



HC32F460_F45x_A460 Series

32-bit ARM® Cortex®-M4

microcontroller

**reference
manual**

Rev1.6 September 2023

declaration Ming Dynasty (1368-1644)

- ★ Xiaohua Semiconductor Corporation ("XHSC") reserves the right to make changes, corrections, enhancements, or modifications to XHSC products and/or this document at any time without notice. Users are encouraged to obtain the most current information prior to placing an order. XHSC products are sold under the terms and conditions of sale set forth in the basic contract of purchase and sale.
- ★ Customer shall select the appropriate XHSC product for your application and 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 confirms that no license to any intellectual property is granted, either expressly or impliedly.
- ★ Resale of XHSC products under terms different from those set forth herein shall void any warranty commitment by XHSC with respect to such products.
- ★ Any graphics or lettering with the "®" or "™" logo are trademarks of XHSC. All other product or service names displayed on XHSC products are the property of their respective owners.
- ★ The information in this notice supersedes and replaces the information in previous versions.

©2023 Siu Wah Semiconductor Limited All Rights Reserved

table of contents**table of contents**

declaration	Ming 2
table of contents.....	Table of Contents 3
Table Index.....	32
Figure Index	36
Introduction (Overview).....	44
1 Memory Mapping	45
1.1 Memory Mapping.....	45
1.2 External Space Mapping	49
1.3 Bit Space	49
1.4 Address Remapping	50
1.5 Remap Registers	51
1.5.1 Access Protection Register (MMF_REMPRT).....	51
1.5.2 Remap control register (MMF_REMCRx x=0, 1)	52
2 Bus Architecture (BUS).....	53
2.1 Overview.....	53
2.2 Bus Architecture	54
2.3 Bus Functions.....	55
3 Reset Control (RMU)	56
3.1 Introduction.....	56
3.2 Reset Mode and Reset Flag Bits	57
3.3 Reset Timing.....	59
3.3.1 Power-on reset	59
3.3.2 NRST Pin Reset	59
3.3.3 Undervoltage Reset	60
3.3.4 Programmable Voltage Detection 1 Reset, Programmable Voltage Detection 2 Reset	60
..... 1 Reset, Programmable Voltage Detection	
3.3.5 Watchdog Reset, Specialized Watchdog Reset.....	63
3.3.6 Power-down Wake-up Reset.....	63
3.3.7 Software Reset	64
3.3.8 MPU Error Reset.....	64
3.3.9 RAM Parity Reset.....	64
3.3.10 RAMECC reset	65
3.3.11 Clock frequency abnormal reset	65

3.3.12	External High Speed Oscillator Abnormal Stop Reset.....	66
3.3.13	Judgment of reset mode	66
3.3.14	Reset conditions for each module	67
3.4	Register Description	68
3.4.1	Reset flag register 0 (RMU_RSTF0)	68
4	Clock Controller (CMU).....	70
4.1	Introduction.....	70
4.2	System Block Diagram.....	71
4.2.1	System Block Diagram	71
4.2.2	Clock Frequency Measurement Block Diagram	72
4.3	Clock Source Specifications.....	73
4.4	Operating Clock Specifications.....	75
4.5	Crystal Circuit	77
4.5.1	External High Speed Oscillator	77
4.5.2	External High Speed Oscillator Fault Detection	79
4.5.3	External Low Speed Oscillator	81
4.6	Internal RC Clock.....	82
4.6.1	HRC Clock.....	82
4.6.2	MRC Clock.....	82
4.6.3	LRC Clock	82
4.6.4	SWDTRC Clock	83
4.7	PLL Clock	83
4.8	Clock switching procedure.....	83
4.8.1	Clock Source Switching.....	84
4.8.2	Clock division switching	85
4.9	Clock output function	86
4.10	Clock Frequency Measurement	87
4.10.1	Clock Frequency Measurement.....	87
4.10.2	Digital Filter Function.....	88
4.10.3	Interrupt/reset function	88
4.11	Register Description	89
4.11.1	CMU XTAL Configuration Register (CMU_XTALCFGR)	91
4.11.2	CMU XTAL validation configuration register (CMU_XTALSTBCR)	92
4.11.3	CMU XTAL Control Register (CMU_XTALCR)	92
4.11.4	CMU XTAL Oscillation Fault Control Register (CMU_XTALSTDRCR)	93

4.11.5	CMU XTAL Oscillation Fault Status Register (CMU_XTALSTDSR)	94
4.11.6	CMU XTAL32 Configuration Register (CMU_XTAL32CFG)	94
4.11.7	CMU XTAL32 Filter Register (CMU_XTAL32NFR)	94
4.11.8	CMU XTAL32 Control Register (CMU_XTAL32CR)	95
4.11.9	CMU HRC Calibration Register (CMU_HRCTRM)	95
4.11.10	CMU HRC Control Register (CMU_HRCCR)	96
4.11.11	CMU MRC Calibration Register (CMU_MRCTRM)	96
4.11.12	CMU MRC Control Register (CMU_MRCCR)	97
4.11.13	CMU LRC Calibration Register (CMU_LRCTRM)	97
4.11.14	CMU LRC Control Register (CMU_LRCCR)	98
4.11.15	CMU MPLL Configuration Register (CMU_PLLCFG)	99
4.11.16	CMU MPLL Control Register (CMU_PLLCR)	101
4.11.17	CMU UPLL Configuration Register (CMU_UPLLCFG)	102
4.11.18	CMU UPLL Control Register (CMU_UPLLCR)	104
4.11.19	CMU Clock Source Stabilizer (CMU_OSCSTBSR)	105
4.11.20	CMU System Clock Source Switching Register (CMU_CKSWR)	105
4.11.21	CMU Clock Division Configuration Register (CMU_SCFGR)	106
4.11.22	CMU USBFS Clock Configuration Memory (CMU_UFSCKCFG)	108
4.11.23	CMU AD/TRNG Clock Configuration Memory (CMU_PERICKSEL)	109
4.11.24	CMU I2S Clock Configuration Store (CMU_I2SCKSEL)	110
4.11.25	CMU Debug Clock Configuration Register (CMU_TPIUCKCFG)	111
4.11.26	CMU MCO1 Configuration Memory (CMU_MCO1CFG)	111
4.11.27	CMU MCO2 Configuration Memory (CMU_MCO2CFG)	112
4.11.28	FCM Lower Limit Comparison Value Register (FCM_LVR)	113
4.11.29	FCM upper limit comparison value register (FCM_UVR)	113
4.11.30	FCM Counter Value Register (FCM_CNTR)	113
4.11.31	FCM Start Stop Register (FCM_STR)	114
4.11.32	FCM Measurement Object Control Register (FCM_MCCR)	114
4.11.33	FCM Measurement Reference Control Register (FCM_RCCR)	115
4.11.34	FCM Interrupt Reset Control Register (FCM_RIER)	116
4.11.35	FCM Flag Register (FCM_SR)	116
4.11.36	FCM Flag Bit Clear Register (FCM_CLR)	117
5	Power Control (PWC)	118
5.1	Introduction	118
5.2	Power Distribution	118

5.3	Description of supply voltage detection unit (PVD)	120
5.3.1	Power-on reset/power-off reset action description	120
5.3.2	Undervoltage Reset (BOR) Description.....	121
5.3.3	Programmable Voltage Detection 1 (PVD1), Programmable Voltage Detection 2 (PVD2)	122
5.3.4	PVD1, PVD2 Interrupt/Reset Block Diagrams.....	123
5.3.5	Input/Output Pins	123
5.3.6	PVD1 Interrupt and Reset	124
5.3.7	PVD2 Interrupt and Reset	125
5.3.8	Internal voltage sampling and detection function.....	126
5.4	Action Mode and Low Power Mode	127
5.4.1	Mode of operation.....	130
5.4.2	Sleep Mode.....	132
5.4.3	Stop Mode	132
5.4.4	Power-down Mode.....	134
5.5	Ways to Reduce Power Consumption.....	138
5.5.1	Reducing System Clock Speed	138
5.5.2	Turning off unused clock sources	138
5.5.3	Function clock stop.....	138
5.5.4	Turning off u n u s e d RAM.....	139
5.6	Register Protection Function.....	140
5.7	Register Description	141
5.7.1	Power Mode Control Register 0 (PWC_PWRC0)	143
5.7.2	Power Mode Control Register 1 (PWC_PWRC1)	144
5.7.3	Power Mode Control Register 2 (PWC_PWRC2)	144
5.7.4	Power Mode Control Register 3 (PWC_PWRC3)	145
5.7.5	Power-down Wake-up Enable Register 0 (PWC_PDWKE0)	145
5.7.6	Power-down wake-up enable register 1 (PWC_PDWKE1)	146
5.7.7	Power-down wake-up enable register 2 (PWC_PDWKE2)	146
5.7.8	Power-down wake-up event edge selection register (PWC_PDWKES)	147
5.7.9	Power-down wake-up flag register 0 (PWC_PDWKF0)	148
5.7.10	Power-down wake-up flag register 1 (PWC_PDWKF1)	149
5.7.11	Power Monitoring Control Register (PWC_PWCMR)	149
5.7.12	Mode Switching Control Register (PWC_MDSWCR)	150
5.7.13	Function Clock C o n t r o l 0 (PWC_FCG0)	151

5.7.14	Function Clock Control 1 (PWC_FCG1)	153
5.7.15	Function Clock Control 2 (PWC_FCG2)	155
5.7.16	Function Clock Control 3 (PWC_FCG3)	157
5.7.17	PWC_FCG0 Protection Control (PWC_FCG0PC).....	159
5.7.18	Function Protection Control Register (PWC_FPRC).....	159
5.7.19	STOP Mode Control Register (PWC_STPMCR)	160
5.7.20	RAM Power Control Register 0 (PWC_RAMPC0).....	161
5.7.21	RAM Running Condition Register (PWC_RAMOPM)	162
5.7.22	XTAL32 Control with Current Source (PWC_XTAL32CS)	162
5.7.23	Wake-up Timer Control Register (PWC_WKTCR)	163
5.7.24	PVD Control Register 0 (PWC_PVDCR0)	163
5.7.25	PVD Control Register 1 (PWC_PVDCR1)	164
5.7.26	PVD Filter Control Register (PWC_PVDFCR)	165
5.7.27	PVD Level Control Register (PWC_PVDLCR)	166
5.7.28	PVD Interrupt Control Register (PWC_PVDICR)	167
5.7.29	PVD Detection Status Register (PWC_PVDDSR).....	168
6	Initialization Configuration (ICG)	169
6.1	Introduction.....	169
6.2	Register Description	170
6.2.1	Initialize Configuration Register 0 (ICG0)	170
6.2.2	Initialize Configuration Register 1 (ICG1).....	172
6.2.3	Initialize configuration register n (ICGn) n=2~3	173
6.2.4	Initialization Configuration Register 4 (ICG4)	173
6.2.5	Initialization Configuration Register 5 (ICG5)	173
6.2.6	Initialize configuration register n (ICGn) n=6~7	174
7	Embedded FLASH (EFM)	175
7.1	Introduction.....	175
7.2	Main Characteristics.....	175
7.3	Embedded FLASH	176
7.4	Read Interface	178
7.4.1	Relationship between CPU Clock and FLASH Read Time.....	178
7.4.2	FLASH Low Power Read.....	178
7.5	FLASH Read Acceleration Cache.....	181
7.6	FLASH Programming and Erase Operations.....	182
7.6.1	Single Programming No Readback Mode	182

7.6.2	Single Programmable Readback Mode	182
7.6.3	Continuous Programming Operation	183
7.6.4	Erase Operation.....	183
7.6.5	Data Security Protection	184
7.6.6	Bus Hold Function	184
7.6.7	FLASH Erase, Programming Window Protection.....	185
7.6.8	Interruptions.....	185
7.7	One-time Programmable Bytes	186
7.8	Bootstrapping exchange.....	187
7.9	Register Description	188
7.9.1	Access Protection Register (EFM_FAPRT)	189
7.9.2	FLASH Stop Register (EFM_FSTP).....	189
7.9.3	Read Mode Register (EFM_FRMC)	190
7.9.4	Erase Write Mode Register (EFM_FWMC).....	191
7.9.5	Status Register (EFM_FSR)	192
7.9.6	Status Clear Register (EFM_FSCLR)	193
7.9.7	Interrupt license register (EFM_FITE)	194
7.9.8	Boot Switching Status Register (EFM_FSWP).....	194
7.9.9	FLASH Window Protection Starting Address Register (EFM_FPMTSW)	195
7.9.10	FLASH Window Protection End Address Register (EFM_FPMTEW)	195
7.9.11	UNIQUE ID register (EFM_UQID0)	195
7.9.12	UNIQUE ID Register (EFM_UQID1)	196
7.9.13	UNIQUE ID Register (EFM_UQID2)	196
7.10	Cautions.....	197
8	Internal SRAM (SRAM)	198
8.1	Introduction.....	198
8.2	Register Description	200
8.2.1	SRAM Waiting Control Register (SRAM_WTCR).....	201
8.2.2	SRAM Wait Protection Register (SRAM_WTPR).....	203
8.2.3	SRAM Checksum Control Register (SRAM_CKCR)	204
8.2.4	SRAM Checksum Protection Register (SRAM_CKPR)	205
8.2.5	SRAM Checksum Status Register (SRAM_CKSR).....	205
9	General Purpose IO (GPIO)	206
9.1	Introduction.....	206
9.2	Port Function Summary.....	207

9.3	Action Description	208
9.3.1	General Purpose Input / Output GPIO Functions.....	208
9.3.2	Peripheral Functions	209
9.3.3	Dual Peripheral Function.....	209
9.3.4	Event Port Input/Output Function.....	209
9.3.5	External Interrupt EIRQ Input Function	210
9.3.6	Analog Functions.....	210
9.3.7	General Purpose Controls	210
9.4	Register Description	211
9.4.1	General Purpose Input Register (PIDRx)	212
9.4.2	General Purpose Output Data Register (PODRx)	212
9.4.3	General Purpose Output License Register (POERx).....	212
9.4.4	General Purpose Output Setting Register (POSRx)	213
9.4.5	General Purpose Output Reset Register (PORRx).....	213
9.4.6	General Purpose Output Flip-Flop Register (POTRx).....	213
9.4.7	Special Control Register (PSPCR)	214
9.4.8	Public Control Register (PCCR)	214
9.4.9	Input Control Register (PINAER)	215
9.4.10	Write Protect Register (PWPR)	215
9.4.11	General Purpose Control Register (PCRxy)	216
9.4.12	Function Selection Register (PFSRxy).....	218
9.4.13	Event Port Direction Selection Register (PEVNTDIRRm)	219
9.4.14	Event Port Input Data Register (PEVNTIDRm)	219
9.4.15	Event Port Output Data Register (PEVNTODRm).....	220
9.4.16	Event Port Output Data Reset Register (PEVNTORRm)	220
9.4.17	Event Port Output Data Setting Register (PEVNTOSRm).....	221
9.4.18	Event Port Rising edge input license register (PEVNTRISRm).....	221
9.4.19	Event Port Falling Edge Input License Register (PEVNTFALRm).....	222
9.4.20	Event Port Input Filter Control Register (PEVNTNFCR)	223
9.4.21	32bit Access.....	224
9.5	Cautions.....	224
10	Interrupt Controller (INTC)	225
10.1	Introduction.....	225
10.2	INTC System Block Diagram	227
10.2.1	System Block Diagram	227

10.3 Vector Table.....	228
10.3.1 Interrupt Vector Table.....	228
10.3.2 Interrupt event request number.....	233
10.4 Function Description	247
10.4.1 Non-maskable interrupts	247
10.4.2 External pin interrupt event request	248
10.4.3 Interrupt Source Selection	248
10.4.4 Software Interrupt.....	248
10.4.5 Interrupt/Event Selection.....	249
10.4.6 WFE Wake Up Event Management	249
10.4.7 Digital Filters.....	250
10.4.8 Low Power Mode Return	251
10.4.9 Internal Trigger Events.....	251
10.5 Register Description	252
10.5.1 NMI Pin Non-Maskable Interrupt Control Register (INT_NMICR).....	257
10.5.2 Non-maskable interrupt enable register (INT_NMIENR)	258
10.5.3 Non-maskable interrupt flag register (INT_NMIFR).....	259
10.5.4 Non-maskable interrupt flag clear register (INT_NMICFR)	260
10.5.5 External Pin Interrupt Control Register (INT_EIRQCRx) (x=0~15)	261
10.5.6 External Pin Interrupt Flag Register (INT_EIFR)	262
10.5.7 External Pin Interrupt Flag Clear Register (EIFCR).....	262
10.5.8 Interrupt/Event Select Register (INT_SEL0~31)	263
10.5.9 Interrupt Select Register (INT_SEL32~127).....	264
10.5.10 Vector Shared Interrupt Select Register (INT_VSEL128~143)	265
10.5.11 Stop Mode Wake-Up Event Enable Register (INT_WUPEN).....	266
10.5.12 Software Interrupt/Event Register (INT_SWIER)	267
10.5.13 Event Enable Register (INT_EVTER)	267
10.5.14 Interrupt Enable Register (INT_IER).....	268
10.6 Precautions for Use.....	269
11 Automatic Operating System (AOS).....	270
11.1 Introduction.....	270
11.1.1 Function Overview	270
11.1.2 Module Schematic.....	271
11.2 Functional Description	272
11.2.1 AOS Source Event List	272

11.2.2	AOS Target List.....	272
11.3	Action Description	273
11.3.1	Dedicated Trigger Sources	273
11.3.2	Common Trigger Sources.....	273
11.4	Register Description	274
11.4.1	Peripheral Trigger Event Register (INTSFTTRG)	275
11.4.2	DCU Trigger Source Selection Register (DCU_TRGSELx) (x=1~4)	275
11.4.3	DMA1 Transmit Start Trigger Source Selection Register (DMA1_TRGSELx) (x=0~3)	276
11.4.4	DMA2 Transmit Start Trigger Source Selection Register (DMA2_TRGSELx) (x=0~3)	276
11.4.5	DMA Channel Reset Trigger Source Selection Register (DMA_RC_TRGSEL)	277
11.4.6	Timer6 Hardware Trigger Event Select Register (TMR6_TRGSELx) (x=0~1) ...	278
11.4.7	Timer0 Hardware Trigger Event Select Register (TMR0_TRGSEL).....	279
11.4.8	Event Port Trigger Source Selection Register (PEVNT_TRGSEL12, PEVNT_TRGSEL34)	
		280
11.4.9	TimerA Internal Trigger Event Selection Register 0 (TMRA_TRGSEL0)	281
11.4.10	TimerA Internal Trigger Event Selection Register 1 (TMRA_TRGSEL1)	282
11.4.11	OTS Trigger Source Selection Register (OTS_TRGSEL)	282
11.4.12	A/D1 conversion start on-chip trigger source selection register ADC1_TRGSELx(x=0,1)	283
11.4.13	A/D2 conversion start on-chip trigger source selection register ADC2_TRGSELx(x=0,1)	283
11.4.14	Common Trigger Source Selection Register 1 (AOS_COMTRG1).....	284
11.4.15	Common Trigger Source Selection Register 2 (AOS_COMTRG2).....	284
12	Keyboard Scanning Control Module (KEYSCAN)	285
12.1	Introduction.....	285
12.2	KEYSCAN System Block Diagram	285
12.3	Pin Description	286
12.4	Functional Description	286
12.4.1	Key Recognition Function	286
12.4.2	Keyboard Scanning Function	286
12.4.3	Precautions for use	287
12.5	Register Description	288
12.5.1	KEYSCAN Scan Control Register (KEYSCAN_SCR).....	289
12.5.2	KEYSCAN Scan Enable Register (KEYSCAN_SER)	291

12.5.3 KEYSCAN Scan Status Register (KEYSCAN_SSR).....	291
13 Memory Protection Unit (MPU).....	292
13.1 Introduction.....	292
13.2 Functional Description	293
13.2.1 Regional Scope Setting	293
13.2.2 Permission Settings.....	293
13.2.3 MPU Action Selection	293
13.2.4 Starting the MPU	293
13.3 Application Examples.....	294
13.3.1 Allow access to only a portion of the space.....	294
13.3.2 Blocking access to only part of the space.....	294
13.4 Register Description	295
13.4.1 Region Range Description Register MPU_RGD _n (n=0 to 15)	296
13.4.2 Region control register MPU_RGCR _n (n=0 to 15).....	297
13.4.3 Control Register MPU_CR	298
13.4.4 Status flag register MPU_SR	300
13.4.5 Flag Clear Register MPU_ECLR.....	300
13.4.6 Write-Protect Register MPU_WP	301
13.4.7 IP Access Protection Register MPU_IPPR	302
14 DMA Controller (DMA).....	304
14.1 Introduction.....	304
14.2 Module Schematic.....	305
14.3 Functional Description	306
14.3.1 Enable DMA Controller	306
14.3.2 Channel Selection and Channel Prioritization	306
14.3.3 Initiating DMA.....	306
14.3.4 Data Blocks	306
14.3.5 Transmit Address Control.....	306
14.3.6 Number of transmissions	307
14.3.7 Interrupt and event signal outputs.....	307
14.3.8 Chain transmission	307
14.3.9 Discontinuous Address Transmission.....	310
14.3.10 Channel Reset.....	311
14.3.11 Early termination of transmission	313
14.4 Application Examples.....	314

14.4.1	Memory-to-memory transfers	314
14.4.2	Memory to Peripheral Circuit Transfer.....	315
14.4.3	Memory-to-memory chain transfer	317
14.5	Register Description	319
14.5.1	DMA Enable Register (DMA_EN)	320
14.5.2	Interrupt status register 0 (DMA_INTSTAT0).....	320
14.5.3	Interrupt Status Register 0 (DMA_INTSTAT1)	321
14.5.4	Interrupt Mask Register (DMA_INTMASK0)	321
14.5.5	Interrupt Mask Register (DMA_INTMASK1)	322
14.5.6	Interrupt Reset Register (DMA_INTCLR0)	322
14.5.7	Interrupt reset register (DMA_INTCLR1)	323
14.5.8	Channel Enable Register (DMA_CHEN)	323
14.5.9	Channel reset control register (DMA_RCFGCTL)	324
14.5.10	Transmission Request Status Register (DMA_REQSTAT)	326
14.5.11	Channel Status Observation Register (DMA_CHSTAT)	327
14.5.12	Transmit Source Address Register (DMA_SARx) (x=0~3)	327
14.5.13	Transmit Destination Address Register (DMA_DARx) (x=0~3)	328
14.5.14	Data Control Register (DMA_DTCTLx) (x=0~3)	328
14.5.15	Repeat Area Size Register (DMA_RPTx) (x=0~3)	329
14.5.16	Repeat Region Size Register B(DMA_RPTBx) (x=0~3)	330
14.5.17	Source Device Discontinuous Address Transfer Control Register (DMA_SNSEQCTLx) (x=0~3).....	331
14.5.18	Source Device Discontinuous Address Transfer Control Register B (DMA_SNSEQCTLBx) (x=0~3).....	332
14.5.19	Target Device Discontinuous Address Transfer Control Register (DMA_DNSEQCTLx) (x=0~3)	333
14.5.20	Target Device Discontinuous Address Transfer Control Register B(DMA_DNSEQCTLBx) (x=0~3).....	334
14.5.21	Chain Pointer Register (DMA_LLPx) (x=0~3)	335
14.5.22	Channel Control Register (DMA_CHxCTL) (x=0~3)	336
14.5.23	Channel Monitor Registers (DMA_MONSARx, DMA_MONDARx, DMA_MONDTCTLx, DMA_MONRPTx, DMA_MONSSEQCTLx, DMA_MONDNSEQCTLx) (x=0~3)	337
14.6	Precautions for Use	337
15	Voltage Comparators (CMP)	339

15.1	Introduction.....	339
15.2	Functional Description.....	341
15.2.1	Voltage Comparison.....	341
15.2.2	Digital Filtering	342
15.2.3	Interrupts and Events.....	342
15.2.4	Scanning Comparison Mode.....	342
15.2.5	8bit-DAC setting	344
15.3	Cautions	344
15.3.1	Module Stop Function.....	344
15.3.2	Action when the module is stopped.....	344
15.3.3	Stopping the action in low-power mode	344
15.3.4	Action in power-down low-power mode	344
15.4	Register Description	345
15.4.1	CMP Control Register (CMP_CTRL)	346
15.4.2	CMP Voltage Select Register (CMP_VLTSEL).....	348
15.4.3	CMP Result Monitor Register (CMP_OUTMON)	350
15.4.4	CMP Stabilization Time Register (CMP_CVSSTB).....	351
15.4.5	CMP Compare Voltage Scan Period Register (CMP_CVSPRD).....	351
15.4.6	8bit-DAC Control Register for CMP (CMP_DACR).....	CMP_DACR
15.4.7	8bit-DAC Data Register for CMP (CMP_DADR1,CMP_DADR2) CMP_DADR1,CMP_DADR2	
15.4.8	CMP Internal Reference Voltage AD Conversion Register (CMP_RVADC) ...	353
16	Analog-to-digital converter modules (ADCs).....	354
16.1	Introduction.....	354
16.2	ADC System Block Diagram	355
16.3	Function Description	357
16.3.1	ADC Clock	357
16.3.2	Channel Selection	357
16.3.3	Trigger Source Selection	358
16.3.4	S e q u e n c e A Single Scan Mode.....	359
16.3.5	S e q u e n c e A Continuous Scan Mode	360
16.3.6	Dual Sequence Scanning Mode	361
16.3.7	Analog Watchdog Function.....	363
16.3.8	Sampling time and conversion time for analog inputs	364
16.3.9	A/D Data Register Auto Clear Function.....	365

16.3.10	Converting Data Averaging Calculation Function	365
16.3.11	Programmable Gain Amplifier PGA.....	366
16.3.12	Multi-ADC Co-Operation Mode	366
16.3.13	Interrupt and Event Signal Outputs.....	371
16.4	Register Description	372
16.4.1	A/D startup register ADC_STR	374
16.4.2	A/D Control Register 0 ADC_CR0	375
16.4.3	A/D Control Register 1 ADC_CR1	376
16.4.4	A/D Conversion Start Trigger Register ADC_TRGSR	377
16.4.5	A/D Channel Select Register A ADC_CHSELRA0	378
16.4.6	A/D Channel Select Register A 1 ADC_CHSELRA1	378
16.4.7	A/D Channel Select Register B ADC_CHSELRB0.....	379
16.4.8	A/D Channel Select Register B1 ADC_CHSELRB1.....	379
16.4.9	A/D Average Channel Selection Register ADC_AVCHSELR0	380
16.4.10	A/D Average Channel Selection Register 1 ADC_AVCHSELR1	380
16.4.11	A/D Sampling Status Register ADC_SSTR.....	381
16.4.12	A/D Channel Mapping Control Register ADC_CHMUXR.....	382
16.4.13	A/D Interrupt Status Register ADC_ISR.....	383
16.4.14	A/D Interrupt Permission Register ADC_ICR.....	383
16.4.15	A/D Synergy Mode Control Register ADC_SYNCCR	384
16.4.16	A/D Data Register ADC_DR	385
16.4.17	Analog Watchdog Control Register ADC_AWDCR	386
16.4.18	Analog Watchdog Threshold Register ADC_AWDDR0, ADC_AWDDR1.....	387
16.4.19	Analog Watchdog Compare Channel Select Register ADC_AWDCHSR0.....	388
16.4.20	Analog Watchdog Compare Channel Select Register 1 ADC_AWDCHSR1.....	388
16.4.21	Analog Watchdog Status Register ADC_AWDSR0.....	389
16.4.22	Analog Watchdog Status Register 1 ADC_AWDSR1	389
16.4.23	A/D Programmable Gain Amplifier Control Register ADC_PGACR	390
16.4.24	A/D Programmable Gain Multiplier Register ADC_PGAGSR	390
16.4.25	A/D Programmable Gain Amplifier Input Select Register ADC_PGAINSRO	391
16.4.26	A/D Programmable Gain Amplifier Input Select Register 1 ADC_PGAINSRI	391
16.5	Precautions for Use.....	392
16.5.1	Precautions to be taken when reading data registers	392
16.5.2	Scan Completion Interrupt Handling Considerations.....	392
16.5.3	Precautions for Module Stop and Low Power Setting	392

16.5.4	Pin Settings for A/D Conversion Analog Channel Inputs	392
16.5.5	Noise Control	392
17	Temperature Sensors (OTS)	393
17.1	Introduction.....	393
17.2	Instructions for use	394
17.3	Register Description	396
17.3.1	OTS Control Register (OTS_CTL)	396
17.3.2	OTS Data Register 1 (OTS_DR1)	397
17.3.3	OTS Data Register 2 (OTS_DR2)	397
17.3.4	OTS Error Compensation Register (OTS_ECR)	397
18	Advanced Control Timer (Timer6)	398
18.1	Introduction.....	398
18.2	Basic Block Diagram	398
18.3	Functional Description	400
18.3.1	Basic Moves	400
18.3.2	Clock Source Selection.....	403
18.3.3	Counting direction	403
18.3.4	Digital Filtering.....	404
18.3.5	Software synchronization	404
18.3.6	Hardware synchronization.....	406
18.3.7	Pulse Width Measurement.....	408
18.3.8	Periodic Measurement.....	408
18.3.9	Cache Functions.....	408
18.3.10	General Purpose PWM Output.....	415
18.3.11	Orthogonal code counting.....	420
18.3.12	Cycle interval response	425
18.3.13	EMB Control	426
18.3.14	Typical Application Examples.....	426
18.3.15	Functional summary table	433
18.4	Interrupt and Event Descriptions	435
18.4.1	Interrupt Output	435
18.4.2	Event Output.....	435
18.5	Register Description	437
18.5.1	General purpose count value register (TMR6_CNTER)	439
18.5.2	General Purpose Periodic Reference Value Register (TMR6_PERAR-PERCR)..	439

18.5.3	Generalized Comparison Reference Value Register (TMR6_GCMAR-GCMFR) .	439
18.5.4	Dedicated Comparison Reference Value Register (TMR6_SCMAR-SCMFR)	440
18.5.5	Dead Time Reference Value Register (TMR6_DTU<D>AR)	440
18.5.6	General Purpose Control Register (TMR6_GCONR)	441
18.5.7	Interrupt Control Register (TMR6_ICONR)	442
18.5.8	Port Control Register (TMR6_PCONR).....	444
18.5.9	Cache Control Register (TMR6_BCONR).....	446
18.5.10	Deadband Control Register (TMR6_DCONR)	448
18.5.11	Filter Control Register (TMR6_FCONR)	449
18.5.12	Valid cycle register (TMR6_VPERR)	450
18.5.13	Status Flag Register (TMR6_STFLR)	451
18.5.14	Hardware Startup Event Selection Register (TMR6_HSTAR)	452
18.5.15	Hardware Stop Event Selection Register (TMR6_HSTPR).....	453
18.5.16	Hardware clear event selection register (TMR6_HCLRR).....	454
18.5.17	Hardware Capture Event Select Register (TMR6_HCPAR)	455
18.5.18	Hardware Capture Event Select Register (TMR6_HCPBR)	456
18.5.19	Hardware-recursive event selection register (TMR6_HCUPR)	457
18.5.20	Hardware Decrement Event Select Register (TMR6_HCDOR)	459
18.5.21	Software Synchronized Startup Control Register (TMR6_SSTAR)	461
18.5.22	Software Synchronization Stop Control Register (TMR6_SSPPR)	461
18.5.23	Software Synchronized Clear Control Register (TMR6_SCLR)	462
19	General Purpose Control Timer (Timer4)	463
19.1	Introduction.....	463
19.2	Basic Block Diagram	463
19.3	Functional Description	465
19.3.1	Basic Moves	465
19.3.2	Cache Functions.....	468
19.3.3	General Purpose PWM Output	474
19.3.4	Cycle interval response	480
19.3.5	EMB Control	482
19.4	Interrupt and Event Descriptions	482
19.4.1	Count Compare Match Interrupt.....	482
19.4.2	Count Cycle Matching Interrupt.....	482
19.4.3	Overloading the Count Match Interrupt.....	482
19.4.4	Dedicated Compare Match Event.....	483

19.5 Register Description	484
19.5.1 Count Value Register (TMR4_CNTR).....	486
19.5.2 Cyclic Reference Register (TMR4_CPSR)	486
19.5.3 Control Status Register (TMR4_CCSR)	487
19.5.4 Valid Cycle Register (TMR4_CVPR)	489
19.5.5 General Purpose Comparison Reference Register (TMR4_OCCRm).....	489
19.5.6 General Purpose Control Status Register (TMR4_OCSRn)	490
19.5.7 General Purpose Extended Control Register (TMR4_OCERn)	491
19.5.8 General Purpose Mode Control Register (TMR4_OCMRm)	493
19.5.9 Specialized Comparison Reference Register (TMR4_SCCRm)	498
19.5.10 Dedicated control status register (TMR4_SCSRm)	499
19.5.11 Specialized Mode Control Register (TMR4_SCMRm)	501
19.5.12 PWM Basic Control Register (TMR4_POCRn).....	502
19.5.13 PWM Filter Control Register (TMR4_PFSRn).....	503
19.5.14 PWM Deadband Control Register (TMR4_PDARn)	503
19.5.15 Reload Control Status Register (TMR4_RCSR)	504
19.5.16 EMB Control Status Register (TMR4_ECSR).....	505
19.5.17 EMB Extended Control Register (TMR4_ECER)	505
20 Emergency Brake Module (EMB)	506
20.1 Introduction.....	506
20.2 Functional Description	507
20.2.1 OVERVIEW	507
20.2.2 Stop PWM signal output when external port input level changes.	507
20.2.3 PWM signal output stops when the PWM output port level is in phase (same high or same low).	508
20.2.4 Stop PWM signal output based on voltage comparator comparison result.....	509
20.2.5 Stop PWM signal output when the external oscillator stops oscillating.	509
20.2.6 Write Register Software Control PWM Signal Outputs.....	509
20.3 Register Description	510
20.3.1 EMB Control Register 0 (EMB_CTL0).....	511
20.3.2 EMB Control Register 1~3 (EMB_CTL1~3)	512
20.3.3 EMB Feedback Level Selection Register 0 (EMB_PWMLV0).....	513
20.3.4 EMB Feedback Level Selection Register 1~3 (EMB_PWMLV1~3)	514
20.3.5 EMB Software Output Enable Control Register (EMB_SOEx) (x=0~3).....	514
20.3.6 EMB Status Register (EMB_STATx) (x=0~3)	515

20.3.7	EMB Status Reset Register (EMB_STATCLR x) ($x=0\sim 3$)	516
20.3.8	EMB interrupt license register (EMB_INTEN x) ($x=0\sim 3$)	517
21	General purpose timer (TimerA)	518
21.1	Introduction.....	518
21.2	Basic Block Diagram	518
21.3	Functional description	520
21.3.1	Basic movements	520
21.3.2	Clock Source Selection.....	523
21.3.3	Synchronized Start	524
21.3.4	Digital Filtering.....	525
21.3.5	Cache Functions.....	526
21.3.6	Cascade counting.....	527
21.3.7	PWM Output.....	528
21.3.8	Orthogonal code counting.....	529
21.4	Interrupt and Event Descriptions	534
21.4.1	Comparing Matching Interrupts and Events.....	534
21.4.2	Cycle Matching Interrupts and Events	534
21.5	Register Description	535
21.5.1	General purpose count value register (TMRA_CNTER)	537
21.5.2	Periodic Reference Value Register (TMRA_PERAR).....	537
21.5.3	Comparison Reference Value Register (TMRA_CMPAR1~8)	537
21.5.4	Control Status Register L (TMRA_BCSTRL).....	538
21.5.5	Control Status Register H (TMRA_BCSTRH).....	539
21.5.6	Interrupt Control Register (TMRA_ICONR)	540
21.5.7	Event Control Register (TMRA_ECONR)	541
21.5.8	Filter Control Register (TMRA_FCONR)	542
21.5.9	Status Flag Register (TMRA_STFLR).....	543
21.5.10	Cache Control Register (TMRA_BCONR1~4)	544
21.5.11	Capture Control Register (TMRA_CCONR1~8)	545
21.5.12	Port Control Register (TMRA_PCONR1~8)	546
21.5.13	Hardware Trigger Event Selection Register (TMRA_HCONR).....	548
21.5.14	Hardware-recursive event selection register (TMRA_HCUPR)	550
21.5.15	Hardware Decrement Event Select Register (TMRA_HCDOR)	552
22	General Purpose Timer (Timer0).....	554
22.1	Introduction.....	554

22.2 Basic Block Diagram	554
22.3 Function Description	555
22.3.1 Clock Source Selection.....	555
22.3.2 Basic Counting Action	556
22.3.3 Hardware Trigger Action	556
22.4 Interrupt and Event Descriptions	557
22.4.1 Interrupt output.....	557
22.4.2 Event Output.....	557
22.5 Register Description	558
22.5.1 Count value register (TMR0_CNTAR).....	558
22.5.2 Base value register (TMR0_CMPAR)	558
22.5.3 Basic Control Register (TMR0_BCONR)	559
22.5.4 Status Flag Register (TMR0_STFLR)	562
22.6 Precautions for use	562
23 Real Time Clock (RTC)	563
23.1 Introduction.....	563
23.2 Basic Block Diagram	564
23.3 Functional Description	565
23.3.1 Power On Setting	565
23.3.2 RTC count start setting	565
23.3.3 System Low Power Mode Switching	565
23.3.4 Read Count Register.....	565
23.3.5 Write Counter Register	566
23.3.6 Alarm Clock Settings	566
23.3.7 Clock Error Compensation.....	566
23.3.8 1Hz Output.....	566
23.4 Interrupt Description.....	568
23.4.1 Alarm Clock Interruption	568
23.4.2 Fixed-cycle interrupt	568
23.5 Register Description	569
23.5.1 Control Register 0 (RTC_CR0).....	570
23.5.2 Control Register 1 (RTC_CR1).....	571
23.5.3 Control Register 2 (RTC_CR2).....	572
23.5.4 Control Register 3 (RTC_CR3).....	573
23.5.5 Seconds Count Register (RTC_SEC)	573

23.5.6	Minute Count Register (RTC_MIN)	574
23.5.7	Time Counter Register (RTC_HOUR).....	575
23.5.8	Day Count Register (RTC_DAY)	577
23.5.9	Week Count Register (RTC_WEEK).....	578
23.5.10	Month Count Register (RTC_MON).....	579
23.5.11	Year Count Register (RTC_YEAR).....	579
23.5.12	Minute alarm clock register (RTC_ALMMIN)	580
23.5.13	Clock alarm register (RTC_ALMHOUR).....	580
23.5.14	Weekly alarm clock register (RTC_ALMWEEK).....	581
23.5.15	Clock error compensation registers (RTC_ERRCRH, RTC_ERRCRL)	582
24	Watchdog Counter (WDT/SWDT)	585
24.1	Introduction.....	585
24.2	Functional Description	586
24.2.1	Starting the Watchdog	586
24.2.2	Hardware boot method.....	586
24.2.3	Software startup method.....	587
24.2.4	Refresh Action.....	588
24.2.5	Flag Bits	588
24.2.6	Interrupt Reset.....	588
24.2.7	Counting underflow	589
24.2.8	Refresh Error.....	590
24.3	Register Description	591
24.3.1	Control Register (WDT_CR).....	592
24.3.2	Status registers (SWDT_SR, WDT_SR).....	593
24.3.3	Refresh registers (SWDT_RR, WDT_RR).....	593
24.4	Precautions for use	593
25	Universal Synchronous Asynchronous Transceiver (USART)	594
25.1	Introduction.....	594
25.2	USART System Block Diagram	595
25.3	Pin Description	595
25.4	Functional Description	596
25.4.1	UART.....	596
25.4.2	Multi-Processor Communications	605
25.4.3	Smart Card.....	609
25.4.4	Clock Synchronization Mode.....	613

25.4.5	Digital Filter Function.....	619
25.5	Register Description	620
25.5.1	Status Register (USART_SR)	621
25.5.2	Transmit Data Register (USART_TDR).....	624
25.5.3	Receive Data Register (USART_RDR).....	624
25.5.4	Baud Rate Register (USART_BRR).....	625
25.5.5	Control Register 1 (USART_CR1)	626
25.5.6	Control Register 2 (USART_CR2)	629
25.5.7	Control Register 3 (USART_CR3)	630
25.5.8	Prescaler register (USART_PR)	631
25.6	Precautions for use	632
25.6.1	UART Notes.....	632
25.6.2	Clock Synchronization Mode Notes.....	632
25.6.3	Other Notes	632
26	Integrated Circuit Bus (I2C)	633
26.1	Introduction.....	633
26.2	I2C System Block Diagram.....	634
26.2.1	System Block Diagram	634
26.2.2	Structural diagrams	635
26.3	Action Description	636
26.3.1	I2C Protocol	636
26.3.2	Address Matching.....	645
26.3.3	SMBus Action.....	651
26.3.4	reset (a dislocated joint)	652
26.3.5	Interrupt and Event Signal Outputs.....	653
26.3.6	Programmable Digital Filtering.....	654
26.4	Application Software Setting I2C Initialization Flow.....	654
26.5	Register Description	655
26.5.1	I2C Control Register 1 (I2C_CR1)	656
26.5.2	I2C Control Register 1 (I2C_CR2)	658
26.5.3	I2C Control Register 1 (I2C_CR3)	660
26.5.4	I2C Control Register 1 (I2C_CR4)	661
26.5.5	I2C Slave Address Register 0 (I2C_SLR0).....	662
26.5.6	I2C Slave Address Register 1 (I2C_SLR1).....	663
26.5.7	I2C SCL Level Timeout Control Register (I2C_SLTR)	663

26.5.8	I2C Status Register (I2C_SR).....	664
26.5.9	I2C Status Clear Register (I2C_CLR).....	668
26.5.10	I2C Data Transmit Register (I2C_DTR)	669
26.5.11	I2C Data Receive Register (I2C_DRR).....	669
26.5.12	I2C Data Shift Register (I2C_DSR).....	669
26.5.13	I2C Clock Control Register (I2C_CCR).....	670
26.5.14	I2C Filter Control Register (I2C_FLTR)	672
27	Serial Peripheral Interface (SPI)	673
27.1	Introduction.....	673
27.2	SPI System Block Diagram.....	674
27.3	Pinouts	674
27.4	SPI Motion System Description	675
27.4.1	Host Mode Pin Status.....	675
27.4.2	Slave Mode Pin Status.....	675
27.4.3	SPI System Connection Example.....	676
27.5	Data Communication Description.....	677
27.5.1	Baud rate	677
27.5.2	Data format.....	678
27.5.3	Transmission Format.....	679
27.5.4	Communication method.....	681
27.6	Running Instructions	683
27.6.1	Summary of Operating Modes	683
27.6.2	Host action during SPI run mode	684
27.6.3	Slave operation in SPI operation mode	684
27.6.4	Host Actions in Clock Synchronized Mode of Operation	686
27.6.5	Slave Maneuvering in Clock Synchronized Mode of Operation	687
27.6.6	Processing Flow of Several SPI Actions.....	688
27.7	Parity Bit Self-Diagnostics	689
27.8	Error Detection.....	690
27.8.1	Underload error	690
27.8.2	Mode Error.....	691
27.8.3	Overload Error	691
27.8.4	Parity error	693
27.9	Initialization of SPI.....	694
27.9.1	Clear SPE bit for initialization	694

27.9.2	System Reset Initialization	694
27.10	Interrupt Source	695
27.11	Available event trigger sources	695
27.12	Register Description	696
27.12.1	SPI Data Register (SPI_DR)	696
27.12.2	SPI Control Register (SPI_CR1)	697
27.12.3	SPI communication configuration register 1 (SPI_CFG1)	698
27.12.4	SPI Status Register (SPI_SR)	700
27.12.5	SPI Communication Configuration Register 2 (SPI_CFG2)	701
28	Four-wire Serial Peripheral Interface (QSPI)	703
28.1	Introduction.....	703
28.2	Memory Mapping	705
28.2.1	Internal bus space.....	705
28.2.2	ROM space and address width of the bus.....	706
28.3	QSPI Bus	707
28.3.1	SPI Protocol	707
28.3.2	SPI Mode	709
28.4	Timing Adjustment for QSPI Bus	710
28.4.1	QSPI Bus Reference Clock	710
28.4.2	SPI Bus Reference Clock	711
28.4.3	QSSN Signal Minimum High Level Width	711
28.4.4	QSSN Establishment Time.....	712
28.4.5	Holding time for QSSN	712
28.4.6	Serial data reception delay	713
28.5	Introduction to SPI Instructions f o r ROM Access	714
28.5.1	Existing QSPI-ROM Command Reference	714
28.5.2	Standard Read Command	715
28.5.3	Quick Read Command.....	715
28.5.4	2-Wire Output Fast Read Command	717
28.5.5	2-Wire Input/Output Quick Read Commands.....	718
28.5.6	4-Wire Output Fast Read Command	719
28.5.7	Four-wire inputs and outputs for fast read commands.....	720
28.5.8	Enter 4-Byte Mode Command.....	721
28.5.9	Exit 4-Byte Mode Command	722
28.5.10	Write License Instruction	722

28.6	QSPI Bus Cycle Arrangement.....	723
28.6.1	Single Flash Read for Independent Conversion.....	723
28.6.2	Flash Reading Using the Pre-Read Function	723
28.6.3	Termination of pre-reading	724
28.6.4	Pre-reading status monitoring.....	724
28.6.5	Flash Read Us ing Q SPI Bus Cycle Extension Feature	724
28.7	XIP Control.....	725
28.7.1	XIP Mode Settings	726
28.7.2	XIP Mode Exit	726
28.8	QSI02 a n d QSI03 Pin Status.....	727
28.9	Direct communication mode	728
28.9.1	About Direct Communication Mode	728
28.9.2	Direct communication mode setting.....	728
28.9.3	Q SPI Bus Cycle Generation in Direct Communication Mode.....	728
28.10	Interrupt	729
28.11	Precautions for use	729
28.11.1	Q SPI Register Setting Order	729
28.11.2	Module stop signal setting.....	729
28.12	Register Description	730
28.12.1	Q SPI Control Register (QSCR)	731
28.12.2	Q SPI Chip Select Control Register (QSCSCR)	734
28.12.3	Q SPI Format Control Register (QSFCR)	735
28.12.4	Q SPI Status Register (QSSR)	737
28.12.5	Q SPI Command Code Register (QSCCMD).....	739
28.12.6	Q SPI Direct Communication Command Register (QSDCOM)	739
28.12.7	Q SPI XIP Mode Code Register (QSXCMD).....	739
28.12.8	Q SPI System Configuration Register (QSSR2).....	740
28.12.9	Q SPI External Extended Address Register (QSEXAR).....	740
29	Integrated Circuit Built-in Audio Bus Module (I2S).....	741
29.1	Introduction.....	741
29.2	I2S System Block Diagram.....	742
29.3	Pin Description	742
29.4	Function Description	743
29.4.1	I2S General Description	743
29.4.2	Communication Methods.....	743

29.4.3	Supported Audio Protocols.....	744
29.4.4	Clock Generator	750
29.4.5	I2S Master Mode	752
29.4.6	I2S Slave Mode.....	753
29.4.7	I2S Interrupt	754
29.4.8	Precautions for use	754
29.5	Register Description	757
29.5.1	I2S Control Register (I2S_CTRL).....	758
29.5.2	I2S Status Register (I2S_SR).....	760
29.5.3	I2S Error Status Register (I2S_ER).....	761
29.5.4	I2S Configuration Register (I2S_CFGR)	762
29.5.5	I2S Transmit Buffer FIFO Data Register (I2S_TXBUF).....	763
29.5.6	I2S Receive Buffer FIFO Data Register (I2S_RXBUF)	763
29.5.7	I2S Frequency Divider Register (I2S_PR)	764
30	Controller Area Network (CAN).....	765
30.1	Introduction.....	765
30.2	CAN System Block Diagram	766
30.3	Pinouts	766
30.4	Function Description.....	767
30.4.1	Baud rate setting.....	767
30.4.2	Transmit Buffer.....	768
30.4.3	Receiver Buffer.....	768
30.4.4	Receive Filter Register Set	769
30.4.5	Data transmission.....	770
30.4.6	Single data transmission.....	770
30.4.7	Cancel data transmission.....	770
30.4.8	Data reception.....	771
30.4.9	Error Handling.....	771
30.4.10	Node shutdown.....	772
30.4.11	Arbitration Failure Location Capture.....	772
30.4.12	Loopback Mode	772
30.4.13	Silent Mode.....	773
30.4.14	Software Reset Function	774
30.4.15	Upward Compatibility with CAN-FD Functions	776
30.4.16	Time Trigger TTCAN	776

30.4.17	Interruptions.....	779
30.5	Register Description	780
30.5.1	CAN receive BUF register (CAN_RBUF).....	783
30.5.2	CAN Transmit BUF Register (CAN_TBUF)	785
30.5.3	CAN Configuration and Status Register (CAN_CFG_STAT)	787
30.5.4	CAN command register (CAN_TCMD)	788
30.5.5	CAN Transmit Control Register (CAN_TCTRL)	790
30.5.6	CAN Receive Control Register (CAN_RCTRL)	792
30.5.7	CAN Receive and Transmit Interrupt Enable Register (CAN_RTIE)	793
30.5.8	CAN Receive and Transmit Interrupt Status Register (CAN_RTIF)	794
30.5.9	CAN error interrupt enable and flag register (CAN_ERRINT)	796
30.5.10	CAN Bit Timing Register (CAN_BT)	797
30.5.11	CAN Error and Arbitration Failure Capture Register (CAN_EALCAP)	798
30.5.12	CAN warning limit register (CAN_LIMIT).....	799
30.5.13	CAN Receive Error Counter Register (CAN_RECNT)	799
30.5.14	CAN Transmit Error Counter Register (CAN_TECNT)	800
30.5.15	CAN Filter Group Control Register (CAN_ACFCTRL).....	801
30.5.16	CAN Filter Group Enable Register (CAN_ACFEN)	802
30.5.17	CAN Filter G r o u p code and mask register (CAN_ACF)	803
30.5.18	TTCAN TB slot pointer register (CAN_TBSLOT).....	804
30.5.19	TTCAN Time Trigger Configuration Register (CAN_TTCFG)	805
30.5.20	TTCAN Reference Message Register (CAN_REF_MSG)	806
30.5.21	TTCAN Trigger Configuration Register (CAN_TRG_CFG).....	807
30.5.22	TTCAN Trigger Time Register (CAN_TT_TRIG)	808
30.5.23	TTCAN Trigger Watch Time Register (CAN_TT_WTRIG)	808
30.6	Precautions for use	809
30.6.1	CAN Bus Anti-Jamming Measures	809
30.6.2	CAN Controller Noise Constraints.....	809
31	USB2.0 Full Speed Module (USBFS).....	810
31.1	Introduction to USBFS	810
31.2	USBFS Key Features.....	810
31.2.1	General Features.....	810
31.2.2	Host Mode Characteristics.....	811
31.2.3	Device Mode Characteristics	811
31.3	USBFS System Block Diagram	812

31.4	USBFS Pinout	812
31.5	USBFS Functional Description	813
31.5.1	USBFS Clock and Operating Modes	813
31.5.2	USBFS Mode Decision	813
31.5.3	USBFS Host Functions.....	814
31.5.4	USBFS Device Functions	818
31.5.5	USBFS SOF Pulse Pin Output Function	821
31.5.6	USBFS Power Consumption Control.....	822
31.5.7	USBFS Dynamic Update USBFS_HFIR Register.....	823
31.5.8	USBFS Data FIFO.....	823
31.5.9	USBFS Host FIFO Architecture.....	824
31.5.10	USBFS Device FIFO Architecture	826
31.5.11	USBFS FIFO RAM Allocation.....	827
31.5.12	USBFS System Performance.....	828
31.5.13	USBFS Interrupts and Events.....	829
31.6	USBFS Programming Model.....	830
31.6.1	USBFS Module	830
initializatio		
n		
31.6.2	USBFS Host	830
Initializatio		
n		
31.6.3	USBFS Device	831
initializatio		
n		
31.6.4	USBFS DMA Mode.....	831
31.6.5	USBFS Host Programming Model	832
31.6.6	USBFS Device Programming Model.....	833
31.6.7	USBFS Operating Model	835
31.7	Register Description	851
31.7.1	USBFS System Control Register.....	855
31.7.2	USBFS Global Registers.....	856
31.7.3	USBFS Host Mode Register.....	881
31.7.4	USBFS Device Mode Register	897
31.7.5	USBFS Clock Gating Control Registers.....	927

32 Cryptographic Coprocessing Module (CPM)	928
32.1 Introduction.....	928
32.2 Encryption and Decryption Algorithm Processor (AES).....	928
32.2.1 Introduction to Algorithms.....	928
32.2.2 Encryption Procedure	929
32.2.3 Demystifying Operational Processes	929
32.2.4 Encryption Decryption Time	930
32.2.5 Operating Precautions.....	930
32.2.6 Register Description	931
32.3 Secure Hash Algorithm (HASH).....	934
32.3.1 Introduction to Algorithms.....	934
32.3.2 Operation Process.....	935
32.3.3 Message Filling.....	935
32.3.4 Register Description	937
32.4 True Random Number Generator (TRNG)	941
32.4.1 Module Block Diagram	941
32.4.2 Operation Process.....	941
32.4.3 Interrupt and Event Output.....	941

32.4.4	Operating Precautions.....	941
32.4.5	Register Description	942
33	Data Computing Unit (DCU).....	944
33.1	Introduction.....	944
33.2	Functional Description	945
33.2.1	Addition mode.....	945
33.2.2	Subtraction mode.....	945
33.2.3	Hardware-triggered startup mode.....	945
33.2.4	Comparison Mode.....	945
33.2.5	Interrupt and Event Signal Outputs.....	946
33.3	Application Examples.....	947
33.3.1	Addition mode.....	947
33.3.2	Trigger Plus Mode	947
33.3.3	Comparison Mode.....	947
33.4	Register Description	948
33.4.1	DCU Control Register (DCU_CTL)	950
33.4.2	DCU Flag Register (DCU_FLAG)	951
33.4.3	DCU Data Register (DCU_DATAx) (x=0,1,2)	952
33.4.4	DCU Flag Reset Register (DCU_FLAGCLR)	953
33.4.5	DCU interrupt condition selection register (DCU_INTEVTSEL)	954
33.5	Cautions.....	955
34	CRC Operations (CRC)	956
34.1	Introduction.....	956
34.2	Function Block Diagram.....	956
34.3	Function Description	957
34.3.1	CRC Code Generation	957
34.3.2	CRC Checksum	957
34.3.3	CRC checksum of XOROUT,REFOUT,REFIN not all 1's.	958
34.4	Register Description	959
34.4.1	Control Register (CRC_CR).....	959
34.4.2	Result register (CRC_RESLT).....	960
34.4.3	Flag Register (CRC_FLG)	961
34.4.4	Data register (CRC_DAT).....	961
35	SDIO Controller (SDIOC)	962
35.1	Introduction.....	962

35.2 Functional Description	963
35.2.1 Port Allocation.....	963
35.2.2 Basic Access Methods.....	963
35.2.3 Data Transmission	964
35.2.4 SD Clock.....	964
35.2.5 Interrupts and DMA Startup Requests	964
35.2.6 Host and Device Initialization.....	965
35.2.7 SD/MMC single b l o c k reading and writing	967
35.2.8 SD/MMC multi block reading and writing.....	967
35.2.9 Transmission abort, suspend, and resume.	969
35.2.10 read wait.....	969
35.2.11 wakeup	970
35.3 Register Description	972
35.3.1 Data block length register (BLKSIZE)	974
35.3.2 Data Block Counter Memory (BLKCNT).....	974
35.3.3 Parameter Register 0 (ARG0).....	974
35.3.4 Parameter Register 1 (ARG1).....	975
35.3.5 Transmission Mode Register (TRANSMODE)	975
35.3.6 Command Register (CMD)	976
35.3.7 Answer register 0 (RESP0).....	976
35.3.8 Response Register 1 (RESP1)	977
35.3.9 Response Register 2 (RESP2)	977
35.3.10 Answer Register 3 (RESP3)	977
35.3.11 Response Register 4 (RESP4).....	978
35.3.12 Response Register 5 (RESP5).....	978
35.3.13 Response Register 6 (RESP6).....	978
35.3.14 Response Register 7 (RESP7).....	979
35.3.15 Data Buffer Register 0 (BUF0)	979
35.3.16 Data Buffer Register 1 (BUF1)	979
35.3.17 Host Status Register (PSTAT)	980
35.3.18 Host Control Register (HOSTCON)	981
35.3.19 Power Control Register (PWRCON)	981
35.3.20 Data block gap control register (BLKGPCON).....	982
35.3.21 Clock Control Register (CLKCON).....	982
35.3.22 Timeout Control Register (TOUTCON)	983

35.3.23 Software Reset Register (SFTRST).....	983
35.3.24 General Interrupt Status Register (NORINTST)	984
35.3.25 Error Interrupt Status Register (ERRINTST).....	984
35.3.26 Normal Interrupt Status Enable Register (NORINTSTEN)	985
35.3.27 Error Interrupt Status Enable Register (ERRINTSTEN)	986
35.3.28 General interrupt signal enable register (NORINTSGEN).....	987
35.3.29 Error interrupt signal enable register (ERRINTSGEN).....	988
35.3.30 Auto Command Error Status Register (ATCERRST)	989
35.3.31 Forced Auto Command Error Status Control Register (FEA)	989
35.3.32 Forced Error Status Control Register (FEE)	990
35.3.33 MMC Mode Enable Register (MMCER).....	991
36 Commissioning Controller (DBGC)	992
36.1 Introduction.....	992
36.2 DBGC System Block Diagram	993
36.3 SWJ-DP Debug Port (SWD and JTAG).....	994
36.3.1 Switching Mechanism for JTAG-DP or SW-DP	995
36.4 Pin Arrangement and Debug Port Pinout	996
36.4.1 SWJ Debug Port Pin	996
36.4.2 Flexible SWJ-DP Pin Assignment.....	996
36.4.3 Internal pull-ups on JTAG pins	997
36.4.4 Using the Serial Interface and Releasing Unused Debug Pins for GPIOs	997
36.5 Registers.....	998
36.5.1 DBG Status Register (MCUDBGSTAT)	998
36.5.2 Peripheral Debug Pause Register (MCUSTPCTL)	999
36.5.3 Debug Component Configuration Register (MCUTRACECTL)	1001
36.6 SW debug port.....	1002
36.6.1 SW Protocol Introduction	1002
36.7 TPIU (Trace Port Interface Unit)	1002
36.7.1 Introduction.....	1002
36.7.2 TRACE Pin Assignment	1003
36.7.3 MCU Internal TRACECLKIN Connection	1004
36.7.4 TPIU Register.....	1004
36.7.5 TPIU Configuration Example.....	1005
Version Revision Record.....	1006

table index

Table 1-1.....	
Memory Mapping	45
Table 1-2.....	QSPI
Address Space Allocation	49
Table 1-3.....	
Destination Address Configuration Example	50
Table 1-4.....	
Register List	51
Table 3-1.....	Reset
Methods and Generation Conditions	57
Table 3-2.....	Reset
Methods and Reset Flags	58
Table 3-3 RMU Register List.....	68
Table 4-1.....	
Specifications of each internal clock	75
Table 5-1.....	BOR
Configuration	121
Table 5-2.....	
PVD1/PVD2 Characteristics	122
Table 5-3.....	
Operation Modes	127
Table 5-4.....	Low
Power Mode	127
Table 5-5 Low-power mode operating conditions and status of each module in low-power mode	128
Table 5-6.....	
Operating Mode Descriptions	130
Table 5-7.....	Power-
down Mode Submodes.....	135
Table 5-8 RAM Module and RAM Power-Down Control Bits.....	139
Table 5-9.....	Register
Protection List.....	140
Table 5-10	List of
Registers 141	
Table 6-1.....	List of

registers	169
Table 7-1 Comparison of	CPU
Clock Frequency and FLASH Read Wait Cycles	179
Table 7-2FLASH Actual Read Cycle Count	181
Table 7-3	OTP
Address Composition	186
Table 7-4.....	List of Registers 188
Table 8-1 Relationship between Wait Cycle Settings for	SRAM
Read and Write Accesses and CPU Clock Frequency	198
Table 8-2.....	SRAM
Space Allocation	199
Table 9-1PORT Register List 1	211
Table 9-2PORT Register List 2	211
Table 9-3.....	List of PORT registers at 32bit access
Table 10-1	224
Interrupt Vector Table	228
Table 10-2	
Interrupt Event Request Sequence Number and Selection.....	233
Table 12-1KEYSCAN Pin Descriptions	286
Table 12-2	
KEYSCAN Register List	288
Table 14-1	Channel
Reset Description	311

Table 15-1 CMP Detailed Specifications	339
Table 15-2 CMP Register List.....	345
Table 16-1 Specifications for	Each ADC Unit 356
Table 16-2 Register Setting Methods for Converting Internal Channels	
358	
Table 16-3 Various Competitions for	Sequences A and B
361	
Table 16-4AD Conversion Times.....	364
Table 16-5 List of	ADC Registers 1/2 372
Table 17-1	OTS Register List 396
Table 18-1.....	Basic Functions and Characteristics of Timer6
398	
Table 18-2.....	Timer6 Port List
399	
Table 18-3	Comparison Table of Functions in Different Modes
433	
Table 18-4	Register List
437	
Table 19-1.....	Basic Functions and Characteristics of Timer4
463	
Table 19-2.....	Timer4 Port List
464	
Table 19-3	Register List
484	
Table 21-1.....	Basic Functions and Characteristics of TimerA
518	
Table 21-2.....	TimerA Port List
519	
Table 21-3.....	Register List
535	
Table 22-1.....	Register List
558	
Table 23-1 Basic Specifications.....	563
Table 23-2	Register List
569	
Table 24-1	Basic Characteristics of the Watchdog Counter

585

Table 24-2.....	Register List
591	
Table 25-1USART Pin Descriptions.....	595
Table 25-2.....	UART Receiver Tolerance w h e n DIV_Fraction is 0
601	
Table 25-3	UART Receiver Tolerance if DIV_Fraction i s n o t 0
601	
Table 25-4	UART Interrupt/Event Table
605	
Table 25-5	Multiprocessor Mode Interrupt/Event Table 609
Table 25-6	Smart Card Mode Interrupt/Event Table
613	
Table 25-7	Clock Synchronization Mode Interrupt/Event Table 618
Table 25-8USART Register List.....	620
Table 25-9 Baud Rate Calculation Formulas (Fractional Baud Rate Not Valid FBME=0).....	625
Table 25-10	Baud Rate Calculation Formulas (Fractional Baud Rate Valid FBME=1) 625
Table 26-1.....	Input/Output Pins
635	
Table 26-2	List of Registers
655	

Table 27-1	SPI Characterization Highlights
673	
Table 27-2	Pin Descriptions
674	
Table 27-3 SPI Pin Status Descriptions	in Host Mode 675
Table 27-4 SPI Pin Status Description	in Slave Mode 675
Table 27-5.....	Partial Setpoint Lower Speed
677	
Table 27-6SPI Mode and Register Setting Relationships.....	683
Table 27-7	Error Detection Correspondence Table
690	
Table 27-8	SPI Interrupt Source Description
695	
Table 28-1.....	QSPI Key Specifications
703	
Table 28-2.....	QSPI Pins
704	
Table 28-3	QSPI Bus Reference Clock Selection List
710	
Table 28-4.....	List of Reference Instructions
714	
Table 28-5.....	QIO2 a n d QIO3 Pin Status
QIO2 and QIO3	
Table 28-6.....	QSPI Register List
730	
Table 29-1.....	I2S Main Characteristics
741	
Table 29-2	I2S Pin Descriptions
742	
Table 29-3	Audio Frequency Accuracy (for VCO Input Frequency = 1MHz)
751	
Table 29-4	I2S Interrupt Requests
754	
Table 29-5.....	I2S Register List
757	
Table 30-1CAN Pin Descriptions.....	766

Table 30-2CAN Bit Time Setting Rules	768
Table 30-3.....	Software Reset Range Table
774	
Table 30-4CAN Interrupt Table	779
Table 30-5CAN Register List.....	780
Table 30-6CAN Register BYTE/HALFWORD/WORD Access Arrangement Table	780
Table 30-7	Standard Format CAN Receive Mailbox Format 783
Table 30-8	Extended Format CAN Receive Mailbox Format 784
Table 30-9	Standard Format CAN Send Mailbox Format 785
Table 30-10	Extended Format CAN Send Mailbox Format 786
Table 31-1USBFS Pin Descriptions	812
Table 31-2USBFS Interrupt Event Table	829
Table 31-3USBFS System Control Registers List	851
Table 31-4USBFS System Control Registers List	852
Table 34-1CRC Register List.....	959
Table 36-1 SWJ Debug Port Pins	996

Table 36-2Flexible SWJ-DP Pin Assignments 996

map index

Figure 2-1.....	Bus Architecture.....	54
Figure 3-1.....	Power-on reset	59
Figure 3-2	NRST	
Reset Timing.....		59
Figure 3-3.....		
Undervoltage Reset		60
Figure 3-4		
Programmable Voltage Detection 1 Reset.....		61
Figure 3-5		
Programmable Voltage Detection 2 Reset.....		62
Figure 3-6.....		
Watchdog and Dedicated Watchdog Reset.....		63
Figure 3-7.....	Power-down Wake-up Reset	63
Figure 3-8.....		
Software Reset		64
Figure 3-9.....	MPU	
Error Reset		64
Figure 3-10	RAM	
Parity Reset.....		64
Figure 3-11		
Ramecc Reset.....		65
Figure 3-12	Clock	
Frequency Abnormal Reset.....		65
Figure 3-13		
External High Speed Oscillation Abnormal Reset.....		66
Figure 4-1.....	Clock	
System Block Diagram		71
Figure 4-2.....	Clock	
Frequency Measurement Block Diagram.....		72
Figure 4-3.....		
External High Speed Oscillator Connection Example		77

Figure 4-4.....	
Connection Example Diagram for External Clock Input	78
Figure 4-5.....	
External High Speed Oscillator Fault Detection Example.....	79
Figure 4-6	System
Clock Selection XTAL, XTAL Oscillation Fault Detected Example.....	80
Figure 4-7.....	
External Low Speed Oscillator Connection Example	81
Figure 4-8.....	Clock
Frequency Measurement Timing Chart.....	87
Figure 5-1	Power
Supply Diagram	119
Figure 5-2	Power-
on reset and power-off reset waveforms.....	120
Figure 5-3	
Undervoltage Reset Waveform.....	121
Figure 5-4.....	PVD1
Interrupt/Reset Block Diagram	123
Figure 5-5.....	
Interrupt/Reset Block Diagram.....	123
Figure 5-6	Power
Monitor 1 Interrupt Timing Diagram	124
Figure 5-7	Power
Monitor 1 Reset Timing Diagram	124
Figure 5-8	Power
Monitor 2 Interrupt Run Timing Diagram.....	125
Figure 5-9	Power
Monitor 2 Reset Operation Timing Chart	125
Figure 5-10	Internal
Voltage Sampling Schematic.....	126
Figure 5-11	PTWKn
Structure Block Diagram	137

Figure 7-1	FLASH Address Structure (512KB Products)	176
Figure 7-2	FLASH Address Structure (256KB Product)	177
Figure 7-3	Boot Sector Swap Function	187
Figure 7-4	Activating the Switching Function	187
Figure 9-1	Schematic of Basic Port Structure	207
Figure 10-1.....	Block Diagram of Interrupt System	
		227
Figure 10-2	Interrupt Event Selection	
		249
Figure 10-3.....	Digital Filter Operation Schematic	
		250
Figure 12-1	KEYSCAN System Block Diagram	285
Figure 12-2	Keyboard Scanning Function Schematic	
		286
Figure 14-1.....	DMA Structure Diagram	305
Figure 14-2.....	Chain Transfer Schematic	
		309
Figure 14-3.....	Discontinuous Address Transmission Schematic	
		310
Figure 14-4	Discontinuous Reset Schematic	
		312
Figure 14-5	Application Example 1: Memory to Memory Transfer	315
Figure 14-6	Application Example 2: Memory to Peripheral Circuit Transfer	316
Figure 15-1	CMP, 8bitDAC Function Connection Diagram	340
Figure 15-2	Schematic diagram of CMP operation	341
Figure 15-3.....	Scan Mode Action Schematic	
		343
Figure 16-1ADC Block Diagram		355
Figure 16-2.....	Internal Analog Channel Selection	
		357
Figure 16-3	Sequence A Single Scan Mode	
		359
Figure 16-4	Continuous Scanning	
		360
Figure 16-5	Dual Sequence Scan Mode (Sequence A restarted from interrupted channel)	
		362

Figure 16-6	Dual Sequence Scan Mode (Sequence A restarted from the first channel)
362	
Figure 16-7.....	Analog Watchdog Protection Area (Comparison Conditions)
363	
Figure 16-8	A/D Conversion Time
364	
Figure 16-9	Switching action when the averaging function is active
365	
Figure 16-10	PGA Operating Schematic 366
Figure 16-11	Single Parallel Trigger Mode (Triple ADC) 367
Figure 16-12	Single Delayed Trigger Mode (Three ADCs) 368
Figure 16-13	Cyclic Parallel Trigger Mode (Triple ADC) 369
Figure 16-14	Cyclic Delay Trigger Mode (Two ADCs) 370
Figure 16-15	Cyclic Delayed Trigger Mode (Tri ADC) 370
Figure 16-16	ADC Interrupt and Event Output Timing 371

Figure 17-1	OTS Function Block Diagram	393
Figure 18-1	Timer6 Basic Block Diagram	
399		
Figure 18-2.....	Sawtooth waveform (incremental counting)	
400		
Figure 18-3.....	Triangle Waveform	
400		
Figure 18-4	Comparison Output Action	
401		
Figure 18-5	Capturing Input Actions	
402		
Figure 18-6.....	Capturing the Filter Function of the Input Port	
404		
Figure 18-7.....	Software Synchronization Action	
405		
Figure 18-8.....	Hardware Synchronization Action	
407		
Figure 18-9	Single-Cache Comparison Output Timing	
409		
Figure 18-10.....	Dual Cache Approach to Capture Input Timing	
410		
Figure 18-11.....	Count Buffer Action in Sawtooth Wave Mode	
411		
Figure 18-12 Count Buffer Action in	Triangle A Mode	
412		
Figure 18-13 Count Buffer Action in	Triangle B Mode	
413		
Figure 18-14	TIM6_<t>_PWMA Output PWM Waves	415
Figure 18-15 Software Setting GCMBR Complementary PWM Output in Triangle A Mode	416	
Figure 18-16 Hardware Setting of GCMBR Complementary PWM Wave Output in Triangle B Mode (Symmetrical Deadband)	417	
Figure 18-17.....	6-Phase PWM Wave	
418		
Figure 18-18 . Three-phase complementary PWM wave output with dead time in triangle wave A mode		
419		
Figure 18-19	Position Mode-Basic Count	

420

Figure 18-20	Position Counting Mode-Phase Difference Counting (1x Counting)	421
Figure 18-21	Position Counting Mode-Phase Difference Counting (2x Counting)	421
Figure 18-22	Position Counting Mode-Phase Difference Counting (4x Counting)	421
Figure 18-23	Position Counting Mode-Direction Counting	
422		
Figure 18-24	Rotation Counting Mode-Z Phase Counting	
422		
Figure 18-25	Rotation Counting Mode-Position Overflow Counting	
423		
Figure 18-26	Rotation Counting Mode-Mixed Counting	
423		
Figure 18-27	Rotation Counting Mode-Mixed Count Z Phase Mask Action Example 1	424
Figure 18-28	Rotation Counting Mode-Mixed Count Z Phase Mask Action Example 2	425
Figure 18-29	Cycle Interval Valid Request Signal Action	
426		
Figure 18-30.	Example of interrupt & event output during sawtooth wave mode	
436		
Figure 19-1	Timer4 Basic Block Diagram	
464		
Figure 19-2.....	Timer4 Sawtooth Waveform	
465		
Figure 19-3	Timer4 Triangle Waveform	
465		
Figure 19-4.....	Timer4 Sawtooth Wave Mode Counting Action	
465		

Figure 19-5	Timer4 Triangle Wave Mode Counting Action
466	
Figure 19-6	Example of Sawtooth Mode Waveform Output
467	
Figure 19-7	Example of Triangle Wave Mode Waveform Output
467	
Figure 19-8.....	Modifying Sawtooth Wave Count Periods with Cache Invalidation
468	
Figure 19-9	Modifying the Ragged Wave Count Period with Cache Enable
469	
Figure 19-10.	Modifying the Triangle Wave Count Period with Cache Enable
469	
Figure 19-11.	OCCR Buffered Data Transmission (When Cycle Interval Response Link Disable)
470	
Figure 19-12.	OCCR Buffered Data Transfer (Cycle Interval Response Link Enable)
471	
Figure 19-13.....	Output Compare Buffer Data Transfer (OCMR Buffer Enable)
471	
Figure 19-14	SCCR Buffered Transfer Operation (Cycle Interval Response Link Transfer Inhibit)
473	
Figure 19-15 SCCR Buffered Transfer Operation (Periodic Interval in Response to Link Transfer Enable)	
474	
Figure 19-16.	Sawtooth Wave Independent PWM Output Example
474	
Figure 19-17.....	Triangle Wave Independent PWM Output Example
475	
Figure 19-18.	Sawtooth Wave Extended PWM Output
476	
Figure 19-19.	Software Implementation of Complementary PWM Outputs
477	
Figure 19-20.....	Complementary PWM Output in Deadband Timer Mode
478	
Figure 19-21.....	Waveform Output in Deadband Timer Mode with Pulse Width Abnormality
478	
Figure 19-22.	Complementary PWM Output in Deadband Timer Filter Mode
479	
Figure 19-23.....	Cycle interval response timing diagram
480	
Figure 19-24.....	Dedicated Event Output Signal Cycle Interval Response Output
481	

Figure 19-25	Output Timing of Dedicated Event Output Signal in Delayed Start Mode
483	
Figure 21-1.....	TimerA Basic Block Diagram
519	
Figure 21-2.....	Sawtooth waveform (incremental counting)
520	
Figure 21-3.....	Triangle Waveform
520	
Figure 21-4	Compare Output Action
521	
Figure 21-5	Capturing Input Actions
522	
Figure 21-6.....	Software Synchronization Action
524	
Figure 21-7.....	Clock Input Port Filter Function
525	
Figure 21-8.....	Cache Action in Sawtooth Wave Mode
526	
Figure 21-9 32 Bit Cascade Count Action.....	527
Figure 21-10	General Purpose PWM Output Example
528	
Figure 21-11	Position Mode-Basic Count
530	
Figure 21-12	Position Counting Mode-Phase Difference Counting (1x Counting) 530
Figure 21-13	Position Counting Mode-Phase Difference Counting (2x Counting) 531
Figure 21-14	Position Counting Mode-Phase Difference Counting (4x Counting) 531

Figure 21-15	Position Counting Mode-Direction Counting
531	
Figure 21-16	Rotation Counting Mode-Z Phase Counting
532	
Figure 21-17	Rotation Counting Mode-Position Overflow Counting
532	
Figure 21-18	Rotation Counting Mode-Mixed Counting
533	
Figure 22-1	Timer0 Basic Block Diagram
554	
Figure 22-2	Timer0 Count Timing Diagram
556	
Figure 23-1 Basic Block Diagram of the	RTC 564
Figure 24-1	Hardware Boot Example
586	
Figure 24-2	Software Startup Example
587	
Fig. 24-3 Timing Examples of Various Refresh Actions (Action Acknowledgement, Falling Edge of Refresh Request Signal, etc.)	588
Figure 24-4	Counter Overflow Example
589	
Figure 24-5	Counter Refresh Action Example
590	
Figure 25-1USART System Block Diagram.....	595
Figure 25-2	UART Data Format 597
Figure 25-3UART Transmit Data Legend 1.....	599
Figure 25-4	UART Transmit Data Legend 2 599
Figure 25-5UART Internal Synchronization and Sampling Timing	600
Figure 25-6	UART Receive Data Legend 1 602
Figure 25-7	UART Receive Data Legend 2 602
Figure 25-8.	Multiprocessor Communication Legend
605	
Figure 25-9	Multiprocessor Mode Data Format
606	
Figure 25-10.....	Multi-Processor Mode Sending Data Legend
607	

Figure 25-11.....	Multi-Processor Mode Receive Data Legend 1
608	
Figure 25-12.....	Multi-Processor Mode Receive Data Legend 2
608	
Figure 25-13	Smart Card Connection Schematic
609	
Figure 25-14.....	Smart Card Mode Synchronization Timing and Sampling Timing Diagrams
610	
Figure 25-15.....	Smart Card Mode Synchronization Timing and Sampling Timing Diagrams
611	
Figure 25-16.	Smart Card Mode Sending Data Legend
612	
Figure 25-17.	Smart Card Mode Receive Data Legend
613	
Figure 25-18.....	Clock Synchronization Mode Data Format
614	
Figure 25-19.....	Clock Synchronization Mode Data Transmission Legend 1
616	
Figure 25-20.....	Clock Synchronization Mode Data Transmission Legend 2
616	
Figure 25-21.....	Clock Synchronization Mode Receive Data Legend 1
617	
Figure 25-22.	Clock Synchronization Mode Receive Data Legend 2
618	
Figure 26-1	I2C System Block Diagram
634	

Figure 26-2	I2C Bus Structure Example
635	
Figure 26-3	Timing Diagram of the I2C Bus
636	
Figure 26-4	I2C Bus Data Format
637	
Figure 26-57 Host Transmit Data Timing Diagram for Bit Address Format (Example)	638
Figure 26-67 Timing Diagram for Host Receiving Data in Bit Address Format (Example).....	639
Figure 26-77 Slave Transmit Mode Timing Diagram for Bit Address Format (Example).....	640
Figure 26-87 Bit Address Format Slave Receive Mode Timing Diagram (Example).....	641
Figure 26-9.....	SCL Synchronization Timing 642
Figure 26-10.....	Slave transmit timing diagram (1) 643
Figure 26-11.....	Slave Transmit Timing Diagram (2) 644
Figure 26-12 Timing when	selecting 7-bit address format
646	
Figure 26-13 Timing when	selecting 10-bit address format
647	
Figure 26-14.....	Digital Filter Circuit Block Diagram
654	
Figure 27-1	System Block Diagram
674	
Figure 27-2	Host Mode Structure
676	
Figure 27-3.....	Three-Wire Clock Synchronization Operation
676	
Figure 27-4.....	Data Format
678	
Figure 27-5.....	Data Transfer Format Diagram (CPHA=0) 679
Figure 27-6	Data Transfer Format (CPHA=1) 680
Figure 27-7	Full-Duplex Synchronous Serial Communication
681	
Figure 27-8.....	Transmit Only Communication
682	
Figure 27-9	Parity Check Flow
689	
Figure 27-10	Overload Error Handling

691	
Figure 27-11.....	Diagram of the action when the clock auto stop function is enabled (CPHA=1) 692
Figure 27-12.....	Diagram of the action when the clock auto stop function is enabled (CPHA=0) 692
Figure 27-13.....	Parity Error
693	
Figure 28-1.....	QSPI Module Composition Diagram
704	
Figure 28-2.....	Default Area Settings and AHB Bus Space Memory Mapping Diagram 705
Figure 28-3	QSPI-ROM Spatial Memory Map
706	
Figure 28-4	Extended SPI Protocol Action Schematic 1 (Fast Read Mode) 707
Figure 28-5	Extended SPI Protocol Action Schematic 2 (4-Wire Input/Output Fast Read Mode) 707
Figure 28-6	2-Wire SPI Protocol Action Schematic (Fast Read Mode) 708
Figure 28-7	4-Wire SPI Protocol Action Schematic (Fast Read Mode) 708
Figure 28-8.....	Basic Timing Diagram for Serial Interface
709	
Figure 28-9 Output Clock Duty Cycle Compensation Schematic for HCLK Triple Frequency Selection for Reference Clock.....	711

Figure 28-10	QSSL Establishment Time Configuration Schematic	712
Figure 28-11.	QSSN Hold Time Configuration Diagram	712
Figure 28-12.....	Data Receiving Delay Schematic	
713		
Figure 28-13.....	Standard Read Bus Cycle Schematic	
715		
Figure 28-14.....	Fast Read Bus Cycle Schematic	
716		
Figure 28-15 Schematic of Fast Read Bus Cycle with XIP Mode	Selected	
716		
Figure 28-16.....	Schematic of 2-wire output fast read bus cycle	
717		
Figure 28-17 Schematic of Fast Read Bus Cycle for 2-Wire Outputs with XIP Mode	Selected	
717		
Figure 28-18.	Schematic of 2-Wire Input/Output Fast Read Bus Cycle	
718		
Figure 28-19 Schematic of 2-Wire Input/Output Fast Read Bus Cycle with XIP Mode	Selected	
718		
Figure 28-20.....	Schematic of Fast Read Bus Cycle for 4-Wire Outputs	
719		
Figure 28-21 Schematic of Fast Read Bus Cycle for Four-Wire Outputs with XIP Mode ...	Selected	
719		
Figure 28-22.....	Schematic of 4-Wire Input/Output Fast Read Bus Cycle	
720		
Figure 28-23 Schematic of Four-Wire Input-Output Fast Read Bus Cycle with XIP Mode	Selected	
721		
Figure 28-24.	Entry into 4-Byte Mode Instruction Bus Cycle Diagram	
721		
Figure 28-25.....	Exit 4-Byte Mode Instruction Bus Cycle Diagram	
722		
Figure 28-26.	Write Permit Instruction Bus Cycle Schematic	
722		
Figure 28-27.	Schematic of Single Flash Data Read Operation with Independent Conversion	
723		
Figure 28-28.....	Diagram of data reading operation when the pre-reading function is active	
724		

Figure 28-29 Data Read Operation Schematic	Using QSPI Bus Cycle Extension Function
Figure 28-29 Data Read Operation Diagram Using QSPI Bus Cycle Extension	
Figure 28-30	XIP Mode Control Schematic
725	
Figure 29-1	I2S System Block Diagram
742	
Figure 29-2	I2S Philips Protocol Waveforms (16/32-bit Full Precision)
744	
Figure 29-3 ...	I2S Philips Protocol Waveform (16-bit data encapsulated in a 32-bit frame) 745
Figure 29-4 ...	I2S Philips Protocol Waveform (24-bit data encapsulated in a 32-bit frame)
Figure 29-4	I2S Philips Protocol Waveform
Figure 29-5.....	I2S MSB Protocol Waveform (16/32-bit Full Precision) 745
Figure 29-6	I2S MSB Protocol Waveform (16-bit data encapsulated in a 32-bit frame)
Figure 29-6	I2S MSB Protocol Waveform
Figure 29-7	I2S MSB Protocol Waveform (24-bit data encapsulated in a 32-bit frame)
Figure 29-7	I2S MSB Protocol Waveform
Figure 29-8.....	I2S LSB Protocol Waveform (16/32-bit Full Precision)
746	
Figure 29-9	I2S LSB Protocol Waveform (16-bit data encapsulated in a 32-bit frame)
Figure 29-9	I2S LSB Protocol Waveform
Figure 29-10.....	I2S LSB Protocol Waveform (24-bit data encapsulated in a 32-bit frame)
Figure 29-10	I2S LSB Protocol Waveform
Figure 29-11.....	I2S PCM Protocol Waveforms (16/32-bit Full Precision) 748
Figure 29-12	I2S PCM Protocol Waveform (16-bit data encapsulated in a 32-bit frame) 748
Figure 29-13	I2S PCM Protocol Waveform (24-bit data encapsulated in a 32-bit frame) 749
Figure 29-14	Audio Sample Frequency Definition
750	

Figure 29-15	Clock Generator Architecture
750	
Figure 29-16	Host Receives Only Temporarily Stopped Receiving
755	
Figure 29-17.	PCM Short Frame Host Transmission Pause and Resend Method I
756	
Figure 29-18.....	PCM Short Frame Host Transmission Pause and Resend Method II
756	
Figure 30-1CAN System Block Diagram.....	766
Figure 30-2CAN Bit Time Definition Diagram.....	767
Figure 30-3CAN TBUF Register Write Transmit Buffer and Diagram.....	768
Figure 30-4CAN RBUF Register Read Receive Buffer Schematic	769
Figure 30-5CAN ACF Register Access Filter Group Schematic	769
Figure 30-6.....	CAN LBMI and LBME Schematics
Figure 30-6 CAN LBMI and LBME Schematic	
Figure 31-1	USBFS System Block Diagram
812	
Figure 31-2	USBFS Host Mode System Build Diagram
814	
Figure 31-3	USBFS Device Mode System Build Diagram
818	
Figure 31-4.....	USBFS Dynamic U p d a t e USBFS_HFIR Register Schematic
823	
Figure 31-5.....	USBFS Host Mode FIFO Architecture Schematic
824	
Figure 31-6.....	FIFO Architecture Schematic in USBFS Device Mode
826	
Figure 32-1AES Encryption Flow Diagram	928
Figure 32-2.....	HASH Algorithm Flowchart
934	
Figure 32-3	TRNG System Block Diagram
941	
Figure 34-1CRC Module Block Diagram.....	956
Figure 36-1.....	Commissioning Control System
993	
Figure 36-2.....	Commissioning Control System
994	
Figure 36-3	JTAG-DP to SW-DP Switching Sequence
995	
Figure 36-4.....	TPIU Block Diagram
1002	

Introduction (Overview)

This series is a high-performance MCU based on the ARM® Cortex®-M4 32-bit RISC CPU, with a maximum operating frequency of 200MHz. The Cortex-M4 core integrates a floating-point unit (FPU) and a DSP, which realizes single-precision floating-point arithmetic operations, and supports all ARM single-precision data-processing instructions and data types, as well as the complete DSP instruction set. The core has an integrated MPU and a dedicated MPU for DMAC on top of it to ensure the safety of the system operation.

The HC32F460_F45x_A460 series integrates high-speed on-chip memories, including up to 512KB of Flash and up to 192KB of SRAM, and a Flash access acceleration unit, which enables the CPU to execute single-cycle programs on Flash. The polled bus matrix supports multiple bus hosts to access memory and peripherals at the same time to improve performance. The bus hosts include CPU, DMA, and USB dedicated DMA. In addition to the bus matrix, it supports data transfer between peripherals, basic arithmetic operations and event triggering, which can significantly reduce the transaction load of the CPU.

This series integrates rich peripheral functions. Including two independent 12bit 2.5MSPS ADCs, one gain-adjustable PGA, three voltage comparators (CMP) three multifunctional 16bit PWM timers (Timer6) supporting six complementary PWM outputs, three motor PWM timers (Timer4) supporting 18 complementary PWM outputs, six 16bit general-purpose timers (TimerA) supporting three 3-phase quadrature encoding inputs and 48 Duty independently programmable PWM outputs, 11 serial communication interfaces (I2C and I2C). Timer (TimerA) supports 3 3-phase quadrature encoding inputs and 48 Duty independently programmable PWM outputs, 11 serial communication interfaces (I2C/UART/SPI)1 QSPI interface, 1 CAN, 4 I2S audio PLLs, 2 SDIOs, 1 USB FS Controller with on-chip FS PHY support. Device/Host.

The series supports wide voltage range (1.8-3.6V)widetemperaturerange(-40-105°C)and various low power modes, and can switch between ultra-high-speed mode ($\leq 200\text{MHz}$) high-speed mode ($\leq 168\text{MHz}$), and ultra-low-speed mode ($\leq 8\text{MHz}$)n Run mode and Sleep mode. Supports fast wake-up in low-power modes, as fast as 2us in STOP mode and 20us in Power Down mode.

typical application

This series is available in 48pin,64pin, and 100pin LQFP packages, 32pin,48pin, and 60pin QFN packages, and 100pin VFBGA packages for automotive electronics, high-performance motor inverter control, smart hardware, IoT connectivity modules, and other applications.

About this manual

This manual describes the functions, operation and usage of the chip. For the specifications of the chip, refer to the corresponding "Data Sheet".

1 Memory Mapping)

1.1 memory map

The MCU supports a 4GB linear address space with addresses from 0x0000_0000 to 0xFFFF_FFFF. see Table 1-1 for details of the memory map.

Table 1-1 Memory Mapping

Memory Classification		starting address	end address	space size	Modules*3	Protection *4	clarification
systems	Private Peripheral External Bus	0xE010_0000	0xFFFF_FFFF	511MB	Reserved		Customized space
		0xE00F_F000	0xE00F_FFFF	4KB	ROMTABLE		
		0xE004_2400	0xE00F_EFFF	755KB			
		0xE004_2000	0xE004_23FF	1KB	DBG		
		0xE004_1000	0xE004_1FFF	4KB	ETM		
	Private Peripheral Internal Bus	0xE004_0000	0xE004_0FFF	4KB	TPIU		
		0xE000_F000	0xE003_FFFF	196KB			
		0xE000_E000	0xE000_EFFF	4KB	SCS		System control space NVIC/MPU et al. (and other authors)
		0xE000_3000	0xE000_DFFF	44KB			
		0xE000_2000	0xE000_2FFF	4KB	FPB		
	External installations	0xE000_1000	0xE000_1FFF	4KB	DWT		
		0xE000_0000	0xE000_0FFF	4KB	ITM		
External RAM	AHB5 Clock: HCLK	0xA000_0000	0xFFFF_FFFF	1024MB	Reserved		
	-	0x9800_0000	0x9FFF_FFFF	128MB	QSPI		
		0x6000_0000	0x97FF_FFFF	896MB	Reserved		
	-	0x4400_0000	0x5FFF_FFFF	448MB	Reserved		
		0x4200_0000	0x43FF_FFFF	32MB	PeriBitBand		Host other than CPU Reserved

periph erals		0x4010_0000	0x41FF_FFFF	31MB	Reserved		
	AHB3 Clock: PCLK1	0x400C_0000	0x400F_FFFF	256KB	USBFS		

Memory Classification		starting address	end address	space size	Modules*3	Protection *4	clarification
AHB2 Clock: PCLK1	0x4007_0800	0x400B_FFFF	318KB	BLANK			
	0x4007_0400	0x4007_07FF	1KB	CAN			
	0x4007_0000	0x4007_03FF	1KB	SDIOC_2			
	AHB4 Clock: PCLK1	0x4006_FC00	0x4006_FFFF	1KB	SDIOC_1		
		0x4006_0000	0x4006_FBFF	63KB	BLANK		
	AHB1 Clock: HCLK	0x4005_5800	0x4005_FFFF	42KB	BLANK		
		0x4005_5400	0x4005_57FF	1KB	PERIC		
		0x4005_4800	0x4005_53FF	3KB	BLANK		
		0x4005_4000	0x4005_47FF	2KB	SYSC	strip protection	
		0x4005_3800	0x4005_3FFF	2KB	GPIO		
		0x4005_3400	0x4005_37FF	1KB	DMA_2		
		0x4005_3000	0x4005_33FF	1KB	dma_1		
		0x4005_2C00	0x4005_2FFF	1KB	dcu_4		
		0x4005_2800	0x4005_2BFF	1KB	dcu_3		
		0x4005_2400	0x4005_27FF	1KB	DCU_2		
	APB4	0x4005_2000	0x4005_23FF	1KB	DCU_1		
		0x4005_1000	0x4005_1FFF	4KB	INTC	strip protection	
		0x4005_0C00	0x4005_0FFF	1KB	KEYSCAN		
		0x4005_0800	0x4005_0BFF	1KB	RAMIF	strip protection	
		0x4005_0400	0x4005_07FF	1KB	BLANK		
		0x4005_0000	0x4005_03FF	1KB	DMPU	strip protection	
	APB4	0x4004_EC00	0x4004_FFFF	5KB	BLANK		
		0x4004_E800	0x4004_EBFF	1KB	I2C_3		
		0x4004_E400	0x4004_E7FF	1KB	I2C_2		
		0x4004_E000	0x4004_E3FF	1KB	I2C_1		
		0x4004_C400	0x4004_DFFF	7KB	BLANK		
		0x4004_C000	0x4004_C3FF	1KB	RTC	strip protection	
		0x4004_A800	0x4004_BFFF	6KB	BLANK		

	Clock: PCLK3	0x4004_A400	0x4004_A7FF	1KB	OTS		
		0x4004_A000	0x4004_A3FF	1KB	CMP		
		0x4004_9800	0x4004_9FFF	2KB	BLANK		
		0x4004_9400	0x4004_97FF	1KB	SWDT	strip protection	
		0x4004_9000	0x4004_93FF	1KB	WDT	strip protection	
		0x4004_8800	0x4004_8FFF	2KB	BLANK		
		0x4004_8400	0x4004_87FF	1KB	FCM		
		0x4004_8000	0x4004_83FF	1KB	MSTP	strip protection	

Memory Classification		starting address	end address	space size	Modules*3	Protection *4	clarification
peripherals	APB3	0x4004_1400	0x4004_7FFF	27KB	BLANK		
		0x4004_1000	0x4004_13FF	1KB	TRNG	strip protection	
		0x4004_0800	0x4004_0FFF	2KB	BLANK		
	Clock: PCLK4	0x4004_0400	0x4004_07FF	1KB	ADC_2		
		0x4004_0000	0x4004_03FF	1KB	ADC_1		
	APB2 Clock: PCLK1	0x4002_5000	0x4003_FFFF	108KB	BLANK		
		0x4002_4C00	0x4002_4FFF	1KB	Timer4_3		
		0x4002_4800	0x4002_4BFF	1KB	Timer4_2		
		0x4002_4400	0x4002_47FF	1KB	Timer0_2		
		0x4002_4000	0x4002_43FF	1KB	Timer0_1		
		0x4002_2800	0x4002_3FFF	6KB	BLANK		
		0x4002_2400	0x4002_27FF	1KB	I2S_4		
		0x4002_2000	0x4002_23FF	1KB	I2S_3		
		0x4002_1800	0x4002_1FFF	2KB	BLANK		
		0x4002_1400	0x4002_17FF	1KB	USART_4		
		0x4002_1000	0x4002_13FF	1KB	USART_3		
		0x4002_0800	0x4002_0FFF	2KB	BLANK		
	APB1 Clock: PCLK1	0x4002_0400	0x4002_07FF	1KB	SPI_4		
		0x4002_0000	0x4002_03FF	1KB	SPI_3		
		0x4001_E800	0x4001_FFFF	6KB	BLANK		
		0x4001_E400	0x4001_E7FF	1KB	I2S_2		
		0x4001_E000	0x4001_E3FF	1KB	I2S_1		
		0x4001_D800	0x4001_DFFF	2KB	BLANK		
		0x4001_D400	0x4001_D7FF	1KB	USART_2		
		0x4001_D000	0x4001_D3FF	1KB	USART_1		
	APB1 Clock: PCLK1	0x4001_C800	0x4001_CFFF	2KB	BLANK		
		0x4001_C400	0x4001_C43F	1KB	SPI_2		
		0x4001_C000	0x4001_C3FF	1KB	SPI_1		
		0x4001_8000	0x4001_BFFF	16KB	Timer6		count Counting Clock:PCL K0
		0x4001_7C00	0x4001_7FFF	1KB	EMB		
		0x4001_7400	0x4001_7BFF	2KB	BLANK		
		0x4001_7000	0x4001_73FF	1KB	Timer4_1		
		0x4001_6800	0x4001_6FFF	2KB	BLANK		

		0x4001_6400	0x4001_67FF	1KB	TimerA_6		
		0x4001_6000	0x4001_63FF	1KB	TimerA_5		
		0x4001_5C00	0x4001_5FFF	1KB	TimerA_4		

Memory Classification		starting address	end address	space size	Modules*3	Protection *4	clarification
SRA M	AHB3 Clock: PCLK1	0x4001_5800	0x4001_5BFF	1KB	TimerA_3		
		0x4001_5400	0x4001_57FF	1KB	TimerA_2		
		0x4001_5000	0x4001_53FF	1KB	TimerA_1		
		0x4001_0C00	0x4001_4FFF	17KB	BLANK		
		0x4001_0800	0x4001_0BFF	1KB	AOS		Internal trigger event register area
		0x4001_0400	0x4001_07FF	1KB	EFM	strip protection	
		0x4001_0000	0x4001_03FF	1KB	BLANK		
	SRAM Clock: HCLK	0x4000_9000	0x4000_FFFF	28KB	BLANK		
		0x4000_8C00	0x4000_8FFF	1KB	CRC	strip protection	
		0x4000_8800	0x4000_8BFF	1KB	BLANK		
		0x4000_8400	0x4000_87FF	1KB	HASH256	strip protection	
		0x4000_8000	0x4000_83FF	1KB	AES128	strip protection	
	-	0x4000_0000	0x4000_7FFF	32KB	Reserved		
SRAM M	-	0x2400_0000	0x3FFF_FFFF	448MB	Reserved		
		0x2200_0000	0x23FF_FFFF	32MB	SRAMBitBand		Host other than CPU Reserved
		0x2010_0000	0x21FF_FFFF	31MB	Reserved		
	SRAM Clock: HCLK	0x200F_1000	0x200F_FFFF	60KB	BLANK		
		0x200F_0000	0x200F_0FFF	4KB	Ret_SRAM		
		0x2002_7000	0x200E_FFFF	804KB	BLANK		
		0x2002_0000	0x2002_6FFF	28KB	SRAM3		ECCRAM
		0x2001_0000	0x2001_FFFF	64KB	SRAM2		
		0x2000_0000	0x2000_FFFF	64KB	SRAM1		
	SRAM Clock: HCLK	0x1FFF_8000	0x1FFF_FFFF	32KB	SRAMH		
	-	0x0318_0000	0x1FFF_7FFF	462.4MB	BLANK		

CODE	Flash Clock: HCLK	0x0317_FFE0	0x0317_FFFF	32B	Data security protection		For configuri ng data security protection
	-	0x0210_0000	0x0317_FFDF	16.5MB	BLANK		
		0x0208_0000	0x020F_FFFF	512KB	REMAP1		address remapping Region 1

Memory Classification	starting address	end address	space size	Modules ^{*3}	Protection ^{*4}	clarification
REMAPPING AREA	REMAPP Clock: HCLK	0x0200_0000	0x0207_FFFF	512KB	REMAP0	Address Remapping Area 0
	-	0x0008_0000	0x01FF_FFFF	31.5MB	BLANK	
	Flash Clock: HCLK	0x0000_0000	0x0007_FFFF	512KB	Embedded Flash ^{*5}	

*1 Please refer to the ARM Cortex-M4 instruction manual "Memory System".

*2 Refer to the [Bus Architecture (BUS)] section for bus descriptions.

*3 Reserved: access to the bus causes a bus error; BLANK: write access is invalid, read access reads 0.

*4 Modules with protection function only support CPU privileged mode access when protection function is active. Refer to the DMPU chapter for specific registers and descriptions.

*5 In the 256KB product, the Flash address is 0x0000_0000~0x0003_FFFF.

1.2 External Space Mapping

The QSPI space is divided into two segments, the QSPI I/O register space 64MB and the external QSPI device space 64MB. The QSPI space is divided into two segments, QSPI I/O register space of 64MB and external QSPI device space of 64MB, please refer to Table 1-2 for the allocation relationship.

Table 1-2 QSPI Address Space Allocation

QSPI	0x9800_0000	0x9FFF_FFFF	128MB	QSPI I/O Register	0x9C00_0000	0x9FFF_FFFF	64MB
				External QSPI Device	0x9800_0000	0x9BFF_FFFF	64MB

1.3 space of bits

The Cortex™-M4F memory map consists of two bit segment regions. These regions map each word in the memory alias region to the corresponding bit in the memory bit segment region. Writing a word in the alias region is equivalent to performing a read-modify-write operation to the target bit in the bit segment region.

In this MCU, the peripheral registers and SRAM are mapped to a single bit segment area, which enables read and write operations on a single bit segment. These operations are only available

for Cortex™-M4F accesses and are not available for other bus master interfaces such as
DMA.

1.4 address remapping

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

Remap address 0:

0x0200_0000~0x0208_0000 (depending on remap space MMF_REMCR0/1.RMSIZE[4:0])

Remap Address 1:

0x0208_0000~0x0210_0000 (depending on remap space MMF_REMCR0/1.RMSIZE[4:0])

When the remapping function is effective, the address correspondence is shown in Table 1-3.

Table 1-3 Destination Address Configuration Example

Register Setting	remap address (CPU address - CPUADDR[31:0])	source address		
		Higher 3-bit address	medium address	low address
RMSIZE[4:0]=01110 circumstances (Remapping space: 16K)	0x0200_0000~0x0200_3FFF	All 0	RMTADDR[16:2]	CPUADDR[13:0]
RMSIZE[4:0]=01111 Situation (Remapping space: 32K)	0x0200_0000~0x0200_7FFF	All 0	RMTADDR[16:3]	CPUADDR[14:0]
RMSIZE[4:0]=10000 scenarios (Remapping space: 64K)	0x0208_0000~0x0208_FFFF	All 0	RMTADDR[16:4]	CPUADDR[15:0]
RMSIZE[4:0]=10001 circumstances (Remapping space: 128K)	0x0208_0000~0x0209_FFFF	All 0	RMTADDR[16:5]	CPUADDR[16:0]

For example, to use the remap address 0 function, set the source address to be the main flash FLASH address 0x0000_8000, the remap space is 32K, and the register MMF_REMCR0 should be set to 0x8000_800F.

To use the remap address 1 function, set the source address to the high-speed SRAM address 0x1FFF_8000, the remap space is 16K, and the register MMF_REMCR1 should be set to 0x9FFF_800E.

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

1.5 Remap Register

There are three registers in the remap module. The address space is as follows:
 Register base address: 0x4001_0500

Table 1-4 Register List

register name	notation	offset address	bit width	reset value
Access Protection Register	MMF_REMPRT	0x0000	32	0x0000_0000
Remap Control Register 0	MMF_REMCRO	0x0004	32	0x0000_0000
Remap Control Register 1	MMF_REMCR1	0x0008	32	0x0000_0000

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]																
<hr/>																
classifier for marking	celebrity	functionality	fill out or in (information on a form)													
honorable people																
b31~b16	Reserved	-	Reads "0", writes "0".	R												
b15~b0	MMF_REMPRT[15:0]	Protection Register	Registers MMF_REMCRO and MMF_REMCR1 are write-protected: Write 0x0123 to MMF_REMPRT[15:0] followed by 0x3210 to unlock Protection; Read registers MMF_REMCRO and MMF_REMCR1 in write-protected state to 0 When registers MMF_REMCRO and MMF_REMCR1 are released from the write-protect state, the read send Depository is 1	R/W												

1.5.2 Remap control register (MMF_REMCRx|x=0, 1)

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EN	-														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]							
<hr/>															
classified for people	markin EN		celebrity Remap Valid Bit	function 0: Remap Invalid 1: Remapping is effective											fill out or in
b30~b29	Reserved		-0" for reading, "0" for writing. writes "0".												(Reads "0", rma)
b28~b12	RMTADDR[16:0]		source address	The number of valid bits is related to the RMSIZE[4:0] setting. The settings can be found in Table 1-3.											tion R/W on a for
b11~b5	Reserved	-		Reads "0" and writes "0".											R)
			Setting the remapping space 00000~01011: Reserved, setup prohibited 01100: 4KB 01101: 8KB 01110: 16KB 01111: 32KB 10000: 64KB 10001: 128KB 10010: 256KB 10011: 512KB (256KB disabled by product setting) 10100 to 11111: Reserved, setup disabled												R/W

2 Bus Architecture (BUS)

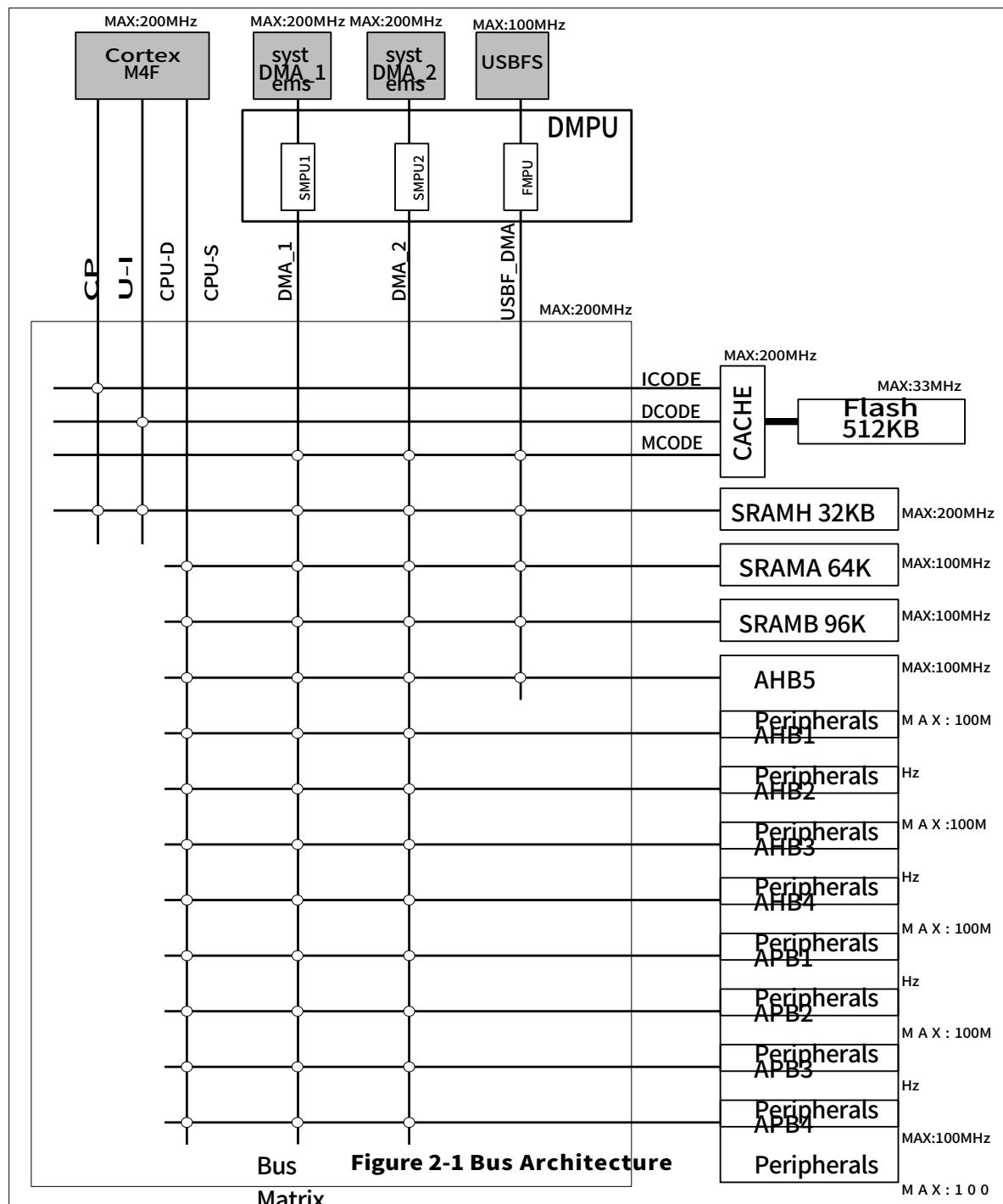
2.1 summarize

The master system consists of a 32-bit multilayer AHB bus matrix that interconnects the following master and slave buses:

- Host Bus
 - Cortex-M4F Core CPU-I Bus, CPU-D Bus, CPU-S Bus
 - System DMA_1 Bus, System DMA_2 Bus
 - USBFS_DMA Bus
- slave bus (computer)
 - Flash ICODE Bus
 - Flash DCODE Bus
 - Flash MCODE bus (bus for hosts other than the CPU to access Flash)
 - High-speed SRAMH (SRAMH 32KB) bus
 - System SRAMA (SRAM1 64KB) bus
 - System SRAMB (SRAM2 64KB, SRAM3 28KB, Ret_SRAM 4KB) Bus
 - APB1 Peripheral Bus (EMB/Timers/SPI/USART/I2S)
 - APB2 Peripheral Bus (Timers/SPI/USART/I2S)
 - APB3 Peripheral Bus (ADC/PGA/TRNG)
 - APB4 Peripheral Bus (FCM/WDT/CMP/OTS/RTC/WKTM/I2C)
 - AHB1 Peripheral bus (KEYSCAN/INTC/DCU/GPIO/SYSC)
 - AHB2 Peripheral Bus (CAN/SDIOC)
 - AHB3 Peripheral Bus (AES/HASH/CRC/USBFS)
 - AHB4 Peripheral Bus (SDIOC)
 - AHB5 Peripheral Bus (QSPI)

With the help of the bus matrix, efficient concurrent access from the master bus to the slave bus can be realized.

2.2 bus architecture



HC32F460_F45x_A460 Series Reference
Manual_Rev1.6

The bus matrix is used to manage access arbitration between the host buses. The arbitration is done using a round-robin scheduling algorithm.

- CPU-I bus

The M4F kernel has an instruction bus through which the CPU acquires instructions. The access objects are Flash and SRAMH containing the code.

- CPU-D bus

The M4F kernel has a data bus through which the CPU performs immediate digital loads and debug accesses. The access objects are Flash and SRAMH containing code or data.

- CPU-S bus

The M4F kernel's system bus, through which the CPU accesses peripherals or the system SRAM, and through which it can also fetch instructions and immediate numbers (less efficiently than through the **CPU-I** and CPU-D buses) The access objects are SRAMA/SRAMB/all peripherals and the AHB5 external expansion space.

- DMA_1 Bus, DMA_2 Bus

System DMA_1/System DMA_2 dedicated bus through which DMA_1/DMA_2 accesses data memory and peripherals for Flash/SRAMH/SRAMA/SRAMB/all peripherals as well as AHB5 external expansion space.

- USBFS_DMA Bus

USBFS has a dedicated DMA bus through which USBFS accesses all memory spaces. The access objects are Flash/SRAMH/SRAMA/SRAMB/AHB5 external expansion space.

2.3 bus function

The bus is responsible for realizing read and write accesses from the master to the slave. When the operating frequency of the host module is higher than that of the slave module (e.g., CPU-S accesses RTC), the bus automatically performs down-synchronization. When the operating frequency of the host module is lower than that of the slave module (e.g. USBFS_DMA accesses SRAMH), the bus automatically performs the up-synchronization process.

Through the bus matrix, when the access targets of different host buses do not conflict, each access can be performed at the same time. For example, **CPU-I** accesses Flash, CPU-D accesses SRAMH, CPU-S accesses APB peripheral, DMA_1 accesses SRAMA, DMA_2 accesses SRAMB, and USBFS_DMA accesses the external expansion space of AHB5, and these accesses can be performed simultaneously.

3 Reset control (RMU)

3.1 summary

The chip is configured with 14 reset methods.

- Power-On Reset (POR)
- NRST Pin Reset (NRST)
- Undervoltage Reset (BOR)
- Programmable Voltage Detection 1 Reset (PVD1R)
- Programmable Voltage Detection 2 Reset (PVD2R)
- Watchdog Reset (WDTR)
- Specialized 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 stop reset (XTALER)

3.2 Reset mode and reset flag bit

The reset mode and generation conditions are shown in Table 3-1.

Table 3-1 Reset Methods and Generation Conditions

reset mode	conditions for the emergence of
Power-on reset	VCC Power-up
NRST Pin Reset	NRST pin input low
Undervoltage 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
Dedicated Watchdog Reset	Refresh error or overflow error occurs with the dedicated watchdog
Wake on power down reset	By setting the reset generated by the power-down mode, the kernel wakes up from the reset state after a power-down wake-up event occurs
software reset	Setting the reset register bit (ARM register AIRCR.SYSRESETREQ bit)
MPU Error Reset	Reset from MPU access error
RAM Parity Reset	RAM Reset generated when a parity error occurs
RAM ECC Error Reset	Reset generated when an ECC error occurs in the RAM.
Clock frequency abnormal reset	When the clock frequency monitoring function (FCM) detects a clock cycle error
External high-speed oscillator abnormal stop reset	Reset generated when the external high-speed oscillator stops oscillating abnormally.

When a reset occurs, the chip sets the corresponding reset flag bit according to the reset mode, and the reset flag bits are shown in Table 3-2. For example, if a pin reset occurs, the pin reset flag bit PINRF is set to 1. After PINRF is set, PINRF can be cleared by writing CLRF.

Table 3-2 Reset Methods and Reset Flags

reset flag	reset mode											
External high-speed oscillator	-	-	-	-	-	-	-	-	-	-	-	-
Clock frequency abnormal	-	-	-	-	-	-	-	-	-	-	-	-
false reset	-	-	-	-	-	-	-	-	-	-	-	-
Parity error reset	-	-	-	-	-	-	-	-	-	-	-	-
false reset	-	-	-	-	-	-	-	-	-	-	-	-
software reset	-	-	-	-	-	-	-	-	-	-	-	-
Wake on power down reset	-	-	-	-	-	-	-	-	-	-	-	-
Dedicated Watchdog Reset	-	-	-	-	-	-	-	-	-	-	-	-
watchdog reset	-	-	-	-	-	-	-	-	-	-	-	-
Voltage Detection Reset	-	-	-	-	-	-	-	-	-	-	-	-
Voltage Detection Reset	-	-	-	-	-	-	-	-	-	-	-	-
Undervoltage reset	-	-	-	-	-	-	-	-	-	-	-	-
Programmable voltage detection 1 reset flag (RMU_RSTF0.PVD1RF)	X	X	X	✓	-	-	-	-	-	-	-	-
Programmable voltage detection 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 exception reset (RMU_RSTF0.CKFERF)	X	X	X	-	-	-	-	X	-	-	-	✓
External high-speed oscillator abnormal stop reset (RMU_RSTF0.XTALERF)	X	X	X	-	-	-	-	X	-	-	-	✓

✓: set X: clear - : unchanged

3.3 reset time sequence

3.3.1 Power-on reset

Power-on reset is an internal reset caused by the power-on reset circuit, and the timing is shown in Figure 3-1. A power-on reset is generated by turning on the power with the NRST pin set high, and the internal reset is released and the CPU starts to execute the code after the VCC voltage is higher than the power-on reset voltage, V_{POR} , for a certain period of time (T_{RSTPOR}). When power-on reset is generated, the power-on reset flag RMU_RSTF0.PORF is set. For details of power-on reset, please refer to [Power-on reset/power-down reset action description]

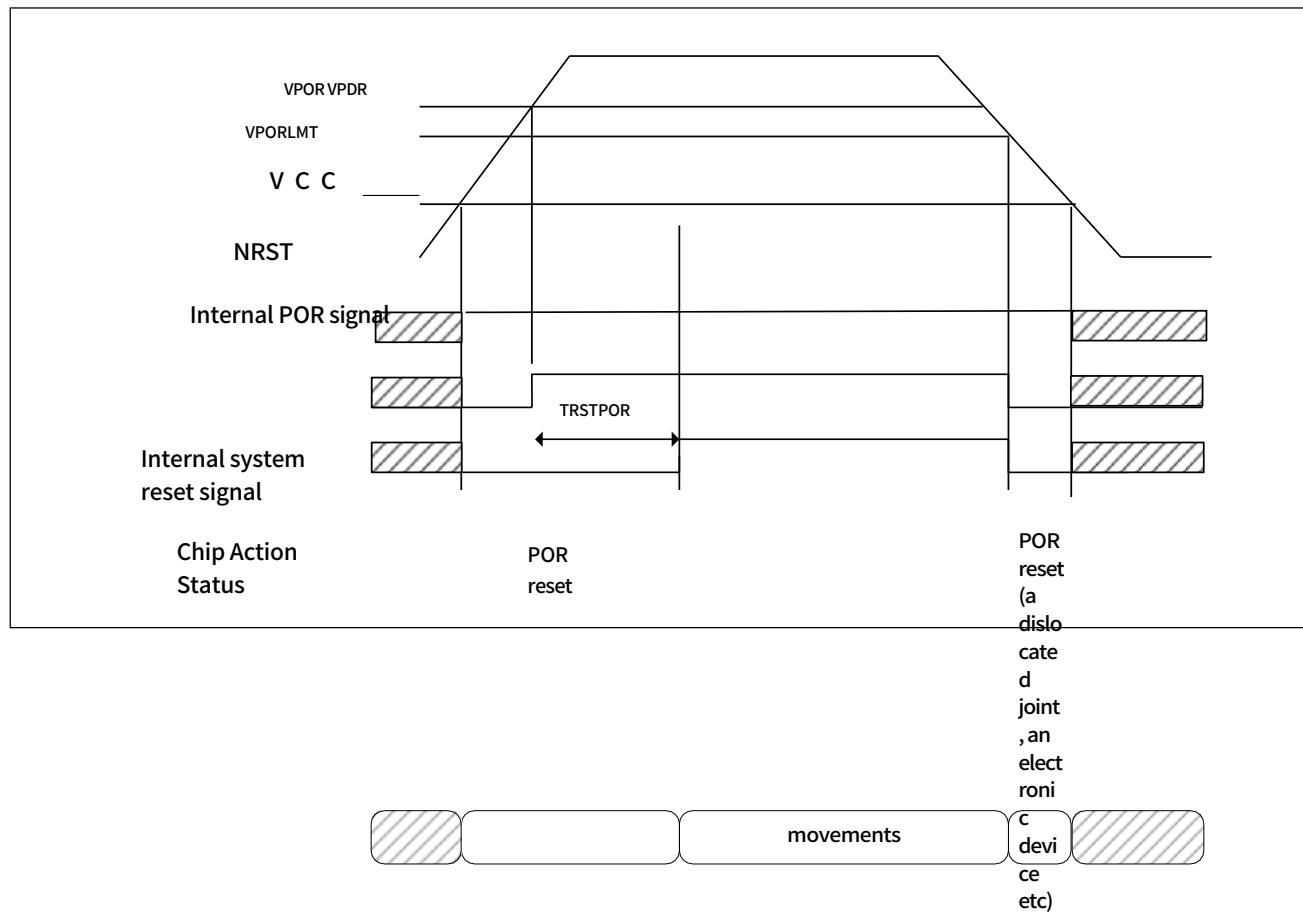


Figure 3-1 Power-on Reset

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 Figure 3-2. After the NRST pin maintains a low level above the T_{NRST} width, it is released from internal reset after a certain internal reset time (T_{INRST})

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

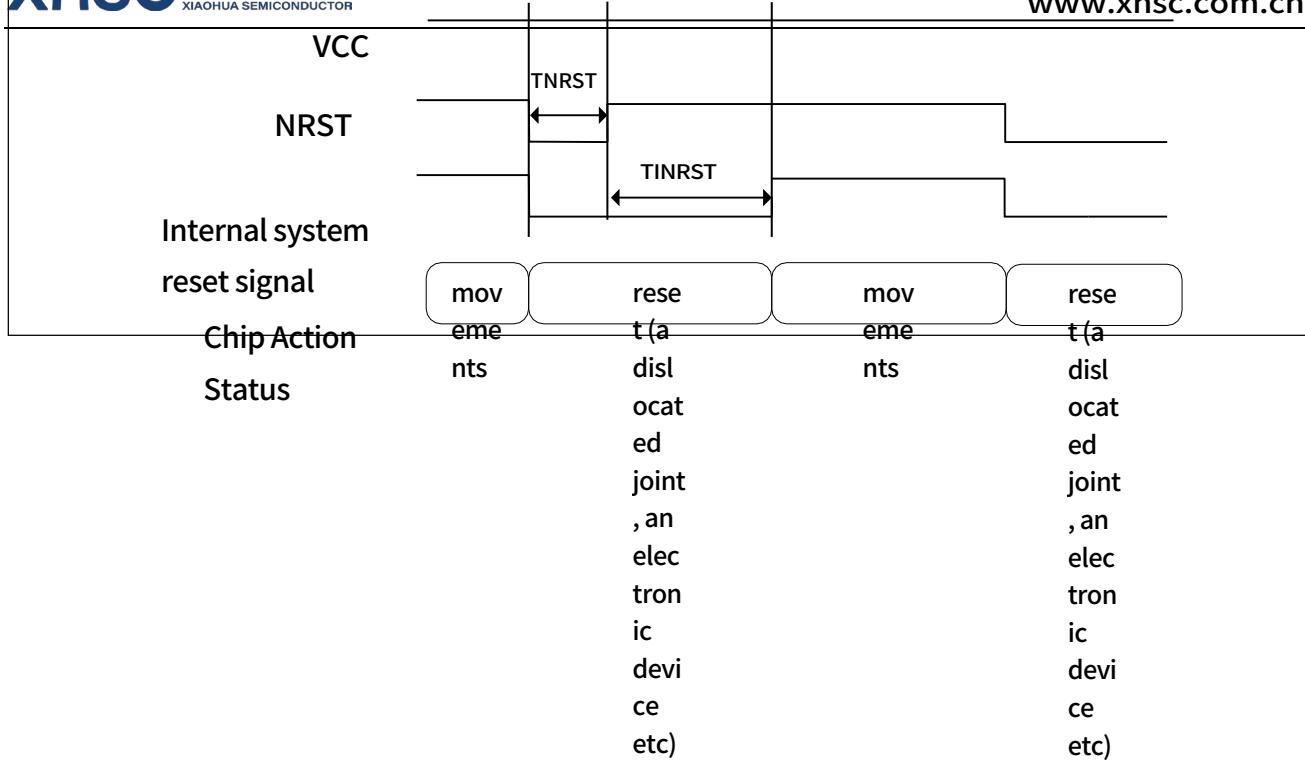


Figure 3-2 NRST Reset Timing

3.3.3 Undervoltage reset

The undervoltage reset is an internal reset caused by the voltage monitoring circuit, and the timing is shown in Figure 3-3. After the undervoltage is set to reset enable through the ICG register, if the VCC voltage is lower than the monitoring voltage V_{BOR} , RMU_RSTF0.VBORF is set. When the VCC voltage is higher than the monitoring voltage V_{BOR} , the reset is released after the reset time (t_{RSTBOR}) of V_{BOR} .

For the reset setting of undervoltage, refer to [Undervoltage Reset (BOR) Description]

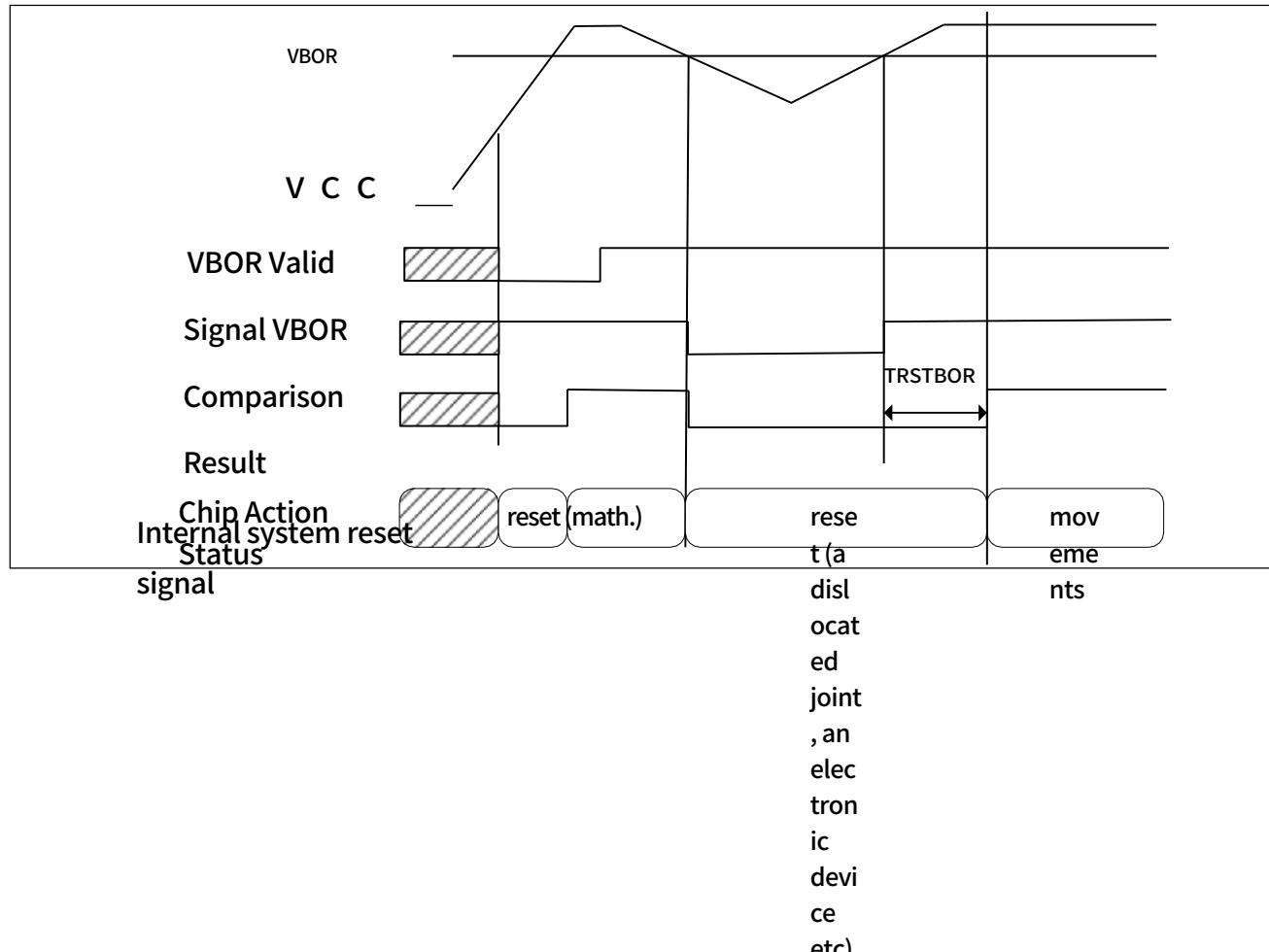


Figure 3-3 Undervoltage Reset

3.3.4 Programmable Voltage Detection 1 reset, Programmable Voltage Detection 2 reset

Programmable Voltage Detection 1 and Programmable Voltage Detection 2 reset caused by the voltage monitoring circuit.

After Programmable Voltage Detection 1 is active and set as reset enable, if VCC is lower than the monitoring voltage of Programmable Voltage Detection 1, Programmable Voltage Detection 1 reset is generated and RMU_RSTF0.PVD1F is set. When the VCC voltage is higher than the monitored voltage of programmable voltage detection 1, the reset is released after the reset time of PVD1 (T_{IPVD1}).

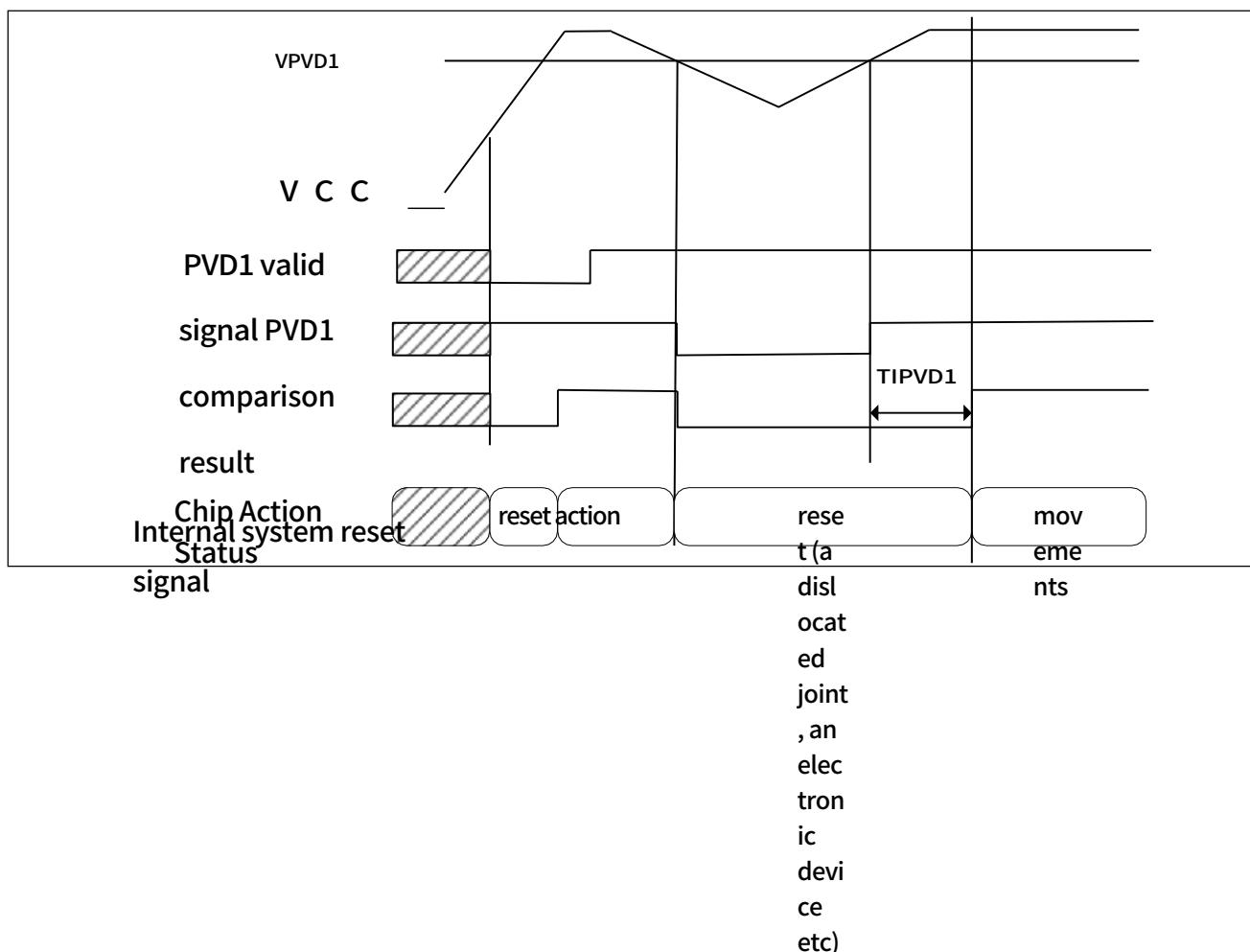


Figure 3-4 Programmable Voltage Detection 1 Reset

After Programmable Voltage Detection 2 is active and set as reset enable, if VCC is lower than the monitoring voltage of Programmable Voltage Detection 2, Programmable Voltage Detection 2 reset is generated and RMU_RSTF0.PVD2F is set. When the VCC voltage is higher than the monitored voltage of programmable voltage detection 2, the reset is released after the reset time of PVD2 (T_{IPVD2}).

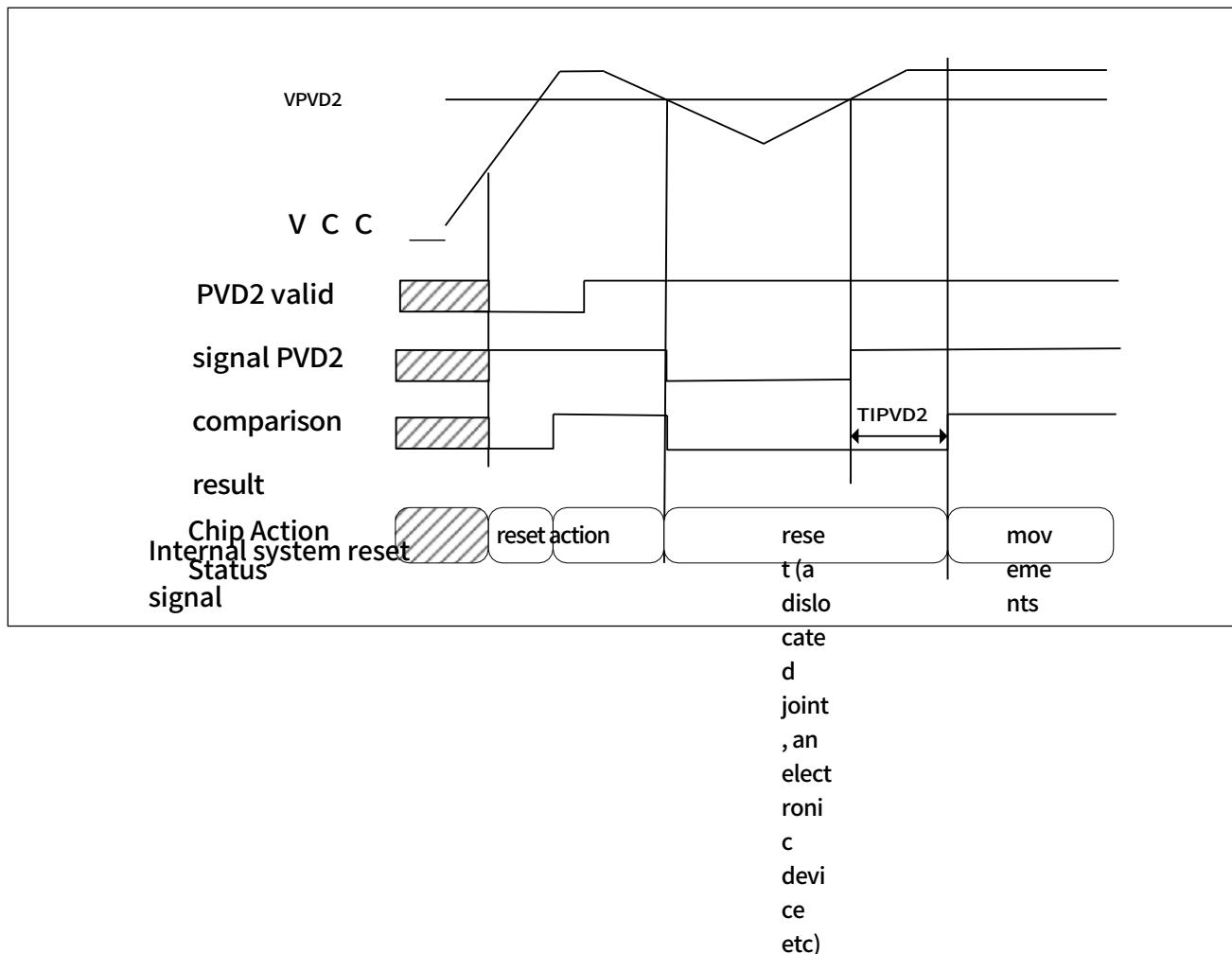


Figure 3-5 Programmable Voltage Detection 2 Reset

For the reset setting of Programmable Voltage Detection 1 and Programmable Voltage Detection 2, refer to [Power Supply Voltage Detection Unit (PWD) Description]

3.3.5 Watchdog Reset, Dedicated Watchdog Reset

Watchdog reset is an internal reset caused by the watchdog timer, and dedicated watchdog reset is an internal reset caused by the dedicated watchdog timer, and the reset timing is shown in Figure 3-6.

Setting the watchdog reset active generates a watchdog reset when the watchdog timer generates an underflow or a write operation is not performed during the refresh allow period. The 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} .

Setting the dedicated watchdog reset active generates a watchdog reset when the dedicated watchdog timer generates an underflow or a write operation is not performed during the refresh allow period. The dedicated watchdog reset sets RMU_RSTF0.SWDRF. After the dedicated watchdog reset is generated, the chip is released from reset after the internal reset time T_{RIPT} .

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

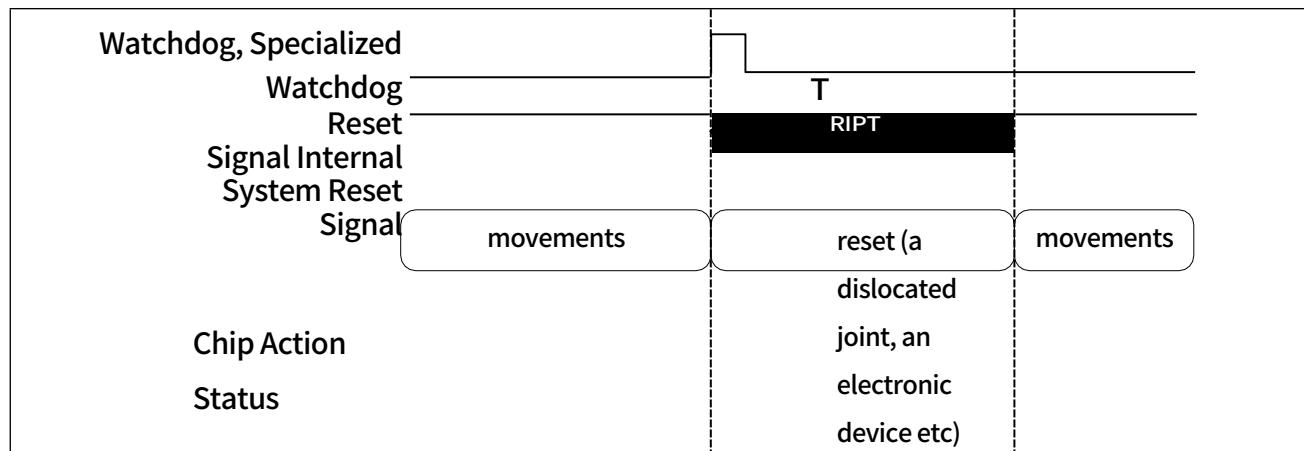


Figure 3-6 Watchdog and Dedicated Watchdog Reset

3.3.6 Wake on

power

down 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, and enters into the power-down mode, and then releases the power-down mode through the power-down mode wake-up event, the timing is shown in Fig. 3-7. After releasing the power-down mode and passing the return time (T_{IPDX} , $x=1,2,3,4$), the power-down wake-up reset is released. The return time varies according to the specific power-down mode set, and is the smallest in power-down mode 1 and the largest in power-down mode 3.

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

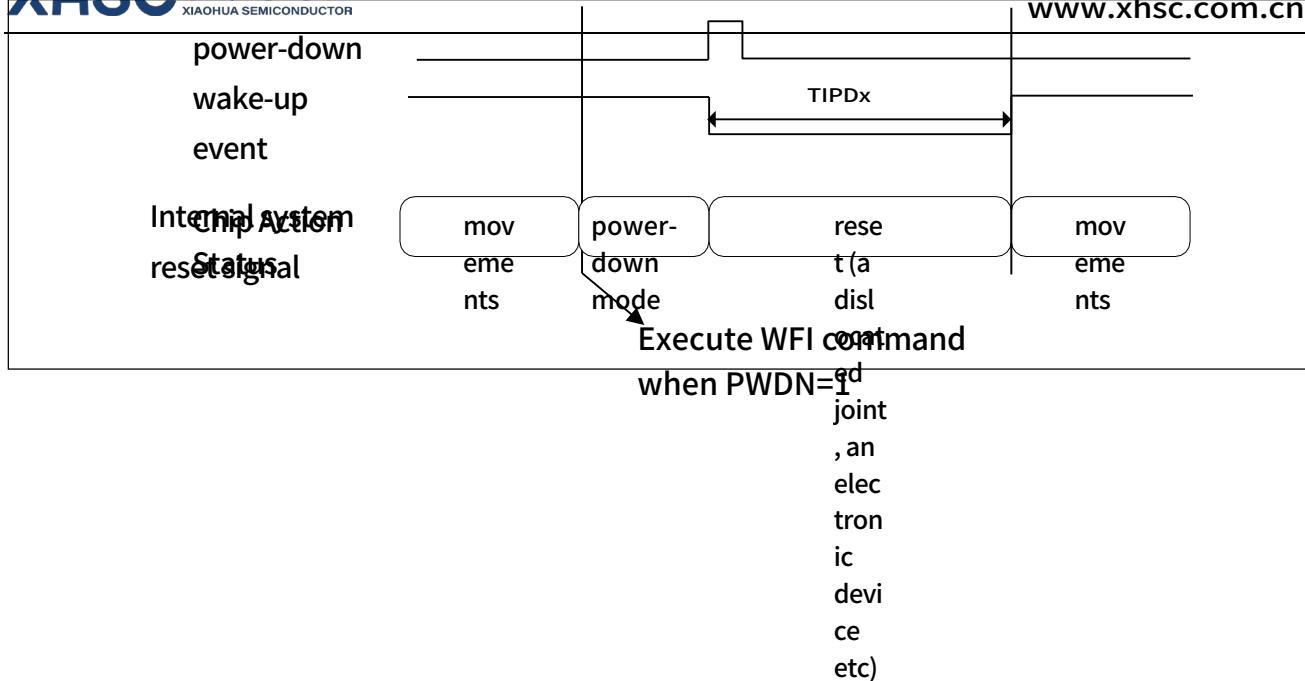


Figure 3-7 Power-down Wake-up Reset

3.3.7 software reset

A software reset is generated by writing the SYSRESETREQ bit of ARM register AIRCR. The RMU_RSTF0.SWRF bit is set when the software reset is generated. After the internal reset time TRIPT , the chip is released from reset.

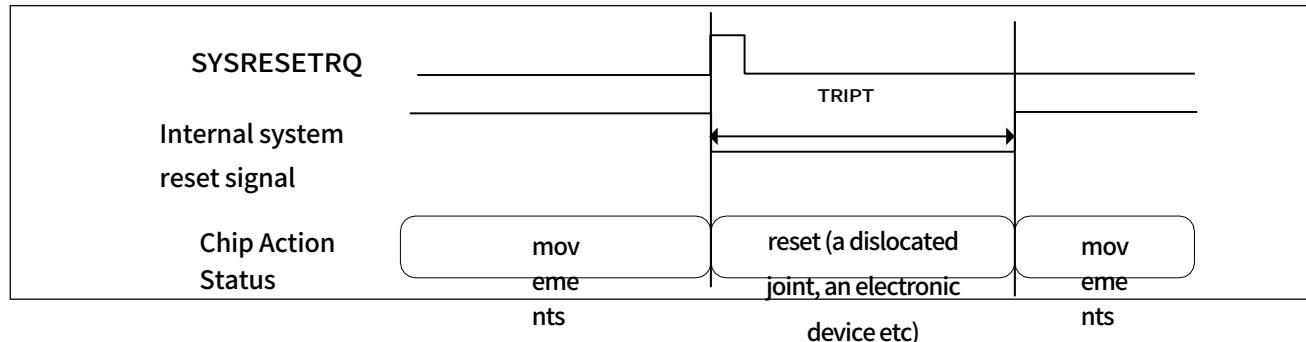


Figure 3-8 Software Reset

3.3.8 MPU Error

Reset

An MPU error reset sets RMU_RSTF0.MPUERF as shown in Figure 3-9. After the internal reset time TRIPT , the chip is released from reset.

Refer to [Memory Protection Unit (MPU)] for MPU error reset settings.

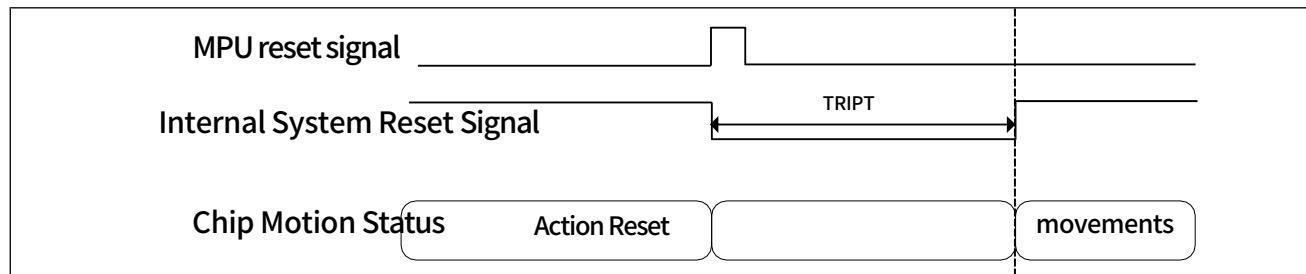
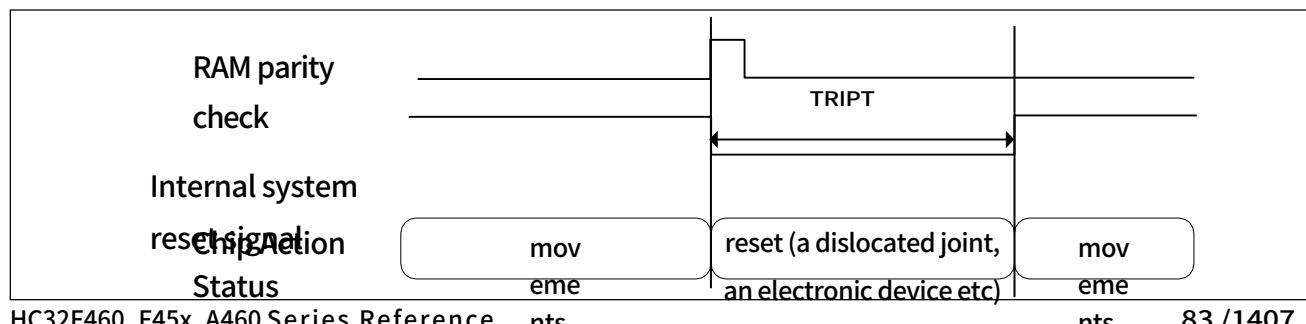


Figure 3-9 MPU Error Reset

3.3.9 RAM Parity Reset

When a RAM parity error occurs, a RAM parity reset is generated with the timing as shown in Figure 3-10. A RAM parity error sets RMU_RSTF0.RAPERF. After the internal reset time TRIPT , the chip is released from reset.

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



3.3.10 RAMECC reset

The RAMECC reset is generated when an error occurs in the RAMECC validation as shown in Figure

3-11. RAMECC reset sets

RMU_RSTF0.RAECRF. After the internal reset time TRIPT , the chip is unreset.

For the RAMECC reset setting, refer to [Built-in SRAM (SRAM)]

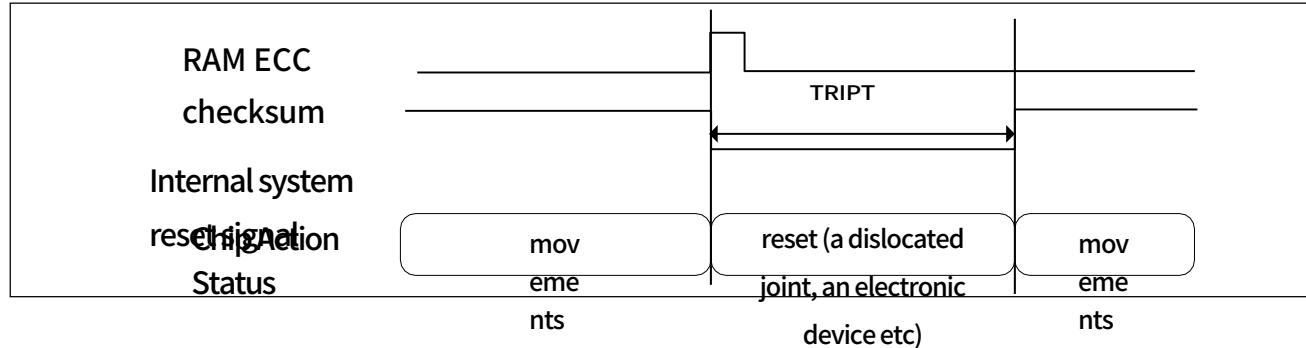


Figure 3-11 RAMECC Reset

3.3.11 Clock frequency abnormal reset

When the built-in FCM module monitors the clock frequency abnormality, it will generate a clock frequency abnormal reset if it is set to reset valid, the timing sequence is shown in Figure 3-12, the clock frequency abnormal reset will set RMU_RSTF0.CKFERF. CKFERF. After the internal reset time TRIPT , the chip is released from reset.

Refer to [Clock Frequency Measurement] for the setting of clock frequency abnormal reset.

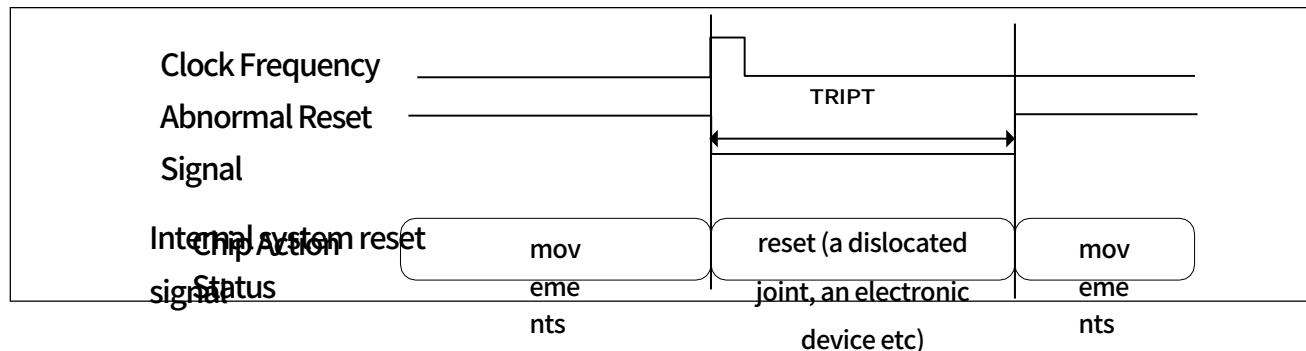


Figure 3-12 Clock Frequency Abnormal Reset

3.3.12 External high-speed oscillator abnormal stop reset

When the chip's oscillation stop detection module is valid and reset is enabled, if an abnormal external high-speed oscillator stop occurs, an abnormal external high-speed oscillator stop reset is generated and RMU_RSTF0.XTALERF is set. After the internal reset time TRIPT, the chip is released from reset.

Refer to [External High Speed Oscillator Failure Detection] for the setting of reset for abnormal stopping of the external high-speed oscillator.

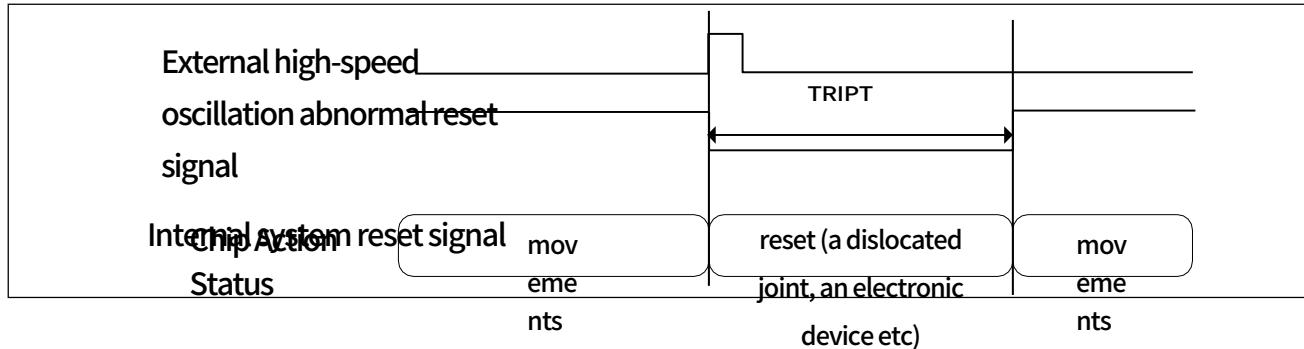


Figure 3-13 External High-Speed Oscillation Abnormal Reset

3.3.13 Judgment of reset mode

The reset mode can be judged based on the reset flag of RMU_RSTF0. When two or more resets are generated at the same time, multiple reset flags may be generated. A MULTIRF bit of 1 in RMU_RSTF0 indicates that multiple resets have occurred. After reading RMU_RSTF0, all reset flags can be cleared to 0 by setting the CLRF bit, and after setting RMU_RSTF0 to 0, it will take at least 6 CPU clock cycles before reading RMU_RSTF0 register again.

3.3.14 Reset conditions for each module

module (in software)	processor register	reset source
Commissioning Controller (DBG)	MCUSTPCTL MCUTRACECTL MCUDBGSTAT	1. Power-on reset 2. Wake on power down reset
Real Time Clock (RTC)	RTC internal registers	Module software reset control bit: RTC_CR0.RESET
Power Control (PWC) Clock Control (CMU)	pwc_pwrc0 pwc_pwrc1 pwc_pdwke0 pwc_pdwke1 pwc_pdwke2 pwc_pdwkes CMU_XTALCFG	All reset sources other than down memory reset: power-down mode 1 wake-up reset Power-down mode 2 Wake-up reset Power-down mode 4 Wake-up reset
	PWC_PDWKF0 PWC_PDWKF1	All reset sources other than power-down wake-up reset
	pwc_pvdlcr pwc_pvdcr1 pwc_pvdocr pwc_pvdcr0 pwc_pwrc2 PWC_PWCMR	1. Power-on reset 2. Pin Reset 3. Undervoltage reset 4. watchdog reset 5. Dedicated Watchdog Reset 6. Power-down mode 3 Wake-up reset
	pwc_pvdcir[0] pwc_pvddsr.pvd1detflg pwc_pvdcir[4] pwc_pvddsr.pvd2detflg	1. Power-on reset 2. Pin Reset 3. Undervoltage reset 4. watchdog reset 5. Dedicated Watchdog Reset 6. Wake on power down reset
	superscript	All reset sources
	Registers other than the above modules	All reset sources

All reset sources in the table refer to the 14 reset sources described in the introduction to this chapter.

3.4 Register Description

The registers are listed in

Table 3-3. BASE ADDR:

0x400540C0

Table 3-3 RMU Register List

register name	notation	offset address	bit width	reset value
Reset Status Register	RMU_RSTF0	0x00	16	Different reset values according to different reset methods

3.4.1 Reset flag register 0 (RMU_RSTF0)

Reset value: 0xXXXXXX (different reset values depending on the reset method)

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CLRF	MULTI RF	XTALE RF	CKFER F	RAECR F	RAPER F	MPUER F	SWRF	PDRF	SWDRF	WDRF	PVD2R F	PVD1R F	BORF	PINRF	PORF

Bit Flag		Bit Name		Function		Read/Write		Description								
b15	CLRF	Clear Reset Flag														Software set 1 to clear the reset flag bit.
																0 on read, the set action must be performed after reading RMU_RSTF0. 0: No operation
b14	MULTIRF	More than 2 reset occurrence flag bits														1: Zero reset flag Set by hardware when two or more resets occur. Cleared by setting CLRF. 0: Two or more resets did not occur <u>1: When two or more resets occur</u>
b13	XTALERF	External high-speed oscillator abnormal stop reset flag														Set by hardware when an abnormal external high-speed oscillator stop reset occurs. Cleared by setting CLRF. 0: No external high-speed oscillator abnormal stop reset occurred <u>1: Occurrence of external high-speed oscillator abnormal stop reset The occurrence of clock frequency</u>
b12	CKFERF	Clock frequency exception reset flag														abnormal reset is set by hardware. It is cleared by setting CLRF. 0: No clock frequency abnormality reset occurred <u>1: Abnormal clock frequency reset occurs</u>
b11	RAECRF	RAMECC reset symbolize														Set by hardware when a RAMECC reset occurs. Cleared by setting CLRF. 0: RAMECC reset did not occur
		RAM parity error														1: RAMECC reset occurs Set by hardware when a RAM parity error reset occurs. Cleared by setting CLRF.

Set by hardware when an MPU error
reset occurs.

Cleared by setting
CLRF. 0: No MPU error
reset occurred

R/W

b9 MPUERF MPU error reset
flag

1: MPU error reset
occurred

Set by hardware when a software reset occurs.				
b8	SWRF	Software reset flag	Software reset occurs. Cleared by setting CLRF. 0: No software reset occurred 1: Software reset occurs	R/W
b7	PDRF	Power-down mode reset	Set by hardware when a power-down mode reset occurs. Cleared by writing to CLRF. 0: No power-down mode reset occurred 1: Power-down mode reset occurs	R/W
b6	SWDRF	Specialized Watchdog Reset Flag	Set by hardware when a dedicated watchdog reset occurs. Cleared by writing to CLRF. 0: Dedicated watchdog reset did not occur 1: Dedicated watchdog reset occurs	R/W
b5	WDRF	Watchdog reset flag	Set by hardware when a watchdog reset occurs. Cleared by writing to CLRF. 0: Watchdog reset did not occur 1: Watchdog reset occurs	R/W
b4	PVD2RF	Programmable voltage detect 2 reset flag	Programmable Voltage Detection 2 reset occurs when it is set by hardware. Cleared by writing to CLRF. 0: Programmable voltage detection 2 reset did not occur 1: Programmable voltage detection 2	R/W
b3	PVD1RF	Programmable Voltage Detection 1 Reset Flag	reset occurs Programmable Voltage Detection 1 reset occurs when it is set by hardware. It is cleared by writing to CLRF. 0: Programmable voltage detection not occurring 1 Reset 1: Programmable Voltage	R/W
b2	BORF	Undervoltage	Reset Flag	Detection occurs 1 Reset is set by hardware when an undervoltage reset occurs. 0: Programmable voltage detection not occurring 1 Reset 1: Programmable Voltage
b1	PINRF	NRST pin reset flag	Cleared by writing CLRF. 0: No undervoltage reset occurred 1: Undervoltage reset occurs	R/W
b0	PORF	Power-on reset flag	Set by hardware when a pin reset occurs. Cleared by writing to CLRF. 0: No NRST reset occurred 1: NRST reset occurs	R/W
			Set by hardware when a power-on reset occurs. Cleared by writing to CLRF. 0: Power-on reset did not occur 1: Power-on reset occurs	

4 Clock Controller (CMU)

4.1 brief introduction

The Clock Control Unit provides clocking functions for a range of frequencies, including: an external high-speed oscillator, an external low-speed oscillator, two PLL clocks, an internal high-speed oscillator, an internal medium-speed oscillator, an internal low-speed oscillator, an internal low-speed oscillator dedicated to the SWDT, clock prescaler, clock multiplexing, and clock gating circuitry.

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

The AHB, APB, and Cortex-M4 clocks are derived from the system clock, which

can be sourced from a choice of six clock sources: 1) external high-speed oscillator (XTAL)

- 2) External low-speed oscillator (XTAL32)
- 3) MPLL Clock (MPLL)
- 4) Internal High Speed Oscillator (HRC)
- 5) Internal medium-rate oscillator (MRC)
- 6) Internal low rate oscillator (LRC)

The system clock can run at a maximum clock frequency of 200MHz. the SWDT has a separate clock source: the 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 clock source. The Real Time Clock (RTC) uses either an external low-speed oscillator or an internal low-speed oscillator as the clock source.The 48MHz clock of the USB-FS can be selected from the system clock, MPLL, and UPLL as the clock source.

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

4.2 system block diagram

4.2.1 system block diagram

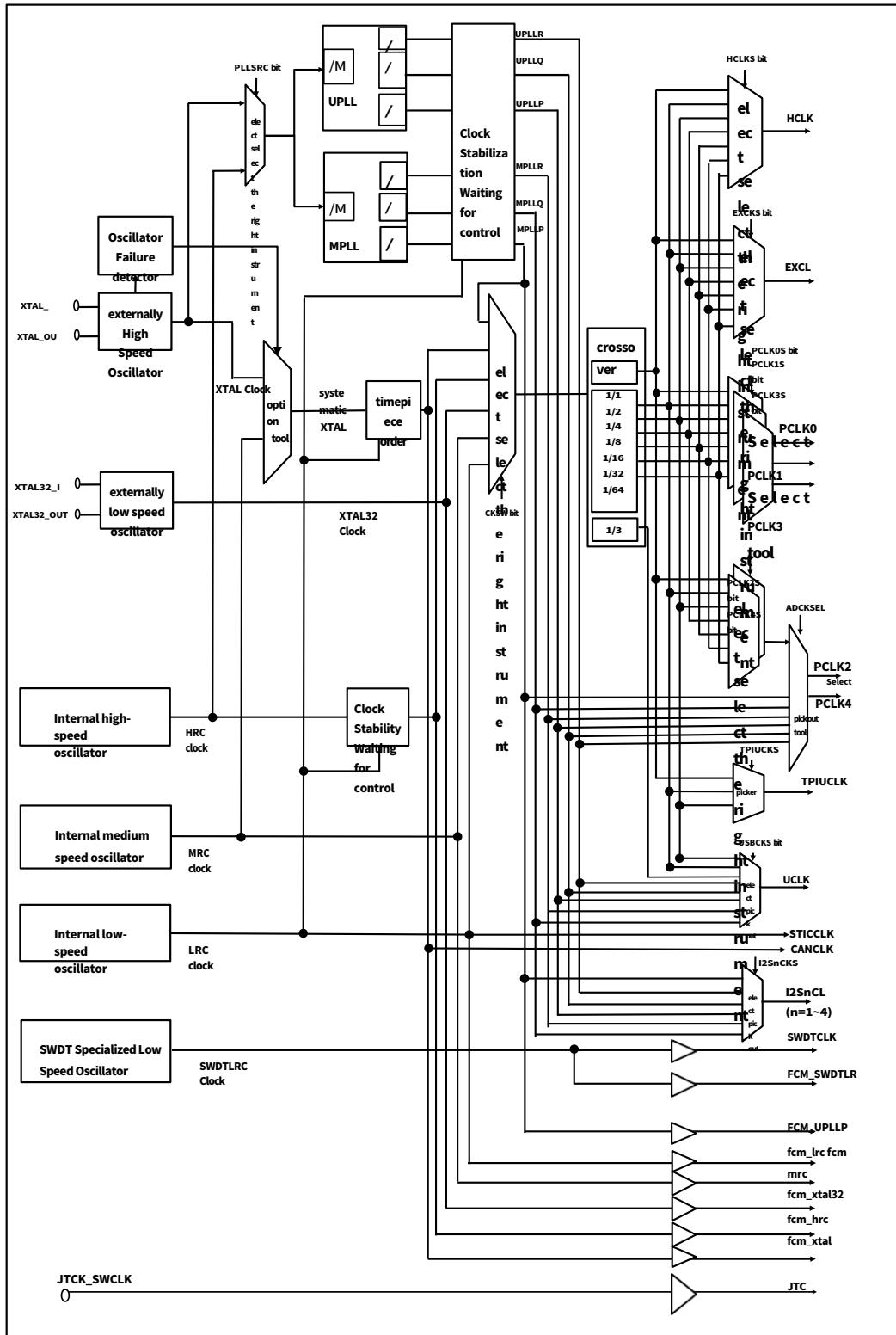


Figure 4-1 Clock System Block Diagram

4.2.2 Clock Frequency Measurement Block Diagram

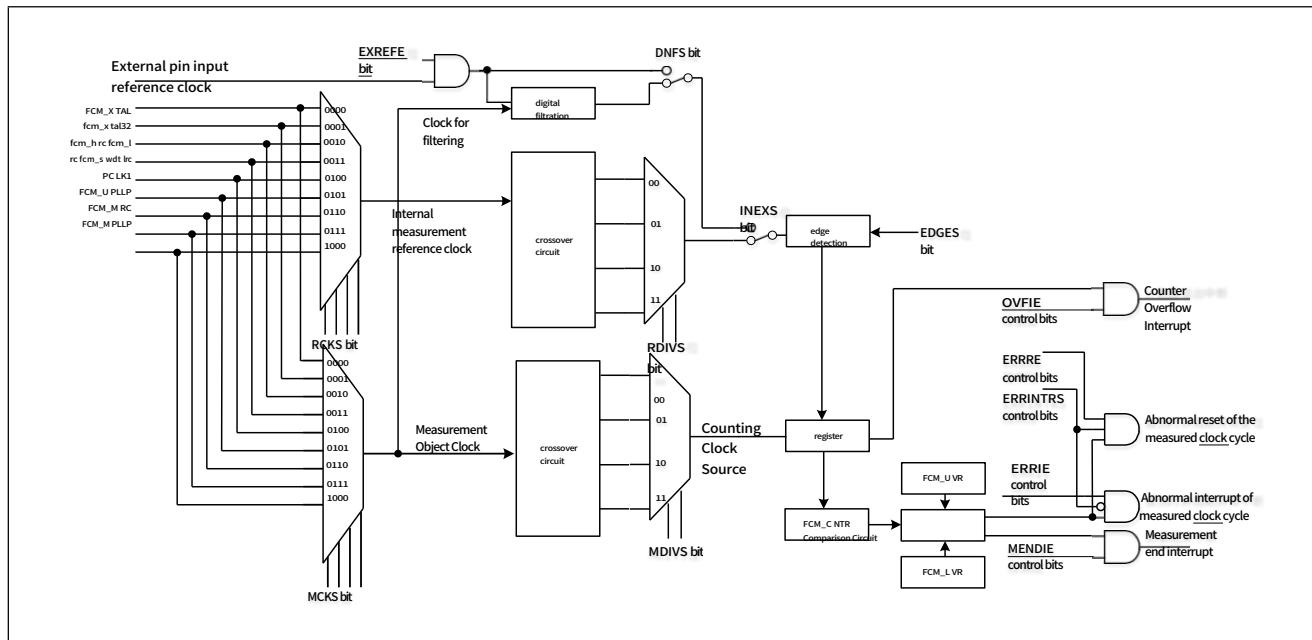


Figure 4-2 Clock Frequency Measurement Block Diagram

4.3 Clock Source Specifications

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

clock source	norm
External High Speed Oscillator (XTAL)	Frequency range of crystal: 4~25MHz External clock input: up to 25MHz Oscillator fault detection function
External low-speed oscillator (XTAL32)	Frequency of crystal: 32.768KHz
MPLL Clock (MPLL)	Input clock: external high-speed oscillator or internal high-speed oscillator MPLL Input clock division frequency: 1~24 arbitrary division frequency selectable PFD Input Frequency = Input Clock/MPLL Input Clock Division, Frequency Range 1MHz~25MHz MPLL Octave Factor: 20~480 times VCO Oscillation frequency: 240MHz~480MHz MPLLQ Output crossover ratio: 2~16 arbitrary crossover frequency MPLLP Output crossover ratio: 2~16 arbitrary crossover frequency MPLL Output crossover ratio: 2~16 arbitrary crossover frequency MPLL Output Frequency = (Input Clock/MPLL Input Clock Divider) * MPLL Octave Factor /MPLL Output Crossover Ratio MPLLQ Output Frequency = (Input Clock/MPLL Input Clock Divider) * MPLL Octave Factor /MPLLQ Output Crossover Ratio MPLL Output Frequency = (Input Clock/MPLL Input Clock Divider) * MPLL Octave Factor /MPLL Output Crossover Ratio

UPLL Clock (UPLL)	<p>Input clock: external high-speed oscillator or internal high-speed oscillator UPLL Input</p> <p>clock division frequency: 1~24 arbitrary</p> <p>division frequency selectable</p> <p>PFD Input Frequency = Input Clock/UPLL Input Clock Divider,</p> <p>Frequency Range 1MHz~25MHz UPLL Octave Factor: 20~480 times</p> <p>VCO Oscillation Frequency:</p> <p>240MHz~480MHz UPLLP Output</p> <p>Frequency Division Ratio: 2~16</p> <p>Arbitrary Division Frequency</p> <p>UPLLQ Output Frequency Division</p> <p>Ratio: 2~16 Arbitrary Division</p> <p>Frequency UPLL R Output</p> <p>Frequency Division Ratio: 2~16</p> <p>Arbitrary Division Frequency</p> <p>UPLLP Output Frequency = (Input Clock/UPLL Input Clock Divider)</p> <p>* UPLL Octave Factor</p> <p>/UPLLP Output Crossover Ratio</p> <p>UPLLQ Output Frequency = (Input Clock/UPLL Input Clock Divider)</p> <p>* UPLL Octave Factor</p> <p>/UPLLQ Output Crossover Ratio</p> <p>UPLL R Output Frequency = (Input Clock/UPLL Input Clock Divider)</p> <p>* UPLL Octave Factor</p> <p>/UPLL R Output Crossover Ratio</p>
Internal High Speed Oscillator (HRC)	<p>Frequency: 16MHz or 20MHz</p> <p>User-writable registers to fine-tune the frequency</p>
Internal medium-rate oscillator (MRC)	<p>Frequency: 8MHz</p> <p>User-writable registers to fine-tune the frequency</p>

clock source	norm
Internal low rate oscillator (LRC)	Frequency: 32.768KHz User-writable registers to fine-tune the frequency It can be used as a count clock for RTC, a count clock for wake-up timer WKTM, a standby clock for XTAL32, a count clock for WKTM, a standby clock for XTAL32, and a count clock for XTAL32. quarterly clock
SWDT Specialized Internal Low Speed Oscillator (SWDTRC)	Frequency: 10KHz

4.4 Operating

Clock

Specification

ns

Table 4-1 Specifications of Each Internal Clock

clocks	scope of action	norm
HCLK	CPU, DMAn (n=1, 2), EFM (main flash), SRAM1, SRAM2, SRAM3, SRAMH, Ret-SRAM, MPU, GPIO, DCU, INTC, QSPI	Maximum frequency 200MHz Optional clock source divider: 1, 2, 4, 8, 16, 32, 64
PCLK0	Timer6 Counter Clock	Maximum frequency 200MHz Optional clock source divider: 1, 2, 4, 8, 16, 32, 64
PCLK1	USARTn(n=1~4), SPI(n=1~4), USBFS (control logic) Timer0n(n=1, 2), TimerAn(n=1~6), Timer4n(n=1~3), Timer6 (control logic) EMB, CRC, HASH, AES, I2Sn(n=1~4) control system logic	Maximum frequency 100MHz Optional clock source divider: 1, 2, 4, 8, 16, 32, 64
PCLK2	ADC Transform Clock	Maximum frequency 60MHz Selectable clock source divisions: 1, 2, 4, 8, 16, 32, 64 Independently selectable clock sources: UPLL, UPLLQ, UPLL, MPLL, MPLLQ. MPLL
PCLK3	RTC (control logic) I2Cn (n=1, 2, 3), CMP, WDT, SWDT (control logic)	Maximum frequency 50MHz Optional clock source divider: 1, 2, 4, 8, 16, 32, 64
PCLK4	ADC (control logic) TRNG	Maximum frequency 100MHz Selectable clock source divisions: 1, 2, 4, 8, 16, 32, 64 Independently selectable clock sources: UPLL, UPLLQ, UPLL, MPLL, MPLLQ. MPLL
EXCLK	SDION (n=1, 2), CAN	Maximum frequency 100MHz Optional clock source divider: 1, 2, 4, 8, 16, 32, 64
UCLK	USBFS Communication Clock	Frequency 48MHz Clock source optional system clock divider 2, 3, 4. independently selectable clock source. uplp, uplq, uplr, mppll, mppllq. MPLL
CANCLK	CAN Communication Clock	Frequency range 4~25MHz



XHSC 小华半导体

STICCLK	Clock for CPU's SysTickTimer counter, Clock Source is LRC	www.xhsc.com.cn Configurable as clock source LRC or system clock
SWDTCLK	SWDT Counter Clock	Frequency 10KHz
JTCK	Clock for JTAG	Maximum frequency 25MHz
TPIUCLK	Cortex-M4 Clock for Debug Tracker	Maximum frequency 50MHz

clocks	range of action	norm
		Optional clock source divider: 1, 2, 4
I2SnCLK (n=1~4)	I2Sn (n=1~4)	Maximum frequency 200MHz Individually selectable clock sources. uplp, uplq, uplr, mppll, mpllq. MPLL R

Attention:

The following rules need to be observed between clocks:

- HCLK frequency >= PCLK1 frequency, HCLK frequency >= PCLK3 frequency, HCLK frequency >= PCLK4 frequency
- HCLK frequency:EXCLK frequency = 2:1,4:1,8:1,16:1,32:1
- PCLK0 frequency >= PCLK1 frequency, PCLK0 frequency >= PCLK3 frequency
- HCLK Frequency:PCLK0 Frequency=N:1,1:N
- PCLK2 Frequency:PCLK4 Frequency = 1:4,1:2,1:1,2:1,4:1,8:1

4.5 crystal oscillator circuit

4.5.1 External High Speed Oscillator

4.5.1.1 oscillator mode

An external high-speed oscillator provides a more accurate clock source for the system clock. The frequency range is 4~25MHz. XTAL is turned on and off by the XTALSTP bit of CMU_XTALCR.

The XTALSTBF flag bit of CMU_OSCSTBSR indicates whether the external high-speed oscillator is stabilized or not, and the stabilization time is configured through register CMU_XTALSTBCR. The stabilization time set by CMU_XTALSTBCR must be greater than or equal to the stabilization time required by the crystal manufacturer.

The circuit constants of the crystal oscillator vary depending on the parasitic capacitance of the crystal and the mounting circuit, and therefore must be decided after careful discussion with the crystal manufacturer. Various characteristics of the oscillator are closely related to the user's board design. The crystal and load capacitance must be placed as close as possible to the oscillator pins to minimize output distortion and start-up stabilization time. The load capacitance value must be adjusted according to the selected oscillator. Signal lines must not be passed near the oscillator circuit or it may not oscillate properly due to inductance.

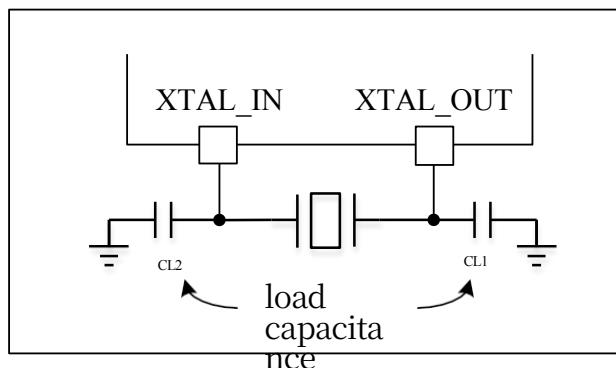


Figure 4-3 External High-Speed Oscillator Connection Example

4.5.1.2 Clock Input Mode

In Clock Input mode, an external clock source must be provided. This mode is selected via XTALMS position 1 of CMU_XTALCFG and XTALSTP position 0 of CMU_XTALCR. An external clock signal with a duty cycle of approximately 50% must be used to drive the PH0/XTAL_EXT/XTAL_OUT pin. At this point the PH1/XTAL_IN pin can be configured as GPIO based on the register settings.

An example of the connections for the external clock input is shown in the following figure.

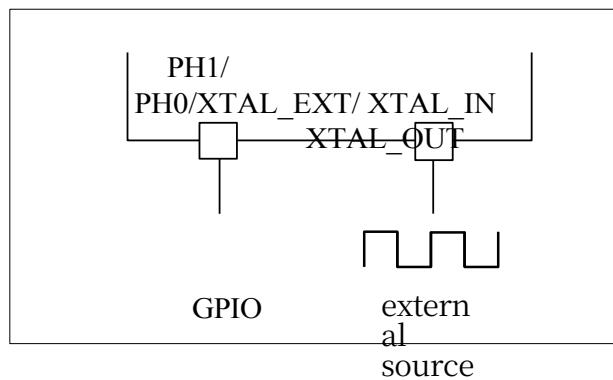


Figure 4-4 Connection Example Diagram for External Clock Inputs

4.5.2 External high-speed oscillator fault detection

The Oscillator Fault Detection detects if the external high-speed oscillator (XTAL) is oscillating properly. It is turned on or off by the XTALSTDE bit of register CMU_XTALSTDCR.

When the reset is released, 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 fault detection function, the external high-speed oscillator must be oscillated, and the external high-speed oscillator must be stabilized until the external high-speed oscillator is stabilized, i.e., CMU_OSCSTBSR.XTALSTBF is set to 1, and is turned on by the XTALSTDE bit of register CMU_XTALSTDCR.

When MPLL and UPLL select the XTAL clock as the input source, only the XTAL oscillation fault generation reset function can be selected.

Because oscillator fault detection detects abnormal oscillation of the oscillator caused by external factors, the oscillator oscillation fault detection function should be invalidated when the external high-speed oscillator stops oscillating or is transferred to the stop mode and power-down mode through software.

If an external high-speed oscillator fails, the action waveform is shown below. Refer to [XTAL Failure Detection Action Detected] for the operation procedure.

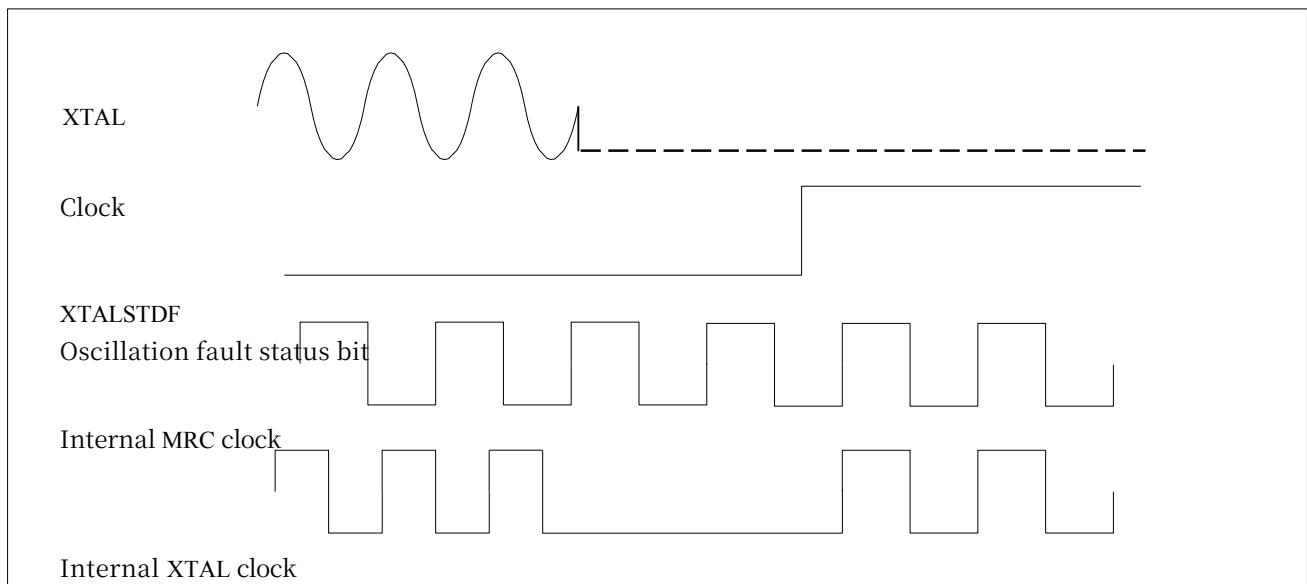


Figure 4-5 External High-Speed Oscillator Failure Detection Example

4.5.2.1 XTAL fault detection action detected

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

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

The system clock is selected as XTAL, and when an XTAL fault is detected, the action example is shown in the following figure.

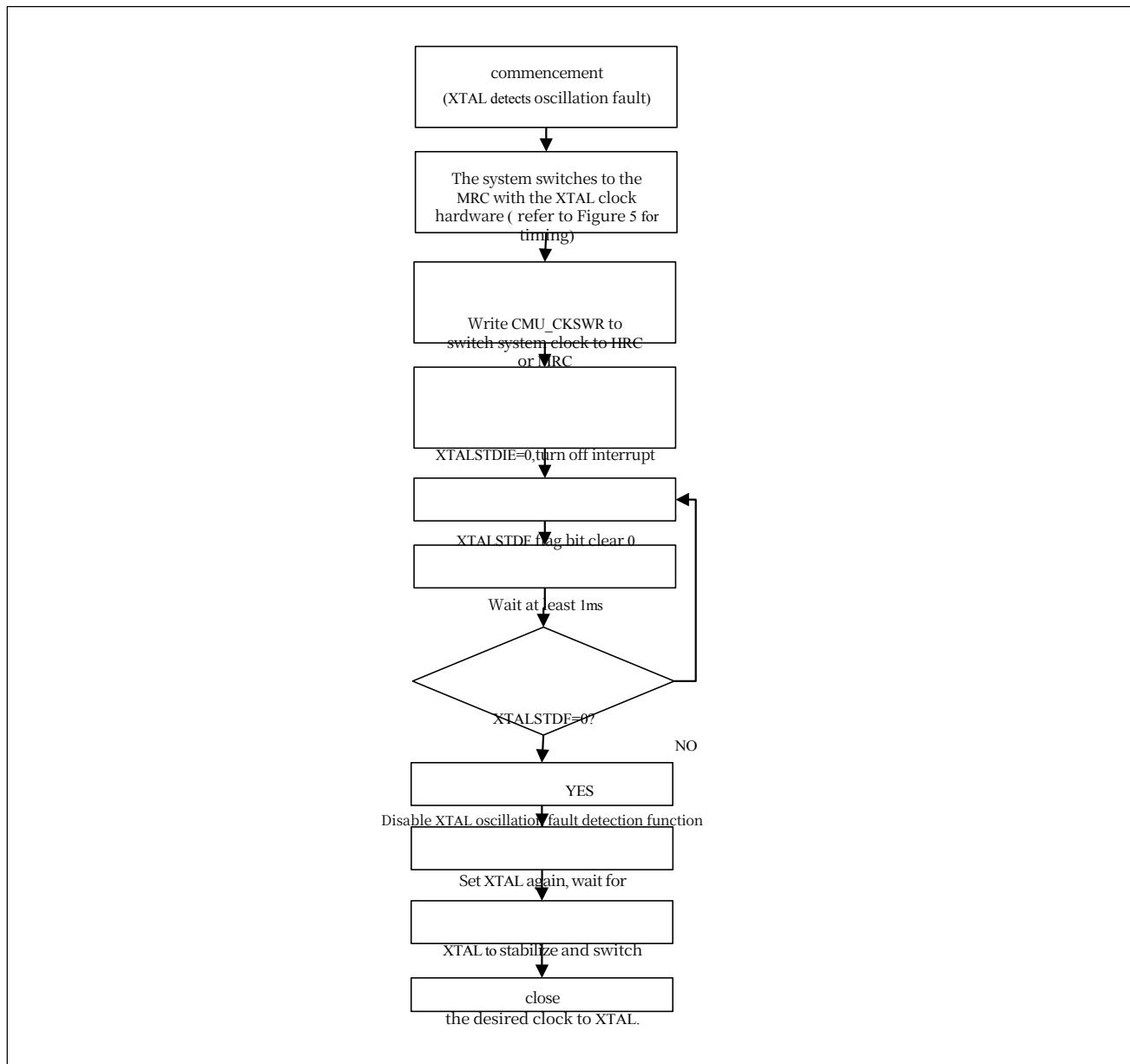


Figure 4-6 System Clock Selection XTAL, XTAL Oscillation Fault Detected Example

4.5.2.2 XTAL oscillation fault detected to generate interrupt reset

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

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

4.5.3 External low-speed oscillator

The 32.768KHz external low-speed oscillator provides a more accurate clock source for system clocks and real-time clock circuits (RTCs). It has the advantages of low power consumption and high accuracy.

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

The circuit constants of the crystal oscillator vary depending on the parasitic capacitance of the crystal and the mounting circuit, and therefore must be decided after careful discussion with the crystal manufacturer. Various characteristics of the oscillator are closely related to the user's board design. The crystal and load capacitance must be placed as close as possible to the oscillator pins to minimize output distortion and start-up stabilization time. The load capacitance value must be adjusted according to the selected drive capability. Signal lines must not be passed near the oscillator circuit or it may not oscillate properly due to inductance.

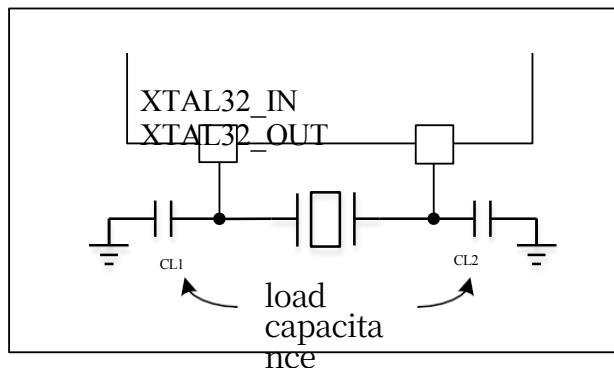


Figure 4-7 Example of External Low Speed Oscillator Connection

The initialization flow for the initial power-up of the XTAL32 is shown below:

1. CMU_XTAL32CR.XTAL32STP Bit Write 1 to stop XTAL32
2. Set matching XTAL32 driver capability via CMU_XTAL32FGR
3. Setting of the filter function via CMU_XTAL32FGR
4. CMU_XTAL32CR.XTAL32STP Bit Write 0, XTAL32 Oscillation
5. The software waits for the XTAL32 to stabilize, refer to the **Electrical Characteristics** section for stabilization times.

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 MPLL/UPLL input. The frequency of HRC can be configured as 16MHz or 20MHz by ICG1.

HRC oscillators have the advantage of being less expensive (no external components are required) They are also faster to start up than XTAL blocks, but are not as accurate as external crystals, even when calibrated.

Frequency Calibration

Because the RC oscillator frequency varies from chip to chip due to different manufacturing processes, each device is factory calibrated to ensure accuracy refer to the **Internal High Speed (HRC) Oscillator** chapter **in the Electrical Characteristics section of the datasheet**.

If the application is affected by temperature variations, this may also affect the speed of the RC oscillator. The user can fine tune the HRC frequency through the CMU_HRCTRM register.

The HRCSTBF flag in CMU_OSCSTBSR indicates whether the HRC is stable. At startup, the hardware sets this to position 1 before the HRC can be used.

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

4.6.2 MRC Clock

The MRC clock signal is generated by an internal 8MHz medium speed oscillator and can be used directly as the system clock. The advantage of the MRC oscillator is that it is fast to start up and can be used without waiting for stabilization. **Frequency**

Calibration

Because the RC oscillator frequency varies from chip to chip due to different manufacturing processes, each device is factory calibrated to ensure accuracy refer to the **Internal Medium Rate (MRC) Oscillator** section **of the datasheet Electrical Characteristics**.

If the application is affected by temperature variations, this may also affect the speed of the RC oscillator. The user can fine tune the MRC frequency through the CMU_MRCTRM register.

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

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

4.6.3 LRC Clock

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

LRC oscillators are fast to start up and can be used without waiting for stabilization after startup.

Frequency Calibration

Because the RC oscillator frequency varies from chip to chip due to different manufacturing processes, each device is factory calibrated to ensure accuracy refer to the **Internal Low Rate (LRC) Oscillator section in the Electrical Characteristics section of the datasheet**.

If the application is affected by voltage or temperature variations, this may also affect the speed of the RC oscillator. The user can fine tune the LRC frequency through the CMU_LRCTR register.

LRC can be turned on or off by the LRCSTP bit in the CMU_LRCCR control register.

4.6.4 SWDTRC Clock

The SWDTRC clock signal is generated by the internal 10KHz low-speed oscillator, the SWDT dedicated clock. the SWDT has been activated by means of the ICG setting, then the SWDT dedicated internal low-speed oscillator will be forced on and cannot be disabled.

Because the RC oscillator frequency varies from chip to chip due to different manufacturing processes, each device is factory calibrated to ensure accuracy Refer to the **SWDT Specialized Internal Low Speed (SWDTLRC) Oscillator section of the datasheet Electrical Characteristics**.

4.7 PLL Clock

The HC32F46xx device has two PLLs:

- The MPLL is clocked by an XTAL or HRC oscillator and has three different output clocks:
 - P divider output for generating system clock (up to 200 MHz)
 - All three outputs can be used to generate USBFS, TRNG, ADC and I2S clocks.
- The three UPLL outputs can also be used to generate USBFS, TRNG, ADC and I2S clocks.
The UPLL uses the same input clock source as the MPLL and can select either the HRC or XTAL oscillator as the clock source, as configured by the CMU_PLLCFGR.PLLSRC bit. The PLL is configured after the HRC or XTAL oscillator has stabilized.

The crossover coefficients M, N, P, Q, and R of MPLL/UPLL 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.

Both PLLs will be disabled by hardware when entering power-down and stop mode.

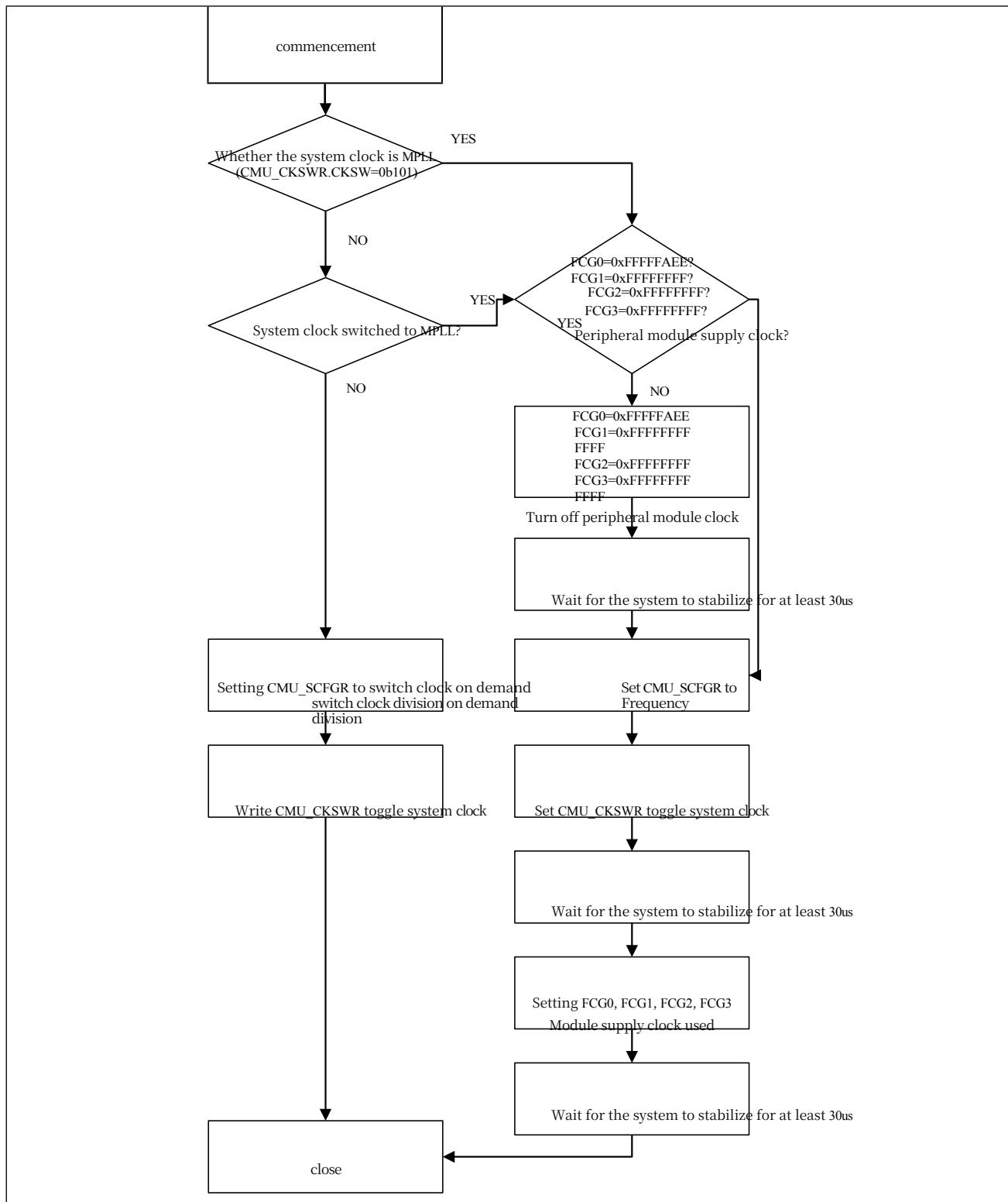
4.8 Clock switching procedure

After a system reset, the default system clock is MRC. switch the clock source by setting register CMU_CKSW, refer to Clock Source Switching for the switching procedure. Switching

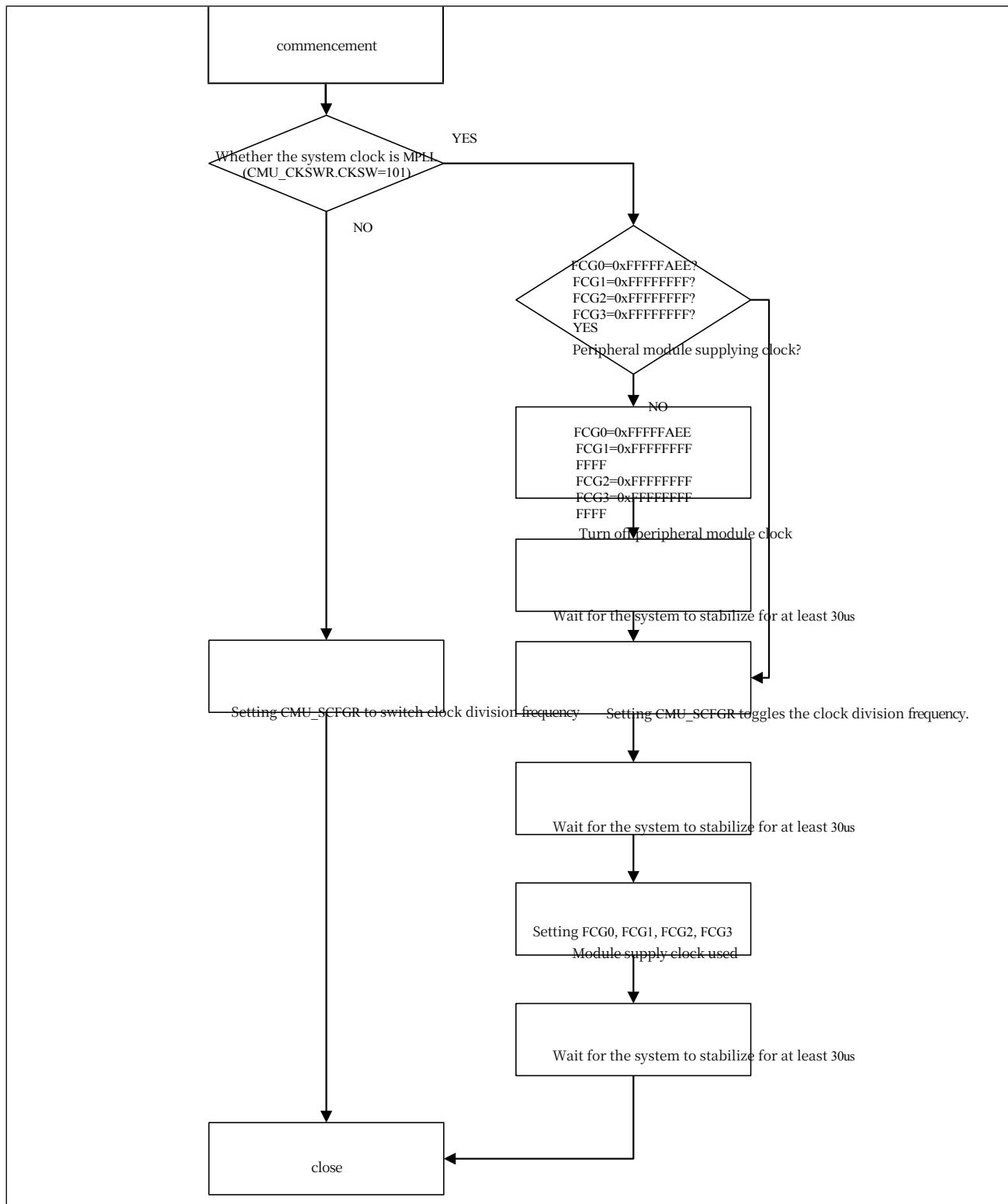
from one clock source to another is only possible if the target clock source has stabilized.

When switching the clock, it is necessary to correctly configure the wait period of FLASH/SRAM to prevent the system clock frequency from being greater than the maximum operation frequency of FLASH/SRAM. Refer to [Relationship between CPU clock and FLASH read time] Built-in SRAM (SRAM) configuration.

4.8.1 Clock source switching



4.8.2 Clock division switching



4.9 Clock output function

There are two clock outputs:

■ MCO_1

The user can output different clock sources to the MCO_1 pin through a configurable pre-distributor (from 1 to 128):

- HRC Clock
- MRC Clock
- LRC Clock
- XTAL Clock
- XTAL32 Clock
- MPLLP/MPLLQ Clock
- UPLLQ/UPLLQ Clock
- system clock

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

■ MCO_2

The user can output different clock sources to the MCO_2 pin through configurable pre-distributors (from 1 to 128):

- HRC Clock
- MRC Clock
- LRC Clock
- XTAL Clock
- XTAL32 Clock
- MPLLP/MPLLQ Clock
- UPLLQ/UPLLQ Clock
- system clock

The desired clock source is selected with 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

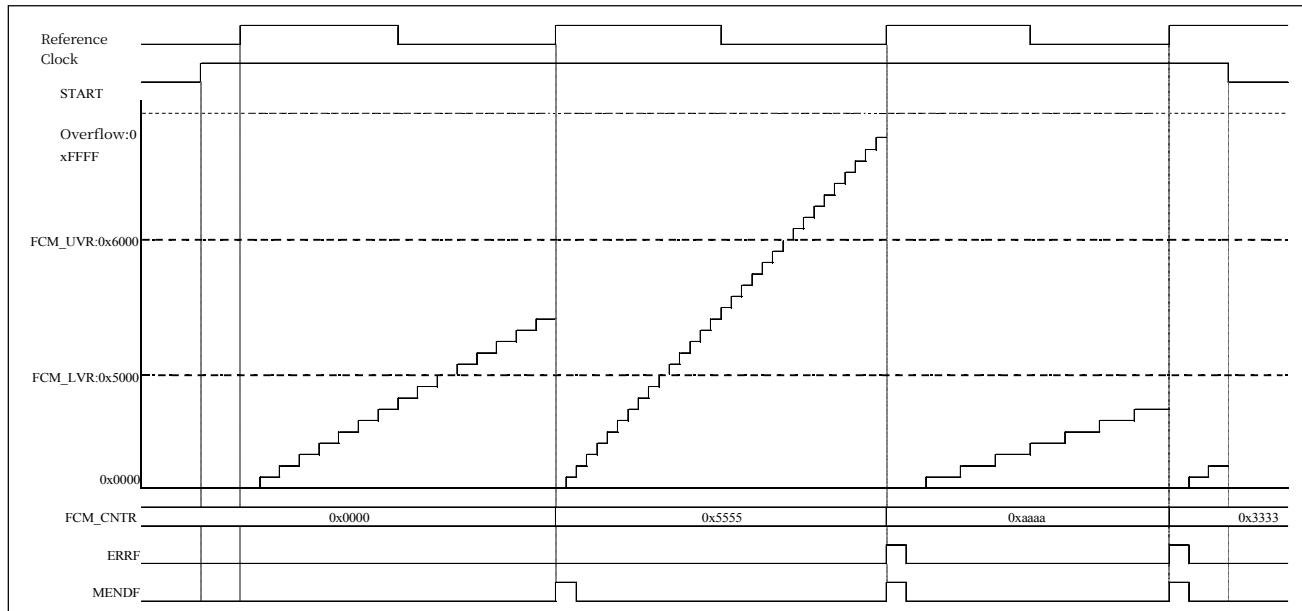


Figure 4-8 Clock Frequency Measurement Timing Chart

1. Use FCM_MCCR/FCM_RCCR to select the reference clock to be measured, the division frequency of the clock, and the effective edge of the selected reference clock.
2. After writing a 1 to the START bit of FCM_STR, a valid edge selected by the EDGES bit is detected, and the counter starts to count incrementally.
3. Upon detecting a valid edge selected by the next EDGES bit of the reference clock, the counter value 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 is measured normally. 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 setting.
4. After writing 0 to the START bit of FCM_STR, the counter stops and clears to zero.

4.10.2 Digital Filtering Function

The external pin input reference clock FCMREF has a digital filter function. The digital filter function takes three samples according to the sample clock selected by the DNFS bit, and when the levels of the three samples are the same, the level is sent to the internal.

The digital filter function allows you to set the digital filter function to be valid or invalid and the sampling clock.

4.10.3 Interrupt/reset function

The clock frequency measurement circuit

has three types of interrupt requests.

- They are:
 - 1) Frequency exception interrupt
 - 2) Frequency measurement end interrupt
 - 3) Counter Overflow Interrupt

The clock frequency measurement circuit has a kind of reset request:

- 1) Frequency Abnormal Reset

4.11 Register Description

Base address 1: 0x4004_8400

register name	notation	offset address	bit width	reset value
FCM lower limit comparison value register	FCM_LVR	0x00	32	0x0000_0000
FCM Upper Limit Comparison Value Register	FCM_UVR	0x04	32	0x0000_0000
FCM Counter Value Register	FCM_CNTR	0x08	32	0x0000_0000
FCM Start Stop Register	FCM_STR	0x0C	32	0x0000_0000
FCM Measurement Object Control Register	FCM_MCCR	0x10	32	0x0000_0000
FCM Measurement Reference Control Register	FCM_RCCR	0x14	32	0x0000_0000
FCM Interrupt Reset Control Register	FCM_RIER	0x18	32	0x0000_0000
FCM Flag Register	FCM_SR	0x1C	32	0x0000_0000
FCM Flag Bit Clear Register	FCM_CLR	0x20	32	0x0000_0000

Base address 2: 0x40054000

register name	notation	offset address	bit width	reset value
CMU_XTAL Configuration Registers	CMU_XTALCFGR	0x410	8	0x80
CMU_XTAL Valium Configuration Registers	CMU_XTALSTBCR	0x0A2	8	0x05
CMU_XTAL control registers	CMU_XTALCR	0x032	8	0x01
CMU_XTAL Oscillation Fault Control Register	CMU_XTALSTDCR	0x040	8	0x00
CMU_XTAL Oscillation Fault Status Register	CMU_XTALSTDSR	0x041	8	0x00
CMU_HRC Calibration Registers	CMU_HRCTRM	0x062	8	0x00
CMU_HRC Control Register	CMU_HRCCR	0x036	8	Determined by ICG1.HRCSTP value
CMU_MRC Calibration Registers	CMU_MRCTRM	0x061	8	0x00
CMU_MRC control registers	CMU_MRCCR	0x038	8	0x80
CMU_MPLL Configuration Registers	CMU_PLLCFGR	0x100	32	0x1110_1300
CMU_MPLL control registers	CMU_PLLCR	0x02A	8	0x01
CMU_UPLL Configuration Registers	CMU_UPLLCFGR	0x104	32	0x1110_1300
CMU_UPLL control registers	CMU_UPLLCR	0x02E	8	0x01
CMU_Clock Source Stabilization Status Register	CMU_OSCSTBSR	0x03C	8	0x00
CMU_System Clock Source Switching Register	CMU_CKSWR	0x026	8	0x01
CMU_Clock Division Configuration Registers	CMU_SCFG	0x020	32	0x0000_0000
CMU_USBFS Clock Configuration Registers	CMU_UFSCKCFGR	0x024	8	0x40
CMU_AD/TRNG clock configuration registers	CMU_PERICKSEL	0x010	16	0x0000
CMU_I2S Clock Configuration Registers	CMU_I2SCKSEL	0x012	16	0xBBB
CMU_Debug Clock Configuration Registers	CMU_TPIUCKCFGR	0x03F	8	0x00
CMU_MCO1 Clock Output Configuration Register	CMU_MCO1CFGR	0x03D	8	0x00
CMU_MCO2 Clock Output Configuration Registers	CMU_MCO2CFGR	0x03E	8	0x00
CMU_XTAL32 Control Registers	CMU_XTAL32CR	0x420	8	0x00
CMU_XTAL32 Configuration	CMU_XTAL32CFGR	0x421	8	0x00

Registers				
CMU_XTAL32 Filtering Registers	CMU_XTAL32NFR	0x425	8	0x00
CMU_LRC control registers	CMU_LRCCR	0x427	8	0x00
CMU_LRC calibration registers	CMU_LRCTRIM	0x429	8	0x00

4.11.1 CMU XTAL Configuration Register (CMU_XTALCFGREG)

Reset value: 0x80

b7	b6	b5	b4	b3	b2	b1	b0
SUPDRV	XTALMS	XTALDRV[1:0]		-	-	-	-
<hr/>							
Bit Flag Bit Name Function Read/Write							
b7	SUPDRV	XTAL Ultra High Speed Drive Ultra High Speed Drive R/W When ultra-high-speed drive is allowed, XTAL stabilizes and disregards this bit setting to disable ultra-high-speed.	0: Disable ultra-high speed drive 1: Allows				
b6	XTALMS	XTAL mode selection bit Oscillator mode R/W 1: External clock input mode	0:				
b5~b4	XTALDRV[1:0]	XTAL drive capability selection Medium drive capability ($16 < f_{XTAL_IN} \leq 20\text{MHz}$ crystal) 10: Small drive capability (recommended for 8 $< f_{XTAL_IN} \leq 16\text{MHz}$ crystal) 11: Ultra-small drive capability ($4 \leq f_{XTAL_IN} \leq 8\text{MHz}$ crystal is recommended)	00: High drive capability ($20 < f_{XTAL_IN} \leq 25\text{MHz}$ crystal recommended) 01:				
b3~b0	Reserved	-0" for reading, "0" for writing. and writes "0". R/W	Reads "0"				

4.11.2 CMU XTAL Settlement Configuration Register (CMU_XTALSTBCR)

Reset value: 0x05

b7	b6	b5	b4	b3	b2	b1	b0			
-	-	-	-	XTALSTB[3:0]						
<hr/>										
classifier for honorific people	marking	Bit Name	Function	Read/Write						
b7~b4	Reserved	-O" for reading, "0" for writing. and writes "0".	R/W	0001: Stabilization counter 35 cycles 0010: Stabilization counter 67 cycles 0011: Stabilization counter 131 cycles 0111: Stabilization counter 2147 cycles 0100: Stabilization counter 259 cycles 1000: Stabilization counter 4291 cycles 0101: Stabilization counter 347 cycles	Reads "0"					
b3~b0	XTALSTB[3:0]	XTAL Stabilization Time Selection Stabilize counter for 1059 cycles counter = LRC cycle/8	One count cycle of the stabilization CMU_XTALCR.XTALSTP bit 1 and Configure this register with the CMU_OSCSTBSR.XTALSTBF bit 0.	1001: 1010: 0110: R/W						

4.11.3 CMU XTAL Control Register (CMU_XTALCR)

Reset value: 0x01

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	XTALSTP
<hr/>							
classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)			
b7~b1	Reserved	-	Reads "0" and writes "0".	R/W			
b0	XTALSTP	XTAL oscillator on stop bit	0: XTAL oscillator oscillation 1: XTAL oscillator stops		R/W		

Attention:

- When XTAL is selected as the system clock or MPLL/UPLL clock source, XTALSTP write 1 is disabled to stop the XTAL oscillator.
- The software sets the XTAL oscillator to oscillate and confirms that the XTAL oscillator has stabilized with the XTALSTBF bit before entering stop mode, power-down mode, or software setting the XTAL oscillator to stop.
- The software sets the XTAL oscillator to stop and confirms that the XTAL oscillator has stopped with the XTALSTBF bit before the XTAL oscillator can be entered into stop mode, power-down mode, or started again.

4.11.4 CMU XTAL Oscillation Fault Control Register (CMU_XTALSTDCR)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
XTALSTDE	-	-	-	-	XTALSTDRI	XTALSTDRE	XTALSTDIE
Bit Flag							Bit Name Function Read/Write
0: Disable XTAL oscillation fault detection							
The XTAL oscillation fault detection function allows							1: Allow XTAL oscillation fault detection
Notes: Oscillator fault detection is to detect abnormal oscillation of the oscillator caused by external factors. Before entering the stop mode or power-down mode, please disable the oscillator oscillation fault detection function.							R/W
b6~b3 Reserved -Reserved Read "0", write "0".							R/W
0: XTAL oscillation fault generates an interrupt							
1: XTAL oscillation fault generates a reset Note: When MPLL and UPLL select the XTAL clock as the input source, only the XTAL oscillation fault generating reset function can be selected.							R/W
b2 XTALSTDRI XTAL oscillation fault reset interrupt selection 0: disable XTAL oscillation fault reset 1: Allow XTAL oscillation fault reset							R/W
b0 XTALSTDIE XTAL oscillation fault interrupt allowed 0: Disable XTAL oscillation fault interruption 1: Allow XTAL oscillation fault interruption Set the PWM output of Timer6/Timer4 to Hiz output via EMB, and the XTALSTDIE bit needs to be set to 1.							R/W

Attention:

-Disable XTALSTP. When XTAL is selected as the system clock or MPLL/UPLL clock source, disable XTALSTP Write 1 to stop the XTAL oscillator.

4.11.5 CMU XTAL Oscillation Fault Status Register (CMU_XTALSTDSR)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	XTALSTDF
classifier for honorific people	marking	celebrity	functionality				fill out or in (information on a form)
b7~b1	Reserved	-	Reads "0" and writes "0".				R/W
b0	XTALSTDF	XTAL oscillation fault status bit	0: XTAL oscillation fault not detected 1: XTAL oscillation fault detected Placement conditions: XTAL oscillatory faults under conditions of XTALSTDE=1 Zeroing condition: read 1 write 0 when the system clock selects a clock other than XTAL.				R/W

4.11.6 CMU XTAL32 Configuration Register (CMU_XTAL32CFGR)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	XTAL32DRV[2:0]
classifier for honorific people	marking	celebrity	functionality				fill out or in (information on a form)
b7~b3	Reserved	-	Reads "0" and writes "0".				R/W
b2~b0	XTAL32DRV[2:0]	XTAL32 drive capability selection	000: Medium drive capacity 001: Large drive capacity Other: Prohibited settings Note: Refer to the Electrical Characteristics section for usage [Crystal/Ceramic Resonator Generated Low-speed external clock]				R/W

4.11.7 CMU XTAL32 Filter Register (CMU_XTAL32NFR)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	XTAL32NF[1:0]
classifier for honorific people	marking	celebrity	functionality				fill out or in (information on a form)
b7~b2	Reserved	-	Reads "0" and writes "0".				R/W
			00: RUN mode/stop mode/power-down mode, 3us filtering of XTAL32 is effective				

b1~b0	XTAL32	01: RUN mode XTAL32's 3us filtering is valid, stop mode XTAL32's 3us filtering is invalid	www.xhsc.com.cn	R/W
XTAL32NF[1:0]	Oscillator Filter Selection	10: Setting Prohibition 11: RUN mode/stop mode/power-down mode, XTAL32's 3us filtering is invalidated		

4.11.8 CMU XTAL32 Control Register (CMU_XTAL32CR)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	XTAL32STP
classifier for honorific people	marking	celebrity	functionality				fill out or in (information on a form)
b7~b1	Reserved	-	Reads "0" and writes "0".				R/W
b0 on	XTAL32STP	XTAL32 oscillator stop bit	0: XTAL32 oscillator oscillation 1: XTAL32 oscillator stops				R/W

Attention:

- When XTAL32 is selected as the system clock source, XTAL32STP write 1 is disabled to stop the XTAL32 oscillator.
- The software sets the XTAL32 action to start and waits for 5 XTAL32 cycles before the XTAL32 can be stopped again.
- The software sets the XTAL32 to stop and wait 5 XTAL32 cycles before the XTAL32 can be started again.

4.11.9 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/Write					
b7~b0 calibration bits	HRCTRM[7:0]	HRC frequency		Frequency calibration needs to be within the HRC frequency guarantee. 10000000: -128 10000001: -127 11111111: -1 00000000: Center Code 00000001: +1 01111110: +126 01111111: +127				R/W

4.11.10 CMU HRC Control Register (CMU_HRCCR)

Reset value: determined by ICG1.HRCSTP value

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	HRCSTP
classifier for honorific people	marking	celebrity	functionality				fill out or in (information on a form)
b31~b1	Reserved	-	Reads "0" and writes "0".				R/W
b0	HRCSTP	HRC oscillator on stop bit	0: HRC oscillator oscillation 1: HRC oscillator stops HRC starts to stop after reset according to ICG1.HRCSTOP configuration.				R/W

Attention:

- When HRC is selected as the system clock source or MPLL/UPLL clock source, CMU_HRCCR.HRCSTP Write 1 to stop HRC clock is disabled.
- The software sets the HRC oscillation and confirms that the HRC is stable with the HRCSTBF bit before entering stop mode, power-down mode, or stopping the HRC.
- The software sets the HRC to stop and confirms the MPLL stop with the HRCSTBF bit before the HRC can be entered into stop mode, power down mode, or started again.

4.11.11 CMU MRC Calibration Register (CMU_MRCTRIM)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
MRCTRIM[7:0]							
classifier for honorific people	markings	celebrity	functionality				fill out or in (information on a form)
b7~b0	MRCTRIM[7:0]	MRC frequency calibration bits	10000000: -128 10000001: -127 11111111: -1 00000000: Center Code 00000001: +1 01111110: +126 01111111: +127				R/W

Attention:

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

4.11.12 CMU MRC Control Register (CMU_MRCCR)

Reset value: 0x80

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	MRCSTP
classifier for honorific people	marking	celebrity	functionality				fill out or in (information on a form)
b7	-	-	Reads "1" and writes "1".				R/W
b6~b1	Reserved	-	Reads "0" and writes "0".				R/W
b0	MRCSTP	MRC oscillator turn on stop bit	Notes: 1) When the XTAL oscillation fault function is active, this bit is cleared simultaneously and the MRC oscillates. 2) The stop mode wake-up action when the				R/W

Attention:

PWC_STPMCR.CKSMRC bit is 1 is set when the MRC oscillator is in oscillation.

- When MRC is selected as the system clock source, MRCSTP write 1 is disabled to stop the MRC clock.
- The software sets the MRC oscillation and waits for 5 MRC cycles before entering stop mode, power-down mode, or stopping the MRC.
- The software sets the MRC to stop and waits for 5 MRC cycles before entering stop mode, power down mode, or starting the MRC again.
- MRC is used as the RTC calibration clock. possibility of MRC oscillation when RTC is not initialized. disregarding the MRCSTP bit setting when the RTC calibration function is active, MRC oscillates.

4.11.13 CMU LRC Calibration Register (CMU_LRCTRIM)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
LRCTRIM[7:0]							
classifier for honorific people	marking	celebrity	functionality				fill out or in (information on a form)
b7~b0	LRCTRIM[7:0]	LRC frequency calibration bits	10000000: -128 10000001: -127 11111111: -1 00000000: Center Code 00000001: +1				R/W

01111110: +126

01111111: +127

Attention:

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

4.11.14 CMU LRC Control Register (CMU_LRCCR)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	LCRSTP
classifier for honorific people	marking	celebrity	functionality				fill out or in (information on a form)
b31~b1	Reserved	-	Reads "0" and writes "0".				R/W
b0	LCRSTP	LRC oscillator on stop classifier for honorific people	0: LRC oscillator oscillation 1: LRC oscillator stop				R/W

Attention:

- When LRC is selected as the system clock source, LRCSTP write 1 is disabled to stop the LRC clock.
- The software sets the start of the LRC action and waits for 5 LRC cycles before entering stop mode, power-down mode, or stopping the LRC.
- The software sets the LRC to stop and waits 5 LRC cycles before entering stop mode, power down mode, or starting the LRC again.
- Wait for XTAL oscillator, HRC, MPLL, UPLL clock to stabilize while LRCSTP bit is set to ignore and LRC forces oscillation.
- When the RTC selects LRC as the clock source, the LRC ignores this register bit and the LRC oscillates. When the RTC is not initialized, the LRC may oscillate.

4.11.15 CMU MPLL Configuration Register (CMU_PLLCFGR)

Reset value: 0x1110_1300

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17		
MPLL P[3:0]		MPLL Q[3:0]		MPLL R[3:0]				-	-	-	-	-	-	-	MPL LN[8]	
b16 b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1		
MPLL N[7:0]								PLL SRC	-	-	MPLL M[4:0]					
b0																

classifier for honorific people	marking	Bit Name	Function	Read/Write
			Frequency used for MPLLP clock to write MPLLP under MPLL stop condition M P L L output clock frequency = MPLL's VCO frequency/MPLL P 0000: Disable setting	
b31-b28	MPLL P[3:0]		0001: 2-way	0010: 3
		MPLL division factor for system clock divisions	R/W 0011: 4-way 1101: 14 crossover 1110: 15 crossover frequency 1111: 16 crossover	
b27-b24	MPLL Q[3:0]		Frequency used for MPLLQ clock to write MPLLQ under MPLL stop condition M P L L output clock frequency = MPLL's VCO frequency/MPLL Q 0000: Disable setting	0010: 3
		MPLL division factor for system clock divisions	0001..21101: 14 crossover 1110: 15 crossover frequency R/W 1111: 16 crossover	
b23-b20	MPLL R[3:0]	MPLL division factor for system clock	Frequency used for MPLLR clock to write MPLLR under MPLL stop condition M P L L output clock frequency = MPLL's VCO frequency/MPLL R 0000: Disable setting	R/W
			0001: 2-way 0010: 3-way frequency 0011: 4-way 1101: 14 crossover 1110: 15 crossover frequency 1111: 16 crossover	
b19-b17	-	-	Reads "0" and writes "0".	R/W
b16-b8	MPLL N[8:0]	MPLL multiplication factor	The octave factor used to control the VCO of the MPLL. write MPLLN under the MPLL stop condition. ensure that the VCO frequency of the MPLL is between 240 MHz and 480 MHz. Between.	R/W

MPLL's VCO frequency = MPLL's PFD input frequency * *

MPLL_N

000010011: 20

			000010100: 21 000010101: 22 000010110: 23 111011101: 478 111011110: 479 111011111: 480	
b7	PLL SRC	MPLL/UPLL Input Clock Source Selection pickout	0: Selects an external high-speed oscillator as the input clock for MPLL/UPLL 1: Selecting the internal high-speed oscillator as the input clock for MPLL/UPLL	R/W
b6-b5	-	-	Reads "0" and writes "0".	R/W
			Used to divide the MPLL input clock before the MPLL's VCO. In the MPLL Write MPLLM under a stop condition. ensure that the PFD input frequency of the MPLL is between 1 MHz and Between 25MHz. 00000: 1 crossover frequency 00001: 2-way	
b4-b0	MPLLM[4:0]	MPLL Input Clock Division Factor	00010: 3-way 10111: 24 crossover frequency Other prohibitions	R/W

4.11.16 CMU MPLL control register (CMU_PLLCR)

Reset value: 0x01

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	MPLLOFF
classifier for honorific people	marking	celebrity	functionality				fill out or in (information on a form)
b7~b1	Reserved	-	Reads "0" and writes "0".				R/W
b0	MPLLOFF	MPLL Enable	Used to start stopping the MPLL. do not set this bit to 1 if the MPLL clock is used as the system clock. 0: MPLL action starts 1: MPLL stops				R/W

Attention:

- When MPLL is selected as the system clock source, MPLLOFF write 1 is disabled to stop the MPLL clock.
- The software sets the start of the MPLL action and confirms that the MPLL is stable with the MPLLSTBF bit before entering the stop mode, power-down mode, or software setting to stop the MPLL.
- The software sets the MPLL to stop, and the MPLL stop is confirmed with the MPLLSTBF bit before the MPLL can enter stop mode, power-down mode, or start the MPLL again.
- When the XTAL oscillator is selected as the clock source for MPLL, confirm that the XTAL oscillator is stabilized with the XTALSTBF bit to set the start of the MPLL action, and confirm that the HRC is stabilized with the HRCSTBF bit to set the start of the MPLL action when the HRC is selected as the clock source for MPLL.

4.11.17 CMU UPLL Configuration Register (CMU_UPLLCFGR)

Reset value: 0x1110_1300

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	
UPLL[3:0]		UPLLQ[3:0]			UPLL[3:0]			-		-		-		UPL LN[8]	
b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1
UPLLN[7:0]								-	-	-	UPLLM[4:0]				

b0

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31-b28	UPLL[3:0]	UPLL division factor for system clock	Used to control the frequency of the UPLL clock, written during UPLL stop conditions UPLL. UPLL output clock frequency = VCO frequency of UPLL/UPLL 0000: Prohibit setting 0001: 2-way 0010: 3-way frequency 0011: 4-way 1101: 14 crossover frequency 1110: 15 crossover frequency 1111: 16 crossover	R/W
b27-b24	UPLLQ[3:0]	UPLL division factor for system clock	Used to control the frequency of the UPLLQ clock, written during UPLL stop conditions UPLLQ. UPLL output clock frequency = VCO frequency of UPLL/UPLLQ 0000: Prohibit setting 0001: 2-way 0010: 3-way frequency 0011: 4-way 1101: 14 crossover frequency 1110: 15 crossover frequency 1111: 16 crossover	R/W
b23-b20	UPLL[3:0]	UPLL division factor for	Used to control the frequency of the UPLL[3:0] clock, written during UPLL stop conditions UPLL. UPLL output clock frequency = VCO frequency of UPLL/UPLL 0000: Prohibit setting 0001: 2-way 0010: 3-way frequency	R/W

system clock			
		0011: 4-way	
		
		1101: 14 crossover frequency	
		1110: 15 crossover frequency	
		1111: 16 crossover	
b19-b17	-	-	Reads "0" and writes "0". R/W
b16-b8	UPLLN[8:0]	UPLL multiplication factor	Octave factor used to control the VCO of the UPLL, written under the UPLL stop condition UPLLN. ensures that the VCO frequency of the UPLL is between 240MHz and 480MHz. Between. R/W

VCO frequency of UPLL = PFD input frequency of
UPLL * UPLLN 000010011:20

000010100: 21

000010101: 22

000010110: 23

.....

111011101:478

111011110: 479

111011111: 480

Other Prohibited Settings

b7-b5	-	b7-b5 - b7-b5 - b7-b5 - b7-b5 - b7-b5 -	"0"
b4-b0	UPLLM[4:0] division factor	UPLL input clock	<p>for read and "0" for write.</p> <p>R/W is used to divide the UPLL input clock before the VCO of the UPLL. The R/W is used to divide the UPLL input clock before the VCO of the UPLL.</p> <p>Write UPLLM under UPLL stop condition. make sure that the PFD input frequency of the UPLL is between 1MHz and 25MHz.</p> <p>00000: Prohibit setting</p> <p>00001: 2-way</p> <p>00010: 3-way</p> <p>..... 10111: 24 crossover</p>
			R/W

4.11.18 CMU UPLL Control Register (CMU_UPLLCSR)

Reset value: 0x01

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	UPLLOFF
classifier for honorific people	marking	celebrity	functionality				fill out or in (information on a form)
b7~b1	Reserved	-	Reads "0" and writes "0".				R/W
b0	UPLLOFF	UPLL Enable	Used to start stopping the UPLL. 0: Start of UPLL action 1: UPLL stop				R/W

Attention:

- When UPLL is selected as the clock source for I2S/TRNG/ADC/USBFS, write 1 to UPLLOFF is disabled to stop the UPLL clock.
- The software sets the start of the UPLL action and confirms that the UPLL is stable with the UPLLSTBF bit before entering the stop mode, power-down mode, or software setting to stop the UPLL.
- The software sets the UPLL to stop and confirms that the UPLL is stopped with the UPLLSTBF bit before the UPLL can be entered into stop mode, power-down mode, or started again.
- When the XTAL oscillator is selected as the clock source for UPLL, confirm that the XTAL oscillator is stabilized with the XTALSTBF bit to set the start of UPLL operation, and confirm that the HRC is stabilized with the HRCSTBF bit to set the start of UPLL operation when the HRC is selected as the clock source for UPLL.

4.11.19 CMU Clock Source Stabilizer (CMU_OSCSTBSR)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	UPLLSTBF	MPLLSTBF	-	XTALSTBF	-	-	HRCSTBF
classifier for honorific people	marking	celebrity		functionality			fill out or in (information on a form)
b7	Reserved	-		Reads "0" and writes "0".		R	
b6 or not stabilized	R			UPLLSTBFUPLL stabilization flag bit			0: UPLL stopped
				1: UPLL stabilization			
b5 or not stabilized	R			MPLLSTBFMPLL stabilization flag bit			0: MPLL stopped
				1: MPLL stabilization			
b4	Reserved	-Reserved		Reads "0", writes "0".		R	
b3 or not stabilized	R			XTALSTBFXTAL stabilization flag bit			0: XTAL stopped
				1: XTAL stabilization			
b2~b1	Reserved	-0" for reading, "0" for writing.					Read "0", write
"0".	R						
b0 or not stabilized	R	HRCSTBF		HRC Stabilization Flag Bit			0: HRC stopped
				1: HRC stabilization			

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]
classifier for honorific people	marking	celebrity		functionality			fill out or in (information on a form)
b31-b3	Reserved	-		Reads "0" and writes "0".		R/W	
				000: Select HRC clock as system clock			
				001: Select MRC clock as system clock			
				010: Select LRC clock as system clock			
				011: Select XTAL clock as system clock			
				100: Select XTAL32 clock as system clock			
				101: Select MPLL as system clock			
b2-b0	CKSW [2:0]	System clock source switching		110: Prohibited settings			R/W
				111: Prohibited settings			
				Notes:			

-
- 1, the switching target clock source, need to ensure that in the clock stable state.
 - 2, process refer to [clock source switching] chapter
 - 3, when the PWC_STPMCR.CKSMRC bit is 1, after the stop mode wakes up, this register
- The device is initialized and the MRC clock is selected for the system clock source.
-

4.11.21 CMU Clock Division Configuration Register (CMU_SCFGR)

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	HC KS[2:0]	-	EX KS[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]	-	-	-	-	-	-	-	-

classifier for honorific people	marking	Bit Name	Function	Read/Write
b31~b27	Reserved		-0" for reading, "0" for writing. and writes "0".	Reads "0"
			000: 1 division of the system clock 001: 2 divisions of the system clock 010: 4 divisions of the system clock 011: 8 divisions of the system clock	
b26~24	HCLKS[2:0]	HCLK clock division frequency select bit divisions of the system clock		100: 16
			101: 32 divisions of the system clock 110: 64 divisions of the system clock 111: Prohibited settings	R/W
			Note: When the PWC_STPMCR.CKSMRC bit is 1, after the stop mode wakes up, this register is initialized and HCLK is 1 division of the system clock.	
b23	Reserved		-0" for reading, "0" for writing. and writes "0".	Reads "0"
			000: 1 division of the system clock 001: 2 divisions of the system clock 010: 4 divisions of the system clock 011: 8 divisions of the system clock	
b22~b20	EXCKS [2:0]	ExMC Clock Division Frequency Selection Bits divisions of the system clock		100: 16
			101: 32 divisions of the system clock 110: 64 divisions of the system clock 111: Prohibited settings	R/W
			Note: When the PWC_STPMCR.CKSMRC bit is 1, after the stop mode wakes up, this register is initialized and EXCLK is 1 division of the system clock.	
b19	Reserved		-0" for reading, "0" for writing. and writes "0".	Reads "0"
			000: 1 division of the system clock 001: 2 divisions of the system clock 010: 4 divisions of the system clock 011: 8 divisions of the system clock	
b18~b16	PCLK4S[2:0]	PCLK4 clock division frequency selection bit divisions of the system clock		100: 16
			101: 32 divisions of the system clock 110: 64 divisions of the system clock 111: Prohibited settings	R/W

101: 32 divisions of the system clock

110: 64 divisions of the system clock

111: Prohibited settings

Note: When the PWC_STPMCR.CKSMRC bit is 1, after the stop mode wakes up, this

registers are initialized and PCLK3 is 1 division of the system clock.

b11	Reserved	-0" for reading, "0" for writing. and writes "0".	Reads "0" R/W
b10~b8	PCLK2S[2:0]	PCLK2 clock division frequency selection bit divisions of the system clock	100: 16 R/W
b7	Reserved	-0" for reading, "0" for writing. and writes "0".	Reads "0" R/W
b6~b4	PCLK1S[2:0]	PCLK1 clock division frequency selection bit divisions of the system clock	100: 16 R/W
b3	Reserved	-0" for reading, "0" for writing. and writes "0".	Reads "0" R/W
b2~b0	PCLK0S[2:0]	PCLK0 clock division frequency selection bit divisions of the system clock	100: 16 R/W

4.11.22 CMU USBFS Clock Configuration Memory (CMU_USBCKCFGR)

Reset value: 0x40

b7	b6	b5	b4	b3	b2	b1	b0																																																																													
USBCKS[3:0]				-	-	-	-																																																																													
<table border="1"> <thead> <tr> <th>Bit Flag</th><th>Bit Name</th><th>Function</th><th>Read/Write</th></tr> </thead> <tbody> <tr> <td></td><td>0010:</td><td>System clock 2 division</td><td></td></tr> <tr> <td></td><td>0011:</td><td>System clock 3 divisions</td><td></td></tr> <tr> <td></td><td>0100:</td><td>System clock 4 divisions</td><td></td></tr> <tr> <td></td><td>1000:</td><td>MPLL/P</td><td></td></tr> <tr> <td></td><td>1001:</td><td>MPLL/Q</td><td></td></tr> <tr> <td></td><td>1010:</td><td>MPLL/R</td><td></td></tr> <tr> <td></td><td>1011:</td><td>UPLL/P</td><td></td></tr> <tr> <td></td><td>1100:</td><td>UPLL/Q</td><td></td></tr> <tr> <td></td><td>1101:</td><td>UPLL/R</td><td></td></tr> <tr> <td></td><td>Other</td><td></td><td></td></tr> <tr> <td>b7~b4</td><td>USBCKS[3:0]</td><td>USB-FS at 48MHz</td><td>Bell Source Selection</td><td>prohibited setting notes:</td><td></td><td>R/W</td><td></td></tr> <tr> <td></td><td></td><td></td><td></td><td>1. When the target clock source for switching is MPLL/UPLL, it is necessary to ensure that the MPLL/UPLL clock is in a stable state.</td><td></td><td></td><td></td></tr> <tr> <td></td><td></td><td></td><td></td><td>2. When MPLL is selected for the system clock, it is necessary to set the USB, CAN, QSPI, SPI, general-purpose timer, FCM, ADC, and DAC to the module stop state, and then write the CMU_SCFG register to switch the clock division frequency. After writing the CMU_USBCKCFGR register, the software waits for the system to stabilize for 30us.</td><td></td><td></td><td></td></tr> <tr> <td></td><td></td><td></td><td></td><td>3. When the PWC_STPMCR.CKSMRC bit is 1, this register is initialized after a stop mode wake-up, and USBCLK is the 4-division frequency of the system clock.</td><td></td><td></td><td></td></tr> <tr> <td>b3~b0</td><td>Reserved</td><td>-Reserved</td><td></td><td>Read "0", write "0".</td><td></td><td>R/W</td><td></td></tr> </tbody> </table>	Bit Flag	Bit Name	Function	Read/Write		0010:	System clock 2 division			0011:	System clock 3 divisions			0100:	System clock 4 divisions			1000:	MPLL/P			1001:	MPLL/Q			1010:	MPLL/R			1011:	UPLL/P			1100:	UPLL/Q			1101:	UPLL/R			Other			b7~b4	USBCKS[3:0]	USB-FS at 48MHz	Bell Source Selection	prohibited setting notes:		R/W						1. When the target clock source for switching is MPLL/UPLL, it is necessary to ensure that the MPLL/UPLL clock is in a stable state.								2. When MPLL is selected for the system clock, it is necessary to set the USB, CAN, QSPI, SPI, general-purpose timer, FCM, ADC, and DAC to the module stop state, and then write the CMU_SCFG register to switch the clock division frequency. After writing the CMU_USBCKCFGR register, the software waits for the system to stabilize for 30us.								3. When the PWC_STPMCR.CKSMRC bit is 1, this register is initialized after a stop mode wake-up, and USBCLK is the 4-division frequency of the system clock.				b3~b0	Reserved	-Reserved		Read "0", write "0".		R/W	
Bit Flag	Bit Name	Function	Read/Write																																																																																	
	0010:	System clock 2 division																																																																																		
	0011:	System clock 3 divisions																																																																																		
	0100:	System clock 4 divisions																																																																																		
	1000:	MPLL/P																																																																																		
	1001:	MPLL/Q																																																																																		
	1010:	MPLL/R																																																																																		
	1011:	UPLL/P																																																																																		
	1100:	UPLL/Q																																																																																		
	1101:	UPLL/R																																																																																		
	Other																																																																																			
b7~b4	USBCKS[3:0]	USB-FS at 48MHz	Bell Source Selection	prohibited setting notes:		R/W																																																																														
				1. When the target clock source for switching is MPLL/UPLL, it is necessary to ensure that the MPLL/UPLL clock is in a stable state.																																																																																
				2. When MPLL is selected for the system clock, it is necessary to set the USB, CAN, QSPI, SPI, general-purpose timer, FCM, ADC, and DAC to the module stop state, and then write the CMU_SCFG register to switch the clock division frequency. After writing the CMU_USBCKCFGR register, the software waits for the system to stabilize for 30us.																																																																																
				3. When the PWC_STPMCR.CKSMRC bit is 1, this register is initialized after a stop mode wake-up, and USBCLK is the 4-division frequency of the system clock.																																																																																
b3~b0	Reserved	-Reserved		Read "0", write "0".		R/W																																																																														

4.11.23 CMU AD/TRNG Clock Configuration Memory (CMU_PERICKSEL)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PERICKSEL[3:0]
classifier for honorable people	marking	celebrity										functionality fill out or in (information on a form)			
b15~b4	Reserved	-										Reads "0" and writes "0". R/W			
b3~b0 source selection	PERICKSEL[3:0]	AD/TRNG clock 1000: PCLK2/PCLK4 set by CMU_SCFGR 1001: MPLL/Q 1010: MPLL/R 1011: UPLL/P 1100: UPLL/Q 1101: UPLL/R Settings other than these are										R/W			

Attention: prohibited.

-When the target clock source for switching is MPLL/UPLL, it is necessary to ensure that the clock is stable at MPLL/UPLL.

When the target clock source for switching is MPLL/UPLL, it is necessary to ensure that it is in the MPLL/UPLL clock steady state.

4.11.24 CMU I2S Clock Configuration Store (CMU_I2SCKSEL)

Reset value: 0xBBB

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
I2S4CKSELc				I2S3CKSEL[3:0]				I2S2CKSEL[3:0]				I2S1CKSEL[3:0]			
classifier for honorific people	marking	celebrity				functionality				fill out or in (information on a form)					
b15~b12	I2S4CKSEL	I2S clock source selection				0000: PCLK3 set by CMU_SCFG 1000: MPLL/P 1001: MPLL/Q 1010: MPLL/R 1011: UPLL/P 1100: UPLL/Q 1101: UPLL/R Otherwise, it is prohibited. Note: No clock for settings other than above								R/W	
b11~b8	I2S3CKSEL	I2S clock source selection				0000: PCLK3 set by CMU_SCFG 1000: MPLL/P 1001: MPLL/Q 1010: MPLL/R 1011: UPLL/P 1100: UPLL/Q 1101: UPLL/R Otherwise, it is prohibited. Note: No clock for settings other than above								R/W	
b7~b4	I2S2CKSEL	I2S clock source selection				0000: PCLK3 set by CMU_SCFG 1000: MPLL/P 1001: MPLL/Q 1010: MPLL/R 1011: UPLL/P 1100: UPLL/Q 1101: UPLL/R Otherwise, it is prohibited. Note: No clock for settings other than above								R/W	
b3~b0	I2S1CKSEL	I2S clock source selection				0000: PCLK3 set by CMU_SCFG 1000: MPLL/P 1001: MPLL/Q 1010: MPLL/R 1011: UPLL/P 1100: UPLL/Q 1101: UPLL/R Otherwise, it is prohibited. Note: No clock for settings other than above								R/W	

Attention:

- When the target clock source for switching is MPLL/UPLL, it is necessary to ensure that the MPLL/UPLL oscillates in a stable state.
- When MPLL/UPLL is selected as the target clock source, refer to [Clock Controller (CMU)] for details on how to configure MPLL/UPLL.

4.11.25 CMU Debug Clock Configuration Register (CMU_TPIUCKCFGGR)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
TPIUCKOE							TPIUCKS [1:0]
b7	-	-	-	-	-	-	
						Bit Flag	Bit Name Function Read/Write
b7	TPIUCKOE	TPIU clock supply allow bit 0: Prohibi t 1: Allow					
b6~b2	-	b6~b2 - b6~b2 - b6~b2 - b6~b2 - b6~b2 writes "0". R/W 00:1 crossover 01:2 crossover frequency 10:4 crossover frequency Other prohibited settings					
b1~0	TPIUCKS[1:0]	TPIU clock division select bits 01:2 crossover frequency 10:4 crossover frequency Other prohibited settings					

4.11.26 CMU MCO1 Configuration Memory (CMU_MCO1CFGGR)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0			
MCO1EN				MCO1DIV[2:0]	MCO1SEL[3:0]					
b7	-	-	-	-	Bit Flag					
disabled	MCO1EN	MCO_1 output license 0: MCO_1 output 1: MCO_1 output allowed						R/W		
b6~b4	MCO1DIV[2:0]	MCO_1 Crossover Selection 000: 1 crossover 001: 2 crossover 010: 4-way 011: 8-way frequency 100: 16 crossover 101: 32 crossover 110: 64 crossover 111: 128 crossover						R/W		
b3~b0	MCO1SEL[3:0]	MCO_1 Clock source selection 0000: HRC Clock 0001: MRC Clock 0010: LRC clock 0011: XTAL Clock 0100: XTAL32 clock 0110: MPLLP						R/W		

1000: MPLL Q

1001: UPLL Q

1011: System clock

Other prohibited settings.

4.11.27 CMU MCO2 Configuration Memory (CMU_MCO2CFGR)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
MCO2EN		MCO2DIV[2:0]			MCO2SEL [3:0]		
<hr/>							
b7 disabled	MCO2EN	MCO_2 output license		0: MCO_2 output	Bit Flag	Bit Name	Function Read/Write
<hr/>							
1: MCO_2 output allowed							
<hr/>							
b6~b4 Selection	MCO2DIV[2:0]	MCO_2 Crossover frequency		000: 1 crossover 001: 2-way 010: 4-way 011: 8-way 100: 16 crossover 101: 32 crossover 110: 64 crossover 111: 128 crossover			R/W
<hr/>							
b3~b0 source selection	MCO2SEL[3:0]	MCO_2 clock		0000: HRC Clock 0001: MRC Clock 0010: LRC clock 0011: XTAL Clock 0100: XTAL32 clock 0110: MPLLP 0111: U PLLP 1000: MPLLQ 1001: U PLLQ 1011: System clock other disable setting.			R/W
<hr/>							

4.11.28 FCM Lower Limit Comparison Value Register (FCM_LVR)

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
LVR [15:0]															

classifier for honorable people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b16	Reserved	-	Reads "0" and writes "0".	R/W
b15~b0	LVR [15:0]	lower bound comparison value	This register is configured when the START bit is 0.	R/W

4.11.29 FCM upper limit comparison value register (FCM_UVR)

Reset value: 0x0000_0000

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

classifier for honorable people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b16	Reserved	-	Reads "0" and writes "0".	R/W
b15~b0	UVR[15:0]	upper bound	This register is configured when the START bit is 0.	R/W

4.11.30 FCM Counter Value Register (FCM_CNTR)

Reset value: 0x0000_0000

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

classifier for honorable people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b16	Reserved	-	Reads "0" and writes "0".	R/W
b15~b0 clock is detected	CNTR[15:0]	Counter Value	Saves the counter value when a valid edge selected by the EDGES bit of the reference to this register (except for the first valid edge after START=1)	R

4.11.31 FCM Start Stop Register (FCM_STR)

Reset value: 0x0000_0000

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

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b1	Reserved	-	Reads "0" and writes "0".	R/W
b0	START	Frequency measurement start bit	0: Frequency measurement stopped 1: Start of frequency measurement	R/W

4.11.32 FCM Measurement Object Control Register (FCM_MCCR)

Reset value: 0x0000_0000

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

classifier for honorific people	marking	Bit Name	Function	Read/Write
b31~b8	Reserved	-	-0" for reading, "0" for writing. and writes "0".	Reads "0" R/W
			0000: XTAL 0001: XTAL32 0010: HRC 0011: LRC	
b7~b4	MCKS [3:0]	Measurement object clock selection bit SWDTRC	0100: R/W 0101: PCLK1 0110: UPLL 0111: MRC 1000: MPLP 1010: Crossover Other: Setting Prohibition	0100: 00: No crossover
b3~b2	Reserved	-	-0" for reading, "0" for writing. and writes "0".	Reads "0" R/W
b1~b0	MDIVS[1:0]	Measurement object crossover frequency selection crossover frequency	01: 4 R/W	01: 4

4.11.33 FCM Measurement Reference Control Register (FCM_RCCR)

Reset value: 0x0000_0000

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

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b16	Reserved	-	Reads "0" and writes "0".	R/W
b15	EXREFE	External pin input reference clock Disable external pin input reference clock FCMREF FCMREF allow bit R/W	0: 1: Allow external pin input reference clock FCMREF	0:
b14	Reserved	-0" for reading, "0" for writing. and writes "0".	R/W	Reads "0"
b13~b12	EDGES [1:0]	Measurement reference edge selection bit edge	00: Rising edge 01: Falling	00: Rising edge 01: Falling
b11~b10	RESERVED	Measurement reference, internal clock and terminal selection bits	11: 16 divisions of the MCKS bit-selected clock 00: 4 divisions of the MCKS bit-selected clock 10: Rising and falling edges are used as the filter clock 01: Prohibited Settings	Reads "0"
b9~b8	DNFS[1:0]	Digital Filter Function Selection Bits selected by MCKS bit as filter clock	0000: XTAL 0001: XTAL32 0010: HRC 0011: LRC 0100: SWDTRC R/W 0101: PCLK1 0110: UPLL 0111: MRC 1000: MPLLP Other: Setting Prohibition	01: Clock R/W
b6~b3	RCKS[3:0]	Measurement reference clock selection bit	0000: XTAL 0001: XTAL32 0010: HRC 0011: LRC 0100: SWDTRC R/W 0101: PCLK1 0110: UPLL 0111: MRC 1000: MPLLP Other: Setting Prohibition	01: Clock R/W
b2	Reserved	-0" for reading, "0" for writing. and writes "0".	R/W	Reads "0"
b1~b0	RDIVS[1:0]	Measurement reference crossover frequency crossover frequency	00: 32 crossover frequency 01: 1024 crossover frequency 11: 8192 crossover R/W	01: 128

4.11.34 FCM Interrupt Reset Control Register (FCM_RIER)

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	ERR E	-	-	ERRI NTRS	-	OVF IE	MEN DIE	ERR IE

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b8	Reserved	-	Reads "0" and writes "0".	R/W
b7	ERRE	Frequency Abnormal Reset Allowable Bit	0: Prohibited 1: Permission	R/W
b6~b5	Reserved	-	Reads "0" and writes "0".	R/W
b4	ERRINTRS	Frequency Abnormal Interrupt Reset Selection select a seat	0: Frequency abnormality interrupt occurs 1: Frequency anomalies occurring reset	R/W
b3	Reserved	-	Reads "0" and writes "0".	R/W
b2	OVFIE	Counter overflow interrupt allowed Counter overflow interrupt disabled bit	0: 1: Counter overflow interrupt allowed	R/W
b1	MENDIE	End-of-measurement interrupt allow bit end-of-measurement interrupt	0: Disable 1: Allow interruptions to occur at the end of the measurement	R/W
b0	ERRIE	Frequency anomaly interrupt allow bit Interrups generated by frequency anomalies are prohibited	0: 1: Frequency exception interrupt allowed	R/W

4.11.35 FCM Flag Register (FCM_SR)

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
-	-	-	-	-	-	-	-	-	-	-	-	-	OVF	MEN DF	ERR F

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b8	Reserved	-	Reads "0" and writes "0".	R/W
b2	OVF	Counter overflow flag bit	0: Counter not overflowed 1: Counter overflow	R
b1	MENDF	Measurement end flag bit Measurement in progress	R	0:

1: End of measurement

b0	ERRF	Frequency anomaly flag bit frequency anomaly occurs	0: No R
1: Abnormal frequency of occurrence			

4.11.36 FCM Flag Bit Clear Register (FCM_CLR)

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
-	-	-	-	-	-	-	-	-	-	-	-	-	OVF CLR	MEN DFC LR	ERR FCLR

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b3	Reserved	-	Reads "0" and writes "0".	R/W
b2	OVFCLR	Counter overflow flag clear		Write "1"
to clear the counter overflow flag bit.	classifier for honorific people	The counter overflow flag bit is cleared to zero.		
b1	MENDFCLR	End-of-measurement flag clearing bit	Write "1" to clear the end-of-measurement flag bit.	W
b0	ERRFCLR	Frequency Abnormal Flag Clear Bit	Write "1" to clear the end-of-measurement flag bit.	W

5 Power Control (PWC)

5.1 summary

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

The chip operates from 1.8 V to 3.6 V. The voltage regulator (LDO) supplies power to the VDD domain and the VDDR domain, and the VDDR voltage regulator (RLDO) supplies power to the VDDR domain in power-down mode. The chip provides three modes of operation, including ultra-high speed, high speed, and ultra-low speed, and three low-power modes, including sleep, stop, and power-down, through the power consumption control logic (PWCL).

The power supply voltage detection unit (PVD) provides functions such as power-on reset (POR), power-down reset (PDR), undervoltage reset (BOR), programmable voltage detection 1 (PVD1), programmable voltage detection 2 (PVD2), etc., of which, POR, PDR, BOR control chip reset by detecting the VCC voltage; PVD1 detects the VCC voltage to generate a reset or interrupt by setting according to the register; PVD2 detects VCC voltage or external input detection voltage to generate a reset or interrupt according to the register; PVD2 detects VCC voltage or external input detection voltage to generate a reset or interrupt by selecting according to the register. PVD1 detects VCC voltage and generates reset or interrupt according to the register setting, while PVD2 detects VCC voltage or external input detection voltage and generates reset or interrupt according to the register setting.

The VDDR area can maintain power through the RLDO after the chip enters power-down mode, which ensures that the real-time clock module (RTC) and wake-up timer (WKTM) can continue to operate and maintain 4KB of low-power SRAM (Ret-SRAM) data. The analog module is equipped with dedicated power supply pins to improve analog performance.

5.2 Power Distribution

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

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

The VDD domain consists of the CPU, digital logic such as digital peripherals, RAM, FLASH, etc., and is powered by the VDD generated by the LDO. The RAM in the VDD domain is divided into four groups, which can be independently powered down by register control.

The VDDR domain consists of a 4KB holding RAM (Ret-SRAM), a real-time clock (RTC), and a wake-up timer (WKTM). The power is supplied through the RLDO in power-down mode and through the LDO in modes other than power-down mode. In power-down mode, Ret-SRAM can hold data, and the real-time clock RTC and wake-up timer WKTM can continue to operate. When the functions of the VDDR domain are not required, the VDDR domain can be powered off in power-down mode by setting PWC_PWRC0.VVDRSD to further reduce power consumption.

The analog power domain consists mainly of the digital-to-analog converters (ADCs), comparators (CMPS), programmable gain amplifiers (PGAs), and the input and output pins of the analog system, which are powered by the AVCC/AVSS pins. To provide high-precision analog performance, the analog area is equipped with a separate power supply. To ensure higher accuracy of the ADC, a dedicated pin is used for the ADC reference voltage VREFH.

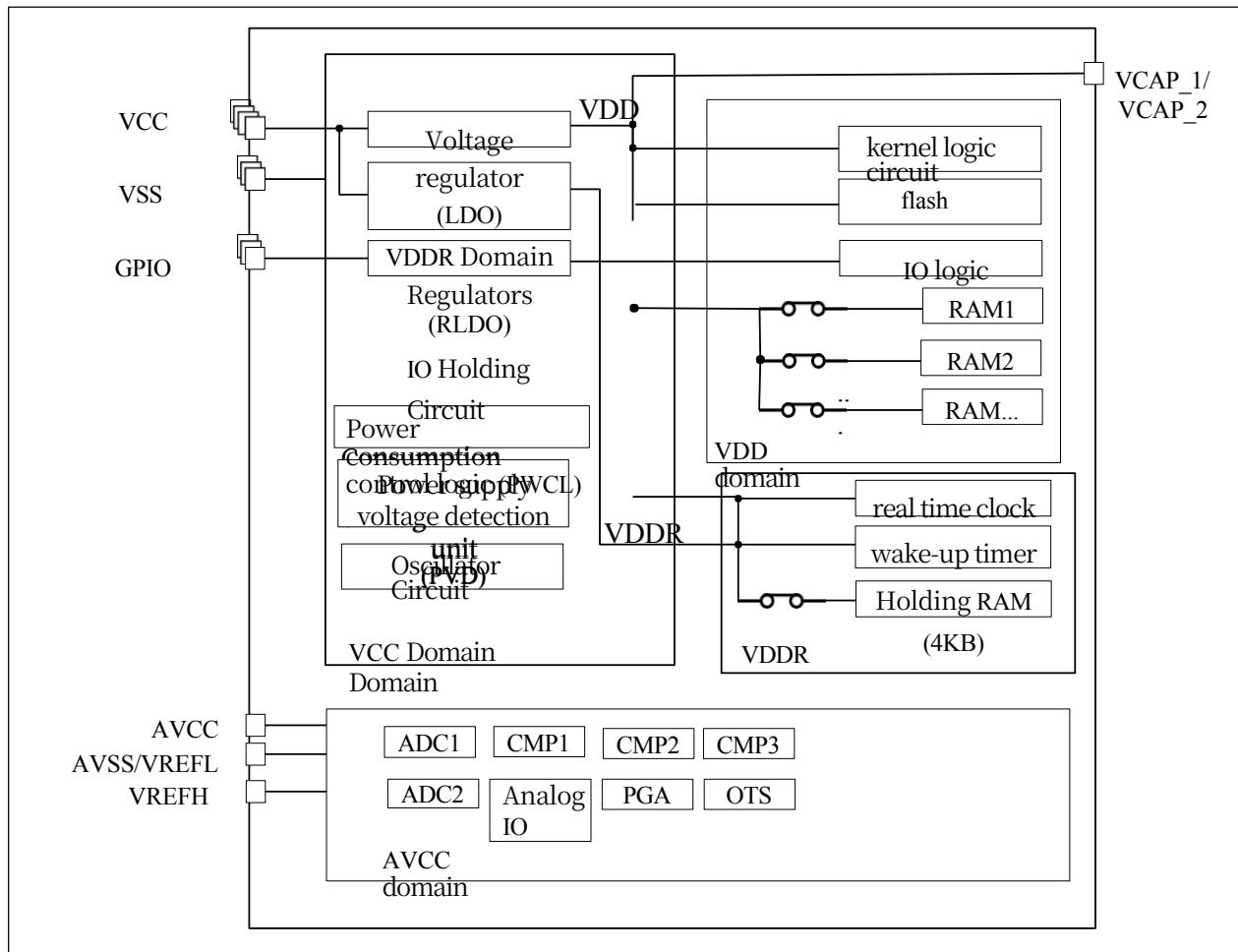


Figure 5-1 Power Supply Composition Diagram

5.3 Description of Power Supply Voltage Detection Unit (PVD)

The power supply voltage detection unit (PVD) includes power-on reset (POR), power-down reset (PDR), undervoltage reset (BOR), programmable voltage detection 1 (PVD1), and programmable voltage detection 2 (PVD2).

5.3.1 Power-on reset/power-off reset action description

The chip integrates power-on reset and power-off reset circuits. The power-on reset and power-off reset waveforms are shown in Figure 5-2. When VCC is higher than the specified threshold V_{POR} , the chip will release the power-on reset state after the T_{RSTPOR} time, and the CPU will start to execute the 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 by pin reset to start.

For detailed information on the V_{POR} , V_{PDR} , and T_{RSTPOR} parameters, refer to **Electrical Characteristics in the datasheet**.

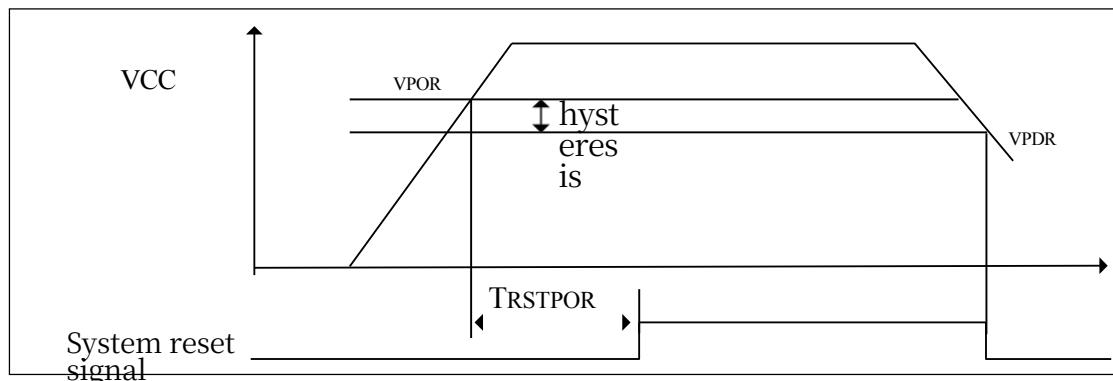


Figure 5-2 Power-on Reset and Power-off Reset Waveforms

5.3.2 Description of undervoltage reset (BOR)

During power-up, until VCC is above V_{BOR} , the undervoltage reset (BOR) will put the chip in a reset state.

The V_{BOR} thresholds are configured by BORLEV, BORDIS of the Initialization Configuration Bit (ICG). when BORDIS=0, the BOR detection voltage can be selected from 4 thresholds. when BORDIS is configured to 1, the chip performs the reset control by power-on reset, power-off reset.

Table 5-1 BOR Configuration

BORDIS	BORLEV	clarification
1	XX	BOR is not valid
0	00	BOR active, select BOR threshold 0 (V_{BOR0})
0	01	BOR active, select BOR threshold 1 (V_{BOR1})
0	10	BOR valid, select BOR threshold 2 (V_{BOR2})
0	11	BOR active, select BOR threshold 3 (V_{BOR3})

For the electrical characteristics of the BOR threshold, refer to **Electrical Characteristics**.

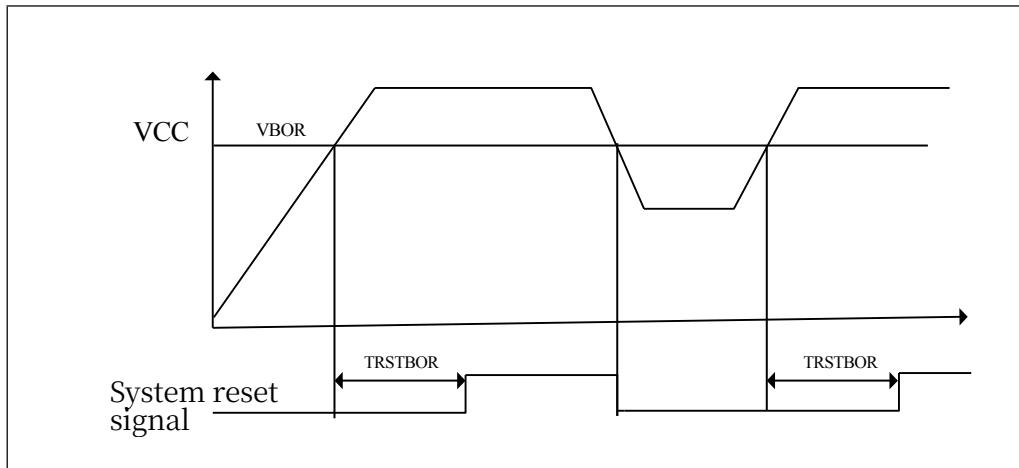


Figure 5-3 Undervoltage Reset Waveform

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

Programmable Voltage Detection 1 and Programmable Voltage Detection 2 trigger a reset or interrupt by detecting whether the VCC supply voltage passes the detection threshold. Each detection circuit is programmable.

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

The main characteristics of programmable voltage detection are shown in Table 5-2.

Table 5-2 PVD1/PVD2 Characteristics

sports event	PVD1	PVD2
target of detection	Whether or not the threshold voltage point (VPVD1) is passed during VCC fall/rise	<ol style="list-style-type: none">1. PWC_PVDLCR.PVD2LVL[2:0] is set to Whether or not the threshold voltage point (VPVD2) is passed during VCC fall/rise for values other than 1112. When PWC_PVDLCR.PVD2LVL[2:0]=111. Whether the falling/rising process of the external input voltage passes through the threshold value Voltage point (VPVD2)
Detecting Voltage Points	Configured by PVD1LVL[2:0]	Configured by PVD2LVL[2:0]
reset (a dislocated joint, an electronic device etc)	Reset: VCC<VPVD1; Reset release: V C C > VPVD1 after certain reset place Reasonable time.	Reset: VCC<VPVD2; Reset release: V C C > VPVD2 after some reset processing Between.
disruptions	Configured as Voltage Detect 1 interrupt or non-maskable interrupt	Configured as Voltage Detect 2 interrupt or non-maskable interrupt
	VCC drops past the threshold voltage point (VPVD1)	VCC drops past the threshold voltage point (VPVD2)
filtering function	digital filtration	digital filtration
Peripheral Circuit 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 Diagrams

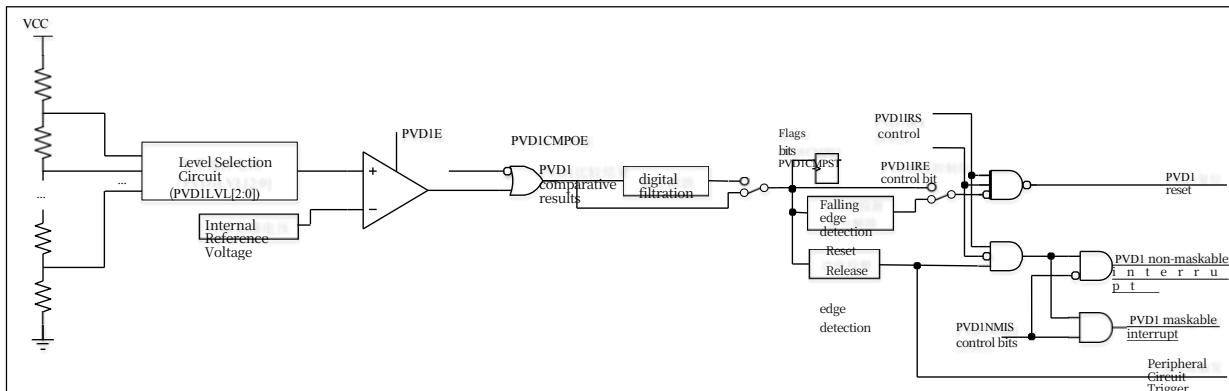


Figure 5-4 PVD1 Interrupt/Reset Block Diagram

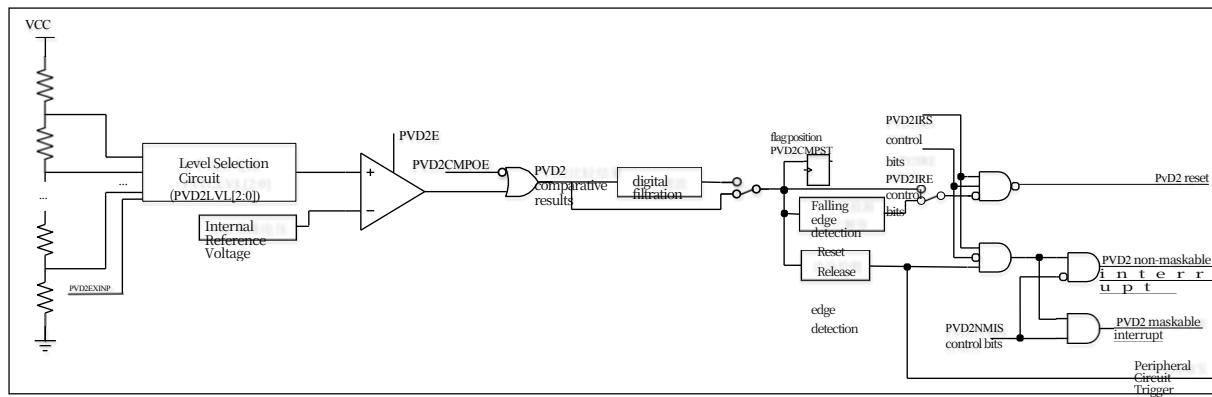


Figure 5-5 Interrupt/Reset Block Diagram

5.3.5 Input/Output

Pins

pinout	Input/Output	functionality
PVD2EXINP	importation	External input PVD2 comparison voltage

5.3.6 PVD1 Interrupt and Reset

Observe the following precautions when using the PVD1 circuit in stop mode or power-down mode.

1. Stop Mode

- 1) The digital filter must be invalidated.

2. power-down mode

- 1) The digital filter must be invalidated.
- 2) PVD1IRS is set to 0 to select PVD1 to generate an interrupt; when the reset function is selected, the power-down mode cannot be entered. The following figure shows the operation timing of the voltage monitoring 1 interrupt. PVD1DETFLG needs to be cleared to zero before the interrupt can be generated again.

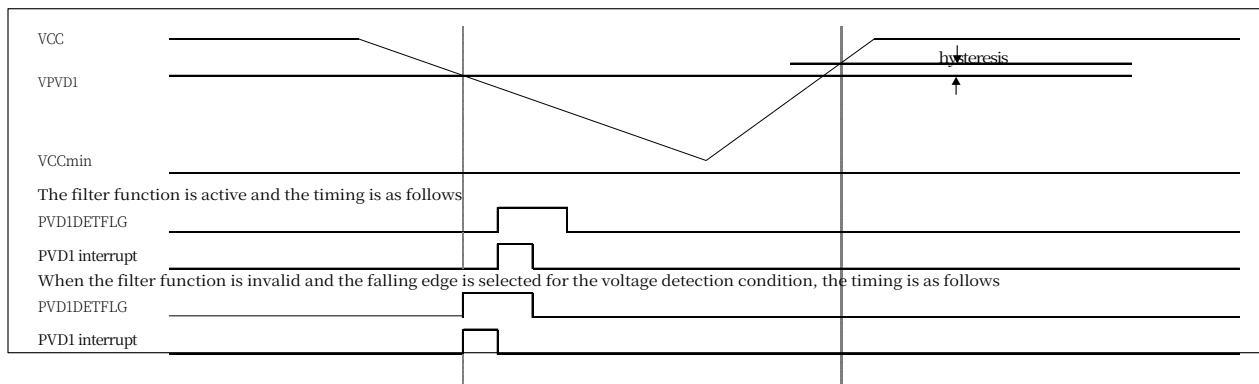


Figure 5-6 Power Monitor 1 Interrupt Timing Diagram

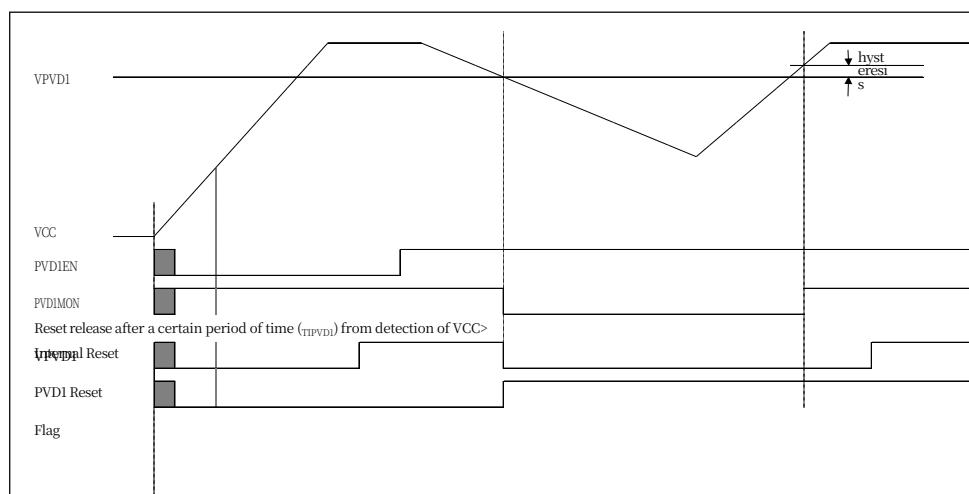


Figure 5-7 Power Monitor 1 Reset Timing Diagram

5.3.7 PVD2 Interrupt and Reset

Observe the following precautions when using the PVD2 circuit in stop mode or power-down mode:

1. Stop Mode

1) The digital filter must be invalidated.

2. power-down mode

1) The digital filter must be invalidated.

2) PVD2IRS is set to 0 to select PVD2 to generate an interrupt; when the reset function is selected, the power-down mode cannot be entered. The following figure shows the operation timing of the voltage monitoring 2 interrupt. PVD2DETFLG needs to be cleared to zero before the interrupt can be generated again.

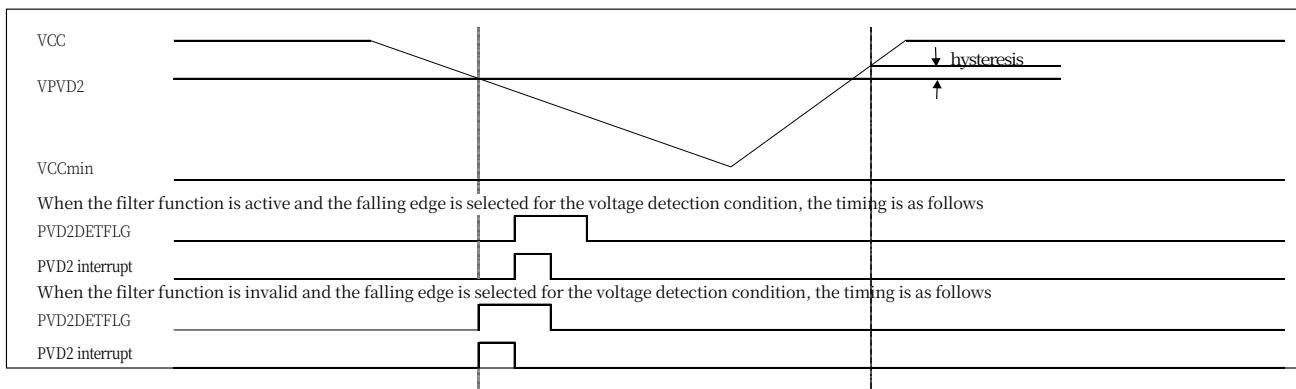


Figure 5-8 Power Monitor 2 Interrupt Run Timing Diagram

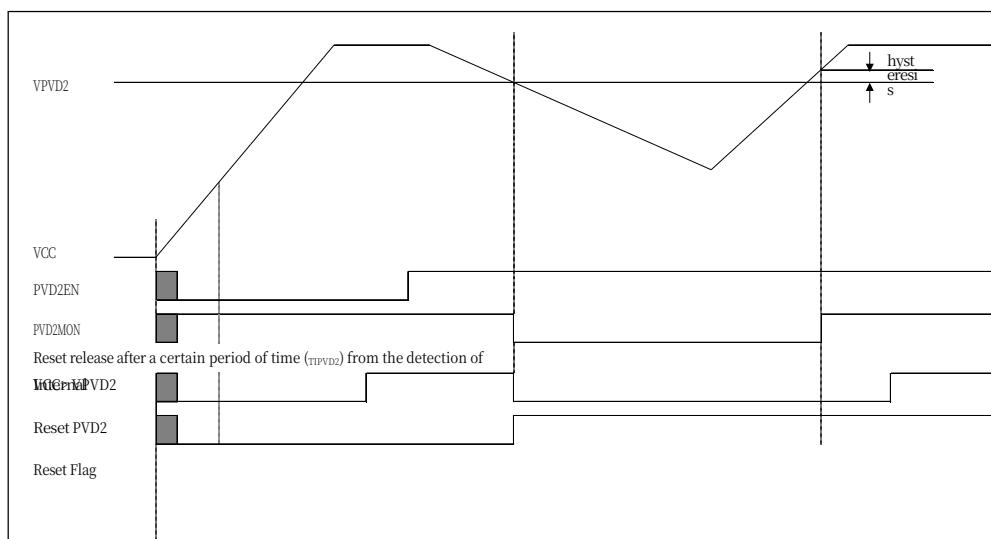


Figure 5-9 Power Monitor 2 Reset Operation Timing Chart

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 function. The reference voltage measurement path measures the reference voltage by ADC. The internal reference voltage is about 1.15V.

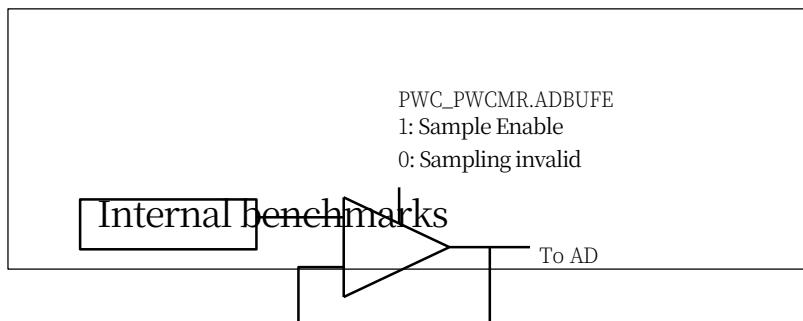


Figure 5-10 Internal Voltage Sampling Schematic

- Reference voltage measurement path

To use the reference voltage measurement path, you need to select the reference voltage measurement path as follows.

1. PWC_PWCMR.ADBUFE=1 to enable the internal voltage measurement function
2. Configure the registers according to Table 16-2 in the ADC section to enable the ADC to select the internal voltage measurement channel.
3. Wait 50uS and measure the internal reference voltage with the ADC.

5.4 Action Mode and Low Power Mode

After a system reset or power-on reset, all power domains of the chip are powered and the chip enters the high-speed run mode. In the operation mode, the CPU provides clock through HCLK and executes the program code. The chip provides three operation modes such as ultra high-speed operation mode, high-speed operation mode, and ultra-low-speed operation mode. The configurable operation modes are shown in Table 5-3.

In order to save power consumption when the CPU does not need to run, the system provides three low-power modes such as sleep mode, stop mode, and power-down mode. The configurable low-power modes are shown in Table 5-4. In sleep mode, the CortexTM-M4F core of the chip stops and the peripherals remain running; in stop mode, the peripherals of the chip and the CPU stop; and in power-down mode, the power of the VDD domain is turned off and the peripherals in the VDD domain stop. The real-time clock and wake-up timer in the VDDR domain can be operated in low-power mode, and the hold SRAM can hold data; when the real-time clock, wake-up timer and hold SRAM in the VDDR domain are not needed, the regulator RLDO in the VDDR domain can be set to be turned off, which can further reduce the power consumption after entering into the power-down mode.

Users can choose between run mode and low-power mode depending on the application to find the best balance between low power consumption, short startup time, wakeable 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 shown in Table 5-5.

