



Holtek 32-Bit Microcontroller with Arm® Cortex®-M0+ Core

HT32F0008 User Manual

Revision: V1.20 Date: November 22, 2019

[**www.holtek.com**](http://www.holtek.com)

Table of Contents

1 Introduction	21
Overview	21
Features	21
Device Information	25
Block Diagram	26
2 Document Conventions	27
3 System Architecture	28
Arm® Cortex®-M0+ Processor	28
Bus Architecture	29
Memory Organization	30
Memory Map	31
Embedded Flash Memory	33
Embedded SRAM Memory	33
AHB Peripherals	33
APB Peripherals	33
4 Flash Memory Controller (FMC)	34
Introduction	34
Features	34
Functional Descriptions	35
Flash Memory Map	35
Flash Memory Architecture	36
Wait State Setting	36
Bootling Configuration	37
Page Erase	37
Mass Erase	39
Word Programming	40
Option Byte Description	41
Page Erase/Program Protection	42
Security Protection	43
Register Map	44
Register Descriptions	45
Flash Target Address Register – TADR	45
Flash Write Data Register – WRDR	46
Flash Operation Command Register – OCMR	47
Flash Operation Control Register – OPCR	48
Flash Operation Interrupt Enable Register – OIER	49
Flash Operation Interrupt and Status Register – OISR	50
Flash Page Erase/Program Protection Status Register – PPSR	52
Flash Security Protection Status Register – CPSR	53

Flash Vector Mapping Control Register – VMCR	54
Flash Manufacturer and Device ID Register – MDID	55
Flash Page Number Status Register – PNSR	56
Flash Page Size Status Register – PSSR	57
Device ID Register – DID	57
Flash Pre-fetch Control Register – CFPCR	58
Custom ID Register n – CIDRn (n = 0 ~ 3)	59
5 Power Control Unit (PWRCU)	60
Introduction	60
Features	61
Functional Descriptions	61
V _{DD} Power Domain	61
1.5 V Power Domain	63
Operation Modes	63
Register Map	65
Register Descriptions	66
Power Control Status Register – PWRSR	66
Power Control Register – PWRCR	67
V _{DD} Power Domain Test Register – PWRTEST	69
Low Voltage / Brown-Out Detect Control and Status Register – LVDCSR	70
6 Clock Control Unit (CKCU)	72
Introduction	72
Features	74
Functional Descriptions	74
High Speed External Crystal Oscillator – HSE	74
High Speed Internal RC Oscillator – HSI	75
Auto Trimming of High Speed Internal RC Oscillator – HSI	75
Phase Locked Loop – PLL	77
USB Phase Locked Loop – USB PLL	78
Low Speed External Crystal Oscillator – LSE	80
Low Speed Internal RC Oscillator – LSI	80
Clock Ready Flag	80
System Clock (CK_SYS) Selection	80
HSE Clock Monitor	81
Clock Output Capability	81
Register Map	82
Register Descriptions	83
Global Clock Configuration Register – GCFGR	83
Global Clock Control Register – GCCR	84
Global Clock Status Register – GCSR	86
Global Clock Interrupt Register – GCIR	87
PLL Configuration Register – PLLCFGR	88

PLL Control Register – PLLCR	89
AHB Configuration Register – AHBCFGR	90
AHB Clock Control Register – AHBCCR	91
APB Clock Control Register 0 – APBCCR0	93
APB Clock Control Register 1 – APBCCR1	94
Clock Source Status Register – CKST	95
APB Peripheral Clock Selection Register 0 – APBPCSR0	96
APB Peripheral Clock Selection Register 1 – APBPCSR1	97
HSI Control Register – HSICR	98
HSI Auto Trimming Counter Register – HSIATCR	100
APB Peripheral Clock Selection Register 2 – APBPCSR2	101
Low Power Control Register – LPCR	102
MCU Debug Control Register – MCUDBGCR	103
7 Reset Control Unit (RSTCU)	105
Introduction	105
Functional Descriptions	105
Power-On Reset	105
System Reset	106
AHB and APB Unit Reset	106
Register Map	106
Register Descriptions	107
Global Reset Status Register – GRSR	107
AHB Peripheral Reset Register – AHBPRSTR	108
APB Peripheral Reset Register 0 – APBPRSTR0	109
APB Peripheral Reset Register 1 – APBPRSTR1	110
8 General Purpose I/O (GPIO)	112
Introduction	112
Features	113
Functional Descriptions	113
Default GPIO Pin Configuration	113
General Purpose I/O – GPIO	113
GPIO Locking Mechanism	115
Register Map	115
Register Descriptions	116
Port A Data Direction Control Register – PADIRCR	116
Port A Input Function Enable Control Register – PAINER	117
Port A Pull-Up Selection Register – PAPUR	118
Port A Pull-Down Selection Register – PAPDR	119
Port A Open-Drain Selection Register – PAODR	120
Port A Drive Current Selection Register – PADRVCR	121
Port A Lock Register – PALOCKR	122
Port A Data Input Register – PADINR	123

Port A Output Data Register – PADOUTR	123
Port A Output Set/Reset Control Register – PASRR	124
Port A Output Reset Register – PARR	125
Port B Data Direction Control Register – PBDIRCR	125
Port B Input Function Enable Control Register – PBINER	126
Port B Pull-Up Selection Register – PBPUR	127
Port B Pull-Down Selection Register – PBPDR	128
Port B Open-Drain Selection Register – PBODR	129
Port B Drive Current Selection Register – PBDRVR	130
Port B Lock Register – PBLOCKR	131
Port B Data Input Register – PBDINR	132
Port B Output Data Register – PBDOUTR	132
Port B Output Set/Reset Control Register – PBSRR	133
Port B Output Reset Register – PBRR	134
Port C Data Direction Control Register – PCDIRCR	134
Port C Input Function Enable Control Register – PCINER	135
Port C Pull-Up Selection Register – PCPUR	136
Port C Pull-Down Selection Register – PCPDR	137
Port C Open-Drain Selection Register – PCODR	138
Port C Drive Current Selection Register – PCDRVR	139
Port C Lock Register – PCLOCKR	140
Port C Data Input Register – PCDINR	141
Port C Output Data Register – PCDOUTR	141
Port C Output Set/Reset Control Register – PCSRR	142
Port C Output Reset Register – PCRR	143
Port F Data Direction Control Register – PFDIRCR	143
Port F Input Function Enable Control Register – PFINER	144
Port F Pull-Up Selection Register – PFPUR	145
Port F Pull-Down Selection Register – PFPDR	146
Port F Open-Drain Selection Register – PFODR	147
Port F Drive Current Selection Register – PFDRVR	148
Port F Lock Register – PFLOCKR	149
Port F Data Input Register – PFDINR	150
Port F Output Data Register – PFDOUTR	150
Port F Output Set/Reset Control Register – PFSRR	151
Port F Output Reset Register – PFRR	152
9 Alternate Function Input/Output Control Unit (AFIO)	153
Introduction	153
Features	154
Functional Descriptions	154
External Interrupt Pin Selection	154
Alternate Function	155
Lock Mechanism	155
Register Map	155

Register Descriptions.....	156
EXTI Source Selection Register 0 – ESSR0	156
EXTI Source Selection Register 1 – ESSR1	157
GPIO x Configuration Low Register – GPxCFGLR, x = A, B, C, F	158
GPIO x Configuration High Register – GPxCFGHR, x = A, B, C, F.....	159
10 Nested Vectored Interrupt Controller (NVIC).....	160
Introduction	160
Features.....	161
Functional Descriptions	161
SysTick Calibration	161
Register Map	162
11 External Interrupt/Event Controller (EXTI).....	163
Introduction	163
Features.....	163
Functional Descriptions	164
Wakeup Event Management.....	164
External Interrupt/Event Line Mapping	164
Interrupt and Debounce	164
Register Map	165
Register Descriptions.....	166
EXTI Interrupt n Configuration Register – EXTICFGRn, n = 0 ~ 15	166
EXTI Interrupt Control Register – EXTICR	167
EXTI Interrupt Edge Flag Register – EXTIEDGEFLGR.....	168
EXTI Interrupt Edge Status Register – EXTIEDGESR	169
EXTI Interrupt Software Set Command Register – EXTISSCR.....	170
EXTI Interrupt Wakeup Control Register – EXTIWAKUPCR	171
EXTI Interrupt Wakeup Polarity Register – EXTIWAKUPPOLR	172
EXTI Interrupt Wakeup Flag Register – EXTIWAKUPFLG	173
12 General-Purpose Timer (GPTM)	174
Introduction	174
Features.....	175
Functional Descriptions	175
Counter Mode	175
Clock Controller	177
Trigger Controller	179
Slave Controller	180
Master Controller	182
Channel Controller	183
Input Stage	185
Quadrature Decoder	187
Output Stage.....	188

Update Management	192
Single Pulse Mode	193
Asymmetric PWM Mode	195
Timer Interconnection	195
PDMA Request	198
Register Map	199
Register Descriptions.....	200
Timer Counter Configuration Register – CNTCFR	200
Timer Mode Configuration Register – MDCFR	201
Timer Trigger Configuration Register – TRCFR.....	204
Timer Counter Register – CTR	205
Channel 0 Input Configuration Register – CH0ICFR	206
Channel 1 Input Configuration Register – CH1ICFR	207
Channel 2 Input Configuration Register – CH2ICFR	209
Channel 3 Input Configuration Register – CH3ICFR	210
Channel 0 Output Configuration Register – CH0OCFR	212
Channel 1 Output Configuration Register – CH1OCFR	214
Channel 2 Output Configuration Register – CH2OCFR	216
Channel 3 Output Configuration Register – CH3OCFR	218
Channel Control Register – CHCTR.....	220
Channel Polarity Configuration Register – CHPOLR.....	221
Timer PDMA/Interrupt Control Register – DICTR	222
Timer Event Generator Register – EVGR.....	224
Timer Interrupt Status Register – INTSR.....	225
Timer Counter Register – CNTR.....	228
Timer Prescaler Register – PSCR	229
Timer Counter-Reload Register – CRR	230
Channel 0 Capture/Compare Register – CH0CCR	231
Channel 1 Capture/Compare Register – CH1CCR	232
Channel 2 Capture/Compare Register – CH2CCR	233
Channel 3 Capture/Compare Register – CH3CCR	234
Channel 0 Asymmetric Compare Register – CH0ACR.....	235
Channel 1 Asymmetric Compare Register – CH1ACR.....	235
Channel 2 Asymmetric Compare Register – CH2ACR.....	236
Channel 3 Asymmetric Compare Register – CH3ACR.....	236
13 Pulse-Width-Modulation Timer (PWM).....	237
Introduction	237
Features.....	238
Functional Descriptions	238
Counter Mode	238
Clock Controller	241
Trigger Controller	242
Slave Controller	243

Master Controller	245
Channel Controller	246
Output Stage	246
Update Management	250
Single Pulse Mode	250
Asymmetric PWM Mode	253
Timer Interconnection	253
Trigger Peripherals Start	256
PDMA Request	256
Register Map	257
Register Descriptions	258
Timer Counter Configuration Register – CNTCFR	258
Timer Mode Configuration Register – MDCFR	259
Timer Trigger Configuration Register – TRCFR	262
Timer Counter Register – CTR	263
Channel 0 Output Configuration Register – CH0OCFR	264
Channel 1 Output Configuration Register – CH1OCFR	266
Channel 2 Output Configuration Register – CH2OCFR	268
Channel 3 Output Configuration Register – CH3OCFR	270
Channel Control Register – CHCTR	272
Channel Polarity Configuration Register – CHPOLR	273
Timer PDMA/Interrupt Control Register – DICTR	274
Timer Event Generator Register – EVGR	275
Timer Interrupt Status Register – INTSR	276
Timer Counter Register – CNTR	278
Timer Prescaler Register – PSCR	278
Timer Counter-Reload Register – CRR	279
Channel 0 Compare Register – CH0CR	279
Channel 1 Compare Register – CH1CR	280
Channel 2 Compare Register – CH2CR	280
Channel 3 Compare Register – CH3CR	281
Channel 0 Asymmetric Compare Register – CH0ACR	281
Channel 1 Asymmetric Compare Register – CH1ACR	282
Channel 2 Asymmetric Compare Register – CH2ACR	282
Channel 3 Asymmetric Compare Register – CH3ACR	283
14 Basic Function Timer (BFTM)	284
Introduction	284
Features	284
Functional Description	285
Repetitive Mode	285
One Shot Mode	286
Register Map	287
Register Descriptions	287

BFTM Control Register – BFTMCR	287
BFTM Status Register – BFTMSR.....	288
BFTM Counter Value Register – BFTMCNTR	289
BFTM Compare Value Register – BFTMCMPR	289
15 Real Time Clock (RTC)	290
Introduction	290
Features.....	290
Functional Descriptions	291
RTC Related Register Reset	291
Reading RTC Register.....	291
Low Speed Clock Configuration	291
RTC Counter Operation.....	292
Interrupt and Wakeup Control.....	292
RTCOUNT Output Pin Configuration.....	292
Register Map	294
Register Descriptions.....	294
RTC Counter Register – RTCCNT.....	294
RTC Compare Register – RTCCMP	295
RTC Control Register – RTCCR.....	296
RTC Status Register – RTCSR.....	298
RTC Interrupt and Wakeup Enable Register – RTCIWEN.....	299
16 Watchdog Timer (WDT)	300
Introduction	300
Features.....	301
Functional Description	301
Register Map	303
Register Descriptions.....	303
Watchdog Timer Control Register – WDTCR	303
Watchdog Timer Mode Register 0 – WDTMR0.....	304
Watchdog Timer Mode Register 1 – WDTMR1.....	305
Watchdog Timer Status Register – WDTSR.....	306
Watchdog Timer Protection Register – WDTPR.....	307
Watchdog Timer Clock Selection Register – WDTCSR.....	308
17 Inter-Integrated Circuit (I²C).....	309
Introduction	309
Features.....	310
Functional Descriptions	310
Two-Wire Serial Interface	310
START and STOP Conditions.....	310
Data Validity.....	311
Addressing Format	311

Data Transfer and Acknowledge	313
Clock Synchronization	313
Arbitration	314
General Call Addressing	314
Bus Error	314
Address Mask Enable	315
Address Snoop	315
Operation Mode	315
Conditions of Holding SCL Line	319
I ² C Timeout Function	320
PDMA Interface	320
Register Map	321
Register Descriptions	321
I ² C Control Register – I2CCR	321
I ² C Interrupt Enable Register – I2CIER	323
I ² C Address Register – I2CADDR	325
I ² C Status Register – I2CSR	326
I ² C SCL High Period Generation Register – I2CSHPGR	329
I ² C SCL Low Period Generation Register – I2CSLPGR	330
I ² C Data Register – I2CDR	331
I ² C Target Register – I2CTAR	332
I ² C Address Mask Register – I2CADDRMR	333
I ² C Address Snoop Register – I2CADDRSR	334
I ² C Timeout Register – I2CTOUT	335
18 Serial Peripheral Interface (SPI)	336
Introduction	336
Features	337
Functional Descriptions	337
Master Mode	337
Slave Mode	337
SPI Serial Frame Format	337
Status Flags	342
PDMA Interface	344
Register Map	345
Register Descriptions	345
SPI Control Register 0 – SPICR0	345
SPI Control Register 1 – SPICR1	347
SPI Interrupt Enable Register – SPIIER	349
SPI Clock Prescaler Register – SPICPR	350
SPI Data Register – SPIDR	351
SPI Status Register – SPISR	352
SPI FIFO Control Register – SPIFCR	353
SPI FIFO Status Register – SPIFSR	354
SPI FIFO Time Out Counter Register – SPIFTOCR	355

19 Universal Synchronous Asynchronous Receiver Transmitter (USART)	356
Introduction	356
Features	357
Functional Descriptions	357
Serial Data Format	357
Baud Rate Generation	358
Hardware Flow Control	360
IrDA	361
RS485 Mode	363
Synchronous Master Mode	365
Interrupts and Status	367
PDMA Interface	367
Register Map	367
Register Descriptions	368
USART Data Register – USRDR	368
USART Control Register – USRCR	369
USART FIFO Control Register – USRFCR	371
USART Interrupt Enable Register – USRIER	372
USART Status & Interrupt Flag Register – USRSIFR	374
USART Timing Parameter Register – USRTPR	376
USART IrDA Control Register – IrDACR	377
USART RS485 Control Register – RS485CR	378
USART Synchronous Control Register – SYNCR	379
USART Divider Latch Register – USRDLR	380
USART Test Register – USRTSTR	381
20 Universal Asynchronous Receiver Transmitter (UART)	382
Introduction	382
Features	383
Functional Descriptions	383
Serial Data Format	383
Baud Rate Generation	384
Interrupts and Status	385
PDMA Interface	385
Register Map	386
Register Descriptions	386
UART Data Register – URDR	386
UART Control Register – URCR	387
UART Interrupt Enable Register – URIER	388
UART Status & Interrupt Flag Register – URSIFR	390
UART Divider Latch Register – URDLR	391
UART Test Register – URTSTR	392

21 USB Device Controller (USB)	393
Introduction	393
Features	393
Functional Descriptions	394
Endpoints	394
EP_SRAM	394
Serial Interface Engine – SIE	395
Double-Buffering	395
Suspend Mode and Wake-up	396
Remote Wake-up	397
Register Map	397
Register Descriptions	398
USB Control and Status Register – USBCSR	398
USB Interrupt Enable Register – USBIER	400
USB Interrupt Status Register – USBISR	401
USB Frame Count Register – USBFCR	402
USB Device Address Register – USBDEVAR	403
USB Endpoint 0 Control and Status Register – USBEP0CSR	404
USB Endpoint 0 Interrupt Enable Register – USBEP0IER	405
USB Endpoint 0 Interrupt Status Register – USBEP0ISR	407
USB Endpoint 0 Transfer Count Register – USBEP0TCR	408
USB Endpoint 0 Configuration Register – USBEP0CFGR	409
USB Endpoint 1 ~ 3 Control and Status Register – USBEPnCSR, n = 1 ~ 3	410
USB Endpoint 1 ~ 3 Interrupt Enable Register – USBEPnIER, n = 1 ~ 3	411
USB Endpoint 1 ~ 3 Interrupt Status Register – USBEPnISR, n = 1 ~ 3	412
USB Endpoint 1 ~ 3 Transfer Count Register – USBEPnTCR, n = 1 ~ 3	413
USB Endpoint 1 ~ 3 Configuration Register – USBEPnCFGR, n = 1 ~ 3	414
USB Endpoint 4 ~ 7 Control and Status Register – USBEPnCSR, n = 4 ~ 7	415
USB Endpoint 4 ~ 7 Interrupt Enable Register – USBEPnIER, n = 4 ~ 7	417
USB Endpoint 4 ~ 7 Interrupt Status Register – USBEPnISR, n = 4 ~ 7	418
USB Endpoint 4 ~ 7 Transfer Count Register – USBEPnTCR, n = 4 ~ 7	419
USB Endpoint 4 ~ 7 Configuration Register – USBEPnCFGR, n = 4 ~ 7	420
22 Peripheral Direct Memory Access (PDMA)	421
Introduction	421
Features	421
Functional Description	422
AHB Master	422
PDMA Channel	422
PDMA Request Mapping	422
Channel Transfer	423
Channel Priority	423
Transfer Request	424
Address Mode	424

Auto-Reload	425
Transfer Interrupt	425
Register Map	425
Register Descriptions.....	427
PDMA Channel n Control Register – PDMACHnCR, n = 0 ~ 5	427
PDMA Channel n Source Address Register – PDMACHnSADR, n = 0 ~ 5.....	429
PDMA Channel n Destination Address Register – PDMACHnDADR, n = 0 ~ 5.....	430
PDMA Channel n Transfer Size Register – PDMACHnTSR, n = 0 ~ 5.....	431
PDMA Channel n Current Transfer Size Register – PDMACHnCTSR, n = 0 ~ 5.....	432
PDMA Interrupt Status Register – PDMAISR	433
PDMA Interrupt Status Clear Register – PDMAISCR	434
PDMA Interrupt Enable Register – PDMAIER	435
23 Divider (DIV)	437
Introduction	437
Features.....	437
Functional Descriptions	437
Register Map	438
Register Descriptions.....	438
Divider Control Register – CR	438
Dividend Data Register – DDR.....	439
Divisor Data Register – DSR	439
Quotient Data Register – QTR.....	440
Remainder Data Register – RMR.....	440
24 Cyclic Redundancy Check (CRC).....	441
Introduction	441
Features.....	441
Functional Descriptions	442
CRC Computation.....	442
Byte and Bit Reversal for CRC Computation.....	442
CRC with PDMA	443
Register Map	443
Register Descriptions.....	443
CRC Control Register – CRCCR	443
CRC Seed Register – CRCSDR.....	444
CRC Checksum Register – CRCCSR	445
CRC Data Register – CRCDR	446
25 AES Encrypt/Decrypt Interface (AES).....	447
Introduction	447
Features.....	447
Functional Descriptions	448

AES Mode Description.....	448
AES Status	450
AES PDMA Interface	450
AES Interrupt	451
AES Initial Vector	451
AES Word Swap	452
Register Map	452
Register Descriptions.....	453
AES Control Register – AESCR	453
AES Status Register – AEISSR	454
AES DMA Register – AESDMAR.....	455
AES Interrupt Status Register – AESISR.....	456
AES Interrupt Enable Register – AESIER.....	457
AES DATA Input Register – AESDINR.....	458
AES DATA Output Register – AESDOUTR.....	458
AES Key Register n – AESKEYRn, n = 0 ~ 3	459
AES Initial Vector Register n – AESIVRn, n = 0 ~ 3	459

List of Tables

Table 1. Features and Peripheral List	25
Table 2. Document Conventions	27
Table 3. Register Map	32
Table 4. Flash Memory and Option Byte	36
Table 5. Relationship between Wait State Cycle and HCLK	36
Table 6. Booting Modes	37
Table 7. Option Byte Memory Map	41
Table 8. Access Permission of Protected Main Flash Page	42
Table 9. Access Permission When Security Protection is Enabled	43
Table 10. FMC Register Map	44
Table 11. Operation Mode Definitions	63
Table 12. Enter/Exit Power Saving Modes	64
Table 13. Power Status after System Reset	65
Table 14. PWRCU Register Map	65
Table 15. Output Divider 2 Value Mapping	77
Table 16. Feedback Divider 2 Value Mapping	78
Table 17. USB PLL Output Divider 2 Value Mapping	79
Table 18. USB PLL Feedback Divider 2 Value Mapping	79
Table 19. CKOUT Clock Source	82
Table 20. CKCU Register Map	82
Table 21. RSTCU Register Map	106
Table 22. AFIO, GPIO and IO Pad Control Signal True Table	114
Table 23. GPIO Register Map	115
Table 24. AFIO Selection for Peripheral Map Example	155
Table 25. AFIO Register Map	155
Table 26. Exception Types	160
Table 27. NVIC Register Map	162
Table 28. EXTI Register Map	165
Table 29. Counting Direction and Encoding Signals	188
Table 30. Compare Match Output Setup	189
Table 31. GPTM Register Map	199
Table 32. GPTM Internal Trigger Connection	204
Table 33. Compare Match Output Setup	247
Table 34. PWM Register Map	257
Table 35. PWM Internal Trigger Connection	262
Table 36. BFTM Register Map	287
Table 37. LSE Startup Mode Operating Current and Startup Time	291
Table 38. RTCOUT Output Mode and Active Level Setting	293
Table 39. RTC Register Map	294

Table 40. Watchdog Timer Register Map	303
Table 41. Conditions of Holding SCL line.....	319
Table 42. I ² C Register Map	321
Table 43. I ² C Clock Setting Example	330
Table 44. SPI Interface Format Setup.....	338
Table 45. SPI Mode Fault Trigger Conditions	343
Table 46. SPI Master Mode SEL Pin Status	343
Table 47. SPI Register Map	345
Table 48. Baud Rate Deviation Error Calculation – CK_USART = 40 MHz	359
Table 49. Baud Rate Deviation Error Calculation – CK_USART = 48 MHz	359
Table 50. Baud Rate Deviation Error Calculation – CK_USART = 60 MHz	359
Table 51. USART Register Map.....	367
Table 52. Baud Rate Deviation Error Calculation – CK_USART = 40 MHz	384
Table 53. Baud Rate Deviation Error Calculation – CK_USART = 48 MHz	385
Table 54. Baud Rate Deviation Error Calculation – CK_USART = 60 MHz	385
Table 55. UART Register Map	386
Table 56. Endpoint Characteristics	394
Table 57. USB Data Types and Buffer Size	394
Table 58. USB Register Map	397
Table 59. Resume Event Detection	399
Table 60. PDMA Channel Assignments	423
Table 61. PDMA Address Modes	424
Table 62. PDMA Register Map.....	425
Table 63. DIV Register Map.....	438
Table 64. CRC Register Map	443
Table 65. AES Register Map.....	452

List of Figures

Figure 1. Block Diagram	26
Figure 2. Cortex®-M0+ Block Diagram.....	29
Figure 3. Bus Architecture	30
Figure 4. Memory Map.....	31
Figure 5. Flash Memory Controller Block Diagram.....	34
Figure 6. Flash Memory Map.....	35
Figure 7. Vector Remapping	37
Figure 8. Page Erase Operation Flowchart	38
Figure 9. Mass Erase Operation Flowchart	39
Figure 10. Word Programming Operation Flowchart	40
Figure 11. PWRCU Block Diagram	60
Figure 12. Power-On Reset / Power-Down Reset Waveform.....	62
Figure 13. CKCU Block Diagram	73
Figure 14. External Crystal, Ceramic and Resonators for HSE.....	74
Figure 15. HSI Auto Trimming Block Diagram	76
Figure 16. PLL Block Diagram	77
Figure 17. USB PLL Block Diagram.....	78
Figure 18. External Crystal, Ceramic and Resonators for LSE	80
Figure 19. RSTCU Block Diagram.....	105
Figure 20. Power-On Reset Sequence.....	106
Figure 21. GPIO Block Diagram	112
Figure 22. AFIO/GPIO Control Signal.....	114
Figure 23. AFIO Block Diagram	153
Figure 24. EXTI Channel Input Selection	154
Figure 25. EXTI Block Diagram	163
Figure 26. EXTI Wakeup Event Management	164
Figure 27. EXTI Interrupt Debounce Function	165
Figure 28. GPTM Block Diagram.....	174
Figure 29. Up-counting Example	176
Figure 30. Down-counting Example.....	176
Figure 31. Center-aligned Counting Example.....	177
Figure 32. GPTM Clock Source Selection	178
Figure 33. Trigger Control Block.....	179
Figure 34. Slave Controller Diagram	180
Figure 35. GPTM in Restart Mode.....	180
Figure 36. GPTM in Pause Mode	181
Figure 37. GPTM in Trigger Mode	181
Figure 38. Master GPTMn and Slave GPTMm/MCTMm Connection.....	182
Figure 39. MTO Selection	182

Figure 40. Capture/Compare Block Diagram.....	183
Figure 41. Input Capture Mode.....	184
Figure 42. PWM Pulse Width Measurement Example.....	185
Figure 43. Channel 0 and Channel 1 Input Stages.....	185
Figure 44. Channel 2 and Channel 3 Input Stages.....	186
Figure 45. TI0 Digital Filter Diagram with N = 2.....	186
Figure 46. Input Stage and Quadrature Decoder Block Diagram.....	187
Figure 47. Both TI0 and TI1 Quadrature Decoder Counting.....	188
Figure 48. Output Stage Block Diagram.....	188
Figure 49. Toggle Mode Channel Output Reference Signal (CHxPRE = 0)	189
Figure 50. Toggle Mode Channel Output Reference Signal (CHxPRE = 1)	190
Figure 51. PWM Mode Channel Output Reference Signal and Counter in Up-counting Mode	190
Figure 52. PWM Mode Channel Output Reference Signal and Counter in Down-counting Mode	191
Figure 53. PWM Mode Channel Output Reference Signal and Counter in Centre-aligned Mode.....	191
Figure 54. Update Event Setting Diagram	192
Figure 55. Single Pulse Mode.....	193
Figure 56. Immediate Active Mode Minimum Delay	194
Figure 57. Asymmetric PWM Mode versus Center-Aligned Counting Mode	195
Figure 58. Pausing PWM0 using the GPTM CH0OREF Signal.....	196
Figure 59. Triggering PWM0 with GPTM Update Event	196
Figure 60. Trigger GPTM and PWM0 with the GPTM CH0 Input	197
Figure 61. GPTM PDMA Mapping Diagram.....	198
Figure 62. PWM Block Diagram	237
Figure 63. Up-counting Example	238
Figure 64. Down-counting Example.....	239
Figure 65. Center-aligned Counting Example.....	240
Figure 66. PWM Clock Selection Source.....	241
Figure 67. Trigger Control Block	242
Figure 68. Slave Controller Diagram	243
Figure 69. PWM in Restart Mode	243
Figure 70. PWM in Pause Mode.....	244
Figure 71. PWM in Trigger Mode.....	244
Figure 72. Master PWMn and Slave PWMm/TMm Connection.....	245
Figure 73. MTO Selection	245
Figure 74. Compare Block Diagram	246
Figure 75. Output Stage Block Diagram.....	246
Figure 76. Toggle Mode Channel Output Reference Signal (CHxPRE = 0)	247
Figure 77. Toggle Mode Channel Output Reference Signal (CHxPRE = 1)	248
Figure 78. PWM Mode Channel Output Reference Signal and Counter in Up-counting Mode	248
Figure 79. PWM Mode Channel Output Reference Signal and Counter in Down-counting Mode	249
Figure 80. PWM Mode Channel Output Reference Signal and Counter in Center-aligned Mode.....	249

Figure 81. Update Event Setting Diagram	250
Figure 82. Single Pulse Mode.....	251
Figure 83. Immediate Active Mode Minimum Delay	252
Figure 84. Asymmetric PWM Mode versus Center-aligned Counting Mode.....	253
Figure 85. Pausing PWM1 using the PWM0 CH0OREF Signal	254
Figure 86. Triggering PWM1 with PWM0 Update Event.....	254
Figure 88. PWM PDMA Mapping Diagram	256
Figure 89. BFTM Block Diagram	284
Figure 90. BFTM – Repetitive Mode.....	285
Figure 91. BFTM – One Shot Mode.....	286
Figure 92. BFTM – One Shot Mode Counter Updating	286
Figure 93. RTC Block Diagram.....	290
Figure 94. Watchdog Timer Block Diagram	300
Figure 95. Watchdog Timer Behavior	302
Figure 96. I ² C Module Block Diagram.....	309
Figure 97. START and STOP Condition	311
Figure 98. Data Validity.....	311
Figure 99. 7-bit Addressing Mode.....	312
Figure 100. 10-bit Addressing Write Transmit Mode	312
Figure 101. 10-bits Addressing Read Receive Mode	312
Figure 102. I ² C Bus Acknowledge	313
Figure 103. Clock Synchronization during Arbitration.....	313
Figure 104. Two Masters Arbitration Procedure.....	314
Figure 105. Master Transmitter Timing Diagram	316
Figure 106. Master Receiver Timing Diagram	317
Figure 107. Slave Transmitter Timing Diagram	318
Figure 108. Slave Receiver Timing Diagram	319
Figure 109. SCL Timing Diagram.....	330
Figure 110. SPI Block Diagram.....	336
Figure 111. SPI Single Byte Transfer Timing Diagram – CPOL = 0, CPHA = 0	338
Figure 112. SPI Continuous Data Transfer Timing Diagram – CPOL = 0, CPHA = 0	338
Figure 113. SPI Single Byte Transfer Timing Diagram – CPOL = 0, CPHA = 1	339
Figure 114. SPI Continuous Transfer Timing Diagram – CPOL = 0, CPHA = 1	339
Figure 115. SPI Single Byte Transfer Timing Diagram – CPOL = 1, CPHA = 0	340
Figure 116. SPI Continuous Transfer Timing Diagram – CPOL = 1, CPHA = 0	340
Figure 117. SPI Single Byte Transfer Timing Diagram – CPOL = 1, CPHA = 1	341
Figure 118. SPI Continuous Transfer Timing Diagram – CPOL = 1, CPHA = 1	341
Figure 119. SPI Multi-Master Slave Environment.....	343
Figure 120. USART Block Diagram	356
Figure 121. USART Serial Data Format	358
Figure 122. USART Clock CK_USART and Data Frame Timing	358

Figure 123. Hardware Flow Control between 2 USARTs.....	360
Figure 124. USART RTS Flow Control	360
Figure 125. USART CTS Flow Control	361
Figure 126. IrDA Modulation and Demodulation.....	361
Figure 127. USART I/O and IrDA Block Diagram	363
Figure 128. RS485 Interface and Waveform	364
Figure 129. USART Synchronous Transmission Example	365
Figure 130. 8-bit Format USART Synchronous Waveform	366
Figure 131. UART Block Diagram.....	382
Figure 132. UART Serial Data Format.....	383
Figure 133. UART Clock CK_UART and Data Frame Timing.....	384
Figure 134. USB Block Diagram.....	393
Figure 135. Endpoint Buffer Allocation Example.....	395
Figure 136. Double-buffering Operation Example	396
Figure 137. PDMA Block Diagram	421
Figure 138. PDMA Request Mapping Architecture	422
Figure 139. PDMA Channel Arbitration and Scheduling Example	424
Figure 140. Divider Functional Diagram	437
Figure 141. CRC Block Diagram	441
Figure 142. CRC Data Bit and Byte Reversal Example.....	442
Figure 143. AES Block Diagram	447
Figure 144. AES-ECB Mode	448
Figure 145. AES-CBC Mode.....	449
Figure 146. AES-CTR Mode	450
Figure 147. AES Interrupt	451
Figure 148. Initial Vector for CTR Mode	451
Figure 149. AES Word Swap Function	452

1 Introduction

Overview

This user manual provides detailed information including how to use the HT32F0008 device, system and bus architecture, memory organization and peripheral instructions. The target audiences for this document are software developers, application developers and hardware developers. For more information regarding pin assignment, package and electrical characteristics, please refer to the HT32F0008 datasheet.

The HT32F0008 device is a high performance and low power consumption 32-bit microcontroller based around an Arm® Cortex®-M0+ processor core. The Cortex®-M0+ is a next-generation processor core which is tightly coupled with Nested Vectored Interrupt Controller (NVIC), SysTick timer and advanced debug support.

The HT32F0008 device operates at a frequency of up to 60 MHz with a Flash accelerator to obtain maximum efficiency. It provides 64 KB of embedded Flash memory for code/data storage and 16 KB of embedded SRAM memory for system operation and application program usage. A variety of peripherals, such as I²C, USART, UART, SPI, PDMA, GPTM, PWM, BFTM, AES-128, DIV, CRC-16/32, RTC, WDT, USB2.0 FS, SW-DP (Serial Wire Debug Port), etc., are also implemented in the device. Several power saving modes provide the flexibility for maximum optimization between wakeup latency and power consumption, which is an especially important consideration in low power applications.

The above features ensure that the HT32F0008 device is suitable for use in a wide range of applications, especially in areas such as sensor hub, Data Bridge, Fingerprint Bridge and so on.

Features

- Core
 - 32-bit Arm® Cortex®-M0+ processor core
 - Up to 60 MHz operating frequency
 - Single-cycle multiplication
 - Integrated Nested Vectored Interrupt Controller (NVIC)
 - 24-bit SysTick timer
- On-chip Memory
 - 64 KB on-chip Flash memory for instruction/data and option bytes
 - 16 KB on-chip SRAM
 - Supports multiple booting modes
- Flash Memory Controller – FMC
 - Flash accelerator to obtain maximum efficiency
 - 32-bit word programming with In System Programming Interface (ISP) and In Application Programming (IAP)
 - Flash protection capability to prevent illegal access
- Reset Control Unit – RSTCU
 - Supply supervisor: Power-On Reset / Power-Down Reset (POR/PDR), Brown-Out Detector and Programmable Low Voltage Detector (LVD)

- Clock Control Unit – CKCU
 - External 4 to 16 MHz crystal oscillator
 - External 32,768 Hz crystal oscillator
 - Internal 8 MHz RC oscillator trimmed to ± 2 % accuracy at 3.3 V operating voltage and 25 °C operating temperature
 - Internal 32 kHz RC oscillator
 - Integrated system clock PLL and USB PLL
 - Independent clock divider and gating bits for peripheral clock sources
- Power Management – PWRCU
 - Single V_{DD} power supply: 1.65 V to 3.6 V
 - Integrated 1.5 V LDO regulator for CPU core, peripherals and memories power supply
 - V_{DD} power supply for RTC
 - Two power domains: V_{DD} and 1.5 V
 - Four power saving modes: Sleep, Deep-Sleep1, Deep-Sleep2, Power-Down
- External Interrupt/Event Controller – EXTI
 - Up to 16 EXTI lines with configurable trigger source and type
 - All GPIO pins can be selected as EXTI trigger source
 - Source trigger type can be high level, low level, negative edge, positive edge or both edge
 - Individual interrupt enable, wakeup enable and status bits for each EXTI line
 - Software interrupt trigger mode for each EXTI line
 - Integrated deglitch filter for short pulse blocking
- I/O ports – GPIO
 - Up to 42 GPIOs
 - Port A, B, C, F are mapped on 16 external interrupts – EXTI
 - Output driving current of almost I/O pins are configurable
- General-Purpose Timer – GPTM
 - 16-bit up/down auto-reload counter
 - Up to 4 independent channels
 - 16-bit programmable prescaler allowing counter clock frequency division by any factor between 1 and 65536
 - Input Capture function
 - Compare Match Output
 - PWM waveform generation with Edge-aligned and Center-aligned Counting Modes
 - Single Pulse Mode Output
 - Encoder interface controller with two inputs using quadrature decoder
- Pulse-Width-Modulation Timer – PWM
 - 16-bit up/down auto-reload counter
 - Up to 4 independent channels for each timer
 - 16-bit programmable prescaler allowing counter clock frequency division by any factor between 1 and 65536
 - Compare Match Output
 - PWM waveform generation with Edge-aligned and Center-aligned Counting Modes
 - Single Pulse Mode Output

- Basic Function Timer – BFTM
 - 32-bit compare/match count-up counter – no I/O control features
 - One shot mode – counting stops after a match condition
 - Repetitive mode – restart counter after a match condition
- Watchdog Timer – WDT
 - 12-bit down-counter with a 3-bit prescaler
 - Reset event for the system
 - Programmable watchdog timer window function
 - Registers write protection function
- Real Time Clock – RTC
 - 24-bit up-counter with a programmable prescaler
 - Alarm function
 - Interrupt and wakeup event
- Inter-integrated Circuit – I²C
 - Supports both master and slave modes with a frequency of up to 1 MHz
 - Provides an arbitration function and clock synchronization
 - Supports 7-bit and 10-bit addressing modes and general call addressing
 - Supports slave multi-addressing mode with maskable address
- Serial Peripheral Interface – SPI
 - Supports both master and slave mode
 - Frequency of up to ($f_{PCLK}/2$) MHz for master mode and ($f_{PCLK}/3$) MHz for slave mode
 - FIFO Depth: 8 levels
 - Multi-master and multi-slave operation
- Universal Synchronous Asynchronous Receiver Transmitter – USART
 - Supports both asynchronous and clocked synchronous serial communication modes
 - Asynchronous operating baud rate clock frequency up to ($f_{PCLK}/16$) MHz and synchronous operating baud rate clock frequency up to ($f_{PCLK}/8$) MHz
 - Capability of full duplex communication
 - Fully programmable serial communication characteristics including word length, parity bit, stop bit and bit order
 - Error detection: Parity, overrun, and frame error
 - Supports Auto hardware flow control mode – RTS, CTS
 - IrDA SIR encoder and decoder
 - RS485 mode with output enable control
 - FIFO Depth: 8-level for both receiver and transmitter
- Universal Asynchronous Receiver Transmitter – UART
 - Asynchronous serial communication operating baud rate clock frequency up to ($f_{PCLK}/16$) MHz
 - Capability of full duplex communication
 - Fully programmable serial communication characteristics including word length, parity bit, stop bit and bit order
 - Error detection: Parity, overrun, and frame error
- Cyclic Redundancy Check – CRC
 - Supports CRC16 polynomial: $0x8005, X^{16} + X^{15} + X^2 + 1$
 - Supports CCITT CRC16 polynomial: $0x1021, X^{16} + X^{12} + X^5 + 1$
 - Supports IEEE-802.3 CRC32 polynomial: $0x04C11DB7, X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$

- Supports 1's complement, byte reverse and bit reverse operation on data and checksum
- Supports byte, half-word and word data size
- Programmable CRC initial seed value
- CRC computation done in 1 AHB clock cycle for 8-bit data and 4 AHB clock cycles for 32-bit data
- Supports PDMA to complete a CRC computation of a block of memory
- Peripheral Direct Memory Access – PDMA
 - 6 channels with trigger source grouping
 - 8/16/32-bit width data transfer
 - Supports linear address, circular address and fixed address mode
 - 4-level programmable channel priority
 - Auto reload mode
 - Supports trigger sources:
SPI, USART, UART, I²C, GPTM, PWM, AES and software request
- Divider – DIV
 - Signed/unsigned 32-bit divider
 - Operation in 8 clock cycles, load in 1 clock cycle
 - Division by zero error flag
- Universal Serial Bus Device Controller – USB
 - Complies with USB 2.0 full-speed (12 Mbps) specification
 - On-chip USB full-speed transceiver
 - 1 control endpoint (EP0) for control transfer
 - 3 single-buffered endpoints for bulk and interrupt transfer
 - 4 double-buffered endpoints for bulk, interrupt and isochronous transfer
 - 1,024 bytes EP_SRAM used as the endpoint data buffers
- Advanced Encryption Standard – AES-128
 - Supports AES Encrypt / Decrypt Function
 - Supports AES ECB/CBC/CTR mode
 - Supports Key Size of 128 bits
 - Supports 4 words of Initial Vector for CBC and CTR mode
 - 4 × 32 bits AES data buffer – each IN and OUT FIFO capacity
 - Supports Word Data Swap Function
 - Supports PDMA Interface
- Debug Support
 - Serial Wire Debug Port – SW-DP
 - 4 comparators for hardware breakpoint or code/literal patch
 - 2 comparators for hardware watchpoints
- Package and Operation Temperature
 - 24/33/46-pin QFN and 48-pin LQFP packages
 - Operation temperature range: -40 °C to + 85 °C

Device Information

Table 1. Features and Peripheral List

Peripherals		HT32F0008
Main Flash (KB)		63
Option Bytes Flash (KB)		1
SRAM (KB)		16
Timer	GPTM	1
	PWM	2
	BFTM	2
	RTC	1
	WDT	1
Communication	USB	1
	SPI	1
	USART	1
	UART	1
	I ² C	1
CRC-16/32		1
DIV		1
PDMA		6 channels
AES-128		1
EXTI		16
GPIO		Up to 42
CPU frequency		Up to 60 MHz
Operating voltage		1.65 V ~ 3.6 V
Operating temperature		-40 °C ~ 85 °C
Package		24/33/46-pin QFN 48-pin LQFP

Block Diagram

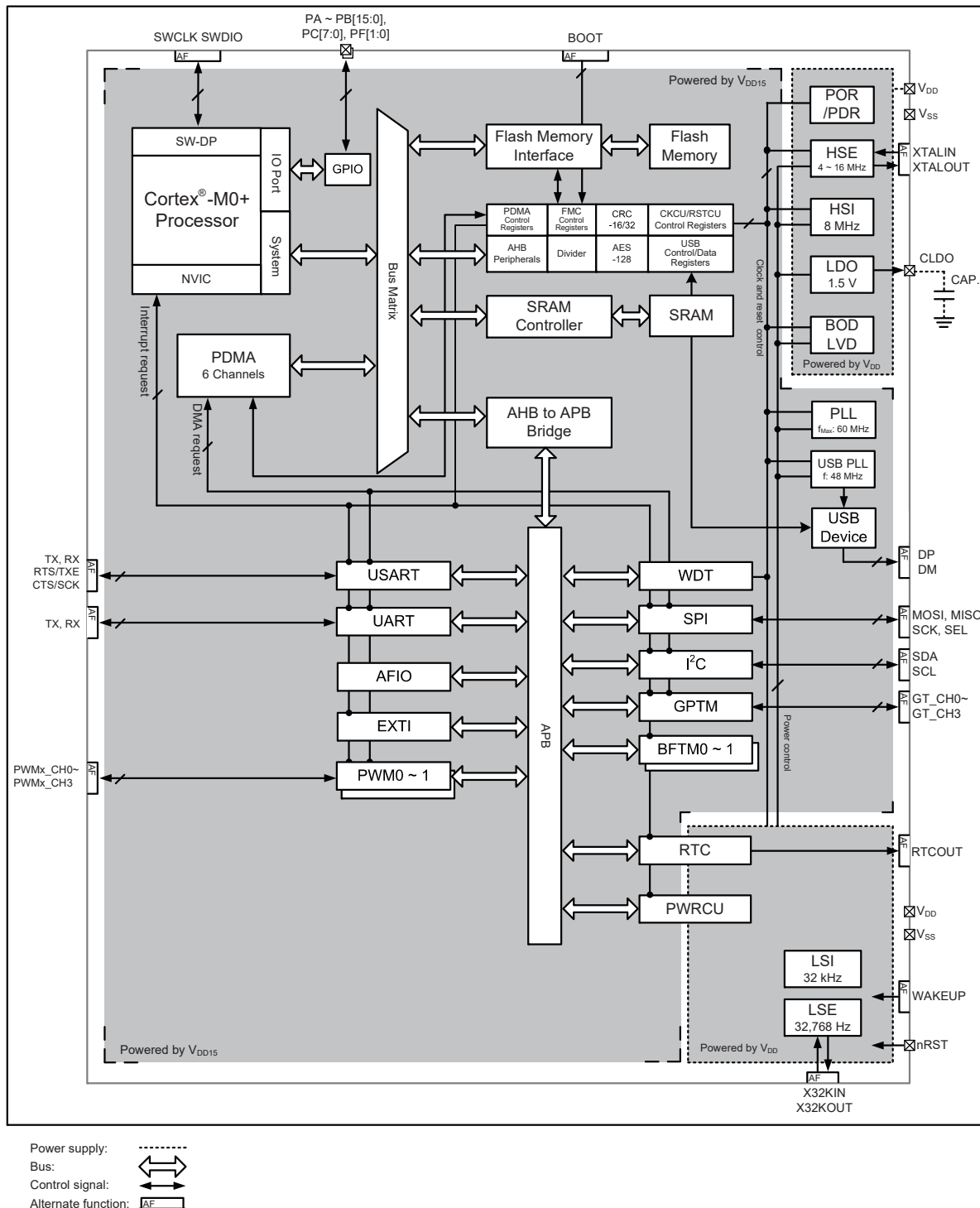


Figure 1. Block Diagram

2 Document Conventions

The conventions used in this document are shown in the following table.

Table 2. Document Conventions

Notation	Example	Description
0x	0x5a05	The number string with a 0x prefix indicates a hexadecimal number.
0xnnnn_nnnn	0x2000_0100	32-bit Hexadecimal address or data.
b	b0101	The number string with a lowercase b prefix indicates a binary number.
NAME [n]	ADDR [5]	Specific bit of NAME. NAME can be a register or field of register. For example, ADDR [5] means bit 5 of ADDR register (field).
NAME [m:n]	ADDR [11:5]	Specific bits of NAME. NAME can be a register or field of register. For example, ADDR [11:5] means bit 11 to 5 of ADDR register (field).
X	b10X1	Don't care notation which means any value is allowed.
RW	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> 19 <div style="border: 1px solid black; padding: 2px;">HSEEN</div> RW </div> <div style="text-align: center;"> 18 <div style="border: 1px solid black; padding: 2px;">PLEN</div> 0 RW 0 </div> </div>	Software can read and write to this bit.
RO	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> 3 <div style="border: 1px solid black; padding: 2px;">HSIRDY</div> RO </div> <div style="text-align: center;"> 2 <div style="border: 1px solid black; padding: 2px;">HSERDY</div> 1 RO 0 </div> </div>	Software can only read this bit. A write operation will have no effect.
RC	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> 1 <div style="border: 1px solid black; padding: 2px;">PDF</div> RC </div> <div style="text-align: center;"> 0 <div style="border: 1px solid black; padding: 2px;">PORF</div> 0 RC 1 </div> </div>	Software can only read this bit. A read operation will clear it to 0 automatically.
WC	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> 3 <div style="border: 1px solid black; padding: 2px;"></div> WC </div> <div style="text-align: center;"> 2 <div style="border: 1px solid black; padding: 2px;">CKSF</div> 0 WC 0 </div> </div>	Software can read this bit or clear it by writing 1. Writing 0 to it will have no effect.
W0C	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> 1 <div style="border: 1px solid black; padding: 2px;"></div> W0C </div> <div style="text-align: center;"> 0 <div style="border: 1px solid black; padding: 2px;">MIF</div> W0C 0 </div> </div>	Software can read this bit or clear it by writing 0. Writing 1 to it will have no effect.
WO	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> 1 <div style="border: 1px solid black; padding: 2px;">CH1CCG</div> WO </div> <div style="text-align: center;"> 0 <div style="border: 1px solid black; padding: 2px;">CH0CCG</div> 0 WO 0 </div> </div>	Software can only write to this bit. A read operation always returns 0.
Reserved	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> 7 <div style="border: 1px solid black; padding: 2px;"></div> </div> <div style="text-align: center;"> 6 <div style="border: 1px solid black; padding: 2px;">Reserved</div> </div> </div>	Reserved bit(s) for future use. Data read from these bits is not well defined and should be treated as random data. Normally these reserved bits should be set to a 0 value. Note that reserved bit must be kept at reset value.
Word		Data length of a word is 32-bit.
Half-word		Data length of a half-word is 16-bit.
Byte		Data length of a byte is 8-bit.

3 System Architecture

The system architecture of the device that includes the Arm® Cortex®-M0+ processor, bus architecture and memory organization will be described in the following sections. The Cortex®-M0+ is a next generation processor core which offers many new features. Integrated and advanced features make the Cortex®-M0+ processor suitable for market products that require microcontrollers with high performance and low power consumption. In brief, The Cortex®-M0+ processor includes the AHB-Lite bus interface. All memory accesses of the Cortex®-M0+ processor are executed on the AHB-Lite bus according to the different purposes and the target memory spaces. The memory organization uses a Harvard architecture, pre-defined memory map and up to 4 GB of memory space, making the system flexible and extendable.

Arm® Cortex®-M0+ Processor

The Cortex®-M0+ processor is a very low gate count, highly energy efficient processor that is intended for microcontroller and deeply embedded applications that require an area optimized, low power processor. The processor is based on the ARMv6-M architecture and supports Thumb® instruction sets, single-cycle I/O ports, hardware multiplier and low latency interrupt respond time. Some system peripherals listed below are also provided by Cortex®-M0+:

- Internal Bus Matrix connected with AHB-Lite Interface, Single-cycle I/O ports and Debug Accesses Port (DAP)
- Nested Vectored Interrupt Controller (NVIC)
- Optional Wakeup Interrupt Controller (WIC)
- Breakpoint and Watchpoint Unit
- Optional Memory Protection Unit (MPU)
- Serial Wire debug Port (SW-DP)
- Optional Micro Trace Buffer Interface (MTB)

The following figure shows the Cortex®-M0+ processor block diagram. For more information, refer to the Arm® Cortex®-M0+ Technical Reference Manual.

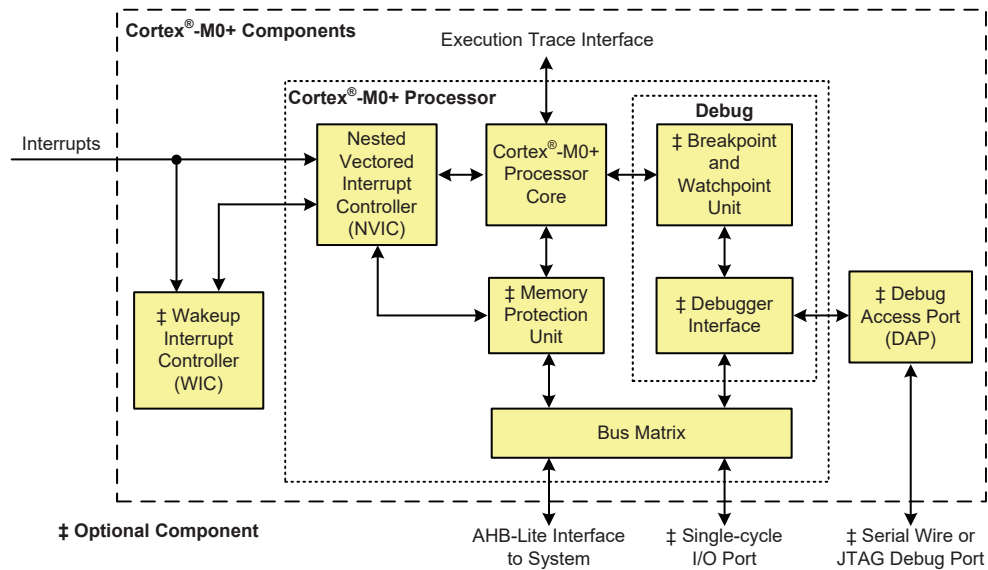


Figure 2. Cortex®-M0+ Block Diagram

Bus Architecture

The HT32F0008 device consists of one master and four slaves in the bus architecture. The Cortex®-M0+ AHB-Lite bus is the master while the internal SRAM access bus, the internal Flash memory access bus, the AHB peripherals access bus and the AHB to APB bridges are the slaves. The single 32-bit AHB-Lite system interface provides simple integration to all system regions including the internal SRAM region and the peripheral region. All of the master buses are based on 32-bit Advanced High-performance Bus-Lite (AHB-Lite) protocol. The following figure shows the bus architecture of the HT32F0008 device.

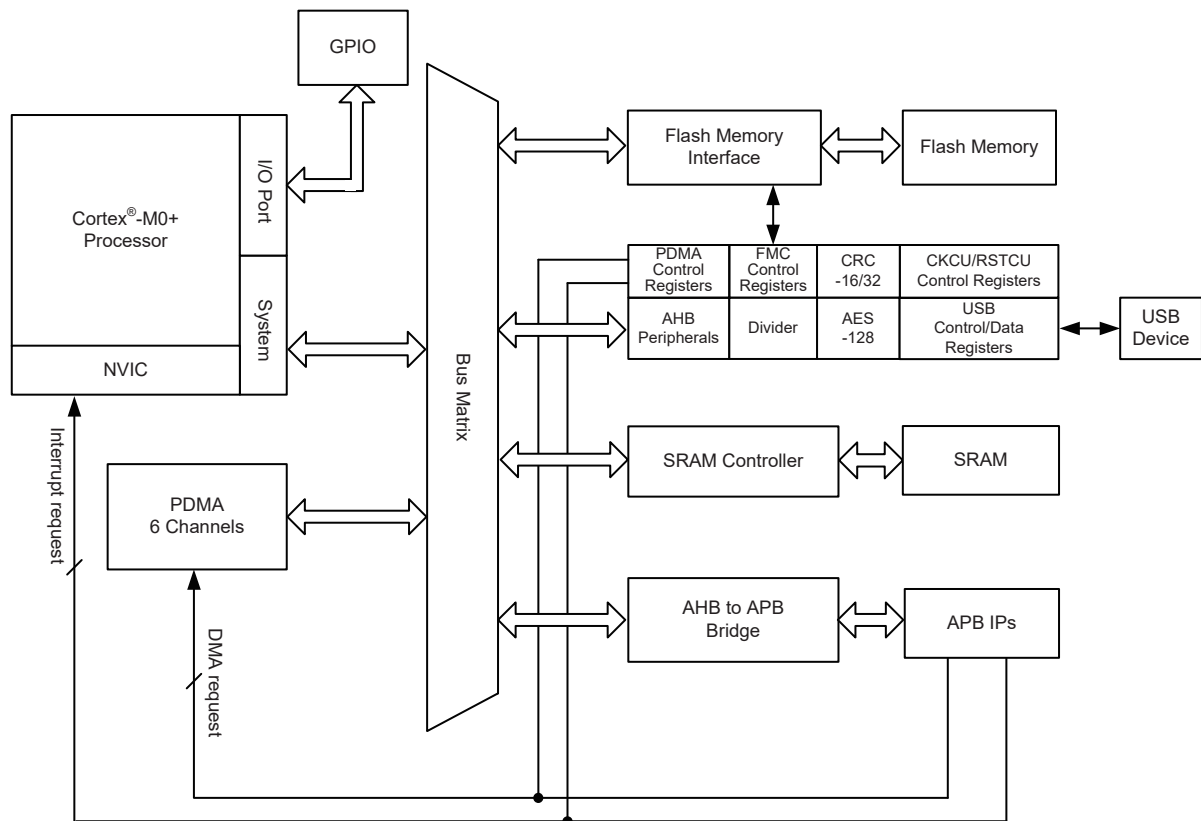


Figure 3. Bus Architecture

Memory Organization

The Arm® Cortex®-M0+ processor accesses and debug accesses share the single external interface to external AHB peripherals. The processor accesses take priority over debug accesses. The maximum address range of the Cortex®-M0+ is 4 GB since it has 32-bit bus address width. Additionally, a pre-defined memory map is provided by the Cortex®-M0+ processor to reduce the software complexity of repeated implementation of different device vendors. However, some regions are used by the Arm® Cortex®-M0+ system peripherals. Refer to the Arm® Cortex®-M0+ Technical Reference Manual for more information. The following figure shows the memory map of the HT32F0008 device, including Code, SRAM, peripheral, and other pre-defined regions.

Memory Map

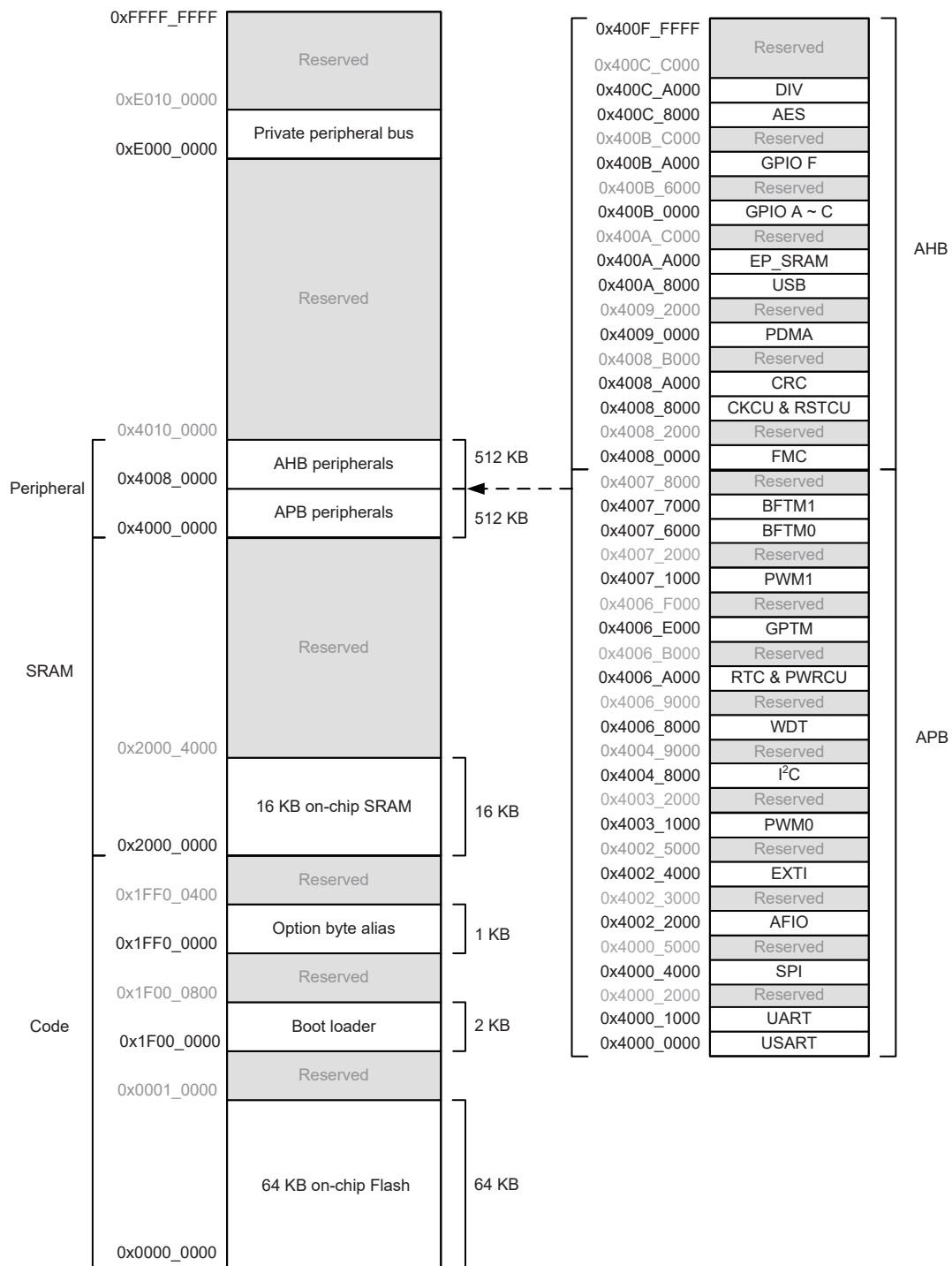


Figure 4. Memory Map

Table 3. Register Map

Start Address	End Address	Peripheral	Bus
0x4000_0000	0x4000_0FFF	USART	APB
0x4000_1000	0x4000_1FFF	UART	
0x4000_2000	0x4000_3FFF	Reserved	
0x4000_4000	0x4000_4FFF	SPI	
0x4000_5000	0x4002_1FFF	Reserved	
0x4002_2000	0x4002_2FFF	AFIO	
0x4002_3000	0x4002_3FFF	Reserved	
0x4002_4000	0x4002_4FFF	EXTI	
0x4002_5000	0x4003_0FFF	Reserved	
0x4003_1000	0x4003_1FFF	PWM0	
0x4003_2000	0x4004_7FFF	Reserved	
0x4004_8000	0x4004_8FFF	I ² C	
0x4004_9000	0x4006_7FFF	Reserved	
0x4006_8000	0x4006_8FFF	WDT	
0x4006_9000	0x4006_9FFF	Reserved	
0x4006_A000	0x4006_AFFF	RTC & PWRCU	
0x4006_B000	0x4006_DFFF	Reserved	
0x4006_E000	0x4006_EFFF	GPTM	
0x4006_F000	0x4007_0FFF	Reserved	
0x4007_1000	0x4007_1FFF	PWM1	
0x4007_2000	0x4007_5FFF	Reserved	
0x4007_6000	0x4007_6FFF	BFTM0	
0x4007_7000	0x4007_7FFF	BFTM1	
0x4007_8000	0x4007_FFFF	Reserved	
0x4008_0000	0x4008_1FFF	FMC	AHB
0x4008_2000	0x4008_7FFF	Reserved	
0x4008_8000	0x4008_9FFF	CKCU & RSTCU	
0x4008_A000	0x4008_AFFF	CRC	
0x4008_B000	0x4008_FFFF	Reserved	
0x4009_0000	0x4009_1FFF	PDMA	
0x4009_2000	0x400A_7FFF	Reserved	
0x400A_8000	0x400A_BFFF	USB	
0x400A_C000	0x400A_FFFF	Reserved	
0x400B_0000	0x400B_1FFF	GPIO A	
0x400B_2000	0x400B_3FFF	GPIO B	
0x400B_4000	0x400B_5FFF	GPIO C	
0x400B_6000	0x400B_9FFF	Reserved	
0x400B_A000	0x400B_BFFF	GPIO F	
0x400B_C000	0x400C_7FFF	Reserved	
0x400C_8000	0x400C_9FFF	AES	
0x400C_A000	0x400C_BFFF	DIV	
0x400C_C000	0x400F_FFFF	Reserved	

Embedded Flash Memory

The HT32F0008 device provides a 64 KB on-chip Flash memory which is located at address 0x0000_0000. It supports byte, half-word and word access operations. Note that the Flash memory only supports read operations for the bus access. Any write operations to the Flash memory will cause a bus fault exception. The Flash memory has 64 pages. Each page has a memory capacity of 1 KB and can be erased independently. A 32-bit programming interface provides the capability of changing bits from 1 to 0. A data storage or firmware upgrade can be implemented using several methods such as In System Programming (ISP), In Application Programming (IAP) or In Circuit Programming (ICP). For more information, refer to the Flash Memory Controller section.

Embedded SRAM Memory

This device contains a 16 KB on-chip SRAM which is located at address 0x2000_0000. It supports byte, half-word and word access operations.

AHB Peripherals

The address of the AHB peripherals ranges from 0x4008_0000 to 0x400F_FFFF. Some peripherals such as Clock Control Unit, Reset Control Unit and Flash Memory Controller are connected to the AHB bus directly. The AHB peripherals clocks are always enabled after a system reset. Access to registers for these peripherals can be achieved directly via the AHB bus. Note that all peripheral registers in the AHB bus support only word access.

APB Peripherals

The address of APB peripherals ranges from 0x4000_0000 to 0x4007_FFFF. An APB to AHB Bridge provides access capability between the CPU and the APB peripherals. Additionally, the APB peripheral clocks are disabled after a system reset. Software must enable the peripheral clocks by setting up the APBCCRn register in the Clock Control Unit before accessing the corresponding peripheral register. Note that the APB to AHB Bridge will duplicate the half-word or byte data to word width when a half-word or byte access is performed on the APB peripheral registers. In other words, the access result of a half-word or byte access on the APB peripheral register will vary depending on the data bit width of the access operation on the peripheral registers.

4 Flash Memory Controller (FMC)

Introduction

The Flash Memory Controller, FMC, provides all the necessary functions and pre-fetch buffer for the embedded on-chip Flash memory. The figure below shows the block diagram of FMC which includes programming interface, control register, pre-fetch buffer and access interface. Since the access speed of the Flash memory is slower than the CPU, a wide access interface with a pre-fetch buffer is provided to the Flash memory in order to reduce the CPU waiting time which will cause CPU instruction execution delay. The Flash memory word program/page erase functions are also provided for instruction/data storage.

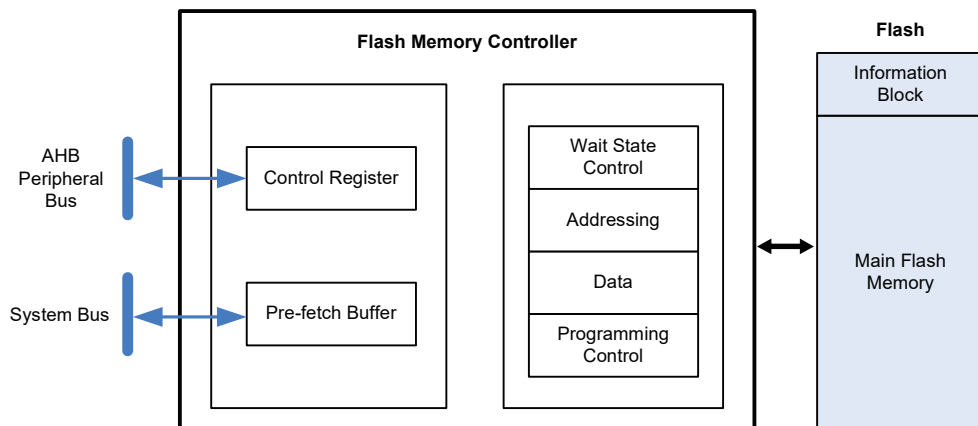


Figure 5. Flash Memory Controller Block Diagram

Features

- 64 KB of on-chip Flash memory for storing instruction/data and option bytes
 - 64 KB (instruction/data + Option Byte)
- Page size of 1 KB, totally 64 pages
- Wide access interface with a pre-fetch buffer to reduce instruction gaps
- Page erase and mass erase capability
- 32-bit word programming
- Interrupt function to indicate end of Flash memory operations or an error occurrence
- Flash read protection to prevent illegal code/data access
- Page erase/program protection to prevent unexpected operations

Functional Descriptions

Flash Memory Map

The following figure is the Flash memory map of the system. The address ranges from 0x0000_0000 to 0x1FFF_FFFF (0.5 GB). The address from 0x1F00_0000 to 0x1F00_07FF is mapped to the Boot Loader Block with a capacity of 2 KB. Additionally, the region addressed from 0x1FF0_0000 to 0x1FF0_03FF is the alias of the Option Byte block with a capacity of 1 KB, which physically locates at the last page of the main Flash. The memory mapping on system view is shown below.

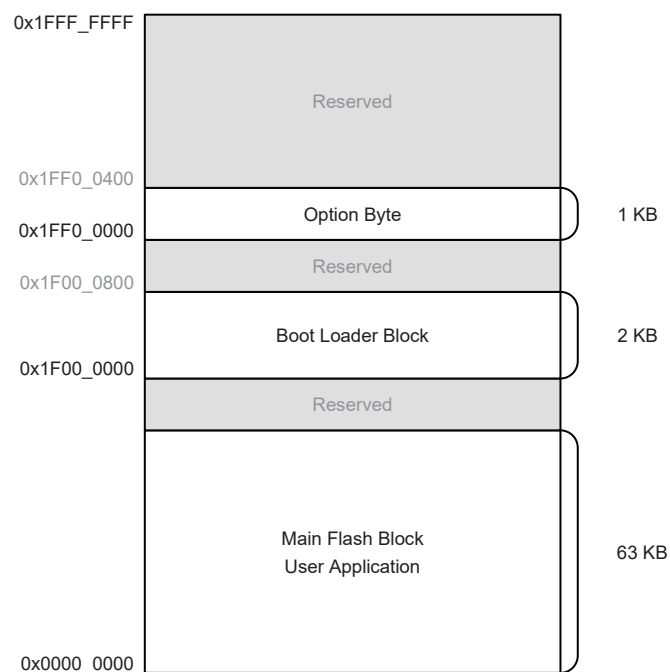


Figure 6. Flash Memory Map

Flash Memory Architecture

The Flash memory consists of 64 KB main Flash Block with 1 KB per page and 2 KB Information Block for Boot Loader. The main Flash memory contains a total of 64 pages which can be erased individually. The following table shows the base address, size, and protection setting bit of each page.

Table 4. Flash Memory and Option Byte

Block	Name	Address	Page Protection Bit	Size
Main Flash Block	Page 0	0x0000_0000 ~ 0x0000_03FF	OB_PP [0]	1 KB
	Page 1	0x0000_0400 ~ 0x0000_07FF	OB_PP [1]	1 KB
	Page 2	0x0000_0800 ~ 0x0000_0BFF	OB_PP [2]	1 KB
	Page 3	0x0000_0C00 ~ 0x0000_0FFF	OB_PP [3]	1 KB
	⋮	⋮	⋮	⋮
	Page 60	0x0000_F000 ~ 0x0000_F3FF	OB_PP [60]	1 KB
	Page 61	0x0000_F400 ~ 0x0000_F7FF	OB_PP [61]	1 KB
	Page 62	0x0000_F800 ~ 0x0000_FBFF	OB_PP [62]	1 KB
	Page 63 (Option Byte)	Physical:0x0000_FC00 ~ 0x0000_FFFF Alias: 0x1FF0_0000 ~ 0x1FF0_03FF	OB_CP [1]	1 KB
Information Block	Boot Loader	0x1F00_0000 ~ 0x1F00_07FF	NA	2 KB

Notes: 1. The Information Block stores the boot loader and this block can not be programmed or erased by users.
2. The Option Byte is always located at the last page of the Main Flash Block.

Wait State Setting

When the CPU clock, HCLK, is faster than the Flash memory access speed, the wait state cycles must be inserted during CPU fetching instructions or loading data from the Flash memory. The wait state can be changed by setting the WAIT [2:0] field of the Flash Pre-fetch Control Register, CFCR. In order to meet the wait state requirement, the following two rules should be considered.

- The HCLK clock is switched from low to high frequency:
Change the wait state setting first and then switch the HCLK clock.
- The HCLK clock is switched from high to low frequency:
Switch the HCLK clock first and then change the wait state setting.

The following table shows the relationship between the wait state cycle and HCLK. The default wait state is 0 since the High Speed Internal oscillator, HSI, which operates at a frequency of 8 MHz is selected as the HCLK clock source after a system reset.

Table 5. Relationship between Wait State Cycle and HCLK

Wait State Cycle	HCLK
0	0 MHz < HCLK ≤ 20 MHz
1	20 MHz < HCLK ≤ 40 MHz
2	40 MHz < HCLK ≤ 60 MHz

Booting Configuration

The system provides two kinds of booting modes which can be selected using the BOOT pin. The BOOT pin status is sampled during the power-on reset or system reset. Once the logic value is decided, the first 4 words of vector will be remapped to the corresponding source according to the booting mode. The booting modes are shown in the following table.

Table 6. Booting Modes

Booting Mode Selection Pin BOOT	Mode	Descriptions
0	Boot Loader	The vector source is Boot Loader
1	Main Flash	The vector source is main Flash

The Flash Vector Mapping Control Register, VMCR, is provided to change the vector remapping setting temporarily after the chip reset. The initial reset value of the VMCR register is determined by the BOOT pin status which will be sampled during the reset duration.

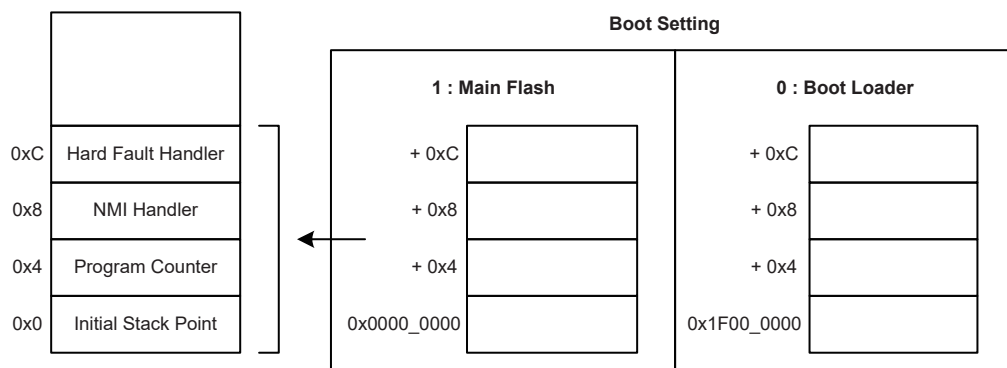


Figure 7. Vector Remapping

Page Erase

The FMC provides a page erase function which is used to reset partial content of the Flash memory. Any page can be erased independently without affecting others. The following steps show the page erase operation register access sequence.

