCLK A [1:0]=00 & BCONR.CKDIV A [3:0] set up)
- The input valid selection edge of TIM2_<t>_TRIGA and TIM2_<t>_TRIGB ports is sampled by the internal PCLK1 and used as the synchronous count clock (BCONR.SYNCLK A [1:0] = 01 or 10)
- Internal Hardware Trigger Event Input as Sync Count Clock (BCONR.SYNCLK A [1:0]=11)
- Timer6 count overflow event (BCONR.SYNCLKAT [0]=1)
- Timer6 count underflow event (BCONR.SYNCLKAT [1]=1)

The counting clock source is a software counting mode, and the counting clock source is b, c, d, e hardware counting mode. 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. When selecting clock sources d and e, please set PCLK0 and PCLK1 to the same frequency.

24.3.1.2 Asynchronous Counting Clock Source

In the asynchronous counting mode ($\text{BCONR.SYNSA} \langle B \rangle = 1$), the clock source can have the following options ($\text{BCONR.ASYNCLK A} \langle B \rangle$ setting):

- a) LRctimer input and its 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 frequency division as asynchronous counting clock ($\text{BCONR.ASYNCLK A} \langle B \rangle = 00$ & $\text{BCONR.CKDIV A} \langle B \rangle$ [3:0] setting)
- b) XTAL32 clock source input and its 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 division as asynchronous counting clock ($\text{BCONR.ASYNCLK A} \langle B \rangle = 01$ & $\text{BCONR.CKDIV A} \langle B \rangle$ [3:0] setting)
- c) $\text{TIM2}_{\langle t \rangle\_CLKA}$, $\text{TIM2}_{\langle t \rangle\_CLKB}$ clock input ports and their 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 divisions as asynchronous count clocks ($\text{BCONR.ASYNCLK A} \langle B \rangle = 10$ & $\text{BCONR.CKDIV A} \langle B \rangle$ [3:0] settings)

24.3.2 Comparison Output

Timer2 has two output ports ($\text{TIM2}_{\langle t \rangle\_PWMA}$, $\text{TIM2}_{\langle t \rangle\_PWMB}$) that can achieve basic square wave output.

The CMPAR and CMPBR registers correspond to the count comparison reference values of $\text{TIM2}_{\langle t \rangle\_PWMA}$ and $\text{TIM2}_{\langle t \rangle\_PWMB}$. When the timer count values CNTAR and CMPAR are equal, the $\text{TIM2}_{\langle t \rangle\_PWMA}$ port outputs the specified level; when the timer count values CNTBR and CMPBR are equal, the $\text{TIM2}_{\langle t \rangle\_PWMB}$ port outputs the specified level.

The count start level, stop level, count compare match level, etc. of the $\text{TIM2}_{\langle t \rangle\_PWMA/B}$ port can be set by the $\text{PCONR.STACA} \langle B \rangle$ [1:0], $\text{PCONR.STPCA} \langle B \rangle$ [1:0], and $\text{PCONR.CMPCA} \langle B \rangle$ [1:0] bits of the port control register (PCONR). Figure 24-2 This is an instance of comparison output operation.

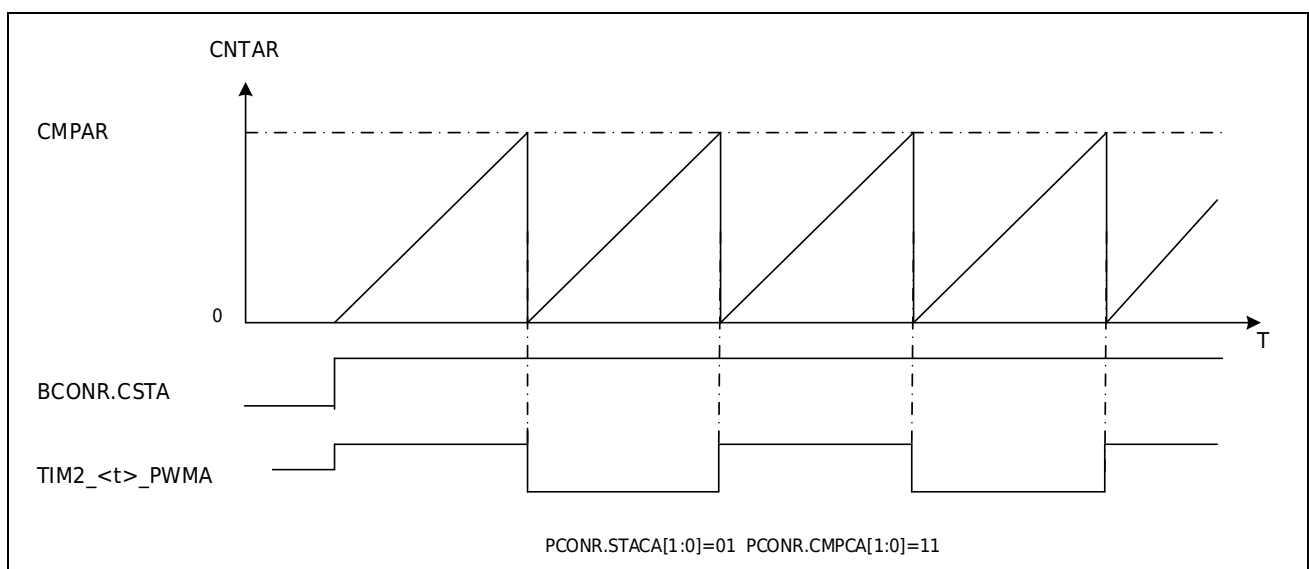


Figure 24-2 Compare output action

24.3.3.2 Capture Input Action

The capture input function is implemented when the TIM2_<t>_TRIGA/B port input or internal trigger event is used as the capture input condition. When the capture input condition selected by the hardware control register (HCONR) is valid, the current count value is saved in the corresponding register (CMPAR , CMPBR). Figure 24-4 An instance of an action for capturing input.

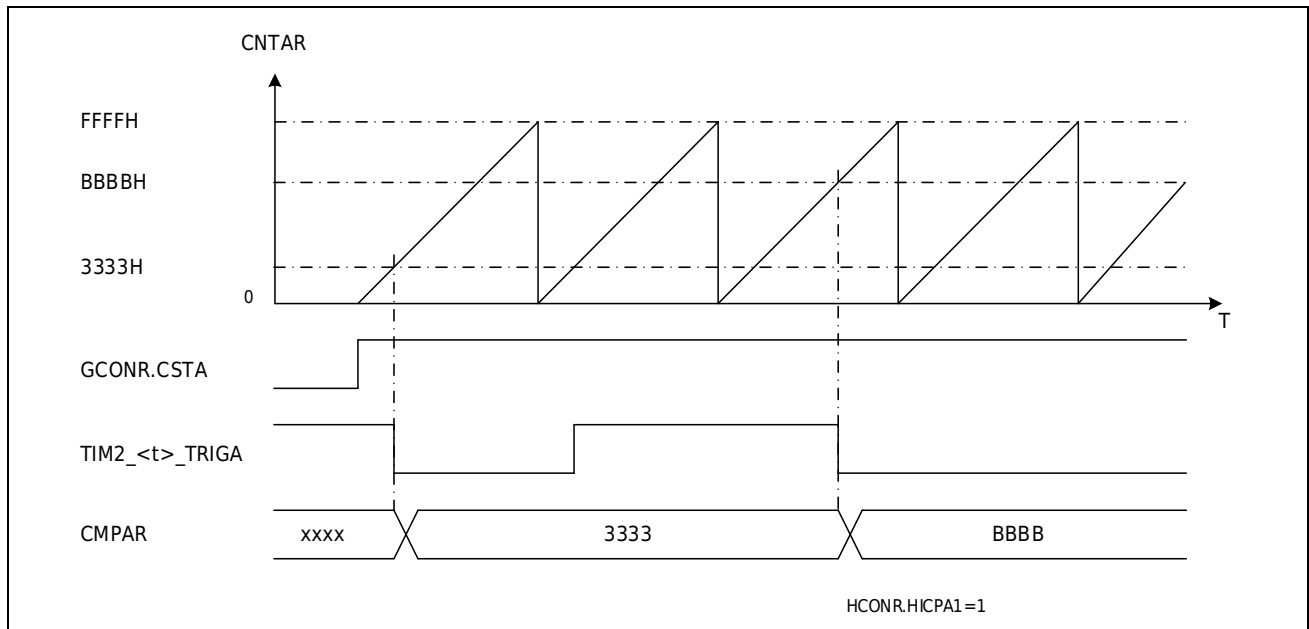


Figure 24-4 Capturing Input Actions

24.3.3.3 Pulse Width Measurement

The continuous pulse width measurement can be realized by setting the hardware startup condition of A channel as the rising edge of TIM2_<t>_TRIGA, and setting the hardware reset condition, stop condition and capture input condition as the falling edge of TIM2_<t>_TRIGA. B can achieve the same function. The corresponding actions are shown in FigureFigure 24-5.

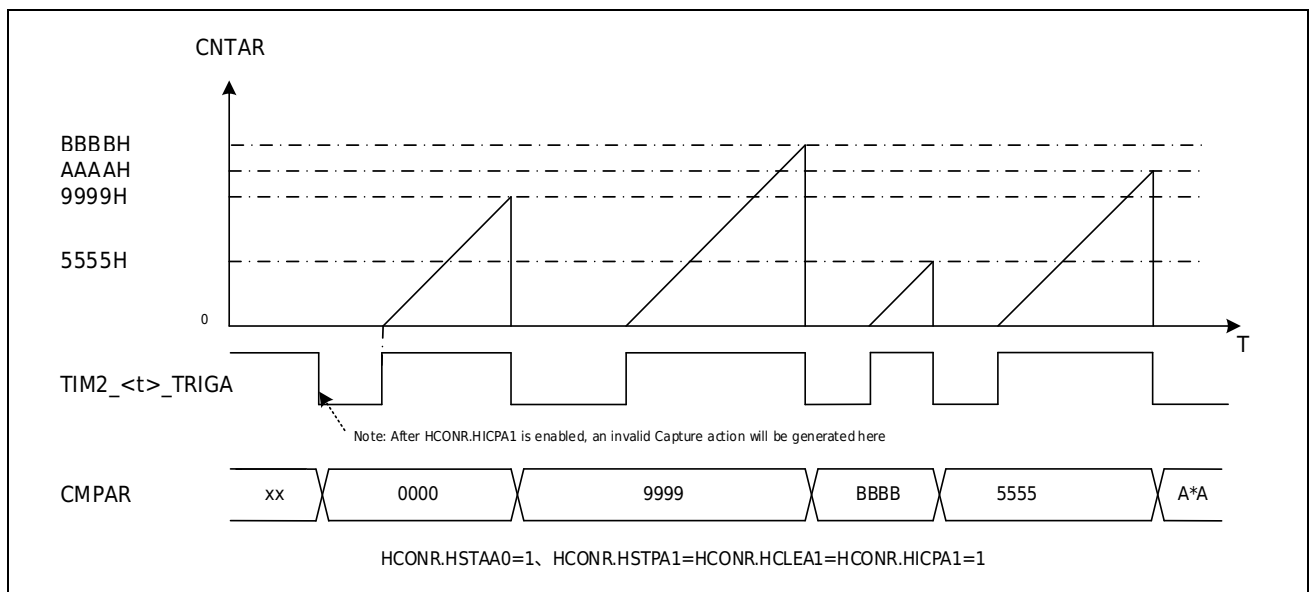


Figure 24-5 Pulse Width Measurement

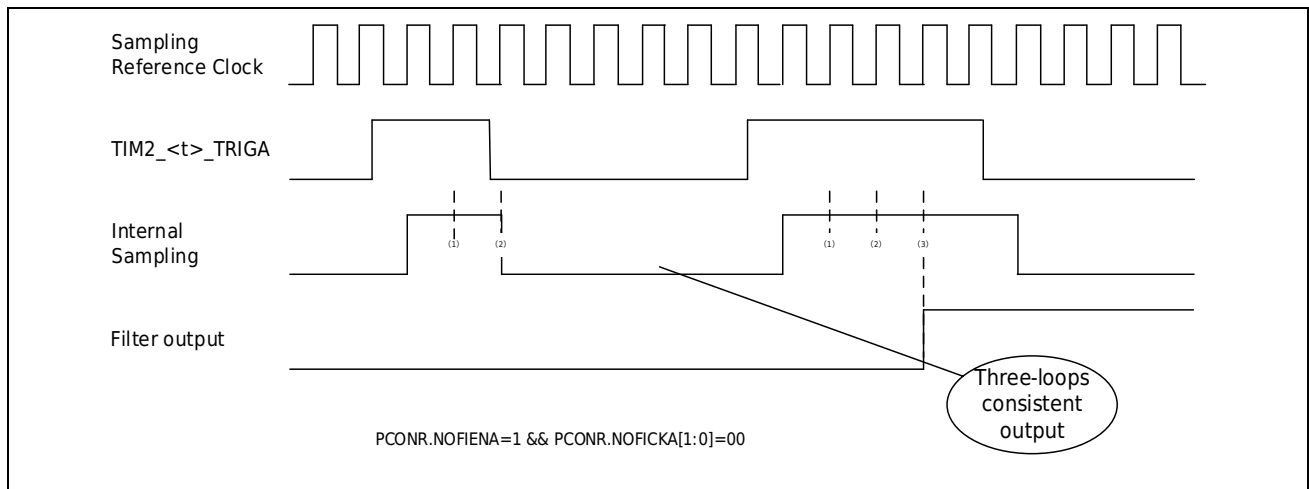


Figure 24-7 Digital Filtering At The -TRIG Input Port

24.4 Interrupt and Event Description

24.4.1 Interrupt Output

One Timer2 has 4 interrupts, namely, the count compare match interrupt or capture input interrupt of channel A and channel B, and the count overflow interrupt of channel A and channel B.

There are 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 bit in the status flag register (STFLR) will be set to 1. the ICONR.CMENA bit of the interrupt control register (ICONR) is set to enable interrupt, the corresponding interrupt request (TMR2_<t>_CMPm, m=A, B) will also be triggered.

When the capture input valid condition selected by the hardware control register (HCONR) occurs, the corresponding capture input action can be generated. the ICONR.CMENA bit of the interrupt control register (ICONR) is set to enable interrupt, the corresponding interrupt request (TMR2_<t>_CMPm, m=A, B) is triggered.

When the count value register (CNTAR, CNTBR) counts to 0XFFFF, a count overflow event occurs. At this time, the STFLR.OVFA bit in the status flag register (STFLR) will be set to 1. the ICONR.OVENA bit in the interrupt control register (ICONR) is set to enable interrupt, the corresponding interrupt request (TMR2_<t>_OVFm, m=A, B) will also be triggered.

When the asynchronous counting mode is selected (except BCONR.ASYNCLKA[1:0]=2'b10), the compare match interrupt generated by the reference value register (CMPAR) of unit 1 can be used to wake up the system in low power mode. For details, please refer to the [Interrupt Controller (INTC)] section.

24.4.2 Event Output

One Timer2 has four events, namely, the count compare match events or capture input events of channel A and channel B, and the count overflow events of channel A and channel B.

When a count compare match, capture input action or count overflow occurs during the counting process, a corresponding event request (TMR2_<t>_CMPm or TMR2_<t>_OVFm , m=A , B) output signal will be generated, which can be used to select and trigger other modules.

24.5 Register Description

Table 24-3 shows the register list of the Timer2 module.

Table 24-3 Timer2 Register List

BASE ADDR:

0x40024800(U1), 0x40024C00(U2), 0x40025000(U3), 0x40025400(U4)

Register Name	Symbol	Offset	Bit Width	Reset Value
Register count value	TMR2_CNTAR	0x0000	32	0x00000000
Register count value	TMR2_CNTBR	0x0004	32	0x00000000
Reference value register	TMR2_CMPAR	0x0008	32	0x0000FFFF
Reference value register	TMR2_CMPBR	0x000C	32	0x0000FFFF
Basic control register	TMR2_BCONR	0x0010	32	0x00000000
Interrupt control register	TMR2_ICONR	0x0014	32	0x00000000
Port control register	TMR2_PCONR	0x0018	32	0x00000000
Hardware control register	TMR2_HCONR	0x001c	32	0x00000000
Status flag register	TMR2_STFLR	0x0020	32	0x00000000

24.5.1 Count Value Register (TMR2_CNTmR) (m=A , B)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CNTA [15:0]															
Bit	Flag		Bit Name		Function										Read and write
b31~b16	Reserved		-		Read as "0"										R
b15~b0	CNTA [15:0]		Count value		Current timer count										R/W

24.5.2 Reference Value Register (TMR2_CMPmR) (m=A , B)

Reset value: 0x0000FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CMPA [15:0]															
Bit	Flag		Bit Name		Function										Read and write
b31~b16	Reserved		-		Read as "0"										R
b15~b0	CMPA [15:0]		Reference value		Setting the count reference to produce the Compare Match event										R/W

24.5.3 Basic Control Register (TMR2 _ BCONR)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	SYN CLKBT [1:0]	ASYN CLKB [1:0]	SYN CLKB [1:0]					CKDIV B [3:0]			SYNS B	-	CAP MDB	CST B
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	SYN CLKAT [1:0]	ASYN CLKA [1:0]	SYN CLKA [1:0]					CKDIV A [3:0]			SYNS A	-	CAP MDA	CST A

Bit	Flag	Bit Name	Function	Read and write
b31~b30	Reserved	-	Read as "0", write as "0"	R/W
b29	SYNCLKBT [1]	Synchronous counting and clock source selection BT	Condition: Count underflow occurs in unit n of Timer6 0: When the condition is matched, the synchronous counting of Timer2 unit m is invalid 1: When the condition is matched, Timer2 unit m performs a synchronous count (When m=1, 2, 3, 4, n=2, 4, 6, 8)	R/W
b28	SYNCLKBT [0]	Synchronous counting and clock source selection BT	Condition: Count overflow occurs in unit n of Timer6 0: When the condition is matched, the synchronous counting of Timer2 unit m is invalid 1: When the condition is matched, Timer2 unit m performs a synchronous count (When m=1, 2, 3, 4, n=2, 4, 6, 8)	R/W
b27~b26	ASYNCLKB [1:0]	Asynchronous counting and clock source selection B	00: LRC 01: XTAL32 10: TIM2_<t>_CLKB clock input 11: Setting prohibited	R/W
b25~b24	SYNCLKB [1:0]	Synchronous counting and clock source selection B	00: PCLK1 01: TIM2_<t>_TRIGB rising edge (internal PCLK1 synchronization) 10: TIM2_<t>_TRIGB falling edge (internal PCLK1 synchronization) 11: Internal Trigger Event Input	R/W
b23~b20	CKDIVB [3:0]	Count clock frequency division selection B	Counting clock frequency division options: 0000: clock source 0001: clock source/2 0010: clock source/4 0011: clock source/8 0100: clock source/16 0101: clock source/32 0110: clock source/64 0111: clock source/128 1000: clock source/256 1001: clock source/512 1010: clock source/1024 Please do not set other values Note: The timer to be divided can be a variety of timer at asynchronous counting, PCLK1 at synchronous counting.	R/W
b19	SYNSB	Counting mode selection B	0: Synchronous counting mode 1: Asynchronous counting	R/W
b18	Reserved	-	Read as "0", write as "0"	R/W
b17	CAPMDB	Function mode selection B	0: Comparison output function 1: Capture input function	R/W
b16	CSTB	Timer Start B	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	R/W
b15~b14	Reserved	-	Read as "0", write as "0"	R/W
b13	SYNCLKAT [1]	Synchronous counting and clock source selection AT	Condition: Count underflow occurs in unit n of Timer6 0: When the condition is matched, the synchronous counting of Timer2 unit m is invalid 1: When the condition is matched, Timer2 unit m performs a synchronous count (When m=1, 2, 3, 4, n=1, 3, 5, 7)	R/W
b12	SYNCLKAT [0]	Synchronous counting and clock source selection AT	Condition: Count overflow occurs in unit n of Timer6 0: When the condition is matched, the synchronous counting of Timer2 unit m is invalid 1: When the condition is matched, Timer2 unit m performs a synchronous count	R/W

(When m=1, 2, 3, 4, n=1, 3, 5, 7)			
b11~b10	ASYNCLKA [1:0]	Asynchronous count with timer selection A	00: LRC 01: XTAL32 10: TIM2_ < t > _CLKA clock input 11: Disable setting R/W
b9~b8	SYNCLKA [1:0]	Synchronous count with timer selection A	00: PCLK1 01: TIM2_ < t > _TRIGA rising edge (internal HCLK synchronization) 10: TIM2_ < t > _TRIGA falling edge (internal HCLK synchronization) 11: Internal Trigger Event Input R/W
b7~b4	CKDIVA [3:0]	Counting clock frequency division selection A	Counting clock frequency division options: 0000: clock source 0001: clock source/2 0010: clock source/4 0011: clock source/8 0100: clock source/16 0101: clock source/32 0110: clock source/64 0111: clock source/128 1000: clock source/256 1001: clock source/512 1010: clock source/1024 Please do not set other values Note: The timer to be divided can be a variety of timer at asynchronous counting, PCLK1 at synchronous counting. R/W
b3	SYNSA	Counting mode selection A	0: Synchronous counting mode 1: Asynchronous counting R/W
b2	Reserved	-	Read as "0", write as "0" R/W
b1	CAPMDA	Functional mode selection A	0: Comparison output function 1: Capture input function R/W
b0	CSTA	Timer Startup A	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 R/W

24.5.4 Interrupt Control Register (TMR2_ICONR)

Reset value: 0x00000000

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

Bit	Flag	Bit Name	Function	Read and write
b31~b18	Reserved	-	Read as "0", write as "0"	R/W
b17	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	R/W
b16	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	R/W
b15~b2	Reserved	-	Read as "0", write as "0"	R/W
b1	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	R/W
b0	CMENA	Count Match interrupt enable A	0: CMPAR register 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	R/W

24.5.5 Port Control Register (TMR2_PCONR)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	NOFI CKB [1:0]	NOFI ENB	-	-	-	OUT ENB	-	-	-	CMP CB [1:0]	STP CB [1:0]	STA CB [1:0]			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	NOFI CKA [1:0]	NOFI ENA	-	-	-	OUT ENA	-	-	-	CMP CA [1:0]	STP CA [1:0]	STA CA [1:0]			

Bit	Flag	Bit Name	Function	Read and write
b31	Reserved	-	Read as "0", write as "0"	R/W
b30~b29	NOFICKB	Filtering sampling reference clock selection B	00: clock source 01: clock source/4 10: clock source/16 11: clock source/64 Note: The clock source is the clock source at asynchronous count and the HCLK at synchronous count.	R/W
b28	NOFIENB	Trigger port filtering B	0: TIM2_<t>_TRIGB input port filtering function invalid 1: TIM2_<t>_TRIGB input port filter function enable	R/W
b27~b25	Reserved	-	Read as "0", write as "0"	R/W
b24	OUTENB	Port output enable B	0: TIM2_<t>_PWMB port output is invalid 1 : TIM2_<t>_PWMB port output is valid	R/W
b23~b22	Reserved	-	Read as "0", write as "0"	R/W
b21~b20	CMPCB [1:0]	Port status setting when comparison values matchB	00: When CNTBR is equal to CMPBR, TIM2_<t>_PWMB port output is set to low 01: When CNTBR is equal to CMPBR, TIM2_<t>_PWMB port output is set to high 10: When the count value (CNTBR) is equal to the CMPBR, the TIM2_<t>_PWMB port output remains in the previous state 11: When the count value (CNTBR) is equal to the CMPBR, the output of the TIM2_<t>_PWMB port is set to the inverted level	R/W
b19~b18	STPCB [1:0]	When counting stops, the port status is set to B	00: When count stops, TIM2_<t>_PWMB port output is set to low 01: When count stops, TIM2_<t>_PWMB port output is set to high 10: When the count stops, the TIM2_<t>_PWMB port output remains in the previous state 11: When the count stops, the TIM2_<t>_PWMB port output remains in the previous state	R/W
b17~b16	STACB [1:0]	When counting starts, the port status is set to B	00: When counting starts, the TIM2_<t>_PWMB port output is set to a low level 01: When counting starts, the TIM2_<t>_PWMB port output is set to a high level 10: When the count starts, the TIM2_<t>_PWMB port output remains in the previous state 11: When the count starts, the TIM2_<t>_PWMB port output remains in the previous state Note: This bit setting is only valid when there is no frequency division (BCONR.CKDIVB = 4'h0). For other frequency divisions, please set it to 2'b10 or 2'b11	R/W
b15	Reserved	-	Read as "0", write as "0"	R/W
b14~b13	NOFICKA	Filtering sampling reference clock selection A	00: clock source 01: clock source/4 10: clock source/16 11: clock source/64 Note: The clock source is the clock source at asynchronous count and the HCLK at synchronous count.	R/W
b12	NOFIENA	Trigger Port Filter A	0: Invalid TIM2_<t>_TRIGA input port filtering function 1: TIM2_<t>_TRIGA input port filter enabled	R/W
b11~b9	Reserved	-	Read as "0", write as "0"	R/W
b8	OUTENA	Port Output Enable A	0: TIM2_<t>_PWMA port output invalid 1: TIM2_<t>_PWMA port output valid	R/W
b7~b6	Reserved	-	Read as "0", write as "0"	R/W
b5~b4	CMPCA [1:0]	Port State Setting A When Comparing	00: When CNTAR is equal to CMPAR, TIM2_<t>_PWMA port output is set to low 01: When CNTAR is equal to CMPAR, TIM2_<t>_PWMA port	R/W

		Values Match	output is set to high 10: When the count value (CNTAR) is equal to the CMPAR, the TIM2 _ < t > _ PWMA port output remains in the previous state 11: When the count value (CNTAR) is equal to the CMPAR, the output of the TIM2 _ < t > _ PWMA port is set to the inverted level	
b3~b2	STPCA [1:0]	Port Status Setting A at Count Stop	00: When count stops, TIM2 _ < t > _ PWMA port output is set to low 01: When count stops, TIM2 _ < t > _ PWMA port output is set to high 10: When the count stops, the TIM2 _ < t > _ PWMA port output remains in the previous state 11: When the count stops, the TIM2 _ < t > _ PWMA port output remains in the previous state	R/W
b1~b0	STACA [1:0]	Count port status setting A at start-up	00: When count starts, TIM2 _ < t > _ PWMA port output is set to low 01: When count starts, TIM2 _ < t > _ PWMA port output is set to high 10: When the count starts, the TIM2 _ < t > _ PWMA port output remains in the previous state 11: When the count starts, the TIM2 _ < t > _ PWMA port output remains in the previous state Note: This bit setting is only valid when there is no frequency division (BCONR.CKDIVA=4'h0). For other frequency divisions, please set it to 2'b10 or 2'b11	R/W

24.5.6 Hardware Control Register (TMR 2 _ HCONR)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	HICP B2	HICP B1	HICP B0	-	HCLE B2	HCLE B1	HCLE B0	-	HSTP B2	HSTP B1	HSTP B0	-	HSTA B2	HSTA B1	HSTA B0
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	HICP A2	HICP A1	HICP A0	-	HCLE A2	HCLE A1	HCLE A0	-	HSTP A2	HSTP A1	HSTP A0	-	HSTA A2	HSTA A1	HSTA A0

Bit	Flag	Bit Name	Function	Read and write
b31	Reserved	-	Read as "0", write as "0"	R/W
b30	HICPB2	Hardware Capture Input Condition B2	Condition: Internal Hardware Trigger Event Valid 0: Hardware capture input is invalid when conditions match 1: Hardware capture input is valid when conditions match	R/W
b29	HICPB1	Hardware Capture Input Condition B1	Condition: TIM2_<t>_TRIGB port samples to drop edge 0: Hardware capture input is invalid when conditions match 1: Hardware capture input is valid when conditions match	R/W
b28	HICPB0	Hardware Capture Input Condition B0	Condition: TIM2_<t>_TRIGB port samples up to rising edge 0: Hardware capture input is invalid when conditions match 1: Hardware capture input is valid when conditions match	R/W
b27	Reserved	-	Read as "0", write as "0"	R/W
b26	HCLEB2	Hardware zeroing condition B2	Condition: Internal Hardware Trigger Event Valid 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	R/W
b25	HCLEB1	Hardware zeroing condition B1	Condition: TIM2_<t>_TRIGB port samples to drop edge 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	R/W
b24	HCLEB0	Hardware zeroing condition B0	Condition: TIM2_<t>_TRIGB port samples up to rising edge 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	R/W
b23	Reserved	-	Read as "0", write as "0"	R/W
b22	HSTPB2	Hardware stop condition B2	Condition: Internal Hardware Trigger Event Valid 0: Hardware stop invalid when condition matches 1: Hardware stops working when conditions match	R/W
b21	HSTPB1	Hardware stop condition B1	Condition: TIM2_<t>_TRIGB port samples to drop edge 0: Hardware stop invalid when condition matches 1: Hardware stops working when conditions match	R/W
b20	HSTPB0	Hardware stop condition B0	Condition: TIM2_<t>_TRIGB port samples up to rising edge 0: Hardware stop invalid when condition matches 1: Hardware stops working when conditions match	R/W
b19	Reserved	-	Read as "0", write as "0"	R/W
b18	HSTAB2	Hardware startup condition B2	Condition: Internal Hardware Trigger Event Valid 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	R/W
b17	HSTAB1	Hardware startup condition B1	Condition: TIM2_<t>_TRIGB port samples to drop edge 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	R/W
b16	HSTAB0	Hardware startup condition B0	Condition: TIM2_<t>_TRIGB port samples up to rising edge 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	R/W
b15	Reserved	-	Read as "0", write as "0"	R/W
b14	HICPA2	Hardware Capture Input Condition A2	Condition: Internal Hardware Trigger Event Valid 0: Hardware capture input is invalid when conditions match 1: Hardware capture input is valid when conditions match	R/W
b13	HICPA1	Hardware Capture Input Condition A1	Condition: TIM2_<t>_TRIGA port samples to drop edge 0: Hardware capture input is invalid when conditions match 1: Hardware capture input is valid when conditions match	R/W
b12	HICPA0	Hardware Capture Input Condition A0	Condition: TIM2_<t>_TRIGA port samples up to rising edge 0: Hardware capture input is invalid when conditions match 1: Hardware capture input is valid when conditions match	R/W
b11	Reserved	-	Read as "0", write as "0"	R/W
b10	HCLEA2	Hardware zeroing condition A2	Condition: Internal Hardware Trigger Event Valid 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	R/W
b9	HCLEA1	Hardware zeroing condition A1	Condition: TIM2_<t>_TRIGA port samples to drop edge 0: Hardware zeroing is invalid when conditions match	R/W

			1: Hardware zeroing is valid when conditions match	
b8	HCLEA0	Hardware zeroing condition A0	Condition: TIM2_ < t > _TRIGA port samples up to rising edge 0: Hardware zeroing is invalid when conditions match 1: Hardware zeroing is valid when conditions match	R/W
b7	Reserved	-	Read as "0", write as "0"	R/W
b6	HSTPA2	Hardware stop condition A2	Condition: Internal Hardware Trigger Event Valid 0: Hardware stop invalid when condition matches 1: Hardware stops working when conditions match	R/W
b5	HSTPA1	Hardware stop condition A1	Condition: TIM2_ < t > _TRIGA port samples to drop edge 0: Hardware stop invalid when condition matches 1: Hardware stops working when conditions match	R/W
b4	HSTPA0	Hardware stop condition A0	Condition: TIM2_ < t > _TRIGA port samples up to rising edge 0: Hardware stop invalid when condition matches 1: Hardware stops working when conditions match	R/W
b3	Reserved	-	Read as "0", write as "0"	R/W
b2	HSTAA2	Hardware startup condition A2	Condition: Internal Hardware Trigger Event Valid 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	R/W
b1	HSTAA1	Hardware startup condition A1	Condition: TIM2_ < t > _TRIGA port samples to drop edge 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	R/W
b0	HSTAA0	Hardware startup condition A0	Condition: TIM2_ < t > _TRIGA port samples up to rising edge 0: Hardware startup is invalid when conditions match 1: Hardware startup is valid when conditions match	R/W

24.5.7 Status Flag Register (TMR2_STFLR)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved													ICPB	OVFB	CMFB
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved													ICPA	OVFA	CMFA

Bit	Flag	Bit Name	Function	Read and write
b31~b19	Reserved	-	Read as "0", write as "0"	R
b18	ICPB	Repeat Capture Flag B	0: After the count match flag CMFB is 1, no capture input action occurs. 1: After the count match flag CMFB is 1, the capture input action occurs again	R/W
b17	OVFB	Count overflow flag B	0: The count value (CNTBR) has not reached 0xFFFF 1: The count value (CNTBR) counts to 0xFFFF	R/W
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.	R/W
b15~b3	Reserved	-	Read as "0", write as "0"	R
b2	ICPA	Repeat capture flag A	0: After the count match flag CMFA is 1, no capture input action occurs. 1: After the count match flag CMFA is 1, the capture input action occurs again	R/W
b1	OVFA	Count Overflow Flag A	0: The count value (CNTAR) has not reached 0xFFFF 1: The count value (CNTAR) counts to 0xFFFF	R/W
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.	R/W

Note:

- The flag bit in this register can only be written to 0, not 1.

24.6 Precautions

- In the asynchronous counting operation, first set the BCONR.ASYNCLKA bit to select the asynchronous clock source, then set the BCONR.SYNSA bit to select the asynchronous counting mode, and then start Timer2.
- When asynchronous counting is selected, when the count value (CNTAR), reference value (CMPAR), start bit (BCONR.CSTA), and status bit (STFLR.CMFA, STFLR.OVFA, STFLR.ICPA) are modified, Timer2 will write the modified value into the corresponding register after 6 asynchronous counting clocks after receiving the write action.
- When asynchronous counting is selected, the count value (CNTAR), reference value (CMPAR), start bit (BCONR.CSTA), and status bit (STFLR.CMFA, STFLR.OVFA, STFLR.ICPA) must be written continuously with an interval of at least 6 asynchronous counting clocks.
- To select asynchronous counting, set BCONR.SYNCLKA [1:0] to 0.

25 General Timer (Timer0)

25.1 Introduction

General timer 0 (Timer0) is a basic timer that can implement synchronous counting and asynchronous counting. The timer contains 2 channels (CH-A and CH-B), which can generate compare match events during counting. The event can trigger interrupt, or can be output as an event to control other modules. This series of products is equipped with 2 units of Timer0.

25.2 Basic Block Diagram

The basic block diagram of Timer0 is shown in Figure 25-1.

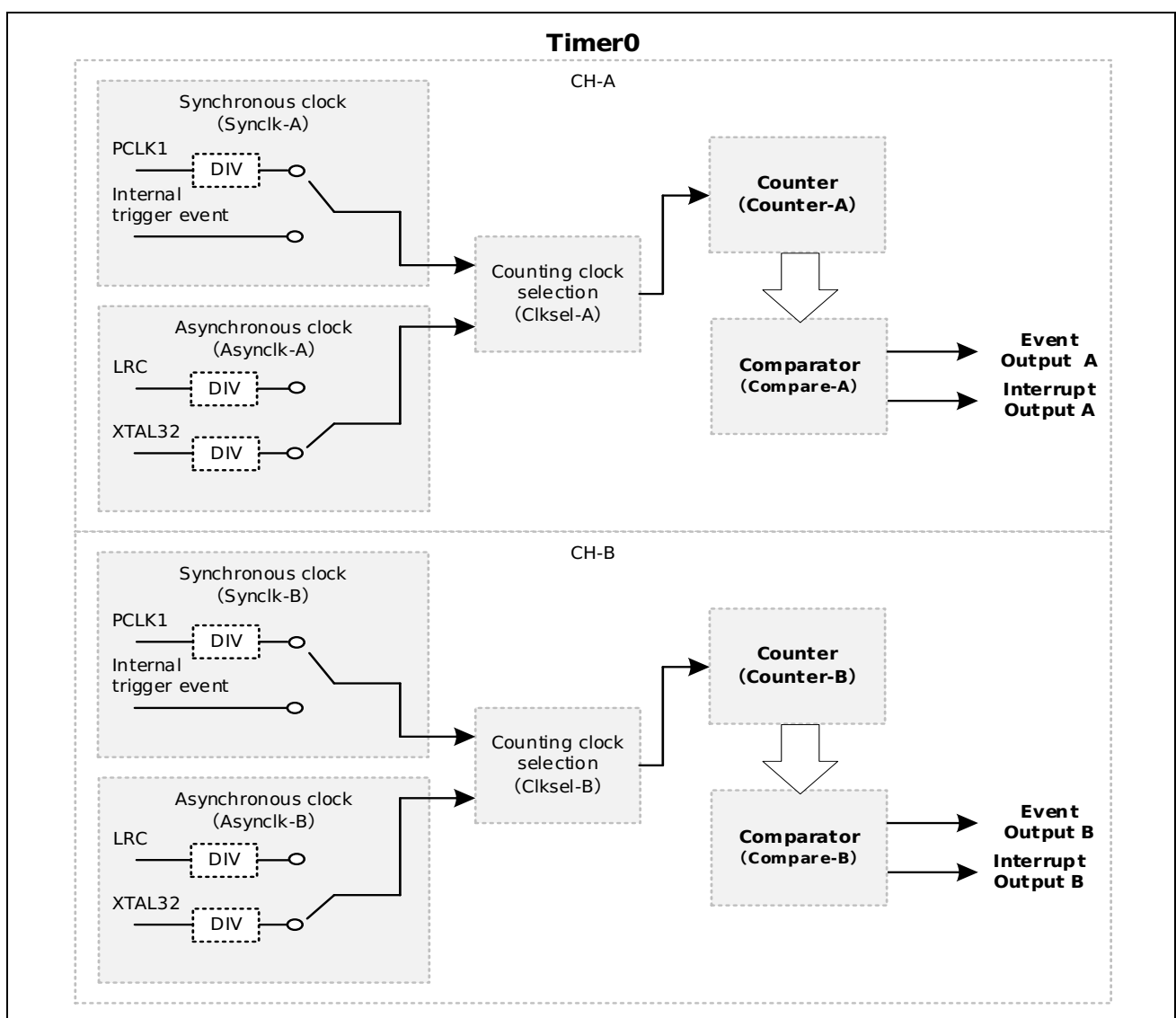


Figure 25-1 Timer0 Basic Block Diagram

25.3 Functional Description

25.3.1 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.

25.3.1.1 Synchronous Count Clock Source

In synchronous counting mode (BCONR.SYNSA=0), the clock source can have the following options (BCONR.SYNCLKA setting):

- PCLK1 and PCLK1 divided by 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 as synchronous count clock (BCONR.SYNCLK A=0 & BCONR.CKDIV A [3:0] setting)
- Internal hardware trigger event input as synchronous count clock (BCONR.SYNCLK A=1)

25.3.1.2 Asynchronous Counting Clock Source

In asynchronous counting mode (BCONR.SYNS A=1), the clock source can be selected from the following options (BCONR.ASYNCLK A setting selection):

- LRC clock source input and its 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 divisions as asynchronous count clocks (BCONR.ASYNCLK A=0 & BCONR.CKDIV A [3:0] settings)
- XTAL32 clock source input and its 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 division as asynchronous counting clock (BCONR.ASYNCLK A=1 & BCONR.CKDIV A [3:0] setting)

25.3.2 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 FigureFigure 25-2.

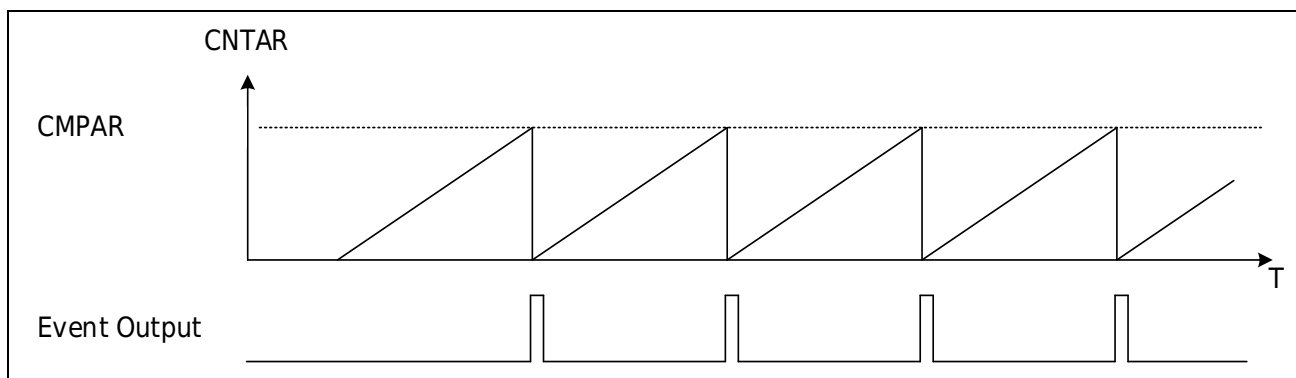


Figure 25-2 Timer0 Counting Timing Diagram

25.3.3 Hardware Triggering

Timer0 contains 2 channels has a common internal hardware trigger source that can control the status of the timer (counting, starting, stopping, clearing, etc.) and capture input actions by basic control register (BCONR) related settings

The source selection of the hardware trigger source is realized by entering the corresponding number into the hardware trigger select register (TMR0_TRGSEL). For the specific event correspondence, refer to the INTC chapter. When using the internal hardware trigger function, you need to enable the peripheral circuit trigger function bit of the function clock control register 0 (PWC_FCG0) first.

25.4 Interrupt and Event Description

25.4.1 Interrupt Output

One Timer0 has two interrupt outputs, namely, count compare match interrupt or capture input 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 bit in the status flag register (STFLR) will be set to 1. At this time, if the BCONR.INTENA bit of the basic control register (BCONR) is set to enable the 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.INTENA bit of the basic control register (BCONR) is set to enable the interrupt, the corresponding interrupt request (TMR0_m_CMPn, m=1, 2; n=A, B) will 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 low power mode. For details, please refer to the [Interrupt Controller (INTC)] section.

25.4.2 Event Output

A Timer0 contains 2 event outputs, which are the count compare match event or capture input event of channel A and channel B respectively.

When a count comparison match or capture input action occurs during the counting process, the corresponding event request (TMR0_m_CMPn, m=1, 2; n=A, B) output signal will be generated respectively, which can be used to select and trigger other modules.

25.5 Register Description

Table 25-1 shows the register list of the Timer0 module.

Table 25-1 Timer0 Register List

BASE ADDR: 0x40024000(U1), 0x40024400(U2)

Register Name	Symbol	Offset	Bit Width	Reset Value
Register count value	TMRO_CNTAR	0x0000	32	0x00000000
Register count value	TMRO_CNTBR	0x0004	32	0x00000000
Reference value register	TMRO_CMPAR	0x0008	32	0x0000FFFF
Reference value register	TMRO_CMPBR	0x000c	32	0x0000FFFF
Basic control register	TMRO_BCONR	0x0010	32	0x00000000
Status flag register	TMRO_STFLR	0x0014	32	0x00000000

25.5.1 Count Value Register (TMR0_CNTmR) (m=A~B)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CNTA [15:0]															
Bit	Flag		Bit Name		Function										Read and write
b31~b16	Reserved		-		Read as "0"										R
b15~b0	CNTA [15:0]		Count value		Current timer count										R/W

25.5.2 Reference Value Register (TMR0_CMPmR) (m=A~B)

Reset value: 0x0000FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CMPA [15:0]															
Bit	Flag		Bit Name		Function										Read and write
b31~b16	Reserved		-		Read as "0"										R
b15~b0	CMPA [15:0]		Reference value		Setting the count reference to produce the Compare Match event										R/W

25.5.3 Basic Control Register (TMR0 _ BCONR)

Reset Value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
HICP B	HCLE B	HSTP B	HSTA B	-	ASYN CLKB	SYN CLKB	SYN SB		CKDIV B [3:0]		-	INT ENB	CAP MDB	CST B	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
HICP A	HCLE A	HSTP A	HSTA A	-	ASYN CLKA	SYN CLKA	SYN SA		CKDIV A [3:0]		-	INT ENA	CAP MDA	CST A	

Bit	Flag	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	R/W
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	R/W
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	R/W
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	R/W
b27	Reserved	-	Read as "0", write as "0"	R/W
b26	ASYNCLKB	Channel B asynchronous counting clock source selection	0: LRC 1: XTAL32	R/W
b25	SYNCLKB	Channel B synchronous counting clock source selection	0: PCLK1 1: Internal Hardware Triggering Event	R/W
b24	SYNSB	Channel B counting mode selection	0: Synchronous counting mode 1: Asynchronous counting	R/W
b23~b20	CKDIVB [3:0]	Channel B counting clock frequency division selection	Channel B counting clock frequency division selection: 0000: clock source 0001: clock source/2 0010: clock source/4 0011: clock source/8 0100: clock source/16 0101: clock source/32 0110: clock source/64 0111: clock source/128 1000: clock source/256 1001: clock source/512 1010: clock source/1024 Please do not set other values Note: The timer to be divided can be a variety of timer at asynchronous counting, PCLK1 at synchronous counting.	R/W
b19	Reserved	-	Read as "0", write as "0"	R/W
b18	INTENB	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	R/W
b17	CAPMDB	Function mode selection B	0: Comparison output function 1: Capture input function	R/W
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	R/W
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	R/W
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	R/W
b13	HSTPA	Hardware Trigger Stop A	Condition: Internal Hardware Trigger Event Valid 0: Timer stops invalid when condition matches	R/W

			1: When the condition matches, the timer stops working Condition: Internal Hardware Trigger Event Valid 0: Timer start invalid when condition matches 1: Timer is active when condition matches	
b12	HSTAA	Hardware Trigger Startup A		R/W
b11	Reserved	-	Read as "0", write as "0"	R/W
b10	ASYNCLKA	Channel A asynchronous counting clock source selection	0: LRC 1: XTAL32	R/W
b9	SYNCLKA	Channel A Synchronous Count Timer Selection	0: PCLK1 1: Internal Hardware Triggering Event	R/W
b8	SYNSA	Channel A counting mode selection	0: Synchronous counting mode 1: Asynchronous counting	R/W
b7~b4	CKDIVA [3:0]	Channel A Clock Frequency Division Selection	Channel A count clock frequency division options: 0000: Timer 0001: Timer/2 0010: clock source/4 0011: clock source/8 0100: clock source/16 0101: clock source/32 0110: clock source/64 0111: clock source/128 1000: clock source/256 1001: clock source/512 1010: clock source/1024 Please do not set other values Note: The timer to be divided can be a variety of timer at asynchronous counting, PCLK1 at synchronous counting.	R/W
b3	Reserved	-	Read as "0", write as "0"	R/W
b2	INTENA	Count Match interrupt enable A	0: CMPAR register 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	R/W
b1	CAPMDA	Functional mode selection A	0: Comparison output function 1: Capture input function	R/W
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	R/W

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 [General Synchronous Asynchronous Transceiver (USART)] section.

25.5.4 Status Flag Register (TMR0_STFLR)

Reset value: 0x00000000

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

Bit	Flag	Bit Name	Function	Read and write
b31~b18	Reserved	-	Read as "0", write as "0"	R/W
b17	ICPB	Repeated capture 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	R/W
b16	CMFB	Count match 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.	R/W
b15~b2	Reserved	-	Read as "0", write as "0"	R/W
b2	ICPA	Repeated capture 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	R/W
b0	CMFA	Count Match A	0: The value of CMPAR register is not equal to the count value (CNTAR) and no capture input action occurs 1: The value of CMPAR register is equal to the count value (CNTAR) or captures the input action.	R/W

Note:

- The flag bit in this register can only be written to 0, not 1.

25.6 Precautions

- In the asynchronous counting operation, first set the BCONR.ASYNCLKA bit to select the asynchronous clock source, then set the BCONR.SYNSA bit to select the asynchronous counting mode, and then start Timer0.
- When asynchronous counting is selected, when the count value (CNTAR), reference value (CMPAR), start bit (BCONR.CSTA), status bit (STFLR.CMFA), and repeated capture status bit (STFLR.ICPA) are modified, Timer0 will write the modified value to the corresponding register after 6 asynchronous counting clocks after receiving the write action.
- When asynchronous counting is selected, the count value (CNTAR), reference value (CMPAR), start bit (BCONR.CSTA), status bit (STFLR.CMFA), and repeated capture status bit (STFLR.ICPA) must be written continuously with an interval of at least 6 asynchronous counting clocks.
- To select asynchronous counting, set BCONR.SYNCLKA to 0.
- When writing to BCONR.CSTA, do not rewrite BCONR.SYNSA at the same time.
- When Timer0 is used as an event output, the reference value must be set to be greater than 5.

26 Real Time Clock (RTC)

26.1 Introduction

A real-time clock (RTC) is a counter that stores time information in BCD format. Record the specific calendar time from 00 to 99. Support 12/24 hour time system, automatically calculate the number of days 28, 29 (leap year), 30 and 31 according to the month and year. Table 26-1 Shown are its basic characteristics.

Table 26-1 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
Interrupt	Cycle interruption
	Alarm interrupt

26.2 Basic Block Diagram

The basic block diagram of the RTC is Figure 26-1 shown in the figure.

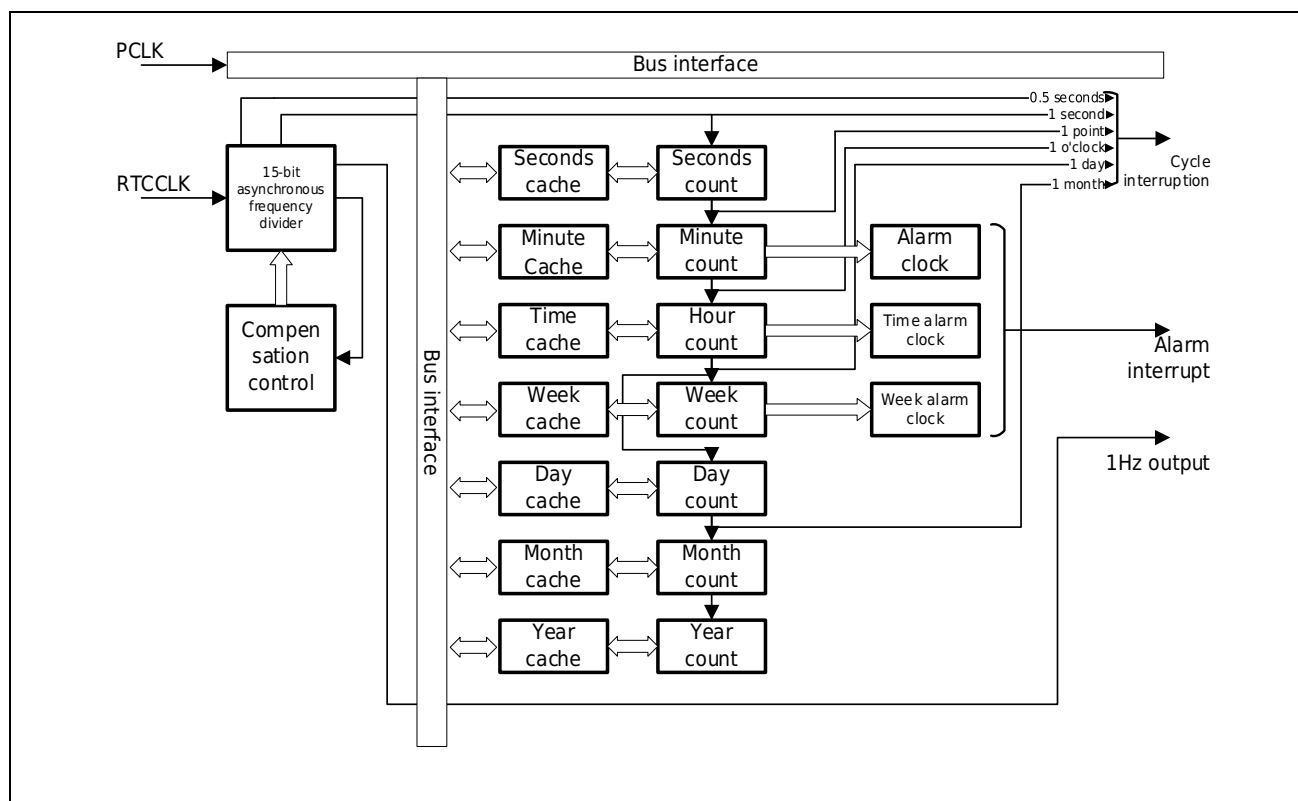


Figure 26-1 Basic block diagram of RTC

26.3 Functional Description

26.3.1 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.

26.3.2 RTC Count Start Setting

- 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.
- Set RTC_CR1.START=0, stop counting;
- Set the system clock register, turn on the external low-speed oscillator, then set RTC_CR3 to select the RTC count clock source;
- Set RTC_CR1, set time, period, 1Hz clock output;
- Set the calendar counting registers for seconds, minutes, hours, weeks, days, months, and years;
- When clock error compensation is required, set the counting clock error compensation registers RTC_ERRCRL, RTC_ERRCRH;
- Clear the flag register bit in the register RTC_CR2 and enable the interrupt;
- Set RTC_CR1.START=1, counting starts.

26.3.3 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.

- After setting RTC_CR1.START=1, switch modes after more than 2 RTC count clocks.
- After setting RTC_CR1.START=1, set RTC_CR2.RWREQ=1 and query RTC_CR2.RWEN=1. Set the calendar count register, then set RTC_CR2.RWREQ=0, query RTC_CR2.RWEN=0, and switch modes.

26.3.4 Read Count Register

- After RTC_CR1.START=1 is set, after more than 2 RTC count clocks, set RTC_CR2.RWREQ=1 to make a calendar register read request;
- Query until RTC_CR2.RWEN=1;
- Read all or part of the seconds, minutes, hours, weeks, days, months, and years counter register values;
- Set RTC_CR2.RWREQ=0;

- Query until RTC_CR2.RWEN=0.

26.3.5 Write Count Register

- After setting RTC_CR1.START=1, after more than 2 RTC counting clocks, set RTC_CR2.RWREQ=1 to make a calendar register write request;
- Query until RTC_CR2.RWEN=1;
- Write all or part of the seconds, minutes, hours, weeks, days, months, and years counter register values;
- Set RTC_CR2.RWREQ=0. Note that all write operations must be completed within 1 second;
- Query until RTC_CR2.RWEN=0.

26.3.6 Alarm Clock Setting

- Set RTC_CR2.ALME=0, the alarm clock is disabled;
- Set RTC_CR2.ALMIE=1, the alarm clock interruption permission;
- Minute alarm clock RTC_ALMMIN, hour alarm clock RTC_ALMHOUR, week alarm clock RTC_ALMWEED setting;
- Set RTC_CR2.ALME=1, alarm clock permission;
- Wait for the alarm to interrupt;
- When an alarm occurs, RTC_CR2.ALMF=1, and the alarm interrupt processing begins.

26.3.7 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. For compensation method, please refer to [Clock Error Compensation Registers (RTC_ERRCRH, RTC_ERRCRL)].

26.3.8 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:

- Set RTC_CR0.RESET=0, after confirming that the RESET bit is "0", set RTC_CR0.RESET=1, reset the calendar count register;
- Set RTC_CR1.START=0, counting stops;
- 1Hz output pin setting;

- RTC_CR1.ONEHZOE=1, clock output permission;
- Set RTC_CR1.START=1, start counting;
- Wait for more than 2 count periods;
- 1Hz output starts.

The Distributed Compensation 1Hz Output Setting is as Follows:

- Set RTC_CR0.RESET=0, after confirming that the RESET bit is "0", set RTC_CR0.RESET=1, reset the calendar count register;
- Set RTC_CR1.START=0, counting stops;
- 1Hz output pin setting;
- RTC_CR1.ONEHZOE=1, clock output permission;
- Clock error compensation register RTC_ERRCRL.COMP [7:0] and RTC_ERRCRH.COMP [8] compensation number setting;
- Clock error compensation register RTC_ERRCRH.COMPEN=1, error compensation is valid;
- Set RTC_CR1.START=1, start counting;
- Wait for more than 2 count periods;
- 1Hz output starts.

Uniform Compensation 1Hz Output Setting is as Follows:

- Set RTC_CR0.RESET=0, after confirming that the RESET bit is "0", set RTC_CR0.RESET=1, reset the calendar count register;
- Set RTC_CR1.START=0, counting stops;
- RTC output pin setting;
- RTC_CR1.ONEHZOE=1, clock output permission;
- RTC_CR1.ONEHZSEL=1, select output uniform compensation 1Hz clock;
- Clock error compensation register RTC_ERRCRL.COMP [7:0] and RTC_ERRCRH.COMP [8] compensation number setting;
- Clock error compensation register RTC_ERRCRH.COMPEN=1, accuracy compensation is valid;
- Set RTC_CR1.START=1, start counting;
- Wait for more than 2 count periods;
- 1Hz output starts.

26.4 Interrupt Description

RTC supports 3 interrupt types. Timing alarm interrupt and periodic interrupt.

26.4.1 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 resets the alarm flag.

26.4.2 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-period interrupt is configured with an independent flag register RTC_CR2.PRDF, and the fixed-period flag can be reseted by writing 0 to the RTC_CR2.PRDF bit.

26.5 Register Description

Table 26-2 shows the register list of the RTC module.

Table 26-2 Register List

Register base address: 0x4004C000

Register Name	Symbol	Offset	Bit Width	Reset Value
Control register 0	RTC_CR0	0x0000	8	0x00
Control register 1	RTC_CR1	0x0004	8	0x00
Control register 2	RTC_CR2	0x0008	8	0x00
Control register 3	RTC_CR3	0x000C	8	0x00
Second Count Register	RTC_SEC	0x0010	8	0x00
Minute counter register	RTC_MIN	0x0014	8	0x00
Time counter register	RTC_HOUR	0x0018	8	0x12
Week Count Register	RTC_WEEK	0x001C	8	0x00
Day Count Register	RTC_DAY	0x0020	8	0x00
Month Count Register	RTC_MON	0x0024	8	0x00
Year Count Register	RTC_YEAR	0x0028	8	0x00
Minute alarm register	RTC_ALMMIN	0x002C	8	0x00
Time alarm register	RTC_ALMHOUR	0x0030	8	0x12
Weekly Alarm Register	RTC_ALM WEEK	0x0034	8	0x00
Clock Error Compensation Register	RTC_ERRCRH	0x0038	8	0x00
Clock Error Compensation Register	RTC_ERRCRL	0x003C	8	0x20

26.5.1 Control Register 0 (RTC_CR0)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res							RESET

Bit	Flag	Bit Name	Function	Read and write
b7~b1	Reserved	-	Read as "0", write as "0"	R/W
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	R/W

26.5.2 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	Flag	Bit Name	Function	Read and write			
b7	START	RTC counting starts	0: RTC count stop 1: RTC counting starts	R/W			
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.	R/W			
b5	ONEHZOE	1Hz output enabled	0: 1Hz output disabled 1: 1Hz output permission	R/W			
b4	Reserved	-	Read as "0", write as "0"	R/W			
b3	AMPM	Time selection	0:12 hour clock 1: 24-hour time format	R/W			
b2~0	PRDS [2:0]	Cycle Interrupt Selection	Cycle selection settings:		R/W		
			PRDS [2]	PRDS [1]		PRDS [0]	Cycle selection
			0	0		0	Do not choose
			0	0		1	Every 0.5 second period
			0	1		0	Every 1 second period
			0	1		1	Every 1 minute period
			1	0		0	Every 1 hour period
			1	0		1	Every 1 day period (every day at 00:00:00:00)
			1	1		X	Every January period (00:00:00 on the 1st of each month)

Note: When writing period selection during START=1 counting, please disable the period interrupt permission to prevent malfunction. And the relevant flag should be cleared after writing.

26.5.3 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	Flag	Bit Name	Function	Read and write
b7	ALME	Alarm function enable	0: Alarm clock function disabled 1: Alarm clock function permission	R/W
b6	ALMIE	Alarm clock interrupt enable	0: Alarm interrupt disabled 1: Alarm interrupt disable	R/W
b5	PRDIE	Periodic interrupt enable	0: Periodic interrupt disabled 1: Periodic interrupt disable	R/W
b4	Reserved	-	Read as "0", write as "0"	R/W
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" resets the flag, and writing "1" invalid.	R/W
b2	PRDF	Cycle sign	0: The period does not occur 1: Periodic occurrence Note: After the set period occurs, this position is set to "1". Writing "0" resets the flag bit, writing "1" invalid.	R/W
b1	RWEN	Read/write enable	0: Read/write disabled 1: Read/write enabled Note: Calendar register read and write permission 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.	R/W
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" .	R/W

26.5.4 Control Register 3 (RTC_CR3)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
RCKSEL	Res		LRCEN	Res			

Bit	Flag	Bit Name	Function	Read and write
b7	RCKSEL	RTC count clock selection	0: Select the external low-speed oscillator XTAL32 clock 1 : Select another count clock You can select LRC , terminal input, or XTAL fractional clock. When selecting these count clocks, you need to configure the PWRC6.RTCKSEL1 [1:0] register bit.	R/W
b6	Reserved	-	Read as "0", write as "0"	R/W
b5	Reserved	-	Read as "0", write as "0"	R/W
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.	R/W
b3~b1	Reserved	-	Read as "0", write as "0"	R/W
b0	Reserved	-	Read as "0", write as "0"	R/W

26.5.5 Second Count Register (RTC_SEC)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res	SECD [2:0]			SECU [3:0]			

Bit	Flag	Bit Name	Function	Read and write
b7	Reserved	-	Read as "0", write as "0"	R/W
b6~b4	SECD [2:0]	Tens of seconds	Second tens digit count value	R/W
b3~b0	SECU [3:0]	Seconds	Second count value	R/W

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.

26.5.6 Minute Count Register (RTC_MIN)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res	MIND [2:0]			MINU [3:0]			

Bit	Flag	Bit Name	Function	Read and write
b7	Reserved	-	Read as "0", write as "0"	R/W
b6~b4	MIND [2:0]	Tens	Minute and tens count value	R/W
b3~b0	MINU [3:0]	Minutes	Minute count value	R/W

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.

26.5.7 Hour Count Register (RTC_HOUR)

Reset value: 0x12

b7	b6	b5	b4	b3	b2	b1	b0
Res	HOURL [1:0]			HOURU [3:0]			

Bit	Flag	Bit Name	Function	Read and write
b7~b6	Reserved	-	Read as "0", write as "0"	R/W
b5~b4	HOURL [1:0]	Time ten	Hour tens count value	R/W
b3~b0	HOURU [3:0]	When ones	Hour count value	R/W

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 01~12 means afternoon.

Please set the correct decimal 0~23 or 01~12, 01~12 BCD code according to the value of the control bit 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
00 o'clock	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
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

26.5.8 Day Count Register (RTC_DAY)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res		DAYD [1:0]		DAYU [3:0]			

Bit	Flag	Bit Name	Function	Read and write
b7~b6	Reserved	-	Read as "0", write as "0"	R/W
b5~b4	HOARD [1:0]	day ten	Day tens digit count value	R/W
b3~b0	HOARD [3:0]	Day units digit	Day units digit value	R/W

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

26.5.9 Week Count Register (RTC_WEEK)

Reset value: 0x00

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

Bit	Flag	Bit Name	Function	Read and write
b7~b3	Reserved	-	Read as "0", write as "0"	R/W
b2~b0	WEEK [2:0]	week	Week count	R/W

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	00H
Monday	01H
Tuesday	02H
Wednesday	03H
Thursday	04H
friday	05H
Saturday	06H

26.5.10 Month Count Register (RTC_MON)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res				MON [4:0]			

Bit	Flag	Bit Name	Function	Read and write
b7~b5	Reserved	-	Read as "0", write as "0"	R/W
b4~b0	MON [4:0]	January	Month count	R/W

Decimal 1~12 means Sunday~ 12 January. Please write in the correct decimal 1~12 BCD code, other values will be ignored.

26.5.11 Year Count Register (RTC_YEAR)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
YEAR [3:0]				YEARU [3:0]			

Bit	Flag	Bit Name	Function	Read and write
b31~b8	Reserved	-	Read as "0", write as "0"	R/W
b7~b4	YEAR [3:0]	year ten	Year tens digit count value	R/W
b3~b0	YEARU [3:0]	Year unit digit	Year count value	R/W

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.

26.5.12 Minute Alarm Register (RTC_ALMMIN)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res	ALMMIND [2:0]			ALMMINU [3:0]			

Bit	Flag	Bit Name	Function	Read and write
b7	Reserved	-	Read as "0", write as "0"	R/W
b6~b4	ALMMIND [3:0]	Alarm clock Alarm tens	Alarm clock Alarm tens match Value	R/W
b3~b0	ALMMINU [3:0]	minute alarm clock	Alarm clock Alarm bit match Value	R/W

Please set the BCD code of decimal 0~59.

26.5.13 Hour Alarm register (RTC_ALMHOUR)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res	ALMHOURD [1:0]			ALMHOURU [3:0]			

Bit	Flag	Bit Name	Function	Read and write
b7~b6	Reserved	-	Read as "0", write as "0"	R/W
b5~b4	ALMHOURD [1:0]	alarm clock tens	Alarm When Alarm tens match Value	R/W
b3~b0	ALMHOURU [3:0]	alarm clock unit	Alarm When Alarm bit match Value	R/W

Please set the correct alarm value according to the time system.

26.5.14 Week Alarm Register (RTC_ALMWEEEK)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Res	ALMWEEEK [6:0]						

Bit	Flag	Bit Name	Function	Read and write
b7	Reserved	-	Read as "0", write as "0"	R/W
b6~b0	ALMWEEEK [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.	R/W

Please set the correct alarm value according to the time system.

26.5.15 Clock Error Compensation Registers (RTC_ERRCRH, RTC_ERRCRL)

Reset value: 0x00

RTC_ERRCRH

b7	b6	b5	b4	b3	b2	b1	b0
COMPEN	Res						COMP [8]

Reset value: 0x20

RTC_ERRCRL

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

RTC_ERRCRH

Bit	Flag	Bit Name	Function	Read and write
b7	COMPEN	Compensation enable	0: Clock error compensation invalid 1: Clock error compensation valid	R/W
B6~b1	Reserved	-	Read as "0", write as "0"	R/W
b0	COMP [8]	compensation value	Set the compensation value together with COMP [7:0]	R/W

RTC_ERRCRL

Bit	Flag	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:									
			Compensation value setting									Compensation
			COMPEN	COMP [8:0]								
			1	1	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	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 = $32768 \times 0.9999888 \approx 32767.63$

Compensation target value = $(\text{actual oscillation frequency} - \text{target frequency}) / \text{target frequency} \times 10^6$

$$= (32767.63 - 32768) / 32768 \times 10^6$$

$$= -11.29\text{ppm}$$

Setpoint calculation:

$$\text{COMP}[8:0] = \left(\frac{\text{Compensation target value [ppm]} \times 2^{15}}{10^6} \right)_{\text{Take 2's complement}} + 0001.00000\text{B}$$

If the compensation target value is +20.3ppm, calculate the corresponding register value as follows:

$$\begin{aligned} \text{COMP}[8:0] &= (20.3 \times 2^{15} / 10^6)_{\text{Take 2's complement}} + 0001.00000\text{B} \\ &= (0.6651904)_{\text{take 2's complement}} + 0001.00000\text{B} \\ &= 0000.10101\text{B} + 0001.00000\text{B} \\ &= 0001.10101\text{B} \end{aligned}$$

If the compensation target value is -20.3ppm, calculate the corresponding register value as follows:

$$\begin{aligned} \text{COMP}[8:0] &= (-20.3 \times 2^{15} / 10^6)_{\text{Take 2's complement}} + 0001.00000\text{B} \\ &= (-0.6651904)_{\text{take 2's complement}} + 0001.00000\text{B} \\ &= 1111.01011\text{B} + 0001.00000\text{B} \\ &= 0000.01011\text{B} \end{aligned}$$

27 Watchdog Counter (WDT/SWDT)

27.1 Introduction

There are two watchdog counters, one is the dedicated watchdog counter (SWDT) whose counting clock source is dedicated internal RC (SWDTLRC: 10KHz), and the other is the general watchdog counter (WDT) whose counting clock source is PCLK3. The dedicated watchdog and general 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 27-1.

Table 27-1 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 frequency division of PCLK3
Maximum overflow time	SWDT:3.72hour(max) WDT:10.7s (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

27.2 Functional description

27.2.1 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.

27.2.2 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 27-1 Hardware Startup Instance is an instance of how the hardware is started.

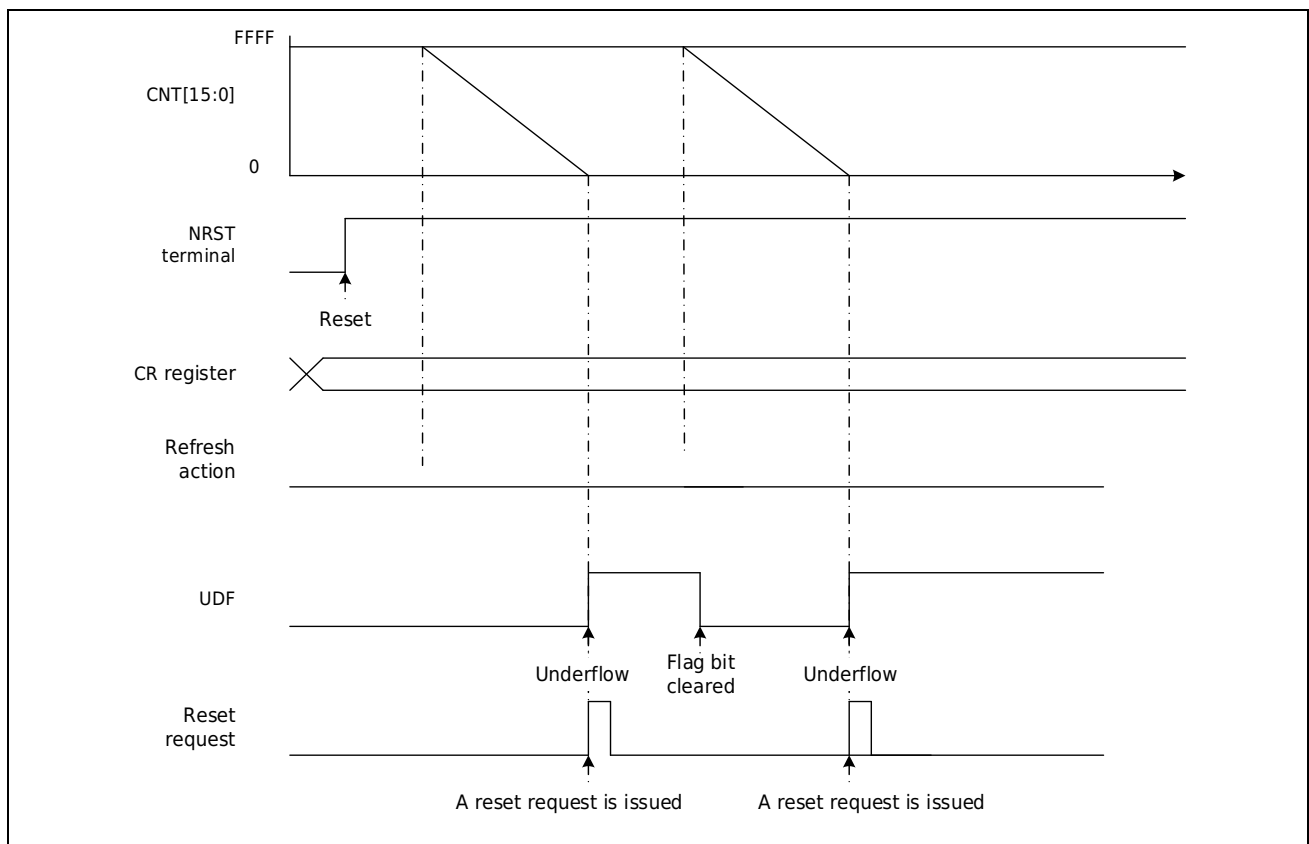


Figure 27-1 Hardware Startup Instance

27.2.3 Software Startup Method

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 27-2 This is an instance of the operation of the software startup method.

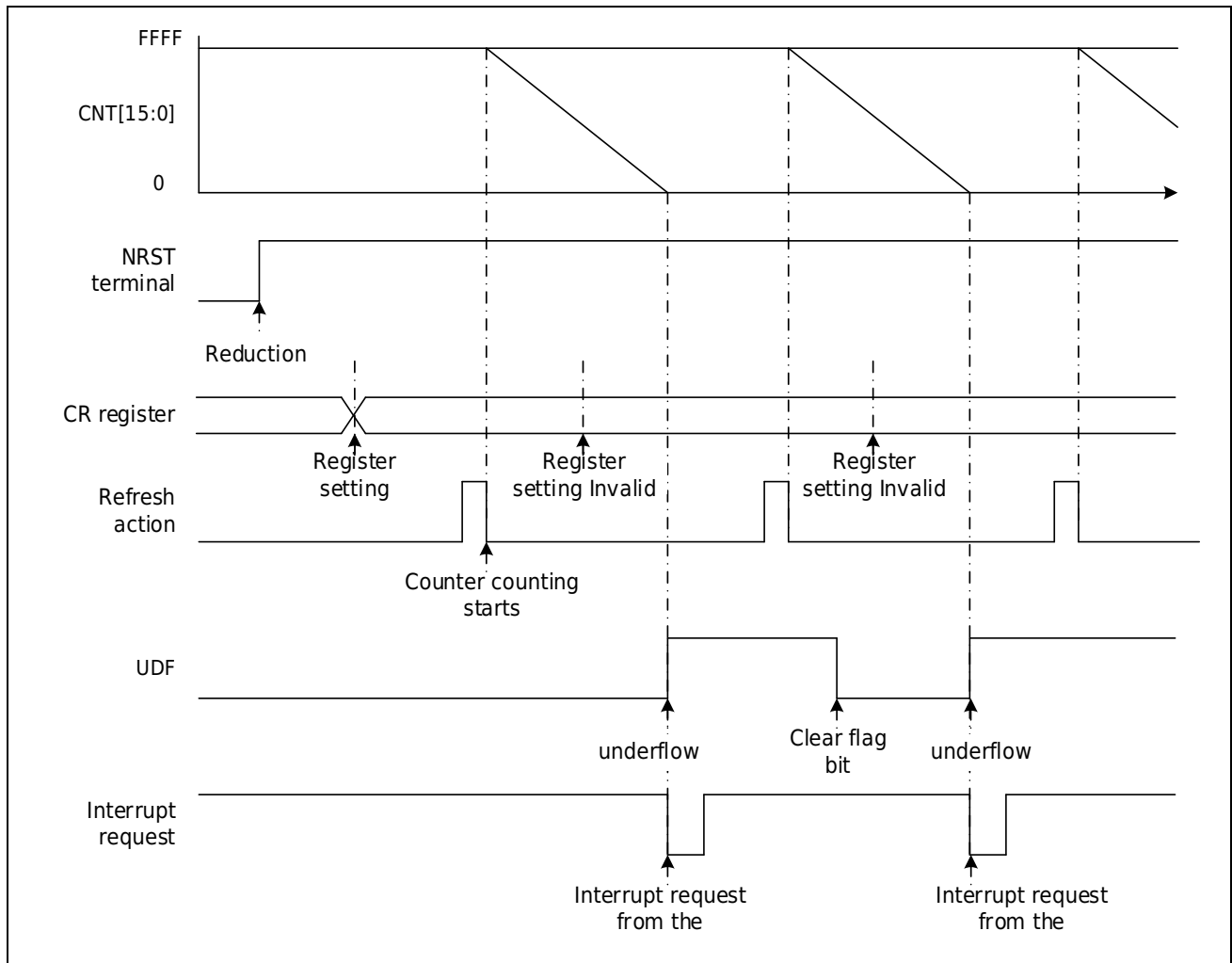


Figure 27-2 Software Startup Instance

27.2.4 Refresh Action

(S)WDT_RR register writes 0x0123 first and 0x3210 then completes a refresh action. WDT/SWDT's counter starts counting (software startup) or starts counting again.

(S)WDT_RR register is between write 0x0123 and write 0x3210. If you access or read (S)WDT_RR register to another register, the normal refresh action is not affected.

As Figure 27-3 shown action.

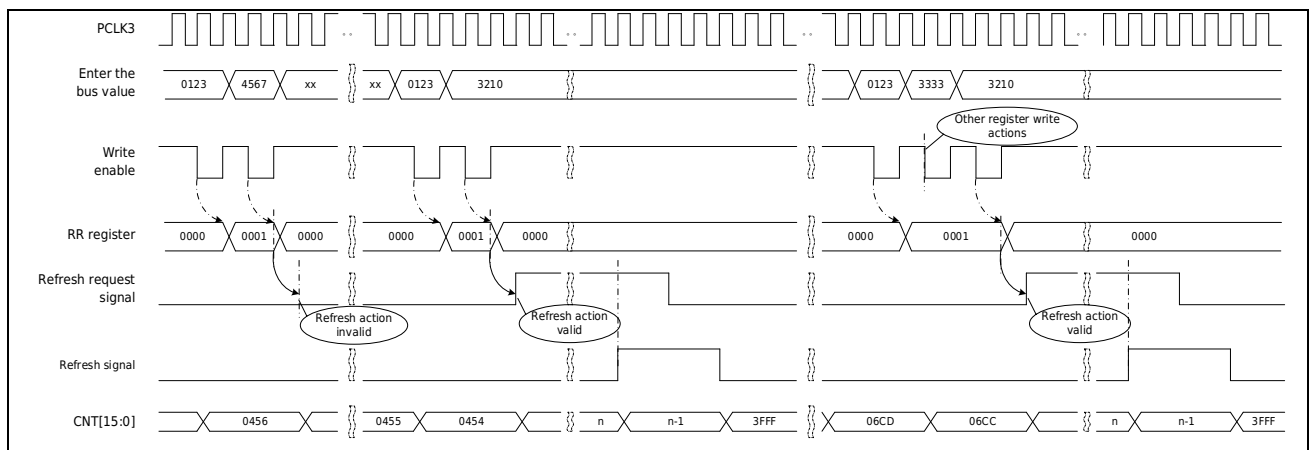


Figure 27-3 Instances of Refresh Action Timing (Action Confirmation, Refresh Request Signal Falling Edge, etc.)

The refresh action requires four count periods 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.

27.2.5 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. Flag bit zeroing: Read "1" before writing "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 reset; WDT needs at most 5 PCLK3 times before the register bit can be reset. In addition, within a certain period of time when a refresh error or underflow error occurs, it is invalid to read "1" and reset the flag bit. This period of time is: 1 counting period + 2 SWDTLRC (SWDT module); 1 counting period + 2 PCLK3 (WDT module).

27.2.6 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 WDT/SWDT interrupt reset generation. One is that counter counts produce underflow; One is to perform a refresh action outside the refresh allowed interval, resulting in a refresh error.

27.2.7 Count Underflow

For Figure 27-4 instance, underflow occurs when the decrement counts to zero as shown in Instances of counter underflow.

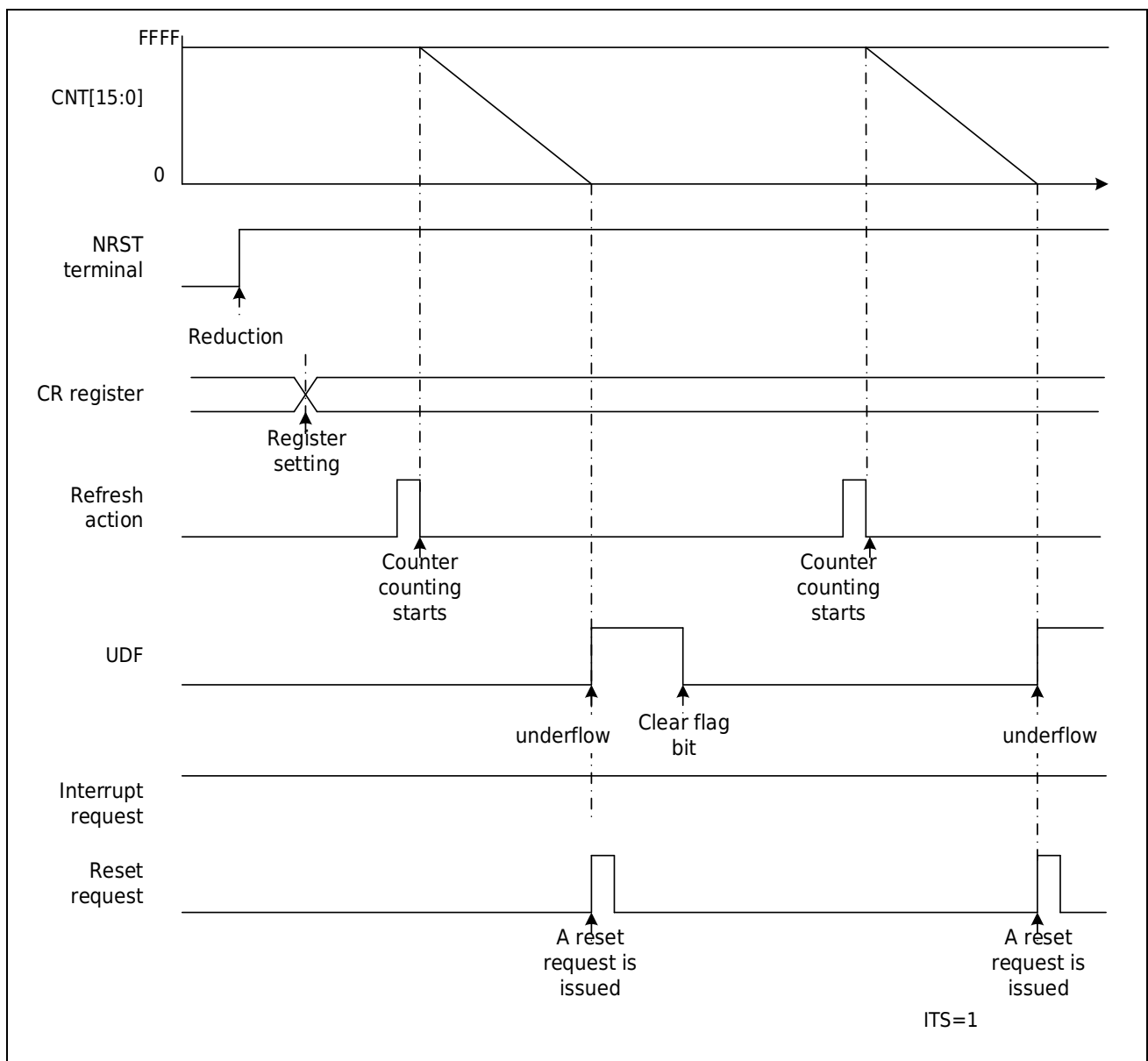


Figure 27-4 Counter Underflow Action Instance

27.2.8 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 27-5 is an example of a refresh action.

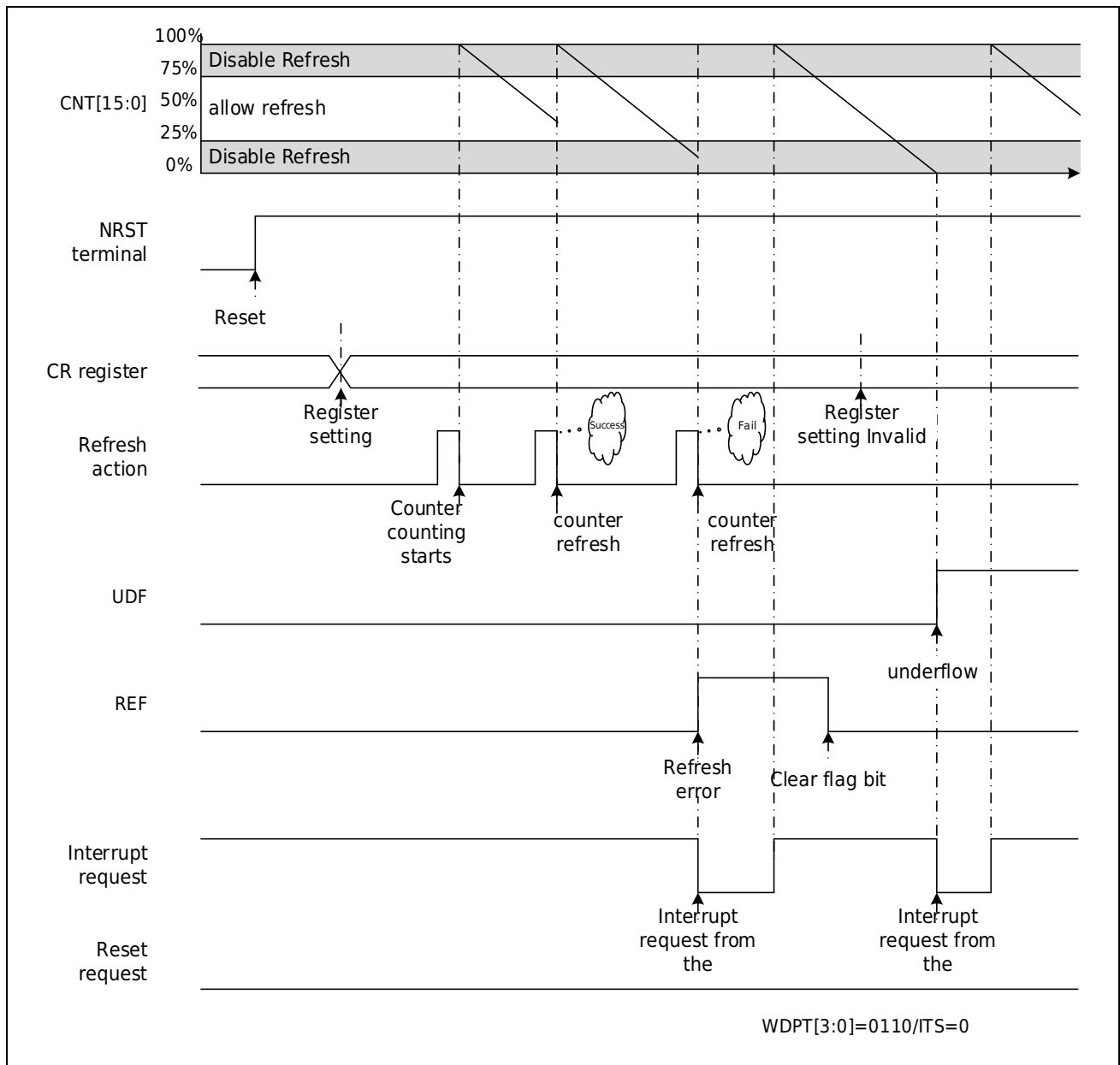


Figure 27-5 Counter Refresh Action Instance

27.3 Register Description

Table 27-2 Shown is the register list of the WDT and SWDT module.

Table 27-2 Register List

WDT_BASE_ADDR: 0x40049000

SWDT_BASE_ADDR: 0x40049400

Register name	Symbol	Offset Address	Bit Width	Reset Value
SWDT control register	SWDT_CR	0x00	32	0x8001_0FF3
SWDT status register	SWDT_SR	0x04	32	0x0000_0000
SWDT Refresh register	SWDT_RR	0x08	32	0x0000_0000
WDT control register	WDT_CR	0x00	32	0x8001_0FF3
WDT status register	WDT_SR	0x04	32	0x0000_0000
WDT Refresh register	WDT_RR	0x08	32	0x0000_0000

27.3.1 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	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SLPOFF
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	WDPT				CKS [3:0]				-	-	PERI [1:0]	

Bit	Flag	Bit Name	Function	Read and write
b31	ITS	Refresh Error/Overflow Interrupt/Reset Selection	0: Interrupt request 1: Reset request	R/W
b30~b17	Reserved	-	Read as "0", write as "0"	R/W
b16	SLPOFF	(S)WDT counting is 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	R/W
b15~b12	Reserved	-	Read as "0", write as "0"	R/W
b11~b8	WDPT [3:0]	Refresh Allowed Area Count%	0000: 0%~100%	R/W
			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%	
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	R/W

			0101: SWDTCLK/32	
			0110: SWDTCLK/64	
			0111: SWDTCLK/128	
			1000: SWDTCLK/256	
			1011: SWDTCLK/2048	
			Other values: Reserved	
b3~b2	Reserved	-	Read as "0", write as "0"	R/W
			00: 256 cycle	
b1~b0	PERI [1:0]	Counting period	01: 4096 cycle	R/W
			10: 16384 cycle	
			11: 65536 cycle	

27.3.2 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
-	-	-	-	-	-	-	-	-	-	-	-	-	-	REF	UDF
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CNT [15:0]															

Bit	Flag	Bit Name	Function	Read and write
b31~b18	Reserved	-	Read as "0", write as "0"	R/W
b17	REF	Refresh error flag	0: No refresh error 1: Refresh error occurred to the alignment After reading 1 write 0 and reset.	R/W
b16	UDF	Counting underflow flag	0: No count underflow 1: Occurrence count underflow to the alignment After reading 1 write 0 and reset.	R/W
b15~b0	CNT [15:0]	Count value	Counter current count	R

27.3.3 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
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RF [15:0]															

Bit	Flag	Bit Name	Function	Read and write
b31~b16	Reserved	-	Read as "0", write as "0"	R/W
b15~b0	RF [15:0]	Refresh value	Write 0x0123 and 0x3210 to complete the refresh. After the register is written with 0x0123h, the read register is 0x00000001; the read value of the rest is 0x00000000.	R/W

27.4 Precautions

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 $\geq$ counting clock frequency $\times 4$.

28 General Synchronous Asynchronous Transceiver (USART)

28.1 Introduction

The product is equipped with 6 units of General Serial Transceiver Module (USART). The general serial transceiver module (USART) can flexibly exchange full-duplex data with external devices; this USART supports the general asynchronous serial communication interface (UART), clock synchronous communication interface, smart card interface (ISO/IEC7816-3) and LIN communication interface. Support modem operation (CTS/RTS operation), multiprocessor operation. Cooperate with Timer0 module to support UART receive TIMEOUT function. USART_1 supports the STOP mode wake-up function via the RX line.

The specific functions are allocated as follows:

- 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: USART_3, USART_6 supported
- Smart Card: USART_1, USART_2, USART_4, USART_5 support
- UART receive timeout function: supported by USART_1, USART_2, USART_4, USART_5

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
- Modem operation supported (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

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
- Timer: Internal timer (clock generated by internal baud rate generator)/External timer (clock entered by 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 sync break (BF) 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

28.2 USART System Block Diagram

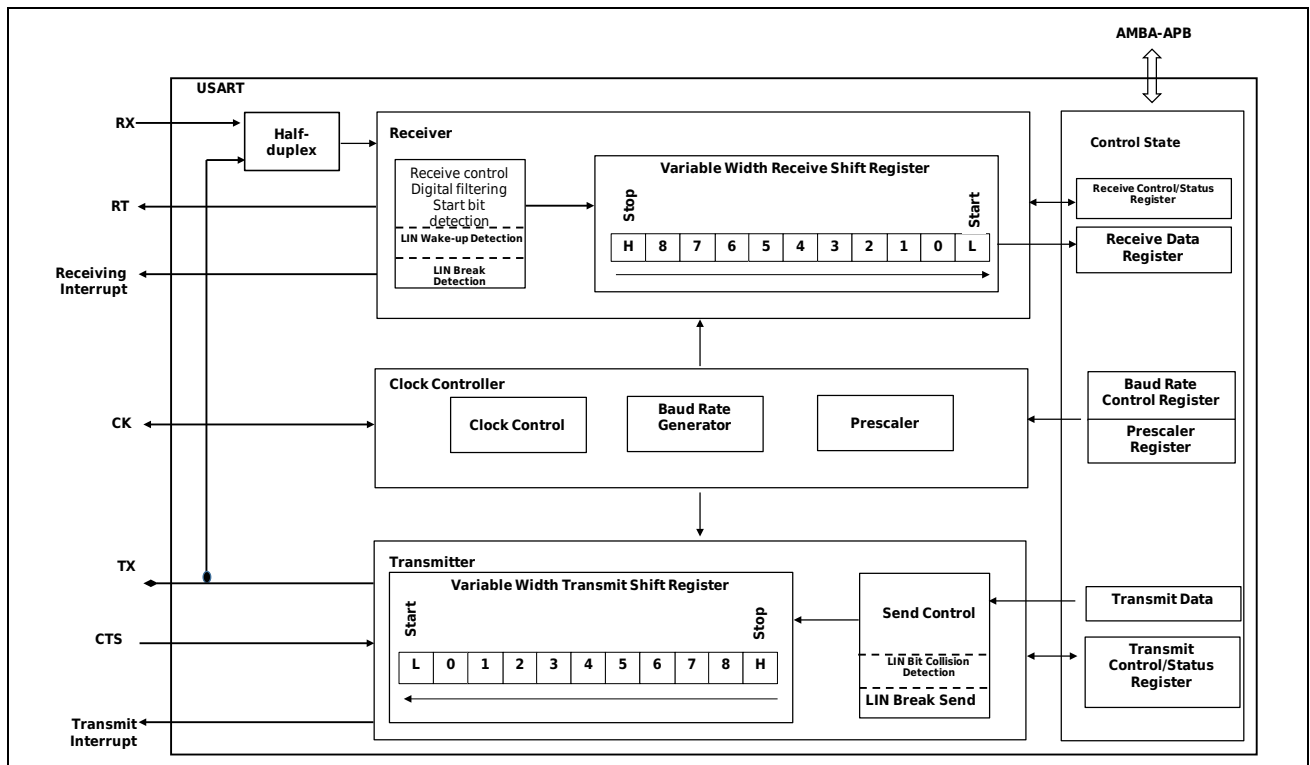


Figure 28-1 USART System Block Diagram

28.3 Pin Description

Table 28-1 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	Output	Modem operation pin Request send pin

n:1~6

28.4 Functional Description

This chapter will detail the functions of UART, multiprocessor communication, smart card, and clock synchronization mode.

28.4.1 UART

28.4.1.1 Clock

UART can select the clock generated by the internal baud rate generator (internal timer) or the clock entered by the USARTn _ CK pin (external timer) as the clock source for communication.

Internal Clock Source

USARTn_CR2. When the CLKC [1: 0] bit is set to 00b or 01b, select timer as the clock generated by the internal timer or internal baud rate generator.

USARTn_CR2. The USARTn _ CK pin is not used as a clock pin at CLKC [1: 0] = 00b and can be used as a common IO.

USARTn_CR2.CLKC [1: 0] = 01b outputs the clock at the same frequency as the communication baud rate from the USARTn _ CK pin.

Timer of internal baud rate generator is set to PCLK, PCLK/4, PCLK/16, PCLK/64 by USARTn _ PR. PSC [1: 0] bit.

External Clock Source

USARTn_CR2.CLKC [1: 0] bit is set to 10b or 11b, the clock source is selected as 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 timer, the Baud Rate formula generated by the internal Baud Rate Generator is:

B: Baud rate unit: MBps

C: Clock set by USARTn_PR.PSC [1:0] bits (PCLK, PCLK/4, PCLK/16, PCLK/64) Unit: MHz

OVER8: USARTn_CR1.OVER8 setting

DIV _ Integer: USARTn _ BRR.DIV _ Integer setting

The highest baud rate was PCLK/8 (MBps).

When the external clock source is used, the highest frequency of the external input UART clock is required to be PCLK(MHz)/4, so when the clock source is the external input clock, the highest baud rate is PCLK/64(MBps) (when USARTn_CR1.OVER8=0) or PCLK /32 (MBps) (when USARTn_CR1.OVER8=1).

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 Electrical Characteristics chapter.

28.4.1.2 Data Format

A frame of data in UART mode consists of start bits, data bits, check bits, and stop bits.

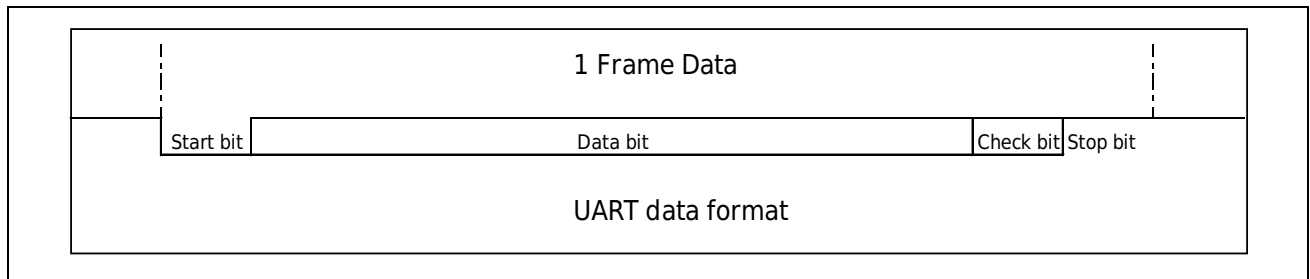


Figure 28-2 UART Data Format

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.

28.4.1.3 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_CTS pin 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 _ DR.RDRregister
- No receive errors, including frame errors, check errors and overflow errors

28.4.1.4 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 register TDR 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 USARTn_CR1register to reset
2. Set the pin that UART needs to use
3. Through USARTn_CR2.CLKC [1: 0] bit selection timer
4. Set USARTn_CR1, USARTn_CR2, USARTn_CR3register
5. Set USARTn_PR to select the pre-frequency division value and USARTn_BRRregister to set the communication baud rate (not required when timer is external timer).
6. Enable the sender (USARTn_CR1.TE=1). If the send data register null interrupt is required, set USARTn_CR1.TXEIE=1.
7. Wait to send data register empty, write communication data to USARTn_DR.TDR, transfer data to send shift register, send start
(When the CTS feature is active, the USARTn_CTS input is at low level and the data is transmitted to the send shift register, and the send starts)
8. Repeat Step 7 if you need to send data continuously
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.

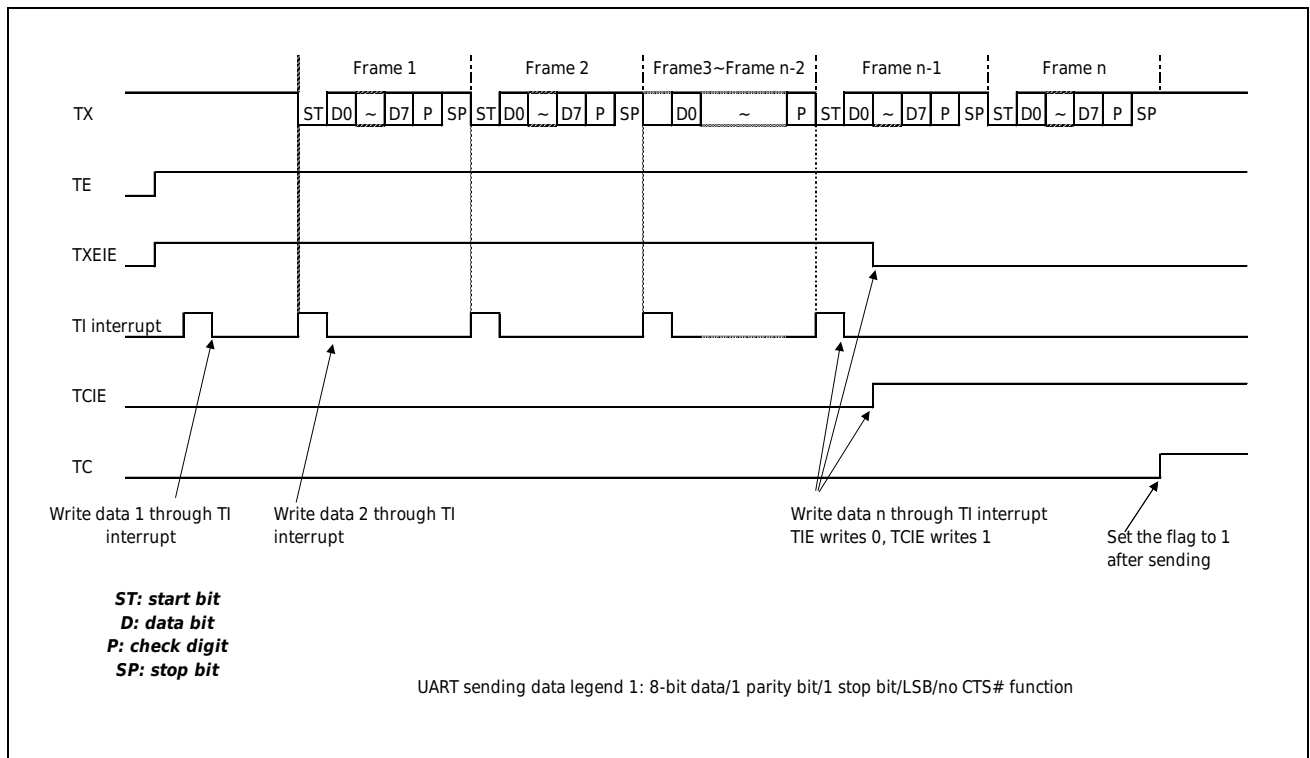


Figure 28-3 UART Send Data Legend 1

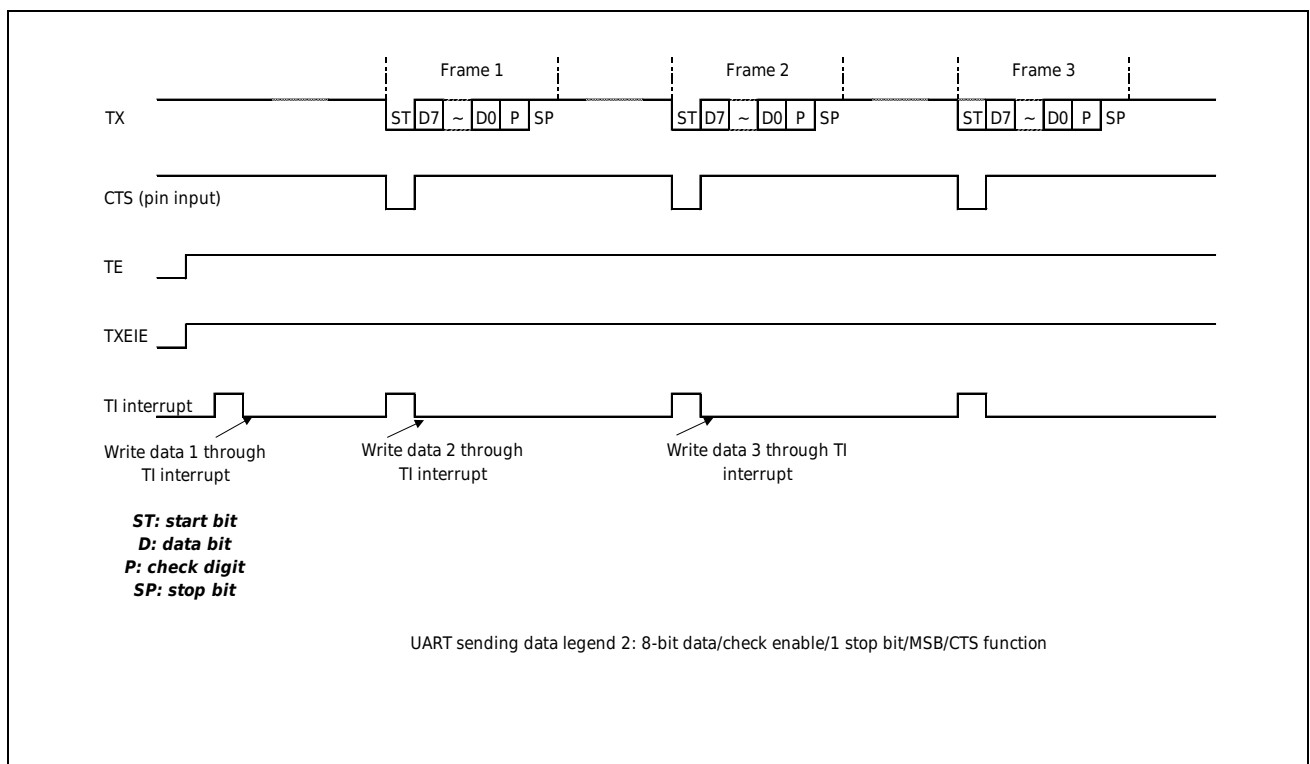


Figure 28-4 UART Send Data Legend 2

Transmitter Interrupt

The UART mode transmitter supports two types of interrupts, sending data register null interrupt (TI) and sending complete interrupt (TC).

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 TC interrupt occurs.

28.4.1.5 Receiver

Receivers can receive 8 or 9 bits of data depending on USARTn_CR1. The set value of the M bit. After the receiver sets the energy bit (USARTn_CR.RE) to 1 and detects the start bit, the RX pin receives the received shift register, fills a frame of data, and the data is transferred from the received shift register to the received data register USARTn_DR.RDR.

The sequence of receiving data is: Start bit - > Data bit (MSB/LSB) - > Check bit (with or without) - > 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.

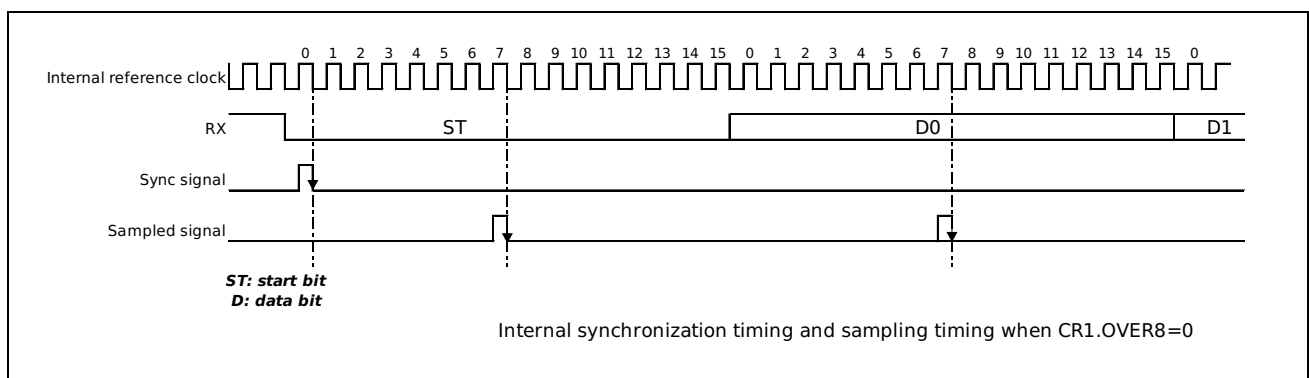


Figure 28-5 UART Internal Synchronization and Sampling Timing

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. 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 28-2 Tolerance of the UART Receiver when DIV_Fraction is 0

FL	OVER8 bit=0	OVER8 bit=1
10	4.375%	3.75%
11	3.97%	3.41%
12	3.646%	3.125%

Table 28-3 Tolerance of UART receiver when DIV_Fraction is not 0

FL	OVER8 bit=0	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 29-2 and Table 29-3 may be slightly different.

Receive Data Setting Steps

1. Set USARTn_CR1 register to reset
2. Set the pin that UART needs to use
3. Through USARTn_CR2.CLKC [1: 0] bit selection timer
4. Set USARTn_CR1, USARTn_CR2, USARTn_CR3 register
5. Set USARTn_PR to select the pre-frequency division value and USARTn_BRR register to set the communication baud rate (not required when timer is external timer).
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.RDR register 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.RDR register 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_DR.RDR register and the USARTn_SR.RXNE flag is set. After reading the received data, repeat step 7 to continuously receive data.

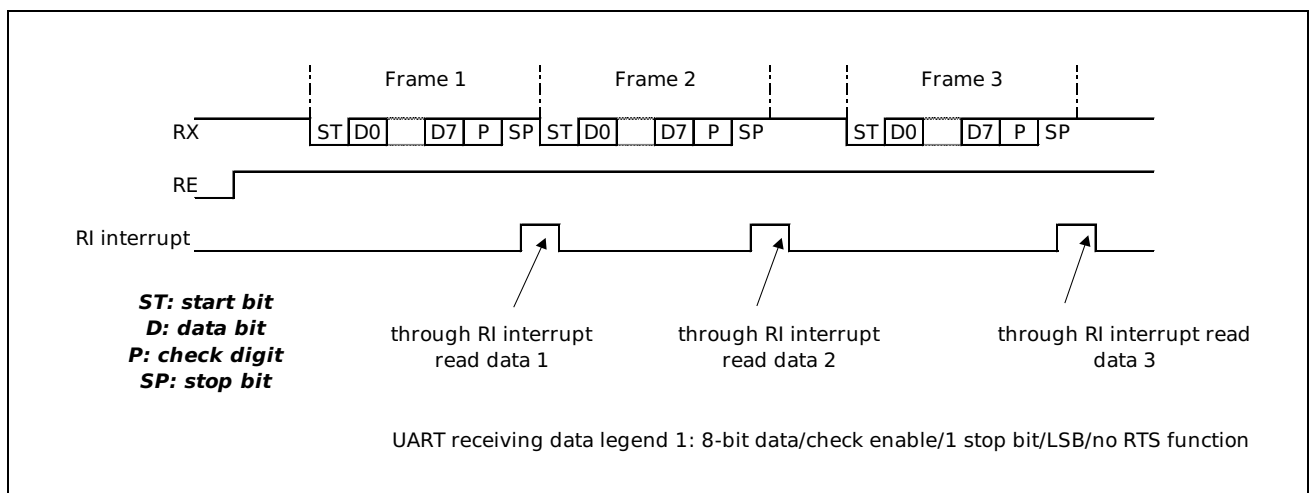


Figure 28-6 UART Received Data Legend 1

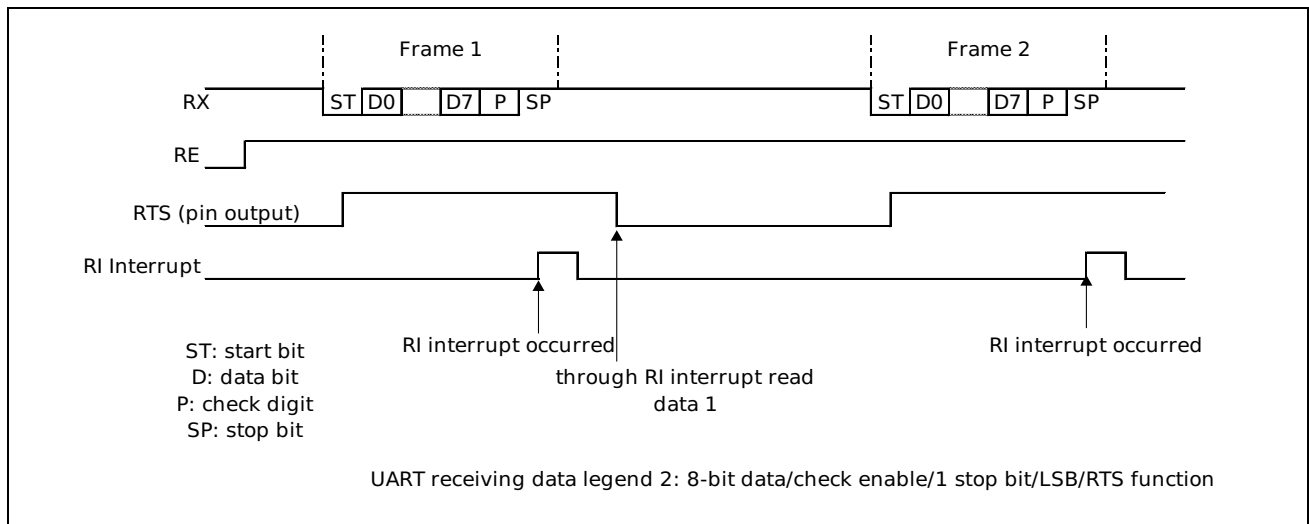


Figure 28-7 UART Received Data Legend 2

Error Handling

There are three types of receiving errors at the time of receiving data: Overflow error (USARTn _SR.ORE), verification 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, Rlinterrupt 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 Rlinterrupt 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.

28.4.1.6 UART Receive TIMEOUT Function

When the stop bit of UART received data is detected, the TIMEOUT counter starts. When the next frame of received data is not detected after the set TIMEOUT time (the setting unit is the number of received bits), TIMEOUT occurs. If CR1.RE=1 at this time, Then the TIMEOUT status bit USARTn_SR.RTOF is set. If USARTn_CR1.RE=0 at this time, wait for USARTn_CR1.RE=1 and then the TIMEOUT status bit USARTn_SR.RTOF is set.

The TIMEOUT counter uses the counter of the Timer0 module. The Timer0 counting mode needs to be set to asynchronous counting mode, and the asynchronous counting clock source is set to XTAL32 (Note: The Timer0 counting clock is provided by the clock tick generated by the UART baud rate generator, and the XTAL32 crystal oscillator may not exist). The specific corresponding relationship is as follows:

USART_1: Timer0 Unit1 A channel

USART_2: Timer0 Unit1 B channel

USART_4: Timer0 Unit2 A channel

USART_5: Timer0 Unit2 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_CMPAR value setting is as follows:

CMPAR: TMR0_CMPAR register value. Please round up the calculated result.

α : Delay caused by Timer0 asynchronous counting synchronization circuit

When the counting clock is not divided, $\alpha = 7$

When the counting clock is divided by 2, $\alpha = 5$

When the counting clock is divided by 4, 8, or 16, $\alpha = 3$

Count clock divided by 32 or more, $\alpha = 2$

RTB: Receive Timeout Bits, minimum value = Receive data frame length + $\alpha \times 2^{CKDIRA}$

There is a certain error between the actual time when TIMEOUT occurs and the RTB value, the error is $\leq 2^{CKDIRA}$.

CKDIRA: TMR0.BCONR.CKDIRA 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 by USARTn_CR2.CLK [1:0] bits (if you choose the internal clock source, you need to set CR2.CLKC [0]=1)
4. Set USARTn_CR1, USARTn_CR2, USARTn_CR3 register
5. Set USARTn_PR to select the pre-frequency division value and USARTn_BRR register to set the communication baud rate (not required when timer is external timer).
6. USARTn_CR1.RTOE=1, if you need to use interrupts, set USARTn_CR1.RTOIE=1;
7. Set TMR0.BCONR.CSTA=0
8. Set TMR0_CNTAR to 0, set TMR0_CMPAR register and TMR0.BCONR.CKDIVA register to determine TIMETOU time,
9. Set TMR0.BCONR.HCLEA=1, TMR0.BCONR.HSTAA=1, TMR0.BCONR.ASYNCLKA=1, TMR0.BCONR.SYNCLKA=1, TMR0.BCONR.SYNSA=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=1 and TMR0.BCONR.SYNA=0 at the same time, then set TMR0.BCONR.CSTA=0, TMR0.BCONR.SYNCLKA and TMR0.BCONR.SYNSA=1 at the same time.
Clear the USARTn_SR.RTOF status bit by writing USARTn_CR1.CRTOF.

28.4.1.7 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 the UART communication is idle, set the USART_1_WUPI interrupt vector and INT_WUPEN. The RXWUEN bit enables the UART receive signal line to wake up the 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.

UART RX line wake-up function can filter the noise on the RX line. For details, please refer to USART1_NFC register.

28.4.1.8 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.

28.4.1.9 UART interrupt and Events

Table 28-4 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	No
TIMEOUT interrupt	RTOIE	RTOF	Yes
RX stop mode wake-up interrupt	INT_WUPEN.RXWUEN	-	No

28.4.2 Multiprocessor Communication

28.4.2.1 Function Introduction

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.

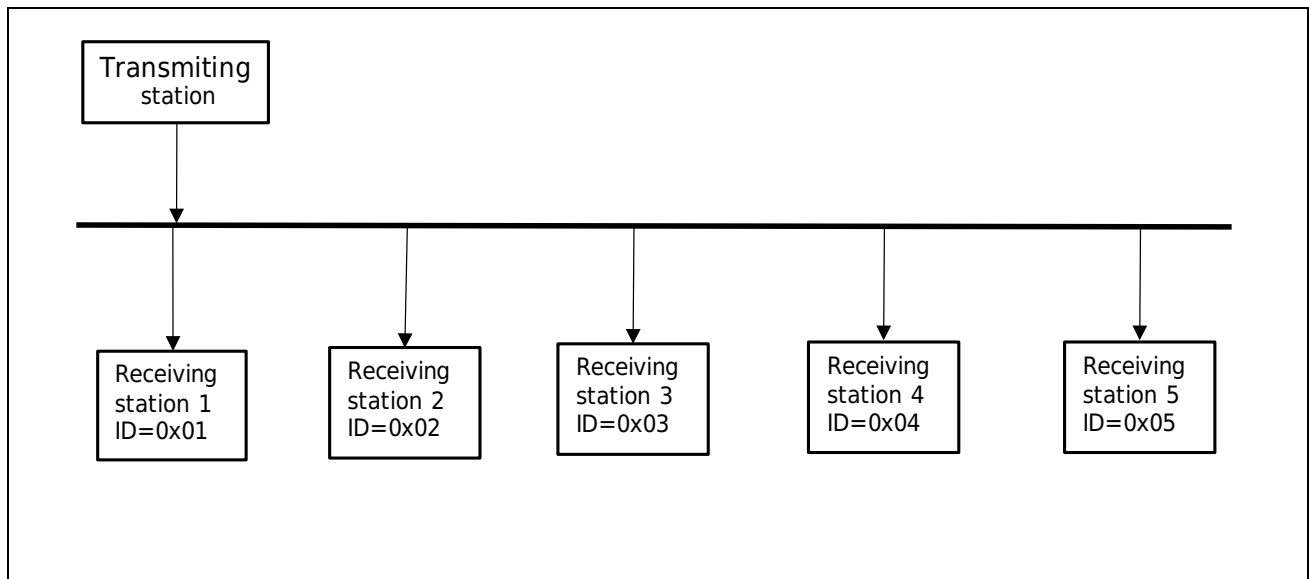


Figure 28-8 Multiprocessor Communication Diagrams

28.4.2.2 Data Format

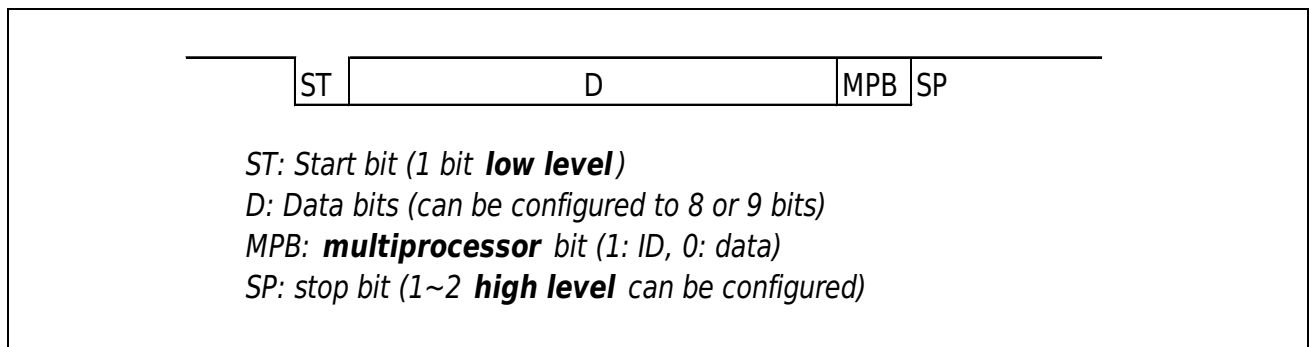


Figure 28-9 Multiprocessor Mode Data Format

28.4.2.3 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 USARTn_CR1 register to reset
2. Set the pins to be used
3. Through USARTn_CR2.CLKC [1: 0] bit selection timer
4. Set USARTn_CR1, USARTn_CR2, USARTn_CR3 register
5. Set USARTn_PR to select the pre-frequency division value and USARTn_BRR register to set the communication baud rate (not required when timer is external timer).
6. Enable the sender (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, set USARTn_TDR.MPID bit to 1 (send ID), write ID value to USARTn_TDR, send ID

- (When the CTS feature is active, the USARTn_CTS input is at low level and the data is transmitted to the send shift register, and the send starts)
8. Set USARTn_TDR.MPID bit to 0 (send data), write data to USARTn_TDR, send data
(When the CTS feature is active, the USARTn_CTS input low level data transfer to the send shift register, send start)
 9. If you need to send the data continuously, repeat step 8. If you need to change the ID, repeat steps 7 and 8.
 10. 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.

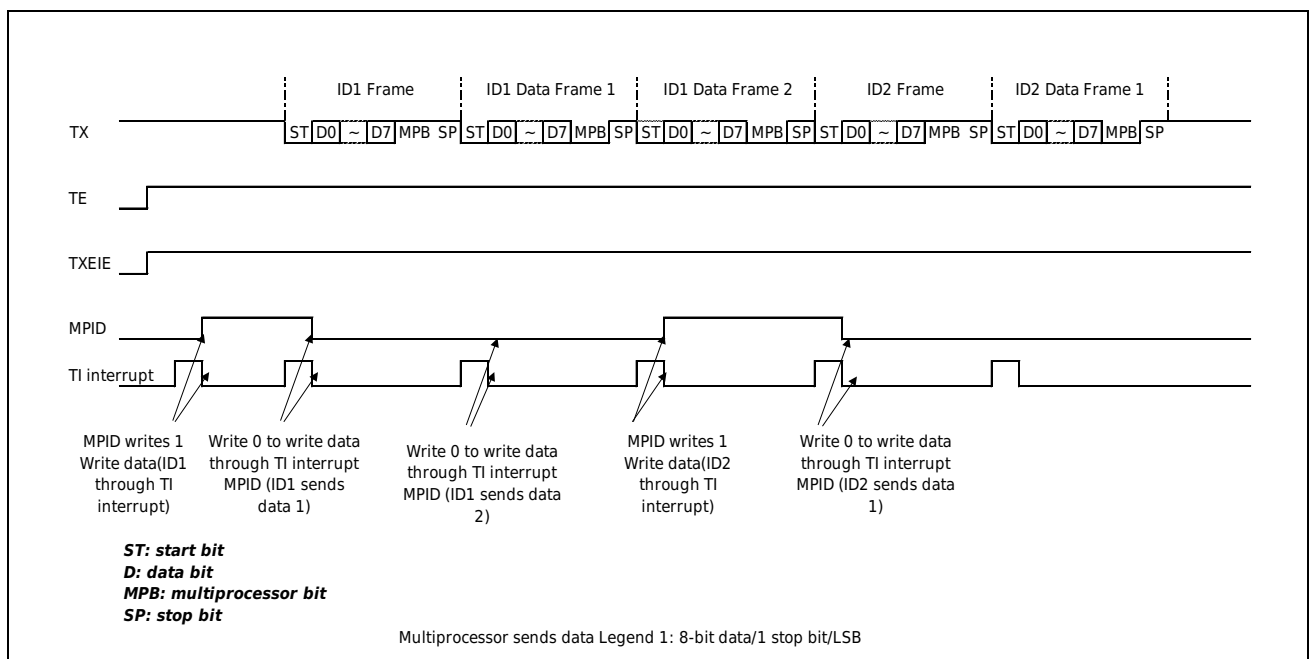


Figure 28-10 Diagram of Multi-Processor Mode Send Data

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.

USARTn_CR1. When SLME = 1, no data will be received unless data with MPB bit 1 (ID) is received, no Rlinterrupt will occur, and the error flags FE and ORE will not be set. When data with MPB bit 1 is received (ID), USARTn_CR1.SLME bits are cleared automatically, data are normally received and interruptd.