Table 5-3 Operating Modes

operating mode	clarification
Ultra high speed operation mode (Ultra high Speed Run Mode)	Mains frequency below 200MHz
High-speed operation mode (High Speed Run Mode)	Mains frequency 168MHz or less
Ultra-low speed operation mode (Ultra low Speed Run Mode)	Mains frequency below 8MHz

Table 5-4 Low Power Consumption Modes

paradigm		clarification
Sleep Mode		CPU clock stops, peripherals stay running
Stop Mode		The clocks of both the chip peripherals and the CPU are stopped
Power Down Mode	Power-down mode 1 (PDMD1)	VDD domain power down
	Power-down mode 2	Voltage detection unit invalid outside of VDD domain power down

	(PDMD2)	
	Power Down Mode 3 (PDMD3)	In addition to the VDD domain being powered down, the VDDR domain is powered down, the power-on reset circuit enters the low-power mode, and the voltage detection unit (PVD) is invalidated. Compared to the power-down mode 4, the wake-up from power-down except for the PWC_PDWF0/PWC_PDWF1/RSTF0 external chip is completely reset.
	Power down mode 4 (PDMD4)	In addition to the VDD domain being powered down, the VDDR domain is powered down, the power-on reset circuit enters a low-power mode, and the voltage Invalid detection unit (PVD)

Table 5-5 Operating conditions of low-power mode and status of each module in low-power mode

sports event	sleep mode	Stop Mode	power-down mode
go into	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
relieve (sb. of their duties)	Arbitrary interrupt or reset	Interrupts that can be used in stop mode or reset (a dislocated joint, an electronic device etc)	Wake-up events that can be used in power-down mode Pieces or reset
External High Speed Oscillator	Work can be set	cessation	cessation
External low-speed oscillator	Work can be set	Work can be set	Work can be set
Internal high-speed oscillator	Work can be set	cessation	power down
Internal medium speed oscillator	Work can be set	cessation	power down
Internal low-speed oscillator	Work can be set	Work can be set	Work can be set
WDT Specialized Clock Oscillator	Work can be set	Work can be set	power down
PLL	Work can be set	cessation	power down
PLLM	Work can be set	cessation	power down
CPU	stop	stop	power down
RAM	Work can be set Can be set to work, power down	stop Depending on the setting before entering standby Maintains power down or sleep	power down
flash	Work can be set	stop	Power down, content maintained
DMA	Work can be set	stop	power down
regulator	(of a machine) operate The driver can be adjusted	(of a machine) operate The driver can be adjusted	cessation
Power-on reset circuit	(of a machine) operate	(of a machine) operate	(of a machine) operate Power-down mode 1, power-down mode 2 reset circuit accuracy can be guaranteed, power-up reset circuit in power-down mode 3 and power-down mode 4 can be guaranteed, and power-up

			reset circuit in power-down mode 4 can be guaranteed. pressure does not guarantee
Undervoltage Reset BOR	Work can be set	Work can be set	Power-down mode 1 operation can be set Stop in power-down mode 2/3/4
Voltage Detection Module PVD	Work can be set	Work can be set	Power-down mode 1 operation can be set Stop in power-down mode 2/3/4
WDT	Work can be set	stop	power down
SWDT	Work can be set	Work can be set	power down
RTC	Work can be set	Work can be set	Power-down mode 1/2 work can be set Power down in power down mode 3/4
USB-FS	Work can be set	stop	power down
Timer0	Work can be set	Work can be set	power down
Ret-SRAM	Work can be set	stop	Stop (hold) in power-down mode 1/2

sports event	sleep mode	Stop Mode	power-down mode
	Can be set to work, power down, sleep	Power down, sleep can be set	Power down, sleep can be set Power down mode 3/4 power down
WKT M	Work can be set	Work can be set	Power-down mode 1/2 work can be set Power down in power down mode 3/4
Other peripheral modules	Work can be set	stop	power down
AD	Work can be set	cessation	power down
DA	Work can be set	Work can be set	power down
PGA	Work can be set	Work can be set	power down
CMP	Work can be set	Work can be set	power down
pa0-pa10, pb0-pb2, pb5-pb10, pb12-pb15, pc0-pc13, pd0-pd15, pe0-PE15, PH2	Work can be set	remain	Hold or high resistance
PC14-PC15	Work can be set	When used as an external low-speed oscillator pin, the oscillator operation is maintained; when used as a GPIO or other peripheral function, set the setting to maintain both pins at the same level. power level (elec.)	When set to GPIO or other peripheral functions, the state of PC14 and PC15 can be set to hold or high resistance, please set to keep both pins at the same level.
PH0-PH1	Work can be set	When used as an external high-speed oscillator, the oscillator stops oscillating and the pin state remains as it was before entering STOP mode; set to GPIO or other peripheral functions. When the status before STOP is maintained	When used as an external high-speed oscillator, the oscillator stops oscillating and the pin state remains as it was before entering STOP mode; set to GPIO or other peripheral functions. When the status before STOP is maintained
NRST reset pin	The outside of the chip is pulled up to VCC via a circuit	The outside of the chip is pulled up to VCC through a resistor	The outside of the chip is pulled up to VCC through a resistor

PA11-PA12	Work can be set	Keep; Since redundant currents are generated when the level of this pin is pulled high into the STOP mode The pull-up is prohibited during the style.	Hold or high resistance; Since redundant currents are generated when the level of this pin is pulled high, it enters a power-down mode When pull-up is prohibited.
PB11/MD	Work can be set	Keep;	Keep; The outside of the chip is pulled up to VCC through a resistor
PA13-PA15, PB3,PB4	Work can be set. Built-in pull-up when used as a JTAG function Circuit active	Keep; Built-in pull-up when used as a JTAG function Circuit active	Keep; Built-in pull-up when used as a JTAG function Circuit active

5.4.1 operating mode

The chip has three operation modes such as ultra-high speed, high speed and ultra-low speed. According to the speed requirement of the system, the best operation mode is set so that the core voltage and driving capability are adapted to the system clock frequency, thus achieving the purpose of reducing power consumption. As shown in Table 5-6, according to the DVS and DDAS bits of PWC_PWRC2, the chip can be made to work in the corresponding operation mode. When the chip selects to operate in ultra-low-speed mode, the FLASH and RAM also need to be set to operate at low core voltage, so it is necessary to set registers EFM_FRMC.LVM=1 and PWC_RAMOPM=0x9062.

Table 5-6 Description of Operation Modes

operating mode	frequency range	Register Setting	
		PWC_PWRC2.DVS	PWC_PWRC2.DDAS
Ultra high speed operation mode	≤200MHz	00	1111
High-speed operation mode	≤168MHz or less	11	1111
Ultra-low speed operation mode*	≤8MHz or less	10	1000

*: Ultra low speed mode only supports Ta=-40°C~85°C

The switching between ultra-high speed operation mode, high speed operation mode, and ultra-low speed operation mode needs to follow the following flow 1 to flow 6.

1. High-speed mode to ultra-low-speed mode

- 1) Set the clock source to be used in ultra-low-speed mode to ensure that the clock source meets the frequency requirements in ultra-low-speed mode
- 2) Turn off clock sources and modules that are not required in ultra-low-speed mode, and make sure the FLASH is not programmed or erased.
- 3) Set FRMC.LVM=1 for FLASH and PWC_RAMOPM for RAM action mode register to 0x9062.
- 4) Confirm FRMC.LVM=1 and PWC_RAMOPM=0x9062
- 5) Set PWC_PWRC2.DDAS[3:0] to 1000; PWC_PWRC2.DVS[1:0] to 10
- 6) Write PWC_MDSWCR=0x10
- 7) Waiting for TSWMD1 (30uS)
- 8) Chip acting in ultra-low speed mode

2. Switching from ultra-low speed mode to high speed mode

- 1) Set PWC_PWRC2.DDAS[3:0] to 1111; PWC_PWRC2.DVS[1:0] according to the

system frequency to be

Asked to be set to 11

- 2) Write PWC_MDSWCR=0x10
- 3) Waiting for TSWMD2 (30uS)
- 4) Set FRMC.LVM=0 for FLASH and PWC_RAMOPM to 0x8043 for RAM action mode register.
- 5) Verify FRMC.LVM=0 and PWC_RAMOPM=0x8043
- 6) Chip action in high speed mode

3. High-speed mode to ultra-high-speed mode

- 1) Set PWC_PWRC2. DDAS[3:0] to 1111; PWC_PWRC2. DVS[1:0] set to 00
- 2) Write PWC_MDSWCR=0x10
- 3) Waiting for T SWMD2 (30uS)
- 4) Chip action in ultra-high speed mode

4. Switching from ultra-high speed mode to high speed mode

- 1) Set the clock source to be used in high-speed mode to ensure that the clock source meets the frequency requirements in high-speed mode.
- 2) Set PWC_PWRC2. DDAS[3:0] to 1111; PWC_PWRC2. DVS[1:0] set to 11
- 3) Write PWC_MDSWCR =0x10
- 4) Waiting for TSWMD2 (30uS)
- 5) Chip action in high speed mode

5. Switching from ultra-low speed mode to ultra-high speed mode

- 1) Set PWC_PWRC2. DDAS[3:0] to 1111; PWC_PWRC2. DVS[1:0] set to 00
- 2) Write PWC_MDSWCR =0x10
- 3) Waiting for TSWMD2 (30uS)
- 4) Set FRMC.LVM=0 for FLASH and PWC_RAMOPM to 0x8043 for RAM action mode register.
- 5) Verify FRMC.LVM=0 and PWC_RAMOPM=0x8043
- 6) Chip action in ultra-high speed mode

6. Switching from ultra-high speed mode to ultra-low speed mode

- 1) Set the clock source to be used in ultra-low-speed mode to ensure that the clock source meets the frequency requirements in ultra-low-speed mode
- 2) Turn off clock sources and modules that are not required in ultra-low-speed mode, and make sure the FLASH is not programmed or erased.
- 3) Set FRMC.LVM=1 for FLASH and PWC_RAMOPM for RAM action mode register to 0x9062.
- 4) Verify FRMC.LVM=1 and RAMOPT=0x9062
- 5) Set PWC_PWRC2.DDAS[3:0] to 1000; PWC_PWRC2.DVS[1:0] to 10
- 6) Write PWC_MDSWCR =0x10
- 7) Waiting for TSWMD1 (30uS)
- 8) Chip acting in ultra-low speed mode

5.4.2 sleep mode

In sleep mode, the CPU stops running and its internal registers remain in the state they were in before entering sleep mode. The state of the peripherals and other system module actions other than watchdogs and specialized watchdogs do not change.

When set to automatic startup via ICG, the watchdog stops counting in sleep mode if the WDTSLPOFF bit of ICG is 1; if the WDTSLPOFF bit is 0, the watchdog continues counting in sleep mode. If the ICG is not set to autostart and the watchdog is started by software startup, the watchdog stops counting in sleep mode if the WDT_CR.SLPOFF bit is 1, and the watchdog does not stop counting in sleep mode if the WDT_CR.

When set to auto-start via the ICG, the dedicated watchdog stops counting in sleep mode if the SWDTSLPOFF bit of the ICG is a 1. The dedicated watchdog continues counting in sleep mode if the SWDTSLPOFF bit is 0.

- Go to sleep mode

The sleep mode can be entered by executing the WFI command with PWC_STPMCR.STOP=0.

- Exit Sleep Mode

Any interrupt or reset can wake up the chip from sleep mode. When waking up via interrupt, the chip enters the interrupt handling program; when exiting sleep mode via reset, the chip enters the reset state.

5.4.3 Stop Mode

In stop mode, the CPU, most peripherals, and the clock source stop acting. The chip maintains the CPU internal registers and SRAM data, peripheral states and pin states. In stop mode, the chip power consumption is significantly reduced because most of the clock sources are out of action and the regulator has reduced drive capability.

When set to auto-start via the ICG, the dedicated watchdog stops counting in stop mode if the SWDTSLPOFF bit of the ICG is 1, and continues counting in stop mode if the SWDTSLPOFF bit is 0.

Before executing the WFI command to enter the stop mode, you need to make sure that the FLASH is not in the programmed or erased state, and the oscillation stop monitoring function is invalid, otherwise the chip will enter the sleep mode instead of the stop mode.

In stop mode, the ADC and DAC will also consume power unless they are disabled before entering stop mode. To disable DAC, you need to clear DACR.DAE, DAOE0, DAOE1 to 0. To disable ADC, you need to clear ADC_STR.START bit and then write 1 to the corresponding bit of ADC in PWC_FCG3 to make the ADC enter into the module stop state before executing the WFI instruction to enter the stop mode.

When waking up in STOP mode, the clock after waking up and whether to wait for FLASH to stabilize are selected by bits CKSMRC and FLNWT of PWC_STPMCR register. CKSMRC is used to control the clock source after waking up, when CKSMRC=1, the system clock source after waking up is selected as MRC, when CKSMRC=0, the system clock after waking up remains unchanged before entering STOP mode. FLNWT is used to control whether to wait for FLASH stabilization after wake-up, when FLNWT=0, wake-up needs to wait for FLASH stabilization; when FLNWT=1, wake-up does not need to wait for FLASH stabilization; FLNWT can only be set when the program runs on RAM, otherwise, the chip will not be stabilized after wake-up from STOP.

The action is not guaranteed. Selecting CKSMRC =1, FLNWT=1 will wake up the system in the shortest possible time while the program is running in STOP mode on RAM.

Before executing the WFI command to enter the stop mode, it is necessary to make sure that the DMA is in the stop state, otherwise the chip may have an unguaranteed action.

Before executing the WFI command to enter the stop mode, you need to make sure that the division ratio of HCLK to PCLK1/2/3/4 and EXCLK is controlled at 1:1 or 1:2 or 1:4. If the division ratio of HCLK to any of the peripheral clocks is more than 1:4, you need to configure PCLK1=PCLK3 with PCLK1 and PCLK3 as the slowest clocks.

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

Before executing the WFI instruction to enter the stop mode, the digital filter of EIRQ needs to be set to invalid, otherwise the interrupt cannot be used for STOP wakeup.

The chip needs to set PWC_PWRC1.STPDAS=11 before entering STOP mode in ultra-low-speed mode; if the chip enters STOP mode when PWC_PWRC1.STPDAS=00, the chip will consume more current in STOP mode.

To release the stop mode by non-maskable interrupt, set the corresponding bit of INT_NMIER to enable the interrupt; to release the stop mode by maskable interrupt, set the corresponding bit of INT_WUPENR register to enable the interrupt to wake up. Before executing the WFI or WFE command, make sure that all interrupts that are not used for wake-up from stop mode have been turned off.

- Entering stop mode

Execute the WFI instruction when PWC_STPMCR.STOP=1,PWC_PWRC0.PWDN=0 to enter the stop mode. Table 5-5 shows the status of the peripherals and clock sources in stop mode.

- Disengage stop mode

The stop mode can be released by reset and interrupt. The resets that can be used to release the stop mode are pin reset, power-on reset, undervoltage reset (BOR), programmable voltage monitoring 1/2 reset, and dedicated watchdog reset. The interrupt events that can be used to release the stop mode are as follows:

NMI pin interrupt, pin interrupt EIRQ0-15, and voltage monitor 1 interrupt,
Voltage monitoring 2 interrupt, dedicated watchdog underflow
interrupt, cycle interrupt of real-time clock, alarm interrupt, wake-up
timer interrupt, comparator interrupt, USART1 RX interrupt, Timer0
compare match interrupt

When the chip deactivates the stop mode by interrupting, it first starts the clock sources used before entering the stop mode. After all clock sources have stabilized, the chip deactivates the stop mode.

If an interrupt event that is not used to deactivate the stop mode is generated at the same time that the WFI instruction is executed to enter the stop mode, the chip will prioritize responding to the interrupt instead of entering the STOP mode. If it is necessary to prioritize entering STOP mode without executing the interrupt, the interrupt that is not used to release the stop mode needs to be turned off before executing the WFI instruction.

5.4.4 power-down mode

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

If the SWDTSLPOFF bit in ICG is 1 when set to auto-start via ICG, the dedicated watchdog will be powered down and will not count any more like other modules in the VDD domain. If the SWDTSLPOFF bit is 0, the chip will enter the stop mode instead of the power-down mode, and the oscillator of the dedicated watchdog and the dedicated watchdog will continue to run if it is set to auto-start in the ICG.

When reset is enabled for Voltage Monitor 1 and Voltage Monitor 2, the chip enters stop mode instead of power-down mode.

Before executing the WFI command to enter the power-down mode, make sure that the FLASH is not in the programmed or erased state, and the oscillation stop monitoring function is disabled, otherwise the chip will enter the sleep mode instead of the power-down mode.

The capacitors used on the VCAP_1/VCAP_2 pins of the chip are as follows: 1) For chips with both VCAP_1 and VCAP_2 pins, 0.047uF or 0.1uF capacitance can be used on each pin (total capacity of 0.094uF or 0.2uF) 2) For chips with only VCAP_1 pin, 0.1uF or 0.22uF capacitance can be used. 3) For chips with VCAP_1 pin, 0.1uF or 0.22uF capacitance can be used. 4) For chips with VCAP_1 pin, 0.1uF or 0.22uF capacitance can be used. capacitors. When waking up from power-down mode, VCAP_1/VCAP_2 needs to be charged during the kernel voltage establishment process. On the one hand, a smaller total capacity of VCAP_1/VCAP_2 reduces the charging time and provides fast response time for the application; on the other hand, a larger total capacity of VCAP_1/VCAP_2 increases the charging time but also provides better electromagnetic compatibility (EMC). Users can choose smaller or larger capacitors depending on the EMC and system response speed requirements. The total capacity of VCAP_1/VCAP_2 on the chip must match the value assigned to the PWC_PWRC3.PDTS bit. When the total capacity of VCAP_1/VCAP_2 is 0.2uF or 0.22uF, it is necessary to make sure that the PWC_PWRC3.PDTS bit is cleared to zero before entering power-down mode. If the total capacity of VCAP_1/VCAP_2 is 0.094uF or 0.1uF, make sure the PWC_PWRC3.PDTS bit is clear before entering power-down mode.

The power consumption of power-down mode can be further reduced by setting PWC_PWRC0.PDMDS[1:0]. The sub-modes of power-down mode are shown in Table 5-7. In power-down mode 1, the voltage monitoring circuit is available and the power-on reset detection circuit is in action. Since there is no need to wait for the stabilization of the VCC domain reference voltage circuit, the voltage monitoring circuit, and the power-on reset detection circuit when waking up, the wake-up time is the shortest while achieving low power consumption. In power-down mode 2, the VCC domain reference voltage circuit, voltage

monitoring circuit stop working, and the power-on reset detection circuit is in the active state, so the wake-up time needs to wait for the stabilization of the VCC domain reference voltage circuit and voltage monitoring circuit. In power-down mode 3, the VCC domain reference voltage circuit, voltage monitoring circuit, and power-on reset detection circuit stop working, and the wake-up time needs to wait for these circuits to stabilize, so the wake-up time is longer than that of power-down mode 2 and power-down mode 1 while achieving the lowest power consumption. Power-down mode 4 has the same circuits that stop working in power-down mode 3, so power-down mode 4 has the same power consumption as power-down mode 3. Refer to **Electrical Characteristics-Low Power Mode** for specific power consumption values and wake-up times.

The VDDR field works 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 in power-down mode. the Ret-SRAM can still hold data in power-down mode. If the real-time clock, wake-up timer, and Ret-SRAM are not needed in power-down mode, the low-power regulator can be turned off by setting PWC_PWRC0.VVDRSD to further reduce power consumption. The VDDR field is also powered down in power-down mode 3 and power-down mode 4.

Table 5-7 Power-down Mode Submodes

power-down mode	PDMD[1:0]	power wastag e	wake-up time	clarification
Power down mode 1	00	IPD1	TPD1	VCC domain power supply voltage detection unit valid
Power-down mode 2	10	IPD2	TPD2	VCC domain POR, PDR detection circuit valid, BOR, PVD1, PVD2 no efficacy
Power-down mode 3	01	IPD3	TPD3	VCC domain POR, PDR, BOR, PVD1, PVD2 invalidated VDDR domain power down
Power-down mode 4	11	IPD4	TPD4	VCC domain POR, PDR, BOR, PVD1, PVD2 invalidated VDDR domain power down

Relationship between power consumption and wakeup time: $\text{IpD1} > \text{IpD2} > \text{IpD3} = \text{IpD4}$, $\text{TpD1} < \text{TpD2} < \text{TpD4} < \text{TpD3}$

■ Entering power-down mode

Execute the WFI command when PWC_STPMCR.STOP=1, PWC_PWRC0.PWDN=1 to enter power-down mode.

■ Deactivate power-down mode

The power-down mode can be disarmed by a power-down mode wake-up event or a reset.

Resets that can be used to wake up from power-down mode are pin reset, power-up reset, and voltage monitor 0 reset. Events that can be used to wake up from power-down mode include:

NMI wake-up event, WKUPn_0/1/2/3 (n=0/1/2/3) wake-up event, real-time clock alarm and timer events, voltage monitoring 1 wake-up event, voltage monitoring 2 wake-up event, wake-up timer wake-up event

After waking up from power-down mode 1 and power-down mode 2, the chip resets and re-executes the program. The wake-up event can be queried by the power-down wake-up flag bit, and the reset flag bit can be queried by RSTF0.PDRF.

In power-down mode 3, POR, PDR, BOR, PVD1, PVD2 circuits are invalid, after waking up from power-down mode 3, all registers except PWC_PDWKF0/PWC_PDWKF1/RSTF0 are reset, and the chip operates similar to the power-on reset; the reset flag can be queried by RSTF0.PDRF. After waking up from power-down mode 3, the RTC/WKTM of VDDR domain is reset and the data in Ret-SRAM cannot be guaranteed.

In power-down mode 4, POR, PDR, BOR, PVD1, PVD2 circuits are invalidated, and the program is re-executed after chip reset; the reset flag bit can be queried by RSTF0.PDRF. When waking up from power-down mode 4, the RTC/WKTM of VDDR domain is reset,

and the data in Ret-SRAM cannot be guaranteed.

The power-down wake-up event is controlled by the power-down wake-up enable registers (**PWC_PDWKE0-PDWKE3**) and the power-down wake-up event edge selection register (**PWC_PDWKES**). When a wake-up from power-down event occurs, the corresponding wake-up from power-down flag (**PWC_PDWKF0-PWC_PDWKF1**) is set. After a power-down wake-up event, the chip cannot enter the power-down mode again without clearing the power-down wake-up flag. The edge of the power-down wake-up event can be selected by **PWC_PDWKES**.

Upon wake-up from power-down mode, the VDD domain will be repowered, the system performs a wake-up from power-down reset, and the operating clock internal medium-

power-down mode	Registers that are not reset	ode are
Power-down mode 1	pwc_pwrc0 pwc_pwrc1 pwc_pwrc3 pwc_pdwke0 pwc_pdwke1 pwc_pdwke2 pwc_pdwkes pwc_pdwkf0 pwc_pdwkf1 pwc_pwcmr	
Power-down mode 2		
Power-down mode 4		

	pwc_xtal32cs pwc_pvdcr0 pwc_pvdcr1 pwc_pvdcr pwc_pvdlcr pwc_pvdlcr PWC_PVDDSR
Power-down mode 3	pwc_pdwkf0 pwc_pdwkf1 rstf0

- Pin status after power-down mode is removed

In power-down mode, the chip pins will remain in the state before entering power-down mode or be set to high-resistance state through registers. If PWC_PWRC0.IORTN[1:0]=10 or 11, the pin state is high resistance in power-down mode and is initialized after power-down mode is released. If PWC_PWRC0.IORTN[1:0]=00, the pin state remains in power-down mode as it was before power-down mode, and the pin is initialized to a high-resistance state after wake-up. If PWC_PWRC0.IORTN[1:0]=01, the chip pin will keep the state before entering the power-down mode, and the state of the chip pin will not be changed even if the registers of the peripheral or the pin are set after wake-up. IORTN will be controlled by the register setting of the peripheral or the pin only after PWC_PWRC0.IORTN is cleared to zero by software.

- WKTM pin for wake-up from power-down

The chip has a built-in counter WKTM for wake-up from power-down, which can select internal low-speed oscillator, external low-speed oscillator, and 64Hz internal clock signal as the clock source, of which the 64Hz internal clock signal is valid when the RTC uses the external low-speed oscillator as the clock source and the RTC is actuated. The counter is a counting counter. After WKTC0.WKTCE is set, the counter starts counting, and when the counting value equals to the WKTCMP[11:0] setting, the counting stops and a wake-up event is generated to wake up the chip from the power-down mode. WKTC0.WKTCE needs to be reset and set again when WKTM is used again.

- PTWK Power-down mode wake-up event

The chip has 4 PTWK events for wake-up from power-down mode, PTWK0, PTWK1, PTWK2, PTWK3. PWC_PDWKE0/PWC_PDWKE1 can be set in software to enable one of the 4 pins, WKUPn_0, WKUPn_1, WKUPn_2, WKUPn_3 to trigger the PTWK event. Each PTWK event can be selected to trigger the rising or falling trigger edge of the pin and has a separate flag bit.

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

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

The structure block diagram for configuring PTWK_n is shown in Figure 5-11.

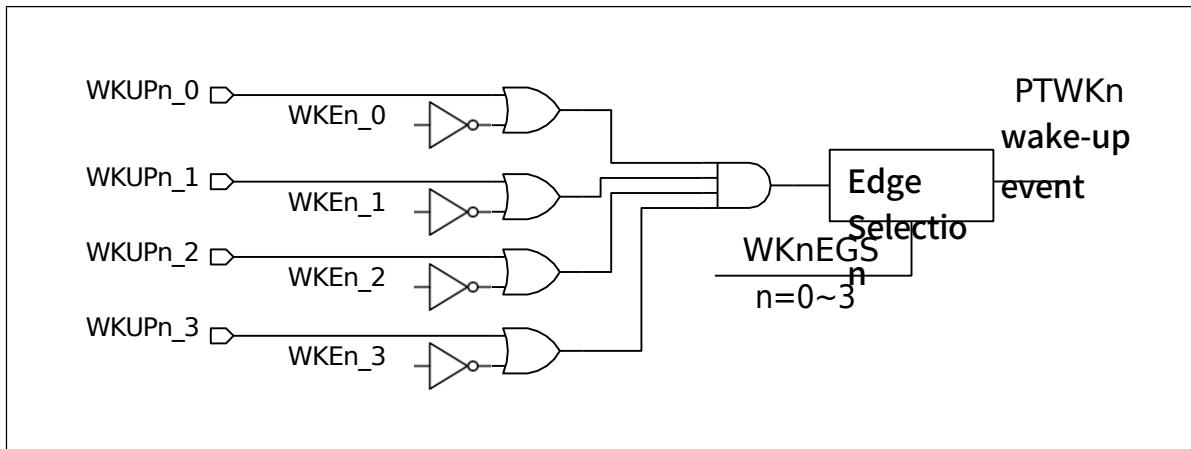


Figure 5-11 PTWK_n Structure Block Diagram

■ Description of the VDDR domain's actions during power-down mode

The VDDR domain continues to be powered by the RLDO after the chip enters Power Down Mode 1 or Power Down Mode 2, so that the RTC/WKTM/Ret-SRAM can continue to act or hold data. The VDDR domain is powered by the LDO after wake-up from power-down. In power-down mode 3 or power-down mode 4 the VDDR field is de-energized.

5.5 Ways to Reduce Power Consumption

You can optimize the power consumption in run mode by the following methods.

4. Setting the optimal operating mode
5. Reduced system clock speed
6. Turn off unused clock sources
7. Set Function Clock Control Register PWC_FCGn (n=0/1/2/3) Turns off functions that do not need to be used
8. Powerdown the RAM

5.5.1 Reduced system clock speed

In Run mode, the system clock (**HCLK**), external bus clock (**EXCLK**), and peripheral peripheral clocks **PCLK0/PCLK1/PCLK2/PCLK3/PCLK4** can be slowed down by programming the prescaler registers. These prescalers can also be used to reduce the peripheral speed before entering sleep mode. For more information, refer to [Clock Controller (CMU)]

5.5.2 Turn off unused clock sources

The system clock of the chip has 6 clock sources:

- External high-speed oscillator (XTAL)
- External low-speed oscillator (XTAL32)
- MPLL Clock (MPLL)
- Internal High Speed Oscillator (HRC)
- Internal medium-rate oscillator (MRC)
- Internal low rate oscillator (LRC)

The SWDT has a separate dedicated internal low-speed oscillator (SWDTLRC); the RTC can choose either an external low-speed oscillator or an internal low-speed oscillator as the clock source. In addition, the chip is equipped with a UPLL clock source for I2S.

Each clock source can be individually turned off when not in use to reduce system power consumption. the HRC and PLL are equipped with separate power supply circuits, which can further reduce power consumption by cutting off the power to the HRC after the HRC is turned off by setting the PWC_PWRC1.VHRCSD bit, and by cutting off the power to the UPLL and the MPLL after both of them are turned off by setting the PWC_PWRC1.VPLLSD bit. VPLLSD bit to cut off the power supply of UPLL and MPLL to further reduce the power consumption.

For more information, refer to [Clock Controller (CMU)]

5.5.3 Function clock stop

The peripheral modules of the chip are equipped with a functional clock stop function. By placing the

registers in the corresponding positions, the modules that are not required to be used can be stopped from running, and the clocks of the corresponding modules can be stopped from being supplied to reduce power consumption. In the module stop state, the registers inside the module will maintain the state before stopping.

5.5.4 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 stop bit of the module, the clock is stopped for the RAMs that are not needed, thus reducing power consumption. By setting the power-down control bit of the module, the corresponding RAM module can be powered down to reduce power consumption. Table 5-8 shows the correspondence between the RAM module and the power-down control bits. The corresponding RAM can be powered down by setting the corresponding RAMPDCn (n=0-8), PW_PWRC0.RETRAMSD bits in the PWC_RAMPC0 register.

Table 5-8 RAM Module and RAM Power-Down Control Bits

RAM module	clarification	Brown-out control bit
SRAM1	0x2000_0000~0x2000_FFFF RAM for address space	PWC_RAMPC0.RAMPDC0
SRAM2	0x2001_0000~0x2001_FFFF RAM for address space	PWC_RAMPC0.RAMPDC1
SRAM3	0x2002_0000~0x2002_6FFF RAM for address space	PWC_RAMPC0.RAMPDC2
SRAMH	0x1FFF_8000~0x1FFF_FFFF RAM for address space	PWC_RAMPC0.RAMPDC3
USBFS	RAM for USBFS FIFO	PWC_RAMPC0.RAMPDC4
SDIO0RAM	RAM for SDIO0	PWC_RAMPC0.RAMPDC5
SDIO1RAM	RAM for SDIO1	PWC_RAMPC0.RAMPDC6
CANRAM	RAM for CAN	PWC_RAMPC0.RAMPDC7
CACHERAM	RAM for Cache	PWC_RAMPC0.RAMPDC8
Ret-SRAM	0x200F_0000~0x200F_0FFF RAM for address space	PWC_PWRC0.RETRAMSD

5.6 Register protection function

The register protection function is used to invalidate a register write operation to protect the register from being accidentally rewritten. Table 5-9 is a list of register protection bits and protected registers.

Table 5-9 Register Protection List

Protection Register Bits	protected register
pwc_fprc.fprcb0	CMU_XTALCFGR, CMU_XTALSTBCR, CMU_XTALCR, CMU_XTALSTDRCR, CMU_XTALSTDTSR, CMU_HRCTRM, CMU_HRCCR, CMU_MRCTR, CMU_MRCCR, CMU_PLLCFG, CMU_PLLCR, CMU_UPLLCFG, CMU_UPLLCR, CMU_OSCSTBSR, CMU_CKSWR, CMU_SCFGR, CMU_UFSCKCFG, CMU_TPIUCKCFG, CMU_MCO1CFG, CMU_MCO2CFG, CMU_XTAL32CR, CMU_xtalc32cfg, CMU_xtal32nfr. cmu_lrccr, cmu_lrctr, pwc_xtal32cs
pwc_fprc.fprcb1	pwc_pwrc0, pwc_pwrc1, pwc_pwrc2, pwc_pwrc3, pwc_pdwke0, pwc_pdwke1, pwc_pdwke2, pwc_pdwkes, pwc_pdwkf0, pwc_pdwkf1, pwc_pwcmr, cmu_pericksel, CMU_I2SCKSEL. pwc_mdswcr, pwc_stpmcr, pwc_rampc0, pwc_rampc0, rmu_rstf0
PWC_FPRC.FPRCB3	pwc_pvdcr0, pwc_pvdcr1, pwc_pvdfcr, pwc_pvdicr, PWC_PVDDSR

Protection Register Bits	protected register
PWC_FCG0PRC.	PWC_FCG0

5.7 Register Description

A list of registers is shown in Table 5-10.

Table 5-10 Register List

BASE ADDR:0x4005_4400				
register name	notation	offset address	bit width	reset value
Power Mode Control Register 0	PWC_PWRC0	0x00	8	0x00
Power Mode Control Register 1	PWC_PWRC1	0x01	8	0x00
Power Mode Control Register 2	PWC_PWRC2	0x02	8	0xFF
Power Mode Control Register 3	PWC_PWRC3	0x03	8	0x07
Wake on Power Down Enable Register 0	PWC_PDWKE0	0x04	8	0x00
Wake on Power Down Enable Register 1	PWC_PDWKE1	0x05	8	0x00
Wake on Power Down Enable Register 2	PWC_PDWKE2	0x06	8	0x00
Power-down wake-up event edge selection registers	PWC_PDWKES	0x07	8	0x00
Power-down wake-up flag register 0	PWC_PDWKF0	0x08	8	0x00
Power-down wake-up flag register 1	PWC_PDWKF1	0x09	8	0x00
Power monitoring registers	PWC_PWCMR	0x0A	8	0x00
Mode switching control register	PWC_MDSWCR	0x0F	8	0x00
PVD control register 0	PWC_PVDCR0	0x12	8	0x00
PVD control register 1	PWC_PVDCR1	0x13	8	0x00
PVD Filter Control Register	PWC_PVDFCR	0x14	8	0x11
PVD Level Control Register	PWC_PVDLCR	0x15	8	0x00
XTAL32 Current Control Register	PWC_XTAL32CS	0x2B	8	0x02
BASE ADDR:0x4005_4000				
register name	notation	offset address	bit width	reset value
STOP mode wake-up control registers	PWC_STPMCR	0x0C	16	0x4000
RAM Power Control Register 0	PWC_RAMPC0	0x14	32	0x0000_0000
RAM Running Condition Register	PWC_RAMOPM	0x18	16	0x8043
PVD Interrupt Control Register	PWC_PVDICR	0xE0	8	0x00
PVD Detection Status Register	PWC_PVDDSR	0xE1	8	0x11
Function Protection Control Register	PWC_FPRC	0x3FE	16	0x0000
BASE ADDR:0x4004_C400				
register name	notation	offset address	bit width	reset value
Wake-up Timer Control Register	PWC_WKTCR	0x00	16	0x0000
BASE ADDR:0x4004_8000				
register name	notation	offset address	bit width	reset value

Function clock control 0	PWC_FCG0	0x00	32	0xFFFF_FAEE
Function clock control 1	PWC_FCG1	0x04	32	0xFFFF_FFFF

Function Clock Control 2	PWC_FCG2	0x08	32	0xFFFF_FFFF
Function Clock Control 3	PWC_FCG3	0x0C	32	0xFFFF_FFFF
PWC_FCG0 Protection Control	PWC_FCG0PC	0x10	32	0x0000_0000

5.7.1 Power Mode Control Register 0 (PWC_PWRC0)

Reset value: 0x00

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

bit	Flags	Bit Name	Function
Read/Write			
b7	PWDN	Power-down mode control bits	0: Power-down mode disabled R/W 1: Power-down mode enable
b6	Reserved	-0" for reading, "0" for writing. writes "0".	Reads "0", R/W
b5-b4	IORTN[1:0]	IO hold control in power-down mode	00: IO hold state in power-down mode, hardware release 01: IO hold state after power-down wake-up 10: When RLDO is turned off and the chip enters power-down
b3	RETRAMSD	Hold RAM power-down control 0: Hold RAM power-down 1: Hold RAM does not power down	0: Ret-SRAM 1: Ret-SRAM power down
b1-b0	PDMDS[1:0]	Power-down mode control	0: Use of RLDO 00: Power-down mode 1 01: Power-down mode 2 10: Power-down mode 3 11: Power-down mode 4
b2	VVDRSD	Power-down mode control	R/W

5.7.2 Power Mode Control Register 1 (PWC_PWRC1)

Reset value: 0x00

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

Bit Flag	Bit Name	Function	Read/Write
b7-b6	STPDAS[1:0]	STOP mode LDO driver selection	00: Drive energy set when entering STOP mode in ultra-high-speed mode and high-speed mode Force 11: Drive capacity set when entering STOP mode in ultra-low speed mode 00\01: Disable setting.
b5-b2	Reserved	-	Reads "0000" and writes "0000".
b1	VHRCSD	HRC power off	0: HRC power enable 1: HRC power off When the HRC is not in use, turn off the power supply for the HRC after setting VHRCSD, into a HRC module
b0	VPLLSD	PLL power off	0: PLL power enable 1: PLL power off After both UPLL and MPPLL are turned off and waiting 50us, after setting VPLLSD Turning off the power supply for the PLL further reduces power consumption. VPLLSD clearing requires the To wait 25uS before starting the PLL module.

5.7.3 Power Mode Control Register 2 (PWC_PWRC2)

Reset value: 0xFF

b7	b6	b5	b4	b3	b2	b1	b0
-	-	DVS[1:0]	-	DDAS [3:0]	-	-	-
classifier for honorific people	marking	celebrity	functionality				fill out or in (information on a form)
b7	Reserved	-	Reads "1" and writes "1".				R/W
b6	Reserved	-	Reads "1" and writes "1".				R/W
b5-b4	DVS[1:0]	Voltage selection in action mode	00: Selection of ultra-high speed action mode voltage 01: Setting prohibition 10: Selection of ultra-low speed action voltage 11: Selection of high speed action mode voltage				R/W
b3-b0	DDAS [3:0]	Power Driver Selection	1111: Ultra-high-speed operation mode, high-speed operation mode drive capability selection 1110: Setting prohibition 1001: Setting prohibition				R/W

1000: Ultra-low speed operation mode drive capability selection
Other: Setting Prohibition

5.7.4 Power Mode Control Register 3 (PWC_PWRC3)

Reset value: 0x07

b7	b6	b5	b4	b3	b2	b1	b0
		-		PDTS	-	-	
clas sifie rfor hon orifi c peo ple	marking	celebrit y		functi onalit y		fill out or in (information on a form)	
b7-b3	Reserved	-		Reserved bit, write "00000"when writing.		R/W	
b2	PDTS Wake-Up Time Control Bit	Power-Down		0: When the total capacitance of VCAP_1/VCAP_2 is two 0.1uF or 1 At 0.22uF 1: When the total capacitance of VCAP_1/VCAP_2 is 2 0.047uF or 1 At 0.1uF		R/W	
b1	Reserved	-Reserved		Read "1", write "1".		R/W	
b0	Reserved	-Reserved		Read "1", write "1".		R/W	

5.7.5 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
classifi erfor honori fic peopl e	marking	Bit Name	Function	Read/Write			
b7	WKE13	WKUP1_3 wake-up event enable		0: WKUP1_3 wake-up event disabled R/W 1: WKUP1_3 wake-up event enable			
b6	WKE12	WKUP1_2 wakeup event enable		0: WKUP1_2 wake-up event disabled R/W 1: WKUP1_2 wake-up event enable			
b5	WKE11	WKUP1_1 wakeup event enable		0: WKUP1_1 wake-up event disabled R/W 1: WKUP1_1 wake-up event enable			
b4	WKE10	WKUP1_0 wakeup event enable		0: WKUP1_0 wake-up event disabled R/W 1: WKUP1_0 wake-up event enable			
b3	WKE03	WKUP0_3 wake-up event enable		0: WKUP0_3 wake-up event disabled R/W 1: WKUP0_3 wake-up event enable			
b2	WKE02	WKUP0_2 wake-up event enable		0: WKUP0_2 wake-up event disabled R/W 1: WKUP0_2 wake-up event enable			

b0

WKE00

WKUP0_0 wake-up event enable

0: WKUP0_0 wake-up event disabled

R/W 1: WKUP00 wake-up event enable

5.7.6 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
<hr/>							
classifier	marking	Bit Name Function Read/Write					
erfor							
honori							
fic							
peopl							
e							
b7	WKE33	WKUP3_3 wakeup event enable		0: WKUP3_3 wake-up event disabled R/W 1: WKUP3_3 wake-up event enable			
b6	WKE32	WKUP3_2 wakeup event enable		0: WKUP3_2 wake-up event disabled R/W 1: WKUP3_2 wake-up event enable			
b5	WKE31	WKUP3_1 wakeup event enable		0: WKUP3_1 wake-up event disabled R/W 1: WKUP3_1 wake-up event enable			
b4	WKE30	WKUP3_0 wakeup event enable		0: WKUP3_0 wake-up event disabled R/W 1: WKUP3_0 wake-up event enable			
b3	WKE23	WKUP2_3 wakeup event enable		0: WKUP2_3 wake-up event disabled R/W 1: WKUP2_3 wake-up event enable			
b2	WKE22	WKUP2_2 wakeup event enable		0: WKUP2_2 wake-up event disabled R/W 1: WKUP2_2 wake-up event enable			
b1	WKE21	WKUP2_1 wakeup event enable		0: WKUP2_1 wake-up event disabled R/W 1: WKUP2_1 wake-up event enable			
b0	WKE20	WKUP2_0 wakeup event enable		0: WKUP2_0 wake-up event disabled R/W 1: WKUP2_0 wake-up event enable			

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

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
WKTMWKE	-	RTCALMWKE	RTCPRDWKE	-	NMIWKE	PVD2WKE	PVD1WKE
<hr/>							
classifier	marking	Bit Name Function Read/Write					
for							
honori							
fic							
people							
b7	WKTMWKE	WKT M wake-up event enable		0: WKT M wake-up event disabled R/W 1: WKT M wake-up event enable			
b6	Reserved	-0" for reading, "0" for writing. writes "0".				Reads "0", R/W	
b5	RTCALMWKE	RTC alarm clock wake-up event enable		0: RTC alarm clock wake-up event disabled R/W 1: RTC wake-up event enabled			
b4	RTCPRDWKE	RTC cycle wakeup event enable		0: RTC cycle wake-up event disabled R/W 1: RTC cycle wake-up event enable			
b3	Reserved	-Reserve bit, write "0".		Reserved bit, write "0" when writing.		R/W	

b2	NMIWKE	NMI wakeup event enable	0: NMI wake-up event disabled R/W 1: NMI wake-up event enabled
b1	PVD2WKE	PVD2 wake-up event enable	0: PVD2 wake-up event disabled R/W 1: PVD2 wake-up event enable
b0	PVD1WKE	PVD1 wake-up event enable	0: PVD1 wake-up event disabled R/W 1: PVD1 wake-up event enable

5.7.8 Power-down wake-up event edge selection register (PWC_PDWKES)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
bit	Flags	Bit Name	Function				
b7	Reserved	-0" for reading, "0" for writing.					Reads "0", writes "0".
Read/Write		R/W					
b6	NMIEGS	NMI wake-up event edge selection edge					0: Falling
		R/W 1: Rising edge					
b5	VD2EGS	VD2 edge selection					0: VCC<VPVD2
		R/W 1: VCC> VPVD2					
b4	VD1EGS	VD1 edge selection					0: VCC<VPVD2
		R/W 1: VCC> VPVD2					
b3	WK3EGS	PTWK3 edge selection					0: Falling edge
		R/W 1: Rising edge					
b2	WK2EGS	PTWK2 edge selection					0: Falling edge
		R/W 1: Rising edge					
b1	WK1EGS	PTWK1 edge selection					0: Falling edge
		R/W 1: Rising edge					
b0	WK0EGS	PTWK0 edge selection					0: Falling edge
		R/W 1: Rising edge					

5.7.9 Power-down wake-up flag register 0 (PWC_PDWKF0)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	NMIWKF	PVD2WKF	PVD1WKF	PTWK3F	PTWK2F	PTWK1F	PTWK0F
classifier for honorific people	marking	celebrity	functionality				fill out or in (information on a form)
b7	Reserved	-	Reads "1" and writes "1".				R/W
b6	NMIWKF	NMI wake-up flag bit	0: No NMI pin wakeup event occurred 1: NMI pin wakeup event occurs After power down and wake up, you need to write zero to clear this bit.				R/W
b5	PVD2WKF	PVD2 wake-up flag bit	0: No PVD2 wakeup event occurred 1: PVD2 wake-up event occurs After power down and wake up, you need to write zero to clear this bit.				R/W
b4	PVD1WKF	PVD1 wake-up flag bit	0: No PVD1 wakeup event has occurred 1: PVD1 wake-up event occurs After power down and wake up, you need to write zero to clear this bit.				R/W
b3	PTWK3F	PTWK3 wake-up flag bit	0: No PTWK3 wakeup event occurred 1: PTWK3 wake-up event occurs After power down and wake up, you need to write zero to clear this bit.				R/W
b2	PTWK2F	PTWK2 wake-up flag bit	0: No PTWK2 wakeup event occurred 1: PTWK2 wakeup event occurs After power down and wake up, you need to write zero to clear this bit.				R/W
b1	PTWK1F	PTWK1 wake-up flag bit	0: No PTWK1 wakeup event occurred 1: PTWK1 wake-up event occurs After power down and wake up, you need to write zero to clear this bit.				R/W
b0	PTWK0F	PTWK0 wake-up flag bit	0: No PTWK0 wakeup event occurred 1: PTWK0 wake-up event occurs After power down and wake up, you need to write zero to clear this bit.				R/W

5.7.10 Power-down wake-up flag register 1 (PWC_PDWKF1)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
WKTMWKF	-	RTCALMWKF	RTCPRDWKF	-	-	-	
classifier marking celebrity functionality fill out or in for honorific people (information on a form)							
b7	WKTMWKF	WKTM wake-up flag bit	0: No WKTM wakeup event occurred 1: WKTM wake-up event occurs After power down and wake up, you need to write zero to clear this bit.			R/W	
b6	Reserved	-	Reserved bit, write "0"when writing.			R/W	
b5	RTCALMWKF	RTC alarm clock wake-up flag bit	0: No RTC alarm wakeup event occurred 1: RTC alarm clock wakeup event occurs After power down and wake up, you need to write zero to clear this bit.			R/W	
b4	RTCPRDWKF	RTC cycle wake-up flag bit	0: No RTC cycle wakeup event occurred 1: An RTC cycle wakeup event occurs After power down and wake up, you need to write zero to clear this bit.			R/W	
b3	Reserved	-	Reserved bit, write "0"when writing.			R/W	
b2	Reserved	-	Reserved bit, write "0"when writing.			R/W	
b1	Reserved	-	Reserved bit, write "0"when writing.			R/W	
b0	Reserved	-	Reserved bit, write "0"when writing.			R/W	

5.7.11 Power Monitoring Control Register (PWC_PWCMR)

Reset value:0x00

b7	b6	b5	b4	b3	b2	b1	b0
ADBUFE	-	-	-	-	-	-	-
classifier marking celebrity functionality fill out or in for honorific people (information on a form)							
b7	ADBUFE	ADBUF enable	When using AD to measure the internal voltage of the chip, you need to set this bit to 1 0: Invalid 1: Effective			R/W	
b6	Reserved	-	Reads "0", writes "0".			R/W	
b5	Reserved	-	Reads "0", writes "0".			R/W	
b4	Reserved	-	Reads "0" and writes "0".			R/W	
b2	Reserved	-	Reads "0" and writes "0".			R/W	
b1	Reserved	-	Reads "0" and writes "0".			R/W	

b0

Reserved

-

Reads "0" and writes "0".

R/W

5.7.12 Mode Switching Control Register (PWC_MDSWCR)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
MDSWC [7:0]							
classifier for honorific people	marking	celebrity	functionality			fill out or in (information on a form)	
b7-b0	MDSWC [7:0]	Mode switch enable	When performing action mode switching, after setting PWRC2, you need to set the MDSWC[7:0] is set to 0x10 to take effect. Only 0x10 can be written, other values are prohibited, and 0x00 is read.			R/W	

5.7.13 Function Clock Control 0 (PWC_FCG0)

Reset value:0xFFFF_FAEE

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
KEY	-	-	-	DCU4	DCU3	DCU2	DCU1	CRC	TRNG	HASH	AES	-	-	AOS	FCM
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DMA2	DMA1	-	-	-	SRAMRET	-	SRAM3	-	-	-	SRAM12	-	-	-	SRAMH

classifier for marking		Bit Name	Function	Read/Write
honorific	people			
b31	KEY	KEYSCAN Function Control	0: Keyboard scanning control module KEYSCAN function enabled 1: R/W KEYSCAN function disabled	
b30	Reserved	-1" for reading, "1" for writing. writes "1".	R/W	Reads "1",
b29	Reserved	-1" for reading, "1" for writing. writes "1".	R/W	Reads "1",
b28	Reserved	-1" for reading, "1" for writing. writes "1".	R/W	Reads "1",
b27	DCU4	DCU4 function control	0: Digital Computing Unit DCU3 function enable 1: R/W Digital Computing Unit DCU3 function disabled	
b26	DCU3	DCU3 function control	0: Digital Computing Unit DCU2 function enable 1: R/W Invalid function of Digital Computing Unit DCU2	R/W
b25	DCU2	DCU2 function control	0: Digital Computing Unit DCU1 function enabled 1: R/W Digital Computing Unit DCU1 function disabled	
b24	DCU1	DCU1 function control	0: Digital Computing Unit DCU0 function enabled 1: R/W Digital Computation Unit DCU0 function disabled	
b23	CRC	CRC function control	0: CRC function enable 1: R/W CRC function disabled	
b22	TRNG	TRNG function control	0: True Random Generator TRNG function in the CPM of the cryptographic co-processing module is enabled 1: R/W True Random Generator TRNG function in the CPM of the cryptographic co-processing module is disabled	R/W
b21	HASH	HASH Function Control	0: The HASH function of the Secure Hash Algorithm module in the CPM module is enabled. 1: Functions R/W Invalid function of Secure Hash Algorithm module in CPM, a cryptographic co-processing module.	R/W
b20	AES	AES Function Control	0: The AES function of the encryption/decryption algorithm processor in the CPM of the encryption co-processing module is enabled. 1: R/W The AES function of the encryption/decryption algorithm processor in the CPM of the cryptographic co- processing module is disabled.	
b19	Reserved	-1" for reading, "1" for writing. writes "1".	R/W	Reads "1",
b18	Reserved	-1" for reading, "1" for writing. writes "1".	R/W	Reads "1",

b17	AOS	Peripheral Circuit Trigger Function Control Peripheral circuit trigger function enable control R/W	0: 1:Peripheral circuit trigger function disabled
b16	FCM	FCM function control controller CMU R/W	0:Clock frequency measurement module FCM function enable in the clock 1: Invalid Clock Frequency Measurement Module FCM function in the Clock Controller CMU
b15	DMA2	DMA2 Function Control R/W	0: DMA controller DMA2 function enabled 1: DMA controller DMA2 function disabled
b14	DMA1	DMA1 Function Control R/W	0: DMA controller DMA1 function enable 1: Invalid DMA1 function of DMA controller
b13	Reserved	-1" for reading, "1" for writing. Reads "1", writes "1".	R/W
b12	Reserved	-1" for reading, "1" for writing. Reads "1", writes "1".	R/W
b11	Reserved	-1" for reading, "1" for writing. Reads "1", writes "1".	R/W
b10 built-in SRAM	SRAMRET	Ret_SRAM function control R/W	0: Ret_SRAM function enable in 1: Invalid Ret_SRAM function in built-in SRAM

b9	Reserved	-	Reads "1" and writes "1".	R/W 0: SRAM3
b8	SRAM3 function enable in built-in SRAM	ECCRAM function control R/W	1: Invalid SRAM3 function in built-in SRAM	
b7	Reserved	-1" for reading, "1" for writing.	Reads "1", writes "1".	R/W
b6	Reserved	-1" for reading, "1" for writing.	Reads "1", writes "1".	R/W
b5	Reserved	-1" for reading, "1" for writing.	Reads "1", writes "1".	R/W
b4	SRAM12	SRAM1/SRAM2 function containment	0: SRAM1 and SRAM2 functions in the internal SRAM are enabled. 1: SRAM1 and SRAM2 functions in built-in SRAM are invalid.	R/W
b3	Reserved	-	Reads "1" and writes "1".	R/W
b2	Reserved	-	Reads "1" and writes "1".	R/W
b1	Reserved	-	Reads "1" and writes "1".	R/W
b0	built-in SRAM	SRAMH	RAMHS function control 0: SRAMH function enable in 1: Invalid SRAMH function in built-in SRAM	R/W

5.7.14 Function Clock Control 1 (PWC_FCG1)

Reset value: 0xFFFF_FFFF

b31	b30	b29	b28	b27	b26	b25	b24
-	-	-	-	USART4	USART3	USART2	USART1
b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	SPI4	SPI3	SPI2	SPI1
b15	b14	b13	b12	b11	b10	b9	b8
I2S4	I2S3	I2S2	I2S1	SDIOC2	SDIOC1	-	USBFS
b7	b6	b5	b4	b3	b2	b1	b0
-	I2C3	I2C2	I2C1	QSPI	-	-	CAN

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31	Reserved	-	Reads "1" and writes "1".	R/W
b30	Reserved	-	Reads "1" and writes "1".	R/W
b29	Reserved	-	Reads "1" and writes "1".	R/W
b28	Reserved	-	Reads "1" and writes "1".	R/W
b27	USART4	USART4 Function Control USART4 Function Enable	0:Universal Synchronous Asynchronous Transceiver R/W 1: Universal Synchronous Asynchronous Transceiver USART4 function not available	
b26	USART3	USART3 Function Control	0: Universal Synchronous Asynchronous Transceiver USART3 function enabled R/W 1: Universal Synchronous Asynchronous Transceiver USART3 function disabled	
b25	USART2	USART2 Function Control	0: Universal Synchronous Asynchronous Transceiver USART2 function enabled R/W 1: Universal Synchronous Asynchronous Transceiver USART2 function disabled	
b24	USART1 function enable	USART1USART1 function control R/W	0: Universal synchronous asynchronous transceiver 1: Universal Synchronous Asynchronous Transceiver USART1 function is disabled	
b23	Reserved	-1" for reading, "1" for writing. Reads "1", writes "1".	R/W	
b22	Reserved	-1" for reading, "1" for writing. Reads "1", writes "1".	R/W	
b21	Reserved	-1" for reading, "1" for writing. Reads "1", writes "1".	R/W	
b20	Reserved	-1" for reading, "1" for writing. Reads "1", writes "1".	R/W	
b19	SPI4	SPI4 Function Control	0: Serial Peripheral Interface SPI4 function enable 1: Serial Peripheral Interface SPI4 function disabled	
b18	SPI3	SPI3 Function Control	0: Serial Peripheral Interface SPI3 Function Enable 1: Serial Peripheral Interface SPI3 function is disabled	R/W
b17	SPI2	SPI2 Function Control	0: Serial Peripheral Interface SPI2 function enable 1: Serial Peripheral Interface SPI2 function disabled	
b16	SPI1	SPI1 Function Control	0: Serial Peripheral Interface SPI1 Function Enable 1: Serial Peripheral Interface SPI1 function disabled	R/W

b15	I2S4	I2S4 Function Control	0:IC built-in audio bus module I2S4 function enable R/W 1: Built-in audio bus module I2S4 function of the integrated circuit is invalidated
b14	I2S3	I2S3 Function Control	0: I2S3 function of the integrated circuit's built-in audio bus module enabled R/W 1: I2S3 function of the integrated circuit's built-in audio bus module disabled
b13	I2S2	I2S2 Function Control audio bus module	0: I2S2 function enable of the integrated circuit's built-in audio bus module R/W 1: The built-in audio bus module I2S2 function of the integrated circuit is invalid.
b12	I2S1	I2S1 function control	0: I2S1 function of the built-in audio bus module of the IC is enabled R/W 1: I2S1 function of the built-in audio bus module of the IC is disabled
b11	SDIOC2	SDIOC2 function control	0:SDIOC2 function enable of SDIO controller R/W

			1: Invalid SDIOC2 function of SDIO controller
b10	SDIOC1	SDIOC1 Function Control	0: SDIOC1 function of SDIO controller enabled 1: SDIO controller SDIOC1 function R/W disabled
b9	Reserved	-1" for reading, "1" for writing. "1", writes "1".	Reads R/W
b8	USBFS	USBFS Function Control	0: USB2.0 full speed module USBFS function enable 1: USB2.0 full speed module USBFS R/W function disabled
b7	Reserved	-1" for reading, "1" for writing. "1", writes "1".	Reads R/W
b6	I2C3	I2C3 Function Control	0: I C bus I2C3 function enable 1: I C bus I2C3 function disabled R/W
b5	I2C2	I2C2 Function Control	0: I C bus I2C2 function enable 1: I C bus I2C2 function disabled R/W
b4	I2C1	I2C1 Function Control	0: I C bus I2C1 function enable 1: I C bus I2C1 function disabled R/W
b3	QSPI	QSPI Function Control	0: 4-wire serial peripheral interface QSPI function enabled 1: 4-wire serial peripheral interface QSPI R/W function disabled
b2	Reserved	-1" for reading, "1" for writing. "1", writes "1".	Reads R/W
b1	Reserved	-1" for reading, "1" for writing. "1", writes "1".	Reads R/W
b0	CAN	CAN function control	0: Controller LAN CAN function enabled 1: Controller LAN CAN function R/W disabled

5.7.15 Function Clock Control 2 (PWC_FCG2)

Reset value: 0xFFFF_FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17		
-	-	-	-	-	-	-	-	-	-	-	-	-	TIME R6_3	TIME R6_2	TIME R6_1	
b16	b15		b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1
EMB	-	-	-	-	TIME R4_3	TIME R4_2	TIME R4_1	TIME RA_6	TIME RA_5	TIME RA_4	TIME RA_3	TIME RA_2	TIME RA_1	TIME R0_2	TIME R0_1	
				b0												

class	marking	celebrity	functionality	fill out or in (information on a form)
ifier				
for				
hono				
rific				
peop				
le				
b31	Reserved	-	Reads "1" and writes "1".	R/W
b30	Reserved	-	Reads "1" and writes "1".	R/W
b29	Reserved	-	Reads "1" and writes "1".	R/W
b28	Reserved	-	Reads "1" and writes "1".	R/W
b27	Reserved	-	Reads "1" and writes "1".	R/W
b26	Reserved	-	Reads "1" and writes "1".	R/W
b25	Reserved	-	Reads "1" and writes "1".	R/W
b24	Reserved	-	Reads "1" and writes "1".	R/W
b23	Reserved	-	Reads "1" and writes "1".	R/W
b22	Reserved	-	Reads "1" and writes "1".	R/W
b21	Reserved	-	Reads "1" and writes "1".	R/W
b20	Reserved	-	Reads "1" and writes "1".	R/W
b19	Reserved	-	Reads "1" and writes "1".	R/W
b18	TIMER6_3	TIMER6_3 function control	0:TIMER6_3 function enable 1:TIMER6_3 function is invalidated	R/W
b17	TIMER6_2	TIMER6_2 function control	0:TIMER6_2 function enable 1:TIMER6_2 function is invalidated	R/W
b16	TIMER6_1	TIMER6_1 function control	0:TIMER6_1 function enable 1:TIMER6_1 function is invalidated	R/W
b15	EMB	EMB Function Control	O: Emergency Brake Module EMB function enabled 1: Emergency Brake Module EMB function is invalid	R/W
b14	Reserved	-1" for reading, "1" for writing.	Reads "1", writes "1".	R/W
b13	Reserved	-1" for reading, "1" for writing.	Reads "1", writes "1".	R/W
b12	Reserved	-1" for reading, "1" for writing.	Reads "1", writes "1".	R/W
b11	Reserved	-1" for reading, "1" for writing.	Reads "1", writes "1".	R/W
b10	TIMER4_3	TIMER4_3 function control	0:TIMER4_3 function enable 1:TIMER4_3 function is invalidated	R/W

b9	TIMER4_2	TIMER4_2 function control 0:TIMER4_2 function enable 1:TIMER4_2 function is invalidated	R/W
b8	TIMER4_1	TIMER4_1 function control 0:TIMER4_1 function enable 1:TIMER4_1 function is invalidated	R/W
b7	TIMERA_6	TIMERA_6 function control 0:TIMERA_6 function enable 1:TIMERA_6 function is invalidated	R/W
b6	TIMERA_5	TIMERA_5 function control 0:TIMERA_5 function enable 1:TIMERA_5 function is invalidated	R/W

b5	TIMERA_4	TIMERA_4 function control	0:TIMERA_4 function enable 1:TIMERA_4 function is invalidated	R/W
b4	TIMERA_3	TIMERA_3 function control	0:TIMERA_3 function enable 1:TIMERA_3 function is invalidated	R/W
b3	TIMERA_2	TIMERA_2 function control	0:TIMERA_2 function enable 1:TIMERA_2 function is invalidated	R/W
b2	TIMERA_1	TIMERA_1 function control	0:TIMERA_1 function enable 1:TIMERA_1 function is invalidated	R/W
b1	TIMER0_2	TIMER0_2 function control	0:TIMER0_2 function enable 1:TIMER0_2 function is not effective	R/W
b0	TIMER0_1	TIMER0_1 function control	0:TIMER0_1 function enable 1:TIMER0_1 function is not effective	R/W

5.7.16 Function Clock Control 3 (PWC_FCG3)

Reset value: 0xFFFF_FFFF

b31	b30	b29	b28	b27	b26	b25	b24
-	-	-	-	-	-	-	-
b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8
-	-	-	OTS	-	-	-	CMP
b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	ADC2	ADC1

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31	Reserved	-	Reads "1" and writes "1".	R/W
b30	Reserved	-	Reads "1" and writes "1".	R/W
b29	Reserved	-	Reads "1" and writes "1".	R/W
b28	Reserved	-	Reads "1" and writes "1".	R/W
b27	Reserved	-	Reads "1" and writes "1".	R/W
b26	Reserved	-	Reads "1" and writes "1".	R/W
b25	Reserved	-	Reads "1" and writes "1".	R/W
b24	Reserved	-	Reads "1" and writes "1".	R/W
b23	Reserved	-	Reads "1" and writes "1".	R/W
b22	Reserved	-	Reads "1" and writes "1".	R/W
b21	Reserved	-	Reads "1" and writes "1".	R/W
b20	Reserved	-	Reads "1" and writes "1".	R/W
b19	Reserved	-	Reads "1" and writes "1".	R/W
b18	Reserved	-	Reads "1" and writes "1".	R/W
b17	Reserved	-	Reads "1" and writes "1".	R/W
b16	Reserved	-	Reads "1" and writes "1".	R/W
b15	Reserved	-	Reads "1" and writes "1".	R/W
b14	Reserved	-	Reads "1" and writes "1".	R/W
b13	Reserved	-	Reads "1" and writes "1".	R/W
b12	OTS	OTS function control	0: Temperature sensor OTS function active 1: Invalid temperature sensor OTS function	R/W
b11	Reserved	-1" for reading, "1" for writing.	Reads "1", writes "1".	R/W
b10	Reserved	-1" for reading, "1" for writing.	Reads "1", writes "1".	R/W
b9	Reserved	-1" for reading, "1" for writing.	Reads "1", writes "1".	R/W
b8	CMP	CMP Function Control	0: Voltage comparator CMP function enable 1: Invalid voltage comparator CMP function	R/W
b7	Reserved	-	Reads "1" and writes "1".	R/W
b6	Reserved	-	Reads "1" and writes "1".	R/W
b5	Reserved	-	Reads "1" and writes "1".	R/W
b4	Reserved	-	Reads "1" and writes "1".	R/W

b3	Reserved	-	Reads "1" and writes "1".	R/W
b2	Reserved	-	Reads "1" and writes "1".	R/W
b1	ADC2	ADC2 function control	0: Analog-to-digital converter module ADC2 function enable	R/W

1: Analog-to-digital converter

module ADC2 function invalid	b0	ADC1ADC1 function control	R/W
0: ADC1 function of analog-to-digital converter module enabled			
<u>1: Invalid ADC1 function of analog-to-digital converter module</u>			

5.7.17 PWC_FCG0 Protection Control (PWC_FCG0PC)

Reset value:0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16															
FCG0PCWE [15:0]																														
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0															
-																														
PRT0																														
classifier for marking																														
honorific people																														
fill out or in (information on a form)																														
b31~b16	FCG0PCWE[15:0]										PWC_FCG0PC write enable																			
0xA5A5 while changing the value of PRT0 bit																														
R/W (usually used in the negative) have the possibility of																														
b15-b1	Reserved				-				Reads "0", writes "0".				R/W																	
PWC_FCG0 write enable control bit																														
b0	PRT0				guardianship				0:PWC_FCG0 write invalid 1:PWC_FCG0 write enable				R/W																	

5.7.18 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]																														
-																														
FPRCB3																														
FPRCB2																														
FPRCB1																														
FPRCB0																														
classifier for marking																														
honorific people																														
fill out or in (information on a form)																														
b15~b8	PWC_FPRCWE										PWC_FPRC register writes to 0xA5h while being able to update the PWC_FPRC value, otherwise for the lower 8 bits																			
enable																														
Invalid value on write. 0x00 on read.																														
b7	Reserved				-				Reserved bit, write "0" when writing.				R/W																	
b6	Reserved				-				Reserved bit, write "0" when writing.				R/W																	
b5	Reserved				-				Reserved bit, write "0" when writing.				R/W																	
b4	Reserved				-				Reserved bit, write "0" when writing.				R/W																	
Protect the register bits, and refer to Table 5-9 for the protection object.																														
b3	FPRCB3				FPRC bit 3				0:Write Protect 1:Write Enable				R/W																	
b2	Reserved				-				Reserved bit, write "0" when writing.				R/W																	
Protect the register bits and refer to Table 5-9 for the protection object.																														
b1	FPRCB1				FPRC bit 1				0:Write Protect 1:Write Enable				R/W																	
Protect the register bits and refer to Table 5-9 for the protection object.																														
b0	FPRCB0				FPRC bit 0				0:Write Protect				R/W																	

1:Write Enable

5.7.19 STOP Mode Control Register (PWC_STPMCR)

Reset value: 0x4000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
STOP	-	-	-	-	-	-	-	-	-	-	-	-	-	CKSMRC	FLNWT
classifier for honorific people	marking	celebrity												functionality fill out or in (information on a form) 0: STOP	
b15 mode invalid	STOP	STOP mode selection bit												1: STOP mode effective	
b14	Reserved	- Reads "1" and writes "1".												R/W	
b13-b2	Reserved	- Reads "0", writes "0".												R/W	
b1	CKSMRC	Clock switch to MRC option	0: Maintain system clock and frequency division before entering STOP mode 1: System clock switches to MRC, SCKCFGR registers when waking up in STOP mode The device is initialized												R/W
b0	FLASH stabilization when waking up in STOP mode												O: Wait for FLASH stabilization when waking up in STOP mode	R/W-	
			1: STOP mode wakeup without waiting for FLASH stabilization												

5.7.20 RAM Power Control Register 0 (PWC_RAMPC0)

Reset value: 0x0000_0000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	RAMPDC8
7	6	5	4	3	2	1	0
RAMPDC7	RAMPDC6	RAMPDC5	RAMPDC4	RAMPDC3	RAMPDC2	RAMPDC1	RAMPDC0

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b32-b23	Reserved	-	Reads "0", writes "0".	R/W
b24	Reserved	-	Reads "0" and writes "0".	R/W
b23	Reserved	-	Reads "0" and writes "0".	R/W
b22	Reserved	-	Reads "0" and writes "0".	R/W
b21	Reserved	-	Reads "0" and writes "0".	R/W
b20	Reserved	-	Reads "0", writes "0".	R/W
b19	Reserved	-	Reads "0", writes "0".	R/W
b18	Reserved	-	Reads "0" and writes "0".	R/W
b17	Reserved	-	Reads "0" and writes "0".	R/W
b16	Reserved	-	Reads "0" and writes "0".	R/W
b15-b9	Reserved	-	Reads "0", writes "0".	R/W
b8	RAMPDC8	RAM power-down control bit 8 RAM action for cache	0: 1: RAM power-down for cache R/W	0: RAM action
b7	RAMPDC7	RAM power-down control bit 7 for can	R/W 1: RAM power down for can	0: RAM action
b6	RAMPDC6	RAM power-down control bit 6 sdio1 actuates with RAM	0: 1: sdio1 power down with RAM R/W	0: RAM action
b5	RAMPDC5	RAM power-down control bit 5 sdio0 actuated with RAM	0: 1: sdio0 power down with RAM R/W	0: RAM action
b4	RAMPDC4	RAM power-down control bit 4 with usbfs	R/W 1: usbfs power down with RAM	0: RAM action
b3	RAMPDC3	RAM power-down control bit 3 0x1FFF_8000~0x1FFF_FFFF space RAM (SRAMH) operation	0: 1: 0x1FFF_8000~0x1FFF_FFFF space RAM (SRAMH) power down R/W	0: RAM action
b2	RAMPDC2	RAM power-down control bit 2 0x2002_0000~0x2002_6FFF space RAM (SRAM3) action	0: 1: 0x2002_0000~0x2002_6FFF space RAM (SRAM3) power down R/W	0: RAM action
b1	RAMPDC1	RAM power-down control bit 1 0x2001_0000~0x2001_FFFF space RAM (SRAM2) operation	0:	0: RAM action

R/W 1: 0x2001_0000~0x2001_FFFF space RAM (SRAM2) power down

b0	RAMPDC0	RAM power-down control bit 0 0x2000_0000~0x2000_FFFF space RAM (SRAM1) action 1: 0x2000_0000~0x2000_FFFF space RAM (SRAM1) power down	0:
----	---------	---	----

5.7.21 RAM Running Condition Register (PWC_RAMOPM)

Reset value: 0x8043

15	14	13	12	11	10	9	8
pwc_ramopm[15:8]							
7	6	5	4	3	2	1	0
PWC_RAMOPM[7:0]							
bit	Marker	Bit Name		Function			
b15-b0	PWC_RAMOPM[15:0]	The RAM action mode is selected as		When the read/write chip operates in the ultra-high-speed/high-speed operation mode, the PWC_RAMOPM is set to 0x8043.		R/W	
				PWC_RAMOPM is set to 0x9062 when the chip is operating in ultra-low-speed mode.			

5.7.22 XTAL32 controlled by current source (PWC_XTAL32CS)

Reset value: 0x02

b7	b6	b5	b4	b3	b2	b1	b0
CSDIS			-		-	-	
Bit Flag Bit Name Function Read/Write							
7	CSDIS	Current source null control	0: Current source active		1: Invalid current source		R/W
					When XTAL32/RTC/WTKM/Ret-SRAM etc. are not required to be used, the		
					to reset CSDIS to reduce power consumption.		
b6-b2	Reserved	-0" for reading, "0" for writing.		Reads "0", writes "0".		R/W	
b1	Reserved	-1" for reading, "1" for writing.		Reads "1", writes "1".		R/W	
b0	Reserved	-0" for reading, "0" for writing.		Reads "0", writes "0".		R/W	

5.7.23 Wake-up Timer Control Register (PWC_WKTCR)

Reset value: 0x0000

b15	b14-b13	b12	b11	-b11	b0
WKTCE	WKCKS[1:0]	WKOVF		WKTMCMP[11:0]	
Bit Flag	Bit Name	Function	Read/Write		
b15	WKTCE	WKT enable	0: WKT stop 1: WKT count		R/W
b14-b13	WKCKS[1:0]	WKT Clock Selection	00: 64Hz clock 01: XTAL32 10: LRC 11: Reservations		R/W
b12	WKOVF	Counter comparison result flag does not agree with WKTMCMP value	0: Counter 1: Counter is consistent with WKTMCMP value		R/W
b11-b0	WKTMCMP[11:0]	WKT Compare Bit	WKT counter comparison value		R/W

5.7.24 PVD Control Register 0 (PWC_PVDCR0)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	PVD2EN	PVD1EN	-	-	-	-	EXVCCINEN
classifier for marking celebrity functionality fill out or in (information on a form)							
b7	Reserved	-		Reads "0", writes "0".			R/W
b6	PVD2EN	Voltage detection 2 allowed		0: Voltage detection 2 circuit invalid 1: Voltage detection 2 circuit active			R/W
b5	PVD1EN	Voltage detection 1 allowed		0: Voltage detection 1 circuit invalid 1: Voltage detection 1 circuit active			R/W
b4	Reserved	-0" for reading, "0" for writing. writes "0".		Reads "0",			R/W
b3~b2	Reserved	-0" for reading, "0" for writing. writes "0".		Reads "0",			R/W
b1	Reserved	-0" for reading, "0" for writing. writes "0".		Reads "0",			R/W
b0	EXVCCINEN input disabled			0: External VCC voltage External VCC voltage input enable 1: External VCC voltage input valid			R/W

5.7.25 PVD Control Register 1 (PWC_PVDCR1)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	PVD2CMPOE	PVD2IRS	PVD2IRE	-	PVD1CMPOE	PVD1IRS	PVD1IRE
<hr/>							
Bit Flag	Bit Name	Function	Read/Write				
b7	Reserved	-Reserved		Read "0", write "0".		R/W	
b6		PVD2CMPOEPVD2 comparison result output enable of comparison result of comparator 2			0: Disable the output 1: Allow output of comparator 2 comparison result		R/W
b5	PVD2IRS	PVD2 Interrupt Reset Selection		0: VCC or external input voltage generates PVD2 interrupt when falling through VPVD2 1: VCC or external input voltage is falling through VPVD2 to generate PVD reset			
b4	PVD2IRE	PVD2 interrupt reset enable	R/W	Note: When the PVD1IRS bit is "1" or the PVD2IRS bit is "1", it is not possible to enter power-down mode. R/W can enter power-down mode, to enter power-down mode, the PVD1IRS bit must be set to "1" or "1". "0" and position PVD2IRS "0".		1: Allow	
b3	Reserved	-0" for reading, "0" for writing. writes "0".	R/W	Note: Please make sure that the PVD2EN bit is "1" and the PVD2CMPOE bit is "1". Write "1" to the PVD2IRE bit.		Reads "0",	
b2	PVD1CMPOE	PVD1 comparison result output enable output of comparison result of comparator 1		0: Disable 1: Allow output of comparator 1 comparison result		0: Disable R/W	
b1	PVD1IRS	PVD1 Interrupt Reset Selection		0: VCC generates a PVD1 interrupt as it falls past VPVD1 1: VCC is falling through VPVD1 to generate PVD1 reset			
b0	PVD1IRE	PVD1 interrupt reset enable	R/W	Note 1: When the PVD1IRS bit is "1" or the PVD2IRS bit is "1," the When the PVD1IRS bit is "1" or the PVD2IRS bit is "1", the R/W cannot enter the power-down mode. Set "0" and set PVD2IRS position "0".		0: Prohibited 1: Allow	
<hr/>							
<hr/>							

5.7.26 PVD Filter Control Register (PWC_PVDFCR)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	PVD2NFCKS[1:0]	PVD2NFDIS	-	PVD1NFCKS[1:0]	PVD1NFDIS		
<hr/>							
classifier for honorific people	marking	celebrity	functionality			fill out or in (information on a form)	
b7	Reserved	-	Reads "0", writes "0". 00: 0.25 LRC cycle 01: 0.5 LRC cycle 10: LRC's 1st crossover 11: LRC's 2-way			R/W	
b6~b5 PVD2NFCKS	PVD2 Digital Filter Sampling Clock Selection					R/W	
			Note: This bit can only be overridden if the PVD2NFDIS bit is "1".				
b4 Digital filter active	PVD2NFDISPVD2 digital filter shielding		0: 1: Invalid digital filter			R/W	
b3	Reserved	-Reserved	Reads "0", writes "0". 00: 0.25 LRC cycle 01: 0.5 LRC cycle 10: LRC's 1st crossover 11: LRC's 2-way			R/W	
b2~b1 PVD1NFCKS	PVD1 Digital Filter Sample Clock Selection					R/W	
			Note: This bit can only be overwritten if the PVD1NFDIS bit is "1".				
b0 Digital filter active	PVD1NFDISPVD1 Digital Filter Masking		0: 1: Invalid digital filter			R/W	

5.7.27 PVD Level Control Register (PWC_PVDLCR)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
-		PVD2LVL[2:0]		-		PVD1LVL[2:0]	
classifier for honorific people	marking	celebrity		functionality		fill out or in (information on a form)	
b7	Reserved	-		Reads "0", writes "0".		R/W	
				000: 2.1V 001: 2.3V 010: 2.5V 011: 2.6V 100: 2.7V			
b6~b4	PVD2LVL	PVD2 threshold voltage selection		101: 2.8V 110: 2.9V		R/W	
				111: 1.1V (only if PWC_PVDCR0.EXVCCINEN=1) (This value is valid in the case of the) Note: The above noted thresholds are the chip operating in high speed mode, ultra-low speed mode, Threshold when in stop mode, threshold when the chip is operating in ultra-high-speed mode Refer to Electrical Characteristics.			
b3	Reserved	-		Reads "0" and writes "0".		R/W	
				000: 2.0V 001: 2.1V 010: 2.3V 011: 2.5V 100: 2.6V			
b2~b0	PVD1LVL	PVD1 threshold voltage selection		101: 2.7V		R/W	
				110: 2.8V 111: 2.9V			
				Note: The above noted thresholds are the chip operating in high speed mode, ultra-low speed mode, Threshold when in stop mode, threshold when the chip is operating in ultra-high-speed mode Refer to Electrical Characteristics.			

5.7.28 PVD Interrupt Control Register (PWC_PVDICR)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	-	PVD2NMIS	-	-	-	-	PVD1NMIS

clas	mar	celebrity	functionality	fill
sifie	king			out
rfor				or in
hon				(info
orifi				rmat
c				ion
peo				on a
ple				form
)

b7	Reserved	-	Reads "0" and writes "0".	R/W
b6~b5	Reserved	-	Reads "0" and writes "0".	R/W

b4	PVD2NMIS	PVD2 interrupt type selection 0: PVD2 interrupt as non-maskable interrupt	R/W
----	----------	---	-----

b3	Reserved	-Reserved	Read "0", write "0".	R/W
b2~b1	Reserved	-0" for reading, "0" for writing.	Read "0", write "0".	R/W

b0	PVD1NMIS	PVD1 interrupt type selection 0: PVD1 interrupt as non-maskable interrupt	R/W
----	----------	---	-----

		1: PVD1 interrupt as maskable interrupt	
--	--	---	--

5.7.29 PVD Detection Status Register (PWC_PVDDSR)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	-	PVD2DETFLG	PVD2MON	-	-	PVD1DETFLG	PVD1MON
classifier for honorific people	marking	celebrity			functionality		
b7~b6	Reserved	-			Reads "0", writes "0".		
b5 detection flag bit	PVD2DETFLGPVD2			0: PVD2 does not detect VCC past VPVD2 1: PVD2 detects VCC passing through VPVD2 Write 0 to bit 4 after reading can clear this bit. Note: This flag bit is valid when the PVD2EN bit			
b4	PVD2MON	PVD2 monitor and detect flag bit clear			is "1" and the PVD2CMPOE bit is "1" 0: VCC ≤ VPVD2 or external input comparison voltage ≤ PVD2 internal reference voltage		
b3~b2	Reserved	-0" for reading, "0" for writing.			1: When PVD2 is invalid or VCC>VPVD2 or external input comparison voltage > PVD2 internal reference voltage Writing a 0 to this bit clears the PVD2DETFLG bit.		
"0".	R/W				Read "0", write		
b1 detection flag bit	PVD1DETFLGPVD1			0: PVD1 does not detect VCC passing through VPVD1 1: PVD1 detects VCC passing through VPVD1 Writing 0 to bit 0 after reading can clear this bit. Note: This flag bit is valid when the PVD1EN bit			
b0	PVD1MON	PVD1 monitoring and detection flag bit clear			is "1" and the PVD1CMPOE bit is "1" 0: VCC ≤ VPVD1 1: PVD1DETFLG bit is cleared when PVD1 is invalid or VCC>VPVD1 writes 0 to this bit.		

6 Initialization Configuration (ICG)

6.1 brief

After the chip reset is released, the hardware circuit will read the FLASH address 0x0000_0400~0x0000_041F and load the data into the initialization configuration register. Addresses 0x0000_0408~0x0000_040F and 0x0000_0418~0x0000_041F are reserved functions, please write all 1s to ensure the normal operation of the chip. User can modify the **Initial Config register** by programming or erasing sector 0. The register reset value is determined by the FLASH data.

The list of initialization configuration

register addresses is as follows:

ICG_BASE_ADDR: 0x0000_0400

Table 6-1 Register List

register name	notation	offset address	bit width	reset value
Initialize configuration register 0	ICG0	0x000	32	(botany) adventitious
Initialize Configuration Register 1	ICG1	0x004	32	(botany) adventitious
Initialize Configuration Register 2	ICG2	0x008	32	(botany) adventitious
Initialize Configuration Register 3	ICG3	0x00C	32	(botany) adventitious
Initialize Configuration Register 4	ICG4	0x010	32	(botany) adventitious
Initialize configuration register 5	ICG5	0x014	32	(botany) adventitious
Initialize Configuration Register 6	ICG6	0x018	32	(botany) adventitious
Initialize Configuration Register 7	ICG7	0x01C	32	(botany) adventitious

6.2 Register Description

6.2.1 Initialization Configuration Register 0 (ICG0)

Reset value: variable

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16																	
-	-	-	WDT SLP OFF	WDTWDPT[3:0]			WDTCKS [3:0]			WDTPERI [1: :0]		WDTI TS	WDTA UTS																			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																	
-	-	-	SWDT SLPO FF	SWDTWDPT[3:0]			SWDTCKS[3:0]			SWDTPERI[1:0]		SWDT ITS	SWDTA UTS																			
<hr/>																																
classifier for honorific people	marking			celebrity			functionality						fill out or in (information on a form)																			
b31~b29	Reserved			-			functional reservation						R																			
b28	WDTSLPOFF			WDT counting stops in sleep mode.			0: WDT counting does not stop in sleep mode 1: WDT counting stops in sleep mode						R																			
<hr/>																																
b27~b24	WDTWDPT[3:0] Count Value Hundred			Refresh Allowed Area			Allowable interval for WDT count value refresh 0000: 0% to 100% 0001: 0% to 25% 0010: 25% to 50% 0011: 0% to 50% 0100: 50% to 75% 0101: 0%~25%, 50%~75% 0110: 25% to 75%																									
<hr/>																																
<hr/>																																
<hr/>																																
<hr/>																																
b23~b20	WDTCKS[3:0]			WDT Count Clock			percentage						R																			
<hr/>																																
<hr/>																																
<hr/>																																
b19~b18	WDTPERI[1:0] cycle			WDT count overflow			Other values: reserved 00:256 cycles																									

b17	WDTITS	WDT interrupt selection	0: Interrupt request 1: Reset request	R
b16	WDTAUTS	WDT auto start	0: WDT starts automatically after reset (hardware start) 1: WDT stop state after reset	R
b15~b13	Reserved	-	Function Reserved Bits	R
b12	SWDTSLOFF	SWDT stops counting in Sleep and Stop modes.	0: SWDT does not stop counting in sleep,stop mode. 1: SWDT counting stop in sleep,stop mode	R
b11~b8	SWDTWDPT[3:0] Count Value Hundred	Refresh Allowed Area percentage	Allowable interval for SWDT count value refresh 0000: 0% to 100% 0001: 0% to 25% 0010: 25% to 50% 0011: 0% to 50% 0100: 50% to 75% 0101: 0%~25%,50%~75% 0110: 25% to 75% 0111: 0% to 75% 1000: 75% to 100% 1001: 0% to 25%, 75% to 100% 1010: 25% to 50%, 75% to 100% 1011: 0% to 50%, 75% to 100% 1100: 50% to 100% 1101: 0% to 25%,50% to 100% 1110: 25% to 100% 1111: 0% to 100% 0000: SWDTCLK 0100: SWDTCLK/16 0101: SWDTCLK/32 0110: SWDTCLK/64 0111: SWDTCLK/128 1000: SWDTCLK/256 1011: SWDTCLK/2048 Other values: reserved 00:256 cycles	R
b7~b4	SWDTCKS[3:0]	SWDT Count Clock	0100: SWDTCLK/16 0101: SWDTCLK/32 0110: SWDTCLK/64 0111: SWDTCLK/128 1000: SWDTCLK/256 1011: SWDTCLK/2048 Other values: reserved 00:256 cycles	R

b0

SWDTAUTS

SWDT auto start

10: SWDT stop state after reset (hardware start)

R

6.2.2 Initialization Configuration Register 1 (ICG1)

Reset value: variable

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
NMI ICG ENA	NFE N	NMI ENR	NMI TRG	SMPCLK[1:0]]	-	-	-	-	-	-	-	-	BOR DIS	BOR_lev[1: 0]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	HRC STO P	-	-	-	-	-	-	-	HRC FRE QSE L

classifier for marking		Bit Name	Function	Read/Write
honorific	people			
b31	NMIICGENA	NMI pin ICG setting enable ICG set enable		0: NMI pin R 1: NMI pin ICG setting disabled
b30	NFEN	NMI digital filter enable	0: Disable digital filter function 1: Licensed digital filter function	R
b29	NMIENR	NMI Pin Interrupt Selection NMI pin interrupt		0: Disable R 1: License NMI pin interrupt
b28	NMITRG	NMI pin edge trigger	0: falling edge 1: Rising edge	R
b27~b26 Selection	SMPCLK[1:0]	Filter Sample Clock	0 0: PCLK3 0 1: PCLK3/8 1 0: PCLK3/32 1 1: PCLK3/64	R
b25~b19	Reserved	-Function Reserved Bits		
Reserved Bit	R			Function
b18	BORDIS	BOR action selection	0: BOR action allowed after reset 1: Prohibit BOR action after reset	R
			00: 1.9v	

b17~b16	BOR_LEV[1:0]		R 10: 2.1v 11: 2.3v	
b15~b9	Reserved	-	Function Reserved Bits 0: HRC oscillation	www.xhsc.com.cn
b8	HRCSTOP	HRC oscillation stop bit	1: HRC stops	R
b7~b1	Reserved	-1" for reading and "1" for writing. "1", writes "1".	R	Reads
b0	HRCFREQSEL	HRC frequency selection	0: 20MHz	R

6.2.3 Initialize Configuration Register n(ICGn)n=2~3

Reset value: variable

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16					
ICGn [31:16]																				
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0					
ICGn[15:0]																				
Bit Flag	Bit Name Function Read/Write										Function Reserved Bit									
b31~b0	ICGn[31:0]				-															
User setting of all 1s is required to ensure proper chip operation.																				

6.2.4 Initialization Configuration Register 4 (ICG4)

Reset value: variable

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
ICG4 [31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ICG4 [15:0]															
classifier for honorific people	marking				celebrity				functionality				fill out or in (information on a form)		
b31~b0	ICG4 [31:0]				Initialization Configuration 4				Chip data security protection level 1 enable configuration.				R		
Data security protection level 1 is enabled when the data is 0xAF180402. Data security protection level 1 is invalid when the data is any other value.															

6.2.5 Initialization Configuration Register 5 (ICG5)

Reset value: variable

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
ICG5 [31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ICG5 [15:0]															
classifier for honorific people	marking				celebrity				functionality				fill out or in (information on a form)		
b31~b0	ICG5 [31:0]				Initialization Configuration 5				Chip data security protection level 2 enable configuration.				R		
Data security protection level 2 is enabled when the data is 0xA85173AE. Data security protection level 2 is not valid when the															

data is any other value.

6.2.6 Initialize Configuration Register n(ICGn) n=6~7

Reset value: variable

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16												
ICGn [31:16]																											
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0												
ICGn[15:0]																											
<hr/>																											
Bit Flag	Bit Name	Function	Read/Write																								
b31~b0	ICGn[31:0]	-		Function Reserved Bits																							
User setting of all 1s is required to ensure proper chip operation.																											
<hr/>																											

7 Embedded FLASH (EFM)

7.1 summary

The FLASH interface provides access to FLASH via the FLASH ICODE, DCODE and MCODE buses. The interface performs programming, sector erase and full erase operations on the FLASH, and accelerates code execution through instruction prefetching and caching mechanisms.

7.2 Main characteristics

- Maximum 512KByte FLASH space
- ICODE Bus 16Byte Prefetched Value
- Shared 64 caches (1KByte) on ICODE and DCODE buses
- Provides 960Byte One-Time Programming Area (OTP)
- Supports low-power read operations
- Support for Guided Exchange Function
- Support for data security protection

7.3 Embedded FLASH

FLASH has the following key features:

- The capacity is up to 512 KBytes (of which 32 Bytes are functionally reserved) divided into 64 sectors of 8 KBytes each.
- The OTP (One Time Program) area is 1020 Bytes in total, which is divided into a data area of 960 Bytes and a latch area of 60 Bytes.
- 128-bit wide data read.
- The programming unit is 4Bytes and the erasing unit is 8KBytes. in the 512KB product, the FLASH address structure is shown below.

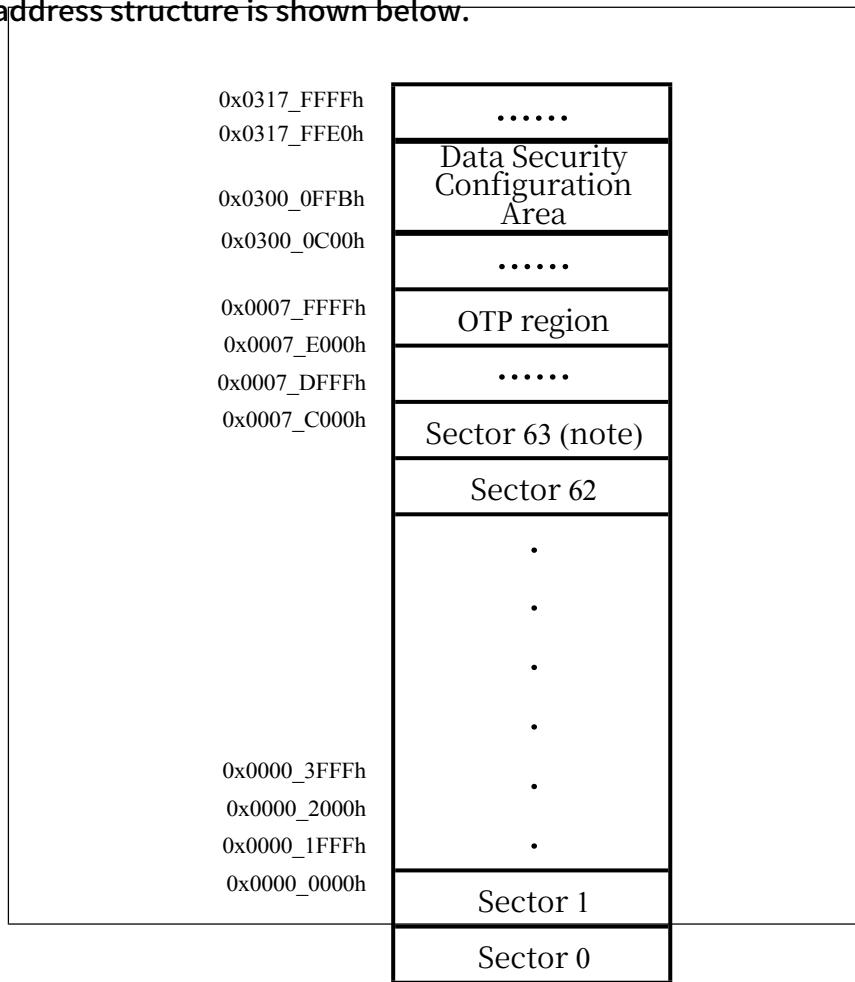


Figure 7-1 FLASH Address Structure (512KB Products)

Attention:

ion:

- Addresses 0x0007_FFE0~0x0007_FFFF in sector 63 are functionally reserved addresses; FLASH data will not be changed when programming, sector erase, or full erase is performed on these 32Bytes addresses, and the data read from these

In the 256KB product, the FLASH address structure is shown below.

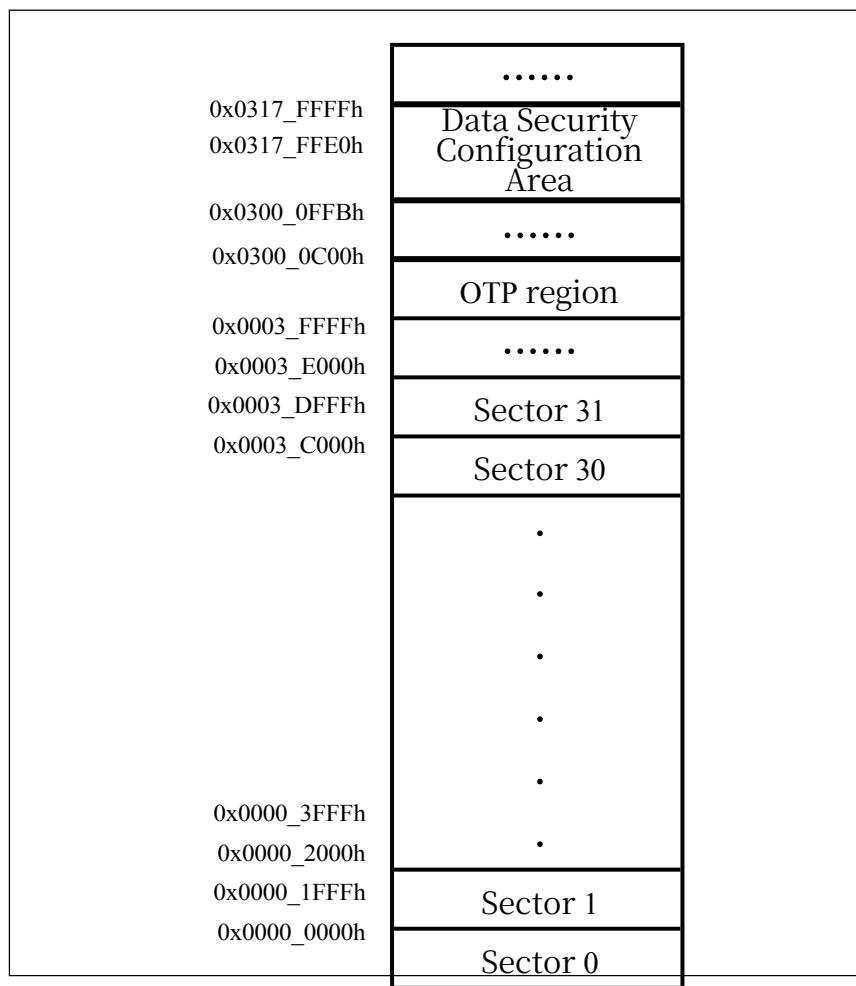


Figure 7-2 FLASH Address Structure (256KB Products)

7.4 readout

7.4.1 Relationship between CPU Clock and FLASH Read Time

To read FLASH data correctly, users need to set the correct number of wait cycles (FLWT[3:0]) in the FLASH read mode register (EFM_FRMC) according to the CPU action frequency.

After system reset, the CPU clock source is MRC(8MHz) and the FLASH read wait cycle is 0. It is recommended that the user follow the steps below to modify the CPU master frequency and FLASH read wait cycle bits. Please refer to Table 7-1 for the number of wait cycles.

CPU frequency increase step:

1. Write the new read wait cycle set value (FLWT[3:0]) to register EFM_FRMC.
2. Read register EFM_FRMC to check if the new wait cycle is set successfully.
3. The CPU clock frequency is boosted 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 register CMU_CKSWR or CMU_SCFGR to check if 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 register CMU_CKSWR or CMU_SCFGR to check if the new setting is successful.
3. Write the new read wait cycle set value (FLWT[3:0]) to register EFM_FRMC.
4. Read register EFM_FRMC to check if the new wait cycle is set successfully.

7.4.2 FLASH Low Power Read

When the CPU clock frequency is lower than 2MHz, user can set the register bit EFM_FRMC.SLPMD to enter the low power read mode to reduce the chip current. After entering the low power read mode, the programming and erasing operation of FLASH will be ignored, and the FLASH will be erased and written (programmed and erased, the same below) in this mode, with the status bit EFM_FSR.COLERR set.

Go to the Ultra Low Power Read step:

1. Write the new read wait cycle set value (FLWT[3:0]) to register EFM_FRMC according to the recommended value for the ultra-low power read mode according to Table 7-1.
2. Set register EFM_FRMC.SLPMD.

Exits the Ultra Low Power Read step:

1. Register bit EFM_FRMC.SLPMD cleared.
2. Write the new Read Insertion Waiting Cycle Setting Value (FLWT[3:0]) to register EFM_FRMC at the recommended value for Normal Read Mode according to Table 7-1.

Table 7-1 Comparison of CPU Clock Frequency and FLASH Read Wait Cycles

CPU clock frequency (HCLK)	FRMC register bits FLWT[3:0] set	
	Normal read mode (SLPMD=0)	Ultra low power read mode (SLPMD=1)
168MHz<FHCLK≤200MHz	FLWT[3:0]=4'b0101 Insert 5 wait read cycles	unsupported
132MHz<FHCLK≤168MHz	FLWT[3:0]=4'b0100 Insert 4 wait read cycles	unsupported
99MHz<FHCLK≤132MHz	FLWT[3:0]=4'b0011 Insert 3 wait read cycles	unsupported
66MHz<FHCLK≤99MHz	FLWT[3:0]=4'b0010 Insert 2 wait read cycles	unsupported
33MHz<FHCLK≤66MHz	FLWT[3:0]=4'b0001 Insert 1 wait read cycle	unsupported
2MHz<FHCLK≤33MHz	FLWT[3:0]=4'b0000 no waiting to read	unsupported
1.876MHz<FHCLK≤2MHz	ibid	FLWT[3:0]=4'b1111 Insert 15 wait read cycles
1.752MHz<FHCLK≤1.876MHz	ibid	FLWT[3:0]=4'b1110 Insert 14 wait read cycles
1.628MHz<FHCLK≤1.752MHz	ibid	FLWT[3:0]=4'b1101 Insert 13 wait read cycles
1.504MHz<FHCLK≤1.628MHz	ibid	FLWT[3:0]=4'b1100 Insert 12 wait read cycles
1.38MHz<FHCLK≤1.504MHz	ibid	FLWT[3:0]=4'b1011 Insert 11 wait read cycles
1.256MHz<FHCLK≤1.38MHz	ibid	FLWT[3:0]=4'b1010 Insert 10 wait read cycles
1.132MHz<FHCLK≤1.256MHz	ibid	FLWT[3:0]=4'b1001 Insert 9 wait read cycles
1.008MHz<FHCLK≤1.256MHz	ibid	FLWT[3:0]=4'b1000 Insert 8 wait read cycles
884KHz<FHCLK≤1.008MHz	ibid	FLWT[3:0]=4'b0111 Insert 7 wait read cycles
760KHz<FHCLK≤884KHz	ibid	FLWT[3:0]=4'b0110 Insert 6 wait read cycles
636KHz<FHCLK≤760KHz	ibid	FLWT[3:0]=4'b0101 Insert 5 wait read cycles
512KHz<FHCLK≤636KHz	ibid	FLWT[3:0]=4'b0100 Insert 4 wait read cycles

CPU clock frequency (HCLK)	FRMC register bits FLWT[3:0] set	
	Normal read mode (SLPMD=0)	Ultra low power read mode (SLPMD=1)
388KHz<FHCLK≤512KHz	ibid	FLWT[3:0]=4'b0011 Insert 3 wait read cycles
264KHz<FHCLK≤388KHz	ibid	FLWT[3:0]=4'b0010 Insert 2 wait read cycles
140KHz<FHCLK≤264KHz	ibid	FLWT[3:0]=4'b0001 Insert 1 wait read cycle
FHCLK≤140KHz	ibid	FLWT[3:0]=4'b0000 no waiting to read

7.5 FLASH Read Acceleration Cache

Each FLASH read operation is a 128-bit read, and the data is sent to the CPU and also stored in the buffer memory. The 128-bit data can be four lines of 32-bit instructions or eight lines of 16-bit instructions, depending on the program burned into the FLASH.

In order to read FLASH data quickly, the FLASH controller is configured with a read acceleration cache that optimizes the read wait cycle. In order to take advantage of the processor performance, the gas pedal saves the ICODE, DCODE bus access data to the FLASH into the cache registers, thus increasing the program execution speed.

The system provides 1KBytes of space as cache memory, which can effectively minimize the time loss due to instruction jumps. The cache function is enabled by the cache enable (CACHE) bit 1 in the EFM_FRMC register. Whenever there is an instruction or data miss (i.e., the requested instruction or data does not exist in the currently used instruction line or cache memory), the system copies the newly read data line (128-bit) into the cache memory. If the instruction or data requested by the CPU already exists in the cache, it is fetched immediately without any delay. When the cache memory is full, the LRU (Least Recently Used) policy is used to determine the data to be replaced in the cache memory.

The number of read FLASH cycles changes when the CPU reads instructions or data and the FLASH address hits in the buffer, cache, refer to Table 7-2 for details.

Table 7-2 FLASH Actual Read Cycles

EFM_FRMC. FLWT[3:0] setting	Cache not enabled (EFM_FRMC.CACHE=0)		Cache Enable (EFM_FRMC.CACHE=1)	
	Buffering, Cache Hit	Buffering, cache misses	Buffering, Cache Hit	Buffering, cache misses
0	1	1	1	1
1	1	2	1	2
2	1	3	1	3
3	1	4	1	5
4	1	5	1	6
N (N>4)	1	N+1	1	N+2

7.6 FLASH Programming and Erase Operations

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

FLASH programming unit is 4Bytes, the last bit of the programmed address must be aligned with 4 (last bit address: 0x0, 0x4, 0x8, 0xC), duplicate programming will not ensure the correctness of the programming. The unit of FLASH sector erase is 8KBytes. Please disable the cache before FLASH erase programming. The following describes the setting procedures for programming and erasing operations.

7.6.1 Single programming without readback mode

The procedure for setting the single programmed no readback mode is as follows:

- 1) Unprotect the registers of FLASH. (EFM_FAPRT writes 0x0123 first, then 0x3210)
- 2) Sets the programming, erase and write mode license. (EFM_FWMC.PEMODE=1)
- 3) Sets the single programming mode. (EFM_FWMC.PEMODE[2:0]=001)
- 4) Writes 32-bit data to the programmed address.
- 5) Wait for FLASH to be in idle state. (EFM_FSR.RDY=1)
- 6) Read the programmed address value to determine if it matches the written value;
Consistency indicates successful programming; inconsistency indicates that the FLASH address has been corrupted and is permanently discarded.
- 7) Clears the programming end flag bit. (EFM_FSR.OPTEND)

7.6.2 Single Programmable Readback Mode

The single programmed readback mode is to read the programmed address since the end of programming and compare it with the written data, and output the judgment agreement flag bit EFM_FSR.

PGMISMTCH.

The procedure for setting the single programmed readback mode is as follows:

- 1) Unprotect the registers of FLASH. (EFM_FAPRT writes 0x0123 first, then 0x3210)
- 2) Sets the programming, erase and write mode license. (EFM_FWMC.PEMODE=1)
- 3) Sets the single programmed readback mode. (EFM_FWMC.PEMODE[2:0]=010)
- 4) Writes 32-bit data to the programmed address.
- 5) Wait for FLASH to be in idle state. (EFM_FSR.RDY=1)
- 6) Judgement Programming Self-Read Result Flag Bit. (EFM_FSR.PGMISMTCH) A value of 0 indicates successful programming; a value of 1 indicates that the FLASH address has been corrupted and is permanently discarded.
- 7) Clears the programming end flag bit. (EFM_FSR.OPTEND)

7.6.3 Continuous Programming Operation

Continuous programming mode is recommended when programming FLASH addresses in succession. Sequential programming mode can save more than 50% of time than single programming mode. In continuous programming mode, the interval between successive programming writes must be less than 16 us. the procedure for setting up continuous programming operation is as follows:

- 1) Unprotect the registers of FLASH. (EFM_FAPRT writes 0x0123 first, then 0x3210)
- 2) Sets the Programming, Erase and Write mode license. (EFM_FWMC.PEMODE=1)
- 3) Sets the continuous programming mode. (EFM_FWMC.PEMOD[2:0]=011)
- 4) Send steps 8) 9) 10) 11) 12) 13) 14) to an area other than FLASH.
- 5) Jump to transfer to step 4) destination address.
- 6) Reads the programmed address and determines if it matches the written value.
- 7) Consistency indicates successful programming; inconsistency indicates that the FLASH address has been corrupted and is permanently discarded.
- 8) Write 32-bit data to the programmed address.
- 9) Wait for the operation end flag bit (EFM_FSR.OPTEND) to be set.
- 10) Clear the end-of-operation flag bit until the flag bit EFM_FSR.OPTEND is read as 0.
- 11) Repeat 8), 9), 10) until all data is written.
- 12) Modify the Erase Mode Control Register to discontinuous programming mode.
- 13) Wait for FLASH to be in idle state. (EFM_FSR.RDY=1)
- 14) Jumps back to the main program.

Attention:

-If a read operation of FLASH occurs during continuous programming of FLASH, an indefinite value will be read. The read operation will set the read conflict bit EFM_FSR.COLERR, which should be cleared by setting EFM_FSCLR register.

7.6.4 erase operation

EFM provides two types of erase methods: sector erase and full erase. After sector erase operation, the address (8KBytes space) data in the sector will be refreshed to all 1s; after full erase operation, all addresses (except OTP) in the whole FLASH area will be refreshed to all 1s. The setting procedure of sector erase and full erase operation is as follows:

- 1) Unprotect the registers of FLASH. (EFM_FAPRT writes 0x0123 first, then 0x3210)
- 2) Sets the programming, erase and write mode license. (EFM_FWMC.PEMODE=1)
- 3) Set the erase mode. (Sector Erase EFM_FWMC.PEMOD[2:0]=100, Full Erase EFM_FWMC.PEMOD[2:0]=101)

-
- 4) Write a 32-bit arbitrary value to any address in the sector to be erased (addresses must be aligned with 4).
 - 5) Write 32-bit arbitrary value to any FLASH address (address must be aligned with 4) during full erase.
 - 6) Wait for FLASH to be in idle state. (EFM_FSR.RDY=1)
 - 7) Clear the end-of-erase flag bit. (EFM_FSR.OPTEND)

7.6.5 Data security protection

This product provides 3 levels of protection for FLASH data to prevent untrusted users from reading and tampering with the FLASH through the debug interface (JTAG and SWD interfaces), ISP interface (In System Program) and the test interface.

Protection level 0: no protection

The debug, ISP and test interfaces can access (read and rewrite) MCU

resources, including FLASH data. Protection level 1:

FLASH address 0x0000_0410~0x0000_0413 is programmed to write data 0xAF180402, while address 0x0317_FFE0~0x0317_FFF7 is programmed to write 96-bit password, and address 0x0317_FFE0~0x0317_FFEB is programmed to write 96-bit 0, with protection level 1 enabled.

Protection level 1 When enabled

- The debug interface is turned off and the ISP interface and test interface cannot access the FLASH data.
- Sector 0, Sector 1, and Sector 63 cannot be programmed and erased by the user program.
- The data at addresses 0x0317_FFE0~0x0317_FFF7 cannot be read.

After activating protection level 1, you can revert to protection level 0 by password authentication method and full erase method, and consult the sales window for password authentication method and full erase method.

Protection level 2:

FLASH address 0x0000_0414~0x0000_0417 is programmed to write data 0xA85173AE, while address 0x0317_FFE0~0x0317_FFEB is programmed to write 96-bit 0 with protection level 2 enabled. When protection level 2 is enabled

- The debug interface is turned off and the ISP interface and test interface cannot access the FLASH data.
- Sector 0, Sector 1, and Sector 63 cannot be programmed and erased by the user program.
- The data at addresses 0x0317_FFE0 to 0x0317_FFF7 cannot be read. When protection level 2 is activated, it can be reverted to protection level 0 by full erase.

Protection level 1 and protection level 2 can be enabled separately or simultaneously. When both are enabled, the protection measures are superimposed.

7.6.6 bus hold function

BUSHLDCTL bit can be set to hold or release the bus during FLASH programming and erase period, the control bit must be set to 0 when the FLASH programming and erase instruction is executed on FLASH, and the control bit can be set freely according to the need when the erase instruction is executed in the space other than FLASH (e.g., high-speed RAM). When the erase instruction is executed in a space other than FLASH (e.g., high-speed RAM), the control bit can be freely set as needed.

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

7.6.7 FLASH Erase, Programming Window Protection

Window protection is provided for FLASH, only in the allowable area FLASH can be erased by sector and programmed, otherwise erase error interrupt occurs. When programming and sector erase operation, the hardware circuit will pre-determine whether it is in the allowed area or not, and the full erase mode is not restricted by the window protection. The start position and end position of the window are set by registers EFM_FPMTEW, EFM_FPMTSW.

The specific protection functions are as follows:

- Register EFM_FPMTEW = Register EFM_FPMTSW The entire FLASH area is erasable and programmable.
- Register EFM_FPMTEW > Register EFM_FPMTSW erasable, programming area in between.
- Register EFM_FPMTEW < Register EFM_FPMTSW The entire FLASH area is not erasable, programmed.

7.6.8 disruptions

The EFM module has three interrupts, PE (Program/Erase) Error Interrupt, Read/Write Conflict Interrupt, and End of Operation Interrupt.

After the PE error interrupt is set in FLASH, FLASH cannot be programmed/erased/fully erased. The flag bit must be cleared before programming/erasing/full erasing the FLASH again.

1. PE error interrupt EFM_PEERR:

Placement:

- The programmed address is not 4-aligned or the data size is not 32 bits (PGSZERR=1).
- Performs programming of the address within the FLASH window protection, sector erase operation (PEPRERR=1).
- Performs a write operation to FLASH when the erase mode is not set (PEWERR=1).
- The programmed address self-read value does not match the written value when in single programmed readback mode (PGMISMTCH=1).

Zeroing:

Register EFM_FSCLR Corresponding Flag Clear Bit Write 1, Status Bit Clear Zero.

2. FLASH Read/Write Conflict Interrupt EFM_COLERR:

Placement:

- FLASH read operation occurs in FLASH sequential programming mode.
- FLASH read/write operation occurs during FLASH stop mode.
- FLASH programming, FLASH read/write operation occurs before erase is completed.
- Write operations to FLASH occur in low-power read mode.

Zeroing:

Register EFM_FSCLR corresponds to clear position 1, status bit clear.

3. End of operation interrupt EFM_OPTEND:**Placement:**

- Programming Mode: End of individual address programming.
- Erase Mode: Sector Erase, End of Full Erase.

Zeroing:

Register EFM_FSCLR corresponds to clear position 1, status bit clear.

7.7 One-time programmable bytes

The following table shows the one-time programmable address composition of the OTP area, which is divided into a 960Bytes data area and a 60Bytes latch area.

Table 7-3 OTP Address Composition

block name	OTP Data Block Address	OTP Locked Address
Block 0	0x0300_0C00~0x0300_0C3F	0x0300_0FC0~0x0300_0FC3
Block 1	0x0300_0C40~0x0300_0C7F	0x0300_0FC4~0x0300_0FC7
Block 2	0x0300_0C80~0x0300_0CBF	0x0300_0FC8~0x0300_0FCB
Block 3	0x0300_0CC0~0x0300_0CFF	0x0300_0FCC~0x0300_0FCF
Block 4	0x0300_0D00~0x0300_0D3F	0x0300_0FD0~0x0300_0FD3
Block 5	0x0300_0D40~0x0300_0D7F	0x0300_0FD4~0x0300_0FD7
Block 6	0x0300_0D80~0x0300_0DBF	0x0300_0FD8~0x0300_0FD8
Block 7	0x0300_0DC0~0x0300_0DFF	0x0300_0FDC~0x0300_0FDF
Block 8	0x0300_0E00~0x0300_0E3F	0x0300_0FE0~0x0300_0FE3
9 pieces	0x0300_0E40~0x0300_0E7F	0x0300_0FE4~0x0300_0FE7
Block 10	0x0300_0E80~0x0300_0EBF	0x0300_0FE8~0x0300_0FEB
Block 11	0x0300_0EC0~0x0300_0EFF	0x0300_0FEC~0x0300_0FEF
Block 12	0x0300_0F00~0x0300_0F3F	0x0300_0FF0~0x0300_0FF3
Block 13	0x0300_0F40~0x0300_0F7F	0x0300_0FF4~0x0300_0FF7
Block 14	0x0300_0F80~0x0300_0FBF	0x0300_0FF8~0x0300_0FFB

The OTP area is divided into 15 64-byte data blocks, each of which corresponds to a 4-bytes latch address. The latch address is used to latch the corresponding data block. When the latch address data is all 1s, the corresponding OTP area data block is programmable; when

the latch address data is all 0s, the corresponding OTP area data is not programmable. All OTP data blocks and latched addresses cannot be erased.

7.8 bootstrap exchange

If users want to upgrade the boot program, they need to erase and write sector 0 (0x0000_0000~0x0000_1FFF), and if the erase encounters unexpected accidents (e.g. reset, power down), it may cause the whole chip not to boot normally. For 512KB products, EFM provides a boot sector swap function (256KB products do not have this function). Before erasing sector 0, the new boot program is written to sector 1 (0x0000_2000~0x0000_3FFF), and then the EFM address 0x0007_FFDC is programmed with the data 0xFFFF_4321. After the MCU reset, the CPU starts the new boot program from sector 1, and then erases sector 0 and reprograms the new user program. The user program is reprogrammed by erasing sector 0 at this point.

Refer to Figure 7-3 for startup exchange operation.

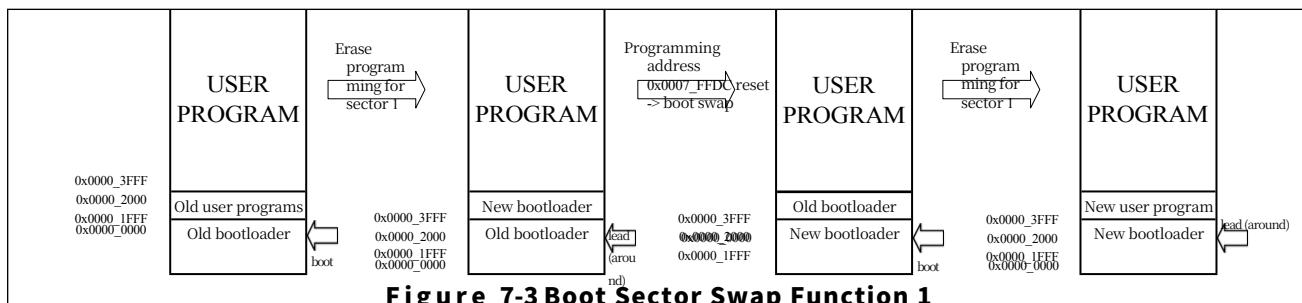


Figure 7-3 Boot Sector Swap Function 1

When users need to upgrade the bootloader again, since the address 0x0007_FFDC where the boot sector swap information is stored has been programmed (users can determine whether the boot swap function has been used by reading the FLASH address or EFM_FSWP register), it is necessary to perform a sector erase on sector 63 to upgrade the bootloader again. Before erasing sector 0, write the new boot program into sector 1 in advance, and then erase sector 63. After MCU reset, the CPU will start the new boot program from sector 1, and then erase sector 0 to reprogram the new user program. The operation flow is shown in Figure 7-4.

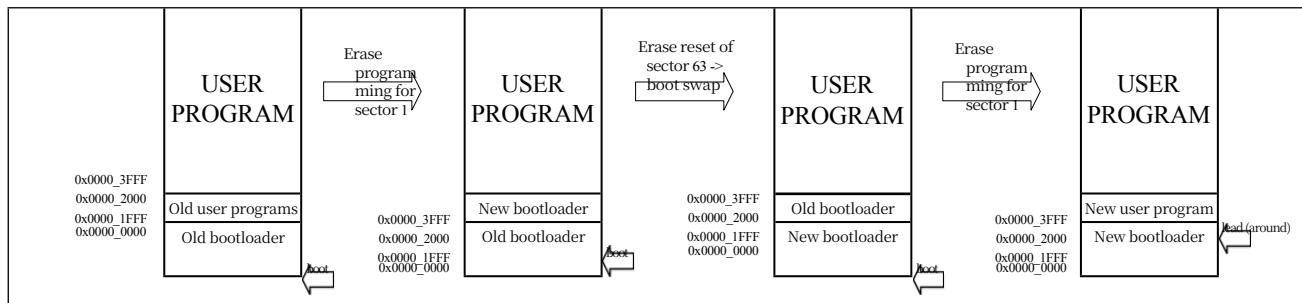


Figure 7-4 Starting the Switching Function 2

7.9 Register Description

EFM_BASE_ADDR: 0x4001_0400

Table 7-4 Register List

Register Description	register name	offset	bit width	reset value
FLASH Access Protection Register	EFM_FAPRT	0x0000	32	0x0000_0000
FLASH Stop Register	EFM_FSTP	0x0004	32	0x0000_0000
FLASH Read Mode Register	EFM_FRMC	0x0008	32	0x0000_0000
FLASH Erase Mode Register	EFM_FWMC	0x000C	32	0x0000_0000
FLASH Status Register	EFM_FSR	0x0010	32	0x0000_0100
FLASH Status Clear Register	EFM_FSCLR	0x0014	32	0x0000_0000
FLASH Interrupt License Register	EFM_FITE	0x0018	32	0x0000_0000
FLASH Boot Swap Status Register	EFM_FSWP	0x001C	32	(botany) adventitious
FLASH Rewrite Allowed Area Start Address Register	EFM_FPMTSW	0x0020	32	0x0000_0000
FLASH Rewrite Allowed Area End Address Registers	EFM_FPMTEW	0x0024	32	0x0000_0000
FLASHuniqueID register	EFM_UQID0	0x0050	32	(botany) adventitious
FLASHuniqueID register	EFM_UQID1	0x0054	32	(botany) adventitious
FLASHuniqueID register	EFM_UQID2	0x0058	32	(botany) adventitious

7.9.1 Access Protection Register (EFM_FAPRT)

Reset value: 0x0000_0000

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

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31-b16	Reserved	-	Reads "0", writes "0".	R
b15-b0	FAPRT [15:0]	EFM Register Write Protect	Accesses the EFM register protection register. Solution: Write "16-bit data 0x0123" to FAPRT first and then Write "16-bit data 0x3210". In the unprotected state, any data is written and the EFM register is again	R/W
			When EFM register access protection is in effect, this register reads the value of 0x0000_0000. When the EFM register access protection is disabled, this register reads the value of 0x0000_0001.	

7.9.2 FLASH Stop Register (EFM_FSTP)

Reset value: 0x0000_0000

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

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31-b1	Reserved	-	Reads "0", writes "0".	R
b0 mode control	FSTPFLASH stop	0: FLASH activity status 1: FLASH is in stop mode When the register bit is set from 1 to 0, please perform FLASH access after confirming that the FSR.RDY bit is 1.	R/W	

7.9.3 Read Mode Register (EFM_FRMC)

Reset value: 0x0000_0000

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

-	-	-	-	-	-	-	LVM	FLWT[3:0]	-	-	-	-	-	SLP MD
b0														

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b25	Reserved	-	Reads "0", writes "0".	R
b24	CRST	Cache Reset Bit	0: Cached data not reset 1: Reset cached data	R/W
b23~b17	Reserved	-	Reads "0", writes "0".	R
b16	CACHE	cache permission bit	0: Turn off the cache function 1: Cache function enable	R/W
b15~b9	Reserved	-	Reads "0", writes "0".	R
b8	LVM	Ultra-low speed operation mode	0: Disable ultra-low speed operation mode 1: Turn on the ultra-low speed operation mode	R/W
0000b: No insertion of a wait cycle 0001b: insert 1 wait				
b7-b4	FLWT[3:0]	cycle FLASH read insert wait wait cycles Periodicity	0010b: insert 2 1110b: Insert 14 wait cycles 1111b: Insert 15 wait cycles	R/W
b3~b1	Reserved	-	Reads "0", writes "0".	R
b0	SLPMD	Ultra low power read	0: Normal CPU read mode 1: Ultra low power read mode	R/W

7.9.4 Erase Write Mode Register (EFM_FWMC)

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
-	-	-	-	-	-	-	BUS HLD CTL	-	PEMOD[2:0]	-	-	-	-	PEM ODE	

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b9	Reserved	-	Reads "0", writes "0".	R
b8	BUSHLDCTL	FLASH Erase, Bus Control During Programming	0: The bus is occupied during FLASH programmed erase. 1: Bus release during FLASH programmed erase.	R/W
b7	Reserved	-Reserved	Read "0", write "0". R000: read-only mode 001: Single Programming Mode 010: Single programmable readback mode 011: Continuous Programming Mode 100: Sector Erase Mode 101: Full Erase Mode 110: Read-only mode 111: Read-only mode PEMOD[2:0] can be written only if PEMODE=1.	R/W
b6~b4 Programming Mode	PEMOD[2:0]	FLASH Erase, Programming Mode	100: Sector Erase Mode 101: Full Erase Mode 110: Read-only mode 111: Read-only mode PEMOD[2:0] can be written only if PEMODE=1.	R/W
b3~b1	Reserved	-Reserved	Read "0", write "0".	R
b0	PEMODE	FLASH Erase, Programming License Mode	0: PEMOD[2:0] rewrite not allowed 1: PEMOD [2:0] rewrite license	R/W

7.9.5 Status Register (EFM_FSR)

Reset value: 0x0000_0100

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
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

-	-	-	-	-	-	-	RDY	-	-	COL ERR	OPT END	PGM ISM TCH	PGS ZER R	PEP RTE RR	PEW ERR
---	---	---	---	---	---	---	-----	---	---	------------	------------	-------------------	-----------------	------------------	------------

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b9	Reserved	-	Reads "0", writes "0".	R
b8	RDY	FLASH busy/idle status	0: FLASH busy status 1: FLASH idle state	R
b7~b6	Reserved	-Reserved	Read "0", write "0".	
			R0: FLASH read/write access is normal. 1: FLASH read/write access error Causes the FLASH read/write access operation to this location bit to be Neglected.	
b5	write access error flag bit	COLERRFLASH read and	0: FLASH is not erased or FLASH is being erased. 1: End of FLASH erasure.	R
b4	OPTEND	End-of-operation flag bit	Placement conditions: End of Program/Erase/Full Erase operation	R
b3	PGMISMTCH	Single programmed readback value inconsistency flag bit programmed readback value is consistent	0: Single 1: Inconsistent single programmed readback values	R
b2	PGSZERR	Programming address and size misalignment flag bits Programming address and size aligned	0: 1: Programming address and size not aligned	R
b1	PEPRERR	Program/erase error flag bit for protected address Program/erase address is an allowable rewrite area	0: 1: Program/erase action for protection window address	R
b0	PEWERR	Erase Mode Error Flag Bit	0: Erase FLASH in erase-permit mode 1: Erase FLASH in erase-unauthorized mode	R

7.9.6 Status Clear Register (EFM_FSCLR)

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	COL ERR CLR	OPT ENDC LR	PG MIS MTC HCL R	pgs zer rcl r	PEP RTE RRC LR	PEW ERR CLR

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b6	Reserved	-	Reads "0", writes "0".	R
b5	COLERRCLR	Clear the read/write conflict error flag bit	0: No clearing action occurs 1: Clear read/write conflict errors	R/W
			This bit is always 0 when read.	
b4	OPTENDCLR	Clear end-of-operation flag	0: No clearing action occurs 1: Clear the end-of-operation flag	R/W
			This bit is always 0 when read.	
b3	PGMISMTCHCLR	Clear the Programmed Readback Inconsistency Flag Bit	0: No clearing action occurs 1: Clear the programmed readback inconsistency flag bit	R/W
			This bit is always 0 when read.	
b2	PGSZERRCLR	Clear the programmed address and size misalignment flag bits	0: No clearing action occurs 1: Clear misalignment error flag bit This bit is always 0 when read.	R/W
b1	PEPRTERRCLR	Clear the Program/Erase for Error Flag Bit Protected Addresses	0: No clearing action occurs 1: Clear programming/erasure errors	R/W
			This bit is always 0 when read.	
b0	PEWERRCLR	Erase Mode Error Flag Bit	0: No clearing action occurs 1: Clear pattern write error	R/W
			This bit is always 0 when read.	

7.9.7 Interrupt license register (EFM_FITE)

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	
-	-	-	-	-	-	-	-	-	-	-	-	-	-	COL ERR ITE	OPT END ITE	PEE RRI TE

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b3	Reserved	-	Reads "0", writes "0".	R
b2	COLERRITE	Read-write conflict error interrupt license conflict error interrupt not allowed	0: Read/write 1: Read/Write Conflict Error Interrupt License	R/W
b1	OPTENDITE	End-of-operation interrupt license operation interrupt not allowed	0: End-of- 1: End-of-operation interrupt license	R/W
b0	PEERRITE	Program/erase error interrupt license Program/erase error interrupt not allowed	0: 1: Program/erase error interrupt license	R/W

7.9.8 Boot Swap Status Register (EFM_FSWP)

Reset value: variable

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
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	FSW P

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31-b1	Reserved	-	Reads "0", writes "0".	R
b0	FSWP address swap bit	sector 0 and sector 1	0: After the sector 0 and sector 1 address exchange reset, the CPU boots from sector 1. 1: Sector 0 and Sector 1 addresses are not exchanged After reset, the CPU boots from sector 0.	R
			The initial value of the register is determined by the FLASH address The decision of 0x0007_FFDC~0x0007_FFDF, whose address data is 0xFFFF_4321, is reset with the initial value of FSWP register bit FSWP as 0, otherwise it is 1.	

7.9.9 FLASH Window Protection Starting Address Register (EFM_FPMTSW)

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
															FPM SW[18:16]
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
FPMTSW [15:0]															

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31-b19	Reserved	-	Reads "0", writes "0".	R
b18-b0	FPMTSW [18:0]	Protection window start address	FLASH protection window start address	R/W

7.9.10 FLASH Window Protection End Address Register (EFM_FPMTEW)

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
															FPMTEW [18:16]
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
FPMTEW [15:0]															

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31-b19	Reserved	-	Reads "0", writes "0".	R
b18-b0	FPMTEW [18:0]	Protection window end address	End address of the FLASH protection window	R/W

7.9.11 UNIQUE ID register (EFM_UQID0)

Reset value: variable

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]															

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31-b0	UQID0[31:0]	unique code	Chip LOT number	R

7.9.12 UNIQUE ID register (EFM_UQID1)

Reset value: variable

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]															

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31-b24	UQID1 [31:24]	unique code	Fixed to all zeros	R
b23-b16	UQID1[23:16]	unique code	Wafer number	R
b15-b8	UQID1[15:8]	unique code	Fixed to all zeros	R
b7-b0	UQID1[7:0]	unique code	Chip LOT number	R

7.9.13 UNIQUE ID register (EFM_UQID2)

Reset value: variable

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]															

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31-b16	UQID2 [31:16]	unique code	Fixed to all zeros	R
b15-b8	UQID2 [15:8]	unique code	Chip on Wafer Y coordinate	R
b7-b0	UQID2[7:0]	unique code	Chip on Wafer X-coordinate	R

7.10 caveat

1. When FLASH is being erased, reset occurs and the erase operation is forced to stop, and the FLASH data will not be guaranteed. Users need to erase the address after the reset is released and then operate again.
2. If the erase/write operation will involve data in the cache, reset the cache loop after the erase/write operation (FRMC.CRST0=1).
3. Programming, please set the cache to invalid (FRMC.CACHE=0) before the erase operation.
4. One-time programmable area programming operations are not controlled by the window protection registers (FPMTSW,FPMTEW).
5. During continuous programming mode, the FLASH analog circuit will have high voltage state, and the long-term high voltage state will affect the FLASH characteristics, once the continuous programming is finished, please exit the continuous programming mode immediately. It is prohibited for the MCU to enter low power mode (sleep mode, stop mode, power-down mode) in continuous programming mode.
6. When bus release (FWMC.BUSHLDCTL=1) is set during programming and erasing, set the interrupt vector and interrupt subroutine to RAM if you need to respond to an interrupt during programming and erasing.

8 Internal SRAM (SRAM)

8.1 summary

This product comes with 4KB of power-down mode retention SRAM (Ret_SRAM) and 188KB of system SRAM (SRAMH/SRAM1/SRAM2/SRAM3)

SRAM can be accessed in bytes, half words (16-bit) or full words (32-bit). Read and write operations are performed at CPU speed and wait cycles can be inserted. The relationship between the wait cycle setting for read/write access and the CPU clock frequency is shown in Table 8-1. The wait cycle for each SRAM read/write access is set by the SRAM Wait Control Register (SRAM_WTCR).

Table 8-1 Relationship between Wait Cycle Settings for SRAM Read and Write Accesses and CPU Clock Frequency

Waiting cycles (CPU access cycles)	Range of CPU clock frequencies allowed for accessing high-speed SRAM (SRAMH)	Access to other SRAMs (SRAM1,2,3, Ret_SRAM) Allowable CPU clock frequency range
0wait (1 CPU cycle access)	0~200MHz	0~100MHz
1wait (2 CPU cycle accesses)	0~200MHz	0~200MHz
2wait (3 CPU cycle accesses)	0~200MHz	0~200MHz
3wait (4 CPU cycle accesses)	0~200MHz	0~200MHz
4wait (5 CPU cycle accesses)	0~200MHz	0~200MHz
5wait (6 CPU cycle accesses)	0~200MHz	0~200MHz
6wait (7 CPU cycle accesses)	0~200MHz	0~200MHz
7wait (8 CPU cycle accesses)	0~200MHz	0~200MHz

Ret_SRAM provides 4KB of data retention space in power down mode.

SRAM3 has ECC checking (Error Checking and Correcting) ECC checking for correcting one and checking two codes, i.e., one error can be corrected and two errors can be checked; SRAMH/SRAM1/SRAM2/Ret_SRAM has Even-parity check, each byte of data has a parity bit; SRAMH/SRAM1/SRAM2/Ret_SRAM has parity check, each byte of data has a parity bit. See Table 8-2 for detailed SRAM definitions.

Table 8-2 SRAM Space Allocation

name (of a thing)	quantitative (science)	address range	Calibration method
SRAM1	64KB	0x2000_0000~0x2000_FFFF	Even-parity check
SRAM2	64KB	0x2001_0000~0x2001_FFFF	Even-parity check
SRAM3	28KB	0x2002_0000~0x2002_6FFF	ECC check
Ret_SRAM	4KB	0x200F_0000~0x200F_0FFF	Even-parity check
SRAMH	32KB	0x1FFF_8000~0x1FFF_FFFF	Even-parity check

Attention:

- When using SRAM3 as the stack space, the wait time of SRAM3 must be set to 1wait, i.e., accessed over 2 CPU cycles.
- In the case where RAM parity errors are allowed to generate NMI interrupts and resets, the RAM space used must be initialized in word units when data is accessed, and the area of the RAM space used + 3 words must be initialized in word units when instructions are executed from SRAMH space.
- In cases where it is permissible to generate RAM ECC checksum errors to generate NMI interrupts and resets, the RAM space used must be initialized in word units when data is accessed.
- The SRAMH and SRAM1 boundaries do not support non-aligned accesses and must be used to avoid 32-bit accesses to the 0x1FFF_FFFD/0x1FFF_FFFE/0x1FFF_FFFF addresses, and 32-bit accesses to the 0x1FFF_FFFD/0x1FFF_FFFE/0x1FFF_FFFF addresses.
0x1FFF_FFFF The address initiates a 16bit access.

8.2 Register Description

register name	starting address	reset value
SRAM wait control register (SRAM_WTCR)	0x4005_0800	0x0000_0000
SRAM wait protection register (SRAM_WTPR)	0x4005_0804	0x0000_0000
SRAM checksum control register (SRAM_CKCR)	0x4005_0808	0x0000_0000
SRAM checksum protection register (SRAM_CKPR)	0x4005_080C	0x0000_0000
SRAM checksum status register (SRAM_CKSR)	0x4005_0810	0x0000_0000

8.2.1 SRAM Waiting Control Register (SRAM_WTCR)

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17				
Rev	SRAMR_ WWT [2:0]		Rev	SRAMR_ RWT [2:0]		Rev	SRAMH_ WWT [2:0]		Rev	SRAMH_ RWT [2:0]		Rev	SRAMH_ RWT [2:0]					
b16 b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0			
Rev	SRAM3_ WWT [2:0]		Rev	SRAM3_ RWT [2:0]		Rev	SRAM12_ WWT [2:0]		Rev	SRAM12_ RWT [2:0]		Rev	SRAM12_ RWT [2:0]					
classifier for honorific people	marking		celebrity		functionality		fill out or in (information on a form)											
b31	Reserved		-		Reads "0", writes "0".		R/W											
									000b: 1-cycle write									
									001b: 2-cycle write									
									010b: 3-cycle write									
b30~b28	SRAMR_ WWT [2:0]		Ret_SRAM write cycle selection		011b: 4-cycle write		R/W											
									100b: 5-cycle write									
									101b: 6-cycle write									
									110b: 7-cycle write									
									111b: 8-cycle write									
b27	Reserved		-0" for reading, "0" for writing.		Reads "0", writes "0".		R/W											
									000b: 1-cycle read									
									001b: 2-cycle read									
									010b: 3-cycle read									
b26~b24	SRAMR_ RWT [2:0]		Ret_SRAM read cycle selection		011b: 4-cycle reading		R/W											
									100b: 5-cycle read									
									101b: 6-cycle read									
									110b: 7-cycle read									
									111b: 8-cycle read									
b23	Reserved		-0" for reading, "0" for writing.		Reads "0", writes "0".		R/W											
									000b: 1-cycle write									
									001b: 2-cycle write									
									010b: 3-cycle write									
b22~b20	SRAMH_ WWT [2:0]		SRAMH write cycle selection		011b: 4-cycle write		R/W											
									100b: 5-cycle write									
									101b: 6-cycle write									
									110b: 7-cycle write									
									111b: 8-cycle write									
b19	Reserved		-0" for reading, "0" for writing.		Reads "0", writes "0".		R/W											
									000b: 1-cycle read									
									001b: 2-cycle read									
									010b: 3-cycle read									
b18~b16	SRAMH_ RWT[2:0]		SRAMH read cycle selection		011b: 4-cycle reading		R/W											
									100b: 5-cycle read									
									101b: 6-cycle read									
									110b: 7-cycle read									

b15	Reserved	-Reserved	Read "0", write "0". R/W 000b: 1-cycle write 001b: 2-cycle write 010b: 3-cycle write	
b14~b12	SRAM3_ WWT [2:0]	SRAM3 write cycle selection	011b: 4-cycle write 100b: 5-cycle write 101b: 6-cycle write 110b: 7-cycle write 111b: 8-cycle write	R/W
b11	Reserved	-0" for reading, "0" for writing.	Reads "0", writes "0".	R/W
b10~b8	SRAM3_ RWT[2:0]	SRAM3 read cycle selection	000b: 1-cycle read 001b: 2-cycle read 010b: 3-cycle read 011b: 4-cycle read 100b: 5-cycle read 101b: 6-cycle read 110b: 7-cycle read 111b: 8-cycle read	R/W
b7	Reserved	-	Reads "0" and writes "0".	R/W
b6~b4	SRAM12_ WWT [2:0]	SRAM1 and SRAM2 Write Weeks rescheduling	000b: 1-cycle write 001b: 2-cycle write 010b: 3-cycle write 011b: 4-cycle write 100b: 5-cycle write 101b: 6-cycle write 110b: 7-cycle write 111b: 8-cycle write	R/W
b3	Reserved	-	Reads "0" and writes "0".	R/W
b2~b0	SRAM12_ RWT [2:0]	SRAM1 and SRAM2 Read Weeks rescheduling	000b: 1-cycle read 001b: 2-cycle read 010b: 3-cycle read 011b: 4-cycle read 100b: 5-cycle read 101b: 6-cycle read 110b: 7-cycle read 111b: 8-cycle read	R/W

Attention:

-The SRAM with ECC function supports only 32bit write access to SRAM when ECC function is enabled. SRAM with ECC function, SRAM write access only supports 32bit write when ECC function is enabled.

8.2.2 SRAM Wait Protection Register (SRAM_WTPR)

Reset value: 0x0000_0000

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

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b8	Reserved	-	Reads "0" and writes "0".	R/W
b7~b1 bits to enable the current register.	WTPRKW[6:0]	Write key code	To write to the current register, write "3b" to these disable	R/W
b0	WTPRC	SRAM wait control register write control	0: SRAM wait control register write disable 1: SRAM wait control register write enable	R/W

WTPRC: Controls the SRAMWTCR register write operation. When WTPRC is set to 1, write operation to SRAMWTCR is allowed, if it is set to 0, write operation to SRAMWTCR is not allowed. When this bit is written, 0x3B must be written to WTPRKW[6:0] at the same time.

8.2.3 SRAM Checksum Control Register (SRAM_CKCR)

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16			
Rev					ECCMOD [1:0]		Rev					ECC OAD						
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0			
Rev														PYOAD				
<hr/>																		
classifier for honorific people	marking					functionality					fill out or in (information on a form)							
b31~b26	Reserved					Reads "0", writes "0".					R/W							
b25~b24	ECC MOD [1:0]	ECC checksum allowable bits for SRAM3					00: Disable ECC checksum function 01: ECC error correction if 1 bit is wrong. No 1-bit error flag is generated and no interrupt/reset is generated; if a 2-bit error is generated, the ECC detects the error and the Generate 2-bit error flag, generate interrupt/reset.					R/W						
b23~b17	Reserved	10: ECC error correction if 1 bit is wrong. Generate 1-bit error flag, no interrupt/reset; if 2-bit error, ECC detects the error and Generate 2-bit error flag, generate interrupt/reset.					11: ECC error correction if 1 bit is wrong. Generate 1-bit error flag, generate interrupt/reset; if 2-bit error, ECC detects error and Generate 2-bit error flag, generate interrupt/reset.					R/W						
b16	ECCOAD	ECC Calibration post-event operation	0: Non-maskable interrupt 1: Reset					R/W					R/W					
b15~b1	Reserved	Reads "0", writes "0".					R/W					R/W						
b0	PYOAD	Parity check Operation after error	0: Non-maskable interrupt 1: Reset					R/W					R/W					

Attention:

- In the case where RAM parity errors are allowed to generate NMI interrupts and resets, the RAM space used must be initialized in word units when data is accessed, and the area of the RAM space used + 3 words must be initialized in word units when

- In cases where it is permissible to generate RAM ECC checksum errors to generate NMI interrupts and resets, the RAM space used must be initialized in word units when data is accessed.

8.2.4 SRAM Checksum Protection Register (SRAM_CKPR)

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Rev															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Rev								CKPRKW [6:0]							
<hr/>															
classifier for honorific people	marking	celebrity				functionality				fill out or in (information on a form)					
b31~b8	Reserved	-				Reads "0", writes "0".				R/W					
b7~b1	CKPRKW [6:0]	Write key code				To write to the current register, write "3b" to these bits to enable the current register.				register enable R/W					
b0	CKPRC	SRAM checksum control register write enable checksum control register write disable				0: SRAM 1: SRAM checksum control register write enable				R/W					

CKPRC: Controls the write of SRAMCKCR register. When CKPRC is set to 1, write operations to SRAMCKCR are allowed, if it is set to 0, write operations to SRAMCKCR are not allowed. When this bit is written, 0x3B must be written to CKPRKW[6:0] at the same time.

8.2.5 SRAM Checksum Status Register (SRAM_CKSR)

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	
Rev																
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
Rev																
<hr/>																
Bitemarker	Bit Name	functionality				fill out or in (information on a form)										
b31~b5	Reserved	-Reserve				Reads "0", writes "0".				R/W						
b4	SRAMR_PYERR	Ret_SRAM parity				0: No parity error occurs				R/W						
		Checksum Error Flag				1: A parity error has occurred				(Note 1)						
b3	SRAMH_PYERR	SRAMH Parity				0: No parity error occurs				R/W						
		Checksum Error Flag				1: A parity error has occurred				(Note 1)						
b2	SRAM12_PYERR	SRAM1 and SRAM2				0: No parity error occurs				R/W						
		Checksum Error Flag				1: A parity error has occurred				(Note 1)						
b1	ECC 2-bit	SRAM3_2ERR	SRAM3 generates				0: No 2-bit ECC error				occurs					
			error message				1: A 2-bit ECC error has				1: A 1-bit ECC error					
b0	ECC 1-bit	SRAM3_1ERR	SRAM3 generates				0: No 1-bit ECC error				has occurred					
			error message				1: A 1-bit ECC error				has occurred					

Attention:

-Write 1 to clear 0. Write 1 to clear 0.

9 General Purpose IO (GPIO)

Some of the abbreviations used in this chapter:

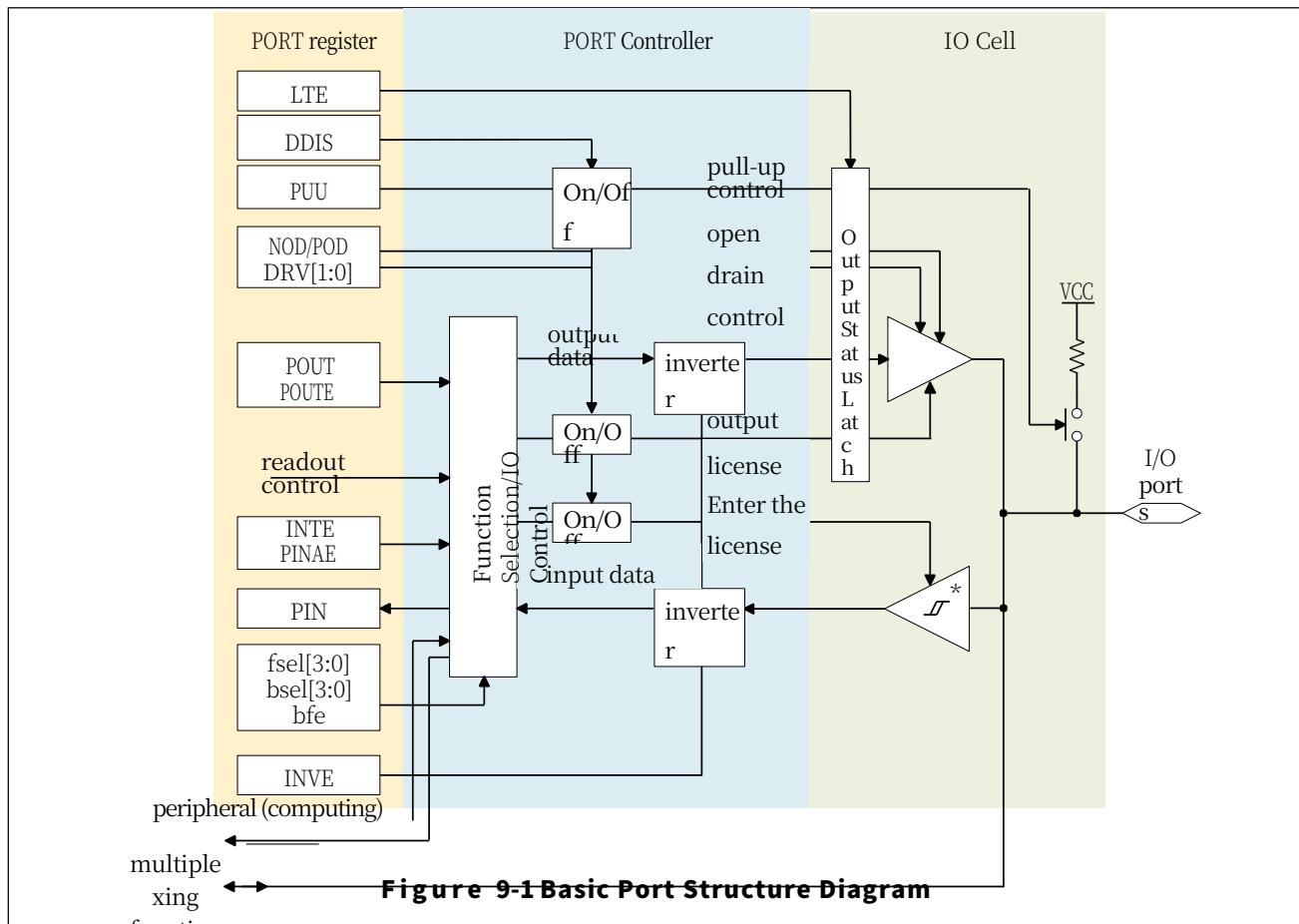
- Px (x=A~E,H) denotes a group of ports, such as PA denotes the 16 I/O ports in the group of PA0~PA15.
- Pxy (x= A~E,H, y=0~15, same as below) denotes a single I/O port, e.g., PB10 port denotes the 10th I/O in the PB group.
- GPIO (General Purpose Input Output) General Purpose Input Output.
- NOD/POD (Nmos/Pmos Open Drain) NMOS/PMOS open drain output mode.

9.1 summary

Key Features:

- 16 I/O pins per port group, may be less than 16 depending on actual configuration
- Support for pull-ups
- Supports push-pull, open-drain output modes
- Supports high, medium and low drive modes
- Inputs that support external interrupts
- Supports I/O pin peripheral function multiplexing, one I/O pin can have up to 64 selectable multiplexing functions.
- Each I/O pin can be programmed independently
- Each I/O pin can be selected to have 2 functions active at the same time (2 output functions active at the same time are not supported).

9.2 Port Function Summary



Please refer to the **Pin Configuration and Function** section in the datasheet for details on GPIO port number, 5V withstand voltage, and drive capability configuration.

multiple
function
analog
(device
as
opposed
digital)
input
and
output

9.3 Action Description

9.3.1 General Purpose Inputs and Outputs GPIO Functions

General Purpose Input Function GPI:

Each I/O has a general-purpose input GPI function, and the GPI function is always active when the digital function disable bit PCRxy.DDIS is 0, independent of the FSEL[5:0] setting of PFSRxy in the function select register. The current port status can be obtained by accessing the port input data register PIDRx. The status of the corresponding single I/O port can also be queried through the PIN bit of the port control register PCRxy. The PIDRx.PIN[y] register bit is equivalent to the PCRxy.PIN bit.

To reduce power consumption, the input MOS of the I/O is turned off when the I/O does not select the peripheral function. It is turned on only when the PIDRxPCRxy registers are read, and this operating mode of the I/O prevents the input hysteresis characteristic from functioning properly because the input MOS switches from a high level in the off state to a high or low (depending on the input value of the I/O) level for each read operation, so the threshold VIL for a high to a low level is normal but the threshold VIH for a low to a high level is not functioning. cannot be utilized. In order for the I/O to properly utilize the input hysteresis characteristics, it is necessary to keep the MOS turned on at all times. This can be achieved by setting register PINAER.PINAE[x] to 1, or by setting PFSRxy to select a peripheral function (other than GPO).

Due to the latency of the I/O inputs, a single cycle may not be able to read the input status value correctly when the system is running at a high speed clock. In this case, it is necessary to set register PCCR.RDWT[1:0] to insert a number of wait cycles. For details, refer to the description of register PCCR.

General Purpose Output Function GPO:

All I/O ports have a general-purpose output GPO function except the input-only PB11 port. The GPO function is available by setting the port function selection register PFSRxy.FSEL[5:0] to 0x0.

When the GPO function is active, you can set the general-purpose output license register POERx to allow or disable the output of I/O, and the general-purpose output data register PODRx to control the output value. The output value of I/O can also be controlled by using the following three registers: output data clear register PORRx, output data set register POSRx, and output data flip-flop register POTRx. Writing 1 to the corresponding bit in the above registers can cause the corresponding I/O output to be 0, 1, or flip-flop. The I/O output status is not changed when 0 is written.

The above registers are all operated together as a group of 16 PORTs. In order to

facilitate the control of individual I/Os, the output of I/Os can also be allowed or disabled by setting PCRxy.POUTE, which is equivalent to POERx.POUTE[y]. POUT can be set to control the output value of I/O. PCRxy.POOUT register bit is equivalent to PODRx.POOUT[y]. PCRxy is suitable for controlling a single PORT, and POERx/PODRx is suitable for controlling a 16-bit integer PORT.

After system reset, all ports are initially functioning as GPOs (FSEL[5:0]=0x0) and are in the high resistance state (output disable POUTExy=0), except for the JTAG multiplexing ports PA13, PA14, PA15, PB3, PB4, and the secondary oscillator multiplexing ports PC14 and PC15.

Attention:

Port PB11 is multiplexed with MD as an input-only port with no output capability. Port PB11 is multiplexed with MD and is dedicated for inputs and has no output function.

9.3.2 Peripheral Functions

Up to 64 functions per port can be configured via FSEL[5:0] of the function selection register PFSRxy. These include the general-purpose output GPO functions corresponding to FSEL[5:0]=0x0. Please refer to the pinout **table in the datasheet** for the specific functions configured for each port.

JTAG/SWD debug function, use register PSPCR to select. pspcr.spfe[z], z=0~4 is 1, the corresponding port's PFSRxy.FSEL[5:0] register bits are invalid, i.e., the SPFE priority is higher than FSELpSPCR register initial value is 0x1F, JTAG/SWD function is valid. If you want to set these ports to 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 select one function by **PFSRxy.FSEL[5:0]**, and then select the second function by setting **PFSRxy.BFE** to 1, and setting the common control register **PCCR.BFSEL[3:0]**. For example, if you set **PFSRxy.FSEL[5:0]=0x2**, **PCCR.BFSEL[3:0]=0x5**, and **PFSRxy.BFE=0x1**, function 2 and function 5 will be valid at the same time on Pxy. It is forbidden to have 2 output functions valid on the same port at the same time.

9.3.4 Event Port Input/Output Function

Supports 4 groups of Event Ports, 16 ports in each group, Event Port1 contains EVNTP100~EVNTP115, Event Port2 contains EVNTP200~EVNTP215, and so on EVNTPmn (m=1~4, n=0~15) ports can be used as a trigger source to generate an event to trigger other peripheral devices (e.g. TIMER, ADC, DMA, etc.) to start specific actions according to the input from the ports. The port can be used as a trigger source to generate events based on port inputs to trigger other peripheral devices (e.g., TIMER, ADC, DMA, etc.) to start specific actions. It can also be used as a triggered object, which accepts events and automatically inputs or outputs.

When used as a trigger source, set PEVNTRISRm,PEVNTFALRm,PEVNTNFRC to select the rising or falling edge detection, and digital filtering function, and set function selection register PFSRxy to select the EVNTPmn function. When the selected edge is input from the port, the event EVENT_PORTm is generated and output to other peripheral devices to trigger them to start action.

When used as the triggered object, set PEVNT_TRGSELm to select the trigger event source, and set PEVNTDIRRm to select the output or input function. For the output function, when the selected event occurs, EVNTPmn outputs the specified level or flips according to the set values of PEVNTODRm, PEVNTORRm, and PEVNTOSRm. For input function, when the selected event occurs, EVNTPmn input status is saved into register PEVNTIDRx.

To use the Event Port function, you need to set the Auto Run System AOS Function Enable

bit of the Function Clock Control 0 register (PWC_FCG0) to valid first.

Attention:

Port PB11 is multiplexed with MD as an input-only port.

Port PB11 is multiplexed with MD as an input-only port, so there is no output function for EVNTP211.

9.3.5 External Interrupt EIRQ Input Function

Each I/O port has an external interrupt input function. When the PCRxy.INTE bit is set to 1, this I/O is allowed to input as an external interrupt source EIRQy. Each EIRQy can be configured with more than one I/O, so do not allow more than one I/O input per EIRQy at the same time. The EIRQy input function can be valid simultaneously with the peripheral functions (including GPIO) selected by PFSRxy.FSEL.

In addition, the external non-maskable interrupt NMI is multiplexed with the PB11/MD port.

When the I/O port is used as an external interrupt EIRQ, it is necessary to set the filter, interrupt trigger edge, interrupt number, etc., in conjunction with the interrupt controller INTC. For details, please refer to "Interrupt Controller (INTC)"

9.3.6 analog function

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

9.3.7 Universal Control

1. Pull-up/down resistor

Each I/O port has an internal pull-up resistor. The register PFSRxy.PUU bit can be set to allow this function to be internally in a weak 1 state when there is no input to the I/O port. The pull-up function is automatically disabled when the I/O port is in the output state.

When the I2Cx_SCL/I2Cx_SDA function is selected for the I/O port, the setting of register PUU is ignored and the internal pull-up function is forced to be invalid.

PA11, PA12 are multiplexed with the USBFS_DM, USBFS_DP pins and have built-in pull-down resistors of about 400KΩ that are always active.

2. Driveability control

Each I/O port has an adjustable 3-stage drive capability of high, medium, and low, and register PFRSxy.DRV[1:0] can be set as desired. This function is effective only when the port is in output state.

3. open-drain output mode

Setting the PFRSxy.NOD bit sets the I/O port to NMOS open-drain output mode. When NOD is valid, the corresponding port can output 0 normally, while the port will be in high resistance state when 1 is output.

When the I2Cx_SCL/I2Cx_SDA function is selected for the I/O port, the setting of register NOD is ignored and the open-drain output mode is forced to be valid.

The general control functions described above, unless otherwise stated, are independent of the function specifically selected for the port, i.e. the setting of FSEL[5:0].

9.4 Register Description

BASE_ADDR: 0x4005_3800

Table 9-1 PORT Register List 1

register name	notation	offset address	bit width	reset value
General Purpose Input Data Register	PIDRx	0x00+0x10*n ^{*1}	16/32	0xFFFF
General Purpose Output Data Register	PODRx	0x04+0x10*n	16/32	0x0000
General Purpose Output License Register	POERx	0x06+0x10*n	16/32	0x0000
General Purpose Output Setting Register	POSRx	0x08+0x10*n	16/32	0x0000
General Purpose Output Reset Register	PORRx	0x0A+0x10*n	16/32	0x0000
General Purpose Output Flip-Flop Register	POTRx	0x0C+0x10*n	16/32	0x0000
Special Control Registers	PSPCR	0x3F4	16/32	0x001F
Common Control Register	PCCR	0x3F8	16/32	0x4000
Input Control Register	PINAER	0x3FA	16/32	0x0000
Write-Protect Register	PWPR	0x3FC	16/32	0x0000
General Purpose Control Register	PCRxy	0x400+0x40*n+0x4*y	16/32	0x000 ^{*2}
Function Selection Register	PFSRxy	0x402+0x40*n+0x4*y	16/32	0x0000

Note *1: x=A~E,H corresponds to n=0~4,5 in the address calculation formula.

*2: The reset value of 32K sub-oscillator multiplexing port PCRC14, PCRC15 is 0x8100.

BASE_ADDR: 0x4001_0800

Table 9-2 PORT Register List 2

register name	notation	offset address	bit width	reset value
Event Port Direction Selection Register	PEVNTDIRm	0x100+0x1C*(m-1)	32	0x0000_0000
Event Port Input Data Register	PEVNTIDRm	0x104+0x1C*(m-1)	32	0x0000_0000
Event Port Output Data Register	PEVNTODRm	0x108+0x1C*(m-1)	32	0x0000_0000
Event Port Output Data Reset Register	PEVNTORRm	0x10C+0x1C*(m-1)	32	0x0000_0000
Event Port Output Data Setting Register	PEVNTOSRm	0x110+0x1C*(m-1)	32	0x0000_0000
Event Port Rising Edge Input	PEVNTRISRm	0x114+0x1C*(m-1)	32	0x0000_0000

				www.xhsc.com.cn
Event Port Falling Edge Input License Register	PEVNTFALRm	0x118+0x1C*(m-1)	32	0x0000_0000
Event Port Input Filter Control Register	PEVNTNFCR	0x170	32	0x0000_0000

Note: m=1~4

9.4.1 General Purpose Input Register (PIDRx)

Reset value: 0xFFFF

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/Write
b15~b0	PIN[15:0]	Input Status	0: I/O port input status is low 1: I/O port input state is high

This register is a read-only register and is not valid for writing. When the digital function is not disabled DDIS=0, the input status of the port can be obtained by reading this register, independent of the PFSRxy.FSEL[5:0] setting of the function selection register. There is no indeterminate readout value of the corresponding bit of the port. When the port's digital function disable state DDIS=1, the corresponding PIN bit readout value is fixed at 0x1 because the I/O input MOS is off.

9.4.2 General Purpose 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/Write
b15~b0	POUT[15:0]	Output data	0: Output low level 1: Output high level

When an I/O port is set to GPO function, rewriting this register can change the output state of the corresponding port.

9.4.3 General Purpose Output License Register (POERx)

Reset value: 0x0000

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

classifierfor honorific people	marking	Bit Name Function Read/Write	
b15~b0	POUTE[15:0]	Output license	0: Output Prohibit 1: Output license

When the I/O port is set to GPO function and this register is set to 1, the PODRx set value is output to the corresponding I/O port. When this register is set to 0, the output is turned off

9.4.4 General Purpose Output Setting 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/Write
b15~b0 in PODRx.POUT	POS[15:0]	output high	0: corresponds to no change R/W
			1: corresponds to PODRx.POUT set to 1

The readout value of this register is always 0x000032bit access, when POR[y] and POS[y] of the same I/O write 1 at the same time, POR[y] has a higher priority, i.e., the corresponding POUT[y] is cleared to zero.

9.4.5 General purpose output reset register (PORRx)

Reset value: 0x0000

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

Bit Flag	Bit Name	Function	Read/Write
b15~b0 in PODRx.POUT	POR[15:0]	output low	0: corresponds to no change R/W
			1: Corresponding to PODRx.POUT cleared to zero

The readout value of this register is always 0x0000.

9.4.6 General purpose output flip-flop 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/Write
b15~b0 in PODRx.POUT	POT[15:0]	Output flip-flop	0: corresponds to no change R/W
			1: Corresponds to PODRx.POUT inverted

The readout value of 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]														

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b15~b6	Reserved	-	0 when reading, please write 0 when writing	R
b4	SPFE[4]	Special Function Options	0: NJTRST function is invalid 1: NJTRST function is valid	R/W
b3	SPFE[3]	Special Function Options	0: JTDI function is invalid 1: JTDI function is effective	R/W
b2 function is not available	SPFE[2]	Special function selection	0: JTDO_TRACESWO 1: JTDO_TRACESWO function is effective	R/W
b1 function is not available	SPFE[1]	Special function selection	0: JTMS_SWDIO 1: JTMS_SWDIO function is active	R/W
b0 function is not available	SPFE[0]	Special function selection	0: JTCK_SWCLK 1: JTCK_SWCLK function is effective	

Attention:

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

9.4.8 Public Control Register (PCCR)

Reset value: 0x4000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0											
RDWT[1:0]		-	-	-	-	-	-	-	-	-	-	BFSEL [3:0]														

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
Set the number of wait cycles to insert when reading registers PIDRx, PCRxy				
b15-b14	RDWT[1:0]	Read Port Wait	Setting value Operating frequency	Waiting period Recommended
HC32F460_F45x_A460 Series Reference Manual_Rev1.6	01 (default)	01 cycle	00 No Wait 42~84MHz	~42MHz

b13~b4	Reserved	-	0 when reading, please write 0 when writing	R
b3~b0	BFSEL [3:0]	Sub-Function Selection	Please refer to the pinout table in the datasheet for the functional configuration of each port.	R/W

9.4.9 Input Control Register (PINAER)

Reset value: 0x0000

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

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b15~b6	Reserved	-	0 when reading, please write 0 when writing	R
			0: Input MOS normally open invalid 1: Input MOS normally open active PINAЕ[0] controls PA0~PA15.	
b5~b0	PINAЕ [5:0]	Input Normally Open	PINAЕ [1] controls PB0 to PB15. PINAЕ [2] controls PC0 to PC15. PINAЕ [3] controls PD0 to PD15. PINAЕ [4] controls PE0 to PE15. PINAЕ [5] control PH0~PH2	R/W

9.4.10 Write Protect 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

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b15~b8	WP[7:0]	write-protect code	0x00 on readout When b15~b8 write value is 0xA5, b0 value is written to WE	W
			WE is automatically cleared when writing a value other than 0xA5	
b7~b1	Reserved	-	0 when reading, please write 0 when writing	R
b0	WE	authorization	0: PSPCR, PCCR, PINAER, PCRxy, PFSRxy registers write disabled 1: PSPCR, PCCR, PINAER, PCRxy, PFSRxy register write permission	R/W

9.4.11 General Purpose 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	-	INTE	-	-	INVE	PIN	-	PUU	DRV[1:0]	-	NOD	POUTE	POUT	

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b15	DDIS	Digital Function Disable	0: Digital function active 1: Digital Function Prohibition	R/W
b14	LTE	Output state latching	0: Output latch invalid 1: Output latch active	R/W
b13	Reserved	-	0 when reading, please write 0 when writing	R
b12	INTE	External Interrupt License	0: External interrupt input disabled 1: External interrupt input license	R/W
b11~b10	Reserved	-	0 when reading, please write 0 when writing	R
b9	INVE	inversion license	0: Input and output data are not inverted 1: Inversion of input and output data	R/W
b8	PIN	input state	0: I/O port input status is low 1: I/O port input state is high Same function as PIN[y] in register PIDRx	R
b7	Reserved	-	0 when reading, please write 0 when writing	R
b6	PUU	pull-up license	0: Invalid internal pullup resistor 1: Internal pullup resistor active	R/W
b5~b4	DRV[1:0]	Drive Mode Selection	b00: Low drive mode b01: Mid-drive model b1*: High-drive model	R/W
b3	Reserved	-	0 when reading, please write 0 when writing	R
b2	NOD	NMOS open drain	0: Normal CMOS output mode 1: NMOS open-drain output	R/W
b1	POUTE	output license	0: Output disabled 1: Output license Same function as POUTE[y] in register POERx	R/W
b0	POUT	output data	0: Output low level 1: Output high level Consistent with the function of POUT[y] in register PODRx	R/W

When DDIS is set to 1, all digital functions of the corresponding port are forced to be disabled, including general-purpose inputs and outputs, peripheral digital inputs and outputs, pull-up/pull-down functions, and external interrupt input functions. Set the DDIS bit to 1 when the port is used as an analog input.

When LTE is set to 1 and output latch is valid, the current output state of the port is maintained until LTE is written to 0. This function is mainly used when the port function is

switched. In order to avoid unexpected burrs on the port output during function switching which may lead to system malfunction, before function switching, first write LTE to 1 to latch the output state of the port, then rewrite the register to select the register to switch the function, and finally write LTE to 0 to un-latch it, and the port state will be updated to the new function.

When INVE is set to 1, the input and output data of the port are inverted, including GPIO functions, and other peripheral input and output functions.

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

9.4.12 Function Selection 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]			

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b15~b9	Reserved	-	0 when reading, please write 0 when writing	R
b8	BFE	Sub-functional licenses	Controls whether the sub-function selected by PCCR.BFSEL[3:0] is active or not 0: Sub-function disabled 1: Parafunctional validity	R/W
b7~b6	Reserved	-	0 when reading, please write 0 when writing	R
b5~b0	FSEL[5:0]	Function Selection	Please refer to the pinout table in the datasheet for the functional configuration of each port.	R/W

Each I/O port can select one of several functions configured on that port via FSEL[5:0]. Referring to the pin menu in the datasheet, FSEL[5:0] is set to b000000 to select Func0, b000001 to select Func1, and so on, and b001111 to select Func15, which corresponds to the general-purpose output function GPO.

Attention:

- PA13, PA14, PA15, PB3, PB4 ports initial state after reset is JTAG/SWD function is valid, when configuring FSEL[5:0] to select the function, you need to write 0 to the corresponding bit of register PSPCR to invalidate the JTAG/SWD function. PC14, PC15 ports initial state after reset is digital function prohibited, when selecting digital functions When selecting the digital function, it is necessary to write 0 to the DDIS bit of the corresponding register PCRxy first to make the digital function valid.

9.4.13 Event Port Direction Selection Register (PEVNTDIRRm)

Reset value: 0x0000_0000

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

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b16	Reserved	-	0 when reading, please write 0 when writing	R
b15~b0	PDIR[15:0]	Orientation	0: Event Port is an input function 1: Event Port is an output function	R/W

Attention:

-EVNTP211 Function No output function (configured on PB11 port)

9.4.14 Event Port Input Data Register (PEVNTIDRm)

Reset value: 0x0000_0000

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

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b16	Reserved	-	0 when reading, please write 0 when writing	R
b15~b0	PIN[15:0]	Port Input Status	0: Event Port input state is low when an event is triggered 1: Event Port input state is high when the event is triggered	R

When the direction of Event Port is set to Input State, the input state of the corresponding I/O port is saved to this register when the set event is triggered.

9.4.15 Event Port Output Data Register (PEVNTODRm)

Reset value: 0x0000_0000

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

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b16	Reserved	-	0 when reading, please write 0 when writing	R
b15~b0 level	POUT[15:0]	Port Output Value	0: Event Port output low 1: Event Port output high	R/W

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

9.4.16 Event Port Output Data Reset Register (PEVNTORRm)

Reset value: 0x0000_0000

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

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b16	Reserved	-	0 when reading, please write 0 when writing	R
b15~b0	POR[15:0]	Output value reset	0: No change in the corresponding PEVNTODRm.POUT when the event is triggered 1: Corresponds to PEVNTODRm.POUT reset when the event is triggered	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 Output Data Setting Register (PEVNTOSRm)

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
POS[15:0]															

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b16	Reserved	-	0 when reading, please write 0 when writing	R
b15~b0	POS[15:0]	Output value set	0: No change in the corresponding PEVNTODRm.POUT when the event is triggered 1: Corresponding PEVNTODRm.POUT is set when the event is triggered	W

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

9.4.18 Event Port rising edge input license register (PEVNTRISRm)

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
RIS [15:0]															

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b16	Reserved	-	0 when reading, please write 0 when writing	R
b15~b0	RIS [15:0]	Rising along the inspection license	0: EVNTPmn rising edge event checkout is invalidated 1: EVNTPmn rising edge event checkout is active PEVNTRISRm.RIS[n] corresponds to EVNTPmn	R/W

Event Port acts as an event source, when the RIS bit is set to 1, corresponding to the rising edge of the input of EVNTPmn, it outputs an event to trigger other peripheral modules. The edge events of EVNTPm0~15 are combined into a single event EVENT_PORTm to be output, where any one of the ports detects an edge and outputs the event EVENT_PORTm.

9.4.19 Event Port Falling Edge Input License Register (PEVNTFALRm)

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
FAL[15:0]															

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b16	Reserved	-	0 when reading, please write 0 when writing	R
b15~b0	FAL[15:0]	Falling edge detection license	0: EVNTPmn falling edge event checkout is invalidated 1: EVNTPmn falling edge event checkout is active PEVNTRISRm.FAL[n] corresponds to EVNTPmn	R/W

Event Port acts as an event source, when the FAL bit is set to 1, corresponding to the falling edge of the input of EVNTP, it outputs an event to trigger other peripheral modules. The edge events of EVNTPm0~15 are combined into one event EVENT_PORTm output, in which any one of the ports detects an edge and outputs the event EVENT_PORTm.

9.4.20 Event Port Input Filter Control Register (PEVNTNFCR)

Reset value: 0x0000_0000

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

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b27	Reserved	-	0 when reading, please write 0 when writing	R
b26-b25	DIVS4[1:0]	Digital Filter Sampling Clock Selection	Event Port4 Digital Filter Sample Clock Selection 00: PCLK1 01: PCLK1/8 10: PCLK1/32 11: PCLK1/64	R/W
b24	NFEN4 Digital Filtering Invalid	Digital Filtering License	0: Event Port4 1: Event Port4 digital filter active	R/W
b23~b19	Reserved	-0 for readout, write 0		for writeout
b18-b17	DIVS3[1:0]	Digital Filter Sampling Clock Selection	Event Port3 Digital Filter Sample Clock Selection 00: PCLK1 01: PCLK1/8 10: PCLK1/32 11: PCLK1/64	R/W
b16	NFEN3 Digital Filtering Invalid	Digital Filtering License	0: Event Port3 1: Event Port3 digital filter active	R/W
b15~b11	Reserved	-0 for reading, write 0		for writing
b10-b9	DIVS2[1:0]	Digital Filter Sampling Clock Selection	Event Port2 Digital Filter Sample Clock Selection 00: PCLK1 01: PCLK1/8 10: PCLK1/32 11: PCLK1/64	R/W
b8	NFEN2 Digital Filtering Invalid	Digital Filtering License	0: Event Port2 1: Event Port2 digital filter active	R/W
b7-b3	Reserved	-0 for readout, please write 0		for writeout
b2-b1	DIVS1[1:0]	Digital Filter Sampling Clock Selection	Event Port1 Digital Filter Sample Clock Selection Selection 00: PCLK1 01: PCLK1/8 10: PCLK1/32 11: PCLK1/64	

R/W

b0 NFEN1
Digital Filtering Invalid

Digital Filtering License

0: Event Port1

1: Event Port1 digital filter
active

R/W

9.4.21 32bit access

Among the registers mentioned above, except for Event The registers mentioned above support 16bit and 32bit access, except for the registers related to Event and Port, which support 32bit access, and they are combined as follows:

Table 9-3 List of PORT registers for 32-bit accesses

Addresses	b31 ~	b16	b15 ~b15	b15 ~ b0
0x4005_3800+0x10*n *1	Reserved		PIDRx	
0x4005_3804+0x10*n	POERx		PODRx	
0x4005_3808+0x10*n	PORRx		POSRx	
0x4005_380C+0x10*n	Reserved		POTRx	
0x4005_3BF4	Reserved		PSPCR	
0x4005_3BF8	PINAER		PCCR	
0x4005_3BFC	Reserved		PWPR	
0x4005_3C00+0x40*n+0x04*y	PFSRxy		PCRxy	

Note *1: x=A~E,H corresponds to n=0~4,5 in the address calculation formula.

9.5 caveat

Do not set the same function to multiple ports.

When using the analog function, turn off the digital function of the corresponding port (DDIS=1).

Please perform port function switching when output latching is active (LTE=1) to avoid outputting burrs other than those expected on the port during switching.

10 Interrupt Controller (INTC)

10.1 summary

The functions of the interrupt controller (INTC) are selecting interrupt event requests as interrupt inputs to the NVIC to wake up the WFI; selecting interrupt event requests as event inputs to wake up the WFE; selecting interrupt event requests as wake-up conditions for low-power modes (sleep and stop modes); the interrupt control function for the external pins NMI and EIRQ; and the interrupt/event selection function for software interrupts.

Key Specifications:

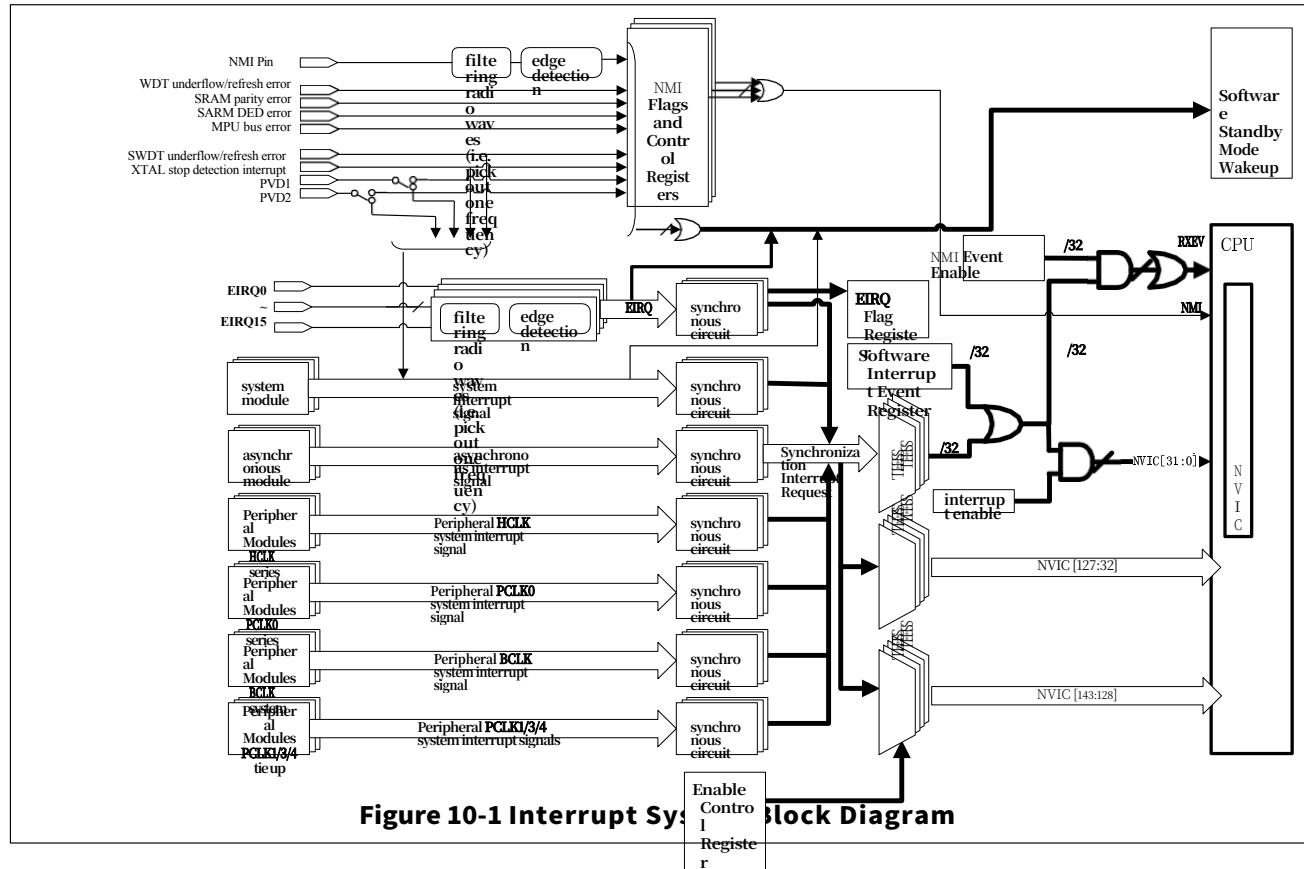
- 1) NVIC interrupt vectors: Please refer to [Interrupt Vector Table] the actual number of interrupt vectors used (excluding the 16 interrupt lines of the Cortex™-M4F) and each interrupt vector can be selected according to the interrupt selection register to correspond to the peripheral interrupt event request. For more information on exceptions and NVIC programming, please refer to Chapter 5: Exceptions and Chapter 8: Nested Vector Interrupt Controller in the ARM Cortex™-M4F Technical Reference Manual.
- 2) Programmable Priority: 16 programmable priorities (4-bit interrupt priority register is used)
- 3) Non-maskable interrupts: In addition to the NMI pin as the source of non-maskable interrupts, a variety of system interrupt requests can be independently selected as non-maskable interrupts, and each interrupt request is equipped with independent enable selection, flag, and flag clear registers.
- 4) Equipped with 16 external pin interrupts.
- 5) Configure a variety of peripheral interrupt event requests, refer to [Interrupt Event Request Sequence]
- 6) Equipped with 32 software interrupt event requests.
- 7) Interrupts can wake up the system in sleep mode and stop mode.

Input Pins:

pin name	I/O	clarification
NMI	importation	Non-maskable interrupt request pin
EIRQ0	importation	External pin interrupt event request 0
EIRQ1	importation	External pin interrupt event request 1
EIRQ2	importation	External Pin Interrupt Event Request 2
EIRQ3	importation	External Pin Interrupt Event Request 3
EIRQ4	importation	External Pin Interrupt Event Request 4
EIRQ5	importation	External Pin Interrupt Event Request 5
EIRQ6	importation	External Pin Interrupt Event Request 6
EIRQ7	importation	External Pin Interrupt Event Request 7
EIRQ8	importation	External Pin Interrupt Event Request 8
EIRQ9	importation	External Pin Interrupt Event Request 9
EIRQ10	importation	External pin interrupt event request 10
EIRQ11	importation	External Pin Interrupt Event Request 11
EIRQ12	importation	External pin interrupt event request 12
EIRQ13	importation	External Pin Interrupt Event Request 13
EIRQ14	importation	External Pin Interrupt Event Request 14
EIRQ15	importation	External Pin Interrupt Event Request 15

10.2 INTC System Block Diagram

10.2.1 system block diagram



10.3 vector table

10.3.1 interrupt vector table (computing)

Table 10-1 Interrupt vector table

address	vecto rs serial numb er	IRQ serial numb er	source of interruption	clarification
ARM core interrupt handling vector				
0x0000_0000	0	-	ARM core	Initial stack pointer
0x0000_0004	1	-	ARM core	Initial Program Counter
0x0000_0008	2	-	ARM core	Non-maskable Interrupt (NMI)
0x0000_000C	3	-	ARM core	Hard Fault
0x0000_0010	4	-	ARM core	MemManage Fault
0x0000_0014	5	-	ARM core	Bus Fault
0x0000_0018	6	-	ARM core	Usage Fault
0x0000_001C	7	-	ARM core	Reserved
0x0000_0020	8	-	ARM core	Reserved
0x0000_0024	9	-	ARM core	Reserved
0x0000_0028	10	-	ARM core	Reserved
0x0000_002C	11	-	ARM core	Supervisor call (SVCall)
0x0000_0030	12	-	ARM core	Debug Monitor
0x0000_0034	13	-	ARM core	Reserved
0x0000_0038	14	-	ARM core	Pendable request for system Service(PendableSrvReq)
0x0000_003C	15	-	ARM core	System tick timer (SysTick)
Non-ARM core interrupt handling vectors				
0x0000_0040	16	0	INT_SEL0	Interrupt event request/software interrupt selected by register INT_SEL0.
0x0000_0044	17	1	INT_SEL1	Interrupt event request/software interrupt selected by register INT_SEL1.
0x0000_0048	18	2	INT_SEL2	Interrupt event request/software interrupt selected by register INT_SEL2.
0x0000_004C	19	3	INT_SEL3	Interrupt event request/software interrupt selected by register INT_SEL3.
0x0000_0050	20	4	INT_SEL4	Interrupt event request/software interrupt selected by register INT_SEL4.
0x0000_0054	21	5	INT_SEL5	Interrupt event request/software interrupt selected by register INT_SEL5.
0x0000_0058	22	6	INT_SEL6	Interrupt event request/software interrupt selected by register INT_SEL6.

0x0000_005C	23	7	INT_SEL7	Interrupt event request/software interrupt selected by register INT_SEL7.
0x0000_0060	24	8	INT_SEL8	Interrupt event request/software interrupt selected by register INT_SEL8.
0x0000_0064	25	9	INT_SEL9	Interrupt event request/software interrupt selected by register INT_SEL9.
0x0000_0068	26	10	INT_SEL10	Interrupt event request/software interrupt selected by register INT_SEL10.
0x0000_006C	27	11	INT_SEL11	Interrupt event request/software interrupt selected by register INT_SEL11.
0x0000_0070	28	12	INT_SEL12	Interrupt event request/software interrupt selected by register INT_SEL12.

address	vecto rs serial numb er	IRQ serial numb er	source of interruption	clarification
0x0000_0074	29	13	INT_SEL13	Interrupt event request/software interrupt selected by register INT_SEL13.
0x0000_0078	30	14	INT_SEL14	Interrupt event request/software interrupt selected by register INT_SEL14.
0x0000_007C	31	15	INT_SEL15	Interrupt event request/software interrupt selected by register INT_SEL15.
0x0000_0080	32	16	INT_SEL16	Interrupt event request/software interrupt selected by register INT_SEL16.
0x0000_0084	33	17	INT_SEL17	Interrupt event request/software interrupt selected by register INT_SEL17.
0x0000_0088	34	18	INT_SEL18	Interrupt event request/software interrupt selected by register INT_SEL18.
0x0000_008C	35	19	INT_SEL19	Interrupt event request/software interrupt selected by register INT_SEL19.
0x0000_0090	36	20	INT_SEL20	Interrupt event request/software interrupt selected by register INT_SEL20.
0x0000_0094	37	21	INT_SEL21	Interrupt event request/software interrupt selected by register INT_SEL21.
0x0000_0098	38	22	INT_SEL22	Interrupt event request/software interrupt selected by register INT_SEL22.
0x0000_009C	39	23	INT_SEL23	Interrupt event request/software interrupt selected by register INT_SEL23.
0x0000_00A0	40	24	INT_SEL24	Interrupt event request/software interrupt selected by register INT_SEL24.
0x0000_00A4	41	25	INT_SEL25	Interrupt event request/software interrupt selected by register INT_SEL25.
0x0000_00A8	42	26	INT_SEL26	Interrupt event request/software interrupt selected by register INT_SEL26.
0x0000_00AC	43	27	INT_SEL27	Interrupt event request/software interrupt selected by register INT_SEL27.
0x0000_00B0	44	28	INT_SEL28	Interrupt event request/software interrupt selected by register INT_SEL28.
0x0000_00B4	45	29	INT_SEL29	Interrupt event request/software interrupt selected by register INT_SEL29.
0x0000_00B8	46	30	INT_SEL30	Interrupt event request/software interrupt selected by register INT_SEL30.
0x0000_00BC	47	31	INT_SEL31	Interrupt event request/software interrupt selected by register INT_SEL31.
0x0000_00C0	48	32	INT_SEL32	Interrupt event request selected by register INT_SEL32.

0x0000_00C4	49	33	INT_SEL33	Interrupt event request selected by register INT_SEL33.
0x0000_00C8	50	34	INT_SEL34	Interrupt event request selected by register INT_SEL34.
0x0000_00CC	51	35	INT_SEL35	Interrupt event request selected by register INT_SEL35.
0x0000_00D0	52	36	INT_SEL36	Interrupt event request selected by register INT_SEL36.
0x0000_00D4	53	37	INT_SEL37	Interrupt event request selected by register INT_SEL37.
0x0000_00D8	54	38	INT_SEL38	Interrupt event request selected by register INT_SEL38.
0x0000_00DC	55	39	INT_SEL39	Interrupt event request selected by register INT_SEL39.
0x0000_00E0	56	40	INT_SEL40	Interrupt event request selected by register INT_SEL40.
0x0000_00E4	57	41	INT_SEL41	Interrupt event request selected by register INT_SEL41.
0x0000_00E8	58	42	INT_SEL42	Interrupt event request selected by register INT_SEL42.
0x0000_00EC	59	43	INT_SEL43	Interrupt event request selected by register INT_SEL43.
0x0000_00F0	60	44	INT_SEL44	Interrupt event request selected by register INT_SEL44.
0x0000_00F4	61	45	INT_SEL45	Interrupt event request selected by register INT_SEL45.
0x0000_00F8	62	46	INT_SEL46	Interrupt event request selected by register INT_SEL46.
0x0000_00FC	63	47	INT_SEL47	Interrupt event request selected by register INT_SEL47.
0x0000_0100	64	48	INT_SEL48	Interrupt event request selected by register INT_SEL48.
0x0000_0104	65	49	INT_SEL49	Interrupt event request selected by register INT_SEL49.

address	vecto rs serial num ber	IRQ serial numb er	source of interruption	clarification
0x0000_0108	66	50	INT_SEL50	Interrupt event request selected by register INT_SEL50.
0x0000_010C	67	51	INT_SEL51	Interrupt event request selected by register INT_SEL51.
0x0000_0110	68	52	INT_SEL52	Interrupt event request selected by register INT_SEL52.
0x0000_0114	69	53	INT_SEL53	Interrupt event request selected by register INT_SEL53.
0x0000_0118	70	54	INT_SEL54	Interrupt event request selected by register INT_SEL54.
0x0000_011C	71	55	INT_SEL55	Interrupt event request selected by register INT_SEL55.
0x0000_0120	72	56	INT_SEL56	Interrupt event request selected by register INT_SEL56.
0x0000_0124	73	57	INT_SEL57	Interrupt event request selected by register INT_SEL57.
0x0000_0128	74	58	INT_SEL58	Interrupt event request selected by register INT_SEL58.
0x0000_012C	75	59	INT_SEL59	Interrupt event request selected by register INT_SEL59.
0x0000_0130	76	60	INT_SEL60	Interrupt event request selected by register INT_SEL60.
0x0000_0134	77	61	INT_SEL61	Interrupt event request selected by register INT_SEL61.
0x0000_0138	78	62	INT_SEL62	Interrupt event request selected by register INT_SEL62.
0x0000_013C	79	63	INT_SEL63	Interrupt event request selected by register INT_SEL63.
0x0000_0140	80	64	INT_SEL64	Interrupt event request selected by register INT_SEL64.
0x0000_0144	81	65	INT_SEL65	Interrupt event request selected by register INT_SEL65.
0x0000_0148	82	66	INT_SEL66	Interrupt event request selected by register INT_SEL66.
0x0000_014C	83	67	INT_SEL67	Interrupt event request selected by register INT_SEL67.
0x0000_0150	84	68	INT_SEL68	Interrupt event request selected by register INT_SEL68.
0x0000_0154	85	69	INT_SEL69	Interrupt event request selected by register INT_SEL69.
0x0000_0158	86	70	INT_SEL70	Interrupt event request selected by register INT_SEL70.
0x0000_015C	87	71	INT_SEL71	Interrupt event request selected by register INT_SEL71.
0x0000_0160	88	72	INT_SEL72	Interrupt event request selected by register INT_SEL72.
0x0000_0164	89	73	INT_SEL73	Interrupt event request selected by register INT_SEL73.
0x0000_0168	90	74	INT_SEL74	Interrupt event request selected by register INT_SEL74.
0x0000_016C	91	75	INT_SEL75	Interrupt event request selected by register INT_SEL75.
0x0000_0170	92	76	INT_SEL76	Interrupt event request selected by register INT_SEL76.
0x0000_0174	93	77	INT_SEL77	Interrupt event request selected by register INT_SEL77.
0x0000_0178	94	78	INT_SEL78	Interrupt event request selected by register INT_SEL78.
0x0000_017C	95	79	INT_SEL79	Interrupt event request selected by register INT_SEL79.
0x0000_0180	96	80	INT_SEL80	Interrupt event request selected by register INT_SEL80.
0x0000_0184	97	81	INT_SEL81	Interrupt event request selected by register INT_SEL81.
0x0000_0188	98	82	INT_SEL82	Interrupt event request selected by register INT_SEL82.
0x0000_018C	99	83	INT_SEL83	Interrupt event request selected by register INT_SEL83.
0x0000_0190	100	84	INT_SEL84	Interrupt event request selected by register INT_SEL84.
0x0000_0194	101	85	INT_SEL85	Interrupt event request selected by register INT_SEL85.

0x0000_0198	102	86	INT_SEL86	Interrupt event request selected by register INT_SEL86.
-------------	-----	----	-----------	---

address	vecto rs serial numb er	IRQ serial numb er	source of interruption	clarification
0x0000_019C	103	87	INT_SEL87	Interrupt event request selected by register INT_SEL87.
0x0000_01A0	104	88	INT_SEL88	Interrupt event request selected by register INT_SEL88.
0x0000_01A4	105	89	INT_SEL89	Interrupt event request selected by register INT_SEL89.
0x0000_01A8	106	90	INT_SEL90	Interrupt event request selected by register INT_SEL90.
0x0000_01AC	107	91	INT_SEL91	Interrupt event request selected by register INT_SEL91.
0x0000_01B0	108	92	INT_SEL92	Interrupt event request selected by register INT_SEL92.
0x0000_01B4	109	93	INT_SEL93	Interrupt event request selected by register INT_SEL93.
0x0000_01B8	110	94	INT_SEL94	Interrupt event request selected by register INT_SEL94.
0x0000_01BC	111	95	INT_SEL95	Interrupt event request selected by register INT_SEL95.
0x0000_01C0	112	96	INT_SEL96	Interrupt event request selected by register INT_SEL96.
0x0000_01C4	113	97	INT_SEL97	Interrupt event request selected by register INT_SEL97.
0x0000_01C8	114	98	INT_SEL98	Interrupt event request selected by register INT_SEL98.
0x0000_01CC	115	99	INT_SEL99	Interrupt event request selected by register INT_SEL99.
0x0000_01D0	116	100	INT_SEL100	Interrupt event request selected by register INT_SEL100.
0x0000_01D4	117	101	INT_SEL101	Interrupt event request selected by register INT_SEL101.
0x0000_01D8	118	102	INT_SEL102	Interrupt event request selected by register INT_SEL102.
0x0000_01DC	119	103	INT_SEL103	Interrupt event request selected by register INT_SEL103.
0x0000_01E0	120	104	INT_SEL104	Interrupt event request selected by register INT_SEL104.
0x0000_01E4	121	105	INT_SEL105	Interrupt event request selected by register INT_SEL105.
0x0000_01E8	122	106	INT_SEL106	Interrupt event request selected by register INT_SEL106.
0x0000_01EC	123	107	INT_SEL107	Interrupt event request selected by register INT_SEL107.
0x0000_01F0	124	108	INT_SEL108	Interrupt event request selected by register INT_SEL108.
0x0000_01F4	125	109	INT_SEL109	Interrupt event request selected by register INT_SEL109.
0x0000_01F8	126	110	INT_SEL110	Interrupt event request selected by register INT_SEL110.
0x0000_01FC	127	111	INT_SEL111	Interrupt event request selected by register INT_SEL111.
0x0000_0200	128	112	INT_SEL112	Interrupt event request selected by register INT_SEL112.
0x0000_0204	129	113	INT_SEL113	Interrupt event request selected by register INT_SEL113.
0x0000_0208	130	114	INT_SEL114	Interrupt event request selected by register INT_SEL114.
0x0000_020C	131	115	INT_SEL115	Interrupt event request selected by register INT_SEL115.
0x0000_0210	132	116	INT_SEL116	Interrupt event request selected by register INT_SEL116.
0x0000_0214	133	117	INT_SEL117	Register INT_SEL117 Selected interrupt event request.
0x0000_0218	134	118	INT_SEL118	Interrupt event request selected by register INT_SEL118.
0x0000_021C	135	119	INT_SEL119	Register INT_SEL119 Selected interrupt event request.
0x0000_0220	136	120	INT_SEL120	Interrupt event request selected by register INT_SEL120.
0x0000_0224	137	121	INT_SEL121	Interrupt event request selected by register INT_SEL121.
0x0000_0228	138	122	INT_SEL122	Interrupt event request selected by register INT_SEL122.

0x0000_022C	139	123	INT_SEL123	Interrupt event request selected by register INT_SEL123.
-------------	-----	-----	------------	--

address	vecto rs serial num ber	IRQ serial numb er	source of interruption	clarification
0x0000_0230	140	124	INT_SEL124	Interrupt event request selected by register INT_SEL124.
0x0000_0234	141	125	INT_SEL125	Interrupt event request selected by register INT_SEL125.
0x0000_0238	142	126	INT_SEL126	Interrupt event request selected by register INT_SEL126.
0x0000_023C	143	127	INT_SEL127	Interrupt event request selected by register INT_SEL127.
0x0000_0240	144	128	INT_VSSEL128	Register INT_VSSEL128 Interrupt Event Requests Selected by Enable Bit Total Use this vector.
0x0000_0244	145	129	INT_VSSEL129	Register INT_VSSEL129 Interrupt Event Request Selected by Enable Bit Total Use this vector.
0x0000_0248	146	130	INT_VSSEL130	Register INT_VSSEL130 Interrupt Event Request Selected by Enable Bit Total Use this vector.
0x0000_024C	147	131	INT_VSSEL131	Register INT_VSSEL131 Interrupt Event Request Selected by Enable Bit Total Use this vector.
0x0000_0250	148	132	INT_VSSEL132	Register INT_VSSEL132 Interrupt Event Requests Selected by Enable Bit Total Use this vector.
0x0000_0254	149	133	INT_VSSEL133	Register INT_VSSEL133 Interrupt Event Request Selected by Enable Bit Total Use this vector.
0x0000_0258	150	134	INT_VSSEL134	Register INT_VSSEL134 Interrupt Event Request Selected by Enable Bit Total Use this vector.
0x0000_025C	151	135	INT_VSSEL135	Register INT_VSSEL135 Interrupt Event Requests Selected by Enable Bit Total Use this vector.
0x0000_0260	152	136	INT_VSSEL136	Register INT_VSSEL136 Interrupt Event Request Selected by Enable Bit Total Use this vector.
0x0000_0264	153	137	INT_VSSEL137	Register INT_VSSEL137 Interrupt Event Request Selected by Enable Bit Total Use this vector.
0x0000_0268	154	138	INT_VSSEL138	Register INT_VSSEL138 Interrupt Event Request Selected by Enable Bit Total Use this vector.

0x0000_026C	155	139	INT_VSSEL139	Register INT_VSSEL139 Interrupt Event Request Selected by Enable Bit Total Use this vector.
0x0000_0270	156	140	INT_VSSEL140	Register INT_VSSEL140 Interrupt Event Requests Selected by Enable Bit Total Use this vector.
0x0000_0274	157	141	INT_VSSEL141	Register INT_VSSEL141 Interrupt Event Request Selected by Enable Bit Total Use this vector.
0x0000_0278	158	142	INT_VSSEL142	Register INT_VSSEL142 Interrupt Event Request Selected by Enable Bit Total Use this vector.
0x0000_027C	159	143	INT_VSSEL143	Register INT_VSSEL143 Interrupt Event Request Selected by Enable Bit Total Use this vector.

Note: Refer to the Register Description section for the specific interrupt event request number selected.

10.3.2 Interrupt event request number

Interrupt event requests are generated by the peripheral. Interrupt event requests are referred to as interrupt sources when they are selected as inputs to the NVIC by the interrupt controller, and as event sources when they are selected as event inputs. The peripheral interrupt event request can also be used as a condition for the MCU low-power mode return.

Table 10-2 Interrupt Event Request Sequence Number and Selection

serial num ber	Interr upt event reque st numb er	functio nality	Function Name	Wh ether or not it can be sele cte d as an inte rru pt	Opt ion for inte rnal tou ch deriv ation	Interrupt selection register corresponding to NVIC vector ^{*1}		
						NVIC Vector 0~31	NVIC Vector 32~127	NVIC Vector 128~143
0	000h	PORT	PORT_EIRQ0	✓	✓	INT_SEL0~31	INT_SEL32~37	INT_VSEL128[0]
1	001h		PORT_EIRQ1	✓	✓	INT_SEL0~31	INT_SEL32~37	INT_VSEL128[1]
2	002h		PORT_EIRQ2	✓	✓	INT_SEL0~31	INT_SEL32~37	INT_VSEL128[2]
3	003h		PORT_EIRQ3	✓	✓	INT_SEL0~31	INT_SEL32~37	INT_VSEL128[3]
4	004h		PORT_EIRQ4	✓	✓	INT_SEL0~31	INT_SEL32~37	INT_VSEL128[4]
5	005h		PORT_EIRQ5	✓	✓	INT_SEL0~31	INT_SEL32~37	INT_VSEL128[5]
6	006h		PORT_EIRQ6	✓	✓	INT_SEL0~31	INT_SEL32~37	INT_VSEL128[6]
7	007h		PORT_EIRQ7	✓	✓	INT_SEL0~31	INT_SEL32~37	INT_VSEL128[7]
8	008h		PORT_EIRQ8	✓	✓	INT_SEL0~31	INT_SEL32~37	INT_VSEL128[8]
9	009h		PORT_EIRQ9	✓	✓	INT_SEL0~31	INT_SEL32~37	INT_VSEL128[9]
10	00Ah		PORT_EIRQ10	✓	✓	INT_SEL0~31	INT_SEL32~37	INT_VSEL128[10]
11	00Bh		PORT_EIRQ11	✓	✓	INT_SEL0~31	INT_SEL32~37	INT_VSEL128[11]
12	00Ch		PORT_EIRQ12	✓	✓	INT_SEL0~31	INT_SEL32~37	INT_VSEL128[12]
13	00Dh		PORT_EIRQ13	✓	✓	INT_SEL0~31	INT_SEL32~37	INT_VSEL128[13]
14	00Eh		PORT_EIRQ14	✓	✓	INT_SEL0~31	INT_SEL32~37	INT_VSEL128[14]
15	00Fh		PORT_EIRQ15	✓	✓	INT_SEL0~31	INT_SEL32~37	INT_VSEL128[15]
16	010h	-	-	-	-	INT_SEL0~31	INT_SEL32~37	INT_VSEL128[16]
17	011h	-	-	-	-	INT_SEL0~31	INT_SEL32~37	INT_VSEL128[17]
18	012h	-	-	-	-	INT_SEL0~31	INT_SEL32~37	INT_VSEL128[18]
19	013h	-	-	-	-	INT_SEL0~31	INT_SEL32~37	INT_VSEL128[19]
20	014h	-	-	-	-	INT_SEL0~31	INT_SEL32~37	INT_VSEL128[20]
21	015h	-	-	-	-	INT_SEL0~31	INT_SEL32~37	INT_VSEL128[21]
22	016h	-	-	-	-	INT_SEL0~31	INT_SEL32~37	INT_VSEL128[22]
23	017h	-	-	-	-	INT_SEL0~31	INT_SEL32~37	INT_VSEL128[23]
24	018h	-	-	-	-	INT_SEL0~31	INT_SEL32~37	INT_VSEL128[24]
25	019h	-	-	-	-	INT_SEL0~31	INT_SEL32~37	INT_VSEL128[25]

26	01Ah	-	-	-	-	INT_SEL0~31	INT_SEL32~37	INT_VSSEL128[26]
27	01Bh	-	-	-	-	INT_SEL0~31	INT_SEL32~37	INT_VSSEL128[27]
28	01Ch	-	-	-	-	INT_SEL0~31	INT_SEL32~37	INT_VSSEL128[28]
29	01Dh	-	-	-	-	INT_SEL0~31	INT_SEL32~37	INT_VSSEL128[29]
30	01Eh	-	-	-	-	INT_SEL0~31	INT_SEL32~37	INT_VSSEL128[30]
31	01Fh	-	-	-	-	INT_SEL0~31	INT_SEL32~37	INT_VSSEL128[31]
32	020h	DMA	DMA1_TC0	✓	✓	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[0]
33	021h		DMA1_TC1	✓	✓	INT_SEL0~31	INT_SEL38~43	INT_VSSEL129[1]

serial num ber	Interr upt event reques t numb er	functio nality	Function Name	Wh ether or not it can be sele cte d as an inte rru pt	Can you cho ose for the inte rnal tou ch deriv ation	Interrupt selection register corresponding to NVIC vector *1		
						NVIC Vector 0~31	NVIC Vector 32~127	NVIC Vector 128~143
34	022h	DMA	DMA1_TC2	✓	✓	INT_SEL0~31	INT_SEL38~43	INT_VSEL129[2]
35	023h		DMA1_TC3	✓	✓	INT_SEL0~31	INT_SEL38~43	INT_VSEL129[3]
36	024h		DMA2_TC0	✓	✓	INT_SEL0~31	INT_SEL38~43	INT_VSEL129[4]
37	025h		DMA2_TC1	✓	✓	INT_SEL0~31	INT_SEL38~43	INT_VSEL129[5]
38	026h		DMA2_TC2	✓	✓	INT_SEL0~31	INT_SEL38~43	INT_VSEL129[6]
39	027h		DMA2_TC3	✓	✓	INT_SEL0~31	INT_SEL38~43	INT_VSEL129[7]
40	028h		DMA1_BT0	✓	✓	INT_SEL0~31	INT_SEL38~43	INT_VSEL129[8]
41	029h		DMA1_BT1	✓	✓	INT_SEL0~31	INT_SEL38~43	INT_VSEL129[9]
42	02Ah		DMA1_BT2	✓	✓	INT_SEL0~31	INT_SEL38~43	INT_VSEL129[10]
43	02Bh		DMA1_BT3	✓	✓	INT_SEL0~31	INT_SEL38~43	INT_VSEL129[11]
44	02Ch		DMA2_BT0	✓	✓	INT_SEL0~31	INT_SEL38~43	INT_VSEL129[12]
45	02Dh		DMA2_BT1	✓	✓	INT_SEL0~31	INT_SEL38~43	INT_VSEL129[13]
46	02Eh		DMA2_BT2	✓	✓	INT_SEL0~31	INT_SEL38~43	INT_VSEL129[14]
47	02Fh		DMA2_BT3	✓	✓	INT_SEL0~31	INT_SEL38~43	INT_VSEL129[15]
48	030h	DMA	DMA1_ERR	✓	-	INT_SEL0~31	INT_SEL38~43	INT_VSEL129[16]
49	031h		DMA2_ERR	✓	-	INT_SEL0~31	INT_SEL38~43	INT_VSEL129[17]
50	032h	EFM	EFM_PEERR	✓	-	INT_SEL0~31	INT_SEL38~43	INT_VSEL129[18]
51	033h		EFM_COLERR	✓	-	INT_SEL0~31	INT_SEL38~43	INT_VSEL129[19]
52	034h		EFM_OPTEND	✓	✓	INT_SEL0~31	INT_SEL38~43	INT_VSEL129[20]
53	035h	USBFS	USBFS_SOF	-	✓	INT_SEL0~31	INT_SEL38~43	INT_VSEL129[21]
54	036h	QSPI	QSPI_INTR	✓	-	INT_SEL0~31	INT_SEL38~43	INT_VSEL129[22]
55	037h	DCU	DCU1	✓	✓	INT_SEL0~31	INT_SEL38~43	INT_VSEL129[23]
56	038h		DCU2	✓	✓	INT_SEL0~31	INT_SEL38~43	INT_VSEL129[24]
57	039h		DCU3	✓	✓	INT_SEL0~31	INT_SEL38~43	INT_VSEL129[25]
58	03Ah		DCU4	✓	✓	INT_SEL0~31	INT_SEL38~43	INT_VSEL129[26]
59	03Bh	-	-	-	-	INT_SEL0~31	INT_SEL38~43	INT_VSEL129[27]
60	03Ch	-	-	-	-	INT_SEL0~31	INT_SEL38~43	INT_VSEL129[28]
61	03Dh	-	-	-	-	INT_SEL0~31	INT_SEL38~43	INT_VSEL129[29]
62	03Eh	-	-	-	-	INT_SEL0~31	INT_SEL38~43	INT_VSEL129[30]
63	03Fh	-	-	-	-	INT_SEL0~31	INT_SEL38~43	INT_VSEL129[31]
64	040h	Timer0 _1	TMR01_GCMA	✓	✓	INT_SEL0~31	INT_SEL44~49	INT_VSEL130[0]
65	041h		TMR01_GCMB	✓	✓	INT_SEL0~31	INT_SEL44~49	INT_VSEL130[1]
66	042h	Timer0	TMR02_GCMA	✓	✓	INT_SEL0~31	INT_SEL44~49	INT_VSEL130[2]

67	043h	_2	TMR02_GCMB	✓	✓	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[3]
68	044h	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[4]
69	045h	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[5]
70	046h	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[6]
71	047h	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[7]
72	048h	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[8]
73	049h	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSSEL130[9]

serial num ber	Interr upt event reques t numb er	functio nality	Function Name	Wh ether or not it can be sele cte d as an inte rru pt	Can you cho ose for the inte rnal tou ch deriv ation	Interrupt selection register corresponding to NVIC vector *1		
						NVIC Vector 0~31	NVIC Vector 32~127	NVIC Vector 128~143
74	04Ah	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSEL130[10]
75	04Bh	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSEL130[11]
76	04Ch	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSEL130[12]
77	04Dh	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSEL130[13]
78	04Eh	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSEL130[14]
79	04Fh	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSEL130[15]
80	050h	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSEL130[16]
81	051h	RTC	RTC_ALM	✓	✓	INT_SEL0~31	INT_SEL44~49	-
82	052h		RTC_PRD	✓	✓	INT_SEL0~31	INT_SEL44~49	INT_VSEL130[18]
83	053h		-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSEL130[19]
84	054h	XTAL32	XTAL32_STOP	✓	-	INT_SEL0~31	INT_SEL44~49	-
85	055h	XTAL	XTAL_STOP	✓	-	INT_SEL0~31	INT_SEL44~49	INT_VSEL130[21]
86	056h	WKTM	WKTM_PRD	✓	-	INT_SEL0~31	INT_SEL44~49	INT_VSEL130[22]
87	057h	SWDT	SWDT_REFUDF	✓	-	INT_SEL0~31	INT_SEL44~49	INT_VSEL130[23]
88	058h	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSEL130[24]
89	059h	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSEL130[25]
90	05Ah	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSEL130[26]
91	05Bh	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSEL130[27]
92	05Ch	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSEL130[28]
93	05Dh	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSEL130[29]
94	05Eh	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSEL130[30]
95	05Fh	-	-	-	-	INT_SEL0~31	INT_SEL44~49	INT_VSEL130[31]
96	060h	Timer6 _1	TMR61_GCMA	✓	✓	INT_SEL0~31	INT_SEL50~55	INT_VSEL131[0]
97	061h		TMR61_GCMB	✓	✓	INT_SEL0~31	INT_SEL50~55	INT_VSEL131[1]
98	062h		TMR61_GCMC	✓	✓	INT_SEL0~31	INT_SEL50~55	INT_VSEL131[2]
99	063h		TMR61_GCMD	✓	✓	INT_SEL0~31	INT_SEL50~55	INT_VSEL131[3]
100	064h		TMR61_GCME	✓	✓	INT_SEL0~31	INT_SEL50~55	INT_VSEL131[4]
101	065h		TMR61_GCMF	✓	✓	INT_SEL0~31	INT_SEL50~55	INT_VSEL131[5]
102	066h		TMR61_GOVF	✓	✓	INT_SEL0~31	INT_SEL50~55	INT_VSEL131[6]
103	067h		TMR61_GUDF	✓	✓	INT_SEL0~31	INT_SEL50~55	INT_VSEL131[7]
104	068h		TMR61_GDTE	✓	-	INT_SEL0~31	INT_SEL50~55	INT_VSEL131[8]
105	069h		-	-	-	INT_SEL0~31	INT_SEL50~55	INT_VSEL131[9]
106	06Ah		-	-	-	INT_SEL0~31	INT_SEL50~55	INT_VSEL131[10]
107	06Bh		TMR61_SCMA	✓	✓	INT_SEL0~31	INT_SEL50~55	INT_VSEL131[11]

108	06Ch		TMR61_SCMB	✓	✓	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[12]
109	06Dh	-	-	-	-	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[13]
110	06Eh	-	-	-	-	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[14]
111	06Fh	-	-	-	-	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[15]
112	070h	Timer6_2	TMR62_GCMA	✓	✓	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[16]
			TMR62_GCMB	✓	✓	INT_SEL0~31	INT_SEL50~55	INT_VSSEL131[17]

serial number	Interrupt request number	functionality	Function Name	Whether or not it can be selected as an interrupt	Can you choose for the internal touch derivation	Interrupt selection register corresponding to NVIC vector *1		
						NVIC Vector 0~31	NVIC Vector 32~127	NVIC Vector 128~143
114	072h	Timer6_3	TMR62_GCMC	✓	✓	INT_SEL0~31	INT_SEL50~55	INT_VSEL131[18]
115	073h		TMR62_GCMD	✓	✓	INT_SEL0~31	INT_SEL50~55	INT_VSEL131[19]
116	074h		TMR62_GCME	✓	✓	INT_SEL0~31	INT_SEL50~55	INT_VSEL131[20]
117	075h		TMR62_GCMF	✓	✓	INT_SEL0~31	INT_SEL50~55	INT_VSEL131[21]
118	076h		TMR62_GOVF	✓	✓	INT_SEL0~31	INT_SEL50~55	INT_VSEL131[22]
119	077h		TMR62_GUDF	✓	✓	INT_SEL0~31	INT_SEL50~55	INT_VSEL131[23]
120	078h		TMR62_GDTE	✓	-	INT_SEL0~31	INT_SEL50~55	INT_VSEL131[24]
121	079h		-	-	-	INT_SEL0~31	INT_SEL50~55	INT_VSEL131[25]
122	07Ah		-	-	-	INT_SEL0~31	INT_SEL50~55	INT_VSEL131[26]
123	07Bh		TMR62_SCMA	✓	✓	INT_SEL0~31	INT_SEL50~55	INT_VSEL131[27]
124	07Ch		TMR62_SCMB	✓	✓	INT_SEL0~31	INT_SEL50~55	INT_VSEL131[28]
125	07Dh		-	-	-	INT_SEL0~31	INT_SEL50~55	INT_VSEL131[29]
126	07Eh		-	-	-	INT_SEL0~31	INT_SEL50~55	INT_VSEL131[30]
127	07Fh		-	-	-	INT_SEL0~31	INT_SEL50~55	INT_VSEL131[31]
128	080h	Timer6_3	TMR63_GCMA	✓	✓	INT_SEL0~31	INT_SEL56~61	INT_VSEL132[0]
129	081h		TMR63_GCMB	✓	✓	INT_SEL0~31	INT_SEL56~61	INT_VSEL132[1]
130	082h		TMR63_GCMC	✓	✓	INT_SEL0~31	INT_SEL56~61	INT_VSEL132[2]
131	083h		TMR63_GCMD	✓	✓	INT_SEL0~31	INT_SEL56~61	INT_VSEL132[3]
132	084h		TMR63_GCME	✓	✓	INT_SEL0~31	INT_SEL56~61	INT_VSEL132[4]
133	085h		TMR63_GCMF	✓	✓	INT_SEL0~31	INT_SEL56~61	INT_VSEL132[5]
134	086h		TMR63_GOVF	✓	✓	INT_SEL0~31	INT_SEL56~61	INT_VSEL132[6]
135	087h		TMR63_GUDF	✓	✓	INT_SEL0~31	INT_SEL56~61	INT_VSEL132[7]
136	088h		TMR63_GDTE	✓	-	INT_SEL0~31	INT_SEL56~61	INT_VSEL132[8]
137	089h		-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSEL132[9]
138	08Ah		-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSEL132[10]
139	08Bh		TMR63_SCMA	✓	✓	INT_SEL0~31	INT_SEL56~61	INT_VSEL132[11]
140	08Ch		TMR63_SCMB	✓	✓	INT_SEL0~31	INT_SEL56~61	INT_VSEL132[12]
141	08Dh		-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSEL132[13]
142	08Eh		-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSEL132[14]
143	08Fh		-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSEL132[15]
144	090h		-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSEL132[16]
145	091h		-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSEL132[17]
146	092h		-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSEL132[18]
147	093h		-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSEL132[19]

148	094h	-	-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[20]
149	095h	-	-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[21]
150	096h	-	-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[22]
151	097h	-	-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[23]
152	098h	-	-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[24]
153	099h	-	-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[25]
154	09Ah	-	-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSSEL132[26]

serial number	Interrupt request number	functionality	Function Name	Whether or not it can be selected as an interrupt	Can you choose for the internal touch derivation	Interrupt selection register corresponding to NVIC vector *1		
						NVIC Vector 0~31	NVIC Vector 32~127	NVIC Vector 128~143
155	09Bh	-	-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSEL132[27]
156	09Ch	-	-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSEL132[28]
157	09Dh	-	-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSEL132[29]
158	09Eh	-	-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSEL132[30]
159	09Fh	-	-	-	-	INT_SEL0~31	INT_SEL56~61	INT_VSEL132[31]
160	0A0h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSEL133[0]
161	0A1h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSEL133[1]
162	0A2h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSEL133[2]
163	0A3h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSEL133[3]
164	0A4h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSEL133[4]
165	0A5h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSEL133[5]
166	0A6h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSEL133[6]
167	0A7h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSEL133[7]
168	0A8h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSEL133[8]
169	0A9h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSEL133[9]
170	0AAh	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSEL133[10]
171	0ABh	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSEL133[11]
172	0ACH	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSEL133[12]
173	0ADh	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSEL133[13]
174	0AEh	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSEL133[14]
175	0AFh	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSEL133[15]
176	0B0h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSEL133[16]
177	0B1h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSEL133[17]
178	0B2h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSEL133[18]
179	0B3h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSEL133[19]
180	0B4h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSEL133[20]
181	0B5h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSEL133[21]
182	0B6h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSEL133[22]
183	0B7h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSEL133[23]
184	0B8h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSEL133[24]
185	0B9h	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSEL133[25]
186	0BAh	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSEL133[26]
187	0BBh	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSEL133[27]
188	0BCh	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSEL133[28]
189	0BDh	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSEL133[29]

190	0BEh	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[30]
191	0BFh	-	-	-	-	INT_SEL0~31	INT_SEL62~67	INT_VSSEL133[31]
192	0C0h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[0]
193	0C1h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[1]
194	0C2h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[2]
195	0C3h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[3]
196	0C4h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSSEL134[4]

serial number	Interrrupt event request number	functionality	Function Name	Whether or not it can be selected as an interrupt	Can you choose for the internal touch derivation	Interrupt selection register corresponding to NVIC vector *1		
						NVIC Vector 0~31	NVIC Vector 32~127	NVIC Vector 128~143
197	0C5h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSEL134[5]
198	0C6h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSEL134[6]
199	0C7h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSEL134[7]
200	0C8h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSEL134[8]
201	0C9h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSEL134[9]
202	0CAh	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSEL134[10]
203	0CBh	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSEL134[11]
204	0CCh	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSEL134[12]
205	0CDh	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSEL134[13]
206	0CEh	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSEL134[14]
207	0CFh	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSEL134[15]
208	0D0h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSEL134[16]
209	0D1h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSEL134[17]
210	0D2h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSEL134[18]
211	0D3h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSEL134[19]
212	0D4h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSEL134[20]
213	0D5h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSEL134[21]
214	0D6h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSEL134[22]
215	0D7h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSEL134[23]
216	0D8h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSEL134[24]
217	0D9h	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSEL134[25]
218	0DAh	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSEL134[26]
219	0DBh	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSEL134[27]
220	0DCh	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSEL134[28]
221	0DDh	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSEL134[29]
222	0DEh	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSEL134[30]
223	0DFh	-	-	-	-	INT_SEL0~31	INT_SEL68~73	INT_VSEL134[31]
224	0E0h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSEL135[0]
225	0E1h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSEL135[1]
226	0E2h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSEL135[2]
227	0E3h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSEL135[3]
228	0E4h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSEL135[4]
229	0E5h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSEL135[5]
230	0E6h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSEL135[6]
231	0E7h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSEL135[7]

232	0E8h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[8]
233	0E9h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[9]
234	0EAh	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[10]
235	0EBh	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[11]
236	0Ec _h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[12]
237	0Ed _h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[13]
238	0E _{Eh}	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSSEL135[14]

serial number	Interrrupt event request number	functionality	Function Name	Whether or not it can be selected as an interrupt	Can you choose for the internal touch derivation	Interrupt selection register corresponding to NVIC vector *1		
						NVIC Vector 0~31	NVIC Vector 32~127	NVIC Vector 128~143
239	0EEh	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSEL135[15]
240	0F0h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSEL135[16]
241	0F1h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSEL135[17]
242	0F2h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSEL135[18]
243	0F3h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSEL135[19]
244	0F4h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSEL135[20]
245	0F5h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSEL135[21]
246	0F6h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSEL135[22]
247	0F7h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSEL135[23]
248	0F8h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSEL135[24]
249	0F9h	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSEL135[25]
250	0FAh	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSEL135[26]
251	0FBh	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSEL135[27]
252	0FcH	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSEL135[28]
253	0FdH	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSEL135[29]
254	0FeH	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSEL135[30]
255	0FFh	-	-	-	-	INT_SEL0~31	INT_SEL74~79	INT_VSEL135[31]
256	100h	TimerA_1	TMRA1_OVF	✓	✓	INT_SEL0~31	INT_SEL80~85	INT_VSEL136[0]
257	101h		TMRA1_UDF	✓	✓	INT_SEL0~31	INT_SEL80~85	INT_VSEL136[1]
258	102h		TMRA1_CMP	✓	✓	INT_SEL0~31	INT_SEL80~85	INT_VSEL136[2]
259	103h	TimerA_2	TMRA2_OVF	✓	✓	INT_SEL0~31	INT_SEL80~85	INT_VSEL136[3]
260	104h		TMRA2_UDF	✓	✓	INT_SEL0~31	INT_SEL80~85	INT_VSEL136[4]
261	105h		TMRA2_CMP	✓	✓	INT_SEL0~31	INT_SEL80~85	INT_VSEL136[5]
262	106h	TimerA_3	TMRA3_OVF	✓	✓	INT_SEL0~31	INT_SEL80~85	INT_VSEL136[6]
263	107h		TMRA3_UDF	✓	✓	INT_SEL0~31	INT_SEL80~85	INT_VSEL136[7]
264	108h		TMRA3_CMP	✓	✓	INT_SEL0~31	INT_SEL80~85	INT_VSEL136[8]
265	109h	TimerA_4	TMRA4_OVF	✓	✓	INT_SEL0~31	INT_SEL80~85	INT_VSEL136[9]
266	10Ah		TMRA4_UDF	✓	✓	INT_SEL0~31	INT_SEL80~85	INT_VSEL136[10]
267	10Bh		TMRA4_CMP	✓	✓	INT_SEL0~31	INT_SEL80~85	INT_VSEL136[11]
268	10Ch	TimerA	TMRA5_OVF	✓	✓	INT_SEL0~31	INT_SEL80~85	INT_VSEL136[12]

269	10Dh	_5	TMRA5_UDF	✓	✓	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[13]
270	10Eh		TMRA5_CMP	✓	✓	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[14]
271	10Fh	-	-	-	-	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[15]
272	110h	TimerA _6	TMRA6_OVF	✓	✓	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[16]
273	111h		TMRA6_UDF	✓	✓	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[17]
274	112h		TMRA6_CMP	✓	✓	INT_SEL0~31	INT_SEL80~85	INT_VSSEL136[18]

serial num ber	Interr upt event reques t numb er	functio nality	Function Name	Wh ether or not it can be sele cte d as an inte rru pt	Can you cho ose for the inte rnal tou ch deriv ation	Interrupt selection register corresponding to NVIC vector *1		
						NVIC Vector 0~31	NVIC Vector 32~127	NVIC Vector 128~143
275	113h	USBFS	USBFS_GLB	✓	-	INT_SEL0~31	INT_SEL80~85	INT_VSEL136[19]
276	114h	-	-	-	-	INT_SEL0~31	INT_SEL80~85	INT_VSEL136[20]
277	115h	-	-	-	-	INT_SEL0~31	INT_SEL80~85	INT_VSEL136[21]
278	116h	USART1	USART1_REI	✓	✓	INT_SEL0~31	INT_SEL80~85	INT_VSEL136[22]
279	117h		USART1_RI	✓	✓	INT_SEL0~31	INT_SEL80~85	INT_VSEL136[23]
280	118h		USART1_TI	✓	✓	INT_SEL0~31	INT_SEL80~85	INT_VSEL136[24]
281	119h		USART1_TCI	✓	✓	INT_SEL0~31	INT_SEL80~85	INT_VSEL136[25]
282	11Ah		USART1_RTOI	✓	✓	INT_SEL0~31	INT_SEL80~85	INT_VSEL136[26]
283	11Bh	USART2	USART2_REI	✓	✓	INT_SEL0~31	INT_SEL80~85	INT_VSEL136[27]
284	11Ch		USART2_RI	✓	✓	INT_SEL0~31	INT_SEL80~85	INT_VSEL136[28]
285	11Dh		USART2_TI	✓	✓	INT_SEL0~31	INT_SEL80~85	INT_VSEL136[29]
286	11Eh		USART2_TCI	✓	✓	INT_SEL0~31	INT_SEL80~85	INT_VSEL136[30]
287	11Fh		USART2_RTOI	✓	✓	INT_SEL0~31	INT_SEL80~85	INT_VSEL136[31]
288	120h	USART3	USART3_REI	✓	✓	INT_SEL0~31	INT_SEL86~91	INT_VSEL137[0]
289	121h		USART3_RI	✓	✓	INT_SEL0~31	INT_SEL86~91	INT_VSEL137[1]
290	122h		USART3_TI	✓	✓	INT_SEL0~31	INT_SEL86~91	INT_VSEL137[2]
291	123h		USART3_TCI	✓	✓	INT_SEL0~31	INT_SEL86~91	INT_VSEL137[3]
292	124h		USART3_RTOI	✓	✓	INT_SEL0~31	INT_SEL86~91	INT_VSEL137[4]
293	125h	USART4	USART4_REI	✓	✓	INT_SEL0~31	INT_SEL86~91	INT_VSEL137[5]
294	126h		USART4_RI	✓	✓	INT_SEL0~31	INT_SEL86~91	INT_VSEL137[6]
295	127h		USART4_TI	✓	✓	INT_SEL0~31	INT_SEL86~91	INT_VSEL137[7]
296	128h		USART4_TCI	✓	✓	INT_SEL0~31	INT_SEL86~91	INT_VSEL137[8]
297	129h		USART4_RTOI	✓	✓	INT_SEL0~31	INT_SEL86~91	INT_VSEL137[9]
298	12Ah	-	-	-	-	INT_SEL0~31	INT_SEL86~91	INT_VSEL137[10]
299	12Bh	SPI1	SPI1_SPRI	✓	✓	INT_SEL0~31	INT_SEL86~91	INT_VSEL137[11]
300	12Ch		SPI1_SPTI	✓	✓	INT_SEL0~31	INT_SEL86~91	INT_VSEL137[12]
301	12Dh		SPI1_SPII	✓	✓	INT_SEL0~31	INT_SEL86~91	INT_VSEL137[13]
302	12Eh		SPI1_SPEI	✓	✓	INT_SEL0~31	INT_SEL86~91	INT_VSEL137[14]
303	12Fh		SPI1_SPTEND	-	✓	INT_SEL0~31	INT_SEL86~91	INT_VSEL137[15]
304	130h	SPI2	SPI2_SPRI	✓	✓	INT_SEL0~31	INT_SEL86~91	INT_VSEL137[16]
305	131h		SPI2_SPTI	✓	✓	INT_SEL0~31	INT_SEL86~91	INT_VSEL137[17]
306	132h		SPI2_SPII	✓	✓	INT_SEL0~31	INT_SEL86~91	INT_VSEL137[18]
307	133h		SPI2_SPEI	✓	✓	INT_SEL0~31	INT_SEL86~91	INT_VSEL137[19]
308	134h		SPI2_SPTEND	-	✓	INT_SEL0~31	INT_SEL86~91	INT_VSEL137[20]
309	135h	SPI3	SPI3_SPRI	✓	✓	INT_SEL0~31	INT_SEL86~91	INT_VSEL137[21]

310	136h	SPI4	SPI3_SPTI	✓	✓	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[22]
311	137h		SPI3_SPII	✓	✓	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[23]
312	138h		SPI3_SPEI	✓	✓	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[24]
313	139h		SPI3_SPTEND	-	✓	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[25]
314	13Ah		SPI4_SPRI	✓	✓	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[26]
315	13Bh		SPI4_SPTI	✓	✓	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[27]
316	13Ch		SPI4_SPII	✓	✓	INT_SEL0~31	INT_SEL86~91	INT_VSSEL137[28]

serial num ber	Interr upt event reques t numb er	functio nality	Function Name	Wh ether or not it can be sele cte d as an inte rru pt	Can you cho ose for the inte rnal tou ch deriv ation	Interrupt selection register corresponding to NVIC vector *1		
						NVIC Vector 0~31	NVIC Vector 32~127	NVIC Vector 128~143
317	13Dh		SPI4_SPEI	✓	✓	INT_SEL0~31	INT_SEL86~91	INT_VSEL137[29]
318	13Eh		SPI4_SPTEND	-	✓	INT_SEL0~31	INT_SEL86~91	INT_VSEL137[30]
319	13Fh	AOS_ST RG	AOS_STRG*2	-	✓	INT_SEL0~31	INT_SEL86~91	INT_VSEL137[31]
320	140h	Timer4 _1	TMR41_GCMUH	✓	-	INT_SEL0~31	INT_SEL92~97	INT_VSEL138[0]
321	141h		TMR41_GCMUL	✓	-	INT_SEL0~31	INT_SEL92~97	INT_VSEL138[1]
322	142h		TMR41_GCMVH	✓	-	INT_SEL0~31	INT_SEL92~97	INT_VSEL138[2]
323	143h		TMR41_GCMVL	✓	-	INT_SEL0~31	INT_SEL92~97	INT_VSEL138[3]
324	144h		TMR41_GCMWH	✓	-	INT_SEL0~31	INT_SEL92~97	INT_VSEL138[4]
325	145h		TMR41_GCMWL	✓	-	INT_SEL0~31	INT_SEL92~97	INT_VSEL138[5]
326	146h		TMR41_GOVF	✓	-	INT_SEL0~31	INT_SEL92~97	INT_VSEL138[6]
327	147h		TMR41_GUDF	✓	-	INT_SEL0~31	INT_SEL92~97	INT_VSEL138[7]
328	148h		TMR41_RLOU	✓	-	INT_SEL0~31	INT_SEL92~97	INT_VSEL138[8]
329	149h		TMR41_RLOV	✓	-	INT_SEL0~31	INT_SEL92~97	INT_VSEL138[9]
330	14Ah		TMR41_RLOW	✓	-	INT_SEL0~31	INT_SEL92~97	INT_VSEL138[10]
331	14Bh		-	-	-	INT_SEL0~31	INT_SEL92~97	INT_VSEL138[11]
332	14Ch		-	-	-	INT_SEL0~31	INT_SEL92~97	INT_VSEL138[12]
333	14Dh		-	-	-	INT_SEL0~31	INT_SEL92~97	INT_VSEL138[13]
334	14Eh		-	-	-	INT_SEL0~31	INT_SEL92~97	INT_VSEL138[14]
335	14Fh		-	-	-	INT_SEL0~31	INT_SEL92~97	INT_VSEL138[15]
336	150h	Timer4 _2	TMR42_GCMUH	✓	-	INT_SEL0~31	INT_SEL92~97	INT_VSEL138[16]
337	151h		TMR42_GCMUL	✓	-	INT_SEL0~31	INT_SEL92~97	INT_VSEL138[17]
338	152h		TMR42_GCMVH	✓	-	INT_SEL0~31	INT_SEL92~97	INT_VSEL138[18]
339	153h		TMR42_GCMVL	✓	-	INT_SEL0~31	INT_SEL92~97	INT_VSEL138[19]
340	154h		TMR42_GCMWH	✓	-	INT_SEL0~31	INT_SEL92~97	INT_VSEL138[20]
341	155h		TMR42_GCMWL	✓	-	INT_SEL0~31	INT_SEL92~97	INT_VSEL138[21]
342	156h		TMR42_GOVF	✓	-	INT_SEL0~31	INT_SEL92~97	INT_VSEL138[22]
343	157h		TMR42_GUDF	✓	-	INT_SEL0~31	INT_SEL92~97	INT_VSEL138[23]
344	158h		TMR42_RLOU	✓	-	INT_SEL0~31	INT_SEL92~97	INT_VSEL138[24]
345	159h		TMR42_RLOV	✓	-	INT_SEL0~31	INT_SEL92~97	INT_VSEL138[25]
346	15Ah		TMR42_RLOW	✓	-	INT_SEL0~31	INT_SEL92~97	INT_VSEL138[26]
347	15Bh		-	-	-	INT_SEL0~31	INT_SEL92~97	INT_VSEL138[27]
348	15Ch		-	-	-	INT_SEL0~31	INT_SEL92~97	INT_VSEL138[28]

349	15Dh	-	-	-	-	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[29]
350	15Eh	-	-	-	-	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[30]
351	15Fh	-	-	-	-	INT_SEL0~31	INT_SEL92~97	INT_VSSEL138[31]
352	160h	Timer4 _3	TMR43_GCMUH	✓	-	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[0]
353	161h		TMR43_GCMUL	✓	-	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[1]
354	162h		TMR43_GCMVH	✓	-	INT_SEL0~31	INT_SEL98~103	INT_VSSEL139[2]

serial num ber	Interr upt event reques t numb er	functio nality	Function Name	Wh ether or not it can be sele cte d as an inte rru pt	Can you cho ose for the inte rnal tou ch deriv ation	Interrupt selection register corresponding to NVIC vector *1		
						NVIC Vector 0~31	NVIC Vector 32~127	NVIC Vector 128~143
355	163h		TMR43_GCMVL	✓	-	INT_SEL0~31	INT_SEL98~103	INT_VSEL139[3]
356	164h		TMR43_GCMWH	✓	-	INT_SEL0~31	INT_SEL98~103	INT_VSEL139[4]
357	165h		TMR43_GCMWL	✓	-	INT_SEL0~31	INT_SEL98~103	INT_VSEL139[5]
358	166h		TMR43_GOVF	✓	-	INT_SEL0~31	INT_SEL98~103	INT_VSEL139[6]
359	167h		TMR43_GUDF	✓	-	INT_SEL0~31	INT_SEL98~103	INT_VSEL139[7]
360	168h		TMR43_RLOU	✓	-	INT_SEL0~31	INT_SEL98~103	INT_VSEL139[8]
361	169h		TMR43_RLOV	✓	-	INT_SEL0~31	INT_SEL98~103	INT_VSEL139[9]
362	16Ah		TMR43_RLOW	✓	-	INT_SEL0~31	INT_SEL98~103	INT_VSEL139[10]
363	16Bh		-	-	-	INT_SEL0~31	INT_SEL98~103	INT_VSEL139[11]
364	16Ch		-	-	-	INT_SEL0~31	INT_SEL98~103	INT_VSEL139[12]
365	16Dh		-	-	-	INT_SEL0~31	INT_SEL98~103	INT_VSEL139[13]
366	16Eh		-	-	-	INT_SEL0~31	INT_SEL98~103	INT_VSEL139[14]
367	16Fh		-	-	-	INT_SEL0~31	INT_SEL98~103	INT_VSEL139[15]
368	170h	Timer4 _1 EVT	TMR41_SCM0	-	✓	INT_SEL0~31	INT_SEL98~103	INT_VSEL139[16]
369	171h		TMR41_SCM1	-	✓	INT_SEL0~31	INT_SEL98~103	INT_VSEL139[17]
370	172h		TMR41_SCM2	-	✓	INT_SEL0~31	INT_SEL98~103	INT_VSEL139[18]
371	173h		TMR41_SCM3	-	✓	INT_SEL0~31	INT_SEL98~103	INT_VSEL139[19]
372	174h		TMR41_SCM4	-	✓	INT_SEL0~31	INT_SEL98~103	INT_VSEL139[20]
373	175h		TMR41_SCM5	-	✓	INT_SEL0~31	INT_SEL98~103	INT_VSEL139[21]
374	176h	Timer4 _2 EVT	TMR42_SCM0	-	✓	INT_SEL0~31	INT_SEL98~103	INT_VSEL139[22]
375	177h		TMR42_SCM1	-	✓	INT_SEL0~31	INT_SEL98~103	INT_VSEL139[23]
376	178h		TMR42_SCM2	-	✓	INT_SEL0~31	INT_SEL98~103	INT_VSEL139[24]
377	179h		TMR42_SCM3	-	✓	INT_SEL0~31	INT_SEL98~103	INT_VSEL139[25]
378	17Ah		TMR42_SCM4	-	✓	INT_SEL0~31	INT_SEL98~103	INT_VSEL139[26]
379	17Bh		TMR42_SCM5	-	✓	INT_SEL0~31	INT_SEL98~103	INT_VSEL139[27]
380	17Ch		-	-	-	INT_SEL0~31	INT_SEL98~103	INT_VSEL139[28]
381	17Dh		-	-	-	INT_SEL0~31	INT_SEL98~103	INT_VSEL139[29]
382	17Eh	-	-	-	-	INT_SEL0~31	INT_SEL98~103	INT_VSEL139[30]
383	17Fh	-	-	-	-	INT_SEL0~31	INT_SEL98~103	INT_VSEL139[31]
384	180h	Timer4 _3 EVT	TMR43_SCM0	-	✓	INT_SEL0~31	INT_SEL104~109	INT_VSEL140[0]
385	181h		TMR43_SCM1	-	✓	INT_SEL0~31	INT_SEL104~109	INT_VSEL140[1]
386	182h		TMR43_SCM2	-	✓	INT_SEL0~31	INT_SEL104~109	INT_VSEL140[2]

387	183h		TMR43_SCM3	-	✓	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[3]
388	184h		TMR43_SCM4	-	✓	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[4]
389	185h		TMR43_SCM5	-	✓	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[5]
390	186h	EMB	EMB_GR0	✓	-	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[6]
391	187h		EMB_GR1	✓	-	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[7]
392	188h		EMB_GR2	✓	-	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[8]
393	189h		EMB_GR3	✓	-	INT_SEL0~31	INT_SEL104~109	INT_VSSEL140[9]

serial number	Interrrupt event request number	functionality	Function Name	Whether or not it can be selected as an interrupt	Can you choose for the internal touch derivation	Interrupt selection register corresponding to NVIC vector *1		
						NVIC Vector 0~31	NVIC Vector 32~127	NVIC Vector 128~143
394	18Ah	EVENT port	EVENT_PORT1	✓	✓	INT_SEL0~31	INT_SEL104~109	-
395	18Bh		EVENT_PORT2	✓	✓	INT_SEL0~31	INT_SEL104~109	-
396	18Ch		EVENT_PORT3	✓	✓	INT_SEL0~31	INT_SEL104~109	-
397	18Dh		EVENT_PORT4	✓	✓	INT_SEL0~31	INT_SEL104~109	-
398	18Eh		-	-	-	INT_SEL0~31	INT_SEL104~109	INT_VSEL140[14]
399	18Fh		-	-	-	INT_SEL0~31	INT_SEL104~109	INT_VSEL140[15]
400	190h	I2S1	I2S1_TXIRQOUT	✓	✓	INT_SEL0~31	INT_SEL104~109	INT_VSEL140[16]
401	191h		I2S1_RXIRQOUT	✓	✓	INT_SEL0~31	INT_SEL104~109	INT_VSEL140[17]
402	192h		I2S1_ERRIRQOUT	✓	-	INT_SEL0~31	INT_SEL104~109	INT_VSEL140[18]
403	193h	I2S2	I2S2_TXIRQOUT	✓	✓	INT_SEL0~31	INT_SEL104~109	INT_VSEL140[19]
404	194h		I2S2_RXIRQOUT	✓	✓	INT_SEL0~31	INT_SEL104~109	INT_VSEL140[20]
405	195h		I2S2_ERRIRQOUT	✓	-	INT_SEL0~31	INT_SEL104~109	INT_VSEL140[21]
406	196h	I2S3	I2S3_TXIRQOUT	✓	✓	INT_SEL0~31	INT_SEL104~109	INT_VSEL140[22]
407	197h		I2S3_RXIRQOUT	✓	✓	INT_SEL0~31	INT_SEL104~109	INT_VSEL140[23]
408	198h		I2S3_ERRIRQOUT	✓	-	INT_SEL0~31	INT_SEL104~109	INT_VSEL140[24]
409	199h	I2S4	I2S4_TXIRQOUT	✓	✓	INT_SEL0~31	INT_SEL104~109	INT_VSEL140[25]
410	19Ah		I2S4_RXIRQOUT	✓	✓	INT_SEL0~31	INT_SEL104~109	INT_VSEL140[26]
411	19Bh		I2S4_ERRIRQOUT	✓	-	INT_SEL0~31	INT_SEL104~109	INT_VSEL140[27]
412	19Ch	-	-	-	-	INT_SEL0~31	INT_SEL104~109	INT_VSEL140[28]
413	19Dh	-	-	-	-	INT_SEL0~31	INT_SEL104~109	INT_VSEL140[29]
414	19Eh	-	-	-	-	INT_SEL0~31	INT_SEL104~109	INT_VSEL140[30]
415	19Fh	-	-	-	-	INT_SEL0~31	INT_SEL104~109	INT_VSEL140[31]

416	1A0h	ACMP	ACMP1	✓	✓	INT_SEL0~31	INT_SEL110~115	-
417	1A1h		ACMP2	✓	✓	INT_SEL0~31	INT_SEL110~115	-
418	1A2h		ACMP3	✓	✓	INT_SEL0~31	INT_SEL110~115	-
419	1A3h	I2C1	-	-	-	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[3]
420	1A4h		I2C1_RXI	✓	✓	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[4]
421	1A5h		I2C1_TXI	✓	✓	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[5]
422	1A6h		I2C1_TEI	✓	✓	INT_SEL0~31	INT_SEL110~115	INT_VSSEL141[6]

serial num ber	Interr upt event reques t numb er	functio nality	Function Name	Wh ether or not it can be sele cte d as an inte rru pt	Can you cho ose for the inte rnal tou ch deriv ation	Interrupt selection register corresponding to NVIC vector *1		
						NVIC Vector 0~31	NVIC Vector 32~127	NVIC Vector 128~143
423	1A7h		I2C1_EE1	✓	✓	INT_SEL0~31	INT_SEL110~115	INT_VSEL141[7]
424	1A8h	I2C2	I2C2_RXI	✓	✓	INT_SEL0~31	INT_SEL110~115	INT_VSEL141[8]
425	1A9h		I2C2_TXI	✓	✓	INT_SEL0~31	INT_SEL110~115	INT_VSEL141[9]
426	1AAh		I2C2_TEI	✓	✓	INT_SEL0~31	INT_SEL110~115	INT_VSEL141[10]
427	1ABh		I2C2_EE1	✓	✓	INT_SEL0~31	INT_SEL110~115	INT_VSEL141[11]
428	1ACh	I2C3	I2C3_RXI	✓	✓	INT_SEL0~31	INT_SEL110~115	INT_VSEL141[12]
429	1ADh		I2C3_TXI	✓	✓	INT_SEL0~31	INT_SEL110~115	INT_VSEL141[13]
430	1AEh		I2C3_TEI	✓	✓	INT_SEL0~31	INT_SEL110~115	INT_VSEL141[14]
431	1AFh		I2C3_EE1	✓	✓	INT_SEL0~31	INT_SEL110~115	INT_VSEL141[15]
432	1B0h	USART1	USART1_WUPI	✓	-	INT_SEL0~31	INT_SEL110~115	-
433	1B1h	PVD	PVD_PVD1	✓	✓	INT_SEL0~31	INT_SEL110~115	INT_VSEL141[17]
434	1B2h		PVD_PVD2	✓	✓	INT_SEL0~31	INT_SEL110~115	INT_VSEL141[18]
435	1B3h	OTS	OTS	✓	✓	INT_SEL0~31	INT_SEL110~115	-
436	1B4h	FCM	FCMFERRI	✓	-	INT_SEL0~31	INT_SEL110~115	INT_VSEL141[20]
437	1B5h		FCMMENDI	✓	-	INT_SEL0~31	INT_SEL110~115	INT_VSEL141[21]
438	1B6h		FCMCOVFI	✓	-	INT_SEL0~31	INT_SEL110~115	INT_VSEL141[22]
439	1B7h	WDT	WDT_REFUDF	✓	✓	INT_SEL0~31	INT_SEL110~115	INT_VSEL141[23]
440	1B8h	-	-	-	-	INT_SEL0~31	INT_SEL110~115	INT_VSEL141[24]
441	1B9h	-	-	-	-	INT_SEL0~31	INT_SEL110~115	INT_VSEL141[25]
442	1BAh	-	-	-	-	INT_SEL0~31	INT_SEL110~115	INT_VSEL141[26]
443	1BBh	-	-	-	-	INT_SEL0~31	INT_SEL110~115	INT_VSEL141[27]
444	1BCh	-	-	-	-	INT_SEL0~31	INT_SEL110~115	INT_VSEL141[28]
445	1BDh	-	-	-	-	INT_SEL0~31	INT_SEL110~115	INT_VSEL141[29]
446	1BEh	-	-	-	-	INT_SEL0~31	INT_SEL110~115	INT_VSEL141[30]
447	1BFh	-	-	-	-	INT_SEL0~31	INT_SEL110~115	INT_VSEL141[31]
448	1C0h	ADC1	ADC1_EOCA	✓	✓	INT_SEL0~31	INT_SEL116~121	INT_VSEL142[0]
449	1C1h		ADC1_EOCB	✓	✓	INT_SEL0~31	INT_SEL116~121	INT_VSEL142[1]
450	1C2h		ADC1_CHCMP	✓	✓	INT_SEL0~31	INT_SEL116~121	INT_VSEL142[2]
451	1C3h		ADC1_SEQCMP	✓	✓	INT_SEL0~31	INT_SEL116~121	INT_VSEL142[3]
452	1C4h	ADC2	ADC2_EOCA	✓	✓	INT_SEL0~31	INT_SEL116~121	INT_VSEL142[4]
453	1C5h		ADC2_EOCB	✓	✓	INT_SEL0~31	INT_SEL116~121	INT_VSEL142[5]
454	1C6h		ADC2_CHCMP	✓	✓	INT_SEL0~31	INT_SEL116~121	INT_VSEL142[6]
455	1C7h		ADC2_SEQCMP	✓	✓	INT_SEL0~31	INT_SEL116~121	INT_VSEL142[7]
456	1C8h	TRNG	TRNG_END	✓	✓	INT_SEL0~31	INT_SEL116~121	-
457	1C9h	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSEL142[9]

458	1CAh	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[10]
459	1CBh	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[11]
460	1CCh	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[12]
461	1CDh	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[13]
462	1CEh	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[14]
463	1CFh	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[15]
464	1D0h	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSSEL142[16]

serial number	Interrrupt event request number	functionality	Function Name	Whether or not it can be selected as an interrupt	Can you choose for the internal touch derivation	Interrupt selection register corresponding to NVIC vector *1		
						NVIC Vector 0~31	NVIC Vector 32~127	NVIC Vector 128~143
465	1D1h	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSEL142[17]
466	1D2h	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSEL142[18]
467	1D3h	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSEL142[19]
468	1D4h	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSEL142[20]
469	1D5h	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSEL142[21]
470	1D6h	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSEL142[22]
471	1D7h	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSEL142[23]
472	1D8h	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSEL142[24]
473	1D9h	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSEL142[25]
474	1DAh	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSEL142[26]
475	1DBh	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSEL142[27]
476	1DCh	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSEL142[28]
477	1DDh	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSEL142[29]
478	1DEh	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSEL142[30]
479	1DFh	-	-	-	-	INT_SEL0~31	INT_SEL116~121	INT_VSEL142[31]
480	1E0h	SDIOC1	SDIOC1_DMAR	-	✓	INT_SEL0~31	INT_SEL122~127	INT_VSEL143[0]
481	1E1h		SDIOC1_DMAW	-	✓	INT_SEL0~31	INT_SEL122~127	INT_VSEL143[1]
482	1E2h		SDIOC1_SD	✓	-	INT_SEL0~31	INT_SEL122~127	INT_VSEL143[2]
483	1E3h	SDIOC2	SDIOC2_DMAR	-	✓	INT_SEL0~31	INT_SEL122~127	INT_VSEL143[3]
484	1E4h		SDIOC2_DMAW	-	✓	INT_SEL0~31	INT_SEL122~127	INT_VSEL143[4]
485	1E5h		SDIOC2_SD	✓	-	INT_SEL0~31	INT_SEL122~127	INT_VSEL143[5]
486	1E6h	CAN	CAN_INT	✓	-	INT_SEL0~31	INT_SEL122~127	INT_VSEL143[6]
487	1E7h	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSEL143[7]
488	1E8h	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSEL143[8]
489	1E9h	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSEL143[9]
490	1EAh	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSEL143[10]
491	1EBh	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSEL143[11]
492	1ECb	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSEL143[12]
493	1EDh	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSEL143[13]
494	1EEh	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSEL143[14]
495	1EFh	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSEL143[15]
496	1F0h	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSEL143[16]
497	1F1h	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSEL143[17]
498	1F2h	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSEL143[18]
499	1F3h	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSEL143[19]

500	1F4h	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[20]
501	1F5h	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[21]
502	1F6h	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[22]
503	1F7h	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[23]
504	1F8h	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[24]
505	1F9h	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[25]
506	1FAh	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSSEL143[26]

serial num ber	Interr upt event reques t numb er	functio nality	Function Name	Wh ether or not it can be sele cte d as an inte rru pt	Can you cho ose for the inte rnal tou ch deriv ation	Interrupt selection register corresponding to NVIC vector *1		
						NVIC Vector 0~31	NVIC Vector 32~127	NVIC Vector 128~143
507	1FBh	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSEL143[27]
508	1FCCh	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSEL143[28]
509	1FDh	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSEL143[29]
510	1FEh	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSEL143[30]
511	1FFh	-	-	-	-	INT_SEL0~31	INT_SEL122~127	INT_VSEL143[31]

*1: Interrupt event request sequence number selected by the Interrupt Select Register If not configured, this register or bit setting is invalid.

*2: AOS_STRG is generated by the SFTG bit of the Software Set Peripheral Trigger Event Register (INTSFTTRG).

10.4 Functional Description

10.4.1 unmaskable interrupt

The non-maskable interrupt sources are listed below:

- NMI Pin Interrupt
- Detecting master oscillator stop interrupts
- WDT underflow/refresh interrupt
- SWDT underflow/refresh interrupt
- Low voltage detection PVD1 interrupt
- Low voltage detection PVD2 interrupt
- SRAM Parity Error Interrupt
- SRAM ECC Checksum Error Interrupt
- MPU Bus Error Interrupt

Non-maskable interrupts have the highest priority. Since a non-maskable interrupt can select multiple interrupt event requests, the status of each interrupt event request can be determined by querying the flag register (INT_NMIFR). Make sure that all the flag bits are "0" before exiting the non-maskable interrupt processing. The default state of non-maskable interrupt is disabled, which can be set by ICG or control register.

When using the register setting, follow the procedure below:

1. When you need to use the NMI pin, first clear the INT_NMICR.NFEN bit to disable the digital filter; set the NMITRG bit of the INT_NMICR register to select the NMI trigger edge; set the SMPCLK bit to select the sampling clock of the digital filter; set the NFEN bit to enable the digital filter.
2. When using other non-maskable interrupt event requests, configure the appropriate function.
3. Write 1 to each INT_NMICFR register bit to clear the INT_NMIFR flag register bit to prevent malfunction.
4. Enable non-maskable interrupt events by setting the INT_NMIENR selection register.

Attention:

- Once the INT_NMIENR bit is set to 1, it cannot be changed unless it is reset with RESET. ICG setting is only for external NMI pin, enable ICG setting by configuring ICG register ICG1.NMIIICGENA bit. NMITRG to select the NMI trigger edge; ICG1.SMPCLK to select the digital filter sample clock; ICG1.NFEN to enable the digital filter; and ICG1.NMIENR to enable the NMI pin interrupt; the register setting is invalidated after ICG is set; refer to the section of Initialization Configuration (ICG) for the description of ICG1 registers. (For the register description of ICG1, please refer to the "Initialization Configuration (ICG)" section.

10.4.2 External pin interrupt event request

If you need to use an external pin interrupt event request, set up the following procedure:

1. Clear the INT_EIRQCRm.EFEN bit ($m=0\sim15$) to disable the digital filter.
2. Set the IRQTRG[1:0] bits of INT_EIRQCRm to select the trigger edge or level; set the SMPCLK[1:0] bits to select the digital filter sampling clock; set the EFEN bits to enable the digital filter.

10.4.3 Interrupt Source Selection

This interrupt controller uses a total of 144 interrupt vectors and provides three interrupt event request selection methods to meet various interrupt configuration needs through flexible combinations.

The first way

There are 32 interrupt vectors in total, and all interrupt event requests select 1 as the interrupt source, which is selected through the interrupt/event selection registers INT_SEL0~31, and enabled through the INT_IER registers, and the corresponding interrupt vectors of NVIC are 0~31.

The second way

There are 96 interrupt vectors in total, and 32 select 1 as interrupt source, which is selected by interrupt selection register INT_SEL32~127, and the corresponding interrupt vectors are 32~127.

The third way

There are 16 interrupt vectors in total, 32 peripheral interrupt event requests share 1 interrupt vector, each peripheral can apply for interrupt, differentiated by the peripheral flag bit, and the interrupt event request is enabled by the interrupt enable register INT_VSEL128~143, corresponding to the interrupt vectors of the NVIC are 128~143. Select the interrupt event request that has the peripheral flag bit, and refer to the NVIC vector 128~143 columns in the "Interrupt Event Request Serial Number". To select the interrupt event request with peripheral flag bit, please refer to the NVIC vector 128~143 columns in [InterruptEvent Request Serial Number].

Refer to the [Interrupt Event Request Sequence Numbers] section for specific interrupt vector assignments.

10.4.4 software interruption

The software interrupt function can be licensed by directly writing the software interrupt control register INT_SWIER to generate an interrupt event request controlled by the INT_IER interrupt enable bit. A total of 32 software interrupt event

requests are configured, corresponding to interrupt vectors 0~31, please refer to [Interrupt Vector Table] section for details.

10.4.5 Interrupt/Event Selection

The interrupts selected by the interrupt selection registers INT_SEL0~31 share the interrupt vectors 0~31 of NVIC with the software interrupts, which are controlled and licensed by the INT_IER interrupt enable register; at the same time, these interrupt event requests can be selected as event inputs to wake up the kernel (WFE) via the selected by the event enable register INT_EVTER, as shown in the following block diagram.

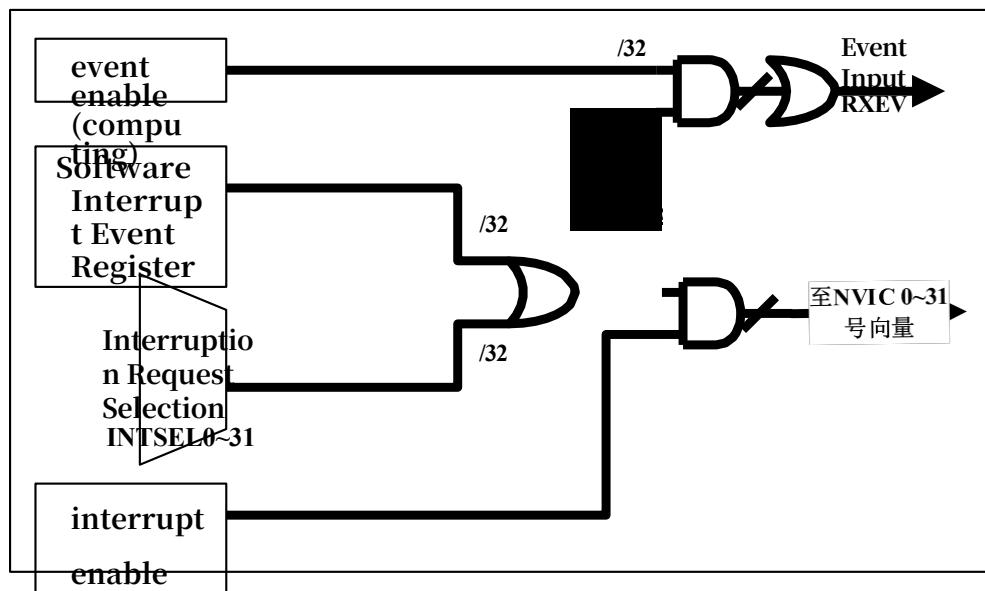


Figure 10-2 Interrupt Event Selection

10.4.6 WFE Wake Up

Event

Management

The MCU is capable of handling events to wake up the core (WFE). Wake-up events can be generated either by interrupt input from the NVIC or by event input. The setup flow is as follows.

- Wake up the WFE from the interrupt input of the NVIC. enable an interrupt in the control register of the peripheral, set the INT_SEL n and INT_IER registers according to the selected interrupt vector, but do not enable it in the NVIC, and set INT_EVTER to disable, and enable the SEVONPEND bit in the SCR of the Cortex™-M4F system control register. When the MCU recovers from WFE, the interrupt flag bit of the corresponding peripheral and the NVIC interrupt flag register need to be cleared.

The WFE Stop Mode Entry and Wake-Up procedure is as follows:

- 1) Sets the stop mode register;
- 2) Sets the stop mode wake-up register INT_WUPEN;
- 3) Sets the pin EIRQ input with the EIRQ control register INT_EIRQCR n ;
- 4) Select INT_SEL n to select the corresponding EIRQ interrupt event request serial

- 5) INT_IER register for enabling the corresponding interrupt event request, INT_EVTER non-enabling;
- 6) Set the SEVONPEND bit in the SCR to 1;
- 7) Perform the following to ensure that the system enters the stop mode:
SEV(). Setting the internal event register
WFE(). Clears the event register

WFE(). The system enters stop mode

- 8) Waiting for the selected interrupt event request to occur, the system will wake up from the stop mode, but not enter the interrupt handling subroutine.
- Wake up the WFE from the event input of the NVIC. configure an interrupt event request as an event input to be enabled via the event enable register INT_EVTER. When the CPU recovers from the WFE, the interrupt flag bit of the corresponding peripheral needs to be cleared.

The WFE stop mode wake-up process is as follows.

- 1) Sets the stop mode register;
- 2) Sets the stop mode wake-up register INT_WUPEN;
- 3) Sets the pin EIRQ input with the EIRQ control register INT_EIRQCRn;
- 4) Select INT_SELn to select the corresponding EIRQ interrupt event request serial number;
- 5) INT_IER register is non-enabled and INT_EVTER enables the corresponding interrupt event request;
- 6) Perform the following to ensure that the system enters the stop mode:

SEV(). Setting the internal event register

WFE(). Clears the event register

WFE(). The system enters stop mode

- 7) Waiting for the selected interrupt event request to occur, the system will wake up from the stop mode, but not enter the interrupt handling subroutine.

10.4.7 digital filter

For NMI and EIRQx (x=0~15) pin event inputs, a digital filter can be selected for noise filtering. The sampling clock of the filter is PCLK3, and input signals with less than 3 filtering cycles will be filtered out. The specific operation timing diagram is as follows:

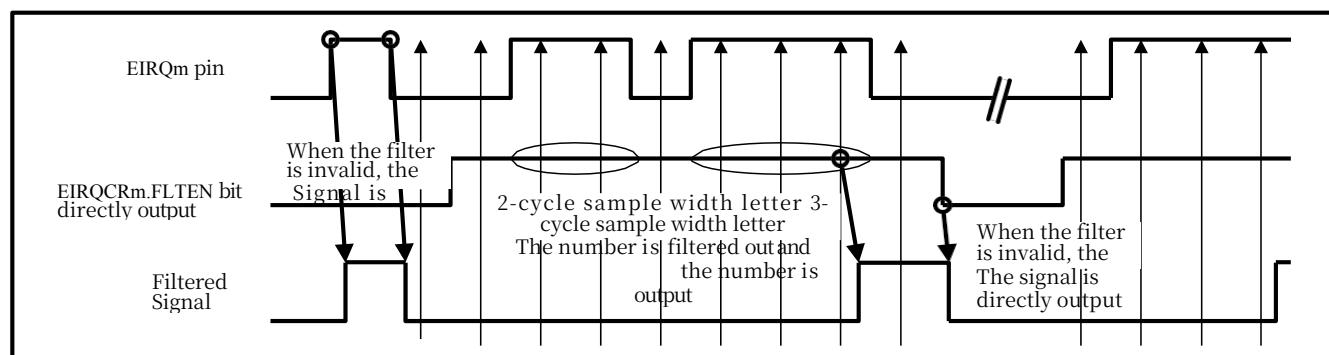


Figure 10-3 Digital Filter Operating Schematic

Set INT_NMICR.NFEN and INT_EIRQCRm.EFEN to disable the use of digital filters before entering stop mode. Enable the digital filter again after returning from the stop mode. The setting procedure is as follows:

-
- 1) Sets the system stop mode register
 - 2) Configuring Stop Mode Wake-Up Interrupts
 - 3) Stop Digital Filter
 - 4) Execute WFI, system enters stop mode

10.4.8 Low Power Mode Return

10.4.8.1 Sleep mode return

When the event interrupt source is selected as the hibernation mode return condition, the following settings are required:

- Select event as CPU interrupt source
- Enabling Control Registers in the NVIC
- If you want to use non-maskable interrupts, you need to set the INT_NMIENR enable register.

10.4.8.2 Stop Mode Return

A non-maskable event interrupt source or a maskable event source selected in the INT_WUPEN register can be selected as the return condition for stop mode.

The following settings are required to return from the stop mode:

- Select event interrupt source as return condition for stop mode
 - A. For non-maskable interrupts, NMI pin interrupts are set through the INT_NMIENR enable register, and SWDT, PVD1, and PVD2 can wake up in stop mode.
 - B. For maskable interrupts, this is set via the INT_WUPEN enable register.
- Select event as CPU interrupt source
- Enabling Control Registers in the NVIC

For unchecked EIRQ pins, they will not be detected because the clock is turned off.

10.4.8.3 Power-down mode return

Return from power-down mode can be returned by the conditions indicated in the Power Control (PWC) section of Chapter 5, RES# pin reset, power-up reset, and low voltage detect 0 conditions. The CPU enters reset interrupt processing after the return. Refer to the [Power Control (PWC)] section for detailed description.

10.4.8.4 Non-maskable interrupt and WFI instructions

Before the WFI instruction is executed, make sure that all status bits of the non-maskable interrupt flag register INT_NMIFR are 0.

10.4.9 Internally triggered events

Peripheral peripherals such as ADC, Timer, DMA, PORT, DCU, etc. can be triggered to start working by writing to the Peripheral Trigger Event Register in addition to configuring the module's own registers to start working. To use the internal trigger event, you need to clear the PWC_FCG0.AOS bit to enable the peripheral circuit trigger function. Please refer to each

module's chapter for the detailed setup procedure.

10.5 Register Description

The following table shows

the INTC register list. INTC

base

address:0x4005_1000

register name	notation	offset address	bit width	reset value
NMI pin non-maskable interrupt control registers	INT_NMICR	0x0000	32	0x0000_0000
Non-maskable Interrupt Enable Register	INT_NMIENR	0x0004	32	0x0000_0000
Non-maskable interrupt flag register	INT_NMIFR	0x0008	32	0x0000_0000
Non-maskable interrupt flag clear register	INT_NMICFR	0x000C	32	0x0000_0000
External Pin Interrupt Control Register 0	INT_EIRQCR0	0x0010	32	0x0000_0000
External Pin Interrupt Control Register 1	INT_EIRQCR1	0x0014	32	0x0000_0000
External Pin Interrupt Control Register 2	INT_EIRQCR2	0x0018	32	0x0000_0000
External Pin Interrupt Control Register 3	INT_EIRQCR3	0x001C	32	0x0000_0000
External Pin Interrupt Control Register 4	INT_EIRQCR4	0x0020	32	0x0000_0000
External Pin Interrupt Control Register 5	INT_EIRQCR5	0x0024	32	0x0000_0000
External Pin Interrupt Control Register 6	INT_EIRQCR6	0x0028	32	0x0000_0000
External Pin Interrupt Control Register 7	INT_EIRQCR7	0x002C	32	0x0000_0000
External Pin Interrupt Control Register 8	INT_EIRQCR8	0x0030	32	0x0000_0000
External Pin Interrupt Control Register 9	INT_EIRQCR9	0x0034	32	0x0000_0000
External Pin Interrupt Control Register 10	INT_EIRQCR10	0x0038	32	0x0000_0000
External Pin Interrupt Control Register 11	INT_EIRQCR11	0x003C	32	0x0000_0000
External Pin Interrupt Control Register 12	INT_EIRQCR12	0x0040	32	0x0000_0000
External Pin Interrupt Control Register 13	INT_EIRQCR13	0x0044	32	0x0000_0000
External Pin Interrupt Control Register 14	INT_EIRQCR14	0x0048	32	0x0000_0000
External Pin Interrupt Control Register 15	INT_EIRQCR15	0x004C	32	0x0000_0000
Stop Mode Wake-Up Event Enable Register	INT_WUPEN	0x0050	32	0x0000_0000
External Pin Interrupt Flag Register	INT_EIFR	0x0054	32	0x0000_0000
External Pin Interrupt Flag Clear Register	INT{EIFCR}	0x0058	32	0x0000_0000
Interrupt/Event Selection Register 0	INT_SEL0	0x005C	32	0x0000_01FF

Interrupt/Event Selection Register 1	INT_SEL1	0x0060	32	0x0000_01FF
Interrupt/Event Selection Register 2	INT_SEL2	0x0064	32	0x0000_01FF
Interrupt/Event Selection Register 3	INT_SEL3	0x0068	32	0x0000_01FF
Interrupt/Event Selection Register 4	INT_SEL4	0x006C	32	0x0000_01FF
Interrupt/Event Selection Register 5	INT_SEL5	0x0070	32	0x0000_01FF
Interrupt/Event Selection Register 6	INT_SEL6	0x0074	32	0x0000_01FF
Interrupt/Event Selection Register 7	INT_SEL7	0x0078	32	0x0000_01FF
Interrupt/Event Selection Register 8	INT_SEL8	0x007C	32	0x0000_01FF
Interrupt/Event Selection Register 9	INT_SEL9	0x0080	32	0x0000_01FF

register name	notation	offset address	bit width	reset value
Interrupt/Event Selection Register 10	INT_SEL10	0x0084	32	0x0000_01FF
Interrupt/Event Selection Register 11	INT_SEL11	0x0088	32	0x0000_01FF
Interrupt/Event Selection Register 12	INT_SEL12	0x008C	32	0x0000_01FF
Interrupt/Event Selection Register 13	INT_SEL13	0x0090	32	0x0000_01FF
Interrupt/Event Selection Register 14	INT_SEL14	0x0094	32	0x0000_01FF
Interrupt/Event Selection Register 15	INT_SEL15	0x0098	32	0x0000_01FF
Interrupt/Event Selection Register 16	INT_SEL16	0x009C	32	0x0000_01FF
Interrupt/Event Selection Register 17	INT_SEL17	0x00A0	32	0x0000_01FF
Interrupt/Event Select Register 18	INT_SEL18	0x00A4	32	0x0000_01FF
Interrupt/Event Selection Register 19	INT_SEL19	0x00A8	32	0x0000_01FF
Interrupt/Event Selection Register 20	INT_SEL20	0x00AC	32	0x0000_01FF
Interrupt/Event Selection Register 21	INT_SEL21	0x00B0	32	0x0000_01FF
Interrupt/Event Selection Register 22	INT_SEL22	0x00B4	32	0x0000_01FF
Interrupt/Event Selection Register 23	INT_SEL23	0x00B8	32	0x0000_01FF
Interrupt/Event Selection Register 24	INT_SEL24	0x00BC	32	0x0000_01FF
Interrupt/Event Selection Register 25	INT_SEL25	0x00C0	32	0x0000_01FF
Interrupt/Event Selection Register 26	INT_SEL26	0x00C4	32	0x0000_01FF
Interrupt/Event Selection Register 27	INT_SEL27	0x00C8	32	0x0000_01FF
Interrupt/Event Select Register 28	INT_SEL28	0x00CC	32	0x0000_01FF
Interrupt/Event Selection Register 29	INT_SEL29	0x00D0	32	0x0000_01FF
Interrupt/Event Selection Register 30	INT_SEL30	0x00D4	32	0x0000_01FF
Interrupt/Event Selection Register 31	INT_SEL31	0x00D8	32	0x0000_01FF
Interrupt Select Register 32	INT_SEL32	0x00DC	32	0x0000_01FF
Interrupt Select Register 33	INT_SEL33	0x00E0	32	0x0000_01FF
Interrupt Select Register 34	INT_SEL34	0x00E4	32	0x0000_01FF
Interrupt Select Register 35	INT_SEL35	0x00E8	32	0x0000_01FF
Interrupt Select Register 36	INT_SEL36	0x00EC	32	0x0000_01FF
Interrupt Select Register 37	INT_SEL37	0x00F0	32	0x0000_01FF
Interrupt Select Register 38	INT_SEL38	0x00F4	32	0x0000_01FF
Interrupt Select Register 39	INT_SEL39	0x00F8	32	0x0000_01FF
Interrupt Select Register 40	INT_SEL40	0x00FC	32	0x0000_01FF
Interrupt Select Register 41	INT_SEL41	0x0100	32	0x0000_01FF
Interrupt Select Register 42	INT_SEL42	0x0104	32	0x0000_01FF
Interrupt Select Register 43	INT_SEL43	0x0108	32	0x0000_01FF
Interrupt Select Register 44	INT_SEL44	0x010C	32	0x0000_01FF
Interrupt Select Register 45	INT_SEL45	0x0110	32	0x0000_01FF
Interrupt Select Register 46	INT_SEL46	0x0114	32	0x0000_01FF
Interrupt Select Register 47	INT_SEL47	0x0118	32	0x0000_01FF

register name	notation	offset address	bit width	reset value
Interrupt Select Register 48	INT_SEL48	0x011C	32	0x0000_01FF
Interrupt Select Register 49	INT_SEL49	0x0120	32	0x0000_01FF
Interrupt Select Register 50	INT_SEL50	0x0124	32	0x0000_01FF
Interrupt Select Register 51	INT_SEL51	0x0128	32	0x0000_01FF
Interrupt Select Register 52	INT_SEL52	0x012C	32	0x0000_01FF
Interrupt Select Register 53	INT_SEL53	0x0130	32	0x0000_01FF
Interrupt Select Register 54	INT_SEL54	0x0134	32	0x0000_01FF
Interrupt Select Register 55	INT_SEL55	0x0138	32	0x0000_01FF
Interrupt Select Register 56	INT_SEL56	0x013C	32	0x0000_01FF
Interrupt Select Register 57	INT_SEL57	0x0140	32	0x0000_01FF
Interrupt Select Register 58	INT_SEL58	0x0144	32	0x0000_01FF
Interrupt Select Register 59	INT_SEL59	0x0148	32	0x0000_01FF
Interrupt Select Register 60	INT_SEL60	0x014C	32	0x0000_01FF
Interrupt Select Register 61	INT_SEL61	0x0150	32	0x0000_01FF
Interrupt Select Register 62	INT_SEL62	0x0154	32	0x0000_01FF
Interrupt Select Register 63	INT_SEL63	0x0158	32	0x0000_01FF
Interrupt Select Register 64	INT_SEL64	0x015C	32	0x0000_01FF
Interrupt Select Register 65	INT_SEL65	0x0160	32	0x0000_01FF
Interrupt Select Register 66	INT_SEL66	0x0164	32	0x0000_01FF
Interrupt Select Register 67	INT_SEL67	0x0168	32	0x0000_01FF
Interrupt Select Register 68	INT_SEL68	0x016C	32	0x0000_01FF
Interrupt Select Register 69	INT_SEL69	0x0170	32	0x0000_01FF
Interrupt Select Register 70	INT_SEL70	0x0174	32	0x0000_01FF
Interrupt Select Register 71	INT_SEL71	0x0178	32	0x0000_01FF
Interrupt Select Register 72	INT_SEL72	0x017C	32	0x0000_01FF
Interrupt Select Register 73	INT_SEL73	0x0180	32	0x0000_01FF
Interrupt Select Register 74	INT_SEL74	0x0184	32	0x0000_01FF
Interrupt Select Register 75	INT_SEL75	0x0188	32	0x0000_01FF
Interrupt Select Register 76	INT_SEL76	0x018C	32	0x0000_01FF
Interrupt Select Register 77	INT_SEL77	0x0190	32	0x0000_01FF
Interrupt Select Register 78	INT_SEL78	0x0194	32	0x0000_01FF
Interrupt Select Register 79	INT_SEL79	0x0198	32	0x0000_01FF
Interrupt Select Register 80	INT_SEL80	0x019C	32	0x0000_01FF
Interrupt Select Register 81	INT_SEL81	0x01A0	32	0x0000_01FF
Interrupt Select Register 82	INT_SEL82	0x01A4	32	0x0000_01FF
Interrupt Select Register 83	INT_SEL83	0x01A8	32	0x0000_01FF
Interrupt Select Register 84	INT_SEL84	0x01AC	32	0x0000_01FF
Interrupt Select Register 85	INT_SEL85	0x01B0	32	0x0000_01FF

register name	notation	offset address	bit width	reset value
Interrupt Select Register 86	INT_SEL86	0x01B4	32	0x0000_01FF
Interrupt Select Register 87	INT_SEL87	0x01B8	32	0x0000_01FF
Interrupt Select Register 88	INT_SEL88	0x01BC	32	0x0000_01FF
Interrupt Select Register 89	INT_SEL89	0x01C0	32	0x0000_01FF
Interrupt Select Register 90	INT_SEL90	0x01C4	32	0x0000_01FF
Interrupt Select Register 91	INT_SEL91	0x01C8	32	0x0000_01FF
Interrupt Select Register 92	INT_SEL92	0x01CC	32	0x0000_01FF
Interrupt Select Register 93	INT_SEL93	0x01D0	32	0x0000_01FF
Interrupt Select Register 94	INT_SEL94	0x01D4	32	0x0000_01FF
Interrupt Select Register 95	INT_SEL95	0x01D8	32	0x0000_01FF
Interrupt Select Register 96	INT_SEL96	0x01DC	32	0x0000_01FF
Interrupt Select Register 97	INT_SEL97	0x01E0	32	0x0000_01FF
Interrupt Select Register 98	INT_SEL98	0x01E4	32	0x0000_01FF
Interrupt Select Register 99	INT_SEL99	0x01E8	32	0x0000_01FF
Interrupt Select Register 100	INT_SEL100	0x01EC	32	0x0000_01FF
Interrupt Select Register 101	INT_SEL101	0x01F0	32	0x0000_01FF
Interrupt Select Register 102	INT_SEL102	0x01F4	32	0x0000_01FF
Interrupt Select Register 103	INT_SEL103	0x01F8	32	0x0000_01FF
Interrupt Select Register 104	INT_SEL104	0x01FC	32	0x0000_01FF
Interrupt Select Register 105	INT_SEL105	0x0200	32	0x0000_01FF
Interrupt Select Register 106	INT_SEL106	0x0204	32	0x0000_01FF
Interrupt Select Register 107	INT_SEL107	0x0208	32	0x0000_01FF
Interrupt Select Register 108	INT_SEL108	0x020C	32	0x0000_01FF
Interrupt Select Register 109	INT_SEL109	0x0210	32	0x0000_01FF
Interrupt Select Register 110	INT_SEL110	0x0214	32	0x0000_01FF
Interrupt Select Register 111	INT_SEL111	0x0218	32	0x0000_01FF
Interrupt Select Register 112	INT_SEL112	0x021C	32	0x0000_01FF
Interrupt Select Register 113	INT_SEL113	0x0220	32	0x0000_01FF
Interrupt Select Register 114	INT_SEL114	0x0224	32	0x0000_01FF
Interrupt Select Register 115	INT_SEL115	0x0228	32	0x0000_01FF
Interrupt Select Register 116	INT_SEL116	0x022C	32	0x0000_01FF
Interrupt Select Register 117	INT_SEL117	0x0230	32	0x0000_01FF
Interrupt Select Register 118	INT_SEL118	0x0234	32	0x0000_01FF
Interrupt Select Register 119	INT_SEL119	0x0238	32	0x0000_01FF
Interrupt Select Register 120	INT_SEL120	0x023C	32	0x0000_01FF
Interrupt Select Register 121	INT_SEL121	0x0240	32	0x0000_01FF
Interrupt Select Register 122	INT_SEL122	0x0244	32	0x0000_01FF
Interrupt Select Register 123	INT_SEL123	0x0248	32	0x0000_01FF

register name	notation	offset address	bit width	reset value
Interrupt Select Register 124	INT_SEL124	0x024C	32	0x0000_01FF
Interrupt Select Register 125	INT_SEL125	0x0250	32	0x0000_01FF
Interrupt Select Register 126	INT_SEL126	0x0254	32	0x0000_01FF
Interrupt Select Register 127	INT_SEL127	0x0258	32	0x0000_01FF
Vector Shared Interrupt Select Register 128	INT_VSSEL128	0x025C	32	0x0000_0000
Vector Shared Interrupt Select Register 129	INT_VSSEL129	0x0260	32	0x0000_0000
Vector Shared Interrupt Select Register 130	INT_VSSEL130	0x0264	32	0x0000_0000
Vector Shared Interrupt Select Register 131	INT_VSSEL131	0x0268	32	0x0000_0000
Vector Shared Interrupt Select Register 132	INT_VSSEL132	0x026C	32	0x0000_0000
Vector Shared Interrupt Select Register 133	INT_VSSEL133	0x0270	32	0x0000_0000
Vector Shared Interrupt Select Register 134	INT_VSSEL134	0x0274	32	0x0000_0000
Vector Shared Interrupt Select Register 135	INT_VSSEL135	0x0278	32	0x0000_0000
Vector Shared Interrupt Select Register 136	INT_VSSEL136	0x027C	32	0x0000_0000
Vector Shared Interrupt Select Register 137	INT_VSSEL137	0x0280	32	0x0000_0000
Vector Shared Interrupt Select Register 138	INT_VSSEL138	0x0284	32	0x0000_0000
Vector Shared Interrupt Select Register 139	INT_VSSEL139	0x0288	32	0x0000_0000
Vector Shared Interrupt Select Register 140	INT_VSSEL140	0x028C	32	0x0000_0000
Vector Shared Interrupt Select Register 141	INT_VSSEL141	0x0290	32	0x0000_0000
Vector Shared Interrupt Select Register 142	INT_VSSEL142	0x0294	32	0x0000_0000
Vector Shared Interrupt Select Register 143	INT_VSSEL143	0x0298	32	0x0000_0000
Software Interrupt Event Register	INT_SWIER	0x029C	32	0x0000_0000
Event Enable Register	INT_EVTER	0x02A0	32	0x0000_0000
Interrupt Enable Register	INT_IER	0x02A4	32	0xFFFF_FFFF

10.5.1 NMI pin non-maskable interrupt control register (INT_NMICR)

NMI Interrupt Control Register (INT_NMICR)

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															

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b8	Reserved	-	Reads "0", writes "0". 0: Disable	R/W
b7 digital filter function		NFENNMI digital filter enable		R/W
			1: Licensed digital filter function	
b7~b6	Reserved	-Reserved	Read "0", write "0". 0 0: PCLK3 0 1: PCLK3/8 1 0: PCLK3/32 1 1:: PCLK3/64	R/W
b5~b4 Clock Selection	SMPCLK[1:0]	Filter Sample		R/W
b3~b1	Reserved	-Reserved	Read "0", write "0". 0: Falling	R/W
b0 edge	NMITRG	Trigger edge selection		R/W
			1: Rising edge	

10.5.2 Non-maskable Interrupt Enable Register

(INT_NMIENR) NMI Interrupt Enable Register

(INT_NMIENR) 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
-	-	-	-	WDT ENR	BUS MEN R	REC CEN R	REP ENR	-	-	XTA LST PEN R	-	PVD 2EN R	PVD 1EN R	SWD TEN R	NMI ENR

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b13	Reserved	-	Reads "0", writes "0".	R/W
b12	Reserved	-	Reads "0", writes "0".	R/W
b11	WDTENR	WDT underflow/refresh interrupt selection interrupt as non-maskable interrupt source	0: Disable 1: Select interrupts as non-maskable interrupt sources	R/W
b10	BUSMENR	MPU main bus error interrupt selection interrupt as a non-maskable interrupt source	0: Disable 1: Select interrupts as non-maskable interrupt sources	R/W
b9	RECCENR as non-maskable interrupt source	SRAM ECC checksum error interrupt selection	0: Disable interrupt 1: Select interrupts as non- maskable interrupt sources	R/W
b8	REPENR maskable interrupt source	SRAM parity error interrupt selection	0: Disable interrupt as non- maskable interrupt sources 1: Select interrupts as non- maskable interrupt sources	R/W
b7	Reserved	-Reserved	Read "0", write "0".	R/W
b6	Reserved	-Reserved	Read "0", write "0".	R/W
b5	XATLSTPENR	Detect master transmitter stop interrupt selection Disable interrupt as non-maskable interrupt source	0: 1: Select interrupts as non-maskable interrupt sources	R/W
b4	Reserved	-Reserved	Reads "0", writes "0".	R
b3	PVD2ENR as non-maskable interrupt source	Low voltage detection PVD2 interrupt selection	0: Disable interrupt 1: Select interrupts as non-maskable interrupt sources	R/W
b2	PVD1ENR as non-maskable interrupt source	Low voltage detection PVD1 interrupt selection	0: Disable interrupt 1: Select interrupts as non-maskable interrupt sources	R/W
b1		SWDTENRSWDT underflow/refresh interrupt selection Disable interrupt as non-maskable interrupt source	0: 1: Select interrupts as non-maskable interrupt sources	R/W
b0	NMIENR maskable interrupt source	NMI pin interrupt selection	0: Disable interrupt as a non- maskable interrupt source 1: Select interrupts as non-maskable interrupt sources	R/W

10.5.3 Non-maskable interrupt flag register (INT_NMIFR)

NMI Flag Register (INT_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
-	-	-	-	WDT FR	BUS MFR	REC CFR	REP FR	-	-	XTA LST PFR		PVD 2FR	PVD 1FR	SWD TFR	NMI FR

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b13	Reserved	-	Reads "0", writes "0".	R
b12	Reserved	-	Reads "0", writes "0".	R
b11	WDTFR	WDT underflow/refresh interrupt flag	0: No WDT underflow/refresh request occurred 1: WDT underflow/refresh request occurs	R
b10	BUSMFR	MPU main bus error interrupt flag	0: No MPU main bus error request occurred 1: MPU master bus error request occurred	R
b9	RECCFR	SRAM DED Check Error Interrupt Flag	0: No SRAM DED Check Error request has occurred. 1: SRAM DED checksum error request occurred	R
b8	REPFR	SRAM parity error interrupt flag	0: No SRAM parity error request occurred 1: SRAM parity error request occurred	R
b7	Reserved	-Reserved	Reads "0", writes "0".	R
b6	Reserved	-Reserved	Reads "0", writes "0".	R
b5	XTALSTPFR	Detect master oscillator stop interrupt flag Master Transmitter Stop request has occurred.	0: No Detect R 1: Occurrence of detection of master oscillator stopping application	
b4	Reserved	-Reserved	Reads "0", writes "0".	R
b3	PVD2FR	Low voltage detection PVD2 interrupt flag	0: no low voltage detection PVD2 request occurred R 1: Occurrence of low voltage detection PVD2 application	
b2	PVD1FR	Low voltage detection PVD1 interrupt flag	0: no low voltage detection PVD1 request occurred R 1: Occurrence of low voltage detection PVD1 application	
b1	SWDTFR	SWDT overflow/refresh interrupt flag	0: no SWDT overflow/refresh request occurred R 1: SWDT underflow/refresh request occurs	
b0		NMIFRNMI pin interrupt flag	0: No NMI pin request occurred R 1: NMI pin application occurs	

10.5.4 Non-maskable interrupt flag clear register (INT_NMICFR)

NMI Clear Flag Register (INT_NMICFR)

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
-	-	-	-	WDT CFR	BUS MCF R	REC CCF R	REP CFR	-	-	XTA LST PCF R	-	PVD 2CF R	PVD 1CF R	SWD TCF R	NMI CFR

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b13	Reserved	-	Reads "0", writes "0".	R/W [Note 1]
b12	Reserved	-	Reads "0", writes "0".	R/W [Note 1]
b11	WDTCFR	WDT underflow/refresh interrupt flag clear 1].	0: invalid 1: Clear WDT underflow/refresh flag	R/W [Note 1]
b10	BUSMCFR	MPU main bus error interrupt flag clear 1]	0: Invalid 1: Clear the MPU main bus error flag	R/W [Note 1]
b9	RECCCFR	SRAM DED checksum error interrupt flag clear 1].	0: invalid 1: Clear SRAM DED checksum error flag	R/W [Note 1]
b8	REPCFR	SRAM parity error interrupt flag clear	0: invalid 1: Clear the SRAM parity error flag	R/W [Note 1]
b7	Reserved	-Reserved	Read "0", write "0".	R/W [Note 1]
b6	Reserved	-Reserved	Read "0", write "0".	R/W [Note 1]
b5	XTALSTPCFR	Detect master oscillator stop interrupt flag clear 1: Clear the detection of the master oscillator stop sign	0: Invalid 1: Clear the detection of the master oscillator stop sign	R/W [Note 1]
b4	Reserved	-Reserved	Reads "0", writes "0".	R
b3	PVD2CFR	Low voltage detection PVD2 interrupt flag clear 0: invalid 1: Clear low voltage detection PVD2 flag	0: invalid 1: Clear low voltage detection PVD2 flag	r/w
b2	PVD1CFR	Low voltage detection PVD1 interrupt flag clear 0: invalid 1: Clear low voltage detection PVD1 flag	0: invalid 1: Clear low voltage detection PVD1 flag	[note 1] r/w
b1	SWDTCFR	SWDT underflow/refresh interrupt flag clear 0: invalid 1: Clear SWDT	0: invalid 1: Clear SWDT	[note 1]
b0		NMICFRNMI pin interrupt flag cleared	0: Invalid 1: Clear the NMI pin flag	r/w [note 1] r/w

[note 1]

[Note 1] Only "1" can be written and "0" can be read out.