1. Check the OPCR register to confirm that no Flash memory operation is in progress (OPM [3:0] is equal to 0xE or 0x6). Otherwise, wait until the previous operation has been finished.
2. Write the page address to the TADR register.
3. Write the page erase command to the OCMR register (Set CMD [3:0] = 0x8).
4. Commit the page erase command to the FMC by setting the OPCR register (Set OPM [3:0] = 0xA).
5. Wait until all the operations have been completed by checking the value of the OPCR register (OPM [3:0] is equal to 0xE).
6. Read and verify the page if required.

Note that a correct address of the target page must be confirmed. The software may run out of control if the target erase page is under the code fetching or data accessing status. The FMC will not provide any notification when this happens. Additionally, the page erase operation will be ignored on the protected pages. When this occurs, the OREF bit will be set by the FMC and then a Flash Operation Error interrupt will be generated if the OREIEN bit in the OIER register is set. The software can check the PPEF bit in the OISR register to detect this condition in the interrupt handler. The following figure shows the page erase operation flow.

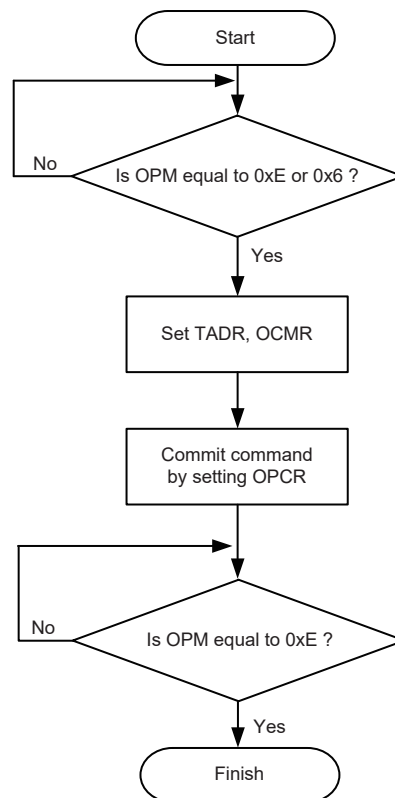


Figure 8. Page Erase Operation Flowchart

Mass Erase

The FMC provides a mass erase function which is used to initialize all the main Flash memory contents to a high state. The following steps show the mass erase operation register access sequence.

1. Check the OPCR register to confirm that no Flash memory operation is in progress (OPM [3:0] is equal to 0xE or 0x6). Otherwise, wait until the previous operation has been finished.
2. Write the mass erase command to the OCMR register (Set CMD [3:0] = 0xA).
3. Commit the mass erase command to the FMC by setting the OPCR register (Set OPM [3:0] = 0xA).
4. Wait until all the operations have been finished by checking the value of the OPCR register (OPM [3:0] is equal to 0xE).
5. Read and verify the Flash memory if required.

Since all Flash data will be reset as 0xFFFF_FFFF, the mass erase operation can be implemented by the program that runs on the SRAM or by the debugging tool that accesses the FMC registers directly. The application program that is executed on the Flash memory will not trigger a mass erase operation. The following figure shows the mass erase operation flow.

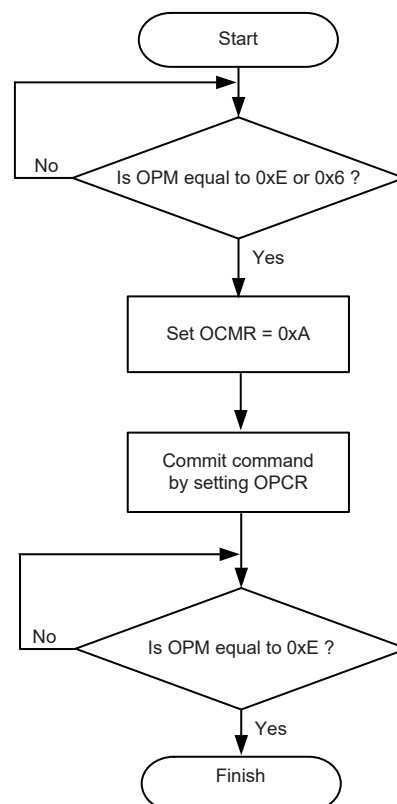


Figure 9. Mass Erase Operation Flowchart

Word Programming

The FMC provides a 32-bit word programming function which is used to modify the Flash memory contents. The following steps show the word programming operation register access sequence.

1. Check the OPCR register to confirm that no Flash memory operation is in progress (OPM [3:0] is equal to 0xE or 0x6). Otherwise, wait until the previous operation has been finished.
2. Write the word address to the TADR register. Write the word data to the WRDR register.
3. Write the word programming command to the OCMR register (Set CMD [3:0] = 0x4).
4. Commit the word programming command to the FMC by setting the OPCR register (Set OPM [3:0] = 0xA).
5. Wait until all the operations have been finished by checking the value of the OPCR register (OPM [3:0] is equal to 0xE).
6. Read and verify the Flash memory if required.

Note that the word programming operation can not be successively applied to the same address twice. Successive word programming operation to the same address must be separated by a page erase operation. Additionally, the word programming operation will be ignored on the protected pages. When this occurs, the OREF bit will be set by the FMC and then a Flash Operation Error interrupt will be generated if the OREIEN bit in the OIER register is set. The software can check the PPEF bit in the OISR register to detect this condition in the interrupt handler. The following figure shows the word programming operation flow.

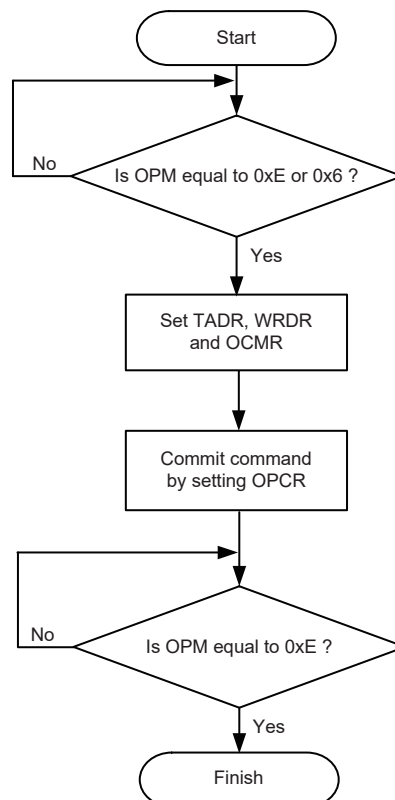


Figure 10. Word Programming Operation Flowchart

Option Byte Description

The Option Byte area can be treated as an independent Flash memory of which the base address is 0x1FF0_0000. The following table shows the functional description and the Option Byte memory map.

Table 7. Option Byte Memory Map

Option Byte	Offset	Description	Reset Value
Option Byte Base Address = 0x1FF0_0000			
OB_PP	0x000 0x004 0x008 0x00C	Flash Page Erase / Program Protection (Page 127 ~ Page 0) OB_PP [n] (n = 0 ~ 62) 0: Flash Page n Erase / Program Protection is enabled 1: Flash Page n Erase / Program Protection is disabled OB_PP [n] (n = 63 ~ 127) Reserved	0xFFFF_FFFF 0xFFFF_FFFF 0xFFFF_FFFF 0xFFFF_FFFF
OB_CP	0x010	Flash Security Protection OB_CP [0] 0: Flash Security protection is enabled 1: Flash Security protection is disabled Option Byte Protection OB_CP [1] 0: Option Byte protection is enabled 1: Option Byte protection is disabled OB_CP [31:2]: Reserved	0xFFFF_FFFF
OB_CK	0x020	Flash Option Byte Checksum OB_CK [31:0] OB_CK should be set as the sum of 5 words of Option Byte content, of which the offset address ranges from 0x000 to 0x010 (0x000 + 0x004 + 0x008 + 0x00C + 0x010), when the OB_PP or OB_CP register content is not equal to 0xFFFF_FFFF. Otherwise, both page erase/program protection and security protection will be enabled.	0xFFFF_FFFF
OB_TOOL	0x024 ~ 0x04C	Reserved for Flash writer tool and boot loader.	0xFFFF_FFFF
OB_WDT	0x3F0	Flash Option Watchdog Timer Enable OB_WDT [15:0]: 0x7A92 If the OB_WDT [15:0] is set to 0x7A92, the WDT will be enabled immediately when the MCU power on reset or system reset occurs. The WDT can be disabled by software. OB_WDT [31:16]: Reserved	0xFFFF_FFFF

Page Erase/Program Protection

The FMC provides a page erase/program protection function to prevent unexpected operations on the protected Flash memory area. The page erase (CMD [3:0] = 0x8 in the OCMR register) or word programming (CMD [3:0] = 0x4) command will not be accepted by the FMC on the protected pages. When the page erase or word programming command aimed at the protected pages is sent to the FMC, the PPEF bit in the OISR register will then be set by the FMC and the Flash Operation Error interrupt will be triggered to inform the CPU if the OREIEN bit in the OIER register is set. The page protection function can be individually enabled for each page by configuring the OB_PP registers in the Option Byte area. The following table shows the access permission of the main Flash page when the page protection is enabled.

Table 8. Access Permission of Protected Main Flash Page

Operation \ Mode	ISP/IAP	ICP/Debug Mode
Read	O	O
Program	X	X
Page Erase	X	X
Mass Erase	O	O

- Notes:**
1. Each write protection bit setting is for one specific page. The above access permission only affects the pages of which the protection function has been enabled. Other pages are not affected.
 2. The main Flash page protection is configured by OB_PP [127:0]. Option Byte is physically located at the last page of the main Flash. The Option Byte page protection is configured by the OB_CP [1] bit.
 3. The page erase operation on the Option Byte area can disable the page protection of the main Flash.
 4. The page protection of the Option Byte can only be disabled by a mass erase operation.

The following steps show the register access sequence for the page erase/program protection procedure.

1. Check the OPCR register to confirm that no Flash memory operation is in progress (OPM [3:0] is equal to 0xE or 0x6). Otherwise, wait until the previous operation has been finished.
2. Write the OB_PP address to the TADR register (Set TADR = 0x1FF0_0000).
3. Write the desired data which indicates the protection function of the corresponding page is to be enabled or disabled, into the WRDR register (0: Enabled, 1: Disabled).
4. Write the word programming command to the OCMR register (Set CMD [3:0] = 0x4).
5. Commit the word programming command to the FMC by setting the OPCR register (Set OPM [3:0] = 0xA).
6. Wait until all the operations have been finished by checking the value of the OPCR register (OPM [3:0] is equal to 0xE).
7. Read and verify the Option Byte if required.
8. The OB_CK field in the Option Byte area must be updated according to the Option Byte checksum rule.
9. Apply a system reset to activate the new OB_PP setting.

Security Protection

The FMC provides a security protection function to prevent illegal code/data access to the Flash memory. This function is useful for protecting the software/firmware from illegal users. The function is activated by setting OB_CP [0] in the Option Byte. Once the function has been enabled, all the main Flash data access through ICP/Debug mode, programming and page erase operation will not be allowed except the user's application. However the mass erase operation will still be accepted by the FMC in order to disable this security protection function. The following table shows the access permission of the Flash memory when the security protection is enabled.

Table 9. Access Permission When Security Protection is Enabled

Operation \ Mode	User Application ⁽¹⁾	ICP/Debug Mode
Read	O	X (read as 0)
Program	O ⁽¹⁾	X
Page Erase	O ⁽¹⁾	X
Mass Erase	O	O

Notes: 1. User application means the software that is executed or booted from the main Flash memory with the JTAG/SW debugger being disconnected. However, the Option Byte area and page 0 are still under protection, where the Program/Page Erase operations are not accepted.
2. The Mass Erase operation can erase the Option Byte area and disable the security protection.

The following steps show the register access sequence for the security protection procedure.

1. Check the OPCR register to confirm that no Flash memory operation is in progress (OPM [3:0] is equal to 0xE or 0x6). Otherwise, wait until the previous operation has been finished.
2. Write the OB_CP address to the TADR register (Set TADR = 0x1FF0_0010).
3. Write data to the WRDR register to set OB_CP [0] to 0.
4. Write the word program command to the OCMR register (Set CMD [3:0] = 0x4).
5. Commit the word program command to the FMC by setting the OPCR register (Set OPM = 0xA).
6. Wait until all the operations have been finished by checking the value of the OPCR register (OPM [3:0] is equal to 0xE).
7. Read and verify the Option Byte if required.
8. The OB_CK field in the Option Byte area must be updated according to the Option Byte checksum rule.
9. Apply a system reset to active the new OB_CP setting.

Register Map

The following table shows the FMC registers and reset values.

Table 10. FMC Register Map

Register	Offset	Description	Reset Value
TADR	0x000	Flash Target Address Register	0x0000_0000
WRDR	0x004	Flash Write Data Register	0x0000_0000
OCMR	0x00C	Flash Operation Command Register	0x0000_0000
OPCR	0x010	Flash Operation Control Register	0x0000_000C
OIER	0x014	Flash Operation Interrupt Enable Register	0x0000_0000
OISR	0x018	Flash Operation Interrupt and Status Register	0x0001_0000
PPSR	0x020 0x024 0x028 0x02C	Flash Page Erase/Program Protection Status Register	0xFFFF_XXXX 0xFFFF_XXXX 0xFFFF_XXXX 0xFFFF_XXXX
CPSR	0x030	Flash Security Protection Status Register	0x0000_000X
VMCR	0x100	Flash Vector Mapping Control Register	0x0000_000X
MDID	0x180	Flash Manufacturer and Device ID Register	0x0376_XXXX
PNSR	0x184	Flash Page Number Status Register	0x0000_00XX
PSSR	0x188	Flash Page Size Status Register	0x0000_0400
DID	0x18C	Device ID Register	0x000X_XXXX
CFCR	0x200	Flash Pre-fetch Control Register	0x0000_0011
CIDR0	0x310	Custom ID Register 0	0xFFFF_XXXX
CIDR1	0x314	Custom ID Register 1	0xFFFF_XXXX
CIDR2	0x318	Custom ID Register 2	0xFFFF_XXXX
CIDR3	0x31C	Custom ID Register 3	0xFFFF_XXXX

Note: “X” means various reset values which depend on the Device, Flash value, Option Byte value or power on reset setting.

Register Descriptions

Flash Target Address Register – TADR

This register specifies the target address of the page erase and word programming operations.

Offset: 0x000

Reset value: 0x0000_0000

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

Bits	Field	Descriptions
[31:0]	TADB	<p>Flash Target Address Bits</p> <p>For programming operations, the TADR register specifies the address where the data is written to. Since the programming length is 32-bit, the TADR register should be set as word-aligned (4 bytes). The TADB [1:0] bits will be ignored during programming operations. For page erase operations, the TADR register contains the page address which is to be erased. Since the page size is 1 KB, the TADB [9:0] bits will be ignored in order to limit the target address as 1 Kbyte-aligned. For 64 KB main Flash addressing, the TADB [31:16] bits should be zero. The region of which the address ranges from 0x1FF0_0000 to 0x1FF0_03FF is the 1 KB Option Byte. This field for available Flash address must be within the range of 0x0000_0000 to 0x1FFF_FFFF. Otherwise, an Invalid Target Address interrupt will be generated if the corresponding interrupt enable bit is set.</p>

Flash Write Data Register – WRDR

This register specifies the data to be written for the programming operation.

Offset: 0x004

Reset value: 0x0000_0000

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

Bits	Field	Descriptions
[31:0]	WRDB	Flash Write Data Bits The data value for the programming operation.

Flash Operation Command Register – OCMR

This register is used to specify the Flash operation commands that include word programming, page erase and mass erase.

Offset: 0x00C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved				CMD			
					RW	0	RW	0
						RW	0	RW
							RW	0
								RW
								0

Bits	Field	Descriptions												
[3:0]	CMD	<p>Flash Operation Command</p> <p>The following table shows the definitions of the operation command field, CMD [3:0], which specifies the Flash memory operation. If an invalid command is set and the IOCMIEN bit is set to 1, an Invalid Operation Command interrupt will be generated.</p> <table><tr><th>CMD [3:0]</th><th>Description</th></tr><tr><td>0x0</td><td>Idle (default)</td></tr><tr><td>0x4</td><td>Word programming</td></tr><tr><td>0x8</td><td>Page erase</td></tr><tr><td>0xA</td><td>Mass erase</td></tr><tr><td>Others</td><td>Reserved</td></tr></table>	CMD [3:0]	Description	0x0	Idle (default)	0x4	Word programming	0x8	Page erase	0xA	Mass erase	Others	Reserved
CMD [3:0]	Description													
0x0	Idle (default)													
0x4	Word programming													
0x8	Page erase													
0xA	Mass erase													
Others	Reserved													

Flash Operation Control Register – OPCR

This register is used for controlling the command commitment and checking the status of the FMC operations.

Offset: 0x010

Reset value: 0x0000_000C

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved			OPM				Reserved
	RW			0	RW	1	RW	1
							RW	0

Bits	Field	Descriptions										
[4:1]	OPM	<p>Operation Mode</p> <p>The following table shows the FMC operation modes. Users can commit command which is set by the OCMR register to the FMC according to the address alias setting in the TADR register. The contents of the TADR, WRDR and OCMR registers should be prepared before setting this register. After all the operations have been finished, the OPM field will be set to 0xE by the FMC hardware. The Idle mode can be set when all the operations have been finished for power saving purpose. Note that the operation status should be checked before executing next operation. The contents of the TADR, WRDR, OCMR and OPCR registers should not be changed until the previous operation has been finished.</p> <table><tr><th>OPM [3:0]</th><th>Description</th></tr><tr><td>0x6</td><td>Idle (default)</td></tr><tr><td>0xA</td><td>Commit command to main Flash</td></tr><tr><td>0xE</td><td>All operation finished on main Flash</td></tr><tr><td>Others</td><td>Reserved</td></tr></table>	OPM [3:0]	Description	0x6	Idle (default)	0xA	Commit command to main Flash	0xE	All operation finished on main Flash	Others	Reserved
OPM [3:0]	Description											
0x6	Idle (default)											
0xA	Commit command to main Flash											
0xE	All operation finished on main Flash											
Others	Reserved											

Flash Operation Interrupt Enable Register – OIER

This register is used to enable or disable the FMC interrupt function. The FMC will generate the interrupt when the corresponding interrupt enable bit is set and the interrupt condition occurs.

Offset: 0x014

Reset value: 0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
Type/Reset							
23	22	21	20	19	18	17	16
Reserved							
Type/Reset							
15	14	13	12	11	10	9	8
Reserved							
Type/Reset							
7	6	5	4	3	2	1	0
Reserved			OREIEN	IOCMIEN	OBEIEN	ITADIEN	ORFIEN
Type/Reset			RW	0 RW	0 RW	0 RW	0 RW

Bits	Field	Descriptions
[4]	OREIEN	Operation Error Interrupt Enable 0: Operation Error Interrupt is disabled 1: Operation Error Interrupt is enabled
[3]	IOCMIEN	Invalid Operation Command Interrupt Enable 0: Invalid Operation Command Interrupt is disabled 1: Invalid Operation Command Interrupt is enabled
[2]	OBEIEN	Option Byte Check Sum Error Interrupt Enable 0: Option Byte Check Sum Error Interrupt is disabled 1: Option Byte Check Sum Error Interrupt is enabled
[1]	ITADIEN	Invalid Target Address Interrupt Enable 0: Invalid Target Address Interrupt is disabled 1: Invalid Target Address Interrupt is enabled
[0]	ORFIEN	Operation Finished Interrupt Enable 0: Operation Finished Interrupt is disabled 1: Operation Finished Interrupt is enabled

4 Flash Memory Controller (FMC)

Offset:	0x018
---------	-------

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved						PPEF	RORFF
							RO 0	RO 1
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved			OREF	IOCMF	OBEF	ITADF	ORFF
				WC 0	WC 0	WC 0	WC 0	WC 0

Rev. 1.20 50 of 460 November 22, 2019

Bits	Field	Descriptions
[2]	OBEF	<p>Option Byte Checksum Error Flag</p> <p>0: Option Byte checksum is correct 1: Option Byte checksum is incorrect</p> <p>This bit will be set high when the Option Byte checksum is incorrect. The Option Byte Checksum Error interrupt will be generated if the OBEIEN bit in the OIER register is set. This bit is cleared to zero by software writing “1” into it. However, the Option Byte Checksum Error Flag can not be cleared by software until the interrupt condition is released, which means that the Option Byte checksum value has been correctly modified or the corresponding interrupt control is disabled. Otherwise, the interrupt will be continually generated.</p>
[1]	ITADF	<p>Invalid Target Address Flag</p> <p>0: The target address is valid 1: The target address is invalid</p> <p>The data in the TADR field must be within the range from 0x0000_0000 to 0x1FFF_FFFF. Otherwise, this bit will be set high and an Invalid Target Address interrupt will be generated if the ITADIEN bit in the OIER register is set. Reset this bit by writing 1.</p>
[0]	ORFF	<p>Operation Finished Flag</p> <p>0: Last Flash operation has not finished 1: Last Flash operation has finished</p> <p>This bit will be set when the last Flash operation has finished. Then the Operation Finished interrupt will be generated if the ORFIEN bit in the OIER register is set. Reset this bit by writing 1.</p>

Flash Page Erase/Program Protection Status Register – PPSR

This register indicates the page protection status of the Flash page erase/program protection functions.

Offset: 0x020 (0) ~ 0x02C (3)

Reset value: 0xFFFF_FFFF

	31	30	29	28	27	26	25	24	
	PPSBn								
Type/Reset	RO	X	RO	X	RO	X	RO	X	RO
	23	22	21	20	19	18	17	16	
	PPSBn								
Type/Reset	RO	X	RO	X	RO	X	RO	X	RO
	15	14	13	12	11	10	9	8	
	PPSBn								
Type/Reset	RO	X	RO	X	RO	X	RO	X	RO
	7	6	5	4	3	2	1	0	
	PPSBn								
Type/Reset	RO	X	RO	X	RO	X	RO	X	RO

Bits	Field	Descriptions
[127:0]	PPSBn	<p>Page Erase/Program Protection Status Bits (n = 0 ~ 127)</p> <p>PPSB[n] = OB_PP[n]</p> <p>0: The corresponding page is protected</p> <p>1: The corresponding page is not protected</p> <p>The content of this register is not dynamically updated and will only be reloaded from the Option Byte when any kind of reset occurs. The erase or program function of the specific pages is not allowed when the corresponding bits of the PPSR registers are reset. The reset value of PPSR [127:0] is determined by the Option Byte OB_PP [127:0] bits. Since the maximum page number of the main Flash is various and dependent on the chip specification. Therefore, the every page erase/program protection status bit may protect one or two pages, depending on the chip specification. Other bits of the OB_PP and PPSR registers are reserved for future use.</p>

Flash Security Protection Status Register – CPSR

This register indicates the Flash memory security protection status. The content of this register is not dynamically updated and will only be reloaded by the Option Byte loader which is activated when any kind of reset occurs.

Offset: 0x030

Reset value: 0x0000_000X

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved						OBPSB	CPSB
							RO	X RO X

Bits	Field	Descriptions
[1]	OBPSB	Option Byte Page Erase/Program Protection Status Bit 0: The Option Byte page is protected 1: The Option Byte page is not protected The reset value of the OBPSB bit is determined by the Option Byte OB_CP [1] bit.
[0]	CPSB	Flash Security Protection Status Bit 0: Flash Security protection is enabled 1: Flash Security protection is not enabled The reset value of the CPSB bit is determined by the Option Byte OB_CP [0] bit.

Flash Vector Mapping Control Register – VMCR

This register is used to control the vector mapping. The reset value of the VMCR register is determined by the external booting pin, BOOT, during the power-on reset period.

Offset: 0x100

Reset value: 0x0000_000X

31	30	29	28	27	26	25	24
Reserved							
Type/Reset							
23	22	21	20	19	18	17	16
Reserved							
Type/Reset							
15	14	13	12	11	10	9	8
Reserved							
Type/Reset							
7	6	5	4	3	2	1	0
Reserved						VMCB	Reserved
Type/Reset						RW	X

Bits	Field	Descriptions									
[1]	VMCB	<p>Vector Mapping Control Bit</p> <p>The VMCB bit is used to control the mapping source of the first 4 words of vector addressed from 0x0 to 0xC. The following table shows the vector mapping setting.</p> <table> <tr> <th>BOOT</th><th>VMCB</th><th>Descriptions</th></tr> <tr> <td>Low</td><td>0</td><td>Boot Loader mode The vector mapping source is the boot loader area.</td></tr> <tr> <td>High</td><td>1</td><td>Main Flash mode The vector mapping source is the main Flash area.</td></tr> </table> <p>The reset value of the VMCB bit is determined by the pin status of the external BOOT pin during power-on reset and system reset. The vector mapping setting can be changed temporarily by configuring the VMCB bit when the application program is executed.</p>	BOOT	VMCB	Descriptions	Low	0	Boot Loader mode The vector mapping source is the boot loader area.	High	1	Main Flash mode The vector mapping source is the main Flash area.
BOOT	VMCB	Descriptions									
Low	0	Boot Loader mode The vector mapping source is the boot loader area.									
High	1	Main Flash mode The vector mapping source is the main Flash area.									

Flash Manufacturer and Device ID Register – MDID

This register specifies the manufacture ID and device part number information which can be used as the product identity.

Offset: 0x180

Reset value: 0x0376_XXXX

	31	30	29	28	27	26	25	24				
	MFID											
Type/Reset	RO	0	RO	0	RO	0	RO	0	RO	1	RO	1
	23	22	21	20	19	18	17	16				
	MFID											
Type/Reset	RO	0	RO	1	RO	1	RO	1	RO	1	RO	0
	15	14	13	12	11	10	9	8				
	ChipID											
Type/Reset	RO	X	RO	X	RO	X	RO	X	RO	X	RO	X
	7	6	5	4	3	2	1	0				
	ChipID											
Type/Reset	RO	X	RO	X	RO	X	RO	X	RO	X	RO	X

Bits	Field	Descriptions
[31:16]	MFID	Manufacturer ID Read as 0x0376.
[15:0]	ChipID	Chip ID The last 4 digital codes of the MCU device part number.

Flash Page Number Status Register – PNSR

This register indicates the Flash memory page number.

Offset: 0x184

Reset value: 0x0000_00XX

	31	30	29	28	27	26	25	24	
	PNSB								
Type/Reset	RO	0	RO	0	RO	0	RO	0	RO
	23	22	21	20	19	18	17	16	
	PNSB								
Type/Reset	RO	0	RO	0	RO	0	RO	0	RO
	15	14	13	12	11	10	9	8	
	PNSB								
Type/Reset	RO	0	RO	0	RO	0	RO	0	RO
	7	6	5	4	3	2	1	0	
	PNSB								
Type/Reset	RO	X	RO	X	RO	X	RO	X	RO

Bits	Field	Descriptions
[31:0]	PNSB	Flash Page Number Status Bits 0x0000_0010: Totally 16 pages for the on-chip Flash memory device 0x0000_0020: Totally 32 pages for the on-chip Flash memory device 0x0000_0040: Totally 64 pages for the on-chip Flash memory device 0x0000_0080: Totally 128 pages for the on-chip Flash memory device 0x0000_00FF: Totally 255 pages for the on-chip Flash memory device

Flash Page Size Status Register – PSSR

This register indicates the page size in bytes.

Offset: 0x188

Reset value: 0x0000_0400

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

Bits	Field	Descriptions
[31:0]	PSSB	Flash Page Size Status Bits 0x200: The page size is 512 Bytes per page 0x400: The page size is 1 K Bytes per page 0x800: The page size is 2 K Bytes per page

Device ID Register – DID

This register specifies the device part number information which can be used as the product identity.

Offset: 0x18C

Reset value: 0x000X_XXXX

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved				ChipID			
Type/Reset					RO	X	RO	X
	15	14	13	12	11	10	9	8
	ChipID							
Type/Reset	RO	X	RO	X	RO	X	RO	X
	7	6	5	4	3	2	1	0
	ChipID							
Type/Reset	RO	X	RO	X	RO	X	RO	X

Bits	Field	Descriptions
[19:0]	ChipID	Chip ID The complete 5 digital codes of the MCU device part number.

Flash Pre-fetch Control Register – CFCR

This register is used for controlling the FMC pre-fetch module.

Offset: 0x200

Reset value: 0x0000_0011

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved			PFBE	Reserved	WAIT		
				RW	1	RW	0	RW
							0	RW
								1

Bits	Field	Descriptions
[4]	PFBE	Pre-fetch Buffer Enable Bit 0: Pre-fetch buffer is disabled 1: Pre-fetch buffer is enabled
[2:0]	WAIT	Flash Wait State Setting The WAIT[2:0] field is used to set the HCLK wait clock during a non-sequential Flash access. The actual wait clock is given by (WAIT[2:0] - 1). Since a wide access interface with a pre-fetch buffer is provided, the wait state of sequential Flash access is very close to zero.

WAIT [2:0]	Wait Status	Allowed HCLK Range
001	0	0 MHz < HCLK ≤ 20 MHz
010	1	20 MHz < HCLK ≤ 40 MHz
011	2	40 MHz < HCLK ≤ 60 MHz
Others	Reserved	Reserved

Custom ID Register n – CIDRn (n = 0 ~ 3)

This register specifies the custom ID information which can be used as the custom identity.

Offset: 0x310 (0) ~ 0x31C (3)

Reset value: 0xFFFF_FFFF – Various depending on Flash Manufacture Privilege Information Block

	31	30	29	28	27	26	25	24	
	CID								
Type/Reset	RO	X RO	X RO	X RO	X RO	X RO	X RO	X RO	X
	23	22	21	20	19	18	17	16	
	CID								
Type/Reset	RO	X RO	X RO	X RO	X RO	X RO	X RO	X RO	X
	15	14	13	12	11	10	9	8	
	CID								
Type/Reset	RO	X RO	X RO	X RO	X RO	X RO	X RO	X RO	X
	7	6	5	4	3	2	1	0	
	CID								
Type/Reset	RO	X RO	X RO	X RO	X RO	X RO	X RO	X RO	X

Bits	Field	Descriptions
[31:0]	CIDn	Custom ID Read as the CIDn[31:0] (n = 0 ~ 3) field in the Custom ID registers in Flash Manufacture Privilege Block.

5 Power Control Unit (PWRCU)

Introduction

The power consumption can be regarded as one of the most important issues for many embedded system applications. Accordingly the Power Control Unit, PWRCU, provides many types of power saving modes such as Sleep, Deep-Sleep1, Deep-Sleep2 and Power-Down modes. These modes reduce the power consumption and allow the application to achieve the best trade-off between the conflicting demands of CPU operating time, speed and power consumption. The dash line in the Figure 11 indicates the power supply source of two digital power domains.

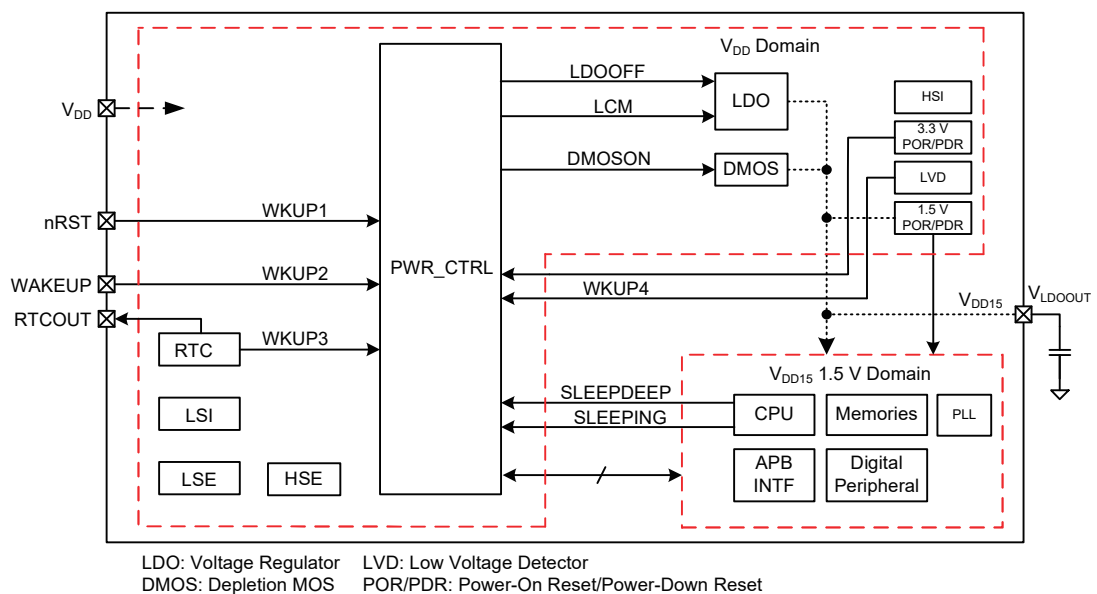


Figure 11. PWRCU Block Diagram

Features

- Two power domains: V_{DD} 3.3 V and V_{DD15} 1.5 V power domains
- Four power saving modes: Sleep, Deep-Sleep1, Deep-Sleep2 and Power-Down modes
- Internal Voltage regulator supplies 1.5 V voltage source
- Additional Depletion MOS supplies 1.5 V voltage source with low leakage and low operating current
- A power reset is generated when one of the following events occurs:
 - Power-on / Power-down reset (POR / PDR reset)
 - When exiting Power-Down mode
 - The control bits $BODEN = 1$, $BODRIS = 0$ and the supply power $V_{DD} \leq V_{BOD}$
- The Brown-Out Detector can issue a system reset or an interrupt when V_{DD} power source is lower than the Brown-Out Detector voltage V_{BOD} .
- The Low Voltage Detector can issue an interrupt or wakeup event when V_{DD} is lower than a programmable threshold voltage V_{LVD} .

Functional Descriptions

V_{DD} Power Domain

LDO Power Control

The LDO will be automatically switched off when one of the following conditions occurs:

- The Power-Down or Deep-Sleep2 mode is entered
- The control bits $BODEN = 1$, $BODRIS = 0$ and the supply power $V_{DD} \leq V_{BOD}$
- The supply power $V_{DD} \leq V_{PDR}$

The LDO will be automatically switched on by hardware when the supply power $V_{DD} > V_{POR}$ if any of the following conditions occurs:

- Resume operation from the power saving mode – RTC wakeup, LVD wakeup and WAKEUP pin rising edge
- Detect a falling edge on the external reset pin (nRST)
- The control bit $BODEN = 1$ and the supply power $V_{DD} > V_{BOD}$

To enter the Deep-Sleep1 mode, the PWRCU will request the LDO to operate in the low current mode, LCM. To enter the Deep-Sleep2 mode, the PWRCU will turn off the LDO and turn on the DMOS to supply an alternative 1.5 V power.

Voltage Regulator

The voltage regulator, LDO, Depletion MOS, DMOS, Low voltage Detector, LVD, High Speed Internal oscillator, HSI, Low Speed Internal RC oscillator, LSI, High Speed External Crystal oscillator, HSE, and the Low Speed External Crystal oscillator, LSE, are operated under the V_{DD} power domain. The LDO can be configured to operate in either normal mode ($LD OFF = 0$, $LDOLCM = 0$, $I_{OUT} =$ high current mode) or low current mode ($LD OFF = 0$, $LDOLCM = 1$, $I_{OUT} =$ low current mode) to supply the 1.5 V power. An alternative 1.5 V power source is the output of the DMOS which has low static and driving current characteristics. It is controlled using the DMOSON bit in the PWRCR register. The DMOS output has weak output current and regulation capability and only operates in the Deep-Sleep2 mode for data retention purposes in the V_{DD15} power domain.

Power-On Reset (POR) / Power-Down Reset (PDR)

The device has an integrated POR/PDR circuitry that allows proper operation starting from/down to 1.65 V. When the device enters the Power-Down mode, it will remain in the Power-Down mode if V_{DD} is below a specified threshold V_{PDR} , without the need for an external reset circuit. For more details concerning the power-on/power-down reset threshold voltage, refer to the electrical characteristics of the corresponding datasheet.

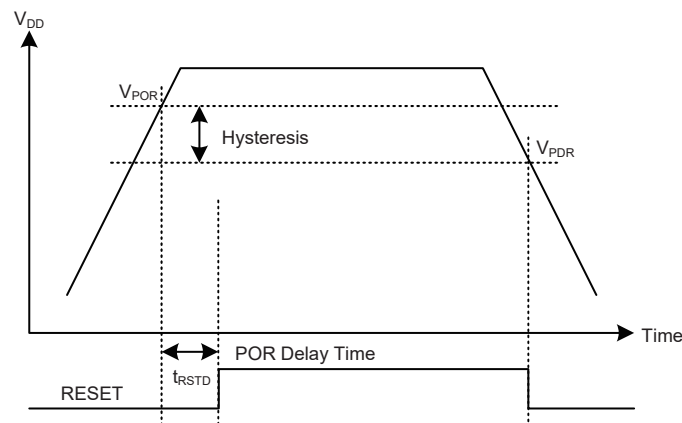


Figure 12. Power-On Reset / Power-Down Reset Waveform

Low Voltage Detector / Brown-Out Detector

The Low Voltage Detector, LVD, can detect whether the supply voltage V_{DD} is lower than a programmable threshold voltage V_{LVD} . It is selected by the LVDS field in the LVDCSR register. When a low voltage on the VDD power pin is detected, the LVDF flag will be active and an interrupt will be generated and sent to the CPU core if the LVDEN and LVDIWEN bits in the LVDCSR register are set. For more details concerning the LVD programmable threshold voltage V_{LVD} , refer to the electrical characteristics of the corresponding datasheet.

The Brown-Out Detector, BOD, is used to detect if the V_{DD} supply voltage is equal to or lower than V_{BOD} . When the BODEN bit in the LVDCSR register is set to 1 and the V_{DD} supply voltage is lower than V_{BOD} then the BODF flag is active. The PWRCU will regard this as a power-down reset situation and then immediately disable the internal LDO regulator when the BODRIS bit is cleared to 0 or issue an interrupt to notify the CPU to execute a power-down procedure when the BODRIS bit is set to 1. For more details concerning the Brown-Out Detector voltage V_{BOD} , refer to the electrical characteristics of the corresponding datasheet.

High Speed Internal Oscillator

The High Speed Internal Oscillator, HSI, is located in the V_{DD} power domain. When exiting from the Deep-Sleep mode, the HSI clock can be configured as the system clock for a certain period by setting the PSRCEN bit to 1. This bit is located in the Global Clock Control Register, GCCR, in the Clock Control Unit, CKCU. The system clock will not be switched back to the original clock source used before entering the Deep-Sleep mode until the original clock source, which may be either sourced from the PLL or HSE stabilizes. Also the system will force the HSI oscillator to be the system clock after a wakeup from Power-Down mode since a 1.5 V power-on reset will occur.

High Speed External Oscillator

The High Speed External Oscillator, HSE, is located in the V_{DD} power domain. The HSE crystal oscillator can be switched on or off using the HSEEN bit in the Global Clock Control Register, GCCR. The HSE clock can then be used directly as the system clock source or be used as the PLL input clock.

LSE, LSI and RTC

The Real Time Clock Timer clock source can be derived from either the Low Speed Internal RC oscillator, LSI, or the Low Speed External Crystal oscillator, LSE. Before entering the power saving mode by executing the WFI/WFE instruction, the MCU needs to setup the compare register with an expected wakeup time and enable the wakeup function to achieve the RTC timer wakeup event. After entering the power saving mode for a certain amount of time, the Compare Match flag, CMFLAG, will be asserted to wake up the device when the compare match event occurs. The details of the RTC configuration for wakeup timer will be described in the RTC chapter.

1.5 V Power Domain

The main functions that include the APB interface for the V_{DD} domain, CPU core logic, AHB/APB peripherals and memories and so on are located in this power domain. Once the 1.5 V is powered up, the POR will generate a reset sequence on the 1.5 V power domain. Subsequently, to enter the expected power saving mode, the associated control bits including the LDOOFF, DMOSON and LDOLCM bits must be configured. Then, once a WFI or WFE instruction is executed, the device will enter the expected power saving mode which will be discussed in the following section.

Operation Modes

Run Mode

In the Run mode, the system operates with full functions and all power domains are active. There are two ways to reduce the power consumption in this mode. The first is to slow down the system clock by setting the AHBPRE field in the CKCU AHBCFGR register, and the second is to turn off the unused peripherals clock by setting the APBCCR0 and APBCCR1 registers or slow down the peripherals clock by setting the APBPCSR0 and APBPCSR1 registers to meet the application requirement. Reducing the system clock speed before entering the sleep mode will also help to minimize power consumption.

Additionally, there are several power saving modes to provide maximum optimization between device performance and power consumption.

Table 11. Operation Mode Definitions

Mode Name	Hardware Action
Run	After system reset, CPU fetches instructions to execute.
Sleep	CPU clock will be stopped. Peripherals, Flash and SRAM clocks can be stopped by setting.
Deep-Sleep 1 ~ 2	Stop all clocks in the 1.5 V power domain. Disable HSI, HSE, and PLL. Turn on the LDO low current mode or DMOS to reduce the 1.5 V power domain current.
Power-Down	Shut down the 1.5 V power domain.

Sleep Mode

By default, only the CPU clock will be stopped in the Sleep mode. Clearing the FMCEN or SRAMEN bit in the CKCU AHBCCR register to zero will have the effect of stopping the Flash clock or SRAM clock after the system enters the Sleep mode. If it is not necessary for the CPU to access the Flash memory and SRAM in the Sleep mode, it is recommended to clear the FMCEN and SRAMEN bits in the AHBCCR register to minimize power consumption. To enter the Sleep mode, it is only necessary to execute a WFI or WFE instruction and let the SLEEPDEEP signal to be 0. The system will exit from the Sleep mode via any interrupt or event trigger. The accompanying table provides more information about the power saving modes.

Table 12. Enter/Exit Power Saving Modes

Mode	Mode Entry				Mode Exit
	CPU Instruction	CPU SLEEPDEEP	LDOOFF	DMOSON	
Sleep	WFI or WFE (Takes effect)	0	X	X	WFI: Any interrupt WFE: Any wakeup event ⁽¹⁾ or Any interrupt (NVIC on) or Any interrupt with SEVONPEND = 1 (NVIC off)
Deep-Sleep1		1	0	0	Any EXTI in event mode or RTC wakeup or LVD wakeup ⁽²⁾ or WAKEUP pin rising edge or USB resume
Deep-Sleep2		1	X	1	Any EXTI in event mode or RTC wakeup or LVD wakeup ⁽²⁾ or WAKEUP pin rising edge
Power-Down		1	1	0	RTC wakeup or LVD wakeup ⁽²⁾ or WAKEUP pin rising edge or External reset (nRST)

Notes: 1. Wakeup event means EXTI line trigger event, RTC event, LVD event or WAKEUP pin rising edge.

2. If the system allows the LVD activity to wake it up after the system has entered the power saving mode, the LVDEWEN and LVDEN bits in the LVDCSR register must be set to 1 to make sure that the system can be woken up by a LVD event and then the LDO regulator can be turned on when the system is woken up from the Deep-Sleep2 mode and Power-Down mode.

Deep-Sleep Mode

To enter the Deep-Sleep mode, configure the registers as shown in the preceding table and execute the WFI or WFE instruction. In the Deep-Sleep mode, all clocks including the PLL and high speed oscillators, known as HSI and HSE, will be stopped. In addition, the Deep-Sleep1 mode turns the LDO into low current mode while the Deep-Sleep2 mode turns off the LDO and uses a DMOS to keep the 1.5 V power. Once the PWRCU receives a wakeup event or an interrupt as shown in the preceding Mode-Exiting table, the LDO will then operate in normal mode and the high speed oscillators will be enabled. Finally, the CPU will return to the Run mode to handle the wakeup interrupt if required. A Low Voltage Detection also can be regarded as a wakeup event if the corresponding wakeup control bit LVDEWEN in the LVDCSR register is enabled. The last wakeup event is a transition from low to high on the external WAKEUP pin sent to the PWRCU to resume

from the Deep-Sleep mode. During the Deep-Sleep mode, retaining the register and memory contents will shorten the wakeup latency.

Power-Down Mode

The Power-Down mode is derived from the Deep-Sleep mode of the CPU together with the additional control bits LDOOFF and DMOSON. To enter the Power-Down mode, users can configure the registers shown in the preceding Mode-Entering table and execute the WFI or WFE instruction. An RTC wakeup trigger event, an LVD wakeup, a low to high transition on the external WAKEUP pin or an external reset (nRST) signal will force the MCU out of the Power-Down mode. In the Power-Down mode, the 1.5 V power supply will be turned off. The remaining active power supplies are the 3.3 V power (V_{DD} / V_{DDA}).

After a system reset, the PORSTF bit in the RSTCU GRSR register, the PDF and PORF bits in the PWRSR register should be checked by software to confirm if the device is being resumed from the Power-Down mode by a power-on reset or other reset events (nRST, WDT, ...). If the device has entered the Power-Down mode under the correct firmware procedure, the PDF bit will be set. The system information could be saved in the V_{DD} power domain registers and be retrieved when the 1.5 V power domain is powered on again. More information about the PDF and PORF bits in the PWRSR register and PORSTF bit in the RSTCU GRSR register is shown in the following table.

Table 13. Power Status after System Reset

PORF	PDF	PORSTF	Description
1	0	1	Power-up for the first time after the V_{DD} power domain is reset: Power-on reset when V_{DD} is applied for the first time or executing software reset command on the V_{DD} domain.
0	0	1	Restart from unexpected loss of the 1.5 V power or other reset (nRST, WDT, ...)
0	1	1	Restart from the Power-Down mode.
1	1	x	Reserved

Register Map

The following table shows the PWRCU registers and reset values. Note all the registers in this unit are located in the V_{DD} power domain.

Table 14. PWRCU Register Map

Register	Offset	Description	Reset Value
PWRSR	0x100	Power Control Status Register	0x0000_0001
PWRCR	0x104	Power Control Register	0x0000_0000
PWRTEST	0x108	V_{DD} Power Domain Test Register	0x0000_0027
LVDCSR	0x110	Low Voltage / Brown-Out Detect Control and Status Register	0x0000_0000

Register Descriptions

Power Control Status Register – PWRSR

This register indicates the power control status.

Offset: 0x100

Reset value: 0x0000_0001 (Reset only by VDD domain power-on reset)

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							WUPF
	7	6	5	4	3	2	1	0
Type/Reset	Reserved						PDF	PORF
							RC	0
							RC	1

Bits	Field	Descriptions
[8]	WUPF	External WAKEUP Pin Flag 0: The WAKEUP pin is not asserted 1: The WAKEUP pin is asserted This bit is set by hardware when the WAKEUP pin is asserted and is cleared by software read. Software should read this bit to clear it after a system wakeup from the power saving mode.
[1]	PDF	Power-Down Flag 0: Wakeup from abnormal VDD15 shutdown (loss of VDD15 is unexpected) 1: Wakeup from Power-Down mode (loss of VDD15 is under expectation) This bit is set by hardware when the system has successfully entered the Power-Down mode under the correct firmware procedure. This bit is cleared by software read.
[0]	PORF	Power-On Reset Flag 0: V _{DD} Power Domain reset does not occur 1: V _{DD} Power Domain reset occurs This bit is set by hardware when the V _{DD} power-on reset occurs, either a hardware power-on reset or software reset. The bit is cleared by software read. This bit must be cleared after the system is first powered on, otherwise it will be impossible to detect when a V _{DD} Power Domain reset has been triggered. When this bit is read as 1, a read software loop must be implemented until the bit returns again to 0. This software loop is necessary to confirm that the V _{DD} Power Domain is ready for access.

Power Control Register – PWRCR

This register provides power control bits for the different kinds of power saving modes.

Offset: 0x104

Reset value: 0x0000_0000 (Reset only by V_{DD} domain power-on reset)

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	DMOSSTS	Reserved				Reserved		WUPIEN
	RO	0					RW	0
	7	6	5	4	3	2	1	0
Type/Reset	DMOSON	Reserved			LDOOFF	LDOLCM	Reserved	PWRST
	RW	0			RW	0	RW	0

Bits	Field	Descriptions
[15]	DMOSSTS	Depletion MOS Status This bit is set to 1 if the DMOSON bit in this register has been set to 1. This bit is cleared to 0 if the DMOSON bit has been set to 0 or if a POR/PDR reset has occurred.
[9]	WUPIEN	External WAKEUP Pin Interrupt Enable 0: Disable WAKEUP pin interrupt function 1: Enable WAKEUP pin interrupt function The software can set the WUPIEN bit to 1 to assert the WKUP interrupt in the NVIC unit when both the WUPEN and WUPF bits are set to 1.
[8]	WUPEN	External WAKEUP Pin Enable 0: Disable WAKEUP pin function 1: Enable WAKEUP pin function The software can set the WUPEN bit as 1 to enable the WAKEUP pin function before entering the power saving mode. When WUPEN = 1, a rising edge on the WAKEUP pin wakes up the system from the power saving mode. As the WAKEUP pin is active high, this pin should be configured to an input and pull-down state. Note: This bit is reset by a V _{DD} Power Domain reset. Because this bit is located in the V _{DD} Power Domain, after the reset activity there will be a delay until the bit is active. The bit will not be active until the system reset finished and the V _{DD} Domain ISO signal has been disabled. This means that the bit can not be immediately set by software after a system reset finished and the V _{DD} domain ISO signal disabled. The delay time needed is a minimum of three 32 KHz clock periods until the bit reset activity has finished.
[7]	DMOSON	DMOS Control 0: DMOS is OFF 1: DMOS is ON A DMOS is implemented to provide an alternative voltage source for the 1.5 V power domain when the CPU enters the Deep-Sleep mode (SLEEPDEEP = 1). The control bit DMOSON is set by software and cleared by software or V _{DD} power domain reset. If the DMOSON bit is set to 1, the LDO will automatically be turned off when the CPU enters the Deep-Sleep mode.

Bits	Field	Descriptions
[3]	LDOOFF	<p>LDO Operating Mode Control</p> <p>0: The LDO operates in a low current mode when CPU enters the Deep-Sleep mode (SLEEPDEEP = 1). The V_{DD15} power is available.</p> <p>1: The LDO is turned off when the CPU enters the Deep-Sleep mode (SLEEPDEEP = 1). The V_{DD15} power is not available.</p> <p>Note: This bit is only available when the DMOSON bit is cleared to 0.</p>
[2]	LDOLCM	<p>LDO Low Current Mode</p> <p>0: The LDO is operated in normal current mode</p> <p>1: The LDO is operated in low current mode</p> <p>Note: This bit is only available when CPU is in the Run mode. The LDO output current capability will be limited at 10 mA below and lower static current when the LDOLCM bit is set. It is suitable for CPU which is operated at lower speed system clock to get a lower current consumption. This bit will be clear to 0 when the LDO is powered down or V_{DD} power domain reset occurs.</p>
[0]	PWRST	<p>V_{DD} Power Domain Software Reset</p> <p>0: No action</p> <p>1: V_{DD} Power Domain Software Reset is activated</p> <p>When this bit is set high, it will reset all RTC and PWRCU related registers.</p>

V_{DD} Power Domain Test Register – PWRTTEST

This register specifies a read-only value for the software to recognize whether V_{DD} Power Domain is ready for access.

Offset: 0x108

Reset value: 0x0000_0027

	31	30	29	28	27	26	25	24								
Type/Reset	Reserved															
	23	22	21	20	19	18	17	16								
Type/Reset	Reserved															
	15	14	13	12	11	10	9	8								
Type/Reset	Reserved															
	7	6	5	4	3	2	1	0								
Type/Reset	PWRTEST															
	RO	0	RO	0	RO	1	RO	0	RO	0	RO	1	RO	1	RO	1

Bits	Field	Descriptions
[7:0]	PWRTTEST	V _{DD} Power Domain Test Bits A constant 0x27 will be read when the V _{DD} Power Domain is ready for CPU access.

Low Voltage / Brown-Out Detect Control and Status Register – LVDCSR

This register specifies flags, enable bits and option bits for low voltage detector.

Offset: 0x110

Reset value: 0x0000_0000 (Reset only by V_{DD} domain power-on reset)

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved	LVDS [2]	LVDEWEN	LVDIWEN	LVDF	LVDS [1:0]		LVDEN	
		RW	0	RW	0	RO	0	RW	0
	15	14	13	12	11	10	9	8	
Type/Reset	Reserved								
	7	6	5	4	3	2	1	0	
Type/Reset	Reserved				BODF	Reserved	BODRIS	BODEN	
					RO	0	RW	0	RW
								0	

Bits	Field	Descriptions
[21]	LVDEWEN	<p>LVD Event Wakeup Enable</p> <p>0: LVD event wakeup is disabled</p> <p>1: LVD event wakeup is enabled</p> <p>Setting this bit to 1 will enable the LVD event wakeup function to wake up the system when an LVD condition occurs which results in the LVDF bit being asserted. If the system requires to be woken up from the Deep-Sleep or Power-Down mode by a LVD condition, this bit must be set to 1.</p>
[20]	LVDIWEN	<p>LVD Interrupt Wakeup Enable</p> <p>0: LVD interrupt wakeup is disabled</p> <p>1: LVD interrupt wakeup is enabled</p> <p>Setting this bit to 1 will enable the LVD interrupt function. When an LVD condition occurs and the LVDIWEN bit is set to 1, an LVD interrupt will be generated and sent to the CPU NVIC unit.</p>
[19]	LVDF	<p>Low Voltage Detect Status Flag</p> <p>0: V_{DD} is higher than the specific voltage level</p> <p>1: V_{DD} is equal to or lower than the specific voltage level</p> <p>When the LVD condition occurs, the LVDF flag will be asserted. When the LVDF flag is asserted, an LVD interrupt will be generated for CPU if the LVDIWEN bit is set to 1. However, if the LVDEWEN bit is set to 1 and the LVDIWEN bit is cleared to 0, only an LVD event will be generated rather than an LVD interrupt when the LVDF flag is asserted.</p>
[22], [18:17]	LVDS [2:0]	<p>Low Voltage Detect Level Selection</p> <p>For more details concerning the LVD programmable threshold voltage, refer to the electrical characteristics of the corresponding datasheet.</p>

Bits	Field	Descriptions
[16]	LVDEN	Low Voltage Detect Enable 0: Disable Low Voltage Detect 1: Enable Low Voltage Detect Setting this bit to 1 will generate an LVD event when the V_{DD} power is equal to or lower than the voltage set by the LVDS bits. Therefore when the LVD function is enabled before the system is into the Deep-Sleep2 (DMOS is turned on and LDO is powered down) or Power-Down mode (DMOS and LDO are powered down), the LVDEWEN bit has to be enabled to avoid that the LDO does not activate in the meantime when the CPU is woken up by the low voltage detection activity.
[3]	BODF	Brown-Out Detect Flag 0: $V_{DD} > V_{BOD}$ 1: $V_{DD} \leq V_{BOD}$
[1]	BODRIS	BOD Reset or Interrupt Selection 0: Reset the whole chip 1: Generate Interrupt
[0]	BODEN	Brown-Out Detector Enable 0: Disable Brown-Out Detector 1: Enable Brown-Out Detector

6 Clock Control Unit (CKCU)

Introduction

The Clock Control unit, CKCU, provides functions of High Speed Internal RC oscillator (HSI), High Speed External crystal oscillator (HSE), Low Speed Internal RC oscillator (LSI), Low Speed External crystal oscillator (LSE), Phase Lock Loop (PLL), HSE clock monitor, clock prescaler, clock multiplexer and clock gating. The clocks of AHB, APB, and CPU are derived from system clock (CK_SYS) which can come from HSI, HSE, LSI, LSE or PLL. Watchdog Timer and Real Time Clock (RTC) use either LSI or LSE as their clock source. The maximum operating frequency of system clock f_{CK_AHB} can be up to 60 MHz.

A variety of internal clocks can also be wired out through CKOUT for debugging purpose. The clock monitor can be used to get clock failure detection of HSE. Once the clock of HSE does not function such as being broken down or removed, etc., CKCU will force to switch the system clock source to the HSI clock to prevent system halt.

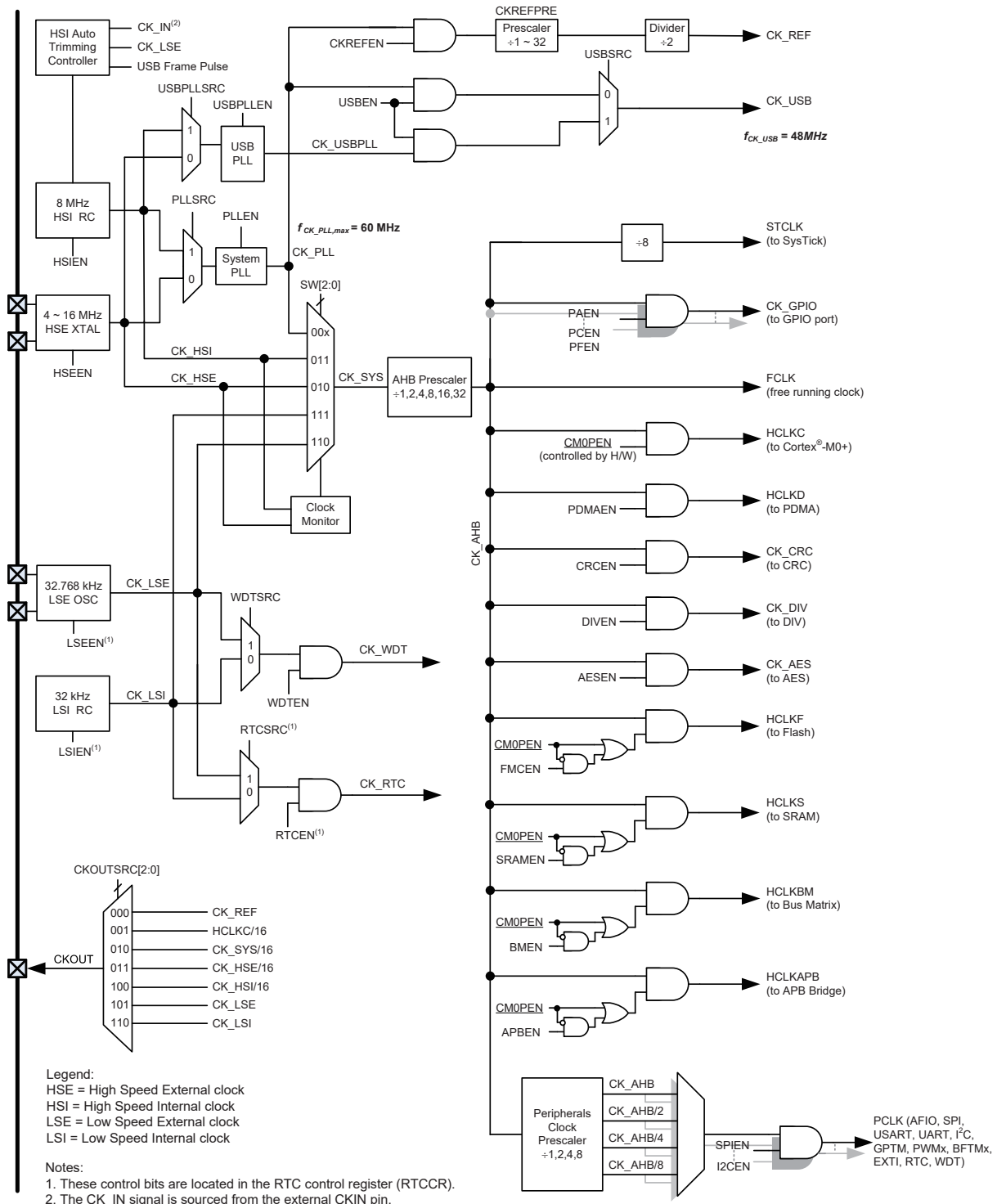


Figure 13. CKCU Block Diagram

Features

- 4 ~ 16 MHz external crystal oscillator (HSE)
- Internal 8 MHz RC oscillator (HSI) with configuration option calibration and custom trimming capability
- PLL with selectable clock source (from HSE or HSI) for system clock
- 32,768 Hz external crystal oscillator (LSE) for Watchdog Timer, RTC or system clock
- Internal 32 kHz RC oscillator (LSI) for Watchdog Timer, RTC or system clock
- HSE clock monitor

Functional Descriptions

High Speed External Crystal Oscillator – HSE

The high speed external 4 to 16 MHz crystal oscillator (HSE) produces a highly accurate clock source to the system clock. The related hardware configuration is shown in the following figure. The crystal with specific frequency must be placed across the two HSE pins (XTALIN / XTALOUT) and the external components such as resistors and capacitors are necessary to make it oscillate properly.

The following guidelines are provided to improve the stability of the crystal circuit PCB layout.

- The crystal oscillator should be located as close as possible to the MCU so that the trace lengths are kept as short as possible to reduce any parasitic capacitances.
- Shield any lines in the vicinity of the crystal by using a ground plane to isolate signals and reduce noise.
- Keep frequently switching signal lines away from the crystal area to prevent crosstalk.

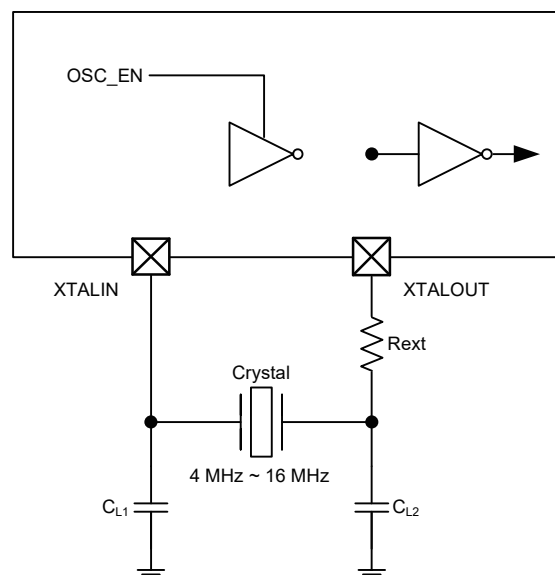


Figure 14. External Crystal, Ceramic and Resonators for HSE

The HSE crystal oscillator can be switched on or off using the HSEEN bit in the Global Clock Control Register (GCCR). The HSERDY flag in the Global Clock Status Register (GCSR) will indicate if the high speed external crystal oscillator is stable. While switching on the HSE, the HSE clock will still not be released until this HSERDY bit is set by the hardware. The specific delay period is well-known as “Start-up time”. As the HSE becomes stable, the HSE clock can then be used directly as the system clock source or be used as the PLL input clock.

High Speed Internal RC Oscillator – HSI

The high speed internal 8 MHz RC oscillator (HSI) is the default selection of clock source for the CPU when the device is powered up. The HSI RC oscillator provides a clock source in a lower cost because no external components are required. The HSI RC oscillator can be switched on or off using the HSIEN bit in the Global Clock Control Register (GCCR). The HSIRDY flag in the Global Clock Status Register (GCSR) will indicate if the internal RC oscillator is stable. The start-up time of HSI is shorter than the HSE crystal oscillator. The HSI clock can also be used as the PLL input clock.

The accuracy of the frequency of the high speed internal RC oscillator HSI can be calibrated via the configuration options, but it is still less accurate than the HSE crystal oscillator. The applications, the environments and the cost will determine the use of the oscillators.

Software could configure the Power Saving Wakeup RC Clock Enable bit, PSRCEN, to 1 to force the HSI clock to be the system clock when waking up from the Deep-Sleep1 or Deep-Sleep2 mode. Subsequently, the system clock is back to the original clock source (HSE or PLL) if the original clock source ready flag is asserted. This function can reduce the wakeup time when using HSE or PLL as system clock.

Auto Trimming of High Speed Internal RC Oscillator – HSI

The frequency accuracy of the high speed internal RC oscillator HSI can vary from one chip to another due to manufacturing process variations, this is why each device is factory calibrated by Holtek for $\pm 2\%$ accuracy at $V_{DD} = 3.3\text{ V}$ and $T_A = 25\text{ }^{\circ}\text{C}$. But the accuracy is not enough for some application and environment requirements. Therefore, this device provides the trimming mechanism for HSI frequency calibration using a more accurate external reference clock. The detailed block diagram is shown as the following figure.

After reset, the factory trimming value is loaded in HSICOARSE[4:0] and HSIFINE[7:0] bits in the HSI Control Register (HSICR). The HSI frequency accuracy may be affected by voltage or temperature variations. If the application has to be driven by a more accurate HSI frequency, users can manually trim the HIS frequency using the HSIFINE[7:0] in the HSI Control Register (HSICR) or automatically adjust the HIS frequency using the Auto Trimming Controller (ATC) together with an external reference clock in the application. The reference clock can be provided from the following clock sources:

- 32,768 Hz low speed external crystal or ceramic resonator oscillator LSE output clock
- 1 kHz USB frame pulse
- External pin (CKIN) with 1 kHz pulse

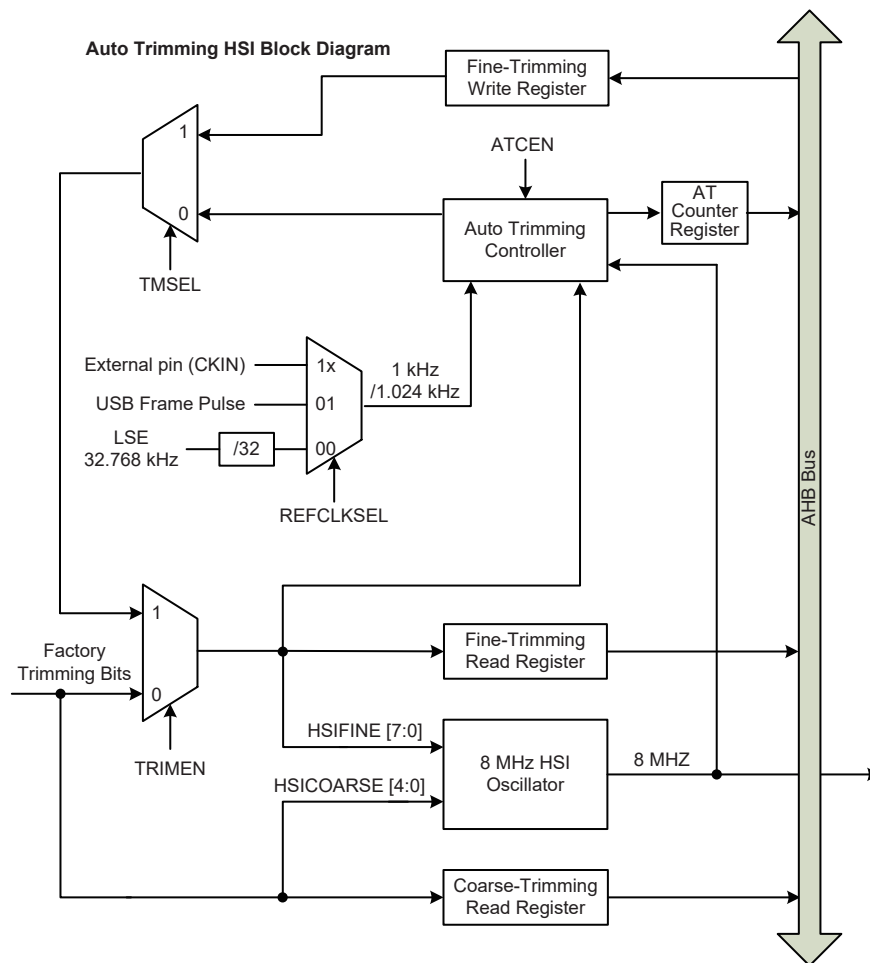


Figure 15. HSI Auto Trimming Block Diagram

Phase Locked Loop – PLL

This PLL can provide a 4 ~ 60 MHz clock output which is 1 ~ 15 multiples of a fundamental reference frequency of 4 ~ 16 MHz. The rationale of the clock synthesizer relies on the digital Phase Locked Loop (PLL) which includes a reference divider, a two-stage feedback divider, a two-stage output divider, a digital phase frequency detector (PFD), a current-controlled charge pump, a built-in loop filter and a voltage-controlled oscillator (VCO) to achieve a stable phase-locked state.

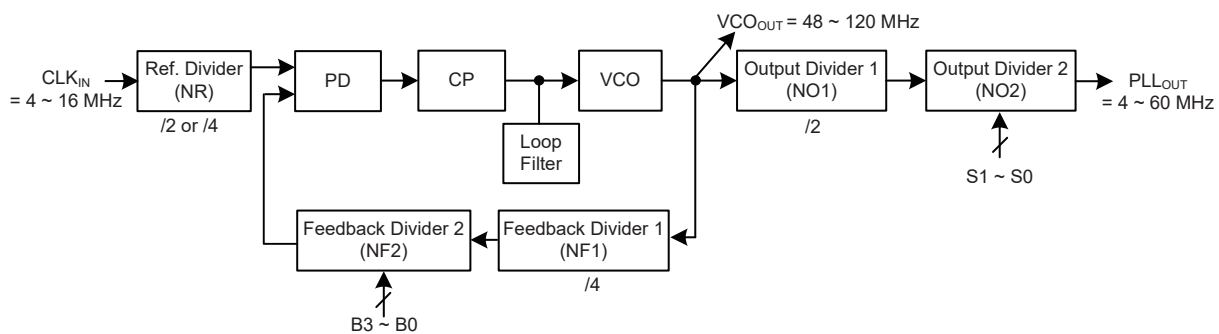


Figure 16. PLL Block Diagram

Frequency of the PLL output clock can be determined by the following formula:

$$PLL_{OUT} = CLK_{IN} \times \frac{NF1 \times NF2}{NR \times NO1 \times NO2} = CLK_{IN} \times \frac{4 \times NF2}{2 \times 2 \times NO2} = CLK_{IN} \times \frac{NF2}{NO2}$$

where NR = Ref divider = 2 or 4, NF1 = Feedback Divider 1 = 4, NF2 = Feedback Divider 2 = 1 ~ 16, NO1 = Output Divider 1 = 2, NO2 = Output Divider 2 = 1, 2, 4, or 8

Considering the duty cycle with 50%, both input frequency and output frequency is divided by 2. Assume that a given CLK_{IN} frequency as PLL input generates a specific PLL output frequency; a larger number of NF2 is suggested because it will cause the PLL more stable and less jittered but enlarges the settling time. The output and feedback of divider 2 value are described in Table 15 and Table 16. All the configuration bits (S1 ~ S0, B3 ~ B0) in Table 15 and Table 16 as well as the Bypass mode control are defined in the PLL Configuration Register (PLLCFGR) and PLL Control Register (PLLCR) in the section of Register Definition. Note that VCO_{OUT} is ranged from 48 MHz to 120 MHz. If VCO_{OUT} by user's configurations exceeds this range, the output frequency of the PLL will not be promised to match the above PLL_{OUT} formula.

The PLL can be switched on or off by using the PLEN bit in the Global Clock Control Register (GCCR). The PLLRDY flag in the Global Clock Status Register (GCSR) will indicate if the PLL clock is stable.

Table 15. Output Divider 2 Value Mapping

Output Divider 2 Setup Bits S1 ~ S0 (POTD Field in the PLLCFGR Register)	NO2 (Output Divider 2 Value)
00	1
01	2
10	4
11	8

Table 16. Feedback Divider 2 Value Mapping

Feedback Divider 2 Setup Bits B3 ~ B0 (PFBD Field in the PLLCFGR Register)	NF2 (Feedback Divider 2 Value)
0000	16
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
:	:
:	:
1111	15

USB Phase Locked Loop – USB PLL

This USB PLL can provide a 48 MHz clock output for USB peripheral which is 3 ~ 12 multiples of a fundamental reference frequency of 4 ~ 16 MHz. The rationale of the clock synthesizer relies on the digital Phase Locked Loop (PLL) which includes a reference divider, a two-stage feedback divider, a two-stage output divider, a digital phase frequency detector (PFD), a current-controlled charge pump, a built-in loop filter and a voltage-controlled oscillator (VCO) to achieve a stable phase-locked state.

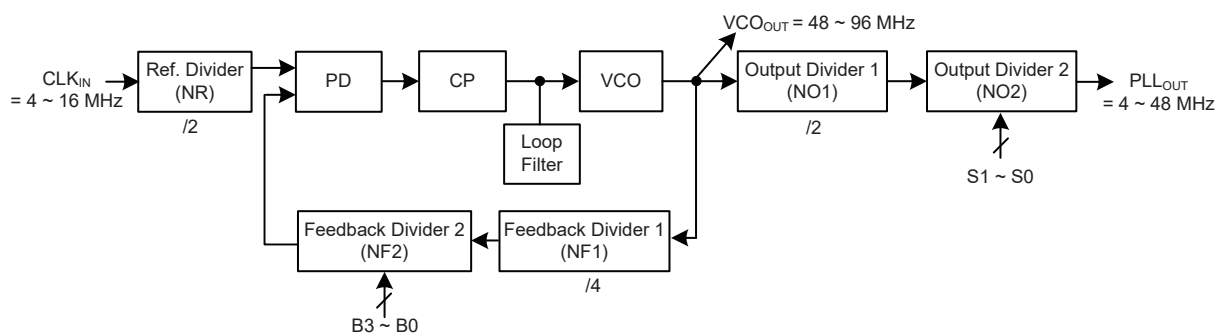


Figure 17. USB PLL Block Diagram

Frequency of the PLL output clock can be determined by the following formula:

$$PLL_{OUT} = CLK_{IN} \times \frac{NF1 \times NF2}{NR \times NO1 \times NO2} = CLK_{IN} \times \frac{4 \times NF2}{2 \times 2 \times NO2} = CLK_{IN} \times \frac{NF2}{NO2}$$

where NR = Ref divider = 2, NF1 = Feedback Divider 1 = 4, NF2 = Feedback Divider 2 = 1 ~ 16,
NO1 = Output Divider 1 = 2, NO2 = Output Divider 2 = 1, 2, 4, or 8

Considering the duty cycle with 50%, both input frequency and output frequency is divided by 2. Assume that a given CLK_{IN} frequency as USB PLL input generates a specific USB PLL output frequency; a larger number of NF2 is suggested because it will cause the USB PLL more stable and less jittered but enlarges the settling time. The output and feedback of divider 2 value are described in Table 17 and Table 18. All the configuration bits (S1 ~ S0, B3 ~ B0) in Table 17 and Table 18 are defined in the PLL Configuration Register (PLLCFGR) in the section of Register Definition. Note that VCO_{OUT} is ranged from 48 MHz to 96 MHz. If VCO_{OUT} by user's configurations exceeds this range, the output frequency of USB PLL will not be promised to match the above PLL_{OUT} formula.