Action steps:

1. Set USARTn_CR1 register to reset
2. Set the pins to be used
3. Through USARTn_CR1.CLKC [1: 0] bit selection timer
4. Set USARTn_CR1, USARTn_CR2, USARTn_CR3 register
5. Set USARTn_PR to select the pre-frequency division value and USARTn_BRR register to set the communication baud rate (not required when timer is external timer).
6. USARTn_CR1.RE=1, USARTn_CR1.SLME=1 (waiting to receive ID), if using receive interrupt, set USARTn_CR1.RIE=1
7. When the start bit is detected, the receiver receives the received shift register and checks the USARTn_SR.MPB bit
8. If USARTn_SR.MPB = 1, USARTn_CR1.SLME bits are reset automatically and receive data normally. The software compares the received ID with its own ID.
 - If the ID is consistent, the data is normally received, interruptd, and the error is detected, the same as the data received by UART
 - If the ID does not match, the software will USARTn_CR1 again. SLME bit write 1, repeat 8

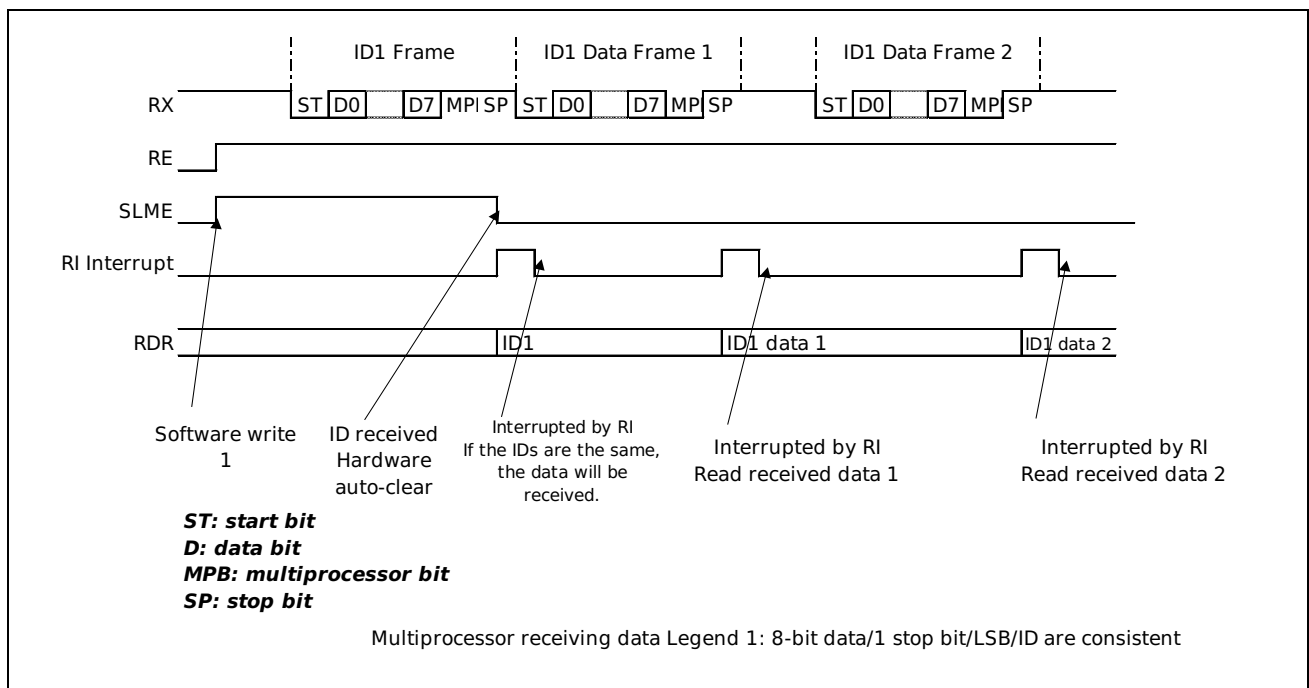


Figure 28-11 Diagram of Multiprocessor Mode Receive Data 1

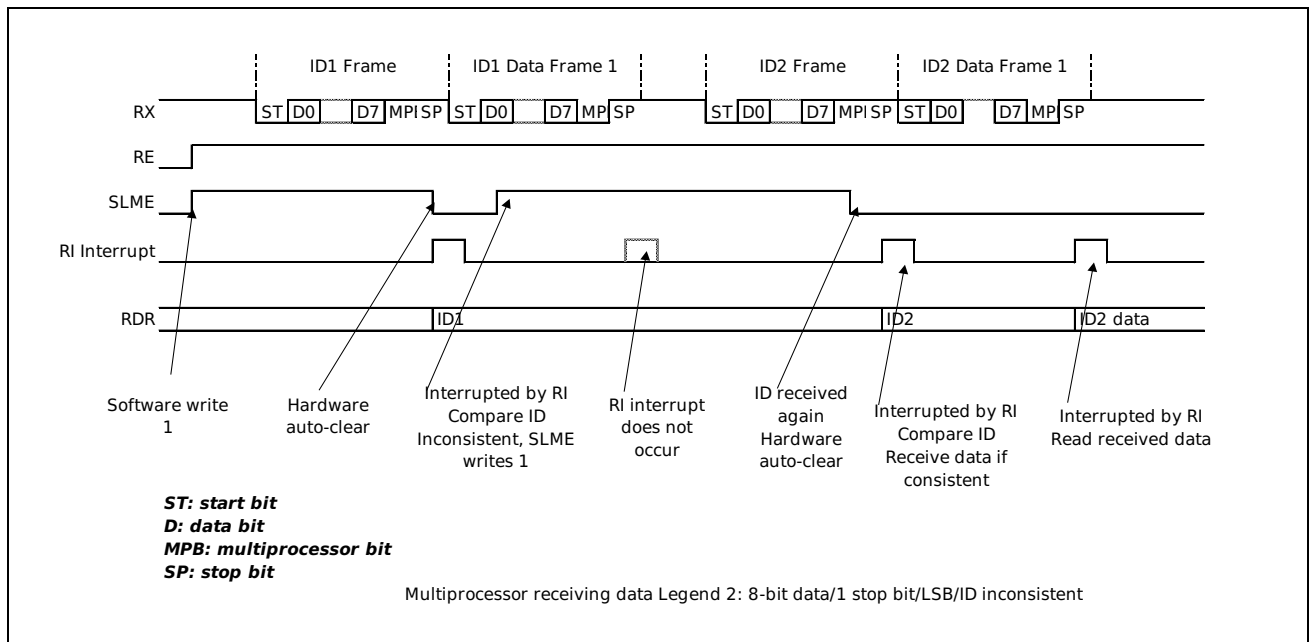


Figure 28-12 Diagram of Multiprocessor Mode Receive Data 2

28.4.2.4 Interrupt and Events

The interrupt mode is the same as the UART mode except for no check error.

Table 28-5 Multi-Processor Mode Interrupt/Event

Functional Name	Enable Bit (interrupt only)	Flag	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

28.4.3 UART-LIN

28.4.3.1 Function Introduction

LIN is short for Local Interconnect Network, which is a low-speed (1~20kbps) serial communication protocol to reduce the cost of automotive network.

28.4.3.2 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.

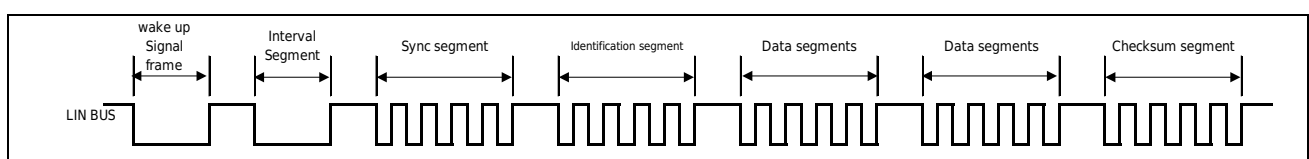


Figure 28-13 LIN Bus Data Behavior

28.4.3.3 LIN send

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.SBKL 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.

28.4.3.4 LIN Receive

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 (130uS at 19.2Kbps), the USARTn_SR.WKUP flag is set. When USARTn_CR2.WKUPE=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 host 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.

Receive Data Setting Steps

1. Set USARTn_CR1 register to reset
2. Set the pin that UART needs to use
3. Through USARTn_CR2.CLKC [1: 0] bit selection timer
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.WKUP=1, wait for the wake-up signal;
6. Detect the wake-up signal, USARTn_CR2.WKUP=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, reset 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. Receive Data
12. Receiving checksum

28.4.3.5 LIN Interrupt and Events

Table 28-6 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/interval 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	No

28.4.4 Smart Card

28.4.4.1 Connection Diagram

Support the smart card communication protocol specified by ISO/IEC 7816-3. underflow - Smart card Pattern mode with connected legend.

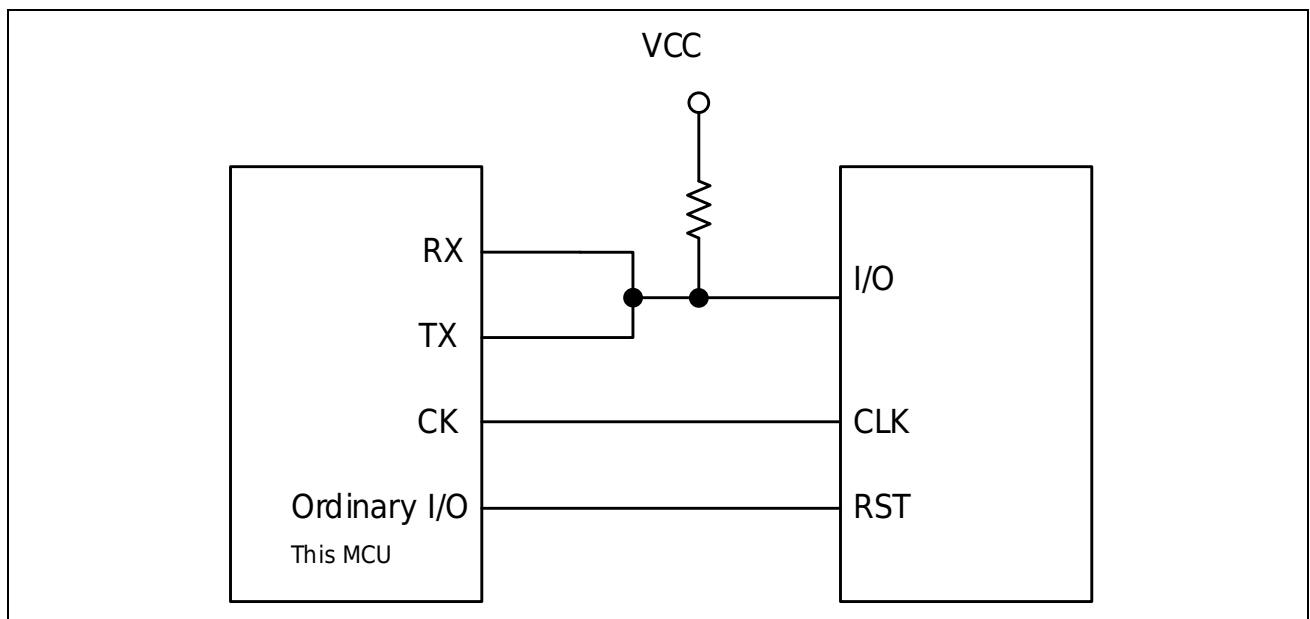


Figure 28-14 Smart card Pattern connected legend

28.4.4.2 Clock

Internal Clock Source

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].

Maximum Baud Rate

For internal timer, the Baud Rate formula generated by the internal Baud Rate Generator is:

B: Baud rate unit: MBps

C: Clock set by USARTn_PR.PSC [1:0] bits (PCLK, PCLK/4, PCLK/16, PCLK/64) Unit: MHz

DIV _ Integer: USARTn _ BRR.DIV _ Integer setting

BCN: USARTn_CR3.BCN register setting value

When C is PCLK , DIV_Integer=0 , BCN=0 , the baud rate is the maximum baud rate PCLK/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.

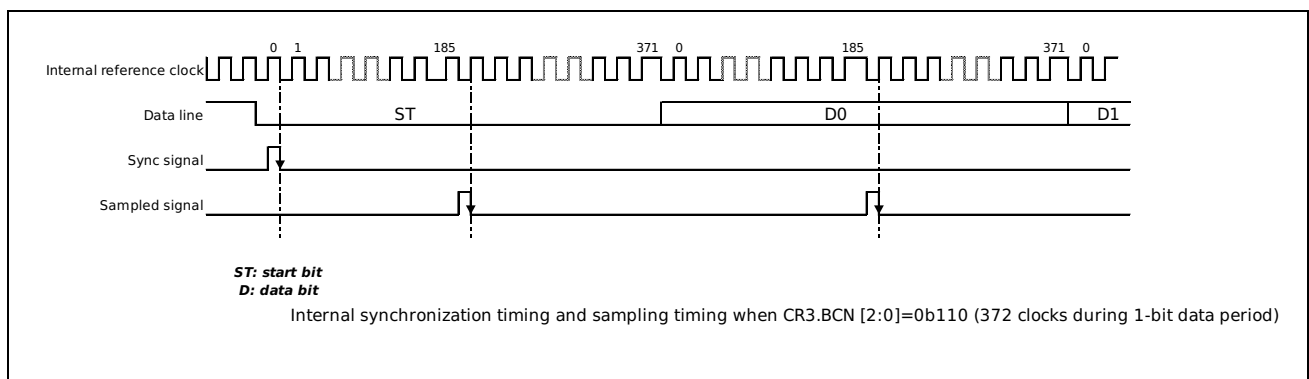


Figure 28-15 Smart Card Pattern Synchronization and Sampling Timing Figure

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

28.4.4.3 Data Format

In smart card mode, a frame of data consists of start bit, data bit and check bit.

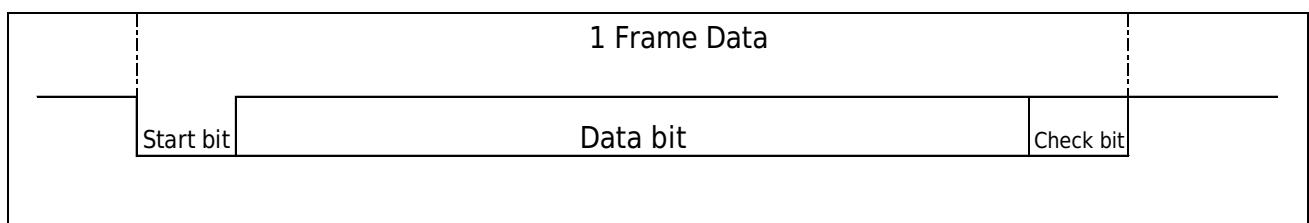


Figure 28-16 Smart Card Mode Data Format

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.

28.4.4.4 Initial Setting Procedure of Smart Card

1. Set USARTn_CR1 register to reset
2. Set the pins to be used
3. Set USARTn_CR1, Status Register and confirm that SR register to reset
4. Set USARTn_CR1, USARTn_CR2, USARTn_CR3 register
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.

28.4.4.5 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

1. 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 retransmitted. The USARTn_SR.FE bit must be cleared before receiving the next frame parity bit.
2. 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.

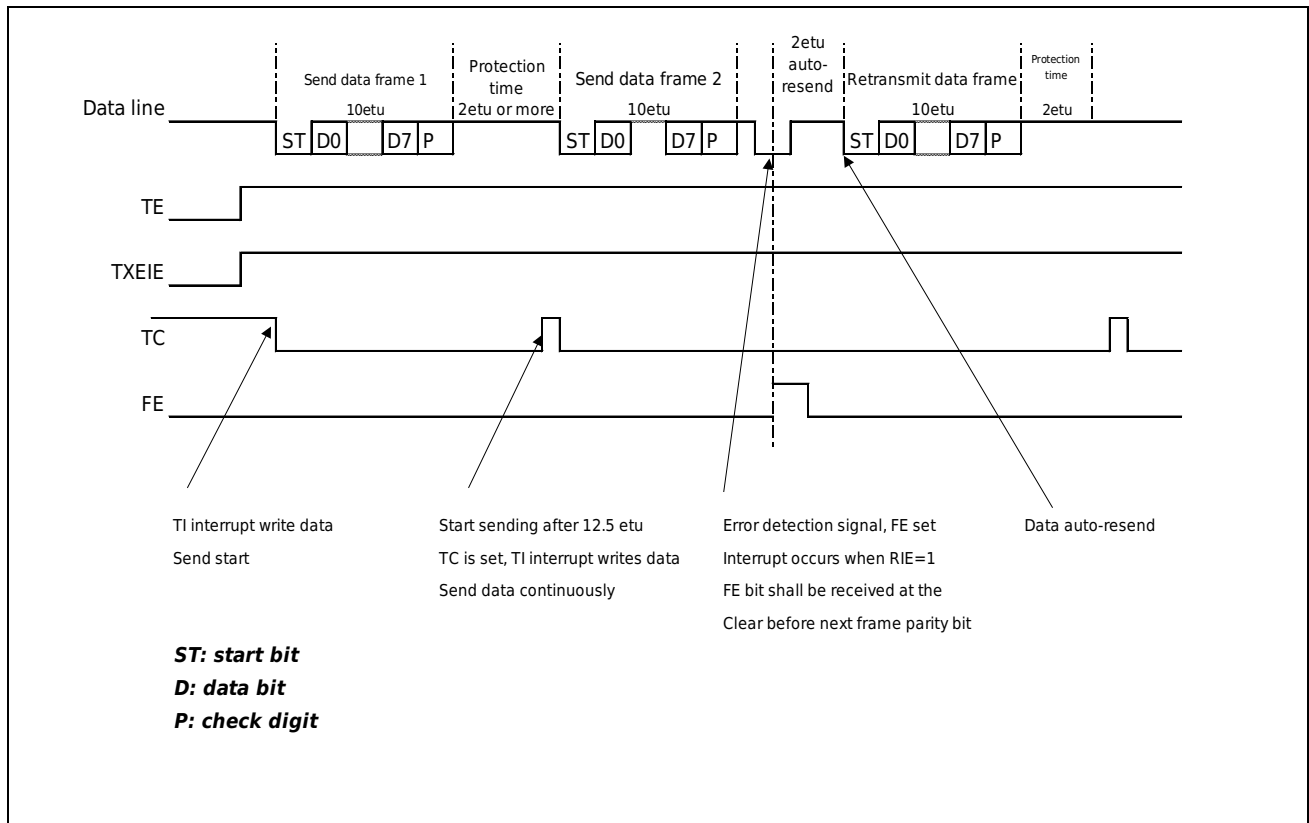


Figure 28-17 Smart Card Pattern Sending Data Legend

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 1etv 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.

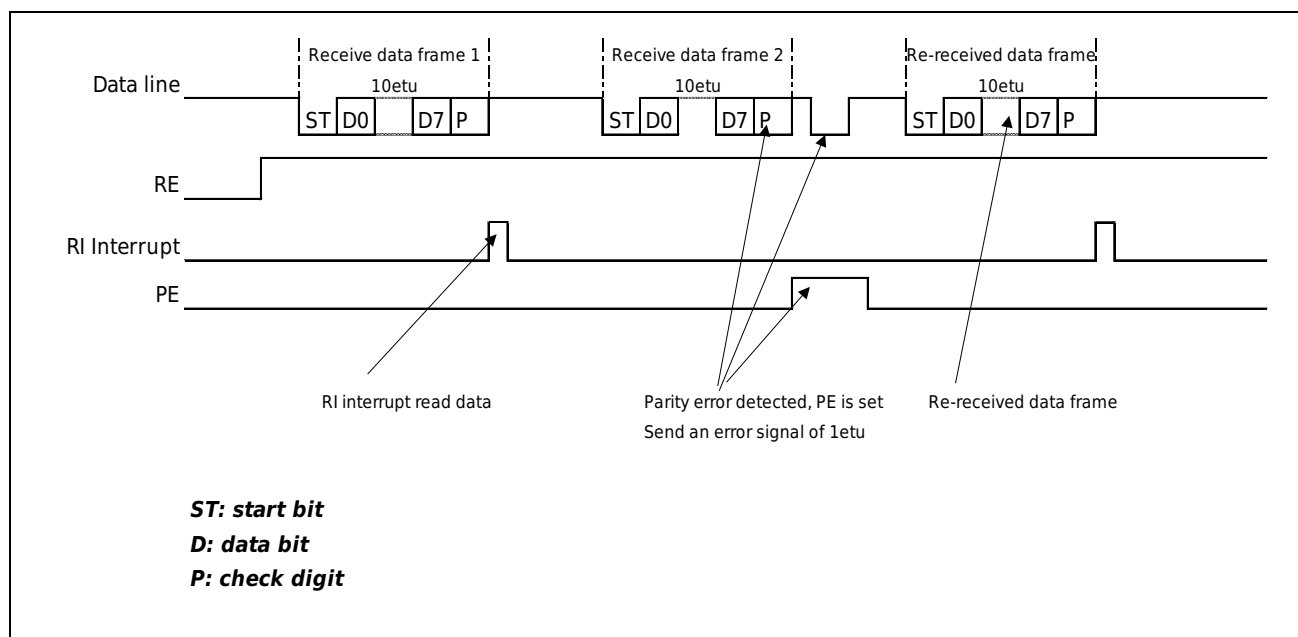


Figure 28-18 Smart Card Pattern Receive Data Legend

28.4.4.6 Interrupt and Events

Table 28-7 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

28.4.5 Clock Synchronization Mode

28.4.5.1 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 clock source for communication.

Internal Clock Source

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 Clock Source

The external timer is the input clock from the USARTn _ CK pin as the communication clock.

Maximum Baud Rate

For internal timer, the Baud Rate formula generated by the internal Baud Rate Generator is:

B: Baud rate unit: MBps

C: USARTn_PR.PSC [1: 0] bit set clock (PCLK,PCLK/4,PCLK/16,PCLK/64) unit: MHz

DIV _ Integer: USARTn_BRR.DIV _ Integer setting

When C is PCLK and DIV _ Integer = 1, the highest baud rate is PCLK/8 (MBps). Note that the DIV _ Integer disable is set to 0 in synchronous mode.

For external timer, the maximum frequency of external input clock is PCLK (MHz)/6, so the maximum baud rate is PCLK/6 (MBps).

It should be noted that in addition to the PCLK-based calculus described above, the Synchronous Mode Maximum Communication Potter also needs to refer to the Maximum Communication Baud Rate specified in the **Electrical Characteristics** chapter.

28.4.5.2 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.

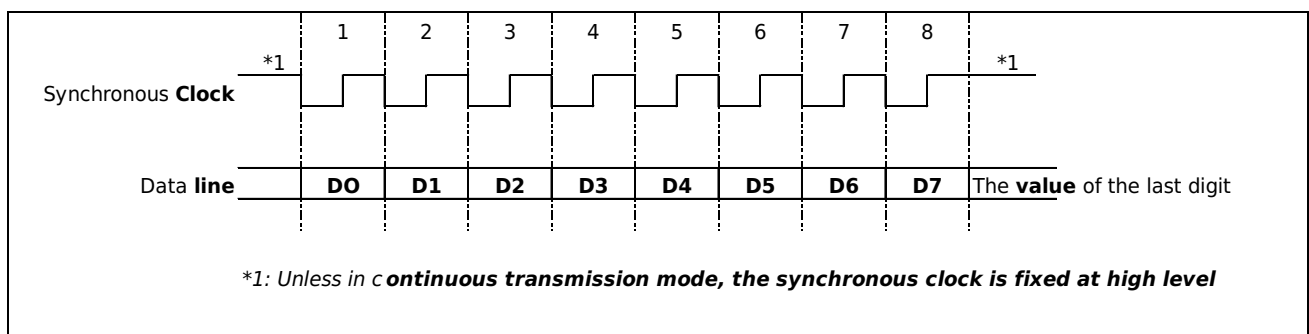


Figure 28-19 Clock Synchronization Mode Data Format

28.4.5.3 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_CTS pin 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 data in USARTn_DR. RDRregister (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.

28.4.5.4 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 register to reset
2. Set the pins to be used
3. Through USARTn_CR2.CLK [1: 0] bit selection timer
4. Set USARTn_CR1, USARTn_CR2, USARTn_CR3 register
5. Set USARTn_PR to select the pre-frequency division value and USARTn_BRR register to set the communication baud rate (not required when timer is external timer).
6. Enable the sender (USARTn_CR1.TE=1). If the send data register null interrupt is required, set USARTn_CR1.TXEIE=1.
7. Wait to send data register empty, write communication data to USARTn_DR.TDR, transfer data to send shift register, send start
(When the CTS feature is active, the USARTn_CTS input is at low level and the data is transmitted to the send shift register, and the send starts)
8. Repeat Step 7 if you need to send data continuously
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.

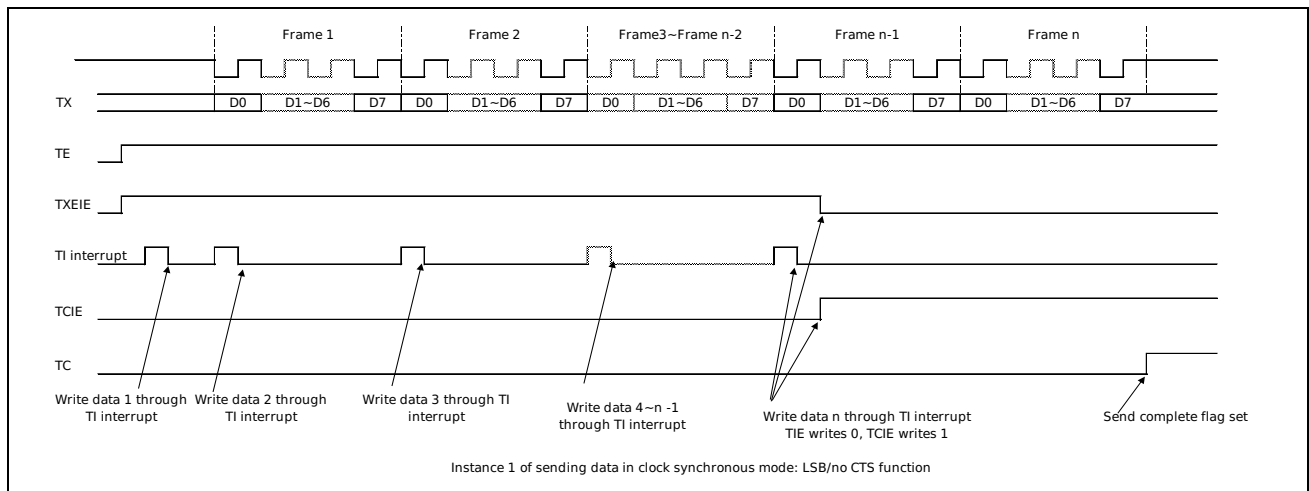


Figure 28-20 Clock Synchronization Mode Sending Data Legend 1

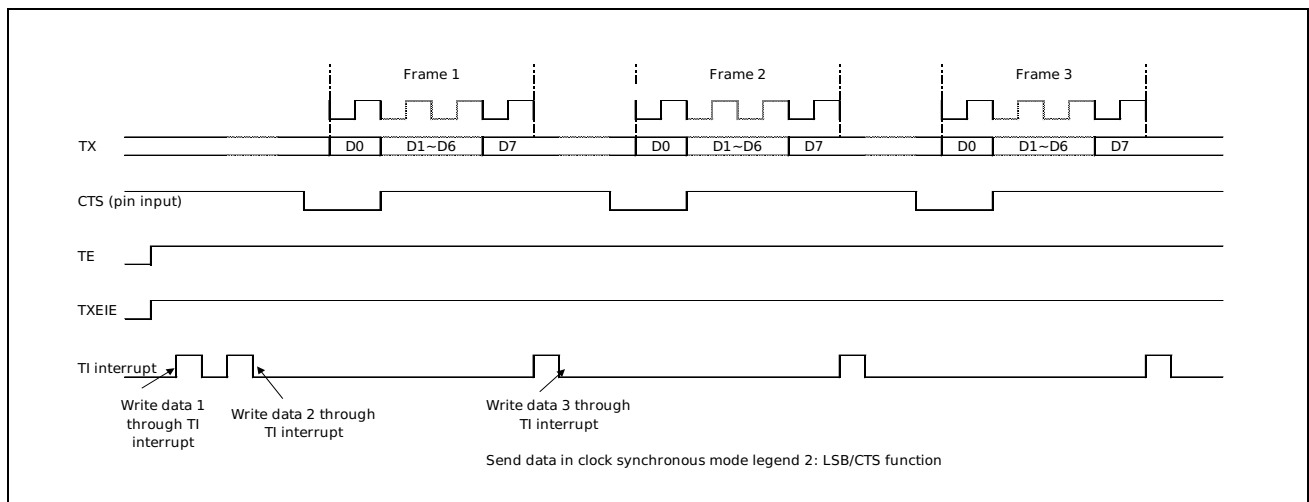


Figure 28-21 Clock Synchronization Mode Sending Data Legend 2

Transmitter Interrupt

The Clock Synchronization Mode Transmitter supports two types of interrupts, sending data register null interrupt TI and sending complete interrupt TC.

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 TC interrupt occurs.

28.4.5.5 Receiver

Receive Data Setting Steps

1. Set USARTn_CR1, USARTn_SR register to reset
2. Set the pins to be used
3. Through USARTn_CR2.CLK [1: 0] bit selection timer
4. Set USARTn_CR1, USARTn_CR2, USARTn_CR3 register

5. Set USARTn _ PR to select the pre-frequency division value and USARTn _ BRR register to set the communication baud rate (not required when timer is external timer).
 6. Enable the receiver (USARTn_CR1. RE=1), if you need to use receive interrupt, set USARTn_CR1. RIE=1
(When using the RTS feature, RE = 1 rear USARTn_RTS output low level)
 7. Synchronize with the input synchronization clock or internally generated synchronization clock to start receiving data, receiving data to the receive shift register.
 - 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_DR.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)

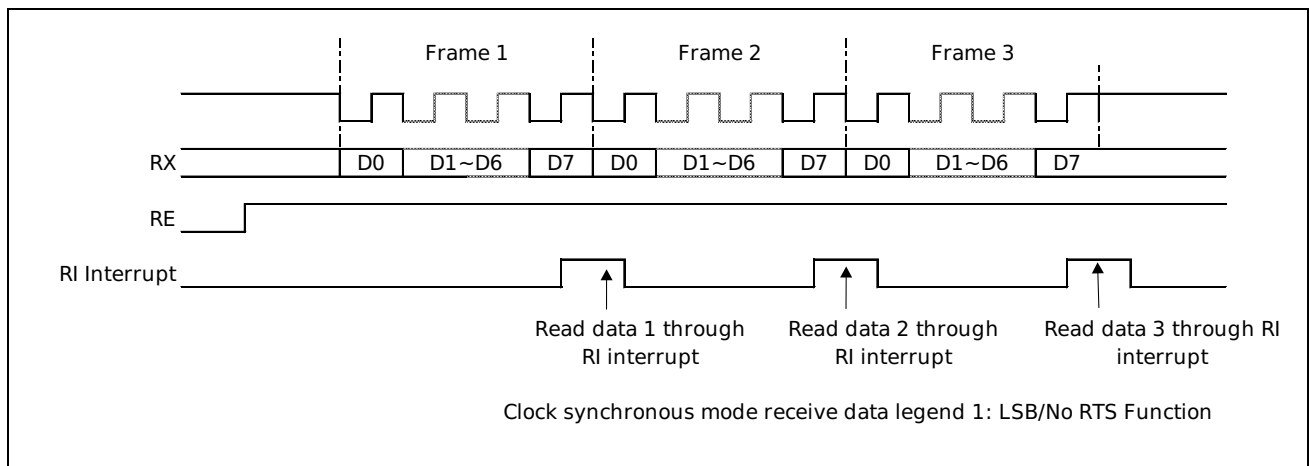


Figure 28-22 Diagram of Clock Synchronization Mode Receive Data 1

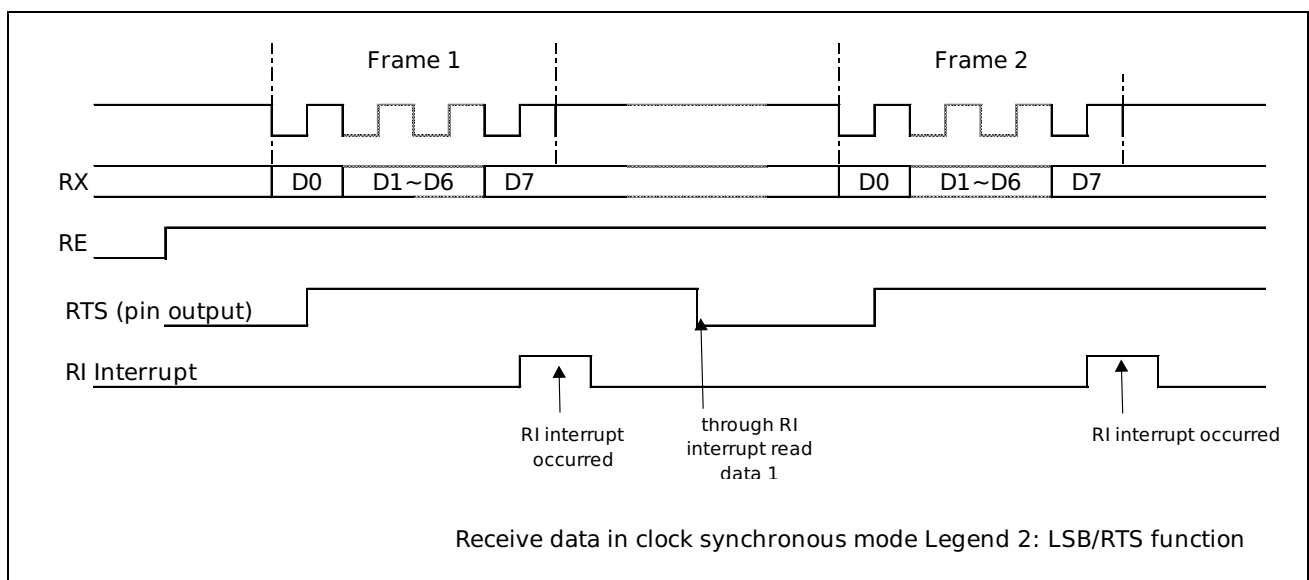


Figure 28-23 Diagram of Clock Synchronization Mode Receive Data 2

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, Rlinterrupt 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, Rlinterrupt 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.

28.4.5.6 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.

28.4.5.7 Clock Synchronization Mode Interrupt and Events

Table 28-8 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	No

28.4.6 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.

28.4.7 Interrupt

The following table gives the USART overall interrupt:

Table 28-9 USART Overall Interrupt List

Interrupt Name	Symbol	Description
Error Interrupt	USART_n_EI n=1~6	UART/Multiprocessor/Smart Card/Clock Synchronous Mode Receive Error Interrupt LIN Error Interrupt
Receiving data register full interrupt	USART_n_RI n=1~6	UART/Multiprocessor/LIN/Smart Card/Clock Synchronous Mode Receive Data Register Full Interrupt LIN Error Interrupt
Transmit data register empty interrupt	USART_n_TI n=1~6	UART/Multiprocessor/LIN/Smart Card/Clock Synchronous Mode send Data Register null Interrupt
Transmit completion interrupt	USART_n_TCI n=1~6	UART/Multiprocessor/LIN/Smart Card/Clock Synchronous Mode send Complete Interrupt (Writing send data to the send data register can clear the interrupt)
Transmit completion interrupt	USART_n_TENDI n=1~6	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/interval detection interrupt	USART_n_BRKWKPI n=3/6	LIN wake-up signal/break detection interrupt
UART receive TIMEOUT interrupt	USART_n_RTO n=1/2/4/5	UART receive TIMEOUT interrupt
RX stop mode wake-up interrupt	USART_1_WUPI	USART_1 RX line wake-up stop mode interrupt

28.5 Register Description

This chapter will describe in detail the USART module control status and other related registers. 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.

The base address is as follows:

USART_1_BASE_ADDR:0x4001CC00

USART_2_BASE_ADDR:0x4001D000

USART_3_BASE_ADDR:0x4001D400

USART_4_BASE_ADDR:0x40020C00

USART_5_BASE_ADDR:0x40021000

USART_6_BASE_ADDR:0x40021400

Table 28-10 List of USART registers

Register Name	Offset Address	Reset Value
Status register (USART _ SR)	0x00	0x0000 00C0
Transmit Data Register(USART_TDR)	0x04	0x01FF
Receive Data Register(USART_RDR)	0x06	0x0000
Baud rate register (USART _ BRR)	0x08	0x0000 FFFF
Control register 1 (USART_CR1)	0x0C	0x8000 0000
Control register2 (USART _ CR2)	0x10	0x0000 0600
Control register3 (USART _ CR3)	0x14	0x0000 00000
Pre-divider register (USART _ PR)	0x18	0x0000 0000
LIN Baud Rate Measurement Count Register (USART_LBMC)	0x1C	0x0000 0000

28.5.1 Status Register (USART _ SR)

USART Status Register

Offset address: 0x00

Reset value: 0x000000C0

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	MPB
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	TEND	LBD	WKUP	RTOF	TXE	TC	RXNE	BE	ORE	-	FE	PE

Bit	Flag	Bit Name	Function	Read and write
b31~b17	Reserved	-	Read as "0", write as "0"	R/W
b16	MPB	Multiprocessor position	Multi-processor bit flag 0: Current received data is communication data 1: Current received data is ID Note: MPB bits are only valid in multiprocessor mode	R
b15~b12	Reserved	-	Read as "0", write as "0"	R/W
b11	TEND	Transmit completion flag	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 send 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 Mode 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 (valid only for USART_3/USART_6) 0: No interval detected 1: Interval segment detected LBD setting condition When CR2.WKUP=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 position	R
b9	WKUP	LIN Wakeup Signal Flag	LIN Wakeup signal detection flag (valid only for USART_3/USART_6) 0: No wakeup signal detected 1: Wakeup signal detected WKUP setting condition When CR2.WKUP=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 position After detecting that the WKUP signal is 1, it indicates that the LIN node has exited from sleep mode. After setting register CR2.WKUP=0, subsequent operations such as interval field detection are performed.	R
b8	RTOF	UART receive TIMEOUT Flag bit	UART receive TIMEOUT flag (only valid for USART_1/USART_2/ USART_4/ USART_5) 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	R

			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 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	Transmit completion flag	Send completion flag bit 0: Send data 1: Completion of data transmission UART mode, Clock synchronization mode TC setting condition • TE = 0 Send Disabled • When the last bit of a frame of data is sent, the value of the send data register is not updated TC zeroing condition • When TE = 1, write the sending data to the sending data register Smart Card Mode TC setting condition • TE = 0 Send Disabled • 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 is changed from 0 to 1, TC is kept to 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_3/USART_6) 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 position 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 ORE 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	Reserved	-	Read as "0", write as "0"	R/W
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 Rlinterrupt will not occur and cannot continue to receive data after FE = 1 Smart Card Mode Fe setting condition • Sampled low error signal flag • FE zeroing condition	R

			• Write 1 to the reset register CLR.CFE bit	
			Note: In smart card mode, RE=0 does not reset the FE bit	
			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 position	
			Data received at PE = 1 will be retained but Rlinterrupt will not occur and cannot continue to receive data after PE = 1	
b0	PE	Received data verification		R

28.5.2 Transmit Data Register (USART_TDR)

USART Transmit Data Register

Offset address: 0x04

Reset value: 0x01FF

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	MPID	TDR [8:0]								

Bit	Flag	Bit Name	Function	Read and write
b15~b10	Reserved	-	Read as "0", write as "0"	R/W
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	R/W
b8~b0	TDR [8:0]	Transmit data	Transmit data Note: The highest TDR [8] is only valid in UART mode and the data length is set to 9 bits	R/W

28.5.3 Receive Data Register (USART_RDR)

USART Receive Data Register

Offset address: 0x06

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	RDR [8:0]								

Bit	Flag	Bit Name	Function	Read and write
b15~b9	Reserved	-	Read as "0", write as "0"	R/W
b8~b0	RDR [8:0]	Receiving data register	Receiving data register Note: The highest RDR [8] is only valid in UART mode and the data length is set to 9 bits	R

28.5.4 Baud Rate Register (USART _ BRR)

USART Bit Rate Register

Offset address: 0x08

Reset value: 0x0000FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DIV_Integer [7:0]								-	DIV_Fraction [6:0]						

Bit	Flag	Bit Name	Function	Read and write
b31~b16	Reserved	-	Read as "0", write as "0"	R/W
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)	R/W
b7	Reserved	-	When writing, you need to write the reset value	R/W
b6~b0	DIV_Fraction [6:0]	Fractional division register	Fractional division register Note: [6: 0] only be set at TE = 0 & RE = 0 (send/receive disabled), and Only Set value valid at FBME = 1	R/W

Table 28-11 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.PSC [1:0] bits: MHz

BCN: CR3.BCN register setting value

Table 28-12 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.PSC [1:0] bits: MHz

BCN: CR3.BCN register setting value

28.5.5 Control Register 1 (USART_CR1)

USART Control Register 1

Offset Address: 0x0C

Reset Value: 0x80000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SBS	NFE	FBME	ML	-	-	CTE ND	MS	CLB D	CWK UP	CBE	CRT OF	COR E	-	CFE	CPE
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
OVER 8	-	-	M	-	PCE	PS	TEN DIE	TXE E	TCIE	RIE	SLME	TE	RE	RTO IE	RTOE

Bit	Flag	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: In non-UART mode, the SBS bit must remain reset SBS bits can only be set at TE = 0 & RE = 0 (Send/Receive Disabled)	R/W
b30	NFE	Digital filter enable bit	Digital filter enable bit 0: Disable digital filtering 1: Enable data filtering Note: NFE bits must remain reset in non-UART mode NFE bits can only be set when TE = 0 & RE = 0 (Send/Receive Disabled)	R/W
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) =1 is not supported when LINEN =1	R/W
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)	R/W
b27~b26	Reserved	-	Read as "0", write as "0"	R/W
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	R/W
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	R/W
b23	CLBD	LBD reset bit	LBD reset bit (valid only for USART_3/USART_6) 0: Not zero LBD flag 1: Clear LBD flag Note: CLBD bit write 1 resets LBD flag and returns 0 when read	R/W
b22	CWKUP	WKUP reset bit	WKUP reset bit (valid only for USART_3/USART_6) 0: Not zero WKUP flag 1: Clear WKUP flag Note: CWKUP bit write 1 resets WKUP flag and returns 0 when read	R/W
b21	CBE	BE reset bit	BE reset bit (only valid for USART_3/USART_6) 0: Not zero BE flag 1: Clear BE flag Note: CBE bit write 1 resets BE flag and returns 0 when read	R/W
b20	CRTOF	RTOF reset bit	RTOF reset bit (only valid for USART_1/USART_2/ USART_4/USART_5) 0: Not zero RTOF flag 1: Clear RTOF flag Note: CRTOF bit write 1 resets RTOF flag and returns 0 when read	R/W
b19	CORE	ORE flag zeroing	ORE flag zeroing 0: Not zeroed ORE logo 1: Clear ORE logo Note: The CORE bit write 1 clears the ORE flag and returns 0 when read	R/W
b18	Reserved	-	Read as "0", write as "0"	R/W

b17	CFE	FE flag zeroing bit	FE flag zeroing bit 0: Unzeroed FE logo 1: Reset FE logo Note: CFE bit write 1 clears FE flag and returns 0 when read	R/W
b16	CPE	PE flag zeroing bit	PE flag zeroing bit 0: Not Zero PE Mark 1: Clear PE logo Note: CPE bit write 1 clears FE flag and returns 0 when read	R/W
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: OVER8 bits must remain reset in non-UART mode OVER8 bits can only be set at TE = 0 & RE = 0 (Send/Receive Disabled)	R/W
b14~b13	Reserved	-	Read as "0", write as "0"	R/W
b12	M	Data length setting	In UART mode, transmit/receive data length setting 0: 8 bits 1: 9 bits Note: M bits must remain reset in non-UART mode M bits can only be set when TE = 0 & RE = 0 (Send/Receive Disabled)	R/W
b11	Reserved	-	Read as "0", write as "0"	R/W
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 when TE=0&RE=0 (Send/Receive Disabled)	R/W
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	R/W
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	R/W
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	R/W
b6	TCIE	Send complete interrupt	Send complete interrupt 0: TCInterrupt request is invalid, TCInterrupt does not occur 1: TCInterrupt request enabled, TCInterrupt occurs Note: Reading in non-smart card mode valid, reading in smart card mode is 0	R/W
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	R/W
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.	R/W
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).	R/W
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	R/W
b1	RTOIE	UART TIMEOUT Interrupt enable bit	UART TIMEOUT interrupt enable bit (valid only for USART_1/USART_2/USART_4/USART_5) 0: UART TIMEOUT interrupt request is invalid, RTOI interrupt does not occur 1: UART TIMEOUT interrupt request is valid, RTOI interrupt occurs Non-smart card mode / non-clock synchronization mode reading is valid, smart card mode / clock synchronization mode reading is 0	R/W

b0	RTOE	UART TIMEOUT function enable bit	UART TIMEOUT function enable bit (only valid for USART_1/USART_2/ USART_4/USART_5) 0: UART TIMEOUT function disable 1: UART TIMEOUT function enable Non-smart card mode / non-clock synchronization mode reading is valid, smart card mode / clock synchronization mode reading is 0	R/W

28.5.6 Control Register 2 (USART_CR2)

USART Control Register 2

Offset address: 0x10

Reset value: 0x00000600

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	SBKM	SBK
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	LINE N	STOP	CLKC [1:0]	-	-	WKU PE	SBKL [1:0]	LBDL	LBDI E	BEE	BEIE	WKU PIE	MPE		

Bit	Flag	Bit Name	Function	Read and write
b31~b18	Reserved	-	Read as "0", write as "0"	R/W
b17	SBKM	Interval Segment Transmission Mode	Interval segment transmission mode selection (only valid for USART_3/USART_6) 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	R/W
b16	SBK	Interval Segment Transmission Enable Bit	Interval segment transmission enable bit (valid only for USART_3/USART_6) 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	R/W
b15	Reserved	-	Read as "0", write as "0"	R/W
b14	LINEN	LIN function enabled	In UART mode, LIN function enable bit (valid only for USART_3/USART_6) 0: Disable LIN function 1: Enable LIN function Note: LINEN bits can only be set at TE=0&RE=0 (Send/Receive Disabled)	R/W
b13	STOP	Stop positioning	Stop bit length setting in UART mode 0: 1 stop bit 1: 2 stop bit Note: The STOP bit must remain reset in non-UART mode The STOP bit can only be set at TE = 0 & RE = 0 (Send/Receive Disabled)	R/W
b12~b11	CLKC [1:0]	Clock control bit	UART mode 00b: 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 01b: Clock generated by timer for internal baud rate generator, clock output to USARTn_CK pin, output clock frequency and baud rate same 10b or 11b: Timer is external input clock, input clock frequency is 16 times (OVER8 = 0) or 8 times (OVER8 = 1) higher than baud rate Clock synchronization mode 00b or 01b: Clock generated by timer for internal baud rate generator, output to USARTn_CK pin 10b or 11b: Timer is an external input clock with the same frequency and baud rate Smart card pattern 00b: 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 01b: Clock generated by timer for internal baud rate generator, Clock output to USARTn_CK pin 10b or 11b: Disable setting Note: CLKC [1: 0] bits can only be set at TE = 0 & RE = 0 (Send/Receive Disabled)	R/W
b10~b9	Reserved	-	The reset value 11b needs to be maintained, and 11b is written when writing	R/W

b8	WKUPE	LIN wake-up signal detection enable bit	LIN wake-up signal detection enable bit (valid only for USART_3/USART_6) 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	R/W
b7~b6	SBKL [1:0]	LIN interval low level width setting bit	LIN interval low level width setting bit (only valid for USART_3/USART_6) 00b: 10 bits 01b: 11 bits 10b: 13 bits 11b: 14 bits Note: In non- UART-LIN mode, the SBKL bit must keep the reset value SBKL bits can only be set when TE=0&RE=0 (Send/Receive Disabled)	R/W
b5	LBDL	LIN interval low level width detection threshold setting bit	LIN interval low level width detection threshold setting bit (only valid for USART_3/USART_6) 0: ≥10 digits 1: ≥11 digits Note: In non- UART-LIN mode, the LBDL bit must keep the reset value LBDL bits can only be set when TE=0&RE=0 (Send/Receive Disabled)	R/W
b4	LBDIE	LIN interval detection interrupt enable bit	LIN break segment detection interrupt enable bit (valid only for USART_3/USART_6) 0: LIN break segment detection interrupt request invalid and no interrupt occurs 1: LIN interval detection interrupt request valid and an interrupt occurs Note: In non- UART-LIN mode, the LBDIE bit must keep the reset value	R/W
b3	BEE	LIN bus error detection function enable bit	LIN bus error detection function enable bit (only valid for USART_3/USART_6) 0: Disable LIN bus error detection function 1: Enable LIN bus error detection function Note: In non- UART-LIN mode, the BEE bit must keep the reset value BEE bits can only be set when TE=0&RE=0 (Send/Receive Disabled)	R/W
b2	BEIE	LIN bus error interrupt enable bit	LIN bus error interrupt enable bit (valid only for USART_3/USART_6) 0: LIN bus error interrupt request invalid and no interrupt occurs 1: LIN bus error interrupt request valid and an interrupt occurs Note: In non- UART-LIN mode, the LBDIE bit must keep the reset value	R/W
b1	WKUPIE	LIN wake-up signal detection Interrupt Enable bit	LIN wake-up signal detection interrupt enable bit (valid only for USART_3/USART_6) 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 keep the reset value	R/W
b0	MPE	Multi-processor function	In UART mode, the multiprocessor function enables the bit 0: Prohibited 1: Enable Note: MPE bits must remain reset in non-UART mode MP bits can only be set when TE = 0 & RE = 0 (Send/Receive Disabled)	R/W

28.5.7 Control register 3 (USART_CR3)

USART Control Register 3

Offset address: 0x14

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	BCN [2:0]		-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b15	b14
-	-	-	-	-	-	CTSE	RTSE	-	-	SCE N	LOOP P	HDSEL	-	-	-

Bit	Flag	Bit Name	Function	Read and write
b31~b24	Reserved	-	Read as "0", write as "0"	R/W
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 000b 32 001b 64 010b Setting prohibited 011b 128 100b Setting prohibited 101b 256 110b 372 111b 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_3/USART_6 do not support this function, and the output is 0	R/W
b20~b10	Reserved	-	Read as "0", write as "0"	R/W
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)	R/W
b8	RTSE	RTS function	RTS function 0: RTS function Invalid 1: RTS function valid Note: RTSE bits can only be set at TE = 0 & RE = 0 (Send/Receive Disabled)	R/W
b7~b6	Reserved	-	Read as "0", write as "0"	R/W
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 remain reset in non- Smart Card mode SCEN bits can only be set when TE = 0 & RE = 0 (Send/Receive Disabled) USART_3/USART_6 do not support this function, and the output is 0	R/W
b4	LOOP	LIN loopback mode enable bit	LIN loopback mode enable bit 0: Normal mode 1: Loopback mode Note: LOOP bits can only be set at TE = 0 & RE = 0 (Send/Receive Disabled)	R/W
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)	R/W
b2~b0	Reserved	-	Read as "0", write as "0"	R/W

28.5.8 Pre-Divider Register (USART _ PR)

USART prescaler register

Offset address: 0x18

Reset value: 0x00000000

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

Bit	Flag	Bit Name	Function	Read and write
b31~b4	Reserved	-	Read as "0", write as "0"	R/W
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_3/USART_6) 00: PCLK 01: PCLK/2 10: PCLK/4 11: PCLK/8 Note: LBMPSC [1: 0] bits can only be set at TE = 0 & RE = 0 (Send/Receive Disabled)	R/W
b1~b0	PSC [1:0]	Pre-divider value	Pre-divider divider selection bit when internal timer 00: PCLK 01: PCLK/4 10: PCLK/16 11: PCLK/64 Note: PSC [1: 0] bits can only be set at TE = 0 & RE = 0 (Send/Receive Disabled)	R/W

28.5.9 LIN Baud Rate Measurement Count Register (USART_LBMC)

USART LIN Baudrate Measuring Counter

Offset address: 0x1C

Reset value: 0x00000000

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

Bit	Flag	Bit Name	Function	Read and write
b31~b16	Reserved	-	Read as "0", write as "0"	R/W
b15~b0	LBMC [15:0]	LIN baud rate measurement counter	LIN baud rate measurement result counter (only valid for USART_3/USART_6) 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 host 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

28.5.10 USART1 Filter Control Register (USART1_NFC)

USART1 Noise Filting Control Counter

Address: 0x4005541C

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	-	USART1_NFE	USART1_NFS [1:0]	

Bit	Flag	Bit Name	Function	Read and write
b31~b3	Reserved	-	Read as "0", write as "0"	R/W
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].	R/W
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. 00b: Filtering width level 1 01b : Filtering width level 2 10b : Filtering width level 3 11b : Filtering width level 4 For the specific values of each gear, please refer to the " Electrical Characteristics USART1 RX Filter Characteristics in STOP Mode " section.	R/W

28.6 Precautions

28.6.1 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.

28.6.2 Clock Synchronization Mode Considerations

- 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.
- When data is sent consecutively, the next frame of data needs to be updated before the last bit of the current frame is sent.

28.6.3 Other Considerations

- 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

29 Integrated Circuit Bus (I2C)

29.1 Introduction

I2C (integrated circuit bus) used as an interface between the microcontroller and the I2C serial bus. Provide multi-host mode function, which can control the protocol and arbitration of all I2C buses. Standard mode and fast mode are supported. SMBus is also supported.

I2C main features:

- I2C bus mode and SMBUS bus mode are optional. Host 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 1MSPS
- 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. 7-bit address format and 10-bit address format can be set simultaneously. Broadcast call address, SMBus host address, SMBus device default address, SMBus alarm address can be detected
- 2-stage transmit FIFO and 2-stage receive FIFO
- 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 unanimately interrupt.

29.2 I2C System Block Diagram

29.2.1 System Block Diagram

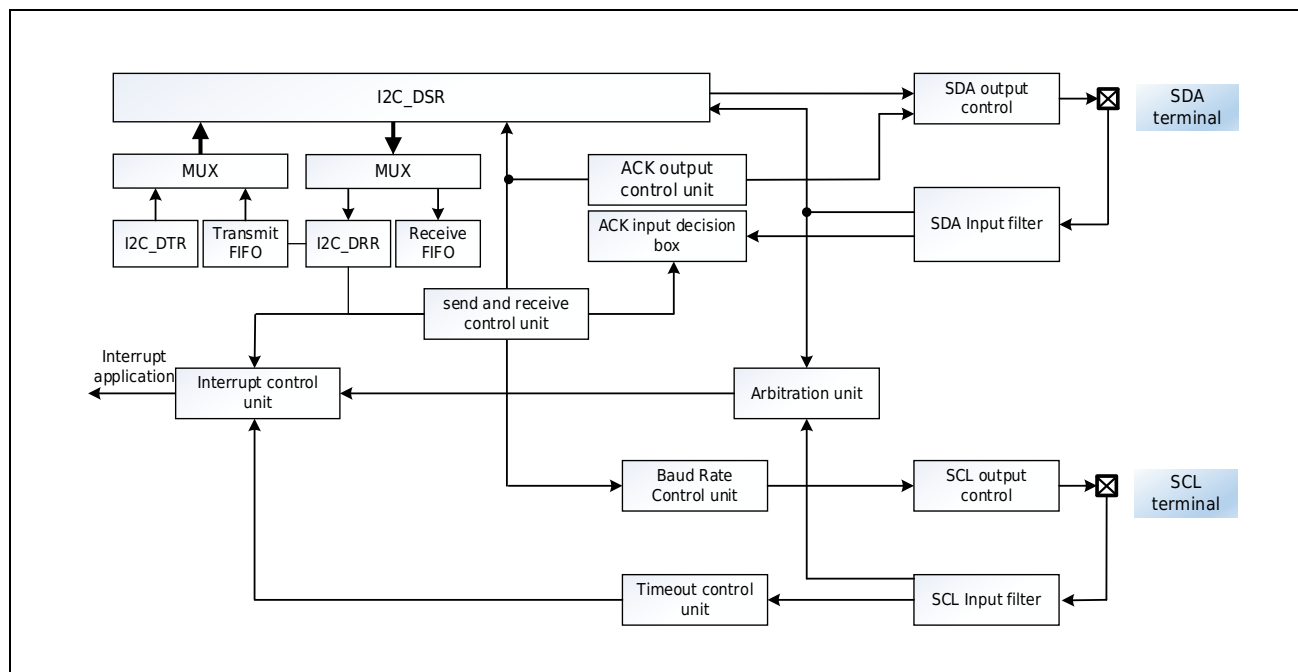


Figure 29-1 I2C System Block Diagram

29.2.2 Structural Diagram

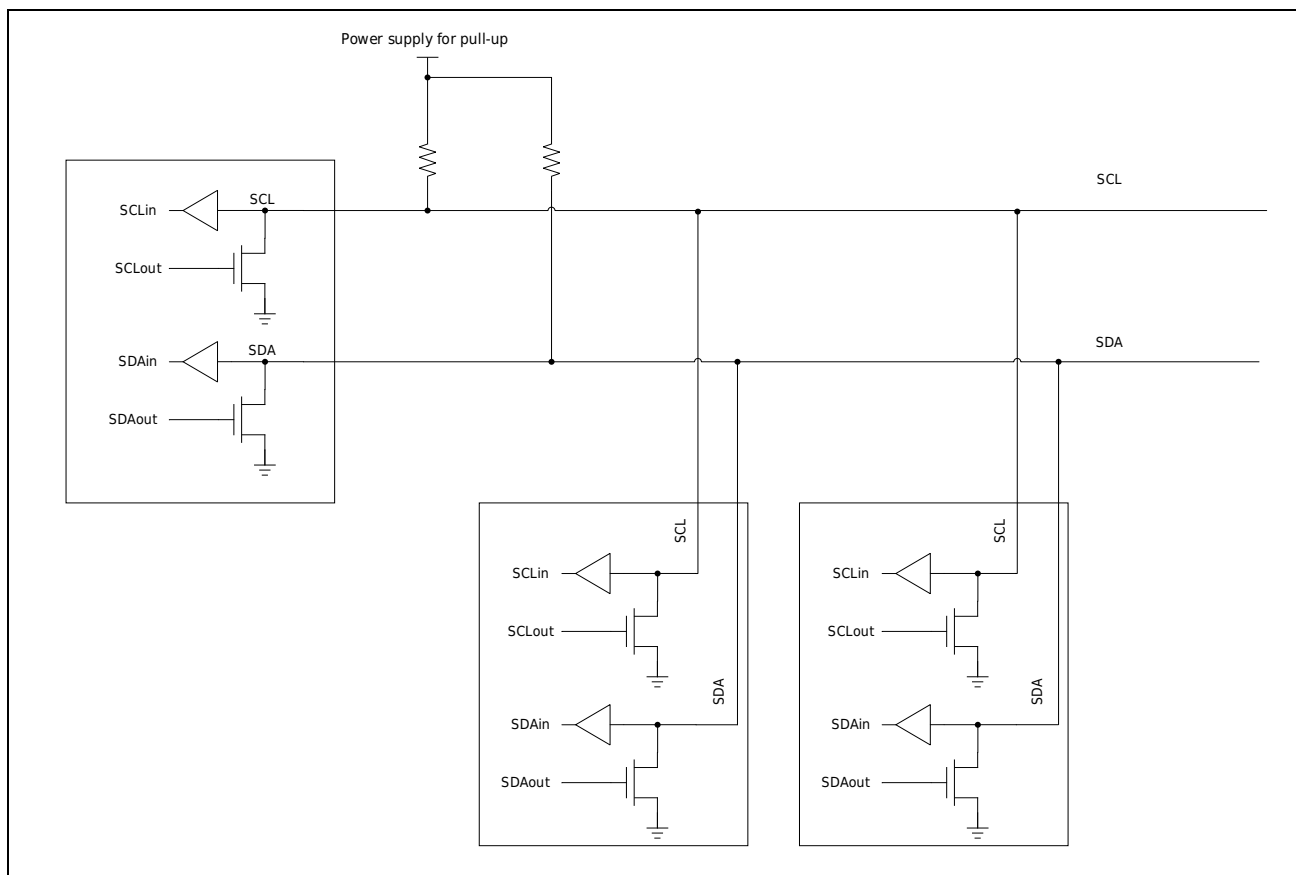


Figure 29-2 Instance of I2C Bus Structure

Table 29-1 Input/Output Pin

Pin Name	Input/Output	Function
SCL	Input/Output	Clock input / output pin
SDA	Input/Output	Data input / output pins

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 [General IO (GPIO)] section.

29.3 Action Description

This section provides a description of the functions of the I2C module.

29.3.1 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 general, 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.

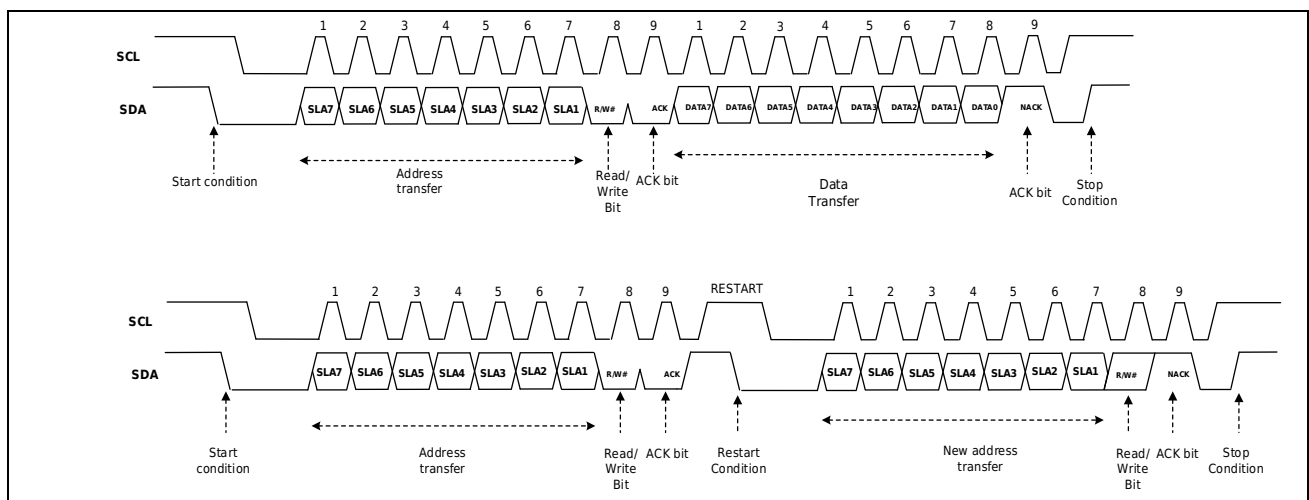


Figure 29-3 The Following Figure is a Sequence Diagram of the I2C bus.

29.3.1.1 Start Condition

When the host 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 position is set to "1" in the I2C _ SR.BUSY flag of "0" (bus is idle), the launch condition is issued. If a start condition is detected, the I2C _ SR.BUSY flag and I2C _ SR.STARTF flag are automatically set to "1" and the START position is automatically set to "0". At this point, if the status of the SDA signal sent in the START bit "1" is the same as that of the SDA line, and the start condition is detected, it is considered that the start condition is correctly issued by the START bit, and the I2C _ SR.MSL bit and I2C _ SR.TRA bit are automatically set to "1" and then changed to the master send mode. In addition, I _ 2C _ SR.The EMPTYF automatically changes to '1' because the TRA bit is '1'. Next write the slave address to the I2C _ DTRregister and send the address.

29.3.1.2 Address Transfer

The frame after the start condition or restart condition is an address frame that specifies the object address for host 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.

- The 7-bit addressing pattern is shown in the following figure [7-bit address format]
Host send mode, host send address frame 8th bit is 0
Host receive mode, host send address frame 8th bit is 1
- The 10-bit addressing pattern is shown in the following figure [10-bit address format]
Host send mode, host first frame send header sequence (11110XX0, where XX represents the top two digits of the 10-bit address), then the second frame send the lower octet slave address.
Host accept mode, host first frame send header sequence(11110XX0, 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 (11110XX1, where XX represents the top two digits of the 10-bit address).

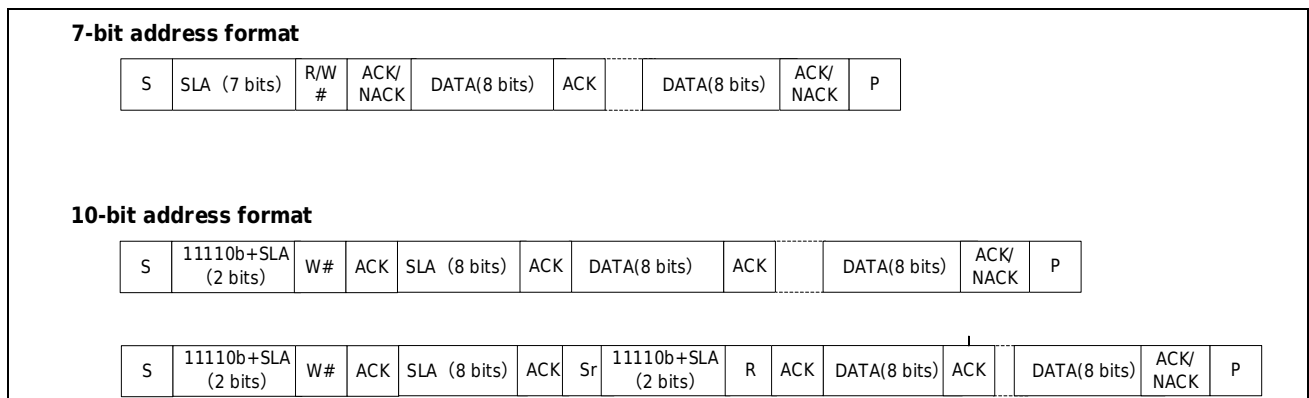


Figure 29-4 I2C Data Format of I2C Bus

S : indicates the start condition.

SLA : indicates the slave address.

R/W# : indicates the direction of sending and receiving. Send data from slave to host when R/W # is "1"; When R/W # is "0", the data is sent from the host to the slave.

Sr : indicates a restart condition.

DATA : indicates the data sent and received.

P : indicates a stop condition.

29.3.1.3 Data Transfer

After address matching is consistent, the host on the bus transmits data frame by frame according to the direction defined by R/W.

The data transmitted after all address frames are treated as data frames. Even a 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 period. The 9th clock after the data frame is an answer bit and is a handshake signal transmitted to the receiver.

If data is received from the slave computer on the bus and does not respond to the host in the ninth clock period, the slave computer must send NACK. If the host receives data on the bus, NACK is sent in the 9th period, NACK is received from the slave, and data is stopped from the slave.

Whether the host or the slave sends NACK, the data transfer interrupts. The host can do any of the following:

- Transmit stop conditional release bus
- Send the restart condition to start a new communication.

Host Sends Data

In host transmit mode, the host outputs SCL clock and transmit data, receives data from the host and returns an answer. The sequence of host sending data is shown in the following figure.

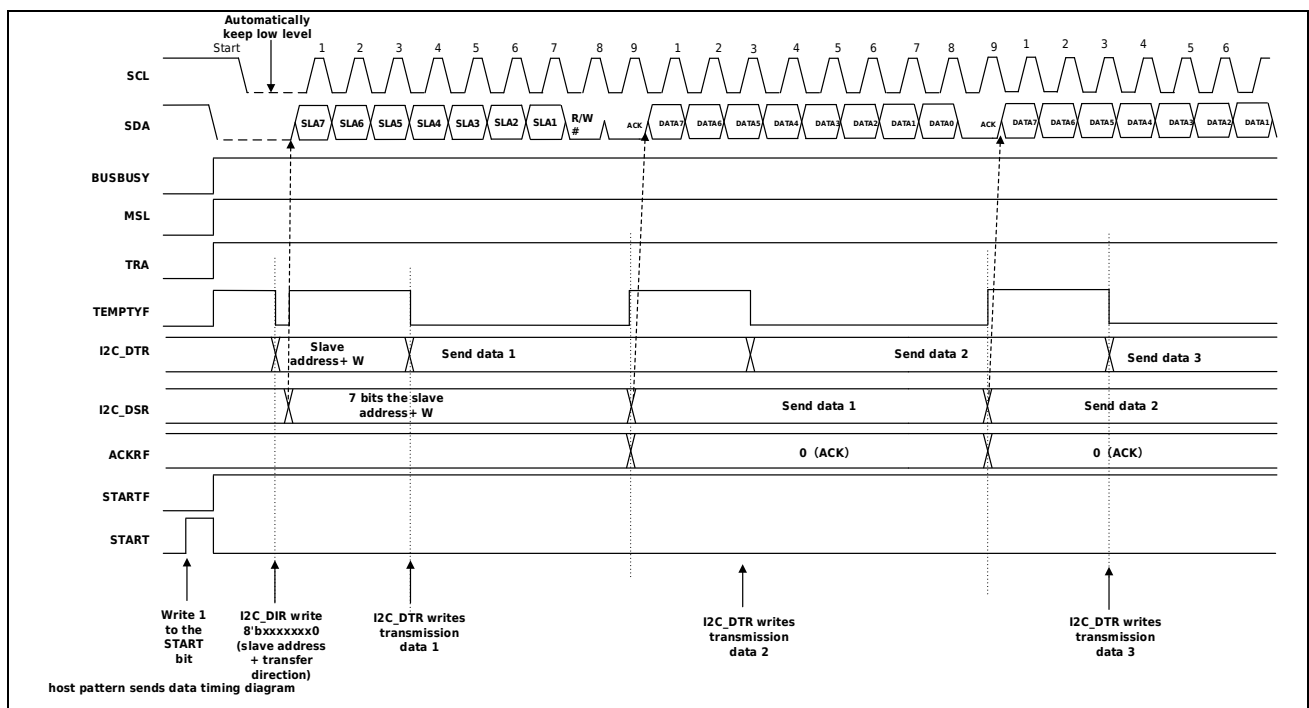


Figure 29-5 Sequence Diagram of host Sending Data in 7-bit Address Format (Instance)

Host Receiving Data

In host receiving mode, the host outputs an SCL clock, receives slave data, and returns an answer. The running sequence of receiving data by the host is shown in the following figure.

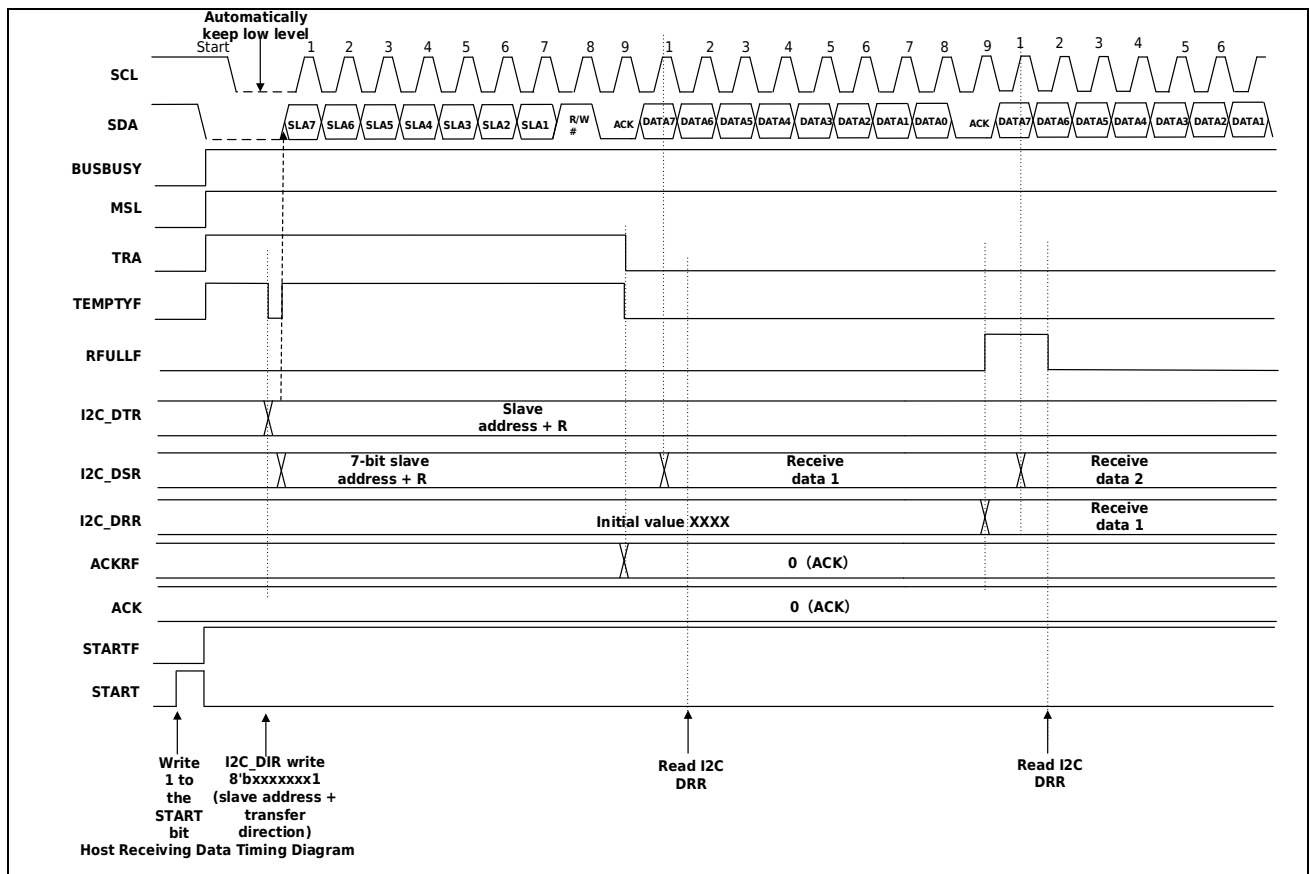


Figure 29-6 Sequence Diagram of Received Data for Host in 7-bit Address Format (Instance)

Sending Data from A slave Computer

In Slave Send mode, the SCL clock from the host is received. This product sends data for the slave and receives a reply from the host. The sequence of data sent from the machine is shown in the following figure.

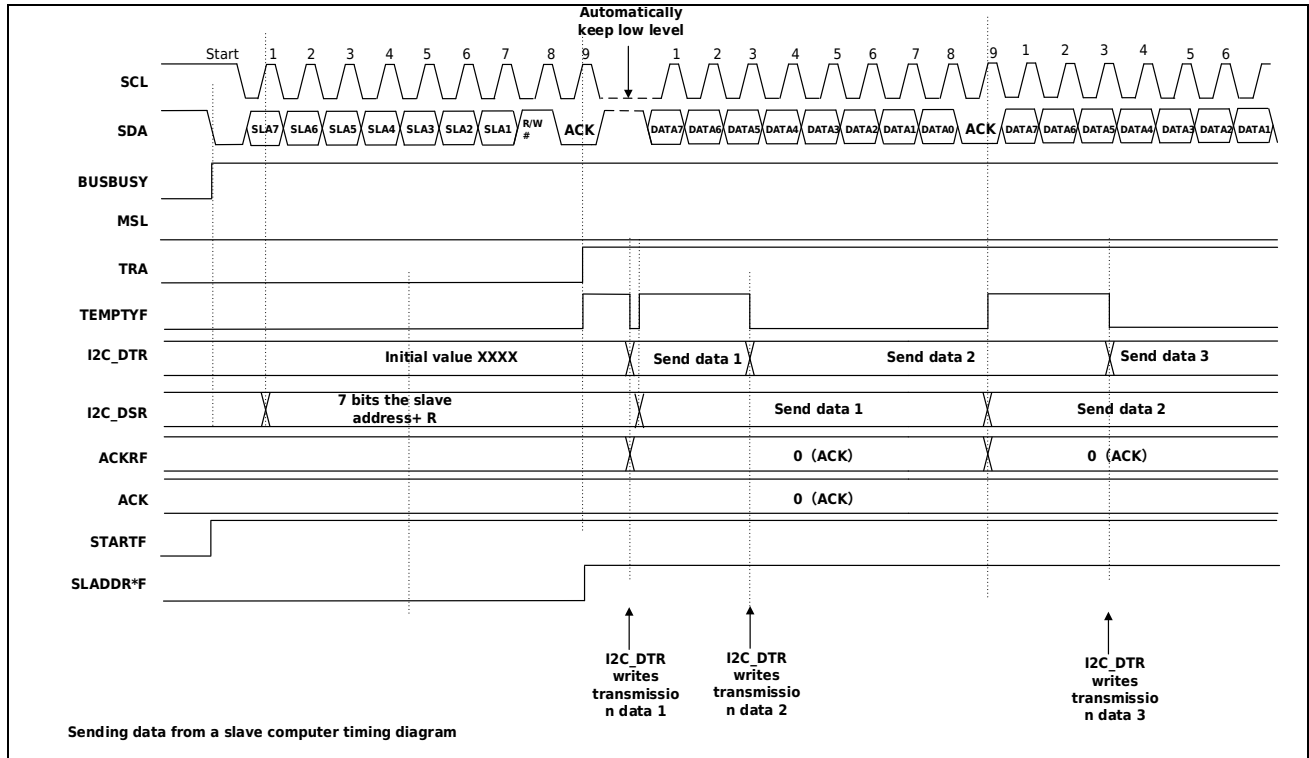


Figure 29-7 Sequence Diagram of Slave Send Mode in 7-bit Address Format (Instance)

Receive Data from Machine

In slave receiving mode, the SCL clock and data from the host 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.

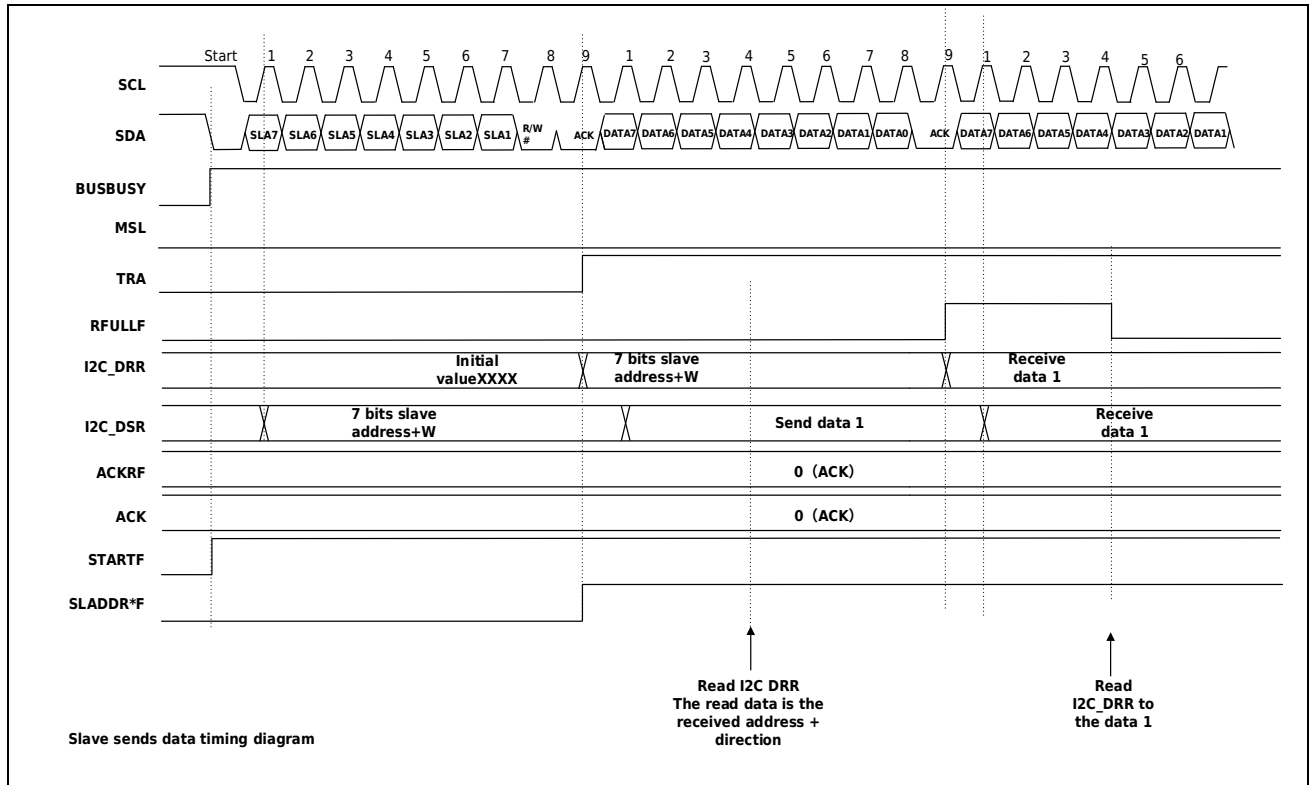


Figure 29-8 Sequence Diagram of 7-bit Address Format Slave Receive Mode (Instance)

29.3.1.4 Stop Condition

Through I2C_CR1.The STOP bit release stop condition.

Issue the stop condition by placing the STOP position "1" in the state where the I2C_SR.BUSY flag is "1" (busy bus) and the I2C_SR.MSL bit is "1" (host mode).

29.3.1.5 Restart Condition

Through I2C_CR1.The RSTART bit generation restart condition.

The RSTART position "1" generates a restart condition when the I2C_SR.BUSY flag is "1" (busy bus) and the I2C_SR.MSL bit is "1" (host mode).

By restarting the condition, the host can switch transmit/receive modes without releasing BUS authority. Communication can also be established with another slave without releasing BUS authority.

29.3.1.6 SCL Clock Synchronization

When using the I2C bus in the multi-host mode, the SCL clock and SCL clock may conflict due to competition with other hosts. If the SCL clock conflicts, the host 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 hosts, 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 interrupted and the SCL line is released. At this time, if the low level width of the SCL clock of other hosts is greater than the low level width set by SLOWW, the low level width of the SCL clock is prolonged. When other hosts 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.

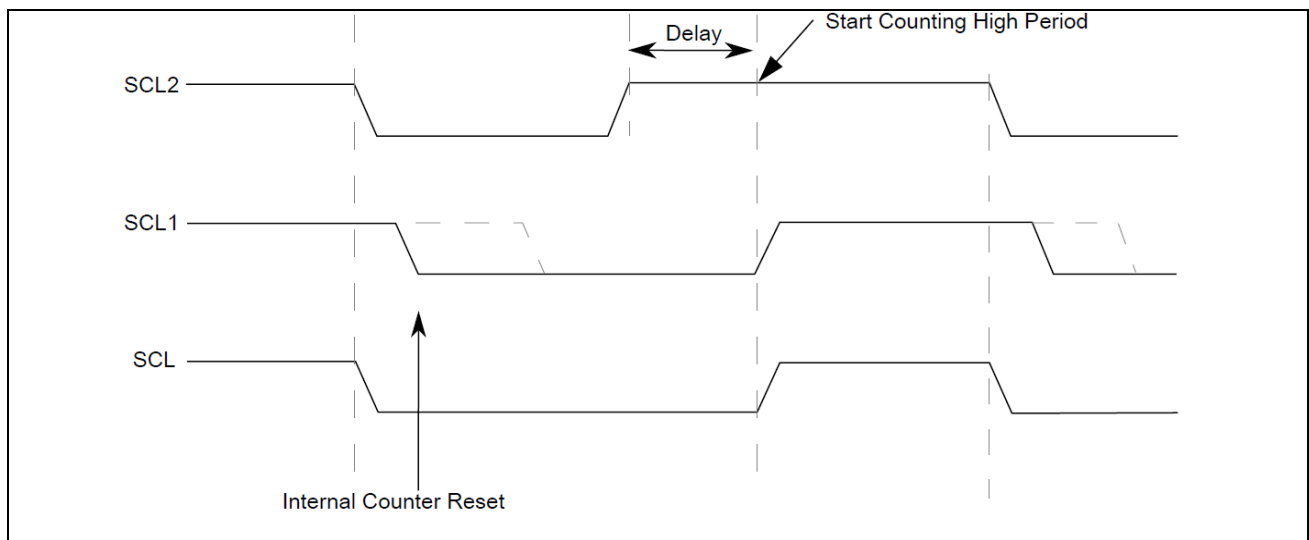


Figure 29-9 SSCL Synchronization Time Series

29.3.1.7 Arbitration

I2C bus is a real multi-host bus that allows multiple hosts to connect.

If two or more hosts try to control the bus at the same time, the SCL clock synchronization process determines the bus clock. The low period of the bus clock depends on the longest low-level clock and the high period depends on the shortest high-level clock. The results of arbitration are determined by the data collected at high levels. When the SDA output sent is a high level output (the SDA pin is in a high impedance state) and the SDA line is detected to be at a low level, arbitration fails. The I2C _ SR.AROLF bit will be hardware "1". If a host 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.

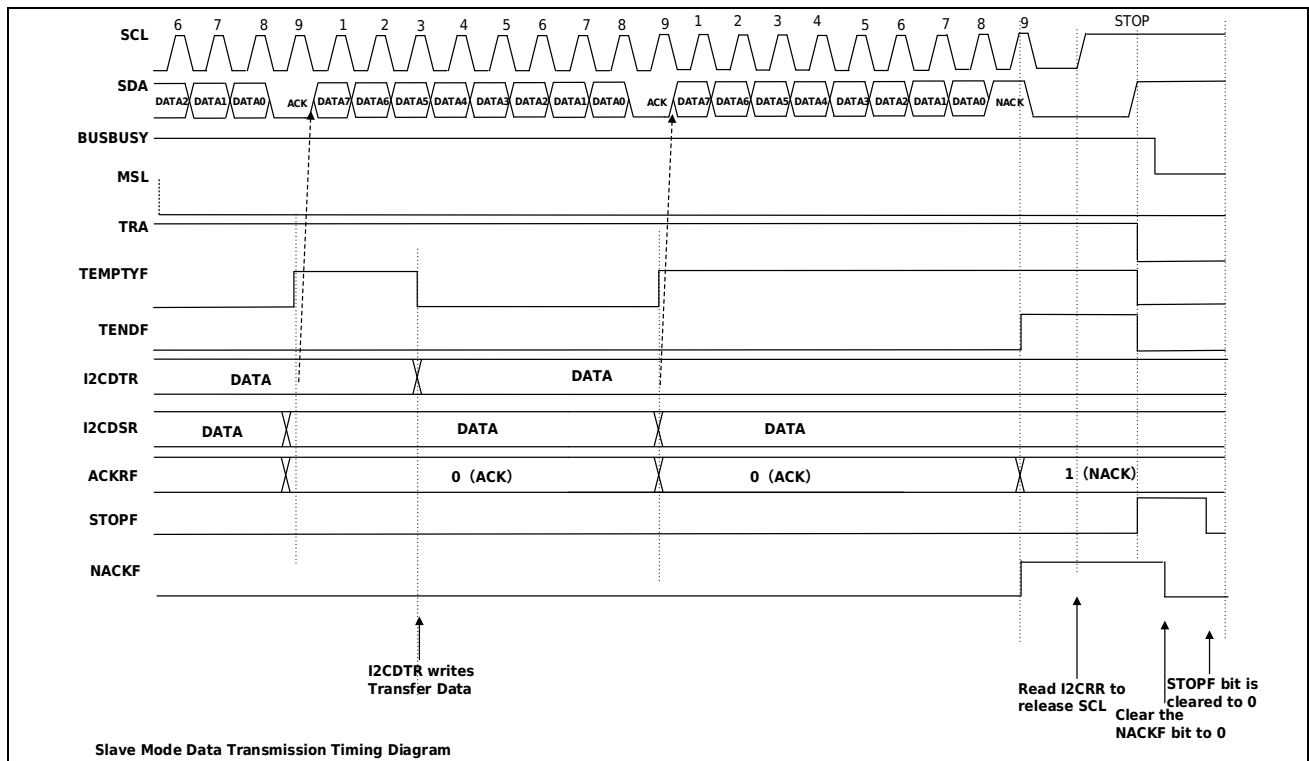


Figure 29-11 Sending Sequence Diagram of Slave (2)

[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.

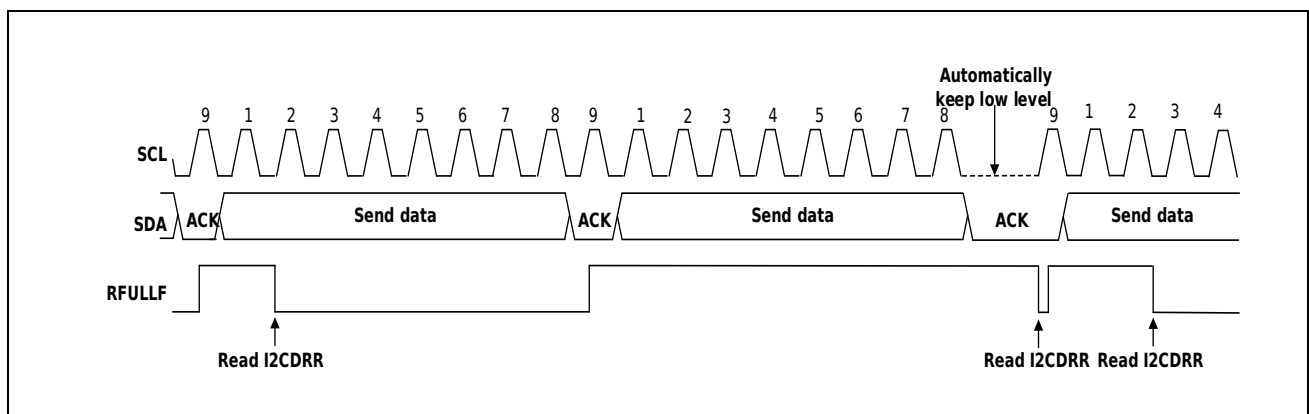


Figure 29-12 Slave Receiving Timing Diagram

[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.

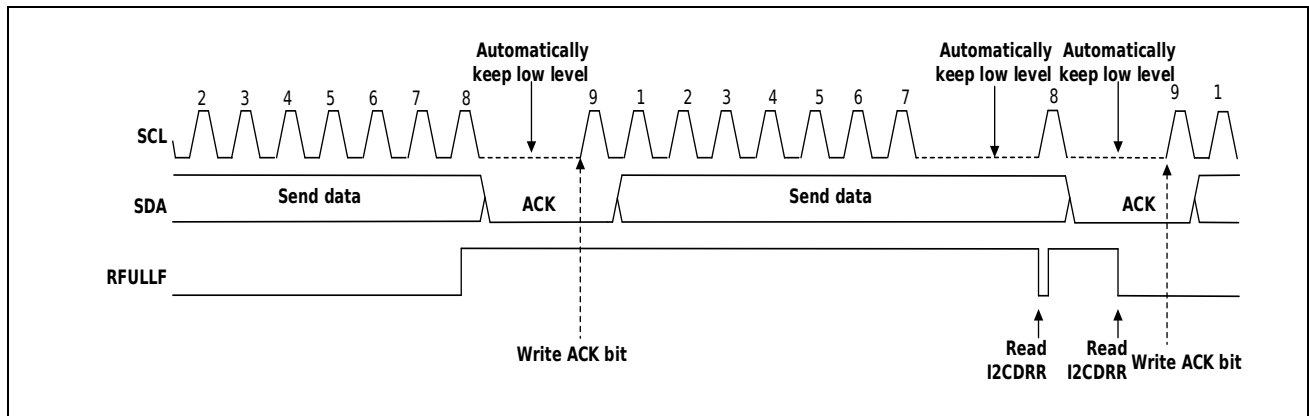


Figure 29-13 Fast ACK/NACK Timing Diagram

29.3.2 Address Matching

As a slave, two kinds of addresses other than broadcast address and host notification address can be set, and 7-bit address or 10-bit address format can be set for slave address.

29.3.2.1 Slave address matching

This I2C C bus can set two kinds of slave address, and has corresponding slave address detection function. When SLADDR1EN and SLADDR0EN are "1", I2C _ SLR1 and I2C _ SLR0 register subordinate addresses can be detected.

If the set slave addresses match, the corresponding SLADDR1F and SLADDR0F are set to "1" at the rising edge of the 9th clock of the SCL clock, and then the I2C_SR.RFULLF flag or I2C_SR.EMPTIF flag is set to "1" according to the subsequent R/W# bit. As a result, I2C _ 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 becoming "1" are shown in the following figures.

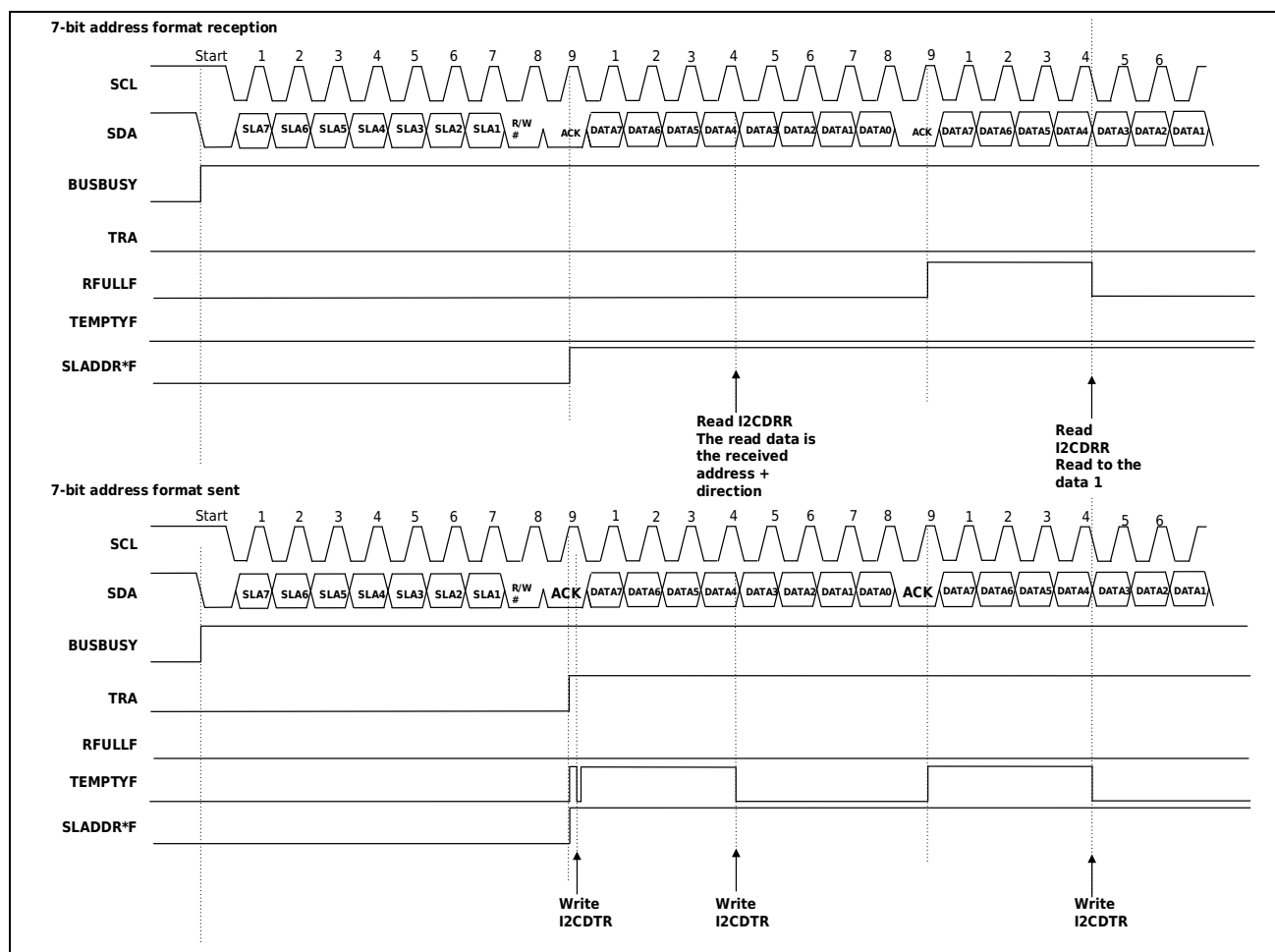


Figure 29-14 Timing when selecting 7-bit address format

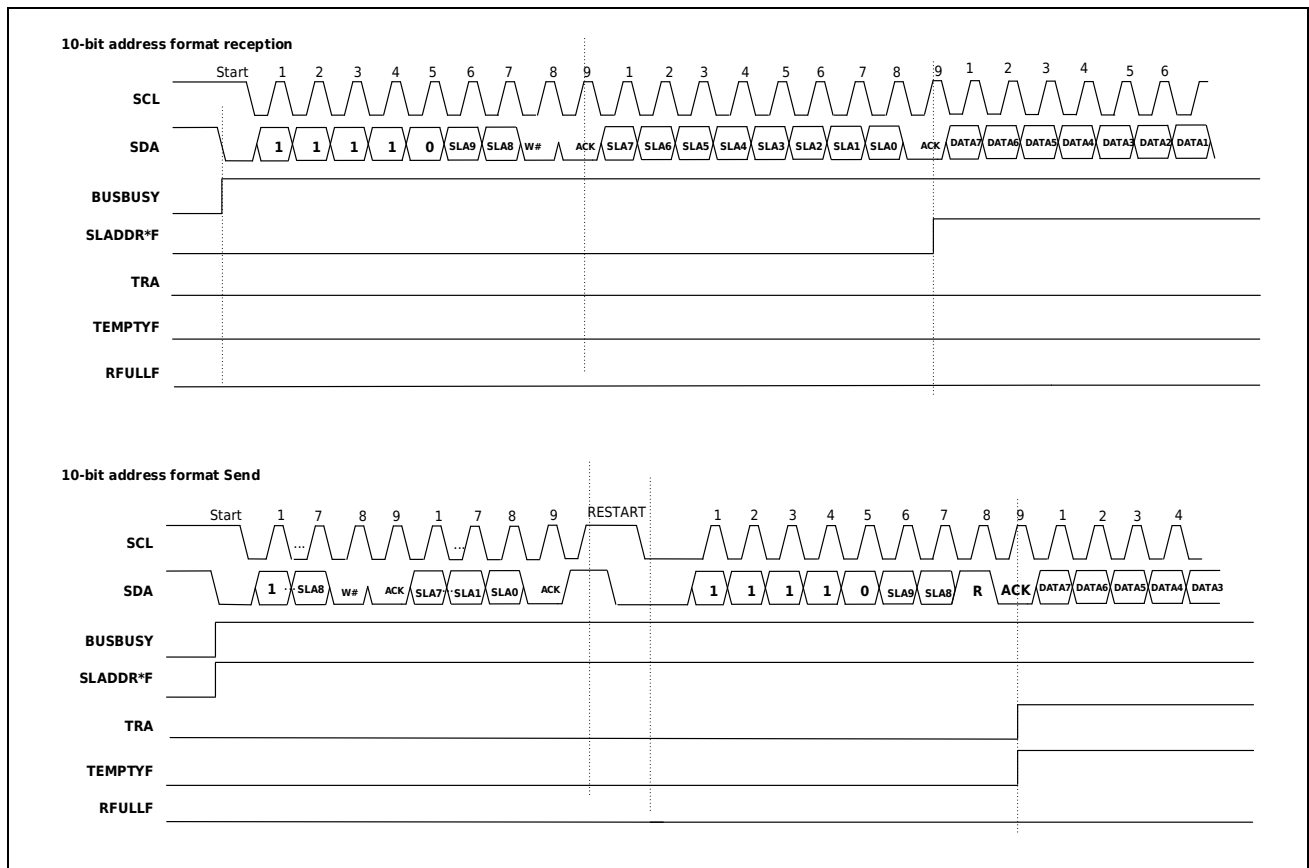


Figure 29-15 Timing when selecting 10-bit address format

29.3.2.2 Broadcast Address Matching

When I₂C_CR1. When the GCEN bit is "1", the broadcast address can be detected (0000 000 b + 0 [W]).

But after the start condition or restart condition, the address is 0000 000 b + 1 [R], which is considered the slave address of All "0" and not the broadcast address.

If the broadcast address is matched, the I2C_SR.GCF flag is set to "1" at the rising edge of the 9th SCL clock.

The broadcast address matches the same run as the normal slave receive run.

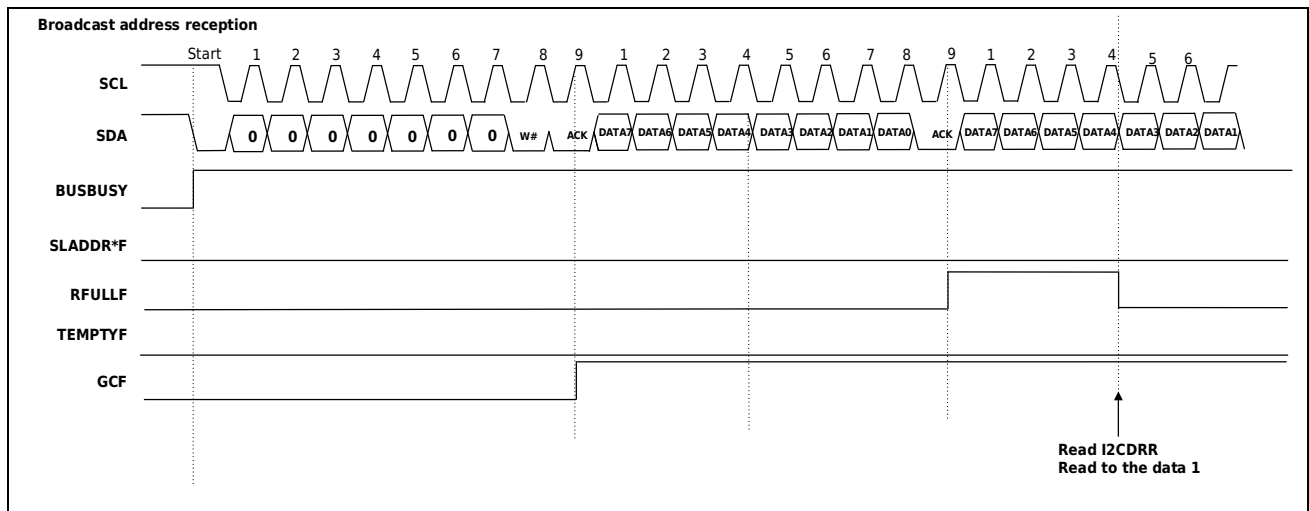


Figure 29-16 Broadcast Address Matching Timing Diagram

29.3.2.3 SMBus Host Address Matching

This product has the host address detection function when SMBus runs. If the I2C_CR1.SMBHOSTEN bit is set to "1" when the I2C_CR1.SMBUS bit is "1", the host address (0001 000b) can be detected in slave receive mode (I2C_CR1.MSL bit TRA bit is "00b").

If the SMBUS host address is detected, the I2C_SR.SMBHOSTF flag is set to "1" at the rising edge of the 9th SCL clock.

Even if the bit following the SMBUS host address (0001000b) is an Rd bit (R/W # bit receives "1"), the SMBUS host address can be detected. The running of SMBUS host address detection is the same as that of normal slave mode.

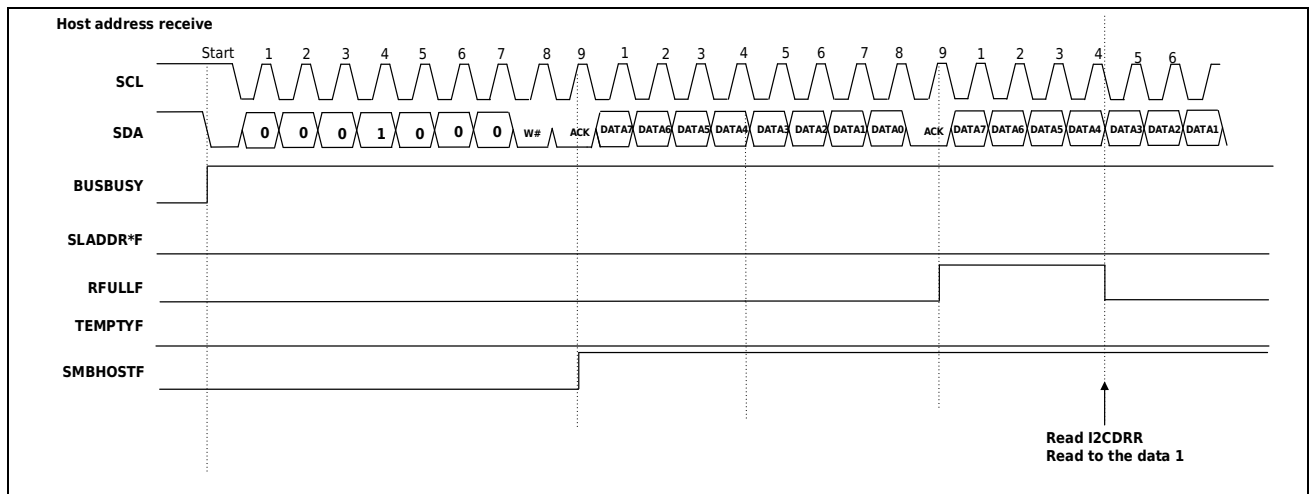


Figure 29-17 SMBus Host Address Matching Timing Diagram

29.3.2.4 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 "1" when the I2C_CR1.SMBUS bit is "1", the SMBUS alarm response address (0001 100b) can be detected in the slave receiver mode (I2C_CR1.MSL bit TRA bit is "00b").

If the SMBUS alert response address is detected, the I2C_SR.SMBARLERTF flag is set to "1" at the rising edge of the 9th SCL clock.

The operation of SMBUS alarm response address detection is the same as that of normal slave mode.

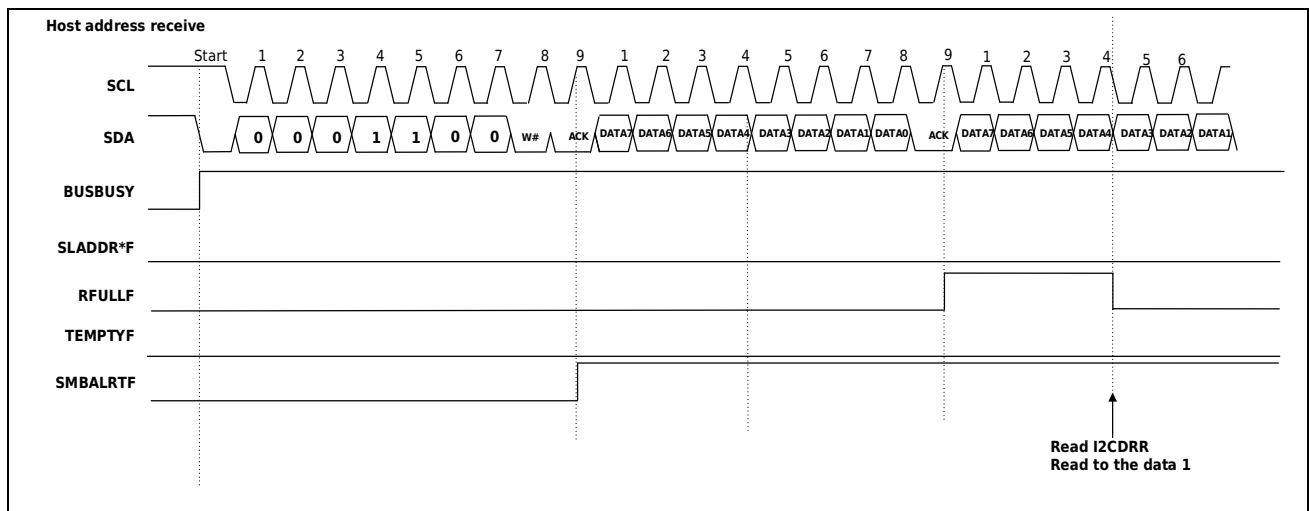


Figure 29-18 SMBus Alarm Response Address Matching Timing Diagram

29.3.2.5 SM Bus 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 (1100 001b) can be detected in slave receive mode (I2C_CR1.MSL bit TRA bit is "00b").

If the SMBUS default address is detected, the I2C_SR.SMBDEFAULTF flag is set to "1" at the rising edge of the 9th SCL clock.

The SMBUS default address detection runs the same as the normal slave mode.

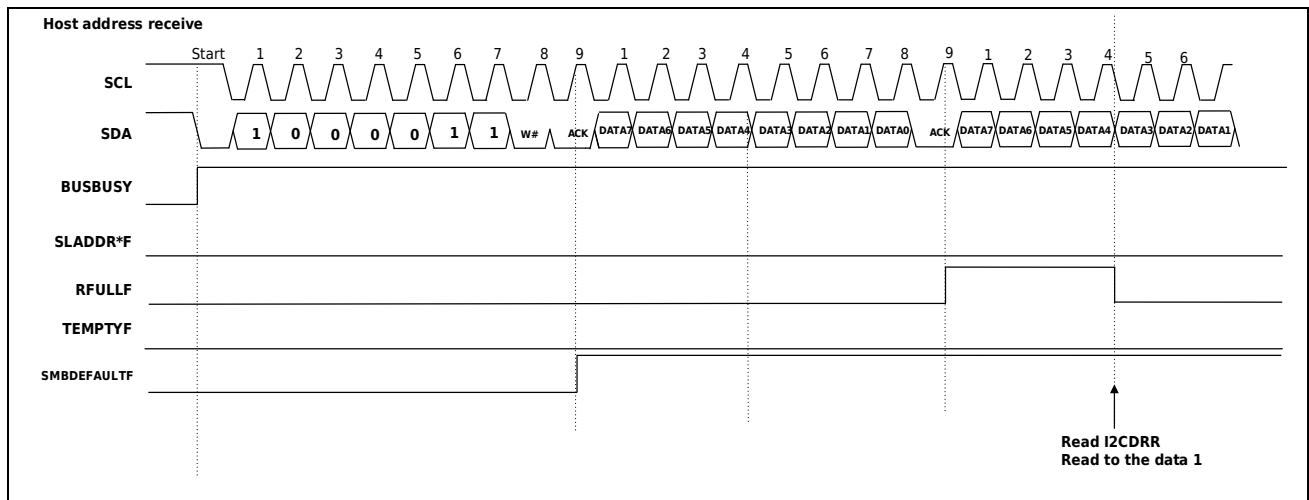


Figure 29-19 SMBus Default Address Timing Diagram

29.3.3 SM Bus Action

This I2C interface can be implemented with SMBus (Ver.2.0) communication as a reference. I₂C _CR1 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.

29.3.3.1 SMBus Timeout Measurement

1) SCL level When timeout Measurement

In the busy state of the bus, it can be detected by detecting that the low level or high level of the SCL line has been fixed for a certain period of time, and an abnormal state of the bus can be 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: Set).

■ 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 TTIMEOUT: 25 ms (min), the slave computer needs to release the bus.

3) Timeout Measurement of Host

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 host 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 [host] 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 (TTIMEOUT: 25ms (min)), the host needs to abort processing. When sending a host, you must immediately abort the sending (write I2C _ DTRregister). Abort host processing through release stop conditions.

29.3.3.2 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.

29.3.4 Reset

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 "01b") The bus status such as start condition cannot be monitored during the process.

29.3.5 Interrupt and Event Signal Output

I₂C has four interrupts and event outputs for triggering the activation of other peripheral circuits for user selection. These include: Occurrence of communication error (arbitration failure detection, NACK the timeout detection, start condition detection, stop condition detection), receiving end, sending data null, sending end.

Table 29-2 Interruption List

Name	Interrupt Source	Interrupt Flag	Interrupt Condition
I2C_EEI	Communication error/communication event	ARLOF	ARLOF=1&ARLOIE=1
		SLADDR0F	SLADDR0F=1& SLADDR0IE=1
		SLADDR1F	SLADDR1F=1& SLADDR1IE=1
		SMBALRTF	SMBALRTF =1& SMBALRTIE=1
		SMBHOSTF	SMBHOSTF =1& SMBHOSTFIE=1
		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	Transmit data null	EMPTYF	EMPTYF=1&EMPTYIE=1
I2C_TEI	End of transmission	TENDF	TENDF=1&TENDIE=1

Table 29-3 Event Signal Output List

Name	Event source	Event conditions
I2C_EEI	Communication error/communication time	ARLOF=1
		SLADDR0F=1
		SLADDR1F=1
		SMBALRTF=1
		SMBHOSTF=1
		SMBDEFAULTF=1
		GENCALLF=1
		NACKF=1
		TMOUTF=1
		STARTF=1
		STOPF=1
I2C_RXI	Received data full	RFULLF=1
	Received data is not empty	RFREQF=1
I2C_TXI	Transmit data null	EMPTYF=1
I2C_TEI	End of transmission	TENDF=1

Note: When I2C_RXI , I2C_TXI , I2C_TEI , and I2C_EEI events are used as AOS sources, they can trigger DMA module actions, but cannot trigger other AOS target modules.

29.3.6 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.

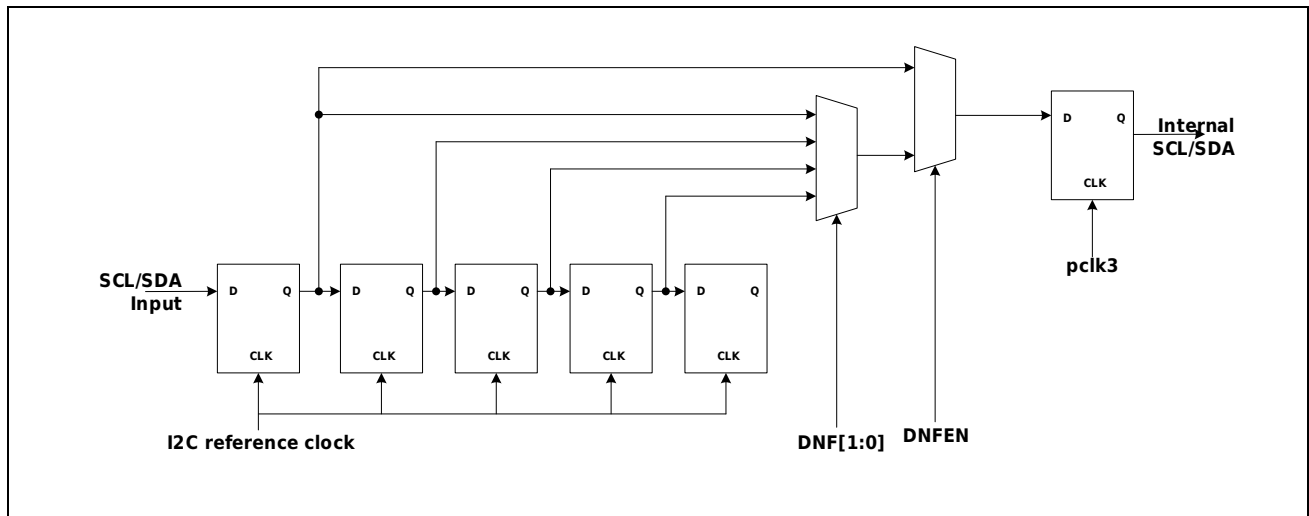


Figure 29-20 Digital Filtering Circuit Block Diagram

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 periods.

The input signal of the SCL pin (or the input signal of the SDA pin) is sampled on the falling edge of the I2C internal clock. If the output of the trigger circuit with the number of 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.

29.4 Application Software Setting I2C Initialization Process

When starting to send or receive data, the following steps must be followed for initialization.

1. PE bit is set to 0.
2. Set SWRST to 1, communication reset
3. PE bit set to 1, internal state reset
4. Set slave address format and address
5. Set baud rate
6. Set control register function and interrupt as needed
7. SWRST bit is set to 0 to cancel internal state reset.
8. The initialization ends. Can send and receive data.

29.5 Register Description

Table 29-4 Register Summary

I2C1 reference address: 0x4003B400

I2C2 base address: 0x4003B800

I2C3 base address: 0x4003BC00

Register Name	Symbol	Offset Address	Bit Width	Reset Value
I2C control register 1	I2C_CR1	0x00	32	0x0000 0040
I2C control register 2	I2C_CR2	0x04	32	0x0000 0000
I2C control register 3	I2C_CR3	0x08	32	0x0000 0006
I2C control register 4	I2C_CR4	0x0C	32	0x0030 0307
I2C slave address register0	I2C_SLR0	0x10	32	0x0000 1000
I2C slave address register1	I2C_SLR1	0x14	32	0x0000 0000
I2C SCL Level Timeout Control Register	I2C_SLTR	0x18	32	0xFFFF FFFF
I2C state register	I2C_SR	0x1C	32	0x0000 0000
I2C state reset	I2C_CLR	0x20	32	0x0000 0000
I2C data sending register	I2C_DTR	0x24	8	0xFF
I2C data receiving register	I2C_DRR	0x28	8	0x00
I2C Data Shift Register	I2C_DSR	-	-	-
I2C clock control register	I2C_CCR	0x2C	32	0x0000 1F1F
I2C Filter Control Register	I2C_FLTR	0x30	32	0x0000 0010
I2C FIFO Control Register	I2C_FSTR	0x34	32	0x0000 0000
I2C Slave Address Receive Register	I2C_SLVADRR	0x38	32	0x0000 0000

29.5.1 I2C Control Register 1 (I2C_CR1)

Reset value: 0x00000040

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SWRST	-	-	-	-	ACK	STOP	START	RESTART	GCEN	-	SMBHOSTEN	SMBDEFAULTEN	SMBALRTEN	SMBUS	PE

Bit	Flag	Bit Name	Function	Read and write									
b31-16	Reserved	-	Read as "0", write as "0"	R/W									
b15	SWRST	Software Reset	0: Decommission 1: Software Reset Combination of native and PE bits, selecting internal state reset or communication reset <table><tr><th>SWRST</th><th>PE</th><th>Reset content</th></tr><tr><td>1</td><td>0</td><td>Communication Reset: All registers and internal state resets within I2C.</td></tr><tr><td>1</td><td>1</td><td>Internal state reset: I2C _SR, I2C _DSRregister and internal state machine reset</td></tr></table>	SWRST	PE	Reset content	1	0	Communication Reset: All registers and internal state resets within I2C.	1	1	Internal state reset: I2C _SR, I2C _DSRregister and internal state machine reset	R/W
SWRST	PE	Reset content											
1	0	Communication Reset: All registers and internal state resets within I2C.											
1	1	Internal state reset: I2C _SR, I2C _DSRregister and internal state machine reset											
b14-11	Reserved	-	Read as "0", write as "0"	R/W									
b10	ACK	Send an answer	0: Reply bit sends' 0 '(send ACK) 1: Reply bit sends "1" (send NACK)	R/W									
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	R/W									
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	R/W									
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	R/W									
b6	GCEN	Broadcast call time	0: Invalid broadcast address detection 1: Broadcast address detection is valid	R/W									
b5	Reserved	-	Read as "0", write as "0"	R/W									
b4	SMBHOSTEN	Allow matching of SMBUS host address	0: Do not match SMBUS host address 1: Allow match of SMBUS host address	R/W									
b3	SMBDEFAULTEN	Allow matching of SMBUS default address bits	0: Do not match SMBUS default address 1: Allow match of SMBUS default address	R/W									
b2	SMBALRTEN	Allow matching of SMBUS alarm response address bits	0: Disable SMBUS alarm response address 1: Allow SMBUS alarm response address										
b1	SMBUS	SMBUS/I2C C bus mode selection bit	0: I2C bus mode 1: SMBUS bus mode	R/W									
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	R/W									

29.5.2 I2C Control Register 2 (I2C_CR2)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	SMB ALRTI E	SMB HOST IE	SMBD EFAUL TIE	GEN CALLI E	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	TMOU TIE	-	NAC KIE	RF REQ IE	-	ARL OIE	-	TEMP TYIE	RFUL LIE	-	STOP IE	TEN DIE	SLAD DR1IE	SLAD DR0IE	STAR TIE

Bit	Flag	Bit Name	Function	Read and write
b31~b24	Reserved	-	Read as "0", write as "0"	R/W
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	R/W
b22	SMBHOSTIE	SMBUS host address matching consistent Interrupt enable	0: SMBUS host address matching consistent interrupt disable 1: SMBUS host address matching consistent interrupt enable	R/W
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	R/W
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	R/W
b19~b15	Reserved	-	Read as "0", write as "0"	R/W
b14	TMOUTIE	When timeout interrupt permission	0: When timeout interrupt disabled 1: When timeout interrupt permission	R/W
b13	Reserved	-	Read as "0", write as "0"	R/W
b12	NACKIE	NACK interrupt Allowed	0: NACK interrupt disable received 1: NACK interrupt enable received	R/W
b11	RFREQIE	FIFO receive request interrupt enable	0: Rx FIFO interrupt request disabled 1: Receive FIFO interrupt request enable	R/W
b10	Reserved	-	Read as "0", write as "0"	R/W
b9	ARLOIE	Arbitration failure interrupt permission	0: Arbitration failure interrupt prohibition 1: Arbitration failure interrupt permission	R/W
b8	Reserved	-	Read as "0", write as "0"	R/W
b7	EMPTYIE	Send data null interrupt allowed bit	0: Send data null interrupt disable 1: Null interrupt permission for sending data	R/W
b6	RFULLIE	Received data full interrupt allowed bit	0: Full interrupt disablement of received data 1: Receive data full interrupt permission	R/W
b5	Reserved	-	Read as "0", write as "0"	R/W
b4	STOPIE	Stop condition interrupt permission	0: Bus detected stop condition interrupt disable 1: Bus detects stop condition interrupt permit	R/W
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	R/W
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	R/W
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	R/W
b0	STARTIE	Start condition/restart condition interrupt permission	0: Bus detected start condition interrupt disable 1: Bus detected start condition interrupt permit	R/W

29.5.3 I2C Control Register 3 (I2C_CR3)

Reset value: 0x00000006

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	ACK EN	-	-	-	-	HTM OUT	LTM OUT	TMOU TEN

Bit	Flag	Bit Name	Function	Read and write
b31~b8	Reserved	-	Read as "0", write as "0"	R/W
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 RFULLF flag is "1". (On the falling edge of the 8th clock, the SCL line remains low. Unhold low by writing ACK bits.	R/W
b6~b3	Reserved	-	Read as "0", write as "0"	R/W
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.	R/W
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.	R/W
b0	TMOUTEN	When timeout feature enable bit	0: SCL low level When timeout detected clock function disabled 1: The timeout function of detecting SCL level is enabled	R/W

29.5.4 I2C Control Register 4 (I2C_CR4)

Reset value: 0x00301b07

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	SDADLY[3:0]			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	BUSFRE E CLREN	-	BUSW AIT	-	-	-	-	-	-	-	-	-	-

Bit	Flag	Bit Name	Function	Read and write
b31~b22	Reserved	-	Read as "0", write as "0"	R/W
b21~b20	Reserved	-	Read as "1", write as "1"	R/W
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	R/W
b15~b13	Reserved	-	Read as "0", write as "0"	R/W
b12	BUSFREE_CLREN	Bus end status reset bit	0: Disable the I2C bus STOP flag to clear the SMBALRTF, SMBHOSTF, SMBDEFAULTF, GENCALLF, MSL, SLADDR1F, SLADDR0F, STARTF bits in the I2C_SLVADRR slave receive address register and the I2C_SR status register 1: Enable the I2C bus STOP flag to clear the SMBALRTF, SMBHOSTF, SMBDEFAULTF, GENCALLF, MSL, SLADDR1F, SLADDR0F, STARTF bits in the I2C_SLVADRR slave receive address register and the I2C_SR status register	R/W
b11	Reserved	-	Read as "1", write as "1"	R/W
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.	
b9~b8	Reserved	-	Read as "1", write as "1"	R/W
b7~b3	Reserved	-	Read as "0", write as "0"	R/W
b2~b0	Reserved	-	Read as "1", write as "1"	R/W

29.5.5 I2C Slave Address Register 0 (I2C _ SLR0)

Reset value: 0x00001000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	MASK0EN	MSLADDR0 [9:0]									
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ADDRMOD0	-	-	SLADDR0EN	-	-	SLADDR0 [9:0]									

Bit	Flag	Bit Name	Function	Read and write
b31~b27	Reserved	-	Read as "0", write as "0"	R/W
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	
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.	
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.	
b15	ADDRMOD0	7-bit/10-bit address format selection bit	0: Select 7-bit address format 1: Select 10-bit address format	R/W
b14~b13	Reserved	-	Read as "0", write as "0"	R/W
b12	SLADDR0EN	Slave address 0 significant bit	0: Invalid set value from machine address register0 1: Valid from machine address register0	R/W
b11~b10	Reserved	-	Read as "0", write as "0"	R/W
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.	R/W
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.	R/W

29.5.6 I2C Slave Address Register1 (I2C _ SLR 1)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	MASK1EN	MSLADDR1 [9:0]									
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ADDRMOD1	-	-	SLADDR1EN	-	-	SLADDR1 [9:0]									

Bit	Flag	Bit Name	Function	Read and write
b31~b27	Reserved	-	Read as "0", write as "0"	R/W
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	R/W
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.	
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 MASK1EN 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.	
b15	ADDRMOD1	7-bit/10-bit address format selection bit	0: Select 7-bit address format 1: Select 10-bit address format	R/W
b14~b13	Reserved	-	Read as "0", write as "0"	R/W
b12	SLADDR1EN	Slave address 1 significant bit	0: Invalid set value from machine address register1 1: Valid from machine address register1	R/W
b11~b10	Reserved	-	Read as "0", write as "0"	R/W
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.	R/W
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.	R/W

29.5.7 I2C SCL Level When Timeout Control Register(I2C_SLTR)

Reset value: 0xFFFFFFFF

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	Flag	Bit Name	Function	Read and write
b31~b16	TOUTHIGH	SCL high level timeout period	TOUTHIGH sets the SCL high timeout period. SCL high level timeout time = TOUTHIGH * I2C reference clock period	R/W
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	R/W

29.5.8 I2C State Register (I2C _ SR)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
RF REQ	-	TFST [1:0]	RF FULL	RF EMPTY	TF FULL	TF EMPTY	SMB ALRT F	SMB HOST F	SMB DEFA ULTF	GEN CALL F	-	TRA	BUS Y	MSL	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	TMOU TF	-	NAC KF	-	ACK RF	ARL OF	-	TEM PTYF	RFUL LF	-	STOP F	TEN DF	SLAD DR1F	SLAD DR0F	STAR TF

Bit	Flag	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	Reserved	-	Read as "0", write as "0"	R
b29~b28	TFST [1:0]	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 byte of valid data 11: Reserved bit	R
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 host address Set the "1" condition: Received address matches 0001 100b Clear "0" condition: SMBALRTFCLR writes "1" Stopped condition detected Communication reset	R
b22	SMBHOSTF	SMBUS Host Address Matching Consistent Flag Bit	0: Not matched to SMBUS host address 1: Match to SMBUS host address The consistent conditions for address matching are as follows: Set the "1" condition: Received address matches 0001 000b Clear "1" condition: SMBHOSTFCLR writes "1" Stopped condition detected Communication reset and internal 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: Received addresses match 1100,001b Clear "1" condition: SMBDEFAULTFCLR Write "1" Stopped condition detected Communication 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	R

b19	Reserved	-	Read as "0", write as "0"	R															
b18	TRA	Transmit receive select bit	This bit indicates whether to choose to send or receive data. 0: Receive data 1: Send data This bit can be set to 1 and cleared to 0. Hardware setting "1" condition Start condition detected In host mode, the sent R/W bit is 0 In slave mode, the address matches and the received R/W bit is 1 "0" Condition of Hardware Cleaning Stopped condition detected In host mode, the sent R/W bit is 1 In slave mode, the address matches and the received R/W bit is 0 Communication reset	R/W															
b17	BUSY	Bus busy flag bit	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	R															
b16	MSL	Host-slave selection bit	This bit indicates whether the host is still a slave. 0: Slave mode 1: Host mode The operating mode of I2C is represented by a combination of TRA bits. <table><tr><td>MSL</td><td>TRA</td><td>I2C operating mode</td></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>Host receiving mode</td></tr><tr><td>1</td><td>1</td><td>Host 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	MSL	TRA	I2C operating mode	0	0	Slave receiving mode	0	1	Slave mode	1	0	Host receiving mode	1	1	Host send mode	R/W
MSL	TRA	I2C operating mode																	
0	0	Slave receiving mode																	
0	1	Slave mode																	
1	0	Host receiving mode																	
1	1	Host send mode																	
b15	Reserved	-	Read as "0", write as "0"	R															
14	TMOUTF	When timeout flag bit	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: 1: SCL level when timeout writes "1" Communication reset	R															
b13	Reserved	-	Read as "0", write as "0"	R															
b12	NACKF	Nack flag bit	0: No NACK Received 1: NACK received Set the "1" condition: Receive NACK in send mode Clear "0" condition: NACKFCLR writes "1" Communication reset	R															
b11	Reserved	-	Read as "0", write as "0"	R															
b10	ACKRF	Receiving bit	0: Received ACK bit is "0" (ACK received) 1: Received reply bit is' 1 '(receive NACK) Set the "1" condition: Receive NACK in send mode Clear "0" condition: Receive ACK in transmit mode Communication reset	R															

b9	ARLOF	Arbitration failure flag bit	0: No arbitration failure 1: Arbitration failure Set the "1" condition: Failure to arbitrate Clear "0" condition: ARLOFCLR Write "1" Communication reset	R
b8	Reserved	-	Read as "0", write as "0"	R
b7	EMPTYF	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	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	R
b5	Reserved	-	Read as "0", write as "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	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 EMPTYF = 1, the 9th rising edge of SCL is "1" Clear "0" condition: Stopped condition detected Write I2C_DTR TENDFCLR writes "1" Communication 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.ADDRMODE1 bit is "0", the received slave address matches I2C_SLR1.SLADDR1 [7:1]. When I2C _ SLR 1.When the ADDRMODE1 bit is "1", the first byte address of the 10-bit slave address is received and 11110b + 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	R
b1	SLADDR0F	Register 0 matching consistency flag	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 ADDRMODE0 is "0", the received slave address and I2C_SLR0. When SLADDR0 [7: 1] matches. When I2C _ SLR 0.When ADDRMODE0 bit is "1", receive the first byte address of 10-bit slave address and 11110b + 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	R

b0	STARTF	Start condition/restart condition flag bit	0: Bus not detected starting condition 1: Start condition detected by bus Set "1" condition 1) Starting condition detected "0" Condition in Qing Dynasty 1) Stop condition detected 2) STARTFCLR writes "1" 3) Communications resetting	R

29.5.9 I2C State Reset Register (I2C _ CLR)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	SMB ALRT FCLR	SMB HOST FCLR	SMB DEFA ULTF CLR	GEN CALL FCLR	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	TMO UTFCL R	-	NAC KFCL R	-	RF REQ CLR	ARL OFCL R	-	-	RFUL LFCL R	-	STOP FCLR	TEN DFCL R	SLA DDR1 FCLR	SLA DDR0 FCLR	STAR TFCL R

Bit	Flag	Bit Name	Function	Read and write
b31~b24	Reserved	-	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	SMBHOSTFCLR	SMBUS Host address matching consistency flag zero bit	Write "1" to reset the SMBHOSTF 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	Reserved	-	Write "0" on write	W
b14	TMOUTFCLR	When timeout flag bit	Write '1' to clear the TMOUTF flag bit	W
b13	Reserved	-	Write "0" on write	W
b12	NACKFCLR	Nack flag bit	Write "1" to clear the NACKF flag bit	W
b11	Reserved	-	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 reset the ARLOF flag bit	W
b8~b7	Reserved	-	Write "0" on write	W
b6	RFULLFCLR	Received data full flag bit	Write "1" to clear RFULLF flag bit	W
b5	Reserved	-	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 clear the STARTF flag bit	W

29.5.10 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 _ DTRregister 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.