10.5.5 External pin interrupt control register (INT_EIRQCRx) (x=0~15)

EIRQ Control Register (INT_EIRQCRx)

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
-	-	-	-	-	-	-	-	EEFN	-	EISMPCLK [1:0]	-	-	-	EIRQTRG[1:0]	

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b10	Reserved	-	Reads "0", writes "0".	R/W
b9	Reserved	-	Reads "0" and writes "0".	R/W
b8	Reserved	-	Reads "0" and writes "0".	R/W
b7 function	EFENEIRQ digital filter enable	0: Disable digital filter 1: Allow digital filter function		R/W
b6	Reserved	-Reserved	Read "0", write "0".	R/W
b5~b4 Selection	EISMPCLK[1:0]	Filter Sample Clock 0 0: PCLK3 0 1: PCLK3/8 1 0: PCLK3/32 1 1:: PCLK3/64		R/W
b3~b2 "0".	Reserved R/W	-0" for reading, "0" for writing.	Read "0", write	
b1~b0 Selection	EIRQTRG[1:0]	Trigger 0 0: falling edge 0 1: Rising edge 1 0: Double edge 1 1:: low level		R/W

10.5.6 External Pin Interrupt Flag Register (INT_EIFR)

EIRQ Flag Register (INT_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 R15	EIF R14	EIF R13	EIF R12	EIF R11	EIF R10	EIF R9	EIF R8	EIF R7	EIF R6	EIF R5	EIF R4	EIF R3	EIF R2	EIF R1	EIF R0
<hr/>															
classifier for honorific people	marking	celebrity	functionality										fill out or in (information on a form)		
b31~b10	Reserved	-	Reads "0" and writes "0".										R/W		
b15~b0	EIFR	EIFR flag bit	0: EIRQ event did not occur, or write EIFCR bit clear bit 1: Selected EIRQ event occurred										R		

10.5.7 External Pin Interrupt Flag Clear

Register (EIFCR) EIRQ Flag Clear Register

(INT_EIFCR) 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
EIFCR [15:0]															
classifier for honorific people	marking	celebrity	functionality										fill out or in (information on a form)		
b31~b10	Reserved	-	Reads "0" and writes "0".										R/W		
b15~b0	EIFCR	EIFR clear bit	0: Write "0" is invalid. 1: Write "1" to clear INT_EIFR register										R/W		

10.5.8 Interrupts/Events Select Register

(INT_SEL0~31) Interrupt Source Select Register

(INT_SEL0~31) 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
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	INTSEL[8:0]

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b9	Reserved	-	Reads "0", writes "0".	R/W
b8~b0	INTSEL[8:0]	Interrupt event request selection	9'h000~9'h1FE: 10.3.2 The event corresponding to the interrupt event request serial number	R/W

10.5.9 Interrupt Select Register (INT_SEL32~127)

Interrupt Source Select Register (INT_SEL32~127)

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												
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	INTSEL[8:0]												
<hr/>																											
Bit Flag	Bit Name	Function	Read/Write																								
b31~b9	Reserved	-Reserved																									
write "0".	R/W																										
Select the event corresponding to the serial number of 10.3.2 interrupt event request respectively, the specific correspondence is as follows:																											
INT_SEL32~INT_SEL37: Select the interrupt event request corresponding to 9'h000~9'h01F, other selections are invalid.																											
INT_SEL38~INT_SEL43: Select the interrupt event request corresponding to 9'h020~9'h03F, other selections are invalid.																											
INT_SEL44~INT_SEL49: Select the interrupt event request corresponding to 9'h040~9'h05F, other selections are invalid.																											
INT_SEL50~INT_SEL55: Select the interrupt event request corresponding to 9'h060~9'h07F, other selections are invalid.																											
INT_SEL56~INT_SEL61: Select the interrupt event request corresponding to 9'h080~9'h09F, other selections are invalid.																											
INT_SEL62~INT_SEL67: Select the interrupt event request corresponding to 9'h0A0~9'h0BF, other selections are invalid.																											
INT_SEL68~INT_SEL73: Select the interrupt event request corresponding to 9'h0C0~9'h0DF, other selections are invalid.																											
INT_SEL74~INT_SEL79: Select the interrupt event request corresponding to 9'h0E0~9'h0FF, other selections																											
Invalid.																											
INT_SEL80~INT_SEL85: Select the interrupt event request corresponding to 9'h100~9'h11F, other selections are invalid.																											
INT_SEL86~INT_SEL91: Select the interrupt event request corresponding to 9'h120~9'h13F, other selections are invalid.																											
INT_SEL92~INT_SEL97: Select the interrupt event request corresponding to 9'h140~9'h15F, other selections are invalid.																											
INT_SEL98~INT_SEL103: Select the interrupt event request corresponding to 9'h160~9'h17F, other selections are invalid.																											
INT_SEL104~INT_SEL109: Select the interrupt event request corresponding to 9'h180~9'h19F, other selections are invalid.																											
INT_SEL110~INT_SEL115: Select the interrupt event request corresponding to 9'h1A0~9'h1BF, other selections are invalid.																											
INT_SEL116~INT_SEL121: Select the interrupt event request corresponding to 9'h1C0~9'h1DF, other selections are invalid.																											
INT_SEL122~INT_SEL127: Select the interrupt event request corresponding to 9'h1E0~9'h1FF, other selections are invalid.																											
R/W																											

10.5.10 Vector Shared Interrupt Select Register (INT_VSEL128~143)

Vector Sharing Interrupt Source Select Register (INT_VSEL128~143) 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	b10	b9	b8
VSE L31	VSE L30	VSE L29	VSE L28	VSE L27	VSE L26	VSE L25	VSE L24	VSE L23	VSE L22	VSE L21	VSE L20	VSE L19	VSE L18	VSE L17	VSE L16										
					b7	b6	b6	b5	b4	b3	b2	b1	b0												
VSE L15	VSE L14	VSE L13	VSE L12	VSE L11	VSE L10	VSE L9	VSE L8	VSE L7	VSE L6	VSE L5	VSE L4	VSE L3	VSE L2	VSE L1	VSE L0										

Bit Flag Bit Name Function Read/Write

b31~b0	VSEL[31:0]	INT_VSEL128: Each bit enables the interrupt event request corresponding to 9'h000~9'h01F respectively.	R/W
Interrupt		INT_VSEL129: Each bit enables the interrupt event request corresponding to 9'h020~9'h03F respectively.	
	enable	INT_VSEL130: Each bit enables the interrupt event request corresponding to 9'h040~9'h05F respectively.	
		INT_VSEL131: Each bit enables the interrupt event request corresponding to 9'h060~9'h07F respectively.	
		INT_VSEL132: Each bit enables the interrupt event request corresponding to 9'h080~9'h09F respectively.	
		INT_VSEL133: Each bit enables the interrupt event request corresponding to 9'h0A0~9'h0BF respectively.	
		INT_VSEL134: Each bit enables the interrupt event request corresponding to 9'h0C0~9'h0DF respectively.	
		INT_VSEL135: Each bit enables the interrupt event request corresponding to 9'h0E0~9'h0FF respectively.	
		INT_VSEL136: Each bit enables the interrupt event request corresponding to 9'h100~9'h11F respectively.	
		INT_VSEL137: Each bit enables the interrupt event request corresponding to 9'h120~9'h13F respectively.	
		INT_VSEL138: Each bit enables the interrupt event request corresponding to 9'h140~9'h15F respectively.	
		INT_VSEL139: Each bit enables the interrupt event request corresponding to 9'h160~9'h17F respectively.	
		INT_VSEL140: Each bit enables the interrupt event request corresponding to 9'h180~9'h19F respectively.	
		INT_VSEL141: Each bit enables the interrupt event request corresponding to 9'h1A0~9'h1BF respectively.	
		INT_VSEL142: Each bit enables the interrupt event request corresponding to 9'h1C0~9'h1DF respectively.	
		INT_VSEL143: Each bit enables the interrupt event request corresponding to 9'h1E0~9'h1FF respectively. Refer to Table 10-2 Interrupt Event Request Sequence Number and Selection for the interrupt event request sequence number.	

10.5.11 Stop Mode Wake-Up Event Enable Register (INT_WUPEN)

Soft-standby Wake Up Enable Register (INT_WUPEN)

Reset Value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	RXW UEN	-	TMR 0WU EN	RTC PRD WUE N	RTC ALM WUE N	WKT MWU EN	CMP IOW UEN	PVD 2WU EN	PVD 1WU EN	SWD TWU EN
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EIRQWUEN [15:0]															

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
USART1_WUPI Stop Mode Wakeup Enable				
b25	RXWUEN	-	Wakeup Disable 0: Wake-up Permit	0: R/W
b31~27	Reserved	-	Reads "0", writes "0" 1: Wake-up Permit	R/W
b26	Reserved	-	Reads "0", writes "0". 0" for reading, "0" for writing.	R/W Reads
b24	Reserved	"0", writes "0".	R/W	
b23	TMR0WUEN	TMRO1_GCMA Stop Mode Wakeup Enable	Wakeup Disable 0: Wake-up Permit	0: R/W
b22	RTCPRDWUEN	RTC_PRD stop mode wake-up enable	Wakeup Disable 0: Wake-up disabled 1: Wake-up Permit	R/W
b21	RTCALMWUEN	RTC_ALM Stop Mode Wakeup Enable	0: Wakeup Disable 1: Wake-up Permit	R/W
b20	WKTM_PRD WKT MWUEN	WKTM_PRD Cycle Stop Mode Wakeup Enable	Wakeup Disable 0: Wake-up disabled 1: Wake-up Permit	0: R/W
B19	CMPI1WUEN	ACMP1 stop mode wakeup enable	0: Wake-up disabled 1: Wake-up Permit	R/W
B18	PVD2WUEN	PVD_PVD2 stop mode wake-up enable	0: Wake-up disabled 1: Wake-up Permit	R/W
B17	PVD1WUEN	PVD_PVD1 stop mode wake-up enable	0: Wake-up disabled 1: Wake-up Permit	R/W
B16	SWDTWUEN	SWDT_REFUDF stop mode wakeup enable	Wake-up disabled 0: Wake-up disabled 1: Wake-up Permit	0: R/W
B15~b0	EIRQWUEN [15:0]	EIRQ stop mode wake-up enable	0: Wake-up disabled 1: Wake-up Permit	R/W

10.5.12 Software Interrupt/Event Register (INT_SWIER)

Software Interrupt & Event Register (INT_SWIER) reset

value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	
SWI E31	SWI E30	SWI E29	SWI E28	SWI E27	SWI E26	SWI E25	SWI E24	SWI E23	SWI E22	SWI E21	SWI E20	SWI E19	SWI E18	SWI E17	SWI E16
b16 b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	
SWI E15	SWI E14	SWI E13	SWI E12	SWI E11	SWI E10	SWI E9	SWI E8	SWI E7	SWI E6	SWI E5	SWI E4	SWI E3	SWI E2	SWI E1	SWI E0
b0															

Bit Flag Bit Name Function Read/Write

0: Invalid

b31~b0 SWIE Software
Interrupt/Event Register Bit

1: Software interrupt event occurs
Note: Software interrupt/event
occurs after writing "1". Write
"0" to clear.

R/W

10.5.13 Event Enable Register

(INT_EVTER) Event Enable Register

(INT_EVTER) Reset Value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	
EVT E31	EVT E30	EVT E29	EVT E28	EVT E27	EVT E26	EVT E25	EVT E24	EVT E23	EVT E22	EVT E21	EVT E20	EVT E19	EVT E18	EVT E17	EVT E16
b16 b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	
EVT E15	EVT E14	EVT E13	EVT E12	EVT E11	EVT E10	EVT E9	EVT E8	EVT E7	EVT E6	EVT E5	EVT E4	EVT E3	EVT E2	EVT E1	EVT E0
b0															

Bit Flag Bit Name Function Read/Write

0: Event

b31~b0 EVTE Event Enable Register Bit
Selection Disable

R/W

1: Event Selection License

10.5.14 Interrupt Enable Register

(INT_IER) Interrupt Enable Register

(INT_IER) Reset Value: 0xFFFF_FFFF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	
IER 31	IER 30	IER 29	IER 28	IER 27	IER 26	IER 25	IER 24	IER 23	IER 22	IER 21	IER 20	IER 19	IER 18	IER 17	IER 16
b16 b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	
IER 15	IER 14	IER 13	IER 12	IER 11	IER 10	IER 9	IER 8	IER 7	IER 6	IER 5	IER 4	IER 3	IER 2	IER 1	IER 0
b0															

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
			Register bits 0 to 31 correspond to NVIC interrupt vectors 0 to 31, respectively. When disabled, each interrupt vector has no effect on the interrupt event requests selected by the corresponding selection registers INTSEL[8:0] will not be accepted by the NVIC.	
b31~b0	IER	Interrupt Enable Register Bits	Collections. 0: Interrupt event request selected by INTSEL[8:0] and software interrupt event request are disabled 1: Interrupt event request selected by INTSEL[8:0] is licensed with software interrupt event request	R/W

10.6 Precautions for use

For a description of ARM core interrupts, refer to the ARM manual ARM Processor Cortex®-M4 Technical Reference Manual (ARM DDI 0439D).

11 Automated Operating System (AOS)

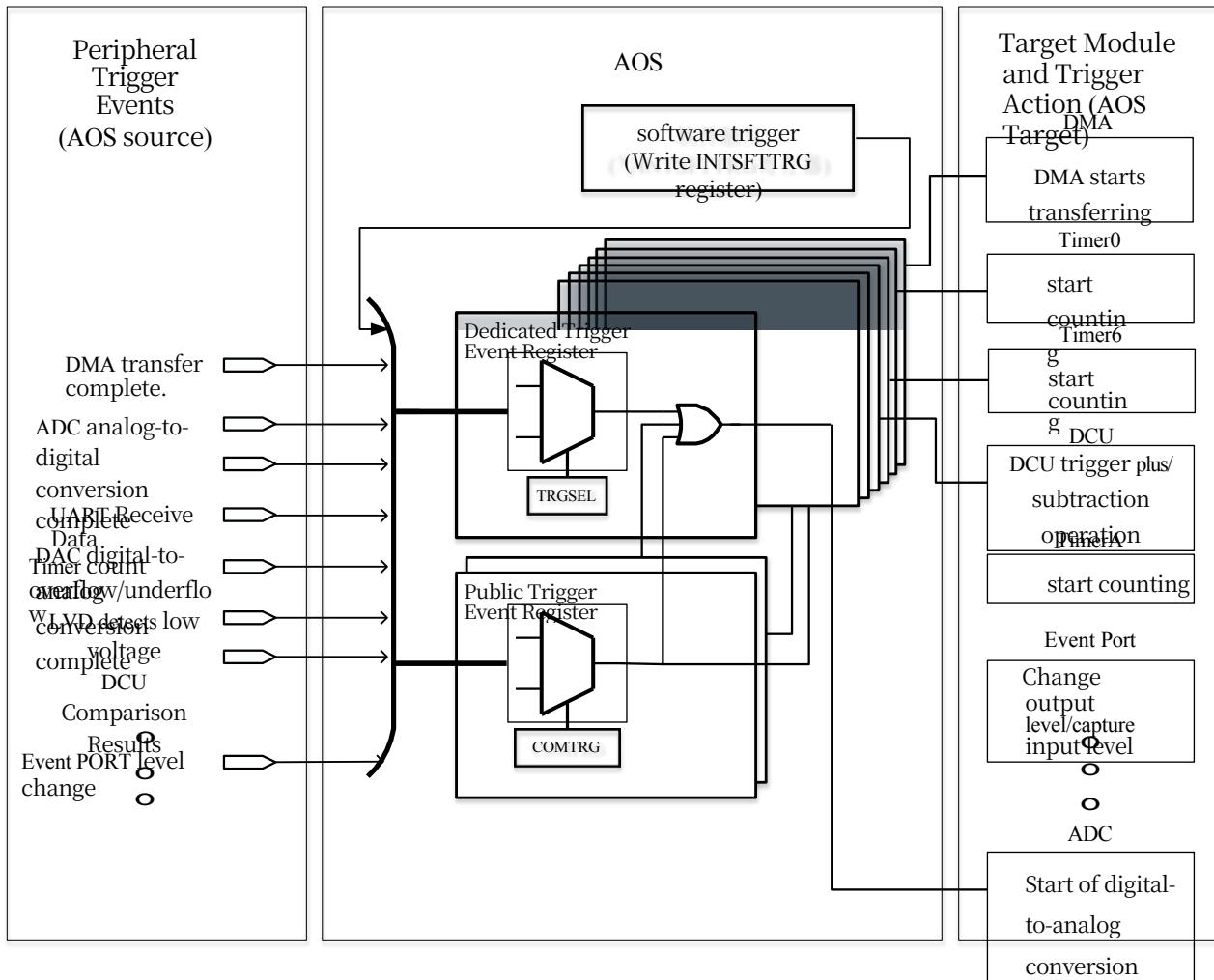
11.1 summary

The Automatic Operation System (AOS) is used to realize the linkage between peripheral hardware circuits without the aid of the CPU. The events generated by the peripheral circuits are used as the AOS Source, such as the comparison match of the timer, the timing overflow, the cycle signal of the RTC, the various states of sending and receiving data of the communication module (idle, full of receiving data, end of sending data, empty of sending data) and the end of the conversion of the ADC, etc., which triggers the actions of the other peripheral circuits. The triggered peripheral circuit action is called AOS Target.

11.1.1 Function Overview

- There are 195 kinds of AOS sources, except for special restrictions, each AOS target can select one of them as trigger source, and two additional trigger sources can be selected through Common Trigger Source Selection Register 1 and Common Trigger Source Selection Register 2, and the AOS target can be triggered when a trigger event occurs in any of the three trigger sources. All AOS targets share these two common trigger sources.
- It can be triggered by peripheral circuits in hardware or by software by writing registers.
- The peripheral circuits that can be used as AOS targets act as follows:
 - 4 DCU trigger targets for triggering DCU1~DCU4
 - 9 DMA trigger targets for two 4-channel DMA start data transfers and one DMA event trigger channel reset
 - 2 Advanced Control Timers (Timer6) Trigger Targets
 - 1 General Purpose Timer 0 (Timer0) Trigger target
 - 2 Event Port trigger targets, where Event Port Group1 and Event Port Group2 share an AOS target, and Event Port Group3 and Event Port Group4 share an AOS target
 - 2 General Purpose Timer A (TimerA) Trigger Targets
 - 1 Temperature Sensor (OTS) Trigger Targets
 - 2 groups of 2 AD trigger targets each for AD1~AD2 sequence triggering

11.1.2 Module Schematic



11.2 Functional Description

11.2.1 AOS Source Event List

The AOS source event number is shown in Table 10-2 in the "Interrupt Event Request Number" section of the "Interrupt Controller (INTC)" chapter, and the event marked with "✓" in the "Selectable as Event" column of the table can be used as an AOS source.

11.2.2 AOS Target List

module (in software)	serial number	movements
DCU1	0	Trigger addition/subtraction operations
DCU2	1	Trigger addition/subtraction operations
DCU3	2	Trigger addition/subtraction operations
DCU4	3	Trigger addition/subtraction operations
DMA1	4	Trigger channel 0 to start transmission Trigger channel 1 to start transmission Trigger channel 2 to start transmission Trigger channel 3 to start transmission
DMA2	5	Trigger channel 0 to start transmission Trigger channel 1 to start transmission Trigger channel 2 to start transmission Trigger channel 3 to start transmission
DMA1&2	6	Event triggered channel reset
Timer6	7	Trigger Start Count
Timer0	8	Trigger Start Count
Event Port	9	Trigger Event Port1&2 action Trigger Event Port3&4 action
TimerA	10	Trigger to start counting/capturing
OTS	11	Trigger to start temperature measurement
ADC1	12	Trigger to start analog-to-digital conversion
ADC2	13	Trigger to start analog-to-digital conversion

11.3 Action Description

11.3.1 Dedicated Trigger Sources

Peripheral circuit modules with AOS targets have a dedicated peripheral trigger source selection register for each AOS target. When the event number corresponding to the AOS source is written to this register, the AOS target selects this AOS source as the trigger source. When an event occurs in the AOS source, the event will be transmitted to the AOS target through the AOS, and the peripheral circuitry of the AOS target will start to act according to its own settings.

11.3.2 common trigger source

In addition to the dedicated peripheral trigger source selection registers for each AOS target, the AOS is configured with two common trigger source selection registers (AOS_COMTRG1,AOS_COMTRG2). These registers are used to realize the function of multiple AOS sources triggering the same AOS target. First, enable the common trigger source in the peripheral trigger source selection register for AOS target, and then write the corresponding event number of AOS source in the common trigger source selection register. When an event occurs in the AOS source, the event will be passed to the AOS target through the public trigger source of AOS, and the peripheral circuit as the AOS target will start to act according to its own setting. When the dedicated trigger source and the public trigger source are set at the same time, up to three AOS sources can trigger the same AOS target at the same time, and the AOS target will be triggered when a trigger event occurs in any one of the three AOS sources.

All AOS targets share these two common trigger sources. Therefore, when other AOS targets do not use the events selected in the Common Trigger Source Selection Register, it is necessary to disable the Common Trigger Source Enable position in its dedicated Peripheral Trigger Source Selection Register to prevent incorrect triggering actions.

11.4 Register Description

List of registers

Register base address: 0x4001_0800

abridge	name (of a thing)	offset address
INTSFTTRG	Peripheral Trigger Event Register	0x00
DCU_TRGSELx(x=1~4)	DCU Trigger Source Selection Register	0x04,0x08,0x0c,0x10
DMA1_TRGSELx(x=0~3)	DMA1 Transmit Startup Trigger Source Selection Register	0x14,0x18,0x1c,0x20
DMA2_TRGSELx(x=0~3)	DMA2 Transmit Startup Trigger Source Selection Register	0x24,0x28,0x2c,0x30
DMA_RC_TRGSEL	DMA Channel Reset Trigger Source Selection Register	0x34
TMR6_TRGSELx(x=0~1)	Timer6 Hardware Trigger Event Selection Register	0x38,0x3C
TMR0_TRGSEL	Timer0 Trigger Select Register	0x40
PEVNT_TRGSEL12	Event Port1,2 Trigger Source Selection Registers	0x44
PEVNT_TRGSEL34	Event Port3,4 Trigger Source Selection Registers	0x48
TMRA_TRGSEL0	TimerA Internal Trigger Event Selection Register 0	0x4C.
TMRA_TRGSEL1	TimerA Internal Trigger Event Selection Register 1	0x50
OTS_TRGSEL	OTS Trigger Source Selection Register	0x54
ADC1_TRGSELx(x=0,1)	A/D1 Activate on-chip trigger source selection registers	0x58, 0x5C
ADC2_TRGSELx(x=0,1)	A/D2 Initiates on-chip trigger source selection registers	0x60, 0x64
AOS_COMTRG1	Common Trigger Source Selection Register 1	0x68
AOS_COMTRG2	Common Trigger Source Selection Register 2	0x6C

11.4.1 Peripheral Trigger Event

Register (INTSFTTRG) Register

Description: writing this register will generate a trigger event. Offset address: 0x00

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															
SFTG															

classifier	marking
for	
honorific	
people	
b31~b1	Reserved
b0	SFTG

11.4.2 DCU Trigger Source Selection Register (DCU_TRGSELx) (x=1~4)

Register description: After DCU selects hardware trigger startup mode, the number of the event to be triggered will be written into this register, and DCU will be triggered by the event to startup and perform operation when the peripheral circuit event corresponding to the number occurs. When DCU_TRGSEL1 writes the event number, DCU1 will be triggered when the numbered event occurs; when DCU_TRGSEL2 writes the event number, DCU2 will be triggered when the numbered event occurs; when DCU_TRGSEL3 writes the event number, DCU3 will be triggered when the numbered event occurs; when DCU_TRGSEL4 writes the event number, DCU4 will be triggered when the numbered event occurs. event occurs, DCU4 will be triggered.

Offset address: 0x04, 0x08, 0x0C,

0x10 Reset value: 0x0000_01FF

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/Write
b31	COMEN[1]	Common Trigger Enable AOS_COMTRG2 from triggering the DCU	0: Disables the public trigger event of AOS_COMTRG2 to trigger the DCU 1: Allow the public trigger event of AOS_COMTRG2 to trigger the DCU R/W
b30	COMEN[0]	Public Trigger Enable AOS_COMTRG1 from triggering the DCU	0: Disables the public trigger event of AOS_COMTRG1 to trigger the DCU 1: Allow the public trigger event of AOS_COMTRG1 to trigger the DCU R/W
b29~b9	Reserved	-0" for reading, "0" for writing.	Read "0", write "0". R/W
b8~b0	TRGSEL[8:0]	Trigger Source Selection event to be selected	Write the number of the peripheral circuit Refer to the [Interrupt Controller (INTC)] section for specific numbers. R/W

11.4.3 DMA1 Transmit Start Trigger Source Selection Register (DMA1_TRGSELx) (x=0~3)

Register Description: After DMA1 selects the hardware trigger start mode, the number of the event to be triggered is written into this register, and when the peripheral circuit event corresponding to the number occurs, DMA1 will be triggered to start and transmit by the event.

Offset Address: 0x14, 0x18, 0x1C,

0x20 Reset Value: 0x0000_01FF

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/Write
b31 to trigger DMA1 transfer	COMEN[1]	Common Trigger Enable	0: Disable the public trigger event of AOS_COMTRG2 1: Allow the public trigger event of AOS_COMTRG2 to trigger a DMA1 transfer
b30 to trigger DMA1 transfer	COMEN[0]	Common Trigger Enable	0: Disable the public trigger event of AOS_COMTRG1 1: Allow the public trigger event of AOS_COMTRG1 to trigger a DMA1 transfer
b29~b9	Reserved	-0" for reading, "0" for writing.	Read "0", write "0".
b8~b0	TRGSEL[8:0]	Trigger Source Selection	Selects the event number that initiates the corresponding channel for transmission. Refer to the [Interrupt Controller (INTC)] section for specific numbers.

11.4.4 DMA2 Transmit Startup Trigger Source Selection Register (DMA2_TRGSELx) (x=0~3)

Register Description: After DMA2 selects the hardware trigger startup mode, the number of the event to be triggered is written into this register, and when the peripheral circuit event corresponding to the number occurs, DMA2 will be triggered to start and transmit by this event.

Offset Address: 0x24, 0x28, 0x2C,

0x30 Reset Value: 0x0000_01FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	www.xhsc.com.cn
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/Write
b31 to trigger DMA2 transfers	COMEN[1]	Common Trigger Enable	0: Disable the public trigger event of AOS_COMTRG2 1: Allow the public trigger event of AOS_COMTRG2 to trigger DMA2 transfers
b30 from triggering DMA2 transfers	COMEN[0]	Common Trigger Enable	0: Disables the public trigger event of AOS_COMTRG1 1: Allow the public trigger event of AOS_COMTRG1 to trigger a DMA2 transfer
b29~b9	Reserved	-0" for reading, "0" for writing.	Read "0", write "0".
b8~b0	TRGSEL[8:0]	Trigger Source Selection	Selects the event number that initiates the corresponding channel for transmission. Refer to the [Interrupt Controller (INTC)] section for specific numbers.

11.4.5 DMA Channel Reset Trigger Source Selection Register (DMA_RC_TRGSEL)

Register Description: After the DMA selects the hardware-triggered startup mode, the number of the event to be triggered is written into this register, and the DMA will be triggered by the event to reset the channel when the peripheral circuit event corresponding to the number occurs. This register is shared by DMA1 and DMA2.

Offset address: 0x34

Reset value: 0x0000_01FF

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/Write
b31	COMEN[1]	Common Trigger Enable	0: Disables the public trigger event of AOS_COMTRG2 1: Allow the public trigger event of AOS_COMTRG2 to trigger the DMA channel reset
b30	COMEN[0]	Common Trigger Enable	0: Disable the public trigger event of AOS_COMTRG1 to trigger DMA channel reset 1: Allow the public trigger event of AOS_COMTRG1 to trigger the DMA channel reset
b29~b9	Reserved	-	Reads "0" and writes "0".
b8~b0	TRGSEL[8:0]	Trigger Source Selection	Select the event number that triggers the channel for reset. Refer to the [Interrupt Controller (INTC)] section for specific numbers. DMA_1, DMA_2 share a common reset trigger source.

11.4.6 Timer6 Hardware Trigger Event Select Register (TMR6_TRGSELx) (x=0~1)

Register Description: After Timer6 selects the hardware trigger startup mode, the number of the event to be triggered is written into this register, and Timer6 will be triggered by the event when the peripheral circuit event corresponding to the number occurs.

Offset Address: 0x38,

0x3C Reset Value:

0x0000_01FF

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/Write
b31	COMEN[1]	Common Trigger Enable AOS_COMTRG2 to trigger TMR6	0: Disable the public trigger event of AOS_COMTRG2 to trigger TMR6 1: Allow the public trigger event of AOS_COMTRG2 to trigger TMR6 R/W
b30	COMEN[0]	Common Trigger Enable AOS_COMTRG1 to trigger TMR6	0: Disable the public trigger event of AOS_COMTRG1 to trigger TMR6 1: Allow the public trigger event of AOS_COMTRG1 to trigger TMR6 R/W
b29~b9	Reserved	-0" for reading, "0" for writing.	Read "0", write "0".
b8~b0	TRGSEL[8:0]	Trigger source selection	Trigger source number write Refer to the [Interrupt Controller (INTC)] section for specific numbers. R/W

Attention:

- The Trigger Select Registers (TMR6_TRGSEL0~1) are 2 Unit-independent registers common to the 3 Unit Timer6.

11.4.7 Timer0 Hardware Trigger Event Select Register (TMR0_TRGSEL)

Register Description: After Timer0 selects the hardware trigger startup mode, the number of the event to be triggered is written into this register, and Timer0 will be triggered by the event when the peripheral circuit event corresponding to the number occurs.

Offset address: 0x40

Reset Value: 0x0000_01FF

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/Write
b31	COMEN[1]	Common Trigger Enable AOS_COMTRG2 from triggering TMR0	0: Disables the public trigger event of AOS_COMTRG2 to trigger TMR0 1: Allow the public trigger event of AOS_COMTRG2 to trigger TMR0 R/W
b30	COMEN[0]	Public Trigger Enable AOS_COMTRG1 from triggering TMR0	0: Disables the public trigger event of AOS_COMTRG1 to trigger TMR0 1: Allow the public trigger event of AOS_COMTRG1 to trigger TMR0 R/W
b29~b9	Reserved	-0" for reading, "0" for writing.	Read "0", write "0".
b8~b0	TRGSEL[8:0]	Trigger source selection	Trigger source number write Refer to the [Interrupt Controller (INTC)] section for specific numbers. R/W

Attention:

-The Trigger Select Register (TMR0_TRGSEL) is a separate register, common to 2 units of Timer0.

11.4.8 Event Port Trigger Source Selection Register (PEVNT_TRGSEL12, PEVNT_TRGSEL34)

PEVNT_TRGSEL12 sets the trigger source of Event Port 1 and 2, and PEVNT_TRGSEL34 sets the trigger source of Event Port 3 and 4.

Offset address: 0x44,

0x48 Reset value:

0x0000_01FF

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/Write
b31	COMEN[1]	Public Trigger Enable	0: Disable AOS_COMTRG2's public trigger event to trigger Event Port 1: Allow the public trigger event of AOS_COMTRG2 to trigger R/W the Event Port.
b30	COMEN[0]	Public Trigger Enable	0: Disable public trigger event of AOS_COMTRG1 to trigger Event Port 1: Allow public trigger event of AOS_COMTRG1 to trigger the R/W Event Port.
b29~b9	Reserved	-	Reads "0" and writes "0". R/W
b8~b0	TRGSEL[8:0]	Trigger Source Selection	Set the corresponding event number to trigger the Event Port to output the specified level or latch the I/O port input state. R/W PEVNT_TRGSEL12 sets the trigger source for Event Port 1 and 2. PEVNT_TRGSEL34 sets the trigger source for Event Port 3 and 4. Refer to the [Interrupt Controller (INTC)] section for specific numbers.

11.4.9 TimerA Internal Trigger Event Selection Register 0 (TMRA_TRGSEL0)

Register Description: After TimerA selects the hardware trigger start mode, the number of the event to be triggered is written into this register, and TimerA will be triggered by the event when the peripheral circuit event corresponding to the number occurs.

Offset address: 0x4C

Reset Value: 0x0000_01FF

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								CNTTRGSEL[8:0]							

Bit Flag	Bit Name	Function	Read/Write
b31 from triggering the TMRA counter	COMEN[1]	Common Trigger Enable	0: Disables the public trigger event of AOS_COMTRG2 1: Allow the public trigger event of AOS_COMTRG2 to trigger the TMRA counter
b30 from triggering the TMRA counter	COMEN[0]	Common Trigger Enable	0: Disables the public trigger event of AOS_COMTRG1 1: Allow the public trigger event of AOS_COMTRG1 to trigger the TMRA counter
b29~b9	Reserved	-0" for reading, "0" for writing.	Read "0", write "0".
b8~b0 Event Trigger	CNTTRGSEL[8:0]	Counter Trigger source selection	Counter Trigger Event Trigger Source Number Write Refer to the [Interrupt Controller (INTC)] section for specific numbers.

Attention:

-The internal trigger event selection registers (TMRA_TRGSEL0~1) are two independent registers common to the 6 units of TimerA.

11.4.10 TimerA Internal Trigger Event Selection Register 1 (TMRA_TRGSEL1)

Register Description: After TimerA selects the hardware trigger start mode, the number of the event to be triggered is written into this register, and TimerA will be triggered by the event when the peripheral circuit event corresponding to the number occurs.

Offset address: 0x50

Reset value: 0x0000_01FF

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								ICPTRGSEL[8:0]							

Bit Flag	Bit Name	Function	Read/Write
b31	COMEN[1]	Common Trigger Enable from triggering the TMRA capture action	0: Disables the public trigger event of AOS_COMTRG2 1: Allow the public trigger event of AOS_COMTRG2 to trigger the TMRA capture action R/W
b30	COMEN[0]	Public Trigger Enable from triggering the TMRA capture action	0: Disables the public trigger event of AOS_COMTRG1 1: Allow the public trigger event of AOS_COMTRG1 to trigger the TMRA capture action R/W
b29~b9	Reserved	-0" for reading, "0" for writing.	Read "0", write
"0".	R/W		
b8~b0	ICPTRGSEL[8:0]	Capture Action Trigger Event Touch source selection	Capture Action Trigger Event Trigger Source Number Write Refer to the [Interrupt Controller (INTC)] section for specific numbers. R/W

11.4.11 OTS Trigger Source Selection Register (OTS_TRGSEL)

Register Description: After the OTS selects the hardware trigger startup mode, the number of the event to be triggered is written into this register, and the OTS will be triggered to start by the event when the peripheral circuit event corresponding to the number occurs.

Offset address: 0x54

Reset Value: 0x0000_01FF

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/Write

b30 COMEN[0]
for AOS_COMTRG1

Public Trigger Enable

1: Allow the public trigger event of

AOS_COMTRG2 to trigger the OTS

R/W

0: Disable public trigger event triggering OTS

1: Allow the public trigger event of
AOS_COMTRG1 to trigger the OTS

b29~b9 Reserved

-0" for reading, "0" for writing.

Read "0", write

"0". R/W

b8~b0 TRGSEL[8:0]

Trigger Source Selection

Selects the trigger source number for

hardware trigger startup.

R/W

Refer to the [Interrupt Controller (INTC)] section
for specific numbers.

11.4.12 A/D1 conversion start on-chip trigger source selection register

ADC1_TRGSELx(x=0,1)

Register Description: After ADC1 selects the hardware trigger start mode, the number of the event to be triggered is written into this register, and ADC1 will be triggered by the event when the peripheral circuit event corresponding to the number occurs.

Offset Address: 0x58,

0x5C Reset Value:

0x0000_01FF

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/Write
b31	COMEN[1]	Common Trigger Enable	0: Disable the public trigger event of AOS_COMTRG2 to trigger ADC1 R/W 1: Allow the public trigger event of AOS_COMTRG2 to trigger ADC1
b30	COMEN[0]	Common Trigger Enable	0: Disable public trigger event of AOS_COMTRG1 R/W 1: Allow the public trigger event of AOS_COMTRG1 to trigger ADC1
b29~b9	Reserved	-0" for reading, "0" for writing.	Read "0", write "0". R/W
b8~b0	TRGSEL[8:0]	Trigger Source Selection	Write the number of the peripheral circuit event to be selected R/W Refer to the [Interrupt Controller (INTC)] section for specific numbers.

11.4.13 A/D2 conversion start on-chip trigger source selection register

ADC2_TRGSELx(x=0,1)

Register Description: After ADC2 selects the hardware trigger start mode, the number of the event to be triggered is written into this register, and ADC2 will be triggered by the event when the peripheral circuit event corresponding to the number occurs.

Offset address: 0x60,

0x64 Reset value:

0x0000_01FF

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	www.xhsc.com.cn
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/Write
b31	COMEN[1]	Common Trigger Enable AOS_COMTRG2 from triggering ADC2	0: Disables the public trigger event of AOS_COMTRG2 to trigger ADC2 1: Allow the public trigger event of AOS_COMTRG2 to trigger ADC2 R/W
b30	COMEN[0]	Common Trigger Enable AOS_COMTRG1 to trigger ADC2	0: Disable the public trigger event of AOS_COMTRG1 to trigger ADC2 1: Allow the public trigger event of AOS_COMTRG1 to trigger ADC2 R/W
b29~b9	Reserved	-0" for reading, "0" for writing.	Read "0", write "0". R/W
b8~b0	TRGSEL[8:0]	Trigger Source Selection event to be selected	Write the number of the peripheral circuit Refer to the [Interrupt Controller (INTC)] section for specific numbers. R/W

11.4.14 Common Trigger Source Selection Register 1 (AOS_COMTRG1)

Register Description: Write the number of the event that will generate the trigger in AOS_COMTRG1. When the peripheral circuit event corresponding to the number occurs, the peripheral circuit event corresponding to the number will trigger the startup of one or more AOS targets if the value of the COMEN[1] bit of the Dedicated Trigger Source Selection Register for one or more of the AOS targets is one.

Offset address: 0x68

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										COMTRG[8:0]											
<hr/>																					
classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)																	
b31~b9	Reserved	-	Reads "0" and writes "0".	R/W																	
b8~b0	COMTRG[8:0] event to be selected	Trigger source selection	Write the number of the peripheral circuit	R/W																	
Refer to the [Interrupt Controller (INTC)] section for specific numbers.																					

11.4.15 Common Trigger Source Selection Register 2 (AOS_COMTRG2)

Register Description: Write the number of the event that will generate the trigger in AOS_COMTRG2. When the peripheral circuit event corresponding to the number occurs, the peripheral circuit event corresponding to the number will trigger one or more AOS targets to start up if the value of the COMEN[0] bit of the Dedicated Trigger Source Selection Register for one or more of the AOS targets is one.

Offset address: 0x6C

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										COMTRG[8:0]											
<hr/>																					
classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)																	
b31~b9	Reserved	-	Reads "0" and writes "0".	R/W																	
b8~b0	COMTRG[8:0] event to be selected	Trigger source selection	Write the number of the peripheral circuit																		

12 Keyboard Scanning Control Module (KEYSCAN)

12.1 brief introduction

This product is equipped with one unit of Keyboard Control Module (KEYSCAN). KEYSCAN module supports keyboard array (row and column) scanning, where columns are driven by independent scanning output KEYOUT_m ($m=0\sim7$) and KEYIN_n ($n=0\sim15$) is detected as EIRQ_n ($n=0\sim15$) input. This module realizes the key recognition function by line scan query method.

KEYSCAN Key Features:

- EIRQ0~EIRQ15 can be independently selected as line inputs to the keyboard array.
- KEYOUT can be selected by register.
- The keyboard array is scanned by sequentially outputting low levels at regular intervals.
- The scan time can be set.
- IRQ interrupt detection stops scanning and locates the pressed key based on the SSR.INDEX value and the IRQ interrupt flag (INT_EIFR.EIFR).

12.2 KEYSCAN System Block Diagram

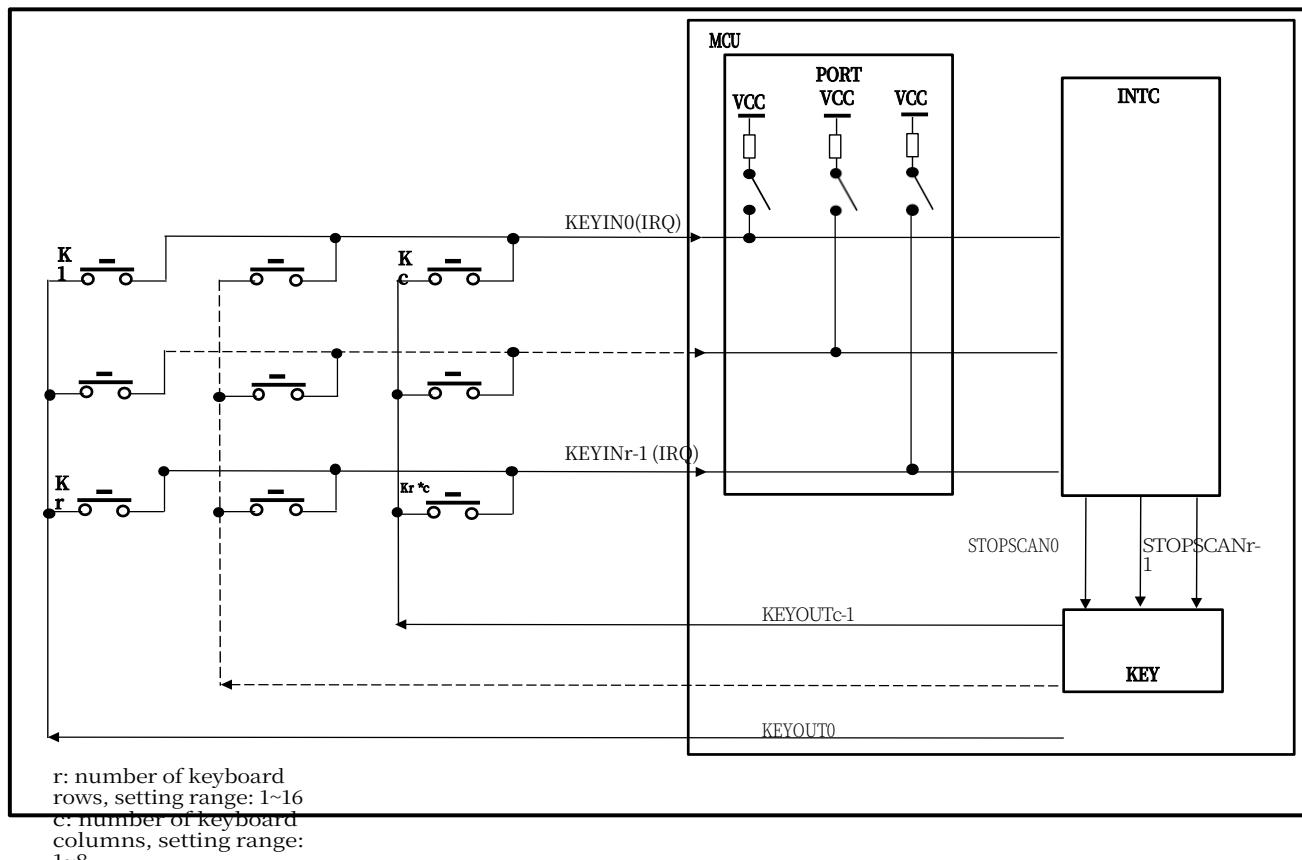


Figure 12-1 KEYSCAN System Block Diagram

12.3 Pin

**Descrip
tion**

Table 12-1 KEYS CAN Pin Descriptions

pin name	orientations	Functional Description
KEYINn	importation	Keyboard line input signal
KEYOUTm	exports	Keypad Column Output Signal

n:0~15 m:0~7

12.4 Functional Description

In this chapter, the keyboard scanning function and key recognition function are explained in detail.

12.4.1 keystroke recognition

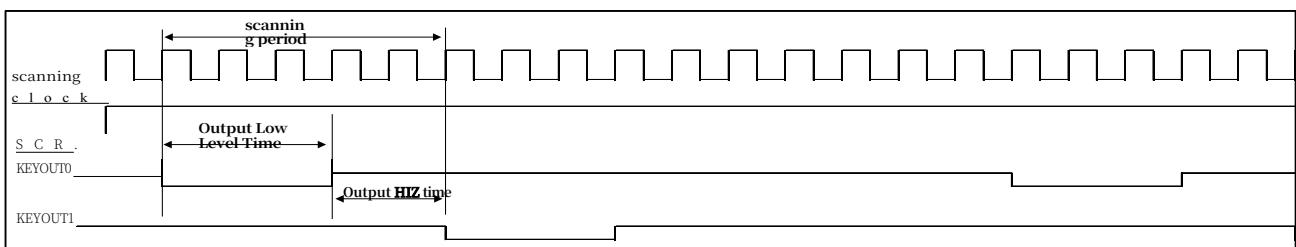
When a key is pressed, the rows and columns of the keypad are shorted and the rows generate a falling edge, which generates the EIRQ interrupt flag, which locates the currently pressed key by comparing the value of the interrupt flag bits (INT_EIFR.EIFR) and SSR.INDEX[2:0].

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

12.4.2 Keyboard scanning function

The keyboard scanning function is to continually cycle through the columns of the keyboard array to output a low level so that when a key is pressed, the corresponding EIRQ interrupt flag is generated.

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



12.4.3 Precautions for use

This module drives the keypad columns, while the keypad row detection is realized by the external EIRQ function of the interrupt control module (INTC). EIRQ needs to select the falling edge detection and turn on the digital filtering function to set the appropriate filtering time.

If this function is used in the STOP mode, the internal low-speed oscillator LRC or the external low-speed oscillator XTAL32 should be selected as the scanning clock after setting the scanning-related parameters.

If the internal pull-up resistor is used, please refer to the PORT characteristics to select the appropriate sweep time as well as the filter time.

12.5 Register Description

KEYSCAN_BASE_ADDR: 0x4005_0C00

Table 12-2 KEYS defense Register List

register name	notation	offset address	bit width	reset value
KEYSCAN scan control registers	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

12.5.1 KEYS defense Scan Control Register (KEYSCAN_SCR)

KEYSCAN Scan 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
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]															

classif erfor honorif ic people	markin g	celebrity	functionality	fill out or in (information on a form)
			KEYOUT output low HIZ time (number of scan clocks) Scan period = output low time + output HIZ time Setting value: Number of HIZ cycles 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 validly when SER.SEN = 0	
b31~b29	T-HIZ[2:0]	Output HIZ time		R/W
b28~b24	T_LLEVEL[4:0]	Output Low Level Time		R/W
b23~b22	Reserved	-	KEYOUT output low time (number of scan clocks) Scan period = output low time + output HIZ time Output low time = number of scan clocks to the T_LLEVEL power of 2 Note: SCR.T-LLEVEL[4:0] can only be set validly when SER.SEN=0 and 00000b and 00001b settings are prohibited, and the maximum value that can be set is 11000b.	R
b21~b20	CKSEL[1:0]	Scan Clock Source Selection Bits	Scan Clock Source Select Bit 00b:System clock HCLK 01b:Internal low-speed oscillator LRC 10b:External low-speed oscillator XTAL32 11b:Setting Prohibition Note: SCR.CKSEL[1:0] can only be set validly when SER.SEN=0	R/W
b19	Reserved	-Reserved	Read "0", write "0". KEYOUT output selection bit set value: Output	R

b18~b16	KEYOUTSEL [2:0]	Output Selection	000b:	prohibited	R/W
			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 validly when

SER.SEN=0

Line input selection bit, the selected line is used as a line
of the keyboard array and is detected as EIRQn (n: 0~15)

R/W

b15~b0 KEYINSEL[15:0] Line Input
Select Bits

KEYINSEL[n]=0: KEYINSEL[n] is not used as a line of the
keyboard array KEYINSEL[n]=1: KEYINSEL[n] is used as a
line of the keyboard array n: range 0~15

Note: SCR.KEYINSEL[15:0] can only be set validly when

SER.SEN = 0

12.5.2 KEYS defense Scan Enable Register (KEYSCAN_SER)

KEYSCAN Scan Enable Register

offset address: 0x04

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
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SEN

classifier	marking	celebrity	functionality	fill out or in (information on a form)
for				
honorific				
people				
b31~b30	Reserved	-	Reads "0" and writes "0".	R
b0	SEN	Scan Enable Bit		
		0: Scanning disabled		R/W
		Bit		
		1: Scan Enable		

12.5.3 KEYS defense Scan Status Register (KEYSCAN_SSR)

KEYSCAN Scan Status Register

offset address: 0x08

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
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	INDEX[2:0]

classifier	marking	celebrity	functionality	fill out or in (information on a form)
for				
honorific				
people				
b31~b3	Reserved	-	Reads "0", writes "0".	R
b2~b0	INDEX[2:0]	ongoing index (math.)	Currently working SCAN pin index bit 000: Currently working SCAN pin is KEYOUT0 001: Currently working SCAN pin is KEYOUT1 010: The current working SCAN pin is KEYOUT2 011: Currently working SCAN pin is KEYOUT3 100: Currently working SCAN pin is KEYOUT4 101: Currently working SCAN pin is KEYOUT5 110: Currently working SCAN pin is KEYOUT6 111: Currently working SCAN pin is KEYOUT7	R

Note: The SSR.INDEX[2:0] bits are read-only registers, and only data read
when SER.SEN=1 is
have significance

13 Storage Protection Unit (MPU)

13.1 summary

The MPUs can provide protection of the memory and can improve the security of the system by blocking unauthorized access. Four host-specific MPU units and one IP-specific MPU unit are built into this chip.

module (in software)	element
ARM MPU	CPU's storage protection unit 8 regions, see ARM MPU description for details
System DMA_1 MPU: SMPU1	Storage protection unit for system DMA_1 16 areas, 8 areas dedicated to system DMA, 8 areas common to all DMAs
System DMA_2 MPU: SMPU2	Storage protection unit for system DMA_2 16 areas, 8 areas dedicated to system DMA, 8 areas common to all DMAs
USBFS-DMA MPU: FMPU	Storage Protection Unit for USBFS-DMA 8 areas, shared by all DMAs
IPMPU	Access protection unit for system IPs and security-related IPs

The ARM MPU provides the CPU with access control to the entire 4G address space.

SMPU1/SMPU2/FMPU provide read/write access control to all 4G address space for system DMA_1/system DMA_2/USBFS-DMA respectively. The MPU action can be set to Ignore/Bus Error/Non-maskable Interrupt/Reset when an access to the prohibited space occurs.

The IPMPU provides access control to system IPs and security-related IPs in unprivileged mode.

13.2 Functional Description

13.2.1 Regional Scope Setting

The MPU manages the storage space authority on a region basis. Each area can independently set the base address and area size, which can be set from 32Byte to 4GByte, and the size must be 2^n Byte ($n=5\sim32$) and the corresponding low n-bit of the base address is zero.

The address space that is not covered by any region is called the background region.

13.2.2 Permission settings

Each area, including the background area, can be set independently for each DMA to allow read/prohibit read and allow write/prohibit write. If an address overlap occurs between different areas, the set prohibition takes precedence.

13.2.3 MPU Action Selection

When a prohibited access occurs, the access is ignored (read access read to 0, write access ignored) while the corresponding action can be set, which can be set to:

- defy
- bus error
- unmaskable interrupt
- reset (a dislocated joint, an electronic device etc)

13.2.4 Starting the MPU

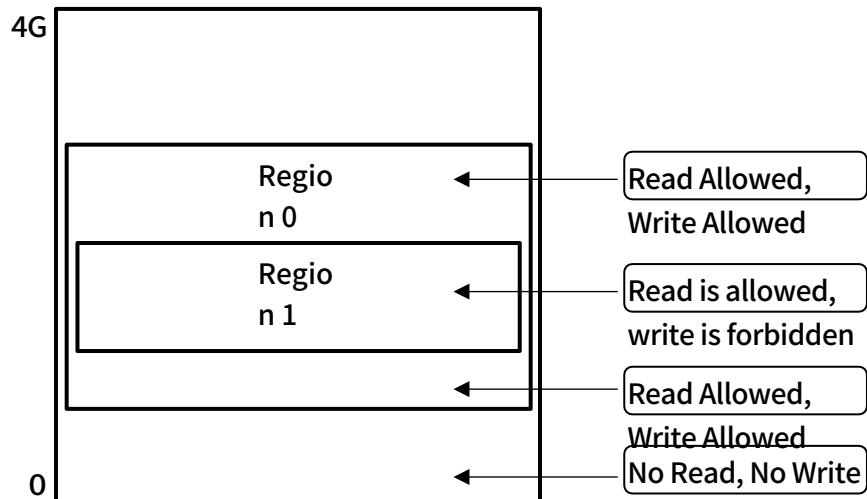
SMPU1/SMPU2/FMPU can be enabled independently.

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

13.3 Application Examples

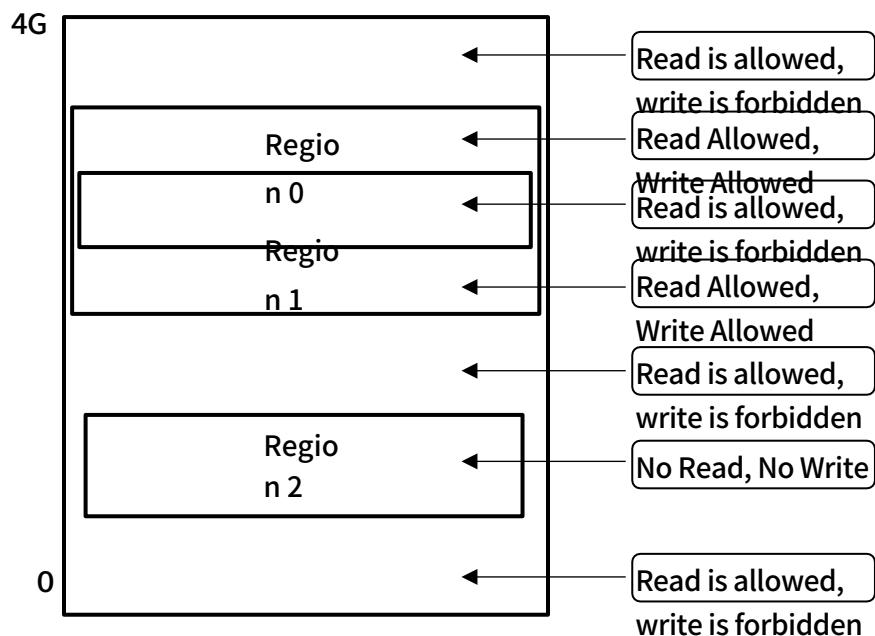
13.3.1 Allow only partial space access

Example: Set the background area permissions to No Read/No Write, area 0 to Allow Read/Allow Write, area 1 to Allow Read/No Write, and area 0 range over area 1.



13.3.2 Blocking access to only a portion of the space

Example: Set the background area permissions to Allow Read/No Write, area 0 is set to Allow Read/No Write, area 1 is set to Allow Read/No Write, area 0 overrides area 1, and area 2 is set to No Read/No Write.



13.4 Register Description

The registers of this module can only

be set by the CPU. MPU base

address: 0x4005_0000

offset address	register name	starting value	name (of a thing)	write-protected
+00 to +3C	MPU_RGD0~15	0x0000_0000	Area 0~15 Range Description Register	MPUWE
+40 to +7C	MPU_RGCR0~15	0x0000_0000	Area 0~15 control registers	MPUWE
+80	MPU_CR	0x0000_0000	MPU Control Register	MPUWE
+84	MPU_SR	0x0000_0000	MPU Status Register	not have
+88	MPU_ECLR	0x0000_0000	MPU Error Flag Clear Register	not have
+8C	MPU_WP	0x0000_0000	MPU Write Protect Register	WKEY

Base address: 0x40054000

offset address	register name	starting value	name (of a thing)	write-protected
+1C	MPU_IPPR	0x0000_0000	IP Access Protection Register	SYS

13.4.1 Area Range Description Register MPU_RGDr (n=0 to 15)

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
MPURGnADDR [31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
MPURGnADDR[15:5]										MPURGnSIZE[4:0]					

Bit Flag	Bit Name	Function	Read/Write
b31~b5	MPURGnADDR[31:5]	Area Base Address	Sets the base address of area n. The number of valid bits is related to the size of the area.
b4~b0	MPURGnSIZE[4:0]	Area Size	(MPURGnSIZE+1) bit fixed to 0 Setting the size of area n 00000~00011: Reserved, Setting prohibited 00100: 32 Byte 00101: 64 Byte11110: 2GByte 11111: 4GByte

13.4.2 Region Control Register MPU_RGCRn (n=0 to 15)

Reset value: 0x0000_0000

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

S1RGn E	-	-	-	-	-	S1RGn WP	S1RGn RP	S2RGn E	-	-	-	-	-	S2RGn WP	S2RGn RP	
b0																

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b24	reserved	-	Reserved bits, 0 for reading, 0 for writing	R
b23 disabled	FRGnE	FMPU region n enable	0: FMPU area n	R/W
1: Area n of FMPU is valid				
b22~b18	reserved	-reserved	Reserved bits, 0 for read, 0 for write	R
b17	FRGnWP	FMPU area n write permissions	0: Area n allows USBFS-DMA write	R/W
b16	FRGnRP	FMPU region n read access	1: Area n disables USBFS- DMA write	R/W
b15	S1RGnE	SMPU1 region n enables (usually used in the negative) have the possibility of	0: Area n allows USBFS-DMA 0: Area n of SMPU1 is not valid reads 1: Area n of SMPU1 is active 1: Area n disables USBFS- DMA reads	R/W
b14~b10	reserved	-	Reserved bits, 0 for reading, 0 for writing	R
b9	S1RGnWP	SMPU1 region n write permissions	0: Area n allows system DMA_1 write 1: Area n disables system	R/W
b8	S1RGnRP	SMPU1 region n read access	DMA_1 write	R/W
b7	S2RGnE	SMPU2 region n enables (usually used in the negative) have the possibility of	0: Area n allows system 0: Area n of SMPU2 is not valid DMA_1 to read 1: Area n of SMPU2 is active 1: Area n disables system DMA_1 reads	R/W
b6~b2	reserved	-	Reserved bits, 0 for reading, 0 for writing	R/W
b1	S2RGnWP	SMPU2 region n write permissions	0: Area n allows system DMA_2 write	R/W
b0	S2RGnRP	SMPU2 region n read access	1: Area n prohibits system DMA_2 write 0: Area n allows system DMA_2 to read 1: Area n disable system DMA_2 reads	R/W

-Control registers b31 to b16 in areas 8 to 15 are reserved.

Control registers b31 to b16 in areas 8 to 15 are reserved bits.

13.4.3 Control Register MPU_CR

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17
-	-	-	-	-	-	-	-	FMPUE	-	-	-	FMPUACT[1:0]	FMPUB WP	FMPUB RP
b16 b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1
SMPU 1E	-	-	-	SMPU1ACT[1. 0]	SMPU 1BWP	SMPU 1BRP	SMPU2E	-	-	-	SMPU2ACT [1. 0]	SMPU2 BWP	SMPU2 BRP	
			b0											

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b24	reserved	-	Reserved bits, 0 for reading, 0 for writing	R
b23	FMPUE	FMPU Enable	0: FMPU invalid 1: FMPU effective	R/W
b22~b20	reserved	-	Reserved bits, 0 for reading, 0 for writing	R
b19~b18 selection	FMPUACT[1:0]	FMPU action	read to 0, write access ignored 01: Read access reads 0, write access ignored; bus error generated 10: Read accesses read to 0, write accesses ignored; non-maskable interrupt generated 11: Reset	R/W
b17 Background Write Right	FMPUBWP	FMPU	0: FMPU background space allows USBFS-DMA write 1: FMPU background space disable USBFS-DMA write	R/W
b16 background reading rights	FMPUBRP	limit settings FMPU	0: FMPU background space allows USBFS-DMA reads 1: FMPU background space disable USBFS-DMA reads 0: SMPU1	R/W
b15 disabled	SMPU1E	SMPU1 enable	1: SMPU1 is valid	R/W
b14~b12	reserved	-reserved	Reserved bits, 0 for read, 0 for write	R
b11~b10 selection	SMPU1ACT[1:0]	SMPU1 action	Sets the action when a prohibited access occurs to system DMAC_1 00: Read access read to 0, write access ignored	
b8	SMPU1BRP			01: Read access reads 0, write access ignored; bus error generated
b9	SMPU2E SMPU1BWP	SMPU2 enable	0: SMPU2 not valid 1: SMPU2 effective	t setti ngs
b6~b4 Background Write Right	reserved	-	Reserved bits, 0 for reading, 0 for writing	10:R/W read to 0, write R accesses ignored;
b3~b2	SMPUACT[1:0]	SMPU Action Selection	limit settings Sets the action when a prohibited access occurs to system DMA_2. 00: read access reads 0, write access ignores it 01: Read access reads 0, write access ignored; bus error generated	non-maskable 408 /1012 R/W

11: Reset

0: SMPU1 background space allows system DMAC_1 write

1: SMPU1 background space prohibits system DMAC_1 write

R/W R/W

0: SMPU1 background space allows system DMAC_1 to read the

1: SMPU1 background space prohibits system DMAC_1 reads

Limit Settings 1: SMPU2 background space disable system DMA_2 reads

When multiple zone settings overlap, the priority is: set prohibited > set allowed.

13.4.4 Status flag register MPU_SR

Reset value: 0x0000_0000

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

-	-	-	-	-	-	-	SMPU1 EAF	-	-	-	-	-	-	-	SMPU2 EAF
b0															

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b17	reserved	-	Reserved bits, 0 for reading, 0 for writing	R
b16	FMPUEAF	FMPU error flag	0: USBFS-DMA no error access occurred 1: USBFS-DMA incorrect access occurred	R
b15~b9	reserved	-reserved	Reserved bits, 0 for reading, 0 for writing	R
b8	SMPU1EAF	SMPU1 error flag	0: No error access to system DMA_1 has occurred 1: An incorrect access has occurred to system DMA_1	R
b7~b1	reserved	-reserved	Reserved bits, 0 for reading, 0 for writing	R
b0	SMPU2EAF	SMPU2 error flag	0: No error access to system DMA_2 has occurred 1: An incorrect access has occurred to the system DMA_2	R

Write operations to this register are ignored; to clear the error flag, use MPUECLR.

13.4.5 Flag Clear Register MPU_ECLR

Reset value: 0x0000_0000

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

-	-	-	-	-	-	-	SMPU1 ECLR	-	-	-	-	-	-	-	SMPU2 ECLR
b0															

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b17	reserved	-	Reserved bits, 0 for reading, 0 for writing	R
b16 flag	FMPUECLR	FMPU error	Write 1 to clear FMPUEAF to 0	W
		removals		
b15~b9	reserved	-reserved	Reserved bits, 0 for reading, 0 for writing	R

b8
flag

removals

b7~b1	reserved	-reserved	Reserved bits, 0 for reading, 0 for writing	R
b0 flag	SMPU2ECLR	SMPU2 error	Write 1 clears SMPU2EAF to 0	W

removals

The readout value of this register is fixed to 0x0000_0000.

13.4.6 Write-Protect Register MPU_WP

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
WKEY[15:1]															MPUWE
classifier for honorific people	marking		celebrity		functionality						fill out or in (information on a form)				
b31~b16	reserved		-		Reserved bits, 0 for reading, 0 for writing						R				
b15~b1	WKEY[15:1]		write code		When writing to the MPUWE, you must also write to the WKEY 'b100101101010010, read as 0						W				
b0	MPUWE		MPU register write allow		0: MPU address registers/control registers are not allowed to be written to 1: MPU address register/control register write allowed						RW				

Write 0x96A5 to this register to set MPUWE to 1. Write 0x96A4 to clear MPUWE to 0. Write other value cannot

Change MPUWE.

13.4.7 IP Access Protection Register MPU_IPPR

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	
BUSER RE	-	MSTPW RP	MSTPR DP	SYSCW RP	SYSCR DP	INTCW RP	INTCR DP	SRAMC WRP	SRAMC RDP	DMPUW RP	DMPUR DP	RTCWR P	RTCRD P	BKSRA MWRP	BKSRA MRDP
b16 b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	
SWDTW RP	SWDTR DP	WDTWR P	WDTRD P	-	-	EFMCW RP	EFMCR DP	CRCWR P	CRCRD P	TRNGW RP	TRNGR DP	HASHW RP	HASHR DP	AESWR P	AESRD P
			b0												

classifier for honorific people	marking	Bit Name	Function	Read/Write
b31	BUSERRE	Bus Error Allowed	0: Ignore access to protected IP 1: A bus error is returned when an access to the protected IP occurs	RW
b30	-	-	Reserved bit, reads 0, writes 0 on write	R
b29	MSTPWRP	MSTP write protection	0: Write operation to registers PWC_FCG0/1/2/3, PWC_FCG0PC is permitted 1: Write operation to registers PWC_FCG0/1/2/3, PWC_FCG0PC is prohibited	RW
b28	MSTPRDP	MSTP read protection	0: Read operation allowed for registers PWC_FCG0/1/2/3, PWC_FCG0PC	RW
b27	SYSCWRP	SYSC write protection	0: operation to RMU/CMU/PWC allowed 1: Disable write operation to RMU/CMU/PWC	RW
b26	SYSCRDP	SYSC read protection	0: Read operation to RMU/CMU/PWC is allowed 1: Read operation to RMU/CMU/PWC prohibited	RW
b25	INTCWRP	INTC write protection	0: Write operation to INTC is allowed 1: Write operation to INTC prohibited	RW
b24	INTCRDP	INTC read protection	0: Read operation to INTC is allowed	RW
b23	SRAMCWRP	SRAMC write protection	0: Allow register write operation to [10. Built-in SRAM]	RW
b22	SRAMCRDP	SRAMC read protection	0: Allow register read operation of [10. Built-in SRAM]	RW
b21	DMPUWRP	DMPU write protection	0: Write operation to SMPU1/SMPU2/FMPU/IPMPU allowed 1: Write operation to SMPU1/SMPU2/FMPU/IPMPU is prohibited	RW
b20	DMPURDP	DMPU read protection	0: Read operation to SMPU1/SMPU2/FMPU/IPMPU allowed 1: Read operation to SMPU1/SMPU2/FMPU/IPMPU is prohibited	RW

b19	RTCWRP	RTC write protection operations to the RTC are allowed 1: Disable write operations to the RTC	0: Write RW
b18	RTCRDP	RTC read protection operations to the RTC are allowed 1: Disable read operation to RTC	0: Read RW
b17	BKSRAMWRP	BKSRAM write protection Write operation to Ret-SRAM is allowed 1: Write operation to Ret-SRAM is prohibited RW	0: RW
b16	BKSRAMRDP	BKSRAM read protection Read operation to Ret-SRAM is allowed 1: Read operation to Ret-SRAM is prohibited RW	0: RW
b15	SWDTWRP	SWDT write protection Write operation to SWDT is allowed 1: Write operation to SWDT prohibited RW	0: RW
b14	SWDTRDP	SWDT read protection operation of SWDT allowed 1: Disable read operation to SWDT	0: Read RW
b13	WDTWRP	WDT write protection 0: Write operations to the WDT are allowed	RW

			1: Disable write operations to the WDT	
b12	WDTRDP	WDT read protection	0: Read operation of the WDT is allowed 1: Disable read operation to WDT	RW
b11~b10	-	-	Reserved bit, reads 0, writes 0 on write	R
b9	EFMCWRP	EFM Write Protect	0: Allow register writes to [9 Embedded Flash (EFM) 1 : Disable register write operation to [9 E m b e d d e d Flash (EFM)].	RW
b8	EFMCRDP	EFM Read Protection register read operation for [9 Embedded Flash (EFM)]	0: Allow 1 : Disable register read operation of [9 E m b e d d e d Flash (EFM)].	RW
b7	CRCWRP	CRC write protection	CRC write operation is allowed 1: Write operation to CRC is prohibited RW	0:
b6	CRCRDP	CRC read protection	Read operation of CRC is allowed 1: CRC read operation is prohibited. RW	0:
b5	TRNGWRP	TRNG write protection	Write operation to TRNG is allowed 1: Write operation to TRNG is prohibited RW	0:
b4	TRNGRDP	TRNG read protection	operation to TRNG is allowed 1: Disable read operation to TRNG	0: Read RW
b3	HASHWRP	HASH write protection	operation to HASH is allowed 1: Disable write operations to HASH	0: Write RW
b2	HASHRDP	HASH read protection	Read operation of HASH is allowed 1: Read operation to HASH is prohibited RW	0:
b1	AESWRP	AES write protection	AES write operation is allowed 1: Write operation to AES is prohibited RW	0:
b0	AESRDP	AES read protection	AES read operation is allowed 1: Read operation to AES is prohibited RW	0:

Privileged mode is not affected by this register can read and write access object IP.

14DMA Controller (DMA)

14.1 summary

DMA is used to transfer data between memories and peripheral function modules, enabling data exchange between memories, between memories and peripheral function modules, and between peripheral function modules without CPU involvement.

- The DMA bus is independent of the CPU bus and is transmitted according to the AMBA AHB-Lite bus protocol.
- 2 DMA control units with a total of 8 independent channels for independent operation of different DMA transfer functions
- The start source for each channel is configured through a separate trigger source selection register
- Transmit one block of data per request
- The minimum data block is 1 data and the maximum is 1024 data.
- Each data width can be configured as 8bit, 16bit or 32bit.
- Up to 65535 transmissions can be configured
- Source and destination addresses can be independently configured as fixed, self-incrementing, self-decrementing, cyclic or jumps with specified offsets
- Three types of interrupts can be generated: block transfer completion interrupt, transfer completion interrupt, and transfer error interrupt. Each interrupt can be configured to be blocked or not. The block transfer completion and transfer completion can be used as event outputs and can be used as triggers for other peripheral modules.
- Support chain transmission function, can realize one request to transmit multiple data blocks
- Supports external event triggered channel reset
- Can be set to enter the module stop state when not in use to reduce power consumption

14.2 Module Schematic

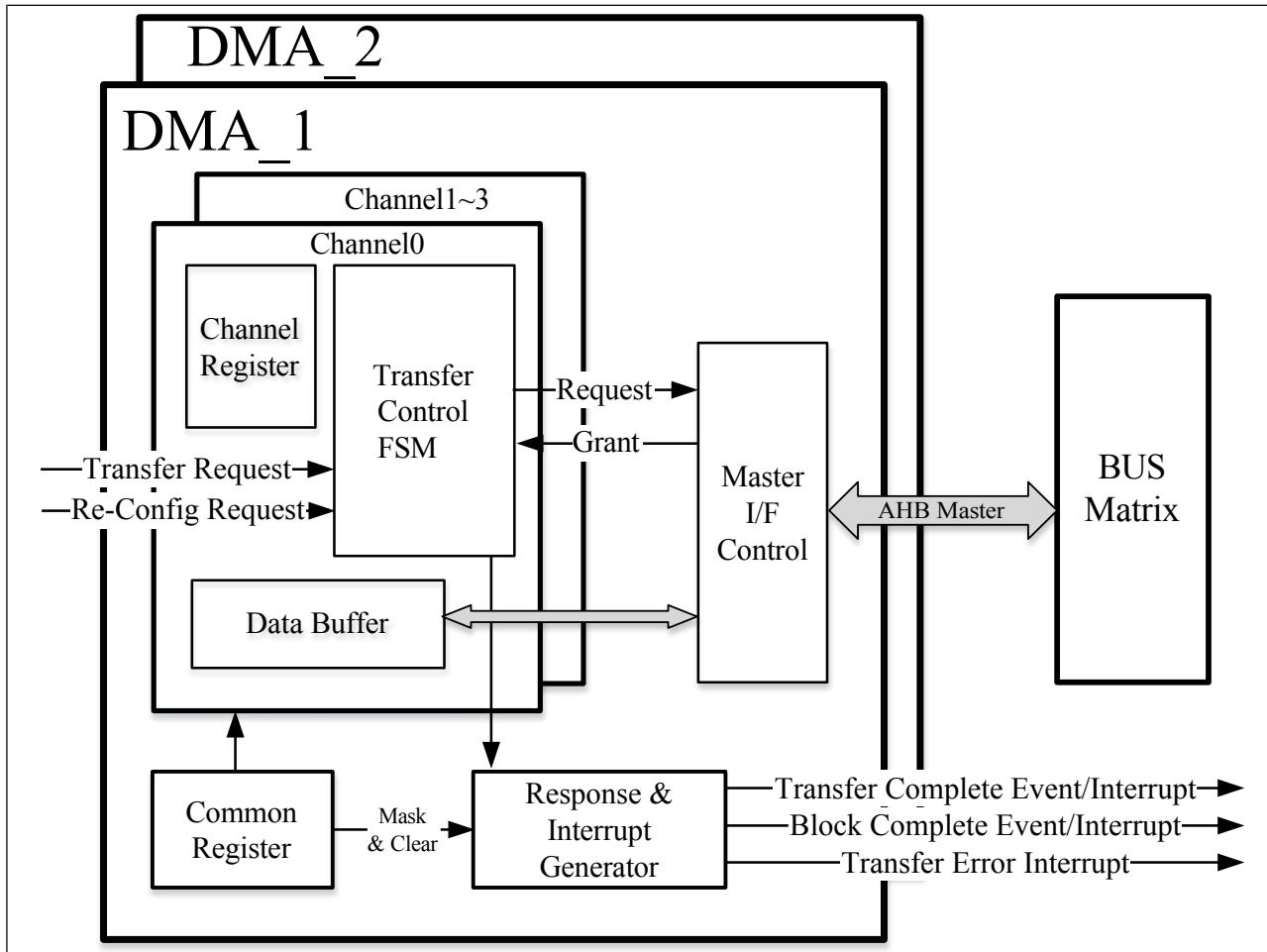


Figure 14-1 DMA Structure

14.3 Functional Description

14.3.1 Enable DMA Controller

To use DMA, you need to enable the DMA controller by writing the DMA_EN.EN bit in the DMA enable register.

When DMA is not used, or the chip needs to enter STOP mode, please set DMA_EN.EN to 0. Before writing EN to 0, please make sure register DMA_CHSTAT.DMAACT is 0, and make sure that DMA has completed all the transfers.

14.3.2 Channel Selection and Channel Prioritization

Each DMA control unit contains 4 channels, each of which can be independently configured for transmission. The order of priority of the 4 channels is as follows: Channel 0>Channel 1>Channel 2>Channel 3.

When a DMA unit has more than one channel requesting a transfer it will be executed in priority order. However, channels already in transmission will not be interrupted, and high priority channels will not start until the current channel has completed transmission.

14.3.3 Start DMA

The DMA is initiated by requests generated by peripheral circuits that are configured through the Trigger Source Selection Register DMA_TRGSELx (x=0~3), which configures the source of the startup request for channel x. The startup request can be configured through the DMA_EN.EN=1, and the DMA transfer enable is active DMA_EN. Channel x transmission is initiated when a start request is generated by the peripheral circuitry or a start request is generated by a software write register and the DMA transmission enable is active DMA_EN.EN=1 and the transmission channel is in the licensed state DMA_CHEN.CHEN[x]=1.

Before use, you need to enable the peripheral circuit trigger function and DMA function enable position of the function clock control 0 register (FCG0).

14.3.4 data block

The amount of data transferred per DMA startup is represented by a block, the size of which is set by the data control register DMA_DTCTLx.BLKSIZE, and the maximum number of data that can be set is 1024. The data width of each block is determined by DMA_CHxCTL.HIZE.

14.3.5 Transmit Address Control

The source and destination addresses for transmission can be set by registers to be fixed, incremental, decremental, reloaded, or discontinuous jumps.

Fixed: The source and destination addresses will be fixed during transmission.

Increment and Decrement: The source address and destination address will jump forward or backward according to the value of HSIZE after every 1 data is transmitted. For example, when HSIZE is 8bit, the address will be incremented/decremented by 1 each time, incremented/decremented by 2 each time when it is 16bit, and incremented/decremented by 4 each time when it is 32bit.

Reload: After transmitting the specified amount of data, the source and destination addresses will return to the initial address setting. The amount of data to be transmitted before address reloading, i.e. the size of the repeat area, is set by 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 skip, and the amount of data to be transmitted before the skip, i.e., the size of the discontinuity area, is set by register DMA_SNSEQCTL/DMA_DNSEQCTL. Address reloading is performed when the conditions for address reloading and discontinuous jumping are both met.

14.3.6 Number of transmissions

The total number of data blocks transferred by DMA is set by the CNT bit of the data control register DMA_DTCTLx. The maximum number of transfers can be set to 65535. The register value is reduced by 1 for each data block transferred. When the register value is reduced to 0, it means that all data transfers on this channel are completed, the channel transfer permission bit DMA_CHEN.CHEN[x] is cleared automatically, and a transfer completion interrupt is generated. If the register is set to 0 at the start of the transfer, it represents an unlimited transfer, with one block of data transferred per start request, but the channel transfer permission bit is not cleared and no transfer completion interrupt is generated.

14.3.7 Interrupt and event signal output

The DMA controller can generate the following 3 types of interrupts:

Data Block Completion Interrupt DMA_BTxC: Generated after completion of a data block transfer.

Transfer completion interrupt DMA_TCx: Generated after completion of the number of transfers set in register DMA_DTCTLx.CNT.

Transmission Error Interrupt DMA_ERR: An interrupt is generated when a start request overflows (i.e., the channel triggers a start request again when the channel's previous request has not yet been responded to) or, alternatively, when a bus error occurs during a transmission (e.g., an illegal or protected address is accessed), where a bus error immediately terminates this transmission.

All of the above interrupts, except for the startup request overflow error, can be set to be valid or invalid interrupts through the register DMA_CHxCTL.IE. In addition all interrupts are also equipped with a separate MASK register to mask the interrupts.

The above DMA_BTxC, DMA_TCx interrupts can also be used as event signal outputs, which can be used as trigger sources for other peripheral circuits. The event outputs are controlled by the MASK register but not by the interrupt license bit DMA_CHxCTL.IE.

14.3.8 chain transmission

The DMA controller has a chain transfer function. Chain transfer requires the configuration of the following eight registers totaling eight words, known as a descriptor (descriptor), contains the chain transfer of the source address, destination address, data control

information, address control information, chain pointer and transmission control information.

DMA_SARx

DMA_DARx

DMA_DTCTLx

DMA_RPTx

DMA_SNSEQCTLx

DMA_DNSEQCTLx

DMA_LLPx

DMA_CHxCTL

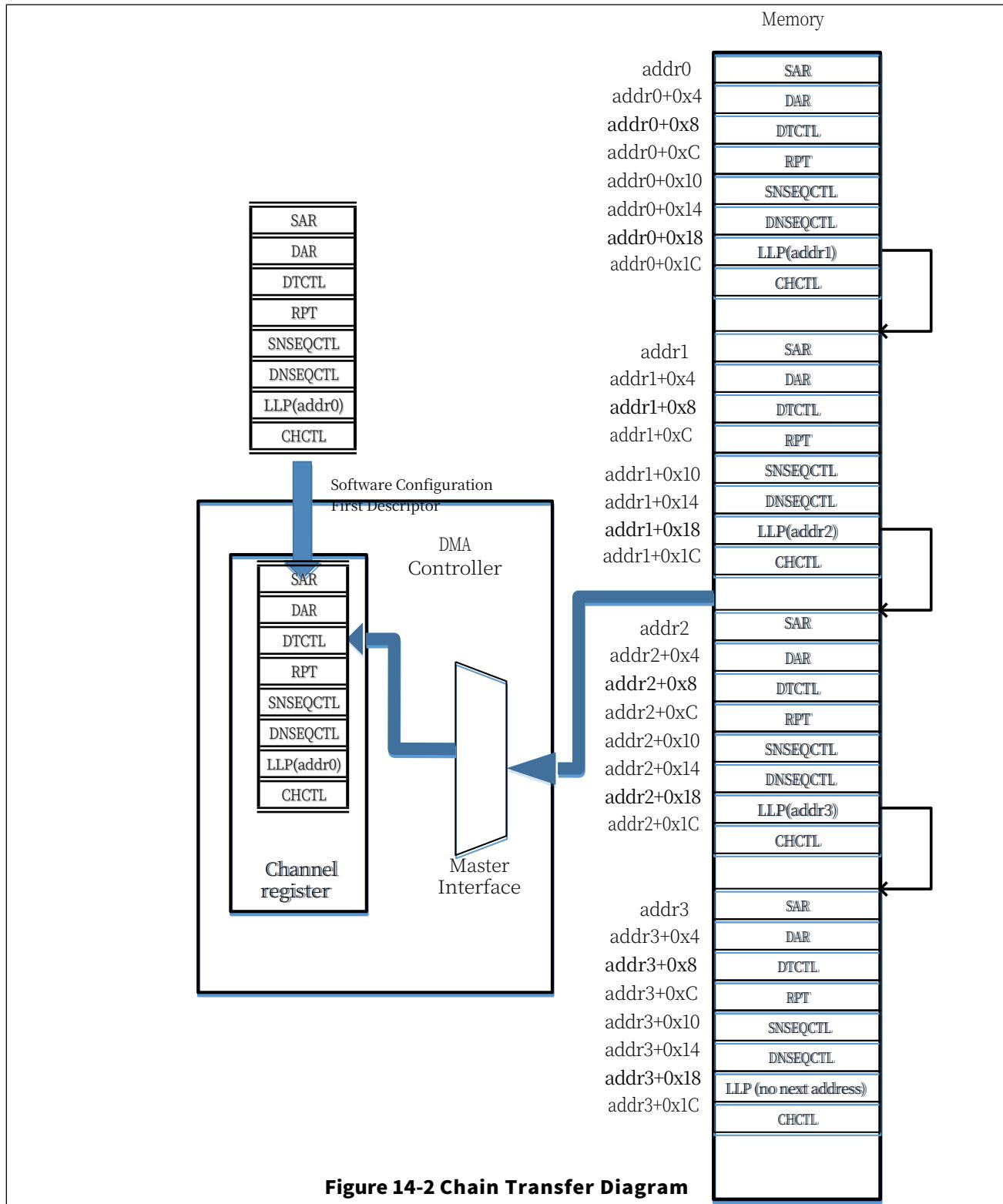
The LLP is known as the **Linked-List Pointer**, where the value represents the first address of the next descriptor in memory. When chained transfers are used, first the LL PEN of the channel control register DMA_CHxCTLx is written to enable the chained transfer, and the descriptor information of the first transfer is written to the corresponding register. The descriptors of the subsequent transfers are then sequentially initialized in memory. When it is necessary to end the chain transfer, set LL PEN 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 transmission of a descriptor ends:

If LL PEN=1, LL PRUN=0, BTC and TC interrupts are generated according to the configuration of the interrupt license, and the channel license CHEN[x] is not automatically cleared to 0. The next descriptor specified by LLP is loaded into the Channel Configuration Register from memory, and waits for the next transfer request input to begin the first transfer of the new descriptor.

If LL PEN=1 and LL PRUN=1, BTC and TC interrupts are not generated and the channel license CHEN[x] is not automatically cleared to 0. The next descriptor specified by LLP begins the first transfer of the new descriptor directly after it has been loaded from memory into the channel configuration register.

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



14.3.9 Discontinuous Address Transmission

The use of Discontinuous Address Transfer enables the source and destination addresses to jump by a certain offset after a certain amount of data has been transferred. The direction of the jump is forward or backward according to the settings of DMA_CHxCTL.SINC and DMA_CHxCTL.DINC. To use, first set the channel control registers DMA_CHxCTL.SNSEQEN and DMA_CHxCTL.DNSEQEN to 1 to enable discontinuous address transfers as required. Then, configure the source and target device discontinuous address transfer control registers DMA_SNSEQCTLx and DMA_DNSEQCTLx. The transfer process proceeds as shown in the following figure.

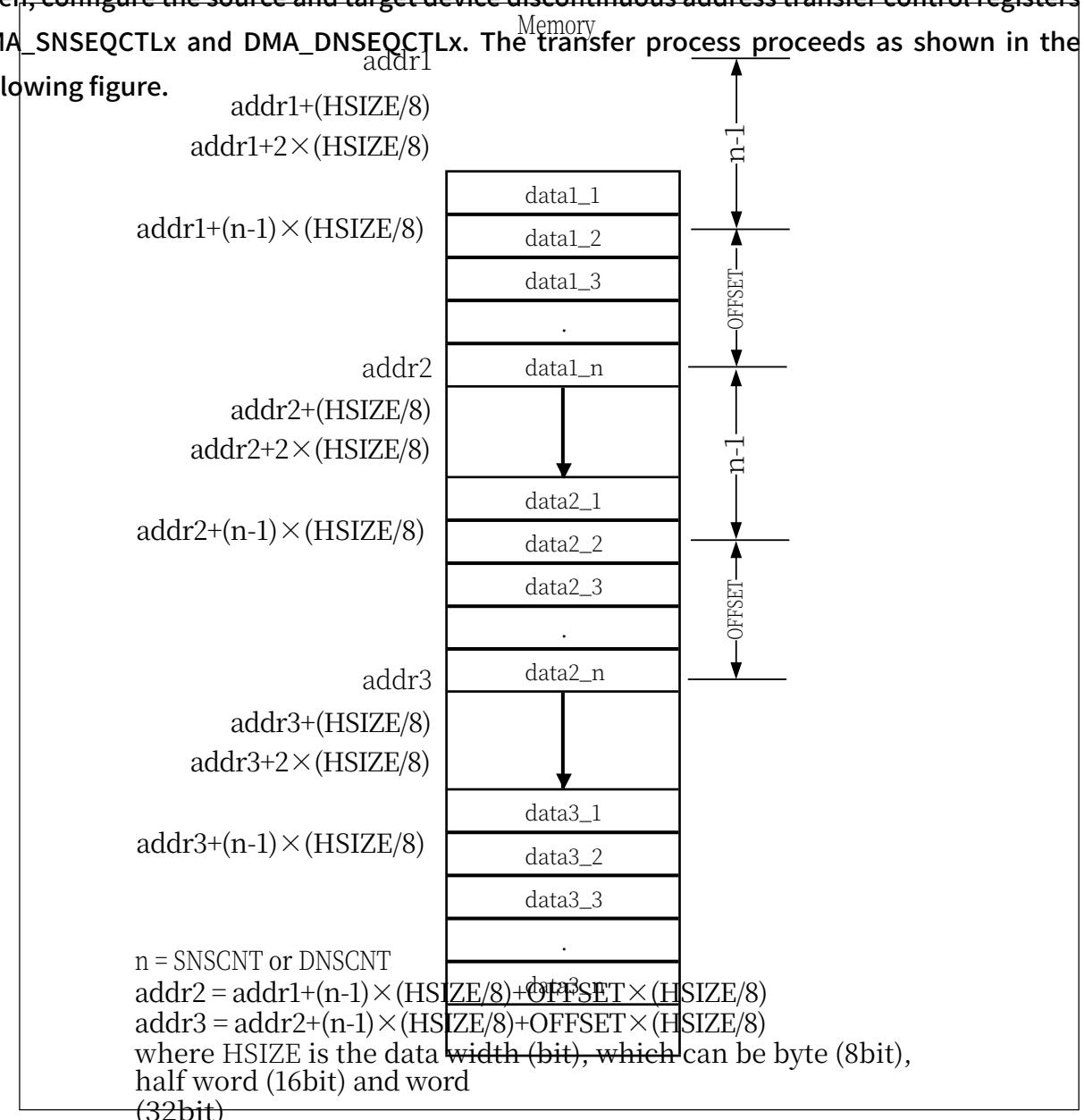


Figure 14-3 Discontinuous Address Transmission Schematic

14.3.10 Channel Reset

The channel reset function refers to the modification of the channel's internal status registers to reconfigure the next data transfer via an event request from the peripheral circuit. Setting register DMA_RCFGCTL.RCFGGEN to 1 allows channel reset. The reset request source is selected through the trigger source selection register DMA_RC_TRGSEL. When the selected reset request source is input, the channel selected by register DMA_RCFGCTL.RCFGCHS is updated as specified. The reset request only updates the internal state and does not initiate an actual data transfer.

Channels can be reset in the following three ways: chain pointer, discontinuous, and repetitive.

When chain pointer reset is selected, the channel's descriptor and internal state are all updated to the new descriptor pointed to by chain pointer LLP. Subsequent transmission requests are transmitted with the new descriptor.

When Discontinuous, Heavy Duty Reset is selected, the internal state of the channel is updated as described in the table below.

Table 14-1 Channel Reset Description

Channel internal status	Reset method	
	discontinuous	heavy duty
Remaining transmission counter	Updated to the value after the next address discontinuity jump occurs in normal state	Updated to the normal state, after the next reload occurs
Source/destination address for next transmission	Updated to the first address of the next discontinuous transmission area	Updated to the initial setting of register DMA_SARx/DARx

Attention:

- When the reset function is active, the channel uses registers DMA_RPTBx and DMA_SNSEQCTLBx, DMA_DNSEQCTLBx to control the reloading and discontinuous jumping of the transmit address. Registers DMA_RPTx and DMA_SNSEQCTLx, DMA_DNSEQCTLx are invalid.

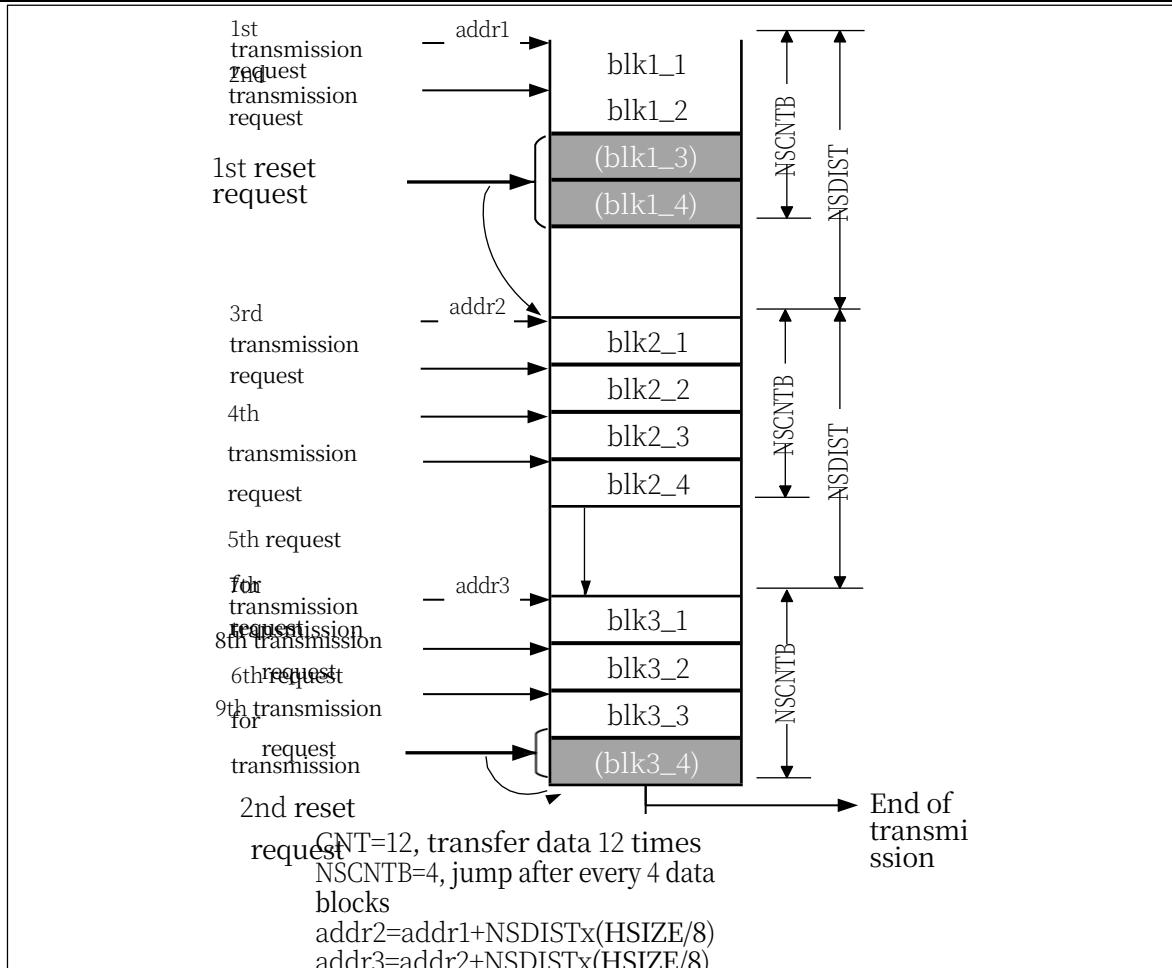


Figure 14-4 Discontinuous Reset Schematic

In the DMA action shown in Figure 14-4, each transfer request initiates the transfer of one data block. After the 1st reset request occurs, the controller skips the data blocks **blk1_3,blk1_4**, and the transfer address is updated to the first address of the next discontinuous region, i.e., **addr2**. After the 2nd reset request occurs, the number of remaining transfers is updated to 0, i.e., all the data transfers are completed, and the channel license bit is automatically cleared to 0, generating the transfer completion interrupt and event.

14.3.11 premature termination of transmission

CHENx remains active during transmission, and is automatically invalidated upon completion of the number of transfers set in DMA_DTCTLx for non-chained transfers, and upon completion of the number of transfers for the last chained transfer for chained transfers. If the software writes DMA_CHEN.CHENx to 0 during transmission, the DMA will terminate the transmission after completing the current data read/write.

Attention:

- When software terminates a transfer early by writing 0 to the CHENx bit, the DMA does not internally save the transfer state at the time of termination. When the channel is allowed again by writing CHENx to 1 without resetting the channel configuration register (descriptor) state, the DMA will re-transmit the terminated block of data after the transfer request is entered, rather than continue the transfer at the end of the period.

14.4 Application Examples

14.4.1 Memory-to-memory transfer

Objective: To transfer 22 data from RAM address 0x2000_0000 to 0x2000_1000 with a data width of 32bit.

1. Register Setting

- DMA_EN.EN Write 1 to enable the DMA controller.
- Select a channel, e.g. channel 0, and configure the channel registers to enable it:
 - Write DMA_SAR0 Configure source address to SRAM area 0x2000_0000
 - Write DMA_DAR0 Configure source address to SRAM area 0x2000_1000
 - Write DMA_DTCTL0 to configure the size of the data block to be 4 and the number of transfers to be 3, generating a block transfer completion interrupt after each transfer of 1 data block, and generating a transfer completion interrupt after 3 transfers are complete
 - Write the DMA_RPT register to configure the source address repeat region size to 6, i.e., reload the initial source address after 6 addresses have been transferred.
 - Configure the channel control register DMA_CH0CTL to do so:
 - * Source and Destination Address Chain Transmission Invalid
 - * The source address overload is valid, and the destination address is updated in a self-incrementing manner.
 - * Data width is word (32bit)
 - * interrupt enable active
 - Channel enable bit DMA_CHEN.CHENO Write 1 to enable channel 0
 - Configure the trigger source controller, DMA_TRGSEL0, to select a software trigger as the startup request for DMA channel 0
- Write peripheral event software trigger register INTSFTTRG STRG to 1, send the first software start request, DMA start transferring data

2. transmission process

Since the block size is 4, when the software writes INTSFTTRG STRG to 1 to start the first transmission, when a block is completed, the transmission count DMA_DTCTL0.CNT will be reduced by 1 and a block completion interrupt will be generated so that the software can continue to write INTSFTTRG STRG in the interrupt subroutine to start the second transmission. In the second transfer, since the source address duplication area size is set to 6, the source address will be reloaded to the initial address 0x20000000 after 2 addresses have been transferred and continue to transfer the remaining 2 addresses. After the second

transfer is completed, the transfer count DMA_DTCTL0.CNT is decremented by 1 and a block transfer completion interrupt is generated so that the software can continue to write INTSFTTRG.STRG in the interrupt subroutine to start the third transfer. After the third transmission is completed, the transmission count DMA_DTCTL0.CNT is reduced to 0, i.e., this transmission is fully completed, the DMA generates a block transmission completion interrupt and a transmission completion interrupt, and at the same time, the channel enable bit DMA_CHEN.CHEN0 will be cleared to zero automatically.

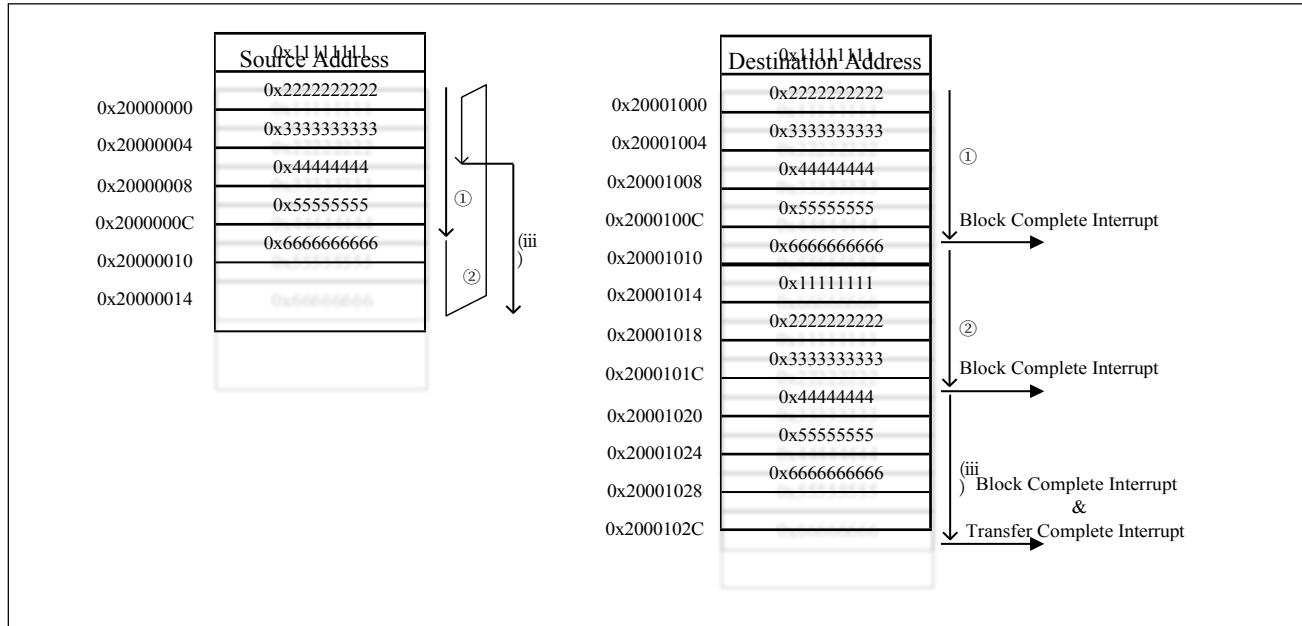


Figure 14-5 Application Example 1: Memory to Memory Transfer

14.4.2 Memory to Peripheral Circuit Transfer

Objective: To transmit 10 half-words of data from RAM address 0x2000_0000 to the transmit buffer register of the communication module, and the communication module generates a transmit request for each data sent. The communication module generates a transfer request after each data is sent. When the last data is sent, the DMA generates a transfer completion interrupt.

1. Register Setting

- DMA_EN.EN Write 1 to enable the DMA controller.
- Configure the DMA_INTMSK register to block the block transfer completion interrupt and enable the transfer completion interrupt
- Select a channel to configure the channel registers, e.g. select channel 0
 - Write DMA_SAR0 Configure source address to SRAM area 0x2000_0000
 - Write DMA_DAR0 Configure the source address to be the register address of the peripheral circuit 0x4000_0000
 - Write DMA_DTCTL0 Configure the data block size to be 1 and the number of transfers to be 10, with 1 data transferred once per transfer request.
 - Configure the channel control register DMA_CH0CTL to do so:
 - * Source and Destination Address Chain Transmission Invalid
 - * The source address is updated in a self-incrementing manner and the destination address is fixed
 - * Source and destination address data width is half a word (16bit)
 - * interrupt enable active

- Configure the trigger source controller, DMA_TRGSEL0, to select the communication module's transmit register empty event as the start request for DMA channel 0
- Channel enable bit DMA_CHEN.CHEN0 Write 1 to enable channel 0

2. transmission process

After the channel is enabled, the DMA waits for a transmission request from the communication module. When a transfer request is generated, the DMA transfers the data in RAM to the transmit buffer register of the communication module and waits for the second transfer request from the communication module. The DMA does not generate an interrupt request at this time due to block transfer completion interrupt blocking.

When all 10 data have been transferred, the DMA generates a transfer enable bit DMA_CHEN.CHEN0 is automatically cleared to zero.

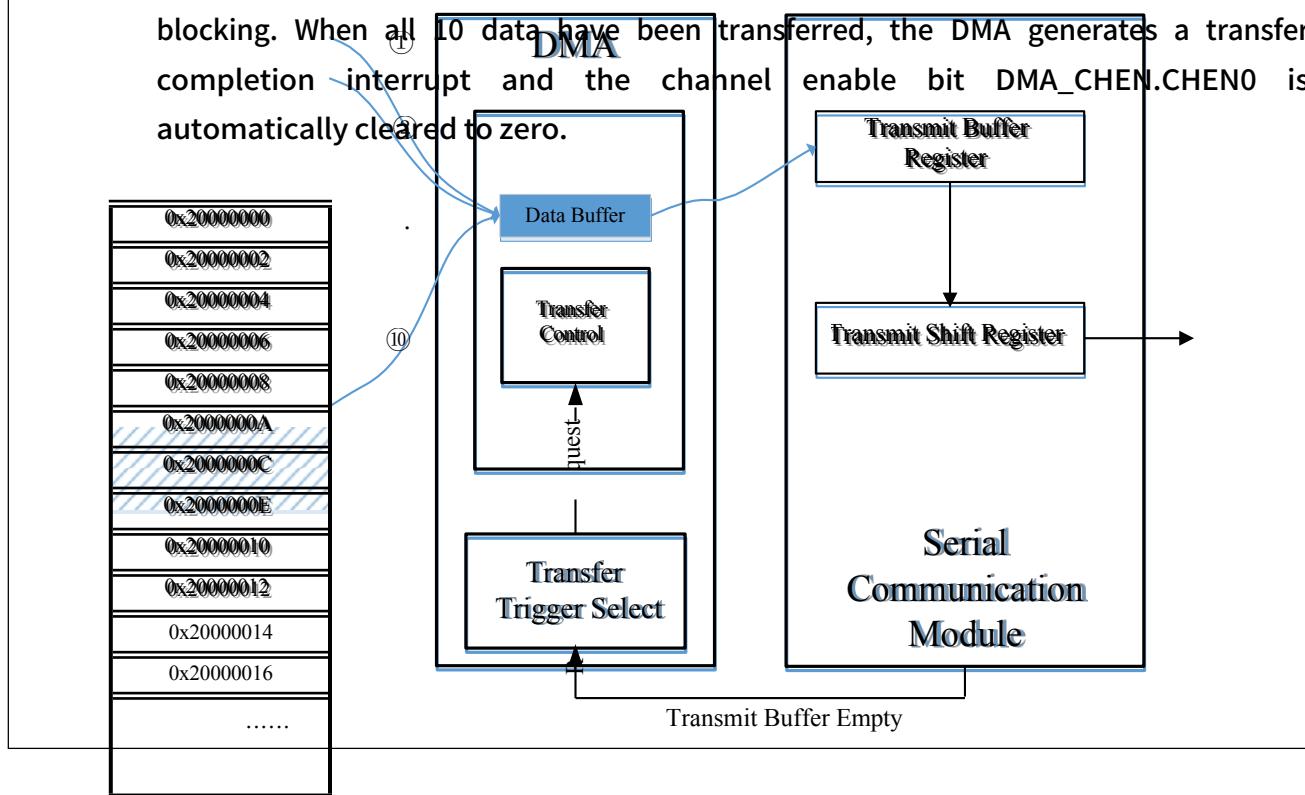


Figure 14-6 Application Example 2: Memory to Peripheral Circuit Transfer

14.4.3 Memory-to-memory chain transfer

1. Register Setting

- DMA_EN.EN Write 1 to enable the DMA controller.
- Select a channel, configure the channel registers, e.g. select channel 0 & configure the descriptor for the first transmission (descriptor0)
 - Write DMA_SAR0 Configure source address to SRAM area 0x2000_0000
 - Write DMA_DAR0 Configure source address to SRAM area 0x2000_1000
 - Write DMA_DTCTL0 to configure the data block size to 10
 - Write the second descriptor (descriptor1) to address 0x20002000 in the chain pointer register DMA_LLPO.
 - Configure the channel control register, DMA_CH0CTL, to configure the transfer parameters for the first data block to be implemented:
 - * Chain transmission is effective
 - * Chained transmission mode for direct initiation of the next transmission
 - * Source and destination addresses are updated in a self-incrementing manner
 - * Data width in words (32bit)
 - * Invalid interrupt enable
- Configure the descriptor (descriptor1) for the second transmission in the 0x2000_2000 address in RAM space, including:
 - Write 32-bit data 0x2000_0100 in 0x2000_2000, which is the source address of the second transfer.
 - Write 32-bit data 0x2000_1100 in 0x2000_2004, which is the destination address of the second transfer.
 - Configure the data block size in 0x2000_2008 to 20
 - Write 32-bit data 0x2000_2100 to 0x2000_2018, which is the address of the descriptor2 for the third transfer.
 - 0x2000_021C Write the control data for the second transmission which is realized in 0x2000_021C:
 - * Chain transmission is effective
 - * Chained transmission mode for direct initiation of the next transmission
 - * Source and destination addresses are updated in a self-incrementing manner
 - * Data width is half word (16bit)
 - * Invalid interrupt enable
- Configure the descriptor (descriptor2) for the third transmission in the 0x2000_2100 address in RAM space, including:
 - Write 32-bit data 0x2000_0200 in 0x2000_2100, which is the source address of the

third transmission.

- Write 32-bit data 0x2000_1200 in 0x2000_2104, which is the destination address of the third transfer.
- 0x2000_2108 Configure the data block size to 40
 - The 32-bit data 0x0 is written in 0x2000_2118, which means that this transmission is the last transmission of a chained transmission
 - Write the control data realization for the third transmission in 0x2000_211C:
 - * Invalid chain transmission

- * Source and destination addresses are updated in a self-incrementing manner
 - * Data width in bytes (8bit)
 - * interrupt enable active
 - Channel enable bit DMA_CHEN.CHEN0 Write 1 to enable channel 0
 - Configure the transmit startup trigger source selection register DMA_TRGSEL0 to select a software trigger as the startup request for DMA channel 0
 - Write software trigger register INTSFTTRG STRG to 1 to send a start request and the DMA starts transmitting data
2. transmission process

The software initiates the DMA to start the transfer. After the first transfer is completed, the DMA reads the descriptor (descriptor1) for the second transfer into the channel register, since the chain transfer mode is set to directly start the next transfer and the interrupt is disabled. The second transfer is started directly according to the parameters configured in the descriptor. After the second transfer is completed, the descriptor for the third transfer (descriptor2) is read into the channel register. The third transmission is started according to the parameters configured in the descriptor. Upon completion of the third transfer, the DMA generates a transfer completion interrupt and clears the channel enable bit, DMA_CHEN.CHEN0, since this is the last transfer in the chain according to the configuration information and since the interrupt enable is active.

14.5 Register Description

DMA_1 BASE_ADDR:0x4005_3000

DMA_2 BASE_ADDR:0x4005_3400

register name	notation	offset address	bit width	reset value
DMA Enable Register	DMA_EN	0x00	32	0x0000_0000
Interrupt status register 0	DMA_INTSTAT0	0x04	32	0x0000_0000
Interrupt status register 1	DMA_INTSTAT1	0x08	32	0x0000_0000
Interrupt Mask Register 0	DMA_INTMASK0	0x0C	32	0x0000_0000
Interrupt Mask Register 1	DMA_INTMASK1	0x10	32	0x0000_0000
Interrupt reset register 0	DMA_INTCLR0	0x14	32	0x0000_0000
Interrupt reset register 1	DMA_INTCLR1	0x18	32	0x0000_0000
Channel Enable Register	DMA_CHEN	0x1C	32	0x0000_0000
Transmission Request Status Register	DMA_REQSTAT	0x20	32	0x0000_0000
In-transit channel monitor register	DMA_CHSTAT	0x24	32	0x0000_0000
Channel Reset Control Register	DMA_RCFGCTL	0x2c	32	0x0000_0000
Transmit Source Address Register	DMA_SARx *1	0x40+0x40*x	32	0x0000_0000
Transmit Destination Address Register	DMA_DARx	0x44+0x40*x	32	0x0000_0000
Data Control Register	DMA_DTCTLx	0x48+0x40*x	32	0x0000_0001
Repeat Area Size Register	DMA_RPTx	0x4C+0x40*x	32	0x0000_0000
Repeat Area Size Register B	DMA_RPTBx			
Source Device Discontinuous Address Transfer Control Register	DMA_SNSEQCTLx	0x50+0x40*x	32	0x0000_0000
Source Device Discontinuous Address Transfer Control Register B	DMA_SNSEQCTLBx			
Target Device Discontinuous Address Transfer Control Register	DMA_DNSEQCTLx	0x54+0x40*x	32	0x0000_0000
Target Device Discontinuous Address Transfer Control Register B	DMA_DNSEQCTBx			
Chain Pointer Register	DMA_LLPx	0x58+0x40*x	32	0x0000_0000
Channel Control Register	DMA_CHxCTL	0x5C+0x40*x	32	0x0000_1000
Transmit Source Address Monitor Register	DMA_MONSARx	0x60+0x40*x	32	0x0000_0000
Transmit Destination Address Monitor Register	DMA_MONDARx	0x64+0x40*x	32	0x0000_0000
Data Control Monitor Register	DMA_MONDTCTLx	0x68+0x40*x	32	0x0000_0001
Repeat Area Counter Monitor Register	DMA_MONRPTx	0x6C+0x40*x	32	0x0000_0000
Source Device Discontinuous Transmission Counter Monitor Register	DMA_MONSNSEQCTLx	0x70+0x40*x	32	0x0000_0000
Target Device Discontinuous	DMA_MONDNSEQCTLx	0x74+0x40*x	32	0x0000_0000

Transmission Counter Monitor Register				
--	--	--	--	--

14.5.1 DMA Enable Register (DMA_EN)

Reset value: 0x0000_0000

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

b0

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31-b1	Reserved	-	Reads "0", writes "0".	R/W
b0	EN	DMA enable bit	0: DMA invalid 1: DMA enable	R/W

14.5.2 Interrupt status register 0 (DMA_INTSTAT0)

Reset value: 0x0000_0000

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

b0

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31-b20	Reserved	-	Reads "0", writes "0".	R/W
b19-b16	REQERR[3:0]	Transmission request overflow error interrupt	0: No transmission request overflow error occurred on this channel 1: A transmission request overflow error has occurred on this channel, i.e., the last request is still in the wait state	
b15-b4	Reserved	-Reserved	The last request is still in the waiting state.	
b3-b0	TRNERR[3:0] error occurred on this channel	Transmit error interrupt bit R	Read "0", write "0". 1: A transmission error has occurred on this channel	R/W 0: No transmission

14.5.3 Interrupt status register 0 (DMA_INTSTAT1)

Reset value: 0x0000_0000

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

b0

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31-b20	Reserved	-	Reads "0", writes "0".	R/W
b19-b16	BTC[3:0]	Block transfer completion interrupt bit	This interrupt occurs after the completion of a block of data transfer 0: No block transfer interrupt occurs on this channel 1: A block transfer interruption occurs on this channel	R
b15-b4	Reserved	-	Reads "0", writes "0".	R/W
b3-b0	TC[3:0]	Transmission completion interrupt bit	This interrupt occurs after completion of the number of transfers set by the transfer count register DMA_CNTx 0: No transmission completion interrupt occurs on this channel 1: A transmission completion interrupt occurs on this channel	R

14.5.4 Interrupt Mask Register (DMA_INTMASK0)

Reset value: 0x0000_0000

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

b0

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31-b20	Reserved	-	Reads "0" and writes "0".	R/W
b19-b16	MSKREQERR[3:0] mask transmission request overflow interrupt	Transmission request overflow interrupt R/W block (sth. or sb)	0: Do not 1: Mask transmission request overflow interrupt	
b15-b4	Reserved	-	Reads "0", writes "0".	R/W

0: Do not mask transmission error interrupt

1: Masked transmission error interrupt

14.5.5 Interrupt Mask Register (DMA_INTMASK1)

Reset value: 0x0000_0000

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

b0

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31-b20	Reserved	-	Reads "0", writes "0".	R/W
b19-b16	MSKBTC[3:0]	Block transfer completion interrupt masking Block transfer completion interrupt not masked	0: 1: Masked block transfer completion interrupt	R/W
b15-b4	Reserved	-Reserved	Read "0", write "0".	R/W
b3-b0	MSKTC[3:0]	Transmission completion interrupt masking 0: do not mask transmission completion interrupt	1: Masked transmission completion interrupt	R/W

14.5.6 Interrupt Reset Register (DMA_INTCLR0)

Reset value: 0x0000_0000

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

b0

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31-b20	Reserved	-	Reads "0", writes "0".	R/W
b19-b16	CLRREQERR[3:0]	Transmission request overflow error no effect, write 1 resets the transmission request overflow error interrupt status bit	Write 0 has W	Write 0 has W
		interrupt reset (computing)	Readout is always 0	
b15-b4	Reserved	-	Reads "0", writes "0".	R/W
b3-b0	CLRTRNERR[3:0]	Transmission error interrupt reset effect, writing 1 resets the transmission error interrupt status bit	Writing 0 has no W	Writing 0 has no W
			Readout is always 0	

14.5.7 Interrupt Reset Register (DMA_INTCLR1)

Reset value: 0x0000_0000

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

b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CLRTC[3:0]

b0

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31-b20	Reserved	-	Reads "0", writes "0".	R/W
b19-b16	CLRBTC[3:0]	Block transfer completion interrupt reset effect, write 1 to reset block transfer completion interrupt status bit	Write 0 has no W	Readout is always 0
b15-b4	Reserved	-Reserved	Read "0", write "0".	R/W
b3-b0	CLRTC[3:0]	Transmission completion interrupt reset effect, writing 1 resets the transmission completion interrupt status bit	Writing 0 has no W	Readout is always 0

14.5.8 Channel Enable Register (DMA_CHEN)

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CHEN[3:0]

b0

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31-b4	Reserved	-	Reads "0", writes "0".	R/W
b3-b0	CHEN[3:0]	Channel Enable Bit	Each bit corresponds to one channel. To use the channel, this bit needs to be set to 1. The enable bit is held to 1 during transmission and will be cleared automatically when the number of transmissions set by the transmission count register DMA_DTCTLx.CNT has been transmitted. If DMA_DTCTLx.CNT is set to 0, the It will not be automatically cleared after the transmission is completed, i.e. unlimited transmissions.	R/W

0: The channel is invalid
1: The channel is valid

14.5.9 Channel reset control register (DMA_RCFGCTL)

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	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]		-	-	-	-	-	-	RCF GLL P	RCF GEN	

classifier for marking honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31-b22	Reserved	-	Reads "0", writes "0".	R/W
b21-b20	CNTMD[1:0] transmission count	Remaining processor reset mode	00: remain unchanged 01: By source address method 10, 11: By Destination Address When the source address method is selected and the source address selects Discontinuous Reset, the remaining transmission count counter is updated to the state after the number of transmissions specified by DMA_SNSEQCTLBx.SNSCNTB; when the source address selects Repeat, the remaining transmission count counter is updated to the state after the number of transmissions specified by <u>DMA_RPTBx_SRPTB</u> . The remaining transmission count counter	T he sou rce ad dre ss
b19-b18	DARMD[1:0] Address Reset Method	Destination	also remains unchanged when source address selection is held. When the destination address method is selected, it is similar to the source address method. 00: remain unchanged 01: Discontinuous reset The destination address for the next transmission is updated to addr_base + (DNSDIST x HSIZE(bit)/8) where: addr_base indicates the first address of the current discontinuous transmission area 10, 11: Repetitive reset <u>The destination address for the next transmission is</u>	for the nex t tra ns mis sio n is up dat ed
b17-b16	SARMD[1:0] Reset Method	Source Address	updated to the value initially set in the DMA_DARx register. NOTE: Setting DARMD[1:0] to 01 is only allowed in the This Channel Destination Address Not Continuously Transmitted Valid (DMA_CHxCTL.DNSEQEN=1) state. Setting DARMD[1] to 1 is only allowed in the This Channel Destination Address Reload Valid (DMA_CHxCTL.DRPTEN=1) state. 00: remain unchanged 01: Discontinuous reset The source address for the next transmission is updated to addr_base + (SNSDIST x HSIZE(bit)/8) where: addr_base indicates the first address of the current discontinuous transmission area 10, 11: Repetitive reset	to the val ue init iall y set in the DM A_S ARx reg

source address

(DMA_CHxCTL.SNSEQEN=1) state sets

SARMD[1:0] to 01.

Setting SARMD[1] to 1 is only allowed in the This
Channel Source Address Reload Valid

(DMA_CHxCTL.SRPTEEN=1) state.

R/W

R/W

b15-b12	Reserved	-0" for reading, "0" for writing. "0", writes "0". R/W	Reads
b11-b8	RCFGCHS[3:0]	Reset channel selection Channel 0 R/W 0x1: Channel 1	0x0:

		0x2: Channel 2 0x3: Channel 3 Others: Reserved, set to disable	
b7-b2	Reserved	-0" for reading, "0" for writing. "0", writes "0". R/W	Reads
b1	RCFGLLP	Chain Pointer Channel Reset pointer reset active R/W Note: When RCFGPLL is set to 1, the channel will reload the new descriptor in memory, so this sends the The bit16-bit21 of the memory are all invalid.	1: Chain
b0	RCFGEN	Channel Reset License Disable event-triggered forced update of channel configuration registers R/W registers	0: 1: Allow event-triggered forced update of channel configuration

Attention:

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

14.5.10 Transmission request status register (DMA_REQSTAT)

Reset value: 0x0000_0000

b31	b 30	b29	b28 2	b 26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	7			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	CHREQ[3:0]		
<hr/>																		
classifier for honorific people	marking		celebrity		functionality											fill out or in (information on a form)		
b31-b17	Reserved		-		Reads "0" and writes "0".											R/W		
b16	RCFGREQ		Channel reset request flag		Set to 1 when an external reset request is input. Clear 0 when channel reset is initiated, or when channel reset is disabled.											R		
b15-b4	Reserved		-Reserved		0: No channel reset request 1: There is a channel reset request											R/W		
b3~b0 transfer request marker	CHREQ[3:0]		Channel aspiration		Read "0", write "0". Each bit corresponds to one channel. Corresponding to position 1 when an external transmission request is input. When this channel transmission is initiated, or a transmission error occurs, or the transmission transmission license bit is transmitted (DMAEN or CHEN[x]) This bit is cleared to 0 when 0 is written. When this bit is in the 1 state and a transmission request is entered again for this channel, a transmission request overflow occurs error, the second request is ignored. 0: No transmission request for this channel 1: There is a transmission request for this channel													R

14.5.11 Channel status observation register (DMA_CHSTAT)

Reset value: 0x0000_0000

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

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31-b20	Reserved	-	Reads "0" and writes "0".	R/W
b19-b16	CHACT[3:0]	Channel Monitor Bit in Transmission Motion	Each bit corresponds to a channel. 0: the channel is idle 1: The channel is in action	R
b15-b2	Reserved	-	Reads "0" and writes "0".	R/W
b1	RCFGACT	Monitor bit in DMA channel reset action	0: DMA is not in channel reset action 1: DMA is in channel reset action	R
b0	DMAACT	DMA action monitor bit	0: DMA is not in transfer action 1: DMA is in transfer action	R

14.5.12 Transmit Source Address Register (DMA_SARx) (x=0~3)

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SAR [31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SAR [15:0]															
Bit Flag	Bit Name Function Read/Write														
b31-b0	SAR [31:0]	Transmission source address	Sets the transmission source address Note: <ul style="list-style-type: none"> - When the transmission data width is 16bit, i.e. DMA_CHxctl.HSIZE=01, SAR[0] is invalid. R/W Pass When the input data width is 32bit, i.e. DMA_CHxCTL.HSIZE=1x, SAR[1:0] no Effect. 												

14.5.13 Transmit Destination Address Register (DMA_DARx) (x=0~3)

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
DAR[31:16]															

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

Bit Flag	Bit Name	Function	Read/Write
b31-b0	DAR[31:0]	Transport Destination Address	Sets the transmission destination address Note: <ul style="list-style-type: none"> - When the transmission data width is 16bit, i.e. DMA_CHxCTL.HSIZE=01, DAR[0] is invalid. R/W Pass When the input data width is 32bit, i.e. DMA_CHxCTL.HSIZE=1x, DAR[1:0] no Effect.
b31-b16	CNT [15:0]	Number of transmissions	Total number of transmissions, one block of data initiated per request, number of transmissions at completion

14.5.14 Data Control Register (DMA_DTCTLx) (x=0~3)

Reset value: 0x0000_0001

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CNT [15:0]															

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1
BLKSIZE[9:0]														

b0 bit	Flags	Bit Name	Function
Read/Write			

b31-b16	CNT [15:0]	Number of transmissions	Total number of transmissions, one block of data initiated per request, number of transmissions at completion The counter is decremented by 1, and a transmission completion interrupt occurs when it is decremented to 0. If set to 0, it indicates no limit transfers, one block of data per startup request, number of transfers counter on completion Holding 0 unchanged does not generate a transmission completion interrupt.	R/W
b15-b10	Reserved	-	Reads "0" and writes "0".	R/W
b9-b0	BLKSIZE[9:0]	Data block size	Sets the size of the data block, a maximum of 1024 data can be configured. Width of each data determined by the HSIZE bit of the DMA_CHxCTL register. A register value of 1 results in an error every time the Transmit 1 data, set to 0 to transmit 1024 data at a time.	R/W

14.5.15 Repeat Region Size Register (DMA_RPTx) (x=0~3)

Reset value: 0x0000_0000

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

b0

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31-b26	Reserved	-	Reads "0" and writes "0".	R/W
b25-b16	DRPT[9:0]	Destination Address Repeat Area Size	Setting the size of the destination address repeat area The target device reloads the target address after every DRPT data transfer to the value initially set in the DMA_DARx register. A register set to 10 results in address reloading after every 10 data transfers, and a register set to A value of 0 results in an address reload after every 1024 data passes. The width of each data is defined by the The HSIZEx bit of the DMA_CHxCTL register determines this.	R/W
b15-b10	Reserved	-	Reads "0" and writes "0".	R/W
b9-b0	SRPT[9:0]	Source Address Repeat Area Size	Setting the source address repeat area size The source device reloads the source address after every SRPT data transfer to the value initially set in the DMA_SARx register. The register is set to 10 for address reloading after every 10 data transfers and 0 for address reloading after every 1024 data transfers. The width of each data is determined by the HSIZEx bit of the DMA_CHxCTL register.	R/W

This register configures the size of the source and destination address duplicate regions. The use of duplicate address transfer requires that the SRPTEN/DRPREN bits of the DMA_CHxCTL register be configured to be valid and that the SINC/DINC bits of the DMA_CHxCTL register be configured so that the address update method is either self-incrementing or self-decrementing, and that the address reloading function is disabled if it is fixed.

DMA_RPTx, DMA_RPTBx These two registers share the same address and are both used to define the repeat region size. Which one to use depends on whether reset is enabled for that channel. DMA_RPTx is used when reset is not enabled and DMA_PRTBx is used when reset is enabled.

14.5.16 Repeat Region Size Register B(DMA_RPTBx) (x=0~3)

Reset value: 0x0000_0000

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

b0

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31-b26	Reserved	-	Reads "0" and writes "0".	R/W
b25-b16 Address Repetition Block	DRPTB[9:0]	Destination domain size	Setting the size of the destination address repeat area The target device reloads the target address to the initial setting of the DMA_DARx register after every DRPTB block of data is transmitted. The data block size is determined by DMA_DTCTLx.BLKSIZE with DMA_CHxCTL.HSIZE.	R/W
b15-b10 Address Repetition Region	Reserved	-Reserved	Read "0", write "0".	R/W
b9-b0	SRPTB[9:0]	Source magnitude	Setting the source address repeat area size The source device reloads the source address to the value initially set in the DMA_SARx register after every SRPTB block of data is transmitted. The data block size is determined by DMA_DTCTLx.BLKSIZE with DMA_CHxCTL.HSIZE.	R/W

This register configures the size of the source and destination address duplicate regions. The use of Repeat Address Transmission requires that the SRPTEN/DRPREN bits of the DMA_CHxCTL register be configured to be valid and that the SINC/DINC bits of the DMA_CHxCTL register be configured so that the address update method is either self-incrementing or self-decrementing, and that the Repeat Address Transmission function is not available if it is fixed.

DMA_RPTx, DMA_RPTBx These two registers share the same address and are both used to define the repeat region size. Which one to use depends on whether reset is enabled for that channel. DMA_RPTx is used when reset is not enabled and DMA_PRTBx is used when reset is enabled.

14.5.17 Source Device Discontinuous Address Transfer Control Register (DMA_SNSEQCTLx) (x=0~3)

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SNSCNT [11:0]												SOFFSET [19:16]			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SOFFSET[15:0]															

Bit Flag	Bit Name	Function	Read/Write
b31-b20 source address jumps	SNSCNT[11:0]	Number of quantitatively	Sets the size of the amount of data transferred before the source address jumps. The source device specifies the offset by SOFFSET for the source address after every SNSCNT data transmission. Jump. Setting the register to 10 causes the address to jump after every 10 data transfers, and setting it to 0 causes the address to jump after every 4096 data transfers.
b19-b0 source address jump	SOFFSET[19:0]	Ground for address offset	Sets the offset for source address jumps when discontinuous addresses are transmitted. The offset is relative to the current transfer address, the last transfer address before the jump. The direction of the jump is forward or backward based on the value of the channel control register DMA_CHxCTL.SINC. Refer to Figure 14-3. when DMA_CHxCTL.SINC is set to Address Fixed, discontinuous address transfers are not valid. The jump address will be calculated based on the number of bits set by the width of the data (DMA_CHxCTL.HSIZE) and the value of SOFFSET. Address offset = SOFFSET x (HSIZE(bit)/8) For example, when SOFFSET is set to 10 and HSIZE is word (32bit), the address offset is $10 \times 4 = 40$, if HSIZE is half word (16bit), the offset is $10 \times 2 = 20$, and if HSIZE is byte (8bit), the offset is $10 \times 1 = 10$. Source address for next transmission = source address for current transmission \pm address offset

Using source device discontinuous transfers requires that the SNSEQEN bit of the DMA_CHxCTL register be configured to be active and that the SINC bit of the DMA_CHxCTL register be configured so that the address update method is either self-incrementing or self-decrementing.

DMA_SNSEQCTLx, DMA_SNSEQCTLBx These two registers share the same address and are both used to define discontinuous transfers. Which one to use depends on whether reset is enabled for that channel. DMA_SNSEQCTLx is used when reset is not enabled and DMA_SNSEQCTLBx is used when reset is enabled.

14.5.18 Source Device Discontinuous Address Transfer Control Register

B(DMA_SNSEQCTLBx) (x=0~3)

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SNSCNTB [11:0]										SNSDIST [19:16]					
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SNSDIST [15:0]															

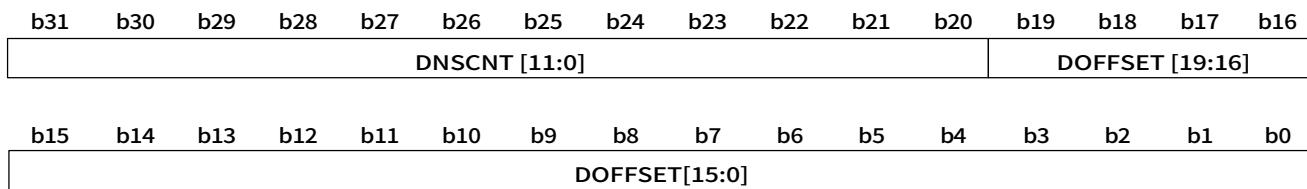
Bit Flag	Bit Name	Function	Read/Write
b31-b20 source address jumps	SNSCNTB[11:0]	Number of quantitatively	Sets the size of the amount of data transferred before the source address jumps. Source device after every SNSCNTB data block is transmitted source address as specified by SNSDIST ground Address spacing jumps. The data block size is determined by DMA_DTCTLx.BLKSIZE with DMA_CHxCTL.HSIZE.
b19-b0 SNSDIST[19:0]	site spacin g	Source Discontinuous Area Ground	Sets the spacing between the two discontinuous regions of the source device when discontinuous addresses are transmitted. The direction of the jump is forward or backward depending on the value of the channel control register DMA_CHxCTL.SINC. Refer to Figure 14-4. discontinuous address transfers are not valid when DMA_CHxCTL.SINC is set to address fixed. The address spacing will be based on the width of the data (DMA_CHxCTL.HSIZE) set by bit number and SNSDIST values were calculated. Address spacing = SNSDIST x (HSIZE(bit)/8) For example, when SNSDIST is set to 10 and HSIZE is word (32bit), the address spacing is $10 \times 4 = 40$, if HSIZE is half word (16bit), the spacing is $10 \times 2 = 20$, and if HSIZE is byte (8bit), the spacing is $10 \times 1 = 10$. Source address for next transmission = first address of current source discontinuity region \pm address spacing
			R/W

Using source device discontinuous transfers requires that the SNSEQEN bit of the DMA_CHxCTL register be configured to be active and that the SINC bit of the DMA_CHxCTL register be configured so that the address update method is either self-incrementing or self-decrementing.

DMA_SNSEQCTLx, DMA_SNSEQCTLBx These two registers share the same address and are both used to define discontinuous transfers. Which one to use depends on whether reset is enabled for that channel. DMA_SNSEQCTLx is used when reset is not enabled and DMA_SNSEQCTLBx is used when reset is enabled.

14.5.19 Target Device Discontinuous Address Transfer Control Register (DMA_DNSEQCTLx) (x=0~3)

Reset value: 0x0000_0000



Bit Flag	Bit Name	Function	Read/Write
b31-b20 address jump of the	DNSCNT[11:0]	Destination data volume	Sets the size of the amount of data transferred before the destination address jumps. The destination address of the target device after each DNSEQCNT data transmission is specified as DOFFSET biased. Shift Jump. A register set to 10 will address jump after every 10 data transfers, and a register set to 0 will address jump after every 4096 data transfers.
b19-b0 address jump of the	DSOFFSET[19:0]	Destination address offset	R/W Sets the offset of the destination address jump when discontinuous addresses are transmitted. The offset is relative to the current transfer address, the last transfer address before the jump. The direction of the jump is forward or backward based on the value of the channel control register DMA_CHxCTL.DINC. Refer to Figure 14-3. when DMA_CHxCTL.DINC is set to Address Fixed, Discontinuous Address Transfers are disabled. The jump address will be calculated based on the number of bits set by the width of the data (DMA_CHxCTL.HSIZE) and the value of DOFFSET. Address offset = DOFFSET x (HSIZE(bit)/8) For example, when DOFFSET is set to 10 and HSIZE is word (32bit), the address offset is $10 \times 4 = 40$, if HSIZE is half word (16bit), the offset is $10 \times 2 = 20$, and if HSIZE is byte (8bit), the offset is $10 \times 1 = 10$. Destination address for next transmission = destination address for current transmission \pm address offset

Using the target device discontinuous transfer requires that the DNSEQEN bit of the DMA_CHxCTL register be configured to be active and that the DINC bit of the DMA_CHxCTL register be configured so that the address update method is self-incrementing or self-decrementing.

DMA_DNSEQCTLx, DMA_DNSEQCTLBx These two registers share the same address and are both used to define discontinuous transfers. Which one to use depends on whether reset is enabled for that channel. DMA_DNSEQCTLx is used when reset is not enabled and DMA_DNSEQCTLBx is used when reset is enabled.

14.5.20 Target Device Discontinuous Address Transfer Control Register B(DMA_DNSEQCTLBx) (x=0~3)

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
DNSCNTB [11:0]												DNSDIST [19:16]			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DNSDIST [15:0]															

Bit Flag	Bit Name	Function	Read/Write
b31-b20 address jump of the	DNSCNTB[11:0]	Destination data volume	Sets the size of the amount of data transferred before the destination address jumps. The destination address of the target device after each DNSCNTB data block transmission is indicated by DNSDIST. Fixed address spacing jumps. The data block size is determined by DMA_DTCTLx.BLKSIZE with DMA_CHxCTL.HSIZE.
b19-b0	DNSDIST[19:0]	Targeted Discontinuous Areas address distance	When a discontinuous address is transferred, the spacing between the two discontinuous regions of the target device is set. The direction of the jump is based on the value of the channel control register DMA_CHxCTL.DINC forward or Jump backward. Refer to Figure 14-4. discontinuous address transfers are invalidated when DMA_CHxCTL.DINC is set to address fixed. The address spacing will be based on the width of the data (DMA_CHxCTL.HSIZE) set by bit numbers and DNSDIST values were calculated. Address spacing = DNSDIST x (HSIZE(bit)/8) For example, when DNSDIST is set to 10 and HSIZE is word (32bit), the address spacing is $10 \times 4 = 40$, if HSIZE is half word (16bit), the spacing is $10 \times 2 = 20$, and if HSIZE is byte (8bit), the spacing is $10 \times 1 = 10$. Destination address for next transmission = first address of current destination discontinuity region \pm address spacing
			R/W

Using the target device discontinuous transfer requires that the DNSEQEN bit of the DMA_CHxCTL register be configured to be active and that the DINC bit of the DMA_CHxCTL register be configured so that the address update method is self-incrementing or self-decrementing.

DMA_DNSEQCTLx, DMA_DNSEQCTLBx These two registers share the same address and are both used to define discontinuous transfers. Which one to use depends on whether reset is enabled for that channel. DMA_DNSEQCTLx is used when reset is not enabled and DMA_DNSEQCTLBx is used when reset is enabled.

14.5.21 Chain Pointer Register (DMA_LLPx) (x=0~3)

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16												
LLP [31:16]																											
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1													
LLP [15:2]																											
<hr/>																											
b0 bit	Flags	Bit Name	Function																								
<hr/>																											
Read/Write																											
<hr/>																											
b31-b2	LLP[31:2]	chain pointer descriptor for the next transfer is located at word	When a chain transfer is valid, sets the address where the chain pointer descriptor for the next transfer is located at word																								
Alignment, i.e., LLP[1:0] is fixed to 0																											
b1-b0	Reserved	-Reserved	Read "0", write "0".																								
R/W																											

14.5.22 Channel Control Register (DMA_CHxCTL) (x=0~3)

Reset value: 0x0000_1000

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
-	-	-	IE	LLP RUN	LLP EN	HSIZE[1:0]	DNSE QEN	SNSE QEN	DRP TEN	SRP TEN	DINC[1:0]	SINC[1:0]			

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31-b13	Reserved	-	Reads "0", writes "0".	R/W
b12	IE	interrupt enable bit	Configure whether the channel generates interrupts. 0: No interrupt is generated for this channel 1: The channel generates an interrupt	R/W
b11	LLPRUN	Chained Transfer Mode Selection Whether or not to start the transmission corresponding to the new descriptor immediately after the character R/W selection 0: do not transmit immediately, wait for the next transmission request to be generated and start transmission 1: Transmission starts immediately after the new descriptor is loaded	When a chained transfer is valid, set the new description pointed to by the load chain pointer when the current transfer completes	0: Chain 1: Chain transmission effective
b10	LLPEN	Chain transmission enable transmission disabled		R/W
b9-b8	HSIZE[1:0]	Width of transmitted data R/W 00: 8bit 10, 11: 32bit		01: 16bit
b7	DNSEQEN	Destination address discontinuous transmission enable	0: Discontinuous address transfer not allowed 1: Discontinuous address transfer allowed	R/W
b6	SNSEQEN	Source address discontinuous transmission enable	0: Discontinuous address transfer not allowed 1: Discontinuous address transfer allowed	R/W
b5	DRPTEN	b3-b2	Target Repeat Transmission Function Sets whether to allow the destination address to reload the initial value DINC[1:0] Enable Bit	How the destination address is updated
b4	SRPTEN	b1-b0	Source Repeat Transmit SINC[1:0] Function Enable Bit	How the source

1: Overloading
Sets whether
the source R/W R/W R/W R/W
address is
allowed to
reload the initial
value 0: No
reloading
1: Overloading
00: Fixed
01: Incremental
10, 11:
decreasing
00: Fixed
01: Incremental
10, 11:
decreasing

14.5.23 Channel Monitor Registers (DMA_MONSARx, DMA_MONDARx, DMA_MONDTCTLx, DMA_MONRPTx, DMA_MONSSEQCTLx, DMA_MONDNSEQCTLx) (x=0~3)

These monitor registers correspond to the corresponding channel configuration registers with the same register bit configuration, but all are read-only registers.

The Channel Configuration Register remains unchanged before and after DMA transfers, while the Channel Monitor Register is updated after the DMA completes the transfer corresponding to each request, i.e., after each completed data block transfer. The content and method of updating are as follows:

- DMA_MONSARx.SAR[31:0] DMA_MONDARx.DAR[31:0]: updated to the address of the next transmission in the fixed/incrementing/decrementing/reloading/discontinuous hopping mode as set in the channel configuration register.
- DMA_MONDTCTLx.CNT[15:0]: minus 1, or hold to 0 if already 0.
- DMA_MONRPTx.SRPT[9:0], DRPT[9:0]: when channel reset is invalid, subtract the block size, minus 0 to subtract or reload the DMA_RPTx setpoint when the value is less than the block size. When reset is valid, minus 1, minus to 0 to reload DMA_RPTBx set value.
- DMA_MONSSEQCTLx.SNSCNT[11:0], DMA_MONDNSEQCTLx.DNSCNT[11:0]: When the channel reset is invalid, subtract the block size, reduce to 0 or the original DMA_SNSEQCTLx/DMA_DNSEQCTLx set value when the value is less than the block size value. When the reset is valid, minus 1, minus 0 reloads the DMA_SNSEQCTLBx/DMA_DNSEQCTLBx set value.

Monitor register bits other than the above remain the same as the configuration registers.

14.6 Precautions for use

- 1 The registers of DMA only support 32bit read/write, 8/16bit read/write operation is invalid.
- 2 Write channel configuration registers are not valid during DMA transfers. The channel configuration registers include DMA_SARx, DMA_DARx, DMA_DTCTLx, DMA_RPTx, DMA_RPTBx, DMA_SNSEQCTLx, DMA_SNSEQCTLBx, DMA_DNSEQCTLx, DMA_DNSEQCTLBx, DMA_LLPx, DMA_DTCTLx, DMA_DTCTLBx, and DMA_LLPx. DMA_CHxCTL (x=0~3). Please make sure to write the above registers when DMA is in idle state, or read them out after writing the registers to confirm whether the write is successful or not.
- 3 When a bus error occurs during a DMA transfer and another channel is waiting, the DMA

enters a locked state and cannot respond to all subsequent channel transfer requests. After entering this state, the DMA cannot be unlocked by configuring its own registers. It is necessary to perform a software reset or notify the peripheral circuitry to reset the system.

Method to detect DMA lockup: Query the error flag bit DMA_INTSTAT0.TRNERR[7:0] (which can be placed in the DMA error interrupt processing) if the flag bit is 0, it means that there is no bus error and the DMA is not locked up. If the flag bit is not 0, continue to query the channel status bits DMA_CHSTAT.CHACT[7:0], if CHACT[7:0]=0x0, it means that the DMA is not locked. If one of the bits, such as CHACT[x]=1 for channel x, remains 1 for a long period of time (more than the time required for its normal transmission) and CHACT[x] remains 1 even if the corresponding channel license bit, DMA_CHEN.CHEN[x], is cleared to 0 by the software, it means that the DMA is locked.

To prevent the DMA from going into a lockup state, you need to avoid giving the DMA access to areas where bus errors can occur, such as reserved address space, protected address space, etc. If you want to access areas where bus errors can occur, you can disable the other channels of the DMA unit so that no other channel is waiting for a bus error. If you want to access an area where a bus error may occur, you can disable the other channels of this DMA unit to avoid having other channels waiting for a bus error.

15 Voltage Comparator (CMP)

15.1 summary

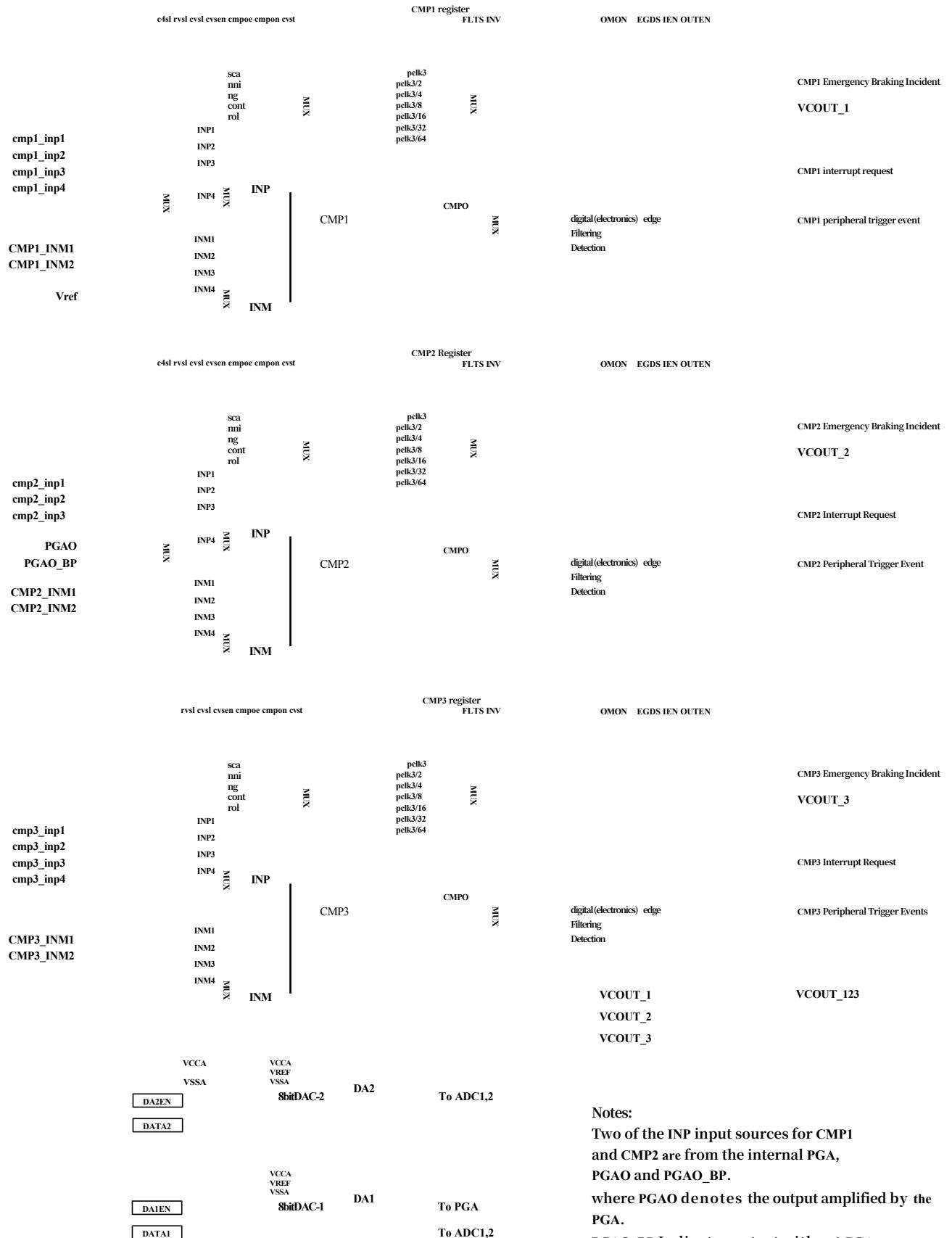
The Comparator (hereinafter referred to as CMP) is a peripheral module that compares two analog voltages (positive voltage INP and negative voltage INM) and outputs the result of the comparison, with three independent comparison channels. Each comparison channel has three independent comparison channels.

There are four input sources for each of the voltages, and each can be selected for a single comparison, or multiple positive voltages can be selected for scanning comparison with the same negative voltage. Comparison results can be read through registers or output to external pins, and interrupts and events can be generated.

This series is also equipped with two 8-bit digital-to-analog converters (hereinafter referred to as 8bitDACs) whose analog outputs can be used as negative-terminal voltage input sources for CMP.

Table 15-1 CMP Detailed Specifications

sports event	norm
comparison channel	3 comparison channels: CMP1~3
Positive Voltage Input Source	12 input sources: 10 external analog inputs, 2 internal PGA outputs
Negative voltage input source	7 input sources: 4 external analog inputs, 1 internal Vref, 2 internal D/A outputs
Comparative results	Can be read from registers or output to external pin VCOUT with selectable polarity
operating mode	Single Comparison Mode and Scanning Comparison Mode
digital filtration	7 sampling frequencies selectable, can be turned off when filtering is not required
Interrupts and Events	Detects the active edge of the comparison result and generates interrupts and peripheral trigger events
Low Power Control	Module stop function reduces system power consumption



Notes:

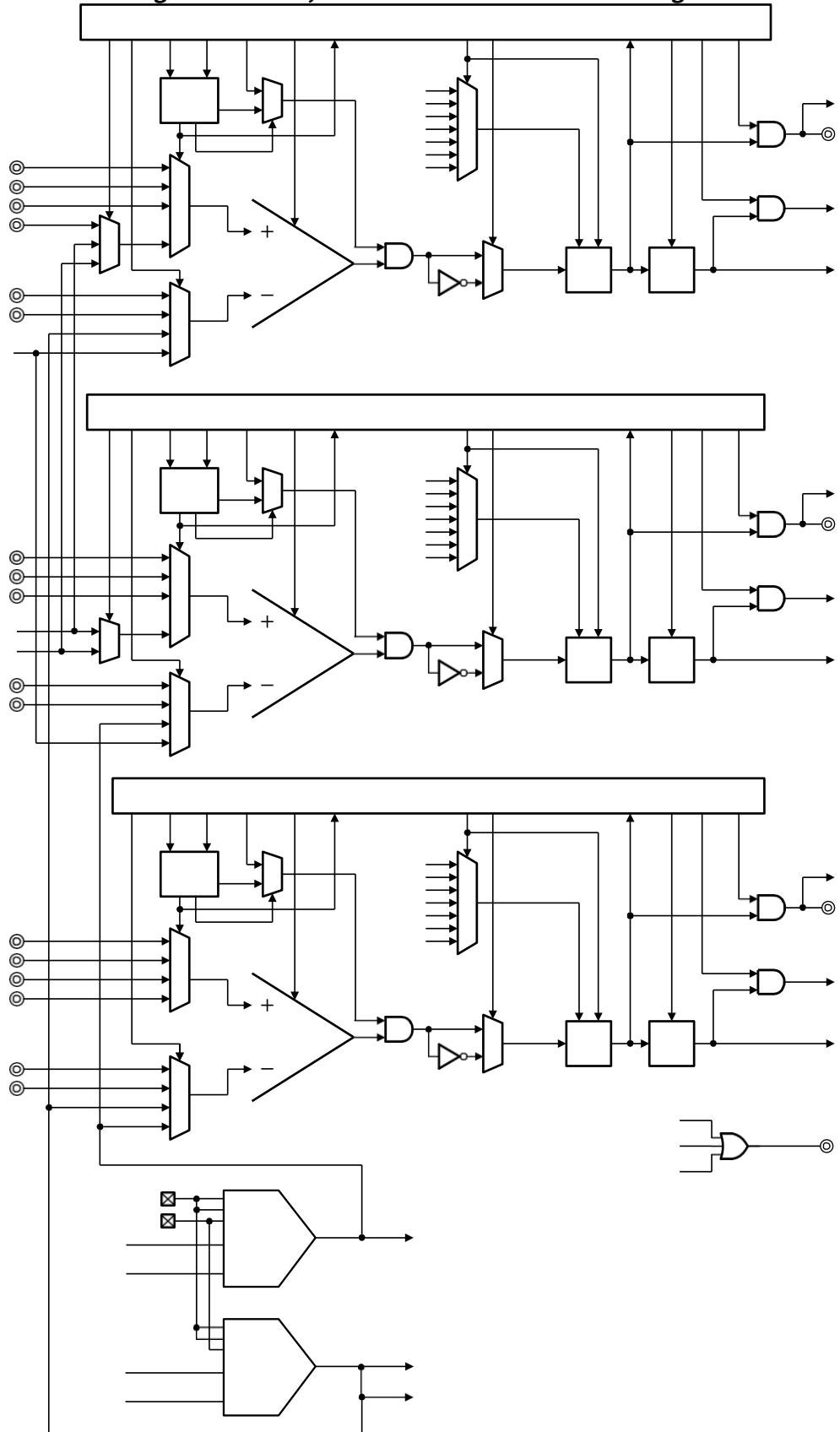
Two of the INP input sources for CMP1 and CMP2 are from the internal PGA, PGAO and PGAO_BP.

where PGAO denotes the output amplified by the PGA.

PGAO_BP Indicates output without PGA amplification.

Both inputs require the PGA to be set to active, please refer to ADC_PGACR register description for setting.

Figure 15-1 CMP, 8bitDAC Function Connection Diagram



15.2 Functional Description

15.2.1 voltage comparison

When the CMP operates in the single comparison mode, selecting any comparison channel and choosing the positive voltage INP and the negative voltage INM for it can realize the comparison of these two voltages. Figure 15-2 illustrates the operation of the CMP. If the positive polarity output is set, the CMP outputs a high level when INP is higher than INM, and a low level when INP is lower than INM. Conversely, if the negative polarity output is set, the CMP outputs a low level when INP is higher than INM and a high level when INP is lower than INM. Finally, the comparison result can be read from the respective comparison result monitor register CMPMON.OMON bit of each comparison channel, or it can be output to the external pin VCOUT.

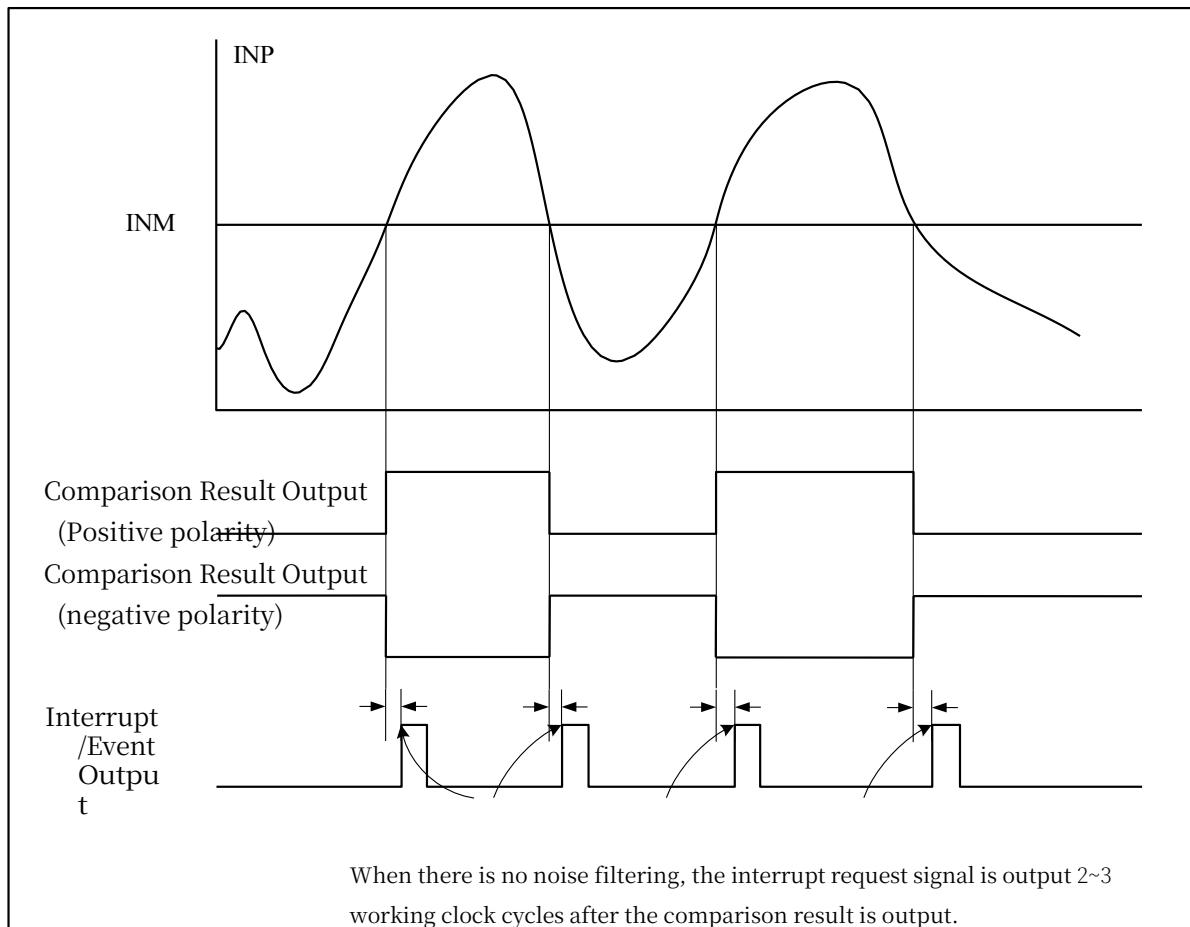


Figure 15-2 CMP Working Diagram

15.2.2 digital filtration

Each comparison channel has a digital filter circuit, which can filter the noise of the comparison result. The filter circuit samples the comparator output CMPO three times and then compares it; if the three sampling results are consistent, the sampled result will be output as the comparison result; if the three sampling results are not consistent, the comparison result will be kept unchanged. The sampling frequency has seven selectable steps, please refer to the function description of register CMP_CTRL.FLTS[2:0] for details. FLTS[2:0]. The filtering function can also be turned off, in which case the comparison result output is exactly the same as the comparator output CMPO.

15.2.3 Interrupts and Events

The interrupt request can be generated when the comparison result output is changed, as well as the event triggered by other peripheral devices, and the edge for generating the interrupt and the event can be selected as the rising edge, the falling edge, or the double edge of the comparison result output. The interrupt request for comparison channel 1 (CMP1) can also wake up/stop the low-power mode, but the interrupt edge must be selected as a rising edge and the digital filter function must be turned off.

Refer to the following procedure to set up when comparing interrupts and events for the first time.

- 1) Set CMP_VLTSEL to select the INP and INM input sources.
- 2) Set CMP_CTRL.INV to select the comparator output polarity.
- 3) Set CMP_CTRL.EDGSL[1:0] to select the interrupt generation edge.
- 4) Set CMP_CTRL.CMPON to start the comparator and wait for 300ns stabilization time.
- 5) Set CMP_CTRL.CMPOE to allow comparator output.
- 6) If interrupts are used, set CMP_CTRL.IEN to allow interrupts.
If events are used, the trigger event of the activated peripheral is selected as the CMP Compare event.

15.2.4 Scanning Comparison Mode

When setting register CMP_VLTSEL.CVSL[3:0] to select two or more INPs as the comparison object, the CMP enters into the scanning comparison mode, and then sets register CMP_CTRL. The INM input remains unchanged while the INP input is automatically switched according to the set scanning period. If a valid edge selected by the register CMP_CTRL.EDGSL[1:0] bit is detected during the scanning process, the register CMP_CTRL.CVSEN bit will be cleared and the scanning process will be suspended automatically. At this point, the register CMP_OUTMON.CVST[3:0] bits can be read to determine the current INP input. When the CMP_CTRL.CVSEN bit is written to 1 again, the scanning action continues.

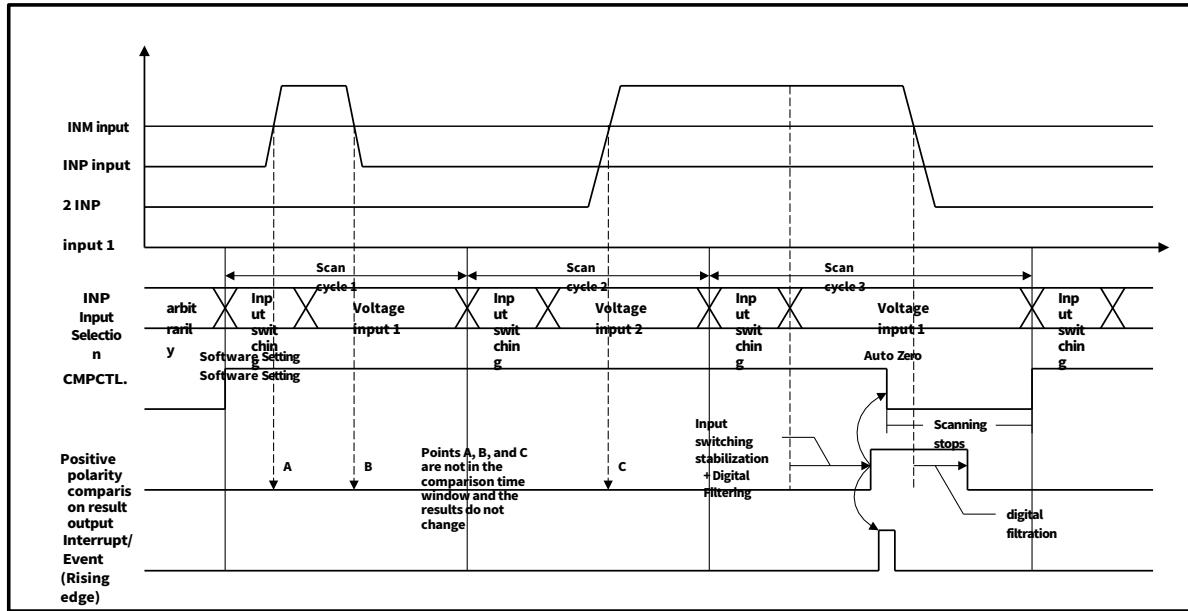


Figure 15-3 Scan Mode Action Diagram

The above diagram shows the action of scanning two INP inputs. During the first two scan cycles, the comparison result output remains unchanged because neither INP Input 1 nor INP Input 2 changes within their respective comparison time windows. Entering scan cycle 3, INP input 1 is higher than INM causing the comparator to invert, and the output is digitally filtered and passed to the comparison result output. At this point, the scanning action automatically stops and interrupts and events are generated. The CMP then remains in the INP input 1 vs. INM state until the software restarts the scan mode.

Refer to the following procedure to set up the scanning comparison mode.

- 1) Set CMP_VLTSEL[3:0] to select INP and INM input sources (two or more INP selections)
- 2) Set CMP_CVSSTB to set the voltage switching stabilization time.
- 3) Set CMP_CVSPRD to set the voltage sweep period.
- 4) Set CMP_CTRL.INV to select the comparator output polarity.
- 5) Set CMP_CTRL.FLTS[2:0] to select the digital filter sampling frequency.
- 6) Set CMP_CTRL.EDGSL[1:0] to select the interrupt generation edge.
- 7) Set CMP_CTRL.IEN to allow interrupts.
- 8) Set CMP_CTRL.CMPON to start the comparator and wait for 300ns stabilization time.
- 9) Set CMP_CTRL.CMPOE to allow comparator output.
- 10) Set CMP_CTRL.CVSEN to start scanning.

The voltage switching stabilization time and the voltage scanning period correspond to the number of PCLK3 periods. To ensure correct operation, make sure that the setting value satisfies the condition "Scan period > Switching stabilization time + Filter sampling period × 4 + 5".

15.2.5 8bit-DAC setting

Two 8bit-DACs provide the CMP with two internal voltages, DA1 and DA2. the output voltage during operation is:

$$\text{Output voltage} = \text{VCCA} \times$$

conversion data/255 The conversion data for DA1 and DA2 are set by CMP_DADR1 and CMP_DADR2, respectively.

Please refer to the following procedure for setting when using:

- 1) Set CMP_DADR1 and CMP_DADR2 to set the conversion data.
- 2) Set CMP_DACR to start DA1 and DA2.
- 3) Wait for DA conversion stabilization time.
- 4) Sets the comparator.

15.3 caveat

15.3.1 Module stop function

The CMP has a module stop function, and the digital portion of the module can be turned off by setting the module stop register. The CMP is initially stopped, and the CMP register can be accessed only when the set module is operating. Please refer to the Low Power Consumption section for detailed description.

15.3.2 Action when the module is stopped

When the CMP enters the stop state from the working state, the analog portion of the CMP continues to remain operational and the power consumption is equal to the working state. To reduce power consumption, set CMP_CTRL.CMPON to 0 first to stop the analog part of CMP.

Similarly, when the 8bit-DAC enters the stop state in the working state, the analog portion of the DA continues to operate and the power consumption is equivalent to the working state. To reduce the power consumption, set the DA1EN and DA2EN bits of the DACR to 0 to stop the analog part of the 8bit-DAC.

15.3.3 Stopping the action in low-power mode

When the HC32F46xx enters Stop Low Power Mode in the CMP operating state, the CMP will continue to operate with power consumption equal to the level before entering Stop Low Power Mode. To further reduce the power consumption in Stop Low Power Mode, please clear CMP_CTRL.CMPON to 0 before entering Stop Low Power Mode to stop the CMP operation.

Similarly, when the HC32F46xx enters Stop Low Power Mode during 8bit-DAC conversion, the D/A outputs will be held and the power consumption will be equal to the level before

entering Stop Low Power Mode. To further reduce the power consumption in stop-low mode, set the DA1EN and DA2EN bits of DACR to 0 to stop the 8bit-DAC before entering stop-low mode.

15.3.4 Action in power-down low-power mode

When the HC32F46xx enters the power-down low-power mode, both CMP and 8bit-DAC will stop working. the CMP comparison result output is Low, and the 8bit-DAC output becomes high-resistance state.

15.4 Register Description

CMP1 Base Address:

0x4004_A000 CMP2 Base

Address: 0x4004_A010 CMP3

Base Address: 0x4004_A020

Table 15-2 CMP Register List

register name	notation	offset address	bit width	reset value
CMP Control Register	CMP_CTRL	0x000	16	0x0000
CMP Voltage Selection Register	CMP_VLTSEL	0x002	16	0x0000
CMP Result Monitor Register	CMP_OUTMON	0x004	16	0x0000
CMP Stabilization Time Register	CMP_CVSSTB	0x006	16	0x0005
CMP Scan Cycle Register	CMP_CVSPRD	0x008	16	0x000F
8bitDAC data register 1 for CMP	CMP_DADR1	0x100	16	0x0000
8bitDAC Data Register 2 for CMP	CMP_DADR2	0x102	16	0x0000
8bit DAC Control Register for CMP	CMP_DACR	0x108	16	0x0000
CMP Internal Reference Voltage AD Conversion Registers	CMP_RVADC	0x10C	16	0x0000

Attention:

-CMP_DADR1,CMP_DADR2,CMP_DACR are common registers and exist only in CMP1.

15.4.1 CMP Control Register (CMP_CTRL)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CMPON	CMPOE	INV	OUTEN	-	-	-	CVSEN	IEN	EDGSL[1:0]	-	-	FLTSL[2:0]			
classifier for honorific people	marking	celebrity	functionality										fill out or in (information on a form)		
b15	CMPON	Comparator Operating License	0: Comparator stop (Comparator output Low) (reset value) 1: Comparator operation Attention: -Comparator Operating License CMPON After the comparator operating license CMPON is set to "1", it needs to wait for about 300ns. of the work stabilization time before subsequent operations can be performed.										R/W		
b14	CMPOE	Comparator output license	0: Comparator output disabled (Comparator output Low) (reset value) 1: Comparator output license 0: Comparator positive polarity output (reset value) 1: Comparator negative polarity output (comparator output is inverted) Note: - Please rewrite when the comparator output is disabled (i.e. CMPOE bit is "0") INV. changing the INV bit may cause an interrupt or peripheral trigger event due to the Please set this register when the interrupt disable or peripheral trigger function is disabled. After setting the register, clear the corresponding interrupt flag.										R/W		
b13	INV	Comparator output polarity selection	0: Comparison result output prohibited 1: Allow comparison results to be output 0: comparative voltage scanning stop (reset value) 1: Start of comparative voltage scan note: - CVSEN will be automatically cleared when a valid edge of the comparison result is detected, if required To continue scanning, write "1" again.										R/W		
b12	OUTEN	Comparison result output license (comparison result output Low) (reset value)	0: Edge of comparison result not detected (reset value) 1 0: Detecting the falling edge of the comparison result 1 1: Detecting the rising and falling edges of the comparison result -O" for reading, "O" for writing. writes "0". R										Reads "0", R		
b8	CVSEN	Comparison voltage scanning license	0: Compare interrupt license Compare interrupt disabled (reset value) 1: Compare interrupt allowed note: - CVSEN will be automatically cleared when a valid edge of the comparison result is detected, if required To continue scanning, write "1" again.										0: R/W		
b7	IEN	Compare interrupt license	0: Compare interrupt disabled (reset value) 1: Compare interrupt allowed 0 0: Edge of comparison result not detected (reset value) 1 0: Detecting the falling edge of the comparison result 1 1: Detecting the rising and falling edges of the comparison result -O" for reading, "O" for writing. writes "0". R										0: R/W		
b6~b5	EDGSL [1:0]	Comparison result valid edge Selection edge of comparison result detected	0 1: Rising 1 0: Detecting the falling edge of the comparison result 1 1: Detecting the rising and falling edges of the comparison result -O" for reading, "O" for writing. writes "0". R										0 1: Rising R/W		
b4~b3	Reserved		1 1: Detecting the rising and falling edges of the comparison result -O" for reading, "O" for writing. writes "0". R										Reads "0", R		

b2~b0 Selection	FLTSL [2:0]	Filter Sample	0 0 0: Noise filtering is not used (reset value)	www.xhsc.com.cn		
			0 0 1: Sampling with PCLK3			
			0 1 0: 2-division sampling using PCLK3			
			0 1 1: 4-division sampling using PCLK3			
			1 0 0: 8-division sampling using PCLK3			
			1 0 1: 16-division sampling using PCLK3	R/W		
			1 1 0: 32-division sampling using PCLK3			
			1 1 1: 64-division sampling using PCLK3			
Attention:						
- Please rewrite FLTSL[2:0] when the comparator output is disabled (i.e., the CMPOE bit is "0"). FLTSL[2:0] switches from "000b" to the other						

If you want to use the filtered output after 4 samples as an interrupt request or a peripheral trigger event, use the filtered output after 4 samples as an interrupt request or a peripheral trigger event.

- Changing **FLTSL[2:0]** may cause an interrupt or peripheral trigger event, so please set this register in the interrupt disable or peripheral trigger function disabled state. After setting the register, clear the corresponding interrupt flag.
-

15.4.2 CMP Voltage Select Register (CMP_VLTSEL)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	C4SL [2:0]			CVSL[3:0]			-	-	-	-	-	RVSL[3:0]			

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b15	Reserved	-	Reads "0", writes "0".	R

b14~b12	C4SL [2:0]	INP4 Input Selection	0 0 0: no input (reset value) 0 0 1: Selects the internal PGAO output (valid only for CMP1 and CMP2) 0 1 0: Select internal PGAO_BP output (valid only for CMP1 and CMP2) 1 0 0: External CMP1_INP4 input selected (valid only for CMP1) Attention: 1. Setting is not valid for CMP3. 2. Disable setting of other values.	R/W
			3. When rewriting C4SL[2:0], refer to the following steps: (1) CMP_CTRL.CMPOE bit is cleared "0". (2) The C4SL[2:0] bits are written to the new value (note that only 1 bit is "1"). (3) Wait for the input to stabilize. (4) CMP_CTRL.CMPOE position "1". (5) Clear the corresponding interrupt flag bit. 0 0 0 1: Select INP1	

b11~b8	CVSL[3:0]	INP input selection	0 0 1 0: Select INP2 0 0 1 1: Select INP1, INP2 (scanning mode) 0 1 0 0: Select INP3 0 1 0 1: Select INP1, INP3 (scanning mode) 0 1 1 0: Select INP2, INP3 (scanning mode) 0 1 1 1: Select INP1, INP2, INP3 (scanning mode) 1 0 0 0: Select INP4 1 0 0 1: Select INP1, INP4 (scan mode) 1 0 1 0: Select INP2, INP4 (scan mode) 1 0 1 1: Select INP1, INP2, INP4 (scanning mode) 1 1 0 0: Select INP3, INP4 (scan mode) 1 1 0 1: Select INP1, INP3, INP4 (scanning mode) 1 1 1 0: Selection of INP2, INP3, INP4 (scan mode) 1 1 1 1: Select INP1, INP 2, INP 3, INP 4 (scanning mode)	R/W
--------	-----------	---------------------	---	-----

		(Style) Attention: ion: 1. See "Figure 15-1" for specific input sources. 2. Set C4SL[2:0] to select the input source when INP4 is selected. 3. Refer to the following steps when rewriting CVSL[3:0]: (1) CMP_CTRL.CMPOE bit cleared "0". (2) The CVSL[3:0] bits write the new value. (3) Wait for the input to stabilize. (4) CMP_CTRL.CMPOE position "1". (5) Clear the corresponding interrupt flag bit.
--	--	---

b7~b4	Reserved	-Reserved	Read "0", write "0".	R
			0 0 0 0: No input (reset value)	
			0 0 0 1: Select INM1	
			0 0 1 0: Select INM2	
			0 1 0 0: Select INM3	
			1 0 0 0: Select INM4	
			Attention:	
b3~b0 selection	RVSL[3:0]	INM input	<ul style="list-style-type: none"> 1. See "Figure 15-1" for specific input sources. 2. Setting other values is prohibited. 3. When changing RVSL[3:0], refer to the following procedure: <ul style="list-style-type: none"> (1) CMP_CTRL.CMPOE bit cleared "0". (2) The RVSL[3:0] bits are written to the new value (note that only 1 bit is "1"). (3) Wait for the input to stabilize. (4) CMP_CTRL.CMPOE position "1". (5) Clear the corresponding interrupt flag bit. 	R/W

15.4.3 CMP Result Monitor Register (CMP_OUTMON)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-		CVST[3:0]		-	-	-	-	-	-	-	-	OMON
<hr/>															
classifier for honorific people	marking	celebrity		functionality											fill out or in (information on a form)
b15~b12	Reserved	-		Reads "0", writes "0".											R
				0 0 0 0: No current INP input											
				0 0 0 1: The current INP input selection is INP1											
				0 0 1 0: Current INP input selection is INP2											
b11~b8	CVST[3..0]	INP input status bits		0 1 0 0 : The current INP input selection is INP3											R
				1 0 0 0: The current INP input selection is INP4											
				Attention:											
				1. This register is read-only and writes all "0" when written.											
				2. See Figure 15-1 for specific input sources.											
b7~b1	Reserved	-		Reads "0" and writes "0".											R
				0: Comparison result output "0" (reset value)											
				1: Comparison result output as "1"											
				Attention:											
b0	OMON	Comparison result monitor bit		1. This register is read-only and writes all "0" when written.											R
				2. In CMP_CTRL.FLTS[2:0]=000b (no noise filtering is used).											
				When setting the comparator operation in the case of (road), use 2 readings of the OMON bit status consistently. method to determine the comparison state.											

15.4.4 CMP Stabilization Time Register (CMP_CVSSTB)

Reset value: 0x0005

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	STB[3:0]
classifier for honorific people	marking	celebrity	functionality												fill out or in (information on a form)
b15~b4	Reserved	-	Reads "0", writes "0".												R
b3~b0	STB[3:0]	Input switching stabilization time	Sets the output stabilization time of the CMP when the INP input is switched in scanning comparison mode. The reset value is 0x5 and the set value can be any value between 0x0 ~ 0xF. $\text{CMP output stabilization time} = \text{PCLK3 period} \times \text{STB set value}$ For example, if PCLK3 is 40 MHz and STB is set to 0x5, the Scan period = $25(\text{nS}) \times 0x5 = 125(\text{nS})$												R/W
Attention: 1. Refer to "Comparator Characteristics - Input Channel Switching Stabilization" in the Electrical Characteristics section of the datasheet. Between." 2. To rewrite the STB, first stop scanning and switch to single comparison mode.															

15.4.5 CMP Compare Voltage Scan Period Register (CMP_CVSPRD)

Reset value: 0x000F

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PRD[7:0]
classifier for honorific people	marking	celebrity	functionality												fill out or in (information on a form)
b15~b8	Reserved	-	Reads "0" and writes "0".												R
b7~b0	PRD[7:0]	Comparison Voltage Scan Period	Sets the time interval, or scan period, for switching the INP input in the scan compare mode. The reset value is 0x0F and the set value can be any value between 0x0F ~ 0xFF. $\text{Scan period} = \text{PCLK3 period} \times \text{PRD set value}$ For example, if PCLK3 is 40 MHz and PRD is set to 0x50, the Scan period = $25(\text{nS}) \times 0x50 = 2(\mu\text{s})$												R/W
Attention:															

1. When setting the PRD, make sure that the operation is correct.

Scan period > Switching stabilization time + Filter sampling period × 4 + 5

2. To rewrite the PRD, first stop scanning and switch to single comparison mode.

15.4.6 8bit-DAC control register for CMP (CMP_DACR)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	DA2 EN	DA1 EN

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b15~b2	Reserved	-	Read "0" when you read it and write "0" when you write it.	R
b1	DA2EN	DA2 start bit	0: DA2 stop (reset value) 1: DA2 work	R/W
b0	DA1EN	DA1 start bit	0: DA1 stop (reset value) 1: DA1 work	R/W

15.4.7 CMP with 8bit-DAC data register (CMP_DADR1,CMP_DADR2)

Reset value: 0x0000

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

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b15~b8	Reserved	-	Read "0" when you read it and write "0" when you write it.	R
b7~b0	DATA[7:0]	DA conversion data	Any value between 8'h00 (reset value) ~ 8'hFF	R/W

15.4.8 CMP Internal Reference Voltage AD Conversion Register (CMP_RVADC)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
								-	-	-	VREF SW	-	-	DA2 SW	DA1 SW

classifier for honorific people	marking	celebrity	functionality	fill out or in (informa tion on a form)
b15~b8	WPRT[7:0]	write- protected bit	8'h55: VREFSW, DA2SW, DA1SW bit write active others: VREFSW, DA2SW, DA1SW Bit Write Invalid	R/W
			Note: WPRT will revert to "8'h00" for any value written at a time other than "8'h55".	
b7~b5	Reserved	-	Read "0" when you read it and write "0" when you write it.	R
b4	VREFSW	Internal Vref AD conversion switch	0: Internal Vref AD conversion pathway disconnected 1: Internal Vref AD conversion pathway connection Note: VREFSW, DA2SW, DA1SW can only have one bit "1".	R/W
b3~b2	Reserved	-0" is read when reading and "0" is written when writing. when reading, please write "0" when writing.	Read "0"	
b1	DA2SW	DA2 AD Transfer Switch	0: DA2 AD conversion pathway disconnected 1: DA2 AD conversion pathway connection Note: Only one bit of VREFSW, DA2SW, DA1SW can be "1". 0: DA1 AD conversion path is disconnected.	R/W
b0	DA1SW	DA1 AD Transfer Switch	1: DA1 AD conversion pathway connection Note: VREFSW, DA2SW, DA1SW can only have one bit "1".	R/W

16 Analog-to-digital converter modules (ADC)

16.1 summary

The ADC is a successive approximation analog-to-digital converter that supports up to 17 analog input channels at up to 12-bit resolution and converts analog signals from external pins as well as from within the chip. These analog input channels can be arbitrarily combined into a sequence, and a sequence can be converted in a single sweep or in a continuous sweep. The ADC module also includes an analog watchdog function that monitors the conversion results of any specified channel for exceeding a user-defined threshold.

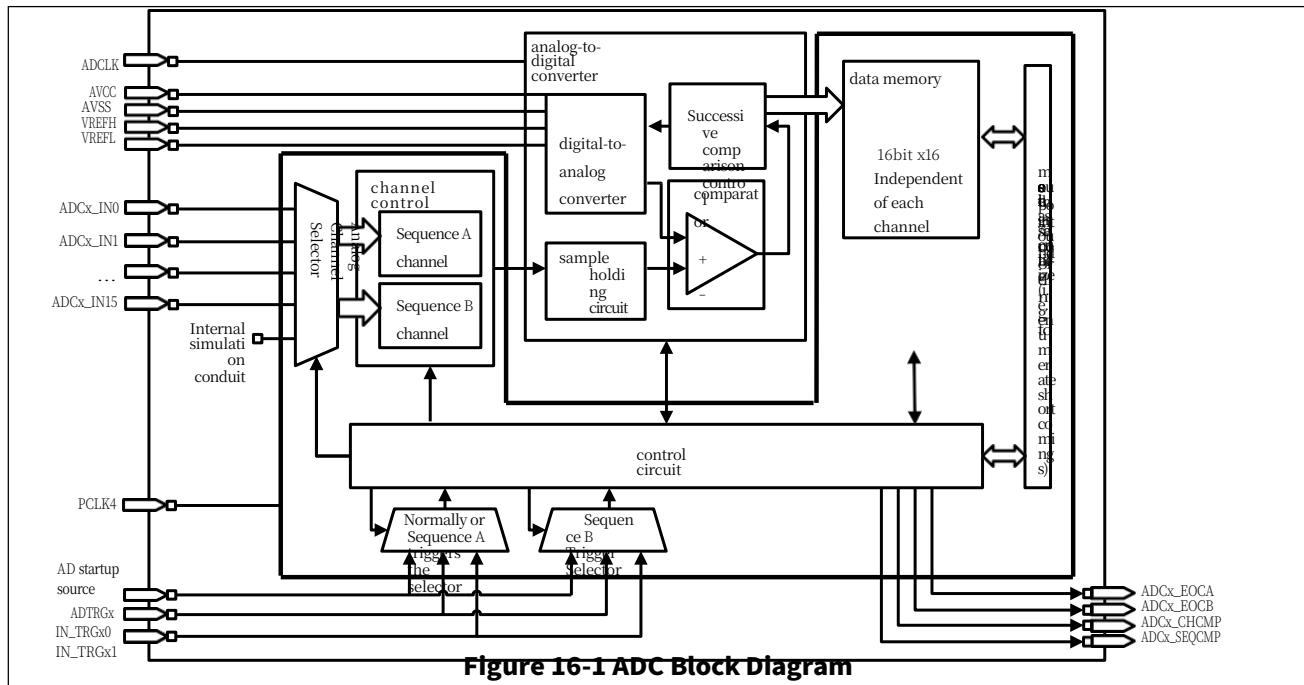
ADC Main Characteristics:

- high performance
 - Configurable 12-bit, 10-bit and 8-bit resolutions
 - The frequency ratio of the digital interface clock PCLK4 to the A/D converter clock ADCLK is selectable:
PCLK4:ADCLK=1:1, 2:1, 4:1, 8:1, 1:2, 1:4 ADCLK is selectable asynchronous to the system clock HCLK PLL clock, in which case PCLK4:ADCLK=1:1.
 - Sampling rate: 2.5MSPS (PCLK4=ADCLK=60MHz, 12-bit, 11-cycle sampling)
 - Individually programmable sampling time for each channel
 - Independent data registers for each channel
 - Data registers can be configured for data alignment
 - Consecutive multiple conversion averaging function
 - Analog watchdog to monitor conversion results
 - The ADC module can be set to stop when not in use.
- Analog Input Channels
 - Up to 16 external analog input channels
 - 1 detection channel for internal reference voltage/8bitDAC outputs
- Conversion start condition
 - Software Setup Conversion Starts
 - External event triggers conversion start
 - External pin triggers conversion start
- conversion mode
 - 2 scanning sequences A, B, single or multiple channels can be specified at will
 - Sequence A Single scan
 - Sequence A Continuous Scan
 - Dual sequence scanning, sequence A and B select trigger source independently,

sequence B has higher priority than A

- Synchronized mode (for devices with two or three ADCs)
- Interrupt and event signal output
 - Sequence A End-of-Scan Interrupts and Events ADC_EOCA
 - Sequence B End-of-Scan Interrupts and Events ADC_EOCB
 - Analog Watchdog Channel Compare Interrupt and Event ADC_CHCMP, Sequence Compare Interrupt and Event ADC_SEQCMP
 - All four of the above event outputs can be used to activate the DMA.

16.2 ADC System Block Diagram



The chip is equipped with two ADC module units, the configuration of each unit is different, refer to the following table:

Table 16-1 Specifications of Each ADC Unit

sports event	Module 1 (ADC1)		Module 2 (ADC2)
power supply	AVCC		
	AVSS/VREFL		
reference voltage	VREFH *1		
analog channel *2	CH0	ADC1_IN0	ADC12_IN4
	CH1	ADC1_IN1	ADC12_IN5
	CH2	ADC1_IN2	ADC12_IN6
	CH3	ADC1_IN3	ADC12_IN7
	CH4	ADC12_IN4	ADC12_IN8
	CH5	ADC12_IN5	ADC12_IN9
	CH6	ADC12_IN6	ADC12_IN10
	CH7	ADC12_IN7	ADC12_IN11
	CH8	ADC12_IN8	Internal analog channels (reference voltage/8bit DAC output) (out)
	CH9	ADC12_IN9	-
	CH10	ADC12_IN10	-
	CH11	ADC12_IN11	-
	CH12	ADC1_IN12	-
	CH13	ADC1_IN13	-
	CH14	ADC1_IN14	-
	CH15	ADC1_IN15	-
	CH16	Internal analog channel (reference voltage/8bit DAC output)	-
PGA	ADC1_IN0~3, ADC12_IN4~7. Any 1 channel of the 8bitDAC_1 outputs		-
hardware trigger source	external pin	ADTRG1	ADTRG2
	Interior and exterior of the film	IN_TRG10	IN_TRG20
		IN_TRG11	IN_TRG21

Attention:

- VREFH is available in LQFP100 package, not available in other packages, use AVCC instead of VREFH.

-
- The ADC analog channels CH0~CH15 and the actual input ADCx_INy can be set to register free mapping, and this table shows the default mapping relationship after reset.

16.3 Functional Description

16.3.1 ADC Clock

The ADC module requires the use of two clocks: the analog circuit clock ADCLK, and the digital interface clock PCLK4.

Both clocks come from the frequency divider in the clock controller. ADCLK is equivalent to PCLK2 and is synchronized with PCLK4, which has a frequency ratio relationship of 1:1, 2:1, 4:1, 8:1, 1:2, 1:4 with ADCLK.

ADCLK can select a PLL clock source that is asynchronous to the system clock HCLK, in which case PCLK4 and ADCLK are the same and synchronized with the same frequency. the maximum frequency of ADCLK is 60MHz.

16.3.2 Channel Selection

The ADC module has multiple channels that can be configured to convert in two sequences: Sequence A and Sequence B. Sequence A and B have separate channel selection registers ADC_CHSELRA and ADC_CHSELRB. Sequence A and B are equipped with independent channel selection registers ADC_CHSELRA and ADC_CHSELRB, each bit of which represents a channel, e.g., a bit0 bit with a 1 means to convert CH0, a bit0 bit with a 0 means not to convert CH0. The two sequences can independently select any one or more channels for conversion. For example, ADC_CHSELRA is set to 0x0055 and ADC_CHSELRB is set to 0x0002, then when the trigger condition of sequence A occurs, the four channels CH0, CH2, CH4 and CH6 will be converted sequentially. When the trigger condition of sequence B occurs, one channel, CH1, will be converted.

For the internal analog channel, there are 3 selectable analogs: 8bitDAC_1 output, 8bitDAC_2 output, and internal reference voltage (~1.1V) This is shown in the figure below. When selecting the internal analog channel as the object of ADC conversion, you need to set the internal channel selection register first, and then set the ADC's channel selection register ADC_CHSELRA/ADC_CHSELRB to select the internal channel before you can start ADC conversion.

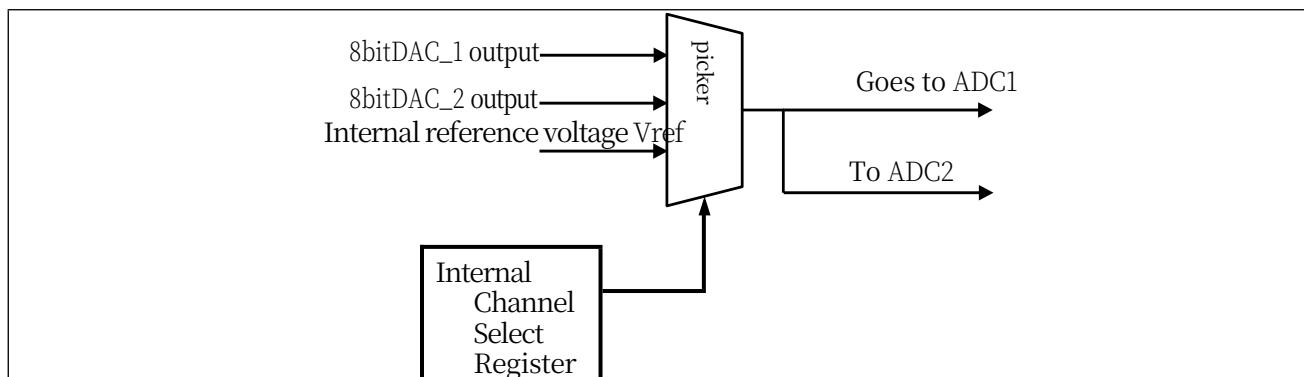


Figure 16-2 Internal Analog Channel Selection

and the description of register PWC_PWCMR in the [Power Control (PWC) chapter for the specific setting method of the internal channel selection register.

Table 16-2 Register Setting Method for Internal Channel Conversion

conversion target	ADC_CHSELR	CMP_RVADC	CMP_DADR	CMP_DACR	PWC_PWCMR
8bitDAC_1 -> ADC1	ADC1_CHSELR A1 or ADC1_CHSELRB 1 bit0 write 1 to select CH16	Write 0x0001 *1	Write CMP_DADR1 means Fixed 8bitDAC_1 input Output voltage value	bit0 write 1 to allow 8bitDAC_1 outputs	-
8bitDAC_2 -> ADC1		Write 0x0002 *1	Write CMP_DADR1 means Constant 8bitDAC_2 input Output voltage value	bit1 write 1 to allow 8bitDAC_2 outputs	-
Internal Reference Voltage vref -> ADC1		Write 0x0010 *1	-	-	Write 0x80 *2
8bitDAC_1 -> ADC2	ADC2_CHSELR A0 or ADC2_CHSELRB 0 bit8 write 1, select CH8	Write 0x0001 *1	Write CMP_DADR1 means Fixed 8bitDAC_1 input Output voltage value	bit0 write 1 to allow 8bitDAC_1 outputs	-
8bitDAC_2 -> ADC2		Write 0x0002 *1	Write CMP_DADR1 means Constant 8bitDAC_2 input Output voltage value	bit1 write 1 to allow 8bitDAC_2 outputs	-
Internal Reference Voltage vref -> ADC2		Write 0x0010 *1	-	-	Write 0x80 *2

Attention:

- CMP_RVADC needs to write 0x5500 before writing the target setting value.
- The PWC_PWCMR register requires that the protection bit PWC_FPRC.FPRCB1 be turned

on first.

Note: Do not select the same channel in Sequences A and B. For channels that do not exist, do not set the corresponding registers and leave them as they are after reset.

CH0 denotes channel 0, and its correspondence to the actual analog input channel ADCx_INy can be freely set via register ADC_CHMUXR. For example, for ADC1, CH00MUX set to 0x0 means CH0 is mapped to ADC1_IN0, and set to 0xf means CH0 is mapped to ADC1_IN15; for ADC2, CH00MUX set to 0x0 means CH0 is mapped to ADC12_IN4, and set to 0xF means invalid mapping. Similarly, CH1~CH15 can be mapped through the corresponding ADC_CHMUXR registers.

16.3.3 Trigger Source Selection

Sequence A and Sequence B select trigger sources independently. Selectable trigger sources include external port ADTRGx, internal events IN_TRGx0, IN_TRGx1. Port ADTRGx falling edge input is active. IN_TRGx0,IN_TRGx1 are set by register **ADC_ITRGSELRO,1**, which can be used to select rich internal event sources. In addition, write register ADC_STR to generate a software trigger signal, which can only be used when the ADC is in standby mode and only applies to sequence A. The software trigger can be used only when the ADC is in standby mode.

16.3.4 Sequence A Single Scan Mode

A/D control register ADC_CR0.MS[1:0] is set to 00b to select Sequence A single scan mode.

In this mode, when the start condition of sequence A selected in register ADC_TRGSR occurs, or when ADC_STR.START bit is triggered by writing 1 to the software, the ADC starts to sample and convert all the channels selected in sequence A channel selection register ADC_CHSELRA, and the results of the conversion are stored in the corresponding data register ADC_DR. ADC_STR.START is kept as 1, when all channels are converted, it will be cleared to 0 automatically, and ADC enters into conversion standby state, waiting for the next trigger condition.

When all channels have finished converting, the Sequence A end-of-conversion flag bit ADC_ISR.EOCAF is set to 1 and the Sequence A end-of-conversion event ADC_EOCA is generated, which can be used to start the DMA. If ADC_ICR.EOCAIEN is 1, an interrupt-permitted state is generated, and the Sequence A end-of-conversion interrupt request is generated at the same time.

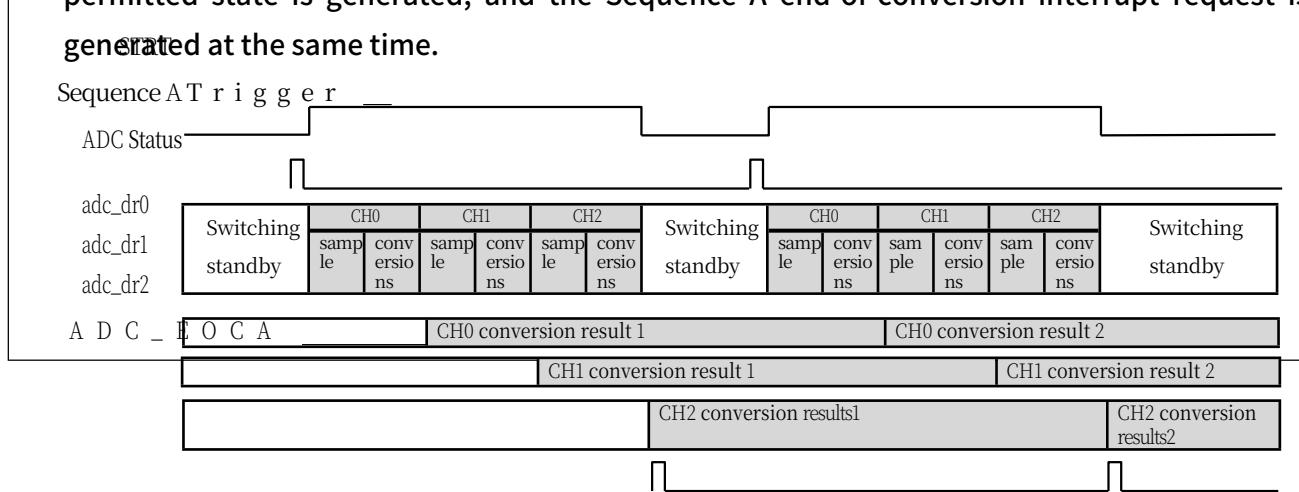


Figure 16-3 Sequence A Single Scan Mode

Sequence A Software flow for single scan mode:

1. Verify that ADC_STR.START is 0 and the ADC is in conversion standby.
2. A/D control register ADC_CR0.MS[1:0] is set to 00b to select Sequence A single scan mode.
3. Sets the Sequence A channel selection register ADC_CHSELRA.
4. Sets the sample time register ADC_SSTR.
5. ADC_STR.START Write 1 software trigger sequence A, or set register ADC_TRGSR to select sequence A trigger condition.
6. Query Sequence A End of Conversion Flag Bit EOCAF.
7. Reads each channel data register ADC_DR.
8. Write 0 clears the EOCAF flag bit in preparation for the next conversion.

The CPU query method in step 6~8 above can also be replaced by interrupt method, using ADC_EOCA interrupt to process the conversion data. Or use ADC_EOCA event to start DMA to

read data.

16.3.5 Sequence A Continuous Scan Mode

A/D control register ADCR0.MS[1:0] is set to 01b to select Sequence A continuous scan mode.

Sequence A Continuous Scan mode is similar to Sequence A Single Scan mode, except that instead of going into conversion standby at the end of the conversion of the used channel, Continuous mode restarts the conversion of Sequence A. The STRT bit is not cleared to 0 automatically.

When it is necessary to stop continuous scanning, write 0 to the STRT bit and read STRT to confirm 0 to determine that the ADC has entered the conversion standby state.

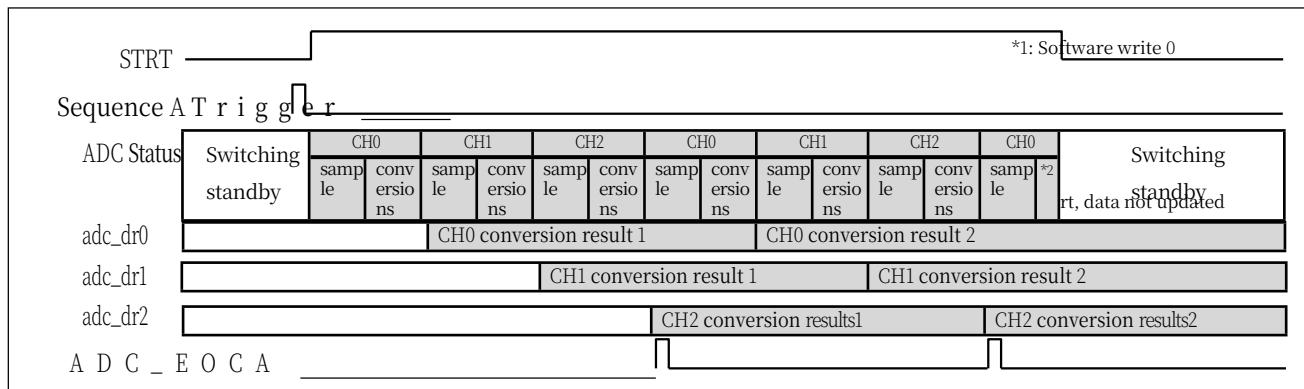


Figure 16-4 Continuous Scan

Sequence A Software flow for continuous scanning mode:

1. Verify that ADC_STR.STRT is 0 and the ADC is in conversion standby.
2. A/D control register ADC_CR0.MS[1:0] is set to 01b to select Sequence A continuous scan mode.
3. Sets the Sequence A channel selection register ADC_CHSELRA.
4. Sets the sample time register ADC_SSTR.
5. ADC_STR.STRT Write 1 software trigger sequence A, or set register ADC_TRGSR to select sequence A trigger condition.
6. Query Sequence A End of conversion flag bit EOCAF.
7. Reads each channel data register ADC_DR.
8. Write 0 clears the EOCAF flag bit in preparation for the next conversion.
9. When there is no need to continue the conversion, write 0 to the STRT bit and read STRT to confirm 0 to judge that the ADC has entered the conversion standby state. The query method in steps 6 to 8 above can also be replaced by interrupt method, using ADC_EOCA interrupt to process the conversion data. Or use ADC_EOCA event to start DMA to read data.

Attention:

----- Due to the continuous conversion,
the interval between each scan is relatively short, especially when only 1

channel is selected for conversion. It is recommended to use ADC_EOCA event to start DMA reading data to avoid data loss due to untimely processing in query mode.

16.3.6 Dual Sequence Scanning Mode

A/D control register ADC_CR0.MS[1:0] is set to 10b or 11b to select the dual sequence scan mode, i.e., both sequence A and sequence B can be initiated by the trigger condition selected by each.

When MS[1:0]=10b, sequences A and B are equivalent to two independent single-scan sequences. MS[1:0]=11b Sequence A is in continuous scan mode and B is in single-scan mode.

Sequence A The trigger source is selected by ADC_TRGSR.TRGSEL_A[2:0] and the channel to be converted is selected by ADC_CHSEL_A. **Sequence B** The trigger source is selected by ADC_TRGSR.TRGSEL_B[2:0] and the converted channel is selected by ADC_CHSEL_B.

When all channels of sequence A are converted to the end, the sequence A end-of-conversion flag bit ADC_ISR.EOCAF is set to 1 and the sequence A end-of-conversion event ADC_EOCA is generated, and if ADC_ISCR.EOCAIEN is 1, an interrupt license is granted, and an end-of-conversion interrupt request is generated for sequence A at the same time. When all channels of Sequence B are converted, the Sequence B end-of-conversion flag bit ADC_ISR.EOCBF is set to 1 and the Sequence B end-of-conversion event ADC_EOCB is generated; if ADC_ISCR.EOCBIEN is 1, the interrupt-authorized state generates the Sequence B end-of-conversion interrupt request at the same time.

In dual sequence scanning mode, when sequence A competes with sequence B, sequence B will be prioritized, i.e., sequence B has a higher priority than sequence A. For details, please refer to the following table.

Table 16-3 Various Competitions for Sequences A and B

A/D conversion	Trigger signal generation	Treatment	
		ADC_CR1.RSCHSEL=0	ADC_CR1.RSCHSEL=1
Sequence A conversion process	Sequence A Trigger	Invalid trigger signal	
	Sequence B trigger	1) Sequence A conversion is interrupted and sequence B conversion is started. 2) After the transformation of sequence B is complete, sequence A is transferred from the The interrupted channel starts to continue the conversion	1) Sequence A conversion is interrupted and sequence B conversion is started 2) After the conversion of sequence B is completed, sequence A is reconverted from the first pass
Sequence	Sequence A	After all channels of Sequence B have been converted, Sequence A conversion	

B Conversion in the process of	Trigger	begins.
	Sequence B Trigger	Invalid trigger signal
Sequence A, B triggered simultaneously in AD idle	Sequence B starts first, and after all channels have been converted, sequence A conversion begins.	

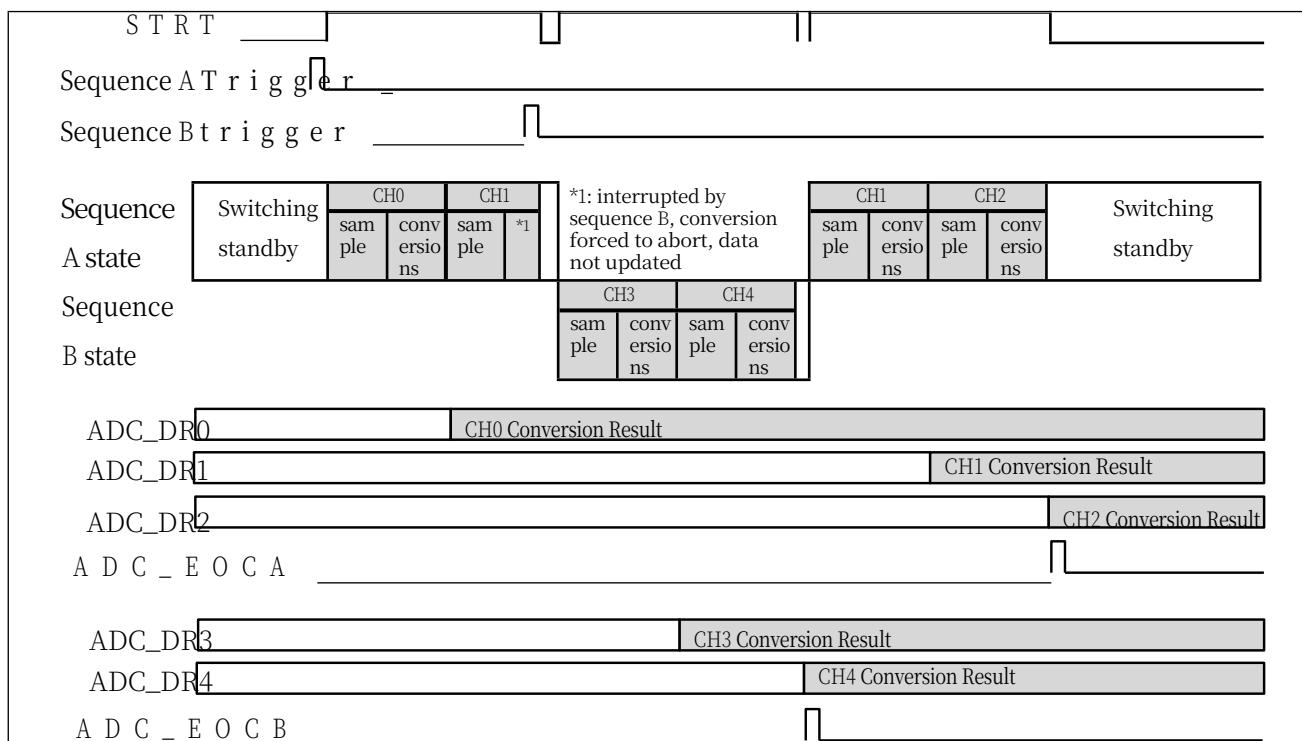


Figure 16-5 Dual Sequence Scan Mode (Sequence A restarts from interrupted channel)

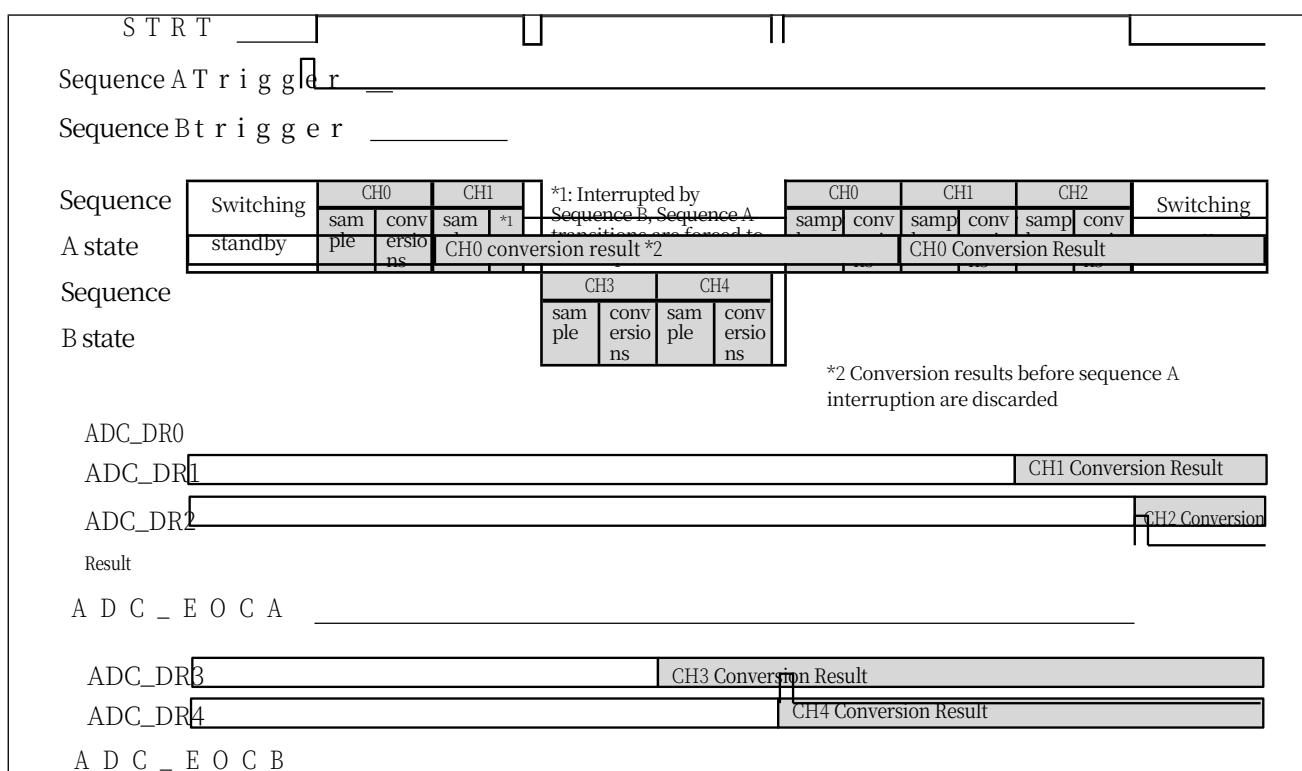


Figure 16-6 Dual Sequence Scan Mode (Sequence A restarted from the first channel)

Software flow for dual sequence scanning mode:

1. Verify that ADC_STR.STRT is 0 and the ADC is in conversion standby.
2. A/D control register ADC_CR0.MS[1:0] is set to 10b or 11b to select dual sequence scan mode.

-
- 3. Setting register ADC_CR1.RSCHSEL selects the startup method after sequence A is interrupted.
 - 4. Sets the Sequence A channel selection register ADC_CHSELRA.
 - 5. Sets the Sequence B channel selection register ADC_CHSELRB.
 - 6. Sets the sample time register ADC_SSTR.

7. Setting register ADC_TRGSR selects sequence A and B trigger conditions.
8. Interrupts by querying EOCAF, EOCBF, or ADC_EOCA, ADC_EOCB, or initiates DMA to process the converted data at the end of a sequence A or B conversion.

Attention:

- Do not select the same channels in Sequences A and B. Do not select the same trigger source for Sequences A and B.

16.3.7 Analog Watchdog Function

The analog watchdog function compares the conversion results at the end of the A/D conversion of a channel, as shown in the figure below, and generates a channel comparison interrupt and event ADC_CHCMP if the conversion result is within the protection area. and generates a sequence comparison interrupt and event ADC_SEQCMP based on the result of the comparison of the channels at the end of the entire sequence scan. you can choose to perform the comparison on any single or multiple channels. Comparison. Multiple channel comparisons are valid, and sequence comparison interrupts and events are generated when any one channel is compared consistently.

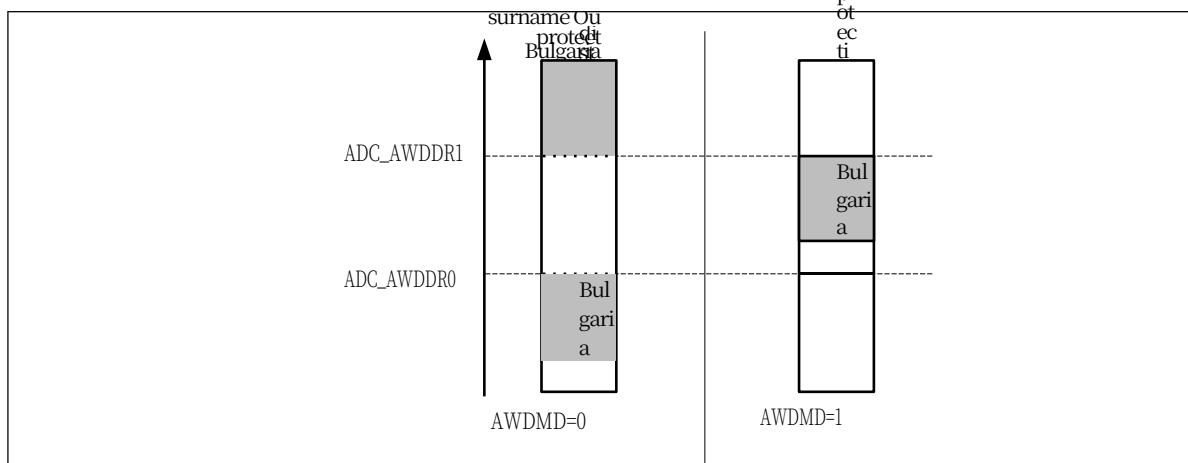


Figure 16-7 Analog Watchdog Protection Area (Comparison Condition)

Software flow using the analog watchdog function:

1. Set threshold registers AD_AWDR0, ADC_AWDR1
2. Set the compare channel register ADC_AWDCHSR to select any single or multiple channels to be compared
3. Set ADC_AWDCR.AWDMD Selection of comparison conditions
4. Setting ADC_AWDCR.AWDSS[1:0] Select sequence comparison interrupt and event output to set ADC_AWDCR.AWDIEN interrupt license bit.
5. Set ADC_AWDCR.AWDEN Allow analog watchdog function
6. According to the previous section, set the scan mode and start the AD for conversion.
7. In the ADC_CHCMP/ADC_SEQCMP interrupt, or after the A/D conversion is finished, query the comparison status register ADC_AWDSR and process the

comparison result accordingly.

16.3.8 Sampling time and conversion time for analog inputs

In single-scan mode, the A/D conversion can be set up by software, internally triggered by IN_TRGx0,1 and externally triggered by the ADTRGx startup pin. The ADC module samples and converts the analog channel after the scanning conversion delay time t_D , and enters the standby state after the end-of-conversion delay time t_{ED} after all conversions have been completed, and a scan is finally completed. The continuous scan mode is similar to the single scan except that there is no t_D time for the second and subsequent starts of the sequence.

The conversion time of a single channel $t_{CONV}=t_{SPL}+t_{CMP}$. 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_SSTRx. t_{CMP} represents the time of the successive comparisons, with 12-bit precision of 13 ADCLKs, 10-bit precision of 11 ADCLKs, and 8-bit precision of 9 ADCLKs.

The time for one scan conversion $t_{SCAN}=t_D+\sum t_{CONV}+t_{ED}$. Where $\sum t_{CONV}$ represents the sum of the conversion times of all scanning channels, and the conversion time t_{CONV} can be different for each channel because the sampling time t_{SPL} can be set independently.

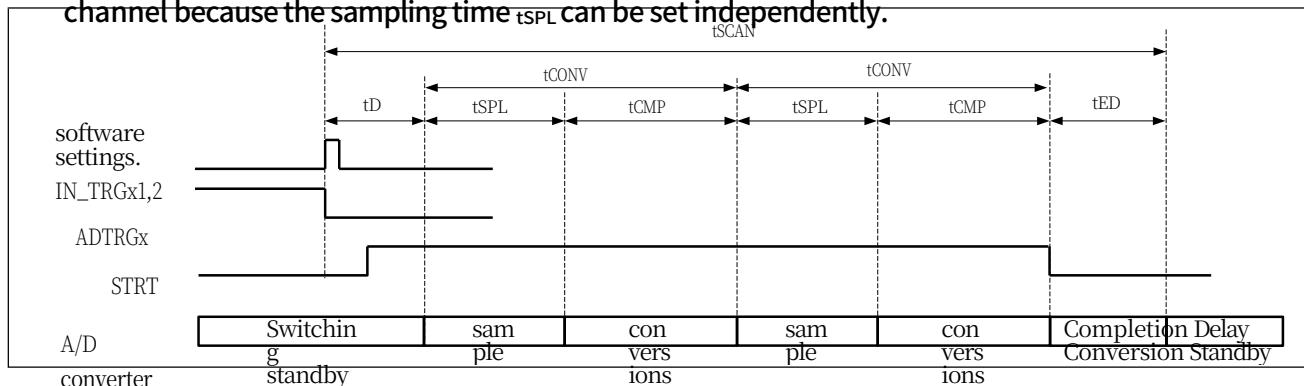


Figure 16-8 A/D

Conversion Times

Table 16-4 AD

Conversion Times

marking	clarification		prerequisite			
			Synchronized Peripheral Trigger	Asynchronous perimeter trigger *Note	External Pin Trigger	software trigger
t_D	Scan start	ADC idle, start conversion	1 PCLK4 + 4 ADCLK	3 PCLK4 + 4 ADCLK + 1 PCLK4_SYNC	3 PCLK4 + 4 ADCLK	4 ADCLK

	proc essi ng time	Sequence A is interrupted in transition, initiating sequence B conversions	2 PCLK4 + 6 ADCLK	4 PCLK4 + 6 ADCLK + 1 PCLK4_SYNC	4 PCLK4 + 6 ADCLK	-			
tCONV	tSPL	sampling time		ADSSTRx.SST[7:0] x ADCLK					
	tCMP	Con versi on time	12-bit resolution	13 ADCLK					
			10-bit resolution	11 ADCLK					
			8-bit resolution	9 ADCLK					
tED		Scan completion processing time	1 PCLK4 + 3 ADCLK						
tTD		Minimum Continuous Trigger Interval	$\Sigma t_{CONV} + 2 PCLK4 + 5 ADCLK$						

Attention:

- Asynchronous peripheral trigger refers to the situation when the ADC module selects the PLL clock action which is asynchronous with the system clock, then the peripheral module clock is asynchronous with the ADC module clock. At this time, the peripheral module clock and ADC module clock are asynchronous, PCLK4_SYNC indicates the original synchronous clock of ADC module, at this time, PCLK4, ADCLK are the same, and they are all asynchronous PLL clock.

16.3.9 A/D data register auto-clear function

When ADC_CR0.CLREN is "1", the A/D conversion data register ADC_DR will be cleared to "0x0000" automatically after it is read by the CPU or DMA.

Use this function to detect if the data register ADC_DR has been updated. The following is an example.

- When ADC_CR0.CLREN is "0" and the auto-clear function is disabled, the A/D conversion result (0x0222) is not updated to the data register ADC_DR for some reason, and the ADC_DR register continues to hold the previous conversion value (0x0111). The unupdated (0x0111) will be read in the A/D conversion completion interrupt processing. In order to detect whether the A/D conversion value has been updated or not, it is necessary to additionally store the previous conversion value into RAM and judge it by comparing the conversion result.
- If ADC_CR0.CLREN is "1", the ADC_DR register will be cleared to "0x0000" automatically after the previous conversion result (0x0111) is read by the CPU or DMA if the auto-clear function is allowed. If the result of the previous conversion (0x0111) is read by the CPU or DMA, the ADC_DR register will be cleared to "0x0000" automatically, and if the conversion result is not correctly transferred to the ADC_DR register after A/D conversion, the ADC_DR register will be kept at "0x0000", and if the interrupt is read out with "0x0000", it will be very easy for the CPU or DMA to clear the ADC_DR register. At this time, if "0x0000" is read in the interrupt processing, it will be easy to determine whether the A/D conversion data is stored correctly.

16.3.10 Converted data averaging calculation function

The A/D conversion averaging function is a function that averages the results of 2, 4, 8, 16, 32, 64, 128 or 256 conversions of the same channel and saves them in the data register. The averaging function makes it possible to remove certain noise components to make the conversion results more accurate.

Register ADC_CR0.AVCNT[2:0] sets the number of consecutive conversions, and register ADC_AVCHSEL selects any one or more channels to be averaged.

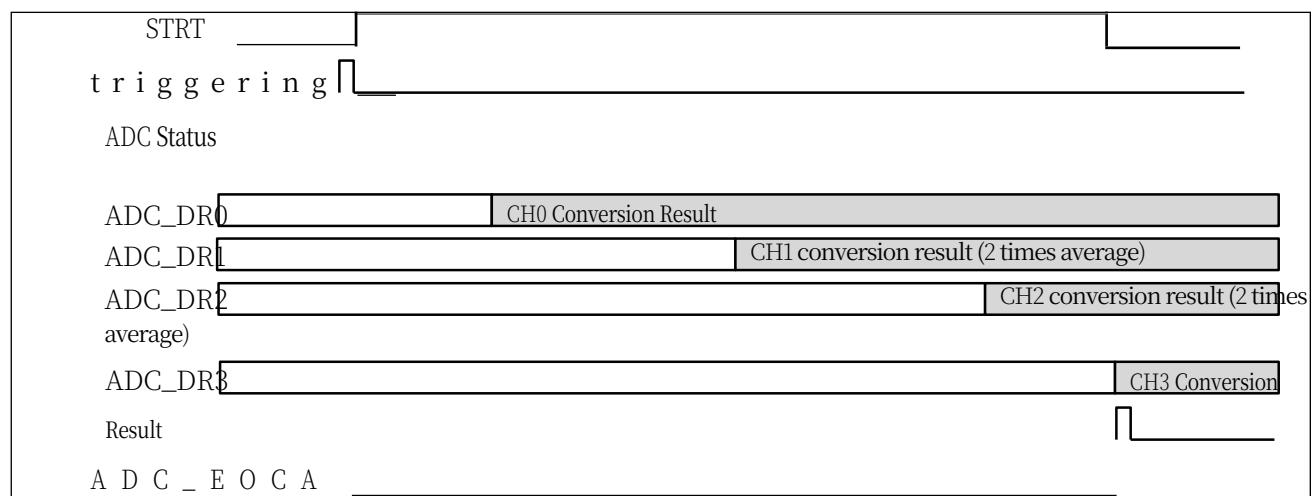
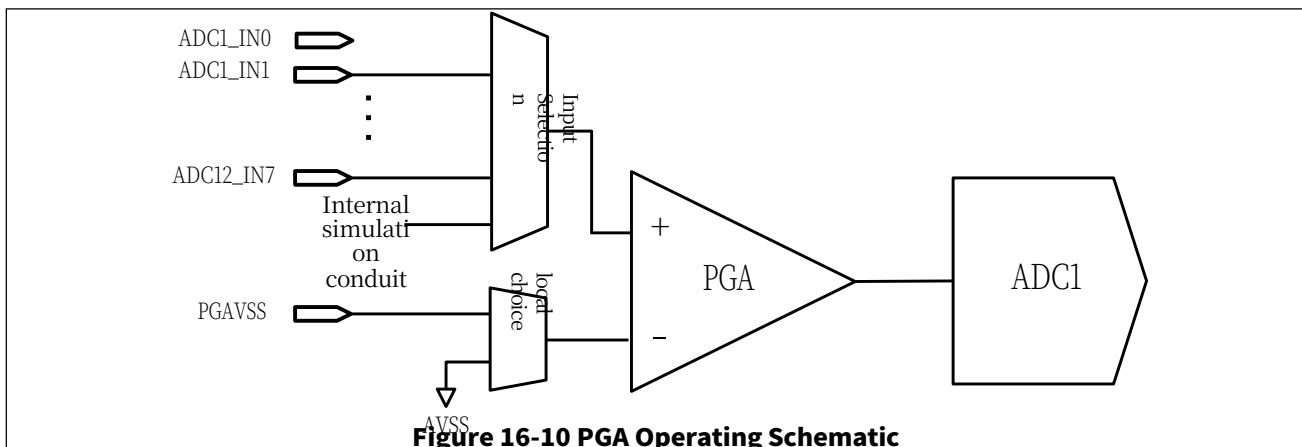


Fig. 16-9 Switching action when the averaging function is active

In Figure 16-9, Sequence A Single Scan Mode is selected to convert four channels, CH0~3, of which CH1 and 2 are set as two times averaging mode. During the scanning process, CH1 and 2 will perform two consecutive conversions and save the averaged results into the data registers ADC_DR1 and 2 of the corresponding channels.

16.3.11 Programmable Gain Amplifier PGA

搭载可编程增益放大器 PGA 时，可以设置寄存器 ADC_PGAINSR 选择 PGA 的输入源，设置寄存器 ADC_PGACR，使 PGA 电路有效，设置寄存器 ADC_PGAGSR 选择增益倍数，增益范围 x2~x32 可选择。此时，模拟输入先经过PGA 电路进行放大，然后再输入到 ADC 模块进行转换。



For example, if you need to convert ADC1_IN2 after amplification, you can set ADC_CHSELRA0 to 0x4 to select CH2 (ADC1_IN2), set ADC_PGAINSR to 0x4 to select ADC1_IN2, and set PGA validity and amplification to start the conversion. Similarly, if you need to amplify the internal analog channel, set ADC_CHSELRA1 to 0x1 to select CH16, set ADC_PGAINSR to 0x100, and then start the conversion.

Attention:

- When the internal analog channel is selected for the PGA input source, the analog is fixed to the 8bitDAC_1 output, independent of the CMP_RVADC setting.
- Only ADC1 supports PGA.

16.3.12 Multi-ADC Co-working Mode

On chips with two or three ADC modules, the ADC Co-operation mode can be used.

In ADC co-operation mode, the conversion of ADC2 and ADC3 is synchronized by the trigger signal of ADC1. That is, the setting of sequence A trigger source selection register ADC_TRGSR.TRGSEL[2:0] of ADC2 and ADC3 is invalid. All ADC modules are triggered by the trigger source selected in the Sequence A Trigger Source Selection Register of ADC1. A write of 1 to the ADC_STR.START register in this mode does not initiate a conversion, i.e., software startup is disabled.

When using the Cooperative mode, disable the Sequence B action to avoid disrupting the synchronization.

You can set two ADC modules, ADC1 and ADC2, to work together, or three ADC modules, ADC1, ADC2 and ADC3, to work together. Depending on the product specifications, ADC3 may not be equipped.

The ADC can be configured to work in the following four cooperative modes:

- Single Parallel Trigger Mode
- Single delayed trigger mode
- Cyclic Parallel Trigger Mode
- or cyclic delayed trigger mode

Single Parallel Trigger Mode

The Sequence A trigger condition for ADC1 triggers all ADC modules in cooperative mode simultaneously and only once.

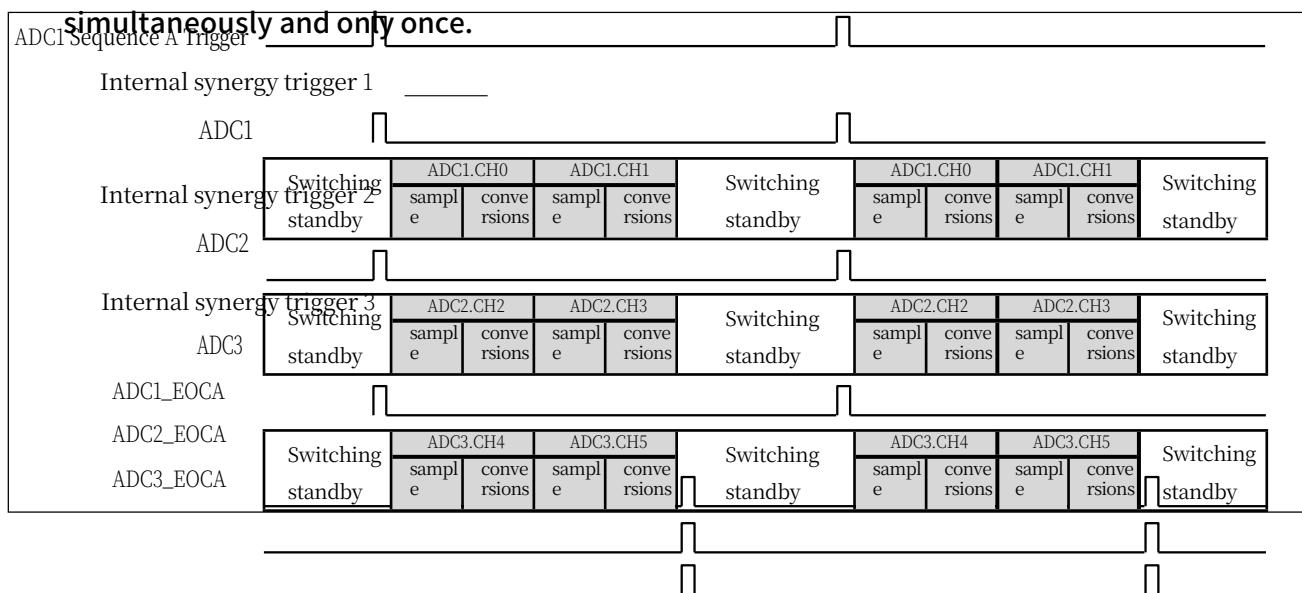


Figure 16-11 Single Parallel Trigger Mode (Triple ADC)

Attention:

ion:

-Prohibit multiple ADCs from converting the same analog input at the same time. Multiple ADCs are prohibited from converting the same analog input at the same time, and an analog channel can only sample one ADC module at the same time, same below.

The software setting process for this mode is as follows:

1. Co-working license register ADC_SYNCCR.SYNCEN Write 0 to confirm that the co-working is invalid.
2. Setting up the ADC1 Module
 - a) Verify that ADC1_STR.STRT is 0 and ADC1 is in conversion standby.
 - b) Set control register ADC1_CR0.MS[1:0] to 00b: Sequence A single scan mode,

- c) Sets the sequence A channel selection register ADC1_CHSELRA
- d) Setting the Sample Time Register ADC1_SSTR
- e) Set Sequence A Trigger Source Selection Register ADC1_TRGSR

3. Setting up the ADC2 Module

- Verify that ADC2_STR.STRT is 0 and ADC1 is in conversion standby.
- Setup Control Register ADC2_CR0.MS[1:0], Channel Select Register ADC2_CHSELRA, Channel Sample Time Register ADC2_SSTR.

Attention:

- In order to ensure the synchronized operation of ADC2 and ADC1, the above registers should be set to the same values as those 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 the same.

4. Setting up the ADC3 module (when three ADCs are working together)

- Verify that ADC3_STR.STRT is 0 and ADC2 is in conversion standby.
- Setup Control Register ADC3_CR0.MS[1:0], Channel Select Register ADC3_CHSELRA, Channel Sample Time Register ADC3_SSTR.

Attention:

- Same as ADC2, to ensure synchronized operation of ADC3 and ADC1, the above registers should be set to the same values as those of ADC1 as far as possible.

5. Set the Co-op Mode Control Register ADC_SYNCCR.SYNCMD[2:0], write 010b: ADC1, ADC2 two ADCs work together. Or write 011b: ADC1, ADC2, ADC3 Three ADCs work together.

6. COOPERATION LICENSE REGISTER ADC_SYNCCR.SYNCEN Write 1 to make co-operation valid.

7. Wait for ADC1 sequence A trigger source input to process the result after ADC1, ADC2, and ADC3 have completed the conversion.

Single delayed trigger mode

After the sequence A trigger condition of ADC1 triggers ADC1, ADC2 is triggered to start the conversion after a set delay, and ADC3 is triggered to start the conversion after a set delay,

and each ADC module is triggered only once.

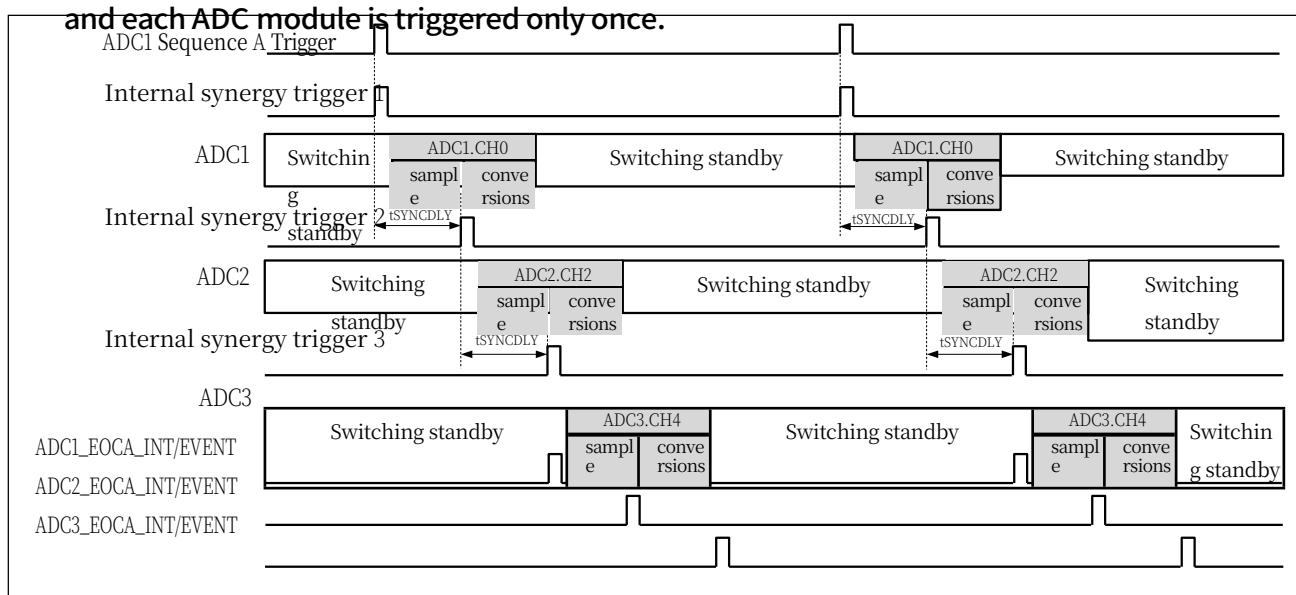


Figure 16-12 Single Delay Trigger Mode (Triple ADC)

Attention:

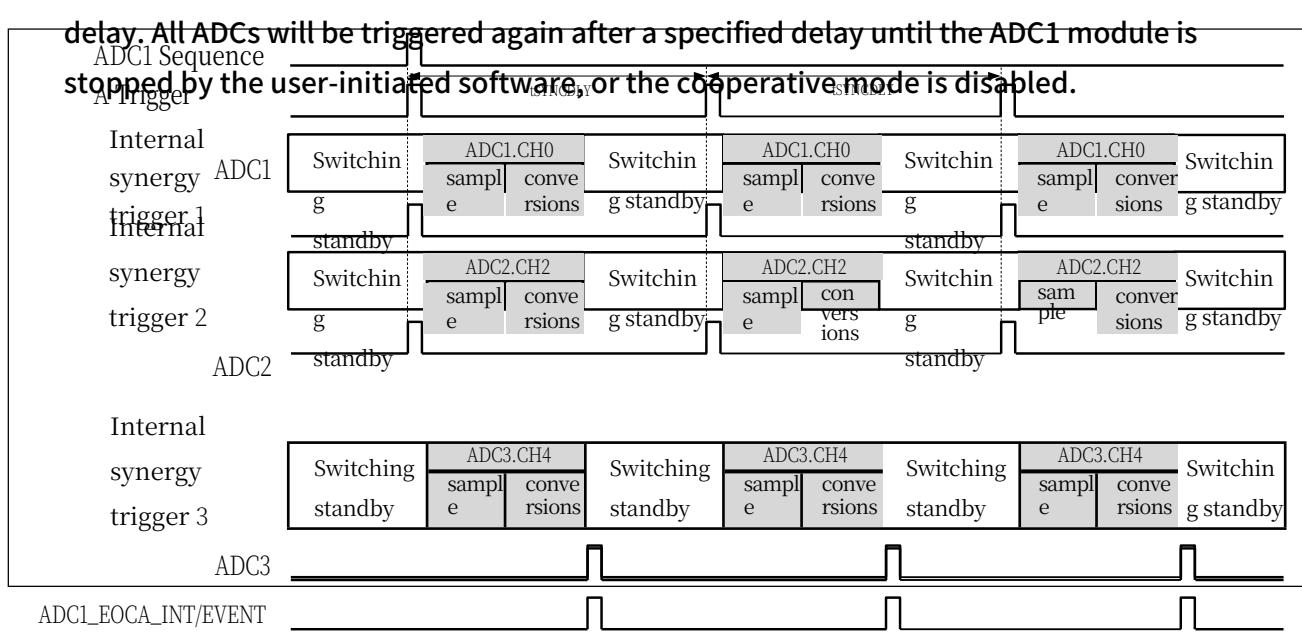
- ADC1 Sequence A Trigger After the first input of ADC1 Sequence A Trigger and before the ADC3 Co-Trigger occurs, another input of ADC1 Sequence A Trigger will be ignored.
- If each ADC unit converts the same analog channel, the sampling time needs to be staggered, i.e., the delay time $t_{SYNCDLY}$ and the channel mining sample time t_{SPL} need to satisfy: $t_{SYNCDLY} > t_{SPL}$.

The software setting process for this mode is as follows:

1. Co-working license register ADC_SYNCCR.SYNCEN Write 0 to confirm that the co-working is invalid.
2. Setting up the ADC1, ADC2, ADC3 modules (refer to Single Parallel Mode)
3. Set the Co-Mode Control Register ADC_SYNCCR.SYNCDLY[7:0] to set the startup delay of both ADCs.
4. Set Co-op Mode Control Register ADC_SYNCCR.SYNCMD[2:0], write 000b: ADC1, ADC2 two ADCs work together. Or write 001b: ADC1, ADC2, ADC3 three ADCs work together.
5. Co-working License Register ADC_SYNCCR.SYNCEN Write 1, co-working is valid.
6. Wait for ADC1 sequence A trigger source input to process the result after ADC1, ADC2, and ADC3 have completed the conversion.

Cyclic Parallel Trigger Mode

The Sequence A trigger condition of ADC1 triggers all ADC modules in cooperative mode at the same time, and then triggers all ADC modules again at the same time after a specified delay. All ADCs will be triggered again after a specified delay until the ADC1 module is stopped by the user-initiated software, or the cooperative mode is disabled.



Attention:

- The delay time $t_{SYNCDLY}$ and the time of one scan conversion t_{SCAN}

should satisfy: $t_{SYNCDLY} > t_{SCAN}$. The software setting procedure of this mode is as

follows:

1. Co-working license register ADC_SYNCCR.SYNCEN Write 0 to confirm that the co-working is invalid.
2. Set up ADC1, ADC2, and ADC3 modules to refer to single parallel mode.
adc_cr0.ms[1:0] is set to 00b:

Sequence A Single Scan Mode

3. Set the Co-Mode Control Register ADC_SYNCCR.SYNCMD[2:0], write 110b: ADC1, ADC2 two ADCs work together. Or write 111b: ADC1, ADC2, ADC3 three ADCs work together.
4. Set the Co-op Mode Control Register ADC_SYNCCR.SYNCMD[2:0], write 110b: ADC1, ADC2 two ADCs work together. Or write 111b: ADC1, ADC2, ADC3 three ADCs work together.
5. Co-working License Register ADC_SYNCCR.SYNCEN Write 1, co-working is valid.
6. Wait for ADC1 sequence A trigger source input to process the result after ADC1, ADC2, and ADC3 have completed the conversion.

Cyclic Delayed Trigger Mode

After the ADC1 is triggered by the sequence A trigger condition of ADC1, it will trigger ADC2, ADC3, ADC1, ADC2... in turn, after each set delay, until the ADC1 module is stopped by the user-initiated software or the co-operation mode is disabled. ADC2, ADC3, ADC1, ADC2..., until user-initiated software stops the ADC1 module or disables the cooperative mode of operation.

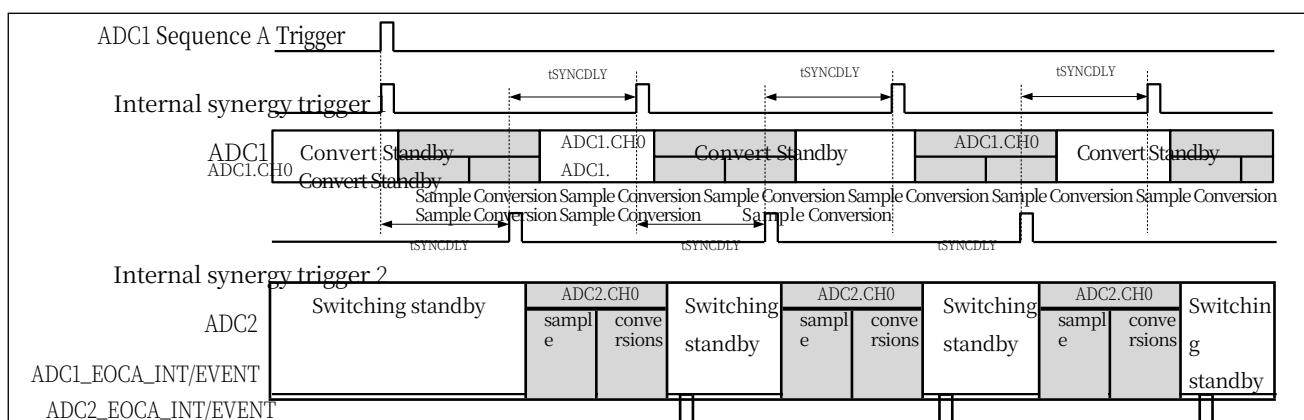
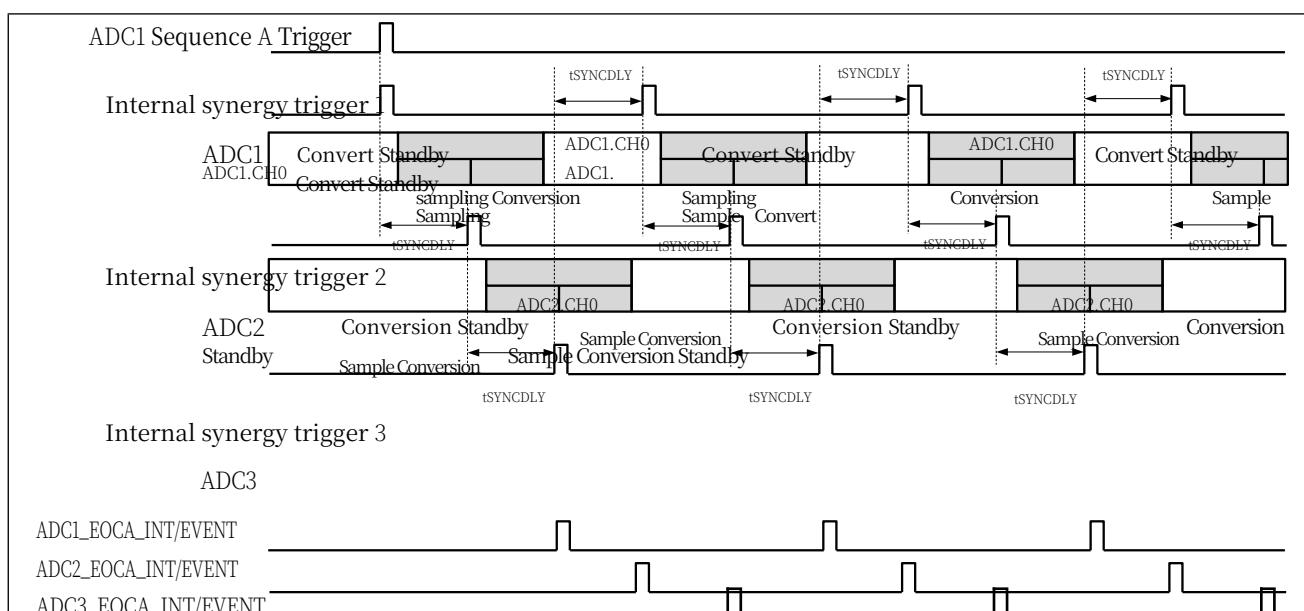


Figure 16-14 Cyclic Delay Trigger Mode (Two ADCs)



Attention:**ion:**

- When two ADCs work together, the delay time $t_{SYNCDLY}$ and the time of one scanning conversion t_{SCAN} need to satisfy $t_{SYNCDLY} > t_{SCAN}/2$. When three ADCs work together, they need to satisfy: $t_{SYNCDLY} > t_{SCAN}/3$. At the same time, if ADC1, ADC2, and ADC3 are converting the same analog channel, the sample time needs to be staggered, i.e., $t_{SYNCDLY} > t_{SPL}$. At the same time, if ADC1, ADC2 and ADC3 are converting the same analog channel, the sampling times need to be staggered, i.e., $t_{SYNCDLY} > t_{SPL}$.

The software setting process for this mode is as follows:

1. Co-working license register ADC_SYNCCR.SYNCEN Write 0 to confirm that the co-working is invalid.
2. Set up the ADC1, ADC2, and ADC3 modules to refer to cyclic parallel trigger mode.
3. Set the Co-Mode Control Register ADC_SYNCCR.SYNCDLY[7:0] to set the delay for each trigger.
4. Set Co-op Mode Control Register ADC_SYNCCR.SYNCMD[2:0], write 100b: ADC1, ADC2 two ADCs work together. Or write 101b: ADC1, ADC2, ADC3 three ADCs work together.
5. COOPERATION LICENSE REGISTER ADC_SYNCCR.SYNCEN Write 1 to make co-operation valid.
6. Wait for ADC1 sequence A trigger source input to process the result after ADC1, ADC2, and ADC3 have completed the conversion.

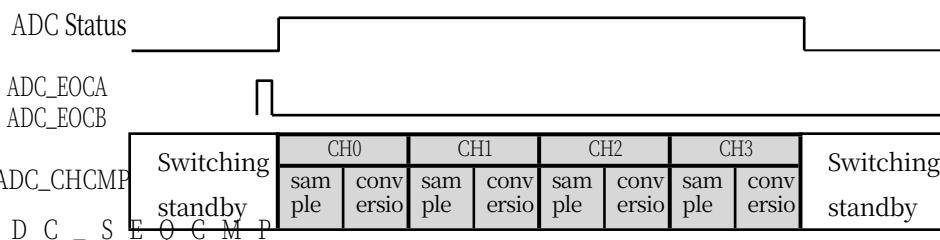
16.3.13 Interrupt and event signal output

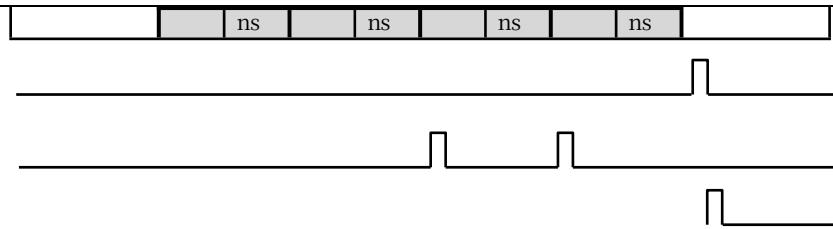
The ADC module can generate the following four types of event outputs. When each event occurs, an interrupt request is output at the same time if the corresponding interrupt license register is set to valid.

1. Sequence A End of scan ADC_EOCA, corresponding to interrupt license register ADC_ICR.EOCAIEN
2. Sequence B End of Scan ADC_EOCB, corresponding to interrupt license register ADC_ICR.EOCBIEN
3. Channel compare ADC_CHCMP, corresponding to interrupt license register ADC_AWDCR.AWDIEN
4. Sequence compare ADC_SEQCMP, corresponding to interrupt license register ADC_AWDCR.AWDIEN

The above four event outputs can activate other on-chip peripheral modules, including DMA transfer. The DMA transfer can be used to read the A/D conversion result continuously without software intervention, which is completely realized by hardware and reduces the load on the CPU. ~~please refer to the DMA description chapter for the DMA setting. The event signal output has nothing to do with the control of the interrupt enable bit, and will be output as soon as the condition occurs.~~ SRTI

Sequence A or B trigger _____



**Figure 16-16 ADC Interrupt and Event Output Timing**

In the above figure, the comparison conditions for the CH1, CH2 watchdogs are satisfied. The channel compare event occurs at the end of each channel transition, and the sequence compare event occurs at the end of the sequence scan, one cycle later than the sequence scan event ADC_EOCA/B.

16.4 Register Description

List of registers

Unit 1 BASE_ADDR: 0x4004_0000

Unit 2 BASE_ADDR: 0x4004_0400

Table 16-5 ADC Register List 1/2

register name	notation	offset address	bit width	reset value
A/D startup register	ADC_STR	0x00	8	0x00
A/D control register 0	ADC_CR0	0x02	16	0x0000
A/D control register 1	ADC_CR1	0x04	16	0x0000
A/D conversion start trigger register	ADC_TRGSR	0x0A	16	0x0000
A/D channel selection register A0	ADC_CHSELRA0	0x0C	16	0x0000
A/D channel selection register A1	ADC_CHSELRA1	0x0E	16	0x0000
A/D channel selection register B0	ADC_CHSELRB0	0x10	16	0x0000
A/D channel selection register B1	ADC_CHSELRB1	0x12	16	0x0000
A/D average channel selection register 0	ADC_AVCHSELR0	0x14	16	0x0000
A/D average channel selection register 1	ADC_AVCHSELR1	0x16	16	0x0000
A/D Sample Period Register	ADC_SSTRx	0x20+x	8	0x0B
	ADC_SSTRL	0x30	8	0x0B
A/D Channel Mapping Control Register 0	CHMUXR0	0x38	16	0x3210
A/D channel mapping control register 1	CHMUXR1	0x3A	16	0x7654
A/D Channel Mapping Control Register 2	CHMUXR2	0x3C	16	0xBA98
A/D channel mapping control register 3	CHMUXR3	0x3E	16	0xFEDC
A/D Interrupt Status Register	ADC_ISR	0x46	8	0x00
A/D interrupt license register	ADC_ICR	0x47	8	0x03
A/D Co-Mode Control Register	ADC_SYNCCR	0x4C	16	0x0C00
A/D Data Register	ADC_DRx	0x50+2*x	16	0x0000
Analog Watchdog Control Register	ADC_AWDCR	0xA0	16	0x0000
Analog Watchdog Threshold Register	ADC_AWDDR0	0xA4	16	0x0000
	ADC_AWDDR1	0xA6	16	0x0000
Analog Watchdog Compare Channel Selection Registers	ADC_AWDCHSR0	0xAC	16	0x0000
Analog Watchdog Compare Channel Select Register 1	ADC_AWDCHSR1	0xAE	16	0x0000
Analog Watchdog Status Register	ADC_AWDSR0	0xB0	16	0x0000
Analog Watchdog Status Register 1	ADC_AWDSR1	0xB2	16	0x0000
A/D Programmable Gain Amplifier Control Registers	ADC_PGACR	0xC0	16	0x0000
A/D programmable gain multiplier registers	ADC_PGAGSR	0xC2	16	0x0000

A/D Programmable Gain Amplifier Input Selection Registers	ADC_PGAINSRO	0xCC	16	0x0000
---	--------------	------	----	--------

register name	notation	offset address	bit width	reset value
A/D Programmable Gain Amplifier Input Selection Register 1	ADC_PGAINSR1	0xCE	16	0x0000

16.4.1 A/D startup register ADC_STR

Reset value: 0x00

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

classifier for honorific people	marking	celebrity	functionality	R/W
b7-b1	-	-	0 when reading, 0 when writing	R/W
			0: Stop conversion 1: Start conversion Place a "1" condition: (1) Software Settings (2) Selected trigger conditions occur (3) In A/D conversion Clear "0" condition: (1) Software Clear "0" (2) Automatic clearing of "0" at the end of conversion Attention: -Write 1 when STRT is 0 (ADC idle) generates a software trigger to start sequence A -Write 1 is invalid when STRT is 1 (in ADC action). -Writing 0 when STRT is 1 indicates that the AD conversion is forced to stop. If ADC_TRGSR is set to a value other than 0x0 and Do not want the ADC to restart, set ADC_TRGSR to 0 before writing 0 to STRT. -Write 0 is invalid when STRT is 0.	
b0	STRT	AD conversion started		R/W

16.4.2 A/D 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]

bitmarker	celebrity	functionality	R/W	
b15- b11	-	0 when reading, 0 when writing	R/W	
b10-b8	AVCNT[2:0]	Count Selection	R/W	
		0 0 0: 2 consecutive conversions averaged 0 0 1: average of 4 0 1 0: average of 8 consecutive conversions 0 1 1: average of 16 consecutive conversions 1 0 0: 32 consecutive conversions averaged 1 0 1: average of 64 consecutive conversions 1 1 0: average of 128 consecutive conversions 1 1 1: average of 256		
b7	DFMT	Data format right-justified	R/W	
		consecutive conversions 0: Convert data 1: Convert data left-aligned 0: Automatic clearing prohibited		
b6	CLREN	Data register auto-clear	R/W	
		1: Automatic clearance of licenses Note: After the CLREN bit is set, register ADC_DRx will be cleared automatically after it is read by the CPU, DMA, etc. The auto-clear function is mainly used to detect if the data registers are updated.		
b5-b4	ACCSEL[1:0]	Resolution Selection bit resolution	0 1: 10- R/W	
		0 0: 12-bit resolution 1 0: 8-bit resolution 1 1: Setting prohibitions		
b3-b2	-	-0 when reading, 0 writing	R/W when	
b1-b0	MS[1:0]	Mode Selection	0 0: Sequence A single scan mode, Sequence B invalid scan mode 0 1: Sequence A continuous scanning mode, Sequence B invalid 1 1: Sequence A continuous scan mode, Sequence B single scan mode	R/W

Attention:

ion:

ADC_STR.SRT is "0".

16.4.3 A/D Control Register 1 ADC_CR1

Reset value: 0x0000

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

classifier for honorific people	marking	celebrity	functionality	R/W
b15-b3	-	-	0 when reading, 0 when writing	R/W
b2	RSCHSEL	Sequence A restart channel selection	0: After being interrupted by Sequence B, Sequence A restarts to continue scanning from the interrupted channel 1: After being interrupted by Sequence B, Sequence A restarts with a rescan from the first channel it was interrupted by.	R/W
b1-b0	-	-	-0 when reading, 0	when writing
		R/W		

Attention:

-Set this register when ADC_STR.
ADC_STR.STRT is "0". Set this register when

16.4.4 A/D conversion start trigger register ADC_TRGSR

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8
TRGENB	-	-	-	-	TRGSELB [2]	TRGSELB [1]	TRGSELB[0]
b7	b6	b5	b4	b3	b2	b1	b0
TRGENA	-	-	-	-	TRGSEL A [2]	TRGSEL A [1]	TRGSEL A[0]

clas sifie rfor hon orifi c peo ple	mar king	celebri ty	functionality	R/W
b15	TRGENB	Sequence B trigger enable	0: Sequence B on-chip or external pin-trigger inhibit 1: Sequence B on-chip or external pin-triggered license Note: Selecting external pin trigger is valid. If ADTRGx changes from "High" to "Low" and a falling edge is detected, scanning conversion starts, please keep "Low" for 1.5*PCLK4 cycles. Please keep "Low" for more than 1.5*PCLK4 cycles.	R/W
b14-b11 -			-0 when reading, 0	when writing
		R/W	In Sequence B valid mode (ADC_CR0.MS[1]=1), as a trigger condition for Sequence B 000b: ADTRGx 001b: IN_TRGx0 010b: IN_TRGx1 011b: IN_TRGx0 + IN_TRGx1 Other: not selected x=1, 2 NOTE: Valid only in Sequence B Valid Mode. Other mode settings are not valid. The interval between the two triggers must be greater than or equal to the scanning period tSCAN, if less then the trigger is no effect.	
b10-b8	TRGSELB[2:0]	Sequence B trigger condition selection	0: Sequence A on-chip or external pin-trigger inhibit 1: Sequence A On-chip or External Pin Trigger License Note: Selecting external pin trigger is valid. If ADTRGx changes from "High" to "Low" and a falling edge is detected, scanning conversion starts, please keep "Low" for 1.5*PCLK4 cycles. Please keep "Low" for more than 1.5*PCLK4 cycles.	R/W
b7	TRGENA	Sequence A Trigger Enable	0 when reading, 0 when writing	R/W
b6-b3	-	-	Trigger condition for sequence A. 000b: ADTRGx 001b: IN_TRGx0 010b: IN_TRGx1	R/W

b2-b0	TRGSELA [2:0]	Sequence A Trigger Condition Selection	Other: not selected	R/W
			n=1, 2	
			Attention:	
			ADC_STR.STRT write 1 software trigger, ignoring the TRGENA, or TRGSELA[2:0] settings, and starting the A/D conversion directly.	

The interval between the two triggers must be greater than or equal to the scan period

Atten

tSCAN, if less then the trigger is invalid.

ion:

-STRT This register is set when ADC_STR.STRT is "0".

16.4.5 A/D Channel Select Register A ADC_CHSELRA0

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8
CHSEL A [15:8]							
b7	b6	b5	b4	b3	b2	b1	b0
CHSEL A [7:0]							

Bit Flag	Bit Name	Function	R/W	Notes
b15-b0	CHSEL A[15:0]	Conversion channel selection	R/W	Channel selection for sequence A. Each bit represents a channel, and CHSEL A[x] represents channel CHx, which can be selected in any combination. 0: Corresponding channel not selected 1: Select the corresponding channel The corresponding bit for a non-existent channel is the RESERVED bit, which is 0 when read and 0 when written. Note: Do not select the same channel in sequence A and sequence B again.

Attention:

-Set this register when ADC_STR. Set this register when ADC_STR.STRT is "0".

16.4.6 A/D channel selection register A 1 ADC_CHSELRA1

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8
CHSEL A [31:24]							
b7	b6	b5	b4	b3	b2	b1	b0
CHSEL A [23:16]							

Bit Flag	Bit Name	Function	R/W	Notes
b15-b0	CHSEL A[31:16]	Conversion channel selection	R/W	Channel selection for sequence A. Each bit represents a channel, and CHSEL A[x] represents channel CHx, which can be selected in any combination. 0: Corresponding channel not selected 1: Select the corresponding channel The corresponding bit for a non-existent channel is the RESERVED bit, which is 0 when read and 0 when written. Note: Do not select the same channel in sequence A and sequence B again.

Attention:

-Set this register when ADC_STR. Set this register when ADC_STR.STRT is "0".

16.4.7 A/D Channel Select Register B ADC_CHSELRB0

Reset value: 0x0000

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
Channel selection for sequence B. Each bit represents a channel, and CHSELB[x] represents channel CHx, which can be selected in any combination. Only valid in dual sequence scan mode.			
b15-b0	CHSELB[15:0]	Conversion channel selection	0: Corresponding channel not selected 1: Select the corresponding channel The corresponding bit for a non-existent channel is the RESERVED bit, which is 0 when read and 0 when written. Note: Do not select the same channel in sequence A and sequence B again.

Attention:

-Set this register when ADC_STR. Set this register when ADC_STR.STRT is "0".

16.4.8 A/D Channel Selection Register B1 ADC_CHSELRB1

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8
CHSELB [31:24]							
b7	b6	b5	b4	b3	b2	b1	b0
CHSELB [23:16]							

Bit Flag	Bit Name	Function	R/W
Channel selection for sequence B. Each bit represents a channel, and CHSELB[x] represents channel CHx, which can be selected in any combination. Only valid in dual sequence scan mode.			
b15-b0	CHSELB[31:16]	Conversion channel selection	0: Corresponding channel not selected 1: Select the corresponding channel The corresponding bit for a non-existent channel is the RESERVED bit, which is 0 when read and 0 when written. Note: Do not select the same channel in sequence A and sequence B again.

Attention:

-Set this register when ADC_STR. Set this register when ADC_STR.STRT is "0".

16.4.9 A/D average channel selection register ADC_AVCHSELR0

Reset value: 0x0000

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	
b15-b0	AVCHSEL[15:0] Average channel selection		Each bit represents a channel and AVCHSEL[x] represents channel CHx, which can be selected in any combination. 0: corresponding channel is not selected 1: Select the corresponding channel The corresponding bit for a non-existent channel is the RESERVED bit, which is 0 when read and 0 when written. Note: When AVCHSEL is selected at the same time as the channel corresponding to ADC_CHSELRA or ADC_CHSELB, the channel will perform the set number of A/D conversions consecutively during scanning, and the conversion results will be averaged and updated into the data registers. If the corresponding channel AVCHSEL is not set, the channel performs a normal one-time conversion.
Attention:			-Set this register when ADC_STR. Set this register when ADC_STR.STRT is "0".

16.4.10 A/D Average Channel Selection Register 1 ADC_AVCHSELR1

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8
AVCHSEL [31:24]							
b7	b6	b5	b4	b3	b2	b1	b0
AVCHSEL [23:16]							

Bit Flag	Bit Name Function	R/W	
b15-b0	ADAVSEL[31:16] Average channel selection		Each bit represents a channel and AVCHSEL[x] represents channel CHx, which can be selected in any combination. 0: corresponding channel is not selected 1: Select the corresponding channel The corresponding bit for a non-existent channel is the RESERVED bit, which is 0 when read and 0 when written. Note: When AVCHSEL is selected at the same time as the channel corresponding to ADC_CHSELRA or ADC_CHSELB, the channel will perform the set number of A/D conversions consecutively during scanning, and the conversion results will be averaged and updated into the data registers. If the corresponding channel AVCHSEL is not set, the channel performs a normal one-time conversion.
Attention:			-Set this register when ADC_STR. Set this register when ADC_STR.STRT is "0".

16.4.11 A/D Sampling Status Register ADC_SSTR

Reset value: 0x0B

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

classifier for honorific people	marking	celebrity	functionality		R/W
The number of sampling cycles can be set from 5 to 255 cycles.					
Channels CH0~15 are set by ADC_SSTRx, x=0~15, and other channels are set by ADC_SSTRL.					
b7-b0	SST[7:0]	Number of sampling periods	Adjust the sampling time. Do not sample for less than 5 cycles.		R/W
$SST \geq (R_{AIN} + R_{ADC}) * C_{ADC} * \ln(2^{N+2}) * f_{ADC} + 1$ <p>Where: R_{AIN} denotes the external input impedance (Ω), R_{ADC} denotes the internal sampling switching resistance (Ω), C_{ADC} denotes the internal sampling sample and hold capacitance (F), N denotes AD resolution (12/10/8), and f_{ADC} denotes ADCLK frequency (Hz). Specific Refer to the description related to electrical characteristics.</p>					

Attention:

-Set this register when ADC_STR. Set this register when
ADC_STR.STRT is "0".

16.4.12 A/D 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 bits of the non-existing channel are 0 when read and 0			
when written The mapping relationship between the setpoint and the different ADC units CHx is as follows:			
setpoint	ADC1 Mapping Objects	ADC2 Mapping Objects	
0x0	ADC1_IN0	ADC12_IN4	
0x1	ADC1_IN1	ADC12_IN5	
0x2	ADC1_IN2	ADC12_IN6	
0x3	ADC1_IN3	ADC12_IN7	
0x4	ADC12_IN4	ADC12_IN8	
0x5	ADC12_IN5	ADC12_IN9	
0x6	ADC12_IN6	ADC12_IN10	
0x7	ADC12_IN7	ADC12_IN11	
0x8	ADC12_IN8	Internal analog channels	R/W
0x9	ADC12_IN9	-	
0xa	ADC12_IN10	-	
0xb	ADC12_IN11	-	
0xc	ADC1_IN12	-	
0xd	ADC1_IN13	-	
0xe	ADC1_IN14	-	
0xf	ADC1_IN15	-	

Note: Do not set to an analog input that does not exist.

Attention:

-Set this register when ADC_STR. Set this register when ADC_STR.STRT is "0".

16.4.13 A/D Interrupt Status Register ADC_ISR

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	EOCBF	EOCAF

classifier for honorific people	marking	celebrity	functionality	R/W
b7-b2	-	-	0 when reading, 0 when writing	R/W
b1	EOCBF	Sequence B conversion completion flag	Set 1 when all channels selected in sequence B are scanned.	R/W
b0	EOCAF	Sequence A Conversion Completion Flag	When the register is set and needs to be cleared, read "1" and then write "O". Set 1 when all channels selected in sequence A have been scanned. When a register is set and needs to be cleared, read "1" and then write "0".	R/W

16.4.14 A/D interrupt license register ADC_ICR

Reset value: 0x03

b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	EOCBIEN	EOCAIEN

classifier for honorific people	marking	celebrity	functionality	R/W
b7-b2	-	-	0 when reading, 0 when writing	R/W
b1	EOCBIEN	Sequence B conversion completion interrupt enable	0: Sequence B conversion complete interrupt disable 1: Sequence B conversion complete interrupt license	R/W
b0	EOCAIEN	Sequence A conversion completion interrupt enable	0: Sequence A conversion complete interrupt disable 1: Sequence A conversion completion interrupt license	R/W

16.4.15 A/D Synergy Mode Control Register ADC_SYNCCR

Reset value: 0x0C00

b15	b14	b13	b12	b11	b10	b9	b8
SYNCMDLY[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	SYNCMDLY[7:0]	Synchronization Delay Time Set as follows: Single delayed trigger mode: $t_{SYNCMDLY} > t_{SPL}$ Two ADC cyclic delayed trigger mode: $t_{SYNCMDLY} > t_{SPL}$ and $t_{SYNCMDLY} > t_{SCAN}/2$ Triple ADC cyclic delayed trigger mode: $t_{SYNCMDLY} > t_{SPL}$ and $t_{SYNCMDLY} > t_{SCAN}/3$ Single parallel trigger mode: this register setting is invalid. Cyclic Parallel Trigger Mode: $t_{SYNCMDLY} > t_{SCAN}$		R/W	
b7	-	-0 when reading, 0 when writing SYNCMD [2] 0: Single trigger 1: Cyclic 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, if there is no ADC3, this bit is forbidden to be set to 1. Note: Set this register when SYNCEN is "0". When using single trigger, set the ADC to be synchronized to "0". Set to Sequence A Single Scan, or Sequence A Continuous Scan mode. When using cyclic trigger mode, set the ADC to Sequence A Single Scan or Sequence A Continuous Scan mode. Column A Single Scan Mode.		R/W	
b3-b1	-	-0 when reading, 0 when writing 0: Synchronization mode invalid 1: Synchronized mode effective Attention:		R/W	
b0	SYNCEN	Synchronization Mode License Synchronization mode is supported only in sequence A. Before writing 1 to SYNCEN, turn off sequence B for 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		R/W	

ADC_SSTRx. to avoid inconsistencies in the scan time t_{SCAN} across ADCs, causing subsequent synchronization failures.

SYNCEN is automatically cleared to 0 when the software writes 0 to ADC1_STR.START to force the conversion to stop.

Attention:

- This register is carried in ADC1 only, and is not available in ADC2.

16.4.16 A/D Data Register ADC_DR

ADC_DRx (ADC1 x=0~16, ADC2 x=0~8)

Channel x Data

Register The ADC_DR register is a read-only register used to store the A/D conversion data for each channel. The reset value is 0x0000 Depending on the data alignment and conversion resolution, the conversion result data is stored differently.