The USB PLL can be switched on or off by using the USBPLEN bit in the Global Clock Control Register (GCCR). The USBPLLRDY flag in the Global Clock Status Register (GCSR) will indicate if the USB PLL clock is stable.

Table 17. USB PLL Output Divider 2 Value Mapping

Output Divider 2 Setup Bits S1 ~ S0 (USBPOTD Field in the PLLCFGR Register)	NO2 (Output Divider 2 Value)
00	1
01	2
10	4
11	8

Table 18. USB PLL Feedback Divider 2 Value Mapping

Feedback Divider 2 Setup Bits B3 ~ B0 (USBPFBD Field in the PLLCFGR Register)	NF2 (Feedback Divider 2 Value)
0000	16
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
:	:
:	:
1111	15

Low Speed External Crystal Oscillator – LSE

The low speed external crystal or ceramic resonator oscillator with a 32,768 Hz frequency produces a low power but highly accurate clock source for the circuits of Real-Time-Clock peripheral, Watchdog Timer or system clock. The associated hardware configuration is shown in the following figure. The crystal or ceramic resonator must be placed across the two LSE pins (X32KIN / X32KOUT) and the external components such as resistors and capacitors are necessary to make it oscillate properly. The LSE oscillator can be switched on or off by using the LSEEN bit in the RTC Control Register (RTCCR). The LSERDY flag in the Global Clock Status Register (GCSR) will indicate if the LSE clock is stable.

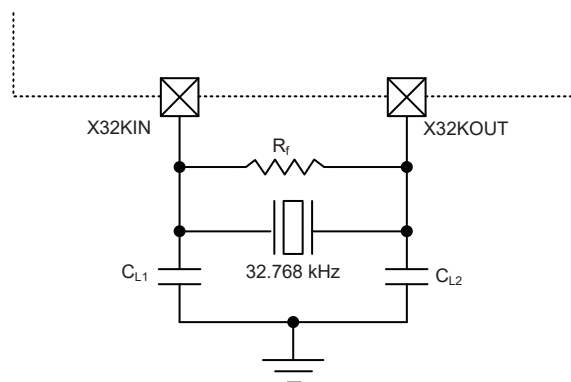


Figure 18. External Crystal, Ceramic and Resonators for LSE

Low Speed Internal RC Oscillator – LSI

The low speed internal RC oscillator with a frequency of about 32 kHz produces a low power clock source for the circuits of Real-Time-Clock peripheral, Watchdog Timer or system clock. The LSI is also a low cost clock source because no external component is needed to make it oscillate. The LSI RC oscillator can be switched on or off by using the LSIEN bit in the RTC Control Register (RTCCR). The frequency accuracy of the low speed internal RC oscillator LSI is shown in the corresponding data sheet. The LSIRDY flag in the Global Clock Status Register (GCSR) will indicate if the LSI clock is stable.

Clock Ready Flag

The CKCU provides clock ready flags for HIS, HSE, PLL, LSI, and LSE to confirm these clocks are stable before using them as system clock source or other purposes. Software can check specific clock is ready or not by polling separate clock ready status bits in GCSR register.

System Clock (CK_SYS) Selection

After a system reset occurs, the default source of the system clock CK_SYS will be the high speed internal RC oscillator HSI. The CK_SYS clock may come from the HSI, HSE, LSE, LSI or PLL output clock and it can be switched from one clock source to another via the System Clock Switch field, SW, in the Global Clock Control Register (GCCR). The system will still run under the original clock until the destination clock gets ready. The corresponding clock ready status bit in the Global Clock Status Register (GCSR) will indicate whether the selected clock is ready to use or not.

The CKCU also contains the clock source status bits in the Clock Source Status Register (CKST) to indicate which clock is currently used as the system clock. If a clock source or the PLL output clock is used as the system clock, it is not possible to stop it. More details about clock enable function are described below.

If any following action takes effect, the HSI is always under enable state.

- Enable PLL and configure its source clock to HSI. (PLEN, PLLSRC)
- Enable Clock monitor. (CKMEN)
- Configure clock switch field to select HSI. (SW)
- Configure HSI enable bit to 1. (HSIEN)

If any following action takes effect, the HSE is always under enable state.

- Enable PLL and configure its source clock to HSE. (PLEN, PLLSRC)
- Configure clock switch field to select HSE. (SW)
- Configure HSE enable bit to 1. (HSEEN)

If any following action takes effect, the PLL is always under enable state.

- Configure USB clock enable bit to 1. (USBEN)
- Configure clock switch field to select PLL (SW)
- Configure PLL enable bit to 1. (PLEN)

Programming guide of system clock selection is listed below.

1. Enable any source clock which will become the system clock or PLL input clock.
2. Configure the PLLSRC bit after the ready flags of both HSI and HSE are asserted.
3. Configure the SW field to change the system clock source after the corresponding clock ready flag is asserted. Note that the system clock will force to HSI if the clock monitor is enabled and the PLL output clock or HSE clock configured as system clock is stuck at 0 or 1.

HSE Clock Monitor

The main function of the oscillator check is enabled by the HSE Clock Monitor Enable bit CKMEN in the Global Clock Control Register (GCCR). The HSE clock monitor should be enabled after the HSE oscillator start-up delay and be disabled when the HSE oscillator is stopped. Once the HSE oscillator failure is detected, the HSE oscillator will automatically be disabled. The HSE clock stuck flag CKSF in the Global Clock Interrupt Register (GCIR) will be set and an interrupt of main oscillator failure will be generated if the clock stuck interrupt enable bit CKSIE in the GCIR is set. This failure interrupt is connected to the exception vector of CPU Non-Maskable Interrupt (NMI). If the HSE is directly used as the system clock source, when the HSE oscillator failure occurs, the HSE will be turned off and the system clock will be switched to the HSI automatically by the hardware. If the HSE is used as the clock input of the PLL circuit and the system clock comes from the PLL circuit output, the PLL circuit will also be turned off as well as the HSE when the failure happens.

Clock Output Capability

The device has the clock output capability to allow the clocks to be output on the specific external output pin CKOUT. The configuration registers of the corresponding GPIO port must be well configured in the Alternate Function I/O section, AFIO, to output the selected clock signal. There are seven output clock signals to be selected via the device clock output source selection field CKOUTSRC in the Global Clock Configuration Register (GCFGR).

Table 19. CKOUT Clock Source

CKOUTSRC[2:0]	Clock Source
000	CK_REF = CK_PLL / (CKREFPRE + 1) / 2
001	HCLKC / 16
010	CK_SYS / 16
011	CK_HSE / 16
100	CK_HSI / 16
101	CK_LSE
110	CK_LSI

Register Map

The following table shows the CKCU registers and reset values.

Table 20. CKCU Register Map

Register	Offset	Description	Reset Value
GCFGR	0x000	Global Clock Configuration Register	0x0000_0302
GCCR	0x004	Global Clock Control Register	0x0000_0803
GCSR	0x008	Global Clock Status Register	0x0000_0028
GCIR	0x00C	Global Clock Interrupt Register	0x0000_0000
PLLCFGR	0x018	PLL Configuration Register	0x0000_0000
PLLCR	0x01C	PLL Control Register	0x0000_0000
AHBCFGR	0x020	AHB Configuration Register	0x0000_0000
AHBCCR	0x024	AHB Clock Control Register	0x0000_0065
APBCFGR	0x028	APB Configuration Register	0x0000_0000
APBCCR0	0x02C	APB Clock Control Register 0	0x0000_0000
APBCCR1	0x030	APB Clock Control Register 1	0x0000_0000
CKST	0x034	Clock Source Status Register	0x0100_0003
APBPCSR0	0x038	APB Peripheral Clock Selection Register 0	0x0000_0000
APBPCSR1	0x03C	APB Peripheral Clock Selection Register 1	0x0000_0000
HSICR	0x040	HSI Control Register	0xFFFF_0000 where X is undefined
HSIATCR	0x044	HSI Auto Trimming Counter Register	0x0000_0000
APBPCSR2	0x048	APB Peripheral Clock Selection Register 2	0x0000_0000
LPCR	0x300	Low Power Control Register	0x0000_0000
MCUDBGCR	0x304	MCU Debug Control Register	0x0000_0000

Register Descriptions

Global Clock Configuration Register – GCFGR

This register specifies the clock source for PLL/USART/Watchdog Timer/CKOUT.

Offset: 0x000

Reset value: 0x0000_0302

	31	30	29	28	27	26	25	24
	LPMOD				Reserved			
Type/Reset	RO	0	RO	0	RO	0		
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	CKREFPRE					USBSRC	USBPLLSRC	PLLSRC
Type/Reset	RW	0	RW	0	RW	0	RW	1
	7	6	5	4	3	2	1	0
	Reserved					CKOUTSRC		
Type/Reset						RW	0	RW

Bits	Field	Descriptions
[31:29]	LPMOD	Lower Power Mode Status 000: When chip is in running mode 001: When chip wants to enter Sleep mode 010: When chip wants to enter Deep-Sleep1 mode 011: When chip wants to enter Deep-Sleep2 mode 100: When chip wants to enter Power-Down mode Set and reset by hardware.
[15:11]	CKREFPRE	CK_REF Clock Prescaler Selection $CK_REF = CK_PLL / (CKREFPRE + 1) / 2$ 00000: $CK_REF = CK_PLL / 2$ 00001: $CK_REF = CK_PLL / 4$... 11111: $CK_REF = CK_PLL / 64$ Set and reset by software to control the CK_REF clock prescaler setting.
[10]	USBSRC	USB Clock Source Selection 0: CK_PLL clock is selected 1: CK_USBPLL clock is selected Set and reset by software to control the USB clock source.
[9]	USBPLLSRC	USB PLL Clock Source Selection 0: External 4 ~ 16 MHz crystal oscillator clock is selected (HSE) 1: Internal 8 MHz RC oscillator clock is selected (HSI) Set and reset by software to control the USB PLL clock source.
[8]	PLLSRC	PLL Clock Source Selection 0: External 4 ~ 16 MHz crystal oscillator clock is selected (HSE) 1: Internal 8 MHz RC oscillator clock is selected (HSI) Set and reset by software to control the PLL clock source.

Bits	Field	Descriptions
[2:0]	CKOUTSRC	CKOUT Clock Source Selection 000: CK_REF is selected, $CK_REF = CK_PLL / (CKREFPRE + 1) / 2$ 001: (HCLKC / 16) is selected 010: (CK_SYS / 16) is selected 011: (CK_HSE / 16) is selected 100: (CK_HSI / 16) is selected 101: CK_LSE is selected 110: CK_LSI is selected 111: Reserved Set and reset by software to control the CKOUT clock source.

Global Clock Control Register – GCCR

This register specifies the clock enable bits.

Offset: 0x004

Reset value: 0x0000_0803

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved						PSRCEN	CKMEN	
							RW	0	RW
	15	14	13	12	11	10	9	8	
Type/Reset	Reserved				HSIEN	HSEEN	PLEN	HSEGAIN	
					RW	1	RW	0	RW
							0	RW	0
	7	6	5	4	3	2	1	0	
Type/Reset	Reserved				USBPLEN	SW			
					RW	0	RW	0	RW
							1	RW	1

Bits	Field	Descriptions
[17]	PSRCEN	Power Saving Wakeup RC Clock Enable 0: No action 1: Use Internal 8 MHz RC clock (HSI) as system clock after a Deep-Sleep1/2 mode wakeup Software can set PSRCEN high before entering the Deep-Sleep1/2 mode in order to reduce the waiting time after wakeup. When PSRCEN = 1, hardware will select HSI as clock source after the system wakeup from Deep-Sleep1/2 mode. Meanwhile, instruction can start execution since the HSI clock is provided to CPU. After the original clock source, which is selected as CK_SYS before entering the Deep-Sleep1/2 mode, is ready, hardware will switch back the clock source as originally.
[16]	CKMEN	HSE Clock Monitor Enable 0: Disable external 4 ~ 16 MHz crystal oscillator clock monitor 1: Enable external 4 ~ 16 MHz crystal oscillator clock monitor When hardware detects the HSE clock stuck at low/high state, internal hardware will switch the system clock to the internal high speed RC clock (HSI).

Bits	Field	Descriptions
[11]	HSIEN	Internal High Speed Clock Enable 0: Internal 8 MHz RC oscillator clock is disabled 1: Internal 8 MHz RC oscillator clock is enabled Set and reset by software. This bit can not be reset if the HSI clock is used as system clock or PLL input clock.
[10]	HSEEN	External High Speed Clock Enable 0: External 4 ~ 16 MHz crystal oscillator clock is disabled 1: External 4 ~ 16 MHz crystal oscillator clock is enabled Set and reset by software. This bit can not be reset if the HSE clock is used as system clock or PLL input clock.
[9]	PLLEN	PLL Enable 0: PLL off 1: PLL on Set and reset by software to control the PLL. This bit can not be reset if the PLL clock is used as system clock.
[8]	HSEGAIN	External High Speed Clock Gain Selection 0: HSE low gain mode 1: HSE high gain mode
[3]	USBPLEN	USB PLL Enable 0: USB PLL is disabled 1: USB PLL is enabled Set and reset by software to control the USB PLL.
[2:0]	SW	System Clock Switch 00x: CK_PLL clock out as system clock 010: CK_HSE as system clock 011: CK_HSI as system clock 110: CK_LSE as system clock 111: CK_LSI as system clock Other: CK_HSI as system clock These bits are used to select the CK_SYS source. If the HSE oscillator is used directly or indirectly as the system clock and the HSE clock monitor function is enabled, once the HSE failure is detected, these bits will be set by hardware to force HSI (b011) as the system clock. Note: When switching the system clock using the SW field, the system clock will not be immediately switched and a certain delay is necessary. Software can monitor the CKSWST field in the clock source status register CKSTR to make sure which clock is currently used as the system clock.

Global Clock Status Register – GCSR

This register indicates the clock ready status.

Offset: 0x008

Reset value: 0x0000_0028

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved	LSIRDY	LSERDY	HSIRDY	HSERDY	PLLRDY	USBPLLRDY	
		RO	1 RO	0 RO	1 RO	0 RO	0 RO	0

Bits	Field	Descriptions
[5]	LSIRDY	Internal Low Speed Clock Ready Flag 0: Internal 32 kHz RC oscillator clock is not ready 1: Internal 32 kHz RC oscillator clock is ready Set by hardware to indicate that the LSI is stable to be used.
[4]	LSERDY	External Low Speed Clock Ready Flag 0: External 32,768 Hz crystal oscillator clock is not ready 1: External 32,768 Hz crystal oscillator clock is ready Set by hardware to indicate that the LSE is stable to be used.
[3]	HSIRDY	Internal High Speed Clock Ready Flag 0: Internal 8 MHz RC oscillator clock is not ready 1: Internal 8 MHz RC oscillator clock is ready Set by hardware to indicate whether the HSI is stable or not.
[2]	HSERDY	External High Speed Clock Ready Flag 0: External 4 ~ 16 MHz crystal oscillator clock is not ready 1: External 4 ~ 16 MHz crystal oscillator clock is ready Set by hardware to indicate that the HSE is stable to be used.
[1]	PLLRDY	PLL Clock Ready Flag 0: PLL is not ready 1: PLL is ready Set by hardware to indicate that the PLL is stable to be used.
[0]	USBPLLRDY	USB PLL Clock Ready Flag 0: USB PLL is not ready 1: USB PLL is ready Set by hardware to indicate that the USB PLL is stable to be used.

Global Clock Interrupt Register – GCIR

This register specifies interrupt enable and flag bits.

Offset: 0x00C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							CKSIE
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved							CKSF
								WC 0

Bits	Field	Descriptions
[16]	CKSIE	Clock Stuck Interrupt Enable 0: Disable clock fail interrupt 1: Enable clock fail interrupt Set and reset by software to enable/disable interrupt caused by clock monitor.
[0]	CKSF	Clock Stuck Interrupt Flag 0: Clock works normally 1: HSE clock is stuck Set by hardware when the HSE clock stuck and CKMEN is set. Reset by software writing 1.

PLL Configuration Register – PLLCFGR

This register specifies PLL configuration.

Offset: 0x018

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved			REFDIV	Reserved	PFBD		
Type/Reset				RW 0		RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	PFBD	POTD			Reserved			
Type/Reset	RW 0	RW 0	RW 0					
	15	14	13	12	11	10	9	8
	Reserved					USBPFBD		
Type/Reset						RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	USBPFBD	USBPOTD			Reserved			
Type/Reset	RW 0	RW 0	RW 0					

Bits	Field	Descriptions
[28]	REFDIV	PLL Input Reference Clock Divider 0: Reference divider NR = 2 1: Reference divider NR = 4
[26:23]	PFBD	PLL VCO Output Clock Feedback Divider (B3 ~ B0 in PLL Block Diagram) Feedback Divider divides the output clock from VCO of PLL.
[22:21]	POTD	PLL Output Clock Divider (S1 ~ S0 in PLL Block Diagram)
[10:7]	USBPFBD	USB PLL VCO Output Clock Feedback Divider (B3 ~ B0 in USB PLL Block Diagram) Feedback Divider divides the output clock from VCO of USB PLL.
[6:5]	USBPOTD	USB PLL Output Clock Divider (S1 ~ S0 in USB PLL Block Diagram)

PLL Control Register – PLLCR

This register specifies Bypass mode control of PLL.

Offset: 0x01C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	PLLBPS	Reserved						
Type/Reset	RW	0						
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved							
Type/Reset								

Bits	Field	Descriptions
[31]	PLLBPS	PLL Bypass Mode Enable 0: Disable PLL Bypass mode 1: Enable PLL Bypass mode where $f_{OUT} = f_{IN}$

AHB Configuration Register – AHBCFGR

This register specifies the system clock frequency.

Offset: 0x020

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved				AHBPRE			
					RW	0	RW	0

Bits	Field	Descriptions
[2:0]	AHBPRE	<p>AHB Pre-scaler</p> <p>000: CK_AHB = CK_SYS</p> <p>001: CK_AHB = CK_SYS / 2</p> <p>010: CK_AHB = CK_SYS / 4</p> <p>011: CK_AHB = CK_SYS / 8</p> <p>100: CK_AHB = CK_SYS / 16</p> <p>101: CK_AHB = CK_SYS / 32</p> <p>110: CK_AHB = CK_SYS / 32</p> <p>111: CK_AHB = CK_SYS / 32</p> <p>Set and reset by software to control the division factor of the AHB clock.</p>

AHB Clock Control Register – AHBCCR

This register specifies clock enable bits of AHB.

Offset: 0x024

Reset value: 0x0000_0065

	31	30	29	28	27	26	25	24
	Reserved							DIVEN
Type/Reset								RW 0
	23	22	21	20	19	18	17	16
	Reserved		PFEN	Reserved		PCEN	PBEN	PAEN
Type/Reset			RW 0			RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	AESEN	Reserved	CRCEN	Reserved	CKREFEN	USBEN	Reserved	
Type/Reset	RW 0		RW 0		RW 0	RW 0		
	7	6	5	4	3	2	1	0
	Reserved	APBEN	BMEN	PDMAEN	Reserved	SRAMEN	Reserved	FMCEN
Type/Reset		RW 1	RW 1	RW 0		RW 1		RW 1

Bits	Field	Descriptions
[24]	DIVEN	Divider Clock Enable 0: Divider clock is disabled 1: Divider clock is enabled Set and reset by software.
[21]	PFEN	GPIO Port F Clock Enable 0: Port F clock is disabled 1: Port F clock is enabled Set and reset by software.
[18]	PCEN	GPIO Port C Clock Enable 0: Port C clock is disabled 1: Port C clock is enabled Set and reset by software.
[17]	PBEN	GPIO Port B Clock Enable 0: Port B clock is disabled 1: Port B clock is enabled Set and reset by software.
[16]	PAEN	GPIO Port A Clock Enable 0: Port A clock is disabled 1: Port A clock is enabled Set and reset by software.
[15]	AESEN	AES Module Clock Enable 0: AES clock is disabled 1: AES clock is enabled Set and reset by software.
[13]	CRCEN	CRC Module Clock Enable 0: CRC clock is disabled 1: CRC clock is enabled Set and reset by software.

Bits	Field	Descriptions
[11]	CKREFEN	CK_REF Clock Enable 0: CK_REF clock is disabled 1: CK_REF clock is enabled Set and reset by software.
[10]	USBEN	USB Clock Enable 0: USB clock is disabled 1: USB clock is enabled Set and reset by software.
[6]	APBEN	APB bridge Clock Enable 0: APB bridge clock is automatically disabled by hardware during Sleep mode 1: APB bridge clock is always enabled during Sleep mode Set and reset by software. Users can set APBEN as 0 to reduce power consumption if the APB bridge is unused during Sleep mode.
[5]	BMEN	Bus Matrix Clock Enable 0: Bus Matrix clock is automatically disabled by hardware during Sleep mode 1: Bus Matrix clock is always enabled during Sleep mode Set and reset by software. Users can set BMEN as 0 to reduce power consumption if the bus matrix is unused during Sleep mode.
[4]	PDMAEN	Peripheral DMA Clock Enable 0: PDMA clock is disabled 1: PDMA clock is enabled Set and reset by software. Note: The PDMA can independently operate when the processor enters the sleep mode. But the relative clock of AHB bus slave or peripherals has to be enabled.
[2]	SRAMEN	SRAM Clock Enable 0: SRAM clock is automatically disabled by hardware during Sleep mode 1: SRAM clock is always enabled during Sleep mode Set and reset by software. Users can set SRAMEN as 0 to reduce power consumption if the SRAM is unused during Sleep mode.
[0]	FMCEN	Flash Memory Controller Clock Enable 0: FMC clock is automatically disabled by hardware during Sleep mode 1: FMC clock is always enabled during Sleep mode Set and reset by software. Users can set FMCEN as 0 to reduce power consumption if the Flash Memory is unused during Sleep mode.

APB Clock Control Register 0 – APBCCR0

This register specifies clock enable bits of APB peripherals.

Offset: 0x02C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	EXTIEN	AFIOEN	Reserved			UREN	Reserved	USREN
	RW 0	RW 0				RW 0		RW 0
	7	6	5	4	3	2	1	0
Type/Reset	Reserved			SPIEN	Reserved		I2C0EN	
				RW 0			RW 0	

Bits	Field	Descriptions
[15]	EXTIEN	External Interrupt Clock Enable 0: EXTI clock is disabled 1: EXTI clock is enabled Set and reset by software.
[14]	AFIOEN	Alternate Function I/O Clock Enable 0: AFIO clock is disabled 1: AFIO clock is enabled Set and reset by software.
[10]	UREN	UART Clock Enable 0: UART clock is disabled 1: UART clock is enabled Set and reset by software.
[8]	USREN	USART Clock Enable 0: USART clock is disabled 1: USART clock is enabled Set and reset by software.
[4]	SPIEN	SPI Clock Enable 0: SPI clock is disabled 1: SPI clock is enabled Set and reset by software.
[0]	I2CEN	I ² C Clock Enable 0: I ² C clock is disabled 1: I ² C clock is enabled Set and reset by software.

APB Clock Control Register 1 – APBCCR1

This register specifies clock enable bits APB peripherals.

Offset: 0x030

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved						BFTM1EN	BFTM0EN	
							RW 0	RW 0	
	15	14	13	12	11	10	9	8	
Type/Reset	Reserved		PWM1	PWM0	Reserved			GPTMEN	
			RW 0	RW 0				RW 0	
	7	6	5	4	3	2	1	0	
Type/Reset	Reserved	VDDREN	Reserved	WDTREN	Reserved				
		RW 0		RW 0					

Bits	Field	Descriptions
[17]	BFTM1EN	BFTM1 Clock Enable 0: BFTM1 clock is disabled 1: BFTM1 clock is enabled Set and reset by software.
[16]	BFTM0EN	BFTM0 Clock Enable 0: BFTM0 clock is disabled 1: BFTM0 clock is enabled Set and reset by software.
[13]	PWM1EN	PWM1 Clock Enable 0: PWM1 clock is disabled 1: PWM1 clock is enabled Set and reset by software.
[12]	PWM0EN	PWM0 Clock Enable 0: PWM0 clock is disabled 1: PWM0 clock is enabled Set and reset by software.
[8]	GPTMEN	GPTM Clock Enable 0: GPTM clock is disabled 1: GPTM clock is enabled Set and reset by software.
[6]	VDDREN	V _{DD} Domain Clock Enable for Registers Access 0: Register access clock is disabled 1: Register access clock is enabled Set and reset by software.
[4]	WDTREN	Watchdog Timer Clock Enable for Registers Access 0: Register access clock is disabled 1: Register access clock is enabled Set and reset by software.

Clock Source Status Register – CKST

This register specifies status of various clock sources.

Offset: 0x034

Reset value: 0x0100_0003

	31	30	29	28	27	26	25	24				
	Reserved				HSIST							
Type/Reset					RO	0	RO	0	RO	0	RO	1
	23	22	21	20	19	18	17	16				
	Reserved				HSEST							
Type/Reset						RO	0	RO	0	RO	0	0
	15	14	13	12	11	10	9	8				
	Reserved				PLLST							
Type/Reset						RO	0	RO	0	RO	0	0
	7	6	5	4	3	2	1	0				
	Reserved				CKSWST							
Type/Reset						RO	0	RO	1	RO	1	1

Bits	Field	Descriptions
[27:24]	HSIST	Internal High Speed Clock Occupation Status (CK_HSI) xxx1: HSI is used by System Clock (CK_SYS) (SW = 0x3) xx1x: HSI is used by System PLL x1xx: HSI is used by Clock Monitor 1xxx: HSI is used by USB PLL
[18:16]	HSEST	External High Speed Clock Occupation Status (CK_HSE) xx1: HSE is used by System Clock (CK_SYS) (SW = 0x2) x1x: HSE is used by System PLL 1xx: HSE is used by USB PLL
[11:8]	PLLST	PLL Clock Occupation Status xxx1: PLL is used by System Clock (CK_SYS) xx1x: PLL is used by USART x1xx: PLL is used by USB 1xxx: PLL is used by CK_REF
[2:0]	CKSWST	Clock Switch Status 00x: CK_PLL clock out as system clock 010: CK_HSE as system clock 011: CK_HSI as system clock 110: CK_LSE as system clock 111: CK_LSI as system clock The fields are status to indicate which clock source is using as system clock currently.

APB Peripheral Clock Selection Register 0 – APBPCSR0

This register specifies APB peripheral clock prescaler selection.

Offset: 0x038

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved		URPCLK		Reserved		USRCLK	
Type/Reset			RW	0	RW	0	RW	0
	23	22	21	20	19	18	17	16
	Reserved		GPTMCLK		Reserved			
Type/Reset			RW	0	RW	0		
	15	14	13	12	11	10	9	8
	BFTM1CLK		BFTM0CLK		Reserved			
Type/Reset	RW	0	RW	0	RW	0		
	7	6	5	4	3	2	1	0
	Reserved		SPICLK		Reserved		I2CCLK	
Type/Reset			RW	0	RW	0	RW	0

Bits	Field	Descriptions
[29:28]	URPCLK	UART Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[25:24]	USRCLK	USART Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[21:20]	GPTMCLK	GPTM Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[15:14]	BFTM1CLK	BFTM1 Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB/2 10: PCLK = CK_AHB/4 11: PCLK = CK_AHB/8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[13:12]	BFTM0CLK	BFTM0 Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock

Bits	Field	Descriptions
[5:4]	SPIPCLK	SPI Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[1:0]	I2CPCLK	I ² C Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock

APB Peripheral Clock Selection Register 1 – APBPCSR1

This register specifies APB peripheral clock prescaler selection.

Offset: 0x03C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	VDDRPCLK		WDTRPCLK		Reserved			
	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
Type/Reset	Reserved				EXTIPCLK		AFIOPCLK	
	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[15:14]	VDDRCLK	V _{DD} Domain Register Access Clock Selection 00: PCLK = CK_AHB / 4 01: PCLK = CK_AHB / 8 10: PCLK = CK_AHB / 16 11: PCLK = CK_AHB / 32 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[13:12]	WDTRPCLK	WDT Register Access Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock

Bits	Field	Descriptions
[3:2]	EXTIPCLK	EXTI Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[1:0]	AFIOPCLK	AFIO Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock

HSI Control Register – HSICR

This register is used to control the frequency trimming of the HSI RC oscillation.

Offset: 0x040

Reset value: 0xXXXX_0000 where X is undefined

	31	30	29	28	27	26	25	24
	Reserved				HSICOARSE			
Type/Reset				RO	X RO	X RO	X RO	X RO
	23	22	21	20	19	18	17	16
	HSIFINE							
Type/Reset	RW	X RW	X RW	X RW	X RW	X RW	X RW	X RW
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	FLOCK	REFCLKSEL		TMSEL	ATMSEL	LTRSEL	ATCEN	TRIMEN
Type/Reset	RO	0 RW	0 RW	0 RW	0 RW	0 RW	0 RW	0

Bits	Field	Descriptions
[28:24]	HSICOARSE	HSI Clock Coarse Trimming Value These bits are initialized automatically at startup. They are adjusted by factory trimming and can not be trimmed by program.
[23:16]	HSIFINE	HSI Clock Fine Trimming Value These bits are initialized automatically at startup. They are also adjusted by factory trimming. But these bits provide an additional user-programmable trimming value that is added to the HSICOARSE[4:0] bits to get more accurate or compensate the variations in voltage and temperature that influence the HSI frequency. It can be programmed by software or automatically adjusted by the Auto Trimming Controller (ATC) together with an external reference clock.
[7]	FLOCK	Frequency Lock 0: HSI frequency is not trimmed into target range 1: HSI frequency is trimmed into target range

Bits	Field	Descriptions
[6:5]	REFCLKSEL	Reference Clock Selection 00: Select 32.768 kHz external low speed clock source (LSE) 01: Select 1 kHz USB frame pulse 1x: Select external pin (CKIN) 1 kHz pulse These bits are used to select the reference clock for the HSI Auto Trimming Controller.
[4]	TMSEL	Trimming Mode Selection 0: Automatic by Auto Trimming Controller 1: Manual by user program This bit is used to select the HSI RC oscillator trimming function by ATC hardware or user programming via the HSIFINE field in this register.
[3]	ATMSEL	Automatic Trimming Mode Selection 0: Auto Trimming Controller is used binary search to approach the target range 1: Auto Trimming Controller is used linear search to approach the target range This bit is used to select the automatic trimming method by ATC hardware for the HSI RC oscillator.
[2]	LTRSEL	Lock Target Range Selection 0: 0.1 % variation 1: 0.2 % variation This bit is used to select the lock target range of the internal HSI RC oscillator trimming function for 0.1 % or 0.2 % variation.
[1]	ATCEN	ATC Enable 0: Disable Auto Trimming Controller 1: Enable Auto Trimming Controller
[0]	TRIMEN	Trimming Enable 0: HSI Trimming is disabled 1: HSI Trimming is enabled Setting this bit high enables the HSI RC oscillator trimming function by ATC hardware or user programming.

HSI Auto Trimming Counter Register – HSIATCR

This register contains the counter value of the HSI auto trimming controller

Offset: 0x044

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24						
	Reserved													
Type/Reset														
	23	22	21	20	19	18	17	16						
	Reserved													
Type/Reset														
	15	14	13	12	11	10	9	8						
	Reserved		ATCNT											
Type/Reset			RO	0	RO	0	RO	0	RO	0	RO	0	RO	0
	7	6	5	4	3	2	1	0						
	ATCNT													
Type/Reset	RO	0	RO	0	RO	0	RO	0	RO	0	RO	0	RO	0

Bits	Field	Descriptions
[13:0]	ATCNT	Auto Trimming Counter This bits contains the counter value of the HSI auto trimming controller

APB Peripheral Clock Selection Register 2 – APBPCSR2

This register specifies the APB peripheral clock prescaler selection.

Offset: 0x048

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved				PWM1PCLK		PWM0PCLK	
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2		
Type/Reset	Reserved							

Bits	Field	Descriptions
[19:18]	PWM1PCLK	PWM1 Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[17:16]	PWM0PCLK	PWM0 Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock

Low Power Control Register – LPCR

This register specifies low power control.

Offset: 0x300

Reset value: 0x0000_0000

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

Bits	Field	Descriptions
[8]	USBSLEEP	USB Sleep Software Control Enable 0: Disable USB Software Sleeping 1: Enable USB Software Sleeping Set and reset by software. Please refer to the Power Control Unit chapter for more information.

MCU Debug Control Register – MCUDBGCR

This register specifies the debug control of MCU.

Offset: 0x304

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	DBPWM1	DBPWM0	Reserved					
Type/Reset	R/W 0	R/W 0						
	23	22	21	20	19	18	17	16
	Reserved					DBUR	DBBFTM1	DBBFTM0
Type/Reset						R/W 0	R/W 0	R/W 0
	15	14	13	12	11	10	9	8
	Reserved	DBDSLP2	Reserved	DBI2C	Reserved	DBSPI	Reserved	DBUSR
Type/Reset		R/W 0		R/W 0		R/W 0		R/W 0
	7	6	5	4	3	2	1	0
	Reserved	DBGPTM	Reserved		DBWDT	DBPD	DBDSLP1	DBSLP
Type/Reset		R/W 0			R/W 0	R/W 0	R/W 0	R/W 0

Bits	Field	Descriptions
[31]	DBPWM1	PWM1 Debug Mode Enable 0: PWM1 counter continues to count even if the core is halted 1: PWM1 counter is stopped when the core is halted Set and reset by software.
[30]	DBPWM0	PWM0 Debug Mode Enable 0: PWM0 counter continues to count even if the core is halted 1: PWM0 counter is stopped when the core is halted Set and reset by software.
[18]	DBUR	UART Debug Mode Enable 0: Same behavior as in normal mode 1: UART FIFO timeout is frozen when the core is halted Set and reset by software.
[17]	DBBFTM1	BFTM1 Debug Mode Enable 0: BFTM1 counter continues to count even if the core is halted 1: BFTM1 counter is stopped when the core is halted Set and reset by software.
[16]	DBBFTM0	BFTM0 Debug Mode Enable 0: BFTM0 counter continues to count even if the core is halted 1: BFTM0 counter is stopped when the core is halted Set and reset by software.
[14]	DBDSLP2	Debug Deep-Sleep2 Mode 0: LDO = Off (but turn on DMOS), FCLK = Off and HCLK = Off in Deep-Sleep2 mode 1: LDO = On, FCLK = On and HCLK = On in Deep-Sleep2 mode Set and reset by software.
[12]	DBI2C	I ² C Debug Mode Enable 0: Same behavior as in normal mode 1: I ² C timeout is frozen when the core is halted Set and reset by software.

Bits	Field	Descriptions
[10]	DBSPI	SPI Debug Mode Enable 0: Same behavior as in normal mode 1: SPI FIFO timeout is frozen when the core is halted Set and reset by software.
[8]	DBUSR	USART Debug Mode Enable 0: Same behavior as in normal mode 1: USART FIFO timeout is frozen when the core is halted Set and reset by software.
[6]	DBGPTM	GPTM Debug Mode Enable 0: GPTM counter continues to count even if the core is halted 1: GPTM counter is stopped when the core is halted Set and reset by software.
[3]	DBWDT	Watchdog Timer Debug Mode Enable 0: Watchdog Timer counter continues to count even if the core is halted 1: Watchdog Timer counter is stopped when the core is halted Set and reset by software.
[2]	DBPD	Debug Power-Down Mode 0: LDO = Off, FCLK = Off and HCLK = Off in Power-Down mode 1: LDO = On, FCLK = On and HCLK = On in Power-Down mode Set and reset by software.
[1]	DBDSLP1	Debug Deep-Sleep1 Mode 0: LDO = Low power mode, FCLK = Off and HCLK = Off in Deep-Sleep1 mode 1: LDO = On, FCLK = On and HCLK = On in Deep-Sleep1 mode Set and reset by software.
[0]	DBSLP	Debug Sleep Mode 0: LDO = On, FCLK = On and HCLK = Off in Sleep mode 1: LDO = On, FCLK = On and HCLK = On in Sleep mode Set and reset by software.

7 Reset Control Unit (RSTCU)

Introduction

The Reset Control Unit, RSTCU, has three kinds of reset, the power-on reset, system reset and APB unit reset. The power-on reset, known as a cold reset, resets the full system during a power up. A system reset resets the processor core and peripheral IP components with the exception of the debug port controller. The resets can be triggered by an external signal, internal events and the reset generators. More information about these resets will be described in the following section.

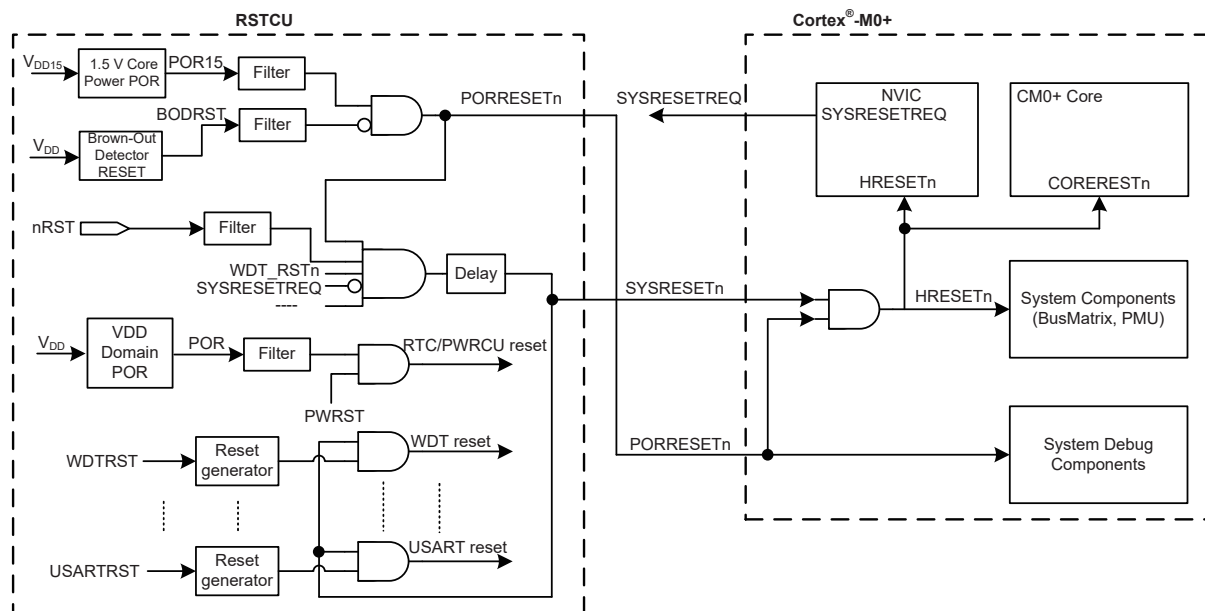


Figure 19. RSTCU Block Diagram

Functional Descriptions

Power-On Reset

The Power-on reset, POR, is generated by either an external reset or the internal reset generator. Both types have an internal filter to prevent glitches from causing erroneous reset operations. By referring to Figure 19, the POR15 active low signal will be de-asserted when the internal LDO voltage regulator is ready to provide the 1.5 V power. In addition to the POR15 signal, the Power Control Unit, PWRCU, will assert the BODF signal as a Power-Down Reset, PDR, when the BODEN bit in the LVDCSR register is set and the brown-out event occurs. For more details about the PWRCU function, refer to the PWRCU chapter.

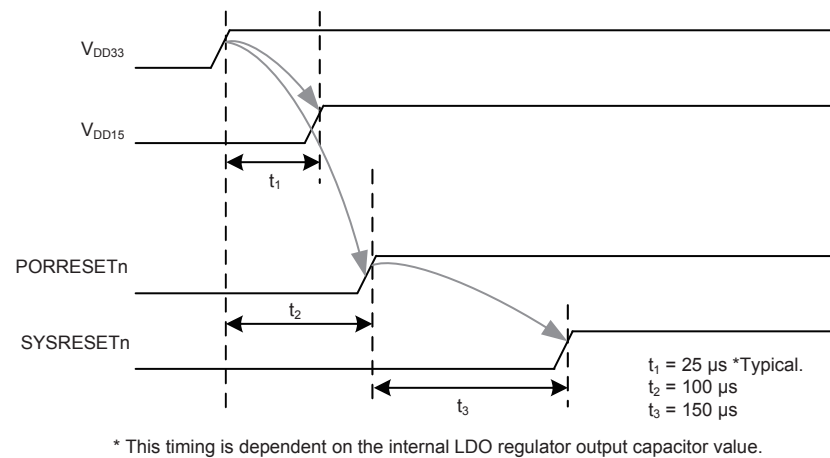


Figure 20. Power-On Reset Sequence

System Reset

A system reset is generated by a power-on reset (PORRESETEn), a Watchdog Timer reset (WDT_RSTn), a nRST pin event or a software reset (SYSRESETREQ) event. For more information about SYSRESETREQ event, refer to the related chapter in the Cortex®-M0+ reference manual.

AHB and APB Unit Reset

The AHB and APB unit reset can be divided into hardware and software resets. A hardware reset can be generated by either power on reset or system reset for all AHB and APB units. Each functional IP connected to the AHB and APB buses can be reset individually through the associated software reset bits in the RSTCU. For example, the application software can generate a USART reset via the USRRST bit in the APBPRSTR0 register.

Register Map

The following table shows the RSTCU registers and reset values.

Table 21. RSTCU Register Map

Register	Offset	Description	Reset Value
GRSR	0x100	Global Reset Status Register	0x0000_0008
AHBPRSTR	0x104	AHB Peripheral Reset Register	0x0000_0000
APBPRSTR0	0x108	APB Peripheral Reset Register 0	0x0000_0000
APBPRSTR1	0x10C	APB Peripheral Reset Register 1	0x0000_0000

Register Descriptions

Global Reset Status Register – GRSR

This register specifies a variety of reset status conditions.

Offset: 0x100

Reset value: 0x0000_0008

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved		USBRST	Reserved	PORSTF	WDTRSTF	EXTRSTF	NVICRSTF
	RW		0		WC	1	WC	0
						0	WC	0
								0

Bits	Field	Descriptions
[5]	USBRST	USB Reset Control 0: No reset 1: Reset USB This bit is set by software and cleared to 0 by hardware automatically.
[3]	PORSTF	Core 1.5 V Power On Reset Flag 0: No POR occurred 1: POR occurred This bit is set by hardware when a power-on reset occurs and reset by writing 1 into it.
[2]	WDTRSTF	Watchdog Timer Reset Flag 0: No Watchdog Timer reset occurred 1: Watchdog Timer occurred This bit is set by hardware when a watchdog timer reset occurs and reset by writing 1 into it or by hardware when a power-on reset occurs.
[1]	EXTRSTF	External Pin Reset Flag 0: No pin reset occurred 1: Pin reset occurred This bit is set by hardware when an external pin reset occurs and reset by writing 1 into it or by hardware when a power-on reset occurs.
[0]	NVICRSTF	NVIC Reset Flag 0: No NVIC asserting system reset occurred 1: NVIC asserting system reset occurred This bit is set by hardware when a system reset occurs and reset by writing 1 into it or by hardware when a power-on reset occurs.

AHB Peripheral Reset Register – AHBPRSTR

This register specifies several AHB peripherals software reset control bits.

Offset: 0x104

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							DIVRST
Type/Reset								RW 0
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	AESRST	Reserved	PFRST	Reserved		PCRST	PBRST	PARST
Type/Reset	RW 0		RW 0			RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	CRCRST	Reserved	USBRST	Reserved			DMARST	
Type/Reset	RW 0		RW 0				RW 0	

Bits	Field	Descriptions
[24]	DIVRST	Divider Reset Control 0: No reset 1: Reset Divider This bit is set by software and cleared to 0 by hardware automatically.
[15]	AESRST	AES Reset Control 0: No reset 1: Reset AES This bit is set by software and cleared to 0 by hardware automatically.
[13]	PFRST	GPIO Port F Reset Control 0: No reset 1: Reset Port F This bit is set by software and cleared to 0 by hardware automatically.
[10]	PCRST	GPIO Port C Reset Control 0: No reset 1: Reset Port C This bit is set by software and cleared to 0 by hardware automatically.
[9]	PBRST	GPIO Port B Reset Control 0: No reset 1: Reset Port B This bit is set by software and cleared to 0 by hardware automatically.
[8]	PARST	GPIO Port A Reset Control 0: No reset 1: Reset Port A This bit is set by software and cleared to 0 by hardware automatically.
[7]	CRCRST	CRC Reset Control 0: No reset 1: Reset CRC This bit is set by software and cleared to 0 by hardware automatically.

Bits	Field	Descriptions
[5]	USBRST	USB Reset Control 0: No reset 1: Reset USB This bit is set by software and cleared to 0 by hardware automatically.
[0]	DMARST	Peripheral DMA (PDMA) Reset Control 0: No reset 1: Reset Peripheral DMA (PDMA) This bit is set by software and cleared to 0 by hardware automatically.

APB Peripheral Reset Register 0 – APBPRSTR0

This register specifies several APB peripherals software reset control bits.

Offset: 0x108

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	EXTIRST	AFIORST	Reserved			URRST	Reserved	USRRST
	RW 0	RW 0				RW 0		RW 0
	7	6	5	4	3	2	1	0
Type/Reset	Reserved			SPIRST	Reserved		I2CRST	
				RW 0			RW 0	

Bits	Field	Descriptions
[15]	EXTIRST	External Interrupt Controller Reset Control 0: No reset 1: Reset EXTI This bit is set by software and cleared to 0 by hardware automatically.
[14]	AFIORST	Alternate Function I/O Reset Control 0: No reset 1: Reset Alternate Function I/O This bit is set by software and cleared to 0 by hardware automatically.
[10]	URRST	UART Reset Control 0: No reset 1: Reset UART This bit is set by software and cleared to 0 by hardware automatically.
[8]	USRRST	USART Reset Control 0: No reset 1: Reset USART This bit is set by software and cleared to 0 by hardware automatically.

Bits	Field	Descriptions
[4]	SPIRST	SPI Reset Control 0: No reset 1: Reset SPI This bit is set by software and cleared to 0 by hardware automatically.
[0]	I2CRST	I ² C Reset Control 0: No reset 1: Reset I ² C This bit is set by software and cleared to 0 by hardware automatically.

APB Peripheral Reset Register 1 – APBPRSTR1

This register specifies several APB peripherals software reset control bits.

Offset: 0x10C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved						BFTM1RST	BFTM0RST
							RW 0	RW 0
	15	14	13	12	11	10	9	8
Type/Reset	Reserved		PWM1RST	PWM0RST	Reserved			GPTMRST
			RW 0	RW 0				RW 0
	7	6	5	4	3	2	1	0
Type/Reset	Reserved			WDTRST	Reserved			
				RW 0				

Bits	Field	Descriptions
[17]	BFTM1RST	BFTM1 Reset Control 0: No reset 1: Reset BFTM1 This bit is set by software and cleared to 0 by hardware automatically.
[16]	BFTM0RST	BFTM0 Reset Control 0: No reset 1: Reset BFTM0 This bit is set by software and cleared to 0 by hardware automatically.
[16]	PWM1RST	PWM1 Reset Control 0: No reset 1: Reset PWM1 This bit is set by software and cleared to 0 by hardware automatically.
[12]	PWM0RST	PWM0 Reset Control 0: No reset 1: Reset PWM0 This bit is set by software and cleared to 0 by hardware automatically.

Bits	Field	Descriptions
[8]	GPTMRST	GPTM Reset Control 0: No reset 1: Reset GPTM This bit is set by software and cleared to 0 by hardware automatically.
[4]	WDTRST	Watchdog Timer Reset Control 0: No reset 1: Reset Watchdog Timer This bit is set by software and cleared to 0 by hardware automatically.

8 General Purpose I/O (GPIO)

Introduction

There are up to 42 General Purpose I/O ports, GPIO, named PA0 ~ PA15, PB0 ~ PB15, PC0 ~ PC7 and PF0 ~ PF1 for the device to implement the logic input/output functions. Each of the GPIO ports has related control and configuration registers to satisfy the requirement of specific applications. The actual available General Purpose I/O port numbers are dependent on the device specification and package type. Refer to the device datasheet for detailed information.

The GPIO ports are pin-shared with other alternative functions (AFs) to obtain maximum flexibility on the package pins. The GPIO pins can be used as alternative functional pins by configuring the corresponding registers regardless of the AF input or output pins.

The external interrupts on the GPIO pins of the device have related control and configuration registers in the External Interrupt Control Unit (EXTI).

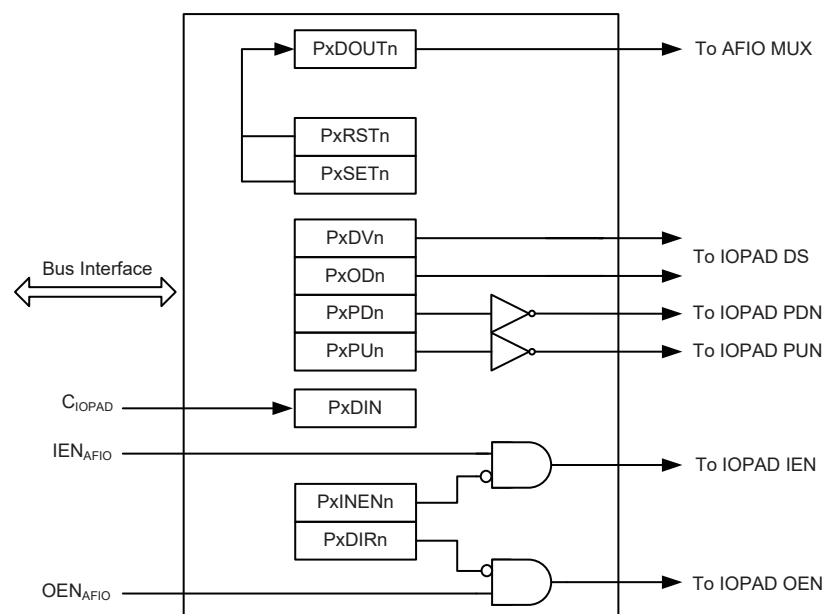


Figure 21. GPIO Block Diagram

Features

- Input/output direction control
- Schmitt Trigger Input function enable control
- Input weak pull-up/pull-down control
- Output push-pull/open-drain enable control
- Output set/reset control
- Output drive current selection
- External interrupt with programmable trigger edge – using EXTI configuration registers
- Analog input/output configurations – using AFIO configuration registers
- Alternate function input/output configurations – using AFIO configuration registers
- Port configuration lock

Functional Descriptions

Default GPIO Pin Configuration

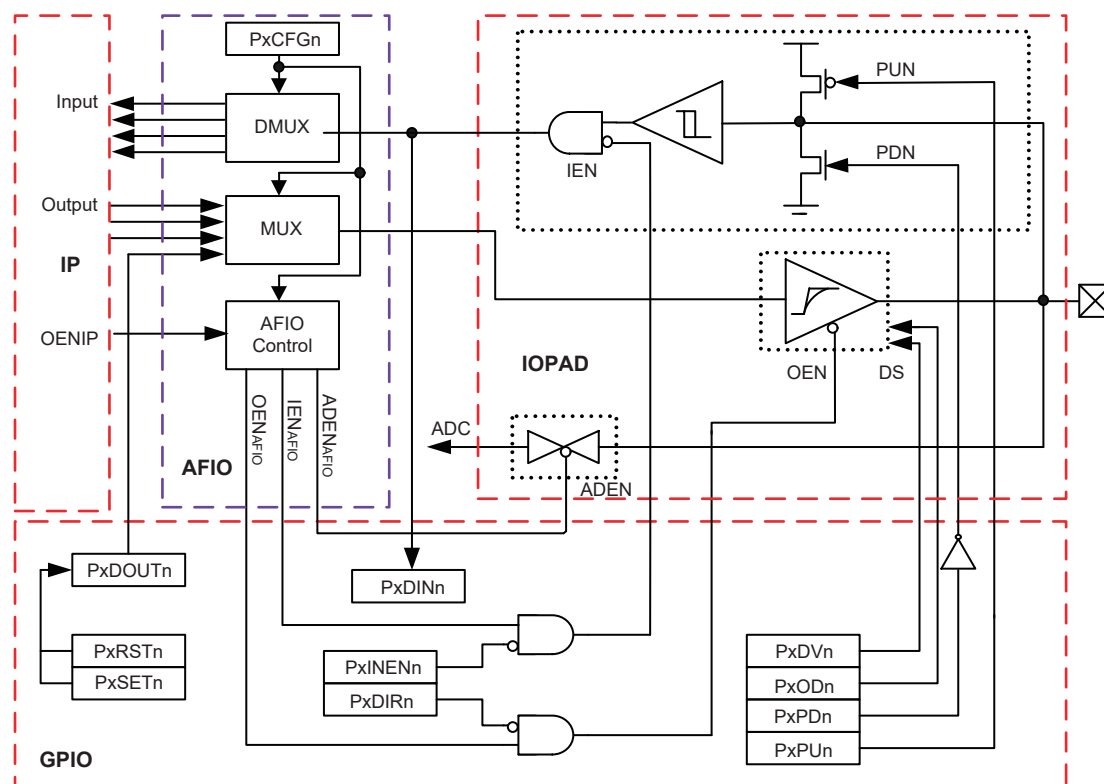
During or just after the reset period, the alternative functions are all inactive and the GPIO ports are configured into the input disable floating mode, i.e. input disabled without pull-up/pull-down resistors. Only the boot and Serial-Wired Debug pins which are pin-shared with the I/O pins are active after a device reset.

- PA9_BOOT: Input enable with internal pull-up
- SWCLK: Input enable with internal pull-up
- SWDIO: Input enable with internal pull-up

General Purpose I/O – GPIO

The GPIO pins can be configured as inputs or outputs via the data direction control registers PxDIRCR (where x = A ~ C, F). When the GPIO pins are configured as input pins, the data on the external pads can be read if the enable bits in the input enable function register PxINER are set. The GPIO pull-up/pull-down registers PxPUR/PxPDR can be configured to fit specific applications. When the pull-up and pull-down functions are both enabled, the pull-up function has the higher priority while the pull-down function will be blocked until the pull-up function is released.

The GPIO pins can be configured as output pins where the output data is latched into the data register PxDOCTR. The output type can be setup to be either push-pull or open-drain by the open-drain selection register PxODR. Only one or several specific bits of the output data will be set or reset by configuring the port output set/reset control register PxSRR or the port output reset register PxRR without affecting the unselected bits. As the port output set and reset functions are both enabled, the port output set function has the higher priority and the port output reset function will be blocked. The output driving current of the GPIO pins can be selected by configuring the drive current selection register PxDRVR.



PxDINn/PxDOUTn (x = A ~ C, F): Data Input/Data Output	PxRSTn/PxSETn (x = A ~ C, F): Reset/Set
PxDIRn (x = A ~ C, F): Direction	PxINENn (x = A ~ C, F): Input Enable
PxDVn (x = A ~ C, F): Output Drive	PxODn (x = A ~ C, F): Open-Drain
PxPDn/PxPUn (x = A ~ C, F): Pull-Down/Up	PxCFGn (x = A ~ C, F): AFIO Configuration

Table 22. AFIO, GPIO and IO Pad Control Signal True Table

Type	AFIO			GPIO		PAD		
	ADEN _{AFIO}	OEN _{AFIO}	IEN _{AFIO}	PxDIRn	PxINENn	ADEN	OEN	IEN
GPIO Input ^(Note)	1	1	1	0	1	1	1	0
GPIO Output ^(Note)	1	1	1	1	0 (1 if need)	1	0	1 (0)
AFIO Input	1	1	0	0	X	1	1	0
AFIO Output	1	0	1	X	0 (1 if need)	1	0	1 (0)
ADC Input	0	1	1	0	0 (1 if need)	0	1	1 (0)
OSC Output	0	1	1	0	0 (1 if need)	0	1	1 (0)

Note: The signals, IEN and OEN, for I/O pads are derived from the GPIO register bits PxINENn and PxDIRn respectively when the associated pin is configured in the GPIO input/output mode.

GPIO Locking Mechanism

The GPIO also offers a lock function to lock the port until a reset event occurs. The PxLOCKR (x = A ~ C, F) registers are used to lock the port x and lock control options. The value 0x5FA0 is written into the PxLKEY field in the PxLOCKR registers to freeze the PxDIRCR, PxINER, PxPUR, PxPDR, PxODR, PxDRVR control and AFIO mode configuration (GPxCFGHR or GPxCFGRLR, where x = A ~ C, F). If the value in the PxLOCKR register is 0x5FA0_0001, it means that the Port x Lock function is enabled and the Port x pin 0 is frozen.

Register Map

The following table shows the GPIO registers and reset values of the Port A ~ C, F.

Table 23. GPIO Register Map

Register	Offset	Description	Reset Value
GPIO A Base Address = 0x400B_0000			
PADIRCR	0x000	Port A Data Direction Control Register	0x0000_0000
PAINER	0x004	Port A Input Function Enable Control Register	0x0000_0200
PAPUR	0x008	Port A Pull-Up Selection Register	0x0000_3200
PAPDR	0x00C	Port A Pull-Down Selection Register	0x0000_0000
PAODR	0x010	Port A Open-Drain Selection Register	0x0000_0000
PADRVR	0x014	Port A Drive Current Selection Register	0x0000_0000
PALOCKR	0x018	Port A Lock Register	0x0000_0000
PADINR	0x01C	Port A Data Input Register	0x0000_3200
PADOUTR	0x020	Port A Data Output Register	0x0000_0000
PASRR	0x024	Port A Output Set/Reset Control Register	0x0000_0000
PARR	0x028	Port A Output Reset Control Register	0x0000_0000
GPIO B Base Address = 0x400B_2000			
PBDIRCR	0x000	Port B Data Direction Control Register	0x0000_0000
PBINER	0x004	Port B Input Function Enable Control Register	0x0000_0000
PBPUR	0x008	Port B Pull-Up Selection Register	0x0000_0000
PBPDR	0x00C	Port B Pull-Down Selection Register	0x0000_0000
PBODR	0x010	Port B Open-Drain Selection Register	0x0000_0000
PBDRVR	0x014	Port B Drive Current Selection Register	0x0000_0000
PBLOCKR	0x018	Port B Lock Register	0x0000_0000
PBDINR	0x01C	Port B Data Input Register	0x0000_0000
PBDOUTR	0x020	Port B Data Output Register	0x0000_0000
PBSRR	0x024	Port B Output Set/Reset Control Register	0x0000_0000
PBRR	0x028	Port B Output Reset Control Register	0x0000_0000
GPIO C Base Address = 0x400B_4000			
PCDIRCR	0x000	Port C Data Direction Control Register	0x0000_0000
PCINER	0x004	Port C Input Function Enable Control Register	0x0000_0000
PCPUR	0x008	Port C Pull-Up Selection Register	0x0000_0000
PCPDR	0x00C	Port C Pull-Down Selection Register	0x0000_0000
PCODR	0x010	Port C Open-Drain Selection Register	0x0000_0000
PCDRVR	0x014	Port C Drive Current Selection Register	0x0000_0000
PCLOCKR	0x018	Port C Lock Register	0x0000_0000
PCDINR	0x01C	Port C Data Input Register	0x0000_0000

Register	Offset	Description	Reset Value
PCDOUTR	0x020	Port C Data Output Register	0x0000_0000
PCSRR	0x024	Port C Output Set/Reset Control Register	0x0000_0000
PCRR	0x028	Port C Output Reset Control Register	0x0000_0000
GPIO F Base Address = 0x400B_A000			
PFDIRCR	0x000	Port F Data Direction Control Register	0x0000_0000
PFINER	0x004	Port F Input Function Enable Control Register	0x0000_0000
PFPUR	0x008	Port F Pull-Up Selection Register	0x0000_0000
PFPDR	0x00C	Port F Pull-Down Selection Register	0x0000_0000
PFODR	0x010	Port F Open-Drain Selection Register	0x0000_0000
PFDRVR	0x014	Port F Drive Current Selection Register	0x0000_0000
PFLOCKR	0x018	Port F Lock Register	0x0000_0000
PFDINR	0x01C	Port F Data Input Register	0x0000_0000
PFDOUTR	0x020	Port F Data Output Register	0x0000_0000
PFSRR	0x024	Port F Output Set/Reset Control Register	0x0000_0000
PFRR	0x028	Port F Output Reset Control Register	0x0000_0000

Register Descriptions

Port A Data Direction Control Register – PADIRCR

This register is used to control the direction of the GPIO Port A pin as input or output.

Offset: 0x000

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	PADIR							
Type/Reset	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
	PADIR							
Type/Reset	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[15:0]	PADIRn	GPIO Port A pin n Direction Control Bits (n = 0 ~ 15) 0: Pin n is in input mode 1: Pin n is in output mode

Port A Input Function Enable Control Register – PAINER

This register is used to enable or disable the GPIO Port A input function.

Offset: 0x004

Reset value: 0x0000_0200

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	PAINEN							
	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
Type/Reset	PAINEN							
	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[15:0]	PAINENn	<p>GPIO Port A pin n Input Enable Control Bits (n = 0 ~ 15)</p> <p>0: Pin n input function is disabled</p> <p>1: Pin n input function is enabled</p> <p>When the pin n input function is disabled, the input Schmitt trigger will be turned off and the Schmitt trigger output will remain at a zero state.</p>

Port A Pull-Up Selection Register – PAPUR

This register is used to enable or disable the GPIO Port A pull-up function.

Offset: 0x008

Reset value: 0x0000_3200

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	PAPU							
	RW	0	RW	0	RW	1	RW	1
	RW	0	RW	0	RW	0	RW	0
	RW	1	RW	0	RW	0	RW	1
	RW	0						0
	7	6	5	4	3	2	1	0
Type/Reset	PAPU							
	RW	0	RW	0	RW	0	RW	0
	RW	0	RW	0	RW	0	RW	0
	RW	0	RW	0	RW	0	RW	0
	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[15:0]	PAPUn	<p>GPIO Port A pin n Pull-Up Selection Control Bits (n = 0 ~ 15)</p> <p>0: Pin n pull-up function is disabled</p> <p>1: Pin n pull-up function is enabled</p> <p>Note: When the pull-up and pull-down functions are both enabled, the pull-up function will have the higher priority and therefore the pull-down function will be blocked and disabled.</p>

Port A Pull-Down Selection Register – PAPDR

This register is used to enable or disable the GPIO Port A pull-down function.

Offset: 0x00C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	PAPD								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	PAPD								
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:0]	PAPDn	<p>GPIO Port A pin n Pull-Down Selection Control Bits (n = 0 ~ 15)</p> <p>0: Pin n pull-down function is disabled</p> <p>1: Pin n pull-down function is enabled</p> <p>Note: When the pull-up and pull-down functions are both enabled, the pull-up function will have the higher priority and therefore the pull-down function will be blocked and disabled.</p>

Port A Open-Drain Selection Register – PAODR

This register is used to enable or disable the GPIO Port A open-drain function.

Offset: 0x010

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	PAOD								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	PAOD								
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:0]	PAODn	<p>GPIO Port A pin n Open-Drain Selection Control Bits (n = 0 ~ 15)</p> <p>0: Pin n Open-Drain output is disabled. (The output type is CMOS output)</p> <p>1: Pin n Open-Drain output is enabled. (The output type is open-drain output)</p> <p>Note: When the open-drain function is enabled, the pin n internal pull-up or pull-down configuration will be invalid.</p>

Port A Drive Current Selection Register – PADDRVR

This register specifies the GPIO Port A output driving current.

Offset: 0x014

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
	PADV15		PADV14		PADV13		PADV12		
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	PADV11		PADV10		PADV9		PADV8		
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	PADV7		PADV6		PADV5		PADV4		
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	PADV3		PADV2		PADV1		PADV0		
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[31:0]	PADVn[1:0]	GPIO Port A pin n Output Current Drive Selection Control Bits (n = 0 ~ 15) 00: 4 mA source/sink current 01: 8 mA source/sink current 10: 12 mA source/sink current 11: 16 mA source/sink current

Port A Lock Register – PALOCKR

This register specifies the GPIO Port A lock configuration.

Offset: 0x018

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
	PALKEY								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	PALKEY								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	PALOCK								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	PALOCK								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[31:16]	PALKEY	<p>GPIO Port A Lock Key</p> <p>0x5FA0: Port A Lock function is enabled Others: Port A Lock function is disabled</p> <p>To lock the Port A function, a value of 0x5FA0 should be written into the PALKEY field in this register. To execute a successful write operation on this lock register, the value written into the PALKEY field must be 0x5FA0. If the value written into this field is not equal to 0x5FA0, any write operations on the PALOCKR register will be aborted. The result of a read operation on the PALKEY field returns the GPIO Port A Lock Status which indicates whether the GPIO Port A is locked or not. If the read value of the PALKEY field is 0, this indicates that the GPIO Port A Lock function is disabled. Otherwise, it indicates that the GPIO Port A Lock function is enabled as the read value is equal to 1.</p>
[15:0]	PALOCKn	<p>GPIO Port A Pin n Lock Control Bits (n = 0 ~ 15)</p> <p>0: Port A Pin n is not locked 1: Port A Pin n is locked</p> <p>The PALOCKn bits are used to lock the configurations of corresponding GPIO Pins when the correct Lock Key is applied to the PALKEY field. The locked configurations including PADIRn, PAINENn, PAPUn, PAPDn, PAODn and PADVn setting in the related GPIO registers. Additionally, the GPACFGHR or GPACFGLR register which is used to configure the alternative function of the associated GPIO pin will also be locked. Note that the PALOCKR register can only be written once which means that PALKEY and PALOCKn (lock control bit) should be written together and can not be changed until a system reset or GPIO Port A reset occurs.</p>

Port A Data Input Register – PADINR

This register specifies the GPIO Port A input data.

Offset: 0x01C

Reset value: 0x0000_3200

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	PADIN							
	7	6	5	4	3	2	1	0
Type/Reset	PADIN							
	RO	0	RO	0	RO	1	RO	1
	RO	0	RO	0	RO	0	RO	1
	RO	0	RO	0	RO	0	RO	0

Bits	Field	Descriptions
[15:0]	PADINn	GPIO Port A pin n Data Input Bits (n = 0 ~ 15) 0: The input data of the corresponding pin is 0 1: The input data of the corresponding pin is 1

Port A Output Data Register – PADOUTR

This register specifies the GPIO Port A output data.

Offset: 0x020

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	PADOUT							
	7	6	5	4	3	2	1	0
Type/Reset	PADOUT							
	RW	0	RW	0	RW	0	RW	0
	RW	0	RW	0	RW	0	RW	0
	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[15:0]	PADOUTn	GPIO Port A pin n Data Output Bits (n = 0 ~ 15) 0: Data to be output on pin n is 0 1: Data to be output on pin n is 1

Port A Output Set/Reset Control Register – PASRR

This register is used to set or reset the corresponding bit of the GPIO Port A output data.

Offset: 0x024

Reset value: 0x0000_0000

	31		30		29		28		27		26		25		24	
	PARST															
Type/Reset	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0
	23		22		21		20		19		18		17		16	
	PARST															
Type/Reset	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0
	15		14		13		12		11		10		9		8	
	PASET															
Type/Reset	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0
	7		6		5		4		3		2		1		0	
	PASET															
Type/Reset	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0

Bits	Field	Descriptions
[31:16]	PARSTn	GPIO Port A pin n Output Reset Control Bits (n = 0 ~ 15) 0: No effect on the PADOUTn bit 1: Reset the PADOUTn bit Note that when the PARSTn bit in this register or (and) the PARSTn bit in the PARR register is enabled, the reset function on the PADOUTn bit will take effect.
[15:0]	PASETn	GPIO Port A pin n Output Set Control Bits (n = 0 ~ 15) 0: No effect on the PADOUTn bit 1: Set the PADOUTn bit Note that the function enabled by the PASETn bit has the higher priority if both the PASETn and PARSTn bits are set at the same time.

Port A Output Reset Register – PARR

This register is used to reset the corresponding bit of the GPIO Port A output data.

Offset: 0x028

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	PARST								
	7	6	5	4	3	2	1	0	
Type/Reset	PARST								
	WO	0	WO	0	WO	0	WO	0	WO
	0	WO	0	WO	0	WO	0	WO	0

Bits	Field	Descriptions
[15:0]	PARSTn	GPIO Port A pin n Output Reset Control Bits (n = 0 ~ 15) 0: No effect on the PADOUTn bit 1: Reset the PADOUTn bit

Port B Data Direction Control Register – PBDIRCR

This register is used to control the direction of GPIO Port B pin as input or output.

Offset: 0x000

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	PBDIR								
	RW	0	RW	0	RW	0	RW	0	RW
	0	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0	
Type/Reset	PBDIR								
	RW	0	RW	0	RW	0	RW	0	RW
	0	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[15:0]	PBDIRn	GPIO Port B pin n Direction Control Bits (n = 0 ~ 15) 0: Pin n is in input mode 1: Pin n is in output mode

Port B Input Function Enable Control Register – PBINER

This register is used to enable or disable the GPIO Port B input function.

Offset: 0x004

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	PBINEN								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	PBINEN								
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:0]	PBINENn	<p>GPIO Port B pin n Input Enable Control Bits (n = 0 ~ 15)</p> <p>0: Pin n input function is disabled</p> <p>1: Pin n input function is enabled</p> <p>When the pin n input function is disabled, the input Schmitt trigger will be turned off and the Schmitt trigger output will remain at a zero state.</p>

Port B Pull-Up Selection Register – PBPUR

This register is used to enable or disable the GPIO Port B pull-up function.

Offset: 0x008

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	PBPU								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	PBPU								
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:0]	PBPU _n	<p>GPIO Port B pin n Pull-Up Selection Control Bits (n = 0 ~ 15)</p> <p>0: Pin n pull-up function is disabled</p> <p>1: Pin n pull-up function is enabled</p> <p>Note: When the pull-up and pull-down functions are both enabled, the pull-up function will have the higher priority and therefore the pull-down function will be blocked and disabled.</p>

Port B Pull-Down Selection Register – PBPDR

This register is used to enable or disable the GPIO Port B pull-down function.

Offset: 0x00C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	PBPD								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	PBPD								
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:0]	PBPDn	<p>GPIO Port B pin n Pull-Down Selection Control Bits (n = 0 ~ 15)</p> <p>0: Pin n pull-down function is disabled</p> <p>1: Pin n pull-down function is enabled</p> <p>Note: When the pull-up and pull-down functions are both enabled, the pull-up function will have the higher priority and therefore the pull-down function will be blocked and disabled.</p>

Port B Open-Drain Selection Register – PBODR

This register is used to enable or disable the GPIO Port B open-drain function.

Offset: 0x010

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	PBOD								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	PBOD								
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:0]	PBODn	<p>GPIO Port B pin n Open-Drain Selection Control Bits (n = 0 ~ 15)</p> <p>0: Pin n Open-Drain output is disabled. (The output type is CMOS output)</p> <p>1: Pin n Open-Drain output is enabled. (The output type is open-drain output)</p> <p>Note: When the open-drain function is enabled, the pin n internal pull-up or pull-down configuration will be invalid.</p>

Port B Drive Current Selection Register – PBDVR

This register specifies the GPIO Port B output driving current.

Offset: 0x014

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
	PBDV15		PBDV14		PBDV13		PBDV12		
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	PBDV11		PBDV10		PBDV9		PBDV8		
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	PBDV7		PBDV6		PBDV5		PBDV4		
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	PBDV3		PBDV2		PBDV1		PBDV0		
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[31:0]	PBDVn[1:0]	GPIO Port B pin n Output Current Drive Selection Control Bits (n = 0 ~ 15) 00: 4 mA source/sink current 01: 8 mA source/sink current 10: 12 mA source/sink current 11: 16 mA source/sink current

Port B Lock Register – PBLCKR

This register specifies the GPIO Port B lock configuration.

Offset: 0x018

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
	PBLKEY								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	PBLKEY								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	PBLOCK								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	PBLOCK								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[31:16]	PBLKEY	<p>GPIO Port B lock Key</p> <p>0x5FA0: Port B lock function is enabled</p> <p>Others: Port B Lock function is disabled</p> <p>To lock the Port B function, a value of 0x5FA0 should be written into the PBLKEY field in this register. To execute a successful write operation on this lock register, the value written into the PBLKEY field must be 0x5FA0. If the value written into this field is not equal to 0x5FA0, any write operations on the PBLCKR register will be aborted. The result of a read operation on the PBLKEY field returns the GPIO Port B Lock Status which indicates whether the GPIO Port B is locked or not. If the read value of the PBLKEY field is 0, this indicates that the GPIO Port B Lock function is disabled. Otherwise, it indicates that the GPIO Port B Lock function is enabled as the read value is equal to 1.</p>
[15:0]	PBLOCKn	<p>GPIO Port B pin n Lock Control Bits (n = 0 ~ 15)</p> <p>0: Port B pin n is not locked</p> <p>1: Port B pin n is locked</p> <p>The PBLOCKn bits are used to lock the configurations of corresponding GPIO Pins when the correct Lock Key is applied to the PBLKEY field. The locked configurations including PBDIRn, PBINENn, PBPUn, PBPDn, PBODn and PBDVn setting in the related GPIO registers. Additionally, the GPBCFGHR or GPBCFGLR register which is used to configure the alternative function of the associated GPIO pin will also be locked. Note that the PBLCKR register can only be written once which means that PBLKEY and PBLOCKn (lock control bit) should be written together and can not be changed until a system reset or GPIO Port B reset occurs.</p>

Port B Data Input Register – PBDINR

This register specifies the GPIO Port B input data.

Offset: 0x01C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	PBDIN							
	7	6	5	4	3	2	1	0
Type/Reset	PBDIN							
	RO	0	RO	0	RO	0	RO	0
	RO	0	RO	0	RO	0	RO	0
	RO	0	RO	0	RO	0	RO	0
	RO	0	RO	0	RO	0	RO	0

Bits	Field	Descriptions
[15:0]	PBDINn	GPIO Port B pin n Data Input Bits (n = 0 ~ 15) 0: The input data of corresponding pin is 0 1: The input data of corresponding pin is 1

Port B Output Data Register – PBDOUTR

This register specifies the GPIO Port B output data.

Offset: 0x020

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	PBDOUT							
	RW	0	RW	0	RW	0	RW	0
	RW	0	RW	0	RW	0	RW	0
	RW	0	RW	0	RW	0	RW	0
	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[15:0]	PBDOUTn	GPIO Port B pin n Data Output Bits (n = 0 ~ 15) 0: Data to be output on pin n is 0 1: Data to be output on pin n is 1

Port B Output Set/Reset Control Register – PBSRR

This register is used to set or reset the corresponding bit of the GPIO Port B output data.