29.5.11 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 (I2C _ DSR) to the I2C _ 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 29-12 and Figure 29-13) 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.

I2C works in FIFO mode, when the I2C_SR.RFULLF flag is "1" (that is, the receive FIFO has stored 2 bytes of data and the FIFO is full), if the data in the receive FIFO 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 of the data reception until the data in the receive FIFO is read.

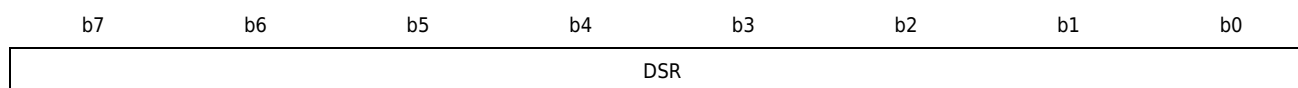
The I2C receive path is selected by the FEN bit in the I2C FIFO status register.

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.

29.5.12 I2C Data Shift Register (I2C _ 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.

29.5.13 I2C Clock Control Register (I2C_CCR)

Reset value: 0x00001F1F

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	FMPLUSEN	-	-	-		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	Flag	Bit Name	Function	Read and write
b31~b24	Reserved	-	Read as "0", write as "0"	R/W
b23	FMPLUSEN	Fast plus mode enable bit	0: Fast plus mode disable 1: Fast plus mode enable	R/W
b22~b19	Reserved	-	Read as "1", write as "1"	R/W
b18~b16	CKDIV [2:0]	Setting of I2C reference clock frequency	0 0 0: I2C reference clock frequency = PCLK0/1 0 0 1: I2C reference clock frequency = PCLK0/2 0 1 0: I2C reference clock frequency = PCLK0/4 0 1 1: I2C reference clock frequency = PCLK0/8 1 0 0: I2C reference clock frequency = PCLK0/16 1 0 1: I2C reference clock frequency = PCLK0/32 1 1 0: I2C reference clock frequency = PCLK0/64 1 1 1: I2C reference clock frequency = PCLK0/128	R/W
b15~b8	SHIGHW [7:0]	Set SCL high level width bit	Setting the High Level Width of SCL Clock	R/W
b7~b0	SLOWW [7:0]	Set SCL low level width bit	Setting the Low Level Width of SCL Clock	R/W

SHIGHW bit (set SCL high level width bit)

In host mode, SHIGHW is used to set the high level width of the SCL clock. The setting is invalid in slave mode.

SLOWW bit (set SCL low level width bit)

SLOWW is used to set the low 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}) 250ns (~100kbps : standard mode), 100ns (~400kbps : fast mode), 50ns (~1Mbps : FM+ mode)

Baud rate:

DNFE=0, CKDIV=000

Baud rate = 1/{ [(SHIGHW + 3) + (SLOWW + 3)]/ΦI2C + SCL rise time + SCL fall time}

DNFE=1, CKDIV=000

Baud rate = 1/{ [(SHIGHW + 3 + filtering capability) + (SLOWW + 3 + filtering capability)]/ΦI2C + SCL rise time + SCL fall time}

DNFE=0, CKDIV!=000

Baud rate = 1/{ [(SHIGHW + +2) + (SLOWW + +2)]/ΦI2C + SCL rise time + SCL fall time}

DNFE=1, CKDIV!=000

Baud rate = $1/\{[(\text{SHIGHW} + 2 + \text{filter capability}) + (\text{SLOWW} + 2 + \text{filter capability})]/\Phi\text{I}2\text{C} + \text{SCL rise time} + \text{SCL fall time}\}$

Note:

- 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 protocol.
- The $\Phi\text{I}2\text{C}$ in the above formula refers to the I2C reference clock period set by $\text{I}2\text{C\_CCR.CKDIV}$.

29.5.14 I2C Filtering Control Register (I2C_FLTR)

Reset value: 0x00000010

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

Bit	Flag	Bit Name	Function	Read and write
b31~b6	Reserved	-	Read as "0", write as "0"	R/W
b5	ANFEN	Analog filtering allowed bit	0: Analog filtering function disabled 1: Analog Filtering Allowed	R/W
b4	DNFEN	Digital filtering allowed bit	0: Digital filtering disabled 1: Digital filtering is allowed	R/W
b3~b2	Reserved	-	Read as "0", write as "0"	R/W
b1-b0	DNF [1:0]	Selection of Filtering Capabilities for Digital Filters	00: Filter capability 1 I2C reference clock period 01: Filter capability 2 I2C reference clock period 10: Filter capability 3 I2C reference clock period 11: Filter capability 4 I2C reference clock period	R/W

29.5.15 I2CFIFO Control Register (I2C_FSTR)

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	RFST [1:0]	TFST [1:0]	NACK TFFLUSH	RFFLUSH	TFFLUSH	FEN		

Bit	Flag	Bit Name	Function	Read and write
b31~b7	Reserved	-	Read as "0", write as "0"	R/W
b7~b6	RFST [1:0]	Rx FIFO status bits	00: FIFO null 01: FIFO half full 10: FIFO full 11: reserved	R
b5~b4	TFST [1:0]	Tx FIFO status bits	00: FIFO null 01: FIFO half null 10: FIFO full 11: Reserved bit	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	R/W
b2	RFFLUSH	RX FIFO pointer reset	0: write invalid 1: Reset RX FIFO pointer	R/W
b1	TFFLUSH	TX FIFO pointer reset	0: write invalid 1: Reset TX FIFO pointer	R/W
b0	FEN	FIFO function enable bit	0: FIFO function disabled 1: FIFO Function Enable	R/W

29.5.16 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
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	SLVADR									

Bit	Flag	Bit Name	Function	Read and write
b31~b10	Reserved	-	Read as "0", write as "0"	R/W
b9~b0	SLVADR	Slave Address Receive Register	6:0: 7-bit slave receiving address 9:0: 10-bit slave receiving address	R

30 Serial Peripheral Interface (SPI)

30.1 Introduction

This product is equipped with 4-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, host/slave and baud rate as needed.

Main Features of SPI:

Table 30-1 SPI Features Key Points

Essentials	Description
Number of channels	1 channel
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 in four single frame
Baud rate	- The baud rate can be adjusted by the built-in special baud rate generator in host 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 - Odd-even 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 host 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/odd-even) - SPI idle - 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].

30.2 SPI System Block Diagram

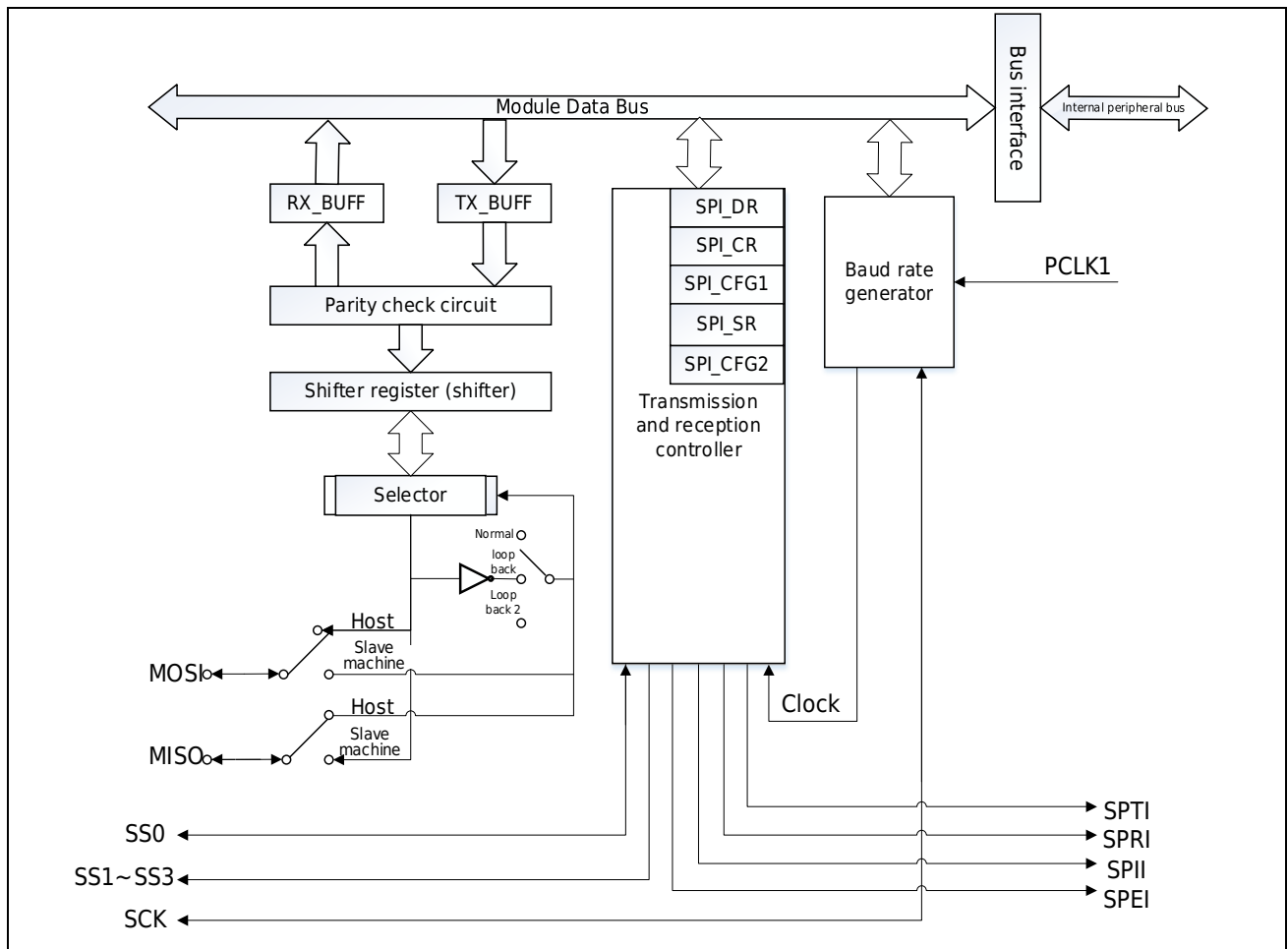


Figure 30-1 System Block Diagram

30.3 Pin Description

Table 30-2 Pin Description

Pin Name	Port Direction	Function
SCK	Input/Output	Communication clock pin
MOSI	Input/Output	Host 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

30.4 SPI Action System Description

30.4.1 Pin Status in Host Mode

When the SPI works in host mode, the status of each pin is shown in the Table 30-3 below.

Table 30-3 SPI Pin Status Description in Host Mode

Mode		Pin Name	Pin Status (PFS.ODS = 0)	Pin Status (PFS.ODS = 1)
SPI action (SPIMDS=0)	Host mode (MSTR=1, MODFE=0)	SCK	CMOS Output	OD Output
		SS0~SS3	CMOS Output	OD Output
		MOSI	CMOS Output	OD Output
		MISO	Input	Input
Clock synchronization operation (SPIMDS=1)	Host mode (MSTR=1)	SCK	CMOS Output	OD Output
		Not used	Hi-Z (available as general I/O)	Hi-Z (available as general I/O)
		MOSI	CMOS Output	OD Output
		MISO	Input	Input

Note: Please set the pin input type to CMOS input and the output to high drive mode. Please refer to [General Control Register (PCRxy)] for setting.

30.4.2 Pin State in Slave Mode

When the SPI works in slave mode, the status of each pin is shown in the Table 30-4 below.

Table 30-4 SPI Pin Status Description in Slave Mode

Mode		Pin Name	Pin Status (PFS.ODS=0)	Pin Status (PFS.ODS = 1)
SPI action (SPIMDS=0)	Slave mode (MSTR=0, MODFE=0)	SCK	Input	Input
		SS0	Input	Input
		Not used	Hi-Z (available as general I/O)	Hi-Z (available as general I/O)
		MOSI	Input	Input
		MISO (Note 2)	CMOS Output/Hi-Z	OD Output/Hi-Z
Clock synchronization operation (SPIMDS=1)	Slave mode (MSTR=0)	SCK	Input	Input
		Not used	Hi-Z (available as general I/O)	Hi-Z (available as general I/O)
		MOSI	Input	Input
		MISO	CMOS Output	OD Output

Note: Please set the pin input type to CMOS input and the output to high drive mode. Please refer to [General Control Register (PCRxy)] for setting.

30.4.3 SPI System Connection Instances

Host mode

In the host-multi-slave mode SPI system architecture, the host 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.

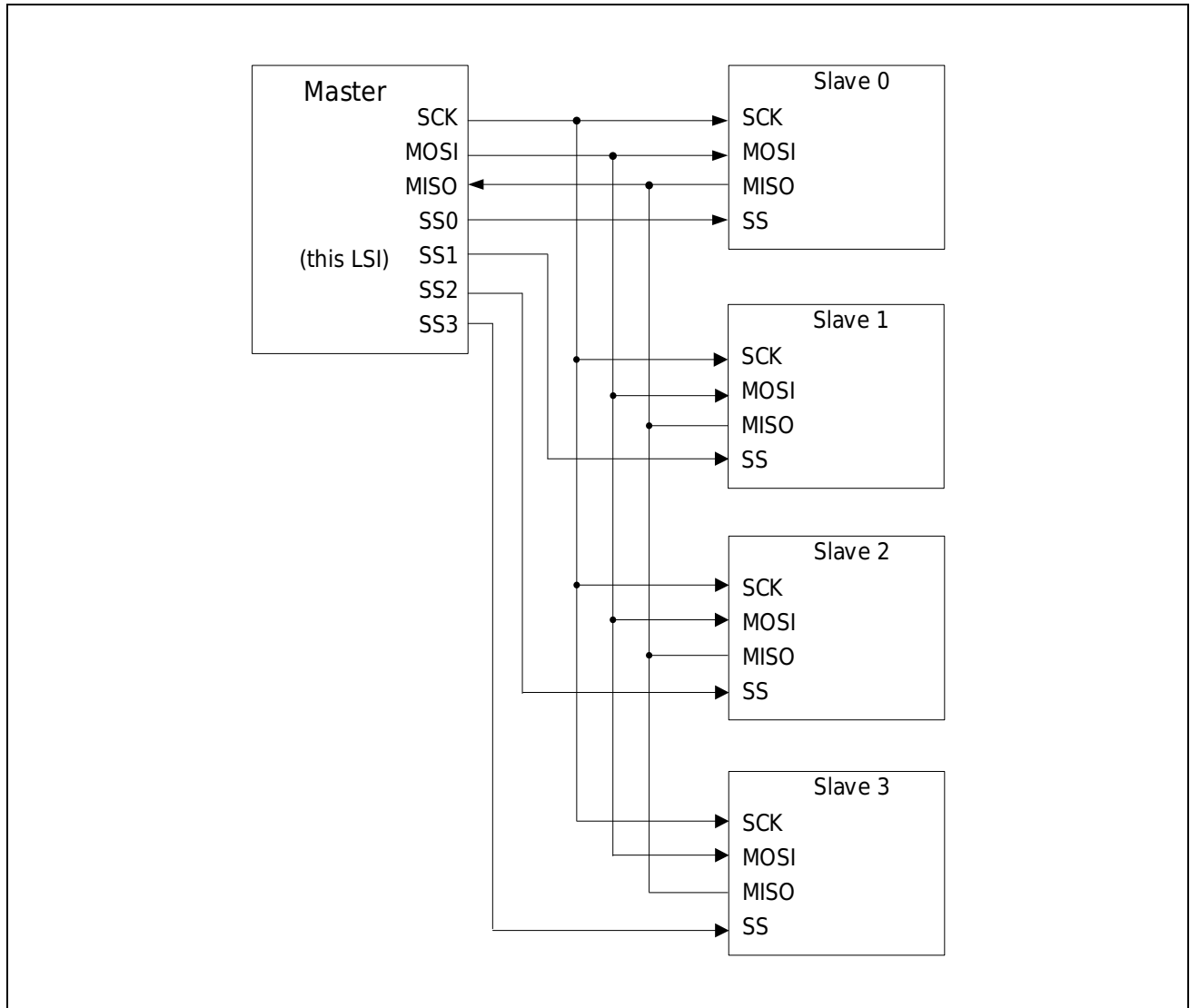


Figure 30-2 Host Mode Structure

Clock Synchronization Operation

In the SPI system structure used for clock synchronous operation, the host device drives SCK and MOSI, and the slave device drives MISO. SS pins are not used.

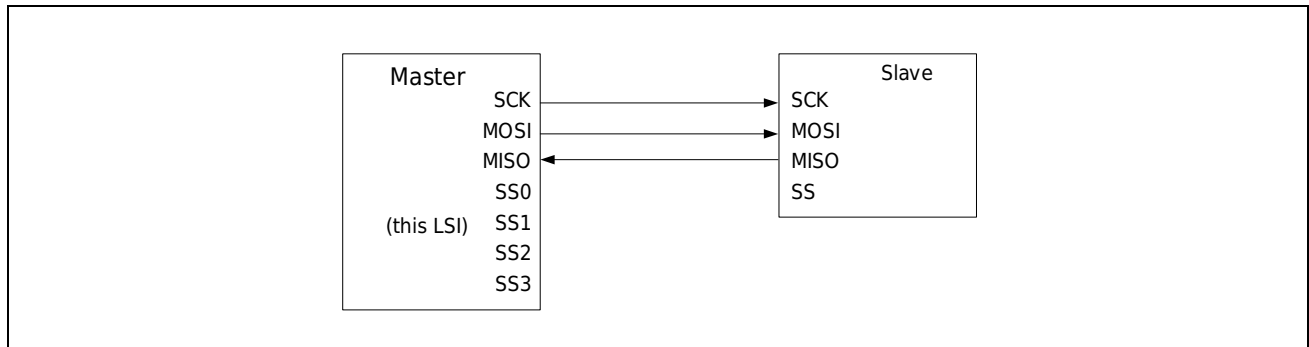


Figure 30-3 Three-wire Clock Synchronization Operation

30.5 Data Communication Description

30.5.1 Baud Rate

In host mode, SPI clock is provided by internal baud rate generator. In slave mode, the clock is input to the SCK pin.

The baud rate depends on the settings of the SPI_CFG2.MBR [1:0] bits and SPI_CFG1.CLKDIV [3:0]. 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. f_{PCLK1} This is the PCLK1 clock frequency.

$$\text{Baud rate} = \frac{f_{PCLK1}}{(M+1) \cdot 2^{N+1}}$$

Table 30-5 Some Setting Values Bit Speed

Set Value of MBR [1: 0] Bit	Set Value of CLKDIV [3: 0] Bit	Frequency Division Ratio	Baud Rate			
			PCLK1=5MHz	PCLK1=10MHz	PCLK1=20MHz	PCLK1=40MHz
0	0	2	2.50Mbps	5.00Mbps	10.0Mbps	20.0Mbps
0	1	4	1.25Mbps	2.50Mbps	5.00Mbps	10.0Mbps
0	3	8	625kbps	1.25Mbps	2.50Mbps	5.00Mbps
0	7	16	313kbps	625kbps	1.25Mbps	2.50Mbps
0	15	32	156kbps	313kbps	625kbps	1.25Mbps
1	15	64	78kbps	156kbps	313kbps	625kbps
2	15	128	39kbps	78kbps	156kbps	313kbps
3	15	256	20kbps	39kbps	78kbps	156kbps

30.5.2 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 30-4.

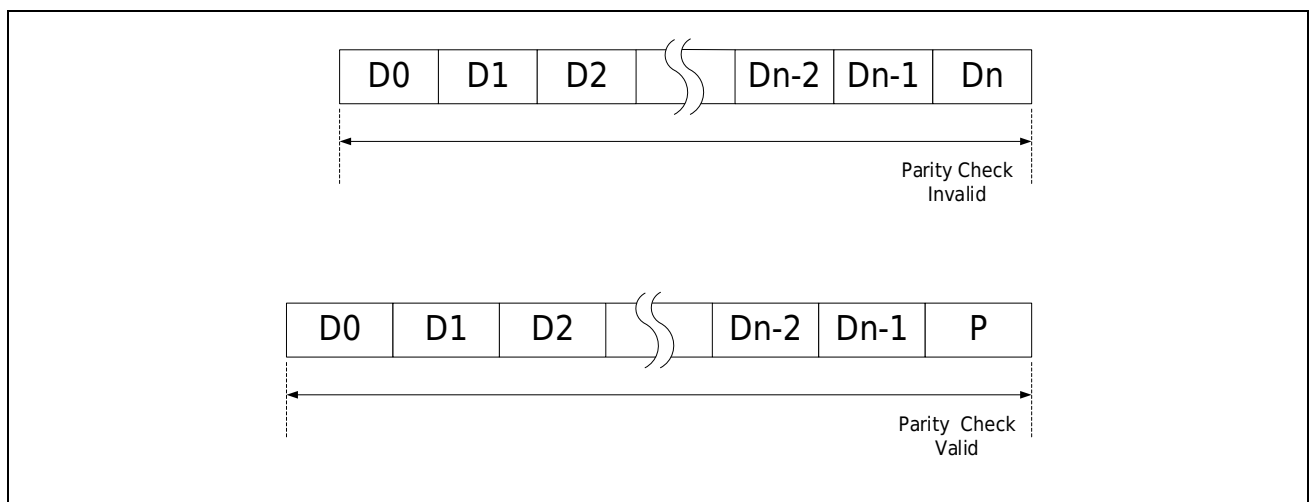


Figure 30-4 Data Format

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:

- 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.

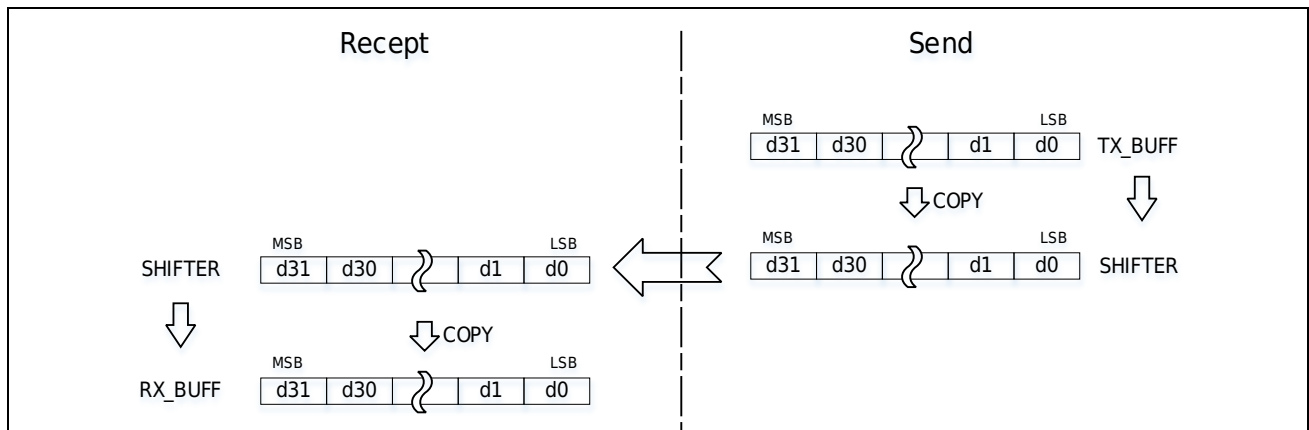


Figure 30-5 MSB sent First, Parity Check Invalid

■ 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.

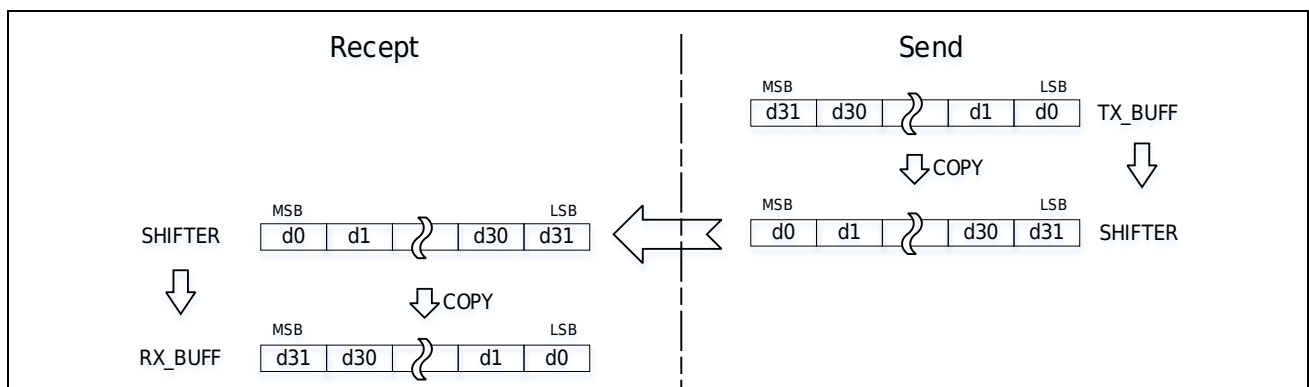


Figure 30-6 LSB sent First, Parity Check Invalid

■ 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.

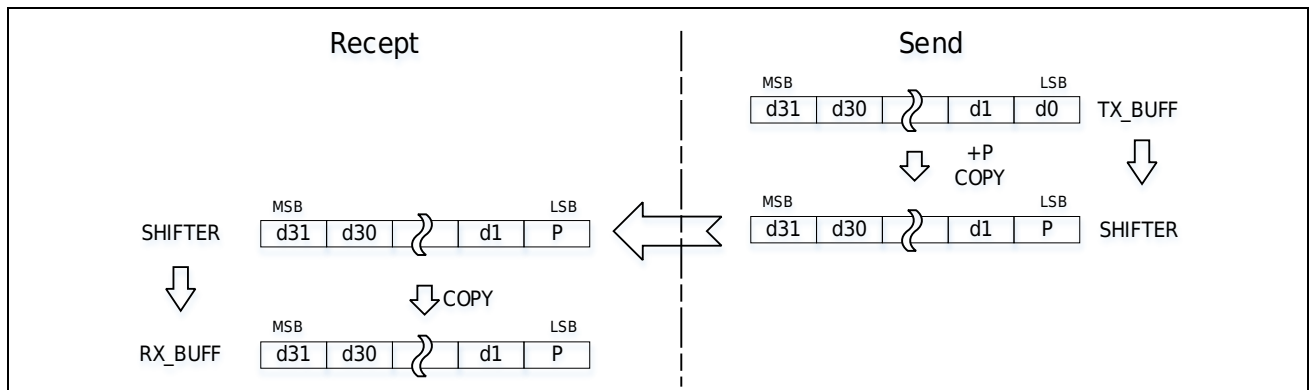


Figure 30-7 MSB sent First, when parity check valid

■ 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.

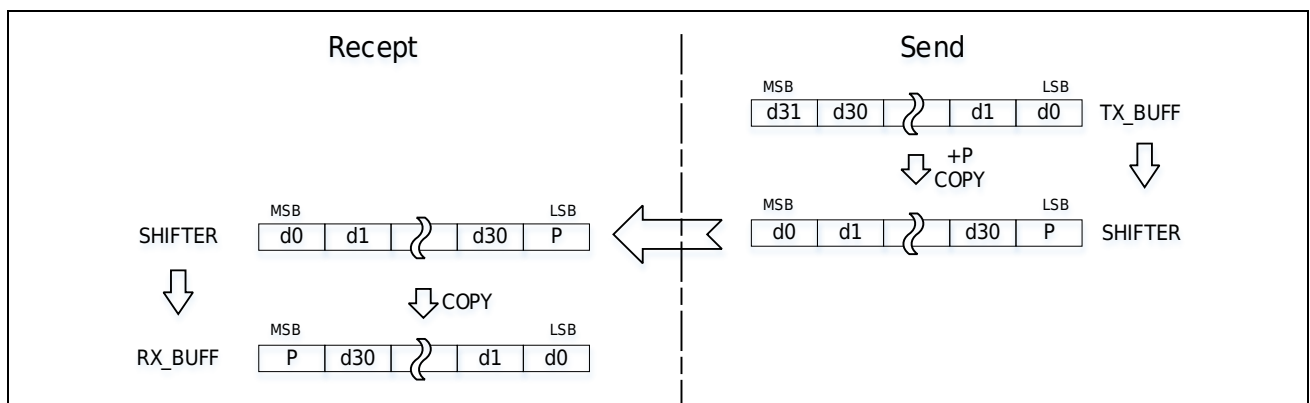


Figure 30-8 LSB sent First, when parity check valid

30.5.3 Transfer format

■ 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 30-9 It 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 period. MOSI/MISO signals are updated at 1/2 SCK periods after each sampling. The set value of CPOL bit does not affect the running time of SCK signal, but only affects the polarity of signal.

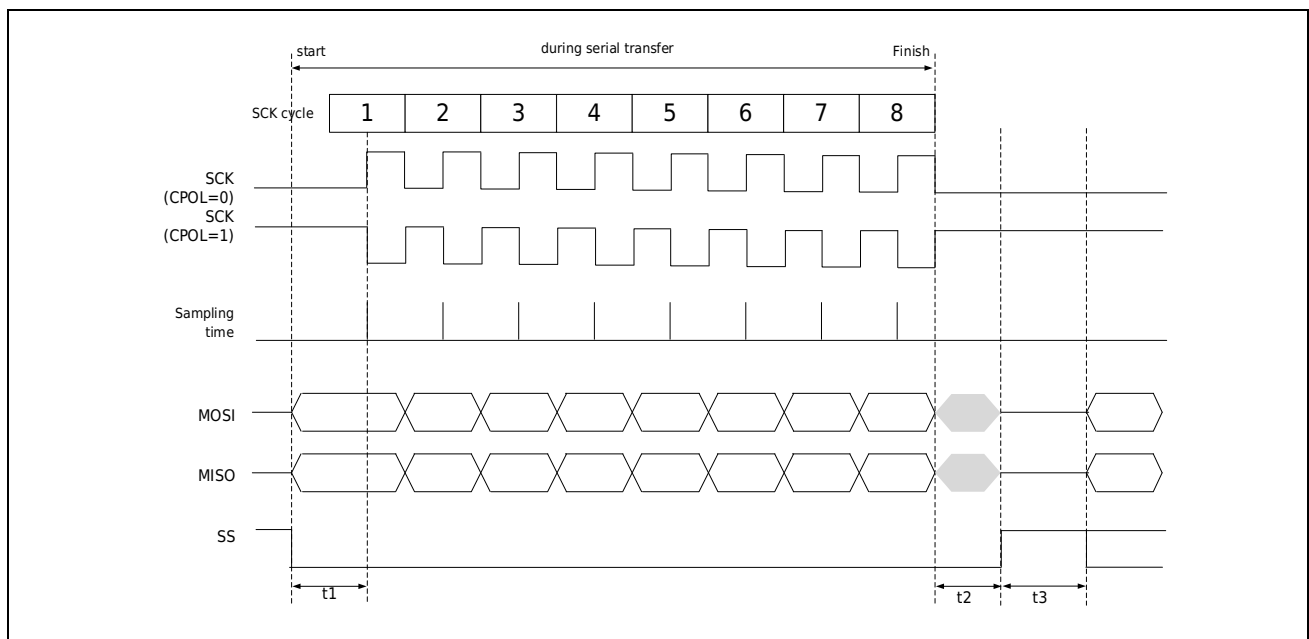


Figure 30-9 Data Transfer Format Diagram (CPHA = 0)

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 host device on the SPI system. For details, please refer to the chapter [Host Action in SPI Operation Mode].

■ CPHA = 1

When SPI_CFG2.When the CPHA bit is "1", SPI updates the data at the odd edge of SCK and samples the data at the even edge. Figure 30-10 Is the transport Sequence Diagram of SPI when CPHA = 1. MOSI/MISO starts to transmit data updates at the edge of the first SCK signal after the SSi signal becomes active. After this, data is updated every SCK period. Data is

sampled at 1/2 SCK periods after each update. SPI_CFG2.CPOL The set value of CPOL bit does not affect the running time of SCK signal, but only affects the polarity of signal.

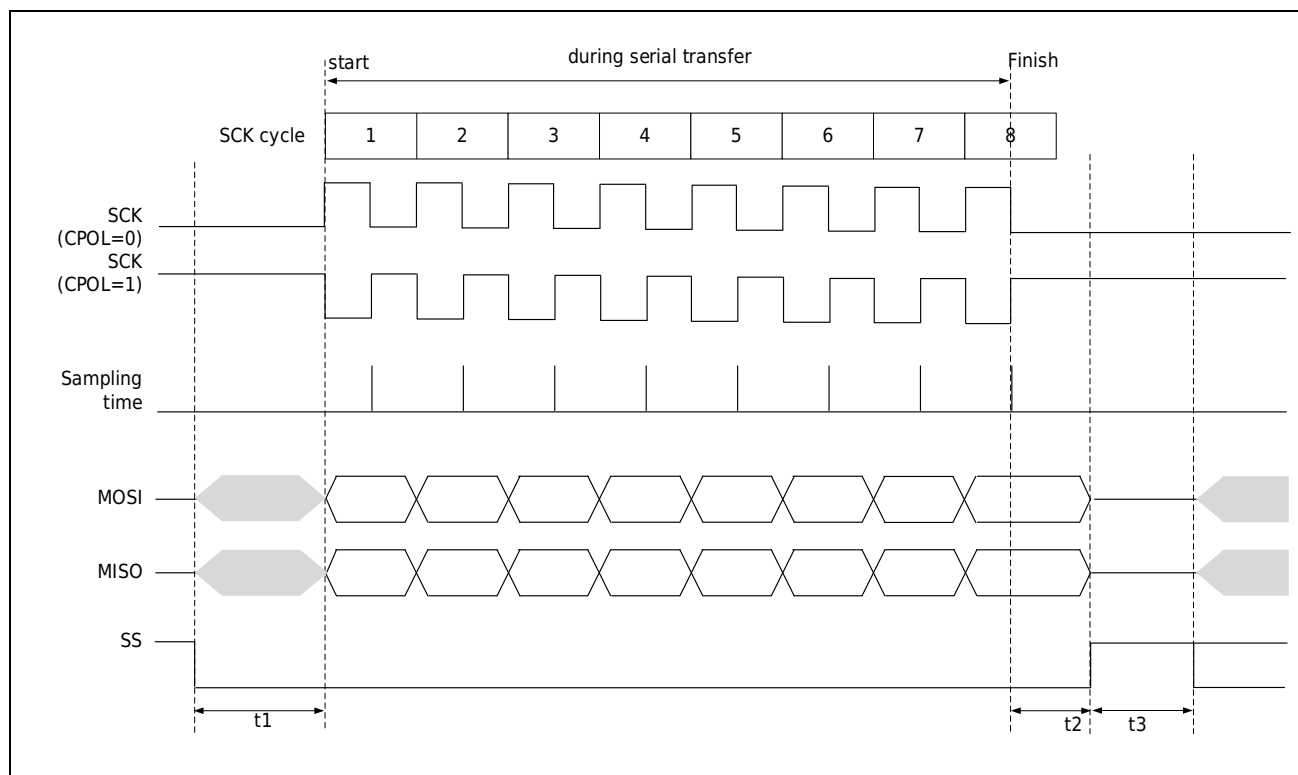


Figure 30-10 Data Transfer Format (CPHA = 1)

The same is true for t1, t2, t3 and CPHA bits.

30.5.4 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).

■ 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 Figure 30-11 shown, the SPI_CFG1.FTHLV [1:0] bits are "00b", the SPI_CFG2.CPHA bits are "1" and the SPI_CFG2.CPOL bits are "0", and the SPI performs 8-bit serial transfers.

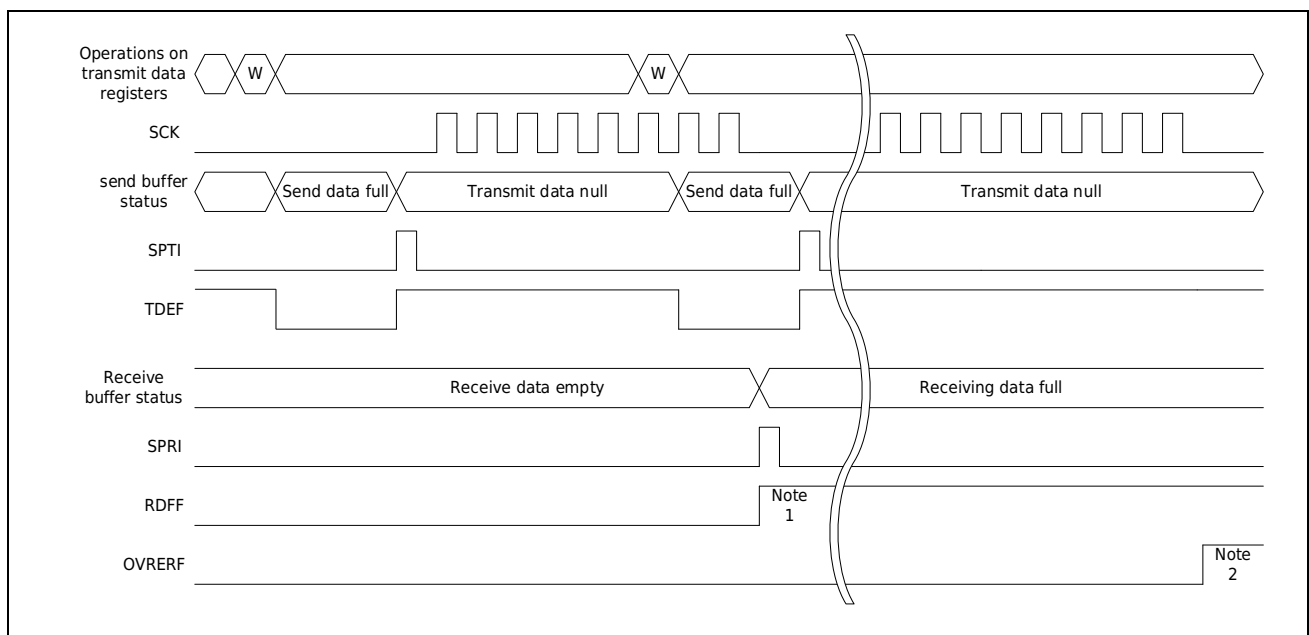


Figure 30-11 Full Duplex Synchronous Serial Communication

Note:

- 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.
- 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.

■ Send only communication method

When the SPI_CR.TXMDS bit is 1, the SPI operates in send-only communication mode. As Figure 30-12 shown, the SPI_CFG1.FTHLV [1:0] bits are "00b", the SPI_CFG2.CPHA bits are "1" and the SPI_CFG2.CPOL bits are "0", and the SPI performs 8-bit serial transfers.

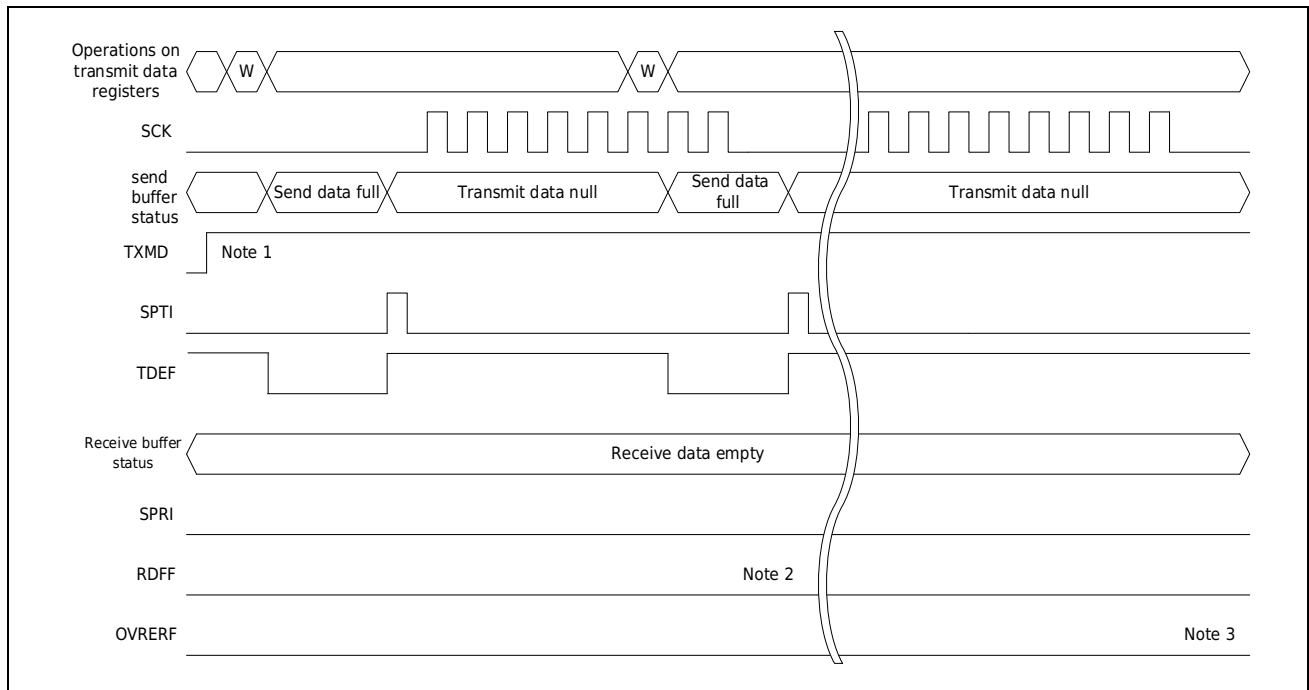


Figure 30-12 Sending communication only

Note:

- 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.
- 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.
- 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.

30.5.5 Communication Interval Time

■ 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 $1 \times \text{SCK} + 2 \times \text{PCLK1}$. As shown in Figure 30-13, the SPI_CFG2.CPHA bit is 1 and the SPI_CFG2.CPOL bit is 0, and the SPI performs serial transfer.

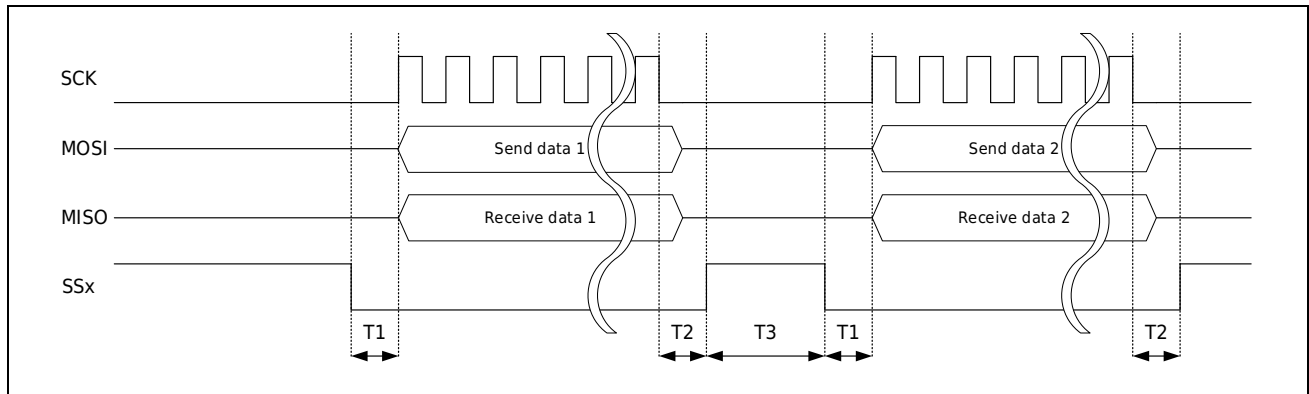


Figure 30-13 General Communication

In the above figure, t_1 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), t_2 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 t_3 represents the minimum waiting time from the end of this transmission to the beginning of the next transmission (the next access data delay, which can be controlled by setting SPI_CFG1.MIDI [2:0] and SPI_CFG2.MIDIE).

■ 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 30-14, the SPI_CFG2.CPHA bit is 1 and the SPI_CFG2.CPOL bit is 0, and the SPI performs serial transmission.

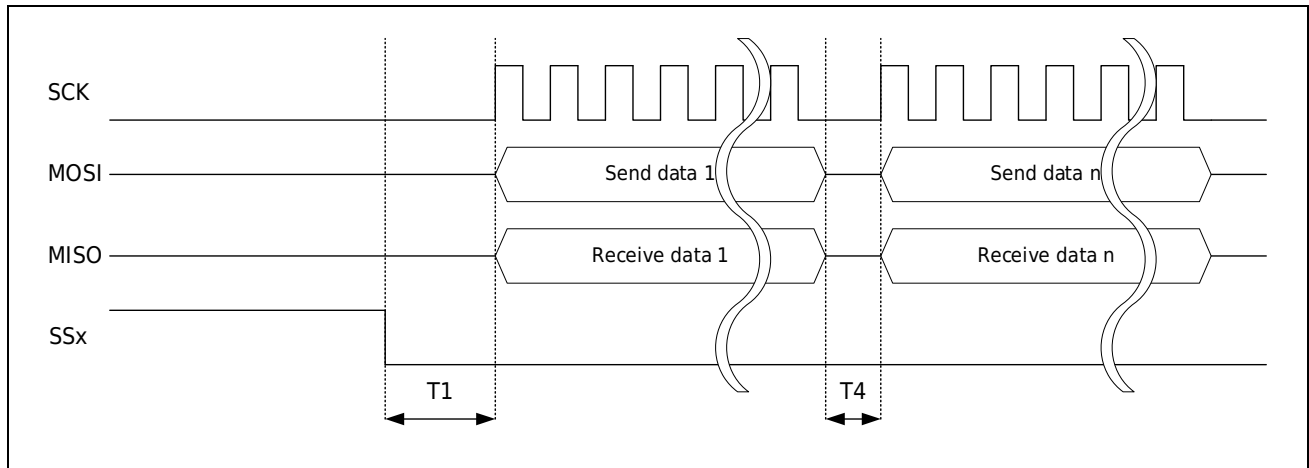


Figure 30-14 Continuous Communication

In the above figure, t_1 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 t_4 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 t_1 and t_4 is defined as $t_4 = t_1 - 1 * SCK$. **Note:**

1. In continuous communication mode, if the host 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 host 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.

30.6 Operating Instructions

30.6.1 Operational Mode Summary

This SPI supports 4-wire SPI mode and 3-wire clock synchronization mode. Serial communication can be performed as a host 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 30-6.

Table 30-6 SPI Pattern and Register Setting Relationship

Mode	Host (SPI operation)	Slave (SPI operation)	Host (Clock synchronous operation)	Slave (Clock synchronous operation)
MSTR bit setting	1	0	1	0
SPIMDS bit setting	0	0	1	1
SCK signal	Output	Input	Output	Input
MOSI signal	Output	Input	Output	Input
MISO signal	Input	Output/Hi-Z	Input	Output
SS0 signal	Output	Input	Hi-Z (not used)	Hi-Z (not used)
SS1~SS3 signal	Output	Hi-Z	Hi-Z (not used)	Hi-Z (not used)
SS Polarity Change Function	Yes	Yes	-	-
Maximum transmission rate	~PCLK1/2	~PCLK1/6	~PCLK1/2	~PCLK1/6
Clock source	Internal baud rate generator	SCK input	Internal baud rate generator	SCK input
Clock polarity	2 species	2 species	2 species	2 species
Clock phase	2 species	2 species	2 species	1 species (CHPA = 1)
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	Yes	No	Yes	No

Note:

- When the SPI_CR.TXMDS bit is 1, the Rx Buffer full detection, overrun error detection and parity check error detection are not performed.
- When the SPI_CR.PAE bit is 0, no parity check error detection is performed.

30.6.2 Host Action in SPI Operation Mode

■ Action Description for as Host

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 host 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 more information about SPI transport format, refer to [Transfer format].

■ Initialization of SPI Host 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.

30.6.3 The Slave Action in SPI Operation Mode

■ SPI as Slave Action Description

When SPI_CFG2.CPHA bit is 0, if SPI detects that the SS0 input signal becomes active, it needs to start to output the signal to MISO to drive the active data. Therefore, when the CPHA bit is 0, the SS0 input signal level is changed from 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 more information about SPI transport format, refer to [Transfer format].

Note:

- When SPI_CFG2.CPHA bit is "0", the SS0 input signal level is changed from invalid to valid 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.

■ 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.

30.6.4 Host 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.

■ Action Description for SPI as Host

When the SPI data transmission buffer register (TX_BUFF) is empty (the TDEF flag in the status register SPI_SR is 0), write the data with 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 timing depends on the bit length of the sent data. The SPI data length in host mode depends on the setting value of the SPI_CFG2.DSIZE [3:0] bit. For more information about SPI transport format, refer to [Transfer format].

■ Initialization setting of host 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.

30.6.5 Slave Maneuver in Clock Synchronous Operation Mode

■ SPI as Slave Action Description

When SPI_CFG2.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 clock synchronization mode, normal communication cannot be performed when the CPHA bit is 0.

When the CPHA bit is "1", if SPI detects the original SCK edge when the SS0 input signal is at an active level, it needs to start to drive 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.

■ 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.

30.6.6 Processing FLOW OF SEVERAL SPI ACTIONS

- SPI data transfer process as host
 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 until the last data was 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. Send SPI idle state interrupt.
 6. Set SPE to 0, stop SPI, and reset IDIE.
- 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 step (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.
- 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.

30.7 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.

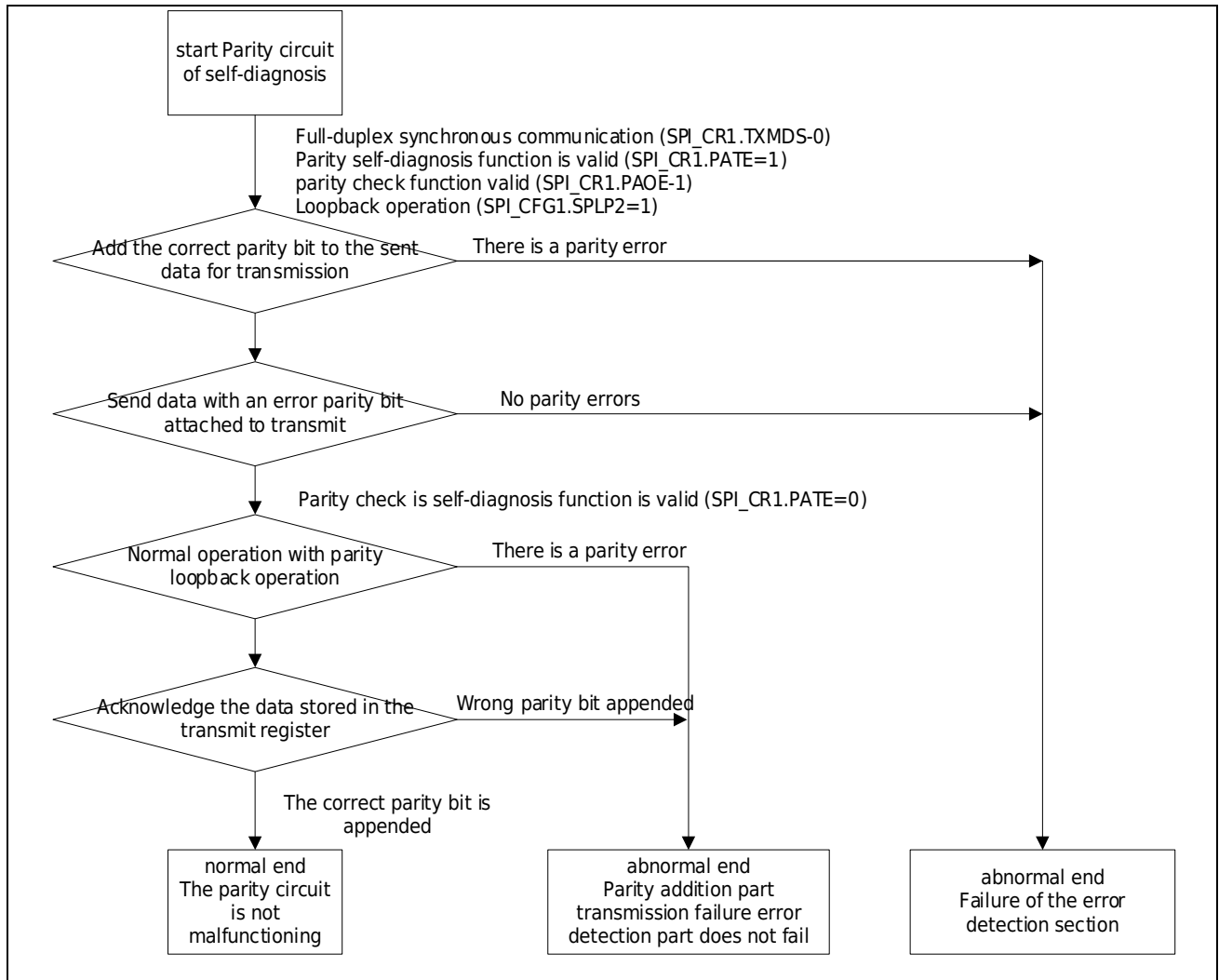


Figure 30-15 Parity Checking Process

30.8 Error Detection

In the normal SPI serial transmission, the system sends the data serially to SPI _ DRregister and obtains the data serially received by reading SPI _ DRregister. 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 delivery and SPI error detection is shown Table 30-7 below.

Table 30-7 Correspondence Table for Error Detection

SN	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.

30.8.1 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.

30.8.2 Pattern Error

When SPI is in host 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.MODFERF and then set SPI_CR.SPE to "1".

30.8.3 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 written to "0" after the SPI_SR register is read when the OVRERF flag is 1.

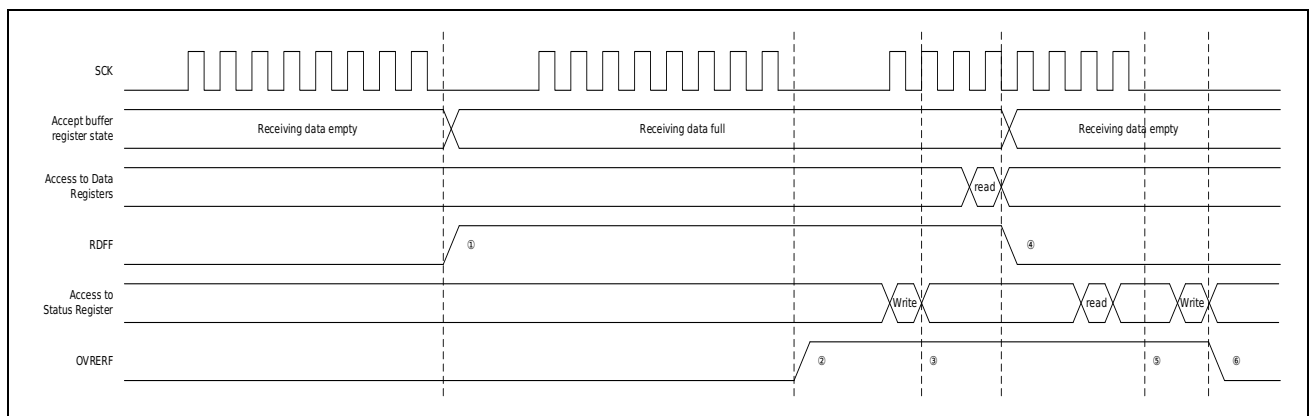


Figure 30-16 Overload Error Handling

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 When the receive buffer is full, the serial transmission ends and the SPI detects an overrun error and sets the OVRERF flag to "1". 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. The occurrence of parity errors must be Detect as soon as possible by means of read the of transmissions DR register SPI _SR.

In host 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 Figure 30-17 and Figure 30-18.

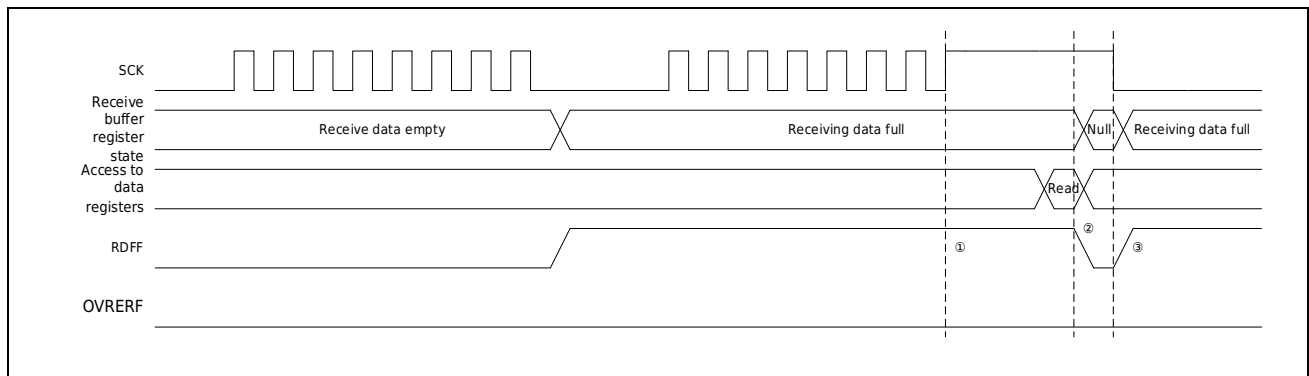


Figure 30-17 Schematic diagram of the Action when the Clock Automatic Stop Function is Enabled (CPHA=1)

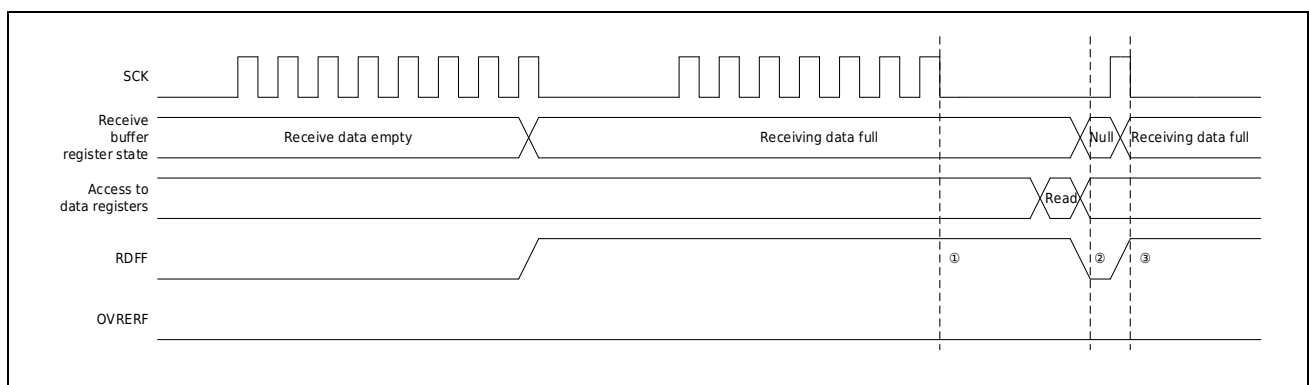


Figure 30-18 Schematic diagram of the Action when the Clock Automatic Stop Function is Enabled (CPHA=0)

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.

30.8.4 Parity Check 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_SR register with PERF flag “1” before writing 0 to the PERF flag.

Instances of running OVRERF flags and PERF flags are shown in Figure 30-19 below. In the instance 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”.

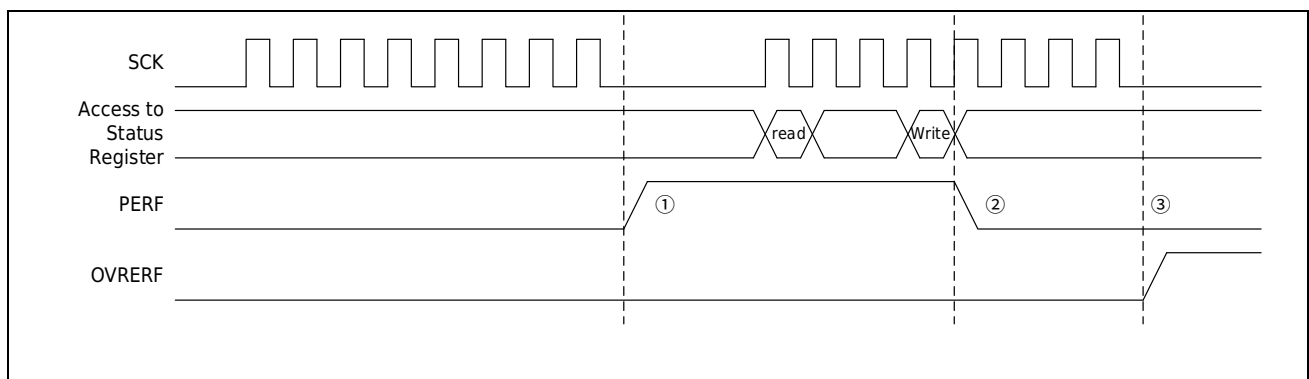


Figure 30-19 Parity Error

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 register SPI _ SR.

30.9 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.

30.9.1 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.

30.9.2 System Reset Initialization

All SPI control bits, status bits and data registers are initialized by system reset.

30.10 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 30-8. 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 30-8 SPI interrupt Source Description

Interrupt/Internal Trigger Source	Abbreviation	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" Host 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.

30.11 Available Event Trigger Sources

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)].

30.12 Register Description

Table 30-9 Register List

Register base address: SPI1_BASE: 0x4001C000; SPI2_BASE: 0x4001C400

SPI3_BASE: 0x4001C800; SPI4_BASE: 0x40020000

Register Name	Offset Address	Reset Value
SPI Data Register SPI_DR	0x00	0x0000 0000
SPI Control Register (SPI_CR)	0x04	0x0000 0000
SPI Communication Configuration Register1 SPI_CFG1	0x0C	0x0000 0010
SPI Status Register SPI_SR	0x14	0x0000 0020
SPI Communication Configuration Register2 SPI_CFG2	0x18	0x0000 0F1D

30.12.1 SPI Data Register (SPI_DR)

Reset value: 0x0000

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	Flag	Bit Name	Function											Read and write	
b31~b0	SPD [31:0]	Serial data	SPI Data Storage											R/W	

30.12.2 SPI Control Register (SPI_CR)

Reset value: 0x0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	—	TXMDS	SPIMDS

Bit	Flag	Bit Name	Function	Read and Write
b31~b16	Reserved	-	Read as "0", write as "0"	R/W
b15	PAE	Parity check enable	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)	R/W
b14	PAOE	Parity mode selection	0: Select parity for sending and receiving 1: Select odd check for sending and receiving	R/W
b13	PATE	Parity self-diagnosis	0: The parity self-diagnostic function is invalid 1: Parity self-diagnosis function is valid	R/W
b12	MODFE	Mode fault error detection permit	0: Disable mode fault error detection 1: Allow mode fault error detection	R/W
b11	IDIE	SPI Idle Interrupt Enable	0: Disable idle interrupt request generation 1: Enable idle interrupt request generation	R/W
b10	RXIE	SPI Receive Interrupt Permit	0: Disable SPI from receiving interrupt request generation 1: Enable SPI from sending interrupt request generation	R/W
b9	TXIE	SPI Send Interrupt Permit	0: Disable SPI from sending interrupt request generation 1: Enable SPI from sending interrupt request generation	R/W
b8	EIE	SPI error interrupt disable	0: Disable SPI error interrupt request generation 1: Enable SPI error interrupt request generation	R/W
b7	CSUSPE	Communication auto suspend function enable	0: Communication auto suspend function invalid 1: Communication auto-suspend function valid	R/W
b6	SPE	SPI Feature Enable	0: Invalid SPI feature 1: SPI function valid	R/W
b5	SPLPBK2	SPI Loop 2-bit	0: Normal mode 1: Loopback mode (send data = receive data)	R/W
b4	SPLPBK	SPI loopback position	0: Normal mode 1: Loopback mode (inverse of transmitted data = received data)	R/W
b3	MSTR	SPI Host-slave mode selection	0: Slave mode 1: Host mode	R/W
b2	Reserved	-	Read as "0", write as "0"	R/W
b1	TXMDS	Communication mode selection	0: Full-duplex synchronous serial communication 1: Send serial communication only	R/W
b0	SPIMDS	SPI mode selection	0: SPI operation (4-line) 1: Clock synchronous operation (3-wire)	R/W

30.12.3 SPI Communication Configuration Register1 (SPI _ CFG1)

Reset value: 0x0010

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	MIDI [2:0]			-	MSSDL [2:0]			-	MSSI [2:0]			-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CLKDIV [3:0]				SS3PV	SS2PV	SS1PV	SS0PV	-	SPRDTD	-	-	-	CTMDS	FTHLV [1:0]	

Bit	Flag	Bit Name	Function	Read and Write
b31	Reserved	-	Read as "0", write as "0"	R/W
b30~b28	MIDI [2:0]	Host next access data interval idle time setting bit	b30~b28 0 0 0: 1 SCKs + 2 PCLK1s 0 1 1: 2 SCKs + 2 PCLK1s 0 1 0: 3 SCKs + 2 PCLK1s 0 1 1: 4 SCKs + 2 PCLK1s 1 0 0: 5 SCKs + 2 PCLK1s 1 0 1: 6 SCKs + 2 PCLK1s 0 1 0: 7 SCKs + 2 PCLK1s 1 1 0: 8 SCKs + 2 PCLK1s	R/W
b27	Reserved	-	Read as "0", write as "0"	R/W
b26~b24	MSSDL [2:0]	Master SS invalid delay setting bit	b26~b24 0 0 0: 1 SCK 0 0 1: 2 SCKs 0 1 0: 3 SCKs 0 1 1: 4 SCKs 1 0 0: 5 SCKs 1 0 1: 6 SCKs 1 1 0: 7 SCKs 1 1 1: 8 SCKs	R/W
b23	Reserved	-	Read as "0", write as "0"	R/W
b22~b20	MSSI [2:0]	Host SS idle time setting bit	b22~b20 0 0 0: 1 SCK 0 0 1: 2 SCKs 0 1 0: 3 SCKs 0 1 1: 4 SCKs 1 0 0: 5 SCKs 1 0 1: 6 SCKs 1 1 0: 7 SCKs 1 1 1: 8 SCKs	R/W
b19~b16	Reserved	-	Read as "0", write as "0"	R/W
b15~b12	CLKDIV	Communication reference clock selection bit	b15~b12 0 0 0 0: pclk/2 0 0 0 1: pclk/4 0 0 1 0: pclk/6 0 0 1 1: pclk/8 0 1 0 0: pclk/10 0 1 0 1: pclk/12 0 1 1 0: pclk/14 0 1 1 1: pclk/16 1 0 0 0: pclk/18 1 0 0 1: pclk/20 1 0 1 0: pclk/22 1 0 1 1: pclk/24 1 1 0 0: pclk/26 1 1 0 1: pclk/28 1 1 1 0: pclk/30 1 1 1 1: pclk/32	R/W
b11	SS3PV	SS3 signal polarity set	0: Low level of SS3 signal valid 1: High level of SS3 signal valid	R/W
b10	SS2PV	SS2 signal polarity set	0: Low level of SS2 signal valid 1: High level of SS2 signal valid	R/W
b9	SS1PV	SS1 signal polarity set	0: Low level of SS1 signal valid 1: High level of SS1 signal valid	R/W
b8	SS0PV	SS0 signal Polarity setting	0: Low level of SS0 signal valid 1: High level of SS0 signal valid	R/W

b7	Reserved	-	Read as "0", write as "0"	R/W
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)	R/W
b5	Reserved	-	Read as "0", write as "0"	R/W
b4	Reserved	-	Read as "1", write as "1"	R/W
b3	Reserved	-	Read as "0", write as "0"	R/W
b2	CTMDS	Continuous communication mode selection	0: Normal communication mode 1: Continuous communication mode	R/W
b1~b0	FTHLV [1:0]	FPS	b1~b0 0 0: 1 frame 0 1: 2 frames 1 0: 3 Frame 1 1: 4 Frame	R/W

30.12.4 SPI Status Register (SPI_SR)

Reset value: 0x0020

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	RDF	—	TDEF	UDRERF	PERF	MODFERF	IDLNF	OVRERF

Bit	Flag	Bit Name	Function	Read and Write
b31~b8	Reserved	-	Read as "0", write as "0"	R/W
b7	RDF	Receive buffer full flag	0: The Rx Buffer is not full. 1: The Rx Buffer is full. Hardware set, clear	R
b6	Reserved	-	Read as "0", write as "0"	R/W
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	Underload error flag	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.	R/W
b3	PERF	Parity check error flag	0: No parity error occurred 1: Parity error Hardware setting alignment After reading 1 0 and reset	R/W
b2	MODFERF	Mode fault and error flag	0: Unsent mode failure error 1: Mode failure error Hardware setting alignment After reading 1 0 and reset	R/W
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	R/W

30.12.5 SPI Communication Configuration Register2 (SPI _ CFG2)

Reset value: 0x0F1D

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
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]			-	MBR [1:0]		CPOL	CPHA

Bit	Flag	Bit Name	Function	Read and Write
b31~b16	Reserved	-	Read as "0", write as "0"	R/W
b15	MSSIE	SCK delay allows	0: SCK delay is 1 SCK 1: SCK delay is the set value of SSI	R/W
b14	MSSDLE	SS Invalid delay allows	0: SS invalid delay is 1 SCK 1: SS Invalid delay is the set value of SSDL	R/W
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	R/W
b12	LSBF	SPI LSB first bit	0: MSB first 1: LSB first	R/W
b11~b8	DSIZE [3: 0]	Positioning SPI Data Length	b11~b8 0 0 0 0: 4 bits 0 0 0 1: 5 bits 0 0 1 0: 6 bits 0 0 1 1: 7 bits 0 1 0 0: 8 bits 0 1 0 1: 9 bits 0 1 1 0: 10 bits 0 1 1 1: 11 bits 1 0 0 0: 12 bits 1 0 0 1: 13 bits 1 0 1 0: 14 bits 1 0 1 1: 15 bits 1 1 0 0: 16 bits 1 1 0 1: 20 bits 1 1 1 0: 24 bits 1 1 1 1: 32 bits	R/W
b7~b5	SSA [2: 0]	SS signal valid setting bit	b7~b5 0 0 0: SS0 0 0 1: SS1 0 1 0: SS2 0 1 1: SS3 1 x x: Setting prohibited	R/W
b4	Reserved	-	Read as "1", write as "1"	R/W
b3~b2	MBR [1: 0]	Bit Rate Frequency Division Location	b3~b2 0 0: Select the reference communication clock 0 1: Select the 2-division of the reference communication clock 1 0: Select the 4-division of the reference communication clock 1 1: Select the 8-division of the reference communication clock	R/W
b1	CPOL	SCK polarity set bit	0: Low SCK at idle 1: SCK is high when idle	R/W
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	R/W

31 Quad-wire Serial Peripheral Interface (QSPI)

31.1 Introduction

The Quad Serial Peripheral Interface (QSPI) is a memory control module designed to communicate with serial ROMs with an SPI-compatible interface. Its objects mainly include serial flash memory, serial EEPROM and serial FeRAM.

Table 31-1 QSPI main Specifications

Parameter	Specifications
Number of channels	1 channel
SPI	- Support multiple protocols such as extended SPI, two-wire SPI and four-wire SPI - SPI mode 0 and SPI mode 3 are supported. - Address line width can be selected from 8 bits/16 bits/24 bits/32 bits
timing adjustment	Timing can be adjusted to support various serial flash memories
flash read	- Support multiple reading methods a. Standard Read/Fast Read b. Two-wire output fast read/two-wire input and output fast read c. Two-wire output fast read/two-wire input and output fast read - Free setting command - Adjustable number of dummy periods 16-byte read-ahead function - status query function - SPI bus period extension function - XIP control function
direct communication function	Flexible and extensive support for a large number of serial flash software control commands, including erasing, writing, ID reading and power-down control, etc.
Interrupt source	QSPI bus error interrupt
Module stop function	Power consumption can be reduced by stopping the module

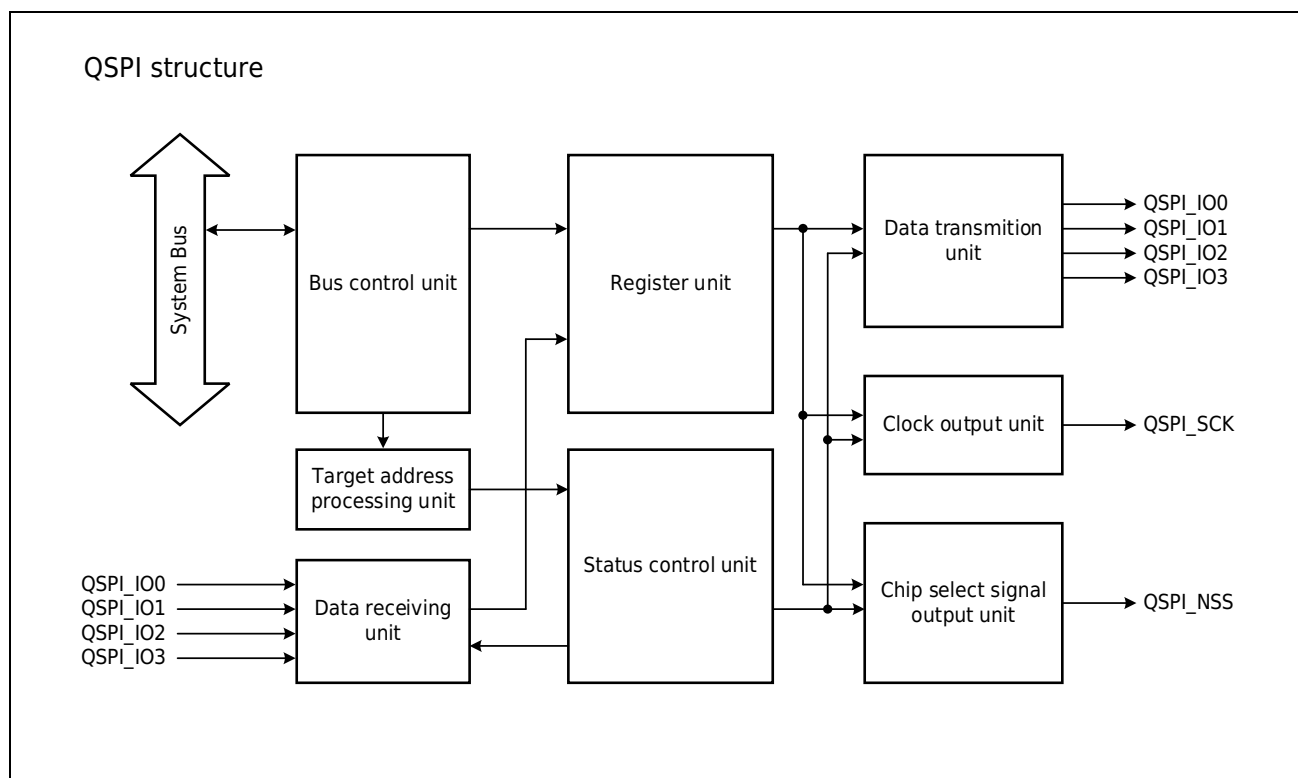


Figure 31-1 Module Composition Diagram of QSPI

Table 31-2 QSPI Pin

Pin Name	Input/Output	Functional Description
QSPI_SCK	Output	QSPI clock output pin
QSPI_NSS	Output	QSPI Select pin from machine
QSPI_IO0	Input/Output	Data line 0
QSPI_IO1	Input/Output	Data line 1
QSPI_IO2	Input/Output	Data line 2
QSPI_IO3	Input/Output	Data line 3

Note:

QSPI_SCK , QSPI_NSS , QSPI_IO0 , QSPI_IO1 , QSPI_IO2 , and QSPI_IO3 are hereinafter referred to as QSCK , QSSN , QIO0 , QIO1 , QIO2 , and QIO3 respectively.

31.2 Memory Map

31.2.1 Internal Bus Space

The location of the serial flash memory and related control registers in the AHB bus space is determined by the overall address range configuration.

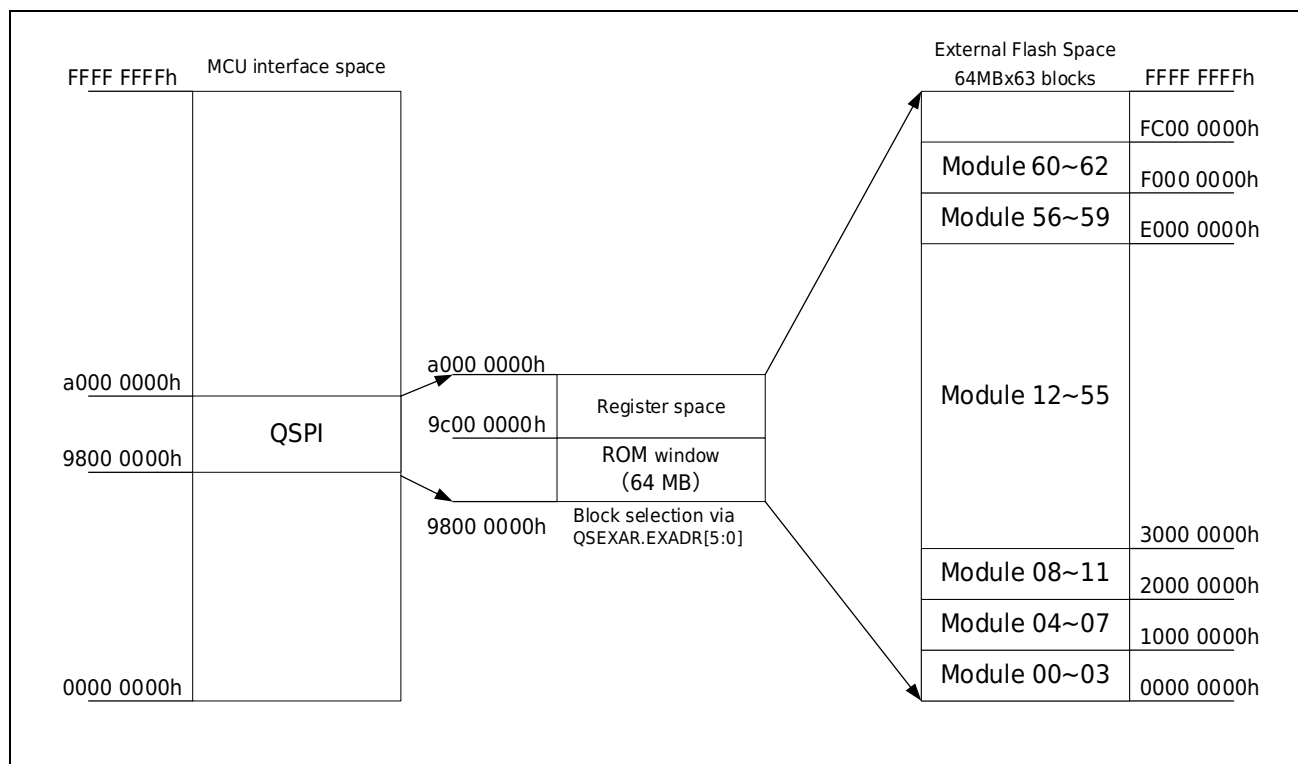


Figure 31-2 Default Area Setting and AHB Bus Space Memory Mapping Relationship Diagram

31.3 QSPI Bus

31.3.1 SPI Protocol

this QSPI Support three ways to as extended wire SPI, two-wire SPI and four-wire SPI. The initial default protocol is the extended SPI protocol. The protocol at each stage can be configured separately by setting the DPRSL [1:0]/APRSL [1:0]/IPRSL [1:0] bits in the QSPI_CR register. The extended SPI protocol only uses the QSIO0 pin single-wire for instruction output, and the subsequent addresses and data are output in single-wire/two-wire/four-wire according to the specific read mode instructions.

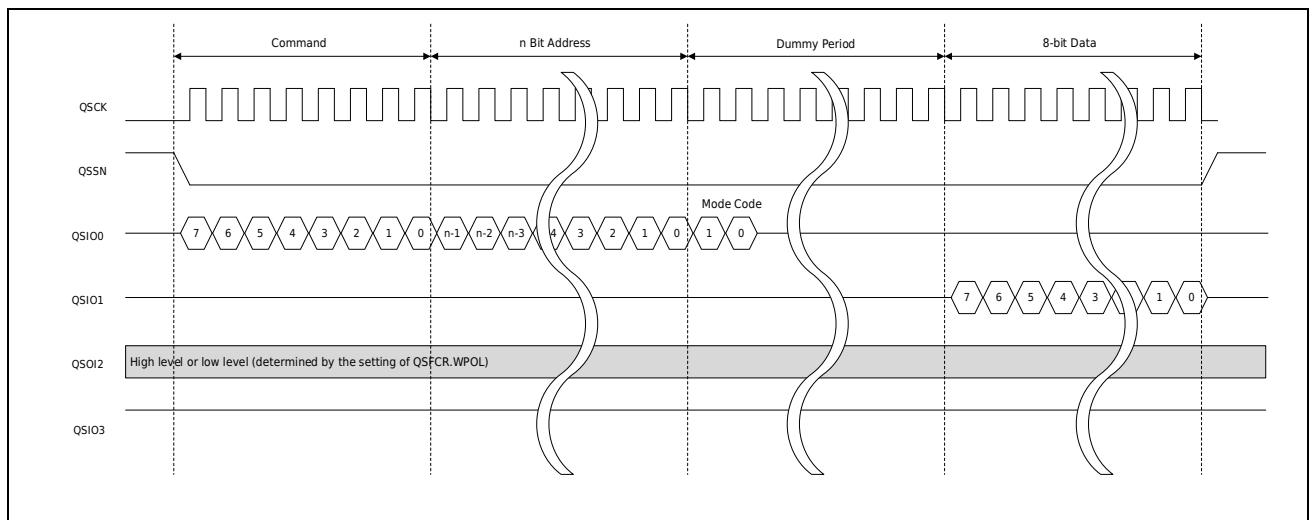


Figure 31-4 Schematic Diagram of Extended SPI Protocol Action 1 (Fast Read Mode)

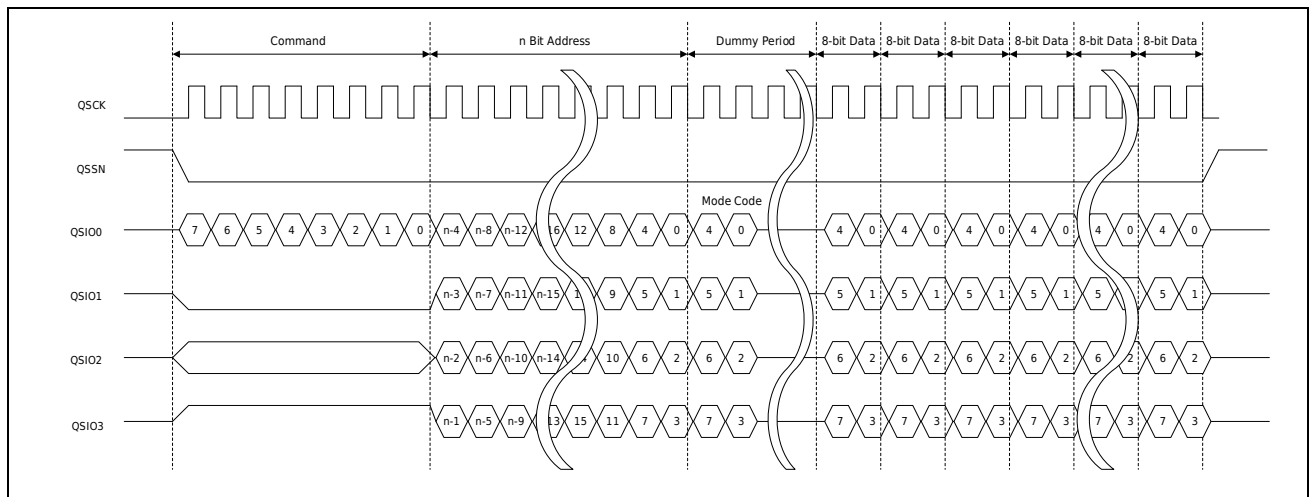


Figure 31-5 Schematic Diagram of Extended SPI Protocol Action 2 (Four-Wire Input and Output Fast Read Mode)

The two-wire SPI protocol uses the two pins QSI00 and QSI01 to perform corresponding operations, including sending instructions, addresses, receiving data, and so on.

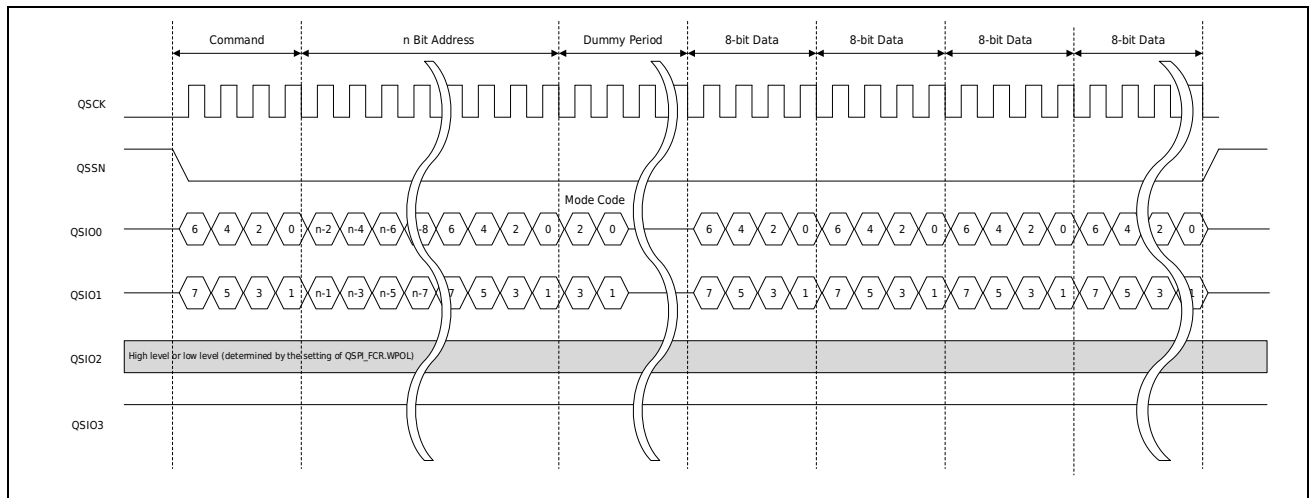


Figure 31-6 Schematic Diagram of Two-wire SPI Protocol Action (Fast Read Mode)

The four-wire SPI protocol uses four pins QSI00, QSI01, QSI02, and QSI03 to perform all related operations such as sending instructions, addressing, and receiving data.

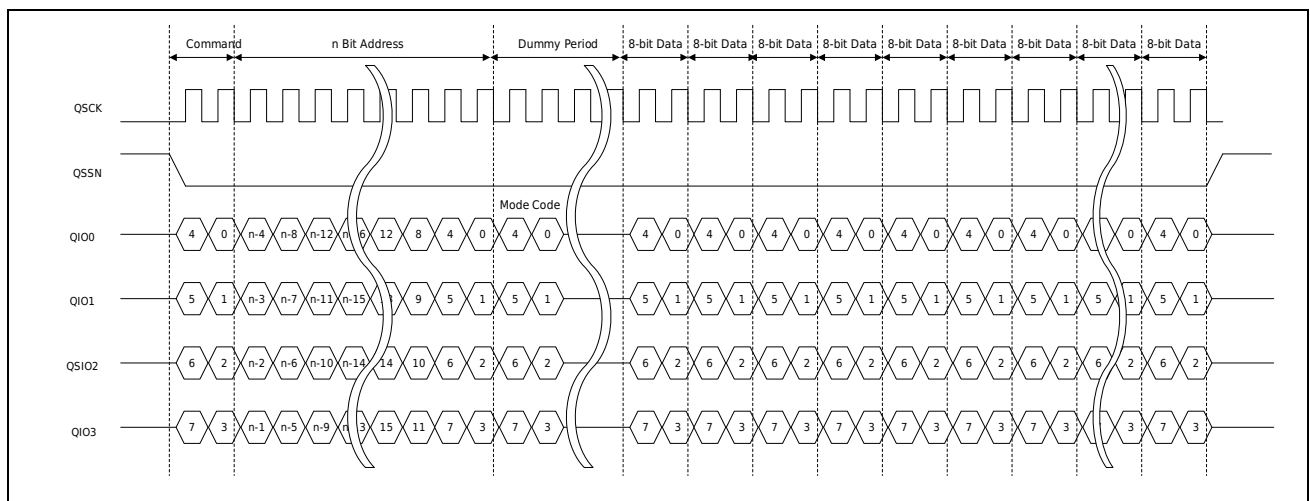


Figure 31-7 Schematic diagram wire SPI protocol action (fast read mode)

31.3.2 SPI Mode

SPI modes: mode 0 and mode 3. The mode switching can be achieved by setting the SPIMD3 bit in the QSPI_CR register. The difference between SPI mode 0 and mode 3 is that the level of QSCK is different in standby state. In SPI mode 0, the standby level of QSCK is low, while in mode 3, the standby level is high.

Serial data is output from the QSPI on the falling edge of the serial clock and read into the external flash memory on the rising edge. The external flash memory outputs serial data at the falling edge of the serial clock and is read in by QSPI at the falling edge of the next clock.

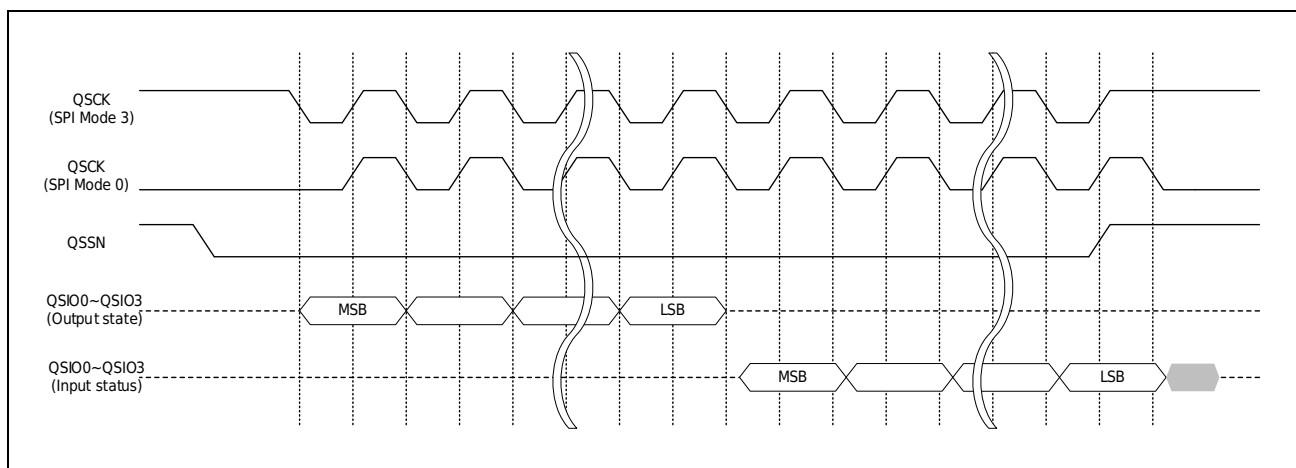


Figure 31-8 Basic Timing Diagram of Serial Interface

31.4 Timing Adjustment of QSPI Bus

The timing of QSPI bus signals can be fine-tuned through registers, and the timing after fine-tuning can be better applied to various QSPI bus accesses, whether it is ROM access mode or direct communication mode.

31.4.1 QSPI Bus Reference Clock

The reference clock of the QSPI bus is obtained by HCLK after frequency division. By setting the DIV [5:0] bits of the QSPI_CR register , you can select multiple clock sources from 2 to 64 divisions of HCLK as the reference clock for the QSPI bus.

Table 31-3 QSPI Bus Reference Clock Selection List

DIV [5:0]	Frequency Division Ratio	Actual Action Frequency (HCLK=200MHz)	DIV [5:0]	Frequency Division Ratio	Actual Action Frequency (HCLK=200MHz)
6'b000000	2	100.00	6'b100000	33	6.06
6'b000001	2	100.00	6'b100001	34	5.88
6'b000010	3	66.67	6'b100010	35	5.71
6'b000011	4	50.00	6'b100011	36	5.56
6'b000100	5	40.00	6'b100100	37	5.41
6'b000101	6	33.33	6'b100101	38	5.26
6'b000110	7	28.57	6'b100110	39	5.13
6'b000111	8	25.00	6'b100111	40	5.00
6'b001000	9	22.22	6'b101000	41	4.88
6'b001001	10	20.00	6'b101001	42	4.76
6'b001010	11	18.18	6'b101010	43	4.65
6'b001011	12	16.67	6'b101011	44	4.55
6'b001100	13	15.38	6'b101100	45	4.44
6'b001101	14	14.29	6'b101101	46	4.35
6'b001110	15	13.33	6'b101110	47	4.26
6'b001111	16	12.50	6'b101111	48	4.17
6'b010000	17	11.76	6'b110000	49	4.08
6'b010001	18	11.11	6'b110001	50	4.00
6'b010010	19	10.53	6'b110010	51	3.92
6'b010011	20	10.00	6'b110011	52	3.85
6'b010100	21	9.52	6'b110100	53	3.77
6'b010101	22	9.09	6'b110101	54	3.70
6'b010110	23	8.70	6'b110110	55	3.64
6'b010111	24	8.33	6'b110111	56	3.57
6'b011000	25	8.00	6'b111000	57	3.51
6'b011001	26	7.69	6'b111001	58	3.45

DIV [5:0]	Frequency Division Ratio	Actual Action Frequency	DIV [5:0]	Frequency Division Ratio	Actual Action Frequency
		(HCLK=200MHz)			(HCLK=200MHz)
6'b011010	27	7.41	6'b111010	59	3.39
6'b011011	28	7.14	6'b111011	60	3.33
6'b011100	29	6.90	6'b111100	61	3.28
6'b011101	30	6.67	6'b111101	62	3.23
6'b011110	31	6.45	6'b111110	63	3.17
6'b011111	32	6.25	6'b111111	64	3.13

31.4.2 SPI Bus Reference Clock

When the reference clock selects the even multiple frequency division of HCLK, the high and low level time of the QSCK signal is consistent. If the odd multiple frequency division is selected, the high level time of the QSCK signal will be one HCLK cycle longer than the low level.

QSCK to output a clock signal with a duty cycle of about 50% when odd-number division is selected, you can set the DUTY bit in the QSPI_FCR register to 1. Through this setting, the output time of the rising edge of the QSCK signal will be half a HCLK period later than before the adjustment, and the output time of the falling edge remains unchanged. Thus, a QSCK signal with a duty cycle of 50% can be obtained. When the base clock is divided by an even multiple of HCLK, please set the DUTY bit to 0. In the initial state, the DUTY bit defaults to 1.

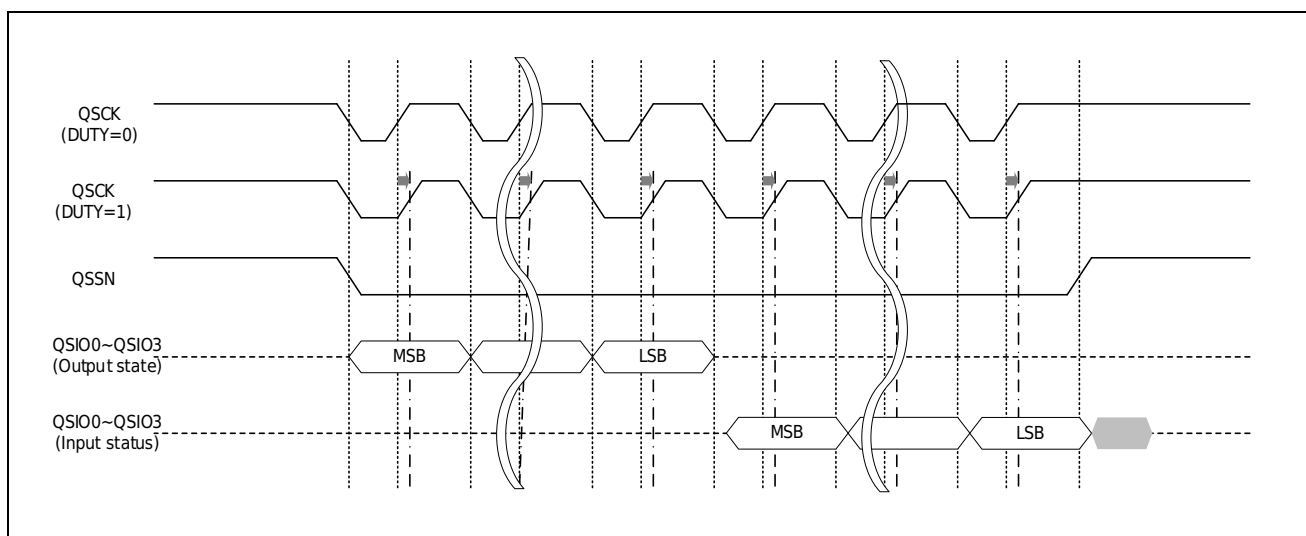


Figure 31-9 Schematic Diagram of Output Clock Duty Ratio Correction when HCLK is Divided by Three as the Reference Clock

31.4.3 QSSN Signal Minimum High Level Width

In order to meet the cancellation time requirement required by the serial flash memory, the QSSN signal must maintain a high level state (that is, an idle state) for a long enough time between two adjacent QSPI bus periods. QSSN can be selected by setting the SSHW [3:0] bits in the QSPI_CSCR register, ranging from 1 to 16 QSPI reference clock periods.

31.4.4 QSSN Establishment Time

The time from when the QSSN signal starts to output low level (that is, becomes active) to the first edge of the QSCCK signal output is called the QSSN setup time, which can be configured through register settings to meet the requirements of the external serial flash memory. Require. Set the SSNLD bit in the QSPI_FCR register to select the QSSN setup time as 0.5 or 1.5 QSPI reference clock periods. This setting can also be used to configure the setup time of the data pin from the data output permission to the first edge of the QSCCK signal output, which can be reasonably used according to the needs.

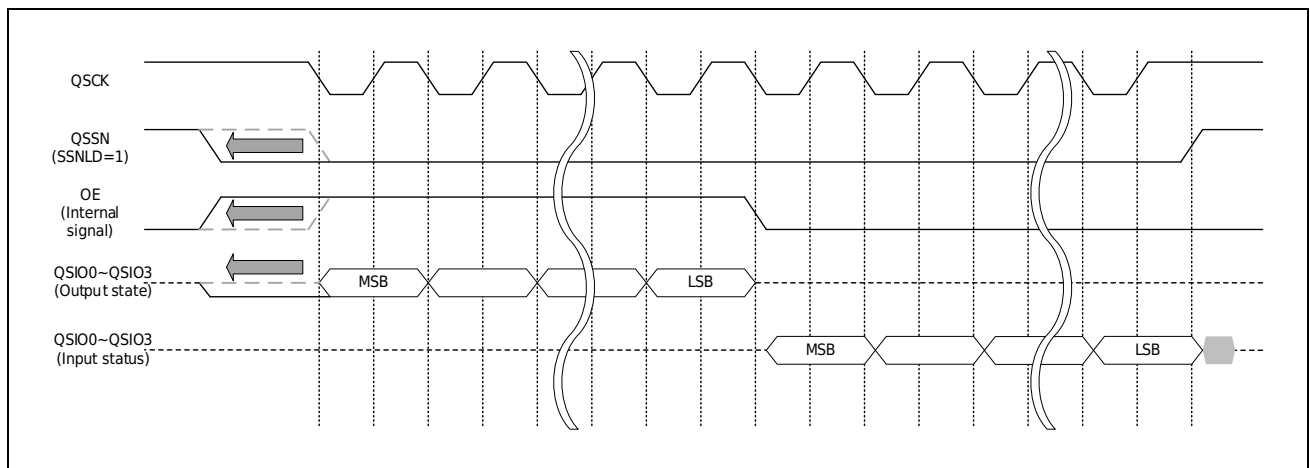


Figure 31-10 Schematic Diagram of QSSN Establishment Time Configuration

31.4.5 QSSN Hold Time

The time between the last edge of QSCCK signal output and the start of QSSN signal output high level (that is, becomes idle state) is called the hold time of QSSN, which can be configured through register settings to meet the requirements of external devices. Set the SSNHD bit in the QSPI_FCR register to select the QSSN hold time to be 0.5 or 1.5 QSPI reference clock periods.

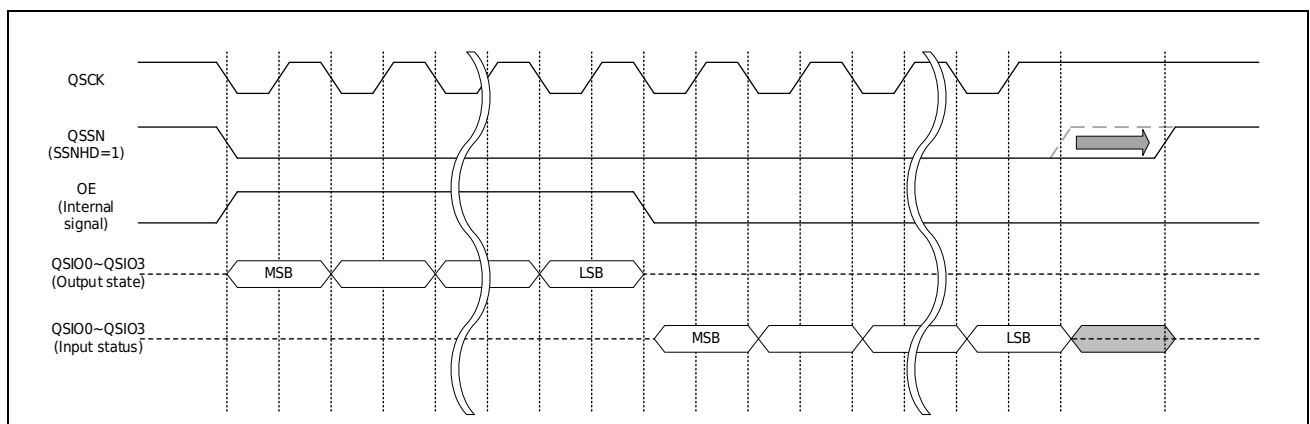


Figure 31-11 QSSN Hold Time Configuration Diagram

31.4.6 Serial Data Receive Delay

The data of the serial flash memory is output synchronously with the falling edge of Q_{SCK}, and the QSPI receives the data on the next falling edge of Q_{SCK}. The period from when the serial flash starts outputting data to when the data is received by QSPI is called receive latency. QSPI adds a delay adjustment period before the first data receive period. From the perspective of serial flash memory, this period can be regarded as an increase in the counterparty data receiving period. This delay adjustment period will only be generated when data is received.

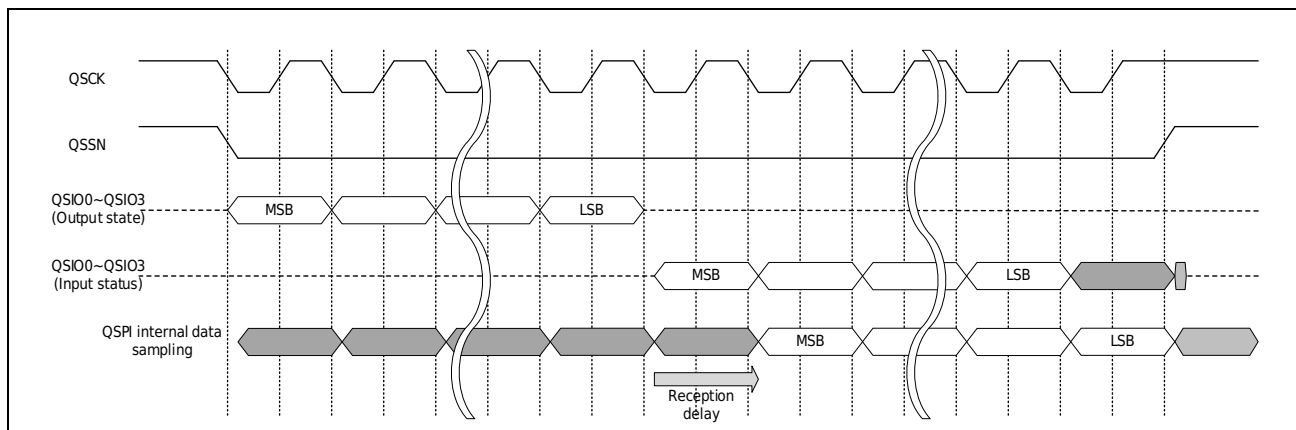


Figure 31-12 Schematic Diagram of Data Receiving Delay

31.5 Introduction to SPI Instructions for ROM Access

31.5.1 Existing QSPI-ROM Instruction Reference

Table 31-4 List of Reference Commands

Pattern		Command Code	Description
4-byte mode	standard read	8'h13	-
	Fast read	8'h0c	-
	Two-wire output fast read	8'h3c	-
	Two-wire input and output fast read	8'hbc	-
	Four-wire output fast read	8'h6c	-
	Four-wire input and output fast read	8'hec	-
	Enter 4-byte command mode	8'hb7	-
3-byte mode	Standard read	8'h03	-
	Standard read	8'h0b	When 8-bit address is selected and A8=1
	Fast read	8'h0b	-
	Two-wire output fast read	8'h3b	-
	Two-wire input and output fast read	8'hbb	-
	Four-wire output fast read	8'h6b	-
	Four-wire input and output fast read	8'heb	-
	Exit 4-byte command mode	8'he9	-
-	Write mode	8'h06	-

When accessing the serial flash memory, the command needs to be set through the command register QSPI_CCMD.

31.5.2 Standard Read Command

The standard read command is a common read command supported by most serial flash memory. When a serial bus period begins, the serial flash memory select signal is set to a valid state, and the random QSPI outputs the command code (03h/13h) of the command, and then outputs the target address. The address width can be set by the AWSL [1:0] bits in the QSPI_FCR register. Then the data can be received. The instruction selected by the initial state of QSPI is the standard read command.

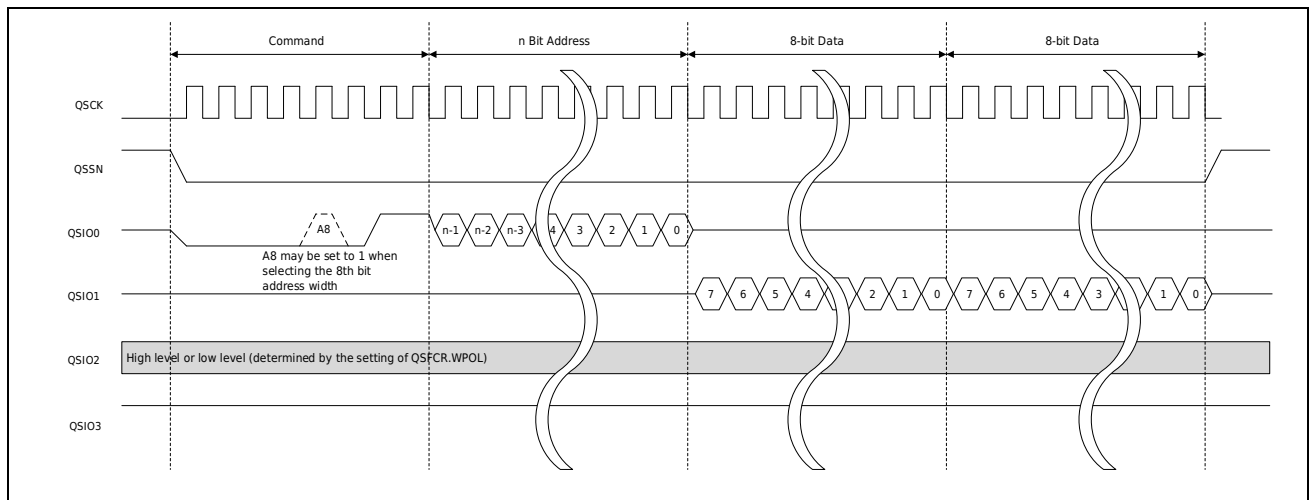


Figure 31-13 Schematic Diagram of a Standard Read Bus Period

31.5.3 Fast Read Command

The fast read command is a read command that supports a faster communication clock. When a serial bus period begins, the serial flash memory select signal is set to a valid state, and the random QSPI outputs the command code (0Bh/0Ch) of the command, and then outputs the target address. The address width can be set by the AWSL [1:0] bits in the QSPI_FCR register. The address output is followed by a certain number of dummy periods, the specific number of which is determined by DMCYCN [3:0] in the QSPI_FCR register. Next is the reception of data.

The first two periods of the dummy period are used to decide whether to select XIP mode. When XIP mode is selected, the command used in this transmission will be applied to the next SPI bus period, and the command transmission part will be omitted in the next SPI bus period. For details, please refer to [XIP control].

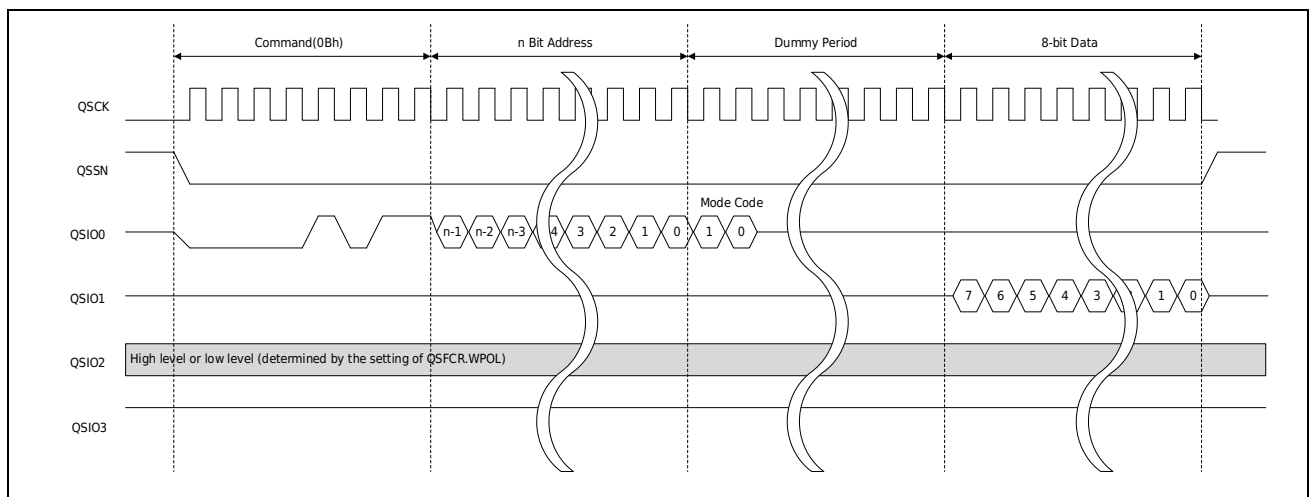


Figure 31-14 Schematic Diagram of Fast Read Bus Period

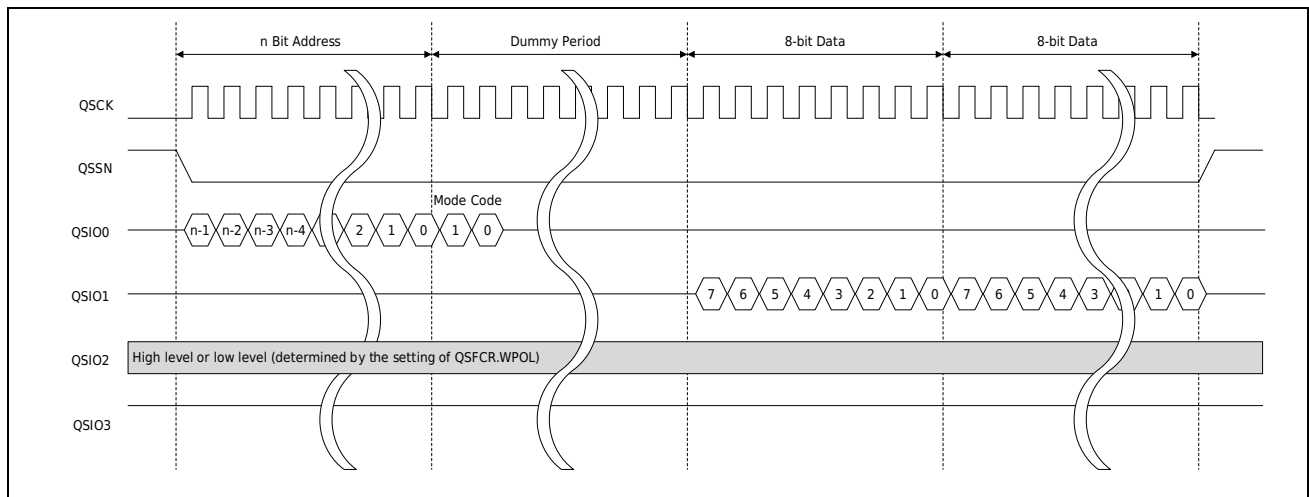


Figure 31-15 Schematic Diagram of Selecting XIP Mode Four Fast Read Bus Period

Note:

- To use the fast read command, make sure to use a serial flash that supports the fast read function.

31.5.4 Two-wire Output Fast Read Command

Two-wire output fast read is a read command that uses two signal lines for data reception. When a serial bus period starts, the serial flash select signal is set to the valid state, and the QSPI starts to output the Command Code (3Bh/3Ch) and target address of the command from the QSI00 pin. The address width can be set by the AWSL [1:0] bits in the QSPI_FCR register. After this there is a certain number of dummy periods, the specific number of which is determined by DMCCN [3:0] in the QSPI_FCR register. Then start to receive data through the two pins QSI00 and QSI01. Even bit data is received at QSI00 and odd bit at QSI01.

The first two periods of the dummy period are used to decide whether to select XIP mode. When XIP mode is selected, the command used in this transmission will be applied to the next QSPI bus period, and the command transmission part will be omitted in the next QSPI bus period. For details, please refer to [XIP control].

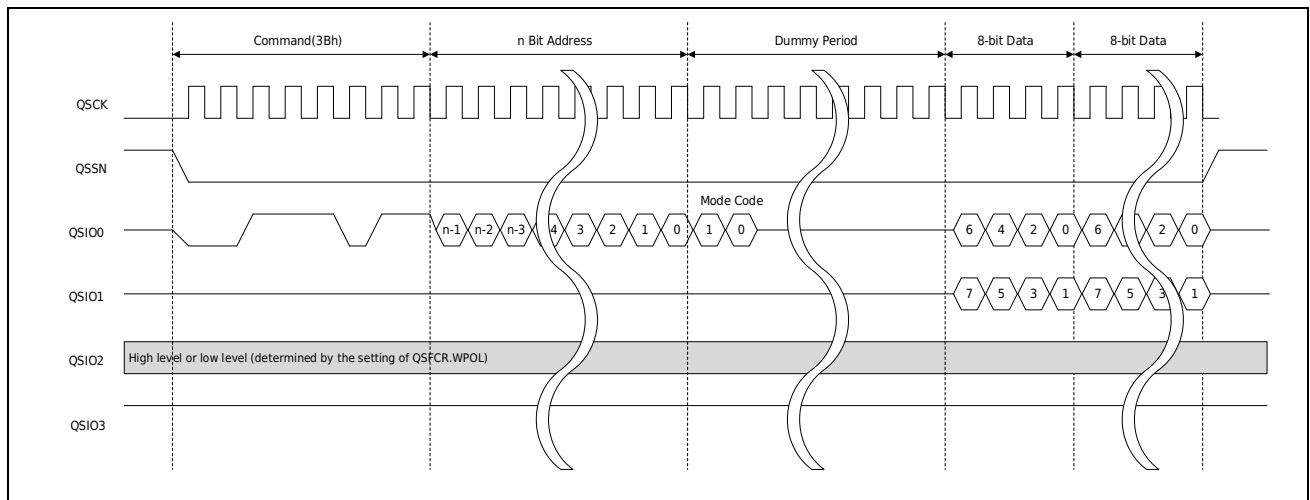


Figure 31-16 Schematic Diagram of Fast Read Bus Period

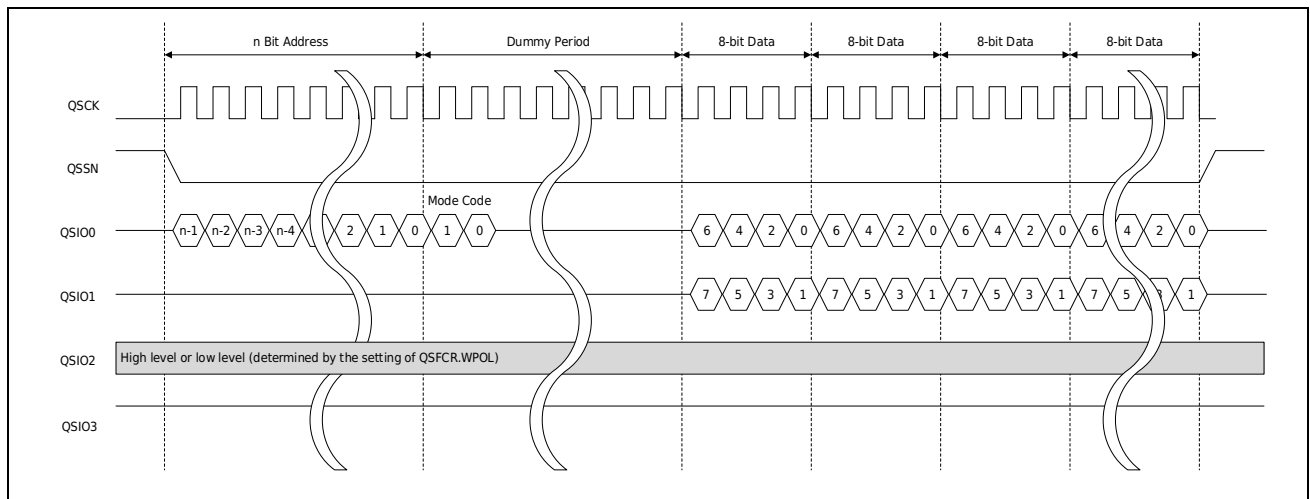


Figure 31-17 Schematic Diagram of Two-wire Output Fast Read Bus Period with XIP Mode Selected

Note:

- To use the fast read command, make sure to use a serial flash that supports the fast read function.