Data right-aligned-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]														

Data right-aligned-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]														

Data right-aligned - 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-aligned-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	0	0	0	0

Data left-aligned-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	0	0

Data left-aligned - 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

16.4.17 Analog Watchdog Control Register ADC_AWDCR

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8
-	-	-	-	-	-	-	AWDIEN
b7	b6	b5	b4	b3	b2	b1	b0
AWDSS [1]	AWDSS[0]	-	AWDMD	-	-	-	AWDEN

classifier for honorific people	marking	celebrity	functionality	R/W
b15-b9	-	-	0 when reading, 0 when writing	R/W
b8	AWDIEN	Watchdog interrupt enable	0: Disable ADC_CHCMP, ADC_SEQCMP interrupts 1: ADC_CHCMP, ADC_SEQCMP interrupts allowed	R/W
			Note: This register does not affect ADC_CHCMP, ADC_SEQCMP event outputs	
			00: ADC_SEQCMP interrupt/event is output at the completion of both Sequence A and Sequence B scans. 01: Output ADC_SEQCMP interrupt/event on completion of scan in sequence A, not in sequence B 10: Output ADC_SEQCMP interrupt/event when Sequence B scanning is complete, not Sequence A 11: Same as 00	
b7-b6	AWDSS[1:0] Watchdog Sequence Selection			R/W
			Note: The channel watchdog interrupt/event ADC_CHCMP is not controlled by this register and is output normally at the end of each channel conversion based on the comparison result. Setting 1 of the ADC_AWDSR comparison status registers for each channel is not controlled by this register.	
b5	-	-0 when reading, 0 writing		when
b4	AWDMD	Watchdog Compare Mode	0: Comparison condition is satisfied when AWDDR0 > conversion result, or conversion result > AWDDR1 1: Comparison condition is satisfied when AWDDR0 ≤ conversion result ≤ AWDDR1 When the compare condition is satisfied, output ADC_CHCMP event, if interrupt allow AWDIEN=1 then output the middle R/W break at the same time. At the completion of sequence scanning, if one or more channels in this sequence satisfy the comparison condition and AWDSS[1:0] Allow this sequence, then output ADC_SEQCMP event, if interrupt allowed AWDIEN=1 then output interrupt at the same time.	
b3-b1	-	-0 when reading, 0 writing		when
b0	AWDEN	Watchdog compare function enabled Watchdog compare function disabled	0: 1: Watchdog comparison function is valid	R/W

16.4.18 Analog Watchdog Threshold Register ADC_AWDDR0, ADC_AWDDR1

Reset values: ADC_AWDDR0=0x0000, ADC_AWDDR1=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	ADDR[15:0]	Compare Data	Compare Data
		R/W	

AWDDR0 sets the low threshold and AWDDR1 sets the high threshold. Data can also be written to registers during A/D conversion to realize the dynamic threshold comparison function.

AWDDR0 and AWDDR1 have different resolutions (12-bit, 10-bit, or 8-bit) depending on the alignment (right-aligned or left-aligned data).

- Data right-aligned-12-bit resolution Low 12-bit [11:0] available
- Data right-aligned-10-bit resolution Lower 10 bits [9:0] available
- Data right-aligned - 8-bit resolution Lower 8 bits [7:0] available
- Data left-aligned-12-bit resolution High 12-bit [15:4] available
- Data left-aligned-10-bit resolution High 10-bit [15:6] available
- Data **left-aligned** - 8-bit resolution
High 8-bit [15:8] available when the
multiple averaging function is active and only the
last average value is compared.

16.4.19 Analog Watchdog Compare Channel Select Register ADC_AWDCHSR0

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8
AWDCH [15:8]							
b7	b6	b5	b4	b3	b2	b1	b0
AWDCH[7:0]							

Bit Flag	Bit Name	Function	R/W
b15-b0	AWDCH[15:0]	Watchdog Compare Function Channel Selection	R/W
		Each bit corresponds to one channel each. The bit corresponding to a non-existent channel is a RESERVED bit, which is 0 when read and 0 when written. Note: This is only valid if the corresponding channel is selected in Sequence A or B scan, i.e., the corresponding bit of ADC_CHSELRA or ADC_CHSELRB in the channel selection register is "1".	

Attention:

-Set this register when ADC_STR. Set this register when ADC_STR.STRT is "0" and ADC_AWDCR.AWDEN is "0".

16.4.20 Analog Watchdog Compare Channel Select Register 1 ADC_AWDCHSR1

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8
AWDCH [31:24]							
b7	b6	b5	b4	b3	b2	b1	b0
AWDCH [23:16]							

Bit Flag	Bit Name	Function	R/W
b15-b0	AWDCH[31:16]	Watchdog compare function Capable of channel selection	R/W
		Each bit corresponds to one channel each. The bit corresponding to a non-existent channel is a RESERVED bit, which is 0 when read and 0 when written. Note: This is only valid if the corresponding channel is selected in Sequence A or B scan, i.e., the corresponding bit of the channel selection register ADC_CHSELRA or ADC_CHSELRB is "1".	

Attention:

-Set this register when ADC_STR. Set this register when ADC_STR.STRT is "0" and ADC_AWDCR.AWDEN is "0".

16.4.21 Analog Watchdog Status Register ADC_AWDSR0

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8
AWDF [15:8]							
b7	b6	b5	b4	b3	b2	b1	b0
AWDF [7:0]							

clas	markin	celebrit	function	R/W
sifie	g	y	ality	
rfor				
hon				
orifi				
c				
peo				
ple				

b15-b0	AWDF[15:0]	Watchdog Comparator	0: The comparison condition does not hold 1: Comparative conditions hold Note: Each bit corresponds to one channel each. The bit corresponding to a non-existent channel is a RESERVED bit, which is 0 when read and 0 when written. When a register is set and needs to be cleared, read "1" and then write "0".	R/W
--------	------------	---------------------	---	-----

16.4.22 Analog Watchdog Status Register 1 ADC_AWDSR1

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8
AWDF [31:24]							
b7	b6	b5	b4	b3	b2	b1	b0
AWDF [23:16]							

Bit Flag	Bit Name	Function	R/W	
b15-b0	AWDF[31:16]	Watchdog Comparator	0: The comparison condition does not hold 1: Comparative conditions hold Attention: Each bit corresponds to one channel each. The bit corresponding to a non-existent channel is a RESERVED bit, which is 0 when read and 0 when written. When a register is set and needs to be cleared, read "1" and then write "0".	R/W

16.4.23 A/D Programmable Gain Amplifier Control Register ADC_PGACR

Reset value: 0x0000

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

classifier for	marking	celebrity	functionality	R/W
honorable				
people				
b15-b12	-	-	0 when reading, 0 when writing	R/W
b3~b0	PGACTL[3:0]	amplifier control	0000: Amplifier not valid 1110: Amplifier is active, signal is amplified according to ADC_PGAGSR.GAIN[3:0] set value Note: Setting a value other than the above is prohibited.	R/W

16.4.24 A/D Programmable Gain Multiplier Register ADC_PGAGSR

Reset value: 0x0000

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

classifier	marking	celebrity	functionality	R/W
for				
honorable				
people				
b15-b4	-	-	0 when reading, 0 when writing 0 0 0 1: x 2.133 0 0 1 0: x 2.286 0 0 1 1: x 2.667 0 1 0 0: x 2.909 0 1 0 1: x 3.2 0 1 1 0: x 3.556 0 1 1 1: x 4.000	R/W
b3-b0	GAIN[3]	01	Amplifier gain setting 1 0 0 0: x 4.571 1 0 0 1: x 5.333 1 0 1 0: x 6.4 1 0 1 1: x 8 1 1 0 0: x 10.667 1 1 0 1: x 16 1 1 1 0: x 32 Note: Setting of other values is prohibited.	R/W

16.4.25 A/D Programmable Gain Amplifier Input Selection Register

ADC_PGAINSR0

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8
-	-	-	-	-	-	-	PGAINSEL [8]
b7	b6	b5	b4	b3	b2	b1	b0
PGAINSEL [7:0]							

classifier for honorific people	mark ing	celebrity	functionality	R/W
b15-b9	-	-	0 when reading, 0 when writing	R/W
b8~b0	PGAINSEL[8:0]	Amplifier Analog Input Selection	0x000: Input not selected 0x001: ADC1_IN0 0x002: ADC1_IN1 0x004: ADC1_IN2 0x008: ADC1_IN3 0x010: ADC12_IN4 0x020: ADC12_IN5 0x040: ADC12_IN6 0x080: ADC12_IN7 0x100: Internal analog channel (8bitDAC_1 output) Other: Disable setting	R/W

16.4.26 A/D Programmable Gain Amplifier Input Selection Register 1

ADC_PGAINSR1

Reset value: 0x0000

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

classifier for honorific people	marking	celebrity	functionality	R/W
b15-b1	-	-	0 when reading, 0 when writing	R/W
b0	PGAVSSEN	Amplifier Ground Cutout Control Selection	0: Use external port PGAVSS as PGA negative phase input 1: Use the internal analog ground AVSS as the PGA negative phase input	R/W

16.5 Precautions for use

16.5.1 Precautions when reading data registers

A/D Data Register ADC_DR Please access in half-word unit. Do not access the data register in byte units.

16.5.2 Notes on Scan Completion Interrupt Handling

When two consecutive scanning conversions are performed for the same channel, the second conversion data overwrites the first conversion data if the CPU does not read the first conversion data in time during the period from the first conversion completion interrupt processing to the second conversion completion interrupt processing.

16.5.3 Notes on module stop and low power setting

Setting register PWC_FCG3 can set the ADC module to stop and reduce the power consumption, and the initial state of ADC is stop. The ADC is initially stopped, so if you want the ADC module to work, please set the corresponding bit of PWC_FCG3 register to cancel the stopping and wait for 1us before starting the ADC conversion.

Before setting the module to stop, make sure that the A/D is in conversion stop, i.e., the ADC_STR.START bit is "0". Before setting the system to stop mode (STOP), set the ADC to module stop mode.

Please refer to the Low Power Consumption Description section for details.

16.5.4 Pin setting of analog channel input for A/D conversion

When a chip pin is set to A/D analog channel input, please disable the digital function of the corresponding pin (PCRxy.DDIS) first. Refer to the GPIO chapter.

16.5.5 noise control

To prevent abnormal voltages such as surges from damaging the analog input pins, it is recommended to use the protection circuit shown in the **Electrical Characteristics** section of the **datasheet**.

17 Temperature Sensors (OTS)

17.1 summary

The On-chip Temperature Sensor (hereinafter referred to as OTS) acquires the temperature inside the chip to support reliable operation of the system. The OTS provides a set of digital quantities related to the temperature, which can be calculated to obtain the temperature value. The OTS can be turned off when not in use by the module stop function to reduce system power consumption.

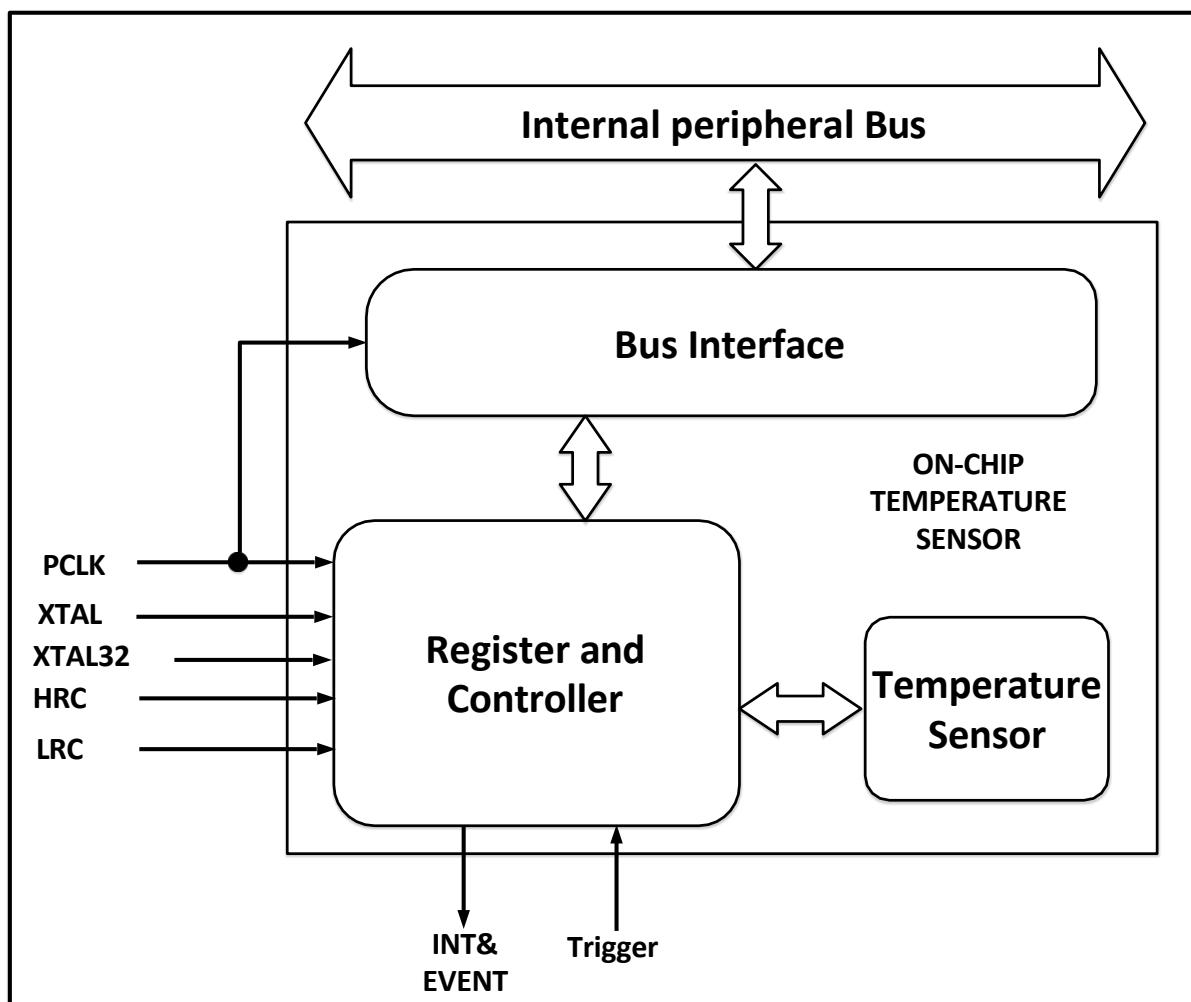


Figure 17-1 OTS Function Block Diagram

17.2 Instructions for use

Before using the OTS to obtain the internal temperature of the chip, turn off the module stop function and start the internal low-speed clock LRC, and select 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 situation.

OTSST is set to 1 to start the temperature measurement, and the OTSST bit is automatically cleared to 0 when the temperature measurement is completed. Therefore, after confirming that OTSST is 0, read the temperature parameters in registers OTS_DR1,2 and OTS_ECR, and use the following formula to find the temperature value.

$$T = K \times (1/D1 - 1/D2) \times E_{hrc} + M$$

[Description of
parameters]

T:

Temperature

(°C)

K: temperature slope (determined by calibration experiments)

D1: Temperature parameter 1 (read from

register OTS_DR1) D2: Temperature

parameter 2 (read from register OTS_DR2)

E_{hrc} : HRC frequency error compensation amount

(read from register OTS_ECR) M: temperature

offset amount (determined by calibration
experiment)

[Calibration experiment]

Calibration experiments were performed at

two defined temperatures to calculate K

and $M = (T2 - T1) / (A2 - A1)$

$M = T1 - K \times A1 = T2 - K \times A2$

T1: Experimental temperature 1

T2: Experimental temperature 2

$A1 = (1/D1T1 - 1/D2T1) \times E_{hrcT1}$

$D1T1, D1T1, E_{hrcT1}$ are D1, D2, and E_{hrc} measured at temperature T1; $A2 = (1/D1T2 - 1/D2T2) \times E_{hrcT2}$

$D1T2, D1T2, E_{hrcT2}$ are D1, D2, E_{hrc} ; measured at temperature T2.

Register OTS_CTL.OTSCK is used to select the temperature measurement clock. When HRC action is selected, the frequency error may affect the accuracy of the final calculated temperature. To eliminate this error, start the XTAL32 before the temperature measurement and use Ehrc when calculating the temperature. set Ehrc to constant 1 when selecting the XTAL action clock.

TSSTP is used to select whether or not to turn off the analog temperature sensor after a temperature measurement is completed. an initial value of 0 indicates that the analog temperature sensor will be turned on after a temperature measurement is completed, so that the stabilization time when the analog temperature sensor is turned off to on will automatically be skipped during the next temperature measurement. To turn off the analog temperature sensor after each temperature measurement, set TSSTP to 1.

Temperature measurement can be triggered by other peripheral events, please set the trigger target of the trigger source to OTS. When the temperature measurement is completed, it will also generate an event to trigger the start of other peripheral devices, please set the register OTS_TRGSEL to select the trigger target. When using the temperature measurement completion interrupt, please set the register OTS_CTL.OTSIE to 1.

17.3 Register Description

Base address: 0x4004_A400

Table 17-1 OTS Register List

register name	notation	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

17.3.1 OTS Control Register (OTS_CTL)

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	TSS TP	OTS IE	OTS CK	OTS ST

Reset value: 0x0000

classifier for honorific people	marking	celebrity	functionality	fill out or in (informati on on a form)
b15~b4	Reserved	-	Read 0 when reading, please write 0 when writing	R
b3	TSSTP	Turn off the analog temperature sensor	Select whether the analog temperature sensor is automatically switched off at the end of the temperature measurement. 0: Analog temperature sensor not turned off 1: Turn off the analog temperature sensor	R/W
b2	OTSIE	Interrupt enable bit	0: Disable end of temperature measurement interrupt request 1: Allow end-of-temperature 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: Starting temperature measureme nt Setting "1" condition: (1) Software "1" (2) Hardware triggered "1" clear "0" condition: (1) Software Clear "0" (2) Automatic clearing of "0" at the end of the temperature	measurem ent.

17.3.2 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

classifierfor marking Bit Name Function Read/Write
honorable
people

b15~b0 TSDC [15:0] Temperature data 1 R
Automatically updated when temperature measurement is completed. Make sure OTS_CTL.OTSST is "0" before reading.

17.3.3 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

classifierfor marking Bit Name Function Read/Write
honorable
people

b15~b0 TSDC [15:0] Temperature data 2 R
Automatically updated when temperature measurement is completed. Make sure OTS_CTL.OTSST is "0" before reading.

17.3.4 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

classifierfor marking Bit Name Function Read/Write
honorable
people

b15~b0 TSEC [15:0] Error Coefficient Error factor Ehrc R
Automatically updated when temperature measurement is completed. Make sure OTS_CTL.OTSST is "0" before reading.

18 Advanced Control Timer (Timer6)

18.1 brief

The Advanced Control Timer 6 (Timer6) is a 16-bit count width high-performance timer, which can be used to count and generate different forms of clock waveforms for external use. The timer supports triangle and sawtooth waveform modes, and can generate various PWM waveforms; software synchronized counting and hardware synchronized counting can be realized between units; caching function is supported in each reference value register; 2-phase quadrature encoding and 3-phase quadrature encoding are supported; and EMB control is supported. This series of products is equipped with 3 units of Timer6.

18.2 basic block diagram

The basic functions and features of Timer6 are shown in Table 18-1.

Table 18-1 Basic Functions and Characteristics of Timer6

Waveform Mode	Sawtooth wave (incremental and decremental counting) triangle wave (incremental and decremental counting)
Basic Functions	- Capture Input
	- software synchronization
	- hardware synchronization
	- caching function
	- Pulse Width Measurement
	- Cycle Measurement
	- orthogonal code counting
	- Universal PWM output
	- EMB control
interrupt output	Count Compare Match Interrupt
	Count Cycle Matching Interrupt
	Dead Time Error Interrupt
event output	Count Compare Match Event
	Counting Cycle Matching Events

The basic block diagram of Timer6 is shown in Figure 18-1. The "<t>" shown in the block diagram indicates the unit number, i.e., "<t>" is 1~3, and all references to "<t>" later in this chapter refer to the unit number. later in this chapter refer to the unit number and will not be repeated.

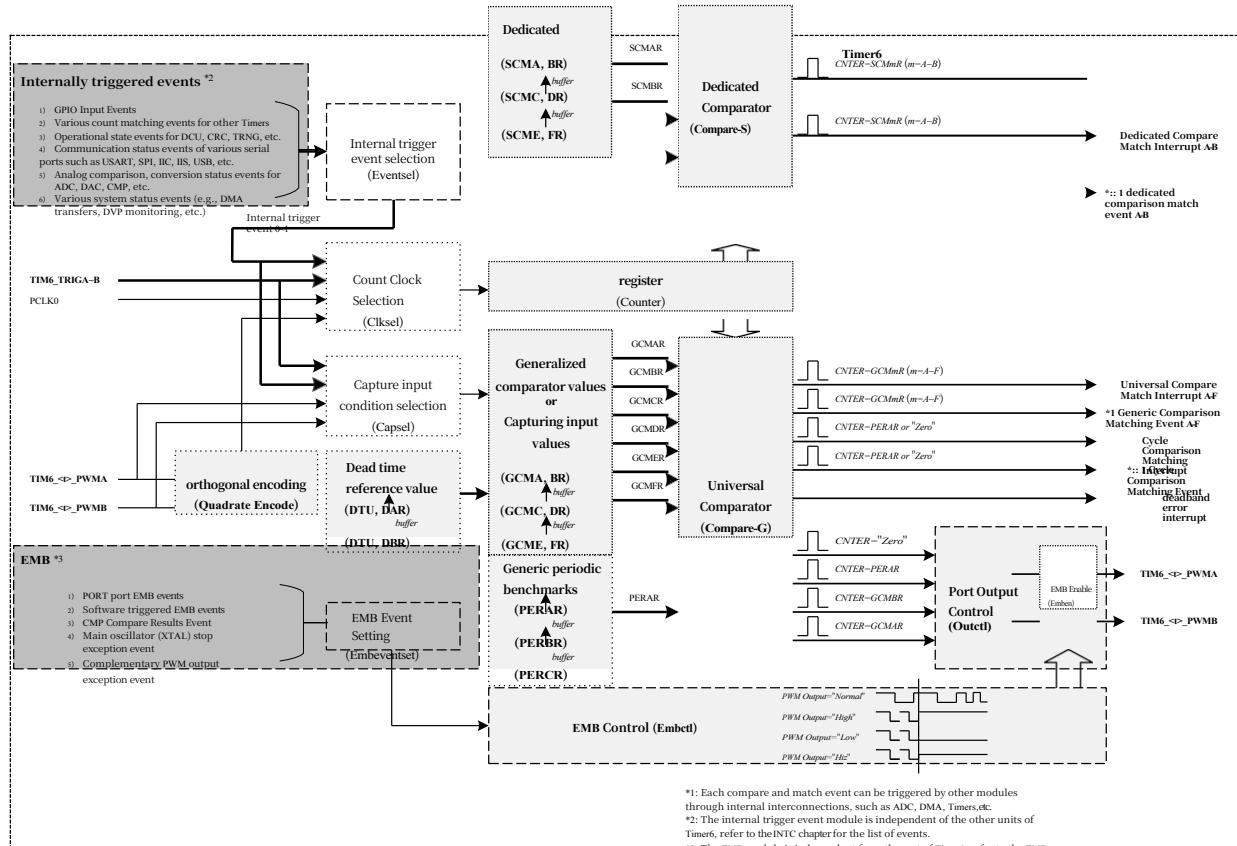


Figure 18-1 Timer6 Basic Block Diagram

Table 18-2 shows the list of input and output ports of Timer6.

Table 18-2 Timer6 Port List

port name	orientations	functionality
TIM6_<t>_PWMA	in or out	1) Quadrature encoded counting clock input port or capture input port or compare output port
TIM6_<t>_PWMB		2) Hardware start, stop, and clear condition input ports
TIM6_TRIGA	in	1) Hardware counting clock input port or capture input port
TIM6_TRIGB		2) Hardware start, stop, and clear condition input ports

18.3 Functional Description

18.3.1 basic movement

18.3.1.1 Waveform Mode

Timer6 has 2 basic counting waveform modes, sawtooth waveform mode and triangle waveform mode. The triangle waveform modes are divided into triangle A mode and triangle B mode due to different internal counting actions. The basic waveforms of Sawtooth and Triangle are shown in Figure 18-2 and Figure 18-3.

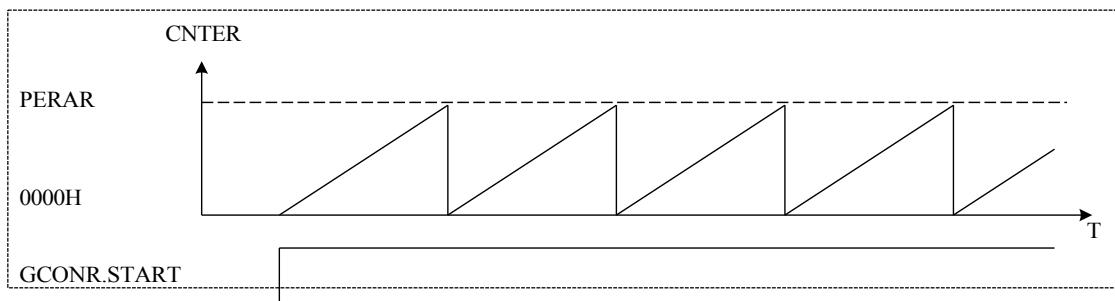


Figure 18-2 Sawtooth Waveform (Incremental Counting)

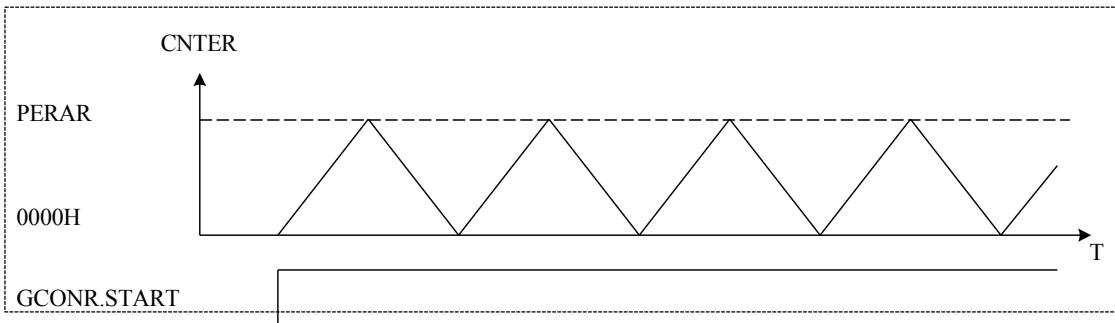


Figure 18-3 Triangle Waveform

18.3.1.2 Comparison Output

Timer6 of each cell has two comparison output ports (TIM6_<t>_PWMA, TIM6_<t>_PWMB) which can output the specified level when the count value matches with the comparison reference value. GCMAR and GCMBR registers correspond to the count comparison reference value of TIM6_<t>_PWMA, TIM6_<t>_PWMB, and GCMAR and GCMBR registers respectively. GCMAR and GCMBR registers correspond to the count comparison reference value of TIM6_<t>_PWMA and TIM6_<t>_PWMB respectively. When the count value of timer is equal to GCMAR, TIM6_<t>_PWMA port outputs the specified level; when the count value of timer is equal to GCMBR, TIM6_<t>_PWMB port outputs the specified level.

The levels of the TIM6_<t>_PWMA and TIM6_<t>_PWMB ports at the start of counting, at the stop of counting, at the match of counting comparison, and at the match of counting period can be determined by the levels of PCONR.STACA, PCONR.STPCA, PCONR.STASTPSA, PCONR.CMPCA[1:0], PCONR.CMPCA[1:0], PCONR. STPCA, PCONR.STASTPSA, PCONR.CMPCA[1:0],

PCONR.PERCA[1:0] bits are set. Fig. 18-4 shows an example of comparison output

operation.

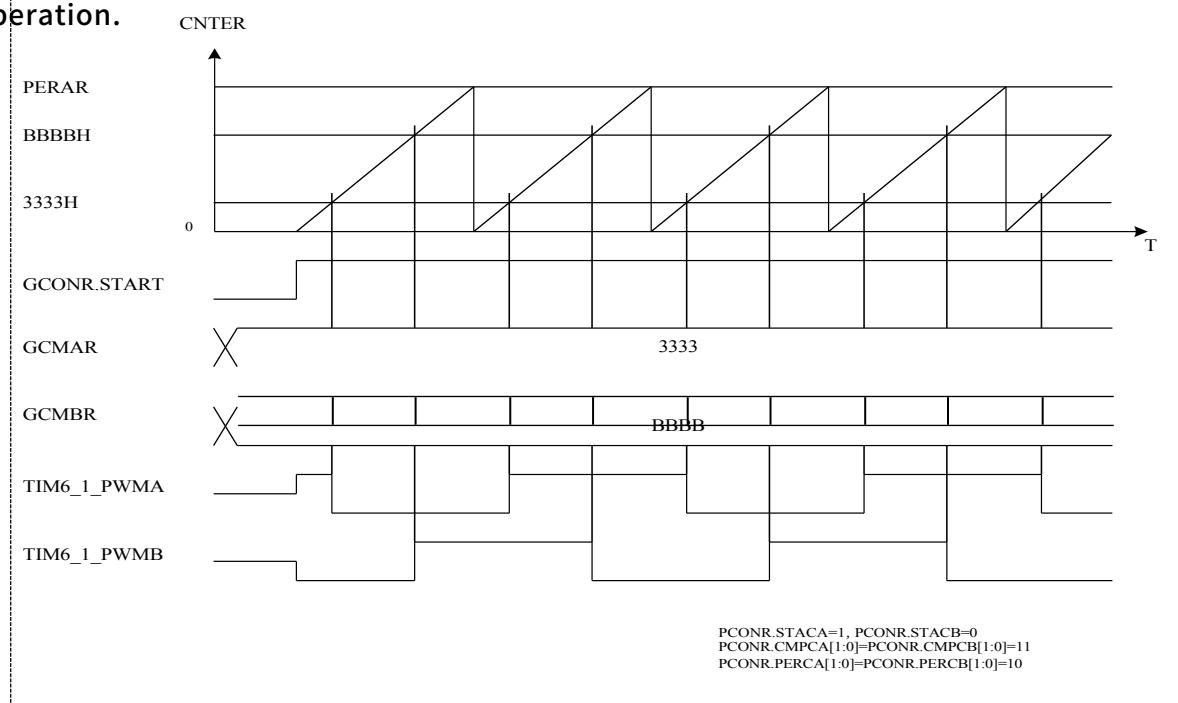


Figure 18-4 Comparison Output Action

18.3.1.3 Capture Input

Each unit has a capture input function with two sets of capture input registers (GCMAR, GCMBR) for storing captured count values. Setting the PCONR.CAPMDA and PCONR.CAPMDB bits of the port control register (PCONR) to 1 makes the capture input function effective. When the corresponding capture input condition is set and the condition is valid, the current count value is saved in the corresponding capture registers (GCMAR, GCMBR).

The conditions for each set of capture inputs of each unit can be internal trigger event inputs, TIM6_TRIGA or TIM6_TRIGB port inputs, TIM6_<t>_PWMA or TIM6_<t>_PWMB port inputs, etc. The specific selection of conditions can be set through the hardware capture event selection registers (HCPAR, HCPBR). The specific condition selection can be set through the hardware capture event selection registers (HCPAR, HCPBR). Figure 18-5 shows the action example of capturing inputs.

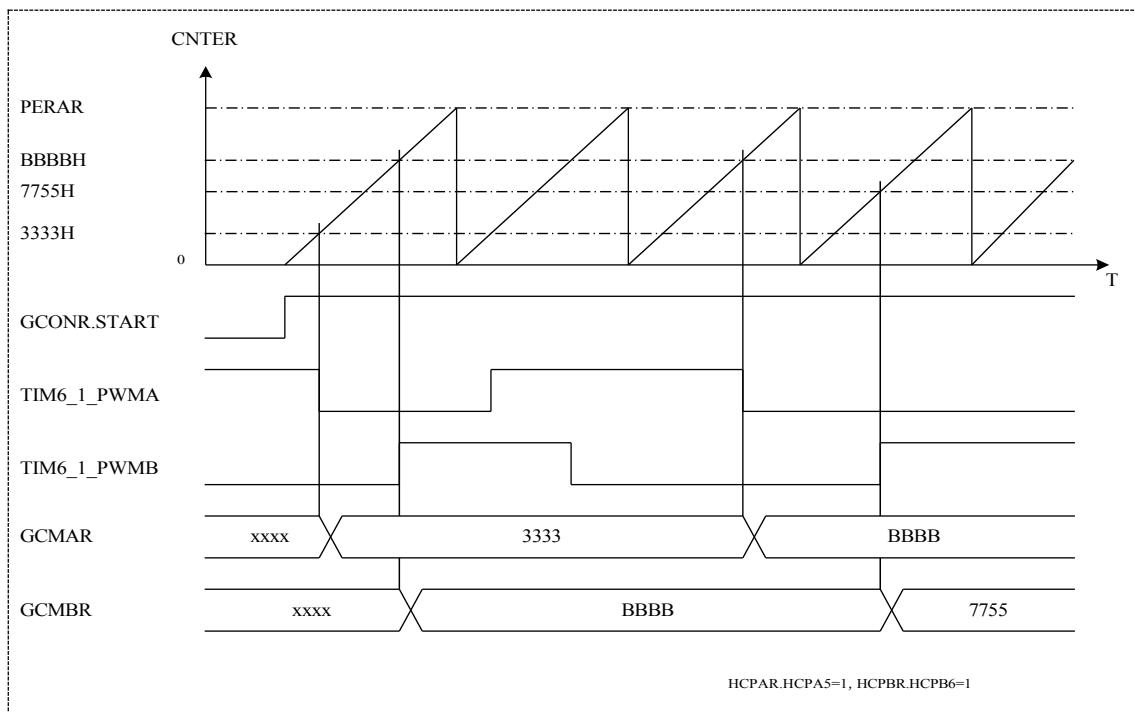


Figure 18-5 Capture Input Action

18.3.2 Clock source selection

The following options are available for the Timer6 count clock:

- a) PCLK0 and 2, 4, 8, 16, 64, 256, 1024 divisions of PCLK0 (GCONR.CKDIV[2:0] setting)
- b) Internal trigger event trigger input (HCUPR[17:16] or HCDOR[17:16] setting)
- c) Port input for TIM6_TRIGA-B (HCUPR[11:8] or HCDOR[11:8] setting)
- d) Quadrature coded inputs to TIM6_<t>_PWMA and TIM6_<t>_PWMB (HCUPR[7:0] or HCDOR[7:0])
(Settings)

When a is selected for the counting clock source, it is the software counting mode, and when b, c, or d is selected for the counting clock source, it is the hardware counting mode. As can be seen from the above description, the b, c, and d clocks are independent of each other and can be set to be valid or invalid respectively, and the a clock is automatically invalidated when the b, c, and d clocks are selected.

18.3.3 direction of counting

The timer counting direction of Timer6 can be changed by software. The method of changing the counting direction is slightly different for different waveform modes.

18.3.3.1 Sawtooth wave counting direction

In sawtooth wave mode, the counting direction can be set while the timer is counting or when it is stopped.

In incremental counting, set GCONR.DIR=0 (decremental counting) then the timer will change to decremental counting mode after counting up to overflow; in decremental counting, set GCONR.DIR=1 (incremental counting) then the timer will change to incremental counting mode after counting down to overflow.

DIR bit is set when counting is stopped, and the setting of GCONR.DIR is reflected in counting after counting is started until overflow or underflow.

18.3.3.2 Triangular wave counting direction

In the triangle wave mode, the counting direction setting is invalid, and the counting direction is automatically changed when the count reaches the counting peak or counting valley.

18.3.4 digital filtration

The TIM6_<t>_PWMA, TIM6_<t>_PWMB, and TIM6_TRIGA~B port inputs of Timer6 have digital filtering functions. The filtering function of the corresponding port can be enabled by setting the relevant enable bit of the Filter Control Register (FCONR). The filter reference clock when filtering is active can also be set through the filter control register (FCONR).

When the filter sampling reference clock samples three times the same level on the port, the level is treated as a valid level and transmitted to the internal module; levels less than three times the same are filtered out as external interference and are not transmitted to the internal module. An example of this action is shown in Figure 18-6.

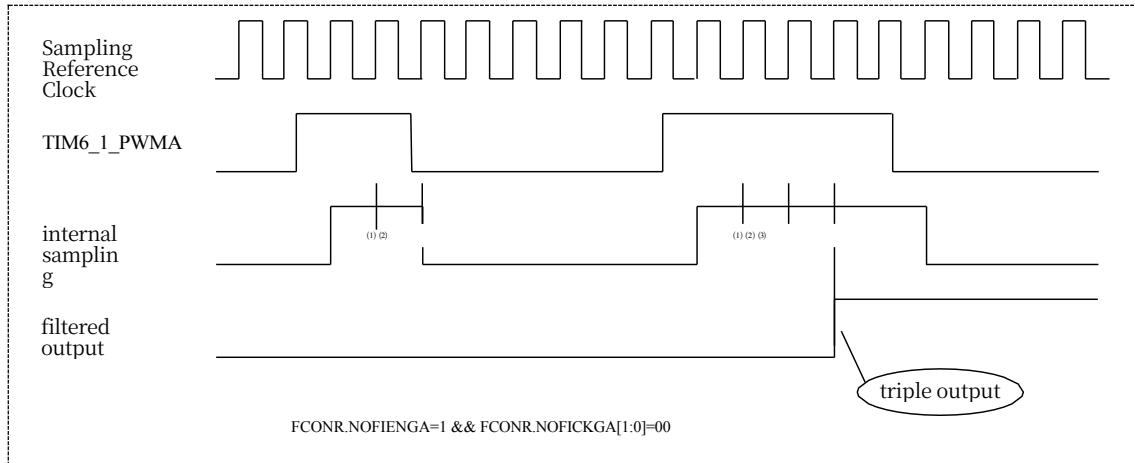


Figure 18-6 Capture Input Port Filtering Function

TIM6_TRIGA~B port is a group of ports shared among units, the digital filter function of this group of ports is set by the FCONR of unit 1, and the FCONR of other units is invalid for the digital filter function setting of this group of ports. When any unit uses the digital filter function, it is necessary to clear the TIMER6_1 bit in the function controller (PWC_FCG2) to zero.

18.3.5 software synchronization

18.3.5.1 Synchronized software startup

Each unit can synchronize the startup of the target unit by setting the relevant bits of the Software Synchronized Startup Control Register (SSTAR).

18.3.5.2 software synchronization stop

Each unit can be synchronized to stop the target unit by setting the relevant bit of the Software Synchronized Stop Control Register (SSTPR).

18.3.5.3 Software synchronized zeroing

Each unit can be synchronized to clear the target unit by setting the relevant bits of the software synchronization clearing control register (SCLRR).

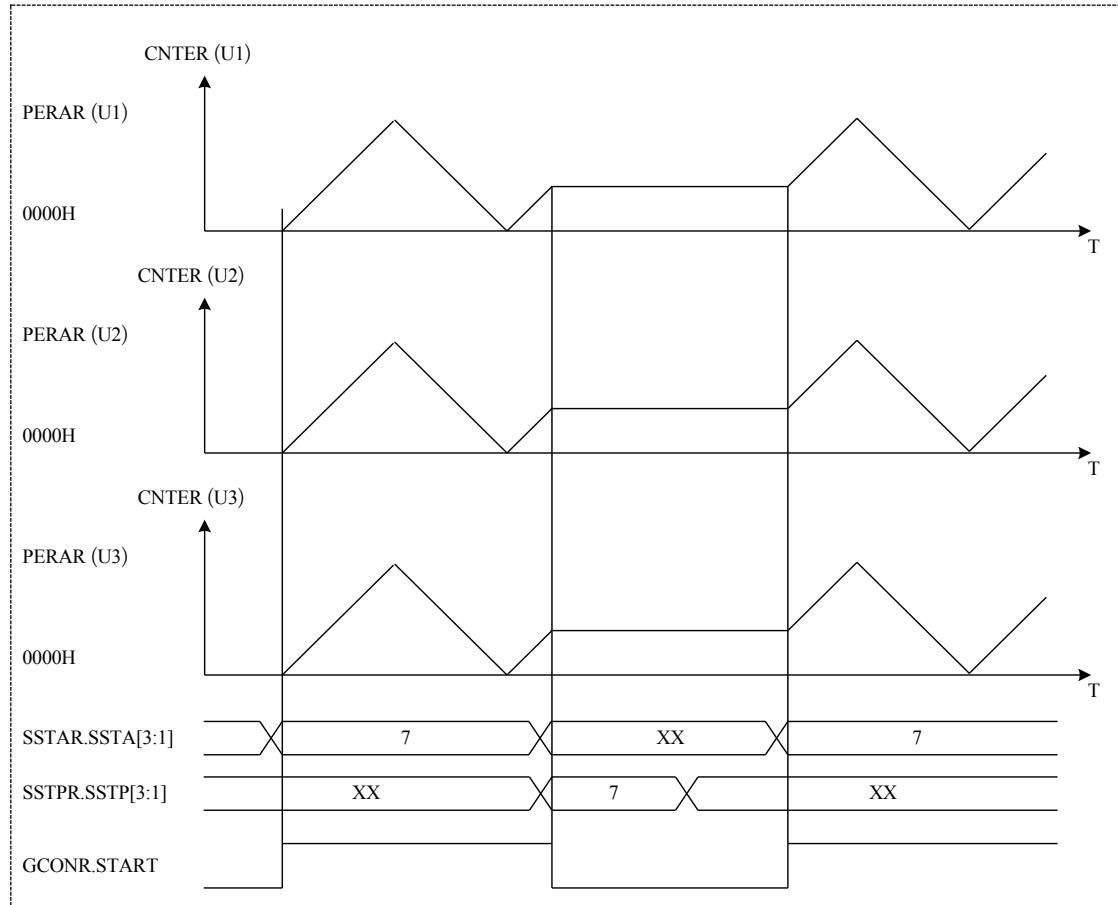


Figure 18-7 Software Synchronization Action

If you set **SSTAR.SSTA1=SSTAR.SSTA2=SSTAR.SSTA3=1** you can realize unit 1~3 as shown in Figure 18-7.
of the software starts synchronously.

The Software Synchronization Action Related Registers (SSTAR, SSTPR, SCLRR) are a set of registers that are independent of the unit and shared among the units; the bits in this set of registers are only valid on a write of 1, and a write of 0 is invalid. When the SSTAR register is read, the timer status of each unit is read, and when SSTPR or SCLRR is read, 0 is read.

18.3.6 hardware synchronization

In addition to the two general-purpose input ports (TIM6_<t>_PWMA, TIM6_<t>_PWMB) independently, each unit also shares two external general-purpose input ports (TIM6_TRIGA, TIM6_TRIGB) and two internal triggering event input conditions, which enable hardware synchronization of actions between units.

The event source of the internal hardware trigger event can be selected by the corresponding number setting in the Hardware Trigger Event Selection Register (HTSSR0~1), please refer to the [Interrupt Controller (INTC) chapter for the specific event correspondence. When using the internal hardware trigger function, it is necessary to enable the peripheral circuit trigger function of the function clock control register 0 (PWC_FCG0) to position 1 first.

18.3.6.1 hardware synchronous boot

Each unit can choose to start the timer in hardware, and units with the same hardware start condition can realize synchronous start when the start condition is valid. The specific hardware startup condition is determined by the setting of the hardware startup event selection register (HSTAR).

18.3.6.2 Hardware synchronization stop

Each unit can choose to stop the timer by hardware, and units that select the same hardware stop condition can realize synchronous stop when the stop condition is valid. The specific hardware stop condition is determined by the setting of the hardware stop event selection register (HSTPR).

18.3.6.3 Hardware synchronization clearing

Each unit can choose to clear the timer in hardware, and units with the same hardware clearing condition can realize synchronous clearing when the clearing condition is valid. The specific hardware clearing condition is determined by the setting of the hardware clearing event selection register (HCLRR).

18.3.6.4 Hardware synchronized capture input

Each unit can choose to implement the capture input function in hardware, and units with the same capture input function conditions can realize synchronized capture input when the capture input function conditions are valid. The specific hardware capture input function conditions are determined by the settings of the hardware capture event selection registers (HCPAR, HCPBR).

18.3.6.5 hardware synchronous counting

Each unit can choose to use the hardware input as the CLOCK for counting, and units with the same hardware counting condition can synchronize the counting when the hardware counting clock is valid. The specific hardware counting conditions are determined by the settings of the hardware count-up event selection register (HCUPR) and the hardware count-down event

selection register (HCDOR).

Figure 18-8 shows an example of hardware synchronization for Units 1 to 3.

When the hardware synchronized counting function is selected, only the external input clock source is selected, and it does not affect the start, stop, and clear actions of the timer. The start, stop, and clear of the timer still need to be set separately.

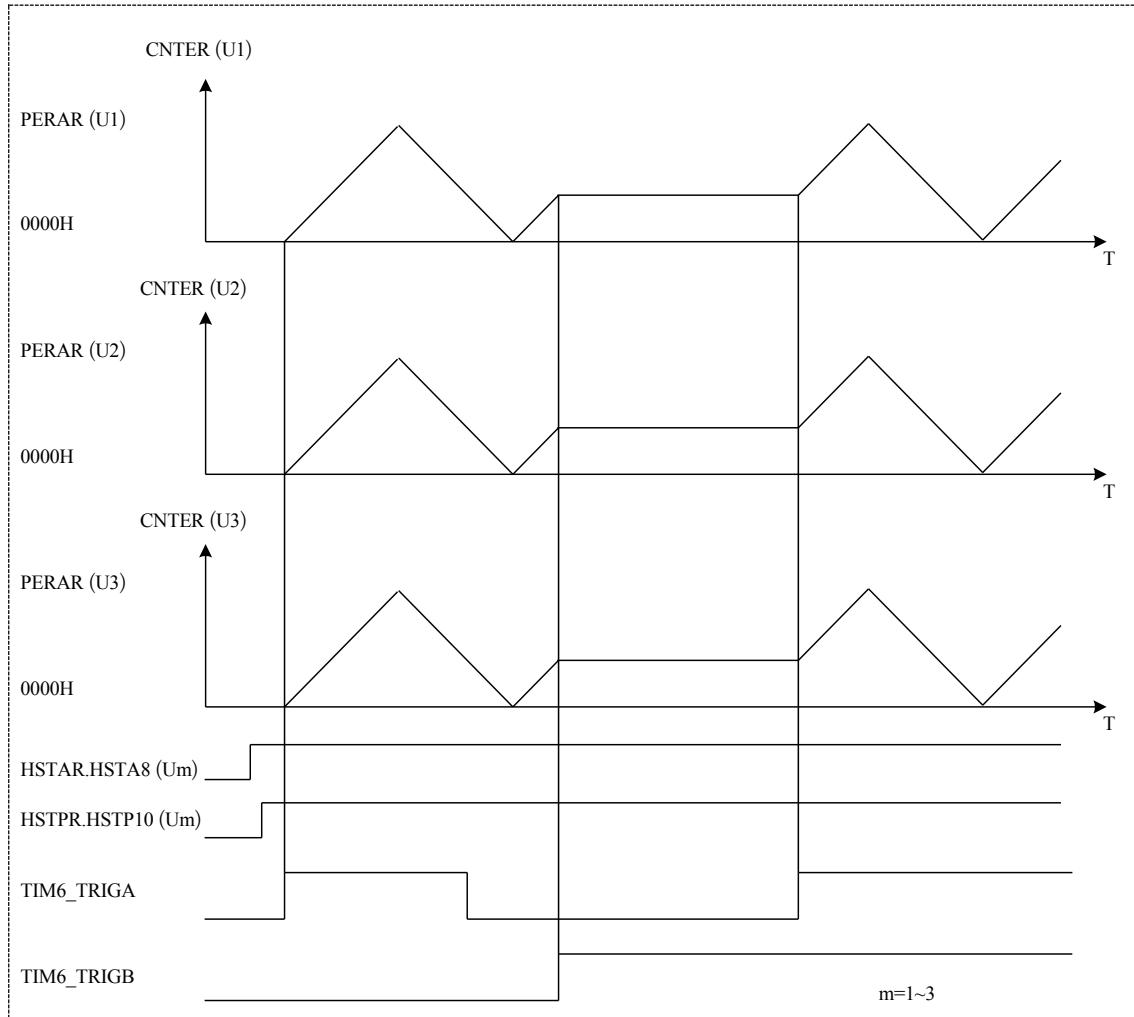


Figure 18-8 Hardware Synchronization

18.3.7 Pulse Width Measurement

When using the hardware trigger related functions of the TIM6_<t>_TRIGA~B port (refer to the Hardware Synchronization chapter), each unit can implement 2 independent pulse width measurement functions.

For example, continuous pulse width measurement can be realized by setting the hardware start condition of the counter to the rising edge of TIM6_<t>_TRIGA, and the hardware clearing condition, stop condition, and capture input condition of the GCMAR register to the falling edge of TIM6_<t>_TRIGA.

18.3.8 periodic measurement

When using the hardware trigger-related functions of the TIM6_<t>_TRIGA~B port (refer to the Hardware Synchronization chapter), each unit can implement 2 independent cycle measurement functions.

For example, by setting the counter's hardware start condition, hardware clear condition, and GCMBR register capture input condition to the rising edge of TIM6_<t>_TRIGB, continuous cycle measurement can be realized.

18.3.9 caching function

Timer6's counting period value, general-purpose comparison reference value, special-purpose comparison reference value, and dead time setting value all have cache functions to realize period change, duty cycle change, and dead time change during counting. The counting period value, general-purpose comparison reference value, and special-purpose comparison reference value have single and double cache functions, and the dead time setting value has a single cache function.

18.3.9.1 Single Cache Action

A single cache action is the selection of the following events to occur at the cache transfer time point by setting the cache control register (BCONR) and the deadzone control register (DCONR)

- a) The value of the general-purpose cycle reference value cache register (PERBR) is automatically transferred into the general-purpose cycle reference value register (PERAR)
- b) The value of the general-purpose comparison reference value cache register (GCMCR\$GCMDR) is automatically transferred into the general-purpose comparison reference value register (GCMAR, (GCMBR) in (when comparing outputs)
- c) The value of the Generalized Comparison Reference Value Register (GCMARG C M B R) is automatically transferred to the Generalized Comparison Reference Value Cache Register (GCMCR, GCMDR) (at capture input)
- d) Dedicated Comparison Reference Value Cache Register (SCMCR\$CMDR) values are automatically transferred to the dedicated Comparison Reference Value Registers (SCMAR, SCMBR)

-
- e) The value of the dead-time reference value cache registers (DTUBRD, DTDBR) is automatically transferred to the dead-time reference value registers (DTUAR, DTDAR).

As shown in Figure 18-9, this is a timing diagram of the single-cache mode of the generalized comparison reference value register when Unit 1 compares the output actions.

As can be seen from the figure, changing the value of the generalized comparison reference value register (GCMAR) during counting adjusts the output duty cycle, and changing the value of the generalized period reference value register (PERAR) adjusts the output period.

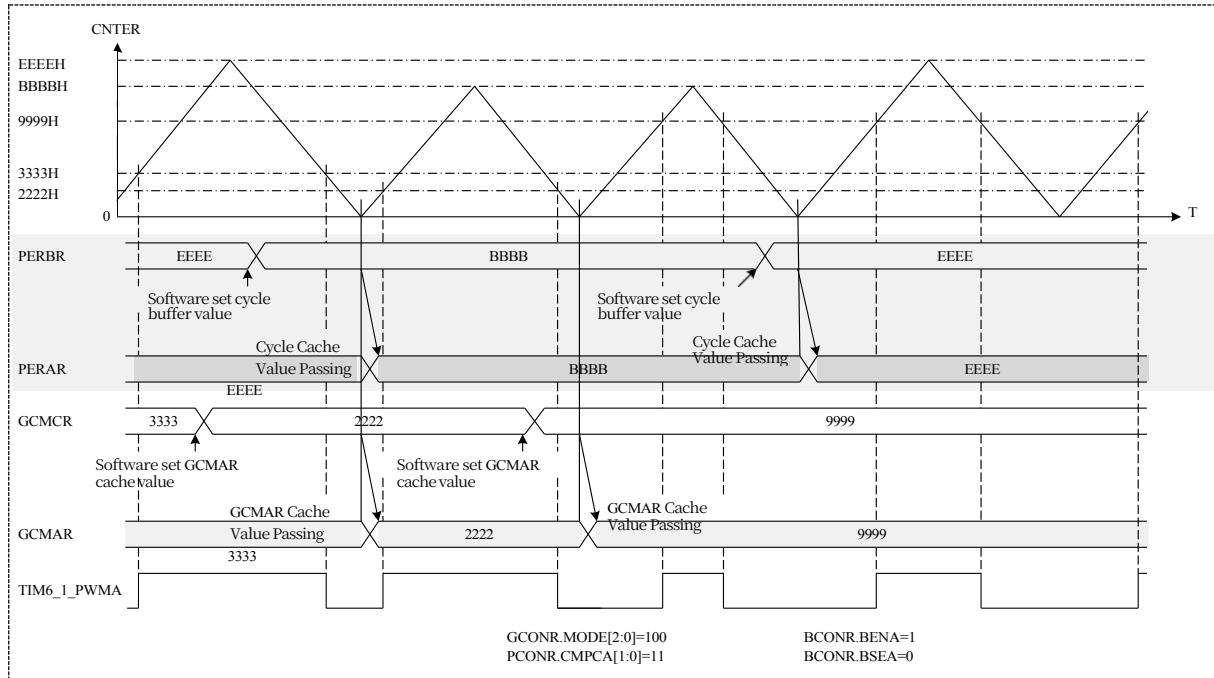


Figure 18-9 Single-Cache Comparison Output Timing

18.3.9.2 Double Cache Action

A double cache action is the selection of the following events to occur at the cache transfer time point by setting the cache control register (BCONR)

- The value of the general-purpose period reference value cache register (PERBR) is automatically transferred to the general-purpose period reference value register (PERAR), and the value of the general-purpose period reference value double cache register (PERCR) is automatically transferred to the general-purpose period reference value cache register (PERBR).
- The value of the Generalized Comparison Reference Value Cache Registers (GCMCR, GCMDR) is automatically transferred to the Generalized Comparison Reference Value Registers (GCMAR, GCMBR) and the value of the Generalized Comparison Reference Value Dual Cache Registers (GCMER, GCMFR) is automatically transferred to the Generalized Comparison Reference Value Cache Registers (GCMCR, GCMDR) (when comparing outputs)
- The value of the Generalized Comparison Reference Value Cache Registers (GCMCR, GCMDR) is automatically transferred to the Generalized Comparison Reference Value Dual Cache Registers

(GCMAR, GCMBR) are automatically transferred to the generalized comparison reference value cache registers (GCMCR, GCMDR) (when capturing inputs)

- d) The values of the dedicated comparison base value cache registers (SCMCR, SCMDR) are automatically transferred into the dedicated comparison base value registers (SCMAR, SCMBR), and the values of the dedicated comparison base value dual cache registers (SCMER, SCMFR) are automatically transferred into the dedicated comparison base value cache registers (SCMCR, SCMDR).

Figure 18-10 shows the timing diagram for the dual-cache method when internal trigger event 0 triggers the capture input.

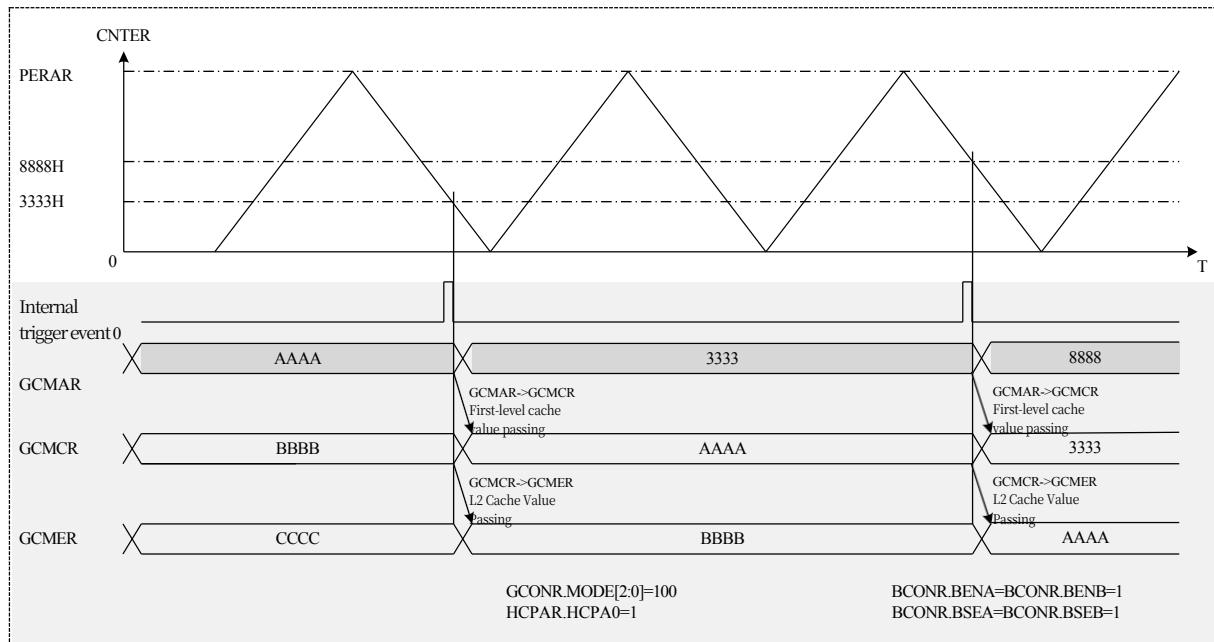


Figure 18-10 Capturing Input Timing in Dual-Cache Mode

18.3.9.3 Cache Delivery Point in Time

Generic Cycle Base Value Cache Transmission Timing Points

The cycle reference value can be selected from single cache function or double cache function (BCONR.BSEP) The cache transfer time points are the incremental count up overflow point or decremental count down overflow point for sawtooth wave, and the count valley point for triangle wave.

Generic Baseline Value Cache Transmission Timing Points

Setting BCONR.BENA=1 or BCONR.BNEB=1 when in Ramp Wave mode makes the cache action effective. The cache action can be selected as single cache function or double cache function. The cache transfer occurs at the overflow point or underflow point, as shown in Figure 18-11.

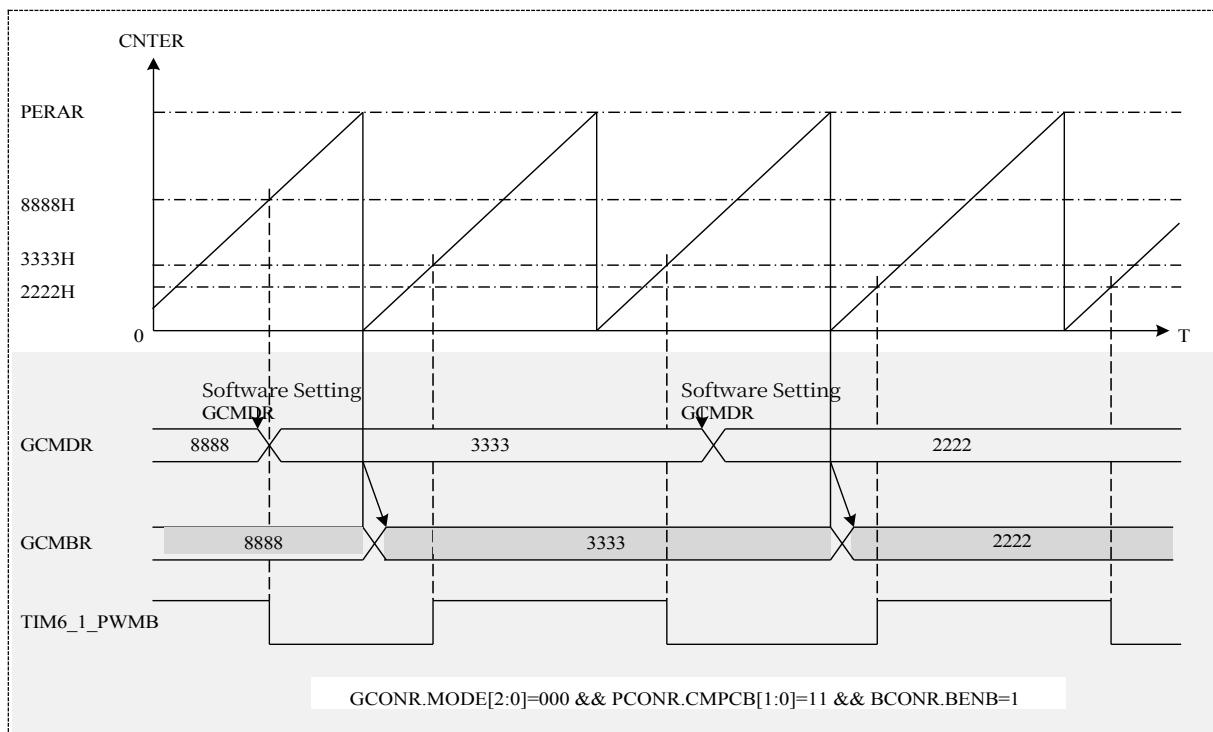


Figure 18-11 Count Buffer Action in Sawtooth Wave Mode

Set BCONR.BENA=1 or BCONR.BNEB=1 to make the cache action effective in triangle wave A mode. The cache action can be selected as single cache function or double cache function. The cache transfer occurs at the count valley as shown in Figure 18-12.

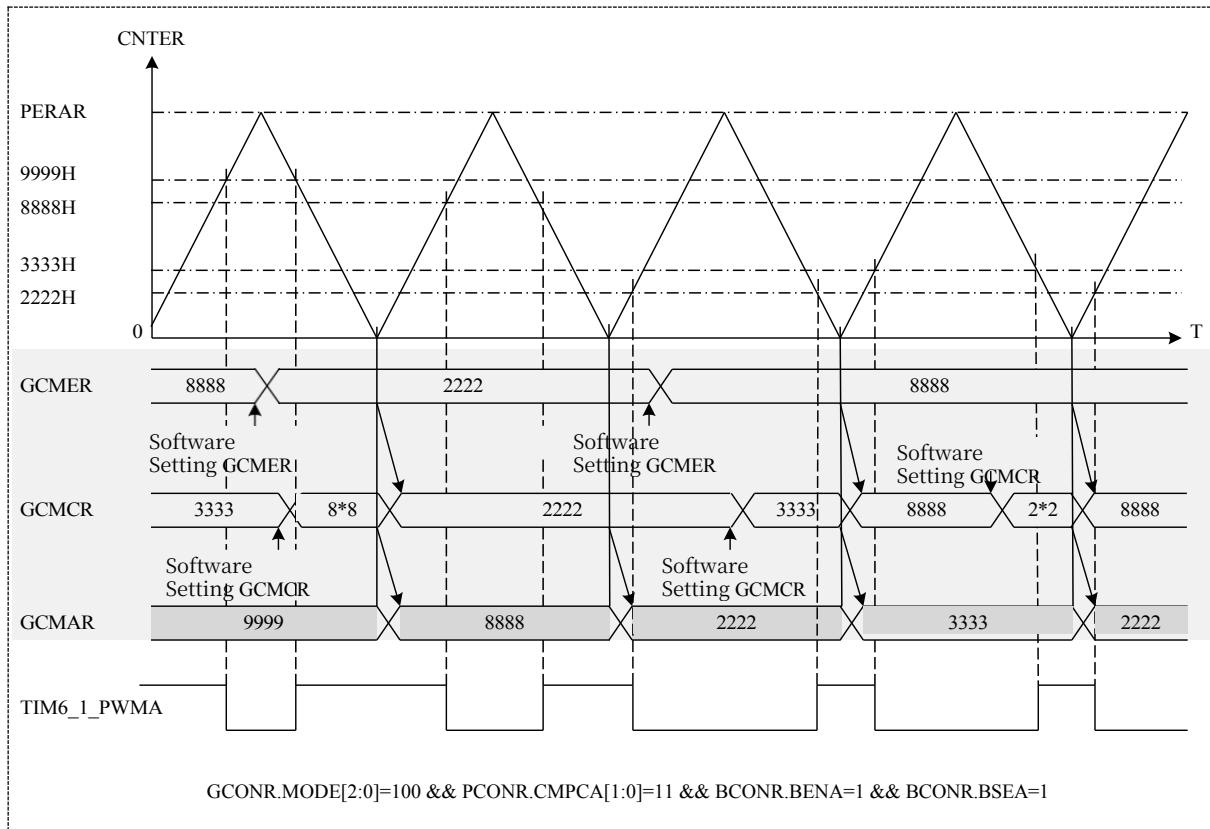


Fig. 18-12 Counter Buffer Action in Triangle A Mode

Set BCONR.BENA=1 or BCONR.BNEB=1 to make the cache action effective in Triangle B mode. The cache action can be selected as single cache function or double cache function. The cache transfer occurs at the count valley or count peak as shown in Figure 18-13.

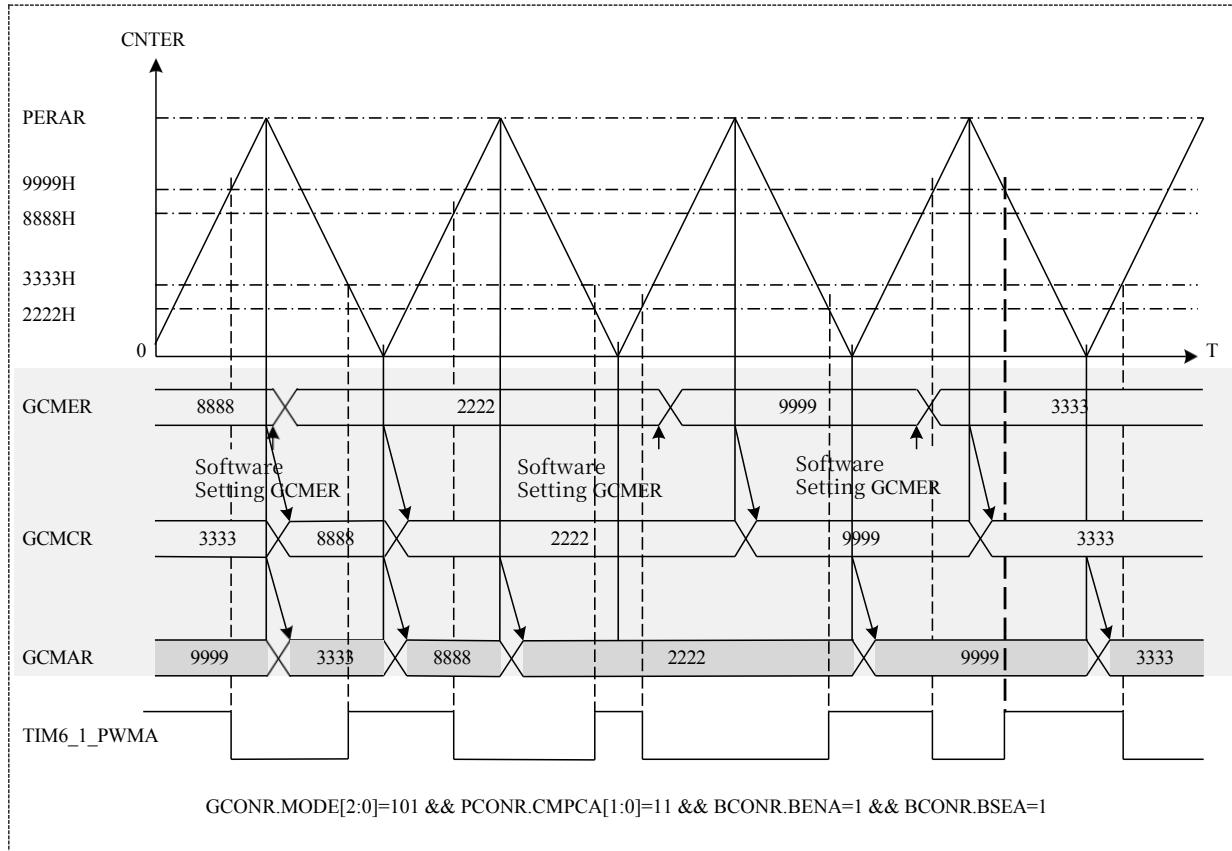


Fig. 18-13 Counter Buffer Action in Triangle B Mode

Single-cache or double-cache transfers are determined by BCONR.BENA, BCONR.BENB, BCONR.BSEA, BCONR.BSEB.

Capture input value cache transfer time point

Capturing an input action selects either the single cache function or the dual cache function (BCONR.BSEA or BCONR.BSEB) The cache transfer time point is when the input action is captured.

Dedicated Comparison Baseline Value Cache Transmission Point in Time

Dedicated comparison reference values can be selected for single or dual cache functions (BCONR.BSESPA or BCONR.BSESPB) The cache transfer time point is set by BCONR.BTRSPA and BCONR.BTRSPB of the cache control register BCONR.

Dead Time Baseline Cache Transmission Timing Points

The dead time reference has a single cache function. The cache transfer time points are the incremental count up overflow or decremental count down overflow for sawtooth wave, and the count valley for triangle wave.

Cache transfer during zero action

In the sawtooth counting mode or the hardware counting mode, if a clearing action is generated during the comparison output action, a cache transfer occurs in the general-purpose cycle reference value, general-purpose comparison reference value, special-purpose comparison reference value, and dead time reference value registers, etc., according to the corresponding cache action setting status (single-cache, double-cache, etc.).

18.3.10 General Purpose PWM Output

18.3.10.1 Independent PWM output

The two ports TIM6_<t>_PWMA and TIM6_<t>_PWMB of each unit can output PWM wave independently. As shown in Figure 18-14, TIM6_<t>_PWMA port outputs PWM wave.

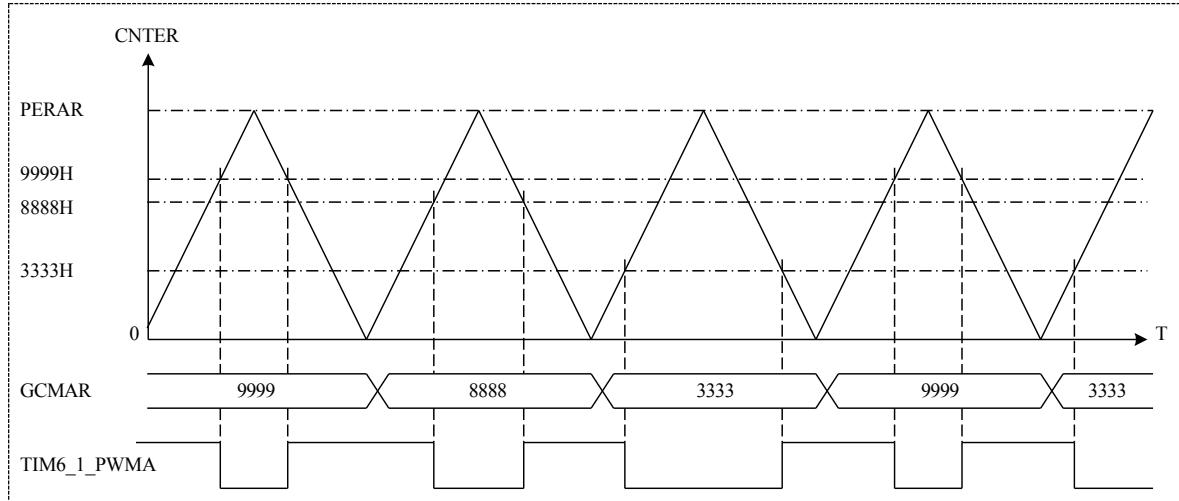


Figure 18-14 TIM6_<t>_PWMA Output PWM Waves

18.3.10.2 Complementary PWM output

TIM6_<t>_PWMA port and TIM6_<t>_PWMB port can be combined to output complementary PWM waveforms in different modes.

Software set GCMBR Complementary PWM Outputs

Software setting GCMBR Complementary PWM Output means that the Generalized Comparison Reference Value Register (GCMBR) used for waveform output of TIM6_<t>_PWMB port is directly written by the CPU, etc. in sawtooth and triangle wave modes, and is not directly related to the value of GCMAR.

Figure 18-15 shows an example of software setting the output of GCMBR complementary PWM waveform.

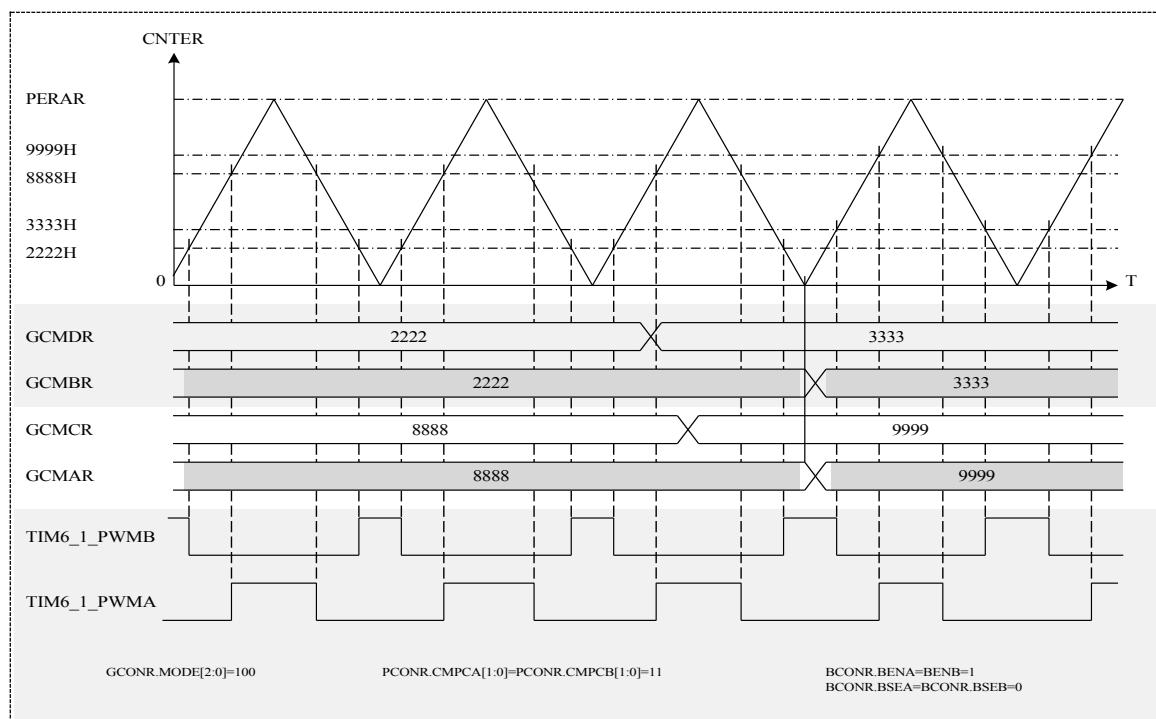


Fig. 18-15 Software Setting GCMBR Complementary PWM Output in Triangle A Mode

Hardware Setting GCMBR Complementary PWM Outputs

Hardware Setting GCMBR Complementary PWM Output means that the value of the Generalized Comparison Reference Value Register (GCMBR) used for the waveform output of the TIM6_<t>_PWMB port is determined by the value of the Generalized Comparison Reference Value Register (GCMAR) and the Deadtime Reference Value Register (DTRV) in the triangle wave mode.

(DTU<D>AR) is determined by the value operation.

The dead time setting also has a cache function. When the cache function is active (DCONR.DTBENU/DTBEND=1), the value of DTUBR is transmitted to DTUAR and the value of DTDBR is transmitted to DTDAR at the cache transmission time point (counting valley point during delta wave)

Figure 18-16 shows an example of hardware setting of GCMBR complementary PWM wave output.

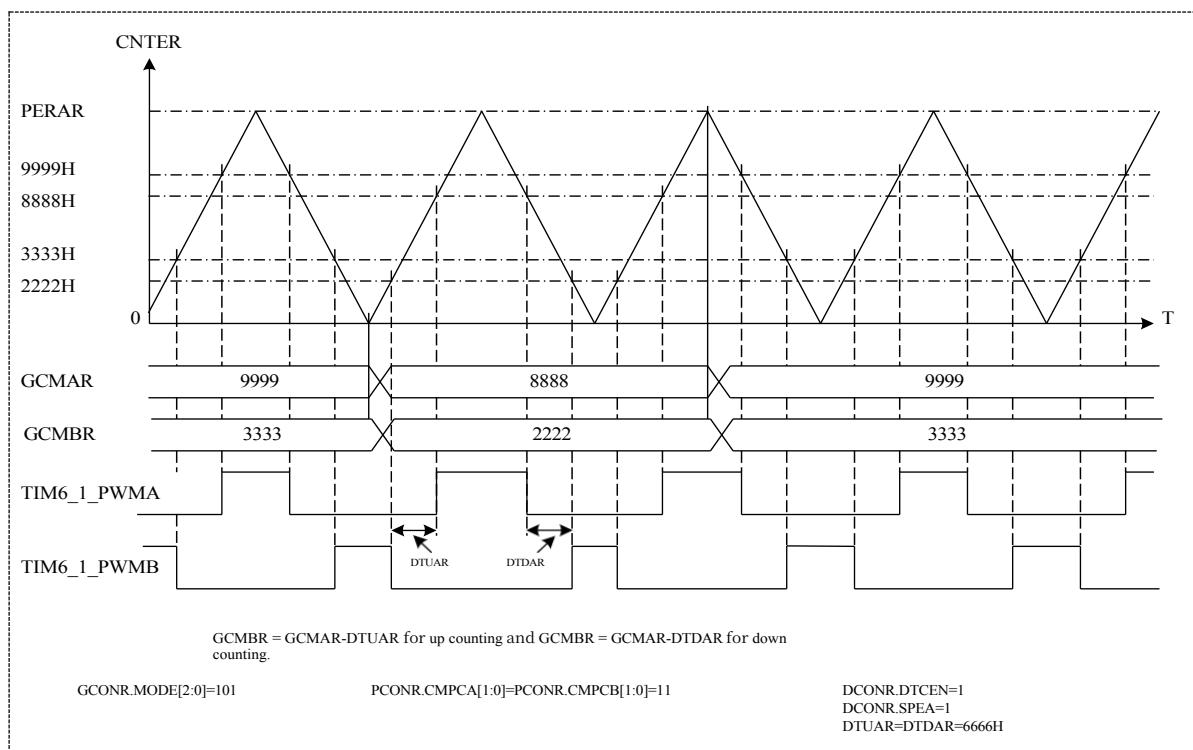


Figure 18-16 Hardware Setting of GCMBR Complementary PWM Waveform Output in Triangle B Mode (Symmetric Deadband)

18.3.10.3 Multi-phase PWM output

The TIM6_<t>_PWMA and TIM6_<t>_PWMB ports of each unit are capable of outputting 2-phase independent PWM waveforms or a set of complementary PWM waveforms, and multi-phase PWM waveforms can be output by the combination of multiple units, and the combination of synchronized actions of software and hardware. As shown in Figure 18-17, Unit 1, Unit 2 and Unit 3 combine to output 6-phase PWM wave; as shown in Figure 18-18, Unit 1, Unit 2 and Unit 3 combine to output 3-phase complementary PWM wave.

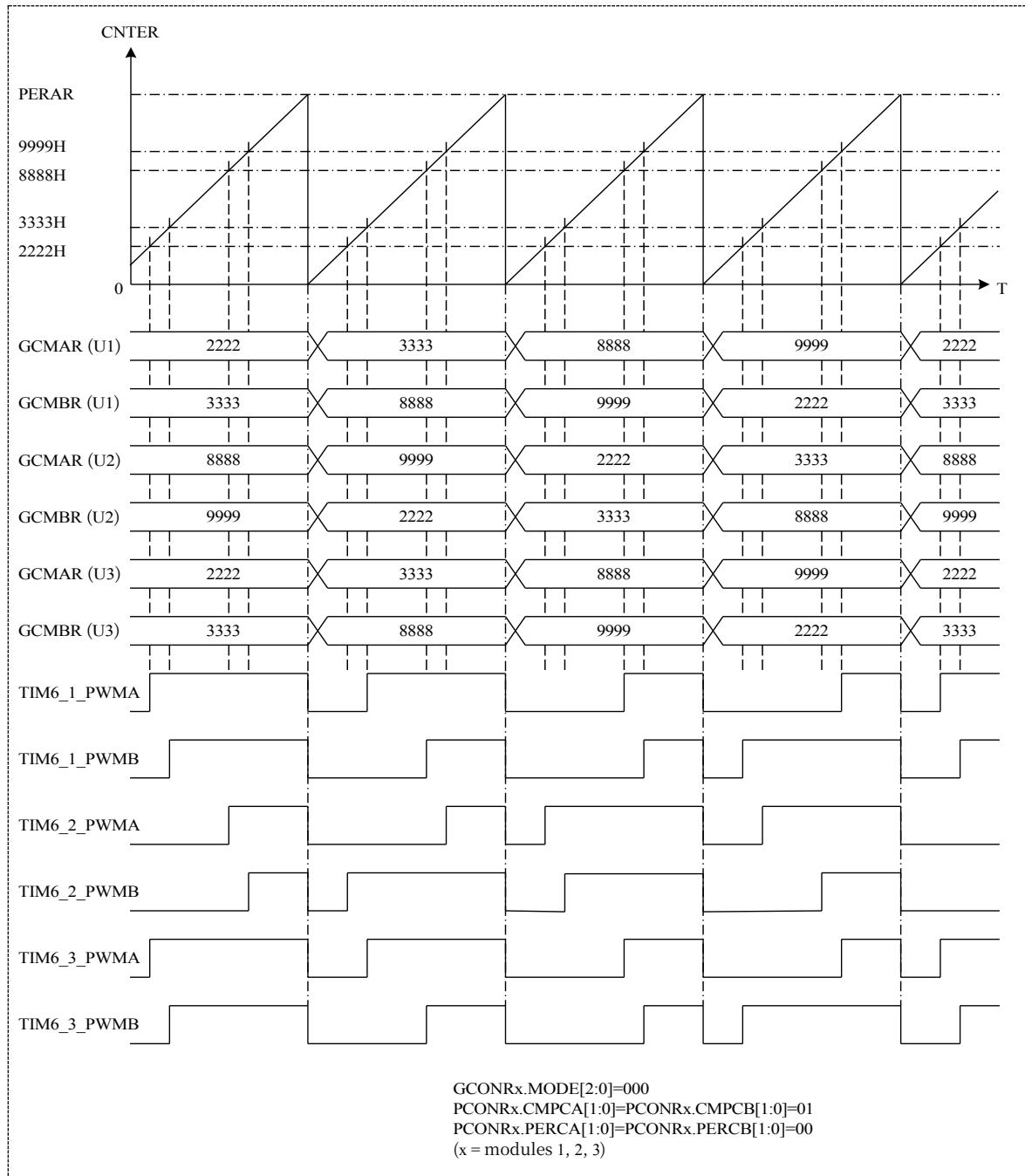


Figure 18-17 6-Phase PWM Wave

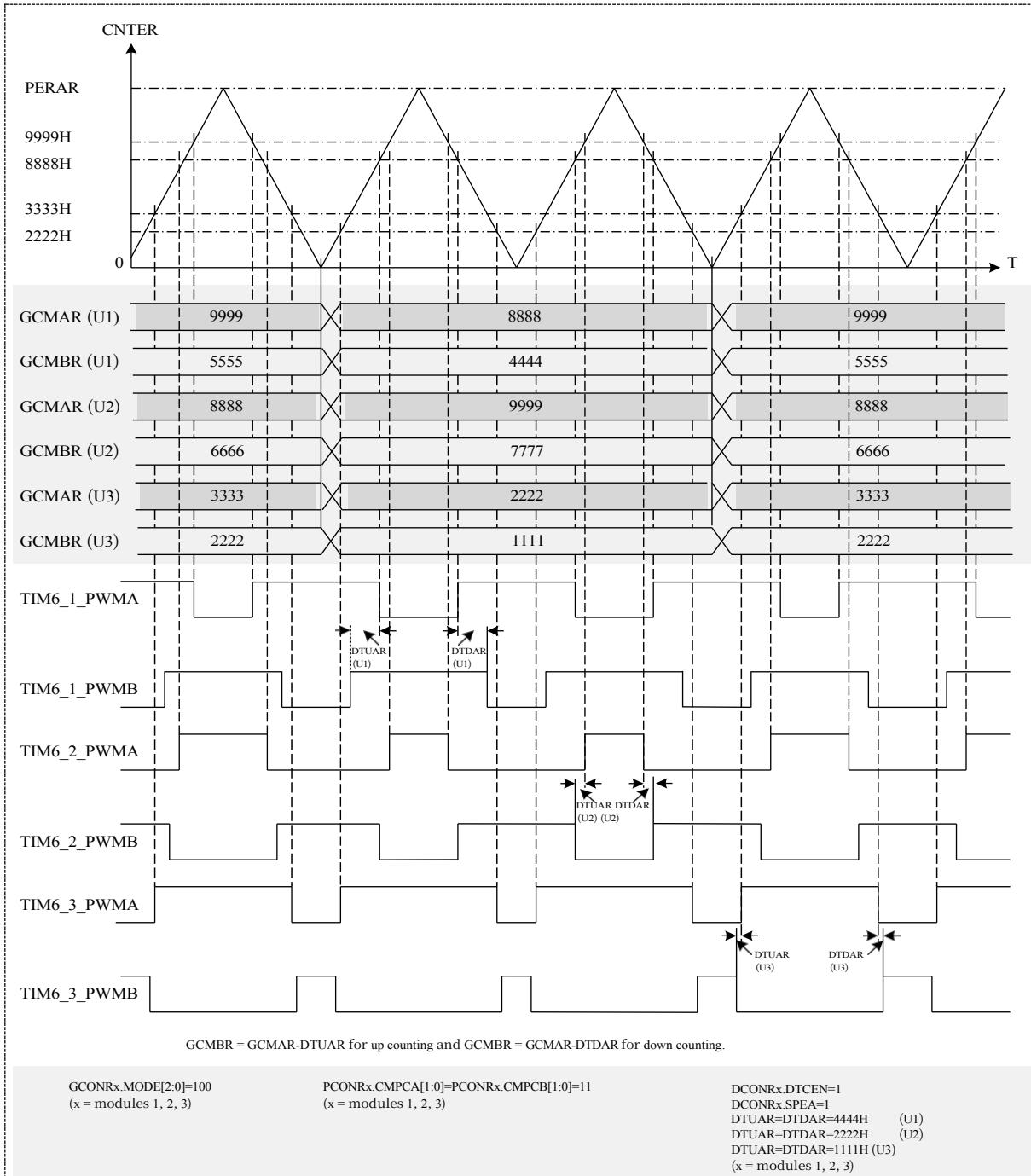


Figure 18-18 Three-Phase Complementary PWM Waveform Output with Dead Time in Triangle A Mode

18.3.11 orthogonal code counting

Think of the TIM6_<t>_PWMA input as the AIN input, the TIM6_<t>_PWMB input as the BIN input, and the TIM6_TRIGA-

If any one of the inputs in B is considered as ZIN input, Timer6 can realize the orthogonal coded counting of three inputs.

A 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 one unit is used for revolution counting.

In the revolution counting mode, Units 1 and 2 are combined, with Unit 1 acting as the position counting unit and Unit 2 acting as the revolution counting unit to realize position counting and revolution counting, respectively. Unit 3 is not used in the revolution counting mode.

The counting conditions for AIN and BIN are set by setting the Hardware Counter Event Select Register (HCUPR) and the Hardware Counter Event Select Register (HCUPR).

(The ZIN input action is realized by setting the hardware clear event selection register (HCLRR) of the position unit to clear the position timer of the position counter unit, and by setting the hardware increment event selection register (HCUPR) of the revolution unit to count the revolution timer of the revolution counter unit. HCUPR) of the revolution counting unit to realize the revolution timer counting of the revolution counting unit.

18.3.11.1 Position Counting Mode

Orthogonal encoding position counting mode means that the basic counting function, phase difference counting function and direction counting function are realized according to the inputs of AIN and BIN.

basic count

The basic counting action is based on the input clock of the AIN or BIN port as shown in Figure 18-19 below.

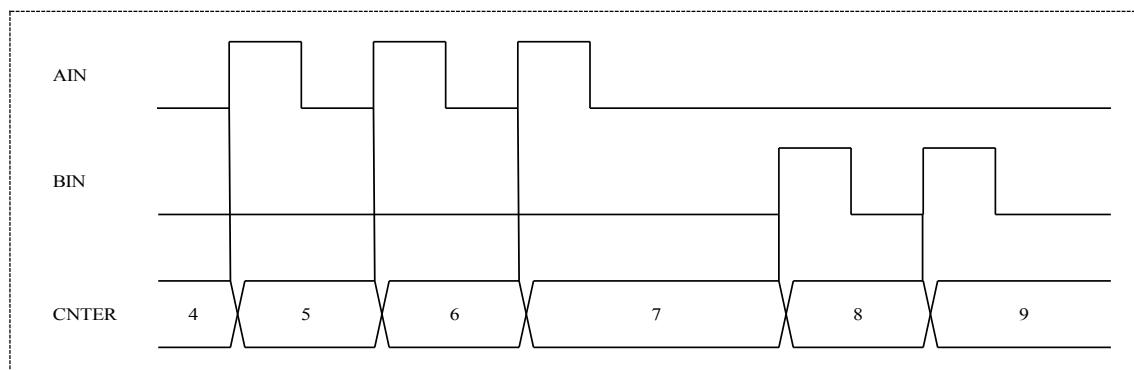


Figure 18-19 Position Mode-Basic Count

phase difference counting

Phase difference counting refers to counting according to the phase relationship between AIN and BIN. Depending on the setting, 1x counting, 2x counting, 4x counting, etc. can be realized as shown in Figure 18-20 to Figure 18-22 below.

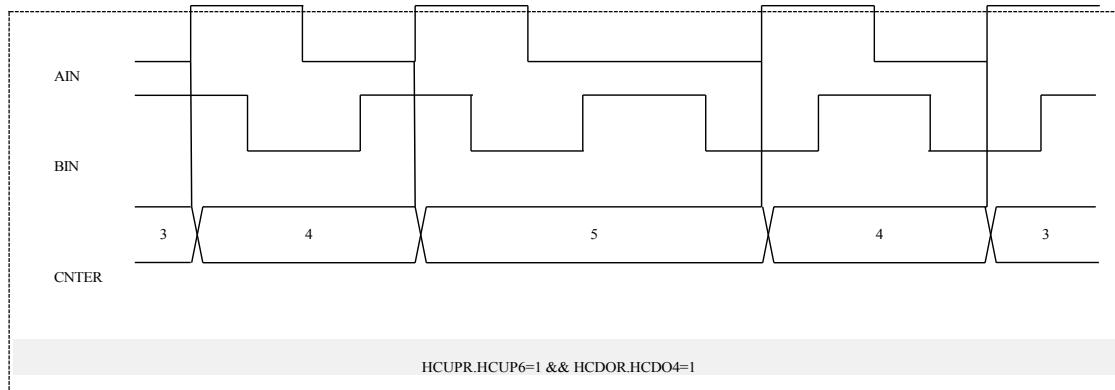


Figure 18-20 Position Count Mode-Phase Difference Count (1x Count)

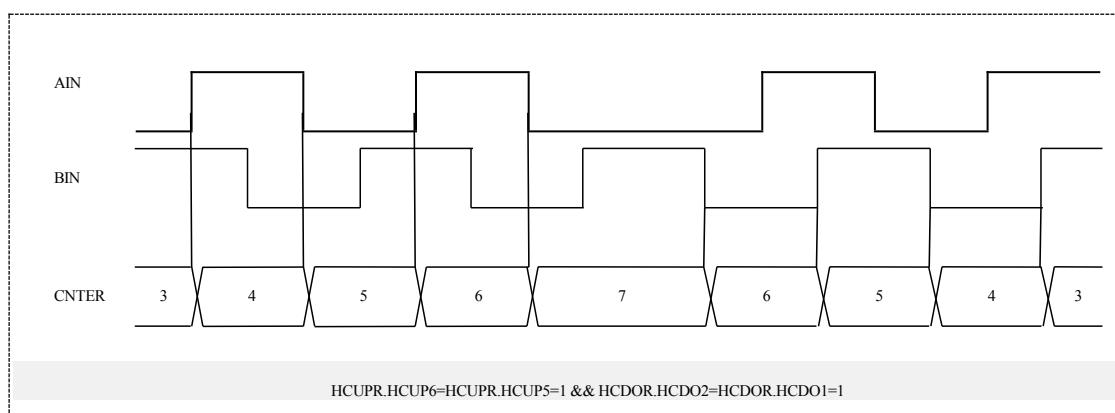


Figure 18-21 Position Count Mode-Phase Difference Count (2x Count)

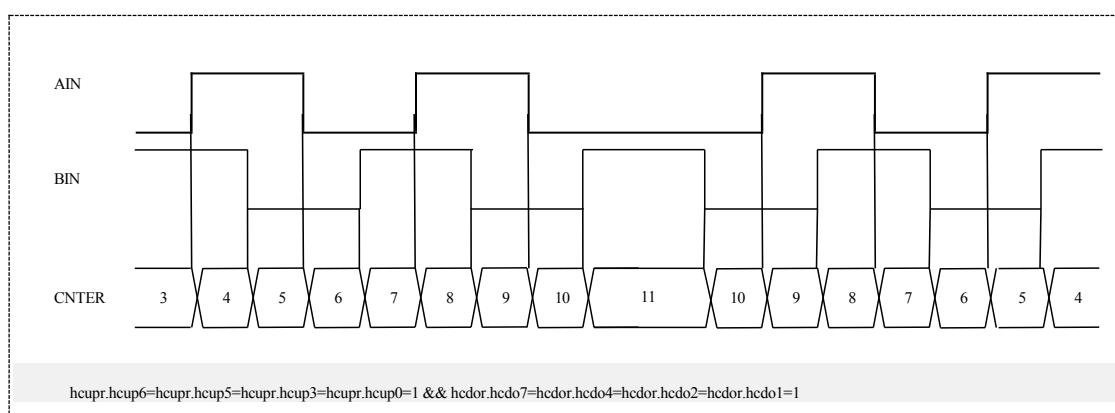


Figure 18-22 Position Count Mode-Phase Difference Count (4x Count)

Direction counting

Direction counting means that the input state of AIN is set as direction control and the input of BIN is used as clock counting as shown in Figure 18-23 below.

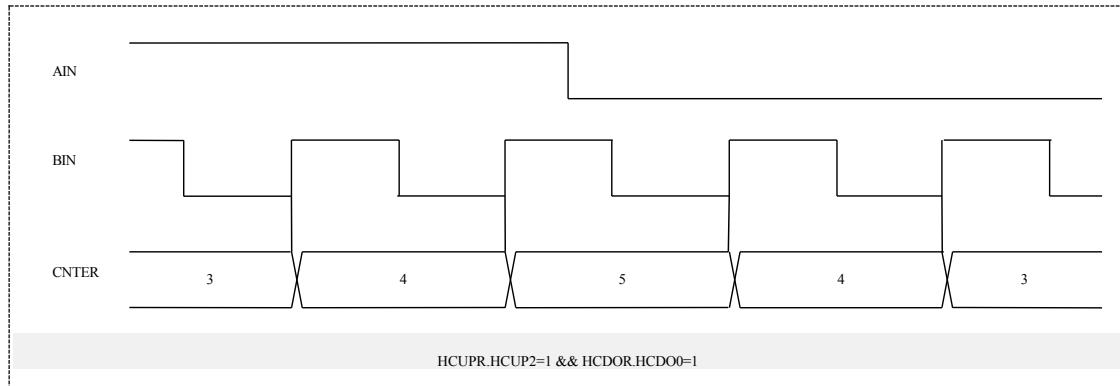


Figure 18-23 Position Count Mode-Direction Count

18.3.11.2 Rotation counting mode

The orthogonal encoding revolution counting mode is to add the ZIN input event to the AIN and BIN counting to realize the judgment of the number of revolutions. In the revolution counting mode, the Z-phase counting function, position overflow counting function, and mixed counting function can be realized according to the counting method of the revolution timer.

Z-phase counting

Z-phase counting is a counting action in which the revolution counting unit counts and the position counting unit is cleared to zero based on the ZIN input. This is shown in Figure 18-24 below.

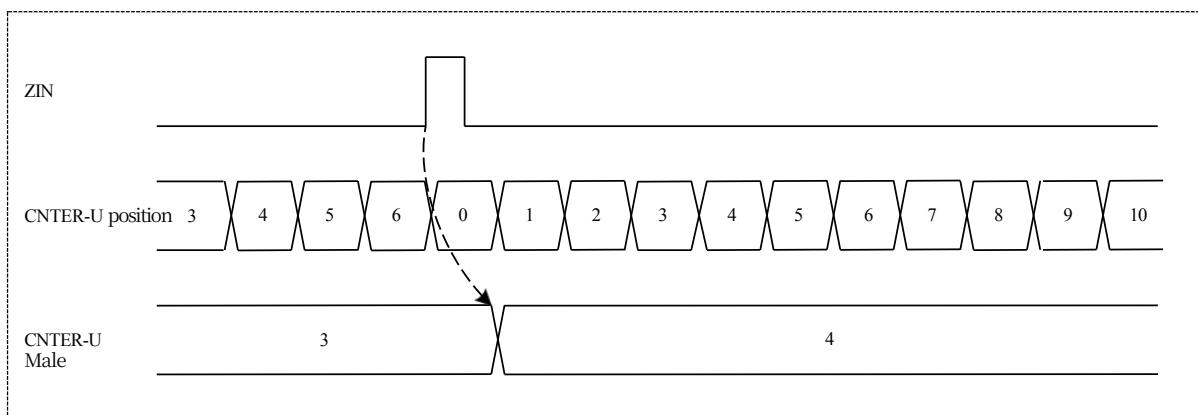


Figure 18-24 Rotation Counting Mode-Z Phase Counting

Position Overflow Count

position counting unit overflows or underflows, which triggers the timer of the rotary counting unit to perform a count (the ZIN input does not perform the counting of the rotary counting unit or the clearing of the position counting unit in this counting method)

The overflow event of the position counting unit is selected through the internal trigger event interface to realize the counting of the revolution counting unit, which can realize the position overflow counting. The hardware incremental (decremental) event selection register (HCUPR or HCDOR) of the revolution counting unit selects bit 1 of the incremental (decremental) event selection register (Bit16~Bit7), and at the same time, the event number in the corresponding event trigger selection register (HTSSR0~1) is set to be the overflow or underflow event of the position counting unit. Refer to the [Interrupt Controller (INTC) section for the specific event number. Figure 18-25 below shows the event number.

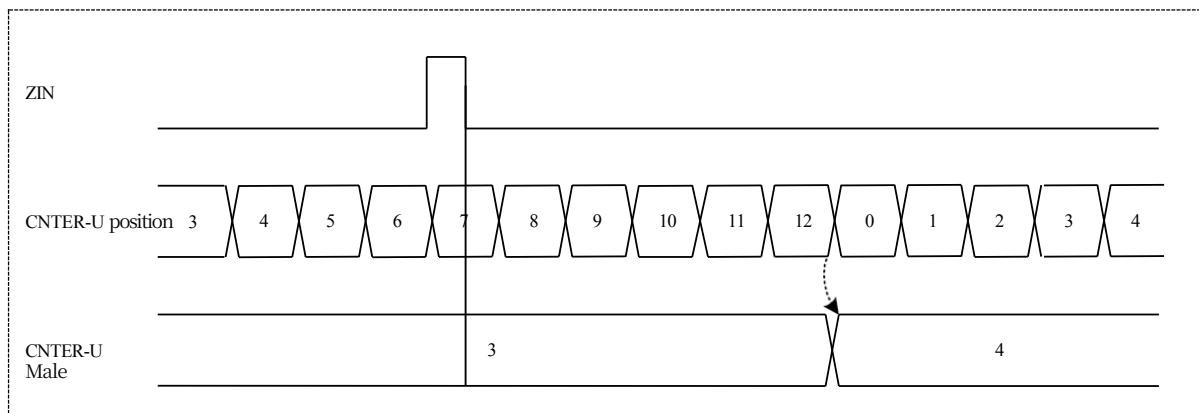


Figure 18-25 Rotation Counting Mode-Position Overflow Counting

Mixed

countin

g

Mixed counting refers to the counting action that combines the above two counting methods, Z-phase counting and position overflow counting, and its realization is also a combination of the above two counting methods. It is realized as a combination of the above two counting methods as shown in Figure 18-26 below.

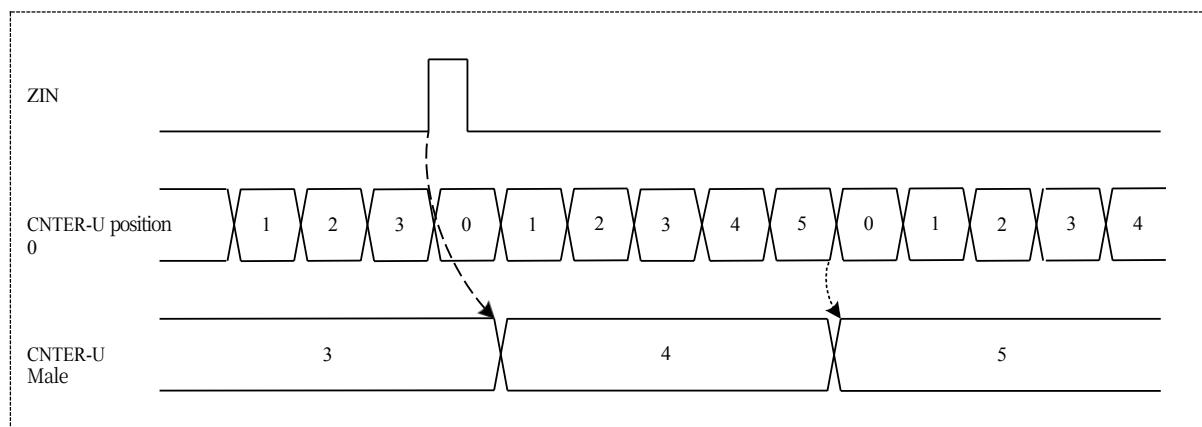


Figure 18-26 Rotation Counting Mode-Mixed Counting

18.3.11.3 Z-phase action shielding

In the Z-phase counting function or the mixed counting function of the revolution counting mode, it is possible to set that the valid input of ZIN is blocked for a few cycles after the overflow or underflow of the position timer (set by GCONR.ZMSKVAL[0:1]) and that the counting of the revolution counting unit and the clearing of the position counting unit are not performed. This function can only be realized by combining Unit 1 and Unit2, with Unit 1 as the position counter unit and Unit 2 as the revolution counter unit.

ZMSKPOS of the general-purpose 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 cycles of Z-phase masking is set by GCONR.ZMSKVAL; and the Z-phase masking function of the revolution counting unit is enabled when GCONR.ZMSKREV of the general-purpose control register (GCONR) of the revolution counting unit is 1. ZMSKVAL is set by GCONR.

Fig. 18-27 shows that in the case of mixed counting in the revolution counting mode, when there is ZIN phase input within 4 counting cycles after the position counting unit counts the overflow, the action of ZIN phase input is invalid, i.e., the revolution counting unit does not count, and the position counting unit is not cleared to zero; and after that, the ZIN phase input comes back to normal action.

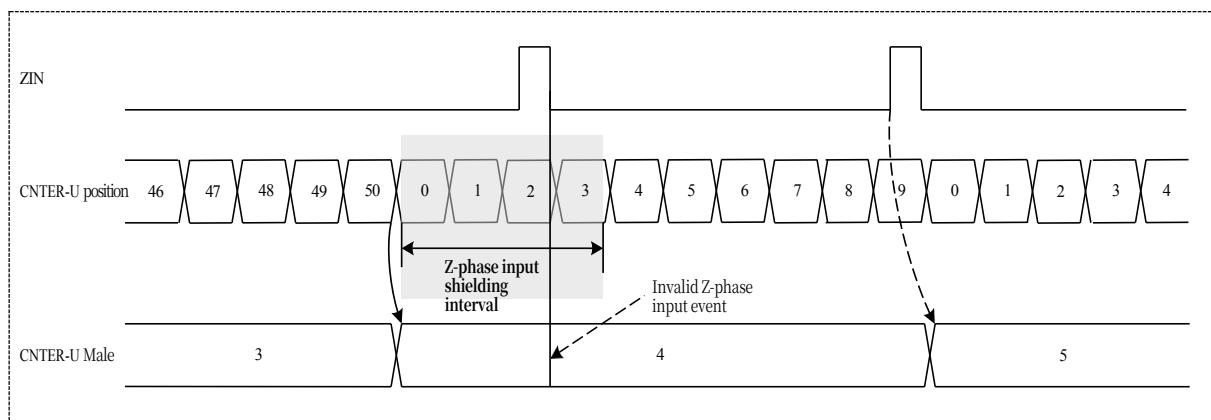


Figure 18-27 Rotation Counting Mode-Mixed Count Z Phase Mask Action Example 1

Fig. 18-28 shows that during mixed counting in the revolution counting mode, the counting direction changes in the 3rd cycle after the position counting unit overflows, and then the set shielding period of 4 cycles becomes invalid (the ZIN phase shielding function actually maintains for 3 cycles) and counting starts downward. After a count overflow occurs in the Position Counting Unit, the ZIN phase blocking function is turned on again and becomes invalid after 4 cycles. During the ZIN phase blocking period, the ZIN phase input function is invalid, i.e., the revolution counting unit does not count and the position counting unit is not cleared to zero; after that, the ZIN phase input comes back to operate normally.

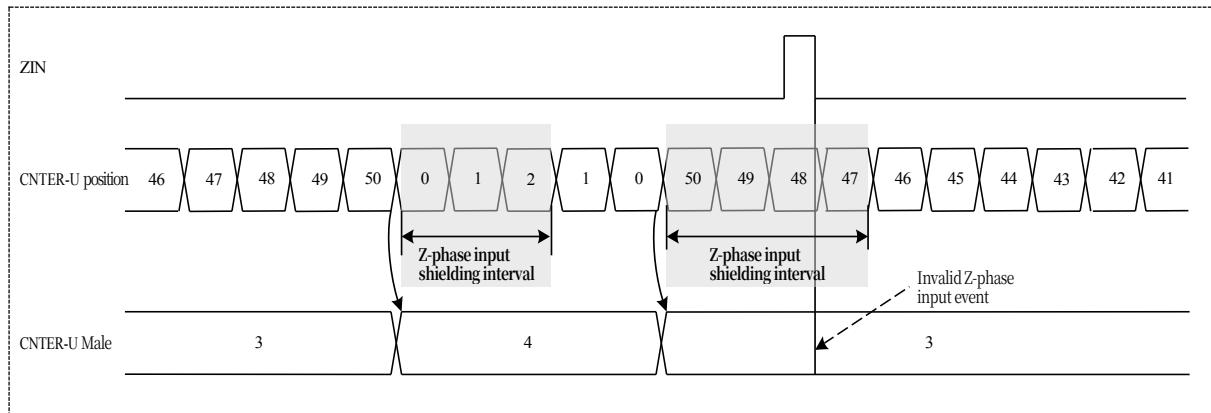


Figure 18-28 Rotation Counting Mode-Mixed Count Z Phase Mask Action Example 2

18.3.12 periodic interval response

The two dedicated comparison reference value registers (SCMAR, SCMBR) of Timer6 can output dedicated comparison matching interrupt A signal and dedicated comparison matching interrupt B signal to INTC to generate corresponding interrupt when counting comparison matching; at the same time, they can output dedicated comparison matching event A signal and dedicated comparison matching event B signal respectively, which can be used to correlate the action with other modules, and are mostly used to start the ADC, etc. At the same time, the dedicated compare and match event A signal and the dedicated compare and match event B signal can be output separately for correlation with other modules, which are mostly used to start ADC, etc.

The request signal for this interrupt and event can generate a valid request signal after every few cycles, i.e., cycle interval response is realized. This function is enabled by setting the VPERR.PCNTE[1:0] bits and VPERR.SPPERIA/B bits of the Valid Period Register (VPERR) PCNTS[2:0] bits to specify the number of cycles at which the request signal is valid. A valid request signal will not be output during other cycles even if the count value is equal to the value of the dedicated comparison reference value register SCMAR or SCMBR.

If you stop and restart the timer while using the cycle interval response function, configure VPERR.PCNTE[1:0]=00 before stopping the timer, otherwise the moment at which the cycle interval valid request signal is generated for the first time after restarting may deviate.

When this function is valid, the cycle match interrupt and cycle match event in each waveform mode are also output only in the valid cycle of the dedicated compare match interrupt and event output (the cycle with STFLR.VPERNUM=0 in the figure below). Figure 18-29 shows an example of the action of the cycle interval valid request signal.

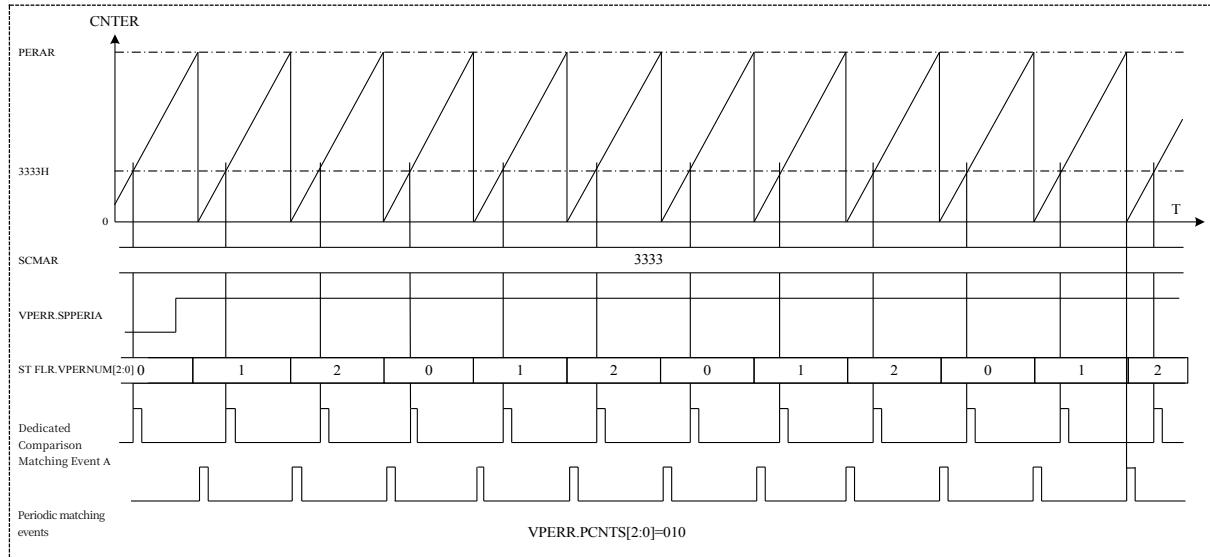


Figure 18-29 Cycle Interval Valid Request Signal Action

18.3.13 EMB

Control

Timer6 provides protection control over the output state of the port, fixing the port state to a predefined safe state in the event of an exception. All units have a common port output control interface, which is connected to a group of EMB events output by the EMB module. At the same time, the abnormal condition events selected on the interface can be set from the EMB side (see section "Emergency Brake Module (EMB)") and when abnormal conditions are detected on these interfaces, control of the general-purpose PWM outputs can be realized.

If the port monitors an EMB event coming from the EMB during normal output, the output state of the port can change to a preset state. The general-purpose PWM output port can change the port state to output high resistance state, output low level, or output high level (determined by the setting of PCONR.EMBVALA, PCONR.EMBVALB) when an EMB abnormal event occurs. For example, if PCONR.EMBVALA=01 setting

Timing, then if an EMB event is generated during the normal output of the TIM6_<t>_PWMA port, the output on the TIM6_<t>_PWMA port changes to a high resistance state.

After the abnormal event selected by the EMB module disappears and the EMB module resets the corresponding event status bit, Timer6 will automatically release the protection state and turn into a normal PWM output at the next immediately adjacent cycle point (counting trough of triangle waveform, overflow point of sawtooth waveform, or underflow point of sawtooth waveform) thus realizing the Cycle By Cycle control of the PWM port.

18.3.14 Typical Application Examples

The following describes the basic settings of Timer6 related registers in several typical applications for users' reference.

18.3.14.1 Basic Counting and Interrupt Action

- b) Setting the required reference values, including generalized reference values (GCMAR~GCMFR) and specialized reference values. (SCMAR~SCMBR), etc.
- c) Set the desired interrupt enable bits, including the count up overflow interrupt (ICONR.INTENOVF) count down overflow interrupt

- (ICONR.INTENUDF) Count Match Interrupt (ICONR.INTENA~F, ICONR.INTENSAU, ICONR.INTENSAD, ICONR.INTENSBU, ICONR.INTENSBD), and so on.
- d) Setting the internal count clock division frequency (GCONR.CKDIV[2:0])
 - e) Setting the waveform mode (GCONR.MODE[2:0])
 - f) Setting the counting direction (required only in sawtooth wave mode GCONR.MODE[2:0]=000)
 - g) Start counter (GCONR.START=1)

18.3.14.2 Comparison output and interrupt action

- a) Setting the generalized periodic reference value (PERAR)
- b) Sets the comparison reference value for each channel, including Generalized Comparison Reference Value A (GCMAR) and Generalized Comparison Reference Value B (GCMBR).
- c) Set the desired interrupt enable bits, including the count up overflow interrupt (ICONR.INTENOVF) count down overflow interrupt (ICONR.INTENUDF) Count Match Interrupt (ICONR.INTENA~B), etc.
- d) Setting the port output state of each channel in different counting states (refer to bit7~bit1 and bit23~bit17 of PCONR for related control).
- e) Setting the internal count clock division frequency (GCONR.CKDIV[2:0])
- f) Setting the waveform mode (GCONR.MODE[2:0])
- g) Setting the counting direction (required only in sawtooth wave mode GCONR.MODE[2:0]=000)
- h) Setting the comparison output mode for each channel (PCONR.CAPMDA=0, PCONR.CAPMDB=0)
- i) Setting each channel output enable (PCONR.OUTENA=1, PCONR.OUTENB=1)
- j) Start counter (GCONR.START=1)

18.3.14.3 Capture inputs and interrupt actions

- a) Setting the generalized periodic reference value (PERAR)
- b) Set the desired interrupt enable bits, including the count up overflow interrupt (ICONR.INTENOVF) count down overflow interrupt (ICONR.INTENUDF) Capture Input Interrupt (ICONR.INTENA~B), etc.
- c) Set the capture input external conditions for each channel (refer to all valid control bits of HCPAR or HCPBR. The valid control bits are independent of each other, and more than one can be selected at the same time to be the capture input condition for a channel.)
- d) Setting the internal count clock division frequency (GCONR.CKDIV[2:0])
- e) Setting the waveform mode (GCONR.MODE[2:0])

- f) Setting the counting direction (required only in sawtooth wave mode
GCONR.MODE[2:0]=000)
- g) Setting the capture input mode (PCONR.CAPMDA=1, PCONR.CAPMDB=1)
- h) Start counter (GCONR.START=1)
- i) Wait for a capture input condition to be generated, read the capture input value of the corresponding channel (GCMAR or GCMBR) or wait for the corresponding interrupt to be generated.

18.3.14.4 Cache transfer action (cycle reference value)

- a) Setting the required generalized cycle reference values (PERAR, PERBR, PERCR)
- b) Setting the single and double cache transfer method (BCONR.BSEP)
- c) Setting the internal count clock division frequency (GCONR.CKDIV[2:0])
- d) Setting the waveform mode (GCONR.MODE[2:0])(the cache transfer time point differs for different waveform modes)
- e) Setting the counting direction (required only in sawtooth wave mode GCONR.MODE[2:0]=000)
- f) Setting cache function active (BCONR.BENP=1)
- g) Start counter (GCONR.START=1)
- h) Waiting for the corresponding cache transfer time point, cache action occurs (PERBR->PERAR (when BCONR.BSEP=0)PERCR->PERBR->PERAR (when BCONR.BSEP=1))

18.3.14.5 Cache transfer action (generic baseline value)

- a) Setting of the required common comparator values (GCMAR, GCMCR, GCMER, GCMBR, GCMDR, GCMFR)
- b) Setting the single and double buffer transmission method for each channel (BCONR.BSEA, BCONR.BSEB)
- c) Setting the internal count clock division frequency (GCONR.CKDIV[2:0])
- d) Setting the waveform mode (GCONR.MODE[2:0])(the cache transfer time point differs for different waveform modes)
- e) Setting the counting direction (required only in sawtooth wave mode GCONR.MODE[2:0]=000)
- f) Setting each channel cache function effective (BCONR.BENA=1, BCONR.BENB=1)
- g) Start counter (GCONR.START=1)
- h) Wait for the corresponding cache transmission time point set by each channel to generate a cache action (GCMCR->GCMAR).
(when BCONR.BSEA=0), GCMER->GCMCR->GCMAR (when BCONR.BSEA=1),
GCMRD->GCMBR (when BCONR.BSEB=0), GCMFR->GCMRD->GCMBR (when
BCONR.BSEB=1), GCMFR->GCMRD->GCMBR (when BCONR.BSEB=1), GCMRD->GCMRD->GCMBR (when BCONR.BSEB=1)
time)

18.3.14.6 Cache transfer action (dedicated baseline value)

- a) Setting the required specialized comparison reference values (SCMAR, SCMCR, SCMER, SCMBR, SCMDR, SCMFR)

- b) Setting the single and double cache transfer method for each channel (BCONR.BSESPA, BCONR.BSESPB)
 - c) Setting the cache transfer time point for each channel (BCONR.BTRSPA[1:0], BCONR.BTRSPB[1:0])
 - d) Setting the internal count clock division frequency (GCONR.CKDIV[2:0])
 - e) Setting the waveform mode (GCONR.MODE[2:0])
 - f) Setting the counting direction (required only in sawtooth wave mode GCONR.MODE[2:0]=000)
 - g) Setting each channel cache function effective (BCONR.BENSPA=1, BCONR.BENSPB=1)
 - h) Start counter (GCONR.START=1)
 - i) Wait for the corresponding cache transmission time point set by each channel to generate a cache action (SCMCR->SCMAR).
- (when BCONR.BSESPA=0) SCMER->SCMCR->SCMAR (when BCONR.BSESPA=1)

SCMDR->SCMBR (BCONR.BSESPB=0 BSESPB=0), SCMFR->SCMDR->SCMBR
(When BCONR.BSESPB=1)

18.3.14.7 Cache transfer action (deadband reference value)

- a) Setting the desired dead time reference value (DTUAR, DTUBR, DTDAR, DTDBR)
- b) Setting the internal count clock division frequency (GCONR.CKDIV[2:0])
- c) Setting the waveform mode (GCONR.MODE[2:0])
- d) Setting the counting direction (required only in sawtooth wave mode GCONR.MODE[2:0]=000)
- e) Setting cache function active (DCONR.DTBENU=1, DCONR.DTBEND=1)
- f) Set hardware deadband function active (DCONR.DTCEN=1)
- g) Start counter (GCONR.START=1)
- h) Waiting for the corresponding cache transfer time point for a cache action to occur (DTUBR->DTUAR, DTDBR->DTDAR)

18.3.14.8 Synchronized start-up action (software method)

- a) Refer to steps a~f in the "Basic Counting and Interrupt Action" section to set each unit to be synchronized.
- b) Synchronized startup counter (set the corresponding bit of the SSTAR register to 1, one register bit per cell)

18.3.14.9 Synchronized start-up action (hardware method)

- a) Setting the generalized periodic reference value (PERAR)
- b) Setting the required reference values, including generalized reference values (GCMAR~GCMFR) and specialized reference values. (SCMAR~SCMBR), etc.
- c) Set the desired interrupt enable bits, including the count overflow interrupt (ICONR.INTENOVF) count underflow interrupt (ICONR.INTENUDF) Count Match Interrupt (ICONR.INTENA~F, ICONR.INTENSAU, ICONR.INTENSAD, ICONR.INTENSBU, ICONR.INTENSBD), and so on.
- d) Set hardware startup conditions (selected via HSTAR.HSTAx, x=0~1, 8~11)
- e) Set hardware startup enable (HSTAR.STARTS=1)
- f) Repeat steps a to e above for each unit to be synchronized (the setting in step d should be the same for each unit to be synchronized).
- g) Wait for the set trigger event to be generated to confirm the synchronized start of each unit's counter.

18.3.14.10 Orthogonal coded counting action (2 phases)

- a) Setting the generalized periodic reference value (PERAR)
- b) Setting the required reference values, including generalized reference values (GCMAR~GCMFR) and specialized reference values. (SCMAR~SCMBR), etc.
- c) Set the desired interrupt enable bits, including the count up overflow interrupt (ICONR.INTENOVF) count down overflow interrupt (ICONR.INTENUDF) count match interrupt (ICONR.INTENA~F, ICONR.INTENSAU,

ICONR.INTENSAD, ICONR.INTENSBU, ICONR.INTENSBD), etc.

- d) Set the desired hardware up count condition (selected via HCUPR.HCUPx, x=0~7)
- e) Set the desired hardware down count condition (selected via HCDOR.HCDOx, x=0~7)
- f) Start counter (GCONR.START=1)
- g) Wait for the set quadrature code count event to be generated to confirm that the counter is counting normally

18.3.14.11 Orthogonal coded counting action (3 phases)

- a) Set the position counting unit by referring to steps a~e in the section "Quadrature code counting action (2 phases)".
- b) Setting the hardware clear condition of the position counting unit (selected by HCLRR.HCLR_x, x=8~11)
- c) Setting the hardware clear enable of the position counting unit (HCLRR.CLEARS=1)
- d) Setting the general-purpose cycle reference value (PERAR) of the revolution counting unit
- e) Setting the comparison reference value of the revolution counting unit, including the generalized comparison reference value (GCMAR~GCMFR) and the specialized comparison reference value.
(SCMAR~SCMBR), etc.
- f) Set the interrupt enable bits required by the revolution counting unit, including the 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) Setting of hardware up count condition 1 (ZIN phase input) of the revolution counter unit (selected by HCUPR.HCUP_x, x=8~11, the set event here should be the same as the event set by the position counter unit in step b)
- h) Setting the hardware up count condition of the revolution counter unit 2 (overflow event input of the position counter unit)(internal hardware trigger event 0 selected via HCUPR.HCUP16)
- i) Setting of the hardware down count condition of the revolution counter unit (overflow event input of the position counter unit)
(internal hardware trigger event 1 selected via HCDOR.HCDO17)
- j) Set the trigger source number in HTSSR0 to the count overflow event of the position counting unit (refer to the INTC section for this overflow event number)
- k) Set the trigger source number in HTSSR1 to the count underflow event of the position counting unit (refer to the INTC section for this underflow event number).
- l) Start the revolution counting unit counter (GCONR.START=1)
- m) Start position counting unit counter (GCONR.START=1)

-
- n) Wait for the set AIN, BIN, and ZIN phase count events to be generated to confirm that the counter is counting normally.

18.3.14.12 Single PWM output

- a) Refer to steps a~j in [Comparison Output and Interrupt Action] section. (The output states of the two PWM channels TIM6_<t>_PWMA and TIM6_<t>_PWMB can be set independently in each unit to form two unrelated single PWM outputs).

18.3.14.13 Complementary PWM output (software deadband)

- a) Setting the generalized periodic reference value (PERAR)
- b) Setting of Generalized Comparator Value A (GCMAR) Generalized Comparator Value B (GCMBR)
- c) Set the desired interrupt enable bits, including the count up overflow interrupt (ICONR.INTENOVF) count down overflow interrupt (ICONR.INTENUDF) Count Match Interrupt (ICONR.INTENA~B), etc.
- d) Setting the output state of the port in different counting states (refer to bit7~bit1 and bit23~bit16 of PCONR, and combine with the setting values of GCMAR and GCMBR, it is necessary to make sure that the 2 PWM outputs form a complementary deadband).
- e) Setting the internal count clock division frequency (GCONR.CKDIV[2:0])
- f) Setting the waveform mode to triangular waveform mode (GCONR.MODE=100 or 101)
- g) Setting the comparison output mode (PCONR.CAPMDA=0, PCONR.CAPMDB=0)
- h) Setting output enable (PCONR.OUTENA=1, PCONR.OUTENB=1)
- i) Start counter (GCONR.START=1)

18.3.14.14 Complementary PWM output (hardware deadband)

- a) Setting the generalized periodic reference value (PERAR)
- b) Setting of generalized comparison reference value A (GCMAR) dead time reference value (DTUAR, DTDAR)
- c) Set the desired interrupt enable bits, including the count up overflow interrupt (ICONR.INTENOVF) count down overflow interrupt (ICONR.INTENUDF) Count Matching Interrupt (ICONR.INTENA~B) Deadband Error Interrupt (ICONR.INTENDTE), etc.
- d) Setting the port output state in different counting states (refer to bit7~bit1, bit23~bit16 of PCONR, combined with the setting values of GCMAR, DTUAR and DTDAR, it is necessary to ensure that the 2 PWM outputs form a complementary dead zone)
- e) Setting the internal count clock division frequency (GCONR.CKDIV[2:0])
- f) Setting the waveform mode to triangle waveform mode (GCONR.MODE[2:0]=100 or 101)
- g) Setting the comparison output mode for each channel (PCONR.CAPMDA=0, PCONR.CAPMDB=0)
- h) Setting each channel output enable (PCONR.OUTENA=1, PCONR.OUTENB=1)
- i) Set hardware deadband function active (DCONR.DTCEN=1)

- j) Start counter (GCONR.START=1)

18.3.14.15 EMB Monitoring and Interrupt Action

- a) Set the complementary PWM output action by referring to steps a~h in the [Complementary PWM Output (Software Deadband) section or steps a~i in the [Complementary PWM Output (Hardware Deadband) section.
- b) Sets the state of the PWM port (PCONR.EMBVALA, PCONR.EMBVAB) when an EMB event occurs (select the appropriate protection state depending on the system application)
- c) Setting the registers related to the EMB module (including the EMB interrupt license register (EMB_INTEN0) EMB control register 0 (EMB_CTL0), etc.)
- d) Start counter (GCONR.START=1) EMB module monitors system status in real time

18.3.15 Functional summary table

The summary table of the main functions in the sawtooth wave mode and triangle wave A and B modes of Timer6 is shown in the table below.

Table 18-3 Comparison of Functions in Different Modes

PWM output function			sawtoo th wave	triangle wave		Corresponding register setting (X = A, B)	note
				triangle wave A	triangle wave B		
Independent PWM exports	port state control	Port Input at Startup deciding whether or not to proceed with a decision	be in favor of	be in favor of	be in favor of	PCONR.STACX PCONR.STASTPSX	
		Port output when stopped deciding whether or not to proceed with a decision	be in favor of	be in favor of	be in favor of	PCONR.STOPCX PCONR.STASTPSX	
		Compare Matching Clock End Port Output Setting	be in favor of	be in favor of	be in favor of	PCONR.CMPCX	
		Periodic Matching Clock End Port Output Setting	be in favor of	be in favor of	be in favor of	PCONR.PERCX	
	cached transport	periodic ben chmark	single-step survive	be in favor of	be in favor of	Control Bit: BCONR.BENP Baseline: PERAR, PERBR	
		dual cache		be in favor of	be in favor of	Control Bit: BCONR.BSEP Baseline value: PERAR, PERBR, PERCR	
		referenc	single cache	be in favor of	be in favor of	Control bits: BCONR.BENX Baseline: GCMAR, GCMCR GCMBR, GCMDR	Different cache

		e value	dual cache	be in favor of	be in favor of	be in favor of	Control Bit: BCONR.BSEX Baseline value: GCMAR, GCMCR, GCMER gcmbr, gcmdr, gcmfr	transfer points for delta wave A mode and delta wave B mode
		emergency braking			be in favor of	be in favor of	be in favor of	PCONR.EMBVALX
Complementary PWM exports	port state control	Port Input at Startup deciding whether or not to proceed with a decision			be in favor of	be in favor of	be in favor of	PCONR.STACX PCONR.STASTPSX
		Port output when stopped deciding whether or not to proceed with a decision			be in favor of	be in favor of	be in favor of	PCONR.STACX PCONR.STASTPSX
		Compare Matching Clock End Port Output Setting			be in favor of	be in favor of	be in favor of	PCONR.CMPCX
		Periodic Matching Clock End Port Output Setting			be in favor of	be in favor of	be in favor of	PCONR.PERCX
	(computing) cache transmission	cyclicality stand	single-step standard of reference	be in favor of	be in favor of	be in favor of	Control Bit: BCONR.BENP Baseline: PERAR, PERBR	

PWM output function			sawtooth wave	triangle wave		Corresponding register setting (X = A, B)	note
				triangle wave A	triangle wave B		
(be) worth	(be) worth	dual cache	be in favor of	be in favor of	be in favor of	Control Bit: BCONR.BSEP Baseline value: PERAR, PERBR, PERC	
		sing le cache	be in favor of	be in favor of	be in favor of	Control bits: BCONR.BENX Baseline: GCMAR, GCMCR GCMBR, GCMDR	Different cache transfer points for delta wave A mode and delta wave B mode
		dual cache	be in favor of	be in favor of	be in favor of	Control Bit: BCONR.BSEX Baseline value: GCMAR, GCMCR, GCMER gcmbr, gcmdr, gcmfr	
	Deadband reference value	sing le cache	be in favor of	be in favor of	be in favor of	Control Bits: DCONR.DTBENU DCONR.DTBEND Baseline: DTUAR, DTDAR DTUBR, DTDBR	
		Deadband-free PWM output		be in favor of	be in favor of	GCMAR=GCMBR	
		hardware way (of life)	be in favor of	be in favor of	be in favor of	GCMAR≠GCMBR	
		hardware method	unsupported	be in favor of	be in favor of	Control bits: DCONR.DTCEN Baseline value: GCMAR, DTUAR, DTDAR	
emergency braking		be in favor of	be in favor of	be in favor of	PCONR.EMBVALX		

18.4 Interrupt and Event Description

18.4.1 interrupt output

Timer6 contains 6 general-purpose count compare match interrupts (including 2 capture input interrupts) 2 specialized count compare match interrupts, 2 count cycle match interrupts, and 1 dead time error interrupt.

18.4.1.1 Count Compare Match Interrupt

There are 6 Generalized Comparison Reference Value Registers (GCMAR-GCMFR), which can be used to compare with the count value to generate a comparison match. When the count comparison is matched, the bits STFLR.CMAF~STFLR.CMFF in the Status Flag Register (STFLR) will be set to 1. If the corresponding bits INTENA~INTENF in the Interrupt Control Register (ICONR) are set to 1 to enable interrupt, the corresponding interrupt request will be sent to the interrupt control register (ICONR), and the interrupt request will be sent to the interrupt control register (ICONR).

(TMR6_U<t>_GCMA~F) is also triggered.

The capture input action occurs when the capture input valid condition selected by the hardware capture event selection registers (HCPARHCPBR) is generated. If the INTENA or INTENB bit of the Interrupt Control Register (ICONR) is set to 1 to enable the interrupt at this time, the corresponding interrupt request

(TMR6_U<t>_GCMA~B) is triggered.

The two dedicated comparison reference value registers (SCMAR-SCMBR) can also be used to generate a comparison match with the count value comparison respectively. When the count is matched, the STFLR.CMSPAF~CMSPBF bits in the Status Flag Register (STFLR) are set to 1. If the corresponding bit in INTENSSAU<D> or INTENSBU<D> of the Interrupt Control Register (ICONR) is set to 1 to enable interrupt, the corresponding interrupt request (TMR6_U<t>_SCMA~B) will also be triggered. TMR6_U<t>_SCMA~B) will also be triggered.

18.4.1.2 Count Cycle Matching Interrupt

The STFLR.OVFF or STFLR.UDFF bit of the Status Flag Register (STFLR) is set to 1 when the sawtooth wave increment count reaches the upper overflow point, the sawtooth wave decrement count reaches the lower overflow point, the delta wave count reaches the valley point, or the delta wave count reaches the peak point, and the interrupt cycle matching interrupt (TMR6_U<t>_GOVF and TMR6_U<t>_GUDF) can be triggered at the corresponding point in time by setting the ICONR.INTENOVF or ICONR. INTENUDF bit of the Interrupt Control Register (ICONR) is set to enable the interrupt, the count cycle matching interrupt (TMR6_U<t>_GOVF and TMR6_U<t>_GUDF) can be triggered at the corresponding point in time.

18.4.1.3 Dead Time Error Interrupt

If the value of the Dead Time Base Value Register (DTU<D>AR) is loaded into the Generalized Comparison Base Value Register (GCMBR) and the cycle limit is exceeded, a dead time error is generated and the STFLR.DTEF bit in the Status Flag Register (STFLR) is set to 1. If the INTENDTE bit in the Interrupt Control Register (ICONR) is set to enable an interrupt, a dead time error interrupt is triggered at that moment. If the INTENDTE bit of the Interrupt Control Register (ICONR) is set to enable the interrupt, a dead time error interrupt will be triggered at that time.

(TMR6_U<t>_GDTE)

18.4.2 event output

During the clock counting process, if a cycle matching event (overflow and underflow points of sawtooth wave, counting peaks or valleys of triangular wave) a general-purpose counting comparison matching event, or a special-purpose counting comparison matching event is generated, the corresponding event output signal will be generated, which is used for selecting to trigger another module.

The following figure shows the action examples of generalized compare-and-match interrupts A-F & events A-F, dedicated compare-and-match interrupts A-B & events A-B, and cycle-match interrupts & events for Unit 1.

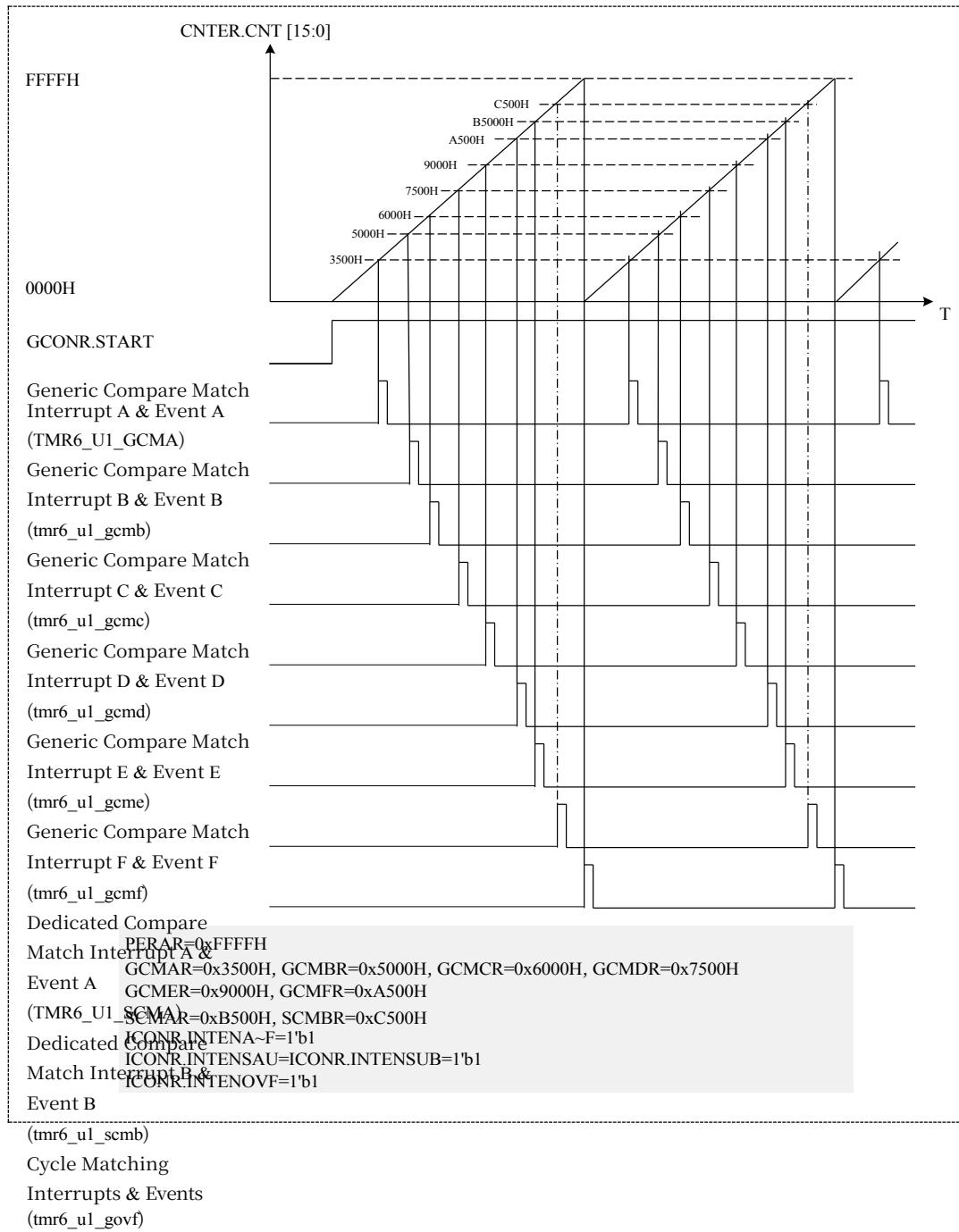


Figure 18-30 Interrupt & Event Output Example in Sawtooth Wave Mode

18.5 Register Description

Table 18-4 shows the register list of the Timer6 module.

BASE ADDR. 0x4001_8000 (U1) 0x4001_8400 (U2) 0x4001_8800 (U3)

Table 18-4 Register List

register name	notation	offset	bit width	reset value
General Purpose Count Register	TMR6_CNTER	0x0000	32	0x0000_0000
General Purpose Cycle Reference Value Register A	TMR6_PERAR	0x0004	32	0x0000_FFFF
General Purpose Cycle Reference Value Register B	TMR6_PERBR	0x0008	32	0x0000_FFFF
General purpose cycle reference value register C	TMR6_PERCR	0x000C	32	0x0000_FFFF
Generalized Comparison Reference Value Register A	TMR6_GCMAR	0x0010	32	0x0000_FFFF
Generalized Comparison Reference Value Register B	TMR6_GCMBR	0x0014	32	0x0000_FFFF
Generalized Comparison Reference Value Register C	TMR6_GCMCR	0x0018	32	0x0000_FFFF
Generalized Comparison Reference Value Register D	TMR6_GCMDR	0x001C	32	0x0000_FFFF
Generalized Comparison Reference Value Register E	TMR6_GCMER	0x0020	32	0x0000_FFFF
Generalized Comparison Reference Value Register F	TMR6_GCMFR	0x0024	32	0x0000_FFFF
Dedicated Comparison Reference Value Register A	TMR6_SCMAR	0x0028	32	0x0000_FFFF
Dedicated Comparison Reference Value Register B	TMR6_SCMBR	0x002C	32	0x0000_FFFF
Dedicated Comparison Reference Value Register C	TMR6_SCMCR	0x0030	32	0x0000_FFFF
Dedicated comparison reference value register D	TMR6_SCMDR	0x0034	32	0x0000_FFFF
Dedicated Comparison Reference Value Register E	TMR6_SCMER	0x0038	32	0x0000_FFFF
Dedicated Comparison Reference Value Register F	TMR6_SCMFR	0x003C	32	0x0000_FFFF
Dead time reference value register UA	TMR6_DTUAR	0x0040	32	0x0000_FFFF
Dead time reference register DA	TMR6_DTDAR	0x0044	32	0x0000_FFFF
Dead time reference register UB	TMR6_DTUBR	0x0048	32	0x0000_FFFF

Dead time reference value register DB	TMR6_DTDBR	0x004C	32	0x0000_FFFF
General Purpose Control Register	TMR6_GCONR	0x0050	32	0x0000_0100
Interrupt Control Register	TMR6_ICONR	0x0054	32	0x0000_0000
Port Control Register	TMR6_PCONR	0x0058	32	0x0000_0000
Cache Control Register	TMR6_BCONR	0x005C	32	0x0000_0000
Deadband Control Register	TMR6_DCONR	0x0060	32	0x0000_0000
Filter Control Register	TMR6_FCONR	0x0068	32	0x0000_0000
Efficient Cycle Register	TMR6_VPERR	0x006C	32	0x0000_0000
Status flag register	TMR6_STFLR	0x0070	32	0x8000_0000
Hardware Boot Event Selection Register	TMR6_HSTAR	0x0074	32	0x0000_0000
Hardware Stop Event Selection Register	TMR6_HSTPR	0x0078	32	0x0000_0000
Hardware clear event selection register	TMR6_HCLRR	0x007C	32	0x0000_0000
Hardware Capture Event Select Register A	TMR6_HCPAR	0x0080	32	0x0000_0000

register name	notation	offset	bit width	reset value
Hardware Capture Event Selection Register B	TMR6_HCPBR	0x0084	32	0x0000_0000
Hardware incremental event selection register	TMR6_HCUPR	0x0088	32	0x0000_0000
Hardware Decrement Event Select Register	TMR6_HCDOR	0x008C	32	0x0000_0000
Software Synchronized Startup Control Register	TMR6_SSTAR	(0x4001_83F4)	32	0x0000_0000
Software Synchronization Stop Control Register	TMR6_SSTPR	(0x4001_83F8)	32	0x0000_0000
Software Synchronized Zero Control Register	TMR6_SCLRR	(0x4001_83FC)	32	0x0000_0000

Attention:

-The software synchronization registers (TMR6_SSTAR, TMR6_SSTPR, TMR6_SCLRR) are three unit-independent registers common to all three unit Timer6s.

18.5.1 General purpose count value register (TMR6_CNTER)

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
CNT [15:0]															

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b16	Reserved	-	Reads out as "0."	R
b15~b0	CNT [15:0]	numerical value	Current timer count value	R/W

18.5.2 General Purpose Periodic Reference Value Register (TMR6_PERAR-PERCR)

Reset value: 0x0000_FFFF

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]															

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b16	Reserved	-	Reads out as "0."	R
b15~b0	PERA-C [15:0]	countervalue	Setting the counting period value and the corresponding buffer value for each round of counting	R/W

18.5.3 Generalized Comparison Reference Value Register (TMR6_GCMAR-GCMFR)

Reset value: 0x0000_FFFF

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]															

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b16	Reserved	-	Reads out as "0."	R
b15~b0	GCMA-F [15:0]	Counting benchmark values	Comparison reference value setting, matching signal valid when equal to the count value	R/W

18.5.4 Dedicated Comparison Reference Value Register (TMR6_SCMAR-SCMFR)

Reset value: 0x0000_FFFF

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]															

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b16	Reserved	-	Reads out as "0."	R
b15~b0	SCMA-F [15:0]	Dedicated benchmark value	Setting the comparison reference value and buffer value	R/W

18.5.5 Dead Time Reference Value Register (TMR6_DTU<D>AR)

Reset value: 0x0000_FFFF

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]															

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b16	Reserved	-	Reads out as "0."	R
b15~b0	DTU/DA-B [15:0]	Dead time value	Dead Time Setting Value and Cache Value	R/W

18.5.6 General Purpose Control Register (TMR6_GCONR)

Reset value: 0x0000_0100

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

b0

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b20	Reserved	-	Reads "0", writes "0".	R/W
b19~b18 shielding week	ZMSKVAL[1:0]	Z-phase input phase	Quadrature encoded Z-phase input masked counting period value 00: Z-phase input shielding function is invalid 01: Z-phase input blocked for 4 counting cycles after position count overflow or underflow 10: Z-phase input masked for 8 counting cycles after position count overflow or underflow 11: Z-phase input masked for 16 counting cycles after position count overflow or underflow	R/W
b17	ZMSKPOS	Z phase input position timer selection	0: The unit acts as a position timer during Z-phase input, and the position timer clearing function operates normally during the blocking cycle.	R/W
b16	ZMSKREV	Z-phase input metric timer selection	1: The unit acts as a position timer during Z-phase input, and the position timer clearing function is blocked during the blocking cycle. 0: The unit acts as a rotary timer during Z-phase input, and the rotary timer counting function operates normally during the blocking cycle. 1: The unit acts as a rotary timer during Z-phase input, and the rotary timer counting function is blocked during the blocking cycle.	R/W
b15~b9	Reserved	-Reserved	Read "0", write "0".	R/W
b8	DIR	Counting direction 0: Decreasing count 1: Increasing count	0: Decreasing count	R/W
b7	Reserved	-Reserved	Read "0", write "0".	R/W
b6~b4 Selection	CKDIV[2:0]	Count Clock	000: PCLK0 001: PCLK0/2 010: PCLK0/4 011: PCLK0/8 100: PCLK0/16 101: PCLK0/64 110: PCLK0/256 111: PCLK0/1024	000: Sawtooth wave mode

b3~b1	MODE[2:0]	counting mode	100: Triangular wave A mode 101: Triangle wave B mode Please do not set other values	R/W
b0	START	timer activation	0: Timer off 1: Timer start Note: This bit automatically changes to 0 when a software stop condition or a hardware stop condition is in effect	R/W

18.5.7 Interrupt Control Register (TMR6_ICONR)

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	INT EN DTE	INT EN UDF	INT EN OVF	INT EN F	INT EN E	INT EN D	INT EN C	INT EN B	INT EN A

		classifier for marking	Bit Name	Function	Read/Write											
		honorable people														
b31~b20	Reserved	-0" for reading, "0" for writing. "0", writes "0".			R/W											Reads
b19	INTENSBD		0: The interrupt is invalid when the SCMBR register and the count value are equal during the downward counting period													
b18	INTENSBU	Dedicated down count interrupt enable B				1: This interrupt is enabled when the SCMBR register and the count value are equal during downward counting.										
b17	INTENSAD		Note: This bit is also enabled by this bit when used as a dedicated compare match event output.													
b16	INTENSAU	0: The interrupt is invalidated when the SCMAR register and the count value are equal during the count up period														
b15~b9	Reserved	-0" for reading, "0" for writing. "0", writes "0".			R/W											Reads
b8	INTENDTE	Dead time error interrupt enable	In case of dead time error, this interrupt is disabled													0:
b7	INTENUDF	Underflow interrupt enable	underflow occurs during sawtooth wave or triangular wave counts to the valley point, this interrupt is disabled													0:
			1: count to valley when underflow occurs during sawtooth wave or triangle wave, this interrupt is enabled													

b6	INTENOVF	Overflow interrupt enable when counting to the peak during triangle wave.	0: The interrupt is invalidated when overflow occurs during sawtooth wave or 1: The interrupt is enabled when overflow occurs during a sawtooth wave or when the count reaches the peak during a triangle wave. R/W
b5	INTENF	Count Match Interrupt Enable F interrupt is disabled when the GCMFR register is equal to the count value 1: This interrupt is enabled when the GCMFR register is equal to the count value	0: This R/W
b4	INTENE	Count Match Interrupt Enable E interrupt is disabled when the GCMER register is equal to the count value 1: This interrupt is enabled when the GCMER register is equal to the count value	0: The R/W
b3	INTEND	Count Match Interrupt Enable D This interrupt is disabled when the GCMDR register is equal to the count value 1: This interrupt is enabled when the GCMDR register is equal to the count value R/W	0: R/W
b2	INTENC	Count Match Interrupt Enable C This interrupt is disabled when the GCMCR register is equal to the count value 1: This interrupt is enabled when the GCMCR register is equal to the count value R/W	0: R/W
b1	INTENB	Count Match Interrupt Enable B Invalid R/W 1: When the GCMBR register is equal to the count value, or when a capture input event occurs, the interrupt enable	0: When the GCMBR register is equal to the count value, or when a capture input event occurs, the interrupt enable Interrupt
b0	INTENA	Count Match Interrupt Enable A When the GCMAR register is equal to the count value, or when a capture input event occurs, this interrupt is disabled R/W	0: R/W

1: This interrupt is enabled when the GCMAR register is equal to
the count value, or when a capture input event occurs

18.5.8 Port Control Register (TMR6_PCONR)

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17
-	-	-	EMBVAL B[1:0]	-	-	OUT ENB	PER CB[1:0]	CMP CB[1:0]	STASTP SB	STP CB	STA CB	CAP MDB		
-	-	-	EMBVAL A[1:0]	-	-	OUT ENA	PER CA[1:0]	CMP CA[1:0]	STASTP SA	STP CA	STA CA	CAP MDA		

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b29	Reserved	-	Reads "0" and writes "0".	R/W
b28~b27 Control B	EMBVALB[1:0]	EMB Status	00: TIM6_<t>_PWMB port outputs normally when EMB condition holds 01: TIM6_<t>_PWMB port outputs high resistance state when EMB condition holds 10: TIM6_<t>_PWMB port outputs low when EMB condition holds 11: TIM6_<t>_PWMB port outputs high when the EMB condition is established	R/W
b26~b25	Reserved	-Reserved	Read "0", write "0".	R/W
b24 function	OUTENB	Output enable B	0: Invalid TIM6_<t>_PWMB port output during Timer6 1: TIM6_<t>_PWMB port output valid for Timer6 function 00: TIM6_<t>_PWMB port output set low when timer count value is equal to cycle value 01: TIM6_<t>_PWMB port output when timer count value is equal to cycle value	R/W
b23~b22 period value matches	PERCB[1:0]	Port when Status setting B	Port B Status Selection Counting Stop Port Status Setting B Count start port state setting B	Set to high level 10: When the timer count value is equal to the cycle value, the TIM6_<t>_PWMB port output maintains the previous state 11: When the timer count value is equal to the cycle value, the TIM6_<t>_PWMB port output is set to the inverted level 00:
b21~b20 comparing value match	CMPBC[1:0]	Port when Status setting B	TIM6_<t>_PWM	TIM6_<t>_PWMB port output is set to the inverted level 00:
b19	STASTPSB	Count Start Stop		TIM6_<t>_PWM

01: TIM6_<t>_PWMB port output set high when timer count value is equal to GCMBR

10: When the timer count value is equal to GCMBR, the TIM6_<t>_PWMB port output maintains the previous state

11: When the timer count value is equal to GCMBR, the TIM6_<t>_PWMB port output is set to inverted level

0: When counting starts or stops, TIM6_<t>_PWMB port output is determined by STACB, STPCB

R/W

1: When counting starts or stops, the TIM6_<t>_PWMB port output maintains the previous state

Note: Count start here means initial count start or stop and start again; count stop means stop at the beginning or stop after count start

0: TIM6_<t>_PWMB port output set low when counting is stopped

R/W

1: TIM6_<t>_PWMB port output set high when counting stops

R/W R/W

0: TIM6_<t>_PWMB port output set low when counting begins

1: TIM6_<t>_PWMB port output set high when counting starts

0:

R/W

b16 CAPMDB Comparison output function

Function mode selection B

1: Capture input function

b15~b13	Reserved	-Reserved	Read "0", write "0".	R/W
b12~b11 Control A	EMBVALA[1:0]	EMB Status	00: TIM6_<t>_PWMA port outputs normally when EMB condition holds 01: TIM6_<t>_PWMA port outputs high resistance state when EMB condition holds 10: TIM6_<t>_PWMA port outputs low when EMB condition holds 11: TIM6_<t>_PWMA port outputs high when the EMB condition is established	R/W
b10~b9	Reserved	-Reserved	Read "0", write "0".	R/W
b8	OUTENA	Output enable A	0: Invalid TIM6_<t>_PWMA port output for Timer6 function 1: TIM6_<t>_PWMA port output valid for Timer6 function	R/W
b7~b6 period value matches	PERCA[1:0]	Port when Status setting A	00: TIM6_<t>_PWMA port output set low when timer count value is equal to cycle value 01: TIM6_<t>_PWMA port output when timer count value is equal to cycle value Set to high level 10: When the timer count value is equal to the cycle value, the TIM6_<t>_PWMA port output maintains the previous state 11: TIM6_<t>_PWMA port output set to invert level when timer count value is equal to cycle value 00: TIM6_<t>_PWMA port output set low when timer count value is equal to GCMAR 01: TIM6_<t>_PWMA port output set high when timer count value is equal to GCMAR	R/W
b5~b4 comparison value matches	CMPCA[1:0]	Port when Status setting A	10: TIM6_<t>_PWMA port output stays in previous state when timer count value is equal to GCMAR 11: TIM6_<t>_PWMA port output set to invert level when timer count value is equal to GCMAR 0: When counting starts or stops, TIM6_<t>_PWMA port output is determined by STACA, STPCA 1: When counting starts or stops, the TIM6_<t>_PWMA port output maintains the previous state	R/W
b3	STASTPSA	Count Start Stop Port Status Selection A	Note: Count start here means initial count start or stop and start again; count stop means stop at the beginning or stop after count start 0: TIM6_<t>_PWMA port output set low when counting is stopped 1: TIM6_<t>_PWMA port output set high when counting stops	R/W
b2	STPCA	Count Stop Port Status Setting A	0: TIM6_<t>_PWMA port output set low at start of counting 1: TIM6_<t>_PWMA port output set high when counting starts	R/W
b1	STACA	Count start port state setting A	0:	
b0	CAPMDA Comparison output function	Function mode selection A	0: 1: Capture input function	R/W

18.5.9 Cache Control Register (TMR6_BCONR)

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	
-	-	BTRSPB[1:0]	-	-	BSE SPB	BEN SPB	-	-	BTRSPA[1:0]	-	-	BSE spa	BEN spa		
b16 b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	
-	-	-	-	-	-	BSE P	BEN P	-	-	-	-	BSE B	BEN B	BSE A	BEN A
					b0										

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b30	Reserved	-	Reads "0", writes "0".	R/W
b29~28	BTRSPB[1:0]	Dedicated baseline value cache Transmission time point B	sawtooth wave 00: No transmission Other than 00: Transmission in case of overflow or underflow triangle wave 00: No transmission 01: Cache transfer when counting to peak point 10: Cache transfer when count reaches valley point 11: Cache transfer when counting to peaks or valleys	R/W
b27-b26	Reserved	-	Reads "0", writes "0".	R/W
b25	BSESPB	Dedicated Comparison Reference Value Cache Transfer Option B	0: Single Cache Transfer (SCMDR->SCMBR) 1: Dual-cache transfer (SCMFR->SCMDR->SCMBR)	R/W
b24	BENSPB	Dedicated baseline value cache Transmission B	0: Cache transfer invalid 1: Cache transfer enable	R/W
b23~b22	Reserved	-	Reads "0", writes "0".	R/W
b21~b20	BTRSPA[1:0]	Dedicated baseline value cache Transmission time point A	sawtooth wave 00: No transmission Other than 00: Transmission in case of overflow or underflow triangle wave 00: No transmission 01: Cache transfer when counting to peaks 10: Cache transfer when count reaches valley point 11: Cache transfer when counting to peaks or	R/W

b19~b18	Reserved	-	Reads "0", writes "0".	R/W
b17	BSESPA	Dedicated Comparison Reference Value Cache Transfer Option A	0: Single cache transfer (SCMCR->SCMAR) 1: Double-cache transfer (SCMER->SCMCR->SCMAR)	R/W
b16	BENSPA	Dedicated Comparison Reference Value Cache Cache transfer invalid Cache transfer enabled	Transmit A R/W	0: 1:
b15~b10	Reserved	-0" for reading, "0" for writing. writes "0".		Reads "0", R/W
b9	BSEP	Periodic value cache transfer options Dual-cache transfer (PERCR->PERBR->PERAR) Note: The transmission time point has nothing to do with the counting mode, but only at the upflow point of the sawtooth wave, the downflow point of the sawtooth wave, and the downflow point of the sawtooth wave.	Dots or troughs of triangular waves	1:
b8	BENP	Periodic value cache transfer transfer invalid	R/W 1: Cache transfer enable	0: Cache
b7~b4	Reserved	-0" for reading, "0" for writing. writes "0".		Reads "0", R/W

			When comparing output functions: 0: single cache transfer (GCMDR->GCMBR) 1: Dual-cache transfer (GCMFR->GCMDR->GCMBR)	
b3	BSEB	Generalized Comparison Value Cache Transfer Selection B	When the input function is captured: 0: single cache transfer (GCMBR->GCMDR) 1: Dual-cache transfer (GCMBR->GCMDR->GCMFR)	R/W
b2	BENB	Generic Comparison Value Cache Transmission B 0: Cache transfer disabled 1: Cache transfer enabled		R/W
		When comparing output functions: 0: single cache transfer (GCMCR->GCMAR) 1: Double-cache transfer (GCMER->GCMCR->GCMAR)		
b1	BSEA	Generalized Comparison Value Cache Transfer Selection A	When the input function is captured: 0: Single cache transfer (GCMAR->GCMCR) 1: Double-cache transfer (GCMAR->GCMCR->GCMER)	R/W
b0	BENA	Generic Comparison Value Cache Transmission A 0: Cache transfer disabled 1: Cache transfer enabled		R/W

18.5.10 Deadband Control Register (TMR6_DCONR)

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							SEPA	-	-	DTB END	DTB ENU	-	-	-	DTC EN

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b9	Reserved	-	Reads "0" and writes "0".	R/W
b8	SEPA	Separation setting	0: DTUAR and DTDAR set separately 1: The value of DTDAR and the value of DTUAR are automatically equalized	R/W
b7~b6	Reserved	-Reserved	Read "0", write "0".	R/W
b5	DTBEND	Dead time value cache transfer D	0: Cache transfer invalid 1: Cache transfer enable (DTDBR->DTDAR)	R/W
b4	DTBENU	Dead Time Value Cache Transfer U	0: cache transfer invalid 1: Cache transfer enable (DTUBR->DTUAR)	R/W
b3~b1	Reserved	-Reserved	Read "0", write "0".	R/W
b0	DTCEN	Deadband function	0: Deadband function not valid 1: Deadband function valid	R/W

18.5.11 Filter Control Register (TMR6_FCONR)

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17
Reserved									NOFI CKTB [1:0]	NOFI ENTB	-	NOFI CKTA [1:0]	NOFI ENTA	
b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2
Reserved									NOFI CKGB[1:0]	NOFI ENGB	-	NOFI CKGA[1:0]	NOFI ENGA	
b0														

classifier for honorific people	marking	Bit Name	Function	Read/Write
b31~b23	Reserved	-0" for reading, "0" for writing. writes "0".		Reads "0", R/W
		00: PCLK0		
b22~b21	NOFICKTB [1:0]	Filter Sample Reference Clock Selection TB	01: PCLK0/4	
		R/W 10: PCLK0/16 11: PCLK0/64		
b20	NOFIENTB	Capture input port filtering TB TIM6_TRIGB port input filtering is disabled	0: R/W	
		1: TIM6_TRIGB port input filter function enable		
b19	Reserved	-0" for reading, "0" for writing. writes "0".		Reads "0", R/W
		00: PCLK0		
b18~b17	NOFICKTA [1:0]	Filter Sample Reference Clock Selection TA	01: PCLK0/4	
		R/W 10: PCLK0/16 11: PCLK0/64		
b16	NOFIENTA	Capture input port filtering TA TIM6_TRIGA port input filtering is disabled	0: R/W	
		1: TIM6_TRIGA port input filter function enable		
b15~b7	Reserved	-0" for reading, "0" for writing. writes "0".		Reads "0", R/W
		00: PCLK0		
b6~b5	NOFICKGB [1:0]	Filter Sample Reference Clock Selection GB	01: PCLK0/4	
		R/W 10: PCLK0/16 11: PCLK0/64		
b4	NOFIENGB	Capture input port filter GB	0: Invalid TIM6_<t>_PWMB input port filtering for this unit 1: This unit TIM6_<t>_PWMB input port filtering function enable	
		R/W		
b3	Reserved	-0" for reading, "0" for writing. writes "0".		Reads "0", R/W

		00: PCLK0	01: PCLK0/4
b2~b1	NOFICKGA [1:0]	Filter Sample Reference Clock Selection GA	
		R/W	
		10: PCLK0/16	
		11: PCLK0/64	
b0	NOFIENGA	Capture input port filter GA	0: Invalid TIM6_<t>_PWMA input port filtering for this unit 1: This unit TIM6_<t>_PWMA input port filtering function enable
			R/W

18.5.12 Valid cycle register (TMR6_VPERR)

Reset value: 0x0000_0000

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									

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b21	Reserved	-	Reads "0", writes "0".	R/W
			000: Invalid cycle selection function	
			001: valid every 1 cycle	
			010: valid every 2 cycles	
b20~b18	PCNTS [2:0]	Effective period selection	011: valid every 3 cycles 100: valid every 4 cycles 101: valid every 5 cycles 110: valid every 6 cycles 111: valid every 7 cycles	R/W
			00: Invalid cycle selection function	
b17~b16	PCNTE[1:0]	Effective cycle counting condition selection	01: Sawtooth counting up, down or triangle wave valley as counting condition 10: Sawtooth wave count up, down or triangle wave peak as R/W count condition 11: Sawtooth wave counts the upper and lower overflow points or triangular wave valleys and peaks as counting conditions.	
b15~b10	Reserved	-0" for reading, "0" for writing. writes "0".	Reads "0", R/W	
b9	SPPERIB	Dedicated signal effective period selection B	0: Effective period selection function disabled R/W 1: Valid cycle selection function enabled	
b8	SPPERIA	Dedicated signal effective period selection A	0: Effective period selection function is invalid R/W 1: Effective period selection function enabled	
b7~b0	Reserved	-0" for reading, "0" for writing. writes "0".	Reads "0", R/W	

18.5.13 Status Flag Register (TMR6_STFLR)

Reset 0x8000_0000

value. b29 b28 b27 b26 b25 b24 b23 b22 b21 b20 b19 b18 b17 b16

DIRF	Reserved								VPERNUM[2:0]		Reserved					
b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0																
-	-	-	CMS BDF	CMS BUF	CMS ADF	CMS AUF	DTE F	UDF F	OVF F	CMF F	CME F	CMD F	CMC F	CMB F	CMA F	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31	DIRF	direction of counting	0: Decreasing count 1: Incremental counting	R
b30-b24	Reserved	-	Reads "0", writes "0".	R
b23~b21	VPERNUM[2:0]	periodic number	Number of cycles after counting when the effective cycle selection function is enabled	R
b20~b13	Reserved	-	Reads "0", writes "0".	R
b12	CMSBDF	Downward Counting Dedicated Comparison Datum Match B	0: The value of the SCMBR register is not equal to the count value during downward counting	R/W
b11	CMSBUF	Count Up Specialized Comparison Reference Value Match B	1: When counting down, the value of the SCMBR register is equal to the count value	R/W
b10	CMSADF	Count Down Specialized Comparison Reference Value Match B	0: When counting up, the value of 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
b09	CMSAUF	Count Up Dedicated Comparison Base Reference Match A register value is not equal to the count up value when counting up Value Match A value	0: SCMAR register value is not equal to the count value when counting down 1: When counting down, the value of the SCMAR register is not equal to the count up value 1: When counting up, the value of the SCMAR register is equal to the count register is equal to the count value	0: SCMAR R/W
b8	DTEF	Dead time error	0: No dead time error occurred 1: Dead time error occurred	R
b7	UDFF	Underflow Matching	0: No sawtooth underflow or triangular wave counts to valley 1: sawtooth underflow or triangle count to valley occurs	R/W
b6	OVFF	Overflow Match occurs	0: No sawtooth wave overflow or triangle wave counting to the peak point 1: Sawtooth wave overflow occurs or triangle wave counts to the peak point	R/W
b5	CMFF	Count Match F	0: GCMFR register value not equal to count value 1: GCMFR register value is equal to count value	R/W
b4	CMEF	Count Match E	0: GCMER register value does not match count value 1: GCMER register value is equal to the count value	R/W
b3	CMDF	Count Match D	0: GCMDR register value does not match count value 1: GCMDR register value is equal to the count value	R/W
b2	CMCF	Count Match C	0: GCMCR register value not equal to count value 1: GCMCR register value is equal to the count value	R/W

TIM6_<t>_PWMA capture completion action

18.5.14 Hardware Startup Event Selection Register (TMR6_HSTAR)

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
STA RTS	Reserved														
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	HSTA 11	HSTA 10	HSTA 9	HSTA 8	HSTA 7	HSTA 6	HSTA 5	HSTA 4	-	-	HSTA 1	HSTA 0

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31	STARTS	hardware boot enable (computing)	0: Invalid hardware startup 1: Hardware startup is valid Note: SSTAR settings are not valid when hardware startup is in effect	R/W
b30~b12	Reserved	-	Reads "0", writes "0".	R/W
b11	HSTA11	Hardware startup conditions11	Condition: Sampling to falling edge on TIM6_TRIGB port 0: Hardware startup invalid when condition matches 1: Hardware startup is valid when conditions match	R/W
b10	HSTA10	Hardware startup conditions10	Condition: Sampling to rising edge on TIM6_TRIGB port 0: Hardware startup invalid when conditions match 1: Hardware startup is valid when conditions match	R/W
b9	HSTA9	Hardware startup conditions9	Condition: Sampling to falling edge on TIM6_TRIGA port 0: Hardware startup invalid when conditions match 1: Hardware startup is valid when conditions match	R/W
b8	HSTA8	Hardware startup conditions8	Condition: Sampling to rising edge on TIM6_TRIGA port 0: Hardware startup invalid when conditions match 1: Hardware startup is valid when conditions match	R/W
b7	HSTA7	Hardware startup conditions7	Condition: Sampling to falling edge on TIM6_<t>_PWMB port 0: Hardware startup invalid when conditions match 1: Hardware startup is valid when conditions match	R/W
b6	HSTA6	Hardware startup condition 6	Condition: Sampling to rising edge on TIM6_<t>_PWMB port 0: Hardware startup invalid when conditions match 1: Hardware startup is valid when conditions match	R/W
b5	HSTA5	Hardware startup condition 5	Condition: Sampling to falling edge on TIM6_<t>_PWMA port 0: Hardware startup invalid when conditions match 1: Hardware startup is valid when conditions match	R/W

			1: Hardware startup is valid when conditions match	
			Condition: TIM6_<t>_PWMA port upsampled to rising edge	
b4	HSTA4	Hardware startup condition 4	0: Hardware startup invalid when conditions match 1: Hardware startup is valid when conditions match	R/W
b3~b2	Reserved	-	Reads "0", writes "0".	R/W
b1	HSTA1	Hardware startup condition 1	Condition: Internal hardware trigger event 1 is valid 0: Hardware startup invalid when conditions match 1: Hardware startup is valid when conditions match	R/W
b0	HSTA0	Hardware startup condition 0	Condition: Internal hardware trigger event 0 is valid 0: Hardware startup invalid when conditions match 1: Hardware startup is valid when conditions match	R/W

18.5.15 Hardware Stop Event Selection Register (TMR6_HSTPR)

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
STOPs	Reserved														
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	HSTP 11	HSTP 10	HSTP 9	HSTP 8	HSTP 7	HSTP 6	HSTP 5	HSTP 4	-	-	HSTP 1	HSTP 0

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31	STOPs	hardware stop enable (computing)	0: Invalid hardware stop 1: Hardware stops being effective	R/W
			Note: SSTPR setting is invalid when hardware stop is active	
b30~b12	Reserved	-	Reads "0", writes "0".	R/W
b11	HSTP11	Hardware stop condition 11	Condition: Sampling to falling edge on TIM6_TRIGB port 0: Hardware stop is invalid when condition matches 1: Hardware stop is valid when conditions match	R/W
b10	HSTP10	Hardware stop condition 10	Condition: Sampling to rising edge on TIM6_TRIGB port 0: Hardware stop is invalid when condition matches 1: Hardware stop is valid when conditions match	R/W
b9	HSTP9	Hardware stop condition 9	Condition: Sampling to falling edge on TIM6_TRIGA port 0: Hardware stop is invalid when condition matches 1: Hardware stop is valid when conditions match	R/W
b8	HSTP8	Hardware stop condition 8	Condition: Sampling to rising edge on TIM6_TRIGA port 0: Hardware stop is invalid when condition matches 1: Hardware stop is valid when conditions match	R/W
b7	HSTP7	Hardware stop condition 7	Condition: Sampling to falling edge on TIM6_<t>_PWMB port 0: Hardware stop is invalid when condition matches 1: Hardware stop is valid when conditions match	R/W
b6	HSTP6	Hardware stop condition 6	Condition: Sampling to rising edge on TIM6_<t>_PWMB port 0: Invalid hardware stop when condition matches	R/W

			1: Hardware stop is valid when conditions match	
b5	HSTP5	Hardware stop condition 5	Condition: Sampling to falling edge on TIM6_<t>_PWMA port 0: Invalid hardware stop when condition matches	R/W
			1: Hardware stop is valid when conditions match	
b4	HSTP4	Hardware stop condition 4	Condition: TIM6_<t>_PWMA port upsampled to rising edge 0: Hardware stop is invalid when condition matches	R/W
			1: Hardware stop is valid when conditions match	
b3~b2	Reserved	-	Reads "0", writes "0".	R/W
b1	HSTP1	Hardware stop condition 1	Condition: Internal hardware trigger event 1 is valid 0: Hardware stop invalidated when condition matches 1: Hardware stop is valid when conditions match	R/W
b0	HSTP0	Hardware stop condition 0	Condition: Internal hardware trigger event 0 is valid 0: Hardware stop invalidated when condition matches 1: Hardware stop is valid when conditions match	R/W

18.5.16 Hardware clear event selection register (TMR6_HCLRR)

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
CLEARs	Reserved														
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	HCLE 11	HCLE 10	HCLE 9	HCLE 8	HCLE 7	HCLE 6	HCLE 5	HCLE 4	-	-	HCLE 1	HCLE 0

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31	CLEARs	Hardware zero enable	0: Invalid hardware clear 1: Hardware clear valid	R/W
b30~b12	Reserved	-	Reads "0", writes "0".	R/W
b11	HCLE11	Hardware clear condition 11	Note: SCLRR setting is invalid when hardware clear is active Condition: Sampling to falling edge on TIM6_TRIGB port 0: Invalid hardware clear when condition matches 1: Hardware clear is valid when conditions are matched	R/W
b10	HCLE10	Hardware clear condition 10	Condition: Sampling to rising edge on TIM6_TRIGB port 0: Invalid hardware clear when condition matches 1: Hardware clear is valid when conditions are matched	R/W
b9	HCLE9	Hardware clear condition 9	Condition: Sampling to falling edge on TIM6_TRIGA port 0: Invalid hardware clear when condition matches 1: Hardware clear is valid when conditions are matched	R/W
b8	HCLE8	Hardware clear condition 8	Condition: Sampling to rising edge on TIM6_TRIGA port 0: Invalid hardware clear when condition matches 1: Hardware clear is valid when conditions are matched	R/W
b7	HCLE7	Hardware clear condition 7	Condition: Sampling to falling edge on TIM6_<t>_PWMB port 0: Invalid hardware clear when condition matches 1: Hardware clear is valid when conditions are matched	R/W

			Condition: rising edge sampled on TIM6_<t>_PWMB port	
b6	HCLE6	Hardware clear condition 6	0: Invalid hardware clear when condition matches 1: Hardware clear is valid when conditions are matched	R/W
b5	HCLE5	Hardware clear condition 5	Condition: Sampling to falling edge on TIM6_<t>_PWMA port 0: Invalid hardware clear when condition matches 1: Hardware clear is valid when conditions are matched	R/W
b4	HCLE4	Hardware clear condition 4	Condition: TIM6_<t>_PWMA port upsampled to rising edge 0: Invalid hardware clear when condition matches 1: Hardware clear valid when conditions match	R/W
b3~b2	Reserved	-	Reads "0", writes "0".	R/W
b1	HCLE1	Hardware clear condition 1	Condition: Internal hardware trigger event 1 is valid 0: Invalid hardware clear when condition matches 1: Hardware clear valid when conditions match	R/W
b0	HCLE0	Hardware clear condition 0	Condition: Internal hardware trigger event 0 is valid 0: Invalid hardware clear when condition matches 1: Hardware clear is valid when conditions are matched	R/W

18.5.17 Hardware Capture Event Selection Register (TMR6_HCPAR)

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
-	-	-	-	HCP A11	HCP A10	HCP A9	HCP A8	HCP A7	HCP A6	HCP A5	HCP A4	-	-	HCP A1	HCP A0

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b12	Reserved	-	Reads "0", writes "0".	R/W
b11	HCRA11	Hardware Capture A Condition 11	Condition: Sampling to falling edge on TIM6_TRIGB port 0: Hardware capture A is invalid when condition matches 1: Hardware capture A is valid when conditions match	R/W
b10	HCRA10	Hardware Capture A Condition 10	Condition: Sampling to rising edge on TIM6_TRIGB port 0: Hardware capture A is invalid when condition matches 1: Hardware Capture A is valid when conditions match	R/W
b9	HCRA9	Hardware Capture A Condition 9	Condition: Sampling to falling edge on TIM6_TRIGA port 0: Hardware capture A is invalid when the condition matches 1: Hardware Capture A is valid when conditions match	R/W
b8	HCRA8	Hardware Capture A Condition 8	Condition: Sampling to rising edge on TIM6_TRIGA port 0: Hardware capture A is invalid when the condition matches 1: Hardware Capture A is valid when conditions match	R/W
b7	HCRA7	Hardware Capture A Condition 7	Condition: Sampling to falling edge on TIM6_<t>_PWMB port 0: Hardware capture A is invalid when condition matches 1: Hardware capture A is valid when conditions match	R/W
			Condition: rising edge sampled on TIM6_<t>_PWMB port	

b6	HCPA6	Hardware Capture A Condition 6	0: Hardware capture A is invalid when the condition matches 1: Hardware capture A is valid when conditions match	R/W
b5	HCPA5	Hardware Capture A Condition 5	Condition: Sampling to falling edge on TIM6_<t>_PWMA port 0: Hardware capture A is invalid when the condition matches 1: Hardware capture A is valid when conditions match	R/W
b4	HCPA4	Hardware Capture A Condition 4	Condition: TIM6_<t>_PWMA port upsampled to rising edge 0: Hardware capture A is invalid when the condition matches 1: Hardware capture A is valid when conditions match	R/W
b3~b2	Reserved	-	Reads "0", writes "0".	R/W
b1	HCPA1	Hardware Capture A Condition 1	Condition: Internal hardware trigger event 1 is valid 0: Hardware capture A is invalid when condition matches 1: Hardware capture A is valid when conditions match	R/W
b0	HCPA0	Hardware Capture A Condition 0	Condition: Internal hardware trigger event 0 is valid 0: Hardware capture A is invalid when condition matches 1: Hardware capture A is valid when conditions match	R/W

18.5.18 Hardware Capture Event Select Register (TMR6_HCPBR)

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
-	-	-	-	HCP B11	HCP B10	HCP B9	HCP B8	HCP B7	HCP B6	HCP B5	HCP B4	-	-	HCP B1	HCP B0

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b12	Reserved	-	Reads "0", writes "0".	R/W
b11	HCPB11	Hardware Capture B Condition 11	Condition: Sampling to falling edge on TIM6_TRIGB port 0: Hardware capture B is invalid when condition matches 1: Hardware capture B is valid when conditions match	R/W
b10	HCPB10	Hardware Capture B Condition 10	Condition: Sampling to rising edge on TIM6_TRIGB port 0: Hardware capture B is invalid when condition matches 1: Hardware capture B is valid when conditions match	R/W
b9	HCPB9	Hardware Capture B Condition 9	Condition: Sampling to falling edge on TIM6_TRIGA port 0: Hardware capture B is invalid when condition matches 1: Hardware capture B is valid when conditions match	R/W
b8	HCPB8	Hardware Capture B Condition 8	Condition: Sampling to rising edge on TIM6_TRIGA port 0: Hardware capture B is invalid when condition matches 1: Hardware capture B is valid when conditions match	R/W
b7	HCPB7	Hardware Capture B Condition 7	Condition: Sampling to falling edge on TIM6_<t>_PWMB port 0: Hardware capture B is invalid when condition matches 1: Hardware capture B is valid when conditions match	R/W
			Condition: rising edge sampled on TIM6_<t>_PWMB port	

b6	HCPB6	Hardware Capture B Condition 6	0: Hardware capture B is invalid when condition matches 1: Hardware capture B is valid when conditions match	R/W
b5	HCPB5	Hardware Capture B Condition 5	Condition: Sampling to falling edge on TIM6_<t>_PWMA port 0: Hardware capture B is invalid when condition matches 1: Hardware capture B is valid when conditions match	R/W
b4	HCPB4	Hardware Capture B Condition 4	Condition: TIM6_<t>_PWMA port upsampled to rising edge 0: Hardware capture B is invalid when condition matches 1: Hardware capture B is valid when conditions match	R/W
b3~b2	Reserved	-	Reads "0", writes "0".	R/W
b1	HCPB1	Hardware Capture B Condition 1	Condition: Internal hardware trigger event 1 is valid 0: Hardware capture B is invalid when condition matches 1: Hardware capture B is valid when conditions match	R/W
b0	HCPB0	Hardware Capture B Condition 0	Condition: Internal hardware trigger event 0 is valid 0: Hardware capture B is invalid when condition matches 1: Hardware capture B is valid when conditions match	R/W

18.5.19 Hardware-recursive event selection register (TMR6_HCUPR)

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17			
Reserved															HC UP17	HC UP16	
b16	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		
classifier for honorific people	marking	celebrity														fill out or in (information on a form)	
b31~b18	Reserved	-														R/W	
																Condition: Internal hardware trigger event 1 is valid	
b17	HCUP17	Hardware incremental condition 17														0: Hardware increment is invalid when condition matches	R/W
																1: Hardware recursion is valid when conditions are matched	
																Condition: Internal hardware trigger event 0 is valid	
b16	HCUP16	Hardware incremental condition 16														0: Hardware increment is invalid when condition matches	R/W
																1: Hardware recursion is valid when conditions are matched	
b15~b12	Reserved	-														Reads "0", writes "0".	R/W
																Condition: Sampling to falling edge on TIM6_TRIGB port	
b11	HCUP11	Hardware incremental condition 11														0: Hardware increment is invalid when condition matches	R/W
																1: Hardware recursion is valid when conditions are matched	
																Condition: Sampling to rising edge on TIM6_TRIGB port	
b10	HCUP10	Hardware incremental condition 10														0: Hardware increment is invalid when condition matches	R/W
																1: Hardware recursion is valid when conditions are matched	
																Condition: Sampling to falling edge on TIM6_TRIGA port	
b9	HCUP9	Hardware incremental condition 9														0: Hardware increment is invalid when condition matches	R/W
																1: Hardware recursion is valid when conditions are matched	
																Condition: Sampling to rising edge on TIM6_TRIGA port	
b8	HCUP8	Hardware														0: Hardware increment is invalid when condition	R/W

		incremental condition 8	matches 1: Hardware recursion is valid when conditions are matched	
b7	HCUP7	Hardware incremental condition 7	Condition: when the TIM6_<t>_PWMB port is high TIM6_<t>_PWMA port upsample to falling edge 0: Hardware increment is invalid when condition matches 1: Hardware recursion is valid when conditions are matched	R/W
b6	HCUP6 Incremental Condition 6	Hardware	Condition: TIM6_<t>_PWMB port is high, TIM6_<t>_PWMA port upsampled to rising edge 0: Hardware increment is invalid when condition matches 1: Hardware recursion is valid when conditions are matched	R/W
b5	HCUP5 Incremental Condition 5	Hardware	Condition: TIM6_<t>_PWMB port is low, TIM6_<t>_PWMA port up-sampled to falling edge 0: Hardware increment is invalid when condition matches 1: Hardware recursion is valid when conditions are matched	R/W
b4	HCUP4 Handoff Condition 4	Hardware	Condition: TIM6_<t>_PWMB port is low, TIM6_<t>_PWMA port upsampled to rising edge 0: Hardware increment is invalid when condition matches 1: Hardware recursion is valid when conditions are matched	R/W

			Condition: TIM6_<t>_PWMA port is high, TIM6_<t>_PWMB port upsampled to falling edge 0: Hardware increment is invalid when condition matches 1: Hardware recursion is valid when conditions are matched	
b3	HCUP3	Hardware	TIM6_<t>_PWMB port upsampled to falling edge 0: Hardware increment is invalid when condition matches 1: Hardware recursion is valid when conditions are matched	R/W
			Condition: TIM6_<t>_PWMA port is high, TIM6_<t>_PWMB port upsampled to rising edge 0: Hardware increment is invalid when condition matches 1: Hardware recursion is valid when conditions are matched	R/W
b2	HCUP2	Hardware	Condition: TIM6_<t>_PWMA port is high, TIM6_<t>_PWMB port upsampled to rising edge 0: Hardware increment is invalid when condition matches 1: Hardware recursion is valid when conditions are matched	R/W
			Condition: TIM6_<t>_PWMA port is low, TIM6_<t>_PWMB port up-sampled to falling edge 0: Hardware increment is invalid when condition matches 1: Hardware recursion is valid when conditions are matched	R/W
b1	HCUP1	Hardware	Condition: TIM6_<t>_PWMA port is low, TIM6_<t>_PWMB port up-sampled to falling edge 0: Hardware increment is invalid when condition matches 1: Hardware recursion is valid when conditions are matched	R/W
			Condition: rising edge sampled on TIM6_<t>_PWMB port when TIM6_<t>_PWMA port is low 0: hardware increment is invalid when condition matches 1: Hardware recursion is valid when conditions are matched	
b0	HCUP0	Hardware	Condition: rising edge sampled on TIM6_<t>_PWMB port when TIM6_<t>_PWMA port is low 0: hardware increment is invalid when condition matches 1: Hardware recursion is valid when conditions are matched	

18.5.20 Hardware Decrement Event Selection Register (TMR6_HCDOR)

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17		
Reserved															HC DO17	HC DO16
b16	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
classifier for honorific people	marking	celebrity													fill out or in (information on a form)	
b31~b18	Reserved	-														R/W
															Condition: Internal hardware trigger event 1 is valid	
b17	HCDO17	Hardware decreasing condition 17													0: hardware decrement is invalid when condition matches	R/W
															1: Hardware decrement is valid when conditions match	
															Condition: Internal hardware trigger event 0 is valid	
b16	HCDO16	Hardware decreasing condition 16													0: hardware decrement is invalid when condition matches	R/W
															1: Hardware decrement is valid when conditions match	
b15~b12	Reserved	-													Reads "0", writes "0".	R/W
															Condition: Sampling to falling edge on TIM6_TRIGB port	
b11	HCDO11	Hardware decreasing condition 11													0: hardware decrement is invalid when condition matches	R/W
															1: Hardware decrement is valid when conditions match	
															Condition: Sampling to rising edge on TIM6_TRIGB port	
b10	HCDO10	Hardware Decrement Condition 10													0: hardware decrement is invalid when condition matches	R/W
															1: Hardware decrement is valid when conditions match	
															Condition: Sampling to falling edge on TIM6_TRIGA port	
b9	HCDO9	Hardware Decreasing Condition 9													0: hardware decrement is invalid when condition matches	R/W
															1: Hardware decrement is valid when conditions match	
															Condition: Sampling to rising edge on TIM6_TRIGA port	
b8	HCDO8	Hardware													0: hardware decrement is invalid when condition	R/W

		Decrement Condition 8	matches	
			1: Hardware decrement is valid when conditions are matched	
			Condition: when the TIM6_<t>_PWMB port is high	
b7	HCDO7	Hardware decreasing condition 7	TIM6_<t>_PWMA port upsample to falling edge 0: hardware decrement is invalid when condition matches	R/W
			1: Hardware decrement is valid when conditions are matched	
b6	HCDO6	Hardware decreasing condition6	Condition: TIM6_<t>_PWMB port is high, TIM6_<t>_PWMA port upsampled to rising edge 0: Hardware decrementing invalid when conditions match	R/W
b5	HCDO5	Hardware Decreasing Condition 5	1: Hardware decrement is valid when conditions match	R/W
b4	HCDO4	Hardware decreasing condition 4	Condition: TIM6_<t>_PWMB port is low, TIM6_<t>_PWMA port upsampled to rising edge 0: Hardware decrement invalid when condition matches	R/W
			1: Hardware decrement is valid when conditions are matched	

			Condition: TIM6_<t>_PWMA port is high, TIM6_<t>_PWMB port upsampled to falling edge 0: Hardware decrement invalid when condition matches 1: Hardware decrement is valid when conditions are matched	R/W
b3	HCDO3	Hardware	decreasing condition 3	
b2	HCDO2	Hardware	decreasing condition 2	Condition: TIM6_<t>_PWMA port is high when TIM6_<t>_PWMB port is upsampled to rising edge 0: Hardware decrement is invalid when condition matches 1: Hardware decrement is valid when
b1	HCDO1	Hardware	decreasing condition 1	conditions are matched Condition: TIM6_<t>_PWMA port is low, TIM6_<t>_PWMB port up-sampled to falling edge 0: Hardware decrement invalid when condition matches 1: Hardware decrement is valid when
b0	HCDO0	Hardware	decreasing condition 0	conditions are matched Condition: TIM6_<t>_PWMA port is low, TIM6_<t>_PWMB port upsampled to rising edge 0: Hardware decrement invalid when condition matches 1: Hardware decrement is valid when conditions are matched

18.5.21 Software Synchronized Startup Control Register (TMR6_SSTAR)

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												SSTA3	SSTA2	SSTA1	

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b3	Reserved	-	Reads "0", writes "0".	R/W
b2	SSTA3	Unit 3 Software Startup	0: Software startup disabled R/W 1: Software startup enabled	
b1	SSTA2	Unit 2 Software Startup	0: Software startup disabled R/W 1: Software startup enabled	
b0	SSTA1	Module 1 Software Startup	0: Software startup disabled R/W 1: Software startup enabled	

18.5.22 Software Synchronization Stop Control Register (TMR6_SSTPR)

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved												SSTP3	SSTP2	SSTP1	

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b3	Reserved	-	Reads "0", writes "0".	R/W
b2	SSTP3	Unit 3 Software Stop	0: Software stop disabled R/W 1: Software stop enable	
b1	SSTP2	Unit 2 Software Stop	0: Software stop disabled R/W 1: Software stop enable	
b0	SSTP1	Module 1 Software Stop	0: Software stop disabled R/W 1: Software stop enable	

18.5.23 Software Synchronized Clear Control Register (TMR6_SCLRR)

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved												SCLE3	SCLE2	SCLE1	

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b3	Reserved	-	Reads "0", writes "0".	R/W
b2	SCLE3	Unit 3 Software Zeroing	0: Software clear invalid R/W 1: Software clear enable	
b1	SCLE2	Unit 2 Software Zeroing	0: Software clear invalid R/W 1: Software clear enable	
b0	SCLE1	Unit 1 Software Zeroing	0: Software clear invalid R/W 1: Software clear enable	

19 Universal Control Timer (Timer4)

19.1 summary

The Universal Control Timer 4 (Timer4) is a timer module for three-phase motor control, providing a variety of three-phase motor control solutions for different applications. The timer supports triangular and sawtooth waveform modes, and can generate various PWM waveforms; supports cache function; and supports EMB control. This series of products is equipped with 3 units of Timer4.

19.2 basic block diagram

The basic functions and features of Timer4 are shown in Table 19-1.

Table 19-1 Basic Functions and Characteristics of Timer4

Waveform Mode	Sawtooth wave, triangle wave
Basic Functions	- Incremental and decremental counting direction
	- Cache Functions
	- General Purpose PWM Output
	- Dedicated event output
	- EMB control
Interrupt Type	Count Compare Match Interrupt
	Count Cycle Matching Interrupt
	Overloaded Count Match Interrupt

The basic architecture of the general-purpose control timer Timer4 is depicted in Figure 19-1. The "<t>" shown in the block diagram indicates the cell number, i.e., "<t>" is 1~3, and all references to "<t>" later in this chapter refer to the cell number. later in this chapter refers to the unit number and will not be repeated.

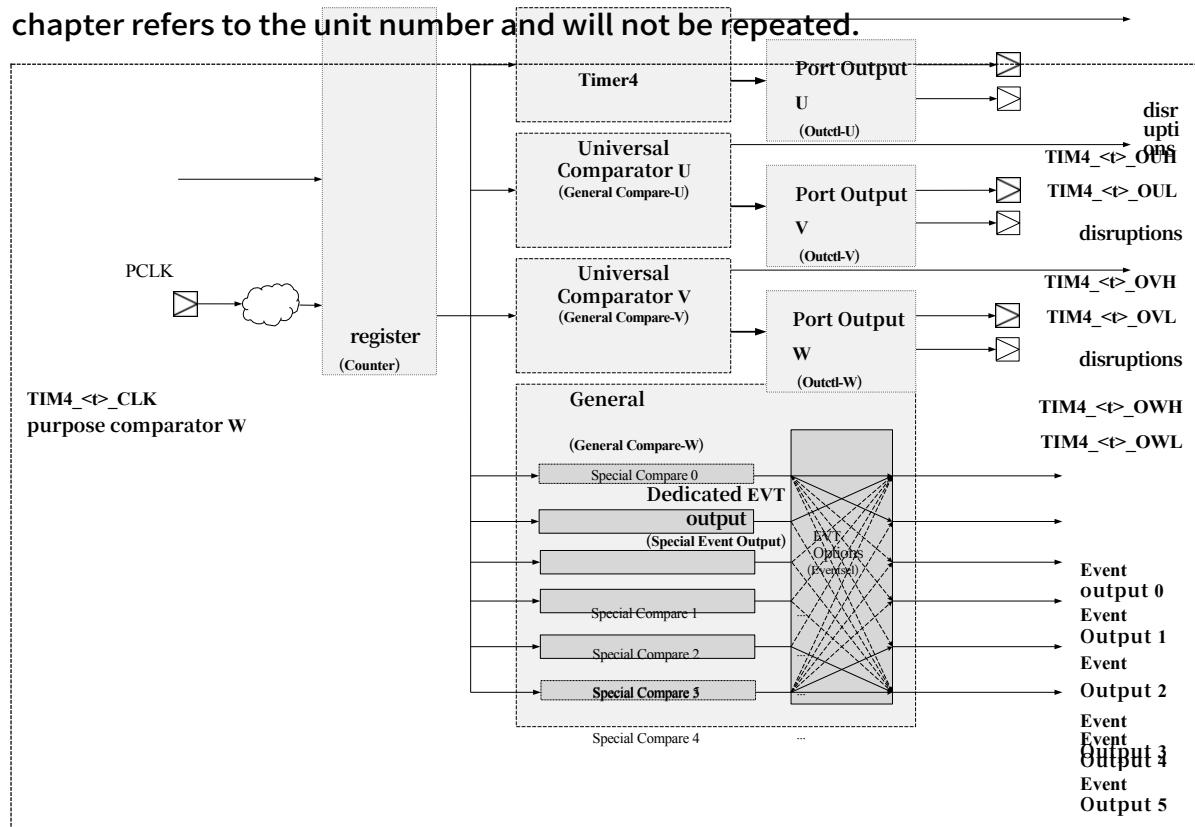


Figure 19-1 Timer4 Basic Block Diagram

Table 19-2 shows the list of input and output ports of Timer4.

Table 19-2 Timer4 Port List

port name	orientations	functionality
TIM4_<t>_CLK	in	Counting Clock Input Port
TIM4_<t>_OUH		
TIM4_<t>_OUL		
TIM4_<t>_OVH		
TIM4_<t>_OVL		PWM output port
TIM4_<t>_OWH		
TIM4_<t>_OWL		

19.3 Functional Description

19.3.1 basic movement

19.3.1.1 Waveform Mode

Timer4 has two basic counting waveform modes, sawtooth waveform mode and triangle waveform mode. The two modes can be realized according to the setting of CCSR.MODE.

Figure 19-2 and Figure 19-3 below show the waveforms of sawtooth waveform and triangle waveform respectively.

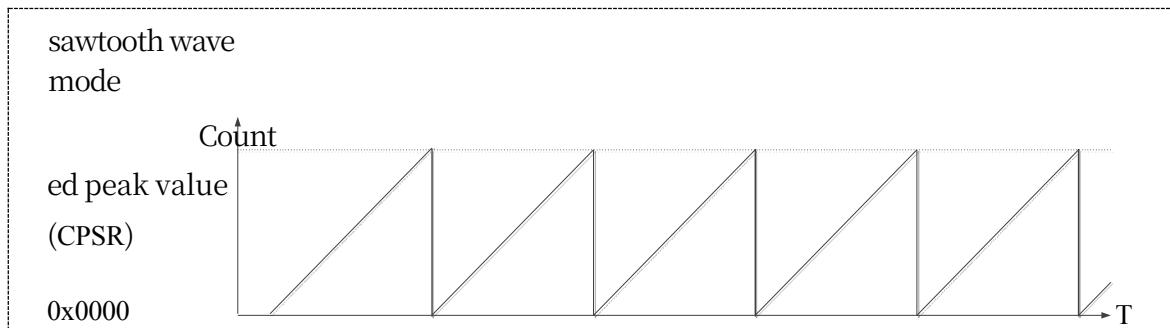


Figure 19-2 Timer4 Sawtooth Waveforms

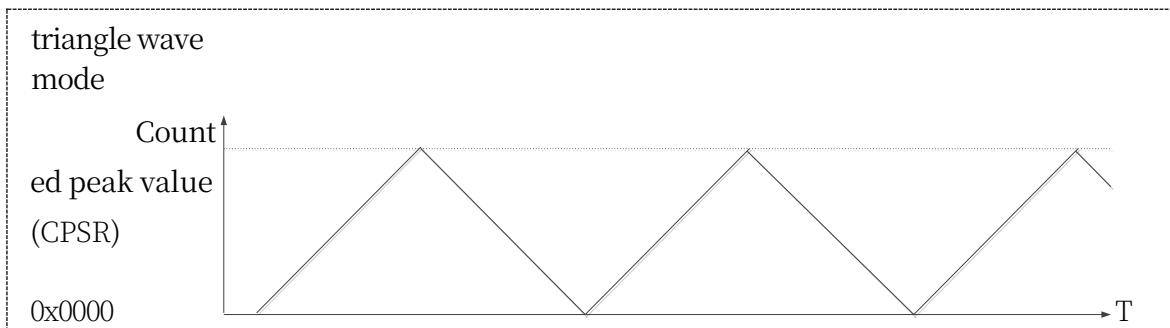


Figure 19-3 Timer4 Triangle Waveforms

19.3.1.2 countin g action

- 1) The sawtooth wave counting operation and control flow is shown in Figure 19-4.

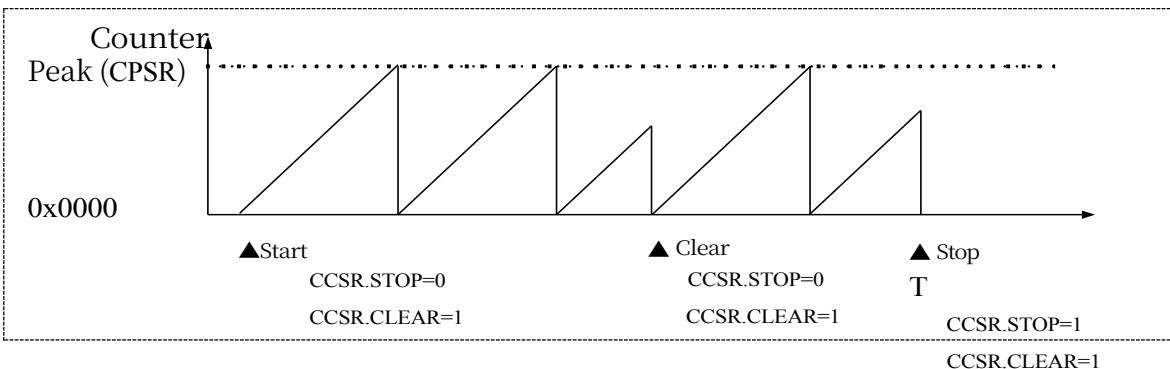


Figure 19-4 Timer4 Sawtooth Wave Mode Counting Action

- a) Set mode CCSR.MODE=0.
 - b) Sets the count peak CPSR register.
 - c) By writing CCSR.STOP=0 and CCSR.CLEAR=1, the counter count value (CNTR) is initialized to 0x0000 and the counting operation is started. The counter value is incremented from 0x0000 and when the peak value (CPSR) is reached, the counter value returns to 0x0000 and the operation is repeated.
 - d) Count period = (CPSR+1) x count clock period
 - e) During counting, writing CCSR.STOP=0 and CCSR.CLEAR=1 initializes the count value to 0x0000 and continues the counting operation.
 - f) During counting, writing CCSR.STOP=1 and CCSR.CLEAR=1 initializes the count value to 0x0000 and stops the counting operation.
- 2) The triangular wave counting operation and control flow is shown in Figure 19-5.

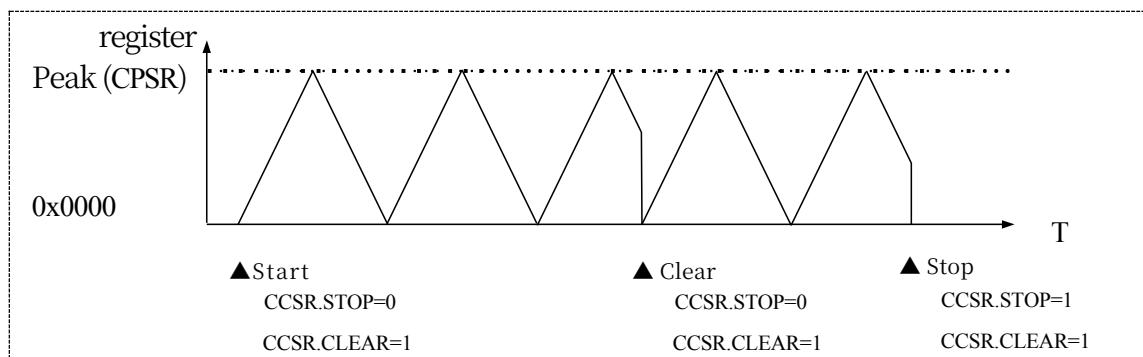


Figure 19-5 Timer4 Triangular Wave Mode Counting Action

- a) Set the mode CCSR.MODE=1.
- b) Sets the count peak CPSR register.
- c) By writing CCSR.STOP=0 and CCSR.CLEAR=1, the counter count value (CNTR) is initialized to 0x0000 and counting operation is started. The counter value starts from 0x0000 and counts incrementally until it reaches the peak value; when it reaches the peak value (CPSR), the counter starts to count decrementally until the count value returns to 0x0000; after that, it re-increments again and repeats this operation.
- d) Count period = (CPSR) × 2 × count clock period
- e) During counting, write CCSR.STOP=0 and CCSR.CLEAR=1 to initialize the count value to 0x0000 and re-increment the counting operation, then repeat the above operation.
- f) During counting, writing CCSR.STOP=1 and CCSR.CLEAR=1 initializes the count value to 0x0000 and stops the counting operation.

19.3.1.3 Comparison Output

- Figure 19-6 below shows an example of the waveform output of the compare output module in the sawtooth wave counting mode.

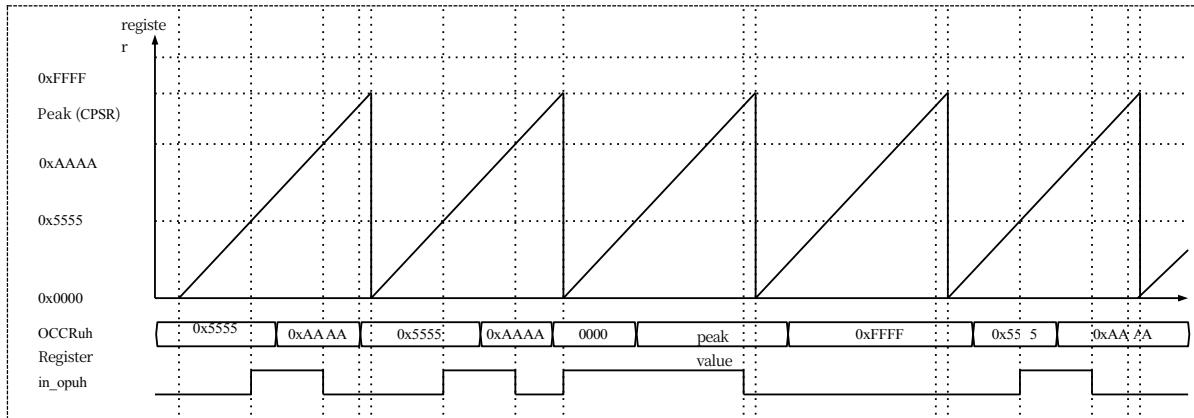


Figure 19-6 Sawtooth Mode Waveform Output Example

- Figure 19-7 below shows an example of the waveform output of the Compare Output Module in the triangle wave counting mode.

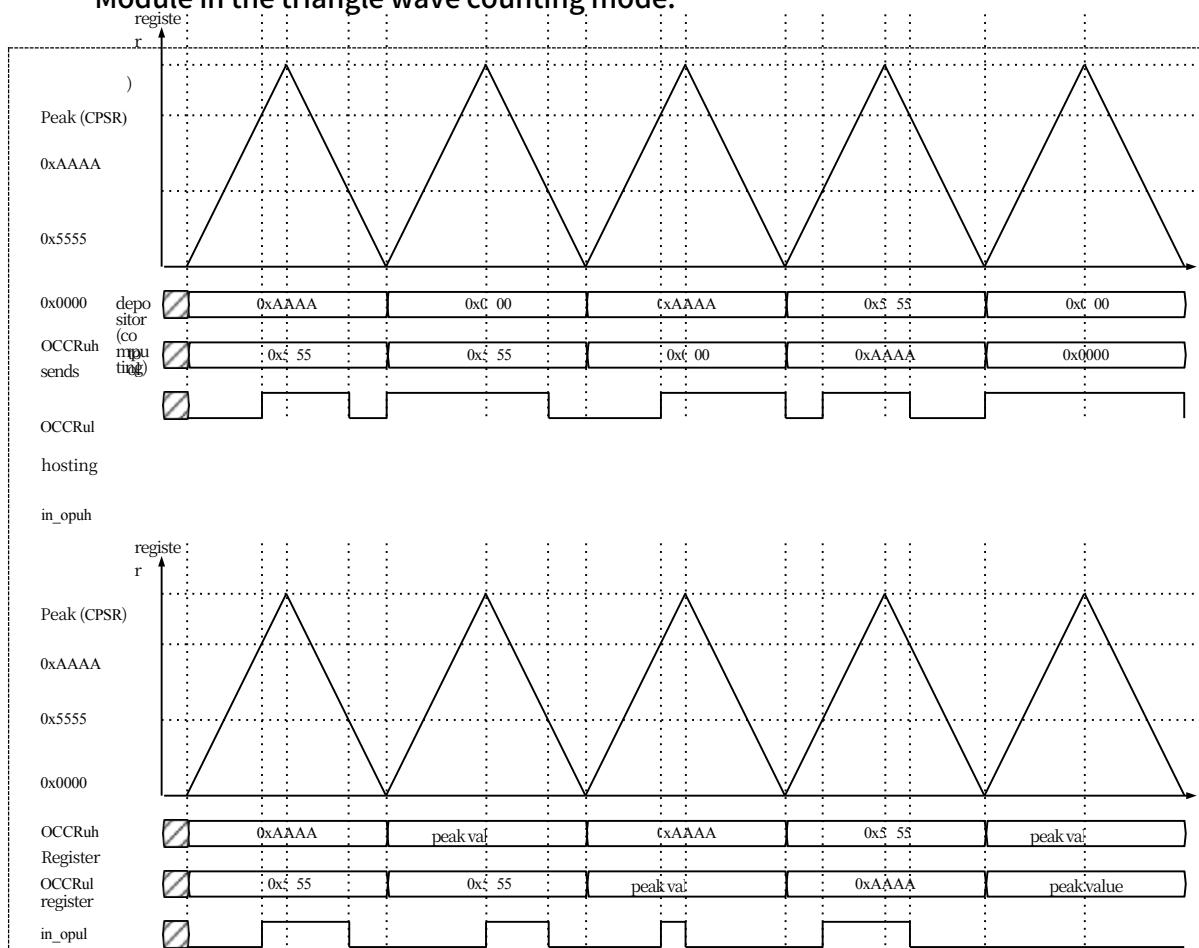


Figure 19-7 Triangle Mode Waveform Output Example

19.3.2 caching function

Timer4's cycle reference register (CPSR) general-purpose comparison reference register (OCCR) general-purpose mode control register (OCMR), special-comparison reference register (SCCR), and special-comparison mode control register (SCMR) all have caching functions.

19.3.2.1 Cycle Base Register Cache Function

The CPSR has a buffer function register, and the written count peak data is first stowed in the buffer register. Data is transferred from the buffer register to the CPSR register under the following conditions.

- When the buffer function is disabled (CCSR.BUFEN=0) write data is immediately transferred from the buffer register to the CPSR register.
- When the buffer function is enabled (CCSR.BUFEN=1) data is transferred from the buffer register to the CPSR register when the counter is stopped (CCSR.STOP=1) or when the counter count value is "0x0000".

Attention:

-When data is read from the CPSR, the value read is not the CPSR buffer register value, but the CPSR register value. When the buffer function is enabled, the value read before the transfer is completed is not the most recently written value, but the CPSR value of the last write.

Figure 19-8 shows the operation of modifying the count peak CPSR when in sawtooth wave mode and the buffer function is disabled.

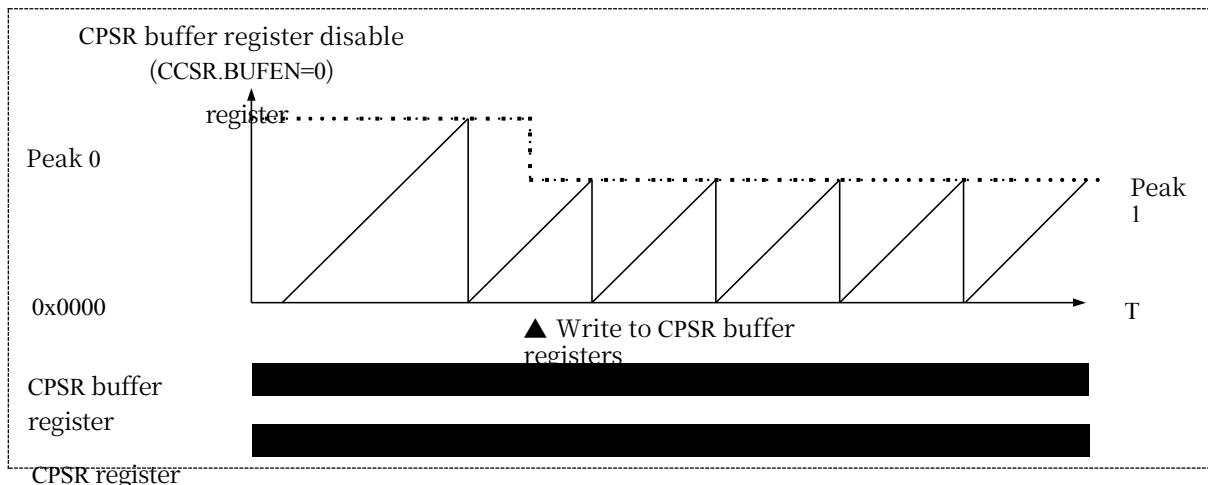


Figure 19-8 Modifying the Ramp Count Period When the Cache is Invalidated

Attention:

- When the buffer function is disabled, the write data is immediately transferred from the buffer register to the CPSR counter, and the counting period is changed as soon as the write operation is completed. In this case, if the value written is less than the

Figure 19-9 shows the operation of modifying the count peak CPSR when the sawtooth wave mode and the buffer function are enabled.

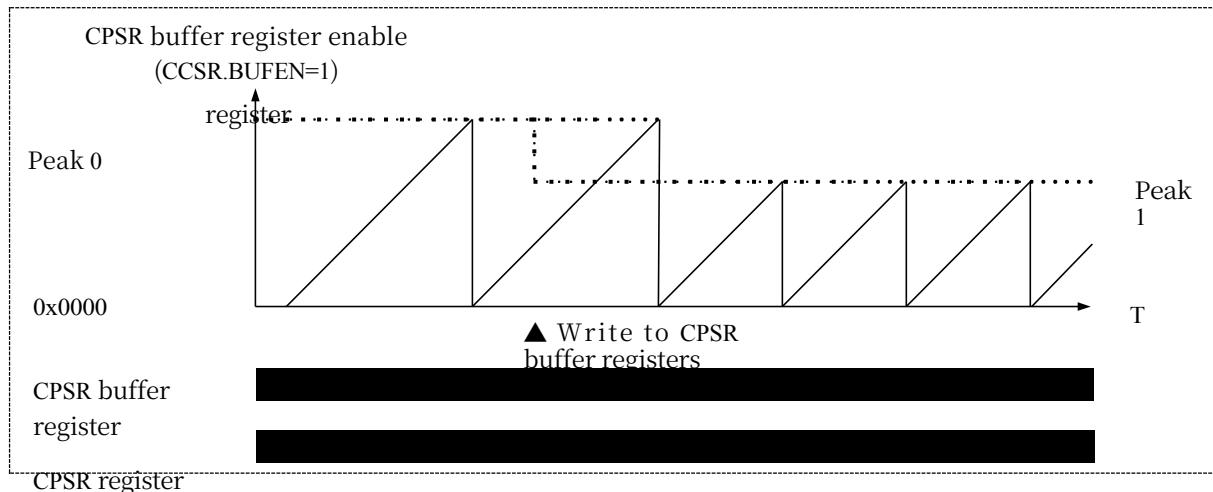


Figure 19-9 Modifying the Ramp Count Period with Cache Enable

As shown in the figure, when the buffer function is enabled, the written data is transferred from the buffer register to the CPSR register when the counter is stopped or when the counter count value is "0x0000".

Figure 19-10 shows the operation of modifying the count peak CPSR when in triangle wave mode and the buffer function is enabled.

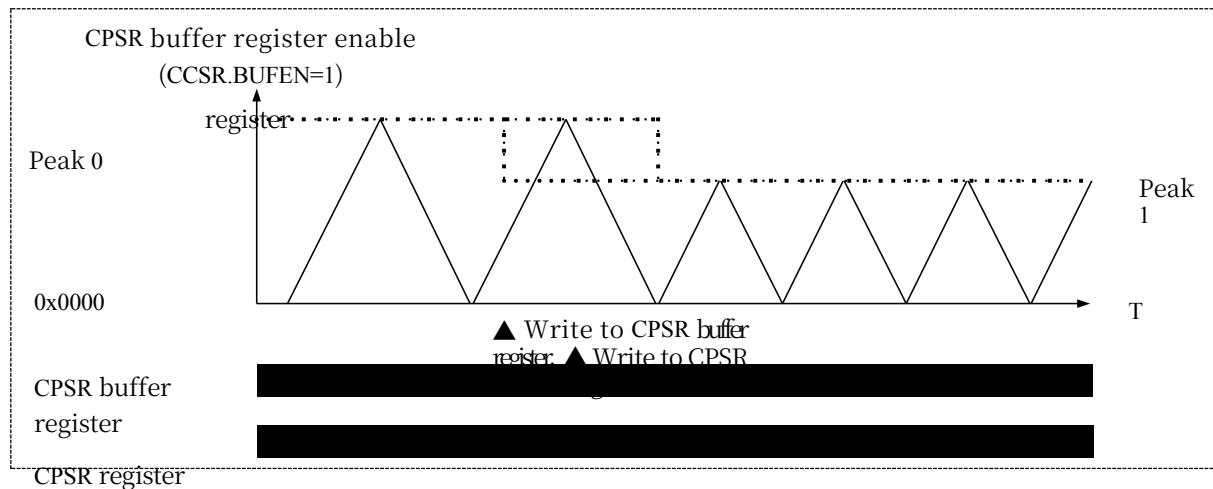


Figure 19-10 Modifying the Triangle Count Period with Cache Enable

As shown in the figure, in the triangle wave counting 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 when the next count value of the counter is "0x0000". The counter cycle change starts at the next counter cycle after the write operation is completed.

19.3.2.2 General purpose comparison register cache function

Both the general-purpose comparison reference value register (OCCR) and the general-purpose mode control register (OCMR) have a buffer register function that transmits loads to the OCCR and OCMR registers at specified transmission moments when the buffer function is

enabled. The OCCR buffer function can be used to synchronize the change of the comparison value during counting, and the OCMR buffer function can be used to count the up-overflow point (sawtooth wave) during counting. The OCMR buffer function can be used to synchronously change the compare value during counting, and the OCMR buffer function can be used to synchronously change the internal PWM output during counting of the overflow point (sawtooth wave), counting of the valley point, or counting of the peak point (triangular wave).

- a) When the link transmission of the output compare and counter cycle interval response function is disabled, the buffer value is loaded into the register at the set count state. The loading condition at this time is independent of the counter cycle interval counter.

Figure 19-11 shows the waveform ($x=L$ or H) when the general-purpose output comparison OCCR buffer function is enabled, the counter is loaded at zero (OCER.CxBUFEN=01), the counter cycle interval response link is disabled (OCER.LMCx=0).

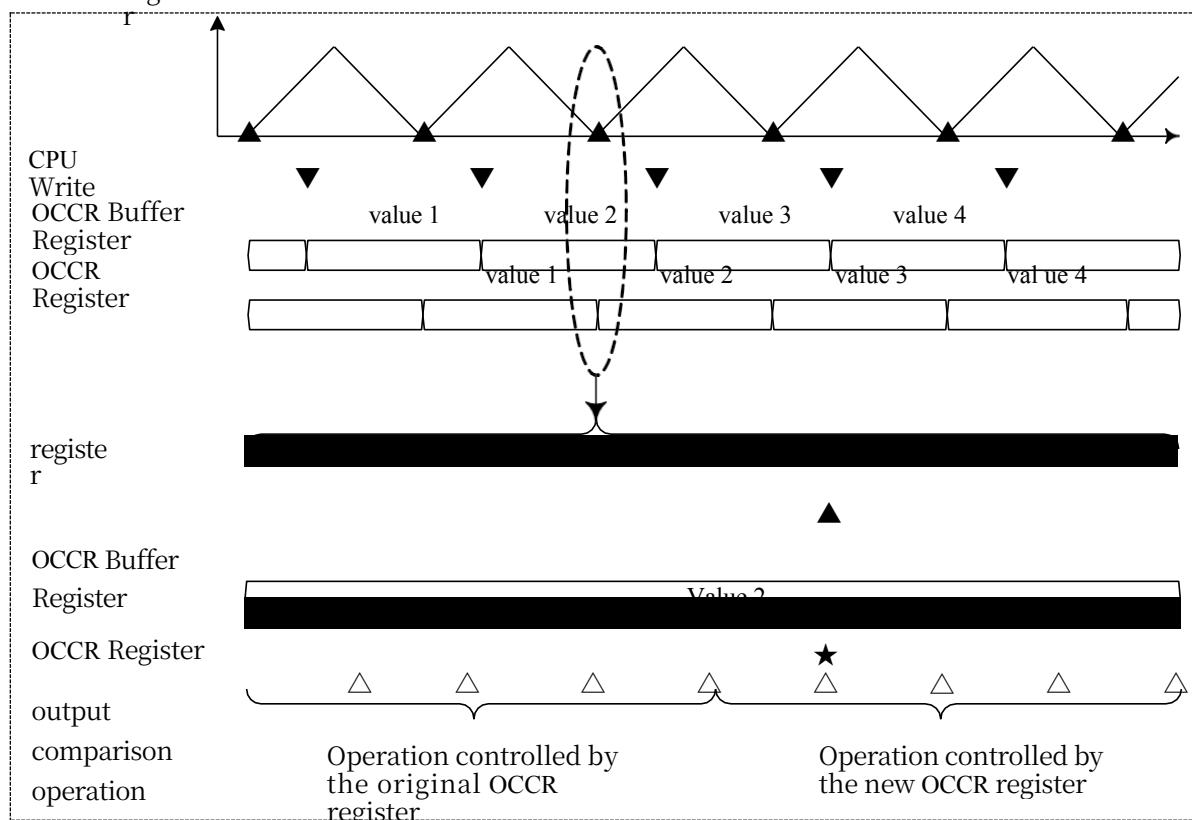


Figure 19-11 OCCR Buffered Data Transmission (Cycle Interval Response Link Disable)

The top half of the figure is a global schematic and the bottom half is an enlarged view during transmission operation.

The counter is in delta wave counting mode, and the underflow interrupt is generated at the moment of flag ▲ (count valley). At the ▼ moment, the CPU overwrites the OCCR register and the written data is stored in the OCCR buffer register. Later, when the counter count value is 0x0000, a data load operation is performed and the data goes from the buffer register to the OCCR register, and the interrupt flag IRQZF is generated.

At moment △, the output comparison performs the event of changing the PWM output and setting the OCSR.OCFx bit ($x=L$ or H) according to the set OCCR register value matching the count value. After the moment ★ (count value = 0x0000), the port output performs the operation according to the new OCCR data. Before the moment ★, the port output performs the operation according to the original OCCR data.

The figure illustrates the transfer operation of the OCCR buffer register at the count valley point.

similar to the cache transfer operation of the OCMR's buffer register; similarly, the transfer operation at the count peak point is similar. The new data takes effect immediately after the moment of transfer (the newly written data will control the PWM output and the interrupt flag is set)

- b) When the counter cycle interval response link is enabled, a buffer register transfer operation is performed when the buffer value is in the set count state and the cycle interval counter count value is zero.**

Figure 19-12 shows the waveforms (x=L or H) when the general-purpose output comparison OCCR buffer function is enabled, the counter is loaded at zero (OCER.CxBUFEN=01), counter cycle interval response link is enabled (OCER.LMCx=1).

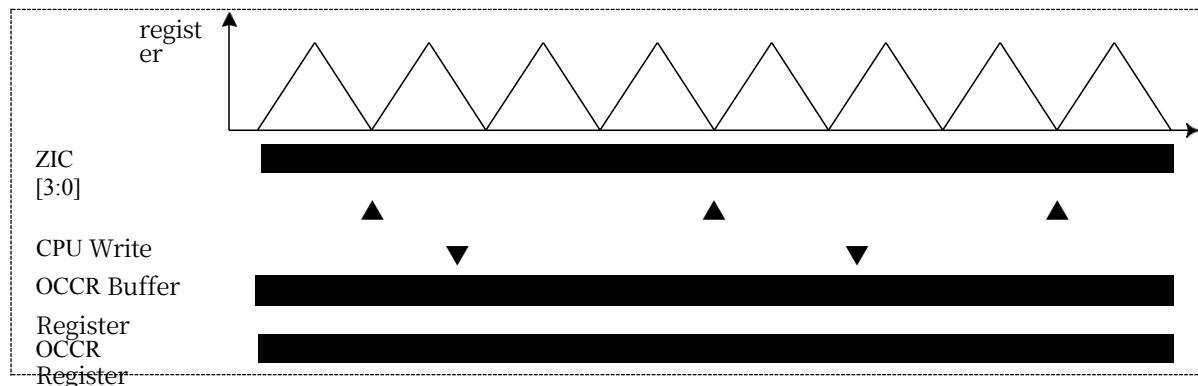


Figure 19-12 OCCR Buffered Data Transfer (Cycle Interval Response Link Enable)

The counter is in delta wave counting mode, the cycle interval counter (CVPR.ZIC) counts decrementally from 2 to 0, and the underflow interrupt is at flag ▲.
(count valley)moment is generated. At the ▼ moment, the CPU overwrites the OCCR register and the written data is stored in the OCCR buffer register. Afterwards, when the counter count value is 0x0000 and the cycle interval counter (CVPR.ZIC) is 0, a data load operation is performed, and the data is transferred from the buffer register to the OCCR register, and the interrupt flag IRQZF is generated.

The figure illustrates the transfer operation of the OCCR buffer register at the count valley point, similar to the cache transfer operation of the OCMR buffer register; similarly, the transfer operation at the count peak point is similar. The new data takes effect immediately after the transfer moment (the newly written data will control the PWM output and the interrupt flag is set).

When using the channel link operation mode, enabling the OCCRh and OCCRul buffer functions at the same time can generate various PWM output waveforms. Figure 19-13 illustrates the TIM4 <t>-QUL mode by changing the OC MR register value to generate different output waveforms while keeping the output comparison registers OCCRh and OCCRul unchanged.

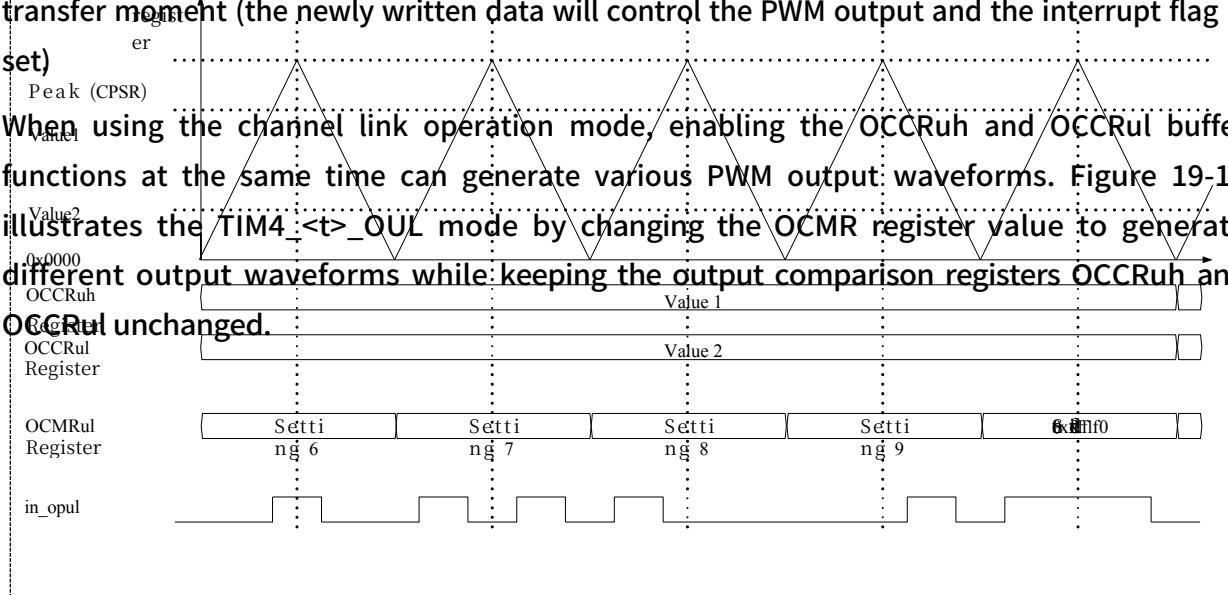


Figure 19-13 Output Compare Buffer Data Transfer (OCMR Buffer Enable)

Attention:

- Setting 6: EOPNUCL=11, EOPNDCL=11, OPUCL=00, OPDCL=00, and the rest of EOP* and OP* are sent
The memory is 00
- Setting 7: EOPNUCL=11, EOPNDCL=11, OPUCL=11, OPDCL=11, the rest of EOP* and OP* sends
The memory is 00
- Setting 8: EOPNUCL=11, EOPNDCL=00, OPUCL=11, OPDCL=00, and the rest of EOP* and OP* sent
The memory is 00
- Setting 9: EOPNUCL=00, EOPNDCL=11, OPUCL=00, OPDCL=11, and the rest of EOP* and OP* are sent to
The memory is 00
- Setting 10: EOPNUCL=00, EOPNDCL=00, OPUCL=11, OPDCL=11, and the rest of EOP* and OP*.
The register is 00

19.3.2.3 Dedicated comparison register cache function

The Specialized Comparison Reference Register (SCCR) and in the Specialized Mode Control Register (SCMR) both have buffer function registers. When the buffer function is enabled, the values written by the CPU to the SCCR and SCMR buffer registers are loaded into the SCCR and SCMR registers at the set counter state.

- a) **When the counter cycle interval response link transfer is disabled, the buffered transfer operation is related only to the counter state and is not affected by the counter cycle interval counter.**

Shown in Figure 19-14 is the Enable Register Buffer function that disables the counter cycle interval response link transfer (SCSR.LMC=0) and the transfer to the SCCR and SCMR registers at counter zero (SCSR.BUFEN=01).

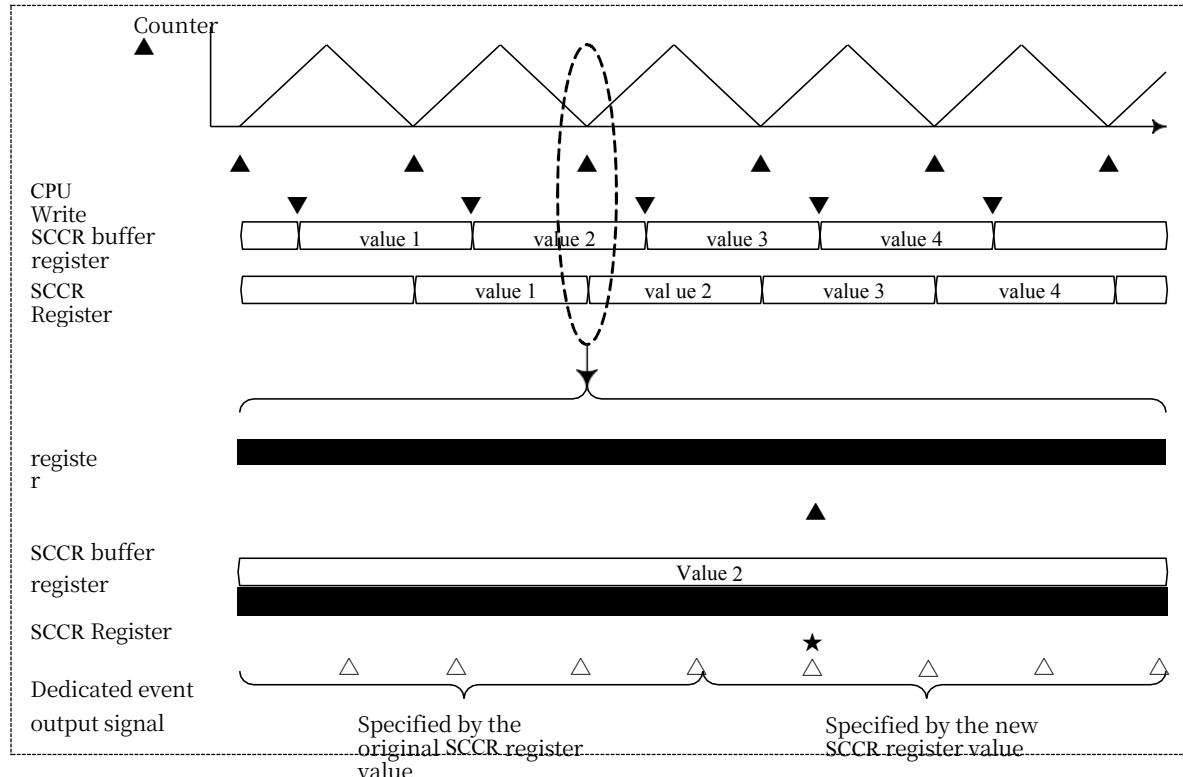


Figure 19-14 SCCR Buffered Transfer Operation (Cycle Interval Response Link Transfer Disable)

The top half of the figure is a global schematic, and the bottom half is a partially enlarged view of the buffer register transfer operation.

The counter is in delta wave counting mode, and the underflow interrupt is generated at the moment of flag ▲ (count valley). At the ▼ moment, the CPU overwrites the SCCR register and the written data is stored in the SCCR buffer register. Later, when the counter count value is 0x0000, a data load operation is performed and the data goes from the buffer register to the SCCR register, and the interrupt flag IRQZF is generated.

At the moment △, a comparison operation is performed between the set SCCR register value and the count value. After the moment ★ (count value = 0x0000), the dedicated event output signal performs an operation based on the new SCCR data. The operation is executed according to the SCCR data before the moment ★.

The figure illustrates the transfer operation of the SCCR buffer register at the count valley point, similar to the cache transfer operation of the SCMR's buffer register; ditto for the transfer operation at the count peak point. The new data takes effect immediately after the moment of transfer (the new write data sets the dedicated event output signal and the interrupt flag).

- b) When the counter cycle interval response link is enabled, a buffer register transfer operation is performed when the buffer value is in the set count state and the cycle interval counter count value is zero.

Figure 19-15 shows a schematic for enabling the SCCR buffer function, loading at counter zero (SCSR.BUFEN=01) and spacing between counter cycles in response to a link transfer

enable (SCSR.LMC=1).

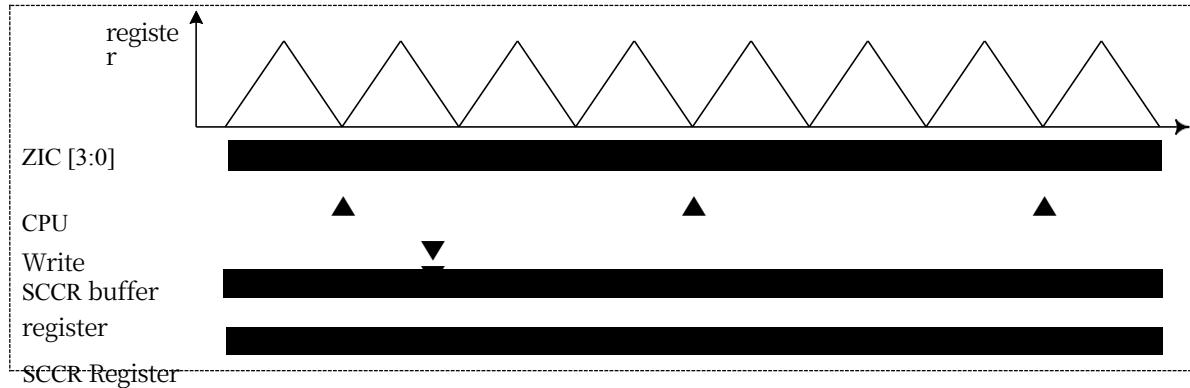


Figure 19-15 SCCR Buffered Transfer Operation (Periodic Interval in Response to Link Transfer Enable)

The counter is in delta wave counting mode, the cycle interval counter (CVPR.ZIC) counts decrementally from 2 to 0, and the underflow interrupt is at flag ▲.

(count valley)moment is generated. At the ▼ moment, the CPU overwrites the SCCR register and the written data is stored in the SCCR buffer register. Afterwards, when the counter count value is 0x0000 and the cycle interval counter (CVPR.ZIC) is 0, a data load operation is performed, and the data is transferred from the buffer register to the SCCR register, and the interrupt flag IRQZF is generated.

19.3.3 General Purpose PWM Output

19.3.3.1 Independent PWM output

In the pass-through mode (POCR.PWMMD=00), different PWM outputs can be realized by setting the reference values of OCCRxh and OCCRxl and the port output states of OCMRxh and OC MRxl ($x=u, v, w$) respectively. In this case, the PWM output of each port is controlled independently. Figures 19-16 and 19-17 show the examples of independent PWM outputs under sawtooth wave and triangle wave of Unit 1, respectively.

Attention:

- The straight-through mode means that the internal output signals (in_opxh, in_opxl) generated by the comparison match of the values of the general-purpose comparison reference registers (OCCRxh, OCCRxl) are directly output to the corresponding ports (TIM4_<t>_OXH, TIM4_<t>_OXL) ($x=U, V, W, x=u, v, w$). $x=u, v, w$)

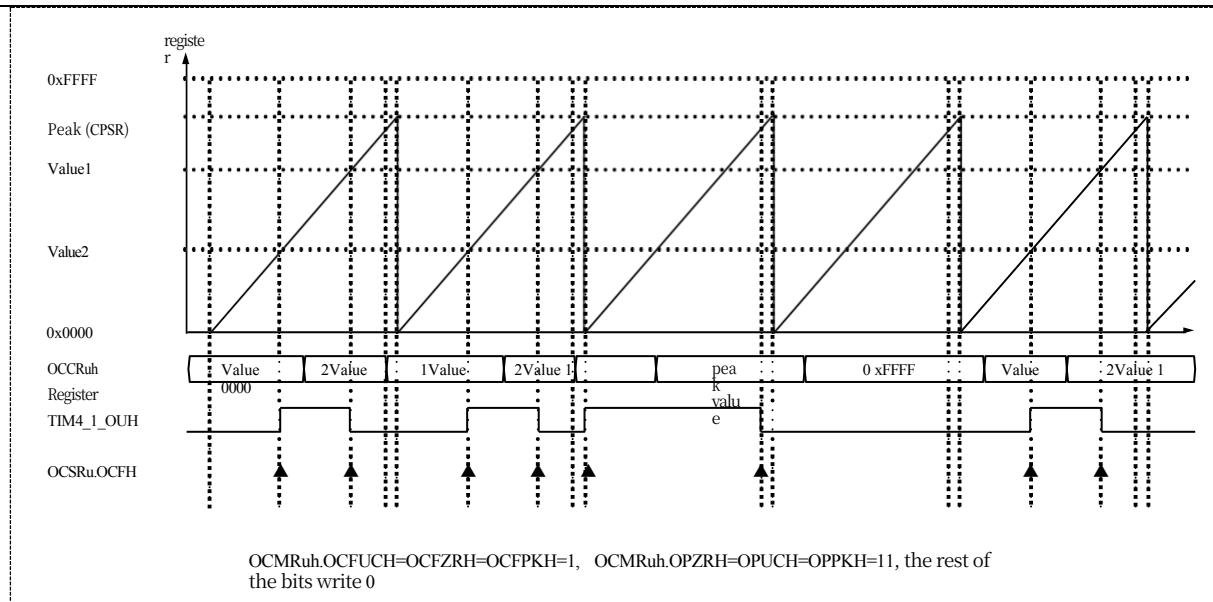


Figure 19-16 Sawtooth Independent PWM Output Example

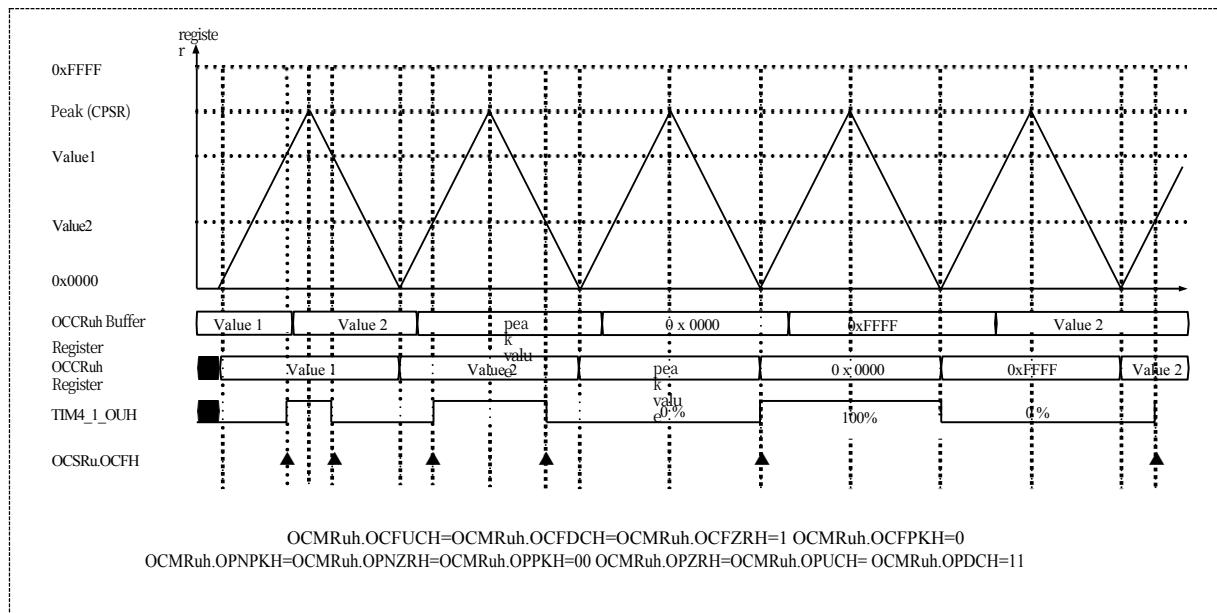


Figure 19-17 Triangle Wave Independent PWM Output Example

19.3.3.2 Extended PWM output

In pass-through mode (POCR.PWMMD=00), the port output state of TIM4_<t>_OXL can also be determined by the expansion bit (bit32~16) in the OCMRxl register, which is set with respect to the OCCRxh reference value, thus realizing the expanded PWM outputs on the TIM4_<t>_OXL port (x=U, V, W, x=u, v, w) Figure 19-18 shows the PWM outputs of TIM4_<t>_OUH and TIM4_<t>_OUL ports in this mode.

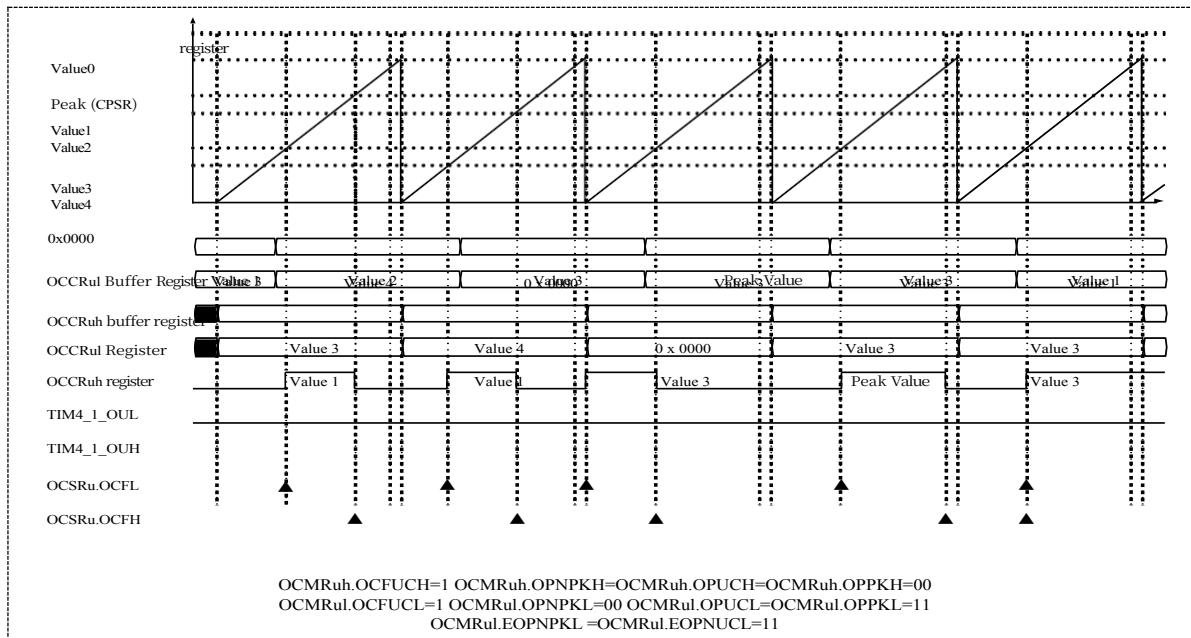


Figure 19-18 Sawtooth Expanded PWM Outputs

Attention:

- In independent PWM output mode, the port state of TIM4_<t>_OXL is determined by bit15~bit4 bits of OCMRxl register, which is only related to the OCCRxh reference value (X=U, V, W, x=u, v, w)

19.3.3.3 Complementary PWM output

Software setup for complementary PWM output

In pass-through mode (POCR.PWMMD=00), set the reference values of OCCRx_x and OCCRx_l ($x=u, v, w$) directly to realize a pair of complementary PWM waveform outputs to the ports, and set the ports of 3 groups in the same way to realize 3 complementary PWM outputs. As shown in Figure 19-19.

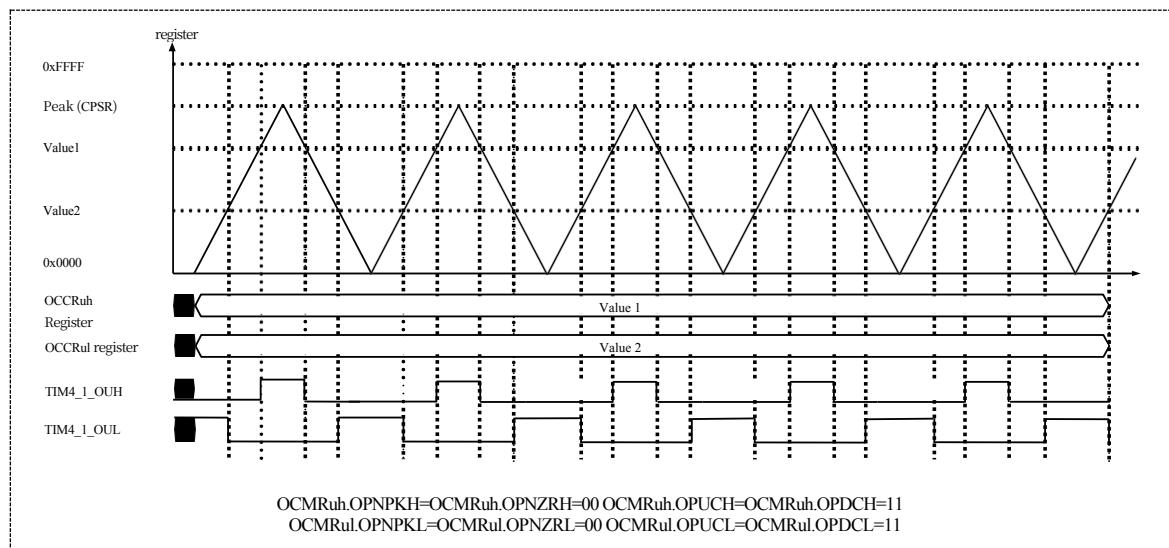


Figure 19-19 Software Implementation of Complementary PWM

Hardware setting for complementary PWM outputs

In deadband timer mode (POCR.PWMMD=01), the internal output signal (`in_opxl`) generated by comparison matching of the value of the general-purpose comparison reference register (OCCRx_l) and the set value of the PWM deadband control registers (PDAR/PDBR) are shifted through the timing to realize complementary PWM outputs in hardware.

In this mode, the polarity of the `TIM4_<t>_OXH` port output is the same as `in_opxl`, and the polarity of the `TIM4_<t>_OXL` port output is the opposite of `in_opxl` ($X=U, V, W, x=u, v, w$)

Figure 19-20 shows an example of a complementary PWM output in deadband timer mode.

If `in_opxl` rising edge is detected, `TIM4_<t>_OXL` output becomes low, the deadband counter loads the set value of PDBRx register and starts decremental counting, and when the count value becomes 0x0000, the counter stops and `TIM4_<t>_OXH` output goes high; if `in_opxl` falling edge is detected, `TIM4_<t>_OXH` output becomes low, the deadband counter loads the set value of PDARx register and starts decrementing count, when the count value becomes 0x0000, the counter stops and `TIM4_<t>_OXL` output goes high ($X=U, V, W, x=u, v, w$)

By setting the PWM deadband control registers PDAR and PDBR, the deadtime of the output rise

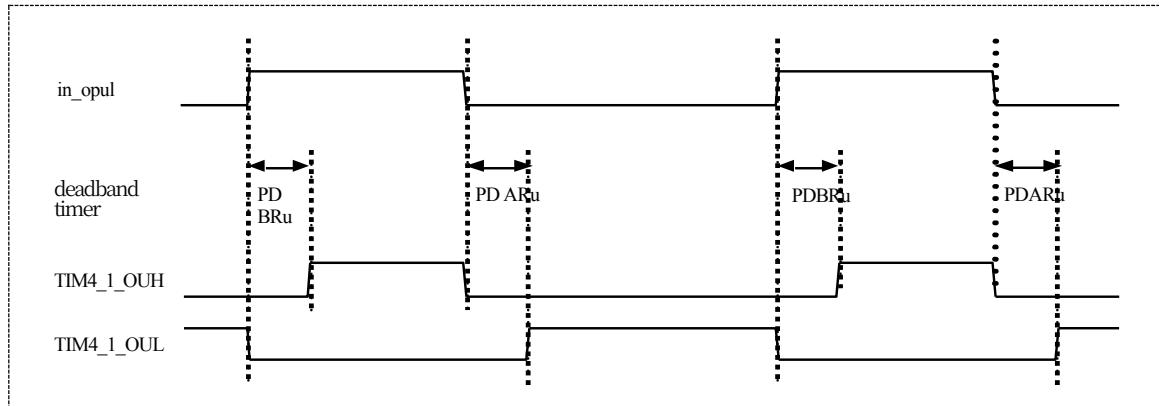


Figure 19-20 Complementary PWM Output in Deadband Timer Mode

When the width of the high level pulse of **in_opxl** is less than the dead time set by **PDBR**, only the **TIM4_<t>_OXL** output goes low. The **TIM4_<t>_OXL** output goes high when the dead time set by the **PDAR** register passes after the falling edge of **in_opxl**. In this case, the **TIM4_<t>_OXH** output will remain low continuously.

When the width of the low pulse of **in_opxl** is less than the dead time set by **PDAR**, only the **TIM4_<t>_OXH** output goes low. The **TIM4_<t>_OXH** output goes high when the dead time set by the **PDBR** register passes after the rising edge of **in_opxl**. In this case, the **TIM4_<t>_OXL** output will remain continuously low ($x=U, V, W, x=u, v, w$) As shown in Figure 19-21.

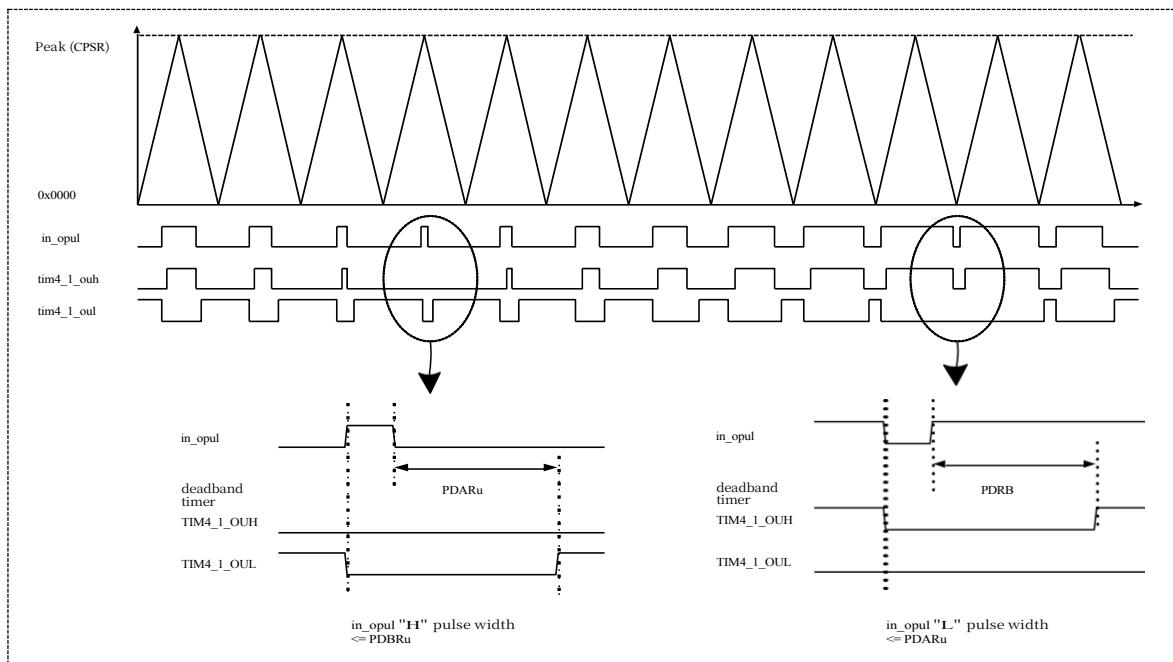


Figure 19-21 Waveform Output in Deadband Timer Mode for Pulse Width Abnormalities

On the basis of the above hardware implementation of the deadband output mode, it is also possible to monitor the pulse width of the internal comparison matching signal (**in_opxl**), thus realizing the filtering control of the **in_opxl** signal. This deadband output implementation of **in_opxl** with pulse width filtering is called Deadband Counter Filter Mode (POCR.PWMMD=10)($x=u, v, w$)

In the deadband counter filter mode, the filter width is determined by the setting of the PWM filter control register (PFSRn). When the pulse width of in_opxl is greater than the time set by register PFSRn, the filter counter delays the in_opxl signal for the time set by PFSR before outputting it, and then generates complementary PWM outputs (x=u, v, w) in the manner described in the deadband timer mode.

If the rising edge of signal in_opxl is detected, the filter counter loads the value of PFSR register and starts to measure the width of the high level of in_opxl, when the width of the high level pulse of in_opxl is greater than the time set by the register PFSR, after the time set by the PFSR, the output of TIM4_<t>_OXL goes low, and the dead zone counter When the count value becomes 0x0000, the counter stops and TIM4_<t>_OXH output goes high; if the falling edge of signal in_opxl is detected, the filter counter loads the value of PFSR register and starts to measure the width of the low level of in_opxl, when the width of the low level pulse of in_opxl is greater than the time set by PFSR register, TIM4_<t>_OXL output turns low, the dead zone counter will be stopped, the dead zone counter will be stopped and TIM4_<t>_OXH output goes high. When the pulse width of in_opxl is larger than the time set by PFSR, after the time set by PFSR, the output of TIM4_<t>_OXH becomes low, the deadband counter loads the set value of PDAR register and starts to count decreasingly, when the counting value becomes 0x0000, the counter stops and the output of TIM4_<t>_OXL goes high. When the count value becomes 0x0000, the counter stops and causes TIM4_<t>_OXL to output high. The outputs TIM4_<t>_OXH and TIM4_<t>_OXL will remain unchanged when the level pulse width of in_opxl is less than the time set by register PFSR (X=U,V,W, x=u,v,w)

Figure 19-22 shows an example of a complementary PWM output in deadband counter filter mode.

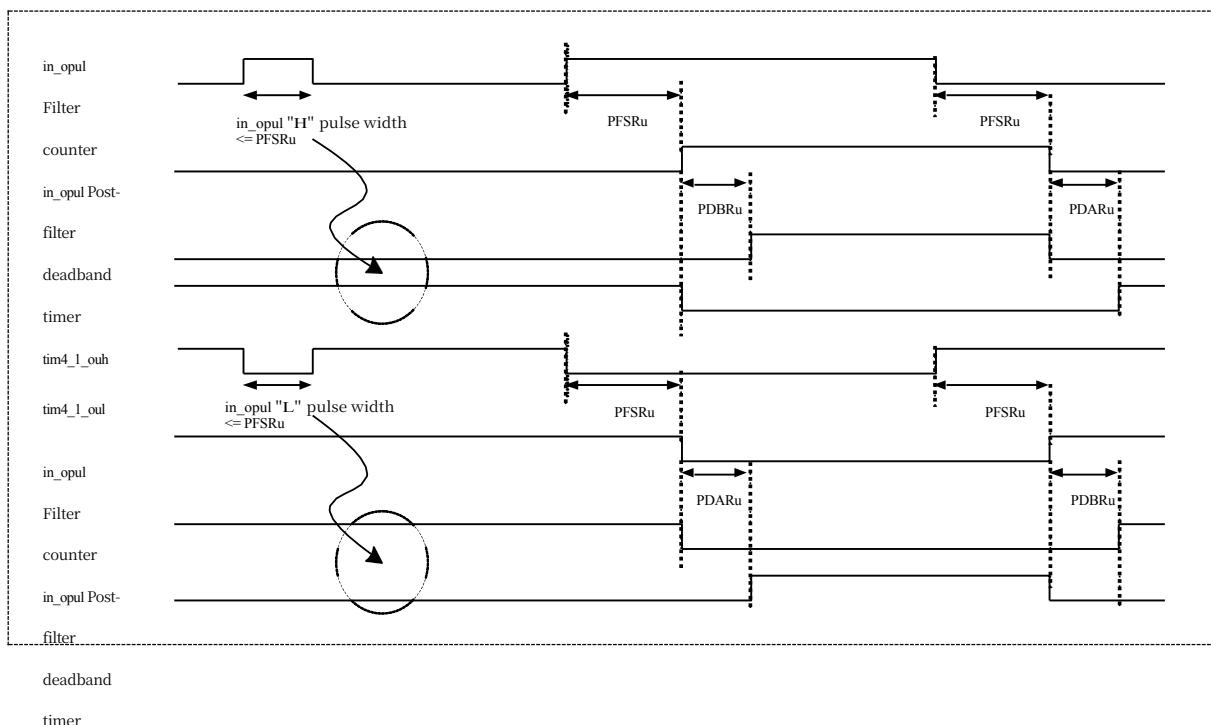


Figure 19-22 Complementary PWM Output in Deadband Timer Filter Mode

19.3.4 periodic interval response

The underflow interrupt mask counter is used to mask the number of times the underflow flag bit (CCSR.IRQZF) has been set. Underflow Interrupt Mask Counter (CVPR.ZIC[3:0]) operates as a decrementing counter and loads the value set by CVPR.ZIM[3:0] at the beginning, and when CVPR.ZIC[3:0] = "0", the underflow flag bit (CCSR.IRQZF) is set to "1".

The overflow interrupt mask counter is used to mask the number of times the overflow flag bit (CCSR.IRQPF) has been set. Overflow Interrupt Mask Counter (CVPR.PIC[3:0]) operates as a decrementing counter, loading the value set by CVPR.PIM[3:0] at the beginning, and when CVPR.PIC[3:0] = "0", the overflow flag bit (CCSR.IRQPF) is set to "1".

Figure 19-23 below shows the timing diagram of IRQZF and IRQPF setup for cycle interval response.

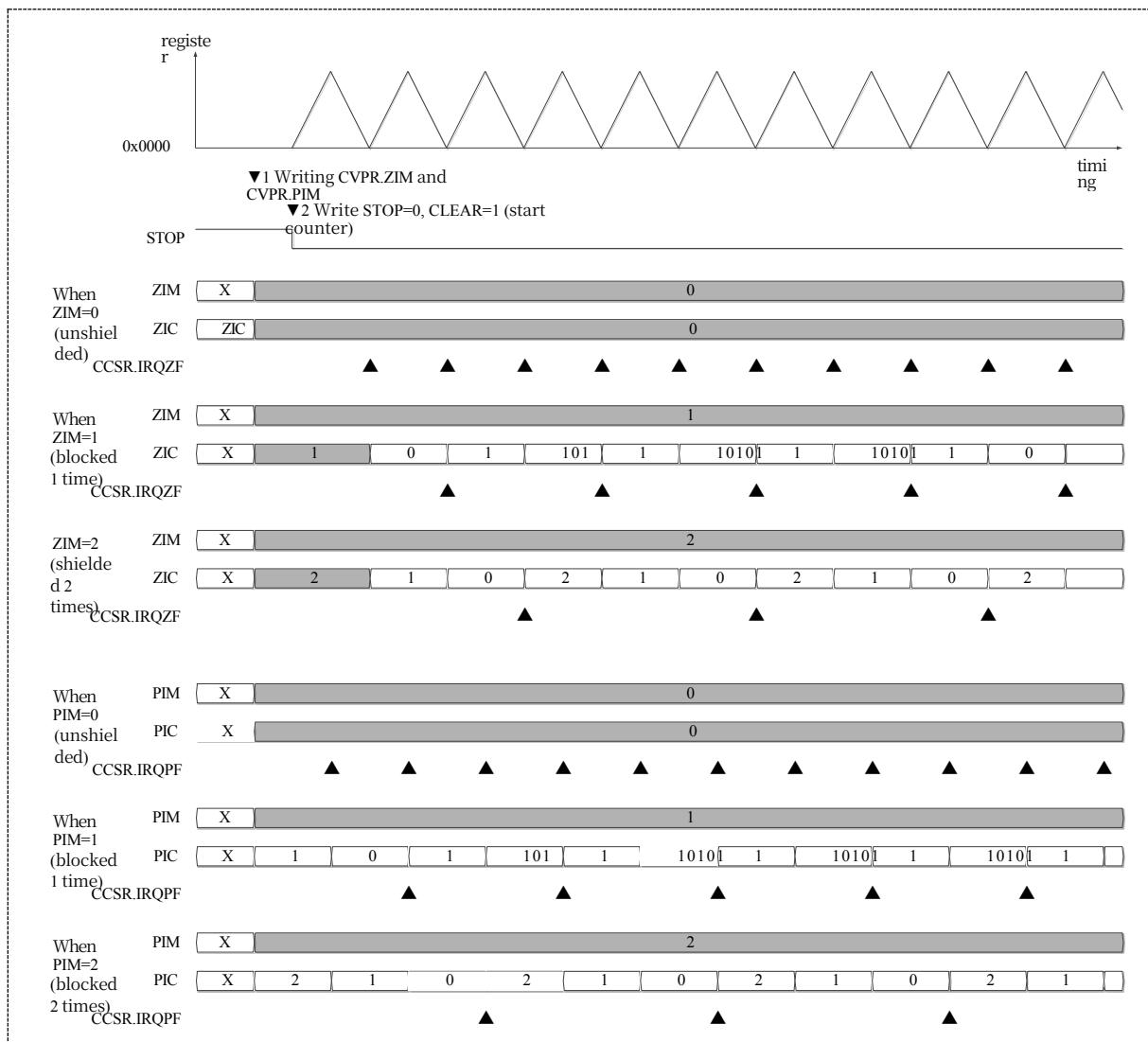


Figure 19-23 Cycle Interval Response Timing Diagram

▼1 When the counter stops, the CVPR.ZIM and CVPR.PIM initialization values are written, and the initial values are immediately reflected in the internal counters (CVPR.ZIC, CVPR.PIC).

▼2 Initialize and start the counter (**STOP=0** and **CLEAR=1**), the counter starts counting incrementally from zero after a bus reset or from software initialization with **CLEAR=1**, the **CCSR.IRQZF** flag will not be set immediately at this moment, **and the flag ▲** is the moment when **CCSR.IRQZF** or **CCSR.IRQPF** is set, and the flag **▲** is the moment when **CCSR.IRQZF** or **CCSR.IRQPF** is set, and the flag **▲** is the moment when **CCSR.IRQPF** is set, and the counter is set. **IRQZF** or **CCSR.IRQPF** is set.

Attention:

- Write **CVPR.ZIM** and **CVPR.PIM** while the counter is running and the set values will not immediately react to the interrupt mask counters (**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 values of the interrupt mask counters.

The comparison matching event (dedicated event output) of the dedicated comparison reference register (SCCRm) also has a cycle interval response function. When the dedicated control status register SCSR is set to the comparison mode output (**EVTMS=0**), the EVT event startup enable is for the upward overflow point (**ZEN**) upward counting time (**UEN**), downward overflow point (**PEN**), and downward counting time (**DEN**). When the cycle interval function is not used, turning on the enable sets the dedicated event output signals whenever the corresponding count match event occurs. Among them, the cycle interval function is not supported for the up-overflow point matching event and the down-overflow point matching event. When the cycle interval function is used, in the triangle wave mode, when **UEN=1** or **DEN=1** and the count value reaches the value set in the **SCMR_m** register, one EVT event is generated in one valid cycle; when **UEN=1** and **DEN=1** and the count value reaches the value set in the **SCMR_m** register, two EVT events are generated in one valid cycle. Figure 19-24 below shows the cycle interval response output graph of the dedicated event output signal.

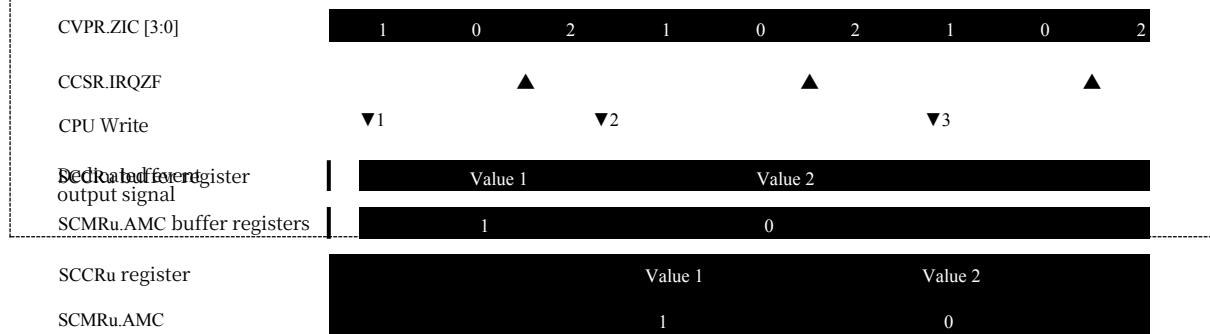


Figure 19-24 Dedicated Event Output Signal Cycle Interval Response Outputs

The counter is in delta wave counting mode and the underflow interrupt mask counter (CVPR.ZIC) counts in decrements from 2-0. The underflow interrupt is counted at the moment

▲ Generation.

At time ▼1, write value 1 to the SCCR_U buffer register, and at the same time, MZCE=1, MPCE=0, and AMC=0001 are written to the SCMR_U buffer register. Turn on the upward counting EVT enable (UEN=1) in the SCSR register, and the cycle interval response function link enable (LMC=1), and set the underflow buffer (BUFEN=01) and the buffer register SCCR_U and SCMR_U transfer operation is executed at ▲ time. Since the count value = SCCR_U=Value 1 and MZCE=1 and AMC=1 during upward counting, the dedicated event output signal is set at moment △1.

At moment ▼2, value 2 is written to the SCCR_U buffer register, and at the same time, MZCE=1, MPCE=0, and AMC=0000 are written to the SCMR_U buffer register. After that, the buffer register SCCR_U and SCMR_U transfer operation is executed at the ▲ moment. Since MZCE=1 and AMC=0 and count value=SCCR_U=Value 2 during upward counting, the dedicated event output signal is set at moment △2.

19.3.5 EMB Control

Each Timer4 unit has an Output Invalid Event interface to the EMB events output by the EMB module. The abnormal condition events selected on this interface can be set from the EMB (see [Emergency Brake Module (EMB] section)

The output state of the 3 PWM ports in each unit can be changed to a preset state if an abnormal EMB event from the EMB is monitored during normal output. This preset port state can be Output High Resistive, Output Low, Output High, Maintain Normal Output, and Maintain the previous state unchanged. (Set by ECER.EMBVAL and ECSR.HOLD)

For example, if an EMB event is generated during the normal output of Timer4's PWM port when ECER.EMBVAL=01 is set, the output on the PWM port changes to a high resistance state.

19.4 Interrupt and Event Description

19.4.1 Count Compare Match Interrupt

There are six general-purpose comparison reference value registers (OCCR_m), which can be used to compare with the count value to generate a valid signal for comparison. OCSR_n.OCFH and OCSR_n.OCFL bits in the general control status register (OCSR_n) are set to 1 when the count is matched, and if OCSR_n.OCIEH and OCSR_n.OCIEL are set to enable interrupts, the corresponding interrupt requests (TMR4_U<t>_GCM_m, m=U, V, W; n=H, L) will also be triggered. H, L) will also be triggered.

19.4.2 Count Cycle Matching Interrupt

The CCSR.IRQPF or CCSR.IRQZF bit of the Control Status Register (CCSR) is set to 1 when the sawtooth wave incremental count reaches the upper overflow point, the delta wave count reaches the valley point, or the delta wave count reaches the peak point, and the CCSR.IRQZF bit of the Control Status Register (CCSR) is set to 1 when the CCSR.IRQPEN or

CCSR.IRQZEN bit is set.

Enabling interrupts allows the count cycle matching interrupts (TMR4_U<math><math>_GOVF and TMR4_U<math><math>_GUDF) to be triggered at the corresponding time points.

19.4.3 Overloaded Count Match Interrupt

When the reload function is active, TMR4_PFSRn serves as the cycle count value of the reload timer, which is reloaded from register TMR4_PFSRn to the initial value of the counter when the counting is enabled, and performs the decrement operation. After completing a cycle, a reload count interrupt request is generated, and the RCSR.RTIFU, RCSR.RTIFV bits in the reload control status register (RCSR) will be set to 1 respectively, RCSR.RTIFU, RCSR.RTIFV, and RCSR.RTIFW bits in the Reload Control Status Register (RCSR) are set to 1 respectively.

At this time, if the setting RCSR.RTIDU, RCSR.RTIDV, RCSR.RTIDW interrupt masks are invalid, the corresponding reload count match interrupt request (TMR4_U<t>_RLOm, m=U, V, W) will also be triggered.

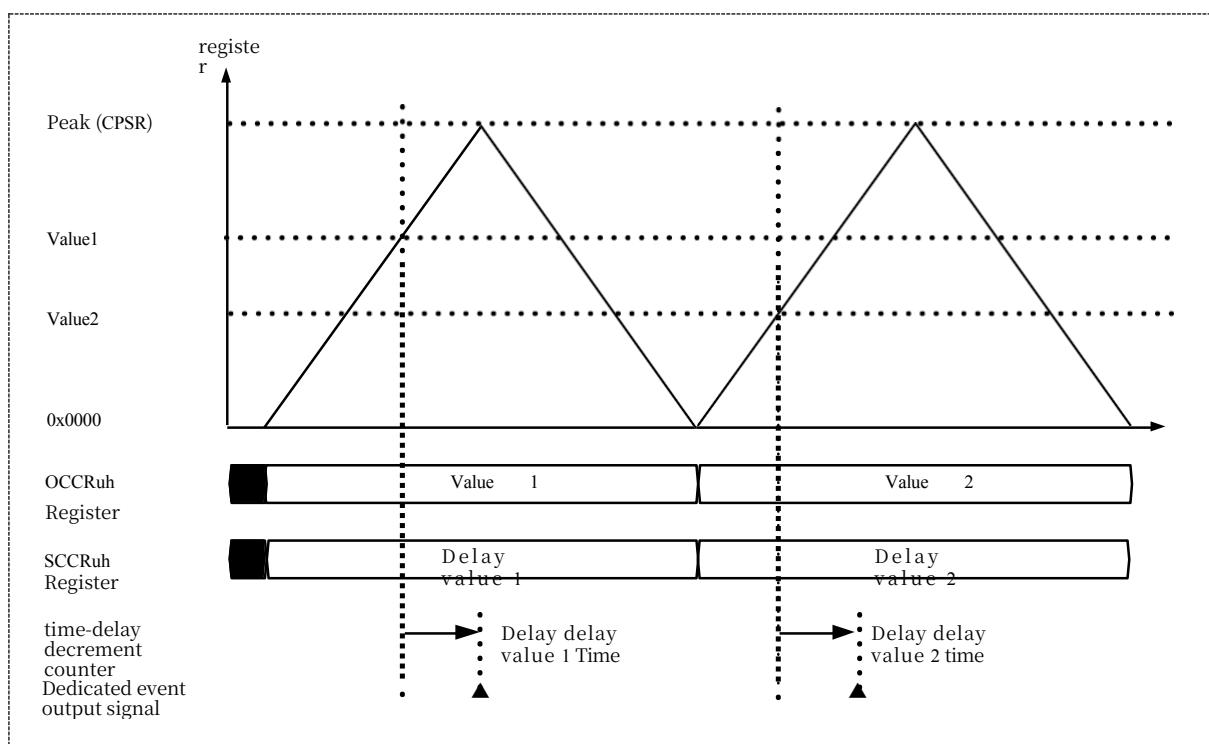
19.4.4 Dedicated Compare Match Event

The six dedicated comparison reference registers (SCCRm) of Timer4 generate six dedicated event output signals that can be used to selectively trigger other modules, such as starting the ADC.

During clock counting, if a count comparison match event (TMR4<t>_SCM0~5) occurs for the dedicated comparison reference value (SCCRm), a corresponding valid request signal is generated, which can be configured to any of the event EVT output signals (set by the SCSR.EVTOS bit) for triggering other modules.

The output of the event request signal can be selected in either compare start mode or delayed start mode. In comparative start mode

(SCSR.EVTMS=0) the dedicated event output signal is output directly after generating the count comparison matching event of SCCR; in delayed start mode (SCSR.EVTMS=1) the dedicated event output signal is output after generating the count comparison matching event of OCCRxh or OCCRxI (selected by the bit of SCSR.EVTDS; x=u, v, w) after the base reference set by SCCR. After the cycle time set by SCCR, the dedicated event output signal is output. Figure 19-25 below shows a request output example of the dedicated event output signal in delayed start mode.



Attention:

- During a delayed counting run, if an OCCR match event to the counter occurs again, the delayed counter reloads the count value and resumes decremental counting. Therefore, if the OCCR match event interval is less than the set delay time SCCR, the request signal for the dedicated event output may never be generated.

19.5 Register Description

Table 19-3 shows the register list of the Timer4 module.

BASE ADDR: 0x4001_7000 (U1) 0x4002_4800 (U2) 0x4002_4C00 (U3)

Table 19-3 Register List

register name	notation	offset	bit width	reset value
register for counting values	TMR4_CNTR	0x0046	16	0x0000
Cycle Base Register	TMR4_CPSR	0x0042	16	0xFFFF
Control Status Register	TMR4_CCSR	0x0048	16	0x0040
Efficient Cycle Register	TMR4_CVPR	0x004A	16	0x0000
Universal Comparison Reference Register UH	TMR4_OCCRuh	0x0002	16	0x0000
Universal Comparison Reference Register UL	TMR4_OCCRul	0x0006	16	0x0000
General purpose comparison reference register VH	TMR4_OCCRvh	0x000A	16	0x0000
Generalized Comparison Reference Register VL	TMR4_OCCRvl	0x000E	16	0x0000
Generalized Comparison Reference Register WH	TMR4_OCCRwh	0x0012	16	0x0000
Generalized Comparison Reference Register WL	TMR4_OCCRwl	0x0016	16	0x0000
General-purpose control status register U	TMR4_OCSRu	0x0018	16	0xFF00
General purpose control status register V	TMR4_OCSRv	0x001C	16	0xFF00
General control status register W	TMR4_OCSRw	0x0020	16	0xFF00
General Purpose Extended Control Register U	TMR4_OCERu	0x001A	16	0x0000
General Purpose Extended Control Register V	TMR4_OCERv	0x001E	16	0x0000
General Purpose Extended Control Register W	TMR4_OCERw	0x0022	16	0x0000
General-purpose mode control register UH	TMR4_OCMRuh	0x0024	16	0x0000
General Purpose Mode Control Register UL	TMR4_OCMRul	0x0028	32	0x0000_0000
General-purpose mode control register VH	TMR4_OCMRvh	0x002C	16	0x0000
General Purpose Mode Control	TMR4_OCMRvl	0x0030	32	0x0000_0000

Register VL				
General Purpose Mode Control Register WH	TMR4_OCMRwh	0x0034	16	0x0000
General Purpose Mode Control Register WL	TMR4_OCMRwl	0x0038	32	0x0000_0000
Dedicated Comparison Reference Register UH	TMR4_SCCRuh	0x00B2	16	0x0000
Dedicated Comparison Reference Register UL	TMR4_SCCRul	0x00B6	16	0x0000
Dedicated Comparison Reference Register VH	TMR4_SCCRvh	0x00BA	16	0x0000
Dedicated Comparison Reference Register VL	TMR4_SCCRvl	0x00BE	16	0x0000
Dedicated Comparison Reference Register WH	TMR4_SCCRwh	0x00C2	16	0x0000
Dedicated Comparison Reference Register WL	TMR4_SCCRwl	0x00C6	16	0x0000
Dedicated control status register UH	TMR4_SCSRuh	0x00C8	16	0x0000
Dedicated control status register UL	TMR4_SCSRul	0x00CC	16	0x0000
Dedicated control status register VH	TMR4_SCSRvh	0x00D0	16	0x0000
Dedicated control status register VL	TMR4_SCSRvl	0x00D4	16	0x0000

Dedicated control status register WH	TMR4_SCSRwh	0x00D8	16	0x0000
Dedicated control status register WL	TMR4_SCSRwl	0x00DC	16	0x0000
Dedicated mode control register UH	TMR4_SCMRuh	0x00CA	16	0xFF00
Dedicated mode control register UL	TMR4_SCMRul	0x00CE	16	0xFF00
Dedicated mode control register VH	TMR4_SCMRvh	0x00D2	16	0xFF00
Dedicated mode control register VL	TMR4_SCMRvl	0x00D6	16	0xFF00
Dedicated mode control register WH	TMR4_SCMRwh	0x00DA	16	0xFF00
Dedicated mode control register WL	TMR4_SCMRwl	0x00DE	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 register W	TMR4_POCRw	0x00A0	16	0xFF00
PWM Filter Control Register U	TMR4_PFSRu	0x0082	16	0x0000
PWM filter control register V	TMR4_PFSRv	0x008A	16	0x0000
PWM filter control register W	TMR4_PFSRw	0x0092	16	0x0000
PWM deadband control register AU	TMR4_PDARu	0x0084	16	0x0000
PWM deadband control register BU	TMR4_PDBRu	0x0086	16	0x0000
PWM deadband control register AV	TMR4_PDARv	0x008C	16	0x0000
PWM Deadband Control Register BV	TMR4_PDBRv	0x008E	16	0x0000
PWM deadband control register AW	TMR4_PDARw	0x0094	16	0x0000
PWM deadband control register BW	TMR4_PDBRw	0x0096	16	0x0000
Reload Control Status Register	TMR4_RCSR	0x00A4	16	0x0000
EMB Control Status Register	TMR4_ECSR	0x00F0	16	0x0000
EMB Extended Control Register	TMR4_ECER	U1: (0x4005_5408) U2: (0x4005_540C) U3: (0x4005_5410)	32	0x0000_0000

Attention:

- 下述寄存器详细描述中，m=uh、ul、vh、vl、wh、wl，n=u、v、w。m所指寄存器分别在功能实现时对应端口 TIM4_{t}_OUH、TIM4_{t}_OUL、TIM4_{t}_OVH、

TIM4_<t>_OVL, TIM4_<t>_OWH, TIM4_<t>_OWL output control, etc.; n the registers referred to correspond to the output control of ports **TIM4_<t>_OUx, TIM4_<t>_OVx, TIM4_<t>_OWx**, etc., respectively, at the time of function realization. where x = H or L, and the specific control of H or L has a corresponding symmetry bit in these registers.