Offset: 0x024

Reset value: 0x0000_0000

	31		30		29		28		27		26		25		24	
	PBRST															
Type/Reset	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0
	23		22		21		20		19		18		17		16	
	PBRST															
Type/Reset	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0
	15		14		13		12		11		10		9		8	
	PBSET															
Type/Reset	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0
	7		6		5		4		3		2		1		0	
	PBSET															
Type/Reset	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0

Bits	Field	Descriptions
[31:16]	PBRSTn	GPIO Port B pin n Output Reset Control Bits (n = 0 ~ 15) 0: No effect on the PBDOUTn bit 1: Reset the PBDOUTn bit Note that when the PBRSTn bit in this register or (and) the PBRSTn bit in the PBSRR register is enabled, the reset function on the PBDOUTn bit will take effect.
[15:0]	PBSETn	GPIO Port B pin n Output Set Control Bits (n = 0 ~ 15) 0: No effect on the PBDOUTn bit 1: Set the PBDOUTn bit Note that the function enabled by the PBSETn bit has the higher priority if both the PBSETn and PBRSTn bits are set at the same time.

Port B Output Reset Register – PBRR

This register is used to reset the corresponding bit of the GPIO Port B output data.

Offset: 0x028

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	PBRST							
	WO	0	WO	0	WO	0	WO	0
	WO	0	WO	0	WO	0	WO	0
	7	6	5	4	3	2	1	0
Type/Reset	PBRST							
	WO	0	WO	0	WO	0	WO	0
	WO	0	WO	0	WO	0	WO	0

Bits	Field	Descriptions
[15:0]	PBRSTn	GPIO Port B pin n Output Reset Control Bits (n = 0 ~ 15) 0: No effect on the PBDOUTn bit 1: Reset the PBDOUTn bit

Port C Data Direction Control Register – PCDIRCR

This register is used to control the direction of GPIO Port C pin as input or output.

Offset: 0x000

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24		
Type/Reset	Reserved									
	23	22	21	20	19	18	17	16		
Type/Reset	Reserved									
	15	14	13	12	11	10	9	8		
Type/Reset	Reserved									
	7	6	5	4	3	2	1	0		
Type/Reset	PCDIR									
	RW	0	RW	0	RW	0	RW	0	RW	0
	RW	0	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[7:0]	PCDIRn	GPIO Port C pin n Direction Control Bits (n = 0 ~ 7) 0: Pin n is in input mode 1: Pin n is in output mode

Port C Input Function Enable Control Register – PCINER

This register is used to enable or disable the GPIO Port C input function.

Offset: 0x004

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	PCINEN							
	RW	0	RW	0	RW	0	RW	0
	0	RW	0	RW	0	RW	0	RW
	0	RW	0	RW	0	RW	0	RW
	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[7:0]	PCINENn	<p>GPIO Port C pin n Input Enable Control Bits (n = 0 ~ 7)</p> <p>0: Pin n input function is disabled</p> <p>1: Pin n input function is enabled</p> <p>When the pin n input function is disabled, the input Schmitt trigger will be turned off and the Schmitt trigger output will remain at a zero state.</p>

Port C Pull-Up Selection Register – PCPUR

This register is used to enable or disable the GPIO Port C pull-up function.

Offset: 0x008

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	PCPU							
	RW	0	RW	0	RW	0	RW	0
	0	RW	0	RW	0	RW	0	RW
	0	RW	0	RW	0	RW	0	RW
	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[7:0]	PCPU _n	<p>GPIO Port C pin n Pull-Up Selection Control Bits (n = 0 ~ 7)</p> <p>0: Pin n pull-up function is disabled</p> <p>1: Pin n pull-up function is enabled</p> <p>Note: When the pull-up and pull-down functions are both enabled, the pull-up function will have the higher priority and therefore the pull-down function will be blocked and disabled.</p>

Port C Pull-Down Selection Register – PCPDR

This register is used to enable or disable the GPIO Port C pull-down function.

Offset: 0x00C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	PCPD							
	RW	0	RW	0	RW	0	RW	0
	0	RW	0	RW	0	RW	0	RW
	0	RW	0	RW	0	RW	0	RW
	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[7:0]	PCPDn	<p>GPIO Port C pin n Pull-Down Selection Control Bits (n = 0 ~ 7)</p> <p>0: Pin n pull-down function is disabled</p> <p>1: Pin n pull-down function is enabled</p> <p>Note: When the pull-up and pull-down functions are both enabled, the pull-up function will have the higher priority and therefore the pull-down function will be blocked and disabled.</p>

Port C Open-Drain Selection Register – PCODR

This register is used to enable or disable the GPIO Port C open-drain function.

Offset: 0x010

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	PCOD							
	RW	0	RW	0	RW	0	RW	0
	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[7:0]	PCODn	GPIO Port C pin n Open-Drain Selection Control Bits (n = 0 ~ 7) 0: Pin n Open-Drain output is disabled. (The output type is CMOS output) 1: Pin n Open-Drain output is enabled. (The output type is open-drain output) Note: When the open-drain function is enabled, the pin n internal pull-up or pull-down configuration will be invalid.

Port C Drive Current Selection Register – PCDVR

This register specifies the GPIO Port C output driving current.

Offset: 0x014

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	PCDV7		PCDV6		PCDV5		PCDV4		
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	PCDV3		PCDV2		PCDV1		PCDV0		
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	

Bits	Field	Descriptions
[15:0]	PCDVn[1:0]	GPIO Port C pin n Output Current Drive Selection Control Bits (n = 0 ~ 7) 00: 4 mA source/sink current 01: 8 mA source/sink current 10: 12 mA source/sink current 11: 16 mA source/sink current

Port C Lock Register – PCLOCKR

This register specifies the GPIO Port C lock configuration.

Offset: 0x018

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
	PCLKEY								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	PCLKEY								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	Reserved								
Type/Reset									
	7	6	5	4	3	2	1	0	
	PCLOCK								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[31:16]	PCLKEY	GPIO Port C lock Key 0x5FA0: Port C Lock function is enable Others: Port C Lock function is disable To lock the Port C function, a value of 0x5FA0 should be written into the PCLKEY field in this register. To execute a successful write operation on this lock register, the value written into the PCLKEY field must be 0x5FA0. If the value written into this field is not equal to 0x5FA0, any write operations on the PCLOCKR register will be aborted. The result of a read operation on the PCLKEY field returns the GPIO Port C Lock Status which indicates whether the GPIO Port C is locked or not. If the read value of the PCLKEY field is 0, this indicates that the GPIO Port C Lock function is disabled. Otherwise, it indicates that the GPIO Port C Lock function is enabled as the read value is equal to 1.
[7:0]	PCLOCKn	GPIO Port C pin n Lock Control Bits (n = 0 ~ 7) 0: Port C pin n is not locked 1: Port C pin n is locked The PCLOCKn bits are used to lock the configurations of corresponding GPIO Pins when the correct Lock Key is applied to the PCLKEY field. The locked configurations including PCDIRn, PCINENn, PCPUn, PCPDn, PCODn and PCDVn setting in the related GPIO registers. Additionally, the GPCCFGHR or GPCCFGLR register which is used to configure the alternative function of the associated GPIO pin will also be locked. Note that the PCLOCKR register can only be written once which means that PCLKEY and PCLOCKn (lock control bit) should be written together and can not be changed until a system reset or GPIO Port C reset occurs.

Port C Data Input Register – PCDINR

This register specifies the GPIO Port C input data.

Offset: 0x01C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	PCDIN							
	RO	0	RO	0	RO	0	RO	0

Bits	Field	Descriptions
[7:0]	PCDINn	GPIO Port C pin n Data Input Bits (n = 0 ~ 7) 0: The input data of corresponding pin is 0 1: The input data of corresponding pin is 1

Port C Output Data Register – PCDOUR

This register specifies the GPIO Port C output data.

Offset: 0x020

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	PCDOUT							
	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[7:0]	PCDOUTn	GPIO Port C pin n Data Output Bits (n = 0 ~ 7) 0: Data to be output on pin n is 0 1: Data to be output on pin n is 1

Port C Output Set/Reset Control Register – PCSRR

This register is used to set or reset the corresponding bit of the GPIO Port C output data.

Offset: 0x024

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	PCRST								
	15	14	13	12	11	10	9	8	
Type/Reset	Reserved								
	7	6	5	4	3	2	1	0	
Type/Reset	PCSET								
	WO	0	WO	0	WO	0	WO	0	WO

Bits	Field	Descriptions
[23:16]	PCRSTn	GPIO Port C pin n Output Reset Control Bits (n = 0 ~ 7) 0: No effect on the PCDOUn bit 1: Reset the PCDOUn bit Note that when the PCRSTn bit in this register or (and) the PCRSTn bit in the PCSRR register is enabled, the reset function on the PCDOUn bit will take effect.
[7:0]	PCSETn	GPIO Port C pin n Output Set Control Bits (n = 0 ~ 7) 0: No effect on the PCDOUn bit 1: Set the PCDOUn bit Note that the function enabled by the PCSETn bit has the higher priority if both the PCSETn and PCRSTn bits are set at the same time.

Port C Output Reset Register – PCRR

This register is used to reset the corresponding bit of the GPIO Port C output data.

Offset: 0x028

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	PCRST							
	WO	0	WO	0	WO	0	WO	0

Bits	Field	Descriptions
[7:0]	PCRSTn	GPIO Port C pin n Output Reset Control Bits (n = 0 ~ 7) 0: No effect on the PCDOUTn bit 1: Reset the PCDOUTn bit

Port F Data Direction Control Register – PFDIRCR

This register is used to control the direction of GPIO Port F pin as input or output.

Offset: 0x000

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved						PFDIR	
							RW	0 RW 0

Bits	Field	Descriptions
[1:0]	PFDIRn	GPIO Port F pin n Direction Control Bits (n = 0 ~ 1) 0: Pin n is in input mode 1: Pin n is in output mode

Port F Input Function Enable Control Register – PFINER

This register is used to enable or disable the GPIO Port F input function.

Offset: 0x004

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved						PFINEN	
							RW	0 RW 0

Bits	Field	Descriptions
[1:0]	PFINENn	GPIO Port F pin n Input Enable Control Bits (n = 0 ~ 1) 0: Pin n input function is disabled 1: Pin n input function is enabled When the pin n input function is disabled, the input Schmitt trigger will be turned off and the Schmitt trigger output will remain at a zero state.

Port F Pull-Up Selection Register – PFPUR

This register is used to enable or disable the GPIO Port F pull-up function.

Offset: 0x008

Reset value: 0x0000_0000

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

Bits	Field	Descriptions
[1:0]	PFPUn	GPIO Port F pin n Pull-Up Selection Control Bits (n = 0 ~ 1) 0: Pin n pull-up function is disabled 1: Pin n pull-up function is enabled Note: When the pull-up and pull-down functions are both enabled, the pull-up function will have the higher priority and therefore the pull-down function will be blocked and disabled.

Port F Pull-Down Selection Register – PFPDR

This register is used to enable or disable the GPIO Port F pull-down function.

Offset: 0x00C

Reset value: 0x0000_0000

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

Bits	Field	Descriptions
[1:0]	PFPDn	GPIO Port F pin n Pull-Down Selection Control Bits (n = 0 ~ 1) 0: Pin n pull-down function is disabled 1: Pin n pull-down function is enabled Note: When the pull-up and pull-down functions are both enabled, the pull-up function will have the higher priority and therefore the pull-down function will be blocked and disabled.

Port F Open-Drain Selection Register – PFODR

This register is used to enable or disable the GPIO Port F open-drain function.

Offset: 0x010

Reset value: 0x0000_0000

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

Bits	Field	Descriptions
[1:0]	PFODn	GPIO Port F pin n Open-Drain Selection Control Bits (n = 0 ~ 1) 0: Pin n Open-Drain output is disabled. (The output type is CMOS output) 1: Pin n Open-Drain output is enabled. (The output type is open-drain output) Note: When the open-drain function is enabled, the pin n internal pull-up or pull-down configuration will be invalid.

Port F Drive Current Selection Register – PFDRVR

This register specifies the GPIO Port F output driving current.

Offset: 0x014

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved				PFDV1		PFDV0	
					RW	0	RW	0
						0	RW	0

Bits	Field	Descriptions
[3:0]	PFDVn[1:0]	GPIO Port F pin n Output Current Drive Selection Control Bits (n = 0 ~ 1) 00: 4 mA source/sink current 01: 8 mA source/sink current 10: 12 mA source/sink current 11: 16 mA source/sink current

Port F Lock Register – PFLOCKR

This register specifies the GPIO Port F lock configuration.

Offset: 0x018

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
	PFLKEY								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	PFLKEY								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	Reserved								
Type/Reset									
	7	6	5	4	3	2	1	0	
	Reserved						PFLOCK		
Type/Reset							RW	0	RW

Bits	Field	Descriptions
[31:16]	PFLKEY	<p>GPIO Port F Lock Key</p> <p>0x5FA0: Port F Lock function is enabled Others: Port F Lock function is disabled</p> <p>To lock the Port F function, a value of 0x5FA0 should be written into the PFLKEY field in this register. To execute a successful write operation on this lock register, the value written into the PFLKEY field must be 0x5FA0. If the value written into this field is not equal to 0x5FA0, any write operations on the PFLOCKR register will be aborted. The result of a read operation on the PFLKEY field returns the GPIO Port F Lock Status which indicates whether the GPIO Port F is locked or not. If the read value of the PFLKEY field is 0, this indicates that the GPIO Port F Lock function is disabled. Otherwise, it indicates that the GPIO Port F Lock function is enabled as the read value is equal to 1.</p>
[1:0]	PFLOCKn	<p>GPIO Port F pin n Lock Control Bits (n = 0 ~ 1)</p> <p>0: Port F pin n is not locked 1: Port F pin n is locked</p> <p>The PFLOCKn bits are used to lock the configurations of corresponding GPIO Pins when the correct Lock Key is applied to the PFLKEY field. The locked configurations including PFDIRn, PFINENn, PFPUn, PFPDn, PFODn and PFDVn setting in the related GPIO registers. Additionally, the GPF CFGHR or GPF CFGLR register which is used to configure the alternative function of the associated GPIO pin will also be locked. Note that the PFLOCKR register can only be written once which means that PFLKEY and PFLOCKn (lock control bit) should be written together and can not be changed until a system reset or GPIO Port F reset occurs.</p>

Port F Data Input Register – PFDINR

This register specifies the GPIO Port F input data.

Offset: 0x01C

Reset value: 0x0000_0000

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

Bits	Field	Descriptions
[1:0]	PFDINn	GPIO Port F pin n Data Input Bits (n = 0 ~ 1) 0: The input data of corresponding pin is 0 1: The input data of corresponding pin is 1

Port F Output Data Register – PFDOUTR

This register specifies the GPIO Port F output data.

Offset: 0x020

Reset value: 0x0000_0000

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

Bits	Field	Descriptions
[1:0]	PFDOUTn	GPIO Port F pin n Data Output Bits (n = 0 ~ 1) 0: Data to be output on pin n is 0 1: Data to be output on pin n is 1

Port F Output Set/Reset Control Register – PFSRR

This register is used to set or reset the corresponding bit of the GPIO Port F output data.

Offset: 0x024

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved						PFRST	
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved						PFSET	
							WO	0
							WO	0

Bits	Field	Descriptions
[17:16]	PFRSTn	GPIO Port F pin n Output Reset Control Bits (n = 0 ~ 1) 0: No effect on the PFDOUTn bit 1: Reset the PFDOUTn bit Note that when the PFRSTn bit in this register or (and) the PFRSTn bit in the PFSRR register is enabled, the reset function on the PFDOUTn bit will take effect.
[1:0]	PFSETn	GPIO Port F pin n Output Set Control Bits (n = 0 ~ 1) 0: No effect on the PFDOUTn bit 1: Set the PFDOUTn bit Note that the function enabled by the PFSETn bit has the higher priority if both the PFSETn and PFRSTn bits are set at the same time.

Port F Output Reset Register – PFRR

This register is used to reset the corresponding bit of the GPIO Port F output data.

Offset: 0x028

Reset value: 0x0000_0000

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

Bits	Field	Descriptions
[1:0]	PFRSTn	GPIO Port F pin n Output Reset Control Bits (n = 0 ~ 1) 0: No effect on the PFDOUTn bit 1: Reset the PFDOUTn bit

9 Alternate Function Input/Output Control Unit (AFIO)

Introduction

In order to expand the flexibility of the GPIO or the usage of peripheral functions, each I/O pin can be configured to have up to sixteen different functions such as GPIO or IP functions by setting the GPxCFGLR or GPxCFGHR register where x is the different port name. According to the usage of the IP resource and application requirements, suitable pin-out locations can be selected by using the peripheral I/O remapping mechanism. Additionally, various GPIO pins can be selected to be the EXTI interrupt line by setting the EXTIInPIN [3:0] field in the ESSRn register to trigger an interrupt or event. Please refer to the EXTI section for more details.

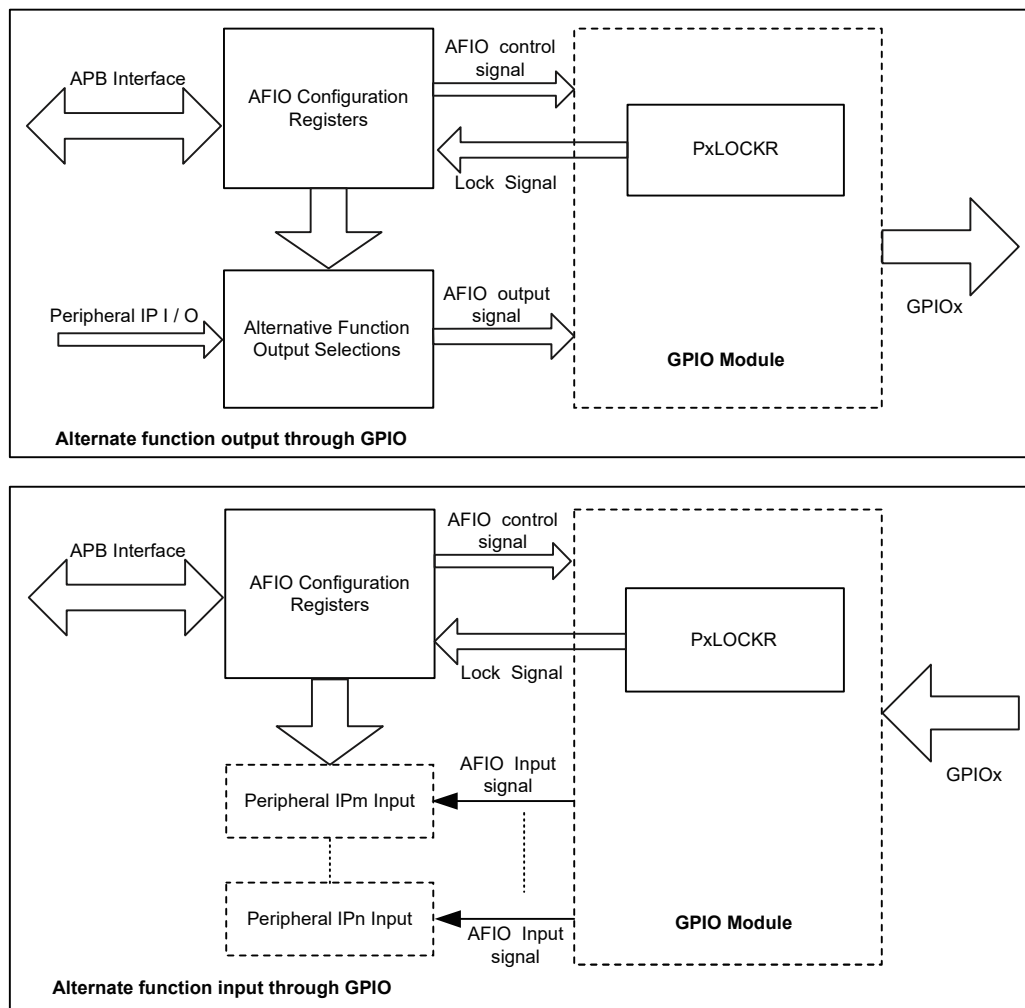


Figure 23. AFIO Block Diagram

Features

- APB slave interface for register access
- EXTI source selection
- Configurable pin function for each GPIO, up to sixteen alternative functions on each pin
- AFIO lock mechanism

Functional Descriptions

External Interrupt Pin Selection

The GPIO pins are connected to the 16 EXTI lines as shown in the accompanying figure. For example, the user can set the EXTI0PIN [3:0] field in the ESSR0 register to b0000 to select the GPIO PA0 pin as EXTI line 0 input. Since not all the pins of the Port A ~ C, F are available in all package types, refer to the pin assignment section for detailed pin information. The setting of the EXTI_nPIN [3:0] field is invalid when the corresponding pin is not available.

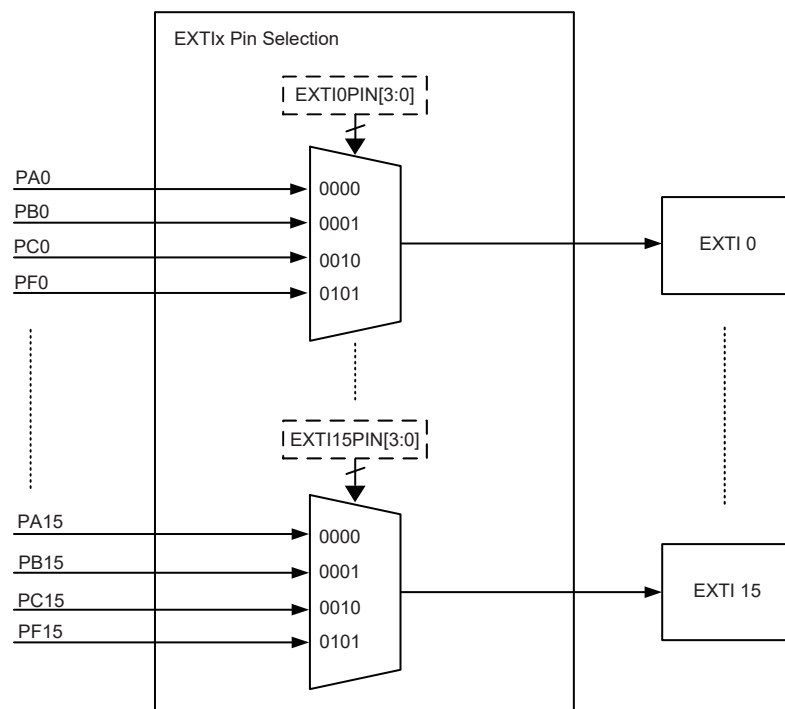


Figure 24. EXTI Channel Input Selection

Alternate Function

Up to sixteen alternative functions can be chosen for each I/O pad by setting the PxCFGn [3:0] field in the GPxCFGn (n = 0 ~ 15, x = A ~ C, F) registers. If the pin is selected as unavailable item which is noted as “N/A” in the “Alternate Function Mapping” table of the device datasheet, this pin will be defined as default alternate function. Refer to the “Alternate Function Mapping” table in the device datasheet for detailed mapping of the alternate function I/O pins. In addition to this flexible I/O multiplexing architecture, each peripheral has alternate functions mapped onto different I/O pins to optimize the number of peripherals available in smaller packages. The following description shows the setting of the PxCFGn [3:0] field.

- PxCFGn [3:0] = 0000: The default alternated function (after reset, AF0)
- PxCFGn [3:0] = 0001: Alternate Function 1 (AF1)
- PxCFGn [3:0] = 0010: Alternate Function 2 (AF2)
-
- PxCFGn [3:0] = 1110: Alternate Function 14 (AF14)
- PxCFGn [3:0] = 1111: Alternate Function 15 (AF15)

Table 24. AFIO Selection for Peripheral Map Example

AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7	AF8	AF9	AF10	AF11	AF12	AF13	AF14	AF15
System Default	GPIO	N/A	N/A	GPTM /PWM	SPI	USART /UART	I ² C	N/A	N/A	N/A	N/A	N/A	N/A	N/A	System Other

Lock Mechanism

The device also offers a lock function to lock the AFIO configuration using the GPIO lock register, PxLOCKR (x = A ~ C, F), until a reset event occurs. Refer to the GPIO Locking Mechanism section in the GPIO chapter for more details.

Register Map

The following table shows the AFIO registers and reset values.

Table 25. AFIO Register Map

Register	Offset	Description	Reset Value
ESSR0	0x000	EXTI Source Selection Register 0	0x0000_0000
ESSR1	0x004	EXTI Source Selection Register 1	0x0000_0000
GPACFGLR	0x020	GPIO Port A Configuration Low Register	0x0000_0000
GPACFGHR	0x024	GPIO Port A Configuration High Register	0x0000_0000
GPBCFGLR	0x028	GPIO Port B Configuration Low Register	0x0000_0000
GPBCFGHR	0x02C	GPIO Port B Configuration High Register	0x0000_0000
GPCCFGLR	0x030	GPIO Port C Configuration Low Register	0x0000_0000
GPCCFGHR	0x034	GPIO Port C Configuration High Register	0x0000_0000
GPFCFGLR	0x048	GPIO Port F Configuration Low Register	0x0000_0000
GPFCFGHR	0x04C	GPIO Port F Configuration High Register	0x0000_0000

Register Descriptions

EXTI Source Selection Register 0 – ESSR0

This register specifies the I/O selection of EXTI0 ~ EXTI7.

Offset: 0x000

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
	EXTI7PIN				EXTI6PIN				
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	EXTI5PIN				EXTI4PIN				
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	EXTI3PIN				EXTI2PIN				
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	EXTI1PIN				EXTI0PIN				
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[31:0]	EXTInPIN[3:0]	EXTI n Pin Selection ($n = 0 \sim 7$) 0000: PA Bit n is selected as EXTI n source signal 0001: PB Bit n is selected as EXTI n source signal 0010: PC Bit n is selected as EXTI n source signal 0101: PF Bit n is selected as EXTI n source signal Others: Reserved Note: Since not all GPIO pins are available in all products and package types, refer to the pin assignment section for detailed pin information. The EXTI n PIN [3:0] field setting is invalid when the corresponding pin is not available.

EXTI Source Selection Register 1 – ESSR1

This register specifies the IO selection of EXTI8 ~ EXTI15.

Offset: 0x004

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
	EXTI15PIN								
Type/Reset	RW	0	RW	0	RW	0	RW	0	0
	23	22	21	20	19	18	17	16	
	EXTI13PIN								
Type/Reset	RW	0	RW	0	RW	0	RW	0	0
	15	14	13	12	11	10	9	8	
	EXTI11PIN								
Type/Reset	RW	0	RW	0	RW	0	RW	0	0
	7	6	5	4	3	2	1	0	
	EXTI9PIN								
Type/Reset	RW	0	RW	0	RW	0	RW	0	0

Bits	Field	Descriptions
[31:0]	EXTInPIN[3:0]	<p>EXTIn Pin Selection (n = 8 ~ 15)</p> <p>0000: PA Bit n is selected as EXTIn source signal</p> <p>0001: PB Bit n is selected as EXTIn source signal</p> <p>0010: PC Bit n is selected as EXTIn source signal</p> <p>0101: PE Bit n is selected as EXTIn source signal</p> <p>Others: Reserved</p> <p>Note: Since not all GPIO pins are available in all products and package types, refer to the pin assignment section for detailed pin information. The EXTInPIN [3:0] field setting is invalid when the corresponding pin is not available.</p>

GPIO x Configuration Low Register – GPxCFGxLR, x = A, B, C, F

This low register specifies the alternate function of GPIO Port x, x = A, B, C, F.

Offset: 0x020, 0x028, 0x030, 0x048

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	PxCFG7				PxCFG6			
Type/Reset	RW	0	RW	0	RW	0	RW	0
	23	22	21	20	19	18	17	16
	PxCFG5				PxCFG4			
Type/Reset	RW	0	RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8
	PxCFG3				PxCFG2			
Type/Reset	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
	PxCFG1				PxCFG0			
Type/Reset	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[31:0]	PxCFGn[3:0]	<p>Alternate function selection for port x pin n (n = 0 ~ 7)</p> <p>0000: Port x pin n is selected as AF0</p> <p>0001: Port x pin n is selected as AF1</p> <p>.</p> <p>1110: Port x pin n is selected as AF14</p> <p>1111: Port x pin n is selected as AF15</p> <p>If the pin is selected as unavailable item which is noted as “N/A” in the “Alternate Function Mapping” table of the device datasheet, this pin will be defined as default alternate function. Refer to the “Alternate Function Mapping” table in the device datasheet for detailed mapping of the alternate function I/O pins.</p>

GPIO x Configuration High Register – GPxCFGHR, x = A, B, C, F

This high register specifies the alternate function of GPIO Port x, x = A, B, C, F.

Offset: 0x024, 0x02C, 0x034, 0x04C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	PxCFG15							
Type/Reset	RW	0	RW	0	RW	0	RW	0
	23	22	21	20	19	18	17	16
	PxCFG13							
Type/Reset	RW	0	RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8
	PxCFG11							
Type/Reset	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
	PxCFG9							
Type/Reset	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[31:0]	PxCFGn[3:0]	<p>Alternate function selection for port x pin n (n = 8 ~ 15)</p> <p>0000: Port x pin n is selected as AF0</p> <p>0001: Port x pin n is selected as AF1</p> <p>.</p> <p>1110: Port x pin n is selected as AF14</p> <p>1111: Port x pin n is selected as AF15</p> <p>If the pin is selected as unavailable item which is noted as “N/A” in the “Alternate Function Mapping” table of the device datasheet, this pin will be defined as default alternate function. Refer to the “Alternate Function Mapping” table in the device datasheet for detailed mapping of the alternate function I/O pins.</p>

10 Nested Vectored Interrupt Controller (NVIC)

Introduction

In order to reduce the latency and increase the interrupt processing efficiency, a tightly coupled integrated section, which is named as Nested Vectored Interrupt Controller (NVIC) is provided by the Cortex®-M0+. The NVIC controls the system exceptions and the peripheral interrupts which include functions such as the enable/disable control, priority, clear-pending, active status report, software trigger and vector table remapping. Refer to the Technical Reference Manual of Cortex®-M0+ for more details.

Additionally, an integrated simple, 24-bit down-count timer (SysTick) is provided by the Cortex®-M0+ to be used as a tick timer for the Real Time Operation System (RTOS) or as a simple counter. The SysTick counts down from the reloaded value and generates a system interrupt when it reaches zero. The accompanying table lists the system exceptions types and a variety of peripheral interrupts.

Table 26. Exception Types

Interrupt Number	Exception Number	Exception Type	Priority	Vector Address	Description
—	0	—	—	0x000	Initial Stack Point value
—	1	Reset	-3 (Highest)	0x004	Reset
-14	2	NMI	-2	0x008	Non-Maskable Interrupt. The clock stuck interrupt signal (clock monitor function provided by Clock Control Unit) is connected to the NMI input
-13	3	Hard Fault	-1	0x00C	All fault classes
-	4-10	Reserved	—	—	—
-5	11	SVCall	Configurable ⁽¹⁾	0x02C	SVC instruction System service call
-	12-13	Reserved	—	—	—
-2	14	PendSV	Configurable ⁽¹⁾	0x038	System Service Pendable request
-1	15	SysTick	Configurable ⁽¹⁾	0x03C	SysTick timer decreased to zero
0	16	LVD	Configurable ⁽²⁾	0x040	Low voltage detection interrupt
1	17	RTC	Configurable ⁽²⁾	0x044	RTC global interrupt
2	18	FMC	Configurable ⁽²⁾	0x048	FMC global interrupt
3	19	WKUP	Configurable ⁽²⁾	0x04C	EXTI event wakeup or external WAKEUP pin interrupt
4	20	EXTI0 ~ 1	Configurable ⁽²⁾	0x050	EXTI Line 0 & 1 interrupt
5	21	EXTI2 ~ 3	Configurable ⁽²⁾	0x054	EXTI Line 2 & 3 interrupt
6	22	EXTI4 ~ 15	Configurable ⁽²⁾	0x058	EXTI Line 4 ~ 15 interrupt
7	23	Reserved	—	0x05C	—
8	24	Reserved	—	0x060	—
9	25	Reserved	—	0x064	—
10	26	Reserved	—	0x068	—
11	27	Reserved	—	0x06C	—

Interrupt Number	Exception Number	Exception Type	Priority	Vector Address	Description
12	28	GPTM	Configurable ⁽²⁾	0x070	GPTM global interrupt
13	29	Reserved	—	0x074	—
14	30	Reserved	—	0x078	—
15	31	PWM0	Configurable ⁽²⁾	0x07C	PWM0 global interrupt
16	32	PWM1	Configurable ⁽²⁾	0x080	PWM1 global interrupt
17	33	BFTM0	Configurable ⁽²⁾	0x084	BFTM0 global interrupt
18	34	BFTM1	Configurable ⁽²⁾	0x088	BFTM1 global interrupt
19	35	I ² C	Configurable ⁽²⁾	0x08C	I ² C global interrupt
20	36	Reserved	—	0x090	—
21	37	SPI	Configurable ⁽²⁾	0x094	SPI global interrupt
22	38	Reserved	—	0x098	—
23	39	USART	Configurable ⁽²⁾	0x09C	USART global interrupt
24	40	Reserved	—	0x0A0	—
25	41	UART	Configurable ⁽²⁾	0x0A4	UART global interrupt
26	42	Reserved	—	0x0A8	—
27	43	Reserved	—	0x0AC	—
28	44	AES	Configurable ⁽²⁾	0x0B0	AES global interrupt
29	45	USB	Configurable ⁽²⁾	0x0B4	USB global interrupt
30	46	PDMA_CH0 ~ 1	Configurable ⁽²⁾	0x0B8	PDMA channel 0 & 1 global interrupt
31	47	PDMA_CH2 ~ 5	Configurable ⁽²⁾	0x0BC	PDMA channel 2 ~ 5 global interrupt

Notes: 1. The exception priority can be changed using the NVIC System Handler Priority Registers. For more information, refer to the ARM “Cortex®-M0+ Devices Generic User Guide” document.
2. The interrupt priority can be changed using the NVIC Interrupt Priority Registers. For more information, refer to the ARM “Cortex®-M0+ Devices Generic User Guide” document.

Features

- 7 system Cortex®-M0+ exceptions
- Up to 32 Maskable peripheral interrupts
- 4 programmable priority levels (2 bits for interrupt priority setting)
- Non-Maskable interrupt
- Low-latency exception and interrupt handling
- Vector table remapping capability
 - Integrated simple, 24-bit system timer, SysTick
 - 24-bit down-counter
 - Auto-reloading capability
 - Maskable system interrupt generation when counter decreases to 0
 - SysTick clock source derived from the HCLK clock divided by 8

Functional Descriptions

SysTick Calibration

The SysTick Calibration Value Register (SYST_CALIB) is provided by the NVIC to give a reference time base of 1ms for the RTOS tick timer or other purposes. The TENMS field in the SYST_CALIB register has a fixed value of 7500 which is the counter reload value to indicate 1 ms when the

clock source comes from the SysTick reference input clock STCLK with a frequency of 7.5 MHz (60 MHz divide by 8).

Register Map

The following table shows the NVIC registers and reset values.

Table 27. NVIC Register Map

Register	Offset	Description	Reset Value
NVIC Base Address = 0xE000_E000			
SYST_CSR	0x010	SysTick Control and Status Register	0x0000_0000
SYST_RVR	0x014	SysTick Reload Value Register	Unpredictable
SYST_CVR	0x018	SysTick Current Value Register	Unpredictable
SYST_CALIB	0x01C	SysTick Calibration Value Register	0x4000_1D4C
NVIC_ISER	0x100	Interrupt Set Enable Register	0x0000_0000
NVIC_ICER	0x180	Interrupt Clear Enable Register	0x0000_0000
NVIC_ISPR	0x200	Interrupt Set Pending Register	0x0000_0000
NVIC_ICPR	0x280	Interrupt Clear Pending Register	0x0000_0000
NVIC_IPR0	0x400	Interrupt 0 ~ 3 Priority Register	0x0000_0000
NVIC_IPR1	0x404	Interrupt 4 ~ 7 Priority Register	0x0000_0000
NVIC_IPR2	0x408	Interrupt 8 ~ 11 Priority Register	0x0000_0000
NVIC_IPR3	0x40C	Interrupt 12 ~ 15 Priority Register	0x0000_0000
NVIC_IPR4	0x410	Interrupt 16 ~ 19 Priority Register	0x0000_0000
NVIC_IPR5	0x414	Interrupt 20 ~ 23 Priority Register	0x0000_0000
NVIC_IPR6	0x418	Interrupt 24 ~ 27 Priority Register	0x0000_0000
NVIC_IPR7	0x41C	Interrupt 28 ~ 31 Priority Register	0x0000_0000
CPUID	0xD00	CPUID register	0x410C_C601
ICSR	0xD04	Interrupt Control and State Register	0x0000_0000
VTOR	0xD08	Vector Table Offset Register	0x0000_0000
AIRCR	0xD0C	Application Interrupt and Reset Control Register	0xFA05_0000
SCR	0xD10	System Control Register	0x0000_0000
CCR	0xD14	Configuration and Control Register	0x0000_0204
SHPR2	0xD1C	System Handlers Priority Register 2	0x0000_0000
SHPR3	0xD20	System Handlers Priority Register 3	0x0000_0000

Note: For more detailed descriptions of the above registers, refer to the “Cortex®-M0+ Devices Generic User Guide” document from Arm.

11 External Interrupt/Event Controller (EXTI)

Introduction

The External Interrupt/Event Controller, EXTI, comprises 16 edge detectors which can generate wakeup events or interrupt requests independently. In the interrupt mode there are five trigger types which can be selected as the external interrupt trigger type, low level, high level, negative edge, positive edge and both edges, selectable using the SRCnTYPE field in the EXTICFGRn (n = 0 ~ 15) register. In the wakeup event mode, the wakeup event polarity can be configured by setting the EXTInWPOL (n = 0 ~ 15) field in the EXTIWAKUPPOLR register. If the EVWUPIEN bit in the EXTIWAKUPCR Register is set, the EVWUP interrupt can be generated when the associated wakeup event occurs and the corresponding EXTI wakeup enable bit is set. Each EXTI line can also be masked independently.

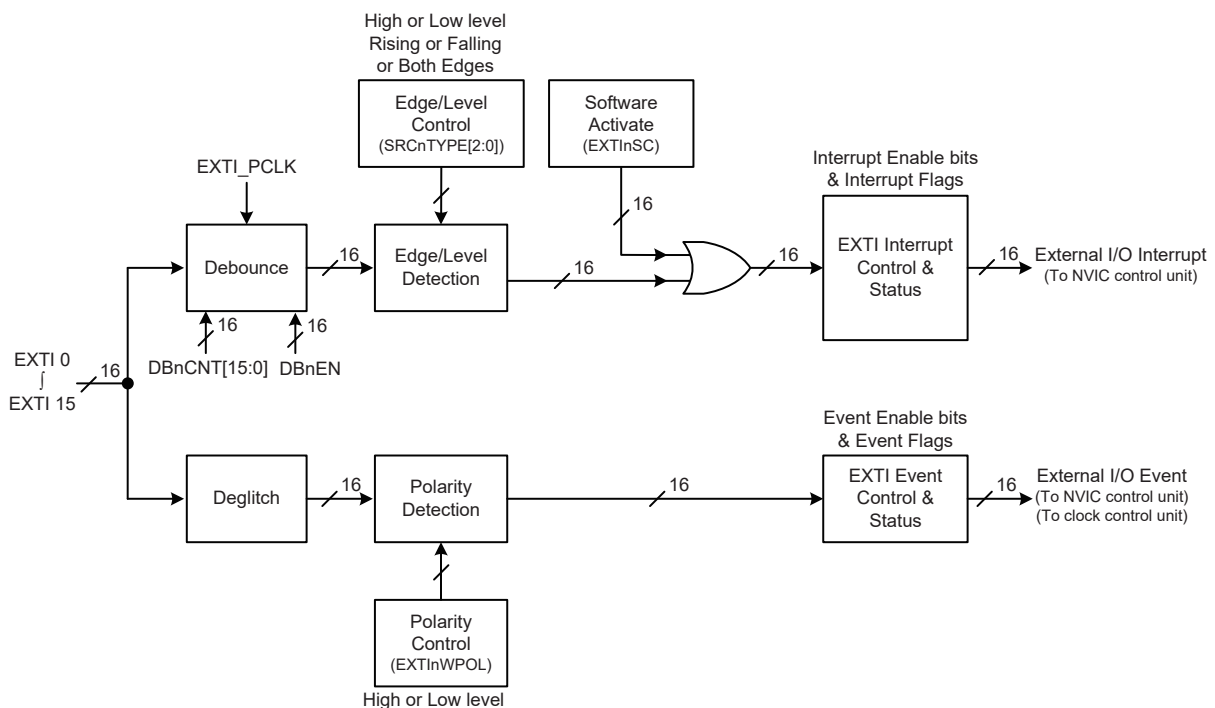


Figure 25. EXTI Block Diagram

Features

- Up to 16 EXTI lines with configurable trigger source and type
 - All GPIO pins can be selected as EXTI trigger source
 - Source trigger type includes high level, low level, negative edge, positive edge or both edge
- Individual interrupt enable, wakeup enable and status bits for each EXTI line
- Software interrupt trigger mode for each EXTI line
- Integrated deglitch filter for short pulse blocking

Functional Descriptions

Wakeup Event Management

In order to wake up the system from the power saving mode, the EXTI controller provides a function which can monitor external events and send them to the CPU core and the Clock Control Unit, CKCU. These external events include EXTI events, Low Voltage Detection, WAKEUP input pin, USB and RTC wakeup functions. By configuring the wakeup event enable bit in the corresponding peripheral, the wakeup signal will be sent to the CPU core and the CKCU via the EXTI controller when the corresponding wakeup event occurs. Additionally, the software can enable the event wakeup interrupt function by setting the EVWUPIEN bit in the EXTIWAKUPCR register and the EXTI controller will then assert an interrupt when the wakeup event occurs.

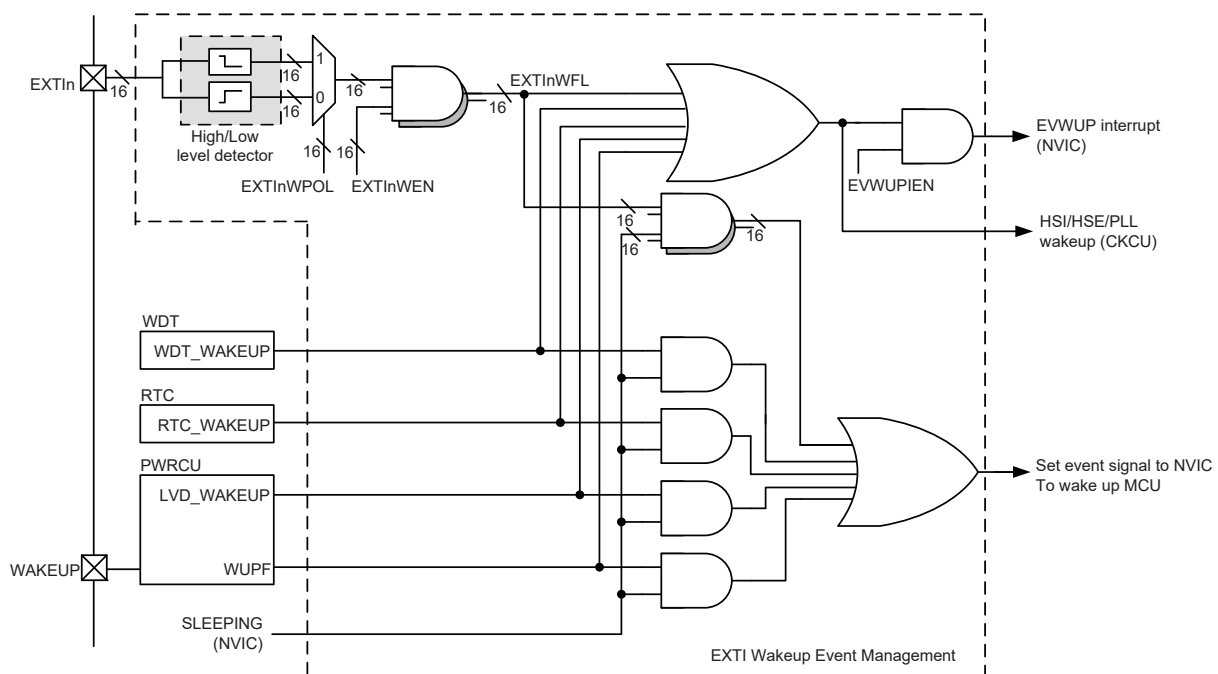


Figure 26. EXTI Wakeup Event Management

External Interrupt/Event Line Mapping

All GPIO pins can be selected as EXTI trigger sources by configuring the `EXTInPIN[3:0]` field in the `AFIO_ESSRn` ($n = 0 \sim 1$) register to trigger an interrupt or event. Refer to the AFIO section for more details.

Interrupt and Debounce

The application software can set the `DBnEN` bit in the `EXTIn` Interrupt Configuration Register `EXTICFGRn` ($n = 0 \sim 15$) to enable the corresponding pin debounce function and configure the `DBnCNT` field in the `EXTICFGRn` register so as to select an appropriate debounce time for specific applications. The interrupt signal will however be delayed due to the debounce function. When the device is woken up from the power saving mode by an external interrupt, an interrupt request will be generated by the EXTI wakeup flag. After the device has been woken up and the clock has

recovered, the EXTI wakeup flag that was triggered by the EXTI line must be read and then cleared by application software. The accompanying diagram shows the relationship between the EXTI input signal and the EXTI interrupt/event request signal.

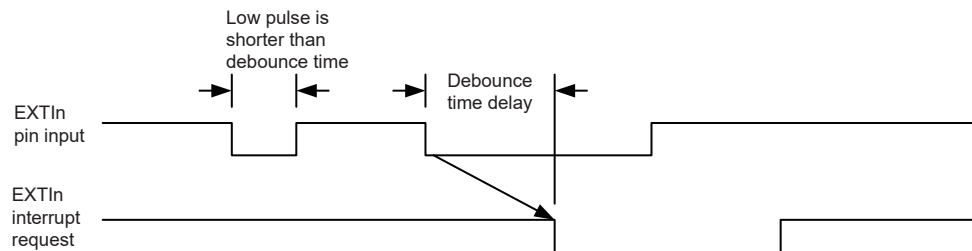


Figure 27. EXTI Interrupt Debounce Function

Register Map

The following table shows the EXTI registers and reset values.

Table 28. EXTI Register Map

Register	Offset	Description	Reset Value
EXTICFGR0	0x000	EXTI Interrupt 0 Configuration Register	0x0000_0000
EXTICFGR1	0x004	EXTI Interrupt 1 Configuration Register	0x0000_0000
EXTICFGR2	0x008	EXTI Interrupt 2 Configuration Register	0x0000_0000
EXTICFGR3	0x00C	EXTI Interrupt 3 Configuration Register	0x0000_0000
EXTICFGR4	0x010	EXTI Interrupt 4 Configuration Register	0x0000_0000
EXTICFGR5	0x014	EXTI Interrupt 5 Configuration Register	0x0000_0000
EXTICFGR6	0x018	EXTI Interrupt 6 Configuration Register	0x0000_0000
EXTICFGR7	0x01C	EXTI Interrupt 7 Configuration Register	0x0000_0000
EXTICFGR8	0x020	EXTI Interrupt 8 Configuration Register	0x0000_0000
EXTICFGR9	0x024	EXTI Interrupt 9 Configuration Register	0x0000_0000
EXTICFGR10	0x028	EXTI Interrupt 10 Configuration Register	0x0000_0000
EXTICFGR11	0x02C	EXTI Interrupt 11 Configuration Register	0x0000_0000
EXTICFGR12	0x030	EXTI Interrupt 12 Configuration Register	0x0000_0000
EXTICFGR13	0x034	EXTI Interrupt 13 Configuration Register	0x0000_0000
EXTICFGR14	0x038	EXTI Interrupt 14 Configuration Register	0x0000_0000
EXTICFGR15	0x03C	EXTI Interrupt 15 Configuration Register	0x0000_0000
EXTICR	0x040	EXTI Interrupt Control Register	0x0000_0000
EXTIEDGEFLGR	0x044	EXTI Interrupt Edge Flag Register	0x0000_0000
EXTIEDGESR	0x048	EXTI Interrupt Edge Status Register	0x0000_0000
EXTISSCR	0x04C	EXTI Interrupt Software Set Command Register	0x0000_0000
EXTIWAKUPCR	0x050	EXTI Interrupt Wakeup Control Register	0x0000_0000
EXTIWAKUPPOLR	0x054	EXTI Interrupt Wakeup Polarity Register	0x0000_0000
EXTIWAKUPFLG	0x058	EXTI Interrupt Wakeup Flag Register	0x0000_0000

Register Descriptions

EXTI Interrupt n Configuration Register – EXTICFGRn, n = 0 ~ 15

This register is used to specify the debounce function and select the trigger type.

Offset: 0x000 (0) ~ 0x03C (15)

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	DBnEN	SRCnTYPE				Reserved		
Type/Reset	RW 0	RW 0	RW 0	RW 0				
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	DBnCNT							
	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
Type/Reset	DBnCNT							
	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions																								
[31]	DBnEN	EXTIn Debounce Circuit Enable Bit (n = 0 ~ 15) 0: Debounce circuit is disabled 1: Debounce circuit is enabled																								
[30:28]	SRCnTYPE	EXTIn Interrupt Source Trigger Type (n = 0 ~ 15) <table><tr><th colspan="3">SRCnTYPE [2:0]</th><th>Interrupt Source Type</th></tr><tr><td>0</td><td>0</td><td>0</td><td>Low-level Sensitive</td></tr><tr><td>0</td><td>0</td><td>1</td><td>High-level Sensitive</td></tr><tr><td>0</td><td>1</td><td>0</td><td>Negative-edge Triggered</td></tr><tr><td>0</td><td>1</td><td>1</td><td>Positive-edge Triggered</td></tr><tr><td>1</td><td>X</td><td>X</td><td>Both-edge Triggered</td></tr></table>	SRCnTYPE [2:0]			Interrupt Source Type	0	0	0	Low-level Sensitive	0	0	1	High-level Sensitive	0	1	0	Negative-edge Triggered	0	1	1	Positive-edge Triggered	1	X	X	Both-edge Triggered
SRCnTYPE [2:0]			Interrupt Source Type																							
0	0	0	Low-level Sensitive																							
0	0	1	High-level Sensitive																							
0	1	0	Negative-edge Triggered																							
0	1	1	Positive-edge Triggered																							
1	X	X	Both-edge Triggered																							
[15:0]	DBnCNT	EXTIn Debounce Counter (n = 0 ~ 15) The debounce time is calculated with DBnCNT × APB clock (EXTI_PCLK) period and should be long enough to take effect on the input signal.																								

EXTI Interrupt Control Register – EXTICR

This register is used to control the EXTI interrupt.

Offset: 0x040

Reset value: 0x0000_0000

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

Bits	Field	Descriptions
[15:0]	EXTInEN	EXTIn Interrupt Enable Bit (n = 0 ~ 15) 0: EXTI line n interrupt is disabled 1: EXTI line n interrupt is enabled

EXTI Interrupt Edge Flag Register – EXTIEDGEFLGR

This register is used to indicate if an EXTI edge has been detected.

Offset: 0x044

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	EXTI15EDF	EXTI14EDF	EXTI13EDF	EXTI12EDF	EXTI11EDF	EXTI10EDF	EXTI9EDF	EXTI8EDF	
	WC	0	WC	0	WC	0	WC	0	WC
	7	6	5	4	3	2	1	0	
Type/Reset	EXTI7EDF	EXTI6EDF	EXTI5EDF	EXTI4EDF	EXTI3EDF	EXTI2EDF	EXTI1EDF	EXTI0EDF	
	WC	0	WC	0	WC	0	WC	0	WC

Bits	Field	Descriptions
[15:0]	EXTInEDF	<p>EXTIn Edge Detection Flag (n = 0 ~ 15)</p> <p>0: No edge is detected</p> <p>1: Positive or negative edge is detected</p> <p>This bit is set by the hardware circuitry when a positive or negative edge is detected on the corresponding EXTI line. Software should write 1 to clear it.</p>

EXTI Interrupt Edge Status Register – EXTIEDGESR

This register indicates the polarity of a detected EXTI edge.

Offset: 0x048

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	EXTI15EDS	EXTI14EDS	EXTI13EDS	EXTI12EDS	EXTI11EDS	EXTI10EDS	EXTI9EDS	EXTI8EDS	
	WC	0	WC	0	WC	0	WC	0	WC
	7	6	5	4	3	2	1	0	
Type/Reset	EXTI7EDS	EXTI6EDS	EXTI5EDS	EXTI4EDS	EXTI3EDS	EXTI2EDS	EXTI1EDS	EXTI0EDS	
	WC	0	WC	0	WC	0	WC	0	WC

Bits	Field	Descriptions
[15:0]	EXTInEDS	EXTIn Edge Detection Status (n = 0 ~ 15) 0: Negative edge is detected 1: Positive edge is detected Software should write 1 to clear it.

EXTI Interrupt Software Set Command Register – EXTISSCR

This register is used to activate the EXTI interrupt.

Offset: 0x04C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	EXTI15SC	EXTI14SC	EXTI13SC	EXTI12SC	EXTI11SC	EXTI10SC	EXTI9SC	EXTI8SC	
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	EXTI7SC	EXTI6SC	EXTI5SC	EXTI4SC	EXTI3SC	EXTI2SC	EXTI1SC	EXTI0SC	
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:0]	EXTInSC	EXTIn Software Set Command (n = 0 ~ 15) 0: Deactivates the corresponding EXTI interrupt 1: Activates the corresponding EXTI interrupt

EXTI Interrupt Wakeup Control Register – EXTIWAKUPCR

This register is used to control the EXTI interrupt and wakeup function.

Offset: 0x050

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	EVWUPIEN		Reserved					
Type/Reset	RW	0						
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	EXTI15WEN		EXTI14WEN		EXTI13WEN		EXTI12WEN	
Type/Reset	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
	EXTI7WEN		EXTI6WEN		EXTI5WEN		EXTI4WEN	
Type/Reset	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[31]	EVWUPIEN	EXTI Event Wakeup Interrupt Enable Bit 0: Disable EVWUP interrupt 1: Enable EVWUP interrupt
[15:0]	EXTInWEN	EXTIn Wakeup Enable Bit (n = 0 ~ 15) 0: Power saving mode wakeup is disabled 1: Power saving mode wakeup is enabled

EXTI Interrupt Wakeup Polarity Register – EXTIWAKUPPOLR

This register is used to select the EXTI line interrupt wakeup polarity.

Offset: 0x054

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	EXTI15WPOL	EXTI14WPOL	EXTI13WPOL	EXTI12WPOL	EXTI11WPOL	EXTI10WPOL	EXTI9WPOL	EXTI8WPOL	
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	EXTI7WPOL	EXTI6WPOL	EXTI5WPOL	EXTI4WPOL	EXTI3WPOL	EXTI2WPOL	EXTI1WPOL	EXTI0WPOL	
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:0]	EXTInWPOL	EXTIn Wakeup Polarity (n = 0 ~ 15) 0: EXTIn wakeup is high level active 1: EXTIn wakeup is low level active

EXTI Interrupt Wakeup Flag Register – EXTIWAKUPFLG

This register is the EXTI interrupt wakeup flag register.

Offset: 0x058

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	EXTI15WFL	EXTI14WFL	EXTI13WFL	EXTI12WFL	EXTI11WFL	EXTI10WFL	EXTI9WFL	EXTI8WFL
	WC	0	WC	0	WC	0	WC	0
	7	6	5	4	3	2	1	0
Type/Reset	EXTI7WFL	EXTI6WFL	EXTI5WFL	EXTI4WFL	EXTI3WFL	EXTI2WFL	EXTI1WFL	EXTI0WFL
	WC	0	WC	0	WC	0	WC	0

Bits	Field	Descriptions
[15:0]	EXTInWFL	EXTIn Wakeup Flag (n = 0 ~ 15) 0: No wakeup occurs 1: System is woken up by EXTIn Software should write 1 to clear it.

12 General-Purpose Timer (GPTM)

Introduction

The General-Purpose Timer consists of one 16-bit up/down-counter, four 16-bit Capture/Compare Registers (CCRs), one 16-bit Counter-Reload Register (CRR) and several control/status registers. It can be used for a variety of purposes including general timer, input signal pulse width measurement or output waveform generation such as single pulse generation or PWM output. The GPTM supports an encoder interface using a quadrature decoder with two inputs.

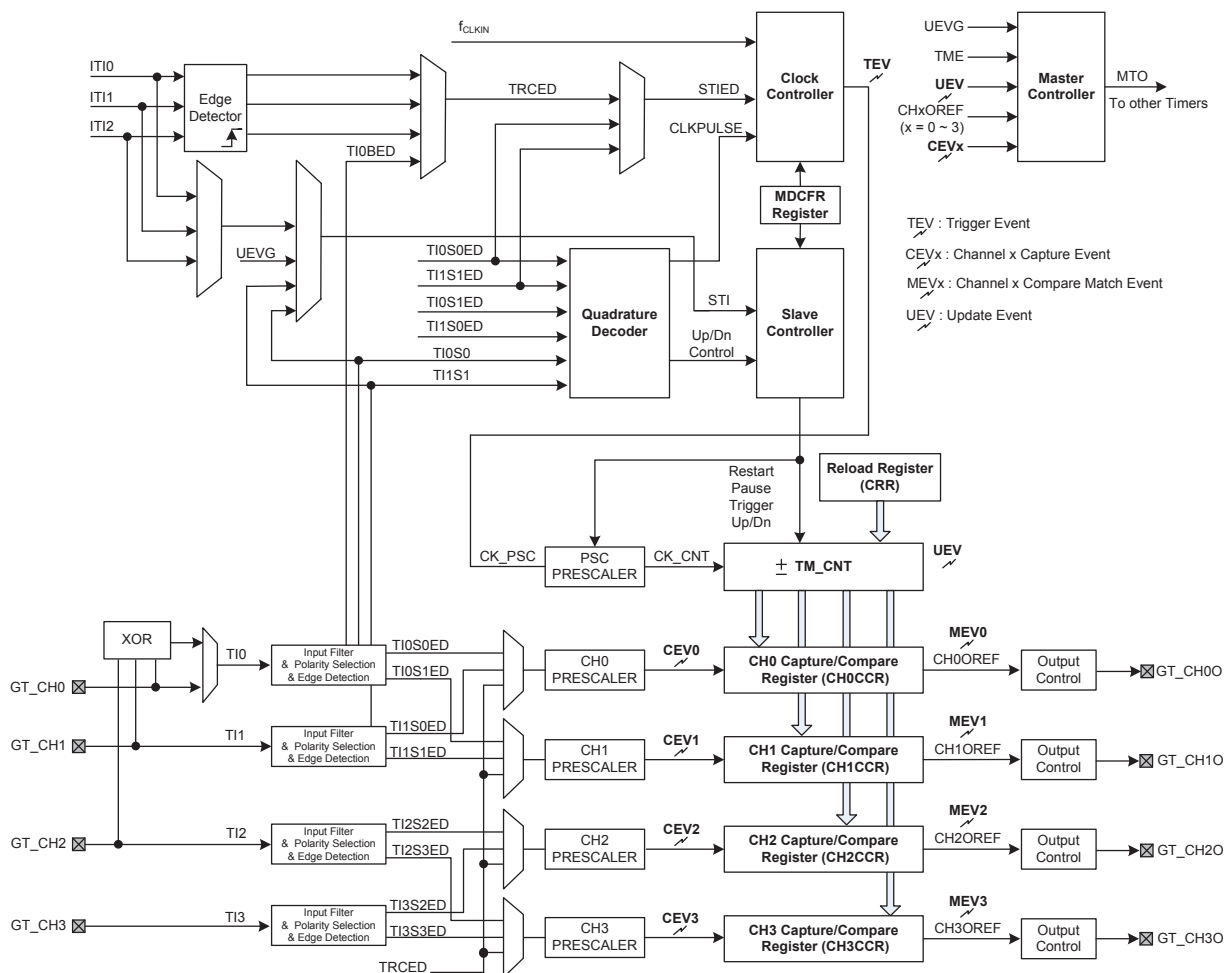


Figure 28. GPTM Block Diagram

Features

- 16-bit up/down auto-reload counter
- 16-bit programmable prescaler that allows division of the prescaler clock source by any factor between 1 and 65536 to generate the counter clock frequency
- Up to 4 independent channels for:
 - Input Capture function
 - Compare Match Output
 - Generation of PWM waveform – Edge and Center-aligned Mode
 - Single Pulse Mode Output
- Encoder interface controller with two inputs using quadrature decoder
- Synchronization circuit to control the timer with external signals and to interconnect several timers together
- Interrupt/PDMA generation with the following events:
 - Update event
 - Trigger event
 - Input capture event
 - Output compare match event
- GPTM Master/Slave mode controller

Functional Descriptions

Counter Mode

Up-Counting

In this mode the counter counts continuously from 0 to the counter-reload value, which is defined in the CRR register, in a count-up direction. Once the counter reaches the counter-reload value, the Timer Module generates an overflow event and the counter restarts to count once again from 0. This action will continue repeatedly. The counting direction bit DIR in the CNTCFR register should be set to 0 for the up-counting mode.

When the update event is generated by setting the UEVG bit in the EVGR register to 1, the counter value will also be initialized to 0.

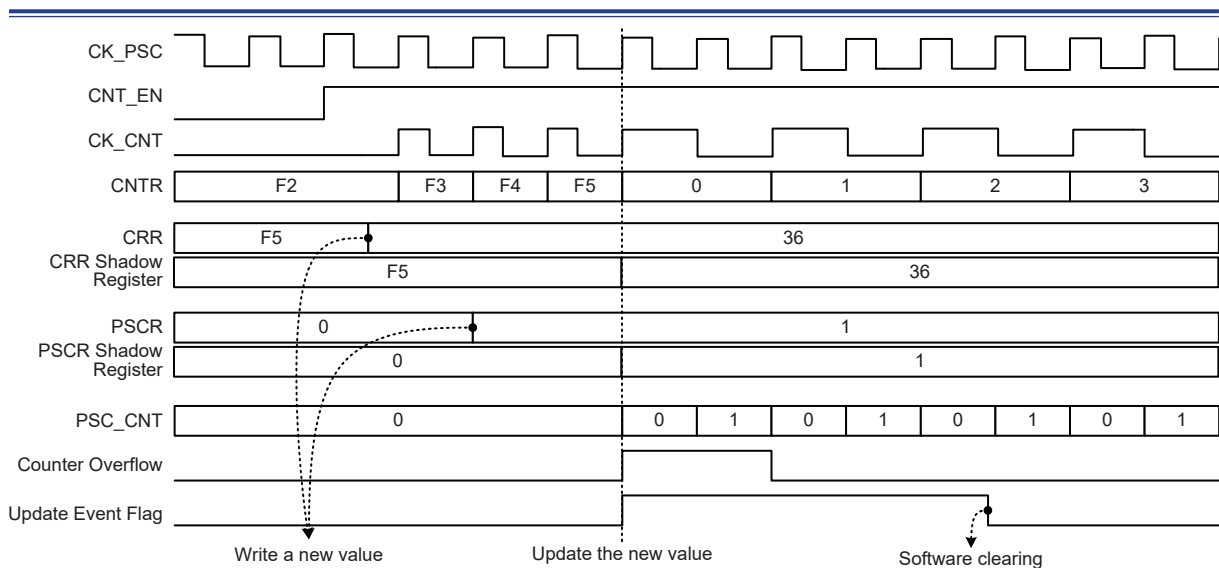


Figure 29. Up-counting Example

Down-Counting

In this mode the counter counts continuously from the counter-reload value, which is defined in the CRR register, to 0 in a count-down direction. Once the counter reaches 0, the Timer module generates an underflow event and the counter restarts to count once again from the counter-reload value. This action will continue repeatedly. The counting direction bit DIR in the CNTCFR register should be set to 1 for the down-counting mode.

When the update event is set by the UEVG bit in the EVGR register, the counter value will also be initialized to the counter-reload value.

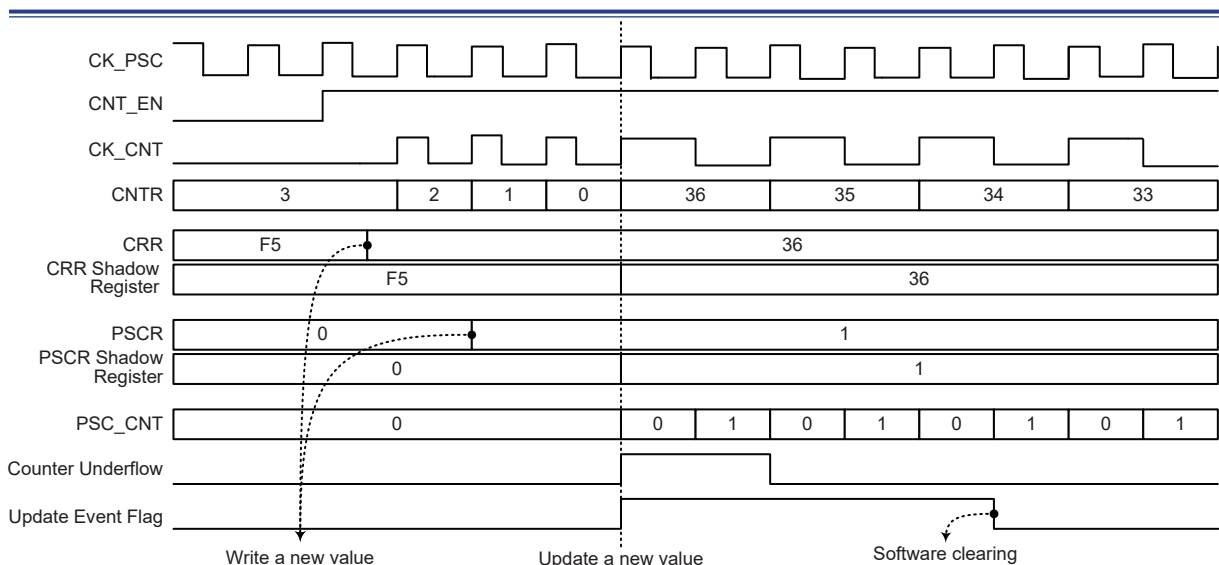


Figure 30. Down-counting Example

Center-Aligned Counting

In the center-aligned counting mode, the counter counts up from 0 to the counter-reload value and then counts down to 0 alternatively. The Timer module generates an overflow event when the counter counts to the counter-reload value in the up-counting mode and generates an underflow event when the counter counts to 0 in the down-counting mode. The counting direction bit DIR in the CNTCFR register is read-only and indicates the counting direction when in the center-align mode. The counting direction is updated by hardware automatically.

Setting the UEVG bit in the EVGR register will initialize the counter value to 0 irrespective of whether the counter is counting up or down in the center-align counting mode.

The UEVIF bit in the INTSR register will be set to 1 when an overflow or underflow event or both of them occur according to the CMSEL field setting in the CNTCFR register.

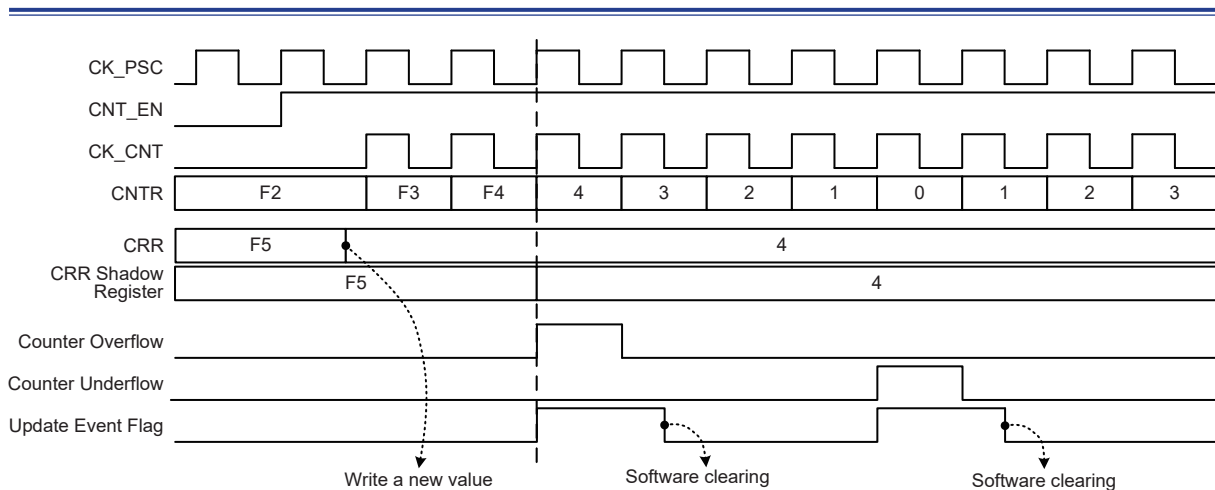


Figure 31. Center-aligned Counting Example

Clock Controller

The following describes the Timer Module clock controller which determines the clock source of the internal prescaler counter.

■ Internal APB clock f_{CLKIN} :

The default internal clock source is the APB clock f_{CLKIN} used to drive the counter prescaler when the slave mode is disabled. When the slave mode selection bits SMSEL in the MDCFR register are set to 0x4, 0x5 or 0x6, the internal APB clock f_{CLKIN} is the counter prescaler driving clock source. If the slave mode controller is enabled by setting SMSEL field in the MDCFR register to an available value including 0x1, 0x2, 0x3 and 0x7, the prescaler is clocked by other clock sources selected by the TRSEL field in the TRCFR register and described as follows.

■ Quadrature Decoder:

To select Quadrature Decoder mode the SMSEL field should be set to 0x1, 0x2 or 0x3 in the MDCFR register. The Quadrature Decoder function uses two input states of the GT_CH0 and GT_CH1 pins to generate the clock pulse to drive the counter prescaler. The counting direction bit DIR is modified by hardware automatically at each transition on the input source signal. The input source signal can be derived from the GT_CH0 pin only, the GT_CH1 pin only or both GT_CH0 and GT_CH1 pins.

■ STIED:

The counter prescaler can count during each rising edge of the STI signal. This mode can be selected by setting the SMSEL field to 0x7 in the MDCFR register. Here the counter will act as an event counter. The input event, known as STI here, can be selected by setting the TRSEL field to an available value except the value of 0x0. When the STI signal is selected as the clock source, the internal edge detection circuitry will generate a clock pulse during each STI signal rising edge to drive the counter prescaler. It is important to note that if the TRSEL field is set to 0x0 to select the software UEVG bit as the trigger source, then when the SMSEL field is set to 0x7, the counter will be updated instead of counting.

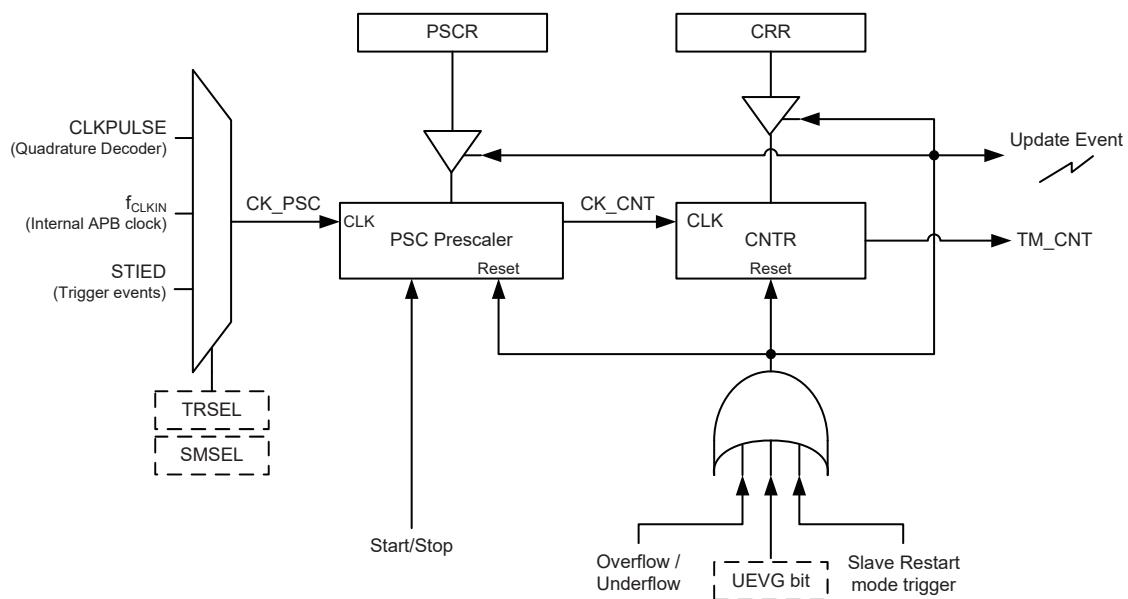


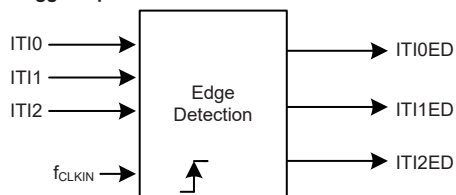
Figure 32. GPTM Clock Source Selection

Trigger Controller

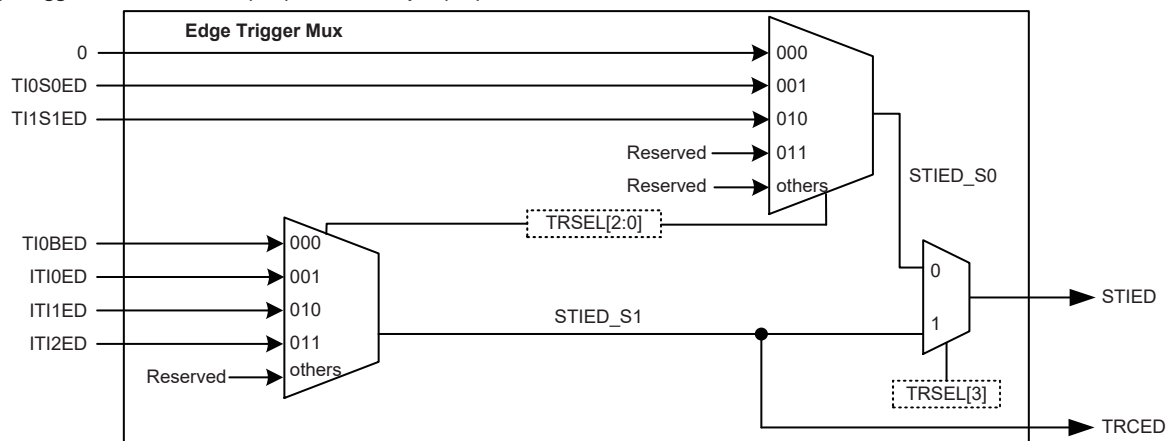
The trigger controller is used to select the trigger source and setup the trigger level or trigger edge condition. For the internal trigger input, it can be selected by the Trigger Selection bits TRSEL in the TRCFR register. For all the trigger sources except the UEVG bit software trigger, the internal edge detection circuitry will generate a clock pulse at each trigger signal rising edge to stimulate some GPTM functions which are triggered by a trigger signal rising edge.