31.5.5 Two-Wire Input Output Fast Read Command

Two-wire output fast read is a read command that uses two signal lines for for sending and data reception. When a serial bus period starts, the serial flash select signal is asserted, and the QSPI starts to output the Command Code (BBh/BCh) of the command from the QSI00 pin. After that, QSPI outputs the target address from the QSI00 and QSI01 pins. The address width can be set by the AWSL [1:0] bits in the QSPI_FCR register. After this there is a certain number of dummy periods, the specific number of which is determined by DMCYCN [3:0] in the QSPI_FCR register. Then start to receive data through the two pins QSI00 and QSI01. The transmission and data reception of even-numbered addresses and dummy periods (including XIP mode selection information) use QSI00 pins, and odd-numbered bits use QSI01 pins.

The first two periods of the dummy period are used to decide whether to select XIP mode. When XIP mode is selected, the command used in this transmission will be applied to the next QSPI bus period, and the command transmission part will be omitted in the next QSPI bus period. For details, please refer to [XIP control].

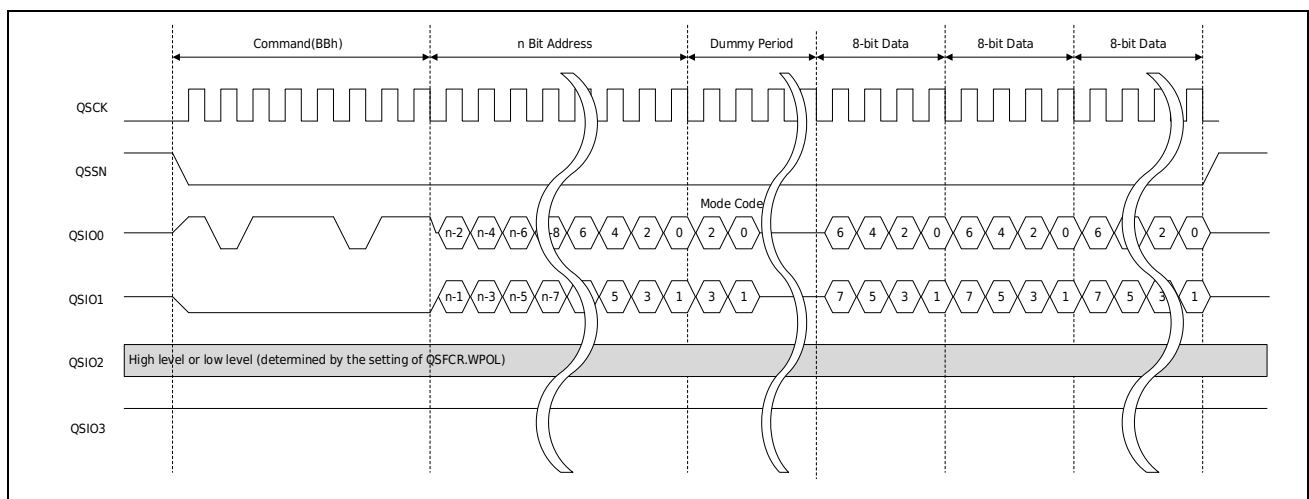


Figure 31-18 Schematic Diagram of Two-Wire Input and Output Fast Read Bus Period

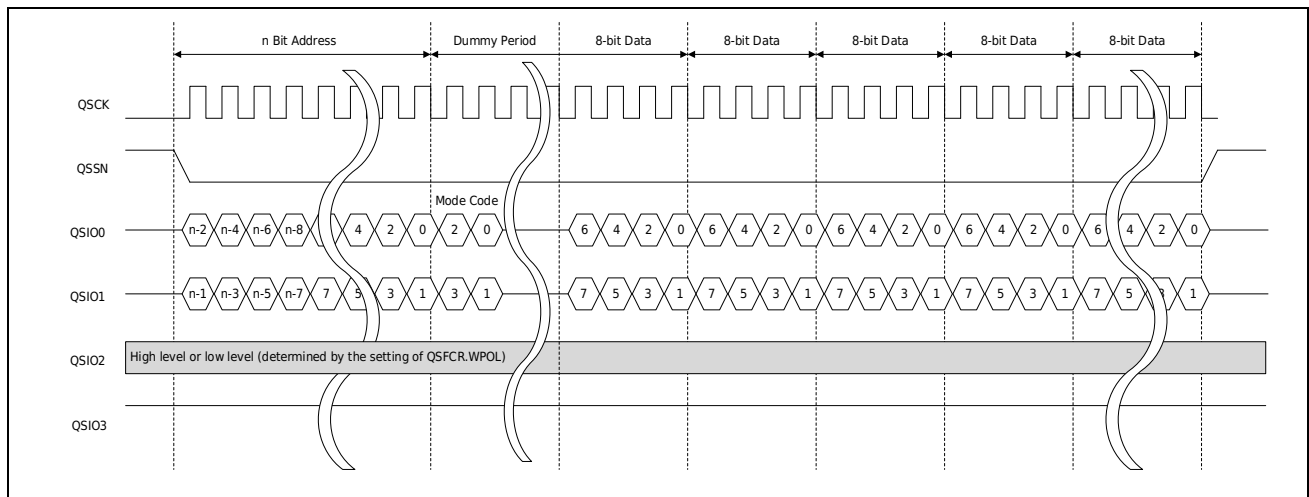


Figure 31-19 Schematic Diagram of Two-wire Input/Output Fast Read Bus Period with XIP Mode Selected

Note:

- To use the fast read command, make sure to use a serial flash that supports the fast read function.

31.5.6 Four-wire Output Fast Read Command

Four wire output fast read is a read command that uses two signal lines for data reception. When a serial bus period starts, the serial flash select signal is set to the valid state, and the QSPI starts to output the Command Code (6Bh/6Ch) and target address of the command from the QSIO0 pin. The address width can be set by the AWSL [1:0] bits in the QSPI_FCR register. After this there is a certain number of dummy periods, the specific number of which is determined by DMCYCN [3:0] in the QSPI_FCR register. Then start to receive data through the four pins QSIO0, QSIO1, QSIO2 and QSIO3.

The first two periods of the dummy period are used to decide whether to select XIP mode. When XIP mode is selected, the command used in this transmission will be applied to the next SPI bus period, and the command transmission part will be omitted in the next SPI bus period. For details, please refer to [XIP control].

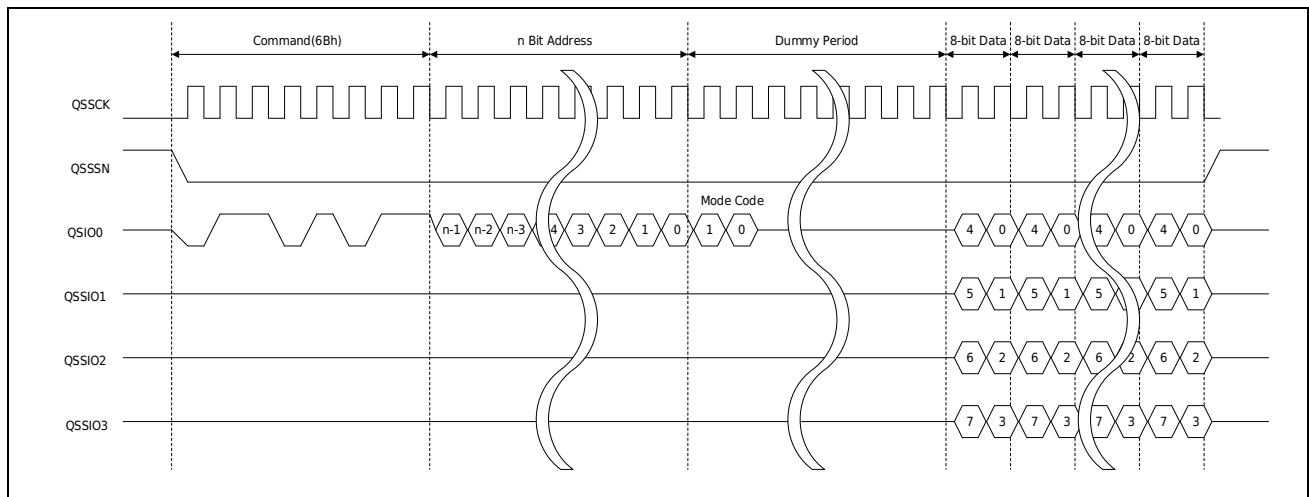


Figure 31-20 Schematic Diagram of Four-wire Output Fast Read Bus Period

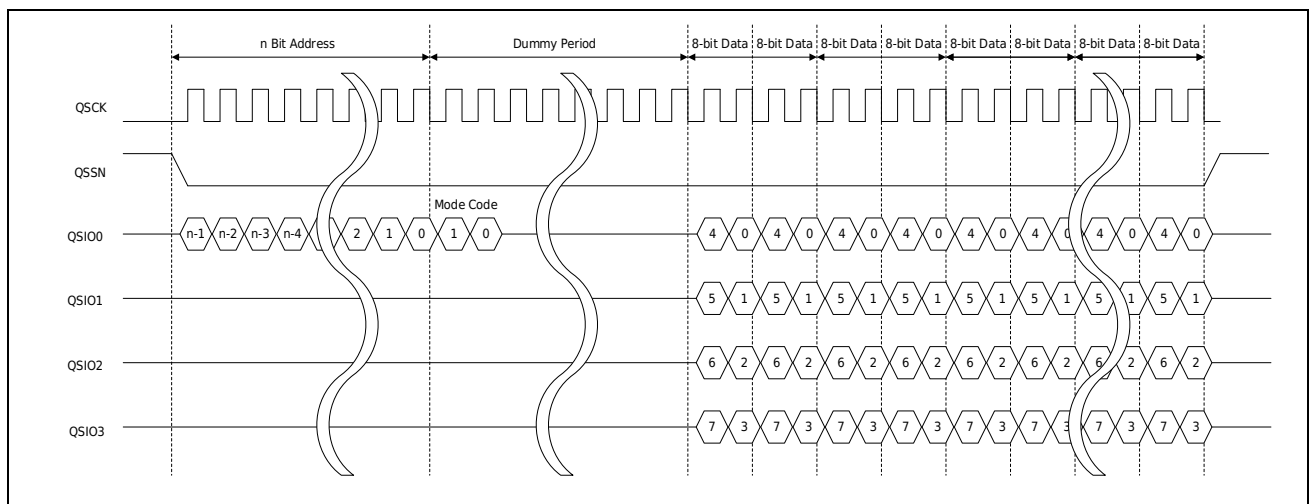


Figure 31-21 Schematic Diagram of Four-wire Output Fast Read Bus Period with XIP Mode Selected

Note:

- To use the fast read command, make sure to use a serial flash that supports the fast read function.

31.5.7 Four-wire Input Output Fast Read Command

Four wire output fast read is a read command that uses two signal lines for for sending and data reception. When a serial bus period starts, the serial flash select signal is asserted, and the QSPI starts to output the Command Code (BBh/BCh) of the command from the QSI00 pin. After that, QSPI outputs the target address from the four pins QSI00, QSI01, QSI02 and QSI03. The address width can be set by the AWSL [1:0] bits in the QSPI_FCR register. After this there is a certain number of dummy periods, the specific number of which is determined by DMCYCN [3:0] in the QSPI_FCR register. Then start to receive data through the four pins QSI00, QSI01, QSI02 and QSI03.

The first two periods of the dummy period are used to decide whether to select XIP mode. When XIP mode is selected, the command used in this transmission will be applied to the next QSPI bus

period, and the command transmission part will be omitted in the next QSPI bus period. For details, please refer to [XIP control].

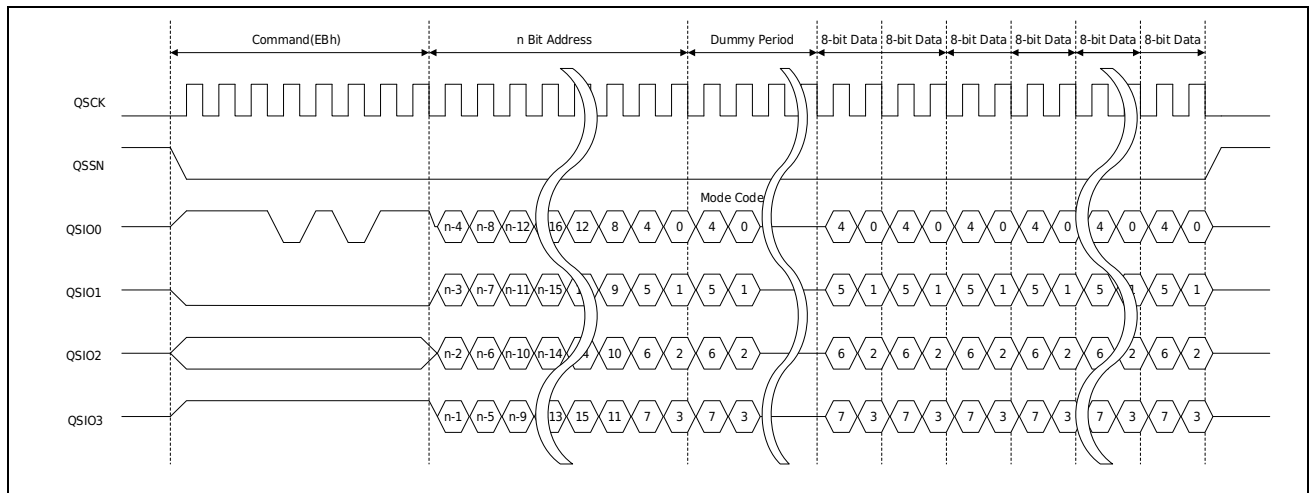


Figure 31-22 Schematic Diagram of Four-wire Input and Output Fast Read Bus Period

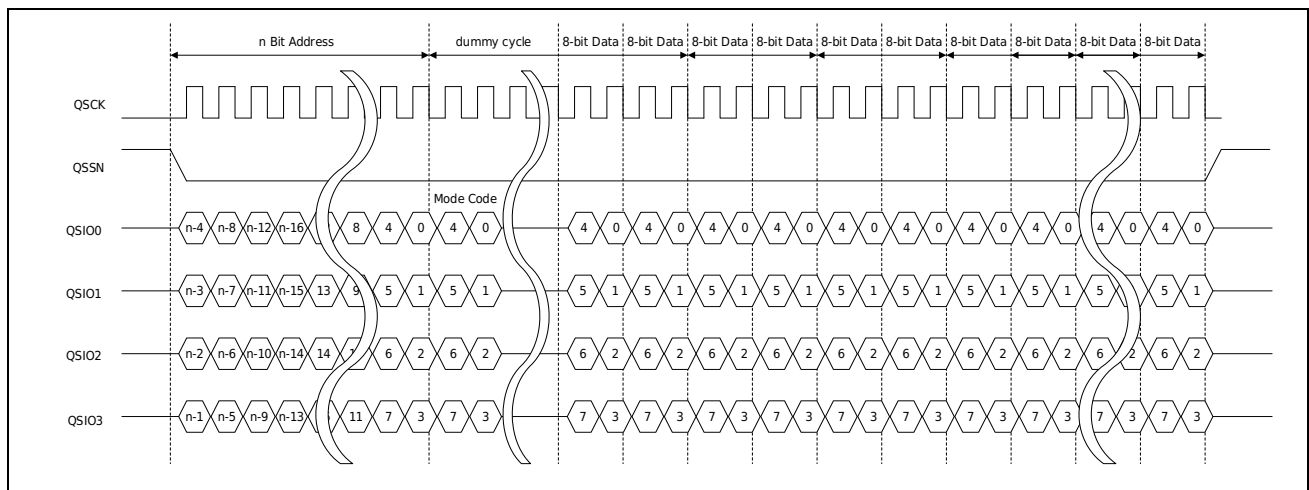


Figure 31-23 Schematic Diagram of Four-wire Input and Output Fast Read Bus Period with XIP Mode Selected

Note:

- To use the fast read command, make sure to use a serial flash that supports the fast read function.

31.5.8 Enter 4-Byte Mode Command

The command to enter 4-Byte mode can set the address width of the serial flash memory to 4 bytes. When a serial bus period starts, the serial flash select signal is asserted, and the QSPI starts outputting the command code (B7h) of the command from the QSI00 pin.

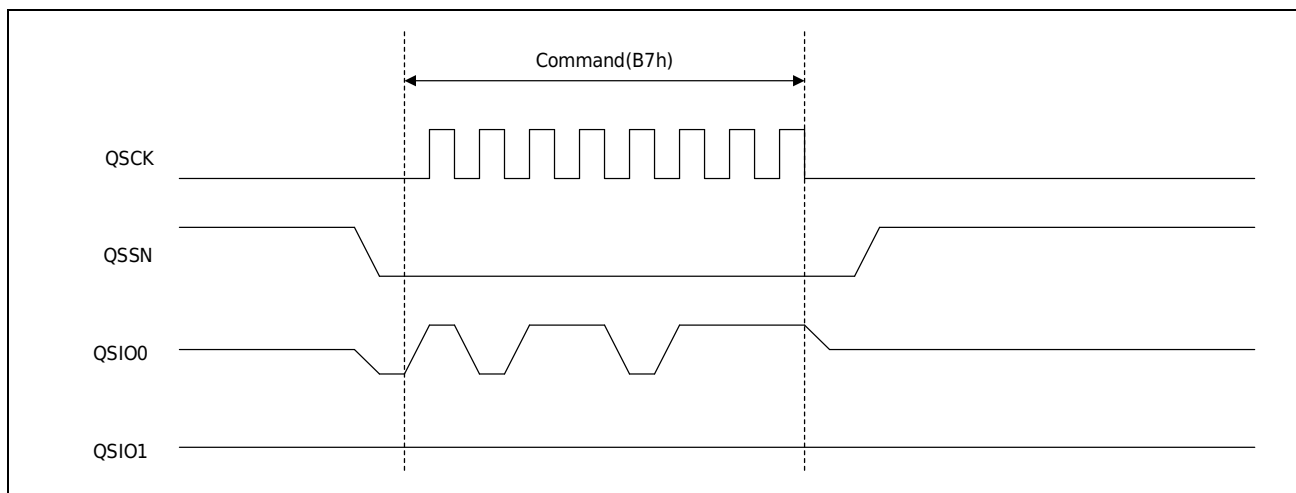


Figure 31-24 Enter 4-Byte Mode Command Bus Period Schematic Diagram

Note:

- This command can be issued regardless of whether the serial flash memory is in 4-Byte or 3-Byte mode.

31.5.9 Exit 4-Byte Mode Command

The command to exit 4-Byte mode can set the address width of the serial flash memory to 3 bytes. When a serial bus cycle starts, the serial flash select signal is asserted, and the QSPI starts to output the instruction code (BBh/BCh) of the instruction from the QSI00 pin.

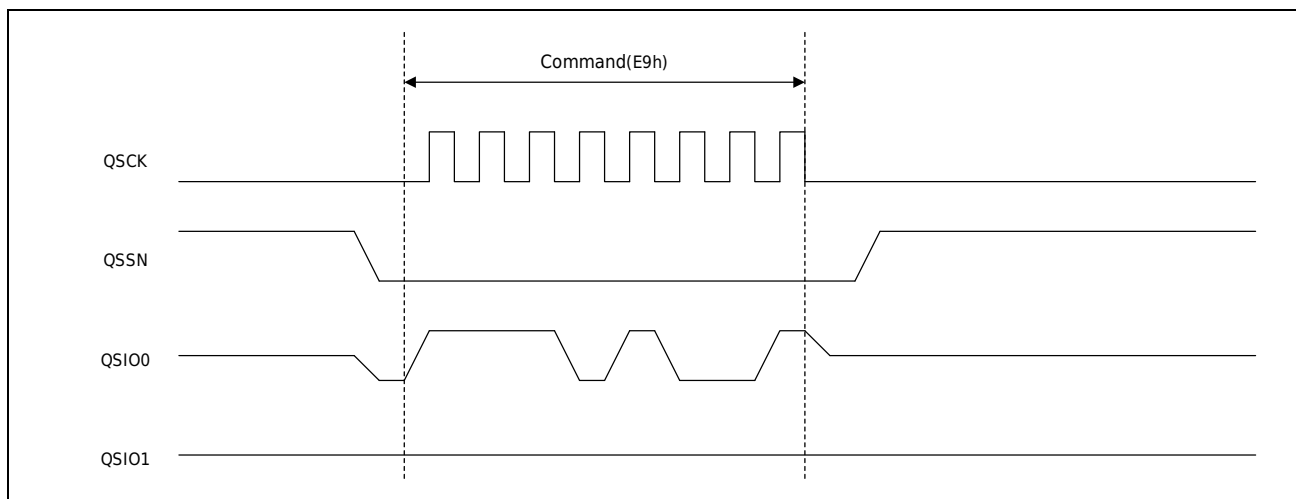


Figure 31-25 Exit 4-Byte Mode Command Bus Period Schematic Diagram

Note:

- This command can be issued regardless of whether the serial flash memory is in 4-Byte or 3-Byte mode.

31.5.10 Write Permission Command

The Write Permit command allows changing the address width of the serial flash memory. When a serial bus cycle starts, the serial flash select signal is asserted, and the QSPI starts to output the instruction code (06h) of the instruction from the QSI00 pin.

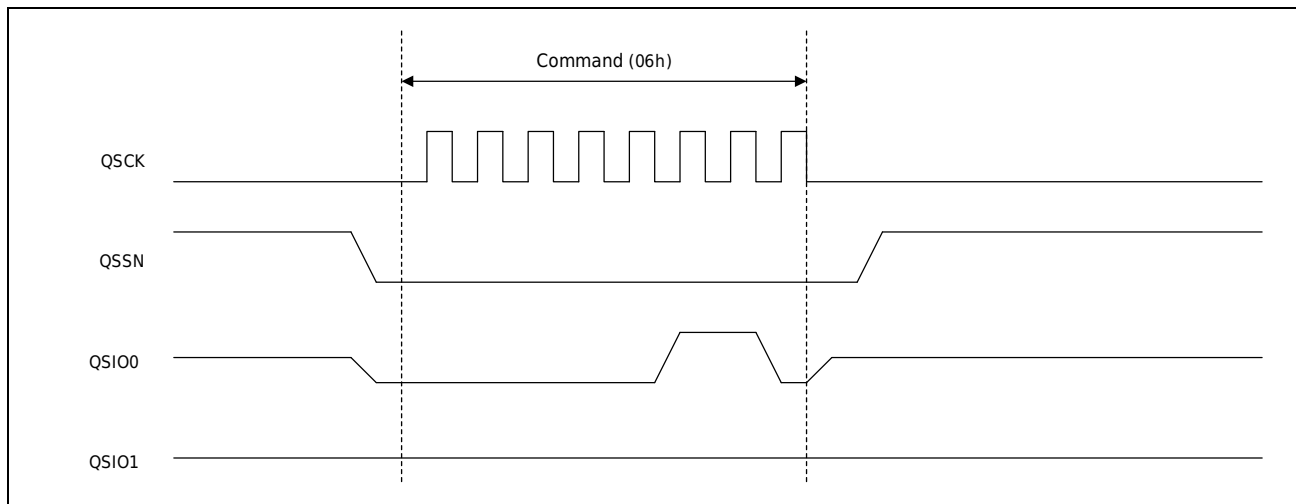


Figure 31-26 Schematic Diagram of Write Permission Command Bus Period

31.6 Scheduling of QSPI Bus Period

31.6.1 Single Flash Read with Independent Conversion

A single read command for ROM will be independently converted one-to-one from the internal bus period of the chip to the QSPI bus period. When a ROM read bus period is detected, the QSSN signal is asserted to initiate a QSPI bus period. After receiving the data from the serial flash memory, the QSSN signal becomes inactive, and the QSPI bus period is declared complete.

When another ROM read bus period is detected, the QSSN signal will be set to an active state again after ensuring that the invalid hold time has exceeded the minimum invalid hold width, and a new QSPI bus period begins.

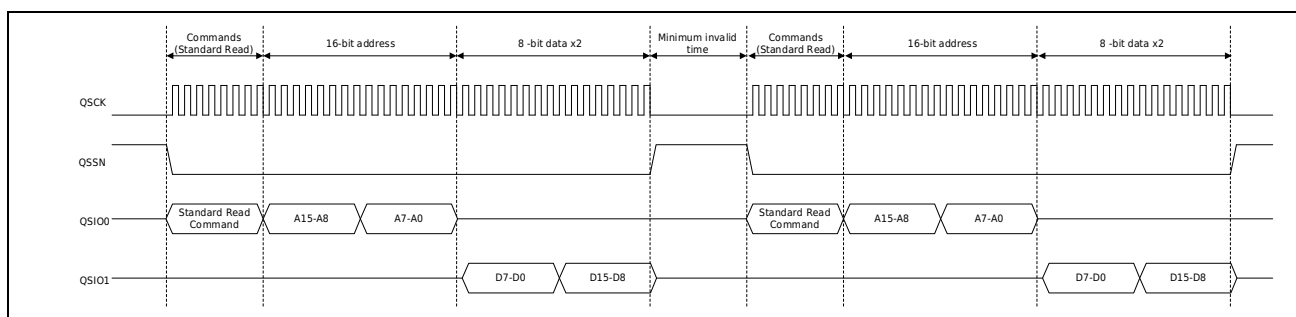


Figure 31-27 Schematic Diagram of Single Flash Memory Data Read Operation with Independent Conversion

31.6.2 Flash Read Using Pre-read Function

For the transfer of CPU commands or data blocks, the system usually reads data in a sequentially increasing flash memory address sequence. Serial flash memory has continuous data transfer capability without sending command codes and addresses again. However, if the internal bus period released by the MCU is converted independently, the QSPI bus period is also divided into independent individuals, resulting in the inability to effectively utilize the advantages of continuous data transmission of the serial flash memory. In this regard, QSPI provides a pre-read function for continuous data reception.

The pre-read function is activated by setting the PFE bit in the QSPI_CR register to 1. When this function is enabled, data will be continuously received and stored in the buffer without waiting for another flash read request. When the MCU issues a flash read operation, the QSPI will match the access address. If the match is successful, the data in the buffer at the corresponding position will be transmitted to the MCU. If the match fails, the data in the buffer will be discarded and a new QSPI bus period will be reissued.

The pre-read buffer can store a maximum of 16 bytes of data. In addition, there is a 2-byte data receiving buffer that can also store pre-read data. When all the buffer data is full, The QSPI bus period ends. When the buffer data is read and a new buffer space is generated, QSPI will automatically start a new QSPI bus period to resume the pre-read action.

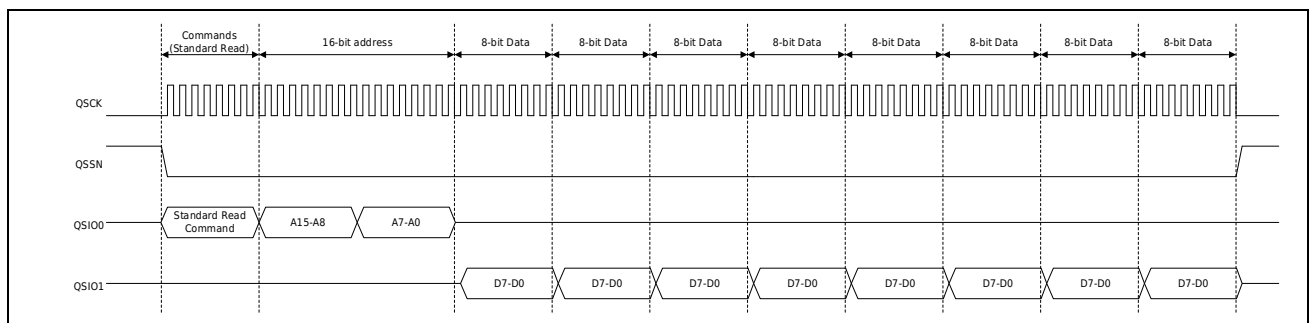


Figure 31-28 Schematic Diagram of Data Reading Operation when Pre-reading Function Valid

31.6.3 Termination of Prefetching

If a ROM read bus period to other addresses of the serial flash memory occurs during the pre-read transfer, the original pre-read action will be terminated and a new QSPI bus period will start. Normally, the pre-read action will terminate after the current byte transfer is completed, but if the PFSAE bit in the QSPI_CR register is set to 1, QSPI will immediately stop the pre-read action without waiting for the current byte transfer to be completed. Use of this feature requires the serial flash device to support the instant stop action feature.

31.6.4 Prefetch Status Monitoring

Reading data from a low-speed serial flash memory will increase the load on the system, because the internal bus needs to be in a waiting state until the QSPI bus period of the receiving signal is completed. QSPI provides read-ahead status monitoring to reduce this load.

In the pre-read status register QSPI_SR, the PFAN bit shows the current pre-read working status, the PFFUL bit indicates that the pre-read data buffer is full, and PFNUM [4:0] shows the number of bytes of data that have been read into the buffer. Through these status bits, it is convenient to determine the current pre-reading status through a CPU command.

Note:

- *When executing a read-ahead status monitoring program, place the program code outside the target serial flash area or enable command cache. Otherwise, the pre-reading object will frequently switch between the object data area and the command area, losing the meaning of pre-reading, and the monitoring program will enter an infinite loop because the pre-reading cannot be completed.*

31.6.5 Flash Read Using QSPI Bus Period Stretching Feature

If SSNW [1:0] in the QSPI_CSCR register is set to a value other than 00, the QSPI bus period will be in a hold state after receiving the data and wait for the next data to be read. At this time, the QSSN signal will remain in a low-level active state, while the QSCK is in a stopped state. If the next flash memory read command arrives, if the address of the read object is sequentially incremented next to the current address, QSPI will restart QSCK and directly accept data. If it is a discontinuous address, the QSSN signal will be set to a high-level inactive state to end the previously maintained QSPI bus period, and then start a new SPI bus period.

This function can prevent the system from repeatedly sending command codes and addresses when performing discontinuous reading of continuously increasing addresses, thereby improving reading efficiency.

The extension time of the QSPI bus period can be set by SSNW [1:0]. When the next reading does not occur after the set time, QSSN will be automatically set to a high-level invalid state to end the QSPI bus period. If SSNW [1:0] is set to 11, the QSSN will be extended indefinitely, and the QSPI bus period will always be in the hold state, but this will increase the power consumption of the serial flash memory.

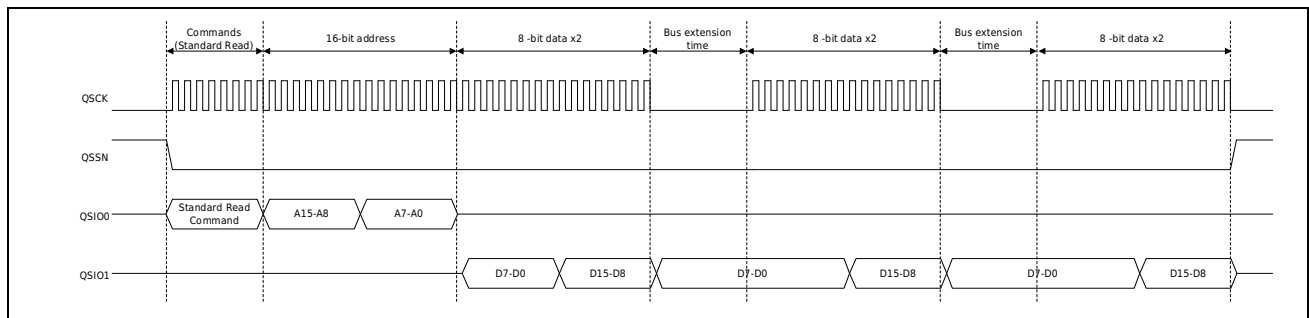


Figure 31-29 Schematic Diagram of Data Read Operation Using QSPI Bus Period Extension Function

31.7 XIP control

Some serial flash devices can reduce latency by omitting to receive a read command. This function is selected by a mode code sent during the dummy period.

During dummy periods at fast speed instructions, QSPI controls the XIP mode of the serial flash by sending XIP mode codes during the first two periods. Different serial flash memories have different XIP mode codes, which can be set specifically through the XIPMC [7:0] bits of the QSPI_XCMD register. Specific reference below .

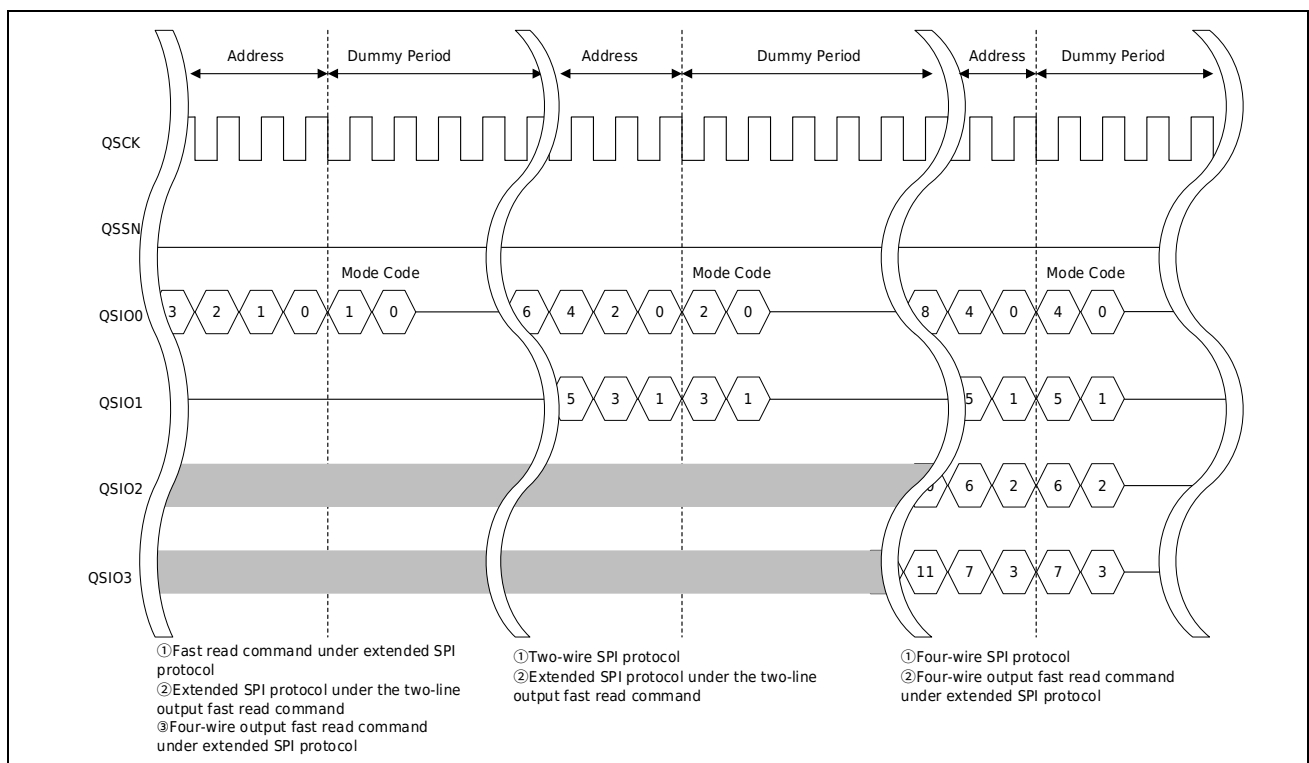


Figure 31-30 Schematic Diagram of XIP Mode Control

31.7.1 Setting of XIP mode

When the XIP mode code of the corresponding serial flash memory is written into XIPMC [7:0] of register QSPI_XCMD and the XIPE bit of register QSPI_CR is set to 1 , when the next fast read command occurs, the set mode code will be transmitted to the target serial flash memory during the first two periods of the virtual period. After the mode code is received, the serial flash memory

and its control unit will start the XIP mode. Check whether you have entered XIP mode by accessing the XIPF bit of QSPI_SR.

Note:

- XIP mode of the serial flash memory , you need to set the corresponding mode code in QSPI_XCMD [7:0]. The XIP mode of the control part only needs to set the XIPE bit to 1 , regardless of the value of QSPI_XCMD [7:0].

31.7.2 Exit from XIP mode

When the code for exiting the XIP mode of the corresponding serial flash memory is written into the XIPMC [7:0] of the QSPI_XCMD register and the XIPE bit of the QSPI_CR register is set to 0, when the next fast read command occurs, the set exit mode code will be transmitted to the target serial flash memory during the first two periods of the virtual period. After the exit mode code is received, the XIP mode of the serial flash memory and its control unit is terminated. Confirm whether you have exited XIP mode by accessing the XIPF bit of QSPI_SR.

Note:

- To exit the XIP mode of the serial flash memory, you need to set the corresponding exit mode code in QSPI_XCMD [7:0]. The XIP mode of the control part only needs to clear the XIPE bit, regardless of the value of QSPI_XCMD [7:0].

31.8 Pin Status of QSIO2 and QSIO3

the QSIO2 and QSIO3 pins depends on the serial read mode set by the MDSEL [2:0] bits in the QSPI_CR register.

Table 31-5 Pin Status of QIO2 and QIO3

QSPI_CR Register MDSEL [2:0] Bits	QSIO2 Status	QSIO3 Status	Remark
000	Output state, the output level is determined by the WPOL bit of the QSPI_FCR register , and the initial output is low level	Output high level	Standard read (initial state)
001			Fast read
010			Two-wire output fast read
011			Two-wire input and output fast read
100	As the third data line for input or output, the standby state is Hi-Z	As the third data line for input or output, the standby state is Hi-Z	Four-wire output fast read
101			Four-wire input and output fast read
110	Refer to the specific protocol settings for each stage	Refer to the specific protocol settings for each stage	Custom Protocol Standard Read
111			Custom Protocol Fast Read

Note:

- The QSIO2 pin can also be used as the WP# function of the serial flash.

- *HOLD or QSI03 pin can also be used as the RESET# function of the serial flash.*

31.9 Direct Communication Mode

31.9.1 About Direct Communication Mode

QSPI can read serial flash memory by automatically converting the MCU's external ROM read bus period to QSPI bus period. But the serial flash memory also has many different additional functions, such as ID information reading, erasing, writing, and status information reading. These functions do not have a set of standard instructions to set, and with the rapid increase of new functions of serial flash memory, the correspondence at the hardware level becomes more and more difficult.

In response to this situation, QSPI provides a direct communication mode, and users can directly control the serial flash memory through software. This mode software can generate any desired QSPI bus period.

31.9.2 Direct Communication Mode Setting of

Setting the DCOME bit in the QSPI_CR register to 1 can enter direct communication mode. Once it enters the direct communication mode, it will not be able to perform normal flash memory read operations. If you want to perform regular flash memory read operations, you need to clear the DCOME bit to exit the direct communication mode.

Note:

- *If you are in XIP mode, you need to exit XIP mode first and then start direct communication mode.*

31.9.3 QSPI Bus Period Generation in Direct Communication Mode

A complete QSPI bus period in direct communication mode starts from the first operation of DCOM [7:0] of the QSPI_DCOM register and ends after a write operation to the QSPI_CR register. During this period, multiple operations can be performed on DCOM [7:0]. Writing to DCOM [7:0] will be converted into a single-byte data transfer on the QSPI bus, while reading DCOM [7:0] will be converted into a single-byte data reception of the QSPI bus.

From the first operation on DCOM [7:0] of register QSCOM to the last write operation on QSPI_CR register, the QSSN signal always remains in the valid low state.

Direct communication mode does not support multi-line actions.

Note:

- *In direct communication mode, it is not possible to write registers other than QSPI_CR and QSPI_DCOM. Writing to other registers will exit direct communication mode. Exiting in this way may lead to unpredictable situations and is not recommended.*

31.10 Interrupt

a read access operation to the ROM is detected in direct communication mode , the RAER bit of the QSPI_SR register is set to 1 , and a QSPI bus error interrupt will be generated. The interrupt request will be held until the RAER bit is cleared. Please refer to [Interrupt Controller (INTC)] for details.

31.11 Precautions for Use

31.11.1 Setting Sequence of QSPI Registers

QSPI_EXAR can be set or changed dynamically without paying attention to the setting order. However, the setting of QSPI_EXAR must ensure that all settings are completed, otherwise the QSPI bus period may start before the registers are fully set. Therefore, please carefully configure the setting order of the registers to avoid such situations.

31.11.2 Module Stop Signal Setting

QSPI is in the module stop state after the system is reset, and the register can only be set after the QSPI module stop signal in the module stop control register is cleared. For details, please refer to [Action Mode and Low Power Mode]

31.12 Register Description

Table 31-6 List of QSPI Registers

Register base address: 0x9C000000

Register Name	Offset Address	Reset Value
QSPI control register QSPI_CR	0x0000	0x003F0000
QSPI chip select control register QSPI_CSCR	0x0004	0x0000000F
QSPI format control register QSPI_FCR	0x0008	0x000080B3
QSPI Status Register QSPI_SR	0x000C	0x00008000
QSPI direct communication instruction register QSPI_DCOM	0x0010	0xFFFFFFFF
QSPI command code register QSPI_CCMD	0x0014	0x00000000
QSPIXIP mode code register QSPI_XCMD	0x0018	0x000000FF
QSPI flag clear register QSPI_SR2 (write only)	0x0024	-
QSPI external address register QSPI_EXAR	0x0804	0x00000000

31.12.1 QSPI Control Register (QSPI_CR)

b31	b30	b29	b28	b27	b26	b25	b24
-	-	-	-	-	-	-	-
b23	b22	b21	b20	b19	b18	b17	b16
-	-	DIV [5:0]					
b15	b14	b13	b12	b11	b10	b9	b8
-	-	DPRSL [1:0]		APRSL [1:0]		IPRSL [1:0]	
b7	b6	b5	b4	b3	b2	b1	b0
SPIMD3	XIPE	DCOME	PFSAE	PFE	MDSEL [2:0]		

Bit	Flag	Bit Name	Function	Read and Write
b31~b22	Reserved	-	Read as "0", write as "0"	R/W
Serial interface reference clock selection				
b5 b4 b3 b2 b1 b0				
0 0 0 0 0 0: 2 HCLK periods				
0 0 0 0 0 1: 2 HCLK periods*				
0 0 0 0 1 0: 3 HCLK periods				
0 0 0 0 1 1: 4 HCLK periods*				
0 0 0 1 0 0: 5 HCLK periods				
0 0 0 1 0 1: 6 HCLK periods*				
0 0 0 1 1 0: 7 HCLK periods				
0 0 0 1 1 1: 8 HCLK periods*				
0 0 1 0 0 0: 9 HCLK periods				
0 0 1 0 0 1: 10 HCLK periods*				
0 0 1 0 1 0: 11 HCLK periods				
0 0 1 0 1 1: 12 HCLK periods*				
0 0 1 1 0 0: 13 HCLK periods				
0 0 1 1 0 1: 14 HCLK periods*				
0 0 1 1 1 0: 15 HCLK periods				
0 0 1 1 1 1: 16 HCLK periods*				
0 1 0 0 0 0: 17 HCLK periods				
0 1 0 0 0 1: 18 HCLK periods				
0 1 0 0 1 0: 19 HCLK periods				
0 1 0 0 1 1: 20 HCLK periods				
0 1 0 1 0 0: 21 HCLK periods				
0 1 0 1 0 1: 22 HCLK periods				
0 1 0 1 1 0: 23 HCLK periods				
0 1 0 1 1 1: 24 HCLK periods				
0 1 1 0 0 0: 25 HCLK periods				
0 1 1 0 0 1: 26 HCLK periods				
0 1 1 0 1 0: 27 HCLK periods				
0 1 1 0 1 1: 28 HCLK periods				
0 1 1 1 0 0: 29 HCLK periods				
0 1 1 1 0 1: 30 HCLK periods				
0 1 1 1 1 0: 31 HCLK periods				
0 1 1 1 1 1: 32 HCLK periods				
1 0 0 0 0 0: 33 HCLK periods				
1 0 0 0 0 1: 34 HCLK periods*				
1 0 0 0 1 0: 35 HCLK periods				
1 0 0 0 1 1: 36 HCLK periods*				
1 0 0 1 0 0: 37 HCLK periods				
1 0 0 1 0 1: 38 HCLK periods*				
1 0 0 1 1 0: 39 HCLK periods				
1 0 0 1 1 1: 40 HCLK periods*				
1 0 1 0 0 0: 41 HCLK periods				
1 0 1 0 0 1: 42 HCLK periods*				
1 0 1 0 1 0: 43 HCLK periods				
1 0 1 0 1 1: 44 HCLK periods*				
1 0 1 1 0 0: 45 HCLK periods				
1 0 1 1 0 1: 46 HCLK periods*				
1 0 1 1 1 0: 47 HCLK periods				
1 0 1 1 1 1: 48 HCLK periods*				
1 1 0 0 0 0: 49 HCLK periods				
1 1 0 0 0 1: 50 HCLK periods				
1 1 0 0 1 0: 51 HCLK periods				
1 1 0 0 1 1: 52 HCLK periods				
1 1 0 1 0 0: 53 HCLK periods				
b21~b16	DIV [5:0]	reference clock selection bit		R/W

			1 1 0 1 0 1: 54 HCLK periods 1 1 0 1 1 0: 55 HCLK periods 1 1 0 1 1 1: 56 HCLK periods 1 1 1 0 0 0: 57 HCLK periods 1 1 1 0 0 1: 58 HCLK periods 1 1 1 0 1 0: 59 HCLK periods 1 1 1 0 1 1: 60 HCLK periods 1 1 1 1 0 0: 61 HCLK periods 1 1 1 1 0 1: 62 HCLK periods 1 1 1 1 1 0: 63 HCLK periods 1 1 1 1 1 1: 64 HCLK periods	
b15~b14	Reserved	-	Read as "0", write as "0"	R/W
b13~b12	DPRSL [1:0]	SPI protocol selection in data receiving phase	SPI protocol selection in data receiving phase. b1/ b0 0 0: Extended SPI protocol 0 1: Two-wire SPI protocol 1 0: Four-wire SPI protocol 1 1: Prohibitions	R/W
b11~b10	APRSL [1:0]	SPI protocol selection in address sending phase	SPI protocol selection in address sending phase b1/ b0 0 0: Extended SPI protocol 0 1: Two-wire SPI protocol 1 0: Four-wire SPI protocol 1 1: Prohibitions	R/W
b9~b8	IPRSL [1:0]	SPI protocol selection in command sending phase	SPI protocol selection in instruction sending phase b1/ b0 0 0: Extended SPI protocol 0 1: Two-wire SPI protocol 1 0: Four-wire SPI protocol 1 1: Prohibitions	R/W
b7	SPIMD3	SPI mode selection	SPI mode selection 0: SPI mode 0 1: SPI mode 3	R/W
b6	XIPE	XIP Mode License	0: XIP mode Prohibition 1: XIP mode permission	R/W
b5	DCOME	Direct Communications License	QSPI bus Communication mode selection 0: ROM access mode 1: Direct Communication mode	R/W
b4	PFSAE	Read-Ahead Immediate Stop Permission	Choose where to reset the prefetch action 0: The current prefetch operation is aborted on a byte boundary 1: The current prefetch operation is aborted on a byte boundary	R/W
b3	PFE	Read-Ahead Immediate Stop Permission	Enable/disable selection of prefetch function 0: Invalid read the feature 1: The read the function is valid	R/W
b2~b0	MDSEL [2:0]	QSPI read the mode selection	Serial interface read mode selection b2 b1 b0 0 0 0: standard read 0 0 1: fast read 0 1 0: two-wire output fast read 0 1 1: Two-wire input and output fast read 1 0 0: Four-wire output fast read 1 0 1: Four-wire input and output fast read 1 1 0: custom standard read 1 1 1: Custom fast read	R/W

31.12.2 QSPI Chip Select Control Register (QSPI_CSCR)

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

Bit	Flag	Bit Name	Function	Read and Write
b31~b6	Reserved	-	Read as "0", write as "0"	R/W
b5~b4	SSNW [1:0]	QSSN effective time set	QSSN valid time extension function selection after QSPI bus access b5 b4 0 0: Does not QSSN effective time 0 1: Extend the valid time of QSSN by 32 QSCK periods 1 0: Extend the valid time of QSSN by 128 QSCK periods 1 1: Extend the valid time of QSSN indefinitely	R/W
b3~b0	SSHW [3:0]	QSSN minimum invalid time setting	QSSN signal minimum invalid time selection b3 b2 b1 b0 0 0 0 0: 1 QSCK periods 0 0 0 1: 2 QSCK periods 0 0 1 0: 3 QSCK periods 0 0 1 1: 4 QSCK periods 0 1 0 0: 5 QSCK periods 0 1 0 1: 6 QSCK periods 0 1 1 0: 7 QSCK periods 0 1 1 1: 8 QSCK periods 1 0 0 0: 9 QSCK periods 1 0 0 1: 10 QSCK periods 1 0 1 0: 11 QSCK periods 1 0 1 1: 12 QSCK periods 1 1 0 0: 13 QSCK periods 1 1 0 1: 14 QSCK periods 1 1 1 0: 15 QSCK periods 1 1 1 1: 16 QSCK periods	R/W

31.12.3 QSPI Format Control Register QSPI_FCR

b31	b30	b29	b28	b27	b26	b25	b24
-	-	-	-	-	-	-	-
b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8
DUTY	-	-	-	DMCYCN [3:0]			
b7	b6	b5	b4	b3	b2	b1	b0
-	WPOL	SSNLD	SSNHD	-	4BIC	AWSL [1:0]	

Bit	Flag	Bit Name	Function	Read and Write
b31~b16	Reserved	-	Read as "0", write as "0"	R/W
b15	DUTY	Duty correction	QSKC output waveform duty ratio correction 0: Do not perform duty correction 1: Delay the rising edge of QSKC by 0.5 HCLK period (valid when the frequency selected by QSKC is an odd multiple of HCLK)	R/W
b14~b12	Reserved	-	Read as "0", write as "0"	R/W
b11~b8	DMCYCN [3:0]	Dummy period setting	Dummy Cycles Selection When Using Fast Read Instructions b3 b2 b1 b0 0 0 0 0: 3 QSKC periods 0 0 0 1: 4 QSKC periods 0 0 1 0: 5 QSKC periods 0 0 1 1: 6 QSKC periods 0 1 0 0: 7 QSKC periods 0 1 0 1: 8 QSKC periods 0 1 1 0: 9 QSKC periods 0 1 1 1: 10 QSKC periods 1 0 0 0: 11 QSKC periods 1 0 0 1: 12 QSKC periods 1 0 1 0: 13 QSKC periods 1 0 1 1: 14 QSKC periods 1 1 0 0: 15 QSKC periods 1 1 0 1: 16 QSKC periods 1 1 1 0: 17 QSKC periods 1 1 1 1: 18 QSKC periods	R/W
b7	Reserved	-	Read as "0", write as "0"	R/W
b6	WPOL	WP pin output low level	WP pin (QIO2) level setting 0: low level 1: high level	R/W
b5	SSNLD	QSSN signal output time delay setting	QSSN signal Output Selection 0: Output QSSN 0.5 QSKC ahead of the first rising edge of QSKC 1: Output QSSN 1.5 QSKC ahead of the first rising edge of QSKC	R/W
b4	SSNHD	QSSN signal release time delay setting	QSSN signal release Selection 0: Disable QSSN 0.5 QSKC later than the last rising edge of QSKC 1: Disable QSSN 1.5 QSKC later than the last rising edge of QSKC	R/W
b3	Reserved	-	Read as "0", write as "0"	R/W
b2	4BIC	4-byte address read command code selection	Read command code selection when the address width is 4 bytes 0: Not 4-byte address read command code 1: Using 4-byte address read command code	R/W
b1~b0	AWSL [1:0]	Address Width Selection	Serial Interface Address Width Selection b1/ b0 0 0: 1 byte 0 1: 2 bytes 1 0: 3 bytes 1 1: 4 bytes	R/W

31.12.4 QSPI Status Register (QSPI_SR)

b31	b30	b29	b28	b27	b26	b25	b24
-	-	-	-	-	-	-	-
b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8
PFAN	PFFUL	-	PFNUM [4:0]				
b7	b6	b5	b4	b3	b2	b1	b0
RAER	XIPF	-	-	-	-	-	BUSY

Bit	Flag	Bit Name	Function	Read and Write
b31~b16	Reserved	-	Read as "0", write as "0"	R/W
b15	PFAN	read-ahead action status	Pre-read action status signal 0: Read ahead is active 1: read the is stopped	R
b14	PFFUL	read-ahead buffer status	Pre-read buffer status signal 0: There is space left in the prefetch buffer 1: Pre-read buffer data is full	R
b13	Reserved	-	Read as "0", write as "0"	R/W
b12~b8	PFNUM [4:0]	The number of bytes of data stored in the read-ahead buffer	Display of the number of data bytes stored in the pre-read buffer b4 b3 b2 b1 b0 0 0 0 0 0: 0 byte 0 0 0 0 1: 1 byte 0 0 0 1 0: 2 bytes 0 0 0 1 1: 3 bytes 0 0 1 0 0: 4 bytes 0 0 1 0 1: 5 bytes 0 0 1 1 0: 6 bytes 0 0 1 1 1: 7 bytes 0 1 0 0 0: 8 bytes 0 1 0 0 1: 9 bytes 0 1 0 1 0: 10 bytes 0 1 0 1 1: 11 bytes 0 1 1 0 0: 12 bytes 0 1 1 0 1: 13 bytes 0 1 1 1 0: 14 bytes 0 1 1 1 1: 15 bytes 1 0 0 0 0: 16 bytes 1 0 0 0 1: 17 bytes 1 0 0 1 0: 18 bytes Other settings are invalid	R
b7	RAER*1	ROM access error flag	Error flag bit for ROM access in direct communication mode 0: No ROM access detected 1: ROM access detected	R/W
b6	XIPF	XIP mode flag	XIP mode status signal 0: non-XIP mode 1: XIP mode	R
b5~b1	Reserved	-	Read as "0", write as "0"	R/W
b0	BUSY	bus busy flag	QSPI bus working status flag bit in direct communication mode 0: The bus is idle, no serial transmission in progress 1: The bus is busy, and the serial transmission process is in progress	R

*1: RAER needs to be cleared by the RAERCLR bit of QSPI_SR2

31.12.5 QSPI Command Code register (QSPI_CCMD)

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

Bit	Flag	Bit Name	Function	Read and Write
b31~b8	Reserved	-	Read as "0", write as "0"	R/W
b7~b0	RIC [7:0]	Replace command code	Serial Flash command codes to replace default commands	R/W

31.12.6 QSPI Direct Communication Command Register (QSPI_DCOM)

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

Bit	Flag	Bit Name	Function	Read and Write
b31~b8	Reserved	-	Read as "0", write as "0"	R/W
b7~b0	DCOM [7:0]	Direct communication mode command	The interface in the direct communication mode performs direct communication through the QSPI bus. Read and write accesses to this interface are translated into a corresponding QSPI bus period. This interface is only valid in direct communication mode, and it is forbidden to access in ROM access mode.	R/W

31.12.7 QSPI XIP Mode Code Register (QSPI_XCMD)

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

Bit	Flag	Bit Name	Function	Read and Write
b31~b8	Reserved	-	Read as "0", write as "0"	R/W
b7~b0	XIPMC [7:0]	XIP mode code	Mode code for serial flash. (Set XIP mode)	R/W

31.12.8 QSPI Flag Clear Register (QSPI_SR2)

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

Bit	Flag	Bit Name	Function	Read and Write
b31~b8	Reserved	-	Write "0" on write	W
b7	RAERCLR	RAERclear	Writing 1 clears the RAER bit in QSPI_SR	W
b6~b0	Reserved	-	Write "0" on write	W

31.12.9 QSPI External Extended Address Register (QSPI_EXAR)

b31	b30	b29	b28	b27	b26	b25	b24
EXADR [5:0]						-	-
b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8
-	-	-	-	-	-	-	-
b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-

Bit	Flag	Bit Name	Function	Read and write
b31~b26	EXADR [5:0]	External Extended Address Code	The upper 6 bits of the QSPI external address are set, and the ROM access window address of the QSPI can access a maximum of 64MB×63 blocks of external ROM space	R/W
b25~b0	Reserved	-	Read as "0", write as "0"	R/W

32 Programmable Logic Array (PLA)

32.1 Introduction

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 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.

32.2 PLA Module Block Diagram

32.2.1 Module Block Diagram

The PLA module mainly consists of the following parts:

- Configure PLA through APB interface
- 16 PLA units

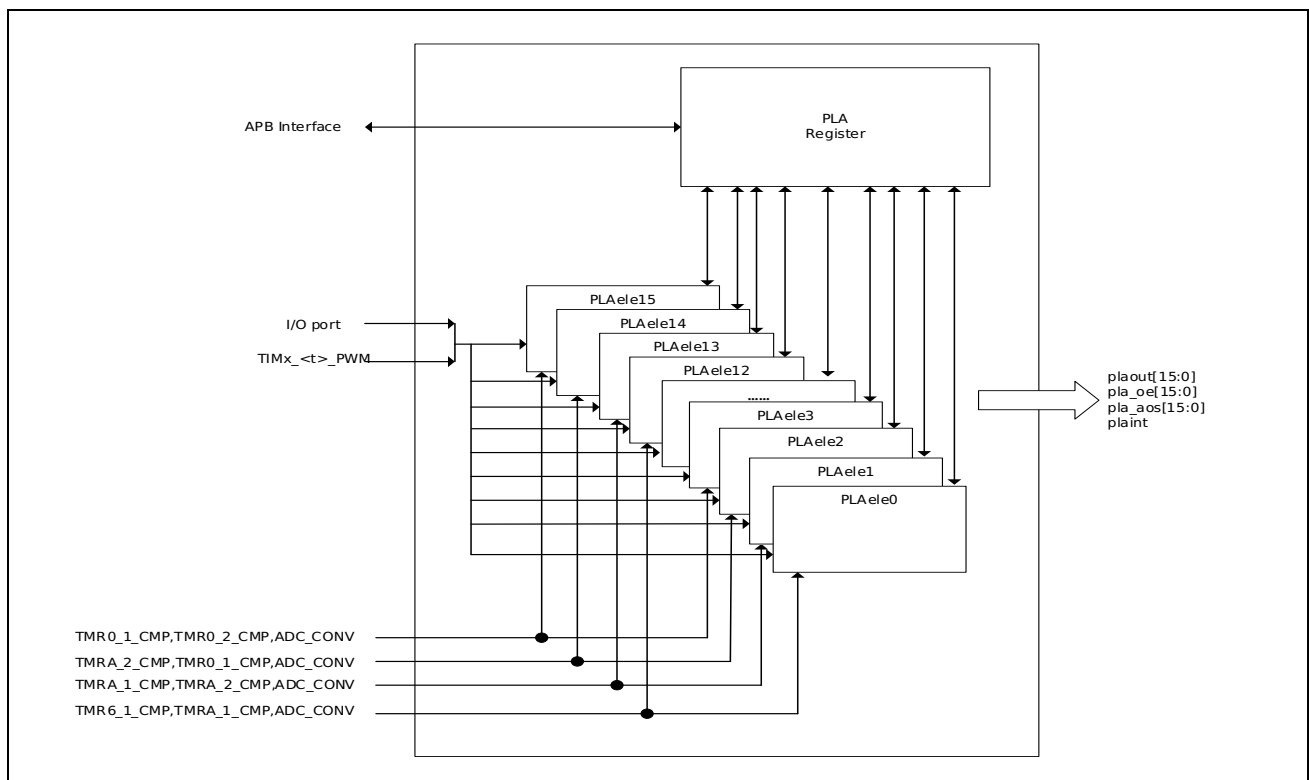


Figure 32-1 PLA Module Block Diagram

32.2.2 Action Description

This section provides a description of the PLA module functionality. The module contains sixteen identical PLA units, and the block diagram of the PLA unit is shown Figure 32-2. Each PLA unit contains two multiplexers and one LUT.

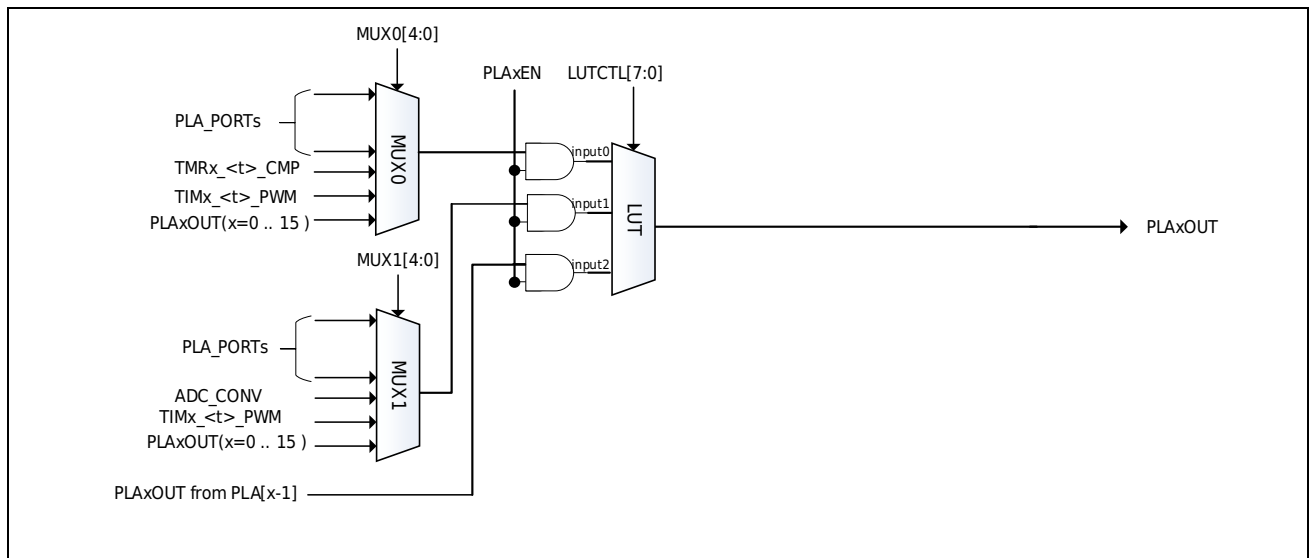


Figure 32-2 PLA Unit Block Diagram

32.2.3 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 signals (TIMx_<t>_CMP), timer channels (TIMx_<t>_PWM), ADC conversion signals (ADC_CONV), and PLA unit outputs (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 32-1 and

Table 32-2 show the input selections for PLAxMUX0 and PLAxMUX1.

Table 32-1 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_CMP	TMRA_2_CMP	TIM0_1_TRIG	TIM0_2_TRIG	TMRA_1_CMP	TMRA_2_CMP	TIM0_1_TRIG	TIM0_2_TRIG	TMRA_1_CMP	TMRA_2_CMP	TIM0_1_TRIG	TIM0_2_TRIG	TMRA_1_CMP	TMRA_2_CMP	TIM0_1_TRIG	TIM0_2_TRIG
00101	TIM6_7_PWMA	TIM6_7_PWMA	TIM6_8_PWMA	TIM6_9_PWMA	TIM6_7_PWMA	TIM6_7_PWMA	TIM6_8_PWMA	TIM6_9_PWMA	TIM6_7_PWMA	TIM6_7_PWMA	TIM6_8_PWMA	TIM6_9_PWMA	TIM6_7_PWMA	TIM6_7_PWMA	TIM6_8_PWMA	TIM6_9_PWMA
00110	TIM6_8_PWMA	TIM6_10_PWMA	TIM6_10_PWMA	TIMA_1_PWM1	TIM6_8_PWMA	TIM6_10_PWMA	TIM6_10_PWMA	TIMA_1_PWM1	TIM6_8_PWMA	TIM6_10_PWMA	TIM6_10_PWMA	TIMA_1_PWM1	TIM6_8_PWMA	TIM6_10_PWMA	TIM6_10_PWMA	TIMA_1_PWM1
00111	TIM6_9_PWMA	TIMA_1_PWM1	TIMA_1_PWM2	TIMA_1_PWM2	TIM6_9_PWMA	TIMA_1_PWM1	TIMA_1_PWM2	TIMA_1_PWM2	TIM6_9_PWMA	TIMA_1_PWM1	TIMA_1_PWM2	TIMA_1_PWM2	TIM6_9_PWMA	TIMA_1_PWM1	TIMA_1_PWM2	TIMA_1_PWM2
01000	PLAIN0 (PA15)	PLAIN4 (PB6)	PLAIN0 (PA15)	PLAIN20 (PF4)	PLAIN0 (PA15)	PLAIN4 (PB6)	PLAIN0 (PA15)	PLAIN20 (PF4)	PLAIN0 (PA15)	PLAIN4 (PB6)	PLAIN0 (PA15)	PLAIN20 (PF4)	PLAIN0 (PA15)	PLAIN4 (PB6)	PLAIN0 (PA15)	PLAIN20 (PF4)
01001	PLAIN2 (PB4)	PLAIN5 (PB7)	PLAIN20 (PB15)	PLAIN2 (PB4)	PLAIN2 (PB4)	PLAIN5 (PB7)	PLAIN20 (PB15)	PLAIN2 (PB4)	PLAIN2 (PB4)	PLAIN5 (PB7)	PLAIN20 (PB15)	PLAIN2 (PB4)	PLAIN2 (PB4)	PLAIN5 (PB7)	PLAIN20 (PB15)	PLAIN2 (PB4)
01010	PLAIN4 (PB6)	PLAIN8 (PB0)	PLAIN8 (PB0)	PLAIN6 (PB8)	PLAIN4 (PB6)	PLAIN8 (PB0)	PLAIN8 (PB0)	PLAIN6 (PB8)	PLAIN4 (PB6)	PLAIN8 (PB0)	PLAIN8 (PB0)	PLAIN6 (PB8)	PLAIN4 (PB6)	PLAIN8 (PB0)	PLAIN8 (PB0)	PLAIN6 (PB8)
01011	PLAIN6 (PB8)	PLAIN10 (PB2)	PLAIN9 (PB1)	PLAIN7 (PB9)	PLAIN6 (PB8)	PLAIN10 (PB2)	PLAIN9 (PB1)	PLAIN7 (PB9)	PLAIN6 (PB8)	PLAIN10 (PB2)	PLAIN9 (PB1)	PLAIN7 (PB9)	PLAIN6 (PB8)	PLAIN10 (PB2)	PLAIN9 (PB1)	PLAIN7 (PB9)
01100	PLAIN8 (PB0)	PLAIN12 (PA9)	PLAIN14 (PA11)	PLAIN10 (PB2)	PLAIN8 (PB0)	PLAIN12 (PA9)	PLAIN14 (PA11)	PLAIN10 (PB2)	PLAIN8 (PB0)	PLAIN12 (PA9)	PLAIN14 (PA11)	PLAIN10 (PB2)	PLAIN8 (PB0)	PLAIN12 (PA9)	PLAIN14 (PA11)	PLAIN10 (PB2)
01101	PLAIN10 (PB2)	PLAIN13 (PA10)	PLAIN15 (PA12)	PLAIN11 (PA8)	PLAIN10 (PB2)	PLAIN13 (PA10)	PLAIN15 (PA12)	PLAIN11 (PA8)	PLAIN10 (PB2)	PLAIN13 (PA10)	PLAIN15 (PA12)	PLAIN11 (PA8)	PLAIN10 (PB2)	PLAIN13 (PA10)	PLAIN15 (PA12)	PLAIN11 (PA8)
01110	PLAIN12 (PA9)	PLAIN16 (PF0)	PLAIN16 (PF0)	PLAIN18 (PA0)	PLAIN12 (PA9)	PLAIN16 (PF0)	PLAIN16 (PF0)	PLAIN18 (PA0)	PLAIN12 (PA9)	PLAIN16 (PF0)	PLAIN16 (PF0)	PLAIN18 (PA0)	PLAIN12 (PA9)	PLAIN16 (PF0)	PLAIN16 (PF0)	PLAIN18 (PA0)
01111	PLAIN14 (PA11)	PLAIN18 (PA0)	PLAIN17 (PF1)	PLAIN19 (PA1)	PLAIN14 (PA11)	PLAIN18 (PA0)	PLAIN17 (PF1)	PLAIN19 (PA1)	PLAIN14 (PA11)	PLAIN18 (PA0)	PLAIN17 (PF1)	PLAIN19 (PA1)	PLAIN14 (PA11)	PLAIN18 (PA0)	PLAIN17 (PF1)	PLAIN19 (PA1)
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 32-2 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	TIM6_10_PWMA	TIM6_8_PWMA	TIM6_7_PWMA	TIM6_7_PWMA	TIM6_10_PWMA	TIM6_8_PWMA	TIM6_7_PWMA	TIM6_7_PWMA	TIM6_10_PWMA	TIM6_8_PWMA	TIM6_7_PWMA	TIM6_7_PWMA	TIM6_10_PWMA	TIM6_8_PWMA	TIM6_7_PWMA	TIM6_7_PWMA
00110	TIMA_1_PWM1	TIM6_9_PWMA	TIM6_9_PWMA	TIM6_8_PWMA	TIMA_1_PWM1	TIM6_9_PWMA	TIM6_9_PWMA	TIM6_8_PWMA	TIMA_1_PWM1	TIM6_9_PWMA	TIM6_9_PWMA	TIM6_8_PWMA	TIMA_1_PWM1	TIM6_9_PWMA	TIM6_9_PWMA	TIM6_8_PWMA
00111	TIMA_1_PWM2	TIMA_1_PWM2	TIMA_1_PWM1	TIM6_10_PWMA	TIMA_1_PWM2	TIMA_1_PWM2	TIMA_1_PWM1	TIM6_10_PWMA	TIMA_1_PWM2	TIMA_1_PWM2	TIMA_1_PWM1	TIM6_10_PWMA	TIMA_1_PWM2	TIMA_1_PWM2	TIMA_1_PWM1	TIM6_10_PWMA
01000	PLAIN1 (PB3)	PLAIN6 (PB8)	PLAIN2 (PB4)	PLAIN0 (PA15)	PLAIN1 (PB3)	PLAIN6 (PB8)	PLAIN2 (PB4)	PLAIN0 (PA15)	PLAIN1 (PB3)	PLAIN6 (PB8)	PLAIN2 (PB4)	PLAIN0 (PA15)	PLAIN1 (PB3)	PLAIN6 (PB8)	PLAIN2 (PB4)	PLAIN0 (PA15)
01001	PLAIN3 (PB5)	PLAIN7 (PB9)	PLAIN3 (PB5)	PLAIN1 (PB3)	PLAIN3 (PB5)	PLAIN7 (PB9)	PLAIN3 (PB5)	PLAIN1 (PB3)	PLAIN3 (PB5)	PLAIN7 (PB9)	PLAIN3 (PB5)	PLAIN1 (PB3)	PLAIN3 (PB5)	PLAIN7 (PB9)	PLAIN3 (PB5)	PLAIN1 (PB3)
01010	PLAIN5 (PB7)	PLAIN9 (PB1)	PLAIN10 (PB2)	PLAIN4 (PB6)	PLAIN5 (PB7)	PLAIN9 (PB1)	PLAIN10 (PB2)	PLAIN4 (PB6)	PLAIN5 (PB7)	PLAIN9 (PB1)	PLAIN10 (PB2)	PLAIN4 (PB6)	PLAIN5 (PB7)	PLAIN9 (PB1)	PLAIN10 (PB2)	PLAIN4 (PB6)
01011	PLAIN6 (PB8)	PLAIN11 (PA8)	PLAIN11 (PA8)	PLAIN5 (PB7)	PLAIN6 (PB8)	PLAIN11 (PA8)	PLAIN11 (PA8)	PLAIN5 (PB7)	PLAIN6 (PB8)	PLAIN11 (PA8)	PLAIN11 (PA8)	PLAIN5 (PB7)	PLAIN6 (PB8)	PLAIN11 (PA8)	PLAIN11 (PA8)	PLAIN5 (PB7)
01100	PLAIN9 (PB1)	PLAIN14 (PA11)	PLAIN12 (PA9)	PLAIN8 (PB0)	PLAIN9 (PB1)	PLAIN14 (PA11)	PLAIN12 (PA9)	PLAIN8 (PB0)	PLAIN9 (PB1)	PLAIN14 (PA11)	PLAIN12 (PA9)	PLAIN8 (PB0)	PLAIN9 (PB1)	PLAIN14 (PA11)	PLAIN12 (PA9)	PLAIN8 (PB0)
01101	PLAIN11 (PA8)	PLAIN15 (PA12)	PLAIN13 (PA10)	PLAIN9 (PB1)	PLAIN11 (PA8)	PLAIN15 (PA12)	PLAIN13 (PA10)	PLAIN9 (PB1)	PLAIN11 (PA8)	PLAIN15 (PA12)	PLAIN13 (PA10)	PLAIN9 (PB1)	PLAIN11 (PA8)	PLAIN15 (PA12)	PLAIN13 (PA10)	PLAIN9 (PB1)
01110	PLAIN13 (PA10)	PLAIN17 (PF1)	PLAIN18 (PA0)	PLAIN16 (PF0)	PLAIN13 (PA10)	PLAIN17 (PF1)	PLAIN18 (PA0)	PLAIN16 (PF0)	PLAIN13 (PA10)	PLAIN17 (PF1)	PLAIN18 (PA0)	PLAIN16 (PF0)	PLAIN13 (PA10)	PLAIN17 (PF1)	PLAIN18 (PA0)	PLAIN16 (PF0)
01111	PLAIN15 (PA12)	PLAIN19 (PA1)	PLAIN19 (PA1)	PLAIN17 (PF1)	PLAIN15 (PA12)	PLAIN19 (PA1)	PLAIN19 (PA1)	PLAIN17 (PF1)	PLAIN15 (PA12)	PLAIN19 (PA1)	PLAIN19 (PA1)	PLAIN17 (PF1)	PLAIN15 (PA12)	PLAIN19 (PA1)	PLAIN19 (PA1)	PLAIN17 (PF1)
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

32.2.4 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 33-3.

Table 32-3 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 $\{\text{input2} \wedge \text{input1} \wedge \text{input0}\}$, you need to set LUTCTL [7:0] = 8'b10010110.

Application instance: To implement the logic of $\{\text{input2} \wedge \text{input1} \wedge \text{input0}\}$ 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, PLAIN1 to PB3,

Set PFSR(0,2) to 0xa, PLA0OUT to PA2,

PLA0_LUTCTL=0x00000096, implements the logic of $\{\text{input2} \wedge \text{input1} \wedge \text{input0}\}$,

PLA0_MUXS = 0x00000088, select MUX0 01000 and MUX1 01000,

PLA_GCTL=0x00000001, enables PLA0 unit.

PLA0_CTL=0x000000C0, select direct output of LUT result, and allow PLA_OUT0 output.

32.2.5 PLA Output

Each PLA unit has a GPIO port as its output which is the result of the LUT.

Table 32-4 PLA Output Port Correspondence Table

Pin Name	Input/Output	Function
PA2/PB3	O	PLA0OUT
PB0/PB4/PA7	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/PC3	O	PLA7OUT
PA13	O	PLA8OUT
PF6	O	PLA9OUT
PF7	O	PLA10OUT
PA4	O	PLA11OUT
PC4	O	PLA12OUT
PC5	O	PLA13OUT
PC11	O	PLA14OUT
PC12	O	PLA15OUT

32.3 Register Description

Table 32-5 Register List

BASE_ADDR: 0x4004FC00

Register Name	Symbol	Offset Address	Bit Width	Reset Value
Global Control Register	PLA_GCTL	0x00	32	0x00000000
Status Register	PLA_STAT	0x0C	32	0x00000000
Multiplexer Selection Register	PLAx_MUXS	$0x10 + 0x0C * x$	32	0x00000000
Lookup Table Control Register	PLAx_LUTCTL	$0x14 + 0x0C * x$	32	0x00000000
Control Register	PLAx_CTL	$0x18 + 0x0C * x$	32	0x00000000

Note*1: In the address calculation formula, $x = 0 \sim 15$

32.3.1 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
-															
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	Flag	Bit Name	Function	Read and Write
b31~b16	Reserved	-	Read at 0, write at 0	R/W
b15	PLA15EN	PLA15 unit enable	0: Disable PLA15 unit 1: Allow PLA15 unit	R/W
b14	PLA14EN	PLA14 unit enable	0: Disable PLA14 unit 1: Allow PLA14 unit	R/W
b13	PLA13EN	PLA13 unit enable	0: Disable PLA13 unit 1: Allow PLA13 unit	R/W
b12	PLA12EN	PLA12 unit enable	0: Disable PLA12 unit 1: Allow PLA12 unit	R/W
b11	PLA11EN	PLA11 unit enable	0: Disable PLA11 unit 1: Allow PLA11 unit	R/W
b10	PLA10EN	PLA10 unit enable	0: Disable PLA10 unit 1: Allow PLA10 unit	R/W
b9	PLA9EN	PLA9 unit enable	0: Disable PLA9 unit 1: Allow PLA9 unit	R/W
b8	PLA8EN	PLA8 unit enable	0: Disable PLA8 unit 1: Allow PLA8 unit	R/W
b7	PLA7EN	PLA7 unit enable	0: Disable PLA7 unit 1: Allow PLA7 unit	R/W
b6	PLA6EN	PLA6 unit enable	0: Disable PLA6 unit 1: Allow PLA6 unit	R/W
b5	PLA5EN	PLA5 unit enable	0: Disable PLA5 unit 1: Allow PLA5 unit	R/W
b4	PLA4EN	PLA4 unit enable	0: Disable PLA4 unit 1: Allow PLA4 unit	R/W
b3	PLA3EN	PLA3 unit enable	0: Disable PLA3 unit 1: Allow PLA3 unit	R/W
b2	PLA2EN	PLA2 unit enable	0: Disable PLA2 unit 1: Allow PLA2 unit	R/W
b1	PLA1EN	PLA1 unit enable	0: Disable PLA1 unit 1: Allow PLA1 unit	R/W
b0	PLA0EN	PLA0 unit enable	0: Disable PLA0 unit 1: Allow PLA0 unit	R/W

32.3.2 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
-															
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	Flag	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

32.3.3 Multiplexer Selection Register (PLA to _MU XS) (small = 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
-															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MUX0 [4]	MUX1 [4]	-						MUX0 [3:0]				MUX1 [3:0]			

Bit	Flag	Bit Name	Function	Read and Write
b31~b16	Reserved	-	Read as 0	R/W
b15	MUX0 [4]	Multiplexer 0 input selects bit 4	This bit selects the input of MUX0 Refer to "13.4.1 Input Multiplexer " Table 13-1 PLAxMUX0 input selection	R/W
b14	MUX1 [4]	Multiplexer 1 input selects bit 4	This bit selects the input of MUX1 Refer to "13.4.2 Input Multiplexer " Table 13-1 PLAxMUX1 input selection	R/W
b13~b8	Reserved	-	Read as 0, write as 0	R/W
b7~b4	MUX0 [3:0]	Multiplexer 0 input selects bits 3~0	These bits select the input of MUX0 Refer to "13.4.1 Input Multiplexer " Table 13-1 PLAxMUX0 input selection	R/W
b3~b0	MUX1 [3:0]	Multiplexer 1 input selection bits 3 to 0	These bits select the input of MUX1 Refer to "13.4.2 Input Multiplexer " Table 13-1 PLAxMUX1 input selection	R/W

32.3.4 Lookup Table Control Register (PLA to UTC TL) (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
-															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-								LUTCTL [7:0]							

Bit	Flag	Bit Name	Function	Read and Write
b31~b8	Reserved	-	Read as 0, write as 0	R/W
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	R/W

32.3.5 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
-															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-								OSEL	OEN	-	-	-	-	-	-

Bit	Flag	Bit Name	Function	Read and Write
b31~b7	Reserved	-	Read as 0, write as 0	R/W
b7	OSEL	LUT enable	0: The result of LUT is not output as PLAxOUT 1: The result of LUT is output as PLAxOUT	R/W
b6	OEN	Output enable	0: PLAxOUT output disabled 1: PLAxOUT output enabled	R/W
b5~b0	Reserved	-	Read as 0, write as 0	R/W

32.4 Precautions

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 combinational loop. When modifying the PLA register, the PLA_GCTL register needs to be cleared to prevent combinational loop.

33 USB2.0 Full Speed Module (USBFS)

33.1 Introduction to USBFS

The USB Full Speed (USBFS) controller provides a set of USB communication solutions for portable devices. The USBFS controller supports host mode and device mode, and a full-speed PHY is integrated inside the chip. In host mode, the USBFS controller supports full-speed (FS, 12Mb/s) and low-speed (LS, 1.5Mb/s) transceivers, while in device mode it only supports full-speed (FS, 12Mb/s) transceivers. The USBFS controller supports all four transfer modes defined by the USB 2.0 protocol (control transfer, bulk transfer, interrupt transfer, and isochronous transfer). The protocol followed is as follows:

- Universal Serial Bus Revision 2.0 Specification
- USB 2.0 Link Power Management Addendum Engineering Change Notice to the USB 2.0 specification, July 16, 2007
- Errata for USB 2.0 ECN: Link Power Management (LPM) - 7/2007

33.2 Main Features of USBFS

There are three main categories: general features, host mode features, and device mode features.

33.2.1 General features

- Built-in on-chip USB2.0 full-speed PHY
- Support host mode and device mode
- Supports FS SOF and low-speed "Keep-alive" tokens and has the following functions:
 - SOF pulse pin output function
 - The SOF pulse can be used as an event source inside the chip to trigger the work of TIMER, DMA and other modules
 - Configurable frame period
 - Configurable end-of-frame interrupt
- The module is embedded with DMA, and the AHB burst transfer type can be configured by software
- Power-saving features such as USB suspend, stop RAM clock, stop PHY domain clock
- 1.25KB of dedicated RAM with advanced FIFO control
 - The RAM space can be divided into different FIFOs for flexible and effective use of RAM
 - Each FIFO can store multiple packets
 - Dynamically allocate memory
 - The FIFO size can be configured as a power of 2 to allow for continuous use of storage cells.
- No application intervention is required within one frame to achieve maximum USB bandwidth

- Can automatically determine the host mode or device mode according to the level of the ID line

33.2.2 Host Mode Features

- Host mode supports USB2.0 full speed (FS, 12Mb/s) and low speed (LS, 1.5Mb/s) transmission
- Need to generate VBUS voltage through external power chip
- Up to 12 host channels (pipes): each channel can be dynamically reconfigured to support any type of USB transfer
- Built-in hardware scheduler that can:
 - Store up to 8 interrupt plus isochronous transfer requests in a periodic hardware queue
 - Store up to 8 control plus bulk transfer requests in an aperiodic hardware queue
- Manages a shared RX FIFO, a periodic TX FIFO, and an aperiodic TX FIFO for efficient use of USB data RAM

33.2.3 Device Mode Properties

- The slave mode supports USB2.0 full speed (FS, 12Mb/s) transmission.
- 1 bidirectional control endpoint 0
- 5 OUT endpoints that can be configured to support bulk transfers, interrupt transfers, or isochronous transfers
- 5 IN endpoints that can be configured to support bulk transfers, interrupt transfers, or isochronous transfers
- Contains 6 transmit FIFOs (one for each IN endpoint) and one receive FIFO (shared by all OUT endpoints)
- Support remote wake-up function.
- Support soft disconnect function
- VBUS PIN supports 5V withstand voltage.

33.3 USBFS System Block Diagram

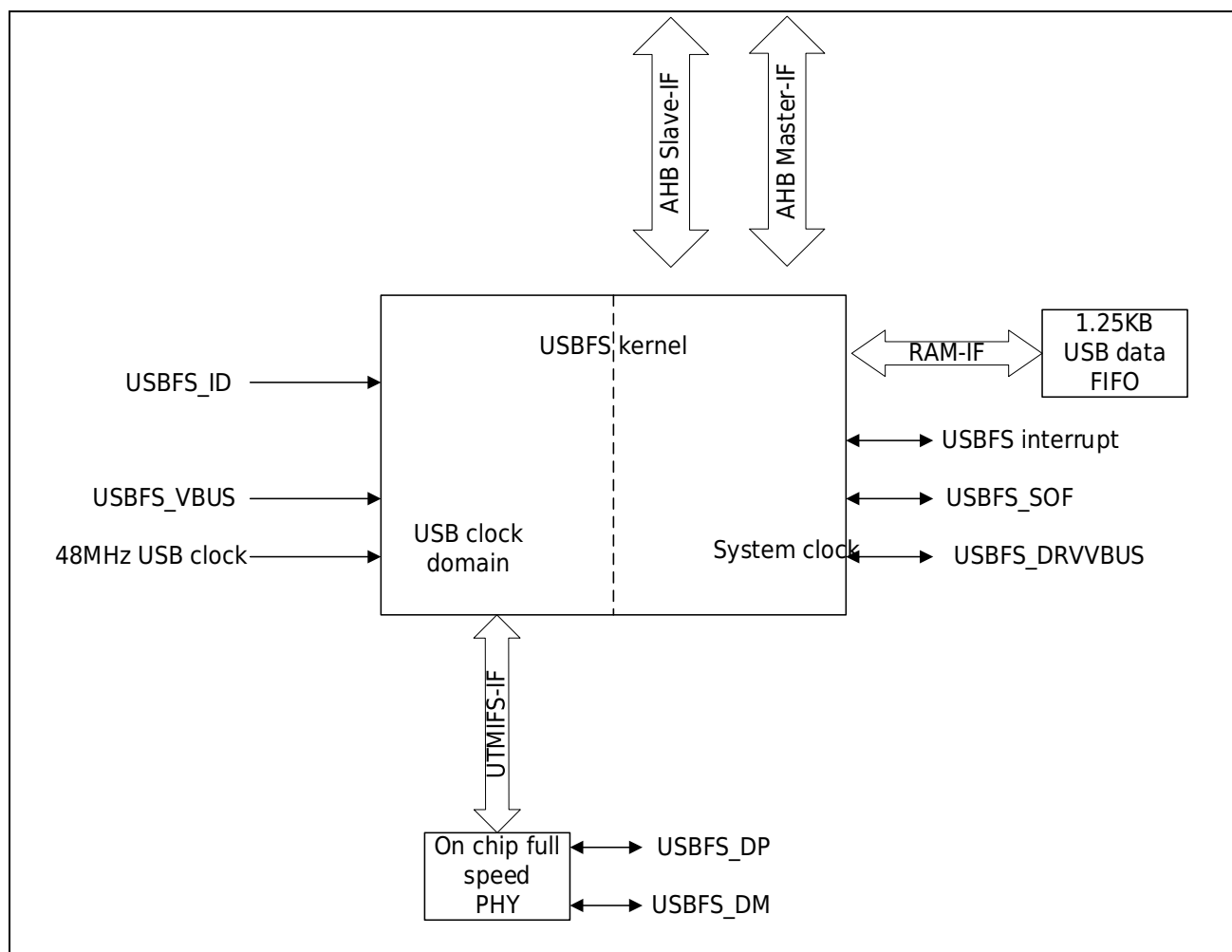


Figure 33-1 USBFS System Block Diagram

33.4 USBFS Pin Description

Table 33-1 USBFS Pin Description

Pin Name	Direction	Applicable Mode	Corresponding Pin	Functional Description
USBFS_VBUS	Input	Equipment	PA9	Power port, 5V withstand voltage
USBFS_DP	Input/Output	Host, device	PA12	Differential data D+ signal
USBFS_DM	Input/Output	Host, device	PA11	Differential Data D-Signal
USBFS_DRVVBUS	Output	Host	PB8	External power chip enable signal
USBFS_ID	Input	Host	PA10	USB AB device identification signal
USBFS_SOF	Output	Host, device	PA8	SOF output pulse signal

Since the USBFS_DP and USBFS_DM pins are multiplexed with general GPIO , it is recommended to turn off the digital functions of the corresponding pins when using USBFS. The USBFS_DP and USBFS_DM functions are analog functions and have nothing to do with the corresponding PFSR register settings. Please refer to the [General IO (GPIO)] section for details. In addition, when the

USBFS function is not in use, when the digital function pins corresponding to the USBFS_DP and USBFS_DM pins are flipped, additional current consumption will be generated.

33.5 USBFS Function Description

33.5.1 USBFS Clock and Working Mode

The clock used by USBFS needs to be configured as 48MHz. The 48MHz clock is generated by the internal PLL circuit. The PLL clock source needs to select an external high-speed oscillator. Before using the USBFS module, the USBFS clock needs to be configured in the CMU module.

USBFS can be used as a host or a device and includes an on-chip full-speed PHY.

Pull-up and pull-down resistors have been integrated inside the on-chip full-speed PHY, and USBFS can be automatically selected according to the current mode and connection status.

When USBFS is working, the VCC voltage range is 3.0~3.6V.

33.5.2 USBFS Mode Decision

There are two ways for USBFS to determine the current working mode:

Method 1: Automatically identify according to the state of the USBFS_ID line. When the USBFS_ID line is detected to be at a high level, the module will work in the device mode. When the USBFS_ID line is detected to be at a low level, the module will work in the host state.

Method 2: Force the host/device mode. By setting the FDMOD or FHMOD bit of the register USBFS_GUSBCFG to 1, the module ignores the level of the USBFS_ID line and is forced to work in the device or host mode.

33.5.3 USBFS Host Function

33.5.3.1 Introduction to Host Functions

When USBFS works in the host mode, VBUS is the 5V power supply pin stipulated by the USB protocol. The internal PHY does not support 5V power supply, so an external USB power chip is required to power the device. USBFS_DRVVBUS is used to enable the external USB power supply chip, and the overcurrent detection of the external power supply chip can be realized through the external interrupt IRQ of this MCU. USBFS_VBUS can be used as GPIO in host mode.

A typical USB host mode system construction diagram is as follows:

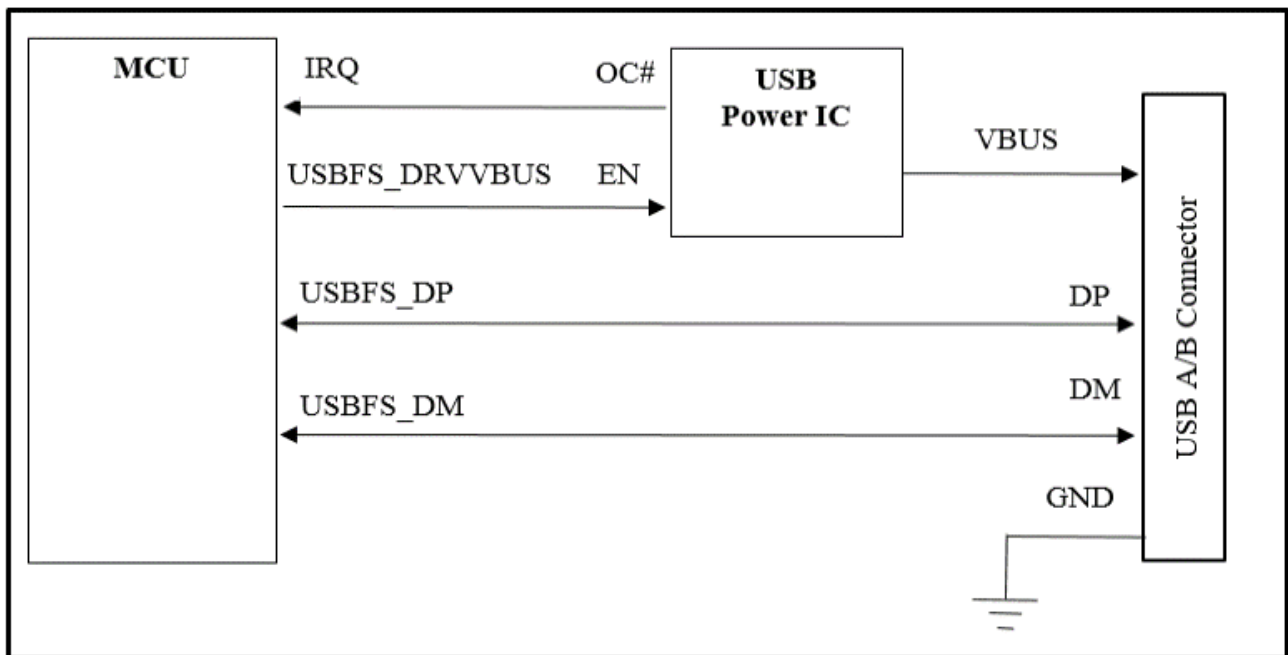


Figure 33-2 USBFS Host Mode System Construction Diagram

33.5.3.2 Host Port Power Supply

This MCU cannot output 5V to provide VBUS. To do this, a USB power chip or a basic power switch (such as an application board providing 5V power) must be added outside the microcontroller to drive the 5V VBUS line. The external power chip can be driven by any GPIO output or USBFS_DRVVBUS. When the application determines to use GPIO to control external devices to provide VBUS, the port power bit (PPWR bit in USBFS_HPRT) in the host port control and status register must still be set to 1.

33.5.3.3 Host Detects Device Connection and Disconnection

The USB device will be detected immediately after connection. The USBFS module will signal a host port interrupt, which is triggered by the device connection bit (PCDET bit in USBFS_HPRT) in the host port control and status register.

A device disconnect event will trigger a disconnect detection interrupt (DISCINT bit in USBFS_GINTSTS).

33.5.3.4 Host Enumeration

After a device connection is detected, if a new device is connected, the host must initiate the enumeration process by sending a USB reset and configuration command to the new device.

The application drives the USB reset signal (single-ended zero) over the USB by setting the port reset bit in the host port control and status register (PRST bit in USBFS_HPRT) for a minimum of 10 ms and a maximum of 20 ms. The application calculates the duration of this process and then clears the port reset bit.

Immediately after the completion of the USB reset sequence, the port enable/disable change bit (PENCHNG bit in USBFS_HPRT) triggers a host port interrupt, which in turn notifies the application that the port speed field in the host port control and status register (in USBFS_HPRT PSPD) reads the enumerated device speed, and the host has started driving SOF (FS) or Keep-alive tokens (LS). At this point the host is ready to enumerate the device by sending commands to the device.

33.5.3.5 Host Suspend

The application suspends the USB bus by setting the Port Suspend bit in the Host Port Control and Status Register (PSUSP in USBFS_HPRT). USBFS module stops sending SOF and enters suspend state.

The bus can be taken out of suspend by autonomous activity (remote wake-up) by a remote device. In this case, the remote wake-up signal will trigger the remote wake-up interrupt (the WKUPINT bit in USBFS_GINTSTS), and the hardware will automatically reset the port recovery bit in the host port control and status register (the PRES bit in USBFS_HPRT), and automatically drive the recovery through USB Signal. The application must time the resume window, then clear the port resume bit to exit suspend and restart SOF.

If the exit from suspend is initiated by the host, the application must set the port resume bit to initiate resume signaling on the host port, time the resume window and finally clear the port resume bit.

33.5.3.6 Host Channel

The USBFS module implements 12 host channels. Each host channel can be used for USB host transfers (USB pipes). The host can handle up to 8 transfer requests at the same time. If the application has more than 8 transfer requests pending, after the channel is released from the previous task (i.e. after receiving the transfer complete and channel stop interrupts), the host controller driver (HCD) must reallocate channels for outstanding transfer requests.

Each host channel can be configured to support input/output and periodic/asynchronous transactions. Each host channel uses dedicated control (HCCHARx) registers, transfer configuration (HCTSIZx) registers/interrupt (HCINTx) registers, and its associated interrupt mask register (HCINTMSKx).

Host Channel Control

The application can control the host channel as follows through the host channel x characteristics register (HCCHARx):

- Channel enable/disable
- Set the speed of the target USB device: FS/LS
- Set the address of the target USB device
- Sets the number of the endpoint on the target USB device that communicates with this channel
- Sets the transfer direction on this channel: IN/OUT

- Set the type of USB transfer on this channel: control/bulk/interrupt/isochronous
- Set the maximum packet length for device endpoints communicating with this channel
- Set the frame to be transmitted periodically: odd-numbered frame/even frame

Host Channel Transport

The host channel transfer size registers (HCTSIZE_x) allow the application to program the transfer size parameters and read the transfer status. Setting this register must be done before setting the channel enable bit in the host channel characteristics register. After the endpoint is enabled, the packet count field immediately becomes read-only, and the USBFS module updates this field according to the current transmission status.

The following transfer parameters can be programmed:

- Transfer size in bytes
- The number of packets that make up the entire transfer size
- Initial Data PID

Host Channel Status/Interrupt

The Host Channel x Interrupt Register (HCINT_x) indicates the status of the endpoint on USB and AHB related events. When the host channel interrupt bit in the interrupt register (HCINT bit in USBFS_GINTSTS) is set, the application must read these registers for detailed information. Before reading these registers, the application must first read the host collective channel interrupt (HCAINT) register to obtain the channel number of the host channel x interrupt register. The application must clear the corresponding bits in this register to clear the corresponding bits in the HCAINT and GINTSTS registers. The USBFS_HCINTMSK_x registers also provide mask bits for each interrupt source per channel.

The host module provides the following status checking and interrupt generation functions:

- Transfer complete interrupt, indicating that both the application (AHB) and the USB side have completed the data transfer
- Channel stopped due to transfer complete, USB transaction error, or application issued a prohibit command
- The associated transmit FIFO is half empty or fully empty (IN endpoint)
- ACK response received
- NAK response received
- STALL response received
- USB transaction errors due to CRC check failures, timeouts, bit stuffing errors, and incorrect EOPs
- Crosstalk error
- Frame overflow
- Toggle bit error for data synchronization

33.5.3.7 Host Scheduler

The host module has a built-in hardware scheduler, which can independently reorder and manage the USB transaction requests sent by the application. At the beginning of each frame, the host performs periodic (synchronous and interrupt) transactions followed by aperiodic (control and bulk) transactions to comply with the USB specification's guarantee of high priority for isochronous and interrupt transfers.

The host handles USB transactions through request queues (one periodic request queue and one aperiodic request queue). Each request queue can store up to 8 entries. Each entry represents a USB transaction request initiated by an application program but has not yet been responded to, and stores the number of the IN or OUT channel used to execute the USB transaction, as well as other related information. The order in which USB transaction requests are written in the queue determines the order in which transactions are executed on the USB interface.

At the beginning of each frame, the host processes the periodic request queue first and then the aperiodic request queue. If an isochronous or interrupt-type USB transfer transaction request scheduled for the current frame is still pending at the end of the current frame, the host will issue an outstanding periodic transfer interrupt (IPXFR bit in USBFS_GINTSTS). The USBFS module is responsible for the management of periodic and aperiodic request queues. Both the periodic transmit FIFO and queue status register (HPTXSTS) and the aperiodic transmit FIFO and queue status register (HNPTXSTS) are read-only registers, and applications can use them to read the status of each request queue. These include:

- The number of free entries currently available in the periodic (non-periodic) request queue (up to 8)
- Free space currently available in the periodic (aperiodic) TxFIFO (OUT transaction)
- IN/OUT tokens, host channel number, and other status information

Since each request queue can store up to 8 USB transaction requests, the application can send the host USB transaction request to the scheduler in advance; Appears on the USB bus after an acyclic transaction completes.

To issue a transaction request to the host scheduler (queue), the application must read the PTXQSAV bit in the USBFS_HPTXSTS register or the NPTQXSAV bit in the USBFS_HNPTXSTS register, ensuring that at least one of the periodic (non-periodic) request queues has space available to store the current ask.

33.5.4 USBFS Device Capabilities

33.5.4.1 Introduction to Device Functions

When USBFS works in device mode, VBUS is the 5V power supply pin stipulated by the USB protocol, and it is a 5V withstand voltage pin. This module always detects the level status of the VBUS line to connect or disconnect the device. USBFS_VBUS port must first pass through a resistor of more than 100 ohms before it can be connected to the VBUS power supply.

A typical USB device mode system construction diagram is as follows:

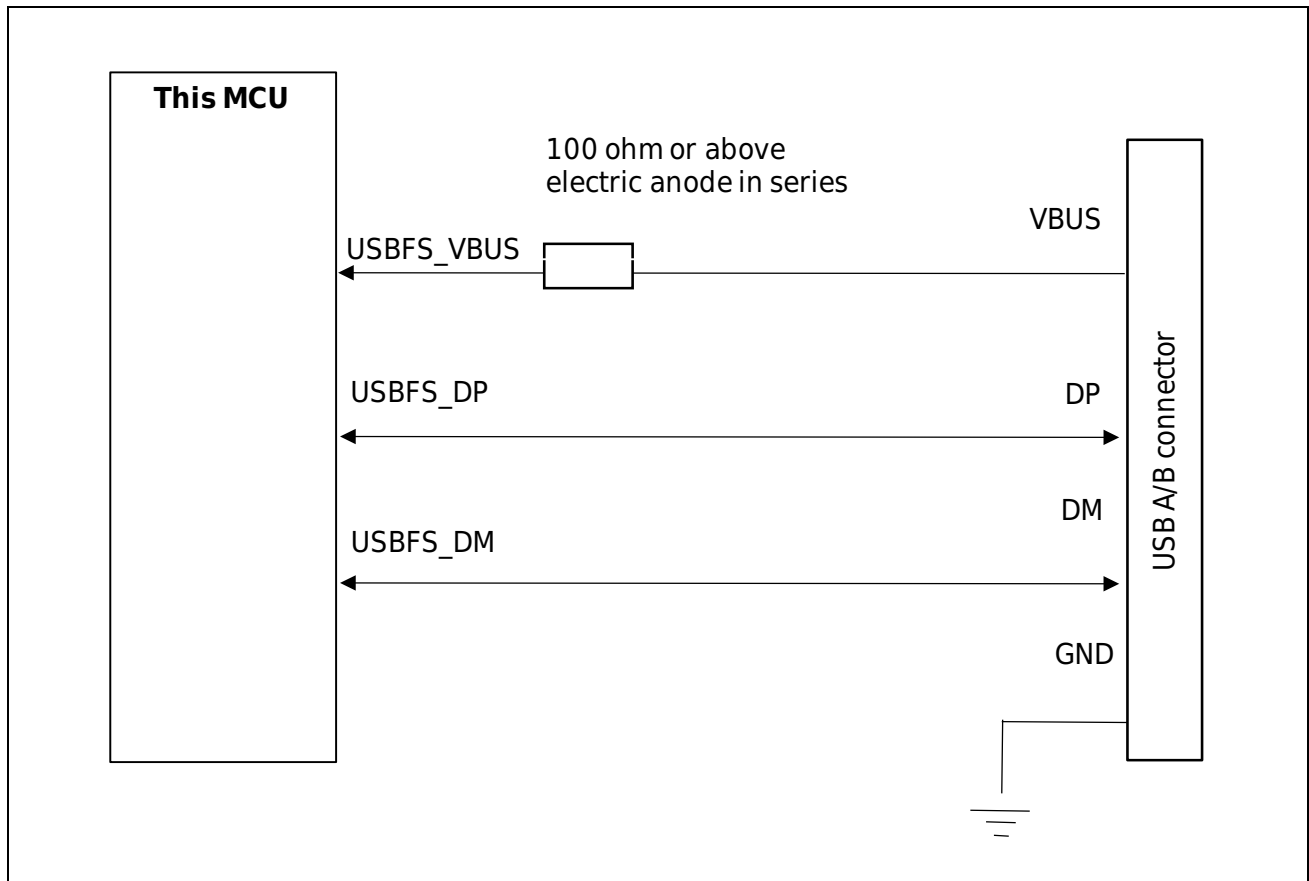


Figure 33-3 USBFS Device Mode System Construction Diagram

33.5.4.2 Device Power Status

When the module detects that USBFS_VBUS is high level, it will make the USB device enter the power supply state. Then, USBFS automatically connects the DP pull-up resistor, sends a signal that the full-speed device is connected to the host and generates a session request interrupt (VBUSVINT bit in USBFS_GINTSTS), indicating to enter the power supply state.

In addition, the USBFS_VBUS input ensures that the host provides a valid VBUS level during USB operation. USBFS will automatically disconnect if it detects that the VBUS level is low (for example, caused by power disturbance or host port shutdown).

In the powered state, USBFS expects to receive a reset signal from the host. Other USB operations cannot be performed. As soon as the reset signal is received, a reset detected interrupt (USBRST in USBFS_GINTST) is generated. After the reset signal ends, an enumeration complete interrupt (ENUMDNE bit in USBFS_GINTSTS) will be generated, and USBFS will then enter the default state.

33.5.4.3 Device Default State

By default, USBFS expects to receive a SET_ADDRESS command from the host. Other USB operations cannot be performed. When a valid SET_ADDRESS command is decoded on the USB, the application will write the corresponding address value to the device address field in the device configuration register (DAD bit in USBFS_DCFG). The USBF then enters the address state and is ready to answer host transactions with the configured USB address.

33.5.4.4 Device Suspend State

The USBFS device continuously monitors USB activity. After the USB idle time reaches 3ms, an early suspend interrupt (ESUSP bit in USBFS_GINTSTS) will be issued, and the device will enter the suspend state confirmed by the suspend interrupt (USBSUSP bit in USBFS_GINTSTS) after 3ms. Then, the device suspend bit in the device status register (SUSPSTS bit in USBFS_DSTS) is automatically set to 1, and USBFS then enters the suspend state.

Suspend can be exited through the device itself. In this case, the application will set the remote wakeup signal bit in the device control register (RWUSIG bit in USBFS_DCTL) to 1, and clear it to 0 within 1ms to 15ms.

However, if the device detects the resume signal sent by the host, it will generate a resume interrupt (WKUPINT bit in USBFS_GINTSTS), and the device suspend bit will be automatically cleared.

33.5.4.5 Device Soft Disconnect

The powered state can be exited by software with the soft-disconnect function. The DP pull-up resistor can be removed by setting the soft disconnect bit (SDIS bit in USBFS_DCTL) in the device control register to 1. At this time, although the USB cable is not actually pulled out from the host port, the device disconnection will still occur on the host side Detect interruption.

33.5.4.6 Device Endpoint

Endpoint Class

The USBFS module implements the following USB endpoints:

- Control endpoint 0:
 - Bi-directional and only handles control messages
 - Use a separate set of registers for IN and OUT transactions
 - Dedicated Control (USBFS_DIEPCTL0/ USBFS_DOEPCTL0) register, Transfer Configuration (USBFS_DIEPTSIZ0/ USBFS_DIEPTSIZ0) register, and Status Interrupt (USBFS_DIEPINTx/

USBFS_DOEPINT0) register. The set of bits available in the control and transfer size registers is slightly different from other endpoints

- 5 IN endpoints
 - Each endpoint can be configured to support isochronous, bulk, or interrupt transfer types
 - Each endpoint has dedicated control (USBFS_DIEPCTLx) registers, transfer configuration (USBFS_DIEPTSIZx) registers, and status interrupt (USBFS_DIEPINTx) registers
 - The device IN endpoint generic interrupt mask register (USBFS_DIEPMSK) can be used to enable/disable the same class of endpoint interrupt sources on all IN endpoints (including EP0)
 - Supports incomplete isochronous IN transfer interrupt (IISOIXFR bit in USBFS_GINTSTS), which will trigger when at least one transfer on an isochronous IN endpoint is incomplete in the current frame. This interrupt is triggered together with the periodic frame interrupt (USBFS_GINTSTS/EOPF)
- 5 OUT endpoints
 - Each endpoint can be configured to support isochronous, bulk, or interrupt transfer types
 - Each endpoint has dedicated control (USBFS_DOEPCTLx) registers, transfer configuration (USBFS_DOEPTSIZx) registers, and status interrupt (USBFS_DOEPINTx) registers
 - The device OUT endpoint general interrupt mask register (USBFS_DOEPMSK) can be used to enable/disable the same type of endpoint interrupt source on all OUT endpoints (including EP0)
 - Support for incomplete isochronous OUT transfer interrupts (INCOMPISOOOUT bit in USBFS_GINTSTS), which will trigger when at least one transfer on an isochronous OUT endpoint is incomplete in the current frame. This interrupt is triggered together with the periodic frame interrupt (USBFS_GINTSTS/EOPF)

Endpoint Control

The application can take the following control over the endpoint through the device endpoint x IN/OUT control registers (DIEPCTLx/DOEPCTLx):

- Endpoint enable/disable
- Activate the endpoint under the current configuration
- Set USB transfer type (isochronous, bulk and interrupt)
- Set the supported packet size
- Sets the Tx-FIFO number associated with the IN endpoint
- Set the data0/data1 PID you want to receive or use when sending (bulk/interrupt transfers only)
- Set the odd/even frame for receiving or sending transactions (only for synchronous transfers)
- The NAK bit can be set, so that regardless of the state of the FIFO at this time, a NAK is replied to the host's request

- The STALL bit can be set so that the host's tokens for this endpoint are all replied to STALL by the hardware
- The OUT endpoint can be set to listen mode, that is, no CRC check is performed on the received data

Endpoint Transport

The Device Endpoint x transmit size registers (DIEPTSIZx/DOEPTSIZx) allow the application to program the transfer size parameters and read the transfer status. Setting this register must be done before setting the Endpoint Enable bit in the Endpoint Control register. After the endpoint is enabled, these fields become read-only immediately, and the USBFS module updates these fields according to the current transmission status.

The following transfer parameters can be programmed:

- Transfer size in bytes
- The number of packets that make up the entire transfer

Endpoint Status/Status

The device endpoint x interrupt registers (DIEPINTx/DOPEPINTx) indicate the status of the endpoint on USB and AHB related events. When the OUT endpoint interrupt bit or the IN endpoint interrupt bit (OEPINT bit in USBFS_GINTSTS or IEPINT bit in USBFS_GINTSTS, respectively) is set in the module interrupt register, the application must read these registers for detailed information. Before an application can read these registers, it must first read the Device Overall Endpoint Interrupt (USBFS_DAINTE) register to obtain the endpoint number of the device endpoint x interrupt register. The application must clear the corresponding bits in this register to clear the corresponding bits in the DAINTE and GINTSTS registers.

The module provides the following status checking and interrupt generation functions:

- Transfer complete interrupt, indicating to the application that both the AHB and the USB side have completed the data transfer
- Setup phase completed (only for OUT endpoints of control transfer type)
- The associated transmit FIFO is half empty or fully empty (IN endpoint)
- A NAK reply has been sent to the host (IN endpoint for isochronous transfers only)
- IN token received while TxFIFO is empty (only for IN endpoints of bulk and interrupt transfer types)
- OUT token received when the endpoint has not been enabled
- babble error detected
- Application shutdown endpoint takes effect
- The application sets NAK on the endpoint to take effect (only for the IN endpoint of the synchronous transmission type)
- Received more than 3 consecutive setup packets (only for control type OUT endpoints)

- A timeout condition was detected (only for IN endpoints of control transfer type)

33.5.5 USBFS SOF Pulse Pin Output Function

USBFS can monitor, track and configure SOF frames in both host and device modes and also has SOF pulse output function. The SOF pulse is output through the USBFS_SOF pin, and the output width is 16 system clock periods.

33.5.5.1 Host SOF

In host mode, the number of PHY clocks that occur between two consecutive SOF (FS) or keep-alive (LS) tokens generated can be programmed in the Host Frame Interval Register (HFIR), allowing the application to program the SOF frame period. An interrupt is generated at the start of a frame (SOF bit in USBFS_GINTSTS). The current frame number and the time remaining until the next SOF can be tracked by the application in the host frame number register (HFNUM).

Using the SOFEN bit in the USBFS system control register USB_SYCTLREG, the SOF pulse signal with a width of 16 system clock periods generated at the same time as any SOF token is issued can be output from the USBFS_SOF pin.

In addition, the SOF pulse can also work as an internal event triggering DMA transfer, TIMER counting and other external modules.

33.5.5.2 Device SOF

In device mode, each time the USB receives a SOF token, it will trigger a start-of-frame interrupt (SOF bit in USBFS_GINTSTS). The corresponding frame number can be read from the Device Status Register (FNSOF bit in USBFS_DSTS). Using the SOFEN bit in the USBFS system control register USB_SYCTLREG, you can also generate a SOF pulse signal with a width of 16 system clock periods, and make the signal output on the USBFS_SOF pin for external availability.

In addition, the SOF pulse can also work as an internal event triggering DMA transfer, TIMER counting and other external modules.

The periodic end-of-frame interrupt (GINTSTS/EOPF) is used to notify the application when 80%, 85%, 90%, or 95% of the frame interval time has elapsed, depending on the periodic frame interval field in the device configuration register (USBFS_DCFG PFIVL bit in the). This function can be used to determine if all isochronous communications for that frame are complete.

33.5.6 USBFS Power Control

When the USBFS module is not used, the HCLK and PHY clocks of the USBFS module can be stopped through the CMU module to reduce power consumption.

Power reduction techniques can be used in the USB suspend state when the USB module is used but the device USB session is not started or the device is not connected.

■ Stop the PHY clock (STPPCLK bit in USBFS_GCCTL)

Most of the 48 MHz internal clock domains of the USBFS full-speed module are clock-gated off when the Stop PHY Clock bit in the Clock-Gating Control Register is set to 1. Even if the application still provides the clock input, it will save the dynamic power consumption of the module due to the flipping of the clock signal. It will also turn off most of the units of the transceiver, and only the part responsible for detecting an asynchronous recovery event or a remote wake-up event will remain active..

■ HCLK gating (GATEHCLK bit in USBFS_GCCTL)

When the GATEHCLK bit in the Clock Gating Control Register is set to 1, most of the system clock domains inside the USBFS module are clock-gated off. Only the register read and write interfaces remain active. Even if the application still provides the clock input, it will save the module's dynamic power consumption due to the clock signal flipping.

To save dynamic power, the USB data FIFO is only clocked when it is accessed by the USBFS module.

33.5.7 USBFS Dynamically Updates the USBFS_HFIR Register

In the host mode, the USB module has the function of dynamically fine-tuning the frame period, and can synchronize the external device with the SOF frame. If the USBFS_HFIR register changes within the current SOF frame, the SOF period will be corrected accordingly in the next frame, see Figure 33-4 for details.

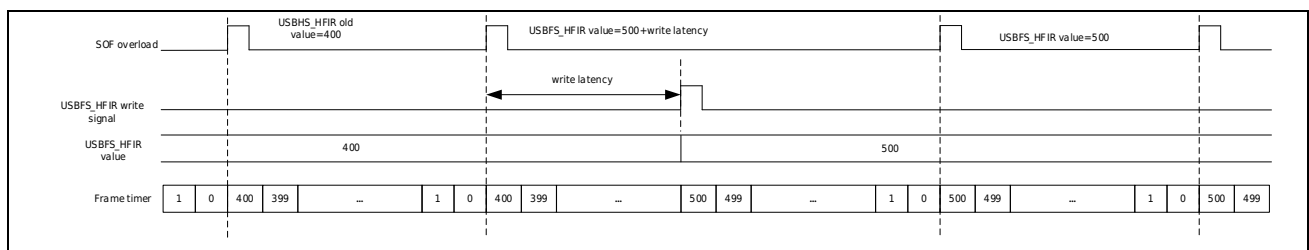


Figure 33-4 Schematic Diagram Of USBFS Dynamically Updating USBFS_HFIR Register

33.5.8 USBFS Data FIFO

The USBFS system has 2.5KB dedicated RAM and adopts an efficient FIFO control mechanism. The packet FIFO controller module in the USBFS module divides the RAM space into multiple TxFIFOs (before USB transmission, the application program pushes data into it for short-term storage) and a single Rx FIFO (before the data received from the USB is read by the application program), where transient storage takes place).

The number and organization of FIFOs built in RAM depends on the role of the device. In device mode, one TxFIFO is configured for each active IN endpoint. The sizes of the FIFOs are all software configurable to better meet application requirements.

33.5.9 USBFS Host FIFO Architecture

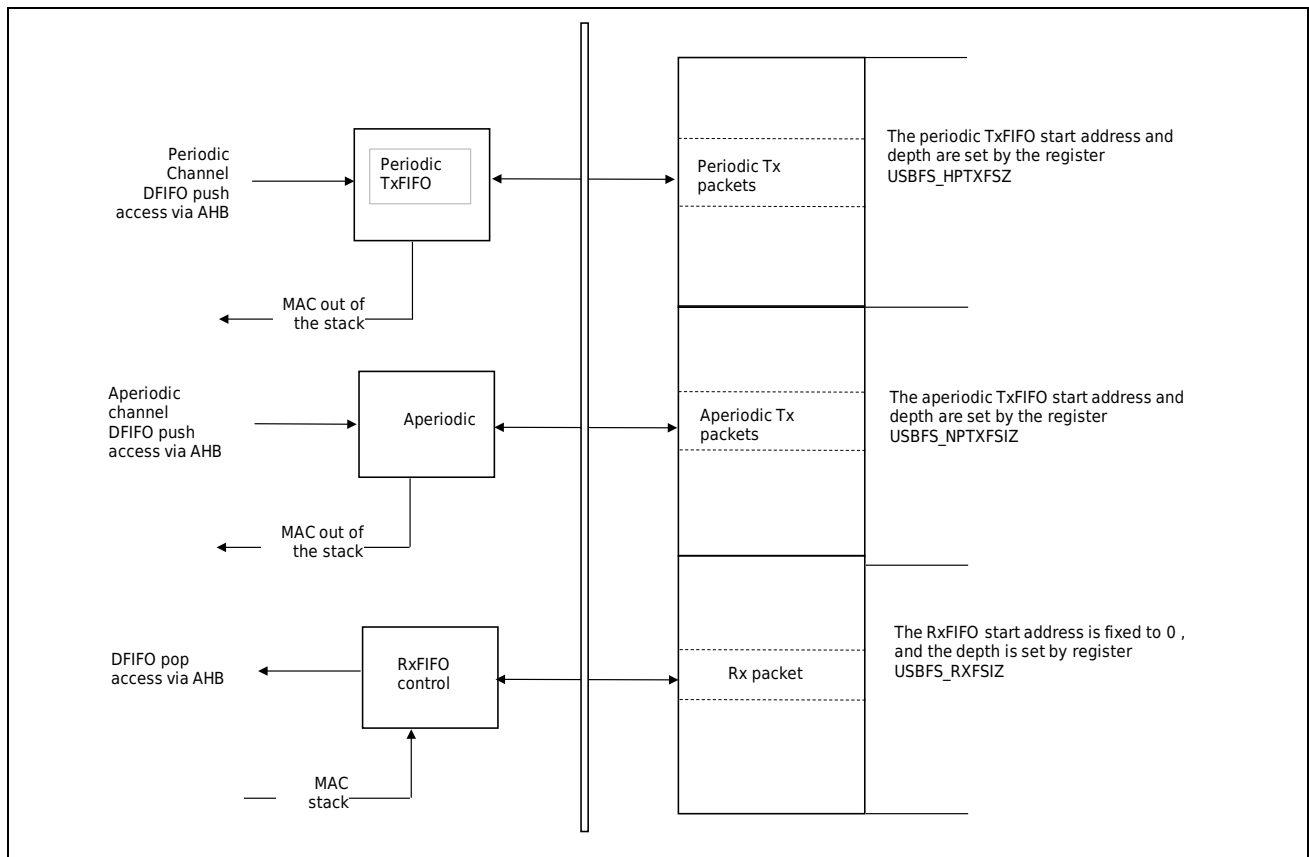


Figure 33-5 Schematic Diagram of FIFO Architecture in USBFS Host Mode

33.5.9.1 Host RxFIFO

The host uses a receive FIFO for all periodic and aperiodic transactions. The FIFO is used as a receive buffer to hold data received from the USB (data portion of a received packet) until it is transferred to system memory. Packets from the IN endpoint of the device are received and stored one by one as long as there is room in the FIFO. The status of each packet received (including host target channel, byte count, data PID, and checksum on the received data) is also stored in the FIFO. The size of the receive FIFO is configured in the receive FIFO size register (GRXFSIZ).

A single receive FIFO architecture enables the USB host to efficiently fill the receive data buffer:

- All IN configuration host channels share the same RAM buffer (shared FIFO)
- The USBFS module can fill the receive FIFO to the limit for any sequence of IN tokens driven by host software

The application receives an Rx FIFO not empty interrupt as soon as at least one packet is available for reading in the Rx FIFO. The application program reads the packet information from the receive status read and pop registers, and finally reads the data from the Rx FIFO.

33.5.9.2 Host TxFIFO

The host uses one transmit FIFO for all aperiodic (control and bulk) OUT transactions and the other transmit FIFO for all periodic (synchronous and interrupt) OUT transactions. The FIFO is used as a transmit buffer to hold data to be sent over USB (transmit packets). The periodic (aperiodic) TxFIFO size is configured in the host periodic (aperiodic) transmit FIFO size (HPTXFSIZ/HNPTXFSIZ) register.

The two Tx FIFOs are operated according to the priority, and the priority of the periodic communication is higher, so the periodic communication is performed first in the time of one USB frame. At the beginning of a frame, the built-in host scheduler processes the periodic request queue first, and then the aperiodic request queue.

The architecture of two transmit FIFOs enables the USB host to optimize the management of periodic and aperiodic transmit data buffers:

- All host channels configured to support periodic (aperiodic) OUT transactions share the same RAM buffer (shared FIFO)
- For arbitrary sequences of OUT tokens driven by host software, the USBFS module can fill the periodic (aperiodic) transmit FIFO to the limit

The USBFS module issues a periodic TxFIFO empty interrupt (PTXFE bit in USBFS_GINTSTS) whenever the periodic TxFIFO is half empty or fully empty, depending on the Periodic Tx-FIFO Empty Level bit in the AHB configuration register (PTXFELVL bit in USBFS_GAHBCFG) value. As long as there is free space in both the periodic TxFIFO and the periodic request queue, the application can write transmit data ahead of time. The available space of both can be known by reading the host periodic transmit FIFO and queue status register (HPTXSTS).

The USBFS module issues an aperiodic TxFIFO empty interrupt (NPTXFE bit in USBFS_GINTSTS) whenever the aperiodic TxFIFO is half empty or fully empty, depending on the aperiodic TxFIFO empty level bit in the AHB configuration register (TXFELVL in USBFS_GAHBCFG bit). The application can write transmit data as long as there is free space in both the aperiodic TxFIFO and the aperiodic request queue. The available space of both can be known by reading the host aperiodic transmit FIFO and queue status register (HNPTXSTS).