19.5.1 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]															
classifier for honorific people	marking	celebrity						functionality							
b15~b0	CNTR[15:0]	Counter Current Value	The counter count value can be initialized by writing a value to this register when counting is stopped This bit indicates the current counter count value when counting is in progress Note: No value can be written to this register while the count is in progress						R/W						

19.5.2 Cycle 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]															
classifier for honorific people	marking	celebrity						functionality							
b15~b0	CPSR [15:0]	Generic periodic benchmarks	Counter cycle value of the counter Note: When reading data from this address area, it is not the value of the buffer register that is read, but the value of the CPSR register						R/W						

19.5.3 Control Status Register (TMR4_CCSR)

Reset value: 0x0040

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ECK EN	IRQ ZF	IRQ ZEN	-	-	-	IRQ PF	IRQ PEN	BUF EN	STOP	MODE	CLEAR				CKDIV[3:0]

classifier for marking		Bit Name	Function	Read/Write
honorific	people			
b15	ECKEN	Clock Source Selection	<p>0: Internal PCLK1 clock</p> <p>1: External TIM4_<t>_CLK port input clock</p> <p>Note 1: This bit is set when the counter is stopped</p> <p>Note 2: Write when using the external TIM4_<t>_CLK port for R/W clock input</p> <p>After STOP="1", the first edge of the external input clock is regarded as invalid, and the counting action starts from the second edge, and both the rising and falling edges are valid edges.</p>	
b14	IRQZF	Underflow status	<p>0: No count overflow occurred</p> <p>1: Count overflow occurs</p> <p>Note 1: When the cycle interval response function is used, the set condition of this bit is set by CVPR.</p> <p>Cycle interval counter setting for R/W setting</p> <p>Note 2: When the counter is reset by the bus or CLEAR="1" is written, the IRQZF bit will not be set.</p>	
b13	IRQZEN	Underflow interrupt enable	<p>0: Disable IRQZF to generate interrupt to CPU</p> <p>1: allow IRQZF to generate interrupt to CPU</p> <p>R/W</p>	0:
b12~b10	Reserved	-O" for reading, "O" for writing. "0", writes "0".	R/W	Reads
b9	IRQPF	Overflow status	<p>0: No count overflow occurred</p> <p>1: Count overflow occurs</p> <p>Note 1: When the cycle interval response function is used, the set condition of this bit is set by the CVPR</p> <p>the cycle interval counter.</p> <p>R/W set by</p> <p>Note 2: When the counter is reset by the bus or CLEAR="1" is written, the IRQZF bit will not be set.</p>	
b8	IRQOPEN	Overflow interrupt enable	<p>0: Disable IRQPF from generating interrupts to the CPU</p> <p>1: Allow IRQPF to generate interrupts to the CPU</p> <p>R/W</p>	0: Disable
b7	BUFEN	Cache Enable	<p>0: Disable CPSR cache function</p> <p>1: Enable CPSR cache function</p>	R/W
b6	STOP	Counter Enable	<p>0: Counter start</p> <p>1: Counter stop</p>	R/W
b5	MODE	Waveform Mode	<p>0: Ragged wave mode (upward counting only supported)</p> <p>1: Triangle wave mode</p> <p>R/W</p>	
b4	CLEAR	Counter Clear	<p>0: No operation</p> <p>1: Counter Clear</p> <p>Note: This bit is always 0 when read out</p>	R/W

		This bit indicates the count clock division of the basic counter		
		0000: the count clock is PCLK1		
b3~b0	CKDIV	counter clock division	0001: Count clock is PCLK1/2	R/W
			0010: Count clock is PCLK1/4	
			0011: Count clock is PCLK1/8	
			0100: Count clock is PCLK1/16	
			0101: Count clock is PCLK1/32	

0110: Count clock is PCLK1/64

0111: Count clock is PCLK1/128

1000: Count clock is PCLK1/256

1001: count clock is PCLK1/512

1010: The counting clock is

PCLK1/1024, please do not set

other values.

Note: The counting clock source is the external TIM4_<t>_CLK port input clock, the divider setting is invalidated

19.5.4 Valid Cycle 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/Write	
b15~b12	PIC[3:0]	Overflow interrupt mask state		Number of overflow interrupts that need to be masked currently Note 1: Rewriting the PIM when the counter is stopped, the PIC is immediately updated to the new PIM value Note 2: If the PIM is rewritten before the overflow occurs after counting begins, the PIC is updated at the time of overflow to the new PIM value and the overflow interrupt is still triggered. R the PIC is decremented for each subsequent overflow Note 3: If the PIM is rewritten during an overflow interrupt, the PIC is updated to the new PIM value at the next overflow and the next overflow interrupt is still triggered. The PIC is decremented for each subsequent overflow, and the number of underflow interrupts that need to be masked is decremented.
b11~b8	ZIC [3:0]	underflow interrupt mask state		Note 1: Rewriting the ZIM when the counter is stopped immediately updates the ZIC to the new ZIM value Note 2: If ZIM is rewritten before the underflow occurs after counting begins, ZIC is updated at the time of the underflow as new ZIM value and still triggers this underflow interrupt. The ZIC is decremented for each underflow thereafter Note 3: If the ZIM is rewritten in the underflow interrupt, the PIC is updated to the new value at the next underflow. of the ZIM value and still triggers the next underflow interrupt. Each subsequent underflow ZIC decrements
b7~b4	PIM[3:0]	Overflow interrupt mask setting		R/W Setting the number of masked overflow interrupts
b3~b0	ZIM [3:0]	Underflow Interrupt Mask Setting		R/W Setting the number of masked underflow interrupts

19.5.5 General purpose comparison reference 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]															
classifier for honorific people	marking				celebrity				functionality				fill out or in (information on a form)		
b15~b0	OCCR[15:0]				Generalized comparator values				Note: When reading data from this address area, it is not the value of the buffer register that is read, but the value of the is the value of the OCCR register				R/W		

19.5.6 General Purpose Control Status Register (TMR4_OCSRn)

Reset value: 0xFF00

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
Reserved								OCFL L	OCFH H	OCIEL L	OCIEH H	OCPLOCEL=0	OCPH L	OCPH H	OCEL L	OCEH H
classifier for honorific people	marking	celebrity	functionality								fill out or in (information on a form)					
b15~b8	Reserved	-	Reads "1" and writes "1".								R/W					
b7	OCFL	Count Match L	0: Counter count value is not equal to OCCRxl setting value 1: Counter count value is equal to OCCRxl set value (x = u, v, w) Note: This bit must be valid when OCEL=1								R/W					
b6	OCFH	Count Match H	0: Counter count value is not equal to OCCRhx set value 1: Counter count value is equal to OCCRhx set value (x = u, v, w) Note: This bit must be valid when OCEH=1								R/W					
b5	OCIEL	Count Match L interrupt enable	interrupt occurs when OCFL is set 0: No 1: Interrupt occurs when OCFL is set								R/W					
b4	OCIEH	Count Match H interrupt enable	interrupt occurs when OCFH is set 0: No 1: Interrupt occurs when OCFH is set								R/W					
b3	Port status with Enable L	OCPL OCEL=0	0: Low output on in_opxl when OCEL=0 1: Output high on in_opxl when OCEL=0 (x = u, v, w) Note: Read OCPL, 0: current output high; 1: current output low Note: When OCEL=1, write operation is invalid								R/W					
b2	Port state with Enable H	OCPH OCEH=0	0: Low output on in_opxh when OCEH=0 1: Output high on in_opxh when OCEH=0 (x = u, v, w) Note: Write operation is invalid when OCEH=1								R/W					
b1	OCEL	Port Output	0: Compare output is invalid, in_opxl port state is determined by OCPL 1: Comparison output valid, in_opxl port state is determined by OCMRyl setting and OCFL state (x = u, v, w, y = U, V, W)								R/W					
b0	OCEH	Port Output	0: Comparison output is invalid, in_opxh port state is determined by OCPH 1: Compare output valid, in_opxh port state is determined by OCMRyh setting and OCFH state (x = u, v, w, y = U, V, W)								R/W					

19.5.7 General Purpose 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]				

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b15~b14	Reserved	-	Reads "0", writes "0".	R/W
b13	MCECL	Upflow point match enable L	0: When the count value reaches the overflow point (CNTR=CPSR) and OCCRxI > CPSR, comparative matching is disabled L 1: The count value reaches the overflow point (CNTR=CPSR) and OCCRxI > CPSR A comparison match is considered to have occurred when L ($x = u, v, w$)	R/W
b12	MCECH Enable H	Overflow Match	0: When the count value reaches the overflow point (CNTR=CPSR) and OCCRxI > CPSR, comparative matching is disabled H 1: When the count value reaches the overflow point (CNTR=CPSR) and OCCRxI > CPSR, a comparison match is considered to have occurred H ($x = u, v, w$)	R/W
		OCMRxI cycle interval ringing	0: Cycle interval response function link is invalid, OCMRxI cache transfer is determined by MLBUFEN setting 1: The Cycle Interval Response function link is valid, and the OCMRxI's cache is transmitted in the	
b11	LMMI	b8 LMCH	Should the functional link L	The MLBUFEN setting must also be satisfied on the basis of the CVPR.PIC[3:0]=0000 (when counting overflow) or CVPR.ZIC[3 :0]=0000 (when counting underflow) ($x = u, v, w$)
b10	LMMH		OCCRxh Cycle Interval Response Function Link H	0: Cycle interval response function link is invalid, and the cache transfer of OCMRxh is determined by the MHBUFEN[1:0]
b9	LMCL		OCCRxh Cycle Interval Response Function Link H	

1: Cycle Interval

Response Function Link

is valid, and the cache transfer of OCCRxh must also satisfy

CVPR.PIC[3:0]=0000

(in case of count overflow) or

CVPR.ZIC[3:0]=0000

(in case of count underflow) in addition to the MHBUFEN[1:0] setting

($x = u, v, w$) R/W

0: Cycle interval response

function link is invalid, and the cache transfer of OCCRxl is determined by the CLBUFEN[1:0] setting

1: The cycle interval R/W

response function link is valid, and the cache transfer of OCCRxl must also satisfy

CVPR.PIC[3:0]=0000

(in case of count overflow) or

CVPR.ZIC[3:0]=0000

(in case of count underflow) in addition to the CLBUFEN[1:0] setting

($x = u, v, w$)

0: Cycle interval response

function link is invalid, OCCRxh cache transfer is determined by CHBUFEN[1:0] setting

1: The cycle interval R/W

response function link is valid, and the cache transfer of OCCRxh must also satisfy

CVPR.PIC[3:0]=0000

(in case of count overflow) or

CVPR.ZIC[3:0]=0000

(in case of count underflow) in addition to the CHBUFEN[1:0] setting

($x = u, v, w$)

			00: The value of the OCMRxl cache register is written directly to OCMRxl	
b7~b6 transfer	MLBUFEN[1:0]	OCMRxl cache	01: The value of the OCMRxl cache register is written to OCMRxl on count underflow	R/W
			10: The value of the OCMRxl cache register is written to OCMRxl on count overflow	
			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 forwarding	MHBUFEN[1:0]	OCMRxh cache	00: The value of the OCMRxh cache register is written directly to OCMRxh	
			01: The value of the OCMRxh cache register is written to OCMRxh on count underflow	
			10: The value of the OCMRxh cache register is written to OCMRxh on count overflow	R/W
b3~b2 transfer	CLBUFEN[1:0]	OCCRxl cache	11: The value of the OCMRxh cache register is written to OCMRxh on count underflow or overflow	
			(x = u, v, w)	
			00: The value of the OCCRxl cache register is written directly to the OCCRxl	
			01: The value of the OCCRxl cache register is written to OCCRxl on count underflow	R/W
b1~b0 transfer	CHBUFEN[1:0]	OCCRxh cache	10: The value of the OCCRxl cache register is written to OCCRxl on count overflow	
			11: The value of the OCCRxl cache register is written to OCCRxl on count underflow or overflow	
			(x = u, v, w)	
			00: The value of the OCCRxh cache register is written directly to OCCRxh	
			01: The value of the OCCRxh cache register is written to OCCRxh on count underflow	
			10: The value of the OCCRxh cache register is written to OCCRxh on count overflow	
			11: The value of the OCCRxh cache register is written to OCCRxh on count underflow or overflow	
			(x = u, v, w)	

19.5.8 General Purpose 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]	OP ZRH	OCF UCH	OCF PKH	OCF DCH						

Note: This register bit description is used when OCMRuh, OCMRvh, OCMRwh

Bit Flag	Bit Name	Function	Read/Write	Condition	($x = u, v, w$)	Condition:	Count	Overflow &	OCCR _{xh} count	mismatch	($x = u, v, w$) 00:	In_opxh port	bits remain unchanged when condition is met	when condition is met	01: In_opxh port bit outputs high when the condition is met	10: In_opxh port bit outputs low when the condition is met	11: In_opxh port bit output inverted when condition is met	
b15~b14	OPNZRH[1:0]	The underflow point OCCR _{xh} does not match the mating state H		b5~b4	OPDCH[1:0]	Counting down										01: In_opxh port bit outputs high when the condition is met	10: In_opxh port bit outputs low when the condition is met	11: In_opxh port bit output inverted when condition is met
b13~b12	OPNPKH[1:0]	The upper overflow point OCCR _{xh} does not match the mating state H																
b11~b10	OPZRH[1:0]	Underflow point OCCR _{xh} match																
b9~b8	OPUCH[1:0]	Count Up OCCR _{xh}																
b7~b6	OPPKH[1:0]	Overflow point OCCR _{xh} match																

($x = u, v, w$)

Condition: count underflow & OCCRxh count

match ($x=u, v, w$) 00: In_opxh port bits remain

unchanged when condition is met

01: In_opxh port bit outputs high when the condition is

met

R/W

10: When the condition is satisfied, the In_opxh port bit
outputs a low level

11: In_opxh port bit output inverted when condition is
met

($x = u, v, w$)

Condition: counter counts up & OCCRxh counts match

R/W

($x=u, v, w$) 00: In_opxh port bit remains unchanged
when condition is met

01: In_opxh port bit outputs high when the condition is
met

R/W

10: When the condition is satisfied, the In_opxh port bit
outputs a low level

11: In_opxh port bit output inverted when condition is
met

($x = u, v, w$)

Condition: counter overflow & OCCRxh count

R/W

match ($x=u, v, w$) 00: In_opxh port bits remain
unchanged when condition is met

01: In_opxh port bit outputs high when the condition is
met

R/W

10: In_opxh port bit outputs low when the condition is
met

11: In_opxh port bit output inverted when condition is
met

($x = u, v, w$)

Condition: counter counts down & OCCRxh counts

R/W

match ($x=u, v, w$) 00: In_opxh port bits remain
unchanged when condition is satisfied

01: In_opxh port bit outputs high when the condition is
met

10: In_opxh port bit outputs low when the condition is
met

11: In_opxh port bit output inverted when condition is
met

($x = u, v, w$)

b3 OCFZRH Underflow point OCFH state H
underflow & OCCRxh count match ($x=u, v, w$) R/W

condition: count

			0: OCSR.OCFH bits remain unchanged when the condition is met 1: OCSR.OCFH position bit when the condition is satisfied	
b2	OCFUCH	Count Up OCFH Status H	Condition: counter count up & OCCRxh count match ($x=u, v, w$) 0: OCSR.OCFH bits remain unchanged when the condition is met 1: OCSR.OCFH position bit when the condition is satisfied	R/W
b1	OCFPKH	Overflow point OCFH state H	Condition: count overflow & OCCRxh count match ($x=u, v, w$) 0: OCSR.OCFH bits remain unchanged when the condition is met 1: OCSR.OCFH position bit when the condition is satisfied	R/W
b0	OCFDCH	Counting Down OCFH Status H	Condition: counter count down & OCCRxh count match ($x=u, v, w$) 0: OCSR.OCFH bits remain unchanged when the condition is met 1: OCSR.OCFH position bit when the condition is satisfied	R/W

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17
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]							
b16 b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1
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					
50														

Note: This register bit description is used when OCMRul, OCMRvl, OCMRwl

Bit Flag	Bit Name Function Read/Write	Conditions: count underflow & OCCRxl count mismatch & OCCRxh count match ($x=u, v, w$)	
b31~b30 EOPNZRL[1:0]		Extended underflow point OCCRxl does not match & OCCRxh matches state L	Extended underflow point OCCRxl match & OCCRxh match state L
b27~b26 EOPZRL[1:0]			
b29~b28 EOPNPKL[1:0]		Extended overflow point OCCRxl does not match & OCCRxh matches state L	

unchanged when the condition is R/W

satisfied

01: In_opxl port bit outputs high

when the condition is met

10: In_opxl port bit outputs low

when the condition is met

11: In_opxl port bit output inverted

R/W

when condition is met

($x = u, v, w$)

Conditions: count overflow &

OCCRxl count mismatch &

OCCRhx count match ($x=u, v, w$)

00: In_opxl port bits remain

unchanged when the condition is R/W
met

01: In_opxl port bit outputs high

when the condition is met

10: In_opxl port bit outputs low

when the condition is met

11: In_opxl port bit output inverted

when condition is met

($x = u, v, w$)

Conditions: count underflow &

OCCRxl count match & OCCRhx

count match ($x=u, v, w$)

00: In_opxl port bits remain

unchanged when the condition is
satisfied

01: In_opxl port bit outputs high

when the condition is met

10: In_opxl port bit outputs low

when the condition is met

11: In_opxl port bit output inverted

when condition is met

($x = u, v, w$)

b25~b24

EOPUCL[1:0]

Extended Up Count

Condition: counter count up & OCCRxl

R/W

OCCRxl \overline{PC}

count match &

Match &OCCRhx Match

OCCRhx count match ($x=u, v, w$)

Status L

		OCCRxl mismatch & OCCRxh mismatch state L	00: In_opxl port bits remain unchanged when the condition is met 01: In_opxl port bit outputs high when the condition is met 10: In_opxl port bit outputs low when the condition is met 11: In_opxl port bit output inverted when condition is met (x = u, v, w)
b23~b22	EOPPKL[1:0]	Extended overflow point OCCRxl match & OCCRxh match state L	Conditions: count overflow & OCCRxl count match & OCCRxh count match (x=u, v, w)
b21~b20	EOPDCL[1:0]	Extended Down Count OCCRxl Match & OCCRxh Match Status L	00: In_opxl port bits remain unchanged when the condition is met 01: In_opxl port bit outputs high when the condition is met 10: In_opxl port bit outputs low when the condition is met 11: In_opxl port bit output inverted when condition is met (x = u, v, w)
b19~b18	EOPNUCL[1:0]	Extended Up Count OCCRxl Mismatch & OCCRxh Match Status L	00: In_opxl port bits remain unchanged when the condition is met 01: In_opxl port bit outputs high when the condition is met 10: In_opxl port bit outputs low when the condition is met 11: In_opxl port bit output inverted when condition is met (x = u, v, w)
b17~b16	EOPNDCL[1:0]	Extended Down Count OCCRxl Mismatch & OCCRxh Match Status L	00: In_opxl port bits remain unchanged when the condition is met 01: In_opxl port bit outputs high when the condition is met 10: In_opxl port bit outputs low when the condition is met 11: In_opxl port bit output inverted when condition is met (x = u, v, w)
b15~b14	OPNZRL[1:0]	Underflow point OCCRxl mismatch & OCCRxh mismatch state L	Condition: counter count down & OCCRxl count match &
b13~b12	OPNPKL[1:0]	Overflow point	

OCCR _h count match ($x=u, v, w$)	when condition is met ($x = u, v, w$)	
00: In_opxl port bits remain unchanged when the condition is satisfied	Conditions: count overflow & OCCR _l count mismatch & OCCR _h	
01: In_opxl port bit outputs high when the condition is met	count mismatch ($x=u, v, w$)	
10: In_opxl port bit outputs low when the condition is met	00: In_opxl port bits remain unchanged when the condition is met	R/W
11: In_opxl port bit output inverted when condition is met ($x = u, v, w$)	01: In_opxl port bit outputs high when the condition is met	
Condition: counter count up & OCCR _l count mismatch & OCCR _h count match ($x=u, v, w$)	condition is met	
00: In_opxl port bits remain unchanged when the condition is met	10: In_opxl port bit outputs low when the condition is met	R/W
01: In_opxl port bit outputs high when the condition is met	11: In_opxl port bit output inverted when condition is met	
10: In_opxl port bit outputs low when the condition is met	($x = u, v, w$)	
11: In_opxl port bit output inverted when condition is met ($x = u, v, w$)		
Condition: counter counts down & OCCR _l counts don't match & OCCR _h counts match ($x=u, v, w$)		
00: In_opxl port bits remain unchanged when the condition is met	R/W	
01: In_opxl port bit outputs high when the condition is met		
10: In_opxl port bit outputs low when the condition is met		
11: In_opxl port bit output inverted when condition is met ($x = u, v, w$)	R/W	
Conditions: count underflow & OCCR _l count mismatch & OCCR _h count mismatch ($x=u, v, w$)		
00: In_opxl port bits remain unchanged when the condition is met	R/W	
01: In_opxl port bit outputs high when the condition is met		
10: In_opxl port bit outputs low when the condition is met		
11: In_opxl port bit output inverted		

		&OCCR _h mismatch state L	mismatch ($x=u, v, w$)	
			00: In_opxl port bits remain unchanged when the condition is satisfied	
			01: In_opxl port bit outputs high when the condition is met	
			10: In_opxl port bit outputs low when the condition is met	
			11: In_opxl port bit output inverted when condition is met ($x = u, v, w$)	
b9~b8	OPUCL[1:0]	Up Count OCCR _{xl} Match & OCCR _h Mismatch Status L	Condition: counter counts up & OCCR _{xl} counts match & OCCR _h counts don't match ($x=u, v, w$)	R/W
b7~b6	OPPKL[1:0]	Overflow point OCCR _{xl} match & OCCR _h mismatch state L	00: In_opxl port bit remains unchanged when the condition is met	R/W
b5~b4	OPDCL[1:0]	Count down OCCR _{xl} match & OCCR _h mismatch status L	01: In_opxl port bit outputs high when the condition is met	R/W
b3	OCFZRL	Underflow point OCFL state remains unchanged	10: In_opxl port bit outputs low when the condition is met	
b2	OCFUCL	Upward counting of OCFL states L	11: In_opxl port bit output inverted when condition is met ($x = u, v, w$)	
			Condition: count underflow & OCCR _{xl} count match ($x=u, v, w$)	
			L0: OCSR.OCFL bit when the condition is satisfied	R/W
			1: OCSR.OCFL position bit when the condition is satisfied	
			Condition: counter counts up & OCCR _{xl} counts match ($x=u, v, w$)	

			Condition: count overflow & OCCRxl count match (x=u, v, w)	
b1	OCFPKL	Upper overflow point OCFL state remains unchanged	when the condition is satisfied	L0: OCSR.OCFL bit R/W
			1: OCSR.OCFL position bit when the condition is satisfied	
b0	OCFDCL	Counting down the OCFL state L	Condition: counter counts down & OCCRxl counts match (x=u, v, w)	
		R/W	0: OCSR.OCFL bits remain unchanged when the condition is met 1: OCSR.OCFL position bit when the condition is satisfied	

Attention:

- When reading data from this address area, it is not the value of the buffer register that is read, but the value of the OCML register.
- **In_opxl** can be determined by the count value of **OCCRxl** with the counter (standalone operation mode), or the count value of **OCCRhx** with the counter and **OCCRxl** with the counter (linked operation mode). Write the same 12-bit value to **bit[31:20]** and **bit[15:4]** of register **OCMRxl**, and write '**0000'b** **OCMRxl[19:16]**'.

In this case, the In_opxl output will be unaffected by OCCRxh and determined only by OCCRxl. This mode is called independent operation mode. Channel xh is determined by OCCRxh and channel xl is configured by OCCRxl. If the above conditions are not met

The independent operation mode of the link operation mode channel xl output is affected by both OCCRxh and OCCRxl.

(x = u, v, w)

19.5.9 Dedicated Comparison 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]															

classifierfor honorific people	marking	Bit Name Function Read/Write	Dedicated comparison reference value (comparison start mode) or delayed reference value (delayed start mode)	Note: R/W
b15~b0	SCCR[15:0]	Dedicated comparison reference value When reading data from this address area, what is read is not the buffer register's value, but rather the value of the SCCR register		

19.5.10 Dedicated control status 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]			

classifier for marking honorific people		celebrity		functionality										fill out or in (information on a form)
				0: No operation of EVT in case of underflow counting 1: When counting underflow: EVTMS=0 & SCCR compare match & SCMR set match, EVT start output EVTMS=1&OCCR compare match & SCMR set match when, EVT delay mode activate (a plan)										
b15	ZEN	Underflow point EVT enable		EVTMS=0&SCCR compare match & SCMR set match, EVT R/W start output EVTMS=1&OCCR compare match & SCMR set match when, EVT delay mode activate (a plan)										R/W
b14 enable	UEN	up count EVT		0: No operation of EVT when counting upwards 1: When counting up: EVTMS=0&SCCR compare match & SCMR set match, EVT R/W start output EVTMS=1&OCCR compare match & SCMR set match, EVT delay mode start										R/W
b13 EVT enable	PEN	overflow point		0: No EVT operation when count overflow occurs 1: When counting overflow: EVTMS=0&SCCR compare match & SCMR set match, EVT R/W start output EVTMS=1&OCCR compare match & SCMR set match, EVT delay mode start										R/W
b12 enable	DEN	down count EVT		0: No operation of EVT when counting down 1: When counting down: EVTMS=0&SCCR compare match & SCMR set match, EVT R/W start output EVTMS=1&OCCR compare match & SCMR set match, EVT delay mode start										R/W
b11~b10	Reserved	-Reserved		Read "0", write "0".										R/W
b9 Object Selection	EVTDSEVT Delay		0: In delayed start mode, OCCRxh is used as a delayed comparison match 1: In delayed start mode, OCCRxl acts as a delayed comparison match (x = u, v, w) Note: This bit is invalid when EVTMS=0										R/W	
b8 comparison result of CNTR and SCCR)	EVTMSEVT mode selection		0: Comparison start mode (triggered by 1: Delayed start mode (compare match event triggered by SCCR delay)										R/W	
b7~b6	Reserved	-Reserved		Read "0", write "0".										R/W
b5	LMC	Link to Cycle Interval Response Function		0: Cycle interval response function link is invalid, SCCR cache transfer set by BUFEN decide (to do something) 1: The cycle interval response function link is valid, and the cache transmission of SCCR must meet CVPR.PIC[3:0]=0000 (in case of count overflow) or CVPR.ZIC[3:0]=0000 (in case of count underflow) in addition to the BUFEN setting 000: EVT output of Special										Event 0 is valid 001: EVT output of Special 001: EVT output valid for

R/W			
b4~b2	EVTOS[2:0]	EVT output selection	010: EVT output valid for Special Evnet 2 011: EVT output valid for Special Evnet 3 100: EVT output valid for Special Evnet 4 101: EVT output valid for Special Evnet 5

Please do not set other values			
b1~b0	BUFEN[1:0]	SCCR & SCMR Cache Transfer	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 overflow occurs
			11: SCCR, SCMR cache register values are written to SCCR, SCMR on count underflow or overflow
			R/W

19.5.11 Specialized 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]							
classifier for honorific people	marking	celebrity								functionality fill out or in (information on a form)									
b15~b8	Reserved	-								Reads "1" and writes "1".									
b7	MPCE	Cycle interval response enable								0: Comparison of AMC with CVPR.PIC is disabled 1: Enable AMC to compare with CVPR.PIC									
b6	MZCE	The cycle interval response enables (usually used in the negative) have the possibility of								0: Comparison of AMC with CVPR.ZIC is disabled 1: Enable AMC to compare with CVPR.ZIC									
b5~b4	Reserved	-								Reads "0", writes "0".									
b3~b0	AMC[3:0]	Dedicated event output cycle interval value								This bit sets the cycle interval value when the dedicated event output function is active when AMC and CVPR.PIC or CVPR.ZIC are equal.									

Attention:

-When reading data from this address area, it is not the value of the buffer register that is read, but the value of the SCMR register.

19.5.12 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]										
classifier for honorific people	marking	celebrity								functionality								fill out or in (information on a form)						
b15~b8	Reserved	-								Reads "1" and writes "1".								R/W						
b7~b6	LVLS[1:0]	PWM output polarity control output of TIM4_<t>_OxH is inverted and the output of TIM4_<t>_OxL is not invert (upside-down, inside-out, back-to-front, white to black etc)								00: Neither TIM4_<t>_OxH nor TIM4_<t>_OxL outputs are inverted 01: Both TIM4_<t>_OxH and TIM4_<t>_OxL outputs inverted								10: The R/W						
b5~b4	PWMMD[1:0]	PWM Output Modes								00: Straight-through mode 01: Deadband timer mode 10: Deadband Timer Filter Mode 11: Setting Prohibition								R/W						
b3	Reserved	-0" for reading, "0" for writing. "0", writes "0".								R/W								Reads						
b2~b0	DIVCK[2:0]	counter clock division								This bit indicates the count clock division frequency of the filter counter and the deadband counter. 000: Count clock is PCLK1 001: Count clock is PCLK1/2 010: Count clock is PCLK1/4 011: Count clock is PCLK1/8 100: Count clock is PCLK1/16 101: Count clock is PCLK1/32 110: Count clock is PCLK1/64 111: Count clock is PCLK1/128								R/W		R/W				

19.5.13 PWM Filter 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/Write							
Filter Count Initial Value															
b15~b0 initial value	PFSR[15:0]	Filter	Note: When the PWM waveform output mode does not select the deadband timer filter mode, the 16-bit filtering The counter is used as a 16-bit reload counter, when the 16-bit filter counter can periodically generate interrupt outputs, this function has nothing to do with the PWM waveform generator function.												R/W

19.5.14 PWM Deadband Control Register (TMR4_PDARn)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PDA/BR [15:0]															
classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)											
b15~b0	PDA/BR [15:0]	Deadband Initial Value	Deadband Count Initial Value												

19.5.15 Reload 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

classifier for honorific people	marking	Bit Name Function Read/Write	
		0: No operation	
b15	RTSW	Heavy Load Counter Stop W heavy load counter W and clear RTIFW	1: Stop R/W
		Note: This bit is always 0 on readout	
b14	RTEW	Reload counter start W 0: Write 0 to invalidate, read out as 0 to indicate that the heavy load counter W has stopped 1: Write 1 to start heavy load counter W, read 1 to indicate that heavy load counter W is started	R/W
		0: No operation	
b13	RTICW	Clear Count Match Status RTIFW Flag Bit R/W	W1: Clear
		Note: This bit is always 0 on readout	
b12	RTIFW	Count Match Status W 0: No comparison match between the reload counter count value and PFSRw occurs 1: Comparison match between heavy load counter count value and PFSRw occurs	R and PFSRw occurs
		0: No operation	
b11	RTSV	Heavy load counter stop V V1: stops heavy load counter V and clears RTIFV	R/W
		Note: This bit is always 0 on readout	
b10	RTEV	Heavy load counter start V 0: Write 0 to invalidate, read out as 0 to indicate that the heavy load counter V has stopped 1: Write 1 to start heavy load counter V, read 1 to indicate heavy load counter V is started	R/W
		0: No operation	
b9	RTICV	Clear Count Match Status RTIFV Flag Bit R/W	V1: Clear
		Note: This bit is always 0 on readout	
b8	RTIFV	Count Match Status V 0: The heavy load counter count value has not been compared and matched with PFSRv. 1: Comparison match between the heavy load counter count value and PFSRv occurs	R and PFSRv occurs
		0: No operation	
b7	RTSU	Heavy Load Counter Stop U heavy load counter U and clear RTIFU	1: Stop R/W
		Note: This bit is always 0 on readout	
b6	RTEU	Reload counter start U 0: Write 0 to invalidate, read out as 0 to indicate that the heavy load counter U has stopped 1: Write 1 to start heavy load counter U, read 1 to indicate heavy load counter U is started	R/W
		0: No operation	
b5	RTICU	Clear Count Match Status RTIFU Flag Bit R/W	U1: Clear

Note: This bit is always 0 on readout

b4	RTIFU	Count Match Status U 0: no comparison match between the heavy load counter count value and PFSRu occurs 1: Comparison match between heavy load counter count value R and PFSRu occurs	
b3	Reserved	-0" for reading, "0" for writing. "0", writes "0". R/W	Reads
b2	RTIDW	Heavy load interrupt mask W reload function is active, the reload interrupt W output is active 1: When the reload function is active, the reload interrupt W output is invalidated	0: When the R/W
b1	RTIDV	Heavy load interrupt mask V Heavy load interrupt V output valid when heavy load function is active 1: Heavy load interrupt V output is invalid when the heavy load function is active R/W	0:
b0	RTIDU	Heavy load interrupt mask U Heavy load interrupt U output valid when heavy load function is active 1: Heavy load interrupt U output is invalid when the heavy load function is active R/W	0:

19.5.16 EMB Control Status Register (TMR4_ECSR)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
Reserved										HOLD	Reserved					
classifier for honorific people	marking	celebrity		functionality				fill out or in (information on a form)								
b15~b8	Reserved	-		Reads "0", writes "0".										R/W		
b7	HOLD	PWM Hold		0: Normal PWM output when EMB input event is detected 1: When an EMB input event is detected, the current PWM output state is maintained without further change										R/W		
b6~b0	Reserved	-Reserved "0".		R/W										Read "0", write "0".		

19.5.17 EMB Extended Control Register (TMR4_ECER)

Reset value: 0x0000

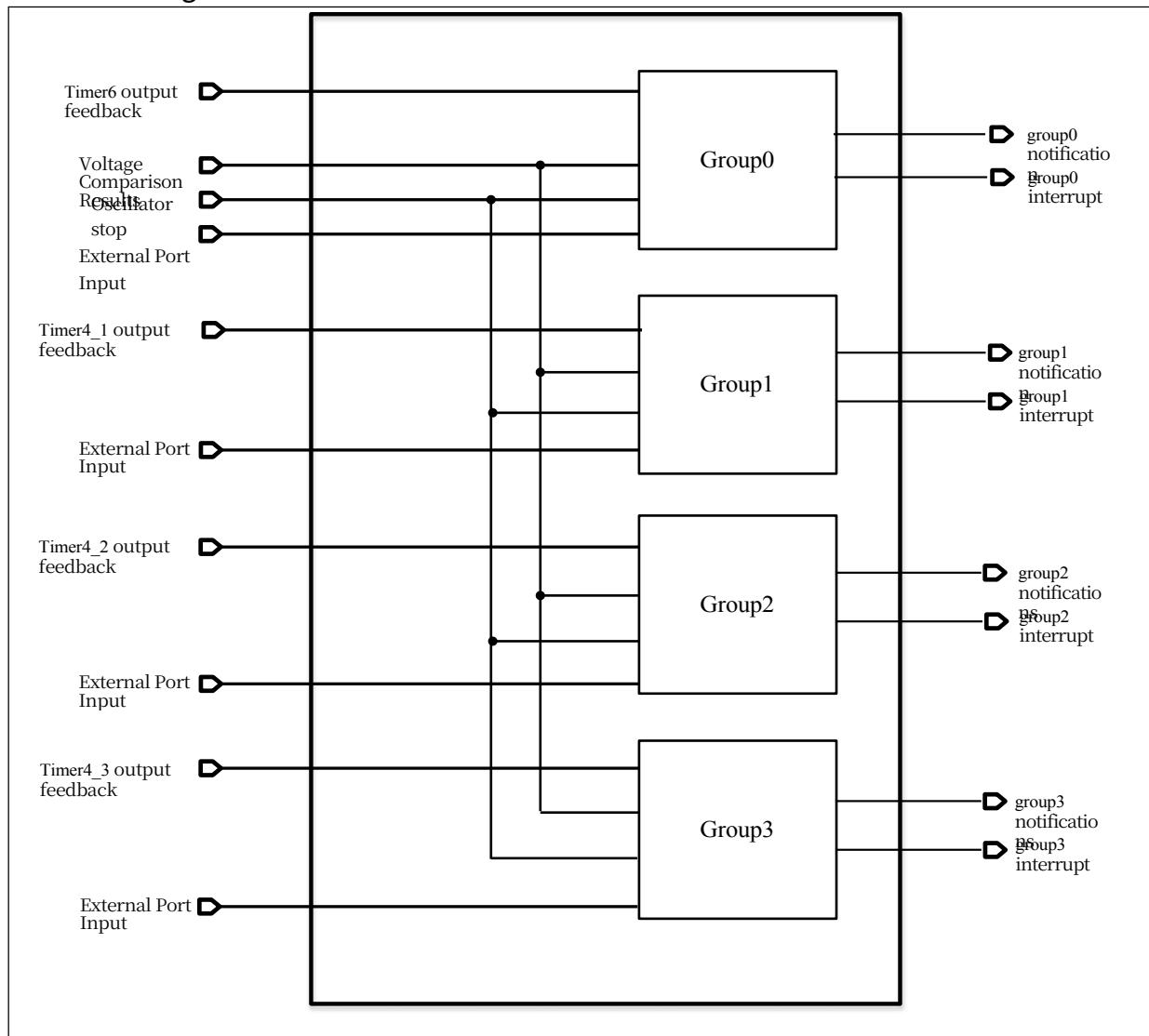
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
Reserved														EMBVAL[1:0]		
classifier for honorific people	marking	celebrity		functionality										fill out or in (information on a form)		
b15~b2	Reserved	-		Reads "0", writes "0".										R/W		
b1~b0 Control	EMBVAL[1:0]	EMB Status		00: When an EMB event occurs, the PWM port state is determined by the ECSR.HOLD bit setting 01: When an EMB event occurs, the PWM port output becomes Hiz 10: PWM port output is fixed low when an EMB event occurs 11: PWM port output fixed high when EMB event occurs										R/W		

20 Emergency Brake Module (EMB)

20.1 summary

The Emergency Brake Module is a functional module that notifies the timer when certain conditions are met so that the timer stops outputting PWM signals to the external motor, and the following events are used to generate the notification:

- External port input level change
- PWM output port levels are in-phase (same high or low)
- Voltage Comparator Comparison Results
- External oscillator stops oscillating
- Write Register Software Control



20.2 Functional Description

20.2.1 summarize

EMB is used to output a notification signal to the timer modules (Timer4, Timer6) with PWM function to notify the timer module to turn off the current PWM output when a certain condition is met. EMB module has 4 groups, among which group 0 corresponds to Timer6, which is set by Timer6 register to select and use, and group 1~3 corresponds to the 3 units of Timer4. The EMB module has 4 groups, among which group 0 corresponds to Timer6, which is selected by Timer6 register, and group 1~3 correspond to the 3 units of Timer4.

20.2.2 Stop PWM signal output when external port input level changes.

Each cluster of the EMB is assigned a port to realize stopping the PWM signal output when the input level of the external port changes. The port assignments are shown in the table below.

functional name	functionality	Corresponding Timer Module
EMB_IN1	group 0 port input control signal	Timer6
EMB_IN2	group 1 port input control signal	Timer4_1
EMB_IN3	group 2 port input control signal	Timer4_2
EMB_IN4	group 3 port input control signal	Timer4_3

The EMB can send a notification signal when the port level is high (INVSEL=0) or when the port level is low (INVSEL=1) according to the setting of control register EMB_CTLx ($x=0\sim3$). At the same time, each port is equipped with a digital filtering function, and the filtering intensity can be set as required. When a qualified level change occurs on the port, the EMB will generate a notification signal to notify Timer6 and Timer4. Timer6 and Timer4 can set the output port to high, low or high resistance according to the register setting after receiving the notification. At the same time, the EMB can also generate interrupt requests. When the port level becomes invalid, user can clear the notification signal by writing EMB Status Reset Register (EMB_STATCLR x) ($x=0\sim3$), so that Timer6 and Timer4 can resume outputting PWM waveforms.

20.2.3 Stop PWM signal output when the PWM output port level is in phase (same high or same low).

The EMB is able to monitor the complementary PWM output signals of Timer6 and Timer4, and generate notification signals to Timer6 and Timer4 when the output signals are in the same high or low condition, and Timer6 and Timer4 can set the output ports to be high, low, or high resistive according to the register settings after the notification. The EMB can also generate interrupt requests. Group0 can be used to monitor the complementary PWM output signal of Timer6, and group1-3 can be used to monitor the complementary PWM output signal of **Timer4_1/ Timer4_2/ Timer4_3**. When the same-high or same-low condition of the port is released, i.e., the PWMSF bit of EMB Status Register (**EMB_STATx**) ($x=0\sim 3$) is reset, the user can clear the notification signal by writing to EMB Status Reset Register (**EMB_STATCLRx**) ($x=0\sim 3$), which will enable Timer6 and Timer4 to resume outputting PWM waveforms.

port name	functionality	corresponding group	EMB_CTLx control bits	EMB_PWMLVx control bits
TIM6Am (m=1~3)	Complementary PWM input for Timer6 send a signal	group0	PWMSEN[2:0]	PWMLV[2:0]
TIM6Bm (m=1~3)				
TIM4OUH_1	Complementary PWM output signal of Timer4_1	group1	PWMSEN [2]	PWMLV [2]
TIM4OUL_1			PWMSEN[1]	PWMLV[1]
TIM4OVH_1			PWMSEN[0]	PWMLV[0]
TIM4OVL_1				
TIM4OWH_1				
TIM4OWL_1				
TIM4OUH_2	Complementary PWM output signal of Timer4_2	group2	PWMSEN [2]	PWMLV [2]
TIM4OUL_2			PWMSEN[1]	PWMLV[1]
TIM4OVH_2			PWMSEN[0]	PWMLV[0]
TIM4OVL_2				
TIM4OWH_2				
TIM4OWL_2				
TIM4OUH_3	Complementary PWM output signal of Timer4_3	group3	PWMSEN [2]	PWMLV [2]
TIM4OUL_3			PWMSEN[1]	PWMLV[1]
TIM4OVH_3			PWMSEN[0]	PWMLV[0]
TIM4OVL_3				
TIM4OWH_3				
TIM4OWL_3				

20.2.4 Stop PWM signal output based on the comparison result of the voltage comparator.

Each cluster of EMB can send notification signal to Timer6 and Timer4 according to the comparison result of voltage comparator, please refer to Voltage Comparator section for the setting of voltage comparator output result. The EMB will generate a notification signal to Timer6 and Timer4 when the VOLTAGE COMPARATOR RESULT flag is raised, and Timer6 and Timer4 can set their output ports to high, low, or high-resistance according to the register settings after receiving the notification. The EMB can also generate interrupt requests. After the voltage comparator flag bit is reset, user can clear the notification signal by writing EMB status reset register (EMB_STATCLR x) ($x=0\sim 3$), so that Timer6 and Timer4 can resume outputting PWM wave.

20.2.5 Stop PWM signal output when the external oscillator stops oscillating.

Each group of EMB can send a notification signal to Timer6 and Timer4 when the external oscillator stops oscillating, please refer to the Voltage Comparator section for the setting of external oscillator stop oscillating. When the external oscillator stop flag is raised, the EMB will generate a notification signal to Timer6 and Timer4, and Timer6 and Timer4 can set their output ports to high, low, or high resistance according to the register settings after receiving the notification. At the same time, EMB can generate interrupt request. When the external oscillator stop oscillation flag bit is reset, user can clear the notification signal by writing EMB status reset register (EMB_STATCLR x) ($x=0\sim 3$), so that Timer6 and Timer4 can resume outputting PWM wave.

20.2.6 Write Register Software Control PWM Signal Outputs

The EMB's Software Output Enable Control Register (EMB_SOEx) ($x=0\sim 3$) can allow the user to send notification signals to Timer6 and Timer4 through software direct set and reset, and no interrupt request will be generated when software controls the PWM output.

20.3 Register Description

List of registers

name (of a thing)	Abbreviations	clarification	offset address
EMB Output Control Register	EMB_CTL	Controls the conditions under which each output is enabled	0x0
EMB Feedback Level Selection Register	EMB_PWMLV	Selects the active level of the PWM feedback signal	0x4
EMB Software Output Enable Control Registers	EMB_SOE	Software generates an output prohibition event	0x8
EMB Status Register	EMB_STAT	Indicates the state of output enable	0xC
EMB Status Reset Register	EMB_STATCLR	Clear output enable state	0x10
EMB Interrupt License Register	EMB_INTEN	Control Interrupt Enable	0x14

20.3.1 EMB Control Register 0 (EMB_CTL0)

This register is a single write register, i.e., it can be written only once after reset.

Address:

0x4001_7C00 Reset

value: 0x0000_0000

b31 SEL	b30 N	b29~b28 NFSEL[1:0]	Reserved												
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved					PWM SEN 2	PWM SEN 1	PWM SEN 0	OS CST PEN	Re s. EN3	CMP EN3	CMP EN2	CMP EN1	PORT INEN		

classifier for honorific people	marking	Bit Name	Function	Read/Write
b31	INVSEL	Port Input Valid Level Selection		0: active high level 1: Active low
b30	NFEN	Port Input Digital Filter Enable		0: Filter disabled 1: Filter effective
b29~b28	NFSEL[1:0]	Digital Filter Filter Clock Selection		00: Using Bus Clock Filtering 01: 8-division filtering using the bus clock 10: 32-division filtering using bus clocks 11: 128-division filtering using the bus clock
b27~b9	Reserved	Read 0 when reading, please write 0 when writing		
b8	PWMSEN2	TIM6A/B_3 Short circuit output control enable control disabled for short circuit		
b7	PWMSEN1	TIM6A/B_2 short circuit output control enable control disabled for short circuit		
b6	PWMSEN0	TIM6A/B_1 short circuit output control enable control disabled for short circuit		
b5	OSCSTPEN	Oscillator stop output control enable oscillating.		
b4	Reserved	Read 0 when reading, please write 0 when writing		
b3	CMPEN3	CMP3 Voltage Comparator Comparison Result Control Enable comparator compare result output control disabled		
b2	CMPEN2	CMP2 Voltage comparator comparison result control enable comparator compare result output control disabled		

b1	CMPEN1	CMP1 Voltage Comparator Comparison Result Control Enable comparator compare result output control disabled	0: Voltage R/W 1: Voltage comparator comparison result output control active
b0	PORINEN	Port Input Control Enable	0: Port input control disabled R/W 1: Port input control active

20.3.2 EMB Control Register 1~3 (EMB_CTL1~3)

This register is a single write register, i.e., it can be written only once after reset.

Address: 0x4001_7C20, 0x4001_7C40, 0x4001_7C60

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16		
INVSEL	NFEN	NFSEL[1:0]		Reserved													
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0		
F served								PWM SEN2	PWM SEN1	PWM SEN0	OSCST PEN	Res.	CMP EN3	CMP EN2	CMP EN1	PORTI NEN	
classifier for honorifi c people	marking	celebrity		functionality		fill out or in (information on a form)											
b31	INVSEL	Port Input Valid Level Selection				0: active high level 1: Active low									R/W		
b30	NFEN	Port Input Digital Filter Enable				0: Filter disabled 1: Filter effective									R/W		
b29~b28 Selection	NFSEL[1:0]	Digital Filter Filter Clock Selection				00: Using Bus Clock Filtering 01: 8-division filtering using bus clocks 10: 32-division filtering using bus clocks 11: 128-division filtering using the bus clock									R/W		
b27~b9	Reserved reads				0 when reading, please write 0				when writing								
R																	
b8 for TIM4_x (x=1~3)	PWMSEN2	U-phase short-circuit input				0: Invalid output control in case of short circuit									R/W		
b7 for TIM4_x (x=1~3)	PWMSEN1	Out Control Enable				V-phase short-circuit input									R/W		
b6	PWMSEN0	Out Control Enable				TIM4_x (x=1~3) W-phase short-circuit input				0: 1: Output control valid at short circuit					0: 1: Output control valid at short circuit		
b5	OSCSTPEN	Output control disabled for short circuit				R/W				0: 1: Output control valid at short circuit					0: 1: Output control valid at short circuit		
b4	Oscillator stop output control enable				R/W				0: 1: Disable output control when the oscillator stops oscillating.								
b3	CMPEN3	Output control enable				R/W				0: 1: Output control valid at short circuit							
b2	CMPEN2	Output control enable				R/W				0: 1: Output control valid at short circuit							
b1	CMPEN1	Output control enable				R/W				0: 1: Output control valid at short circuit							
HC32F460_F45x_A460 Series Reference																	
Manual_Rev1.6	PORTINEN	Port Input Control Enable				R/W				0: 1: Port input control disabled							
b0										R/W 1: Port input control active							

20.3.3 EMB Feedback level selection register 0 (EMB_PWMLV0)

This register is a single write register, i.e., it can be written only once after reset.

Address:

0x4001_7C04 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															
PWMLV2	PWMLV1	PWMLV0													

classifier for honorific people	marking	celebrity	functionality	fill out or in (informa tion on a form)
b31~b3	Reserved		Read 0 when reading, please write 0 when writing	R
b2	PWMLV2	TIM6A/B_3 output valid level selection level is active level	R/W 1: High level is active level	0: Low
b1	PWMLV1	TIM6A/B_2 output valid level selection level is active level	R/W 1: High level is active level	0: Low
b0	PWMLV0	TIM6A/B_1 output valid level selection level is active level	R/W 1: High level is active level	0: Low

20.3.4 EMB Feedback Level Selection Registers 1~3 (EMB_PWMLV1~3)

This register is a single write register, i.e., it can be written only once after reset.

Address: 0x4001_7C24, 0x4001_7C44, 0x4001_7C64

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															
classifier for marking celebrity functionality honorific people															
fill out or in (information on a form)															
b31~b3	Reserved				Read 0 when reading, please write 0 when writing										
b2	PWMLV2				U-phase output active level selection for TIM4_x (x=1~3) 0: Low level is active 1: High level is active										
b1	PWMLV1				V phase output active level selection for TIM4_x (x=1~3) 0: Low level is active 1: High level is active										
b0	PWMLV0				W phase output active level selection for TIM4_x (x=1~3) 0: Low level is active 1: High level is active										
R/W R/W R/W R/W															

20.3.5 EMB Software Output Enable Control Register (EMB_SOEx) (x=0~3)

Address: 0x4001_7C08, 0x4001_7C28, 0x4001_7C48, 0x4001_7C68

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															
classifier for marking celebrity functionality honorific people															
fill out or in (information on a form)															
b31~b1	Reserved				Read 0 when reading, please write 0 when writing										
b0	SOE				0: PWM normal output 1: PWM stop output										
R R/W															

20.3.6 EMB Status Register (EMB_STATx) (x=0~3)

Address: 0x4001_7C0C, 0x4001_7C2C, 0x4001_7C4C, 0x4001_7C6C

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															

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b6	Reserved		Read 0 when reading, please write 0 when writing	R
b5	PWMST	PWM output status	0: No PWM output in-phase occurred 1: PWM output in-phase occurs	R
b4	PORTINST	Port Input Control Status	0: Port input control is inactive 1: Port input control is active	R
b3	OSF	State in which the oscillator stops oscillating to stop PWM outputs	0: PWM output is not currently stopped due to the oscillator stopping oscillation 1: Currently stop PWM output due to oscillator stop oscillation	R
b2	CMPF	State in which the voltage comparator stops the PWM outputs	0: PWM output is not currently stopped due to voltage comparator comparison result 1: PWM output currently stopped due to voltage comparator comparison result	R
b1	PWMSF	PWM output in-phase stop PWM output state	0: No PWM output is currently stopped due to PWM output feedback being in-phase 1: Currently stop PWM output due to in-phase PWM output feedback	R
b0	PORTINF	Port Input Control Stop PWM Output Status	0: No PWM output currently stopped due to port input control 1: PWM output currently stopped due to port input control	R

20.3.7 EMB Status Reset Register (EMB_STATCLR x) ($x=0\sim3$)

Address: 0x4001_7C10, 0x4001_7C30, 0x4001_7C50, 0x4001_7C70

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															

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b4	Reserved		Read 0 when reading, please write 0 when writing	R
b3	OSFCLR	Reset EMB_STAT.OSF	0: No effect 1: When CMU.MOSCSTP=0, clears the EMB_STAT.OSF bit and restores the PWM output that was stopped due to the oscillating	W
b2	CMPFCLR	Reset EMB_STAT.CMPF	0: No effect 1: When CMPMONn. OMON=0, clears the EMB_STAT. CMPF bit and restores	
b1	PWMSFCLR	Reset EMB_STAT.PWMSF	he PWM output that was stopped by Was a result of the voltage comparator comparison result	t
b0	PORTINFCLR	Reset EMB_STAT.PORTINF	0: No effect 1: When EMB_STAT.PWMST = 0, clear the EMB_STAT. PWMSF bit and restore the Resume PWM outputs stopped due to PWM	W
			output feedback being in phase 0: no effect 1: When EMB_STAT.PORTINST=0, clear EMB_STAT. PORTINF bit and restores PWM output stopped by port input control	W

20.3.8 EMB interrupt license register (EMB_INTENx) (x=0~3)

Address: 0x4001_7C14, 0x4001_7C34, 0x4001_7C54, 0x4001_7C74

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															

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b4	Reserved		Read 0 when reading, please write 0 when writing	R
b3	OSINTEN	Oscillator stop Oscillation stop PWM interrupt enable	0: no interrupt is generated when the oscillator stops oscillating and stops PWM 1: Interrupt generated when the oscillator stops oscillating and stops PWM	R/W
b2	CMPINTEN	Voltage comparator stop PWM interrupt enable	0: No interrupt is generated when the comparison result of the voltage comparator stops PWM 1: Interrupt generated when PWM is stopped by the comparison result of voltage comparator	R/W
b1	PWMSINTEN	PWM output in- phase stop PWM interrupt enable	0: No interrupt is generated when the PWM output stops PWM in the same phase 1: Interrupt generated when PWM output stops PWM in the same phase	R/W
b0	PORINTINTEN	Port Input Control Stop Interrupt Enable for PWMs	0: Port input control stops PWM without generating an interrupt 1: Interrupt generated when port input control stops PWM	R/W

21 General-purpose timer (TimerA)

21.1 summary

The general-purpose Timer A (TimerA) is a 16-bit count width timer with 8 PWM outputs. The timer supports triangle and sawtooth waveform modes to generate various PWM waveforms, synchronized startup of counters, caching of comparison reference value registers, 32-bit counting by cascading between units, 2-phase orthogonal coded counting and 3-phase orthogonal coded counting. This series of products is equipped with 6 units of TimerA, which can realize 48 PWM outputs at most.

21.2 basic block diagram

The basic functions and features of TimerA are shown in Table 21-1.

Table 21-1 Basic Functions and Characteristics of TimerA

Waveform Mode	Sawtooth wave, triangle wave
Basic Functions	- Incremental and decremental counting direction
	- Synchronized Startup Counter
	- Reference value caching function
	-32-bit cascade counting
	- Orthogonal code counting
	- 8 PWM outputs
	- Compare Match Event Output
Interrupt Type	- Compare Match Interrupt
	- Cycle Matching Interrupt

The basic block diagram of TimerA is shown in Figure 21-1. In the diagram, "<t>" is the unit number, i.e., "<t>" is 1~6, and all references to "<t>" later in this chapter refer to the unit number. unit number, will not repeat.

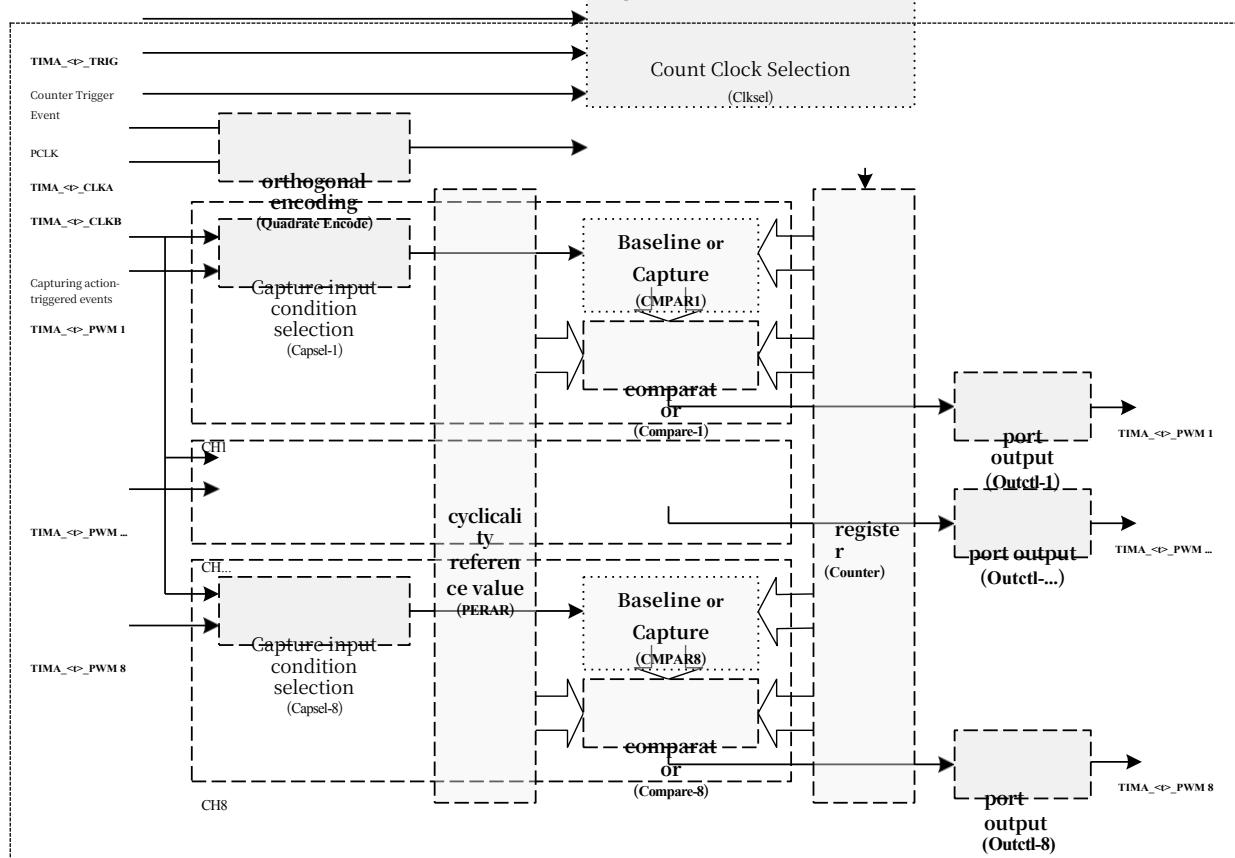


Figure 21-1 TimerA Basic Block Diagram

Table 21-2 shows the list of input and output ports of TimerA.

Table 21-2 TimerA Port List

port name	orientations	functionality
TIMA_<t>_PWMr	in or out	Capture input event port or PWM output port (m=1~8)
TIMA_<t>_CLKA	in	Orthogonal coded counting event input port
TIMA_<t>_CLKB	in	
TIMA_<t>_TRIG	in	Hardware-triggered start, stop, and clear event input port

21.3 Functional Description

21.3.1 basic movement

21.3.1.1 Waveform Mode

TimerA has two basic counting waveform modes, sawtooth waveform mode and triangle waveform mode. The basic waveforms of the two waveform modes are shown in Figure 21-2 and Figure 21-3.

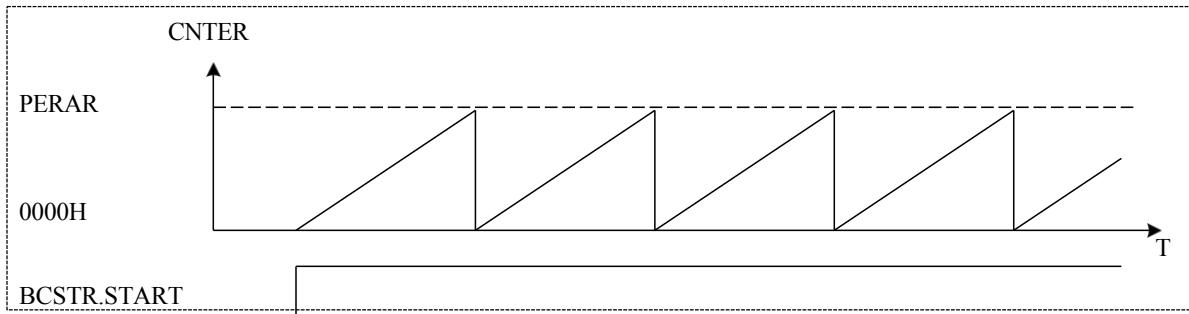


Figure 21-2 Sawtooth Waveform (Incremental Counting)

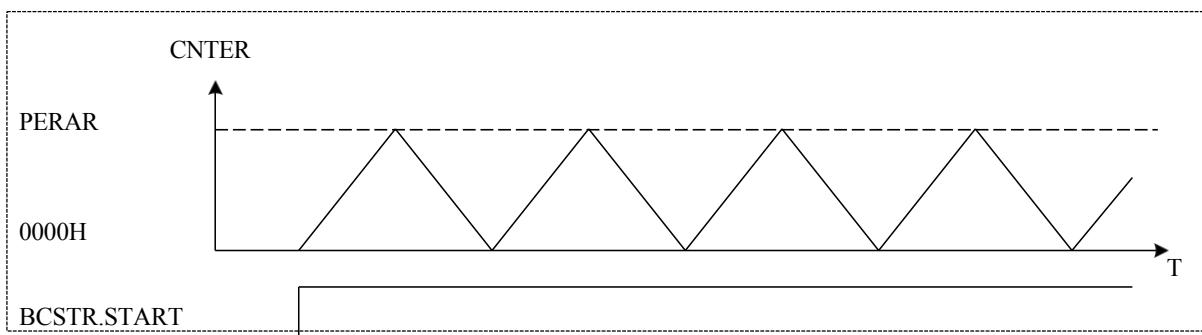
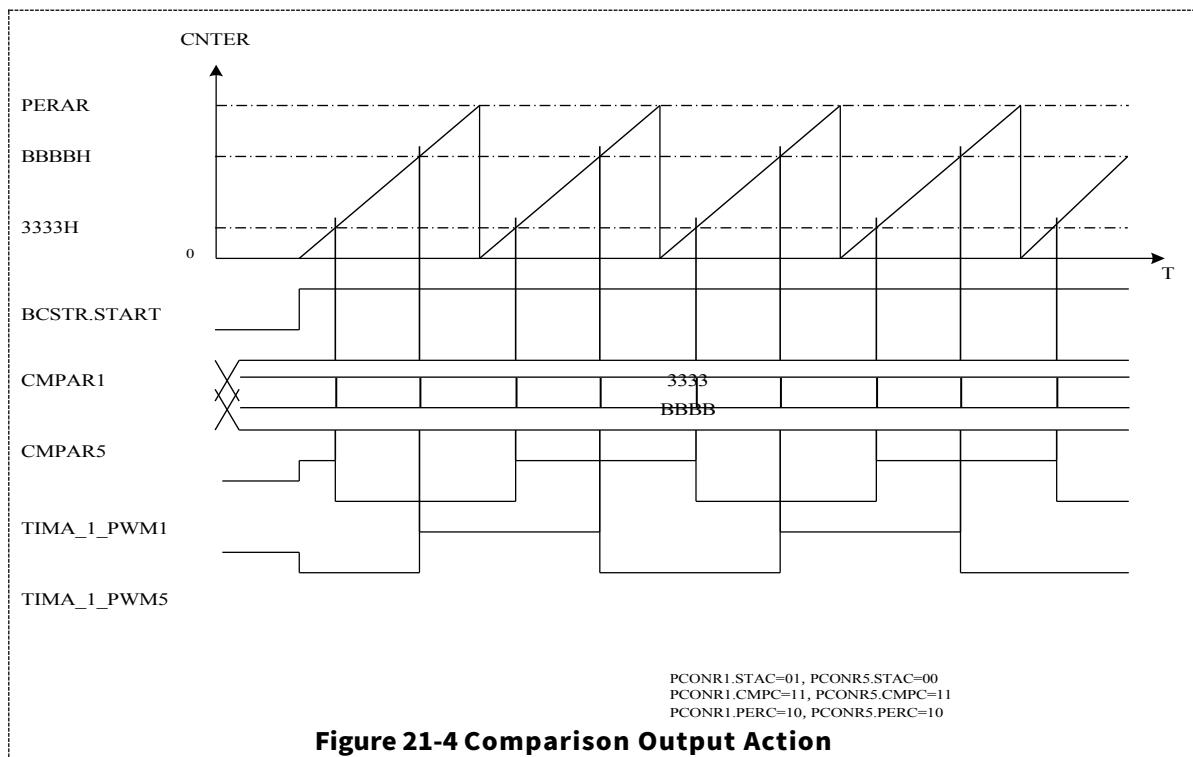


Figure 21-3 Triangle Waveform

21.3.1.2 Comparison Output

Each TimerA unit contains an 8-channel comparison output (TIMA_<t>_PWMn) that outputs the specified level when the count value matches the comparison reference value. The TMRA_CMPARn register corresponds to the count comparison reference value of the TIMA_<t>_PWMn output port, respectively. The TIMA_<t>_PWMn output port outputs the specified level when the timer count value and TMRA_CMPARn are equal.

(n=1~8)



The level at which counting starts, the level at which counting stops, the level at which counting comparison matches, and the level at which counting period matches for the TIMA_<t>_PWMn port can be set and controlled by the STAC, STPC, CMPC, PERC, and FORC bits of the port control register (PCONRn) (n=1~8). Fig. 21-4 shows an example of the comparison output action of Unit 1.

21.3.1.3 Capture Input

Each PWM output channel of each TimerA Unit has a capture input function to save the captured count value. Setting the CCONR.CAPMD bit of the capture control register (CCONRn) to 1 enables the capture input function. When the corresponding capture input condition is selected and the condition is valid, the current count value will be saved in the corresponding register (CMPARn) (n=1~8)

The capture input condition can be selected from internal capture action trigger event (selected via HTSSR1 register)TIMA_<t>_PWMn port input, etc. The specific condition selection can be set via the HICP bit of the capture control register (CCONRn) (n=1~8) Figure 21-5 shows an example of the capture input action for Unit 2.

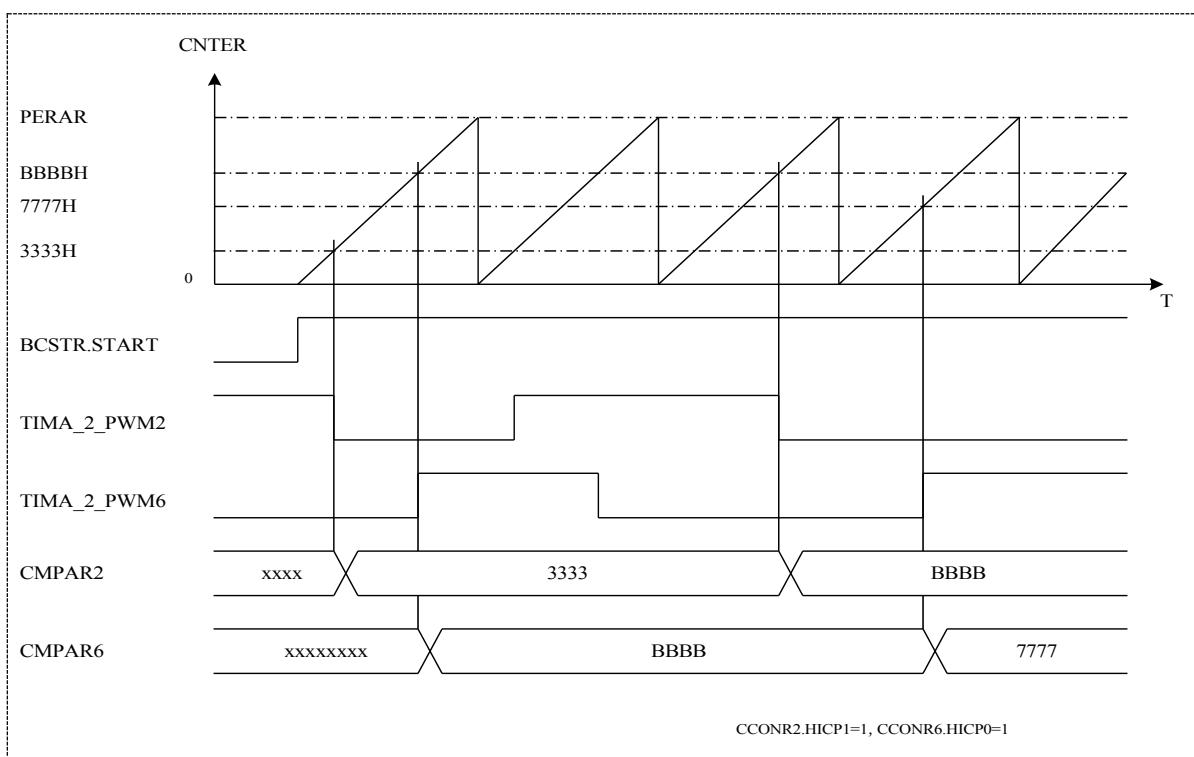


Figure 21-5 Capture Input Action

21.3.2 Clock source selection

The following options are available for the count clock of TimerA:

- a) 12481632641282565121024 divisions of PCLK1 (BCSTR.CKDIV[3:0])
(Settings)
- b) TIMA_<t>_TRIG Port event input (HCUPR[9:8] or HCDOR[9:8] setting)
- c) Internal counter trigger event input (HCUPR[10] or HCDOR[10] setting)
- d) Count overflow or count underflow event input for symmetrical units (HCUPR[12:11]
or HCDOR[12:11] setting)
- e) Port quadrature code input for TIMA_<t>_CLKA, TIMA_<t>_CLKB
(HCUPR[7:0] or HCDOR[7:0] setting)

Software counting mode is available when a is selected for the counting clock source, and hardware counting mode is available when b, c, d, or e is selected for the counting clock source. The counting clock selection d is mostly used for three-phase orthogonal coded counting in the revolution counting mode (see sections [Position Overflow Counting] and [Mixed Counting]) and can also be used for cascade counting. As can be seen from the above description, the b, c, d, and e clocks are independent of each other and can be set to be active or inactive, and the a clock is automatically inactive when the b, c, d, and e clocks are selected.

21.3.3 synchronized startup

The 6 units of TimerA can be synchronized with either software or hardware. Unit 2 to Unit 6 can be selected to synchronize with Unit 1. When the BCSTR.SYNST bit in Unit 2 to Unit 6 is set to 1, the synchronized startup function between the corresponding Unit and Unit 1 is effective. At this time, if the BCSTR.START bit of Unit 1 is set to 1 by the software, the counter of the synchronized unit (Unit 2~Unit 6) starts software synchronized counting; if any bit of HCONR.HSTA1~0 of Unit 1 is set to 1 by the hardware, and the corresponding hardware event of Unit 1 occurs, the counter of the synchronized unit (Unit 2~Unit 6) starts hardware synchronized counting. When the hardware synchronous counting start function is selected, the corresponding bits of HCONR.HSTA1~0 of the synchronized units (Unit 2~Unit 6) must also be set to valid.

Figure 21-6 shows an example of software synchronization startup when BCSTR.SYNST=1 is set for Unit 2, Unit 4, and Unit 5.

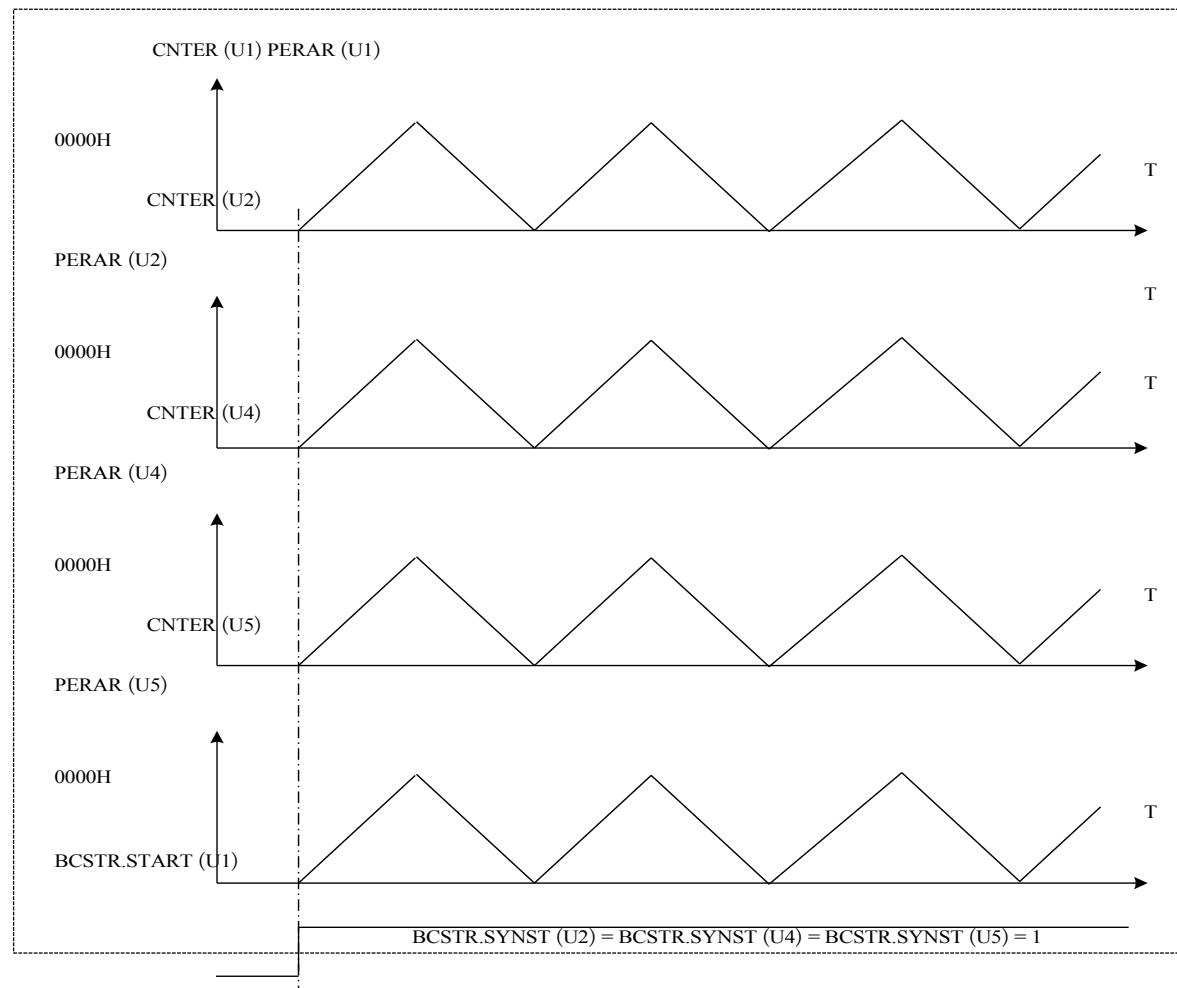


Figure 21-6 Software Synchronization Action

21.3.4 digital filtration

The TIMA_<t>_CLKA, TIMA_<t>_CLKB, TIMA_<t>_TRIG, and TIMA_<t>_PWMn (when the capture input function is active) port inputs of each Unit have digital filtering functions. The enabling of the filter function and the selection of the filter clock for each port can be realized by setting the corresponding bits of the filter control register (FCONR) and the capture control register (CCONRn) (n=1~8)

If the filter sampling reference clock samples a level on the port that is consistent three times, the level is treated as a valid level and transmitted to the module; levels that are less than three times consistent are filtered out as external interference and are not transmitted to the module. Figure 21-7 shows an example of the CLK_A port filtering operation of Unit 1.

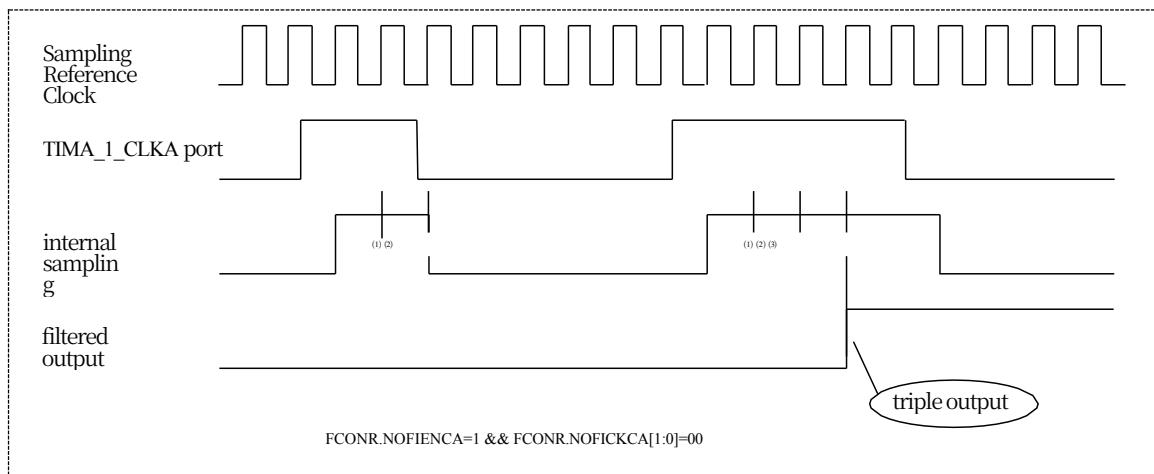


Figure 21-7 Clock Input Port Filter Function

21.3.5 caching function

A total of 8 comparison reference registers (CMPARn) of TimerA can realize the cache function in pairs ($n=1\sim 8$) That is, CMPAR2 is used as the cache reference value of CMPAR1, CMPAR4 is used as the cache reference value of CMPAR3, CMPAR6 is used as the cache reference value of CMPAR5, and CMPAR8 is used as the cache reference value of CMPAR7. The cache control register (BCONRm) realizes the control of each of the four groups of cache functions ($m=1\sim 4$)

The cache function becomes active when the BEN bit of the cache control register (BCONRm) is set ($m=1\sim 4$) A cache transfer occurs when the counter counts up to a specific time point (CMPAR8/6/4/2->CMPAR7/5/3/1) The "specific point in time" can be as follows:

- a) Hardware counting mode counts up to the overflow point (when BCSTR.DIR=1) or down to the underflow point (when BCSTR.DIR=0)
- b) In sawtooth wave counting mode (BCSTR.MODE=0), the counter counts to the upflow point (when BCSTR.DIR=1) or downflow point (when BCSTR.DIR=0)
- c) Triangular wave counting mode (BCSTR.MODE=1) when counting to peak (BCSTR.

DI

R=1&&BCONRn.BSE0=1{n=1~4}

- d) Triangular wave counting mode (BCSTR.MODE=1) when counting to valley point (BCSTR.

DI

R=0&&BCONRn.BSE1=1{n=1~4}

- e) Zero clearing occurs in hardware counting mode or sawtooth wave counting mode Figure 21-8 below shows the cache transfer schematic for Unit 2 sawtooth wave mode.

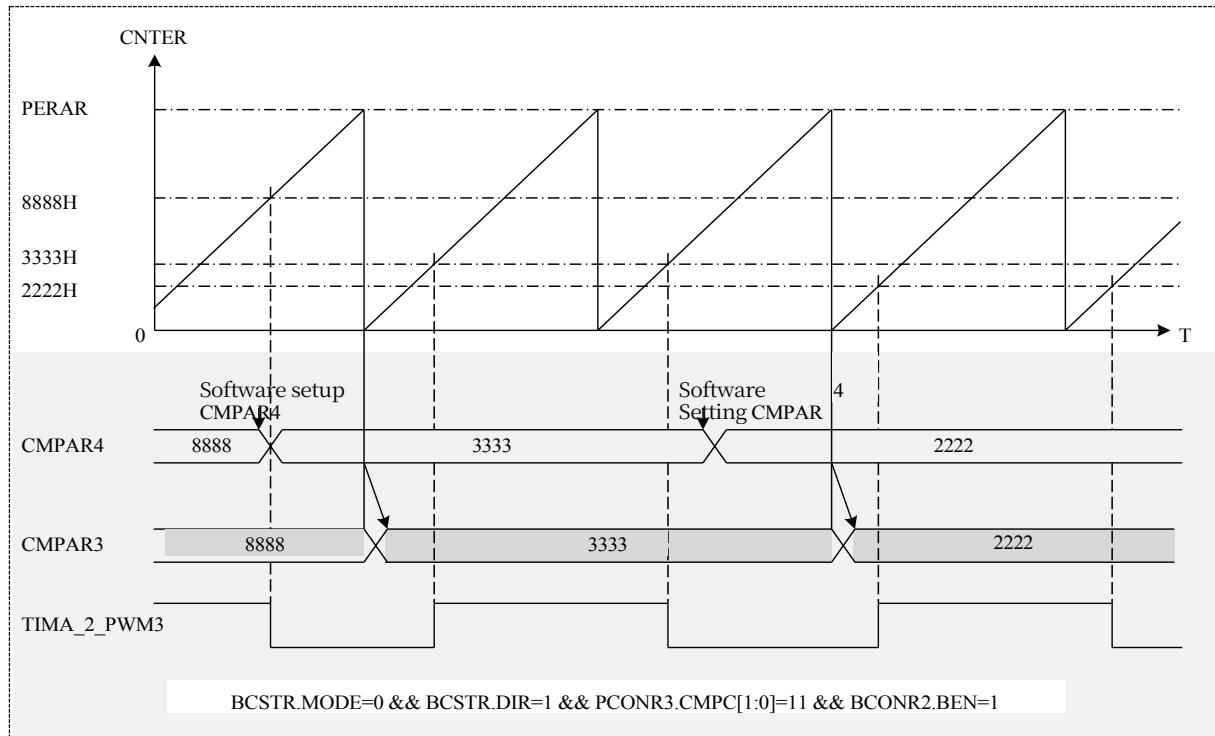


Figure 21-8 Cache Action in Sawtooth Wave Mode

21.3.6 cascade count

In the Count Clock Source Selection section, when d) is selected as the clock source, the count clock of this unit is selected as the count overflow (count up or count down) event of the symmetry unit, and then the two units are cascaded and can realize 32-bit counters. In cascade counting, the CNTER of the symmetry unit is the low 16-bit counter and the CNTER of the unit is the high 16-bit counter.

For example, in the delta wave count up mode (BCSTR.MODE=0, BCSTR.DIR=1), set the count clock of Unit 1 to PCLK1 and set the count clock source of Unit 2 to the count overflow event of Unit 1 (TMRA_HCUPR_HCUP11=1 of Unit 2) and then start the counting of Units 1 and 2 (start Unit 2 first, then start Unit 1). Cascade counting is realized. The CNTER of Unit 1 is a low 16-bit counter and the CNTER of Unit 2 is a high 16-bit counter. Figure 21-9 shows the cascade counting diagram of Unit 1 and Unit 2.

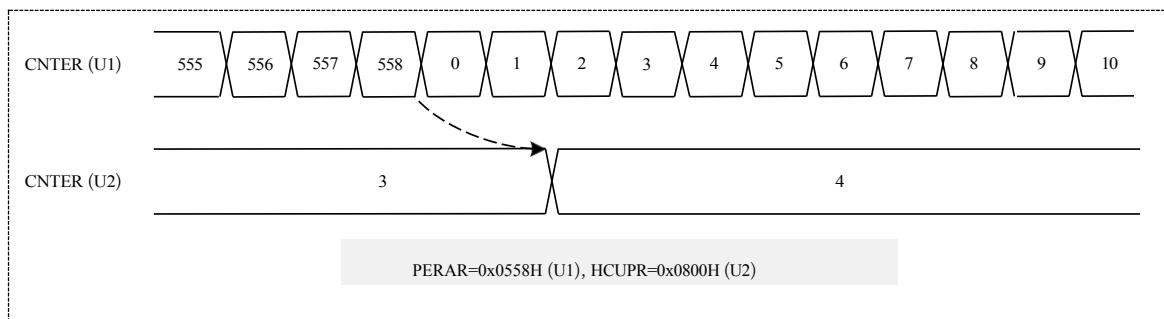


Figure 21-9 32-Bit Cascade Count Action

21.3.7 PWM Output

21.3.7.1 General Purpose PWM Output

Each of the 8 internal TimerA outputs, TIMA_<t>_PWMn, can be realized with different output waveforms ($n=1\sim 8$) through the relevant control bits in the Port Control Register (PCONRn).

Figure 21-10 shows an example of the PWM output waveforms for channels 1, 2, 6, and 7 in the triangular wave mode of Unit 1.

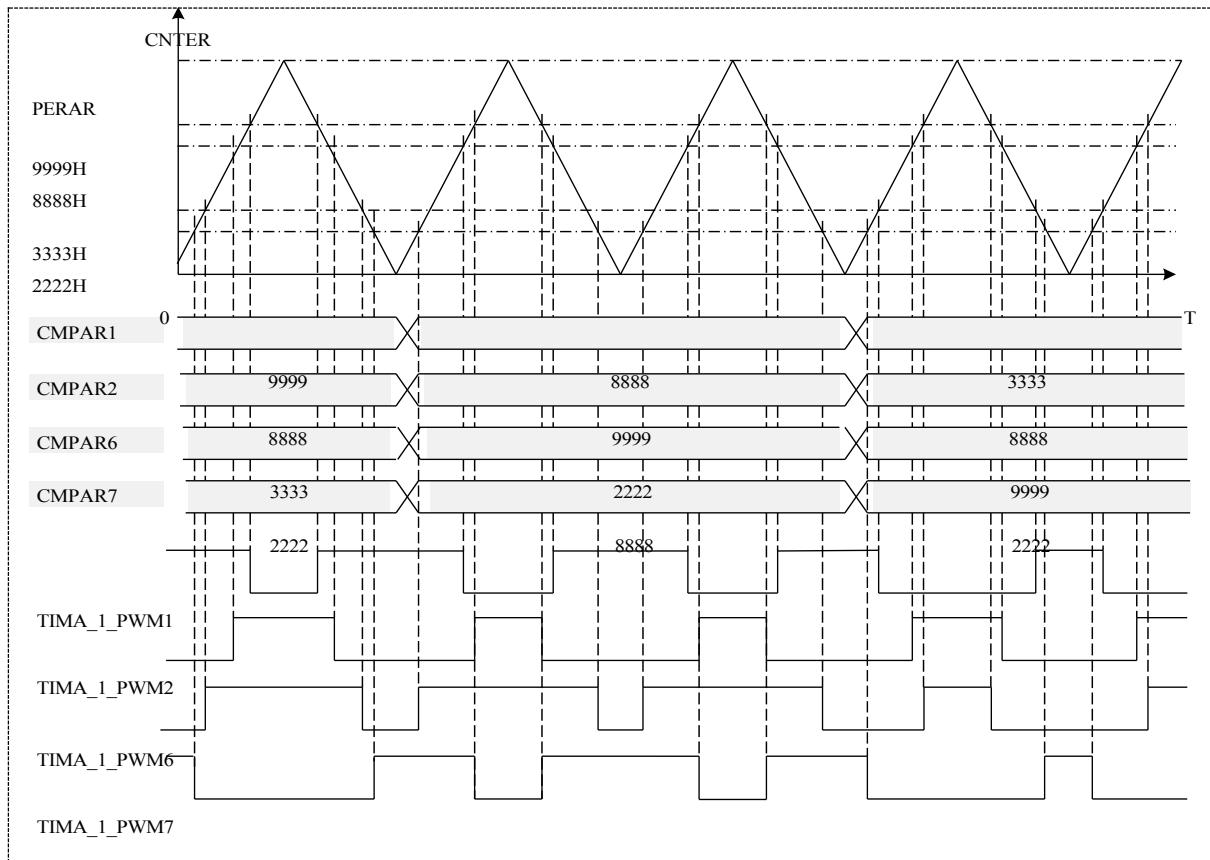


Figure 21-10 General Purpose PWM Output Example

21.3.7.2 Single pulse PWM output

In the sawtooth wave counting mode, the PWM output that generates a single pulse in one cycle can be realized by setting the flip-flop when the comparison reference value is compared and matched (PCONR.CMPC=11) and the flip-flop when the period reference value is compared and matched (PCONR.PERC=11).

Figure 21-12 shows an example of a single pulse PWM output waveform in the triangular wave mode of Unit 1.

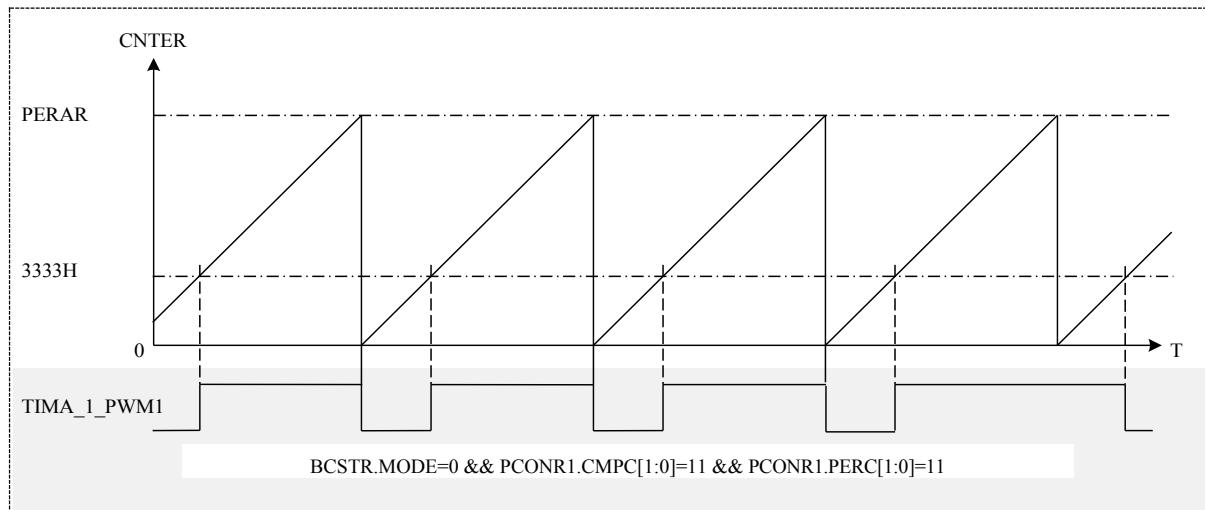


Figure 21-11 Single Pulse PWM Output

21.3.8 orthogonal code counting

By treating the TIMA_<t>_CLKA input as the AIN input, the TIMA_<t>_CLKB input as the BIN input, and the TIMA_<t>_TRIG input as the ZIN input, TimerA is able to realize orthogonal coded counting with three inputs.

The position counting mode can be realized by the individual action of AIN and BIN of each unit; the rotation counting mode can be realized by the combined action of AIN, BIN and ZIN of two units, in which the unit used for position counting is called the position counting unit and the unit used for rotation counting is called the rotation counting unit. In the revolution counting mode, every two units can be combined with each other (unit 1 and 2 combination; unit 3 and 4 combination; unit 5 and 6 combination), and the position counting unit and the revolution counting unit can be specified arbitrarily within the combination.

The counting condition enable of AIN and BIN is realized by setting the orthogonal relationship of TIMA_<t>_CLKA and TIMA_<t>_CLKB in the hardware incremental event selection register (HCUPR) and the hardware decremental event selection register (HCDOR), and the ZIN input is realized by setting the clear enable bit of the hardware triggering event selection register (HCONR) of the position counting unit to zero the position timer, and by

the rotary timer. The ZIN input action is realized by setting the zero enable bit of the position counter unit's hardware trigger event selection register (HCONR) to clear the position timer, and by setting the hardware incremental event selection register (HCUPR) of the revolution unit to count the revolution timer.

21.3.8.1 Position Counting Mode

Orthogonal encoding position counting mode means that the basic counting function, phase difference counting function and direction counting function are realized according to the inputs of AIN and BIN.

basic count

The basic counting action is based on the input clock of the AIN or BIN port as shown in Figure 21-11 below.

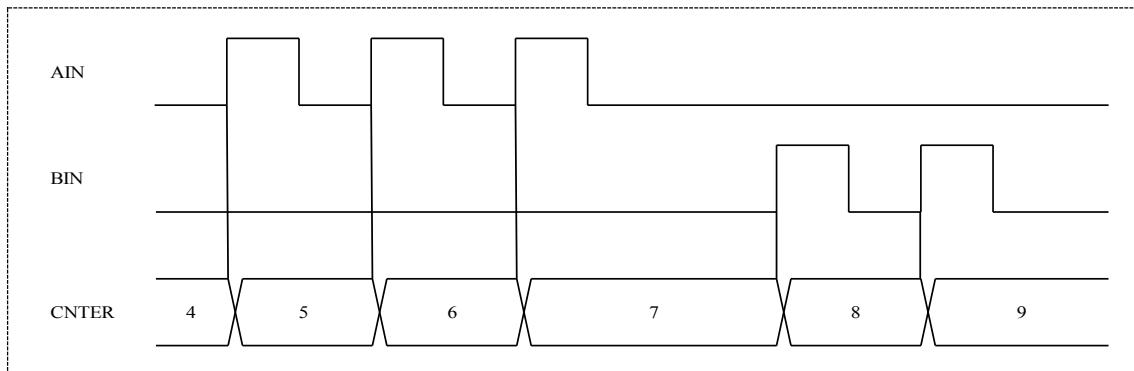


Figure 21-11 Position Mode-Basic Count

phase difference counting

Phase difference counting refers to counting based on the phase relationship between AIN and BIN. Depending on the setting, it is possible to realize 1x counting, 2x counting, 4x counting, and so on, as shown in Fig. 21-12 to Fig. 21-14 below.

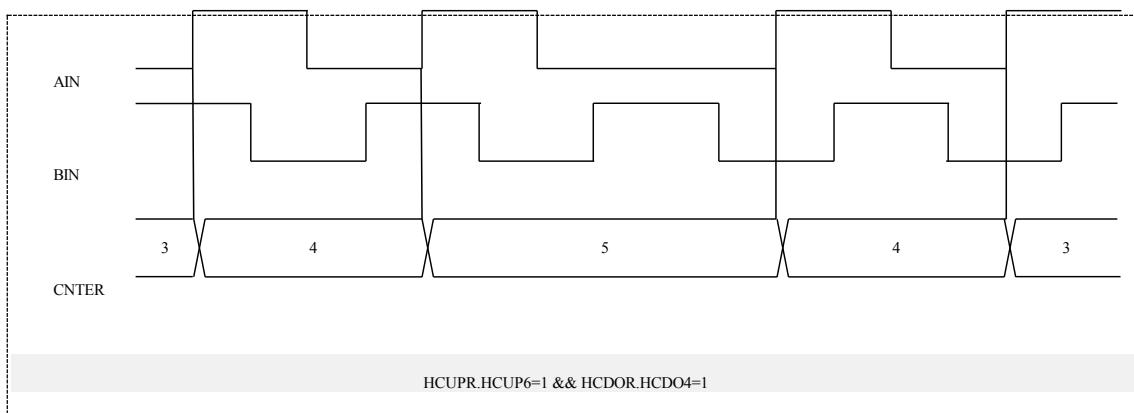


Figure 21-12 Position Count Mode-Phase Difference Count (1x Count)

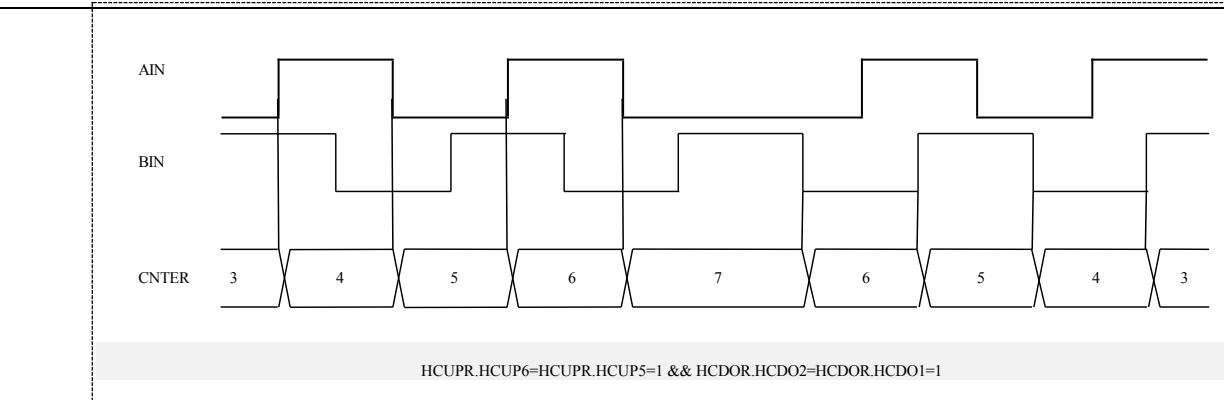


Figure 21-13 Position Count Mode-Phase Difference Count (2x Count)

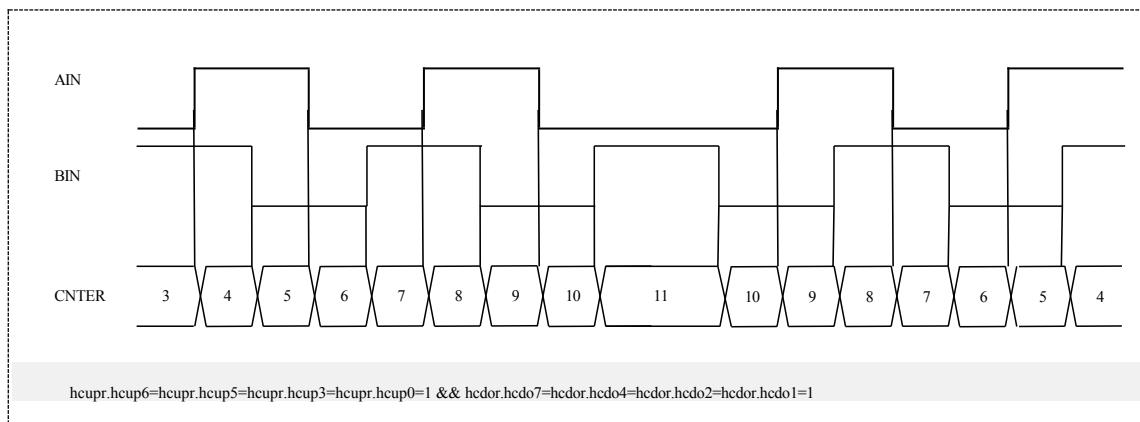


Figure 21-14 Position Count Mode-Phase Difference Count (4x Count)

Direction Counting

Direction counting means that the input state of AIN is set as direction control and the input of BIN is used as clock counting as shown in Figure 21-15 below.

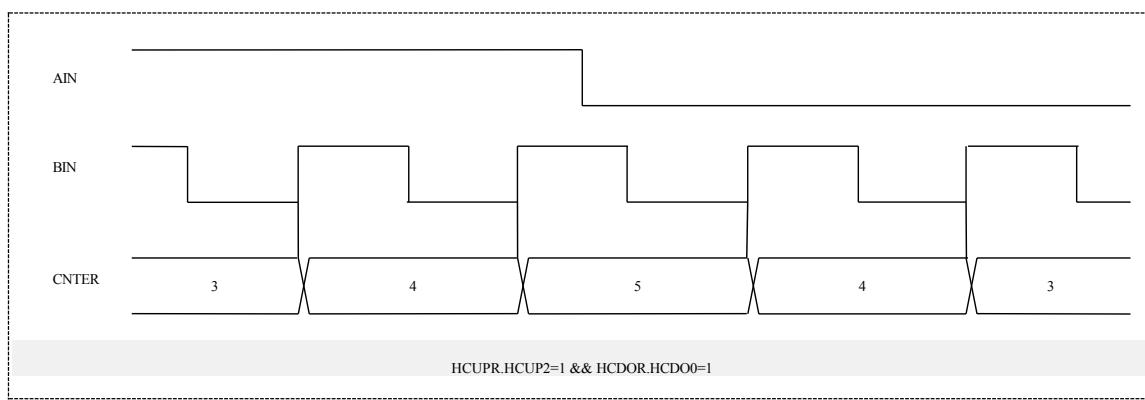


Figure 21-15 Position Count Mode-Direction Count

21.3.8.2 Rotation counting mode

The orthogonal encoding revolution counting mode is to add the ZIN input event to the AIN and BIN counting to realize the judgment of the number of revolutions. In the revolution counting mode, the Z-phase counting function, position overflow counting function, and mixed counting function can be realized according to the counting method of the revolution timer.

Z-phase counting

Z-phase counting is a counting action in which the revolution counting unit counts and the position counting unit is cleared to zero based on the ZIN input. This is shown below.

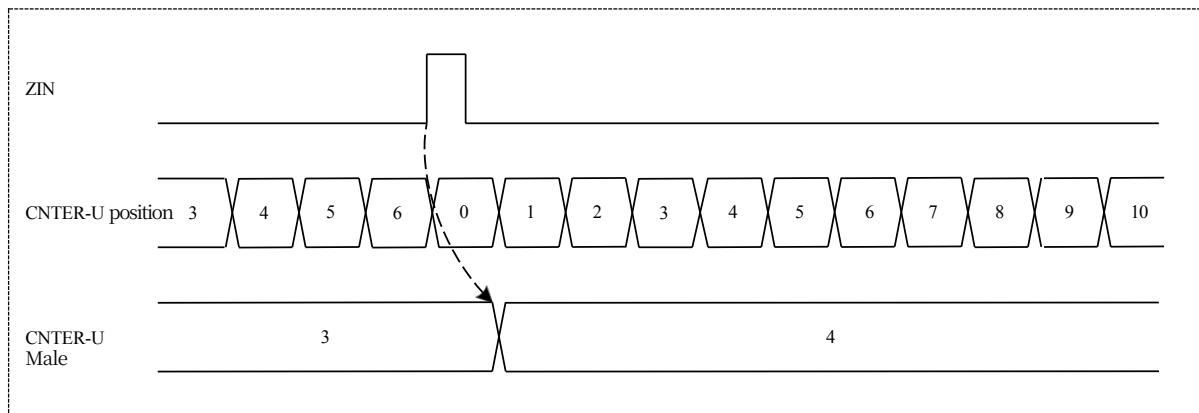


Figure 21-16 Rotation Counting Mode-Z Phase Counting

Position Overflow Count

Position overflow counting means that an overflow event is generated when the count of the position counting unit overflows or underflows, which triggers the timer of the revolution counting unit to perform a count (the ZIN input does not perform the counting of the revolution counting unit or the clearing of the position counting unit in this counting method)

If the hardware increment/decrement event selection register (HCUPR or HCDOR) of the revolution counting unit is enabled by bit12~11 of the increment/decrement event bit, the overflow event of the position counting unit can trigger the revolution counting unit to realize one count. This is shown in Figure 21-17 below.

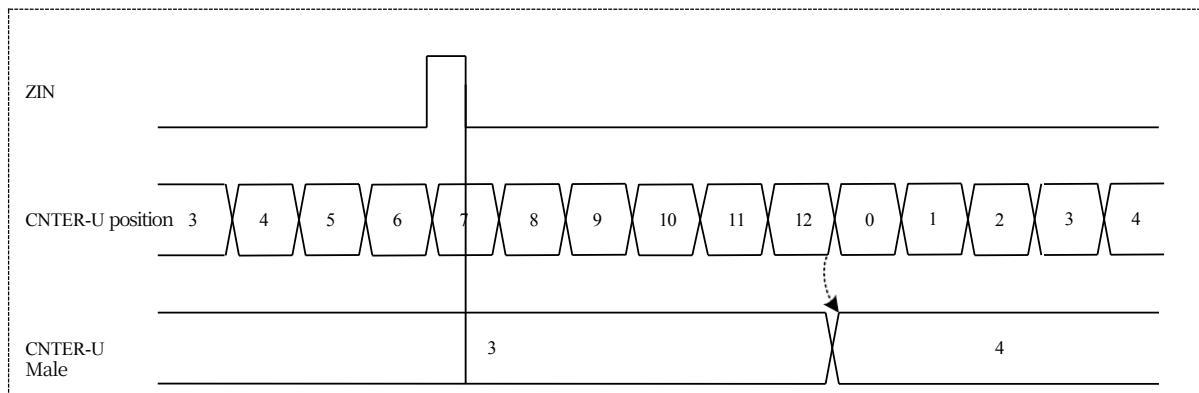


Figure 21-17 Rotation Counting Mode-Position Overflow Counting

Mixed counting

Mixed counting refers to the counting action that combines the above two counting methods, Z-phase counting and position overflow counting, and its realization is also a combination of the above two counting methods. It is realized as a combination of the above two counting methods as shown in Figure 21-18 below.

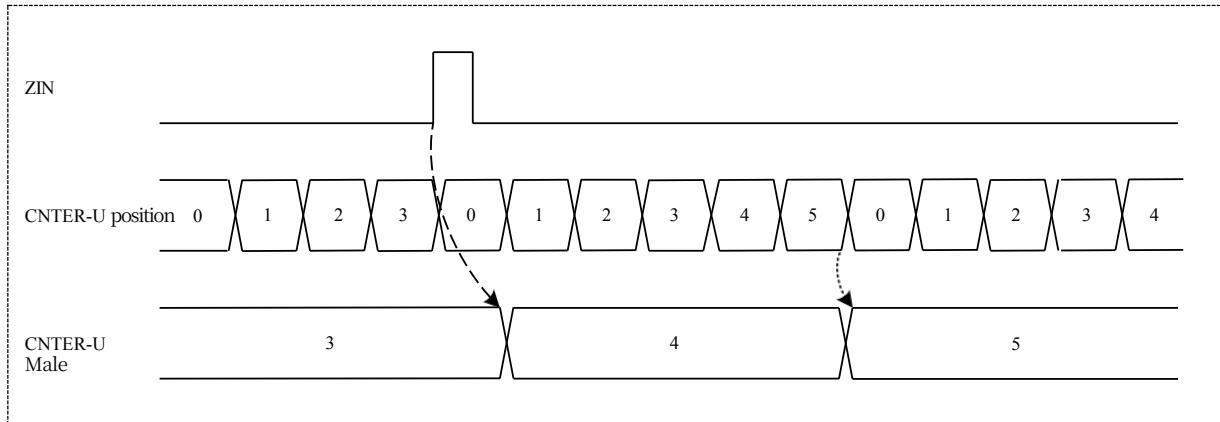


Figure 21-18 Rotation Counting Mode-Mixed Counting

21.4 Interrupt and Event Description

TimerA contains 3 interrupt outputs and 3 event outputs, 1 compare match interrupt and event, and 2 cycle match interrupts and events.

21.4.1 Compare Matching Interrupts and Events

When a comparison match occurs between the comparison reference value register (CMPARn) and the count value comparison, the corresponding bit (STFLR.CMPFn) in the status flag register (STFLR) is set to 1. At this time, if the corresponding bit (ICONR.ITENn) in the interrupt control register (ICONR) is set to 1, the corresponding interrupt request (TMRA_U<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_U<t>_CMP) will be triggered (n=1~8)

The capture input action occurs when the capture input validity condition selected by the capture control register (CCONRn) is generated. At this time, if the corresponding bit (ICONR.ITENn) of the interrupt control register (ICONR) is set to 1, the corresponding interrupt request (TMRA_U<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_U<t>_CMP) will be triggered (n=1~8)

The Compare Match Interrupt and Compare Match Event for the 8 reference values within each unit are not independent outputs. The Compare Match Interrupt is summarized into an interrupt output to the interrupt module through "or logic" (see [Interrupt Controller (INTC] section) and the Compare Match Event is summarized into an event output for selectively triggering another module through "or logic" (see [Interrupt Controller (INTC] section), and the compare match event is aggregated into an event output to selectively trigger another module through "or logic".

21.4.2 Cycle Matching Interrupts and Events

The OVFF or UDFF bit of the Control Status Register (BCSTR) will be set to 1 when the sawtooth mode incremental count reaches the overflow point, the sawtooth mode decremental count reaches the underflow point, or the triangle wave mode count reaches the valley point or peak point. If the BCSTR.ITENOVF or BCSTR.ITENUDF bit is set to 1 to enable the interrupt, the cycle matching interrupt (TMRA_U<t>_OVF and TMRA_U<t>_UDF) can be triggered at the corresponding cycle point for outputting to the Interrupt Module (INTC) while there is no corresponding enable bit to control the cycle matching event. The cycle match event is triggered at the corresponding count cycle point (TMRA_U<t>_OVF and TMRA_U<t>_UDF) and is used to selectively trigger another module.

21.5 Register Description

Table 21-3 shows the register list of the

TimerA module. BASE ADDR.

0x4001_5000 (U1) 0x4001_5400 (U2) 0x4001_5800 (U3)

0x4001_5C00 (U4) 0x4001_6000 (U5) 0x4001_6400 (U6)

Table 21-3 Register List

register name	notation	offset	bit width	reset value
General Purpose Count Register	TMRA_CNTER	0x0000	16	0x0000
Periodic Reference Value Register	TMRA_PERAR	0x0004	16	0xFFFF
Comparison Reference Value Register 1	TMRA_CMPAR1	0x0040	16	0xFFFF
Comparison Reference Value Register 2	TMRA_CMPAR2	0x0044	16	0xFFFF
Comparison Reference Value Register 3	TMRA_CMPAR3	0x0048	16	0xFFFF
Comparison Reference Value Register 4	TMRA_CMPAR4	0x004C	16	0xFFFF
Comparison Reference Value Register 5	TMRA_CMPAR5	0x0050	16	0xFFFF
Comparison Reference Value Register 6	TMRA_CMPAR6	0x0054	16	0xFFFF
Comparison Reference Value Register 7	TMRA_CMPAR7	0x0058	16	0xFFFF
Comparison Reference Value Register 8	TMRA_CMPAR8	0x005C	16	0xFFFF
Control status register L	TMRA_BCCTRL	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
Cache Control Register 3	TMRA_BCONR3	0x00D0	16	0x0000
Cache Control Register 4	TMRA_BCONR4	0x00D8	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

Capture Control Register 5	TMRA_CCONR5	0x0110	16	0x0000
Capture Control Register 6	TMRA_CCONR6	0x0114	16	0x0000
Capture Control Register 7	TMRA_CCONR7	0x0118	16	0x0000
Capture Control Register 8	TMRA_CCONR8	0x011C	16	0x0000
Port Control Register 1	TMRA_PCONR1	0x0140	16	0x0000

register name	notation	offset	bit width	reset value
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
Port Control Register 5	TMRA_PCONR5	0x0150	16	0x0000
Port Control Register 6	TMRA_PCONR6	0x0154	16	0x0000
Port Control Register 7	TMRA_PCONR7	0x0158	16	0x0000
Port Control Register 8	TMRA_PCONR8	0x015C	16	0x0000
Hardware Trigger Event Selection Register	TMRA_HCONR	0x0084	16	0x0000
Hardware incremental event selection register	TMRA_HCUPR	0x0088	16	0x0000
Hardware Decrement Event Select Register	TMRA_HCDOR	0x008C	16	0x0000

21.5.1 General purpose count value register (TMRA_CNTER)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CNT [15:0]															
classifier for honorific people	marking	celebrity	functionality								fill out or in (information on a form)				
b15~b0	CNT [15:0]	numerical value	Current timer count value								R/W				

21.5.2 Periodic Reference Value Register (TMRA_PERAR)

Reset value: 0xFFFF

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PER [15:0]															
classifier for honorific people	marking	celebrity	functionality								fill out or in (information on a form)				
b15~b0	PER [15:0]	countralue e	Setting the count period value for each counting round								R/W				

21.5.3 Comparison Reference Value Register (TMRA_CMPAR1~8)

Reset value: 0xFFFF

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CMP[15:0]															
classifier for honorific people	marking	celebrity	functionality								fill out or in (information on a form)				
b15~b0	CMP[15:0]	Counting benchmark values	Setting the benchmark value								R/W				

21.5.4 Control status register L (TMRA_BCSTRL)

Reset value: 0x02

b7	b6	b5	b4	b3	b2	b1	b0	
				CKDIV[3:0]	SYNST	MODE	DIR	START
<hr/>								
classifier for honorific people	marking	Bit Name	Function	Read/Write				
b7~b4	CKDIV[3:0]	Count Clock Selection	0101: PCLK/32 0110: PCLK/64 0111: PCLK/128 1000: PCLK/256 1001: PCLK/512 1010: PCLK/1024	0000: PCLK 0001: PCLK/2 0010: PCLK/4 0011: PCLK/8 0100: PCLK/16		R/W		
b3	SYNST	Synchronous start enable	Synchronous start function with Unit 1 is active	Note: The setting of this bit in cell 1 is invalid, and it is "0" when read out.	1: R/W			
b2	MODE	Counting Mode	0: Sawtooth wave mode 1: Triangular wave mode		R/W			
b1	DIR	Counting direction	0: Counter counts down 1: Counter counts up		R/W			
b0	START	timer activation	0: Timer off 1: Timer start	Note 1: This bit automatically changes to 0 when a hardware stop condition is in effect Note 2: When the synchronous start function of unit 2~6 is valid, the bit of the corresponding unit is in the unit 1 It is also set after software startup	Note 1: This bit automatically changes to 0 when a hardware stop condition is in effect Note 2: When the synchronous start function of unit 2~6 is valid, the bit of the corresponding unit is in the unit 1 It is also set after software startup	R/W		

21.5.5 Control status register H (TMRA_BCSTRH)

Reset value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
UDFF	OVFF	ITENUDF	ITENOVF			Reserved	
<hr/>							
classifierfor marking Bit Name Function Read/Write							
honorable people							
b7	UDFF	Underflow flag	0: count down, no overflow occurred 1: count down, overflow occurred R/W				
b6	OVFF	Overflow flag	0: count up, no overflow occurred 1: count up, overflow occurred R/W				
b5	ITENUDF	Underflow interrupt enable	0: Count underflow interrupt not enabled 1: count underflow interrupt enable R/W				
b4	ITENOVF	Overflow interrupt enable	0: Count overflow interrupt not enabled 1: Count overflow interrupt enable R/W				
b3~b0	Reserved	-0" for reading, "0" for writing.	writes "0".	R/W			Reads "0",

21.5.6 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 EN8	IT EN7	IT EN6	IT EN5	IT EN4	IT EN3	IT EN2	IT EN1
b15~b8	Reserved	-	Reads "0", writes "0".												R/W

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b7	ITEN8	Count Match		This interrupt is invalid
Interrupt Enable 8		b1	ITEN2	1: This interrupt is enabled when
		Enable 2	the CMPAR8 register is equal to the count	
b6	ITEN7	Count Match		value, or when a capture input event occurs
Interrupt Enable 7		b0	ITEN1	0: This interrupt is invalidated when the CMPAR7 register
		Enable 1	is equal to the count value, or when a capture input event occurs	
b5	ITEN6	Count Match		1: This interrupt is enabled when
Interrupt Enable 6				the CMPAR7 register is equal to the count value, or when a capture input event occurs
b4	ITEN5	Count Match		0: This interrupt is invalid when
Interrupt Enable 5				the CMPAR6 register is equal to the count value, or when a capture input event occurs
b3	ITEN4	Count Match		1: This interrupt is enabled when
Interrupt Enable 4				the CMPAR7 register is equal to the count value, or when a capture input event occurs
b2	ITEN3	Count Match		0: This interrupt is invalid when
Interrupt Enable 3				the CMPAR6 register is equal to the count value, or when a capture input event occurs

1: This interrupt is enabled when the CMPAR6 register
is equal to the count value, or when a capture input
event occurs

R/W

0: This interrupt is invalid when the CMPAR5 register
is equal to the count value, or when a capture input
event occurs

1: This interrupt is enabled when the CMPAR5 register
is equal to the count value, or when a capture input
event occurs

R/W

0: This interrupt is invalid when the CMPAR4 register
is equal to the count value, or when a capture input
event occurs

R/W

1: This interrupt is enabled when the CMPAR4 register
is equal to the count value, or when a capture input
event occurs

0: This interrupt is invalid when the CMPAR3 register
is equal to the count value, or when a capture input
event occurs

R/W

1: This interrupt is enabled when the CMPAR3 register
is equal to the count value, or when a capture input
event occurs

R/W

0: This interrupt is invalid when the CMPAR2 register
is equal to the count value, or when a capture input
event occurs

R/W

1: This interrupt is enabled when the CMPAR2 register
is equal to the count value, or when a capture input
event occurs

0: This interrupt is invalid when the CMPAR1 register
is equal to the count value, or when a capture input
event occurs

R/W

1: When the CMPAR1 register is equal to the count
value, or when a capture input event occurs, the
This interrupt enables

21.5.7 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 EN8	ET EN7	ET EN6	ET EN5	ET EN4	ET EN3	ET EN2	ET EN1
classifier for honorific people	marking	celebrity	functionality								fill out or in (information on a form)				
b15~b8	Reserved	-	Reads "0", writes "0".								R/W				
b7	Event Enable 8	ETEN8	Count Match								This event output is invalid				
b6	Event Enable 7	ETEN7	Count Match								1: This event output is enabled when the CMPAR8 register is equal to the count value, or when a capture input event occurs				
b5	Event Enable 6	ETEN6	Count Match								0: This event output is invalid when the CMPAR7 register is equal to the count value, or when a capture input event occurs				
b4	Event Enable 5	ETEN5	Count Match								1: When the CMPAR7 register is equal to the count value, or when a capture input event occurs				
b3	Event Enable 4	ETEN4	Count Match								is equal to the count value, or when a capture input event occurs, this event output is enabled				
b2	Event Enable 3	ETEN3	Count Match								0: This event				

1: This event output is enabled when the CMPAR6
register is equal to the count value, or when a capture
input event occurs

0: This event output is invalid when the CMPAR5
register is equal to the count value, or when a capture
input event occurs

1: This event output is enabled when the CMPAR5
register is equal to the count value, or when a capture
input event occurs

0: This event output is invalid when the CMPAR4
register is equal to the count value, or when a capture
input event occurs

1: This event output is enabled when the CMPAR4
register is equal to the count value, or when a capture
input event occurs

0: This event output is invalid when the CMPAR3
register is equal to the count value, or when a capture
input event occurs

1: This event output is enabled when the CMPAR3
register is equal to the count value, or when a capture
input event occurs

0: This event output is invalid when the CMPAR2
register is equal to the count value, or when a capture
input event occurs

1: This event output is enabled when the CMPAR2
register is equal to the count value, or when a capture
input event occurs

0: This event output is invalid when the CMPAR1
register is equal to the count value, or when a capture
input event occurs

1: When the CMPAR1 register is equal to the count
value, or when a capture input event occurs, the
This event output enables

21.5.8 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		

classifier for marking		celebrity	functionality	fill out or in (information on a form)
b15	Reserved	-	Reads "0" and writes "0".	R/W
b14~b13	NOFICKCB[1:0]	Filter Sample 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 TIMA_<t>_CLKB port input filter function 1: TIMA_<t>_CLKB port input filter function enable	R/W
b11	Reserved	-Reserved	Read "0", write "0".	R/W
b10~b9 reference time	NOFICKCA[1:0]	Filter sampling Clock Select CA	00: PCLK 01: PCLK/4 10: PCLK/16 11: PCLK/64 0: TIMA_<t>_CLKA port input filtering	R/W
b8	NOFIENCA	Capture input port filter CA	function is disabled 1: TIMA_<t>_CLKA port input filter function enable	
b7~b3	Reserved	-Reserved	Read "0", write "0".	R/W
b2~b1 reference time	NOFICKTG[1:0]	Filter sampling The clock chooses TG.	00: PCLK 01: PCLK/4 10: PCLK/16 11: PCLK/64 0: TIMA_<t>_TRIG input port filtering	R/W
b0	NOFIENTG	Capture input port filtering TG	function is disabled 1: TIMA_<t>_TRIG input port filtering function enable	

21.5.9 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								CMP F8	CMP F7	CMP F6	CMP F5	CMP F4	CMP F3	CMP F2	CMP F1
classifier for honorific people	marking	celebrity													fill out or in (information on a form)
b15-b8	Reserved	-													R
b7 Flag 8	CMPF8	Count Match													0: The value of the CMPAR8 register is not equal to the count value and TIMA_<t>_PWM8 has not occurred implementation of its programme of work. Match Flag 1
b6 Flag 7	CMPF7	Count Match													1: The value of the CMPAR8 register is equal to the count value, or the TIMA_<t>_PW M8 capture completion action occurs
b5 Flag 6	CMPF6	Count Match													0: The value of the CMPAR7 register is not equal to the count value and the TIMA_<t>_PWM7 capture completion action occurs
b4 Flag 5	CMPF5	Count Match													1: The value of the CMPAR7 register is equal to the count value, or the TIMA_<t>_PW M7 capture completion action occurs
b3 Flag 4	CMPF4	Count Match													0: The value of the CMPAR6 register is not equal to the count value and
b2 Flag 3	CMPF3	Count Match													1: The value of the CMPAR7 register is equal to the count value, or the TIMA_<t>_PW M7 capture completion action occurs
b1 Flag 2	CMPF2	Count Match													0: The value of the CMPAR6 register is not equal to the count value and
(b) The Organization's contribution to the implementation of the Convention and to the															

has not occurred

R/W

1: The value of the CMPAR6 register is equal to
the count value, or the TIMA_<t>_PWM6

capture completion action occurs

R/W

0: The value of the CMPAR5 register is not equal
to the count value and the TIMA_<t>_PWM5

capture completion action has not occurred

1: The value of the CMPAR5 register is equal to
the count value, or the TIMA_<t>_PWM5

R/W

capture completion action occurs

0: The value of the CMPAR4 register is not equal
to the count value and the TIMA_<t>_PWM4

R/W

capture completion action has not occurred

1: The value of the CMPAR4 register is equal to
the count value, or the TIMA_<t>_PWM4

capture completion action occurs

0: The value of the CMPAR3 register is not equal
to the count value and the TIMA_<t>_PWM3

capture completion action has not occurred

1: The value of the CMPAR3 register is equal to
the count value, or the TIMA_<t>_PWM3

capture completion action occurs

0: The value of the CMPAR2 register is not equal
to the count value and the TIMA_<t>_PWM2

R/W

capture completion action has not occurred

1: The value of the CMPAR2 register is equal to
the count value, or the TIMA_<t>_PWM2

capture completion action occurs

0: The value of the CMPAR1 register is not equal
to the count value and the TIMA_<t>_PWM1

capture completion action has not occurred

1: The value of the CMPAR1 register is equal to
the count value, or the TIMA_<t>_PWM1

capture completion action occurs

21.5.10 Cache Control Register (TMRA_BCONR1~4)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0		
Reserved														BSE1	BSE0	BEN	
classifier for honorific people	marking	celebrity				functionality				fill out or in (information on a form)							
b15~b3	Reserved	-				Reads "0" and writes "0".				R/W							
b2	BSE1	Triangular Wave Cache	0: Cache value is not transmitted when counting to valley in triangle wave counting mode				R/W				1: The cache value is transmitted when the delta wave counting mode counts to the valley point, i.e: CMMARm -> CMPARn (m=2, 4, 6, 8, n=1, 3, 5, 7)						
b1	BSE0	Triangle Wave Cache Transmit Selection 0	0: The buffer value is not transmitted when the triangle wave counting mode counts to the peak point	R/W				1: The cache value is transmitted when the triangle wave counting mode counts to the peak point, i.e: CMMARm -> CMPARn (m=2, 4, 6, 8, n=1, 3, 5, 7)				R/W					
b0	BEN	Cache Enable	0: Invalid cache function for CMPARn base value	R/W				1: Cache function valid for CMPARn baseline values (n=1, 3, 5, 7)				R/W					

21.5.11 Capture Control Register (TMRA_CCONR1~8)

Reset value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	NOFI CKCP[1:0]	NOFI ENCP	-	-	HICP 4	HICP 3	-	HICP 2	HICP 1	HICP 0	-	-	-	CAP MD	

classifier for honorific people	marking	celebrity	functionality	fill out or in (infor matio n on a form)
b15	Reserved	-	Reads "0", writes "0".	R/W
b14~b13	NOFICKCP[1:0]	Filter Sample Reference Clock Select CP	00: PCLK 01: PCLK/4 10: PCLK/16 11: PCLK/64	R/W
b12	NOFIENCP	Capture input port filtering CP	0: TIMA_<t>_PWMM port input filtering function disabled 1: TIMA_<t>_PWMM port input filtering enabled (n=1~8)	R/W
b11~b10	Reserved	-	Reads "0" and writes "0".	R/W
b9	HICP4	Capture input condition enable 4	0: Channel m+1 does not capture when the TIMA_<t>_TRIG port input is sampled on the falling edge Input Action 1: When the TIMA_<t>_TRIG port input is sampled to the falling edge, channel m+1 generates a capture output entry action Note: This bit is only valid for the CCONR3 register. That is, after this bit is valid and the corresponding event occurs When CCONR4.CAPMD=1, the current counter value is captured and saved in the CMPAR4 and STFLR.CMPF4 is set	R/W
b8	HICP3	Capture input condition enable 3	0: Channel m+1 does not capture when TIMA_<t>_TRIG port inputs are sampled to the rising edge Input Action 1: When the TIMA_<t>_TRIG port input is sampled to the rising edge, channel m+1 generates a capture output entry action Note: This bit is only valid for the CCONR3 register. That is, after this bit is valid and the corresponding event occurs When CCONR4.CAPMD=1, the current counter value is captured and saved in the CMPAR4 and STFLR.CMPF4 is set	R/W
b7	Reserved	-	Reads "0", writes "0".	R/W
b6	HICP2	Capture input condition enable 2	0: No capture input action occurs when the event specified in the TMRA_TRGSEL1 register occurs 1: Capture input action is generated when the event specified in the TMRA_TRGSEL1 register occurs	R/W

			0: No capture input action occurs when the TIMA_<t>_PWMn port input is sampled to falling edge 1: TIMA_<t>_PWMn port input sampled to falling edge generates capture input action (n=1~8)	R/W
b5	HICP1	Capture input condition enable 1	0: No capture input action occurs when the TIMA_<t>_PWMn port input is sampled to a rising edge 1: TIMA_<t>_PWMn port input sampled to rising edge generates capture input action (n=1~8)	R/W
b4	HICP0	Capture input condition enable 0	0: No capture input action occurs when the TIMA_<t>_PWMn port input is sampled to a rising edge 1: TIMA_<t>_PWMn port input sampled to rising edge generates capture input action (n=1~8)	R/W
b3~b1	Reserved	-	Reads "0", writes "0".	R/W
b0	CAPMD	Function mode selection	0: Comparison output function 1: Capture input function	R/W

21.5.12 Port Control Registers (TMRA_PCONR1~8)

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]					
classifier for honorific people	marking		celebrity			functionality									
b15~b13	Reserved	-		Reads "0" and writes "0".										R/W	
b12	OUTEN	output enable (computing)		0: Invalid TIMA_<t>_PWMn port output at PWM output function 1: TIMA_<t>_PWMn port output valid for PWM output function (n=1~8)										R/W	
b11~b10	Reserved	-		Reads "0", writes "0".										R/W	
b9~b8 setting	FORC[1:0]	Force port state		0x: Setting invalid definition 10: Start of next cycle, TIMA_<t>_PWMn port output set low 11: Start of next cycle, TIMA_<t>_PWMn port output is set high Note 1: Lower cycle means hardware counting mode or sawtooth wave counting to the upper or lower overflow point, triangle wave counting to the valley point Note 2: This register bit can be used to realize the control of PWM output duty cycle 0% or 100% 00: When the count value is equal to PERAR, the output of TIMA_<t>_PWMn port is set to low level 01: TIMA_<t>_PWMn port output set high when count value is equal to PERAR										R/W	
b7~b6	PERC[1:0]	Status Setting	Port when period value matches	10: TIMA_<t>_PWMn port output holds when count value is equal to PERAR previous state 11: When the count value is equal to PERAR, the TIMA_<t>_PWMn port output is set to the inverting level (n=1~8) 00: TIMA_<t>_PWMn port output set low when count value is equal to CMPARn 01: TIMA_<t>_PWMn port output set high when count value is equal to CMPARn 10: When the count value is equal to CMPARn, the TIMA_<t>_PWMn port output maintains the previous state 11: When the count value is equal to CMPARn, the TIMA_<t>_PWMn port output is set to the inverted level (n=1~8)										(n=1~8)	
b5~b4 CMPC[1:0]		Port state setting when comparison value matches		00: TIMA_<t>_PWMn port output set low when counting is stopped 01: TIMA_<t>_PWMn port output set high when counting stops 10: When counting stops, the TIMA_<t>_PWMn port output maintains its previous state 11: When counting stops, the TIMA_<t>_PWMn port output maintains its previous state											
b3~b2	STPC[1:0]	Port status setting when counting stops		00: TIMA_<t>_PWMn port output set low when counting is stopped 01: TIMA_<t>_PWMn port output set high when counting stops 10: When counting stops, the TIMA_<t>_PWMn port output maintains its previous state 11: When counting stops, the TIMA_<t>_PWMn port output maintains its previous state											

R/W

b1~b0 start of count	STAC[1:0]	Port state at state setting	10: TIMA_<t>_PWMn port outputs remain in previous state when counting starts 11: TIMA_<t>_PWMn port output holds previous state when counting starts (n=1~8) Note: This bit is set only without crossover (BCSTR.CKDIV=4'h0)	R/W
-------------------------	-----------	--------------------------------	--	-----

Valid, other crossovers please set to 2'b10 or 2'b11.

21.5.13 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
HCL E	HCL E	HCL E	HCL E	-	HCL E	HCL E	HCL E	-	HST P	HST P	HST P	-	HST A	HST A	HST A

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b15	HCLE6	Hardware clear condition 6	Condition: TIMA_<t>_PWM3 port input sampled to falling edge 0: Invalid hardware clear when condition matches 1: Hardware clear is valid when conditions are matched	R/W
b14	HCLE5	Hardware clear condition 5	Condition: TIMA_<t>_PWM3 port input sampled to rising edge 0: Invalid hardware clear when condition matches 1: Hardware clear valid when conditions match	R/W
b13	HCLE4	Hardware clear condition 4	Condition: When this unit is unit m, the TRIG port input of unit n is sampled to the falling edge (when the (n = 2, 4, 6 when m = 1, 3, 5; n = 1, 3, 5 when m = 2, 4, 6) 0: Invalid hardware clear when condition matches 1: Hardware clear valid when conditions match	R/W
b12	HCLE3	Hardware clear condition 3	Condition: When this unit is unit m, the TRIG port input of unit n is sampled to the rising edge (when the (n=2, 4, 6 when m=1, 3, 5; n=1, 3, 5 when m=2, 4, 6) 0: hardware clearing is invalid when conditions are matched 1: Hardware clear is valid when conditions are matched	R/W
b11	Reserved	-	Reads "0", writes "0".	R/W
b10	HCLE2	Hardware zero condition 2	Condition: the event specified in the TMRA_TRGSEL0 register occurs 0: Invalid hardware clear when condition matches 1: Hardware clear valid when conditions match	R/W
b9	HCLE1	Hardware clear condition 1	Condition: TIMA_TRIG port input sampled to falling edge 0: Invalid hardware clear when condition matches 1: Hardware clear valid when conditions match	R/W
b8	HCLE0	Hardware clear condition 0	Condition: TIMA_TRIG port input sampled to rising edge 0: Invalid hardware clear when condition matches 1: Hardware clear is valid when conditions are matched	R/W
b7	Reserved	-	Reads "0", writes "0".	R/W
b6	HSTP2	Hardware stop condition 2	Condition: the event specified in the TMRA_TRGSEL0 register occurs 0: Invalid hardware stop when condition matches 1: Hardware stop is valid when conditions match	R/W

b5	HSTP1	Hardware stop condition 1	0: Invalid hardware stop when condition matches 1: Hardware stop is valid when conditions match	R/W
b4	HSTP0	Hardware stop condition 0	Condition: TIMA_<t>_TRIG port input sampled to rising edge 0: Hardware stop is invalid when condition matches 1: Hardware stop is valid when conditions match	R/W
b3	Reserved	-	Reads "0" and writes "0". Condition: the event specified in the TMRA_TRGSEL0 register occurs	R/W
b2	HSTA2	Hardware startup condition 2	0: Hardware startup invalid when conditions match 1: Hardware startup is valid when conditions match	R/W
b1	HSTA1	Hardware startup condition 1	Condition:	R/W

1) Inputs to this unit's TIMA_<t>_TRIG port are sampled to the falling edge (synchronous start function disabled)

2) TIMA_1_TRIG port input sampled to falling edge (synchronous start function active)

0: Hardware startup invalid when conditions match

1: Hardware startup is valid when conditions match

Note: Condition 2) Only cell 2~6 can be

selected, cell 1 is not valid condition:

1) Input to this unit's TIMA_<t>_TRIG port is sampled to the rising edge (synchronous start function disabled)

R/W

2) TIMA_1_TRIG port input sampled to rising edge (synchronous start function active)

0: Hardware startup invalid when conditions match

1: Hardware startup is valid when conditions match

Note: Condition 2) Only cell 2~6 can be selected, cell 1 is invalid.

b0 HSTA0 Hardware

Startup Condition 0

21.5.14 Hardware-recursive 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

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b15~b13	Reserved	-	Reads "0", writes "0".	R/W
b12	HCUP12	Hardware incremental condition 12	Condition: counting underflow occurs in cell n when this cell is cell m (n = 2, 4, 6 when m = 1, 3, 5; n = 1, 3, 5 when m = 2, 4, 6) 0: Hardware increment is invalid when condition matches	R/W
			1: Hardware recursion is valid when conditions are matched	
b11	HCUP11	Hardware incremental condition 11	Condition: count overflow occurs in cell n when this cell is cell m (n = 2, 4, 6 when m = 1, 3, 5; n = 1, 3, 5 when m = 2, 4, 6) 0: Hardware increment is invalid when condition matches	R/W
			1: Hardware recursion is valid when conditions are matched	
b10	HCUP10	Hardware incremental condition 10	Condition: the event specified in the TMRA_TRGSEL0 register occurs	0: R/W
			1: Hardware recursion is valid when conditions are matched	
b9	HCUP9	Hardware incremental condition 9	Condition: Sampling to falling edge on TIMA_<t>_TRIG port	0: R/W
			Hardware increment is invalid when the condition matches.	
			1: Hardware recursion is valid when conditions are matched	
b8	HCUP8	Hardware incremental condition 8	Condition: Sampling to rising edge on TIMA_<t>_TRIG port	0: R/W
			Hardware increment is invalid when the condition matches.	
			1: Hardware recursion is valid when conditions are matched	
b7	HCUP7	Hardware incremental condition 7 falling edge	Condition: TIMA_<t>_CLKB port is high on the TIMA_<t>_CLKA port	Sample to R/W
			0: Hardware increment is invalid when condition matches	
			1: Hardware recursion is valid when conditions are matched	
b6	HCUP6	Hardware incremental condition 6 rising edge	Condition: TIMA_<t>_CLKB port is high on the TIMA_<t>_CLKA port	Sample to R/W
			0: Hardware increment is invalid when condition matches	
			1: Hardware recursion is valid when conditions are matched	
b5	HCUP5	Hardware incremental condition 5 falling edge	Condition: TIMA_<t>_CLKB port is low on the TIMA_<t>_CLKA port	Sample to R/W
			0: Hardware increment is invalid when condition matches	
			1: Hardware recursion is valid when conditions are matched	

b4	HCUP4	Hardware incremental condition 4 rising edge	R/W	
			0: Hardware increment is invalid when condition matches	
			1: Hardware recursion is valid when conditions are matched	
			Condition: TIMA_<t>_CLKA port is high on TIMA_<t>_CLKB port	Sample to
b3	HCUP3	Hardware incremental condition 3 falling edge	R/W	
			0: Hardware increment is invalid when condition matches	
			1: Hardware recursion is valid when conditions are matched	
b2	HCUP2	Hardware recursive condition 2 when TIMA_<t>_CLKA port is high on TIMA_<t>_CLKB port		condition:
				R/W

			Sample to rising edge	
			0: Hardware increment is invalid when condition matches	
			1: Hardware recursion is valid when conditions are matched	
b1	HCUP1	Hardware	Condition: TIMA_<t>_CLKA port is low, TIMA_<t>_CLKB port is sampled on falling edge	R/W
	incremental condition 1		0: Hardware increment is invalid when condition matches	
			1: Hardware recursion is valid when conditions are matched	
b0	HCUP0	Hardware	Condition: rising edge sampled on TIMA_<t>_CLKB port when TIMA_<t>_CLKA port is low	R/W
	incremental condition 0		0: Hardware increment is invalid when condition matches	
			1: Hardware recursion is valid when conditions are matched	

21.5.15 Hardware Decrement Event Selection 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
<hr/>															
classifier for honorific people	marking	celebrity													fill out or in (information on a form)
b15~b13	Reserved	-													R/W
b12	HCDO12	Hardware decreasing condition 12													Condition: counting underflow occurs in cell n when this cell is cell m (n = 2, 4, 6 when m = 1, 3, 5; n = 1, 3, 5 when m = 2, 4, 6) 0: hardware decrement is invalid when condition matches 1: Hardware decrement is valid when conditions are matched
b11	HCDO11	Hardware decreasing condition11													Condition: count overflow occurs in cell n when this cell is cell m (n = 2, 4, 6 when m = 1, 3, 5; n = 1, 3, 5 when m = 2, 4, 6) 0: hardware decrement is invalid when condition matches 1: Hardware decrement is valid when conditions are matched
b10	HCDO10	Hardware decrement condition 10													Condition: the event specified in the TMRA_TRGSEL0 register occurs 0: R/W
b9	HCDO9	Hardware Decrement Condition 9													Condition: Sampling to falling edge on TIMA_<t>_TRIG port 0: R/W
b8	HCDO8	Hardware decrement condition 8													Condition: Sampling to rising edge on TIMA_<t>_TRIG port 0: R/W
b7	HCDO7	Hardware Decrement Condition 7 falling edge													R/W 0: hardware decrement is invalid when condition matches 1: Hardware decrementing is valid when conditions are matched Condition: TIMA_<t>_CLKB port is high on the TIMA_<t>_CLKA port Sample to
b6	HCDO6	Hardware Decrement Condition 6 rising edge													R/W 0: hardware decrement is invalid when condition matches 1: Hardware decrement is valid when conditions match Condition: TIMA_<t>_CLKB port is high on the TIMA_<t>_CLKA port Sample to
b5	HCDO5	Hardware Decrement Condition 5 falling edge													R/W 0: hardware decrement is invalid when condition matches 1: Hardware decrement is valid when conditions match Condition: TIMA_<t>_CLKB port is low on the TIMA_<t>_CLKA port Sample to

b4	HCDO4	Hardware Decrement Condition 4 rising edge	R/W	Condition: TIMA_<t>_CLKA port is high on TIMA_<t>_CLKB port	Sample to
b3	HCDO3	Hardware Decrement Condition 3 falling edge	R/W	0: hardware decrement is invalid when condition matches 1: Hardware decrement is valid when conditions match	
b2	HCDO2	Hardware Decrement Condition 2	R/W	Condition: when TIMA_<t>_CLKA port is high, TIMA_<t>_CLKB port on the sampled to rising edge	

			0: hardware decrement is invalid when condition matches	
			1: Hardware decrement is valid when conditions match	
b1	HCDO1	Hardware	Condition: TIMA_<t>_CLKA port is low, TIMA_<t>_CLKB port is sampled on falling edge	R/W
		decreasing condition 1	0: hardware decrement is invalid when condition matches	
			1: Hardware decrement is valid when conditions are matched	
b0	HCDO0	Hardware	Condition: rising edge sampled on TIMA_<t>_CLKB port when TIMA_<t>_CLKA port is low	R/W
		decreasing condition 0	0: hardware decrement is invalid when condition matches	
			1: Hardware decrement is valid when conditions are matched	

22 General-purpose timer (Timer0)

22.1 brief

General purpose timer 0 (Timer0) is a basic timer that can be used for both synchronous and asynchronous counting. The timer contains 2 channels (CH-A and CH-B) which can generate a compare match event during counting. This event can trigger an interrupt or be used as an event output to control other modules. This series has a 2-unit Timer0.

22.2 basic block diagram

The basic block diagram of Timer0 is shown in Figure 22-1.

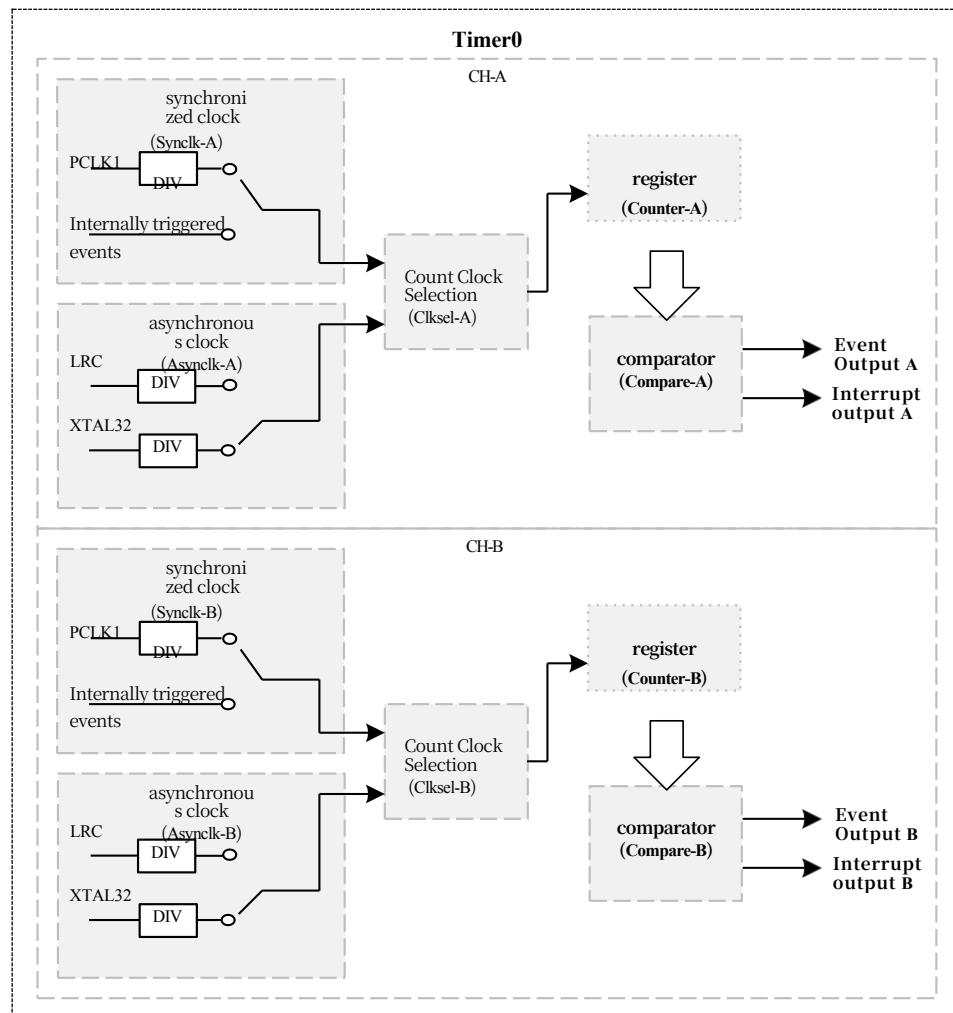


Figure 22-1 Timer0 Basic Block Diagram

22.3 Functional Description

22.3.1 Clock source selection

The counting method of Timer0 can be selected from synchronous counting method or asynchronous counting method.

The synchronous counting method means that the timer count clock and the bus access clock (register read/write operation clock) have a synchronous timing relationship; the asynchronous counting method means that the timer count clock and the bus access clock (register read/write operation clock) have a non-synchronous timing relationship. When a register is read in the asynchronous counting method, the state of the timer, etc., may be changing, and an unanticipated state may be read. Therefore, in the asynchronous counting method, the register read operation must be realized in the counting stop state.

22.3.1.1 Synchronous Counting Clock Source

In synchronous counting mode (BCONR.SYNSA=0) the clock source can be selected as follows (BCONR._SYNCLKA setting)

- a) PCLK1 and 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 divisions of PCLK1 are used as synchronous count clock (BCONR.SYNCLKA=0 & BCONR.CKDIVA [3:0] set)
- b) Internal hardware triggered event input as synchronized count clock (BCONR.SYNCLKA=1)

22.3.1.2 Asynchronous Counting Clock Source

In asynchronous counting mode (BCONR.SYNSA=1) the clock source can be selected as follows (BCONR.ASYNCLK A setting)

- a) LRC clock source input and its 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 divisions as asynchronous count clock (BCONR.ASYNCLK A=0 & BCONR.CKDIVA [3:0] set)
- b) XTAL32 Clock source input and its 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 divisions as an asynchronous count clock (BCONR.ASYNCLK A=1 & BCONR.CKDIVA [3:0] set)

22.3.2 Basic Counting Action

Each channel of Timer0 can set the reference count value, and generate a count comparison match event when the count value and the reference value are equal. This is shown in Figure 22-2.

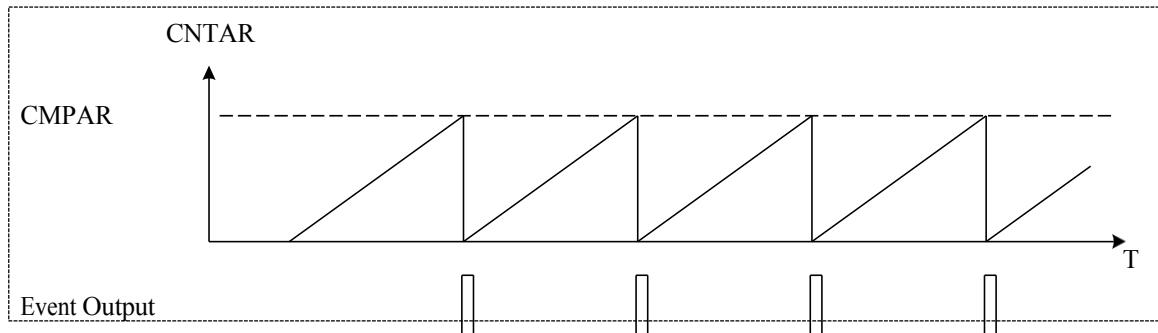


Figure 22-2 Timer0 Count Timing Chart

22.3.3 Hardware-triggered actions

The 2 channels of Timer0 have a common internal hardware trigger source, which can be used to control the timer state (counting, starting, stopping, clearing) as well as the capture input action, etc. through the relevant settings in the Basic Control Register (BCONR).

Source selection for this hardware trigger source is achieved by entering the corresponding number into the Hardware Trigger Selection Register (HTSSR), refer to the Interrupt Controller (INTC) chapter for the specific event correspondence. To use the internal hardware trigger function, you need to enable the Peripheral Circuit Trigger Function bit (AOS) of the Function Clock Control 0 register (PWC_FCG0) first.

22.4 Interrupt and Event Description

22.4.1 interrupt output

A Timer0 contains two interrupts, the count compare match interrupt or the input capture interrupt for channel A and channel B respectively.

There are two base value registers (CMPAR, CMPBR), which can be compared with the count value registers (CNTAR, CNTBR) to generate a valid signal for comparison. When the count comparison is matched, the STFLR.CMAF and STFLR.CMBF bits in the Status Flag Register (STFLR) are set to 1. At this time, if the INTENA bit in the BCONR of the Basic Control Register (BCONR) is set to enable interrupts, the corresponding interrupt requests (TMR0_Um_GCMn, m=1, 2; n=A, B) will also be generated. = A, B) will also be triggered.

When an internal hardware trigger input is used as a capture input condition, a corresponding capture input action can be generated. In this case, if you set the basic control register (The corresponding interrupt request (TMR0_Um_GCMn, m=1, 2; n=A, B) is triggered if the BCONR.INTENA bit of (BCONR) enables the interrupt.

Attention:

-The compare match interrupt (TMR0_U1_GCMA) for channel A of unit 1 is available only in asynchronous counting mode (BCONR.SYNSA=1).

22.4.2 event output

A Timer0 contains 2 event outputs, which are the count comparison match event or capture input event for 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_Um_GCMn, m=1, 2; n=A, B) output signals are generated respectively, which can be used to selectively trigger other modules.

22.5 Register Description

Table 22-1 shows the register list of the Timer0 module.

BASE ADDR: 0x4002_4000 (U1) 0x4002_4400 (U2)

Table 22-1 Register List

register name	notation	offset	bit width	reset value
register for counting values	TMR0_CNTAR	0x0000	32	0x0000_0000
register for counting values	TMR0_CNTBR	0x0004	32	0x0000_0000
Reference Value Register	TMR0_CMPAR	0x0008	32	0x0000_FFFF
Reference Value Register	TMR0_CMPBR	0x000C	32	0x0000_FFFF
Basic Control Register	TMR0_BCONR	0x0010	32	0x0000_0000
Status flag register	TMR0_STFLR	0x0014	32	0x0000_0000

22.5.1 Count value register (TMR0_CNTAR)

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
CNTA[15:0]															
classifier marking celebrity functionality fill out or in (information on a form) for honorific people															
b31~b16	Reserved			b24	Reads out as "0."										
b15~b0	CNTA[15:0]			b24	Current timer count value										

22.5.2 Reference value register (TMR0_CMPAR)

Reset value: 0x0000_FFFF

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]															
classifier marking celebrity functionality fill out or in (information on a form) for honorific people															
b31~b16	Reserved			b24	Reads out as "0."										
b15~b0	CMPA[15:0]			b24	Setting the count base value to generate reference value										

22.5.3 Basic Control Register (TMR0_BCONR)

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17
HICP B	HCLE B	HSTP B	HSTA B	-	ASYN CLKB	SYN CLKB	SYN SB		CKDIV B[3:0]	-		INT ENB	CAP MDB	CST B
b16 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

classifier for marking				celebrity				functionality				fill out or in (information on a form)			
b31	HICPB	Hardware Trigger	Condition: Internal hardware trigger event is valid												R/W
		Input Capture B	0: Invalid capture input when condition matches 1: Capture input is valid when condition matches												
b30	HCLEB	Hardware Trigger	Condition: Internal hardware trigger event is valid												R/W
		Clear B	0: Invalid timer clear when condition matches 1: Timer clear valid when conditions match												
b29	HSTPB	Hardware-triggered stop B	Condition: Internal hardware trigger event is valid												R/W
		stop B	0: Timer stop invalid when condition matches 1: Timer stop is valid when the condition matches												
b28	HSTAB	Hardware-triggered startup B	Condition: Internal hardware trigger event is valid												R/W
		startup B	0: Invalid timer start when conditions match 1: Timer start is valid when conditions match												
b27	Reserved	-	Reads "0", writes "0".												R/W
			0: LRC												
b26	ASYNCLKB	Channel B Asynchronous Count Clock Source Selection	1: XTAL32 Note: When the timeout function of the corresponding channel of the UART is enabled, the clock source is no longer XTAL32, but the clock generated by the UART baud rate generator tick 0:												R/W
			1: Internal hardware triggered events												
b25	SYNCLKB	Channel B Synchronous Count Clock Source Selection	PCLK1												
			1: Internal hardware triggered events												
b24	SYNSB	Synchronized counting method	Channel B counting method selection	0: 1: Asynchronous counting method Channel B count clock division selection: 0000: clock source 0001: Clock source/2 0010: Clock source/4 0011: Clock source/8											0100: Clock source/16
			0: 1: Asynchronous counting method Channel B count clock division selection: 0000: clock source 0001: Clock source/2 0010: Clock source/4 0011: Clock source/8												

b23~b20	CKDIVB[3:0]	Channel B Count Clock Division Frequency Selection	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 clock source being divided can be various clock sources for asynchronous counting, synchronous counting PCLK1 at the time of	R/W
---------	-------------	---	--	-----

b19	Reserved	-Reserved	Read "0", write "0".	R/W
			0: When the CMPBR register is equal to the count value (CNTBR), or when a capture input occurs	
b18	INTENB Interrupt Enable B	Count Match	The interrupt is invalidated when the event 1: This interrupt is enabled when the CMPBR register is equal to the count value (CNTBR), or 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 activation	0: Channel B timer off 1: Channel B timer start Note: This bit will automatically change to 0 when a hardware-triggered stop condition is in effect	R/W
b15	HICPA	Hardware Trigger Input Capture A	Condition: Internal hardware trigger event is valid 0: Invalid capture input when condition matches 1: Capture input is valid when condition matches	R/W
b14	HCLEA	Hardware Trigger Clear A	Condition: Internal hardware trigger event is valid 0: Invalid timer clear when condition matches 1: Timer clear valid when conditions match	R/W
b13	HSTPA	Hardware-triggered stop A	Condition: Internal hardware trigger event is valid 0: Timer stop invalid when condition matches 1: Timer stop is valid when conditions match	R/W
b12	HSTAA	Hardware-triggered startup A	Condition: Internal hardware trigger event is valid 0: Invalid timer start when conditions match 1: Timer start is valid when conditions match	R/W
b11	Reserved	-	Reads "0" and writes "0".	R/W
b10	ASYNCLKA	Channel A Asynchronous Counting Clock source selection	0: LRC 1: XTAL32	R/W
b9	SYNCLKA	Channel A Synchronized Count Clock source selection	0: PCLK1 1: Internal hardware triggered events	R/W
b8	SYNSA	Channel A counting method selection	0: Synchronized counting mode 1: Asynchronous counting method	R/W
			Channel A Count Clock Division Selection: 0000: Clock Source 0001: Clock source/2 0010: Clock source/4 0011: Clock source/8 0100: Clock source/16	

b7~b4	CKDIVA [3:0]	Channel A Count	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 clock source being divided can be various clock sources for asynchronous counting, synchronous counting PCLK1 at the time of	R/W
b3	Reserved	-	Reads "0" and writes "0".	R/W
b2	INTENA	Count Match Interrupt Enable A	0: When the CMPAR register is equal to the count value (CNTBR), or when a capture input occurs 1: When the CMPAR register is equal to the count value (CNTBR), or when a capture input occurs	R/W

		event, this	
interrupt enables b1 selection A	CAPMDA 0: Compare output function	Function mode 1: Capture input function 0: Channel A timer off	R/W
b0	CSTA	Timer start 1: Channel A timer start Note: This bit will automatically change to 0	R/W

Attention: when a hardware-triggered stop condition is in effect

- The internal hardware trigger events (bit31~bit28 and bit15~bit12) and the XTAL32 clock source (bit26 and bit10) during asynchronous counting mentioned in this register are all provided as inputs by the USART module when the TIMEOUT function of the USART module is active, please refer to the USART chapter for details.

22.5.4 Status Flag Register (TMR0_STFLR)

Reset value: 0x0000_0000

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

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b17	Reserved	-	Reads "0", writes "0".	R
b16 and no capture input action has occurred	B	Count Match B	0: CMPBR register value is not equal to the count value 1: The value of the CMPBR register is equal to the count value or capture input action occurs	R/W
b15~b1	Reserved "0".	-0" for reading, "0" for writing. R	Read "0", write	
b0 and no capture input action has occurred	CMAF	Count Match A	0: CMPAR register value is not equal to the count value 1: The value of the CMPAR register is equal to the count value or capture input action occurs	R/W

22.6 Precautions for use

- 1) ASYNCLKA bit to select the asynchronous clock source, and then set BCONR.SYNSA bit to select the asynchronous counting method before starting Timer0.
- 2) With asynchronous counting selected, the count value (CNTAR), reference value (CMPAR) start bit (BCONR.CSTA) and status bit (STFLR.CMAF) for a write action, Timer0 passes 3 asynchronous counting clocks from the reception of the write action before writing the modified value into the corresponding register.

23 Real Time Clock (RTC)

23.1 brief

The Real Time Clock (RTC) is a counter that stores time information in BCD code format. Records specific calendar times from year 00 to year 99. It supports a 12/24 hour time system and automatically counts days 28, 29 (leap year) 30, and 31 based on the month and year. The basic features are shown in Table 23-1.

Table 23-1 Basic Specifications of RTC

Counting Clock Source	External low-speed oscillator (32.768KHz) RTC internal low-speed oscillator (32.768KHz)
Basic Functions	<ul style="list-style-type: none">- BCD code for seconds, minutes, hours, days, weeks, months, years time- Software start or stop- 12/24 hour selectable, leap year automatic recognition- Programmable Alarm Clock-Distributed/uniformly compensated 1Hz clock outputs-Clock Error Compensation Function Clock error compensation function
disruptions	cyclicalities are interrupted
	alarm clock

23.2 basic block diagram

The basic block diagram of the RTC is shown in Figure 23-1.

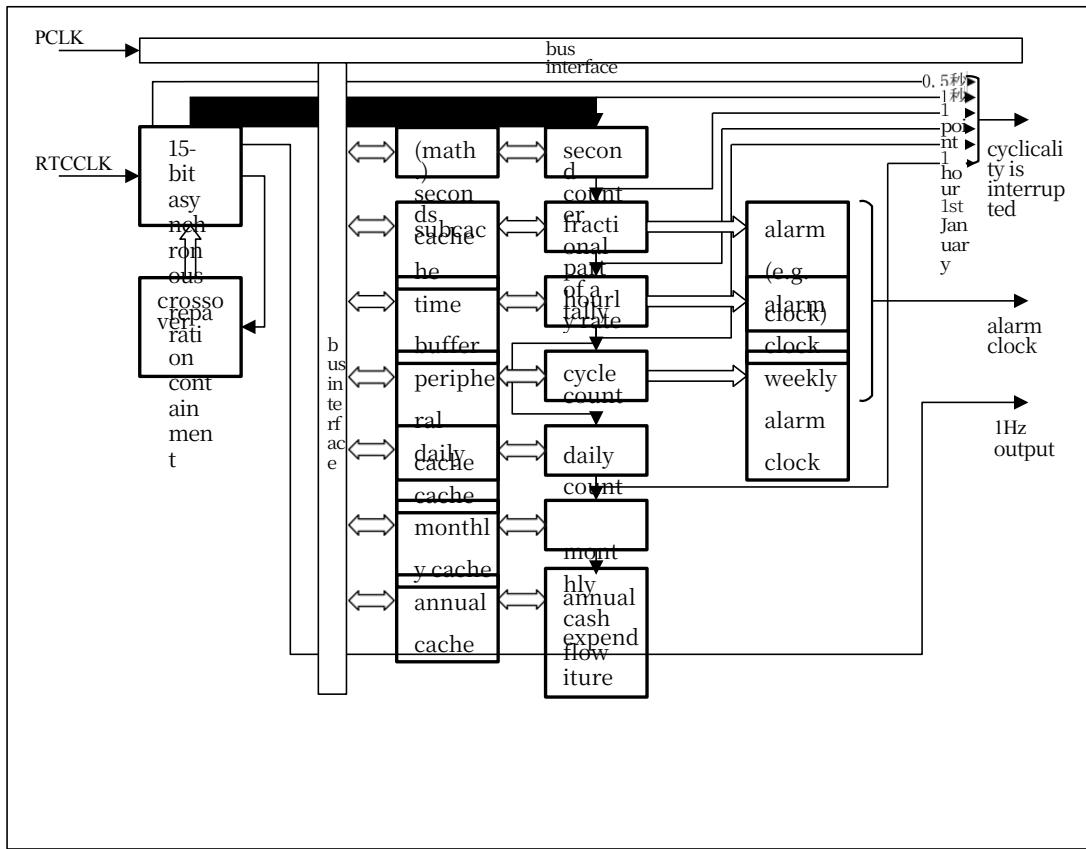


Figure 23-1 Basic Block Diagram of RTC

23.3 Functional Description

23.3.1 Power On Setting

The RTC register can be reset by power-on reset or by setting RTC_CR0.RESET to start the RTC after setting the control register, the initial value of the calendar, the alarm setting, etc. After the RTC is started, no other external reset requests can reset the RTC, and the RTC will be in the working state all the time. The START bit of the control register can be set to "0" to stop the RTC operation, and the clock source must be stabilized when the RTC is set.

23.3.2 RTC count start setting

- 1) After power on, all registers except RTC_CR0 will be reset; you can also set RTC_CR0.RESET=0, after confirming that the RESET bit is "0", set RTC_CR0.RESET=1 to reset all registers;
- 2) Set RTC_CR1.START=0 to stop counting;
- 3) Set the system clock register, turn on the secondary oscillator, and then set RTC_CR3 to select the RTC count clock source;
- 4) Set RTC_CR1 to set the time system, period, and 1Hz clock output;
- 5) Sets the calendar count registers for seconds, minutes, hours, days, weeks, months, and years;
- 6) When clock error compensation is required, set the count clock error compensation registers RTC_ERRCRL, RTC_ERRCRH;
- 7) Clears the flag register bit in register RTC_CR2 and enables the interrupt;
- 8) Set RTC_CR1.START=1 and counting starts.

23.3.3 System low-power mode switching

When the system switches to low-power mode immediately after RTC counting starts, perform any of the following confirmations before switching modes.

- 1) Mode switching is performed after 2 or more RTC count clocks after RTC_CR1.START=1 is set.
- 2) After RTC_CR1.START=1 is set, set RTC_CR2.RWREQ=1 and query RTC_CR2.RWEN=1. Set the calendar count register, then set RTC_CR2.RWREQ=0 and query RTC_CR2.RWEN=0 for mode switching.

23.3.4 Read Counter Register

- 1) After more than 2 RTC counting clocks after RTC_CR1.START=1 is set, set RTC_CR2.RWREQ=1 for calendar register read request;
- 2) Query until RTC_CR2.RWEN=1;
- 3) Reads all or part of the seconds, minutes, hours, days, weeks, months, and years count

register values;

- 4) Set RTC_CR2.RWREQ=0;
- 5) Query until RTC_CR2.RWEN=0.

23.3.5 Write Counter Register

- 1) After more than 2 RTC counting clocks after RTC_CR1.START=1 is set, set RTC_CR2.RWREQ=1 for a calendar register write request;
- 2) Query until RTC_CR2.RWEN=1;
- 3) Writes all or part of the seconds, minutes, hours, days, weeks, months, and years count register values;
- 4) Set RTC_CR2.RWREQ=0. Note that all write operations must be completed within 1 second;
- 5) Query until RTC_CR2.RWEN=0.

23.3.6 Alarm Setting

- 1) Set RTC_CR2.ALME=0, alarm disable;
- 2) Set RTC_CR2.ALMIE=1, alarm clock interrupt license;
- 3) Minute Alarm RTC_ALMMIN, Hour Alarm RTC_ALMHOUR, Week Alarm RTC_ALMEEK Setting;
- 4) Set RTC_CR2.ALME=1, alarm license;
- 5) Wait for an interruption to occur;
- 6) When RTC_CR2.ALMF=1, it enters the alarm clock interrupt processing.

23.3.7 Clock Error Compensation

Since there are errors in the operation of the external sub-firing oscillator crystal under various temperature conditions, it is necessary to compensate for the errors when high precision counting results are required. Refer to 23.5.15 Clock Error Compensation Registers (RTC_ERRCRH, RTC_ERRCRL)for the compensation method.

23.3.8 1Hz Output

The RTC can output 1Hz clock with three types of accuracy output, the first one is normal accuracy 1Hz output without clock error compensation, the second one is distributed compensation 1Hz output with average compensation every 32 seconds, and the third one is uniform compensation 1Hz output with compensation every second. Distributed compensation 1Hz output and uniform compensation 1Hz output can be selected when the clock error compensation function is valid RTC_ERRCRH.COMPEN=1. Among them

The 1Hz output setting for general accuracy is as follows:

- 1) Set RTC_CR0.RESET=0, confirm that the RESET bit is "0", and then set RTC_CR0.RESET=1 to reset.
Calendar Count Register;
- 2) Set RTC_CR1.START=0 to stop counting;

-
- 3) 1Hz Output pin setting;
 - 4) RTC_CR1.ONEHZOE=1, clock output licensed;
 - 5) Set RTC_CR1.START=1, counting starts;
 - 6) Wait for more than 2 counting cycles;
 - 7) 1Hz output starts.

The distributed compensation 1Hz output is set as follows:

- 1) Set RTC_CR0.RESET=0, confirm that the RESET bit is "0", and then set RTC_CR0.RESET=1 to reset.
Calendar Count Register;
- 2) Set RTC_CR1.START=0 to stop counting;
- 3) 1Hz Output pin setting;
- 4) RTC_CR1.ONEHZOE=1, clock output licensed;
- 5) Clock Error Compensation Registers RTC_ERRCRL.COMP[7:0] and RTC_ERRCRH.COMP[8] Compensation number Setting;
- 6) Clock error compensation register RTC_ERRCRH.COMPEN=1, error compensation is valid;
- 7) Set RTC_CR1.START=1 and counting starts;
- 8) Wait for more than 2 counting cycles;
- 9) 1Hz output starts.

The uniform compensation 1Hz output is set as follows:

- 1) Set RTC_CR0.RESET=0, confirm that the RESET bit is "0", and then set RTC_CR0.RESET=1 to reset.
Calendar Count Register;
- 2) Set RTC_CR1.START=0 to stop counting;
- 3) RTC output pin setting;
- 4) RTC_CR1.ONEHZOE=1, clock output licensed;
- 5) RTC_CR1.ONEHZSEL=1 to select the output uniformly compensated 1Hz clock;
- 6) Clock Error Compensation Registers RTC_ERRCRL.COMP[7:0] and RTC_ERRCRH.COMP[8] Compensation number Setting;
- 7) Clock Error Compensation Register RTC_ERRCRH.COMPEN=1, accuracy compensation is valid;
- 8) Set RTC_CR1.START=1, counting starts;
- 9) Wait for more than 2 counting cycles;
- 10) 1Hz output starts.

23.4 Interrupt Description

The RTC supports 2 types of interrupts. Timed Alarm Interrupt and Fixed Cycle Interrupt.

23.4.1 alarm clock

Alarm clock interrupt, RTC_ALM, is triggered if the current calendar time is equal to the minute alarm register (RTC_ALMMIN), hour alarm register (RTC_ALMHOUR), weekly alarm register (RTC_ALMWEEK), when ALMIE=1 in control register 2 (RTC_CR2) and ALME=1 in control register 2 (RTC_CR2). When the alarm register is equal to RTC_ALMWEEK, the alarm interrupt is triggered. ALMF, write "0" to RTC_CR1.ALMFCLR to clear the alarm flag. ALMFCLR bit to clear the alarm flag. The alarm interrupt can only be configured to the corresponding NVIC vector via INTC.INT_SELx (x=0~31,44~49).

23.4.2 fixed-cycle interrupt (computing)

Fixed-cycle interrupt RTC_PRD, with PRDIE=1 in control register 2 (RTC_CR2), triggers a fixed-cycle wake-up interrupt when the selected cycle occurs. There is no independent flag bit configured for fixed-cycle interrupt, you can select RTC_PRD interrupt to NVIC vector through INTC.INT_SELx (x=0~31,44~49) to query the corresponding flag bit of NVIC.

23.5 Register Description

Table 23-2 shows the register list of the

RTC module. Register base address:

0x4004_C000

Table 23-2 Register List

register name	notation	offset	bit width	reset value
Control Register 0	RTC_CR0	0x0000	8	(botany) adventitious
Control Register 1	RTC_CR1	0x0004	8	0x00
Control Register 2	RTC_CR2	0x0008	8	0x00
Control Register 3	RTC_CR3	0x000C	8	0x00
Seconds Count Register	RTC_SEC	0x0010	8	0x00
Minute counter register	RTC_MIN	0x0014	8	0x00
Hour Counter Register	RTC_HOUR	0x0018	8	0x12
Weekly 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
Clock Alarm Register	RTC_ALMHOUR	0x0030	8	0x12
Weekly Alarm Clock Register	RTC_ALMWEEK	0x0034	8	0x00
Clock Error Compensation Register	RTC_ERRCRH	0x0038	8	0x00
Clock Error Compensation Register	RTC_ERRCRL	0x003C	8	0x20

23.5.1 Control Register 0 (RTC_CR0)

Reset value: variable

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															

classifierfor honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b1	Reserved	-	Reads "0", writes "0". write state 0: Initialization register invalid 1: Initialization register valid Initialize all RTC registers. Read the status 0: Normal counting state or end of RTC software reset 1: RTC is in reset state Note: Make sure the bit is "0" before writing "1", otherwise it will not be initialized.	R/W
b0	RESETRTC Calendar Counter Reset			R/W

23.5.2 Control Register 1 (RTC_CR1)

Reset value: 0x00

b3	b3	b2	b2	b2	b2	b2	b23	b22	b21	b20	b19	b18	b17	b16
Reserved														
b1	b1	b1	b1	b1	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved					STA	ONEHZS	ONEHZ	ALMFC	AMP	PRDS[PRDS[PRDS[PRDS[PRDS[

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)			
b31~b8	Reserved	-	Reads "0", writes "0".	R/W			
b7	START	RTC count starts	0: RTC counting stops 1: RTC counting operation	R/W			
b6	ONEHZSEL	1Hz output selection	0: Distributed compensation 1Hz output 1: Uniformly compensated 1Hz output	R/W			
			Note: This bit is set valid when RTC_ERRCRH.COMPEN=1.				
b5	ONEHZOE	1Hz output enable	0: 1Hz output disable 1: 1Hz output license	R/W			
b4	ALMFCLR	ALMF flag clear	0: Clear the RTC_CR2.ALMF flag register 1: Invalid	R/W			
b3	AMPM	Time system selection	0: 12-hour time system 1: 24-hour clock	R/W			
Cycle Selection Settings:							
b2~0 Interrupt Selection	PRDS[2:0]	Cycle	PRDS [2]	PRDS[1]	PRDS[0]	periodicity	R/W
			0	0	0	non-selectivity	
			0	0	1	Every 0.5 second cycle	
			0	1	0	Every 1 second cycle	
			0	1	1	Per 1-minute cycle	
			1	0	0	Per 1-hour cycle	
			1	0	1	Every 1-day cycle (daily 00:00 00 seconds)	
			1	1	X	Every 1-month cycle (00:00 on the 1st of the month)	
Note: When writing the cycle selection during START=1 counting, turn off the cycle interrupt license to prevent false operation. Also, the relevant flag bit should be cleared after writing.							

23.5.3 Control Register 2 (RTC_CR2)

Reset value: 0x00

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								ALME	ALMIE	PRDIE	-	ALMF	-	RWEN	RWREQ

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b8	Reserved	-	Reads "0" and writes "0".	R/W
b7	ALME	Alarm Clock Enable	0: Alarm clock function disabled 1: Alarm clock function license	R/W
b6	ALMIE	Alarm clock interrupt enable clock interrupt disable	R/W 0: Alarm 1: Alarm interruption license 0: Cycle	
b5	PRDIE	Cycle interrupt enable interrupt disable	1: Cycle interrupt license	R/W
b4	Reserved	-Reserved	Read "0", write "0".	R/W
b3	ALMF	Alarm Clock Flag	0: Alarm clock mismatch 1: Alarm Matching	R
b2	Reserved	-	-Indeterminate when reading, write "0"	when writing.
b1	RWEN	Read/Write Allowed	R/W 0: Read/write disabled 1: Read/write allowed Note: The calendar register read/write allow flag. Make sure this bit is "1" before reading/writing. The calendar registers include the second, minute, hour, week, day, month, and year count registers.	R/W
b0	RWREQ	Read/Write Request	0: Normal counting mode 1: Read/write requests Note: When reading/writing the calendar register, please set this bit to "1" to request a read/write. Since the counter is counting continuously, please complete the read/write operation within 1 second and clear this bit to "0".	R/W

23.5.4 Control Register 3 (RTC_CR3)

Reset value: 0x00

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								RCKSEL	-	-	LCREN	-	-	-	-

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b8	Reserved	-	Reads "0", writes "0".	R/W 0: Select
b7 XTAL32	RCKSELRTC count clock selection			
	R/W	1: Select LRC		
b6	Reserved	-0" for reading, "0" for writing.	Reads "0", writes "0".	R/W
b5	Reserved	-0" for reading, "0" for writing.	Reads "0", writes "0".	R/W
	0: Low-speed oscillator stop			
b4 oscillator enable	LCREN	Low-speed	1: Low Speed Vibrator Operation Note: The operation and stopping of the low-speed oscillator is determined by the clock control circuit and any of the LRCEN	R/W
	settings. Set the LRCEN bit to enable when the low-speed oscillator is used as the RTC clock source.			
b3~b1	Reserved	-Reserved	Read "0", write "0".	R/W
b0	Reserved	-Reserved	Reads "0", writes "0".	R/W

23.5.5 Seconds Count Register (RTC_SEC)

Reset value: 0x00

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								-	SECD[2:0]	SECU [3:0]					

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b7	Reserved	-	Reads "0", writes "0".	R/W
b6~b4	SECD[2:0]	the tens place (or column) in the decimal system	Decimal value in seconds	R/W
b3~b0	SECU [3:0]	second digit	Seconds bitmeter value	R/W

Indicates 0-59 seconds with decimal counting. Please write the BCD code of decimal 0-59, and the written value will be ignored when an error value is written.

23.5.6 Minute Count Register (RTC_MIN)

Reset value: variable

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				-	MIND[2:0]				MINU [3:0]						

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b7	Reserved	-	Reads "0", writes "0".	R/W
b6~b4	MIND[2:0]	decile	decimal value	R/W
b3~b0	MINU [3:0]	the units place (or column) in the decimal system	sub-quantile value	R/W

Indicates 0-59 points with decimal counting. Please write the BCD code of decimal 0-59, the written value will be ignored when writing the wrong value.

23.5.7 Time Counter Register (RTC_HOUR)

Reset value: 0x12

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								-	-	HOURD[1:0]	HOURU [3:0]				

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b6	Reserved	-	Reads "0", writes "0".	R/W
b5~b4	HOURD[1:0]	the Tokens place (or column) in the decimal system	time decimal value	R/W
b3~b0	HOURU [3:0]	hourly rate	time-periodic value	R/W

In 24-hour time system, it means 0-23 hours. In 12-hour time system, b5=0 means AM, then 01~12 means morning; b5=1 means PM, then 21~32 means afternoon.

Please set the BCD code of 0~23 or 01~12,21~32 in correct decimal according to the value of control bit AMPM. Write values out of range will be ignored.

Refer to the table below for specific time indications:

24-hour time system	AMPM=1	12-hour clock	AMPM=0
timing	register representation	timing	register representation
00 hours	00H	AM 1200 hours	12H
01 hour	01H	AM 0100 hours	01H
02 hours	02H	AM 0200 hours	02H
03 hours	03H	AM 0300 hours	03H
04 hours	04H	AM 0400 hours	04H
0500 hours	05H	AM 0500 hours	05H
0600 hours	06H	AM 0600 hours	06H
0700 hours	07H	AM 0700 hours	07H
0800 hours	08H	AM 0800 hours	08H
09 o'clock	09H	AM 0900 hours	09H
10 o'clock	10H	AM 10 a.m.	10H
11 o'clock	11H	AM 11 a.m.	11H
12 o'clock	12H	PM 1200 hours	32H
1300 hours	13H	PM 01 hrs.	21H
1400 hours	14H	PM 0200 hours	22H
1500 hours	15H	PM 0300 hours	23H
1600 hours	16H	PM 0400 hours	24H
1700 hours	17H	PM 0500 hours	25H
1800 hours	18H	PM 0600 hours	26H
1900 hours	19H	PM 0700 hours	27H
2000 hours	20H	PM 0800 hours	28H
2100 hours	21H	PM 0900 hours	29H
2200 hours	22H	At PM 10	30H
2300 hours	23H	PM 1100 hours	31H

23.5.8 Day Count Register (RTC_DAY)

Reset value: 0x00

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								-	-	DAYD[1:0]	DAYU [3:0]				

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b6	Reserved	-	Reads "0", writes "0".	R/W
b5~b4	DAYD[1:0]	the sun tens place (or column) in the decimal system	daily decimal value	R/W
b3~b0	DAYU [3:0]	the sun on a single digit	daily value of a single digit	R/W

Decimal representation of 1~31 days, automatic calculation of leap years and months. The specific representations are as follows:

months	day count representation
February (ordinary year)	01~28
February (leap year)	01~29
April, June, September, November	01~30
January, March, May, July, August, October, December	01~31

23.5.9 Week Count Register (RTC_WEEK)

Reset value: 0x00

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								-	-	-	-	-	-	WEEK[2:0]	

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b3	Reserved	-	Reads "0" and writes "0".	R/W
b2~b0	WEEK[2:0]	weekly	weekly value	R/W

Decimal 0~6 means Sunday~Saturday. Please write the correct BCD code for decimal 0~6. Writing other values will be ignored. The correspondence of the weekly counting values is as follows:

weekly	weekly count
diurnal	00H
on Monday	01H
Tuesday	02H
wednesday	03H
thurs	04H
Fridays	05H
Saturdays	06H

23.5.10 Month Count Register (RTC_MON)

Reset value: 0x00

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															

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b5	Reserved	-	Reads "0", writes "0".	R/W
b4~b0	MON[4:0]	moon	Monthly values	R/W

Decimal 1~12 means 1~12 months. Please write the correct BCD code for decimal 1~12, other values will be ignored.

23.5.11 Year Count Register (RTC_YEAR)

Reset value: 0x00

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															

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b8	Reserved	-	Reads "0" and writes "0".	R/W
b7~b4	YEARD [3:0]	the top ten in the year	Annual decile value	R/W
b3~b0	YEARU[3:0]	the single digit of the year (i.e. number of years since the last decimal point)	Yearly value of individual quartiles	R/W

Decimal 0~99 indicates 0~99 years. According to the month rounding counting. Calculate leap year automatically such as: 00, 04, 08, ..., 92, 96, etc. Please write the correct decimal year count value, wrong value will be ignored.

23.5.12 Minute alarm clock register (RTC_ALMMIN)

Reset value: 0x00

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				-	-	ALMMIND[2:0]	-	ALMMINU [3:0]							

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b7	Reserved	-	Reads "0", writes "0".	R/W
b6~b4	ALMMIND[2:0]	Minute Alarm Clock Ten	Split Alarm Clock Ten Digit Match Value	R/W
b3~b0	ALMMINU [3:0]	alarm clock	Split Alarm Clock Digit Match Value	R/W

Please set the BCD code of decimal 0~59. Writing other values will not result in an alarm match.

23.5.13 Clock alarm register (RTC_ALMHOUR)

Reset value: 0x12

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				-	-	-	-	-	-	-	-	-	-	-	-

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b6	Reserved	-	Reads "0", writes "0".	R/W
b5~b4	ALMHOURD[1:0]	Hourly Alarm Clock Ten	Hourly Alarm Clock Ten Digit Match Value	R/W
b3~b0	ALMHOURU [3:0]	local time alarm	Hourly Alarm Clock Digit Match	R/W

Please set the correct alarm match value according to the time system, otherwise the time alarm match will not occur.

23.5.14 Weekly alarm clock register (RTC_ALMWEEK)

Reset value: 0x00

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										-	ALMWEEK[6:0]				

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b7	Reserved	-	Reads "0" and writes "0".	R/W
b6~b0	ALMWEEK[6:0]	weekly alarm clock	Weekly alarm clock match value. b0~b6 corresponds to Sunday~Saturday respectively, and when the corresponding value is set to "1", it means the day of the week. The clock is valid. For example, b0=1 and b5=1 means that the Sunday and Friday alarm settings are valid.	R/W

Please set the correct alarm match value according to the time system, otherwise the time alarm match will not occur.

23.5.15 Clock error compensation registers (RTC_ERRCRH, RTC_ERRCRL)

Reset value: 0x0000_0020

RTC_ERRCRH

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										COMPEN	-	-	-	-	COMP[8]

RTC_ERRCRL

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										COMP[7:0]					

RTC_ERRCRH

classifier	marking	celebrity	functionality	fill out or in (information on a form)
for honorific people				
b31~b8	Reserved	-	Reads "0", writes "0".	R/W
b7	COMPEN	Compensation Enable	0: Invalid clock error compensation 1: Clock error compensation effective	R/W
B6~b1	Reserved	-	Reads "0" and writes "0".	R/W
b0	COMP[8]	compensation value	Set the compensation value together with COMP[7:0].	R/W

RTC_ERRCRL

classifier	marking	celebrity	functionality	fill out or in for honorific people																																																																																																																								
b31~b8	Reserved	-	Reads "0" and writes "0".	R/W																																																																																																																								
b7~b0	COMP[7:0]	compensation value	<p>The compensation value can be set to compensate for an accuracy of +/-0.96 ppm per second. The compensation value is a 9-digit 2's complement code with a decimal point, and the last 5 digits are the decimal portion. Compensable range -275.5ppm to +212.9ppm. Minimum resolution 0.00001. Refer to the following table for specific compensation accuracy:</p> <table border="1"> <tr><td>~</td><td>~</td><td>~</td><td>~</td><td>~</td><td>~</td><td>~</td><td>~</td><td>~</td><td>~</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>-30.5ppm</td></tr> <tr><td>~</td><td>~</td><td>~</td><td>~</td><td>~</td><td>~</td><td>~</td><td>~</td><td>~</td><td>~</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>-0.96ppm</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0ppm</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>+0.96ppm</td></tr> <tr><td>~</td><td>~</td><td>~</td><td>~</td><td>~</td><td>~</td><td>~</td><td>~</td><td>~</td><td>~</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>+30.5ppm</td></tr> <tr><td>~</td><td>~</td><td>~</td><td>~</td><td>~</td><td>~</td><td>~</td><td>~</td><td>~</td><td>~</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>+212.0ppm</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>+212.9ppm</td></tr> <tr><td>0</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>null</td></tr> </table>	~	~	~	~	~	~	~	~	~	~	0	0	0	0	0	0	0	0	0	-30.5ppm	~	~	~	~	~	~	~	~	~	~	0	0	0	0	1	1	1	1	1	-0.96ppm	0	0	0	1	0	0	0	0	0	0ppm	0	0	0	1	0	0	0	0	1	+0.96ppm	~	~	~	~	~	~	~	~	~	~	0	0	1	0	0	0	0	0	0	+30.5ppm	~	~	~	~	~	~	~	~	~	~	0	1	1	1	1	1	1	1	0	+212.0ppm	0	1	1	1	1	1	1	1	1	+212.9ppm	0	X	X	X	X	X	X	X	X	null	R/W
~	~	~	~	~	~	~	~	~	~																																																																																																																			
0	0	0	0	0	0	0	0	0	-30.5ppm																																																																																																																			
~	~	~	~	~	~	~	~	~	~																																																																																																																			
0	0	0	0	1	1	1	1	1	-0.96ppm																																																																																																																			
0	0	0	1	0	0	0	0	0	0ppm																																																																																																																			
0	0	0	1	0	0	0	0	1	+0.96ppm																																																																																																																			
~	~	~	~	~	~	~	~	~	~																																																																																																																			
0	0	1	0	0	0	0	0	0	+30.5ppm																																																																																																																			
~	~	~	~	~	~	~	~	~	~																																																																																																																			
0	1	1	1	1	1	1	1	0	+212.0ppm																																																																																																																			
0	1	1	1	1	1	1	1	1	+212.9ppm																																																																																																																			
0	X	X	X	X	X	X	X	X	null																																																																																																																			

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 this clock. Assume that the actual measured value is 0.9999888Hz:

$$\text{Actual oscillation frequency} = 32768 \times 0.9999888 \approx 32767.63$$

$$\text{Compensation target value} = (\text{actual oscillation frequency} - \text{target frequency})/\text{target frequency} \times 10^6$$

$$= (32767.96 - 32768) / 32768 \times 10^6 \\ = -11.29 \text{ ppm}$$

setpoint calculation:

$$\text{COMP}[8:0] = \left(\frac{\text{Compensation target value [ppm]} \times 2^{15}}{10^6} \right) + 0001.00000B$$

Taking the 2's complement

If the compensation target value is +20.3 ppm, calculate the corresponding register value as follows:

$$\text{COMP[8:0]} = (20.3 \times 215/106) \text{ times 2's complement} + 0001.00000\text{B}$$

$$\begin{aligned}&= (0.6651904) \quad \text{Take 2's complement} + 0001.00000B \\&= 0000.10101B + 0001.00000B \\&= 0001.10101B\end{aligned}$$

If the compensation target value is -20.3 ppm, calculate the corresponding register value as follows:

$$\begin{aligned}\text{COMP[8:0]} &= (-20.3 \times 215/106) \text{ times 2's complement} + 0001.00000B \\&= (-0.6651904) \text{ times 2's complement} + 0001.00000B \\&= 1111.01011B + 0001.00000B \\&= 0000.01011B\end{aligned}$$

24 Watchdog Counter (WDT/SWDT)

24.1 brief

There are two types of watchdog counters, one is a dedicated watchdog counter with a dedicated internal RC (SWDTLRC:10KHz) as the counting clock source.

(SWDT) and the other is a general purpose watchdog counter (WDT) with a count clock source of PCLK3. The Dedicated Watchdog and General Purpose Watchdog are 16-bit decrementing counters that are used to monitor for software failures due to deviations from normal operation of the application program caused by external disturbances or unforeseen logic conditions.

Both watchdogs support the window function. The window interval can be preset before counting starts. When the count value is located in the window interval, the counter can be refreshed and counting starts again. The basic features are shown in Table 24-1.

Table 24-1 Basic Watchdog Counter Characteristics

tachymeter clock	SWDT: 1/16/32/64/128/256/2048 divisions of SWDTLRC WDT: 4/64/128/256/512/1024/2048/8192 divisions of PCLK3
Maximum overflow time	SWDT:3.72hour(max) WDT:10.7s (PCLK3=50MHz)
counting mode	count down
window function	Window interval can be set to define the allowable interval for refresh action
activation method	1) hardware startup 2) Software startup (SWDT does not support software startup)
stop condition	1) reset 2) Underflow, refresh error occurs when Restart: Hardware startup mode, automatic start after reset or interrupt request output Setting the refresh register again in software startup mode
Interrupt/reset conditions	1) underflow 2) refresh error

24.2 Functional Description

24.2.1 start the watchdog

There are two ways to start the watchdog counter: hardware startup and software startup. (SWDT only supports hardware startup)

Hardware startup method means reading the setting information of the watchdog counter (ICG0 register) from the main flash memory area at startup, and the counter starts counting automatically; software startup method means setting the control registers and then writing the refresh register to complete the refresh action, and the counter starts counting.

24.2.2 hardware boot method

When bit 16 (WDTAUTS), bit 0 (SWDTAUTS) of ICG0 register is 0, it is hardware startup mode. When the hardware startup mode is selected, the related setting information of WDT_CR register is invalid.

In the hardware startup method, the WDT/SWDT-related settings (count clock, window setting value, count period, etc.) in the ICG0 register are loaded into the WDT/SWDT module during the reset period, and the counter starts counting automatically according to the settings after the reset. Figure 24-1 shows an example of the hardware startup method.

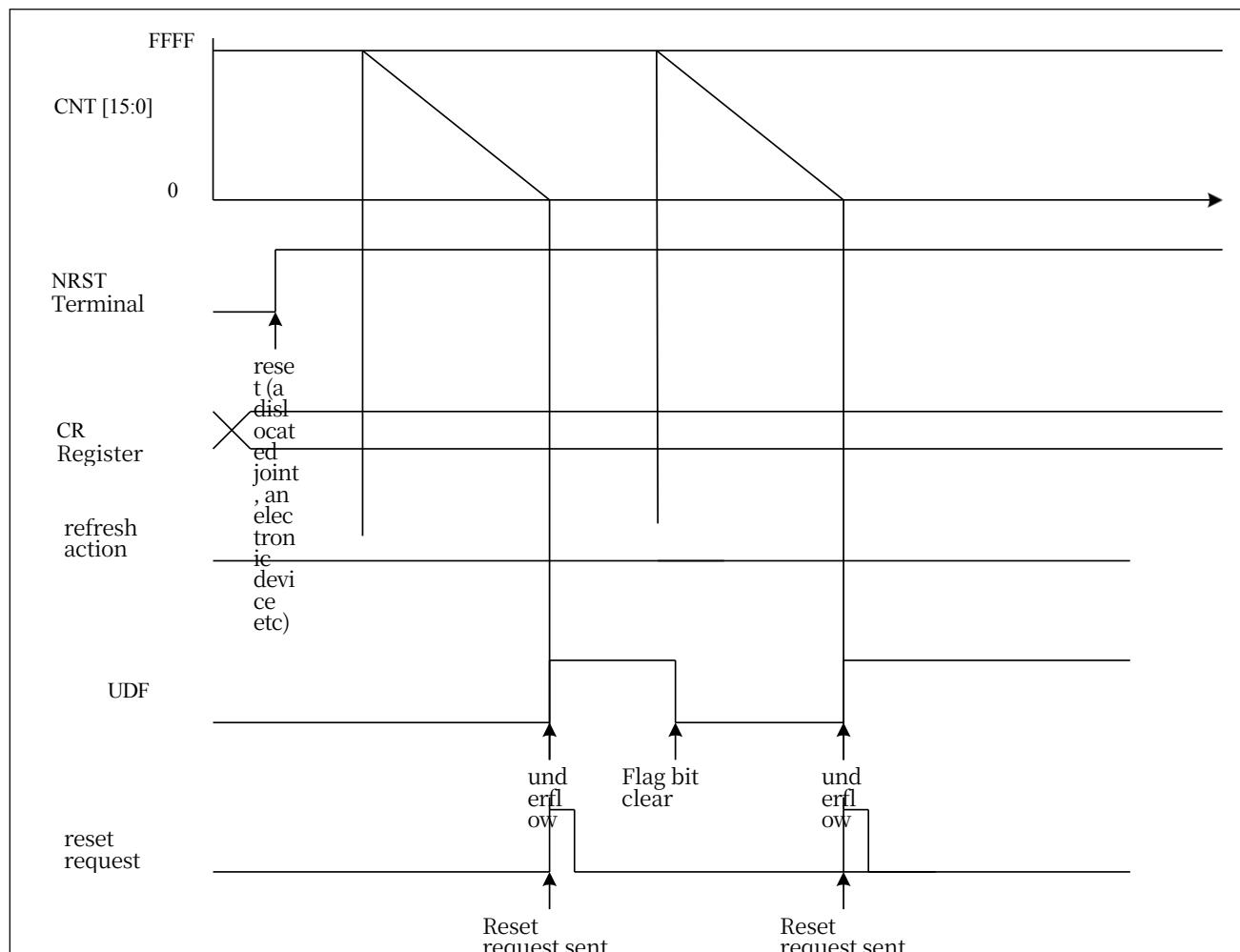


Figure 24-1 Hardware Startup Example

24.2.3 Software startup method

If bit 16 (WDTAUTS) of ICG0 register is 1, WDT startup by setting the refresh register is the software startup method. After reset, set the counting clock, window setting value, counting period, etc. in the WDT_CR register, and then execute the refresh action, and the counter starts counting. The WDT_CR setting can be done only once, and writing the value again is invalid. Figure 24-2 shows an example of software startup mode.

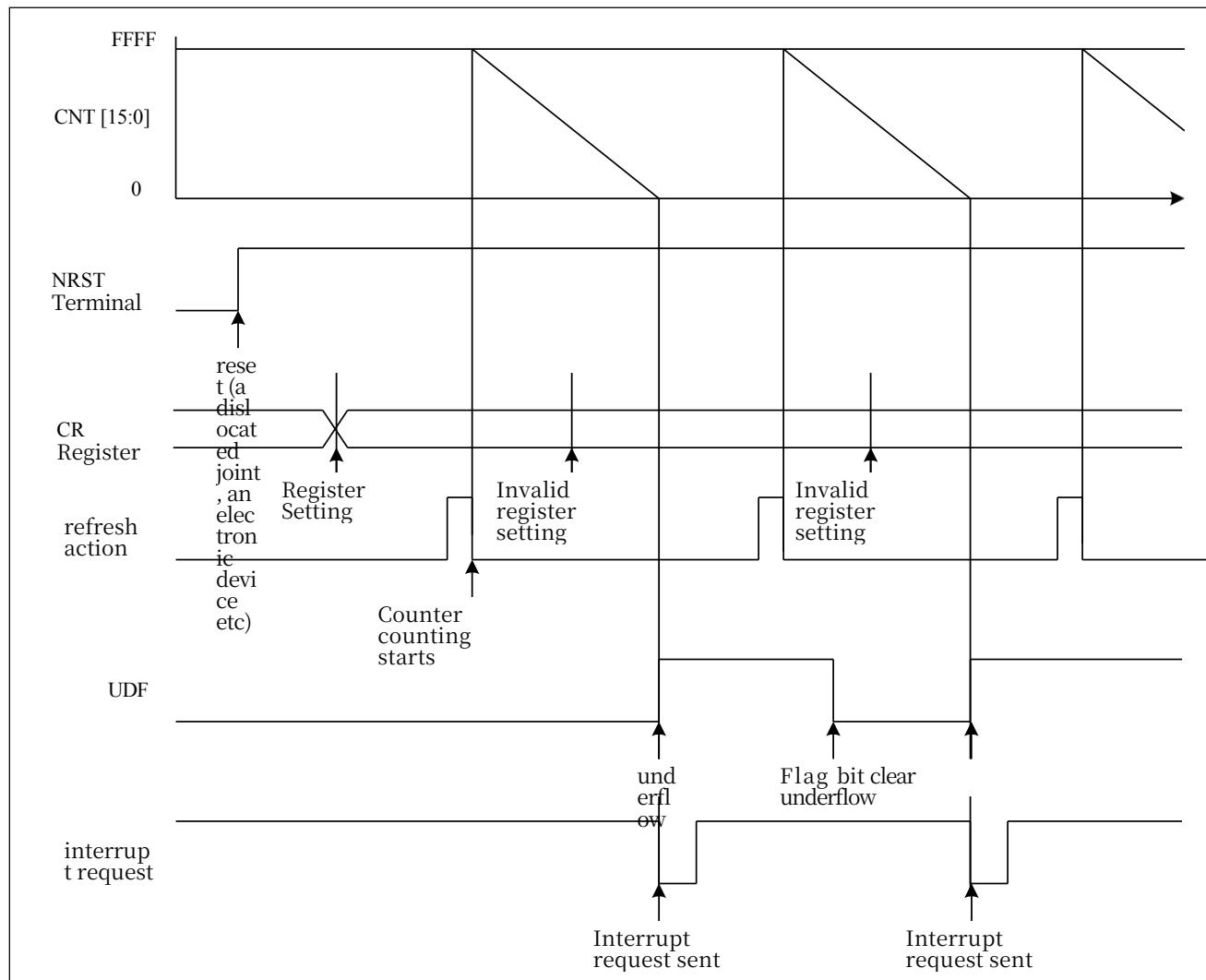


Figure 24-2 Software Startup Example

24.2.4 refresh action

(S) Write 0x0123 and then 0x3210 in the WDT_RR register to complete a refresh action, and the counter of the WDT/SWDT starts counting (software startup) or restarts counting.

(S) WDT_RR register between writing 0x0123,0x3210, if the pair occurs to other registers access or read (S)WDT_RR register, etc., does not affect the normal refresh action.

An example of the action is shown in Figure 24-3.

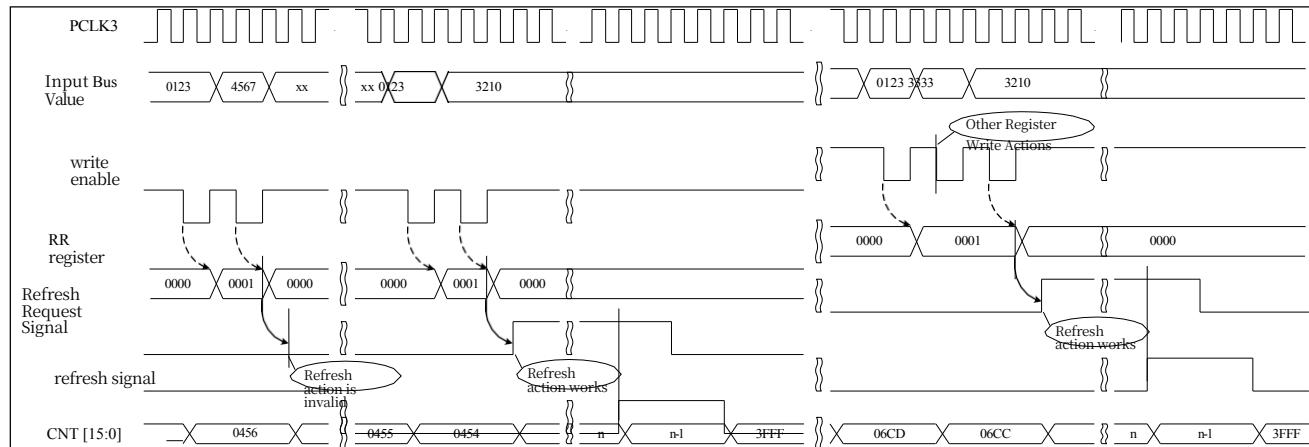


Fig. 24-3 Examples of timing of various refresh actions (action confirmation, falling edge of refresh request signal, etc.)

The refresh action requires 4 counting cycles to complete updating the count value, so please write the refresh register 4 counting cycles in advance of refreshing the lower window and underflow position. To confirm the count value, read the status register.

24.2.5 flag position

The Refresh Error Flag Bit and Count Underflow Flag Bit are held in the case of both reset requests and interrupt requests. When a reset is released or an interrupt is entered, the reset or interrupt cause can be confirmed by querying the flag bits.

Flag bit clear: read "1" then write "0".

The hardware boot mode watchdog count does not stop when there is a refresh error or the count underflow flag is set. For SWDT, when writing "0" to the flag bit, it takes at most 3 SWDTLRC and 2 PCLK3 times before the register bit is cleared to zero. For WDT, when writing "0" to the flag bit, it takes up to 5 PCLK3 times before the register bit is cleared. In addition, reading a "1" to clear a flag bit is not valid until a refresh error or underflow error occurs within a certain period of time: 1 count cycle + 2 SWDTLRC (SWDT module); 1 count cycle + 2 PCLK3 (WDT module).

24.2.6 interrupt reset (computing)

SWDT You can choose to generate an interrupt request or a reset request when the counter count overflows or is refreshed incorrectly. The decision to generate an interrupt request or a

reset request is made by setting the SWDTITS bit of ICG0.

The WDT has the option of generating an interrupt request or a reset request in the event of a counter count underflow or refresh error. The decision to generate an interrupt request or a reset request is made by setting the WDTITS bit of ICG0 during hardware startup and the WDT_CR register ITS bit during software startup.

There are two interrupt reset generation conditions for WDT/SWDT. One is when the counter count generates an underflow; the other is when a refresh error is generated by performing a refresh action outside the refresh allowable interval.

24.2.7 underflow

As in the example in Figure 24-4, an underflow is generated when the decrementing count reaches zero.

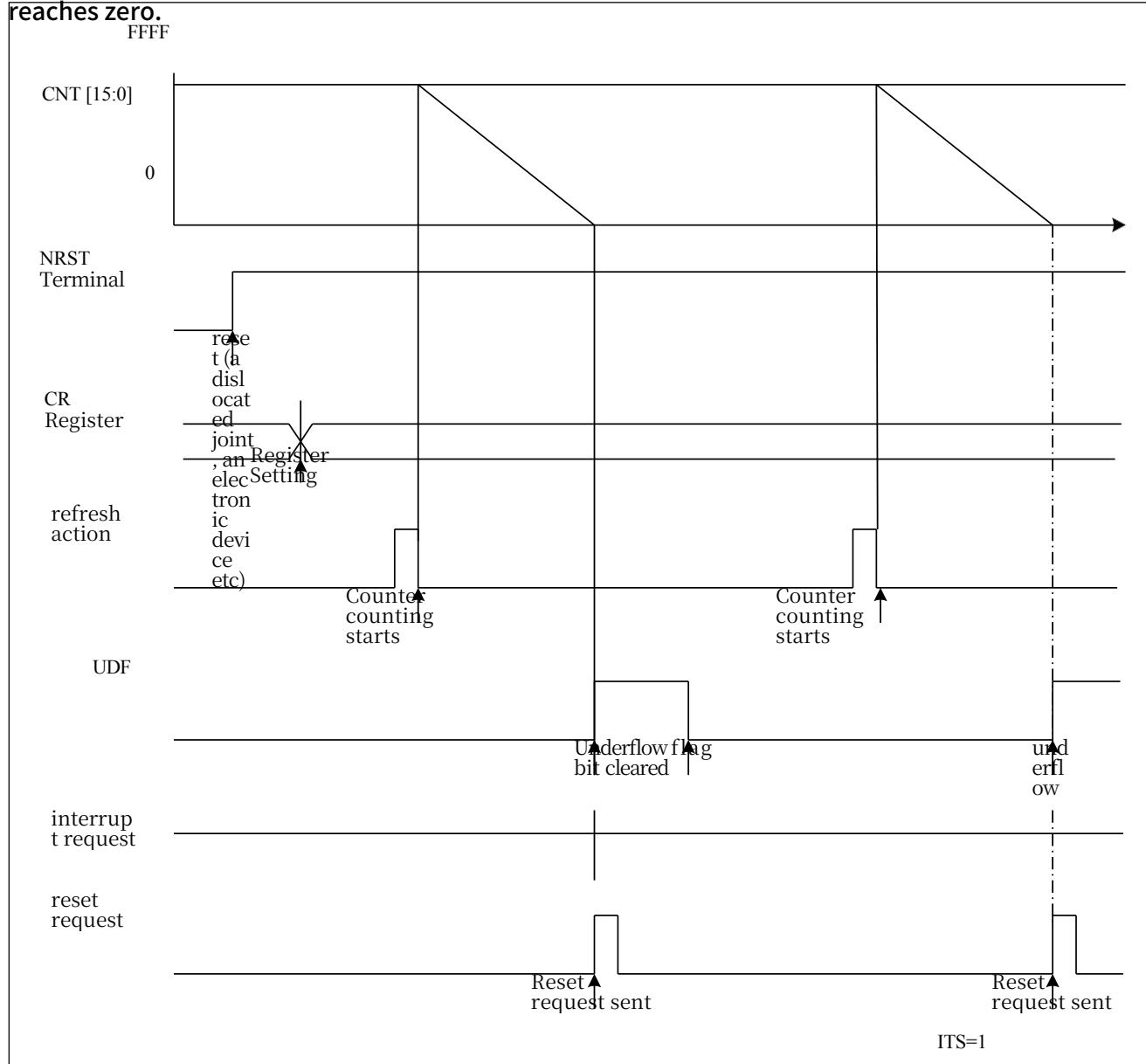


Figure 24-4 Counter Overflow Example

24.2.8 refresh error

When the window interval is set, the counter is refreshed and counting starts again only when the refresh action is performed within the window interval, and a refresh error occurs when the refresh action is performed outside the window interval. Figure 24-5 shows an example of a refresh action.

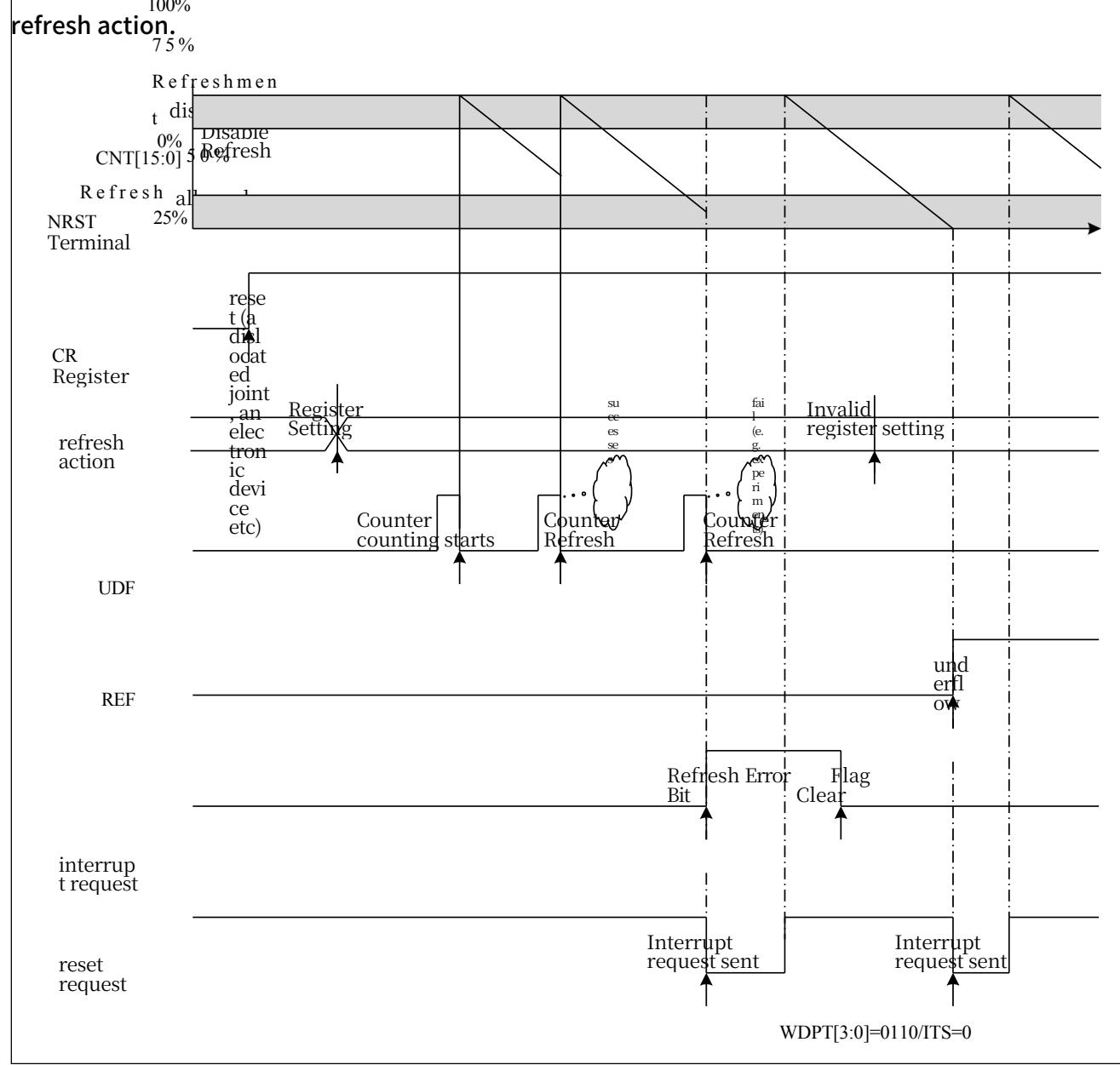


Figure 24-5 Counter Refresh Action Example

24.3 Register Description

Table 24-2 shows the register list for the SWDT

and WDT modules. WDT_BASE_ADDR:

0x4004_9000 SWDT_BASE_ADDR:0x4004_9400

Table 24-2 Register List

register name	notation	offset address	bit width	reset value
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

24.3.1 Control Register (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[3:0]		CKS[3:0]		-	-	PERI[1:0]					

classifier for honorific people		marking	celebrity	functionality	fill out or in (information on a form)
b31		ITS	Refresh error/overflow Interrupt/Reset Selection	0: Interrupt request 1: Reset request	R/W
b30~b17		Reserved	-	Reads "0", writes "0".	R
b16		SLPOFF	WDT in Sleep mode lowercase counting prohibition	0: WDT counting license in sleep mode 1: WDT counting disabled in sleep mode	R/W
b15~b13		Reserved	-	Reads "0", writes "0".	R
b11~b8		WDPT[3:0]	Refresh Allowable Area Meter Numerical percentage	0000: 0% to 100% 0001: 0% to 25% 0010: 25% to 50% 0011: 0% to 50% 0100: 50% to 75% 0101: 0%~25%, 50%~75% 0110: 25% to 75% 0111: 0% to 75% 1000: 75% to 100% 1001: 0% to 25%, 75% to 100% 1010: 25% to 50%, 75% to 100% 1011: 0% to 50%, 75% to 100% 1100: 50% to 100% 1101: 0% to 25%, 50% to 100% 1110: 25% to 100% 1111: 0% to 100%	R/W
b7~b4		CKS[3:0]	tachymeter clock	0010: PCLK3/4 0110: PCLK3/64 0111: PCLK3/128 1000: PCLK3/256 1001: PCLK3/512 1010: PCLK3/1024 1011: PCLK3/2048 1101: PCLK3/8192	R/W

Remaining values: reserved functions				
b3~b2	Reserved	-	Reads "0", writes "0".	R
b1~b0	PERI[1:0]	duty cycle	00: 256 cycles 01: 4096 cycle 10: 16384 cycle 11: 65536 cycle	R/W

24.3.2 Status registers (SWDT_SR, WDT_SR)

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
CNT [15:0]															

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b18	Reserved	-	Reads "0", writes "0".	R
b17	REF	Refresh Error Flag	0: No refresh error 1: Refresh error occurred Reading a 1 to this bit and then writing a 0 clears the bit.	R/W
b16	UDF	underflow flag	0: no count overflow 1: Count overflow occurs Reading a 1 to this bit and then writing a 0 clears the bit.	R/W
b15~b0	CNT [15:0]	numerical value	Counter current count value	R

24.3.3 Flush registers (SWDT_RR, WDT_RR)

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
RF[15:0]															

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b16	Reserved	-	Reads "0" and writes "0".	R
b15~b0	RF[15:0]	refresh value	After writing 0x0123 and 0x3210 sequentially, the refresh action is completed. When the register is written to 0x0123, the read register is 0x0000_0001; the rest of the cases The readout values are all 0x0000_0000.	R/W

24.4 Precautions for use

When SWDT is actuated, the action frequency of the peripheral clock PCLK3 must be greater than or equal to 4 times of the count clock (ICG0.SWDTCKS[3:0] sets the clock after SWDTLRC frequency division), i.e., the PCLK3 frequency \geq the count clock frequency $\times 4$.

25 Universal Synchronous Asynchronous Transceiver (USART)

25.1 summary

This product is equipped with 4 units of Universal Serial Transceiver Module (USART). The Universal Serial Transceiver Module (USART) enables flexible full-duplex data exchange with external devices; the USART supports Universal Asynchronous Serial Interface (UART) Clock Synchronous Communication Interface, and Smart Card Interface (ISO/IEC7816-3). It supports modem operation (CTS/RTS operation)and multiprocessor operation. Supports UART receive TIMEOUT function with Timer0 module.

USART Main Features:

- Supports full-duplex asynchronous communication, full-duplex clock synchronous communication
- Transmitter and receiver have independent enable bits
- Built-in double buffer
- LSB/MSB optional
- Supports modem operation (CTS/RTS)
- Transmission flags: transmit data register empty, transmit data complete, receive data register full, receive error flag, UART receive timeout flag

UART Key Features:

- Programmable data length: 8 bits/9 bits
- Configurable checksum function: odd/even/no checksum
- Stop bit configurable: 1 bit/2 bits
- Clock source selectable: Internal clock source (clock generated by internal baud rate generator)/External clock source (clock input from USARTn_CK pin)
- Receive errors: checksum errors, framing errors, overflow errors
- Supports multiple inter-processor communication
- Built-in digital filter
- Supports receiving data TIMEOUT function
- Unit 1 Wake-up from Stop Mode

Clock Synchronization Mode Main Features:

- Data length: 8 bits
- Receive error: overflow error
- Clock source: Internal clock source (clock generated by internal baud rate generator)/External clock source (clock input from USARTn_CK pin)

Key features of the smart card interface:

- Data length: 8 bits
- Automatic error signaling when a calibration error is detected
- Support for data retransmission

25.2 USART System Block Diagram

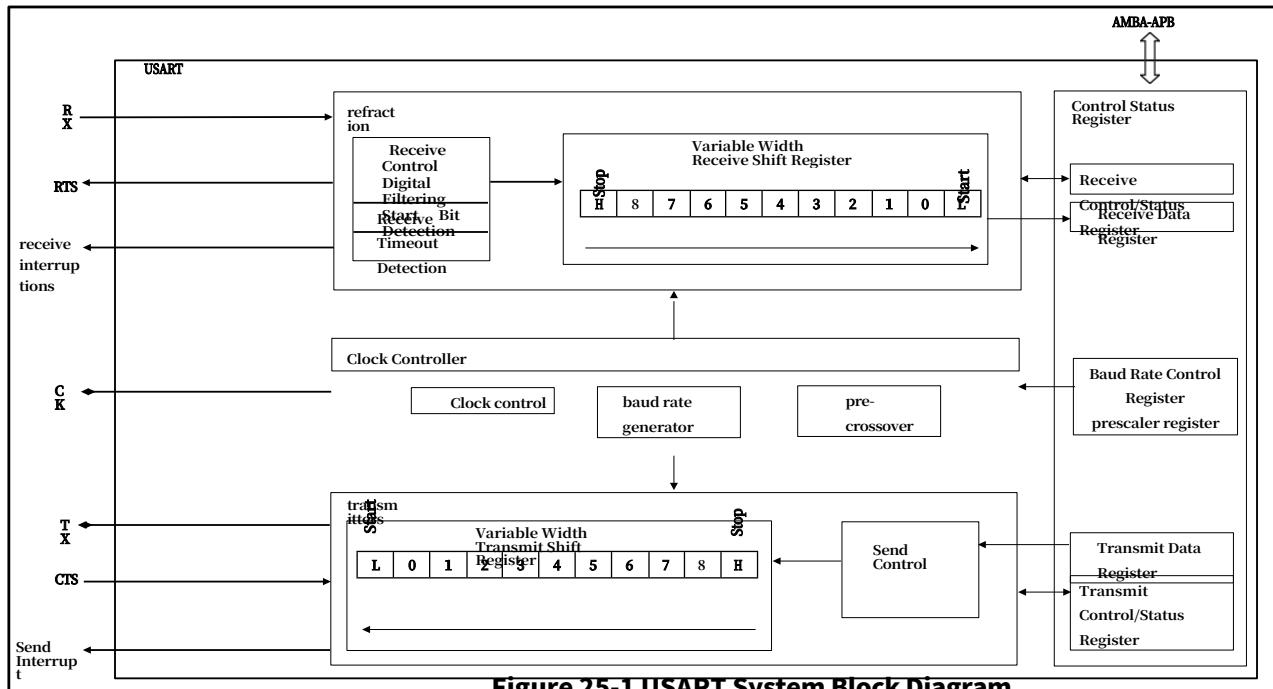


Figure 25-1 USART System Block Diagram

25.3 Pin

Descrip
tion

Table 25-1 USART Pin Descriptions

pin name	directional	Functional Description
USARTn_CK	input and output	clocks
USARTn_TX	exports	Send Data Pin
USARTn_RX	importation	Receive data pin
USARTn_CTS	importation	Modem Operation Pins Clear Transmit Pin
USARTn_RTS	exports	Modem Operation Pins Request to send pin

n:1~4

25.4 Functional Description

In this chapter, the functions of UART, multiprocessor communication, smart card, and clock synchronization mode are explained in detail.

25.4.1 UART

25.4.1.1 clocks

The UART can select the clock generated by the internal baud rate generator (internal clock source) or the clock input from the USARTn_CK pin (external clock source) as the clock source for communication.

Internal clock source

USARTn_CR2.CLKC[1:0] bits set to 00b or 01b selects the clock source as the internal clock source, i.e., the clock generated by the internal baud rate generator.

USARTn_CR2.CLKC[1:0]=00b The USARTn_CK pin is not used as a clock pin and can be used as a normal IO.

When USARTn_CR2.CLKC[1:0]=01b, a clock with the same frequency as the communication baud rate is output from the USARTn_CK pin.

The clock source for the internal baud rate generator is selected as PCLK, PCLK/4, PCLK/16, PCLK/64 by the setting of USARTn_PR.PSC[1:0] bits.

External Clock Source

USARTn_CR2.CLKC[1:0] bits set to 10b or 11b selects the clock source to be the external clock input from the USARTn_CK pin, and the frequency of the input clock is 16 times the baud rate (USARTn_CR1.OVER8=0) or 8 times the baud rate (USARTn_CR1.OVER8=1).

maximum baud rate

When the internal clock source is used, the baud rate generated by the internal baud rate generator is calculated as:

$$B = \frac{C}{8 \times (2 - OVER8) \times (DIV_Integer + 1)}$$

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 value

DIV_Integer:USARTn_BRR.DIV_Integer The

maximum baud rate for the set value is PCLK/8

(Mbps).

When the clock source is external, the maximum frequency requirement of the external input UART clock is PCLK(MHz)/4, so the maximum baud rate when the clock source is the external input clock is PCLK/64 (Mbps) (when USARTn_CR1.OVER8=0) or PCLK/32 (Mbps) (when USARTn_CR1.OVER8=1)

Note that the UART maximum communication baud rate should be referred to the maximum communication baud rate specified in the Electrical Characteristics section in addition to the PCLK-based calculation method described above.

25.4.1.2 data format

A frame of data in UART mode consists of a start bit, a data bit, a parity bit, and a stop bit.

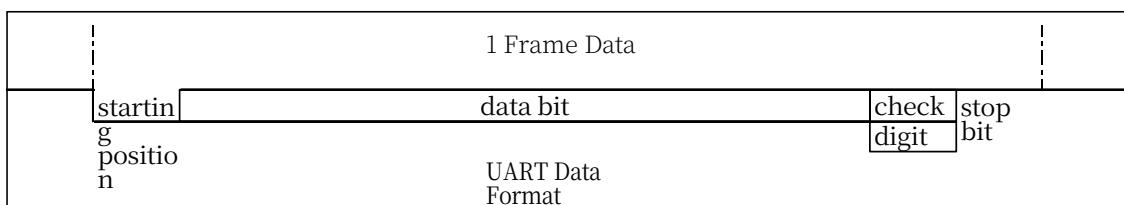


Figure 25-2 UART Data Format

starting position

The start bit is fixed with a one-bit low level composition.

data bit

The data bits can be configured as 8 or 9 bits.

check digit

The parity bit can be configured as 1 bit parity or 1 bit odd parity or no parity bit.

stop bit

The stop bit is fixed high and can be configured as 1 or 2 bits.

25.4.1.3 Modem operation

Modem operation includes CTS function and RTS function, CTS function and RTS function can only be selected one or the other and cannot be used at the same time.

USARTn_CR3.CTSE=0 is valid for RTS function and USARTn_CR3.CTSE=1 is valid for CTS function.

CTS Functions

The CTS function is to control the data sending through the input of USARTn_CTS pin, the data can be sent only when the input of USARTn_CTS pin is low, and the data being sent will not be affected if the input of USARTn_CTS is high during the process of sending data.

RTS Features

The RTS function is to request the other party to send data by outputting a low level on the USARTn_RTS pin. All of the following conditions need to be met for the USARTn_RTS pin to output a low level:

- Receive enable (USARTn_CR1.RE=1) and is not receiving data
- No unread receive data in USARTn_RDR.RDR register
- No receive errors, including framing errors, checksum errors and overflow errors

25.4.1.4 transmitters

The transmitter can send either 8-bit or 9-bit data, depending on the set value of the USARTn_CR1.M bit.

The transmitter enable bit (USARTn_CR1.TE) is set to 1. After the transmit data is written, the transmit data is output serially on the TX pin; the corresponding clock pulse can be selected to be output on the USARTn_CK pin or not.

The sequence of sending data is: start bit -> data bit (MSB/LSB) -> parity bit (yes or no) -> stop bit. The transmit data register TDR and the internal transmit shift register form a double-buffer structure that can send data continuously.

When sending data via Send Data Register Air Break or DMA Write, data can only be written once per request to ensure correct sending.

Send Data Setting Procedure

1. Set the USARTn_CR1 register to the reset value
2. Sets the pins to be used for the UART.
3. Clock source selection via USARTn_CR2.CLKC[1:0] bits
4. Setting USARTn_CR1, USARTn_CR2, USARTn_CR3 Registers
5. Setting USARTn_PR selects the prescaler value, and USARTn_BRR register sets the communication baud rate (not required when the clock source is external)
6. Enable the transmitter (USARTn_CR1.TE=1) and set USARTn_CR1.TXEIE=1 (TE and TXEIE bits are written to 1 at the same time) if you need to use the transmit data register for air breaks
7. Wait for transmit data register to be empty, write communication data to USARTn_TDR.TDR, data transfer to transmit shift register, transmit start

(When the CTS function is active, data is transferred to the transmit shift register when the USARTn_CTS input is low, and transmission starts)

8. Repeat step 7 if you need to send data continuously.
9. Confirm the completion of transmission by checking the USARTn_SR.TC bit. If data is sent continuously and a transmit interrupt is used, the last transmit data can be written through the TI interrupt and the USARTn_CR1.TXEIE is written to 0 and the USARTn_CR1.TCIE is written to 1. A transmit completion interrupt is generated at the end of the last frame of data sent.

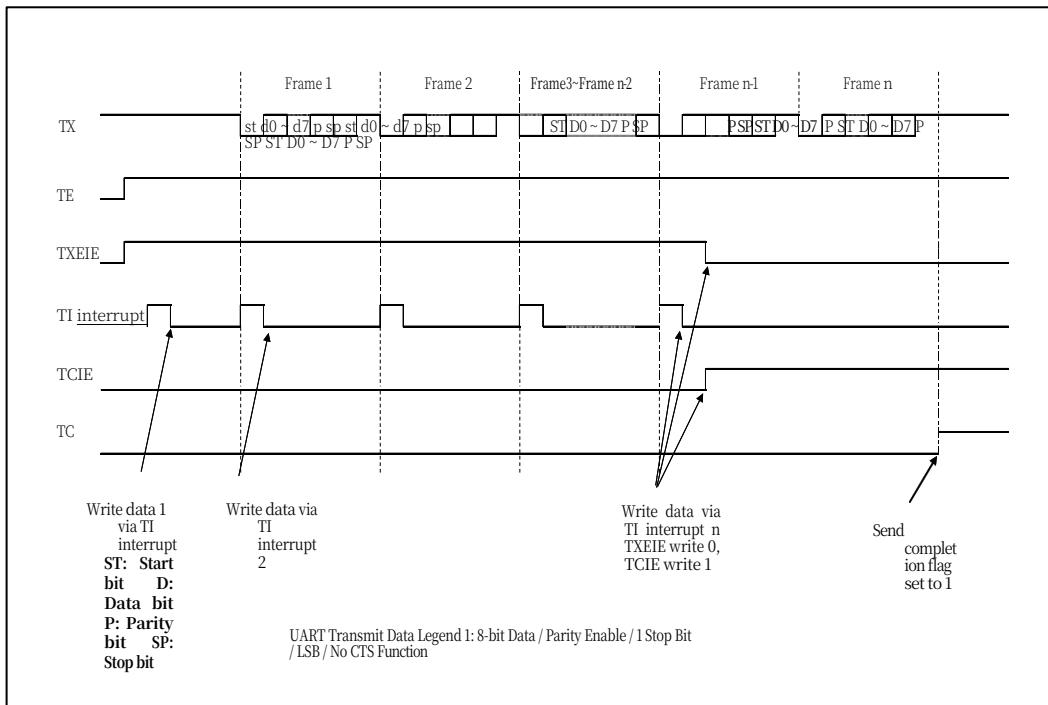


Figure 25-3 UART Transmission Data Legend 1

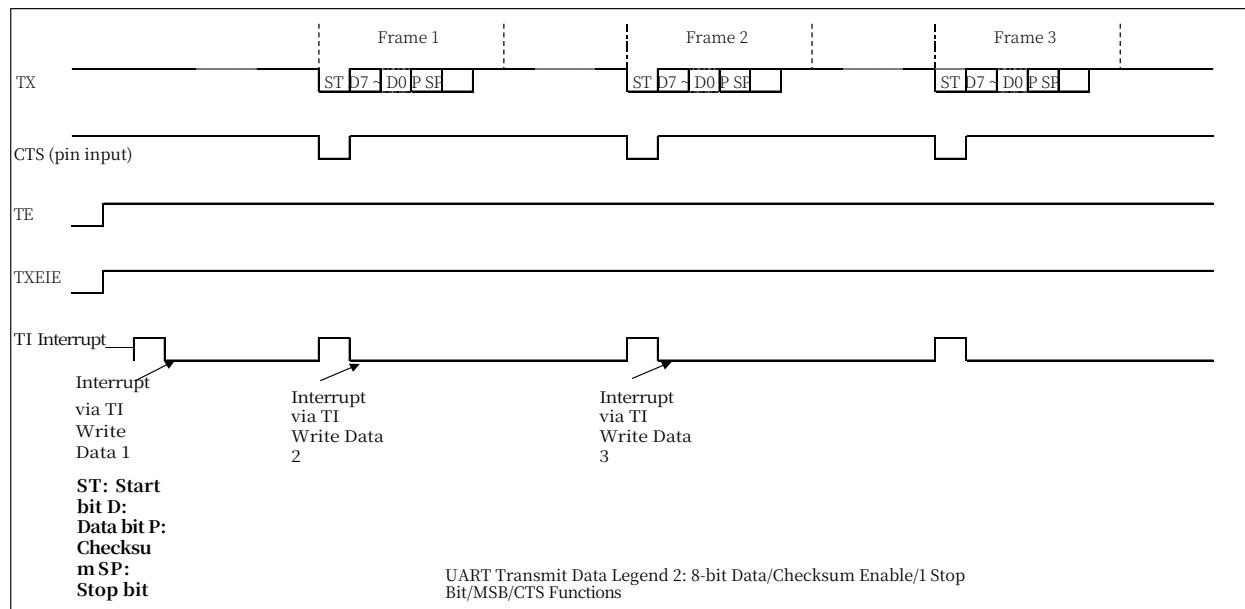


Figure 25-4 UART Transmit Data Legend 2

Transmitt

er

Interrupt

The UART mode transmitter supports two types of interrupts, the Transmit Data Register Interrupt TI and the Transmit Completion Interrupt TCI. TXEIE=1, the TI interrupt occurs when the value of the USARTn_TDR.TDR register is transferred to the transmit shift register. TCIE=1, TCI interrupt occurs if the USARTn_TDR.TDR

25.4.1.5 refraction

The receiver can receive 8-bit or 9-bit data depending on the setting of the USARTn_CR1. After the receiver enable bit (USARTn_CR.RE) is set to 1 and the start bit is detected the data on the RX pin is received into the receive shift register, a full frame of data is received, and the data is transferred from the receive shift register to the receive data register USARTn_RDR.

The order of receiving data is: start bit -> data bit (MSB/LSB) -> parity bit (yes or no) -> stop bit.

The receive data register USARTn_RDR.RDR register and the internal receive shift register form a double-buffer structure to receive data continuously.

When reading receive data through the receive data register full interrupt or DMA, only one data can be read in one request.

Start Bit Detection

The start bit detection can be in low level or falling edge mode depending on the USARTn_CR1.SBS bit, low level detection when USARTn_CR1.SBS=0 and falling edge detection when USARTn_CR1.SBS=1.

Sampling and Receiving Tolerance

Upon detection of a start condition (low or falling edge), the USART starts data reception by clock synchronizing the received data based on the internal base clock.

Data Sampling In Data Center, USARTn_CR1.OVER8=0 samples at the 8th internal base clock, and USARTn_MR.OVER8=1 samples at the 4th internal base clock.

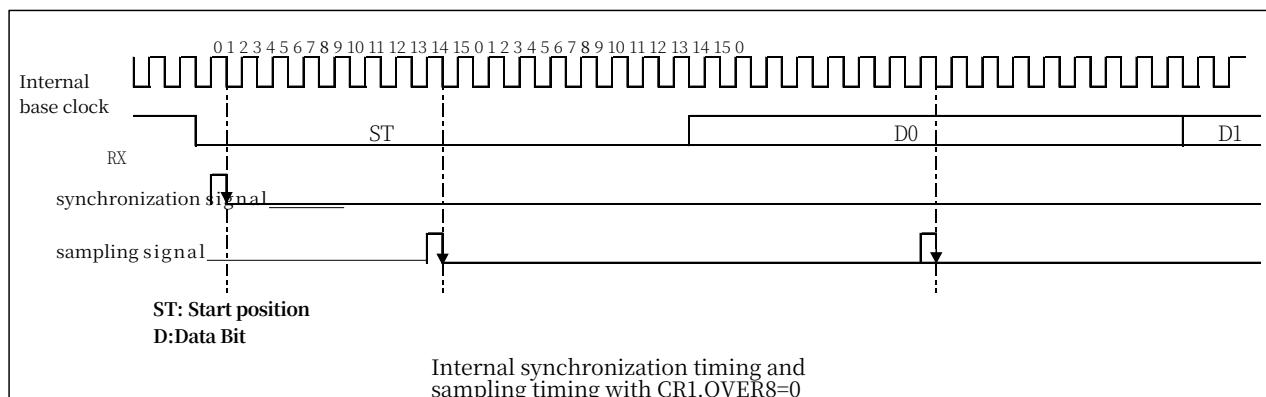


Figure 25-5 UART Internal Synchronization and Sampling Timing

The UART Asynchronous Receiver operates properly only when the total clock system deviation is less than the tolerance of the UART Receiver. Factors that affect the total deviation include:

- Deviations due to transmitter errors (which also include deviations of the transmitter's local oscillator)
- Errors due to baud rate quantization of the receiver

- Receiver local oscillator deviation
- Deviations due to transmission lines

- The maximum deviation allowed by the UART asynchronous receiver for correct data reception depends on the following options:
- Data Length FL (Frame Length). FL is determined by the 8 or 9 data bits defined by the M bit and the parity enable bit defined by the PCE bit in the USART_CR1 register.
- 8x or 16x oversampling defined by the OVER8 bit in the USART_CR1 register
- Whether to use fractional baud rate as defined by the FBME bit in the USART_CR1 register

Table 25-2 UART Receiver Tolerance with DIV_Fraction = 0

FL	OVER8 bits = 0	OVER8 bits = 1
10	4.375%	3.75%
11	3.97%	3.41%
12	3.646%	3.125%

Table 25-3 UART Receiver Tolerance if DIV_Fraction is Not 0

FL	OVER8 bits = 0	OVER8 bits = 1
10	3.88%	3%
11	3.53%	2.73%
12	3.23%	2.5%

In special cases, the data specified in Table 25-2 and Table 25-3 may be slightly different when the received frame contains an idle frame of 10 bits or 11 bits or 12 bits in time.

Receive data setting procedure

1. Set the USARTn_CR1 register to the reset value
2. Sets the pins to be used for the UART.
3. Clock source selection via USARTn_CR2.CLKC[1:0] bits
4. Setting USARTn_CR1, USARTn_CR2, USARTn_CR3 Registers
5. Setting USARTn_PR selects the prescaler value, and USARTn_BRR register sets the communication baud rate (not required when the clock source is an external clock source)
6. Enable the receiver (USARTn_CR1. RE=1) and if a receive interrupt is required, set USARTn_CR1. RIE=1
7. When the start bit is detected, the receiver receives the data into the receive shift register and checks the parity and stop bits
 - 1) In case of a checksum error, the received data is transferred to the USARTn_RDR.RDR register and sets the USARTn_SR.PE flag
 - 2) When the stop bit is not high, a framing error occurs and the received data is

transferred to the USARTn_RDR.RDR register and sets the USARTn_SR.FE flag

- 3) Data is lost and the USARTn_SR.ORE flag is set when an overflow error occurs.
- 4) When no error occurs, the received data is transferred to the USARTn_RDR.RDR register and the USARTn_SR.RXNE flag is set, and the data can be received continuously by repeating step 7 after reading the received data.

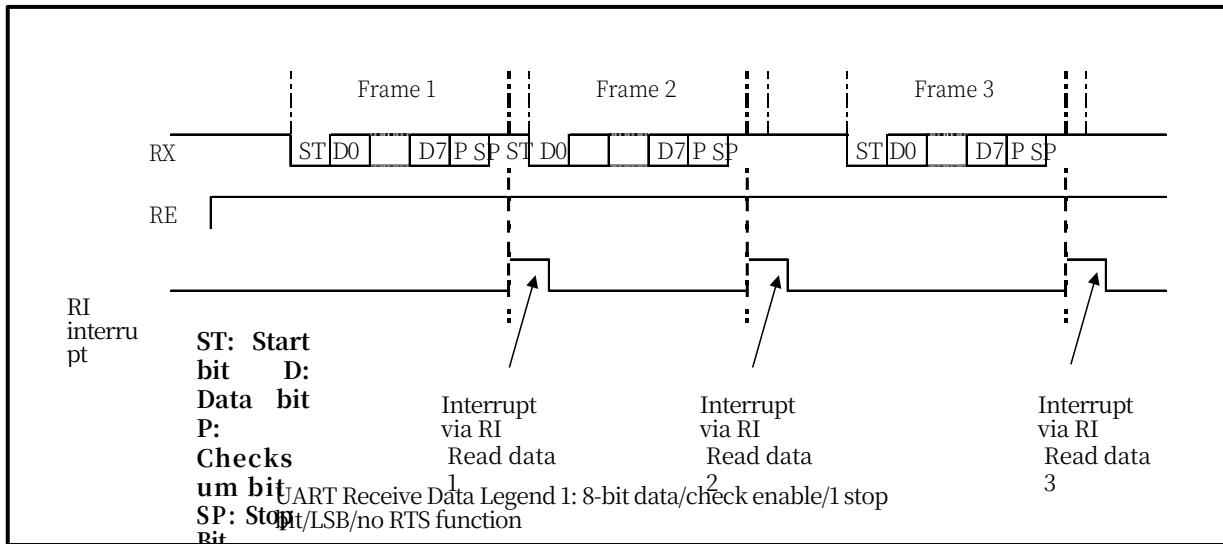


Figure 25-6 UART Receive Data Legend 1

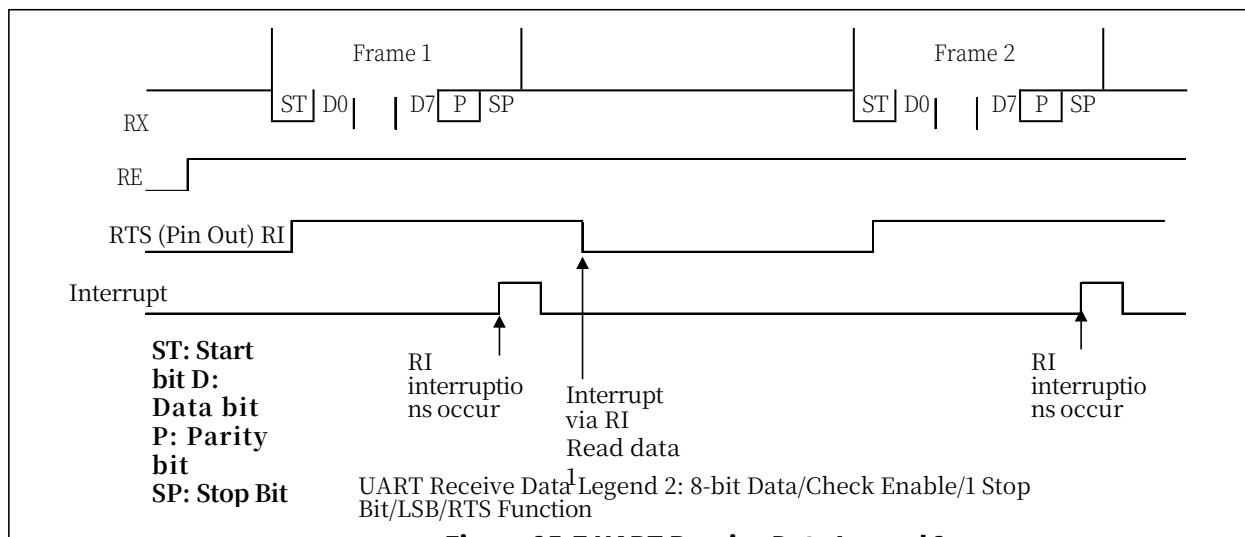


Figure 25-7 UART Receive Data Legend 2

error

handlin

g

There are three types of receive errors when receiving data, namely, Overflow Error (**USARTn_SR.ORE**), Checksum Error (**USARTn_SR.PE**), and Frame Error (**USARTn_SR.FE**). The occurrence of any of these reception errors prevents further data reception. Data reception can be restarted by clearing all error flags by writing to the corresponding clear registers.

value being read, so the previous frame received should be read before the last bit of the current frame is received.

A checksum error occurs when a parity error has occurred.

A frame error occurs when the stop bit is low, and only the first stop bit is checked in the case of two stop bits. When an overflow error occurs, the received data is lost and the RI interrupt does not occur.

Data received when a checksum error occurs is transmitted to USARTn_RDR.RDR, and the RI interrupt does not occur. Data received in the event of a framing error is transmitted to USARTn_RDR.RDR and the RI interrupt does not occur. **Receiver interrupt**

The UART mode receiver supports two types of interrupts, the receive data register full interrupt RI and the receive error interrupt REI.

USARTn_CR.RIE=1, no receive error occurred, RI interrupt occurs when data is transferred from the receive shift register to the receive data register.

USARTn_CR.RIE=1, REI interrupt occurs when an overflow error, checksum error or frame error occurs during reception.

25.4.1.6 UART receive TIMEOUT function

The TIMEOUT counter starts when the UART receive data stop bit is detected, and TIMEOUT occurs when the next frame of receive data is not detected after the set TIMEOUT time (the set unit is the number of receive bits), and the TIMEOUT status bit USARTn_SR.RTOF is reset if CR1.RE=1 at this time, and the TIMEOUT status bit USARTn_SR.RTOF is set after waiting for USARTn_CR1.RE=1 if USARTn_CR1.RE=0 at this time. If CR1.RE=1, then the TIMEOUT status bit USARTn_SR.RTOF is reset, and if USARTn_CR1.RE=0, then the TIMEOUT status bit USARTn_SR.RTOF is reset after waiting for USARTn_CR1.RE=1.

The TIMEOUT counter adopts the counter of Timer0 module, and the specific correspondence is as follows: USART1: Timer0 Unit1 A channel

USART2: Timer0 Unit1 B channel

USART3: Timer0 Unit2 A channel

USART4: Timer0 Unit2 B channel

TIMEOUT Function Timer0 Compare

Counter Value Setting

Timer0 is a 16-bit counter, and the count clock can be selected up to 1024 divisions. The formula for setting the TMR0_CMPAR value is as follows:

$$\text{CMPA}_{<\text{B}>} \text{R} = \frac{\text{RTB}}{2^{\text{CKDIVA}_{<\text{B}>}}} - \alpha$$

CMPAR:TMR0_CMPAR register value, please round up the calculation result. α :Timer0 Asynchronous counting delay caused by simultaneous circuits

When the counting clock is not divided, $\alpha=7$

Counting clock 2 divisions, $\alpha=5$

Counting clock 4, 8, 16 divisions, $\alpha=3$

Counting clock 32 divisions and above, $\alpha=2$

RTB: Receive Timeout Bits, min = Receive Data Frame Length + $\alpha \times 2^{CKDIVA<2>}$

There is a certain error between the actual time of TIMEOUT and the RTB value, the error is $\leq 2^{CKDIVA<2>}.$ CKDIRA $<2>$: CKDIVA $<2>$ bit register value of TMR0.BCONR.

TIMEOUT Function Setting Procedure

1. Set the USARTn_CR1 register to the reset value
2. Sets the pins to be used for the UART.
3. Select the clock source via USARTn_CR2.CLK[1:0] bits (set CR2.CLKC[1]=0 if internal clock source is selected). bit to select the clock source (set CR2.CLKC[1]=0 if internal clock source is selected)
4. Setting USARTn_CR1, USARTn_CR2, USARTn_CR3 Registers
5. Setting USARTn_PR selects the prescaler value, and USARTn_BRR register sets the communication baud rate (not required when the clock source is an external clock source)
6. Clear the USARTn_SR.RTOF flag, set USARTn_CR1.RTOE=1, and if interrupts need to be used, set USARTn_CR1.RTOIE=1
7. Set TMR0.BCONR.CSTA<2>=0
8. Set TMR0_CNTA<2>R to 0. Set TMR0_CMPA<2>R register and TMR0.BCONR.CKDIVA<2> register to determine TIMETOU time.
Set TMR0.BCONR.HCLEA<2>=1, TMR0.BCONR.HSTAA<2>=1,
TMR0.BCONR.ASYNCLKA<2>=1, TMR0.BCONR.SYNSA<2>=1
9. Enable the receiver (USARTn_CR1. RE=1) and if a receive interrupt is required, set USARTn_CR1. RIE=1
10. Set TMR0.BCONR.CSTA<2>=0 after detecting TIMEOUT.

Set TMR0.BCONR.CSTA<2>=0 to clear USARTn_SR.RTOF state by writing USARTn_CR1.CRTOF Bit.

25.4.1.7 RX line wake-up stop mode function

When the UART communication is idle, the system can be put into the stop mode to save current consumption. Without changing the UART PORT setting, UART Unit 1 can wake up the system in the stop mode via the RX line. The steps are as follows:

1. When the UART communication is idle, set the USART_1_WUPI interrupt vector and the INT_WUPEN. RXWUEN bit to enable the UART receive signal line wake-up stop mode function.
2. The system enters stop mode.

-
3. The system returns from stop mode when it detects the falling edge of the RX line and disables this function in the USART_1_WUPI interrupt handler.

It should be noted that when the communicating party needs to wake up the system, it needs to send a frame of wake-up data (0x00 is recommended), which will not be received by the UART and will not set the relevant flags. This data will not be received by the UART and the associated flag will not be set. Also, the communicating party needs to wait for the time required for the system to wake up in stop mode before transmitting the UART communication data.

25.4.1.8 UART Interrupts and Events

Table 25-4 UART Interrupt/Event Table

Function Name	marki ng	Enable bit (interrupt only)	flag position	Can be used as an event source
Receive Error Interrupt	REI	RIE	ORE,FE,PE	may
Receive data register full interrupt	RI	RIE	not have	may
Send Data Register Air Break	TI	TXEIE	not have	may
Send Completion Interrupt	TCI	TCIE	TC	may
TIMEOUT interrupt	RTOI	RTOIE	RTOF	may
RX line wake-up stop mode interrupt	WUPI	INT_WUPEN.RXWUEN	–	unacceptable

25.4.2 multiprocessor communication

25.4.2.1 Function Introduction

The multiprocessor communication mode is a communication mode in which multiple processors share the communication line, and the processors are divided into sending station and receiving station, and each receiving station has its own inherent ID. The sending station sends two types of data: receiving station ID and communication data. The MPB bit is added to the data format to distinguish whether the current frame is receiving station ID or communication data. When the MPB bit is 0, the current frame is communication data. The MPB bit is added to the data format to distinguish whether the current frame is the ID of the receiving station or communication data; when the MPB bit is 0, the current frame is 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 IDs sent by the transmitting station and compare them with their own IDs, and if they are consistent, they will receive the data; if they are not consistent, they will enter the silent mode (neither receiving the data nor setting the receive-related flags) until they receive the IDs again. If not, it enters silent mode (neither receiving data nor setting the receive flag) until it receives the ID again.

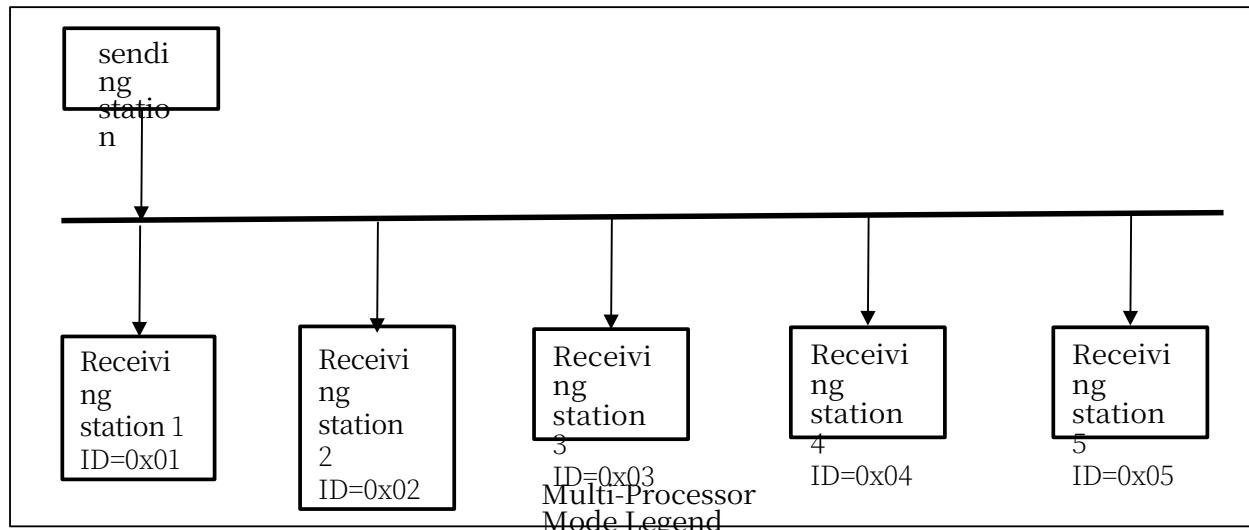


Figure 25-8 Multiprocessor Communication Legend

25.4.2.2 data format

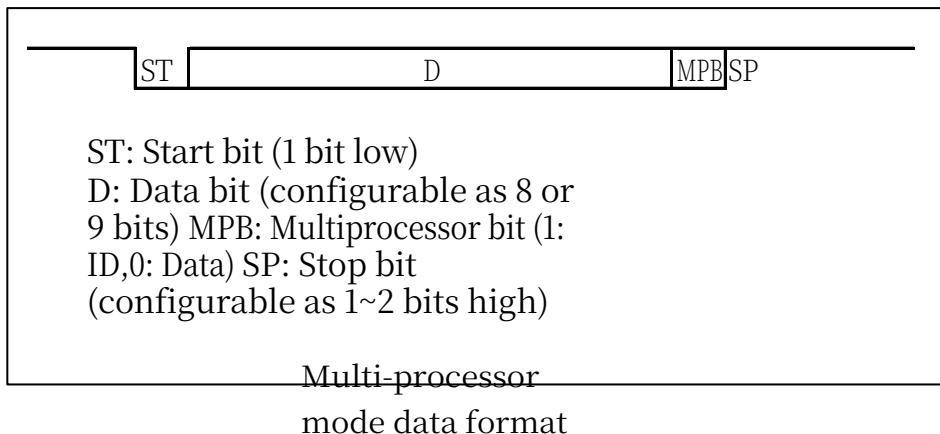


Figure 25-9 Multiprocessor Mode Data Format

25.4.2.3 Action

Description

In multiprocessor mode, the parity bit function is invalidated and the multiprocessor bit function is added. The rest of the functions, such as clock and interrupt, are the same as those in UART mode.

Sending Station Action

1. Set the USARTn_CR1 register to the reset value
2. Setting the pin to be used
3. Clock source selection via USARTn_CR2.CLKC[1:0] bits
4. Setting USARTn_CR1, USARTn_CR2, USARTn_CR3 Registers
5. Setting USARTn_PR selects the prescaler value, and USARTn_BRR register sets the communication baud rate (not required when the clock source is an external clock source)
6. Enable the transmitter (USARTn_CR1. TE=1) and set USARTn_CR1. TXEIE=1 (TE and TXEIE bits are written to 1 at the same time) if you need to use the transmit data register for air breaks
7. Wait for transmit data register to be empty, set USARTn_TDR.MPID bit to 1 (transmit ID), write ID value to USARTn_TDR, transmit ID (When the CTS function is active, data is transferred to the transmit shift register when the USARTn_CTS input is low, and transmission starts)
8. Set the USARTn_TDR.MPID bit to 0 (send data), write data to USARTn_TDR, and send data (when the CTS function is active, data is transferred to the transmit shift register when USARTn_CTS input is low, and sending starts)
9. Repeat step 8 if you need to send data continuously, or repeat 7 and 8 if you need to

10. Confirm the completion of transmission by checking the USARTn_SR.TC bit. If data is sent continuously and a transmit interrupt is used, the last transmit data can be written to the TI interrupt and the USARTn_CR1. TXEIE is written to 0 ~~USARTn_CR1.~~ TCIE is written to 1. A transmit completion interrupt is generated at the end of the last frame of data sent.

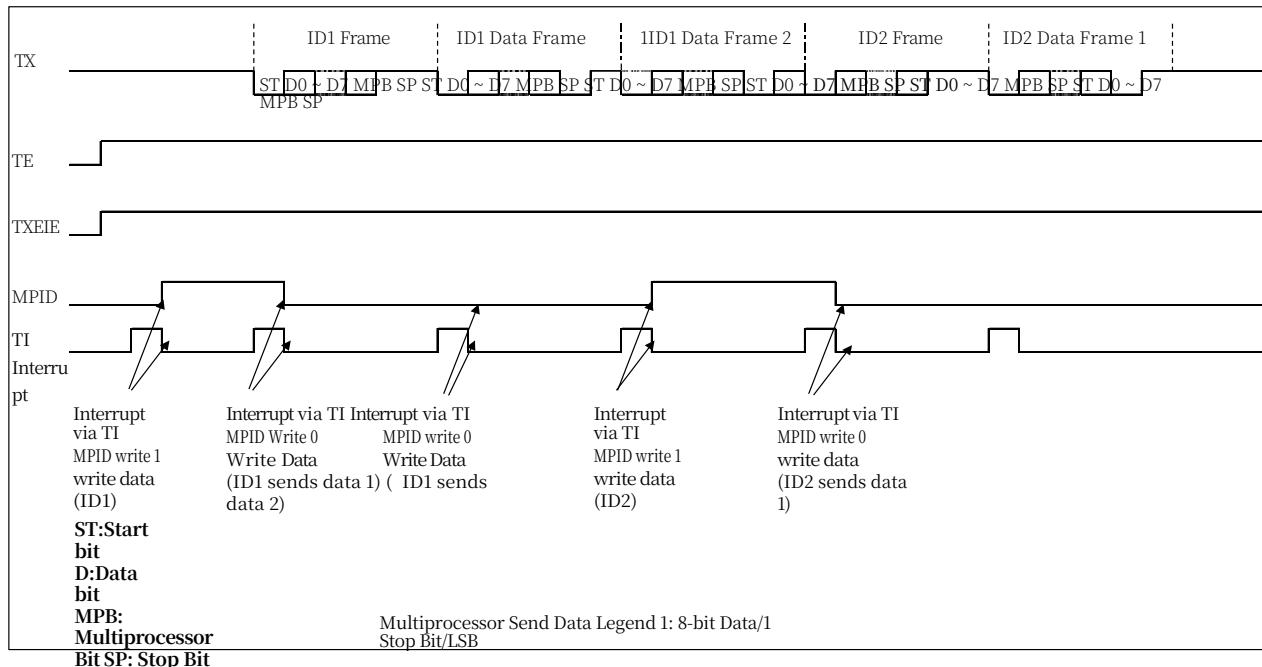


Figure 25-10 Multi-Processor Mode Sending Data Legend

Receiving station action

In multiprocessor mode, the receiving station must ensure that it receives each ID data and compares it with its own ID, if it matches it receives the data, if it does not match it enters silent mode (does not receive the data and does not reset the receive related flags) until it receives the next ID data. This is accomplished through the USARTn_CR1.SLME bit.

Data is received normally when USARTn_CR1.SLME=0.

When **USARTn_CR1.SLME=1**, no data is received, no RI interrupt occurs, and the error flag **FE, ORE** is not set unless data is received with an MPB bit of 1 (ID). When data with an MPB bit of 1 (ID) is received, **USARTn_CR1.SLME** bit is automatically cleared, data is received normally and an interrupt occurs.

Action Steps:

1. Set the USARTn_CR1 register to the reset value
2. Setting the pin to be used
3. Clock source selection via USARTn_CR1.CLKC[1:0] bits
4. Setting USARTn_CR1, USARTn_CR2, USARTn_CR3 Registers
5. Setting USARTn_PR selects the prescaler value, and USARTn_BRR register sets the communication baud rate (not required when the clock source is an external clock source)
6. USARTn_CR1.RE=1, USARTn_CR1.SLME=1 (wait for receive ID), if receive interrupt is used, set USARTn_CR1.RIE=1

8. If USARTn_SR.MPB = 1, the USARTn_CR1.SLME bit is automatically cleared, data is received normally, and the software compares the

Received ID and own ID

- 1) If ID is the same, receive data normally, interrupt occurs and error detection is performed, same as UART receive data.
- 2) If the IDs do not match, the software writes 1 to the USARTn_CR1.SLME bit again and repeats the action of 8.

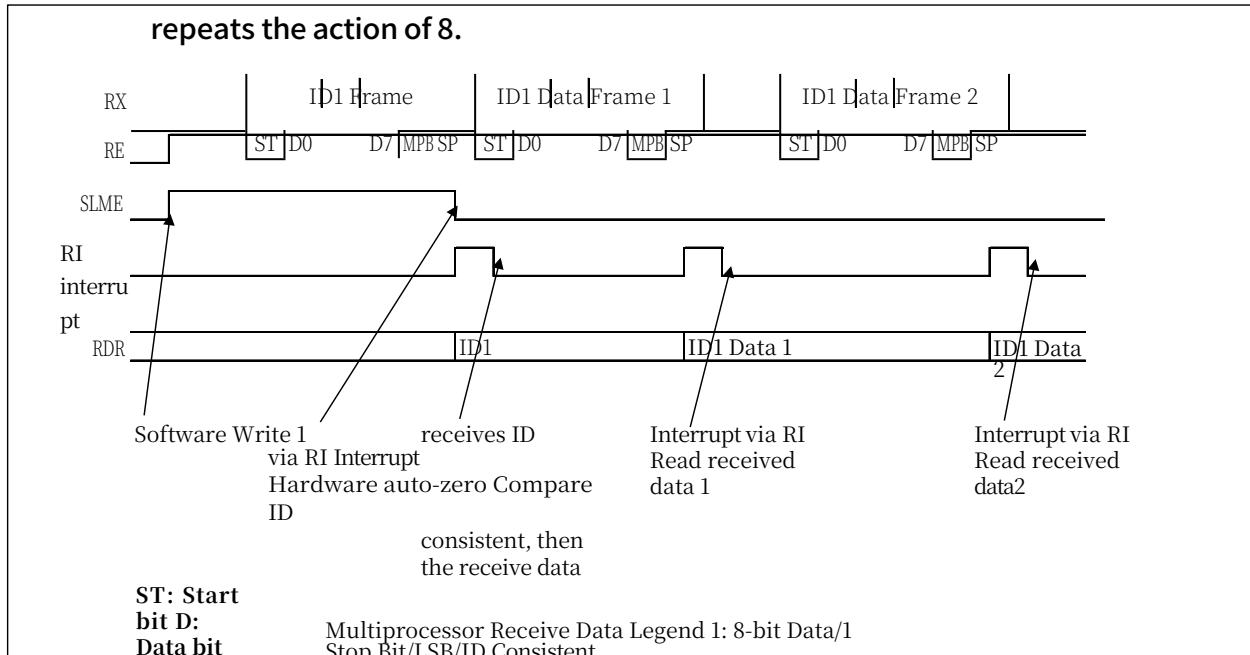


Figure 25-11 Multi-Processor Mode Receive Data Legend 1

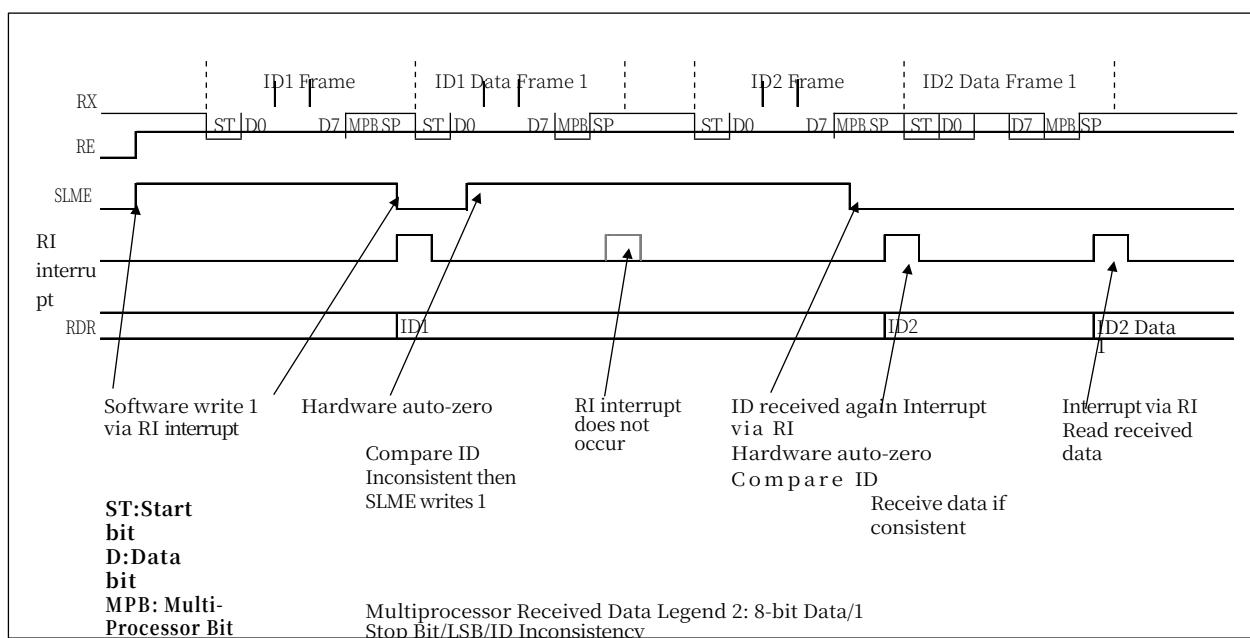


Figure 25-12 Multiprocessor Mode Receive Data Legend 2

25.4.2.4 Interrupts and Events

Multi-processor mode has the same interrupt handling as UART mode except no checksum error.

Table 25-5 Multiprocessor Mode Interrupt/Event Table

Function Name	marking	Enable bit (interrupt only)	symbolize	Can be used as an event source
Receive Error Interrupt	REI	RIE	ORE,FE	unacceptable
Receive data register full interrupt	RI	RIE	RXNE	may
Send Data Register Air Break	TI	TXEIE	TXE	may
Send Completion Interrupt	TCI	TCIE	TC	unacceptable

25.4.3 smart card

25.4.3.1 Connection Schematic

Supports the smart card communication protocol specified in ISO/IEC 7816-3. The following figure shows the connection schematic of the smart card mode.

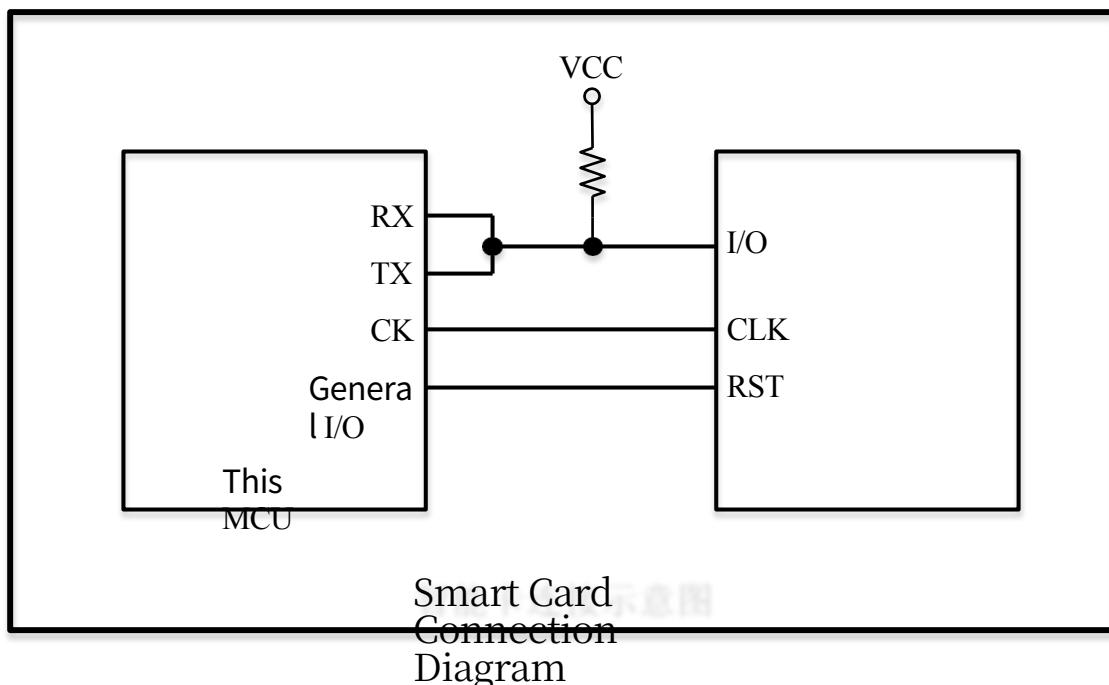


Figure 25-13 Smart Card Connection Schematic

25.4.3.2 clocks

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

USARTn_CR3.BCN[2:0] setting.

The clock output for smart card mode is controlled by setting the register USARTn_CR2.CLKC[1:0] bits.

maximum baud rate

When the internal clock source is used, the baud rate generated by the internal baud rate generator is calculated as:

$$B = \frac{C}{2 \times BCN \times (DIV_Integer + 1)}$$

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

value BCN:USARTn_CR3.BCN Register setting

value

When C is PCLK, DIV_Integer=0, and BCN=0, the baud rate is the highest baud rate of PCLK/64 (MBps).

Sampling and Receiving Tolerance

Upon detecting the falling edge of RX, the USART will clock-synchronize the receive data based on the internal base clock to start data reception. The receive data will be sampled at the center of the data.

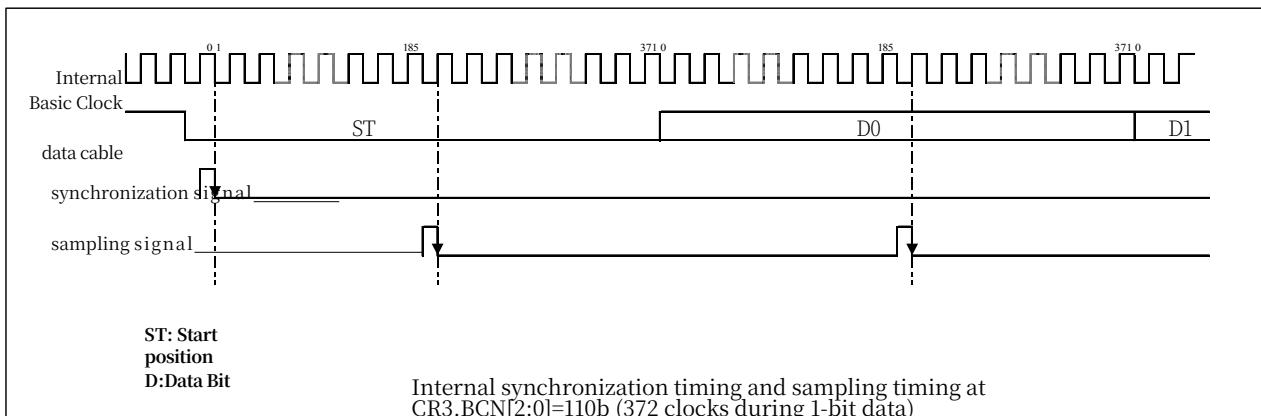


Figure 25-14 Smart Card Mode Synchronization Timing and Sampling

The formula for receiving **Timing Diagrams**

tolerance is as follows:

$$RM[\%] = |0.5 \times \frac{1}{BCN} - 9.5CFD| \times 100$$

RM: Receiving

Tolerance

BCN: Number of clocks required for one bit of data transmission

(USARTn.CR3.BCN[2:0] setting) CFD: Clock frequency deviation

25.4.3.3 data format

A frame of data in smart card mode consists of a start bit, a data bit and a check bit.

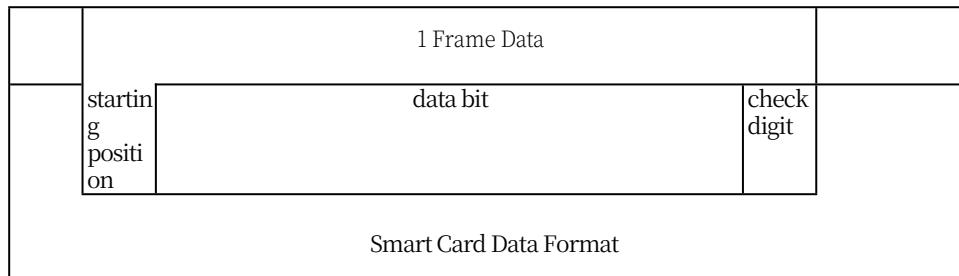


Figure 25-15 Smart Card Mode Synchronization Timing and Sampling Timing Chart

starti

ng

positi

on

The start bit is fixed with a one bit low level composition.

data bit

The data bits are fixed to 8-bit data.

check digit

The parity bit needs to be configured as 1-bit parity.

25.4.3.4 Initialization Setting Procedure of Smart Card

1. Set the USARTn_CR1 register to the reset value
2. Setting the pin to be used
3. Status register acknowledgement, USARTn_SR register set to reset value
4. Setting USARTn_CR1, USARTn_CR2, USARTn_CR3 Registers
5. Setting USARTn_PR Selects the prescaler value and USARTn_BRR register sets the communication baud rate
6. USARTn_CR2.CLKC[1:0] bits set clock control
7. USARTn_CR1 register (TE, RE, RIE, TXEIE bits) setting, except self-test, TE and RE should not be set to 1 at the same time.

When switching from the transmit mode to the receive mode, or from the receive mode to the transmit mode, it is necessary to reset steps 1 to 7 above.

25.4.3.5 Smart Card Mode Action Description

In smart card mode, the flag bit for the TI interrupt (transmit data air interrupt) is the USARTn_SR.TC bit, and the TI interrupt is generated when USARTn_SR.TC=1 and USARTn_CR1.TXEIE=1.

When sending data, between two frames of data (from the end of the parity bit to the start of the next frame), there is a protection time of 2etu (Elementary Time Unit) or more.

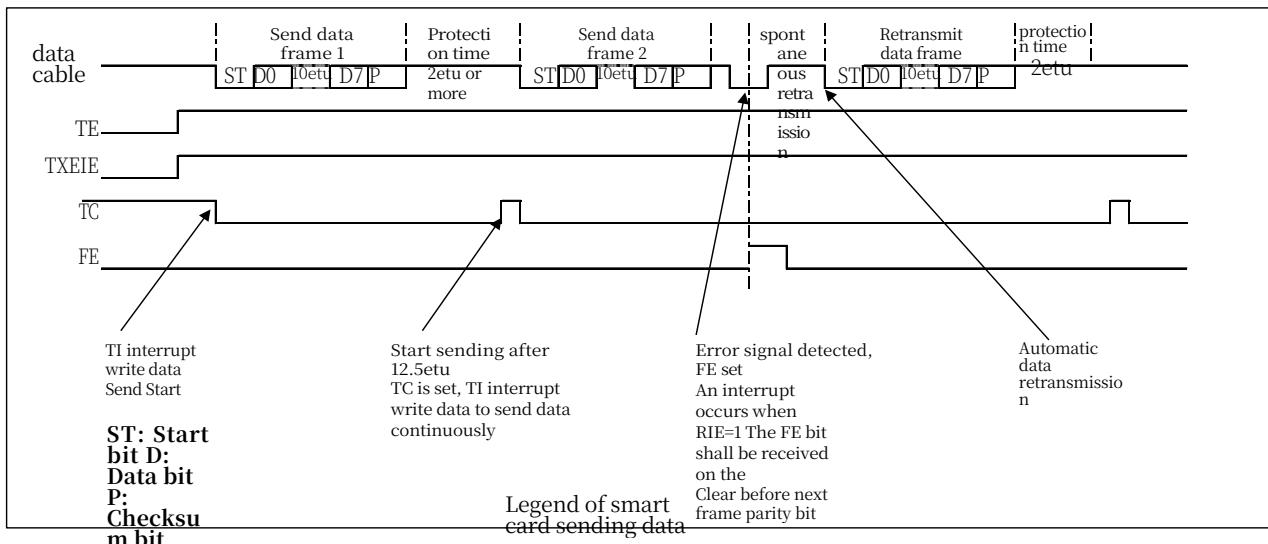
Time Unit) or more between two frames of data (from the end of the parity bit to the start of the next frame).

If an error signal from the receiver is detected when sending data, the data is automatically retransmitted after 2etu.

When a checksum error occurs in the received data, a low level of 1etu is sent as an error signal, and the timing of the error signal is 10.5etu from the start of reception.

Sending instructions

1. After a frame is sent, if an error signal from 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 the next frame parity bit is accepted.
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 continuous



data transmission can be realized.