Trigger Controller Block = Edge Trigger Mux + Level Trigger Mux

Internal Trigger Input



Edge Trigger Source = Internal (ITIx) + Channel input (TIn)



Level Trigger Source = Internal (ITIx) + Channel input (TIn) + Software UEVG bit

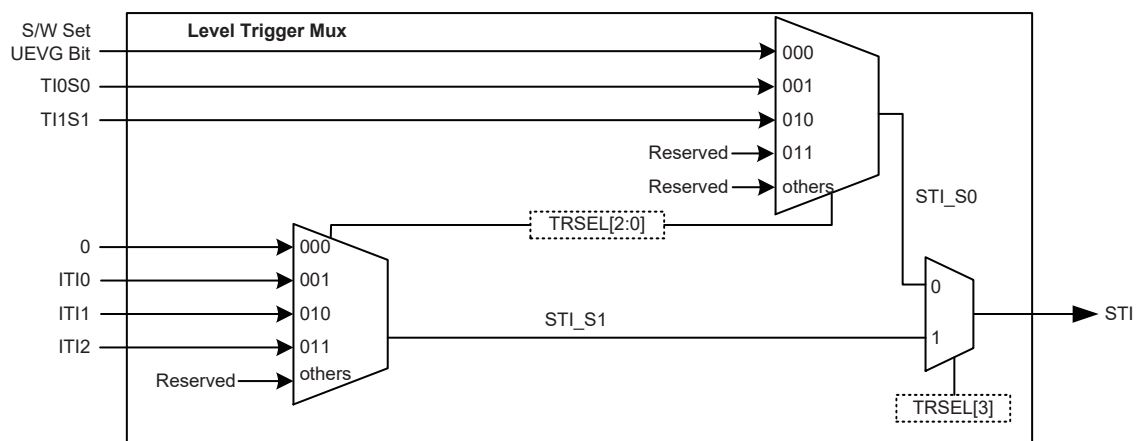


Figure 33. Trigger Control Block

Slave Controller

The GPTM can be synchronized with an external trigger in several modes including the Restart mode, the Pause mode and the Trigger mode which can be selected by the SMSEL field in the MDCFR register. The trigger input of these modes comes from the STI signal which is selected by the TRSEL field in the TRCFR register. The operation modes in the Slave Controller are described in the accompanying sections.

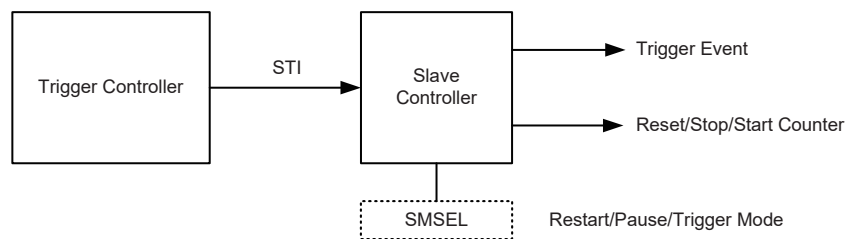


Figure 34. Slave Controller Diagram

Restart Mode

The counter and its prescaler can be reinitialized in response to a rising edge of the STI signal. When a STI rising edge occurs, the update event software generation bit named UEVG will automatically be asserted by hardware and the trigger event flag will also be set. Then the counter and prescaler will be reinitialized. Although the UEVG bit is set to 1 by hardware, the update event does not really occur. It depends upon whether the update event disable control bit UEVDIS is set to 1 or not. If the UEVDIS is set to 1 to disable the update event to occur, there will no update event be generated, however the counter and prescaler are still reinitialized when the STI rising edge occurs. If the UEVDIS bit in the CNTCFR register is cleared to enable the update event to occur, an update event will be generated together with the STI rising edge, then all the preloaded registers will be updated.

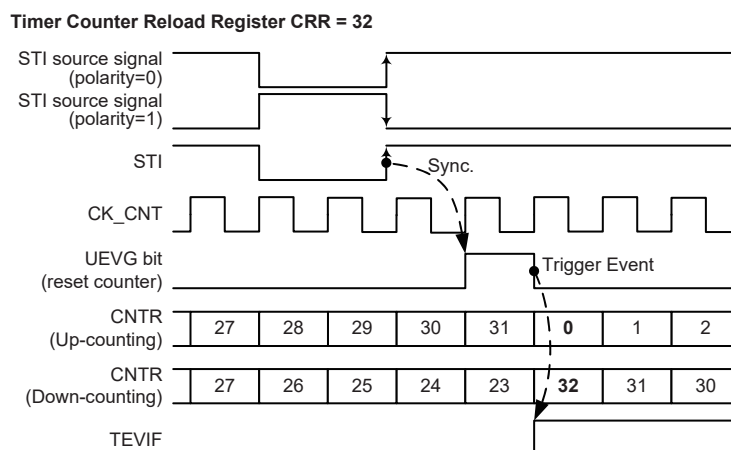


Figure 35. GPTM in Restart Mode

Pause Mode

In the Pause Mode, the selected STI input signal level is used to control the counter start/stop operation. The counter starts to count when the selected STI signal is at a high level and stops counting when the STI signal is changed to a low level, here the counter will maintain its present value and will not be reset. Since the Pause function depends upon the STI level to control the counter stop/start operation, the selected STI trigger signal can not be derived from the TI0BED signal.

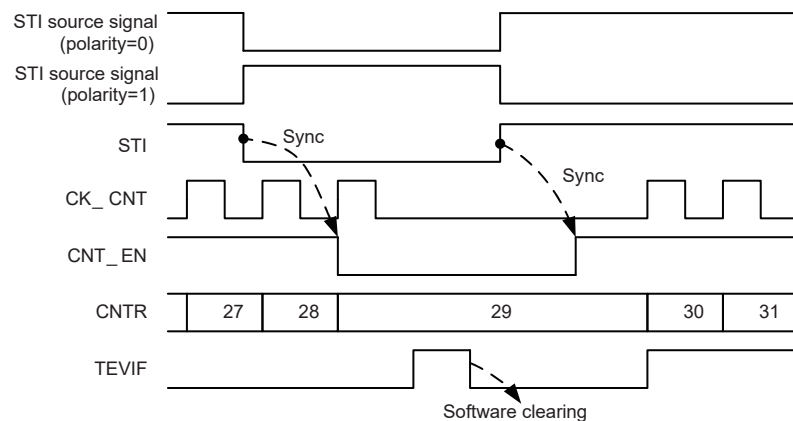


Figure 36. GPTM in Pause Mode

Trigger Mode

After the counter is disabled to count, the counter can resume counting when a STI rising edge signal occurs. When an STI rising edge occurs, the counter will start to count from the current value in the counter. Note that if the STI signal is selected to be derived from the UEVG bit software trigger, the counter will not resume counting. When software triggering using the UEVG bit is selected as the STI source signal, there will be no clock pulse generated which can be used to make the counter resume counting. Note that the STI signal is only used to enable the counter to resume counting and has no effect on controlling the counter to stop counting.

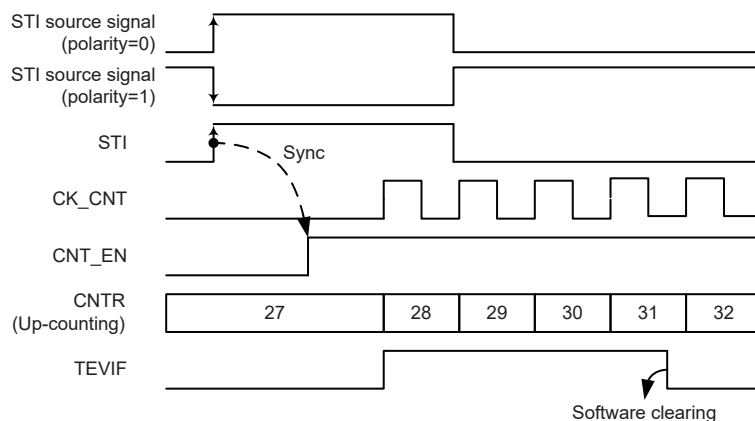


Figure 37. GPTM in Trigger Mode

Master Controller

The GPTMs and MCTMs can be linked together internally for timer synchronization or chaining. When one GPTM is configured to be in the Master Mode, the GPTM Master Controller will generate a Master Trigger Output (MTO) signal which includes a reset, a start, a stop signal or a clock source which is selected by the MMSEL field in the MDCFR register to trigger or drive another GPTM or MCTM, if exists, which is configured in the Slave Mode.

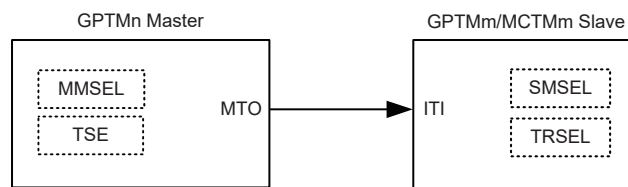


Figure 38. Master GPTMn and Slave GPTMm/MCTMm Connection

The Master Mode Selection field, MMSEL, in the MDCFR register is used to select the MTO source for synchronizing another slave GPTM or MCTM if exists.

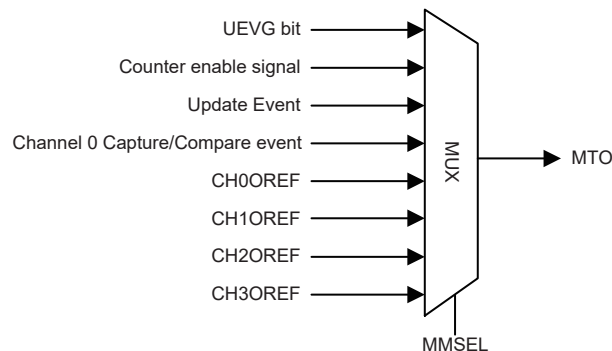


Figure 39. MTO Selection

For example, setting the MMSEL field to 0x5 is to select the CH1OREF signal as the MTO signal to synchronize another slave GPTM or MCTM. For a more detailed description, refer to the related MMSEL field definitions in the MDCFR register.

Channel Controller

The GPTM has four independent channels which can be used as capture inputs or compare match outputs. Each capture input or compare match output channel is composed of a preload register and a shadow register. Data access of the APB bus is always implemented by reading/writing the preload register.

When used in the input capture mode, the counter value is captured into the CHxCCR shadow register first and then transferred into the CHxCCR preload register when the capture event occurs.

When used in the compare match output mode, the contents of the CHxCCR preload register is copied into the associated shadow register, the counter value is then compared with the register value.

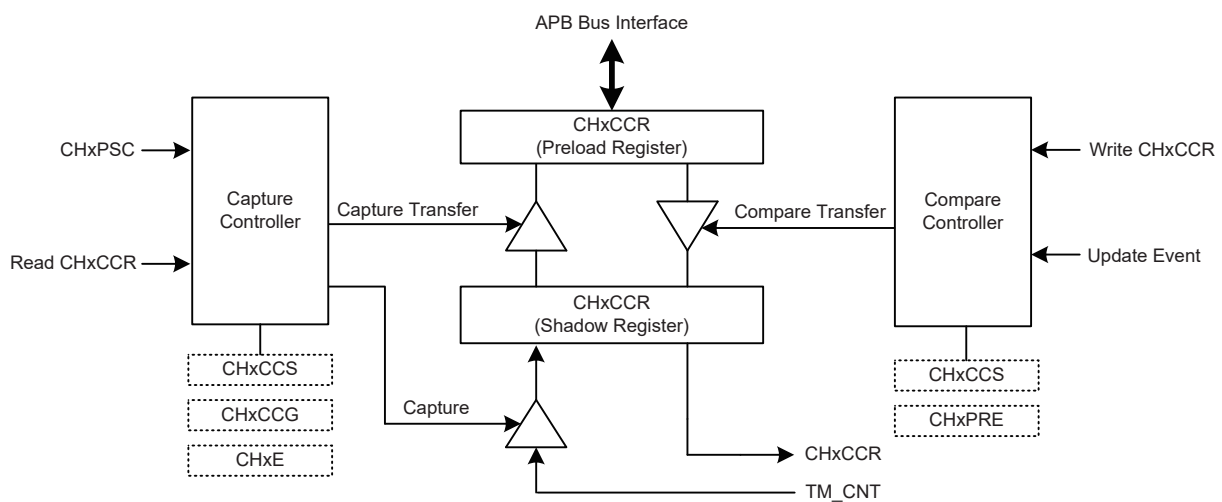


Figure 40. Capture/Compare Block Diagram

Capture Counter Value Transferred to CHxCCR

When the channel is used as a capture input, the counter value is captured into the Channel Capture/Compare Register (CHxCCR) when an effective input signal transition occurs. Once the capture event occurs, the CHxCCIF flag in the INTSR register is set accordingly. If the CHxCCIF bit is already set, i.e., the flag has not yet been cleared by software, and another capture event on this channel occurs, the corresponding channel Over-Capture flag, named CHxOCF, will be set.

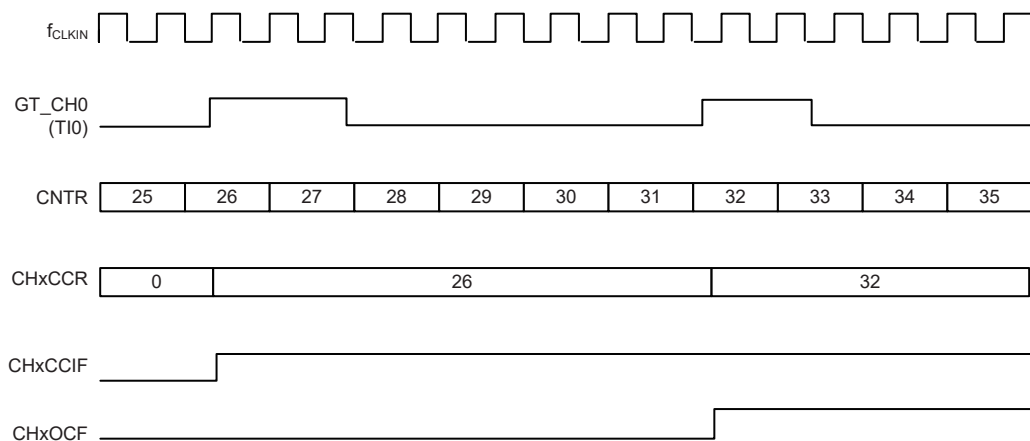


Figure 41. Input Capture Mode

Pulse Width Measurement

The input capture mode can be also used for pulse width measurement from signals on the GT_CHx pins, TIx . The following example shows how to configure the GPTM operated in the input capture mode to measure the high pulse width and the input period on the GT_CH0 pin using channel 0 and channel 1. The basic steps are shown as follows.

- Configure the capture channel 0 ($CH0CCS = 0x1$) to select the $TI0$ signal as the capture input.
- Configure the $CH0P$ bit to 0 to choose the rising edge of the $TI0$ input as the active polarity.
- Configure the capture channel 1 ($CH1CCS = 0x2$) to select the $TI0$ signal as the capture input.
- Configure the $CH1P$ bit to 1 to choose the falling edge of the $TI0$ input as the active polarity.
- Configure the $TRSEL$ bits to $0x1$ to select $TI0S0$ as the trigger input.
- Configure the Slave controller to operate in the Restart mode by setting the $SMSSEL$ field in the $MDCFR$ register to $0x4$.
- Enable the input capture mode by setting the $CH0E$ and $CH1E$ bits in the $CHCTR$ register to 1.

As the following diagram shows, the high pulse width on the GT_CH0 pin will be captured into the $CH1CCR$ register while the input period will be captured into the $CH0CCR$ register after input capture operation.

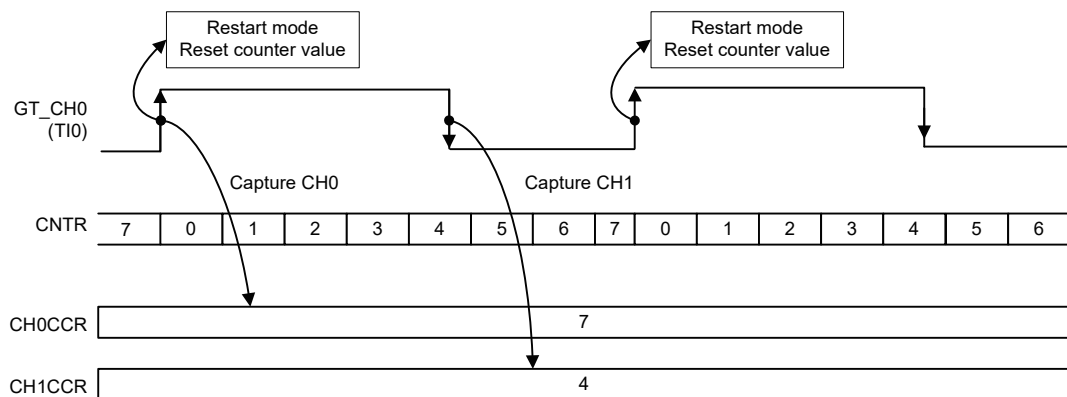


Figure 42. PWM Pulse Width Measurement Example

Input Stage

The input stage consists of a digital filter, a channel polarity selection, edge detection and a channel prescaler. The channel 0 input signal (TIO) can be chosen to come from the GT_CH0 signal or the Exclusive-OR function of the GT_CH0, GT_CH1 and GT_CH2 signals. The channel input signal (Tix) is sampled by a digital filter to generate a filtered input signal TixFP. Then the channel polarity and the edge detection block can generate a TixS0ED or TixS1ED signal for the input capture function. The effective input event number can be set by the channel capture input source prescaler setting field, CHxPSC.

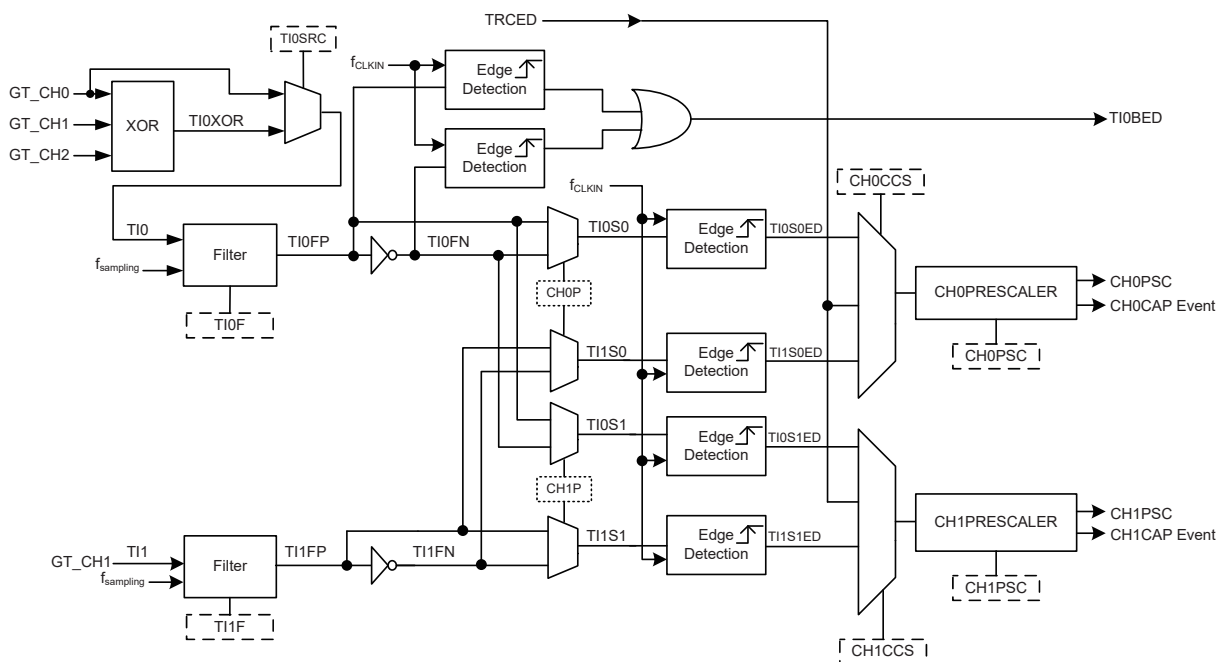
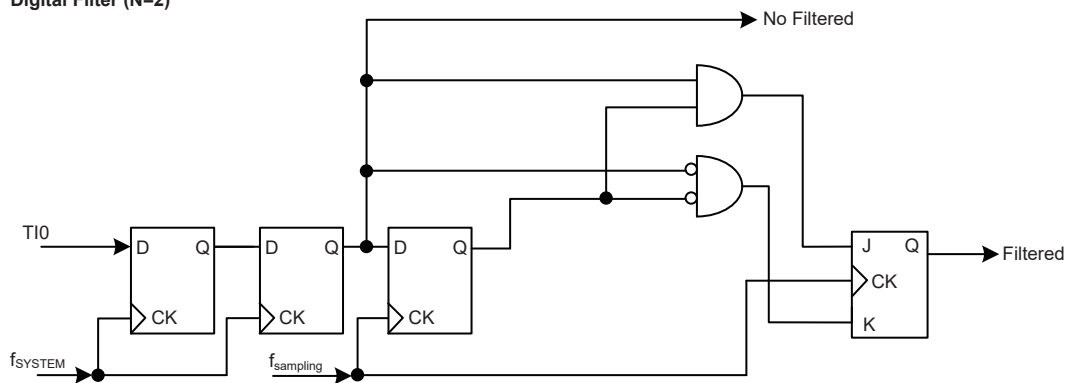


Figure 43. Channel 0 and Channel 1 Input Stages

Digital Filter

Digital Filter (N=2)



November 22, 2019

Quadrature Decoder

The Quadrature Decoder function uses two quadrantal inputs TI0 and TI1 derived from the GT_CH0 and GT_CH1 pins respectively to interact to generate the counter value. The DIR bit is modified by hardware automatically during each input source transition. The input source can be either TI0 only, TI1 only or both TI0 and TI1, the selection made by setting the SMSEL field to 0x1, 0x2 or 0x3. The mechanism for changing the counter direction is shown in the following table. The Quadrature decoder can be regarded as an external clock with a directional selection. This means that the counter counts continuously in the interval between 0 and the counter-reload value. Therefore, users must configure the CRR register before the counter starts to count.

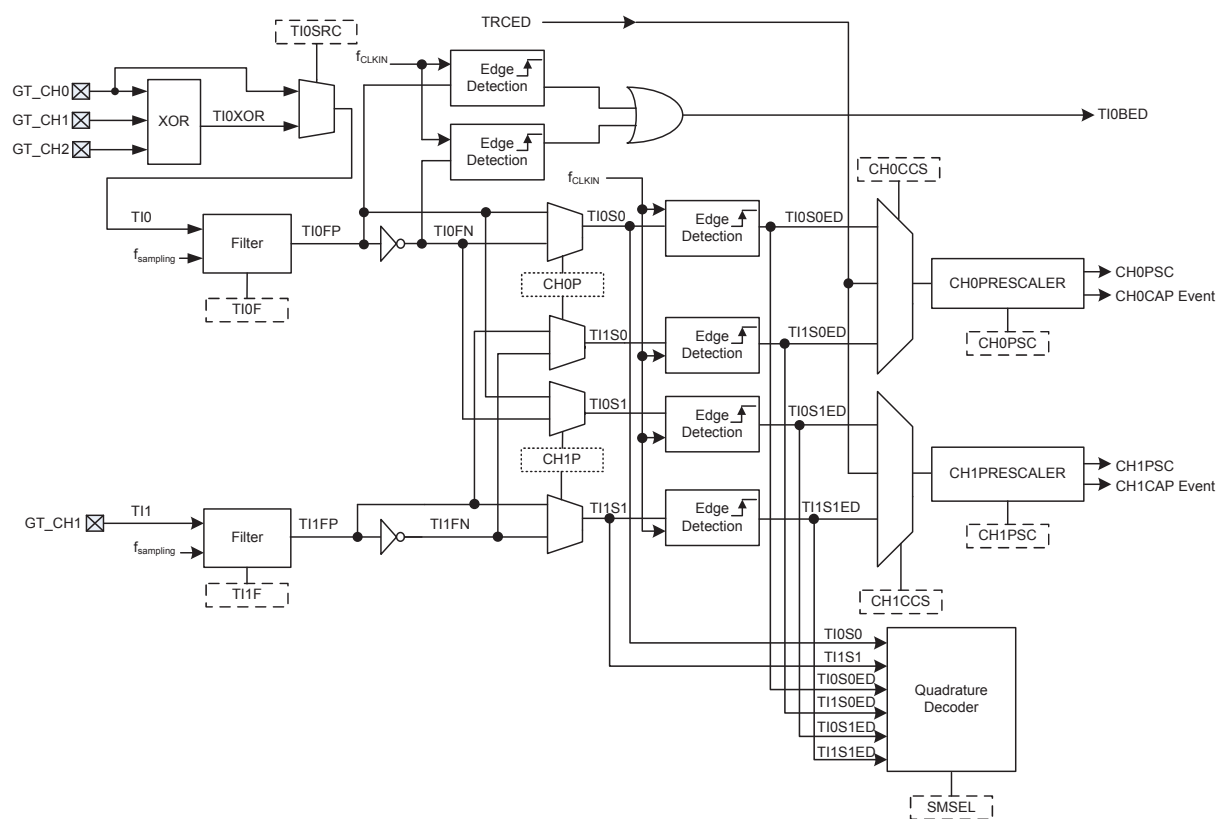


Figure 46. Input Stage and Quadrature Decoder Block Diagram

Table 29. Counting Direction and Encoding Signals

Counting Mode	Level	TI0S0		TI1S1	
		Rising	Falling	Rising	Falling
Counting on TI0 only (SMSEL = 0x1)	TI1S1 = High	Down	Up	—	—
	TI1S1 = Low	Up	Down	—	—
Counting on TI1 only (SMSEL = 0x2)	TI0S0 = High	—	—	Up	Down
	TI0S0 = Low	—	—	Down	Up
Counting on TI0 and TI1 (SMSEL = 0x3)	TI1S1 = High	Down	Up	X	X
	TI1S1 = Low	Up	Down	X	X
	TI0S0 = High	X	X	Up	Down
	TI0S0 = Low	X	X	Down	Up

Note: “—” → means “no counting”; “X” → impossible

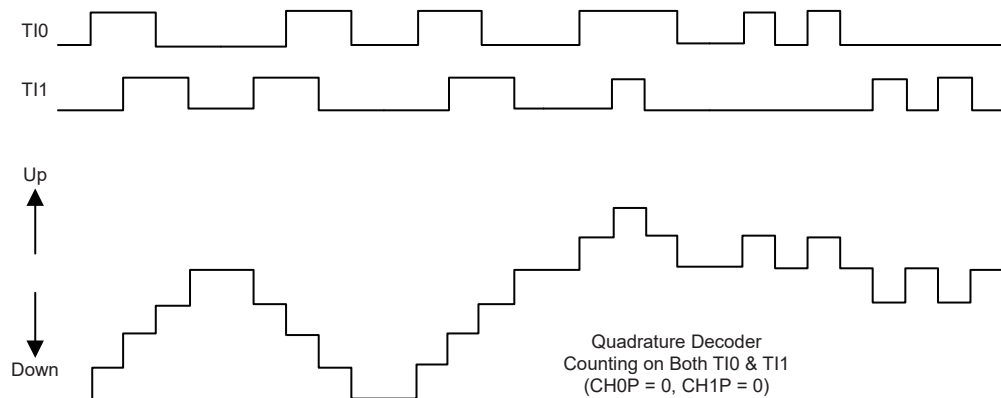


Figure 47. Both TI0 and TI1 Quadrature Decoder Counting

Output Stage

The GPTM has four channels for compare match, single pulse or PWM output function. The channel output GT_CHxO is controlled by the CHxOM, CHxP and CHxE bits in the corresponding CHxOCFR, CHPOLR and CHCTR registers.

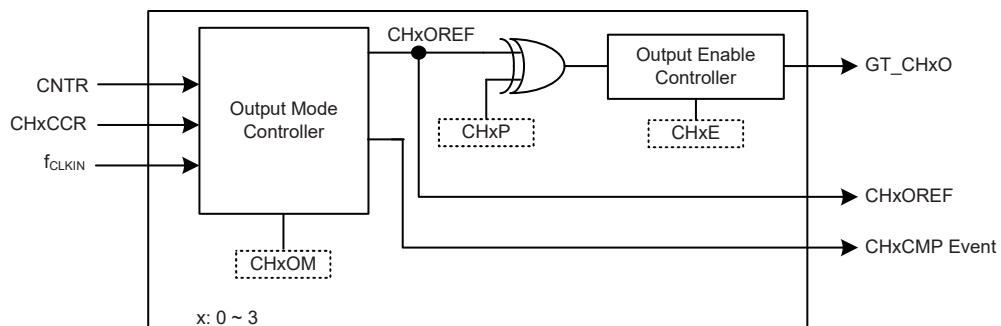


Figure 48. Output Stage Block Diagram

Channel Output Reference Signal

When the GPTM is used in the compare match output mode, the Channel x Output Reference signal, CHxOREF, is defined by the CHxOM field setup. The CHxOREF signal has several types of output function which defines what happens to the output when the counter value matches the contents of the CHxCCR register. In addition to the low, high and toggle CHxOREF output types, there are also PWM mode 1 and PWM mode 2 outputs. In these modes, the CHxOREF signal level is changed according to the count direction and the relationship between the counter value and the CHxCCR content. There are also two modes which will force the output into an inactive or active state irrespective of the CHxCCR content or counter values. With regard to a more detailed description refer to the relative bit definition. The accompanying Table 30 shows a summary of the output type setup.

Table 30. Compare Match Output Setup

CHxOM Value	Compare Match Level
0x0	No change
0x1	Clear Output to 0
0x2	Set Output to 1
0x3	Toggle Output
0x4	Force Inactive Level
0x5	Force Active Level
0x6	PWM Mode 1
0x7	PWM Mode 2

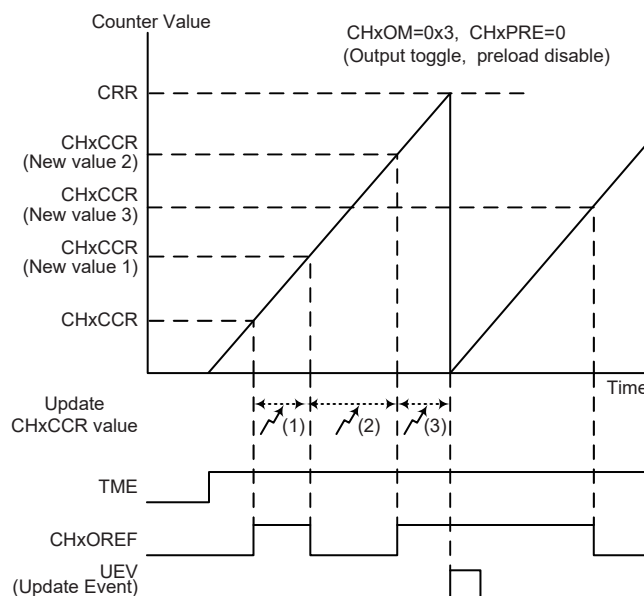


Figure 49. Toggle Mode Channel Output Reference Signal (CHxPRE = 0)

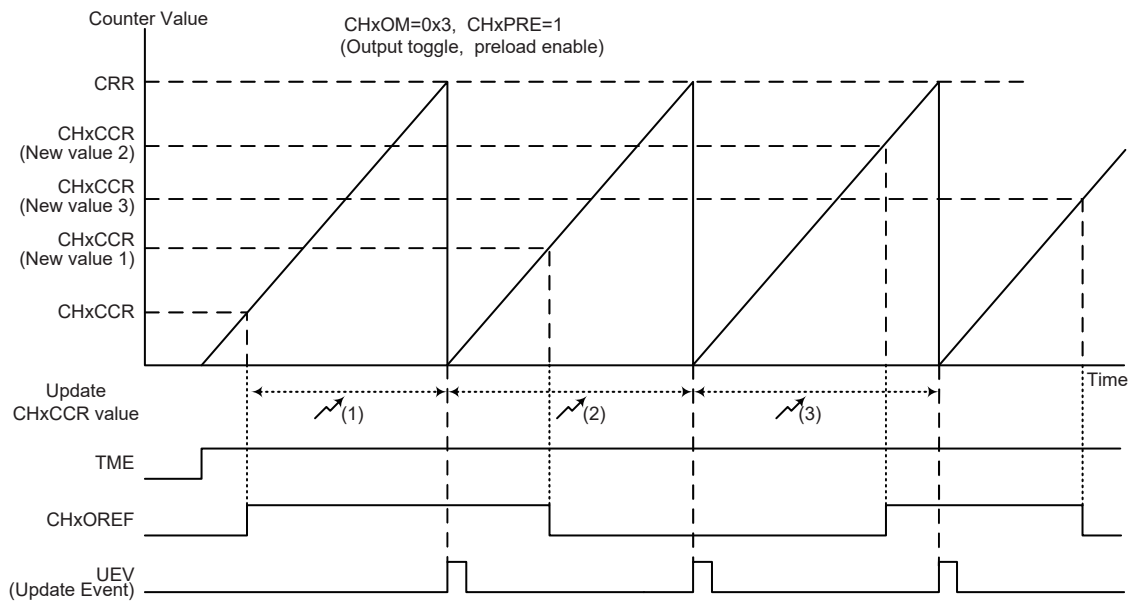


Figure 50. Toggle Mode Channel Output Reference Signal (CHxPRE = 1)

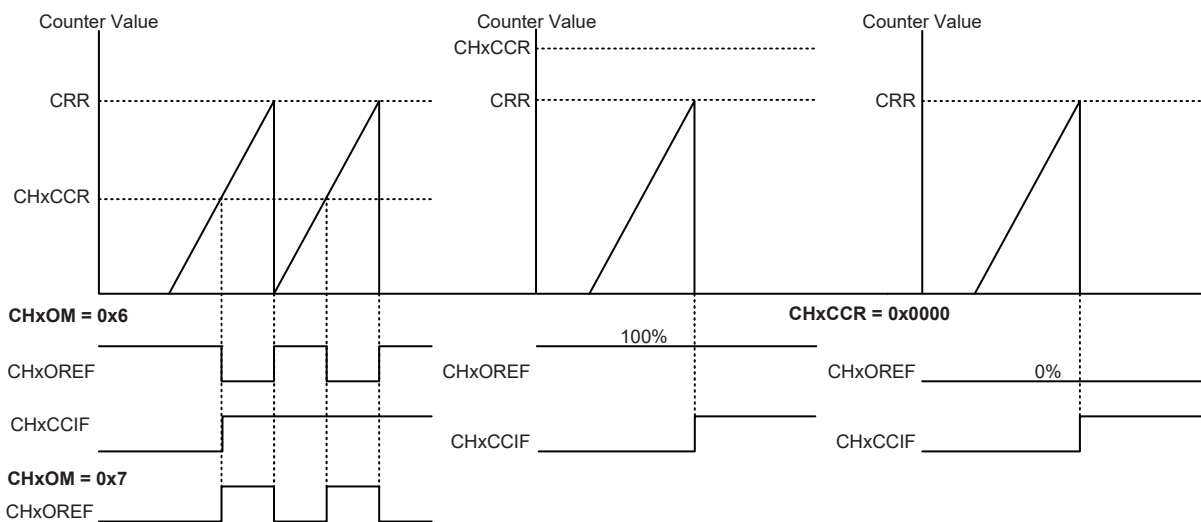


Figure 51. PWM Mode Channel Output Reference Signal and Counter in Up-counting Mode

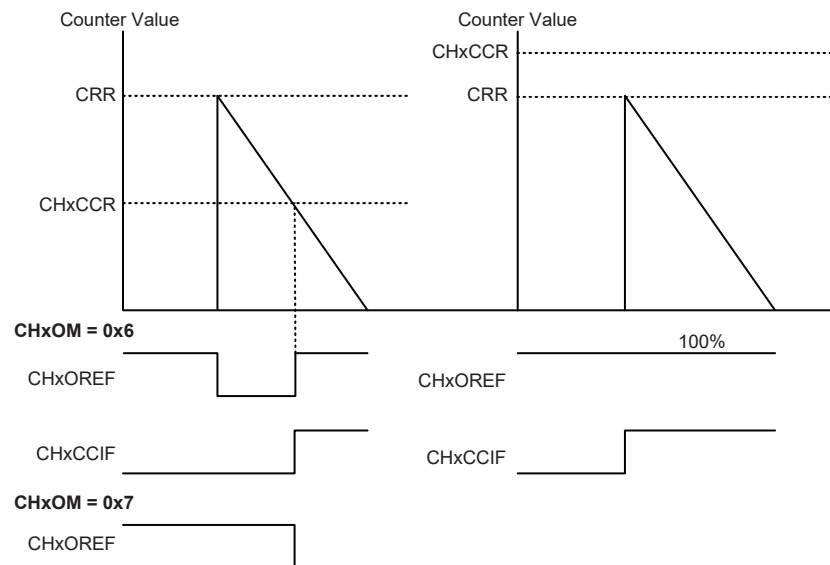


Figure 52. PWM Mode Channel Output Reference Signal and Counter in Down-counting Mode

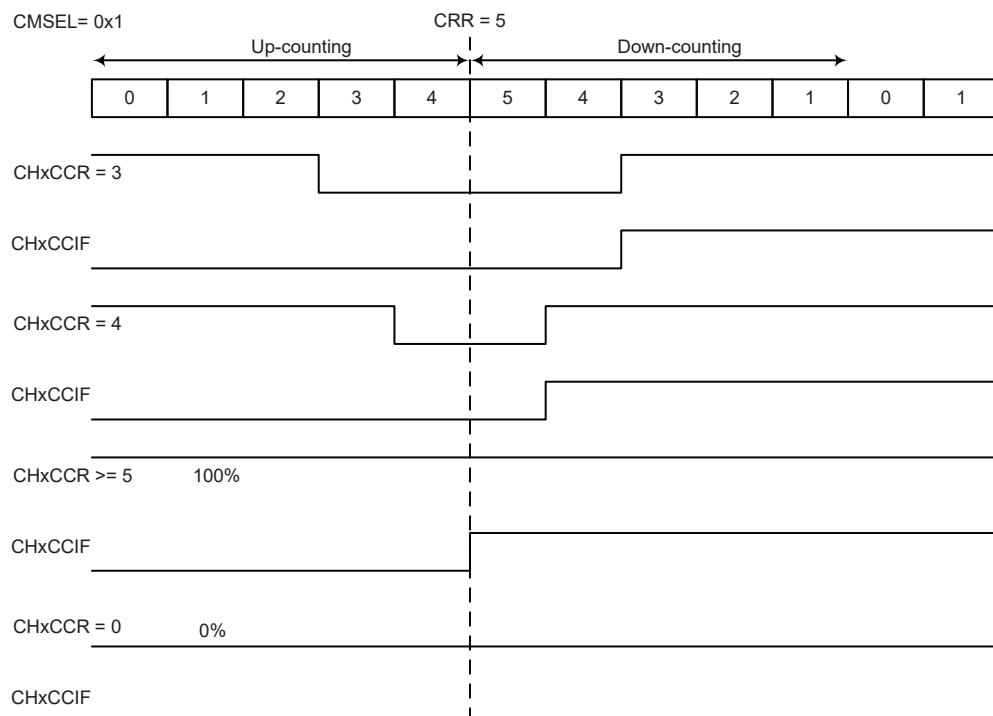


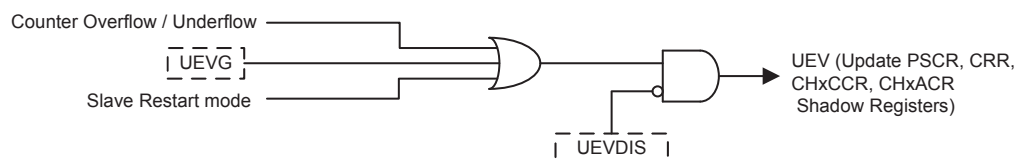
Figure 53. PWM Mode Channel Output Reference Signal and Counter in Centre-aligned Mode

Update Management

The Update event is used to update the CRR, the PSCR, the CHxACR and the CHxCCR values from the actual registers to the corresponding shadow registers. An update event occurs when the counter overflows or underflows, the software update control bit is triggered or an update event from the slave controller is generated.

The UEVDIS bit in the CNTCFR register can determine whether the update event occurs or not. When the update event occurs, the corresponding update event interrupt will be generated depending upon whether the update event interrupt generation function is enabled or not by configuring the UGDIS bit in the CNTCFR register. For more detailed description, refer to the UEVDIS and UGDIS bit definition in the CNTCFR register

Update Event Management



Update Event Interrupt Management

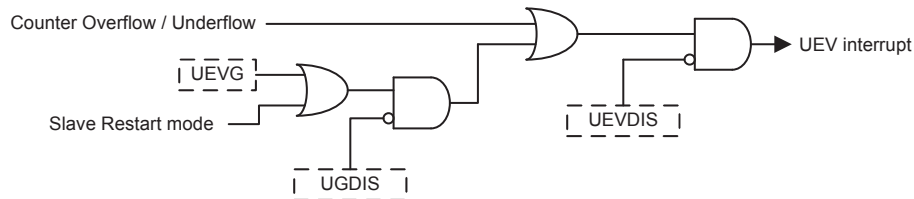


Figure 54. Update Event Setting Diagram

Single Pulse Mode

Once the timer is set to operate in the single pulse mode, it is not necessary to set the timer enable bit TME in the CTR register to 1 to enable the counter. The trigger to generate a pulse can be sourced from the STI signal rising edge or by setting the TME bit to 1 using software. Setting the TME bit to 1 or a trigger from the STI signal rising edge can generate a pulse and then keep the TME bit at a high state until the update event occurs or the TME bit is written to 0 by software. If the TME bit is cleared to 0 using software, the counter will be stopped and its value held. If the TME bit is automatically cleared to 0 by a hardware update event, the counter will be reinitialized.

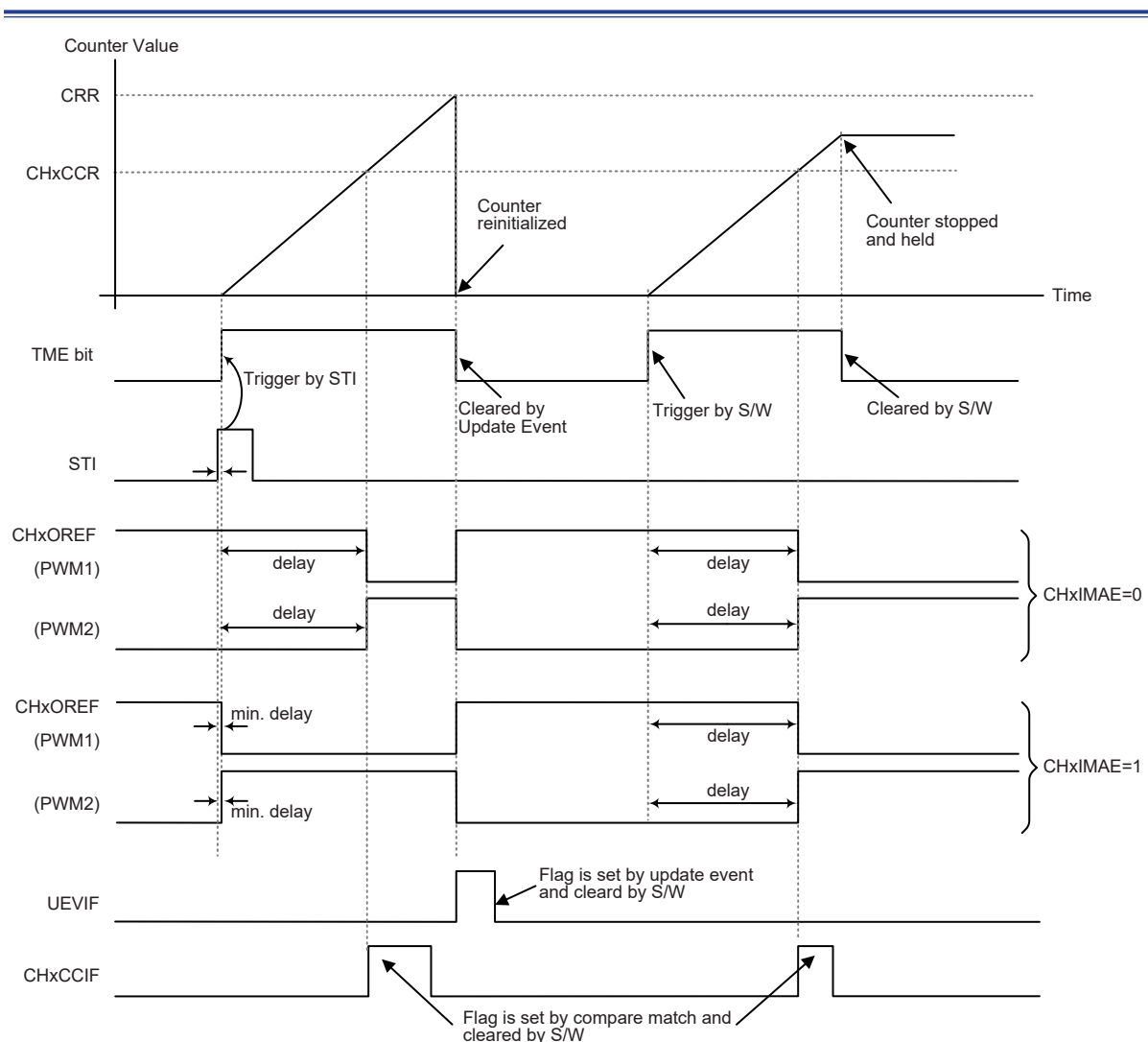


Figure 55. Single Pulse Mode

In the Single Pulse mode, the STI active edge which sets the TME bit to 1 will enable the counter. However, there exist several clock delays to perform the comparison result between the counter value and the CHxCCR value. In order to reduce the delay to a minimum value, the user can set the CHxIMAE bit in each CHxOCFR register. After an STI rising edge trigger occurs in the single pulse mode, the CHxOREF signal will immediately be forced to the state which the CHxOREF signal will change to as the compare match event occurs without taking the comparison result into account. The CHxIMAE bit is available only when the output channel is configured to operate in the PWM mode 1 or PWM mode 2 and the trigger source is derived from the STI signal.

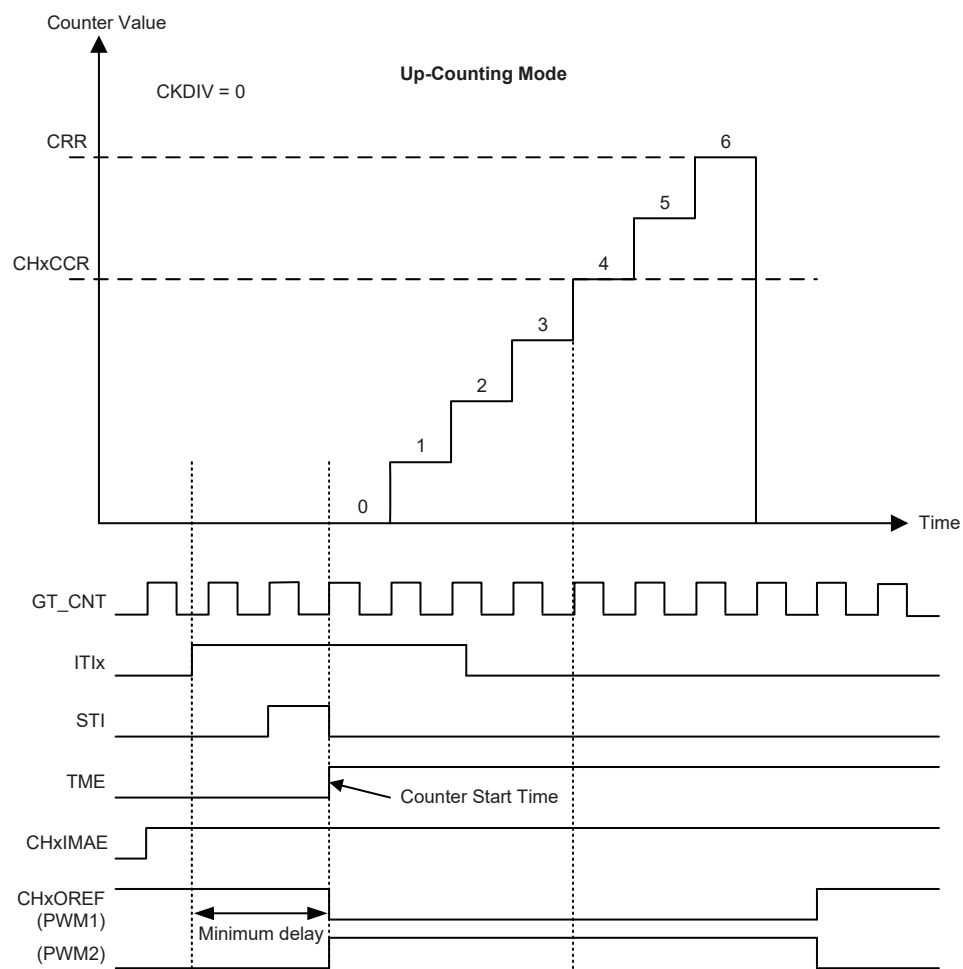


Figure 56. Immediate Active Mode Minimum Delay

Asymmetric PWM Mode

Asymmetric PWM mode allows two center-aligned PWM signals to be generated with a programmable phase shift. While the PWM frequency is determined by the value of the CRR register, the duty cycle and the phase-shift are determined by the CHxCCR and CHxACR register. When the counter is counting up, the PWM uses the value in CHxCCR as up-count compare value. When the counter is into counting down stage, the PWM uses the value in CHxACR as down-count compare value. The Figure 57 is shown as an example for asymmetric PWM mode in center-aligned counting mode.

Note: Asymmetric PWM mode can only be operated in center-aligned counting mode.

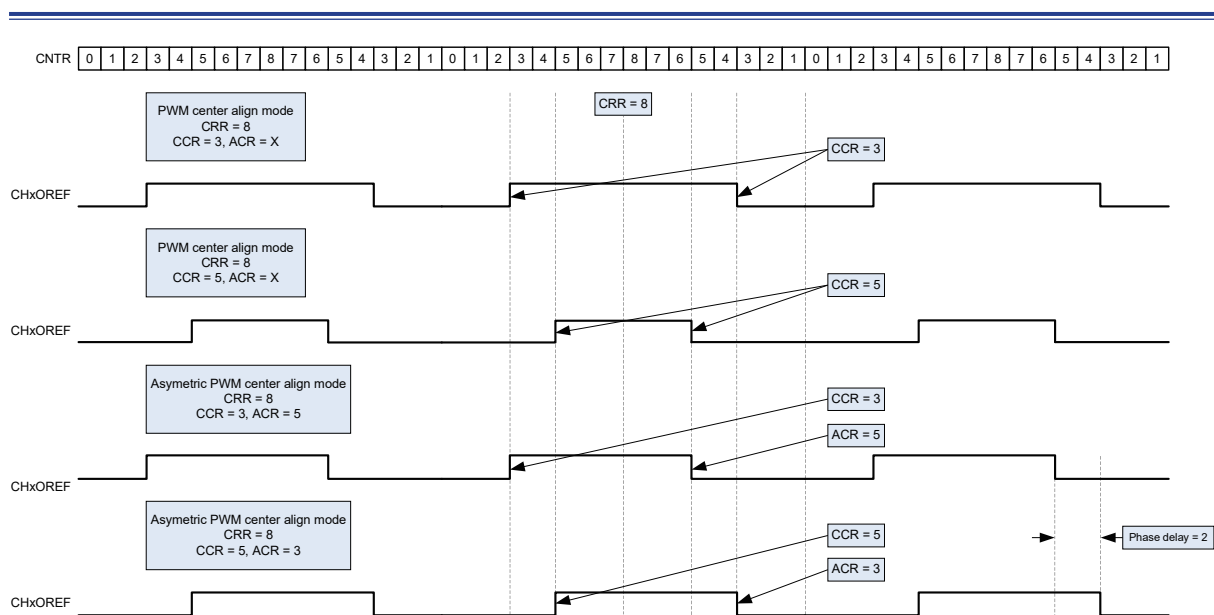


Figure 57. Asymmetric PWM Mode versus Center-Aligned Counting Mode

Timer Interconnection

The timers can be internally connected together for timer chaining or synchronization. This can be implemented by configuring one timer to operate in the Master mode while configuring another timer to be in the Slave mode. The following figures present several examples of trigger selection for the master and slave modes.

Using One Timer to Enable/Disable another Timer Start or Stop Counting

- Configure GPTM as the master mode to send its channel 0 Output Reference signal CH0OREF as a trigger output (MMSEL = 0x4).
- Configure GPTM CH0OREF waveform.
- Configure PWM0 to receive its input trigger source from the GPTM trigger output (TRSEL = 0xA).
- Configure PWM0 to operate in the pause mode (SMSEL = 0x5).
- Enable PWM0 by writing '1' to the TME bit.
- Enable GPTM by writing '1' to the TME bit.

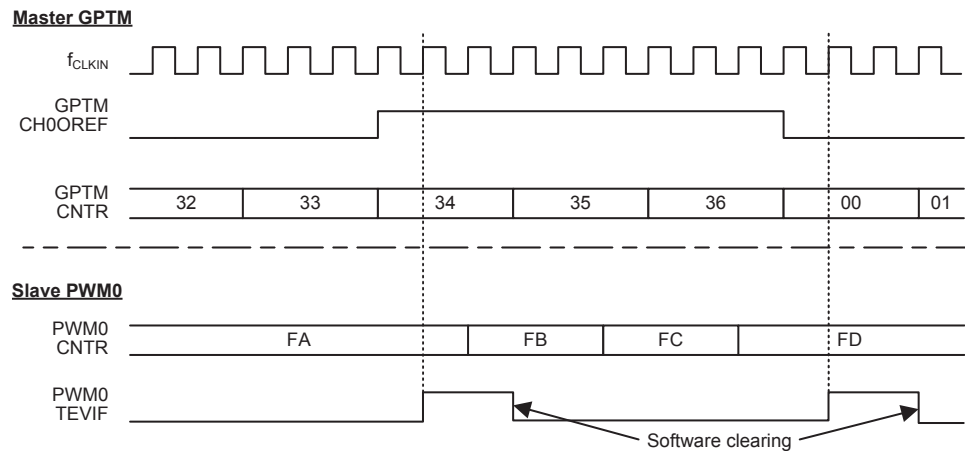


Figure 58. Pausing PWM0 using the GPTM CH0OREF Signal

Using one Timer to Trigger another Timer Start Counting

- Configure GPTM to operate in the master mode to send its Update Event UEV as the trigger output (MMSEL = 0x2).
- Configure the GPTM period by setting the CRR register.
- Configure PWM0 to get the input trigger source from the GPTM trigger output (TRSEL = 0xA).
- Configure PWM0 to be in the slave trigger mode (SMSEL = 0x6).
- Start GPTM by writing '1' to the TME bit.

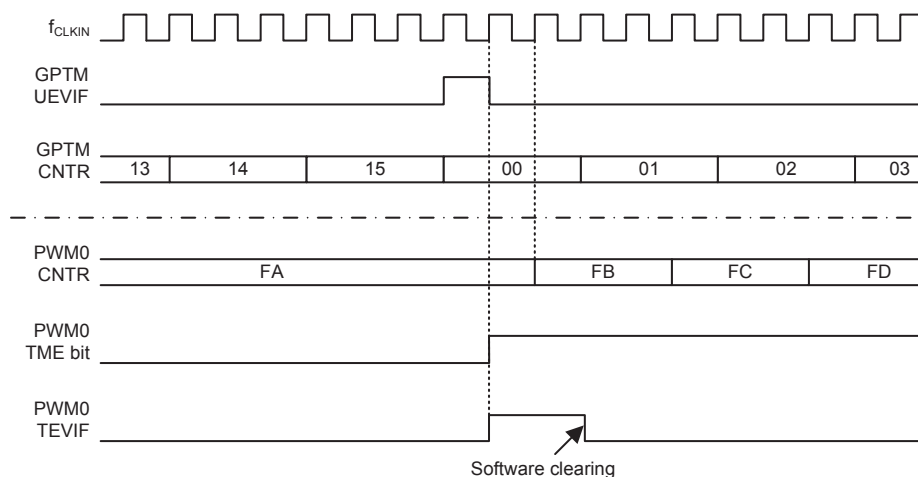
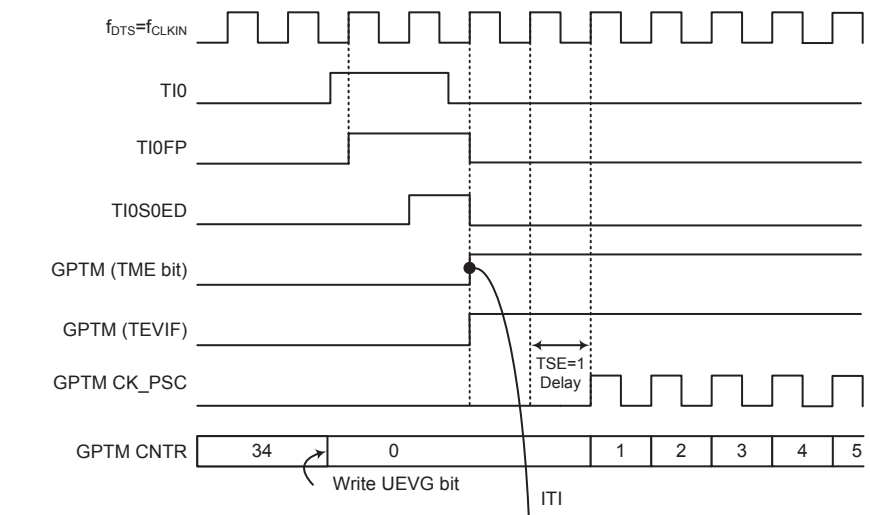


Figure 59. Triggering PWM0 with GPTM Update Event

Starting Two Timers Synchronously in Response to an External Trigger

- Configure GPTM to operate in the master mode to send its enable signal as a trigger output (MMSEL = 0x1).
- Configure GPTM slave mode to receive its input trigger source from GT_CH0 pin (TRSEL = 0x1).
- Configure GPTM to be in the slave trigger mode (SMSEL = 0x6).
- Enable the GPTM master timer synchronization function by setting the TSE bit in the MDCFR register to 1 to synchronize the slave timer.
- Configure PWM0 to receive its input trigger source from the GPTM trigger output (TRSEL = 0xA).
- Configure PWM0 to be in the slave trigger mode (SMSEL = 0x6).

Master GPTM



Slave PWM0

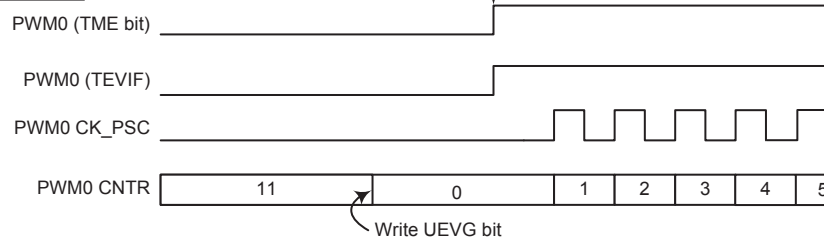


Figure 60. Trigger GPTM and PWM0 with the GPTM CH0 Input

PDMA Request

The GPTM supports the interface for PDMA data transfer. There are certain events which can generate the PDMA requests if the corresponding enable control bits are set to 1 to enable the PDMA access. These events are the GPTM update events, trigger events and channel capture/compare events. When the PDMA request is generated from the GPTM channel, it can be derived from the channel capture/compare event or the GPTM update event selected by the channel PDMA selection bit, CHCCDS, for all channels. For more detailed PDMA configuring information, refer to the corresponding section in the PDMA chapter.

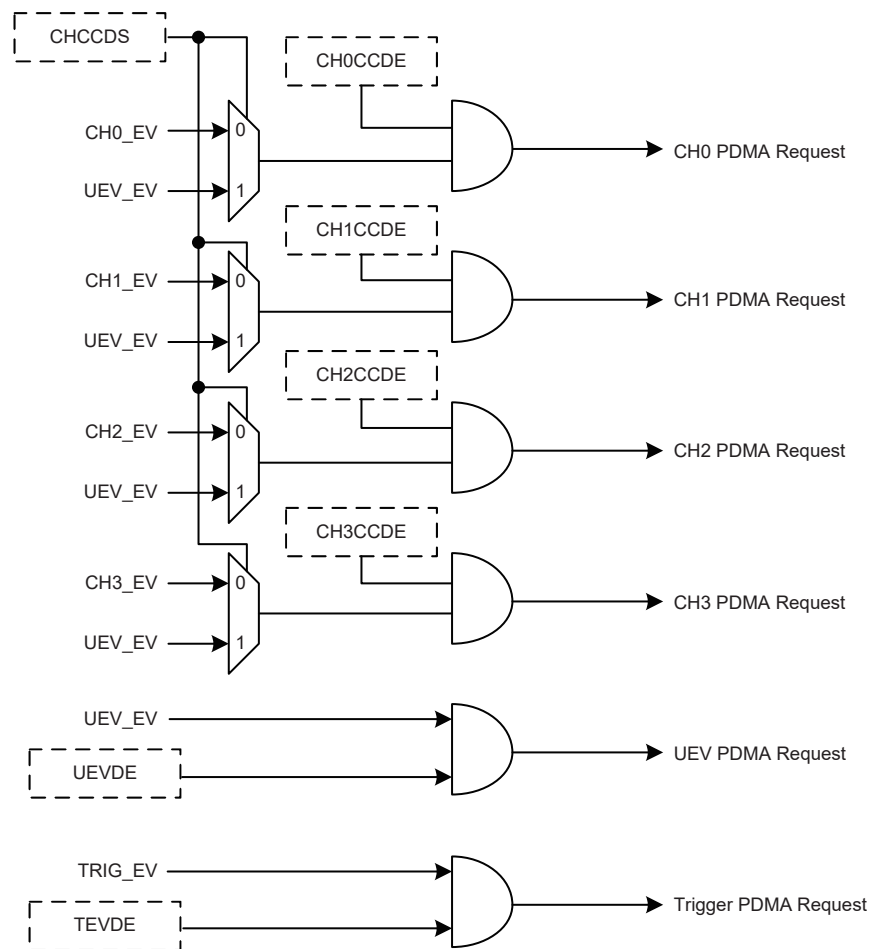


Figure 61. GPTM PDMA Mapping Diagram

Register Map

The following table shows the GPTM registers and reset values.

Table 31. GPTM Register Map

Register	Offset	Description	Reset Value
CNTCFR	0x000	Timer Counter Configuration Register	0x0000_0000
MDCFR	0x004	Timer Mode Configuration Register	0x0000_0000
TRCFR	0x008	Timer Trigger Configuration Register	0x0000_0000
CTR	0x010	Timer Control Register	0x0000_0000
CH0ICFR	0x020	Channel 0 Input Configuration Register	0x0000_0000
CH1ICFR	0x024	Channel 1 Input Configuration Register	0x0000_0000
CH2ICFR	0x028	Channel 2 Input Configuration Register	0x0000_0000
CH3ICFR	0x02C	Channel 3 Input Configuration Register	0x0000_0000
CH0OCFR	0x040	Channel 0 Output Configuration Register	0x0000_0000
CH1OCFR	0x044	Channel 1 Output Configuration Register	0x0000_0000
CH2OCFR	0x048	Channel 2 Output Configuration Register	0x0000_0000
CH3OCFR	0x04C	Channel 3 Output Configuration Register	0x0000_0000
CHCTR	0x050	Channel Control Register	0x0000_0000
CHPOLR	0x054	Channel Polarity Configuration Register	0x0000_0000
DICTR	0x074	Timer PDMA/Interrupt Control Register	0x0000_0000
EVGR	0x078	Timer Event Generator Register	0x0000_0000
INTSR	0x07C	Timer Interrupt Status Register	0x0000_0000
CNTR	0x080	Timer Counter Register	0x0000_0000
PSCR	0x084	Timer Prescaler Register	0x0000_0000
CRR	0x088	Timer Counter-Reload Register	0x0000_FFFF
CH0CCR	0x090	Channel 0 Capture/Compare Register	0x0000_0000
CH1CCR	0x094	Channel 1 Capture/Compare Register	0x0000_0000
CH2CCR	0x098	Channel 2 Capture/Compare Register	0x0000_0000
CH3CCR	0x09C	Channel 3 Capture/Compare Register	0x0000_0000
CH0ACR	0x0A0	Channel 0 Asymmetric Compare Register	0x0000_0000
CH1ACR	0x0A4	Channel 1 Asymmetric Compare Register	0x0000_0000
CH2ACR	0x0A8	Channel 2 Asymmetric Compare Register	0x0000_0000
CH3ACR	0x0AC	Channel 3 Asymmetric Compare Register	0x0000_0000

Register Descriptions

Timer Counter Configuration Register – CNTCFR

This register specifies the GPTM counter configuration.

Offset: 0x000

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							DIR
Type/Reset								RW 0
	23	22	21	20	19	18	17	16
	Reserved						CMSEL	
Type/Reset							RW 0	RW 0
	15	14	13	12	11	10	9	8
	Reserved						CKDIV	
Type/Reset							RW 0	RW 0
	7	6	5	4	3	2	1	0
	Reserved						UGDIS	UEVDIS
Type/Reset							RW 0	RW 0

Bits	Field	Descriptions
[24]	DIR	Counting Direction 0: Count-up 1: Count-down Note: This bit is read only when the Timer is configured to be in the Center-aligned mode or when used as a Quadrature decoder.
[17:16]	CMSEL	Counter Mode Selection 00: Edge-aligned mode. Normal up-counting and down-counting available for this mode. Counting direction is defined by the DIR bit. 01: Center-aligned mode 1. The counter counts up and down alternatively. The compare match interrupt flag is set during the count-down period. 10: Center-aligned mode 2. The counter counts up and down alternatively. The compare match interrupt flag is set during the count-up period. 11: Center-aligned mode 3. The counter counts up and down alternatively. The compare match interrupt flag is set during the count-up and count-down periods.
[9:8]	CKDIV	Clock Division These two bits define the frequency ratio between the timer clock (f_{CLKIN}) and the dead-time clock (f_{DTS}). The dead-time clock is also used for digital filter sampling clock. 00: $f_{DTS} = f_{CLKIN}$ 01: $f_{DTS} = f_{CLKIN} / 2$ 10: $f_{DTS} = f_{CLKIN} / 4$ 11: Reserved

Bits	Field	Descriptions
[1]	UGDIS	Update event interrupt generation disable control 0: Any of the following events will generate an update PDMA request or interrupt - Counter overflow/underflow - Setting the UEVG bit - Update generation through the slave mode 1: Only counter overflow/underflow generates an update PDMA request or interrupt
[0]	UEVDIS	Update Event Disable control 0: Enable the update event request by one of following events: - Counter overflow/underflow - Setting the UEVG bit - Update generation through the slave mode 1: Disable the update event (However the counter and the prescaler are reinitialized if the UEVG bit is set or if a hardware restart is received from the slave mode)

Timer Mode Configuration Register – MDCFR

This register specifies the GPTM master and slave mode selection and single pulse mode.

Offset: 0x004

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
	Reserved							SPMSET	
Type/Reset								RW	0
	23	22	21	20	19	18	17	16	
	Reserved					MMSEL			
Type/Reset						RW	0	RW	0
	15	14	13	12	11	10	9	8	
	Reserved					SMSEL			
Type/Reset						RW	0	RW	0
	7	6	5	4	3	2	1	0	
	Reserved							TSE	
Type/Reset								RW	0

Bits	Field	Descriptions
[24]	SPMSET	Single Pulse Mode Setting 0: Counter counts normally irrespective of whether the update event occurred or not 1: Counter stops counting at the next update event and then the TME bit is cleared by hardware

Bits	Field	Descriptions																											
[18:16]	MMSEL	<p>Master Mode Selection</p> <p>Master mode selection is used to select the MTO signal source which is used to synchronize the other slave timer.</p> <table> <tr> <th>MMSEL [2:0]</th><th>Mode</th><th>Descriptions</th></tr> <tr> <td>000</td><td>Reset Mode</td><td>The MTO signal in the Reset mode is an output derived from one of the following cases: Software setting UEVG bit The STI trigger input signal which will be output on the MTO signal line when the Timer is used in the slave Restart mode</td></tr> <tr> <td>001</td><td>Enable Mode</td><td>The Counter Enable signal is used as the trigger output.</td></tr> <tr> <td>010</td><td>Update Mode</td><td>The update event is used as the trigger output according to one of the following cases when the UEVDIS bit is cleared to 0: Counter overflow / underflow Software setting UEVG Slave trigger input when used in slave restart mode</td></tr> <tr> <td>011</td><td>Capture/Compare Mode</td><td>When a Channel 0 capture or compare match event occurs, it will generate a positive pulse used as the master trigger output.</td></tr> <tr> <td>100</td><td>Compare Mode 0</td><td>The Channel 0 Output reference signal named CH0OREF is used as the trigger output.</td></tr> <tr> <td>101</td><td>Compare Mode 1</td><td>The Channel 1 Output reference signal named CH1OREF is used as the trigger output.</td></tr> <tr> <td>110</td><td>Compare Mode 2</td><td>The Channel 2 Output reference signal named CH2OREF is used as the trigger output.</td></tr> <tr> <td>111</td><td>Compare Mode 3</td><td>The Channel 3 Output reference signal named CH3OREF is used as the trigger output.</td></tr> </table>	MMSEL [2:0]	Mode	Descriptions	000	Reset Mode	The MTO signal in the Reset mode is an output derived from one of the following cases: Software setting UEVG bit The STI trigger input signal which will be output on the MTO signal line when the Timer is used in the slave Restart mode	001	Enable Mode	The Counter Enable signal is used as the trigger output.	010	Update Mode	The update event is used as the trigger output according to one of the following cases when the UEVDIS bit is cleared to 0: Counter overflow / underflow Software setting UEVG Slave trigger input when used in slave restart mode	011	Capture/Compare Mode	When a Channel 0 capture or compare match event occurs, it will generate a positive pulse used as the master trigger output.	100	Compare Mode 0	The Channel 0 Output reference signal named CH0OREF is used as the trigger output.	101	Compare Mode 1	The Channel 1 Output reference signal named CH1OREF is used as the trigger output.	110	Compare Mode 2	The Channel 2 Output reference signal named CH2OREF is used as the trigger output.	111	Compare Mode 3	The Channel 3 Output reference signal named CH3OREF is used as the trigger output.
MMSEL [2:0]	Mode	Descriptions																											
000	Reset Mode	The MTO signal in the Reset mode is an output derived from one of the following cases: Software setting UEVG bit The STI trigger input signal which will be output on the MTO signal line when the Timer is used in the slave Restart mode																											
001	Enable Mode	The Counter Enable signal is used as the trigger output.																											
010	Update Mode	The update event is used as the trigger output according to one of the following cases when the UEVDIS bit is cleared to 0: Counter overflow / underflow Software setting UEVG Slave trigger input when used in slave restart mode																											
011	Capture/Compare Mode	When a Channel 0 capture or compare match event occurs, it will generate a positive pulse used as the master trigger output.																											
100	Compare Mode 0	The Channel 0 Output reference signal named CH0OREF is used as the trigger output.																											
101	Compare Mode 1	The Channel 1 Output reference signal named CH1OREF is used as the trigger output.																											
110	Compare Mode 2	The Channel 2 Output reference signal named CH2OREF is used as the trigger output.																											
111	Compare Mode 3	The Channel 3 Output reference signal named CH3OREF is used as the trigger output.																											

Bits	Field	Descriptions																											
[10:8]	SMSEL	Slave Mode Selection																											
		<table><tr><th>SMSEL [2:0]</th><th>Mode</th><th>Descriptions</th></tr><tr><td>000</td><td>Disable Mode</td><td>The prescaler is clocked directly by the internal clock.</td></tr><tr><td>001</td><td>Quadrature Decoder Mode 1</td><td>The counter uses the clock pulse generated from the interaction between the TI0 and TI1 signals to drive the counter prescaler. A transition of the TI0 edge is used in this mode depending upon the TI1 level.</td></tr><tr><td>010</td><td>Quadrature Decoder Mode 2</td><td>The counter uses the clock pulse generated from the interaction between the TI0 and TI1 signals to drive the counter prescaler. A transition of the TI1 edge is used in this mode depending upon the TI0 level.</td></tr><tr><td>011</td><td>Quadrature Decoder Mode 3</td><td>The counter uses the clock pulse generated from the interaction between the TI0 and TI1 signals to drive the counter prescaler. A transition of one channel edge is used in the quadrature decoder mode 3 depending upon the other channel level.</td></tr><tr><td>100</td><td>Restart Mode</td><td>The counter value restarts from 0 or the CRR shadow register value depending upon the counter mode on the rising edge of the STI signal. The registers will also be updated.</td></tr><tr><td>101</td><td>Pause Mode</td><td>The counter starts to count when the selected trigger input STI is high. The counter stops counting on the instant, not being reset, when the STI signal changes its state to a low level. Both the counter start and stop control are determined by the STI signal.</td></tr><tr><td>110</td><td>Trigger Mode</td><td>The counter starts to count from the original value in the counter on the rising edge of the selected trigger input STI. Only the counter start control is determined by the STI signal.</td></tr><tr><td>111</td><td>STIED</td><td>The rising edge of the selected trigger signal STI will clock the counter.</td></tr></table>	SMSEL [2:0]	Mode	Descriptions	000	Disable Mode	The prescaler is clocked directly by the internal clock.	001	Quadrature Decoder Mode 1	The counter uses the clock pulse generated from the interaction between the TI0 and TI1 signals to drive the counter prescaler. A transition of the TI0 edge is used in this mode depending upon the TI1 level.	010	Quadrature Decoder Mode 2	The counter uses the clock pulse generated from the interaction between the TI0 and TI1 signals to drive the counter prescaler. A transition of the TI1 edge is used in this mode depending upon the TI0 level.	011	Quadrature Decoder Mode 3	The counter uses the clock pulse generated from the interaction between the TI0 and TI1 signals to drive the counter prescaler. A transition of one channel edge is used in the quadrature decoder mode 3 depending upon the other channel level.	100	Restart Mode	The counter value restarts from 0 or the CRR shadow register value depending upon the counter mode on the rising edge of the STI signal. The registers will also be updated.	101	Pause Mode	The counter starts to count when the selected trigger input STI is high. The counter stops counting on the instant, not being reset, when the STI signal changes its state to a low level. Both the counter start and stop control are determined by the STI signal.	110	Trigger Mode	The counter starts to count from the original value in the counter on the rising edge of the selected trigger input STI. Only the counter start control is determined by the STI signal.	111	STIED	The rising edge of the selected trigger signal STI will clock the counter.
		SMSEL [2:0]	Mode	Descriptions																									
		000	Disable Mode	The prescaler is clocked directly by the internal clock.																									
		001	Quadrature Decoder Mode 1	The counter uses the clock pulse generated from the interaction between the TI0 and TI1 signals to drive the counter prescaler. A transition of the TI0 edge is used in this mode depending upon the TI1 level.																									
		010	Quadrature Decoder Mode 2	The counter uses the clock pulse generated from the interaction between the TI0 and TI1 signals to drive the counter prescaler. A transition of the TI1 edge is used in this mode depending upon the TI0 level.																									
		011	Quadrature Decoder Mode 3	The counter uses the clock pulse generated from the interaction between the TI0 and TI1 signals to drive the counter prescaler. A transition of one channel edge is used in the quadrature decoder mode 3 depending upon the other channel level.																									
		100	Restart Mode	The counter value restarts from 0 or the CRR shadow register value depending upon the counter mode on the rising edge of the STI signal. The registers will also be updated.																									
		101	Pause Mode	The counter starts to count when the selected trigger input STI is high. The counter stops counting on the instant, not being reset, when the STI signal changes its state to a low level. Both the counter start and stop control are determined by the STI signal.																									
		110	Trigger Mode	The counter starts to count from the original value in the counter on the rising edge of the selected trigger input STI. Only the counter start control is determined by the STI signal.																									
111	STIED	The rising edge of the selected trigger signal STI will clock the counter.																											
[0]	TSE	Timer Synchronization Enable 0: No action 1: Master timer (current timer) will generate a delay to synchronize its slave timer through the MTO signal.																											