33.5.10 USBFS Device FIFO Architecture

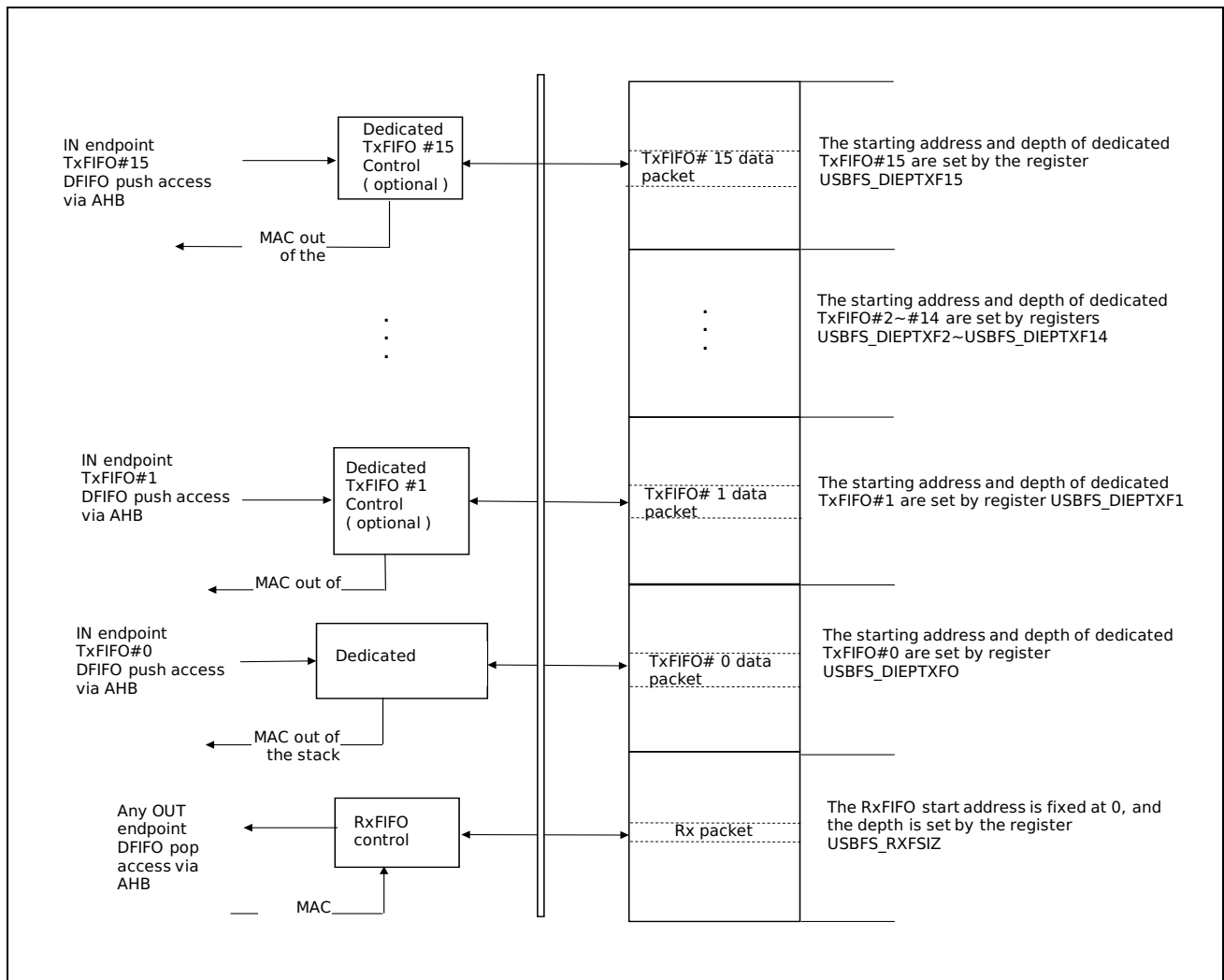


Figure 33-6 Schematic Diagram of FIFO Architecture in USBFS Device Mode

33.5.10.1 Device Rx FIFO

USBFS devices use a single receive FIFO to receive data sent to all OUT endpoints. As long as there is free space in the Rx FIFO, the received data packets are filled into the Rx FIFO one by one. In addition to valid data, received packet status (containing OUT endpoint destination number, byte count, data PID, and validation of received data) is also stored by the module. When no space is available, the device replies with a transaction NAK to the host and triggers an interrupt on the addressed endpoint. The size of the receive FIFO is configured in the receive FIFO size register (GRXFSIZ).

A single receive FIFO architecture enables the USB device to more efficiently fill the receive RAM buffer:

- All OUT endpoints share the same RAM buffer (shared FIFO)
- The USBFS module can fill the receive FIFO to the limit for any host serial OUT token

The application will always receive the RxFIFO not empty interrupt (RXFNE bit in USBFS_GINTSTS) as long as at least one packet is available for reading in the RxFIFO. The application program reads the packet information from the receive status read and stack register (GRXSTSP), and finally reads the corresponding data from the receive FIFO by reading the stack address related to the endpoint.

33.5.10.2 DeviceTxFIFO

The module provides a dedicated FIFO for each IN endpoint. The application configures the FIFO size for IN endpoint 0 through the aperiodic transmit FIFO size register (USBFS_DIEPTSIZE0); configures the FIFO size for IN endpoint x through the device IN endpoint transmit FIFOx register (DIEPTSIZEx).

33.5.11 USBFS FIFO RAM Allocation

33.5.11.1 Host Mode

Receive FIFO RAM Allocation

Status information is written to the FIFO with each received packet. Therefore, at least $(\text{maximum packet size} / 4) + 1$ space must be allocated for received packets. If multiple isochronous channels are enabled, the space allocated for receiving consecutive packets must be at least twice $(\text{maximum packet size} / 4) + 1$. Usually, the recommended space is twice the size of $(\text{maximum packet} / 4 + 1)$, so that when the previous packet is sent to the CPU, the USB can receive subsequent packets at the same time.

The transfer completion status information is written to the FIFO along with the last packet received by this endpoint. So a place has to be allocated for this.

When running in DMA mode, the DMA address registers of each host channel will be stored in the FIFO, so a place needs to be reserved in the FIFO for each channel to store its address registers.

Transmit FIFO RAM allocation

The minimum RAM required for the host aperiodic transmit FIFO is the size of the largest packet transmitted on all supported aperiodic OUT channels.

Usually, the recommended space is twice the maximum packet size, so that the AHB can fill the transmit FIFO with the next packet while the USB is sending the current packet.

The minimum RAM required for the host aperiodic transmit FIFO is the size of the largest packet transmitted on all supported aperiodic OUT channels. If there is at least one isochronous OUT endpoint, the space must be at least twice the size of the largest packet in that channel.

When running in DMA mode, the DMA address registers of each host channel will be stored in the FIFO, so a place needs to be reserved in the FIFO for each channel to store its address registers.

33.5.11.2 Device Mode

Receive FIFO RAM Allocation

The application should allocate RAM for SETUP packets: 11 slots must be reserved in the receive FIFO to receive SETUP packets on the control endpoint. The USBFS module will not write any other data to these locations reserved for SETUP packets. A location will be allocated for the global OUT NAK. Status information is written to the FIFO with each received packet. Therefore, at least $(\text{maximum packet size}/4)+1$ space must be allocated for received packets. If multiple isochronous endpoints are enabled, the space allocated for receiving consecutive packets must be at least twice $(\text{maximum packet size}/4)+1$. Usually, the recommended space is twice the size of $(\text{maximum packet}/4 + 1)$, so that when the previous packet is sent to the CPU, the USB can receive subsequent packets at the same time.

The transfer completion status information is pushed into the FIFO along with the last packet received by this endpoint. In general, it is recommended to assign a location for each OUT endpoint.

Transmit FIFO RAM Allocation

The minimum RAM space required for each IN endpoint's transmit FIFO is the maximum packet size for that particular IN endpoint.

33.5.12 USBFS System Performance

With large RAM buffer, highly configurable FIFO size, fast 32-bit FIFO access via AHB push/pop registers, and especially advanced FIFO control mechanism for optimal USB and system performance. In fact, regardless of the current USB sequence, USBFS can efficiently fill the available RAM space through this mechanism. With these features:

- The application has enough margin to calculate and correct the CPU load to optimize CPU bandwidth utilization:
 - The application program can accumulate a large amount of sending data first, and then send it out via USB
 - Allows sufficient time margin to read data from the receive FIFO
- The USB module can keep working at full speed, that is, to provide the maximum full-speed bandwidth (as much hardware as possible to run automatically, and as little software as possible to participate)
 - The USB module can accumulate a large amount of sending data in advance at its disposal, so that the USB data sending can be autonomously managed
 - There is a lot of empty space in the receive buffer, which can be automatically filled with data from the USB

Because the USBFS module can efficiently fill the 1.25KB RAM buffer and 1.25KB send/receive data is enough to meet the amount of data that can be accommodated in a full-speed frame, the USB

system can reach the maximum USB bandwidth within one frame without application program intervention.

33.5.13 USBFS Interrupt and Events

There are two types of USBFS interrupts, SOTP mode wake-up interrupt USBFS_WKUP and USBFS global interrupt USBFS_GLB.

USBFS_WKUP Interrupt

The USBFS_WKUP interrupt is used in STOP mode to wake up the system's STOP mode through USBFS_DP or USBFS_DM. The interrupt enable bit is INT_WUPEN.USF_WUEN.

using USBFS to wake up the system STOP , you need to ensure that the USBFS controller is in the suspended state, and set the corresponding filtering range and enable the filter function in the register USB_SYCTLREG.

USBFS_GLB Interrupt

USBFS_GLB interrupt is the main interrupt that the software needs to handle. The flag of the global interrupt can be read in the USBFS_GINTSTS register.

Table 33-2 USBFS_GLBInterrupt Event Table

Interrupt Flag	Description	Operating Mode	Internal Event Source
WKUPINT	Resume/Remote Wakeup Interrupt	Host or device	-
VBUSVINT	VBUS active interrupt	Equipment	-
DISCINT	Disconnect interrupt	Host	-
CIDSCHG	Connector ID Line Status Change Interrupt	Host or device	-
PTXFE	Periodic TxFIFO empty interrupt	Host	-
LPMINT	LPM Interrupt	Host or device	
HCINT	Host channel interrupt	Host	-
HPRTINT	Host port interrupt	Host	-
DATAFSUSP	Data fetch pending	Equipment	-
IPXFR/INCOMPISOOUT	Incomplete periodic transmission / outstanding OUT synchronous transmission	Equipment	-
IISOIXFR	IN SYNC TRANSFER NOT COMPLETED	Equipment	-
OEPINT	OUT endpoint interrupt	Equipment	-
IEPINT	IN endpoint interrupt	Equipment	-
EOPF	Periodic end-of-frame interrupt	Equipment	-
ISOODRP	Drop sync OUT packet interrupt	Equipment	-
ENUMDNE	Enumeration complete	Equipment	-
USB_RST	USB reset interrupt	Equipment	-
USBSUSP	USB suspend interrupt	Equipment	-

Interrupt Flag	Description	Operating Mode	Internal Event Source
ESUSP	Early hang interrupt	Equipment	-
GONAKEFF	Global OUT NAK active interrupt	Equipment	-
GINAKEFF	Global aperiodic IN NAK active interrupt	Equipment	-
NPTXFE	Aperiodic TxFIFO empty interrupt	Host	-
RXFNE	RxFIFO not empty interrupt	Host or device	-
SOF	Start of frame interrupt	Host or device	Yes
MMIS	Pattern mismatch interrupt	Host or device	-

33.6 USBFS Programming Model

33.6.1 USBFS Module Initialization

The application must perform the module initialization sequence.

Please refer to [USBFS Mode Decision] for the mode determination method.

This section describes the initialization process after the USBFS controller is powered on. Whether working in host mode or device mode, an application must follow an initialization sequence. Initialize all module global registers according to the module configuration:

- Program the following fields in the USBFS_GAHBCFG register:
 - Global interrupt mask bit GINTMSK = 1
 - RxFIFO is not empty (RXFNE bit in USBFS_GINTSTS)
 - Periodic TxFIFO Empty Threshold
- Program the following fields in the USBFS_GUSBCFG register:
 - FS timeout calibration field
 - USB turnaround time field
- Software must unmask the following bits in the USBFS_GINTMSK register:
 - Pattern Mismatch Interrupt Mask
- By reading the CMOD bit in USBFS_GINTSTS, software can determine whether the USBFS controller is operating in host mode or device mode.

33.6.2 USBFS Host Initialization

To initialize a module as a host, an application must perform the following steps:

1. Program the HPRTIINT in the USBFS_GINTMSK register to unmask.
2. Program the USBFS_HCFG register to select a full-speed host.
3. Program the PPWR bit in USBFS_HPRT to 1 to provide VBUS to the USB bus.
4. Wait for PCDET interrupt in USBFS_HPRT. This indicates that a device is connected to the host port.
5. Programming the PRST bit in USBFS_HPRT to 1 issues a reset signal on the USB bus.

6. Wait at least 10ms for the reset process to complete.
7. Program the PRST bit in USBFS_HPRT to 0.
8. Wait for PENCHNG interrupt in USBFS_HPRT.
9. Read the PSPD bits in USBFS_HPRT to get the enumeration speed.
10. With the selected PHY clock, set the HFIR register accordingly.
11. Program the FSLSPCS field in the USBFS_HCFG register according to the device speed detected in step 9. If the FSLSPCS changes, a port reset must be performed.
12. Program the USBFS_GRXFSIZ register to select the receive FIFO size.
13. Program the USBFS_HNPTXFSIZ register to select the size and starting address of the aperiodic transmit FIFO used for aperiodic communication transactions.
14. Program the USBFS_HPTXFSIZ register to select the size and starting address of the periodic communication transmit FIFO for periodic transactions.

To communicate with the device, system software must initialize and enable at least one channel.

33.6.3 USBFS Device Initialization

During power-up or after switching from host mode to device mode, the application must perform the following steps to initialize the module as a device.

1. Program the following fields in the USBFS_DCFG register:
 - Device speed
 - Non-zero length status OUT handshake signal
2. Program the USBFS_GINTMSK register to unmask the following interrupts:
 - USB reset
 - Enumeration complete
 - Early hang
 - USB suspend
 - SOF
3. Wait for the VBUSVINT interrupt in USBFS_GINTSTS, which means entering the power supply state.
4. Wait for the USBRST interrupt in USBFS_GINTSTS. This indicates that a reset signal has been detected on the USB, and the reset process lasts about 10ms from the reception of this interrupt.
5. Wait for ENUMDNE interrupt in USBFS_GINTSTS. This interrupt signals the end of the reset process on the USB. When this interrupt is received, the application must read the USBFS_DSTS register to determine the enumeration speed and perform the steps listed in Endpoint Initialization on Enumeration Completion.

At this point, the device is ready to accept SOF packets and perform control transfers on control endpoint 0.

33.6.4 USBFS DMA Mode

USB uses the AHB master interface for sending packet data (AHB to USB) and receiving data updates (USB to AHB). The AHB master interface uses programmed DMA addresses (HCDMAx registers in host mode and DIEPDMAx/DOEPDMAx registers in device mode) to access data buffers.

33.6.5 USBFS Host Programming Model

33.6.5.1 Channel Initialization

An application must initialize one or more channels before it can communicate with a connected device.

To initialize and enable a channel, the application must perform the following steps:

1. Program the USBFS_GINTMSK register to unmask the interrupt on the following bits:
 - The aperiodic transmit FIFO used for OUT transactions is empty (applicable when working at the pipeline transaction level and the packet count field is programmed with a value greater than 1).
 - The aperiodic transmit FIFO used for OUT transactions is empty (applicable when working at the pipeline transaction level and the packet count field is programmed with a value greater than 1).
2. Program the USBFS_HAINTMSK register to enable interrupts for the selected channel.
3. Program the USBFS_HCINTMSK register to enable interrupts related to communication transactions reflected in the host channel interrupt register.
4. Program the USBFS_HCTSIZx registers for the selected channel, specifying the total transfer size in bytes and the expected number of packets including short packets. The application must program the PID field with the initial data PID (used for the first OUT transaction or expected from the first IN transaction).
5. Program the USBFS_HCCHARx registers for the selected channel, specifying the device's endpoint characteristics such as type, speed, direction, etc. (A channel can be enabled by setting the channel enable bit only when the application is ready to send or receive packets).

33.6.5.2 Channel Stop

The application can disable any channel by programming the USBFS_HCCHARx registers by setting the CHDIS and CHENA bits to 1. This causes the USBFS host to clear any previous requests made on that channel (if any) and generate a channel stop interrupt. The application must wait for the CHH interrupt in USBFS_HCINTx before reassigning the channel to other communication transactions. The USBFS host will not interrupt communication transactions already initiated on the USB.

Before disabling a channel, the application must ensure that there is free space in at least one of the aperiodic request queue (when disabling aperiodic channels) or the periodic request queue (when disabling periodic channels). The application can clear the request queue by programming the USBFS_HCCHARx register to set the CHDIS bit to 1 and clear the CHENA bit to 0 when the request queue is full (before disabling the channel). The application will ban the channel when any of the following occurs:

1. STALL, TXERR, BBERR or DTERR interrupt received in USBFS_HCINTx of IN or OUT channel. The application must be able to receive other interrupts (DTERR, Nak, Data, TXERR) on the same channel before receiving the channel stop signal.
2. Received a DISCINT (disconnect device) interrupt in USBFS_GINTSTS. (Application will disable all enabled channels).
3. The application aborted the transfer before it could complete normally.

In DMA mode, the application cannot stop the indivisible periodic transfer by overwriting the register.

33.6.6 USBFS Device Programming Model

33.6.6.1 Endpoint Initialization on USB Reset

1. Set the NAK bit to 1 for all OUT endpoints
 - In USBFS_DOEPCTLx, SNAK = 1 (for all OUT endpoints)
2. Unmask the following interrupt bits
 - In USBFS_DAINMSK, INEP0=1 (Control 0 IN endpoint)
 - In USBFS_DAINMSK, OUTEP0=1 (control 0 OUT endpoint)
 - In DOEPMASK, STUP=1
 - In DOEPMASK, XFRC=1
 - In DIEPMASK, XFRC=1
 - In DIEPMASK, TOC=1
3. Set data FIFO RAM for each FIFO
 - Program the USBFS_GRXFSIZ register to be able to receive OUT data and setup data for control transfers. This register must be at least equal to 1 maximum packet size for control endpoint 0 + 2 words (for the status of the control OUT packet) + 10 words (for the SETUP packet).
 - Program the USBFS_TX0FSIZ register (depending on the selected FIFO number) to be able to send control IN data. This register must be at least equal to 1 maximum packet size for Control Endpoint 0.
4. Program the following fields in the endpoint-associated registers to control OUT endpoint 0 to receive SETUP packets
 - STUPCNT=3 in USBFS_DOEPTSIZ0 (receive up to 3 consecutive SETUP packets)

At this point, all initialization required to receive the SETUP packet is complete.

33.6.6.2 USB Enumeration Completes

1. In the enumeration complete interrupt (ENUMDNE in USBFS_GINTSTS), read the USBFS_DSTS register to determine the enumeration speed of the device.
2. Program the MPSIZ field in USBFS_DIEPCTL0 to set the maximum packet size. This step configures control endpoint 0. The maximum packet size for the control endpoint depends on the enumeration speed.

At this point, the device is ready to receive SOF packets and is configured to perform control transfers on control endpoint 0.

33.6.6.3 Endpoint Initialization when a SetAddress Command is Received

This section describes what an application must do when it receives a SetAddress command in a SETUP packet.

1. Program the USBFS_DCFG register with the device address received in the SetAddress command
2. Program the module to issue IN packets for the status phase

33.6.6.4 Endpoint Initialization when a Set Configuration/SetInterface Command is Received

This section describes what an application must do when it receives a SetConfiguration or SetInterface command in a SETUP packet.

1. When the SetConfiguration command is received, the application must program the endpoint registers to configure them with the characteristics of valid endpoints in the new configuration.
2. When a SetInterface command is received, the application must program the endpoint registers for the endpoint specified by the command.
3. An endpoint that was valid in a previous configuration or other setup is not valid in a new configuration or other setup. These invalid endpoints must be disabled.
4. Use the USBFS_DAINTRMSK register to enable interrupts for valid endpoints and mask interrupts for invalid endpoints.
5. Set data FIFO RAM for each FIFO.
6. After configuring all required endpoints, the application must program the module to send IN packets for the status phase.

At this point, the device module is ready to receive and send any type of data packet

33.6.6.5 Endpoint Activation

This section describes the steps required to activate a device endpoint or configure an existing device endpoint to a new type.

1. Program the characteristics of the desired endpoint in the following fields of the USBFS_DIEPCTLx register (for IN or bidirectional endpoints) or the USBFS_DOEPCTLx register (for OUT or bidirectional endpoints).
 - Maximum Packet Size
 - USB active endpoint location 1
 - Endpoint initial data sync bit (for interrupt and bulk endpoints)
 - Endpoint type
 - TxFIFO number
2. Once the endpoint is activated, the module starts decoding tokens sent to the endpoint and replies with a valid handshake if the received token is valid.

33.6.6.6 Endpoint Deactivation

This section describes the steps required to decommission an existing endpoint.

1. In the endpoint to be deactivated, clear the USB active endpoint bit in the USBFS_DIEPCTLx register (for IN or bidirectional endpoint) or USBFS_DOEPCTLx register (for OUT or bidirectional endpoint).
2. When an endpoint is deactivated, the module ignores tokens sent to it, causing a USB timeout.

33.6.7 USBFS Operational Model

33.6.7.1 SETUP and OUT Data Transfer

This section describes the internal data flow and application operation steps during data OUT transfers and SETUP transactions.

Packet Reading

This section describes how to read packets (OUT data and SETUP packets) from the receive FIFO.

1. When an RXFNE interrupt is caught (USBFS_GINTSTS register), the application must read the receive status pop register (USBFS_GRXSTSP).
2. The application can mask the RXFNE interrupt (in USBFS_GINTSTS) by writing RXFNE=0 (in USBFS_GINTMSK) until it reads the packet out of the receive FIFO.
3. If the received data packet has a non-zero byte count, it is popped from the received data FIFO and stored in memory. If the received packet byte count is 0, no data will be popped from the receive data FIFO.
4. The state of the data packet read from the receive FIFO has the following states:
 - Global OUT NAK:
PKTSTS = global OUT NAK, BCNT = 0x000, the values of EPNUM and DPID don't matter.
These data indicate that the global OUT NAK bit is in effect.
 - SETUP packet:

PKTSTS=SETUP, BCNT=0x008, EPNUM=control EP number, DPID=D0. These data indicate that SETUP packets received on the specified endpoint can now be read from the receive FIFO.

- The build phase is complete:

PKTSTS = setup phase complete, BCNT = 0x0, EPNUM = control EP number, DPID value doesn't matter.

These data indicate that the build phase for the specified endpoint is complete and the data phase has started. After this status entry has been popped from the receive FIFO, the module will generate a setup interrupt on this control OUT endpoint.

- OUT packet:

PKTSTS=DataOUT, BCNT=the size of the received OUT packet (BCNT:0~1024), EPNUM=the endpoint number of the received data packet, DPID=actual data PID.

- Data transfer complete:

PKTSTS = OUT data transfer complete, BCNT = 0x0, EPNUM = OUT EP number that completed data transfer, DPID value does not matter.

These data indicate the completion of OUT data transfer for the specified OUT endpoint. After this status entry has been popped from the receive FIFO, the module will raise a "transfer complete" interrupt on the specified OUT endpoint.

5. After popping data from the receive FIFO, the RXFNE interrupt must be unmasked (USBFS_GINTSTS).
6. Steps 1 to 5 will be repeated each time the application detects an RXFNE interrupt in USBFS_GINTSTS. Reading from an empty receive FIFO may result in undefined module behavior.

SETUP Transaction

This section describes how the module handles SETUP packets and the order in which applications process SETUP transactions.

Application Requirements:

1. To receive SETUP packets, the control OUT endpoint STUPCNT field (USBFS_DOEPTSIZx) must be programmed to a non-zero value. If the application programs the STUPCNT field to a non-zero value, the module receives the SETUP packet and writes it to the receive FIFO regardless of the NAK status and the setting of the EPENA bit in USBFS_DOEPCTLx. Every time the control endpoint receives a SETUP packet, the STUPCNT field will be decremented. If the STUPCNT field is not programmed to an appropriate value prior to receiving the SETUP packet, the module can still receive the SETUP packet and decrement the STUPCNT field, but the application may not be able to determine the correct number of SETUP packets to receive during the setup phase of a control transfer.
 - In USBFS_DOEPTSIZx, STUPCNT=3

2. The application must always allocate some extra space in the receive data FIFO to be able to receive up to three consecutive SETUP packets on the control endpoint.
 - Reserve space for 10 characters. The first SETUP packet requires 3 words, the "setup phase complete" status double word requires 1 word, and 6 words are required to store two additional SETUP packets.
 - Each SETUP packet requires 3 words to store 8 bytes of SETUP data and 4 bytes of SETUP status. The module will reserve these spaces in the receive FIFO.
 - This section of FIFO is only used to store SETUP packets, never use this space for data packets.
3. The application must read 2 words of the SETUP packet from the receive FIFO.
4. The application must read and discard the "setup phase complete" status word from the receive FIFO

Internal data flow:

1. When a SETUP packet is received, the module writes the received data into the receive FIFO without checking the available space in the receive FIFO and regardless of the NAK and STALL bit settings of the endpoint.
 - The module will internally set the IN NAK and OUTNAK bits of the control IN/OUT endpoint that received the SETUP packet to 1.
2. For each SETUP packet received on the USB, the module will write 3 words of data into the receive FIFO, and decrement the STUPCNT field by 1.
 - The first word contains internal control information used by the module
 - The second word contains the first 4 bytes of the SETUP command
 - The third word contains the last 4 bytes of the SETUP command
3. When the setup phase ends and the data IN/OUT phase begins, the module writes a status entry ("setup phase complete" word) to the receive FIFO indicating that the setup phase is complete.
4. On the AHB side, the SETUP packet is read by the application.
5. When the application pops the "setup phase complete" word from the receive FIFO, the module will interrupt the application with a STUP interrupt (USBFS_DOEPINTx), indicating that it can process the received SETUP packet.
 - The module will clear the endpoint enable bit that controls the OUT endpoint.

Application programming sequence:

1. Program the USBFS_DOEPTSIZE register.
 - STUPCNT=3
2. Wait for RXFNE interrupt (USBFS_GINTSTS) and read packet from receive FIFO.
3. The triggering of the STUP interrupt (USBFS_DOEPINTx) indicates the successful completion of the SETUP data transfer.

- When this interrupt occurs, the application must read the USBFS_DOEPTSIz registers to determine the number of SETUP packets received and process the last received SETUP packet.

Process more than three consecutive SETUP packets:

According to the USB2.0 specification, in a SETUP packet error, the host usually does not send more than 3 consecutive SETUP packets to the same endpoint. However, the USB2.0 specification does not limit the number of consecutive SETUP packets that the host can send to the same endpoint. When this happens, the USBFS controller will generate an interrupt (B2BSTUP in USBFS_DOEPINTx).

Set global OUT NAK to 1

Internal data flow:

1. If the application sets the global OUT NAK (SGONAK bit in USBFS_DCTL) to 1, the module will stop writing data other than SETUP packets into the receive FIFO. Regardless of the size of the available space in the receive FIFO, the device will reply NAK to the asynchronous OUT token sent by the host, and ignore the synchronous OUT packet directly.
2. The module writes the global OUT NAK to the receive FIFO. The application must allow enough space for this.
3. The module will set the GONAKEFF interrupt (USBFS_GINTSTS) when the application pops the global OUT NAK word from the receive FIFO.
4. After the application program detects this interrupt, it will consider the module to be in global OUT NAK mode. The application can clear this interrupt by clearing the SGONAK bit in USBFS_DCTL.

Application programming sequence:

1. To stop receiving any type of data into the receive FIFO, the application must set the global OUT NAK bit by programming the following fields.
 - In USBFS_DCTL, SGONAK=1
2. Wait for GONAKEFF interrupt in USBFS_GINTST. Once triggered, this interrupt indicates that the module has stopped receiving any type of data other than SETUP packets.
 - If the application has set the SGONAK bit in USBFS_DCTL, the application can receive valid OUT packets before the module raises a GONAKEFF interrupt (USBFS_GINTSTS).
3. The application can temporarily mask this interrupt by writing to the GINAKEFFM bit in the USBFS_GINTMSK register.
 - In the USBFS_GINTMSK register, GINAKEFFM=0
4. When the application is ready to exit global OUT NAK mode, the SGONAK bit in USBFS_DCTL must be cleared. This action also clears the GONAKEFF interrupt (USBFS_GINTSTS).
 - In CGONAK, USBFS_DCTL=1

5. If the application has previously masked this interrupt, it must unmask the interrupt as follows:
 - In GINTMSK, GINAKEFFM=1

Set Global OUT NAK to 1

The application must disable enabled OUT endpoints using the following sequence.

Application programming sequence:

1. Before disabling any OUT endpoints, the application must enable global OUT NAK mode in the module.
 - In USBFS_DCTL, SGONAK=1
2. Wait for GONAKEFF interrupt (USBFS_GINTSTS)
3. Disable the OUT endpoint by programming the following fields:
 - In USBFS_DOEPCTLx, EPDIS=1
 - In USBFS_DOEPCTLx, SNAK=1
4. Wait for the EPDISD interrupt (USBFS_DOEPINTx), which indicates that the OUT endpoint has been completely disabled. When an EPDISD interrupt is raised, the module also clears the following bits:
 - In USBFS_DOEPCTLx, EPDIS=0
 - In USBFS_DOEPCTLx, EPENA=0
5. The application must clear the global OUT NAK bit to start receiving data from other non-disabled OUT endpoints.
 - In USBFS_DCTL, SGONAK=0

General Asynchronous OUT Data Transfer

This section describes a general asynchronous OUT data transfer (control, bulk or interrupt).

Application Requirements:

1. Before establishing an OUT transfer, the application must allocate a buffer in memory to hold all the data to be received as part of the OUT transfer.
2. For OUT transfers, the transfer size field in the endpoint's transfer size register must be a multiple (and word-aligned) of the endpoint's maximum packet size.
 - Transfer Size [EPNUM] = $n \times (\text{MPSIZ}[\text{EPNUM}] + 4 - (\text{MPSIZ}[\text{EPNUM}] \bmod 4))$
 - Packet Count [EPNUM] = n
 - $n > 0$
3. When an OUT endpoint interrupt occurs, the application must read the endpoint's transfer size register to calculate the amount of valid data in memory. The amount of valid data received may be less than the programmed transfer size.
 - Effective data amount in memory = initial transfer amount set by application program - remaining transfer amount after module update

- The number of received USB data packets = the initial data packets set by the application
- the remaining data packets after the module is updated

Internal data flow:

1. The application must set the transfer size and packet count fields in the endpoint-related registers, clear the NAK bit, and enable the endpoint to receive data.
2. Once the NAK bit is cleared, the module starts receiving data and writing the data to the receive FIFO (as long as there is room in the receive FIFO). For every packet received on the USB, the packet and its status are written to the receive FIFO. Every packet written to the receive FIFO (packets up to the maximum packet size or short packets) decrements the packet count field for this endpoint by 1.
 - If the CRC of the received data packet is invalid, it will be automatically cleared from the receive FIFO.
 - After replying with an ACK for the packet on the USB, the module will discard the non-synchronized OUT packet resent by the host because the ACK cannot be detected. The application will not detect multiple consecutive OUT packets on the same endpoint with the same data PID. In this case, the packet count is not decremented.
 - If there is no room in the receive FIFO, isochronous or asynchronous packets are ignored and not written to the receive FIFO. Additionally, non-synchronized OUT tokens will receive a NAK handshake response.
 - In all three cases above, the packet count is not decremented because no data is written to the receive FIFO.
3. When the packet count goes to 0 or a short packet is received on an endpoint, the NAK bit for that endpoint will be set. After the NAK bit is set, synchronous or asynchronous packets will be ignored and will not be written to the receive FIFO, while the non-synchronous OUT token will receive a NAK handshake response.
4. After the data is written into the receive FIFO, the application will read the data from the receive FIFO and write the data to the external memory, one packet at a time, endpoint by endpoint.
5. After each data packet is written to the external memory on AHB, the transfer size of the endpoint is automatically subtracted from the size of the data packet.
6. The OUT data transfer completion status of the OUT endpoint will be written to the receive FIFO when:
 - transfer size is 0 and packet count is 0
 - The last OUT packet written to the receive FIFO is a short packet (packet size: 0~max. packet size-1)
7. When the application program pops up this status entry (OUT data transfer is completed), and generates the transfer completion interrupt of this endpoint, and clears the endpoint enable bit at the same time.

Application programming sequence:

1. Program the USBFS_DOEPSIZx registers with the transfer size and corresponding number of packets.
2. Program the USBFS_DOEPCTLx registers with the endpoint characteristics and set the EPENA and CNAK bits.
 - In USBFS_DOEPCTLx, EPENA=1
 - In USBFS_DOEPCTLx, CNAK=1
3. Wait for the RXFNE interrupt (in USBFS_GINTSTS) and read the packet from the receive FIFO.
 - This step can be repeated multiple times, depending on the transfer size.
4. An XFRC interrupt (USBFS_DOEPINTx) is triggered to indicate the successful completion of the asynchronous OUT data transfer.
5. Read the USBFS_DOEPSIZx registers to determine the amount of valid data.

General Synchronous OUT Data Transfer

This section describes regular synchronous OUT data transfers.

Application Requirements:

1. All application requirements for asynchronous OUT data transfers apply to synchronous OUT data transfers.
2. For the transfer size and packet count fields in isochronous OUT data transfers, they must always be set to the number of packets of the largest packet size that can be received in a single frame. OUT data transfer transactions of synchronous type must be completed within one frame.
3. The application must read all isochronous OUT packets (data entries and status entries) from the receive FIFO before the end of the periodic frame (EOPF interrupt in USBFS_GINTSTS).
4. To receive data in the next frame, a synchronous OUT endpoint must be enabled after EOPF(USBFS_GINTSTS) and before SOF(USBFS_GINTSTS).

Internal data flow:

1. The internal data flow of a synchronous OUT endpoint is basically the same as that of a non-synchronous OUT endpoint, but there are some differences.
2. When a synchronous OUT endpoint is enabled by setting the endpoint enable bit and clearing the NAK bit, the even/odd-numbered frame bit must be set accordingly. The module will receive data in a particular frame on the synchronous OUT endpoint only if the following conditions are met:
 - EONUM (in USBFS_DOEPCTLx) = FNSOF [0] (in USBFS_DSTS)
3. When the application completely reads a synchronous OUT packet (data and status) from

the receive FIFO, the module will update the RXDPID field in USBFS_DOEPTSIZE according to the data PID of the last synchronous OUT packet read from the receive FIFO.

Application programming sequence:

1. Program the USBFS_DOEPTSIZE registers with the transfer size and corresponding packet count
2. Program the USBFS_DOEPCTL registers with endpoint characteristics and set the endpoint enable bit, clear NAK bit, and odd/even frame bit.
 - EPENA1
 - CNAK=1
 - EONUM= (0: Even number / 1: Odd number)
3. Wait for the RXFNE interrupt (in USBFS_GINTSTS) and read the packet from the receive FIFO.
 - This step can be repeated multiple times, depending on the transfer size.
4. XFRC interrupt (in USBFS_DOEPINTx) indicates completion of synchronous OUT data transfer. This interrupt does not necessarily mean that the data in memory is valid.
5. For synchronous OUT transfers, the application may not always detect this interrupt. Instead, the application may detect the IISOXFRM interrupt in USBFS_GINTSTS.
6. Read the USBFS_DOEPTSIZE registers to determine the size of the transfer received and to determine the validity of the data received in the frame. The application must consider data received in memory as valid data only if one of the following conditions is true:
 - RXDPID=D0 (in USBFS_DOEPTSIZE) and the number of USB packets receiving this valid data=1
 - RXDPID=D1 (in USBFS_DOEPTSIZE) and the number of USB packets receiving this valid data=2
 - RXDPID=D2 (in USBFS_DOEPTSIZE) and the number of USB packets receiving this valid data=3

The number of USB data packets receiving the valid data = the initial data packet number of application programming - the remaining data packet number after the module is updated.

Applications can drop invalid packets.

Incomplete Synchronous OUT Data Transfer

This section describes the application programming sequence when a sync OUT packet is lost.

Internal data flow:

1. For synchronous OUT endpoints, XFRC interrupts (in USBFS_DOEPINTx) may not always be raised. If the module drops isochronous OUT packets, the application may not detect XFRC interrupts (USBFS_DOEPINTx) in the following cases:

- When the receive FIFO cannot hold the complete ISO OUT packet, the module will discard the received ISO OU data
 - Received sync OUT packet has CRC error
 - The sync OUT token received by the module is corrupted
 - Application reads data from receive FIFO very slowly
2. If the module detects the periodic end of frame before the transmission of all synchronous OU endpoints is completed, it will trigger the incomplete synchronous OUT data interrupt (IISOOXFRM in USBFS_GINTSTS), indicating that at least one synchronous OUT endpoint has not triggered the XFRC interrupt (in USBFS_DOEPINTx). At this time, the endpoint that has not completed the transfer is still enabled, but on this endpoint of the USB, there is no valid transfer in progress.

Application programming sequence:

1. The hardware triggers the IISOOXFRM interrupt (USBFS_GINTSTS) to indicate that at least one isochronous OUT endpoint has an outstanding transfer in the current frame.
2. If this occurs because the isochronous OUT data was not fully read from the endpoint, the application must ensure that all isochronous OUT data (including data entries and status entries) is read from the receive FIFO before continuing processing.
 - Once all data has been read from the receive FIFO, the application can detect an XFRC interrupt (USBFS_DOEPINTx). In this case, the application must re-enable the endpoint to receive isochronous OUT data in the next frame.
3. When the application receives an IISOOXFRM interrupt (in USBFS_GINTSTS), the application must read the control registers (USBFS_DOEPCTLx) of all synchronous OUT endpoints to determine which endpoints have incomplete transfers in the current frame. An endpoint transfer is incomplete when both of the following conditions are met:
 - EONUM bit (in USBFS_DOEPCTLx) = FNSOF [0] (in USBFS_DSTS)
 - EPENA=1 (in USBFS_DOEPCTLx)
4. Before the SOF interrupt is detected (in USBFS_GINTSTS), the previous step must be performed to ensure that the current frame number has not changed.
5. For isochronous OUT endpoints with incomplete transfers, the application must discard data in memory and disable the endpoint by setting the EPDIS bit in USBFS_DOEPCTLx.
6. Wait for an EPDIS interrupt (in USBFS_DOEPINTx), and enable the endpoint to receive new data in the next frame.
 - Since the module may take some time to inhibit the endpoint, the application may not be able to receive data in the next frame after receiving invalid sync data.

Stop an Asynchronous OUT Endpoint

This section describes how an application can stop an asynchronous endpoint.

1. Puts the module in global OUT NAK mode.

2. Disable required endpoints
 - When disabling an endpoint, set STALL=1 (in USBFS_DOEPCTL) instead of setting the SNAK bit in USBFS_DOEPCTL. The STALL bit always takes precedence over the NAK bit.
3. The STALL bit (in USBFS_DOEPCTLx) must be cleared when the application no longer needs the endpoint to reply to the STALL handshake signal.
4. If the application sets or clears the STALL state of the endpoint due to receiving the SetFeature.Endpoint Halt or ClearFeature.Endpoint Halt command from the host, it must set or clear the STALL bit before the state phase transfer on the control endpoint.

33.6.7.2 IN Data Transmission

Packet Write

This section describes how the application writes packets to the endpoint FIFO when the dedicated transmit FIFO has been enabled.

1. The application can choose polling mode or interrupt mode.
 - In polling mode, the application monitors the status of the endpoint transmit data FIFO by reading the USBFS_DTXFSTSx registers to determine if there is enough space in the data FIFO.
 - In interrupt mode, the application waits for the TXFE interrupt (in USBFS_DIEPINTx), then reads the USBFS_DTXFSTSx registers to determine if there is enough space in the data FIFO.
 - To write a single non-zero length packet, there must be enough room in the data FIFO to hold the entire packet.
 - To write a zero-length packet, the application must not look into the FIFO space.
2. If one of the above methods is used, when the application determines that there is enough space to write the transmit data packet, the application must first write the appropriate write operation to the endpoint control register, and then write the data to the data FIFO. Typically, the application must perform a read-modify-write operation on the USBFS_DIEPCTLx register to avoid modifying other contents in the register while setting the endpoint enable bit to 1.

The application can write multiple packets for the same endpoint to the transmit FIFO if there is enough space. For periodic IN endpoints, applications can only write multiple packets within a frame at a time. The application writes all the packets to be sent in the next frame only after the transmission of the previous frame's communication transaction is complete.

Set IN Endpoint NAK to 1

Internal data flow:

1. When the application program sets the IN NAK of a specific endpoint, the module will stop

the data transmission on the endpoint regardless of whether the data in the endpoint transmit FIFO is available or not.

2. The asynchronous endpoint receives the IN token and replies with a NAK handshake response.
 - The synchronous endpoint receives the IN token and returns a zero-length packet
3. The module triggers the INEPNE (IN endpoint NAK valid) interrupt in USBFS_DIEPINTx in response to the SNAK bit in USBFS_DIEPCTLx.
4. Once the application detects this interruption, it considers the endpoint to be in IN NAK mode. The application can clear this interrupt by setting the CNAK bit in USBFS_DIEPCTLx.

Application programming sequence:

1. To stop sending any data on a particular IN endpoint, the application must set the IN NAK bit. To set this bit, the following fields must be programmed.
 - SNAK=1 in USBFS_DIEPCTLx
2. Wait for INEPNE interrupt trigger in USBFS_DIEPINTx. This interrupt indicates that the module has stopped sending data on the endpoint.
3. The module can send valid IN data on the endpoint when the application program sets the NAK bit to 1 but the "NAK Valid" interrupt has not yet triggered.
4. The application can temporarily mask this interrupt by writing to the INEPNEM bit in DIEPMSK.
 - In DIEPMSK, INEPNEM = 0
5. To exit endpoint NAK mode, the application must clear the NAK status bit (NAKSTS) in USBFS_DIEPCTLx. This action also clears INEPNE interrupts (in USBFS_DIEPINTx).
 - In USBFS_DIEPCTLx, CNAK=1
6. If the application has masked this interrupt, it must be unmasked as follows:
 - In DIEPMSK, INEPNEM=1

Disable IN Endpoints

Use the following sequence to disable specific IN endpoints that were previously enabled.

Application programming sequence:

1. The application must stop writing data on the AHB before disabling the IN endpoint.
2. The application must set the endpoint to NAK mode.
 - SNAK=1 in USBFS_DIEPCTLx
3. Wait for INEPNE interrupt in USBFS_DIEPINTx.
4. Set the following bits in the USBFS_DIEPCTLx registers for endpoints that must be disabled.
 - EPDIS=1 in USBFS_DIEPCTLx
 - SNAK=1 in USBFS_DIEPCTLx
5. The triggering of the EPDISD interrupt in USBFS_DIEPINTx indicates that the module has completely disabled the specified endpoint. At the same time the interrupt is triggered,

the module also clears the following bits:

- In USBFS_DIEPCTLx, EPENA=0
 - In USBFS_DIEPCTLx, EPDIS=0
6. The application must read the USBFS_DIEPTSIZx registers for periodic IN EPs to calculate how much data is being sent on the USB on the endpoint.
 7. The application must clear the data in the endpoint transmit FIFO by setting the following fields in the USBFS_GRSTCTL register:
 - TXFNUM (in USBFS_GRSTCTL) = endpoint transmit FIFO number
 - TXFFLSH (in USBFS_GRSTCTL) = 1

The application must poll the USBFS_GRSTCTL register until the module clears the TXFFLSH bit, which indicates the end of the FIFO flush operation. To send new data on this endpoint, the application can re-enable the endpoint at a later time.

General Acyclic IN Data Transfer

Application Requirements:

1. Before establishing an IN transfer, the application must ensure that each packet that makes up an IN transfer fits in a single buffer.
2. For IN transmission, the transfer size field in the endpoint transfer size register indicates the effective data volume for this transfer, which consists of multiple maximum packet sizes and a single short packet. This short packet is sent at the end of the transfer.
 - To send multiple packets of maximum packet size plus a short packet at the end of the transfer:

Transfer Size [EPNUM] = $x \times \text{MPSIZ}[\text{EPNUM}] + \text{sp}$

If ($\text{sp} > 0$), Packet Count [EPNUM] = $x + 1$.

Otherwise, Packet Count [EPNUM] = x

- To send a single zero-length packet:

transfer_size [EPNUM] = 0

Packet Count [EPNUM] = 1

- To send multiple packets of the maximum packet size plus a zero-length packet at the end of the transfer, the application must split the transfer into two parts.

The first part sends packets of the maximum packet size, the second part sends only zero-length packets.

First transfer: transfer size [EPNUM] = $x \times \text{MPSIZ}[\text{epnum}]$; packet count = n ;

Second transfer: transfer size [EPNUM] = 0; packet count = 1;

3. After enabling an endpoint for data transfer, the module updates the transfer size register. At the end of an IN transfer, the application must read the transfer size register to determine how much of the data that was put into the transmit FIFO has been sent out over the USB.

4. Amount of data put into transmit FIFO = initial transfer size programmed by application - final transfer size after module update
 - Amount of data sent over USB = (initial packet count from application programming - final packet count after module update) × MPSIZ [EPNUM]
 - Amount of data remaining to be sent over USB = (Application programmed initial transfer size - Amount of data already sent over USB)

Internal data flow:

1. The application must set the transfer size and packet count fields in the registers for a specific endpoint and enable that endpoint to send data.
2. The application must also write the necessary data to the endpoint's transmit FIFO.
3. Every time the application writes a packet to the send FIFO, the transfer size of the endpoint is automatically subtracted from the packet size. The application continues fetching data from memory to write to the transmit FIFO until the transfer size for that endpoint becomes 0. After writing data to the FIFO, the "Number of packets in FIFO" count will increase (this is a 3-bit count, which is maintained internally by the module, and each IN endpoint sends a FIFO corresponding to one. In the IN endpoint FIFO, the maximum number of packets maintained by the module is always eight). Each FIFO has a separate flag for zero-length packets, and there is no data in the FIFO.
4. After the data is written into the send FIFO, the module will send the data when it receives the IN token. After each data packet is sent out and the reply ACK handshake signal is received, the data packet count of the endpoint will be decremented by 1 until the data packet count becomes 0. The packet count is not decremented when a timeout occurs.
5. For a zero-length packet (indicated by the internal zero-length flag), the module issues a zero-length packet for the IN token and decrements the value of the packet count field.
6. If there is no data in the FIFO corresponding to the endpoint that received the IN token, and the packet count field for that endpoint is zero, the module generates an "IN Token Received While TxFIFO Empty" (ITTXFE) interrupt for that endpoint (provided that the NAK bit for that endpoint is not set). The module replies with a NAK handshake signal on the asynchronous endpoint.
7. The module resets the FIFO pointers to the beginning internally and does not generate a timeout interrupt.
8. When the transfer size is 0 and the packet count is 0, a transfer complete (XFRC) interrupt for this endpoint is generated and the endpoint enable is cleared.

Application programming sequence:

1. Program the USBFS_DIEPTSIZx registers with the transfer size and corresponding packet count.
2. Program the USBFS_DIEPCTLx registers with the endpoint characteristics and set the CNAK

and EPENA (Endpoint Enable) bits to 1.

3. When sending a non-zero length packet, the application must poll the USBFS_DTXFSTSx registers (where x is the FIFO number associated with this endpoint) to determine if there is enough space in the data FIFO. The application can also select the TXFE bit (in USBFS_DIEPINTx) before writing data.

General Periodic IN Data Transmission

This section describes typical periodic IN data transfers.

Application Requirements:

1. Application requirements 1, 2, 3, 4 for general acyclic IN data transmission are also applicable for cyclic IN data transmission (with a slight modification to requirement 2).
 - The application can only send a number of packets of the maximum packet size or a number of packets of the maximum packet size, plus a short packet at the end of the transfer.

To send multiple packets of the maximum packet size plus a short packet at the end of the transfer, the following conditions must be met:

Transfer Size [EPNUM] = $x \times \text{MPSIZ} [\text{EPNUM}] + \text{sp}$

(where x is an integer greater than 0, and the range of sp is 0 ~ MPSIZ [EPNUM]-1)

if (sp > 0), Packet Count [EPNUM] = x + 1

else, PacketCount [EPNUM] = x;

MCNT [EPNUM] = Packet Count [EPNUM]

- Applications cannot send zero-length packets at the end of a transfer. An application can send a zero-length packet by itself.
 - To send a single zero-length packet:

Transfer_size [EPNUM]=0

Packet Count [EPNUM]=1

MCNT [EPNUM]=packet count [EPNUM]
2. Applications can only schedule data transfers for one frame at a time.
 - $(\text{MCNT} - 1) \times \text{MPSIZ} \leq \text{XFERSIZ} \leq \text{MCNT} \times \text{MPSIZ}$
 - PKTCNT = MCNT (in USBFS_DIEPTSIZx)
 - If XFERSIZ < MCNT × MPSIZ, the last packet transmitted is a short packet
 - Please note: MCNT is located in USBFS_DIEPTSIZx, MPSIZ is located in USBFS_DIEPCTLx, PKTCNT is located in USBFS_DIEPTSIZx, XFERSIZ is located in USBFS_DIEPTSIZx
 3. Before receiving the IN token, the application must write the complete data to be sent in the frame into the transmit FIFO. When the IN token is received, even if the data to be sent in the frame in the sending FIFO is only one double word short of writing, the module will perform the operation when the FIFO is empty. When the transmit FIFO is empty:

- A zero-length packet will be replied on the isochronous endpoint
- NAK handshake signal will be replied on interrupt endpoint

Internal data flow:

1. The application must set the transfer size and packet count fields in the registers for a specific endpoint and enable that endpoint to send data.
2. The application must also write the necessary data to the transmit FIFO associated with this endpoint.
3. Every time the application writes a packet to the send FIFO, the transfer size of the endpoint is automatically subtracted from the packet size. The application continues fetching data from memory to write to the transmit FIFO until the transfer size for that endpoint becomes 0.
4. When the periodic endpoint receives the IN token, the module will start sending the data in the FIFO (if there is data in the FIFO). If there is no complete packet of data to be sent for the frame in the FIFO, the module will generate an "IN token received while TxFIFO is empty" interrupt for the endpoint.
 - A zero-length packet will be replied on the isochronous endpoint
 - A NAK handshake signal will be replied on the interrupt endpoint
5. The packet count for an endpoint is decremented by 1 when:
 - For isochronous endpoints, when sending a zero-length or non-zero-length packet
 - For interrupt endpoints, decrements when sending ACK handshake
 - When the transfer size and packet count are both 0, a transfer complete interrupt for this endpoint is generated and the endpoint enable bit is cleared.
6. During the "periodic frame interval" (controlled by the PFIVL bit in USBFS_DCFG), when the module finds that any data in the synchronous IN endpoint FIFO that should be empty in the current frame has not been sent, it will generate an IISOIXFR interrupt in USBFS_GINTSTS.

Application programming sequence:

1. Program the USBFS_DIEPCTLx registers with the endpoint characteristics and set the CNAK and EPENA bits.
2. Write the data to be sent in the next frame into the transmit FIFO.
3. The hardware triggers the ITTXFE interrupt (in USBFS_DIEPINTx) to indicate that the application has not yet written all the data that needs to be sent into the transmit FIFO.
4. If the interrupt endpoint is enabled before the interrupt is detected, the interrupt will be ignored. If the interrupt endpoint is not enabled, enable it so that data can be sent on the next IN token received.
5. If the hardware triggers the XFRC interrupt (in USBFS_DIEPINTx), if no ITTXFE interrupt is generated in USBFS_DIEPINTx, it means that the synchronous IN transfer is successfully

completed. Reading the USBFS_DIEPTSIZx registers always gives transfer size = 0 and packet count = 0, which means all data has been sent over USB.

6. Regardless of whether an ITTXFE interrupt (in USBFS_DIEPINTx) is generated when the XFRC interrupt (in USBFS_DIEPINTx) is set, it indicates the successful completion of the interrupt IN transfer. Reading the USBFS_DIEPTSIZx registers always gets transfer size = 0 and packet count = 0, which means all data has been sent over USB.
7. If none of the aforementioned interrupts are generated when the incomplete synchronous IN transfer (IISOIXFR) interrupt is set in USBFS_GINTSTS, it means that the module has not received at least 1 periodic IN token in the current frame.

Synchronous IN Data Transfer not Completed

This section describes what an application must do for outstanding isochronous IN data transfers.

Internal data flow:

1. A synchronous IN transfer is considered incomplete when one of the following conditions is met:
 - The module received a corrupt sync IN token on at least one sync IN endpoint. At this point, the application detects an incomplete isochronous IN transfer interrupt (IISOIXFR bit in USBFS_GINTSTS).
 - The application is writing data to the transmit FIFO too slowly, and the IN token is received before the complete data has been written to the FIFO. At this point, the application detects an "IN token received while Tx FIFO is empty" interrupt in USBFS_DIEPINTx. The application can ignore this interrupt as eventually this will generate an incomplete isochronous IN transfer interrupt (IISOIXFR bit in USBFS_GINTSTS) at the end of a periodic frame. The module responds to the received IN token by sending a zero-length packet over USB.
2. The application must stop writing data to the transmit FIFO as soon as possible.
3. The application must set the NAK bit and disable bit of the endpoint.
4. The module disables the endpoint, clears the disable bit and triggers an "Endpoint Disabled" interrupt for the endpoint.

Application programming sequence:

1. Applications can ignore the "IN token received while Tx FIFO is empty" interrupt in USBFS_DIEPINTx on any isochronous IN endpoint, as eventually this will generate an incomplete isochronous IN transfer interrupt (in USBFS_GINTSTS).
2. A hardware triggered outstanding isochronous IN transfer interrupt (in USBFS_GINTSTS) indicates that there is an outstanding isochronous IN transfer on at least one isochronous IN endpoint.
3. The application must read the "Endpoint Control" register of all synchronous IN endpoints to detect endpoints with outstanding IN data transfers.

4. The application must stop writing to the "Periodic Transmit FIFO" associated with these endpoints.
5. Program the following fields in the USBFS_DIEPCTLx registers to disable the endpoint:
 - SNAK=1 in USBFS_DIEPCTLx
 - EPDIS=1 in USBFS_DIEPCTLx
6. The hardware triggers the "Endpoint Disabled" interrupt in USBFS_DIEPINTx to indicate that the module has disabled the endpoint.
 - At this point, the application must empty the data in the associated transmit FIFO, or overwrite the existing data in the FIFO by enabling the endpoint for a new transmission in the next frame. To refresh data, the application must use the USBFS_GRSTCTL register.

Stop the Asynchronous IN Endpoint

This section describes how an application can stop an asynchronous endpoint.

Application programming sequence:

1. Inhibit IN endpoints to stop. Also set the STALL bit to 1.
2. EPDIS=1 in USBFS_DIEPCTLx (when the endpoint is enabled)
 - STALL=1 in USBFS_DIEPCTLx
 - The STALL bit always has priority over the NAK bit
3. The hardware triggers an "endpoint disabled" interrupt (in USBFS_DIEPINTx) to let the application know that the module has disabled the specified endpoint.
4. The application must empty the aperiodic or periodic transmit FIFO depending on the endpoint type. For non-periodic endpoints, the application must re-enable another non-stop non-periodic endpoint to send data.
5. When the application is ready to end the STALL handshake for this endpoint, it must clear the STALL bit of USBFS_DIEPCTLx to 0.
6. If the application sets or clears the STALL state of an endpoint as a result of receiving a SetFeature.Endpoint Halt command or a ClearFeature.Endpoint Halt command from the host, it must set or clear the STALL bit before the state phase transfer of that control endpoint.

Special case: stop controlling the OUT endpoint

If, during the data phase of a control transfer, the host sends more IN/OUT tokens than the value specified in the SETUP packet, the module must reply STALL to these excess IN/OUT tokens. In this case, the application must enable the ITTXFE interrupt for USBFS_DIEPINTx and the OTEPDIS interrupt for USBFS_DOEPINTx during the data phase of the control transfer (after the module has completed transferring the amount of data specified by the SETUP packet). Subsequently, when the application receives this interrupt, it must set the STALL bit in the corresponding endpoint control register and clear the interrupt.

33.7 Register Description

The application program controls the USBFS module by reading and writing the control and status registers through the AHB slave interface. All registers of the USBFS module are 32-bit registers, and their addresses are aligned by 32 bits, so they can only be accessed in 32-bit mode.

The control and status registers are divided into the following categories:

- USBFS system control register
- Module global registers
- Host mode register
- Device mode register
- Power and clock gating control registers
- Data FIFO (DFIFO) access register

The base address of the USBFS system control register is different from other registers. This register is independent of the USBFS module to control the related settings of the USBFS module.

Only module global registers, power and clock gating control registers, and data FIFO access registers can be accessed in both host and device modes. When the USBFS module is in one mode (host or device), the application cannot access registers in another role mode, such as host mode, accessing device mode registers. If an illegal access occurs, a mode mismatch interrupt will be generated and reflected in the module interrupt register USBFS.GINTSTS.NMIS. When the module switches from one role mode to another, the registers in the new operating mode must be reprogrammed to the state after power-on reset.

Control Status Register Memory Map

The host and device mode registers occupy different addresses. All registers are implemented in the AHB clock domain.

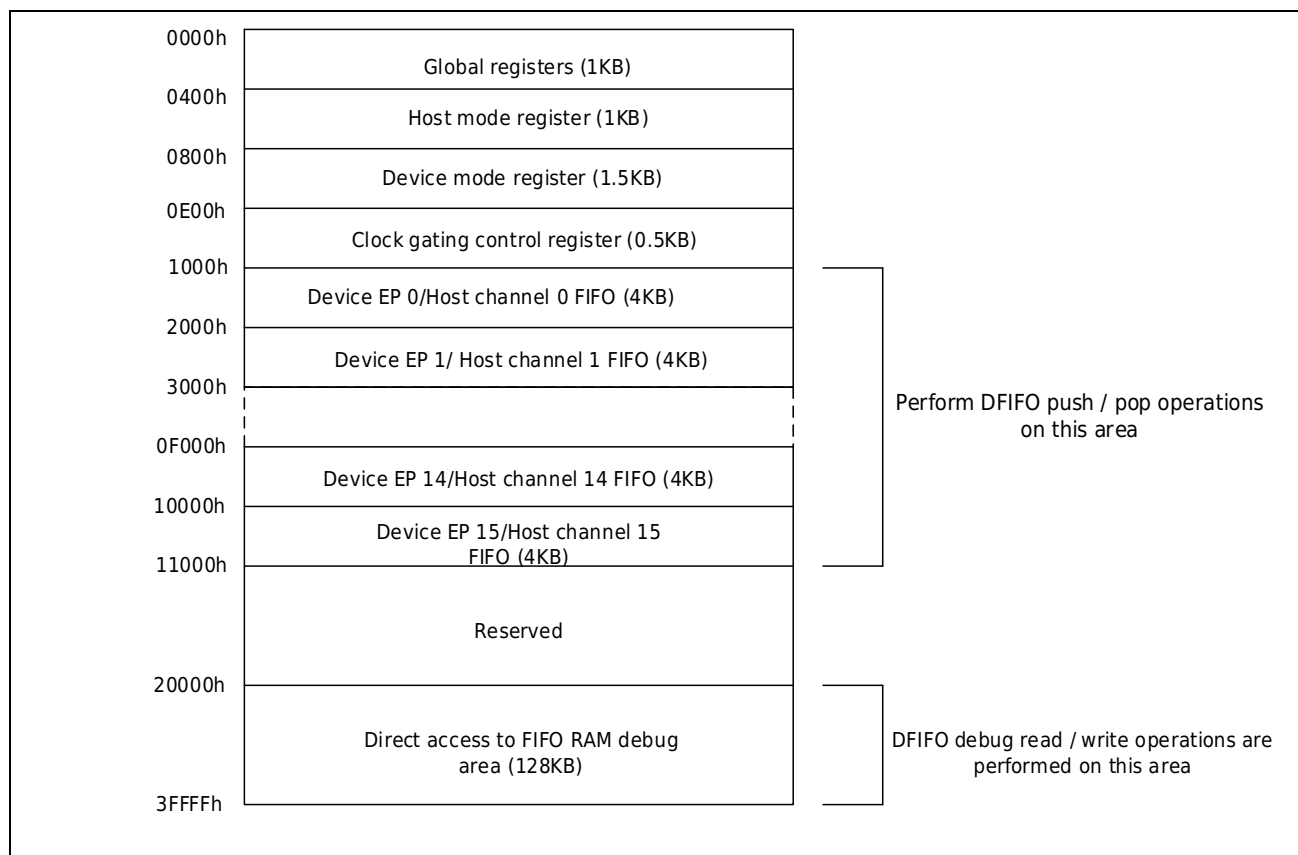


Figure 33-7 USBFS Control Status Register Memory Map

For the USBFS module register list and base address, please refer to Table 33-3~Table USBFS Register List.

Table 33-3 USBFS System Control Register List

USB system control register base address: 0x40055400

(USBFS system control register) register name	Offset Address	Reset Value
USB System Control Register (USB_SYCTLREG)	0x00	0x0000 0000

Table 33-4 USBFS Clock Gating Control Register List

USBFS module register base address: 0x40080000

(USBFSGlobal Register) Register Name	Offset Address	Reset Value
USBFS VBUS Control Register (USBFS_GVBUSCFG)	0x00	0x00XX 0000
USBFS AHB Control Register (USBFS_GAHBCFG)	0x08	0x0000 0000
USBFS USB configuration register (USBFS_GUSBCFG)	0x0c	0x0000 1440
USBFS Reset Register (USBFS_GRSTCTL)	0x10	0x8000 0000
USBFS Global Interrupt Status Register (USBFS_GINTSTS)	0x14	0x1400 0020
USBFS Global Interrupt Mask Register (USBFS_GINTMSK)	0x18	0x0000 0000
USBFS Status Debug Read Register (USBFS_GRXSTSR)	0x1c	0x0000 0000
USBFS status read and pop register (USBFS_GRXSTSP)	0x20	0x0000 0000
USBFS FIFO size register (USBFS_GRXFSIZ)	0x24	0x0000 0280
USBFS host non-periodic transmit FIFO size register (USBFS_HNPTXFSIZ)/ Device endpoint 0 transmit FIFO size register (USBFS_DIEPRXF0)	0x28	0x0280 0280
USBFS aperiodic transmit FIFO/queue status register (USBFS_HNPTXSTS)	0x2c	0x0008 0280
USBFS module ID register (USBFS_CID)	0x3c	0x1234 5678
USBFS LPM configuration register (USBFS_GLPMCFG)	0x54	0x0000 0000
USBFS Periodic Transmit FIFO Size Register (USBFS_HPTXFSIZ)	0x100	0x0280 0500
USBFS device IN endpoint x transmit FIFO size register (USBFS_DIEPTXFx)	0x100+x*4(x=1~5)	0x0280 0500+(x-1)*0x280

(USBFS host control and status register) register name	Offset Address	Reset Value
USBFS host configuration register (USBFS_HCFG)	0x400	0x0000 0200
USBFS host frame interval register (USBFS_HFIR)	0x404	0x0000 EA60
USBFS host frame number/frame remaining interval register (USBFS_HFNUM)	0x408	0x0000 3FFF
USBFS host periodic transmit FIFO/queue status register (USBFS_HPTXSTS)	0x410	0x0008 0280
USBFS host overall channel interrupt register (USBFS_HAINT)	0x414	0x0000 0000
USBFS host all channel interrupt mask register (USBFS_HAINTMSK)	0x418	0x0000 0000
USBFS host port control and status register (USBFS_HPRT)	0x440	0x0000 0000
USBFS host channel x characteristics register (USBFS_HCCHARx)	0x500+x*0x20(x=0~5)	0x0000 0000
USBFS host channel x interrupt register (USBFS_HCINTx)	0x508+x*0x20(x=0~5)	0x0000 0000
USBFS host channel x interrupt mask register (USBFS_HCINTx)	0x50c+x*0x20(x=0~5)	0x0000 0000
USBFS host channel x transfer size register (USBFS_HCTSIZx)	0x510+x*0x20(x=0~5)	0x0000 0000
USBFS host channel x DMA address register (USBFS_HCDMAx)	0x514+x*0x20(x=0~5)	0XXXXX XXXX

(USBFS Device Control and Status Register) Register Name	Offset Address	Reset Value
USBFS device configuration register (USBFS_DCFG)	0x800	0x0820 0000
USBFS device control register (USBFS_DCTL)	0x804	0x0000 0002
USBFS device status register (USBFS_DSTS)	0x808	0x0000 0002
USBFS device IN endpoint general interrupt mask register (USBFS_DIEPMASK)	0x810	0x0000 0000
USBFS device OUT endpoint general interrupt mask register (USBFS_DOEPMASK)	0x814	0x0000 0000
USBFS device overall endpoint interrupt register (USBFS_DAINTR)	0x818	0x0000 0000
USBFS device overall endpoint interrupt mask register (USBFS_DAINTRMASK)	0x81c	0x0000 0000
USBFS device IN endpoint FIFO empty interrupt mask register (USBFS_DIEPEMPMASK)	0x834	0x0000 0000
USBFS device in endpoint 0 control register (USBFS_DIEPCTL0)	0x900	0x0000 8000
USBFS device IN endpoint x control register (USBFS_DIEPCTLx)	0x900+x*0x20(x=1~5)	0x0000 0000
USBFS device in endpoint x interrupt register (USBFS_DIEPINTx)	0x908+x*0x20(x=0~5)	0x0000 0080
USBFS device IN endpoint x transfer size register (USBFS_DIEPSIZx)	0x910+x*0x20(x=0~5)	0x0000 0000
USBFS device IN endpoint x DMA address register (USBFS_DIEPDMAx)	0x914+x*0x20(x=0~5)	0x0000 0000
USBFS device IN Endpoint x transmit FIFO status Register (USBFS_DTXFSTSx)	0x918+x*0x20(x=0~5)	0x0000 0280
USBFS device OUT endpoint 0 control register (USBFS_DOEPCTL0)	0xb00	0x0000 8000
USBFS device OUT endpoint x control register (USBFS_DOEPCTLx)	0xb00+x*0x20(x=1~5)	0x0000 0000
USBFS device OUT endpoint x interrupt register (USBFS_DOEPINTx)	0xb08+x*0x20(x=0~5)	0x0000 0000
USBFS device OUT endpoint x transfer size register (USBFS_DOEPSIZx)	0xb10+x*0x20(x=0~5)	0x0000 0000
USBFS device OUT endpoint x DMA address register (USBFS_DOEPDMAx)	0xb14+x*0x20(x=0~5)	0xFFFFFFFF

(USBFS power and clock gating control register) Register Name	Offset Address	Reset Value
USBFS clock gating control register (USBFS_GCCTL)	0xe00	0x0000 0000

33.7.1 USBFS System Control Register

33.7.1.1 USB System Control Register (USB_SYCTLREG)

USB System Control Register

Offset address: 0x00

Reset value: 0x0000 0000

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

Bit	Flag	Bit Name	Function	Read and Write
b31~b19	Reserved	-	The reset value must be maintained.	R/W
b18	USBFS_NFE	USBFS Register Filter Enable	USBFS Filter Enable Register This register is used to control the switch of the USBFS on-chip full-speed PHY DP/DM analog filter in STOP mode. 0: Analog filter off 1: Analog filter on, please refer to the setting of USBFS_NFS [1:0] for the filtering range	R/W
b17~b16	USBFS_NFS	USBFS Register Filter Select	USBFS Filter Select Register This register is used to control the filtering range of the USBFS on-chip full-speed PHY DP/DM analog filter in STOP mode. 00b: Filtering width level 1 01b: Filtering width level 2 10b: Filtering width level 3 11b: Filtering width level 4 For specific values of each gear, please refer to the "49. Electrical Characteristics 49.3.34 Filter Characteristics in USB On-Chip Full-Speed PHY STOP Mode" section.	R/W
b15~b2	Reserved	-	The reset value must be maintained.	R/W
b1	USBFS_SOFEN	USBFS SOF pulse output enable bit	When the USBFS host sends SOF or the device successfully receives SOF, a SOF pulse with a width of 16 system clock cycles is output from the PAD to enable 0: SOF pulse is not output 1: SOF pulse output Note: Accessible in both device mode and host mode.	R/W
b0	USBFS_DFB	USBFS VBUS/ID pin internal debounce filter bypass enable bit	USBFS VBUS/ID pin module internal debounce filter bypass enable bit 0: The internal debounce filter of the module is valid 1: Bypass module internal debounce filter Note: Accessible in both device mode and host mode.	R/W

33.7.2 USBFS global registers

These registers are available in both host and device modes and do not need to be reprogrammed when switching between the two modes. Bit values in register descriptions are in binary unless otherwise noted.

33.7.2.1 USBFS VBUS Control Register (USBFS_GVBUSCFG)

VBUS Configuration Register

Offset address: 0x00

Reset value: 0x00XX 0000

This register can be used to set the VBUS value regardless of the state of the VBUS pin.

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

Bit	Flag	Bit Name	Function	Read and Write
b31~b8	Reserved	-	The reset value must be maintained.	R/W
b7	VBUSVAL	VBUS value	VBUS Value It is used to set the VBUS value of USBFS. When it is set to 1, and VBUSOVEN is set to 1, the USBFS is powered on. NOTE: Only accessible in device mode.	R/W
b6	VBUSOVEN	VBUS Override enable	VBUS Override Enable (VBUS Override Enable) It is used to reflect the value set by VBUSVAL to the status of USBFS CORE. The value of VBUSVAL is valid only when this bit is set to 1. NOTE: Only accessible in device mode.	R/W
b5~b0	Reserved	-	The reset value must be maintained.	R/W

33.7.2.2 USBFS AHB Control Register (USBFS_GAHBCFG)

AHB Configuration Register

Offset address: 0x08

Reset value: 0x0000 0000

This register can be used to configure the module after power-up or when changing role modes. This register mainly contains configuration parameters related to the AHB system.

The application must program this register before starting any AHB or USB transaction. Do not change this register after initial programming.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved						PTXF ELVL	TXFE LVL	Rese rved	DMA EN	HBSTLEN [3:0]			GINT MSK		

Bit	Flag	Bit Name	Function	Read and write
b31~b9	Reserved	-	The reset value must be maintained.	R/W
b8	PTXFELVL	Periodic TxFIFO Empty Threshold	Periodic TxFIFO empty level Indicates when to trigger the periodic TxFIFO empty interrupt bit in the module interrupt register (PTXFE bit in USBFS_GINTSTS). 0: PTXFE (at USBFS_GINTSTS) interrupt indicates periodic TxFIFO is half-empty 1: PTXFE (at USBFS_GINTSTS) interrupt indicates that the periodic TxFIFO is completely empty NOTE: Only accessible in host mode.	R/W
b7	TXFELVL	Device TxFIFO Empty Threshold	Device TxFIFO empty level In device mode, this bit indicates when the IN endpoint transmit FIFO empty interrupt (TXFE in USBFS_DIEPINTx) is triggered. 0: TXFE (at USBFS_DIEPINTx) interrupt indicates that the IN endpoint TxFIFO is half empty 1: TXFE (located in USBFS_DIEPINTx) interrupt indicates that the IN endpoint TxFIFO is completely empty NOTE: Only accessible in device mode.	R/W
b6	Reserved	-	The reset value must be maintained.	R/W
b5	DMAEN	DMA enable	DMA enable (DMA enable) 0: Module operates in slave mode 1: The module operates in DMA mode Note: Accessible in both device mode and host mode.	R/W
b4~b1	HBSTLEN	Batch length/type	Batch length / type (Burst length/type) 0000b: single 0001b: INCR 0011b: INCR4 0101b: INCR8 0111b: INCR16 Other values: reserved Note: Accessible in both device mode and host mode.	R/W
b0	GINTMSK	Global Interrupt Mask	Global interrupt mask This bit is used to mask or unmask global interrupts. The Interrupt Status Register is updated by the module regardless of the setting of this bit. 0: Mask interrupts triggered by the application 1: Unmask application-triggered interrupts Note: Accessible in both device mode and host mode.	R/W

33.7.2.3 USBFS USB Configuration Register (USBFS_GUSBCFG)

USBFS USB configuration register

Offset address: 0x00C

Reset value: 0x0000 01440

This register can be used to configure the module after power-up or changing role modes. It contains configuration parameters related to USB and USB-PHY.

The application must program this register before starting any AHB or USB transaction. Do not change this register after initial programming.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved	FDMOD	FHMOD	Reserved												
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved		TRDT [3:0]				Reserved			PHYSEL	Reserved		TOTAL [2:0]			

Bit	Flag	Bit Name	Function	Read and Write
b31	Reserved	-	The reset value must be maintained.	R/W
b30	FDMOD	Force device mode	Force device mode Writing a 1 to this bit forces the module into device mode regardless of the state of the USBFS_ID input pin. 0: Normal mode, depending on the input state of the USBFS_ID pin 1: Force device mode After setting the force bit, the application must wait at least 25 ms for the change to take effect. Note: Accessible in both device mode and host mode.	R/W
b29	FHMOD	forced host mode	Force host mode Writing a 1 to this bit forces the module into host mode regardless of the state of the USBFS_ID input pin. 0: Normal mode, depending on the input state of the USBFS_ID pin 1: Force host mode After setting the force bit, the application must wait at least 25 ms for the change to take effect. Note: Accessible in both device mode and host mode.	R/W
b28~b14	Reserved	-	The reset value must be maintained.	R/W
b13~b10	TRDT	USB turnaround time	USB turnaround time Set the turnaround time in units of PHY clocks. To calculate the value of TRDT, use the following formula: $TRDT = 4 \times AHB \text{ clock} + 1 \text{ PHY clock}$ For example: 1. If AHB clock frequency = 84 MHz (PHY clock frequency = 48 MHz), then TRDT is set to 9. 2. If AHB clock frequency = 48 MHz (PHY clock frequency = 48 MHz), then TRDT is set to 5. NOTE: Only accessible in device mode.	R/W
b9~b7	Reserved	-	The reset value must be maintained.	R/W
b6	PHYSEL	Full speed serial transceiver selection	Full Speed serial transceiver select	R/W
b5~b3	Reserved	-	The reset value must be maintained.	R/W
b2~b0	TOTAL	FS timeout calibration	FS timeout calibration (FS timeout calibration) Additional latency introduced by the PHY includes the number of PHY clocks set by the application in this field, and the module's full-speed inter-packet timeout interval. The delay introduced by different PHYs has different effects on the state of the data line. The USB standard timeout value for full speed operation is 16 to 18 single digit times, inclusive. The application must program this field according to the enumerated speed. The number of bit times added per PHY clock is 0.25 bit times. Note: Accessible in both device mode and host mode.	R/W

33.7.2.4 USBFS Reset Register (USBFS_GRSTCTL)

USBFS reset register

Offset address: 0x10

Reset value: 0x8000 0000

The application program resets various hardware features in the module through this register.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
AHBI DL	DMA REQ	CSftR stDon e	Reserved												
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved					TXFNUM [4:0]				TXFF LSH	RXFF LSH	Rese rved	FCRS T	HSR ST	CSRS T	

Bit	Flag	Bit Name	Function	Read and Write
b31	AHBIDL	AHB Master Idle	AHB master idle (AHB master idle) Indicates that the AHB master state machine is in an idle condition. Note: Accessible in both device mode and host mode.	R
b30	DMAREQ	DMA request signal	DMA request signal (DMA request signal) This bit indicates that a DMA request is in progress. Used for debugging. Note: Accessible in both device mode and host mode.	R
b29	CSftRstDone	Reset completion signal	Reset completion signal 1 : All current logic resets are completed 0: No reset Note: This requires writing 1 to clear 0 , that is, when the reset is completed, you need to write '1' to clear the reset completion signal	R/W
b28~b11	Reserved	-	Read as "0", write as "0"	R/W
b10~b6	TXFNUM	TxFIFO number	TxFIFO number (TxFIFO number) FIFO number for FIFO flushing using the TxFIFO flush bit. This field should only be changed after the module has cleared the TxFIFO flush bit. ● 00000: — Flush aperiodic TxFIFO in host mode — Flush Tx FIFO 0 in device mode ● 00001: — Refresh periodic TxFIFO in host mode — Refresh TXFIFO 1 in device mode ● 00010: Refresh TXFIFO 2 in device mode ... ● 00101: Refresh TXFIFO 15 in device mode ● 10000: Flush all transmit FIFOs in device mode or host mode Note: Accessible in both device mode and host mode.	R/W
b5	TXFFLSH	TxFIFO refresh	TxFIFO refresh (TxFIFO flush) This bit selectively flushes one or all transmit FIFOs, but cannot do so while the module is processing a communication transaction. The application can only write to this bit after confirming that the module is not currently reading or writing to the TxFIFO. Confirm with the following registers: — Read: NAK active interrupt ensures that the module is not currently performing a read operation on the FIFO	R/W

			— Write: The AHBIDL bit in USBFS_GRSTCTL ensures that the module is not currently doing any writes to the FIFO Note: Accessible in both device mode and host mode.	
b4	RXFFLSH	RxFIFO refresh	RxFIFO refresh (RxFIFO flush) The application can use this bit to flush the entire RxFIFO, but must first ensure that the module is not currently processing a communication transaction. The application can write to this bit only after confirming that the module is not currently reading or writing to the RxFIFO. The application must wait until this bit is cleared before performing other operations. Usually there is a wait of 8 clock periods (whichever is the slowest of the PHY or AHB clock). Note: Accessible in both device mode and host mode.	R/W
b3	Reserved	-	The reset value must be maintained.	R/W
b2	FCRST	Host frame counter reset	Host frame counter reset When the application program writes to this bit, the frame counter in the module is reset. After the frame counter is reset, the frame number of the next SOF sent by the module is 0. Note: Accessible in both device mode and host mode.	R/W
b1	HSRST	HCLK domain logic soft reset	HCLK domain logic soft reset (HCLK soft reset) Applications use this bit to refresh the control logic in the AHB clock domain. Only the AHB clock domain pipeline is reset. FIFO is not flushed by this bit. After the transaction on the AHB is terminated according to the protocol, all state machines in the AHB clock domain are reset to the idle state. The CSR control bit used by the AHB clock domain state machine is cleared. To clear this interrupt, the status mask bit generated by the AHB clock domain state machine and used to control the status of the interrupt needs to be cleared. Since the interrupt status bit is not cleared, the application can get the status of all module events that occurred after this bit was set. state. This bit is self-clearing and the module will clear it when all necessary logic in it is reset. This process requires several clocks of time, depending on the current state of the module. Note: Accessible in both device mode and host mode.	R/W
b0	CSRST	Module soft reset	Module soft reset (Core soft reset) Reset the HCLK and PCLK domains as follows: Clear individual interrupt and all CSR register bits except: — GATEHCLK bit in USBFS_GCCTL — STPPCLK bit in USBFS_GCCTL — FLSPCS bit in USBFS_HCFG — DSPD bit in USBFS_DCFG Resets all module state machines (except AHB slaves) to idle state and clears all transmit and receive FIFOs. Terminate all transactions on the AHB master as soon as possible after the final data phase of the AHB transfer. Immediately terminates all transactions on the USB. The application can write to this bit anytime it needs to reset the module. The application determines whether the current reset operation is completed by reading the 29th bit of this register. When the reset is completed, the application needs to clear the soft reset and reset completion status signals at the same time. Software reset is usually used in two situations, one is during software development, and the other is after the user dynamically changes the PHY selection bits in the USB configuration registers listed above. When the user changes the PHY, the corresponding clock will be selected for the PHY	R/W

and used in the PHY domain. Once a new clock is selected,
the PHY domain must be reset for proper operation.
Note: Accessible in both device mode and host mode.

33.7.2.5 USBFS Global Interrupt Status Register (USBFS_GINTSTS)

USBFS interrupt status register

Offset address: 0x14

Reset value: 0x14000020

This register is used to interrupt the application with system-level events in the current mode (device or host).

Some bits in this register are valid only in host mode, while others are valid only in device mode. Additionally, this register indicates the current mode.

The FIFO status interrupts are read-only; if software reads or writes to the FIFO while these interrupts are being processed, the FIFO interrupt flag is automatically cleared.

Before enabling the interrupt bit, the application must clear the USBFS_GINTSTS register during initialization to avoid any interrupts before initialization.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
WKU INT	SESS REQ INT	DISC INT	CIDS CHG	LPMI INT	PTXF E	HCIN T	HPRT INT	Rese rved	DATA FSUS P	IPXF R/ INCO MPIS OOU T	IISOI XFR	OEPI NT	IEPIN T	Rese rved	Rese rved
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EOPF	ISOO DRP	ENU MDN E	USB RST	USBS USP	ESUS P	Rese rved	Rese rved	GON AKEF F	GINA KEFF	NPT XFE	RXF NE	SOF	Rese rved	MMI S	CMO D

Bit	Flag	Bit Name	Function	Read and Write
b31	WKUINT	Resume/Remote Wake Interrupt Detected	Resume / remote wakeup detected interrupt In device mode, this interrupt is triggered when a resume signal is detected on the USB bus. In host mode, this interrupt is triggered when a remote wakeup is detected on the USB. Write 1 to this bit to clear it to 0 by software. Note: Accessible in both device mode and host mode.	R/W
b30	SESSREQINT	SESSION interrupt request	USB Session Interrupt Request 0: No session currently 1 : New session request NOTE: Only accessible in device mode.	R/W
b29	DISCINT	Disconnect interrupt detected	Disconnect detected interrupt This interrupt is triggered when a device disconnection is detected. Write 1 to this bit to clear it to 0 by software. NOTE: Only accessible in host mode.	R/W
b28	CIDSCHG	Connector ID Line Status Change Interrupt	Connector ID status change The module sets this bit to 1 when the state of the Connector ID line changes. Write 1 to this bit to clear it to 0 by software. Note: Accessible in both device mode and host mode.	R/W
b27	LPMINT	LPM Interrupt	LPM Interrupt Device mode This bit is set to 1 when the device receives a correct LPM transmission and a valid response. Host Mode This position is set to 1 when the host receives a valid response after sending LPM or when the set number of retries is exceeded.	R/W
b26	PTXFEE	Periodic TxFIFO empty interrupt	Periodic TxFIFO empty interrupt This interrupt is triggered when the periodic transmit FIFO is half empty or fully empty and there is room in the periodic request queue to write at	R

			least one entry. Whether the FIFO is half empty or fully empty is determined by the periodic TxFIFO empty level bit in the USBFS_GAHBCFG register (PTXFELVL bit in USBFS_GAHBCFG). NOTE: Only accessible in host mode.	
b25	HCINT	Host channel interrupt	Host channels interrupt When the module sets this bit, it indicates that there is an interrupt pending on one of the channels in the module (in host mode). The application must read the host USBFS_HAINT register to determine the exact number of the channel on which the interrupt occurred, and then read the corresponding USBFS_HCINTx registers to determine the exact cause of the interrupt. The application must clear the corresponding status bit of the USBFS_HCINTx register before clearing this bit. NOTE: Only accessible in host mode.	R
b24	HPRTINT	Host port interrupt	Host port interrupt When the module sets this bit to 1, it indicates that the state of the USBFS controller port in host mode changes. The application must read the USBFS_HPRT register to determine the exact event that raised this interrupt. The application must clear the corresponding status bit of the USBFS_HPRT register before clearing this bit. NOTE: Only accessible in host mode.	R
b23	Reserved	-	The reset value must be maintained.	R/W
b22	DATAFSUSP	Data fetch pending	Data fetch suspended This interrupt is only valid in DMA mode. This interrupt indicates that the module stopped fetching data for the IN endpoint due to unavailable TxFIFO space or request queue space. The application uses this interrupt in the endpoint mismatch algorithm. For example, after detecting an endpoint mismatch, the application would do the following: — Set the global aperiodic IN NAK handshake signal to 1 — Disable IN endpoints — empty FIFO — Determine the token sequence based on the IN token sequence learning queue — re-enable the endpoint — If the global aperiodic IN NAK is cleared but the module has not yet acquired data for the IN endpoint and an IN token has been received at the same time, clear the global aperiodic IN NAK handshake signal: the module will generate "FIFO is empty when receiving IN token" interrupt. USBFS then sends a NAK response to the host. To avoid this situation, the application can check the DATAFSUSP interrupt in USBFS_GINTSTS, which can ensure that the global NAK handshake signal is cleared after the FIFO is full. Alternatively, the application can mask the "IN token received while FIFO empty interrupt" while clearing the global IN NAK handshake signal. Write 1 to this bit to clear it to 0 by software. NOTE: Only accessible in device mode.	R/W
b21	IPXFR/ INCOMPISOOUT	outstanding periodic transfers/ Incomplete OUT synchronous transfer	IPXFR: Incomplete periodic transfer In host mode, the module sets this interrupt bit if there are outstanding periodic transactions still pending that are scheduled to complete during the current frame. Write 1 to this bit to clear it to 0 by software. NOTE: Only accessible in host mode. INCOMPISOOUT: Incomplete isochronous OUT transfer In device mode, when the module asserts this interrupt, it indicates that at least one transfer on the isochronous OUT endpoint is not complete in the current frame. This interrupt is triggered with the Periodic End of Frame Interrupt (EOPF) bit in this register. Write 1 to this bit to clear it to 0 by software. NOTE: Only accessible in device mode.	R/W
b20	IISOIXFR	IN SYNC TRANSFER NOT COMPLETED	Incomplete isochronous IN transfer When the module sets this interrupt to 1, it indicates that the transmission on at least one synchronous IN endpoint in the current frame is not completed. This interrupt is triggered with the Periodic End of Frame Interrupt (EOPF) bit in this register. Write 1 to this bit to clear it to 0 by software. NOTE: Only accessible in device mode.	R/W
b19	OEPINT	OUT endpoint interrupt	OUT endpoint interrupt (OUT endpoint interrupt) When the module sets this bit, it indicates that there is an interrupt pending on an OUT endpoint in the module (in device mode). The application must read the host USBFS_DAIINT register to determine the exact number of the OUT endpoint on which the interrupt occurred, and then read the corresponding USBFS_DOEPINTx register to determine the exact cause of the interrupt. The application must clear the corresponding status bit of the corresponding USBFS_DOEPINTx register before clearing this bit. NOTE: Only accessible in device mode.	R
b18	IEPINT	IN endpoint interrupt	IN endpoint interrupt (IN endpoint interrupt) When the module sets this bit, it indicates that there is an interrupt pending on an IN endpoint in the module (in device mode). The	R

			application must read the host USBFS_DAIINT register to determine the exact number of the IN endpoint where the interrupt occurred, and then read the corresponding USBFS_DIEPINTx register to determine the exact cause of the interrupt. The application must first convert the corresponding The corresponding status bit of the USBFS_DIEPINTx register must be cleared before the bit can be cleared. NOTE: Only accessible in device mode.	
b17~b16	Reserved	-	The reset value must be maintained.	R/W
b15	EOPF	Periodic end-of-frame interrupt	End of periodic frame interrupt Indicates that the current frame has reached the period specified by the Periodic Frame Interval field (PFIVL bit in USBFS_DCFG) in the USBFS_DCFG register. Write 1 to this bit to clear it to 0 by software. NOTE: Only accessible in device mode.	R/W
b14	ISOODRP	Drop sync OUT packet interrupt	Isochronous OUT packet dropped interrupt The module sets this bit to 1 if the module cannot write a sync OUT packet to the RxFIFO due to insufficient space in the RxFIFO to accommodate the largest packet for the sync OUT endpoint. Write 1 to this bit to clear it to 0 by software. NOTE: Only accessible in device mode.	R/W
b13	ENUMDNE	Enum complete interrupt	Enumeration done interrupt When the module sets this bit to 1, it indicates that the velocity enumeration is complete. The application must read the USBFS_DSTS register to obtain the enumerated speed. Write 1 to this bit to clear it to 0 by software. NOTE: Only accessible in device mode.	R/W
b12	USBRST	USB reset interrupt	USB reset interrupt When the module sets this bit, it indicates that a reset signal was detected on the USB. Write 1 to this bit to clear it to 0 by software. NOTE: Only accessible in device mode.	R/W
b11	USBSUSP	USB suspend interrupt	USB suspend interrupt When the module sets this bit, it indicates that a suspend condition has been detected on the USB. When the idle state on the USB bus remains for 3ms, the module will enter the suspend state. Write 1 to this bit to clear it to 0 by software. NOTE: Only accessible in device mode.	R/W
b10	ESUSP	Early hang interrupt	Early suspend interrupt When the module sets this bit to 1, it indicates that the USB has been detected to be idle for 3ms. NOTE: Only accessible in device mode.	R/W
b9~b8	Reserved	-	The reset value must be maintained.	R/W
b7	GONAKEFF	Global OUT NAK active interrupt	Global OUT NAK effective interrupt (Global OUT NAK effective interrupt) Indicates that the "set global OUT NAK" bit (SGONAK bit in USBFS_DCTL) set by the application in the USBFS_DCTL register has taken effect in the module. This bit is cleared by writing to the "Clear Global OUT NAK" bit in the USBFS_DCTL register (CGONAK bit in USBFS_DCTL). NOTE: Only accessible in device mode.	R
b6	GINAKEFF	Global aperiodic IN NAK active interrupt	Global IN nonperiodic NAK effective interrupt Indicates that the "Set Global Aperiodic IN NAK" bit (SGINAK bit in USBFS_DCTL) set by the application in the USBFS_DCTL register has taken effect in the module. That is, the module has sampled the global IN NAK bit set by the application, and the result has taken effect. This bit is cleared by clearing the Clear Global Aperiodic IN NAK bit in the USBFS_DCTL register (CGINAK bit in UBSFS_DCTL). This interrupt does not necessarily indicate that a NAK handshake has been sent on the USB. The STALL bit has priority over the NAK bit. NOTE: Only accessible in device mode.	R
b5	NPTXFE	Aperiodic TxFIFO empty interrupt	Non-periodic TxFIFO empty interrupt This interrupt is triggered when the aperiodic TxFIFO is half or fully empty and there is room in the aperiodic transmit request queue to write at least one entry. Whether the FIFO is half empty or fully empty is determined by the aperiodic TxFIFO empty level bit in the USBFS_GAHBCFG register (TXFELVL bit in USBFS_GAHBCFG). NOTE: Only accessible in host mode.	R
b4	RXFNE	RxFIFO not empty interrupt	RxFIFO non-empty interrupt (RxFIFO non-empty interrupt) Indicates that there is at least one packet in the RxFIFO waiting to be read. Note: Accessible in both host mode and device mode.	R
b3	SOF	Start of frame interrupt	Start of frame interrupt In host mode, the module, when set to 1, indicates that a SOF (FS) or Keep-Alive (LS) signal has been sent on the USB. The application must set this bit to 1 to clear the interrupt. In device mode, the module, when set to 1, indicates that a SOF token has been received on the USB. The application program can get the	R/W

			current frame number by reading the device status register. This interrupt occurs only when the module is running in FS mode. Write 1 to this bit to clear it to 0 by software. Note: Accessible in both host mode and device mode.	
b2	Reserved	-	The reset value must be maintained.	R/W
b1	MMIS	Pattern mismatch interrupt	Mode mismatch interrupt The module sets this bit when the application attempts to access the following registers: — Module running in device mode accessing host mode registers — Module running in host mode to access device mode registers A register access ends with an OKAY response on the AHB, but the access is internally ignored by the module and does not affect module operation. Write 1 to this bit to clear it to 0 by software. Note: Accessible in both host mode and device mode.	R/W
b0	CMOD	Current working mode	Current mode of operation Indicates the current mode. 0: device mode 1: Host mode Note: Accessible in both host mode and device mode.	R

33.7.2.6 USBFS Global Interrupt Mask Register (USBFS_GINTMSK)

USBFS interrupt mask register

Offset address: 0x18

Reset value: 0x00000000

This register is used in conjunction with the module interrupt register to interrupt the application. If an interrupt bit is masked, the interrupt associated with that bit will not be generated.

However, the module interrupt (USBFS_GINTSTS) register bit corresponding to the interrupt will still be set.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
WKUIM	SESSREQINTMSK	DISCIM	CIDSCHGM	LPMINTM	PTXFEM	HCIM	HPRTIM	Reserved	DATAFSUSPM	IPXFRM/INCOMPISOUTM	IISOIXFRM	OEPI M	IEPI M	Reserved	Reserved
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EOPFM	ISOODRPM	ENUMDNEM	USB RSTM	USBSUSPM	ESUSPM	Reserved	Reserved	GONAKEFFM	GINAKEFFM	NPTXFEM	RXF NEM	SOF M	Reserved	MMIS M	Reserved

Bit	Flag	Bit Name	Function	Read and Write
b31	WKUIM	Resume/Remote Wakeup Detected Interrupt Mask	Resume/remote wakeup detected interrupt mask 0: mask interrupt 1: enable interrupt NOTE: Accessible in both host mode and device mode.	R/W
b30	SESSREQINTMSK	Session request interruption mask	SESSION REQ interrupt mask 0: mask interrupt 1: enable interrupt NOTE: Only accessible in device mode.	R/W
b29	DISCIM	Disconnect detected interrupt mask	Disconnect detected interrupt mask 0: mask interrupt 1: enable interrupt NOTE: Only accessible in host mode.	R/W
b28	CIDSCHGM	Interrupt Connector ID Line Status Change Interrupt Shield	Connector ID status change interrupt mask 0: mask interrupt 1: enable interrupt NOTE: Accessible in both device mode and host mode.	R/W
b27	LPMINTM	LPM interrupt mask	LPM interrupt mask 0: mask interrupt 1: enable interrupt	R/W
b26	PTXFEM	Periodic Tx FIFO empty interrupt mask	Periodic Tx FIFO empty interrupt mask 0: mask interrupt 1: enable interrupt NOTE: Only accessible in host mode.	R/W
b25	HCIM	Host channel interrupt mask	Host channels interrupt mask 0: mask interrupt 1: enable interrupt NOTE: Only accessible in host mode.	R/W
b24	HPRTIM	Host port interrupt mask	Host port interrupt mask 0: mask interrupt 1: enable interrupt NOTE: Only accessible in host mode.	R/W
b23	Reserved	-	The reset value must be maintained.	R/W
b22	DATAFSUSPM	Data Acquisition Pending Interrupt Mask	Data fetch suspended interrupt mask 0: mask interrupt 1: enable interrupt NOTE: Only accessible in device mode.	R/W
b21	IPXFRM/INCOMPISOUTM	Incomplete Periodic Transfer Interrupt Mask/ Incomplete OUT synchronous transfer interrupt mask	IPXFR: Incomplete periodic transfer interrupt mask 0: mask interrupt 1: enable interrupt NOTE: Only accessible in host mode.	R/W

			INCOMPIISOOUT: Incomplete isochronous OUT transfer interrupt mask 0: mask interrupt 1: enable interrupt NOTE: Only accessible in device mode.	
b20	IISOIXFRM	Incomplete IN synchronous transfer interrupt mask	Incomplete isochronous IN transfer interrupt mask 0: mask interrupt 1: enable interrupt NOTE: Only accessible in device mode.	R/W
b19	OEPIM	OUT endpoint interrupt mask	OUT endpoint interrupt mask (OUT endpoint interrupt mask) 0: mask interrupt 1: enable interrupt NOTE: Only accessible in device mode.	R/W
b18	IEPIM	IN endpoint interrupt mask	IN endpoint interrupt mask (IN endpoint interrupt mask) 0: mask interrupt 1: enable interrupt NOTE: Only accessible in device mode.	R/W
b17~b16	Reserved	-	The reset value must be maintained.	R/W
b15	EOPFM	Periodic end-of-frame interrupt mask	End of periodic frame interrupt mask 0: mask interrupt 1: enable interrupt NOTE: Only accessible in device mode.	R/W
b14	ISOODRPM	Discard Sync OUT Packet Interrupt Mask	Isochronous OUT packet dropped interrupt mask 0: mask interrupt 1: enable interrupt NOTE: Only accessible in device mode.	R/W
b13	ENUMDNEM	Enumeration complete interrupt mask	Enumeration done interrupt mask 0: mask interrupt 1: enable interrupt NOTE: Only accessible in device mode.	R/W
b12	USBRSTM	USB reset interrupt mask	USB reset interrupt mask 0: mask interrupt 1: enable interrupt NOTE: Only accessible in device mode.	R/W
b11	USBSUSPM	USB suspend interrupt mask	USB suspend interrupt mask 0: mask interrupt 1: enable interrupt NOTE: Only accessible in device mode.	R/W
b10	ESUSPM	Early Suspend Interrupt Mask	Early suspend interrupt mask 0: mask interrupt 1: enable interrupt NOTE: Only accessible in device mode.	R/W
b9~b8	Reserved	-	The reset value must be maintained.	R/W
b7	GONAKEFFM	Global OUT NAK active interrupt mask	Global OUT NAK effective interrupt mask (Global OUT NAK effective interrupt mask) 0: mask interrupt 1: enable interrupt NOTE: Only accessible in device mode.	R/W
b6	GINAKEFFM	Global aperiodic IN NAK active interrupt mask	Global IN nonperiodic NAK effective interrupt mask 0: mask interrupt 1: enable interrupt NOTE: Only accessible in device mode.	R/W
b5	NPTXFEM	Aperiodic Tx FIFO empty interrupt mask	Non-periodic Tx FIFO empty interrupt mask 0: mask interrupt 1: enable interrupt NOTE: Only accessible in host mode.	R/W
b4	RXFNEM	Rx FIFO Not Empty Interrupt Mask	Rx FIFO non-empty interrupt mask (Rx FIFO non-empty interrupt mask) 0: mask interrupt 1: enable interrupt Note: Accessible in both host mode and device mode.	R/W
b3	SOFM	Start of frame interrupt mask	Start of frame interrupt mask 0: mask interrupt 1: enable interrupt Note: Accessible in both host mode and device mode.	R/W
b2	Reserved	-	The reset value must be maintained.	R/W
b1	MMISM	Pattern Mismatch Interrupt Mask	Mode mismatch interrupt mask 0: mask interrupt 1: enable interrupt Note: Accessible in both host mode and device mode.	R/W
b0	Reserved	-	The reset value must be maintained.	R/W

33.7.2.7 USBFS Receive Status Debug Read /USBFS Status Read and Stack Register (USBFS_GRXSTSR/USBFS_GRXSTSP)

USBFS Receive status debug read/USBFS status read and pop registers

Read offset address: 0x01C

The offset address of the stack: 0x020

Reset value: 0x0000 0000

Reading the receive status debug read register will return the contents of the top of the receive FIFO. Reading the Receive Status Read and Pop registers will additionally pop the data entry from the top of the RxFIFO. The receive status content is interpreted differently in host mode and device mode.

When the receive FIFO is empty, the module will ignore the read or stack operation of the register and return the value 0x0000 0000. The application must only pop the receive status FIFO when the Receive FIFO Not Empty bit of the Module Interrupt Register (RXFNE bit in USBFS_GINTSTS) is set.

Host mode:

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved											PKTSTS [3:0]			DPID [1]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DPID [0]		BCNT [10:0]										CHNUM [3:0]			

Bit	Flag	Bit Name	Function	Read and Write
b31~b21	Reserved	-	The reset value must be maintained.	R/W
b20~b17	PKTSTS	Packet status	Packet status Indicates the status of received packets 0010: IN packet received 0011: IN transfer complete (trigger interrupt) 0101: Data synchronization error (trigger interrupt) 0111: Pause channel (trigger interrupt) Other values: reserved	R
b16~b15	DPID	Data PID	Data PID (Data PID) Indicates the data PID of the received packet 00: DATA0 10: DATA1 01: DATA2 11: MDATA	R
b14~b4	BCNT	Byte count	Byte count Indicates the number of bytes of the IN packet received.	R
b3~b0	CHNUM	Channel number	Channel number Indicates the channel number to which the currently received packet belongs.	R

Device mode:

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved											PKTSTS [3:0]			DPID [1]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DPID [0]		BCNT [11:0]									EPNUM [3:0]				

Bit	Flag	Bit Name	Function	Read and Write
b31~b21	Reserved	-	The reset value must be maintained.	R/W
b20~b17	PKTSTS	Packet status	Packet status Indicates the status of received packets 0001: Global OUT NAK (trigger interrupt) 0010: OUT packet received 0011: OUT transfer completed (interrupt triggered) 0100: SETUP transaction completed (interrupt triggered) 0110: SETUP packet received Other values: reserved	R
b16~b15	DPID	Data PID	Data PID (Data PID) Indicates the data PID of the received OUT packet 00: DATA0 10: DATA1 01: DATA2 11: MDATA	R
b14~b4	BCNT	Byte count	Byte count Indicates the number of bytes of the packet received.	R
b3~b0	EPNUM	Endpoint number	Endpoint number Indicates the endpoint number to which the currently received packet belongs.	R

33.7.2.8 USBFS Receive FIFO Size Register (USBFS_GRXFSIZ)

USBFS Receive FIFO size register

Offset address: 0x024

Reset value: 0x0000 0280

This application can program the size of RAM that must be allocated to the RxFIFO.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved					RXFD [10:0]										
Bit	Flag	Bit Name	Function												Read and Write
b31~b11	Reserved	-	The reset value must be maintained.												R/W
b10~b0	RXFD	RxFIFO depth	RXFD: RxFIFO depth (RxFIFO depth) In units of 32-bit words. Minimum value is 16 The maximum value is 256 Power-on reset value is the maximum Rx data FIFO depth.												R/W

33.7.2.9 USBFS Host Aperiodic Transmit FIFO Size Register (USBFS_HNPTXFSIZ)/Endpoint 0 Transmit FIFO Size Register (USBFS_DIEPTXF0)

USBFS Host non-periodic transmit FIFO size register/Device endpoint0 transmit FIFO size register

Offset address: 0x028

Reset value: 0x02800280

This application can program the size of RAM that must be allocated to the TxFIFO.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
NPTXFD [15:0]/TX0FD [15:0]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
NPTXFSA [15:0]/TX0FSA [15:0]															
Bit	Flag	Bit Name	Function												Read and Write
b31~b16	NPTXFD/ TX0FD	Aperiodic TxFIFO Depth/Endpoint 0 TxFIFO Depth	Host mode: NPTXFD Non -periodic TxFIFO depth In units of 32-bit words. Minimum value is 16 The maximum value is 256												R/W
			Device Mode: TX0FD Endpoint 0 TxFIFO depth In units of 32-bit words. Minimum value is 16 The maximum value is 256												
b15~b0	NPTXFSA/ TX0FSA	Aperiodically send RAM start address/endpoint 0 send RAM start address	Host mode: NPTXFSA Non- periodic transmit RAM start address This field contains the memory start address of the aperiodic transmit FIFO RAM.												R/W
			Device Mode: TX0FSA Endpoint 0 transmit RAM start address This field contains the memory start address of the endpoint 0 transmit FIFO RAM.												

33.7.2.10 USBFS Aperiodic Transmit FIFO/Queue Status Register (USBFS_HNPTXSTS)

USBFS Host non-periodic transmit FIFO size register/Device endpoint0 transmit FIFO size register

Offset address: 0x02C

Reset value: 0x00080280

This read-only register contains free space information for the aperiodic TxFIFO and aperiodic transmit request queues.

This register is valid only in host mode, not in device mode.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved	NPTXQTOP [6:0]							NPTQXSAV [7:0]							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
NPTXFSAV [15:0]															

Bit	Flag	Bit Name	Function	Read and write
b31	Reserved	-	The reset value must be maintained.	R/W
b30~b24	NPTXQTOP	Aperiodic send request queue top	Top of the non-periodic transmit request queue Entries currently being processed by the MAC in the aperiodic send request queue. Bit 30:27 : Channel / endpoint number Bits 26:25: — 00: IN/OUT token — 01: send packet with zero length — 11: Channel stop command Bit 24 : Terminate (last entry for selected channel)	R
b23~b16	NPTQXSAV	Non-periodic transmit request queue available space	Non-periodic transmit request queue available space (Non-periodic transmit request queue space available) Indicates the amount of free space available in the aperiodic send request queue. In host mode, this queue holds IN and OUT requests. 0: Aperiodic send request queue is full 1: 1 location available 2: 2 location available n : n positions available (where, n range: 0~PTXFD) Other values: reserved	R
b15~b0	NPTXFSAV	Non-periodic TxFIFO Available Space	Aperiodic TxFIFO Space Available (Non-periodic TxFIFO space available) Indicates the amount of free space available in the aperiodic TxFIFO. In units of 32-bit words. 00: Aperiodic TxFIFO is full 01: 1 word available 10: 2 word available 0xn: n words are available (among them, n range: 0~256) Other values: reserved	R

33.7.2.11 USBFS Module ID Register (USBFS_CID)

USBFS core ID register

Offset address: 0x03C

Reset value: 0x12345678

This register is a programmable user configuration ID register.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
PRODUCT_ID [31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PRODUCT_ID [15:0]															
Bit	Flag	Bit Name	Function	Read and Write											
b31~b0	PRODUCT_ID	Product ID field	Product ID field (Product ID field) Application-programmable ID field.	R/W											

33.7.2.12 USBFS LPM Configuration Register (USBFS_GLPMCFG)

USBFS LPM configuration register

Offset address: 0x054

Reset value: 0x00000000

This register is the programmable LPM configuration register.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved			ENBS EL	LPMRCNTSTS [2:0]			SNDL PM	LPMRCNT [2:0]			LPMCHIDX [3:0]			L1RS MOK	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SLPS TS	LPMRSP [1:0]		L1DS EN	BSELTHRS [3:0]			L1SS EN	REM WAKE	BSEL [3:0]			LPM ACK		LPME N	

Bit	Flag	Bit Name	Function	Read and write
b31~b29	Reserved	-	The reset value must be maintained.	R/W
b28	ENBSEL	BSEL enable bit	BSEL enable bit 0: Follow USB 2.0 Link Power Management Addendum Engineering Change Notice to the USB 2.0 specification, July 16, 2007 1 : Follow Errata for USB 2.0 ECN: Link Power Management (LPM) - 7/2007	R/W
b27~b25	LPMRCNTSTS [2:0]	LPM Retry Status Register	LPM Retry Status Register Record the remaining number of LPM retries, only valid in host mode	R
b24	SENDLPM	Send LPM transmission	Send LPM transmission 0: Do not send LPM transmission 1 : Send LPM transfer (EXT and LPM tokens) This bit is automatically cleared when the device returns a response (ACK, STALL or NYET). Supported in host mode only.	R/W
b23~b21	LPMRCNT [2:0]	LPM Retry Count Register	LPM Retry Count Register When receiving an ERROR response, the host sends the number of retries for LPM transmission until a valid response (ACK, STALL or NYET) is received. Supported in host mode only.	R/W
b20~b17	LPMCHIDX [2:0]	LPM Transmit Channel Index Register	LPM Transmit Channel Index Register The host channel index used for LPM transmission. The hardware automatically inserts the device address and endpoint information set for the channel. Supported in host mode only.	R/W
b16	L1RSMOK	L1 status bits	L1 status bits 0: The current state cannot be restored from L1 1 : The current state can be restored from L1	R
b15	SLPSTS	sleep status bit	sleep status bit Device mode When the device sends an ACK response to LPM transmission and the time specified by the protocol has passed, this position is set to 1 , indicating that it has entered SLEEP mode. When the bus status changes, or the connection is disconnected, or a remote wake-up signal is sent, this bit is automatically cleared to 0 and the SLEEP mode is exited. Host Mode The host sends an LPM transmission and receives an ACK response from the device, which is set to 1, indicating that it has entered SLEEP mode. When the host receives a remote wake-up signal, or initiates a wake-up, or initiates a reset, this bit is automatically cleared to 0 and exits SLEEP mode. 0: Not in L1 1: In L1	R
b14~b13	LPMRSP [1:0]	LPM Response	LPM Response LPM response received by the host or sent by the slave 00b: ERROR (no response) 01b: STALL 00b: NYET	R

00b: ACK			
b12	L1DSEN	L1 deep sleep enable bit	L1 deep sleep enable bit To minimize power consumption, the application needs to set this bit to 1. R
b11~b8	BSELTHRS [3:0]	BSEL Threshold Setting Register	BSEL Threshold Setting Register Device mode When the device receives a BSEL value greater than or equal to the register setting, it enters L1. R/W
			Host Mode The time when the host sends the resume signal.
b7	L1SSEN	L1 shallow sleep enable bit	L1 shallow sleep enable bit To minimize power consumption, the application needs to set this bit to 1. R/W
b6	REMWAKE	bRemoteWake Value	bRemoteWake Value bRemoteWake value sent by the host or received by the device R/W
b5~b2	BSEL [3:0]	BSEL register	BSEL register Device mode BSEL bmAttribute value, automatically updated after a valid response LPM transmission.
			Host Mode The host sends the BSEL value of the LPM transmission , or the resume signal length value.
			0000b: 125us
			0001b: 150us
			0010b: 200us
			0100b: 400us
			0101b: 500us
			0110b: 1000us
			0111b: 2000us
			1000b: 3000us
			1001b: 4000us
			1010b: 5000us
			1011b: 6000us
			1100b: 7000us
			1101b: 8000us
			1110b: 9000us
			1111b: 10000us
b1	LPMACK	LPM Acknowledge Register	LPM Acknowledge Register 1: ACK Even if this bit is written as 1 , no ACK response is sent if there is an error in the LPM transmission. There is a PID/CRC error, ERROR response bLinkState is not equal to 0001b , STALL response There is a pending data transfer, NYET answer R/W
			0: NYET Only device mode is valid
b0	LPMEN	LPM enable bit	LPM enable bit 0: LPM disabled 1: LPM enable R/W

33.7.2.13 The USBFS Host Periodically Transmits the FIFO Size Register (USBFS_HPTXFSIZ)

USBFS Host periodic transmit FIFO size register

Offset address: 0x100

Reset value: 0x02800500

This application can program the size of RAM that must be allocated to the period TxFIF.

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

Bit	Flag	Bit Name	Function	Read and Write
b31~b16	PTXFD	Host Periodic TxFIFO Depth	Host periodic TxFIFO depth In units of 32-bit words. Minimum value is 16	R/W
b15~b0	PTXSA	Host periodic TxFIFO start address	Host periodic TxFIFO start address The power-on reset value is the sum of the maximum RxFIFO depth and the maximum aperiodic TxFIFO depth.	R/W

33.7.2.14 USBFS Device IN Endpoint x Transmit FIFO Size Register (USBFS_DIEPTXFx) (x = 1..5)

USBFS device IN endpoint transmit FIFO size register

Offset address: $0x104 + (x-1) \times 0x4$

Reset value: $0x02800500 + (x-1) \times 0x280$

This application can program the size that must be allocated to the device TxFIFO.

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

Bit	Flag	Bit Name	Function	Read and Write
b31~b16	INEPTXFD	IN endpoint TxFIFO depth	Device IN endpoint TxFIFO depth In units of 32-bit words. Minimum value is 16	R/W
b15~b0	INEPTXSA	IN endpoint TxFIFOx RAM starting address	IN Endpoint FIFOx RAM Start Address (IN endpoint FIFOx transmit RAM start address) This field contains the memory start address of the IN endpoint transmit FIFOx. This address must be aligned to a 32-bit memory location.	R/W

33.7.3 USBFS Host Mode Register

The host mode register affects the operation of the module in host mode. Host mode registers must not be accessed in device mode, as the results produced are ambiguous.

Bit values in register descriptions are in binary unless otherwise noted.

33.7.3.1 USBFS Host Configuration Register (USBFS_HCFG)

USBFS Host configuration register

Offset address: 0x400

Reset value: 0x00000200

This register will configure the module after power-up. Do not change this register after initializing the host.

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

Bit	Flag	Bit Name	Function	Read and Write
b31~b3	Reserved	-	The reset value must be maintained.	R/W
b2	FSLSS	Only FS and LS are supported	Applications use this bit to control the enumeration speed of modules. Using this bit, the application can make the module work as FS host even if the connected device supports HS communication. Do not change this field after initial programming. 1: FS/LS only, even if the connected device can support HS	R/W
b1~b0	FSLSPCS	FS/LS PHY clock selection	FS/LS PHY clock select When the module is in FS host mode 01: PHY clock runs at 48 MHz Other values: reserved When the module is in LS host mode 00: reserved 01: Select 48MHz PHY clock frequency 10: Select 6MHz PHY clock frequency 11: reserved Note: When the device is connected to the host, the FSLSPCS must be set according to the speed of the connected device (after changing this bit, a software reset must be performed).	R/W

33.7.3.2 USBFS Host Frame Interval Register (USBFS_HFIR)

USBFS Host frame interval register

Offset address: 0x404

Reset value: 0x0000EA60

This register is used to store the frame interval information set by the USBFS controller for the current speed of the connected device.

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

Bit	Flag	Bit Name	Function	Read and Write
b31~b26	Reserved	-	Must hold the reset value	R/W
b15~b0	FRIVL	Frame interval	Frame interval The value programmed by the application in this field is used to specify the time interval between two consecutive SOF(FS) or Keep-Alive tokens(LS). This field contains the number of PHY clocks that make up the desired frame interval. The application can write a value to this register only after setting the port enable bit of the host port control and status register (PENA bit of USBFS_HPRT). If no value is programmed, the module will calculate based on the PHY clock specified in the FS/LS PHY Clock Select field of the Host Configuration Register (FSLSPCS in USBFS_HCFG). Do not change the value of this field after initial configuration. Setting value = frame interval (ms) × (PHY clock frequency) -1 Note: The FRIVL bits can be modified whenever the application needs to change the frame interval time.	R/W

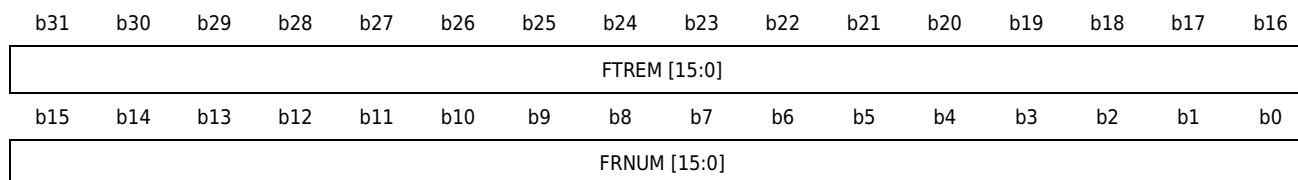
33.7.3.3 USBFS Host Frame Number/Frame Remaining Interval Register (USBFS_HFNUM)

USBFS Host frame interval register

Offset address: 0x408

Reset value: 0x0000 3FFF

This register is used to indicate the current frame number. It also indicates the remaining time of the current frame in number of PHY clocks.



Bit	Flag	Bit Name	Function	Read and write
b31~b16	FTREM	Frame time remaining	Frame time remaining Indicates the remaining time (in PHY clocks) of the current frame. This field is decremented by 1 every 1 PHY clock elapsed. When the value reaches zero, this field is reloaded with the value in the Interframe Register and a new SOF is sent by the module on the USB.	R
b15~b0	FRNUM	frame number	Frame number The value of this field will be incremented by 1 when a new SOF is sent on the USB, and will be cleared to 0 when 0x3FFF is reached.	R

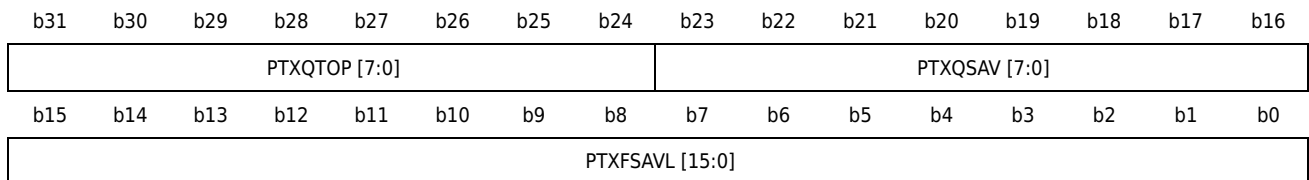
33.7.3.4 USBFS Host Periodic Transmit FIFO/Queue Status Register (USBFS_HPTXSTS)

USBFS Host periodic transmit FIFO/queue status register

Offset address: 0x410

Reset value: 0x00080280

This read-only register contains free space information for the periodic TxFIFO and periodic transmit request queues.



Bit	Flag	Bit Name	Function	Read and Write
b31~b24	PTXQTOP	Periodically send the request to the top of the queue	Periodically send the request to the top of the queue (Top of the periodic transmit request queue) Indicates the item in the periodic Tx request queue that the MAC is currently processing. This register is used for debugging. Bit 31 : Odd / Even frame — 0: send in even frames — 1: send in odd-numbered frames Bit 30:27 : Channel number Bit 26:25 : Type — 00: Input/Output — 01: Zero-length packet — 11: disable channel command Bit 24 : Terminate (last entry for the selected channel)	R
b23~b16	PTXQSAV	Periodically send request queue free space	Periodically send request queue free space (Periodic transmit request queue space available) Indicates the number of free slots in the periodic send request queue available for writing. The queue contains both IN requests and OUT requests. 0: Periodic send request queue is full 1: 1 position available 2: 2 position available n: n positions available (where, n range: 0~PTXFD) Other values: reserved	R
b15~b0	PTXFSAVL	Periodically send data FIFO free space	Periodically send data FIFO free space (Periodic transmit data FIFO space available) Indicates the number of free slots of the periodic TxFIFO available for writing. in units of 32-bit words 0: Periodic TxFIFO is full 1: 1 word available 2: 2 word available n : n words available (where, n range: 0 to PTXFD) Other values: reserved	R

33.7.3.5 USBFS Host Overall Channel Interrupt Register (USBFS_HAINT)

USBFS Host all channels interrupt register

Offset address: 0x414

Reset value: 0x0000 0000

When an event occurs on the channel, the host overall channel interrupt register will use the host channel interrupt bit (HCINT bit in USBFS_GINTSTS) in the module interrupt register to interrupt the application. Each channel corresponds to 1 interrupt bit, up to 16 bits.

Bits in this register are also cleared when the application clears an interrupt via the corresponding host channel x interrupt register.

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

Bit	Flag	Bit Name	Function	Read and Write
b31~b16	Reserved	-	The reset value must be maintained.	R/W
b15~b0	HAINT	Channel interruption	Channel interrupt One bit per channel: bit 0 for channel 0 and bit 15 for channel 15.	R

33.7.3.6 USBFS Host All Channel Interrupt Mask Register (USBFS_HAINTMSK)

USBFS Host all channels interrupt mask register

Offset address: 0x418

Reset value: 0x0000 0000

The Host All Channels Interrupt Mask Register is used in conjunction with the Host All Channels Interrupt Register to interrupt the application when an event occurs on the channel.

Each channel corresponds to 1 interrupt mask bit, up to 16 bits.

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

Bit	Flag	Bit Name	Function	Read and Write
b31~b16	Reserved	-	Read as "0", write as "0"	R/W
b15~b0	HAINTM	Channel interrupt mask	Channel interrupt mask 0: mask interrupt 1: enable interrupt One bit per channel: bit 0 for channel 0 and bit 15 for channel 15.	R/W

33.7.3.7 USBFS Host Port Control and Status Register (USBFS_HPRT)

USBFS Host port control and status register

Offset address: 0x440

Reset value: 0x0000 0000

This register is only available in host mode. Currently, the USBFS host only supports one port.