Figure 25-16 Smart Card Mode Sending Data Legend

Receiving instructions

1. If a checksum error is detected when receiving data, USARTn_SR.PE is set and a REI interrupt occurs when the interrupt is enabled. The USARTn_SR.PE bit needs to be cleared before the next frame checksum bit is received.
2. When a checksum error occurs, a low level of 1etu, the error signal, is sent, requiring the sender to resend the data.
3. To receive data normally, you can read the received data through the RI interrupt and receive it continuously.
4. Overflow error detected when receiving data.

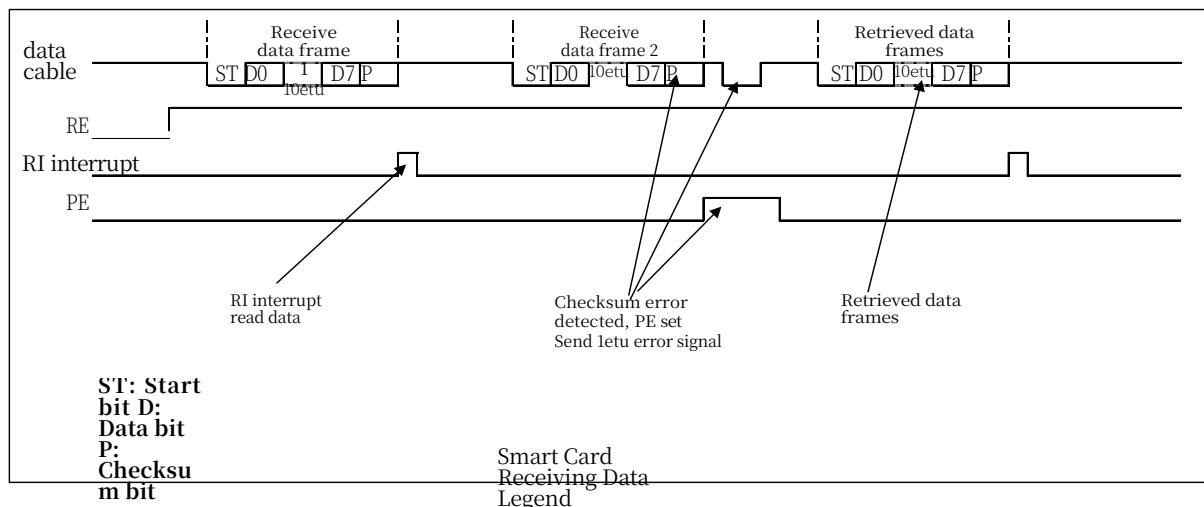


Figure 25-17 Smart Card Mode Receive Data Legend

25.4.3.6 Interrupts

and
Events

Table 25-6 Smart Card Mode Interrupt/Event Table

Function Name	markin g	Enable bit (interrupt only)	flag position	Can be used as an event source
false interruption	REI	RIE	ORE,PE,FE	may
Receive data full interrupt	RI	RIE	RXNE	may
Send data over-the-air disconnect	TI	TXEIE	TC	may

25.4.4 Clock Synchronization Mode

25.4.4.1 clocks

The clock synchronization mode allows you to select the clock generated by the internal baud rate generator (internal clock source) or the clock input from the USARTn_CK pin (external clock source) as the clock source for communication.

Internal clock source

External Clock Source

The external clock source, i.e. the clock input from the USARTn_CK pin, is used as the communication clock.

maximum baud rate

When the internal clock source is used, the baud rate generated by the internal baud rate generator is calculated as:

$$B = \frac{C}{4 \times (\text{DIV_Integer} + 1)}$$

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 value

The maximum baud rate is PCLK/8(MBps) when C is PCLK and DIV_Integer=1 for internal clock source. Note that DIV_Integer is prohibited to be set to 0 in synchronous mode.

For an external clock source, the maximum frequency of the external input clock is required to be PCLK(MHz)/6, so the maximum baud rate is PCLK/6(MBps).

Note that the Synchronous Mode maximum communication baud rate requires reference to the maximum communication baud rate specified in the Electrical Characteristics section in addition to the PCLK-based calculation method described above.

25.4.4.2 data format

Clock Synchronization Mode A frame of data consists of 8 fixed bits, and 8 synchronized clock pulses are required to send and receive a frame of data. When data is sent, it is sent on the falling edge of the synchronization clock, and when data is received, it is sampled on the rising edge of the synchronization clock.

The synchronization clock is fixed high when no data is being transmitted, and the communication line holds the value of the last bit after the last bit has been sent.

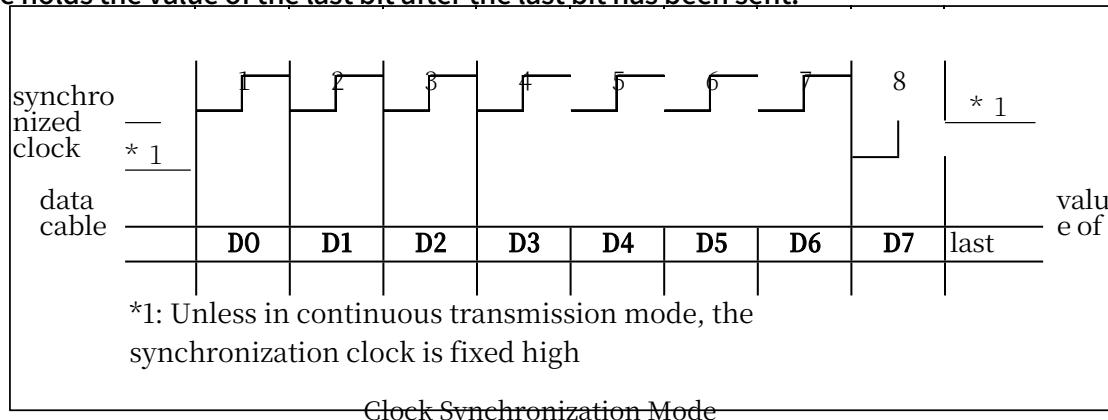


Figure 25-18 Clock Synchronization Mode Data Format

25.4.4.3 Modem operation

Modem operation includes CTS function and RTS function, CTS function and RTS function can only be selected one or the other and cannot be used at the same time.

USARTn_CR3.CTSE=0 is valid for RTS function and USARTn_CR3.CTSE=1 is valid for CTS function.

CTS Functions

The CTS function is to control the data sending through the input of USARTn_CTS pin, the data can be sent only when the input of USARTn_CTS pin is low, and the data being sent will not be affected if the input of USARTn_CTS is high during the process of sending data.

RTS Features

The RTS function is to request the other party to send data by outputting a low level on the USARTn_RTS pin. All of the following conditions need to be met for the USARTn_RTS pin to output a low level:

- Receive enable (USARTn_CR1.RE=1) and is not receiving data
- No unread data in USARTn_RDR.RDR register (when USARTn_CR1.RE=1)
- USARTn_TDR.TDR update completed (when USARTn_CR1.TE=1)
- No reception errors

If all of the above conditions are not met at the same time, USARTn_RTS outputs a high level.

25.4.4.4 transmitters

When the transmitter enable bit (USARTn_CR1.TE) is set to 1, the data in the transmit shift register is output serially on the USARTn_TX pin and the corresponding clock pulse is output on the USARTn_CK pin.

The transmit data register USARTn_TDR.TDR and the internal transmit shift register form a double-buffer structure that allows continuous data transmission.

When sending data through the send data register air break or DMA write, data can only be written once per request to ensure correct sending.

Send Data Setting Procedure

1. Set USARTn_CR1, USARTn_SR1 registers to reset value
2. Setting the pin to be used
3. Clock source selection via USARTn_CR2.CLKC[1:0] bits
4. Setting USARTn_CR1, USARTn_CR2, USARTn_CR3 Registers
5. Setting USARTn_PR selects the prescaler value, and USARTn_BRR register sets the

communication baud rate (not required when the clock source is an external clock source)

6. Enable the transmitter (USARTn_CR1. TE=1) and set USARTn_CR1.TXEIE=1 (TE and TXEIE bits are written to 1 at the same time) if you need to use the transmit data register for air breaks
7. Wait for transmit data register to be empty, write communication data to USARTn_TDR.TDR, data transfer to transmit shift register, the

Send Start

(When the CTS function is active, data is transferred to the transmit shift register when the USARTn_CTS input is low, and transmission starts)

8. Repeat step 7 if you need to send data continuously.
9. Confirm the completion of transmission by checking the USARTn_SR.TC bit. If data is sent continuously and a transmit interrupt is used, write the last transmitted data through the TI interrupt and write 0 to USARTn_CR1.TXEIE and 1 to USARTn_CR1.TCIE, and generate a transmit completion interrupt after the last data has been sent.

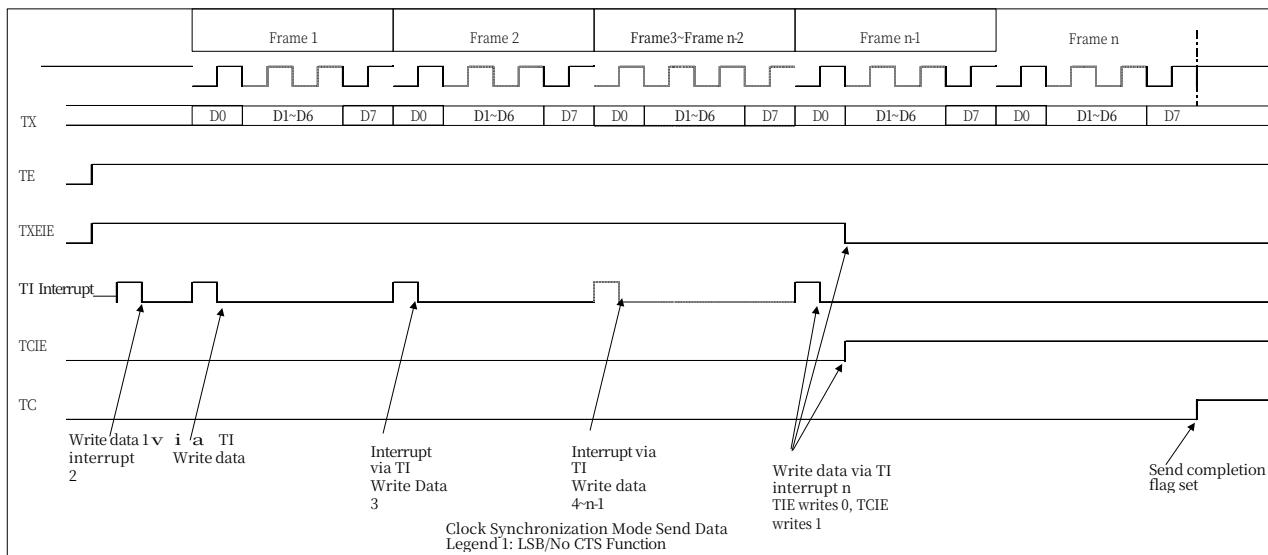


Figure 25-19 Clock Synchronization Mode Data Transmission Legend 1

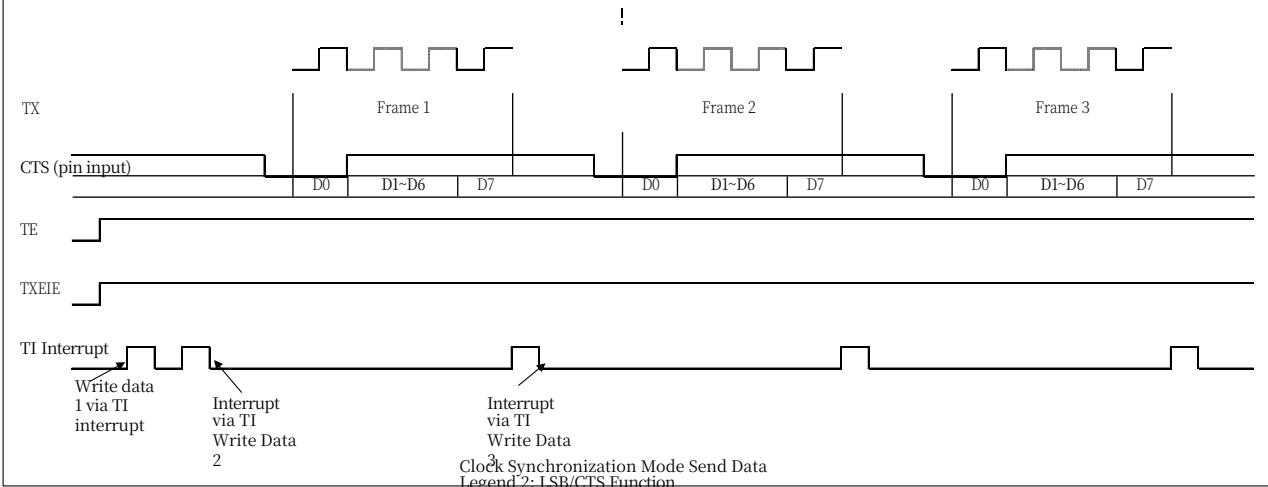


Figure 25-20 Clock Synchronization Mode Data Transmission Legend 2

Transmitter Interrupt

The Clock Synchronized Mode transmitter supports two types of interrupts, the

Transmit Data Register Empty Interrupt TI and the Transmit Complete Interrupt TCI.

transferred to the transmit shift register. TCIE=1, TCI interrupt occurs if the

USARTn_TDR.TDR register is not updated when the last bit of data is sent.

25.4.4.5 refraction

Receive data setting procedure

1. Set USARTn_CR1, USARTn_SR registers to the reset value
2. Setting the pin to be used
3. Clock source selection via USARTn_CR2.CLKC[1:0] bits
4. Setting USARTn_CR1, USARTn_CR2, USARTn_CR3 Registers
5. Setting USARTn_PR selects the prescaler value, and USARTn_BRR register sets the communication baud rate (not required when the clock source is an external clock source)
6. Enable the receiver (USARTn_CR1. RE=1) and if you need to use the receive interrupt, set USARTn_CR1.
RIE=1

(When RTS function is used, USARTn_RTS output is low after RE=1)

7. Synchronized to the input synchronization clock or internally generated synchronization clock to start receiving data to the receive shift register.
 - 1) Data is lost and the USARTn_SR.ORE flag is set when an overflow error occurs.
 - 2) When no error occurs, the received data is transmitted to the USARTn_RDR.RDR register, the USARTn_SR.RXNE flag is set, and the currently received data is read before the last bit of the data of the next frame is received, and then step 7 is repeated to realize the function of continuously receiving data.

(When the RTS function is used, the USARTn_RTS output goes low after the data is read)

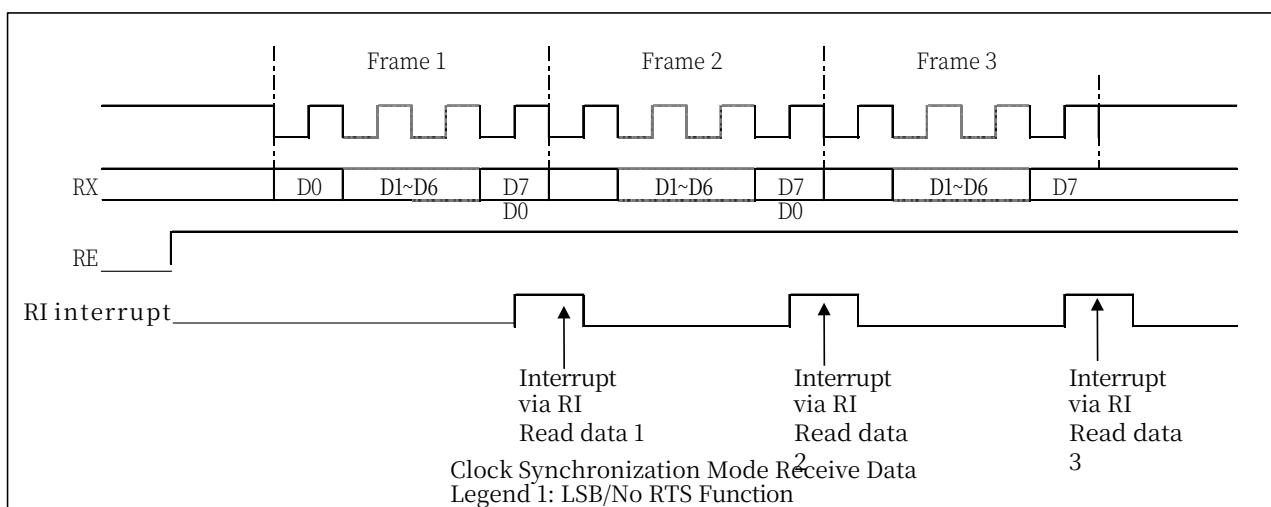
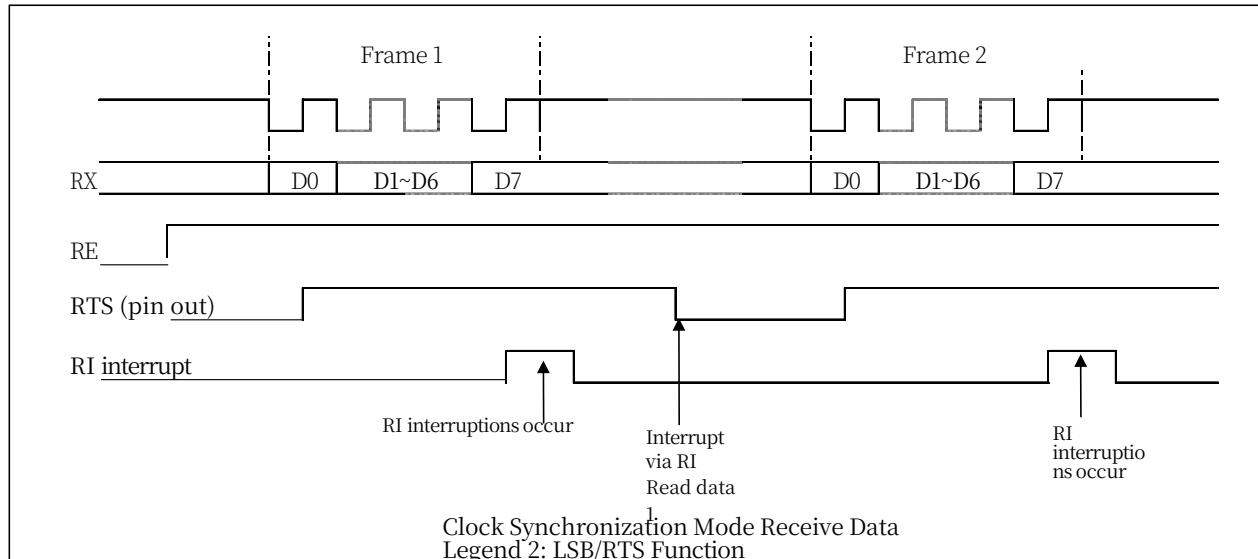


Figure 25-21 Clock Synchronization Mode Receive Data Legend 1

**Figure 25-22 Clock Synchronization Mode Receive Data Legend 2****error****handlin****g**

The receive error when receiving data in clock synchronization mode is an overflow error (USARTn_SR.ORE). Data reception and transmission can no longer be performed when a reception error occurs. Data transmission can be restarted by clearing the error flag by writing the corresponding clear register.

An overflow error occurs when new data is received without the USARTn_RDR.RDR value being read, so the data received in the previous frame should be read before the last bit of the current frame is received. When an overflow error occurs the received data is lost and the RI interrupt does not occur.

receiver interrupt

The Clock Synchronization Mode receiver supports two interrupts, the Receive Data Register Full interrupt RI and the Receive Error interrupt REI. RIE=1, the RI interrupt occurs when data is transferred from the Receive Shift Register to the Receive Data Register.

RIE=1, REI interrupt occurs when an error (overflow error) occurs in receiving data.

25.4.4.6 Simultaneous sending and receiving of data

USART clock synchronization mode supports full-duplex operation to send and receive data at the same time. When sending and receiving data at the same time, a command is needed to write RE,TE,RIE,TXEIE to 1. Other setting procedures are the same as those of the transmitter and receiver.

Table 25-7 Clock Synchronization Mode Interrupt/Event Table

Interrupt name	markin g	Enable bit (interrupt only)	flag position	Can be used as an event source
Receive Error Interrupt	REI	RIE	ORE	unacceptable
Receive data register full interrupt	RI	RIE	RXNE	may
Send Data Register Air Break	TI	TXEIE	TXE	may
Send Completion Interrupt	TCI	TCIE	TC	unacceptable

25.4.5 Digital Filtering Function

When USARTn_CR1.NFE=1, the built-in digital filter function is effective. The digital filter is effective only in UART mode and removes noise on the receive data line RX.

The built-in digital filter filters out noise smaller than one bit of data at 3/16 (USARTn_CR1.OVER8=0) width or 3/8 width (USARTn_CR1.OVER8=1).

If the digital filter clock stops and then starts again, the digital filter continues to operate from the state it held when the clock stopped. USARTn_CR.TE=0 and USARTn_CR.RE=0 reset the Flip-Flop state to 1 inside the digital filter.

25.5 Register Description

USART1_BASE_ADDR:0x4001_D000

USART2_BASE_ADDR:0x4001_D400

USART3_BASE_ADDR:0x4002_1000

USART4_BASE_ADDR:0x4002_1400

Table 25-8 USART Register List

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 Register 2 (USART_CR2)	0x10	0x0000_0000
Control Register 3 (USART_CR3)	0x14	0x0000_0000
Prescaler register (USART_PR)	0x18	0x0000_0000

25.5.1 Status Register (USART_SR)

USART Status Register

offset address: 0x00

Reset value: 0x0000_00C0

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
-	-	-	-	-	-	-	RTOF	TXE	TC	RXNE	-	ORE	-	FE	PE

classifier for honorific people	marking	celebrity	functionality	fill out or in (informati on on a form)
---------------------------------------	---------	-----------	---------------	---

b31~b17	Reserved	-	Reads "0", writes "0".	R
b16	MPB	multiprocessor bit	multiprocessor bit flag (computing) 0: Currently received data is communication data 1: Currently receiving data as ID Note: The MPB bit is only valid in multiprocessor mode	R
b15~b9	Reserved	-Reserved	Read "0", write "0".	R
b8	RTOF	UART receives the TIMEOUT tag aspiration	UART receive TIMEOUT flag bit 0: no UART receive TIMEOUT 1: UART receive TIMEOUT occurs RTOF Setting Conditions • No new data is detected after a set time has elapsed from the STOP bit at which the last frame of data was detected. receive data RTOF Zeroing Conditions • Clear Register CR1.CRTOF Bit Write Note: RTOF is set to 1 by hardware, and it is only set to 1 when CR1.RE=1 and CR1.RTOE=1. When CR1.RE=0, the TIMEOUT function is valid, but RTOF is not set to 1. RTOF is only valid when USART_CR1.RTOE=1, and it is ignored in other cases, and when you use UART to receive Timeout function, you need to clear this bit by software before setting RTOE=1. RTOF is valid only when USART_CR1.	R
b7	TXE	Send data register empty	Send Data Register Empty Flag The TXE bit is valid when the USART is in USART and clock synchronization mode.	
	b6	TC	Transmission completion flag	

1: Data transfer to shift register, send data register

in progress

empty

1: Send data completion

UART Mode, Clock Synchronization Mode

Note: The TXE bit is set and cleared by hardware.

TC Setting Condition

R

Hardware clears TXE to 0 when data is not transferred to the shift register, and sets TXE to 1 when data is transferred to the shift register.

• TE=0 when transmission is disabled

• The value of the transmit data register is not updated when the last bit of a frame is sent TC clear condition

T

r

a

n

s

m

i

s

s

i

o

n

- When TE=1, send data is written to the send data register

smart card mode

TC Setting Conditions

- TE=0 when transmission is disabled
- After the last 1 byte of data has been sent out for a specific period of time, FE=0 and the data is sent out.

The value of the instrument has not been updated.

The specific timing of the TC setup is: 2.5 bits after the parity bit is delivered

TC clearing conditions

- When TE=1, send data is written to the send data register

Note: TC is held at 1 when the TE bit changes from 0 to 1.

Receive data register not empty flag

0: No data received

b5	RXNE	Receive data register not empty	1: Prepare to read the received data	R
----	------	---------------------------------	--------------------------------------	---

Note: The RXNE bit is set to 1 and cleared to 0 by hardware, and when ready to read received data hardware will RXNE set to 1, hardware clear RXNE to 0 after reading received data

b4	Reserved	-	Reads "0" and writes "0".	R
----	----------	---	---------------------------	---

Receive Overflow Error Flag Bit

0: No receive overflow error

1: Receive overflow error occurred

ORE Setting Conditions

- A new frame is received without the receive data register being read

b3	ORE	Receive overflow error	ORE zeroing conditions	R
----	-----	------------------------	------------------------	---

• Clear register CR1.CORE bit written to 1

Note: RE=0 does not reset the ORE bit

Data received before ORE=1 is held, data received with ORE=1 is cast away

You cannot continue to receive data after ORE=1, and you cannot send data in clock-synchronized mode.

b2	Reserved	-	Reads "0" and writes "0".	R
----	----------	---	---------------------------	---

Receive frame

error flag bit 0:

no receive frame

error

1: Receive frame

error occurred

UART mode

FE Setting Conditions

- 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

b1	FE	Receive Frame	first stop bit is checked in the case of two stop bits	R
----	----	---------------	--	---

Error

FE Zeroing Conditions

- Clear register CR1.CFE bit writes 1

Note: RE=0 does not reset the FE bit when in UART mode

When FE=1, received data will be retained but RI interrupt will not occur, and data cannot be received after FE=1.

Smart Card

Mode FE

Setting

Conditions

- Sample to low level error signal flag FE clear condition
- Clear register CR1.CFE bit writes 1

Note: RE=0 does not reset the FE bit when in smart card mode

Received data

checksum error flag 0:

No received data

checksum error

b0 PE Receive Data
Check

1: Received data

checksum error occurred

PE set condition

- When a parity error occurs in the received data

R

PE zeroing conditions

- Clear register CR1.CPE bit

writes 1 Note: RE=0 does not
reset the PE bit

When PE=1, the received data will be retained but RI
interrupt will not occur, and data cannot be received after PE=1.

25.5.2 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]														

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b15~b10	Reserved	-	Reads "0" and writes "0".	R/W
b9	MPID	Multiprocessor mode ID bits data	Selection bit for sending communication data or sending ID in multiprocessor mode R/W 1: Send ID Note: MPID bit is only valid in multiprocessor mode, other modes must be set to reset value.	0: Send
b8~b0	TDR[8:0]	Transmit Data Register	Transmit Data Register Note: The highest bit, TDR[8], is only valid in UART mode when the data length is set to 9 bits.	R/W

25.5.3 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]														

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b15~b9	Reserved	-	Reads "0", writes "0".	R/W
b8~b0	RDR[8:0]	Receive Data Register	Receive Data Register Note: The highest bit, RDR[8], is only valid in UART mode when the data length is set to 9 bits.	R

25.5.4 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]				

classifier for marking	celebrity	functionality	fill out or in (information on a form)
honorific people			
b31~b16 Reserved	-	Reads "0", writes "0".	R
b15~b8 DIV_Integer[7:0]	Integer frequency divider register	Note: DIV_Integer[7:0] can only be used when TE=0 & RE=0 (send/receive disabled). Set when (stop)	R/W
b7 Reserved	-	Reads "1" and writes "1".	R
b6~b0 DIV_Fraction[6:0]	Fractional frequency divider register	Note: DIV_Fraction[6:0] can only be used when TE=0 & RE=0 (send/receive disabled). The set value is set when FBME=1 and is valid only when FBME=1.	R/W

Table 25-9 Baud Rate Calculation Formulas (Fractional Baud Rate Invalid FBME=0)

paradigm	Baud rate formula	Calculation formula for error E (%)
UART mode multiprocessor mode	$B = \frac{C}{8 \times (2 - OVER8) \times (DIV_Integer + 1)}$	$E(\%) = \frac{C}{8 \times (2 - OVER8) \times (DIV_Integer + 1) \times B} - 1 \times 100$
Clock Synchronization Mode	$B = \frac{C}{4 \times (DIV_Integer + 1)}$	-
smart card mode	$B = \frac{C}{2 \times BCN \times (DIV_Integer + 1)}$	$E(\%) = \frac{C}{2 \times BCN \times (DIV_Integer + 1) \times B} - 1 \times 100$

B: Baud rate Unit: Mbps C: Clock set by

PR.PSC[1:0] bits Unit: MHz BCN:

CR3.BCN register setting value

Table 25-10 Baud Rate Calculation Formulas (Fractional Baud Rate Valid FBME=1)

paradigm	Baud rate formula	Calculation formula for error E (%)
UART mode Multi-processor mode	$B = \frac{C \times (128 + DIV_Fraction)}{8 \times (2 - OVER8) \times (DIV_Integer + 1) \times 256}$	$E(\%) = \frac{C \times (128 + DIV_Fraction)}{8 \times (2 - OVER8) \times (DIV_Integer + 1) \times 256 \times B} - 1 \times 100$

Clock Synchronization Mode	$B = \frac{C \times (128 + \text{DIV_Fraction})}{4 \times (\text{DIV_Integer} + 1) \times 256}$	
smart card mode	$B = \frac{C \times (128 + \text{DIV_Fraction})}{2 \times \text{BCN} \times (\text{DIV_Integer} + 1) \times 256}$	$E(\%) = \frac{C \times (128 + \text{DIV_Fraction})}{2 \times \text{BCN} \times (\text{DIV_Integer} + 1) \times 256 \times B} - 1 \times 100$

B: Baud rate Unit: Mbps C: Clock set by

PR.PSC[1:0] bits Unit: MHz BCN:

CR3.BCN register setting value

25.5.5 Control Register 1 (USART_CR1)

USART Control Register 1

Offset Address: 0x0C

Reset value: 0x8000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	
SEBS	NFE	FBME	ML	-	-	-	MS	-	-	-	CRTO F	COR E	-	CFE	CPE
b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1
OVER 8	-	-	M	-	PCE	PS	-	TXE IE	TCI E	RIE	SLME	TE	RE	RTO IE	RTO E

classifier for honorable people	marking	celebrity	functionality	fill out or in (information on a form)
--	---------	-----------	---------------	--

b31	SEBS	According to the beginning of the bit testing party stylesetting	Number of UART mode receivers	When data is received in UART mode, the start bit detection method is set to the position.	R/W
			0: Start bit detection mode for RX pin low level 1: The start bit detection method is the falling edge of RX pin	Note: The SBS bit must remain reset in non-UART mode.	
b30	NFE	Digital filter enable bit	The SBS bit can only be set when TE=0 & RE=0 (transmit/receive disable) Digital Filter Enable Bit 0: Disable digital filter function 1: Enable data filtering	Note: The NFE bit must hold the reset value for non-UART mode	R/W
			The NFE bit can only be set when TE=0 & RE=0 (transmit/receive disabled) Fractional Baud Rate		
b29	FBME	Fractional baud rate function enable	Function Enable Bit 0: Prohibited 1: Enabling	Note: FBME bit can only be set when TE=0 & RE=0 (Transmit/Receive Disable) MSB/LSB mode selection bit for UART mode/Clock Synchronization mode/Smart Card mode	R/W
			0: LSB mode 1: MSB approach	Note: The ML bit can only be set when TE=0 & RE=0 (transmit/receive disable)	
b27~b25	Reserved	-	Reads "0", writes "0".		R
b24	MS	Communication Mode Selection Bit	Communication Mode Selection Bit 0: UART mode 1: Clock synchronization mode		R/W
			Note: The MS bit can only be set when TE=0 & RE=0 (Transmit/Receive Disable), smart card mode MS needs to be written Input Reset Value		

RTOF clear bit			
b20	CRTOF	RTOF clear bit	0:Do not clear the RTOF flag 1:Clear RTOF flag
Note: CRTOF bit writes 1 to clear the RTOF flag and returns 0 when read			
ORE flag clear bit			
b19	CORE	ORE flag clear bit	0: Non-zero ORE flag 1: Clearing the ORE flag

Note: The CORE bit writes 1 to clear the ORE flag and returns 0 when read.				
b18	Reserved	-Reserved	Reads "0", writes "0".	R
b17	clear bit	CFEFE flag	FE flag clear bit 0:Not clearing the FE flag 1:Clear the FE flag Note: CFE bit writes 1 to clear the FE flag and returns 0 when read PE flag clear bit	R/W
b16	CPE	PE flag clear bit	0:Non-clear PE flag 1:Clear the PE flag Note: CPE bit writes 1 to clear the FE flag and returns 0 when read	R/W
b15	OVER8	UART oversampling mode	UART oversampling mode setting, i.e., one bit of base clock during data transfer 0: 16 bits Note: The OVER8 bit must hold the reset value in non-UART mode. The OVER8 bit can only be set when TE=0 & RE=0 (transmit/receive disable)	R/W
b14~b13	Reserved	-	Reads "0" and writes "0".	R/W
b12	M	Data Length Setting Position	Transmit/receive data length setting position in UART mode 0: 8 bits 1: 9 bits Note: The M bit must hold the reset value in non-UART mode. M bit can only be set when TE=0 & RE=0 (transmit/receive disabled)	R/W
b11	Reserved	-	Reads "0" and writes "0".	R
b10	PCE	Check Enable Bit	Parity enable bit in UART mode 0: no checksum 1: Calibration Note: The PCE bit must be 1 in smart card mode, and the PCE bit must hold a reset value in clock synchronization mode The PCE bit can only be set when TE=0 & RE=0 (transmit/receive disabled)	R/W
b9	PS	check digit	Parity selection bit when in UART mode 0: even check 1: odd calibration Note: The PS bit can only be set when TE=0 & RE=0 (transmit/receive prohibited), and the PS bit is only set when PCE=1 validity	R/W
b8	Reserved	-	Reads "0" and writes "0".	R
b7	TXEIE	Transmit Data Register Empty Interrupt Enable Bit	Transmit Data Register Air Break Enable Bit 0: TI interrupt request is invalid, TI interrupt does not occur 1: TI interrupt request is valid, TI interrupt occurs Note: TXEIE bit and TE bit should be written to 1 at the same time	R/W
b6	TCIE	Transmit Completion Interrupt Enable Bit	Request enable bit in transmit completion 0: TCI interrupt request is invalid, TCI interrupt does not occur 1: TCI interrupt request enable, TCI interrupt occurs	R/W

b5	RIE	Receive Interrupt Enable Bit	0: Receive interrupt request is invalid, RI and REI interrupts do not occur 1: Receive interrupt request is valid, RI and REI interrupts occur	R/W
b4	SLME	Silent Mode Enable Bit	Silent mode enable bit for multiprocessor operation 0: Normal mode 1: Silent mode When SLME=1, communication data with MPB bit 0 will not be read from the receive shift register to receive data registers device, while the error flag ORE and FE bits are not set. When ID data with MPB of 1 is received, the SLME self	R/W

<p>The action is cleared to zero and normal data reception action begins. Note: The SLME bit is only valid in UART multiprocessor mode; in other modes this bit must remain at its reset value.</p>			
b3	TE	Transmitter	Transmitter enable bit 0: Transmitter disabled 1: Transmitter enable Note: The TE bit can only be written 1 when TE=0 & RE=0
b2	RE	Receiver enable bit	(transmit/receive disable) in clock synchronization mode. receiver enable bit 0: Receiver disable 1: Receiver Enable Note: The RE bit can only be written 1 when TE=0 & RE=0 (transmit/receive disabled) in clock synchronization mode.
b1	RTOIE	UART TIMEOUT interrupt enable	UART TIMEOUT interrupt enable bit 0: UART TIMEOUT interrupt enable TIMEOUT interrupt request is invalid, RTOI interrupt does not occur 1: UART TIMEOUT interrupt request is valid, RTOI interrupt occurs. Note: RTOIE needs to be set to 1 at the same time as RTOE, or set to 1 after RTOE=1
b0	RTOE	UART TIMEOUT function enable	UART TIMEOUT function enable bit 0: UART TIMEOUT function disabled TIMEOUT function enable Note: Before RTOE=1 is set to 1, the USARTn.SR.RTOF flag needs to be cleared and the corresponding pass is stopped.
Timer0 counting function			

25.5.6 Control Register 2 (USART_CR2)

USART Control Register 2

Offset Address: 0x10

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
-	-	STOP	CLKC[1:0]	-	-	-	-	-	-	-	-	-	-	-	MPE

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
---------------------------------------	---------	-----------	---------------	---

b31~b14	Reserved	-	Reads "0", writes "0".	R
			When in UART mode, the stop bit length is set. 0: 1 stop bit	
b13	STOP	Stop Bit Setting Position	1:2 stop bits Note: The STOP bit must hold the reset value in non-UART mode. The STOP bit can only be set when TE=0 & RE=0 (transmit/receive disabled)	R/W
			UART mode 00b: the clock source is the clock generated by the internal baud rate generator, the clock is not output to USARTn_CK pin, the USARTn_CK pin can be used as a normal IO 01b: Clock source is the clock generated by the internal baud rate generator, clock output to USARTn_CK tube pin, the output clock frequency and baud rate are the same 10b or 11b: the clock source is an external input clock, the frequency of the input clock is 16 times the baud rate (OVER8=0) or 8 times (OVER8=1) Clock Synchronization Mode 00b or 01b: The clock source is the clock generated by the internal baud rate generator and output to the USARTn_CK pin 10b or 11b: the clock source is an external input clock with the same frequency and baud rate as the input clock smart card mode 00b: the clock source is the clock generated by the internal baud rate generator, the clock is not output to the CK pin, the CK pin Legs can be used as normal IO 01b: Clock source is the clock generated by the internal baud rate generator, clock output to CK pin 10b or 11b: Set prohibition Note: CLKC[1:0] bits can only be set when TE=0 & RE=0 (transmit/receive disabled)	

b12~b11	CLKC[1:0]	Clock control bits	00b or 01b: The clock source is the clock generated by the internal baud rate generator and output to the USARTn_CK pin 10b or 11b: the clock source is an external input clock with the same frequency and baud rate as the input clock smart card mode 00b: the clock source is the clock generated by the internal baud rate generator, the clock is not output to the CK pin, the CK pin Legs can be used as normal IO 01b: Clock source is the clock generated by the internal baud rate generator, clock output to CK pin 10b or 11b: Set prohibition Note: CLKC[1:0] bits can only be set when TE=0 & RE=0 (transmit/receive disabled)	R/W
b10~b1	Reserved	-	Reads "0" and writes "0".	R

Multi-processor function enable bit in UART mode

			0: Prohibited	
b0	MPE	Multi-Processor Function Enable Bit	1: Enabling	R/W
			Note: The MPE bit must hold the reset value in non-UART mode. MP bit can only be set when TE=0 & RE=0 (transmit/receive disabled)	

25.5.7 Control Register 3 (USART_CR3)

USART Control Register 3

Offset Address: 0x14

Reset value: 0x0000_0000

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

classifier for marking honorific people	marking	celebrity	functionality	fill out or in (information on a form)																
b31~b24	Reserved	-	Reads "0", writes "0".	R																
b23~b21	BCN[2:0]	Basic Clock Number	<p>When in smart card mode, set the number of base clocks during one bit of data transmission</p> <p>BCN[2:0] set values Basic clock number during one bit data transmission</p> <table> <tr><td>000b</td><td>32</td></tr> <tr><td>001b</td><td>64</td></tr> <tr><td>010b</td><td>set up a prohibition</td></tr> <tr><td>011b</td><td>128</td></tr> <tr><td>100b</td><td>Setting Prohibition</td></tr> <tr><td>101b</td><td>256</td></tr> <tr><td>110b</td><td>372</td></tr> <tr><td>111b</td><td>Setting Prohibition</td></tr> </table> <p>Note: The BCN[2:0] bits must hold the reset value in non-smart card mode. The BCN[2:0] bits can only be set when TE=0 & RE=0 (transmit/receive disabled)</p>	000b	32	001b	64	010b	set up a prohibition	011b	128	100b	Setting Prohibition	101b	256	110b	372	111b	Setting Prohibition	R/W
000b	32																			
001b	64																			
010b	set up a prohibition																			
011b	128																			
100b	Setting Prohibition																			
101b	256																			
110b	372																			
111b	Setting Prohibition																			
b20~b10	Reserved	-Reserved	Read "0", write "0".	R																
b9	CTSE	CTS function enable bit	<p>CTS function enable bit</p> <p>0: RTS function 1: CTS function</p> <p>Note: The CTSE bit can only be set when TE=0 & RE=0 (transmit/receive disabled)</p>	R/W																
b8~b6	Reserved	-	Reads "0" and writes "0".	R																
b5	SCEN	Smart Card Mode Enable Bit	<p>Smart Card Mode</p> <p>Enable Bit 0:</p> <p>0: Disable Smart Card Mode 1: Enable smart card mode</p> <p>Note: The SCEN bit must hold a reset value when in non-smart card mode</p> <p>The SCEN bit can only be set when TE=0 & RE=0 (transmit/receive disabled)</p>	R/W																
b4~b0	Reserved	-Reserved	Read "0", write "0".	R																

25.5.8 Prescaler register (USART_PR)

USART prescaler register

offset address: 0x18

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
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	PSC[1:0]

classifier	marking	celebrity	functionality	fill out or in (informat ion on a form)
for honorific people				
b31~b2	Reserved	-	Reads "0", writes "0".	R
b1~b0	PSC[1:0]	prescaler value	Prescaler divider value selection bit when internal clock source 00: PCLK 01: PCLK/4 10: PCLK/16 11: PCLK/64 Note: PSC[1:0] bits can only be set when TE=0 & RE=0 (transmit/receive disabled)	R/W

25.6 Precautions for use

25.6.1 UART Precautions

transmitters

When the UART mode transmitter transmits forbidden (USARTn_CR1.TE=0), then the TX pin can be used as a normal IO and the output value and direction can be set. If 0 is output, it causes the receiver to generate a framing error, which interrupts the data transmission. If 1 is output, the receiver cannot detect the start bit and thus cannot start data transmission.

refraction

When the UART mode generates a frame error, you can software detect if the subsequent RX line is low to determine if the sender wants to interrupt the transmission.

If the receive data start bit detection method is low level detection, continue to receive all low level data after clearing the error flag and the receive error will occur again.

25.6.2 Clock Synchronization Mode Notes

- 1) When using an external input clock to send data, the update of USARTn_TDR.TDR needs to be done before the clock is input, and after writing the data, you need to wait for at least one bit of data time before inputting the clock.
- 2) When sending data continuously, the next frame of data needs to be updated before the last bit of the current frame is sent.

25.6.3 Other considerations

- 1) To prevent the TX communication line Hi-Z state when transmission is disabled, the following method can be used:
 - Communication line pull-up
 - At the end of sending data, set the TX pin to normal IO output before USARTn_CR1.TE=0
 - Before sending data, set IO to TX function after USARTn_CR1.TE=1.

26 Integrated Circuit Bus (I2C)

26.1 summary

I2C (Integrated Circuit Bus) is used as an interface between the microcontroller and the I2C serial bus. Provides multi-master mode function to control the protocol and arbitration of all I2C buses. Supports standard mode, fast mode. The SMBus bus is also supported.

I2C Main Features.

- 1) I2C bus mode and SMBUS bus mode are selectable. Host mode, slave mode selectable. Various setup times, hold times, and bus idle times corresponding to the transmission rate are automatically ensured.
- 2) Maximum 100Kbps in standard mode and 400Kbps in fast mode.
- 3) Automatically generates start conditions, restart conditions, and stop conditions, and can detect start conditions, restart conditions, and stop conditions for the bus.
- 4) 2 slave mode addresses can be set. 7-bit address format and 10-bit address format can be set at the same time. Broadcast call address, SMBus host address, SMBus device default address, and SMBus alarm address can be detected.
- 5) The answer bit can be determined automatically when sending. The answer bit can be automatically sent when receiving.
- 6) Handshake function.
- 7) Arbitration function.
- 8) Timeout function to detect when the SCL clock has stopped for a long period of time.
- 9) SCL inputs and SDA inputs have built-in digital filters with programmable filtering capability.
- 10) Communication Error, Receive Data Full, Transmit Data Empty, End of One Frame Transmission, Address Matching Consistent Interrupt.

26.2 I2C System Block Diagram

26.2.1 system block diagram

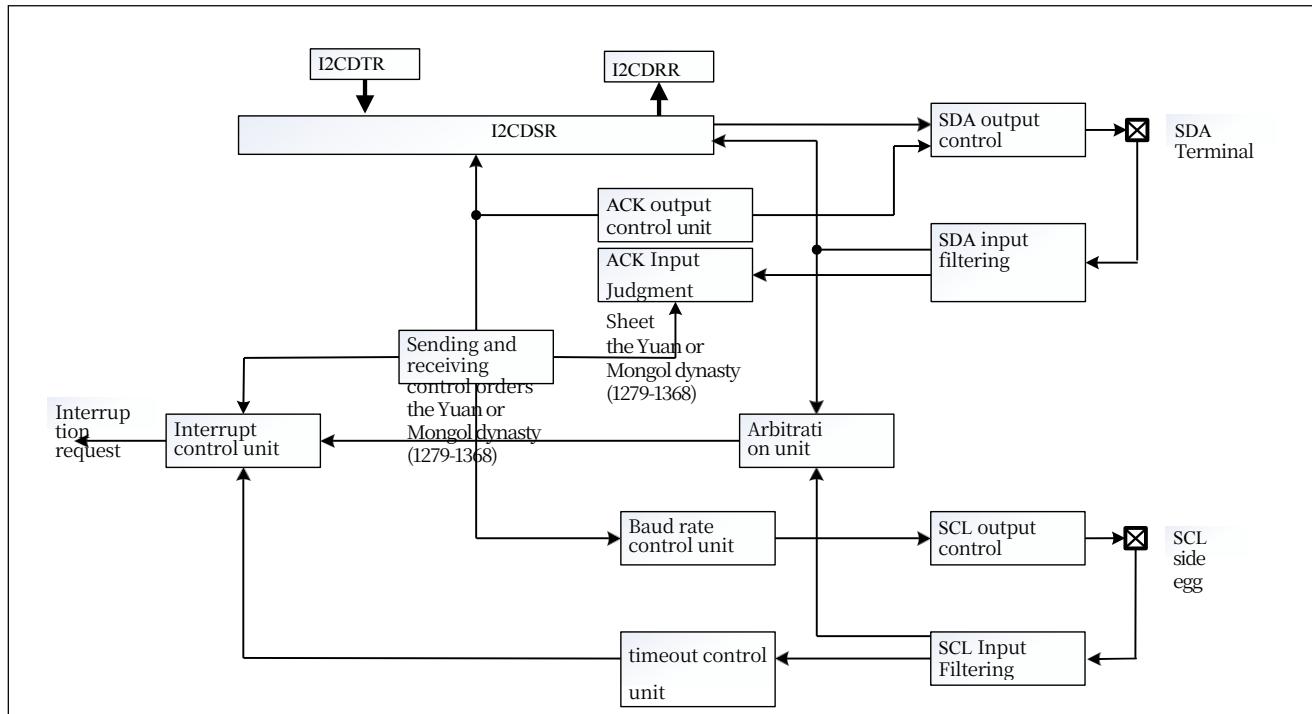
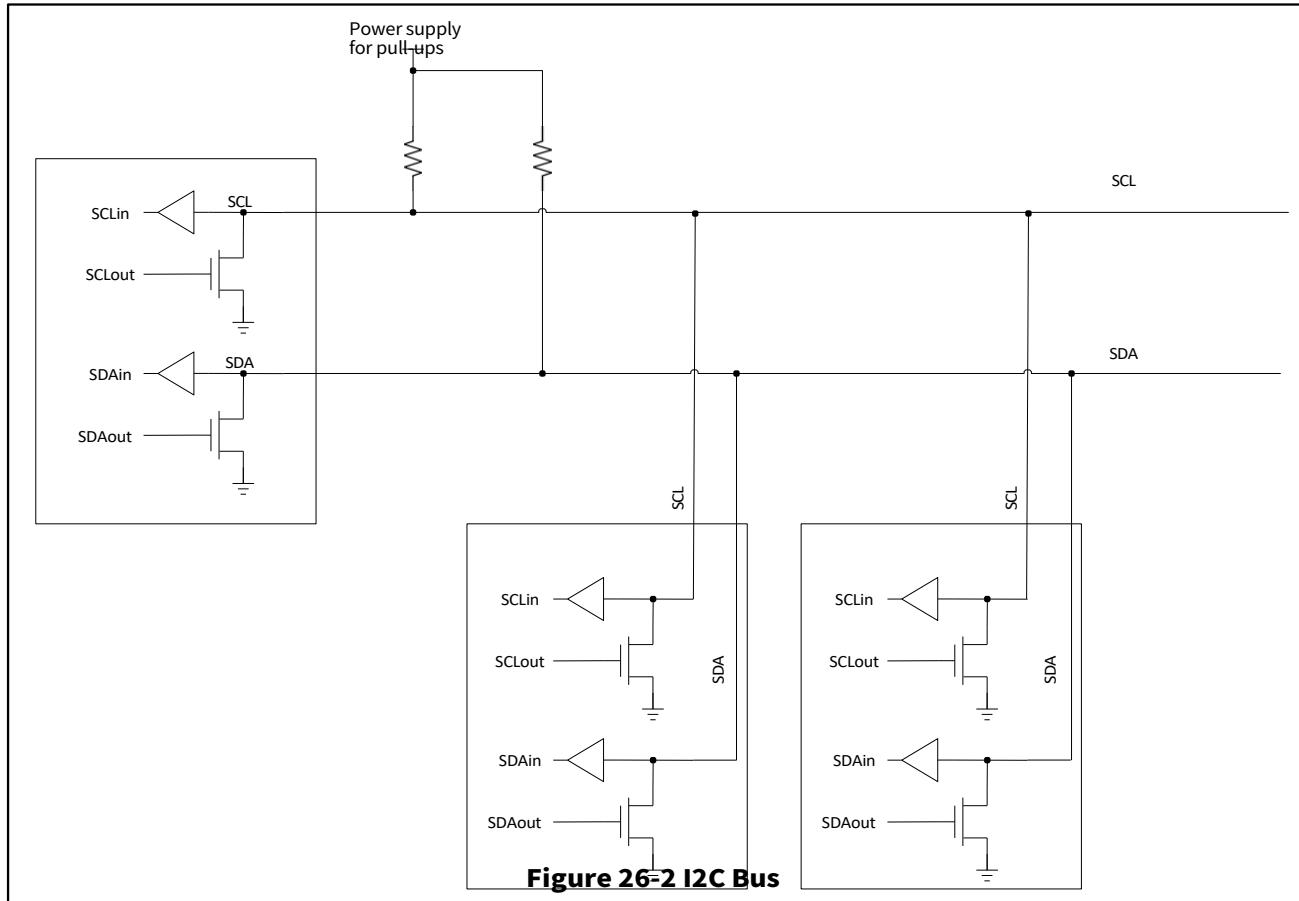


Figure 26-1 I2C System Block Diagram

26.2.2 structure plan



Structure Example

Table 26-1

Input/Output Pins

pinout	Input/Output	functionality
SCL	Input/Output	Serial clock input/output pins
SDA	Input/Output	Serial clock input/output pins

When I2C bus is selected, the SCL/SDA input levels are Schmitt levels. When SMBus is selected, the SCL/SDA input levels are CMOS levels (TTL compatible)

26.3 Action Description

This section provides a description of the I2C module functions.

26.3.1 I2C protocol

The I2C bus consists of one clock line (SCL) and one data line (SDA). All connected devices must be open drain outputs. pull-up resistors are connected externally to the SCL and SDA lines. The resistor values depend on the system application.

Typically, a complete communication process consists of the following 4 parts:

1. commencement condition
2. address transmission
3. data transmission
4. stop condition

The following figure shows the timing diagram of the I2C bus.

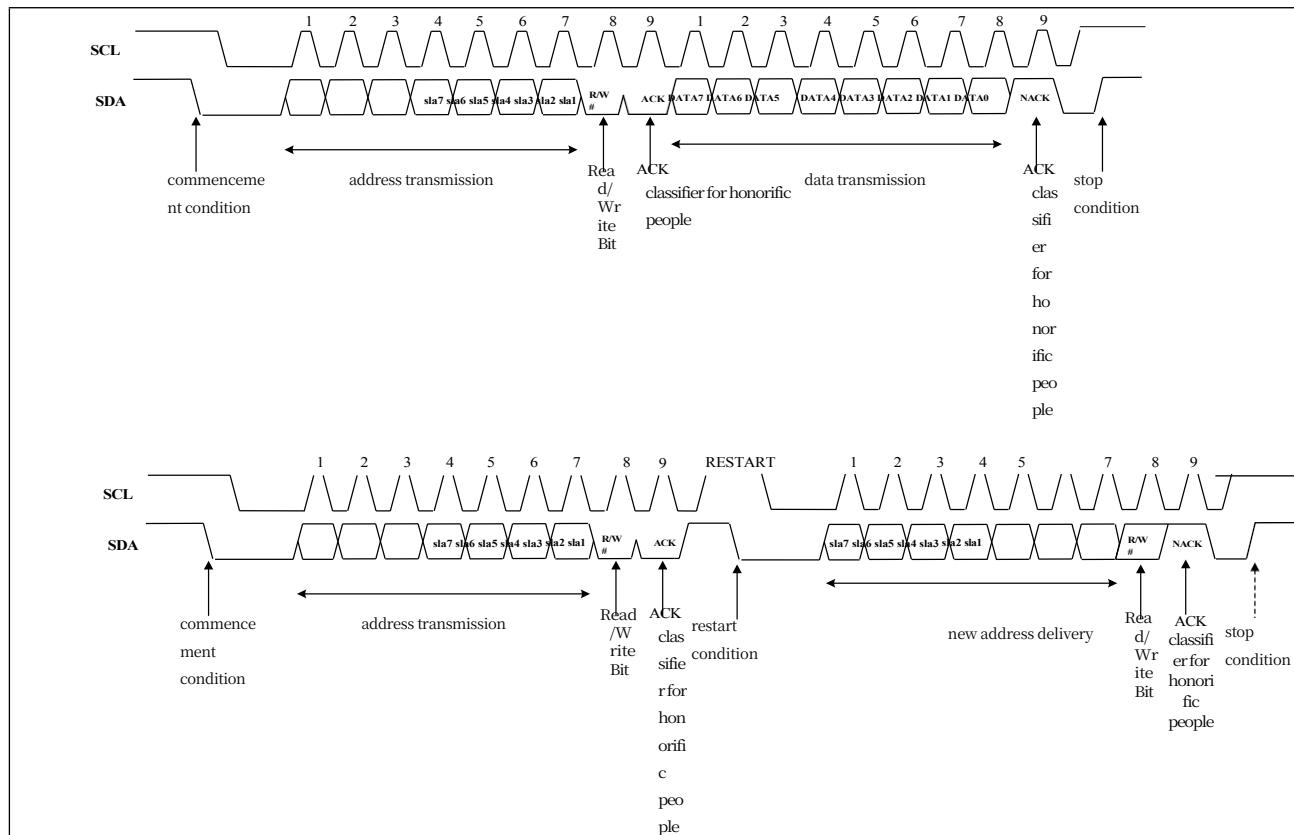


Figure 26-3 Timing Diagram of I2C Bus

26.3.1.1 commencement condition

and SDA are high. Any device on the bus in the idle state can initiate communication by sending a start condition.

If the START is set to "1" when the I2C_SR.BUSY flag is "0" (bus idle), a start condition is issued. If a start condition is detected, the I2C_SR.BUSY flag and the I2C_SR.STARTF flag are automatically set to "1" and the START bit is automatically set to "0". In this case, if the SDA signal and the signal state of the SDA line sent in the START bit "1" state are the same and the start condition is detected, the start condition is considered to be correctly issued by the START bit, and the start condition is automatically set to "1" after the I2C_SR.MSL bit and the I2C_SR.TRA bit are set to "0". "1" to master transmit mode. In addition, I2C_SR.TEMPTYF is automatically changed to "1" because the TRA bit is "1". Next, write the slave address into the I2C_DTR register and send the address.

26.3.1.2 address transmission

The frame following the start condition or restart condition is an address frame, which is used to specify the address of the object with which the host is communicating. The specified slave remains valid until a stop condition is sent.

The high 7 bits of the address frame are the slave address. Bit 8 of the address frame determines the direction in which the data frame is transmitted.

- 1) The 7-bit addressing mode is shown in the following figure [7-bit address format].

Host transmit mode, host transmit address frame bit 8 is 0

Host receive mode, host transmit address frame bit 8 is 1

- 2) The 10-bit addressing mode is shown in the following figure [10-bit address format]

In host transmit mode, the host sends the header sequence (11110XX0, where XX denotes the high two bits of a 10-bit address) the first frame, then the lower eight slave addresses in the second frame.

In host receive mode, the host sends a header sequence (11110XX0, where XX indicates the high two bits of a 10-bit address) the first frame, then the low eight slave addresses in the second frame. A restart condition is sent next, followed by another frame of the header sequence

(11110XX1, where XX represents the upper two digits of a 10-bit address)

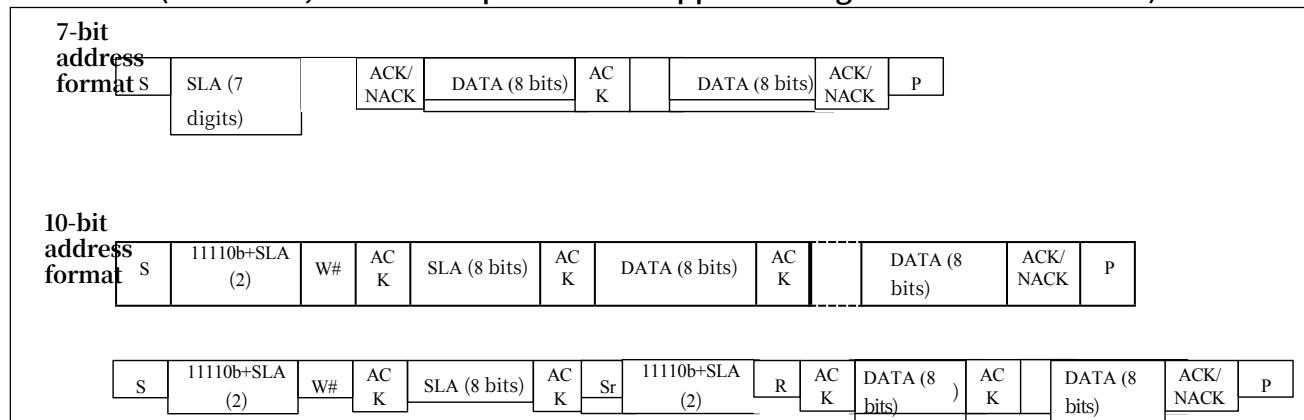


Figure 26-4 I2C Bus Data Format

S : Indicates the start condition. SLA :

Indicates the slave address.

R/W# R/W#: Indicates the direction of sending and receiving. When R/W# is "1", data is sent from the slave to the master; when R/W# is "0", data is sent from the master to the slave.

Sr : Indicates a restart

condition. DATA :

Indicates sent and received data

P : Indicates a stop

condition.

26.3.1.3 data transmission

After the address match is consistent, the hosts on the bus transmit data frame by frame according to the direction defined by R/W.

All data transmitted after an address frame is considered a data frame. Even addresses in the lower 8 bits of the 10-bit address format are considered data frames.

The length of the data frame is 8 bits. the low SDA of SCL changes, the high SDA of SCL is held, and one bit of data is sent per clock cycle. The 9th clock after the data frame is the answer bit, which is a handshake signal from the receiver to the sender.

If the slave on the bus receives data and does not respond to the host on the 9th clock cycle, the slave must send a NACK. if the host on the bus receives data and sends a NACK on the 9th cycle, the slave receives the NACK and the slave stops sending data.

Whether the host or the slave sends a NACK, the data transfer is terminated. The host can do either of the following:

- 1) Send stop condition to release the bus
- 2) Send a restart condition to start a new communication.

Host sends data

In the host transmit mode, the host outputs the SCL clock and sends data, and the slave receives the data and returns an answer. An example of the host transmit data runtime sequence is shown in the following figure.

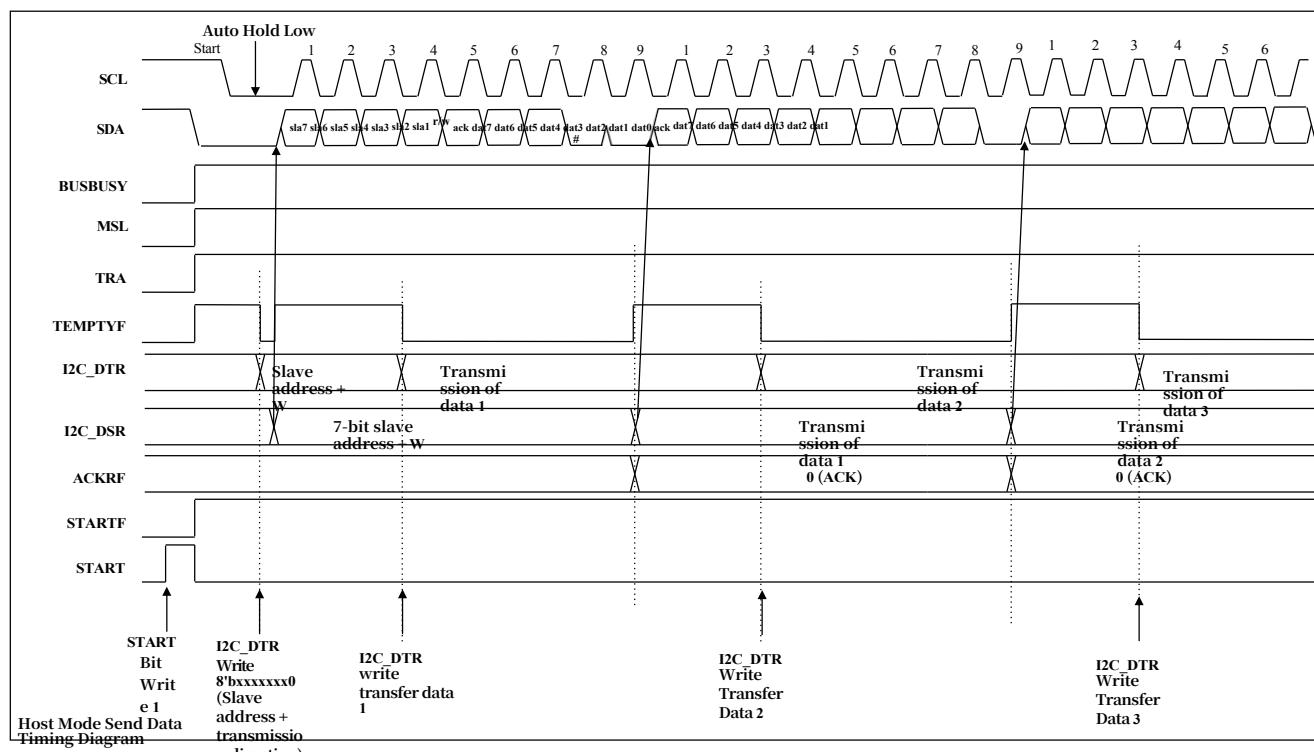


Figure 26-5 Timing Diagram for Host Transmit Data in 7-Bit Address Format (Example)

Host receives data

In the host receive mode, the host outputs the SCL clock, receives slave data and returns an answer. An example of the runtime sequence of the host receiving data is shown in the following figure.

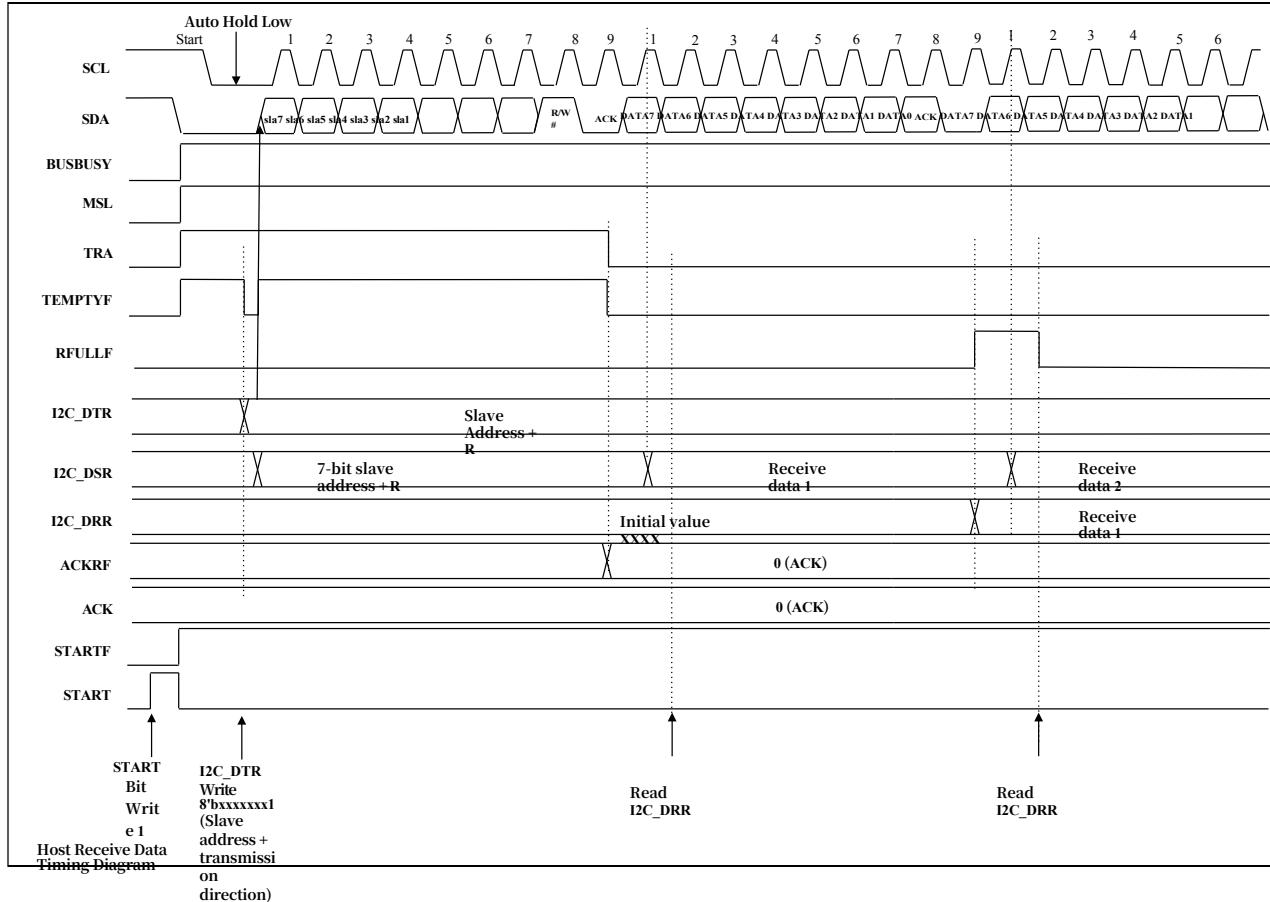


Figure 26-6 Timing Diagram of Host Receiving Data in 7-Bit Address Format (Example)

Slave sends data

In the slave transmit mode, the product receives the SCL clock from the host, sends data for the slave, and receives an answer back from the host. An example of the slave transmit data operation sequence is shown in the following figure.

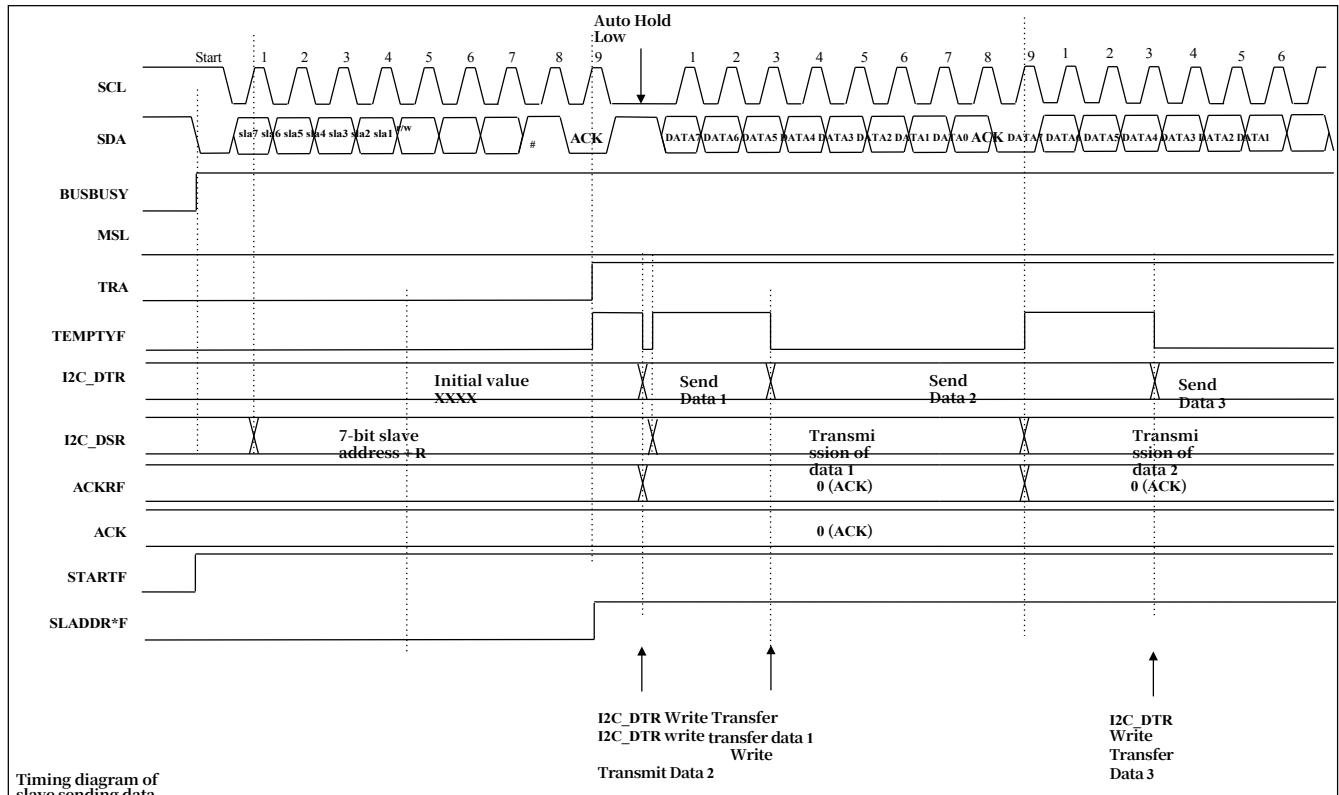


Figure 26-7 Slave Transmit Mode Timing Diagram for 7-Bit Address Format (Example)

Receive data from slave

In the slave receive mode, the SCL clock and data from the host are received, and an answer is returned after the data is received. An example of the slave receive data runtime sequence is shown in the following figure.

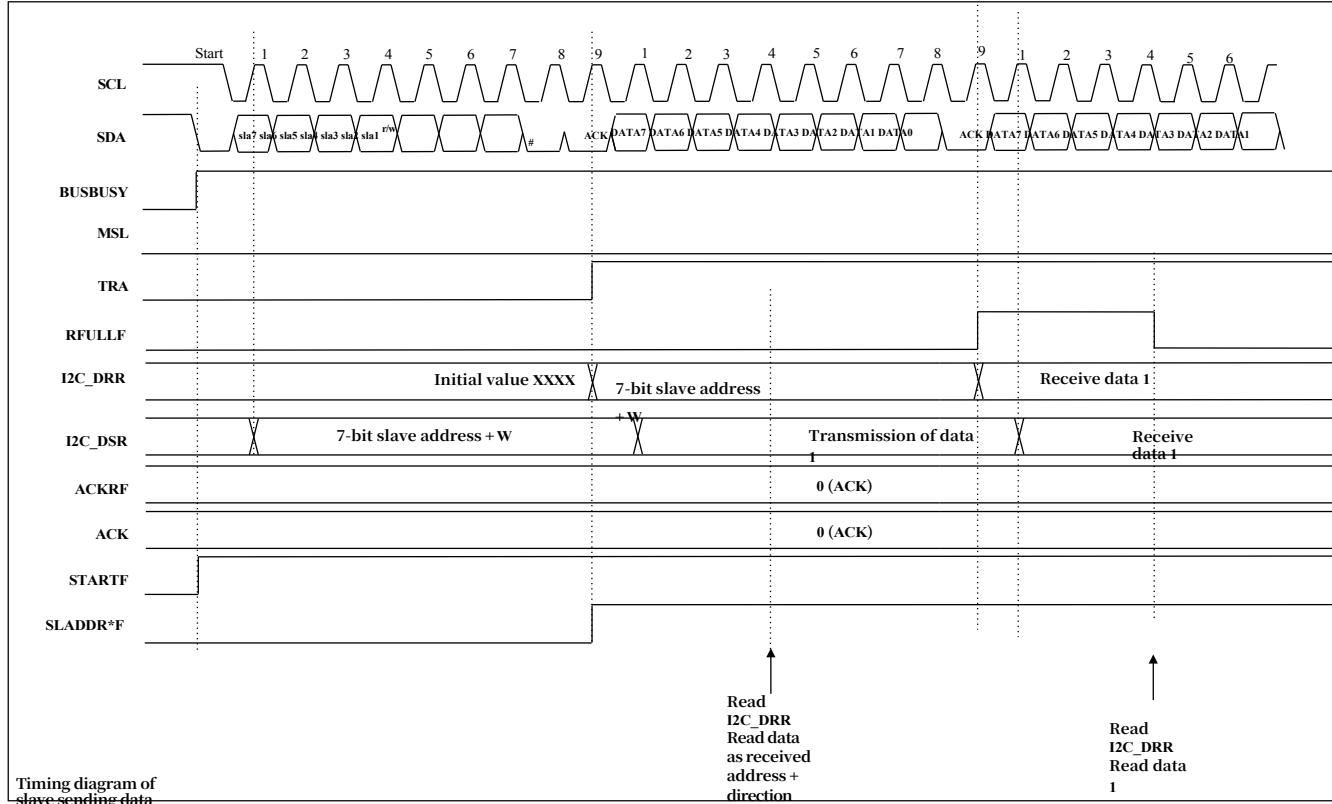


Figure 26-8 7-Bit Address Format Slave Receive Mode Timing Diagram

26.3.1.4 stop (Example) condition n

A stop condition is issued via the I2C_CR1.STOP bit.

STOP bit "1" with I2C_SR.BUSY flag is "1" (Bus Busy) and the I2C_SR.MSL bit is "1" (Host Mode), the stop condition is issued by setting the STOP position "1".

26.3.1.5 restart condition

A restart condition is generated with the I2C_CR1.RSTART bit.

With the I2C_SR.BUSY flag "1" (Bus Busy) and the I2C_SR.MSL bit "1" (Host Mode), RSTART position "1", a line restart condition is generated.

With the restart condition, the master can switch the transmit/receive mode without releasing the BUS right. It is also possible to establish communication with another slave without releasing the BUS right.

26.3.1.6 SCL clock synchronization

When using the I2C bus in multi-host mode, it is possible for the SCL clock to conflict with the SCL clock due to competition with other hosts. If the SCL clock conflicts, the host needs to synchronize with the SCL clock, and the SCL clock needs to be synchronized bit by bit. When a rising edge of the SCL line is detected and during counting of the high level set by the I2C_CCR.SHIGHW register, if the SCL line falls due to the SCL clock output of another host, the incremental counting of the high level width is aborted upon detection of the falling edge of the SCL line and counting of the low level set by the I2C_CCR.SLOWW register is started at the same time that the SCL line is driven low. SLOWW set low level width is incrementally counted, the low level drive of the SCL line is terminated at the end of the low level width count, and the SCL line is released. At this time, if the low-level width of the SCL clock of the other host is greater than the low-level width set by SLOWW, extend the low-level width of the SCL clock. When the other host ends the low-level output, the SCL line is released and the SCL clock rises. Therefore, in the event of an SCL clock output conflict, the high level width of the SCL clock is synchronized with the short clock and the low level width is synchronized with the long clock.

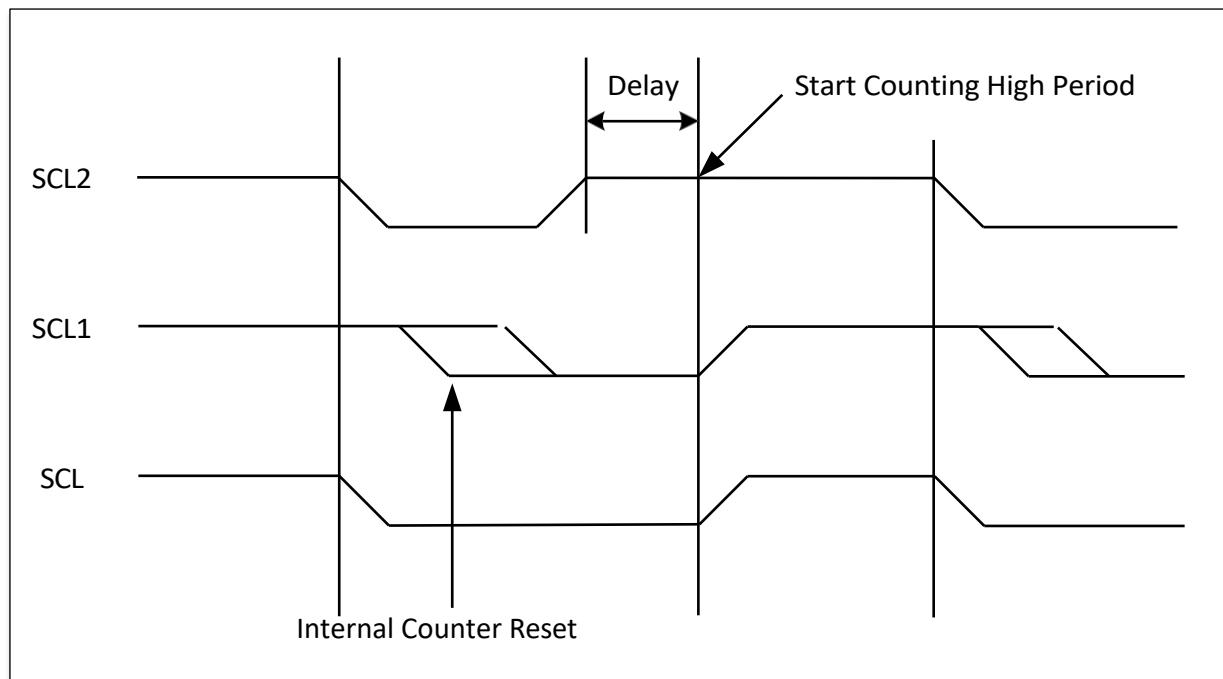


Figure 26-9 SCL Synchronization Timing

26.3.1.7 arbitraton

The I2C bus is a true multi-host bus, allowing multiple host connections.

If two or more hosts attempt to control the bus simultaneously, 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 data captured at high level determines the arbitration result. An arbitration failure occurs when the transmitted SDA output is high (SDA pin is in a high impedance state) and the SDA line is detected to be low. the I2C_SR.AROLF bit is set to "1" in hardware. The I2C_SR.AROLF bit is set to "1" in hardware. If a host arbitration failure occurs, it immediately transfers to the slave receive mode. At this point, if the slave address, including the broadcast address, matches, slave mode operation continues.

26.3.1.8 handshakes

Handshaking during data transfer is achieved through the SCL clock synchronization mechanism. The slave holds the SCL clock line low after transmitting a frame of data (containing ACK bits). In this case, the low level of the SCL clock puts the host into a wait state until the slave releases the SCL line.

[Slave Transmit Mode]

- 1) In transmit mode (I2C_SR.TRA bit = 1) if the shift register (I2C_DSR register) is empty and no transmit data (I2CDT register) has been written, the SCL line is automatically held low during the low level interval between the 9th clock and the 1st clock of the next transmission, with the action timing shown in the following figure.

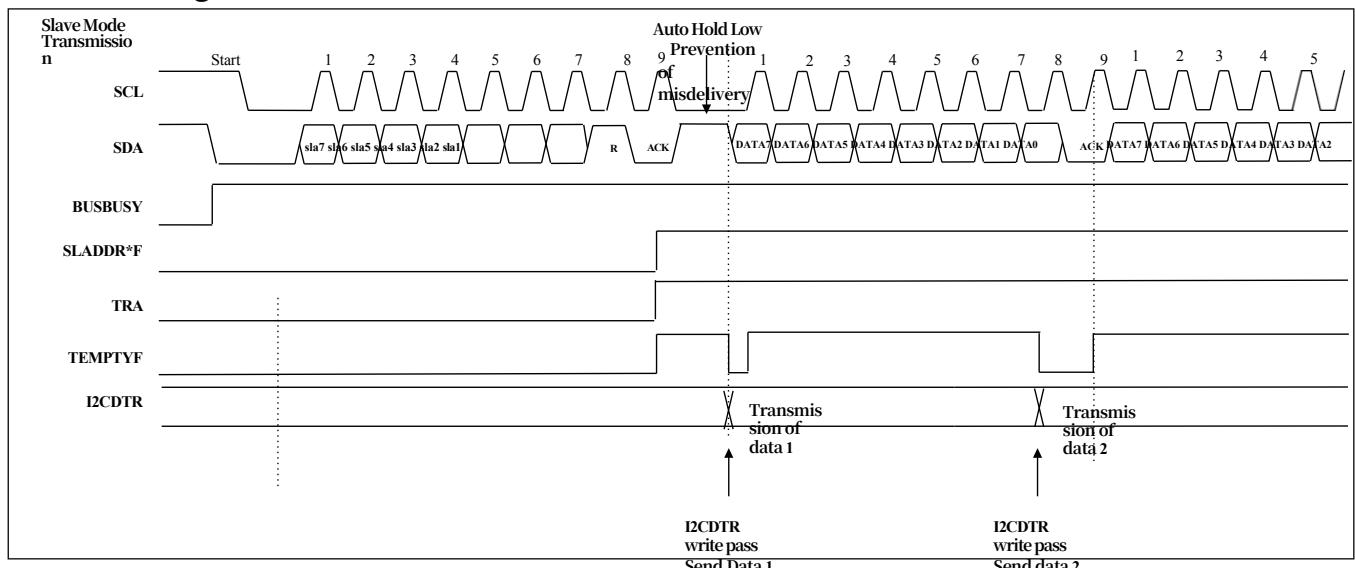


Figure 26-10 Slave Transmit Timing Diagram (1)

- 2) After the I2C_SR.NACKF flag changes to "1" or the last transmit data is written to the I2C_DTR register, wait until the I2C_SR.TEMPTYF flag changes to "1" **until the I2C_SR.TENDF flag changes to "1"**. When the I2C_SR.NACKF flag or the TENDF flag is "1", the SCL line is held low after the 9th clock fall. At this point the communication must be terminated by reading the I2C_DR register to release the SCL line.

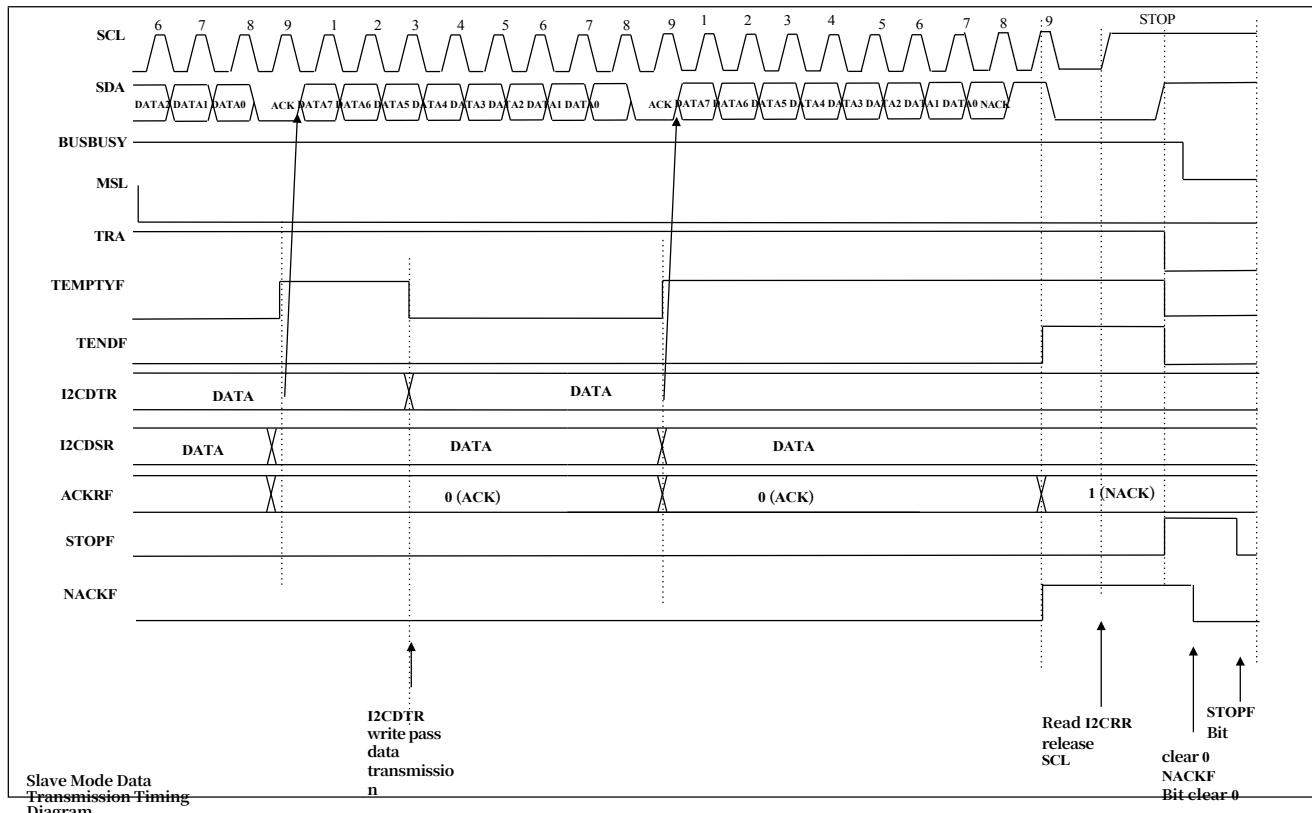
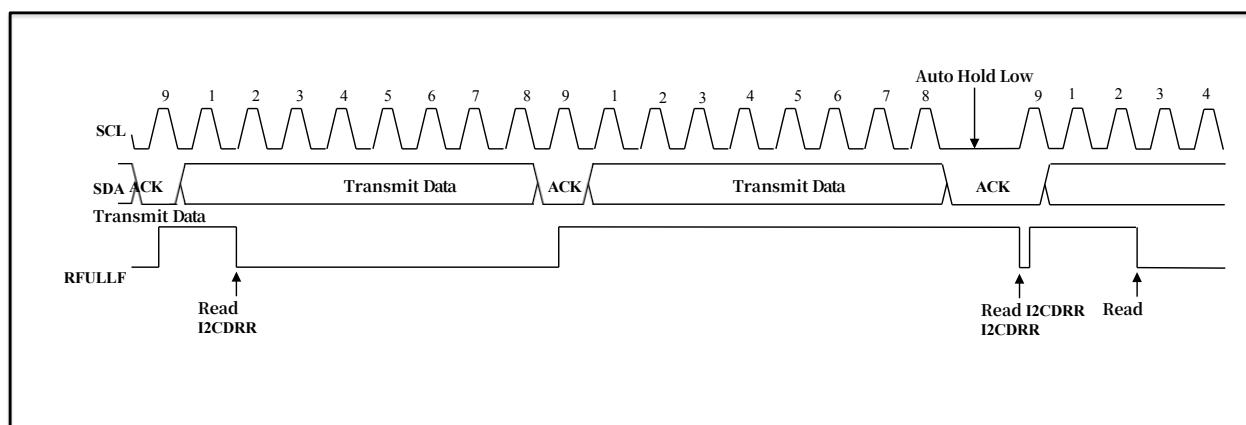


Figure 26-11 Slave Transmit Timing Diagram (2)

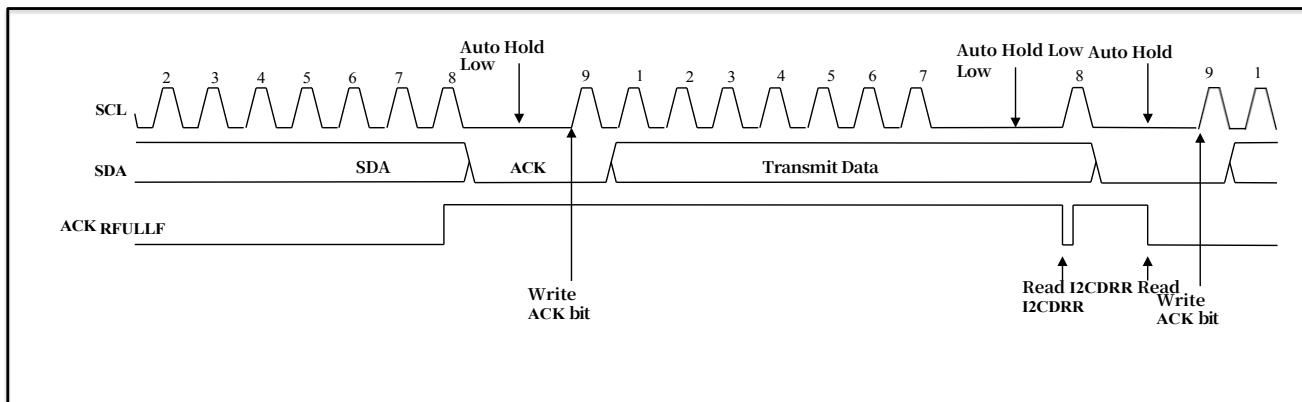
[Slave receive mode]

If a response processing delay occurs due to a delay of at least one transmission frame to read the received data (I2C_DRR register), etc. in the receive mode (I2C_SR.TRA bit = 0) and the received data is full (I2C_SR.RFULLF flag = 1), the SCL line is automatically held low between the 8th SCL and 9th SCL clocks before starting the next data reception as shown in the action sequence below. line is held low between the 8th SCL and 9th SCL clocks, and the action timing is shown below.



[Rapid ACK/NACK]

In SMBUS communication, the built-in CRC operator is used to calculate the SMBUS packet error code (PEC) or to check the received data. In the process of checking the PEC code, an ACK or NACK is sent in the last byte depending on whether it matches or not, which necessitates that the SCL be held low on the falling edge of the 8th clock of the SCL of the last byte received. This must be done to meet the software processing time. Based on the calculation, software writes the I2C_CR1.ACK bit to deassert SCL. Fast ACK/NACK is controlled by the I2C_CR3.FACKEN bit, and the timing is shown below.



26.3.2 address matching

As a slave, two types of addresses can be set in addition to the broadcast address and the host notification address, and the slave address can be set in 7-bit address or 10-bit address format.

26.3.2.1 Slave Address Matching

This I2C bus can set two types of slave addresses, and has a slave address detection function for each. When SLADDR1EN and SLADDR0EN are "1", the slave addresses set in I2C_SLR1 and I2C_SLR0 registers are detected.

If the set slave addresses match, set the corresponding SLADDR1F and SLADDR0F to "1" on the falling edge of the 9th clock of the SCL clock, and then set the I2C_SR.RFULLF flag or the I2C_SR.TEMPTYF flag to "1" in accordance with the subsequent R/W# bits. "TEMPTYF flag according to the subsequent R/W# bit. As a result, a receive data full interrupt or transmit data empty interrupt is generated, and the slave address specified can be determined by checking the I2C_SR.SLADDR1F and SLADDR0F flags.

I2C_SR. The timing of SLADDR1F and SLADDR0F flags changing to "1" is shown below.

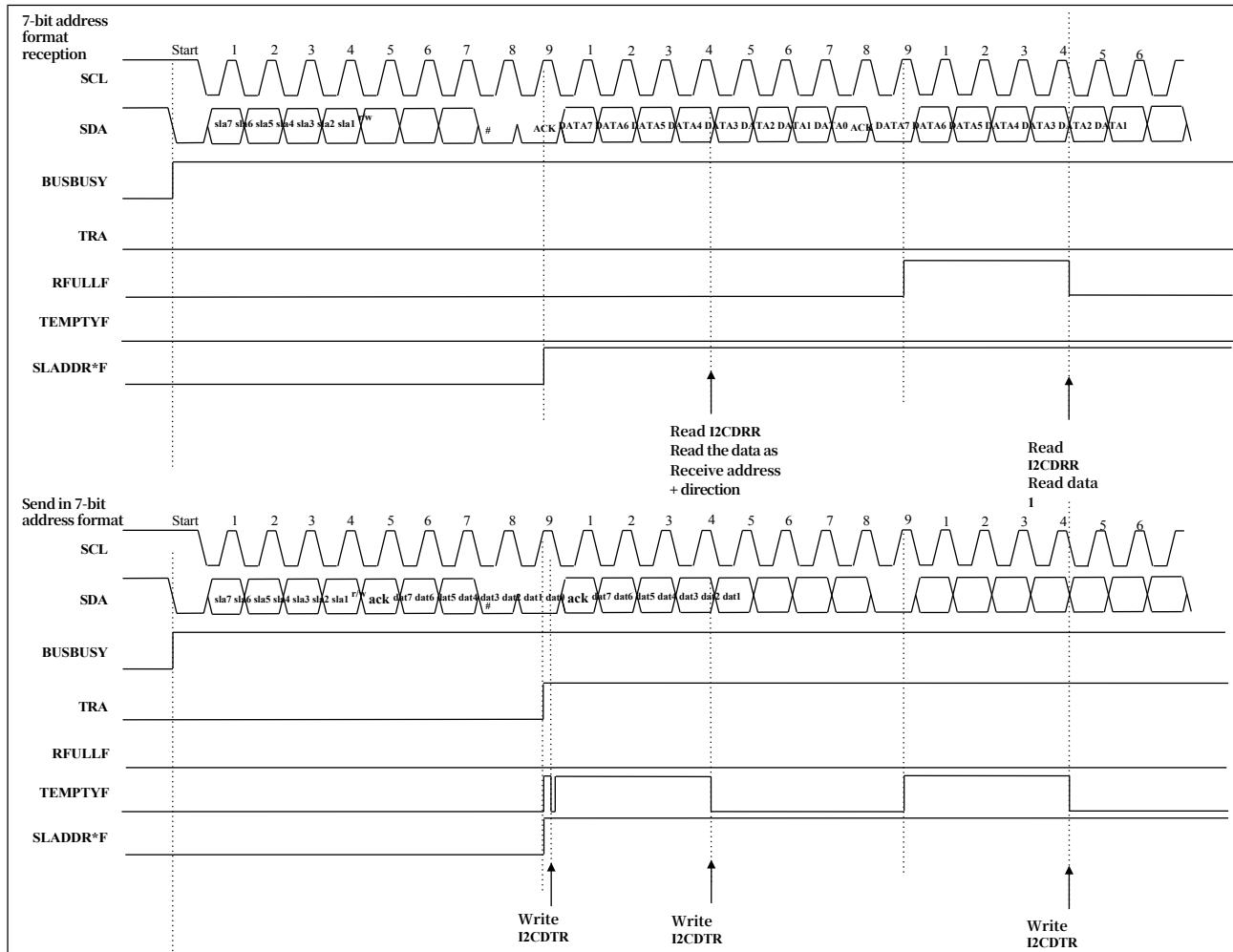


Figure 26-12 Timing when selecting 7-bit address format

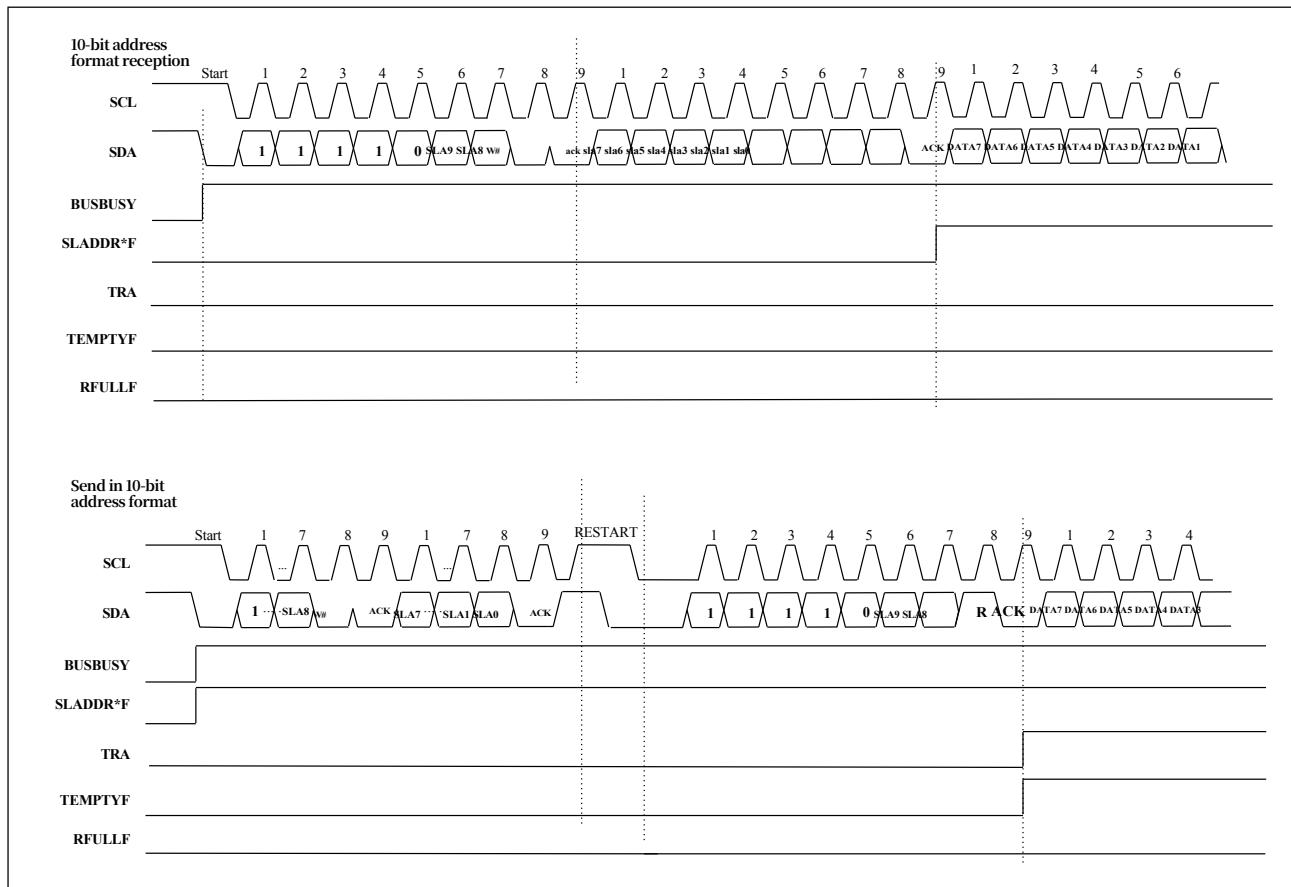


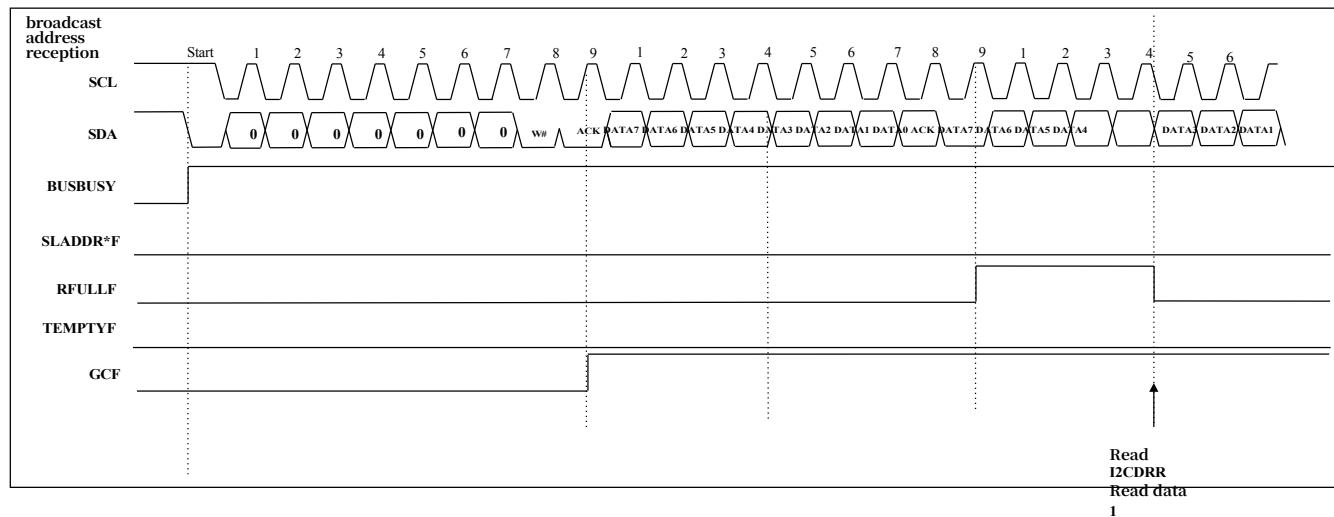
Figure 26-13 Timing when 10-bit Address Format is Selected

26.3.2.2 broadcast address matching

When the I2C_CR1.GCEN bit is "1", the broadcast address (0000 000b+0[W]) can be detected.

However, the address after the start condition or restart condition is 0000 000b+1[R] (start byte) this address is regarded as the slave address of All "0" and not as the broadcast address.

If the broadcast address is matched, the I2C_SR.GCF flag is set to "1" on the falling edge of the 9th clock of the SCL clock. The operation after matching the broadcast



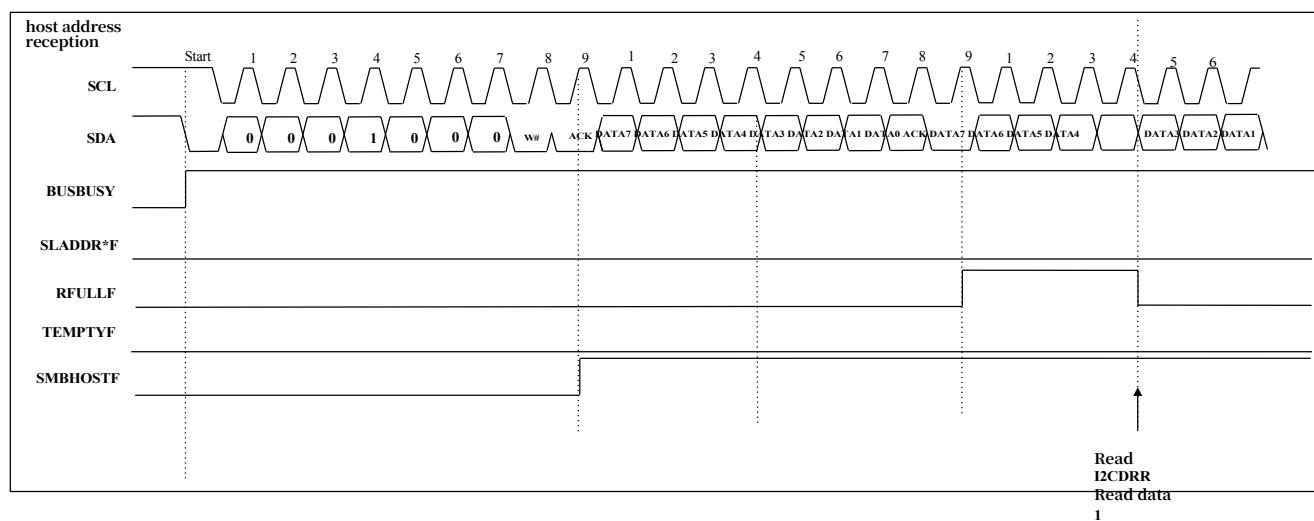
address is the same as the normal slave receive operation.

26.3.2.3 SMBus host address matching

This product has a host address detection function during SMBus operation. If the I2C_CR1.SMBUS bit is set to "1" when the I2C_CR1.SMBHOSTEN bit is "1", the host address (0001 000b) can be detected in the slave receive mode (I2C_CR1.MSL bit TRA bit is "00b"). MSL bit TRA bit is "00 b" to detect the host address (0001 000b) ~~host address (0001 000b)~~ (0001 000b)

If the SMBUS host address is detected, set the I2C_SR.SMBHOSTF flag to "1" on the falling edge of the 9th clock of the SCL clock.

The SMBUS host address can be detected even if the bit following the SMBUS host address (0001 000b) is a Rd bit (R/W# bit receives a "1") operation after SMBUS host address detection is the same as normal slave mode operation.



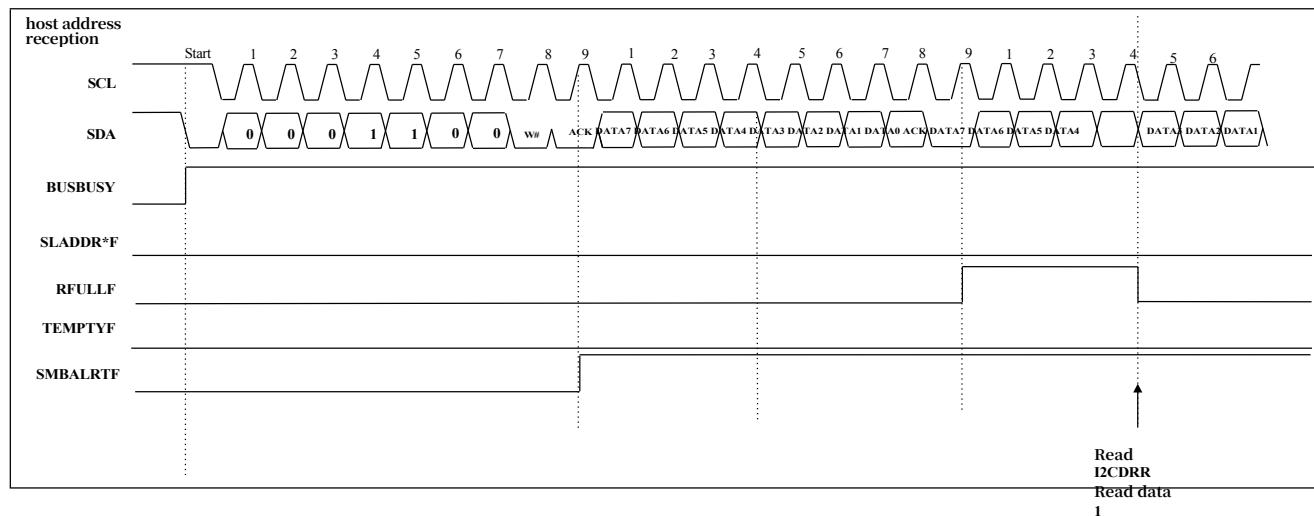
26.3.2.4 SMBus Alarm Response Address Matching

This product has an alarm response address detection function for SMBus operation. If I2C_CR1.SMBUS is set to "1" when the I2C_CR1.

SMBARLERTEN is "1" when the I2C_CR1.SMBUS bit is "1", can be detected in the slave receive mode (I2C_CR1.MSL bit TRA bit is "00b"). MSL bit TRA bit is "00b" in slave receive mode (I2C_CR1.MSL bit TRA bit is "0001 100b")

If the SMBUS alarm response address is detected, it is set at the SCL SMBARLERTF flag to "1" at falling edge of the 9th clock of the SCL clock if the SMBUS alarm address is detected.

Operation after SMBUS alarm response address detection is the same as normal slave mode operation.

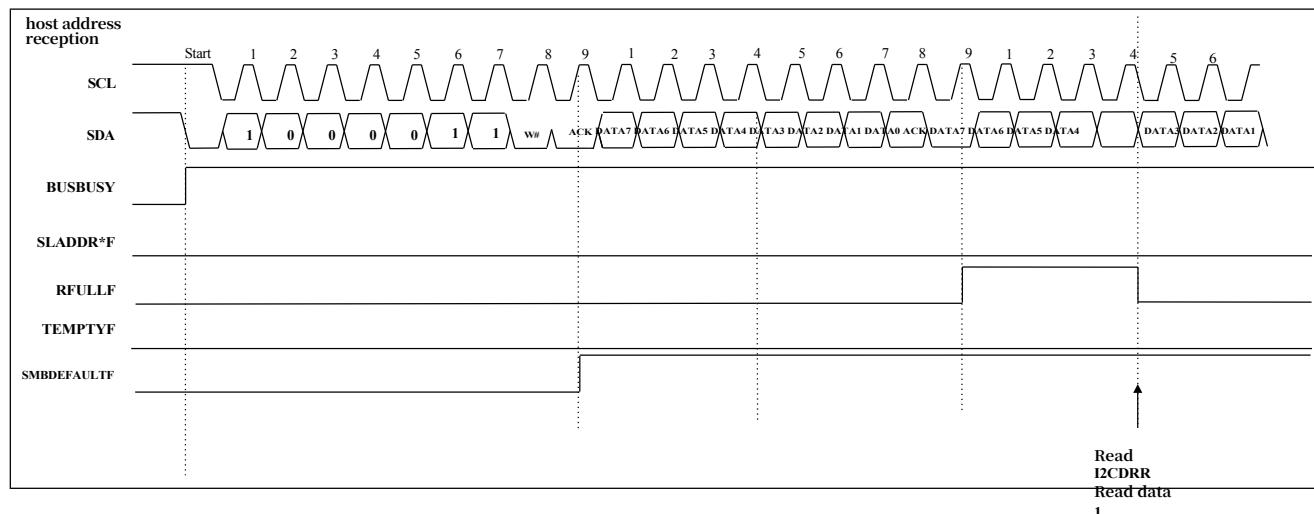


26.3.2.5 SMBus Default Address Matching

This product has a default address detection function for SMBus operation. If the I2C_CR1.SMBDEFAULTEN bit is "1" when the I2C_CR1.SMBUS bit is "1", it can be detected in the Slave Receive Mode (I2C_CR1.MSL bit TRA bit is "00b"). MSL bit TRA bit is "0 0 b " detects the SMBUS default address (1100 001b).

If the SMBUS default address is detected, set the I2C_SR.SMBDEFAULTF flag to "1" on the falling edge of the 9th clock of the SCL clock.

Operation after SMBUS default address detection is the same as normal slave mode operation.



26.3.3 SMBus Action

This I2C interface is capable of SMBus (Ver.2.0) based communication. To perform SMBus communication, I2C_CR1.SMBUS must be set to position "1". CKDIV[2:0] bits and the I2CCCR register must be set to set the transmission speed within the range of 10kbps to 100kbps of the SMBus specification.

26.3.3.1 SMBus Timeout Measurement

1) SCL Level Timeout Measurement

In the bus busy state, an abnormal state of the bus can be detected by detecting that the low or high level of the SCL line has been fixed for a certain period of time or more and by detecting an abnormal state of the bus.

The timeout detection function monitors the status of the SCL line and counts the time it has been high or low by means of an internal counter. If the SCL line changes (rises/falls) the internal counter is reset, otherwise the counting continues. If the internal counter reaches the TOUTHIGH/TOUTLOW setting while there is no change in the SCL line, a timeout is detected and the bus is notified of an abnormal condition.

For internal counter counting, the HTMOUT and LTMOUT bits can be set to select

whether to count in the low or high state of the SCL line, or in both the low and high states. If the HTMOUT and LTMOUT bits are set to "0", no internal counting is performed.

2) Slave timeout measurement

Slave devices for SMBus communication are required to measure the intervals shown below (timeout interval: TLOW:SEXT)

- Interval from start condition to stop condition

For timeout measurement via the slave device, the time from detection of the start condition to detection of the stop condition is measured using the Start Condition Detection Interrupt and Stop Condition Detection Interrupt and via the chip timer. This timeout measurement must be within the cumulative time of clock low [slave device] TLOW:SEXT: 25ms (max) of the SMBus specification.

If the timer measurement exceeds the detection of SMBus specification TTIMEOUT: 25ms (min) the slave will need to release the bus.

3) Timeout Measurement for Hosts

The master device for SMBus communication is required to measure the interval shown below (timeout interval: TLOW:MEXT)

- Interval from start condition to answer bit
- Interval from answer bit to next answer bit
- Interval from answer bit to stop condition

When the host performs timeout measurement, the time of each interval is measured by the chip timer using the start condition detection interrupt, stop condition detection interrupt, and end of transmission interrupt or receive data full interrupt. The measurement time of this timeout must be within the cumulative time of clock low of SMBus specification [host] TLOW:MEXT: 10ms (max), and the cumulative result of all TLOW:MEXT from start condition to stop condition must be within TLOW:SEXT: 25ms (max).

If the timer measurement exceeds the accumulated time for clock low detection of the SMBus specification [Master] TLOW:MEXT: 10ms (max) or if the accumulated result of each measurement exceeds the timeout for clock low detection of the SMBus specification TTIMEOUT: 25ms (min) the host is required to abort processing.

Transmission must be aborted immediately while the host is transmitting (write I2C_DTR register) Abort host processing by issuing a stop condition.

26.3.3.2 Packet Error Code (PEC)

In communication, the CPU is utilized to operate the CRC, send the packet error code (PEC) of SMBus or check the received data.

26.3.4 reset (a dislocated joint, an electronic device etc)

The communication module can be reset. There are two types of reset, one to initialize all registers including the ICCR2.BBSY flag, and the other is an internal reset to release the slave address matching state and initialize the internal counter while maintaining various set values.

After reset, I2C_CR1.SWRST must be set to position "0".

Since either reset disables the output state of the SCL pin/SDA pin to a high impedance state, it can also be used to disable the bus accidental stop state.

Reset in slave mode can cause desynchronization with the master device, so it should be avoided as much as possible. It is important to note that the bus status of the start condition, etc. cannot be monitored during a reset (I2C_CR1.PE bit and I2C_CR1.SWRST bit are "01b").

26.3.5 Interrupt and event signal output

I2C has 4 kinds of interrupts and event outputs for triggering the startup of other peripheral circuits for user selection. These include: communication error occurrence (arbitration failure detection, NACK detection, timeout detection, start condition detection, stop condition detection) end of receive, send data null, end of send.

The list of interrupts is shown below.

name (of a thing)	source of interruption	interrupt flag	interrupt condition
I2C_EEI	Communication errors/communication events	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
		GENDCALLF	GENDCALLF =1& GENDCALLIE=1
		NACKF	NACKF=1&NACKIE-1
		TMOUTF	TMOUTF=1&TMOUTIE=1
		STARTF	STARTF=1&STARTIE=1
		STOPF	STOPF=1&STOPIE=1
I2C_RXI	Receive data full	RFULLF	RFULLF=1&RFULLIE=1
I2C_TXI	Send data null	TEMPTYF	TEMPTYF=1&TEMPTYIE=1
I2C_TEI	Sending end	TENDF	TENDF=1&TENDIE=1

The event signal output list is shown below.

name (of a thing)	event source	event condition
I2C_EEI	Communication error/communication time	ARLOF=1
		SLADDR0F=1
		SLADDR1F=1
		SMBALRTF=1
		SMBHOSTF=1
		SMBDEFAULTF=1
		GENDCALLF=1
		NACKF=1
		TMOUTF=1
		STARTF=1
		STOPF=1
I2C_RXI	Receive data full	RFULLF=1
I2C_TXI	Send Data Empty	TEMPTYF=1
I2C_TEI	Sending end	TENDF=1

26.3.6 Programmable Digital Filtering

The state of the SCL pin and the SDA pin is internally accessed via the analog filter circuit and the digital filter. The block diagram of the digital filter circuit is shown below.

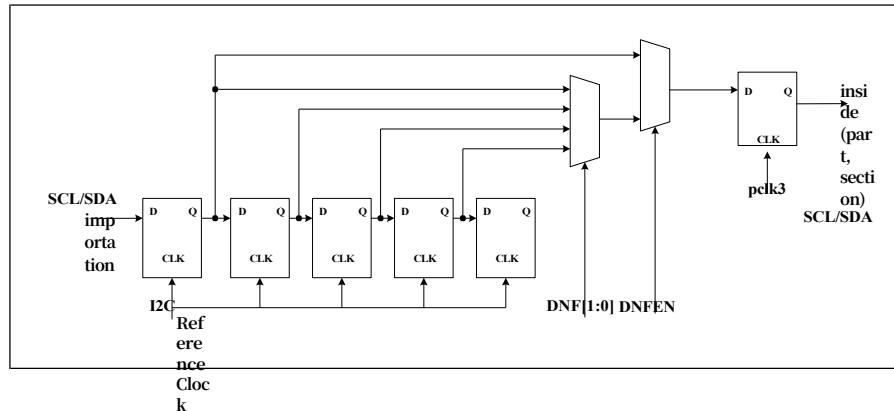


Figure 26-14 Digital Filter Circuit Block Diagram

The internal digital filter circuit consists of a 4-segment series-connected trigger circuit and a match detection circuit.

The number of active segments of the digital filter is selected with the I2C_FLTR.DNF bit, and the noise cancellation capability is 1 to 4 I2C cycles depending on the number of active segments selected.

The input signal to the SCL pin (or the input signal to the SDA pin) is sampled at the falling edge of the I2C internal clock, and if the output of the trigger circuit for the number of valid segments set by the I2C_FLTR.DNF bit matches all of the outputs, the level is transmitted as an internal signal; otherwise, it remains at its original value.

26.4 Application software sets up the I2C initialization process

When you start sending or receiving data, you must initialize the steps shown below.

1. The PE bit is set to 0;
2. SWRST Set to 1, communication reset;
3. The PE bit is set to 1 to reset the internal state;
4. Sets the slave address format and address;
5. Sets the baud rate;
6. Set the control register functions and interrupts as required;
7. The SWRST bit is set to 0 to release the internal state reset;
8. End of initialization. Received data can be sent.

26.5 Register Description

I2C1 base address:

0x4004_E000 I2C2 base

address: 0x4004_E400 I2C3

base address: 0x4004_E800

Table 26-2 Register List

register name	notation	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 Register 0	I2C_SLR0	0x10	32	0x0000_1000
I2C Slave Address Register 1	I2C_SLR1	0x14	32	0x0000_0000
I2C Status Register	I2C_SLTR	0x18	32	0xFFFF_FFFF
I2C Status Register	I2C_SR	0x1C	32	0x0000_0000
I2C Status Register	I2C_CLR	0x20	32	0x0000_0000
I2C Data Transmit Register	I2C_DTR	0x24	8	0xFF
I2C Data Receive Register	I2C_DRR	0x28	8	0x00
I2C Baud Rate Control Register	I2C_CCR	0x2C	32	0x0000_1F1F
I2C Baud Rate Control Register	I2C_FLTR	0x30	32	0x0000_0010

26.5.1 I2C Control Register 1 (I2C_CR1)

Reset value: 0x0000_0040

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SWR ST	-	-	-	-	ACK	STOP	STA RT	RES TART	ENG C	-	SMB HOST EN	SMBD EFAU LTEN	SMB ALRT EN	SMB US	PE

classifier for marking honorific people			celebrity	functionality	fill out or in (information on a form)									
b31-16 Reserved			-	Reads "0", writes "0".	R/W									
				0: Unreset 1: Software reset Combination of this bit and the PE bit selects internal status reset or communication reset										
b15	SWRST	Software Reset		<table border="1"> <tr> <th>SWRST</th><th>PE</th><th>Reset content</th></tr> <tr> <td>1</td><td>0</td><td>Communication reset: all internal I2C registers and internal state reset.</td></tr> <tr> <td>1</td><td>1</td><td>Internal state reset: I2C_SR, I2C_DSR registers and internal state machine into the line reset (computing)</td></tr> </table>	SWRST	PE	Reset content	1	0	Communication reset: all internal I2C registers and internal state reset.	1	1	Internal state reset: I2C_SR, I2C_DSR registers and internal state machine into the line reset (computing)	R/W
SWRST	PE	Reset content												
1	0	Communication reset: all internal I2C registers and internal state reset.												
1	1	Internal state reset: I2C_SR, I2C_DSR registers and internal state machine into the line reset (computing)												
b14-11	Reserved	-Reserved		Read "0", write "0".	R/W									
b10 (ACK)	ACK	Send answer		0: Send "0" for answer bit (send	R/W									
				1: Send "1" for the answer bit (send NACK)										
b9	STOP	Stop condition generation bit		0: No stop condition is generated 1: Generation of stopping conditions This bit can be set to 1 and cleared to 0 by software. Hardware Clear 0 Condition: Stop condition detected Arbitration fails Start condition detected communications reset	R/W									
				0: No starting conditions are generated 1: Generation of starting conditions This bit can be set to 1 and cleared to 0 by software. Hardware Clear 0 Condition: Start condition detected When arbitration fails communications reset										
b8	START	Start Condition Generation Bit		0: No duplicate starting conditions are	R/W									
				0: No starting conditions are generated 1: Generation of starting conditions This bit can be set to 1 and cleared to 0 by software. Hardware Clear 0 Condition: Start condition detected When arbitration fails communications reset										

			1: Generate duplicate starting conditions This bit can be set to 1 and cleared to 0 by software.	
b7	RESTART	Repeat Start Condition Generation Bit	Hardware Clear 0 Condition: <ul style="list-style-type: none">1) Detection of the start condition2) When arbitration fails3) Communication reset	R/W

b6	ENGC address detection	Broadcast Call Enable	0: Invalid broadcast 1: Broadcast address detection is valid	R/W
b5	Reserved	-Reserved	Reads "0", writes "0".	R/W
b4	SMBHOSTEN	Allow matching of SMBUS host address bits	0: Disable matching of SMBUS host addresses 1: Allow matching of SMBUS host addresses	R/W
b3	SMBDEFAULTEN the SMBUS default	Allows matching of address location	0: Disable matching of SMBUS default address 1: Allow matching of SMBUS default address 0: Disable SMBUS alarm response address 1: Allow SMBUS alarm response address	R/W
b2	SMBALRTEN	Allow matching of SMBUS alarm response address bits	0: I2C bus mode 1: SMBUS bus mode	R/W
b1 module	SMBUS	SMBUS/I2C bus mode selector	0: I2C function disabled 1: I2C function allowed This bit, in combination with the SWRST bit, selects an internal state reset or a communication reset	R/W
b0	PE enable	I2C function		

26.5.2 I2C Control Register 1 (I2C_CR2)

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	SMB ALRT IE	SMB HOST IE	SMB DEFA ULTI IE	GEN CALL IE	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	TMO UTIE	-	NAC KIE	-	-	ARL OIE	-	TEMP TYIE	RFUL LIE	-	STO PIE	TEN DIE	SLA DDR1 IE	SLA DDR0 IE	STA RTIE

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b24	Reserved	-	Reads "0", writes "0".	R/W
b23 Address Match Consistent	SMBALRTIE	SMBUS Alarm Response Interrupt Allowed	0: SMBUS alarm response address match consistent interrupt disable 1: SMBUS alarm response address match consistent interrupt allowed	R/W
b22 consistent interrupt	SMBHOSTIE	SMBUS host address match permissible	0: SMBUS host address match consistent 1: SMBUS host address match consistent interrupt allowed	R/W
b21 match consistent interrupt	SMBDEFAULTIE	SMBUS default address permissible	0: SMBUS default address interrupt disable 1: SMBUS host address match consistent interrupt allowed	R/W
b20	GENCALLIE	Broadcast Call Address Matching Permit	0: Broadcast Call Address Matching Consistent Interrupt Allowed 1: Broadcast Call Address Matching Consistent Interrupt disable	0: R/W
b19~b15	Reserved	-0" for reading, "0" for writing. "0", writes "0".	0: SMBUS default address match 1: SMBUS default address consistent interrupt allowed	Reads
b14	TMOUTIE	Timeout interrupt allowed	0: Timeout interrupt prohibited 1: Timeout interrupt allowed	R/W
b13	Reserved	-0" for reading, "0" for writing. "0", writes "0".	0: Receive NACK interrupt disable 1: Receive NACK interrupt permission	Reads
b12	NACKIE	NACK interrupt allowed	0: Receive NACK interrupt disable 1: Receive NACK interrupt permission	R/W
b11~b10	Reserved	-0" for reading, "0" for writing. "0", writes "0".	0: Broadcast Call Address Matching Consistent Interrupt Allowed 1: Broadcast Call Address Matching Consistent Interrupt disable	Reads
b9	ARLOIE	Arbitration Failure Interruption Allowed Arbitration failure interrupt prohibited	0: 1: Failure of arbitration interrupts permission	R/W
b8	Reserved	-0" for reading, "0" for writing. "0", writes "0".	0: Receive NACK interrupt disable 1: Receive NACK interrupt permission	Reads
b7	TEMPTYIE	Transmit Data Empty Interrupt Allow Bit data air-disconnect disable	0: 1: Sending data over-the-air is allowed	0: Transmit
b6	RFULLIE	Receive data full interrupt allow bit data full interrupt disable	0: 1: Receive data full interrupt allowed	0: Receive
b5	Reserved	-0" for reading, "0" for writing. "0", writes "0".	0: Receive NACK interrupt disable 1: Receive NACK interrupt permission	Reads
b3	TENDIE	Send one frame data end interrupt allow bit frame data end interrupt disable	0: Bus detects stop condition interrupt disable 1: Bus detects stop condition interrupt allowed	9.10V/1407 0: Send one

Allow

1: Bus detects start condition interrupt allowed

26.5.3 I2C Control Register 1 (I2C_CR3)

Reset value: 0x0000_0006

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

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b8	Reserved	-	Reads "0" and writes "0".	R/W
b7	FACKEN	RFULLF flag position bit when Interval selection R/W	0: This bit is "1" on the 9th clock rise of the SCL clock(On the 8th (SCL line is not held low on the falling edge of the clock) 1: This bit is "1" on the 8th clock rise of the SCL clock(On the 8th (The SCL line is held low on the falling edge of the clock) Release the held low level by writing the ACK bit.	0: This bit is "1" on the 8th clock rise of the SCL clock(On the 8th (SCL line is not held low on the falling edge of the clock) 1: This bit is "1" on the 8th clock rise of the SCL clock(On the 8th (The SCL line is held low on the falling edge of the clock) Release the held low level by writing the ACK bit.
b6~b3	Reserved	-0" for reading, "0" for writing. writes "0".	R/W	Reads "0", 0: Timeout
b2	HTMOUT	High level timeout detection allowed detection is disabled when the SCL line is high.	1: Timeout detection is allowed when the SCL line is high.	0: Timeout R/W
b1	LTMOUT	Low level timeout detection allowed detection is disabled when the SCL line is low.	1: Timeout detection is allowed when the SCL line is low.	0: Timeout R/W
b0	TMOUTEN	Timeout Function Allowed Bit Detect SCL level timeout function prohibited R/W	1: Detect SCL level timeout function allowable	0:

26.5.4 I2C Control Register 1 (I2C_CR4)

Reset value: 0x0030_0307

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
-	-	-	-	-	BUS WAI	-	-	-	-	-	-	-	-	-	-

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b22	Reserved	-	Reads "0", writes "0".	R/W
b21~b20	Reserved	-	Reads "1" and writes "1".	R/W
b19~b11	Reserved	-	Reads "0" and writes "0".	R/W
b10	BUSWAIT	Bus Wait Bit	0: When I2C_DRR is received full and I2C_DSR is empty, it does not hold low between the 9th clock and the 1st clock of the next transmission and continues to receive the next data. 1: When I2C_DRR receives full and I2C_DSR is empty, on the 9th clock and the next time the Hold low between the 1st clock transmitted by reading the I2C_DRR register to Release the held low level.	R/W
b9~b8	Reserved	-	Reads "1" and writes "1".	R/W
b7~b3	Reserved	-	Reads "0", writes "0".	R/W
b2~b0	Reserved	-	Reads "1" and writes "1".	R/W

26.5.5 I2C Slave Address Register 0 (I2C_SLR0)

Reset value: 0x0000_1000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ADD RMO D0	-	-	SLA DDR0 EN	-	-										SLADDR0[9:0]

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b16	Reserved	-	Reads "0", writes "0".	R/W
b15 address format	ADDRMOD0 Selected	7-bit/10-bit address format selection P/M select a seat		0: 7-bit
b14~b13	Reserved	-	Reads "0" and writes "0".	R/W 0: Slave
b12 address register 0 set value not valid	SLADDR0EN	Slave address 0 valid bit R/W	1: Slave address register 0 set value is valid	
b11~b10	Reserved	-	Reads "0" and writes "0".	R/W
b9~b8	SLADDR0[9:8]	High bit of 10-bit slave address	When the ADDRMOD0 bit is "0", this bit setting is invalid. When the ADDRMOD0 bit is "1", this bit is used as the high two bits of the 10-bit slave address.	R/W
b7~b0	SLADDR0[7:0]	Lower bit of 7-bit address/10-bit address	Sets the slave address. When the ADDRMOD0 bit is "0", SLADDR0[7:1] is the 7-bit slave ground address. the SLADDR0[0] bit is invalid. When the ADDRMOD0 bit is "1", SLADDR0[7:0] is the 10- bit slave address The lower 8 bits of the address of the	R/W

26.5.6 I2C Slave Address Register 1 (I2C_SLR1)

Reset value: 0x0000 0000

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b16	Reserved	-	Reads "0" and writes "0".	R/W
b15 address format	ADDRMOD1 Selected	7-bit/10-bit address format selection D/M select a seat	0: 7-bit 1: Select 10-bit address format	0: 7-bit
b14~b13	Reserved	-	Reads "0" and writes "0".	R/W
b12 address register 1 set value not valid	SLADDR1EN	Slave address 1 valid bit R/W	0: Slave 1: Slave address register 1 set value is valid	
b11~b10	Reserved	-	Reads "0" and writes "0". Sets the slave address.	R/W
b9~b8	SLADDR1[9:8]	High bit of 10-bit slave address	When the ADDRMOD1 bit is "0", this bit setting is invalid. When the ADDRMOD1 bit is "1", this bit is used as the high two bits of the 10-bit slave address.	R/W
b7~b0	SLADDR1[7:0]	Lower bit of 7-bit address/10-bit address	Sets the slave address. When the ADDRMOD1 bit is "0", SLADDR1[7:1] is the 7-bit slave ground address. the SLADDR1[0] bit is invalid. When the ADDRMOD1 bit is "1", SLADDR1[7:0] is a 10-bit slave address The lower 8 bits of the address of the	R/W

26.5.7 I2C SCL Level Timeout Control Register (I2C_SLTR)

Reset value: 0xFFFF FFFF

Bit Flag	Bit Name	Function	Read/Write
b31~b16	TOUTHIGH	SCL high timeout period TOUTHIGH sets the SCL high timeout period. SCL high timeout period = TOUTHIGH x I2C reference clock	R/W period

b15~b0	TOUTLOW	SCL Low Timeout Period TOUTLOW sets the SCL low timeout period. SCL low timeout period = TOUTLOW x I2C reference clock R/W period
--------	---------	--

26.5.8 I2C Status Register (I2C_SR)

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	SMB ALR TF	SMB HOS TF	SMBD EFAU LTF	GENC ALLF	-	TRA	BUSY	MSL
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	TMO UTF	-	NAC KF	-	ACK RF	ARL OF	-	TEM PTYF	RFU LLF	-	STO PF	TEN DF	SLAD DR1F	SLAD DROF	STA RTF

classifier for honorific people	markings	celebrity	functionality	fill out or in (information on a form)
b31~b21	Reserved	-	Reads "0" and writes "0".	R
b23	SMBALRTF	SMBUS Alarm Response Address Match Consistency Flag Bit	<p>0: No match to SMBUS alarm response address</p> <p>1: Host address detected</p> <p>Place a "1" condition: The received address matches 0001 100b. clears the "0" condition: R</p> <p>SMBALRTFCLR write "1"</p> <p>Stop condition</p> <p>communicatio</p> <p>n reset detected</p> <p>0: No match to SMBUS host address</p> <p>1: Match to the SMBUS</p> <p>host address Address</p> <p>match condition is as</p> <p>follows: Set the "1"</p> <p>condition:</p>	
b22	SMBHOSTF	SMBUS host address match one flags	<p>Received address matches 0001 000b</p> <p>Clear "1" condition:</p> <p>SMBHOSTFCLR writes</p> <p>"1".</p> <p>Stop condition</p> <p>communicatio</p> <p>n reset detected</p> <p>0: No match to SMBUS default address</p> <p>1: Match to SMBUS</p> <p>default address set to</p> <p>"1" condition:</p> <p>The received address matches 1100 001b.</p> <p>clears the "0" condition: R</p> <p>SMBDEFAULTFCLR write "1"</p> <p>Stop condition</p> <p>communicatio</p> <p>n reset</p>	R
b21	SMBDEFAULTF	SMBUS default address match consistency flag bit		

e			0: Broadcast call address not matched	
t			1: Match to broadcast	
e			call address set "1"	
c			condition:	
b20	GENCALLF	Broadcast Call Address Matching Consistency symbolize	When the received slave address matches the broadcast call address (All "0") Clear "0" condition: GENCALLFCLR writes "1". Stop condition detected communications reset	R
b19	Reserved	-	Reads "0" and writes "0".	R

			This bit indicates the choice of sending data or receiving data. 0: Receive data 1: Send data																
			This bit can be set to 1 and cleared to 0 by software. Hardware "1" condition Start condition detected																
b18	TRA	Transmit/Receive Select Bit	In host mode, the R/W bit sent is 0 Slave mode, address match and received R/W bit is 1 Hardware clear "0" condition Stop condition detected In host mode, the R/W bit sent is 1 Slave mode, address match and received R/W bit is 0 communications reset	R/W															
b17	BUSY	Bus Busy Flag Bit	0: Idle state, no communication on the bus 1: Possessed state, bus is communicating Place a "1" condition: Start condition detected on the bus Clear "0" condition: Bus stop condition detected communications reset	R															
b16	MSL	master-slave selector bit	This bit indicates whether the host or the slave. 0: Slave mode <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">I: Host Mode - MSL Indicates the mode with the TRA bit.</td> <td style="padding: 2px;">TRA</td> <td style="padding: 2px;">I2C operation of operation of the I2C by combining mode</td> </tr> <tr> <td style="padding: 2px;">0</td> <td style="padding: 2px;">0</td> <td style="padding: 2px;">slave receive mode</td> </tr> <tr> <td style="padding: 2px;">0</td> <td style="padding: 2px;">1</td> <td style="padding: 2px;">slave transmitter mode</td> </tr> <tr> <td style="padding: 2px;">1</td> <td style="padding: 2px;">0</td> <td style="padding: 2px;">Host Receive Mode</td> </tr> <tr> <td style="padding: 2px;">This bit can be set to 1 and cleared to 0 by software. Hardware "1" condition</td> <td style="padding: 2px;"></td> <td style="padding: 2px;">Host Transmit Mode</td> </tr> </table> Start condition is detected in the state where the START bit is 1 Hardware clear "0" condition 1) Stop condition detected 2) Failure of arbitration 3) Communication reset	I: Host Mode - MSL Indicates the mode with the TRA bit.	TRA	I2C operation of operation of the I2C by combining mode	0	0	slave receive mode	0	1	slave transmitter mode	1	0	Host Receive Mode	This bit can be set to 1 and cleared to 0 by software. Hardware "1" condition		Host Transmit Mode	R/W
I: Host Mode - MSL Indicates the mode with the TRA bit.	TRA	I2C operation of operation of the I2C by combining mode																	
0	0	slave receive mode																	
0	1	slave transmitter mode																	
1	0	Host Receive Mode																	
This bit can be set to 1 and cleared to 0 by software. Hardware "1" condition		Host Transmit Mode																	
b15	Reserved	-	Reads "0", writes "0".	R															
14	TMOUTF	timeout flag	0: SCL level timeout not detected 1: SCL level timeout Place a "1" condition: SCL not flipped for the period set by I2C_SLTR Clear "1" condition: TMOUTFCLR write "1" communications reset	R															

b13	Reserved	-	Reads "0", writes "0".	R
b12	NACKF	NACK flag bit	0: NACK not received 1: NACK received Place a "1" condition:	R

			In transmit mode, NACK clear "0" condition is received: NACKFCLR write "1". communications reset	
b11	Reserved	-	Reads "0", writes "0".	R
b10	ACKRF	receive answer bit (computing)	0: Receive answer bit "0" (receive ACK) 1: Receive answer bit "1" (receive NACK) Place a "1" condition: In transmit mode, a NACK is received	R
b9	ARLOF	Arbitration Failure Flag Bit	Clear "0" condition: In transmit mode, an ACK is received communications reset 0: No arbitration failure occurred 1: Failure of arbitration Place a "1" condition: Arbitration fails	R
b8	Reserved	-	Reads "0", writes "0".	R
b7	TEMPTYF	Send Data Empty Flag Bit	0: I2C_DTR register full 1: I2C_DTR register empty Place a "1" condition: I2C_DTR data transfer to I2C_DSR TRA position 1 Clear "0" condition: Write I2C_DTR TRA bit clear 0 communications reset	R
b6	RFULLF	Receive Data Full Flag Bit	0: I2C_DRD register empty 1: I2C_DRD register full Place a "1" condition: The received data is transferred from I2C_DSR to the I2C_DRD clear "0" condition: Read I2C_DRD RFULLFCLR write "1" communications reset	R
b5	Reserved	-	Reads "0" and writes "0".	R
b4	STOPF	Stop condition flag bit	0: Stop condition not detected by the bus 1: Bus detects a stop condition Place a "1" condition: Stop condition detected Clear "0" condition: STOPFCLR write "1"	R

			communications reset	
b3	TENDF	Send Data End Flag Bit	0: I2C_DSR register transmit in progress 1: End of I2C_DSR register transmission	R
			Place a "1" condition:	

			<p>TEMPTYF=1 condition, the 9th rising edge of SCL this position "1" Clear "0" condition: Stop condition detected Write I2C_DTR TENDFCLR write "1" communications reset</p> <hr/>
b2	SLADDR1F	Slave Address Register 1 Match Consistency Flag	<p>0: Slave address register 1 consistent address not detected 1: Slave address register 1 is detected with a consistent address "1" condition: When the I2C_SLR1.ADDRMOD1 bit is "0", when the received slave address and I2C_SLR1. SLADDR1[7:1] match. When the I2C_SLR1.ADDRMOD1 bit is "1", receive 10-bit slave ground The first byte address of the address matches 11110b+I2C_SLR1. RSLADDR1[9:8] matches and the second byte address matches the I2C_SLR1. SLADDR1[7:0] matches are consistent. Clear "0" condition: Stop condition detected</p> <hr/>
b1	SLADDR0F	Slave Address Register 0 Match Consistency Flag	<p>0: Slave address register 0 consistent address not detected 1: Slave address register 0 consistent address "1" condition is detected: When the I2C_SLR0.ADDRMOD0 bit is "0", when the received slave address and I2C_SLR0. SLADDR0[7:1] match. When the I2C_SLR0.ADDRMOD0 bit is "1", receive 10-bit slave ground The first byte address of the address matches 11110b+I2C_SLR0. RSLADDR0[9:8] matches and the second byte address matches the I2C_SLR0. SLADDR0[7:0] matches are consistent. Clear "0" condition: Stop condition detected SLADDR0FCLR write "1" communications reset</p> <hr/>

26.5.9 I2C Status Clear Register (I2C_CLR)

Reset value: 0x0000_0000

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 UTFC LR	-	NAC KFCLR	-	-	ARLO FCLR	-	TEM PTYF CLR	RFU LLFC LR	-	STO PFC LR	TEN DFC LR	SLA FCLR	SLAD DR0F CLR	STA RTFC LR

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b24	Reserved	-	Write "0".	W
b23	SMBALRTFCLR	SMBUS Alarm Response Address Match Flag Clear to clear the SMBALRTF flag bit.	Write "1" to clear the SMBALRTF flag bit. zero position	Write "1"
b22	SMBHOSTFCLR	SMBUS Host Address Match Consistency Flag Clear Bit	Write "1" to clear the SMBHOSTF flag bit.	W
b21	SMBDEFAULTFCLR	SMBUS Default Address Match Consistency Flag Clear Bit	Write "1" to clear the SMBDEFAULTF flag bit.	W
b20	GENCALLFCLR	Broadcast Call Address Match Consistency Flag Bit	Write "1" to clear the GENCALLF flag bit.	W
b19~b15	Reserved	-	Write "0".	W
b14	TMOUTFCLR	timeout flag	Write "1" to clear the TMOUTF flag bit.	W
b13	Reserved	-	Write "0".	W
b12	NACKFCLR	NACK flag bit	Write "1" to clear the NACKF flag bit.	W
b11~b10	Reserved	-	Write "0".	W
b9	ARLOFCLR	Arbitration Failure Flag Bit	Write "1" to clear the ARLOF flag bit.	W
b8	Reserved	-	Write "0".	W
b7	TEMPTYFCLR	Send Data Empty Flag Bit	Write "1" to clear the TEMPTYF flag bit.	W
b6	RFULLFCLR	Receive Data Full Flag Bit	Write "1" to clear the RFULLF flag bit.	W
b5	Reserved	-	Write "0".	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	Slave Address Register 1 Match Consistency Flag Clear Bit	Write "1" to clear the SLADDR1F flag bit.	W
b1	SLADDR0FCLR	Slave Address Register 1 Match Consistency Flag Clear Bit	Write "1" to clear the SLADDR0F flag bit.	W
b0	STARTFCLR	Start condition/restart condition flag clear bit	Write "1" to clear the STARTF flag bit.	W

26.5.10 I2C Data Transmit Register (I2C_DTR)

Reset value: 0xFF

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

If the I2C_DSR register is empty, the transmit data written to the I2C_DTR register is transferred to the I2C_DSR register, and data transmission to the SDA is started in transmit mode.

The I2C_DSR register and I2C_DTR register are double-buffered structures that allow continuous data sending if the I2C_DTR register data is pre-written during the I2C_DSR register data sending process.

The I2C_DTR register is readable and writable. Please write the I2C_DTR register only once when a transmit data air-disconnect request occurs.

26.5.11 I2C Data Receive Register (I2C_DRR)

Reset value: 0x00

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

If 1 frame of data is received, the received data can be dumped from the shift register (I2C_DSR) to the I2C_DRR register, and then you can move to the next data receiving state.

The I2C_DSR register and I2C_DRR register are double-buffered structures that allow continuous data reception if the I2C_DRR register is read during I2C_DSR register data reception.

Write to the I2C_DRR register is disabled. Please read the I2C_DRR register only once when the receive data full interrupt request occurs.

In the state of I2C_SR.RFULLF flag bit is "1", if the data in I2C_DRR register is not read and the next data is received immediately, the SCL clock will be automatically held low one SCL clock before the next RFULLF flag bit changes to "1". If the I2C_DRR register is not read and the next data is received immediately, the SCL clock will be automatically held low before the next RFULLF flag bit "1".

26.5.12 I2C Data Shift Register (I2C_DSR)

b7	b6	b5	b4	b3	b2	b1	b0
DSR							

The I2C_DSR register is used as a shift register for transmitting and receiving data. The I2C_DSR

register is not readable or writable.

During data transmission, transmit data from I2C_DTR register to I2C_DSR register and send data from SDA pin. During data reception, once the 1-frame data reception is finished, transfer the data from I2C_DSR register to I2C_DRR register.

26.5.13 I2C Clock Control Register (I2C_CCR)

Reset value: 0x0000_1F1F

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	FREQ[2:0]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-		SHIGHW[4:0]			-	-	-		SLOWW[4:0]				

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b23	Reserved	-	Reads "0", writes "0".	R/W
b22~b19	Reserved	-	Reads "1" and writes "1".	R/W
b18-b16	FREQ[2:0]	I2C Reference Clock Frequency Setting	0 0 0: I2C reference clock frequency = PCLK3/1 0 0 1: I2C reference clock frequency = PCLK3/2 0 1 0: I2C reference clock frequency = PCLK3/4 0 1 1: I2C reference clock frequency = PCLK3/8 1 0 0: I2C reference clock frequency = PCLK3/16 1 0 1: I2C reference clock frequency = PCLK3/32 1 1 0: I2C reference clock frequency = PCLK3/64 1 1 1: I2C reference clock frequency = PCLK3/128	R/W
b15~b13	Reserved	-	Reads "1" and writes "1".	R/W
b12~b8	SHIGHW[4:0]	Set SCL high level width bit	Setting the high level width of the SCL clock	R/W
b7~b5	Reserved	-	Reads "1" and writes "1".	R/W
b4~b0	SLOWW[4:0]	Set SCL low level width bit	Setting the low level width of the SCL clock	R/W

SHIGHW bit (sets SCL high level width bit)

In master mode, SHIGHW is used to set the high level width of the SCL clock. In slave mode, the setting is invalid.

SLOWW bit (Sets SCL Low Width bit)

SLOWW is used to set the low level width of the SCL clock. In slave mode, the setting is greater than the data ready time. Data ready time (tSU:DAT) 250ns (~100kbps: standard mode) 100ns (~400kbps: fast mode)

Baud rate:

$$\text{DNFE}=0, \text{FREQ}=000$$

$$\text{Baud rate} = 1/\{[(\text{SHIGHW}+3)+(\text{SLOWW}+3)]/\Phi_{\text{I2C}}+\text{SCL rise time}+\text{SCL fall time}\}$$

$$\text{DNFE}=1, \text{FREQ}=000$$

$$\text{Baud rate} = 1/\{[(\text{SHIGHW}+3+\text{filter capacity}) + (\text{SLOWW}+3+\text{filter capacity})]/\Phi_{\text{I2C}}+\text{SCL Rise time} + \text{SCL downtime}\}$$

$$\text{DNFE}=0, \text{FREQ}!=000$$

Baud rate = $1/\{[(\text{SHIGHW}+2)+(\text{SLOWW}+2)]/\Phi\text{I2C}+\text{SCL rise time}+\text{SCL fall time}\}$

DNFE=1,FREQ!=000

Baud rate = $1/\{[(\text{SHIGHW}+2+\text{filter capacity}) + (\text{SLOWW}+2+\text{filter capacity})]/\Phi\text{I2C}+\text{SCL Rise time}+\text{SCL downtime}\}$

Attention:

- The rise time [tr] and fall time [tf] of the SCL line depend on the total bus capacitance [Cb] and pull-up resistor [Rp], please refer to NXP's I2C bus datasheet for details.

26.5.14 I2C Filter Control Register (I2C_FLTR)

Reset value: 0x0000_0010

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

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b6	Reserved	-	Reads "0" and writes "0".	R/W
b5 Filtering Disabled	ANFEN	Analog Filtering Allowed Bit	0: Analog	R/W
b4	DNFEN	Digital Filtering Function Allowed Bit	1: Analog filter function allows Digital filter function disabled 0: Digital filtering function allowed	0: R/W
b3~b2	Reserved	-0" for reading, "0" for writing. writes "0".	Reads "0", R/W	
b1-b0	DNF [1:0]	Digital Filter Filter Capability Selection 00: Filtering capacity 1 I2C reference clock cycle 10: Filtering capacity 3 I2C reference clock cycles 11: Filtering capacity 4 I2C reference clock cycles	01: Filter R/W	

27 Serial Peripheral Interface (SPI)

27.1 summary

This product is equipped with a 4-channel serial peripheral interface SPI, which supports high-speed full-duplex serial synchronous transmission and convenient data exchange with peripheral devices. Users can set up 3-wire/4-wire, master/slave and baud rate range as needed.

SPI Key Features:

Table 27-1 SPI Feature Highlights

pivot	descriptive
channel number	4 channels
Serial communication function	<ul style="list-style-type: none">Supports 4-wire SPI mode and 3-wire clock synchronization mode of operationSupports both full-duplex and transmission-only communication methodsAdjustable communication clock SCK polarity and phase
data format	<ul style="list-style-type: none">Selectable data shift order: MSB start/LSB startSelectable data width: 4/5/6/7/8/9/10/11/12/13/14/15/16/20/24/32 bitsTransmit or receive up to 4 frames of 32-bit width data in a single pass
baud	<ul style="list-style-type: none">The baud rate can be adjusted by the built-in special baud rate generator in host mode, and the baud rate range is 2 divisions of PCLK. ~256 crossover frequencyThe maximum baud rate allowed in slave mode is 6 divisions of PCLK
data buffer	<ul style="list-style-type: none">With 16-byte data bufferSupports double buffering
Error monitoring	<ul style="list-style-type: none">Mode Failure Error MonitoringData overload error monitoringData underload error monitoringParity error monitoring
Chip select signal control	<ul style="list-style-type: none">Four chip select signal lines per channelThe relative timing relationship between the chip select signal and the communication clock can be adjusted.Adjustable chip select signal invalidation time between two consecutive communicationsAdjustable polarity
Transmission in host mode transmission control	<ul style="list-style-type: none">Initiate transmission by writing data to the data registersCommunication auto-suspend function

Interrupt/AOS source	<ul style="list-style-type: none">- Receive data area is full- Send data area is empty- SPI errors (mode/overload/underload/parity)- SPI Idle- Transmission complete
Low Power Control	Module stop can be set
Other Functions	<ul style="list-style-type: none">- SPI initialization function

Attention:

- When using the communication auto hang-up function in the main receive mode, overload error will not occur because the communication clock is stopped. For details, refer to [Overload Error]

27.2 SPI System Block Diagram

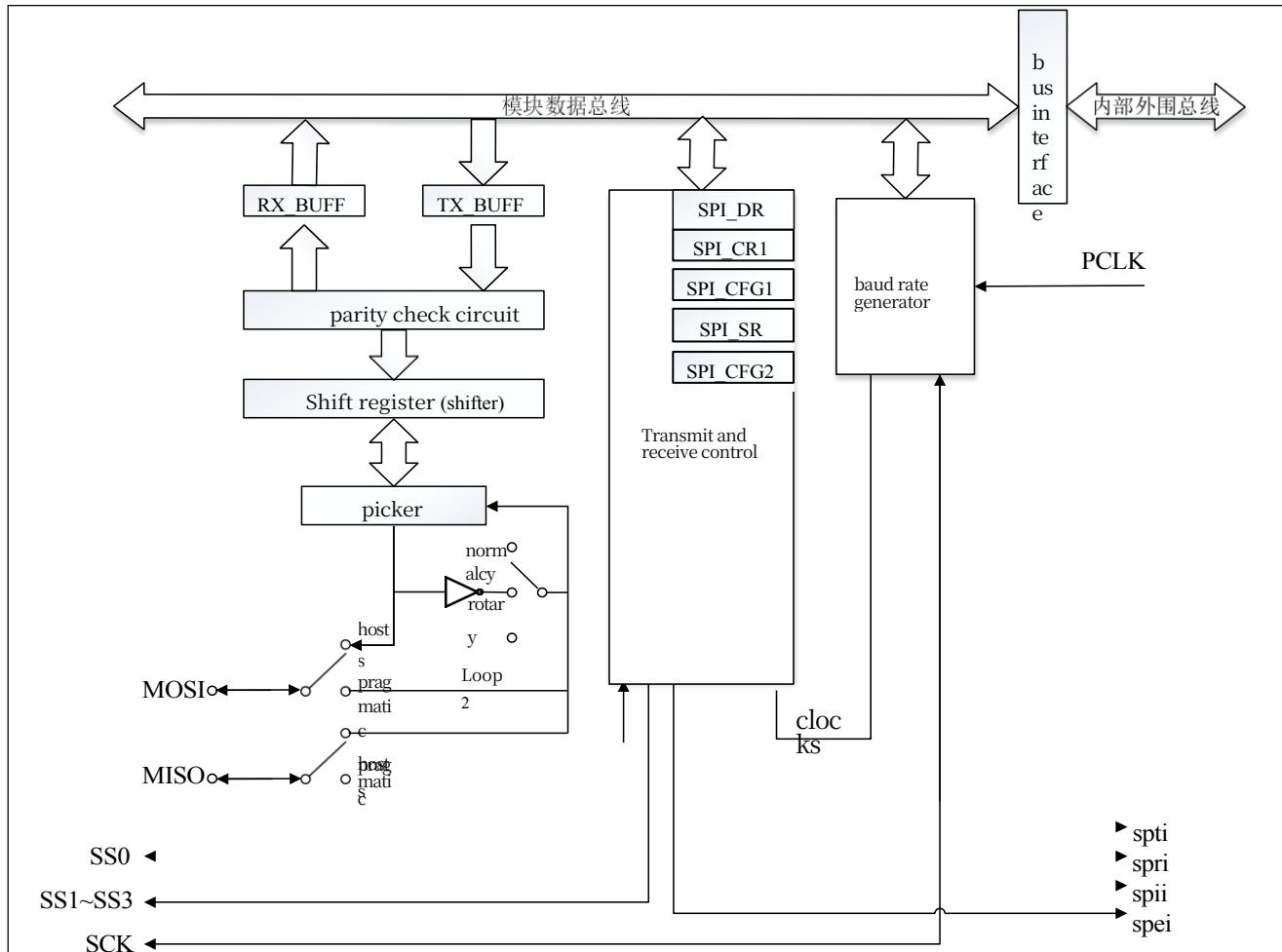


Figure 27-1 System Block Diagram

27.3 Pin

**Descrip
tion**

Table 27-2 Pin Descriptions

pin name	port direction	functionality
SCK	Input/Output	Communication clock pin
MOSI	Input/Output	Host Data Transfer Pins
MISO	Input/Output	Slave Data Transfer Pins
SS0	Input/Output	Slave select input/output pins
SS1	exports	Slave select output pin
SS2	exports	Slave select output pin

27.4 SPI Action System Description

27.4.1 Host Mode Pin Status

The status of each pin when SPI is operating in host mode is shown in the table below.

Table 27-3 SPI Pin Status Descriptions in Host Mode

paradigm		pin name	pin state (PFS.ODS=0)	pin state (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	importation	importation
Clock synchronized operation (SPIMDS=1)	Host Mode (MSTR=1)	SCK	CMOS output	OD Output
		SS0~SS3 (not enable) (used)	Hi-Z (available as generic) I/O)	Hi-Z (available as generic) I/O)
		MOSI	CMOS output	OD Output
		MISO	importation	importation

Attention:

- When a valid level is input to SS0, SPI relinquishes control of the bus and the pin state is Hi-Z.

27.4.2 Slave mode pin status

The status of each pin when SPI is operating in slave mode is shown in the table below.

Table 27-4 SPI Pin Status Description in Slave Mode

paradigm		pin name	Pin Status (PFS.ODS=0)	pin state (PFS.ODS=1)
SPI Action (SPIMDS=0)	slave mode (MSTR=0, MODFE=0)	SCK	importation	importation
		SS0	importation	importation
		SS1~SS3 (do not make) (used)	Hi-Z (available as general purpose I/O)	Hi-Z (available as generic) I/O)
		MOSI	importation	importation
		MISO (note 2)	CMOS output/Hi-Z	OD Output/Hi-Z
Clock synchronized operation (SPIMDS=1)	slave mode (MSTR=0)	SCK	importation	importation
		SS0~SS3 (not enable) (used)	Hi-Z (available as general purpose I/O)	Hi-Z (available as generic) I/O)
		MOSI	importation	importation
		MISO	CMOS output	OD Output

27.4.3 SPI System Connection Example

Host Mode

In the SPI system structure of host-multi-slave mode, the host drives SCK, MOSI and SS0~SS3. In SPI slave devices 0~3, when the SS input of a slave is valid level, the slave device drives MISO.

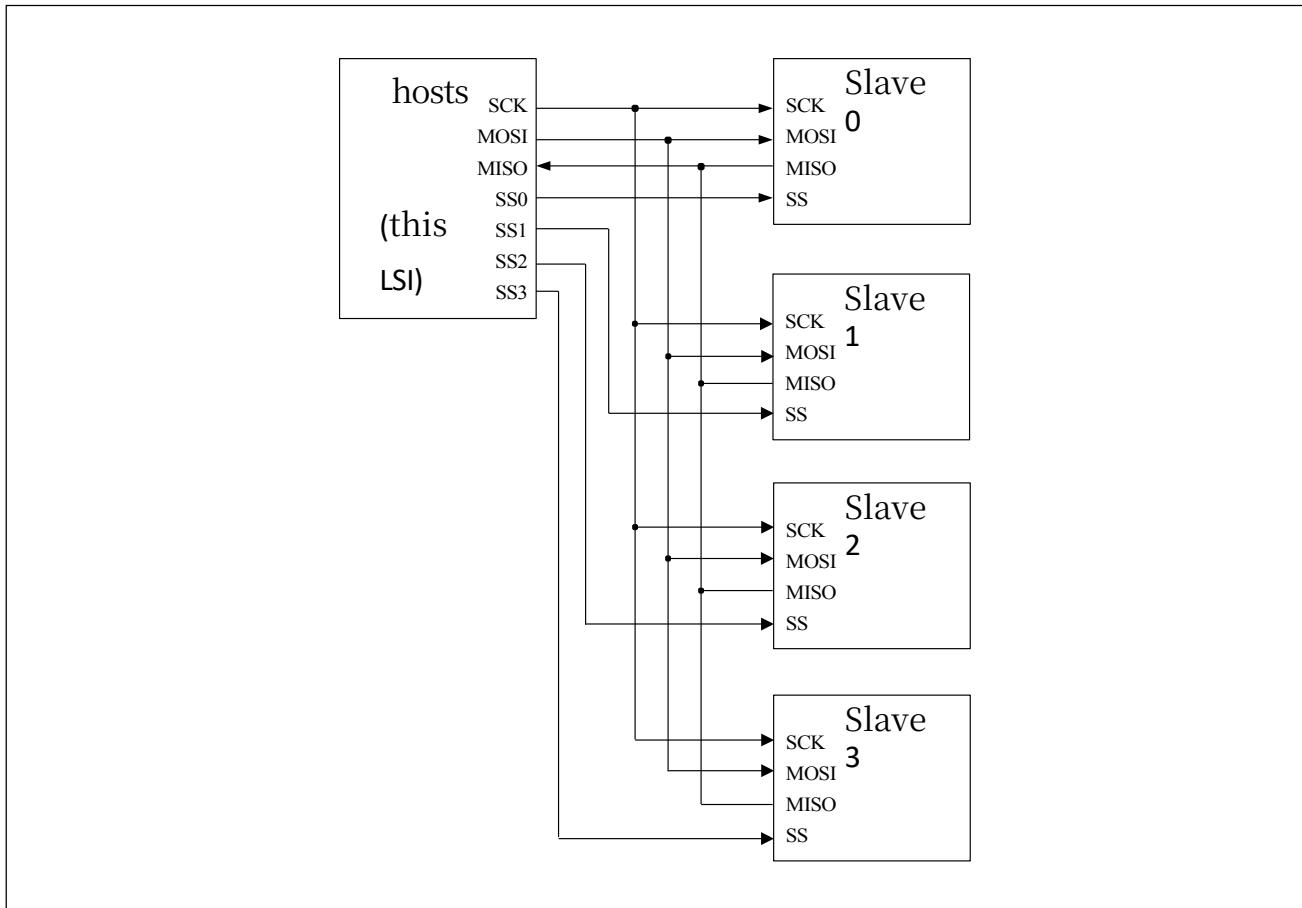


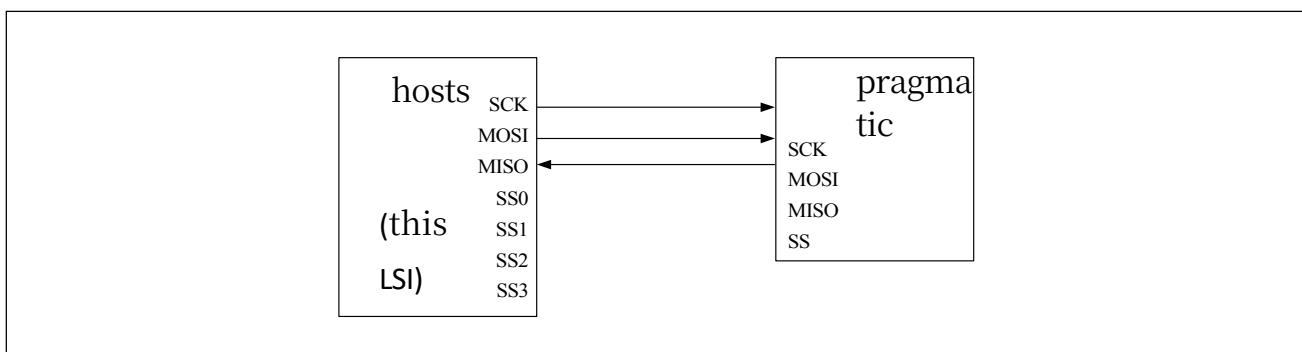
Figure 27-2 Host Mode Structure

Clock

Synchronize

Operation

In an SPI system architecture used as a clock synchronization operation, the host device drives SCK and MOSI, and the slave device drives MISO. The SS pin is not used.



27.5 Data communication description

27.5.1 baud

In host mode, the SPI clock is provided by the internal baud rate generator; in slave mode, the clock is input from the SCK pin.

The baud rate depends on the SPI_CFG2.MBR[2:0] MBR[2:0] bits. It is calculated as shown in the following formula, where N is the setting of the MBR[2:0] bits, and the range is 0 to 7.

$$\text{Baud rate} = \frac{\text{fpck}}{2^{N+1}}$$

Table 27-5 Partial Setpoint Lower Speed

MBR[2:0] bits of the setpoint	crossover ratio	baud			
		PCLK=5MHz	PCLK=10MHz	PCLK=20MHz	PCLK=40MHz
0	2	2.50 Mbps	5.00Mbps	10.0Mbps	20.0 Mbps
1	4	1.25Mbps	2.50 Mbps	5.00Mbps	10.0Mbps
2	8	625kbps	1.25Mbps	2.50 Mbps	5.00Mbps
3	16	313kbps	625kbps	1.25Mbps	2.50 Mbps
4	32	156kbps	313kbps	625kbps	1.25Mbps
5	64	78kbps	156kbps	313kbps	625kbps
6	128	39kbps	78kbps	156kbps	313kbps
7	256	20kbps	39kbps	78kbps	156kbps

27.5.2 data format

The data format of SPI depends on the set value of the parity permission bit PAE in the SPI command register SPI_CFG2 and SPI control register SPI_CR1. SPI treats the data in the data register SPI_DR with a certain length starting from the LSB bit (the data length is set by the DSIZE[3:0] bits in the register SPI_CFG2) as a transfer object and processes it independently of the MSB/LSB shift order. The data in SPI_DR from the LSB bit of a certain length (the data length is set by the DSIZE[3:0] bits in SPI_CFG2) is processed as a 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, and SPI_CR1.PAE determines the last bit of the data, which is used as the parity bit when PAE is 1, and the lowest bit of the data when PAE is 0. PAE determines the last bit of the data.

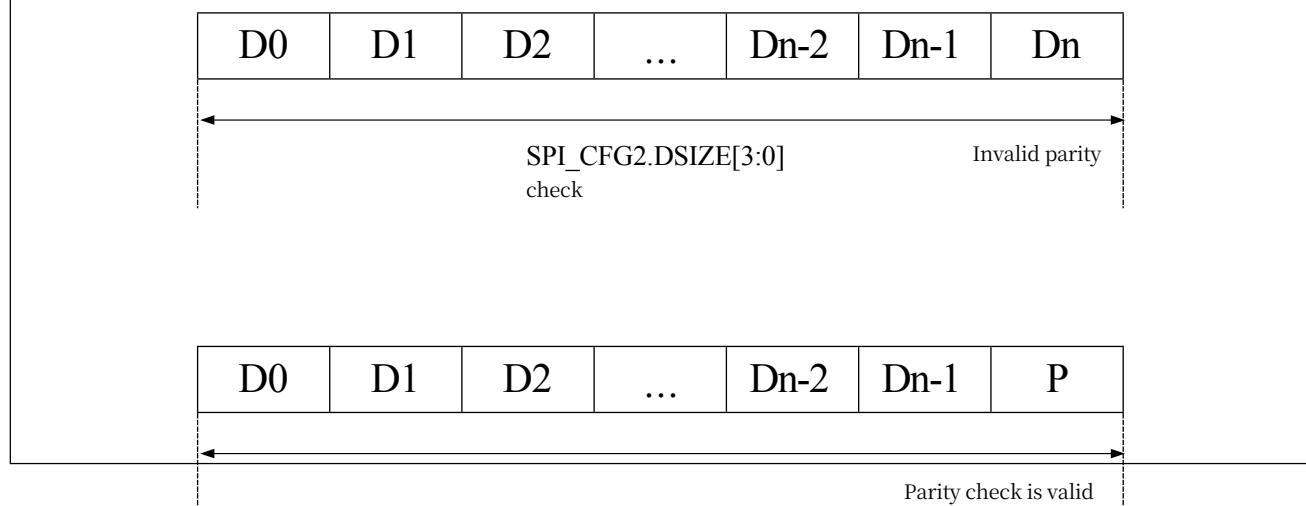


Figure 27-4 Data Format

When SPI data is sent, the transmitted data first enters the transmit buffer (TX_BUFF) then the data in TX_BUFF is copied to the shift register (shifter) and the shifter sends out the data sequentially; when SPI data is received, the data is shifted in from the shifter sequentially, and the data in the shifter is copied to the receive buffer (RX_BUFF) after the shift is completed. BUFF)

Data transfer is categorized into four cases according to the settings of the shift order control bit SPI_CFG2.LSBF and the parity control bit SPI_CR1.PAE:

- 1) MSB first. Parity invalid.

When sending, data d31~d0 are copied from TX_BUFF to shifter in order, and shifted out from the highest bit of shifter in the order of d31~d0;
When receiving, data d31~d0 are shifted in from the lowest bit of the shifter,

and wait until all the data are shifted in before copying the data to RX_BUFF.

- 2) LSB passes first, parity not valid

When sending, data d31~d0 are copied from TX_BUFF to shifter in the order of d0~d31, shifted out from the highest bit of shifter in the order of d0~d31;

When receiving, data d0~d31 are shifted in from the lowest bit of the shifter, and when all the data are shifted in, they are copied from the shifter to RX_BUFF in the order of d31~d0.

3) MSB first, when parity is valid

When sending, the value of parity bit P is calculated according to the value of d31~d1, and then P is used instead of d0 copied to the shifter in the order of d31~d1, P, and shifted out from the highest bit of the shifter in the order of d31~P;

When receiving, data d31~P is shifted in from the lowest bit of the shifter, and parity checking is performed while the data is copied to the shifter. Finally, the data is copied to RX_BUFF.

4) LSB passes first, when parity is valid

When sending, the value of parity bit P is calculated from the value of d30~d0, then P is used instead of d31, and copied from TX_BUFF to shifter in the order of d0~P, and shifted out from the highest bit of the shifter in the order of d0~P.

When receiving, the data d0~P is shifted in from the lowest bit of the shifter, and parity checking is performed when the data is copied to the shifter.

When the data is copied to the shifter, parity check is performed.

Data d0~P are rearranged during copying and copied to RX_BUFF in the order of P~d0.

27.5.3 transmission format

1) CPHA=0 case

When the SPI_CFG2.CPHA bit is "0", SPI samples data on the odd edge of SCK and updates data on the even edge. Figure 27-5 shows the SPI transmission timing when CPHA=0. When the input level of SS_I signal becomes active, MOSI/MISO starts to update the transmitted data. The first data sample is taken at the first SCK signal edge after the SS_I signal becomes active, and after that, the data is sampled every SCK cycle. The data on the MOSI/MISO signal is updated for 1/2 SCK cycle after each sample, and the setting of the CPOL bit does not affect the operation timing of the SCK signal, but only the polarity of the signal.

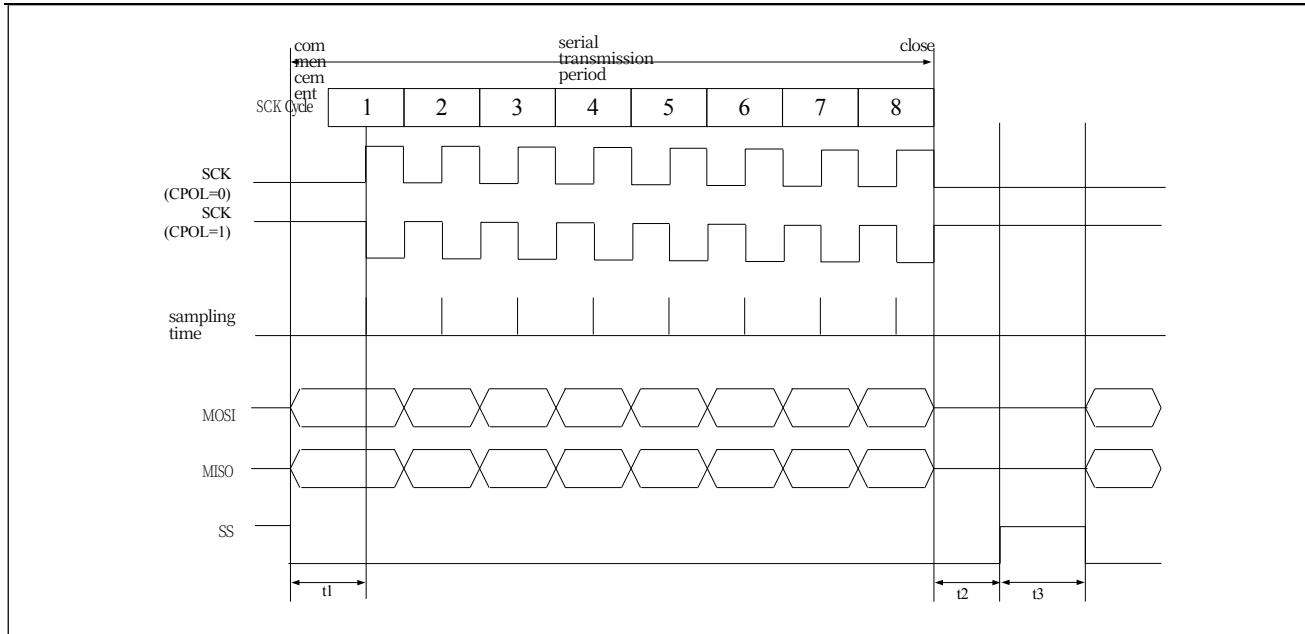


Figure 27-5 Data Transfer Format Diagram (CPHA=0)

In the above figure, t1 represents the interval from the time when the SS_i signal is valid to the SCK oscillation (SCK delay time) t2 represents the interval from the time when the SCK oscillation stops to the time when the SS signal becomes invalid (SS invalid delay time) and t3 represents the interval from the end of the serial transmission to the next

The minimum wait time for the start of the next transfer (next access delay) t_1 , t_2 , and t_3 are controlled by the host device on the SPI system. Refer to Section 27.6.2 for details.

2) CPHA=1 case

When the SPI_CFG2.CPHA bit is "1", SPI performs data update on the odd edge of SCK and data sampling on the even edge. Figure 27-6 shows the SPI transmission timing diagram when CPHA=1. MOSI/MISO starts transmitting data update at the first SCK signal edge after the SS signal becomes active. After that, the data is updated every SCK cycle. The data is sampled at $1/2$ SCK cycle after each update. The setting of SPI_CFG2.CPOL bit does not affect the timing of the SCK signal but only the polarity of the signal.

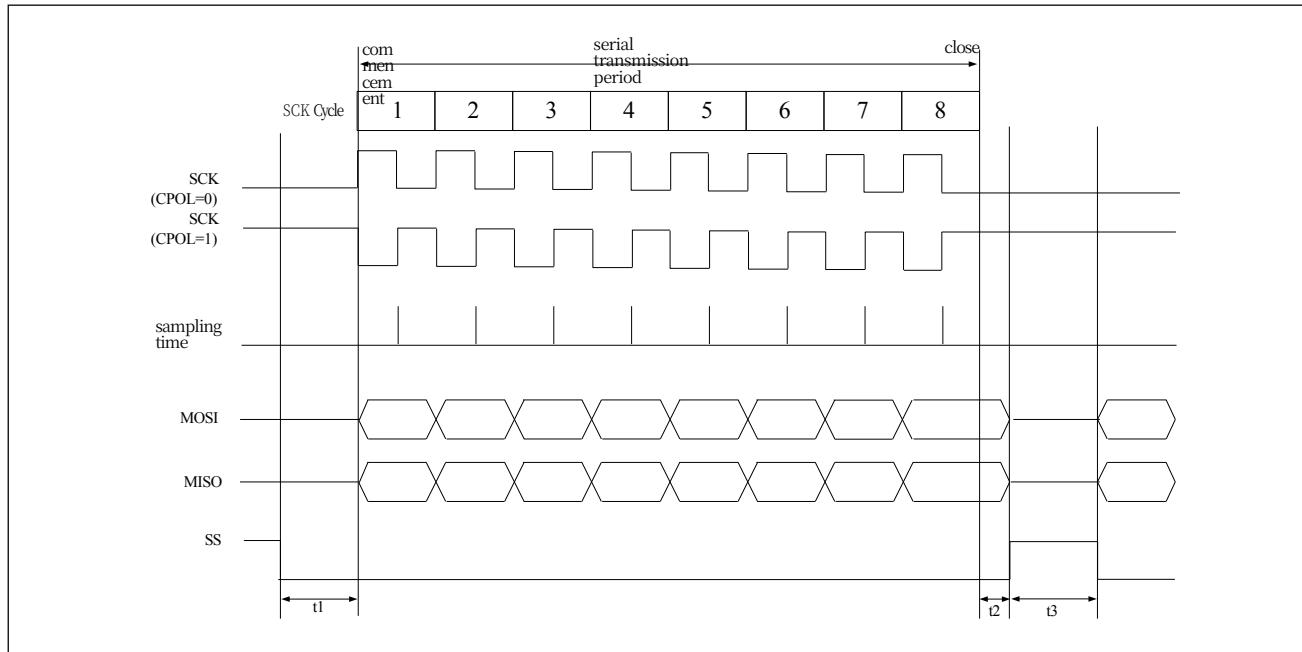


Figure 27-6 Data Transfer Format (CPHA=1)

The t_1 , t_2 , t_3 and CPHA bits are the same as if they were zero.

27.5.4 communication method

This SPI has two communication modes, full-duplex synchronous serial communication and transmit-only serial communication, which can be selected by the TXMDS bit of the SPI control register(SPI_CR1).

1) Full duplex synchronous serial communication

When the SPI_CR1.TXMDS bit is "0", the SPI is running in full duplex synchronous serial communication mode. As shown in Figure 27-7, SPI_CFG1.
FTHLV[1:0] bit is "00b" and SPI_CFG2.
FTHLV[1:0] bit is "00b" and SPI_CFG2.

And

SPI_CFG2.CPOL bit is "0", SPI performs 8-bit serial transmission.

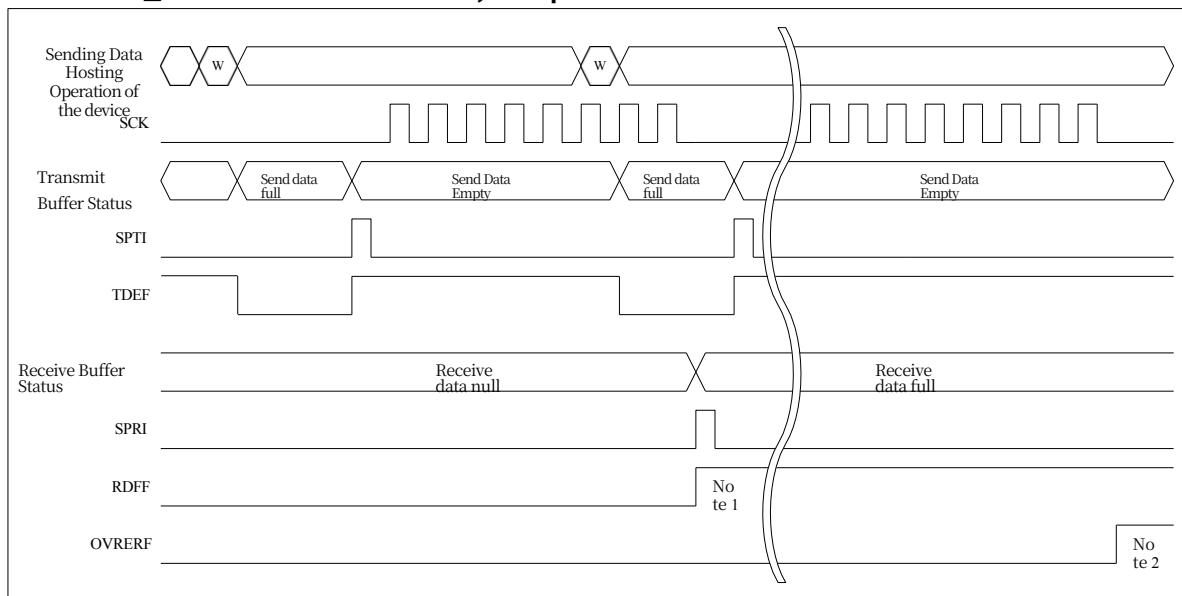


Figure 27-7 Full-Duplex Synchronous Serial Communication

Notes:

1. At the end of this serial transmission, if the receive data buffer register is empty, SPI will copy the received data from the shift register into the receive data buffer register, the receive data buffer register full flag bit is set to 1 (RDFF) and a receive data full interrupt request (SPRI) is generated.
2. At the end of this serial transmission, if the last received data is still held in the receive data buffer register and has not been read by the system, SPI will set the data overload flag to 1, this data reception is invalid, and the data in the receive shift register will be discarded.

2) Send-only communication method

When the SPI_CR1. bit is "1" When SPI_CR1.TXMDS bit is "1", SPI operates in transmit-only communication mode. As shown in the figure 27-8,

SPI_CFG1.FTHLV[1:0] bit is "00b", SPI_CFG2. FTHLV[1:0] bit

is "00b", SPI_CFG2.CPHA bit is "1" and

SPI_CFG2.CPOL bit is "0", SPI performs 8-bit serial transmission.

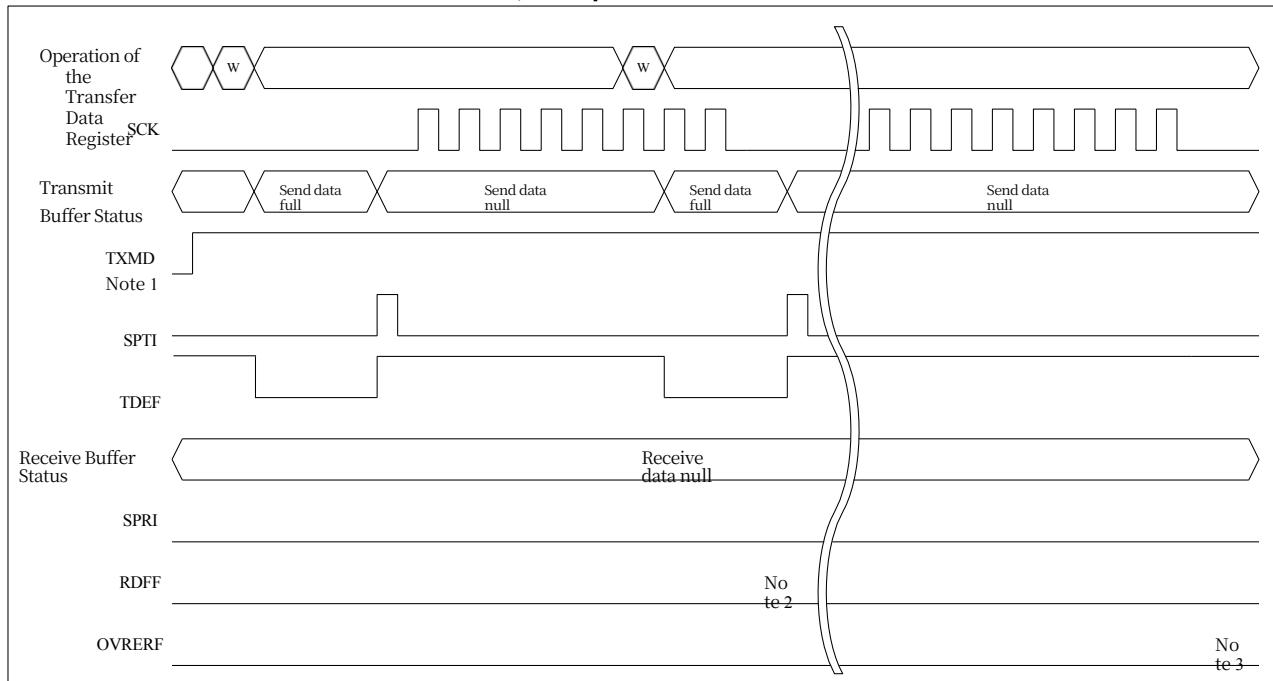


Figure 27-8 Transmit-Only Communication

Notes:

1. Before setting up to enter the transmit-only communication mode, make sure that there is no unread data in the receive buffer register (i.e., that RDFF is 0) and no data overload error (i.e., OVRERF is 0)
2. In the transmit-only communication mode, when this serial transmission is finished, no data will be accepted even if the receive data buffer register is empty, and RDFF will always remain 0.
3. In the transmit-only communication mode, no data overload error occurs because the receive data buffer register is always empty, and the OVRERF flag bit always remains 0.

27.6 Running Instructions

27.6.1 Summary of operating modes

This SPI supports 4-wire SPI mode and 3-wire clock synchronization operation mode. In each operation mode, it can communicate serially as a master or a slave. The SPI mode is set by setting the MSTR and SPIMDS bits in the SPI control register SPI_CR1. The relationship between the SPI modes and the SPI_CR1 register settings, and the summary of each mode are shown in Table 27-6.

Table 27-6 SPI Mode and Register Setting Relationships

paradigm	hosts (SPI running)	pragmatic (SPI running)	hosts (Clocks running synchronized)	pragmatic (Clocks running synchronized)
MSTR bit setting	1	0	1	0
SPIMDS bit setting	0	0	1	1
SCK signal	exports	importation	exports	importation
MOSI signal	exports	importation	exports	importation
MISO signal	importation	Output/Hi-Z	importation	exports
SS0 signal	exports	importation	Hi-Z (not used)	Hi-Z (not used)
SS1~SS3 signals	exports	Hi-Z	Hi-Z (not used)	Hi-Z (not used)
SS polarity change function	there are	there are	-	-
Maximum transmission rate	~PCLK/2	~PCLK/6	~PCLK/2	~PCLK/6
clock source	internal baud rate generator	SCK input	internal baud rate generator	SCK input
clock polarity	2 types	2 types	2 types	2 types
phase of the clock	2 types	2 types	2 types	1 species (CHPA=1)
starting transport 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	there are	not have	there are	not have
SS Invalid Delay Control	there are	not have	there are	not have
Next Access Delay Control	there are	not have	there are	not have
Transmission initiation method	Disconnect through the transmit buffer air Request, Write Send	When SS input is valid or SCK edge of the clock	Disconnect through the transmit buffer air Request, Write Send	SCK oscillation

	Buffer		Buffer	
Send Buffer Empty Detection	there are	there are	there are	there are
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)			
Underload error detection	there are	not have	there are	not have

Notes:

1. When the SPI_CR1.TXMDS bit is "1", receive buffer full detection, overload error detection and parity error detection are not performed.
2. When the SPI_CR1.PAE bit is "0", parity error detection is not performed.

27.6.2 Host action in SPI operation mode

1) Explanation of actions when acting as a host

When the SPI data transmit buffer register (TX_BUFF) is empty (the TDEF flag bit in the status register SPI_SR is 1) the **SPI** will update the SPI_DR data into the TX_BUFF after writing the data with the frame length set in the FTHLV[1:0] bits of the format control register SPI_CFG1 to the SPI data register (SPI_DR). If the shift register (**shifter**) is empty at this time, **SPI** copies the data of TX_BUFF to the shift register to start serial transmission.

SPI will change the shifter state to full when the data sent is copied to the shifter; it will change to empty when the serial transmission is finished. the state of the shifter cannot be read.

The serial transfer ends when the SPI sends the SCK edge required for the last sample timing, the end time is independent of the SPI_CFG2.CPHA bit. If the receive buffer (RX_BUFF) is empty, the SPI copies the data in the **shifter** to RX_BUFF after the end of the serial transmission, which can be read through the data register SPI_DR.

The final sampling timing depends on the bit length of the transmitted data, the SPI data length of the master mode depends on the setting of the SPI_CFG2.DSIZE[3:0] bits, and the polarity of the SS output pin depends on the setting of the SPI_CFG1 register. For details of SPI transmission format, refer to 27.5.3 Transmission Format.

2) SPI Host Mode Initialization

- ① Set the communication configuration register 1 (SPI_CFG1) including the setting of baud rate, the number of frames to be used, and the setting of various delay times.
- ② Setting the communication configuration register 2 (SPI_CFG2). Set communication configuration register 2 (SPI_CFG2), including SS level setting, data shift order setting, various delay allowable bit setting, data format and clock polarity phase setting.
- ③ If you need to use interrupts, set the system's interrupt register.
- ④ If you need to use DMA, set the registers related to DMA.
- ⑤ Sets the input and output pins.
- ⑥ Set the SPI control register SPI_CR1, including the setting of the mode and operation mode, the setting of the self-diagnosis function, the setting of parity check, and so on.
- (7) Confirm the setting of SPI_CR1 register.
 - Ⓐ Clears various flag bits.
 - Ⓑ Set the interrupt license bit.

-
- ⑩ Set the SPE bit of control register SPI_CR1 to 1 to start the action.

27.6.3 Slave operation in SPI operation mode

1) SPI action description when acting as a slave

When the SPI_CFG2.CPHA bit is 0, if the SPI detects that the SS0 input signal has changed to a valid level, it needs to start driving valid data to the MISO output signal.

Therefore, when the CPHA bit is 0, the SS0 input signal level is changed from invalid to valid.

A trigger signal that is validly regarded as the start of serial transmission.

W h e n t h e CPHA bit is "1", if the SPI detects the first SCK edge while the SS0 input signal is at a valid level, it is necessary to start driving valid data by outputting a signal to MISO. Therefore, **w h e n** the CPHA bit is "1", the first SCK edge when the SS0 signal is at a valid level is regarded as the trigger signal to start serial transmission.

If the SPI detects the start of a serial transfer while the shifter is empty, it changes the shifter to full and cannot transfer data from TX_BUFF to the shifter during a serial transfer. If the shifter is full before the start of the serial transfer, the SPI maintains the shifter in a full state.

If the SPI detects the SCK edge of the last sample timing, this serial transfer ends, and the end time is independent of the SPI_CFG2.CPHA bit. If RX_BUFF is null, SPI copies the accepted data from the shifter to RX_BUFF at the end of the serial transfer, which can be read by accessing SPI_DR. SPI **changes the shifter to the null state** at the end of the serial transfer, which is independent of the state of RX_BUFF.

A mode fault error occurs if the SPI detects that the SS0 input signal is invalid during a serial transfer.

The final sampling timing depends on the bit length of the transferred data, the data length of the SPI in slave mode depends on the setting of the SPI_CFG2.DSIZE[3:0] bits, and the polarity of the SS0 input signal depends on the setting of the SPI_CFG1.SS0PV bits. For details about the SPI transmission format, refer to 27.5.3 Transmission Format.

Attention:

- When the SPI_CFG2.CPHA bit is "0", changing the SS0 input signal level from invalid to valid is regarded as a trigger signal to start serial transmission. If the SS0 input signal is fixed to the active state in the single-slave mode structure, the SPI will not be able to start serial transmission normally. Therefore, in a structure where the SS0 input signal is fixed to the active state, the CPHA bit must be set to "**1**" **for** the slave mode **SPI to** transmit and receive normally. If the CPHA bit needs to be set to "**0**", the SS0 input signal cannot be fixed.

2) SPI Slave Mode Initialization

- ① Set communication configuration register 1 (SPI_CFG1) which mainly includes the setting of the number of frames to be used.
- ② Set the communication configuration register [SPI_CFG2] including the transmission rate, data format, and clock polarity phase setting.
- ③ If you need to use interrupts, set the system's interrupt register.

-
- ④ If you need to use DMA, set the registers related to DMA.
 - ⑤ Sets the input and output pins.
 - 6) Set the SPI control register SPI_CR1, including the setting of the mode and operation mode, the setting of the self-diagnosis function, the setting of parity check, and so on.
 - (7) Confirm the setting of SPI_CR1 register.
 - ⑧ Clears various flag bits.
 - ⑨ Set the interrupt license bit.
 - ⑩ Set the SPE bit of control register SPI_CR1 to 1 to start the action.

27.6.4 Host actions in clock-synchronized operation mode

When the SPI_MDS bit in the SPI control register SPI_CR1 is 1, the SPI is in clock synchronous operation mode. In this mode, SPI uses only the three pins SCK, MOSI and MISO for communication, and the SS_i pin is released for normal I/O functions.

Although the SS_i pin is not used during the clock synchronization mode of operation, the internal operation of the module is the same as the SPI mode of operation. However, since there is no more SS_i input, no mode fault error is detected.

1) Description of actions when SPI is used as a host

When the SPI data transmit buffer register (TX_BUFF) is empty (the TDEF flag bit in the status register SPI_SR is 0) the **SPI** will update the SPI_DR data into the TX_BUFF after writing the data with the frame length set in the FTHLV[1:0] bits of the format control register SPI_CFG1 to the SPI data register (SPI_DR). If the shift register (**shifter**) is empty at this time, **SPI** copies the data of TX_BUFF to the shift register to start serial transmission.

SPI will change the shifter state to full when the data sent is copied to the shifter; it will change to empty when the serial transmission is finished. the state of the shifter cannot be read.

The serial transfer ends when the SPI sends the SCK edge required for the last sample timing, the end time is independent of the SPI_CFG2.CPHA bit. If the receive buffer (RX_BUFF) is empty, the SPI copies the data in the shifter to RX_BUFF after the end of the serial transmission, which can be read through the data register SPI_DR.

The final sampling timing depends on the bit length of the transmitted data, the SPI data length of the master mode depends on the setting of the SPI_CFG2.DSIZE[3:0] bits, and the polarity of the SS output pin depends on the setting of the SPI_CFG1 register. For details of SPI transmission format, refer to 27.5.3 Transmission Format.

2) Initialization settings for the host computer in clock synchronous operation mode

- ① Set communication configuration register 1 (SPI_CFG1) including baud rate setting, frame rate setting, and delay time setting.
- ② Setting of communication configuration register 2 (SPI_CFG2) Set communication configuration register 2 (SPI_CFG2), including data shift order setting, various delay allowable bit setting, data format and clock polarity phase setting.
- ③ If you need to use interrupts, set the system's interrupt register.
- ④ If you need to use DMA, set the registers related to DMA.
- ⑤ Sets the input and output pins.
- ⑥ Set the SPI control register SPI_CR1, including the setting of the mode and

operation mode, the setting of the self-diagnosis function, the setting of parity check, and so on.

- (7) Confirm the setting of SPI_CR1 register.
- ⑧ Clears various flag bits.
- ⑥ Set the interrupt license bit.

- ⑩ Set the SPE bit of control register SPI_CR1 to 1 to start the action.

27.6.5 Slave operation in clock synchronized mode of operation

1) SPI action description when acting as a slave

When the SPI_CFG2.CPHA bit is 0, it is necessary for the SPI to detect that the SS0 input signal has changed to a valid level as a trigger signal to start serial communication. Since the SS0 pin is not used in the clock synchronous operation mode, normal communication is not possible when the CPHA bit is 0.

When the CPHA bit is "1", if the SPI detects the first SCK edge while the SS0 input signal is at a valid level, it is necessary to start driving valid data by outputting a signal to MISO. Since the SS0 pin is not applicable in clock synchronous operation mode, the first SCK edge is regarded as the trigger signal to start serial transmission when the CPHA bit is "1".

If the SPI detects the start of a serial transfer while the shifter is empty, it changes the shifter to full and cannot transfer data from TX_BUFF to the shifter during a serial transfer. If the shifter is full before the start of the serial transfer, the SPI maintains the shifter in a full state.

If the SPI detects the SCK edge of the last sample timing, this serial transmission ends. If RX_BUFF is empty, SPI copies the received data of the shifter to RX_BUFF after the end of serial transmission, which can be read by accessing SPI_DR. SPI changes the shifter to empty state after the end of serial transmission, which is independent of the state of RX_BUFF. The final sampling timing depends on the bit length of the transferred data, and the data length of the slave mode SPI depends on the setting of SPI_CFG2.DSIZE[3:0] bits.

2) Initialization settings for slaves in clock synchronous operation mode

- ① Set communication configuration register 1 (SPI_CFG1) which mainly includes the setting of the number of frames to be used.
- ② Set the communication configuration register 2 (SPI_CFG2). Set communication configuration register 2 (SPI_CFG2), including transmission rate, data format and clock polarity phase setting.
- ③ If you need to use interrupts, set the system's interrupt register.
- ④ If you need to use DMA, set the registers related to DMA.
- ⑤ Sets the input and output pins.
- ⑥ Set the SPI control register SPI_CR1, including the setting of the mode and operation mode, the setting of the self-diagnosis function, the setting of parity check, and so on.
- (7) Confirm the setting of SPI_CR1 register.

- ⑧ Clears various flag bits.
- ⑨ Set the interrupt license bit.
- ⑩ Set the SPE bit of control register SPI_CR1 to 1 to start the action.

27.6.6 Processing flow of several SPI actions

1) Data transfer processing flow when SPI is used as a host.

- ① Wait for an interrupt that the data send buffer register is empty or confirm that the data send buffer register is empty by polling.
- ② Write the data to be sent to the data register SPI_DR.
- ③ Repeat ①② until the last data is sent.
- ④ Clear TXIE, the transmit data register air interrupt allow bit, and set IDIE, the SPI idle interrupt allow bit, to 1.
- ⑤ Send SPI idle status interrupt.
- ⑥ Set SPE to 0 to stop SPI action and clear IDIE to zero.

2) Data Receiving Processing Flow

- ① Wait for a data reception buffer register full interrupt or confirm that the data reception buffer register is in a full state by polling.
- ② Read data from the receive buffer register by accessing SPI_DR.
- ③ Repeat step ①② until the last received data is read.
- ④ Clear the interrupt allow bit RXIE of the data reception buffer register to zero.

3) Communication Error Handling Flow

Wait for a communication error interrupt or confirm the communication error flag bit by polling.

- (MODFERF/OVRERF/UDRERF/PERF) is set to 1.
- ② Confirm the SS0 status and troubleshoot the mode fault error.
 - ③ Clear SPE to stop SPI operation.
 - ④ Clear all SPI interrupt allow bits to block SPI interrupts.
 - ⑤ Determine the type of communication error by the error flag bit and perform communication error processing.
 - ⑥ Clear the error flag bit to zero.
- (vii) Start SPI to restart communication.

27.7 Parity bit self-diagnostics

The parity check circuit consists of a parity bit for transmitting data and an error detection section for receiving data. The parity check circuit can be troubleshooted using the self-diagnostic function according to the flow shown below.

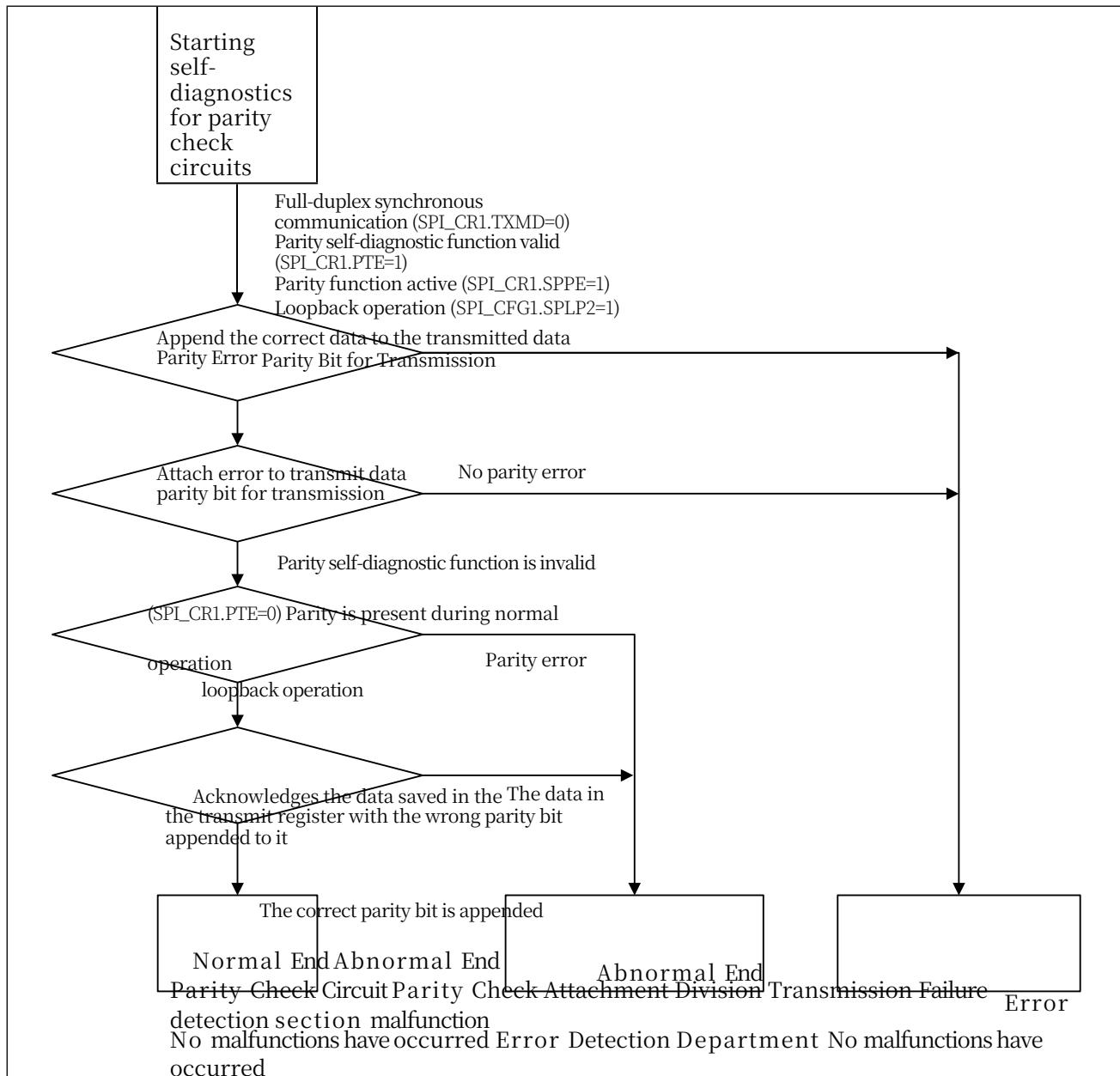


Figure 27-9 Parity Check Flow

27.8 error detection

In a normal SPI serial transfer, the system sends data serially by writing to the SPI_DR register and obtains data received serially by reading to the SPI_DR register. However, due to the state of the transmit/receive buffer and the state of the SPI at the start/end of the serial transfer, an abnormal transfer may occur in some cases. When an abnormal transfer occurs, the SPI detects the transfer as an underload error, an overload error, a parity error, or a mode fault error. The correspondence between abnormal transmission and SPI error detection is shown in Table 27-7 below.

Table 27-7 Error Detection Correspondence Table

serial number	Conditions of occurrence	SPI operation	detection error
①	Write SPI_DR register in transmit buffer full state	<ul style="list-style-type: none"> - Holds the contents of the transmit buffer 	not have
		<ul style="list-style-type: none"> - Loss of write data 	
②	Read the SPI_DR register with the receive buffer empty	Output the last serial received data	not have
(iii)	In slave mode, serial transmission is started without transferring the transmit data to the shift register.	- Abort serial transmission	underload error
		- Loss of transmitted and received data	
		- Stop driving the MISO output signal	
		- Stop SPI function	
④	Slave mode: the effective level width of the SS0 pin has not reached the time required for data transmission.	<ul style="list-style-type: none"> - abort transmission - Loss of transmitted and received data - Stop SPI function 	Schema Error
⑤	End serial transmission in receive buffer full state	- Holds the contents of the receive buffer	overload error
		- Loss of received data	
(vi)	Full-duplex synchronous serial communication with parity check function Valid state, wrong parity bit received	Parity error flag is valid	parity error

In the case described in ①, no detection error occurs for SPI. To prevent data omission during writing data to the SPI_DR register, data must be written to the SPI_DR register by sending a buffer empty interrupt. Similarly, no detection error occurs for SPI in case ②. To prevent extraneous data from being read in, the data read of SPI_DR must be performed by receiving a register full interrupt request.

27.8.1 underload error

When the MSTR bit is 0, the SPI operates in the slave state. If the transmission data is not ready before a valid level is received at the SS0 pin when SPE is set to 1, an underload error occurs in the SPI, and the flags SPI_SR.MODFERF and SPI_SR.

When an underload error is detected, the SPI will stop driving the signal output and set SPI_CR1.SPE to 0 at the same time.

Monitoring for underload errors can be done by accessing the SPI_SR register directly, or by reading the SPI_SR using an SPI error interrupt. If the error interrupt is not used, use the polling method to monitor underload errors.

When SPI_SR.MODFERF is 1, the system disables writing 1 to the SPE bit. To set SPI_CR1.SPE to 1 to enable the SPI function, the MODFERF flag must be cleared first.

27.8.2 Schema Error

SPI_CR1.MODFE should not be set to 1 when SPI is in host mode. In slave mode, a mode fault occurs when the effective level width of SSI does not reach the time required to transmit data, SPI_SR.MODFERF is set to 1, and SPI_CR1.SPE is set to 0. When a transmission is required, clear SPI_SR.MODFREF to zero, then set SPI_CR1. When transmission is required, clear SPI_SR.MODFREF and then set SPI_CR1.SPE to 1.

27.8.3 overload error

If the serial transfer is terminated in the receive buffer full state, an overload error occurs in SPI, and the SPI_SR.OVRERF flag is set to 1. Since SPI does not copy the data from the shift register to the receive buffer in the OVRERF flag 1 state, the receive buffer holds the received data before the error occurred. To set the OVRERF flag to 0, you need to read the SPI_SR register when the OVRERF flag is 1 before writing "0" to the OVRERF flag.

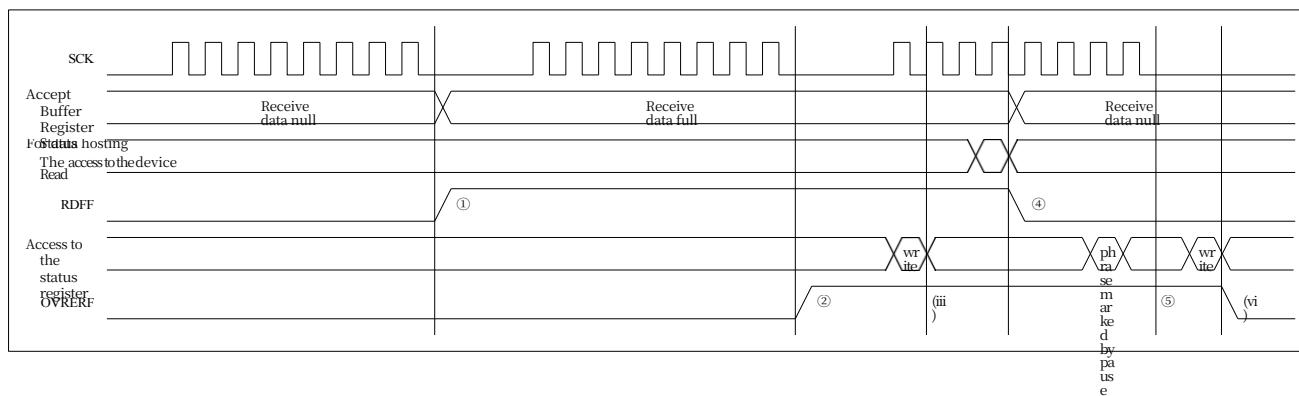


Figure 27-10 Overload Error Handling

The following describes the operation contents of the flags in the timing sequence shown in ① to ⑥ in the figure:

- ① End the serial transmission in the state of acceptance buffer empty, SPI normal action, copy the data from the shift register to the acceptance buffer register, and set the RDFF flag to 1.
- ② Ending a serial transmission with the receive buffer full, SPI detects an overload error and sets the OVRERF flag to 1. SPI will not copy the data from the shift register to the receive buffer. Even if the PAE bit is "1", no parity error is detected.
- ③ Write operation to the OVRERF bit is not possible without a read to the SPI_SR register, and clearing fails.
- ④ A read access to the data register SPI_DR (SPI) to read the data from the receive buffer. the RDFF flag changes to 0. The OVRERF flag does not change to 0 even if the receive buffer state is empty at this time.
- ⑤ If the serial transmission is terminated with the OVRERF flag "1" (overload error),

SPI will not copy the data from the shift register to the receive buffer register and will not generate a receive buffer full interrupt, and the RDFF flag will remain 0. Even if the PAE bit is "1", **no parity error will be detected.** "", no parity error is detected. In the state where an overload error occurs, if the serial transfer is terminated without copying the receive data from the shift register to the receive buffer, **SPI** judges the shift register to be in the empty state and allows the data to be transferred from the transmit buffer register to the shift register.

- (6) The SPI sets the OVRERF flag to 0 by reading the SPI_SR register with the OVRERF flag at 1 and then writing 0 to the OVRERF flag.

Monitoring for overload errors can be done by accessing the SPI_SR register directly or by accessing the SPI_SR register using the SPI error interrupt. During serial transfers, the occurrence of an overload error must be detected as early as possible by methods such as reading the SPI_SR register immediately after reading the SPI_DR register.

Normal receive operation is only possible after the OVRERF flag is changed to 0.

In host mode, if the communication auto suspend function is enabled (set SPI_CR1.CSUSPE bit to 1) SPI will pause the communication clock in the last sampling cycle before an overload error occurs, at this time, the SPI will remain in the normal communication state and an overload error will not occur as the last bit of data reception has not yet been completed in the shift register. During the pause of the communication clock, the receive buffer register can be read, and after reading, the receive buffer register will become empty, and the SPI will restart the communication clock to finish receiving the last bit of data. For detailed actions, refer to Figure 27-11 and Figure 27-12 below.

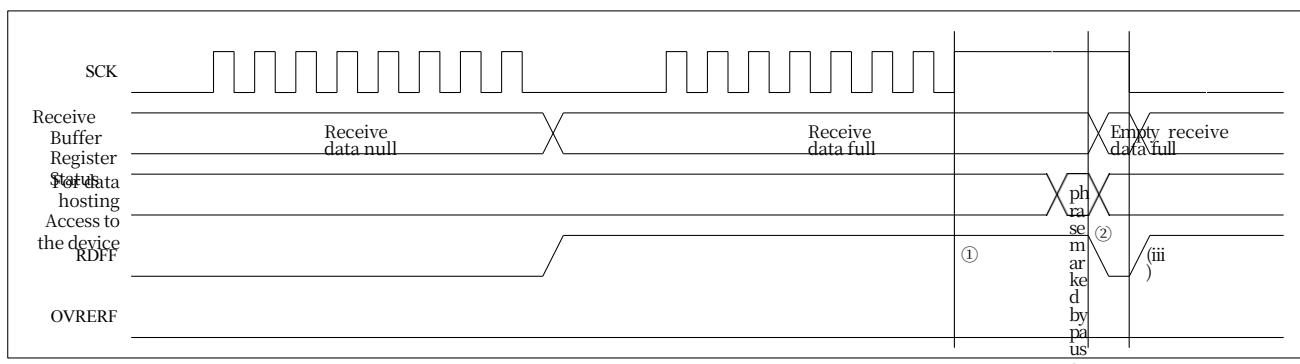


Figure 27-11 Diagram of the action when the clock auto stop function is enabled (CPHA=1)

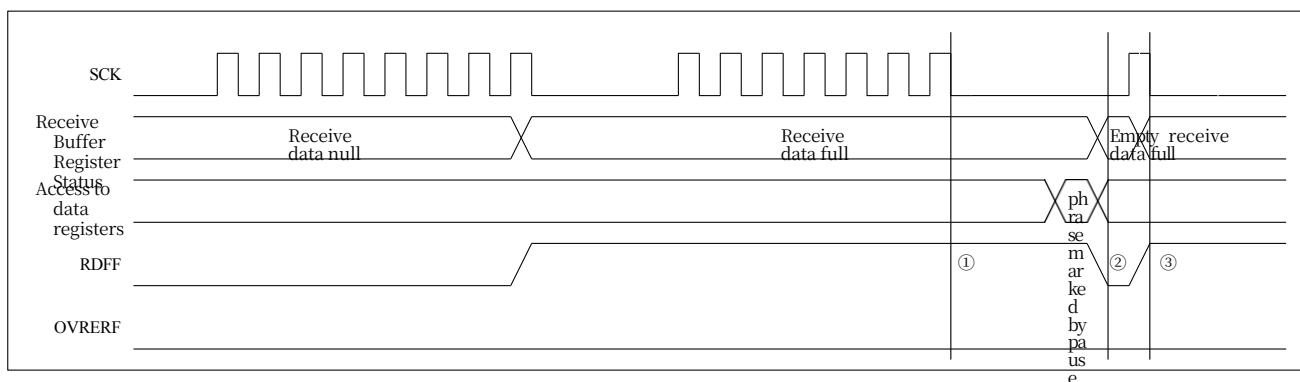


Figure 27-12 Diagram of the action when the clock auto stop function is enabled (CPHA=0)

The following describes the operation contents of the flags in the timing sequence shown in ① to ③ in the figure:

- ① When in the receive buffer register full state, SPI pauses the communication clock before the last bit of data is received. No overload error will occur at this time.
- ② When the data in the receive buffer register is read by accessing SPI_DR, the receive buffer register becomes empty, the RDFF flag is cleared to zero, and the SPI

restarts the communication clock to complete the data communication of the last bit.

- ③ When the communication of the last bit of data 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.

27.8.4 parity error

With the SPI_CR1.TXMDS bit at "0" and the SPI_CR1.PAE bit at "1", the SPI performs parity check at the end of the full-duplex synchronous serial communication. When SPI detects a parity error in the received data, it sets the SPI_SR.PERF flag to 1. In the SPI_SR.OVRERF bit "1" state, SPI does not detect a parity error in the received data because SPI does not copy the data from the shift register to the receive buffer. To clear the PERF flag, read the SPI_SR register with the PERF flag at "1" and then write "0" to the PERF flag.

An example of the operation of the OVRERF flag and the PERF flag is shown in Figure 27-13 below. In the example shown, the SPI performs an 8-bit serial transfer for full-duplex synchronous serial communication with the SPI_CR1.TXMDS bit at 0 and the SPI_CR1.PAE bit at 1.

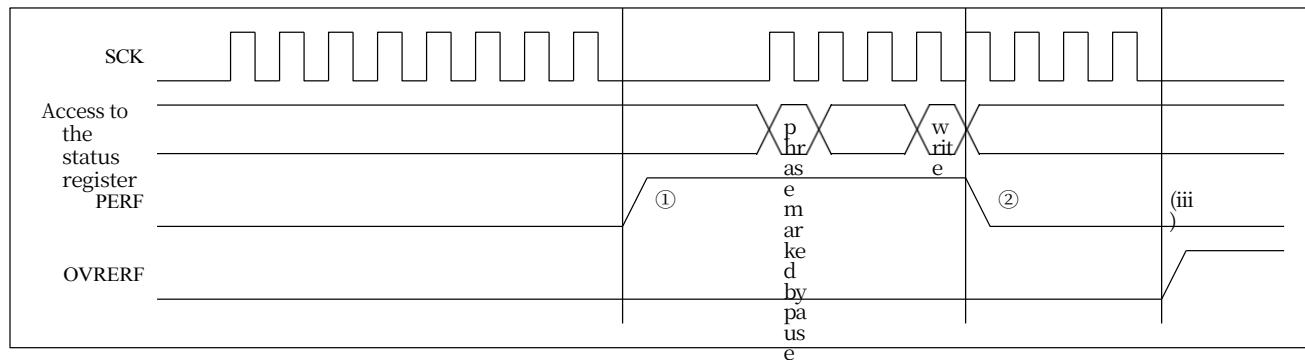


Figure 27-13 Parity Error

The following describes the contents of the operation of the flags in the timing sequence shown in Figs. ① to ③:

- ① SPI does not detect an overload error and the serial transmission ends normally. SPI copies the data from 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 Read the SPI_SR register with the PERF flag at 1 and then write 0 to the PERF flag to clear the RERF flag to zero.
- ③ SPI detects an overload error, at this time, SPI will not copy the data from the shift register to the receive buffer, SPI will not parity check the data, and no parity check error will occur.

The occurrence of a parity error can be monitored by accessing the SPI_SR register directly or by reading the SPI_SR register via the SPI error interrupt. The occurrence of a parity error must be monitored as early as possible during serial transmission by accessing, for example, the status register SPI_SR.

27.9 Initialization of SPI

The SPE bit can be cleared to zero by a write operation or mode fault error detection to disable the SPI function and initialize some of the SPI functions. If a system reset occurs, all SPI functions are initialized.

27.9.1 Clear SPE bit for initialization

When the SPI_CR1.SPE bit is 0, the SPI performs the following initialization operations:

- Aborts an ongoing serial transfer.
- If it is in the slave state it stops driving the output signal (the state changes to Hi-Z)
- Initializes the SPI internal state.
- The transmit buffer register is cleared and the SPI_SR.TDEF flag is set to 1.

When initializing by clearing the SPE bit to zero, the control bits of the SPI are not initialized. Therefore, by resetting the SPE bit to 1, the SPI can be started in the same transmission mode as before initialization.

Clearing the SPE bit does not initialize the error flag bits and sequence status. Therefore, even after SPE is cleared, the occurrence of an error during SPI transmission can be confirmed by reading the data in the receive buffer.

Since clearing the SPE bit clears the transmit buffer register and sets the SPI_SR.TDEF flag to 1. Therefore, if the SPI_CR1.TXIE bit is set to 1 after initialization, an interrupt is generated with the SPI transmit buffer register empty. To prevent this interrupt from occurring in the system, the TXIE bit must be set to 0 at the same time as the SPE bit is cleared.

27.9.2 System Reset Initialization

Initialization via system reset will initialize all control bits, status bits and data registers of the SPI.

27.10 source of interruption

The SPI interrupt sources are receive buffer full, transmit buffer empty, mode fault, overload, underload, parity error and SPI idle. The receive buffer full and transmit buffer empty interrupts can be used to start the DMA for data transfer.

Overload, underload, and parity error interrupts are grouped together as SPI error interrupts SPEI, so it is necessary to use flags to determine the source of the interrupt that actually occurs. The SPI interrupt sources are specified in Table 27-8. Once the interrupt condition is established, the corresponding interrupt request is generated.

The interrupt sources of the receive buffer full and transmit buffer empty need to be cleared by changing the buffer state through data transmission. When using DMA to transmit or receive, DMA must be configured first, and SPI must be set after DMA is set to an action-permitted state.

Table 27-8 SPI Interrupt Source Description

source of interruption	abbreviation	interrupt condition	Initiate DMA
Receive buffer full	SPRI	When the receive buffer becomes full with the SPI_CR1.RXIE bit at "1".	(usually used in the negative) have the possibility of
Generation buffer empty	SPTI	When the transmit buffer becomes empty with the SPI_CR1.TXIE bit at "1".	(usually used in the negative) have the possibility of
SPI errors (overload, underload, parity error)	SPEI	SPI_SR.OVRERF, SPI_SR.PERF or SPI_SR.MODFERF, and SPI_SR.UDRERF flags are changed when the SPI_CR1.EIE bit is "1". For "1"	must not
SPI Idle Interrupt	SPII	When the IDLF flag changes to "0" in the state where the SPI_CR1.IDIE bit is "1".	must not

27.11 Available event trigger sources

The following are the main types of event trigger sources generated by SPI that are available for use:

- Data send buffer register empty

- Data receive buffer register full
- SPI communication errors (including overload, underload, parity, etc. errors)
- SPI is inactive
- End of SPI communication

Users can write the vectors corresponding to the above event triggering sources into different trigger object registers to realize various event triggering functions.

Refer to [Interrupt Controller (INTC] for the vectors corresponding to the above event trigger sources.

27.12 Register Description

List of registers

Register base address: SPI1_BASE:0x4001_C000;

SPI2_BASE:0x4001_C400 SPI3_BASE:0x4002_0000.

SPI4_BASE:0x4002_0400

register name	offset address	reset value
SPI Data Register SPI_DR	0x00	0x0000_0000
SPI Control Register SPI_CR1	0x04	0x0000_0000
SPI communication configuration register 1 SPI_CFG1	0x0C	0x0000_0010
SPI Status Register SPI_SR	0x14	0x0000_0020
SPI Communication Configuration Register 2 SPI_CFG2	0x18	0x0000_0F1D

27.12.1 SPI Data Register (SPI_DR)

Reset value: 0x0000_0000

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

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b0	SPD [31:0]	serial data	SPI Data Storage	R/W

27.12.2 SPI Control Register (SPI_CR1)

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
PAE	PAOE	PATE	MODFE	IDIE	RXIE	TXIE	EIE	CSUSPE	SPE	SPLPBK2	SPLPBK	MSTR	-	TXMDS	SPIMDS

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b16	Reserved	-	Reads "0" and writes "0".	R/W
b15	PAE	Parity Check Allowed R/W	0: send data without additional parity bits, receive data without parity checking 1: Send data with additional parity bit, receive data for parity checking (SPI_CR1.TXMDS=0); send data with additional parity bit to connect to the Receive data without parity check (SPI_CR1.TXMDS=1)	
b14	PAOE	Parity mode selection	0: Select even parity for transmit and receive 1: Select odd parity for sending and receiving	R/W
b13	PATE	Parity self-diagnostics	0: Parity self-diagnostics not available 1: Parity self-diagnostic function effective	R/W
b12	MODFE	Mode Failure Error Detection Allowed Failure Error Detection Disallowed	0: Mode 1: Allow mode fault error detection	R/W
b11	IDIE	SPI Idle Interrupt Allowed	0: Disable idle interrupt request generation 1: Allow idle interrupt request generation	R/W
b10	RXIE	SPI receive interrupt allow	0: Disable SPI receive interrupt request generation 1: Allow SPI receive interrupt request generation	R/W
b9	TXIE	SPI Transmit Interrupt Allow SPI transmit interrupt request generation	0: Disable 1: Allow SPI transmit interrupt request generation	R/W
b8	EIE	SPI error interrupt allowed	0: Disable SPI error interrupt request generation 1: Allow SPI error interrupt request generation	R/W
b7	CSUSPE	Communication auto-resume function is allowed Communication auto-hang-up function is invalid	0: 1: The communication auto-suspend function is effective	R/W
b6	SPE	SPI function allowed	0: SPI function disabled 1: SPI function is valid	R/W
b5	SPLPBK2	SPI loopback 2 bits	0: Normal mode 1: Loopback mode (send data = receive data)	R/W
b4	SPLPBK	SPI Loopback Bit	0: Normal mode 1: Loopback mode (inverse of sent data = received data)	R/W
b3	MSTR	SPI Master-Slave Mode Selection mode	0: Slave 1: Host Mode	R/W
b2	Reserved	-0" for reading, "0" for writing.	Reads	

		"0", writes "0".	R/W	
b1	TXMDS	Communication mode selection duplex synchronous serial communication		0: Full R/W
		1: Transmit serial communication only		
b0	SPIMDS	SPI Mode Selection	0: SPI operation (4-wire) 1: Clock synchronized operation (3-wire)	R/W

27.12.3 SPI communication configuration register 1 (SPI_CFG1)

Reset value: 0x0000_0010

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	MIDI[2:0]			-	MS	DL[2:0]	-	-	MSSI[2:0]			-	-	-	-
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	SS3 PV	SS2 PV	SS1 PV	SS0 PV	-	SPR DTD	-	-	-	-	FTHLV[1:0]	

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31	Reserved	-	Reads "0", writes "0".	R/W
b30~b28	MIDI[2:0]	Host next data access interval idle Time Setting Position	0 0 0: 1 SCK + 2 PCLKs 0 0 1: 2 SCKs + 2 PCLKs 0 1 0: 3 SCKs + 2 PCLKs 1 0 0: 4 SCKs + 2 PCLKs 1 0 1: 5 SCKs + 2 PCLKs 1 0 2: 6 SCKs + 2 PCLKs 1 0 3: 7 SCKs + 2 PCLKs 1 0 4: 8 SCKs + 2 PCLKs	R/W
b27	Reserved	-Reserved	Read "0", write "0".	R/W
b26~b24	MSSD[2:0]	Host SS Invalid Delay Set Positioning	0 1 1: 4 SCKs 1 0 0: 5 SCKs 1 0 1: 6 SCKs 1 0 2: 7 SCKs 1 0 3: 8 SCKs	R/W
b23	Reserved	-	Reads "0", writes "0".	R/W
b22~b20	MSSI[2:0]	Host SS Idle Time Setting Location	0 1 1: 4 SCKs 1 0 0: 5 SCKs 1 0 1: 6 SCKs 1 0 2: 7 SCKs 1 0 3: 8 SCKs	R/W
b19~b12	Reserved	-	Reads "0", writes "0".	R/W
b11	SS3PV	SS3 signal Polarity setting	0: Low level of SS3 signal active 1: High level of SS3 signal active	R/W
b10	SS2PV	SS2 signal Polarity setting	0: Low level of SS2 signal is active 1: High level of SS2 signal active	R/W

b9	SS1PV	SS1 signal Polarity setting	0: Low level of SS1 signal is active 1: High level of SS1 signal active	R/W
----	-------	--------------------------------	--	-----

b8	SSOPV	SS0 signal Polarity setting	0: Low level of SS0 signal is active 1: High level of SS0 signal is active	R/W
b7	Reserved	-	Reads "0" and writes "0".	R/W
b6	SPRD TD Read receive buffer	Data register read object selection R/W	1: SPI_DR Read transmit buffer (must be read with TDEF=1)	0: SPI_DR
b5	Reserved	-0" for reading, "0" for writing.	Reads "0", writes "0".	R/W
b4	Reserved	-1" for reading, "1" for writing.	Reads "1", writes "1".	R/W
b3~b2	Reserved	-0" for reading, "0" for writing.	Reads "0", writes "0".	R/W
b1~b0				
0 0:1 Frame				
1 0:2 frames				
2 0:3 frames				
3 1:4 frame				
b1~b0	FTHLV[1:0]	Frame count setting		R/W

27.12.4 SPI Status Register (SPI_SR)

Reset value: 0x0000_0020

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	RDF F	-	TDE F	UDR ERF	PER F	MOD FER F	IDL NF	OVR ERF

classifier for honorific people	marking	celebrity	functionality	fill out or in (informati on on a form)
b31~b8	Reserved	-	Reads "0" and writes "0".	R/W
b7	RDFF	Receive buffer full flag	0: No data in SPI_DR register 1: SPI_DR register has data Hardware set, clear, write "1" when writing	R
b6	Reserved	-	Reads "0", writes "0". 0: Send buffer has data	R/W
b5	TDEF	Send buffer empty flag	1: No data in the occurrence buffer Hardware set, clear, write "1" when writing	R
			0: Mode Failure Error Occurrence (MODFERF=1)	
b4	UDRERF	Underload error flag	1: Underload error occurs (MODFERF=1) When MODFERF=0, this bit is initialized Hardware set, read 1 write 0, status bit clear	R/W
b3	PERF	Parity error flag	0: No parity error occurred 1: Parity error occurred Hardware set, read 1 write 0, status bit clear	R/W
b2	MODFERF	Mode Fault Error Flag	0: Unsent mode fault error 1: Mode failure error occurred Hardware set, read 1 write 0, status bit clear	R/W
b1	IDLNF	SPI Idle Flag	0: SPI is idle 1: SPI is transmitted Hardware Set, Clear	R
b0	OVRERF	Overload Error Flag	0: No overload error occurred 1: Overload error occurred Hardware set, read 1 write 0, status bit clear	R/W

27.12.5 SPI Communication Configuration Register 2 (SPI_CFG2)

Reset value: 0x0000_0F1D

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
MSS IE	MSS DLE	MID IE	LSB F	DSIZE[3:0]				SSA[2:0]			MBR[2:0]			CPO L	CPH A

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b16	Reserved	-	Reads "0" and writes "0".	R/W
b15	MSSIE	SCK Delay Allowed	0: SCK delay is 1 SCK 1: SCK delay is the set value of MSSI	R/W
b14	MSSDLE	SS Invalid Delay Allowed	0: SS invalid delay is 1 SCK 1: SS Invalid delay is the set value of MSSDL	R/W
b13	MIDIE	SPI Next Access Delay Allowed	0: Next access delay is 1 SCK+2 PCLK 1: The next access 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]	SPI Data Length Setting	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 places 1 0 1 0: 14 bits 1 0 1 1: 15 places 1 1 0 0: 16 bits 1 1 0 1: 20 bits 1 1 1 0: 24 bits 1 1 1 1: 32-bit	R/W
b7~b5 position	SSA[2:0]	SS signal valid set	b7~b5 0 0 0: SS0 0 0 1: SS1 0 1 0: SS2 0 1 1: SS3 1 x x: Disable	basic bit rate 0 1 1: Selects 16 divisions of the basic bit rate
b4~b2 position	MBR[2:0]	Bit rate divider set	setting b4~b2 0 0 0: Selects 2 divisions of the basic bit rate 0 0 1: Selects 4 divisions of the basic bit rate 0 1 0: Selects 8 divisions of the	

R/W

			1 0 0: Selects 32 divisions of the basic bit rate	
			1 0 1: Selects the 64 divisions of the fundamental bit rate	
			1 1 0: Selects 128 divisions of the basic bit rate	
			1 1 1: Selection of 256 divisions of the fundamental bit rate	
b1	CPOL	SCK polarity setting position R/W	0: SCK at idle is at Low level 1: SCK at idle is at High level	R/W
b0	CPHA	SCK phase setting position R/W	0: data sampling at odd edges, data change at even edges 1: Data changes on odd edges, data sampling on even edges	

28 Four-wire Serial Peripheral Interface (QSPI)

28.1 brief

The Quad Serial Peripheral Interface (QSPI) is a memory control module designed to communicate with serial ROMs with SPI-compatible interfaces. The main targets are serial Flash, serial EEPROM and serial FeRAM.

Table 28-1 QSPI Key Specifications

parameters	norm
channel number	1 channel
SPI	- Supports multiple protocols such as Extended SPI, 2-wire SPI and 4-wire SPI
	- Supports SPI mode 0 and SPI mode 3
	- Address line width selectable 8-bit/16-bit/24-bit/32-bit
timing adjustment	Timing can be adjusted to support a variety of serial flash memories
flash memory readout	- Supports multiple reading methods a. Standard reading/fast reading b. 2-wire output fast readout / 2-wire input/output fast readout c. Four-wire output fast readout/four-wire input/output fast readout
	- Free setup commands
	- Adjustable number of virtual cycles
	-16 byte pre-read function
	- Status query function
	-SPI bus cycle extension function
	- XIP control function
direct communication function	Flexible and extensive support for a wide range of serial flash software control commands, including erase, write, ID read and power-down control.
source of interruption	hardware error interrupt
Module stop function	Power consumption can be reduced by module stopping

QSPI architect ure

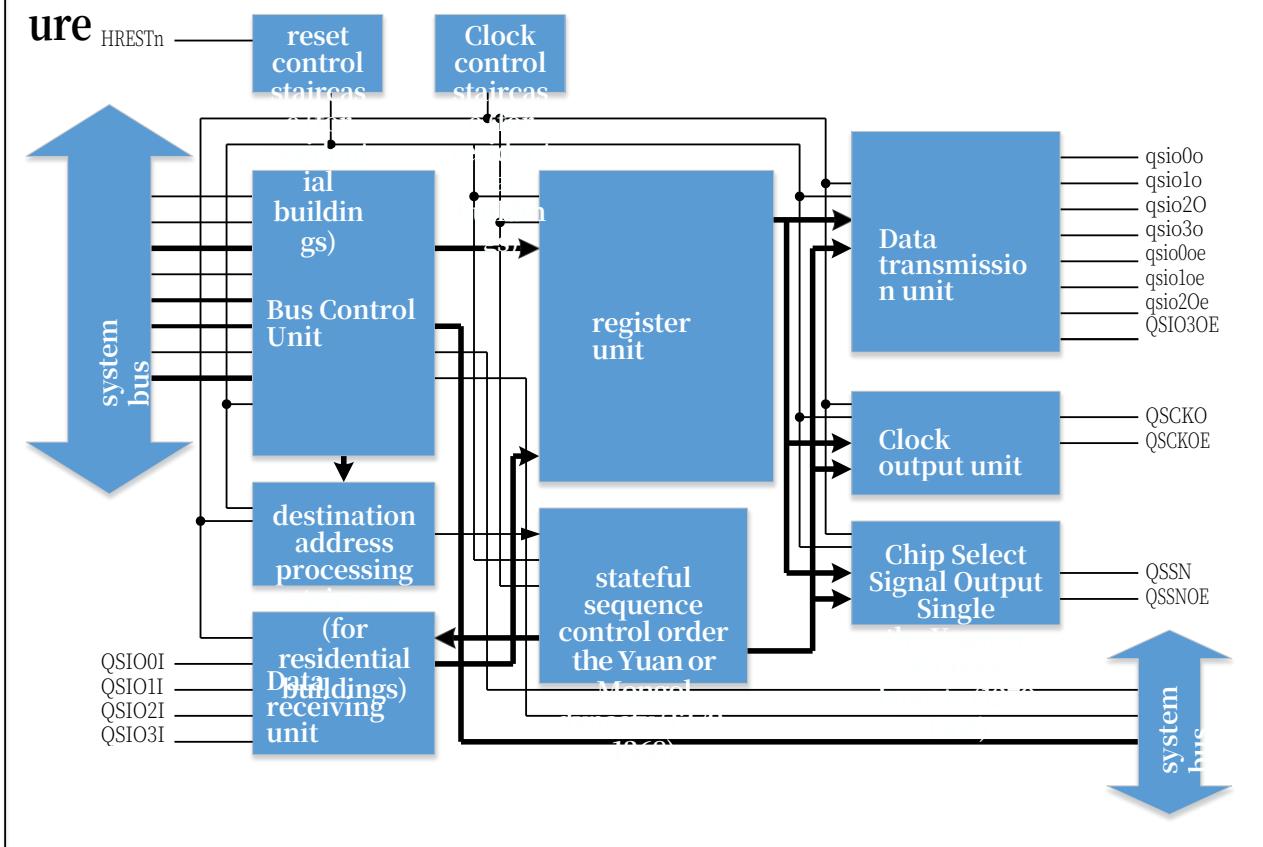


Figure 28-1 Module

Composition Diagram of

QSPI Table 28-2 QSPI

Pins

pin name	Input/Output	Functional Description
QSCK	exports	QSPI clock output pin
QSSN	exports	QSPI Slave Selection Pin
QSI00	Input/Output	Data cable 0
QSI01	Input/Output	Data cable 1
QSI02	Input/Output	Data Cable 2
QSI03	Input/Output	Data Cable 3

28.2 memory map

28.2.1 Internal bus space

The location of the serial flash memory and associated control registers in the AHB bus space is determined by the overall address range configuration.

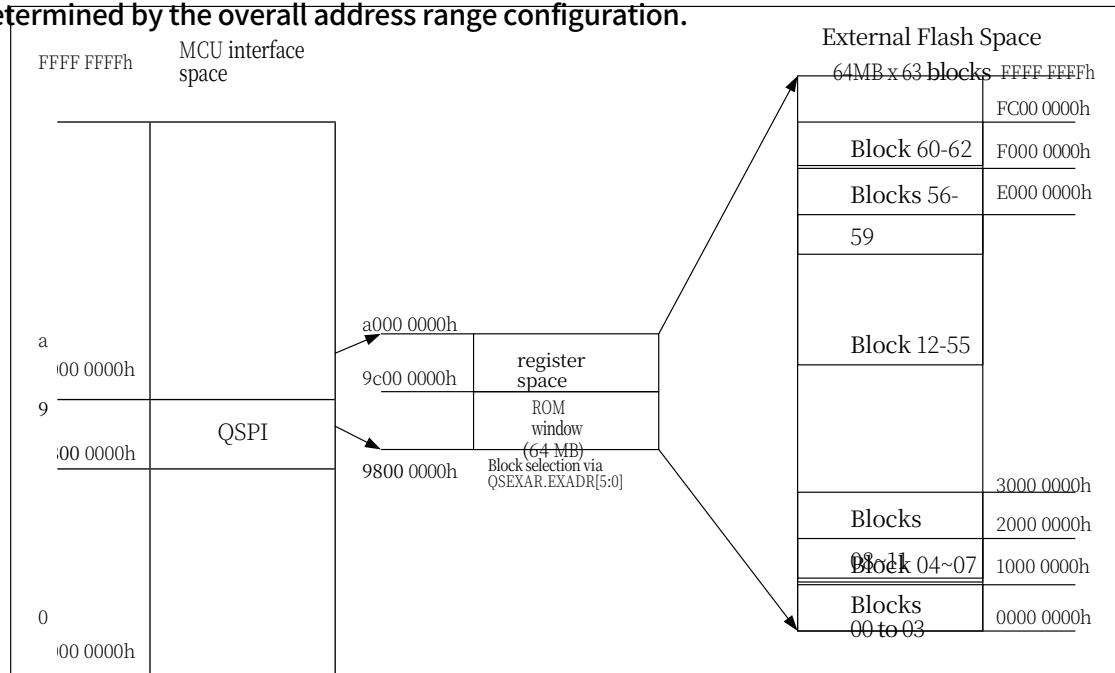


Figure 28-2 Default Area Settings and AHB Bus Space Memory Mapping Relationships

28.2.2 Address width of ROM space and bus

The QSPI has a 32-bit address bus width to match the serial flash memory. Whenever a read access is made to the ROM space of the QSPI, the QSPI bus is automatically activated to transfer the data read from the serial flash memory.

QSPI can not only use 32-bit address bus width, but also can choose to use 8-bit/16-bit/24-bit address bus width by setting AWSL[1:0] in QSFCR register.

If an 8-bit/16-bit/24-bit address bus width is selected, only the lower space with the matching address can be accessed normally, i.e., accessing the high serial flash image space in QSPI will recur the contents of the lower space.

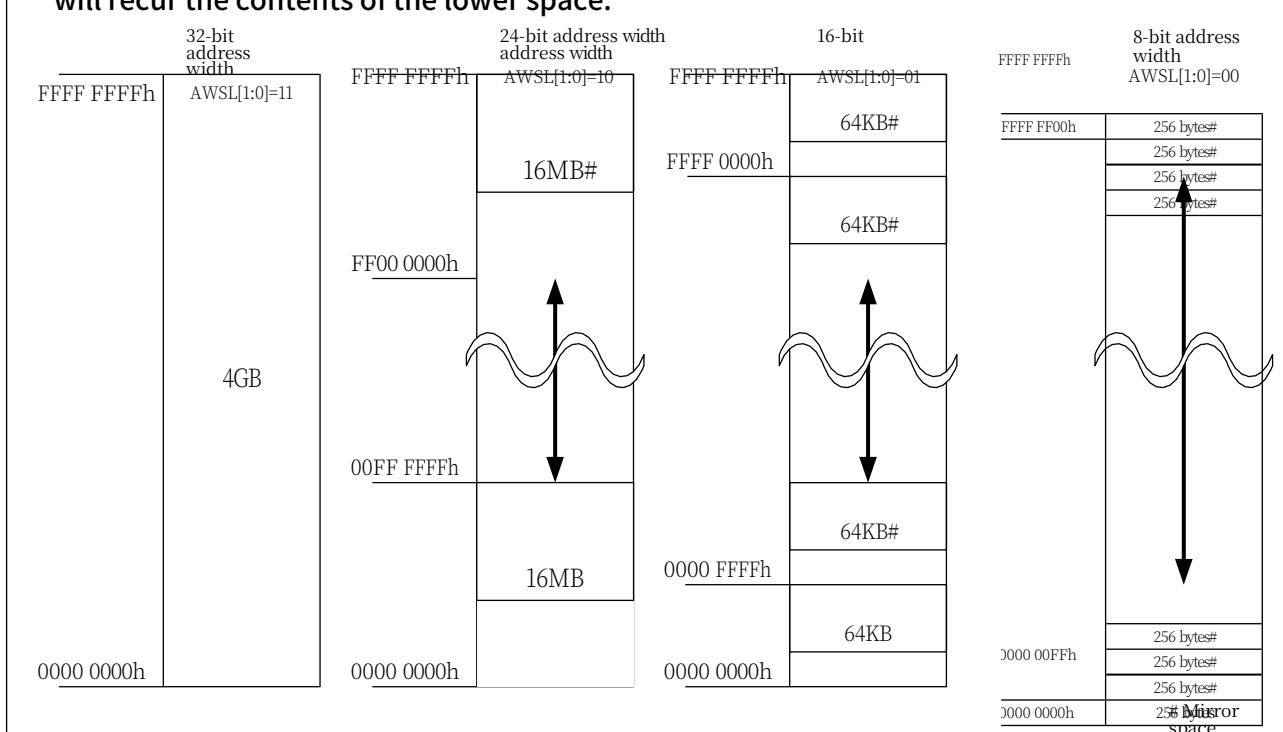


Figure 28-3 QSPI-ROM Space Memory Map

Attention:

ion:

-Address Bus Width The address bus width can be selected to use 8-bit/16-bit/24-bit by setting AWSL[1:0] in the QSFCR register.

/32-bit.

28.3 QSPI Bus

28.3.1 SPI protocol

This QSPI supports three protocols: Extended SPI, 2-wire SPI and 4-wire SPI. The initial default protocol is Extended SPI. The protocols of each phase can be configured by setting the DPRSL[1:0]/APRSL[1:0]/IPRSL[1:0] bits in the QSCR register. The extended SPI protocol uses only a single line of the QSIO0 pin for command output, and the address and data thereafter are output using a single line/two lines/four lines depending on the specific read mode command.

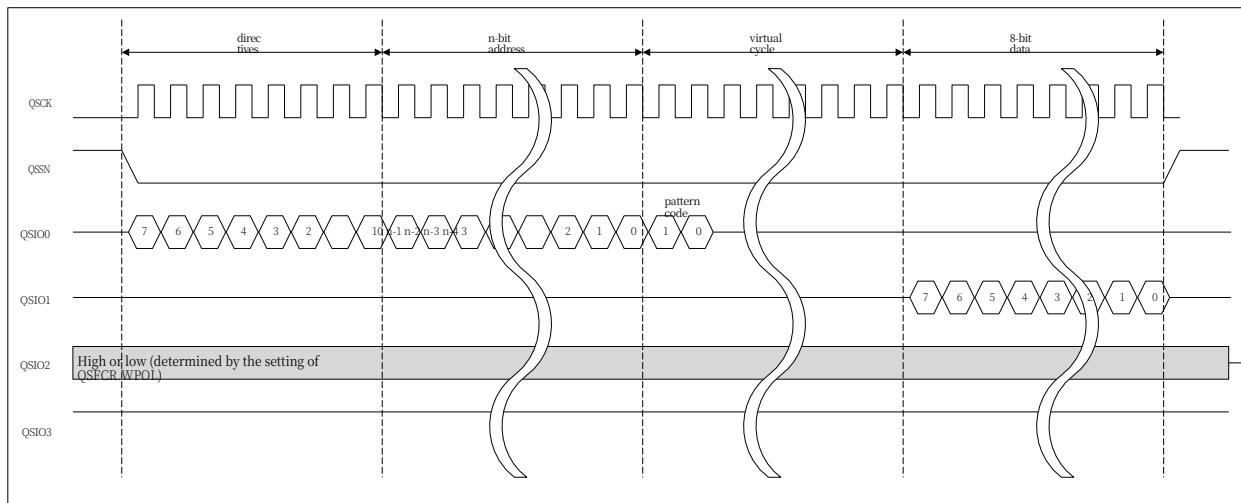


Figure 28-4 Extended SPI Protocol Action Diagram 1 (Fast Read Mode)

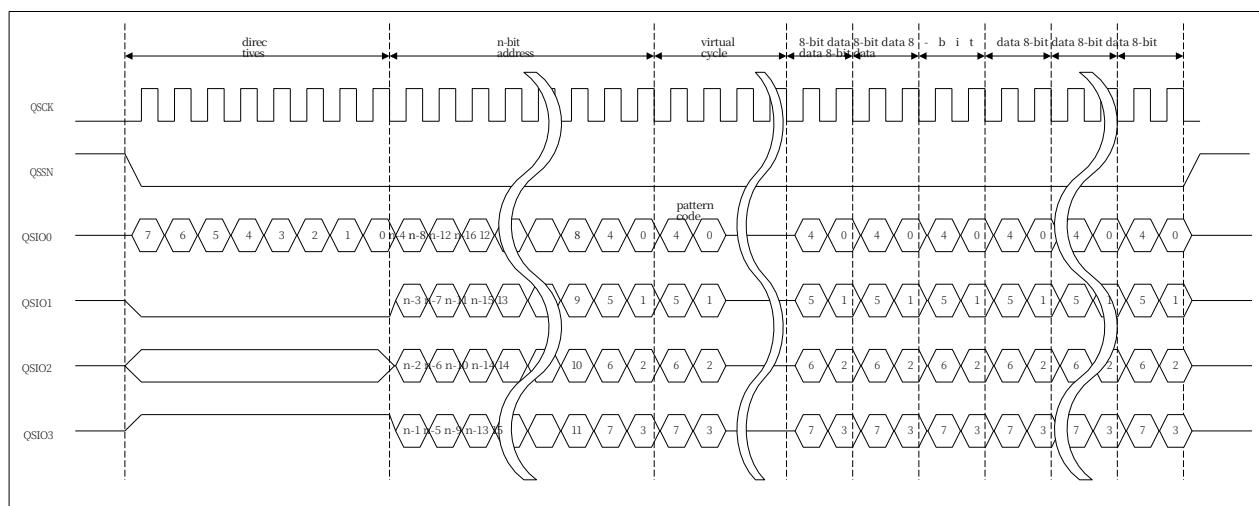


Figure 28-5 Extended SPI Protocol Action Schematic 2 (4-Wire Input/Output Fast Read Mode)

The 2-wire SPI protocol uses the QSIO0 and QSIO1 pins to perform the corresponding operations, including issuing commands, addressing, receiving data, and so on.

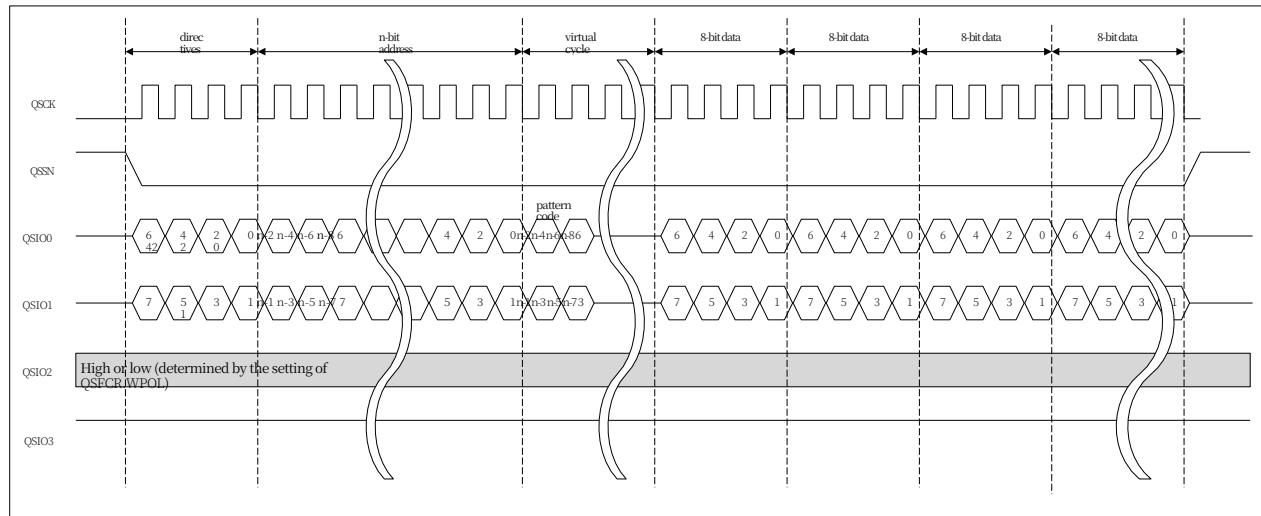


Figure 28-6 2-Wire SPI Protocol Action Schematic (Fast Read Mode)

The four-wire SPI protocol uses four pins, QSI00, QSI01, QSI02, and QSI03, to perform all operations related to issuing commands, addressing, and receiving data.

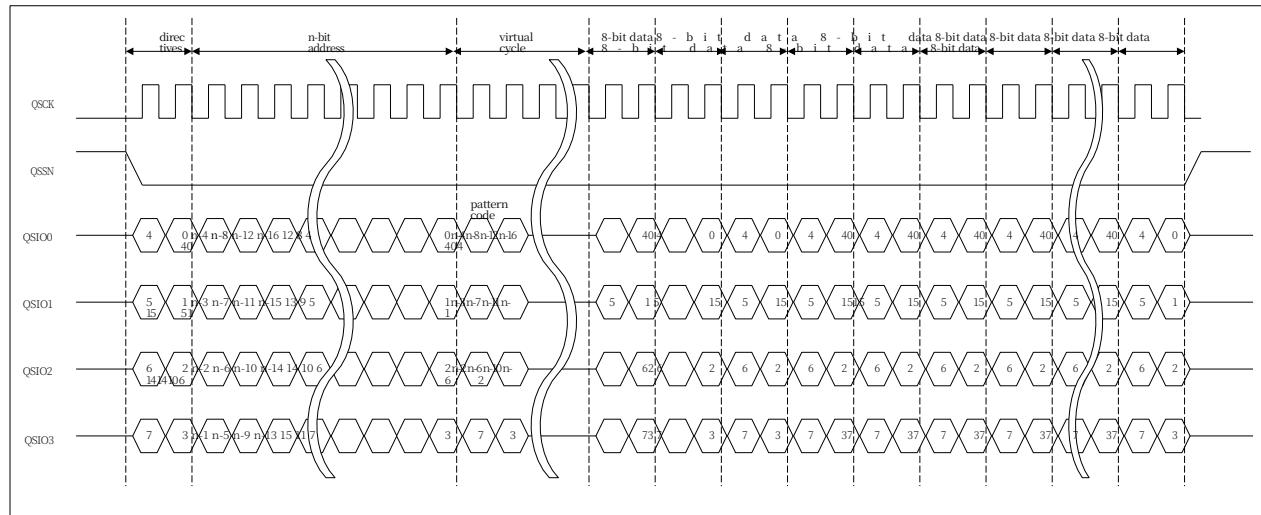


Figure 28-7 4-Wire SPI Protocol Action Schematic (Fast Read Mode)

28.3.2 SPI mode

There are two SPI modes, mode 0 and mode 3, which can be switched by setting the SPIMD3 bit in the QSCR register. The difference between SPI mode 0 and mode 3 is the different level of QSCK in standby state. The difference between SPI mode 0 and mode 3 is that the standby level of QSCK is different in standby state. In SPI mode 0, the standby level of QSCK is low, while the standby level is high in mode 3.

Serial data is output from QSPI on the falling edge of the serial clock and read into the external flash memory on the rising edge. The external flash memory, in turn, outputs serial data on the falling edge of the serial clock and is read in by QSPI on the falling edge of the next clock.

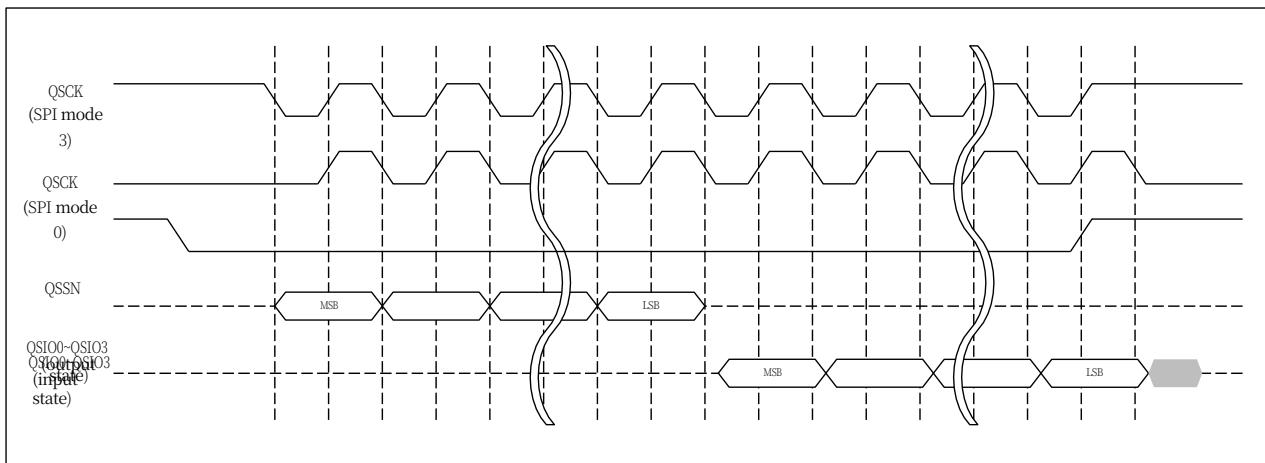


Figure 28-8 Basic Timing Diagram for Serial Interface

28.4 Timing Adjustment for QSPI Bus

The QSPI bus signals can be fine-tuned through registers for better timing for various QSPI bus accesses, either in ROM access mode or direct communication mode.

28.4.1 QSPI Bus Reference Clock

The reference clock of the QSPI bus is obtained by dividing HCLK. By setting the DIV[5:0] bits in the QSCR register, you can select from 2-division to 64-division clock sources of HCLK as the reference clock of the QSPI bus.

Table 28-3 QSPI Bus Reference Clock Selection List

DIV[5:0]	crossover ratio	Actual frequency of movement	DIV[5:0]	crossover ratio	Actual frequency of movement
		(HCLK=200MHz)			(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
6'b011010	27	7.41	6'b111010	59	3.39

DIV[5:0]	crossover r ratio	Actual frequency of movement	DIV[5:0]	crossover r ratio	Actual frequency of movement
		(HCLK=200MHz)			(HCLK=200MHz)
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

28.4.2 SPI Bus Reference Clock

When even multiple divisions of HCLK are selected for the reference clock, the high and low level times of the QSCK signal are the same, if odd multiple divisions are selected, then the high level time of the QSCK signal will be one more HCLK cycle than the low level.

If you want the QSCK to output a clock signal with a duty cycle of about 50% even when you select an odd number of divisions, you can set the DUTY bit in the QSFCR register to 1. By doing so, the rising edge of the QSCK signal will be output half an HCLK cycle later than before the adjustment, and the falling edge output time will remain unchanged. This results in a QSCK signal with a 50% duty cycle. Set the DUTY bit to 0 when the reference clock selects an even multiple of the HCLK frequency division.

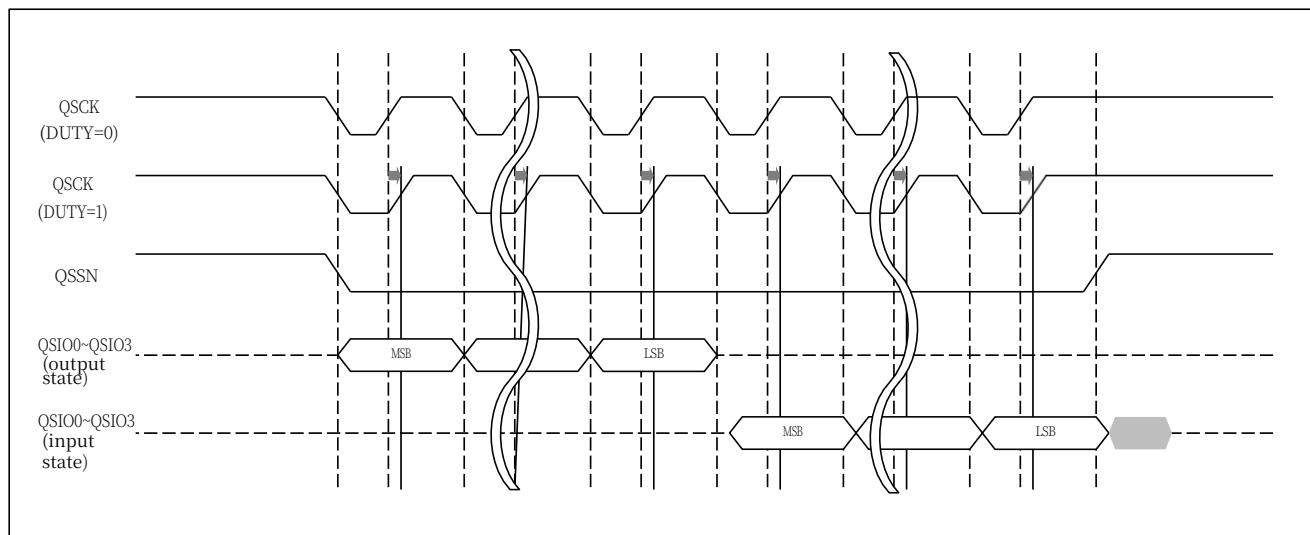


Figure 28-9 Output Clock Duty Cycle Compensation Schematic for HCLK Triple Frequency Selection of Base Clock

28.4.3 QSSN Signal Minimum High Level Width

To meet the cancellation time requirements required by the serial flash memory, the QSSN signal must remain high (i.e., idle) a long period of time between two adjacent QSPI bus cycles. The minimum high level width of QSSN can be selected by setting the SSHW[3:0] bits in the QSCSCR register from 1 to 16 QSPI base clock cycles.

28.4.4 QSSN establishment time

The time between the QSSN signal going low (i.e., becoming active) and the first rising edge of the QSCK signal is known as the QSSN setup time, which can be configured through register settings to meet the requirements of an external serial flash memory. Setting the SSNLD bit of register QSFCR selects whether the QSSN setup time is 0.5 or 1.5 QSPI reference clock cycles. This setting can also be used to configure the build-up time of the data pin from the data output license to the first rising edge of the QSCK signal output, which can be utilized as needed.

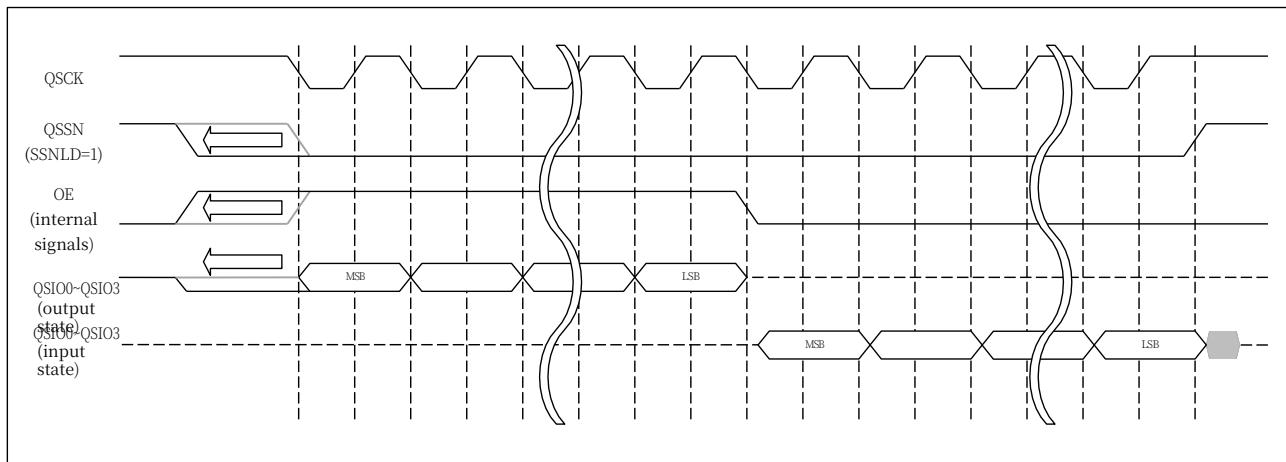


Figure 28-10 QSSL Establishment Time Configuration Schematic

28.4.5 Holding time for QSSN

The time between the last rising edge of the QSCK signal and the start of the QSSN signal going high (i.e., becoming idle) is called the QSSN hold time, which can be configured through register settings to meet the requirements of the external device. Setting the SSNHD bit of register QSFCR selects whether the QSSN hold time is 0.5 or 1.5 QSPI reference clock cycles.

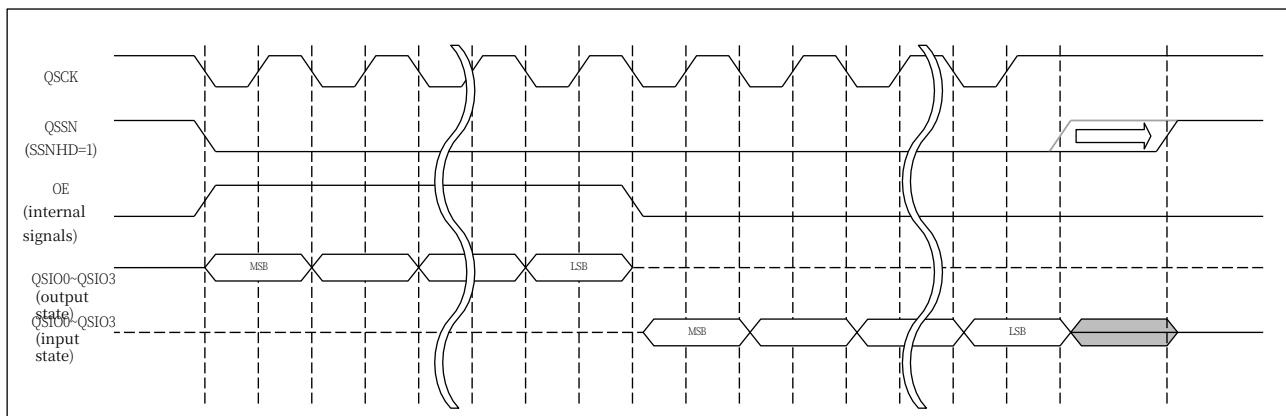


Figure 28-11 QSSN Hold Time Configuration Schematic

28.4.6 Serial data reception delay

The data from the serial flash is output synchronously with the falling edge of QSCK, and the QSPI receives the data on the next falling edge of QSCK. The time between when the serial flash memory starts to output data and when that data is received by the QSPI is called the receive delay, and the QSPI adds a delay adjustment cycle before the first data receive cycle. From the serial flash's perspective, this cycle can be viewed as an increase in the counterparty's data receive cycle. This delay adjustment cycle is only generated during the data receive action.

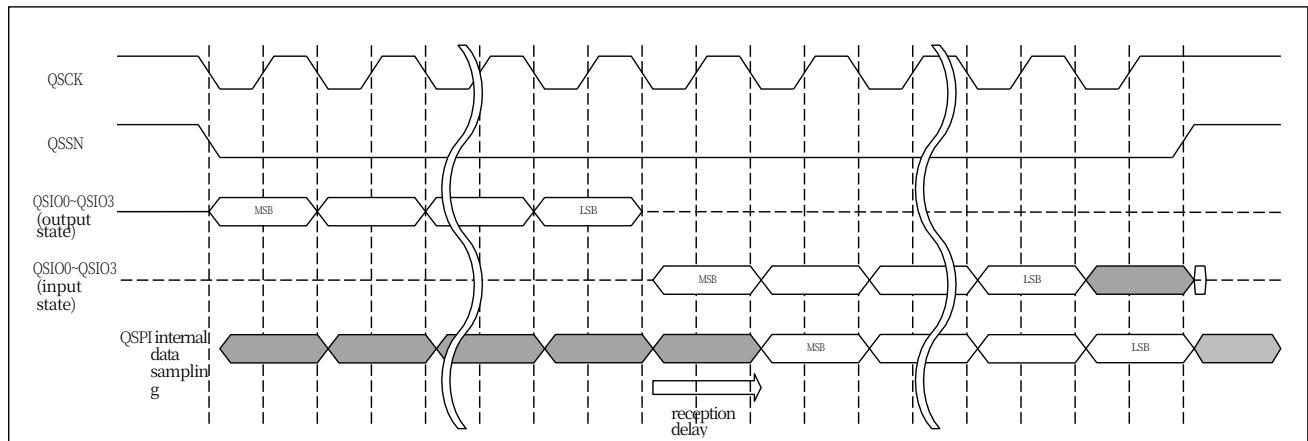


Figure 28-12 Data Receiving Delay Diagram

28.5 Introduction to SPI Instructions for ROM Access

28.5.1 Existing QSPI-ROM Command Reference

Table 28-4 List of Reference Instructions

model name	command code	clarification	
4-Byte Instruction Mode	standard reading	8'h13	
	fast read	8'h0C	
	Two-wire output for fast reading	8'h3C	
	2-wire input/output quick read	8'hbC	
	Four-wire output for fast reading	8'h6C	
	Four-wire inputs and outputs for fast reading	8'hEC	
	Exit 4-byte command mode	8'HB7	
3-Byte Instruction Mode	standard reading	8'h03	
	standard reading	8'h0B	When an 8-bit address is selected and A8 = 1
	fast read	8'h0B	-
	Two-wire output for fast reading	8'h3B	-
	2-Wire Input/Output Quick Read	8'hBB	-
	Four-wire output for fast reading	8'h6B	-
	Four-wire inputs and outputs for fast reading	8'hEB	-
	Entering 4-byte command mode	8'hE9	-
-	write mode	8'h06	-

To access the serial flash memory, the instructions are set through the instruction register QSCCMD.

28.5.2 standard read command (computing)

The standard read instruction is a common read instruction supported by most serial flash memories. At the beginning of a serial bus cycle, the serial flash select signal is enabled, and QSPI outputs the instruction code (03h/13h)*1 for the instruction, followed by the destination address, the width of which can be set by the AWSL[1:0] bits in the QSFCR register. The data is then received. The instruction selected for the initial state of QSPI is the standard read instruction.

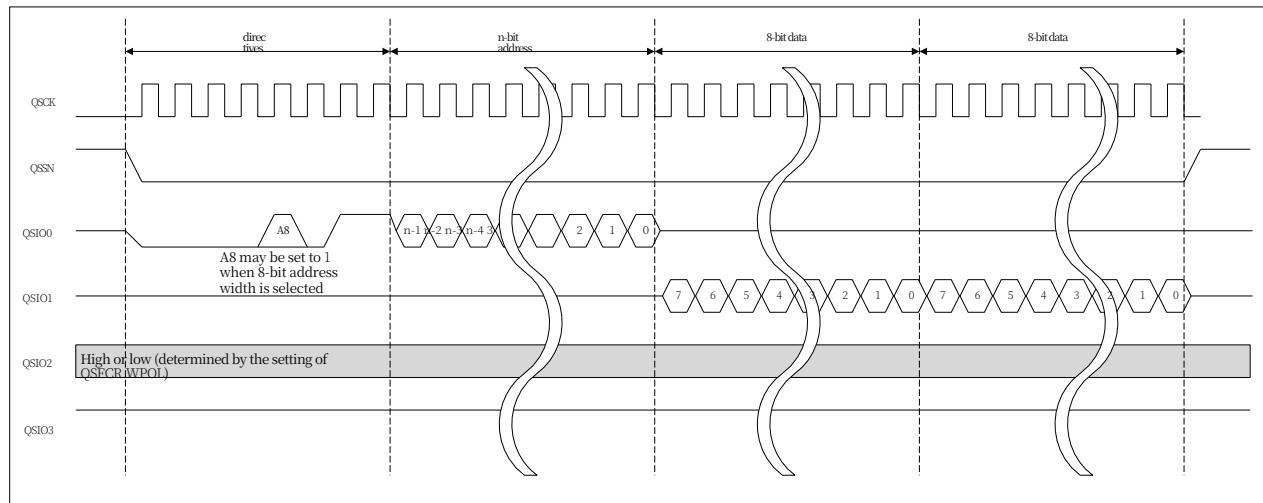


Figure 28-13 Standard Read Bus Cycle Schematic

28.5.3 Quick Read Command

The fast read instruction is a read instruction that supports a faster communication clock. At the beginning of a serial bus cycle, the serial flash select signal is made active, and immediately the QSPI outputs the instruction code (0Bh/0Ch) for the instruction, followed by the destination address, the width of which can be set by the AWSL[1:0] bits in the QSFCR register. The address output is followed by a certain number of virtual cycles, the exact number of which is determined by DMCYCN[3:0] in the QSFCR register. This is immediately followed by data reception.

The first two cycles of the virtual cycle are used to determine if XIP mode is selected. When XIP mode is selected, the instructions used for this transfer will be applied to the next SPI bus cycle, and the instruction transfer portion will be omitted on the next SPI bus cycle. For more details, please refer to

[XIP Control]

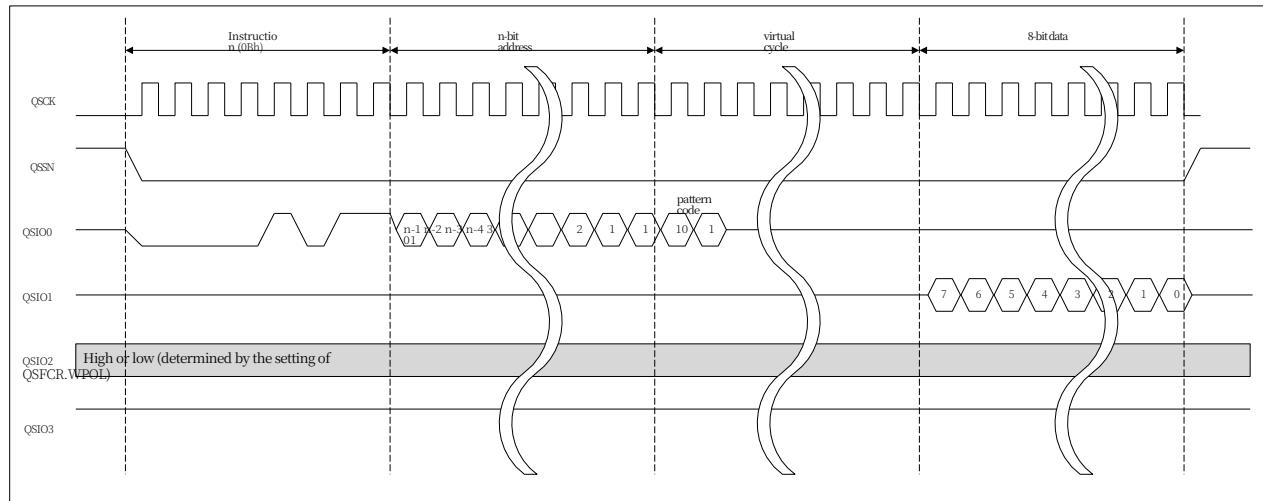


Figure 28-14 Fast Read Bus Cycle Schematic

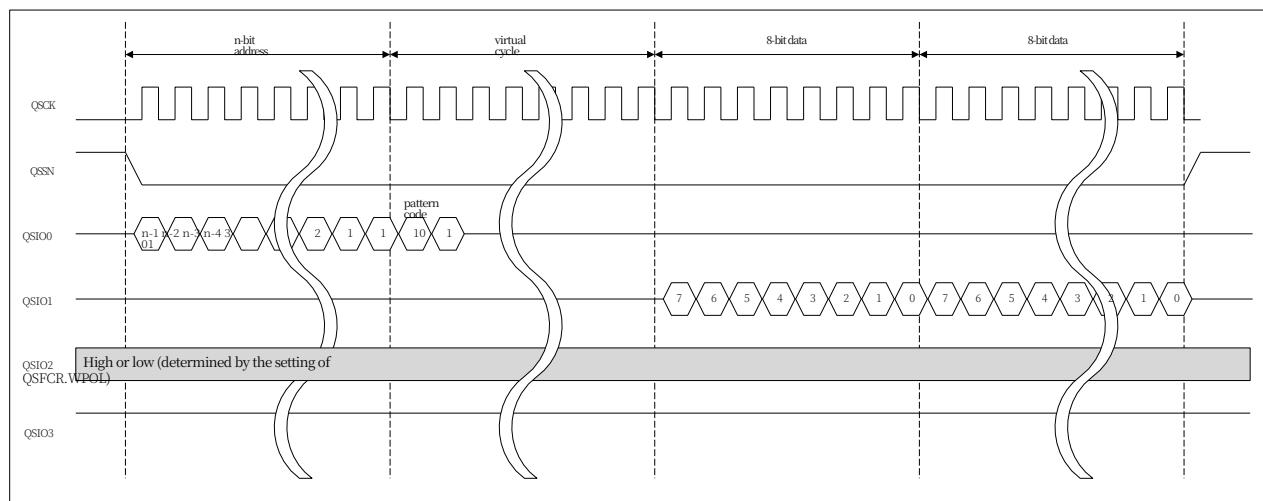


Figure 28-15 Schematic of Fast Read Bus Cycle with XIP Mode Selected

Attention:

-To use the fast read command make sure you use a serial flash memory that supports fast read. To use the fast read command make sure to use a serial flash memory that supports the fast read feature.

28.5.4 2-wire output for fast read commands

The 2-wire output fast read is a read instruction that uses two signal lines for data reception. At the beginning of a serial bus cycle, the serial flash select signal is made active and the **QSPI** starts to output the instruction code (3Bh/3Ch) and destination address of the instruction from the QSIO0 pin, the address width of which can be set by the AWSL[1:0] bits in the QSFCR register. This is followed by a certain number of virtual cycles, the exact number of which is determined by DMCYCN[3:0] in the QSFCR register. Data reception then begins via the QSIO0 and QSIO1 pins. Even bit data is received at QSIO0 and odd bit at QSIO1.