Timer Trigger Configuration Register – TRCFR

This register specifies the trigger source selection of GPTM.

Offset: 0x008

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved				TRSEL			
					RW	0	RW	0

Bits	Field	Descriptions
[3:0]	TRSEL	<p>Trigger Source Selection</p> <p>These bits are used to select the trigger input (STI) for counter synchronization.</p> <p>0000: Software Trigger by setting the UEVG bit</p> <p>0001: Filtered input of channel 0 (TI0S0)</p> <p>0010: Filtered input of channel 1 (TI1S1)</p> <p>0011: Reserved</p> <p>1000: Channel 0 Edge Detector (TI0BED)</p> <p>1001: Internal Timing Module Trigger 0 (ITI0)</p> <p>1010: Internal Timing Module Trigger 1 (ITI1)</p> <p>1011: Internal Timing Module Trigger 2 (ITI2)</p> <p>Others: Reserved</p> <p>Note: These bits must be updated only when they are not in use, i.e. the slave mode is disabled by setting the SMSEL field to 0x0.</p>

Table 32. GPTM Internal Trigger Connection

Slave Timing Module	ITI0	ITI1	ITI2
GPTM	PWM0	PWM1	—

Timer Counter Register – CTR

This register specifies the timer enable bit (TME), CRR buffer enable bit (CRBE) and Channel PDMA selection bit (CHCCDS).

Offset: 0x010

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved							CHCCDS	
	15	14	13	12	11	10	9	8	0
Type/Reset	Reserved								RW
	7	6	5	4	3	2	1	0	
Type/Reset	Reserved						CRBE	TME	
							RW	0	RW 0

Bits	Field	Descriptions
[16]	CHCCDS	Channel PDMA event selection 0: Channel PDMA request derived from the channel capture/compare event. 1: Channel PDMA request derived from the Update event.
[1]	CRBE	Counter-Reload register Buffer Enable 0: Counter-reload register can be updated immediately 1: Counter-reload register can not be updated until the update event occurs
[0]	TME	Timer Enable bit 0: GPTM off 1: GPTM on – GPTM functions normally When the TME bit is cleared to 0, the counter is stopped and the GPTM consumes no power in any operation mode except for the single pulse mode and the slave trigger mode. In these two modes the TME bit can automatically be set to 1 by hardware which permits all the GPTM registers to function normally.

Channel 0 Input Configuration Register – CH0ICFR

This register specifies the channel 0 input mode configuration.

Offset: 0x020

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	TIOSRC	Reserved						
Type/Reset	RW	0						
	23	22	21	20	19	18	17	16
	Reserved				CH0PSC		CH0CCS	
Type/Reset					RW	0	RW	0
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved				TIOF			
Type/Reset					RW	0	RW	0

Bits	Field	Descriptions
[31]	TIOSRC	Channel 0 Input Source TIO Selection 0: The GT_CH0 pin is connected to channel 0 input TIO 1: The XOR operation output of the GT_CH0, GT_CH1, and GT_CH2 pins are connected to the channel 0 input TIO
[19:18]	CH0PSC	Channel 0 Capture Input Source Prescaler Setting These bits define the effective events of the channel 0 capture input. Note that the prescaler is reset once the Channel 0 Capture/Compare Enable bit, CH0E, in the Channel Control register named CHCTR is cleared to 0. 00: No prescaler, channel 0 capture input signal is chosen for each active event 01: Channel 0 Capture input signal is chosen for every 2 events 10: Channel 0 Capture input signal is chosen for every 4 events 11: Channel 0 Capture input signal is chosen for every 8 events
[17:16]	CH0CCS	Channel 0 Capture/Compare Selection 00: Channel 0 is configured as an output 01: Channel 0 is configured as an input derived from the TIO signal 10: Channel 0 is configured as an input derived from the TI1 signal 11: Channel 0 is configured as an input which comes from the TRCED signal derived from the Trigger Controller Note: The CH0CCS field can be accessed only when the CH0E bit is cleared to 0.

Bits	Field	Descriptions
[3:0]	TI0F	<p>Channel 0 Input Source TI0 Filter Setting</p> <p>These bits define the frequency divided ratio used to sample the TI0 signal. The Digital filter in the GPTM is an N-event counter where N is defined as how many valid transitions are necessary to output a filtered signal.</p> <p>0000: No filter, the sampling clock is f_{SYSTEM}</p> <p>0001: $f_{sampling} = f_{CLKIN}$, $N = 2$</p> <p>0010: $f_{sampling} = f_{CLKIN}$, $N = 4$</p> <p>0011: $f_{sampling} = f_{CLKIN}$, $N = 8$</p> <p>0100: $f_{sampling} = f_{DTS} / 2$, $N = 6$</p> <p>0101: $f_{sampling} = f_{DTS} / 2$, $N = 8$</p> <p>0110: $f_{sampling} = f_{DTS} / 4$, $N = 6$</p> <p>0111: $f_{sampling} = f_{DTS} / 4$, $N = 8$</p> <p>1000: $f_{sampling} = f_{DTS} / 8$, $N = 6$</p> <p>1001: $f_{sampling} = f_{DTS} / 8$, $N = 8$</p> <p>1010: $f_{sampling} = f_{DTS} / 16$, $N = 5$</p> <p>1011: $f_{sampling} = f_{DTS} / 16$, $N = 6$</p> <p>1100: $f_{sampling} = f_{DTS} / 16$, $N = 8$</p> <p>1101: $f_{sampling} = f_{DTS} / 32$, $N = 5$</p> <p>1110: $f_{sampling} = f_{DTS} / 32$, $N = 6$</p> <p>1111: $f_{sampling} = f_{DTS} / 32$, $N = 8$</p>

Channel 1 Input Configuration Register – CH1ICFR

This register specifies the channel 1 input mode configuration.

Offset: 0x024

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved				CH1PSC		CH1CCS	
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved				TI1F			
					RW	0	RW	0
					RW	0	RW	0

Bits	Field	Descriptions
[19:18]	CH1PSC	<p>Channel 1 Capture Input Source Prescaler Setting</p> <p>These bits define the effective events of the channel 1 capture input. Note that the prescaler is reset once the Channel 1 Capture/Compare Enable bit, CH1E, in the Channel Control register named CHCTR is cleared to 0.</p> <p>00: No prescaler, channel 1 capture input signal is chosen for each active event</p> <p>01: Channel 1 Capture input signal is chosen for every 2 events</p> <p>10: Channel 1 Capture input signal is chosen for every 4 events</p> <p>11: Channel 1 Capture input signal is chosen for every 8 events</p>

Bits	Field	Descriptions
[17:16]	CH1CCS	<p>Channel 1 Capture/Compare Selection</p> <p>00: Channel 1 is configured as an output 01: Channel 1 is configured as an input derived from the TI1 signal 10: Channel 1 is configured as an input derived from the TI0 signal 11: Channel 1 is configured as an input which comes from the TRCED signal derived from the Trigger Controller</p> <p>Note: The CH1CCS field can be accessed only when the CH1E bit is cleared to 0.</p>
[3:0]	TI1F	<p>Channel 1 Input Source TI1 Filter Setting</p> <p>These bits define the frequency divided ratio used to sample the TI1 signal. The Digital filter in the GPTM is an N-event counter where N is defined as how many valid transitions are necessary to output a filtered signal.</p> <p>0000: No filter, the sampling clock is f_{SYSTEM} 0001: $f_{\text{sampling}} = f_{\text{CLKIN}}$, $N = 2$ 0010: $f_{\text{sampling}} = f_{\text{CLKIN}}$, $N = 4$ 0011: $f_{\text{sampling}} = f_{\text{CLKIN}}$, $N = 8$ 0100: $f_{\text{sampling}} = f_{\text{DTS}} / 2$, $N = 6$ 0101: $f_{\text{sampling}} = f_{\text{DTS}} / 2$, $N = 8$ 0110: $f_{\text{sampling}} = f_{\text{DTS}} / 4$, $N = 6$ 0111: $f_{\text{sampling}} = f_{\text{DTS}} / 4$, $N = 8$ 1000: $f_{\text{sampling}} = f_{\text{DTS}} / 8$, $N = 6$ 1001: $f_{\text{sampling}} = f_{\text{DTS}} / 8$, $N = 8$ 1010: $f_{\text{sampling}} = f_{\text{DTS}} / 16$, $N = 5$ 1011: $f_{\text{sampling}} = f_{\text{DTS}} / 16$, $N = 6$ 1100: $f_{\text{sampling}} = f_{\text{DTS}} / 16$, $N = 8$ 1101: $f_{\text{sampling}} = f_{\text{DTS}} / 32$, $N = 5$ 1110: $f_{\text{sampling}} = f_{\text{DTS}} / 32$, $N = 6$ 1111: $f_{\text{sampling}} = f_{\text{DTS}} / 32$, $N = 8$</p>

Channel 2 Input Configuration Register – CH2ICFR

This register specifies the channel 2 input mode configuration.

Offset: 0x028

Reset value: 0x0000_0000

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

Bits	Field	Descriptions
[19:18]	CH2PSC	Channel 2 Capture Input Source Prescaler Setting These bits define the effective events of the channel 2 capture input. Note that the prescaler is reset once the Channel 2 Capture/Compare Enable bit, CH2E, in the Channel Control register named CHCTR is cleared to 0. 00: No prescaler, channel 2 capture input signal is chosen for each active event 01: Channel 2 Capture input signal is chosen for every 2 events 10: Channel 2 Capture input signal is chosen for every 4 events 11: Channel 2 Capture input signal is chosen for every 8 events
[17:16]	CH2CCS	Channel 2 Capture/Compare Selection 00: Channel 2 is configured as an output 01: Channel 2 is configured as an input derived from the TI2 signal 10: Channel 2 is configured as an input derived from the TI3 signal 11: Channel 2 is configured as an input which comes from the TRCED signal derived from the Trigger Controller Note: The CH2CCS field can be accessed only when the CH2E bit is cleared to 0.

Bits	Field	Descriptions
[3:0]	TI2F	<p>Channel 2 Input Source TI2 Filter Setting</p> <p>These bits define the frequency divided ratio used to sample the TI2 signal. The Digital filter in the GPTM is an N-event counter where N is defined as how many valid transitions are necessary to output a filtered signal.</p> <p>0000: No filter, the sampling clock is f_{SYSTEM}</p> <p>0001: $f_{sampling} = f_{CLKIN}$, $N = 2$</p> <p>0010: $f_{sampling} = f_{CLKIN}$, $N = 4$</p> <p>0011: $f_{sampling} = f_{CLKIN}$, $N = 8$</p> <p>0100: $f_{sampling} = f_{DTS} / 2$, $N = 6$</p> <p>0101: $f_{sampling} = f_{DTS} / 2$, $N = 8$</p> <p>0110: $f_{sampling} = f_{DTS} / 4$, $N = 6$</p> <p>0111: $f_{sampling} = f_{DTS} / 4$, $N = 8$</p> <p>1000: $f_{sampling} = f_{DTS} / 8$, $N = 6$</p> <p>1001: $f_{sampling} = f_{DTS} / 8$, $N = 8$</p> <p>1010: $f_{sampling} = f_{DTS} / 16$, $N = 5$</p> <p>1011: $f_{sampling} = f_{DTS} / 16$, $N = 6$</p> <p>1100: $f_{sampling} = f_{DTS} / 16$, $N = 8$</p> <p>1101: $f_{sampling} = f_{DTS} / 32$, $N = 5$</p> <p>1110: $f_{sampling} = f_{DTS} / 32$, $N = 6$</p> <p>1111: $f_{sampling} = f_{DTS} / 32$, $N = 8$</p>

Channel 3 Input Configuration Register – CH3ICFR

This register specifies the channel 3 input mode configuration.

Offset: 0x02C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved				CH3PSC		CH3CCS	
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved				TI3F			
					RW	0	RW	0
					RW	0	RW	0

Bits	Field	Descriptions
[19:18]	CH3PSC	<p>Channel 3 Capture Input Source Prescaler Setting</p> <p>These bits define the effective events of the channel 3 capture input. Note that the prescaler is reset once the Channel 3 Capture/Compare Enable bit, CH3E, in the Channel Control register named CHCTR is cleared to 0.</p> <p>00: No prescaler, channel 3 capture input signal is chosen for each active event</p> <p>01: Channel 3 Capture input signal is chosen for every 2 events</p> <p>10: Channel 3 Capture input signal is chosen for every 4 events</p> <p>11: Channel 3 Capture input signal is chosen for every 8 events</p>

Bits	Field	Descriptions
[17:16]	CH3CCS	<p>Channel 3 Capture/Compare Selection</p> <p>00: Channel 3 is configured as an output 01: Channel 3 is configured as an input derived from the TI3 signal 10: Channel 3 is configured as an input derived from the TI2 signal 11: Channel 3 is configured as an input which comes from the TRCED signal derived from the Trigger Controller</p> <p>Note: The CH3CCS field can be accessed only when the CH3E bit is cleared to 0.</p>
[3:0]	TI3F	<p>Channel 3 Input Source TI3 Filter Setting</p> <p>These bits define the frequency divided ratio used to sample the TI3 signal. The Digital filter in the GPTM is an N-event counter where N is defined as how many valid transitions are necessary to output a filtered signal.</p> <p>0000: No filter, the sampling clock is f_{SYSTEM} 0001: $f_{sampling} = f_{CLKIN}$, $N = 2$ 0010: $f_{sampling} = f_{CLKIN}$, $N = 4$ 0011: $f_{sampling} = f_{CLKIN}$, $N = 8$ 0100: $f_{sampling} = f_{DTS} / 2$, $N = 6$ 0101: $f_{sampling} = f_{DTS} / 2$, $N = 8$ 0110: $f_{sampling} = f_{DTS} / 4$, $N = 6$ 0111: $f_{sampling} = f_{DTS} / 4$, $N = 8$ 1000: $f_{sampling} = f_{DTS} / 8$, $N = 6$ 1001: $f_{sampling} = f_{DTS} / 8$, $N = 8$ 1010: $f_{sampling} = f_{DTS} / 16$, $N = 5$ 1011: $f_{sampling} = f_{DTS} / 16$, $N = 6$ 1100: $f_{sampling} = f_{DTS} / 16$, $N = 8$ 1101: $f_{sampling} = f_{DTS} / 32$, $N = 5$ 1110: $f_{sampling} = f_{DTS} / 32$, $N = 6$ 1111: $f_{sampling} = f_{DTS} / 32$, $N = 8$</p>

Channel 0 Output Configuration Register – CH0OCFR

This register specifies the channel 0 output mode configuration.

Offset: 0x040

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	Reserved							CH0OM[3]	
									RW 0
	7	6	5	4	3	2	1	0	
Type/Reset	Reserved		CH0IMAE	CH0PRE	Reserved		CH0OM[2:0]		
			RW 0	RW 0			RW 0	RW 0	RW 0

Bits	Field	Descriptions
[5]	CH0IMAE	Channel 0 Immediate Active Enable 0: No action 1: Single pulse Immediate Active Mode is enabled The CH0OREF signal will be forced to the compare matched level immediately after an available trigger event occurs irrespective of the result of the comparison between the CNTR and the CH0CCR values. The effective duration ends automatically at the next overflow or underflow event. Note: The CH0IMAE bit is available only if the channel 0 is configured to be operated in the PWM mode 1 or PWM mode 2.
[4]	CH0PRE	Channel 0 Capture/Compare Register (CH0CCR) Preload Enable 0: CH0CCR preload function is disabled The CH0CCR register can be immediately assigned a new value when the CH0PRE bit is cleared to 0 and the updated CH0CCR value is used immediately. 1: CH0CCR preload function is enabled The new CH0CCR value will not be transferred to its shadow register until the update event occurs.

Bits	Field	Descriptions
[8][2:0]	CH0OM[3:0]	<p>Channel 0 Output Mode Setting</p> <p>These bits define the functional types of the output reference signal CH0OREF.</p> <p>0000: No Change</p> <p>0001: Output 0 on compare match</p> <p>0010: Output 1 on compare match</p> <p>0011: Output toggles on compare match</p> <p>0100: Force inactive – CH0OREF is forced to 0</p> <p>0101: Force active – CH0OREF is forced to 1</p> <p>0110: PWM mode 1</p> <ul style="list-style-type: none"> - During up-counting, channel 0 has an active level when CNTR < CH0CCR or otherwise has an inactive level. - During down-counting, channel 0 has an inactive level when CNTR > CH0CCR or otherwise has an active level. <p>0111: PWM mode 2</p> <ul style="list-style-type: none"> - During up-counting, channel 0 is has an inactive level when CNTR < CH0CCR or otherwise has an active level. - During down-counting, channel 0 has an active level when CNTR > CH0CCR or otherwise has an inactive level. <p>1110: Asymmetric PWM mode 1</p> <ul style="list-style-type: none"> - During up-counting, channel 0 has an active level when CNTR < CH0CCR or otherwise has an inactive level. - During down-counting, channel 0 has an inactive level when CNTR > CH0CCR or otherwise has an active level. <p>1111: Asymmetric PWM mode 2</p> <ul style="list-style-type: none"> - During up-counting, channel 0 has an inactive level when CNTR < CH0CCR or otherwise has an active level. - During down-counting, channel 0 has an active level when CNTR > CH0CCR or otherwise has an inactive level <p>Note: When channel 0 is used as asymmetric PWM output mode, the Counter Mode Selection bit in Counter Configuration Register must be configured as Center-aligned Counting mode (CMSEL = 0x1/0x2/0x3).</p>

Channel 1 Output Configuration Register – CH1OCFR

This register specifies the channel 1 output mode configuration.

Offset: 0x044

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	Reserved							CH1OM[3]	
									RW 0
	7	6	5	4	3	2	1	0	
Type/Reset	Reserved		CH1IMAE	CH1PRE	Reserved	CH1OM[2:0]			
			RW 0	RW 0		RW 0	RW 0	RW 0	

Bits	Field	Descriptions
[5]	CH1IMAE	Channel 1 Immediate Active Enable 0: No action 1: Single pulse Immediate Active Mode is enabled The CH1OREF signal will be forced to the compare matched level immediately after an available trigger event occurs irrespective of the result of the comparison between the CNTR and the CH1CCR values. The effective duration ends automatically at the next overflow or underflow event. Note: The CH1IMAE bit is available only if the channel 1 is configured to be operated in the PWM mode 1 or PWM mode 2.
[4]	CH1PRE	Channel 1 Capture/Compare Register (CH1CCR) Preload Enable 0: CH1CCR preload function is disabled. The CH1CCR register can be immediately assigned a new value when the CH1PRE bit is cleared to 0 and the updated CH1CCR value is used immediately. 1: CH1CCR preload function is enabled The new CH1CCR value will not be transferred to its shadow register until the update event occurs.

Bits	Field	Descriptions
[8][2:0]	CH1OM[3:0]	<p>Channel 1 Output Mode Setting</p> <p>These bits define the functional types of the output reference signal CH1OREF.</p> <p>0000: No Change</p> <p>0001: Output 0 on compare match</p> <p>0010: Output 1 on compare match</p> <p>0011: Output toggles on compare match</p> <p>0100: Force inactive – CH1OREF is forced to 0</p> <p>0101: Force active – CH1OREF is forced to 1</p> <p>0110: PWM mode 1</p> <ul style="list-style-type: none"> - During up-counting, channel 1 has an active level when CNTR < CH1CCR or otherwise has an inactive level. - During down-counting, channel 1 has an inactive level when CNTR > CH1CCR or otherwise has an active level. <p>0111: PWM mode 2</p> <ul style="list-style-type: none"> - During up-counting, channel 1 has an inactive level when CNTR < CH1CCR or otherwise has an active level. - During down-counting, channel 1 has an active level when CNTR > CH1CCR or otherwise has an inactive level. <p>1110: Asymmetric PWM mode 1</p> <ul style="list-style-type: none"> - During up-counting, channel 1 has an active level when CNTR < CH1CCR or otherwise has an inactive level. - During down-counting, channel 1 has an inactive level when CNTR > CH1CCR or otherwise has an active level. <p>1111: Asymmetric PWM mode 2</p> <ul style="list-style-type: none"> - During up-counting, channel 1 has an inactive level when CNTR < CH1CCR or otherwise has an active level. - During down-counting, channel 1 has an active level when CNTR > CH1CCR or otherwise has an inactive level. <p>Note: When channel 1 is used as asymmetric PWM output mode, the Counter Mode Selection bit in Counter Configuration Register must be configured as Center-aligned Counting mode (CMSEL = 0x1/0x2/0x3).</p>

Channel 2 Output Configuration Register – CH2OCFR

This register specifies the channel 2 output mode configuration.

Offset: 0x048

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							CH2OM[3]
								RW 0
	7	6	5	4	3	2	1	0
Type/Reset	Reserved		CH2IMAE	CH2PRE	Reserved	CH2OM[2:0]		
			RW 0	RW 0		RW 0	RW 0	RW 0

Bits	Field	Descriptions
[5]	CH2IMAE	Channel 2 Immediate Active Enable 0: No action 1: Single pulse Immediate Active Mode is enabled The CH2OREF signal will be forced to the compare matched level immediately after an available trigger event occurs irrespective of the result of the comparison between the CNTR and the CH2CCR values. The effective duration ends automatically at the next overflow or underflow event. Note: The CH2IMAE bit is available only if the channel 2 is configured to be operated in the PWM mode 1 or PWM mode 2.
[4]	CH2PRE	Channel 2 Capture/Compare Register (CH2CCR) Preload Enable 0: CH2CCR preload function is disabled. The CH2CCR register can be immediately assigned a new value when the CH2PRE bit is cleared to 0 and the updated CH2CCR value is used immediately. 1: CH2CCR preload function is enabled The new CH2CCR value will not be transferred to its shadow register until the update event occurs.

Bits	Field	Descriptions
[8][2:0]	CH2OM[3:0]	<p>Channel 2 Output Mode Setting</p> <p>These bits define the functional types of the output reference signal CH2OREF.</p> <p>0000: No Change</p> <p>0001: Output 0 on compare match</p> <p>0010: Output 1 on compare match</p> <p>0011: Output toggles on compare match</p> <p>0100: Force inactive – CH2OREF is forced to 0</p> <p>0101: Force active – CH2OREF is forced to 1</p> <p>0110: PWM mode 1</p> <ul style="list-style-type: none"> - During up-counting, channel 2 has an active level when CNTR < CH2CCR or otherwise has an inactive level. - During down-counting, channel 2 has an inactive level when CNTR > CH2CCR or otherwise has an active level. <p>0111: PWM mode 2</p> <ul style="list-style-type: none"> - During up-counting, channel 2 has an inactive level when CNTR < CH2CCR or otherwise has an active level. - During down-counting, channel 2 has an active level when CNTR > CH2CCR or otherwise has an inactive level. <p>1110: Asymmetric PWM mode 1</p> <ul style="list-style-type: none"> - During up-counting, channel 2 has an active level when CNTR < CH2CCR or otherwise has an inactive level. - During down-counting, channel 2 has an inactive level when CNTR > CH2CCR or otherwise has an active level. <p>1111: Asymmetric PWM mode 2</p> <ul style="list-style-type: none"> - During up-counting, channel 2 has an inactive level when CNTR < CH2CCR or otherwise has an active level. - During down-counting, channel 2 has an active level when CNTR > CH2CCR or otherwise has an inactive level. <p>Note: When channel 2 is used as asymmetric PWM output mode, the Counter Mode Selection bit in Counter Configuration Register must be configured as Center-aligned Counting mode (CMSEL = 0x1/0x2/0x3).</p>

Channel 3 Output Configuration Register – CH3OCFR

This register specifies the channel 3 output mode configuration.

Offset: 0x04C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							CH3OM[3]
	7	6	5	4	3	2	1	0
Type/Reset	Reserved		CH3IMAE	CH3PRE	Reserved	CH3OM[2:0]		
			RW	0	RW	0	RW	0

Bits	Field	Descriptions
[5]	CH3IMAE	Channel 3 Immediate Active Enable 0: No action 1: Single pulse Immediate Active Mode is enabled The CH3OREF signal will be forced to the compare matched level immediately after an available trigger event occurs irrespective of the result of the comparison between the CNTR and the CH3CCR values. The effective duration ends automatically at the next overflow or underflow event. Note: The CH3IMAE bit is available only if the channel 3 is configured to be operated in the PWM mode 1 or PWM mode 2.
[4]	CH3PRE	Channel 3 Capture/Compare Register (CH3CCR) Preload Enable 0: CH3CCR preload function is disabled. The CH3CCR register can be immediately assigned a new value when the CH3PRE bit is cleared to 0 and the updated CH3CCR value is used immediately. 1: CH3CCR preload function is enabled The new CH3CCR value will not be transferred to its shadow register until the update event occurs.

Bits	Field	Descriptions
[8][2:0]	CH3OM[3:0]	<p>Channel 3 Output Mode Setting</p> <p>These bits define the functional types of the output reference signal CH3OREF</p> <p>0000: No Change</p> <p>0001: Output 0 on compare match</p> <p>0010: Output 1 on compare match</p> <p>0011: Output toggles on compare match</p> <p>0100: Force inactive – CH3OREF is forced to 0</p> <p>0101: Force active – CH3OREF is forced to 1</p> <p>0110: PWM mode 1</p> <ul style="list-style-type: none"> - During up-counting, channel 3 has an active level when CNTR < CH3CCR or otherwise has an inactive level. - During down-counting, channel 3 has an inactive level when CNTR > CH3CCR or otherwise has an active level. <p>0111: PWM mode 2</p> <ul style="list-style-type: none"> - During up-counting, channel 3 has an inactive level when CNTR < CH3CCR or otherwise has an active level. - During down-counting, channel 3 has an active level when CNTR > CH3CCR or otherwise has an inactive level <p>1110: Asymmetric PWM mode 1</p> <ul style="list-style-type: none"> - During up-counting, channel 3 has an active level when CNTR < CH3CCR or otherwise has an inactive level. - During down-counting, channel 3 has an inactive level when CNTR > CH3CCR or otherwise has an active level. <p>1111: Asymmetric PWM mode 2</p> <ul style="list-style-type: none"> - During up-counting, channel 3 has an inactive level when CNTR < CH3CCR or otherwise has an active level. - During down-counting, channel 3 has an active level when CNTR > CH3CCR or otherwise has an inactive level <p>Note: When channel 3 is used as asymmetric PWM output mode, the Counter Mode Selection bit in Counter Configuration Register must be configured as Center-aligned Counting mode (CMSEL = 0x1/0x2/0x3).</p>

Channel Control Register – CHCTR

This register contains the channel capture input or compare output function enable control bits.

Offset: 0x050

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved	CH3E	Reserved	CH2E	Reserved	CH1E	Reserved	CH0E
	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[6]	CH3E	<p>Channel 3 Capture/Compare Enable</p> <ul style="list-style-type: none"> - Channel 3 is configured as an input (CH3CCS = 0x1/0x2/0x3) <ul style="list-style-type: none"> 0: Input Capture Mode is disabled 1: Input Capture Mode is enabled - Channel 3 is configured as an output (CH3CCS = 0x0) <ul style="list-style-type: none"> 0: Off – Channel 3 output signal CH3O is not active 1: On – Channel 3 output signal CH3O is generated on the corresponding output pin
[4]	CH2E	<p>Channel 2 Capture/Compare Enable</p> <ul style="list-style-type: none"> - Channel 2 is configured as an input (CH2CCS = 0x1/0x2/0x3) <ul style="list-style-type: none"> 0: Input Capture Mode is disabled 1: Input Capture Mode is enabled - Channel 2 is configured as an output (CH2CCS = 0x0) <ul style="list-style-type: none"> 0: Off – Channel 2 output signal CH2O is not active 1: On – Channel 2 output signal CH2O is generated on the corresponding output pin
[2]	CH1E	<p>Channel 1 Capture/Compare Enable</p> <ul style="list-style-type: none"> - Channel 1 is configured as an input (CH1CCS = 0x1/0x2/0x3) <ul style="list-style-type: none"> 0: Input Capture Mode is disabled 1: Input Capture Mode is enabled - Channel 1 is configured as an output (CH1CCS = 0x0) <ul style="list-style-type: none"> 0: Off – Channel 1 output signal CH1O is not active 1: On – Channel 1 output signal CH1O is generated on the corresponding output pin

Bits	Field	Descriptions
[0]	CH0E	Channel 0 Capture/Compare Enable - Channel 0 is configured as an input (CH0CCS = 0x1/0x2/0x3) 0: Input Capture Mode is disabled 1: Input Capture Mode is enabled - Channel 0 is configured as an output (CH0CCS = 0x0) 0: Off – Channel 0 output signal CH0O is not active 1: On – Channel 0 output signal CH0O is generated on the corresponding output pin

Channel Polarity Configuration Register – CHPOLR

This register contains the channel capture input or compare output polarity control.

Offset: 0x054

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved	CH3P	Reserved	CH2P	Reserved	CH1P	Reserved	CH0P
	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[6]	CH3P	Channel 3 Capture/Compare Polarity - When Channel 3 is configured as an input (CH3CCS = 0x1/0x2/0x3) 0: capture event occurs on a Channel 3 rising edge 1: capture event occurs on a Channel 3 falling edge - When Channel 3 is configured as an output (CH3CCS = 0x0) 0: Channel 3 Output is active high 1: Channel 3 Output is active low
[4]	CH2P	Channel 2 Capture/Compare Polarity - When Channel 2 is configured as an input (CH2CCS = 0x1/0x2/0x3) 0: capture event occurs on a Channel 2 rising edge 1: capture event occurs on a Channel 2 falling edge - When Channel 2 is configured as an output (CH2CCS = 0x0) 0: Channel 2 Output is active high 1: Channel 2 Output is active low

Bits	Field	Descriptions
[2]	CH1P	Channel 1 Capture/Compare Polarity - When Channel 1 is configured as an input (CH1CCS = 0x1/0x2/0x3) 0: capture event occurs on a Channel 1 rising edge 1: capture event occurs on a Channel 1 falling edge - Channel 1 is configured as an output (CH1CCS = 0x0) 0: Channel 1 Output is active high 1: Channel 1 Output is active low
[0]	CH0P	Channel 0 Capture/Compare Polarity - When Channel 0 is configured as an input (CH0CCS = 0x1/0x2/0x3) 0: capture event occurs on a Channel 0 rising edge 1: capture event occurs on a Channel 0 falling edge - When Channel 0 is configured as an output (CH0CCS = 0x0) 0: Channel 0 Output is active high 1: Channel 0 Output is active low

Timer PDMA/Interrupt Control Register – DICTR

This register contains the timer PDMA and interrupt enable control bits.

Offset: 0x074

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved					TEVDE	Reserved	UEVDE
Type/Reset						RW 0		RW 0
	23	22	21	20	19	18	17	16
	Reserved				CH3CCDE	CH2CCDE	CH1CCDE	CH0CCDE
Type/Reset					RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	Reserved					TEVIE	Reserved	UEVIE
Type/Reset						RW 0		RW 0
	7	6	5	4	3	2	1	0
	Reserved				CH3CCIE	CH2CCIE	CH1CCIE	CH0CCIE
Type/Reset					RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[26]	TEVDE	Trigger event PDMA Request Enable 0: Trigger PDMA request is disabled 1: Trigger PDMA request is enabled
[24]	UEVDE	Update event PDMA Request Enable 0: Update event PDMA request is disabled 1: Update event PDMA request is enabled
[19]	CH3CCDE	Channel 3 Capture/Compare PDMA Request Enable 0: Channel 3 PDMA request is disabled 1: Channel 3 PDMA request is enabled
[18]	CH2CCDE	Channel 2 Capture/Compare PDMA Request Enable 0: Channel 2 PDMA request is disabled 1: Channel 2 PDMA request is enabled

Bits	Field	Descriptions
[17]	CH1CCDE	Channel 1 Capture/Compare PDMA Request Enable 0: Channel 1 PDMA request is disabled 1: Channel 1 PDMA request is enabled
[16]	CH0CCDE	Channel 0 Capture/Compare PDMA Request Enable 0: Channel 0 PDMA request is disabled 1: Channel 0 PDMA request is enabled
[10]	TEVIE	Trigger event Interrupt Enable 0: Trigger event interrupt is disabled 1: Trigger event interrupt is enabled
[8]	UEVIE	Update event Interrupt Enable 0: Update event interrupt is disabled 1: Update event interrupt is enabled
[3]	CH3CCIE	Channel 3 Capture/Compare Interrupt Enable 0: Channel 3 interrupt is disabled 1: Channel 3 interrupt is enabled
[2]	CH2CCIE	Channel 2 Capture/Compare Interrupt Enable 0: Channel 2 interrupt is disabled 1: Channel 2 interrupt is enabled
[1]	CH1CCIE	Channel 1 Capture/Compare Interrupt Enable 0: Channel 1 interrupt is disabled 1: Channel 1 interrupt is enabled
[0]	CH0CCIE	Channel 0 Capture/Compare Interrupt Enable 0: Channel 0 interrupt is disabled 1: Channel 0 interrupt is enabled

Timer Event Generator Register – EVGR

This register contains the software event generation bits.

Offset: 0x078

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved					TEVG	Reserved	UEVG
	7	6	5	4	3	2	1	0
Type/Reset	Reserved				CH3CCG	CH2CCG	CH1CCG	CH0CCG
					WO 0	WO 0	WO 0	WO 0

Bits	Field	Descriptions
[10]	TEVG	<p>Trigger Event Generation</p> <p>The trigger event TEV can be generated by setting this bit. It is cleared by hardware automatically.</p> <p>0: No action 1: TEVIF flag is set</p>
[8]	UEVG	<p>Update Event Generation</p> <p>The update event UEV can be generated by setting this bit. It is cleared by hardware automatically.</p> <p>0: No action 1: Reinitialize the counter</p> <p>The counter value returns to 0 or the CRR preload value, depending on the counter mode in which the current timer is being used. An update operation of any related registers will also be performed. For more detailed descriptions, refer to the corresponding section.</p>
[3]	CH3CCG	<p>Channel 3 Capture/Compare Generation</p> <p>A Channel 3 capture/compare event can be generated by setting this bit. It is cleared by hardware automatically.</p> <p>0: No action 1: Capture/compare event is generated on channel 3</p> <p>If Channel 3 is configured as an input, the counter value is captured into the CH3CCR register and then the CH3CCIF bit is set. If Channel 3 is configured as an output, the CH3CCIF bit is set.</p>
[2]	CH2CCG	<p>Channel 2 Capture/Compare Generation</p> <p>A Channel 2 capture/compare event can be generated by setting this bit. It is cleared by hardware automatically.</p> <p>0: No action 1: Capture/compare event is generated on channel 2</p> <p>If Channel 2 is configured as an input, the counter value is captured into the CH2CCR register and then the CH2CCIF bit is set. If Channel 2 is configured as an output, the CH2CCIF bit is set.</p>

Bits	Field	Descriptions
[1]	CH1CCG	Channel 1 Capture/Compare Generation A Channel 1 capture/compare event can be generated by setting this bit. It is cleared by hardware automatically. 0: No action 1: Capture/compare event is generated on channel 1 If Channel 1 is configured as an input, the counter value is captured into the CH1CCR register and then the CH1CCIF bit is set. If Channel 1 is configured as an output, the CH1CCIF bit is set.
[0]	CH0CCG	Channel 0 Capture/Compare Generation A Channel 0 capture/compare event can be generated by setting this bit. It is cleared by hardware automatically. 0: No action 1: Capture/compare event is generated on channel 0 If Channel 0 is configured as an input, the counter value is captured into the CH0CCR register and then the CH0CCIF bit is set. If Channel 0 is configured as an output, the CH0CCIF bit is set.

Timer Interrupt Status Register – INTSR

This register stores the timer interrupt status.

Offset: 0x07C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
	Reserved								
Type/Reset									
	23	22	21	20	19	18	17	16	
	Reserved								
Type/Reset									
	15	14	13	12	11	10	9	8	
	Reserved					TEVIF	Reserved	UEVIF	
Type/Reset						W0C 0		W0C 0	
	7	6	5	4	3	2	1	0	
	CH3OCF	CH2OCF	CH1OCF	CH0OCF	CH3CCIF	CH2CCIF	CH1CCIF	CH0CCIF	
Type/Reset	W0C 0	W0C 0	W0C 0	W0C 0	W0C 0	W0C 0	W0C 0	W0C 0	

Bits	Field	Descriptions
[10]	TEVIF	Trigger Event Interrupt Flag This flag is set by hardware on a trigger event and is cleared by software. 0: No trigger event occurs 1: Trigger event occurs

Bits	Field	Descriptions
[8]	UEVIF	Update Event Interrupt Flag This bit is set by hardware on an update event and is cleared by software. 0: No update event occurs 1: Update event occurs Note: The update event is derived from the following conditions: - The counter overflows or underflows - The UEVG bit is asserted - A restart trigger event occurs from the slave trigger input
[7]	CH3OCF	Channel 3 Over-Capture Flag This flag is set by hardware and cleared by software. 0: No over-capture event is detected 1: Capture event occurs again when the CH3CCIF bit is already set and it is not yet cleared by software
[6]	CH2OCF	Channel 2 Over-Capture Flag This flag is set by hardware and cleared by software. 0: No over-capture event is detected 1: Capture event occurs again when the CH2CCIF bit is already set and it is not cleared yet by software
[5]	CH1OCF	Channel 1 Over-Capture Flag This flag is set by hardware and cleared by software. 0: No over-capture event is detected 1: Capture event occurs again when the CH1CCIF bit is already set and it is not cleared yet by software.
[4]	CH0OCF	Channel 0 Over-Capture Flag This flag is set by hardware and cleared by software. 0: No over-capture event is detected 1: Capture event occurs again when the CH0CCIFbit is already set and it is not yet cleared by software.
[3]	CH3CCIF	Channel 3 Capture/Compare Interrupt Flag - Channel 3 is configured as an output: 0: No match event occurs 1: The content of the counter CNTR has matched the content of the CH3CCR register This flag is set by hardware when the counter value matches the CH3CCR value except in the center-aligned mode. It is cleared by software. - Channel 3 is configured as an input: 0: No input capture occurs 1: Input capture occurs This bit is set by hardware on a capture event. It is cleared by software or by reading the CH3CCR register.
[2]	CH2CCIF	Channel 2 Capture/Compare Interrupt Flag - Channel 2 is configured as an output: 0: No match event occurs 1: The content of the counter CNTR has matched the content of the CH2CCR register This flag is set by hardware when the counter value matches the CH2CCR value except in the center-aligned mode. It is cleared by software. - Channel 2 is configured as an input: 0: No input capture occurs 1: Input capture occurs. This bit is set by hardware on a capture event. It is cleared by software or by reading the CH2CCR register.

Bits	Field	Descriptions
[1]	CH1CCIF	<p>Channel 1 Capture/Compare Interrupt Flag</p> <ul style="list-style-type: none"> - Channel 1 is configured as an output: <ul style="list-style-type: none"> 0: No match event occurs 1: The content of the counter CNTR has matched the content of the CH1CCR register <p>This flag is set by hardware when the counter value matches the CH1CCR value except in the center-aligned mode. It is cleared by software.</p> - Channel 1 is configured as an input: <ul style="list-style-type: none"> 0: No input capture occurs 1: Input capture occurs <p>This bit is set by hardware on a capture event. It is cleared by software or by reading the CH1CCR register.</p>
[0]	CH0CCIF	<p>Channel 0 Capture/Compare Interrupt Flag</p> <ul style="list-style-type: none"> - Channel 0 is configured as an output: <ul style="list-style-type: none"> 0: No match event occurs 1: The content of the counter CNTR has matched the content of the CH0CCR register <p>This flag is set by hardware when the counter value matches the CH0CCR value except in the center-aligned mode. It is cleared by software.</p> - Channel 0 is configured as an input: <ul style="list-style-type: none"> 0: No input capture occurs 1: Input capture occurs <p>This bit is set by hardware on a capture event. It is cleared by software or by reading the CH0CCR register.</p>

Timer Counter Register – CNTR

This register stores the timer counter value.

Offset: 0x080

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	CNTV								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	CNTV								
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:0]	CNTV	Counter Value.

Timer Prescaler Register – PSCR

This register specifies the timer prescaler value to generate the counter clock.

Offset: 0x084

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	PSCV								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	PSCV								
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:0]	PSCV	<p>Prescaler Value</p> <p>These bits are used to specify the prescaler value to generate the counter clock frequency f_{CK_CNT}.</p> $f_{CK_CNT} = \frac{f_{CK_PSC}}{PSCV[15:0]+1}$ <p>where the f_{CK_PSC} is the prescaler clock source.</p>

Timer Counter-Reload Register – CRR

This register specifies the timer counter-reload value.

Offset: 0x088

Reset value: 0x0000_FFFF

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	CRV								
	RW	1	RW	1	RW	1	RW	1	RW
	7	6	5	4	3	2	1	0	
Type/Reset	CRV								
	RW	1	RW	1	RW	1	RW	1	RW

Bits	Field	Descriptions
[15:0]	CRV	Counter-Reload Value The CRV is the reload value which is loaded into the actual counter register.

Channel 0 Capture/Compare Register – CH0CCR

This register specifies the timer channel 0 capture/compare value.

Offset: 0x090

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	CH0CCV								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	CH0CCV								
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:0]	CH0CCV	<p>Channel 0 Capture/Compare Value</p> <ul style="list-style-type: none"> - When Channel 0 is configured as an output: The CH0CCR value is compared with the counter value and the comparison result is used to trigger the CH0OREF output signal. - When Channel 0 is configured as an input: The CH0CCR register stores the counter value captured by the last channel 0 capture event.

Channel 1 Capture/Compare Register – CH1CCR

This register specifies the timer channel 1 capture/compare value.

Offset: 0x094

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	CH1CCV								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	CH1CCV								
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:0]	CH1CCV	<p>Channel 1 Capture/Compare Value</p> <ul style="list-style-type: none"> - When Channel 1 is configured as an output: The CH1CCR value is compared with the counter value and the comparison result is used to trigger the CH1OREF output signal. - When Channel 1 is configured as an input: The CH1CCR register stores the counter value captured by the last channel 1 capture event.

Channel 2 Capture/Compare Register – CH2CCR

This register specifies the timer channel 2 capture/compare value.

Offset: 0x098

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	CH2CCV								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	CH2CCV								
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:0]	CH2CCV	<p>Channel 2 Capture/Compare Value</p> <ul style="list-style-type: none"> - When Channel 2 is configured as an output: The CH2CCR value is compared with the counter value and the comparison result is used to trigger the CH2OREF output signal. - When Channel 2 is configured as an input: The CH2CCR register stores the counter value captured by the last channel 2 capture event.

Channel 3 Capture/Compare Register – CH3CCR

This register specifies the timer channel 3 capture/compare value.

Offset: 0x09C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	CH3CCV								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	CH3CCV								
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:0]	CH3CCV	<p>Channel 3 Capture/Compare Value</p> <ul style="list-style-type: none"> - When Channel 3 is configured as an output: The CH3CCR value is compared with the counter value and the comparison result is used to trigger the CH3OREF output signal. - When Channel 3 is configured as an input: The CH3CCR register stores the counter value captured by the last channel 3 capture event.

Channel 0 Asymmetric Compare Register – CH0ACR

This register specifies the timer channel 0 asymmetric compare value.

Offset: 0x0A0

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	CH0ACV								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	CH0ACV								
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:0]	CH0ACV	Channel 0 Asymmetric Compare Value When channel 0 is configured as asymmetric PWM mode and the counter is counting down, the value written in this register will be compared to the counter.

Channel 1 Asymmetric Compare Register – CH1ACR

This register specifies the timer channel 1 asymmetric compare value.

Offset: 0x0A4

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	CH1ACV								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	CH1ACV								
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:0]	CH1ACV	Channel 1 Asymmetric Compare Value When channel 1 is configured as asymmetric PWM mode and the counter is counting down, the value written in this register will be compared to the counter.

Channel 2 Asymmetric Compare Register – CH2ACR

This register specifies the timer channel 2 asymmetric compare value.

Offset: 0x0A8

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	CH2ACV								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	CH2ACV								
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:0]	CH2ACV	Channel 2 Asymmetric Compare Value When channel 2 is configured as asymmetric PWM mode and the counter is counting down, the value written in this register will be compared to the counter.

Channel 3 Asymmetric Compare Register – CH3ACR

This register specifies the timer channel 3 asymmetric compare value.

Offset: 0x0AC

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	CH3ACV								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	CH3ACV								
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:0]	CH3ACV	Channel 3 Asymmetric Compare Value When channel 3 is configured as asymmetric PWM mode and the counter is counting down, the value written in this register will be compared to the counter.

13 Pulse-Width-Modulation Timer (PWM)

Introduction

The Pulse-Width-Modulation Timer consists of one 16-bit up/down-counter, four 16-bit Compare Registers (CRs), one 16-bit Counter-Reload Register (CRR) and several control/status registers. It can be used for a variety of purposes including general timer and output waveform generation such as single pulse generation or PWM output.

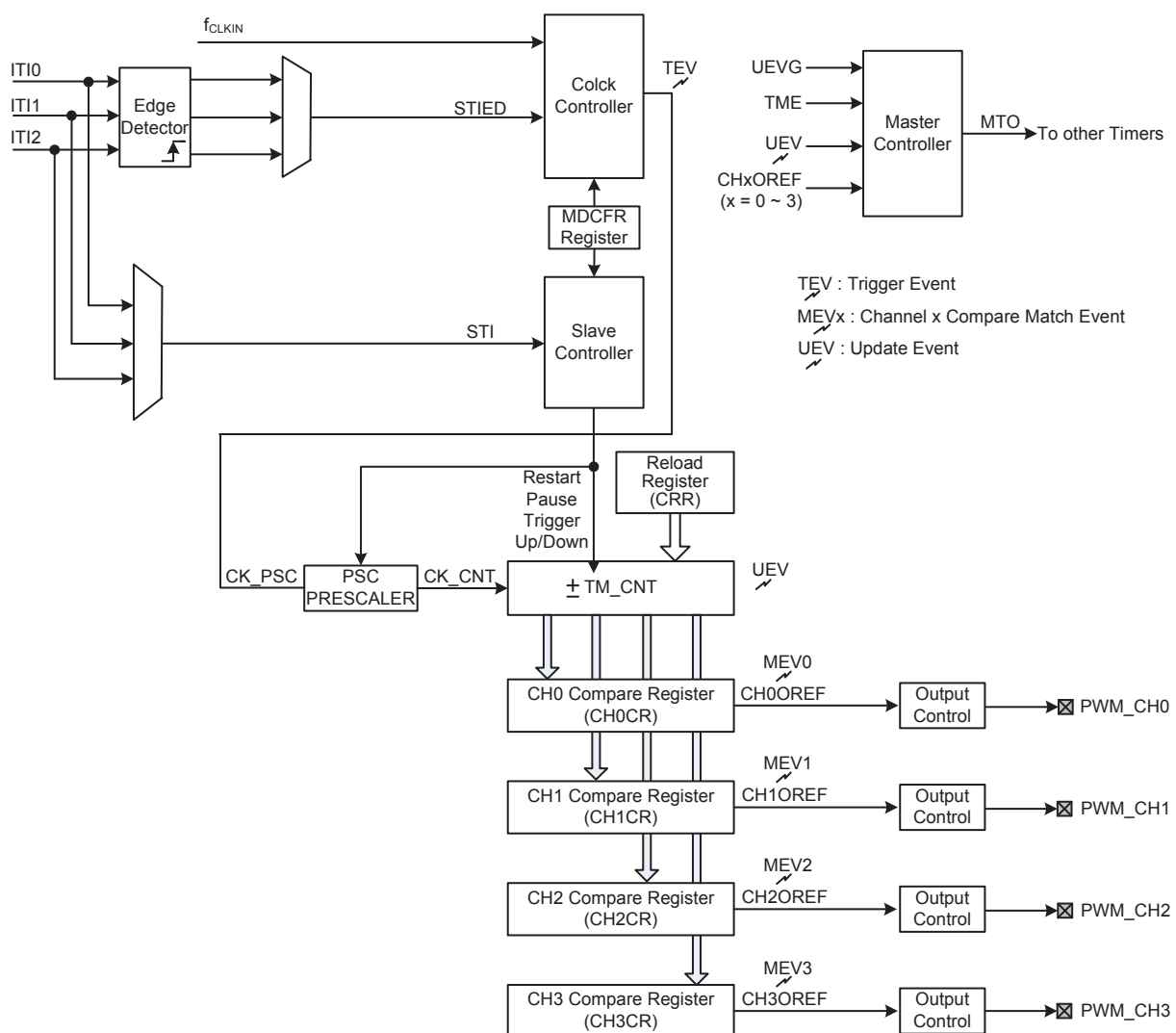


Figure 62. PWM Block Diagram

Features

- 16-bit up/down auto-reload counter
- 16-bit programmable prescaler that allows division of the prescaler clock source by any factor between 1 and 65536 to generate the counter clock frequency
- Up to 4 independent channels for:
 - Compare Match Output
 - Generation of PWM waveform - Edge and Center-aligned Mode
 - Single Pulse Mode Output
- Synchronization circuit to control the timer with external signals and to interconnect several timers together
- Interrupt/PDMA generation with the following events:
 - Update event
 - Trigger event
 - Output compare match event
- PWM Master/Slave mode controller

Functional Descriptions

Counter Mode

Up-Counting

In this mode the counter counts continuously from 0 to the counter-reload value, which is defined in the CRR register, in a count-up direction. Once the counter reaches the counter-reload value, the Timer Module generates an overflow event and the counter restarts to count once again from 0. This action will continue repeatedly. The counting direction bit DIR in the CNTCFR register should be set to 0 for the up-counting mode.

When the update event is generated by setting the UEVG bit in the EVGR register to 1, the counter value will also be initialized to 0.

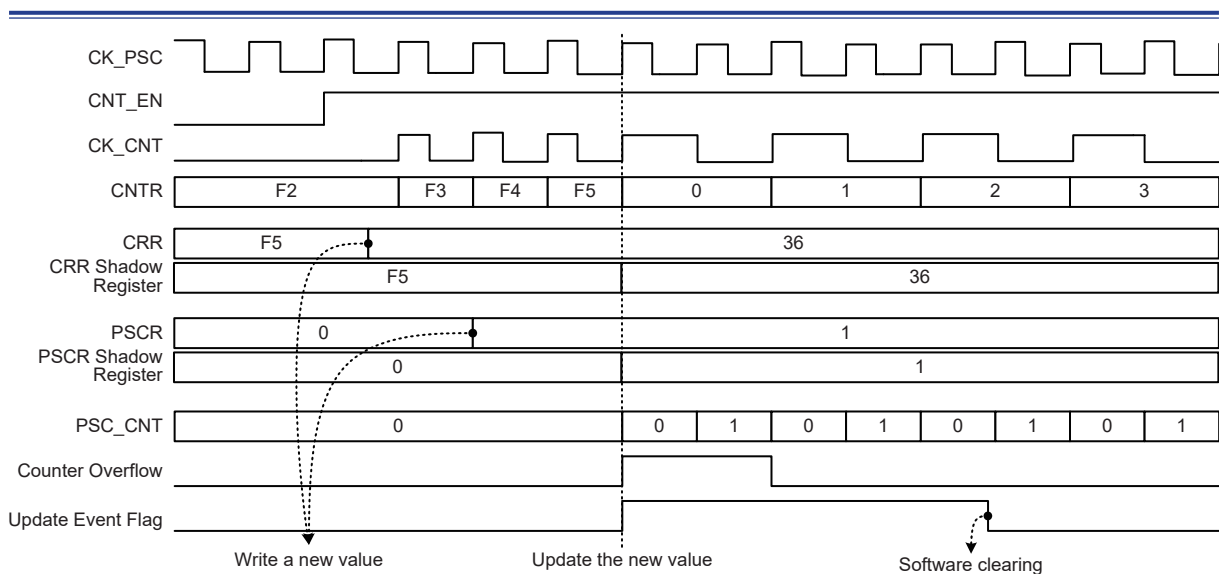


Figure 63. Up-counting Example

Down-Counting

In this mode the counter counts continuously from the counter-reload value, which is defined in the CRR register, to 0 in a count-down direction. Once the counter reaches 0, the Timer module generates an underflow event and the counter restarts to count once again from the counter-reload value. This action will continue repeatedly. The counting direction bit DIR in the CNTCFR register should be set to 1 for the down-counting mode.

When the update event is set by the UEVG bit in the EVGR register, the counter value will also be initialized to the counter-reload value.

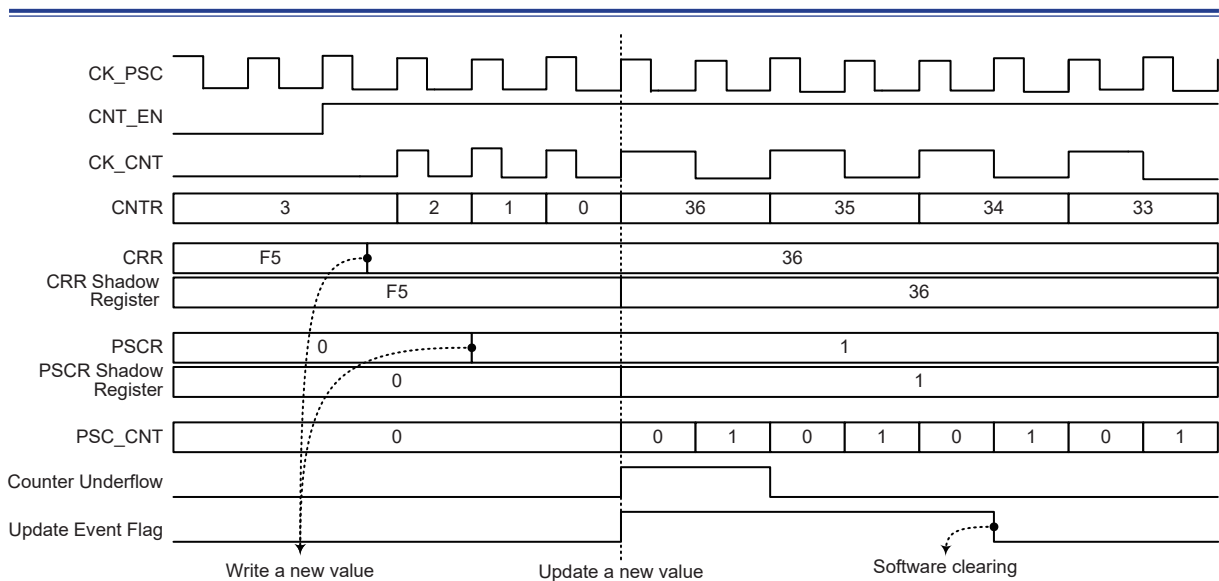


Figure 64. Down-counting Example

Center-Aligned Counting

In the center-aligned counting mode, the counter counts up from 0 to the counter-reload value and then counts down to 0 alternatively. The Timer module generates an overflow event when the counter counts to the counter-reload value in the up-counting mode and generates an underflow event when the counter counts to 0 in the down-counting mode. The counting direction bit DIR in the CNTCFR register is read-only and indicates the counting direction when in the center-align mode. The counting direction is updated by hardware automatically.

Setting the UEVG bit in the EVGR register will initialize the counter value to 0 irrespective of whether the counter is counting up or down in the center-align counting mode.

The UEVIF bit in the INTSR register can be set to 1 when an overflow or underflow event or both of them occur according to the CMSEL field setting in the CNTCFR register.

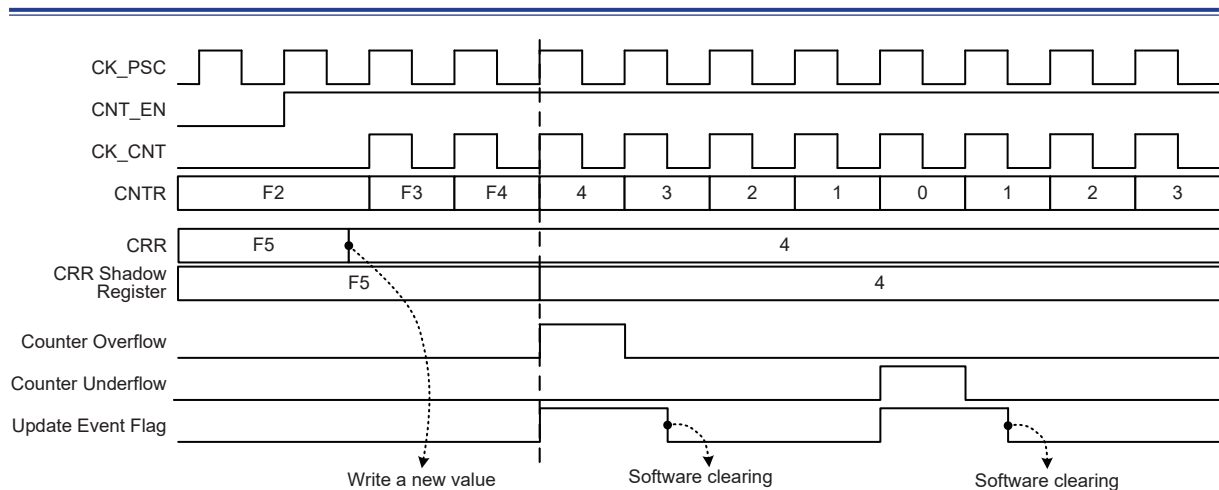


Figure 65. Center-aligned Counting Example

Clock Controller

The following describes the Timer Module clock controller which determines the clock source of the internal prescaler counter.

■ Internal APB clock f_{CLKIN} :

The default internal clock source is the APB clock f_{CLKIN} used to drive the counter prescaler.

■ STIED:

The counter prescaler can count during each rising edge of the STI signal. This mode can be selected by setting the SMSEL field to 0x7 in the MDCFR register. Here the counter will act as an event counter. The input event, known as STI here, can be selected by setting the TRSEL field to an available value except the value of 0x0. When the STI signal is selected as the clock source, the internal edge detection circuitry will generate a clock pulse during each STI signal rising edge to drive the counter prescaler. It is important to note that if the TRSEL field is set to 0x0 to select the software UEVG bit as the trigger source, then when the SMSEL field is set to 0x7, the counter will be updated instead of counting.

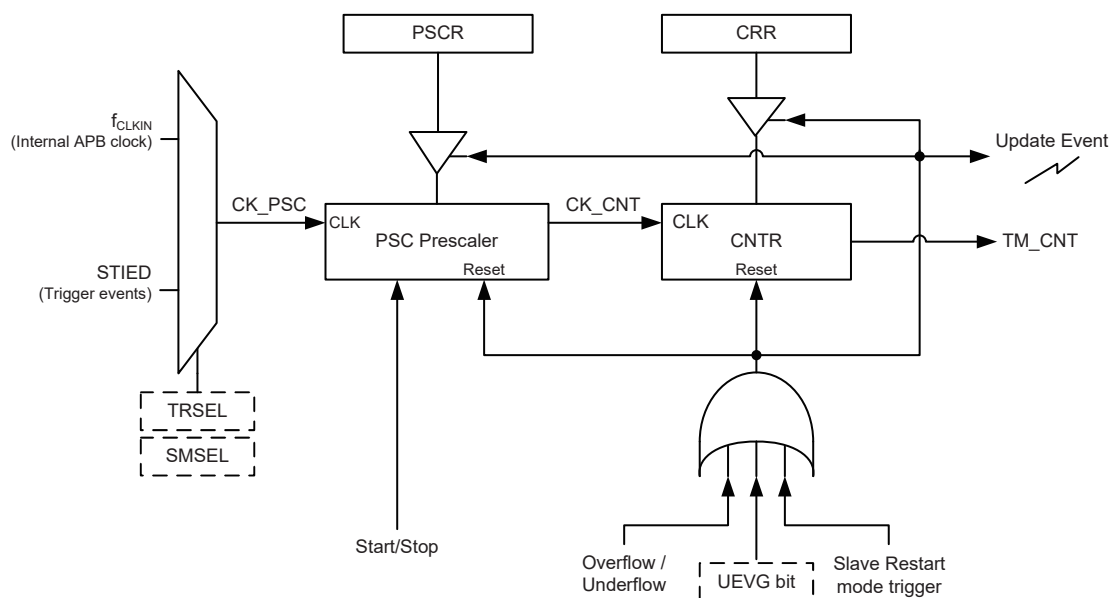


Figure 66. PWM Clock Selection Source

Trigger Controller

The trigger controller is used to select the trigger source and setup the trigger level or edge trigger condition. For the internal trigger input, it can be selected by the Trigger Selection bits TRSEL in the TRCFR register. For all the trigger sources except the UEVG bit software trigger, the internal edge detection circuitry will generate a clock pulse at each trigger signal rising edge to stimulate some PWM functions which are triggered by a trigger signal rising edge.

Trigger Controller Block = Edge Trigger Mux + Level Trigger Mux

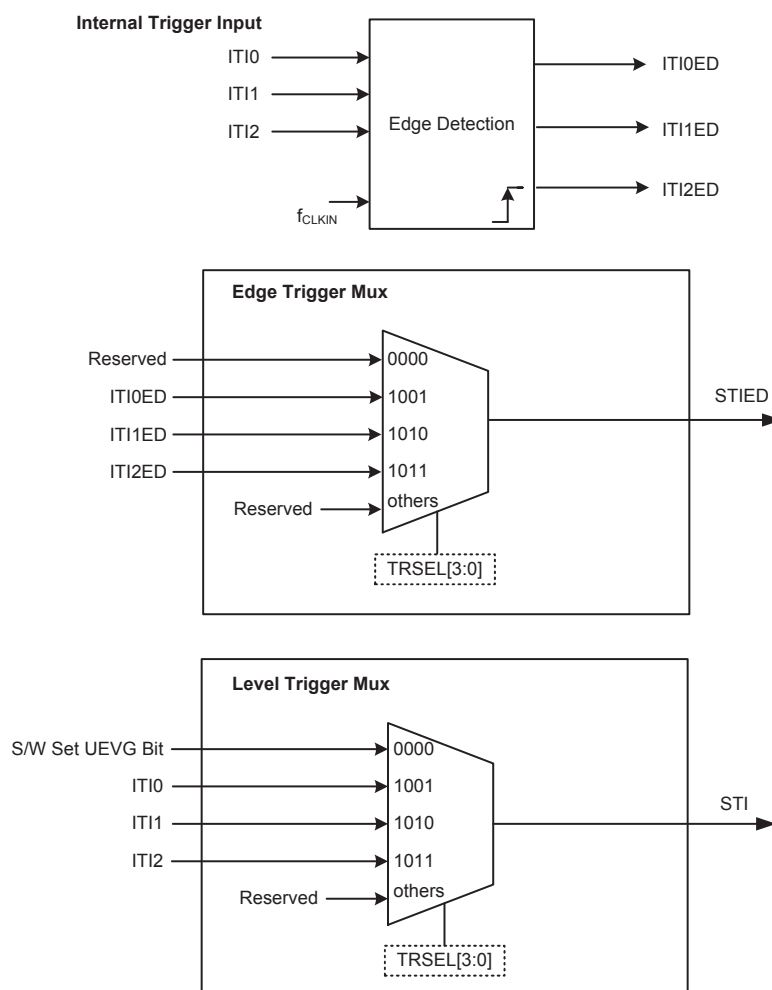


Figure 67. Trigger Control Block

Slave Controller

The PWM can be synchronized with an external trigger in several modes including the Restart mode, the Pause mode and the Trigger mode which is selected by the SMSEL field in the MDCFR register. The trigger input of these modes comes from the STI signal which is selected by the TRSEL field in the TRCFR register. The operation modes in the Slave Controller are described in the accompanying sections.

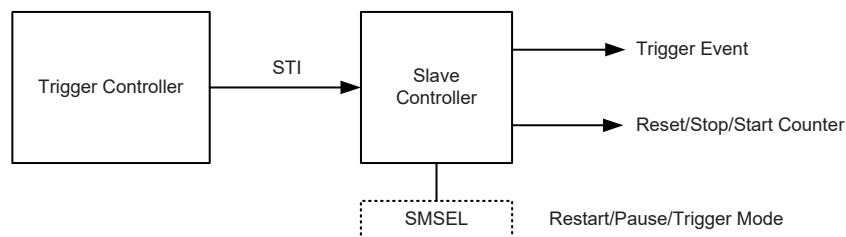


Figure 68. Slave Controller Diagram

Restart Mode

The counter and its prescaler can be reinitialized in response to a rising edge of the STI signal. When a STI rising edge occurs, the update event software generation bit named UEVG will automatically be asserted by hardware and the trigger event flag will also be set. Then the counter and prescaler will be reinitialized. Although the UEVG bit is set to 1 by hardware, the update event does not really occur. It depends upon whether the update event disable control bit UEVDIS is set to 1 or not. If the UEVDIS is set to 1 to disable the update event to occur, there will no update event be generated, however the counter and prescaler are still reinitialized when the STI rising edge occurs. If the UEVDIS bit in the CNTCFR register is cleared to enable the update event to occur, an update event will be generated together with the STI rising edge, then all the preloaded registers will be updated.

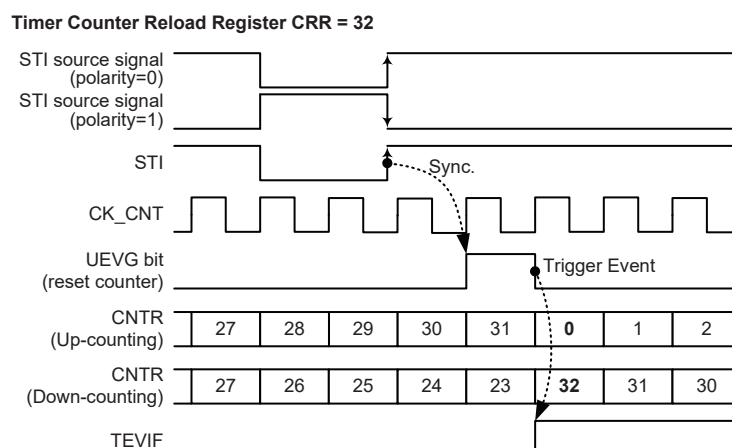


Figure 69. PWM in Restart Mode

Pause Mode

In the Pause Mode, the selected STI input signal level is used to control the counter start/stop operation. The counter starts to count when the selected STI signal is at a high level and stops counting when the STI signal is changed to a low level, here the counter will maintain its present value and will not be reset. Since the Pause function depends upon the STI level to control the counter stop/start operation.

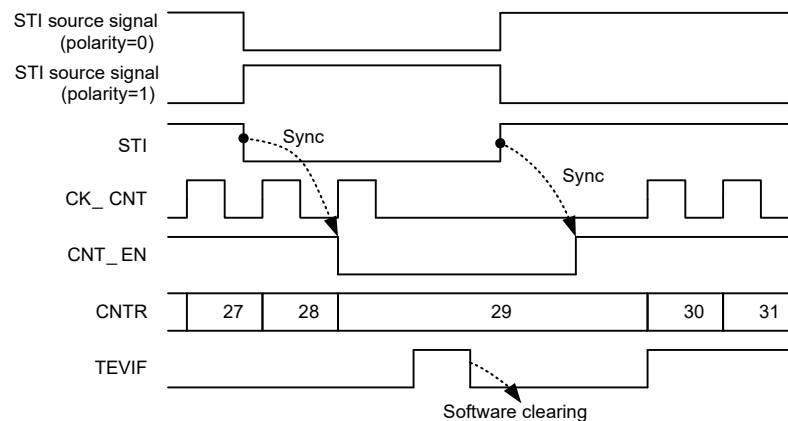


Figure 70. PWM in Pause Mode

Trigger Mode

After the counter is disabled to count, the counter can resume counting when a STI rising edge signal occurs. When an STI rising edge occurs, the counter will start to count from the current value in the counter. Note that if the STI signal is selected to be derived from the UEVG bit software trigger, the counter will not resume counting. When software triggering using the UEVG bit is selected as the STI source signal, there will be no clock pulse generated which can be used to make the counter resume counting. Note that the STI signal is only used to enable the counter to resume counting and has no effect on controlling the counter to stop counting.

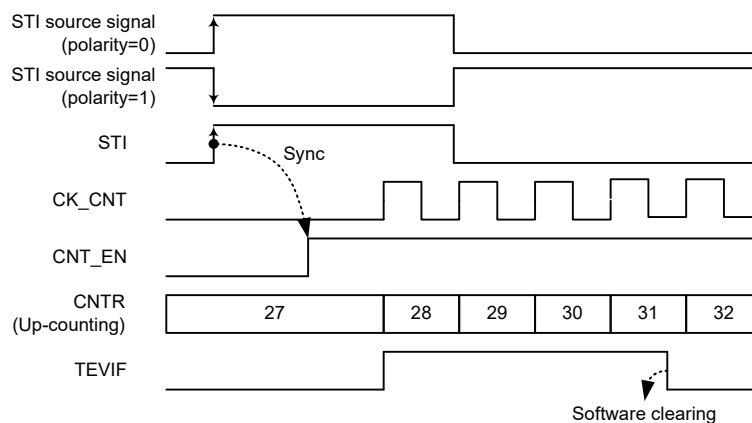


Figure 71. PWM in Trigger Mode

Master Controller

The PWMs and TMs can be linked together internally for timer synchronization or chaining. When one PWM is configured to be in the Master Mode, the PWM Master Controller will generate a Master Trigger Output (MTO) signal which includes a reset, a start, a stop signal or a clock source which is selected by the MMSEL field in the MDCFR register to trigger or drive another PWM or TM, if exists, which is configured in the Slave Mode.

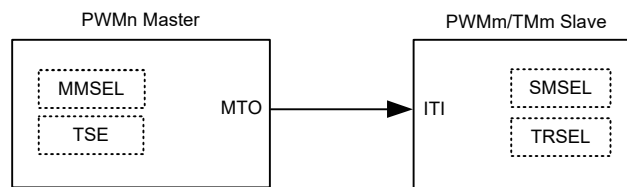


Figure 72. Master PWMn and Slave PWMm/TMm Connection

The Master Mode Selection bits, MMSEL, in the MDCFR register are used to select the MTO source for synchronizing another slave PWM or TM if exists.

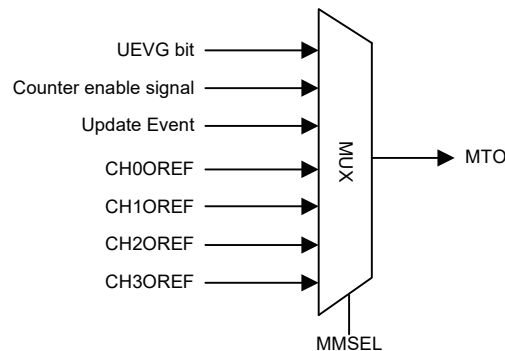


Figure 73. MTO Selection

For example, setting the MMSEL field to 0x5 is to select the CH1OREF signal as the MTO signal to synchronize another slave PWM or TM. For a more detailed description, refer to the related MMSEL field definitions in the MDCFR register.

Channel Controller

The PWM has four independent channels which can be used as compare match outputs. Each compare match output channel is composed of a preload register and a shadow register. Data access of the APB bus is always implemented by reading/writing preload register.

When used in the compare match output mode, the contents of the CHxCR preload register is copied into the associated shadow register; the counter value is then compared with the register value.

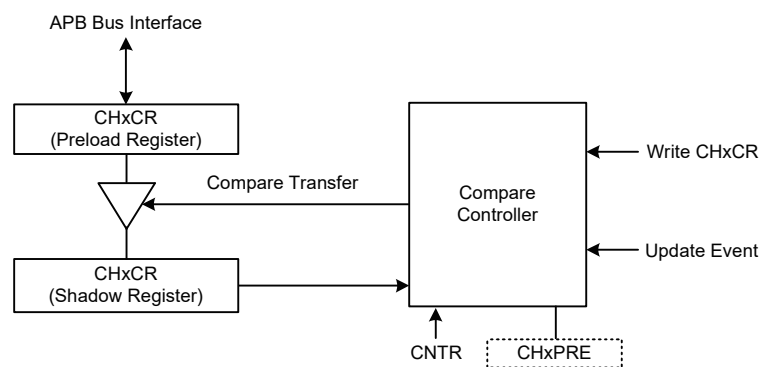


Figure 74. Compare Block Diagram

Output Stage

The PWM has four channels for compare match, single pulse or PWM output function. The channel output PWM_CHx is controlled by the CHxOM, CHxP and CHxE bits in the corresponding CHxOCFR, CHPOLR and CHCTR registers.