This register contains information related to the USB port, such as USB reset, enable, suspend, resume, connection status. The PENCHNG/PCDET bit in this register can trigger an application interrupt through the host port interrupt bit in the module interrupt register (HPRTINT bit in USBFS_GINTST). When a port interrupt occurs, the application must read this register and clear the bit that caused the interrupt. The application must write a 1 to this bit to clear the interrupt.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	
Reserved													PSPD [1:0]		Reserved	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
Reserved			PWPR	PLSTS [1:0]		Reserved	PRST	PSUSP	PRES	Reserved		PENCHG	PEN A	PCDET	PCSTS	
Bit		Flag		Bit Name		Function									Read and Write	
b31~b19		Reserved		-		The reset value must be maintained.									R/W	
b18~b17		PSPD		Port speed		Port speed Indicates the speed of the device connected to the port. 00/11: Reserved 01: full speed 10: low speed									R	
b16~b13		Reserved		-		The reset value must be maintained.									R/W	
b12		PPWR		Port power		Port power Applications use this field to control power to this port. Since the built-in PHY of this USBFS does not have power supply capability, when this bit is set to 1, the external USB power supply chip can be powered by USBFS_DRVVBUS. 0: power down 1: power on									R/W	
b11~b10		PLSTS		Port line status		Indicates the current logic level of the USB data line Bit 11: Logic level of USBFS_DM Bit 10: Logic level of USBFS_DP									R	
b9		Reserved		-		The reset value must be maintained.									R/W	
b8		PRST		Port reset		Port reset When an application sets this bit, it initiates a reset sequence on that port. The application must time the reset period and clear this bit after the reset sequence is complete. 0: Port is not in reset state 1: Port is in reset state The application must set this bit for a minimum of 10 ms to initiate a reset on the port.									R/W	
b7		PSUSP		Port hang		Port suspend Applications set this bit to put this port in suspend mode. Only when this bit is set will the module stop sending SOF. To stop the PHY clock, the application must set the Port Clock Stop bit, which enables the PHY's suspend input pin. A read value of this bit reflects the current pending state of the port. After the remote wake-up signal is detected, or the application program sets the port reset bit or port recovery bit in this register to 1, the module can clear this bit; or the application program sets the recovery/remote wake-up detection interrupt bit or interrupt bit in the module interrupt register The open connection detection interrupt bit (respectively WKUINT or DISCINT in USBFS_GINTSTS) is set to 1, and the module can also clear this bit.									R/W	

			0: Port is not in suspend mode 1: Port is in suspend mode	
b6	PRES	Port recovery	Port resume The application sets this bit to drive a resume signal on this port. The module will continue to drive the resume signal until the application program clears this bit. As indicated by the port resume/remote wakeup detect interrupt bit in the module interrupt register (WKUINT bit in USBFS_GINTSTS), if the module detects a USB remote wakeup sequence, it starts driving the resume signal without intervention from the application; if the module detects a disconnect If connected, clear this bit to 0. The read value of this bit indicates whether the current module The recovery signal is being driven. 0: Do not drive recovery signal 1: drive recovery signal	R/W
b5~b4	Reserved	-	The reset value must be maintained.	R/W
b3	PENCHNG	Port enable/disable change	Port enable / disable change The module sets this bit to 1 when the port enable bit 2 in this register changes state. Write 1 to this bit to clear it to 0 by software.	R/W
b2	PENA	Port enable	Port enable After the port performs a reset sequence, it can only be enabled by the module and can be disabled by an overcurrent condition, a disconnect condition, or by clearing this bit by the application. The application cannot set this bit by writing to the register. The port can only be disabled by clearing this bit. Manipulation of this bit will not trigger any interrupt to the application. 0: port disabled 1: Enable port	R/W
b1	PCDET	Port connection detected	Port connect detected When a device connection is detected, the module sets this bit to 1 to trigger an interrupt to the application using the host port interrupt bit in the module interrupt register (HPRTINT bit in USBFS_GINTSTS). The application must set this bit to 1 to clear the interrupt.	R/W
b0	PCSTS	Port connection status	Port connect status 0: No device is connected to the port 1: The port is connected to the device	R

33.7.3.8 USBFS Host Channel x Characteristic Register (USBFS_HCCHARx) (x = 0..11)

USBFS Host channel-x characteristics register

Offset address: 0x500 + (channel number × 0x20)

Reset value: 0x0000 0000

This register is used to set the host channel characteristics.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CHE NA	CHDI S	ODD FRM	DAD [6:0]							Reserved		EPTYP [1:0]		LSDE V	Rese rved
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EPDI R	EPNUM [3:0]				MPSIZ [10:0]										

Bit	Flag	Bit Name	Function	Read and Write
b31	CHENA	Channel enable	Channel enable This field is set by application software and cleared by USBFS host hardware. 0: Channel disabled 1: enable channel	R/W
b30	CHDIS	Channel prohibited	Channel disable The application sets this bit to 1 to stop sending/receiving data through the channel, even if the transmission through the channel is not completed, the stop operation still takes effect. The application must wait for the disabled channel interrupt to confirm that the channel has been disabled.	R/W
b29	ODDFRM	Odd-numbered frame	Odd frame This field is set or reset by the application to indicate that the USBFS host must transmit odd or even frames, respectively. This field applies only to periodic (synchronous and interrupt) transactions. 0: even frame 1: odd-numbered frames	R/W
b28~b22	DAD	Device address	Device address This field is used to specify a specific device to communicate with this host.	R/W
b21~b20	Reserved	-	The reset value must be maintained.	R/W
b19~b18	EPTYP	Endpoint type	Endpoint type Indicates the selected transport type. 00: control 01: synchronous 10: Batch 11: interrupt	R/W
b17	LSDEV	Low speed equipment	Low-speed device This field is set by the application to indicate that this channel is communicating with a low-speed device.	R/W
b16	Reserved	-	The reset value must be maintained.	R/W
b15	EPDIR	Endpoint direction	Endpoint direction Indicates whether the direction of the communication transaction is input or output. 0: output 1: input	R/W
b14~b11	EPNUM	Endpoint number	Endpoint number Indicates the endpoint number of the USB device to communicate with this host channel.	R/W
b10~b0	MPSIZ	Maximum packet size	Maximum packet size Indicates the maximum packet size for device endpoints communicating with this host channel.	R/W

33.7.3.9 USBFS Host Channel x Interrupt Register (USBFS_HCINTx) (x = 0..11)

USBFS Host channel-x interrupt register

Offset address: 0x508 + (channel number × 0x20)

Reset value: 0x0000 0000

This register indicates the state of the channel when USB and AHB related events occur. The application must read this register when the host channel interrupt bit (HCINT bit in USBFS_GINTSTS) is set in the module interrupt register. Before performing a read operation on the register, the application must first read the Host All Channel Interrupt (USBFS_HAINT) register to obtain the exact channel number of the host channel x interrupt register. The application must clear the corresponding bits in this register to clear the corresponding bits in the USBFS_HAINT and USBFS_GINTSTS registers.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved					DTE RR	FRM OR	BBE RR	TXER R	Rese rved	ACK	NAK	STAL L	AHB ERR	CHH	XFRC
Bit	Flag	Bit Name	Function												Read and Write
b31~b11	Reserved	-	The reset value must be maintained.												R/W
b10	DTERR	Data switching error	Data toggle error The application needs to write 1 to clear this bit.												R/W
b9	FRMOR	Frame overflow error	Frame overrun error The application needs to write 1 to clear this bit.												R/W
b8	BBERR	Crosstalk error	Babble error A typical cause of a crosstalk event is that an endpoint sends a packet, but the packet length exceeds the endpoint's maximum packet size. The application needs to write 1 to clear this bit.												R/W
b7	TXERR	Communication transaction error	Transaction error Indicates that one of the following errors occurred on the USB: CRC check failed Time-out Bit stuffing error Wrong EOP The application needs to write 1 to clear this bit.												R/W
b6	Reserved	-	The reset value must be maintained.												R/W
b5	ACK	ACK response received/sent	Received / transmitted ACK response (ACK response received/transmitted interrupt) The application needs to write 1 to clear this bit.												R/W
b4	NAK	NAK response received	NAK response received (NAK response received interrupt) The application needs to write 1 to clear this bit.												R/W
b3	STALL	STALL response received	STALL response received interrupt												R/W
b2	AHBERR	AHB error	AHB error only when in internal DMA mode and an AHB error occurs during an AHB read / write operation. The application can get the error address by reading the corresponding DMA channel address register. The application needs to write 1 to clear this bit.												R/W
b1	CHH	Channel stop	Channel halted The transfer ended abnormally due to any USB transaction error or in response to an application inhibit request. The application needs to write 1 to clear this bit.												R/W
b0	XFRC	Transmission completion	Transfer completed The transfer completed normally without any errors. The application needs to write 1 to clear this bit.												R/W

33.7.3.10 USBFS Host Channel x Interrupt Mask Register (USBFS_HCINTMSKx) (x = 0..11)

USBFS Host channel-x interrupt mask register

Offset address: 0x50C + (channel number × 0x20)

Reset value: 0x0000 0000

This register is used to select masking of host channel interrupts.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved					DTE RRM	FRM ORM	BBE RRM	TXER RM	Rese rved	ACK M	NAK M	STAL LM	AHB ERRM	CHH M	XFRC M
Bit	Flag	Bit Name				Function									Read and Write
b31~b11	Reserved	-				The reset value must be maintained.									R/W
b10	DTERRM	Data switching error interrupt mask				Data toggle error mask 0: mask interrupt 1: enable interrupt									R/W
b9	FRMORM	Frame overflow error interrupt mask				Frame overrun mask 0: mask interrupt 1: enable interrupt									R/W
b8	BBERRM	Crosstalk Error Interrupt Masking				Babble error mask 0: mask interrupt 1: enable interrupt									R/W
b7	TXERRM	Communication transaction error interrupt mask				Transaction error mask 0: mask interrupt 1: enable interrupt									R/W
b6	Reserved	-				The reset value must be maintained.									R/W
b5	ACKM	Received/sent ACK response interrupt mask				ACK response received/transmitted interrupt mask (ACK response received/transmitted interrupt mask) 0: mask interrupt 1: enable interrupt									R/W
b4	NAKM	NAK response interrupt mask received				NAK response received interrupt mask 0: mask interrupt 1: enable interrupt									R/W
b3	STALLM	Receive STALL response interrupt mask				Received STALL response (STALL response received interrupt mask) 0: mask interrupt 1: enable interrupt									R/W
b2	AHBERRM	AHBERR Interrupt Mask				AHBERR interrupt mask (AHB error mask) 0: mask interrupt 1: enable interrupt									R/W
b1	CHHM	Channel stop interrupt mask				Channel halted mask 0: mask interrupt 1: enable interrupt									R/W
b0	XFRCM	Transport complete interrupt mask				Transfer completed interrupt mask 0: mask interrupt 1: enable interrupt									R/W

33.7.3.11 USBFS Host Channel x Transmit Size Register (USBFS_HCTSIZx) (x = 0..11)

USBFS Host channel-x transfer size register

Offset address: 0x510 + (channel number × 0x20)

Reset value: 0x0000 0000

This register is used to set the host channel transfer size and data PID.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res	DPID [1:0]		PKTCNT [9:0]									XFRSIZ [18:16]			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
XFRSIZ [15:0]															
Bit	Flag	Bit Name	Function											Read and Write	
b31	Reserved	-	The reset value must be maintained.											R/W	
b30~b29	DPID	Data PID	Data PID (Data PID) The application sets the initial sync PID for data communication in this field. The host retains the setting of this field throughout the transfer transaction. 00: DATA0 01: reserved 10: DATA1 11: SETUP											R/W	
b28~b19	PKTCNT	Packet count	Packet count The application sets the number of packets to be sent or received in this field. The host decrements the count value each time a packet is successfully sent or received. After this value reaches 0, the application will be interrupted to indicate that the operation completed normally.											R/W	
b18~b0	XFRSIZ	transfer size	Transfer size For OUT operations, this field is the number of data bytes sent by the host during the transfer. For IN operations, this field is the buffer size reserved by the application for transfers. For IN transactions (periodic and aperiodic), the application will program this field to an integer multiple of the maximum packet size.											R/W	

33.7.3.12 USBFS Host Channel xDMA Address Register (USBFS_HCDMAx) (x = 0..11)

USBFS Host channel-x DMA address register

Offset address: 0x514 + (channel number × 0x20)

Reset value: 0xFFFF XXXX

This register is used to set the DMA address in host DMA mode.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
DMAADDR [31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DMAADDR [15:0]															
Bit	Flag	Bit Name	Function												Read and Write
b31~b0	DMAADDR	DMA address	DMA address (DMA address) This field stores the address of the memory used by the host for DMA transfers to obtain data from or send data to the device endpoint. This register is incremented at the end of each AHB transfer.												R/W

33.7.4 USBFS Device Mode Register

The device mode registers affect the operation of the module in device mode. Device mode registers must not be accessed in host mode because the results are ambiguous.

Bit values in register descriptions are in binary unless otherwise noted.

33.7.4.1 USBFS Device Configuration Register (USBFS_DCFG)

USBFS Device configuration register

Offset address: 0x800

Reset value: 0x0820 0000

This register configures the module into device mode after power-up, certain control commands, or enumeration. Do not change this register after initial programming.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved			PFIVL [1:0]		DAD [6:0]						Reserved	NZLS OHSK	DSPD [1:0]		

Bit	Flag	Bit Name	Function	Read and Write
b31~b13	Reserved	-	The reset value must be maintained.	R/W
b12~b11	PFIVL	periodic frame interval	Periodic frame interval Indicates the point within a frame at which the application must be notified using a periodic frame interrupt. This function can be used to determine if all isochronous communications for that frame are complete. 00: 80% frame interval 01: 85% frame interval 10: 90% frame interval 11: 95% frame interval	R/W
b10~b4	DAD	Device address	Device address The application must set this field according to the command parameter after executing each SetAddress control command.	R/W
b3	Reserved	-	The reset value must be maintained.	R/W
b2	NZLS OHSK	Non-zero length status OUT handshake signal	Non-zero length status OUT handshake signal (Non-zero-length status OUT handshake) During the OUT transaction of the control transfer status phase, the application can use this field to select the handshake signal to be sent after the module receives a non-zero length packet. 1: When receiving a non-zero-length state OUT transaction, reply to the STALL handshake signal, and the received OUT data packet is not sent to the application. 0: Send the received OUT packet (zero length or non-zero length) to the application, and reply the handshake signal based on the NAK and STALL bits of the endpoint in the device endpoint control register.	R/W
b1~b0	DSPD	Device speed	Device speed Indicates the speed at which the application requires the module to enumerate, or the maximum speed supported by the application. However, the actual bus speed can only be determined after the chirp sequence is completed, and this speed is based on the speed of the USB host connected to the module. 00: reserved 01: reserved 10: reserved 11: Full speed (USB 1.1 transceiver clock is 48 MHz)	R/W

33.7.4.2 USBFS Device Control Register (USBFS_DCTL)

USBFS Device control register

Offset address: 0x804

Reset value: 0x0000 0002

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved				POP RGD NE	CGO NAK	SGO NAK	CGIN AK	SGIN AK	Reserved			GON STS	GIN STS	SDIS	RWU SIG

Bit	Flag	Bit Name	Function	Read and write
b31~b12	Reserved	-	The reset value must be maintained.	R/W
b11	POPRGDNE	Power-on programming complete	Power-on programming done The application uses this bit to indicate that the register has finished programming after waking up from power-down mode.	R/W
b10	CGONAK	Clear global OUT NAK	Clear global OUT NAK (Clear global OUT NAK) Writing to this bit will clear the global OUT NAK.	R/W
b9	SGONAK	Set global OUT NAK	Set global OUT NAK (Set global OUT NAK) Writing to this bit sets the global OUT NAK. Applications use this bit to send NAK handshake signals on all OUT endpoints. The application program can only set this bit to 1 when it is sure that the global OUT NAK valid bit (GONAKEFF bit in USBFS_GINTSTS) in the module interrupt register has been cleared.	R/W
b8	CGINAK	Clear global IN NAK	Clear global IN NAK (Clear global IN NAK) Writing to this bit will clear the global IN NAK.	R/W
b7	SGINAK	Set global IN NAK	Set global IN NAK (Set global IN NAK) Writing to this field sets the global aperiodic IN NAK. Applications use this bit to cause all aperiodic IN endpoints to send NAK handshake signals. The application program can only set this bit to 1 when it is sure that the global IN NAK valid bit (GINAKEFF bit in USBFS_GINTSTS) in the module interrupt register has been cleared.	R/W
b6~b4	Reserved	-	The reset value must be maintained.	R/W
b3	GONSTS	Global OUT NAK status	Global OUT NAK status (Global OUT NAK status) 0: Handshake will be sent according to FIFO status and NAK and STALL bit settings. 1: No data is received regardless of whether there is free space in the Rx FIFO. Reply NAK handshake to all received packets except SETUP transactions. All OUT packets of type isochronous will be dropped.	R
b2	GINSTS	Global IN NAK status	Global IN NAK status (Global IN NAK status) 0: Handshake will be replied based on data availability in transmit FIFO. 1: Make all aperiodic IN endpoints reply with NAK handshake signals, regardless of the availability of data in the send FIFO.	R
b1	SDIS	Soft disconnect	Soft disconnect The application uses this bit to signal the USBFS module to perform a soft disconnect. When this bit is set, the host will not see that the device is attached, and the device will not receive signals on the USB. The module remains disconnected until the application program clears this bit. 0: Normal operation. Clearing this bit after a soft disconnect causes the host to receive a device connected event. After reconnecting the device, the USB host restarts device enumeration. 1: Make the host receive a device disconnect event. At full speed, the minimum time for soft disconnection is specified as follows: Suspended state: the minimum time is 1ms+2.5us Idle state: 2.5us Non-idle or suspend state: 2.5us	R/W
b0	RWUSIG	Send a remote wakeup signal	Sending a remote wakeup signal (Remote wakeup signaling) When the application program sets this bit to 1, the module will initiate a remote wake-up signal to wake up the USB host. The application must set this bit to bring the module out of the suspended state. According to the USB 2.0 specification, the application must clear this bit within 1 ms to 15	R/W

ms of setting it.

33.7.4.3 USBFS Device Status Register (USBFS_DSTS)

USBFS Device status register

Offset address: 0x808

Reset value: 0x0000 0002

This register indicates the state of the module when a USB related event occurs. When an interrupt occurs, the interrupted endpoint information must be read from the device overall interrupt (USBFS_DAINTE) register.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved								LNSTS [1:0]		FNSOF [13:8]					
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
FNSOF [7:0]								Reserved			EERR		ENUMSPD [1:0]	SUSPSTS	

Bit	Flag	Bit Name	Function	Read and Write
b31~b24	Reserved	-	The reset value must be maintained.	R/W
b23~b22	LNSTS	USB Bus Status	USB Bus Status LNSTS [1]: Logic level of D+ LNSTS [0]: Logic level of D-	R
b21~b8	FNSOF	Frame number of received SOF	Frame number of the received SOF (Frame number of the received SOF)	R
b7~b4	Reserved	-	The reset value must be maintained.	R/W
b3	EERR	Indeterminate error	Erratic error The module sets this bit to report any pending errors. Due to an indeterminate error, the USBFS controller enters the suspend state and generates an interrupt to the early suspend bit of the USBFS_GINTSTS register (ESUSP bit in USBFS_GINTSTS). If the early hang interrupt was triggered by an indeterminate error, the application can only perform a soft disconnect to resume communication.	R
b2~b1	ENUMSPD	Enumeration speed	Enumerated speed Indicates the speed that the USBFS controller is enumerated after detecting the speed through the chirp sequence. 01: reserved 10: reserved 11: Full speed (PHY clock running at 48 MHz) Other values: reserved	R
b0	SUSPSTS	Suspended state	Suspend status In device mode, this bit is set whenever a suspend condition is detected on the USB. When the idle state on the USB bus is kept for 3ms, the module will enter the suspend state. A module exits the suspended state when: — Activity on the USB cable — The application writes to the remote wakeup signal bit of the USBFS_DCTL register (RWUSIG bit in USBFS_DCTL).	R

33.7.4.4 USBFS Device IN Endpoint General Interrupt Mask Register (USBFS_DIEPMSK)

USBFS Device IN endpoint common interrupt mask register

Offset address: 0x810

Reset value: 0x0000 0000

This register works in conjunction with the individual USBFS_DIEPINTx registers for all endpoints to generate interrupts on each IN endpoint. IN endpoint interrupts in the USBFS_DIEPINTx registers can be masked by writing to the corresponding bit in this register. By default, status interrupts are masked.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved	NAK M	Reserved						INEP NEM	INEP NMM	TTXF EMSK	TOM	Reserved	EPD M	XFRM	M

Bit	Flag	Bit Name	Function	Read and Write
b31~b14	Reserved	-	The reset value must be maintained.	R/W
b13	NAKM	NAK Interrupt Mask	NAK interrupt mask (NAK mask) 0: mask interrupt 1: enable interrupt	R/W
b12~b7	Reserved	-	The reset value must be maintained.	R/W
b6	INEPNEM	IN endpoint NAK effective interrupt mask	IN endpoint NAK effective interrupt mask (IN endpoint NAK effective mask) 0: mask interrupt 1: enable interrupt	R/W
b5	INEPNMM	IN token interrupt mask received on EP mismatch	IN token interrupt mask received on EP mismatch (IN token received with EP mismatch mask) 0: mask interrupt 1: enable interrupt	R/W
b4	TTXFEMSK	IN token interrupt mask received while TxFIFO is empty	IN token interrupt mask received while TxFIFO is empty (IN token received when TxFIFO empty mask) 0: mask interrupt 1: enable interrupt	R/W
b3	TOM	Timeout Interrupt Mask (Asynchronous Endpoints)	Timeout Interrupt Mask (Asynchronous Endpoints) (Timeout condition mask (Non-isochronous endpoints)) 0: mask interrupt 1: enable interrupt	R/W
b2	Reserved	-	The reset value must be maintained.	R/W
b1	EPDM	Endpoint disable interrupt mask	disabled interrupt mask 0: mask interrupt 1: enable interrupt	R/W
b0	XFRM	Transfer complete interrupt mask	Transfer completed interrupt mask 0: mask interrupt 1: enable interrupt	R/W

33.7.4.5 USBFS Device OUT Endpoint General Interrupt Mask Register (USBFS_DOEPMSK)

USBFS Device OUT endpoint common interrupt mask register

Offset address: 0x814

Reset value: 0x0000 0000

This register works in conjunction with the individual USBFS_DOEPINTx registers for all endpoints to generate interrupts on each OUT endpoint. OUT endpoint interrupts in the USBFS_DOEPINTx registers can be masked by writing to the corresponding bit in this register. By default, status interrupts are masked.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved											OTEP DM	STUP M	Rese rved	EPD M	XFRC M
Bit	Flag	Bit Name	Function												Read and write
b31~b5	Reserved	-	The reset value must be maintained.												R/W
b4	OTEPDM	OUT token interrupt mask received when endpoint disabled	OUT token interrupt mask received when endpoint disabled (OUT token received when endpoint disabled mask) Applies to control OUT endpoints only. 0: mask interrupt 1: enable interrupt												R/W
b3	STUPM	The SETUP phase completes the interrupt mask	SETUP phase done interrupt mask (SETUP phase done mask) Applies to control endpoints only. 0: mask interrupt 1: enable interrupt												R/W
b2	Reserved	-	The reset value must be maintained.												R/W
b1	EPDM	Endpoint disable interrupt mask	disabled interrupt mask 0: mask interrupt 1: enable interrupt												R/W
b0	XFRCM	Transport complete interrupt mask	Transfer completed interrupt mask 0: mask interrupt 1: enable interrupt												R/W

33.7.4.6 USBFS Device Overall Endpoint Interrupt Register (USBFS_DAINR)

USBFS Device OUT endpoint common interrupt mask register

Offset address: 0x818

Reset value: 0x0000 0000

When a valid event occurs on the endpoint, the USBFS_DAINR register will interrupt the application via the device OUT endpoint interrupt bit or the device IN endpoint interrupt bit in the USBFS_GINTSTS register (OEPINT or IEPINT bits in USBFS_GINTSTS, respectively). Each endpoint corresponds to an interrupt bit, and both the OUT endpoint and the IN endpoint have up to 16 interrupt bits. Bidirectional endpoints will use the corresponding IN and OUT interrupt bits. When the application sets and clears a bit in the corresponding Device Endpoint x Interrupt Register (USBFS_DIEPINTx/USBFS_DOEPINTx), the corresponding bit in this register is also set and cleared.

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

Bit	Flag	Bit Name	Function	Read and write
b31~b16	OEPINT	OUT endpoint interrupt bit	OUT endpoint interrupt bits (OUT endpoint interrupt bits) One bit per OUT endpoint: OUT endpoint 0 corresponds to bit 16, and OUT endpoint 15 corresponds to bit 31.	R/W
b15~b0	IEPINT	IN endpoint interrupt bits	IN endpoint interrupt bits One bit per IN endpoint: IN endpoint 0 corresponds to bit 0 and IN endpoint 15 corresponds to bit 15.	R/W

33.7.4.7 USBFS Device All Endpoints Interrupt Mask Register (USBFS_DAINTRMSK)

USBFS Device all endpoints interrupt mask register

Offset address: 0x81C

Reset value: 0x0000 0000

The USBFS_DAINTRMSK register is used in conjunction with the device endpoint interrupt register to interrupt the application when an event occurs on the device endpoint. However, the USBFS_DAINTR register bit corresponding to the interrupt will still be set.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
OEPINTM [15:0]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
IEPINTM [15:0]															
Bit	Flag	Bit Name	Function												Read and Write
b31~b16	OEPINTM	OUT endpoint interrupt mask bits	OUT endpoint interrupt mask bits One bit per OUT endpoint: OUT endpoint 0 corresponds to bit 16, and OUT endpoint 15 corresponds to bit 31. 0: mask interrupt 1: enable interrupt												R/W
b15~b0	IEPINTM	IN endpoint interrupt mask bits	IN endpoint interrupt mask bits One bit per IN endpoint: IN endpoint 0 corresponds to bit 0 and IN endpoint 15 corresponds to bit 15. 0: mask interrupt 1: enable interrupt												R/W

33.7.4.8 USBFS Device IN Endpoint FIFO Empty Interrupt Mask Register (USBFS_DIEPEMPMSK)

USBFS Device IN endpoint FIFO empty interrupt mask register

Offset address: 0x834

Reset value: 0x0000 0000

This register is used to control the generation of IN endpoint FIFO empty interrupt.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
INEPTXFEM [5:0]															
Bit	Flag	Bit Name	Function												Read and Write
b31~b16	Reserved	-	The reset value must be maintained.												R/W
b15~b0	INEPTXFEM	IN EP TxFIFO empty interrupt mask bit	IN EP TxFIFO empty interrupt mask bit (IN EP Tx FIFO empty interrupt mask bits) These bits are used as mask bits for USBFS_DIEPINTx. Each bit corresponds to a TXFE interrupt of an IN endpoint: IN endpoint 0 corresponds to bit 0, while IN endpoint 15 corresponds to bit 15 0: mask interrupt 1: enable interrupt												R/W

33.7.4.9 USBFS Device IN Endpoint 0 Control Register (USBFS_DIEPCTL0)

USBFS Device control IN endpoint 0 control register

Offset address: 0x900

Reset value: 0x0000 8000

This register is used to control Control Transfer Endpoint 0.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EPEN A	EPDI S	Reserved		SN AK	CNA K	TXFNUM [3:0]				STAL L	Rese rved	EPTYP [1:0]		NAK STS	Rese rved
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
USB AEP	Reserved														
														MPSIZ [1:0]	

Bit	Flag	Bit Name	Function	Read and Write
b31	EPENA	Endpoint enable	Endpoint enable The application sets this bit to start a data send on endpoint 0. The module clears this bit before any of the following interrupts are triggered on this endpoint: — SETUP phase completed — Endpoint prohibition —Transmission completion	R/W
b30	EPDIS	Endpoint prohibition	Endpoint disable An application can set this bit to stop sending data on an endpoint even before the transfer on that endpoint is complete. The application must wait for an endpoint disable interrupt to occur before the endpoint is considered disabled. The module will clear this bit before the endpoint disable interrupt bit is set. An application can set this bit only if the Endpoint Enable bit for that endpoint is set.	R/W
b29~b28	Reserved	-	The reset value must be maintained.	R/W
b27	SNAK	Set the NAK bit	Set the NAK bit (Set NAK) Writing to this bit will set the NAK bit of the endpoint. With this bit, the application can control the sending of NAK handshake signals on the endpoint. The module can also set this bit of the endpoint to 1 after the endpoint receives the SETUP packet.	R/W
b26	CNAK	Clear the NAK bit	Clear the NAK bit (Clear NAK) Writing to this bit will clear the NAK bit of the endpoint.	R/W
b25~b22	TXFNUM	TxFIFO number	TxFIFO number (TxFIFO number) This value is set to the FIFO number assigned to IN endpoint 0. Only TX-FIFO0 can be used.	R/W
b21	STALL	STALL handshake	STALL handshake (STALL handshake) The application can only set this bit to 1, the module will clear this bit to 0 when the endpoint receives a SETUP token. If the NAK bit, global IN NAK, or global OUT NAK and this bit are both set, the STALL bit takes precedence.	R/W
b20	Reserved	-	The reset value must be maintained.	R/W
b19~b18	EPTYP	Endpoint type	Endpoint type The hardware is set to '00', indicating a control type endpoint.	R
b17	NAKSTS	NAK status	NAK status Indicates the following results: 0: Module replies to non-NAK handshake according to FIFO status. 1: The module replies with a NAK handshake on this endpoint. When this bit is set (either by the application or by the module), the module stops sending data even if there is still data available in the Tx FIFO. Regardless of how this bit is set, the module always responds to SETUP packets with an ACK handshake.	R
b16	Reserved	-	The reset value must be maintained.	R/W
b15	USBAEP	USB active endpoint	USB active endpoint This bit is always set to 1, indicating that Control Endpoint 0 is always active in all configurations and interfaces.	R
b14~b2	Reserved	-	The reset value must be maintained.	R/W
b1~b0	MPSIZ	Maximum packet size	Maximum packet size Applications must program this field to the maximum packet size for the current logical endpoint.	R/W

00: 64 bytes
01: 32 bytes
10: 16 bytes
11: 8 bytes

33.7.4.10 USBFS Device IN Endpoint x Control Register (USBFS_DIEPCTLx) (x=1..5)

USBFS Device IN endpoint x control register

Offset address: 0x900 + (endpoint number × 0x20)

Reset value: 0x0000 0000

Applications use this register to control the behavior of individual logical endpoints (except Endpoint 0).

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EPE NA	EPDI S	SOD DFR M	SD0 PID/ SEV NFR M	SNAK	CNA K	TXFNUM [3:0]				STAL L	Rese rved	EPTYP [1:0]		NAK STS	EONU M/ DPID
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
USB AEP	Reserved					MPSIZ [10:0]									

Bit	Flag	Bit Name	Function	Read and write
b31	EPENA	Endpoint enable	Endpoint enable The application sets this bit to initiate data sending on the endpoint. The module clears this bit before any of the following interrupts are triggered on this endpoint: — Endpoint prohibition —Transmission completion	R/W
b30	EPDIS	Endpoint prohibition	Endpoint disable An application can set this bit to stop sending data on an endpoint even before the transfer on that endpoint is complete. The application must wait for an endpoint disable interrupt to occur before the endpoint is considered disabled. The module will clear this bit before the endpoint disable interrupt bit is set. An application can set this bit only if the Endpoint Enable bit for that endpoint is set.	R/W
b29	SODDFRM	set odd-numbered frames	odd frame Applies only to synchronous IN and OUT endpoints. Writing to this field sets the Even/odd-numbered frame (EONUM) field to odd-numbered frames.	R/W
b28	SD0PID/ SEVNFRM	set DATA0 PID/ SEVNFRM	Set DATA0 PID (Set DATA0 PID) Applies to interrupt/bulk IN endpoints only. Writing to this field sets the endpoint data PID (DPID) field in this register to DATA0. SEVNFRM: Set even frame (Set even frame) Applies only to synchronous IN endpoints. Writing to this field sets the Even/odd-numbered frames (EONUM) field to even frames.	R/W
b27	SNAK	Set the NAK bit	Set the NAK bit (Set NAK) Writing to this bit will set the NAK bit of the endpoint. With this bit, the application can control the sending of NAK handshake signals on the endpoint. The module can also set this bit of the OUT endpoint to 1 when a transfer completion interrupt occurs or after SETUP is received on the endpoint.	R/W
b26	CNAK	Clear the NAK bit	Clear the NAK bit (Clear NAK) Writing to this bit will clear the NAK bit of the endpoint.	R/W
b25~b22	TXFNUM	TxFIFO number	TxFIFO number (TxFIFO number) These bits are used to specify the FIFO number associated with this endpoint. A separate FIFO number must be set for each valid IN endpoint. This field is only valid for IN endpoints.	R/W
b21	STALL	STALL handshake	STALL handshake (STALL handshake) Setting this bit by the application causes the device to reply STALL to all tokens from the USB host. If the NAK bit, global IN NAK, or global OUT NAK is set at the same time as this bit, the STALL bit takes precedence. Only the application can clear this bit, not the module.	R/W
b20	Reserved	-	The reset value must be maintained.	R/W
b19~b18	EPTYP	Endpoint type	Endpoint type The following are the transport types supported by this logical endpoint.	R

			00: control 01: synchronous 10: Batch 11: interrupt	
b17	NAKSTS	NAK status	NAK status Indicates the following results: 0: Module replies to non-NAK handshake according to FIFO status. 1: The module replies with a NAK handshake on this endpoint. When an application or module sets this bit: For non-synchronous IN endpoints: The module stops sending any data through the IN endpoint even if there is data available in the TxFIFO. For synchronous IN endpoints: The module sends zero-length packets even if there is data available in the TxFIFO. Regardless of how this bit is set, the module always responds to SETUP packets with an ACK handshake.	R
b16	EONUM/ DPID	Even/odd-numbered frames/ Endpoint data PID	Even / odd frame Applies only to synchronous IN endpoints. Indicates the number of the frame in which the module sends/receives synchronized data for this endpoint. The application must program the even/odd-numbered frame number through the SEVNFRM and SODDFRM fields in this register for this endpoint to send/receive isochronous data. 0: even frame 1: odd-numbered frames DPID: Endpoint data PID (Endpoint data PID) Applies to interrupt/bulk IN endpoints only. Contains the PID of packets that will be received or sent on this endpoint. After an endpoint is activated, the application must program the PID of the first packet to be received or sent on this endpoint. The application programs the DATA0 or DATA1 PID using the SD0PID register field. 0: DATA0 1: DATA1	R
b15	USBAEP	USB active endpoint	USB active endpoint Indicates whether this endpoint is active in the current configuration and interface. The module clears this bit to 0 for all endpoints except endpoint 0 when a USB reset is detected. After receiving the SetConfiguration and SetInterface commands, the application must program the endpoint register accordingly and set this bit to 1.	R/W
b14~b11	Reserved	-	The reset value must be maintained.	R/W
b10~b0	MPSIZ	Maximum packet size	Maximum packet size Applications must program this field to the maximum packet size for the current logical endpoint. This value is in bytes.	R/W

33.7.4.11 USBFS Device IN Endpoint x Interrupt Register (USBFS_DIEPINTx) (x=0..5)

USBFS Device IN endpoint x interrupt register

Offset address: 0x908 + (endpoint number × 0x20)

Reset value: 0x0000 0080

This register indicates the status of the endpoint on USB and AHB related events. The application must read this register when the IN endpoint interrupt bit (IEPINT bit in USBFS_GINTSTS) is set in the module interrupt register. Before an application can read this register, it must first read the Device Entire Endpoint Interrupt (USBFS_DAINTE) register to obtain the exact endpoint number of the device endpoint x interrupt register. The application must clear the corresponding bits in this register to clear the corresponding bits in the USBFS_DAINTE and USBFS_GINTSTS registers.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								TXFE	INEPNE	Reserved	TTXFE	TOC	Reserved	EPDISD	XFRC
Bit	Flag	Bit Name	Function											Read and Write	
b31~b8	Reserved	-	The reset value must be maintained.											R/W	
b7	TXFE	Transmit FIFO is empty	Transmit FIFO is empty (Transmit FIFO empty) This interrupt is asserted when the Tx FIFO for this endpoint is half or fully empty. Whether the Tx FIFO is half empty or fully empty is determined by the Tx FIFO blank bit in the USBFS_GAHBCFG register (TXFELVL bit in USBFS_GAHBCFG).											R	
b6	INEPNE	IN endpoint NAK valid	INEPNE: IN endpoint NAK effective (IN endpoint NAK effective) This bit can be cleared when the application NAKs the IN endpoint by writing data to the CNAK bit in USBFS_DIEPCTLx. This interrupt indicates that the module has sampled a NAK that was set (by the application or the module), and the result has taken effect. This interrupt indicates that the IN endpoint NAK bit set by the application program has acted in the module. This interrupt does not guarantee that a NAK handshake was sent on the USB. The STALL bit has priority over the NAK bit. Software can also write 1 to clear this bit.											R/W	
b5	Reserved	-	The reset value must be maintained.											R/W	
b4	TTXFE	IN token received when Tx FIFO is empty	IN token received when Tx FIFO is empty (IN token received when Tx FIFO is empty) Applies to acyclic IN endpoints only. When the Tx FIFO (periodic/aperiodic) corresponding to this endpoint is empty, an IN token is received and an interrupt is generated. Write 1 to clear by software.											R/W	
b3	TOC	Time-out	Timeout condition Applies to control IN endpoints only. Indicates that this endpoint timed out on the most recently received IN token response. Write 1 to clear by software.											R/W	
b2	Reserved	-	The reset value must be maintained.											R/W	
b1	EPDISD	Endpoint disable interrupt	Endpoint disable interrupt This bit indicates that the endpoint has been disabled by the application. Write 1 to clear by software.											R/W	
b0	XFRC	Transfer complete interrupt	Transfer completed interrupt This field indicates that transfers set on this endpoint have completed transfers on USB and AHB. Write 1 to clear by software.											R/W	

33.7.4.12 USBFS Device IN Endpoint 0 Transfer Size Register (USBFS_DIEPTSIZ0)

USBFS Device IN endpoint 0 transfer size register

Offset address: 0x910

Reset value: 0x0000 0000

The application must modify this register before enabling Endpoint 0. After enabling endpoint 0 through the endpoint enable bit (EPENA in USBFS_DIEPCTL0) in the device control endpoint 0 control register, the module modifies this register. The application can read this register only after the module has cleared the endpoint enable bit.

Non-zero endpoints use the registers of endpoints 1~15.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved											PKTCNT [1:0]		Reserved		
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved									XFRSIZ [6:0]						

Bit	Flag	Bit Name	Function	Read and Write
b31~b21	Reserved	-	The reset value must be maintained.	R/W
b20~b19	PKTCNT	Packet count	Packet count Indicates the number of data packets included in one data transmission of endpoint 0. This field will be decremented each time a packet (max size or short) is read from the Tx FIFO.	R/W
b18~b7	Reserved	-	The reset value must be maintained.	R/W
b6~b0	XFRSIZ	Transfer size	Transfer size Indicates the amount of data contained in one data transmission of endpoint 0, in bytes. The module interrupts the application only after the application has transferred this data. The transfer size can be set to the endpoint's maximum packet size to break at the end of each packet. The module decrements this field each time a packet from external memory is written to the Tx FIFO.	R/W

33.7.4.13 USBFS Device IN Endpoint x Transfer Size Register (USBFS_DIEPTSIZE_x) (x=1..5)

USBFS Device IN endpoint x transfer size register

Offset address: 0x910 + (endpoint number × 0x20)

Reset value: 0x0000 0000

The application must modify this register before enabling the endpoint. After enabling the endpoint through the endpoint enable bit (EPENA bit in USBFS_DIEPCTL_x) in the USBFS_DIEPCTL_x register, the module modifies this register. The application can read this register only after the module has cleared the endpoint enable bit.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res	MCNT [1:0]		PKTCNT [9:0]										XFRSIZ [18:16]		
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
XFRSIZ [15:0]															

Bit	Flag	Bit Name	Function	Read and Write
b31	Reserved	-	The reset value must be maintained.	R/W
b30~b29	MCNT	Multi count	Multi count For a periodic IN endpoint, this field indicates the number of packets that must be sent per frame on the USB. The module uses this field to calculate the data PID for an isochronous IN endpoint. 01: 1 packet 10: 2 packets 11: 3 packets	R/W
b28~b19	PKTCNT	Packet count	Packet count Indicates the number of data packets contained in a data transmission on this endpoint. This field will be decremented each time a packet (max size or short) is read from the TxFIFO.	R/W
b18~b0	XFRSIZ	Transfer size	Transfer size This field contains the amount of data contained in a data transmission of the current endpoint, in bytes. The module interrupts the application only after the application has transferred this data. The transfer size can be set to the endpoint's maximum packet size to break at the end of each packet. The module decrements this field each time a packet from external memory is written to the TxFIFO.	R/W

33.7.4.14 USBFS Device IN Endpoint x DMA Address Register (USBFS_DIEPDMAx) (x=0..5)

USBFS Device IN endpoint x transfer size register

Offset address: 0x914 + (endpoint number × 0x20)

Reset value: 0x0000 0000

This register is used to set the DMA address in device endpoint DMA mode.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
DMAADDR [31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DMAADDR [15:0]															
Bit	Flag	Bit Name	Function											Read and write	
b31~b0	DMAADDR	DMA address	DMA address (DMA address) This bit contains the start address of the external memory area when using DMA for data storage on the endpoint. Note: For control endpoints, the storage area pointed to by this field is also used to store control OUT packets and SETUP transaction packets. When more than three SETUP packets are received consecutively, the SETUP packets in the memory will be overwritten. This register is incremented for every AHB transfer. The application must set a double word aligned address.											R/W	

33.7.4.15 USBFS Device IN Endpoint Transmit FIFO Status Register (USBFS_DTXFSTSx)(x=0..5)

USBFS Device IN endpoint transmit FIFO status register

Offset address: 0x918 + (endpoint number × 0x20)

Reset value: 0x0000 0280

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
INEPTFSAV [15:0]															
Bit	Flag	Bit Name	Function												Read and Write
b31~b16	Reserved	-	The reset value must be maintained.												R/W
b15~b0	INEPTFSAV	IN endpoint Tx FIFO free space	IN endpoint Tx FIFO space available (IN endpoint Tx FIFO space available) Indicates the amount of free space available in the endpoint Tx FIFO. In 32-bit words: 0x0: Endpoint Tx FIFO is full 0x1: 1 word available 0x2: 2 words available 0xn: n words are available												R

33.7.4.16 USBFS Device OUT Endpoint 0 Control Register (USBFS_DOEPCTL0)

USBFS Device control OUT endpoint 0 control register

Offset address: 0xB00

Reset value: 0x0000 8000

This register is used to control Control Transfer Endpoint 0.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EPEN A	EPDI S	Reserved		SN AK	CNA K	Reserved				STAL L	SNP M	EPTYP [1:0]		NAK STS	Reser ved
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
USB AEP	Reserved														MPSIZ [1:0]

Bit	Flag	Bit Name	Function	Read and write
b31	EPENA	Endpoint enable	Endpoint enable The application sets this bit to 1 to initiate data reception on Endpoint 0. The module clears this bit before any of the following interrupts are triggered on this endpoint: — SETUP phase completed — Endpoint prohibition — Transmission completion	R/W
b30	EPDIS	Endpoint prohibition	Endpoint disable The application cannot disable control of OUT endpoint 0.	R
b29~b28	Reserved	-	The reset value must be maintained.	R/W
b27	SNAK	Set the NAK bit	Set the NAK bit (Set NAK) Writing to this bit will set the NAK bit of the endpoint. With this bit, the application can control the sending of NAK handshake signals on the endpoint. The module can also set this bit of the endpoint to 1 after the endpoint receives the SETUP packet.	R/W
b26	CNAK	Clear the NAK bit	Clear the NAK bit (Clear NAK) Writing to this bit will clear the NAK bit of the endpoint.	R/W
b25~b22	Reserved	-	The reset value must be maintained.	R/W
b21	STALL	STALL handshake	STALL handshake (STALL handshake) When this endpoint receives a SETUP token, the application can only set this bit, and the module will clear it. If the NAK bit, the global OUT NAK, and this bit are set at the same time, the STALL bit takes precedence. Regardless of how this bit is set, the module always responds to SETUP packets with an ACK handshake.	R/W
b20	SNPM	Monitor mode	Snoop mode This bit is used to configure the endpoint into listen mode. In listen mode, the module does not check the OUT packet for correctness before transferring it to the application memory.	R/W
b19~b18	EPTYP	Endpoint type	Endpoint type The hardware is set to '00', indicating a control type endpoint.	R/W
b17	NAKSTS	NAK status	NAK status Indicates the following results: 0: Module replies to non-NAK handshake according to FIFO status. 1: The module replies with a NAK handshake on this endpoint. When the application or module sets this bit, the module stops receiving data even if there is room in the RxFIFO to continue receiving packets. Regardless of how this bit is set, the module always responds to SETUP packets with an ACK handshake.	R
b16	Reserved	-	The reset value must be maintained.	R/W
b15	USBAEP	USB active endpoint	USB active endpoint This bit is always set to 1, indicating that Control Endpoint 0 is always active in all configurations and interfaces.	R
b14~b2	Reserved	-	The reset value must be maintained.	R/W
b1~b0	MPSIZ	Maximum packet size	Maximum packet size The maximum packet size for Control OUT Endpoint 0 is the same value programmed in Control IN Endpoint 0. 00: 64 bytes 01: 32 bytes	R/W

10: 16 bytes
11: 8 bytes

33.7.4.17 USBFS Device OUT Endpoint x Control Register (USBFS_DOEPCTLx) (x=1..5)

USBFS Device OUT endpoint x control register

Offset address: 0xB00 + (endpoint number × 0x20)

Reset value: 0x0000 0000

Applications use this register to control the behavior of individual logical endpoints (except Endpoint 0).

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EPEN A	EPDI S	SOD DFRM / SD1P ID	SD0P ID/ SEV NFRM	SNA K	CNA K	Reserved				STAL L	SNP M	EPTYP [1:0]		NAK STS	EON UM/ DPID
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
USB AEP	Reserved					MPSIZ [10:0]									

Bit	Flag	Bit Name	Function	Read and write
b31	EPENA	Endpoint Enable	Endpoint enable Set by software, cleared by USBFS 0: endpoint disable 1: Endpoint enable	R/W
b30	EPDIS	Endpoint disable	Endpoint disable An application can set this bit to stop data sending/receiving on an endpoint even before the transfer is complete on that endpoint. The application must wait for an endpoint disable interrupt to occur before the endpoint is considered disabled. The module will clear this bit before the endpoint disable interrupt bit is set. An application can set this bit only if the Endpoint Enable bit for that endpoint is set.	R/W
b29	SD1PID/ SODDFRM	Set DATA1 PID/ set odd frames.	Set DATA1 PID (Set DATA1 PID) Applies to interrupt/bulk OUT endpoints only. Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA1. SODDFRM: Set odd frame (Set odd frame) Applies only to synchronous OUT endpoints. Writing to this field sets the Even/odd-numbered frame (EONUM) field to odd-numbered frames.	R/W
b28	SD0PID/ SEVNFRM	Set DATA0 PID/ SEVNFRM	Set DATA0 PID (Set DATA0 PID) Applies to interrupt/bulk OUT endpoints only. Writing to this field sets the endpoint data PID (DPID) field in this register to DATA0. SEVNFRM: Set even frame (Set even frame) Applies only to synchronous OUT endpoints. Writing to this field sets the Even/odd-numbered frames (EONUM) field to even frames.	R/W
b27	SNAK	Set the NAK bit	Set the NAK bit (Set NAK) Writing to this bit will set the NAK bit of the endpoint. With this bit, the application can control the sending of NAK handshake signals on the endpoint. The module can also set this bit of the OUT endpoint to 1 when a transfer complete interrupt occurs or after SETUP is received on the endpoint.	R/W
b26	CNAK	Clear the NAK bit	Clear the NAK bit (Clear NAK) Writing to this bit will clear the NAK bit of the endpoint.	R/W
b25~b22	Reserved	-	The reset value must be maintained.	R/W
b21	STALL	STALL handshake	STALL handshake (STALL handshake) When this endpoint receives a SETUP token, the application can only set this bit, and the module will clear it. If the NAK bit, the global OUT NAK, and this bit are set at the same time, the STALL bit takes precedence. Only the application can clear this bit, not the module.	R/W
b20	SNPM	Monitor Mode	Snoop mode This bit is used to configure the endpoint into listen mode. In the monitor mode, the module will no longer check the correctness of the received data.	R/W
b19~b18	EPTYP	Endpoint Type	Endpoint type The following are the transport types supported by this logical endpoint. 00: control	R/W

			01: synchronous 10: Batch 11: interrupt	
b17	NAKSTS	NAK status	NAK status Indicates the following results: 0: Module replies to non-NAK handshake according to FIFO status. 1: The module replies with a NAK handshake on this endpoint. When an application or module sets this bit: Even if there is room in the RxFIFO to hold the incoming packet, the module stops receiving any data on the OUT endpoint. Regardless of how this bit is set, the module always responds to SETUP packets with an ACK handshake.	R
b16	EONUM/ DPID	Even/odd-numbered frames/ Endpoint data PID	Even / odd frame Applies to synchronous OUT endpoints only. Indicates the number of the frame in which the module sends/receives synchronized data for this endpoint. The application must program the even/odd-numbered frame number through the SEVNFRM and SODDFRM fields in this register so that this endpoint transmits/receives synchronously data. 0: even frame 1: odd-numbered frames DPID: Endpoint data PID (Endpoint data PID) Applies to interrupt/bulk OUT endpoints only. Contains the PID of packets that will be received or sent on this endpoint. After an endpoint is activated, the application must program the PID of the first packet to be received or sent on this endpoint. The application programs the DATA0 or DATA1 PID using the SD0PID register field. 0: DATA0 1: DATA1	R
b15	USBAEP	USB active endpoint	USB active endpoint Indicates whether this endpoint is active in the current configuration and interface. The module clears this bit to 0 for all endpoints except endpoint 0 when a USB reset is detected. After receiving the SetConfiguration and SetInterface commands, the application must program the endpoint register accordingly and set this bit to 1.	R/W
b14~b11	Reserved	-	The reset value must be maintained.	R/W
b10~b0	MPSIZ	Maximum packet size	Maximum packet size Applications must program this field to the maximum packet size for the current logical endpoint. This value is in bytes.	R/W

33.7.4.18 USBFS Device OUT Endpoint x Interrupt Register (USBFS_DOEPINTx) (x=0..5)

USBFS Device OUT endpoint x interrupt register

Offset address: 0xb08 + (endpoint number × 0x20)

Reset value: 0x0000 0080

This register indicates the status of the endpoint on USB and AHB related events. The application must read this register when the OUT endpoint interrupt bit (OEPINT bit in USBFS_GINTSTS) in the USBFS_GINTSTS register is set. Before the application can read this register, it must first read the USBFS_DAINR register to get the exact endpoint number of the USBFS_DOEPINTx registers. The application must clear the corresponding bits in this register before clearing the corresponding bits in the USBFS_DAINR and USBFS_GINTSTS registers.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved									B2BSTUP	Reserved	OEPDIS	STUP	Reserved	EPDISD	XFRC
Bit	Flag	Bit Name	Function											Read and write	
b31~b7	Reserved	-	The reset value must be maintained.											R/W	
b6	B2BSTUP	Received consecutive SETUP packets	Received consecutive SETUP packets (Back-to-back SETUP packets received) Applies to control OUT endpoints only. This bit indicates that this endpoint has received more than three consecutive SETUP packets. Software can also write 1 to clear this bit.											R/W	
b5	Reserved	-	The reset value must be maintained.											R/W	
b4	OEPDIS	An OUT token was received when the endpoint prohibited	OUT token received when endpoint disabled Applies to control OUT endpoints only. Indicates that an OUT token was received when the endpoint was not yet enabled, generating an interrupt. Write 1 to clear by software.											R/W	
b3	STUP	SETUP stage completed	SETUP phase done (SETUP phase done) Applies to control OUT endpoints only. Indicates that the SETUP phase of the control endpoint is complete, and no consecutive SETUP packets are received in the current control transfer. On this interrupt, the application can decode the received SETUP packet. Write 1 to clear by software.											R/W	
b2	Reserved	-	The reset value must be maintained.											R/W	
b1	EPDISD	Endpoint disable interrupt	Endpoint disable interrupt This bit indicates that the endpoint has been disabled by the application. Write 1 to clear by software.											R/W	
b0	XFRC	Transfer complete interrupt	Transfer completed interrupt This field indicates that transfers set on this endpoint have completed transfers on USB and AHB. Write 1 to clear by software.											R/W	

33.7.4.19 USBFS Device OUT Endpoint 0 Transfer Size Register (USBFS_DOEPTSIZ0)

USBFS Device OUT endpoint 0 transfer size register

Offset address: 0xB10

Reset value: 0x0000 0000

The application must modify this register before enabling Endpoint 0. After enabling endpoint 0 through the endpoint enable bit (EPENA in USBFS_DIEPCTL0) in the device control endpoint 0 control register, the module modifies this register. The application can read this register only after the module has cleared the endpoint enable bit.

Non-zero endpoints use the registers of endpoints 1~15.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved	STUPCNT [1:0]	Reserved										PKTCNT	Reserved		
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved									XFRSIZ [6:0]						

Bit	Flag	Bit Name	Function	Read and Write
b31	Reserved	-	The reset value must be maintained.	R/W
b30~b29	STUPCNT	SETUP packet count	SETUP packet count This field specifies the number of SETUP packets the endpoint can receive consecutively. 01: 1 packet 10: 2 packet 11: 3 packet	R/W
b28~b20	Reserved	-	The reset value must be maintained.	R/W
b19	PKTCNT	Packet count	Packet count The number of packets that should be received in one transfer. Before the endpoint is enabled, the software sets this bit. After the transmission starts, every time a data packet is received, the value of this field is automatically reduced.	R/W
b18~b7	Reserved	-	The reset value must be maintained.	R/W
b6~b0	XFRSIZ	Transfer size	Transfer size Indicates the amount of data contained in one data transmission of endpoint 0, in bytes. The module interrupts the application only after the application has transferred this data. The transfer size can be set to the endpoint's maximum packet size to break at the end of each packet. The module decrements this field each time a packet is read from the RxFIFO and written to external memory.	R/W

33.7.4.20 USBFS Device OUT Endpoint x Transfer Size Register (USBFS_DOEPTSIZx) (x=1..5)

USBFS Device OUT endpoint x transfer size register

Offset address: 0xB10 + (endpoint number × 0x20)

Reset value: 0x0000 0000

The application must modify this register before enabling the endpoint. After enabling the endpoint through the endpoint enable bit (EPENA bit in USBFS_DOEPCTLx) in the USBFS_DOEPCTLx register, the module modifies this register. The application can read this register only after the module has cleared the endpoint enable bit.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Res	RXDPID [1:0]/ STUPCNT [1:0]		PKTCNT [9:0]									XFRSIZ [18:16]			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
XFRSIZ [15:0]															

Bit	Flag	Bit Name	Function	Read and write
b31	Reserved	-	The reset value must be maintained.	R/W
b30~b29	RXDPID/ STUPCNT	Received data PID/SETUP packet count	Received data PID Applies to synchronous OUT endpoints only. This is the PID of the last packet received by this endpoint. 00: DATA0 01: DATA2 10: DATA1 11: MDATA STUPCNT : SETUP packet count Applies to control OUT endpoints only. This field specifies the number of SETUP packets the endpoint can receive consecutively. 01: 1 packets 10: 2 packets 11: 3 packets	R/W
b28~b19	PKTCNT	Packet count	Packet count Indicates the number of data packets contained in a data transmission on this endpoint. This field will be decremented after every packet (max size or short) written to the RxFIFO.	R/W
b18~b0	XFRSIZ	Transfer size	Transfer size This field contains the amount of data contained in a data transmission of the current endpoint, in bytes. The module interrupts the application only after the application has transferred this data. The transfer size can be set to the endpoint's maximum packet size to break at the end of each packet. The module decrements this field each time a packet is read from the RxFIFO and written to external memory.	R/W

33.7.4.21 USBFS Device OUT Endpoint x DMA Address Register (USBFS_DOEPDMAx)(x=0..5)

USBFS Device OUT endpoint x transfer size register

Offset address: 0xB14 + (endpoint number × 0x20)

Reset value: 0XXXXX XXXX

This register is used to set the DMA address in device endpoint DMA mode.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
DMAADDR [31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DMAADDR [15:0]															
Bit	Flag	Bit Name	Function												Read and write
b31~b0	DMAADDR	DMA address	DMA address (DMA address) This bit contains the start address of the external memory area when using DMA for data transfer on the endpoint. Note: For control endpoints, the storage area pointed to by this field is also used to store control OUT packets and SETUP transaction packets. When more than three SETUP packets are received consecutively, the SETUP packets in the memory will be overwritten. This register is incremented for every AHB transfer. The application must set a double word aligned address.												R/W

33.7.5 USBFS Clock Gating Control Register

Control HCLK and PHY clock by gating the clock control register to reduce power consumption. Bit values in register descriptions are in binary unless otherwise noted.

33.7.5.1 USBFS Clock Gating Control Register (USBFS_GCCTL)

Offset address: 0xE00

Reset value: 0x0000 0000

This register is available in both host and device modes.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								SUSP	PHYSLEEP	ENL1GTG	Reserved			GATEHCLK	STPPCLK
Bit	Flag	Bit Name				Function									Read and write
b31~b8	Reserved	-				The reset value must be maintained.									R/W
b7	SUSP	Deep Sleep status bit				Deep Sleep (L1 suspended)									R
b6	PHYSLEEP	PHY SLEEP mode status bit				PHY SLEEP mode status bit (PHY in sleep) This bit is used to indicate that the PHY is in SLEEP mode.									R
b5	ENL1GTG	L1 clock gating				L1 clock gating (Enable Sleep Clock Gating) This bit is used for clock gating control in L1 mode.									R/W
b4~b2	Reserved	-				The reset value must be maintained.									R/W
b1	GATEHCLK	Gated HCLK				Gate HCLK (Gate HCLK) When the USB communication is suspended or the session is invalid, the application program will set this bit to stop clocking the modules except the AHB bus slave interface, master interface and wake-up logic. The application clears this bit when USB communication resumes or a new session starts.									R/W
b0	STPPCLK	stop phy clock				Stop PHY clock (Stop PHY clock) The application sets this bit to stop the PH clock when USB communication is suspended, the session is invalidated, or the device is disconnected. The application program clears this bit when USB communication resumes.									R/W

34 CAN FD Controller (CAN FD)

34.1 Introduction

CAN (Controller Area Network) bus is a bus standard that can realize mutual communication between microprocessors or devices without a host.

CAN FD controller complies with the CAN bus CAN2.0 (2.0A , CAN2.0B) and CAN FD protocol.

The CAN bus controller can process the data transmission and reception on the bus. In this product, the CAN FD controller has 16 groups of filters. The acceptance filters are used to select messages for an application to receive.

The application program in the CAN FD bus controller sends the transmission data to the bus through a high-priority Primary Transmit Buffer (PTB) and 3 Secondary Transmit Buffers (STB). The sending scheduler determines the mailbox sending order. The bus data is obtained through 8 receive buffers (Receive Buffer, hereinafter referred to as RB). 3 STBs and 8 RBs can be understood as a 3-level FIFO and a 8-level FIFO, and the FIFO is completely controlled by hardware.

The CAN FD bus controller can also support time-triggered CAN communication (Time-trigger communication).

CAN FD main features:

- Fully supports CAN2.0A/CAN2.0B/CAN FD protocols.
- CAN2.0 supports a maximum communication baud rate of 1Mbit/s
- Support 1~1/256 baud rate prescaler, flexible configuration of baud rate.
- 8 receive buffers
 - FIFO mode
 - Errors or unreceived data will not overwrite stored messages
- 1 high priority main transmit buffer PTB
- 3 secondary transmit buffers STB
 - FIFO mode
 - priority arbitration
- 16 sets of independent filters
 - Support 11-bit standard ID and 29-bit extended ID
 - Programmable ID CODE bits and MASK bits
- Both PTB/STB support single send mode
- Support silent mode
- Support loopback mode
- Supports capturing transmission error types and locating arbitration failure locations
- Programmable error warning value
- Support ISO11898-4 specified time trigger CAN and receive timestamp

34.2 CAN FD System Block Diagram

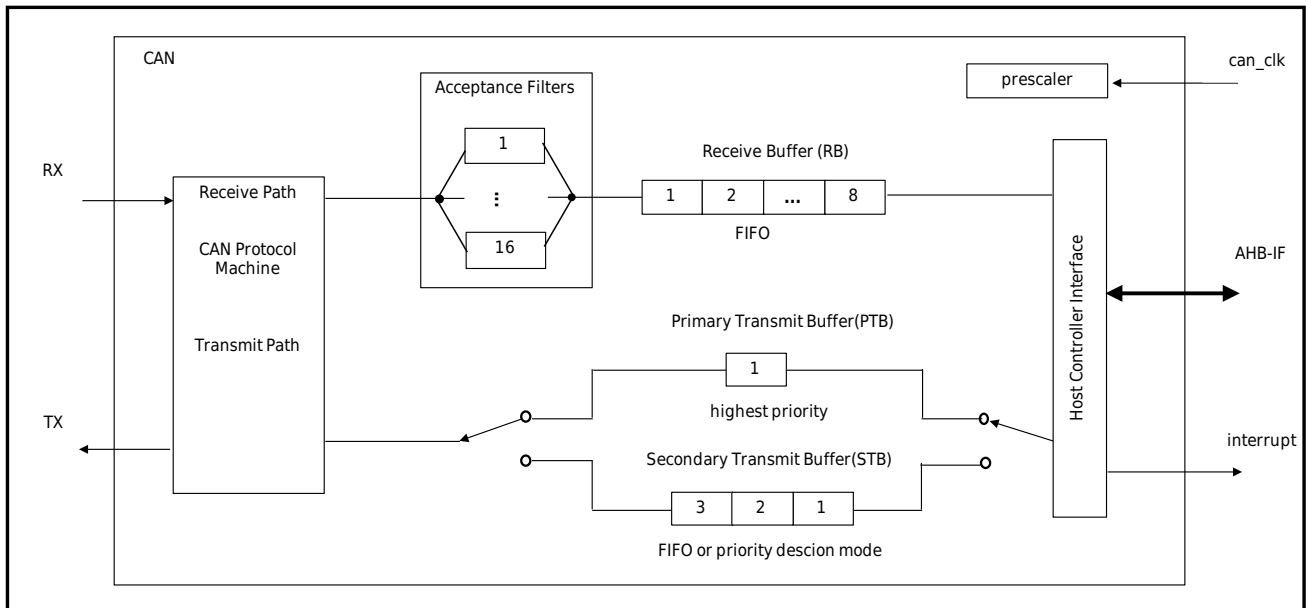


Figure 34-1 CANFD System Block Diagram

34.3 Pin Description

Table 34-1 CAN Pin Description

Pin Name	Direction	Functional Description
CANx_RX(x=1~3)	Input	CAN receive data signal
CANx_TX(x=1~3)	Output	CAN sends data signal
CANx_TST_SAMPLE(x=1~3)	Output	For observation only, sampling position (communication clock width of one period after the sampling point)
CANx_TST_CLOCK(x=1~3)	Output	For observation only, baud rate (the width of the communication clock before a bit starts in one period)

34.4 Functional Description

34.4.1 Action Mode

using the CAN FD function, you need to enable the CAN FD function first. For details, see the description of the CAN global control register in the register section. The CAN FD controller has two operating modes, reset mode (CAN_CFG_STAT.RESET=1) and action mode (CAN_CFG_STAT.RESET=0). When initializing the module, you should first set the registers that can only operate in reset mode in reset mode (see the register description chapter for details). Then exit reset mode and operate the remaining registers in action mode.

34.4.2 Baud Rate Setting

The CAN communication clock (can_clk) is used to generate the baud rate. The system clock division or independent clock can be selected (see the "Clock Controller (CMU)" chapter "Working Clock Specification"), which is specified by the register CMU_CANCKCFGR. When selecting the clock, the setting condition that PCLK0 (CAN control logic clock) is 1.5 times or more of the CAN communication clock must be met.

The figure below shows the CAN bit time definition diagram. The part on the dotted line is the bit time specified by the CAN protocol, and the part below the dotted line is the bit time defined by the CAN controller CAN-CTRL. Segment1 and segment2 can be set through registers SBT and FBT. SBT register and FBT register can only be set when CAN_CFG_STAT.RESET=1, that is, when the CAN software is reset. SBT register is used for the arbitration segment of CAN2.0 and CAN FD, and the FBT register is used for the CAN FD data segment.

During FD communication, it is recommended that the communication clock be 20MHz/40MHz/80MHz.

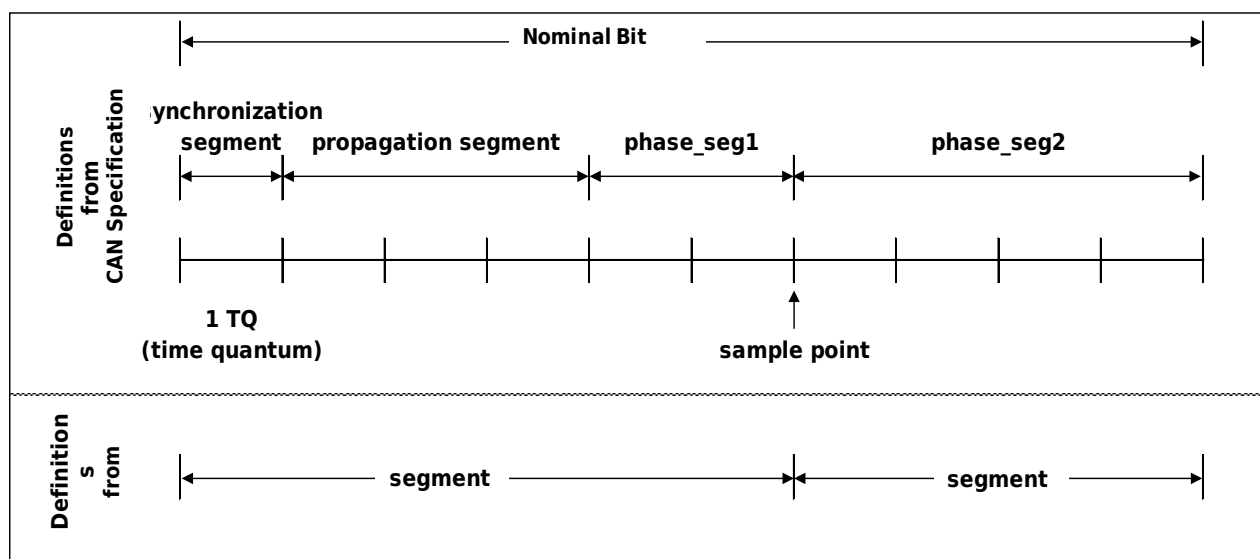


Figure 34-2 CAN Bit Time Definition Diagram

TQ calculation, where PRESC is set by the S_PRESC bit in the SBT register or the F_PRESC bit in the FBT register. f_{can_clk} It is CAN communication clock frequency.

Please refer to the following formula for the bit time calculation method, where S_segment1 and S_segment2 are set by the S_SEG_1 and S_SEG_2 bits of the SBT register, and F_segment1 and F_segment2 are set by the F_SEG_1 and F_SEG_2 bits of the FBT register.

Table 34-2 CAN Bit Time Setting Rules

Bit	Predetermined area	Rule
S_SEG_1 bit of SBT register	[0..63] CAN2.0 bits (slow) [0..63] CAN FD nominal bits (slow)	
S_SEG_2 bit of SBT register	[0..7] CAN2.0 bits (slow) [0..31] CAN FD nominal bits (slow)	
S_SJW bit of the SBT register	[0..15] CAN2.0 bits (slow) [0..15] CAN FD nominal bits (slow)	
F_SEG_1 bit of FBT register	[0..15] CAN FD data bits (fast)	
F_SEG_2 bit of FBT register	[0..7] CAN FD data bits (fast)	
F_SJW bit of the FBT register	[0..7] CAN FD data bits (fast)	

The following is a recommendation for CANFD baud rate settings for reference only.

PSP : Primary Sample Point

SSP : Secondary Sample Point

Seg 1 : Segment 1

Seg 2 : Segment 2

TDC : Transmitter Delay Compensation

Table 34-3 Baud Rate Setting Suggestions for 20MHz Communication Clock

Bit Rate [Mbit/s]	PSP [%]	SSP [%]	Prescaler	Bit Time [TQ]	Seg 1 [TQ]	Seg 2 [TQ]	SJW [TQ]	TDC [CAN Communication Clock]
0.25 (arbitration)	80	-	1	80	64	16	16	-
0.5 (arbitration)	80	-	1	40	32	8	8	-
0.5	80	Disable	1	40	32	8	8	-
1	80	80	1	20	16	4	4	16
2	80	80	1	10	8	2	2	8
4	80	80	1	5	4	1	1	4
5	75	75	1	4	3	1	1	3

Table 34-4 40MHzBaud Rate Setting Suggestions For 20mhz Communication Clock

Bit Rate [Mbit/s]	PSP [%]	SSP [%]	Prescaler	Bit Time [TQ]	Seg 1 [TQ]	Seg 2 [TQ]	SJW [TQ]	TDC [CAN Communication Clock]
0.25 (arbitration)	80	-	2	80	64	16	16	-
0.5 (arbitration)	80	-	1	80	64	16	16	-
0.5	80	Disable	2	40	32	8	8	-
1	80	80	1	40	32	8	8	32
2	80	80	1	20	16	4	4	16
4	80	80	1	10	8	2	2	8
5	75	75	1	8	6	2	2	6
8	80	80	1	5	4	1	1	4

Table 34-5 40MHzBaud Rate Setting Suggestions For 20mhz Communication Clock

Bit Rate [Mbit/s]	PSP [%]	SSP [%]	Prescaler	Bit Time [TQ]	Seg 1 [TQ]	Seg 2 [TQ]	SJW [TQ]	TDC [CAN Communication Clock]
0.25 (arbitration)	80	-	4	80	64	16	16	-
0.5 (arbitration)	80	-	2	80	64	16	16	-
0.5	80	Disable	4	40	32	8	8	-
1	80	80	2	40	32	8	8	64
2	80	80	2	20	16	4	4	32
4	80	80	1	20	16	4	4	16
5	75	75	1	16	12	4	4	12
8	80	80	1	10	8	2	2	8

34.4.3 Transmit Buffer

CAN_CTRL provides two sending buffers for sending data, the main sending data buffer PTB and the secondary sending buffer STB. PTB has the highest priority, but only one frame of data can be buffered. STB has a lower priority than PTB, but it can buffer 3 frames of data, and 3 frames of data in STB can work in FIFO mode or priority arbitration mode.

The 3 frames of data in the STB can be sent by setting the TSALL bit of the TCMD register to 1. In the FIFO mode, the first written data is sent first. In the priority mode, the data with the smaller ID is sent first.

Data in a PTB has the highest priority, so a PTB send can postpone an STB send, but an STB that has already won arbitration and started sending cannot be delayed by a PTB send.

PTB and STB can be accessed through the TBUF register. Select PTB or STB through the TBSEL bit of the TCMD register, TBSEL=0, select PTB, TBSEL=1, select STB. The next SLOT in the STB is selected by the TSNEXT bit of the TCTRL register. The corresponding relationship is shown in the figure below:

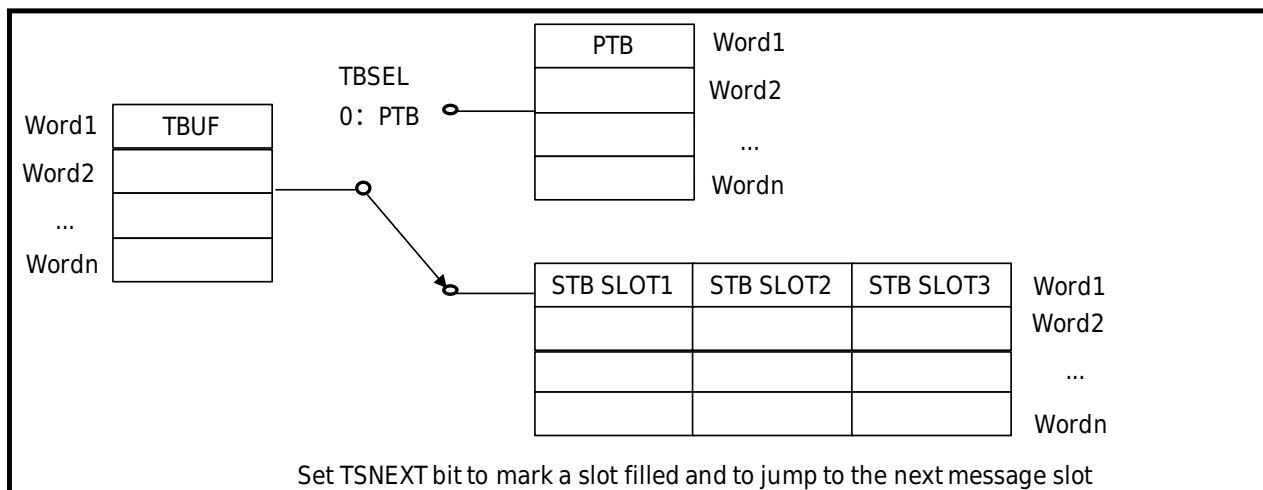


Figure 34-3 CANFD TBUF Register Write Transmit Buffer and Schematic Diagram

34.4.4 Receive Buffer

CAN_CTRL provides 8 SLOT receive buffers for storing received data, and the 8 SLOT receive buffers work in FIFO mode. RB SLOT reads the received data through the RBUF register, always reading the earliest received data first, and releases the RB SLOT that has been read by setting RREL of the RCTRL register to 1, and points to the next RB SLOT.

The schematic diagram of reading RB SLOT through RBUF is as follows.

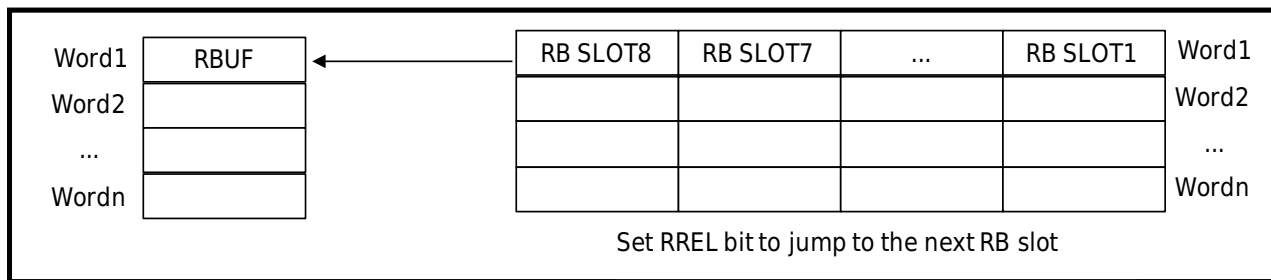


Figure-34-4 Schematic Diagram of CAN RBUF Register Read Receive Buffer

34.4.5 Receive Filter Register Set

CAN_CTRL provides 16 sets of 32-bit filters to filter the received data to reduce CPU load. The filters can support standard format 11-bit ID or extended format 29-bit ID. Each set of filters has a 32-bit ID CODE register and a 32-bit ID MASK register. The ID CODE register is used to compare the received CAN ID, and the ID MASK register is used to select the CAN ID bit for comparison. When the corresponding ID MASK bit is 1, the ID CODE of this bit is not compared.

The received data will be received as long as it passes any of the 16 sets of filters, and the received data will be stored in RB, otherwise the data will not be received or stored.

Each group of filters is enabled or disabled through the ACFEN register. ID CODE and ID MASK are set by the SELMASK bit of the ACFCTRL register. When SELMASK=0, it points to ID CODE, and when SELMASK=1, it points to ID MASK. The filter is selected by the ACFADR bit of the ACFCTRL register. ID CODE and ID MASK are accessed through the ACF register and can only be set when CFG_STAT.RESET=1, that is, CAN software reset. Please refer to the figure below for the method of ACF registering and accessing the filter register group.

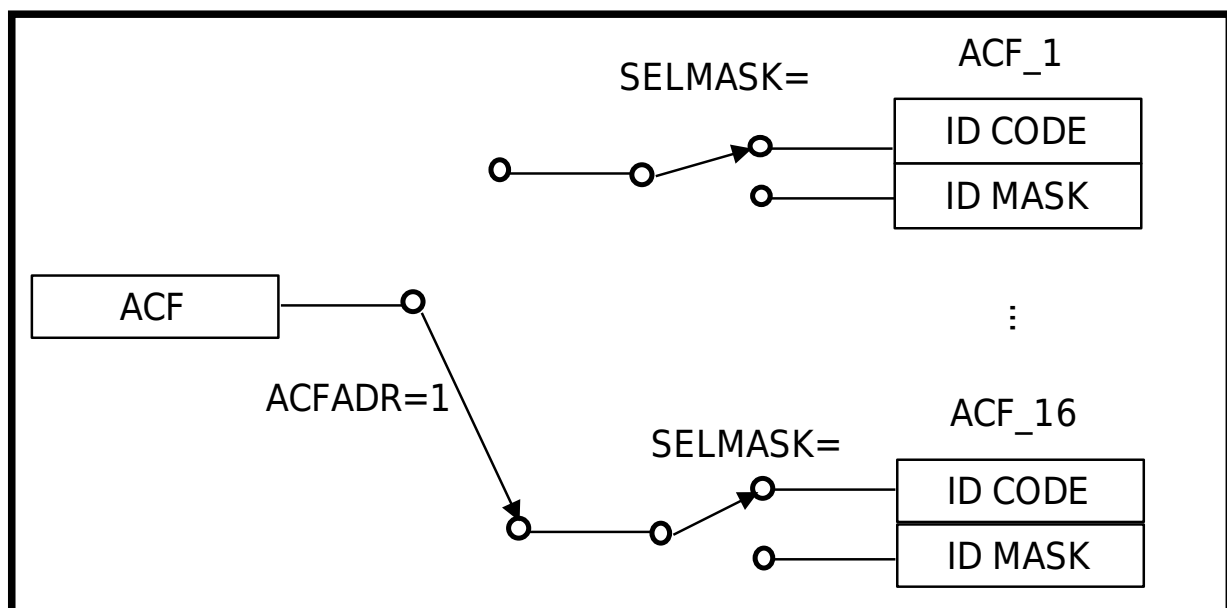


Figure 34-5 Schematic Diagram of CAN ACF Register Access Filter Group

34.4.6 Data transmission

Before starting to send, you must ensure that the data to be sent by the PTB or STB has been filled, and then start the PTB or STB to send. Data is not allowed to be filled again during sending.

The sending data setting steps are as follows:

1. Set TBSEL to select sending BUF from PTB and STB
2. Write the data to be sent through the TBUF register
3. If STB is selected, set TSNEXT=1 to complete the loading of all STB SLOTS
4. Send enable
 - PTB sent using TPE
 - STB sends using TSALL or TSONE
5. send complete status acknowledgment
 - PTB transmission is completed using TPIF, TPIE is used to enable TPIF
 - STB uses TSONE to send TSIF when it is completed, and TSIE is used to enable TSIF
 - STB uses TSIF when TSALL transmission is completed. At this time, after all the STB SLOT data that needs to be set are sent, TSIF is set, and TSIE is used to enable TSIF

34.4.7 Single data transmission

When the automatic resend function is not required, it can be set to single-send mode through the register. The TPSS bit of the CFG_STAT register is used for the single-send mode setting of PTB, and the TSSS bit is used for the single-send mode of STB. When the data is successfully sent, the action of single sending and normal sending mode is the same. But when the data is not sent successfully, the following results will appear:

- When TPIF is set (TPIE=1), the corresponding BUF SLOT data will be cleared.
- When an error is sent, KOER is updated and BEIF is set (BEIE=1).
- Arbitration fails, ALIF is set (ALIE=1).

In the single sending mode, TPIF cannot be used alone to judge whether the sending is completed. It needs to judge whether the sending is completed together with BEIF and ALIF.

34.4.8 Cancel Data Sending

Data transmissions that have been requested but not yet executed can be canceled via TPA or TSA. Cancellation of data transmission will occur in the following situations:

- in arbitration
 - If the node arbitration fails, the data transmission is canceled.
 - If the node arbitration is successful, it will continue to send.
- data sending
 - If data is successfully sent and ACK is received, the corresponding flag and status are normally set. Data transmission is not canceled.

- If the data is sent successfully but no ACK is received, the data sending is canceled and the error counter is incremented.
- The sending data set by TSALL=1, the STB SLOT data that is being sent are sent normally, and the STB SLOT that has not started sending is canceled.

The result of canceling data sending has the following two situations:

- TPA releases PTB and makes TPE=0.
- The TSA releases one STB SLOT or all STB SLOTS depending on whether TSONE or TSALL enabled transmission.

34.4.9 Data Reception

The receive filter group can filter out unnecessary received data, reduce the occurrence of interrupts and the reading of RB, thereby reducing the CPU load. The receiving data setting steps are as follows:

1. Set filter groups.
2. Set RFIE, RAFIE and AFWL.
3. Waiting for RFIF or RAFIF
4. Read the earliest received data from RB FIFO via RBUF.
5. Set RREL=1 to select the next RB SLOT.
6. Repeat 4, 5 until the RB is confirmed to be empty by RSTAT.

34.4.10 Error handling

On the one hand, CAN_CTRL can automatically handle some errors, such as automatically resending data or discarding received frames containing errors, and on the other hand, reports errors to the CPU through interrupts.

CAN nodes have the following three error states:

- Error Active: The node automatically sends an active error flag when it detects an error.
- Error Passive: A passive error flag is automatically sent when a node detects an error.
- Node closed: In the closed state, this node no longer affects the entire CAN network.

CAN_CTRL provides TECNT and RECNT two counters for counting errors. TECNT and RECNT counters increase or decrease according to the rules stipulated in the CAN agreement. In addition, a programmable CAN error warning LIMIT register is used to generate an error interrupt to notify the CPU.

There are the following five error types during CAN communication, and the error types can be identified by the KOER bit of the EALCAP register.

- Bit error
- Wrong form

- Filling error
- Wrong answer
- CRC error

34.4.11 Node Down

When the number of sending errors is greater than 255, the CAN node automatically enters the node shutdown state and does not participate in CAN communication until it returns to the error active state. The CAN node shutdown status can be confirmed by the BUSOFF bit of the CFG_STAT register. BUSOFF is set and EIF interrupt is generated at the same time.

There are two methods for CAN to recover from the node off state to the error active state:

- Power-on Reset
- Received 128 consecutive 11-bit stealth bit sequences (recovery sequence)

In the node shutdown state, the TECNT value remains unchanged, and the RECNT is used for the count recovery sequence. After recovering from node shutdown, TECNT and RECNT are reset to 0.

34.4.12 Lost Arbitration Position Capture

CAN_CTRL can accurately capture the position of the arbitration failure bit and reflect it in the ALC register. The ALC register stores the position of the last arbitration failure bit. If the node wins the arbitration, the ALC bit is not updated.

The ALC value is defined as follows:

After the SOF bit, the first ID data bit ALC is 0, the second ID data bit ALC is 1, and so on. Since arbitration only takes place within the arbitration field, the maximum value of ALC is 31. For example, a standard format remote frame arbitrates with an extended frame, if the extended frame fails at the IDE bit, then ALC=12.

34.4.13 Loop Mode

CAN_CTRL supports the following two loopback modes:

- Internal loopback
- External loopback

Both loopback modes can receive data frames sent by themselves, which are mainly used for testing purposes.

In the internal loopback mode, the module internally connects the receiving data line to the sending data line, and the sending data is not output. In internal loopback mode, nodes generate self-acknowledgment signals to avoid ACK errors.

The external loopback mode maintains the connection with the transceiver so that the data sent can still appear on the CAN bus. With the help of the transceiver, CAN can receive the data sent by

itself. In the external loopback mode, the SACK bit of the RCTRL register can be used to determine whether to generate a self-acknowledgment signal. When SACK=0, no self-acknowledge signal is generated. When SACK=1, a self-acknowledge signal is generated.

In external loopback mode, when SACK=0, the following two situations will occur:

- Other nodes also receive the data frame sent by the node and send a response signal. In this case, the node can successfully send and receive data.
- If no other node returns an acknowledgment signal, an acknowledgment error is generated, the data is resent and the error counter is incremented. At this time, it is recommended to use the single sending mode.

When returning to normal mode from loopback mode, in addition to clearing the mode bit, a software reset of CAN_CTRL is required.

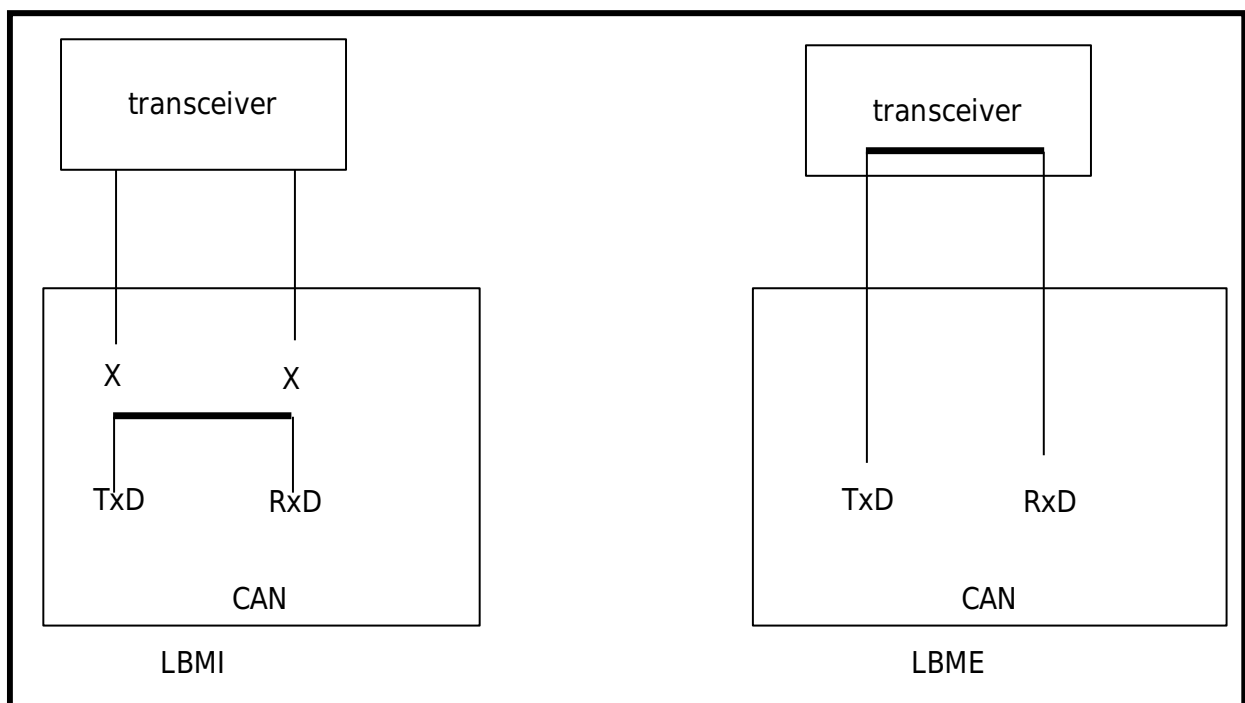


Figure 34-6 Schematic Diagram Of CANFD LBMI And LBME

34.4.14 Silent Mode

Silent mode can be used to monitor CAN network data. In silent mode, data can be received from the CAN bus without sending any data to the bus. Set the LOM in the TCMD register to 1 to make the CAN bus controller enter the silent mode, and clear it to 0 to leave the silent mode.

The external loopback mode can be combined with the silent mode to form an external loopback silent mode. At this time, CAN can be regarded as a quiet receiver, but can send data when necessary. In external loopback silent mode, frames containing self-acknowledgment signals are allowed to be sent, but the node does not generate error flags and overload frames.

34.4.15 Software Reset Function

By setting the RESET bit of the CFG_STAT register to 1, the software reset function is realized. The reset range of the software reset function is shown in the table below.

Table 34-6 Software Reset Range Table

Register Bit Name	Software Reset	Remark	Register Bit Name	Software Reset	Remark
ACFADR	No	-	F_SEG_1	Yes	Can only be written by software reset
ACODE	No	Can only be written by software reset	F_SEG_2	Yes	Can only be written by software reset
AE_x	No	-	F_SJW	Yes	Can only be written by software reset
AFWL	No	-	KOER	Yes	-
AIF	Yes	-	LBME	Yes	-
ALC	Yes	-	LBMI	Yes	-
ALIE	No	-	RACTIVE	Yes	Reception stops immediately and no ACK is generated
ALIF	Yes	-	RAFIE	No	-
AMASK	No	Can only be written by software reset	RAFIF	Yes	-
BEIE	No	-	RBALL	Yes	-
BEIF	Yes	-	RBUF	Yes	RB is flagged as empty, the value is indefinite
BUSOFF	No	Cleared by writing 1	RECNT	No	Cleared by writing 1 to BUSOFF
EIE F	No	-	REF_ID	No	-
EIF	No	-	REF_IDE	No	-
EPASS	No	-	RFIE	No	-
EPIE	No	-	RFIF	Yes	-
EPIF	Yes	-	RIE	No	-
EWARN	No	-	RIF	Yes	-
EWL	Yes	-	ROIE	No	-
FD_ISO	No	Can only be written by software reset	ROIF	Yes	-
F_FRESC	No	Can only be written by software reset	ROM	No	-

Register Bit Name	Software Reset	Remark	Register Bit Name	Software Reset	Remark
ROV	Yes	-	TSMODE	No	-
RREL	Yes	-	TSNEXT	Yes	-
RSTAT	Yes	-	TSONE	Yes	-
SACK	Yes	-	TPIE	No	-
SELMASK	No	-	TPIF	Yes	-
S_PRESC	No	Can only be written by software reset	TPSS	Yes	-

Register Bit Name	Software Reset	Remark	Register Bit Name	Software Reset	Remark
S_SEG_1	No	Can only be written by software reset	TSFF	Yes	All STB SLOTS are flagged empty
S_SEG_2	No	Can only be written by software reset	TSIE	No	-
S_SJW	No	Can only be written by software reset	TSIF	Yes	-
SSPOFF	Yes	-	TSSS	Yes	-
TACTIVE	Yes	Send immediate stop	TSSTAT	Yes	All STB SLOTS are marked empty
TBE	Yes	-	TTEN	Yes Yes	-
TBF	No	-	TTIF	Yes	-
TBPTR	No	-	TTIE	No	-
TBSEL	Yes	-	TTPTR	No	-
TBUF	Yes	STB is flagged as empty, pointing to PTB	TTTBM	No	-
TDCEN	Yes	-	TTYPE	No	-
TECNT	No	be cleared by BUSOFF=1	TT_TRIG	No	-
TEIF	Yes	-	TT_WTRIG	No	-
TPA	Yes	-	T_PRESC	No	-
TPE	Yes	-	WTIE	No	-
TSA	Yes	-	WTIF	Yes	-
TSALL	Yes	-	-	-	-

34.4.16 Upward Compatible with CAN-FD Function

When the CAN FD function is disabled, even if CAN FD frames are received in a CAN FD network , the receiver will automatically ignore these frames and will not return ACK. It will wait until the bus is idle before sending or receiving the next CAN2.0B frame.

34.4.17 Time-triggered TTCAN

CAN-CTRL provides partial (level 1) hardware support for the time-triggered communication method specified in ISO11898-4. This chapter introduces TTCAN functions from the following 5 parts.

34.4.17.1 TBUF Behavior in TTCAN Mode

TTTBM=1

When TTTBM=1, PTB and STB SLOT form TB SLOT as well, and send BUF through the TBPTR register. When TBPTR=0, it points to PTB, TBPTR=1 points to STB SLOT1, and so on. The host can flag the sending BUF SLOT through the TBE and TBF registers. At this time, the TBSEL and TSNEXT registers have no meaning and can be ignored.

When TTTBM=1, PTB does not have any special attributes. Like STB SLOT, the transmission completion flag also uses TSIF.

In TTCAN mode, there is no FIFO mode and priority arbitration mode for sending BUF, and only one selected SLOT can send data.

In TTCAN mode, time-triggered mode is required for transmission start, TPE, TSONE, TSALL, TPSS and TPA are fixed to 0 and ignored.

TTTBM=0

When TTTBM=0, use event-driven communication and receive timestamp function in combination. In this mode, the functions of PTB and STB are the same as when TTEN=0, so PTB always has the highest priority, and STB can work in FIFO mode or priority arbitration mode.

34.4.17.2 TTCAN Function Description

After power on, Time Master needs to be initialized according to the ISO 11898-4 protocol. In a CAN network, there can be up to 8 potential Time Masters. Each Time Master has its own reference message ID (the last 3 digits of the ID). These potential Time Masters send respective reference messages according to their own priorities.

After TTEN=1, the 16-bit counter starts to work. When the reference message is successfully received or the Time Master successfully sends the reference message, the CAN controller copies Sync_Mark to Ref_Mark, and Ref_Mark sets the period time to 0. The successful reception of the reference message sets the RIF flag and the successful transmission of the reference message sets the TPIF flag or the TSIF flag. At this point the host needs to prepare the trigger conditions for the next action.

The trigger condition may be receiving a trigger. This trigger-only interrupt can be used to detect that an expected message has not been received.

The trigger condition can also be a send trigger. This trigger starts sending the data in the TBUF SLOT specified by the TTPTR register. If the selected TBUF SLOT is flagged empty, the transmission is not started, but the interrupt flag is set.

34.4.17.3 TTCAN Timing

CAN_CTRL supports ISO11898-4 level 1. 16-bit counter is included which operates with the bit times defined by S_PRESC, S_SEG_1, S_SEG_2. If TTEN=1, there is an additional prescaler T_PRESC.

When SOF of one frame of data, the value of the counter is Sync_Mark. If the frame data is a reference message, copy Sync_Mark to Ref_Mark. The period time is equal to the value of the counter minus Ref_Mark. This time is used as a time stamp for receiving messages or as a trigger time reference for sending messages.

34.4.17.4 TTCAN Trigger Mode

The trigger mode of TTCAN is defined through the TTYPE register, the TTPTR register specifies sending SLOT, and TT_TRIG specifies the period time of the trigger.

Contains the following five trigger methods:

- Trigger immediately
- Time trigger
- Single send trigger
- Send start trigger
- Send stop trigger

All triggers use the TTIF flag except the immediate trigger mode. When TTTBM=1, only time trigger mode is supported.

Trigger Immediately

By writing the high bit of TT_TRIG (don't care about the value written), the trigger is started. In this mode, the data in the TBUF SLOT selected by TTPTR will be sent immediately. TTIF is not set.

Time Trigger

The time trigger mode only generates interrupts by setting the TTIF flag, and has no other functions. Time triggering can be used if a node expects to receive expected data within a specific time window. If the TT_TRIG value is less than the actual period time, TEIF is set and there is no other action.

Single Send Trigger

The one-shot trigger method is used to send data within the execution time window. At this time, the TSSS bit is ignored.

Set the Tick of up to 16 period times specified by ISO11898-4 through the TEW bit, and the setting range is 1~16. If data transmission does not start within the specified transmit enable time window, the frame is discarded. The corresponding sending BUF SLOT is flagged as empty, and AIF is set, and the data in the corresponding sending BUF will not be rewritten, because it can be sent again by setting TPF.

If the TT_TRIG value is less than the actual period time, TEIF is set and there is no other action.

Send Start Trigger

The sending start trigger mode is used in the arbitration time window to participate in the arbitration. TSSS is used to decide whether to automatically repeat or send a single mode. If the message specified by the TTPTR register is not sent successfully, the send stop trigger can be used to stop the sending.

If the TT_TRIG value is less than the actual period time, TEIF is set and there is no other action.

Send Stop Trigger

The send stop trigger method is used to stop the send that has been started by the send start trigger method. If the sending is stopped, the sending frame is discarded, the AIF is set and the selected TBUF SLOT is flagged as empty, but the data in the TBUF SLOT will not be rewritten, and it can be sent again by setting the TPF.

If the TT_TRIG value is less than the actual period time, TEIF is set and execution stops.

34.4.17.5 TTCAN Trigger Gate Time

The TTCAN trigger watchdog function is similar to the watchdog function and is used when TTTBM=1. Used to watch the time since the last successful reference message received by the gate. The reference message can be received during the period time or after an event, and the application should set the appropriate gate time according to the specific situation.

If period count is equal to TT_WTRIG, set WTIF. Write 0 by WTIE to disable the gate trigger.

If TT_WTRIG is smaller than the actual period time, TEIF is set.

34.4.18 TDC and RDC

CAN FD communication, the data delay may exceed one bit time, so compensation is required. TDC (Transmitter Delay Compensation) and RDC (Receiver Delay Compensation) are used to compensate for data delay. TDC requires software-controlled switches, while RDC does not and is automatically effective. In TDC , the secondary sample point (SSP) is used to compensate for data delay. As shown in the following figure:

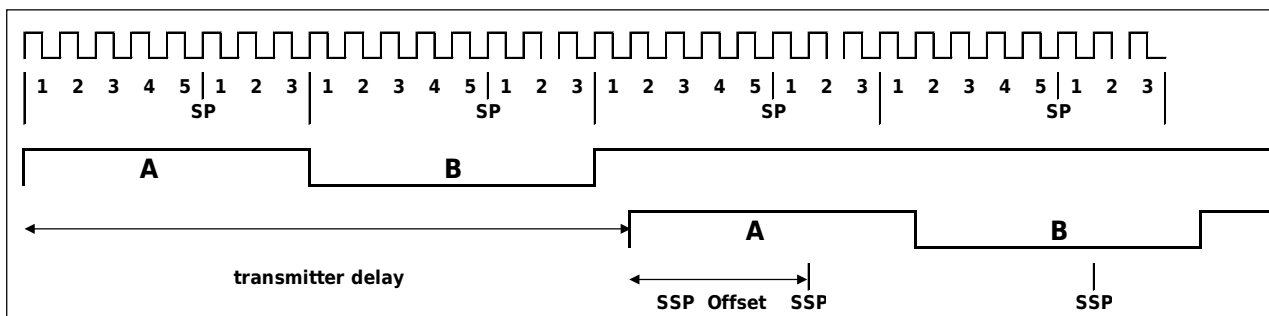


Figure 34-7 TDC Function Diagram

When the software enables the TDC function, the controller can automatically determine the transmitter delay and set the SSP Offset by setting the register TDC.SSPOFF. It is recommended that the SSP Offset setting value be the same as the $t_{F\_segment1}$ setting value.

34.4.19 Interrupt

Table 34-7 CAN Interrupt Table

Symbol	Interrupt Flag	Description
CAN_x_HOST(x=1~3)	RIF	Receiving interrupt
	ROIF	Receive overflow interrupt
	ROIF	Receive BUF full interrupt
	RAFIF	Receive BUF will be full interrupt
	TPIF	PTB send interrupt
	TSIF	STB send interrupt
	EIF	Error Interrupt
	AIF	Cancel send interrupt
	EPIE	Error passive interrupt
	ALIF	Arbitration Lost Interrupt
	BEIF	Bus error interrupt
	WTIF	Trigger gate interrupt
	TEIF	Trigger error interrupt
	TTIF	Time triggered interrupt

34.5 Register Description

Table 34-8 CAN Register List

CAN global controller address: 0x4005 5418

Register Name	Symbol	Address	Bit Width	Reset Value
CAN Global Control Register	CAN_SYCTLREG	0x4005 5418	32	0x000 0002

CAN_1_BASE_ADDR: 0x40078000

CAN_2_BASE_ADDR: 0x40071400

CAN_3_BASE_ADDR: 0x40071000

Register Name	Symbol	Offset Address	Bit Width	Reset Value
CAN receive BUF register	CAN_RBUF	0x00~0x4F	-	0xFFFF XXXX
CAN send BUF register	CAN_TBUF	0x50~0x97	-	0xFFFF XXXX
CAN Configuration and Status Register	CAN_CFG_STAT	0xA0	8	0x80
CAN command register	CAN_TCMD	0xA1	8	0x00
CAN send control register	CAN_TCTRL	0xA2	8	0x90
CAN receive control register	CAN_RCTRL	0xA3	8	0x00
CAN receive and transmit interrupt enable register	CAN_RTIE	0xA4	8	0xFE
CAN Receive and Transmit Interrupt Flag Register	CAN_RTIF	0xA5	8	0x00
CAN Error Interrupt Enable and Flags Register	CAN_ERRINT	0xA6	8	0x00
CAN Warning Limit Register	CAN_LIMIT	0xA7	8	0x1B
CAN Slow Bit Timing Register	CAN_SBT	0xA8	32	0x0102 0203
CAN Fast Bit Timing Register	CAN_FBT	0xAC	32	0x0102 0203
CAN Error and Arbitration Lost Capture Register	CAN_EALCAP	0xB0	8	0x00
CAN Transmitter Delay Compensation Register	CAN_TDC	0xB1	8	0x00
CAN receive error counter register	CAN_RECNT	0xB2	8	0x00
CAN Transmit error counter register	CAN_TECNT	0xB3	8	0x00
CAN filter group control register	CAN_ACFCTRL	0xB4	8	0x00
CAN Filter Group Enable Register	CAN_ACFEN	0xB6	8	0x01
CAN filter group code and mask registers	CAN_ACF	0xB8	32	0xFFFF XXXX
TTCAN TB slot pointer register	CAN_TB SLOT	0xBE	8	0x00
TTCAN time trigger configuration register	CAN_TTCFG	0xBF	8	0x90
TTCAN Reference message register	CAN_REF_MSG	0xC0	32	0xFFFF XXXX
TTCAN trigger configuration register	CAN_TRG_CFG	0xC4	16	0x0000
TTCAN Trigger time register	CAN_TT_TRIG	0xC6	16	0x0000
TTCAN Trigger watchdog time register	CAN_TT_WTRIG	0xC8	16	0xFFFF

34.5.1 CAN Transmit Control Register (CAN_SYCTLREG)

CAN System control Register

Address: 0x40055418

Reset value: 0x0000 0002

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved												CAN 3FDE	CAN 2FDE	CAN 1FDE	
Bit	Flag	Bit Name		Function		Read and Write									
b31~b3	Reserved	-		Must hold the reset value		R									
b2	CAN3FDE	CAN_3	FD enable bit	function	CAN_3 FD function enable bit 0: CAN_3 FD function disabled 1: CAN_3 FD function enable	R/W									
b1	CAN2FDE	CAN_2	FD enable bit	function	CAN_2 FD function enable bit 0: CAN_2 FD function disabled 1: CAN_2 FD function enable	R/W									
b0	CAN1FDE	CAN_1	FD enable bit	function	CAN_1 FD function enable bit 0: CAN_1 FD function disabled 1: CAN_1 FD function enable	R/W									

34.5.2 CAN receive BUF register (CAN_RBUF)

CAN Receive Buffer Registers

Offset address: 0x00

Reset value: 0XXXXX XXXX

The RBUF register points to the RB SLOT address of the earliest received CAN mailbox, and the RBUF register can be read in any order.

The KOER bit is the register EALCAP.KOER, which is only meaningful when RBALL=1.

The TX bit indicates that the message sent by itself is received in the loopback mode.

The CYCLE_TIME bit is only valid in TTCAN mode, indicating the period time when SOF starts.

CAN_RBUF only supports WORD access.

The data format of the CAN receiving mailbox is as follows:

Table 34-9 Standard Format CAN Receiving Mailbox Format

Address	b7	b6	b5	b4	b3	b2	b1	b0	Function
RBUF	ID [7:0]								ID
RBUF+1	-					ID [10:8]			ID
RBUF+2	-								ID
RBUF+3	-								ID
RBUF+4	IDE=0	RTR	FDF	BRS	DLC [3:0]				Control
RBUF+5	KOER [2:0]			TX	-				Status
RBUF+6	CYCLE_TIME [7:0]								TTCAN
RBUF+7	CYCLE_TIME [15:8]								TTCAN
RBUF+8	DATA1								Data
RBUF+9	DATA2								Data
RBUF+10	DATA3								Data
-	-								Data
-	-								Data
-	-								Data
RBUF+70	DATA63								Data
RBUF+71	DATA64								Data

Table 34-10 Extended Format CAN Receive Mailbox Format

Address	b7	b6	b5	b4	b3	b2	b1	b0	Function
RBUF	ID [7:0]								ID
RBUF+1	ID [15:8] ID [10:8]								ID
RBUF+2	ID [23:16]								ID
RBUF+3	-			ID [28:24]				ID	
RBUF+4	IDE=1	RTR	FDF	BRS	DLC [3:0]				Control
RBUF+5	KOER [2:0] TX			TX	-				Status
RBUF+6	CYCLE_TIME [7:0]								TTCAN
RBUF+7	CYCLE_TIME [15:8]								TTCAN
RBUF+8	DATA1								Data
RBUF+9	DATA2								Data
RBUF+10	DATA3								Data
-	-								Data
-	-								Data
-	-								Data
RBUF+70	DATA63								Data
RBUF+71	DATA64								Data

The meanings of the control bits are as follows:

IDE(Identifier Extension)

0: standard format

1: Extended format

RTR(Remote Transmission Request)

0: data frame

1 : Remote frame (only applicable to CAN2.0, fixed to 0 for CAN FD)

FDF(CAN FD Frame)

0: CAN2.0 frame

1: CAN FD frame

BRS (Bit Rate Switch)

0: The whole frame is at low baud rate

1 : Data and CRC are at fast baud rate (only applicable to CAN FD , fixed to 0 when FDF=0)

DLC(Data Length Code)

Data length code, CAN2.0 setting range is 0~8 , corresponding data length is 0Byte~8Byte , CAN FD setting range is 0~15 , corresponding data length is 0Byte~64Byte.

Table 34-11 Control Bits

DLC (binary)	Frame Type	Payload in Bytes
0000~1000	CAN2.0 and CAN FD	0~8
1001~1111	CAN2.0	8
1001	CAN FD	12
1010	CAN FD	16
1011	CAN FD	20
1100	CAN FD	24
1101	CAN FD	32
1110	CAN FD	48
1111	CAN FD	64

The status bits have the following meanings:

KOER: Same as EALCAP.KOER

TX: This bit is set to 1 when receiving data sent by itself in loopback mode.

34.5.3 CAN Transmit BUF Register (CAN_TBUF)

CAN Transmit Buffer Registers

Offset address: 0x50

Reset value: 0XXXXX XXXX

The TBUF register points to the next empty CAN sending BUF SLOT, and the TBUF register can be read in any order. Write 1 to TSNEXT by software to flag that the corresponding TBUF SLOT has written data, thus pointing to the next TBUF SLOT.

TBUF can only be accessed by WORD.

The data format of the CAN sending mailbox is as follows:

Table 34-12 Extended Format CAN Sending Email Format

Address	b7	b6	b5	b4	b3	b2	b1	b0	Function
TBUF	ID [7:0]								ID
TBUF+1	-					ID [10:8]			ID
TBUF+2	-								ID
TBUF+3	-								ID
TBUF+4	IDE=0	RTR	FDF	BRS	DLC [3:0]				Control
TBUF+5	-								-
TBUF+6	-								-
TBUF+7	-								-
TBUF+8	DATA1								Data
TBUF+9	DATA2								Data
TBUF+10	DATA3								Data
-	-								Data
-	-								Data
-	-								Data
TBUF+70	DATA63								Data
TBUF+71	DATA64								Data

Table 34-13 Extended Format CAN Send Mailbox Format

Address	b7	b6	b5	b4	b3	b2	b1	b0	Function
TBUF	ID [7:0]								ID
TBUF+1	ID [15:8]								ID
TBUF+2	ID [23:16]								ID
TBUF+3	-			ID [28:24]					ID
TBUF+4	IDE=0	RTR	FDF	BRS	DLC [3:0]				Control
TBUF+5	-								-
TBUF+6	-								-
TBUF+7	-								-
TBUF+8	DATA1								Data
TBUF+9	DATA2								Data
TBUF+10	DATA3								Data
-	-								Data
-	-								Data
-	-								Data
TBUF+70	DATA63								Data
TBUF+71	DATA64								Data

For the meaning of the control bits, please refer to the CAN Receive BUF Register section.

34.5.4 CAN Configuration and Status Register (CAN_CFG_STAT)

CAN Configuration and Status Register

Offset address: 0xA0

Reset value: 0x80

	b7	b6	b5	b4	b3	b2	b1	b0
	RESET	LBME	LBMI	TPSS	TSSS	RACTIVE	TACTIVE	BUSOFF

Bit	Flag	Bit Name	Function	Read and Write
b7	RESET	Reset request	reset request bit 0: Do not request a partial reset 1: request partial reset Some registers can only be written when RESET=1. For details, please refer to the software reset function. Please note that it takes 11 CAN bit times for the node to participate in communication when RESET=0.	R/W
b6	LBME	External loopback mode enable bit	External loopback mode enable bit 0: disable external loopback mode 1: Enable external loopback mode Note: It is forbidden to set this bit during communication.	R/W
b5	LBMI	Internal loopback mode enable bit	Internal loopback mode enable bit 0: disable internal loopback mode 1: Enable internal loopback mode Note: It is forbidden to set this bit during communication.	R/W
b4	TPSS	PTB single transfer mode	PTB single transfer mode 0: Disable PTB single transfer mode 1: Enable PTB single transfer mode	R/W
b3	TSSS	STB single transfer mode	STB single transfer mode 0: Disable STB single transfer mode 1: Enable STB single transfer mode	R/W
b2	RACTIVE	Receive status signal	Receive status signal 0: not receiving 1: receiving	R
b1	TACTIVE	Sending status signal	Sending status signal 0: not sending 1: Sending	R
b0	BUSOFF	Bus off state	Bus off state 0: Bus valid state 1: Bus off state NOTE: Writing a 1 clears the TECNT and RECNT registers, but only for debug purposes.	R/W

34.5.5 CAN Command Register (CAN_TCMD)

CAN Command Register

Offset address: 0xA1

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
TBSEL	LOM	-	TPE	TPA	TSONE	TSALL	TSA

Bit	Flag	Bit Name	Function	Read and Write
b7	TBSEL	Send BUF selection bit	Send BUF selection bit (Transmit Buffer Select) 0: PTB 1: STB When TTEN=1&TTTBM=1, TBSEL is reset to reset value. Note: When writing TBUF register or TSNEXT bit, this bit needs to keep a constant value.	R/W
b6	LOM	Silent mode enable bit	Silent mode enable bit (Listen Only Mode) 0: disable silent mode 1: Enable silent mode Sending is prohibited when LOM=1&LBME=0. When LOM=1&LBME=1, it is forbidden to respond to the corresponding received frames and error frames, but it can send data. Note: It is forbidden to set this bit during communication.	R/W
b5	Reserved	-	The reset value must be maintained.	R/W
b4	TPE	PTB transmit enable bit	PTB send enable bit (Transmit Primary Enable) 0: Disable PTB sending 1: Enable PTB transmission After this bit is enabled, the Mailbox in PTB will be sent at the next available sending position. STB transmissions that have already started will continue, but the next pending STB transmission will be delayed until the PTB transmission is complete. After this bit is written to 1, it will remain 1 until the PTB transmission is completed or the transmission is canceled by TPA. Software cannot clear this bit by writing a 0. The TPE is reset to the reset value by hardware in the following cases: - RESET=1 - BUSOFF=1 - LOM=1&LBME=0 - TTEN=1&TTTBM=1	R/W
b3	TPA	PTB send cancel bit	PTB send cancel bit (Transmit Primary Abort) 0: do not cancel 1: Cancel the PTB transmission that has been requested by setting TPE to 1 but has not yet started This bit is written 1 by software but cleared by hardware. The TPE bit can be cleared by writing 1, so write 1 at the same time as TPE. The TPE is reset to the reset value by hardware in the following cases: - RESET=1 - BUSOFF=1 - TTEN=1&TTTBM=1	R/W
b2	TSONE	Send a frame of STB data set bit	Send a frame of STB data set bit (Transmit Secondary ONE frame) 0: do not send 1: Send a frame of STB data In FIFO mode, the earliest written data is sent, and in priority mode, the highest priority data is sent After this bit is written to 1, it will remain 1 until the STB transmission is completed or the transmission is canceled by TSA. Software cannot clear this bit by writing a 0. TSONE is reset to the reset value by hardware in the following cases: - RESET=1 - BUSOFF=1 - LOM=1&LBME=0 - TTEN=1&TTTBM=1	R/W

b1	TSALL	Send all STB data set bits	Send all STB data set bits (Transmit Secondary ALL frame) 0: do not send 1: Send all data in STB After this bit is written to 1, it will remain 1 until the STB transmission is completed or the transmission is canceled by TSA. Software cannot clear this bit by writing a 0. The TSALL is reset to the reset value by hardware in the following cases: – RESET=1 – BUSOFF=1 – LOM=1&LBME=0 – TTEN=1&TTTBM=1	R/W
b0	TSA	STB sends cancel bit	STB send abort bit (Transmit Secondary Abort) 0: do not cancel 1: Cancel the STB transmission that has been requested by TSONE or TSALL but has not yet started This bit is written 1 by software but cleared by hardware. Write 1 to clear TSONE or TSALL bit. TSA is reset to the reset value by hardware in the following cases: – RESET=1 – BUSOFF=1	R/W

34.5.6 CAN Transmit Control Register (CAN_TCTRL)

CAN Transmit Control Register

Offset address: 0xA2

Reset value: 0x90

b7	b6	b5	b4	b3	b2	b1	b0
FD_ISO	TSNEXT	TSMODE	TTTBM	-	-	TSSTAT [1:0]	

Bit	Flag	Bit Name	Function	Read and Write
b7	FD_ISO	CAN FD ISO Mode	CAN FD ISO mode (can only be written when CAN_CFG_STAT.RESET=1) 0: Bosch CAN FD mode 1:11898-1:2015 CAN FD mode	R/W
b6	TSNEXT	Next STB SLOT	Next STB (Transmit buffer Secondary NEXT) 0: no action 1: The current STB SLOT is filled, pointing to the next SLOT After the application program finishes writing the data in TBUF, it indicates that the current STB SLOT has been filled by setting the TSNEXT bit, so that the hardware points TBUF to the next STB SLOT. The data in the STB SLOT identified by the TSNEXT bit can be sent through the TSONE or TSALL bit. This bit is written to 1 by the application program and cleared to 0 by hardware. After all STB SLOTS are filled, TSNEXT remains 1 until a STB SLOT is released. Note: This bit is fixed to 0 in TTCAN mode.	R/W
b5	TSMODE	STB send mode	STB send mode (Transmit buffer Secondary operation MODE) 0: FIFO mode 1: priority mode FIFO mode is sent according to the order in which data frames are written. The priority mode is automatically judged according to the ID. The smaller the ID, the higher the priority. Regardless of the mode, PTB has the highest priority. Note: The TSMODE bit can only be set when the STB is empty.	R/W
b4	TTTBM	TTCAN BUF mode	TTCAN BUF mode (TTCAN Transmit Buffer Mode) When TTEN=0, TTTBM is ignored. 0: TSMODE decision, PTB and STB 1: Set by TBPTR and TTPTR In TTCAN mode, this bit can be set to 0 when only receiving the time stamp function is required, and TSMODE is used to determine whether to use PTB or STB. Note: The TSMODE bit can only be set when the STB is empty.	R/W
b3~b2	Reserved	-	The reset value must be maintained.	R/W
b1~b0	TSSTAT	STB state	STB status (Transmission Secondary Status bits) TTEN=0 or TTEN=1 & TTTBM=0 00: STB empty 01: STB is less than or equal to half full 10: STB is more than half full 11: STB full TTEN=1 and TTTBM=1 00: PTB and STB empty 01: PTB and STB are not full 10: reserved 11: PTB and STB full	R

34.5.7 CAN Receive Control Register (CAN_RCTRL)

CAN Receive Control Register

Offset address: 0xA3

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
SACK	ROM	ROV	RREL	RBALL	-	RSTAT [1:0]	

Bit	Flag	Bit Name	Function	Read and Write
b7	SACK	Self-response	Self-acknowledgement (Self-ACKnowledge) 0: no self-response 1: When LBME=1, enable the self-acknowledgment function	R/W
b6	ROM	Receive BUF overflow mode setting bit	Receive BUF overflow mode setting bit (Receive buffer Overflow Mode) 0: The earliest received data is overwritten 1: Newly received data is not stored	R/W
b5	ROV	Receive BUF overflow flag bit	Receive BUF overflow flag (Receive buffer Overflow) 0: no overflow 1: overflow, at least one data is lost Cleared by writing RREL to 1.	R
b4	RREL	Disable receive BUF	Disable receive BUF (Receive buffer RELease) 0: do not release 1: Indicates that the receiving BUF has been read, and the RBUF register points to the next RB SLOT.	R/W
b3	RBALL	Receive BUF data to store all data frames	Receive BUF data stores all data frames (Receive Buffer stores ALL data frames) 0: Normal mode 1: Store all data including error data.	R/W
b2	Reserved	-	The reset value must be maintained.	R/W
b1~b0	RSTAT	Receive BUF status	Receive BUF status (Receive buffer STATus) 00: RBUF empty 01: RBUF is not empty but less than AFWL programming value 10: RBUF is greater than or equal to the programmed value of AFWL but not full 11: full (keep this value on overflow)	R

34.5.8 CAN Receive and Transmit Interrupt Enable Register (CAN_RTIE)

CAN Receive and Transmit Interrupt Enable Register

Offset address: 0xA4

Reset value: 0xFE

b7	b6	b5	b4	b3	b2	b1	b0
RIE	ROIE	RFIE	RAFIE	TPIE	TSIE	EIE	TSFF

Bit	Flag	Bit Name	Function	Read and Write
b7	RIE	Receive interrupt enable	Receive Interrupt Enable (Receive Interrupt Enable) 0: Prohibited 1: Enable	R/W
b6	ROIE	Receive overflow interrupt enable	Receive overflow interrupt enable (Receive Overrun Interrupt Enable) 0: Prohibited 1: Enable	R/W
b5	RFIE	Receive BUF full interrupt enable	Receive BUF full interrupt enable (RB Full Interrupt Enable) 0: Prohibited 1: Enable	R/W
b4	RAFIE	Receive BUF will be full interrupt enable	Receive BUF will be full interrupt enable (RB Almost Full Interrupt Enable) 0: Prohibited 1: Enable	R/W
b3	TPIE	PTB transmit interrupt enable	PTB transmit interrupt enable (Transmission Primary Interrupt Enable) 0: Prohibited 1: Enable	R/W
b2	TSIE	STB transmit interrupt enable	STB transmit interrupt enable (Transmission Secondary Interrupt Enable) 0: Prohibited 1: Enable	R/W
b1	EIE	Error interrupt enable	Error Interrupt Enable (Error Interrupt Enable) 0: Prohibited 1: Enable	R/W
b0	TSFF	Send BUF full flag	TTEN=0 or TTTBM=0: STB full flag (Transmit Secondary buffer Full Flag) 0: STB SLOT is not fully filled 1: STB SLOT is fully filled TTEN=1 and TTTBM=1: TB full flag (Transmit buffer Full Flag) 0: The sending BUF selected by TBPTR is not fully filled 1: The sending BUF selected by TBPTR is fully filled	R

34.5.9 CAN Receive and Transmit Interrupt Mark Register (CAN_RTIF)

CAN Receive and Transmit Interrupt Status Register

Offset address: 0xA5

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
RIF	ROIF	RFIF	RAFIF	TPIF	TSIF	EIF	AIF

Bit	Flag	Bit Name	Function	Read and Write
b7	RIF	Receive interrupt flag	Receive Interrupt Flag (Receive Interrupt Flag) 0: No data frame received 1: A valid data frame or remote frame is received Write 1 to clear 0 by application program.	R/W
b6	ROIF	Receive overflow interrupt flag	Receive overflow interrupt flag (Receive Overrun Interrupt Flag) 0: No RB is overwritten (overwrite) 1: At least one RB is covered Both ROIF and RFIF are set to 1 on overflow. Write 1 to clear 0 by application program.	R/W
b5	RFIF	Receive BUF full interrupt flag	Receive BUF full interrupt flag (RB Full Interrupt Flag) 0: RB FIFO is not full 1: RB FIFO is full Write 1 to clear 0 by application program.	R/W
b4	RAFIF	Receive BUF will be full interrupt flag	Receive BUF will be full interrupt flag (RB Almost Full Interrupt Flag) 0: The number of filled RB SLOTS is less than the AFWL setting value 1: The number of filled RB SLOTS is greater than or equal to the AFWL setting value Write 1 to clear 0 by application program.	R/W
b3	TPIF	PTB send interrupt flag	PTB transmit interrupt flag (Transmission Primary Interrupt Flag) 0: No PTB send completed 1: The requested PTB send completed successfully Write 1 to clear 0 by application program. Note: In TTCAN mode, TPIF is invalid, only the TSIF flag is applicable	R/W
b2	TSIF	STB send interrupt flag	STB transmit interrupt flag (Transmission Secondary Interrupt Flag) 0: No STB transmission completed 1: The requested STB send completed successfully Write 1 to clear 0 by application program. Note: In TTCAN mode, TPIF is invalid, only use TSIF flag	R/W
b1	EIF	Error interrupt flag	Error Interrupt Flag (Error Interrupt Flag) 0: The BUSOFF bit has not changed, or the relative relationship between the error counter value and the ERROR warning limit setting value has not changed. 1: The BUSOFF bit changes, or the relative relationship between the value of the error counter and the setting value of the ERROR warning limit changes. For example, the value of the error counter changes from less than the set value to greater than the set value, or from greater than the set value to less than the set value. Write 1 to clear 0 by application program.	R/W

b0	AIF	Cancel send interrupt flag	Cancel sending interrupt flag (Abort Interrupt Flag) 0: Send data is not canceled 1: The send message requested by TPA and TSA was successfully canceled. Write 1 to clear 0 by application program.	R/W

34.5.10 CAN Error Interrupt Enable and Flags Register (CAN_ERRINT)

CAN ERRor INTerrupt Enable and Flag Register

Offset address: 0xA6

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
EWARN	EPASS	EPIE	EPIF	ALIE	ALIF	BEIE	BEIF

Bit	Flag	Bit Name	Function	Read and Write
b7	EWARN	Reached the set ERROR WARNING LIMIT	Reached the set ERROR WARNING LIMIT (Error WARNING limit reached) 0: RECNT or TECNT is less than the set value of EWL 1: RECNT or TECNT is greater than or equal to the EWL setting value	R
b6	EPASS	Error passive	Error Passive mode active 0: The node is an active error node 1: Passive error node when node	R
b5	EPIE	Error Passive Interrupt Enable	Error Passive Interrupt Enable (Error Passive Interrupt Enable) 0: Prohibited 1: Enable	R/W
b4	EPIF	Error passive interrupt flag	Error Passive Interrupt Flag (Error Passive Interrupt Flag) 0: No change from error active to error passive or error passive to error active 1: Change from error active to error passive or error passive to error active Write 1 to clear 0 by application program.	R/W
b3	ALIE	Arbitration failure interrupt enable	Arbitration Lost Interrupt Enable (Arbitration Lost Interrupt Enable) 0: Prohibited 1: Enable	R/W
b2	ALIF	Arbitration Lost Interrupt Flag	Arbitration Lost Interrupt Flag (Arbitration Lost Interrupt Flag) 0: Arbitration is successful 1: Arbitration failure Write 1 to clear 0 by application program.	R/W
b1	BEIE	Bus Error Interrupt Enable	Bus Error Interrupt Enable (Bus Error Interrupt Enable) 0: Prohibited 1: Enable	R/W
b0	BEIF	Bus error interrupt flag	Bus Error Interrupt Flag (Bus Error Interrupt Flag) 0: No bus error 1: Bus error Write 1 to clear 0 by application program.	R/W

34.5.11 CAN Slow Bit Timing Register (CAN_BT)

CAN Slow Bit Timing Register

Offset address: 0xA8

Reset value: 0x0102 0203

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
S_PRESC [7:0]								-	S_SJW [6:0]						
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	S_SEG_2 [6:0]							S_SEG_1 [7:0]							

Bit	Flag	Bit Name	Function	Read and Write
b31~b24	S_PRESC	Prescaler setting	Prescaler setting (S_Prescaler) When this register is set to a value, the module communication clock is set to (S_PRESC+1) divided as TQ.	R/W
b23	Reserved	-	The reset value must be maintained.	R
b22~b16	S_SJW	Resynchronization compensation width time setting	Resynchronization compensation width time setting (Bit Timing Segment 2) Resynchronization compensation width time=(S_SJW+1)*TQ	R/W
b15	Reserved	-	The reset value must be maintained.	R
b14~b8	S_SEG_2	Bit segment 2 time setting	Bit segment 2 time unit setting (Bit Timing Segment 2) Segment 2 time=(S_SEG_2+1)*TQ	R/W
b7~b0	S_SEG_1	Bit segment 1 time setting	Bit segment 1 time unit setting (Bit Timing Segment 1) Segment 1 time=(S_SEG_1+2)*TQ	R/W

34.5.12 CAN Fast Timing Register (CAN_FBT)

CAN Fast Bit Timing Register

Offset address: 0xAC

Reset value: 0x0102 0203

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
F_PRESC [7:0]								-	-	-	-	F_SJW [3:0]			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	F_SEG_2 [3:0]				-	-	-	F_SEG_1 [4:0]				

Bit	Flag	Bit Name	Function	Read and write
b31~b24	F_PRESC	Prescaler setting	Prescaler setting (Prescaler) When this register is set to a value, the module communication clock is set to (F_PRESC+1) divided as TQ.	R/W
b23~b20	Reserved	-	The reset value must be maintained.	R
b19~b16	F_SJW	Resynchronization compensation width time setting	Resynchronization compensation width time setting (Bit Timing Segment 2) Resynchronization compensation width time = (F_SJW+1)*TQ	R/W
b15~12	Reserved	-	The reset value must be maintained.	R
b11~b8	F_SEG_2	Bit segment 2 time setting	Bit segment 2 time unit setting (Bit Timing Segment 2) Segment 2 time=(F_SEG_2+1)*TQ	R/W
b7~b5	Reserved	-	The reset value must be maintained.	R
b4~b0	F_SEG_1	Bit segment 1 time setting	Bit segment 1 time unit setting (Bit Timing Segment 1) Segment 1 time=(F_SEG_1+2)*TQ	R/W

34.5.13 CAN Send Delay Compensation Register (CAN_TDC)

CAN Transmitter Delay Compensation Register

Offset address: 0xb1

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
TDCEN	SSPOFF [6:0]						

Bit	Flag	Bit Name	Function	Read and write
b7	TDCEN	Transmitter delay compensation function enable bit	Transmitter delay compensation function enable bit For CAN FD BRS= 1 0: Disable 1: Enable	R/W
b6~b0	SSPOFF	TDC function auxiliary sampling point position setting bit	TDC function auxiliary sampling point position setting bit Set the unit to TQ	R/W

34.5.14 CAN Error and Arbitration Lost Capture Register (CAN_EALCAP)

CAN Error and Arbitration Lost Capture Register

Offset address: 0xB0

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
KOER [2:0]			ALC [4:0]				

Bit	Flag	Bit Name	Function	Read and write
b7~b5	KOER	Error Category	Error category 000: no error 001: bit error 010: wrong form 011: Padding error 100: Response error 101: CRC error 110: Other errors 111: reserved The KOER bit is updated when there is an error, and the KOER bit remains unchanged when sending and receiving normally.	R
b4~b0	ALC	Lost Arbitration Position Capture	Lost Arbitration Position Capture When the arbitration fails, the ALC records the position when the arbitration fails in one frame of data.	R

34.5.15 CAN Warning Limit Register (CAN_LIMIT)

CAN Warning Limits Register

Offset address: 0xA7

Reset value: 0x1B

b7	b6	b5	b4	b3	b2	b1	b0
AFWL [3:0]				EWL [3:0]			

Bit	Flag	Bit Name	Function	Read and Write
b7~b4	AFWL	Receive BUF will be full Warning Limit	Receive buffer almost full Warning Limit (receive buffer Almost Full Warning Limit) The setting range is 1~8. AFWL=0 is meaningless, treat it as AFWL=1.	R/W
b3~b0	EWL	Error Waring Limit programming value	Error Waring Limit programming value (Programmable Error Warning Limit) Error Waring Limit= (EWL+1) *8. The value set in this register affects the EIF flag.	R/W

34.5.16 CAN Receive Error Counter Register (CAN_RECNT)

CAN Receive Error CouNT Register

Offset address: 0xB2

Reset value: 0x00

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

Bit	Flag	Bit Name	Function	Read and Write
b7~b0	RECNT	Receive error counter	Receive Error Counter (Receive Error CouNT) The receive error counter increases or decreases according to the error count specified in the CAN protocol. There is no overflow in this counter, and 255 is the maximum value.	R

34.5.17 CAN Transmit Error Counter Register (CAN_TECNT)

CAN Transmit Error CouNT Register

Offset address: 0xB3

Reset value: 0x00

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

Bit	Flag	Bit Name	Function	Read and Write
b7~b0	TECNT	Send error counter	Transmit Error Counter (Transmit Error CouNT) The sending error counter increases or decreases according to the error count specified in the CAN protocol. There is no overflow in this counter, and 255 is the maximum value.	R

34.5.18 CAN Filter Group Control Register (CAN_ACFCTRL)

CAN Acceptance Filter Control Register

Offset address: 0xB4

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	-	SELMASK	-	ACFADR			

Bit	Flag	Bit Name	Function	Read and Write
b7~b6	Reserved	-	The reset value must be maintained.	R
b5	SELMASK	Select the mask register for the filter	Select the mask register of the filter (SElect acceptance MASK) 0: ACF points to the filter ID register 1: ACF points to the filter MASK register Select a specific filter register set by ACFADR	R/W
b4	Reserved	-	The reset value must be maintained.	R
b3~b0	ACFADR	Filter address	Acceptance filter address (acceptance filter address) ACFADR points to a specific filter, and uses SELMASK to distinguish between ID and MASK. 0000: point to ACF_1 0001: point to ACF_2 0010: point to ACF_3 0011: point to ACF_4 0100: point to ACF_5 0101: point to ACF_6 0110: point to ACF_7 0111: point to ACF_8 1000: point to ACF_9 1001: point to ACF_10 1010: point to ACF_11 1011: point to ACF_12 1100: point to ACF_13 1101: point to ACF_14 1110: point to ACF_15 1111: point to ACF_16	R/W

34.5.19 CAN Filter Group Enable Register (CAN_ACFEN)

CAN Acceptance Filter Enable Register

Offset address: 0xB6

Reset value: 0x0001

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
AE_16	AE_15	AE_14	AE_13	AE_12	AE_11	AE_10	AE_9	AE_8	AE_7	AE_6	AE_5	AE_4	AE_3	AE_2	AE_1

Bit	Flag	Bit Name	Function	Read and Write
b15	AE_16	ACF_16 enable	ACF_16 enable (Acceptance Filter 16 Enable) 0: disable 1: enable	R/W
b14	AE_15	ACF_15 enable	ACF_15 enable (Acceptance Filter 15 Enable) 0: disable 1: enable	R/W
b13	AE_14	ACF_14 enable	ACF_14 enable (Acceptance Filter 14 Enable) 0: disable 1: enable	R/W
b12	AE_13	ACF_13 enable	ACF_13 enable (Acceptance Filter 13 Enable) 0: disable 1: enable	R/W
b11	AE_12	ACF_12 enable	ACF_12 enable (Acceptance Filter 12 Enable) 0: disable 1: enable	R/W
b10	AE_11	ACF_11 enable	ACF_11 enable (Acceptance Filter 11 Enable) 0: disable 1: enable	R/W
b9	AE_10	ACF_10 enable	ACF_10 enable (Acceptance Filter 10 Enable) 0: disable 1: enable	R/W
b8	AE_9	ACF_9 enable	ACF_9 enable (Acceptance Filter 9 Enable) 0: disable 1: enable	R/W
b7	AE_8	ACF_8 enable	ACF_8 enable (Acceptance Filter 8 Enable) 0: disable 1: enable	R/W
b6	AE_7	ACF_7 enable	ACF_7 enable (Acceptance Filter 7 Enable) 0: disable 1: enable	R/W
b5	AE_6	ACF_6 enable	ACF_6 enable (Acceptance Filter 6 Enable) 0: disable 1: enable	R/W
b4	AE_5	ACF_5 enable	ACF_5 enable (Acceptance Filter 5 Enable) 0: disable 1: enable	R/W
b3	AE_4	ACF_4 enable	ACF_4 enable (Acceptance Filter 4 Enable) 0: disable 1: enable	R/W
b2	AE_3	ACF_3 enable	ACF_3 enable (Acceptance Filter 3 Enable) 0: disable 1: enable	R/W
b1	AE_2	ACF_2 enable	ACF_2 enable (Acceptance Filter 2 Enable) 0: disable 1: enable	R/W
b0	AE_1	ACF_1 enable	ACF_1 enable (Acceptance Filter 1 Enable) 0: disable 1: enable	R/W

34.5.20 CAN filter group code and mask registers (CAN_ACF)

CAN Acceptance Filter code and mask Register

Offset address: 0xB8

Reset value: 0XXXXX XXXX

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	AIDEE	AIDE	ACODE [28:16] or AMASK [28:16]												
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ACODE [15:0] or AMASK [15:0]															

Bit	Flag	Bit Name	Function	Read and Write
b31	Reserved	-	The readout value is indeterminate.	R
b30	AIDEE	IDE bit compare enable	IDE bit comparison enable (Acceptance mask IDE bit check enable) Valid only when SELMASK=1 0: Filter accepts standard format and extended format frames 1: The filter accepts standard format or extended format frames defined by the AIDE bits	R/W
b29	AIDE	IDE bit MASK	IDE bit MASK 0: filter only accepts standard formats 1: Filter only accepts extended formats	R/W
b28~b0	ACODE/ AMASK	Filter CODE/ Filter MASK	Filter CODE (acceptance filter code) Point to the specific filter via ACFADR. When SELMASK=0, it means the CODE of the filter. Use bit 10~bit 0 for the standard format, and bit 29~bit 0 for the extended format. Filter MASK (acceptance filter mask) Point to the specific filter via ACFADR. When SELMASK=1, it indicates the mask of the filter. Use bit 10~bit 0 for the standard format, and bit 29~bit 0 for the extended format.	R/W

34.5.21 TTCAN TB slot pointer register (CAN_TBSLOT)

TTCAN TB Slot Pointer Register

Offset address: 0xBE

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
TBE	TBF	-	-	-	TBPTR [2:0]		

Bit	Flag	Bit Name	Function	Read and Write
b7	TBE	Set TB to empty	Set TB slot to empty (set TB slot to "empty") 0: No operation 1: The SLOT selected by TBPTR is flagged as empty When SLOT is flagged empty and TSFF=0, TBE is automatically reset to 0. If this bit is set to 1, there is data being sent in the selected SLOT, then TBE=1, then TBE is reset to 0 after the sending is completed, the sending is wrong, or the sending is canceled. TBE has higher priority than TBF.	R/W
b6	TBF	Set TB to filled	Set TB slot to " Filled" (set TB slot to "Filled") 0: No operation 1: SLOT selected by TBPTR is flagged as filled TBE is automatically reset to 0 when SLOT is flagged as filled and TSFF=1.	R/W
b5~b3	Reserved	-	The reset value must be maintained.	R/W
b2~b0	TBPTR	TB SLOT pointer	TB SLOT pointer (Pointer to a TB message slot) 000: point to PTB 001: point to STB SLOT1 010: point to STB SLOT2 011: point to STB SLOT3 Other: prohibitions The pointed TB SLOT can be read and written through TBUF, and can be flagged whether it has been filled through TBE and TBF. In TTCAN mode, the TBSEL and TSNEXT registers are invalid. Note: This bit can only be written when TSFF=0.	R/W

34.5.22 TTCAN Time Triggered Configuration Register (CAN_TTCFG)

TTCAN TB Slot Pointer Register

Offset address: 0xBF

Reset value: 0x90

b7	b6	b5	b4	b3	b2	b1	b0
WTIE	WTIF	TEIF	TTIE	TTIF	T_PRESC [1:0]		TTEN

Bit	Flag	Bit Name	Function	Read and write
b7	WTIE	Trigger gate interrupt enable	Trigger watchdog interrupt enable (Watch Trigger Interrupt Enable) 0: Prohibited 1: Enable	R/W
b6	WTIF	trigger gatekeeper interrupt flag	Trigger the gate interrupt flag (Watch Trigger Interrupt Flag) When CYCLE COUNT value=TT_WTRIG setting value and WTIE=1, WTIF is set. Write 1 to clear 0 by application program.	R/W
b5	TEIF	Trigger error interrupt flag	Error Interrupt Flag (Trigger Error Interrupt Flag) When the set value of TT_TTIG is less than the actual CYCLE_TIME, TEIF is set. Write 1 to clear 0 by application program.	R/W
b4	TTIE	Time-triggered interrupt enable	Time Trigger Interrupt Enable (Time Trigger Interrupt Enable) 0: Prohibited 1: Enable	R/W
b3	TTIF	Time triggered interrupt flag	Time Trigger Interrupt Flag (Time Trigger Interrupt Flag) When CYCLE COUNT value = TT_TRIG set value and TTIE = 1, TTIF is set. If TT_TRIG is not updated, TTIF is only set once, and the next basic CYCLE is not set. Write 1 to clear 0 by application program.	R/W
b2~b1	T_PRESC	TTCAN counter prescaler	TTCAN Counter Prescaler (TTCAN Timer PRESCaler) 00: Divide by 1 the bit time set by the SBT register 01: Divide by 2 the bit time set by the SBT register 10: Divide by 4 the bit time set by the SBT register 11: Eighth frequency division of the bit time set by the SBT register Note: T_PRESC can be written when TTEN=0 or simultaneously when writing TTEN=1.	R/W
b0	TTEN	TTCAN enable	TTCAN enable (Time Trigger Enable) 0: Prohibited 1: Enable TTCAN, the counter starts counting.	R/W

34.5.23 TTCAN Reference Message Register (CAN_REF_MSG)

TTCAN Reference Message Register

Offset address: 0xC0

Reset value: 0x0000 0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
REF_IDE	-	REF_ID [28:16]													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
REF_ID [15:0]															

Bit	Flag	Bit Name	Function	Read and Write
b31	REF_IDE	IDE bits for reference messages	IDE bit of the reference message (REfERENCE message IDE bit) 0: standard format 1: Extended format	R/W
b30~b29	Reserved	-	The readout value is indeterminate.	R/W
b28~b0	REF_ID	ID bit of the reference message	The ID bit of the reference message (REfERENCE message IDentifier) REF_IDE=0: REF_ID [10:0] is valid REF_IDE=1: REF_ID [28:0] is valid REF_ID is used to detect reference messages and is applicable for sending and receiving. After the reference message is detected, the Sync_Mark of the current frame becomes Ref_Mark. REF_ID [2:0] is fixed to 0, and its value is not checked, so that up to 8 potential time masters can be supported. After the highest byte of REF_MSG is written, it is necessary to wait for 6 CAN clock periods to complete the transfer of REF_MSG to the CAN clock domain.	R/W

34.5.24 TTCAN Trigger Configuration Register (CAN_TRG_CFG)

TTCAN Reference Message Register

Offset address: 0xC4

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TEW [3:0]				-	TTYPE [2:0]				-				TTPTR [2:0]		

Bit	Flag	Bit Name	Function	Read and write
b15~b12	TEW	Transmit Enable Window	Transmit Enable Window (Transmit Enable Window) For the Single Shot Transmit Trigger mode of TTCAN, a window of TEW+1 period time can be set, and transmission is only allowed within this window.	R/W
b11	Reserved	-	The reset value must be maintained.	R
b10~b8	TTYPE	Trigger type	Trigger type (Trigger Type) 000: Immediate Trigger (Immediate Trigger for immediate transmission) 001: Time Trigger (Time Trigger for receive triggers) 010: Single Shot Transmit Trigger (Single Shot Transmit Trigger for exclusive time windows) 011: Transmit Start Trigger (Transmit Start Trigger for merged arbitrating time windows) 100: Transmit Stop Trigger (Transmit Stop Trigger for merged arbitrating time windows) Other: reserved The trigger time is set by the TT_TRIG register, and the TB Slot is selected by TTPTR.	R/W
b7~b3	Reserved	-	The reset value must be maintained.	R
b2~b0	TTPTR	Send trigger TB slot pointer	Transmit Trigger TB slot pointer (Transmit Trigger TB slot Pointer) 000: point to PTB 001: point to STB SLOT1 010: point to STB SLOT2 011: point to STB SLOT3 Other: prohibitions If the pointed TB SLOT is flagged empty, TEIF is set when the trigger time is reached.	R/W

34.5.25 TTCAN Trigger Time Register (CAN_TT_TRIG)

TTCAN Reference Message Register

Offset address: 0xC6

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TT_TRIG [15:0]															

Bit	Flag	Bit Name	Function	Read and write
b15~b0	TT_TRIG	Trigger Time	Trigger Time It is used to specify the period time of the trigger. For the sending trigger, the sending SOF time is about TT_TRIG setting value +1 When the highest byte of TT_TRIG is written, the TT_TRIG value begins to be transferred to the CAN clock domain. Therefore, if BYTE is operated, it is necessary to write the low byte first and then the high byte.	R/W

34.5.26 TTCAN Trigger Watchdog Time Register (CAN_TT_WTRIG)

TTCAN Watch Trigger Time Register

Offset address: 0xC8

Reset value: 0xFFFF

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TT_WTRIG [15:0]															

Bit	Flag	Bit Name	Function	Read and Write
b15~b0	TT_WTRIG	Trigger Time	Trigger time (Trigger Time) Used to specify the period time of the gate trigger. When the highest byte of TT_WTRIG is written, the TT_WTRIG value begins to be transferred to the CAN clock domain. Therefore, if BYTE is operated, it is necessary to write the low byte first and then the high byte.	R/W

34.6 Precautions

34.6.1 CAN Bus Anti-jamming Measures

The CAN bus is widely used in automobiles, industrial control and other industries. If the electromagnetic environment of the CAN application site is relatively harsh, there are factors such as circuit imbalance, space electromagnetic field, and grid incoming lines, which will cause a large amount of communication noise on the CAN bus due to radiation and conduction interference. , resulting in the increase of bus error frames, frequent retransmissions, and the failure of correct data to arrive in time, which seriously affects 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)

- Increase electrical isolation of CAN bus interface
- Common transceiver signal ground
- Use shielded twisted pair cables and ground them properly
- Improve twisted pair degree of CAN transmission line
- Add signal protector
- Improve network topology
- Application layer software anti-jamming mechanism

34.6.2 CAN Controller Noise Constraints

In the CAN bus network, it should be ensured that the bit time of the communication meets the requirements of the standard protocol. If the noise interference that does not meet the bit time width is introduced, it may cause the abnormal operation of the CAN controller.