The first two cycles of the virtual cycle are used to determine if XIP mode is selected. When XIP mode is selected, the instructions used for this transfer will be applied to the next QSPI bus cycle, and the instruction transfer portion will be omitted during the next QSPI bus cycle. For details, refer to [XIP Control]

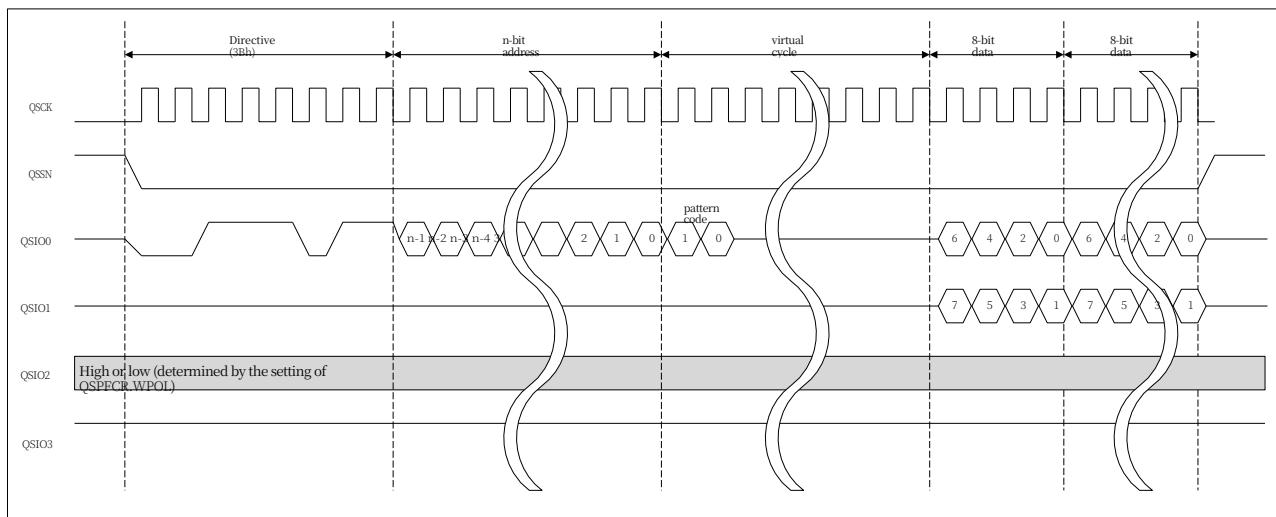
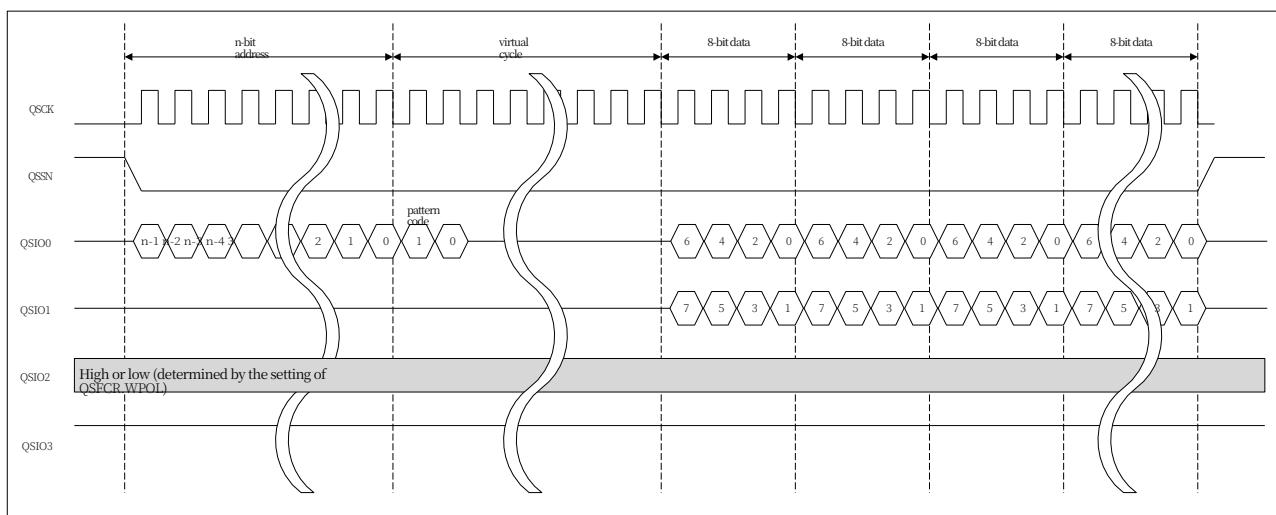


Figure 28-16 2-Wire Output Fast Read Bus Cycle Schematic



ion:

Attent

- To use the 2-wire output fast read command make sure to use a serial flash memory that supports this feature.

28.5.5 Two-wire inputs and outputs for fast read commands

The 2-wire input/output fast read is a read instruction that uses two signal lines for address sending and data receiving. At the beginning of a serial bus cycle, the serial flash select signal is made active and QSPI begins to output the instruction code for this instruction from the QSIO0 pin.

(BBh/BCh) After this QSPI outputs the destination address from the QSIO0 and QSIO1 pins, the width of the address can be set by the AWSL[1:0] bits in the QSFCR register. This is followed by a certain number of virtual cycles, the exact number of which is determined by DMCYCN[3:0] in the QSFCR register. Then data reception starts via the QSIO0 and QSIO1 pins. The QSIO0 pin is used for the transmission of the address and virtual cycles (including XIP mode selection information) and data reception for the even numbered bits, and the QSIO1 pin is used for the odd numbered bits.

The first two cycles of the virtual cycle are used to determine if XIP mode is selected. When XIP mode is selected, the instructions used for this transfer will be applied to the next QSPI bus cycle, and the instruction transfer portion will be omitted during the next QSPI bus cycle. For details, refer to [XIP Control].

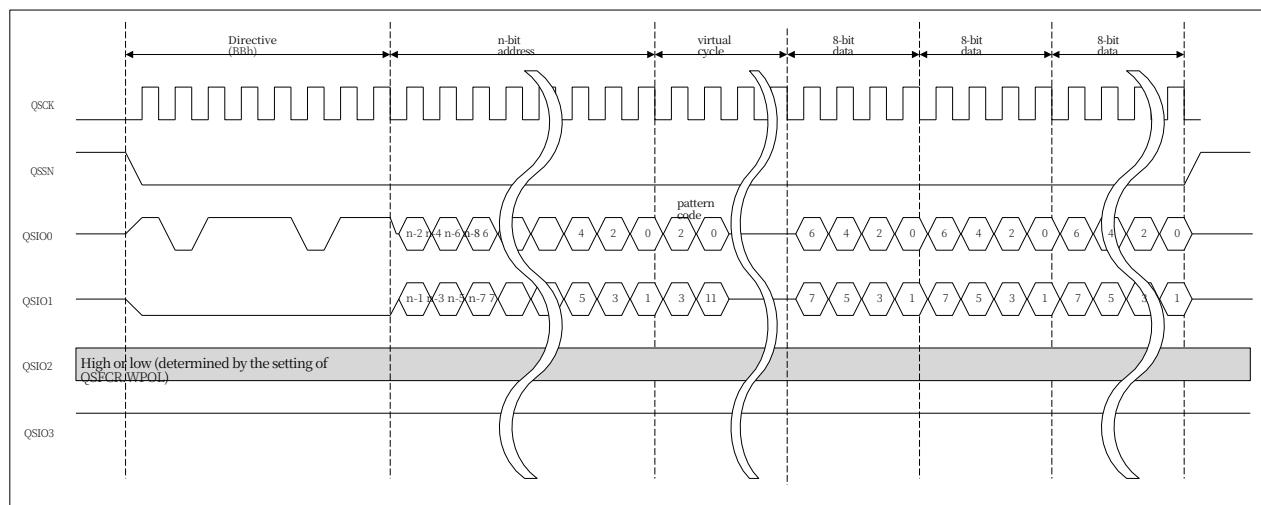


Figure 28-18 2-Wire Input/Output Fast Read Bus Cycle Schematic

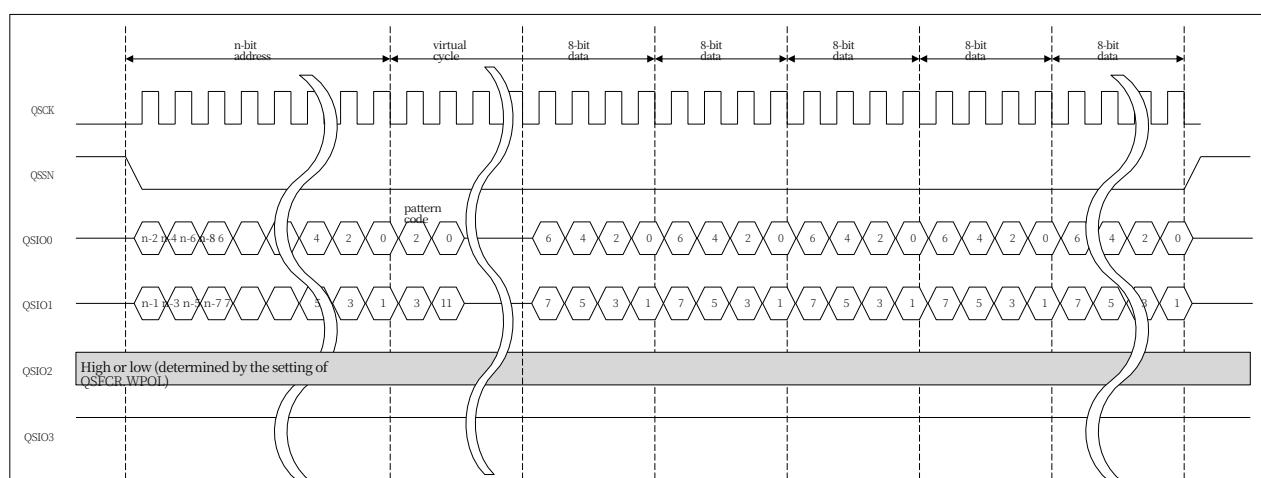


Figure 28-19 Schematic of 2-Wire Input-Output Fast Read Bus Cycle with XIP Mode Selected

Attention:

To use 2-wire inputs and outputs for fast read commands

To use the 2-wire input/output fast read command make sure to use a serial flash memory that supports this feature.

28.5.6 Four-wire output for fast command reading

The four-wire output fast read is a read instruction that uses four signal lines for data reception. At the beginning of a serial bus cycle, the serial flash select signal is made active and **QSPI** starts to output the instruction code (6Bh/6Ch) and destination address of the instruction from the QSIO0 pin, the address width can be set by the AWSL[1:0] bits in the QSFCR register. This is followed by a certain number of virtual cycles, the exact number of which is determined by DMCYCN[3:0] in the QSFCR register. Then data reception starts via the four pins QSIO0, QSIO1, QSIO2 and QSIO3.

The first two cycles of the virtual cycle are used to determine if XIP mode is selected. When XIP mode is selected, the instructions used for this transfer will be applied to the next SPI bus cycle, and the instruction transfer portion will be omitted on the next SPI bus cycle. For more details, please refer to

[XIP Control]

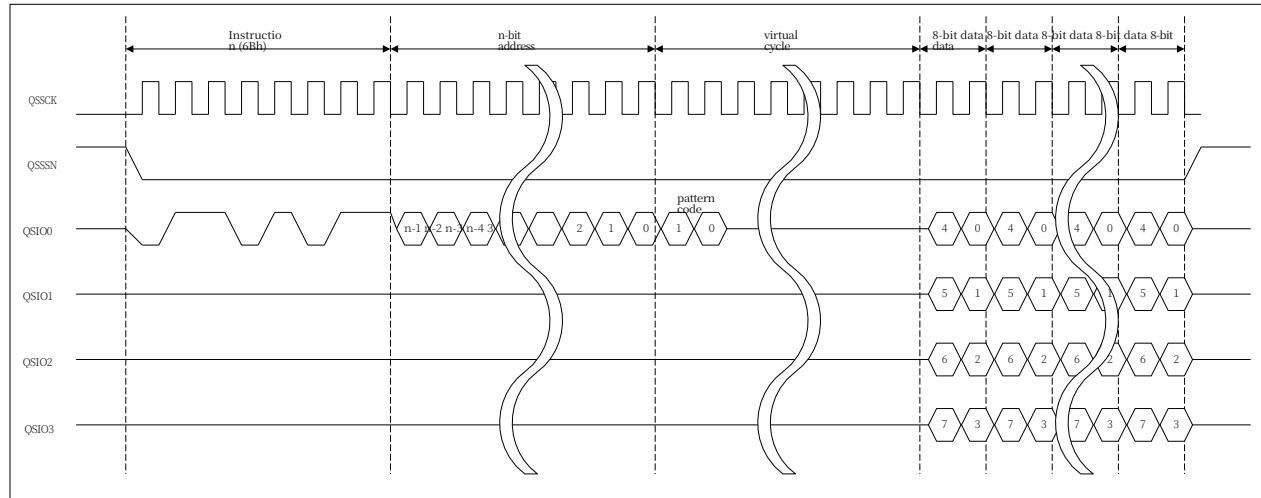


Figure 28-20 4-Wire Output Fast Read Bus Cycle Schematic

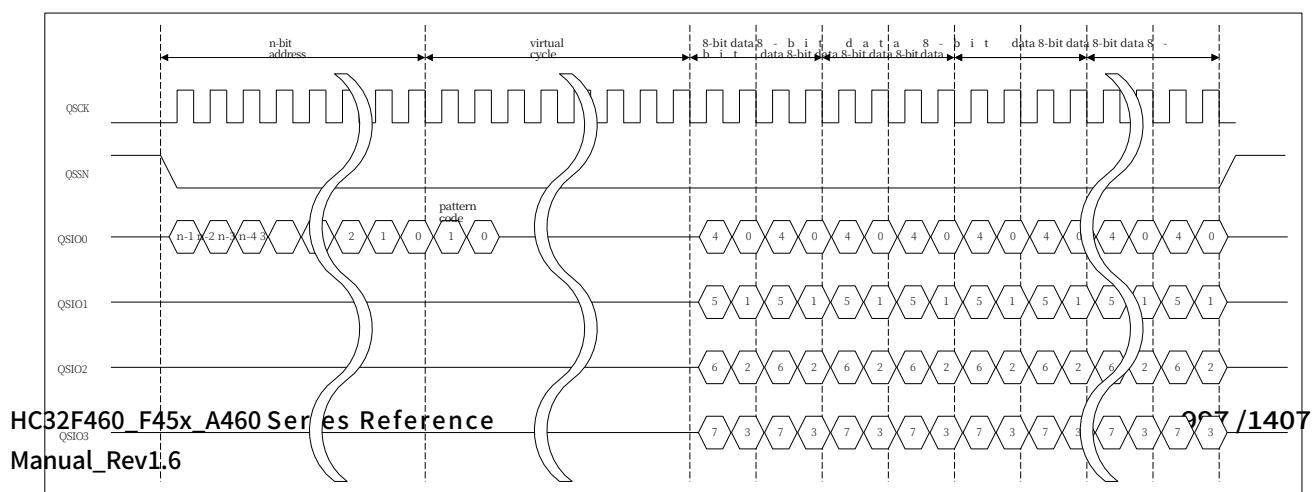


Figure 28-21 Schematic of Fast Read Bus Cycle for Four-Wire Outputs with XIP Mode Selected

Attention:

To use the four-wire output for fast read commands To use the four-wire output fast read command be sure to use a serial flash memory that supports this feature.

28.5.7 Four-wire inputs and outputs for fast command reading

The four-wire input/output fast read is a read instruction that uses four signal lines for address sending and data receiving. At the beginning of a serial bus cycle, the serial flash select signal is made active and QSPI begins to output the instruction code for this instruction from the QSIO0 pin

(EBh/ECh) After this QSPI outputs the destination address from the four pins QSIO0, QSIO1, QSIO2 and QSIO3, the width of the address can be set by the AWSL[1:0] bits in the QSFCR register. This is followed by a certain number of virtual cycles, the exact number of which is determined by DMCYCN[3:0] in the QSFCR register. Then data reception starts through the four pins QSIO0, QSIO1, QSIO2 and QSIO3.

The first two cycles of the virtual cycle are used to determine if XIP mode is selected. When XIP mode is selected, the instructions used for this transfer will be applied to the next QSPI bus cycle, and the instruction transfer portion will be omitted during the next QSPI bus cycle. For details, refer to [XIP Control]

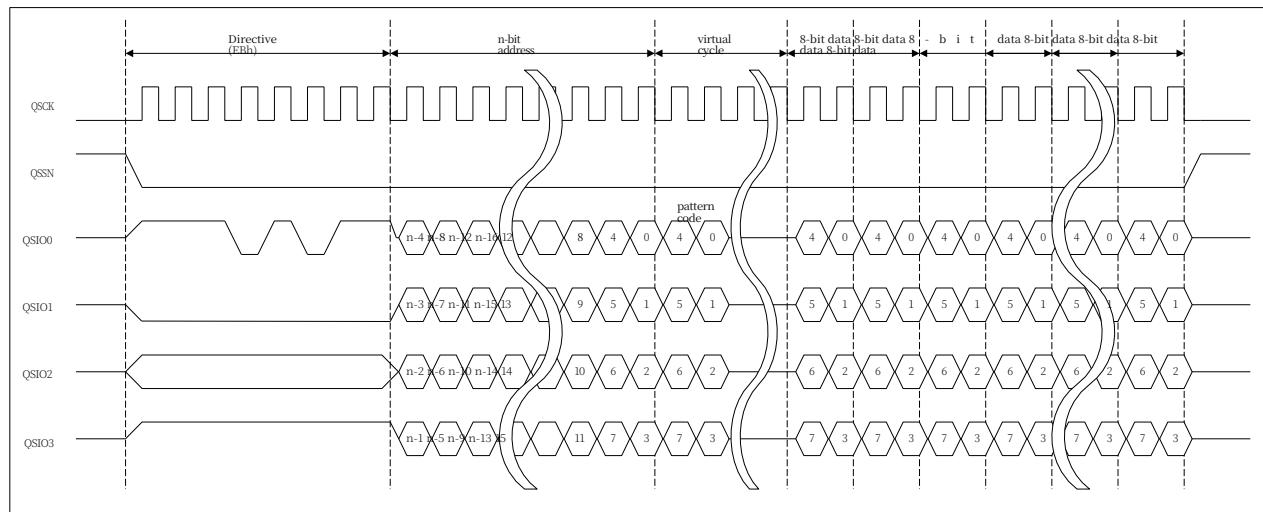


Figure 28-22 4-Wire Input/Output Fast Read Bus Cycle Schematic

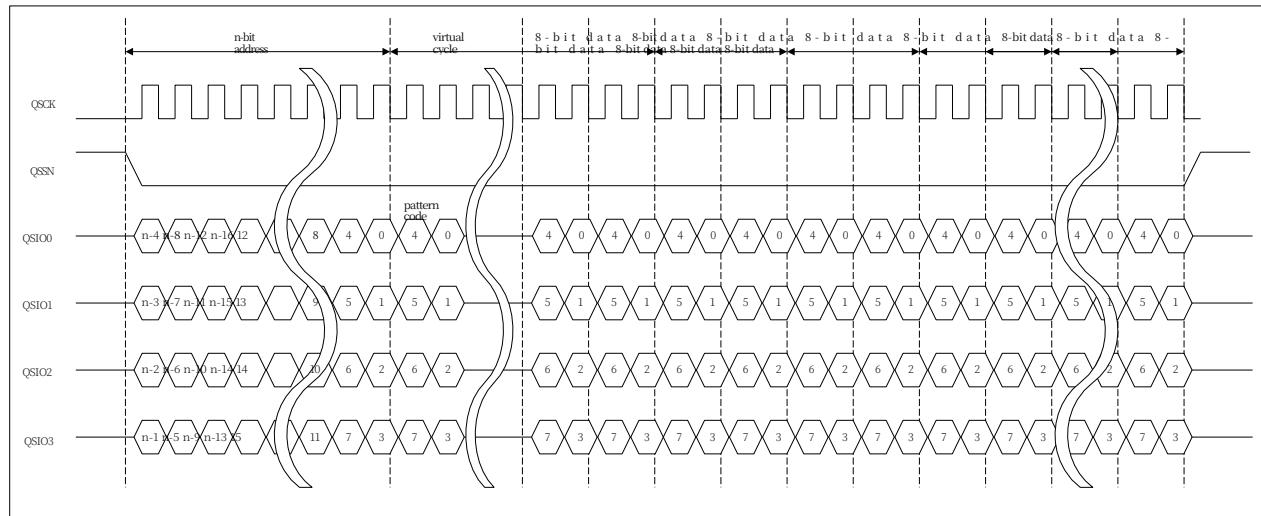


Figure 28-23 Schematic of Four-Wire Input/Output Fast Read Bus Cycle with XIP Mode Selected

Attention:

To use four-wire inputs and outputs for fast read commands To use the four-wire input/output fast read command make sure you use a serial flash memory that supports this feature.

28.5.8 Enter 4-Byte Mode Command

The Enter 4-Byte Mode instruction sets the address width of the serial flash memory to 4 bytes. At the beginning of a serial bus cycle, the serial flash select signal is made active and QSPI starts to output the instruction code (B7h) for this instruction from the QSIO0 pin.

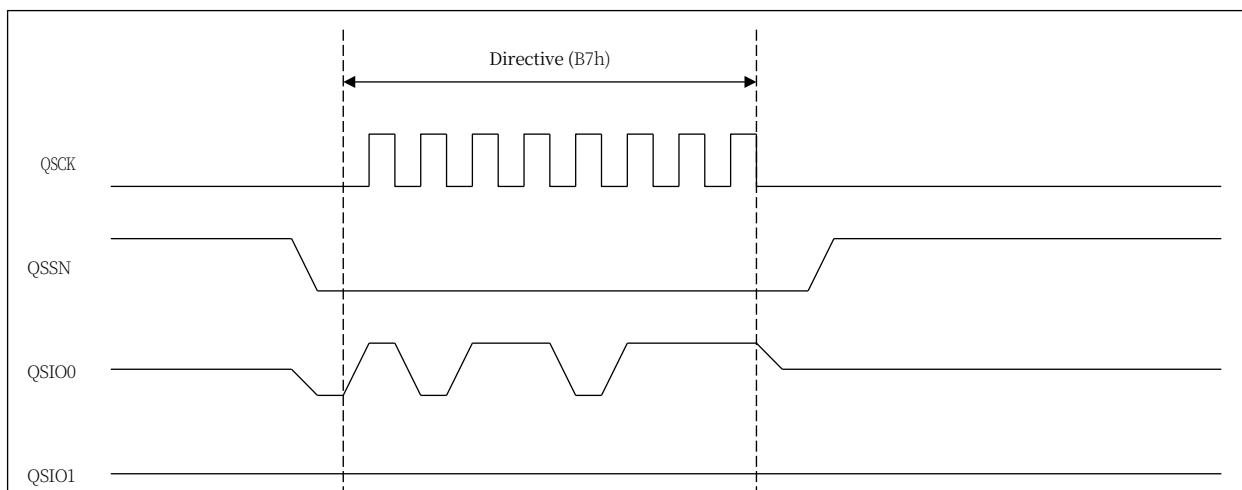


Figure 28-24 4-Byte Mode Instruction Bus Cycle Schematic

Attention:

ion:

The command can be issued regardless of whether the serial flash is in 4-Byte or 3-Byte mode. This command can be issued regardless of whether the serial flash is in 4-Byte or 3-Byte mode.

28.5.9 Exit 4-Byte Mode Command

The Exit 4-Byte Mode instruction sets the address width of the serial flash memory to 3 bytes. At the beginning of a serial bus cycle, the serial flash select signal is made active and QSPI starts to output the instruction code (E9h) for this instruction from the QSIO0 pin.

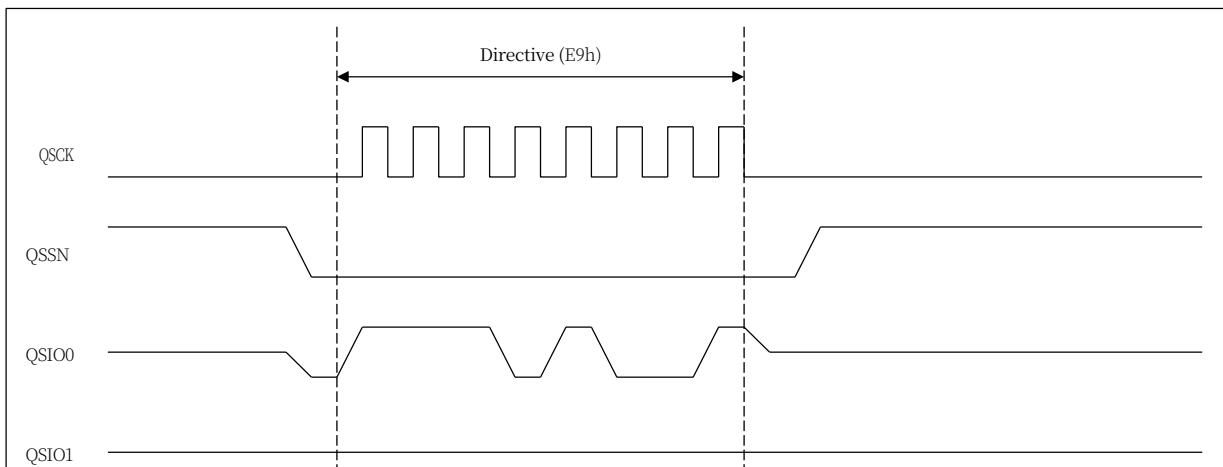


Figure 28-25 Exit 4-Byte Mode Instruction Bus Cycle Diagram

Attention:

ion:

The command can be issued regardless of whether the serial flash is in 4-Byte or 3-Byte mode. This command can be issued regardless of whether the serial flash is in 4-Byte or 3-Byte mode.

28.5.10 Write Permission Command

The Write Permit instruction allows the address width of the serial flash memory to be changed. At the beginning of a serial bus cycle, the serial flash select signal is made active and QSPI begins to output the instruction code (06h) for this instruction from the QSIO0 pin.

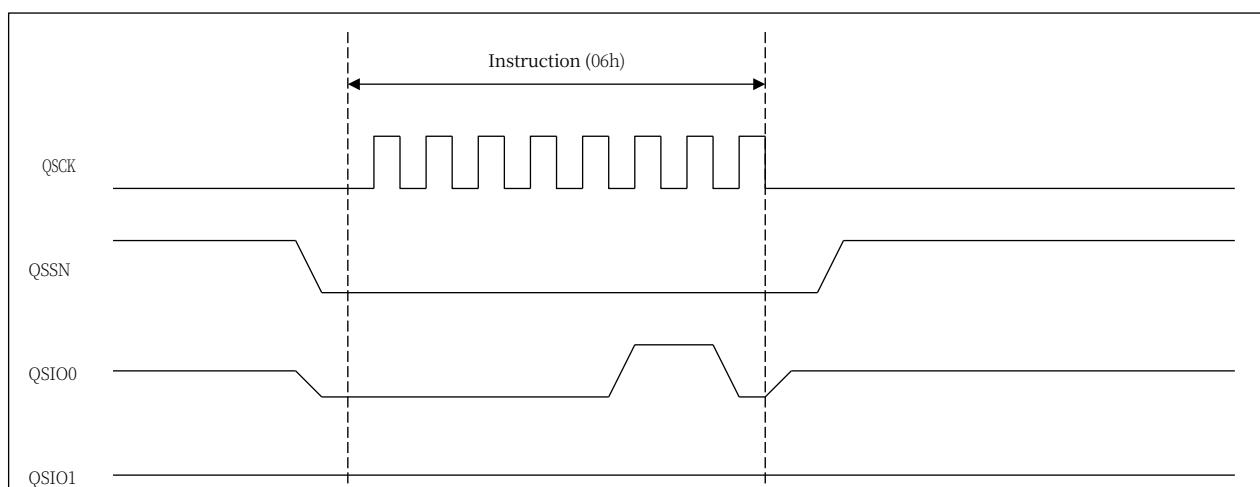


Figure 28-26 Write Permit Instruction Bus Cycle Diagram

28.6 QSPI Bus Cycle Arrangement

28.6.1 Single flash read for independent conversion

A single read command for a ROM is independently converted from an internal bus cycle to a QSPI bus cycle on a one-to-one basis. When a ROM read bus cycle is detected, the QSSN signal becomes active to start a QSPI bus cycle. After receiving data from the serial flash memory, the QSSN signal becomes invalid and the QSPI bus cycle is declared complete.

When another ROM read bus cycle is detected, the QSSN signal becomes active again after ensuring that the invalid hold time has exceeded the minimum invalid hold width, and a new QSPI bus cycle begins.

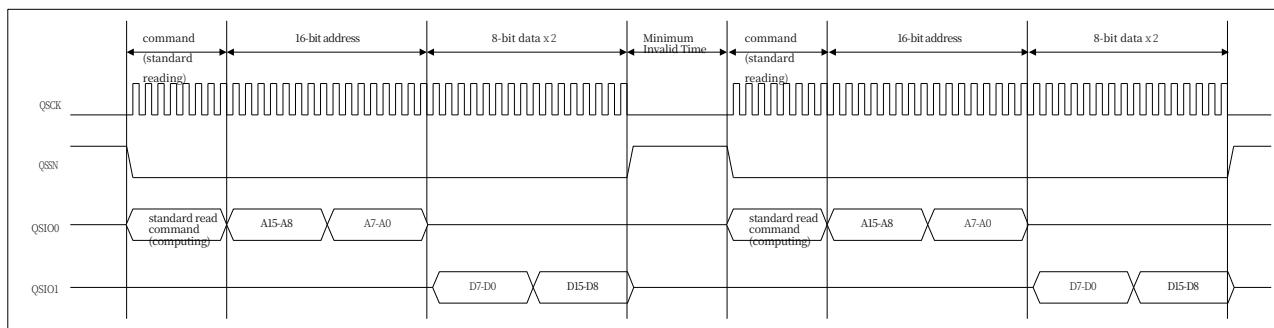


Figure 28-27 Individually Converted Single Flash Data Read Operation Schematic

28.6.2 Flash memory reading using the pre-read function

For transfers such as CPU instructions or blocks of data, the system typically reads the data in a sequence of sequentially increasing flash addresses. Serial flash memory has the ability to transfer data continuously without the need to resend instruction codes and addresses.

However, if the internal bus cycles issued by the MCU are converted independently, the QSPI bus cycles are also divided into separate entities, making it impossible to effectively utilize the advantages of serial flash memory for continuous data transfers. QSPI provides a pre-read function for continuous data reception.

The pre-read function is activated by setting the PFE bit in the QSCR register to one. When this function is active, 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, QSPI will match the access address. If the match is successful, the data in the corresponding buffer will be passed to the MCU, if the match fails, the data in the buffer will be discarded and a new QSPI bus cycle will be issued.

The pre-read buffer can store up to 16 bytes of data, in addition, there are 2 bytes of data receiving buffer can also store the pre-read data, when all the buffer data is full, the QSPI bus cycle ends. When all the buffers are full, the QSPI bus cycle ends. When the buffers are read and new buffer space is created, QSPI will automatically start a new QSPI bus cycle to resume the pre-reading action.

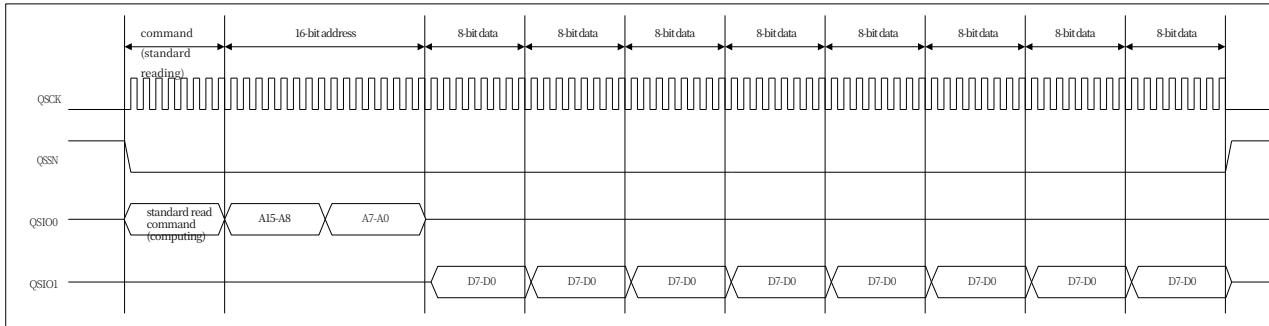


Figure 28-28 Data Reading Operation Diagram when the Pre-reading Function is Effective

28.6.3 Terminatio n of pre-reading

If a ROM read bus cycle to another address in the serial flash memory occurs during a pre-read transfer, the original pre-read action will be terminated and a new QSPI bus cycle will begin. Normally, the pre-read action is terminated after the current byte transfer is complete, but if the PFSAE bit in the QSCR register is set to 1, QSPI stops the pre-read action instantly instead of waiting until the current byte transfer is complete. The use of this feature requires that the serial flash device supports the instant stop action feature.

28.6.4 Pre-reading status monitoring

Reading from a low-speed serial flash memory increases the load on the system because the internal bus needs to remain in a wait state until the incoming **QSPI** bus cycle is complete, and QSPI provides pre-read status monitoring to reduce this load.

In the pre-read status register QSSR, the PFAN bit shows the current pre-read 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. These status bits make it easy to determine the current pre-read status with a single CPU instruction.

Attention:

- When executing a program that monitors the status of a pre-read, please place the program code outside the serial flash area of the object or enable the instruction cache. Otherwise, the pre-read object will frequently switch between the object data area and the instruction area, losing the meaning of pre-reading, and the monitoring program will enter an infinite loop because the pre-read can never be completed.

28.6.5 Flash Reads with QSPI Bus Cycle Extension

If SSNW[1:0] in the QSCSCR register is set to a value other than 00, the QSPI bus cycle will be in a hold state after receiving data and waiting for the next data read. The QSSN signal will be active low and QSCK will be stopped. If the next flash read instruction comes, QSPI will restart QSCK to accept data directly if the address of the read target is incremented immediately after the current address.

If the address is not sequential, the QSSN signal is invalidated by going high to end the previously held QSPI bus cycle and start a new SPI bus cycle.

This feature prevents the system from repeatedly sending command codes and addresses during discontinuous reads of successive incremental addresses, thereby improving read efficiency.

The extension time of the QSPI bus cycle can be set by SSNW[1:0], when the set time is exceeded and the next read has not occurred, QSSN will automatically be invalidated by going high, ending the QSPI bus cycle. If SSNW[1:0] is set to 11, QSSN will be extended indefinitely and the QSPI bus cycle will always be in the hold state, but this will increase the power consumption of the serial flash memory.

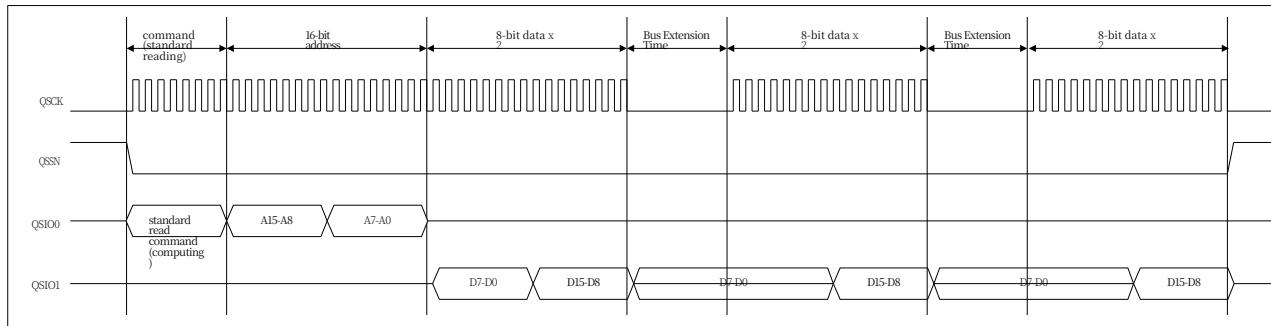
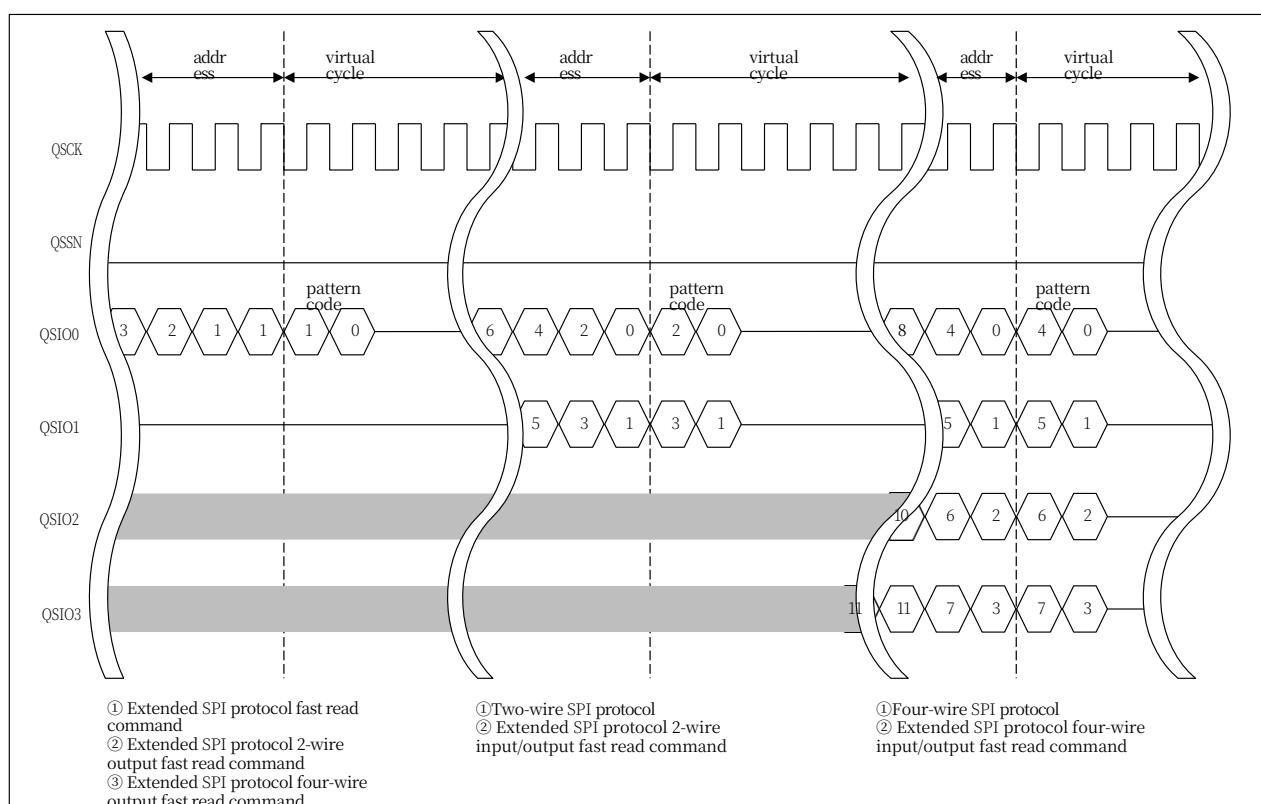


Figure 28-29 Data Read Operation Diagram Using QSPI Bus Cycle Extension Function

28.7 XIP Control

Some serial flash devices can reduce latency by omitting the receive read instruction. This function can be selected by a mode code sent during a virtual cycle.

During virtual cycles at fast instructions, QSPI controls the XIP mode of the serial flash memory by sending the XIP mode code during the first two cycles. The XIP mode code varies for different serial flash memories, and can be customized by using the XIPMC[7:0] bits in register QSXCMD. For details, refer to Figure 28-30 below.



28.7.1 XIP Mode Setting

If the XIP mode code corresponding to the serial flash memory is written to XIPMC[7:0] of register QSXCMD and the XIPE bit of register QSCR is set to 1, when the next fast read instruction occurs, the set mode code will be transmitted to the object serial flash memory during the first two cycles of the virtual cycle, and the serial flash memory and its control section will start the XIP mode when the reception of the mode code is completed. XIP mode. You can verify that the XIP mode has been entered by accessing the XIPF bit of the QSSR.

Attention:

- To start the XIP mode of the serial flash memory, you need to set the corresponding mode code in QSXCMD[7:0], and the XIP mode of the control part only needs to set the XIPE position to 1, which is independent of the value of QSXCMD[7:0].

28.7.2 XIP Mode Exit

If the code corresponding to the exit XIP mode of the serial flash memory is written to XIPMC[7:0] of register QSXCMD and the XIPE bit of register QSCR is set to 0, when the next fast read instruction occurs, the set exit mode code is transmitted to the object serial flash memory during the first two cycles of the virtual cycle, and after the reception of the exit mode code is completed, the XIP mode of the serial flash memory and its control section are terminated in XIP mode. You can verify that the XIP mode has been exited by accessing the XIPF bit of the QSSR.

Attention:

- Exiting the XIP mode of the serial flash memory requires setting the corresponding exit mode code in QSXCMD[7:0], while the XIP mode of the control section only requires clearing the XIPE bit to zero, independent of the value of QSXCMD[7:0].

28.8 QSIO2 and QSIO3 Pin Status

The QSIO2 and QSIO3 pin states depend on the serial read mode set by the MDSEL[2:0] bits in the QSCR register.

Table 28-5 QIO2 and QIO3 Pin Status

QSCR register MDSEL[2:0] bits	QSIO2 Status	QSIO3 status	note
000	Output state, the output level is determined by the WPOL bit of the QSFCR register and the initial output is low	Output High Level	Standard read (initial state)
001			fast read
010			Two-wire output for fast reading
011			2-wire input/output quick read
100	As a third data cable for input or output	As a fourth data cable for input or output	Four-wire output for fast reading
101	Out of action, standby for Hi-Z	Out of action, standby for Hi-Z	Four-wire inputs and outputs for fast reading
110	Refer to specific protocol settings for each phase	Refer to specific protocol settings for each phase	Customized protocol standard reads
111			Custom Protocol Quick Read

Attention:

- The QSIO2 pin can also be used as a WP# function for serial flash memory.
- The QSIO3 pin can also be used as a serial flash HOLD# or RESET# function.

28.9 direct communication mode

28.9.1 About the direct communication model

QSPI can read serial flash memory by automatically converting the MCU's external ROM read bus cycles to QSPI bus cycles. However, there are a number of different additional functions that can be performed on the serial flash memory, such as reading ID information, erasing, writing, and reading status information. There is no standard set of commands to set up these functions, and as new functions are rapidly added to the serial flash memory, it becomes more and more difficult to correspond to them at the hardware level.

For this situation, QSPI provides a direct communication mode that allows the user to control the serial flash memory directly through software. This mode allows the software to generate any desired number of QSPI bus cycles.

28.9.2 Direct communication mode setting

Setting the DCOME bit of the QSCR register to 1 allows you to enter direct communication mode. Once the direct communication mode is entered, the usual flash read operation will not be possible, and if you want to do the regular flash read, you need to clear the DCOME bit to exit the direct communication mode.

Attention:

- If in XIP mode, you need to exit XIP mode before starting direct communication mode.

28.9.3 QSPI Bus Cycle Generation in Direct Communication Mode

A complete QSPI bus cycle in direct communication mode begins with the first operation on DCOM[7:0] of register QSDCOM and ends with a write operation to the QSCR register. Multiple operations can be performed on DCOM[7:0] during this period. A write to DCOM[7:0] is converted to a single byte data transfer to the QSPI bus, while a read to DCOM[7:0] is converted to a single byte data reception from the QSPI bus.

The QSSN signal remains active low from the first operation of DCOM[7:0] of register QSDCOM until the last write operation of the QSCR register.

Direct communication mode does not support multi-line actions.

Attention:

- Write operations to registers other than QSCR and QSDCOM are not possible in direct communication mode. Write operations to other registers will exit the direct communication mode. Exiting in this way may lead to unforeseen situations and is not recommended.

28.10 disruptions

When a read access operation to the ROM is detected in direct communication mode, the RAER bit of the QSSR register is set to 1, at which point the QSPI will generate a bus hardware error interrupt. The interrupt request will be held until the RAER bit is cleared. For details, please refer to [Interrupt Controller (INTC)]

28.11 Precautions for use

28.11.1 QSPI register setting order

The QSPI control registers can be set or changed dynamically during system operation. However, inattention to the order in which the registers are set may cause the QSPI bus cycle to start before the registers are fully set, so configure the order of register settings carefully to avoid such a situation.

28.11.2 Module stop signal setting

QSPI is in module stop state after system reset, and the registers can be set only when the QSPI module stop signal in the module stop control register is cleared to zero. For details, please refer to [Methods to Reduce Power Consumption]

28.12 Register Description

Register base address: 0x9C00_0000

Table 28-6 QSPI Register List

register name	offset address	reset value
Control register QSCR	0x0000	0x003F_0000
Chip Select Control Register QSCSCR	0x0004	0x0000_000F
Format Control Register QSFCR	0x0008	0x0000_8033
Status register QSSR	0x000C	0x0000_8000
Direct communication instruction register QSDCOM	0x0010	-
Instruction code register QSCCMD	0x0014	0x0000_0000
XIP Mode Code Register QSXCMD	0x0018	0x0000_00FF
Flag clear register QSSR2 (write only)	0x0024	-
External address register QSEXAR	0x0804	0x0000_0000

28.12.1 QSPI Control Register (QSCR)

Reset value: 0x003F_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	DIV[5:1]			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	DPRSL[1:0]	APRSL[1:0]	IPRSL[1:0]	SPIMD3	XIPE	DCOME	PFSAE	PFE	MDSEL[2:0]					

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b21	Reserved	-	Reads "0", writes "0".	R/W
b20~b16	DIV[5:0]	Reference Clock Select Bit	Serial Interface Reference Clock Selection b5 b4 b3 b2 b1 b0 0 0 0 0 0 : 2 HCLK cycles 0 0 0 0 0 : 1:2 HCLK cycles* 0 0 0 0 1 : 0:3 HCLK cycles 0 0 0 0 1 : 1:4 HCLK cycles* 0 0 0 1 0 : 0:5 HCLK cycles 0 0 0 1 0 : 1:6 HCLK cycles* 0 0 0 1 1 : 0:7 HCLK cycles 0 0 0 1 1 : 1:8 HCLK cycles* 0 0 1 0 0 : 0:9 HCLK cycles 0 0 1 0 0 : 1:10 HCLK cycles* 0 0 1 0 1 : 0:11 HCLK cycles 0 0 1 0 1 : 1:12 HCLK cycles* 0 0 1 1 0 : 0:13 HCLK cycles 0 0 1 1 0 : 1:14 HCLK cycles* 0 0 1 1 1 : 0:15 HCLK cycles 0 0 1 1 1 : 1:16 HCLK cycles*	R/W

1 0 0 0 0 0:33 HCLK cycles
1 0 0 0 0 1:34 HCLK cycles*
1 0 0 0 1 0:35 HCLK cycles

			1 0 0 0 1 1:36 HCLK cycles*
			1 0 0 1 0 0:37 HCLK cycles
			1 0 0 1 0 1:38 HCLK cycles*
			1 0 0 1 1 0:39 HCLK cycles
			1 0 0 1 1 1:40 HCLK cycles*
			1 0 1 0 0 0:41 HCLK cycles
			1 0 1 0 0 1:42 HCLK cycles*
			1 0 1 0 1 0:43 HCLK cycles
			1 0 1 0 1 1:44 HCLK cycles*
			1 0 1 1 0 0:45 HCLK cycles
			1 0 1 1 0 1:46 HCLK cycles*
			1 0 1 1 1 0:47 HCLK cycles
			1 0 1 1 1 1:48 HCLK cycles*
			1 1 0 0 0 0:49 HCLK cycles
			1 1 0 0 0 1:50 HCLK cycles
			1 1 0 0 1 0:51 HCLK cycles
			1 1 0 0 1 1:52 HCLK cycles
			1 1 0 1 0 0:53 HCLK cycles
			1 1 0 1 0 1:54 HCLK cycles
			1 1 0 1 1 0:55 HCLK cycles
			1 1 0 1 1 1:56 HCLK cycles
			1 1 1 0 0 0:57 HCLK cycles
			1 1 1 0 0 1:58 HCLK cycles
			1 1 1 0 1 0:59 HCLK cycles
			1 1 1 0 1 1:60 HCLK cycles
			1 1 1 1 0 0:61 HCLK cycles
			1 1 1 1 0 1:62 HCLK cycles
			1 1 1 1 1 0:63 HCLK cycles
			1 1 1 1 1 1:64 HCLK cycles
b15~b14	Reserved	-Reserved	Read "O", write "O".
b13~b12	DPRSL[1:0]	Data reception stage SPI protocol selection	R/W data receiving stage SPI protocol selection. b1 b0 0 0: Extended SPI Protocol 0 1: 2-wire SPI 0 1: 2-wire SPI protocol 1 0: 4-wire SPI protocol 1 1: Setting prohibitions
b11~b10	APRSL[1:0]	Address sending phase SPI protocol selection	Address sending phase SPI protocol selection. b1 b0 0 0: Extended SPI Protocol 0 1: 2-wire SPI protocol 1 0: 4-wire SPI protocol 1 1: Setting prohibitions
b9~b8	IPRSL[1:0]	SPI protocol selection for command sending phase	Command sending phase SPI protocol selection. b1 b0 0 0: Extended SPI

R/W

R/W

			SPI mode	
b7	SPIMD3	SPI Mode Selection	selection 0: SPI mode 0 1: SPI mode 3	R/W
b6	XIPE	XIP mode license	0: XIP mode disabled 1: XIP mode license	R/W
b5	DCOME	Direct communication licenses	QSPI bus communication mode selection 0: ROM access mode 1: Direct communication mode	R/W
b4	PFSAE	Pre-read Instant Stop License	Select the location to reset the pre-read action 0: current pre-read action aborted at byte boundary 1: The current pre-reading action is aborted immediately.	R/W
b3	PFE	Pre-reading license	Pre-read function valid/invalid selection 0: Pre-reading function disabled 1: Pre-reading function is effective	R/W
b2~b0	MDSEL[2:0]	QSPI read mode selection	Serial interface read mode selection b2 b1 b0 0 0 0: Standard reading 0 0 1: Quick Read 0 1 0: 2-wire output fast read 0 1 1: 2-wire input/output quick-reading 1 0 0: 4-wire output fast read 1 0 1: Four-wire inputs and outputs for quick reading 1 1 0: Customized standard read 1 1 1: Customized Quick Reads	R/W

28.12.2 QSPI Chip Select Control Register (QSCSCR)

Reset value: 0x0000_000F

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
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b6	Reserved	-	Reads "0" and writes "0".	R/W
b5~b4 extension setting	SSNW[1:0]	QSSN valid time	QSPI bus access after QSSN valid time extension function selection b5 b4 0 0: No extension of QSSN validity time 0 1: Extend the QSSN effective time by 32 QSCK cycles 1 0: Extend the QSSN effective time by 128 QSCK cycles 1 1: Extend QSSN valid time	R/W
b3~b0	SSHW[3:0]	QSSN Minimum Invalid Time Setting	indefinitely QSSN signal minimum invalid time selection b3 b2 b1 b0 0 0 0 0: 1 QSCK cycle 0 0 0 1: 2 QSCK cycles 0 0 1 0: 3 QSCK cycles 0 0 1 1: 4 QSCK cycles 0 1 0 0: 5 QSCK cycles 0 1 0 1: 6 QSCK cycles 0 1 1 0: 7 QSCK cycles 1 0 0 0: 8 QSCK cycles 1 0 0 1: 9 QSCK cycles 1 0 1 0: 10 QSCK cycles 1 0 1 1: 11 QSCK cycles 1 1 0 0: 12 QSCK cycles 1 1 0 1: 13 QSCK cycles 1 1 1 0: 14 QSCK cycles 1 1 1 1: 15 QSCK cycles 1 1 1 1: 16 QSCK cycles	R/W

28.12.3 QSPI Format Control Register (QSFCR)

Reset value: 0x0000_8033

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
DUTY	-	-	-	DMCYCN [3:0]	-	WPOL	SSNLD	SSNHD	-	Four_BIC	AWSL[1:0]				

classifier for marking			celebrity	functionality	fill out or in (information on a form)
b31~b16	Reserved	-		Reads "0", writes "0".	R/W
b15	DUTY	Duty cycle		QSCK Output Waveform Duty Cycle Compensation	R/W
correction			0: No duty cycle correction 1: Lag the rising edge of QSCK by 0.5 HCLK cycle		
b14~b12	Reserved	-Reserved		(valid when the frequency selected for QSCK is an odd multiple of HCLK)	R/W
b11~b8	DMCYCN[3:0]	Virtual Cycle Setting		Read "0", write "0". Selection of the number of virtual cycles when using the fast read instruction b3 b2 b1 b0 0 0 0 0: 3 QSCK cycles *1 0 0 0 1: 4 QSCK cycles 0 0 1 0: 5 QSCK cycles 0 0 1 1: 6 QSCK cycles 0 1 0 0: 7 QSCK cycles 0 1 0 1: 8 QSCK cycles 0 1 1 0: 9 QSCK cycles 0 1 1 1: 10 QSCK cycles 1 0 0 0: 11 QSCK cycles 1 0 0 1: 12 QSCK cycles 1 0 1 0: 13 QSCK cycles 1 0 1 1: 14 QSCK cycles 1 1 0 0: 15 QSCK cycles 1 1 0 1: 16 QSCK cycles 1 1 1 0: 17 QSCK cycles 1 1 1 1: 18 QSCK cycles	R/W
b7	Reserved	-		Reads "0" and writes "0".	R/W
b6	WPOL	WP pin output level setting	0: low level	WP pin (QIO2) level setting	R/W
b5	SSNLD	QSSN signal output time delay setting	0: high level 1: low level	QSSN signal output timing selection	1: Release QSSN 0: Output QSSN 0.5 QSCK ahead of the first rising edge of QSCK 1.5 QSCK lags from the last rising edge of QSCK
b4	SSNHD	QSSN signal release time delay setting	0: release QSSN 0.5 QSCK lags from the last rising edge of QSCK	QSSN signal release timing selection	edge of QSCK edge of QSCK QSSN signal release timing selection edge of QSCK edge of QSCK

b3	Reserved	-Reserved	Read "0", write "0".	R/W
b2	Four_BIC	4-Byte Address Read Instruction Code Selection	Read instruction code selection when address width is 4 bytes 0: Read instruction code without 4-byte address 1: Read instruction code using a 4- byte address	R/W

Serial interface				
b1~b0	AWSL[1:0]	Address	address width	
Width Selection			selection b1 b0	
			0 0: 1 byte	R/W
			0 1: 2 bytes	
			1 0: 3 bytes	
			1 1:4 bytes	

*1: To avoid the conflict of QIO0 terminal when switching the input/output state, if the QSMD.QSOEX bit is set to 1 (processing the license signal is extended by 1 cycle) select 4 virtual cycles or more of QSPICK cycles.

28.12.4 QSPI Status Register (QSSR)

Reset value: 0x0000_8000

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
PFAN	PFFUL	-	PFNUM[4:0]				RAER	XIPF	-	-	-	-	-	-	BUSY

classifier for honorific people		marking	celebrity	functionality	fill out or in (information on a form)
b31~b16		Reserved	-	Reads "0", writes "0".	R/W
b15		PFAN	Pre-reading action status	Pre-reading of motion status signals 0: Pre-reading in action 1: Pre-reading is stopped	R
b14		PFFUL	Pre-read buffer status	Pre-read buffer status signal 0: Pre-read buffer with space remaining 1: Pre-read buffer data is full	R
b13		Reserved	-	Reads "0" and writes "0".	R/W
b12~b8		PFNUM[4:0]	Pre-read buffer is stored Number of data bytes	Display of the number of bytes of data already stored in the pre-read buffer b4 b3 b2 b1 b0 0 0 0 0 0: 0 bytes 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 The rest of the settings are invalid	R
b7		RAER*1	ROM access error flag	ROM access error flag bit occurs in direct communication mode 0: No ROM access detected to have occurred 1: ROM access detected to have occurred	R/W
b6		XIPF	XIP mode flag	XIP mode status signal 0: Non-XIP mode	R

1: XIP mode				
b5~b1	Reserved	-	Reads "0", writes "0".	R/W
b0	BUSY	Bus Busy sign	QSPI bus operating status flag bit in direct communication mode	R

0: bus idle, no serial transfer process

1: Bus busy, serial transfer process in progress

*1: RAER needs to be cleared by the RAERCLR bit of QSSR2.

28.12.5 QSPI Command Code Register (QSCCMD)

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
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	RIC [7:0]

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b8	Reserved	-	Reads "0", writes "0".	R/W
b7~b0	RIC[7:0]	Replacement Instruction Code	Serial Flash Instruction Codes for Replacing Default Instructions	R/W

28.12.6 QSPI Direct Communication Command Register (QSDCOM)

Reset value: none

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]

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b8	Reserved	-	Reads "0" and writes "0".	R/W
b7~b0	DCOM[7:0]	Direct Communication Mode Commands	The interface during direct communication mode communicates directly via the QSPI bus. A read or write access to this interface is translated into a corresponding QSPI bus cycle. The interface is only valid in direct communication mode, the interface is disabled in ROM access mode.	R/W

28.12.7 QSPI XIP Mode Code Register (QSXCMD)

Reset value: 0x0000_00FF

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]

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)

28.12.8 QSPI System Configuration Register (QSSR2)

Reset value: none

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	-	-	-	-	-	-	-

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b8	Reserved	-	Write "0".	W
b7	RAERCLR	RAER clearance	Write 1 to clear the RAER bit in the QSSR to zero	W
b6~b0	Reserved	-	Write "0".	W

28.12.9 QSPI External Extended Address Register (QSEXAR)

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EXADF[5:0]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

classifier for honorifi c people	mark ing	celebrity	function ality	fill out or in (information on on a form)
b31~b26	EXADR[5:0]	External Extended Address Generation QSPI external address high 6-bit setting, with QSPI's ROM access window address can be accessed maximum code R/W	Ask 64MB x 63 blocks of external ROM space.	
b25~b0	Reserved	-	Reads "0" and writes "0".	R/W

29 IC with built-in audio bus module (I2S)

29.1 summary

I2S (Inter_IC Sound Bus) Integrated Circuit Built-in Audio Bus, which specializes in data transfer between audio devices.

Table 29-1 I2S Main Characteristics

functionality	Main characteristics
communication method	<ul style="list-style-type: none">- Supports full-duplex and half-duplex communication- Supports master or slave mode operation
data format	<ul style="list-style-type: none">- Selectable channel length: 16/32 bits- Optional transmission data length: 16/24/32 bits- Data shift order: MSB start
baud	<ul style="list-style-type: none">- 8-bit programmable linear prescaler for accurate audio sampling frequency- Supports sampling frequencies of 192k, 96k, 48k, 44.1k, 32k, 22.05k, 16k, 8k- Drive clock can be output to drive external audio components at a fixed ratio of 256 x Fs (Fs is the audio sampling frequency).
Supports I2S protocol	<ul style="list-style-type: none">- I2S Philips Standard- MSB alignment standard- LSB alignment standard- PCM standard
data buffer	<ul style="list-style-type: none">- Input and output FIFO buffers with 2 word depth and 32 bit width.
clock source	<ul style="list-style-type: none">- Internal I2SCLK (UPLL/R/UPLLQ/UPLL/PMLLR/MPLLQ/MPLLP) can be used; also by the external clock on the I2S_EXCK pin provides the
disruptions	<ul style="list-style-type: none">- Generate an interrupt when the effective space in the transmit buffer reaches the alarm threshold.- Generate an interrupt when the effective space in the receive buffer reaches the alarm threshold.- Receive data area is full and there is still a write data request, receive overflow- Send data area is empty and there is still a request to send, send underflow- Send data area is full and there is still a request to write data, send overflow

29.2 I2S System Block Diagram

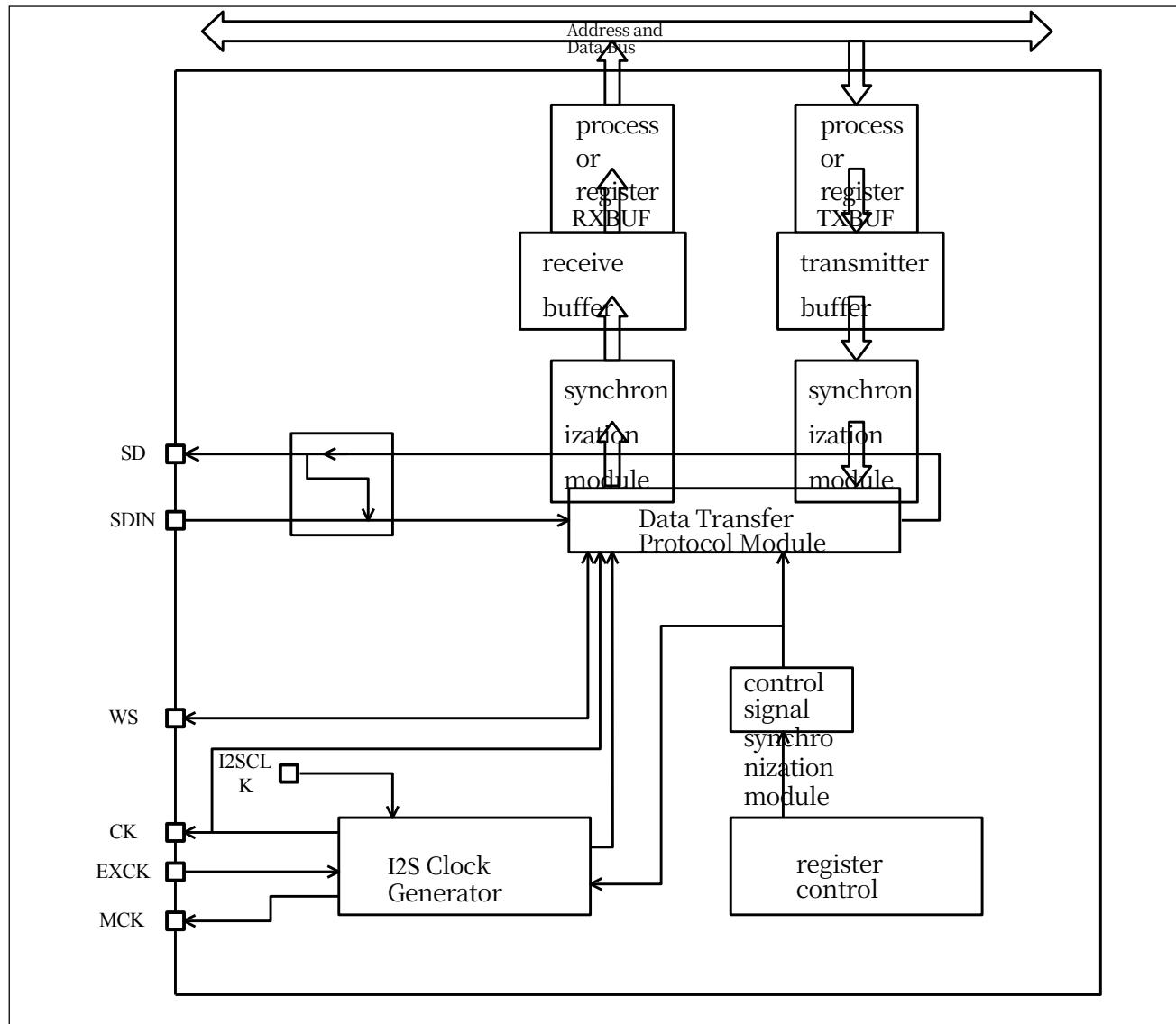


Figure 29-1 I2S System Block Diagram

29.3 Pin

**Descrip
tion**

Table 29-2 I2S Pin Descriptions

pin name	directional	Functional Description
I2Sn_CK	input and output	Communication Clock
I2Sn_WS	input and output	word choice
I2Sn_SD	input and output	serial data
I2Sn_SDIN	importation	Full-duplex audio data input
I2Sn_EXCK	importation	External clock source pin
I2Sn_MCK	exports	drive clock

n:1~4

29.4 Functional Description

The functions of the I2S are explained in detail in this chapter.

29.4.1 I2S General Description

I2S Pin Function

- I2Sn_SD: Serial data for half duplex mode data input, or half/full duplex mode data output.
- I2Sn_WS: word selection, is the data control signal output in master mode and the data control signal input in slave mode.
- I2Sn_CK: Serial clock, is the serial clock output in master mode and the serial clock input in slave mode.
- I2Sn_EXCK: External Clock Source, is the clock generator selecting external clock as the divider clock source in the main mode.
- I2Sn_SDIN: Pin used for serial data input in I2S full duplex mode.
- I2Sn_MCK: When I2S is configured as the master mode (and MCKOE position 1), this additional clock is output using the driver clock (mapped separately), which is output at a frequency of $256 \times F_s$, where F_s is the audio signal sampling frequency.

The I2S uses its own clock generator in master mode to generate the communication clock. This clock generator is also the source that drives the clock output.

29.4.2 communication method

Both half-duplex and full-duplex communication modes are supported and can be selected by the DUPLEX bit of the I2S control register (I2S_CTRL).

When the I2S_CTRL.DUPLEX bit is 0, the I2S operates in half-duplex communication mode, using the I2Sn_SD pin as the output data pin for transmit-only and the I2Sn_SD pin as the input data pin for receive-only.

When the I2S_CTRL.DUPLEX bit is 1, the I2S operates in the full-duplex communication mode, in which case the I2Sn_SDIN pin is used as the input data pin and the I2Sn_SD pin is used as the output data pin to realize full-duplex communication.

29.4.3 Supported Audio Protocols

There are four combinations of data and frame formats that can be used to send data in the following formats:

- Encapsulation of 16-bit data in a 16-bit frame
- Encapsulate 16-bit data in a 32-bit frame
- Encapsulates 24-bit data in a 32-bit frame
- Encapsulation of 32-bit data in 32-bit frames

When using 16-bit data from a 32-bit packet, the first 16 bits (MSB) are valid and the 16-bit LSB is forced to be cleared without any software operation (only one read/write operation is required)

When using 24-bit data in a 32-bit packet, the first 24 bits (MSB) are valid and the 8-bit LSB is forced to be cleared without any software operation (only one read/write operation is required)

For all data formats and communication standards, the most significant bit is always sent first (MSB first)

The I2S interface supports four audio protocols, which can be configured using the I2SSTD[1:0] and PCMSYNC bits in the I2S_CFGR register.

29.4.3.1 I2S Philips Standard

Use the WS signal to indicate the channel to which the data currently being sent belongs. This signal is valid from one clock before the first bit (MSB) of the current channel data.

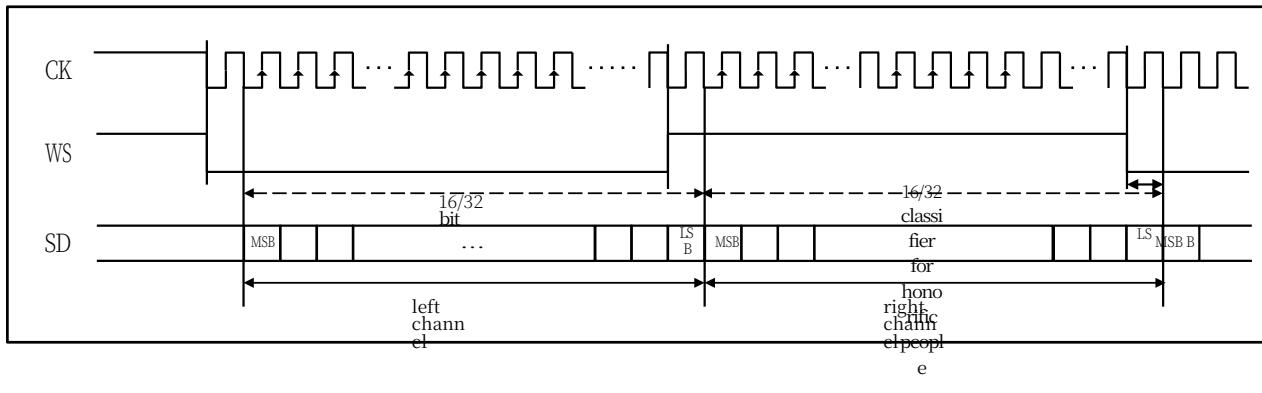


Figure 29-2 I2S Philips Protocol Waveform (16/32-bit Full Precision)

16-bit loaded in 16-bit frame, in transmit mode, write to I2S_TXBUF register

0xFFFF_3344, SD outputs serial data 0x3344jn receive mode\$D inputs serial data 0xEEDD, I2S_RXBUF register reads data 0x0000_EEDD.

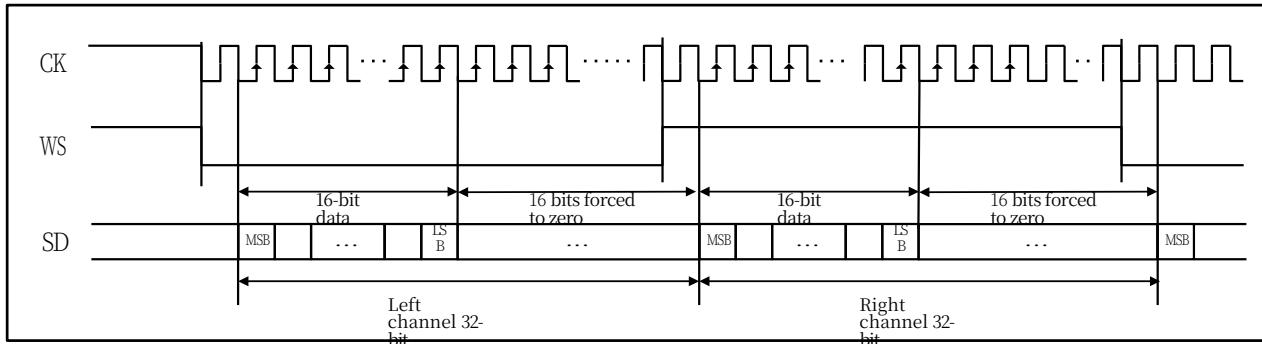


Figure 29-3 I2S Philips Protocol Waveform (16-bit data encapsulated in a 32-bit frame)

16-bit loaded in 32-bit frame, in transmit mode, write to I2S_TXBUF register 0xFFFF_3344, SD output serial data 0x3344_0000; in receive mode, SD input serial data 0xEEEE_EEEE, I2S_RXBUF register read data 0x0000_EEEE.

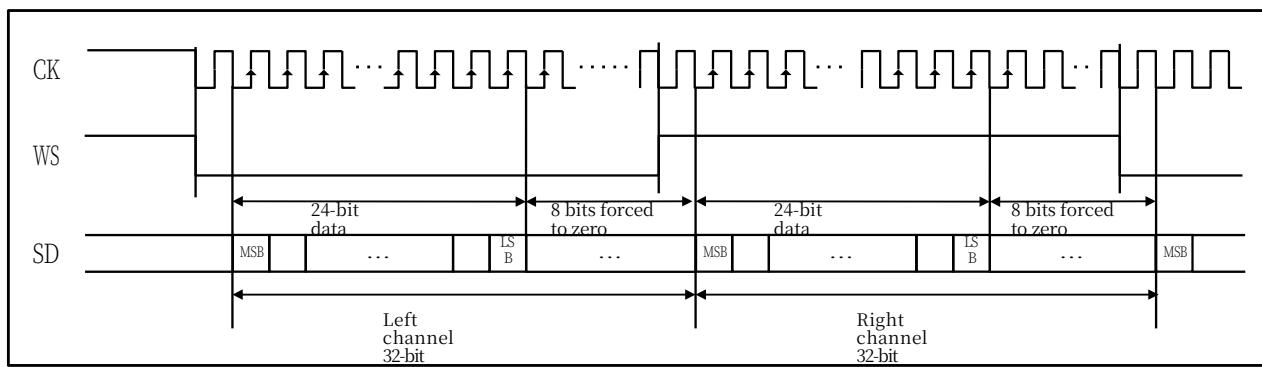


Figure 29-4 I2S Philips Protocol Waveform (24-bit data encapsulated in a 32-bit frame)

24-bit loaded in 32-bit frame, in transmit mode, write to I2S_TXBUF register 0xFF22_3344, SD outputs serial data 0x2233_4400; in receive mode, SD inputs serial data 0xEEEE_11XX, I2S_RXBUF register reads data 0x00EE_DD11.

29.4.3.2 MSB Alignment Criteria

This standard generates both the WS signal and the first data bit (i.e., **MS Bit**)

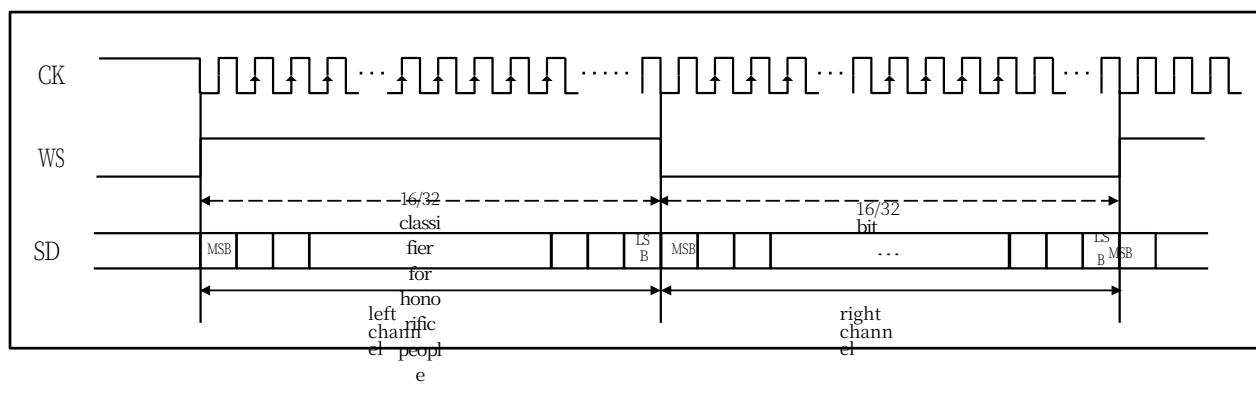


Figure 29-5 I2S MSB Protocol Waveform (16/32-bit Full Precision)

The sender changes the data on the falling edge of the clock signal; the receiver reads the data on the rising edge.

16-bit loaded in 16-bit frame, in transmit mode, write to I2S_TXBUF register 0xXXXX_3344, SD outputs serial data 0x3344jn receive mode\$D inputs serial data 0xEEDD, I2S_RXBUF register reads data 0x0000_EEDD.

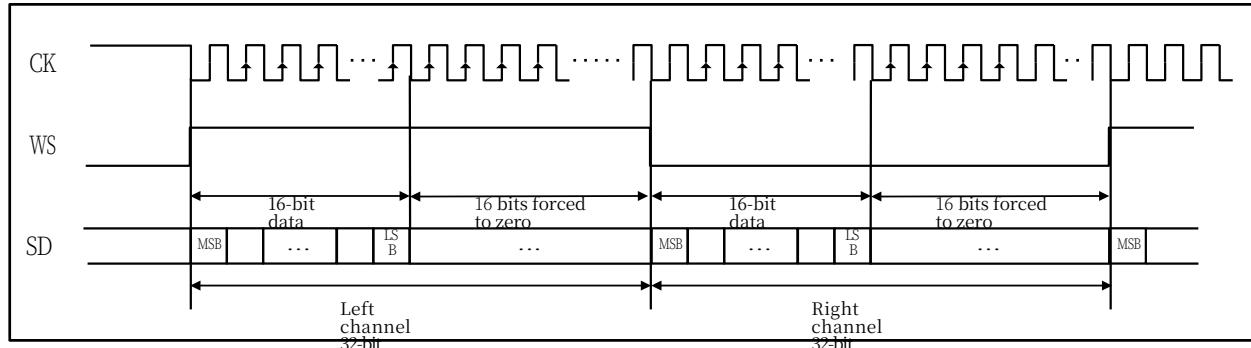


Figure 29-6 I2S MSB Protocol Waveform (16-bit data encapsulated in a 32-bit frame)

16-bit loaded in 32-bit frame, in transmit mode, write to I2S_TXBUF register 0xXXXX_3344, SD output serial data 0x3344_0000; in receive mode, SD input serial data 0xEEDD_XXXX, I2S_RXBUF register read data 0x0000_EEDD.

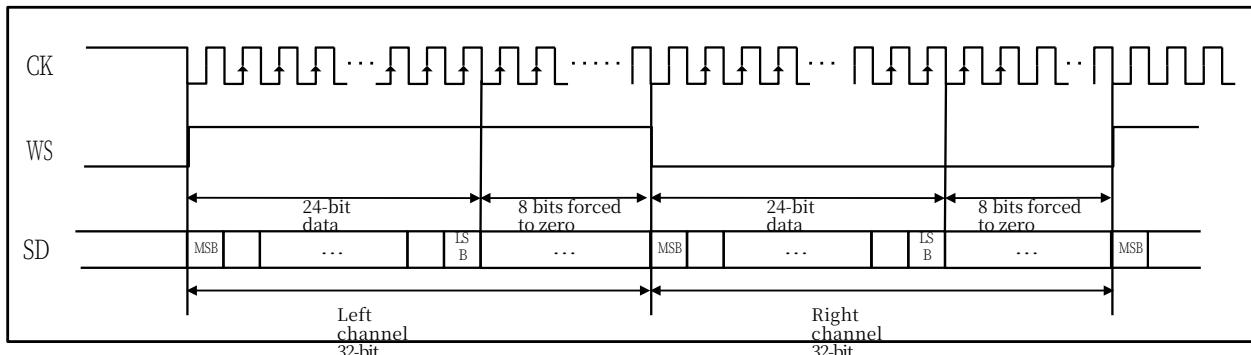


Figure 29-7 I2S MSB Protocol Waveform (24-bit data encapsulated in a 32-bit frame)

24-bit loaded in 32-bit frame, in transmit mode, write to I2S_TXBUF register 0xXX22_3344, SD outputs serial data 0x2233_4400; in receive mode, SD inputs serial data 0xEEDD_11XX, I2S_RXBUF register reads data 0x00EE_DD11.

29.4.3.3 LSB Alignment Criteria

The standard is similar to the MSB alignment standard (there is no difference for 16-bit and 32-bit full-precision frame formats)

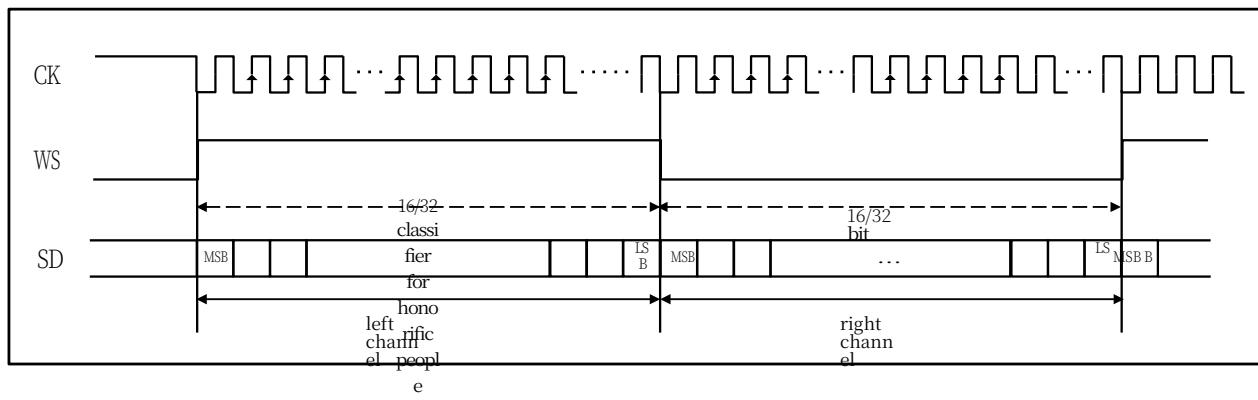


Figure 29-8 I2S LSB Protocol Waveform (16/32-bit Full Precision)

16-bit loaded in 16-bit frame, in transmit mode, write to I2S_TXBUF register 0xXXXX_3344, SD outputs serial data 0x3344jn receive mode\$D inputs serial data 0xEEDD, I2S_RXBUF register reads data 0x0000_EEDD.

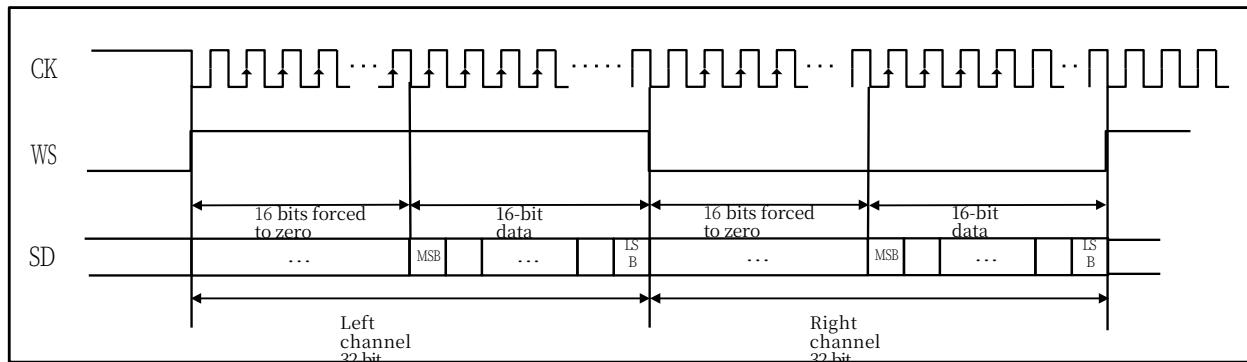


Figure 29-9 I2S LSB Protocol Waveform (16-bit data encapsulated in a 32-bit frame)

16-bit loaded in 32-bit frame, in transmit mode, write to I2S_TXBUF register 0xXXXX_3344, SD outputs serial data 0x0000_3344; in receive mode, SD inputs serial data 0xXXXX_1122, I2S_RXBUF register reads data 0x0000_1122.

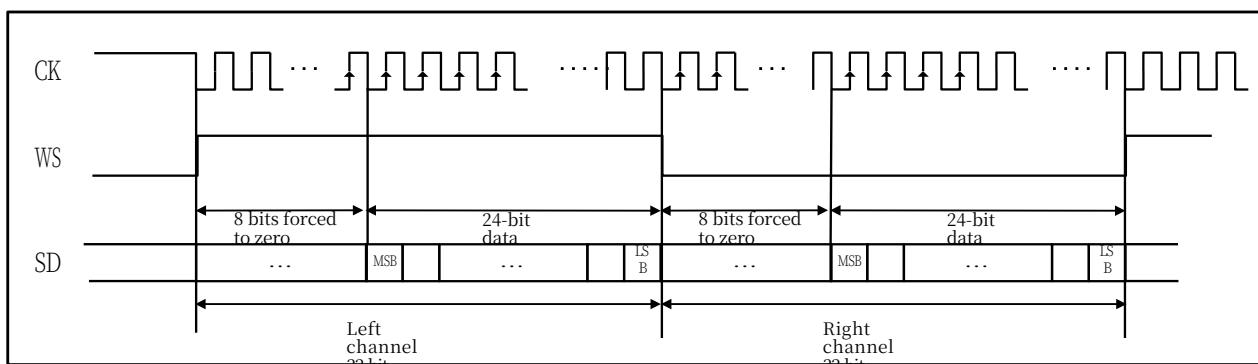


Figure 29-10 I2S LSB Protocol Waveform (24-bit data encapsulated in a 32-bit frame)

24-bit loaded in 32-bit frame, in transmit mode, write to I2S_TXBUF register 0xXX22_3344, SD outputs serial data 0x0022_3344; in receive mode, SD inputs serial data 0xXXDD_1122, I2S_RXBUF register reads data 0x00DD_1122.

29.4.3.4 PCM Standard

For the PCM standard, no channel information is required. There are two PCM modes (short frame and long frame) can be configured using the PCMSYNC bit in I2S_CFGR.

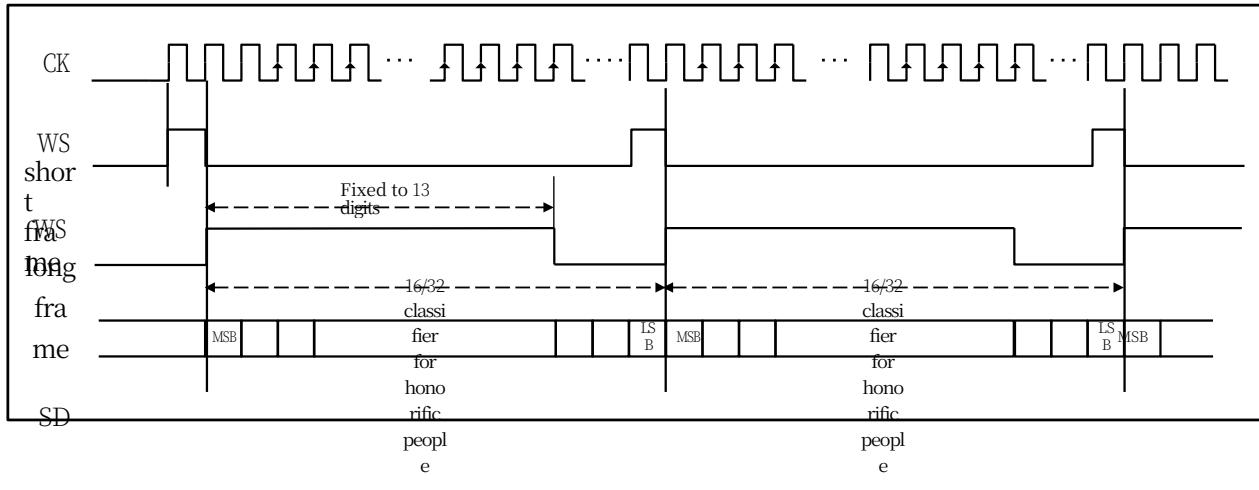


Figure 29-11 I2S PCM Protocol Waveform (16/32-bit Full Precision)

For long frame synchronization, the WS signal is sustained for 13 cycles in master mode. For short frame synchronization, the duration of the WS sync signal is only one cycle.

16-bit loaded in 16-bit frame, in transmit mode, write to I2S_TXBUF register 0xFFFF_3344, SD outputs serial data 0x3344; in receive mode, SD inputs serial data 0xEEEE, I2S_RXBUF register reads data 0x0000_EEEE.

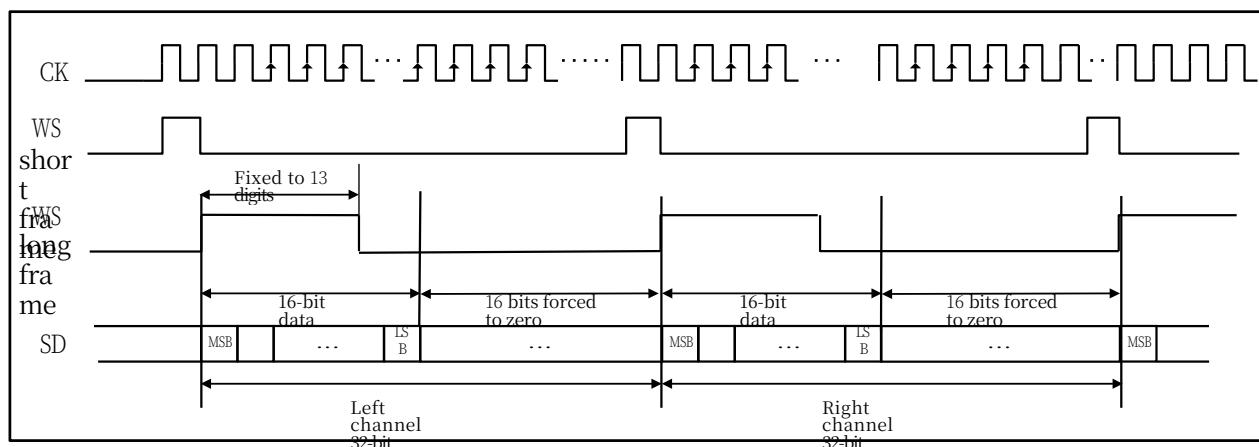


Figure 29-12 I2S PCM Protocol Waveform (16-bit data encapsulated in a 32-bit frame)

16-bit loaded in 32-bit frame, in transmit mode, write to I2S_TXBUF register 0xFFFF_3344, SD output serial data 0x0000_3344; in receive mode, SD input serial data 0xEEEE_XXXX, I2S_RXBUF register read data 0x0000_EEEE.

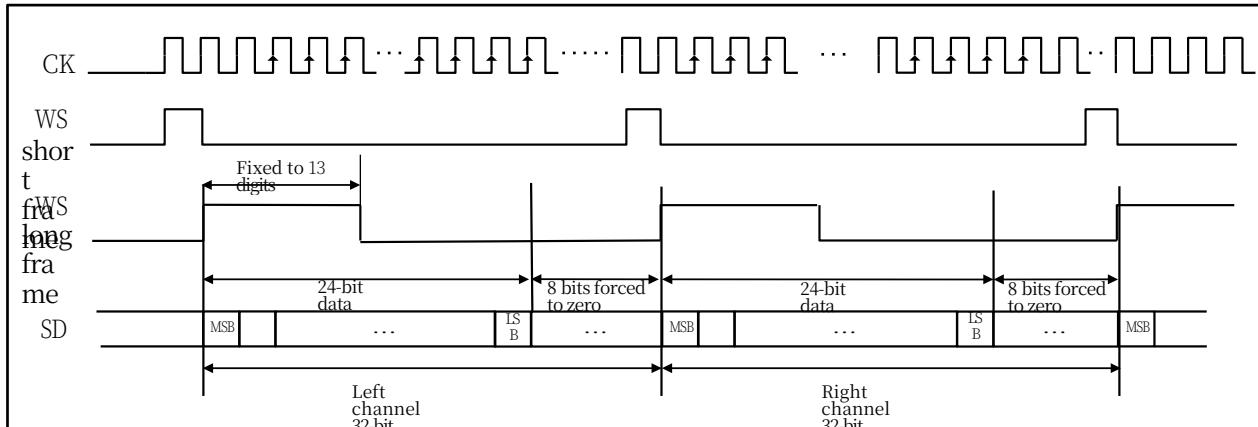


Figure 29-13 I2S PCM Protocol Waveform (24-bit data encapsulated in a 32-bit frame)

24-bit loaded in 32-bit frame, in transmit mode, write to I2S_TXBUF register 0xXX22_3344, SD outputs serial data 0x0022_3344; in receive mode, SD inputs serial data 0xEEDD_11XX, I2S_RXBUF register reads data 0x00EE_DD11.

Attention:

- For both modes (master/slave mode) and both synchronizations (short/long synchronization) it is necessary to specify the number of bits between two consecutive sets of data (and two synchronization signals) (DATLEN bit and CHLEN bit in the I2S_CFGR register) even in slave mode.

29.4.4 clock generator

The I2S bit rate is used to determine the data stream on the

I2S data lines and the I2S clock signal frequency. I2S Bit

Rate = Bits per Channel x Number of Channels x Audio

Sampling Frequency

For 16-bit dual-channel audio, the I2S bit rate is calculated as follows: I2S bit rate = $16 \times 2 \times F_s$. If the packet is 32 bits wide, I2S bit rate = $32 \times 2 \times F_s$.

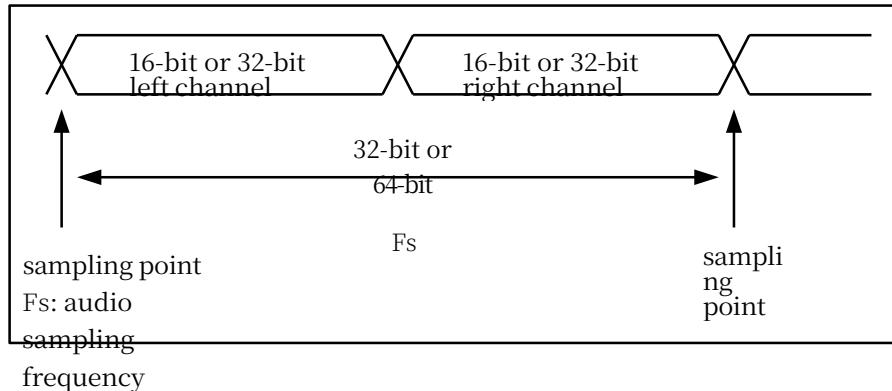


Figure 29-14 Audio Sampling Frequency Definition

When configuring the master mode, the linear crossover needs to be set correctly in order to communicate using the desired audio frequency.

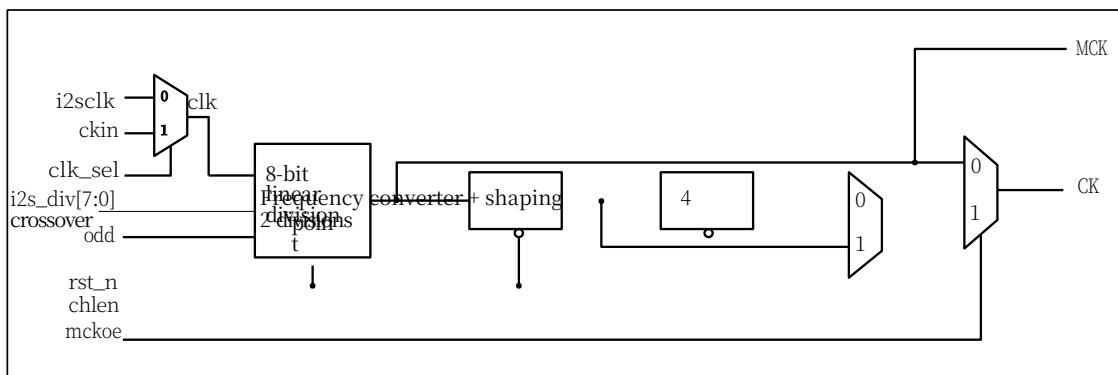


Figure 29-15 Clock Generator Architecture

The clk clock source can be the I2SCLK output or an external clock.

The audio sampling frequency may be 192kHz, 96kHz or 48kHz etc. To achieve the desired frequency, the current crossover needs to be programmed according to the following formula:

When the output driver clock (mckoe is set to 1):

$F_s = \text{clk} / [(16 \times 2) \times ((2 \times \text{I2SDIV} + \text{ODD}) \times 8)]$ (when the channel frame width is 16 bits)
 $F_s = \text{clk} / [(32 \times 2) \times ((2 \times \text{I2SDIV} + \text{ODD}) \times 4)]$ (when the channel frame width is 32 bits)

When the drive clock is turned off (mckoe is cleared):

$F_s = \text{clk} / [(16 \times 2) \times (2 \times \text{I2SDIV} + \text{ODD})]$ (when the channel frame width is 16 bits)

$F_s = \text{clk} / [(32 \times 2) \times (2 \times \text{I2SDIV} + \text{ODD})]$ (when the channel frame width is 32 bits)

The following table provides sample accuracy values for different clock configurations. There are other configurations available to achieve better clock accuracy.

Table 29-3 Audio Frequency Accuracy (for VCO Input Frequency = 1MHz)

drive clock	goal fS(Hz)	data format	octave factor	output crossover ratio	I2SDIV	ODD	Real-time fS (Hz)	inaccuracie s
Output shutdown	8000	16-bit	288	3	187	1	8000	0.0000%
		32-bit (computing)	256	4	62	1	8000	0.0000%
	16000	16-bit	256	4	62	1	16000	0.0000%
		32-bit (computing)	256	2	62	1	16000	0.0000%
	32000	16-bit	256	2	62	1	32000	0.0000%
		32-bit (computing)	256	5	12	1	32000	0.0000%
	48000	16-bit	384	10	12	1	48000	0.0000%
		32-bit (computing)	384	5	12	1	48000	0.0000%
	96000	16-bit	384	5	12	1	96000	0.0000%
		32-bit (computing)	424	3	11	1	96014.49219	0.0151%
output enable (computing)	22050	16-bit	290	3	68	1	22049.87695	0.0006%
		32-bit (computing)	302	2	53	1	22050.23438	0.0011%
	44100	16-bit	302	2	53	1	44100.46875	0.0011%
		32-bit (computing)	429	4	19	0	44099.50781	0.0011%
	192000	16-bit	424	3	11	1	192028.9844	0.0151%
		32-bit (computing)	258	3	3	1	191964.2813	0.0186%
	8000	having nothing do (with sth else)	256	5	12	1	8000	0.0000%

ng)	16000	having nothing do (with sth else)	426	4	13	0	16000.60059	0.0038%
	32000	having nothing do (with sth else)	426	4	6	1	32001.20117	0.0038%
	48000	having nothing do (with sth else)	258	3	3	1	47991.07031	0.0186%
	96000	having nothing do (with sth else)	344	2	3	1	95982.14063	0.0186%
	22050	having nothing do (with sth else)	429	4	9	1	22049.75291	0.0011%
	44100	having nothing do (with sth else)	271	2	6	0	44108.07422	0.0183%

Attention:

- The octave factor can be UPLLN/MPLLN, the output crossover ratio can be UPLLP/UPLLQ/UPLLR when the octave factor is UPLLN, and the output crossover ratio can be MPLLP/MPLLQ/MPLLR when the octave factor is MPLLN. For details, refer to the CMU UPLL Configuration Register and CMU MPLL Configuration Register of the clock controller (CMU). For details, refer to the CMU UPLL Configuration Register and CMU MPLL Configuration Register of the CMU.

29.4.5 I2S Master Mode

I2S can be configured as follows:

- Send master or receive master (half-duplex mode using I2S)
- Simultaneously transmitting and receiving master devices (full-duplex mode using I2S)

The I2S operates in master mode with the serial clock output from pin CK and the word select signal generated from pin WS. You can choose to output or not to output the drive clock (MCK) by setting the MCKOE bit of register I2S_CTRL.

move

1. Sets the pins to be used for I2S.
2. The clock source is selected via the I2S_CTRL.CLKSEL, I2S_CTRL.I2SPLLSEL bits.
3. Set the I2S_PR.I2SDIV[7:0] bits and the I2S_CTRL.ODD bits to define the serial data baud rate to achieve the proper audio sampling frequency.
4. Sets the I2S Configuration Register (I2S_CFGR), selects the I2S standard with the I2SSTD[1:0] and PCMSYNC bits, selects the data length with the DATLEN[1:0] bits and selects the number of bits per channel with the Configuration CHLEN bits.
5. If you need to use interrupts, set the system's interrupt register.
6. If you want to use DMA, set the registers related to DMA.
7. Sets the I2S control register (I2S_CTRL), including operating mode setting, communication mode setting, clock output license setting, data output license setting, FIFO reset setting, transmit/receive buffer threshold setting, etc. The operating mode WMS bit selects the I2S master mode.
8. Sets the interrupt license bit.
9. Set I2S_CTRL.TXE and I2S_CTRL.RXE and the action starts.

Attention:

- When using the TXIRQOUT interrupt to write communication data to TXBUF, if two data are written per interrupt, first turn off the transmit interrupt enable flag bit TXIE after the interrupt starts, and then turn on TXIE after the data has been written. or write only one data per interrupt.

Send Sequence

TXE position 1 in the I2S_CTRL register allows transmitting. The transmit sequence begins as soon as the data is written to the transmit buffer. A complete frame indicates a left channel data send followed by a right channel data send. There are no partial frames where only the left channel is sent. During the first header send, the data is loaded in parallel into the shift register and then shifted serially and output to the SD pin(MSB in front) See [Supported Audio Protocols] for more details on

write operations in the various I2S standard modes.

receive sequence

This mode of operation is essentially the same as transmit mode, differing only in the transmit-receive setting of the I2S_CTRL register, which sets RXE position 1 to allow reception. See [Supported Audio Protocols] for more detailed information on read operations in the various I2S standard modes. If there is a difference in the

Receiving new data when previously received data has not been read will generate an overflow error and set the I2S_ER.RXERR flag to 1. If the I2S_CTRL.EIE position is 1, an interrupt will be generated to indicate the error.

29.4.6 I2S Slave Mode

I2S can be configured as follows:

- Transmit slave or receive slave (half-duplex mode using I2S)

- Slave device that transmits and receives at the same time (full duplex mode using I2S)

This mode of operation follows essentially the same rules as the I2S master mode. In slave mode, the I2S interface does not generate a clock. The clock and WS signals are input from the external master device to which the I2S interface is connected. Thus, the user does not need to configure the clock.

move

1. Sets the pins to be used for I2S.
2. Sets the I2S Configuration Register (I2S_CFGR) selects the I2S standard with the I2SSTD[1:0] and PCMSYNC bits, selects the data length with the DATLEN[1:0] bits and selects the number of bits per channel with the Configuration CHLEN bits.
3. If you need to use interrupts, set the system's interrupt register.
4. If you want to use DMA, set the registers related to DMA.
5. To send data, 1~4 data to be sent should be pre-written to I2S_TXBUF first.
6. Sets the I2S control register (I2S_CTRL), including operating mode setting, communication mode setting, clock output license setting, data output license setting, FIFO reset setting, transmit/receive buffer threshold setting, etc. The operating mode WMS bit selects the I2S slave mode.
7. Sets the interrupt license bit.
8. Set I2S_CTRL.TXE and I2S_CTRL.RXE and the action starts.

Send Sequence

The transmit sequence starts when WMS is set to 1, TXE is set to 1, the external master device transmits the clock and requests data transmission via the WS signal. At the start of communication, data is transferred from the transmit buffer to the shift register. During the first send, data is loaded in parallel from the internal bus into the shift register and then shifted serially and output to the SD pin (MSB first) Each time the transmit buffer FIFO space is greater than the set threshold, an interrupt is generated if I2S_CTRL.TXIE position 1. See [Supported Audio Protocols] for more details on write operations in the various I2S standard modes.

To ensure continuous audio data transmission, the next data to be sent must be written to the TX FIFO before the current data transmission ends. if the first clock edge of the next data

communication arrives before the data has been written to the TX FIFO, a transmit overflow occurs and the I2S_ER.TXERR flag is set to 1 and an interrupt may be generated. If I2S_CTRL.EIE position 1, an interrupt will be generated when the I2S_ER.TXERR flag changes to 1.

receive sequence

This mode of operation is essentially the same as transmit mode, differing only in the transmit-receive setting of the I2S_CTRL register, which sets RXE position 1 to allow reception. See [Supported Audio Protocols] for more detailed information on read operations in the various I2S standard modes. If new data is received when previously received data has not been read, a receive overflow error is generated and the RXERR flag is set to 1. If I2S_CTRL.EIE position 1, an interrupt is generated to indicate the error.

29.4.7 I2S Interrupt

The I2S interrupt sources are transmit buffer valid space greater than alarm threshold, receive buffer valid space less than alarm threshold, receive overflow, transmit underflow, and transmit overflow. Transmit underflow and transmit overflow are integrated into the I2S transmit error interrupt TXERR, so it is necessary to determine the actual interrupt source by the flag. The specific description of the I2S interrupt sources is shown in Table 29-4. Once the interrupt condition is established, the corresponding interrupt request is generated.

Users can write the vectors corresponding to the above event trigger sources into different trigger object registers to realize various event trigger functions. Please refer to "Interrupt Controller (INTC)" for the vectors corresponding to the above event trigger sources.

Table 29-4 shows the list of I2S interrupts.

Table 29-4 I2S Interrupt Requests

disruption event	event marker	Enable Control Bit
Transmit buffer active space greater than alarm threshold	TXBA	TXIE
Receive buffer active space less than alarm threshold	RXBA	RXIE
Receive data area is full and there is still a write data request, receive overflow	RXERR	EIE
Send data area is empty and there is still a request to send, send underflow	TXERR	EIE
Send data area is full and there is still a request to write data, send overflow	TXERR	EIE

29.4.8 Precautions for use

29.4.8.1 Precautions for use as a host

- 1) When the I2S acts as a host for the data only sending action, if all the data in the

I2S_TXBUF has been sent and no new data is written, the I2S will pause the action after the last data has been sent, and at this time, the I2S will no longer generate the communication clock, and the send error flag TXERR will be set to 1. At this time, the user can choose to shut down the I2S by writing the I2S_CTRL.TXE bit to 0, or write new send data to the I2S_TXBUF to continue the sending action. TXE bit to 0 to turn off the I2S, or write new data to I2S_TXBUF to continue sending. The WS will restart from the left channel (Philips, MSB/LSB modes) when transmitting is resumed.

If the I2S_CTRL.TXE bit is directly written to 0 during the transmit action, the I2S will be turned off immediately and the current data transmission will be terminated. This will cause the slave to be unable to grasp the state of the slave, and when the communication is restarted without resetting the slave, the data received by the slave will be messed up. Therefore, it is recommended that the user writes the I2S_CTRL.TXE bit to 0 when the I2S is in the pause state in order to shut down the I2S.

- 2) When the I2S acts as a host for a data-only receive operation, if you want to stop the receive operation temporarily, you can write two frames of dummy in advance.

TXE is set to 1, 4 data TXE is set to 1 in advance when the baud rate is 8k~96k, 5 data TXE is set to 1 in advance when the baud rate is 192k, when these two frames of dummy data are sent, I2S will pause, and at this time, I2S will no longer generate the communication clock, and the transmit error flag bit TXERR will be set to 1. At this time, the user can choose to turn off I2S by clearing I2S_CTRL.TXE and I2S_CTRL.RXE to zero, or write another frame of dummy data to I2S_TXBUF to restart the communication action, and then turn off TXE when receiving the first data after the pause, and then the WS will be restarted from the left channel again when restarting the action (Philips, MSB/LSB mode). The WS will restart from the left channel during the restart action (Philips, MSB/LSB mode) When the communication clock is regenerated, clear I2S_CTRL.TXE to zero to return to the receive-only state. Please refer to the following figure for details.

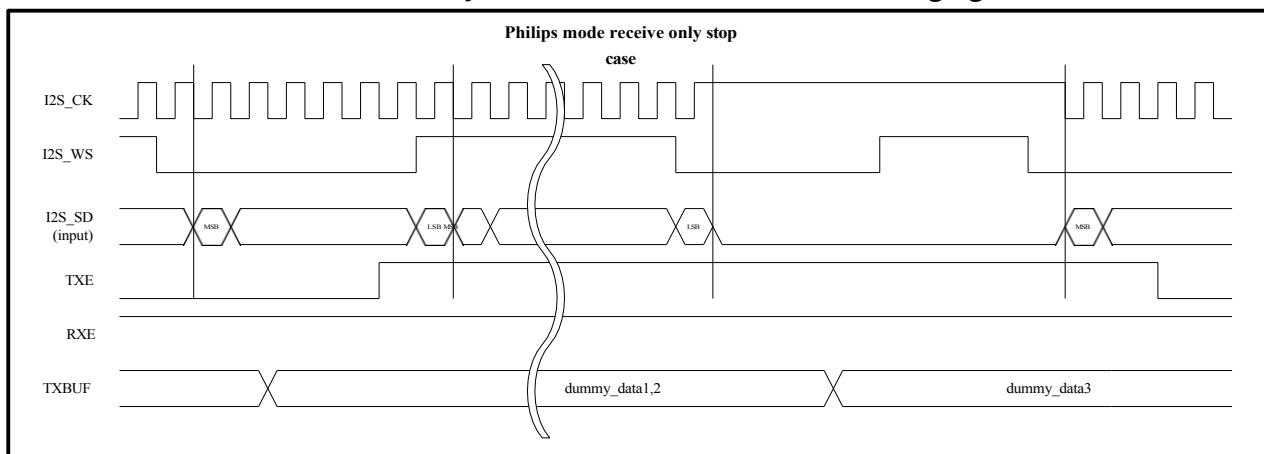


Figure 29-16 Host Receives Only Temporarily Stops Receiving

If the I2S_CTRL.RXE bit is written to 0 directly during the receive action, I2S will be turned off immediately and the current data reception will be terminated. This practice will cause the slave to be unable to grasp the state of the slave, and when communication is restarted without resetting the slave, the data sent by the slave will be messed up. Therefore, it is recommended that the user clear the I2S_CTRL.TXE and I2S_CTRL.RXE bits to zero to turn off the I2S when the I2S is in the suspend state.

3) When the I2S acts as a host for full duplex operation, if all the data in I2S_TXBUF has been sent and no new data is written, the I2S will pause after the last data has been sent, at this time, the I2S will no longer generate the communication clock, and the transmit error flag TXERR will be set to 1. TXE and I2S_CTRL.RXE can be cleared to zero to shut down the I2S, or write new data to I2S_TXBUF to continue sending and receiving. The WS will restart from the left channel when transmitting continues (Philips, MSB/LSB mode)

If I2S_CTRL.TXE and I2S_CTRL.RXE bits are directly written to 0 during full-duplex operation, I2S will be shut down immediately, and the current data transmission and

reception will be terminated. This practice will lead to the slave state can not be grasped, without resetting the slave to restart the communication caused by the communication data confusion, so it is recommended that users in the I2S in the suspended state of the I2S will be I2S_CTRL.TXE and I2S_CTRL.RXE bit clear zero to shut down the I2S.

4) When the I2S acts as a host for PCM short-frame data sending action, when the I2S is paused because there is no new data writing action for I2S_TXBUF, there are two ways to set up to restart sending, and the specific choice of which way needs to be determined according to the data receiving specifications of the slave.

If the slave has to detect the status of WS every time it receives data, it is necessary to set I2S_CTRL.TXE to 0 after I2S pause, and then write new transmit data to I2S_TXBUF and then set I2S_CTRL.TXE to 1 to restart transmission. Specific

The action is shown below.

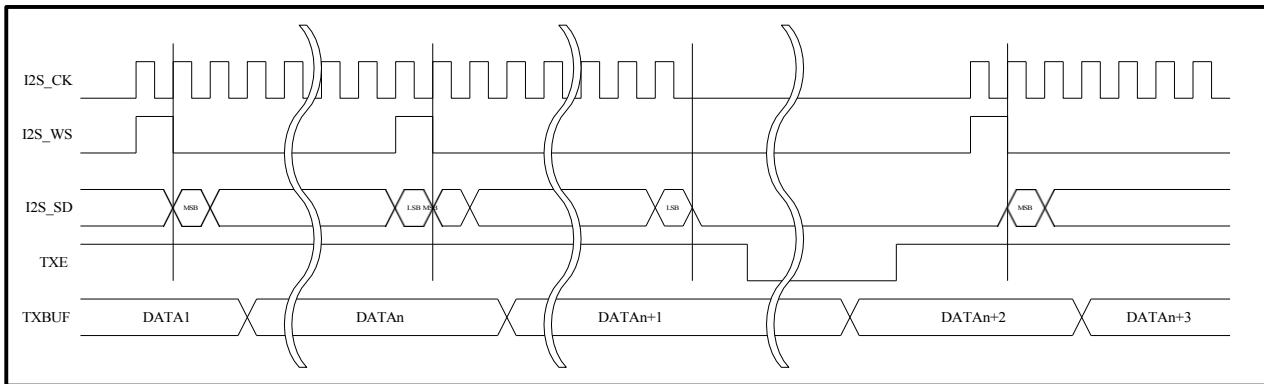


Figure 29-17 PCM Short Frame Host Transmission Pause and Resend Mode 1

If the slave detects the WS status only when it receives the first frame of data, the I2S can be paused and then restarted by writing new transmit data directly to I2S_TXBUF. The specific action is shown in the following figure.

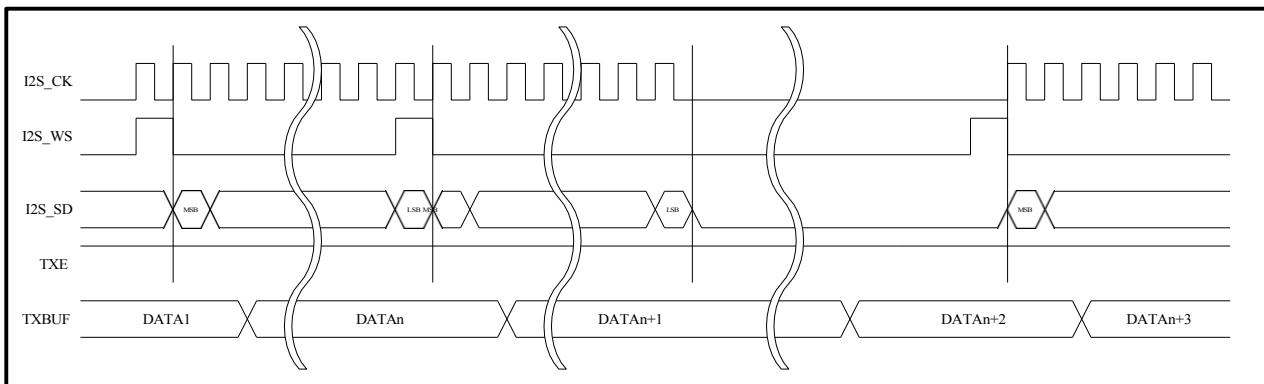


Figure 29-18 PCM Short Frame Host Transmission Pause and Resend Method 2

29.4.8.2 Precautions for use as a slave

- 1) When I2S is used as a slave, make sure that I2S_CTRL.TXE or I2S_CTRL.RXE is turned on last after all registers are configured to start the slave.
- 2) When I2S starts slave operation in Philips, MSB/LSB mode, it is necessary to ensure that the WS signal is at the right channel level at startup, and when I2S starts slave operation in PCM mode, it is necessary to ensure that the WS signal is at a low level at startup.
- 3) When the I2S receives data as a slave, each received data will not be read until the next frame of data reception starts, so when communication is suspended or terminated, the last frame of data received by the I2S will not be read until the next communication starts.
- 4) When the I2S performs data reception action as a slave in Philips, MSB/LSB modes, the WS is checked to see if the WS is at the left channel level before each frame of left channel data reception. When the I2S acts as a slave for data reception in PCM mode, the WS is checked to see if the valid level is generated according to the

standard communication protocol before each frame is received.

29.5 Register Description

I2S1 base address:

0x4001_E000 I2S2 base

address: 0x4001_E400 I2S3

base address: 0x4002_2000

I2S4 base address:

0x4002_2400

Table 29-5 I2S Register List

register name	notation	offset address	bit width	reset value
I2S Control Register	I2S_CTRL	0x000	32	0x0000_2200
I2S Status Register	I2S_SR	0x004	32	0x0000_0014
I2S Error Status Register	I2S_ER	0x008	32	0x0000_0000
I2S Configuration Register	I2S_CFGR	0x00C	32	0x0000_0000
I2S transmit buffer FIFO data registers	I2S_TXBUF	0x010	32	0x0000_0000
I2S receive buffer FIFO data registers	I2S_RXBUF	0x014	32	0x0000_0000
I2S prescaler register	I2S_PR	0x018	32	0x0000_0002

Note: Only 32-bit write registers

are supported CMU_BASE_ADDR2

: 0x40054000

register name	notation	offset address	bit width	reset value
CMU I2S Clock Configuration Registers	CMU_I2SCKSEL	0x012	16	0xB BBBB

Note: This register is detailed in the [Clock Controller (CMU) section.I2S Master Mode Clock Source Selection I2SPLL Use this register to configure the clock source, which can be configured as UPLL/R/UPLL/Q/UPLL/P/MPLL/R/MPLL/Q/MPLL/P.

29.5.1 I2S Control Register (I2S_CTRL)

I2S Control Register

Offset Address: 0x000

Reset value: 0x0000_2200

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	
-	-	-	-	-	-	-	-	CLKSEL L	DUPLEX X	CKOEN	LRCKOE	SDOE	I2SPLL LSEL	CODEC RC	FIFOR
b16 b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	
<hr/>															
-	RXBIRQWL	-	-	TXBIRQWL	MCKOE	ODD	WMS	EIE	RXIE	RXE	TXIE	TXE			
<hr/>															
b0															

classifier for marking	Bit Name	Function	Read/Write	
honorable people				
b31~b27	Reserved	-0" for reading, "0" for writing. "0", writes "0".	R	Reads
b23	CLKSEL	Clock source selection 0: Select I2SPLL 1: Select external clock		R/W
b22	DUPLEX	Communication mode selection duplex R/W 1: Full duplex		0: Half
b21	CKOEN	Communication clock output license Output disable R/W 1: Output license		0:
b20	LRCKOE	Channel clock output license Output disabled R/W 1: Output license 0: Output inhibit		0:
b19	SDOE	Data output license 1: Output license 0: Input inhibited		R/W
b18	I2SPLLSEL	I2SPLL input selection 0: Input disabled 1: Enter the license		R/W
b17	CODECRC	Codec reset control 0: Software reset 1: Unreset		R/W
b16	FIFOR	fifo reset 0: Unreset 1: Software reset		R/W
b15	Reserved	-0" for reading, "0" for writing. "0", writes "0".	R	Reads
b14~b12	RXBIRQWL[2:0]	Receive buffer interrupt request level Interrupt request triggered when the available space is less than the set value Note: Can only be set to 0/1/2 because the fifo space is 2.	R/W	
b11	Reserved	-0" for reading, "0" for writing. "0", writes "0".	R	Reads
b10~b8	TXBIRQWL[2:0]	Send buffer interrupt request level Interrupt request triggered when more space is available than the set value		

			Note: Can only be set to 0/1/2 because the fifo space is 2.
b7	MCKOE	Drive Clock Output Enable 1: Enable Drive Clock Output Note: This bit is only used when in I2S master mode 0: Actual	R/W
b6	ODD factor	Prescaler odd crossover value = I2SDIV x 2 1: Actual crossover value = I2SDIV x 2 + 1 Note: This bit is only used when in I2S master mode, to set ODD to 1, you should set the I2S_PR register first, then the I2S_CTRL register.	R/W
b5	WMS	I2S operating mode selection 0: I2S master mode 1: I2S slave mode	R/W
b4	EIE	Communication error interrupt enable communication error interrupt disabled	0: R/W

			1: Communication error interrupt valid	
b3	RXIE	Receive interrupt enable	0: Receive interrupt disabled 1: Receive interrupt active	R/W
b2	RXE	Receive Enable	0: Receive disabled 1: Permission to receive	R/W
b1	TXIE	Transmit Interrupt Enable	0: Transmit interrupt disabled 1: Transmit interrupt active	R/W
b0	TXE	Transmit Enable	0: Disable transmission 1: Permission to send	R/W

29.5.2 I2S Status Register (I2S_SR)

I2S Status Register

Offset Address: 0x004

Reset value: 0x0000_0014

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
-	-	-	-	-	-	-	-	-	-	RXBF	RXBE	TXBF	TXBE	RXBA	TXBA

classifier for honorable people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b6	Reserved	-	Reads "0", writes "0".	R
b5	RXBF	Receive buffer full	0: Receive buffer not full 1: Receive buffer full R	
b4	RXBE	Receive buffer empty	0: Receive buffer not empty 1: Receive buffer empty R	
b3	TXBF	Send buffer full	0: send buffer not full 1: Send buffer full	R
b2	TXBE	Send buffer empty	0: Send buffer not empty 1: Send buffer empty	R
b1	RXBA	Receive buffer alarm (in relation to water level)	0: Receive buffer not alarmed 1: Receive buffer alarm R	
b0	TXBA	Transmit Buffer Alarm (in relation to water level)	0: Send buffer not alarmed 1: Send buffer alarm	R

29.5.3 I2S Error Status Register (I2S_ER)

I2S Error Status Register

Offset Address: 0x008

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
-	-	-	-	-	-	-	-	-	-	-	-	-	-	RXERR	TXERR

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b2	Reserved	-	Reads "0" and writes "0".	R
b1	RXERR	Receive error	0: Flag bit not cleared 1: clear flag bit R/W	
b0	TXERR	Send error	0: Flag bit not cleared 1: clear flag bit R/W	

TXERR=1 when a transmit overflow/underflow occurs and RXERR=1 when a receive overflow occurs.

Write 1 to the TXERR bit to clear the flag bit when a transmit overflow/underflow occurs, and write 1 to the RXERR bit to clear the flag bit when a receive overflow occurs.

29.5.4 I2S Configuration Register (I2S_CFGR)

I2S Configuration Register

Offset Address: 0x00C

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
-	-	-	-	-	-	-	-	-	-	PCMSY NC	CHLEN	DATLEN[1:0]	I2SSTD[1:0]		

classifier for marking	celebrity	functionality	fill out or in (information on a form)
honorable people			
b31~b6	Reserved	-	Reads "0", writes "0".
			PCM frame synchronization
b5	PCMSYNC	PCM frame synchronization	0: Short frame synchronization 1: Long frame synchronization Note: This bit is only meaningful if I2SSTD=11 (using the PCM standard)
			R/W
b4	CHLEN	Channel length	Mono one-frame data length selection 0: 16bit 1: 32bit
			R/W
b3~b2	DATLEN[1:0]	Transmission data length selection	Transmission data length selection 00: 16bit 01: 24bit 1X: 32bit
			R/W
b1~b0	I2SSTD[1:0]	Communication protocol selection	Communication protocol selection 00: Philips Protocol 01: MSB justified protocol (left-aligned) 10: LSB justified protocol (right-aligned) 11: PCM Protocol
			R/W

29.5.5 I2S Transmit Buffer FIFO Data Register (I2S_TXBUF)

I2S Transmit Buffer FIFO Data Register

offset address: 0x010

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
TXBUF[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TXBUF[15:0]															
<hr/>															
classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)											
b31~b0	TXBUF[31:0]	Sending data	Storing Transmission Data	W											

Notes:

- For 16-bit frames, TXBUF[15:0] stores one frame of left channel or one frame of right channel transmit data.
- For 32-bit frames, TXBUF[31:0] stores one frame of left channel or one frame of right channel transmit data.

29.5.6 I2S receive buffer FIFO data register (I2S_RXBUF)

I2S Receive Buffer FIFO Data Register

Offset Address: 0x014

Reset value: 0x0000_0000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
RXBUF[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RXBUF[15:0]															
<hr/>															
classifier for honorific people	markin g	celebrity	functionality	fill out or in (information on a form)											
b31~b0	RXBUF [31:0]	receive data	Storing Received Data	R											

Notes:

- For 16-bit frames, RXBUF[15:0] stores one frame of left channel or one frame of right channel receive data.
- For 32-bit frames, RXBUF[31:0] stores one frame of left channel or one frame of right channel receive data.

29.5.7 I2S Frequency Divider Register (I2S_PR)

I2S Prescaler Register

Offset Address: 0x018

Reset value: 0x0000_0002

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
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	I2SDIV[[7:0]]

classifier for marking honorific people	celebrity	functionality	fill out or in (information on a form)
b31~b8	Reserved	-	Reads "0", writes "0".
b7~b0	I2SDIV[7:0]	crossover factor	R/W

I2SDIV[7:0]=0 or I2SDIV[7:0]=1 for disabled values
See 4.4 Clock Generator Module
Actual crossover value = I2SDIV x 2 + I2S_CTRL.ODD
00000010: 2 crossover frequency
00000011: 3-way frequency
00000100: 4-way
...
11111101: 253 crossover
11111110: 254 crossover
11111111: 255 crossover
Note: This bit is only used when in I2S master mode

30 Controller Area Network (CAN)

30.1 summary

CAN (Controller Area Network) bus is a bus standard that enables microprocessors or devices to communicate with each other without a host. This module follows the CAN bus protocols 2.0A and 2.0B and is upwardly compatible with CAN-FD. The CAN bus controller handles the sending and receiving of data on the bus, and in this product, CAN has 8 sets of filters. The filters are used to select the messages to be received for the application program.

The application program sends the transmit data to the bus through one high priority **Primary Transmit Buffer** (hereinafter referred to as PTB) and four **Secondary Transmit Buffers** (hereinafter referred to as STB), and the transmit scheduler determines the order in which the mailboxes are sent. The bus data is acquired by 10 **Receive Buffers** (hereinafter referred to as RBs). 4 STBs and 10 RBs can be interpreted as a 4-stage FIFO and a 10-stage FIFO, which is completely controlled by hardware.

The CAN bus controller can also support time-trigger communication.

CAN Main Features:

- Fully supports CAN2.0A/CAN2.0B protocol.
- Upwardly compatible with CAN-FD protocol.
- Supports a maximum communication baud rate of 1Mbit/s
- Supports 1~1/256 baud rate prescaling, flexible baud rate configuration.
- 10 receive buffers
 - FIFO method
 - Errors or non-received data do not overwrite stored messages.
- 1 high-priority master transmitter buffer PTB
- 4 sub transmitter buffers STB
 - FIFO method
 - Prioritized arbitration approach
- 8 groups of independent filters
 - Supports 11-bit standard IDs and 29-bit extended IDs
 - Programmable ID CODE bit and MASK bit
- Both PTB/STB support single transmission mode
- Supports silent mode
- Supports loopback mode
- Support for capturing the type of errors transmitted and locating arbitration failures
- Programmable error warning values

- Supports ISO 11898-4 time-triggered CAN and receive timestamps

30.2 CAN System Block Diagram

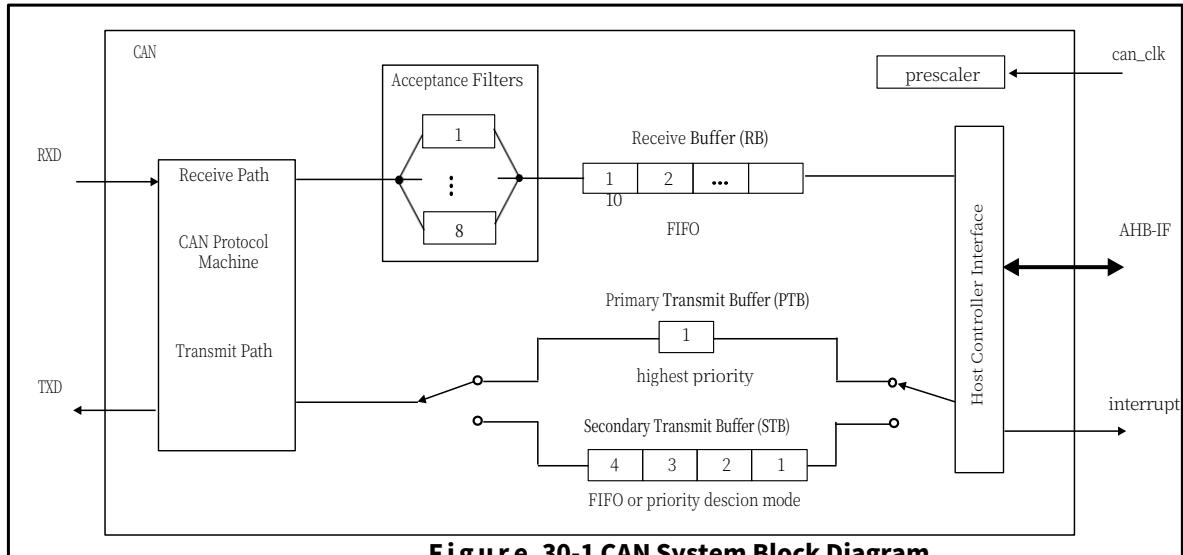


Figure 30-1 CAN System Block Diagram

30.3 Pin

Descrip
tion

Table 30-1 CAN Pin Descriptions

pin name	orientations	Functional Description
CAN_RxD	importation	CAN receive data signal
CAN_TxD	exports	CAN transmit data signal

30.4 Functional Description

This chapter describes the CAN function in detail.

30.4.1 Baud rate setting

The clock source of CAN communication clock can_clk is an external high-speed oscillator. Before using the CAN module, you need to set the CAN communication clock in the CMU section. The setting condition that EXCLK (CAN control logic clock) is 1.5 times or more of can_clk (CAN communication clock) must be satisfied for clock selection.

The following figure shows the CAN bit time definition. The upper dashed line is the bit time defined by the CAN protocol, and the lower dashed line is the bit time defined by the CAN controller CAN-CTRL. The segment1 and segment2 can be set by register BT, which can only be set when CFG_STAT.RESET=1, i.e. CAN software reset.

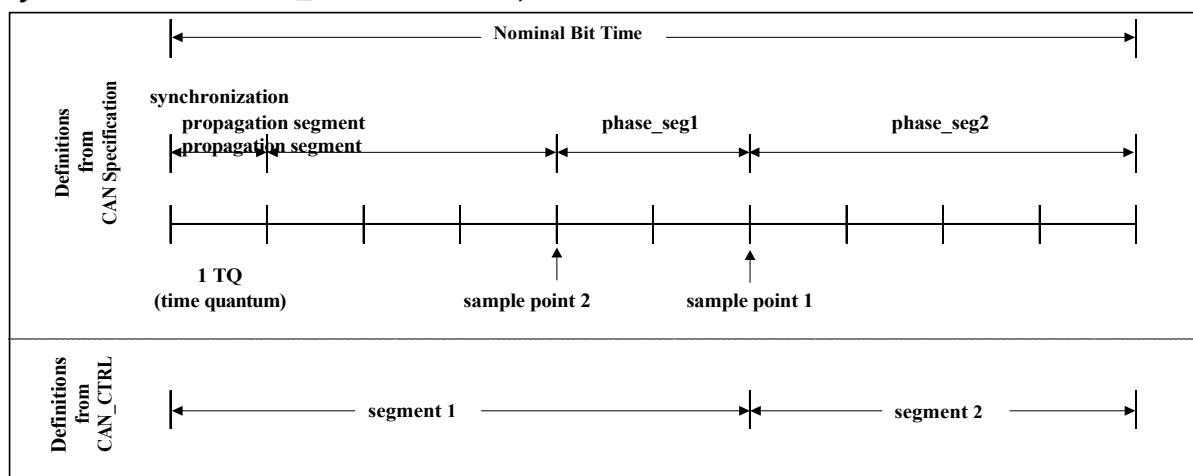


Figure 30-2 CAN Bit Time Definition Diagram

Please refer to the following formula for TQ calculation, where PRESC is set by the PRESC bit of the BT register. fcan_clk is the CAN communication clock frequency.

$$TQ = \frac{PRESC+1}{fcan_clk}$$

Please refer to the following instructions for the calculation of sampling points.

- 1) If the PRESC bit of the BT register is set to ≥ 1 , the sample point is located at the division point of segment1 and segment2 as shown in sample point 1 of Figure 30-2.
- 2) If the PRESC bit of the BT register is set to 0, the sample point is 2 TQs in front of the segment1 and segment2 demarcation points as shown in Figure 30-2. If the PRESC bit in the BT register is set to 0, the sample point is 2 TQs in front of the segment1 and segment2 demarcation point as shown in Figure 30-2.

It is recommended that the PRESC bit of the BT register be set to a value ≥ 1 .

Refer to the following formula for the bit time calculation method, where SEG_1 and SEG_2 are set by the SEG_1 bit and SEG_2 bit of the BT register.

$$BT = t_{SEG1} + t_{SEG2} = ((SEG_1+2) + (SEG_2+1)) \times TQ$$

Table 30-2 CAN Bit Time Setting Rules

classifier for honorific people	Setting range	rules and regulations
SEG_1 bit of the BT register	0~63	SEG_1≥SEG_2+1 SEG_2≥SJW
SEG_2 bit of the BT register	0~7	
SJW bit of the BT register	0~7	

30.4.2 transmitter buffer

CAN_CTRL provides two transmit buffers for sending data, the primary transmit data buffer PTB and the secondary transmit buffer STB. PTB has the highest priority but can only buffer one frame of data, STB has a lower priority than PTB but can buffer 4 frames of data, and the 4 frames of data in STB can work in FIFO mode or priority arbitration mode.

All 4 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 data written first is sent first, and in the priority mode, the data with the small ID is sent first.

The data in the PTB has the highest priority, so PTB transmissions can defer STB transmissions, but STBs that have already won arbitration and started transmitting cannot be deferred by PTB transmissions.

A frame of data in PTB and STB takes up 4 words and can be accessed through TBUF register. The PTB or STB is selected by the TBSEL bit of TCMD register, TBSEL=0, PTB is selected, TBSEL=1, STB is selected, and the next SLOT in STB is selected by the TSNEXT bit of TCTRL register. the correspondence is shown below:

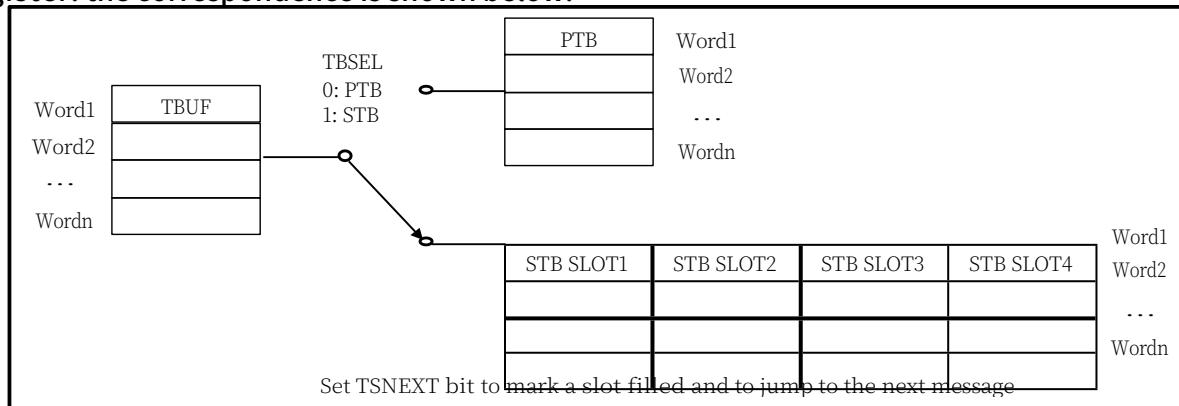


Figure 30-3 CAN TBUF Register Write Transmit Buffer and Diagram

30.4.3 receiver buffer

CAN_CTRL provides 10 SLOTS of receive buffers to store the received data, and the 10 SLOTS

received data through the RBUF register, which is always the first one to be read.

Reads the earliest received data and releases the already read setting RREL

RCTRL register to 1 and next RB SLOT.

RBSLOT by
of the
points to the

The schematic for reading RB SLOT via RBUF is as follows.

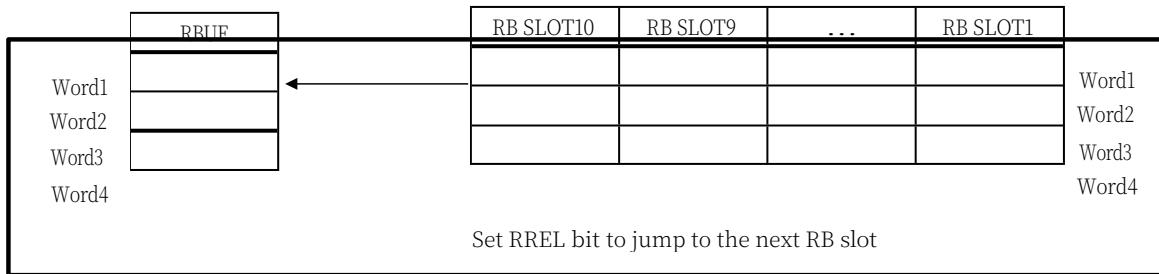


Figure 30-4 CAN RBUF Register Read Receive Buffer Diagram

30.4.4 Receive Filter Register Set

CAN_CTRL provides 8 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 bits to compare. The ID CODE register is used to compare the received CAN IDs and the ID MASK register is used to select the CAN ID bits for comparison. When the corresponding ID MASK bit is 1, the ID CODE of that bit is not compared.

The received data is received as long as it passes any of the 8 groups of filters and the received data is stored in the RB, otherwise the data is not received and not stored.

Each filter group is enabled or disabled by the ACFEN register. ID CODE and ID MASK are set by the SELMASK bit of the ACFCTRL register, SELMASK=0 points to ID CODE, SELMASK=1 points to ID MASK. The filters are selected by the ACFADR bit of the ACFCTRL register. CODE and ID MASK are accessed through the ACF register and can only be set when CFG_STAT.RESET=1, i.e., CAN software reset. Refer to the following figure for how the ACF register accesses the filter register set.

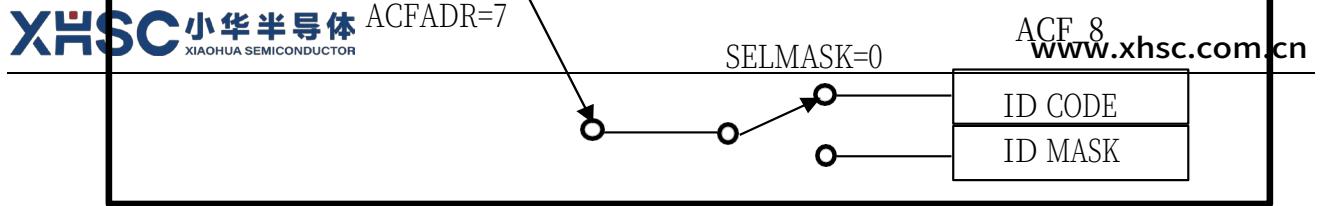


Figure 30-5 CAN ACF Register Access Filter Group Diagram

30.4.5 data transmission

You must ensure that the data to be sent by the PTB or STB has been filled before starting the PTB or STB transmission. No refilling of data is allowed during transmission.

The procedure for sending data settings is as follows:

1. Setting TBSEL Selects the transmit BUFSLOT from PTB and STB.
2. Write the data to be sent via the TBUF register
3. If STB is selected, set TSNEXT=1 to complete loading of all STB SLOTS
4. transmit enable
 - PTB Sending using TPE
 - STB sends using TSALL or TSONE.
5. Send completion status confirmation
 - PTB sends completion using TPIF, TPIE used to enable TPIF
 - STB uses TSONE to send completion with TSIF and TSIE to enable TSIF.
 - TSIF is used when the STB is finished sending by TSALL, and TSIF is set when all the STBSLOT data to be sent are used to enable TSIF.

30.4.6 single data transmission

When the automatic retransmission function is not required, it can be set to the single transmission mode through the registers. The TPSS bit of the CFG_STAT register is used to set the single transmission mode of the PTB, and the TSSS bit is used to set the single transmission mode of the STB. When the data is successfully sent, the action is the same as that in the normal transmission mode. However, the following result occurs when the data is not sent successfully:

- TPIF is set (TPIE=1) and the corresponding BUF SLOT data is cleared.
- When an error is sent, KOER is updated and BEIF is set (BEIE=1)
- Arbitration fails and ALIF is set (ALIE=1)

In single transmission mode, you cannot rely on TPIF alone to judge the completion of transmission, you need to judge the completion of transmission together with BEIF and ALIF.

30.4.7 Cancel data transmission

Data sends that have been requested but have not yet been executed can be canceled via TPA or TSA. Cancellation of data sending occurs in the following cases:

- arbitration
 - If the node fails to arbitrate, the data transmission is canceled.

- The node arbitrates successfully, then the transmission continues.
- Data transmission in progress
 - Data is successfully sent and ACK is received, the corresponding flag and status are set normally. Data transmission is not canceled.
 - Successful data transmission but no ACK received, data transmission canceled, error counter increased.

- TSALL=1 Transmission data set, STB transmitted normally, and STB SLOT that has not started transmission is canceled.

Cancelling a data transmission results in one of the following two scenarios:

- TPA releases PTB and makes TPE=0.
- For TSONE-enabled transmissions, the TSA releases one STB SLOT; for TSALL-enabled transmissions, the TSA releases all STB SLOTs.

30.4.8 data reception

The Receive Filter group filters out unwanted receive data, reducing the occurrence of interrupts and RB reads, thus reducing the CPU load. The receive data setting procedure is as follows:

1. Sets the filter group.
2. Sets RFIE, RAFIE and AFWL.
3. Wait for RFIF or RAIF to be set.
4. Reads the earliest received data from the RB FIFO via RBUF.
5. Set RREL=1 to select the next RB SLOT.
6. Repeat 4,5 until RB is confirmed empty by RSTAT.

30.4.9 error handling

On the one hand, CAN_CTRL performs automatic processing, including automatic message retransmission and automatic deletion of received messages with errors, and on the other hand reports the errors to the CPU via an interrupt.

The CAN node has the following three error states:

- Error active: nodes automatically send active error frames when they detect an error.
- Error passive: nodes automatically send passive error frames when they detect an error.
- Node off: In the off state this node no longer affects the entire CAN network.

CAN_CTRL provides two counters, TECNT and RECNT, for counting errors. the TECNT and RECNT counters are incremented and decremented according to the rules specified in the CAN2.0B protocol. A programmable CAN error warning LIMIT register is also provided to generate an error interrupt to notify the CPU.

There are five types of errors during CAN communication as follows, and the error types can be recognized by the KOER bit in the EALCAP register.

- bit error
- formal error
- padding error
- response error

■ CRC error

30.4.10 Node shutdown

When the Transmit Error Count (TECNT) is greater than 255, the CAN node automatically enters the node shutdown state and the node no longer participates in CAN communication until it returns to the error active state. The CAN node shutdown state can be confirmed by the BUSOFF bit in the CFG_STAT register. An EIF interrupt is generated while BUSOFF is set.

CAN There are two ways to recover from a node shutdown state to an error active state:

- Power-on reset
- Received 128 11-bit invisible bit sequences (recovery sequences)

The TECNT value remains unchanged in the node off state and RECNT is used to count the recovery sequence. Upon recovery from the node off state, TECNT and RECNT are reset to 0.

The RESET bit of the CFG_STAT register is set at the same time that the node shutdown flag BUSOFF is set.

30.4.11 Arbitration Failure Location Capture

CAN_CTRL is able to accurately capture the position of the arbitration failure bit and reflect it in the ALC register. the ALC register holds the position of the most recent arbitration failure bit, and the ALC bit is not updated if the node wins the arbitration.

ALC values are 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. Because arbitration occurs only within the arbitration field, the maximum value of ALC is 31. For example, if a Standard Format Remote Frame is arbitrated with an Extended Frame, and the Extended Frame fails at the IDE bit, ALC = 12.

30.4.12 loopback mode

CAN_CTRL supports the following two loopback modes:

- inner loop
- external loop

Both loopback modes can receive their own outgoing data frames and are mainly used for testing purposes.

In internal loopback mode, the module internally connects the receive data line to the transmit data line and transmit data is not output. In internal loopback mode, the node generates a self-answer signal to avoid ACK errors.

The external loopback mode maintains the connection to the transceiver so that the sent data is still present on the CAN bus and CAN_CTRL receives its own sent data with the help of the transceiver. The external loopback mode can be determined by the SACK bit in the RCTRL

register to determine whether a self-answer signal is generated or not; when SACK=0, no self-answer signal is generated, and when SACK=1, a self-answer signal is generated.

In external loopback mode with SACK=0, sending a frame of data results in the following two conditions:

- Other nodes also receive the data frame sent by this node and send an answer signal, in this case this node can successfully send and receive data.
- If no other node returns an answer signal, an answer error is generated, which resends the data and increases the error counter. The single send mode is recommended when it is not known if there are other nodes on the bus.

The return from loopback mode to normal mode should be achieved by resetting CAN_CTRL by setting RESET.

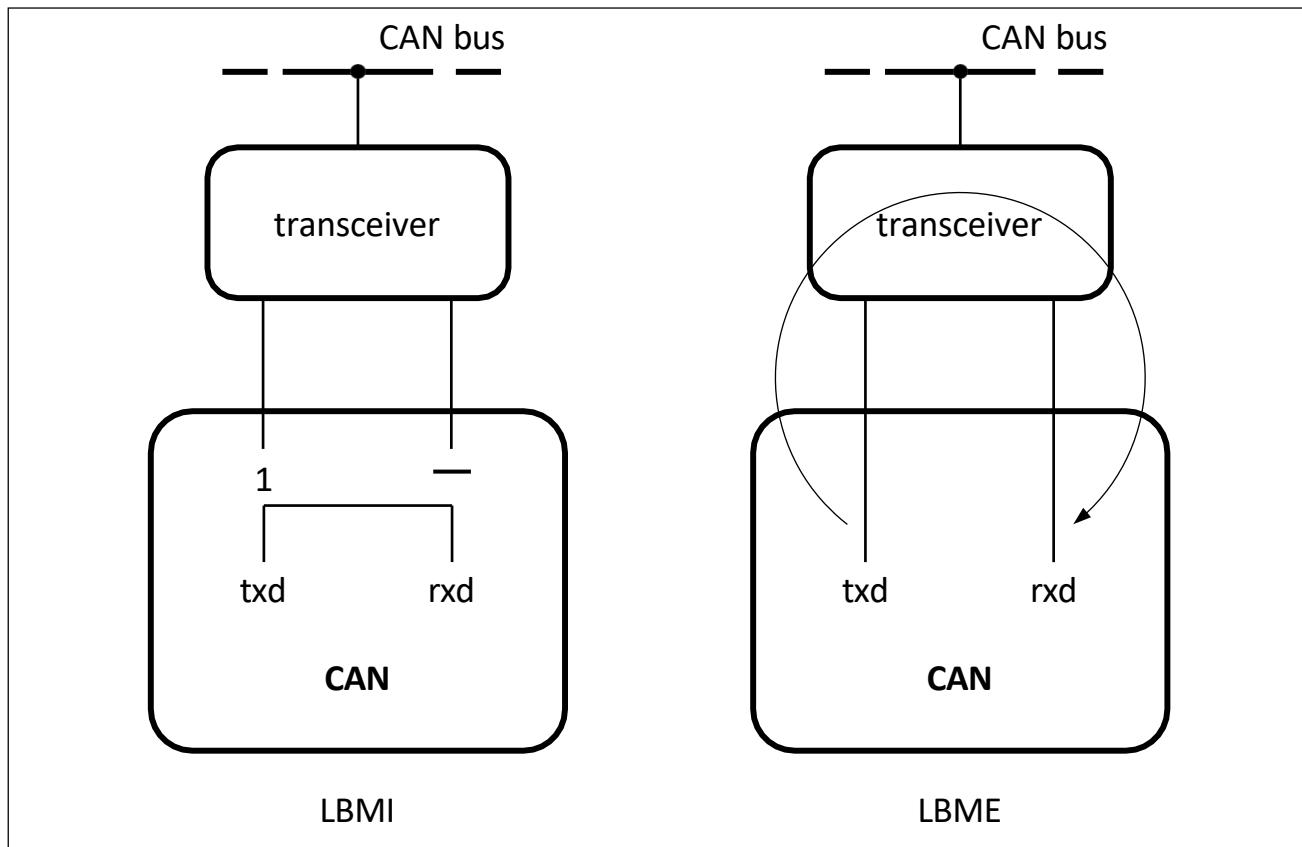


Figure 30-6 CAN LBMI and LBME Schematics

30.4.13 silent mode

Silent mode can be used to listen for CAN network data. In silent mode, you can receive data from the CAN bus and do not send any data to the bus. Set LOM in TCMD register to 1 to put the CAN bus controller into silent mode, and clear it to 0 to exit silent mode.

External Loopback Mode can be combined with Silent Mode to form External Loopback Silent Mode, where the CAN can be considered a silent receiver, but can send data when necessary. In External Loopback Silent Mode, frames containing self-answering signals are allowed to be sent, but the node does not generate error frames and overload frames.

30.4.14 Software reset function

The software reset function is realized by setting the RESET bit of register CFG_STAT register to 1. The reset range of the software reset function is shown in the table below.

Table 30-3 Software Reset Range

processor register celebrity	hardw are reset (a dislocated joint, an electro nic device etc)	note	processor register celebrity	hard ware reset (a dislocated joint, an electr onic devic e etc)	note
ACFADR	clogged	–	EWL	be	–
ACODE	clogged	Can only be written during a software reset	KOER	be	–
AE_X	clogged	–	LBME	be	–
AFWL	clogged	–	LBMI	be	–
AIF	be	–	LOM	clogge d	–
ALC	be	–	RACTIVE	be	Receiving stops immediately and no ACK is generated
ALIE	clogged	–	RAFIE	clogge d	–
ALIF	be	–	RAFIF	be	–
AMASK	clogged	Can only be written during a software reset	RBALL	be	–
BEIE	clogged	–	RBUF	be	All RBs are labeled as empty with variable values
BEIF	be	–	RECNT	clogge d	Write clear by BUSOFF
BUSOFF	clogged	Clear by writing 1	REF_ID	clogge d	–
EIEF	clogged	–	REF_IDE	clogge d	–
EIF	clogged	–	RFIE	clogge d	–
EPASS	clogged	–	RFIF	be	–
EPIE	clogged	–	RIE	clogge d	–

				d	
EPIF	be	-	RIF	be	-
EWARN	clogged	-	ROIE	clogge d	-

processor register celebrity	hardware reset (a dislocated joint, an electronic device etc)	note	processor register celebrity	hardware reset (a dislocated joint, an electronic device etc)	note
ROIF	be	-	TSMODE	clogged	-
ROM	clogged	-	TSNEXT	be	-
ROV	be	-	TSONE	be	-
RREL	be	-	TPIE	clogged	-
PRESC	clogged	Can only be written during a software reset	TPIF	be	-
RSTAT	be	-	TPSS	be	-
SACK	be	-	TSFF	be	All STB SLOTs are marked as empty
SELMASK	clogged	-	TSIE	clogged	-
SEG_1	clogged	Can only be written during a software reset	TSIF	be	-
SEG_2	clogged	Can only be written during a software reset	TSSS	be	-
SJW	clogged	Can only be written during a software reset	TSSTAT	be	All STB SLOTs are marked as empty
TACTIVE	be	Sending stops immediately	TTEN	be	-
TBE	be	-	TTIF	be	-
TBF	clogged	-	TTIE	clogged	-
TBPTR	clogged	-	TTPTR	clogged	-
TBSEL	be	-	TTTBM	clogged	-
TBUF	be	All STBs are marked as null, pointing to PTBs	TTTYPE	clogged	-
TECNT	clogged	Cleared by BUSOFF=1	TT_TRIG	clogged	-
TEIF	be	-	TT_WTRIG	clogged	-
TPA	be	-	T_PRESC	clogged	-

TPE	be	-	WTIE	clogged	-
TSA	be	-	WTIF	be	
TSALL	be	-			

30.4.15 Upwardly compatible with CAN-FD functions

CAN-CTRL Even if CAN-FD frames are received in a network containing CAN-FD, the receiver automatically ignores these frames, does not return an ACK, and waits until the bus is free to send or receive the next CAN2.0B frame.

30.4.16 Time Trigger TTCAN

CAN-CTRL provides partial (lever 1) hardware support for the time-triggered communication method specified in ISO11898-4. This section describes the TTCAN functions in the following five sections.

30.4.16.1 TBUF behavior in TTCAN

mode TTTBM=1

With TTTBM=1, the PTB and STB SLOTS form a TB SLOT as well. the BUF is specified to be sent through the TBPTR register, where TBPTR=0 points to the PTB, and TBPTR=1 points to the STB SLOT1, and so on. The host can mark the transmit BUF SLOT with the TPE and TPF registers. the TBSEL and TSNEXT registers are meaningless and can be ignored.

With TTTBM=1, the PTB does not have any special attributes, and like STB SLOT, the transmission completion flag uses TSIF.

When in TTCAN mode, there is no FIFO mode or priority arbitration mode for sending BUFS, and there is also only one selected SLOT to which data can be sent.

In TTCAN mode, the transmission start is required to be time-triggered and TPE, TSONE, TSALL, TPSS and TPA are fixed to 0 and ignored.

TTTBMM=0

When TTTBM=0, the event-driven communication and receive timestamp functions are used in combination. In this mode, the functions of the PTB and STB are the same as when TTEN=0. Therefore, the PTB always has the highest priority, and the STB can work in FIFO mode or arbitration mode.

30.4.16.2 TTCAN Functional Description

After power-up, the Time Master needs to be initialized according to the ISO 11898-4 protocol. There can be up to 8 potential Time Masters in a CAN network, each with its own reference message ID (last 3 digits of the ID). These potential Time Masters send their own reference messages according to their priority.

After TTEN=1, the 16-bit counter starts to work, and when the reference message is successfully received or the Time Master successfully sends the reference message, the CAN controller copies the Sync_Mark to the Ref_Mark, which sets the cycle time to 0. Successfully receiving the reference message sets the RIF flag and successfully sending

the reference message sets the TPIF flag or TSIF flag. TPIF flag or TSIF flag. At this point the host needs to prepare the trigger condition for the next action.

The trigger condition may be a receive trigger. This trigger only triggers an interrupt that can be used to detect that the expected message was not received.

The trigger condition can also be a send trigger. This trigger starts sending the data in the TBUF SLOT specified through the TTPTTR register. If the selected TBUF SLOT is marked as empty, sending is not started but the interrupt flag is set.

30.4.16.3 TTCAN Timing

CAN_CTRL supports ISO11898-4 level 1. contains a 16-bit counter operating at the bit times defined for PRESC, SEG_1, SET_2. If TTEN=1, there is an additional prescaler T_PRESC.

The value of the counter is Sync_Mark for an SOF frame, and if the frame is a reference message, the Sync_Mark is copied to the Ref_Mark. cycle time is equal to the value of the counter minus the Ref_Mark, which is used as a timestamp for a received message or as a reference for the trigger time of a sent message.

30.4.16.4 TTCAN Trigger Method

The TTCAN trigger method is defined through the TTYPE register, the TTPTTR register specifies the send SLOT, and TT_TRIG specifies the trigger cycle time.

The following five triggers are included:

- immediate trigger
- time trigger
- single-send trigger
- Send Start Trigger
- Send Stop Trigger

All triggers use the TTIF flag except for the immediate trigger method. with TTTBM=1, only the time trigger method is supported.

immediate trigger

The trigger is initiated by writing high of TT_TRIG (not caring about the value written) In this mode, the data in the TBUF SLOT selected by TTPTTR is sent immediately. tTIF is not set.

time trigger

The time-triggered method generates an interrupt only by setting the TTIF flag and has no other function. The time-triggered method can be used if a node expects to receive the expected data within a specific time window. If the TT_TRIG value is less than the actual cycle time, TEIF is set and no other action is taken.

single-send trigger

The single send trigger method is used to send data within the execution time window. In this case, the TSSS bit is ignored and the state remains unchanged.

The maximum 16 cycle time ticks specified in ISO11898-4 are set by the TEW bit in the range of 1 to 16. If the data is not started within the specified transmit enable time window, the frame is discarded. If AIE is set, then AIF is set and the corresponding transmit BUF SLOT is marked as empty, but the data in the corresponding transmit BUF will not be rewritten, so if the same data is sent next time, it is only necessary to mark SLOT as filled again, i.e., to send it again by setting TBF.

If the TT_TRIG value is less than the actual cycle time, TEIF is set and no other action is taken.

Send Start Trigger

The Send Start Trigger method is used within the arbitration time window to participate in arbitration. the TSSS is used to determine whether to automatically retransmit or to send in single send mode. The Send Stop Trigger can be used to stop a specified message from being sent if it is not successfully sent.

If the TT_TRIG value is less than the actual cycle time, TEIF is set and no other action is taken.

Send Stop Trigger

The transmit stop trigger is used to stop a transmission that has been started by the transmit start trigger. If the transmission is stopped, the frame is discarded and AIF is set if AIE is set. the corresponding transmit BUF SLOT is marked as empty, but the data in the corresponding transmit BUF is not rewritten, so that if the same data is sent next time, it is only necessary to mark SLOT as filled again, i.e. to send it again by setting TBF.

If the TT_TRIG value is less than the actual cycle time then TEIF is set and execution stops.

30.4.16.5 TTCAN Trigger Watch Time

The TTCAN Trigger Watchdog Time function is similar to the Watchdog function and is used when TTTBM=1. It is used to watch the watchdog for the time since the last reference message was successfully received. The reference message can be received during the cycle time or after an event, the application program should set the appropriate watchdog time according to the specific situation.

If cycle count is equal to TT_WTRIG, set WTIF. write 0 through WTIE to turn off the watchdog trigger. If TT_WTRIG is less than the actual cycle time, TEIF is set.

30.4.17 disr**upti****Table 30-4 CAN Interrupt Table****ons**

disruption flag	descriptive
RIF	receive interruptions
ROIF	Receive overflow interrupt
RFIF	Receive BUF full interrupt
RAFIF	Receive BUF will be full interrupt
TPIF	PTB transmit interrupt
TSIF	STB transmit interrupt
EIF	false interruption
AIF	Cancel send interrupt
EPIE	false passive interrupt
ALIF	Arbitration failure interruption
BEIF	Bus Error Interrupt
WTIF	Trigger Watchdog Interrupt
TEIF	Trigger error interrupt
TTIF	time-triggered interrupt

30.5 Register Description

CAN_BASE_ADDR:0x40070400

Table 30-5 CAN Register List

register name	notation	offset address	bit width	reset value
CAN receive BUF register	CAN_RBUF	0x00~0x0F	128	0xFFFF XXXX
CAN transmit BUF register	CAN_TBUF	0x50~0x5F	128	0xFFFF XXXX
CAN configuration and status registers	CAN_CFG_STAT	0xA0	8	0x80
CAN Command Register	CAN_TCMD	0xA1	8	0x00
CAN Transmit Control Register	CAN_TCTRL	0xA2	8	0x90
CAN Receive Control Register	CAN_RCTRL	0xA3	8	0x00
CAN receive and transmit interrupt enable registers	CAN_RTIE	0xA4	8	0xFE
CAN receive and transmit interrupt flag registers	CAN_RTIF	0xA5	8	0x00
CAN Error Interrupt Enable and Flag Registers	CAN_ERRINT	0xA6	8	0x00
CAN Warning Qualifier Register	CAN_LIMIT	0xA7	8	0x1B
CAN Bit Timing Register	CAN_BT	0xA8	32	0x0102 0203
CAN Error and Arbitration Failure Capture Registers	CAN_EALCAP	0xB0	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 Registers	CAN_ACFEN	0xB6	8	0x01
CAN filter group code and mask registers	CAN_ACF	0xB8	32	0xFFFF XXXX
TTCAN TB slot pointer register	CAN_TBSLOT	0xBE	8	0x00
TTCAN Time Trigger Configuration Register	CAN_TTCFG	0xBF	8	0x90
TTCAN Reference Message Register	CAN_REF_MSG	0xC0	32	0x0000 0000
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

Table 30-6 CAN Register BYTE/HALFWORD/WORD Access Arrangement Table

address	BYTE access	HALFWORD access	WORD access
HC32F460_F45x_A460 Series Reference			1083
Manual_Rev1.6			/1407

s							
0x00~0x0F	CAN_RBUF	CAN_RBUF		CAN_RBUF			
0x50~0x5F	CAN_TBUF	CAN_TBUF		CAN_TBUF			
0xA0	CAN_CFG_STAT	CAN_TCM_D	CAN_CFG_SSTAT	CAN_RCTLRL	CAN_TCTRRL	CAN_TCM_D	CAN_CFG_STAT

address	BYTE access	HALFWORD access		WORD access					
0xA1	CAN_TCMD	-		-					
0xA2	CAN_TCTRL	CAN_RCTL RL	CAN_TCTRL	-					
0xA3	CAN_RCTRL	-		-					
0xA4	CAN_RTIE	CAN_RTIF	CAN_RTIE	CAN_LIMIT	CAN_ERRINT	CAN_RTIF	CAN_RTIE		
0xA5	CAN_RTIF			-					
0xA6	CAN_ERRINT	CAN_LIMIT	CAN_ERRINT	-					
0xA7	CAN_LIMIT			-					
0xA8	CAN_BT[7:0]	CAN_BT[15:0]		CAN_BT					
0xA9	CAN_BT[15:8]	-		-					
0xAA	CAN_BT[23:16]	CAN_BT[31:16]		-					
0xAB	CAN_BT[31:24]	-							
0xB0	CAN_EALCAP	-		CAN_TECNT	CAN_RECNT	CAN_EALCAP			
0xB1	-	-		-					
0xB2	CAN_RECNT	CAN_TECNT	CAN_RECNT	-					
0xB3	CAN_TECNT	-		-					
0xB4	CAN_ACFCTRL	CAN_ACFC TRL		CAN_ACFE N		CAN_AC FCTRL			
0xB5	CAN_ACFCTRL[1 5:8]	-		-					
0xB6	CAN_ACFEN[7:0]	CAN_ACFEN		-					
0xB7	CAN_ACFEN[15. 8]	-		-					
0xB8	CAN_ACF	CAN_ACF		CAN_ACF					
0xBC	-	-		CAN_TTC FG	CAN_TBSL OT	-			
0xBD	-	-		-					
0xBE	CAN_TBSLOT	CAN_TTC FG	CAN_TBSLO T	-					
0xBF	CAN_TTCFG	-		-					
0xC0	CAN_REF_MSG[7 :0]	CAN_REF_MSG[15:0]		CAN_REF_MSG					
0xC1	CAN_REF_MSG[1			-					

address	BYTE access	HALFWORD access	WORD access	
	5:8]			
0xC2	CAN_REF_MSG[2:16]	CAN_REF_MSG[31:16]	-	
0xC3	CAN_REF_MSG[3:24]	-	-	
0xC4	CAN_TRG_CFG[7:0]	CAN_TRG_CFG	CAN_TT_TRIG	CAN_TRG_CFG
0xC5	CAN_TRG_CFG[1:8]	-	-	
0xC6	CAN_TT_TRIG[7:0]	CAN_TT_TRIG	-	
0xC7	CAN_TT_TRIG[1:8]	-	-	
0xC8	CAN_TT_WTRIG[7:0]	CAN_TT_WTRIG		CAN_TT_WTRIG
0xC9	CAN_TT_WTRIG[15:8]			

30.5.1 CAN receive BUF register (CAN_RBUF)

CAN Receive Buffer Registers

Offset Address: 0x00

Reset value: 0xXXXX XXXX

RBUF register points to the RB

SLOT address of the earliest CAN mailbox

received, and the RBUF register can be read in any order. KOERbit is the register EALCAP.KOER, and is only significant when RBALL=1.

The TX bit indicates that a message sent by itself was received in loopback mode.

The CYCLE_TIME bit is valid only in TTCAN mode and indicates the cycle time at the start of the SOF. The data format of the CAN receive mailbox is as follows:

Table 30-7 Standard Format CAN Receive Mailbox Format

Table 30-8 Extended Format CAN Receive Mailbox Format

address	b7	b6	b5	b4	b3	b2	b1	b0	functionality	
RBUF	ID[7:0]								ID	
RBUF+1	ID[15:8]								ID	
RBUF+2	ID [23:16]								ID	
RBUF+3	-			ID [28:24]					ID	
RBUF+4	IDE=1	RTR	0	0	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	
RBUF+11	DATA4								Data	
RBUF+12	DATA5								Data	
RBUF+13	DATA6								Data	
RBUF+14	DATA7								Data	
RBUF+15	DATA8								Data	

The control bits have the following meanings: IDE
(IDentifier Extension).
 0: Standard format
 1: Extended format

RTR(Remote Transmission Request) 0:
data frame
 1: Remote frames

DLC (Data Length Code).

Data length code, the setting range is 0~8, the corresponding data length is 0Byte~8Byte.

30.5.2 CAN Transmit BUF Register (CAN_TBUF)

CAN Transmit Buffer Registers

Offset Address: 0x50

Reset value: 0xXXXX XXXX

TBUF register points to the next empty CAN transmit BUF SLOT, and the TBUF registers can be written in any order. corresponding TBUF SLOT is marked as having written data by software by writing 1 to TSNEXT, thus pointing to the next TBUF SLOT.

TBUF can only be accessed by WORD.

The data format of the CAN send mailbox is as follows:

Table 30-9 Standard Format CAN Send Mailbox Format

address	b7	b6	b5	b4	b3	b2	b1	b0	functionalit y
TBUF	ID[7:0]								ID
TBUF+1	-				ID[10:8]				ID
TBUF+2	-								ID
TBUF+3	-								ID
TBUF+4	IDE=0	RTR	0	0	DLC[3:0]				Control
TBUF+5	-								-
TBUF+6	-								-
TBUF+7	-								-
TBUF+8	DATA1								Data
TBUF+9	DATA2								Data
TBUF+10	DATA3								Data
TBUF+11	DATA4								Data
TBUF+12	DATA5								Data
TBUF+13	DATA6								Data
TBUF+14	DATA7								Data
TBUF+15	DATA8								Data

Table 30-10 Extended Format CAN Send Mailbox Format

address	b7	b6	b5	b4	b3	b2	b1	b0	functionalit
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=1	RTR	0	0	DLC[3:0]				Control
TBUF+5	-								-
TBUF+6	-								-
TBUF+7	-								-
TBUF+8	DATA1								Data
TBUF+9	DATA2								Data
TBUF+10	DATA3								Data
TBUF+11	DATA4								Data
TBUF+12	DATA5								Data
TBUF+13	DATA6								Data
TBUF+14	DATA7								Data
TBUF+15	DATA8								Data

The control bits have the following meanings: IDE (IDentifier Extension).
 0: Standard format
 1: Extended format

RTR(Remote Transmission Request) 0:

Data frame

1: Remote frames

DLC (Data Length Code).

Data length code, the setting range is 0~8, the corresponding data length is 0Byte~8Byte.

30.5.3 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

cla ssi fie r for ho no rifi c pe op le	marking	Bit Name Function Read/Write	
b7	RESET	Reset request	Reset request bit 0: Local reset not requested 1: Request for local reset Some registers can only be written when RESET=1, please refer to the software reset function, when the node enters the BUS OFF state, the hardware automatically sets the RESET position to 1. Note that when RESET=0 it is necessary to 11 CAN bit time for this node to participate in communication.
b6	LBME	External loopback mode enable bit Disable external loopback mode	External Loopback Mode Enable Bit 0: 1: Enable external loopback mode Note: Setting this bit is prohibited in communication.
b5	LBMI	Internal loopback mode enable bit Disable internal loopback mode	Internal Loopback Mode Enable Bit 0: 1: Enable internal loopback mode Note: Setting this bit is prohibited in communication.
b4	TPSS	PTB single transmission mode single transmission mode disabled	PTB single transmission mode single transmission mode disabled R/W 1: Enable PTB single transmission mode
b3	TSSS	STB single transmission mode single transmission mode	STB single transmission mode 0: STB 1: Enable STB single transmission mode

	disabled	R/W	
		1: Enable STB single transmission mode	
b2	RACTIVE	In-Reception Status Signal Receiving status signal 0: not receiving 1: Receiving	R
b1	TACTIVE	Sending Status Signal Transmitting status signal 0: not transmitting 1: Sending	R
b0	BUSOFF	Bus Off State Bus off state 0: Bus active state 1: Bus off state Note: Writing 1 clears the TECNT and RECNT registers for debugging purposes only.	R/W

30.5.4 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

class	marking	Bit Name	Function	Read/Write
ifier for hon orifi c peo ple				
		Transmit Buffer Select bit 0: PTB		
		Transmit BUF select bit		1: STB
b7	TBSEL	R/W		
		When TTEN=1 & TTTBM=1, TBSEL is reset to the reset value.		
		Note: This bit needs to be held constant when writing the TBUF register or the TSNEXT bit.		
b6	LOM	Silent Mode Enable Bit	Silent Mode Enable Bit (Listen Only) Mode 0: Silent mode disabled 1: Enable silent mode	R/W
		Transmission is disabled when LOM=1 & LBME=0. LOM=1&LBME=1 prohibits answering the corresponding received frames		
b4	TPE	PTB Transmit Enable Bit	PTB transmit enable bit (Transmit Primary) The TPE is reset to the reset value by hardware in the following cases: Enable) 0: Disable PTB transmission	This bit
b5	Reserved	well as error frames, but data can be sent. Note: Setting this bit is will remain 1 after a write 1 until the PTB transmission is completed or canceled by the TPA. The software cannot prohibited during communication. - The reset value must be held.		R/W
		Except for that bit. - RESET=1 1: Enable PTB transmission - BUSOFF=1 When this bit is enabled, the Mailbox in the PTB will be sent at the next available position. - LOM=1&LBME=0 STB already started - TTEN=1&TTTB=1 Transmission will continue, but the next waiting STB transmission will be delayed until the PTB transmission is complete. Abort) 0: not canceled		
b3	TPA	PTB transmit cancel bit	1: Cancel PTB transmissions that have been requested by TPE Set 1 but have not yet started This bit is written 1 by software but cleared by hardware. The TPE bit can be cleared by writing a 1, so it should not be used in the same way as the TPE. Set to 1 when	R/W
		The TPA is reset to the reset value by hardware in the following cases: - RESET=1 - BUSOFF=1 - TTEN=1&TTTB=1		
b2	TSONE	Send a frame of STB data to set the position	Transmit Secondary ONE frame 0: No transmission 1: Send a frame of STB data	R/W

The earliest written data is sent in FIFO mode, and the highest priority

data is sent in Priority mode This bit will remain 1 after a 1 is written until the STB transmission is completed or the transmission is canceled by TSA. Software cannot clear this bit by writing 0.

The following cases TSONE is reset to the reset value by hardware:

- RESET=1
- BUOFF=1
- LOM=1&LBME=0
- TTEN=1&TTTBM=1

Transmit Secondary ALL frame 0: No transmit

1: Send all data in the STB

b1 TSALL Send all STB data
locate
A write 1 to this bit will hold it to 1 until the STB transmission is complete or the transmission is canceled via TSA. Software cannot clear this bit by writing a 0. R/W

The following cases TSALL is reset to the reset value by hardware:

- RESET=1
- BUOFF=1
- LOM=1&LBME=0
- TTEN=1&TTTBM=1

STB Transmit Cancel Bit (Transmit Secondary

Abort) 0: Not canceled

b0 TSASTB transmit
cancel bit
1: Cancel the STB transmission that has been requested by TSONE or
TSALL set to 1 but has not yet started. R/W

This bit is written 1 by software but cleared by hardware. Writing a 1 clears the TSONE or TSALL bit. Therefore it should not be set to 1 at the same time as TSONE or TSALL.

The TSA is reset to the reset value by hardware in the following cases:

- RESET=1
- BUOFF=1

30.5.5 CAN Transmit Control Register (CAN_TCTRL)

CAN Transmit Control Register

Offset Address: 0xA2

Reset value: 0x90

b7	b6	b5	b4	b3	b2	b1	b0
-	TSNEXT	TSMODE	TTTBM	-	-	TSSTAT[1:0]	

classifier for honorific people	marking	celebrity	functionality	fill out or in (infor matio n on a form)
---------------------------------------	---------	-----------	---------------	---

b7	Reserved	-	The reset value must be maintained.	R
b6	TSNEXT	Next STB SLOT	<p>Next STB (Transmit buffer Secondary NEXT)</p> <p>0: no action</p> <p>1: Current STB SLOT filled, pointing to next SLOT</p> <p>After the application program finishes writing the data in the TBUF, it identifies the current STB by setting the TSNEXT bit</p> <p>The SLOT has been populated so that the hardware points the TBUF to the next STB SLOT.</p> <p>Data in the STB SLOT identified by the TSNEXT bit can be sent via the TSONE or TSALL bit.</p>	R/W
b5	TSMODE	STB Transmit Mode	<p>Send.</p> <p>This bit is written 1 by the application program and cleared by hardware.</p> <p>After all STB SLOTS are filled, TSNEXT remains at 1 until a STB SLOT is released.</p> <p>It is possible to set both TSNEXT and TSONE or TSALL in a single write access.</p> <p>If TBSEL=0, setting TSNEXT is meaningless. In this case, TSNEXT will be ignored</p> <p>It is skimmed and automatically erased. It does not have any impact.</p> <p>Note: TSNEXT has no meaning in TTCAN mode and is fixed to 0.</p>	R/W
b4	TTTBM	TTCAN BUF model	<p>STB transmit mode (Transmit buffer Secondary operation) MODE)</p> <p>0: FIFO mode</p> <p>1: Prioritization mode</p> <p>FIFO mode sends data frames in the order in which they are written.</p> <p>Priority mode is automatically judged based on ID, the smaller the ID, the higher the priority</p> <p>The higher the priority, the closer to full</p> <p>Regardless of the ID, the PTB has the highest priority.</p> <p>Note: The TSMODE bit can only be set when STB is empty.</p>	R/W
			<p>TTCAN BUF Mode (TTCAN Transmit Buffer Mode)</p> <p>The TTTBM is ignored when TTEN=0. The following definition is used when TTEN=1:</p> <p>0: TSMODE decision, PTB and STB</p> <p>1: Setting by TBPTR and TTPTR</p> <p>In TTCAN mode, this bit can be set to 0 when only the time stamp function is required to be received by the TSMODE decides whether to use PTB or STB.</p> <p>Note: The TTTBM bit can only be set when TBUF is empty.</p>	R/W

TTEN=1 and
TTTBM=1 00: PTB and
STB empty
01: PTB and STB non-full
10: Reservations
11: PTB and STB full

30.5.6 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]	
classification	marking	celebrity	functionality			fill out or in (information on a form)	
rfor							
honori							
c							
people							
b7	SACK	self-responsive		Self-ACKnowledge (Self-ACKnowledge)			
			0: No self-response			R/W	
			1: Enable self-answer function when LBME=1				
b6	ROM	Receive BUF overflow mode set position		Receive buffer Overflow Mode 0: The earliest received data is overwritten.		R/W	
			1: Newly received data is not stored				
b5	ROV	Receive BUF overflow flag bit		Receive buffer OVerflow Flag Bit		R	
			0: No top overflow				
			1: Overflow with at least one data loss				
			Clear by writing RREL to 1.				
b4	RREL	Release receive BUF		Release receive BUF (Receive buffer RElease)		R/W	
			0: No release				
			1: Indicates that this receive BUF has already been read and the RBUF register points to the next RB				
b3	RBALL	(Receive BUF data store all D/AW)		Receive Buffer stores ALL data frames.			
frames			data frame	0: Normal mode		data	
				1: Store all data including data with errors.			
b2	Reserved	-		The reset value must be maintained.		R	
b1~b0	RSTAT	Receive BUF status		Receive BUF status (Receive buffer STATus)			
			00: RBUF empty				
			01: RBUF non-null but less than AFWL programmed value			R	
			10: RBUF is greater than or equal to the programmed value of AFWL but not full				
			11: Full (holds this value when overflowing)				

30.5.7 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

classification	marking	celebrity	functionality		fill out or in (information on a form)
error for honor ific people e					
			Receive Interrupt Enable		
b7	RIE	Receive Interrupt Enable	0: Prohibited 1: Enabling		R/W
			Receive Overrun Interrupt Enable		
b6	ROIE	Receive overflow interrupt enable	0: Prohibited 1: Enabling		R/W
			Receive BUF Full Interrupt Enable (RB Full Interrupt Enable)		
b5	RFIE	Receive BUF full interrupt enable	0: Prohibited 1: Enabling		R/W
			Receive BUF will be full interrupt enable (RB Almost Full Interrupt Enable)		
b4	RAFIE	Receive BUF will be full interrupt enable	0: Prohibited 1: Enabling		R/W
			PTB Transmission Primary Interrupt Enable		
b3	TPIE	PTB transmit interrupt enable	0: Prohibited 1: Enabling		R/W
			TTEN=1 and TTTBM=1: TB_SLOT Full Flag (Transmit slot STB Transmission Secondary Interrupt Enable buffer Full Flag) Enable)		
b2	TSIE	STB transmit interrupt enable	0: TB SLOT selected by TBPTR is empty 0: Prohibited 1: TB SLOT populated for TBPTR selection 1: Enabling		R/W
			Error Interrupt Enable		
b1	EIE	Error Interrupt Enable	0: Disable 1: Enabling		R/W
			TTEN=0 or TTTBM=0: STB Full Flag (Transmit Secondary buffer Full Flag) 0: STB SLOT is not fully populated 1: STB SLOT is fully filled		
b0	TSFF	Send BUF full flag			R

30.5.8 CAN Receive and Transmit Interrupt Status 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/Write	
b7	RIF	Receive Interrupt Flag (Receive Interrupt Flag)		
Interrupt Flag		0: No data frame or remote frame received 1: A valid data frame is received or a remote frame is cleared to 0	R/W	
		by writing 1 through the application program.		
b6	ROIF	Receive Overrun Interrupt Flag (Receive Overrun Interrupt Flag) 0: No RBs overwritten (overwrite)		
		1: RB FIFO full	R/W	
		ROIF and RFIF are set to 1 at the same time when overflow occurs.		
		Write 1 to clear 0 through the application program.		
b5	RFIF	Receive BUF Full Interrupt Flag (RB Full Interrupt Flag)		0: RB
		1: RB FIFO full	R/W	
		Write 1 to clear 0 through the application program.		
b4	RAFIF	Receive BUF Almost Full Interrupt Flag (RB Almost Full Interrupt Flag) 0: by an application write 1.		
		1: PTB Transmission Primary Interrupt Flag	R/W	
b3		PTB Transmission Primary Interrupt Flag 0: Number of filled RB SLOTS <= Maximum number of slots completed		
transmit interrupt flag		1: Successful completion of the requested PTB transmission clears 0 by application write 1.	R/W	
		Note: When in TTCAN mode, TPIF is not valid and only the TSIF flag is used		
b2	TSIFSTB	Transmission Secondary Interrupt Flag (STB)		R/W
transmit interrupt flag		0: No STB send completed 1: Requested STB send successfully completed by application program write		
		1: clear 0.		
		Note: In TTCAN mode, TPIF is invalidated and only the TSIF flag is used		
		Error Interrupt Flag (Error Interrupt Flag) 0: The BUSOFF bit has not changed or the error counter value has not changed relative to the error warning value.		
b1	EIF	1: The BUSOFF bit changes, or the error counter value changes relative to the error warning value. For example, the error counter value changes		
Interrupt Flag				

			Write 1 to clear 0 through the application program.	
b0	AIF	Cancel transmit interrupt flag	Abort Interrupt Flag (Abort Interrupt Flag) 1: A transmission requested to be canceled via TPA and TSA has been canceled. Clear 0 by application program write 1.	R/W
			Attention: It is recommended not to set both TPA and TSA at the same time, as they use the same flag bit AIF	

30.5.9 CAN Error Interrupt Enable and Flag 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
<hr/>							
Bit Flag	Bit Name	Function	Read/Write				
b7	EWARN	Set error warning value reached		Error WARNING limit reached 0: Both RECNT and TECNT are less than the EWL set value 1: RECNT or TECNT is greater than or equal to the EWL set value Write 1 to clear 0 through the application program.		R	
b6	EPASS	false passive node		Error Passive mode active 0: Node is an active error node 1: Nodes are passive error nodes		R	
b5	EPIE	Error passive interrupt enable		Error Passive Interrupt Enable (Error Passive Interrupt 0: Disable) 1: Enabling		R/W	
b4	EPIF	Error passive interrupt flag		Error Passive Interrupt Flag 0: No change from error active to error passive or error passive to error active has occurred 1: An error-active to error-passive or error-passive to error-active change OCCURS (if EPIE is 1) by the application program writing 1 to clear 0. (Arbitration Lost Interrupt Enable)		R/W	
b3	ALIE	Arbitration Failure Interrupt Enable		(Arbitration Lost 0: Disable) 1: Enabling		R/W	
b2	ALIF	Arbitration Failure Interruption Flag		Arbitration Lost interrupt flag (Arbitration Lost 0: Arbitration Success) 1: Failure of arbitration Write 1 to clear 0 through the application program.		R/W	
b1	BEIE	Bus error interrupt enable		Bus Error Interrupt Enable 0: disabled 1: Enabling		R/W	
b0	BEIF	Bus error interrupt flag		Bus Error Interrupt Flag 1: Bus error 0: No bus error Write 1 to clear 0 through the application program.		R/W	

30.5.10 CAN Bit Timing Register (CAN_BT)

CAN 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		
PRESC[7:0]								-	-	SJW[6:0]							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0		
-	SEG_2[6:0]								SEG_1[7:0]								

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b24	PRESC	Pre-crossover setting	The prescaler divides the CAN communication clock to obtain the TQ (Time quanta) clock. Effective values 0 to 255, and the prescaler value is S_PRESC+1.	R/W
b23	Reserved	-	The reset value must be maintained.	R
b22~b16	SJW Jump Width Setting	Synchronized Jump Width Setting	Synchronized	R/W
b15	Reserved	-	The reset value must be held.	R Bit
b14~b8	SEG_2	Bit segment 2 time setting segment 2 time setting	Bit segment 2 time = (SEG_2+1)*TQ	R/W
b7~b0	SEG_1	Bit 1 time setting	Bit segment 1 time setting Bit segment 1 time = (SEG_1+2)*TQ	R/W

30.5.11 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/Write	
				Kind Of Error 000: No Error
b7~b5	KOER	Error		001: Bit error 010: Formal error 011: Filling error 100: Response error 101: CRC error 110: Other errors 111: Reservations
Category				The KOER bit is updated when there is a new error, and the KOER bit remains unchanged on successful transmission or reception.
b4~b0	ALC	Arbitration Lost Capture		Arbitration Lost Capture On arbitration failure, ALC records the position of the bit that failed arbitration.
				R

30.5.12 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]			
<hr/>							
classifier	marking	Bit Name Function Read/Write					
for							
honorific							
people							
Receive buffer Almost Full Warning							
b7~b4	AFWL	Receive BUF will be full warning value					(Limit)
		R/W					
		The corresponding warning					
		value ranges to 10.					
		AFWL=8, meaningless. This is treated as AFWL=1.					
Programmable Error Warning Limit							
b3~b0	EWL	Programmable Error warning value					Error
		Warning Value = (EWL+1)*8. EWL valid set value 0 to 15.					R/W

30.5.13 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]							
<hr/>							
classifier	marking	Bit Name Function Read/Write					
for							
honorific							
people							
b7~b0	RECNT	Receive Error Counter	The Receive Error CouNT				R
			RECNT is increased or decreased according to the CAN protocol. There is no				
			overflow for this counter.				

30.5.14 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/Write	
b7~b0 Counter	TECNT	Transmit Error	Transmit Error Counter (Transmit Error CouNT) TECNT is increased or decreased according to the CAN protocol. Attention: When entering Bus_Off, TECNT may overflow, so it may remain a small value.	R

30.5.15 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
classifier for honorific people	marking	celebrity		functionality			fill out or in (information on a form)
b7~b6	Reserved	-		The reset value must be maintained.			R
b5 points to the	SELMASK Filter ID Register	Screen mask register		Selection of the filter's mask register (SELect acceptance MASK) 0 for Select Filter D/A 1: ACF points to the filter MASK register Selection of specific filter register groups via ACFADR			: ACF
b4	Reserved	-		The reset value must be maintained. Acceptance filter address) ACFADR points to specific filters that go through SELMASK to distinguish between ID and MASK. 0000: points to ACF_1 0001: pointing to ACF_2 0010: pointing to ACF_3 0011: pointing to ACF_4 0100: pointing to ACF_5 0101: Pointing to ACF_6 0110: pointing to ACF_7 0111~111: pointing to ACF_8			R
b3~b0 Address	ACFADR	Filter		0111~111: pointing to ACF_8			R/W

30.5.16 CAN Filter Group Enable Register

(CAN_ACFEN) CAN Acceptance Filter Enable

Register Offset Address: 0xB6

Reset value: 0x01

b7	b6	b5	b4	b3	b2	b1	b0
AE_8	AE_7	AE_6	AE_5	AE_4	AE_3	AE_2	AE_1

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
			ACF_8 Enable (Acceptance Filter) 8 Enable)	
b7	AE_8	ACF_8 enable	0: Prohibited 1: Enabling	R/W
			ACF_7 Enable (Acceptance Filter) 7 Enable)	
b6	AE_7	ACF_7 enable	0: Prohibited 1: Enabling	R/W
			ACF_6 Enable (Acceptance Filter) 6 Enable)	
b5	AE_6	ACF_6 enable	0: Prohibited 1: Enabling	R/W
			ACF_5 Enable (Acceptance Filter) 5 Enable)	
b4	AE_5	ACF_5 enable	0: Prohibited 1: Enabling	R/W
			ACF_4 Enable (Acceptance Filter) 4 Enable)	
b3	AE_4	ACF_4 enable	0: Prohibited 1: Enabling	R/W
			ACF_3 Enable (Acceptance Filter) 3 Enable)	
b2	AE_3	ACF_3 enable	0: Prohibited 1: Enabling	R/W
			ACF_2 Enable (Acceptance Filter) 2 Enable)	
b1	AE_2	ACF_2 enable	0: Prohibited 1: Enabling	R/W
			ACF_1 Enable (Acceptance Filter) 1 Enable)	
b0	AE_1	ACF_1 enable	0: Prohibited 1: Enabling	R/W

30.5.17 CAN Filter Group code and mask Register

(CAN_ACF) CAN Acceptance Filter code and mask

Register Offset Address: 0xB8

Reset value: 0xFFFF XXXX

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17												
-	AIDEE	AID E	ACODE[28:16] or AMASK[28:16]																							
b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0										
ACODE[15:0] or AMASK[15:0]																										
classifier marking celebrity functionality fill out for or in honorific (inform people ation on a form)																										
b31	Reserved	-	The readout value is indeterminate.												R											
IDE bit compare enable (Acceptance mask IDE bit check enable)																										
b30	AIDEE	IDE bit	Valid as long as SELMASK=1. 0: Filter receives standard or extended format frames 1: Filter receives standard or extended format												R/W											
frames defined by the AIDE bit IDE bit MASK																										
0: Filter only accepts standard formats 1: Filter only accepts extended formats															R/W											
b29	AIDE	IDE bit MASK	The filter CODE (acceptance filter code) points to the specific filter via ACFADR. SELMASK=0 indicates the CODE of the filter. Bits 10 to bit 0 are used in the standard format and bits 28 to bit 0 are used in the extended format.												R/W											
The filter CODE (acceptance filter mask) points to the specific filter via ACFADR. SELMASK=1 indicates the MASK of the filter. Bits 10 to bit 0 are used in the standard format and bits 28 to bit 0 are used in the extended format.																										
b28~b0	ACODE/ AMASK	Screener CODE/Screen er MASK													R/W											

30.5.18 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	-	-	-			TB PTR[2:0]

Bit Flag	Bit Name	Function	Read/Write
b7	TBE	Set TB to empty	R/W
		Set TB to empty (set TB slot to "empty") 0: No operation 1: SLOT selected by TB PTR is marked as empty	
		TBE is automatically reset to 0 when SLOT is marked empty and TSFF=0.	
		If, when this bit is set to 1, there is a data being sent status in the selected SLOT then TBE=1, then wait until the send TBE is reset to 0 after send completion, send error, or send cancel. TBE is prioritized over TBF.	
b6	TBF	Set TB as filled	R/W
		Set TB filled (set TB slot to "Filled") 0: No operation 1: SLOT selected by TB PTR is marked as populated	
		TBE is automatically reset to 0 when SLOT is marked as filled and TSFF=1.	
b5~b3	Reserved	-	R
		The reset value must be maintained.	
b2~b0	TB PTR	TB SLOT pointer	R/W
		TB SLOT Pointer to a TB message slot) 000: pointing to PTB 001: pointing to STB SLOT1 010: Pointing to STB SLOT2 011: Pointing to STB SLOT3 100: pointing to STB SLOT4	
		Other: Setting Prohibition The pointed TB SLOT is read and write accessible via TBUF and can be tagged via TBE and TBF Whether or not it has been populated. The TBSEL and TSNEXT registers are invalid in TTCAN mode. Note: A write operation can be performed on this bit only when TSFF=0.	

30.5.19 TTCAN Time Trigger Configuration

Register (**CAN_TTCFG**) TTCAN Time Trigger

Configuration Register Offset Address: 0xBF

Reset value: 0x90

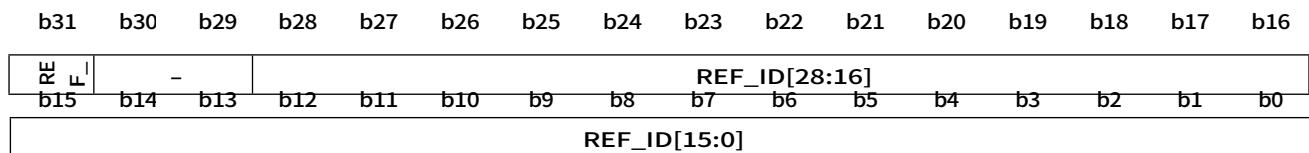
b7	b6	b5	b4	b3	b2	b1	b0
WTIE	WTIF	TEIF	TTIE	TTIF	T_PRESC[1:0]		TTEN
classifier for honorific people	marking	celebrity		functionality			fill out or in (inform ation on a form)
b7	WTIE	Trigger Watchdog Interrupt Enable		Trigger watchdog interrupt enable interrupt Enable (Watch Trigger 0: disabled)			R/W
b6	WTIF	Trigger Watchdog Interrupt Flag WTIF is set		Triggers the Watch Trigger. interrupt Flag			
				n CYCLE COUNT value = TT_WTRIG setting and WTIE = 1. is cleared to 0 by application program write 1.			R/W
b5	TEIF	Trigger Error Interrupt Flag R/W		Trigger Error Interrupt Flag TEIF is set when the TT_TTIG setting is less than the actual CYCLE_TIME.			
b4	TTIE	Time-triggered interrupt enable		Time-triggered interrupt Trigger Interrupt Enable enable (Time 0: disabled) 1: Enabling			R/W
b3	TTIF	Time Trigger Interrupt Flag		Time Trigger Interrupt Trigger Interrupt Flag Flag (Time) TTIF is set when the CYCLE COUNT value = the TT_TRIG setting and TTIE = 1. If TT_TRIG is not updated, TTIF is only set once and the next basic CYCLE is not set. Write 1 to clear 0 through the application program.			R/W
b2~b1	T_PRESC	TTCAN counter prescaler		TTCAN Counter PRESCaler (TTCAN Timer PRESCaler) 00: 1 division of the bit time set in the BT register 01: 2 divisions of the bit time set in the BT register 10: 4 divisions of the bit time set in the BT register 11: 8 divisions of the bit time set by the BT registers Note: T_PRESC can be operated with TTEN=0 for write operations or with TTEN=1 for both.			R/W
b0	TTEN	TTCAN Enable		TTCAN Enable (Time Trigger Enable) 0: Disable 1: Enable TTCAN and the counter starts counting.			R/W

30.5.20 TTCAN Reference Message Register (CAN_REF_MSG)

TTCAN Reference Message Register

offset address: 0xC0

Reset value: 0x0000 0000



classifier for honorific people	marking	celebrity	functionality	fill out or in (inform ation on a form)
b31	REF_IDE	IDE position for reference news	Reference message IDE bit (REFERENCE message IDE bit) 0: Standard format 1: Extended format	R/W
b30~b29	Reserved	-	The readout value is indeterminate.	R
b28~b0	REF_ID	ID bit of the reference message	Reference message ID bit (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 applies to both sending and receiving. When 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 can be supported master. When the highest byte of REF_MSG has been written, then you need to wait for 6 CAN clock cycles to complete the REF_MSG to the CAN clock domain.	R/W

30.5.21 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/Write			
Transmit Enable Window (Transmit Enable Window)						
b15~b12	TEW	Transmit Enable Window	Single Shot Transmit Trigger mode for TTCAN, which can be to set a window of TEW+1 cycle time within which sending is only allowed.	R/W		
b11	Reserved	-	The reset value must be maintained.	R		
Trigger Type						
000: Immediate Trigger for immediate transmission						
001: Time Trigger for receive triggers						
010: Single Shot Transmit Trigger for						
exclusive time windows)						
b10~b8	TTYPE	Trigger Type	011: Transmit Start Trigger for merged arbitrating time windows	R/W		
100: Transmit Stop Trigger for merged arbitrating time windows						
Other: reserved						
The trigger time is set via the TT_TRIG register and the TB Slot is selected via TTPTR.						
b7~b3	Reserved	-Reset value must be maintained.		The reset		
value must be held.		R				
Transmit Trigger TB slot Pointer 000: points to PTB						
001: pointing to STB SLOT1						
010: Pointing to STB SLOT2						
011: Pointing to STB SLOT3						
100: pointing to STB SLOT4						
Other: Setting Prohibition						
If the pointing TB SLOT is marked as empty, TEIF is set when the trigger time is reached.						

30.5.22 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]															
<hr/>															
classifier for marking celebrity functionality fill out or honorific in people (information on a form)															
<hr/>															
Trigger Time Used to specify the cycle time of the trigger, for send trigger send SOF time is about b15~b0 TT_TRIG Trigger time TT_TRIG set value +1 R/W After a write operation of the highest byte of TT_TRIG, the transfer of the TT_TRIG value to the CAN clock domain starts. Therefore as For BYTE operation, you need to write the low byte first and then write the high byte.															

30.5.23 TTCAN Trigger Watch 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]															
<hr/>															
Bit Flag Bit Name Function Read/Write															
<hr/>															
Trigger Time Used to specify the cycle time for a watchdog trigger. After the highest byte of TT_WTRIG is written, the passing of the TT_WTRIG value to the CAN clock domain starts. Therefore, if BYTE operation is performed, the low byte needs to be written before the high byte.															

30.6 Precautions for use

30.6.1 CAN bus anti-interference measures

CAN bus is widely used in automotive, industrial control and other industries, if the CAN application site electromagnetic environment is relatively harsh, there are circuit imbalance, space electromagnetic field, power grid into the line and other factors, it will lead to CAN bus due to radiation, conduction interference and a lot of communication noise, resulting in the increase of bus error frames, reissue frequently, the correct data can not arrive in time and so on, seriously affecting the quality of data communication. Therefore, the practical application should be committed to eliminating noise interference to ensure that the CAN bus network works stably.

The following are a few common types of CAN bus immunity measures (including, but not limited to)

- 1) Increased electrical isolation of the CAN bus interface
- 2) Common transceiver signal ground
- 3) Use shielded twisted pair cable and ground it properly
- 4) Increase in the degree of twisting of CAN transmission lines
- 5) Adding a signal protector
- 6) Improved network topology
- 7) Application layer software anti-jamming mechanisms

30.6.2 CAN controller noise constraints

In a CAN bus network, it should be ensured that the bit time of the communication meets the requirements of the standard protocol. The introduction of noise disturbances that do not meet the width of the bit time may cause the CAN controller to act abnormally.

31USB2.0 Full Speed Module (USBFS)

31.1 Introduction to USBFS

The USB Full Speed (USBFS) controller provides a USB communication solution for portable devices. The USBFS controller supports both host and device modes with an integrated full-speed PHY on-chip. The USBFS controller supports both full-speed (FS, 12Mb/s) and low-speed (LS, 1.5Mb/s) transceivers in host mode, and only full-speed (FS, 12Mb/s) transceivers in device mode. The USBFS controller supports all transfer modes defined by the USB 2.0 protocol (control transfer, bulk transfer, interrupt transfer, and synchronous transfer). In host mode, the USBFS controller supports both full-speed (FS, 12Mb/s) and low-speed (LS, 1.5Mb/s) transceivers, while in device mode, it supports only full-speed (FS, 12Mb/s) transceivers.

31.2 USBFS Key Features

There are three main categories: generic features, host mode features, and device mode features.

31.2.1 General characteristics

- Built-in on-chip USB2.0 full-speed PHY
- Supports host mode and device mode
- Supports FS SOF and low-speed "Keep-alive" tokens with the following features:
 - SOF Pulse Pin Output Function
 - SOF pulse can be used as an internal event source to trigger TIMER, DMA and other modules.
 - Configurable frame period
 - Configurable end-of-frame interrupt
- Module with embedded DMA and software configurable AHB burst transfer type
- Power saving features such as USB hang, stop RAM clock, stop PHY domain clock
- 1.25KB dedicated RAM with advanced FIFO control
 - RAM space can be divided into different FIFOs for flexible and efficient use of RAM
 - Multiple packets per FIFO
 - Dynamic allocation of storage areas
 - The size of the FIFO can be configured to be a non-power-of-2 value to allow for continuous use of the memory cell.
- Maximum USB bandwidth can be achieved within one frame without application intervention
- Automatically determines host mode or device mode based on the level of the ID line

31.2.2 Host Mode Features

- Host mode supports USB2.0 Full Speed (FS, 12Mb/s) and Low Speed (LS, 1.5Mb/s) transfers
- Requires external power supply chip to generate VBUS voltage
- Up to 12 host channels (pipes) each channel can be dynamically reconfigured to support any type of USB transfer
- Built-in hardware scheduler that:
 - Stores up to 8 interrupt plus synchronous transfer requests in a periodic hardware queue
 - Stores up to 8 control plus batch transfer requests in a non-periodic hardware queue
- Manages a shared RX FIFO, a cyclic TX FIFO, and an acyclic TX FIFO for efficient use of USB data RAMs

31.2.3 Device Mode Characteristics

- Slave mode supports USB2.0 Full Speed (FS, 12Mb/s) transfers
- 1 bi-directional control endpoint 0
- 5 OUT endpoints that can be configured to support batch, interrupt or synchronized transfers
- 5 IN endpoints that can be configured to support batch, interrupt or synchronized transfers
- Contains 6 transmit FIFOs (one transmit FIFO per IN endpoint) and one receive FIFO (shared by all OUT endpoints)
- Supports remote wake-up function
- Supports Soft Disconnect
- VBUS PIN support 5V withstand voltage

31.3 USBFS System Block Diagram

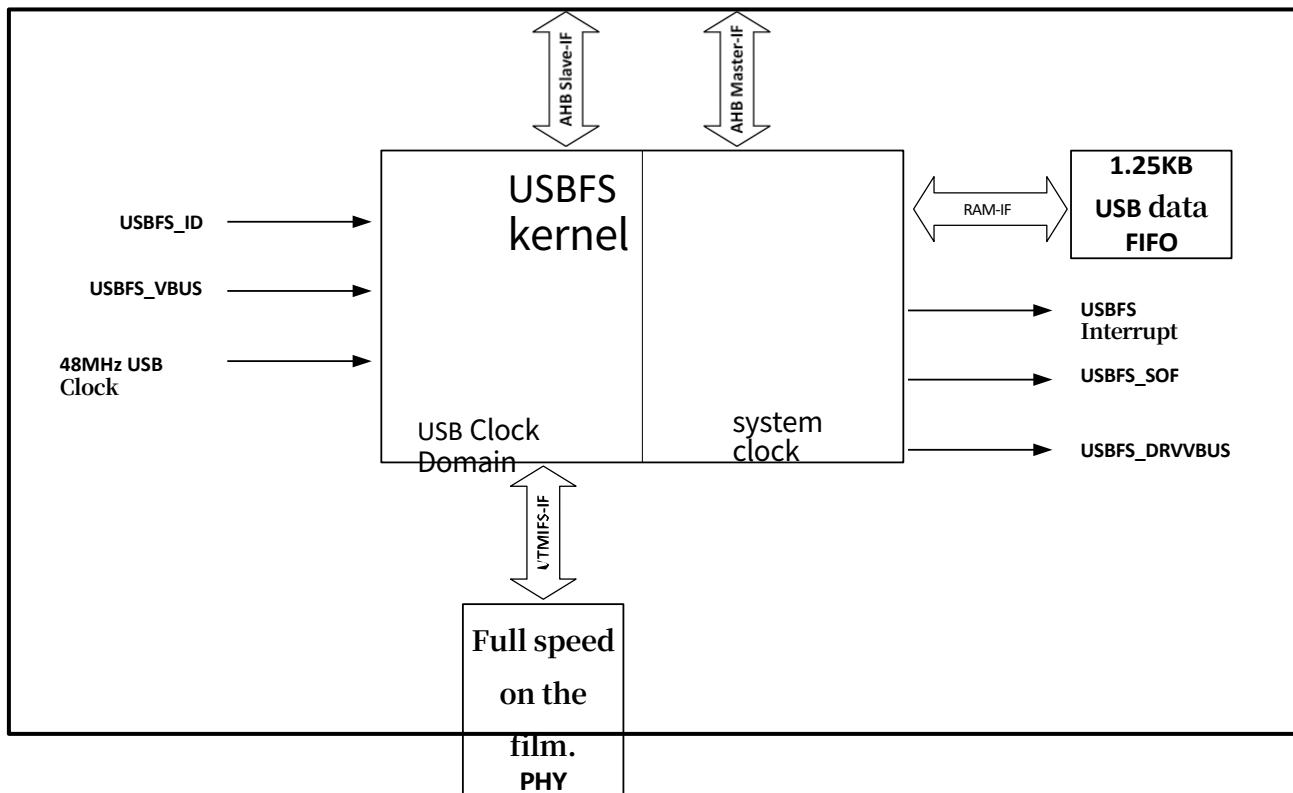


Figure 31-1 USBFS System Block Diagram

31.4 USBFS Pinout

Table 31-1 USBFS Pin Descriptions

pin name	orientations	Functional Description
USBFS_VBUS	importation	Power Port, 5V Withstand Voltage
USBFS_DP	Input/Output	Differential data D+ signal
USBFS_DM	Input/Output	Differential Data D-Signal
USBFS_DRVVBUS	exports	External power chip enable signal
USBFS_ID	importation	USB A-B Device Identification Signal
USBFS_SOF	exports	SOF output pulse signal

Since the USBFS_DP and USBFS_DM pins are multiplexed with the general-purpose GPIOs, it is recommended to disable the digital function of their corresponding pins when using USBFS, please refer to the [General-Purpose GPIO] section for details. In addition, when the USBFS function is not used, the digital function pins corresponding to the USBFS_DP and USBFS_DM pins are flipped, resulting in additional current consumption.

31.5 USBFS Feature Description

31.5.1 USBFS Clock and Operating Modes

The clock used for USBFS needs to be configured at 48MHz, which is generated by the internal PLL circuit, and the PLL clock source needs to be 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 device and includes an on-chip full-speed PHY.

The on-chip full-speed PHY has integrated pull-up and pull-down resistors internally, which can be automatically selected by USBFS based on the current mode and connection status. USBFS operates with a VCC voltage range of 3.0~3.6V.

31.5.2 USBFS mode decision

USBFS determines the current operating mode in the following two ways:

Method 1: Automatically recognize the USBFS_ID line according to its status. When the USBFS_ID line is detected as high, the module works in device mode, and when the USBFS_ID line is detected as low, the module works in host state.

Method 2: Force host/device mode, by setting the FDMOD or FHMOD bit of register USBFS_GUSBCFG to 1 to force the module to work in device or host mode by ignoring the level of USBFS_ID line.

31.5.3 USBFS Host Functions

31.5.3.1 Host Function Introduction

USBFS_DRVVBUS is used to enable the external USB power supply chip when USBFS is working in host mode, VBUS is the 5V power supply pin specified by the USB protocol. The internal PHY does not support 5V power supply, so an external USB power supply chip is needed to supply power to the device. USBFS_DRVVBUS is used to enable the external USB power supply chip, and the over-current detection of the external power supply chip can be realized by the external interrupt IRQ of this MCU. USB_VBUS can be used as GPIO in host mode.

A typical USB host mode system build diagram is shown below:

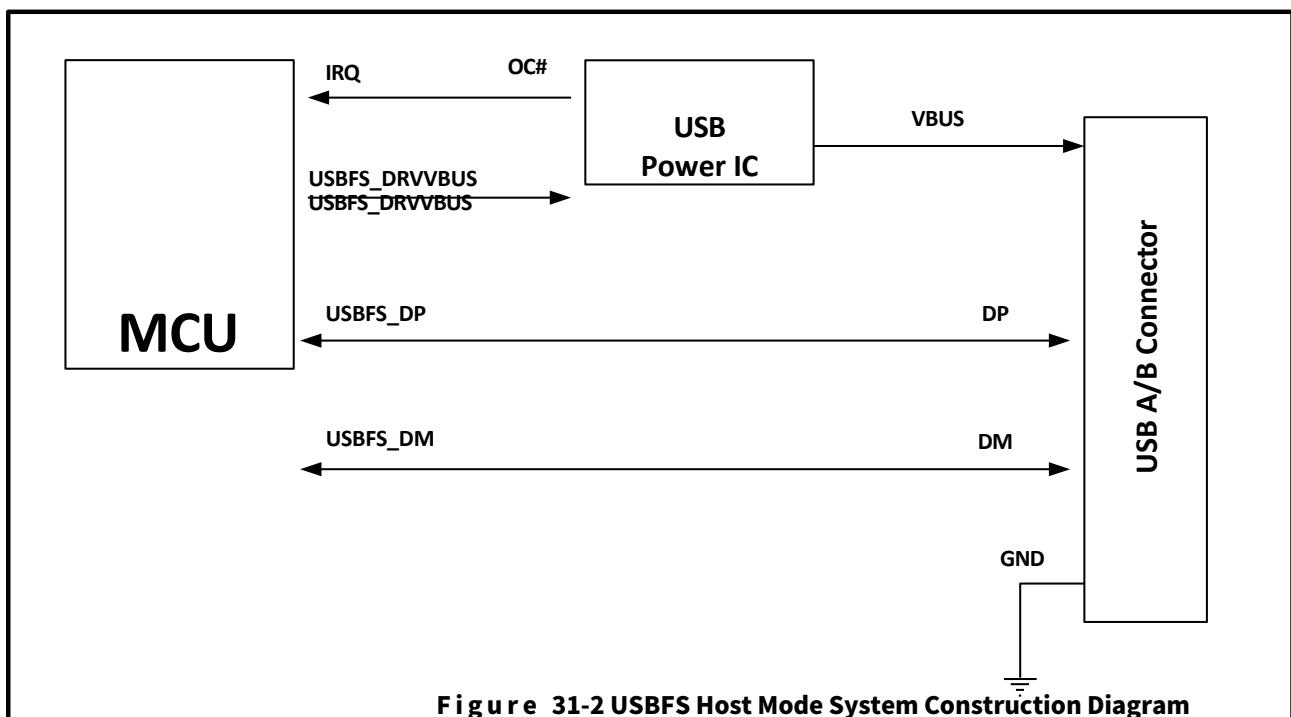


Figure 31-2 USBFS Host Mode System Construction Diagram

31.5.3.2 Host Port

Power

The MCU cannot output 5V to provide VBUS, for which a USB power chip or basic power switch must be added in addition to the microcontroller.

(The external power supply chip can be driven from any GPIO output or from the USBFS_DRVVBUS line if the application board provides a 5V power supply. The external power chip can be driven by any GPIO output or by USBFS_DRVVBUS. When the application determines to use GPIOs to control the external device to provide VBUS, the port power bit in the host port control and status register (PPWR bit in USBFS_HPRT) must still be set to one.

31.5.3.3 Host detects device connections and disconnections

The USB device will be detected as soon as it is connected. the USBFS module will signal a host

Status Register (PCDET bit in USBFS_HPRT).

A device disconnect event will trigger a disconnect detection interrupt (DISCINT bit in USBFS_GINTSTS)

31.5.3.4 host enumeration

After a device connection is detected, if a new device is connected in, the host must initiate the enumeration process by sending a USB reset and configuration command to the new device.

The application drives a USB reset signal (single-ended zero) through the USB by setting the port reset bit in the host port control and status register (PRST bit in USBFS_HPRT) to 1, causing the process to last a minimum of 10 ms and a maximum of 20 ms. The application program calculates the duration of this process and then clears the port reset bit to zero.

Immediately after the USB reset sequence is completed, the port enable/disable change bit (the PENCHNG bit in USBFS_HPRT) triggers an interrupt on the host port, which in turn sends a notification to the application program indicating that the port speed field from the host port's control and status registers is available.

(PSPD in USBFS_HPRT) reads the speed of the enumerated device and the fact that the host has begun driving the SOF (FS) or Keep-alive token (LS). The host is now ready to complete enumeration of the device by sending commands to the device.

31.5.3.5 Host hangs

The application program hangs the USB bus by setting the port hang bit in the host port control and status register (PSUSP in USBFS_HPRT) to 1. The USBFS module stops transmitting SOF and enters the hang state.

The bus can be taken out of the pending state by autonomous activity of the remote device (Remote Wakeup). In this case, the Wake-on-Remote signal triggers the Wake-on-Remote interrupt (WKUPINT bit in USBFS_GINTSTS) and the hardware puts the Port Restore bit in the Host Port Control and Status Registers

(PRES bit in USBFS_HPRT) resets itself and automatically drives the recovery signal via USB. The application must time the recovery window and then clear the port recovery bit to zero to exit the hung state and restart the SOF.

If initiated by the host to exit the hung state, the application must restore the port to position 1 to initiate the restore signal on the host port, timing the restore window and eventually clearing the port restore bit to zero.

31.5.3.6 host channel

The USBFS module implements 12 host channels. Each host channel can be used for USB host transfers (USB pipes). The host can process up to 8 transfer requests simultaneously. If an application has more than 8 transfer requests pending, the Host Controller Driver (HCD) must reassign the channel for unprocessed transfer requests after the channel is released from its

previous assignment (i.e., after receiving a transfer completion and channel stop interrupt)

Each host channel can be configured to support input/output and periodic/non-periodic transactions. Each host channel uses a dedicated control (HCCHARx) register, a transfer configuration (HCTSIZx) register/interrupt (HCINTx) register, and its associated interrupt mask register (HCINTMSKx).

Host Channel Control

The application program can **control** the host channel through the Host Channel x Characterization Register (HCCHARx) as follows:

- 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 the channel
- Set the transmission direction on this channel: IN/OUT
- Set the type of USB transfer on this channel: Control/Batch/Interrupt/Sync
- Set the maximum packet length for device endpoints communicating with this channel
- Setting the frames to be periodically transmitted: odd/even frames

Host Channel Transmission

The Host Channel Transmit Size Register (**HCTSIZx**) allows the application program to program the transmit size parameter and read the transmission status. This register must be set prior to channel enable position 1 in the Host Channel Characteristics Register. Immediately after the endpoint is enabled, the Packet Count field becomes read-only and is updated by the USBFS module based on the current transmission status.

The following transmission parameters can be programmed:

- Transfer size in bytes
- Number of packets that make up the entire transmission size
- Initial data PID

Host channel status/interrupts

The Host Channel x Interrupt Register (**HCINTx**) indicates the status of the endpoint in the event of USB and AHB related events. When the host channel interrupt bit in the interrupt registers (HCINT bit in **USBFS_GINTSTS**) is set to 1, the application must read these registers for details. Before reading these registers, the application program must read the Host All Channel Interrupt (**HCAINT**) register to obtain the channel number of the host channel x interrupt register. The application must clear the corresponding bit in this register before it can clear the corresponding bit in the **HAIINT** and **GINTSTS** registers. The **USBFS_HCINTMSK x** register also provides mask bits for each interrupt source for each channel.

The host module provides the following status checking and interrupt generation functions:

- Transfer complete interrupt, indicating that both the application program (AHB) and the USB side have completed the data transfer
- Channel stops due to transfer completion, USB transaction error, or inhibit command from application program
- 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 checksum failures, timeouts, bit-fill errors, and incorrect EOPs
- crosstalk error
- frame overflow
- Flip-flop bit error for data synchronization

31.5.3.7 host scheduler

The host module has a built-in hardware scheduler that autonomously reorders and manages USB transaction requests from applications. At the beginning of each frame, the host performs a cyclic (synchronization and interrupt) transaction followed by an acyclic (control and batch) transaction to comply with the USB specification's guarantee of high priority for synchronization and interrupt transfers.

The host handles USB transactions through request queues, a periodic request queue and an acyclic request queue. Each request queue can store up to eight entries. Each entry represents an unanswered USB transaction request initiated by an application program and stores the number of the IN or OUT channel used to execute the USB transaction, as well as other pertinent information. The order in which USB transaction requests are written to the queue determines the order in which the transactions are executed on the USB interface.

At the beginning of each frame, the host processes the periodic request queue first and then the nonperiodic request queue. The host will issue an incomplete periodic transfer interrupt (IPXFR bit in USBFS_GINTSTS) if a synchronization or interrupt type of USB transfer transaction request scheduled to be executed at the end of the current frame is still pending at the end of the current frame. The USBFS module is responsible for the management of both the periodic and nonperiodic request queues. The Periodic Transmit FIFO and Queue Status Registers (HPTXSTS) and the Non-Periodic Transmit FIFO and Queue Status Registers (HNPTXSTS) are read-only registers that can be used by an application program to read the status of the respective request queues, including:

- Number of currently available free entries in the periodic (non-periodic) request queue (up to 8)
- Currently available free space in periodic (non-periodic) TxFIFO (OUT transaction)
- IN/OUT tokens, host channel numbers and other status information

Since each request queue can store up to eight USB transaction requests, applications can send host USB transaction requests to the scheduler in advance; the actual communication will appear on the USB bus at the latest after the scheduler has finished processing the eight pending periodic transactions and the eight non-periodic transactions.

To issue a transaction request to the host scheduler (queue) the application must read the PTXQSAV bit in the USBFS_HNPTXSTS register or the NPTQXSAV bit in the USBFS_HNPTXSTS register to ensure that there is at least one available space in the periodic (non-periodic) request queue to store the current request.

31.5.4 USBFS Device Features

31.5.4.1 Equipment Function Introduction

When USBFS is operating in device mode, VBUS is a 5V power pin as defined by the USB protocol and is a 5V voltage tolerant pin. The module always detects the level of the VBUS line to connect or disconnect the device.

A typical USB device mode system build diagram is shown below:

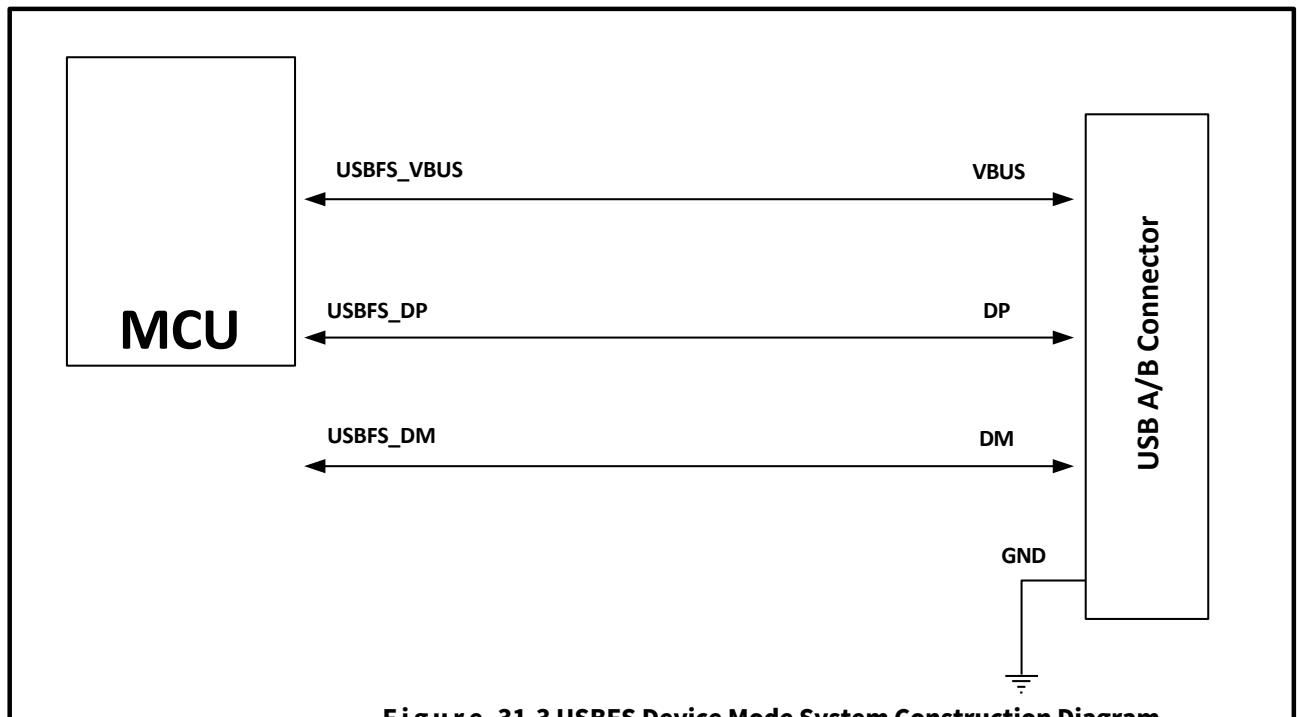


Figure 31.3 USBFS Device Mode System Construction Diagram

31.5.4.2 Device power status

The module puts the USB device into the powered state when it detects that USBFS_VBUS is high. USBFS then automatically connects the DP pull-up resistor, signals the full speed device to the host and generates a session request interrupt (VBUSVINT bit in USBFS_GINTSTS) indicating entry into the powered state.

In addition, the USBFS_VBUS input ensures that the host provides a valid VBUS level during USB operation. If a low VBUS level is detected (e.g., triggered by a power disturbance or host port shutdown) USBFS will automatically disconnect.

In the powered state, USBFS expects a reset signal from the host. No other USB operations can be performed. A detected reset interrupt (USBRST in USBFS_GINTSTS) is generated immediately after the reset signal is received. After the reset signal has expired, an enumeration completion interrupt is generated.

(ENUMDNE bit in USBFS_GINTSTS) USBFS then enters the default state.

31.5.4.3 Device Default State

By default, USBFS expects to receive the SET_ADDRESS command from the host. No other USB operations can be performed. When a valid SET_ADDRESS command is decoded on the USB, the application writes the corresponding address value to the Device Address field in the Device Configuration Register (the DAD bit in USBFS_DCFG). USBFS then enters the address state and is ready to respond to host transactions with the configured USB address.

31.5.4.4 Device pending status

The USBFS device continuously monitors USB activity. An early hang interrupt is issued after the USBidle time reaches 3ms (ESUSP bit in USBFS_GINTSTS) and the device enters the suspend state after 3ms as confirmed by the suspend interrupt (USBSUSP bit in USBFS_GINTSTS). The device suspend bit in the Device Status Register (SUSPSTS bit in USBFS_DSTS) is then automatically set to 1 and USBFS enters the suspend state.

The pending state can be exited by the device itself. In this case, the application program will set the remote wake-up signal bit in the device control registers (RWUSIG bit in USBFS_DCTL) is set to 1 and cleared within 1ms to 15m.

However, if the device detects a recovery signal from the host, a recovery interrupt is generated (WKUPINT bit in USBFS_GINTSTS) and the device hang bit is automatically cleared.

31.5.4.5 soft disconnect

The powered state can be exited via software with the help of the soft-disconnect function. Setting the Soft Disconnect bit in the Device Control Register (SDIS bit in USBFS_DCTL) to 1 removes the DP pull-up resistor, and a Device Disconnect Detect interrupt occurs on the host side, even though the USB cable is not physically unplugged from the host port.

31.5.4.6 Device

Endpoint

Endpoint Class

The USBFS module implements the following USB endpoints:

- Control Endpoint 0:

- Bidirectional and only control messages are processed
- Use a separate set of registers for IN and OUT transactions.
- Dedicated control

(USBFS_DIEPCTL0/USBFS_DOEPCTL0) Register,

Transmission Configuration (USBFS_DIEPTSIZ0/USBFS_DOEPTSIZ0) Register Register
registers, transmission and Status state

Interrupt Disconnect

(USBFS_DIEPINT0/USBFS_DOEPINT0) registers. The set of bits available in the control and transfer size registers is slightly different than in the other endpoints

■ **5 IN endpoints**

- Each endpoint can be configured to support synchronized, bulk or interrupted transmission types

- Each endpoint has a dedicated control (USBFS_DIEPCTLx) register, a transfer configuration (USBFS_DIEPTSIZx) register, and a status interrupt (USBFS_DIEPINTx) register
 - Device IN Endpoint Generalized Interrupt Mask Register (USBFS_DIEPMSK) can be used to enable/disable all IN Endpoints (including EP0) on the same class of endpoint interrupt source
 - Supports an incomplete synchronous IN transmission interrupt (IISOIXFR bit in USBFS_GINTSTS) that is triggered when a transmission on at least one synchronous IN endpoint in the current frame is incomplete. This interrupt is triggered together with the periodic frame interrupt (USBFS_GINTSTS/EOPF)
- 5 OUT endpoints
- Each endpoint can be configured to support synchronized, bulk or interrupted transmission types
 - Each endpoint has dedicated control (USBFS_DOEPCTLx) registers, transmission configuration (USBFS_DOEPTSIZx) registers, and status interrupt (USBFS_DOEPINTx) registers
 - Device OUT Endpoint Common Interrupt Mask Register (USBFS_DOEPMSK) can be used to enable/disable the same type of endpoint interrupt source on all OUT endpoints (including EP0)
 - Supports an incomplete Sync OUT transmission interrupt (INCOMPISOOUT bit in USBFS_GINTSTS) which is triggered when a transmission is incomplete on at least one Sync OUT endpoint in the current frame. This interrupt is triggered in conjunction with the periodic frame interrupt (USBFS_GINTSTS/EOPF).

endpoint control

The application program can take the following controls over the endpoint through the Device Endpoint x IN/OUT Control Register (DIEPCTLx/DOEPCTLx):

- Endpoint Enable/Disable
- Activate the endpoint in the current configuration
- Setting the USB transfer type (synchronous, batch and interrupt)
- Setting the supported packet size
- Setting the Tx-FIFO number associated with an IN endpoint
- Set the data0/data1 PID you want to receive or use when sending (bulk/interrupt transfers only)
- Sets the odd/even frames corresponding to when receiving or sending transactions (synchronous transmission only)
- The NAK bit can be set to reply to host requests with NAK regardless of the state of the FIFO

at the time.

- The STALL bit can be set so that all host tokens to this endpoint are answered by hardware STALL
- The OUT endpoint can be set to listen mode, i.e., no CRC checking is performed on the received data.

endpoint transmission

The Device Endpoint x Transmission Length Register (DIEPTSIZx/DOEPTSIZx) allows the application program to program the transmission length parameters and read the transmission status. Setup of this register must be completed prior to endpoint enable position 1 in the Endpoint Control Register. Immediately after the endpoint is enabled, these fields become read-only and are updated by the USBFS module based on the current transmission status.

The following transmission parameters need to be configured:

- Length of a single transmission in bytes
- The number of packets that make up the entire transmission

Endpoint Status/Status

The Device Endpoint x Interrupt Register (DIEPINTx/DOEPINTx) indicates the status of the endpoint in the event of a USB and AHB related event. When the OUT endpoint interrupt bit or the IN endpoint interrupt bit (the OEPINT bit in USBFS_GINTSTS or the IEPINT bit in USBFS_GINTSTS, respectively)¹, the Module Interrupt Register is set to 1, the application program must read these registers for details. Before the application can read these registers, it must read the Device All Endpoint Interrupt (USBFS_DAINT) register to obtain the endpoint number of the Device Endpoint x Interrupt register. The application program must clear the corresponding bit in this register before it can clear the corresponding bit in the DAINT and GINTSTS registers.

- The module provides the following status check and interrupt functions:
- Transfer complete interrupt, indicating that the application has completed the data transfer on both the AHB and USB side
- Setup phase completed (only for OUT endpoints of control transmission type)
- The associated transmit FIFO is half-empty or fully empty (IN endpoint)
- NAK response sent to host (only for IN endpoints for synchronous transfers)
- IN token received when TxFIFO is empty (only for IN endpoints for bulk and interrupt transfer types)
- OUT token received when the endpoint is not yet enabled
- Babble error detected
- Application shutdown endpoints take effect
- Application sets NAK in effect for endpoints (only for IN endpoints of synchronous transport type)
- More than 3 consecutive setup packets received (for control type OUT endpoints only)
- Timeout condition detected (only for IN endpoints of control transmission type)
- Synchronous transmission type packets are lost without generating interruptions

31.5.5 USBFS SOF Pulse Pin Output Function

The USBFS monitors, traces, and configures SOF frames in both host and device modes and also features SOF pulse output, which is output via the USBFS_SOF pin and is 16 system clock cycles wide.

31.5.5.1 Host SOF

In host mode, the number of PHY clocks that occur during two consecutive SOF (FS) or

keep-alive (LS) tokens generated can be programmed in the Host Frame Interval Register (HFIR), allowing the application program to control the SOF frame period. An interrupt is generated at the start of the frame (SOF bit in USBFS_GINTSTS). The current frame number and the time remaining before the next SOF occurs can be tracked by the application in the Host Frame Number Register (HFNUM).

Using the SOFEN bit in the USBFS system control register, USBFS_SYCTLREG, a simultaneously generated SOF pulse signal of 16 system clock cycles in width issued by any SOF token can be caused to be output from the USBFS_SOF pin.

In addition, SOF pulses can be used as internal events to trigger external modules such as DMA transfers, TIMER counting, and so on.

31.5.5.2 Equipment SOF

In device mode, the USB triggers a frame start interrupt (SOF bit in USBFS_GINTSTS) each time it receives a SOF token. The corresponding frame number can be read from the Device Status Register (FNSOF bit in USBFS_DSTS). Using the SOFEN bit in the USBFS System Control Register, USBFS_SYCTLREG, it is also possible to generate a SOF pulse signal with a width of 16 system clock cycles and have this signal output on the USBFS_SOF pin for external availability.

In addition, SOF pulses can be used as internal events to trigger external modules such as DMA transfers, TIMER counting, and so on.

The periodic end-of-frame interrupt (GINTSTS/EOPF) is used to notify the application program when 80%, 85%, 90%, or 95% of the frame interval time has elapsed, depending on the Periodic Frame Interval field in the Device Configuration Register (PFIVL bit in USBFS_DCFG). This function can be used to determine if all synchronization communications for that frame are complete.

31.5.6 USBFS Power Consumption Control

When the USBFS module is not being used, power reduction can be achieved by stopping the HCLK and PHY clocks of the USBFS module via the CMU module. When the USB module is used, but the device USB session has not started or the device is not connected, power reduction techniques can be used in the USB hang state.

- Stop PHY clock (STPPCLK bit in USBFS_GCCTL)

Most of the 48 MHz internal clock domain of the USBFS Full Speed Module is turned off by clock gating when the Stop PHY Clock position 1 in the Clock Gating Control Register is set. Even if the application still provides clock input, this saves the module from dynamic power consumption due to clock signal flip-flop and turns off most of the transceiver, leaving only the part responsible for detecting asynchronous recovery events or remote wake-up events active.

- HCLK gating (GATEHCLK bit in USBFS_GCCTL)

With GATEHCLK in the Clock Gating Control Register in position 1, most of the system clock domains inside the USBFS module are turned off by clock gating. Only the register read and write interfaces remain active. Even if the application still provides clock inputs,

the dynamic power consumption of the module due to clock signal flip-flop is saved.

To save dynamic power consumption, the USB data FIFO is clocked only when it is accessed by the USBFS module.

31.5.7 USBFS Dynamic update of the USBFS_HFIR register

In host mode, the USB module has the ability to dynamically fine-tune the frame period to synchronize external devices with the SOF frame. If the USBFS_HFIR register is changed during the current SOF frame, the SOF period will be corrected accordingly in the next frame, see Figure 31-4 for details.

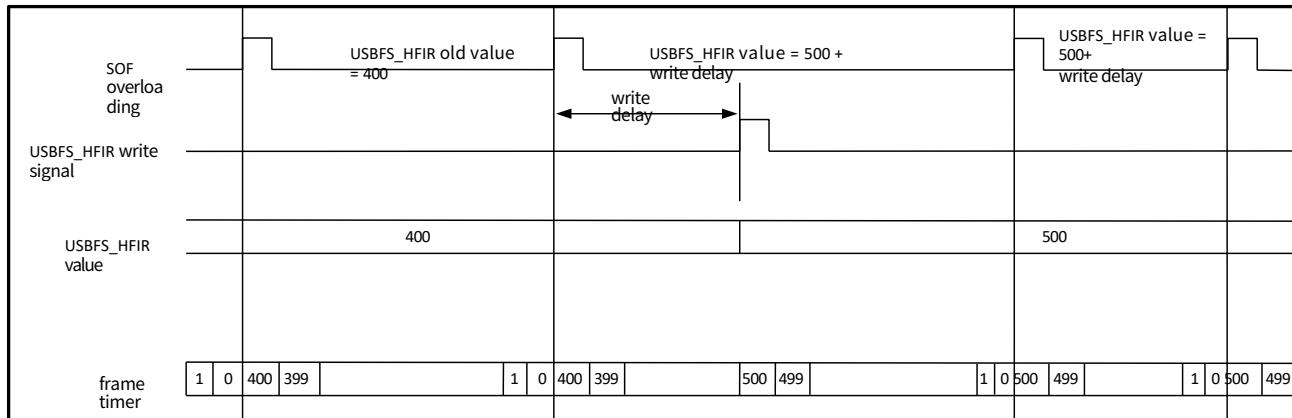


Figure 31-4 USBFS Dynamic Update USBFS_HFIR Register Schematic

31.5.8 USBFS Data FIFO

The USBFS system has 1.25KB of dedicated RAM and utilizes an efficient FIFO control mechanism. The packet FIFO controller module in the USBFS module divides the RAM space into multiple TxFIFOs (into which the application presses data for transient storage prior to the USB transfer) and a single Rx FIFO (into which data received from the USB is stored for transient storage before it is read by the application). (where data received from USB is stored briefly before being read by the application)

The number and organization of the FIFOs built in RAM depends on the role of the device. In device mode, one Tx FIFO is configured for each active IN endpoint. the size of the FIFOs is configured by software to better meet the application requirements.

31.5.9 USBFS Host FIFO Architecture

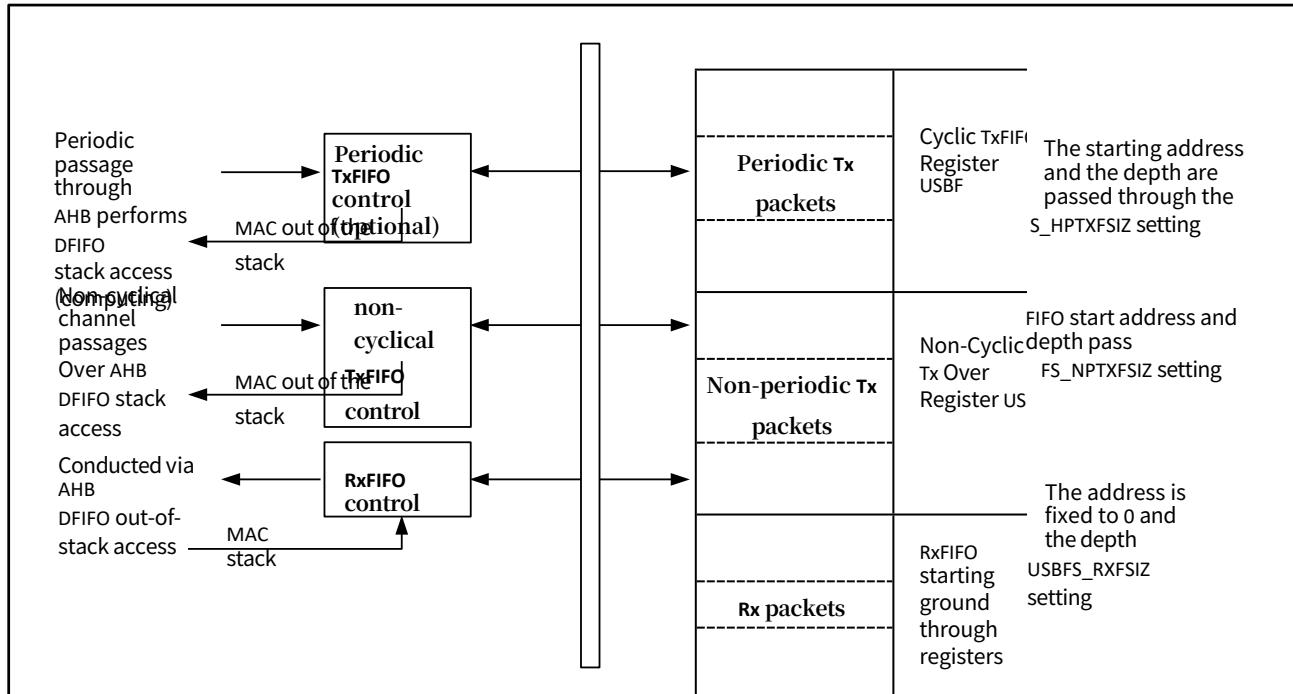


Figure 31-5 FIFO Architecture Schematic in USBFS Host Mode

31.5.9.1 Host

RxFIFO

The host uses a receive FIFO to handle all cyclic and acyclic transactions. The FIFO is used as a receive buffer to hold data received from the USB (the data portion of a received packet) until it is transferred to system memory. As long as there is space in the FIFO, packets from the IN endpoint of the device are received and stored one by one. The status of each received packet (including host destination channel, byte count, data PID, and checksum of the received data) is also stored in the FIFO. The size of the receive FIFO is configured in the Receive FIFO Size Register (GRXFSIZ).

The single receive FIFO architecture allows the USB host to efficiently populate the receive data buffer:

- All IN configured 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 the host software.

The application program receives Rx FIFO out-of-air breaks 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 out registers and finally reads the data from the RxFIFO.

31.5.9.2 Host TxFIFO

and another transmit FIFO to handle all cyclic (synchronization and interrupt) OUT transactions. The FIFO is used as a transmit buffer to hold the data to be sent over USB (transmit packet). The size of the cyclic (non-cyclic) TxFIFO is configured in the Host Cyclic (Non-Cyclic) Send FIFO Size (HPTXFSIZ/HNPTXFSIZ) register.

The two Tx FIFOs implement operations on a priority basis, with cyclic communications having a higher priority, so that cyclic communications are performed first during the time of a USB frame. At the start of the frame, the built-in host scheduler processes the periodic request queue first and then the non-periodic request queue.

The two transmit FIFO architecture allows the USB host to optimize the management of cyclic and acyclic transmit data buffers separately:

- All host channels configured to support cyclic (acyclic) OUT transactions share the same RAM buffer (shared FIFO)
- For any sequence of OUT tokens driven by the host software, the USBFS module can fill the periodic (non-periodic) transmit FIFO to the limits

As long as the periodic TxFIFO is half empty or full empty, the USBFS module will issue a periodic TxFIFO empty interrupt.

(PTXFE bit in USBFS_GINTSTS) depending on the value of the Periodic Tx-FIFO Empty Level bit (PTXFELVL bit in USBFS_GAHBCFG) in the AHB Configuration Register. Applications can write ahead to send data as long as there is free space in both the periodic TxFIFO and the periodic request queue. You can find out how much space is available in the Host Periodic Transmit FIFO and the Queue Status Register (HPTXSTS).

The USBFS module issues a non-periodic TxFIFO null whenever the non-periodic TxFIFO is half-empty or full-empty.

(NPTXFE bit in USBFS_GINTSTS) depending on the Non-Cyclic TxFIFO Empty Level bit (TXFELVL bit in USBFS_GAHBCFG) in the AHB Configuration Register. As long as there is free space in both the noncyclic TxFIFO and the noncyclic request queue, the application program can write to send data. You can find out how much space is available in both the Host Non-Cyclic Transmit FIFO and the Queue Status Register (HNPTXSTS) by reading them.

31.5.10 USBFS Device FIFO Architecture

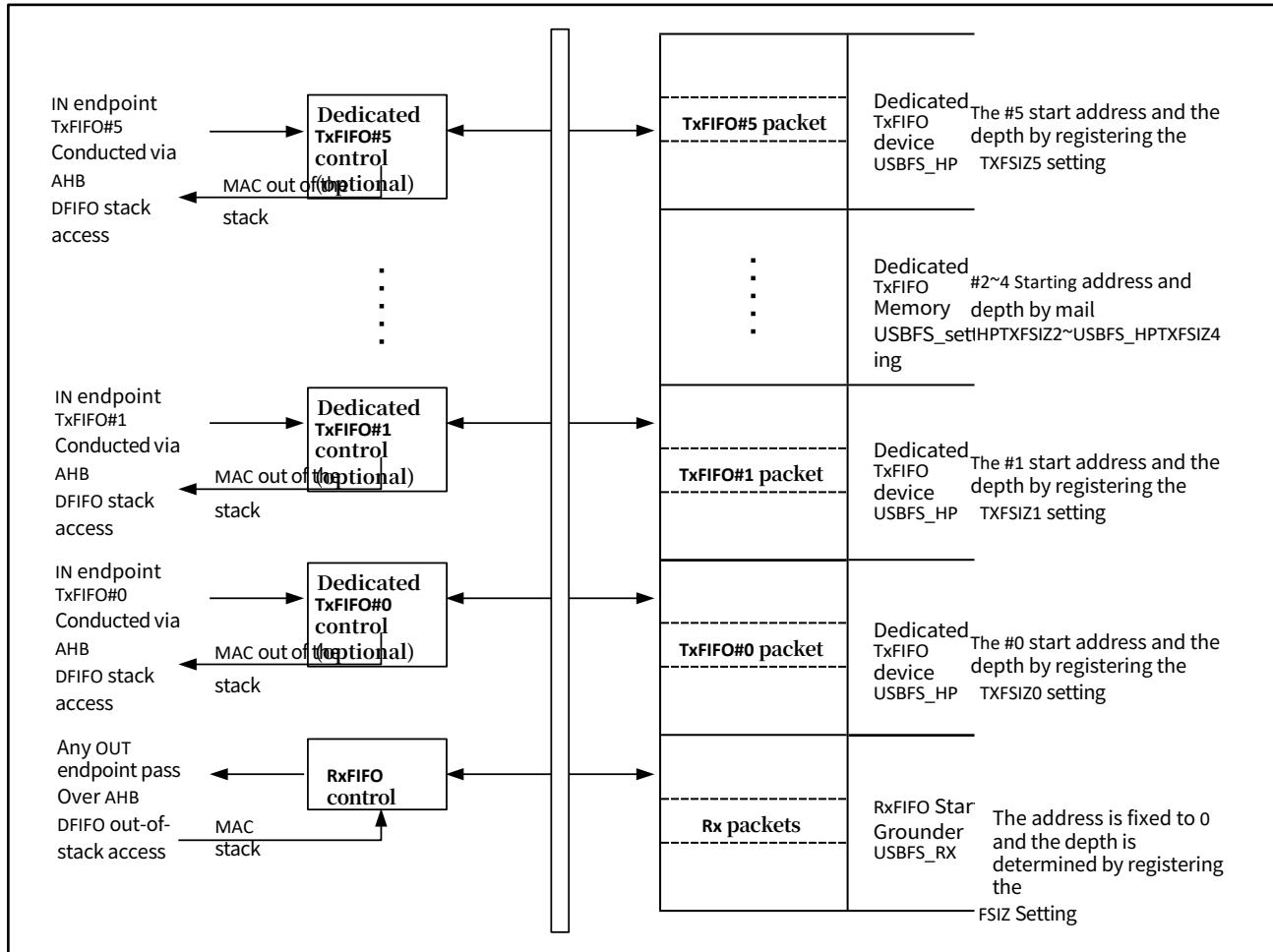


Figure 31-6 FIFO Architecture Schematic in USBFS Device Mode

31.5.10.1 Device RxFIFO

The USBFS device uses a single receive FIFO to receive data to all OUT endpoints. As long as there is free space in the Rx FIFO, received packets are filled into the Rx FIFO one by one, and in addition to the valid data, the status of the received packet (including the OUT endpoint destination number, number of bytes, data PID, and validation of the received data) is also stored by the module. When no space is available, the device responds with a host transaction NAK answer and triggers an interrupt on the addressed endpoint. The size of the receive FIFO is configured in the receive FIFO size register (GRXFSIZ).

The single receive FIFO architecture allows USB devices to fill the receive RAM buffer more efficiently:

- All OUT endpoints share the same RAM buffer (shared FIFO)
 - The USBFS module fills the receive FIFO to the limit for any host sequence OUT token.
- As long as at least one packet is available for reading in the Rx FIFO, the application will always receive Rx FIFO non-air breaks (RXFNE bit in USBFS_GINTSTS) The application program reads the packet information

from the Receive Status Read and Stackout Register (GRXSTSP) and finally reads the corresponding data from the receive FIFO by reading the stackout address associated with the endpoint.

31.5.10.2 Device TxFIFO

The module provides dedicated FIFOs for each IN endpoint. The application program configures the FIFO size for IN endpoint 0 via the Off-Cycle Transmit FIFO Size Register (USBFS_DIEPTSIZE0) and for IN endpoint x via the Device IN Endpoint Transmit FIFOx Register (DIEPTSIZEx).

31.5.11 USBFS FIFO RAM Allocation

31.5.11.1 Host

mode receives

FIFO RAM

allocation

Status information is written to the FIFO with each receive packet; therefore, space of at least (Maximum packet size / 4) + 1 must be allocated for receive packets. If multiple synchronization channels are enabled, the space allocated for receiving successive packets must be at least (maximum packet size / 4) + 1.

Normally, the recommended space is twice (max packets/4 + 1) so that when the last packet is transmitted to the CPU, the USB can receive subsequent packets at the same time.

The transmission completion status information is written to the FIFO along with the last packet received by that endpoint. so a location must be allocated for this.

When running in DMA mode, the DMA address registers for each host channel will be stored in the FIFO, so it is necessary to reserve a location in the FIFO for each channel to store its address registers.

Send FIFO RAM Allocation

The minimum RAM required for the host to send a FIFO non-periodically is the size of the largest packet transmitted on all supported non-periodic OUT channels.

Typically, the recommended space is twice the maximum packet size, so that while the USB is sending the current packet, the AHB can fill the sending FIFO with the next packet.

The minimum RAM required for the host's periodic transmit FIFO is the size of the largest packet transmitted on all supported periodic OUT channels. If there is at least one synchronous 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 for each host channel will be stored in the FIFO, so it is necessary to reserve a location in the FIFO for each channel to store its address registers.

31.5.11.2 Device

mode receives**FIFO RAM****allocation**

The application program shall allocate RAM for SETUP packets¹¹ locations in the receive FIFO must be reserved to receive SETUP packets on the console. the USBFS module will not write any other data to these locations reserved for SETUP packets. The USBFS module will not write any other data to these locations reserved for SETUP packets. A location will be assigned to the global OUT NAK. Status information is written to the FIFO with each receive packet. therefore, at least (maximum packet size/4) + 1 must be allocated for receive packets. If multiple synchronization endpoints are enabled, the space allocated for receiving consecutive packets must be at least

(Maximum packet size/4) twice + 1. Typically, the recommended space is (Maximum packet/4 + 1) twice, so that when the last packet is transmitted to the CPU, the USB can receive subsequent packets at the same time.

The transmission completion status information is pushed into the FIFO along with the last packet received by that endpoint. It is generally recommended that one location be assigned to each OUT endpoint.

Send FIFO RAM Allocation

The minimum RAM space required for each IN endpoint to send a FIFO is the maximum packet size for that particular IN endpoint.

31.5.12 USBFS System Performance

Optimal USB and system performance is achieved with large RAM buffers, highly configurable FIFO sizes, fast 32-bit FIFO access via AHB stack-in/stack-out registers, and especially advanced FIFO control mechanisms. In fact, this mechanism allows USBFS to efficiently fill the available RAM space regardless of the current USB sequence. With these features:

- The application has sufficient margin to calculate and correct the CPU load to optimize CPU bandwidth utilization:
 - Applications can accumulate bulk send data and then send it out via USB.
 - Enough time margin to read data from the receive FIFOs
- The USB module is able to maintain full-speed operation, i.e., provide maximum full-speed bandwidth (as much hardware automation as possible, as little software involvement as possible)
 - The USB module accumulates a large amount of transmission data in advance at its disposal, allowing it to manage USB data transmission autonomously.
 - Large amount of empty space in the receive buffer, which can be automatically filled with data from USB

Because the USBFS module efficiently fills the 1.25KB RAM buffer and 1.25KB of transmit/receive data is more than enough for a full-speed frame, a USB system can reach its maximum USB bandwidth in a single frame without application intervention.

31.5.13 USBFS Interrupts and Events

Global interrupts are the main interrupts to be handled by the software, and the flag bits for global interrupts can be read in the USBFS_GINTSTS register.

Table 31-2 USBFS Interrupt Event Table

interrupt flag	descriptive	operating mode	Internal Event Source
WKUPINT	Resume/remote wake-up interruptions	Host or device	-
VBUSVINT	VBUS active interrupt	installations	-
DISCINT	Disconnect Interrupt	hosts	-
CIDSCHG	Connector ID line status change interrupt	Host or device	-
PTXFE	Periodic TxFIFO air break	hosts	-
HCINT	Host Channel Interrupt	hosts	-
HPRTINT	Host Port Interrupt	hosts	-
DATAFSUSP	Data acquisition hangs	installations	-
IPXFR/INCOMPISOUT	Incomplete Cyclic Transmission/Incomplete OUT Synchronous Transmission	installations	-
IISOIXFR	Incomplete IN synchronized transmission	installations	-
OEPINT	OUT endpoint interrupt	installations	-
IEPINT	IN Endpoint Interrupt	installations	-
EOPF	End-of-periodic-frame interrupt	installations	-
ISOODRP	Discard synchronized OUT packet interrupt	installations	-
ENUMDNE	Enumeration completion	installations	-
USBRST	USB Reset Interrupt	installations	-
USBSUSP	USB hang interrupt	installations	-
ESUSP	Early pending interruptions	installations	-
GONAKEFF	Global OUT NAK active interrupt	installations	-
GINAKEFF	Global non-periodic IN NAK active interrupts	installations	-
NPTXFE	Non-periodic TxFIFO air breaks	hosts	-
RXFNE	RxFIFO Non-Air Break	Host or device	-
SOF	Start of frame interrupt	Host or device	be
MMIS	Pattern mismatch interrupt	Host or device	-

31.6 USBFS Programming Model

31.6.1 USBFS Module Initialization

The application must perform the module initialization sequence.

Refer to 31.5.2 USBFS Mode Determination for the mode determination method.

This section describes the initialization process after the USBFS controller is powered on. The initialization sequence must be followed by the application whether it is working in host mode or device mode. All module global registers are initialized according to the module configuration:

1. Program the following fields in the USBFS_GAHBCFG register:
 - Global interrupt mask bit GINTMSK = 1
 - RxFIFO not empty (RXFNE bit in USBFS_GINTSTS)
 - Periodicity TxFIFO Empty Threshold
2. Program the following fields in the USBFS_GUSBCFG register:
 - FS Timeout Calibration Field
 - USB turnaround time field
3. Software must unmask the following bits in the USBFS_GINTMSK register:
 - Mode mismatch interrupt mask
4. By reading the CMOD bit in USBFS_GINTSTS, software can determine whether the USBFS controller is operating in host or device mode.

31.6.2 USBFS Host Initialization

To initialize the module as a host, the application must perform the following steps:

1. Program the HPRTINT unmask in the USBFS_GINTMSK register.
2. Program the USBFS_HCFG register to select the full-speed host.
3. Program the PPWR bit in USBFS_HPRT to 1 to provide VBUS to the USB bus.
4. Wait for the PCDET interrupt in USBFS_HPRT. This indicates that a device is connected to the host port.
5. Program the PRST bit in USBFS_HPRT to 1 to signal a reset 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 the PENCHNG interrupt in USBFS_HPRT.
9. Read the PSPD bit in USBFS_HPRT for enumeration speed.
10. Use the selected PHY clock and set the HFIR register accordingly.
11. Program the FSLSPCS field in the USBFS_HCFG register based on the device speed detected in step 9. If the FSLSPCS is changed, a port reset must be

performed.

12. Program the USBFS_GRXFSIZ register to select the size of the receive FIFO.

13. Program the USBFS_HNPTXFSIZ register to select the size and starting address of the non-cyclic transmit FIFO used for non-cyclic communication transactions.
14. Program the USBFS_HPTXFSIZ register to select the size and start address of the periodic communication transmit FIFO used for periodic transactions.

To communicate with the device, the system software must initialize and enable at least one channel.

31.6.3 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:
 - Equipment speed
 - Non-zero length status OUT Handshake signal
2. Program the USBFS_GINTMSK register to unmask the following interrupts:
 - USB reset
 - Enumeration completion
 - Early Hanging
 - USB Hanging
 - SOF
3. Wait for VBUSVINT interrupt in USBFS_GINTSTS to indicate entering 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 approximately 10ms after this interrupt is received.
5. Wait for the ENUMDNE interrupt in USBFS_GINTSTS. This interrupt indicates the end of the reset process on the USB. Upon receiving this interrupt, 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 receive SOF packets and perform control transmissions on control endpoint 0.

31.6.4 USBFS DMA mode

The USB uses the AHB master interface to obtain transmit packet data (AHB to USB) and receive data updates (USB to AHB). The AHB master interface uses programmed DMA addresses (HCDMAx register in host mode and DIEPDMAX/DOEPDMAX register in device mode) to access the data buffer.

31.6.5 USBFS Host Programming Model

31.6.5.1 Channel initialization

The application must initialize one or more channels before it can communicate with connected devices. To initialize and enable channels, the application must perform the following steps:

1. Program the USBFS_GINTMSK register to remove interrupt masking for the following bits:
 - The non-periodic transmit FIFO for OUT transactions is empty (applies when operating at the pipelined transaction level and the packet count field is programmed with a value greater than 1)
 - The non-periodic transmit FIFO for OUT transactions is half-empty (applies when operating at the pipelined transaction level and the packet count field is programmed with a value greater than 1)
2. Program the USBFS_HAINTMSK register to enable the selected channel interrupt.
3. Program the USBFS_HCINTMSK register to enable interrupts related to communication transactions as reflected in the host channel interrupt register.
4. Program the USBFS_HCTSIZx register for the selected channel to specify the total transfer size in bytes and the expected number of packets, including short packets. The application program must program the PID field with the initial data PID (for the first OUT transaction or expected from the first IN transaction).
5. Program the USBFS_HCCHARx register of the selected channel to specify the endpoint characteristics of the device, such as type, speed, direction, etc.(Enable the channel by setting the channel enable position 1 only when the application is ready to send or receive packets.)

31.6.5.2 Channel Stop

An application can disable any channel by programming the USBFS_HCCHARx register to have CHDIS and CHENA position 1. This causes the USBFS host to clear previous requests made on that channel, if any, and generate a channel stop interrupt. The application program must wait for the CHH interrupt in USBFS_HCINTx before reassigning the channel to another communication transaction. the USBFS host will not interrupt a communication transaction that has been initiated over USB.

Before disabling the channel, the application program must ensure that there is at least one free space in the non-periodic request queue (when disabling the non-periodic channel) or the periodic request queue (when disabling the periodic channel). The application program can empty the request queue when the request queue is full (before disabling the channel) by programming the USBFS_HCCHARx register to set the CHDIS bit to 1 and clearing the

CHENA bit to zero. The application program will disable the channel when either of the following conditions occurs:

1. STALL, TXERR, BBERR, or DTERR interrupts are received in USBFS_HCINTx for the IN or OUT channel. The application program must be able to receive other interrupts (DTERR, Nak, Data, TXERR) for the same channel before receiving the channel stop signal.
2. Receives a DISCINT (Disconnect Device) interrupt from USBFS_GINTSTS. (The application will disable all enabled channels)
3. The application stops it before the transfer completes properly.

In DMA mode, the application program cannot stop an indivisible periodic transfer with a write register.

31.6.6 USBFS Device Programming Model

31.6.6.1 Endpoint endpoint initialization on USB reset

1. NAK position 1 for all OUT endpoints
 - SNAK = 1 in USBFS_DOEPCTLx (for all OUT endpoints)
2. Unmask the following interrupt bits
 - INEP0=1 in USBFS_DAINTMSK (control 0 IN endpoint)
 - OUTEP0=1 in USBFS_DAINTMSK (control 0 OUT endpoint)
 - STUP=1 in DOEPMASK
 - DOEPMASK in which XFRC=1
 - In DIEPMSK, XFRC=1
 - TOC=1 in DIEPMSK
3. Setting the data FIFO RAM for each FIFO
 - Program the USBFS_GRXFSIZ register to be able to receive OUT data and SETUP data for control transmissions. This register must be at least equal to 1 maximum packet size for control endpoint 0 + 2 words (for control OUT packet status) + 10 words (for SETUP packets)
 - Program the USBFS_TX0FSIZ register (depending on the FIFO number selected) to be able to send control IN data. This register must be at least equal to the 1 maximum packet size for control endpoint 0.
4. Program the following fields in the endpoint-related registers to enable control of OUT endpoint 0 to receive SETUP packets
 - STUPCNT=3 in USBFS_DOEPTSIZ0 (receive up to 3 consecutive SETUP packets)At this point, all the initialization required to receive SETUP packets is complete.

31.6.6.2 Endpoint initialization on USB reset

1. On the enumeration completion 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 transmissions at control endpoint 0.

31.6.6.3 Endpoint initialization when the SetAddress command is received

This section describes the actions that an application must perform when it receives a SetAddress command in a SETUP packet.

1. Use the device address received in the SetAddress command to program the USBFS_DCFG register.
2. Program the module to issue IN packets for the status phase

31.6.6.4 Endpoint initialization upon receipt of SetConfiguration/SetInterface command

This section describes the actions that must be performed when an application receives a SetConfiguration or SetInterface command in a SETUP package.

1. When the SetConfiguration command is received, the application must program the endpoint registers to configure these endpoint registers using the characteristics of the valid endpoints in the new configuration.
2. When the SetInterface command is received, the application program must program the endpoint registers for the endpoint specified by the command.
3. Endpoints that were valid in the previous configuration or other settings are not valid in the new configuration or other settings. These invalid endpoints must be deactivated.
4. Use the USBFS_DAINTMSK register to enable interrupts for valid endpoints and block interrupts for invalid endpoints.
5. Set the data FIFO RAM for each FIFO.
6. After configuring all the 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 transmit any type of packet

31.6.6.5 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 position 1
 - Endpoint initial data synchronization bits (for interrupt and bulk endpoints)
 - Endpoint Type
 - TxFIFO number
2. Once an endpoint is activated, the module starts decoding tokens sent to that endpoint and replies with a valid handshake signal if the received token is valid.

31.6.6.6 Endpoint deactivation

This section describes the steps required to deactivate 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 endpoints) or the USBFS_DOEPCTLx register (for OUT or bi-

directional endpoints) to zero.

2. Deactivating an endpoint causes the module to ignore tokens sent to that endpoint, resulting in a USB timeout.

31.6.7 USBFS Operating Model

31.6.7.1 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 the RXFNE interrupt (USBFS_GINTSTS register) is captured, the application program must read the receive status popup register (USBFS_GRXSTSP).
2. The application can mask the RXFNE interrupt (in USBFS_GINTMSK) by writing RXFNE=0 (in USBFS_GINTSTS) until it reads the packet out of the receive FIFO.
3. If the byte count of a received packet is not 0, the data is popped from the receive data FIFO and stored in memory. If the byte count of a received packet is 0, no data is popped from the receive data FIFO.
4. The packet status read from the receive FIFO has the following states:
 - Global OUT NAK:
PKTSTS=Global OUT NAK, BCNT=0x000, EPNUM and DPID values are irrelevant.
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 are now available for reading from the receive FIFO.
 - Establishment phase completed:
PKTSTS = build phase complete, BCNT = 0x0, EPNUM = control EP number, DPID value is irrelevant.
These data indicate that the build phase for the specified endpoint is complete and the data phase has been initiated. After this status entry is popped out of the receive FIFO, the module will generate a build interrupt on this control OUT endpoint.
 - OUT packet:
PKTSTS=DataOUT, BCNT=Size of received OUT packet (BCNT:0~1024),
EPNUM=Endpoint number of received packet, DPID=Actual Data PID.
 - Data transfer complete:
PKTSTS=OUT data transfer complete, BCNT=0x0, EPNUM=OUT EP number for completed data transfer, DPID value is irrelevant.
This data indicates that the transfer of OUT data for the specified OUT endpoint is complete. After this status entry is popped out of the receive FIFO, the module raises a Transfer Complete interrupt at the specified OUT endpoint.

5. After ejecting data from the receive FIFO, you must unmask the RXFNE interrupt (USBFS_GINTSTS).
6. Steps 1 through 5 are repeated each time the application detects an RXFNE interrupt in USBFS_GINTSTS. reading an empty receive FIFO may result in undefined module behavior.

SETUP transaction

This section describes the way the module handles SETUP packets and the order in which the application processes SETUP transactions. Application Program Requirements:

1. To receive SETUP packets, the control OUT endpoint STUPCNT field (USBFS_DOEPTSIZE0) must be programmed to a non-zero value. If the application programs the STUPCNT field to a non-zero value, the module receives SETUP packets and writes them to the receive FIFO regardless of the NAK state and the EPENA bit setting in USBFS_DOEPCTL0. The STUPCNT field is decremented after each SETUP packet received by the control endpoint. If the STUPCNT field is not programmed to the proper value before receiving a SETUP packet, the module can still receive SETUP packets and decrement the STUPCNT field, but the application program may not be able to determine the correct number of SETUP packets received during the establishment phase of the control transfer.
 - In USBFS_DOEPTSIZE0, 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.
 - Space is reserved for 10 words. Three words are required for the first SETUP packet, one word is required for the "build phase complete" status double word, and six 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 this space in the receive FIFO.
 - This section of the FIFO is only used to store SETUP packets; the space is never used for data packets.
3. The application program must read the 2 words of the SETUP packet from the receive FIFO.
4. The application must read and discard the Establishment Phase Complete status word from the receive FIFO.

Internal data flow:

1. When a SETUP packet is received, the module writes the received data to 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 place the IN NAK and OUTNAK of the control IN/OUT endpoint that received the SETUP packet in position 1.
2. For each SETUP packet received on the USB, the module writes 3 words of

data to the receive FIFO and decrements the STUPCNT field by one.

- The first word contains the 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 build phase ends and the data IN/OUT phase begins, the module writes a status entry (the word "build phase complete") to the receive FIFO indicating that the build phase is complete.
 4. On the AHB side, SETUP packets are read by the application program.
 5. When the application pops up the "build phase complete" word from the receive FIFO, the module will use the STUP interrupt

(USBFS_DOEPINTx) to interrupt the application program to indicate that it can process the received SETUP packet.

- The module clears the endpoint enable bit of the control OUT endpoint to zero.

Application Programming Sequence:

1. Programs the USBFS_DOEPTSI0 register.
 - STUPCNT=3
2. Wait for the RXFNE interrupt (USBFS_GINTSTS) and read the packet from the 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_DOEPTSIx register to determine the number of SETUP packets received and process the last SETUP packet received.

Processes more than three consecutive SETUP packets:

According to the USB 2.0 specification, in the case of a SETUP packet error, the host typically does not send more than three consecutive SETUP packets to the same endpoint. However, the USB 2.0 specification does not limit the number of consecutive SETUP packets that a host can send to the same endpoint. When this occurs, the USBFS controller generates 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 stops writing data other than SETUP packets to the receive FIFO. Regardless of the amount of space available in the receive FIFO, the device responds with a NAK to unsynchronized OUT tokens sent by the host and simply ignores synchronized OUT packets.
2. The module writes the global OUT NAK to the receive FIFO. the application must allow enough space for this.
3. The module sets the GONAKEFF interrupt (USBFS_GINTSTS) to 1 when the application pops the global OUT NAK word from the receive FIFO.
4. The application program detects this interrupt and assumes that the module is in global OUT NAK mode. The application can clear the interrupt by clearing the SGONAK bit in USBFS_DCTL to zero.

Application Programming Sequence:

1. To stop receiving any type of data into the receive FIFO, the application program must program the following field to set the global OUT NAK position 1.
 - In USBFS_DCTL, SGONAK=1
2. Wait for the 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 position in USBFS_DCTL, the application can receive valid OUT packets before the module raises the GONAKEFF interrupt(USBFS_GINTSTS).
3. The application program can temporarily mask this interrupt by performing a write operation to the GINAKEFFM bit in the USBFS_GINTMSK register.
 - In the USBFS_GINTMSK register, GINAKEFFM=0
 4. When the application is ready to exit the global OUT NAK mode, the SGONAK bit in USBFS_DCTL must be cleared to zero. This operation also clears the GONAKEFF interrupt (USBFS_GINTSTS).
 - In USBFS_DCTL, CGONAK=1
 5. If the application program has previously masked this interrupt, it must unmask the interrupt as follows:
 - In GINTMSK, GINAKEFFM=1

Set global OUT NAK to 1

The application program must disable the enabled OUT endpoints using the following sequence. Application Programming Sequence:

1. Before disabling any OUT endpoints, the application must enable the 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. The module also clears the following bits when the EPDISD interrupt is raised:
 - In USBFS_DOEPCTLx, EPDIS=0
 - In USBFS_DOEPCTLx, EPENA=0
5. The application program must clear the global OUT NAK bit to zero to begin receiving data from other unprohibited OUT endpoints.
 - In USBFS_DCTL, SGONAK=0

General purpose asynchronous OUT data transfer

This section describes a conventional asynchronous OUT data transfer (control, bulk or interrupt) Application

Requirements:

1. Before establishing an OUT transfer, the application program must allocate a buffer in memory to hold all the data to be received as part of the OUT transfer.
2. For OUT transmissions, the Transmit Size field in the endpoint's Transmit Size register must be a multiple of the endpoint's maximum packet size

number (and word-aligned)

- Transmission size[EPNUM] = $n \times (\text{MPSIZ}[EPNUM] + 4 - (\text{MPSIZ}[EPNUM] \bmod 4))$
- Packet Count [EPNUM] = n
- $n > 0$
- 3. When an OUT endpoint interrupt occurs, the application program 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.
- Amount of valid data in memory = Initial transfer set by the application - Remaining transfer after module update
- Number of USB packets received = initial number of packets set by application - number of packets remaining after module update

Internal data flow:

1. The application program must set the transmission size and packet count fields in the endpoint-related registers, clear the NAK bit to zero, and enable the endpoint to receive data.
2. Once the NAK bit is cleared, the module starts receiving data and writes it to the receive FIFO (as long as there is space in the receive FIFO) For each packet received on the USB, the packet and its status are written to the receive FIFO, and each packet written to the receive FIFO (a packet with a maximum packet size or a short packet) decrements the packet count field at that endpoint by one.
 - Received packets with invalid CRCs are automatically cleared from the receive FIFO.
 - After replying to an ACK for a packet on the USB, the module discards unsynchronized OUT packets that the host re-transmits because the ACK could not 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, synchronous or asynchronous packets are ignored and not written to the receive FIFO. in addition, the asynchronous OUT token 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 becomes 0 or a short packet is received on the endpoint, the NAK bit of the endpoint will be set to 1. With NAK at 1, synchronous or asynchronous packets will be ignored and will not be written to the receive FIFO while the asynchronous OUT token receives a NAK handshake answer.
4. After the data is written to the receive FIFO, the application program will read the

data from the receive FIFO and write the data to external memory, one packet at a time, coming over endpoint by endpoint.

5. After each packet is written to external memory on the AHB, the transfer size of the endpoint is automatically subtracted from the size of that packet.
6. The OUT data transfer completion status of the OUT endpoint is written to the receive FIFO in the following cases:
 - Transmit size 0 and packet count 0
 - The last OUT packet written to the receive FIFO is a short packet.

(Packet size:0~Maximum packet size-1)

7. When the application pops up this status entry (OUT Data transfer complete) and generates a transfer complete interrupt for this endpoint, and clears the

Endpoint enable bit.

Application Programming Sequence:

1. Program the USBFS_DOEPTSI_Zx register using the transfer size and the corresponding number of packets.
2. Use the endpoint feature to program the USBFS_DOEPCTLx register with EPENA and CNAK position 1.
 - In USBFS_DOEPCTLx, EPENA=1
 - In USBFS_DOEPCTLx, CNAK =1
3. Wait for the RXFNE interrupt (in USBFS_GINTSTS) and read the packet away from the receive FIFO.
 - This procedure can be repeated several times, depending on the size of the transfer.
4. Triggers the XFRC interrupt (USBFS_DOEPINTx) to indicate the successful completion of the asynchronous OUT data transfer.
5. Read the USBFS_DOEPTSI_Zx register to determine the amount of valid data.

Universal Synchronized OUT Data Transfer

This section describes regular synchronized OUT data transfers.

Application Requirements:

1. All application requirements for asynchronous OUT data transmission apply to synchronous OUT data transmission.
2. For the Transmission Size and Packet Count fields in synchronous OUT data transmissions, they must always be set to the number of packets with the maximum packet size that can be received in a single frame. Synchronous type OUT data transfer transactions must be completed in a single frame.
3. Before the end of the periodic frame (EOPF interrupt in USBFS_GINTSTS), the application must read all synchronized OUT packets (data entries and status entries) from the receive FIFO.
4. To receive data in the next frame, a synchronization OUT endpoint must be enabled before SOF(USBFS_GINTSTS) after EOPF(USBFS_GINTSTS).

Internal data flow:

1. The internal data flow of a synchronized OUT endpoint is essentially the same as that of a non-synchronized OUT endpoint, with slight differences.

-
2. When the Synchronized OUT endpoint is enabled by placing the endpoint enable position 1 and clearing the NAK bit to zero, you must place the even/odd frame position 1 accordingly. the module receives data in a specific frame on the Synchronized OUT endpoint only if the following conditions are met:
-EONUM (in USBFS_DOEPCTLx) = FNSOF [0] EONUM (in USBFS_DOEPCTLx) = FNSOF [0] (in USBFS_DSTS)
 3. When the application reads a Sync OUT packet (data and status) in its entirety from the receive FIFO, the module updates the RXDPID field in USBFS_DOEPTSIZx with the data PID of the last Sync OUT packet read from the receive FIFO.

Application Programming Sequence:

1. Programming the USBFS_DOEPTSI_Zx Register with Transfer Size and Corresponding Packet Counts
2. Use the endpoint feature to program the USBFS_DOEPCTLx register with the endpoint enable bit, clear NAK bit, and odd/even frame position 1.
 - EPENA=1
 - CNAK=1
 - EONUM = (0: even/1: odd)
3. Wait for the RXFNE interrupt (in USBFS_GINTSTS) and read the packet away from the receive FIFO.
 - This procedure can be repeated several times, depending on the size of the transfer.
4. The XFRC interrupt (in USBFS_DOEPINTx) indicates that the synchronized OUT data transfer is complete. This interrupt does not necessarily mean that the data in memory is valid.
5. For synchronized OUT transfers, the application may not always detect this interrupt. Instead, the application may detect the IISOOXFRM interrupt in USBFS_GINTSTS.
6. Read the USBFS_DOEPTSI_Zx register to determine the size of the received transmission and to determine the validity of the data received in the frame. The application program must treat data received in memory as valid only if one of the following conditions is met:
 - RXDPID=D0 (in USBFS_DOEPTSI_Zx) and the number of USB packets receiving this valid data=1
 - RXDPID = D1 (in USBFS_DOEPTSI_Zx) and the number of USB packets receiving this valid data = 2
 - RXDPID=D2 (in USBFS_DOEPTSI_Zx) and Number of USB packets receiving this valid data=3 Number of USB packets receiving this valid data=Number of initial packets programmed by the application program – Number of packets remaining after module update.

The application can discard invalid packets.

Incomplete synchronization OUT Data transfer

This section describes the application programming sequence when packet loss occurs in synchronized OUT packets. Internal data flow:

1. For synchronized OUT endpoints, the XFRC interrupt (in USBFS_DOEPINTx) may

not always be raised. If the module drops the synchronization OUT packet, the application may not detect the XFRC interrupt (USBFS_DOEPINTx) in the following cases:

- The module discards received ISO OUT data when the receive FIFO cannot hold a full ISO OUT packet.
 - Received synchronized OUT packets have CRC errors.
 - Synchronized OUT tokens received by the module are corrupted
 - Application reads data from the receiving FIFO very slowly
2. If the module detects an end-of-cycle frame before transmission is complete on all synchronization OUT endpoints, it will trigger an incomplete synchronization OUT data interrupt (IIS00XFRM in USBFS_GINTSTS) indicating that there is an untouched synchronization OUT endpoint on at least one of the synchronization OUT endpoints.

Sends an XFRC interrupt (in USBFS_DOEPINTx). At this point, the endpoint of the incomplete transfer remains enabled, but there is no active transfer in progress at that endpoint of the USB.

Application Programming Sequence:

1. Hardware-triggered IISOOXFRM interrupt (USBFS_GINTSTS) indicates that at least one of the synchronized OUT endpoints in the current frame has an unfinished transmission.
2. If this occurs because the synchronized OUT data has not been fully read from the endpoint, the application must ensure that all synchronized OUT data (including data entries and status entries) is first read away from the receive FIFO before processing continues.
 - The XFRC interrupt (USBFS_DOEPINTx) is detected by the application after all data has been read from the receive FIFO. In this case, the application must re-enable the endpoint to receive the synchronized OUT data in the next frame.
3. When the application receives the IISOOXFRM interrupt (in USBFS_GINTSTS), the application must read the control registers (USBFS_DOEPCTLx) of all synchronized OUT endpoints to determine which endpoints have incomplete transmissions in the current frame. An incomplete endpoint transmission is indicated 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 a SOF interrupt (in USBFS_GINTSTS) is detected, the previous step must be completed to ensure that the current frame number has not changed.
5. For synchronized OUT endpoints with incomplete transfers, the application must discard the data in memory and disable the endpoint by placing EPDIS position 1 in USBFS_DOEPCTLx.
6. Wait for the EPDIS interrupt (in USBFS_DOEPINTx) and enable the endpoint to receive new data in the next frame.
 - Since it may take some time for the module to disable the endpoint, the application may not be able to receive the data in the next frame after receiving invalid synchronization data.

Stop asynchronous OUT endpoints

This section describes how an application can stop unsynchronized endpoints.

1. Places the module in global OUT NAK mode.
2. Disable the required endpoints
 - When disabling endpoints, set STALL=1 (in USBFS_DOEPCTL) instead of SNAK

position 1 in USBFS_DOEPCTL. The STALL bit always has a higher priority than the NAK bit.

3. The STALL bit (in USBFS_DOEPCTLx) must be cleared to zero when the application no longer needs the endpoint to respond to the STALL handshake signal.
4. If the application receives the host's SetFeature.Endpoint Halt Endpoint Halt command from the host, the STALL state of the endpoint must be set or cleared on that

Set STALL position 1 or clear to zero before transferring the status phase on the control endpoint.

31.6.7.2 IN data transmission

Packet Write

This section describes how the application program writes packets to the endpoint FIFO when the dedicated transmit FIFO has been enabled.

1. The application can select either polling mode or interrupt mode.
 - In polling mode, the application program monitors the status of the endpoint transmit data FIFO by reading the USBFS_DTXFSTSx register to determine if there is enough space in the data FIFO.
 - In interrupt mode, the application waits for the TXFE interrupt (in USBFS_DIEPINTx) and then reads the USBFS_DTXFSTSx register to determine if there is enough space in the data FIFO.
 - To write a single non-zero length packet, there must be enough space in the data FIFO to hold the entire packet.
 - To write zero-length packets, the application cannot view FIFO space.
2. If one of the above methods is used, when the application program determines that there is enough space to write the transmit packet, the application program must first perform a corresponding write operation to the endpoint control register before writing the data to the data FIFO. Typically, the application program must perform a **read-modify-write** operation to the USBFS_DIEPCTLx register to avoid modifying other contents of the register at the same time as it enables the endpoint to position 1. register while enabling the endpoint at position 1.

If there is enough space, the application program can write multiple packets from the same endpoint to the transmit FIFO. For periodic IN endpoints, the application program can only write multiple packets from one frame at a time. The application program writes all packets to be sent in the next frame only after the communication transaction of the previous frame has been transmitted.

Set IN endpoint NAK to 1

Internal data flow:

1. When the application sets IN NAK to 1 for a specific endpoint, the module stops data transmission on the endpoint, regardless of the availability of data in the endpoint's transmit FIFO.
2. The asynchronous endpoint receives the IN token and replies with a NAK handshake

answer.

The synchronization endpoint receives an IN token and returns a zero-length packet. Synchronized endpoint receives IN token, returns zero-length packet

3. The module triggers INEPNE (IN) in USBFS_DIEPINTx. Endpoint NAK (valid) interrupt in USBFS_DIEPCTLx in response to the SNAK bit in USBFS_DIEPCTLx.
4. When the application program detects this interrupt, it assumes that the endpoint is in IN NAK mode. The application can clear the interrupt by setting CNAK in USBFS_DIEPCTLx to position 1.