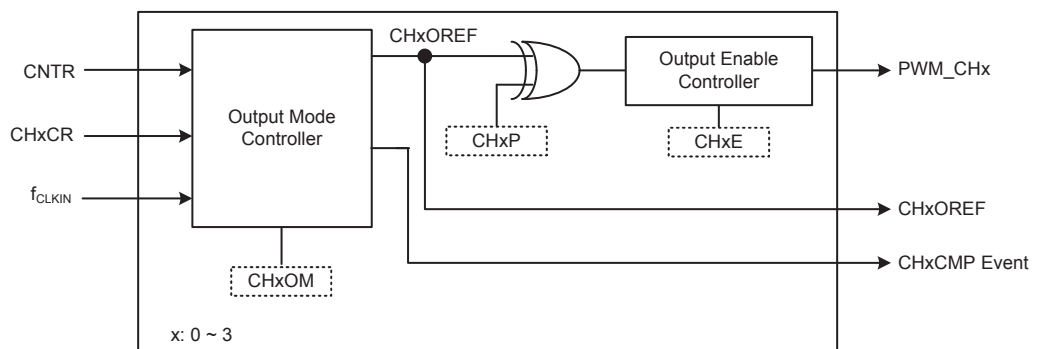


Figure 75. Output Stage Block Diagram

Channel Output Reference Signal

When the PWM is used in the compare match output mode, the Channel x Output Reference signal, CHxOREF, is defined by the CHxOM field setup. The CHxOREF signal has several types of output function which defines what happens to the output when the counter value matches the contents of the CHxCR register. In addition to the low, high and toggle CHxOREF output types; there are also PWM mode 1 and PWM mode 2 outputs. In these modes, the CHxOREF signal level is changed according to the count direction and the relationship between the counter value and the CHxCR content. There are also two modes which will force the output into an inactive or active state irrespective of the CHxCR content or counter values. With regard to a more detailed description refer to the relative bit definition. The accompanying Table 33 shows a summary of the output type setup.

Table 33. Compare Match Output Setup

CHxOM Value	Compare Match Level
0x0	No change
0x1	Clear Output to 0
0x2	Set Output to 1
0x3	Toggle Output
0x4	Force Inactive Level
0x5	Force Active Level
0x6	PWM Mode 1
0x7	PWM Mode 2

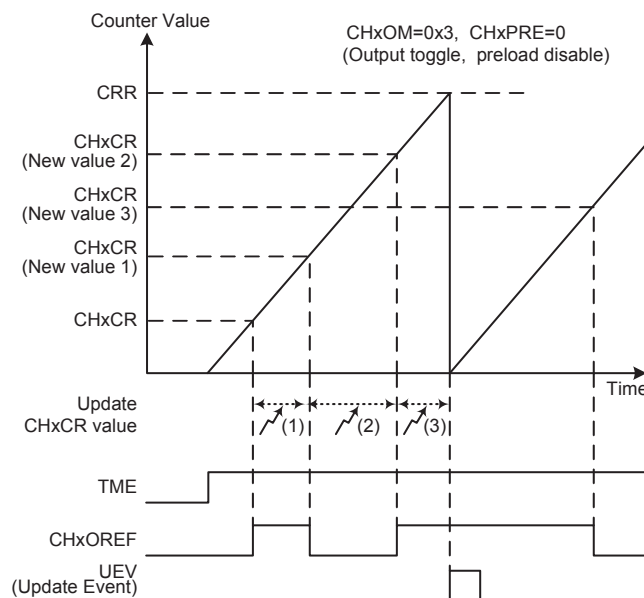


Figure 76. Toggle Mode Channel Output Reference Signal (CHxPRE = 0)

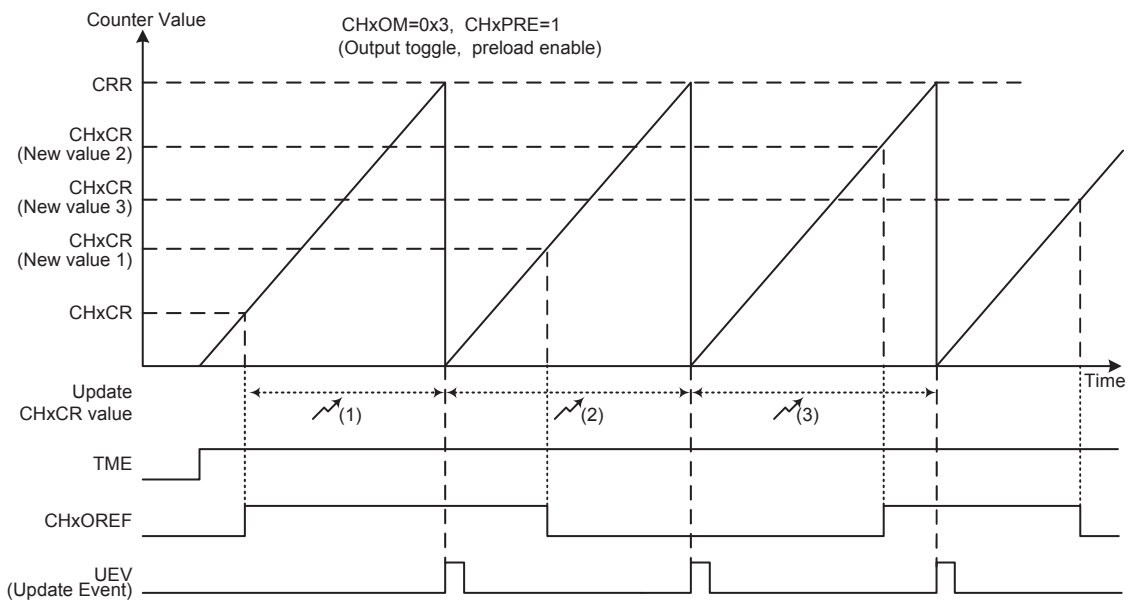


Figure 77. Toggle Mode Channel Output Reference Signal (CHxPRE = 1)

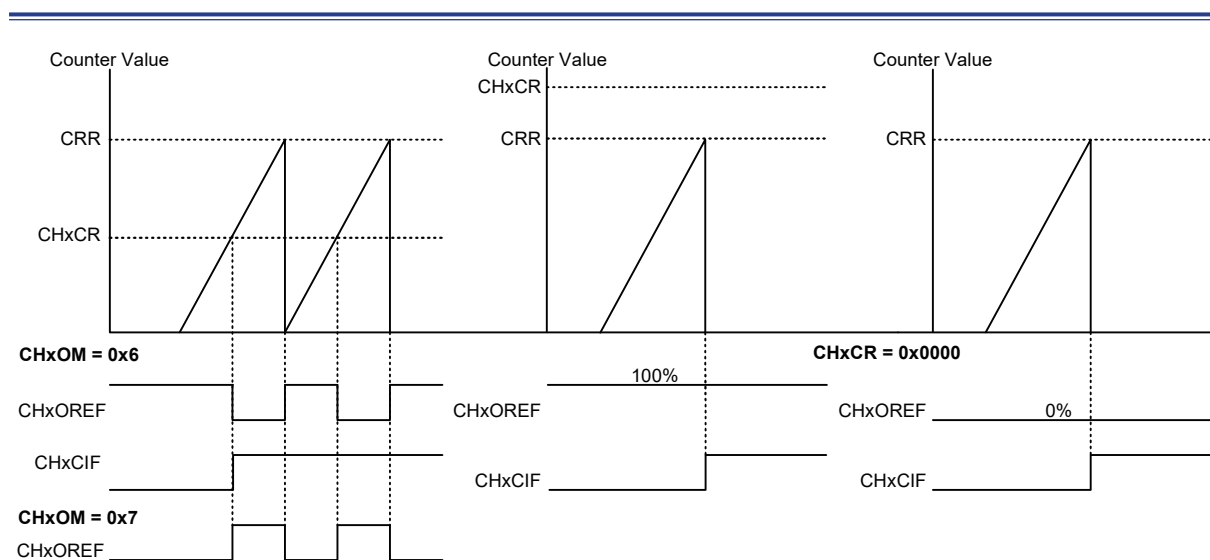


Figure 78. PWM Mode Channel Output Reference Signal and Counter in Up-counting Mode

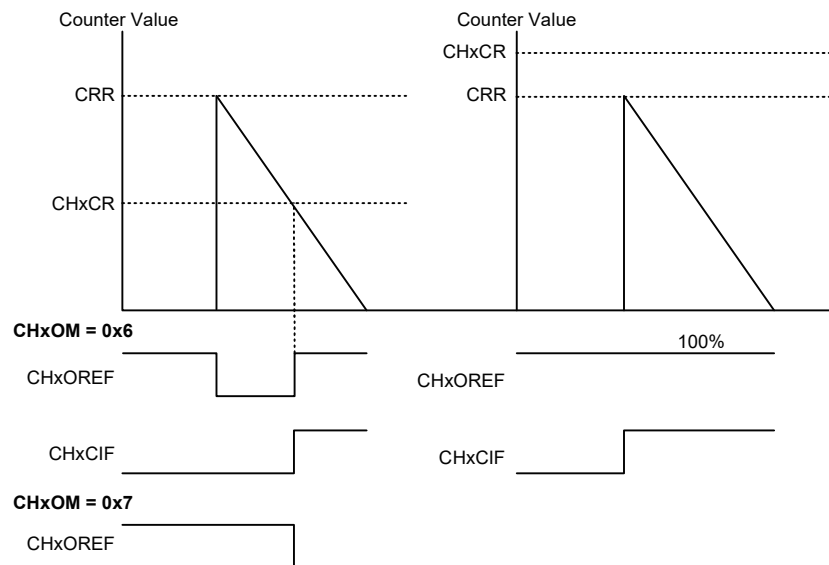


Figure 79. PWM Mode Channel Output Reference Signal and Counter in Down-counting Mode

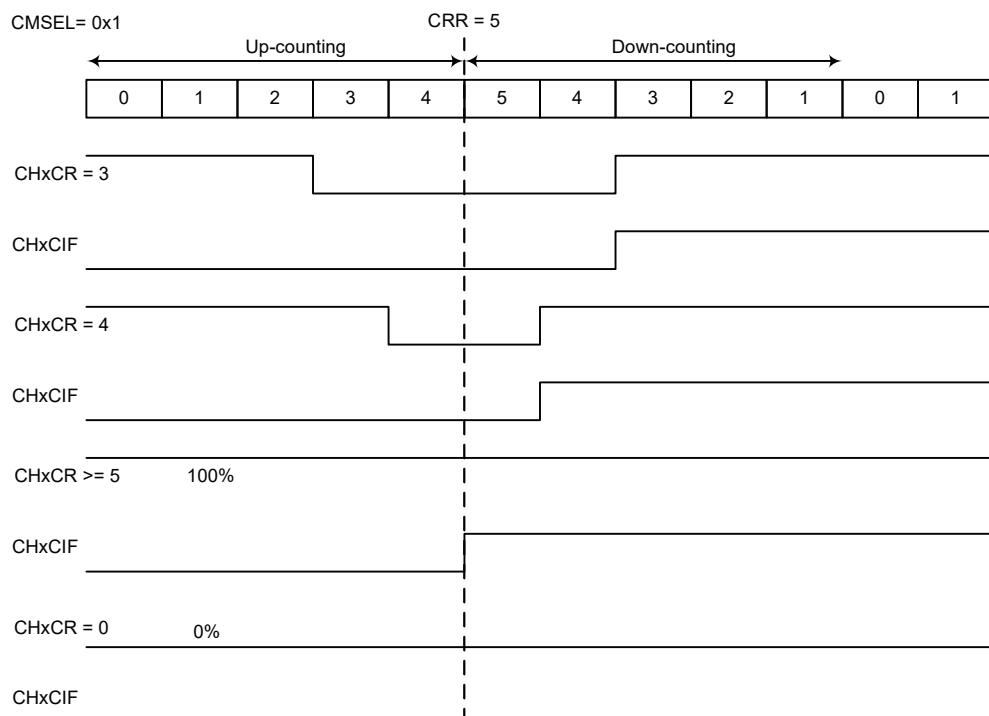


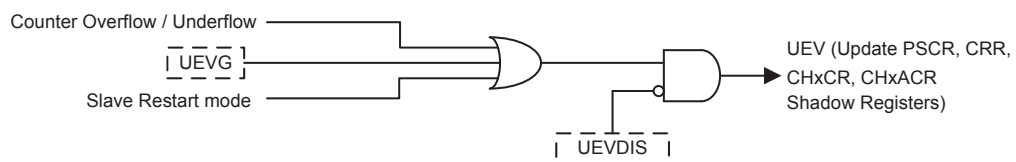
Figure 80. PWM Mode Channel Output Reference Signal and Counter in Center-aligned Mode

Update Management

The Update event is used to update the CRR, the PSCR, the CHxACR and the CHxCR values from the actual registers to the corresponding shadow registers. An update event occurs when the counter overflows or underflows, the software update control bit is triggered or an update event from the slave controller is generated.

The UEVDIS bit in the CNTCFR register can determine whether the update event occurs or not. When the update event occurs, the corresponding update event interrupt will be generated depending upon whether the update event interrupt generation function is enabled or not by configuring the UGDIS bit in the CNTCFR register. For more detailed description, refer to the UEVDIS and UGDIS bit definition in the CNTCFR register.

Update Event Management



Update Event Interrupt Management

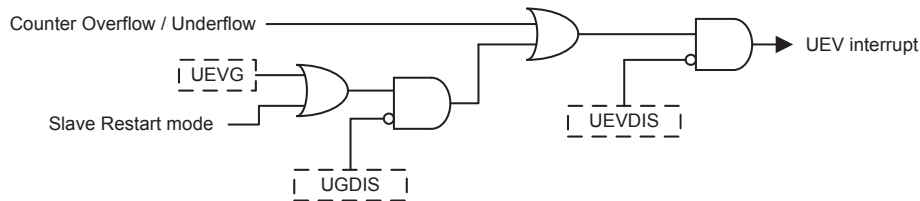


Figure 81. Update Event Setting Diagram

Single Pulse Mode

Once the timer is set to operate in the single pulse mode, it is not necessary to set the timer enable bit TME in the CTR register to 1 to enable the counter. The trigger to generate a pulse can be sourced from the STI signal rising edge or by setting the TME bit to 1 using software. Setting the TME bit to 1 or a trigger from the STI signal rising edge can generate a pulse and then keep the TME bit at a high state until the update event occurs or the TME bit is written to 0 by software. If the TME bit is cleared to 0 using software, the counter will be stopped and its value held. If the TME bit is automatically cleared to 0 by a hardware update event, the counter will be reinitialized.

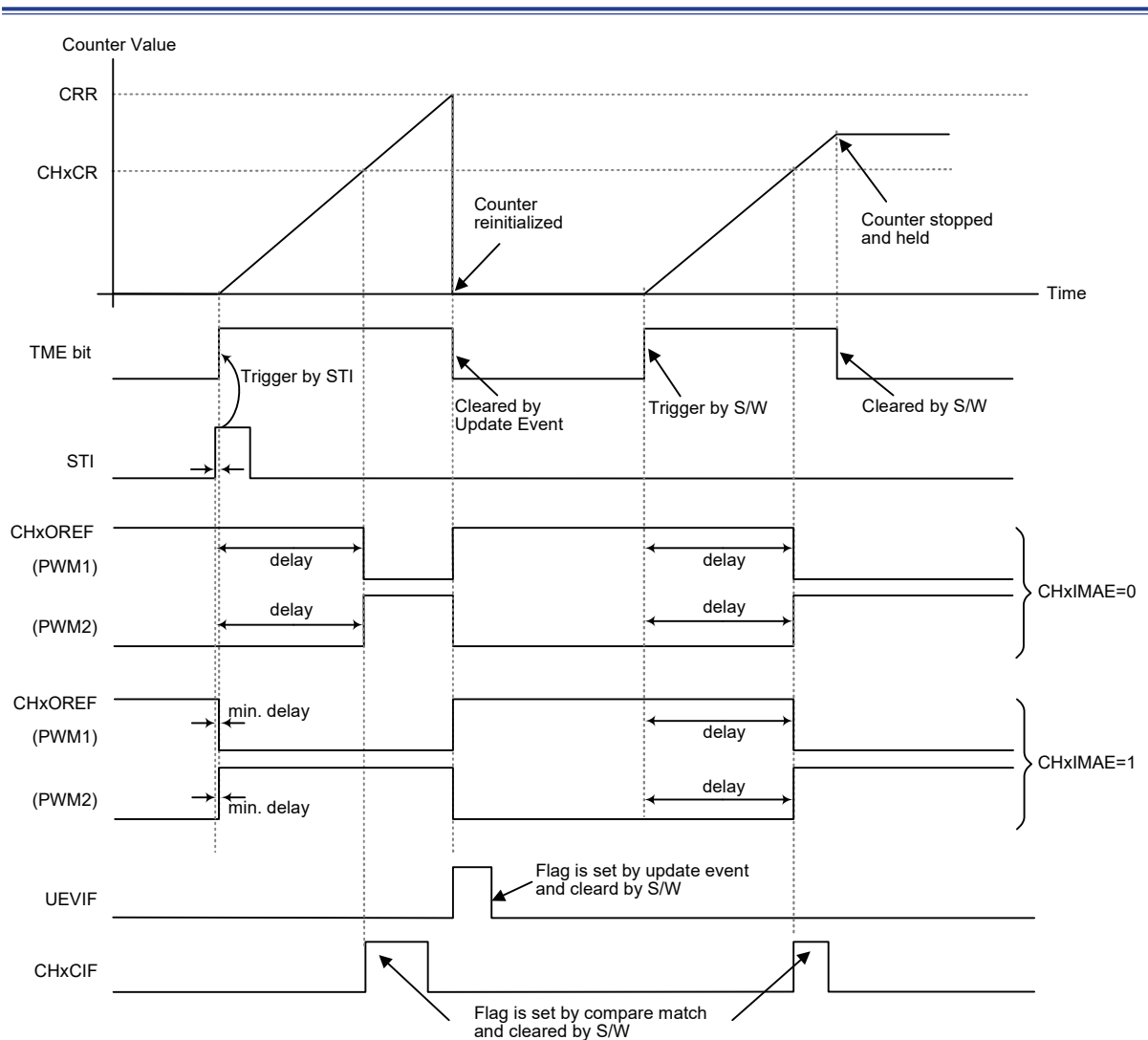


Figure 82. Single Pulse Mode

In the Single Pulse mode, the STI active edge which sets the TME bit to 1 will enable the counter. However, there exist several clock delays to perform the comparison result between the counter value and the CHxCR value. In order to reduce the delay to a minimum value, the user can set the CHxIMAE bit in each CHxOCFR register. After a STI rising edge trigger occurs in the single pulse mode, the CHxOREF signal will immediately be forced to the state which the CHxOREF signal will change to as the compare match event occurs without taking the comparison result into account. The CHxIMAE bit is available only when the output channel is configured to operate in the PWM mode 1 or PWM mode 2 and the trigger source is derived from the STI signal.

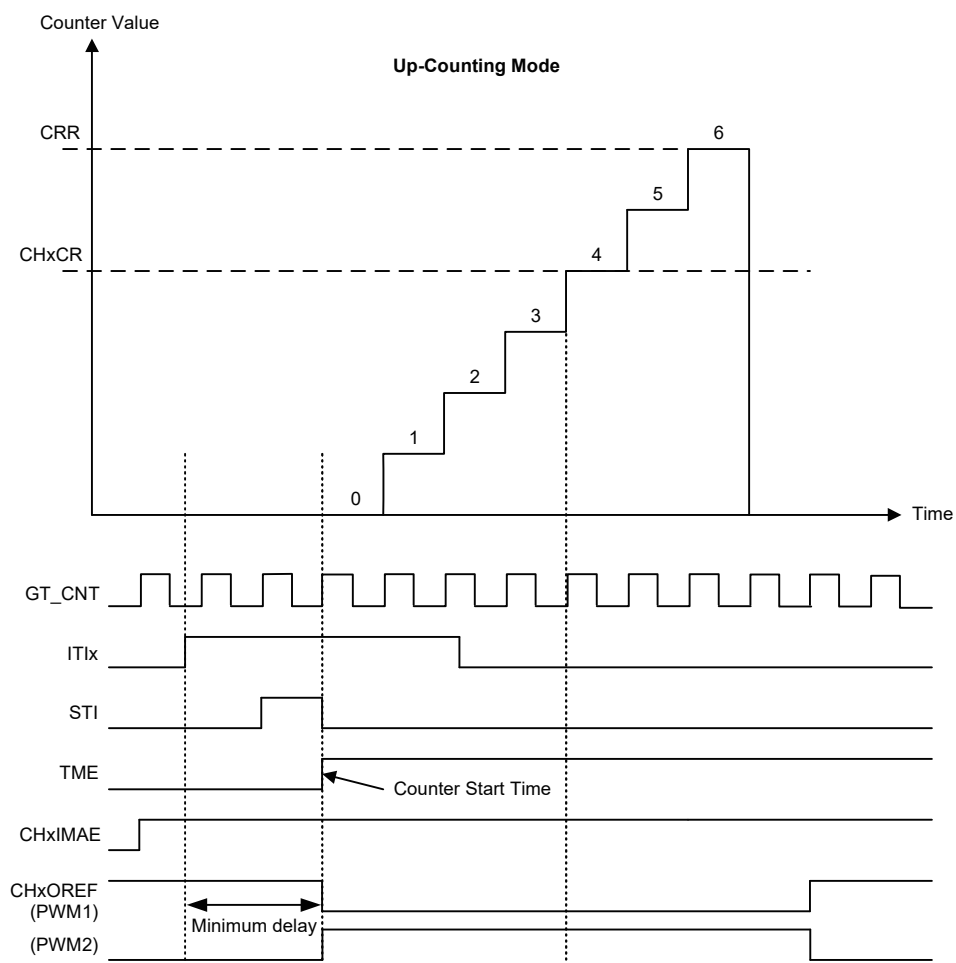


Figure 83. Immediate Active Mode Minimum Delay

Asymmetric PWM Mode

Asymmetric PWM mode allows two center-aligned PWM signals to be generated with a programmable phase shift. While the PWM frequency is determined by the value of the CRR register, the duty cycle and the phase-shift are determined by the CHxCR and CHxACR register. When the counter is counting up, the PWM uses the value in CHxCR as up-count compare value. When the counter is into counting down stage, the PWM uses the value in CHxACR as down-count compare value. The Figure 84 is shown an example for asymmetric PWM mode in center-aligned counting mode.

Note: Asymmetric PWM mode can only be operated in center-aligned counting mode.

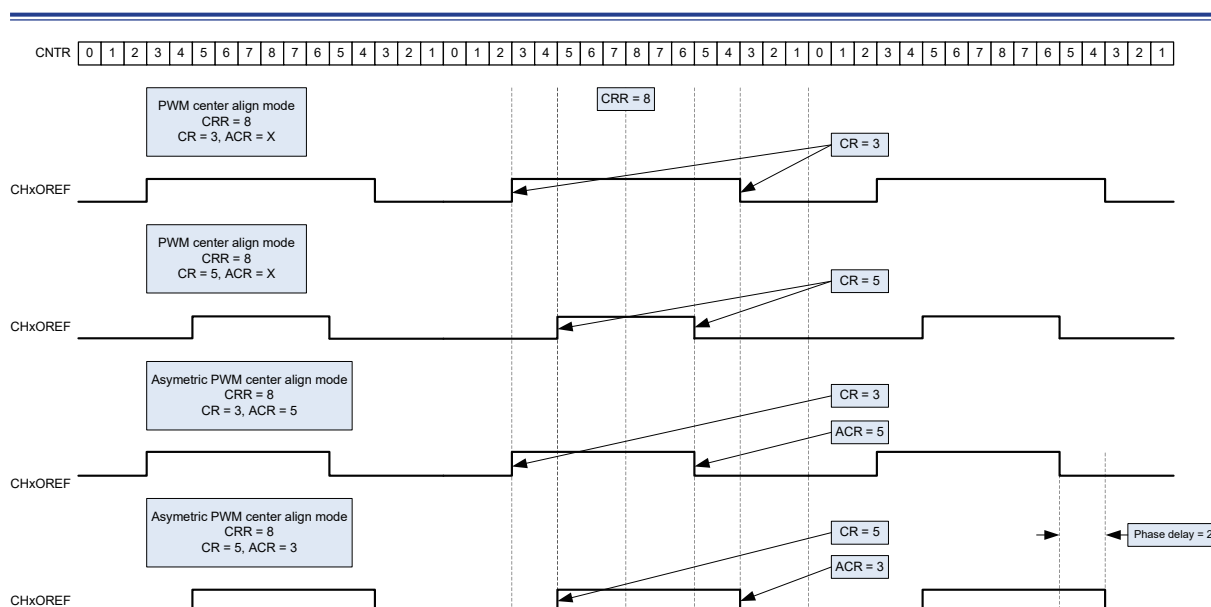


Figure 84. Asymmetric PWM Mode versus Center-aligned Counting Mode

Timer Interconnection

The timers can be internally connected together for timer chaining or synchronization. This can be implemented by configuring one timer to operate in the Master mode while configuring another timer to be in the Slave mode. The following figures present several examples of trigger selection for the master and slave modes.

Using one timer to enable/disable another timer start or stop counting

- Configure PWM0 as the master mode to send its channel 0 Output Reference signal CH0OREF as a trigger output (MMSEL = 0x4).
- Configure PWM0 CH0OREF waveform.
- Configure PWM1 to receive its input trigger source from the PWM0 trigger output (TRSEL = 0x9).
- Configure PWM1 to operate in the pause mode (SMSEL = 0x5).
- Enable PWM1 by writing '1' to the TME bit.
- Enable PWM0 by writing '1' to the TME bit.

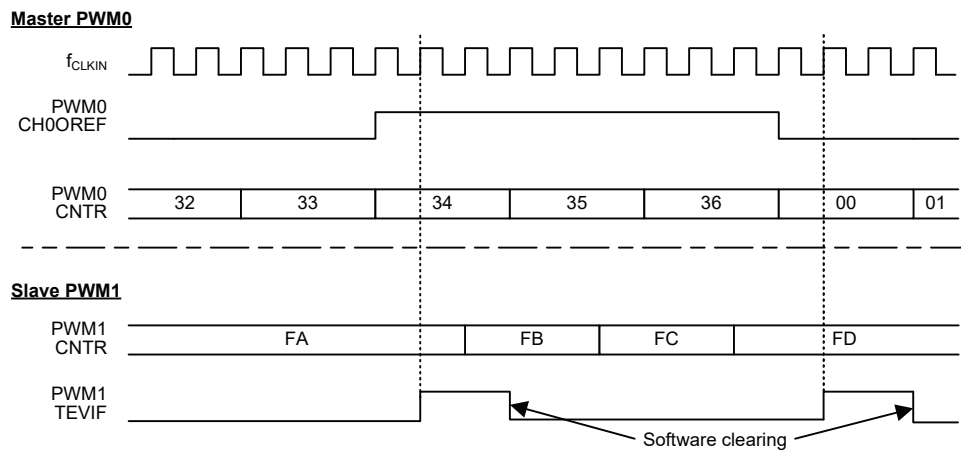


Figure 85. Pausing PWM1 using the PWM0 CH0OREF Signal

Using one timer to trigger another timer start counting

- Configure PWM0 to operate in the master mode to send its Update Event UEV as the trigger output (MMSEL = 0x2).
- Configure the PWM0 period by setting the CRR register.
- Configure PWM1 to get the input trigger source from the PWM0 trigger output (TRSEL = 0x9).
- Configure PWM1 to be in the slave trigger mode (SMSEL = 0x6).
- Start PWM0 by writing '1' to the TME bit.

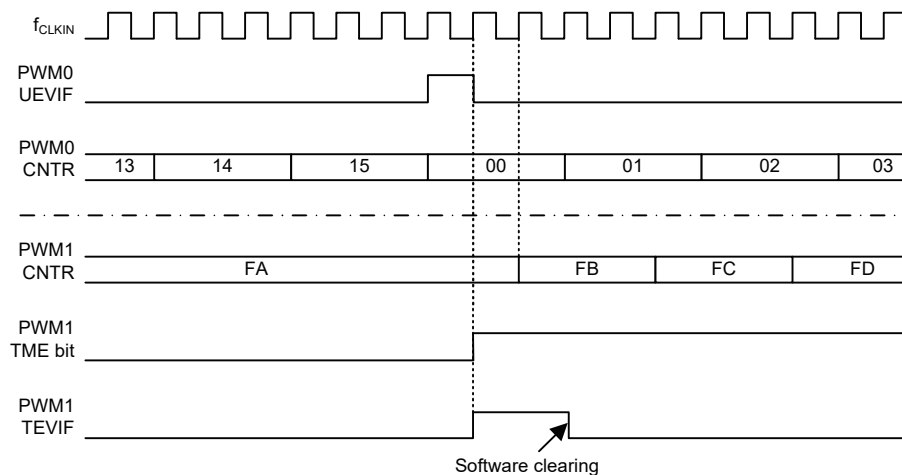
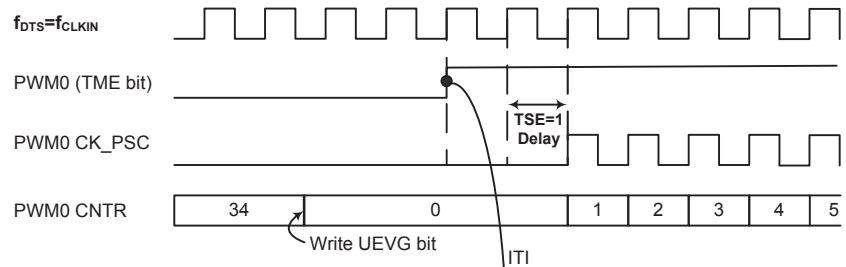


Figure 86. Triggering PWM1 with PWM0 Update Event

Starting two timers synchronously with the master enable MTO signal trigger

- Configure PWM0 to operate in the master mode to send its enable signal as a trigger output (MMSEL = 0x1).
- Enable the PWM0 master timer synchronization function by setting the TSE bit in the MDCFR register to 1 to synchronize the slave timer.
- Configure PWM1 to receive its input trigger source from the PWM0 trigger output (TRSEL = 0x9).
- Configure PWM1 to be in the slave trigger mode (SMSEL = 0x6).
- Start PWM0 by writing '1' to the TME bit.

Master PWM0



Slave PWM1

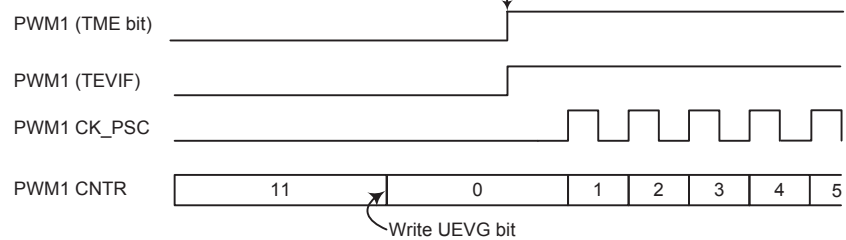


Figure 87. Trigger PWM0 and PWM1 with the PWM0 Timer Enable Signal

Trigger Peripherals Start

To interconnect to the peripherals ex. ADC, DAC and Timer and so on, the PWM could be output the MTO signal or the channel compare match output signal CHxOREF (x = 0 ~ 3) to be used as peripherals input trigger signal and depend on the MCU specification.

PDMA Request

The PWM supports the interface for PDMA data transfer. There are certain events which can generate the PDMA requests if the corresponding enable control bits are set to 1 to enable the PDMA access. These events are the PWM update events, trigger events and channel compare events. When the PDMA request is generated from the PWM channel, it can be derived from the channel compare event or the PWM update event selected by the channel PDMA selection bit, CHCDS, for all channels. For more detailed PDMA configuring information, refer to the corresponding section in the PDMA chapter.

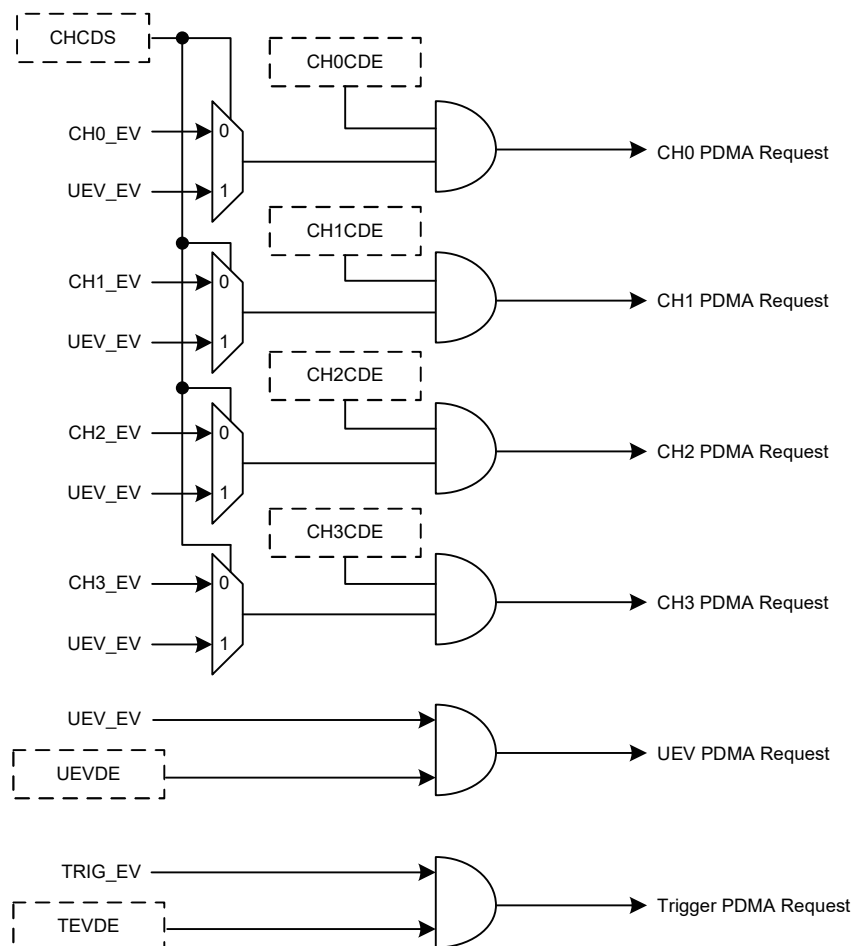


Figure 88. PWM PDMA Mapping Diagram

Register Map

The following table shows the PWM registers and reset values.

Table 34. PWM Register Map

Register	Offset	Description	Reset Value
CNTCFR	0x000	Timer Counter Configuration Register	0x0000_0000
MDCFR	0x004	Timer Mode Configuration Register	0x0000_0000
TRCFR	0x008	Timer Trigger Configuration Register	0x0000_0000
CTR	0x010	Timer Control Register	0x0000_0000
CH0OCFR	0x040	Channel 0 Output Configuration Register	0x0000_0000
CH1OCFR	0x044	Channel 1 Output Configuration Register	0x0000_0000
CH2OCFR	0x048	Channel 2 Output Configuration Register	0x0000_0000
CH3OCFR	0x04C	Channel 3 Output Configuration Register	0x0000_0000
CHCTR	0x050	Channel Control Register	0x0000_0000
CHPOLR	0x054	Channel Polarity Configuration Register	0x0000_0000
DICTR	0x074	Timer PDMA / Interrupt Control Register	0x0000_0000
EVGR	0x078	Timer Event Generator Register	0x0000_0000
INTSR	0x07C	Timer Interrupt Status Register	0x0000_0000
CNTR	0x080	Timer Counter Register	0x0000_0000
PSCR	0x084	Timer Prescaler Register	0x0000_0000
CRR	0x088	Timer Counter-Reload Register	0x0000_FFFF
CH0CR	0x090	Channel 0 Compare Register	0x0000_0000
CH1CR	0x094	Channel 1 Compare Register	0x0000_0000
CH2CR	0x098	Channel 2 Compare Register	0x0000_0000
CH3CR	0x09C	Channel 3 Compare Register	0x0000_0000
CH0ACR	0x0A0	Channel 0 Asymmetric Compare Register	0x0000_0000
CH1ACR	0x0A4	Channel 1 Asymmetric Compare Register	0x0000_0000
CH2ACR	0x0A8	Channel 2 Asymmetric Compare Register	0x0000_0000
CH3ACR	0x0AC	Channel 3 Asymmetric Compare Register	0x0000_0000

Register Descriptions

Timer Counter Configuration Register – CNTCFR

This register specifies the PWM counter configuration.

Offset: 0x000

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							DIR
Type/Reset								RW 0
	23	22	21	20	19	18	17	16
	Reserved						CMSEL	
Type/Reset							RW 0	RW 0
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved						UGDIS	UEVDIS
Type/Reset							RW 0	RW 0

Bits	Field	Descriptions
[24]	DIR	Counting Direction 0: Count-up 1: Count-down Note: This bit is read only when the Timer is configured to be in the Center-aligned mode.
[17:16]	CMSEL	Counter Mode Selection 00: Edge-aligned mode. Normal up-counting and down-counting available for this mode. Counting direction is defined by the DIR bit. 01: Center-aligned mode 1. The counter counts up and down alternatively. The compare match interrupt flag is set during the count-down period. 10: Center-aligned mode 2. The counter counts up and down alternatively. The compare match interrupt flag is set during the count-up period. 11: Center-aligned mode 3. The counter counts up and down alternatively. The compare match interrupt flag is set during the count-up and count-down periods.
[1]	UGDIS	Update event interrupt generation disable control 0: Any of the following events will generate an update PDMA request or interrupt - Counter overflow/underflow - Setting the UEVG bit - Update generation through the slave mode 1: Only counter overflow/underflow generates an update PDMA request or interrupt

Bits	Field	Descriptions
[0]	UEVDIS	Update event Disable control 0: Enable the update event request by one of following events: - Counter overflow/underflow - Setting the UEVG bit - Update generation through the slave mode 1: Disable the update event (However the counter and the prescaler are reinitialized if the UEVG bit is set or if a hardware restart is received from the slave mode)

Timer Mode Configuration Register – MDCFR

This register specifies the PWM master and slave mode selection and single pulse mode.

Offset: 0x004

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
	Reserved							SPMSET	
Type/Reset								RW	0
	23	22	21	20	19	18	17	16	
	Reserved					MMSEL			
Type/Reset						RW	0	RW	0
	15	14	13	12	11	10	9	8	
	Reserved					SMSEL			
Type/Reset						RW	0	RW	0
	7	6	5	4	3	2	1	0	
	Reserved							TSE	
Type/Reset								RW	0

Bits	Field	Descriptions
[24]	SPMSET	Single Pulse Mode Setting 0: Counter counts normally irrespective of whether the update event occurred or not. 1: Counter stops counting at the next update event and then the TME bit is cleared by hardware.

Bits	Field	Descriptions																											
[18:16]	MMSEL	<p>Master Mode Selection</p> <p>Master mode selection is used to select the MTO signal source which is used to synchronize the other slave timer.</p> <table> <tr> <th>MMSEL [2:0]</th><th>Mode</th><th>Descriptions</th></tr> <tr> <td>000</td><td>Reset Mode</td><td>The MTO signal in the Reset mode is an output derived from one of the following cases: 1. Software setting UEVG bit 2. The STI trigger input signal which will be output on the MTO signal line when the Timer is used in the slave Restart mode</td></tr> <tr> <td>001</td><td>Enable Mode</td><td>The Counter Enable signal is used as the trigger output.</td></tr> <tr> <td>010</td><td>Update Mode</td><td>The update event is used as the trigger output according to one of the following cases when the UEVDIS bit is cleared to 0: 1. Counter overflow / underflow 2. Software setting UEVG 3. Slave trigger input when used in slave restart mode</td></tr> <tr> <td>011</td><td>—</td><td>Reserved</td></tr> <tr> <td>100</td><td>Compare Mode 0</td><td>The Channel 0 Output reference signal named CH0OREF is used as the trigger output.</td></tr> <tr> <td>101</td><td>Compare Mode 1</td><td>The Channel 1 Output reference signal named CH1OREF is used as the trigger output.</td></tr> <tr> <td>110</td><td>Compare Mode 2</td><td>The Channel 2 Output reference signal named CH2OREF is used as the trigger output.</td></tr> <tr> <td>111</td><td>Compare Mode 3</td><td>The Channel 3 Output reference signal named CH3OREF is used as the trigger output.</td></tr> </table>	MMSEL [2:0]	Mode	Descriptions	000	Reset Mode	The MTO signal in the Reset mode is an output derived from one of the following cases: 1. Software setting UEVG bit 2. The STI trigger input signal which will be output on the MTO signal line when the Timer is used in the slave Restart mode	001	Enable Mode	The Counter Enable signal is used as the trigger output.	010	Update Mode	The update event is used as the trigger output according to one of the following cases when the UEVDIS bit is cleared to 0: 1. Counter overflow / underflow 2. Software setting UEVG 3. Slave trigger input when used in slave restart mode	011	—	Reserved	100	Compare Mode 0	The Channel 0 Output reference signal named CH0OREF is used as the trigger output.	101	Compare Mode 1	The Channel 1 Output reference signal named CH1OREF is used as the trigger output.	110	Compare Mode 2	The Channel 2 Output reference signal named CH2OREF is used as the trigger output.	111	Compare Mode 3	The Channel 3 Output reference signal named CH3OREF is used as the trigger output.
MMSEL [2:0]	Mode	Descriptions																											
000	Reset Mode	The MTO signal in the Reset mode is an output derived from one of the following cases: 1. Software setting UEVG bit 2. The STI trigger input signal which will be output on the MTO signal line when the Timer is used in the slave Restart mode																											
001	Enable Mode	The Counter Enable signal is used as the trigger output.																											
010	Update Mode	The update event is used as the trigger output according to one of the following cases when the UEVDIS bit is cleared to 0: 1. Counter overflow / underflow 2. Software setting UEVG 3. Slave trigger input when used in slave restart mode																											
011	—	Reserved																											
100	Compare Mode 0	The Channel 0 Output reference signal named CH0OREF is used as the trigger output.																											
101	Compare Mode 1	The Channel 1 Output reference signal named CH1OREF is used as the trigger output.																											
110	Compare Mode 2	The Channel 2 Output reference signal named CH2OREF is used as the trigger output.																											
111	Compare Mode 3	The Channel 3 Output reference signal named CH3OREF is used as the trigger output.																											

Bits	Field	Descriptions																											
[11:8]	SMSEL	Slave Mode Selection																											
		<table><tr><th>SMSEL [2:0]</th><th>Mode</th><th>Descriptions</th></tr><tr><td>000</td><td>Disable mode</td><td>The prescaler is clocked directly by the internal clock.</td></tr><tr><td>001</td><td>—</td><td>Reserved</td></tr><tr><td>010</td><td>—</td><td>Reserved</td></tr><tr><td>011</td><td>—</td><td>Reserved</td></tr><tr><td>100</td><td>Restart Mode</td><td>The counter value restarts from 0 or the CRR shadow register value depending upon the counter mode on the rising edge of the STI signal. The registers will also be updated.</td></tr><tr><td>101</td><td>Pause Mode</td><td>The counter starts to count when the selected trigger input STI is high. The counter stops counting on the instant, not being reset, when the STI signal changes its state to a low level. Both the counter start and stop control are determined by the STI signal.</td></tr><tr><td>110</td><td>Trigger Mode</td><td>The counter starts to count from the original value in the counter on the rising edge of the selected trigger input STI. Only the counter start control is determined by the STI signal.</td></tr><tr><td>111</td><td>STIED</td><td>The rising edge of the selected trigger signal STI will clock the counter.</td></tr></table>	SMSEL [2:0]	Mode	Descriptions	000	Disable mode	The prescaler is clocked directly by the internal clock.	001	—	Reserved	010	—	Reserved	011	—	Reserved	100	Restart Mode	The counter value restarts from 0 or the CRR shadow register value depending upon the counter mode on the rising edge of the STI signal. The registers will also be updated.	101	Pause Mode	The counter starts to count when the selected trigger input STI is high. The counter stops counting on the instant, not being reset, when the STI signal changes its state to a low level. Both the counter start and stop control are determined by the STI signal.	110	Trigger Mode	The counter starts to count from the original value in the counter on the rising edge of the selected trigger input STI. Only the counter start control is determined by the STI signal.	111	STIED	The rising edge of the selected trigger signal STI will clock the counter.
		SMSEL [2:0]	Mode	Descriptions																									
		000	Disable mode	The prescaler is clocked directly by the internal clock.																									
		001	—	Reserved																									
		010	—	Reserved																									
		011	—	Reserved																									
		100	Restart Mode	The counter value restarts from 0 or the CRR shadow register value depending upon the counter mode on the rising edge of the STI signal. The registers will also be updated.																									
		101	Pause Mode	The counter starts to count when the selected trigger input STI is high. The counter stops counting on the instant, not being reset, when the STI signal changes its state to a low level. Both the counter start and stop control are determined by the STI signal.																									
		110	Trigger Mode	The counter starts to count from the original value in the counter on the rising edge of the selected trigger input STI. Only the counter start control is determined by the STI signal.																									
111	STIED	The rising edge of the selected trigger signal STI will clock the counter.																											
[0]	TSE	Timer Synchronization Enable 0: No action 1: Master timer (current timer) will generate a delay to synchronize its slave timer through the MTO signal.																											

Timer Trigger Configuration Register – TRCFR

This register specifies the trigger source selection of PWM.

Offset: 0x008

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved				TRSEL			
					RW	0	RW	0

Bits	Field	Descriptions
[3:0]	TRSEL	<p>Trigger Source Selection</p> <p>These bits are used to select the trigger input (STI) for counter synchronization.</p> <p>0000: Software Trigger by setting the UEVG bit</p> <p>1001: Internal Timing Module Trigger 0 (ITI0)</p> <p>1010: Internal Timing Module Trigger 1 (ITI1)</p> <p>1011: Internal Timing Module Trigger 2 (ITI2)</p> <p>Others: Reserved</p> <p>Note: These bits must be updated only when they are not in use, i.e. the slave mode is disabled by setting the SMSEL field to 0x0.</p>

Table 35. PWM Internal Trigger Connection

Slave Timing Module	ITI0	ITI1	ITI2
PWM0	PWM1	GPTM	—
PWM1	PWM0	GPTM	—

Timer Counter Register – CTR

This register specifies the timer enable bit (TME), CRR buffer enable bit (CRBE) and Channel PDMA selection bit (CHCDS).

Offset: 0x010

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							CHCDS
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved						CRBE	TME
							RW	0
							RW	0

Bits	Field	Descriptions
[16]	CHCDS	Channel PDMA event selection 0: Channel PDMA request derived from the channel compare event. 1: Channel PDMA request derived from the Update event.
[1]	CRBE	Counter-Reload register Buffer Enable 0: Counter-reload register can be updated immediately 1: Counter-reload register can not be updated until the update event occurs
[0]	TME	Timer Enable bit 0: PWM off 1: PWM on When the TME bit is cleared to 0, the counter is stopped and the PWM consumes no power in any operation mode except for the single pulse mode and the slave trigger mode. In these two modes the TME bit can automatically be set to 1 by hardware which permits all the PWM registers to function normally.

Channel 0 Output Configuration Register – CH0OCFR

This register specifies the channel 0 output mode configuration.

Offset: 0x040

Reset value: 0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
Type/Reset							
23	22	21	20	19	18	17	16
Reserved							
Type/Reset							
15	14	13	12	11	10	9	8
Reserved							CH0OM[3]
Type/Reset							RW 0
7	6	5	4	3	2	1	0
Reserved		CH0IMAE	CH0PRE	Reserved	CH0OM[2:0]		
Type/Reset	RW 0		RW 0		RW 0	RW 0	RW 0

Bits	Field	Descriptions
[5]	CH0IMAE	Channel 0 Immediate Active Enable 0: No action 1: Single pulse Immediate Active Mode is enabled The CH0OREF signal will be forced to the compare matched level immediately after an available trigger event occurs irrespective of the result of the comparison between the CNTR and the CH0CR values. The effective duration ends automatically at the next overflow or underflow event. Note: The CH0IMAE bit is available only if the channel 0 is configured to be operated in the PWM mode 1 or PWM mode 2.
[4]	CH0PRE	Channel 0 Compare Register (CH0CR) Preload Enable 0: CH0CR preload function is disabled The CH0CR register can be immediately assigned a new value when the CH0PRE bit is cleared to 0 and the updated CH0CR value is used immediately. 1: CH0CR preload function is enabled The new CH0CR value will not be transferred to its shadow register until the update event occurs.

Bits	Field	Descriptions
[8][2:0]	CH0OM[3:0]	<p>Channel 0 Output Mode Setting</p> <p>These bits define the functional types of the output reference signal CH0OREF.</p> <p>0000: No Change</p> <p>0001: Output 0 on compare match</p> <p>0010: Output 1 on compare match</p> <p>0011: Output toggles on compare match</p> <p>0100: Force inactive – CH0OREF is forced to 0</p> <p>0101: Force active – CH0OREF is forced to 1</p> <p>0110: PWM mode 1</p> <ul style="list-style-type: none"> - During up-counting, channel 0 has an active level when CNTR < CH0CR or otherwise has an inactive level. - During down-counting, channel 0 has an inactive level when CNTR > CH0CR or otherwise has an active level. <p>0111: PWM mode 2</p> <ul style="list-style-type: none"> - During up-counting, channel 0 is has an inactive level when CNTR < CH0CR or otherwise has an active level. - During down-counting, channel 0 has an active level when CNTR > CH0CR or otherwise has an inactive level. <p>1110: Asymmetric PWM mode 1</p> <ul style="list-style-type: none"> - During up-counting, channel 0 has an active level when CNTR < CH0CR or otherwise has an inactive level. - During down-counting, channel 0 has an inactive level when CNTR > CH0ACR or otherwise has an active level. <p>1111: Asymmetric PWM mode 2</p> <ul style="list-style-type: none"> - During up-counting, channel 0 has an inactive level when CNTR < CH0CR or otherwise has an active level. - During down-counting, channel 0 has an active level when CNTR > CH0ACR or otherwise has an inactive level. <p>Note: When channel 0 is used as asymmetric PWM output mode, the Counter Mode Selection bit in Counter Configuration Register must be configured as Center-aligned Counting mode (CMSEL = 0x1/0x2/0x3)</p>

Channel 1 Output Configuration Register – CH1OCFR

This register specifies the channel 1 output mode configuration.

Offset: 0x044

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							CH1OM[3]
								RW 0
	7	6	5	4	3	2	1	0
Type/Reset	Reserved		CH1IMAE	CH1PRE	Reserved	CH1OM[2:0]		
			RW 0	RW 0		RW 0	RW 0	RW 0

Bits	Field	Descriptions
[5]	CH1IMAE	Channel 1 Immediate Active Enable 0: No action 1: Single pulse Immediate Active Mode is enabled The CH1OREF signal will be forced to the compare matched level immediately after an available trigger event occurs irrespective of the result of the comparison between the CNTR and the CH1CR values. The effective duration ends automatically at the next overflow or underflow event. Note: The CH1IMAE bit is available only if the channel 1 is configured to be operated in the PWM mode 1 or PWM mode 2.
[4]	CH1PRE	Channel 1 Compare Register (CH1CR) Preload Enable 0: CH1CR preload function is disabled. The CH1CR register can be immediately assigned a new value when the CH1PRE bit is cleared to 0 and the updated CH1CR value is used immediately. 1: CH1CR preload function is enabled The new CH1CR value will not be transferred to its shadow register until the update event occurs.

Bits	Field	Descriptions
[8][2:0]	CH1OM[3:0]	<p>Channel 1 Output Mode Setting</p> <p>These bits define the functional types of the output reference signal CH1OREF.</p> <p>0000: No Change</p> <p>0001: Output 0 on compare match</p> <p>0010: Output 1 on compare match</p> <p>0011: Output toggles on compare match</p> <p>0100: Force inactive – CH1OREF is forced to 0</p> <p>0101: Force active – CH1OREF is forced to 1</p> <p>0110: PWM mode 1</p> <ul style="list-style-type: none"> - During up-counting, channel 1 has an active level when CNTR < CH1CR or otherwise has an inactive level. - During down-counting, channel 1 has an inactive level when CNTR > CH1CR or otherwise has an active level. <p>0111: PWM mode 2</p> <ul style="list-style-type: none"> - During up-counting, channel 1 has an inactive level when CNTR < CH1CR or otherwise has an active level. - During down-counting, channel 1 has an active level when CNTR > CH1CR or otherwise has an inactive level. <p>1110: Asymmetric PWM mode 1</p> <ul style="list-style-type: none"> - During up-counting, channel 1 has an active level when CNTR < CH1CR or otherwise has an inactive level. - During down-counting, channel 1 has an inactive level when CNTR > CH1CR or otherwise has an active level. <p>1111: Asymmetric PWM mode 2</p> <ul style="list-style-type: none"> - During up-counting, channel 1 has an inactive level when CNTR < CH1CR or otherwise has an active level. - During down-counting, channel 1 has an active level when CNTR > CH1CR or otherwise has an inactive level. <p>Note: When channel 1 is used as asymmetric PWM output mode, the Counter Mode Selection bit in Counter Configuration Register must be configured as Center-aligned Counting mode (CMSEL = 0x1/0x2/0x3)</p>

Channel 2 Output Configuration Register – CH2OCFR

This register specifies the channel 2 output mode configuration.

Offset: 0x048

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							CH2OM[3]
	7	6	5	4	3	2	1	0
Type/Reset	Reserved		CH2IMAE	CH2PRE	Reserved		CH2OM[2:0]	
			RW	0	RW	0	RW	0

Bits	Field	Descriptions
[5]	CH2IMAE	Channel 2 Immediate Active Enable 0: No action 1: Single pulse Immediate Active Mode is enabled The CH2OREF signal will be forced to the compare matched level immediately after an available trigger event occurs irrespective of the result of the comparison between the CNTR and the CH2CR values. The effective duration ends automatically at the next overflow or underflow event. Note: The CH2IMAE bit is available only if the channel 2 is configured to be operated in the PWM mode 1 or PWM mode 2.
[4]	CH2PRE	Channel 2 Compare Register (CH2CR) Preload Enable 0: CH2CR preload function is disabled. The CH2CR register can be immediately assigned a new value when the CH2PRE bit is cleared to 0 and the updated CH2CR value is used immediately. 1: CH2CR preload function is enabled The new CH2CR value will not be transferred to its shadow register until the update event occurs.

Bits	Field	Descriptions
[8][2:0]	CH2OM[3:0]	<p>Channel 2 Output Mode Setting</p> <p>These bits define the functional types of the output reference signal CH2OREF.</p> <p>0000: No Change</p> <p>0001: Output 0 on compare match</p> <p>0010: Output 1 on compare match</p> <p>0011: Output toggles on compare match</p> <p>0100: Force inactive – CH2OREF is forced to 0</p> <p>0101: Force active – CH2OREF is forced to 1</p> <p>0110: PWM mode 1</p> <ul style="list-style-type: none"> - During up-counting, channel 2 has an active level when CNTR < CH2CR or otherwise has an inactive level. - During down-counting, channel 2 has an inactive level when CNTR > CH2CR or otherwise has an active level. <p>0111: PWM mode 2</p> <ul style="list-style-type: none"> - During up-counting, channel 2 has an inactive level when CNTR < CH2CR or otherwise has an active level. - During down-counting, channel 2 has an active level when CNTR > CH2CR or otherwise has an inactive level. <p>1110: Asymmetric PWM mode 1</p> <ul style="list-style-type: none"> - During up-counting, channel 2 has an active level when CNTR < CH2CR or otherwise has an inactive level. - During down-counting, channel 2 has an inactive level when CNTR > CH2CR or otherwise has an active level. <p>1111: Asymmetric PWM mode 2</p> <ul style="list-style-type: none"> - During up-counting, channel 2 has an inactive level when CNTR < CH2CR or otherwise has an active level. - During down-counting, channel 2 has an active level when CNTR > CH2CR or otherwise has an inactive level. <p>Note: When channel 2 is used as asymmetric PWM output mode, the Counter Mode Selection bit in Counter Configuration Register must be configured as Center-aligned Counting mode (CMSEL = 0x1/0x2/0x3)</p>

Channel 3 Output Configuration Register – CH3OCFR

This register specifies the channel 3 output mode configuration.

Offset: 0x04C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	Reserved							CH3OM[3]	
									RW 0
	7	6	5	4	3	2	1	0	
Type/Reset	Reserved		CH3IMAE	CH3PRE	Reserved	CH3OM[2:0]			
			RW 0	RW 0		RW 0	RW 0	RW 0	

Bits	Field	Descriptions
[5]	CH3IMAE	Channel 3 Immediate Active Enable 0: No action 1: Single pulse Immediate Active Mode is enabled The CH3OREF signal will be forced to the compare matched level immediately after an available trigger event occurs irrespective of the result of the comparison between the CNTR and the CH3CR values. The effective duration ends automatically at the next overflow or underflow event. Note: The CH3IMAE bit is available only if the channel 3 is configured to be operated in the PWM mode 1 or PWM mode 2.
[4]	CH3PRE	Channel 3 Compare Register (CH3CR) Preload Enable 0: CH3CR preload function is disabled. The CH3CR register can be immediately assigned a new value when the CH3PRE bit is cleared to 0 and the updated CH3CR value is used immediately. 1: CH3CR preload function is enabled The new CH3CR value will not be transferred to its shadow register until the update event occurs.

Bits	Field	Descriptions
[8][2:0]	CH3OM[3:0]	<p>Channel 3 Output Mode Setting</p> <p>These bits define the functional types of the output reference signal CH3OREF</p> <p>0000: No Change</p> <p>0001: Output 0 on compare match</p> <p>0010: Output 1 on compare match</p> <p>0011: Output toggles on compare match</p> <p>0100: Force inactive – CH3OREF is forced to 0</p> <p>0101: Force active – CH3OREF is forced to 1</p> <p>0110: PWM mode 1</p> <ul style="list-style-type: none"> - During up-counting, channel 3 has an active level when CNTR < CH3CR or otherwise has an inactive level. - During down-counting, channel 3 has an inactive level when CNTR > CH3CR or otherwise has an active level. <p>0111: PWM mode 2</p> <ul style="list-style-type: none"> - During up-counting, channel 3 has an inactive level when CNTR < CH3CR or otherwise has an active level. - During down-counting, channel 3 has an active level when CNTR > CH3CR or otherwise has an inactive level <p>1110: Asymmetric PWM mode 1</p> <ul style="list-style-type: none"> - During up-counting, channel 3 has an active level when CNTR < CH3CR or otherwise has an inactive level. - During down-counting, channel 3 has an inactive level when CNTR > CH3CR or otherwise has an active level. <p>1111: Asymmetric PWM mode 2</p> <ul style="list-style-type: none"> - During up-counting, channel 3 has an inactive level when CNTR < CH3CR or otherwise has an active level. - During down-counting, channel 3 has an active level when CNTR > CH3CR or otherwise has an inactive level <p>Note: When channel 3 is used as asymmetric PWM output mode, the Counter Mode Selection bit in Counter Configuration Register must be configured as Center-aligned Counting mode (CMSEL = 0x1/0x2/0x3)</p>

Channel Control Register – CHCTR

This register contains the channel compare output function enable control bits.

Offset: 0x050

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved	CH3E	Reserved	CH2E	Reserved	CH1E	Reserved	CH0E
	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[6]	CH3E	Channel 3 Compare Enable 0: Off – Channel 3 output signal CH3O is not active 1: On – Channel 3 output signal CH3O is generated on the corresponding output pin
[4]	CH2E	Channel 2 Capture/Compare Enable 0: Off – Channel 2 output signal CH2O is not active 1: On – Channel 2 output signal CH2O is generated on the corresponding output pin
[2]	CH1E	Channel 1 Capture/Compare Enable 0: Off – Channel 1 output signal CH1O is not active 1: On – Channel 1 output signal CH1O is generated on the corresponding output pin
[0]	CH0E	Channel 0 Capture/Compare Enable 0: Off – Channel 0 output signal CH0O is not active 1: On – Channel 0 output signal CH0O is generated on the corresponding output pin

Channel Polarity Configuration Register – CHPOLR

This register contains the channel compare output polarity control.

Offset: 0x054

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved	CH3P	Reserved	CH2P	Reserved	CH1P	Reserved	CH0P
	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[6]	CH3P	Channel 3 Compare Polarity 0: Channel 3 Output is active high 1: Channel 3 Output is active low
[4]	CH2P	Channel 2 Compare Polarity 0: Channel 2 Output is active high 1: Channel 2 Output is active low
[2]	CH1P	Channel 1 Compare Polarity 0: Channel 1 Output is active high 1: Channel 1 Output is active low
[0]	CH0P	Channel 0 Compare Polarity 0: Channel 0 Output is active high 1: Channel 0 Output is active low

Timer PDMA/Interrupt Control Register – DICTR

This register contains the timer PDMA and interrupt enable control bits.

Offset: 0x074

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved					TEVDE	Reserved	UEVDE
Type/Reset						RW 0		RW 0
	23	22	21	20	19	18	17	16
	Reserved				CH3CDE	CH2CDE	CH1CDE	CH0CDE
Type/Reset					RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	Reserved					TEVIE	Reserved	UEVIE
Type/Reset						RW 0		RW 0
	7	6	5	4	3	2	1	0
	Reserved				CH3CIE	CH2CIE	CH1CIE	CH0CIE
Type/Reset					RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[26]	TEVDE	Trigger event PDMA Request Enable 0: Trigger PDMA request is disabled 1: Trigger PDMA request is enabled
[24]	UEVDE	Update event PDMA Request Enable 0: Update event PDMA request is disabled 1: Update event PDMA request is enabled
[19]	CH3CDE	Channel 3 Compare PDMA Request Enable 0: Channel 3 PDMA request is disabled 1: Channel 3 PDMA request is enabled
[18]	CH2CDE	Channel 2 Compare PDMA Request Enable 0: Channel 2 PDMA request is disabled 1: Channel 2 PDMA request is enabled
[17]	CH1CDE	Channel 1 Compare PDMA Request Enable 0: Channel 1 PDMA request is disabled 1: Channel 1 PDMA request is enabled
[16]	CH0CDE	Channel 0 Compare PDMA Request Enable 0: Channel 0 PDMA request is disabled 1: Channel 0 PDMA request is enabled
[10]	TEVIE	Trigger event Interrupt Enable 0: Trigger event interrupt is disabled 1: Trigger event interrupt is enabled
[8]	UEVIE	Update event Interrupt Enable 0: Update event interrupt is disabled 1: Update event interrupt is enabled
[3]	CH3CIE	Channel 3 Compare Interrupt Enable 0: Channel 3 interrupt is disabled 1: Channel 3 interrupt is enabled
[2]	CH2CIE	Channel 2 Compare Interrupt Enable 0: Channel 2 interrupt is disabled 1: Channel 2 interrupt is enabled

Bits	Field	Descriptions
[1]	CH1CIE	Channel 1 Compare Interrupt Enable 0: Channel 1 interrupt is disabled 1: Channel 1 interrupt is enabled
[0]	CH0CIE	Channel 0 Compare Interrupt Enable 0: Channel 0 interrupt is disabled 1: Channel 0 interrupt is enabled

Timer Event Generator Register – EVGR

This register contains the software event generation bits.

Offset: 0x078

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved					TEVG	Reserved	UEVG
						WO 0		WO 0
	7	6	5	4	3	2	1	0
Type/Reset	Reserved				CH3CG	CH2CG	CH1CG	CH0CG
					WO 0	WO 0	WO 0	WO 0

Bits	Field	Descriptions
[10]	TEVG	Trigger Event Generation The trigger event TEV can be generated by setting this bit. It is cleared by hardware automatically. 0: No action 1: TEVIF flag is set
[8]	UEVG	Update Event Generation The update event UEV can be generated by setting this bit. It is cleared by hardware automatically. 0: No action 1: Reinitialize the counter The counter value returns to 0 or the CRR preload value, depending on the counter mode in which the current timer is being used. An update operation of any related registers will also be performed. For more detailed descriptions, refer to the corresponding section.
[3]	CH3CG	Channel 3 Compare Generation A Channel 3 compare event can be generated by software setting this bit. It is cleared by hardware automatically. 0: No action 1: Compare event is generated on channel 3

Bits	Field	Descriptions
[2]	CH2CG	Channel 2 Compare Generation A Channel 2 compare event can be generated by software setting this bit. It is cleared by hardware automatically. 0: No action 1: Compare event is generated on channel 2
[1]	CH1CG	Channel 1 Compare Generation A Channel 1 compare event can be generated by software setting this bit. It is cleared by hardware automatically. 0: No action 1: Compare event is generated on channel 1
[0]	CH0CG	Channel 0 Compare Generation A Channel 0 compare event can be generated by software setting this bit. It is cleared by hardware automatically. 0: No action 1: Compare event is generated on channel 0

Timer Interrupt Status Register – INTSR

This register stores the timer interrupt status.

Offset: 0x07C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved					TEVIF	Reserved	UEVIF
						W0C	0	W0C
	7	6	5	4	3	2	1	0
Type/Reset	Reserved				CH3CIF	CH2CIF	CH1CIF	CH0CIF
					W0C	0	W0C	0

Bits	Field	Descriptions
[10]	TEVIF	Trigger Event Interrupt Flag This flag is set by hardware on a trigger event and is cleared by software. 0: No trigger event occurs 1: Trigger event occurs

Bits	Field	Descriptions
[8]	UEVIF	<p>Update Event Interrupt Flag.</p> <p>This bit is set by hardware on an update event and is cleared by software.</p> <p>0: No update event occurs 1: Update event occurs</p> <p>Note: The update event is derived from the following conditions:</p> <ul style="list-style-type: none"> - The counter overflows or underflows - The UEVG bit is asserted - A restart trigger event occurs from the slave trigger input
[3]	CH3CIF	<p>Channel 3 Compare Interrupt Flag</p> <p>0: No match event occurs 1: The content of the counter CNTR has matched the contents of the CH3CR register.</p> <p>This flag is set by hardware when the counter value matches the CH3CR value except in the center-aligned mode. It is cleared by software.</p>
[2]	CH2CIF	<p>Channel 2 Compare Interrupt Flag</p> <p>0: No match event occurs 1: The content of the counter CNTR has matched the contents of the CH2CR register</p> <p>This flag is set by hardware when the counter value matches the CH2CR value except in the center-aligned mode. It is cleared by software.</p>
[1]	CH1CIF	<p>Channel 1 Compare Interrupt Flag</p> <p>0: No match event occurs 1: The content of the counter CNTR has matched the contents of the CH1CR register</p> <p>This flag is set by hardware when the counter value matches the CH1CR value except in the center-aligned mode. It is cleared by software.</p>
[0]	CH0CIF	<p>Channel 0 Compare Interrupt Flag</p> <p>0: No match event occurs 1: The content of the counter CNTR has matched the content of the CH0CR register</p> <p>This flag is set by hardware when the counter value matches the CH0CR value except in the center-aligned mode. It is cleared by software.</p>

Timer Counter Register – CNTR

This register stores the timer counter value.

Offset: 0x080

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	CNTV								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	CNTV								
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:0]	CNTV	Counter Value.

Timer Prescaler Register – PSCR

This register specifies the timer prescaler value to generate the counter clock.

Offset: 0x084

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	PSCV								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	PSCV								
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:0]	PSCV	Prescaler Value These bits are used to specify the prescaler value to generate the counter clock frequency f_{CK_CNT} . $f_{CK_CNT} = \frac{f_{CK_PSC}}{PSCV[15:0]+1}$, where the f_{CK_PSC} is the prescaler clock source.

Timer Counter-Reload Register – CRR

This register specifies the timer counter-reload value.

Offset: 0x088

Reset value: 0x0000_FFFF

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	CRV							
	7	6	5	4	3	2	1	0
Type/Reset	CRV							
	RW	1	RW	1	RW	1	RW	1
	RW	1	RW	1	RW	1	RW	1

Bits	Field	Descriptions
[15:0]	CRV	Counter-Reload Value The CRV is the reload value which is loaded into the actual counter register.

Channel 0 Compare Register – CH0CR

This register specifies the timer channel 0 compare value.

Offset: 0x090

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	CH0CV							
	7	6	5	4	3	2	1	0
Type/Reset	CH0CV							
	RW	0	RW	0	RW	0	RW	0
	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[15:0]	CH0CV	Channel 0 Compare Value The CH0CR value is compared with the counter value and the comparison result is used to trigger the CH0OREF output signal.

Channel 1 Compare Register – CH1CR

This register specifies the timer channel 1 compare value.

Offset: 0x094

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	CH1CV								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	CH1CV								
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:0]	CH1CV	Channel 1 Compare Value The CH1CR value is compared with the counter value and the comparison result is used to trigger the CH1OREF output signal.

Channel 2 Compare Register – CH2CR

This register specifies the timer channel 2 capture/compare value.

Offset: 0x098

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	CH2CV								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	CH2CV								
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:0]	CH2CV	Channel 2 Compare Value The CH2CR value is compared with the counter value and the comparison result is used to trigger the CH2OREF output signal.

Channel 3 Compare Register – CH3CR

This register specifies the timer channel 3 compare value.

Offset: 0x09C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	CH3CV								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	CH3CV								
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:0]	CH3CV	Channel 3 Compare Value The CH3CR value is compared with the counter value and the comparison result is used to trigger the CH3OREF output signal.

Channel 0 Asymmetric Compare Register – CH0ACR

This register specifies the timer channel 0 asymmetric compare value.

Offset: 0x0A0

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	CH0ACV								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	CH0ACV								
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:0]	CH0ACV	Channel 0 Asymmetric Compare Value When channel 0 is configured as asymmetric PWM mode and the counter is counting down, the value written in this register will be compared to the counter.

Channel 1 Asymmetric Compare Register – CH1ACR

This register specifies the timer channel 1 asymmetric compare value.

Offset: 0x0A4

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	CH1ACV								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	CH1ACV								
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:0]	CH1ACV	Channel 1 Asymmetric Compare Value When channel 1 is configured as asymmetric PWM mode and the counter is counting down, the value written in this register will be compared to the counter.

Channel 2 Asymmetric Compare Register – CH2ACR

This register specifies the timer channel 2 asymmetric compare value.

Offset: 0x0A8

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	CH2ACV								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	CH2ACV								
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:0]	CH2ACV	Channel 2 Asymmetric Compare Value When channel 2 is configured as asymmetric PWM mode and the counter is counting down, the value written in this register will be compared to the counter.

Channel 3 Asymmetric Compare Register – CH3ACR

This register specifies the timer channel 3 asymmetric compare value.

Offset: 0x0AC

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	CH3ACV								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	CH3ACV								
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:0]	CH3ACV	Channel 3 Asymmetric Compare Value When channel 3 is configured as asymmetric PWM mode and the counter is counting down, the value written in this register will be compared to the counter.

14 Basic Function Timer (BFTM)

Introduction

The Basic Function Timer Module, BFTM, is a 32-bit up-counting counter designed to measure time intervals, generate one shot pulses or generate repetitive interrupts. The BFTM can operate in two modes which are repetitive and one shot modes. The repetitive mode restarts the counter at each compare match event which is generated by the internal comparator. The BFTM also supports a one shot mode which will force the counter to stop counting when a compare match event occurs.

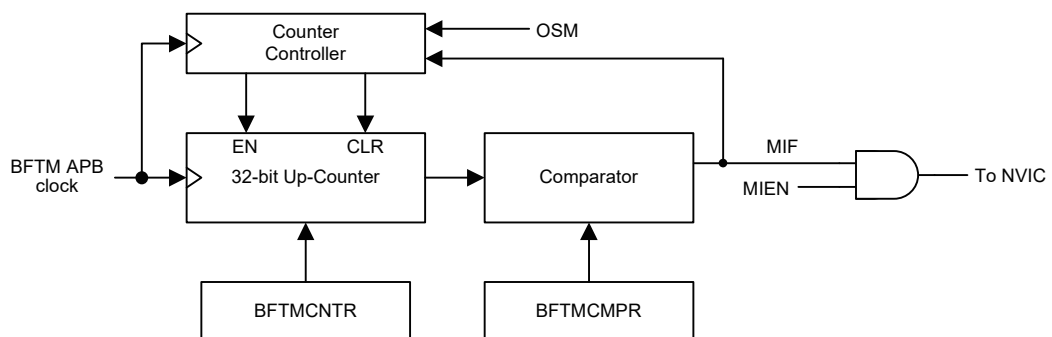


Figure 89. BFTM Block Diagram

Features

- 32-bit up-counting counter
- Compare Match function
- Includes debug mode
- Clock source: BFTM APB clock
- Counter value can be Read/Written on the fly
- One shot mode: counter stops counting when compare match occurs
- Repetitive mode: counter restarts when compare match occurs
- Compare Match interrupt enable/disable control

Functional Description

The BFTM is a 32-bit up-counting counter which is driven by the BFTM APB clock, PCLK. The counter value can be changed or read at any time even when the timer is counting. The BFTM supports two operating modes known as the repetitive mode and one shot mode allowing the measurement of time intervals or the generation of periodic time durations.

Repetitive Mode

The BFTM counts up from zero to a specific compare value which is pre-defined by the BFTMCMPR register. When the BFTM operates in the repetitive mode and the counter reaches a value equal to the specific compare value in the BFTMCMPR register, the timer will generate a compare match event signal, MIF. When this occurs, the counter will be reset to 0 and resume its counting operation. When the MIF signal is generated, a BFTM compare match interrupt will also be generated periodically if the compare match interrupt is enabled by setting the corresponding interrupt control bit, MIEN, to 1. The counter value will remain unchanged and the counter will stop counting if it is disabled by clearing the CEN bit to 0.