35 MDIO Serial Bus Interface (MDIO)

35.1 Introduction

MDIO interface is included in the IEEE802.3 protocol and is a serial bus interface dedicated to the management of Ethernet PHY , using IEEE 802.3 Clause45. The MDIO interface hardware can receive complete MDIO frames without software intervention. The MDIO interface hardware can also send complete MDIO frames without software intervention as long as the data to be sent is provided before the turnaround bit (TA) of a received read data frame or read post-increment address frame. To aid in the use and provision of relevant data, an interrupt is generated at the end of each complete frame. If the received PHYADR or DEVADD does not match the expected value, no action is taken on the frame. An interrupt can also be generated after each valid PHYADR and DEVADD to allow more complex control within the frame.

35.2 Module Diagram

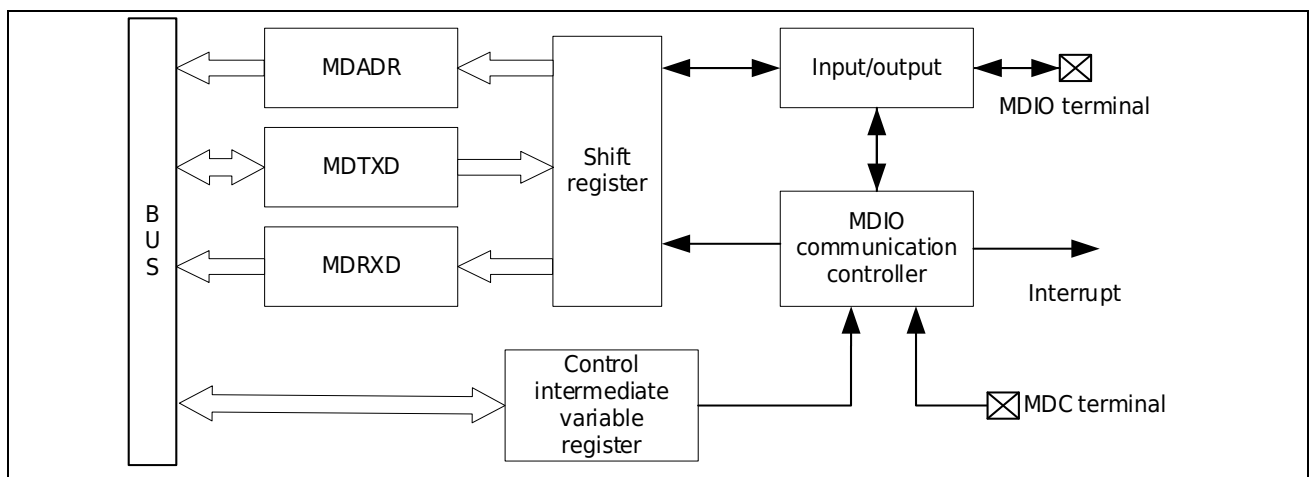


Figure 35-1 MDIO Module Structure Diagram

35.3 Functional Description

1. Support up to 4MHZ in slave mode
2. Support CFP/CFP2 MSA management interface specification
3. Support multiple interrupts
4. Physical address can be configured
 - Software Configuration
 - Hardware pin configuration

35.3.1 MDIO Frame Structure

MDIO interface uses the communication data frame structure defined in IEEE 802.3 Clause45. The frame structure is shown figure Figure 35-2 below.

Each frame can be an address frame or a data frame. The total bit length of each frame is 64 bits, consisting of a 32-bit preamble and a frame command body. The command body consists of six parts, as shown in the Figure 35-2. See Table 35-1 for more information on the various frame types. All values are transmitted big endian.

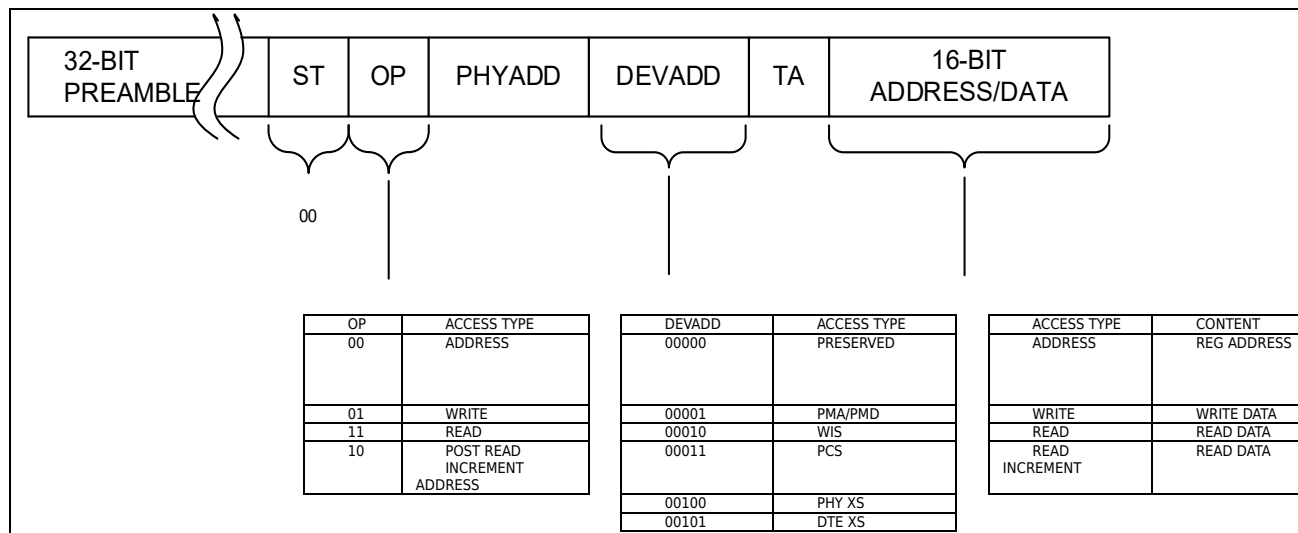


Figure 35-2 MDIO Frame Structure

ST : Start bit (2 bits)

OP : Operation code (2 bits)

PHYADD : Physical port address (5 bits)

DEVADD : MDIO device address (5 bits)

TA : state transition field (2 bits)

16-BIT ADDRESS/DATA : Address / data

Frame details for different frame types

Table 35-1 MDIO Different Types of Frames

Frame	idle	Management Frame Fields							Idle
		PRE	ST	OP	PHYADR	DEVADD	TA	Address/Data	
Write Address	Z	1...1	00	00	aaaaa	aaaaa	10	aaaaaaaaaaaaaaaa	Z
Write Data	Z	1...1	00	01	aaaaa	aaaaa	10	dddddddddddddddd	Z
Read Data	Z	1...1	00	11	aaaaa	aaaaa	z0	dddddddddddddddd	Z
Post Read Increment Address	Z	1...1	00	10	aaaaa	aaaaa	z0	dddddddddddddddd	Z

In the idle condition, MDC and MDIO are not driven active. During the second bit of TA and the 16-bit read data frame and the post-read increment address frame, MDIO is driven by MMD. At all other times, MDIO is driven by STA.

IDLE (Idle Condition)

MDIO is high impedance.

PRE (Preamble)

Before each transaction, the STA (host) sends a sequence of at least 32 consecutive bits to the MDIO, one bit at a time, with 32 response clock periods on the MDC to establish the start of the frame.

ST (Start of Frame) starts

After the preamble, ST (containing two 0 bits) indicates the start of the frame information.

OP (Operation Code)

OP specifies the action to be taken, as shown in Table 35-2

Table 35-2 MDIO Operation Code

OP	Descriptions
00	Set the address for a subsequent write or read frame
01	Write to the previously set address
10	Read from the previously set address
11	Read from the previously set address. Then increment the address. Note that user code must increment the address in the MDADR register

PHYADR (Physical Address) Physical address

This address is 5 bits, allowing 32 unique addresses. PHYADR is set by 5 pins or software.

DEVADD (Device Address) Device address

This address is 5 bits and is used to select the device type. In the CFP standard, only MDIO device address 1 is supported.

TA (Turnaround) state transition

This time is used to change from being driven by the host to being driven by the slave.

Address/Data

The address / data field is 16 bits.

35.4 Application Instances

35.4.1 Application Block Diagram

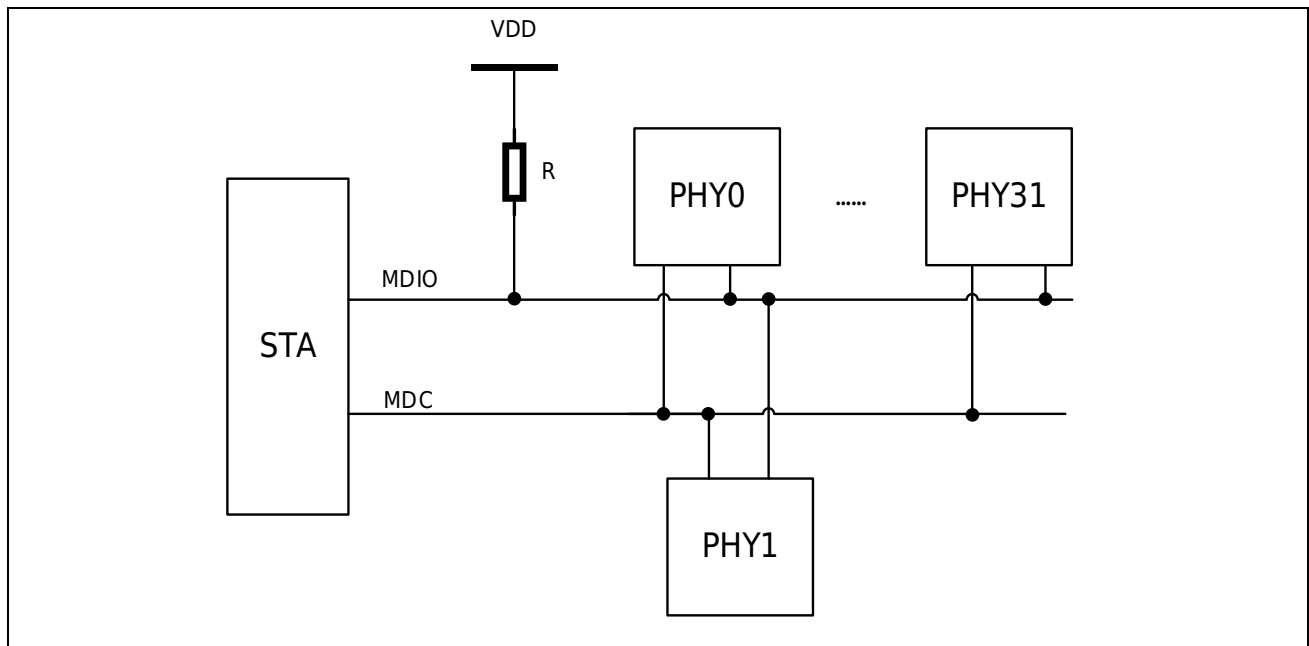


Figure 35-3 MDIO Application Block Diagram

MDIO is done by hardware, and only a few require software configuration.

1. Enable the port to function as MDIO.
2. Configure registers MDPHY , MDCON , and MDPIN to get control information.
3. Enable the interrupt register, MDIEN.
4. Write address frames and write data frames are received in the MDADR and MDRXD registers respectively.
5. MDTXD before a read data frame or a read post-increment address frame so that the data can be automatically inserted.

MDIO slave communication process:

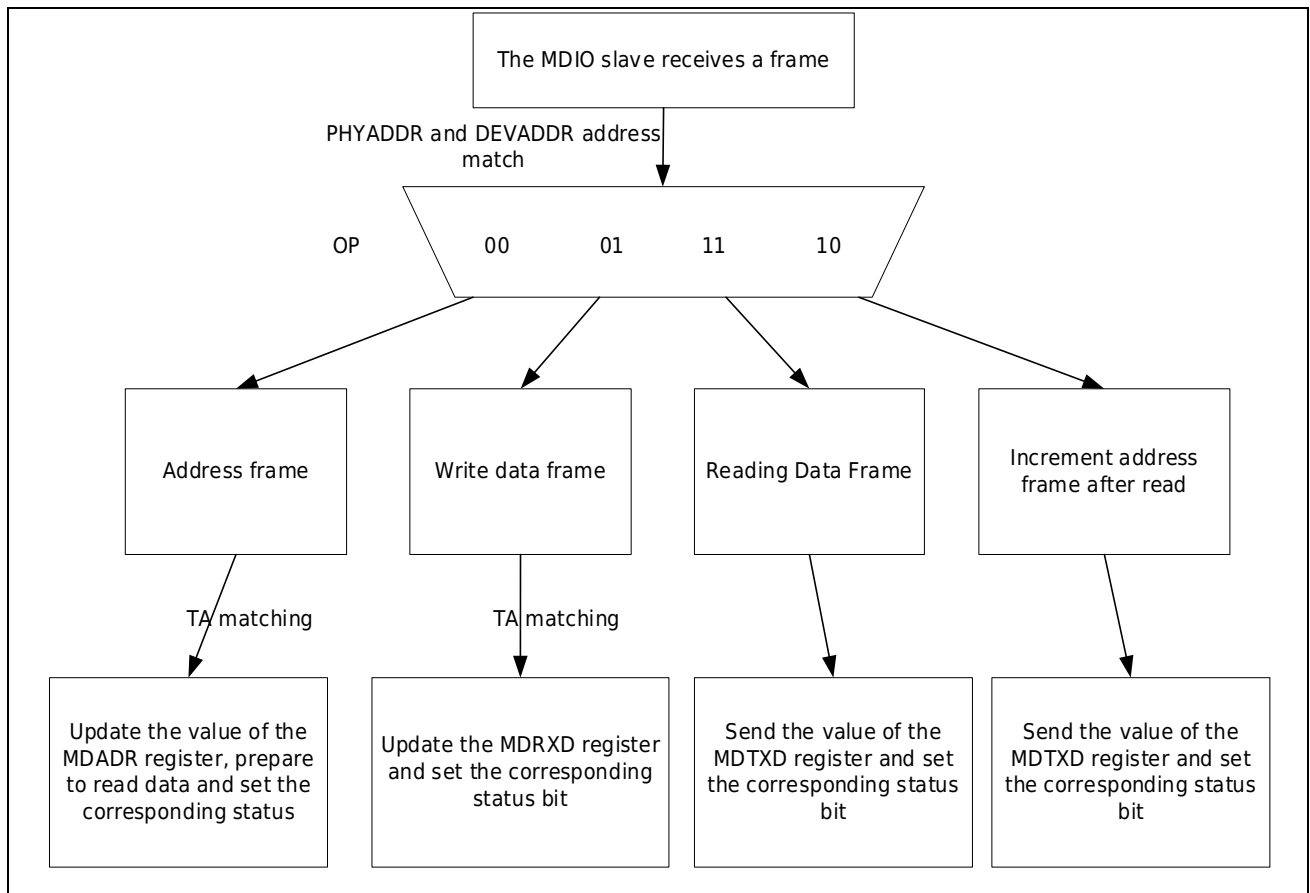


Figure 35-4 MDIO Slave Communication Flow Chart

After receiving the write address frame sent by the host, no matter the next frame is write or read, the data can be preset into the transmit data register in advance. If the next frame is a read operation, MDIO will send out the preset data. If a write address frame is given but no read operation is performed, the new data can overwrite the original transmit register after the next address frame is received.

MDIO slave receiving communication timing:

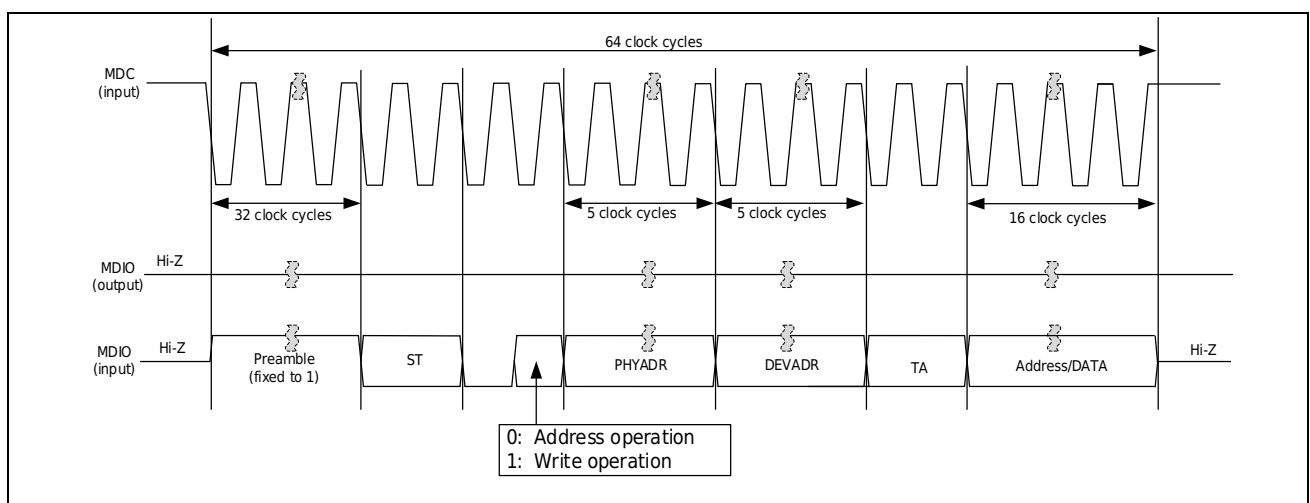


Figure 35-5 MDIO Slave Receiving Communication Timing Diagram

MDIO slave sends communication timing:

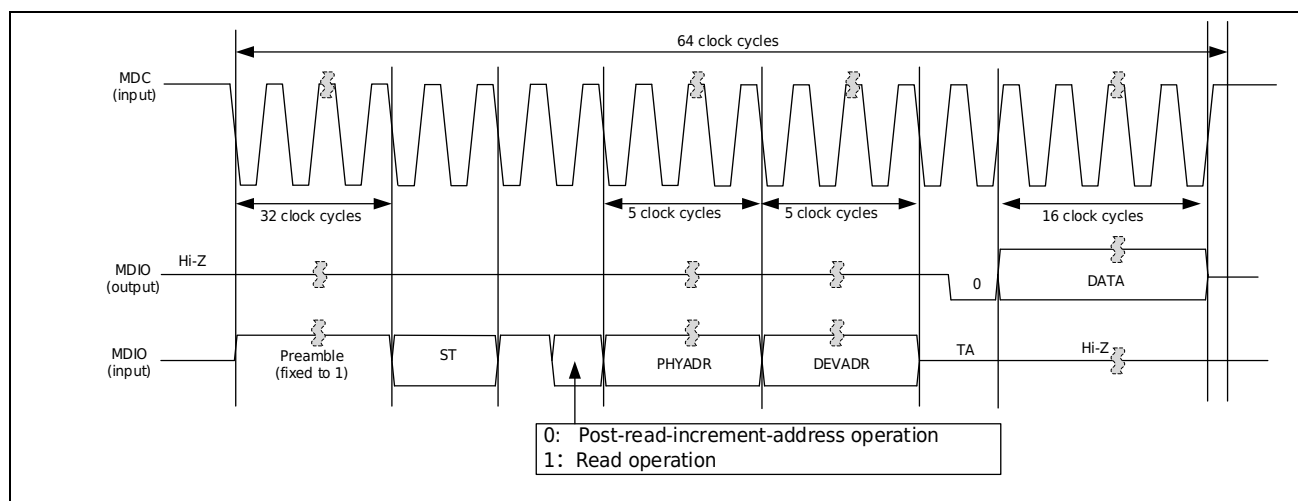


Figure 35-6 MDIO Slave Sending Communication Timing Diagram

MDIO slave continuous communication timing:

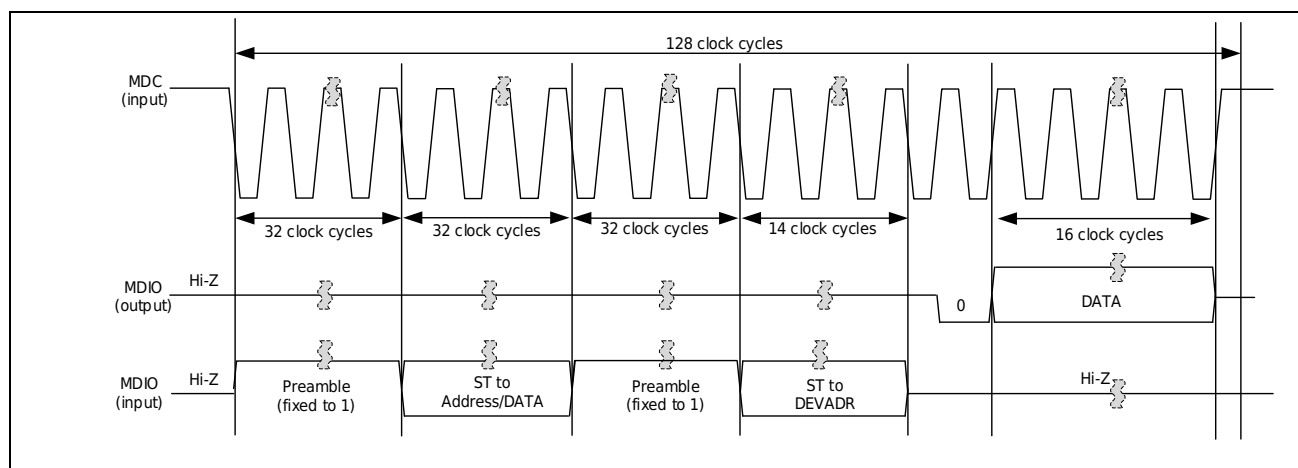


Figure 35-7 MDIO Slave Continuous Communication Timing Diagram

35.5 Register Description

Table 35-3 MDIO Register List

MDIO_BASE_ADDR:0x40028400

Register Name	Symbol	Offset Address	Reset Value
MDIO module control register	MDCON	0x00	0x00000000
MDIO module control register 1	MDCON1	0x04	0x00000000
MDIO receives frame control information and sends	MDFRM	0x08	0x00000000
MDIO receiving data register	MDRXD	0x0C	0x00000000
MDIO receiving address register	MDADR	0x10	0x00000000
MDIO transmission data register	MDTXD	0x14	0x00000000
MDIO physical address register	MDPHY	0x18	0x00000400
MDIO process signal register	MDSTA	0x1C	0x00000000
MDIO Interrupt permission register	MDIEN	0x20	0x00000000
MDIO read physical address pin register	MDPIN	0x24	0x00000000
MDIO timeout register	MDTO	0x28	0x00000000

35.5.1 MDIO Module Control Register(MDCON)

Offset address: 0x00

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
b23	b22	b21	b20	b19	b18	b17	b16
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
b15	b14	b13	b12	b11	b10	b9	b8
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
0	0	0	0	0	0	0	0
b7	b6	b5	b4	b3	b2	b1	b0
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	MD_PHYB	MD_RST
0	0	0	0	0	0	0	0

Bit	Flag	Bit Name	Function	Read and Write
b31-b3	Reserved	-	Read at "0", write at "0"	R/W
b2	Reserved	-	Read at "0", write at "0"	R/W
b1	MD_PHYB	Physical space bits	0: MDIO physical space occupies 5 bits 1: MDIO physical space occupies 3 bits Unused address bits are ignored	R/W
b0	MD_RST	Reset	Write "1" to reset the MDIO module The hardware clears the MD_RST value again immediately.	R/W

35.5.2 MDIO Module Control Register 1 (MDCON1)

Offset address: 0x04

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
b23	b22	b21	b20	b19	b18	b17	b16
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
b15	b14	b13	b12	b11	b10	b9	b8
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
0	0	0	0	0	0	0	0
b7	b6	b5	b4	b3	b2	b1	b0
Reserved	Reserved	Reserved	Reserved	MD_EN	SEL_EDGE	Reserved	MD_MSDEV
0	0	0	0	0	0	0	0

Bit	Flag	Bit Name	Function	Read and Write
b31-b4	Reserved	-	Read at "0", write at "0"	R/W
b3	MD_EN	MDIO module enable bit	1: MDIO enable 0: MDIO invalid	R/W
b2	SEL_EDGE	Timeout detection level selection	1 : Select MDC high level 0: Select MDC low level	R/W
b1	Reserved	-	Read at "0", write at "0"	R/W
b0	MD_MSDEV	Mask dev comparison results	1: DEVSDDR comparison result Mask 0: DEVADDR comparison result is not masked	R/W

35.5.3 MDIO Receive Frame Control Information Register (MDFRM)

Offset address: 0x08

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
b23	b22	b21	b20	b19	b18	b17	b16
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
b15	b14	b13	b12	b11	b10	b9	b8
Reserved	Reserved	MD_RTA [13:12]		MD_RDEVAD [11:8]			
0	0	0	0	0	0	0	0
b7	b6	b5	b4	b3	b2	b1	b0
MD_RDEVAD [7]	MD_RPHYAD [6:2]					MD_ROP [1:0]	
0	0	0	0	0	0	0	0

Bit	Flag	Bit Name	Function	Read and Write
b31-b14	Reserved	-	Read at "0", write at "0"	R/W
b13-b12	MD_RTA [1:0]	Conversion bit	Received frame TA bit information (only supports writing address frame or writing data frame TA bit)	R
b11-b7	MD_RDEVAD [4:0]	Receive device address bit	Receiving device address	R
b6-b2	MD_RPHYAD [4:0]	Receive physical address bits	Receive physical address	R
b1-b0	MD_ROP [1:0]	Receive opcode bits	Receive opcode 00: Address frame 01: Write frame 10 : Address increment frame after reading 11 : Read frame	R

35.5.4 MDIO Receive Data Register (MDRXD)

Offset address: 0x0C

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
b23	b22	b21	b20	b19	b18	b17	b16
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
b15	b14	b13	b12	b11	b10	b9	b8
MD_RXD [15:8]							
0	0	0	0	0	0	0	0
b7	b6	b5	b4	b3	b2	b1	b0
MD_RXD [7:0]							
0	0	0	0	0	0	0	0

Bit	Flag	Bit Name	Function	Read and Write
b31-b16	Reserved	-	Read at "0", write at "0"	R/W
b15-b0	MD_RXD [15:0]	Data reception bit	From the last written frame Received data	R

35.5.5 MDIO Recepting Address Register(MDADR)

Offset address: 0x10

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
b23	b22	b21	b20	b19	b18	b17	b16
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
b15	b14	b13	b12	b11	b10	b9	b8
MD_ADR [15:8]							
0	0	0	0	0	0	0	0
b7	b6	b5	b4	b3	b2	b1	b0
MD_ADR [7:0]							
0	0	0	0	0	0	0	0

Bit	Flag	Bit Name	Function	Read and Write
b31-b16	Reserved	-	Read at "0", write at "0"	R/W
b15-b0	MD_ADR [15:0]	Address reception bit	Slave Address Frame Received data	R

35.5.6 MDIO Transmission Data Register(MDTXD)

Offset address: 0x14

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
b23	b22	b21	b20	b19	b18	b17	b16
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
b15	b14	b13	b12	b11	b10	b9	b8
MD_TXD [15:8]							
0	0	0	0	0	0	0	0
b7	b6	b5	b4	b3	b2	b1	b0
MD_TXD [7:0]							
0	0	0	0	0	0	0	0

Bit	Flag	Bit Name	Function	Read and Write
b31-b16	Reserved	-	Read at "0", write at "0"	R/W
b15-b0	MD_TXD [15:0]	Transfer data bit	The next data frame is Data transferred	R/W

35.5.7 MDIO Physical Address Register(MDPHY)

Offset address: 0x18

Reset value: 0x00000400

b31	b30	b29	b28	b27	b26	b25	b24
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
b23	b22	b21	b20	b19	b18	b17	b16
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
b15	b14	b13	b12	b11	b10	b9	b8
Reserved	MD_DEVADD [14:10]					MD_PHYSEL [9:8]	
0	1	0	0	0	0	0	0
b7	b6	b5	b4	b3	b2	b1	b0
MD_PHYSEL [7:5]			MD_PHYSW [4:0]				
0	0	0	0	0	0	0	0

Bit	Flag	Bit Name	Function	Read and Write
b31-b15	Reserved	-	Read at "0", write at "0"	R/W
b14-b10	MD_DEVADD [4:0]	Expected device address bit	Expected device address Usually 01	R/W
b9-b5	MD_PHYSEL [4:0]	Expected physical address bit selection	Select the desired physical address bit 0: Set PHYADR.x = PRTADR _x pin 1: Set PHYADR.x = MD_PHYSW.x	R/W
b4-b0	MD_PHYSW [4:0]	Software provided physical address bit	Software provided physical address bit Selected by the corresponding MD_PHYSEL bit	R/W

35.5.8 MDIO Process Signal Register(MDSTA)

Offset address: 0x1C

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
b23	b22	b21	b20	b19	b18	b17	b16
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	MD_BUSY
b15	b14	b13	b12	b11	b10	b9	b8
Reserved	Reserved	Reserved	MD_RBNE	MD_OVR	MD_UDR	MD_TO	MD_TANM
0	0	0	0	0	0	0	0
b7	b6	b5	b4	b3	b2	b1	b0
MD_PHYN	MD_PHYM	MD_DEVN	MD_DEVM	MD_RDF	MD_INCF	MD_ADRF	MD_WRF
0	0	0	0	0	0	0	0

Bit	Flag	Bit Name	Function	Read and Write
b31-b17	Reserved	-	Read at "0", write at "0"	R/W
b16	MD_BUSY	Busy communication	Indicates that communication is in progress. Hardware Auto Reset	R
b13-b15	Reserved	-	Read at "0", write at "0"	R/W
b12	MD_RBNE	Data receiving buffer not empty flag	The data has been received and can be read. Read MDSTA to clear	RC(Read Clear)
b11	MD_OVR	Data receiving overflow flag	When MD_RBNE is set, after receiving the TA bit and the MDRXD register has not been read, it will be set to 1 and read to clear MDRXD. If MDRXD is read before the data in the receive shift register is transferred to MDRXD (i.e. before the 63rd MDC), the data of the current frame will not be lost.	RC
b10	MD_UDR	Data transmission underflow flag	When the data transmit buffer is empty, if no data is written to the MDTXD register before the second bit of the TA bit of the read data frame/read post-address increment frame, it will be set to 1 by hardware. Write MDTXD to clear it.	RC
b9	MD_TO	Timeout flag	The time between two bits of a complete frame, including the leading data part, will be set to 1 by hardware if the set timeout period has been reached and no new bit has been received or the data bit of the master and standby has not been sent out. Read MDSTA to clear	RC
b8	MD_TANM	TA bit mismatch	When the last frame of the write address or write data frame TA is received, if the received TA does not match the expected "10", it is set to 1 and cleared after reading MDSTA	RC
b7	MD_PHYN	Physical address mismatch	If the physical address does not match, set 1 at the end. Clear the value after reading MDSTA	RC
b6	MD_PHYM	Physical address match bit	If the physical address matches, set 1 at the end. Clear the value after reading MDSTA	RC
b5	MD_DEVN	Device address mismatch	If the device address does not match, set 1 at the end. Clear the value after reading MDSTA	RC
b4	MD_DEVM	Device address match bit	If the device address matches, set 1 at the end. Clear the value after reading MDSTA	RC
b3	MD_RDF	Both physical and device addresses match the read flag bit	If the device address and physical address match, it is set to 1 at the end of the address frame. Clear the value after reading MDSTA	RC
b2	MD_INCF	After both the physical and device addresses match, the address adds a flag bit	If the device address and physical address match, it is set to 1 at the end of the read auto-increment address frame. Clear the value after reading MDSTA	RC

b1	MD_ADRF	After the physical and device addresses match, the address label Flag bit	If the device address and physical address match, it is set to 1 at the end of the address frame. Clear the value after reading MDSTA	RC
b0	MD_WRF	After the physical and device addresses match, write the label Flag bit	If both the device address and physical address match, set to 1 at the end of the write frame. Clear the value after reading MDSTA	RC

35.5.9 MDIO Interrupt Enable Register (MDIEN)

Offset address: 0x20

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
b23	b22	b21	b20	b19	b18	b17	b16
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
b15	b14	b13	b12	b11	b10	b9	b8
Reserved	Reserved	Reserved	MD_RBNEIE	MD_OVRIE	MD_UDRIE	MD_TOIE	MD_TANMIE
0	0	0	0	0	0	0	0
b7	b6	b5	b4	b3	b2	b1	b0
MD_PHYNIE	MD_PHYMIE	MD_DEVNIE	MD_DEVMIE	MD_RDFIE	MD_INCFIE	MD_ADRFIE	MD_WRFIE
0	0	0	0	0	0	0	0

Bit	Flag	Bit Name	Function	Read and write
b31-b13	Reserved	-	Read at "0", write at "0"	R/W
b12	MD_RBNEIE	MD_RBNE valid interrupt request bit	If set, an interrupt request is issued when MD_RBNE is valid.	R/W
b11	MD_OVRIE	MD_OVR valid interrupt request bit	If set, an interrupt request is issued when MD_OVR is valid.	R/W
b10	MD_UDRIE	MD_UDR valid interrupt request bit	If set, an interrupt request is issued when MD_UDR is valid.	R/W
b9	MD_TOIE	MD_TO valid interrupt request bit	If set, an interrupt request is issued when MD_TO is valid.	R/W
b8	MD_TANMIE	MD_TANM valid interrupt request bit	If set, an interrupt request is issued when MD_TANM is valid.	R/W
b7	MD_PHYNIE	MD_PHYN valid interrupt request bit	If set, an interrupt request is issued when MD_PHYN is valid.	R/W
b6	MD_PHYMIE	MD_PHYM valid interrupt request bit	If set, an interrupt request is issued when MD_PHYM is valid.	R/W
b5	MD_DEVNIE	MD_DEVN valid interrupt request bit	If set, an interrupt request is issued when MD_DEVN is valid.	R/W
b4	MD_DEVMIE	MD_DEVM valid interrupt request bit	If set, an interrupt request is issued when MD_DEVM is valid.	R/W
b3	MD_RDFIE	MD_RDF valid interrupt request bit	If set, an interrupt request is issued when MD_RDF is valid.	R/W
b2	MD_INCFIE	MD_INCF valid interrupt request bit	If set, an interrupt request is issued when MD_INCF is valid.	R/W
b1	MD_ADRFIE	MD_ADRF valid interrupt request bit	If set, an interrupt request is issued when MD_ADRF is valid.	R/W
b0	MD_WRFIE	MD_WRF valid interrupt request bit	If set, an interrupt request is issued when MD_WRF is valid.	R/W

35.5.10 MDIO Read Physical Address Pin Register(MDPIN)

Offset address: 0x24

Reset value: 0x00000000

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

Bit	Flag	Bit Name	Function	Read and write
b31-b5	Reserved	-	Read at "0", write at "0"	R/W
b4-b0	MD_PIN [4:0]	MD pin	Read the value of the PRTADR pin	R

35.5.11 Time-out Register (MDTO)

Offset address: 0x28

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
b23	b22	b21	b20	b19	b18	b17	b16
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	TOEN
b15	b14	b13	b12	b11	b10	b9	b8
TOCNT [15:8]							
0	0	0	0	0	0	0	0
b7	b6	b5	b4	b3	b2	b1	b0
TOCNT [7:0]							
0	0	0	0	0	0	0	0

Bit	Flag	Bit Name	Function	Read and write
b31-b17	Reserved	-	Read at "0", write at "0"	R/W
b16	TOEN	Timeout Enable	0: MDC low level When timeout detected clock function disabled 1: The timeout function of detecting MDC level is enabled	R/W
b15-b0	TOCNT [15:0]	Count value	Timeout count value MDIO timeout = TOCNT*PCLK	R/W

35.6 Precautions

1. The system clock is more than four times that of the MDC.
2. Overflow occurs when there is data not read from MDRXD. When receiving new data, the receive address frame is not affected by whether the receive data register is full.
3. If the MDSTA register is not read to clear the address match flag, both the address match and mismatch flags are 1 at the same time.

36 External Memory Controller EXMC

36.1 Summary

The external memory controller EXMC (External Memory Controller) is an independent module used to access various off-chip memories and realize data exchange. EXMC can convert the internal AMBA protocol interface into various types of dedicated off-chip memory communication protocol interfaces through configuration, including SRAM, PSRAM, NOR Flash, etc.

36.2 Basic Features

36.2.1 Features List

The basic function list of EXMC is as follows.

Table 36-1 Basic Functions of EXMC

Supported external Memory Type	• SRAM
	• PSRAM
	• NOR Flash
Interface Protocol	• Support AHB protocol and various external memory interface conversion
Interface Timing	• Programmable interface timing for various types of memory
Bus width	• 16 -bit, 32 -bit MEM bus width
Bus multiplexing	• SRAM , NOR Flash and PSRAM support address line and data line multiplexing
Automatic segmentation	• AHB bus width does not match the external memory width, it supports automatic segmentation and byte selection control

36.2.2 Controller Architecture

Figure 36-1 is a basic EXMC architecture block diagram. The SMC external memory controller independently generates an interface of the corresponding protocol and sends it to the port.

Note: The SRAM/PSRAM/NOR Flash controller is defined as SMC (Static Memory Controller) in the architecture diagram.

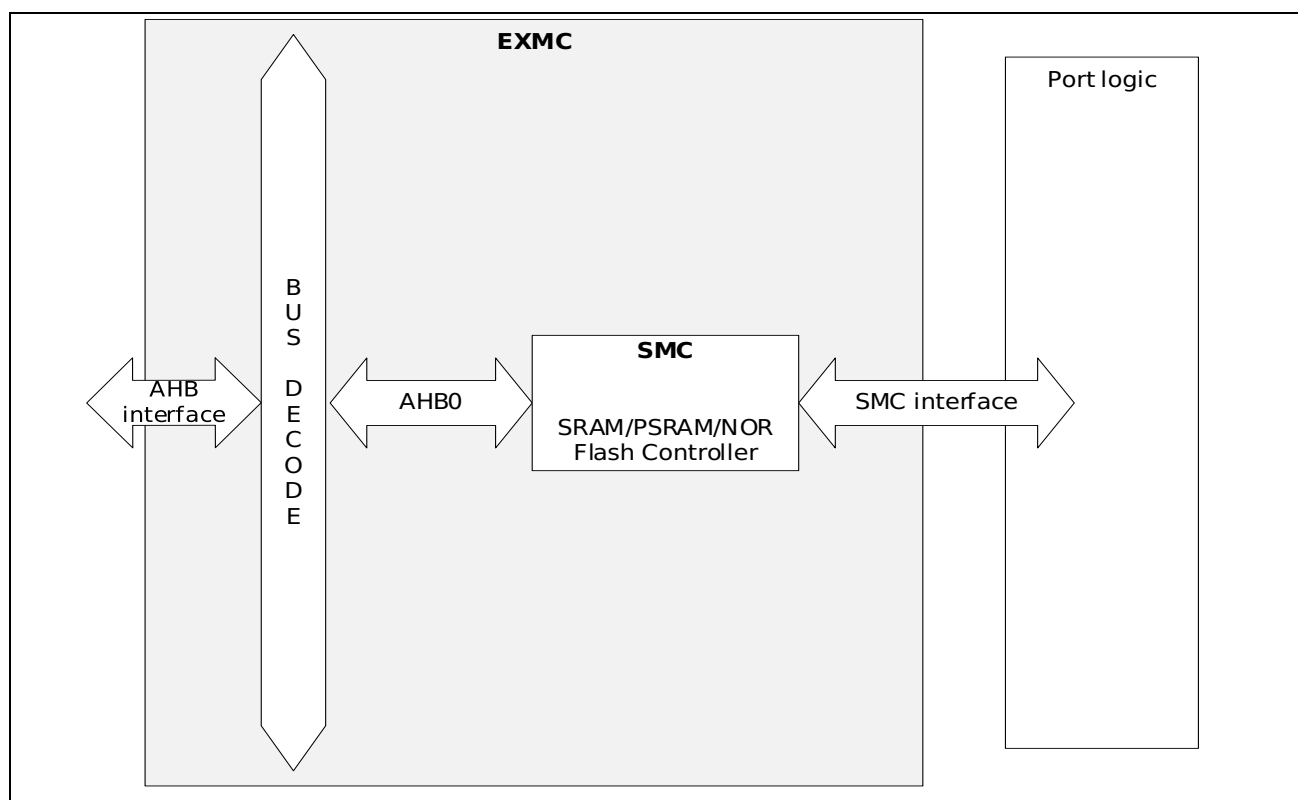


Figure 36-1 EXMC Architecture Diagram

36.2.3 Basic Access Specification

EXMC is the conversion interface between AHB bus protocol and external memory protocol. During the data transmission process of SMC , the AHB data width and the memory data width may be different. To ensure the consistency of data transmission, read and write access must comply with the following specifications:

- AHB access width is equal to memory width, data is accessed normally
- The AHB access width is larger than the memory width, and the AHB access is automatically divided into several consecutive memory data width transmissions.
- The AHB access width is smaller than the memory width, and the corresponding byte is accessed through the byte selection control signal SMC_BLS [3:0]

The specific access methods between AHB access width and various data bit width memories are as follows figure Table .

Table 36-2 AHB Access Width And Memory Bit Width Corresponding Access Mode Table

R/W	AHB access width	External memory width	Note
R	8	16	-
	16		-
	32		Converted into 2 EXMC read operations
	8	32	-
	16		-
	32		-
W	8	16	Use byte control signal BLS [0]
	16		-
	32		Converted into 2 EXMC write operations
	8	32	Use byte control signal BLS [0]
	16		Use byte control signal BLS [1:0]
	32		-

Note: The above table is only applicable to SMC data access

36.2.4 Address Mapping

This product defines an external memory access area with a total size of 1GB , which is used for different types of external memory and internal data exchange, including EXMC and QSPI. The 1GB space is allocated according to the Figure 36-2 scheme, the data access of SRAM/PSRAM/NOR Flash (SMC) maps 512MB of address space, and the control and data access of QSPI maps 128MB of address space.

Note: The access control of the address space 0x98000000~0x9FFFFFFF is implemented by the QSPI module. For details, please refer to the [Quad-wire Serial Peripheral Interface (QSPI)] section.

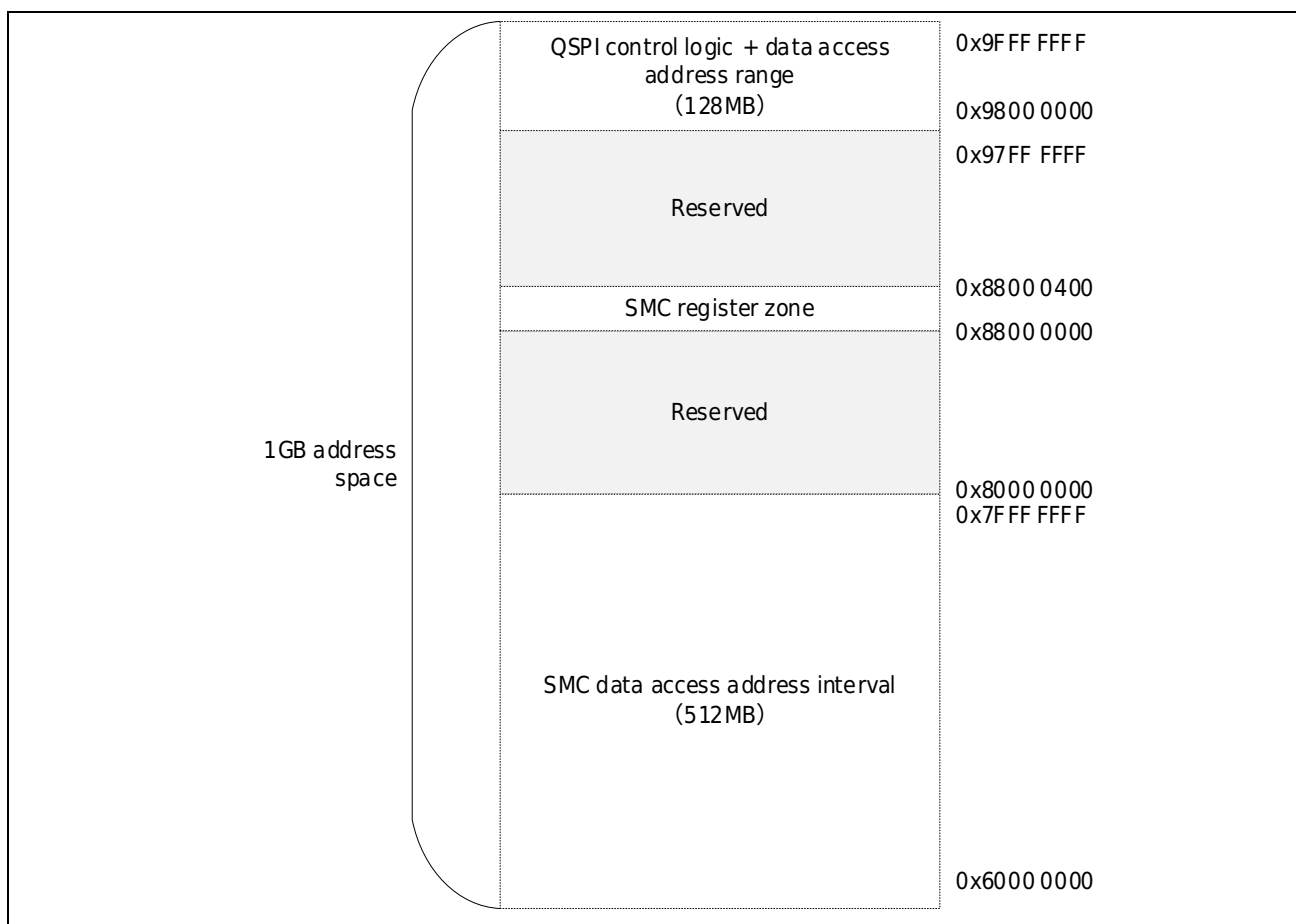


Figure 36-2 External Space Address Allocation

The address space of SMC can be divided into four chips with programmable sizes. The size of each chip can be divided by setting the chip select control register (SMC_CSCR0~1). The external memory objects corresponding to each chip can be configured independently, and have different CS (Chip Select) signals output to the external port. Figure 36-3 It is the Chip Select division of SMC. When the address space accessed by the CPU is in the CS space of a certain SMC , the corresponding SMC_CS signal output becomes a valid level.

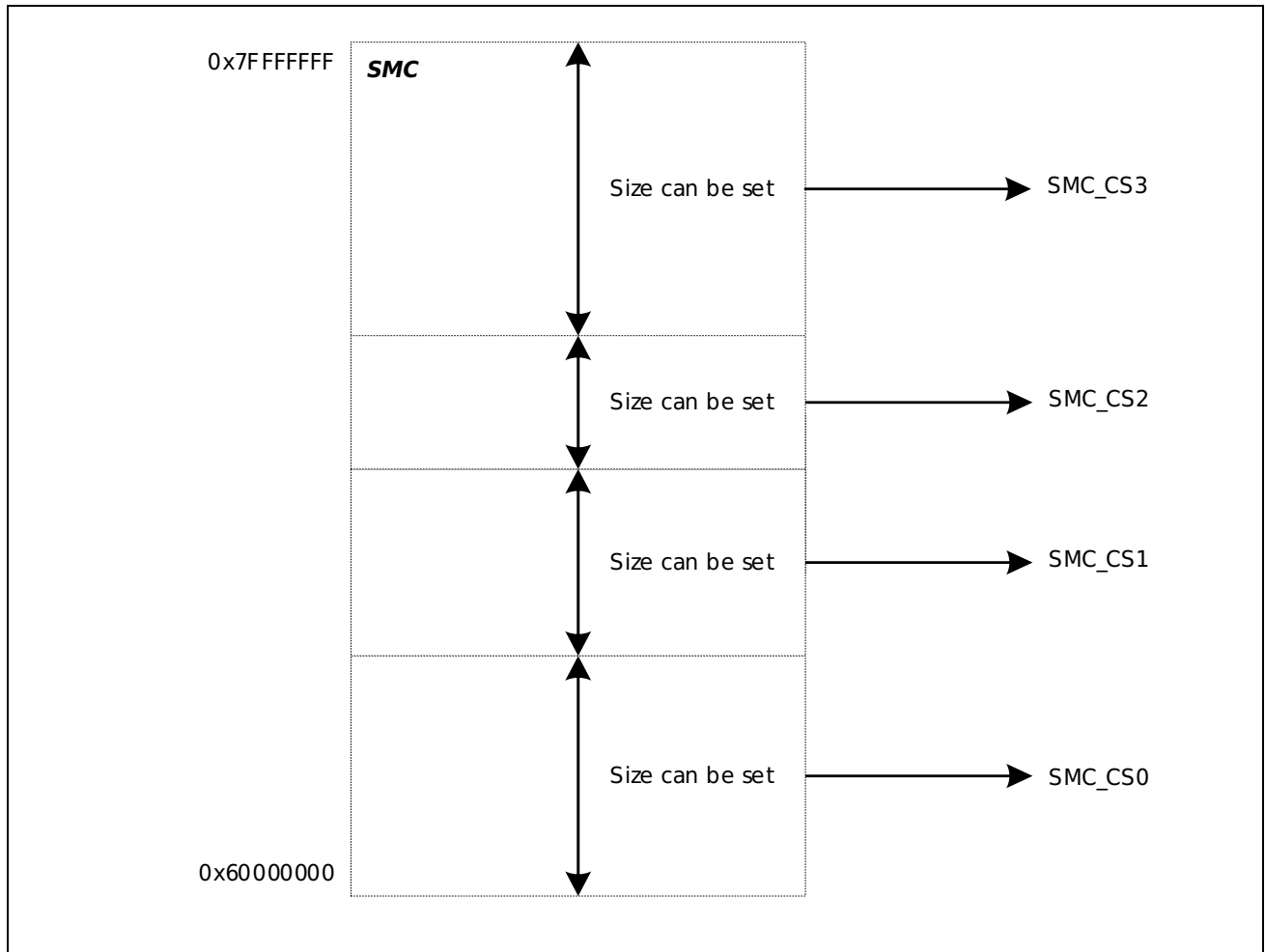


Figure 36-3 SMC Address Space Division

36.2.5 Protocol Interface

The protocol interface signal of the memory corresponding to SMC. EXMC exchanges data with external memory through the interface protocol implemented by SMC. The following is a description of the protocol interface required by the SMC storage.

36.2.5.1 SMC Protocol Interface

The following table shows the protocol interface required for SRAM/PSRAM/NOR Flash Control (SMC) access.

Table 36-3 SMC Protocol Interface

Protocol Interface Name	Direction	Effective level	Functional Description
SMC_CLK	O	-	SMC clock output
SMC_ADD [29:0]	O	-	SMC address output
SMC_DATA [31:0]	IO	-	SMC Data
SMC_WE	O	L	SMC write enable
SMC_OE	O	L	SMC output enable
SMC_CS [3:0]	O	L	SMC chip select signal
SMC_BLS [3:0]	O	L	Byte strobe signal of SMC
SMC_ADV	O	L	SMC address latch signal
SMC_CRE	O	H	SMC configuration register mode signal
SMC_WAIT	I	L	SMC input wait signal
SMC_BAA	O	L	SMC address prompt signal

Based on the above SMC external memory interface signal table, the chip's EXMC port functions are configured as shown in the following table.

Table 36-4 EXMC Port Function Assignment

Pin Name	Direction	Effective level	Functional Description
			SMC
EXMC_CLK	O	H/L	SMC_CLK
EXMC_ADD29~0	O	H/L	SMC_ADD [29:0]
EXMC_DATA31~0	IO	H/L	SMC_DATA [31:0]
EXMC_WE	O	L	SMC_WE
EXMC_CE7~0	O	L	SMC_BLS [3:0]
	O	L	SMC_CS [3:0]
EXMC_OE	O	L	SMC_OE
EXMC_BAA	O	L	SMC_BAA
EXMC_ADV	O	L	SMC_ADV
EXMC_ALE	O	H	SMC_CRE
EXMC_CLE	O	H	-

Pin Name	Direction	Effective level	Functional Description
			SMC
EXMC_RB7~0	I	L	-
	I	L	SMC_WAIT

36.3 Functional Description

36.3.1 SMC-SRAM/PSRAM/NOR Flash Controller

36.3.1.1 Introduction to SRAM/PSRAM/NOR Flash

SRAM/NOR Flash are both memories with static access functions, which can save the data stored inside without refreshing the circuit. The width of its address line determines the number of storage cells. The space corresponding to the memory storage width on each storage cell is used to store data. Therefore, the capacity of SRAM/PSRAM/NOR Flash memory is the memory data width $\times 2^{\text{address line width}}$.

PSRAM is a pseudo static random access memory. Since it has an SRAM interface, it can exchange data with the SMC module inside the system through EXMC , just like SRAM.

36.3.1.2 SMC Basic Functions

of SMC are as follows:

- Support 16 -bit and 32 -bit external memory data width
- AHB word, half-word, byte access
- Provide independent chip select control for each memory chip
- Byte select signal output
- Address line and data line multiplexing
- Programmable protocol timing parameters
- Programmable rate automatic refresh operation (for PSRAM)
- With two 47 - bit command FIFOs
- With four 36 - bit write data FIFOs
- With four 32 - bit read data FIFOs
- Low power management

36.3.1.3 SMC Initial Setting

SMC after power-on reset and the switching between states are shown as follows in Figure 36-4.

At the initial reset, the SMC is in Low Power state; through the setting of the status control register (SMC_STCR0 and SMC_STCR1), the SMC can switch between the Ready state and the Low Power state. For details, please refer to the [SMC Low Power Management] section.

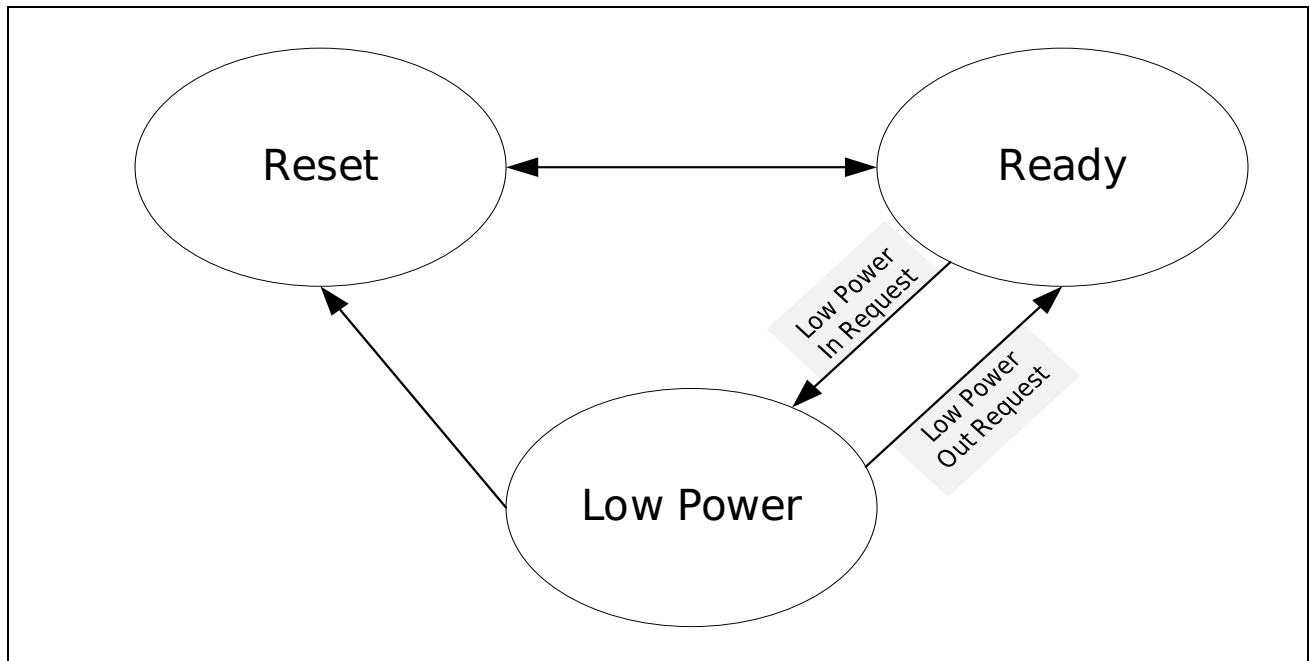


Figure 36-4 SMC Status Diagram

Before the SMC communicates with external SRAM, PSRAM, NOR Flash, etc., it must perform initial settings and configure related parameters to ensure correct data transmission. Please refer to the figure below for the specific setting sequence.

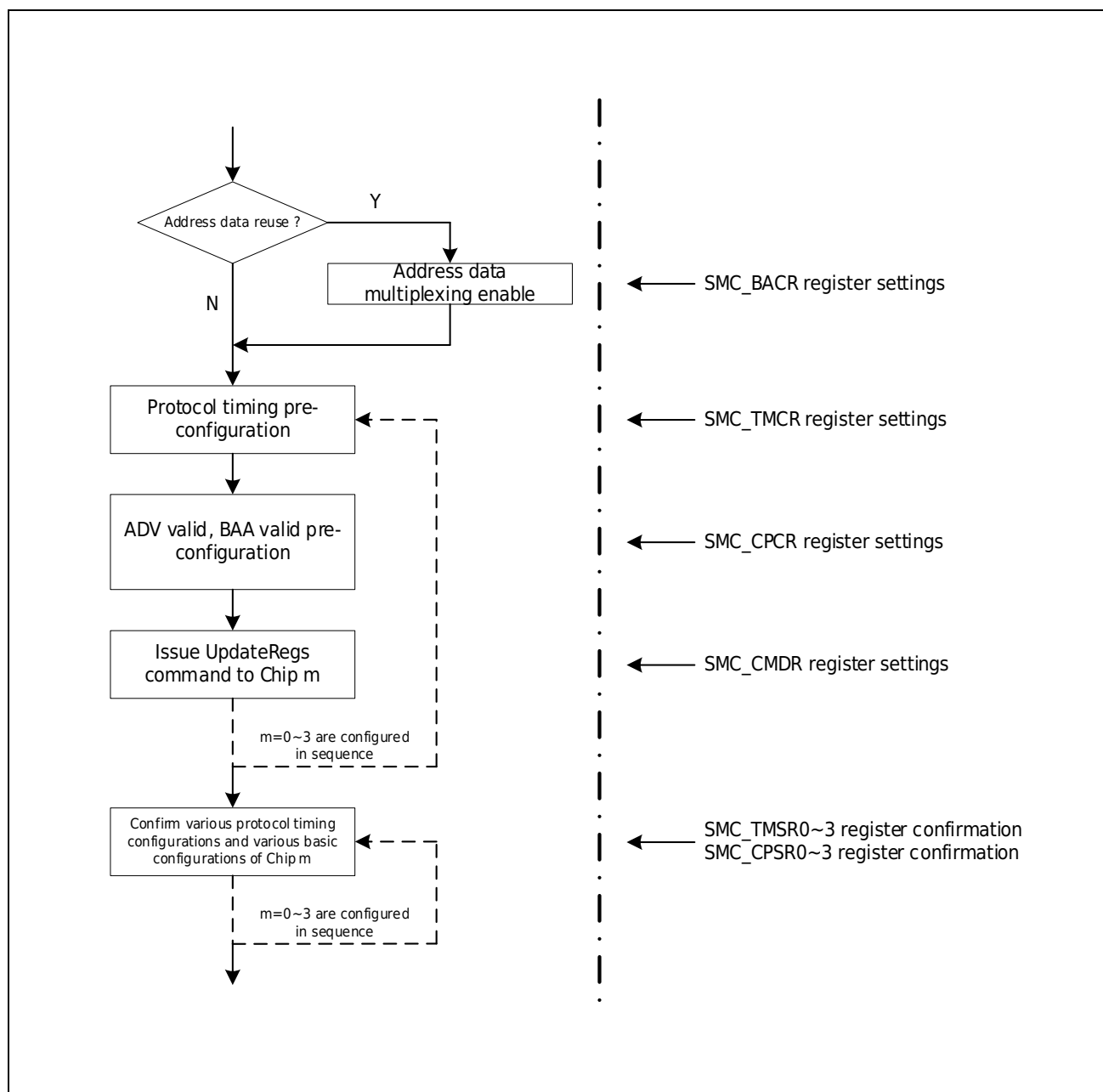


Figure 36-5 SMC Initial Setting Process

36.3.1.4 SMC Access Actions

Access Method

SMC can choose synchronous access mode or asynchronous access mode to read and write external memory.

The synchronous access mode corresponds to the synchronous access type of external memory. When the SMC outputs read and write control signals, the clock signal is also output at the same time. The external memory transmits data with the SMC according to the clock output by the SMC, that is, SMC_CLK is valid; the asynchronous access mode corresponds to the asynchronous access type of external memory. When the SMC outputs read and write control signals, the clock signal is not output (the output is always low), and the external memory implements data access based on other control signals.

In synchronous access mode, the SMC and external memory use the SMC_WAIT signal to control read and write operations. When the SMC_WAIT signal is input at a low level, the SMC is in a waiting state and does not perform any operations. The SMC performs normal read and write access only after SMC_WAIT becomes a high level.

By setting the RSYN and WSYN bits of the CHIP configuration register (SMC_CPCR), you can choose synchronous read and write mode or asynchronous read and write mode.

Access Timing

SMC supports access to multiple types of memory and can convert single read and write operations of AHB into read and write operations of memory. The following are some timing instances of data reading and writing between the controller and external memory:

- Single read action
- Single read operation with multiplexing of address and data lines
- Single write action
- Single write operation with multiplexing of address and data lines

Table 36-5 and Figure 36-6~Figure 36-9 are the basic timing diagram and setting instance of a single read operation.

Table 36-5 Basic Setting Instance Of Single Read Operation

Basic settings	MW	RSYN	WSYN	BAA	ADV	BLS	- means "not paying attention" <set> indicates user-set value
	<set>	<set>	-	-	0b0	0b0	
Timing settings	t_rc	t_wc	t_ceoe	t_wp	t_pc	t_tr	
	0b0011	-	0b001	-	-	-	

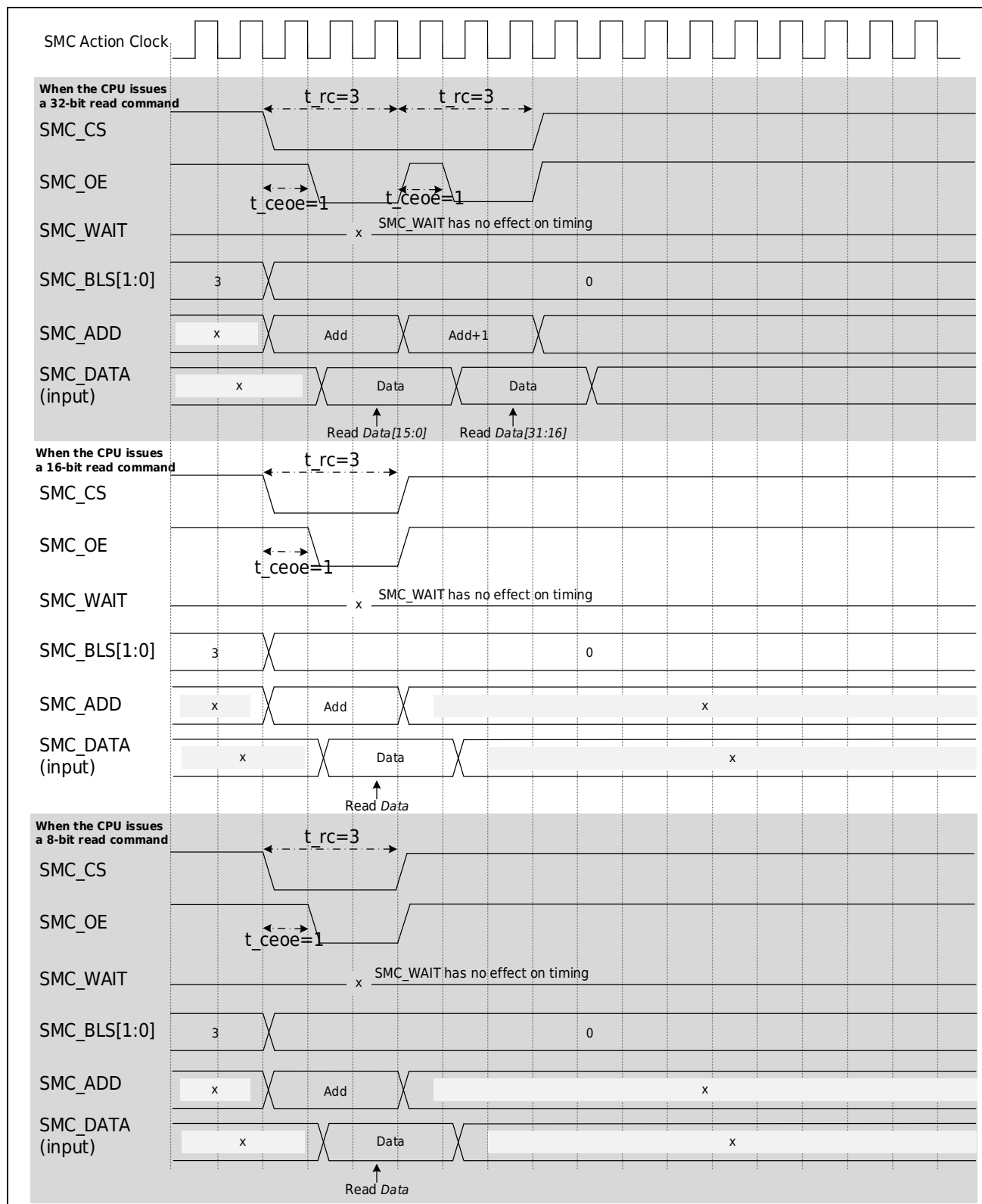


Figure 36-6 Basic Timing Of Single Read Operation (Asynchronous Mode (RSYN=0) & 16 -Bit Width (MW=01))

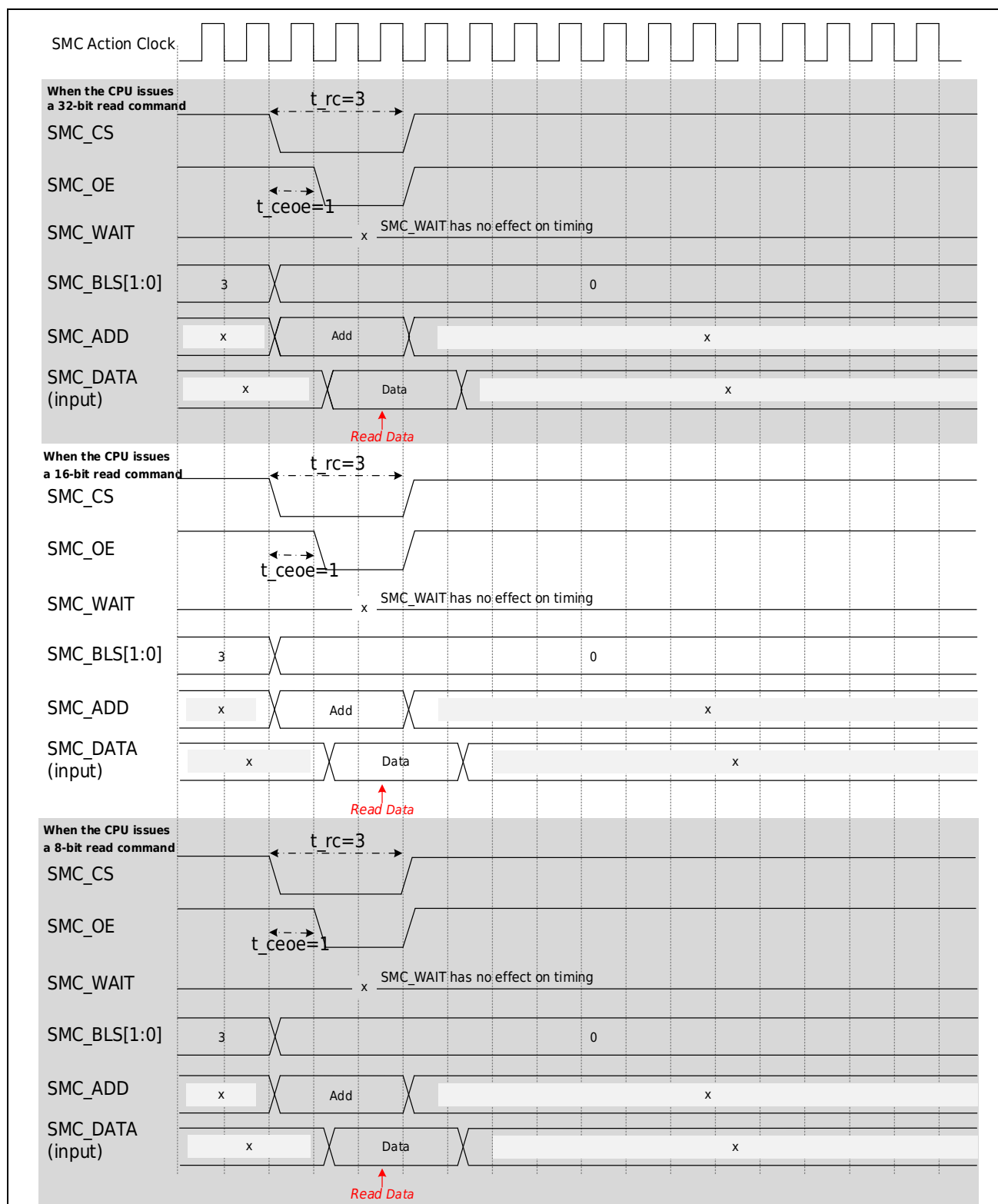


Figure 36-7 Basic Timing Of Single Read Operation (Asynchronous Mode (RSYN=0) & 32 -Bit Width (MW=10))

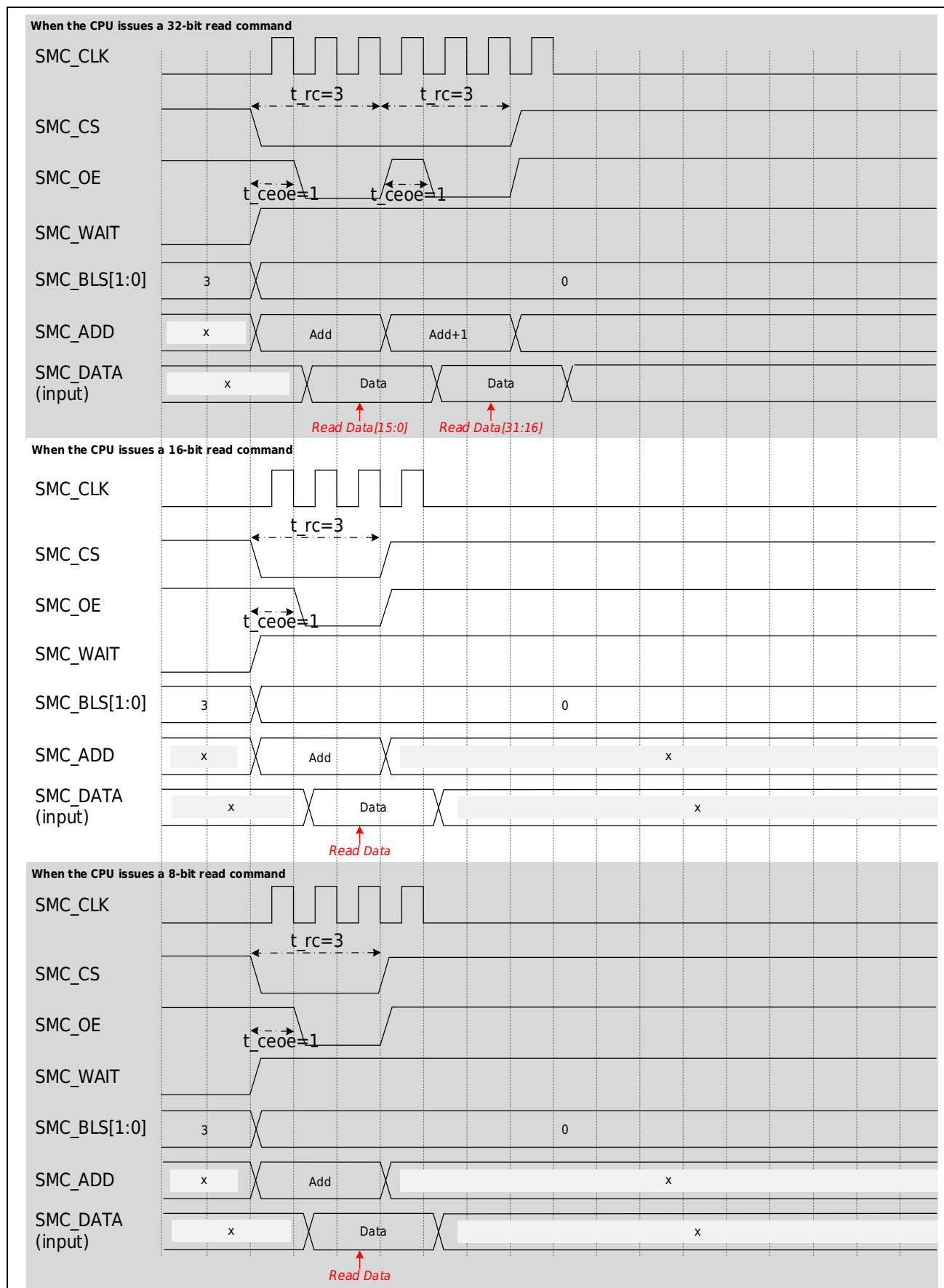


Figure 36-8 Basic Timing Of Single Read Operation (Synchronous mode (RSYN=1) & 16-bit width (MW=01))

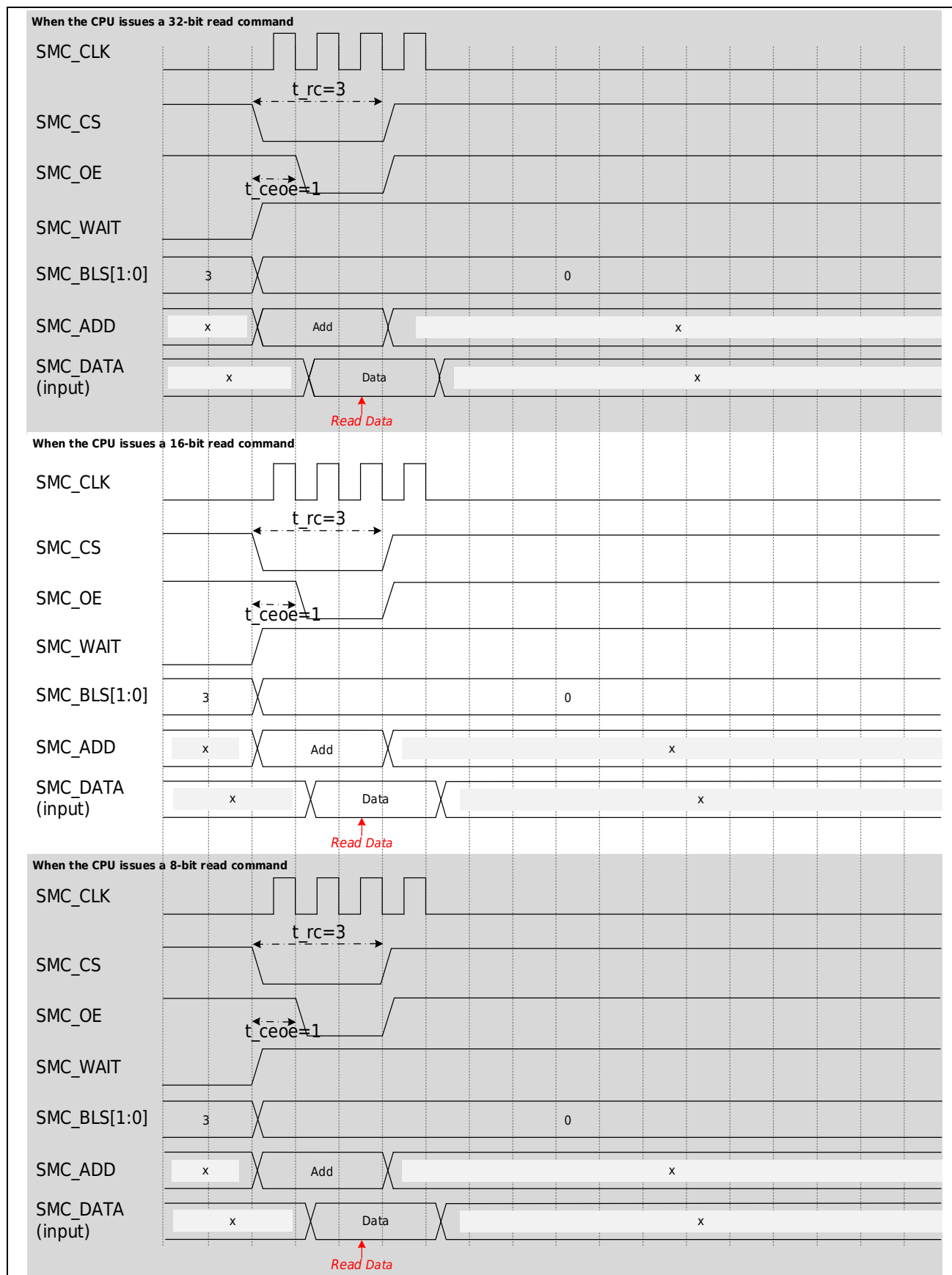


Figure 36-9 Basic Timing Of Single Read Operation (Synchronous mode (RSYN=1) & 32-bit width (MW=10))

Table 36-6 and Figure 36-10~ are the basic timing diagram and setting instance of a single read operation with multiplexing of address and data lines.

Table 36-6 Instance Of Basic Setting For Single Read Operation Using Multiplexed Address and Data Lines

Basic settings	MW	RSYN	WSYN	BAA	ADV	BLS	- means "not paying attention" <set> indicates user-set value
Timing settings	t_rc	t_wc	t_ceoe	t_wp	t_pc	t_tr	
	0b0110	-	0b011	-	-	0b010	

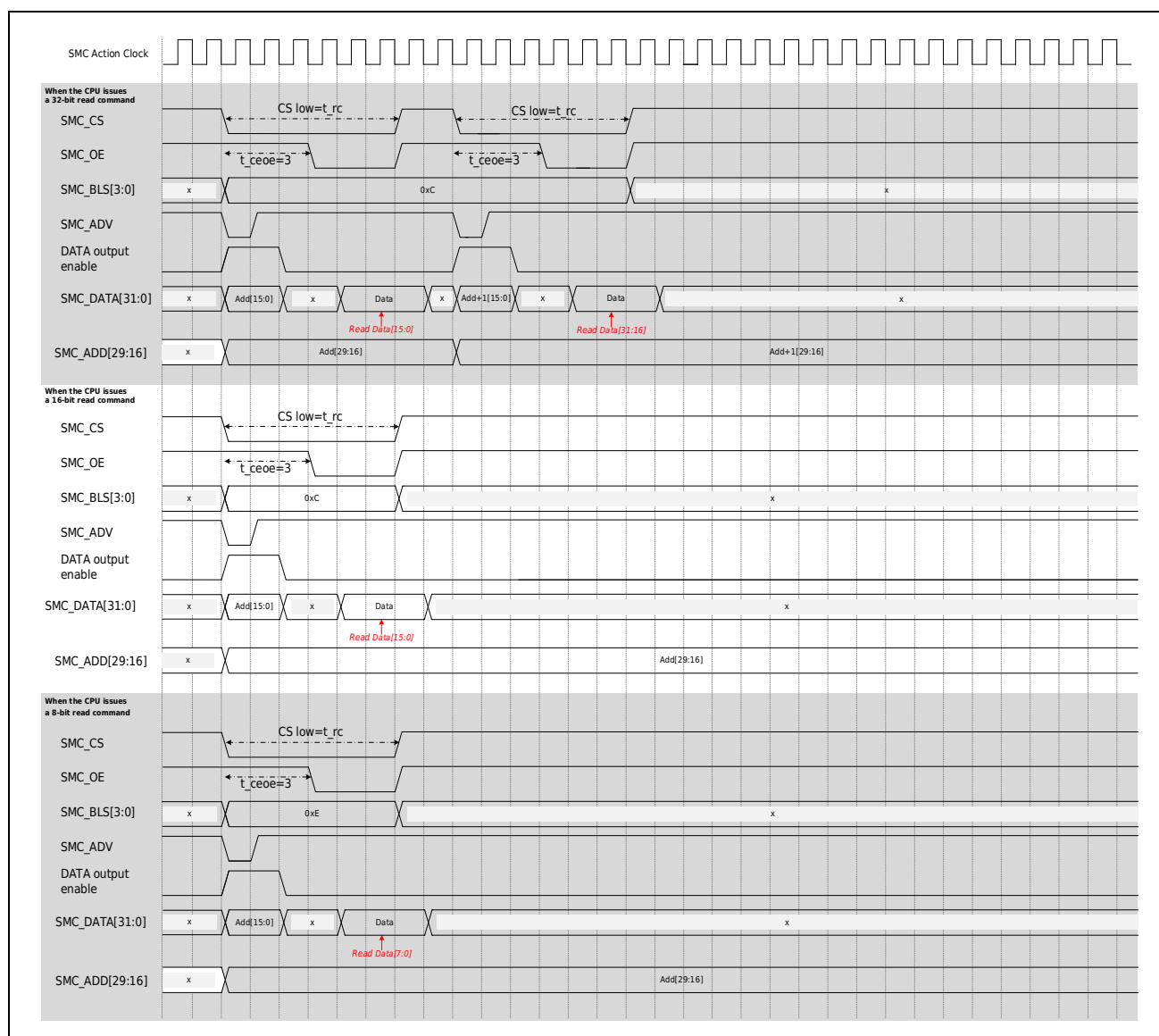


**Figure 36-10 Basic Timing Of Single Read Action With Multiplexing Of Address And Data Lines
(Asynchronous Mode (RSYN=0) & 32 -Bit Width (MW=10))**

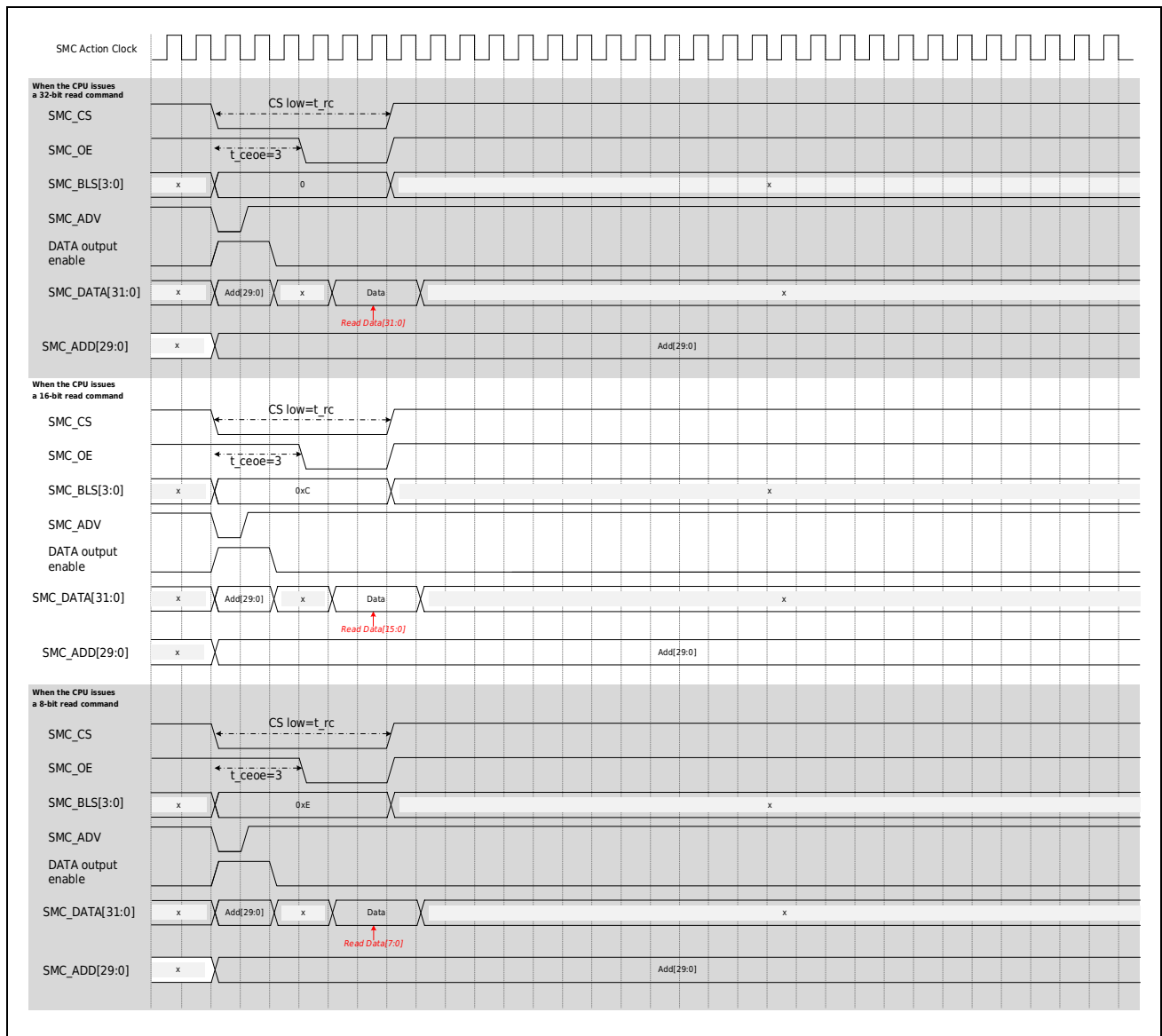


**Figure 36-11 Basic Timing Of Single Read Action With Multiplexing Of Address And Data Lines
(Asynchronous Mode (RSYN=0) & 32 -Bit Width (MW=10))**

Table 36-7 and Figure 36-12~Figure 36-15 are the basic timing diagram and setting instance of a single write operation.

Table 36-7 Basic Setting Instance Of Single Write Operation

Basic settings	MW	RSYN	WSYN	BAA	ADV	BLSS	- means "not paying attention" <set> indicates user-set value
	<set>	-	<set>	-	0b0	<set>	
Timing settings	t_{rc}	t_{wc}	t_{ceoe}	t_{wp}	t_{pc}	t_{tr}	
	-	0b0100	-	0b010	-	-	

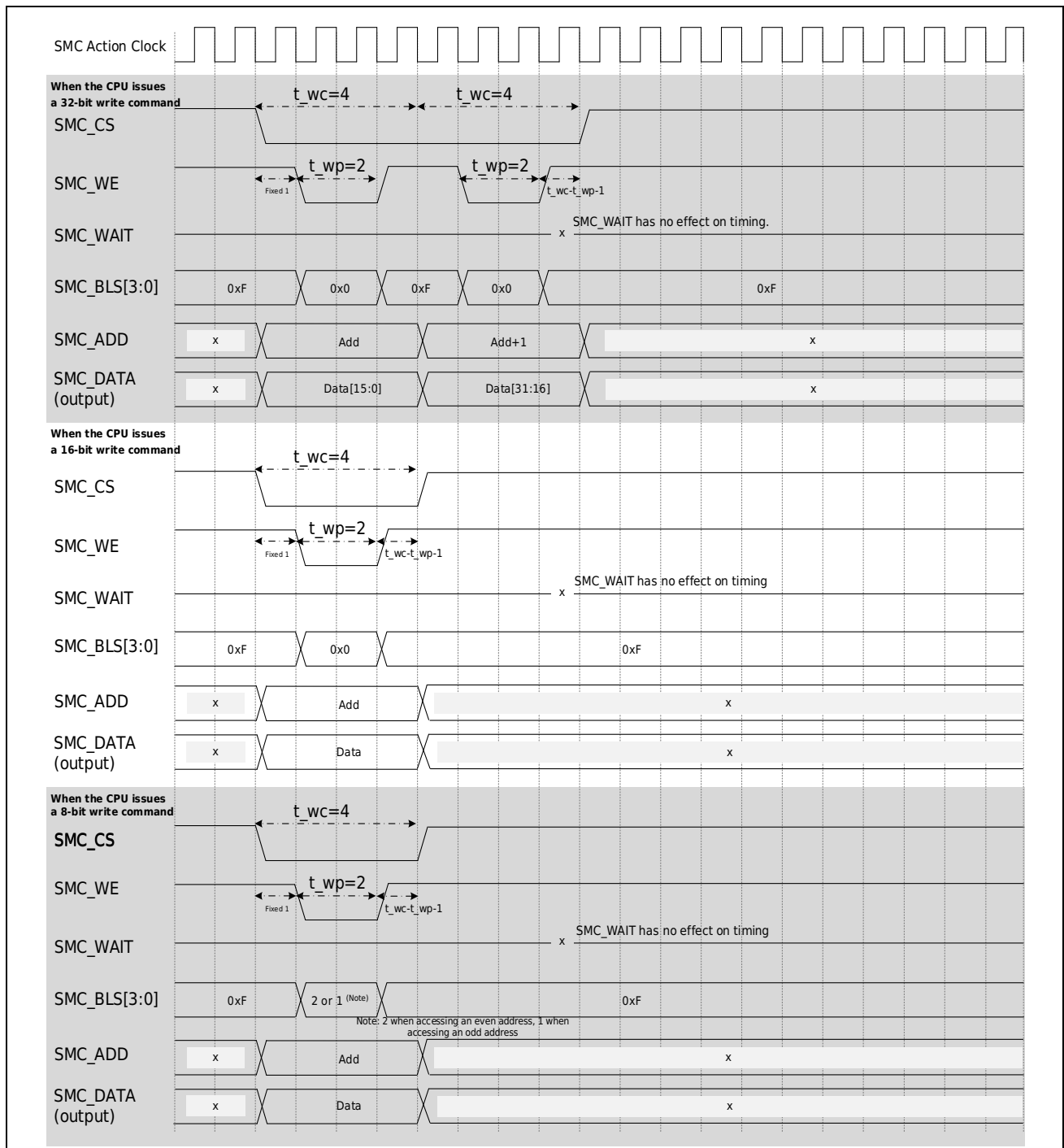


Figure 36-12 Basic Timing Of Single Write Operation (Asynchronous Mode (WSYN=0) & 16 -Bit Width (MW=01) & BLSS=1)

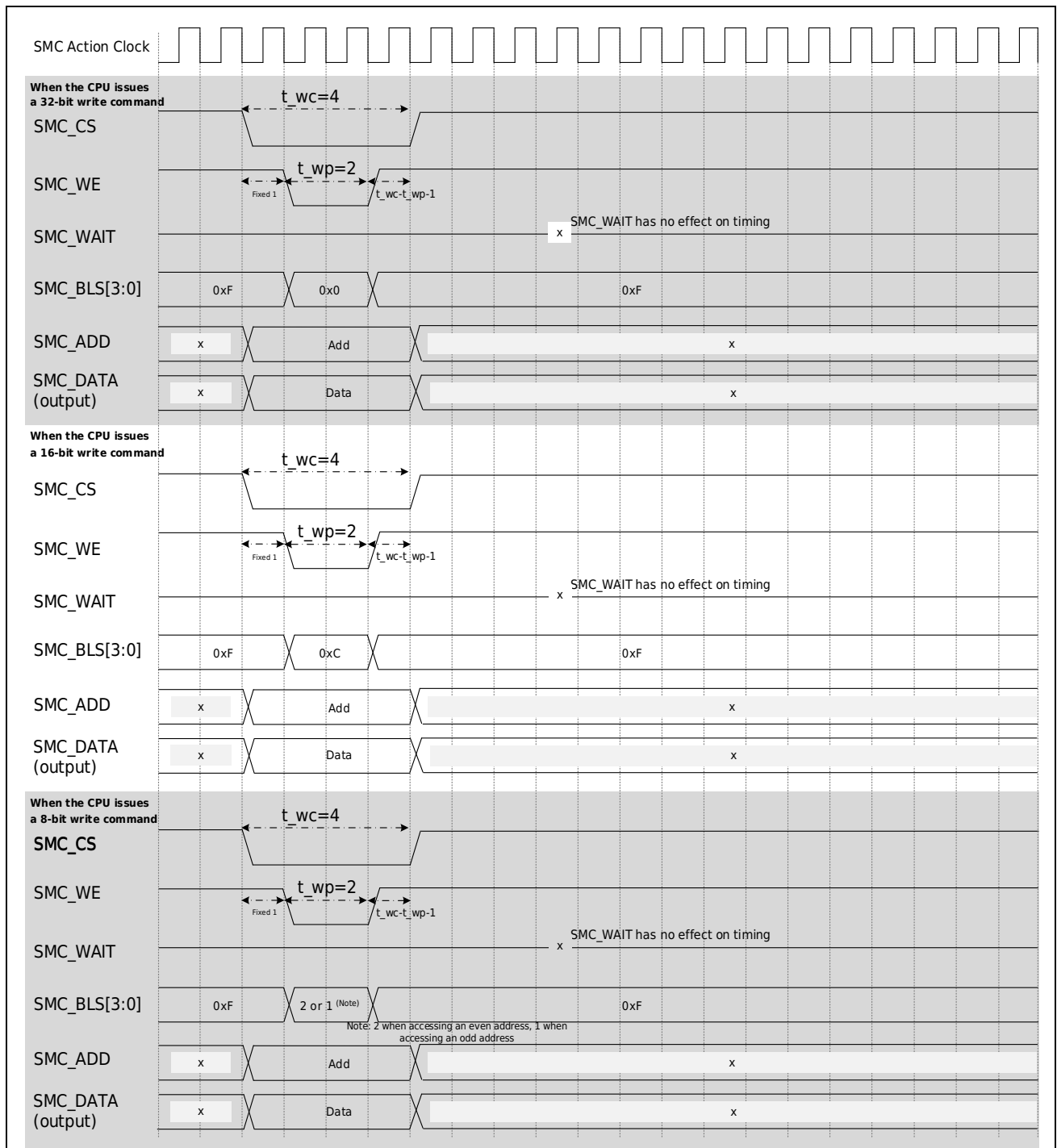


Figure 36-13 Basic Timing Of Single Write Operation (Asynchronous Mode (WSYN=0) & 32 -Bit Width (MW=10) & BLSS=1)

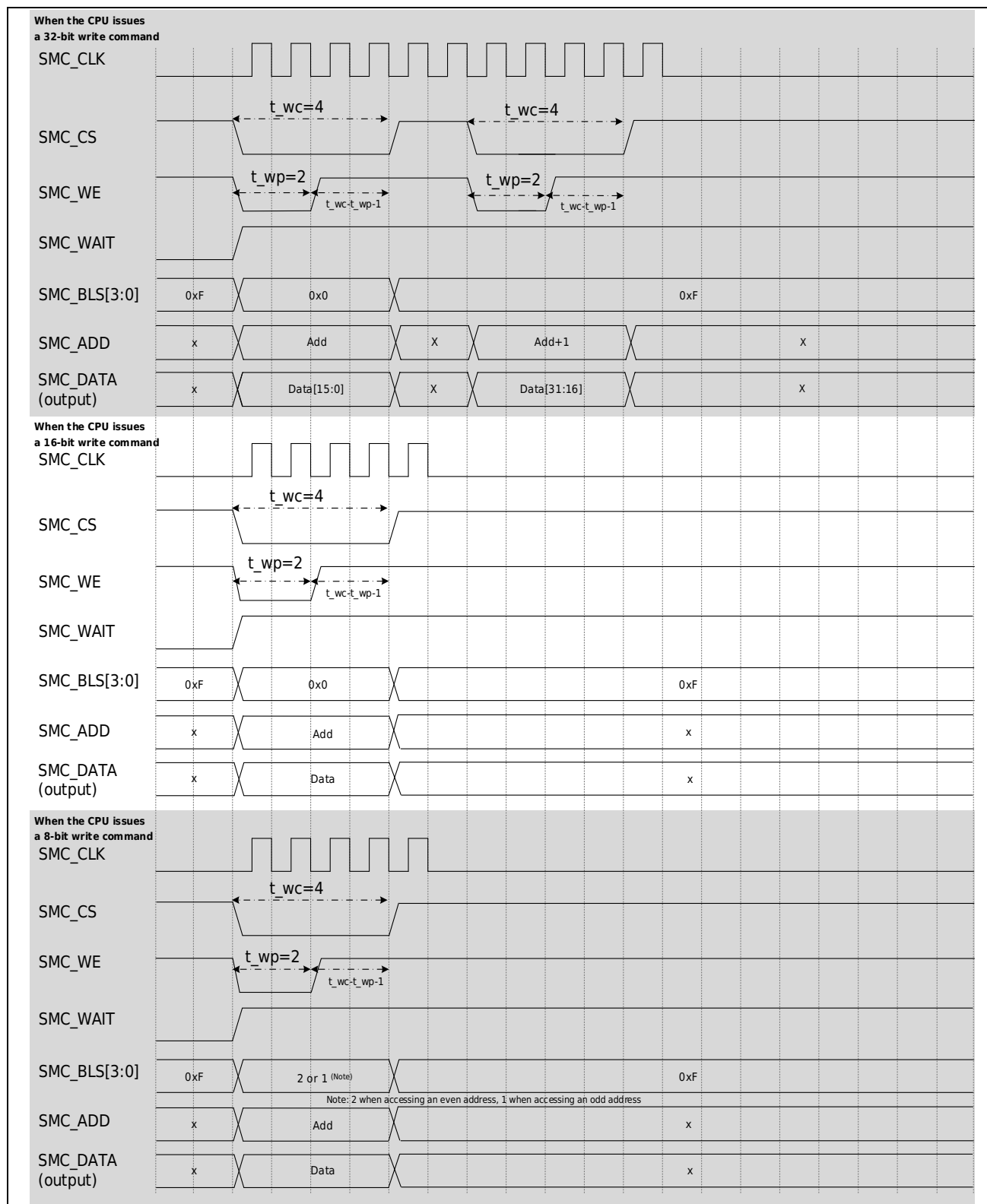


Figure 36-14 Basic Timing Of Single Write Operation (Synchronize Mode (WSYN=1) & 16 -Bit Width (MW=01) & BLSS=0)

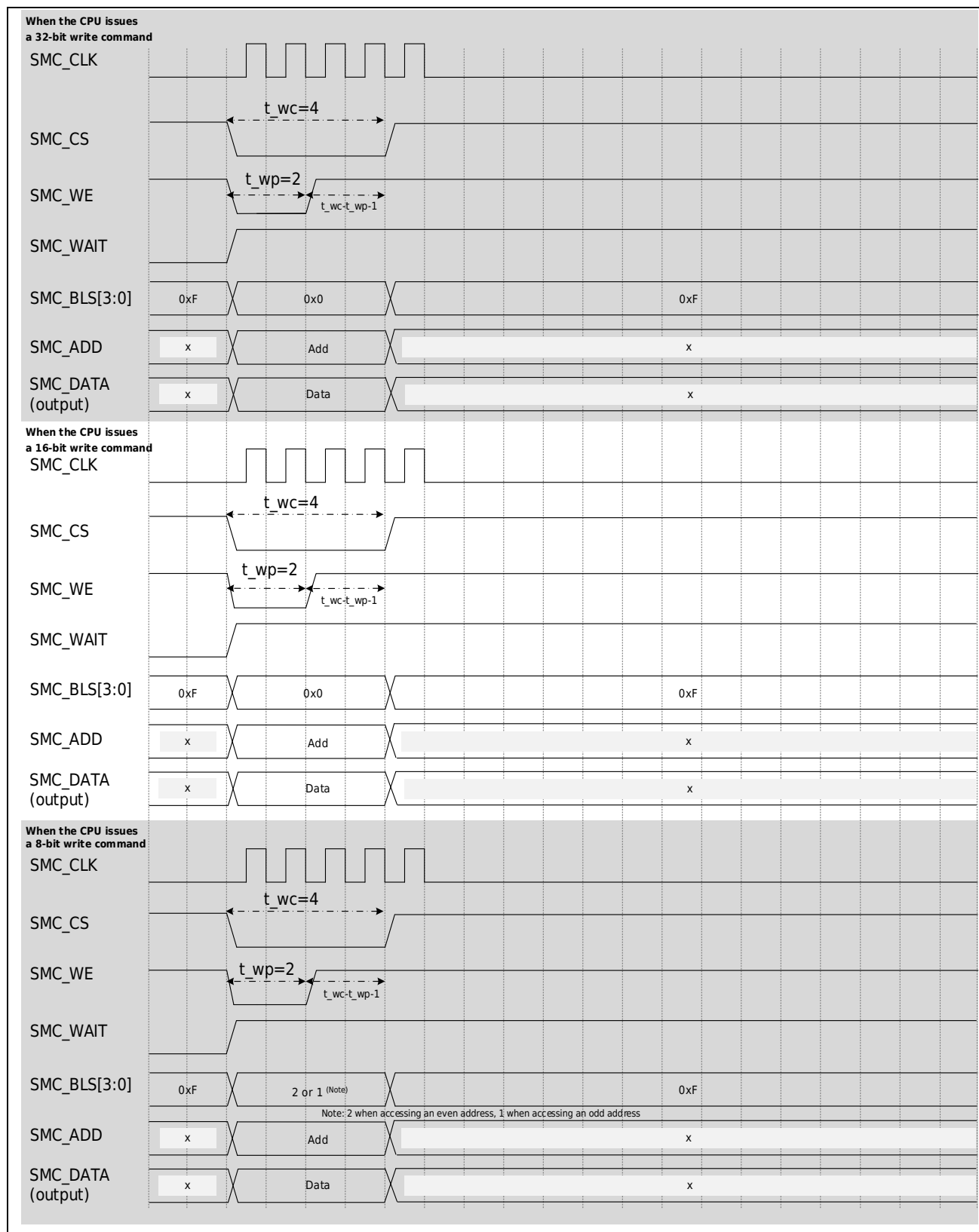


Figure 36-15 Basic Timing Of Single Write Operation (Synchronize Mode (WSYN=1) & 32 -Bit Width (MW=10) & BLSS=0)

Table 36-8 and Figure 36-16 are the basic timing diagram and setting instance of a single write operation with multiplexing of address and data lines.

Table 36-8 Instance Of Basic Setting For Single Write Operation Using Multiplexed Address And Data Lines

Basic settings	MW	RSYN	WSYN	BAA	ADV	BLS	- means "not paying attention" <set> indicates user-set value
	<set>	-	0	0b0	0b1	0b1	
Timing settings	t_rc	t_wc	t_ceoe	t_wp	t_pc	t_tr	
	-	0b0101	-	0b010	-	0b010	

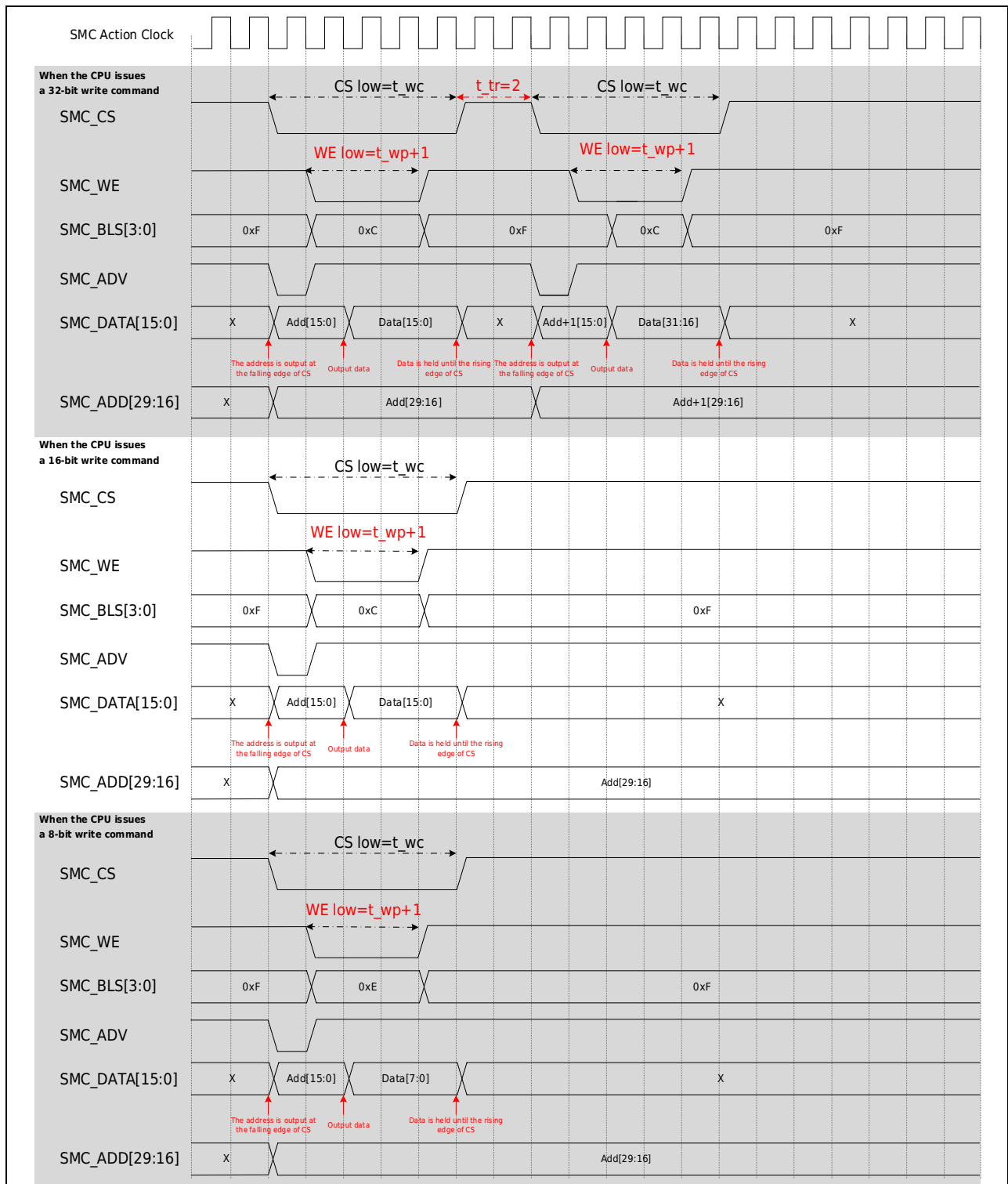


**Figure 36-16 Basic Timing Of Single Write Action With Multiplexing Of Address And Data Lines
(Asynchronous Mode (WSYN=0) & 16 -Bit Width (MW=01))**

Basic timing diagram and setting instance of single write operation in synchronous mode with multiplexing of Table 36-9 and Figure 36-17 address data lines.

Table 36-9 Basic Setting Instance Of Single Write Operation In Synchronous Mode With Multiplexed Address And Data Lines

Basic settings	MW	RSYN	WSYN	BAA	ADVS	BLSS	- means "not paying attention" <set> indicates user-set value
	<set>	-	1	0b0	0b1	0b1	
Timing settings	t_rc	t_wc	t_ceoe	t_wp	t_pc	t_tr	
	-	0b0101	-	0b0100	-	0b010	

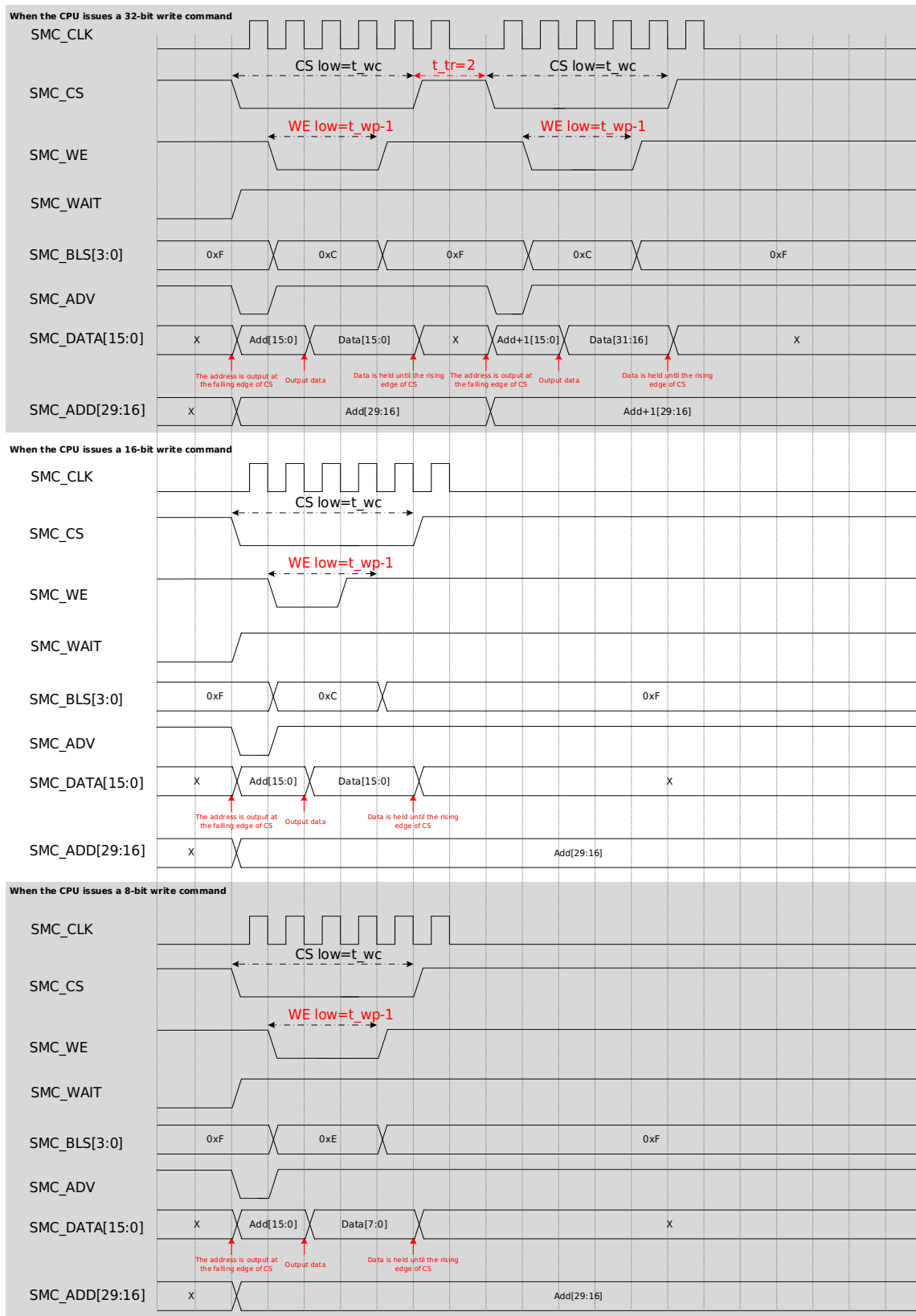


**Figure 36-17 Basic Timing Of Single Write Action With Multiplexing Of Address And Data Lines
(Synchronize Mode (WSYN=1) & 16 -Bit Width (MW=01))**

FIFO Management

SMC contains 3 types of FIFOs. They are command FIFO , write data FIFO , and read data FIFO.

Command FIFO : 2 command FIFOs , used to cache 2 SMC commands and corresponding command address, CS and other information.

Write data FIFO : 4 write data FIFOs are used to cache AHB write data.