Application Programming Sequence:

1. To stop sending any data at a specific IN endpoint, the application must place the IN NAK position 1. To place this position 1, the following fields must be programmed.
 - SNAK=1 in USBFS_DIEPCTLx
2. Wait for the INEPNE interrupt in USBFS_DIEPINTx to trigger. This interrupt indicates that the module has stopped sending data at the endpoint.
3. The module can send valid IN data on the endpoint when the application program has set NAK position 1 but the "NAK valid" interrupt has not been triggered.
4. The application program can temporarily mask this interrupt by writing the INEPNEM bit in DIEPMSK.
 - In DIEPMSK, INEPNEM = 0
5. To exit endpoint NAK mode, the application program must clear the NAK status bit (NAKSTS) in USBFS_DIEPCTLx to zero. This action also clears the INEPNE interrupt (in USBFS_DIEPINTx)
 - In USBFS_DIEPCTLx, CNAK=1
6. If the application has masked the interrupt, it must be unmasked as follows:
 - In DIEPMSK, INEPNEM=1

Disable IN endpoints

Use the following sequence to disable a specific IN endpoint that has been previously enabled. Application Programming Sequence:

1. The application program must stop writing data to the AHB before it can disable the IN endpoint.
2. The application must set the endpoint to NAK mode.
 - SNAK=1 in USBFS_DIEPCTLx
3. Wait for the INEPNE interrupt in USBFS_DIEPINTx.
4. Place the following position 1 in the USBFS_DIEPCTLx register of the endpoint 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. The module also clears the

following bits when the interrupt is triggered:

- In USBFS_DIEPCTLx, EPENA=0
- In USBFS_DIEPCTLx, EPDIS=0
- 6. The application must read the USBFS_DIEPTSI_Zx register for the periodic IN EP to calculate how much data is being sent over the USB on the endpoint.
- 7. The application must clear the endpoint transmit FIFO by setting the following field in the USBFS_GRSTCTL register to 1

The data in the

- 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 to zero, which indicates the end of the FIFO clear operation. To send new data on this endpoint, the application can re-enable the endpoint later.

Universal Non-Cyclic IN Data Transfer

Application Requirements:

1. Before establishing an IN transfer, the application must ensure that each packet comprising an IN transfer can fit in a single buffer.
2. For IN transmissions, the Transmit Size field in the Endpoint Transmit Size register indicates the amount of valid data for this transmission, which consists of multiple maximum packet sizes and a single short packet. This short packet is sent at the end of the transmission.
 - To send multiple maximum packet size packets and add a short packet at the end of the transmission: transmission size[EPNUM] = $x \times \text{MPSIZ}[EPNUM] + sp$
If ($sp > 0$), packet count[EPNUM] = $x + 1$.
Otherwise, packet count[EPNUM] = x
 - To send a single zero-length packet:
transmission size [EPNUM] = 0
Packet Count [EPNUM] = 1
 - To send multiple packets of maximum packet size and add a zero-length packet at the end of the transmission, the application must split the transmission into two parts.
The first part sends packets of maximum packet size and the second part sends only zero-length packets.
First transmission: transmission size [EPNUM] = $x \times \text{MPSIZ}[epnum]$;
packet count = n ; second transmission: transmission 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 program must read the transfer size register to determine how much of the data that was fed into the transmit FIFO has been sent out over the USB.
4. Amount of data fed into the sending FIFO = initial transfer size programmed by the

application - final transfer size after module update

- Amount of data already sent via USB = (Initial packet count programmed by the application - Final packet count after module update) × MPSIZ[EPNUM]
- Remaining amount of data to be sent via USB = (Initial transfer size programmed by the application - amount of data already sent via USB)

Internal data flow:

1. The application must set the transmission size and packet count fields in the registers of a particular endpoint and enable that endpoint to send data.
2. The application must also write the necessary data to the transmit FIFO of this endpoint.
3. Each time the application writes a packet to the transmit FIFO, the transfer size of that endpoint is automatically subtracted from the packet size. The application continues to fetch data from memory to write to the transmit FIFO until the transfer size of the endpoint becomes 0. After writing data to the FIFO, the packet count in FIFO is incremented (this is a 3-bit count that is maintained internally by the module, one for each IN endpoint transmit FIFO). The maximum number of packets maintained by the module in an IN endpoint FIFO is always eight. For zero-length packets, each FIFO has a separate flag and there is no data in the FIFO.
4. When data is written to the transmit FIFO, the module sends this data out when it receives an IN token. After each packet is sent and an ACK handshake signal is received in return, the packet count at that endpoint is decremented by one until the packet count becomes zero. When a timeout occurs, the packet count is not decremented.
5. For zero-length packets (indicated by the internal zero-length flag) the module issues a zero-length packet in response to the IN token and decrements the value of the packet's 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 When TxFIFO is Empty (ITTXFE) interrupt for that endpoint (provided the NAK bit is not set to 1 for that endpoint). The module replies with a NAK handshake signal on the asynchronous endpoint.
7. The module will internally cause the FIFO pointer to return to the beginning and will not generate a timeout interrupt.
8. When the Transmit Size is 0 and the Packet Count is 0, a Transmit Complete (XFRC) interrupt is generated for that endpoint and the endpoint enable is cleared to zero.

Application Programming Sequence:

1. Program the USBFS_DIEPTSI $_Z$ register using the transfer size and the corresponding packet count.
2. Use the endpoint feature to program the USBFS_DIEPCTL $_Z$ register with CNAK and EPENA (endpoint enable) position 1.

-
3. When sending a non-zero length packet, the application must poll the USBFS_DTXFSTS x register (where x is the number of the FIFO associated with that endpoint) to determine if there is enough space in the data FIFO. The TXFE bit (in USBFS_DIEPINT x) is also optional for the application before writing data.

Universal Periodic IN Data Transfer

This section describes typical periodic IN data transfers.

Application Requirements:

1. Application requirements 1, 2, 3, and 4 for generalized non-periodic IN data transfers apply equally to periodic IN data transfers (with minor modifications to requirement 2)
 - The application can only send a number of maximum packet size packets or a number of maximum packet size packets plus a short packet at the end of the transmission.
To send multiple maximum packet size packets and add a short packet at the end of the transmission, the following condition must be satisfied: transmission size[EPNUM] = $x \times \text{MPSIZ}[\text{EPNUM}] + \text{sp}$
(where x is an integer greater than 0 and sp ranges from 0 ~ $\text{MPSIZ}[\text{EPNUM}] - 1$) If ($\text{sp} > 0$), packet count[EPNUM] = $x + 1$
Otherwise, packet count[EPNUM] = x ;
 $\text{MCNT}[\text{EPNUM}] = \text{packet count}[\text{EPNUM}]$
 - An application cannot send a zero-length packet at the end of a transmission. The application can send a separate zero-length packet.
 - To send a single zero-length packet:
transmission size [EPNUM]=0
packet count [EPNUM]=1
 $\text{MCNT}[\text{EPNUM}] = \text{Packet Count}[\text{EPNUM}]$
2. The application can only schedule one frame of data transfer at a time.
 - $(\text{MCNT} - 1) \times \text{MPSIZ} \leq \text{XFERSIZ} \leq \text{MCNT} \times \text{MPSIZ}$
 - $\text{PKTCNT} = \text{MCNT}$ (in `USBFS_DIEPTSIZx`)
 - If $\text{XFERSIZ} < \text{MCNT} \times \text{MPSIZ}$, 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`, and XFERSIZ is located in `USBFS_DIEPTSIZx`.
3. Before receiving the IN token, the application program must write the complete data to be sent in the frame to the transmit FIFO. When the IN token is received, the module performs the operation when the FIFO is empty, even if the data to be

sent in the frame is only one double word short of being written in the send FIFO.

When the transmit FIFO is empty:

- Zero-length packets will be replied to on synchronized endpoints
- The NAK handshake signal will be returned on the interrupt endpoint.

Internal data flow:

1. The application must set the transmission size and packet count fields in the registers of a particular endpoint and enable that endpoint to send data.
2. The application must also write the required data to the transmit FIFO associated with this endpoint.
3. Each time the application writes a packet to the transmit FIFO, the transfer size of that endpoint is automatically subtracted from the packet size. The application continues to fetch data from memory to write to the transmit FIFO until the transfer size of the endpoint becomes zero.
4. When the periodic endpoint receives an IN token, the module will start sending data in the FIFO (if there is data in the FIFO) If the FIFO does not contain a complete packet of the data to be sent for that frame, the module generates a "Received IN Token when TxFIFO is Empty" interrupt for that endpoint.
 - Zero-length packets will be replied to on synchronized endpoints
 - The NAK handshake signal will be returned on the interrupt endpoint.
5. The packet count of an endpoint is decremented by 1:
 - For synchronized endpoints, when sending a zero-length or non-zero-length packet
 - For interrupt endpoints, decremented when the ACK handshake signal is sent.
 - When both the transmission size and packet count are 0, a transmission completion interrupt is generated for that endpoint and the endpoint enable bit is cleared to zero.
6. During the Periodic Frame Interval (controlled by the PFIVL bit in USBFS_DCFG), the module generates an IISOIXFR interrupt in USBFS_GINTSTS when it discovers that data has not been sent to any of the synchronized IN endpoint FIFOs that should be empty during the current frame.

Application Programming Sequence:

1. Use the endpoint feature to program the USBFS_DIEPCTLx register with CNAK and EPENA position 1.
2. Writes the data that needs to be sent in the next frame to the transmit FIFO.
3. The hardware triggered ITTXFE interrupt (in USBFS_DIEPINTx) indicates that the application program has not yet written all the data it needs to send to the send FIFO.
4. If the interrupt endpoint is enabled before an interrupt is detected, the interrupt is ignored. If the interrupt endpoint is not enabled, this endpoint is enabled so that data can be sent out when the next IN token is received.
5. Hardware triggered XFRC interrupt (in USBFS_DIEPINTx) indicates successful completion of synchronous IN transfer if no ITTXFE interrupt is generated in

USBFS_DIEPINTx. When reading USBFS_DIEPTSIz register, it always gets Transmit Size=0 and Packet Count=0, then it means all data has been sent over USB.

6. Setting the XFRC interrupt (in USBFS_DIEPINTx) indicates successful completion of the interrupt IN transfer, regardless of whether the ITTXFE interrupt (in USBFS_DIEPINTx) is generated. When reading USBFS_DIEPTSIz register, it always gets Transmit Size = 0 and Packet Count = 0, then all data has been sent over USB.
7. Failure to generate any of the preceding interrupts when the Outstanding Synchronous IN Transmission (ISOIXFR) interrupt is set in USBFS_GINTSTS indicates that the module has not received at least 1 periodic IN token in the current frame.

Synchronization IN data transfer not completed

This section describes the actions that the application must perform for incomplete synchronized IN data transfers.

Internal Data Flow:

1. Synchronous IN transmission is considered incomplete when one of the following conditions is met:
 - a) The module receives a corrupt Sync IN token on at least one Sync IN endpoint. At this point, the application detects an incomplete synchronization IN transfer interrupt (IISOIXFR bit in USBFS_GINTSTS)
 - b) The application program writes data to the transmit FIFO too slowly and receives an IN token before the full data is written to the FIFO. At this point, the application detects the "IN token received when TxFIFO is empty" interrupt in USBFS_DIEPINTx. The application can ignore this interrupt as it will eventually generate an incomplete synchronized IN transfer interrupt (IISOIXFR bit in USBFS_GINTSTS) end of the 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 sending FIFO as soon as possible.
3. The application program must set the NAK bit of the endpoint and disable position 1.
4. The module disables the endpoint by clearing the disable bit and triggering an "endpoint disabled" interrupt on the endpoint.

Application Programming Sequence:

1. Applications can ignore the "IN token received when TxFIFO is empty" interrupt in USBFS_DIEPINTx on any synchronized IN endpoint, as this will eventually generate an incomplete synchronized IN transfer interrupt (in USBFS_GINTSTS)
2. A hardware-triggered incomplete sync IN transfer interrupt (in USBFS_GINTSTS) indicates that there is an incomplete sync IN transfer on at least one sync IN endpoint.
3. The application must read the Endpoint Control registers of all synchronized IN endpoints to detect the presence of an endpoint that has not completed an IN data transfer.
4. The application program must stop writing to the "periodic transmit FIFO" associated with these endpoints.
5. Program the following fields in the USBFS_DIEPCTLx register to disable endpoints:
 - SNAK=1 in USBFS_DIEPCTLx
 - EPDIS=1 in USBFS_DIEPCTLx

6. A hardware triggered "Endpoint Disable" interrupt in USBFS_DIEPINTx indicates that the module has disabled the endpoint.
 - At this point, the application must either clear the data in the associated transmit FIFO or overwrite the existing data in the FIFO by enabling the newly transmitted endpoint in the next frame. To flush the data, the application must use the USBFS_GRSTCTL register.

Stop unsynchronized IN endpoints

This section describes how an application can stop unsynchronized endpoints.

Application Programming Sequence:

1. Disables the IN endpoint to be stopped. Also set STALL position 1.
2. EPDIS=1 in USBFS_DIEPCTLx (when endpoint is enabled)
 - STALL=1 in USBFS_DIEPCTLx
 - The STALL bit always has a higher priority than the NAK bit.
3. A hardware triggered "endpoint disable" interrupt (in USBFS_DIEPINTx) lets the application know that the module has disabled the specified endpoint.
4. The application must empty either the nonperiodic or periodic transmit FIFO depending on the endpoint type. For nonperiodic endpoints, the application must re-enable another nonperiodic endpoint that does not need to be stopped in order to send data.
5. The STALL bit of USBFS_DIEPCTLx must be cleared to zero when the application is ready to end the STALL handshake signal for this endpoint.
6. If the application program sets or clears the STALL state of an endpoint as a result of receiving a SetFeature.Endpoint Halt command or ClearFeature.Endpoint Halt command from the host, the STALL must be set to position 1 or cleared to zero before the state phase of that control endpoint is transmitted.

Special case: Stop control OUT endpoint

If the host sends more IN/OUT tokens than specified in the SETUP packet during the data phase of the control transfer, the module must reply STALL for these excess IN/OUT tokens, in which case the application program must enable the ITTXFE interrupt for USBFS_DIEPINTx and the OTEPDIS interrupt for USBFS_DIEPINTx (after the module has finished transferring the amount of data specified in the SETUP packet) during the data phase of the control transfer. DOEPINTx's OTEPDIS interrupt during the data phase of the control transfer (after the module has finished transferring the amount of data specified in the SETUP packet) Subsequently, when the application program receives this interrupt, it must set STALL in the corresponding endpoint control register to 1 and clear the interrupt.

31.7 Register Description

The application program controls the USBFS module by reading and writing to the control and status registers through the AHB slave interface. All registers of the USBFS module are 32-bit registers and their addresses are 32-bit aligned, 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

The USBFS system control register, which has a different base address than the other registers, is independent of the USBFS module in order to control USBFS module-related settings.

Only the module global registers, power and clock gating control registers can be accessed in host and device modes. When the USBFS module is in one mode (host or device), the application program must not access the registers in another role mode, such as accessing device mode registers while in host mode. If an illegal access occurs, a mode mismatch interrupt will be generated and reflected in the **USBFS.GINTSTS.NMIS** bit of the module interrupt register. When the module is switched from one role mode to another, the registers in the new operating mode must be reprogrammed to the state they were in after the power-on reset.

Please refer to Table 31-3 to Table 31-4 USBFS Module Register List for the USBFS Module register list and base address.

Table 31-3 USBFS System Control Register List

USBFS system control register base address: 0x40055400

(USBFS System Control Register) Register Name	offset address	reset value
USBFS System Control Register (USBFS_SYCTLREG)	0x00	0x0000 0000

Table 31-4 USBFS System Control Registers List

USBFS module register base address: 0x400C0000

(USBFS Global Register) Register Name	offset address	reset value
USBFS VBUS control register (USBFS_GVBUSCFG)	0x00	0x0000 0000
USBFS AHB Control Register (USBFS_GAHBCFG)	0x08	0x0000 0000
USBFS USB Configuration Register (USBFS_GUSBCFG)	0x0c	0x0000 0A00
USBFS Reset Register (USBFS_GRSTCTL)	0x10	0x8000 0000
USBFS Global Interrupt Register (USBFS_GINTSTS)	0x14	0x1400 0020
USBFS Global Interrupt Mask Register (USBFS_GINTMSK)	0x18	0x0000 0000
USBFS Receive Status Debug Read Register (USBFS_GRXSTSR)	0x1c	0x0000 0000
USBFS Receive status read and out register (USBFS_GRXSTSP)	0x20	0x0000 0000
USBFS Receive FIFO size register (USBFS_GRXFSIZ)	0x24	0x0000 0140
USBFS Host Non-Cyclic Transmit FIFO Size Register (USBFS_HNPTXFSIZ)/Endpoint 0 Transmit FIFO Size Register (USBFS_DIEPRXF0)	0x28	0x0200 0140
USBFS Non-Cyclic Transmit FIFO/Queue Status Register (USBFS_HNPTXSTS)	0x2c	0x0008 0100
USBFS Module ID register (USBFS_CID)	0x3c	0x1234 5678
USBFS Host Periodic Transmit FIFO Size Register (USBFS_HPTXFSIZ)	0x100	0x0000 0000
USBFS Device IN Endpoint x Transmit FIFO Size Register (USBFS_DIEPTXFx)	0x100+x*4(x=1~5)	0x0100 0240+(x- 1)*0x100

(USBFS Host Control and Status Registers) Register Name	offset address	reset value
USBFS Host Configuration Register (USBFS_HCFG)	0x400	0x0000 0000
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 0100
USBFS Host All 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 Characterization Register (USBFS_HCCHARx)	0x500+x*0x20(x=0~11)	0x0000 0000
USBFS host channel x interrupt register (USBFS_HCINTx)	0x508+x*0x20(x=0~11)	0x0000 0000
USBFS host channel x interrupt mask register (USBFS_HCINTx)	0x50c+x*0x20(x=0~11)	0x0000 0000
USBFS Host Channel x Transfer Size Register (USBFS_HCTSIZx)	0x510+x*0x20(x=0~11)	0x0000 0000
USBFS host channel x DMA address register (USBFS_HCDMAx)	0x514+x*0x20(x=0~11)	0xFFFF XXXX

(USBFS Device Control and Status Register)	offset address	reset value
Register Name		
USBFS Device Configuration Register (USBFS_DCFG)	0x800	0x0820 0000
USBFS Device Control Register (USBFS_DCTL)	0x804	0x0000 0000
USBFS Device Status Register (USBFS_DSTS)	0x808	0x0000 0002
USBFS Device IN Endpoint Generic Interrupt Mask Register (USBFS_DIEPMSK)	0x810	0x1400 0000
USBFS Device OUT Endpoint Generalized Interrupt Mask Register (USBFS_DOEPMSK)	0x814	0x0000 0000
USBFS Device All Endpoint Interrupt Register (USBFS_DAINT)	0x818	0x0000 0000
USBFS Device All Endpoint Interrupt Mask Register (USBFS_DAINTMSK)	0x81c	0x0000 0000
USBFS Device IN Endpoint FIFO Air Break Mask Register (USBFS_DIEPEMPMSK)	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(n=1~5)	0x0000 0000
USBFS Device IN Endpoint x Interrupt Register (USBFS_DIEPINTx)	0x908+x*0x20(n=0~5)	0x0000 0000
USBFS Device IN Endpoint x Transfer Size Register (USBFS_DIEPSIZx)	0x910+x*0x20(n=0~5)	0x0000 0000
USBFS device IN endpoint x DMA address register (USBFS_DIEPDMAx)	0x914+x*0x20(n=0~5)	0x0000 0000
USBFS Device IN Endpoint x Transmit FIFO Status Register (USBFS_DTXFSTSx)	0x918+x*0x20(n=0~5)	0x0000 0000
USBFS Device OUT Endpoint 0 Control Register (USBFS_DOEPCTL0)	0xb00	0x0000 8000
USBFS Device OUT Endpoint x Control Register (USBFS_DOEPCTLx)	0xb00+x*0x20(n=1~5)	0x0000 0000
USBFS Device OUT Endpoint x Interrupt Register (USBFS_DOEPINTx)	0xb08+x*0x20(n=0~5)	0x0000 0000
USBFS Device OUT Endpoint x Transfer Size Register (USBFS_DOEPSIZx)	0xb10+x*0x20(n=0~5)	0x0000 0000
USBFS Device OUT Endpoint x DMA Address Register (USBFS_DOEPDMAx)	0xb14+x*0x20(n=0~5)	0xXXXXXXXX

(USBFS Power and Clock Empty Control Register)	offset address	reset value
Register Name		
USBFS Power and Gated Clock Control Register (USBFS_PCGCCTL)	0xe00	0x0000 0000

31.7.1 USBFS System Control Register

31.7.1.1 USBFS System Control Register (USBFS_SYCTLREG)

USBFS 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
Reserved															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved															
classifier for marking celebrity functionality honorific people															
fill out or in (informa tion on a form)															
b31~b2	Reserved	-	The reset value must be maintained.												R
b1	SOFEN	SOF pulse output enable bit	When the host sends a SOF or the device successfully receives a SOF, a 16 system clock cycle width SOF pulse is enabled from the PAD outputs 0: SOF pulse not output 1: SOF pulse output Note: Access is available in both device mode and host mode.												R/W
b0	DFB	VBUS/ID pin internal de-jitter filter bypass enable bit	VBUS/ID pin module internal de-jitter filter bypass enable bit 0: Module internal de-jitter filter active 1: Bypass module internal de-jittering filter Note: Access is available in both device mode and host mode.												R/W

31.7.2 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 expressed in binary unless otherwise noted.

31.7.2.1 USBFS VBUS Control Register (USBFS_GVBUSCFG)

VBUS Configuration Register

Offset Address: 0x00

Reset value: 0x0000 0000

This register can be used to set the VBUS value and thus ignore 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															

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b8	Reserved	-	The reset value must be maintained.	R
b7	VBUSVAL	VBUS Value	<p>VBUS Value</p> <p>Used to set the VBUS value of the USBFS, when set to 1 and after VBUSOVEN is set to 1 on the R/W</p> <p>USBFS completes power-up.</p> <p>Note: Accessible only in device mode.</p>	USB The R/W
b6	VBUSOVEN	VBUS Override enable	<p>VBUS Override Enable (VBUS Override Enable)</p> <p>is used to reflect the value set by VBUSVAL to the state of the FS CORE.</p> <p>value of VBUSVAL is valid only if this bit sets to 1.</p> <p>Note: Accessible only in device mode.</p>	
b5~b0	Reserved	-	The reset value must be held.	R

31.7.2.2 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 contains mainly configuration parameters related to the AHB system.

The application program must program this register before starting any AHB or USB transactions. 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								PTX FEL VL	TXF ELV L	Res erv ed	DMA EN	HBSTLEN[3:0]			GIN TMS K

classifier for honorific people	marking	celebrity	functionality	fill out or in (informa tion on a form)
b31~b9	Reserved	-	The reset value must be maintained.	R
b8	PTXFELVL	Periodic TxFIFO Empty Threshold Limit	Periodic TxFIFO null threshold (Periodic Indicates when the periodic TxFIFO air break bit in the module interrupt register is triggered (PTXFE bit in USBFS_GINTSTS) 0: PTXFE (located at USBFS_GINTSTS) interrupt indicates that the periodic TxFIFO is half empty 1: PTXFE (located at USBFS_GINTSTS) interrupt indicates that the periodic TxFIFO is fully empty Note: Accessible in host mode only.	R/W
b7	TXFELVL	Device TxFIFO Empty Threshold Limit	Device TxFIFO empty level In device mode, this bit indicates when the IN endpoint is triggered to send a FIFO air break (TXFE in USBFS_DIEPINTx). 0: TXFE (at USBFS_DIEPINTx) interrupt indicates that the IN endpoint TxFIFO is half empty 1: TXFE (located at USBFS_DIEPINTx) interrupt indicates that the IN endpoint TxFIFO is fully empty Note: Accessible in host mode only.	R/W
b6	Reserved	-Reset value must be maintained.		The reset
b5	DMAEN	DMA Enable	value must be held. DMA enable b4~b1 HBSTLEN Batch Length/Type	0: Module operates in slave mode 1: Module runs in DMA mode

mode and host mode.

Burst length/type 0000b:single

0001b: INCR

0011b:INCR4

0101b:INCR8

0111b:INCR16

R

Other values: reserved

Note: Access is available in both device

mode and host mode.

b0 GINTMSK Global interrupt mask Global interrupt mask

R/W

This bit is used to mask global interrupts or to unmask global interrupts.

The interrupt status register is updated by the module independent of the setting of this bit.

0: Mask application-triggered interrupts

1: Unmasking application-triggered

interrupts Note: Accessible in both device mode and host mode.

31.7.2.3 USBFS USB Configuration Register (USBFS_GUSBCFG)

USBFS USB configuration register

offset address: 0x00C

Reset value: 0x0000 0A00

This register can be used to configure the module after power-up or change of role mode. It contains configuration parameters related to USB and USB-PHY.

The application program must program this register before starting any AHB or USB transactions. Do not change this register after initial programming.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Rese rv ed	FDM OD	FHM OD													Reserved
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved					TRDT[3:0]			Reserved	PHY SEL		Reserved				TOCAL[2:0]

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31	Reserved	-	The reset value must be maintained.	R
b30	FDMOD	Forced Device Mode	<p>Forced Device Mode (device mode) (Force)</p> <p>Writing a 1 to this bit forces the module into device mode, ignoring the USBFS_ID input pin's Status.</p> <p>0: Normal mode, depending on the input status of the USBFS_ID pin R/W</p> <p>1: Forced device mode</p> <p>After forcing position 1, the application must wait at least 25 ms for the change to take effect.</p> <p>Note: Access is available in both device mode and host mode.</p>	
b29	FHMOD	Forced Host Mode	<p>Force host mode</p> <p>Writing a 1 to this bit forces the module into host mode, ignoring the USBFS_ID input pin's Status.</p> <p>0: Normal mode, depending on the input status of the USBFS_ID pin R/W</p> <p>1: Forced host mode</p> <p>After forcing position 1, the application must wait at least 25 ms for the change to take effect.</p> <p>Note: Access is available in both device mode and host mode.</p>	
b28~b14	Reserved	-	<p>The reset value must be maintained.</p> <p>USB turnaround time</p> <p>Set the turnaround time in number of PHY clocks.</p> <p>To calculate the value of TRDT, use the following formula:</p>	R

			TRDT = 4 × AHB clock + 1 PHY clock	
b13~b10	TRDT	USB	Example: 1. If AHB clock frequency = 84 MHz (PHY clock frequency = 48 MHz), TRDT is set to 9. 2. If AHB clock frequency = 48 MHz (PHY clock frequency = 48 MHz), TRDT is set to 5.	R/W
Turnaround Time			Note: Access is available in both device mode and host mode.	
b9~b7	Reserved	-Reset value must be maintained. value must be held.	R	The reset

			Full Speed serial select	transceiver
b6	PHYSEL	Full Speed Series Transceiver Options	This bit is write-only and always 1.	W
b5~b3	Reserved	-	The reset value must be maintained.	R
b2~b0	TOCAL Calibration	FS Timeout	<p>FS timeout calibration</p> <p>The additional delay introduced by the PHY includes the number of PHY clocks set by the application in this field, as well as the module's full-speed inter-packet timeout interval. The effect of the delay introduced by different PHYs on the state of the data line is different. The standard USB timeout value for full-speed operation is 16 to 18 (inclusive) bit times. The application program must program this field based on the enumeration speed. The number of bits of time added per PHY clock is 0.25 bits of time.</p>	R/W
<p>Note: Access is available in both device mode and host mode.</p>				

31.7.2.4 USBFS reset register (USBFS_GRSTCTL)

USBFS reset register

offset address: 0x10

Reset value: 0x8000 0000

The application program resets each hardware feature in the module through this register.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17
AHB IDL	DMA REQ													
b16 b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1
Reserved b0				TXFNUM[4:0]				TXF FLS H	RXF FLS H	Rese rv ed	FCR ST	HSR ST	CSR ST	

classifier for	marking	celebrity	Function Read/Write
honorific			AHB master idle (AHB master idle)
people			
b31	AHBIDL	AHB Master Device Idle	Indicates that the AHB master device state machine is in an idle condition.
			Note: Access is available in both device mode and host mode.
b30	DMAREQ	AHB Master Device Idle	This
		bit indicates that a DMA request is in progress. It is used for debugging.	R
		Note: Access is available in both device mode and host mode.	
b29~b11	Reserved	-0" for reading, "0" for writing. Reads "0", writes "0".	R
b10~b6	TXFNUM	TxFIFO number(TxFIFO number) The FIFO number of the FIFO to be FIFO refreshed using the TxFIFO Refresh bit. This field can only be changed after the module has cleared the TxFIFO refresh bit to zero. <ul style="list-style-type: none"> ● 00000: – Host mode flushing of non-periodic TxFIFOs – Refresh Tx FIFO in device mode 0 ● 00001: – Host mode refresh of periodic TxFIFOs – Refresh TXFIFO in device mode 1 ● 00010: Refresh TXFIFO in device mode 2 --- ● 00101: Refresh TXFIFO in device mode 15 ● 10000: Flush all transmit FIFOs in device mode or host mode Note: Accessible in both device mode and host mode. 	R/W

TxFIFO flush

This bit selectively refreshes one or all of the transmit FIFOs, but the operation cannot be performed while the module is processing communication transactions.

b5	TXFFLSH	TxFIFO Refresh	The application program can only write to this bit after confirming that the module is not currently reading or writing to the TxFIFO. operation. Use the following registers for confirmation: – Read: A valid NAK interrupt ensures that the module is not currently performing a read operation on the FIFO. – Write: The AHBIDL bit in USBFS_GRSTCTL ensures that the module is not currently performing any write operations to the FIFOs. Note: Access is available in both device mode and host mode.	R/W
----	---------	----------------	--	-----

			RxFIFO flush (RxFIFO flush) The application can use this bit to refresh the entire RxFIFO, but it must first ensure that the module is not currently processing communications. The application program can only write to this bit if it has confirmed that the module is not currently reading or writing to the RxFIFO. The application program can perform a write operation on this bit only after confirming that the module is not currently performing a read or write operation on the RxFIFO.	
b4	RXFFLSH	RxFIFO Refresh	Operation. The application program must wait until this bit is cleared before performing other operations. Typically it is necessary to wait 8 clock cycles (in whichever is the slowest of the PHY or AHB clocks) Note: Access is available in both device mode and host mode.	R/W
b3	Reserved	-	The reset value must be maintained.	R
b2	FCRST	Host frame counter reset	The frame number counter in the module is reset when the application program performs a write operation to this bit. 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	and clear all transmit FIFOs and receive FIFOs.
b1	HSRST	HCLK domain logic soft reset	host mode. 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. FIFOs are not flushed by this bit. Upon termination of a transaction on the AHB in compliance with the protocol, all state machines in the AHB clock domain are reset to the idle state. The CSR control bits used by the state machines in the AHB clock domain are cleared to zero. To clear this interrupt, the status mask bit generated by the AHB clock domain state machine and used to control the interrupt state needs to be cleared. Since the interrupt status bit is not cleared, the application program can obtain the status of all module events occurring after position 1. Status. This bit is a self-clearing bit and the module will clear this bit after all the necessary logic in it has been reset. This process takes several clocks of time, depending on the current state of the module. NOTE: Accessed in	Terminate all transactions on the AHB master device as soon as possible after the final data phase of the AHB transfer.
b0	CSRST	Module soft reset	both device mode and host mode. Core soft reset resets the HCLK and PCLK domains as described below: Clear each interrupt and all CSR register bits to zero except for the following bits: <ul style="list-style-type: none">– RSTPDMODL bit in USBFS_PCGCCTL– GAYEHCLK bit in USBFS_PCGCCTL– PWRCOMP bit in USBFS_PCGCCTL– STPPCLK bit in USBFS_PCGCCTL– FSLSPCS bit in USBFS_HCFG– DSPD bit in USBFS_DCFG Reset all module state machines (except AHB slave devices) to the idle state	Terminate all transactions on the USB immediately. The application program can perform a write operation to this bit any time it needs to reset the module.

This bit is a self-clearing bit that will be cleared by the module after all necessary logic in it has been reset, a process that takes a number of clocks depending on the current state of the module. Once this bit is cleared, software must wait at least 3 PHY clocks before accessing the PHY domain.

R/W

(synchronization delay) In addition, the software

It must also be determined that bit 31 in this register is set to 1 (AHB master device idle) before operation can begin.

R/W

Software reset is typically used in two situations, during software development and after the user dynamically changes the PHY select bit in the USB configuration registers listed above. When the user changes the PHY, the appropriate clock is selected for the PHY and used in the PHY domain. Once the new clock is selected, the PHY domain must be reset to ensure proper operation.

Note: Access is available in both device mode and host mode.

R/W

31.7.2.5 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 in the current mode (device mode or host mode) with the help of system level events.

Some bits in this register are only valid in host mode, while others are only valid in device mode. In addition, this register indicates the current mode.

FIFO status interrupts are read-only; the FIFO interrupt flag is automatically cleared if software performs a read or write operation on the FIFO during the processing of these interrupts.

Before enabling the interrupt bit, the application must clear the USBFS_GINTSTS register to zero during initialization to avoid generating any interrupts before initialization.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
WKU INT	VBU SVI NT	DIS CIN T	CID SCH G	Res erv ed	PTX FE	HCI NT	HPR TIN T	Res erv ed	DAT AFS USP	IPX FR/ INC OMP ISO UT	IIS OIX FR	OEP INT	IEP INT	Res erv ed	Rese rved
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EOPF	ISOO DRP	ENUM DNE	USBR ST	USBS USP	ESU SP	Res erv ed	Res erv ed	GON AKE FF	GIN AKE FF	NPT XFE	RXF NE	SOF	Res erv ed	MMI S	CMO D

Bit Flag		Bit Name Function Read/Write														
		Resume/remote wakeup detected interrupt														
b31	WKUINT	Resume/Remote										In device mode, this interrupt is triggered when a resume signal is detected on the USB bus.				
		Wakeup interrupt detected										when a connector ID line status change interrupt				
b30	VBUSVINTVBUS															
Valid Interrupts																
b29	DISCINT	Disconnect interrupt detected														

Write 1 to this bit to clear it via software.

Note: Access is available in both device mode and host mode.

VBUSy valid interrupt

In device mode, this interrupt is triggered when the USBFS_VBUS pin is detected to go from low to high. The bit is cleared by writing 1 to it through software.

R/W

Note: Accessible only in device mode.

R/W R/W

Disconnect detected interrupt is
triggered when a device is detected to be disconnected.

Write 1 to this bit to clear it via software.

Note: Accessible only in host mode.

Connector ID line status change (Connector ID status change)

The module places this position 1 when the status of the connector ID line changes.

			Write 1 to this bit to clear it via software. Note: Access is available in both device mode and host mode.	
b27	Reserved	-	The reset value must be maintained.	R
b26	PTXFFE		<p>Periodic TxFIFO empty interrupt When the periodic transmit FIFO is half-empty or full-empty and there is a periodic request queue that can be written to the</p> <p>This interrupt is triggered when the periodic TxFIFO is one entry short of space. Whether this FIFO is half-empty or fully empty is determined by the disruptions Periodic TxFIFO null level bit in USBFS_GAHBCFG register (PTXFELVL bit in USBFS_GAHBCFG) is determined.</p> <p>Note: Accessible only in host mode.</p>	R
b25	HCINT	Host Channel Interrupt	<p>Host channels interrupt The module indicates the presence of a pending interrupt on a channel in the module (in host mode) when it sets this position 1 (below) The application program must read the host USBFS_HAINT register to determine the pass-through where the interrupt occurred channel, and then read the corresponding USBFS_HCINTx register to determine the exact number of the interrupt that triggered the the exact cause of the problem. The application program must first clear the corresponding status bit in the USBFS_HCINTx register zero, and only after that can the bit be cleared to zero.</p> <p>Note: Accessible only in host mode.</p>	R
b24	HPRTINT	Host Port Interrupt	<p>Host port interrupt The module indicates a change in the status of the USBFS controller port in host mode when it sets this position 1. The application must read the USBFS_HPRT register to determine the exact event that triggered this interrupt. The application program must first clear the corresponding status bit of the USBFS_HPRT register to zero before it can set this Bit Clear.</p> <p>Note: Accessible only in host mode.</p>	R
b23	Reserved	-	The reset value must be maintained.	R
b22	DATAFSUSP	Data acquisition hangs	<p>Data fetch suspended This interrupt is only valid in DMA mode. This interrupt indicates that the module is inactive due to TxFIFO space or request queue Space is not available and stops fetching data for the IN endpoint. The application uses this interrupt for endpoint mismatch counting. in the method. For example, after detecting an endpoint mismatch, the application performs the following:</p> <ul style="list-style-type: none"> - Set the global non-periodic IN NAK handshake signal to 1. - Prohibition of IN endpoints - Empty FIFO - Determine the token sequence based on the IN token sequence learning queue - Re-enable endpoints - If the global nonperiodic IN NAK is cleared but the module has not yet acquired data for the IN endpoint, and also has received the IN token, then clear the global non-periodic IN NAK handshake signal: the module will generate the "IN token received when FIFO is empty" interrupt. The USBFS then sends the NAK response to the main 	R/W

			machine. To avoid this, the application can check the USBFS_GINTSTS in the DATAFSUSP interrupt, which ensures that the global NAK handshake signal is sent after the FIFO is full. Zeroing. Alternatively, the application can mask the "when" signal when zeroing out the global IN NAK handshake signal. An IN token interrupt is received when the FIFO is empty. Write 1 to this bit to clear it via software. Note: Accessible only in device mode.	
b21	IPXFR/ INCOMPISO OUT	Outstanding cyclic transmissions/ sync pass enter (a password)	<p>IPXFR: Incomplete periodic transfer In host mode, if there are unfinished periodic transactions that are still in the pending state, and these transactions scheduled to complete during the current frame, then the module will place this interrupt at position 1.</p> <p>INCOMPISO: Incomplete OUT synchronization transfer (Incomplete) isochronous OUT transfer</p> <p>Note: Accessible only in host mode.</p>	R/W

			In device mode, the module sets this interrupt to indicate that there is at least one synchronized OUT endpoint in the current frame The transmission on the EOPF has not been completed. This interrupt follows the end-of-period interrupt (EOPF) bit in this register a Same trigger. Write 1 to this bit to clear it via software. Note: Accessible only in device mode.	
b20	IISOIXFR	Incomplete IN synchronized transmission	Incomplete isochronous IN transfer When the module sets this interrupt to 1, it indicates that transmission on at least one of the synchronized IN endpoints in the current frame is not complete The interrupt is triggered with the End of Periodic Frame (EOPF) bit in R/W this register. This interrupt is triggered with the end-of-period interrupt (EOPF) bit in this register. Write 1 to this bit to clear it via software. Note: Accessible only in device mode.	
b19	OEPINT	OUT endpoint interrupt	OUT endpoint interrupt The module indicates the presence of a pending interrupt on one of the OUT endpoints in the module when it sets this position 1 (in the device module). (under the style) The application program must read the host USBFS_DAINT register to determine which interrupt occurred in the R The exact number of the OUT endpoint and then read the corresponding USBFS_DOEPINTx register to determine the The exact cause of the interrupt is triggered. The application program must first set the corresponding USBFS_DOEPINTx registers to of the corresponding status bit can be cleared to zero only after that bit is cleared. Note: Accessible only in device mode.	R
b18	IEPINT	IN Endpoint Interrupt	IN endpoint interrupt The module indicates the presence of a pending interrupt on one of the IN endpoints in the module (in device mode) when it sets this position 1 (below) The application program must read the host USBFS_DAINT register to determine the interrupt that occurred IN The exact number of the endpoint, and then read the corresponding USBFS_DIEPINTx register to determine the number that triggered the The exact cause of the interruption. The application must first set the corresponding The corresponding status bit of the USBFS_DIEPINTx register is cleared to zero before the bit can be cleared to zero. Note: Accessible only in device mode.	R
b17~b16	Reserved	-	The reset value must be maintained.	R
b15	EOPF	End-of-periodic-frame interrupt	End of periodic frame interrupt Indicates that the current frame has reached the periodic frame interval field in the USBFS_DCFG register (The period specified by (PFIVL bit in USBFS_DCFG). Write 1 to this bit to clear it via software. Note: Accessible only in device mode.	R/W
			Isochronous OUT packet dropped (Isochronous OUT packet dropped)	

			interrupt)	
b14	ISOODRP	Discard synchronized OUT packet interrupt	If the module is unable to write synchronization OUT packets to the RxFIFO due to insufficient space in the RxFIFO to accommodate the maximum number of packets for the synchronization OUT endpoints, the module sets this position 1. Write 1 to this bit to clear it via software. Note: Accessible only in device mode.	R/W
b13	ENUMDNE	Enumeration completion interrupt	Enumeration done interrupt The module indicates that the speed enumeration is complete when it places this position 1. The application must read the USBFS_DSTS register to get the enumeration speed.	R/W
b12	USBRST	USB reset interrupt	USB reset interrupt The module indicates that a reset signal is detected on the USB when this bit is 1. Write 1 to this bit to clear it via software. Note: Accessible only in device mode.	R/W
b11	USBSUSP	USB hang interrupt	USB suspend interrupt The module indicates that a hang state is detected on the USB when this position is 1. When the idle state on the USB bus The module enters the hang state when the state is held for 3ms. The bit is cleared by writing 1 to it through software.	R/W

			Note: Accessible only in device mode.
b10	ESUSP interrupt	Early hang	Early suspend interrupt (Early suspend interrupt) The module indicates that the USB has been detected to be idle for 3ms when it sets this position to 1. NOTE: Accessible in device mode only. R/W
b9~b8	Reserved must be held.	-Reset value must be maintained. R	The reset value
b7	GONAKEFF	Global OUT NAK active interrupt	Global OUT NAK effective interrupt (Global OUT NAK effective interrupt) Indicates the "Set global OUT NAK to 1" bit in the USBFS_DCTL register set by the application program. (SGONAK bit in USBFS_DCTL) has taken effect in the module. This is accomplished by writing the Rthe "clear global OUT NAK to zero" bit in the USBFS_DCTL register (USBFS_DCTL). in the CGONAK bit), this bit can be cleared to zero. NOTE: Accessible
b6	GINAKEFF	Global non-periodic IN NAK active interrupts	in device mode only. Global IN nonperiodic N A K effective interrupt (Global IN nonperiodic NAK effective interrupt) Indicates that the "Set Global Non-Cyclic IN NAK to 1" bit in the USBFS_DCTL register (the SGINAK bit in USBFS_DCTL), set by the application program, has taken effect in the module. In other words, the The module has sampled the global IN NAK bit set by the application program and the result has taken effect. The result is validated by clearing the RUSBFS_DCTL register "Zero the global non-periodic IN NAK" bit. (CGINAK bit in UBSFS_DCTL), this bit can be cleared. This interrupt does not necessarily indicate that a NAK handshake signal has been sent on the USB. the STALL bit has higher priority than the NAK bit.
b5	NPTXFE	Non-periodic TxFIFO air breaks	Note: Accessible only in device mode. Non-periodic TxFIFO empty interrupt When the non-periodic TxFIFO is half-empty or full-empty and the non-periodic transmit request queue holds at least This interrupt is triggered when space is available to write an entry. Whether this FIFO is in a half-empty state or a fully Rempty state is determined by the non-periodic TxFIFO empty level bit in the USBFS_GAHBCFG register (TXFELVL bit in USBFS_GAHBCFG) is determined. Note: Accessible only in host mode.
b4	RXFNE	An RxFIFO non-air break	RxFIFO non-empty interrupt (Rx FIFO non-empty interrupt) indicates that there is at least one packet in the Rx FIFO waiting to be read. R Note: Access is available in both host mode and device mode.

		In
b3	SOF	<p>Start of Frame Interrupt</p> <p>In host mode, the module indicates that a SOF (FS) or Keep-Alive (LS) signal has been sent on the USB when it sets this position</p> <p>1. The application program must set this position 1 to clear the interrupt.</p>
b2	Reserved	<p>-</p> <p>The reset value must be held.</p>
b1	MMIS	<p>Mode mismatch interrupt (Mode mismatch interrupt)</p> <p>i n t e r r u p t) The module places this location 1 when the application tries to access the following registers:</p> <ul style="list-style-type: none"> - Module running in device mode accesses host mode registers <p>Mode Mismatch Interrupt</p> <p>Module running in host mode accesses the device mode register</p> <p>R/W</p> <p>Access ends with an OKAY response on the AHB, but the access is internally ignored by the module and does not affect module operation.</p> <p>Write 1 to this bit to clear it via software.</p>
		<p>Note: Access is available in both host mode and device mode.</p>

Current mode of operation indicates the
current mode.

b0	CMOD	Current	0: Device mode	R
operating mode			1: Host Mode	

Note: Access is available in both host mode and device mode.

31.7.2.6 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 program. If an interrupt bit is masked, no interrupt associated with that bit will be generated.

However, the Module Interrupt (USBFS_GINTSTS) register bit corresponding to this interrupt will still be set to 1.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
WKU IM	VBU SVI M	DIS CIM	CID SCH GM	Res erved	PTX FEM	HCI M	HPR TIM	Res erved	DAT AFS USP M	IPX FRM / INC OMP ISO OUT M	IIS OIX FRM	OEP IM	IEP IM	Res erved	Res erved
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EOP FM	ISO ODR PM	ENUM DNEM	USBR STM	USBS USPM	ESU SPM	Res erved	Res erved	GON AKE FFM	GIN AKE FFM	NPT XFE M	RXF NEM	SOF M	Res erved	MMI SM	Res erved

Bit Flag	Bit Name	Function	Read/Write
b31	WKUIM	Recovery/remote wakeup interrupt mask detected	Resume/remote wakeup interrupt blocking detected (Resume/remote wakeup detected interrupt mak) 0: Mask interrupt 1: Enable interrupt Note: Access is available in both host mode and device mode.
b30	VBUSVIM	VBUS valid interrupt mask	VBUS valid interrupt mask 0: Mask interrupt 1: Enable interrupt Note: Accessible only in device mode.
b29	DISCIM	Disconnect interrupt mask detected	Disconnect detected interrupt interrupt mask 0: Mask interrupt 1: Enable interrupt Note: Accessible only in host mode.
b28	CIDSCHGM	Interrupt connector ID line status change interrupt blocking	Connector ID line status change interrupt mask (Connector ID status change interrupt mask) 0: Mask interrupt 1: Enable interrupt Note: Access is available in both device mode and host mode.
b27	Reserved	-	The reset value must be maintained. R
b26	PTXFEM	Cyclic TxFIFO Air Break	Periodic TxFIFO empty R/W

		interrupt mask	interrupt mask) 0: Mask interrupt 1: Enable interrupt Note: Accessible only in host mode.
b25	HCIM	Host Channel Interrupt Masking interrupt	Host channels interrupt mask 0: Mask R/W 1: Enable interrupt Note: Accessible only in host mode.
b24	HPRTIM	Host Port Interrupt Masking	Host port interrupt mask 0: Mask interrupts R/W 1: Enable interrupt Note: Accessible only in host mode.
b23	Reserved	-	The reset value must be held. R
b22	DATAFSUSP interrupt mask M	Data fetch pending	Data fetch suspended interrupt mask 0: Mask interrupt 1: Enable interrupt R/W Note: Accessible only in device mode.
b21	IPXFRM/ INCOMPISO OUTM	Uncompleted cyclic transmission interrupt blocking/ Uncompleted OUT synchronous transmission interrupt blocking	IPXFR: Incomplete periodic transfer interrupt mask 0: Mask interrupt 1: Enable interrupt Note: Accessible only in host mode. INCOMPISOOUT: Incomplete isochronous OUT transfer interrupt mask 0: Mask interrupt 1: Enable interrupt Note: Accessible only in device mode. Incomplete isochronous IN transfer interrupt mask (Incomplete isochronous IN transfer interrupt mask) R/W
b20	IISOIXFRM	Incomplete IN Sync Transfer Interrupt Screen b20 IISOIXFRM	0: Mask interrupt 1: Enable interrupt R/W Note: Accessible only in device mode.
b19	OEPIM	OUT endpoint interrupt mask	OUT endpoint interrupt mask (OUT endpoint interrupt mask) 0: Mask interrupt 1: Enable interrupt R/W Note: Accessible only in device mode.
b18	IEPIM	IN Endpoint Interrupt Mask	IN endpoint interrupt mask 0: Mask interrupt 1: Enable interrupt R/W Note: Accessible only in device mode.
b17~b16	Reserved	-	The reset value must be maintained. R
b15	EOPFM	Cyclic End-of-Frame Interrupt Mask	End of periodic frame interrupt mask 0: Mask interrupt R/W 1: Enable interrupt Note: Accessible only in device mode.

		Mask	dropped interrupt mask) 0: Mask interrupt 1: Enable interrupt Note: Accessible only in device mode.	
b13	ENUMDNEM	Enumeration completes interrupt masking interrupts	Enumeration done interrupt mask (枚举完成中断屏蔽) R/W 0: Mask 1: Enable interrupt Note: Accessible only in device mode.	0: Mask
b12	USBRSTM	USB reset interrupt masking	USB reset interrupt mask 0: Mask interrupt 1: Enable interrupt Note: Accessible only in device mode.	R/W
b11	USBSUSPM	USB hang interrupt masking	USB suspend interrupt mask 0: Mask interrupt 1: Enable interrupt Note: Accessible only in device mode.	R/W
b10	ESUSPM	Early hang interrupt masking interrupt	Early suspend interrupt mask (Early suspend interrupt mask) R/W 0: Mask 1: Enable interrupt Note: Accessible only in device mode.	0: Mask
b9~b8	Reserved	-	The reset value must be held.	R
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: Accessible only in device mode.	R/W
b6	GINAKEFFM	Global non-periodic IN NAK valid interrupt masking	Global IN nonperiodic N A K effective interrupt mask (Global IN nonperiodic NAK effective interrupt mask) 0: Masked interrupt 1: Enable interrupt Note: Accessible only in device mode.	R/W
b5	NPTXFEM	Non-periodic TxFIFO air break shielding	Non-periodic TxFIFO empty interrupt mask 0: Mask interrupt 1: Enable interrupt Note: Accessible only in host mode.	R/W
b4	RXFNEM	RxFIFO Non-Air Interrupt Mask Interrupt	RxFIFO non-empty interrupt mask R/W 0: Mask 1: Enable interrupt Note: Access is available in both host mode and device mode.	0: Mask
b3	SOFM	Frame start interrupt masking interrupt	Start of frame interrupt mask R/W 0: Mask 1: Enable interrupt Note: Access is available in both host mode and device mode.	0: Mask
b2	Reserved	-	The reset value must be held.	R
b1	MMISM	Mode mismatch interrupt mask	Mode mismatch interrupt mask R/W 0: Mask interrupt	

1: Enable interrupt

Note: Access is available in both host mode and device mode.

b0	Reserved	-Reset value must be held.	The reset value <u>must be held. R</u>
----	----------	-----------------------------------	---

31.7.2.7 USBFS Receive state debug read/USBFS Status read and stack out register (USBFS_GRXSTSR/USBFS_GRXSTSP)

USBFS Receive status debug read/USBFS status read and pop registers read

offset address: 0x01C

Offset address for

outgoing 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 Stack Out registers will additionally pop the data entry at the top of the RxFIFO. The receive status contents are interpreted differently in host mode and device mode.

When the receive FIFO is empty, the module ignores read or stack-out operations to this register and returns the value 0x0000 0000. When the receive FIFO non-empty bit in the module's interrupt register (the RXFNE bit in USBFS_GINTSTS) is set, the application program must pop only the receive status FIFO.

Host Mode:

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved										PKTSTS [3:0]				DPI D[1]]	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DPID [O]	BCNT [11:0]										CHNUM[3:0]				

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b21	Reserved	-	The reset value must be maintained.	R
b20~b17	PKTSTS	packet state	Packet status indicates the status of the received packet 0010: IN packet received 0011: IN transmission completed (interrupt triggered) 0101: Data synchronization error (triggered interrupt) 0111: Pause channel (trigger interrupt) Other values: reserved	R
			Data PID Indicates the data PID of the received packet	

b16~b15	DPID	Data PID	00: DATA0 10: DATA1 01: DATA2 11: MDATA	R
b14~b4	BCNT	Byte count	Byte count Indicates the number of bytes of the received IN packet.	R
b3~b0	CHNUM	Channel number	Channel number indicates the channel number to which the currently received packet R belongs.	

Device Mode:

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	
Reserved										PKTSTS [3:0]				DPID [1]	
b16 b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	
DPI D[0]]	BCNT [11:0] b0										EPNUM[3:0]				

classifier for honorific people	marking	celebrity	functionality	fill out or in (informa tion on a form)
b31~b21	Reserved	-	The reset value must be maintained.	R
b20~b17	PKTSTS Status	Packet	Packet status status) (Packet Indicates the status of the received packet 0001: Global OUT NAK (trigger interrupt) 0010: OUT packet received 0011: OUT transmission completed (interrupt triggered) 0100: SETUP transaction completed (triggered interrupt) 0110: SETUP packet received Other values: reserved Data PID (Data PID) Indicates the data PID of the received OUT packet. 00: DATA0 10: DATA1 01: DATA2 11: MDATA	R
b16~b15	DPID	Data PID	00: DATA0 10: DATA1 01: DATA2 11: MDATA	R
b14~b4	BCNT	Byte count	Byte count Indicates the number of bytes in the received packet.	R
b3~b0	EPNUM	Endpoint number	Endpoint number indicates the endpoint number to which the currently received packet R belongs.	

31.7.2.8 USBFS Receive FIFO Size Register (USBFS_GRXFSIZ)

USBFS Receive FIFO size register

offset address: 0x024

Reset value: 0x0000 0140

This application can program the size of the 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]							
<hr/>															
classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)											
b31~b11	Reserved	-	The reset value must be maintained.	R											
b10~b0	RXFD	RxFIFO	RXF D: RxFIFO depth (RxFIFO depth) in 32-bit words. Minimum value is 16 Maximum value is 256 The power-on reset value is the maximum Rx data FIFO depth.	R/W											
Depth															

31.7.2.9 USBFS host non-cyclic send FIFO size register (USBFS_HNPTXFSIZ)/endpoint 0 send FIFO size (USBFS_DIEPTXF0)

USBFS Host non-periodic transmit FIFO size register/Device endpoint0 transmit FIFO size register

Offset address: 0x028

Reset value: 0x02000140

This application can program the size of the 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/Write
b31~b16	NPTXFD/ TX0FD	<p>Host Mode: NPTXFD</p> <p>Non-periodic TxFIFO depth (Non-periodic TxFIFO depth) is measured in 32-bit words.</p> <p>Minimum value is 16</p> <p>Maximum value is 256</p>	R/W
b15~b0	NPTXFSA/ TX0FSA	<p>Device Mode: TX0FD</p> <p>Endpoint 0 TxFIFO depth (Endpoint 0 TxFIFO depth) is measured in 32-bit words.</p> <p>Minimum value is 16</p> <p>Maximum value is 256</p>	R/W
		<p>Host Mode: NPTXFSA</p> <p>Non-periodic transmit RAM start address (Non-periodic transmit RAM start address)</p> <p>This field contains the memory start address of the non-periodic transmit FIFO RAM.</p> <p>Device mode: TX0FSA</p> <p>Endpoint 0 transmit RAM start address (Endpoint 0 transmit RAM start address)</p> <p>This field contains the memory start address of the FIFO RAM to which Endpoint 0 is sent.</p>	

31.7.2.10 USBFS Non-Cyclic 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: 0x00080100

This read-only register contains free space information for the non-periodic TxFIFO and the non-periodic send request queue. 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
Rese rv ed	NPTXQTOP [6:0]										NPTQXSAR [7:0]				
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
NPTXFSAV [15:0]															

classifier	marking	celebrity	functionality	fill out for or in (informa tion on a form)
people	Reserved	-	The reset value must be maintained.	R

The top of the non-periodic send **of the non-periodic request queue (Top) transmit request queue)**



that the MAC is currently processing.	Non-periodic send request queue space available	
Bits 30:27: Channel/endpoi nt number (Channel/endpo int number) bits	(Non-periodic transmit request queue space available) Indicates the amount of free space available in the non-periodic send request queue. In host mode, this queue holds IN and OUT requests.	
26:25: – 00: IN/OUT Token – 01: Send packet with zero length – 11: Channel stop command	00: Off-cycle send request queue is full R 01: 1 position available 10: 2 positions available bxn: n positions available (where n ranges from 0 to 8) Other values: reserved	
Bit 24: Terminate (last <u>entry for selected</u>		
c h a n n e l / e n d p	Non-periodic TxFIFO space available	
NPTxPSAV	top of queue	Indicates the amount of free space available in the non-periodic TxFIFO. In 32-bit word units. 00: Non-periodic TxFIFO is full 01: 1 word available 10: 2 words available

0xn: n words available (where n ranges
from 0 to 256) Other values: reserved

31.7.2.11 USBFS Module ID Register (USBFS_CID)

USBFS core ID register

offset address: 0x03C

Reset value: 0x12345678

This register is the 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]															
<hr/>															
Bit Flag	Bit Name	Function	Read/Write												
b31~b0	PRODUCT_I D	Product ID field Product ID field ID field programmable through the application.	Product ID field R/W												

31.7.2.12 USBFS Host Periodic Transmit FIFO Size Register (USBFS_HPTXFSIZ)

USBFS Host periodic transmit FIFO size register

offset address: 0x100

Reset value: 0x01400280

This application programs the size of the RAM that must be allocated to the cycle TxFIF.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved								PTXFD[10:0]							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								PTXSA [11:0]							

classifier marking
for
honorific
people

b31~b27 Reserved

b26~b16 PTXFD

b15~b12 Reserved

b11~b0	PTXSA	Host Cyclic TxFIFO Start Address	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 non-cyclic TxFIFO depth.	R/W
--------	-------	--	---	-----

31.7.2.13 USBFS Device IN Endpoint Transmit FIFO Size Register (USBFS_DIEPTXF_x) (x = 1..5)

USBFS device IN endpoint transmit FIFO size register

offset address: 0x104+(x-1)*0x4

Reset value: 0x01000240 + (x-1)*0x100

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	
Reserved											INEPTXFD[9:0]
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4
Reserved											INEPTXSA [11:0]

classifier for honorific people	marking
b31~b26	Reserved

b25~b16	INEPTXFD
---------	----------

b15~b12	Reserved
b11~b0 TxFIFOx	INEPTXSA IN Endpoint RAM starting address

IN endpoint TxFIFOx RAM start address
(IN endpoint FIFOx transmit RAM start address)
This field contains the memory start address of the IN endpoint transmit FIFOx.
The address must be aligned with a 32-bit memory location.

R/W

31.7.3 USBFS Host Mode Register

The host mode register affects module operation in host mode. The host mode register must not be accessed in device mode because the resulting result is not clear.

Bit values in register descriptions are expressed in binary unless otherwise noted.

31.7.3.1 USBFS Host Configuration Register (USBFS_HCFG)

USBFS Host configuration register

offset address: 0x400

Reset value: 0x00000000

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														FSL SS	FSLSPCS[1 :0]

classifier for marking honorific people	celebrity	functionality	fill out or in (informa tion on a form)
b31~b3 Reserved	-	The reset value must be maintained.	R
b2 and LS	FSLSS	supports only FS for the 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 selection	FSLSPCS	FS/LS PHY clock select (FS/LS PHY clock select) When the module is in FS host mode 01: PHY clock running 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: Reservations Note: When the device is connected to a host computer, FSLSPCS must be set according to the speed of the connected device (a software reset must be performed after changing this bit)	R/W

31.7.3.2 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]															

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b26	Reserved	-	The reset value must be maintained	R
b15~b0	FRIVL	frame interval	<p>Frame interval The value programmed by the application in this field is used to specify two consecutive SOF (FS) or Keep-Alive order The time interval between cards (LS). This field contains the number of PHY clocks that make up the desired frame interval. It is only possible to set the Port enable bit of the host port control and status register (PENA bit of USBFS_HPRT) set to 1</p> <p>The application program can only write a value to this register after the value has been programmed. If the value is not programmed, the module will write the value based on the value written in the FS/LS PHY clock selection field of the host configuration register (in USBFS_HCFG) (FSLSPCS) to calculate the PHY clock specified in the FSLSPCS. Do not change the value of this field after the initial configuration. Setting value = frame interval (ms) × (PHY clock frequency) -1 Note: The FRIVL bit can be modified whenever the application needs to change the frame interval.</p>	R/W

31.7.3.3 USBFS Host Frame Number/Frame Remaining Time 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 time remaining in the current frame (in units of PHY clocks)

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

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b16	FTREM	frame remaining time	<p>Frame time remaining Indicates the time remaining in the current frame (in number of PHY clocks) For every 1 PHY clock that passes, this word Paragraph decreases by 1.</p> <p>When the value reaches zero, this field is reloaded with the value in the Frame Interval Register, and the value is set by the module on the USB Send a new SOF.</p>	R
b15~b0	FRNUM	frame number	<p>Frame number The value of this field is incremented by 1 when 1 new SOF is sent over USB and cleared when 0x3FFF is reached.</p> <p>Zero.</p>	R

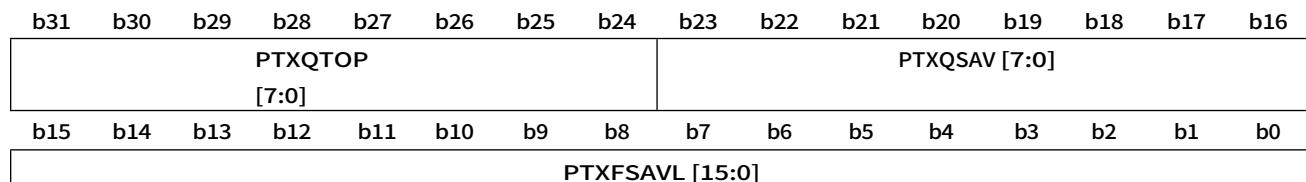
31.7.3.4 USBFS Host periodic transmit FIFO/queue status register

(USBFS_HPTXSTS) USBFS Host periodic transmit FIFO/queue

status register offset address: 0x410

Reset value: 0x00080100

This read-only register contains information about the free space in the periodic TxFIFO and the periodic transmit request queue.



Bit Flag	Bit Name	Function	Read/Write
b31~b24	PTXQTOP Request Queue	Periodic Send top of column	Top of the periodic send request 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: Sent in an odd number of frames Bits 30:27: Channel number Bits 26:25: type (Type) - 00: Input/Output - 01: Zero-length packets - 11: Prohibit Channel Command Bit 24: Terminate (last entry for the selected channel) Space available on the periodic send request queue (Periodic transmit request queue space available)
b23~b16	PTXQSAV	Periodic Send Request Team Column space available	Indicates the number of free slots in the periodic transmit request queue that are available for writing. The queue contains both IN and OUT requests. 00: Cyclic send request queue is full 01: 1 position available 10: 2 positions available bxn: n positions available (where n ranges from 0 to 8) Other values: reserved Cyclic send data FIFO free space (Periodic transmit data FIFO space available)
b15~b0	PTXFSAVL	Periodic send data FIFO free space	Indicates the number of free locations in the periodic TxFIFO that are available for writing. 32-bit word units 0000: Cyclic TxFIFO is full 0001: 1 word available 0010: 2 words available

31.7.3.5 USBFS Host all channels interrupt register

(USBFS_HINT) USBFS Host all channels interrupt

register Offset Address: 0x414

Reset value: 0x0000 0000

When an event occurs on a channel, the host plenary channel interrupt register uses the host channel interrupt bits in the module interrupt registers (HCINT bit in USBFS_GINTSTS) interrupts the application. Each channel corresponds to one interrupt bit, with a maximum of 12 bits.

When the application clears an interrupt through the corresponding host channel x interrupt register, the bits in this register are also cleared.

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				HINT[11:0]											

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b12	Reserved	-	The reset value must be maintained.	R
b11~b0	HINT	Channel interrupt	Channel interrupt Each channel corresponds to one bit: channel 0 corresponds to bit 0 and channel 11 corresponds to bit 11.	R/W

31.7.3.6 USBFS Host all channels interrupt mask register

(USBFS_HAINTP) USBFS Host all channels interrupt

mask register offset address: 0x418

Reset value: 0x0000 0000

The host-wide channel interrupt mask register is used in conjunction with the host-wide channel interrupt register, which in turn interrupts the application program when an event occurs on the channel.

Each channel corresponds to one interrupt mask bit, with a maximum of 12 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
Reserved		HAINTP [11:0]													

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b12	Reserved	-	Reads "0", writes "0".	R
b11~b0	HAINTP	Channel Interrupt Mask	Channel interrupt mask 0: Mask interrupt 1: Enable interrupt Each channel corresponds to one bit: channel 0 corresponds to bit 0 and channel 11 corresponds to bit 11.	R/W

31.7.3.7 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 supports only one port.

This register contains USB port related information such as USB reset, enable, suspend, resume, and connection status. The PENCHNG/PCDET bit in this register triggers an application interrupt via the host port interrupt bit in the module interrupt register (HPRTINT bit in USBFS_GINTST). When a port interrupt occurs, the application program must read this register and clear the bit that caused the interrupt. The application program 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	
Reserved														PSPD[1:0]	Rese rved
b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1
Reserved	b0	PWP R	PLSTS[1:0]	Rese rved	PRS T	PSU SP	PRE S	Reserved	PEN CHN G	PEN A	PCD ET	PCST S			

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b19	Reserved	-	The reset value must be maintained.	R
b18~b17	PSPD	port speed	Port speed Indicates the speed of the device connected to the port. 00/11: Reservations 01: Full speed 10: Low speed	R
b16~b13	Reserved	-	The reset value must be maintained.	R
b12	PPWR	Port Power	Port power The application program uses this field to control power to this port. Since this USBFS built-in PHY does not have a power supply capability, so this bit is set to 1 to enable the external USB power supply chip via USBFS_DRVVBUS Power supply. 0: Power down 1: energize	R/W
b11~b10	PLSTS	port line state	Indicates the current logic level of the USB cable Bit 11: Logic level of USBFS_DM Bit 10: Logic level of USBFS_DP	R

b9	Reserved	-	The reset value must be maintained.	R
b8	PRST	port reset	<p>Port reset</p> <p>The application initiates a reset sequence on this port when it sets this position 1. The application program must set the reset weak for the period timing and clears this bit to zero when the reset sequence is complete.</p> <p>0: Port is not in reset state 1: Port is in reset state</p> <p>The application must set this position 1 and hold it for a minimum of 10 ms to initiate a reset on the port.</p>	R/W
b7	PSUSP	port hangs	Port suspend	R/W

<p>The application places this position 1 to put this port in suspend mode. Only with this position 1 does the module stop Stop sending SOF. To stop the PHY clock, the application program must stop the port clock at position 1, which will cause the Hang up input pin of the capable PHY.</p> <p>The read value of this bit reflects the current pending state of the port. A remote wake-up signal is detected, or the application program The module can clear this bit after the sequence sets the port reset bit or port recovery bit 1 in this register; or should the Program the Resume/Remote Wake Detect Interrupt bit in the module interrupt register or the Disconnect Detect Interrupt bit (WKUINT or DISCINT in USBFS_GINTSTS, respectively) is set to 1, the module can also be Clear this bit to zero.</p> <p>0: Port is not in suspend mode 1: The port is in suspend mode</p>					
b6	PRES	port recovery	Port resume	The application program places this position 1 to drive the recovery signal on this port. The module will continue to drive the recovery signal straight to the application program to clear this bit to zero. If the Port Resume/Remote Wakeup Detect interrupt bit (USBFS_GINTSTS) in the Module Interrupt Register (WKUINT bit) indicates that if the module detects a USB Remote Wakeup sequence, it starts driving the recovery letter number without intervention by the application program; if the module detects a disconnected condition, it clears this bit Zero.	R/W
b5~b4	Reserved	-	The read value of this bit indicates whether the current module is The recovery signal is being driven. 0: Do not drive recovery signal 1: Drive recovery signal		R
b3	PENCHNG	Port Enable/Disable Change	Port enable/disable change	The module sets this bit 1 when the state of port enable bit 2 in this register changes. Write 1 to this bit to clear it via software.	R/W
b2	PENA	port enable (computing)	Port enable	After a port performs a reset sequence, it can only be enabled by the module and can be enabled by an overcurrent condition, a disconnect condition conditions or the application program can disable this bit by clearing it to zero. The application program cannot disable this bit by performing a write operation to the register to move the	R/W
b1	PCDET	Port connection detected	This bit 1. the port can only be disabled by clearing this bit to zero. Operation of this bit does not trigger any of the application program's What interruptions. 0: Disable port 1: Enable port	Port connect detected	When a device connection is detected,

the module sets this bit to 1 to trigger an interrupt from the application program using the host port interrupt bit in the module interrupt register (the HPRTINT bit in USBFS_GINTSTS).

R/W

R

The application

program must set

this position 1 to

clear the

interrupt. Port

connect status

(Port connect

status) 0: Port

is not

connected to

the device

1: Device connected to port

31.7.3.8 USBFS host channel x characterization 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
CHE NA	CHD IS	ODD FRM		DAD[6:0]						Reserved	EPTYP[1:0]	LSD EV	Rese rved	
b16 b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1
EPD IR		EPNUM[3:0]								MPSIZ [10:0]				
		b0												

classifier for marking Bit Name Function Read/Write

honorific

people

b31	CHENA	Channel Enable		Channel enable										
				This field is set to 1 by the application software and cleared by the USBFS host hardware.										
				0: Channel disabled										
				R/W										
				1: Enable Channel										
b30	CHDIS	Channel Disable		Channel disable										
				The application program sets this to position 1 to stop sending/receiving data through the channel, even if the transmission through the channel										
				is not yet complete, the stop operation is still in effect.										
				The application must wait for the disable channel										
				R/W										
b29	ODDFRM	Odd numbered frames		interrupt to confirm that the channel has been disabled.										
				Odd frame										
				This field is set or reset by the application program to indicate that the USBFS host must transmit an odd number of frames or an even number of frames, respectively.										
				Odd numbered frames										
				Number of frames. This field applies only to periodic (synchronized and interrupted) transactions.										
				R/W 0: Even numbered frames										
				1: Odd frames										
b28~b22	DAD	Device address		Device address										
				This field is used to specify a specific device to communicate with this host.										
b21~b20	Reserved	-		The reset value must be held.										
				R										
b19~b18	EPTYP	Endpoint Type		Endpoint type										
				(Endpoint type) indicates the type of transmission selected. 00: Control										
				01: Synchronization										
				10: Batch										
				11: Interruptions										

			Low-speed device	
			This field is set to 1 by the application program to indicate that the R/W channel is communicating with a low-speed device.	
b16	Reserved	-	The reset value must be held.	R
b15	EPDIR	Endpoint Direction	Endpoint direction Indicates whether the direction of the communication transaction is input or output. 0: Output R/W 1: Input	
b14-b11	EPNUM	Endpoint number	Endpoint number indicates the endpoint number of the USB device that is to communicate R/W with this host channel.	The

b10-b0	MPSIZ	Maximum packet size	Maximum packet size	R/W
			Indicates the maximum packet size for device endpoints communicating with this host channel.	

31.7.3.9 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 status of the channel in the event of USB and AHB related events.

When the host channel interrupt bit in the Module Interrupt Register

(The application program must read this register when (HCINT bit in USBFS_GINTSTS) is set to 1. Before performing a read operation on the register, the application program must 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 bit in this register before it can clear the corresponding bit in the USBFS_HAIN 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	TXE RR	Res erv ed	ACK	NAK	STA LL	Res erv ed	CHH	XFRC

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b11	Reserved	-	The reset value must be maintained.	R
b10	DTERR	Data toggle error	Data toggle error The R/W application needs to clear this bit by writing 1.	R/W
b9	FRMOR	Frame overrun error	Frame overrun error The application program needs to clear this bit by writing 1.	R/W
b8	BBERR	Crosstalk Error packet length exceeds the endpoint's R/W.	Babble error A typical cause of a crosstalk event is when the endpoint sends a packet, but the maximum packet length at the point. The application program needs to clear this bit by writing 1.	R/W
b7	TXERR	Communication service error	Transaction error Indicates that one of the following errors has occurred on the USB: CRC checksum failure overtime pay bit-padding error Wrong EOP	R/W

			The application program needs to clear this bit by writing 1.	
b6	Reserved	-	The reset value must be maintained.	R
b5	ACK	ACK response received/issued	ACK response received/transmitted interrupt The application program needs to clear this bit by writing 1.	R/W
b4	NAK	NAK response received	NAK response received interrupt The application program needs to clear this bit by writing 1.	R/W
b3	STALL	STALL response received	Received a STALL response (STALL response) received interrupt	R/W
b2	Reserved	-	Reads "0", writes "0".	R
b1	CHH	Channel Stop	Channel halted	R/W

<p>The transfer ended abnormally due to an arbitrary USB transaction error or in response to a prohibited request from the application program. The application program needs to clear this bit by writing a 1.</p>		
b0	XFRC Transmission complete	Transfer completed (Transfer completed) No error occurs and the transfer is completed normally. <u>The application program needs to clear this bit by writing 1.</u>

31.7.3.10 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 selectively mask 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	TXE RRM	Res erv ed	ACK M	NAK M	STA LLM	Res erv ed	CHH M	XFRC M

classifier for honorific people	marking	celebrity	functionality	fill out or in (informa tion on a form)
b31~b11	Reserved	-	The reset value must be maintained.	R
b10	DTERRM	Data switching error interrupt mask	Data synchronization error masking (Data 0: Mask interrupt) 1: Enable interrupt	toggle error mask) R/W
b9	FRMORM	Frame Overflow Error Interrupt Mask	Frame overrun mask 0: Mask interrupts 1: Enable interrupt	R/W
b8	BBERRM	Crosstalk error interrupt mask	Babble error mask 0: Mask interrupt 1: Enable interrupt Transaction error mask 0: Mask interrupts	R/W
b7	TXERRM	Communication transaction error interrupt mask	1: Enable interrupt	R/W
b6	Reserved	-Reset value must be maintained.	The reset value must be held.	R
b5	ACKM	ACK received/issued Response interrupt mask	ACK response receive/transmit 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 (NAK response received interrupt mask) 0: Mask interrupt 1: Enable interrupt	R/W
b3	STALLM	Received in STALL response ground fault (physics)	STALL response received interrupt mask) 0: Mask interrupt 1: Enable interrupt	R/W

			Channel halted mask	
b1	CHHM	Channel Stop	0: Mask interrupt	R/W
		Interrupt Mask		
			1: Enable interrupt	

Transfer completed mask 0: Mask interrupts			
b0	XFRCM	Transmission	1: Enable interrupt
completion interrupt mask			R/W

31.7.3.11 USBFS host channel x transfer 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 the 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]				
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
XFRSIZ [15:0]															

classify rfor honorifi c people	marking	celebrity	functionality	fill out or in (information on a form)
b31	Reserved	-	The reset value must be maintained.	R
b30~b29	DPID	Data PID	Data PID (Data PID) The application sets the initial synchronization PID for data communication in this field. the host retains the setting of this field for the duration of this transfer transaction. 00: DATA0 01: Reservations 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. R/W The host decrements the count value once for each packet successfully sent or received. When this value reaches 0, the application to indicate that the operation completed properly.	
b18~b0	XFRSIZ	Transmit Size transmission.	Transfer size For OUT operations, this field is the number of data bytes sent by the host during the application for transmission. For IN transactions (periodic and non- periodic), the application programs this field to be an integer multiple of the maximum packet size.	R/W

31.7.3.12 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 when 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]															
classifier for marking celebrity functionality honorific people															
fill out or in (information on a form)															
b31~b0 DMAADDR DMA address DMA address This field stores the memory of the DMA transfer used by the host to get data from or send data to the device endpoint. The address of the register. This register is incremented at the end of each AHB transfer.															
R/W															

31.7.4 USBFS Device Mode Register

The device mode register affects module operation in device mode. The device mode register must not be accessed in host mode because the resulting result is not clear.

Bit values in register descriptions are expressed in binary unless otherwise noted.

31.7.4.1 USBFS Device Configuration Register (USBFS_DCFG)

USBFS Device configuration register

offset address: 0x800

Reset value: 0x0820 0000

This register configures the module to device mode after power-up, execution of 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	NZLSOHSK	DSPD[1:0]				
<hr/>															
classifier for honorific people	marking	celebrity		functionality											fill out or in (inform ation on a form)
b31~b13	Reserved	-		The reset value must be maintained.											R
b12~b11	PFIVL	periodic frame interval (physics)		The periodic frame interval (Periodic frame interval) Indicates the point in a frame at which the application must be notified using a periodic frame break. This feature can be used to ensure that Determines whether all synchronized communications for the frame are complete.											
b10~b4	DAD	device address		00: 80% frame interval 01: 85% frame spacing 10: 90% frame spacing 11: 95% frame spacing											R/W
b3	Reserved	-		The reset value must be maintained.											R
b2	NZLSOHSK	Non-zero length state OUT handshake signal		Non-zero length state OUT handshake signal (Non-zero-length status OUT handshake)											DSPD
<hr/>															

handshake signal to be sent.

1: When a non-zero-length status OUT transaction is received, the STALL handshake signal is replied to and the received OUT packet is not sent to the application program.
0: Sends the received OUT packet (zero-length or non-zero-length) to the application program and replies with a handshake signal based on the NAK and STALL bits of the endpoint in the device's Endpoint Control Register.

R/W

Device speed

Indicates the speed that the application requires the module to use for enumeration, or the maximum speed supported by the application. However, the actual bus speed can only be determined after the chirp sequence is completed, and also this speed is based on the speed of the USB host connected to the module.

R/W

00: Reserved

01: Reservations

10: Reservations

11: Full speed (USB 1.1 transceiver clocked at 48 MHz)

31.7.4.2 USBFS Device Control Register (USBFS_DCTL)

USBFS Device control register

offset address: 0x804

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				POP RGD NE	CGO NAK	SGO NAK	CGI NAK	SGI NAK	Reserved			GON STS	GIN STS	SDI S	RWU SIG

classifier for honorific	marking	celebrity	functionality	fill out or in
people				(informational form)
b11	POPRGDNE	Power-on programming complete	The application program uses this bit to indicate that the register has been programmed after waking up from power-down mode.	B/W on a form)
b31~b12	Reserved	-	The reset value must be maintained.	R
b10	CGONAK	Clear global OUT NAK	Clear global OUT NAK (Clear global OUT NAK) Power-on programming (programming done) Performing a write operation to this bit clears the global OUT NAK to zero. Is complete (Power-on)	W
b9	SGONAK	Set global OUT NAK	Set global OUT NAK (Set global OUT NAK) Performing a write operation to this bit sets the global OUT NAK to 1. The application uses this bit to send NAK handshake signals at all OUT endpoints. The application program only determines the global OUT NAK valid bit in the module interrupt registers (GONAKEFF bit in USBFS_GINTSTS) can only be set to position 1 when it has been cleared to zero.	W
b8	CGINAK	Clear global IN NAK	Clear global IN NAK A write operation to this bit clears the global IN NAK to zero.	W
b7	SGINAK	Setting the global IN NAK	Set global IN NAK Performing a write operation to this field sets the global non-periodic IN NAK to 1. The application program uses this bit to make the All non-periodic IN endpoints send NAK handshake signals.	W
b6~b4	Reserved	-	The application program only determines the global IN NAK valid bit in the module interrupt registers (GINAKEFF bit in USBFS_GINTSTS) is cleared to zero before this position 1.	R
b3	GONSTS	Global OUT NAK status	Global OUT NAK status 0: Handshake signals will be sent based on the FIFO status and NAK and STALL bit settings. 1: Data is not received regardless of whether there is still free space in the RxFIFO. With the exception of SETUP transactions, the Respond to the NAK handshake signal for all received packets. All OUT packets of the synchronization type will be Discard.	R
Global IN NAK status				

1: Enables all non-periodic IN endpoints to reply to the NAK handshake signal without regard to the number in the sending FIFO
According to availability.

Soft disconnect

The application program uses this bit to signal the USBFS module to perform a soft disconnect. When this bit is 1, the host will not see the device connected and the device will not receive a signal R/W on the USB. After the application sets this

y

The module will remain disconnected until the bit is cleared.

0: Normal operation. Clearing this bit after a soft disconnect causes the host to receive the event that the device is connected.

After reconnecting the device, the USB host restarts the device enumeration. 1: Causes the host to receive a device disconnect event.

At full speed, the minimum time for soft disconnection is specified as follows: Suspended state:
minimum time is 1ms+2.5us Idle state: 2.5us

b0 RW USIG Send remote wake-up signal

Non-idle or suspend state: 2.5us R/W

Sending Remote wakeup signaling

When the application program sets this position 1, the module initiates a remote wake-up signal to wake up the USB host. The application program must set this position 1 in order for the module to exit the suspend state. According to the USB 2.0 specification, the application must clear this position 1 within 1 ms to 15 ms after it is set.

31.7.4.3 USBFS Device Status Register (USBFS_DSTS)

USBFS Device status register

offset address: 0x808

Reset value: 0x0000 0002

This register indicates the status of the module in the event of a USB-related event. When an interrupt occurs, information about the endpoint at which the interrupt occurred must be read from the device-wide interrupt (USBFS_DAINT) register.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved										FNSOF [13:8]					
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
FNSOF[7:0]								Reserved				EER R	ENUMSPD[1: 0]	SUS PST S	

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b22	Reserved	-	The reset value must be maintained.	R
b21~b8	FNSOF	Receive SOF frame number	The frame number of the number of the received SOF (Frame received SOF (Frame	R
b7~b4	Reserved	-	The reset value must be maintained.	R
Erratic error				
b3	EERR	indeterminate error	The module places this position 1 to report any indeterminate errors. Due to an indeterminate error, the USBFS controller will go into a hang state and will USBFS_GINTSTS mail the The early pending bit of the memory (the ESUSP bit in USBFS_GINTSTS) generates an interrupt. If the If the early hang interrupt was triggered by an indeterminate error, the application can only perform a soft disconnect to resume communication.	R
Enumerated speed				
b2~b1	ENUMSPD	Enumeration speed	Indicates the speed that the USBFS controller has been enumerated into after detecting the speed through the chirp sequence. 01: Reservations 10: Reservations 11: Full speed (PHY clock running at 48 MHz) Other values: reserved	R
Suspend status				
 In device mode, this bit is set to 1 whenever a hung state is detected on the USB. when the USB bus The module enters the pending state when the idle state on the module is maintained for 3ms. The module retires when the following conditions occur				

b0	SUSPSTS	pending	out of the suspended state: - Activity on USB cable - The remote wake-up signal bit of the USBFS_DCTL register (in the USBFS_DCTL) is used by the application to signal the remote wake-up signal bit of the USBFS_DCTL register (in the RWUSIG bit) performs a write operation.	R
----	---------	---------	--	---

31.7.4.4 USBFS Device IN endpoint common interrupt mask register

(USBFS_DIEPMSK) USBFS Device IN endpoint common interrupt

mask register offset address: 0x810

Reset value: 0x0000 0000

This register is used in conjunction with the individual USBFS_DIEPINTx registers for all endpoints to generate interrupts at each IN endpoint. The IN endpoint interrupts in the USBFS_DIEPINTx register can be masked by performing a write operation 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															

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b7	Reserved	-	The reset value must be maintained.	R
b6 INEPNEM	IN endpoint NAK active interrupt mask	IN endpoint NAK effective mask 0: Mask interrupt 1: Enable interrupt	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 blocking received on EP mismatch (IN token received with EP mismatch mask) 0: mask interrupt 1: Enable interrupt	IN token interrupt mask received when TxFIFO is empty (IN token received when TxFIFO empty mask) 0: mask interrupt 1: Enable interrupt	R/W
b4	ITTXFEMSK	Receive IN Token Interrupt Mask	Timeout interrupt mask (non-synchronized endpoints)	
b3	TOM	Timeout interrupt mask (unsynchronized endpoint)	(Timeout condition m a sk (Non-isochronous endpoints)) 0: Mask interrupt 1: Enable interrupt	R/W
b2	Reserved	-	The reset value must be maintained.	R
b1	EPDM	Endpoint Disable Interrupt Masking	Endpoint disable interrupt disabled interrupt mask) masking (Endpoint 0: Mask interrupt) 1: Enable interrupt	R/W
b0	XFRM	Transmission completion interrupt mask	Transfer completion completed interrupt mask) interrupt mask (Transfer 0: Mask interrupt) 1: Enable interrupt	R/W

31.7.4.5 USBFS Device OUT endpoint common interrupt mask register

(USBFS_DOEPMSK) USBFS Device OUT endpoint common interrupt

mask register offset address: 0x814

Reset value: 0x0000 0000

This register is used in conjunction with the individual USBFS_DOEPINTx registers for all endpoints to generate interrupts at each OUT endpoint. OUT endpoint interrupts in the USBFS_DOEPINTx register can be masked by performing a write operation 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															

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b5	Reserved	-	The reset value must be maintained.	R
b4	OTEPDM	OUT Token Interrupt Mask received when endpoint is disabled	OUT token received when endpoint is disabled Interrupt Masking (OUT token received when endpoint disabled mask) applies only to control OUT endpoints. 0: Mask interrupt 1: Enable interrupt	R/W
b3	STUPM	SETUP phase completes interrupt masking	SETUP phase done mask For control endpoints only. 0: Mask interrupt 1: Enable interrupt	R/W
b2	Reserved	-	The reset value must be maintained.	R
b1	EPDM	Endpoint Disable Interrupt Masking	Endpoint Disable Interrupt disabled interrupt mask) Masking (Endpoint 0: Mask Interrupt 1: Enable interrupt	R/W
b0	XFRCM	Transmission completion interrupt mask	Transfer completion completed interrupt mask) interrupt mask (Transfer 0: Mask interrupt) 1: Enable interrupt	R/W

31.7.4.6 USBFS Device All Endpoint Interrupt Register (USBFS_DAIINT)

USBFS Device OUT endpoint common interrupt mask register

offset address: 0x818

Reset value: 0x0000 0000

When a valid event occurs on an endpoint, the USBFS_DAIINT 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 bit in USBFS_GINTSTS, respectively). Each endpoint has one interrupt bit, with a maximum of 16 interrupt bits for both the OUT and IN endpoints. Bidirectional endpoints will use the corresponding IN and OUT interrupt bits. When the application sets bit 1 and clears bit 0 in the corresponding device endpoint x interrupt register (**USBFS_DIEPINTx/USBFS_DOEPINTx**), the corresponding bit in this register will also be set to 1 and cleared.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved								OEPINT[5:0]							
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								IEPINT[5:0]							

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b22	Reserved	-	The reset value must be maintained.	R
b21~b16	OEPINT	OUT endpoint interrupt bit	OUT endpoint endpoint interrupt bits) interrupt bit (OUT	
b15~b6	Reserved	-	Each OUT endpoint corresponds to one bit: OUT endpoint 0 corresponds to bit 16, while OUT endpoint 5 corresponds to bit 21.	R/W
b5~b0	IEPINT	IN endpoint interrupt bit	IN endpoint interrupt bits (IN endpoint interrupt bits) One bit for each IN endpoint: IN endpoint 0 corresponds to bit 0, while IN endpoint 5 corresponds to bit 5.	R/W

31.7.4.7 USBFS Device all endpoints interrupt mask register

(USBFS_DAINTMSK) USBFS Device all endpoints interrupt

mask register Offset Address: 0x81C

Reset value: 0x0000 0000

The USBFS_DAINTMSK register is used in conjunction with the device endpoint interrupt register to interrupt the application program when an event occurs on the device endpoint. However, the USBFS_DAINT register bit corresponding to that interrupt is still set to 1.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserved										OEPINTM[5:0]					
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved										IEPINTM[5:0]					

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b22	Reserved	-	The reset value must be maintained.	R
b21~b16	OEPINTM R/W	OUT endpoint interrupt mask Bit	OUT endpoint interrupt mask bits) Each OUT endpoint corresponds to one bit: OUT endpoint interrupt mask Bit 0: Mask interrupt 1: Enable interrupt	OUT endpoint 0 corresponds to bit 16, while OUT endpoint 5 corresponds to bit 21.
b15~b6	Reserved	-	The reset value must be maintained.	R
b5~b0	IEPINTM	IN endpoint interrupt mask bit	IN endpoint interrupt mask bits) Each IN endpoint corresponds to one bit: IN endpoint 0 corresponds to bit 0, while IN endpoint 5 corresponds to bit 5. 0: Mask interrupt 1: Enable interrupt	R/W

31.7.4.8 USBFS Device IN Endpoint FIFO Air Break 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 air breaks.

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										INEPTXFEM[5:0]											
<hr/>																					
Bit Flag	Bit Name	Function	Read/Write																		
b31~b6	Reserved	-Reset value must be maintained. value must be held.	R																		
b5~b0	INEPTXFEM	IN EP Tx FIFO airborne disconnect shield	IN EP Tx FIFO Air Break 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 for an IN endpoint: IN endpoint 0 corresponds to bit 0, while IN endpoint 5 corresponds to bit 5 0: Masked interrupt 1: Enable interrupt																		

31.7.4.9 USBFS Device Control 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
EPE NA	EPD IS	Reserved		SNA K	CNA K	TXFNUM[3:0]				STA LL	Res er ved	EPTYP[1:0]		NAK STS	Res er 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/Write
b31	EPENA	Endpoint Enable endpoint:	<p>Endpoint enable The application program places this position 1 to initiate data transmission on endpoint 0.</p> <p>The module clears this bit before triggering any of the following interrupts on this R/W</p> <ul style="list-style-type: none"> – endpoint prohibition – Transmission complete
b30	EPDIS		<p>Endpoint disable Even before the transmission on this endpoint is complete, the application can place this position 1 to stop the endpoint on the</p> <p>The endpoint disables Data Transmission. The application program must wait until an endpoint forbidden interrupt occurs before treating the endpoint as a forbidden end R/W point. The module clears this bit before the endpoint disable interrupt position 1. Only if the endpoint is endpoint enabled at that endpoint position 1 before the application can place that position 1.</p>
b29~b28	Reserved	-	The reset value must be held. R
b27	SNAK	Setting the NAK bit	<p>Set NAK bit (Set NAK) A write operation to this bit sets the endpoint's NAK position 1.</p> <p>With this bit, the application program can control the sending of the NAK R/W handshake signal on the endpoint. The module can also be connected to the endpoint at the</p> <p>Receive the SETUP packet to this position 1 of the endpoint.</p>
b26	CNAK	Clear NAK Bit	<p>Clear NAK Bit (Clear NAK) A write operation to this bit clears the NAK bit of the endpoint. R/W</p>
b25~b22	TXFNUM	TxFIFO number	<p>TxFIFO number (TxFIFO number) This value is set to the FIFO number assigned to IN endpoint 0. Only R/W TX-FIFO0 can be used.</p>
b21	STALL	STALL Handshake.	<p>STALL handshake The application program can only set this bit to 1. The module clears this bit 0. When the module receives a non-NAK handshake based on the FIFO status. 1: The module will ignore both NAK and handshake bits. If the NAK and handshake bits are all set R/W to 1, then the STALL bit is preferred</p> <p>First.</p>
b20	Reserved	-	The reset value must be held. R
b19~b18	EPTYP	Endpoint type	<p>Endpoint type Hardware set to '00' for control type endpoints. R</p>
b17	NAKSTS	NAK Status	<p>NAK status Indicates the following result: /1407 R</p>

When this bit is 1 (either by the application program or by the module), the module stops sending data even if there is still data available in the TxFIFO. Regardless of how this bit is set, the module always responds to SETUP packets with an ACK handshake.

b16	Reserved	-	The reset value must be held.	R
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 held.	R
b1~b0 Packet Size	MPSIZ	Maximum	Maximum packet size The application must program this field to be the maximum packet size for the current logical endpoint. 00: 64 bytes 01: 32 bytes 10: 16 bytes 11: 8 bytes	R/W

31.7.4.10 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 0080

The application uses this register to control the behavior of each logical endpoint (except endpoint 0).

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EPE NA	EPD IS	SOD DFR M	SD0 PID / SEV NFR M	SNAK	CNA K	TXFNUM[3:0]				STA LL	Res erv ed	EPTYP[1:0]]		NAK STS	EON UM /DP ID
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/Write
b31	EPENA	endpoint enable	<p>Endpoint enable</p> <p>The application program places this position 1 to initiate data sending on the endpoint.</p>
b30	EPDIS	endpoint prohibition	<p>The module clears this bit before triggering any of the following interrupts on this endpoint:</p> <ul style="list-style-type: none"> – SETUP phase completed – endpoint prohibition – Transmission complete <p>Endpoint disable</p> <p>Even before the transmission on this endpoint is complete, the application can place this position 1 to stop the endpoint on the</p> <p>Data Sending. The application must wait until an endpoint disable interrupt occurs before it can treat the endpoint as a disable end Point. The module clears this bit before endpoint disable interrupt position 1. The endpoint is only enabled if the endpoint of the position 1 before the application can place that position 1.</p>
b29	SODDFRM	Setting Odd Frames	<p>Set odd frame</p> <p>For synchronized IN and OUT endpoints only.</p> <p>A write operation to this field sets the Even/Odd Frames (EONUM) field to odd frames.</p>
b28	SDOPID/ SEVNFRM	Setting DATA0 SEVNFRM	<p>Set DATA0 PID (Set DATA0 PID)</p> <p>Applies to interrupt/bulk IN endpoints only.</p> <p>A write operation to this field sets the endpoint data PID (DPID) field in this register to</p> <p>PID/ DATA0.</p> <p>S</p> <p>b27 SNAK Sets the NAK bit</p> <p>E</p> <p>V</p>

even frame

applies only to

synchronized

IN endpoints.

R/W

A write operation to this field sets the

Even/Odd Frames (EONUM) field to even frames.

Set NAK bit (Set NAK)

A write operation to this bit sets the NAK position of the endpoint to 1.

This bit allows the application program to control the sending of the NAK handshake signal on the endpoint. When a transmission completion interrupt occurs or after a SETUP is received on the endpoint, the module can also set this bit of the OUT endpoint 1

b26

CNAK

Clear NAK bit

Clear NAK bit (Clear NAK)

R/W

			A write operation to this bit clears the NAK bit of the endpoint. TxFIFO number	
b25~b22	TXFNUM	TxFIFO Number	These bits are used to specify the FIFO number associated with this endpoint. You must set the TxFIFO number for each valid IN endpoint.	R/W
			The FIFOs are individually numbered. This field is valid only for IN endpoints.	
b21	STALL	STALL Handshake	STALL handshake The application program sets this bit to 1 so that the device responds STALL to all tokens from the USB host. if the R/W NAK bit, global IN NAK, or global OUT NAK is set to 1 at the same time as this bit, then the STALL bit is preferred. First. Only the application program can zero this bit, not the module.	
b20	Reserved	-	The reset value must be held.	R
b19~b18	EPTYP	Endpoint Type	Endpoint type The following are the transmission types supported by this logical endpoint. 00: Control 01: Synchronization 10: Batch 11: Interruptions	R
b17	NAKSTS	NAK Status	NAK status (NAK status) indicates the following results: 0: The module replies with a non-NAK handshake based on the FIFO status. 1: The module replies to the NAK handshake on this endpoint. When the application or module places this position 1: For non-synchronized IN endpoints: even if there is available data in the TxFIFO, the module stops passing IN endpoint to send any data. For synchronized IN endpoints: the module sends packets of zero length even if there is available data in the TxFIFO.	R
b16	EONUM/DPID	Even/Odd Frames/Endpoint Data PIDs	Regardless of how this bit is set, the module always responds to SETUP packets with an ACK handshake. Even/odd frames are only available for synchronizing IN endpoints. Indicates the number of the frame in which the module sends/receives synchronized data for this endpoint. The application program must program the even/odd frame number through the SEVNFRM and SODDFRM fields in this register for this endpoint to send/receive synchronized data. 0: even frames 1: Odd frames	R/W
			DPID: Endpoint data PID For interrupt/bulk IN endpoints only. Contains the PID of the packet to be received or sent on this endpoint. After the 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.	
HC32F460_F45x_A460 Series Reference Manual_Rev1.6			1267	
			/1407	

The application must program this field to be the maximum packet size for the current logical endpoint. This value is in bytes.

31.7.4.11 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 0000

This register indicates the status of the endpoint in the event of USB and AHB related events.

When the IN endpoint interrupt bit in the Module Interrupt Register

(The application program must read this register when the IEPINT bit in USBFS_GINTSTS is set to one. Before the application can read this register, it must read the Device All Endpoint Interrupt (USBFS_DAINT) register to obtain the exact endpoint number for the Device Endpoint x Interrupt register. The application program must clear the corresponding bit in this register before it can clear the corresponding bit in the USBFS_DAINT 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															

classifier for marking honorific people	celebrity	functionality	fill out or in (information on a form)
b31~b8	Reserved	-	The reset value must be maintained.
b7 is empty	TXFE	Transmit FIFO	Send FIFO is empty (Transmit FIFO empty) This interrupt is set when the TxFIFO at this endpoint is half-empty or full-empty, and the state of whether the TxFIFO is half-empty or full-empty is determined by the TxFIFO blank bit in the USBFS_GAHBCFG register. (TXFELVL bit in USBFS_GAHBCFG) is determined.
b6 endpoint NAK is valid	INEPNEIN	INEPNE: IN endpoint NAK effective (IN endpoint NAK effective) This bit can be cleared when the application program clears the IN endpoint NAK to zero by writing data to the CNAK bit in USBFS_DIEPCTLx. This interrupt indicates that the module has sampled the NAK set to 1 (by the application program or by the module) and the result has taken effect. This interrupt indicates that the IN endpoint NAK bit set to 1 by the application program has taken effect in the module. This interrupt does not guarantee that a NAK handshake signal was sent on the USB. The STALL bit has a higher priority than the NAK bit. Software write 1 also clears this bit.	R R/W

value must be held.

R

IN token received when TxFIFO is empty

(IN token received when TxFIFO is empty)

b4	TTXFE	Connects when TxFIFO is empty non-periodic IN endpoints only.	For R/W
		IN Token Received When the TxFIFO (periodic/non-periodic) corresponding to this endpoint is empty, the IN token is received, thereby generating an interrupt. Clear by software write 1.	
b3	TOC	Timeout Applies to control IN endpoints only. Indicates that this endpoint timed out in response to the most recently received IN token. Clear by software write 1.	R/W
b2	Reserved	- The reset value must be held.	R

			Endpoint disabled interrupt This bit indicates that the endpoint has been disabled by the application program. Clear by software write 1.	
b1	EPDISD	Endpoint Disable Interrupt	Transfer completed interrupt This field indicates that the transmission set up on this endpoint has been completed on the USB and AHB. Clear by software write 1.	R/W

31.7.4.12 USBFS Device IN Endpoint 0 Transfer Size Register (USBFS_DIEPTSIZ0)**USBFS Device IN endpoint 0 transfer size register**

offset address: 0x910

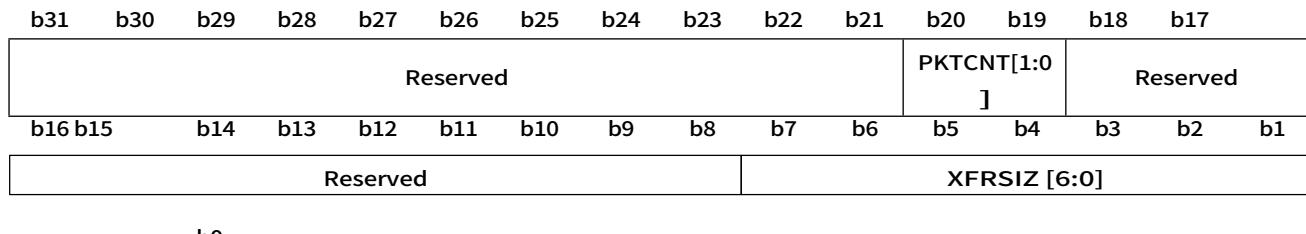
Reset value: 0x0000 0000

The application program must modify this register before enabling endpoint 0. Endpoint enable bits in the Control Register via Device Control Endpoint 0

(The module modifies this register after enabling endpoint 0 (EPENA in USBFS_DIEPCTL0).

The application program can read this register only after the module clears the endpoint enable bit to zero.

Non-zero endpoints use registers with endpoints 1 to 5.



classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b21	Reserved	-	The reset value must be maintained.	R
b20~b19	PKTCNT	packet count	Packet count Indicates the number of packets contained in a single data transmission for endpoint 0. This field is decremented each time a packet (maximum or short packet) is read from the TxFIFO.	R/W
b18~b7	Reserved	-	The reset value must be maintained.	R
b6~b0	XFRSIZ	Transmission size	Transfer size Indicates the amount of data, in bytes, that a single data transfer from endpoint 0 contains. The amount of data in bytes is only available if the application transfers The module will only interrupt the application after this data is finished. The transfer size can be set to the maximum number of endpoints 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

31.7.4.13 USBFS device IN endpoint x transfer size register (USBFS_DIEPTSIZx) (x=1..5)

USBFS Device IN endpoint x transfer size register

offset address: 0x910 + (endpoint number × 0x20)

Reset value: 0x0000 0000

The application program must modify this register before enabling this endpoint. Through the endpoint enable bit in the USBFS_DIEPCTLx register

(This register is modified by the module after the endpoint is enabled (EPENA bit in USBFS_DIEPCTLx). The application program can read this register only after the module clears the endpoint enable bit.

b31	Reserved	b29	b28	b27	b26	b25	PKTCNT [b23]	b22	b21	b20	b19	b18	XFRSIZ [17:16]		
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0

XFRSIZ [15:0]

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b29	Reserved	-	The reset value must be maintained.	R
b28~b19	PKTCNT	packet count	Packet count Indicates the number of packets contained in a single data transmission on this endpoint. This field is decremented each time a packet (maximum size or short packet) is read from the TxFIFO.	R/W
b18~b0	XFRSIZ	Transmissio n size	Transfer size This field contains the amount of data, in bytes, contained in one data transfer for the current endpoint. This field is only available if the application The module will not interrupt the application until the program has transferred this data. The transfer size can be set to the endpoint of the maximum packet size to interrupt 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

31.7.4.14 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 when the device is in 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/Write	
b31~b0	DMAADDR	DMA Address	DMA address	
			This bit contains the starting address of the external memory area when using DMA for on-endpoint data storage. NOTE: For the control endpoint, the memory area pointed to by this field is also used to store control OUT packets as well as SETUP transaction packets. When more than three SETUP packets are received consecutively, the SETUP packets in memory will be overwritten. This register is incremented each time an AHB transfer is made. The application program must set a double-word aligned address.	R/W

31.7.4.15 USBFS Device IN Endpoint Transmit FIFO Status Register (USBFS_DTXFSTS_x) ($x=0..5$)

USBFS Device IN endpoint transmit FIFO status register

offset address: 0x918 + (endpoint number × 0x20)

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
INEPTFSAV [15:0]															

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b16	Reserved	-	The reset value must be maintained.	R
b15~b0	INEPTFSAV	IN endpoint TxFIFO can be space utilization	IN endpoint TxFIFO space available Indicates the amount of free space available in the endpoint TxFIFO. in 32-bit words: 0x0: endpoint TxFIFO is full 0x1: 1 word available 0x2: 2 words available 0xn: n words available	R

31.7.4.16 USBFS Device Control 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
EPE NA	EPD IS	Reserved		SNA K	CNA K	Reserved			STA LL	SNP M	EPTYP[1:0]		NAK STS	Rese rv ed	
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/Write
b31	EPENA	Endpoint enable	
	Endpoint Enable	The application program places this position 1 to initiate data reception on endpoint 0.	
		The module clears this bit before triggering any of the following interrupts on this endpoint:	R/W
		- SETUP phase complete	
		- endpoint prohibition	
		- Transmission complete	
b30	EPDIS	Endpoint disable	R
		The application program cannot disable control of OUT endpoint 0.	
b29~b28	Reserved	-	R
		The reset value must be held.	
b27	SNAK	Setting the NAK bit	
		Set NAK bit (Set NAK)	
		A write operation to this bit sets the endpoint's NAK position 1.	
		With this bit, the application program can control the sending of the NAK handshake signal on the endpoint. The module can also be connected to the endpoint at the	R/W
		Receive the SETUP packet to this location 1 of this endpoint.	
b26	CNAK	Clear NAK Bit	R/W
		A write operation to this bit clears the NAK bit of the endpoint.	
b25~b22	Reserved	-	R
		The reset value must be held.	
b21	STALL	STALL handshake	
		When this endpoint receives a SETUP token, the application program can only set this to position 1, and the module will clear it.	
		If the NAK bit, global OUT NAK, and this bit are both set to 1, the STALL bit takes precedence. Whether this bit is set as	R/W
b20	SNPM	The module always responds to SETUP packets with an ACK handshake, regardless of the setting.	
		Snoop mode	
		Listen Mode	
		This bit is used to configure the endpoint to listen mode. In listen mode, the module does not transmit OUT packets to the endpoint before passing them to the	R/W
		Check that it is correct before transferring it to the application memory.	
b19~b18	EPTYP	Endpoint type	
		hardware set to '00' indicates the control type of endpoint.	R/W

b17	NAKSTS	NAK	status) indicates the following results: 0: The module replies with a non-NAK handshake based on the FIFO status. R 1: The module replies to the NAK handshake on this endpoint. When an application or module sets this location 1, the application or module will not be able to continue to hold the received
-----	--------	-----	---

packet, the module also stops receiving data. Regardless of how this bit is set, the module always responds to SETUP packets with an ACK handshake.

b16	Reserved	-	The reset value must be held.	R
b15	USBAEP	USB active endpoint	USB active endpoint This bit is always set to 1, indicating that control endpoint 0 is always active R in all configurations and interfaces.	
b14~b2	Reserved	-	The reset value must be held.	R
b1~b0	MPSIZ Packet Size	Maximum	Maximum packet size The maximum packet size for control OUT endpoint 0 is the same as the value programmed in control IN endpoint 0. 00: 64 bytes 01: 32 bytes 10:16 bytes 11:8 bytes	R/W

31.7.4.17 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

The application uses this register to control the behavior of each logical endpoint (except endpoint 0).

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
EPE NA	EPD IS	SOD DFR M/ SD1 PID	SD0 PID / SEV NFR M	SNA K	CNA K	Reserved				STA LL	SNP M	EPTYP[1:0]		NAK STS	EON UM /DP ID
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/Write
				Endpoint enable
b31	EPENA	Endpoint Enable	Software set, USBFS clear 0: Endpoint de-energization 1: Endpoint Enable	R/W
b30	EPDIS	Endpoint	Even before the transfer on this endpoint is complete, the application program can place this position 1 to stop the number on the endpoint Disable Data is sent/received. The application must wait until an endpoint forbidden interrupt occurs before it can treat the endpoint as a forbidden endpoint Point. The module clears this bit before the endpoint disable interrupt position 1. Only if the endpoint enable bit for this endpoint The application can only set this position 1 after setting it to 1.	R/W
b29	SD1PID/ S0DDFRM	Setting DATA1 PID/Set Odd one of a pair (scrolls)	Set DATA1 PID (Set DATA1 PID) For interrupt/bulk OUT endpoints only. A write operation to this field sends the endpoint data in this register to the interrupt/bulk OUT endpoint only. The PID(DPID) field is set to DATA1. R	R
b28	SD0PID/ SEVNFRM	Setting the DATA0 PID/ SEVNFRM	Set DATA0 PID (Set DATA0 PID) For interrupt/bulk OUT endpoints only. A write operation to this field sets the endpoint data PID (DPID) field in this register to DATA0. SEVNFRM: Set even frame applies only to synchronized OUT endpoints. R	R

b27	SNAK	Sets the NAK bit	A write operation to this bit sets the NAK position 1 of the endpoint. This bit allows the application program to control the sending of the NAK handshake signal on the endpoint. A transmission completion interrupt occurs	R/W
-----	------	------------------	---	-----

<p>When or after a SETUP is received on the endpoint, the module can also place this position 1 of the OUT endpoint.</p>			
b26	CNAK	Clear NAK Bit	Clear NAK Bit (Clear NAK) A write operation to this bit clears the NAK bit of the endpoint. R/W
b25~b22	Reserved	-	The reset value must be held. R
b21	STALL	STALL Handshake	STALL handshake When this endpoint receives a SETUP token, the application program can only set this to position 1, and the module will clear it. R/W If the NAK bit, global OUT NAK, and this bit are both set to 1, the STALL bit takes precedence. Only the application program
b20	SNPM		This bit can be cleared to zero, while the module cannot. Snoop mode Listen Mode This bit is used to configure the endpoint to listen mode. In listen mode, the module will no longer check the received data for the Correctness. R/W
b19~b18	EPTYP	Endpoint Type	Endpoint type The following are the transmission types supported by this logical endpoint. 00: Control 01: Synchronization 10: Batch 11: Interruptions R/W
b17	NAKSTS	NAK Status	The NAK status indicates the following results: 0: The module replies with a non-NAK handshake based on the FIFO status. 1: The module replies to the NAK handshake on this endpoint. R When an application or module places this position 1: The module stops receiving any data on the OUT endpoint even if space exists in the RxFIFO to accommodate incoming packets. Regardless of how this bit is set, the module always responds to SETUP packets with an ACK handshake.
b16	EONUM/	Even/odd frames/ Data. 0: even frames 1: Odd frames	Even/odd frame is only available for synchronized OUT endpoints. Indicates the number of the frame in which the module is sending/receiving synchronized data for this endpoint. The application program must program the even/odd frame number through the SEVNFRM and SODDFRM fields in this register in order for this endpoint to send/receive synchronized Data. 0: even frames 1: Odd frames ... R
DPID	Endpoint Data PID	<p>DPID: Endpoint data PID For interrupt/bulk OUT endpoints only. Contains the PID of the packet to be received or sent on this endpoint. After the 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.</p> <p>0: DATA0</p>	

1: DATA1

			USB active endpoint	
b15	USBAEP	USB active endpoint	Indicates whether this endpoint is active in the current configuration and interface. Upon detecting a USB reset, the module clears this bit for all endpoints (except endpoint 0). Receiving the SetConfiguration and SetInterface command, the application program must program the endpoint registers accordingly and set this Position 1.	R/W
b14~b11	Reserved	-	The reset value must be maintained.	R/W
b10~b0	MPSIZ	Maximum packet size	Maximum packet size	R/W

The application must program this field to be the maximum packet size for the current logical endpoint. This value is in bytes.

31.7.4.18 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 in the event of a USB and AHB related event. The application program must read this register when the OUT endpoint interrupt bit in the USBFS_GINTSTS register (the OEPINT bit in USBFS_GINTSTS) is set to one. Before the application can read this register, it must read the USBFS_DAINT register to obtain the exact endpoint number for the USBFS_DOEPINTx register. The application must clear the corresponding bit in this register before it can clear the corresponding bit in the USBFS_DAINT 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

classifier for honorific people	marking	celebrity	functionality	fill out or in (informa tion on a form)
b31~b7	Reserved	-	The reset value must be maintained.	R
b6	B2BSTUP	Continuous SETUP packets received	Back-to-back SETUP packets are received. For control OUT endpoints only. This bit indicates that more than three consecutive SETUP packets have been received at this endpoint. A software write of 1 also clears this bit.	R/W
b5	Reserved	-	Reset value must be maintained.	The reset
value must be held.	R			
b4	OTEPDIS	Received when the endpoint is disabled OUT Token	OUT token received when endpoint is disabled (OUT token received when endpoint disabled) Applies only to control OUT endpoints. Indicates that an OUT token is received when the endpoint is not yet enabled, resulting in a medium Disconnect. Clear by software write 1.	R/W
b3	STUP	SETUP phase completed	SETUP phase done For control OUT endpoints only. Indicates that the SETUP phase of the control endpoint is complete and that the current control transmission	R/W
b2	Reserved	-	No more consecutive SETUP packets are received in the middle. On this interrupt, the application program can perform a check on the received SETUP packets are decoded. Clear by software write 1.	R
b1	EPDISD	Endpoint Disable	Endpoint disabled interrupt This bit indicates that the endpoint has been disabled by the application	R/W

			program.	
			Clear by software write 1.	
b0	XFRC	Transmission completion interrupt	Transfer completed interrupt This field indicates that the transmission set up on this endpoint has been completed on the USB and AHB. Clear by software write 1.	R/W

31.7.4.19 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 program must modify this register before enabling endpoint 0. Endpoint enable bits in the Control Register via Device Control Endpoint 0

(The module modifies this register after enabling endpoint 0 (EPENA in USBFS_DIEPCTL0).

The application program can read this register only after the module clears the endpoint enable bit to zero.

Non-zero endpoints use registers with endpoints 1 to 5.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Reserve	STUPCNT[1:0]		Reserved								PKT CNT	Reserved			
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved								XFRSIZ [6:0]							

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31	Reserved	-	The reset value must be maintained.	R
b30~b29	STUPCNT	SETUP packet count	SETUP packet count (SETUP) This field specifies the number of SETUP packets that the endpoint can receive consecutively. 01: 1 packet 10: 2 packets 11: 3 packets	R/W
b28~b20	Reserved	-	The reset value must be maintained.	R
b19	PKTCNT	packet count	Packet count The number of packets that should be received in one transmission. This bit is set by software before the endpoint is enabled, and after transmission begins, whenever a packet is received, the number of fields in the The value is automatically reduced.	R/W
b18~b7	Reserved	-	The reset value must be maintained.	R
b6~b0	XFRSIZ	Transmission size	Transfer size Indicates the amount of data in bytes that a single data transfer from endpoint 0 contains. The module interrupts the application only when it has finished transferring this data. The transfer size can be set to the maximum packet size of the endpoint to interrupt at the end of each packet. The module decrements this field each time a packet is read from the	R/W

RxFIFO and written to external memory.

31.7.4.20 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 program must modify this register before enabling this endpoint. By means of the endpoint enable bit in the USBFS_DOEPCTLx register

(This register is modified by the module after the endpoint is enabled (EPENA bit in USBFS_DOEPCTLx). The application program can read this register only after the module clears the endpoint enable bit.

b31 Reserved	b29	b28	b27	b26	b25	PKTCNT [19:0]	b22	b21	b20	b19	b18	XFRSIZ [17:16]	b17	b16	
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
XFRSIZ [15:0]															

classifier for honorific people	marking	celebrity	functionality	fill out or in (informa tion on a form)
b31~b29	Reserved	-	The reset value must be maintained.	R
b28~b19	PKTCNT	packet count	Packet count Indicates the number of packets contained in a single data transmission on this endpoint. This field is decremented each time a packet (maximum size or short packet) is written to the RxFIFO.	R/W
b18~b0	XFRSIZ	Transmiss on size	Transfer size This field contains the amount of data, in bytes, contained in one data transfer for the current endpoint. This field is only available if the application The module will not interrupt the application until the program has transferred this data. The transfer size can be set to end point's maximum packet size to interrupt 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

31.7.4.21 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: 0xFFFF XXXX

This register is used to set the DMA address when the device is in 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]															

BitFlag	Bit Name	Function	Read/Write	
b31~b0	DMAADDR	DMA Address	DMA address	
			This bit contains the starting address of the external memory area when using DMA for on-endpoint data sending. NOTE: For the control endpoint, the memory area pointed to by this field is also used to store control OUT packets as well as SETUP transaction packets. When more than three SETUP packets are received consecutively, the SETUP packets in memory will be overwritten. This register is incremented each time an AHB transfer is made. The application program must set a double-word aligned address.	R/W

31.7.5 USBFS Clock Gating Control Registers

Power consumption is reduced by controlling the HCLK and PHY clocks through the Gated Clock Control Register. Bit values in register descriptions are expressed in binary unless otherwise noted.

31.7.5.1 USBFS Clock Gating Control Register (USBFS_GCCTL)

Offset address: 0xE00

Reset value: 0x0000 0000

This register is available in both host mode and device mode.

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															
GAT EHC LK STP PCL K															
classifier for marking celebrity functionality honorific people															
b31~b2	Reserved	–	The reset value must be maintained.												R
b1	GATEHCLK	Gated HCLK	Gate HCLK When USB communication hangs or a session is invalid, the application program sets this bit to 1 to stop clocking modules other than the AHB bus slave interface, master interface, and wake-up logic. The application clears this bit to zero when USB communication resumes or a new session is initiated.												R/W
b0	STPPCLK	stops the	Stop PHY clock The application program sets this bit 1 to stop the PHY clock when USB communication hangs, the session is invalid, or the device is disconnected. The application clears this bit to zero when USB communication resumes.												R/W
PHY clock															

32 Cryptographic Coprocessing Module (CPM)

32.1 summary

The Cryptographic Co-Processing Module (CPM) consists of three sub-modules: the AES encryption and decryption algorithm processor, the HASH secure hashing algorithm, and the TRNG true random number generator.

The AES encryption and decryption algorithm processor follows standard data encryption and decryption standards and can realize encryption and decryption operations with a 128-bit key length.

The HASH Secure Hash Algorithm is the SHA-2 version of SHA-256 (Secure Hash Algorithm) which is compliant with the national standard "FIPS PUB 180-3" issued by the National Institute of Standards and Technology, and can produce a 256-bit message digest output for messages up to 2^{64} bits in length.

The TRNG True Random Number Generator is a continuous analog noise-based random number generator that provides 64bit random numbers.

32.2 Encryption and Decryption Algorithm Processor (AES)

32.2.1 Introduction to Algorithms

The AES encryption algorithm is a key iterative grouping cipher that incorporates the repetitive action of wheel transformations on state. The wheel transformation of the encryption process consists of four operations: **SubBytes**, **ShiftRows**, **MixColumns**, and **AddRoundKey**, where **SubBytes** is the modulo inverse of each byte in GF(2^8) and an affine transformation; **ShiftRows** is a byte permutation that cyclically shifts the rows of the state by different offsets; **MixColumns** performs a linear transformation on the columns of the state; **AddRoundKey** performs a linear transformation on the columns of the state; and **AddRoundK** repeats the state. **ShiftRows** is a byte permutation that circularly shifts the rows of the state by different offsets; **MixColumns** performs a linear transformation on the columns of the state; and **AddRoundKey** performs a bit-wise dissimilarity operation between the bytes of the state and the round key. Figure 32-1 below illustrates the encryption process:

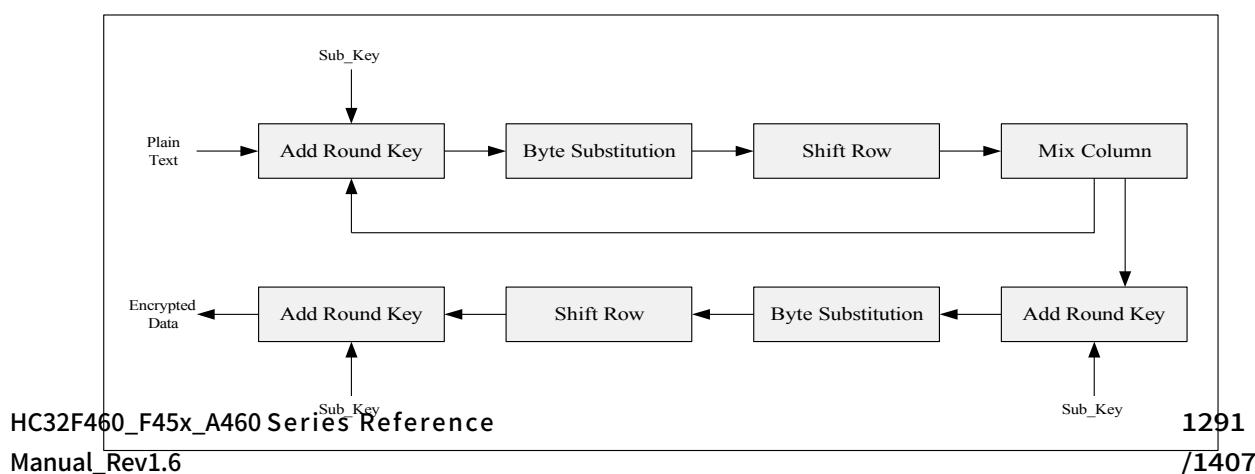


Figure 32-1 AES Encryption Flow Diagram

Sub_Key in the above figure refers to the subkey of each round, except for the initial transformation with the initial key, the subkey used in the later rounds of transformation needs to be expanded from the initial key, and the key expansion process and the encryption process are synchronized.

Since the plaintext is fixed at **128bits**, the number of rounds the encryption process runs depends on the length of the key. For example, if the key is **128bits**, the number of rounds is 10; if the key is **192bits**, the number of rounds is 12; and if the key is **256bits**, the number of rounds is 14.

Rounds. Except for the last round which lacks the MixColumn transform, all other rounds perform a full round transform operation. The AES on this product only supports encryption and decryption processes with a key length of 128bits.

The decryption process differs from the encryption process in that the expansion of all keys must be completed first, and the decryption process works backward from the expanded last round key, and then the four operations of the round transformation become the corresponding inverse operations: InvSubBytes, InvShiftRows, InvMixColumns, and AddRoundKey. InvSubBytes remains the modulo inverse operation, but the affine transformation is changed to inverse transformation; InvShiftRows and InvMixColumns become the corresponding inverse transformation; AddRoundKey remains unchanged. The modulo inverse operation in InvSubBytes remains, but the affine transformation is changed to inverse transformation; InvShiftRows and InvMixColumns become the corresponding inverse transformations; and AddRoundKey remains unchanged.

32.2.2 Encryption process

An example of the AES encryption operation is shown below:

- 1) Writes 128bits plaintext to the data register (AES_DR).
- 2) Write the encryption key to the key register (AES_KR).
- 3) Sets each bit in the control register (AES_CR), including:
 - a) Set CR.Mode to 0
 - b) Write 1 to AES_CR.START to start the module for arithmetic operations

Note: Steps a, b can be performed simultaneously

- 4) Determines whether module arithmetic is finished.

Constantly read AES_CR.START, if its value becomes 0, the operation is finished

- 5) Read the data register (AES_DR) to get the 128-bit ciphertext.
- 6) If you want to continue with a new operation, go back to step 1, otherwise end.

32.2.3 Decryption process

An example of an AES decryption operation is shown below:

- 1) Writes 128bits of ciphertext to the data register (AES_DR).
- 2) Write the decryption key to the key register (AES_KR).
- 3) Sets each bit in the control register (AES_CR), including:
 - a) Set CR.Mode to 1
 - b) Write 1 to AES_CR.START to start the module for arithmetic operations

Note: Steps a, b can be performed simultaneously

- 4) Determines whether module arithmetic is finished.

Constantly read AES_CR.START, if its value becomes 0, the operation is finished

- 5) Read the data register (AES_DR) to get the 128-bit plaintext.
- 6) If you want to continue with a new operation, go back to step 1, otherwise end.

32.2.4 encryption and decryption time

The time required by the AES module from the start of an operation (AES_CR.START writes to 1) to the end of that operation (AES_CR.START reverts to 0) is as follows:

encryption process	440 CPU clock cycles
decryption process	580 CPU clock cycles

32.2.5 Handling Precautions

- 1) After power-up, the module performs an asynchronous reset operation. The clock needs to be stable and valid until the reset is disengaged and continues to be stable during subsequent operation.
- 2) During the encryption and decryption process, the data registers change, so if the data for the next arithmetic operation is the result of this operation, there is no need to rewrite the data.
- 3) In the case of encrypting and decrypting a large amount of data with the same key, there is no need to write the key repeatedly.
- 4) To determine the end of module operation: keep reading AES_CR.START, if its value changes to 0, it means the operation is finished.

32.2.6 Register Description

base adr: 0x4000_8000

register name	notation	offset	bit width	reset value
AES Control Register	AES_CR	0x0000	32	0x0000_0000
AES data register 0	AES_DR0	0x0010	32	0x0000_0000
AES Data Register 1	AES_DR1	0x0014	32	0x0000_0000
AES Data Register 2	AES_DR2	0x0018	32	0x0000_0000
AES Data Register 3	AES_DR3	0x001C	32	0x0000_0000
AES key register 0	AES_KR0	0x0020	32	0x0000_0000
AES Key Register 1	AES_KR1	0x0024	32	0x0000_0000
AES Key Register 2	AES_KR2	0x0028	32	0x0000_0000
AES Key Register 3	AES_KR3	0x002C	32	0x0000_0000

Attention:

- Write operations to all registers are valid only when this module is in the idle state (AES_CR.START=0), otherwise write operations are ignored; read operations to registers other than the control register (AES_CR) are valid only when this module is in the idle state (AES_CR.START=0); can only read out the valid data, otherwise read out the unknown data; the read operation to the control register (AES_CR) can be carried out at any time, all can read out the valid data.

32.2.6.1 AES Control Register (AES_CR)

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																
classifier for marking celebrity functionality honorific people																
fill out or in (information on a form)																
b31~b2	Reserved			-		Reads "0" and writes "0".				R/W						
b1	MODE			Function		0: Cryptographic operations				R/W						
					Selection		1: Decryption operations									
b0	START			activate (a plan)		0: The module is finished or not activated.				R/W						
1: Launch this module for arithmetic																

32.2.6.2 AES Data Register (AES_DR)

The data register consists of four 32-bit registers with 128-bit data, which are used to store the plaintext to be encrypted or the ciphertext to be decrypted before the operation of the module, and to store the encrypted ciphertext or decrypted plaintext after the operation is completed.

cryptographic operation		decryption algorithm	
pre-operational	after calculation	pre-operational	after calculation
128-bit plaintext	128-bit cipher	128-bit cipher	128-bit plaintext

Four 32-bit registers are connected together to form a 128-bit data, and read and write operations need to be performed on each of the four registers. Number of bits : 128 bits

Offset address : 0x10 - data[31:0]
 0x14 - data[63:32]
 0x18 - data[95:64]
 0x1C - data[127:96]

Reset value: 0x0000_0000 (each 32-bit register)

Data example: 0x00112233445566778899AABCCDDEEFF

offset address	Register Name	Fill in the data
0x10	DR0	0x3322_1100
0x14	DR1	0x7766_5544
0x18	DR2	0xBBAA_9988
0x1C	DR3	0xFFEE_DDCC

32.2.6.3 AES Key Register (AES_KR)

The key register consists of four 32-bit registers to form a 128-bit key,
which is used to store the initial key entered. Number of bits : 128
bits

Offset address : 0x20 - key[31: 0]

0x24 - key[63: 32]

0x28 - key[95: 64]

0x2C - key[127: 96]

Reset value : 0x0000_0000 (each 32-bit register)

Data example: 0x000102030405060708090A0B0C0D0E0F

offset address	Register Name	Fill in the data
0x20	KR0	0x0302_0100
0x24	KR1	0x0706_0504
0x28	KR2	0x0B0A_0908
0x2C	KR3	0x0F0E_0D0C

32.3 Secure Hash Algorithm (HASH)

32.3.1 Introduction to Algorithms

The steps of the secure hashing algorithm are as follows:

The message is first padded so that its length is exactly 64 bits less than a multiple of 512. The padding is done by appending a 1 to the message.

Append the requested number of zeros and then append the 64-bit message length (before padding) to it so that the message length is exactly an integer multiple of 512 bits.

Next, a total of eight 32-bit variables, A, B, C, D, E, F, G and H, are initialized in hexadecimal. Then the main loop of the algorithm begins, processing 512-bit messages at a time, the number of loops being the number of 512-bit groups in the message.

The main loop performs a total of 64 operations, which are called compression functions. Each operation consists of shifts, circular shifts, logical operations, modulo 2³² additions, etc. The process of the operations is shown in Figure 32-2 below. The final output is a cascade of A, B, C, D, E, F, G, and H, where W_t is the temporary value used in step t obtained from the 512-bit message, K_t is the constant value used in step t, and t (0 ≤ t ≤ 63) is one step in the 64-step loop.

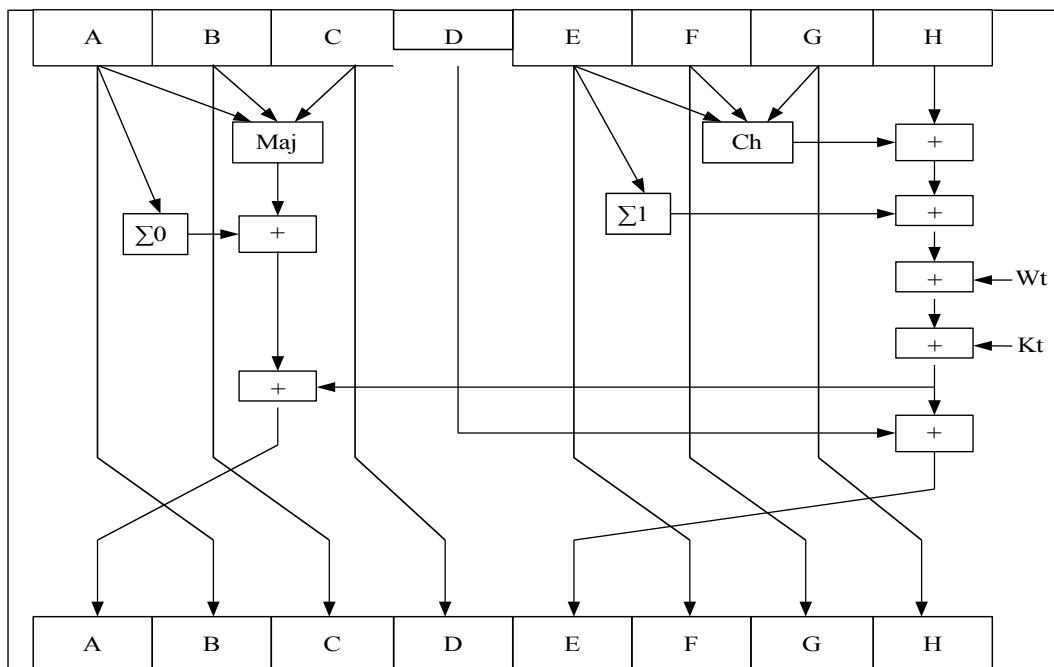


Figure 32-2 HASH Algorithm Flowchart

32.3.2 workflow

After power-up, the module performs an asynchronous reset operation. The clock needs to be stable and valid before the reset is disengaged and continue to be stable during subsequent operation. The operation flow of the HSAH module is as follows:

- 1) The software populates the raw data according to the algorithmic rules and groups the populated messages by 512 bits.
- 2) Writes the operated data into the data register (HASH_DR).
- 3) If this operation is the first set of data in the message grouping, write a 1 to the HASH_CR . FST_GRP bit is written to 1.
- 4) Write 1 to HASH_CR.START to start this module for arithmetic.

Note: 3 and 4 above can be done at the same time.

- 5) Determine whether this module is complete for this operation by the following methods:
Keep reading HASH_CR.START until the bit is 0, which means the operation is completed.
- 6) (If this operation is not the last set of data in the message grouping, return to 2)
- 7) If this operation is the last set of data in the message grouping, read the digest register (HASH_HR) to obtain the result of this operation. Return to step 1 if further operations are required.

32.3.3 message stuffing

The steps for processing a padded group for SHA-256 are as follows:

1. raw message grouping

Divide the original message into L groups of 512 bits. The total number of bits in the original message is l. If $l \% 512 < 448$, then the number of groups L is $l / 512$; if $l \% 512 \geq 448$, then the number of groups L is $l / 512 + 1$.

2. Add Length

① Add filler bits:

Add padding bits at the end of the $l / 512$ th group of the message packet: a 1 and a number of zeros, the number of zeros can be zero. If $l \% 512 < 448$, the padding makes the length of the data bits satisfy $448 \bmod 512$ (the last 64 bits are reserved for the representation of the original message length) if $l \% 512 \geq 448$. A 512-bit block of data in the $l / 512$ th group is filled with a 1 and a number of zeros, and the first 448 bits of the L th ($L = l / 512 + 1$) group are filled with zeros.

② Add the original message length:

A 64bit block representing the original message length as a 64bit unsigned

integer. The original message length is added to the last 64bit of the Lth packet.

An example of the process of populating a grouping is as follows:

1) Fill Example 1:

The original message is the string "ABCDE", the ASCII code is "01100001 01100010 01100011 01100100 01100101" expressed as a binary bit string, and the steps to add the length are as follows:

- A. Add "1". The populated message is "01100001 01100010 01100011 01100100

01100101 1".

- B. Add "0". Since the original message length is 40 bits, the number of zeros to add is $512 - 64 - 40$
 $-1 = 407$.

The padded message becomes (hex)

61626364 65800000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000

- C. Add the original message length. The raw message length of 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) Fill Example 2:

original start News Message for characters Symbol
string "ABCDBCDECDEFDEFGEFGFHFGHIGHHIJKIJKLJKLMKLMN
LMNOMNOPNOPQ": Each character can be converted to 8 bits by its ASCII code, so the length of the message is $l = 56 * 8 = 448$.

- A. Add "1" and "0". The padded message (in 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 raw message length of 448 is expressed in two 32bit words (hexadecimal): 00000000 000001C0.

The padded message (in hexadecimal) is the second message block:

00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 000001C0

32.3.4 Register Description

base adr: 0x4000_8400

register name	notation	offset	bit width	reset value
HASH Control Register	HASH_CR	0x0000	32	0x0000_0000
HASH Digest Register 7	HASH_HR7	0x0010	32	0x0000_0000
HASH Digest Register 6	HASH_HR6	0x0014	32	0x0000_0000
.....
HASH Digest Register 1	HASH_HR1	0x0028	32	0x0000_0000
HASH Digest Register 0	HASH_HR0	0x002C	32	0x0000_0000
HASH data register 15	HASH_DR15	0x0040	32	0x0000_0000
HASH Data Register 14	HASH_DR14	0x0044	32	0x0000_0000
.....
HASH data register 1	HASH_DR1	0x0078	32	0x0000_0000
HASH data register 0	HASH_DR0	0x007C	32	0x0000_0000

32.3.4.1 HASH Control Register (HASH_CR)

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															
classifier for honorific people	marking	celebrity													fill out or in (information on a form)
b31~b2	Reserved	–													R/W
b1	FST_GRP	The first group of the message subgroup													R/W
b0	START	activate (a plan)													R/W

Attention:

- The operation of the START bit is as follows: after the software writes 1 to this bit, the module will start running; after the end of this operation, the hardware will clear this bit automatically; if the software queries this bit to be 0, it means that this operation is completed.
- Write operations to this register can only be performed when the module is not in the computing state (i.e., when the START bit is 0), otherwise the hardware will automatically ignore the write operation. Otherwise, the hardware will automatically ignore the write operation. The read operation is not subject to this restriction.

32.3.4.2 HASH Digest Register (HASH_HR)

Number of bits : 256

Offset address : 0x10 - hash[255:224]

0x14 - hash[223:192]

0x18 - hash[191:160]

0x1C - hash[159:128]

0x20 - hash[127:96]

0x24 - hash[95:64]

0x28 - hash[63:32]

0x2C - hash[31:0]

Reset value: 0x0000_0000 (each 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]																				
<hr/>																				
<table><tr><td>classifier for honorific people</td><td>marking</td><td>celebrity</td><td>functionality</td><td>fill out or in (information on a form)</td></tr></table>																classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)																
<table><tr><td>b31~b0</td><td>HASH[31:0]</td><td>summary value</td><td>Get a message summary by reading this register after the module has finished arithmetic.</td><td>R/W</td></tr></table>																b31~b0	HASH[31:0]	summary value	Get a message summary by reading this register after the module has finished arithmetic.	R/W
b31~b0	HASH[31:0]	summary value	Get a message summary by reading this register after the module has finished arithmetic.	R/W																

Attention:

- This register consists of eight 32-bit registers. The eight 32-bit registers are accessed sequentially.
The corresponding 32-bit register holds the high word of the message digest.
- The hardware will automatically ignore write operations to this register.
- Reads to this register can only be performed when the module is not in the computing state (HASH_CR.START=0), otherwise reads to this register will get all zeros.

32.3.4.3 HASH Data Register (HASH_DR)

```

figure :: 512-
offset :: bit - data[511..480]
address 0x040
          0x044 - data[479..448]
          0x048 - data[447..416]
          0x04C - data[415..384]
          .....
          0x070 - data[127..96]
          0x074 - data[95..64]
          0x078 - data[63..32]
          0x07C - data[31..0]

```

Reset value : 0x0000_0000 (each 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]															
<hr/>															
classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)											
b31~b0	DATA[31:0]	data register	Used to write messages before module operations	R/W											

Attention:

- This register consists of 16 32-bit registers. The 16 32-bit registers are accessed sequentially, and the 32-bit register corresponding to the low address holds the high word of the data.
- Writes to this register can only be performed when the module is not in the computing state (HASH_CR.START), otherwise the hardware will automatically ignore write operations to this register.
- A read of this register will always result in all zeros.

32.4 True Random Number Generator (TRNG)

32.4.1 module block diagram

The TRNG module provides a true random number generator that generates a 64-bit random number.

The block diagram of the TRNG is shown in Figure 32-3 below. The random number generator is an analog random number oscillator circuit used to obtain random noise; the algorithm module captures the random noise and saves the result to the data module and outputs it through the bus; and the control module controls the mode and startup of the TRNG.

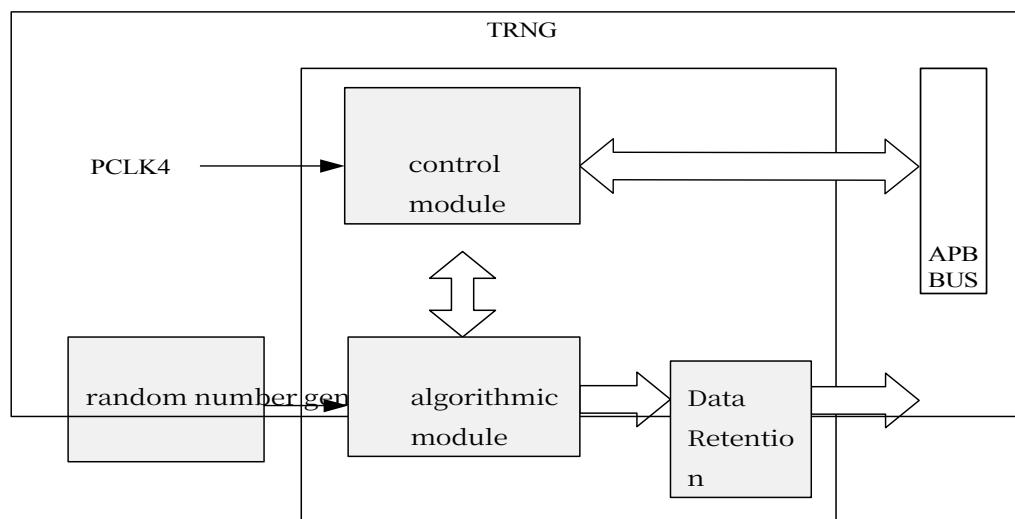


Figure 32-3 TRNG System Block Diagram

32.4.2 workflow

The true random number generation process is as follows:

1. Open 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. Starts random number generation (sets the RUN bit of TRNG_CR to 1)
4. Read random number (read TRNG_DR)
5. Turns off the random number generator circuit (sets the EN bit of TRNG_CR to 0)

32.4.3 Interrupt and event output

When the random number generation is finished, the register bit TRNG_CR.RUN is cleared to zero and a random number completion interrupt request (TRNG_END) is generated. When the random number generation is finished, it also generates an event output, which can be triggered by other modules.

32.4.4 Handling Precautions

To obtain good random numbers, set the frequency of the peripheral clock PCLK4 to less than

1MHz.

32.4.5 Register Description

base adr: 0x4004_1000

register name	notation	offset	bit width	reset value
TRNG Control Register	TRNG_CR	0x0000	32	0x0000_0000
TRNG Mode Register	TRNG_MR	0x0004	32	0x0000_0012
TRNG data register 0	TRNG_DR0	0x000C	32	0x0800_0000
TRNG data register 1	TRNG_DR1	0x0010	32	0x0800_0200

32.4.5.1 TRNG Control Register (TRNG_CR)

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														RUN	EN

classifier for marking honorific people	celebrity	functionality	fill out or in (information on a form)
b31~b3	Reserved	-	Reads "0", writes "0".
b2	Reserved	-	Reads "0", writes "0".
b1	RUN	Random number operation begins	0: Random number operation stops 1: Random number operations begin
			The software writes "1" to generate a new 64-bit random number; after the run is complete, the hardware clears it.
b0 number generator circuit	EN	Analog oscillator enable	0: Disable analog random 1: Open the analog random number generator circuit
			R/W

32.4.5.2 TRNG Mode Register (TRNG_MR)

Reset value: 0x0000_0012

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												CNT [2:0]	-	LOAD		
classifier for marking celebrity functionality fill out or in honorific (information people on a form)																
b31~b5	Reserved		-	Reads "0", writes "0".												R
Shift count control bit when capturing random noise 011: 32 shifts																
b4~2	CNT [2:0]		Shift count control bit	100: 64 shifts 101: 128 shifts 110: 256 shifts 000~010, 111: Function reserved bits												R/W
b1	Reserved		-	Reads "1" and writes "1".												R/W
Whether the data register is loaded with new initial values from the random number generator before random number generation																
b0	LOAD		Load control position	0: no new initial values are loaded 1: Load new initial values												R/W

32.4.5.3 TRNG Data Register (TRNG_DR)

Reset value: DR0: 0x0800_0000 DR1:

0x0800_0200

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/Write																
b31-b0	DATA[31:0]		random number	64-bit random number												R

33 Data Computing Unit (DCU)

33.1 summary

The Data Computing Unit (DCU) is a module that simply processes data without the aid of a CPU. Each DCU unit has 3 data registers and can add or subtract 2 data, compare sizes, and perform window comparison. This product is equipped with 4 DCUs, each of which can perform its own functions independently.

Function Summary:

- 4 types of data processing are possible: 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 results are stored in DATA0.
- Addition and subtraction can optionally be calculated after writing registers or triggered by other peripheral circuit events to perform the operation.
- Addition and subtraction operations can automatically halve the result once and put the halved result and the result of addition and subtraction operations into 2 data registers for other modules.
- The Compare mode can compare 2 data between DATA0 and DATA1 registers, and between DATA0 and DATA2 registers, and can be selected to generate interrupt and flag bits when greater than, less than, or equal to are satisfied, respectively.
- Comparison mode can be used for window comparison, i.e., 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 circuits and generate various interrupt and event signals according to the result of the operation. The event signals are used to start the peripheral circuits when other peripheral circuits with hardware trigger startup function select DCU as the trigger source, and then DCU generates the event signals to start the peripheral circuits to start the action.

33.2 Functional Description

33.2.1 additive mode

The addition mode calculates the sum of DATA0 and DATA1, where DATA0 is used as the added number and DATA1 is used as the additive number. Each write to the DATA1 register performs a $(\text{DATA0} + \text{DATA1})/2$ operation, the result of $\text{DATA0} + \text{DATA1}$ is stored in DATA0, and (DATA0) is stored in DATA0.

+ The result of $(\text{DATA0} + \text{DATA1})/2$ is stored in DATA2. When the result of $\text{DATA0} + \text{DATA1}$ exceeds 0xFF (8-bit mode) or 0xFFFF (16-bit mode) or 0xFFFF_FFFF (32-bit mode), a flag bit is generated and an interrupt is generated.

33.2.2 subtractive mode

The subtraction mode calculates the difference between DATA0 and DATA1, where DATA0 is used as the subtracted number and DATA1 is used as the subtracted number. Each write to the DATA1 register performs a $(\text{DATA0}-\text{DATA1})/2$ operation, the result of $\text{DATA0}-\text{DATA1}$ is stored in DATA0, and the result of (DATA0) is stored in DATA0, while (DATA0) is stored in DATA1.

The result of $(\text{DATA0}-\text{DATA1})/2$ is stored in DATA2, and a flag bit is generated and an interrupt is generated when the result of $\text{DATA0}-\text{DATA1}$ is less than 0x0 (in 8-bit, 16-bit, and 32-bit modes).

33.2.3 hardware-triggered boot mode

The DCU is capable of triggering startup operations based on events generated by peripheral circuits. To use the hardware-triggered startup mode, it is necessary to enable the peripheral circuit trigger function in the Function Clock Control 0 register (FCG0) first. Each DCU unit can independently select the trigger start signal sent from other peripheral circuits. When selecting the start signal, write the number of the peripheral circuit start source to be selected in the Trigger Source Selection Register (DCU_TRGSELx). When this peripheral circuit event occurs, it will input the event signal into DCU and trigger DCU to start to begin operation. The hardware trigger start mode includes trigger plus mode and trigger minus mode. In trigger plus mode, every time an event trigger occurs, the DCU will start and perform a $(\text{DATA0} + \text{DATA1})/2$ operation, the result of $\text{DATA0} + \text{DATA1}$ is stored in DATA0, and the result of $(\text{DATA0} + \text{DATA1})/2$ is stored in DATA2. When the result of $\text{DATA0} + \text{DATA1}$ exceeds 0xFF (8-bit mode) or 0xFFFF (16-bit mode) or 0xFFFF_FFFF (32-bit mode), a flag bit is generated and an interrupt is generated. In Trigger Minus Mode, for each event trigger, DCU will start and perform one $(\text{DATA0}-\text{DATA1})/2$ operation, the result of $\text{DATA0}-\text{DATA1}$ is stored in DATA0, and the result of $(\text{DATA0}-\text{DATA1})/2$ is stored in DATA2. When the result of $\text{DATA0}-\text{DATA1}$ is less than 0xFF (8bit mode) or 0xFFFF (16bit

mode) or 0xFFFF_FFFF (32bit mode), a flag bit will be generated and an interrupt will be generated. DATA0 -DATA1 result is less than 0x0 (**8-bit, 16-bit, 32-bit mode**), a flag bit is generated and an interrupt is generated.

33.2.4 comparison mode

Compares the sizes of DATA0 and DATA1 and DATA0 and DATA2, optionally 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 DATA1.

A flag bit is generated at DATA2 and an interrupt is generated. Compare mode allows you to select the conditions under which the comparison of data begins, compare after writing DATA0 or compare after writing any data register.

33.2.5 Interrupt and event signal output

The DCU has a variety of interrupts and event outputs for triggering the activation of other peripheral circuits for user selection. The control of interrupt and event outputs is controlled by the interrupt condition selection register (**DCU_INTEVTSEL**). When an event signal needs to be output, the user needs to make the corresponding control position of the interrupt condition selection register (**DCU_INTEVTSEL**) active. Each DCU unit outputs one DCU event signal, which is DCU1/DCU2/DCU3/DCU4 in the event list, and when it is necessary to generate an interrupt when the corresponding event occurs, the user needs to set the corresponding control bit of the interrupt condition selection register (**DCU_INTEVTSEL**) to be valid, and at the same time, set the INTEN bit of the control register (**DCU_CTL**) **to be 1**. Set the INTEN bit of the control register (**DCU_CTL**) to 1. Each DCU unit outputs one DCU interrupt signal, which is DCU1/DCU2/DCU3/DCU4 in the interrupt list.

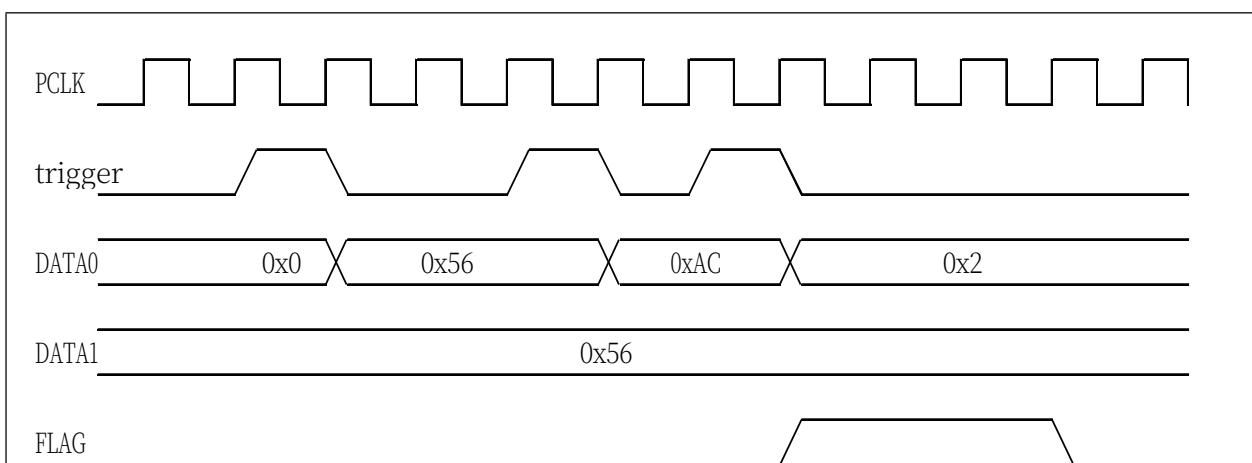
33.3 Application Examples

33.3.1 additive mode

Configure the registers to realize: select the addition mode, the data width is 16bit, write 0xFF00 and 0x55 in DATA0 and DATA1 respectively, then the calculation result is 0xFF55, the result is saved in DATA0, continue to write 0xFF in DATA1, then the calculation result overflows, and the result is generated as FLAG, and the result is cleared by the software by writing the FLAG register. FLAG.

33.3.2 Trigger plus mode

Configure the registers to realize: select the trigger add mode, the data width is 8bit, write 0x00 and 0x56 in DATA0 and DATA1 respectively, and write the event number of the peripheral circuit in the Trigger Source Selection Register. When the peripheral circuit is activated and generates an event, the DCU is triggered by the event and performs an addition operation, the result is 0x56 and is stored in DATA0. After 3 consecutive triggers, the calculation result overflows and generates the result FLAG, which is cleared by writing the FLAG register by the software.



33.3.3 comparison mode

Configure the registers to realize: select the compare mode, the data width is **16bit**, the FLAG output condition is DATA0 > DATA1, and the comparison starts after writing DATA0. First, write 0xBBB and 0xAAAA to DATA1 and DATA0 respectively, then no FLAG is generated because DATA0 > DATA1 is not satisfied, then write 0x8888 to DATA1, then DATA0 > DATA1 is satisfied, but the comparison starts after DATA0 is written, so no FLAG is generated. When 0x9999 is written to DATA0, the FLAG condition is satisfied and a FLAG is generated and reset by writing to the FLAG register.

33.4 Register Description

Register

List Unit 1

name (of a thing)	Abbreviations	clarification	address
DCU1 control register	DCU1_CTL	Configuring the DCU's Action Mode	0x4005_2000
DCU1 Flag Register	DCU1_FLAG	DCU's result marking	0x4005_2004
DCU1 data register 0	DCU1_DATA0	computing data	0x4005_2008
DCU1 data register 1	DCU1_DATA1	computing data	0x4005_200C
DCU1 data register 2	DCU1_DATA2	computing data	0x4005_2010
DCU1 Flag Reset Register	DCU1_FLAGCLR	Clearing the DCU's result marker	0x4005_2014
DCU1 Interrupt Condition Selection Register	DCU1_INTEVTSEL	Selecting conditions for DCU to generate interrupts	0x4005_2018

Module 2

name (of a thing)	Abbreviations	clarification	address
DCU2 Control Register	DCU2_CTL	Configuring the DCU's Action Mode	0x4005_2400
DCU2 Flag Register	DCU2_FLAG	DCU's result marking	0x4005_2404
DCU2 data register 0	DCU2_DATA0	computing data	0x4005_2408
DCU2 data register 1	DCU2_DATA1	computing data	0x4005_240C
DCU2 Data Register 2	DCU2_DATA2	computing data	0x4005_2410
DCU2 Flag Reset Register	DCU2_FLAGCLR	Clearing the DCU's result marker	0x4005_2414
DCU2 Interrupt Condition Selection Register	DCU2_INTEVTSEL	Selecting conditions for DCU to generate interrupts	0x4005_2418

Module 3

name (of a thing)	Abbreviations	clarification	address
DCU3 Control Register	DCU3_CTL	Configuring the DCU's Action Mode	0x4005_2800
DCU3 Flag Register	DCU3_FLAG	DCU's result marking	0x4005_2804
DCU3 data register 0	DCU3_DATA0	computing data	0x4005_2808
DCU3 data register 1	DCU3_DATA1	computing data	0x4005_280C
DCU3 Data Register 2	DCU3_DATA2	computing data	0x4005_2810
DCU3 Flag Reset Register	DCU3_FLAGCLR	Clearing the DCU's result marker	0x4005_2814
DCU3 Interrupt Condition	DCU3_INTEVTSEL	Selecting conditions for DCU	0x4005_2818

Selection Register

to generate interrupts

Module 4

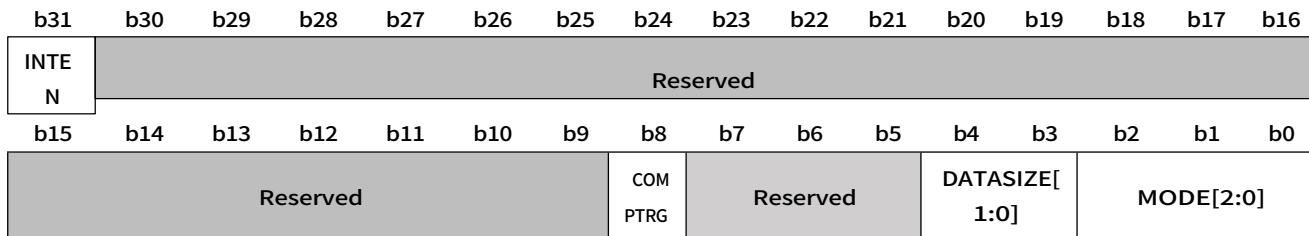
name (of a thing)	Abbreviations	clarification	address
DCU4 Control Register	DCU4_CTL	Configuring the DCU's Action Mode	0x4005_2C00
DCU4 Flag Register	DCU4_FLAG	DCU's result marking	0x4005_2C04
DCU4 data register 0	DCU4_DATA0	computing data	0x4005_2C08
DCU4 data register 1	DCU4_DATA1	computing data	0x4005_2C0C
DCU4 data register 2	DCU4_DATA2	computing data	0x4005_2C10
DCU4 Flag Reset Register	DCU4_FLAGCLR	Clearing the DCU's result marker	0x4005_2C14
DCU4 Interrupt Condition Selection Registers	DCU4_INTEVTSEL	Selecting conditions for DCU to generate interrupts	0x4005_2C18

33.4.1 DCU Control Register (DCU_CTL)

Register Description: This register is used to configure the action mode of the DCU.

Addresses: 0x4005_2000, 0x4005_2400, 0x4005_2800, 0x4005_2C00

Reset value: 0x8000_0000



classifier for honorific people	marking	celebrity	functionality	fill out or in (informati on on a form)
b31	INTEN	interrupt enable	0: Interrupt generation is not allowed 1: Allow generation of interrupts	R/W
b30~b9	Reserved	-	Read 0 when reading, please write 0 when writing	R
b8	COMPTRG	Compare Mode Trigger Timing of Comparison	0: Compare after writing to DATA0 1: Compare after writing DATA0 or DATA1 or DATA2	R/W
b7~b5	Reserved	-	Read 0 when reading, please write 0 when writing	R
b4~b3	DATASIZE[1:0]	data size	00: 8bit 01: 16bit 10: 32bit	R/W
b2~b0	MODE[2:0]	Action Mode	000: DCU invalid 001: addition mode, operation is performed after writing data to DATA1 register 010: Subtraction mode, operation is performed after writing data to DATA1 register 011: Hardware-triggered addition mode, triggered by other peripheral circuits to initiate addition operations 100: Hardware-triggered subtraction mode, triggered by other peripheral circuits to initiate subtraction operations 101: Comparison mode	R/W

33.4.2 DCU Flag Register (DCU_FLAG)

Register Description: This register generates the result identifier of the DCU.

Addresses: 0x4005_2004, 0x4005_2404, 0x4005_2804, 0x4005_2C04

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																																															
<hr/>																																															
<table border="0"> <tr> <td>classifier for honorific people</td><td>marking</td><td>Bit Name</td><td>Function</td><td>Read/Write</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>																classifier for honorific people	marking	Bit Name	Function	Read/Write																											
classifier for honorific people	marking	Bit Name	Function	Read/Write																																											
<table border="0"> <tr> <td>b31~b7</td><td>Reserved</td><td></td><td>-Read 0 when reading, please write 0</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>when</td></tr> <tr> <td></td><td></td><td>writing</td><td>R</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>																b31~b7	Reserved		-Read 0 when reading, please write 0												when			writing	R												
b31~b7	Reserved		-Read 0 when reading, please write 0												when																																
		writing	R																																												
<table border="0"> <tr> <td>b6</td><td>FLAG_GT1</td><td>Greater than flag bit 1 Comparison mode, set when DATA0 > DATA1, DCU Flag Reset Register</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>R</td></tr> <tr> <td></td><td></td><td>Cleared by writing 1 to the CLR_GT1 bit of DCU_FLAGCLR</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>																b6	FLAG_GT1	Greater than flag bit 1 Comparison mode, set when DATA0 > DATA1, DCU Flag Reset Register													R			Cleared by writing 1 to the CLR_GT1 bit of DCU_FLAGCLR													
b6	FLAG_GT1	Greater than flag bit 1 Comparison mode, set when DATA0 > DATA1, DCU Flag Reset Register													R																																
		Cleared by writing 1 to the CLR_GT1 bit of DCU_FLAGCLR																																													
<table border="0"> <tr> <td>b5</td><td>FLAG_EQ1</td><td>Equal to flag bit 1 Comparison mode, set when DATA0 = DATA1, DCU flag reset register</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr> <td></td><td></td><td>Cleared when CLR_EQ1 bit of DCU_FLAGCLR is written 1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>																b5	FLAG_EQ1	Equal to flag bit 1 Comparison mode, set when DATA0 = DATA1, DCU flag reset register																Cleared when CLR_EQ1 bit of DCU_FLAGCLR is written 1													
b5	FLAG_EQ1	Equal to flag bit 1 Comparison mode, set when DATA0 = DATA1, DCU flag reset register																																													
		Cleared when CLR_EQ1 bit of DCU_FLAGCLR is written 1																																													
<table border="0"> <tr> <td>b4</td><td>FLAG_LS1</td><td>Less than flag bit 1 Comparison mode, set when DATA0 < DATA1, DCU Flag Reset Register</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr> <td></td><td></td><td>Cleared when CLR_LS1 bit of DCU_FLAGCLR is written 1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>																b4	FLAG_LS1	Less than flag bit 1 Comparison mode, set when DATA0 < DATA1, DCU Flag Reset Register																Cleared when CLR_LS1 bit of DCU_FLAGCLR is written 1													
b4	FLAG_LS1	Less than flag bit 1 Comparison mode, set when DATA0 < DATA1, DCU Flag Reset Register																																													
		Cleared when CLR_LS1 bit of DCU_FLAGCLR is written 1																																													
<table border="0"> <tr> <td>b3</td><td>FLAG_GT2</td><td>Greater than flag bit 2 Comparison mode, set when DATA0 > DATA2, DCU flag reset register</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>R</td></tr> <tr> <td></td><td></td><td>Cleared by writing 1 to the CLR_GT2 bit of DCU_FLAGCLR</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>																b3	FLAG_GT2	Greater than flag bit 2 Comparison mode, set when DATA0 > DATA2, DCU flag reset register													R			Cleared by writing 1 to the CLR_GT2 bit of DCU_FLAGCLR													
b3	FLAG_GT2	Greater than flag bit 2 Comparison mode, set when DATA0 > DATA2, DCU flag reset register													R																																
		Cleared by writing 1 to the CLR_GT2 bit of DCU_FLAGCLR																																													
<table border="0"> <tr> <td>b2</td><td>FLAG_EQ2</td><td>Equal to flag bit 2 Comparison mode, set when DATA0 = DATA2, DCU flag reset register</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr> <td></td><td></td><td>Cleared when CLR_EQ2 bit of DCU_FLAGCLR is written 1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>																b2	FLAG_EQ2	Equal to flag bit 2 Comparison mode, set when DATA0 = DATA2, DCU flag reset register																Cleared when CLR_EQ2 bit of DCU_FLAGCLR is written 1													
b2	FLAG_EQ2	Equal to flag bit 2 Comparison mode, set when DATA0 = DATA2, DCU flag reset register																																													
		Cleared when CLR_EQ2 bit of DCU_FLAGCLR is written 1																																													
<table border="0"> <tr> <td>b1</td><td>FLAG_LS2</td><td>Less than flag bit 2 Comparison mode, set when DATA0 < DATA2, DCU Flag Reset Register</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr> <td></td><td></td><td>Cleared when CLR_LS2 bit of DCU_FLAGCLR is written 1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>																b1	FLAG_LS2	Less than flag bit 2 Comparison mode, set when DATA0 < DATA2, DCU Flag Reset Register																Cleared when CLR_LS2 bit of DCU_FLAGCLR is written 1													
b1	FLAG_LS2	Less than flag bit 2 Comparison mode, set when DATA0 < DATA2, DCU Flag Reset Register																																													
		Cleared when CLR_LS2 bit of DCU_FLAGCLR is written 1																																													
<table border="0"> <tr> <td>b0</td><td>FLAG_OP</td><td>Operational Flags Add, Subtract and Trigger Add, Trigger Subtract modes are set when addition generates an overflow or subtraction generates an underflow bit, cleared when the CLR_OP bit of the DCU flag reset register DCU_FLAGCLR is written to 1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>																b0	FLAG_OP	Operational Flags Add, Subtract and Trigger Add, Trigger Subtract modes are set when addition generates an overflow or subtraction generates an underflow bit, cleared when the CLR_OP bit of the DCU flag reset register DCU_FLAGCLR is written to 1																													
b0	FLAG_OP	Operational Flags Add, Subtract and Trigger Add, Trigger Subtract modes are set when addition generates an overflow or subtraction generates an underflow bit, cleared when the CLR_OP bit of the DCU flag reset register DCU_FLAGCLR is written to 1																																													

33.4.3 DCU Data Register (DCU_DATAx) (x=0,1,2)

Register Description: Used to store the operation data. The function of each data register in each mode is as follows:

	DATA0	DATA1	DATA2
additive mode	Number of additions/stored results	addition	Halving results
Trigger plus mode	Number of additions/stored results	addition	Halving of results
subtractive mode	Subtracted number/stored result	subtractions	Halving of results
Trigger minus mode	Subtracted number/stored result	subtractions	Halving of results
comparison mode	object of comparison	Comparison Object 1	Comparison object 2
Comparison mode (window comparison)	object of comparison	window ceiling	lower limit of a window

Addresses: 0x4005_2008, 0x4005_2408, 0x4005_2808,

0x4005_2C08 0x4005_200C, 0x4005_240C, 0x4005_280C,

0x4005_2C0C

0x4005_2010, 0x4005_2410, 0x4005_2810, 0x4005_2C10

Reset value: 0x0000_0000

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]															

Bit Flag	Bit Name	Function	Read/Write
b31~b0	DAT[31:0]	Stores the operation data, and the actual number of bits used is set according to DCU_CTL.DATASIZE.	
Operational Data		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

33.4.4 DCU Flag Reset Register (DCU_FLAGCLR)

Register Description: This register is used to clear the result identifier of the DCU.

Addresses: 0x4005_2014, 0x4005_2414, 0x4005_2814, 0x4005_2C14

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															
classifier for honorific people	marking	celebrity	functionality										fill out or in (information on a form)		
b31~b7	Reserved	-	Read 0 when reading, please write 0 when writing										R/W		
b6	CLR_GT1	Clear greater-than flag 1	Write 1 clears the FLAG_GT1 bit of DCU_FLAG, write 0 has no effect										R/W		
b5	CLR_EQ1	Clear equals flag bit 1	Write 1 clears the FLAG_EQ1 bit of DCU_FLAG, write 0 has no effect										R/W		
b4	CLR_LS1	Clear less-than flag 1	Write 1 clears the FLAG_LS1 bit of DCU_FLAG, write 0 has no effect										R/W		
b3	CLR_GT2	Clear greater than flag 2	Write 1 clears the FLAG_GT2 bit of DCU_FLAG, write 0 has no effect										R/W		
b2	CLR_EQ2	Clear the equal flag bit 2	Write 1 clears the FLAG_EQ2 bit of DCU_FLAG, write 0 has no effect										R/W		
b1	CLR_LS2	Clear less than flag 2	Write 1 clears the FLAG_LS2 bit of DCU_FLAG, write 0 has no effect										R/W		
b0	CLR_OP	Clear the operation flag bit	Write 1 clears the FLAG_OP bit of DCU_FLAG, write 0 has no effect										R/W		

33.4.5 DCU interrupt condition selection register (DCU_INTEVTSEL)

Register description: this register enables selection of the conditions under which the DCU generates interrupts and outputs event signals

Addresses: 0x4005_2018, 0x4005_2418, 0x4005_2818, 0x4005_2C18

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							SEL_WIN[1:0]		SEL_GT1	SEL_EQ1	SEL_LS1	SEL_GT2	SEL_EQ2	SEL_LS2	SEL_OP

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b9	Reserved	-	Read 0 when reading, please write 0 when writing	R
b8~b7	SEL_WIN[1:0]	Window Compare Interrupt Conditions Select other interrupt and event signal generation conditions R/W selection window, i.e.	Interrupt and output event signals are generated when the window comparison conditions set by SEL_WIN are met in comparison mode, but no interrupt and output event signals are generated when other comparison conditions are met when the SEL_WIN setting is valid. 00: Window compare interrupt and output event signal are not generated, and this setting is selected by b1~b6 of this register.	
b6	SEL_GT1	Greater than interrupt condition selection 1 compare mode and SEL_WIN=00, interrupt and output are generated when DATA0 > DATA1. The bit is invalid when SEL_WIN ≠ 00.	In R/W	
b5	SEL_EQ1	Equivalent to interrupt condition selection 1 compare mode and SEL_WIN=00, interrupt and output are generated when DATA0 = DATA1. The bit is invalid when SEL_WIN ≠ 00.	In R/W	
b4	SEL_LS1	Less than interrupt condition selection 1 compare mode and SEL_WIN=00, interrupt and output are generated when DATA0 < DATA1. signal, this bit is invalid when SEL_WIN≠00.	In R/W	
b3	SEL_GT2	Greater than interrupt condition selection 2 compare mode and SEL_WIN=00, interrupt and output are generated when DATA0 > DATA2. The bit is invalid when SEL_WIN ≠ 00.	In R/W	
b2	SEL_EQ2	Equivalent to interrupt condition selection 2 compare mode and SEL_WIN=00, interrupt and output are generated when DATA0 = DATA2. The bit is invalid when SEL_WIN ≠ 00.	In R/W	

b1	SEL_LS2	Less than interrupt condition selection 2 compare mode and SEL_WIN=00, interrupt and output are generated when DATA0<DATA2. R/W	In signal, this bit is invalid when SEL_WIN≠00.
b0	SEL_OP	Arithmetic Flag Bits	Generates an interrupt and outputs an event signal when the result of an operation overflows or underflows in addition and subtraction modes. R/W

33.5 caveat

Do not use the first cycle output waveform when selecting the triangle waveform output mode or the incremental sawtooth waveform output mode.

34CRC Operations (CRC)

34.1 summary

In many applications, the CRC algorithm is needed to verify the integrity and correctness of data. Especially in data transmission, CRC check is widely used. This module can use CRC16 and CRC32 algorithms to calculate and check the data.

34.2 functional block diagram

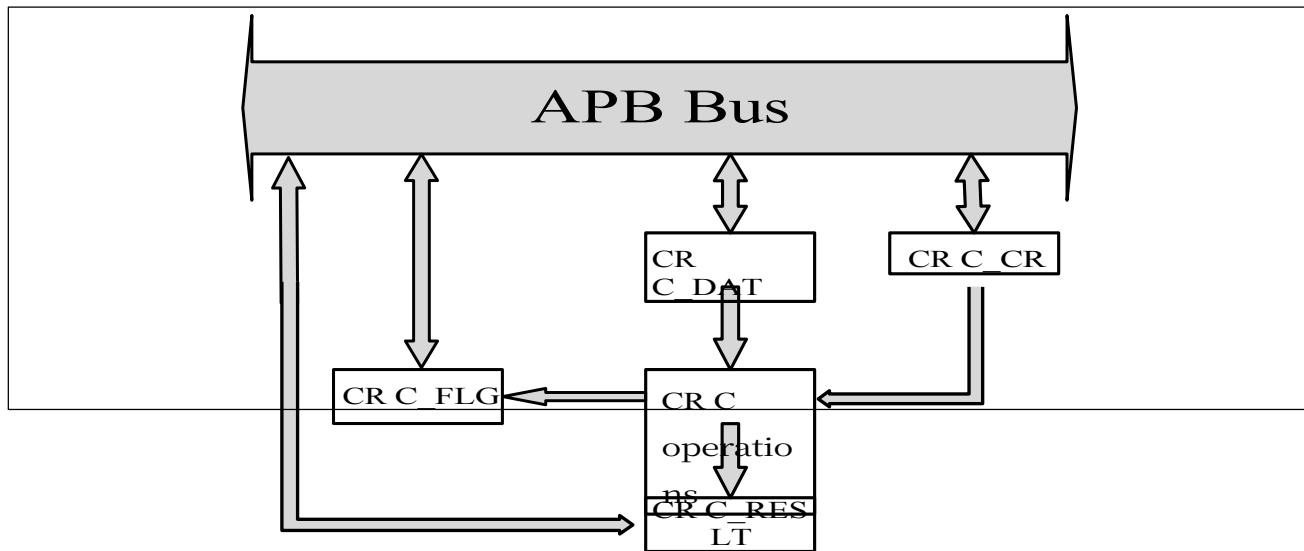


Figure 34-1 CRC Module Block Diagram

34.3 Functional Description

The CRC algorithm in this module follows the definition of ISO/IEC13239 and uses 32-bit and 16-bit CRCs respectively. the generating polynomial for CRC32 is $X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$. the generating polynomial for CRC16 is $X^{16}+X^{12}+X^5+1$. the generating polynomial for CRC16 is $X^{16}+X^{12}+X^5+1$. the generating polynomial for CRC16 is $X^{16}+X^{12}+X^5+1$.

The functions of this module include CRC code generation and CRC code verification.

34.3.1 CRC code generation

CRC encoding is the operation of a string of data to produce a 16/32 bit CRC encoded result. The procedure is as follows:

1. Set the CRC_CR register bits XOROUT,REFOUT,REFIN,CRC_SEL as required.
2. CRC16 (CRC_SEL=0) write 16-bit initial value to CRC_REG[15:0] of CRC_RESLT register
Center.
CRC32 (CRC_SEL=1) writes the 32-bit initial value to CRC_REG[31:0] of the CRC_RESLT register.
3. The data to be calculated is written to the CRC_DAT register sequentially, and each write operation corresponds to the input of 1 data (16-bit, 32-bit). For example, if there are 10 pieces of data, the CRC_DAT register is written 10 times in sequence.
4. After writing all the data to be computed to the CRC_DAT register, read the CRC_RESLT register CRC_REG to get the 16/32-bit CRC encoded value.

34.3.2 CRC calibration

CRC verification is to judge a string of data and the 16 / 32 - bit CRC code to check whether it is correct or not, and it only supports the case that the initial value of 16/32 bits is all 1, and only supports the verification of 16/32-bit CRC in ISO/IEC13239.

1. Set CRC_CR as required, including XOROUT,REFOUT,REFIN,CRC_SEL.
2. CRC16 (CRC_SEL=0):Write the 16-bit initial value into CRC_REG[15:0] of the CRC_RESLT register.
CRC32 (CRC_SEL=1):Write the 32-bit initial value into CRC_REG[31:0] of the CRC_RESLT register.
3. Writes the data to be calculated to the CRC_DAT register.
4. Writes the 16/32-bit CRC encoded value to the CRC_DAT register.
5. Read the CRC_FLG register, if it is 1, it means the checksum is successful, if it is 0, it means the checksum fails (for CRC16, you can also read the CRC_RESLT[16],CRCFLAG_16 bits to judge whether the checksum is

successful).

34.3.3 CRC checksums for XOROUT,REFOUT,REFIN not all 1's.

When the options XOROUT, REFOUT, REFIN are not all 1s, the CRC encoded value that follows the data input needs to be transformed as follows before it is written to CRC_DAT for checking.

1. When REFOUT=0, all BITS of the CRC encoded value are inverted; otherwise, no inversion is necessary.
2. When XOROUT=0, the result of (1) is inverted; otherwise, it is not inverted;
3. When REFIN=0, the BIT of each BYTE in the result of (2) is completely reversed; otherwise, it is not reversed;
4. Write the result of (3) to the CRC_DAT register for checksum.

34.4 Register Description

Table 34-1 shows the register list of the

CRC module. CRC_BASE_ADDR:

0x40008C00

Table 34-1 CRC Register List

register name	notation	offset address	bit width	reset value
CRC Control Register	CRC_CR	0x00	32	0x0000_001C
CRC Result Register	CRC_RESLT	0x04	32	0x0000_0000
CRC Flag Register	CRC_FLG	0x0C	32	0x0000_0000
CRC Data Register	CRC_DAT	0x80~0xFC	32	0xFFFF_FFFF

34.4.1 Control Register (CRC_CR)

Reset value: 0x0000001C

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
-	-	-	-	-	-	-	-	-	-	-	XOROUT	REFOUT	REFIN	CR	-

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b30~b5	Reserved	-	Reads "0", writes "0".	R
b4	XOROUT	Inverted result output	0: Result is not inverted 1: Inverted result output	R/W
b3	REFOUT	Output all digits of the result inverted of the result are not inverted.	0: All digits 1: Output with all digits of the result reversed	R/W
b2	REFIN	Bit reversal in input data byte	0: No bit inversion in input data byte 1: Bit reversal within the input data byte	R/W
b1	CR	Operational bit number selection 0: CRC16 1: CRC32		R/W
b0	Reserved	-	Reserved Bit	R/W

34.4.2 Result Register (CRC_RESLT)

Reset value: 0x00000000

Select CRC16: (CRC_SEL=0)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CRCFLAG_16
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
CRC_REG[15:0]															

classifier for marking	celebrity	functionality	fill out or in (information on a form)
honorable			
people	<u>b16</u> CRCFLAG_16	CRC16 checksum result	0: CRC16
b31~b17	Reserved	arithmetic checksum result. Reader "0", writes "0".	R
		1: CRC16 operation checksums correctly	
b15~b0	CRC_REG[15:0]	Result Bit	This 16-bit register is used to update and save the result of each CRC16 calculation; after the operation, the reads this register to get the 16-bit CRC encoding result. R/W

Select CRC32: (CRC_SEL=1)

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

Bit Flag	Bit Name	Function	Read/Write
b31~b0	CRC_REG[31:0]	result bits each CRC32 calculation; after the operation, the	This 32-bit register is used to update and save the result of Reading this register will result in a 32-bit CRC encoded result R/W

34.4.3 Flag Register (CRC_FLG)

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
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CRC_FLAG

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b1	Reserved	-	Reads "0", writes "0".	R
b0	CRC_FLAG	CRC32 check result	CRC check result flag bit; 0: current checksum error; 1: current checksum is correct	R

34.4.4 Data Register (CRC_DAT)

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
CRC_DAT[31:16]															
CRC_DAT[15:0]															

Bit Flag	Bit Name	Function	Read/Write	
b31~b0 Register	CRC_DAT[31:0]	Data		This register is used to enter the data to be operated on; the address of this register is a range (0x80~0xFC), which corresponds to 32 addresses (each address corresponds to 32 bits of data) An operation on any of these 32 addresses is considered to be a manipulation of this register. Make.

35SDIO Controller (SDIOC)

35.1 summary

The SDIOC provides an SD host interface and an MMC host interface for communicating with SD cards that support the SD2.0 protocol, SDIO devices, and MMC devices that support the eMMC4.2 protocol.

SDIOC features the following:

- Supports SDSC, SDHC, SDXC format SD cards and SDIO devices.
- Supports 1-wire (1bit) and 4-wire (4bit) SD buses
- Supports 1-wire (1bit), 4-wire (4bit) and 8-wire (8bit) MMC buses
- SD clock up to 50MHz
- Card recognition and hardware write protection

This product comes with 2 SDIO controllers that can communicate with 2 SD/MMC/SDIO devices at the same time.

35.2 Functional Description

35.2.1 port allocation

port name	IO	functionality
SDIOx_CK(x=1~2)	o	SD clock output signal
SDIOx_CMD(x=1~2)	I/O	SD command and answer signals
SDIOx_D0(x=1~2)	I/O	SD Data Signal
SDIOx_D1(x=1~2)	I/O	SD Data Signal
SDIOx_D2(x=1~2)	I/O	SD Data Signal
SDIOx_D3(x=1~2)	I/O	SD Data Signal
SDIOx_D4(x=1~2)	I/O	SD Data Signal
SDIOx_D5(x=1~2)	I/O	SD Data Signal
SDIOx_D6(x=1~2)	I/O	SD Data Signal
SDIOx_D7(x=1~2)	I/O	SD Data Signal
SDIOx_CD(x=1~2)	I	SD card recognition status signal
SDIOx_WP(x=1~2)	I	SD write-protect status signal

According to SD2.0 protocol and eMMC4.51 protocol, SD clock output signal (SDIOx_CK ($x=1\sim 2$)) is used to output SD clock during communication between SDIOC and SD/MMC device; SD command and answer signal (SDIOx_CMD ($x=1\sim 2$)) is used to output SD/MMC commands to the device and to receive answer information from the device; SD data signal (SDIOx_Dy ($x=1\sim 2$) ($y=0\sim 7$)) is used to send and receive data during the communication between SDIOC and the device. SD data signal (SDIOx_Dy($x=1\sim 2$) ($y=0\sim 7$)) is used to send and receive data between SDIOC and the device during the communication process. When one-wire (**1bit**) communication is used, only SDIOx_D0($x=1\sim 2$) is valid, and SDIOx_Dy($x=1\sim 2$) ($y=1\sim 7$) is kept at a high level, and four-wire (**4bit**) **communication** is used. SDIOx_Dy($x=1\sim 2$) ($y=0\sim 3$) is valid and SDIOx_Dy($x=1\sim 2$) ($y=4\sim 7$) is held high when communicating, SDIOx_Dy($x=1\sim 2$) ($y=0\sim 7$) is valid when communicating using eight-wire (**8bit**) **communication**, which is limited to MMC device communication.

35.2.2 Basic Access Methods

The user initiates the SDIOC to communicate with off-chip SD/MMC devices by reading and writing the SDIOC registers. Since writing the command register will trigger the send command action of SDIOC, writing the command register must be done last. Before that, the user needs to set the transmission mode through TRANSMODE register, set the command parameters through parameter registers (ARG0,ARG1), and set the correct command **index**, **type**, **response type**, etc. when setting the command register (CMD). Once the write command register is executed, the SDIOC will send the command, and the user can read the

interrupt status register (**NORINTST,ERRINTST**) to check whether the response message is received or not, and whether there is any error message, etc. When a command is executed, the user can read the interrupt status register (**NORINTST,ERRINTST**) to check whether the response message is received or not. After a command is executed, the user can read the answer register (**RESP0~7**) to get the answer of the command.

35.2.3 data transmission

The data buffer registers of the SDIOC are used for data exchange between host devices such as CPU/DMA and SD devices. The SDIOC has built-in FIFOs to speed up data exchange. When the command contains data to be sent, the user writes the data into the data buffer registers (BUF0, BUF1) sequentially, and these data will be cached in the FIFO of SDIOC firstly, and when the number of written data reaches the setting value of data length register (BLKSIZE) or 512 bytes, SDIOC will send the data via SDIOx_Dy(x=1~2) (y=0~7). SDIOx_Dy(x=1~2) (y=0~7) During this time the SDIOC is able to continue writing data to the data buffer register through the FIFO. Similarly, when the command includes data reception, the SDIOC receives data via SDIOx_Dy(x=1~2) (y=0~7) and caches the data in the buffer (BUFFER) of the SDIOC, and the user gets the data by reading the data buffer registers (BUF0, BUF1). Please refer to the SD and MMC protocols for the format of the data.

35.2.4 SD Clock

The SD clock (SDIOx_CK (x=1~2)) is generated by the host computer and output to the device for communication between the two. The SD clock is generated by the action clock (EXCLK) of the SDIOC module through the frequency divider. The frequency divider coefficient is set by the clock control register (CLKCON). According to the SD2.0 protocol, the fastest frequency should be 50MHz in data transfer mode, so the user needs to set a reasonable frequency divider coefficient according to the frequency of EXCLK.

35.2.5 Interrupt and DMA startup requests

35.2.5.1 SD Interrupt

SDIOC provides two types of interrupts, normal interrupt and error interrupt. Normal interrupts are interrupts that occur when various events occur during communication between the SDIOC and the SD/MMC card. Error interrupt is the interrupt generated when various errors occur during the communication process of SDIOC action. The normal interrupt and error interrupt have their own interrupt status register, interrupt status enable register, and interrupt signal enable register. The interrupt status register is used to indicate the cause of the interrupt, the interrupt status enable register is used to enable each status bit of the interrupt status register, each status bit of the interrupt status register needs to be valid when the enable bit of the interrupt status enable register is enabled, and the interrupt signal enable register is used to whether or not to allow each interrupt cause to request an interrupt from the CPU.

35.2.5.2 SDIO Interrupt

The SDIO device can send a card interrupt request to the host when the transmission is idle. To use the SDIO interrupt, you need to enable the card interrupt (CINTEN) in the interrupt status enable register (NORINTSTEN), and if you need to request

an interrupt from the CPU, you also need to enable the card interrupt (CINTEN) in the interrupt signal enable register (NORINTSGEN). The SDIO device requests an interrupt from the host by pulling the SDIOx_D1(x=1~2) data line low. the SDIOC sets the card interrupt (CINT) in the interrupt status register (NORINTST) when it detects that SDIOx_D1(x=1~2) has been pulled low and requests an interrupt from the CPU according to the setting. the SDIO device releases SDIOx_D1(x=1~2) after it determines that the interrupt is finished processing. SDIOx_D1 (x=1~2).

35.2.5.3 DMA Request

SDIOC can read and write data in communication through DMA. When writing data to the device using DMA transmission, the start source of one channel of DMA is set to the write request of SDIOC, and the destination address of DMA transmission is set to the data buffer registers (BUF0, BUF1) and is a fixed address. After the SDIOC sends a write data command, if the data FIFO is empty at this time, it sends a write request signal to the DMA and starts the DMA to write data to the data buffer registers (BUF0, BUF1), which will be entered into the data FIFO in turn. When the data in the FIFO reaches the set value of the data length register (BLKSIZE) or 512 bytes, SDIOC will send the data via SDIOx_Dy ($x=1\sim 2$) ($y=0\sim 7$). In the case of a multi-block write command, the SDIOC will continue to send startup requests to the DMA to write data at the same time. When using DMA to transmit read data from the device, set the start source of the other channel of DMA to the read request of SDIOC, and then set the transmission source address of DMA to the data buffer registers (BUF0, BUF1) and fixed address. After SDIOC sends the read data command, the device will send data through SDIOx_Dy ($x=1\sim 2$) ($y=0\sim 7$), and the data FIFO of SDIOC will start to receive the data, and when the data in the FIFO reaches the setting value of the data length register (BLKSIZE) or 512 bytes, SDIOC sends a read request signal to the DMA, and starts the DMA to read the data from the data buffer register (BUF0, BUF1) and the data buffer register (BUF0, BUF1) and set the transmission source address of the DMA to a fixed address. If it is a multi-block read command, SDIOC will continue to read the next data block from the device at the same time.

35.2.5.4 Card insertion (insert) and removal (remove)

The insertion (**insert**) and removal (**remove**) of SD/MMC/SDIO devices are recognized by the SDIOx_CD ($x=1\sim 2$) signal line. When there is no device in the card slot, the card slot will pull up the SDIOx_CD($x=1\sim 2$) signal through a resistor. When there is a device inserted, the SDIOx_CD($x=1\sim 2$) signal will be pulled low and will become high again when the device is removed. SDIOC determines whether there is a device or not by the level of SDIOx_CD($x=1\sim 2$). The user can determine whether there is a device inserted by reading the CDPL bit of the host status register (PSTAT). SDIOC is able to generate corresponding interrupts when a device is inserted and removed, which are enabled by the interrupt status enable register (NORINTSEN) and interrupt signal enable register (NORINTSGEN).

35.2.6 Host and device initialization

35.2.6.1 Host Initialization

The SDIOC needs to be initialized first when it is used. The SDIOC initialization procedure is as follows:

1. Read the host status register (PSTAT) to query the clock status as well as the device insertion state
2. Configure the power control register (PWRCON) to enable SDIOC
3. Configure the clock control register (CLKCON) to enable the SDIOC to output the SD clock, and configure the SD clock division frequency according to the frequency of EXCLK to ensure that the SD clock frequency does not exceed 400KHz in card recognition mode.
4. Configure the HOSTCON register to select 1-bit mode and disable high speed mode.
5. Configure the timeout control register (TOUTCON) according to the device characteristics so that the host terminates the communication and reports an error when the communication timeout occurs.
6. Configure the Normal and Error Interrupt Status Enable Registers, the Interrupt Signal Enable Register to enable the SDIOC to generate interrupts when needed and to

Setting the sign position from

35.2.6.2 SD Card Initialization

After completing the SDIOC initialization configuration, if the SD card is connected. You need to initialize according to the SD protocol, and the initialization sequence is as follows:

1. Card reset, send the reset command CMD0 to the device, CMD0 no response message (response), this time the card will enter the card recognition mode
2. Confirms the card's operation condition, sends CMD8 to the device and waits for a response.
3. Send initialization command ACMD41 (send CMD55 first, then CMD41), according to the response information to determine whether the device has completed the initialization, otherwise continue to send ACMD41 until the initialization is completed
4. Send CMD2 to get the card's CID information.
5. Send CMD3 to get the RCA

information of the card At this point,
the card will enter the data transmission
mode and the initialization is complete.

35.2.6.3 MMC Card Initialization

After completing the SDIOC initialization configuration, if an MMC card is connected. You need to initialize according to the MMC protocol, and the initialization sequence is as follows:

1. Card reset, send the reset command CMD0 to the device, CMD0 no response message (response), this time the card will enter the card recognition mode
2. Confirm the operation condition of the card, send CMD1 to the device and wait to receive the response, judge whether the device has completed the initialization according to the response, otherwise keep sending CMD1 until the initialization is completed.
3. Send CMD2 to get the card's CID information.
4. Send CMD3 to get the RCA

information of the card At this point,
the card will enter the data transmission
mode and the initialization is complete.

35.2.6.4 SDIO Initialization

After completing the SDIOC initialization configuration, if an SDIO device is connected. You need to initialize according to the SDIO protocol, and the initialization sequence is as follows:

1. Device reset, send reset command CMD0 to the device, CMD0 no answer message (response)
2. Confirm the operation condition of the SDIO device, send CMD5 to the device and wait for the response.
3. Send CMD3 to get the RCA information of the device, initialization is complete.

35.2.7 SD/MMC single block read/write

After the SD/MMC card enters the data transfer mode, the SDIOC is able to access the data of the SD/MMC card through read and write commands. The single data block (block) read/write sequence is as follows:

1. Configure the clock control register (CLKCON) to enable the SDIOC to output the SD clock, and configure the SD clock division frequency according to the frequency of EXCLK so that the SD clock frequency does not exceed the maximum clock speed in the default speed mode (default speed mode, fpp<=25MHz).
2. Send CMD7 and the SD/MMC card will enter the data transfer state.
3. If necessary, for SD card, you can configure the host control register (HOSTCON) to set the host bus width and send ACMD6 to set the SD bus width (1bit or 4bit), for MMC card, you can configure the host control register (HOSTCON) to set the host bus width and rewrite the Ext_CSD register of the MMC card to set the bus width (1bit, 4bit or 8bit) by CMD6(SWITCH) command. CSD register of MMC card via CMD6 (SWITCH) command to set the bus width (1bit, 4bit or 8bit).
4. If necessary, for SD card, you can configure the data length register (BLKSIZE) to set the data block length, and set the data block size (number of bytes) through CMD16 command, the configuration range is 1~512 bytes. For SDHC/SDXC and MMC cards, the block size is fixed at 512 bytes.
5. If necessary, for SD card, you can configure the host control register (HOSTCON) to set the host to high speed mode and send CMD6 to switch the SD card to high speed mode (fpp<=50MHz), and after the successful switching, you can set SDIOx_CK(x=1~2) to Maximum 50MHz.
6. When writing data, first configure the transfer mode register (TRANSMODE) to set the transfer mode to write and single block transfer. Send single block write command CMD24, if CPU is used to write data, write data to data buffer registers (BUF0, BUF1) after confirming that BWR bit of interrupt status register (NORINTST) is 1. If DMA is used to write data, wait for the DMA transmission to complete, and SDIOC will send data through data signal SDIOx_Dy (x=1~2). SDIOC will send the data through the data signal SDIOx_Dy (x=1~2). After the transmission is completed, the TC of the interrupt status register (NORINTST) will be set to 1 and an interrupt request will be generated. If there is an error in the transmission process, the corresponding error flag bit will be set and an interrupt request will be generated.
7. When reading data, first configure the transmission mode register (TRANSMODE) to set the transmission mode as read, single block transmission. Send single block

read command CMD17, SDIOC will receive data by data signal SDIOx_Dy ($x=1\sim 2$). If CPU is used to read the data, after confirming the BRR bit of NORINTST is 1, it will read the data from the data buffer registers (BUF0, BUF1). If DMA is used to read the data, it will wait for the completion of the DMA transmission, and after the completion of the reading, the TC of NORINTST will be set to 1 and an interrupt request will be generated, and if there is an error in transmission, the corresponding error flag bit will be set. If there is an error in the transmission process, the corresponding error flag bit will be set and an interrupt request will be generated.

35.2.8 SD/MMC multi block reading and writing

The order of multiple data block (block) read/write and single data block (block) read/write is as follows:

1. Configure the clock control register (CLKCON) to enable the SDIOC to output the SD clock, and configure the SD clock division frequency according to the frequency of EXCLK so that the SD clock frequency does not exceed the maximum clock speed in the default speed mode.

fpp<=25MHz).

2. Send CMD7 and the SD/MMC card will enter the data transfer state.
3. If necessary, for SD card, you can configure the host control register (HOSTCON) to set the host bus width and send ACMD6 to set the SD bus width (1bit or 4bit), for MMC card, you can configure the host control register (HOSTCON) to set the host bus width and rewrite the Ext_CSD register of the MMC card to set the bus width (1bit, 4bit or 8bit) by CMD6(SWITCH) command. CSD register of MMC card via CMD6 (SWITCH) command to set the bus width (1bit, 4bit or 8bit).
4. If necessary, for SD card, you can configure the data length register (BLKSIZE) to set the data block length, and set the data block size (number of bytes) through CMD16 command, the configuration range is 1~512 bytes. For SDHC/SDXC and MMC cards, the block size is fixed at 512 bytes.
5. Configure the transfer mode register (TRANSMODE) to set the transfer mode (read/write), multi-block transfer. If the allow data block count enable is set, the number of data blocks to be transferred needs to be set in the data block count register (BLKCNT), and multi-block transfer cannot be started without setting the data block count register (BLKCNT). If the Disable Data Block Count Enable is set, there is no need to set the Data Block Count Register (BLKCNT), at this time an unlimited number of multi-block transfers can be performed, and when the host decides to end the transfer, it needs to send a CMD12 to the device to inform the device of the end of the data transfer after the last data block has been transferred.
6. If necessary, for SD card, you can configure the host control register (HOSTCON) to set the host to high speed mode and send CMD6 to switch the SD card to high speed mode (fpp<=50MHz), and after successful switching, you can set SDIOx_CK (x=1~2) to Maximum 50MHz.
7. When writing data, send the multi-block write command CMD25. If CPU is used to write data, after confirming that the BWR bit of the interrupt status register (NORINTST) is set to 1, then write data to the data buffer registers (BUF0, BUF1), and BWR will be kept at 0 during the sending process, and then reset to 1 at the end of the sending process, so that the user can write data of the second block at this time. If DMA is used to write data, wait for DMA transmission to complete, SDIOC will send data via data signal SDIOx_Dy (x=1~2). TC of the interrupt status register (NORINTST) will be set to 1 and an interrupt request will be generated. If an error occurs during the transmission, the corresponding error flag bit will be set and an interrupt request will be generated. After all the data have been sent, the SDIOC will automatically send a CMD12 to end the transmission if an automatic CMD12 has been set, otherwise it is necessary to manually send a CMD12

to the device to inform the device of the end of the data transmission.

8. When reading data, send the multi-block read command CMD18 and SDIOC will receive the data through the data signal SDIOx_Dy (x=1~2). If read data by CPU, after confirming the BRR bit of interrupt status register (NORINTST) is 1, read data from data buffer register (BUF0, BUF1), BRR will be kept 0 during receiving process, and reset to 1 after the end of transmitting, then the user can read the data of the second block at this time. If DMA is used to read the data, wait for the DMA transmission to be completed, after the reading is completed, the TC of the interrupt status register (NORINTST) will be set to 1 and an interrupt request will be generated, if there is an error in the transmission process, the corresponding error flag bit will be set and an interrupt request will be generated. After all data have been received, the SDIOC will automatically send a CMD12 to end the transmission if it is set to send CMD12 automatically, otherwise it needs to manually send a CMD12 to the device to inform the device that the data transmission is finished.

35.2.9 Transmission abort, suspend and resume.

When transferring multiple blocks, you can **abort** or **suspend** by software control. The **abort** operation can be performed regardless of whether the number of transferred blocks is set or not. The abort operation is divided into asynchronous abort and synchronous abort. The asynchronous abort operation requires sending CMD12 through the write command register (CMD) to abort the transmission while it is in progress, and the transmission will be terminated immediately. For write operation, the SD/MMC card will discard the current block of data and enter the write mode (program) to write the previously received block of data into FLASH. For a read operation, the SD/MMC will stop transferring data. Synchronization termination means that the transfer is stopped at the block interval by setting the block interval register (BLKGAP) while the transfer is in progress, and after the setting is completed, the transfer will be stopped at the end of the current block transfer. In this case, it is necessary to send CMD12 to end the transmission.

To perform a suspend operation, first set the SABGR bit in the block interval register (BLKGAP) to stop transmission at the block interval and enable the **read wait** function. After the setting is completed, the transmission stops at the end of the current data block. At this point, the transfer is suspended by sending CMD52 through the Write Command Register (CMD). After sending the suspend command, you need to continue to read the BS bit (**bus status**) of the CCCR register of the SDIO through CMD52, if BS=0, it means the transmission has been suspended, and the SD bus will turn into an idle state. If BS=0, the transfer is suspended and the SD bus turns to idle state. After the transfer is suspended, the host can perform other functions of the SDIO. However, if you want to resume operation later, you need to backup the SDIOC registers (offset address 00h~0Dh) after hangup, and restore these register settings after executing other operations. To perform the resume operation, first clear the block interval register (BLKGAP) setting that stops transmission at the block interval, and then send CMD52 through the write command register (CMD) to resume transmission. After sending the resume command, you need to continue to read the DF bit (**data flag**) of the CCCR register of SDIO through CMD52, if DF=1 means there is data to continue transmission after executing the resume, and if DF=0 means there is no data to be transmitted, at this time, you should write the software reset register (SFTRST) to reset the data line.

Attention:

- Suspend and resume operations require that the SDIO device and the combo card device support such operations as well as **read wait** operations. access to the CIA area (common I/O area) of SDIO does not support suspend and resume operations, and only the **read wait** function can be used. The **read wait** function can only be used.

35.2.10 read wait

Read **wait** allows the host to insert the CMD52 during a continuous data transfer to access other functions of the SDIO device. To perform read **wait**, first stop the transfer at the block interval by setting the SABGR bit in the Block Interval Register (BLKGAP) and enable **the** read **wait** function. After the setting is completed, the host will pull the SDIO_x_D2 ($x=1\sim2$) data line low at the end of the current data block transmission. The transfer will be paused at this time and the host can insert the CMD52 at this time to access other functions that do not require data transfer. When it is necessary to end the read **wait** function, the transfer is resumed by setting CR in the Block Interval Register (BLKGAP) and clearing SABGR.

Attention:

- read wait** Read wait requires SDIO device and combo card device support, and is performed under a four-wire bus transfer.

35.2.11 Wakeup

When not working for a long time, the system can be transferred to a low-power state to reduce power consumption. In the low-power state, **the** system can be woken up by **insertion/removal** and **card interrupt** of SD/MMC/SDIO devices to continue working. To use the wake-up function, you need to enable the **insert/removal** or **card interrupt** in the NORINTSEN interrupt status enable register and NORINTSGEN interrupt signal enable register. At the same time, enable the corresponding wake-up function of SDIOx_CD(x=1~2)/ SDIOx_D1(x=1~2) ports. After the configuration is completed, the system can be transferred to low-power mode. When **insert/removal** and **card interrupt** occur, the wake-up function of SDIOx_CD(x=1~2)/ SDIOx_D1(x=1~2) ports will wake up the system and **insert/removal** and **card interrupt** will occur. **interrupt**.

Use insert to wake up the stop mode process:

1. Configure the SDIOC port by selecting SDIO1_CD as PA10
2. Configure the PCR register of PA10 and select IRQ10 to be active
3. Configure the PWRC3 register of the power control module so that the CPU enters stop mode after executing the WFI command.
4. Configure the WUPEN register of the interrupt control module to enable the wake-up function of IRQ10.
5. Configure the EIRQCR10 register of the interrupt control module to select falling edge triggering
6. Configuring PWRCON Enabling SDIOC
7. Configure the interrupt status enable register (NORINTSEN) and the interrupt signal enable register (NORINTSGEN) by inserting the corresponding enable bits.
8. Configure the Interrupt Source Select Register to select the SDIOC interrupt as the interrupt source and enable the interrupt
9. Execute the WFI command to put the system into stop mode.
10. When the device is inserted, PA10 will be pulled low, waking up the CPU via IRQ10 and entering the interrupt subroutine according to the SD interrupt request.
11. Clear the insert status in the interrupt status register (NORINTST), exit the interrupt subroutine, and perform the subsequent operations.

Use insert to wake up the power down mode process:

1. Configure the SDIOC port by selecting SDIO1_CD as PA10
2. Configure the PCR register of PA10 and select IRQ10 to be active
3. Configure the PWRC3 and PWRC0 registers of the power control module so that the CPU enters power-down mode after executing the WFI command.
4. Configure the PDWKE1 register of the power control module to enable the power-down wake-up function of IRQ10
5. Configure the WUPEN register of the interrupt control module to enable the wake-up function of IRQ10.
6. Configure the EIRQCR10 register of the interrupt control module to select falling edge triggering
7. Configuring PWRCON Enabling SDIOC
8. Configure the interrupt status enable register (NORINTSTEN) and the interrupt signal enable register (NORINTSGEN) by inserting the corresponding enable bits.
9. Configure the Interrupt Source Select Register to select the SDIOC interrupt as the interrupt source and enable the interrupt
10. Execute the WFI command to put the system into power-down mode.
11. When the device is plugged in, PA10 will be pulled low, waking up the system to re-power up via IRQ10

35.3 Register Description

The SDIOC module is designed according to the **SD Host Controller Standard Specification**, so the registers are also the same as the standard description, and the unused addresses and bits are changed to Reserved.

The following table shows the register list of the SDIOC module.

BASE ADDR. 0x4006FC00 (SDIOC1) 0x40070000 (SDIOC2)

register name	notation	offset address	bit width	reset value
Data block length	BLKSIZE	0x04	16	0x0000h
data block count	BLKCNT	0x06	16	0x0000h
Parameter 0	ARG0	0x08	16	0x0000h
Parameter 1	ARG1	0x0A	16	0x0000h
transfer mode	TRANSMODE	0x0C	16	0x0000h
command	CMD	0x0E	16	0x0000h
Response 0	RESP0	0x10	16	0x0000h
Response 1	RESP1	0x12	16	0x0000h
Response 2	RESP2	0x14	16	0x0000h
Response 3	RESP3	0x16	16	0x0000h
Answer 4	RESP4	0x18	16	0x0000h
Answer 5	RESP5	0x1A	16	0x0000h
Response 6	RESP6	0x1C	16	0x0000h
Answer 7	RESP7	0x1E	16	0x0000h
Data buffer 0	BUF0	0x20	16	0x0000h
Data buffer 1	BUF1	0x22	16	0x0000h
host state	PSTAT	0x24	32	0x00000000h
Host Control	HOSTCON	0x28	8	0x00h
power control	PWRCON	0x29	8	0x00h
Data block interval control	BLKGPCON	0x2A	8	0x00h
Clock control	CLKCON	0x2C	16	0x0002h
timeout control	TOUTCON	0x2E	8	0x00h
software reset	SFTRST	0x2F	8	0x00h
General Interrupt Status	NORINTST	0x30	16	0x0000h
error interrupt status (computing)	ERRINTST	0x32	16	0x0000h
General Interrupt Status Enable	NORINTSTEN	0x34	16	0x0000h
Error Interrupt Status Enable	ERRINTSTEN	0x36	16	0x0000h
General Interrupt Signal Enable	NORINTSGEN	0x38	16	0x0000h
Error interrupt signal enable	ERRINTSGEN	0x3A	16	0x0000h
Auto Command Error Status	ATCERRST	0x3C	16	0x0000h
Forced auto-command error	FEA	0x50	16	0x0000h

state control

register name	notation	offset address	bit width	reset value
Forced error state control	FEE	0x52	16	0x0000h

In addition, SDIOC1 and SDIOC2 share an MMC mode enable register for the controller to switch between SD and MMC modes.

register name	notation	address	bit width	reset value
MMC Mode Enable Register	MMCER	0x40055404	32	0x00000000h

35.3.1 Data block length register (BLKSIZE)

Offset address:

0x04 Reset

value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved				TBS [11:0]											
<hr/>															
classifier for honorific people	marking	celebrity	functionality				fill out or in (information on a form)								
b15~b12	Reserved	-	Read 0 when reading, please write 0 when writing				R								
b11~0 Size).	TBS	Data Block Length	Sets the length of the transferred data block (Transfer Block Setting range is 1~512 in bytes.				R/W								

35.3.2 Data block counter register (BLKCNT)

Offset address:

0x6 Reset value:

0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0		
BC [15:0]														<hr/>			
<hr/>																	
Bit Flag	Bit Name	Function	Read/Write														
b15~b0	BC	Data block count	setting this register needs to be set when the transfer stops.	Sets the number of data blocks to be transferred (Block count), When this is done and the Block Count Enable bit (BCE) of the Transfer Mode Register is required to be active				R/W									

35.3.3 Parameter register 0 (ARG0)

Offset address:

0x08 Reset

value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0		
ARG0[15:0]														<hr/>			
<hr/>																	
classifier for honorific people	marking	celebrity	functionality				fill out or in (information on a form)										

35.3.4 Parameter register 1 (ARG1)

Offset address:

0x0A Reset

value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ARG1[15:0]															

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b15~b0	ARG [15:0]	command parameter	Sets the parameter contained in the current transmit command, this register is the high 16 bits of the parameter	R/W

35.3.5 Transmission Mode Register (TRANSMODE)

Offset address:

0x0C Reset

value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0		
Reserved										MULB	DDIR	ATCEN[1:0]		BCE	Rsvd		
classifier for honorific people	marking	celebrity	functionality								fill out or in (information on a form)						
b15~b6	Reserved	-	Read 0 when reading, please write 0 when writing								R						
b5	MULB	Multi-block	0: The current transmission is a single block transmission. 1: The current transmission is a multi block transmission. 10: Prohibited Settings								R/W						
b4	DDIR	Data transfer direction	0: Prohibited Settings 1: Write operation (host sends data) 0: Disable block count 1: Read operation (host receives data)								0: R/W						
b1 enable	BCE	Block Count Enable	0: Disable block count 1: Read operation (host receives data) 00: No automatic commands are sent 11: Allow data block								R/W						
b3~b2 b0	ATCEN Reserved	Auto command enable Automatically send CCR after reading, please write 0 for transmission	0: Count enable 1: Read operation (host receives data) 01: when writing R/W								01: when writing R/W						

35.3.6 Command Register (CMD)

Offset address:

0x0E Reset

value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved							IDX[5:0]		TYP[1:0]	DAT	ICE	CCE	Rsvd	RESTYP[1:0]	
<hr/>															
classifier for honorific people	marking	celebrity	functionality												fill out or in (informa tion on a form)
b15~b14	Reserved	-	Read 0 when reading, please write 0 when writing												R
b13~b8	IDX	Order No.	Send command number 00: Ordinary orders												R/W
b7~b6	TYP	Command Type	01: suspend command R/W 10: resume command 11: Abort command												
b5	DAT	Command with data	0: Current command uses only SDIOx_CMD (x=1~2) command line 1: The current command requires the use of the SDIOx_Dy (x=1~2) data line												R/W
b4	ICE	Number checking	0: Do not check the command number in the response. R/W 1: Check the command number in the response.												
b3 the response.		CCECRC check	0: Do not check the CRC check digit in 1: Check the CRC check digit in the response.												R/W
b2	Reserved	-	Read 0 when reading, please write 0 writing												when
b1~b0 Response Type	RESTYP		00: No answer to this command 01: The command has an answer length of 136 bits. 10: Command has an answer with a length of 48 bit 11: Command has a 48-bit response with a busy status.												R/W

35.3.7 Answer register 0 (RESP0)

Offset address:

0x10 Reset

value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
<hr/>															
classifier for honorific people	marking	celebrity	functionality												fill out or in (inform

b15~b0	RESP0[15:0]	response message	Bits 15 to 0 of the answer message
--------	-------------	------------------	------------------------------------

R

35.3.8 Response Register 1 (RESP1)

Offset address:

0x12 Reset

value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RESP1[15:0]															

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b15~b0	RESP1[15:0]	Answer message	Bits 31 to 16 of the answer message	R

35.3.9 Response Register 2 (RESP2)

Offset address:

0x14 Reset

value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RESP2[15:0]															

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b15~b0	RESP2[15:0]	response message	Bits 47~32 of the answer message	R

35.3.10 Answer register 3 (RESP3)

Offset address:

0x16 Reset

value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RESP3[15:0]															

Bit Flag	Bit Name	Function Read/Write	
b15~b0	RESP3[15:0]	Answer Message	Bits 63~48 of the answer message

35.3.11 Response Register 4 (RESP4)

Offset address:

0x18 Reset

value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RESP4 [15:0]															

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b15~b0	RESP4 [15:0]	response message	79~64 bits of the answer message	R

35.3.12 Response Register 5 (RESP5)

Offset address:

0x1A Reset

value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RESP5[15:0]															

Bit Flag	Bit Name Function Read/Write
b15~b0	RESP5[15:0] Answer Message Bits 95~80 of the answer message R

35.3.13 Response Register 6 (RESP6)

Offset address:

0x1C Reset

value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RESP6[15:0]															

Bit Flag	Bit Name Function Read/Write
b15~b0	RESP6[15:0] Answer Message Bits 111~96 of the answer message R

35.3.14 Response Register 7 (RESP7)

Offset address:

0x1E Reset

value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RESP7[15:0]															

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b15~b0	RESP7[15:0]	response message	Bits 127~112 of the answer message	R

35.3.15 Data buffer register 0 (BUF0)

Offset address:

0x20 Reset

value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
BUF0[15:0]															

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b15~b0	BUF0	data buffer	Write transmit data and read receive data, this register is the lower 16 bits of the data	R/W

35.3.16 Data Buffer Register 1 (BUF1)

Offset address:

0x22 Reset

value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
BUF1 [15:0]															

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b15~b0	BUF1	data buffer	Write transmit data and read receive data, this register is the high 16 bits of the data	R/W

35.3.17 Host Status Register (PSTAT)

Offset address: 0x24

Reset value: 0x00000000

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	
Reserved				CMDL	DATL[3:0]			WPL	CDL	CSS	CIN				
b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1
Reserved		BRE	BWE	RTA	WTA	Reserved			DA	CID	CIC				

b0

classifier for honorable people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b25	Reserved	-	Read 0 when reading, please write 0 when writing	R
B24	CMDL	Command Line Status	State of command line (SDIOx_CMD (x=1~2))	R
B23~b20	DATL	Cable Status	State of data line (SDIOx_Dy (x=1~2) (y=0~3))	R
B19	WPL	Write Protect Line Status	State of write-protect line (SDIOx_WP (x=1~2))	R
B18	CDL	Card Recognition Line Status	Status of card recognition line (SDIOx_CD (x=1~2))	R
B17	CSS	Device Stable Status	0: Card recognition line status unstable 1: Stable state of card recognition line, device inserted or not inserted	R
B16 inserted	CIN	Device insertion status	0: No device 1: With equipment inserted	0: No device
B15~b12	Reserved	R	-Read 0 when reading, please write 0	when writing
B11	BRE	Data buffer full	0: Data buffer does not have enough data for reading 1: Data buffer has enough data for reading	R
B10	BWE	Data buffer empty	0: Data buffer can write data 1: Data buffer cannot write data	R
B9	RTA	Read operation status read operation in progress	1: There is a read operation in progress	0: No R
B8	WTA	Write operation status write operation in progress		0: No R
B7~b3	Reserved	writing	1: There is a write operation in progress -Read 0 when reading, please write 0	when writing
B2	DA	Data line transmission status line idle	R 1: The data cable is transmitting data	0: Data

B1	CID	Command with data suppression sending commands with data	0: Allow 1: Prohibit sending commands with data	R
B0	CIC	Command inhibit	0: Commands allowed to be sent 1: Prohibit the sending of commands	R

35.3.18 Host Control Register (HOSTCON)

Offset address:

0x28 Reset

value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
CDSS	CDTL	EXDW		Reserved	HSEN	DW	Rsvd

Bit Flag	Bit Name	Function	Read/Write	
b7	CDSS	Card Recognition Line Selection to reflect the card recognition state		0: Select the real SDIO _x _CD (x=1~2) line 1: Select card recognition test signal to reflect card recognition status
b6	CDTL	Card recognition test signal status state	R/W	0: Card recognition test signal is low (device inserted) 1: Card recognition test signal is high (no device inserted)
b5	EXDW	Extended Data Bit Width Data line bit width using DW bit setting	R/W	0: 1: Data line bit width is 8 bits (8bit)
b4~b3	Reserved	-Read 0 when reading, please write 0 writing	R	when
b2	HSEN	High speed mode enable disable high speed mode	R/W	0: 1: Enable high speed mode
b1	DW	Data bit width selection Data line bit width is 1 bit (1bit)	R/W	0: 1: Data line bit width is 4 bits (4bit)
b0	Reserved	-Read 0 when reading, please write 0 writing	R	when

35.3.19 Power Control Register (PWRCON)

Offset address:

0x29 Reset

value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Reserved						PWON	
classifier for honorific people	marking	celebrity	functionality			fill out or in (informa tion on a form)	
b7~b1	Reserved	-	Read 0 when reading, please write 0 when writing			R	
b0	PWON	SDIOC Enable	0: Disable SDIOC 1: Enable SDIOC			R/W	

35.3.20 Data block gap control register (BLKGPCON)

Offset address:

0x2A Reset

value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
		Reserved		IABG	RWC	CR	SABGR
classifier for honorific people	marking	celebrity	functionality				fill out or in (information on a form)
b7~b4	Reserved	-	Read 0 when reading, please write 0 when writing				R
b3	IABG	Data block gap interrupt control Receive SDIO device interrupt (card interrupt) during data block gap shutdown. control R/W block gap is turned on.	0: 1: Receive SDIO device interrupt (card interrupt) during the data block gap is turned on.				0:
b2	RWC	Read wait control R/W	0: disable read wait function (read wait) 1: Enable read wait function (read wait)				
b1	CR	Continue transmission effect R/W	0: no 1: Release transmission stopped by SABGR position bit				0: no
b0	SABGR	Data block gap stop transmission	0: Stop transmission when not in data block gap 1: Stop transmission at data block gaps				R/W

35.3.21 Clock Control Register (CLKCON)

Offset address:

0x2C Reset

value: 0x0002

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
FS[7:0]								Reserved					CE	Rsvd	ICE
classifier for honorific people	marking	celebrity	functionality					fill out or in (information on a form)							

SDIO_x_CK (x=1~2) clock division selection, the reference
clock is EXCLK
0x80: 256 divisions of EXCLK
0x40: 128 divisions of EXCLK
0x20: 64 divisions of EXCLK
0x10: 32 divisions of EXCLK

0x04: 8 divisions of EXCLK

0x02: 4 divisions of EXCLK

0x01: 2 divisions of EXCLK

0x00: EXCLK

Others: Prohibited Settings

b2	CE	SDIOx_CK (x=1~2) lose output control	0: SDIOx_CK (x=1~2) stops the output 1: SDIOx_CK (x=1~2) outputs	R/W
b0	ICE	clock enable (computing)	0: SDIOC action clock on 1: SDIOC action clock off	R/W

35.3.22 Timeout Control Register (TOUTCON)

Offset address:

0x2E Reset

value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Reserved						DTO[3:0]	

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b7~b4	Reserved	-	Read 0 when reading, please write 0 when writing	R
b3~b0	DTO	Data Timeout	Set data line SDIOx_Dy (x=1~2) (y=0~7) The time for timeout determination in Clock period of EXCLK 0000: EXCLK x ²¹³ 0001: EXCLK x ²¹⁴ ----- 1110: EXCLK x ²²⁷ 1111: Prohibition of settings	R/W

35.3.23 Software Reset Register (SFTRST)

Offset address:

0x2F Reset

value: 0x00

b7	b6	b5	b4	b3	b2	b1	b0
Reserved						RSTD	RSTC

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b7~b3	Reserved	-	Read 0 when reading, please write 0 when writing	R
b2	RSTD	data reset	Reset all data-related registers containing the following register bits: BUF0, BUF1 pstat.bre, pstat.bwe, pstat.rta, pstat.wta. PSTAT.DLA, PSTAT.CID BLKGPCON.CR, BLKGPCON.SABGR NORINTST.BRR, NORINTST.BWR, NORINTST.BGE. NORINTST.TC 0: Normal operation	R/W

			1: Perform a reset	
			Resets all command-related registers and contains the following register bits: PSTAT.CIC	
b1	RSTC	Command reset	NORINTST.CC	R/W
			0: Normal operation 1: Perform a reset	
			1: Perform a reset	
			Reset all SDIOC registers except the card recognition function	
b0	RSTA	Full reset	0: Normal operation 1: Perform a reset	R/W

35.3.24 Normal Interrupt Status Register (NORINTST)

Offset address:

0x30 Reset

value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
EI		Reserved					CINT	CRM	CIST	BRR	BWR	Rsvd	BGE	TC	CC

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b15	EI	false interruption	Set when any error occurs in the error interrupt status register (ERRINTST).	R
b14~b9	Reserved	-	Read 0 when reading, please write 0 when writing	R
b8 revoked by the SDIO device.	CINT	card interrupt ▷	Set when the SDIO device issues a card interrupt request, and Post-application reset	
b7	CRM	card removal	Reset on device removal, reset on write 1	R/W
b6	CIST	Card Insertion	Set when device is inserted, reset on write 1	R/W
b5	BRR	Buffer readable	Set when data in the buffer can be read (PSTAT.BRE=1), write 1 reset	R/W
b4	BWR	Buffer writable	Set when buffer can write data (PSTAT.BWE=1), write 1 reset	R/W
b3	Reserved	-	Read 0 when reading, please write 0 when writing	R
b2	BGE	Data block gap stop transmission when transmission stops at data block gap, write 1 reset enter(a password)		Set
b1	TC	Transmission complete	Set on completion of read/write transfer, reset on write 1	R/W
b0	CC	Command completion	Bit set after completion of transmission of no-answer command and reception of answer with answer command, reset by writing 1	R/W

35.3.25 Error Interrupt Status Register (ERRINTST)

Offset address:

0x32 Reset

value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
		Reserved					ACE	Rsvd	DEBE	DCE	DTOE	CIE	CEBE	CCE	CTOE

classifier for honorific	marking	celebrity	functionality	fill out or in
-----------------------------	---------	-----------	---------------	-------------------

people			(information on a form)
b15~b9	Reserved	-	Read 0 when reading, please write 0 when writing R
b8	ACE when an error occurs in the auto CMD command, and the type of error can be queried in the auto CMD command.	Auto send command error Command Error Register (ATCERRST), Write 1 Reset	Set R/W
b7	Reserved	-	Read 0 when reading, please write 0 when writing R
b6	DEBE	Data stop bit error Set when the data line detects a low level in the stop bit, write 1 reset	R/W
b5	DCE	Data CRC checksum error Set when a CRC check error occurs on the data line, write 1 reset	R/W
b4	DTOE by the timeout control registers	Data Timeout Error Set when a data timeout occurs, write 1 to reset, data timeout time is determined by (TOUTCON) setting	R/W
b3	CIE	Command number error Set when the command number contained in the received answer is incorrect, reset by writing 1	R/W
b2	CEBE	Command stop bit error Set when the command line detects a low level in the stop bit, write 1 reset	R/W
b1	CCE	Command CRC Checksum Error Set when a CRC check error occurs on the command line, reset by writing 1	R/W
b0	CTOE SDIOx_CK (x=1~2) cycles after the command is sent.	Command Timeout Error Set when no answer is received for more than 64 Write 1 reset	R/W

35.3.26 Normal Interrupt Status Enable Register (NORINTSTEN)

Offset address:

0x34 Reset

value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Reserved						CINTEN	CRMEN	CISTEN	BRREN	BWREN	Rsvd	BGEEN	TCEN	CCEN	

classifier for honorific people	marking	celebrity	functionality	fill out or in (informa tion on a form)
b15~b9	Reserved	-	Read 0 when reading, please write 0 when writing	R
b8	CINTEN	Card Interrupt Status Enable Disable NORINTST.CINT setup	0: 1: Allow NORINTST.CINT to be reset	R
b7	CRMEN	Card Removal Status Enable Disable NORINTST.	0: 1: Allow NORINTST. R/W	
b6	CISTEN	Card insertion status enable Disable NORINTST.CIST setting	0: 1: Allow NORINTST.CIST to be reset	R/W
b5	BRREN	Buffer readable state enable	0: Disable NORINTST.BRR setting	R/W
b4	BWREN	Buffer Writable Status Enable	1: Allow NORINTST.BWR to be set 0: Disable NORINTST.BWR setting 1: Allow NORINTST.BWR to be set	R/W
b3	Reserved R		-Read 0 when reading, please write 0	when wrt
b2	BGENN	Data block gap stop passing Disable NORINTST.BGE Setting transmission status enable R/W Allow NORINTST.	0: 1:	
b1	TCEN	Transmission completion status enable Disable NORINTST.	0: 1: Allow NORINTST. R/W	
b0	CCEN	Command completion status enable Disable NORINTST.	0: 1: Allow NORINTST. R/W	

35.3.27 Error Interrupt Status Enable Register (ERRINTSTEN)

Offset address:

0x36 Reset

value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0		
							Reserved		ACEEN	Rsvd	DEBEEN	DCEEN	DTOEEN	CIEEN	CEBEEN	CCEEN	CTOEEN

classifier for honorific people	marking	celebrity	functionality	fill out or in (informa tion on a form)
---------------------------------------	---------	-----------	---------------	---

b15~b9	Reserved	-	Read 0 when reading, please write 0 when writing	R
b8	ACEEN Disable ERRINTST.ACE Setting	Auto Send Command Error <small>D/A</small> stateful enable (computing)	1: Allow ERRINTST.ACE to be set	0:
b7	Reserved	-	Read 0 when reading, please write 0 when writing	R
b6	DEBEEN	Data stop bit error status enable	0: Disable ERRINTST. 1: Allow ERRINTST.	R/W
b5	DCEEN	Data CRC Checksum Error Status Enable	0: Disable ERRINTST. 1: Allow ERRINTST.	R/W
b4	DTOEEN	Data Timeout Error Status Enable	0: disable ERRINTST.DTOE 1: Allow ERRINTST.DTOE to be reset	R /
b3	CIEEN	Command Number Error Status Enable	0: Disable ERRINTST. 1: Allow ERRINTST.	W
b2	CEBEEN	Command stop bit error state enable	0: Mask ERRINTST.CEBE set bit	R/W
b1	CCEEN	Command CRC Checksum Error Status Enable	1: Allow ERRINTST.CCE to be set 0: Disable ERRINTST.CCE	R/W
b0	CTOEEN	Command timeout error status enable	0: Disable ERRINTST.CTOE setting 1: Allow ERRINTST.CTOE to be set	R/W

35.3.28 Normal Interrupt Signal Enable Register (NORINTSGEN)

Offset address:

0x38 Reset

value: 0x0000

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

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b15~b9	Reserved	-	Read 0 when reading, please write 0 when writing	R
b8	CINTSEN	Card interrupt signal enable 0: Disable NORINTST. 1: Allow NORINTST. R		
b7	CRMSEN	Card Removal Signal Enable Prohibit NORINTST.CRM from requesting an interrupt. 1: Allow NORINTST.CRM to request an interrupt. R/W	0:	
b6	CISTSEN	Card insertion signal enable Disable NORINTST.CIST application interrupt 1: Allow NORINTST.CIST to request an interrupt	0: R/W	
b5	BRRSEN	Buffer readable signal enable Disable NORINTST.BRR application interrupt enable 1: Allow NORINTST.BRR to apply interrupt R/W	0: R/W	
b4	BWRSEN	Buffer writable signal enable disable NORINTST.BWR apply interrupt (usually used in the negative) have the possibility of D/A	0: 1: Allow NORINTST.BWR to request an interrupt	
b3	Reserved	-	Read 0 when reading, please write 0 when writing	R
b2	BGESEN	Data block gap stop passing Disable NORINTST.BGE application interrupt transmission signal enable R/W Allow NORINTST.	0: 1: R/W	
b1	TCSEN	Transmission completion signal enable Disable NORINTST.TC application interrupt 1: Allow NORINTST.TC to request an interrupt	0: R/W	
b0	CCSEN	Command completion signal enable NORINTST.CC is not allowed to apply interrupt. 1: Allow NORINTST. R/W	0: R/W	

35.3.29 Error Interrupt Signal Enable Register (ERRINTSGEN)

Offset address:

0x3A Reset

value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
						Reserved	ACEESN	Rsvd	DEBESEN	DCESEN	DTOESEN	CIESEN	CEBESEN	CCESEN	CTOESEN

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b15~b9	Reserved	-	Read 0 when reading, please write 0 when writing	R
b8	ACESEN Disable ERRINTST.ACE Request Interrupt	Auto Send Command Error signal enable	0: 1: Allow ERRINTST.ACE to request an interrupt	0: R/W
b7	Reserved	-	Read 0 when reading, please write 0 when writing	R
b6	DEBESEN	Data stop bit error signal enable	0: Disable ERRINTST. 1: Allow ERRINTST.DEBE to request an interrupt	R/W
b5	DCESEN	Data CRC check error signal enable	0: Disable ERRINTST. 1: Allow ERRINTST.DCE to request an interrupt	R/W
b4	DTOESEN	Data timeout error signal enable	0: Disable ERRINTST. 1: Allow ERRINTST.DTOE to request an interrupt	R /
b3	CIESEN	Command number error signal enable	0: Prohibit ERRINTST.CIE from requesting an interrupt	W
b2	CEBESEN	Command stop bit error signal enable	1: Allow ERRINTST.CIE to request an interrupt	R/W
b1	CCESEN	Command CRC checksum error signal enable	0: Mask ERRINTST.CEBE application interrupt 1: Allow ERRINTST.CEBE to apply for an interrupt	R/W
b0	CTOESEN	Command timeout error signal enable	0: Disable ERRINTST.CCE request for interrupts 1: Allow ERRINTST.CCE to request an interrupt 0: Disable ERRINTST.CTOE application interrupt 1: Allow ERRINTST.CTOE to request an interrupt	R/W

35.3.30 Auto Command Error Status Register (ATCERRST)

Offset address:

0x3C Reset

value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
								CMDE	Reserved	IE	EBC	CE	TOE	NE	

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b15~b8	Reserved	-	Read 0 when reading, please write 0 when writing	R
b7	CMDE	Unsent error	Set when an error occurs in bits b4~b0 of this register causing other commands not to be sent.	R
b6~b5	Reserved	-	Read 0 when reading, please write 0 when writing	R
b4	IE	Command number error	Set when the number of the auto command contained in the received answer is wrong	R
b3	EBC	stop bit error	Set when the command line detects a low level in the stop bit	R
b2	CE	Data Timeout Error	Set when a CRC checksum error occurs on the command line	R
b1	TOE	Command timeout error after the auto command is sent.	Set when no answer is received for more than 64 SDIOx_CK (x=1~2) cycles	R
b0	NE	Unexecuted error	Set when an automatic command is not sent for other reasons.	R

35.3.31 Forced Auto Command Error Status Control Register (FEA)

Offset address:

0x50 Reset

value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
								FCMDE	Reserved	FIE	FEBC	FCE	FTOE	FNE	

classifier for honorific people	marking	Bit Name Function Read/Write	
b15~b8	Reserved	-Read 0 when reading, please write 0 writing R	when 0: no effect
b7	FCMDE	Forced not sent error 1: Forcing an ATCERRST.CMDE error	W
b6~b5	Reserved	-Read 0 when reading, please write 0 writing W	when 0: no effect
b4	FIE	Mandatory command numbering error effect W 1: Forcing an ATCERRST.IE error	1374

		0: no effect	
b3	FEBE	Forced stop bit error 0: no effect 1: Forcing an ATCERRST.EBE error	W
b2	FCE	Forced data timeout error effect W 0: no 1: Forcing an ATCERRST.CE error	
b1	FTOE	Force command timeout error effect W 0: no 1: Forcing an ATCERRST.TOE error	
b0	FNE	Forced not implemented error effect W 0: no 1: Forced ATCERRST.NE error	

35.3.32 Forced Error Status Control Register (FEE)

Offset address:

0x52 Reset

value: 0x0000

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
							Reserved	FACE	Rsvd	FDEBE	FDCE	FDTOE	FCIE	FCEBE	FCCE	FCTOE

classifier for honorific people	marking	celebrity	functionality	fill out or in (informa tion on a form)
b15~b9	Reserved	-	Read 0 when reading, please write 0 when writing	R
b8	FACE	Force auto send command error	0: No effect <u>1: Forcing an ERRINTST.ACE Error</u>	W
b7	Reserved	-	-Read 0 when reading, please write 0	when writing
b6	FDEBE	Forced data stop bit error	0: No effect <u>1: Forcing an ERRINTST.DEBE Error</u>	W
b5	FDCE	Force data CRC checksum effect	0: no W 1: ERRINTST.DCE error is forced to occur	0: no
b4	FDTOE	Forced data timeout error effect	0: no W <u>1: Forcing an ERRINTST.DTOE error</u>	0: no
b3	FCIE	Forced command stop bit error	0: No effect <u>1: Forcing an ERRINTST.CIE Error</u>	W
b1	FCCE	Force Command CRC Checksum Error	0: No effect <u>1: Forcing an ERRINTST.CCE Error</u>	W
b0	FCTOE	Force command timeout error	<u>1: Forcing an ERRINTST.CTOE error</u>	0: no effect

35.3.33 MMC Mode Enable Register (MMCER)

Address:

0x40055404 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															
classifier for honorific people	marking		celebrity		functionality										fill out or in (information on a form)
b31~b4	Reserved		-		Reads "0" and writes "0".										R
b3	SELMMC2		SDIOC2 MMC mode enable (sb. to do sth)		0: SDIOC2 selects SD mode 1: SDIOC2 selects MMC mode										R/W
b2	Reserved		-		Reads "0" and writes "0".										R
b1	SELMMC1		SDIOC1 MMC mode enable		0: SDIOC1 selects SD mode 1: SDIOC1 selects MMC mode										R/W
b0	Reserved		-		Reads "0" and writes "0".										R

36 Debugging Controller (DBG)

This product is referenced in the following ARM technical documentation:

- Cortex™-M4F r0p1 Technical Reference Manual (TRM)
- ARM Debug Interface V5
- ARM CoreSight Design Suite Version r0p1 Technical Reference Manual

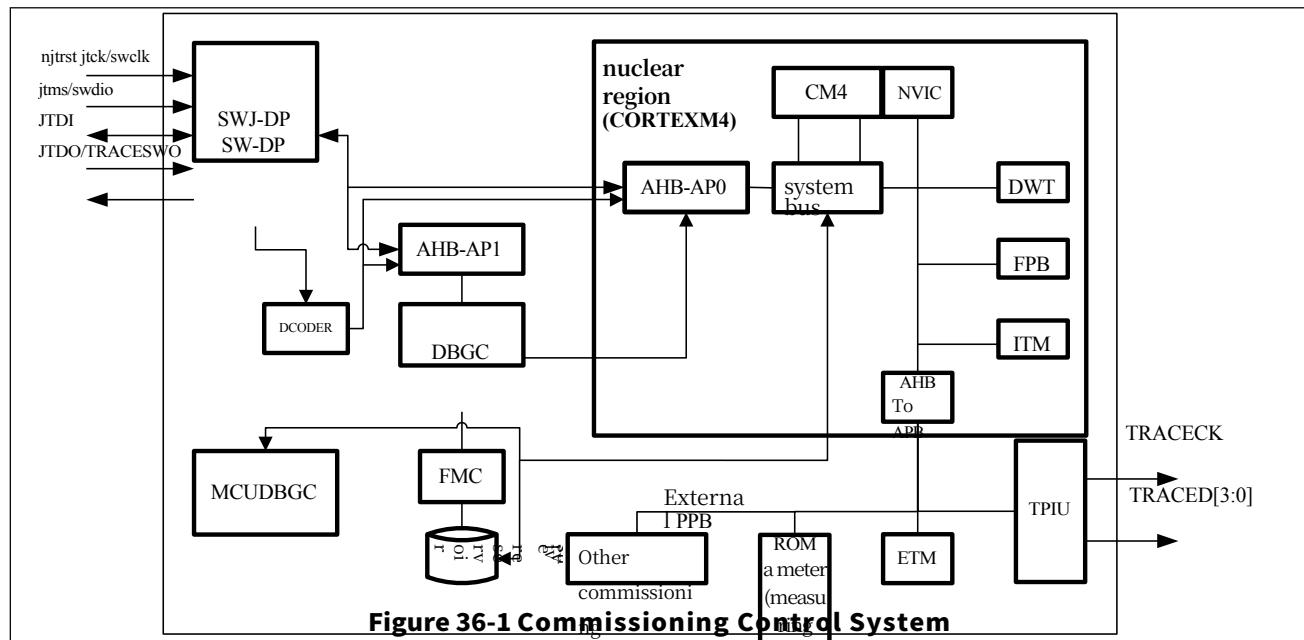
36.1 summary

The core of this MCU is a Cortex™-M4F, which contains hardware for advanced debugging functions. With these debugging features, it is possible to stop the core while fetching instructions (instruction breakpoints) or fetching access data (data breakpoints). When the kernel is stopped, the internal state of the kernel and the external state of the system can be queried. When the query is completed, the kernel and system will be restored and program execution will resume. This MCU is not equipped with an ETM debug device.

Two debugging interfaces are provided:

- Serial Debug Trace Interface SWD
- Parallel Debug Trace Interface JTAG

36.2 DBG System Block Diagram



The ARM Cortex™-M4F core provides integrated on-chip debugging support. It includes:

- SWJ-DP: SWD/JTAG Debug Port
- AHP-AP: AHB Access Port
- ITM: Instruction Tracking Unit
- FPB: Flash Instruction Breakpoint
- DWT: Data Breakpoint Trigger
- TPIU: Trace Port Unit Interface (available on large packages where the appropriate pins will be mapped)
- Flexible debug pin assignment

Attention:

- For more information on the debugging features supported by the ARM Cortex™-M4F core support debugging features, see the Cortex™-M4F- r0p1 Technical Reference Manual and CoreSight Design Suite r0p1 Technical Reference Manual.

36.3 SWJ-DP Debug Port (SWD and JTAG)

The MCU core integrates a SWD/JTAG debug port (SWJ-DP). This port is an ARM standard CoreSight debug port with a JTAG-DP (5-pin) interface and a SW-DP (2-pin) interface.

- The JTAG Debug Port (JTAG-DP) provides a 5-pin standard JTAG interface for connection to the AHP-AP port.
- The Serial Wire Debug Port (SW-DP) provides a 2-pin (clock + data) interface for connection to the AHP-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 multiplexes TRACESWO and TDO, which means that asynchronous tracking can only be realized on SW-DP, not on JTAG-DP.

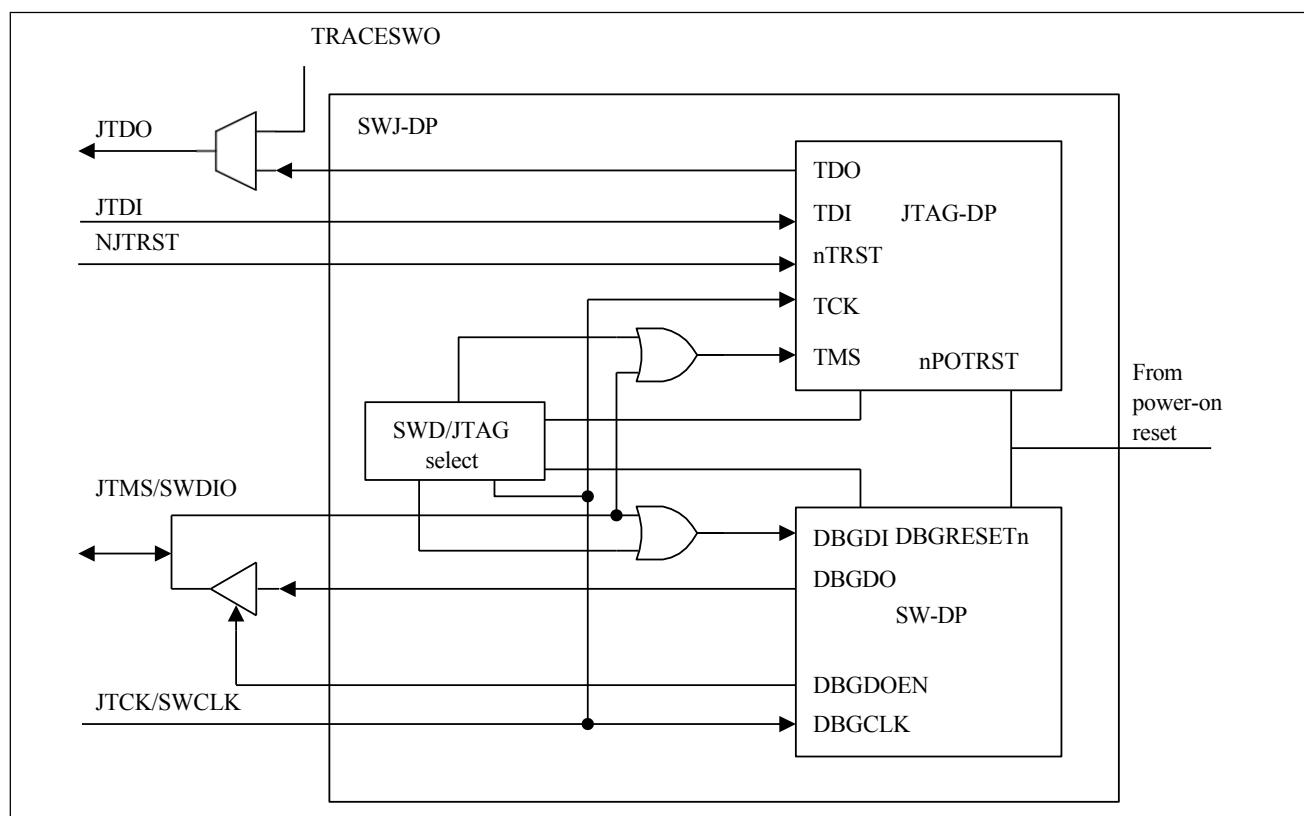


Figure 36-2 Commissioning Control System

36.3.1 Switching mechanism for JTAG-DP or SW-DP

The default debug interface is the JTAG-DP interface.

If the debug tool wants to switch to SW-DP, it must provide a dedicated JTAG sequence on **JTMS(SWDIO)/JTCK(SWCLK)** to disable JTAG-DP and enable SW-DP so that SW-DP can be accessed using only the SWCLK and SWDIO pins.

The sequence is:

1. Outputs JTMS (SWDIO) =1 signal for more than 50 JTCK cycles
2. Output 16 JTMS (SWDIO) signals 0111_1001_1110_0111 (MSB)
3. Outputs JTMS (SWDIO) =1 signal for more than 50 JTCK cycles

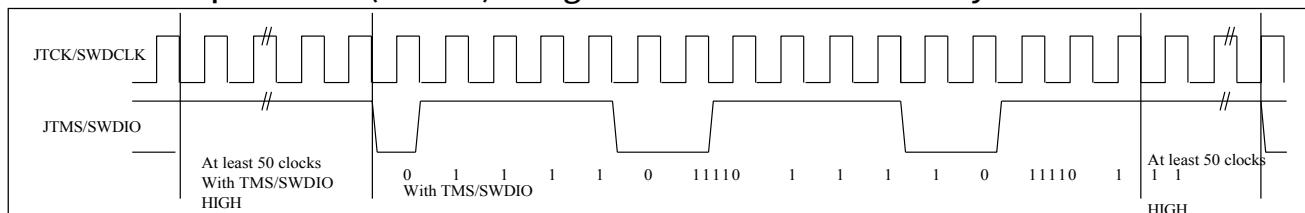


Figure 36-3 JTAG-DP to SW-DP Switching Sequence

36.4 Pinouts and Debug Port Pins

Depending on the package of the MCU there are different numbers of active pins. Therefore, certain pin-related functions may vary with the package.

36.4.1 SWJ Debug Port Pin

The five common I/O ports of the MCU can be used as SWJ-DP interface pins.

Table 36-1 SWJ Debug Port Pins

SWJ-DP Pin Name	JTAG debug port		SW debug port	
	typology	clarification	typology	Debugging Assignment
JTMS/SWDIO	I	JTAG mode selection	I/O	Serial Line 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 tracking is enabled)
NJTRST	I	JTAG Reset	-	-

36.4.2 Flexible SWJ-DP pin assignment

A reset (power-up or pin reset) assigns all 5 pins used for SWJ-DP 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 up the associated pins for use as GPIOs. for more detailed information on how to disable the pins of the SWJ-DP ports, see: General Purpose IO Special Control Registers PSPCR.

Table 36-2 Flexible SWJ-DP Pin Assignments

Available debug ports	Assigned SWJ IO pins				
	JTMS/ SWDIO	JTCK/ SWCLK	JTDI	JTDO	NJTRST
All SWJ (JTAG-DP+SW-DP) - reset state	✓	✓	✓	✓	✓
Disable JTAG-DP and Enable SW-DP	✓	✓	releasable	releasable	releasable
Disable JTGA-DP and Disable SW-DP	releasable	releasable	releasable	releasable	releasable

36.4.3 Internal pull-ups on JTAG pins

According to the JTAG IEEE standard, it is important to ensure that the JTAG input pins are not left dangling as these pins are directly connected to the MCU internally for controlling the debug functions. Special attention must also be paid to the JTCK/SWCLK pins, which are used directly for debug control clock functions. To avoid floating IO levels, the MCU has built-in internal pull-up resistors on the JTAG pins:

- NJTRST: Internal pull-up
- JTDI: internal pull-up
- JTMS/SWDIO: internal pull-ups
- JTCK/SWCLK: internal pull-ups
- JTDO: high resistance state

Without the debugger connected, the user software can release the JTAG IO to be used as a normal I/O port by setting the GPIO special control registers. Since the internal pull-up is a weak pull-up of <100K ohms, it is recommended to use an external pull-up of 10K ohms.

36.4.4 Using the Serial Interface and Releasing Unused Debug Pins for GPIOs

Some GPIOs can be freed up when using SWD, and the user software must change the GPIO configuration in the GPIO control registers so that the corresponding pins can be freed up for use as GPIOs.

During debugging, the host computer performs the following operations:

- Assign all SWJ pins (JTAG-DP+SW-DP) in the system reset state.
- In the 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 kernel stops at the reset address.
- From here the debug port is switched to SW-DP. and then the other JTAG pins are reassigned as GPIOs via user software.

Attention:

- For user software designs that require debug pins to be released, these pins remain on input pull-ups (NJTRST, JTMS, JTDI, JTCK, and JTDO) after reset until the user software releases the pins.

36.5 processor register

The registers are described below:

Base address: 0xE004_2000

register name	notation	offset address	bit width	starting value	Access to the host
DBG Status Register	MCUDBGSTAT	0x001C	32	0x00000000	CPU/debug IDE*
Peripheral Debug Pause Register	MCUSTPCTL	0x0020	32	0x00000003	CPU/debug IDE*
TRACE Port Control Register	MCUTRACECTL	0x0024	32	0x00000000	CPU/debug IDE*

Attention:

-The registers are located in the PPB area. The registers are located in the PPB area and can only be accessed by the CPU in privileged mode.

36.5.1 DBG Status Register (MCUDBGSTAT)

DBG Debug power-up status

acknowledge register. Reset

value: 0x0000_0001

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
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

-	-	-	-	-	-	-	-	-	-	-	-	-	-	CDB	CDB
														G	G
														PWR	PWR
														UPA	UPR
														CK	EQ

classifier for honorific people	marking	celebrity	functionality	fill out or in (information on a form)
b31~b2	Reserved	-	Reads "0", writes "0".	R/W
b1	CDBGPRUPACK	Debugger power-up feedback	0: no feedback 1: Debugging power-up feedback	R/W
b0	CDBGPRUPREQ	Debugger power-up request request	R/W 1: Power-up request	0: No power-up

36.5.2 Peripheral Debug Pause Register (MCUSTPCTL)

The peripheral module suspends control

when the CPU is in the debug state. Reset

value: 0x0000_0003

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
TMR	-	-	-	-											
A6	A5	A4	A3	A2	A1	63	62	61	43	42	41	-	-	-	-
STP	-	-	-	-											
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TMR	TMR	-	-	-	-	-	-	-	-	-	-	RTC	WDT	SWD	
02	01	-	-	-	-	-	-	-	-	-	-	STP	STP	TST	P

classifier for honorific people	marking	Bit Name Function Read/Write	
b31	TMRA6STP	TimerA-6 count pause signal 0: Counter still counts even if the kernel is stopped 1: Counter pauses counting when kernel stops	R/W
b30	TMRA5STP	TimerA-5 count pause signal 0: Counter still counts even if the kernel is stopped 1: Counter pauses counting when kernel stops	R/W
b29	TMRA4STP	TimerA-4 count pause signal 0: Counter still counts even if the kernel is stopped 1: Counter pauses counting when kernel stops	R/W
b28	TMRA3STP	TimerA-3 count pause signal 0: Counter still counts even if the kernel is stopped 1: Counter pauses counting when kernel stops	R/W
b27	TMRA2STP	TimerA-2 count pause signal 0: counter still counts even if kernel stops 1: Counter pauses counting when kernel stops	R/W
b26	TMRA1STP	TimerA-1 count pause signal 0: counter still counts even if the kernel stops 1: Counter pauses counting when kernel stops	R/W
b25	TMR63STP	Timer6-3 count pause signal 0: counter still counts even if kernel stops 1: Counter pauses counting when kernel stops	R/W
b24	TMR62STP	Timer6-2 count pause signal 0: counter still counts even if kernel stops 1: Counter pauses counting when kernel stops	R/W
b23	TMR61STP	Timer6-1 count pause signal 0: counter still counts even if the kernel stops 1: Counter pauses counting when kernel stops	R/W
b22	TMR43STP	Timer4-3 count pause signal 0: counter still counts even if kernel stops 1: Counter pauses counting when kernel stops	R/W
b21	TMR42STP	Timer4-2 count pause signal 0: counter still counts even if kernel stops 1: Counter pauses counting when kernel stops	R/W
b20	TMR41STP	Timer4-1 count pause signal 0: counter still counts even if the kernel stops 1: Counter pauses counting when kernel stops	R/W
b19	Reserved	-0" for reading, "0" for writing. writes "0".	Reads "0", R/W
b18	Reserved	-0" for reading, "0" for writing. writes "0".	Reads "0", R/W
b17	Reserved	-0" for reading, "0" for writing. writes "0".	Reads "0", R/W
b16	Reserved	-0" for reading, "0" for writing.	Reads "0",

		writes "0".	R/W	
b15	TMR02STP	Timer0-2 count pause signal 0: counter still counts even if the kernel stops 1: Counter pauses counting when kernel stops		R/W
b14	TMR01STP	Timer0-1 count pause signal 0: counter still counts even if the kernel stops 1: Counter pauses counting when kernel stops		R/W

b13~b3	Reserved	-0" for reading, "0" for writing. writes "0".	R/W	Reads "0",
b2	RTCSTP	RTC count pause signal	0: RTC counter still counts even if the kernel is stopped 1: RTC counter pauses counting when the kernel stops	R/W
b1	WDTSTP	WDT count pause signal	0: WDT counter still counts even if the kernel stops 1: WDT counter pauses when kernel stops. R/W	
b0	SWDTSTP	SWDT count pause signal	0: SWDT counter still counts even if the kernel is stopped 1: SWDT counter pauses counting when the kernel stops	R/W

36.5.3 Debug Component Configuration Register (MCUTRACECTL)

Configure the TRACE output pin

with 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
-	-	-	-	-	-	-	-	-	-	-	-	-	-	TRA CEI OEN	TRACEMODE

classifier	for marking	celebrity	functionality	fill out or in (information on a form)
honorific				
people				
b31~b3	Reserved	-	Reads "0", writes "0".	R/W
b2	TRACEIOEN	TRACE pin output control	0: Synchronous tracking pin output disabled 1: Synchronous tracking pin output license	R/W
b1~b0	TRACEMODE	TRACEDATA output pin control	00: Asynchronous tracking 01: Synchronized tracking 2-bit TRACEDATA[1:0] 01: Synchronized tracking of 1-bit TRACEDATA[0] 11: Synchronized tracking 4-bit TRACEDATA[3:0]	R/W

36.6 SW debug port

36.6.1 Introduction to the SW Protocol

The synchronous serial protocol uses two pins:

- SWCLK: Clock from host to slave
- SWDIO: Bidirectional

When transmitting data, the LSB comes first.

For SWCLK and SWDIO, a pull-up of the line on the board is required (10K ohms recommended)

36.7 TPIU (Trace Port Interface Unit)

36.7.1 summary

The TPIU is a bridge between the ITM and the on-chip trace data.

The output data stream is encapsulated as a trace source ID and then captured by the Trace Port Analyzer (TPA).

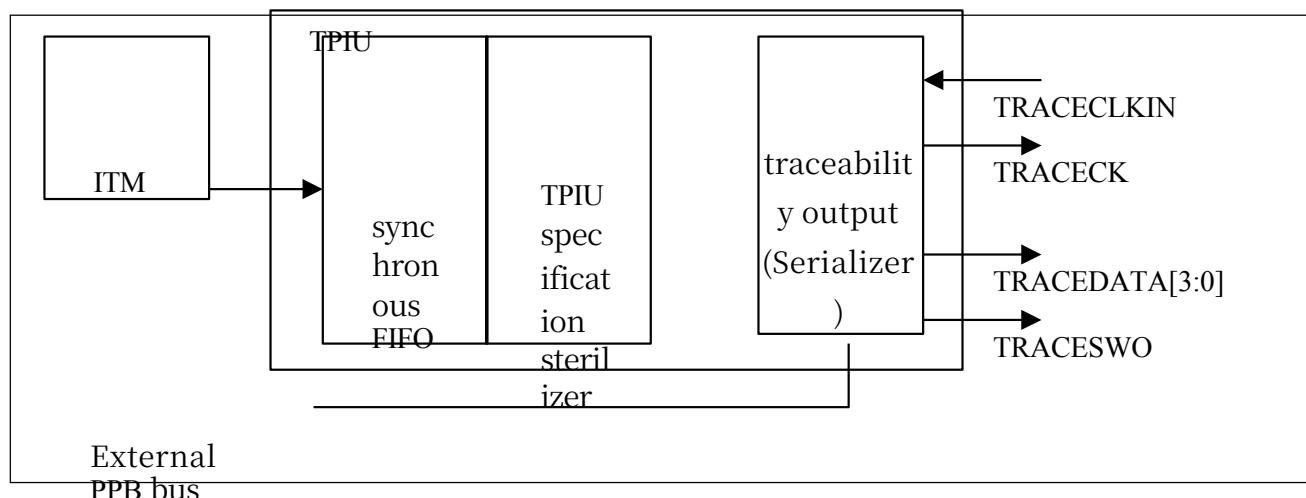


Figure 36-4 TPIU Block Diagram

36.7.2 TRACE Pin Assignment

- asynchronous mode

Asynchronous mode requires 1 extra pin and is available in all packages. Asynchronous mode is only available when using serial line mode (not available in JTAG mode)

TPIU Pin Name	Tracking Asynchronous Mode	
	typology	clarification
TRACESWO	o	TRACE asynchronous data output

- synchronous mode

Synchronous mode requires 2 to 5 additional pins, depending on the length of the trace data, and is only available in larger packages. Additionally, synchronous mode is available in both JTAG and serial modes and provides a higher bandwidth output capability than asynchronous traces.

TPIU Pin Name	Tracking Synchronization Mode	
	typology	clarification
TRACECK	o	TRACE Clock
TRACED[3:0]	o	TRACE synchronized data output, which can be 1, 2 or 4.

TPIU TRACE Pin Assignment

By default, these pins are not assigned. These pins can be configured by placing 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.

In addition, the number of pins to be assigned depends on the trace configuration (asynchronous trace or synchronous trace)

- Asynchronous mode: 1 extra pin required
- Synchronized mode: 5 additional pins required
 - TRACECK
 - TRACED[0] (if port data length is configured as 1, 2, or 4)
 - TRACED[1] (if port data length is configured as 2 or 4)
 - TRACED[2] (if port data length is configured 4)
 - TRACED[3] (if port data length is configured 4)

To assign the TRACE pin, the debug host must program bits TRACE_IOEN and TRACE_MODE[1:0] of the MCU Debug Configuration Register (MCUTRACECTL). The TRACE pin is not assigned by default.

This register is mapped to the external PPB bus and is reset by power-up (not pin reset). This register can be written via the debugger in the pin reset state.

TPIU Pin Usage	Assigned TRACE IO pins					
	JTDO/ TRACESWO	TRACECK	TRACED[0]	TRACED[1]	TRACED[2]	TRACED[3]
untracked (default state) TRACE_IOEN =0 TRACE_MODE=XX	Release*	liberate (a prisoner)				
Asynchronous tracking TRACE_IOEN =1 TRACE_MODE=00	TRACESWO	liberate (a prisoner)				
Synchronization trace 1 bit TRACE_IOEN =1 TRACE_MODE=01	Release*	TRACECK	TRACED[0]	liberate (a prisoner)	liberate (a prisoner)	liberate (a prisoner)
Synchronized trace 2 bits TRACE_IOEN =1 TRACE_MODE=10	Release*	TRACECK	TRACED[0]	TRACED[1]	liberate (a prisoner)	liberate (a prisoner)
Synchronized trace 4 bits TRACE_IOEN =1 TRACE_MODE=11	Release*	TRACECK	TRACED[0]	TRACED[1]	TRACED[2]	TRACED[3]

Attention:

- When using serial mode, this pin is released. However, when using JTAG, this pin is assigned to TDO.

36.7.3 MCU Internal TRACECLKIN Connection

In this MCU, the clock TRACECLKIN of TPIU is connected to the internal clock. The default clock of MCU is the internal MRC oscillator. The frequency in the reset state is different from the frequency after the reset is released. The reason for this is because the default MRC calibration value is used in the system reset state and is updated each 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 synchronized frame packet in the reset state is different from the packet after the reset.

36.7.4 TPIU Register

Read and write operations to the TPIU APB registers can be performed only when bit TRCENA of the Debug Exception and Monitor Control Register (DEMCR) is set to one. Otherwise, these registers will read zero (the output of this bit enables the TPIU clock).

36.7.5 TPIU Configuration Example

- Setting bit TRCENA in the Debug Exception and Monitor Control Register (DEMCR) to 1
- Write the desired value to the TPIU current port size register (default value is 0x1 for 1-bit port sizes)
- Write 0x102 to TPIU formatter and refresh control registers (default)
- Write TPIU Select pin protocol to select synchronous or asynchronous mode. Example: 0x2 indicates asynchronous NRZ mode (similar to USART)
- Write 0x20 to the MCUTRACECTL control register (bit IO_TRACEN) to allocate TRACE I/O for asynchronous mode.
- Send TPIU synchronization packet at this time (FF_FF_FF_7F)
- Configure ITM and write to ITM excitation register to output value

Version Revision Record

version number	revision date	revision
Rev1.0	2019/11/12	First Edition Release.
Rev1.1	2020/01/10	<ul style="list-style-type: none">1) Add 256KB of product descriptions to the full text;2) VFBGA package description added throughout;3) Initialization Configuration (ICG) describes pen mis-modification;4) The control register (WDT_CR) was modified by pen error;5) Table 34-1 CRC_RESTLT modified to CRC_RESLT;6) Modify the current max value for 105°C in power-down mode in the Electrical Characteristics.

Rev1.2	2020/08/26	<p>1) Added description of ultra-high-speed operation mode, updated CoreMark/DMIPS, added switching flow between ultra-high-speed analog and high-speed mode, ultra-low-speed mode, added BOR/PVD characteristics in ultra-high-speed mode, current characteristics. Updated the wait cycle of SRAM and Flash at 200Mhz, and the wait cycle when reading the port. Updated the frequency values in the bus architecture, and updated the functional block diagram;</p> <p>2) The pinout diagram adds 256KB models;</p> <p>3) Add AOS section to add common trigger source enable bits for each AOS target register;</p> <p>4) Figure 6.1 TCK_SWCLK modified to JTCK_SWCLK and TCK modified to JTCK;</p> <p>6.10.1 Fix typos;</p> <p>6.11.7 Misremembered modification of bit1-0 of CMU_XTAL32NFR;</p> <p>6.11.13 "Frequency calibration needs to be within the LRC frequency guarantee" was corrected;</p> <p>6.11.15 Misremembered modification of bit27-24 of CMU_PLLCFG;</p> <p>5) 7.4.3 Add the description of the AD function enable bit corresponding to PWR_FCG3 before entering STOP mode to reduce power consumption;</p> <p>6) 7.7 Updating PWR_PWRC3/PWR_XTAL32CS/PWR_STPMCR bit value; remove the underscore of the WKE_n_m bit in PWR_PDWKE0/PWR_PDWKE1; remove the XTAL32 stop wake-up related bit in PWR_PDWKF1/PWR_PDWKE2; and the function descriptions of PWR_FCG0/PWR_FCG1/PWR_FCG2 are more detailed;</p> <p>7.7.13 Harmonization of PWR_FCG register, 10.2</p> <p style="text-align: right;">SRAM register Ret-</p> <p>SRAM/SRAMHS/SRAMECC labeling;</p> <p>Unifies the way reserved bits are described in Power Control (PWC);</p> <p>Update PVD2DETFLG and PVD1DETFLG bits in PWR_PVDDSR register</p> <p>The clearance instructions for the</p>
--------	------------	--

		7) 9.6.3 Modify step 10;
--	--	--------------------------

version number	revision date	revision
		<p>9.9.2 b1 FSLP modified to FSTP;</p> <p>9.9.3 b24 CRST0 Revised to CRST;</p> <p>9 Overview Erase is modified to Sector Erase;</p> <p>9.9.7RDCOLERRITE modified to COLERRITE and read conflict modified to read-write conflict;</p> <p>8) 12.3 Modify USARTx_EI /USARTx_RTO to USARTx_REI/USARTx_RTOI respectively in the interrupt vector table;</p> <p>12.4.9 Add "Using internal trigger events requires clearing the PWR_FCG0.AOS bit to enable the peripheral circuit trigger function";</p> <p>12.5.2 INT_NMIENR.PREENR Modified to INT_NMIENR.REPENR and INT_NMIENR.RDEDENR to INT_NMIENR.RECCENR;</p> <p>12.5.3 INT_NMIFR.RPEFR is modified to INT_NMIFR.REPFR and INT_NMIFR.RDEDFR is modified to INT_NMIFR.RECCFR;</p> <p>12.5.4 INT_NMICFR.RPECFR Modified to INT_NMICFR.REPCFR and INT_NMICFR.RDEDCFR to INT_NMICFR.RECCCFR;</p> <p>12.5.6 INT_EIFR The INT_ prefix is removed from the register bit description;</p> <p>12.5.7 INT_EICFR The INT_ prefix is removed from the register bit description;</p> <p>12.5.10 INT_VSEL Prefix V added to register bit description;</p> <p>12.5.14 INT_IER The INT_ prefix is removed from the register bit description;</p> <p>9) 15.3 Example of an updated application;</p> <p>15.4.7 Update register descriptions;</p> <p>15.3.1/15.3.2 Delete the description of the illegal access action;</p> <p>15.4.1 Register marking error modification;</p> <p>15.4.3 b17/b16/b9/b8/b1/b0 Marking and function error modification;</p>

		<ul style="list-style-type: none">10) 16.4 Examples of updated applications;11) 18.5.3 Add a description of the AD function enable bit corresponding to PWR_FCG3 before entering STOP mode to reduce power consumption;12) In Chapter 20, optimize the legend of Timer6 section; add chapters of [Pulse Width Measurement], [Cycle Measurement], [Typical Application Example]; add the description of cycle by cycle for EMB control;13) 21.5.3 CCSR.DCLK Revised to CCSR.CKDIV;
--	--	--

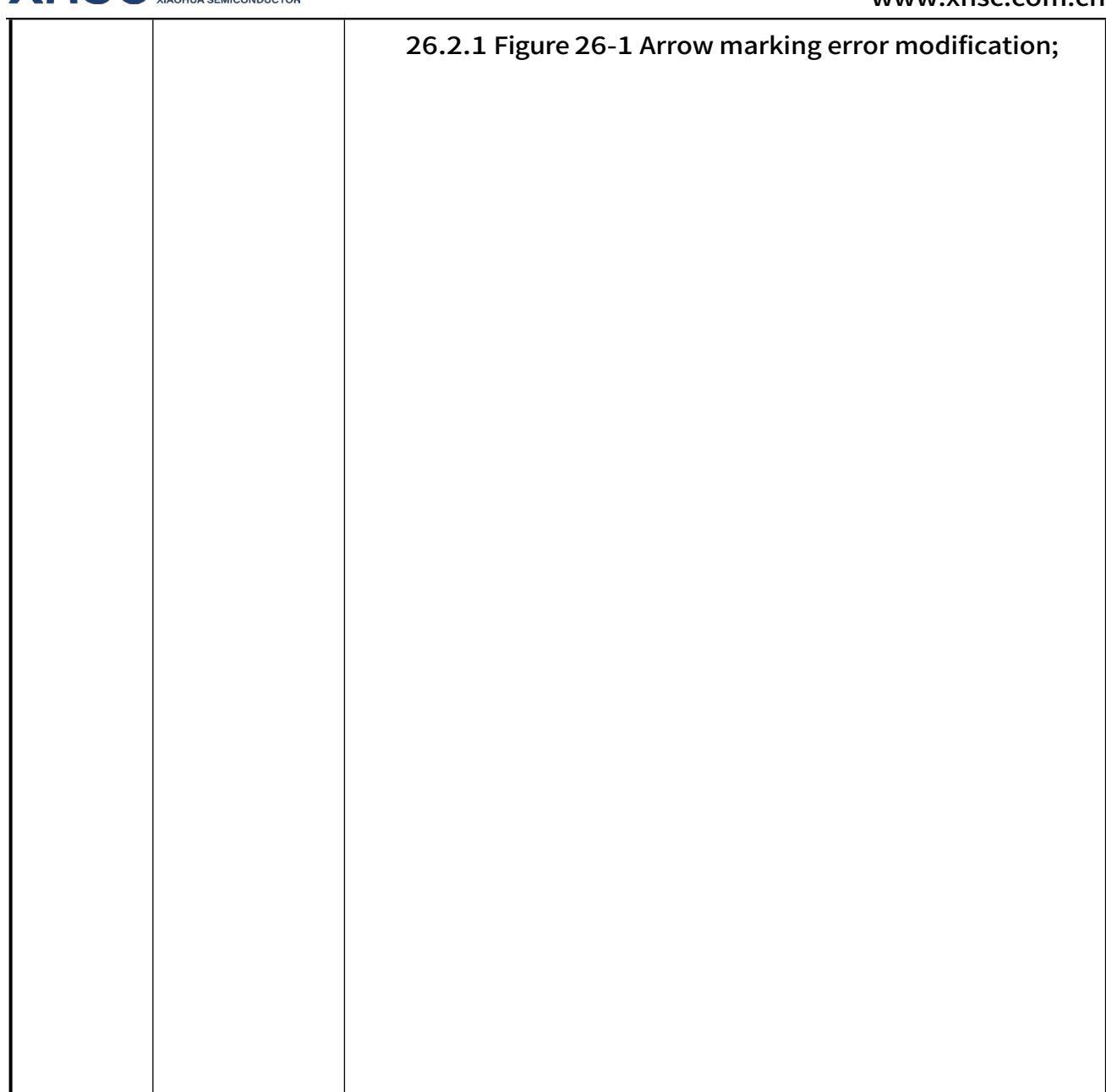
version number	revision date	revision
		<p>14) Optimize the Timer0 and TimerA part of the legend in Chapters 23 and 24;</p> <p>Modify the description of TimerA CCONR register bit9~8 and PCONR register bit1~0 in Chapter 23;</p> <p>15) 25.5.3 ALMF omission in RTC_CR2 register bit description;</p> <p>16) 27.5.3 Initial value modifier bit 0x0000 FFFF for USART_BRR;</p> <p>27.5.6 BCN Bit Width Matching in USART_CR3 Register Table;</p> <p>17) 32.4.3 RAIE Modified to RAFIE;</p> <p>32.5.7 CAN_RTIE bit5 mnemonic modification;</p> <p>32.5.8 Bit5 of CAN_RTIF was modified by mistake;</p> <p>32.5.9 CAN_ERRINT Form b5 WPIE modified to EPIE;</p> <p>32.5.12 CAN_AFWL modified to CAN_LIMIT;</p> <p>32 Full Chapter TRG_TRIG modified to TT_TRIG and TRG_WTRIG modified to TT_WTRIG;</p> <p>32.5. 12CAN_AFWL modified to CAN_LIMITt;</p> <p>18) 37 Add MMC mode enable register description;</p> <p>19) 33.6.2 HFI modified to HFIR;</p> <p>Add 33.7.2.1 USBFS VBUS control register;</p> <p>33.7.2.5 b31 WKUPINT to WKUINT, b7 GOUTNAKEFF to GONAKEFF;</p> <p>20) 33.7.2.6 b31 WKUPINTM is amended to WKUIM, b30 VBUSVM is amended to VBUSWIM, b29 DISCM is amended to DISCIM, b28 CIDSCHG is amended to CIDSCHGM, b7 GOUTNAKEFFM is amended to GONAKEFFM;</p> <p>33.7.3.4 PTXQTOP Revised to PTXQSAV;</p> <p>33.5.4.4 Amend after 15ms to within 15ms;</p> <p>33.7.4.8 INEPxkTXFEM Modified to INEPTXFEM;</p> <p>33.7.4.2 Delete the TCTL bit from the form;</p> <p>33.7.4.11 TO Modified to TOC ITTXFE Modified to TTXFE;</p> <p>21) 38.4.1 Updating the JTAG/SWJ Debug Port Pins.</p>

Rev1.3	2021/12/10	<ul style="list-style-type: none">1) Delete the product characteristics, pin configuration, package information, etc. (Refer to the latest datasheet for related information) and modify the declaration;2) Update some names and optimize descriptions, update pen errors;3) Supplements the precision description of UART communication;
--------	------------	--

version number	revision date	revision
		<p>4) timer 6 Description of the initialization method for adding the interval period valid function;</p> <p>5) Add notes on CAN usage and sample points;</p> <p>6) 18.3.4 The digital filtering function of this group of ports is realized between Unit 1 -> The digital filtering function of this group of ports is set by the FCONR of Unit 1, and the FCONRs of other Units have no effect on the setting of the digital filtering function of this group of ports, so any Unit using this function needs to set the position bit of TIMER6_1 in the function controller (PWR_FCG2);</p> <p>7) 19.5.7 TMR4_OCERn b13,b12 Description modified;</p>
Rev1.4	2021/12/30	Add HC32F45x series and modify Figure 5-6Figure 5-8Figure 18-13, and Table 18-3.
Rev1.41	2022/03/09	Company Logo updated.

		<ul style="list-style-type: none">1) In Table 1-1, add the description of data security protection address, change "ROM" to "FLASH";2) 4.11.1 b7 Bit name clerical error modification; CMU XTAL Configuration Registers b5~b4 Function Description Modified;3) 6.1 "Addresses 0x0000_0408~0x0000_041F are reserved functions" to "address Address 0x0000_0408~0x0000_040F 0x0000_0408~0x0000_040F and 0x0000_0418~0x0000_041F are reserved functions";4) 6.2.3 Initialization Configuration Register split description, new6.2.4 /6.2.5/6.2.6 subsection;5) 7.2 Revise the description of support for data security protection in the main features; 7.3 Add the data security protection address and related descriptions to the FLASH map (Figure 7-1, Figure 7-2).;7.6.5 Add a new description subsection on data security protection;5) 18.3.11.3 Add the description "This function can only be realized with a combination of units 1 and 2, with unit 1 as the position counting unit and unit 2 as the revolution counting unit";6) 20.3.1 EMB_CTL0 Register Bits PWMSEN[2:0],CMPEN[2:0] Description Modification; 20.3.2 EMB_CTL1~3 Register Bits CMPEN[2:0] Description Modification; 20.3.3 EMB_PWMLV0 Register Bit PWMLV Description Modification;7) 21.3.1.3 Figure 21-5 Penmanship modification;8) 25.5.1 b1 "CLR.CFE" changed to "CR1.CFE"; 25.5.4 b20 "CRTOF Bit Write 1 Clear CRTOF" changed to "CRTOF Bit Write 1 clears RTOF";9) 26.1 Corrections for clerical errors;
Rev1.42	2022/03/29	

26.2.1 Figure 26-1 Arrow marking error modification;



version number	revision date	revision
		<p>10) 30.4.1 Add description of CAN communication clock constraints;</p> <p>30.4.5 Add description of data sending constraints;</p> <p>30.5.1 The offset address "0x50" is changed to "0x00", and "mailbox" is changed to "message";</p> <p>30.5.2 The offset address "0x00" is changed to "0x50";</p> <p>30.5.8 b6 Bit Functional Description Modification;</p> <p>30.5.9 b7 bit R/W modification; b6,b0 bit function description modification;</p> <p>30.5.11 b4~b0 Bit R/W Modification;</p> <p>30.5.13 b7~b0 Bit R/W Modification;</p> <p>30.5.14 b7~b0 Bit R/W Modification;</p> <p>30.5.15 b3~b0 Function Description Modification;</p> <p>30.5.19 b5 Bit Functional Description Modification.</p>

Rev1.5	2022/09/14	<ul style="list-style-type: none">1) Added HC32A460 series model number;2) Revise the introduction section "HC32F460" to "HC32F460_F45x_A460"; Add a description for "for automotive electronics";3) 18.3.4 Digital Filtering Function Description Modification;4) 19.3.1.3 Figure 19-6 Original Sawtooth Wave Flat Peaks Modified to Spikes;<ul style="list-style-type: none">19.3.3.1 Figure 19-16 Original Sawtooth Wave Flat Peaks Modified to Spikes;19.3.3.2 Figure 19-18 Original Sawtooth Wave Flat Peaks Modified to Spikes;19.5.3 Register TMR4_CCSR bit b15 Function Description, "Note 2" Description Modified;5) 23.3.1 RTC register reset description modified;<ul style="list-style-type: none">23.5.14 Corrected a clerical error by changing "Hourly Alarm Register" to "Weekly Alarm Register";6) 25.5.1 Register USART_SR Bit b1 Function Description Wrongly modified;7) 27.12.3 Register SPI_CFG1 Bits b30~b28 Written error in function description modified;<ul style="list-style-type: none">28.3.1 Penmanship modifications in Figure 28-4 and Figure 28-7;28.4.5 Penmanship revision;28.5.6 Pen error modification in Figure 28-20;8) 30.4.17 Receive BUF full interrupt flag pen error modification;<ul style="list-style-type: none">30.5.18 Register CAN_TBSLOT Bit b6 Marks pen error modification;30.5.23 Register CAN_TT_WTRIG Bits b15~b0 Marker for pen error modification;
--------	------------	---

version number	revision date	revision
Rev1.6	2023/09/27	<p>9) 33.4.4 Register DCU_FLAGCLR read/write modify to "R/W".</p> <p>1) Table 3-2 Corrected reset method and reset flag;</p> <p>2) Correct Figure 4-4 pin names; Delete Figure 4-1 RTCLRC Leads, Fix adc clk Leads; Delete RTCLRC Option in 4.11.32/ 4.11.33;</p> <p>3) 5.4.3/ 5.4.4 Additional functional description;</p> <p>4) 7.9.11/ 7.9.12/ 7.9.13 Add description of EFM_UQID;</p> <p>5) 8.2.1 Add a note on the use of ECC;</p> <p>6) 10.3.2 Modify the name of TMR4<t>_SCMx; Modify 10.5.7 Register name;</p> <p>7) 11.2.2 Increase the serial number of the AOS target list; 11.4 Harmonization of register names;</p> <p>8) Amend 13.4.6/ 13.4.3 Register description;</p> <p>9) 14.6 Add precautions for use;</p> <p>10) Correction 19.5.4/ 19.5.6/ 19.5.7/ 19.5.8/ 19.5.15 Register Description; Corrected signage in Figures 19-6/ 19-7/ 19-13/ 19-16/ 19-17 and added Note to Figure 19-6;</p> <p>11) Optimization 20.3.8 Register Bit Names;</p> <p>12) Split 21.5.4 Registers;</p> <p>13) Fixed register names in 23.3.2; optimized 23.5.4 Bit Bit Description;</p> <p>14) 25.4.1.5 Additional Abbreviations: Optimization Figures 25-3/25-6; Split 25.5.2 registers; amended 25.5.4 register description;</p>

		<p>Amendment 25.4.1.6 Formulas and explanations;</p> <p>15) Amend the description in Table 26-1;</p> <p>16) Correct Figure 30-4</p> <p>characters; Table 30-3</p> <p>Add reset name;</p> <p>Amendments to 30.4.7/ 30.4.8/ 30.4.9/ 30.4.10/</p> <p>30.4.12/ 30.4.13/ 30.4.16.4 Functional</p> <p>description;</p> <p>Amend Table 30-5/ 30-6/ 30-10 characters;</p>
--	--	---

version number	revision date	revision
		<p>Amendments 30.5.4/ 30.5.5/ 30.5.7/ 30.5.8/ 30.5.9/ 30.5.10/ 30.5.11/ 30.5.12/ 30.5.13/ 30.5.14/ 30.5.19 Register contents;</p> <p>17) Add Chapter 33.5 Notes;</p> <p>18) Correct the 34.4.1/ 34.4.2/ 34.4.3/ 34.4.4 reset values;</p> <p>19) Correct the characters in Figure 36-1;</p> <p>Amendment 36.5.2 Register contents.</p>