Read data FIFO : 4 read data FIFOs are used to cache data read from SRAM/PSRAM/NOR Flash.

36.3.1.5 SMC Low Power Management

When the SMC is in the Low Power state, if the LPWOR bit of the status control register (SMC_STCR1) is set to 1, the SMC changes from the Low Power state to the Ready state, and the SMC starts to work normally; If there is no operation on the SMC for a long time, you can set the LPWIR bit of the status control register (SMC_STCR0) to 1 to make the SMC enter the Low Power state. After entering the Low Power state, the internal actions of the SMC stop and the power consumption will be reduced.

36.4 Register Description

Table 36-10 EXMC Register List

SMC: 0x88000000

Register Name		Symbol	Offset	Bit Width	Reset Value
SMC	Enable Register	SMC_ENAR	0x4005540C	32	0x00000000
	Status Register	SMC_STSR	0x0000	32	0x00000001
	Command Register	SMC_CMDR	0x0010	32	-
	Status Control Register 0	SMC_STCR0	0x0008	32	-
	Status Control Register 1	SMC_STCR1	0x000c	32	-
	Refresh Time Register	SMC_RFTR	0x0020	32	0x00000000
	Basic control register	SMC_BACR	0x0200	32	0x00000300
	Chip Select Control Register	SMC_CSCR0	0x0208	32	0xFFFFFFFF
	Chip Select Control Register	SMC_CSCR1	0x020C	32	0x00000000
	CHIP Configuration Register	SMC_CPCR	0x0018	32	-
	CHIP Status Register 0	SMC_CPSR0	0x0104	32	0x00FF0A00
	CHIP Status Register 1	SMC_CPSR1	0x0124	32	0x00FF0A00
	CHIP Status Register 2	SMC_CPSR2	0x0144	32	0x00FF0A00
	CHIP Status Register 3	SMC_CPSR3	0x0164	32	0x00FF0A00
	Timing Configuration Register	SMC_TMCR	0x0014	32	-
	Timing status register 0	SMC_TMSR0	0x0100	32	0x001263CC
	Timing status register 1	SMC_TMSR1	0x0120	32	0x001263CC
	Timing status register 2	SMC_TMSR2	0x0140	32	0x001263CC
	Timing status register 3	SMC_TMSR3	0x0160	32	0x001263CC

36.4.1 SMC-SRAM/PSRAM/NOR Flash Register

36.4.1.1 Enable Register (SMC_ENAR)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved														SMCEN	-
Bit	Flag	Bit Name	Function											Read and write	
b31~b2	Reserved	-	Read as "0", write as "0"											R/W	
b1	SMCEN	SMC enable	0: The function port of the SMC module is invalid 1: SMC module function port enabled											R/W	
b0	Reserved	-	Read as "0", write as "0"											R/W	

36.4.1.2 Status Register (SMC_STSR)

Reset value: 0x00000001

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved														STATUS	
Bit	Flag	Bit Name	Function											Read and write	
b31~b1	Reserved	-	Reset value when read											R	
b0	STATUS	SMC Current Status	0: Ready 1: Low Power entered											R	

36.4.1.3 Command Register (SMC_CMDR)

Reset value: -

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	
Reserved						CMDCHIP [2:0]			CMD [1:0]		CRES	CMDADD [19:0]				
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
CMDADD [15:0]																
Bit		Flag		Bit Name		Function									Read and write	
b31~b26		Reserved		-		Write "0" on write									W	
b25~b23		CMDCHIP [1:0]		CS object setting		Chip object of CMD command									W	
						000: Chip 0										
						001: Chip 1										
						010: Chip 2										
						011: Chip 3										
b22~b21		CMD [1:0]		Command Input		Please do not set other values									W	
						00: Invalid										
						01: MdRegConfig										
						10: UpdateRegs										
						11: MdRegConfig & UpdateRegs										
b20		CRES		CRE Port Settings		0: SMC_CRE outputs low level 1 : When issuing the MdregConfig command, the SMC_CRE port outputs a high level									W	
b19~b0		CMDADD [19:0]		Address object settings		When CMD=01 or 11 , CMDADD [19:0] indicates the external MEM address to be accessed.									W	

36.4.1.4 Status Control Register (SMC_STCR0)

Reset value: -

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved												LPWIR	-	-	
Bit	Flag	Bit Name	Function											Read and write	
b31~b3	Reserved	-	Write "0" on write											W	
b2	LPWIR	Low Power State Entry Request	0: No request 1: Low Power state is entered											W	
b1~b0	Reserved	-	Write "0" on write											W	

36.4.1.5 Status Control Register (SMC_STCR1)

Reset value: -

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved												LPWOR	-	-	
Bit	Flag	Bit Name	Function											Read and write	
b31~b3	Reserved	-	Write "0" on write											W	
b2	LPWOR	Low Power state exit request	0: No request 1: Low Power state is exited											W	
b1~b0	Reserved	-	Write "0" on write											W	

36.4.1.6 Refresh Time Register (SMC_RFTR)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved												REFPRD [3:0]			
Bit		Flag		Bit Name		Function								Read and write	
b31~b4		Reserved		-		Read as "0", write as "0"								R/W	
b3~b0		REFPRD [3:0]		Refresh period		Refresh period value setting								R/W	

36.4.1.7 Basic Control Register (SMC_BACR)

Reset value: 0x00000300

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CKSEL [1:0]		Reserved									MUXMD		Reserved		
Bit		Flag		Bit Name				Function						Read and write	
b31~b16		Reserved		-				Read as "0", write as "0"						R/W	
b15~b14		CKSEL [1:0]		Sampling Clock Selection				00: Use internal EXCLK when reading latched SMC_DATA and SMC_WAIT port data. 01: When reading the latched SMC_DATA and SMC_WAIT port data, use the internal EXCLK inverted as the clock 10 : When reading the latched SMC_DATA and SMC_WAIT port data, use the EXMC_CLK port status feedback as the clock 11: Disable setting						R/W	
b13~b10		Reserved		-				Read as "0", write as "0"						R/W	
b9~b8		Reserved		-				Read as "1", write as "1"						R/W	
b7~b5		Reserved		-				Read as "0", write as "0"						R/W	
b4		MUXMD		SMC address data multiplexing				0: SMC address and data are not reused 1: SMC address and data multiplexing						R/W	
b3~b0		Reserved		-				Read as "0", write as "0"						R/W	

36.4.1.8 Chip Select Control Register (SMC_CSCR0)

Reset value: 0xFFFFFFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
ADDMSK3 [7:0]								ADDMSK2 [7:0]							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ADDMSK1 [7:0]								ADDMSK0 [7:0]							
Bit	Flag		Bit Name		Function		Read and write								
b31~b24	ADDMSK3 [7:0]		CS3 space mask address		CS3 space mask address setting		R/W								
b23~b16	ADDMSK2 [7:0]		CS2 space mask address		CS2 space mask address setting		R/W								
b15~b8	ADDMSK1 [7:0]		CS1 space mask address		CS1 space mask address setting		R/W								
b7~b0	ADDMSK0 [7:0]		CS0 space mask address		CS0 space mask address setting		R/W								

36.4.1.9 Chip Select Control Register (SMC_CSCR1)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
ADDMAT3 [7:0]								ADDMAT2 [7:0]							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ADDMAT1 [7:0]								ADDMAT0 [7:0]							
Bit	Flag		Bit Name		Function		Read and write								
b31~b24	ADDMAT3 [7:0]		CS3 space matching address		CS3 space matching address setting		R/W								
b23~b16	ADDMAT2 [7:0]		CS2 space matching address		CS2 space matching address setting		R/W								
b15~b8	ADDMAT1 [7:0]		CS1 space matching address		CS1 space matching address setting		R/W								
b7~b0	ADDMAT0 [7:0]		CS0 space matching address		CS0 space matching address setting		R/W								

36.4.1.10 CHIP Configuration Register (SMC_CPCR)

Reset value: -

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-			BLSS	ADVS	BAAS	MW [1:0]		-			WSYN	-		RSYN	

Bit	Flag	Bit Name	Function	Read and write
b31~b16	Reserved	-	Write "0" on write	W
b15~b13	Reserved	-	Write "0" on write	W
b12	BLSS	BLS Port Settings	0: The BLS port status changes synchronously with the CS port status 1: The BLS port status changes synchronously with the WE port status	W
b11	ADVS	ADV port settings	0: ADV port function is invalid 1 : When the address data line multiplexing function is valid, the ADV port function is valid	W
b10	BAAS	BAA Port Settings	0: BAA port function is invalid 1: BAA port function is valid	W
b9~b8	MW [1:0]	Memory width selection	01: 16-bit width 10: 32-bit width Please do not set other values	W
b7~b5	Reserved	-	Write "0" on write	W
b4	WSYN	Write synchronization	0: Asynchronous write enable 1 : Synchronous write enable	W
b3~b1	Reserved	-	Write "0" on write	W
b0	RSYN	Read Synchronization	0: Asynchronous read enable 1 : Synchronous read enable	W

36.4.1.11 CHIP Status Register (SMC_CPSR0~3)

Reset value: 0xFF000A00

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
ADDMAT [7:0]								ADDMSK [7:0]							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-			BLSS	ADVS	BAAS	MW [1:0]		-			WSYN	-		RSYN	

Bit	Flag	Bit Name	Function	Read and write
b31~b24	ADDMAT [7:0]	CS space matching address	Matching address read	R
b23~b16	ADDMSK [7:0]	CS space mask address	Mask address readout	R
b15~b13	-	-	Read to 0	R
b12	BLSS	SMC_BLS status	SMC_BLS status readout	R
b11	ADVS	SMC_ADV status	SMC_ADV status readout	R
b10	BAAS	SMC_BAA status	SMC_BAA status readout	R
b9~b8	MW [1:0]	Memory width status	Memory width status readout	R
b7~b5	WBL [2:0]	-	Read to 0	R
b4	WSYN	Write action mode status	Write action mode status read	R
b3~b1	RBL [2:0]	-	Read to 0	R
b0	RSYN	Read action mode status	Read action mode status readout	R

36.4.1.12 Timing Configuration Register (SMC_TMCR)

Reset value: -

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved								t_tr [2:0]		-	t_pc [2:0]				
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	t_wp [2:0]			-	t_ceoe [2:0]			t_wc [3:0]			t_rc [3:0]				

Bit	Flag	Bit Name	Function	Read and write
b31~b23	Reserved	-	Write "0" on write	W
b22~b20	t_tr [2:0]	t_tr period setting	t_tr setting	W
b19	Reserved	-	Write "0" on write	W
b18~b16	t_pc [2:0]	T_pc period setting	t_pc setting	W
b15	Reserved	-	Write "0" on write	W
b14~b12	t_wp [2:0]	t_wp period setting	t_wp setting	W
b11	Reserved	-	Write "0" on write	W
b10~b8	t_ceoe [2:0]	t_ceoe period setting	t_ceoe setting	W
b7~b4	t_wc [3:0]	t_wc period setting	t_wc setting	W
b3~b0	t_rc [3:0]	t_rc period setting	t_rc setting	W

36.4.1.13 Timing Status Register (SMC_TMSR0~3)

Reset value: 0x001263CC

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved									t_tr [2:0]		-	t_pc [2:0]			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	t_wp [2:0]			-	t_ceoe [2:0]			t_wc [3:0]				t_rc [3:0]			

Bit	Flag	Bit Name	Function	Read and write
b31~b23	Reserved	-	Read as "0"	R
b22~b20	t_tr [2:0]	t_tr period state	t_tr value readout	R
b19	Reserved	-	Read as "0"	R
b18~b16	t_pc [2:0]	t_pc period state	t_pc value readout	R
b15	Reserved	-	Read as "0"	R
b14~b12	t_wp [2:0]	T_wp period state	t_wp value readout	R
b11	Reserved	-	Read as "0"	R
b10~b8	t_ceoe [2:0]	t_ceoe period status	t_ceoe value readout	R
b7~b4	t_wc [3:0]	t_wc period state	t_wc value readout	R
b3~b0	t_rc [3:0]	t_rc period state	t_rc value readout	R

36.5 Precautions

- The relevant registers of SMC can only be set in the Ready state;
- When using the SMC module to read external memory, the Memory Device transmits data to the controller after receiving the read command from the controller. The clock used by the controller to sample the data sent back can be selected through SMC_BACR.CKSEL. For specific settings, please refer to the register section. By selecting a different sampling clock, the time margin of the communication path can be improved.

SMC access in synchronous mode.

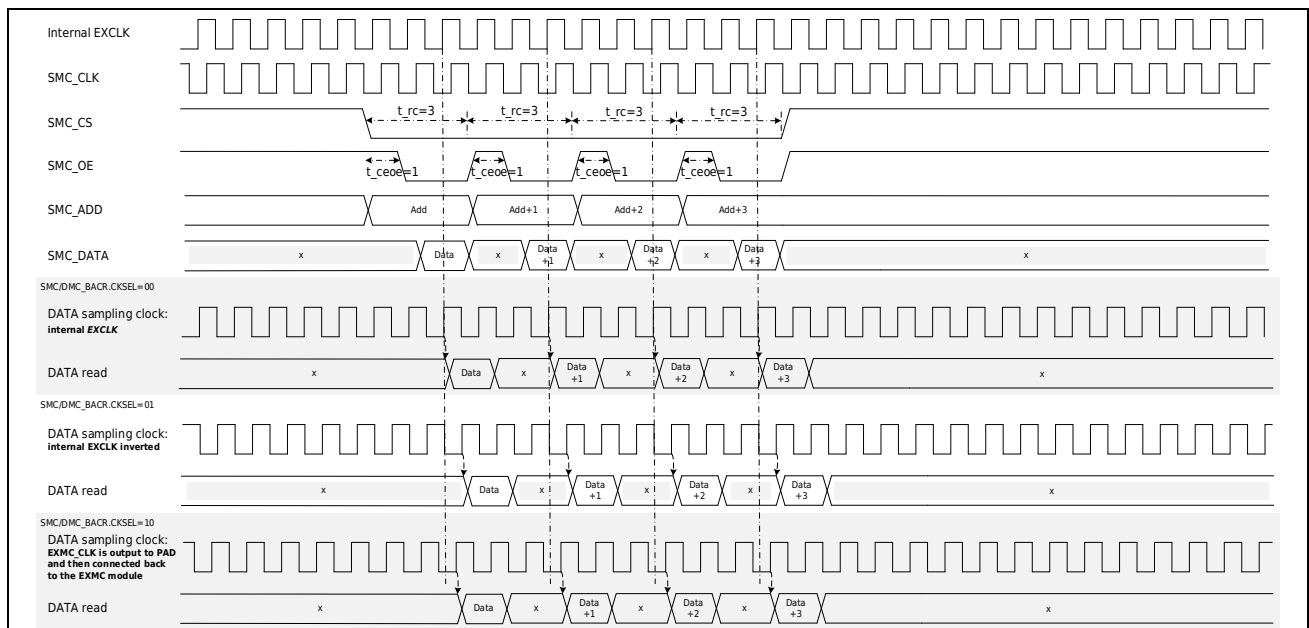


Figure 36-18 Instance Of SMC Access In Synchronous Mode

37 Cryptographic Coprocessing Module (CPM)

37.1 Introduction

The encryption co-processing module (CPM) includes three sub-modules: AES encryption and decryption algorithm processor, HASH security hash algorithm, and TRNG true random number generator.

The AES encryption and decryption algorithm processor complies with the new data encryption standard officially announced by the National Institute of Standards and Technology (NIST) on October 2, 2000. The block length is fixed at 128 bits, while the key length supports 128/192/256 bits.

The HASH secure hash algorithm is the SHA-2 version of SHA-256 (Secure Hash Algorithm), which complies with the national standard "FIPS PUB 180-3" issued by the US National Institute of Standards and Technology, and can process messages with a length of no more than 2^{64} bits. Produces a 256-bit message digest output. Supports HMAC (Keyed Hash Message Authentication Code) application to authenticate messages using the SHA-256 hash function.

TRNG true random number generator is a random number generator based on continuous analog noise, providing 64bit random numbers.

37.2 Encryption and Decryption Algorithm Processor (AES)

37.2.1 Algorithm Introduction

AES (The Advanced Encryption Standard) is a new data encryption standard officially announced by the National Institute of Standards and Technology (NIST) on October 2, 2000.

AES is fixed at 128 bits, while the key length supports 128, 192 and 256 bits. For encryption, the input is a plaintext block and a key, and the output is a ciphertext block; for decryption, the input is a ciphertext block and a key, and the output is a plaintext block. The process is as show in Figure 37-1:

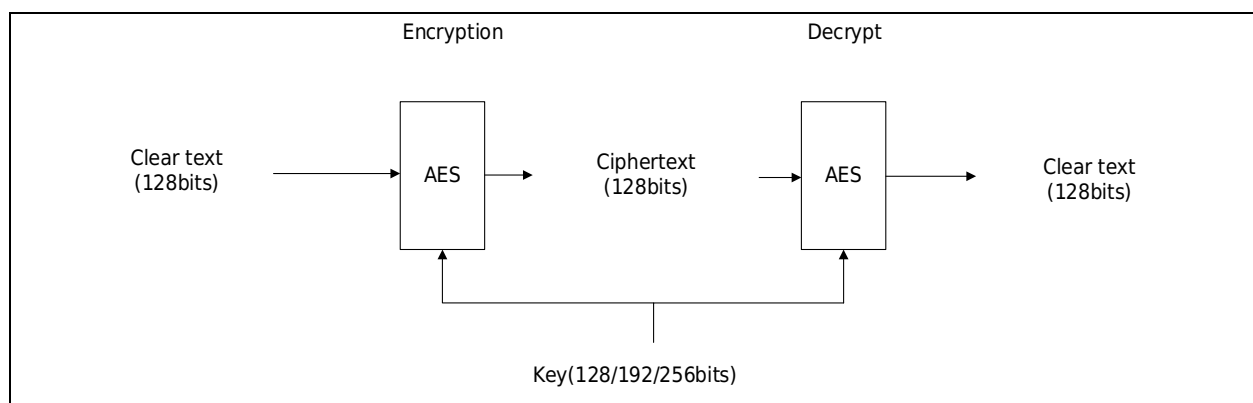


Figure 37-1 AES Encryption And Decryption Process Diagram

The basic unit processed by the AES algorithm is the byte. The 128-bit information is divided into 16 bytes, which are copied into a 4×4 matrix in order, which is called the state (state). All transformations of AES are transformations based on the state matrix, the intermediate results of the calculation are stored on this matrix.

AES is a key iterative block cipher that involves round-robin repetition of state. AES consists of four operations: SubBytes, ShiftRows, MixColumns, AddRoundKey. Among them, SubBytes is to find the modular inverse of each byte in $GF(2^8)$ and an affine transformation; ShiftRows is a byte transposition, which cyclically shifts the rows in the state according to different offsets; MixColumns linearly transforms each column of the state; AddRoundKey performs bit-by-bit XOR operation on each byte in the state and the round key. AES is as shown in Figure 37-2:

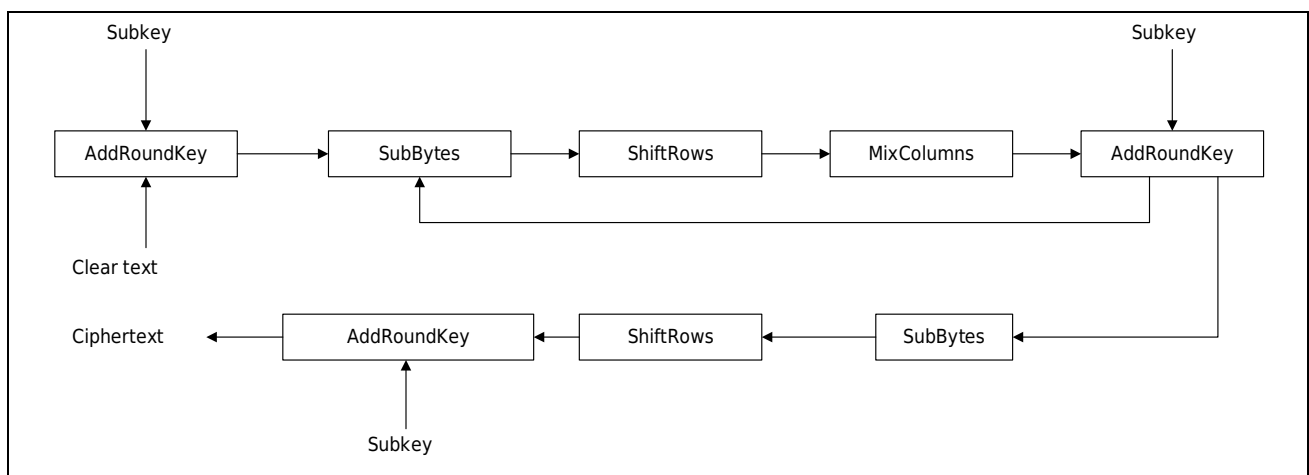


Figure 37-2 AES Encryption Flow Chart

The subkey used in the figure needs to be expanded from the initial key, and the key expansion process and encryption process are carried out synchronously.

Since the plaintext is fixed at 128 bits, the number of rounds the encryption process runs depends on the length of the key. For example, when the key is 128 bits, the number of running rounds is 10; when the key is 192 bits, the number of running rounds is 12; when the key is 256 bits, the number of running rounds is 14 rounds. Except for the lack of MixColumns transformation in the last round, the rest of the rounds perform a complete round transformation operation.

The decryption process is different from the encryption process. First, the expansion of all keys must be completed. The decryption process is used backwards from the last round of expansion, and then the four operations of the round transformation become the corresponding inverse operations: InvSubByte the s , InvShiftRows , InvMixColumns , AddRoundKey. The modular inverse operation in InvSubBytes is still maintained, but the affine transformation is changed to inverse transformation; InvShiftRows and InvMixColumns become corresponding inverse transformations; AddRoundKey remains unchanged.

The calling order of the round transformation of the direct decryption process for the four operations is: InvShiftRows, InvSubBytes, AddRoundKey, InvMixColumns, which is inconsistent with the calling order of the encryption process, but the key used is the same as the encryption process; the calling order of the round transformation of the equivalent decryption process for the four operations is: InvSubBytes, InvShiftRows, InvMixColumns, AddRoundKey, which is exactly the same as the calling order of the encryption process, except that the sub-key of each round needs to perform InvMixColumns operation.

For detailed algorithm expression, please refer to the standard " FIPS PUB 197 "

37.2.2 AES Module Function Description

- Execute the encryption process and decryption process of the AES algorithm standard, and its execution results fully comply with the description of the algorithm principle in " FIPS PUB 197";
- 128, 192 and 256 -bit keys are supported.

37.2.3 Encryption Operation Process

The encryption operation flow of AES is as follows:

- 1) 128 -bit data to be encrypted into the data register (AES_DR).
- 2) Write the encryption key into the key register (AES_KR).
- 3) Set AES_CR.KEYSIZE according to key length
- 4) Set AES_CR.MODE to 0 to enable encryption mode.
- 5) Write 1 to AES_CR.START in the control register to start the module to operate.
3), 4) and 5) can be performed simultaneously.
- 6) Wait for the value of AES_CR.START to return to 0, and the module operation ends.
- 7) Read the data register (AES_DR) to obtain 128-bit ciphertext.

37.2.4 Decryption Operation Process

The AES decryption operation flow is as follows:

- 1) 128 -bit data to be decrypted into the data register (AES_DR).
- 2) Write the decryption key into the key register (AES_KR).
- 3) Set AES_CR.KEYSIZE according to key length
- 4) Set AES_CR.MODE to 1 to enable decryption mode.
- 5) Write 1 to AES_CR.START in the control register to start the module to operate.
3), 4) and 5) can be performed simultaneously.
- 6) Wait for the value of AES_CR.START to return to 0, and the module operation ends.
- 7) Read the data register (AES_DR) to get 128-bit plaintext.

37.2.5 Data Instance

128-bit plaintext

0xFFEEDDCCBBAA99887766554433221100

128-bit key:

0x0F0E0D0C0B0A09080706050403020100

128-bit ciphertext:

0x5AC5B47080B7CDD830047B6AD8E0C469

Table 37-1 128 -Bit Operation Register Example

Before Encryption			
Register	Value (key)	Register	Value (clear text)
Key0	0x03020100	Data0	0x33221100
Key1	0x07060504	Data1	0x77665544
Key2	0x0B0A0908	Data2	0xBBAA9988
Key3	0x0F0E0D0C	Data3	0xFFEEDDCC
After encryption			
Register	Value (key)	Register	Value (ciphertext)
Key0	0x03020100	Data0	0xD8E0C469
Key1	0x07060504	Data1	0x30047B6A
Key2	0x0B0A0908	Data2	0x80B7CDD8
Key3	0x0F0E0D0C	Data3	0x5AC5B470

128-bit plaintext

0xFFEEDDCCBBAA99887766554433221100

192 -bit key:

0x17161514131211100F0E0D0C0B0A09080706050403020100

128-bit ciphertext:

0x5AC5B47080B7CDD830047B6AD8E0C469

Table 37-2 192 -Bit Operation Register Example

Before Encryption			
Register	Value (key)	Register	Value (clear text)
Key0	0x03020100	Data0	0x33221100
Key1	0x07060504	Data1	0x77665544
Key2	0x0B0A0908	Data2	0xBBAA9988
Key3	0x0F0E0D0C	Data3	0xFFEEDDCC
Key4	0x13121110	-	-
Key5	0x17161514	-	-
After encryption			
Register	Value (key)	Register	Value (ciphertext)
Key0	0x03020100	Data0	0xA47CA9DD
Key1	0x07060504	Data1	0xE0DF4C86
Key2	0x0B0A0908	Data2	0xA070AF6E
Key3	0x0F0E0D0C	Data3	0x91710DEC
Key4	0x13121110	-	-
Key5	0x17161514	-	-

128-bit plaintext

0xFFEEDDCCBBAA99887766554433221100

256 -bit key:

0x1F1E1D1C1B1A191817161514131211100F0E0D0C0B0A09080706050403020100

128-bit ciphertext:

0x5AC5B47080B7CDD830047B6AD8E0C469

Table 37-3 256 -Bit Operation Register Example

Before Encryption			
Register	Value (key)	Register	Value (clear text)
Key0	0x03020100	Data0	0x33221100
Key1	0x07060504	Data1	0x77665544
Key2	0x0B0A0908	Data2	0xBBAA9988
Key3	0x0F0E0D0C	Data3	0xFFEEDDCC
Key4	0x13121110	-	-
Key5	0x17161514	-	-
Key6	0x1B1A1918	-	-
Key7	0x1F1E1D1C	-	-
After encryption			
Register	Value (key)	Register	Value (ciphertext)
Key0	0x03020100	Data0	0xCAB7A28E
Key1	0x07060504	Data1	0xBF456751
Key2	0x0B0A0908	Data2	0x9049FCEA
Key3	0x0F0E0D0C	Data3	0x8960494B
Key4	0x13121110	-	-
Key5	0x17161514	-	-
Key6	0x1B1A1918	-	-
Key7	0x1F1E1D1C	-	-

37.2.6 Runtime Instructions

The time required for the AES module to start an operation (write 1 to AES_CR.START) and end the operation (restore AES_CR.START to 0) is shown in the following table:

Table 37-4 AES Encryption And Decryption Running Time

-	128 -bit key	192 -bit key	256 -bit key
Encryption	220 cycles	260 cycles	300 cycles
Decrypt	290 cycles	332 cycles	398 cycles

37.2.7 Operation Precautions

- During the AES encryption and decryption process, the data register will change, if the data of the next operation operation is the result of this operation, there is no need to rewrite the data.
- Support 128, 192 and 256- bit keys, 128- bit key write offset address 0x20~0x2C, 192 -bit key write offset address 0x20~0x34, 256 -bit key write offset address 0x20~0x3C.
- The method of judging the end of the operation of the module: read AES_CR.START continuously, if its value becomes 0, it means the end of the operation.

37.2.8 Register Description

Table 37-5 Register List

Register base address : 0x40008000

Register Name	Symbol	Offset	Bit Width	Reset Value
AES control register	AES_CR	0x0000	32	0x00000000
AES data register 0~3	AES_DR0~3	0x0010~0x001C	32	0x00000000
AES key register 0~7	AES_KR0~7	0x0020~0x003C	32	0x00000000

37.2.8.1 AES Control Register (AES_CR)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved											KEYSIZE		Reserved	MODE	START
Bit		Flag		Bit Name				Function					Read and write		
b31~b5		Reserved		-				Read as "0", write as "0"					R/W		
b4~b3		KEYSIZE		Key length selection				2'b00: key length is 128 bits 2'b01: key length is 192 bits 2'b10: key length is 256 bits 2'b11: key length is 128 bits					R/W		
b2		Reserved		-				Read as "0", write as "0"					-		
b1		MODE		Encryption and decryption mode selection				0: encryption operation 1: Decryption operation					R/W		
b0		START		Start up				0: The operation of this module is completed or has not been started 1: Start this module for calculation					R/W		

Description:

1. The operation method of AES_CR.START bit is: after the software writes 1 to this bit, the module will start running. After this operation, the hardware of this module will automatically clear this bit to 0. If the software checks that this bit is 0, it means that this operation is completed.
2. The write operation to this register can only be performed when the module is not in the operation state (that is, when AES_CR.Start = 0), otherwise the hardware will automatically ignore the write operation. Read operations are not subject to this restriction.

37.2.8.2 AES Data Register (AES_DRx) (x=0-3)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Data [31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Data [15:0]															
Bit	Flag		Bit Name		Function		Read and write								
b31~b0	Data		Encrypting Data		Store 128-bit plaintext / ciphertext of AES algorithm		R/W								

Description:

1. The data register consists of four 32-bit registers consisting of 128-bit data, which are used to store the plaintext that needs to be encrypted or the ciphertext that needs to be decrypted before the operation of the module, and store the encrypted ciphertext or decrypted plaintext after the operation is completed.

Encryption operation		Decryption operation	
Before operation	After operation	Before operation	After operation
128-bit plaintext	128-bit ciphertext	128-bit ciphertext	128-bit plaintext

Four 32-bit registers are connected together to form a 128-bit data, and the four registers need to be operated separately during read and write operations. The operation sequence corresponding to the data register is as follows:

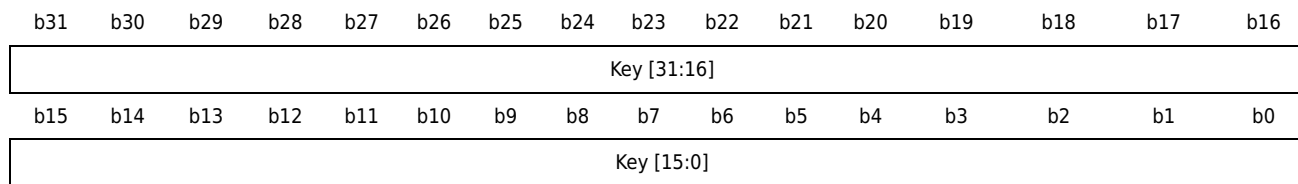
Data example: 0xFFEEDDCCBAA99887766554433221100

Offset Address	Register Name	Fill in Data
0x10	AES_DR0	0x33221100
0x14	AES_DR1	0x77665544
0x18	AES_DR2	0xBBAA9988
0x1C	AES_DR3	0xFFEEDDCC

2. Writing to this register can only be done when the module is not in operation state (that is, when AES_CR.Start = 0), otherwise the hardware will automatically ignore the writing operation to this register.
3. The reading of this register can only be carried out when the module is not in the operation state (that is, when AES_CR.Start = 0), otherwise the reading of this register will get all 0s.

37.2.8.3 AES Key Register (AES_KRx) (x=0~7)

Reset value: 0x00000000



Bit	Flag	Bit Name	Function	Read and write
b31~b0	Key	Key	Stores 128/192/ 256-secret keys for the AES algorithm	R/W

Description:

1. The key register consists of eight 32- bit registers, which store the input initial key. The write operation needs to operate on 8 32 -bit registers respectively.

The corresponding sequence of operations is as follows:

128 -bit secret key: 0x0F0E0D0C_0B0A0908_07060504_03020100

Offset Address	Register Name	Fill in Data
0x20	AES_KR0	0x03020100
0x24	AES_KR1	0x07060504
0x28	AES_KR2	0x0B0A0908
0x2C	AES_KR3	0x0F0E0D0C

192 -bit secret key: 0x17161514_13121110_0F0E0D0C_0B0A0908_07060504_03020100

Offset Address	Register Name	Fill in Data
0x20	AES_KR0	0x03020100
0x24	AES_KR1	0x07060504
0x28	AES_KR2	0x0B0A0908
0x2C	AES_KR3	0x0F0E0D0C
0x30	AES_KR4	0x13121110
0x34	AES_KR5	0x17161514

256 -bit secret key:

0x1F1E1D1C_1B1A1918_17161514_13121110_0F0E0D0C_0B0A0908_07060504_03020100

Offset Address	Register Name	Fill in Data
0x20	AES_KR0	0x03020100
0x24	AES_KR1	0x07060504
0x28	AES_KR2	0x0B0A0908
0x2C	AES_KR3	0x0F0E0D0C
0x30	AES_KR4	0x13121110
0x34	AES_KR5	0x17161514
0x38	AES_KR6	0x1B1A1918
0x3C	AES_KR7	0x1F1E1D1C

2. Writing to this register can only be done when the module is not in operation state (that is, when AES_CR.Start = 0), otherwise the hardware will automatically ignore the writing operation to this register.
3. The reading of this register can only be carried out when the module is not in the operation state (that is, when AES_CR.Start = 0), otherwise the reading of this register will get all 0s.

37.3 Secure Hash Algorithm (HASH)

37.3.1 Algorithm Introduction

The steps of the secure hash algorithm are as follows:

First pad the message so that its length is exactly a number that is only 64 bits less than a multiple of 512. The padding method is to attach a 1 to the back of the message, followed by the required multiple 0s, and then attach a 64-bit message length (before padding) to make the message length exactly an integer multiple of 512 bits.

Secondly, initialize the eight 32-bit variables A, B, C, D, E, F, G, and H using hexadecimal. Then start the main loop of the algorithm, process 512-bit messages at a time, the number of loops is the number of 512-bit packets in the message.

The main loop performs a total of 64 operations, which are called compression functions. Each operation includes shift, cyclic shift, logic operation, modulo 232 addition, etc. The operation process is shown in Figure 37-3 below. The final output is formed by cascading A, B, C, D, E, F, G, H. Among them, W_t is the temporary value used in the t th step obtained from the 512-bit message, K_t is the constant value used in the t th step, and t ($0 \leq t \leq 63$) is a step in the 64-step period.

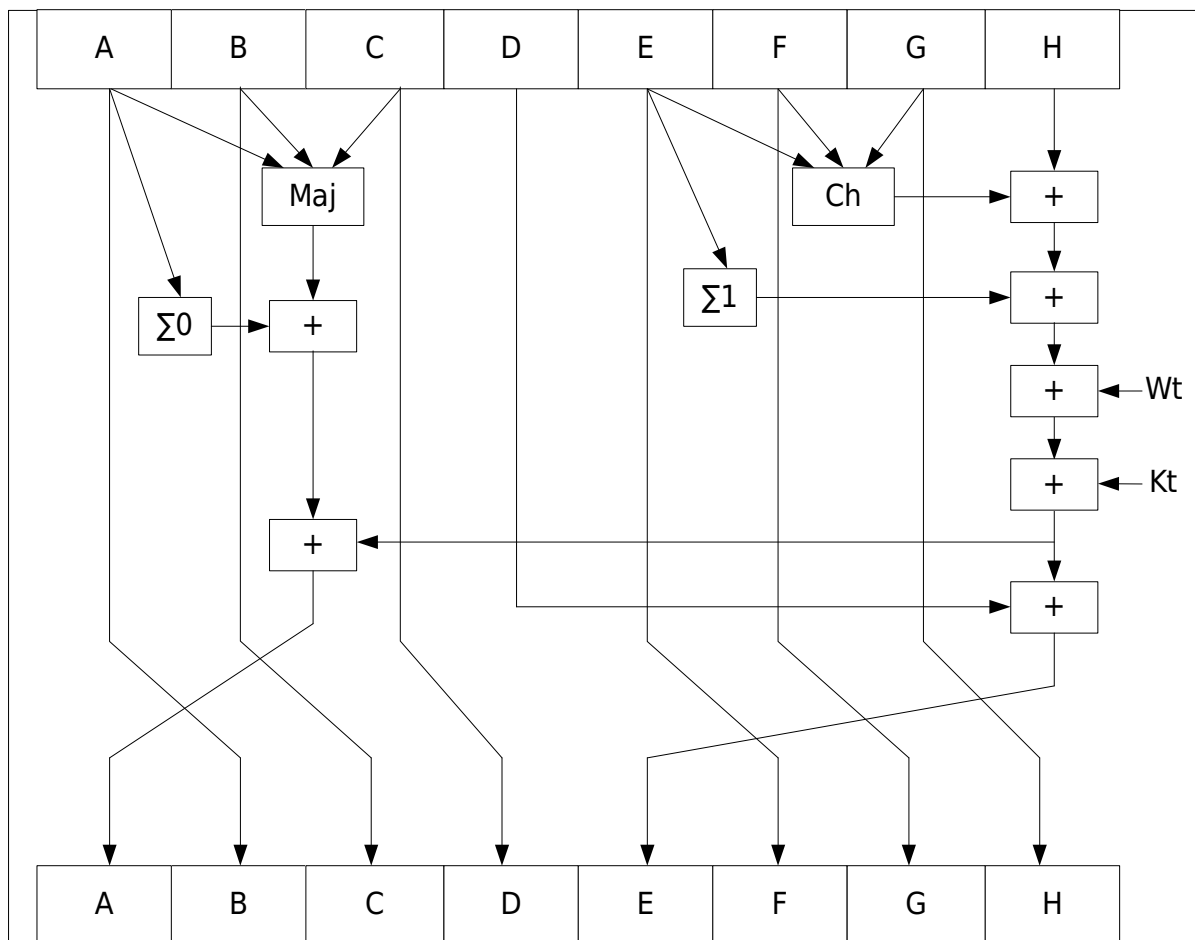


Figure 37-3 HASH Algorithm Flow Chart

HASH module integrates the hardware HMAC function; the HMAC function can also be implemented by directly calling the HASH operation according to the HMAC algorithm through software.

37.3.2 Operating Procedures

The operation process of the HSAH module is as follows:

Software operation process:

- 1) The software fills the original data according to the algorithm rules and groups the filled messages into 512-bit blocks.
- 2) Write the data into the data register HASH_DR.
- 3) If this operation is the first group of data in the message group, set HASH_CR.FST_GRP=1.
- 4) Set HASH_CR.START=1 to start this module for calculation.
Note: 3) and 4) above can be performed simultaneously
- 5) To judge whether the operation of this module is completed, use the following methods:
Continuously read HASH_CR.START until the bit is 0 , which indicates the operation is completed; or continuously read HASH_CR.BUSY until the bit is 0 , which indicates the operation is completed; or set the HASH interrupt and determine the end of the operation through the interrupt method.
- 6) If this operation is not the last set of data in the message group, return to 2).
- 7) If this operation is the last set of data in the message group, read the summary register (HASH_HR) to obtain the result of this operation. If you need to perform the calculation again, return to step 1).

Using DMA operation process:

1. The messages to be processed are grouped into 512- bit blocks, and the last group of messages needs to be pre-filled into 512- bit blocks according to the rules.
2. HASH operation start mode of the internal trigger event register to DMA transfer completion and data block transfer completion; set the DMA transfer start mode of the internal trigger event register to HASH operation completion (or other internal trigger events).
3. Set HASH_CR.MODE=2'b00, HASH operation mode; set HASH_CR.FST_GRP=1, and set HASH_CR.START=1 to start HASH operation. If you select other internal trigger events to start DMA , you need to wait for the trigger event to occur before starting the HASH operation.
4. Wait for HASH_CR.CYC_END=1 , the HASH operation is completed and the bit is cleared.
5. reading the summary register HASH_HR.

Note: When there is only one set of operation data, please set HASH_CR.FST_GRP=1 and HASH_CR.KMSG_END=1 at the same time.

37.3.3 Message Padding

The padding packet processing steps of SHA-256 are as follows:

1. Raw message grouping

Divide the original message into L groups with a size of 512 bits. Let the total number of bits of the original message be I. If $I \% 512 < 448$, then the grouping number L is $I / 512$; if $I \% 512 \geq 448$, then the grouping number L is $I / 512 + 1$.

2. Add length

■ Add padding:

Add padding bits at the end of the 1/512th group of the message packet: one 1 and several 0s, the number of 0s can be zero. If $I \% 512 < 448$, padding makes the length of the data bits meet the length of $448 \bmod 512$ (the last 64 bits are reserved for the original message length); The 512-bit data block of the group is filled, and the first 448 bits of the L ($L = I / 512 + 1$) group are filled with 0.

■ Add raw message length:

A 64bit block, representing the length of the original message, is a 64bit unsigned integer. Add the original message length to the last 64bit of the Lth packet.

An instance to illustrate the process of filling a group is as follows:

1) Example1:

The original message is a string of "abcde". Its ASCII code is represented by a binary bit string : "01100001 01100010 01100011 01100100 01100101". The steps to add the length are as follows:

A. Add "1". The filled message is "01100001 01100010 01100011 01100100 01100101 1".

B. Add "0". Because the length of the original message is 40bit, the number of 0s to be added is $512 - 64 - 40 = 407$.

The padded message becomes (hex):

```
61626364 65800000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000
```

C. Add the original message length. The original message length 40 is expressed in two 32bit words (hexadecimal): 00000000 00000028.

The padded message becomes (hex):

```
61626364 65800000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000028
```

2) Example2:

The original message is the string "abcdbcdecdefdefgefghfghighijhijkijkljklmklmnlmnomnopnopq". Each character can be converted to 8bit through its ASCII code, so the length of the message is $l = 56 * 8 = 448$.

A. Add "1" and "0". The padded message (hexadecimal) is the first message block:

```
61626364 62636465 63646566 64656667
65666768 66676869 6768696A 68696A6B
696A6B6C 6A6B6C6D 6B6C6D6E 6C6D6E6F
6D6E6F70 6E6F7071 80000000 00000000.
```

B. Add the original message length. The original message length of 448 is expressed in two 32bit words (hexadecimal): 00000000 000001C0.

The padded message (hexadecimal) is the second message block:

```
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 000001C0
```

37.3.4 HMAC Operation

HMAC (Keyed-Hashing for Message Authentication) binds the message being processed to a key selected by the user in an irreversible manner for message authentication. Basically, HMAC consists of two nested underlying hash operations with the addition of a shared secret:

$$\text{HMAC}(\text{message}) = \text{Hash} [((\text{key XOR opad}) \mid \text{Hash}((\text{key XOR ipad}) \mid \text{message}))]$$

in:

ipad: 64 bytes (SHA-256) 0x36.

opad: 64 bytes (SHA-256) 0x5C.

| : represents the concatenation operator.

37.3.4.1 HMAC Calculation Process

- 1) Set HASH_CR.MODE=2'b01 to select HMAC working mode;
- 2) If the key length used is longer than 64 bytes, you need to set HASH_CR.LKEY=1 (in this case, HMAC will use the key specified by the specification instead of the given key);
- 3) Write the key to the HASH_DR register until all key group operations are completed;
- 4) Write the message to the HASH_DR register until all message grouping operations are completed;
- 5) Read the final hash operation result from the HASH_HR register.

In the process of HMAC operation, you can choose the software operation process or DMA operation process to load data. The specific instances are as follows:

37.3.4.2 HMAC Software Operation Process

- 1) Set mode HASH_CR.MODE=2'b01 , HMAC working mode; set HASH_CR.LKEY=1, select long key mode;
- 2) Write key group data into HASH_DR;
- 3) If it is the first set of data, set HASH_CR.FST_GRP=1. Set HASH_CR.START=1 to start key operation;
- 4) Wait for HASH_CR.BUSY=0 operation to complete;
- 5) If it is not the last set of data, repeat 2)~4);
- 6) If it is the last set of data, set HASH_CR.KMSG_END=1, and set HASH_CR.START=1 to start the last set of key operations. After waiting for HASH_CR.CYC_END=1, the key operation is completed and the bit is cleared.
- 7) Write the message packet data into HASH_DR;
- 8) If it is the first group of message data, set HASH_CR.FST_GRP=1. Set HASH_CR.START=1 to start message operation;
- 9) Wait for HASH_CR.BUSY=0 operation to complete;
- 10) If it is not the last set of message data, repeat operations 7)~9);
- 11) If it is the last set of messages, set HASH_CR.KMSG_END=1, and set HASH_CR.START=1 to start the operation of the last set of messages. After waiting for HASH_CR.CYC_END=1, all HMAC operations are completed and the bit is cleared;
- 12) Read the final message authentication result in HASH_HR.

37.3.4.3 HMAC Uses DMA Operation Process

- 1) Set DMA to data block transmission mode. The size of the data block is 16 words. Both the key and the message need to be grouped according to the size of 16 words. If it is less than 16 words, please fill it according to the HASH message filling rules. Configure the DMA data block source address as the key address;
- 2) HASH operation start mode of the internal trigger event register to DMA transfer completion and data block transfer completion; set the DMA transfer start mode of the internal trigger event register to HASH operation completion (or other internal trigger events).
- 3) Set HASH_CR.MODE=2'b01 , HMAC operation mode; configure HASH_CR.LKEY bit according to the key length; set HASH_CR.FST_GRP=1 , HASH_CR.START=1 to start the key operation; if other internal trigger events are selected, you need to wait for the trigger event to start the operation;
- 4) After waiting for HASH_CR.CYC_END=1, the key operation is completed and then clear this bit;
- 5) Stop DMA and point the DMA data block source address to the message address;
- 6) Set HASH_CR.FST_GRP=1, HASH_CR.START=1 to start message operation; or wait for other

- 7) After waiting for the HASH_CR.CYC_END=1 message operation to complete, clear this bit;
- 8) Read the final message authentication result in the HASH_HR register.

HMAC operations on the key and the message are as follows:

37.3.4.5 Precautions

1. When the padded key and message have only one set of data, i.e. 512 bits, please set HASH_CR.FST_GRT=1 and HASH_CR.KMSG_END=1 before starting the operation;
2. Regarding key padding, when the key is less than 64 bytes, please fill the part less than 64 bytes with 0x00; when the key is larger than 64 bytes, please fill the part less than 64 bytes according to the HASH padding rules.

37.3.5 Interrupt Description

HASH interrupt flag HASH_INT. Regardless of whether it is in HASH operation mode or HMAC operation mode, when the interrupt permission condition is set to HASH_CR.HEIE=1 , an interrupt request will be generated after each set of data operation is completed. HASH_CR.HCIE=1 , an interrupt request will be generated after all key or message operations are completed. When HEIE and HCIE are permitted at the same time, an interrupt request will be generated if any of the conditions is met. In addition, after HEIE is enabled, after the module is reset or the operation is completed, if HASH_CR.FST_GRP is directly set and HASH_CR.START=1 , an interrupt will be generated for data loading.

37.3.6 Hardware Trigger Event Selection

By configuring the internal hardware trigger event selection registers HASH_TRGSELA and HASH_TRGSELB, the HASH operation is started after the data transfer is completed. It is usually used in conjunction with DMA. For example, when DMA_2 is selected to transfer data and start HASH operation, first, configure DMA2_TRGSELx (x=0~7) to select HASH_INT to start DMA transfer; secondly, configure HASH_TRGSELA to select DMA_BTCx (x=0~7) , indicating that a data block (512bit) transfer is completed and the HASH operation is started, and configure HASH_TRGSELB to select DMA_TCx (x=0~7) , indicating that all data data block transmissions are completed. HASH is started , DMA and HASH trigger each other until all data blocks are transferred and calculated.

37.3.7 Register Description

Table 37-6 Register List

BASE ADDR: 0x40008400

Register Name	Symbol	Offset	Bit Width	Reset Value
HASH control register	HASH_CR	0x0000	32	0x00000000
HASH digest register 7	HASH_HR7	0x0010	32	0x00000000
HASH digest register 6	HASH_HR6	0x0014	32	0x00000000
HASH digest register 5	HASH_HR5	0x0018	32	0x00000000
HASH digest register 4	HASH_HR4	0x001C	32	0x00000000
HASH digest register 3	HASH_HR3	0x0020	32	0x00000000
HASH digest register 2	HASH_HR2	0x0024	32	0x00000000
HASH digest register 1	HASH_HR1	0x0028	32	0x00000000
HASH digest register 0	HASH_HR0	0x002C	32	0x00000000
HASH data register 15	HASH_DR15	0x0040	32	0x00000000
HASH data register 14	HASH_DR14	0x0044	32	0x00000000
HASH data register 13	HASH_DR13	0x0048	32	0x00000000
HASH data register 12	HASH_DR12	0x004C	32	0x00000000
HASH data register 11	HASH_DR11	0x0050	32	0x00000000
HASH data register 10	HASH_DR10	0x0054	32	0x00000000
HASH data register 9	HASH_DR9	0x0058	32	0x00000000
HASH data register 8	HASH_DR8	0x005C	32	0x00000000
HASH data register 7	HASH_DR7	0x0060	32	0x00000000
HASH data register 6	HASH_DR6	0x0064	32	0x00000000
HASH data register 5	HASH_DR5	0x0068	32	0x00000000
HASH data register 4	HASH_DR4	0x006C	32	0x00000000
HASH data register 3	HASH_DR3	0x0070	32	0x00000000
HASH data register 2	HASH_DR2	0x0074	32	0x00000000
HASH data register 1	HASH_DR1	0x0078	32	0x00000000
HASH data register 0	HASH_DR0	0x007c	32	0x00000000

37.3.7.1 HASH Control Register (HASH_CR)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
HEIE	HCIE	-	-	-	HMAC_END	CYC_END	BUSY	-	LKEY	MODE [1:0]	-	KMSG_END	FST_GRP	START	
Bit	Flag	Bit Name		Function		Read and write									
b31~b16	Reserved	-		When reading, it is "0", when writing, please write "0"		R/W									
b15	HEIE	Each group of data operation is completed and interrupt permission is granted		0: Disable interrupt after each operation is completed 1: Each group of operations is completed and the interrupt is allowed		R/W									
b14	HCIE	All key or message operations completed interrupt permission		0: All key or message operation completion interrupts are disabled 1: All secret keys or message operations completed interrupt permission		R/W									
b14~b11	Reserved	-		When reading, it is "0", when writing, please write "0"		R/W									
b10	HMAC_END	HMAC operation completion flag		0: HMAC calculation is not completed 1: HMAC operation completed When the flag is "1", write "0" to clear it. Note: The software should clear "0" in time after confirming the flag bit, otherwise the flag bit will remain "1".		R/W									
b9	CYC_END	Key or message operation completion flag		0: Key or message operation is not completed 1: Secret key or message operation completed When the flag is "1", write "0" to clear it. Note: The software should clear "0" in time after confirming the flag bit, otherwise the flag bit will remain "1".		R/W									
b8	BUSY	Conversion Flag		0: HASH operation is in idle state 1: HASH operation in progress		R									
b7	Reserved	-		When reading, it is "0", when writing, please write "0"		R/W									
b6	LKEY	Long key selection		0: short key (≤ 64 bytes) 1: Long key (> 64 bytes) In HMAC mode, this bit selects between short keys (≤ 64 bytes) or long keys (> 64 bytes). Valid when MODE=2'b01, HMAC mode.		R/W									
b5~b4	MODE	Operating mode		00: SHA-256 working mode 01: HMAC working mode 1X: Setting prohibited		R/W									
b3	Reserved	-		When reading, it is "0", when writing, please write "0"		R/W									
b2	KMSG_END	The last group of keys or messages		0: The key or message is not the last group 1: The last group of secret keys or messages When the last set of keys or message conversion is completed, this bit is automatically cleared to "0"											
b1	FST_GRP	The first group of keys or messages		0: The key or message is not the first group 1: Secret key or message first group When the first set of keys or message conversion is completed, this bit is automatically cleared to "0"		R/W									
b0	START	Start		0: The operation of this module is completed or has not been started 1: Start this module for calculation		R/W									

Note:

- The operation method of the START bit is: after the software writes 1 to this bit, the module will start to run; after the end of this run, the hardware will automatically clear this bit to 0; if the software queries this bit to 0, it means that this run is completed.
- The write operation to this register can only be performed when the module is not in the operation state (that is, when the START bit is 0), otherwise the hardware will automatically ignore the write operation. Read operations are not subject to this restriction.

37.3.7.2 HASH Digest Register (HASH_HR)

Number of bits : 256 bits

Offset address: 10'h010 – hash [255:224]

10'h014 – hash [223:192]

10'h018 – hash [191:160]

10'h01C – hash [159:128]

10'h020 – hash [127:96]

10'h024 – hash [95:64]

10'h028 – hash [63:32]

10'h02C – hash [31:0]

Reset value : 0x00000000 (per 32-bit register)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
HASH [31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
HASH [15:0]															
Bit	Flag	Bit Name		Function		Read and write									
b31~b0	HASH [31:0]	summary value		After the operation of the module is completed, the message summary can be obtained by reading this register		R/W									

Note:

- This register is spliced by 8 32-bit registers. During the access, 8 32-bit registers are operated in turn, and the 32-bit register corresponding to the low address stores the high word of the message digest.
- Hardware will automatically ignore the write operation to this register.
- The reading of this register can only be carried out when the module is not in the operation state (HASH_CR.START=0), otherwise the reading of this register will get all 0s.

37.3.7.3 HASH Data Register (HASH_DR)

Number of bits : 512 bits

Offset address: 10'h040 – data [511: 480]

10'h044 – data [479: 448]

10'h048 – data [447: 416]

10'h04C – data [415: 384]

.....

10'h070 – data [127: 96]

10'h074 – data [95: 64]

10'h078 – data [63: 32]

10'h07C – data [31: 0]

Reset value : 0x00000000 (per 32-bit register)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
DATA [31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DATA [15:0]															
Bit	Flag		Bit Name		Function		Read and write								
b31~b0	DATA [31:0]		Data register		Used to write messages before block operations										R/W

Note:

- This register is spliced by 16 32-bit registers. When accessing, 16 32-bit registers are operated in turn, and the 32-bit register corresponding to the low address stores the high word of the data.
- Writing to this register can only be performed when the module is not in the operation state (HASH_CR.START), otherwise the hardware will automatically ignore the writing operation to this register.
- A read of this register will always return all zeros.

37.4 True Random Number Generator (TRNG)

37.4.1 Module Block Diagram

The TRNG module provides a true random number generator to generate a 64-bit random number.

The system block diagram of TRNG is shown in Figure 37-4 below. The random number generator is an analog random number oscillator circuit, which is used to obtain random noise; the algorithm module captures random noise and saves the result to the data module, and outputs it through the bus; the control module controls the mode and start of TRNG.

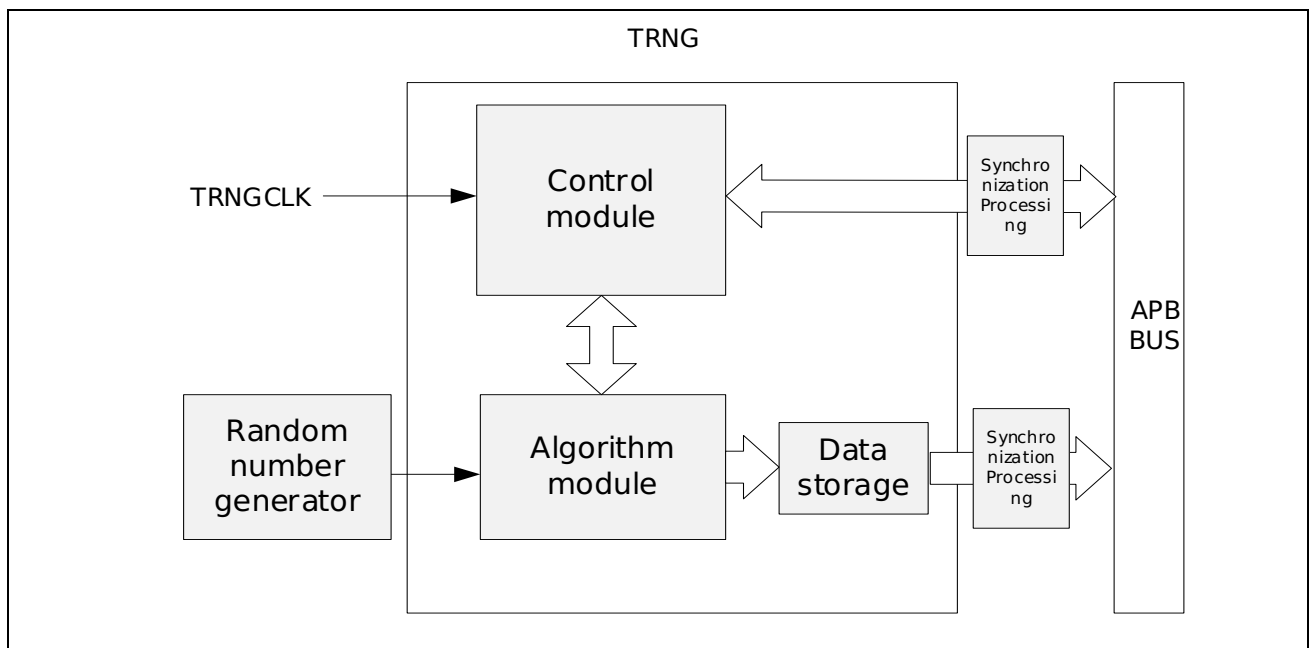


Figure 37-4 TRNG System Block Diagram

37.4.2 Operating Procedures

The true random number generation process is as follows:

1. Turn on the random number generator circuit (set the EN bit of TRNG_CR to 1).
2. Configure the random number generation mode (set TRNG_MR).
3. Start random number generation (set the RUN bit of TRNG_CR to 1).
4. Read random number (read TRNG_DR).
5. Turn off the random number generator circuit (set the EN bit of TRNG_CR to 0).

37.4.3 Interrupt and Event Output

When the random number generation is finished, the register bit TRNG_CR.RUN is cleared by hardware and generates a random number generation interrupt request (TRNG_END). When the random number generation ends, an event output is also generated, which can trigger the linkage of other modules.

37.4.4 Register Description

Table 37-7 Register List

BASE ADDR: 0x40042000

Register Name	Symbol	Offset	Bit Width	Reset Value
TRNG control register	TRNG_CR	0x0000	32	0x00000000
TRNG Mode Register	TRNG_MR	0x0004	32	0x00000012
TRNG data register 0	TRNG_DR0	0x000C	32	0x08000000
TRNG data register 1	TRNG_DR1	0x0010	32	0x08000200

37.4.4.1 TRNG Control Register (TRNG_CR)

Reset value: 0x00000000

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

Bit	Flag	Bit Name	Function	Read and write
b31~b2	Reserved	-	Read as "0", write as "0"	R/W
b1	RUN	Random number operation starts	0: Stop random number operation 1: Random number operation starts The software writes "1" to generate a new 64-bit random number; after the operation is completed, the hardware clears it.	R/W
b0	EN	Analog oscillator enable	0: Turn off the analog random number generator circuit 1: Turn on the analog random number generator circuit	R/W

37.4.4.2 TRNG Mode Register (TRNG_MR)

Reset value: 0x00000012

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	CNT [2:0]			-	LOAD

Bit	Flag	Bit Name	Function	Read and write
b31~b5	Reserved	-	Read as "0", write as "0"	R/W
b4~2	CNT [2:0]	Shift count control bit	When capturing random noise, the number of shift control bits 011: Shift 32 times 100: Shift 64 times 101: Shift 128 times 110: Shift 256 times 000~010, 111: Function reserved bits	R/W
b1	Reserved	-	Read as "1", write as "1"	R/W
b0	LOAD	Load control bit	Whether the data register is loaded with a new initial value from the random number generator before the random number is generated 0: Do not load new initial values 1: Load new initial values	R/W

37.4.4.3 TRNG Data Register (TRNG_DR)

Reset value: DR0 0x08000000

DR1: 0x08000200

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

Bit	Flag	Bit Name	Function	Read and write
b31-b0	DATA [31:0]	Random number	64-bit random number	R

38 CRC Operation (CRC)

38.1 Introduction

In many applications, CRC algorithm is needed to verify the integrity and correctness of data. Especially in data transmission, CRC is widely used. This module can use CRC 16 and CRC 32 algorithms to calculate and check the data.

38.2 Functional Block Diagram

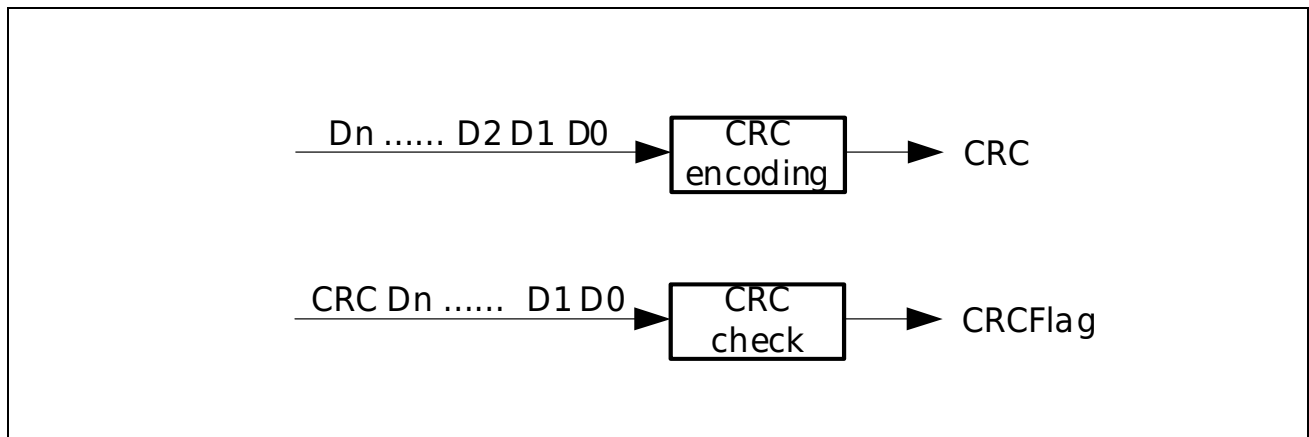


Figure 38-1 CRC Application Diagram

38.3 Functional Description

The CRC algorithm of this module complies with the definition of ISO/IEC 13239 and adopts 32-bit and 16-bit CRC respectively. The generating polynomial of CRC32 is $X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$, and the 32-bit initial value is "0xFFFFFFFF". The generating polynomial of CRC16 is $X^{16}+X^{12}+X^5+1$, and the 16-bit initial value is 0xFFFF.

The functions of this module include:

- CRC coding and CRC check;
- 3 Bitwidth access modes 8 bits, 16 bits, 32 bits:
 - Instances of 8-bit width input data are 0x00, 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77;
 - Instances of 16 bit width input data are 0x1100, 0x3322, 0x5544, 0x7766;
 - Instances of 32-bit width input data are 0x33221100, 0x77665544;

38.3.1 CRC16 Encoding Mode

CRC coding is to encode the original data to calculate its CRC value. The procedure is as follows:

1. Write 1'b0 to CRC _ CR.CR, select CRC16.
2. Write 0xFFFF to 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.

38.3.2 CRC16 Check Mode

Check mode can check if the encoded data has been tampered with. The operation flow is as follows:

1. Write 1'b0 to CRC_CR.CR, select CRC16.
2. Write 0xFFFF to CRC_RESLT [15: 0] and initialize CRC calculation.
3. Write the encoded data to the CRC_DATregister in order of 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. The checksum is written into the CRC_DAT register with a 16-bit width.
5. Read CRC_CR.FLGregister, 1 indicates check success, and 0 indicates check failure.

38.3.3 CRC32 Encoding Mode

CRC coding is to encode the original data to calculate its CRC value. The procedure is as follows:

1. Write 1'b1 to CRC_CR.CR, select CRC32.
2. Write 0xFFFF_FFFF to CRC_RESLT [31: 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 [31:0] to get 32-bit CRC coding value.

38.3.4 CRC32 Check Mode

Check mode can check if the encoded data has been tampered with. The operation flow is as follows:

1. Write 1'b1 to CRC_CR.CR, select CRC32.
2. Write 0xFFFF_FFFF to CRC_RESLT [31: 0] and initialize CRC calculation.
3. Write the encoded data to the CRC_DATregister in order of 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. 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.

38.4 Register Description

Table 38-1 Shown is the register list of the CRC module.

Table 38-1 CRC Register List

CRC_BASE_ADDR: 0x40008C00

Register Name	Symbol	Offset Address	Bit Width	Reset Value
CRC control register	CRC_CR	0x00	32	0x0000_0001
CRC result register	CRC_RESLT	0x04	32	0x0000_0000
CRC data register	CRC_DAT	0x80~0xFF	32	0x0000_0000

38.4.1 Control Register (CRC _ CR)

Reset value: 0x00000001

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

Bit	Flag	Bit Name	Function	Read and write
b30~b2	Reserved	-	Read as "0", write as "0"	R/W
b1	FLAG	Check result bit	0: Current check error 1: Current check is correct	R
b0	CR	Operational control bit	0: CRC16 1: CRC32	R/W

38.4.2 Results Register (CRC _ RESLT)

Reset value: 0x00000000

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	Flag	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.	R/W

38.4.3 Data Register (CRC _ DAT)

Reset value: 0x00000000

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	Flag	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.	R/W

39 Filter Math Accelerator (FMAC)

39.1 Introduction

The Filter Math Accelerator (FMAC) is a hardware acceleration module for FIR filter calculations. This module can perform FIR digital filtering with a maximum order of 16 and configurable order. Built-in 16x16 bit multiplier and 32+5bit adder, users can customize the output data precision. This series of products is equipped with 4 FMAC modules.

39.2 Basic Block Diagram

The basic features of this module are as follows:

- Maximum 16-order FIR filter, configurable order
- Configurable filter coefficients
- 16 -bit signed input and output data, 16 -bit signed filter coefficients
- 16x16 bit multiplier
- 37-bit adder
- Provide interrupt signal and status query
- AHB slave interface configuration parameters and input and output data

The basic block diagram is as follows:

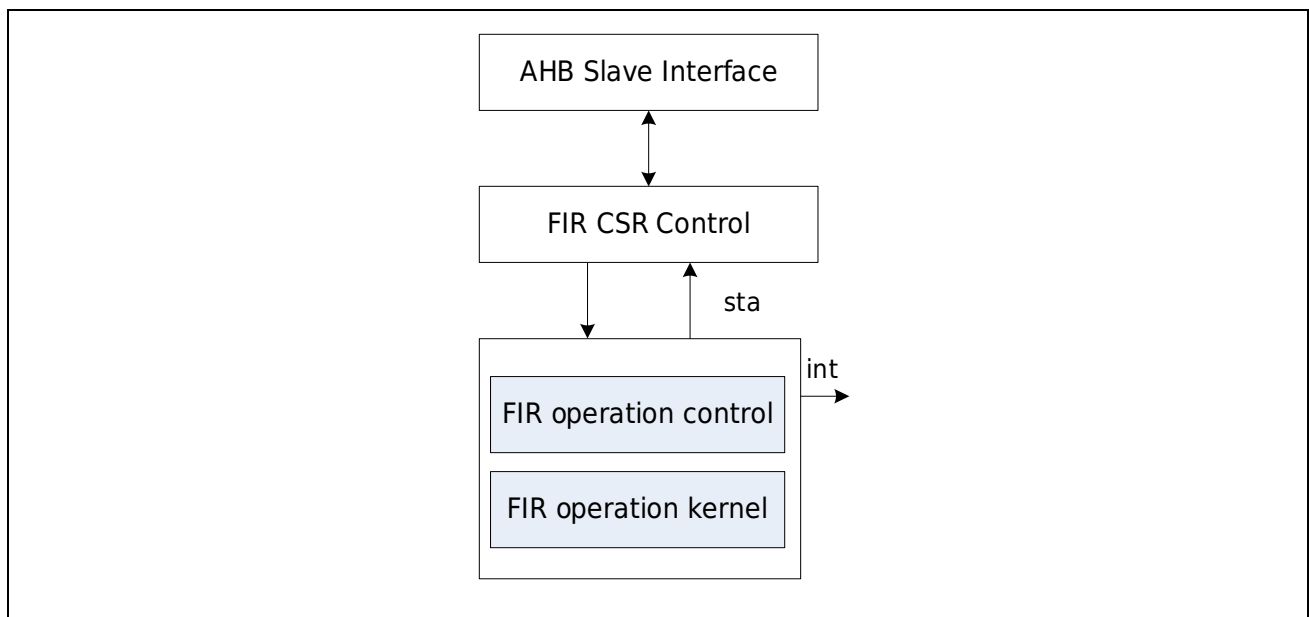


Figure 39-1 FIR Basic Block Diagram

When the 16th-order FIR filter is selected, the FIR operation control and operation core requires 17 clock periods to process one data. At an operating clock frequency of 120MHz, it can process a maximum input data stream of 7MHz. Different orders require different processing times. The smaller the order, the faster the processing speed.

39.3 Operating procedures

This module needs to be used according to the following steps:

- 1) Set the enable bit (FMAC_ENR.FIREN) to 1;
- 2) Configure the Filtering order (FMAC_CTR.STAGE_NUM [4:0]) and the number of shifts to output valid results (FMAC_CTR.SHIFT [4:0]);
- 3) If you need to use interrupts, configure the interrupt enable bit (FMAC_IER.INTEN) to 1;
- 4) Write filter coefficients (if there are N orders, just configure FMAC_COR0~FMAC_CORN);
- 5) Write data to be processed (FMAC_DTR);
- 6) Wait for the completion interrupt or query the completion status (FMAC_STR.READY) to be 1;
- 7) Read result (FAMC_RTR0 or FAMC_RTR1);
- 8) Repeat 5) to 7) until all data has been processed.

39.4 Module Enable

The module enable bit (FMAC_ENR.FIREN) has two functions. The first function is to enable the entire module. Only when it is 1 , the module is allowed to work. The second function is the soft reset of the module. When a certain order and coefficient configuration is used up and another order and coefficient configuration needs to be changed, FIREN needs to be set to 0 to clear the residual information of the previous calculation. Then operate according to the operating procedures.

FIREN is reset to 0 , the configuration parameters are not cleared, only the internal registers and intermediate results are cleared.

39.5 Coefficient Normalization

If the output result is only 16 bits, then the total number of bits truncated during the calculation process and the calculation result is as high as 20 bits. Therefore, if the input data involved in the calculation is too small, the output result may be 0 or a very small number (with a large loss of precision). Therefore, it is necessary to normalize the filter coefficients, that is, normalize the maximum value (if it is a positive number) to 32767 (if it is a negative number, normalize it to -32768), and multiply the other coefficients by the same ratio.

For example, there are 5 coefficients 10 , 13 , 35 , 96 , 42 , and they need to be normalized. First find the maximum value, which is 96 , and then divide 32767 by the maximum value, $32767/96=341.3$, and take the value 341 , which is the normalized multiple. All other coefficients must be multiplied by this multiple to get the normalized number.

So the normalized coefficients are: 3410 , 4433 , 11935 , 32736 , 14322.

39.6 Interrupt and Event Description

39.6.1 Interrupt Output

After each operation is completed, a corresponding operation completion event will be generated. If the interrupt control bit is enabled (FMAC_IER.INTEN=1), the corresponding interrupt request signal (FMAC_m_FIR , m=1~4) will also be generated.

39.6.2 Event Output

After each operation is completed, a corresponding operation completion event will be generated, and the corresponding event request can be used to select and trigger other modules.

39.7 Register Description

Table 39-1 shows the register list of the FMAC module.

Table 39-1 FMAC Register List

BASE ADDR: 0x40058000(U1), 0x40058400(U2),
0x40058800(U3), 0x40058C00(U4)

Register Name	Symbol	Offset	Bit Width	Reset Value
Module enable register	FMAC_ENR	0x0000	32	0x00000000
Basic control register	FMAC_CTR	0x0004	32	0x00000010
Interrupt control register	FMAC_IER	0x0008	32	0x00000000
Data input register	FMAC_DTR	0x000C	32	0x00000000
Filter coefficient register 0	FMAC_COR0	0x0020	32	0x00000000
Filter coefficient register 1	FMAC_COR1	0x0024	32	0x00000000
-	-	-	-	-
Filter coefficient register 16	FMAC_COR16	0x0060	32	0x00000000
Result output register 0	FMAC_RTR0	0x0010	32	0x00000000
Result output register 1	FMAC_RTR1	0x0014	32	0x00000000
Operation status register 1	FMAC_STR	0x0018	32	0x00000000

39.7.1 Module Enable Register (FMAC_ENR)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved														FMACEN	
Bit	Flag	Bit Name	Function											Read and write	
b31~b1	Reserved	-	Read as "0", write as "0"											R/W	
b0	FMACEN	Module Enable	0: Disabled. 1 : Enable, that is, allow to work Note: This signal is used as the soft reset signal of this module, that is, when the filter order and filter coefficient need to be changed, the signal of this bit needs to be set to 0 first, and then set to 1											R/W	

39.7.2 Basic Control Register (FMAC_CTR)

Reset value: 0x00000010

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	SHIFT [4:0]				-	-	-	STAGE_NUM [4:0]					
Bit	Flag		Bit Name		Function						Read and write				
b31~b13	Reserved		-		Read as "0", write as "0"						R/W				
b12~b8	SHIFT [4:0]		Filtering result shift		The number of right shifts of the filtering result controls the output of FMAC_RTR1 00000: No shift, that is, select [31:0] of the operation result to FMAC_RTR1 00001: Shift right by 1 bit, that is, select [32:1] of the operation result to FMAC_RTR1 00010: Shift right by 2 bit, that is, select [33:2] of the operation result to FMAC_RTR1 10100 : right shift 20 bits, that is, select [35:20] of the operation result to FMAC_RTR1 , and move the high bit into the sign bit 10101 : right shift 21 bits, that is, select [36:21] of the operation result to FMAC_RTR1 , and move the high bit into the sign bit Other values: No shift						R/W				
b7~b5	Reserved		-		Read as "0", write as "0"						R/W				
b4~b0	STAGE_NUM [4:0]		Filtering order setting		Set the filtering order						R/W				

39.7.3 Interrupt Control Register (FMAC_IER)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved															INTEN
Bit		Flag		Bit Name		Function								Read and write	
b31~b1		Reserved		-		Read as "0", write as "0"								R/W	
b0		INTEN		Interrupt enable		0: No interrupt will be generated after the operation is completed 1 : Generate an interrupt after the operation is completed								R/W	

39.7.4 Data Input Register (FMAC_DTR)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
FMAC_DIN [15:0]															
Bit	Flag	Bit Name		Function		Read and write									
b31~b16	Reserved	-		Read as "0", write as "0"		R/W									
b15~b0	FMAC_DIN [15:0]	Data Entry		FIR data input		R/W									

39.7.5 Data Input Register (FMAC_DTR)...

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
FMAC_CIN [15:0]															
Bit	Flag	Bit Name		Function		Read and write									
b31~b16	Reserved	-		Read as "0", write as "0"		R/W									
b15~b0	FMAC_CIN [15:0]	Filter coefficient		Filter coefficient input		R/W									

39.7.6 Result Output Register 0 (FMAC_RTR0)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
FMAC_DOUT0 [31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
FMAC_DOUT0 [15:0]															
Bit	Flag	Bit Name		Function		Read and write									
b31~b0	FMAC_DOUT0 [31:0]	Result Output		32 bits of the accumulator result are output		R/W									

39.7.7 Result Output Register 1 (FMAC_RTR1)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
FMAC_DOUT1 [31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
FMAC_DOUT1 [15:0]															
Bit	Flag		Bit Name		Function										Read and write
b31~b0	FMAC_DOUT1 [31:0]		Result Output		32 bits of the accumulator result are output										R/W

39.7.8 Operation Status Register (FMAC_STR)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
READY	Reserved														
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved															
Bit	Flag		Bit Name		Function										Read and write
b31	READY		Operation completed bit		0: The calculation is not completed and the result is not available. 1 : Calculation completed, result available Note: This register is automatically cleared after reading										R/W
b30~b0	Reserved		-		Read as "0", write as "0"										R/W

40 Data Computing Unit (CRC)

40.1 Summary

The Data Computing Unit (Data Computing Unit) is a module that simply processes data without the help of a CPU. Each DCU unit has 3 data registers, which can add, subtract and compare the size of 2 data, as well as the window comparison function. This product is equipped with 3 DCU units, and each unit can independently complete its own functions.

Functional Summary:

- Can carry out 4 kinds of data processing: addition, subtraction, comparison of 2 data and comparison of 3 data windows.
- Addition and subtraction operate on the data in the DATA0 and DATA1 registers, and the result is stored in DATA0.
- Addition and subtraction can be calculated after writing registers or triggered by other peripheral circuit events.
- The addition and subtraction operations can automatically halve the results once, and put the halved results and the addition and subtraction results into two data registers for use by other modules.
- The comparison mode can compare two data between DATA0 and DATA1 registers, and between DATA0 and DATA2 registers. You can choose to generate interrupts and flags when greater than, less than, or equal to.
- The comparison mode can be used for window comparison, that is, set DATA1 and DATA2 as the upper and lower limits of the window respectively, and judge whether DATA0 is inside or outside the window according to the comparison results of DATA0 and DATA1 and DATA0 and DATA2.
- It can be triggered by other peripheral circuit events to perform calculations and generate various interrupts and event signals based on the calculation results. When other peripheral circuits with hardware trigger start function select DCU as the trigger source, DCU generates event signal to start the peripheral circuit to start action.

40.2 Functional Description

40.2.1 Addition Mode

The addition mode calculates the sum of DATA0 and DATA1, where DATA0 is used as the summand and DATA1 is used as the addend. Every time the DATA1 register is written, a $(DATA0+DATA1)/2$ operation is performed, the result of $DATA0+DATA1$ is stored in DATA0, and the result of $(DATA0+DATA1)/2$ is stored in DATA2. When the result of $DATA0+DATA1$ exceeds 0xFF (8bit mode) or 0xFFFF (16bit mode) or 0xFFFFFFFF (32bit mode), a flag bit is generated and an interrupt is generated.

Additive mode application example:

1. The control register DCU_CTL selects the addition mode, and the data width is 16 bits
2. Interrupt condition selection register DCU_INTSEL selects the operation condition
3. Write 0xFF00 and 0x55 in DATA0 and DATA1 respectively, at this time the calculation result is 0xFF55, and the result is saved in DATA0.
4. 0xFF continues to be written into DATA1. At this time, the calculation result overflows and a result flag is generated. Read the flag register DCU_FLAG to get the result flag
5. Write flag reset register DCU_FLAGCLR to clear flag bit

40.2.2 Subtraction Mode

The subtraction mode calculates the difference between DATA0 and DATA1, where DATA0 acts as the minuend and DATA1 acts as the subtrahend. Every time the DATA1 register is written, a $(DATA0-DATA1)/2$ operation is performed, the result of $DATA0-DATA1$ is stored in DATA0, and the result of $(DATA0-DATA1)/2$ is stored in DATA2. When the result of $DATA0-DATA1$ is less than 0x0 (8bit, 16bit, 32bit mode), a flag bit is generated and an interrupt is generated.

40.2.3 Hardware Triggered Boot Mode

The DCU can trigger and start operations according to events generated by peripheral circuits. When using the hardware trigger start mode, it is necessary to first enable the peripheral circuit trigger function enable bit of the function clock control 0 register (FCG0). Each DCU unit can independently select the trigger start signal sent by other peripheral circuits. When selecting the start signal, write the number of the start source of the peripheral circuit to be selected in the trigger source selection register (DCU_TRGSEL). When the peripheral circuit event occurs, the event signal will be input to the DCU and trigger the DCU to start computing. The hardware trigger start mode includes trigger plus mode and trigger minus mode. In the trigger plus mode, every time an event trigger occurs, the DCU will start and perform a calculation of $(DATA0+DATA1)/2$, the result of $DATA0+DATA1$ is stored in DATA0, and the result of $(DATA0+DATA1)/2$ is stored in DATA2. When the result of $DATA0+DATA1$ exceeds 0xFF (8bit mode) or 0xFFFF (16bit mode) or 0xFFFFFFFF (32bit mode), a flag bit is generated and an interrupt is generated. In the trigger subtraction mode, every

time an event trigger occurs, the DCU will start and perform a calculation of $(DATA0-DATA1)/2$, the result of $DATA0-DATA1$ is stored in $DATA0$, and the result of $(DATA0-DATA1)/2$ is stored in $DATA2$. When the result of $DATA0-DATA1$ is less than 0x0 (8bit, 16bit, 32bit mode), a flag bit is generated and an interrupt is generated.

Hardware trigger start mode application example:

1. The control register `DCU_CTL` selects the trigger plus mode, and the data width is 8 bits
2. Interrupt condition selection register `DCU_INTSEL` selects the operation condition
3. Write 0x00 and 0x56 to `DATA0` and `DATA1` respectively
4. Write the event number in the trigger source selection register `DCU_TRGSEL`
5. Make the peripheral circuit act and generate an event, the DCU is triggered by the event and performs an addition operation, and the result is 0x56 and stored in `DATA0`.
6. After three consecutive triggers, the calculation result overflows and a result flag is generated. Read the flag register `DCU_FLAG` to get the result flag
7. Write flag reset register `DCU_FLAGCLR` to clear flag bit

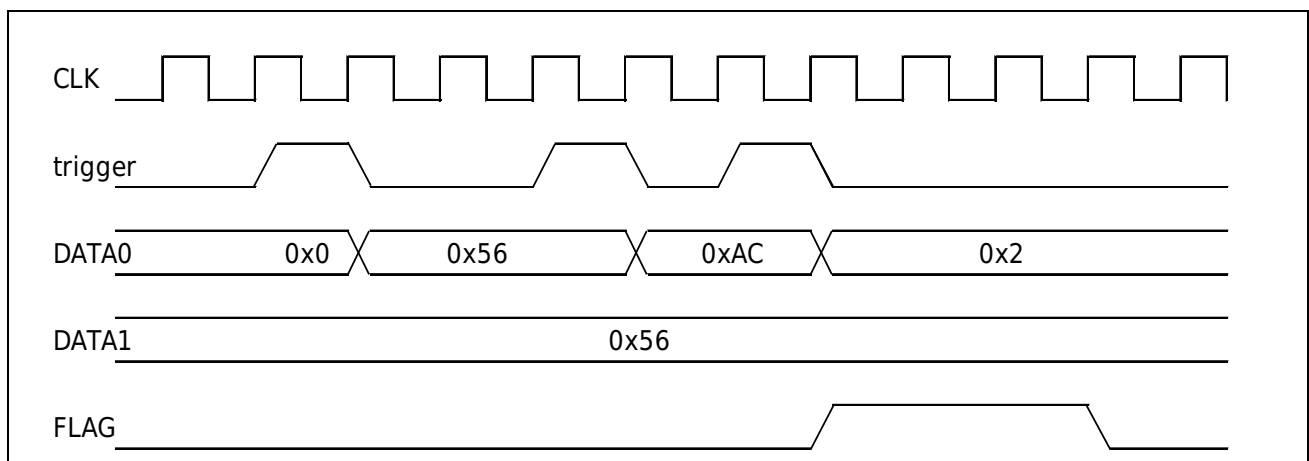


Figure 40-1 DCU Hardware Trigger Start

40.2.4 Comparative Mode

Comparing the size of $DATA0$ and $DATA1$ and $DATA0$ and $DATA2$, you can select when $DATA0$ is greater than $DATA1$, $DATA0$ is less than $DATA1$, $DATA0$ is equal to $DATA1$ and when $DATA0$ is greater than $DATA2$, $DATA0$ is less than $DATA2$, $DATA0$ is equal to $DATA2$, generate a flag and generate an interrupt. In the comparison mode, you can choose the conditions for data comparison, compare after writing $DATA0$ or write any data register.

Comparison mode application example:

1. The control register `DCU_CTL` selects the comparison mode, and the data width is 16 bits. The comparison starts after writing $DATA0$.
2. The interrupt condition selection register `DCU_INTSEL` generates a flag when $DATA0 > DATA1$.

3. 0xB BBB and 0xA AA are written into DATA1 and DATA0 respectively. At this time, no flag is generated because DATA0 > DATA1 is not satisfied.
4. When 0x8888 is written to DATA1 , although DATA0 > DATA1 is satisfied , no flag is generated because the comparison is started after DATA0 is written.
5. After DATA0 is written with 0x9999, the flag generation condition is met and the flag is generated.
6. Write flag reset register DCU_FLAGCLR to reset flag bit

40.2.5 Interrupt and Event Signal Output

DCU has a variety of interrupts and event outputs for triggering and starting other peripheral circuits for users to choose. The control of interrupt and event output is controlled by the interrupt and event register (DCU_INTEVT). When the event signal needs to be output, the user needs to enable the corresponding control position of the interrupt and event register (DCU_INTEVT). Each DCU unit outputs a DCU event signal, which are DCU1~DCU3 in the event list. When an interrupt is required when a corresponding event occurs, the user needs to enable the corresponding control bit of the interrupt and event register (DCU_INTEVT) and set the INTEN bit of the control register (DCU_CTL) to 1. Each DCU unit outputs a DCU interrupt signal, which are DCU1~DCU3 in the interrupt list.

40.3 Register Description

Table 40-1 Register List

Unit 1

Name	Abbreviation	Description	Address
DCU1 Control Register	DCU1_CTL	Configure the action mode of the DCU	0x40056000
DCU1 flag register	DCU1_FLAG	DCU Result Identification	0x40056004
DCU1 Data Register 0	DCU1_DATA0	Store operation data	0x40056008
DCU1 Data Register 1	DCU1_DATA1	Store operation data	0x4005600C
DCU1 Data Register 2	DCU1_DATA2	Store operation data	0x40056010
DCU1 Flag Reset Register	DCU1_FLAGCLR	Clear the result flag of DCU	0x40056014
DCU1 Interrupt and Event Registers	DCU1_INTEVT	DCU interrupt and event conditions	0x40056018

Unit 2

Name	Abbreviation	Description	Address
DCU2 Control Register	DCU2_CTL	Configure the action mode of the DCU	0x40056400
DCU2 flag register	DCU2_FLAG	DCU Result Identification	0x40056404
DCU2 data register 0	DCU2_DATA0	Store operation data	0x40056408
DCU2 data register 1	DCU2_DATA1	Store operation data	0x4005640C
DCU2 data register 2	DCU2_DATA2	Store operation data	0x40056410
DCU2 Flag Reset Register	DCU2_FLAGCLR	Clear the result flag of DCU	0x40056414
DCU2 Interrupt and Event Registers	DCU2_INTEVT	DCU interrupt and event conditions	0x40056418

Unit 3

Name	Abbreviation	Description	Address
DCU3 Control Register	DCU3_CTL	Configure the action mode of the DCU	0x40056800
DCU3 flag register	DCU3_FLAG	DCU Result Identification	0x40056804
DCU3 data register 0	DCU3_DATA0	Store operation data	0x40056808
DCU3 data register 1	DCU3_DATA1	Store operation data	0x4005680C
DCU3 data register 2	DCU3_DATA2	Store operation data	0x40056810
DCU3 Flag Reset Register	DCU3_FLAGCLR	Clear the result flag of DCU	0x40056814
DCU3 Interrupt and Event Registers	DCU3_INTEVT	DCU interrupt and event conditions	0x40056818

40.3.1 DCU Control Register (DCUx_CTL)(x=1~3)

Register Description: This register is used to configure the action mode of the DCU.

Reset value: 0x80000000

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

Bit	Flag	Bit Name	Function	Read and write
b31	INTEN	Interrupt enable	0: Interrupts are not allowed 1: Enable interrupt generation	R/W
b30~b9	Reserved	-	Read 0 when reading, write 0 when writing	R/W
b8	COMPTRG	Timing of compare mode trigger comparison	0: compare after writing DATA0 1: compare after writing DATA0 or DATA1 or DATA2	R/W
b7~b6	Reserved	-	Read 0 when reading, write 0 when writing	R/W
b5~b4	DATASIZE [1:0]	Addition / subtraction mode, comparison mode data size	00: 8bit 01: 16bit 10: 32bit	R/W
b3~b0	MODE [2:0]	Action mode	0000: DCU invalid 0001: Addition mode, the operation is performed after data is written in the DATA1 register 0010: Subtraction mode, the operation is performed after data is written in the DATA1 register 0011: Hardware trigger addition mode, triggered by other peripheral circuits to start addition operation 0100: The hardware triggers the subtraction mode, which is triggered by other peripheral circuits to start the subtraction operation 0101: Compare Mode Other: Prohibitions	R/W

40.3.2 DCU Flag Register (DCUx_FLAG) (x=1~3)

Register Description: This register generates the result flag of the DCU.

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved									FLAG_GT1	FLAG_EQ1	FLAG_LS1	FLAG_GT2	FLAG_EQ2	FLAG_LS2	FLAG_OP
Bit	Flag	Bit Name		Function										Read and write	
b31~b7	Reserved	-		Read 0 when reading, write 0 when writing										R/W	
b6	FLAG_GT1	greater	than	In comparison mode, set when DATA0>DATA1, and clear when the CLR_GT1 bit of the DCU flag reset register DCU_FLAGCLR is written to 1										R	
b5	FLAG_EQ1	equal to flag 1		In comparison mode, set when DATA0=DATA1, and clear when the CLR_EQ1 bit of the DCU flag reset register DCU_FLAGCLR is written to 1										R	
b4	FLAG_LS1	less than flag 1		In comparison mode, set when DATA0<DATA1, and clear when the CLR_LS1 bit of the DCU flag reset register DCU_FLAGCLR is written to 1										R	
b3	FLAG_GT2	greater	than	In comparison mode, set when DATA0>DATA2, and clear when the CLR_GT2 bit of the DCU flag reset register DCU_FLAGCLR is written to 1										R	
b2	FLAG_EQ2	equal to flag 2		In comparison mode, set when DATA0=DATA2, and clear when the CLR_EQ2 bit of the DCU flag reset register DCU_FLAGCLR is written to 1										R	
b1	FLAG_LS2	less than flag 2		In comparison mode, when DATA0<DATA2 is set, it is cleared when the CLR_LS2 bit of the DCU flag reset register DCU_FLAGCLR is written to 1										R	
b0	FLAG_OP	Operation flag		Addition, subtraction, and trigger addition, trigger subtraction mode, set when addition overflows or subtraction underflows, and clear when the CLR_OP bit of the DCU flag reset register DCU_FLAGCLR is written to 1										R	

40.3.3 DCU Data Register (DCUx_DATAy) (x=1~3,y=0,1,2)

The functions of each data register in each mode are as follows

-	DATA0	DATA1	DATA2
Addition Mode	Summand/store result	Addend	Store the halving result
trigger plus mode	Summand/store result	Addend	Store the halving result
Subtraction mode	Minuend/store result	Subtrahend	Store the halving result
Trigger minus mode	Minuend/store result	Subtrahend	Store the halving result
Compare mode	Compared object	Comparison object 1	Comparison object 2
Compare mode (window compare)	Compared object	Window cap	Window lower limit

Reset value: 0x00000000

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

Addition / subtraction mode, trigger addition / subtraction mode, in comparison mode, DCUx_DATAy (x=1~3, y=0,1,2) is used to store calculation data.

Bit	Flag	Bit Name	Function	Read and write
b31~b0	DAT [31:0]	Operational data	To store calculation data, the actual number of digits used is set according to DCU_CTL.DATASIZE, DATA [7:0] is valid data when DCU_CTL.DATASIZE=00, DATA [15:0] is valid data when DCU_CTL.DATASIZE=01, DATA [31:0] is valid data when DCU_CTL.DATASIZE=10	R/W

40.3.4 DCU Flag Reset Register (DCUx_FLAGCLR) (x=1~3)

Register Description: This register is used to clear the result flag of DCU.

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved									CLR GT1	CLR EQ1	CLR LS1	CLR GT2	CLR EQ2	CLR LS2	CLR OP

Bit	Flag	Bit Name	Function	Read and write
b31~b7	Reserved	-	Read 0 when reading, write 0 when writing	R/W
b6	CLR_GT1	Clear greater than flag bit 1	When writing 1, clear the FLAG_GT1 bit of DCU_FLAG	W
b5	CLR_EQ1	Clear equals flag bit 1	When writing 1, clear the FLAG_EQ1 bit of DCU_FLAG	W
b4	CLR_LS1	Clear less than flag bit 1	When writing 1, clear the FLAG_LS1 bit of DCU_FLAG	W
b3	CLR_GT2	Clear greater than flag bit 2	When writing 1, clear the FLAG_GT2 bit of DCU_FLAG	W
b2	CLR_EQ2	Clear equals flag bit 2	When writing 1, clear the FLAG_EQ2 bit of DCU_FLAG	W
b1	CLR_LS2	Clear less than flag bit 2	When writing 1, clear the FLAG_LS2 bit of DCU_FLAG	W
b0	CLR_OP	Clear operation flag	Writing 1 clears the FLAG_OP bit of DCU_FLAG	W

40.3.5 DCU Interrupt and Event Register (DCUx_INTEVTSEL) (x=1~3)

Register Description: This register can select the conditions under which the DCU generates an interrupt and outputs an event signal

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved							SEL_WIN [1:0]	SEL GT1	SEL EQ1	SEL LS1	SEL GT2	SEL EQ2	SEL LS2	SEL OP	

Bit	Flag	Bit Name	Function	Read and write
b31~b9	Reserved	-	Read 0 when reading, write 0 when writing	R/W
b8~b7	SEL_WIN [1:0]	Window comparison condition selection	In comparison mode, an interrupt and an output event signal are generated when the window comparison condition set by INT_WIN is met. When the INT_WIN setting is valid, an interrupt and an output event signal are not generated when other comparison conditions are met. 00: Do not generate window comparison interrupt and output event signal. Under this setting, select other interrupt and event signal generation conditions by b1~b6 of this register 01: Generate interrupt and output event signal when DATA0 data is in the window, that is, DATA2≤DATA0≤DATA1 10: Generate interrupt and output event signal when DATA0 data is outside the window, that is, DATA0>DATA1 or DATA0<DATA2 11: Does not generate any interrupt and event signal in compare mode	R/W
b6	SEL_GT1	Greater than condition, select 1	1 : In compare mode and when INT_WIN=00 , an interrupt is generated and an event signal is output when DATA0>DATA1 0: In comparison mode and when INT_WIN=00, no interrupt or output event signal is generated when DATA0>DATA1 This bit is invalid when INT_WIN≠00	R/W
b5	SEL_EQ1	Equal to condition selection 1	1 : In compare mode and when INT_WIN=00 , an interrupt is generated and an event signal is output when DATA0=DATA1 0: In comparison mode and when INT_WIN=00, no interrupt or output event signal is generated when DATA0=DATA1 This bit is invalid when INT_WIN≠00	R/W
b4	SEL_LS1	Less than condition selection 1	1 : In compare mode and when INT_WIN=00 , an interrupt is generated and an event signal is output when DATA0 < DATA1 0: In comparison mode and when INT_WIN=00, no interrupt or output event signal is generated when DATA0<DATA1 This bit is invalid when INT_WIN≠00	R/W
b3	SEL_GT2	Greater than condition, select 2	1 : In compare mode and when INT_WIN=00 , an interrupt is generated and an event signal is output when DATA0>DATA2 0: In comparison mode and when INT_WIN=00, no interrupt or output event signal is generated when DATA0>DATA2 This bit is invalid when INT_WIN≠00	R/W
b2	SEL_EQ2	Equal to condition selection 2	1 : In compare mode and when INT_WIN=00 , an interrupt is generated and an event signal is output when DATA0=DATA2 0: In comparison mode and when INT_WIN=00, no interrupt or output event signal is generated when DATA0=DATA2 This bit is invalid when INT_WIN≠00	R/W
b1	SEL_LS2	Less than condition selection 2	1 : In compare mode and when INT_WIN=00 , an interrupt is generated and an event signal is output when DATA0 < DATA2 0: In comparison mode and when INT_WIN=00, no interrupt or output event signal is generated when DATA0<DATA2 This bit is invalid when INT_WIN≠00	R/W
b0	SEL_OP	Operation condition selection	1: Generate interrupt and output event signal when the operation result overflows or underflows in addition and subtraction mode 0: Generate interrupt and output event signal when the operation result overflows or underflows in addition and subtraction mode	R/W

41 Mathematical Arithmetic Unit (MAU)

41.1 Introduction

The Mathematical Calculation Unit (MAU) is a hardware accelerated computing module that includes two types of operations: square root and sine operations. It supports square root and sine operations of fixed-point numbers. The sine function supports $360^\circ/2^{12}$ calculation accuracy.

41.2 Functional Description

41.2.1 Square Root Operation

The square root operation supports 32 -bit fixed-point number input, and the operation result is a 17- bit fixed-point number output. The operation execution period will vary depending on the size of the input data.

41.2.1.1 Operating Procedures

When the interrupt is invalid (MAU_CSR.INTEN=0), the operation flow of the square root operation is as follows; when the interrupt is valid, the user customizes the interrupt processing process according to this flow:

- 1) Write the square root data into the data input register 0 (MAU_DTR0)
- 2) Set the start bit of the control status register to 1 (MAU_CSR.START=1)
- 3) After the operation starts, the status bit of the control status register (MAU_CSR.BUSY) will be set until the operation is completed, and the bit will be automatically cleared. Expect the BUSY bit to clear, indicating that the operation is complete
- 4) Read the result from the result output register 0 (MAU_RTR0)
- 5) Repeat the above process 1) ~ 4)

Regarding process 3), after MAU_CSR.START is set, the BUSY bit starts to go high and automatically becomes 0 after 8 to 16 operation clock periods (the operation time varies depending on the size of the input data). When using the square root operation module, users can flexibly judge the current operation status according to the changes of this bit.

41.2.1.2 Result Processing

The result of the square root operation is a 17 -bit data output, and the least significant bit of the result is obtained by rounding. For example, when the input data is 0x0000000CH , the result is 0x00003H; when the input data is 0x0000000DH , the result is 0x00004H.

41.2.1.3 Fixed-point Number Processing

When the input data is a fixed-point decimal and the format of the input and output data needs to be kept consistent, the user can process the output data accordingly by controlling the shift control bit (MAU_CSR.SHIFT) of the status register.

For example, when using the Q14.2 format for fixed-point arithmetic, you can set MAU_CSR.SHIFT=1 to make the output result also follow the Q14.2 format. The fixed-point number 6.25 (0x00000019H in Q14.2 format) has output result bit 5 (0x00000005H) before shifting , and the result after shifting 1 bit is 2.5 (0x0000000AH in Q14.2 format).

41.2.2 Sine Operation

The sine operation supports 12 -bit data input and the operation result is a 16- bit signed number output. The 12 -bit data input represents the rectangular coordinate system angle to be calculated; in the 16- bit data output, the highest bit represents the positive or negative sign of the result.

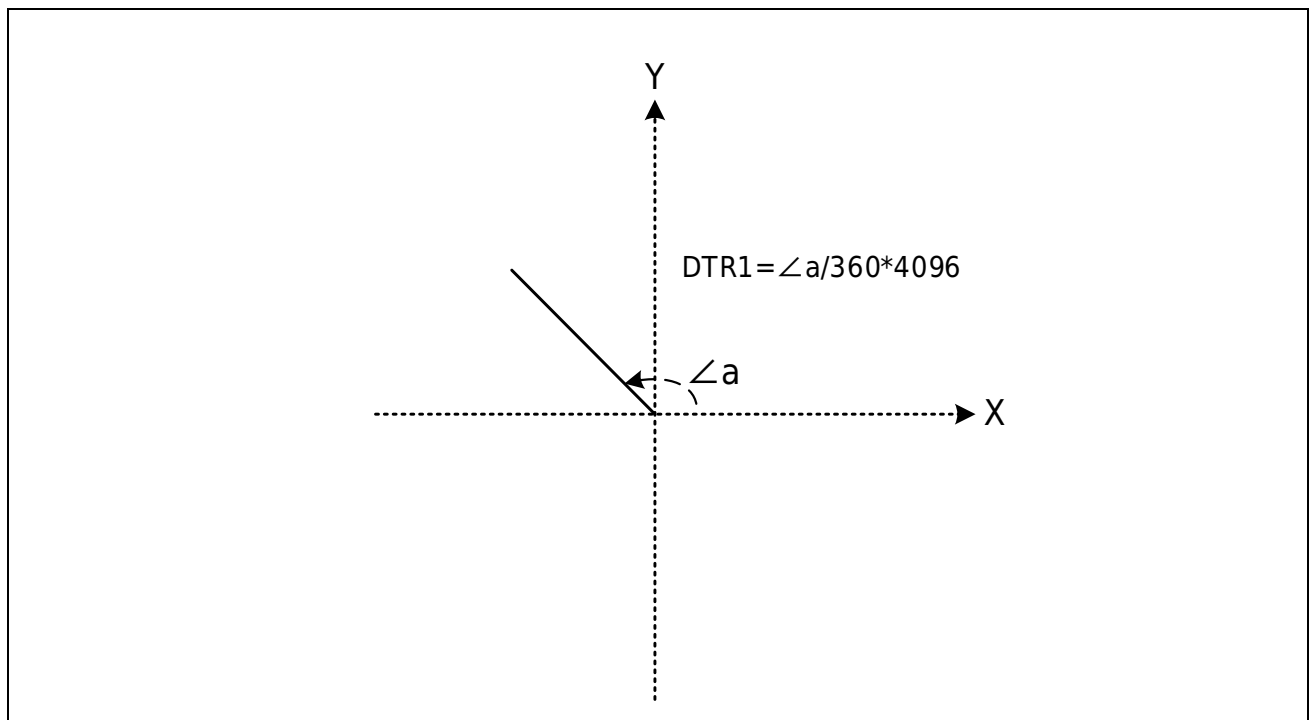


Figure 41-1 Schematic Diagram Of Sine Operation Angle Value

41.2.2.1 Operating Procedures

The operation process of sine operation is as follows:

- 1) Write the angle value in the data input register 1 (MAU_DTR1)
- 2) After 1 period, the operation result is read from the result output register 1 (MAU_RTR1)
- 3) Repeat the above process 1) ~ 2)

41.3 Interrupt and Event Description

41.3.1 Interrupt Output

After each square root operation is completed, a corresponding operation completion event will be generated. If the interrupt control bit is enabled (MAU_CSR.INTEN=1), the corresponding interrupt request signal (MAU_SQRT) will also be generated.

41.3.2 Event Output

After each square root operation is completed, a corresponding operation completion event will be generated, and the corresponding event request signal (MAU_SQRT) will also be output, which can be used to select and trigger other modules.

41.4 Register Description

Table 41-1 shows the register list of the MAU module.

Table 41-1 MAU Register List

BASE ADDR: 0x40055000

Register Name	Symbol	Offset	Bit Width	Reset Value
Control status register	MAU_CSR	0x0000	32	0x00000000
Data input register 0	MAU_DTR0	0x0004	32	0x00000000
Result output register 0	MAU_RTR0	0x000C	32	0x00000000
Data input register 1	MAU_DTR1	0x0010	32	0x00000000
Result output register 1	MAU_RTR1	0x0014	32	0x00000000

41.4.1 Control Status Register (MAU_CSR)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	SHIFT [4:0]				-	-	-	-	BUSY	-	INTEN	START	

Bit	Flag	Bit Name	Function	Read and write
b31~b13	Reserved	-	Read as "0", write as "0"	R/W
b12~b8	SHIFT [4:0]	Shift the square root result	00000: The operation result is not shifted and output 00001: The operation result is shifted left by 1 bit and then output 00010: The calculation result is shifted 2 bits to the left and then output ... 10000: The calculation result is shifted 16 bits to the left and then output Please do not set other values	R/W
b7~b4	Reserved	-	Read as "0", write as "0"	R/W
b3	BUSY	Square root operation status	0: The square root operation has not started or is completed 1 : Square root operation in progress	R
b2	Reserved	-	Read as "0", write as "0"	R/W
b1	INTEN	Square root interrupt enable	0: No interrupt will be generated after the square root operation is completed 1 : An interrupt is generated after the square root operation is completed	R/W
b0	START	Square root operation starts	0: Writing zero to this bit is invalid 1 : Square root operation starts Note: When writing 1 to this bit, a START request is generated internally, and it is always 0 when reading.	R/W

41.4.2 Data Input Register 0 (MAU_DTR0)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SQRT_DIN [31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SQRT_DIN [15:0]															
Bit		Flag		Bit Name		Function								Read and write	
b31~b0		SQRT_DIN [31:0]		Root		Root input								R/W	

41.4.3 Result Output Register 0 (MAU_RTR0)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															SQRT_DOUT [16]
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SQRT_DOUT [15:0]															
Bit		Flag			Bit Name		Function							Read and write	
b31~b17		Reserved			-		Read as "0", write as "0"							R/W	
b16~b0		SQRT_DOUT [16:0]			Prescription results		Square root result output							R/W	

41.4.4 Data Input Register 1 (MAU_DTR1)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved				SIN_DIN [11:0]											
Bit		Flag		Bit Name		Function								Read and write	
b31~b12		Reserved		-		Read as "0", write as "0"								R/W	
b11~b0		SIN_DIN [11:0]		Angle setting		Cartesian coordinate system angle input								R/W	

41.4.5 Result Output Register 1 (MAU_RTR1)

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SIN_DOUT [15:0]															
Bit	Flag	Bit Name	Function										Read and write		
b31~b16	Reserved	-	Read as "0", write as "0"										R/W		
b15~b0	SIN_DOUT [15:0]	Sine operation result	Sine operation result output										R/W		

42 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

42.1 Introduction

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

42.2 DBGC System Block Diagram

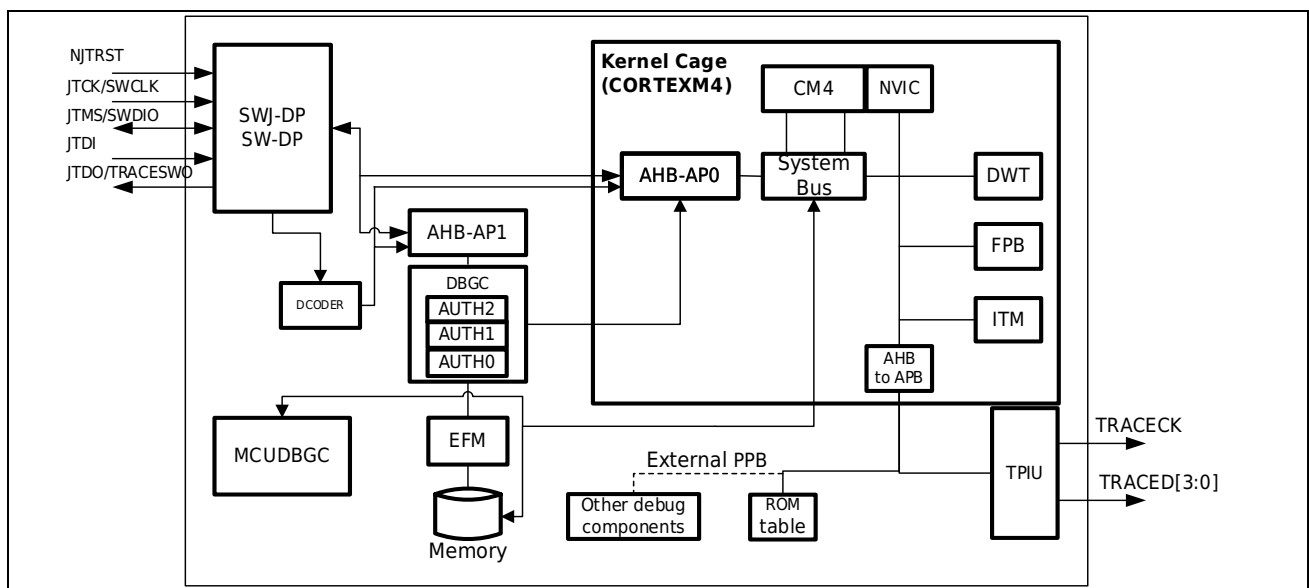


Figure 42-1 Debugging Control Systems

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.

42.3 SWJ-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.

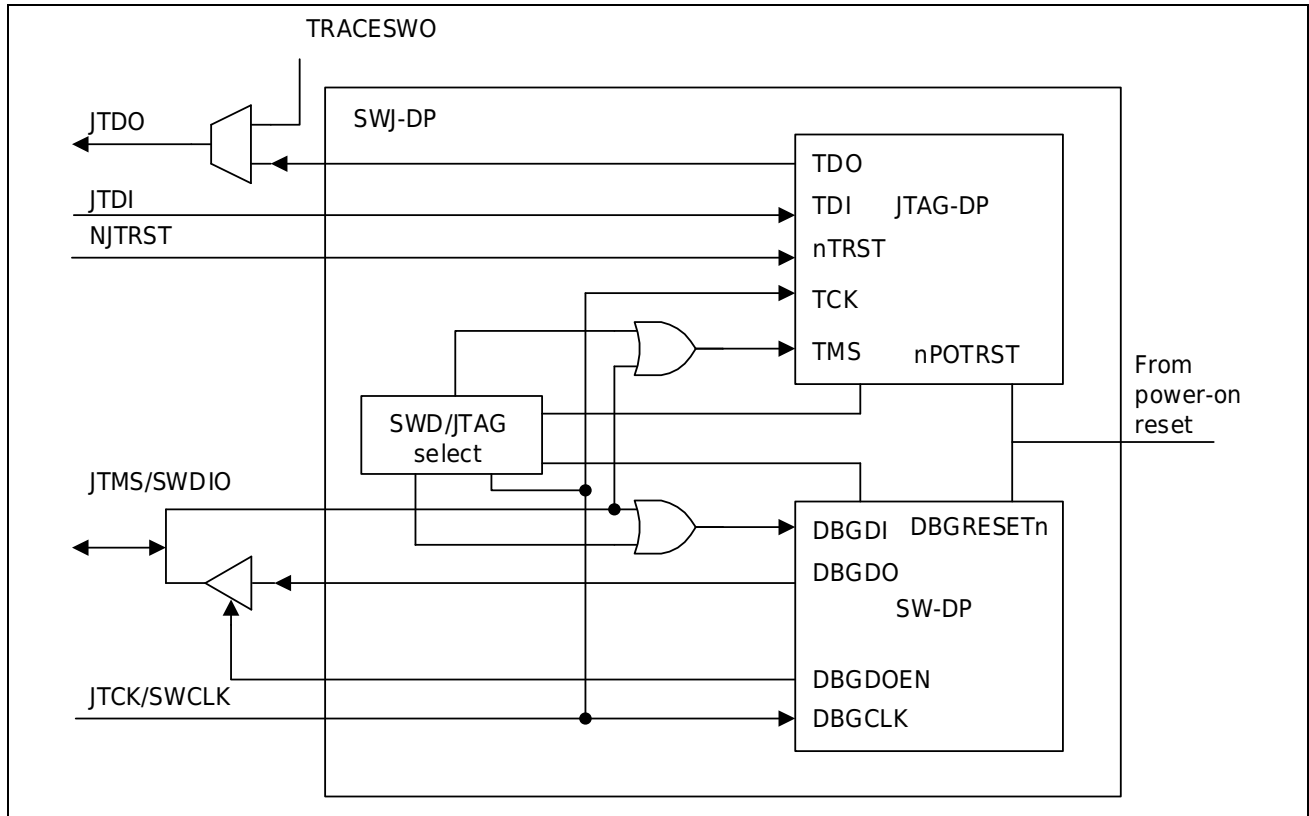


Figure 42-2 Debugging Control Systems

42.3.1 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 periods
2. Output 16 JTMS (SWDIO) signals 0111_1001_1110_0111 (MSB)
3. Output JTMS (SWDIO) = 1 signal for more than 50 JTCK cycles

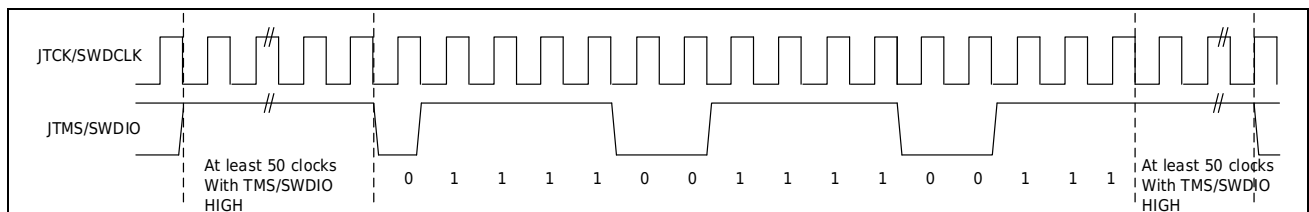


Figure 42-3 JTAG-DP To SW-DP Switching Timing

42.4 Pinout 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.

42.4.1 SWJ Debug Port Pin

5 common I/O ports of MCU can be used as SWJ-DP interface pins.

Table 42-1 SWJ Debug Port Pins

SWJ-DP Pin Name	JTAG Debug Port		SW debug port	
	Types	Description	Types	Debug Allocation
JTMS/SWDIO	I	JTAG mode selection	I/O	Serial data input/output
JTCK/SWCLK	I	JTAG clock	I	Serial clock
JTDI	I	JTAG data input	-	-
JTDO/TRACESWO	O	JTAG data output	-	TRACESWO (if asynchronous tracing is enabled)
NJTRST	I	JTAG reset	-	-

42.4.2 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 SWJ-DP port pins, see: General IO Special Control Register PSPCR.

Table 42-2 Flexible SWJ-DP Pinout

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 JTAG-DP and Disable SW-DP	Releasable	Releasable	Releasable	Releasable	Releasable

42.4.3 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: high-impedance state

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.

42.4.4 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 host does the following:

- In system reset state, assign all SWJ pins (JTAG-DP+SW-DP).
- In system reset state, the debug host sends a JTAG sequence to switch from JTAG-DP to SW-DP.
- Still in the system reset state, the debug host sets a breakpoint at the reset address.
- The reset signal is released and the core stops at the reset address.
- From this debug port switch to SW-DP. The other JTAG pins are then reassigned as GPIOs by user software.

Note:

- For user software design, when it is necessary to release the debug pins, these pins are still input pull-ups after reset until the user software releases the pins (NJTRST, JTMS, JTDI, JTDO, and JTCK)

42.5 Register Description

Table 42-3 Register List

Base address: 0xE0042000

Register name	Symbol	Offset Address	Bit Width	Initial value	Access host
Chip specific flag register	CHIPID	0x000C	32	0x48440682	Debug IDE
DBG status register	MCUDBGSTAT	0x001C	32	0x00000000	CPU/Debug IDE*
Peripheral debugging pause register	MCUSTPCTL	0x0020	32	0x0000003B	CPU/Debug IDE*
Debug component configuration register	MCUTRACECTL	0x0024	32	0x00000000	CPU/Debug IDE*
Peripheral debugging pause register	MCUSTPCTL2	0x0028	32	0x00000000	CPU/Debug IDE*

Note:

- Registers are located in the PPB area and can only be accessed by the CPU in privileged mode.

42.5.1 Chip Specific Identification Register (CHIPID)

Used to indicate company information and product model information.

Reset value: 0x4844 0682

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	Flag	Bit Name	Function	Read and write											
b31~0	CHIPID [31:0]	Chip Specific Flag	CHIPID [31:16]: Corresponding ASCII code of HD, 0x4844 CHIPID [15:0]: Product model, 0x0682	R											

42.5.2 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
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	CDBG PWRUPACK	CDBG PWRUPREQ

Bit	Flag	Bit Name	Function	Read and write
b31~2	Reserved	-	Read as "0", write as "0"	R/W
b1	CDBGPWRUPACK	Power-on feedback of the debugger	0: No feedback 1: Commissioning Power-on Feedback	R/W
b0	CDBGPWRUPREQ	Debugger power-on request	0: No power-on request 1: Power-on request	R/W

42.5.3 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
-	-	-	-	-	-	-	M24 STP	M24 STP	M22 STP	M21 STP	M20 STP	M19 STP	M18 STP	M17 STP	M16 STP
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
M15 STP	-	-	M12 STP	M11 STP	M10 STP	M09 STP	M08 STP	M07 STP	M06 STP	-	-	-	RTC STP	WDT STP	SWD TSTP

Bit	Flag	Bit Name	Function	Read and write
b31	Reserved	-	Read as "0", write as "0"	R/W
b30	Reserved	-	Read as "0", write as "0"	R/W
b29	Reserved	-	Read as "0", write as "0"	R/W
b28	Reserved	-	Read as "0", write as "0"	R/W
b27	Reserved	-	Read as "0", write as "0"	R/W
b26	Reserved	-	Read as "0", write as "0"	R/W
b25	Reserved	-	Read as "0", write as "0"	R/W
b24	M24STP	TMR6_10 count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	R/W
b23	M23STP	TMR6_9 Count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	R/W
b22	M22STP	TMR6_8 Count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	R/W
b21	M21STP	TMR6_7 Count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	R/W
b20	M20STP	TMR6_6 Count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	R/W
b19	M19STP	TMR6_5 Count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	R/W
b18	M18STP	TMR6_4 Count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	R/W
b17	M17STP	TMR6_3 Count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	R/W
b16	M16STP	TMR6_2 Count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	R/W
b15	M15STP	TMR6_1 Count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	R/W
b14	Reserved	-	Read as "0", write as "0"	R/W
b13	Reserved	-	Read as "0", write as "0"	R/W
b12	M12STP	TMR4_1 count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	R/W
b11	M11STP	TMR2_4 Count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	R/W
b10	M10STP	TMR2_3 Count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	R/W
b9	M09STP	TMR2_2 Count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	R/W
b8	M08STP	TMR2_1 Count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	R/W
b7	M07STP	TMR0_2 Count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	R/W
b6	M06STP	TMR0_1 Count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	R/W
b5~3	Reserved	-	Read as "1", write as "1"	R/W
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	R/W

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	R/W
b0	SWDTSTP	SWDT count pause signal	0: SWDTcounter still counts even if the kernel stops 1: SWDTcounter pause count when kernel stops	R/W

42.5.4 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
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	-	TRACE_I _{OE} N	TRACE_ MODE	

Bit	Flag	Bit Name	Function	Read and write
b31~b3	Reserved	-	Read as "0", write as "0"	R/W
b2	TRACE_I _{OE} N	TRACE pin output control	0: Synchronous tracking pin output disabled 1: Synchronous trace pin output permission	R/W
b1~b0	TRACE_MODE	TRACED output pin control	00: asynchronous tracking 01: Synchronous tracking 1 bit TRACED [0] 10: Synchronous tracking 2-bit TRACED [1:0] 11: Synchronous tracking 4-bit TRACED [3:0]	R/W

42.5.5 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
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	M37 STP	M36 STP	M35 STP	M34 STP	M33 STP	M32 STP

Bit	Flag	Bit Name	Function	Read and write
b31	Reserved	-	Read as "0", write as "0"	R/W
b30	Reserved	-	Read as "0", write as "0"	R/W
b29	Reserved	-	Read as "0", write as "0"	R/W
b28	Reserved	-	Read as "0", write as "0"	R/W
b27	Reserved	-	Read as "0", write as "0"	R/W
b26	Reserved	-	Read as "0", write as "0"	R/W
b25	Reserved	-	Read as "0", write as "0"	R/W
b24	Reserved	-	Read as "0", write as "0"	R/W
b23	Reserved	-	Read as "0", write as "0"	R/W
b22	Reserved	-	Read as "0", write as "0"	R/W
b21	Reserved	-	Read as "0", write as "0"	R/W
b20	Reserved	-	Read as "0", write as "0"	R/W
b19	Reserved	-	Read as "0", write as "0"	R/W
b18	Reserved	-	Read as "0", write as "0"	R/W
b17	Reserved	-	Read as "0", write as "0"	R/W
b16	Reserved	-	Read as "0", write as "0"	R/W
b15	Reserved	-	Read as "0", write as "0"	R/W
b14	Reserved	-	Read as "0", write as "0"	R/W
b13	Reserved	-	Read as "0", write as "0"	R/W
b12~6	Reserved	-	Read as "0", write as "0"	R/W
b5	M37STP	TMRA_6 count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	R/W
b4	M36STP	TMRA_5 Count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	R/W
b3	M35STP	TMRA_4 Count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	R/W
b2	M34STP	TMRA_3 Count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	R/W
b1	M33STP	TMRA_2 Count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	R/W
b0	M32STP	TMRA_1 Count pause signal	0: Counter still counts even if the kernel stops 1: Counter pause counting when kernel stops	R/W

42.6 SW Debug Port

42.6.1 Introduction to SW Agreement

This synchronous serial protocol uses two pins:

- SWCLK: clock from host 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).

42.7 TPIU (Trace Port Interface Unit)

42.7.1 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).

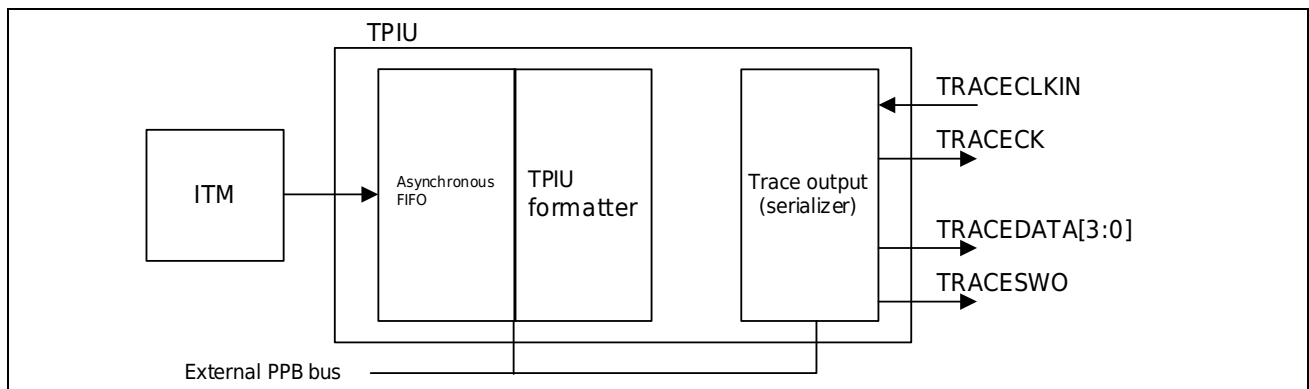


Figure 42-4 TTPIU Block Diagram

42.7.2 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).

Table 42-4 Asynchronous Mode Pin

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.

Table 42-5 Sync Mode Pin

TPIU Pin Name	Track Sync Mode	
	Type	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 host 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 host must program bits TRACE_IOEN and TRACE_MODE [1:0] of the MCU Debugging Configuration Registers (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.

Table 42-6 Pin Disable

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
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.

42.7.3 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.

42.7.4 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).

42.7.5 TPIU Configuration Instance

- Set bit TRCENA in the Debug Exception and Monitor Control Register (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. Instance: 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	Revision Content
Rev1.00	2025/02/18	First edition release.
Rev1.01	2025/12/18	1) Clock Controller (CMU): 4.11.24 CMU AD/DA Clock Configuration Register (CMU_PERICKSEL), added note: "The TRNG clock source is not controlled by this bit. When the TRNG operates, this bit must be set to 0b0000." 2) Power Control (PWC): 5.7 Modified PWC_FPRC register bit width to 16. 3) General IO (GPIO): 9.4.11 Modified note, added: "After reset, the PC14 and PC15 pins are configured as external low-speed oscillator interfaces. In their initial state, digital functions are disabled. To enable digital functions, it is necessary to first set the register CMU_XTAL32CR.XTAL32STP to 1 to disable the external low-speed oscillator, and then write 0 to the DDIS bit of the corresponding PCRxy register to activate the digital functionality." 4) DMA Controller (DMA): 15.6 Modified note to "Prohibiting DMA access to memory regions that may cause bus errors (such as reserved address spaces, protected address spaces, etc.)." 5) Analog-to-Digital Converter (ADC): 17.2 Table 17-1 Corrected ADC channels for unit 2 (ADC2). 6) Integrated Circuit Bus (I2C): 29.5.4 Added new function field SDADLY[3:0] SDA output delay counter to register I2C_CR4.