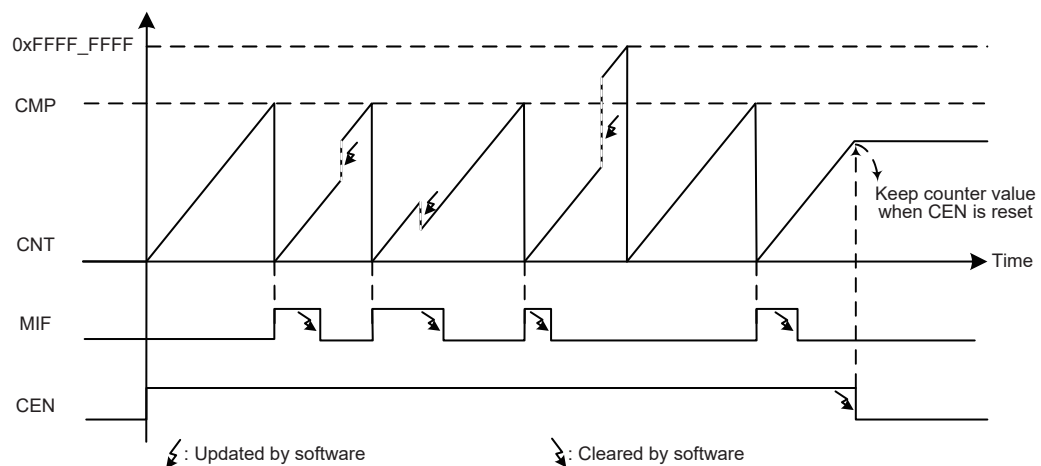


Figure 90. BFTM – Repetitive Mode

One Shot Mode

By setting the OSM bit in BFTMCR register to 1, the BFTM will operate in the one shot mode. The BFTM starts to count when the CEN bit is set to 1 by the application program. The counter value will remain unchanged if the CEN bit is cleared to 0 by the application program. However, the counter value will be reset to 0 and stop counting when the CEN bit is cleared automatically to 0 by the internal hardware when a counter compare match event occurs.

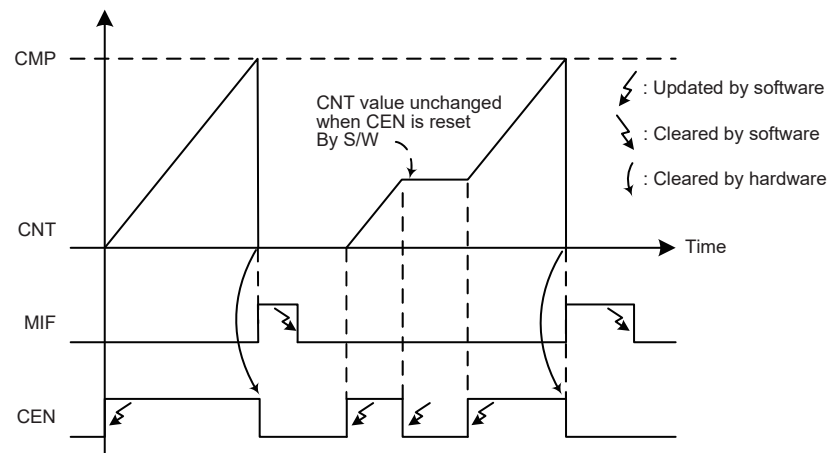


Figure 91. BFTM – One Shot Mode

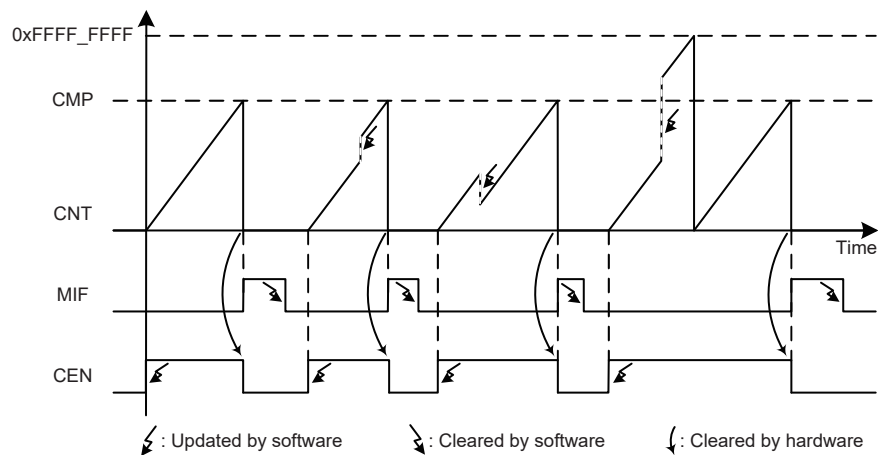


Figure 92. BFTM – One Shot Mode Counter Updating

Register Map

The following table shows the BFTM registers and reset values.

Table 36. BFTM Register Map

Register	Offset	Description	Reset Value
BFTMCR	0x000	BFTM Control Register	0x0000_0000
BFTMSR	0x004	BFTM Status Register	0x0000_0000
BFTMCNTR	0x008	BFTM Counter Value Register	0x0000_0000
BFTMCMR	0x00C	BFTM Compare Value Register	0xFFFF_FFFF

Register Descriptions

BFTM Control Register – BFTMCR

This register specifies the overall BFTM control bits.

Offset: 0x000

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved					CEN	OSM	MIEN
						RW	0	RW
							0	RW
								0

Bits	Field	Descriptions
[2]	CEN	BFTM Counter Enable Control 0: BFTM is disabled 1: BFTM is enabled When this bit is set to 1, the BFTM counter will start to count. The counter will stop counting and the counter value will remain unchanged when the CEN bit is cleared to 0 by the application program regardless of whether it is in the repetitive or one shot mode. However, in the one shot mode, the counter will stop counting and be reset to 0 when the CEN bit is cleared to 0 by the timer hardware circuitry which results from a compare match event.
[1]	OSM	BFTM One Shot Mode Selection 0: Counter operates in repetitive mode 1: Counter operates in one shot mode
[0]	MIEN	BFTM Compare Match Interrupt Enable Control 0: Compare Match Interrupt is disabled 1: Compare Match Interrupt is enabled

BFTM Status Register – BFTMSR

This register specifies the BFTM status.

Offset: 0x004

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved							MIF
								W0C 0

Bits	Field	Descriptions
[0]	MIF	<p>BFTM Compare Match Interrupt Flag</p> <p>0: No compare match event occurs</p> <p>1: Compare match event occurs</p> <p>When the counter value, CNT, is equal to the compare register value, CMP, a compare match event will occur and the corresponding interrupt flag, MIF will be set. The MIF bit is cleared to 0 by writing a data “0”.</p>

BFTM Counter Value Register – BFTMCNTR

This register specifies the BFTM counter value.

Offset: 0x008

Reset value: 0x0000_0000

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

Bits	Field	Descriptions
[31:0]	CNT	BFTM Counter Value A 32-bit BFTM counter value is stored in this field which can be read or written on the fly.

BFTM Compare Value Register – BFTMCMPR

The register specifies the BFTM compare value.

Offset: 0x00C

Reset value: 0xFFFF_FFFF

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

Bits	Field	Descriptions
[31:0]	CMP	BFTM Compare Value This register specifies a 32-bit BFTM compare value which is used for comparison with the BFTM counter value.

15 Real Time Clock (RTC)

Introduction

The Real Time Clock, RTC, circuitry includes the APB interface, a 24-bit up-counter, a control register, a prescaler, a compare register and a status register. Most of the RTC circuits are located in the V_{DD} Power Domain, as shown in dotted red box in the accompanying figure, except for the APB interface. The APB interface is located in the V_{DD15} domain. Therefore, it is necessary to be isolated from the ISO signal that comes from the power control unit when the V_{DD15} domain is powered off, i.e., when the device enters the Power-Down mode. The RTC counter is used as a wakeup timer to let the system resume from the Power-Down mode. The detailed RTC function will be described in the following sections.

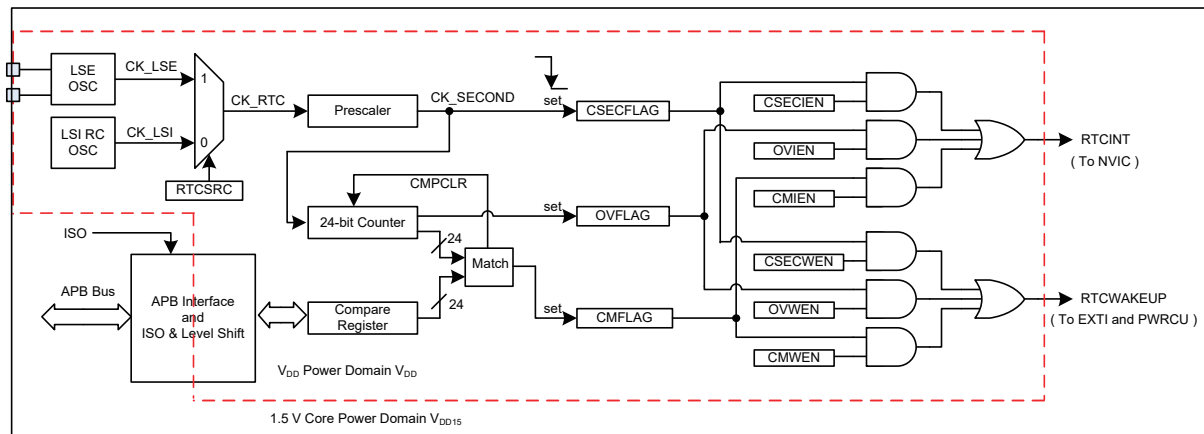


Figure 93. RTC Block Diagram

Features

- 24-bit up-counter for counting elapsed time
- Programmable clock prescaler
 - Division factor: 1, 2, 4, 8..., 32768
- 24-bit compare register for alarm usage
- RTC clock source
 - LSE oscillator clock
 - LSI oscillator clock
- Three RTC Interrupt/wakeup settings
 - RTC second clock interrupt/wakeup
 - RTC compare match interrupt/wakeup
 - RTC counter overflow interrupt/wakeup
- The RTC interrupt/wakeup event can work together with power management to wake up the chip from power saving mode

Functional Descriptions

RTC Related Register Reset

The RTC registers can only be reset by either a V_{DD} Domain power on reset, POR, or by a V_{DD} Domain software reset by setting the PWRST bit in the PWRCCR register. Other reset events have no effect to clear the RTC registers.

Reading RTC Register

The RTC control logic and the related registers are powered by the V_{DD} supply voltage. Therefore, the RTC circuitry remains operational in the Power-Down mode where V_{DD15} is powered off. Only the APB bus, which is located in the V_{DD15} domain, is interconnected to the RTC circuits located in the V_{DD} domain using level shift circuitry and isolated by the ISO signals when the V_{DD15} supply voltage is powered off.

Low Speed Clock Configuration

The default RTC clock source, CK_RTC, is derived from the LSI oscillator. The CK_RTC clock can be derived from either the external 32768 Hz crystal oscillator, named the LSE oscillator, or the internal 32K RC oscillator named the LSI oscillator, by setting the RTCSRC bit in the RTCCR register. A prescaler is provided to divide the CK_RTC by a ratio ranged from 2^0 to 2^{15} determined by the RPRE [3:0] field. For instance, setting the prescaler value RPRE [3:0] to 0xF will generate an exact 1 Hz CK_SECOND clock if the CK_RTC clock frequency is equal to 32,768 Hz. The LSI and LSE oscillators can be enabled by the LSIEN and LSEEN control bits in the RTCCR register respectively. In addition, the LSE oscillator startup mode can be selected by configuring the LSESM bit in the RTCCR register. This enables the LSE oscillator to have either a shorter startup time or a lower power consumption, both of which are traded off depending upon specific application requirements. An example of the startup time and the power consumption for different startup modes are shown in the accompanying table for reference.

Table 37. LSE Startup Mode Operating Current and Startup Time

Startup Mode	LSECM Setting in the RTCCR Register	Operating Current	Startup Time
Normal startup	0	2.0 μ A	Above 500 ms
Fast startup	1	3.5 μ A	Below 300 ms

@ $V_{DD} = 3.3$ V and LSE clock = 32,768 Hz; these values are only for reference, actual values are dependent on the specification of the external 32.768KHz crystal.

RTC Counter Operation

The RTC provides a 24-bit up-counter which increases at the falling edge of the CK_SECOND clock and whose value can be read from the RTCCNT register asynchronously via the APB bus. A 24-bit compare register, RTCCMP, is provided to store the specific value to be compared with the RTCCNT content. This is used to define a pre-determined time interval. When the RTCCNT register content is equal to the RTCCMP register value, the match flag CMFLAG in the RTCSR register will be set by hardware and an interrupt or wakeup event can be sent according to the corresponding enable bits in the RTCIWEN register. The RTC counter will be either reset to zero or keep counting when the compare match event occurs, dependent upon the CMPCLR bit in the RTCCR register. For example, if the RPRE [3:0] is set to 0xF, the RTCCMP register content is set to a decimal value of 60 and the CMPCLR bit is set to 1, then the CMFLAG bit will be set every minute. In addition, the OVFLAG bit in the RTCSR register will be set when the RTC counter overflows. A read operation on the RTCSR register clears the status flags including the CSECFLAG, CMFLAG and OVFLAG bits.

Interrupt and Wakeup Control

The falling edge of the CK_SECOND clock causes the CSECFLAG bit in the RTCSR register to be set and generates an interrupt if the corresponding interrupt enable bit, CSECIEN, in the RTCIWEN register is set. The wakeup event can also be generated to wake up the HSI/HSE oscillators, the PLL circuitry, the LDO and the CPU core if the corresponding wakeup enable bit CSECWEN is set. When the RTC counter overflows or a compare match event occurs, it will generate an interrupt or a wake up event determined by the corresponding interrupt or wakeup enable control bits, OVIEN/OVWEN or CMIEN/CMWEN bits, in the RTCIWEN register. Refer to the related register definitions for more details.

RTCOUNT Output Pin Configuration

The following table shows the RTCOUT output format according to the mode, polarity, and event selection setting.

Table 38. RTCOUT Output Mode and Active Level Setting

ROWM	ROES	RTCOUT Output Waveform	
0 (Pulse mode)	0 (Compare match)	RTCCMP	----- 4 -----
		RTCCNT	----- 3 ----- 4 ----- 5 -----
		RTCOUT (ROAP = 0)	----- T_R -----
		RTCOUT (ROAP = 1)	----- T_R -----
		ROLF	-----
	1 (Second clock)	RTCCMP	----- X -----
		RTCCNT	----- 3 ----- 4 ----- 5 -----
		RTCOUT (ROAP = 0)	----- T_R ----- T_R ----- T_R -----
		RTCOUT (ROAP = 1)	----- T_R ----- T_R ----- T_R -----
		ROLF	-----
1 (Level mode)	0 (Compare match)	RTCCMP	----- 4 -----
		RTCCNT	----- 3 ----- 4 ----- 5 -----
		RTCOUT (ROAP = 0)	----- T_R -----
		RTCOUT (ROAP = 1)	----- T_R -----
		ROLF	----- \rightarrow -----
	1 (Second clock)	RTCCMP	----- X -----
		RTCCNT	----- 3 ----- 4 ----- 5 -----
		RTCOUT (ROAP = 0)	----- T_R -----
		RTCOUT (ROAP = 1)	----- T_R -----
		ROLF	----- \rightarrow -----

T_R : RTCOUT output pulse time = $1 / f_{CK_RTC}$
 \rightarrow : Cleared by software reading ROLF bit

Register Map

The following table shows the RTC registers and reset values. Note all the registers in this unit are located at the V_{DD} power domain.

Table 39. RTC Register Map

Register	Offset	Description	Reset Value
RTCCNT	0x000	RTC Counter Register	0x0000_0000
RTCCMP	0x004	RTC Compare Register	0x0000_0000
RTCCR	0x008	RTC Control Register	0x0000_0F04
RTCSR	0x00C	RTC Status Register	0x0000_0000
RTCIWEN	0x010	RTC Interrupt and Wakeup Enable Register	0x0000_0000

Register Descriptions

RTC Counter Register – RTCCNT

This register defines a 24-bit up-counter which is increased by the CK_SECOND clock.

Address: 0x000

Reset value: 0x0000_0000 (Reset by V_{DD} Power Domain reset only)

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

Bits	Field	Descriptions
[23:0]	RTCCNTV	<p>RTC Counter Value</p> <p>The current value of the RTC counter is returned when reading the RTCCNT register. The RTCCNT register is updated during the falling edge of the CK_SECOND clock. This register is reset by one of the following conditions:</p> <ul style="list-style-type: none"> - V_{DD} Domain software reset – set the PWRST bit in the PWRCCR register - V_{DD} Domain power on reset – POR - Compare match (RTCCNT = RTCCMP) when CMPCLR = 1 (in the RTCCR register) - RTCEN bit changed from 0 to 1

RTC Compare Register – RTCCMP

This register defines a specific value to be compared with the RTC counter value.

Address: 0x004

Reset value: 0x0000_0000 (Reset by V_{DD} Power Domain reset only)

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

Bits	Field	Descriptions
[23:0]	RTCCMPV	<p>RTC Compare Match Value</p> <p>A match condition happens when the value in the RTCCNT register is equal to the RTCCMP value. An interrupt can be generated if the CMIEN bit in the RTCIWEN register is set. When the CMPCLR bit in the RTCCR register is set to 0 and a match condition happens, the CMFLAG bit in the RTCSR register is set while the value in the RTCCNT register is not affected and will continue to count until overflow. When the CMPCLR bit is set to 1 and a match condition happens, the CMFLAG bit in the RTCSR register is set and the RTCCNT register will be reset to zero and then the counter continues to count.</p>

RTC Control Register – RTCCR

This register specifies a range of RTC circuitry control bits.

Address: 0x008

Reset value: 0x0000_0F04 (Reset by V_{DD} Power Domain reset only)

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved			ROLF	ROAP	ROWM	ROES	ROEN	
				RC	0	RW	0	RW	0
	15	14	13	12	11	10	9	8	
Type/Reset	Reserved				RPRE				
					RW	1	RW	1	RW
	7	6	5	4	3	2	1	0	
Type/Reset	Reserved		LSESM	CMPCLR	LSEEN	LSIEN	RTCSRC	RTCEN	
			RW	0	RW	0	RW	1	RW
								0	RW
									0

Bits	Field	Descriptions
[20]	ROLF	<p>RTCOUNT Level Mode Flag</p> <p>0: RTCOUNT Output is inactive</p> <p>1: RTCOUNT Output is holding as active level</p> <p>Set by hardware when in the level mode (ROWM = 1) and a RTCOUNT output event occurs. Cleared by software reading this flag. The RTCOUNT signal will return to the inactive level after software has read this bit.</p>
[19]	ROAP	<p>RTCOUNT Output Active Polarity</p> <p>0: Active level is high</p> <p>1: Active level is low</p>
[18]	ROWM	<p>RTCOUNT Output Waveform Mode</p> <p>0: Pulse mode</p> <p>The output pulse duration is one RTC clock (CK_RTC) period.</p> <p>1: Level mode</p> <p>The RTCOUNT signal will remain at an active level until the ROLF bit is cleared by software reading the ROLF bit.</p>
[17]	ROES	<p>RTCOUNT Output Event Selection</p> <p>0: RTC compare match is selected</p> <p>1: RTC second clock (CK_SECOND) event is selected</p> <p>The ROES bit can be used to select whether the RTCOUNT signal is output on the RTCOUNT pin when a RTC compare match event or the RTC second clock (CK_SECOND) event occurs.</p>
[16]	ROEN	<p>RTCOUNT Output Pin Enable</p> <p>0: Disable RTCOUNT output pin</p> <p>1: Enable RTCOUNT output pin</p> <p>When the ROEN bit is set to 1, the RTCOUNT signal will be at an active level once a RTC compare match or the RTC second clock (CK_SECOND) event occurs. The active polarity and output waveform mode can be configured by the ROAP and ROWM bits respectively. When the ROEN bit is cleared to 0, the RTCOUNT pin will be in a floating state.</p>

Bits	Field	Descriptions
[11:8]	RPRE	RTC Clock Prescaler Select $CK_SECOND = CK_RTC / 2^{RPRE}$ 0000: $CK_SECOND = CK_RTC / 2^0$ 0001: $CK_SECOND = CK_RTC / 2^1$ 0010: $CK_SECOND = CK_RTC / 2^2$... 1111: $CK_SECOND = CK_RTC / 2^{15}$
[5]	LSESM	LSE oscillator Startup Mode 0: Normal startup and requires less operating power 1: Fast startup but requires higher operating current
[4]	CMPCLR	Compare Match Counter Clear 0: 24-bit RTC counter is not affected when compare match condition occurs 1: 24-bit RTC counter is cleared when compare match condition occurs
[3]	LSEEN	LSE oscillator Enable Control 0: LSE oscillator is disabled 1: LSE oscillator is enabled
[2]	LSIEN	LSI oscillator Enable Control 0: LSI oscillator is disabled 1: LSI oscillator is enabled The LSIEN bit default value is 1 which means the LSI oscillator is enabled automatically after the V_{DD} Power Domain is powered up. Note: After the V_{DD} domain is powered on, the internal LSI RC oscillator will start to oscillate. The frequency range of the LSI oscillator is shown in the LSI oscillator electrical characteristics in the datasheet. The device also provides a production trim value to obtain a more accurate oscillation frequency. The procedure is to disable the LSI oscillator and then enable it again after the V_{DD} domain is powered on. After the trimming procedure has completed, the system will automatically load the production trim value to the frequency trimming circuit of the LSI RC oscillator.
[1]	RTC SRC	RTC Clock Source Selection 0: LSI oscillator is selected as the RTC clock source 1: LSE oscillator is selected as the RTC clock source
[0]	RTCEN	RTC Enable Control 0: RTC is disabled 1: RTC is enabled

RTC Status Register – RTCSR

This register stores the counter flags.

Address: 0x00C

Reset value: 0x0000_0000 (Reset by V_{DD} Power Domain reset and RTCEN bit change from 1 to 0)

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved					OVFLAG	CMFLAG	CSECFLAG
						RC	0 RC	0 RC

Bits	Field	Descriptions
[2]	OVFLAG	Counter Overflow Flag 0: Counter overflow has not occurred since the last RTCSR register read operation 1: Counter overflow has occurred since the last RTCSR register read operation This bit is set by hardware when the counter value in the RTCCNT register changes from 0xFFFF_FFFF to 0x0000_0000 and cleared by read operation. This bit is suggested to be read in the RTC IRQ handler and should be taken care when software polling is used.
[1]	CMFLAG	Compare Match Condition Flag 0: Compare match condition has not occurred since the last RTCSR register read operation 1: Compare match condition has occurred since the last RTCSR register read operation. This bit is set by hardware on the CK_SECOND clock falling edge when the RTCCNT register value is equal to the RTCCMP register content. It is cleared by software reading this bit. This bit is suggested for access in the corresponding RTC interrupt routine – do not use software polling during software free running.
[0]	CSECFLAG	CK_SECOND Occurrence Flag 0: CK_SECOND has not occurred since the last RTCSR register read operation 1: CK_SECOND has occurred since the last RTCSR register read operation This bit is set by hardware on the CK_SECOND clock falling edge. It is cleared by software reading this bit. This bit is suggested for access in the corresponding RTC interrupt routine – do not use software polling during software free running.

RTC Interrupt and Wakeup Enable Register – RTCIWEN

This register contains the interrupt and wakeup enable bits.

Address: 0x010

Reset value: 0x0000_0000 (Reset by V_{DD} Power Domain reset only)

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved					OVWEN	CMWEN	CSECWEN
	7	6	5	4	3	2	1	0
Type/Reset	Reserved					RW 0	RW 0	RW 0
						OVIENT	CMIENT	CSECIEN
						RW 0	RW 0	RW 0

Bits	Field	Descriptions
[10]	OVWEN	Counter Overflow Wakeup Enable 0: Counter overflow wakeup is disabled 1: Counter overflow wakeup is enabled
[9]	CMWEN	Compare Match Wakeup Enable 0: Compare match wakeup is disabled 1: Compare match wakeup is enabled
[8]	CSECWEN	Counter Clock CK_SECOND Wakeup Enable 0: Counter Clock CK_SECOND wakeup is disabled 1: Counter Clock CK_SECOND wakeup is enabled
[2]	OVIENT	Counter Overflow Interrupt Enable 0: Counter Overflow Interrupt is disabled 1: Counter Overflow Interrupt is enabled
[1]	CMIENT	Compare Match Interrupt Enable 0: Compare Match Interrupt is disabled 1: Compare Match Interrupt is enabled
[0]	CSECIEN	Counter Clock CK_SECOND Interrupt Enable 0: Counter Clock CK_SECOND Interrupt is disabled 1: Counter Clock CK_SECOND Interrupt is enabled

16 Watchdog Timer (WDT)

Introduction

The Watchdog Timer is a hardware timing circuitry that can be used to detect a system lock-up due to software trapped in a deadlock. The Watchdog Timer can be operated in a reset mode. The Watchdog Timer will generate a reset when the counter counts down to a zero value. Therefore, the software should reload the counter value before a Watchdog Timer underflow occurs. In addition, a reset is also generated if the software reloads the counter before it reaches a delta value. That means that the Watchdog Timer prevents a software deadlock that continuously triggers the Watchdog, the reload must occur when the Watchdog Timer value has a value within a limited window of 0 and WDTD. The Watchdog Timer counter can be stopped when the processor is in the debug or sleep mode. The register write protection function can be enabled to prevent an unexpected change in the Watchdog Timer configuration.

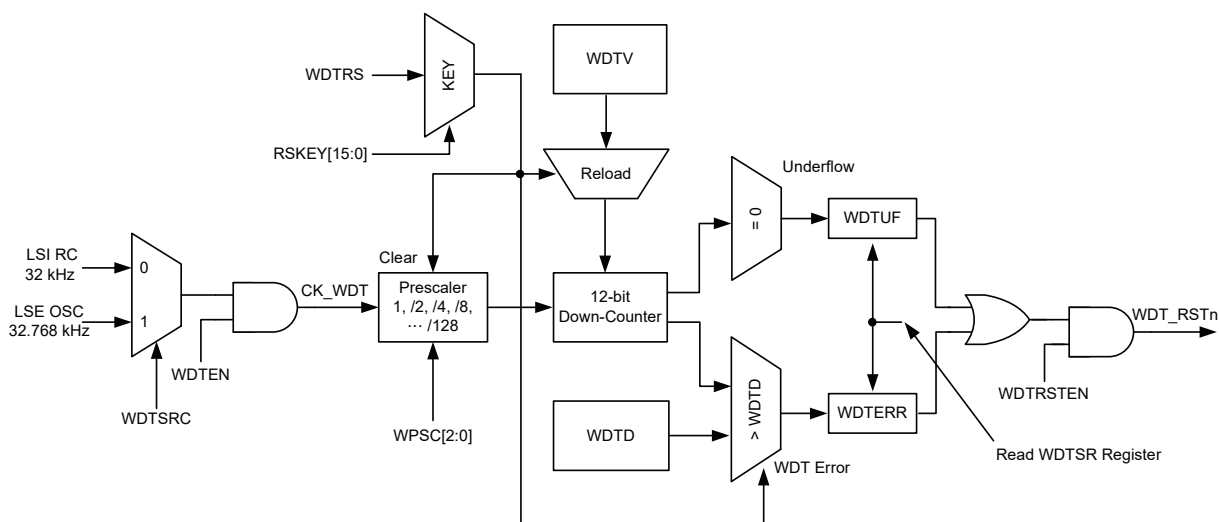


Figure 94. Watchdog Timer Block Diagram

Features

- Clock source from either the internal 32 kHz RC oscillator (LSI) or the external 32,768 Hz oscillator (LSE)
- Can be independently setup to keep running or to stop when entering the Sleep or Deep-Sleep1 mode
- 12-bit down-counter with 3-bit prescaler structure
- Provides reset to the system
- Limited reload window setup function for custom Watchdog Timer reload times
- Watchdog Timer may be stopped when the processor is in the debug
- Reload lock key to prevent unexpected operation
- Configuration register write protection function for counter value, reset enable, delta value, and prescaler value

Functional Description

The Watchdog Timer is formed from a 12-bit count-down counter and a fixed 3-bit prescaler. The largest time-out period is 16 seconds, using the LSE or LSI clock and a 1/128 maximum prescaler value.

The Watchdog Timer configuration setup includes programmable counter reload value, reset enable, window value and prescaler value. These configurations are set using the WDTMR0 and WDTMR1 registers which must be properly programmed before the Watchdog Timer starts counting. In order to prevent unexpected write operations to those configurations, a register write protection function can be enabled by writing any value, other than 0x35CA to PROTECT[15:0], in the WDTPR register. A value of 0x35CA can be written to PROTECT[15:0] to disable the register write protection function before accessing any configuration register. A read operation on PROTECT[0] can obtain the enable/disable status of the register write protection function.

During normal operation, the Watchdog Timer counter should be reloaded before it underflows to prevent the generation of a Watchdog reset. The 12-bit count-down counter can be reloaded with the required Watchdog Timer Counter Value (WDTV) by first setting the WDTRS bit to 1 with the correct key, which is 0x5FA0 in the WDTCR register.

If a software deadlock occurs during a Watchdog Timer reload routine, the reload operation will still go ahead and therefore the software deadlock cannot be detected. To prevent this situation from occurring, the reload operation must be executed in such a way that the value of the Watchdog Timer counter is limited to within a delta value (WDTD). If the Watchdog Timer counter value is greater than the delta value and a reload operation is executed, a Watchdog Timer error will occur. The Watchdog Timer error will cause a Watchdog reset if the related functional control is enabled. Additionally, the above features can be disabled by programming a WDTD value greater than or equal to the WDTV value.

The WDTERR and WDTUF flags in the WDTSR register will be set respectively when the Watchdog Timer error occurs or when a Watchdog Timer underflows. A system reset or writing “1” operation on the WDTSR register will clear the WDTERR and WDTUF flags.

The Watchdog Timer uses two clocks: PCLK and CK_WDT. The PCLK clock is used for APB access to the watchdog registers. The CK_WDT clock is used for the Watchdog Timer functionality and counting. There is some synchronization logic between these two clock domains.

When the system enters the Sleep mode or Deep-Sleep1 mode, the Watchdog Timer counter will either continue to count or stop depending on the WDTSHLT field setup in the WDTMR0 register. However, the Watchdog Timer will always stop when the system is in the Deep-Sleep2 mode. When the Watchdog stops counting, the count value is retained so that it continues counting after the system is woken up from these three sleep modes. A Watchdog reset will occur any time when the Watchdog Timer is running and when it has an operating clock source. When the system enters the debug mode, the Watchdog Timer counter will either continue to count or stop depending on the DBWDT bit of the MCUDBGCR register in the Clock Control Unit.

The Watchdog timer should be used in the following manners:

- Set the Watchdog Timer reload value (WDTV) and reset in the WDTMR0 register.
- Set the Watchdog Timer delta value (WDTD) and prescaler in the WDTMR1 register.
- Start the Watchdog Timer by writing to the WDTCR register with WDTRS = 1 and RSKEY = 0x5FA0.
- Write to the WDTPR register to lock all the Watchdog Timer registers except for WDTCR and WDTPR.
- The Watchdog Timer counter should be reloaded again within the delta value (WDTD).

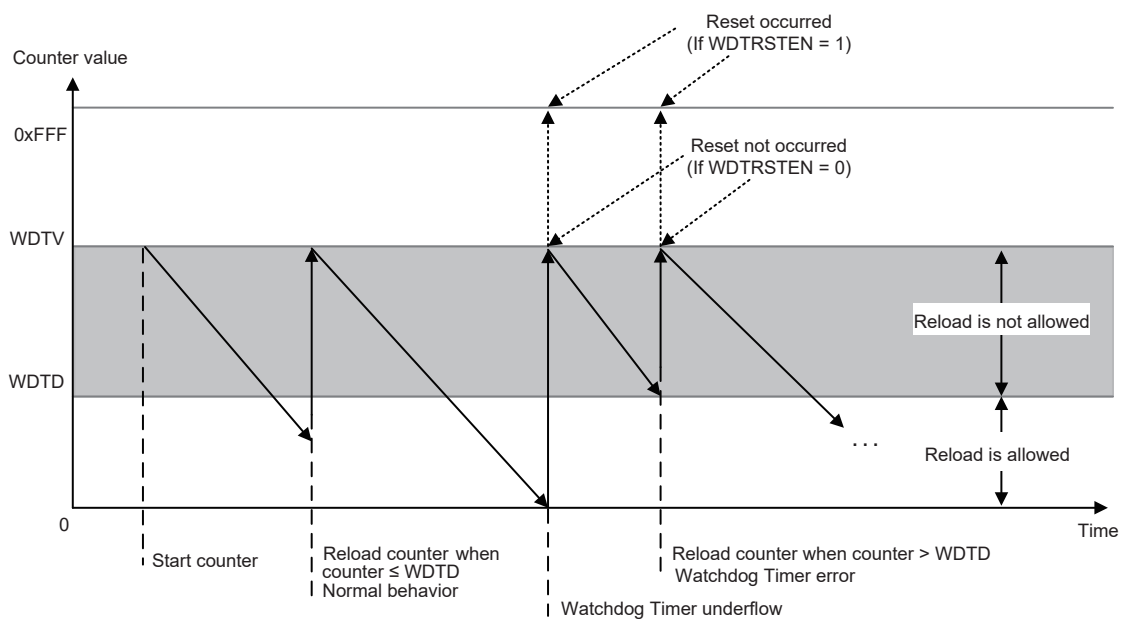


Figure 95. Watchdog Timer Behavior

Register Map

The following table shows the Watchdog Timer registers and reset values.

Table 40. Watchdog Timer Register Map

Register	Offset	Description	Reset Value
WDTCR	0x000	Watchdog Timer Control Register	0x0000_0000
WDTMR0	0x004	Watchdog Timer Mode Register 0	0x0000_0FFF
WDTMR1	0x008	Watchdog Timer Mode Register 1	0x0000_7FFF
WDTSR	0x00C	Watchdog Timer Status Register	0x0000_0000
WDTPR	0x010	Watchdog Timer Protection Register	0x0000_0000
WDTCSR	0x018	Watchdog Timer Clock Selection Register	0x0000_0000

Register Descriptions

Watchdog Timer Control Register – WDTCR

This register is used to reload the Watchdog timer.

Offset: 0x000

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
	RSKEY								
Type/Reset	WO	0	WO	0	WO	0	WO	0	WO
	23	22	21	20	19	18	17	16	
	RSKEY								
Type/Reset	WO	0	WO	0	WO	0	WO	0	WO
	15	14	13	12	11	10	9	8	
	Reserved								
Type/Reset									
	7	6	5	4	3	2	1	0	
	Reserved							WDTRS	
Type/Reset								WO	0

Bits	Field	Descriptions
[31:16]	RSKEY	Watchdog Timer Reload Lock Key The RSKEY [15:0] bits should be written with a 0x5FA0 value to enable the WDT reload operation function. Writing any other value except 0x5FA0 in this field will abort the write operation.
[0]	WDTRS	Watchdog Timer Reload 0: No effect 1: Reload Watchdog Timer This bit is used to reload the Watchdog timer counter as a WDTV value which is stored in the WDTMR0 register. It is set to 1 by software and cleared to 0 by hardware automatically.

Watchdog Timer Mode Register 0 – WDTMR0

This register specifies the Watchdog timer counter reload value and reset enable control.

Offset: 0x004

Reset value: 0x0000_0FFF

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved							WDTEN	
									RW 0
	15	14	13	12	11	10	9	8	
Type/Reset	WDTSHLT		WDRSTEN	Reserved	WDTV				
	RW 0	RW 0	RW 0		RW 1	RW 1	RW 1	RW 1	
	7	6	5	4	3	2	1	0	
Type/Reset	WDTV								
	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	

Bits	Field	Descriptions
[16]	WDTEN	Watchdog Timer Running Enable 0: Watchdog Timer is disabled 1: Watchdog Timer is enabled to run When the Watchdog Timer is disabled, the counter will be reset to its hardware default condition. When the WDTEN bit is set, the Watchdog Timer will be reloaded with the WDTV value and count down.
[15:14]	WDTSHLT	Watchdog Timer Sleep Halt 00: The Watchdog runs when the system is in the Sleep mode or Deep-Sleep1 mode 01: The Watchdog runs when the system is in the Sleep mode and halts in Deep-Sleep1 mode 10 or 11: The Watchdog halts when the system is in the Sleep mode and Deep-Sleep1 mode Note that the Watchdog timer always halts when the system is in the Deep-Sleep2 mode. The Watchdog stops counting when the WDTSHLT field is properly configured in the Sleep mode or Deep-Sleep1 mode. When the Watchdog stops counting, the count value is retained so that it continues counting after the system wakes up from these three sleep modes. If a Watchdog reset occurs in the Sleep or Deep-Sleep1 mode, it will wake up the device.
[13]	WDRSTEN	Watchdog Timer Reset Enable 0: A Watchdog Timer underflow or error has no effect on the system reset 1: A Watchdog Timer underflow or error triggers a Watchdog Timer system reset
[11:0]	WDTV	Watchdog Timer Counter Value WDTV defines the value loaded into the 12-bit Watchdog down-counter.

Watchdog Timer Mode Register 1 – WDTMR1

This register specifies the Watchdog delta value and the prescaler selection.

Offset: 0x008

Reset value: 0x0000_7FFF

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	Reserved	WPSC				WDTD			
		RW	1	RW	1	RW	1	RW	1
	7	6	5	4	3	2	1	0	
Type/Reset	WDTD								
	RW	1	RW	1	RW	1	RW	1	RW

Bits	Field	Descriptions
[14:12]	WPSC	Watchdog Timer Prescaler Selection 000: 1/1 001: 1/2 010: 1/4 011: 1/8 100: 1/16 101: 1/32 110: 1/64 111: 1/128
[11:0]	WDTD	Watchdog Timer Delta Value Define the permitted range to reload the Watchdog Timer. If the Watchdog Timer counter value is less than or equal to WDTD, writing to the WDTCR register with WDTRS = 1 and RSKEY = 0x5FA0 will reload the timer. If the Watchdog Timer value is greater than WDTD, then writing WDTCR with WDTRS = 1 and RSKEY = 0x5FA0 will cause a Watchdog Timer error. This feature can be disabled by programming a WDTD value greater then or equal to the WDTV value.

Watchdog Timer Status Register – WDTSR

This register specifies the Watchdog timer status.

Offset: 0x00C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved						WDTERR	WDTUF
							WC	0 WC 0

Bits	Field	Descriptions
[1]	WDTERR	Watchdog Timer Error 0: No Watchdog Timer error has occurred since the last read of this register 1: A Watchdog Timer error has occurred since the last read of this register Note: A reload operation when the Watchdog Timer counter value is larger than WDTD causes a Watchdog Timer error. Note that this bit is a write-one-clear flag.
[0]	WDTUF	Watchdog Timer Underflow 0: No Watchdog Timer underflow has occurred since the last read of this register 1: A Watchdog Timer underflow has occurred since the last read of this register Note that this bit is a write-one-clear flag.

Watchdog Timer Protection Register – WDTPR

This register specifies the Watchdog timer protect key configuration.

Offset: 0x010

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	PROTECT								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	PROTECT								
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:0]	PROTECT	<p>Watchdog Timer Register Protection</p> <p>For write operation:</p> <p>0x35CA: Disable the Watchdog Timer register write protection</p> <p>Others: Enable the Watchdog Timer register write protection</p> <p>For read operation:</p> <p>0x0000: Watchdog Timer register write protection is disabled</p> <p>0x0001: Watchdog Timer register write protection is enabled</p> <p>This register is used to enable/disable the Watchdog timer configuration register write protection function. All configuration registers become read only except for WDTCR and WDTPR when the register write protection is enabled. Additionally, the read operation of PROTECT[0] can obtain the enable/disable status of the register write protection function.</p>

Watchdog Timer Clock Selection Register – WDTCSR

This register specifies the Watchdog timer clock source selection and lock configuration.

Offset: 0x018

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved			WDTLOCK	Reserved			WDTSRC
				RW 0				RW 0

Bits	Field	Descriptions
[4]	WDTLOCK	Watchdog Timer Lock Mode 0: This bit is only set to 0 on any reset. It can not be cleared by software. 1: This bit is set once only by software and locks the Watchdog Timer function. Software can set this bit to 1 at any time. Once the WDTLOCK bit is set, the function and registers of the Watchdog Timer can not be modified or disabled, including the Watchdog Timer clock source. The lock mode can only be disabled until a system reset occurs.
[0]	WDTSRC	Watchdog Timer Clock Source Selection 0: Internal 32 kHz RC oscillator clock is selected (LSI) 1: External 32.768 kHz crystal oscillator clock is selected (LSE) Select using software to control the Watchdog timer clock source.

17 Inter-Integrated Circuit (I²C)

Introduction

The I²C Module is an internal circuit allowing communication with an external I²C interface which is an industry standard two-line serial interface used for connection to external hardware. These two serial lines are known as a serial data line, SDA, and a serial clock line, SCL. The I²C module provides three data transfer rates: (1) 100 kHz in the Standard mode, (2) 400 kHz in the Fast mode and (3) 1MHz in the Fast mode plus. The SCL period generation registers are used to set different kinds of duty cycle implementation for the SCL pulse.

The SDA line which is connected to the whole I²C bus is a bidirectional data line between the master and slave devices used for the transmission and reception of data. The I²C module also has an arbitration detection function to prevent the situation where more than one master attempts to transmit data on the I²C bus at the same time.

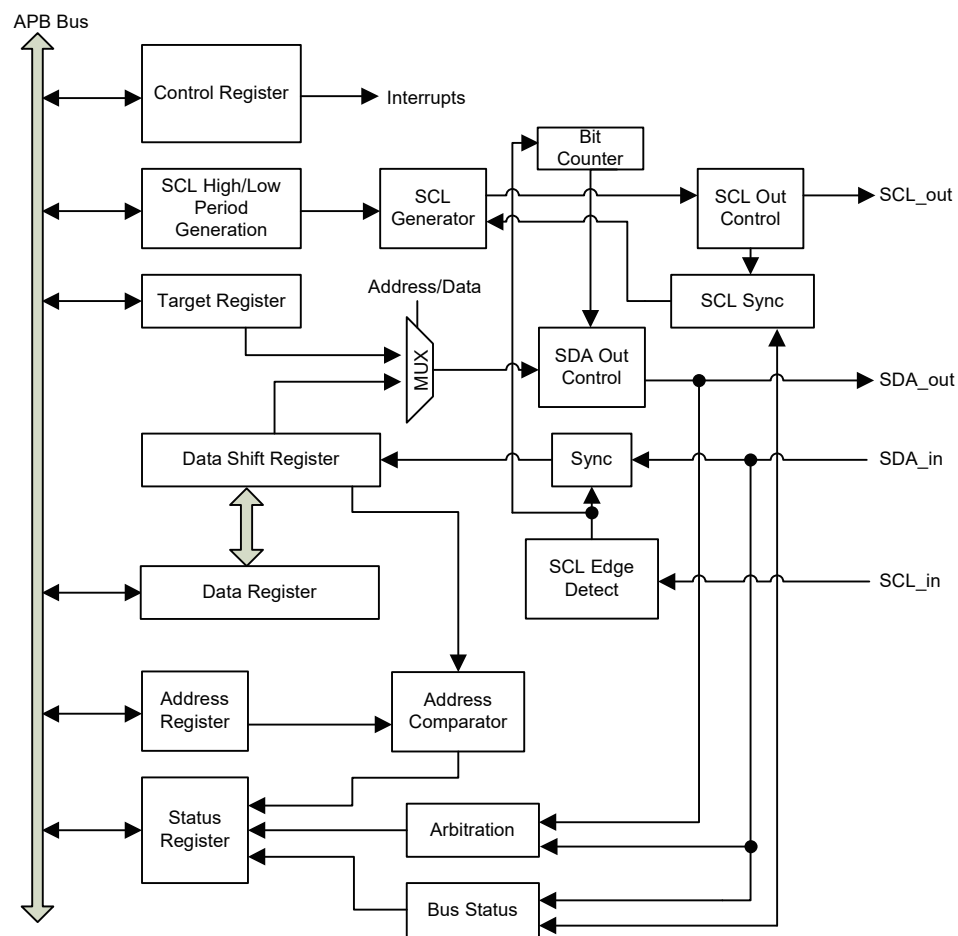


Figure 96. I²C Module Block Diagram

Features

- Two-wire I²C serial interface
 - Serial data line (SDA) and serial clock (SCL)
- Multiple speed modes
 - Standard mode – 100 kHz
 - Fast mode – 400 kHz
 - Fast mode plus – 1 MHz
- Bidirectional data transfer between master and slave
- Multi-master bus – no central master
 - The same interface can act as Master or Slave
- Arbitration among simultaneous transmitting masters without corrupting serial data on the bus
- Clock synchronization
 - Allow devices with different bit rates to communicate via one serial bus
- Supports 7-bit and 10-bit addressing modes and general call addressing
- Multiple slave addresses using address mask function
- Timeout function
- Supports PDMA Interface

Functional Descriptions

Two-Wire Serial Interface

The I²C module has two external lines, the serial data SDA and serial clock SCL lines, to carry information between the interconnected devices connected to the bus. The SCL and SDA lines are both bidirectional and must be connected to a pull-high resistor. When the I²C bus is in the free or idle state, both pins are at a high level to perform the required wired-AND function for multiple connected devices.

START and STOP Conditions

A master device can initialize a transfer by sending a START signal and terminate the transfer with a STOP signal. A START signal is usually referred to as the “S” bit, which is defined as a High to Low transition on the SDA line while the SCL line is high. A STOP signal is usually referred to as the “P” bit, which is defined as a Low to High transition on the SDA line while SCL is high.

A repeated START signal, which is denoted as the “Sr” bit, is functionally identical to the normal START condition. A repeated START signal allows the I²C interface to communicate with another slave device or with the same device but in a different transfer direction without releasing the I²C bus control.

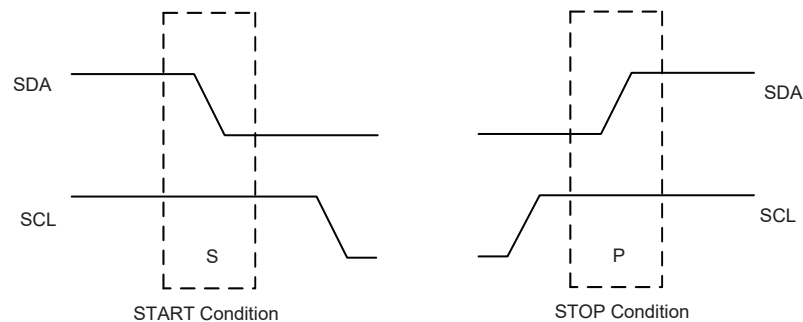


Figure 97. START and STOP Condition

Data Validity

The data on the SDA line must be stable during the high period of the SCL clock. The SDA data state can only be changed when the clock signal on the SCL line is in a low state.

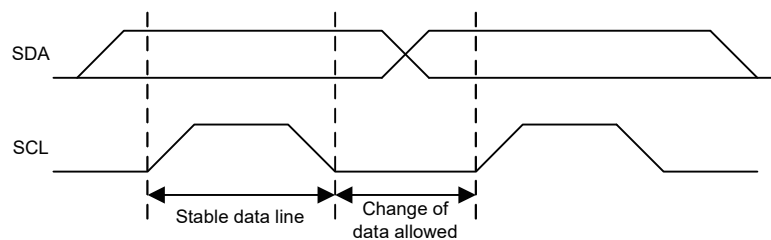


Figure 98. Data Validity

Addressing Format

The I²C interface starts to transfer data after the master device has sent the address to confirm the targeted slave device. The address frame is sent just after the START signal by the master device. The addressing mode selection bit named ADRM in the I2CCR register should be defined to choose either the 7-bit or 10-bit addressing mode.

7-bit Address Format

The 7-bit address format is composed of the 7-bit length slave address, which the master device wants to communicate with, a R/W bit and an ACK bit. The R/W bit defines the direction of the data transfer.

$R/\overline{W} = 0$ (Write): The master transmits data to the addressed slave.

$R/\overline{W} = 1$ (Read): The master receives data from the addressed slave.

The slave address can be assigned through the ADDR field in the I2CADDR register. The slave device sends back the acknowledge bit (ACK) if its slave address matches the transmitted address sent by the master.

Note that it is forbidden to own the same address for two slave devices.

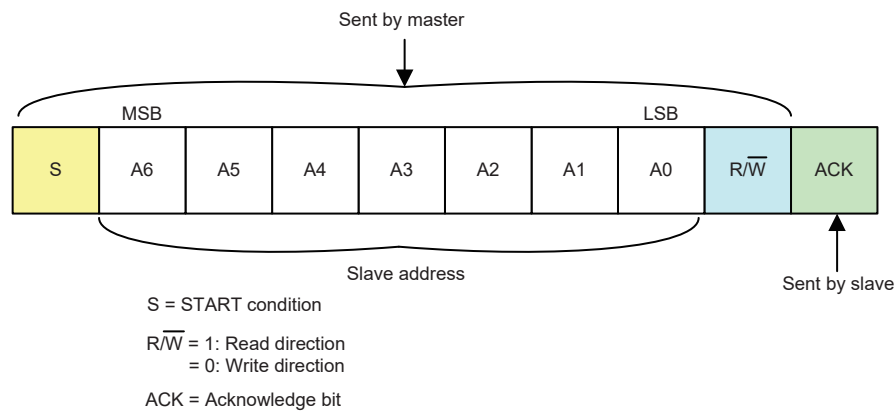


Figure 99. 7-bit Addressing Mode

10-bit Address Format

In order to prevent address clashes, due to the limited range of the 7-bit addresses, a new 10-bit address scheme has been introduced. This enhancement can be mixed with the 7-bit addressing mode which increases the available address range about ten times. For the 10-bit addressing mode, the first two bytes after a START signal include a header byte and an address byte that usually determines which slave will be selected by the master. The header byte is composed of a leading “11110”, the 10th and 9th bits of the slave address. The second byte is the remaining 8 bits of the slave device address.



Figure 100. 10-bit Addressing Write Transmit Mode

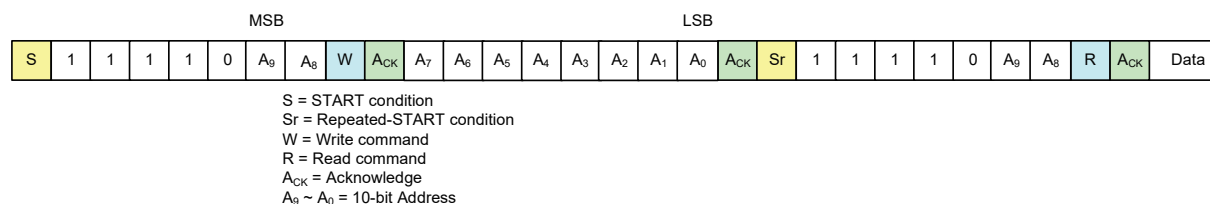


Figure 101. 10-bit Addressing Read Receive Mode

Data Transfer and Acknowledge

Once the slave device address has been matched, the data can be transmitted to or received from the slave device according to the transfer direction specified by the R/W bit. Each byte is followed by an acknowledge bit on the 9th SCL clock.

If the slave device returns a Not-Acknowledge (NACK) signal to the master device, the master device can generate a STOP signal to terminate the data transfer or generate a repeated START signal to restart the transfer.

If the master device sends a Not-Acknowledge (NACK) signal to the slave device, the slave device should release the SDA line for the master device to generate a STOP signal to terminate the transfer.

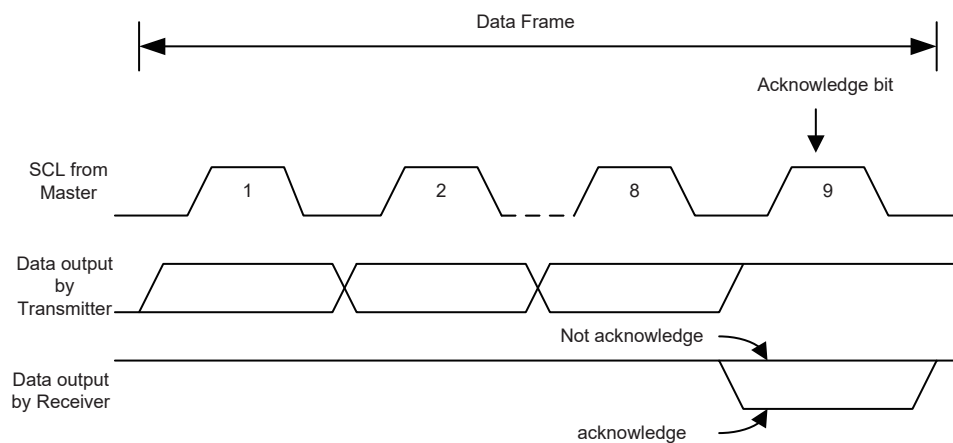


Figure 102. I²C Bus Acknowledge

Clock Synchronization

Only one master device can generate the SCL clock under normal operation. However when there is more than one master trying to generate the SCL clock, the clock should be synchronized so that the data output can be compared. Clock synchronization is performed using the wired-AND connection of the I²C interface to the SCL line.

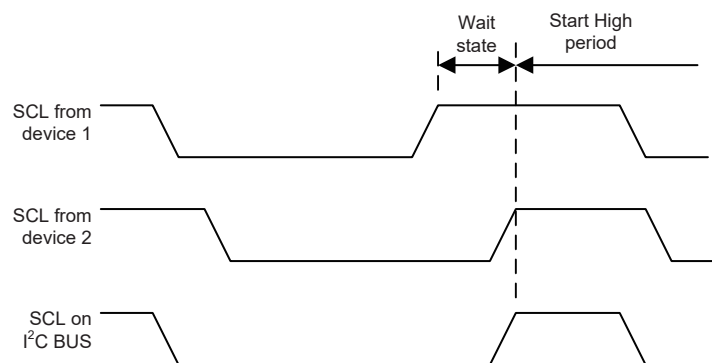


Figure 103. Clock Synchronization during Arbitration

Arbitration

A master may start a transfer only if the I²C bus line is in the free or idle mode. If two or more masters generate a START signal at approximately the same time, an arbitration procedure will occur.

Arbitration takes place on the SDA line and can continue for many bits. The arbitration procedure gives a higher priority to the device that transmits serial data with a binary low bit (logic low). Other master devices which want to transmit binary high bits (logic high) will lose the arbitration. As soon as a master loses the arbitration, the I²C module will set the ARBLOS bit in the I2CSR register and generate an interrupt if the interrupt enable bit, ARBLOSIE, in the I2CIER register is set to 1. Meanwhile, it stops sending data and listens to the bus in order to detect an I²C stop signal. When the stop signal is detected, the master which has lost the arbitration may try to access the bus again.

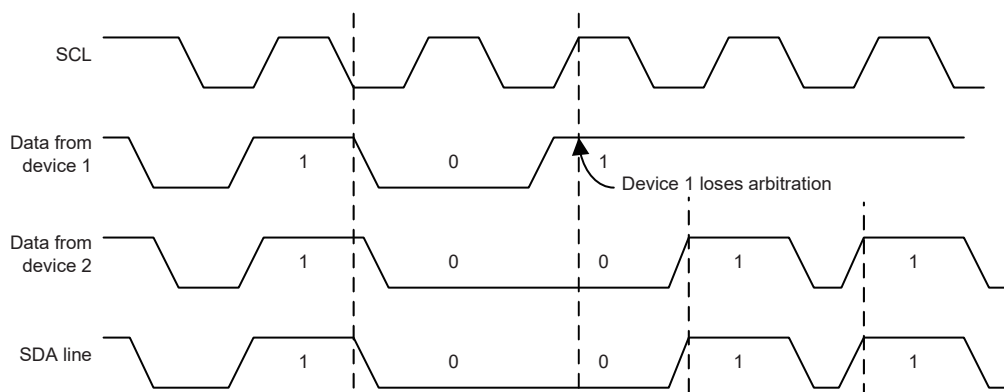


Figure 104. Two Masters Arbitration Procedure

General Call Addressing

The general call addressing function can be used to address all the devices connected to the I²C bus. The master device can activate the general call function by writing a value “00” into the TAR field and setting the RWD bit to 0 in the I2CTAR register on the addressing frame.

The device can support the general call addressing function by setting the corresponding enable control bit GCEN to 1. If the GCEN bit is set to 1 to support the general call addressing, the AA bit in the I2CCR register should also be set to 1 to send an acknowledge signal back when the device receives an address frame with a value of 00H. When this condition occurs, the general call flag, GCS, will be set to 1, but the ADRS flag will not be set.

Bus Error

If an unpredictable START or STOP condition occurs when the data is being transferred on the I²C bus, it will be considered as a bus error and the transferring data will be aborted. When a bus error event occurs, the relevant bus error flag BUSERR in the I2CSR register will set to 1 and both the SDA and SCL lines are released. The BUSERR flag should be cleared by writing a 1 to it to initiate the I²C module to an idle state.

Address Mask Enable

The I²C module provides an address mask function for users to decide which address bit can be ignored during the comparison with the address frame sent from the master. The ADRS flag will be asserted when the unmasked address bits and the address frame sent from the master are matched. Note that this function is only available in the slave mode.

For instance, the user sets a data transfer with the 7-bit addressing mode together with the I2CADDR register value as 0x05 and the I2CADDR register value as 0x55, this means if an address which is sent by an I²C master on the bus is equal to 0x50, 0x51, 0x54 or 0x55, the I²C slave address will all be considered to be matched and the ADRS flag in the I2CSR register will be asserted after the address frame.

Address Snoop

The Address Snoop register, I2CADDRSR, is used to monitor the calling address on the I²C bus during the whole data transfer operation no matter if the I²C module operates as a master or a slave device. Note that the I2CADDRSR register is a read only register and each calling address on the I²C bus will be stored in the I2CADDRSR register automatically even if the I²C device is not addressed.

Operation Mode

The I²C module can operate in the following modes:

- Master Transmitter
- Master Receiver
- Slave Transmitter
- Slave Receiver

The I²C module operates in the slave mode by default. The interface will switch to the master mode automatically after generating a START signal.

Master Transmitter Mode

Start Condition

Users write the target slave device address and communication direction into the I2CTAR register after setting the I2CEN bit in the I2CCR register. The STA flag in the I2CSR register is set by hardware after a start condition occurs. In order to send the following address frame, the STA flag must be cleared to 0 if it has been set to 1. The STA flag is cleared by reading the I2CSR register.

Address Frame

The ADRS flag in the I2CSR register will be set after the address frame is sent by the master device and the acknowledge signal from the address matched slave device is received. In order to send the following data frame, the ADRS flag must be cleared to 0 if it has been set to 1. The ADRS bit is cleared by reading the I2CSR register.

Data Frame

The data to be transmitted to the slave device must be transferred to the I2CDR register.

The TXDE bit in the I2CSR register is set to indicate that the I2CDR register is empty, which results in the SCL line being held at a logic low state. New data must then be transferred to the I2CDR register to continue the data transfer process. Writing a data into the I2CDR register will clear the TXDE flag.

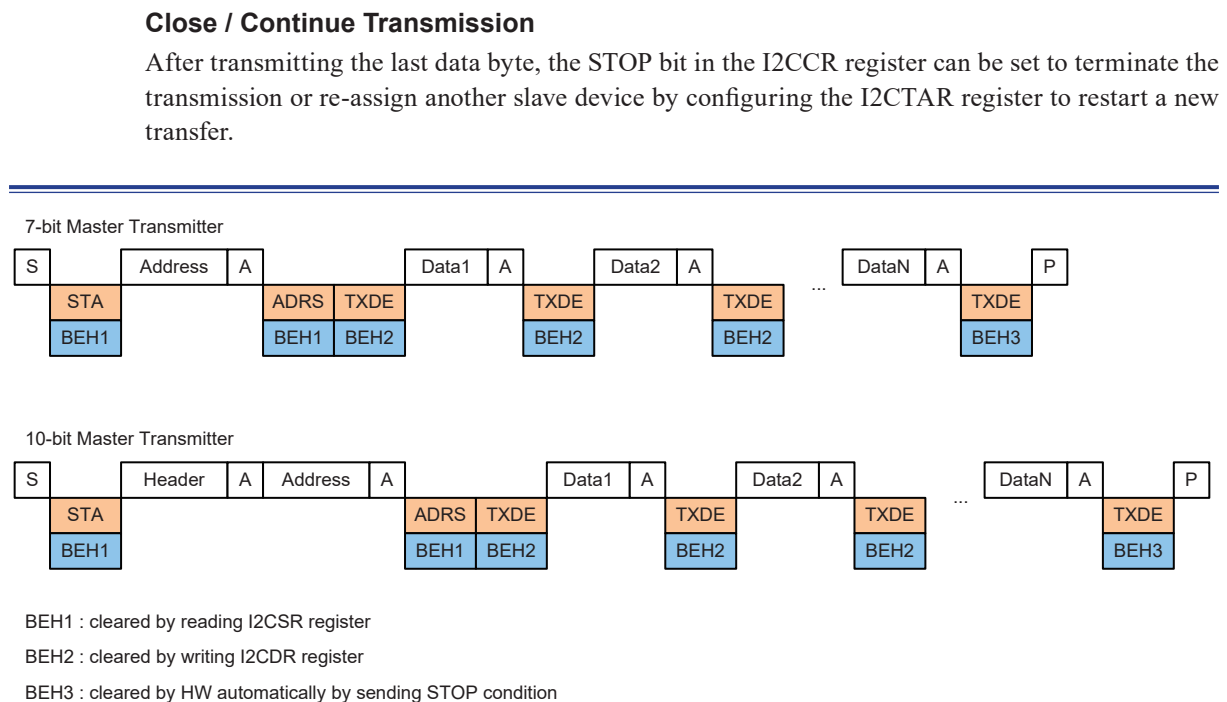


Figure 105. Master Transmitter Timing Diagram

Master Receiver Mode

Start Condition

The target slave device address and communication direction must be written into the I2CTAR register. The STA flag in the I2CSR register is set by hardware after a start condition occurs. In order to send the following address frame, the STA flag must be cleared to 0 if it has been set to 1. The STA flag is cleared by reading the I2CSR register.

Address Frame

In the 7-bit addressing mode: The ADRS flag is set after the address frame is sent by the master device and the acknowledge signal from the address matched slave device is received. In order to receive the following data frame, the ADRS bit must be cleared to 0 if it has been set to 1. The ADRS bit is cleared after reading the I2CSR register.

In the 10-bit addressing mode: The ADRS bit in the I2CSR register will be set twice in the 10-bit addressing mode. The first time the ADRS bit is set is when the first header byte and the second address byte are sent and the acknowledge signals from the slave device are received. The second time the ADRS bit is set is when the second header byte is sent and the slave acknowledge signal is received. In order to receive the following data frame, the ADRS bit must be cleared to 0 if it has been set to 1. The ADRS bit is cleared after reading the I2CSR register. The detailed master receiver mode timing diagram is shown in the following figure.

Data Frame

In the master receiver mode, data is transmitted from the slave device. Once a data is received by the master device, the RXDNE flag in the I2CSR register is set but it will not hold the SCL line. However, if the device receives a complete new data byte and the RXDNE flag has already been set

to 1, the RXBF bit in the I2CSR register will be set to 1 and the SCL line will be held at a logic low state. When this situation occurs, data from the I2CDR register should be read to continue the data transfer process. The RXDNE flag can be cleared after reading the I2CDR register.

Close / Continue Transmission

The master device needs to reset the AA bit in the I2CCR register to send a NACK signal to the slave device before the last data byte transfer has been completed. After the last data byte has been received from the slave device, the master device will hold the SCL line at a logic low state following a NACK signal sent by the master device to the slave device. The STOP bit can be set to terminate the data transfer process or re-assign the I2CTAR register to restart a new transfer.

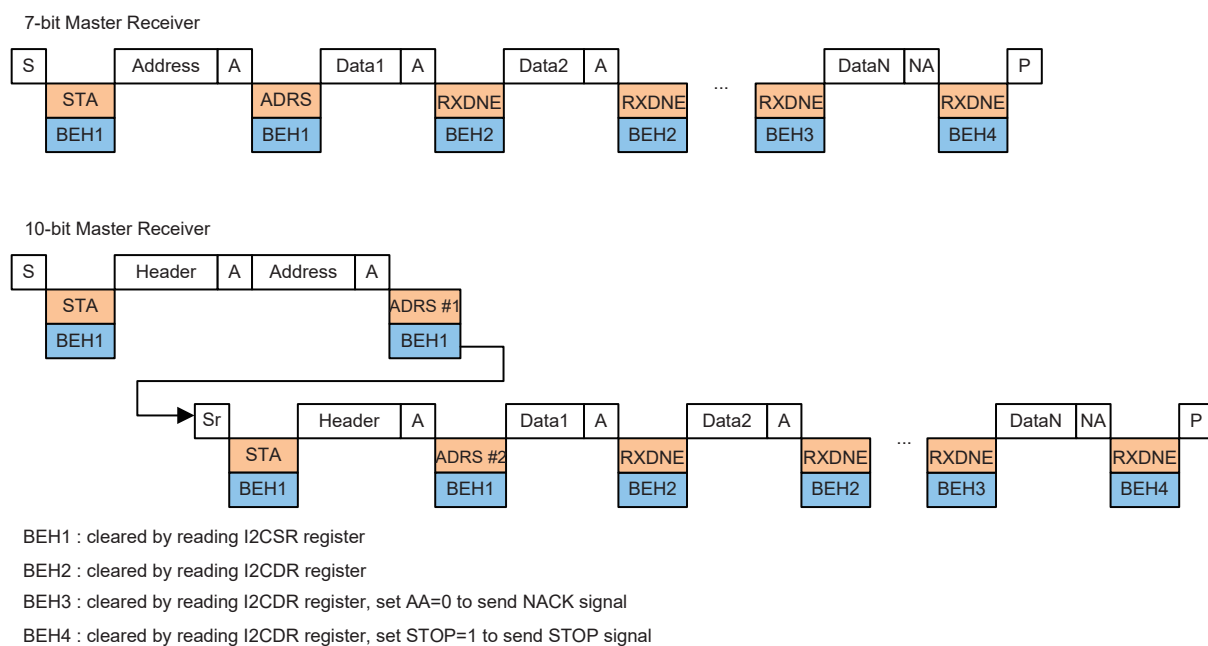


Figure 106. Master Receiver Timing Diagram

Slave Transmitter Mode

Address Frame

In the 7-bit addressing mode, the ADRS bit in the I2CSR register is set after the slave device receives the calling address which matches with the slave device address. In the 10-bit addressing mode, the ADRS bit is set for the first time when the first header byte and the second address byte are both matched. Not that when the second header byte is also matched, the ADRS bit will be set again. After the ADRS bit has been set to 1, it must be cleared to 0 to continue the data transfer process. The ADRS bit is cleared after reading the I2CSR register.

Data Frame

In the Slave transmitter mode, the TXDE bit is set to indicate that the I2CDR is empty, which results in the SCL line being held at a logic low state. New transmission data must then be written into the I2CDR register to continue the data transfer process. Writing a data into the I2CDR register will clear the TXDE bit.

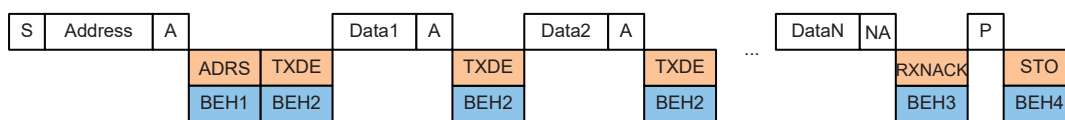
Receive Not-Acknowledge

When the slave device receives a Not-Acknowledge signal, the RXNACK bit in the I2CSR Register is set but it will not hold the SCL line. Writing “1” to RXNACK will clear the RXNACK flag.

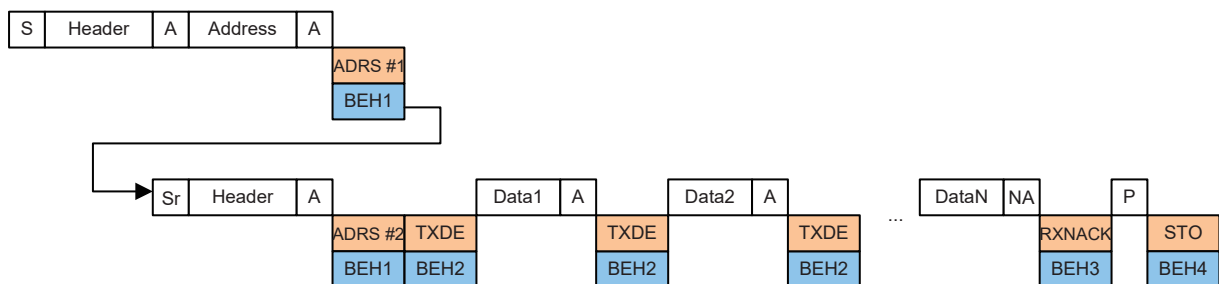
STOP Condition

When the slave device detects a STOP condition, the STO bit in the I2CSR register is set to indicate that the I²C interface transmission is terminated. Reading the I2CSR register can clear the STO flag.

7-bit Slave Transmitter



10-bit Slave Transmitter



BEH1 : cleared by reading I2CSR register

BEH2 : cleared by writing I2CDR register

BEH3 : cleared by writing 1 clear for RXNACK flag, TXDE is not set when NACK is received.

BEH4 : cleared by reading I2CSR register

Figure 107. Slave Transmitter Timing Diagram

Slave Receiver Mode

Address Frame

The ADRS bit in the I2CSR register is set after the slave device receives the calling address which matches with the slave device address. After the ADRS bit has been set to 1, it must be cleared to 0 to continue the data transfer process. The ADRS flag is cleared after reading the I2CSR register.

Data Frame

In the slave receiver mode, the data is transmitted from the master device. Once a data byte is received by the slave device, the RXDNE flag in the I2CSR register is set but it will not hold the SCL line. However, if the device receives a complete new data byte and the RXDNE bit has been set to 1, the RXBF bit in the I2CSR register will be set to 1 and the SCL line will be held at a logic low state. When this situation occurs, data from the I2CDR register should be read to continue the data transfer process. The RXDNE flag bit can be cleared after reading the I2CDR register.

STOP Condition

When the slave device detects a STOP condition, the STO flag bit in the I2CSR register is set to indicate that the I²C interface transmission is terminated. Reading the I2CSR register can clear the STO flag bit.

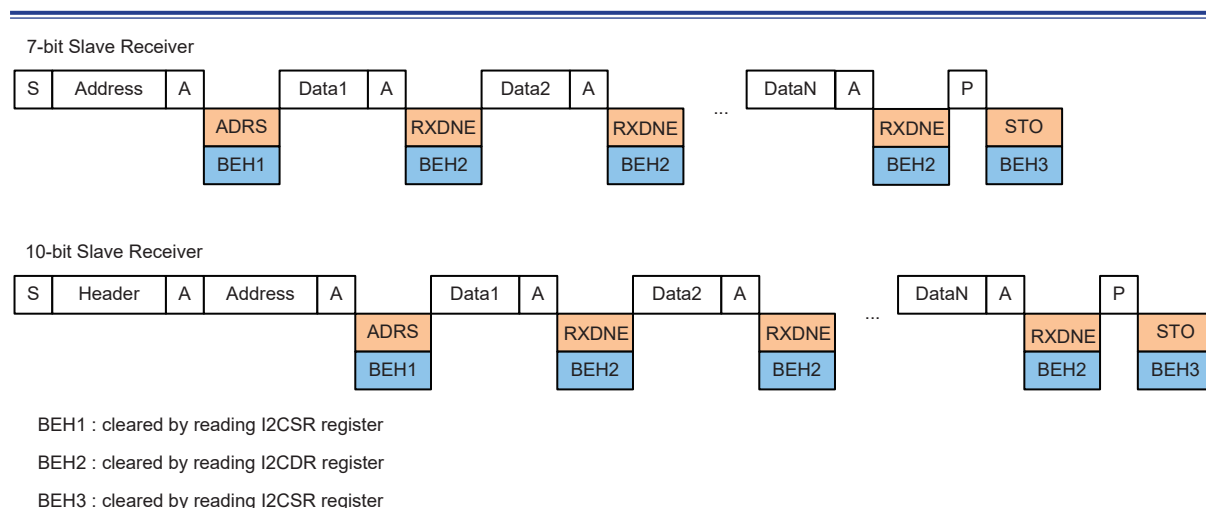


Figure 108. Slave Receiver Timing Diagram

Conditions of Holding SCL Line

The following conditions will cause the SCL line to be held at a logic low state by hardware resulting in all the I²C transfers being stopped. Data transfer will be continued after the creating conditions are eliminated.

Table 41. Conditions of Holding SCL line

Type	Condition	Description	Eliminating Condition
Flag	TXDE	I ² C is used in transmitter mode and I2CDR register needs to have data to transmit. (Note: TXDE won't be asserted after receiving a NACK)	Master case: Writing data to I2CDR register Set TAR Set STOP Slave case: Writing data to I2CDR register
	GCS	I ² C is addressed as slave through general call	Reading I2CSR register
	ADRS	Master: I ² C address frame is sent and an ACK from slave is returned (Note: Reference Figure 105 and Figure 106) Slave: I ² C is addressed as slave device (Note: Reference Figure 107 and Figure 108)	Reading I2CSR register
	STA	Master sends a START signal	Reading I2CSR register
	RXBF	Received a complete new data and meanwhile the RXDNE flag has been set already before.	Reading I2CDR register

Type	Condition	Description	Eliminating Condition
Event	Master receives NACK	No matter in address or data frame, once received a NACK signal will hold SCL line in master mode.	Set TAR Set STOP
	Master sends NACK used in receiver mode	Occurred when receiving the last data byte in Master receiver mode (Note: Reference Figure 105, and RXNACK flag won't be asserted in this case)	Set TAR Set STOP

I²C Timeout Function

In order to reduce the occurrence of I²C lockup problem due to the reception of erroneous clock source, a timeout function is provided. If the I²C bus clock source is not received for a certain timeout period, then a corresponding I²C timeout flag will be asserted. This timeout period is determined by a 16-bit down-counting counter with a programmable preload value. The timeout counter is driven by the I²C timeout clock, f_{I2CTO} , which is specified by the timeout prescaler field in the I2CTOUT register. The TOUT field in the I2CTOUT register is used to define the timeout counter preload value. The timeout function is enabled by setting the ENTOUT bit in the I2CCR register. The timeout counter will start to count down from the preloaded value if the ENTOUT bit is set to 1 and one of the following conditions occurs:

- The I²C master module sends a START signal.
- The I²C slave module detects a START signal.
- The RXBF, TXDE, RXDNE, RXNACK, GCS or ADRS flag is asserted.

The timeout counter will stop counting when the ENTOUT bit is cleared. However, the counter will also stop counting when one of the conditions listed as follows occurs:

- The I²C slave module is not addressed.
- The I²C slave module detects a STOP signal.
- The I²C master module sends a STOP signal.
- The ARBLOS or BUSERR flag in the I2CSR register is asserted.

If the timeout counter underflows, the corresponding timeout flag, TOUTF, in the I2CSR register will be set to 1 and a timeout interrupt will be generated if the relevant interrupt is enabled.

PDMA Interface

The PDMA interface is integrated in the I²C module. The PDMA function can be enabled by setting the TXDMAE or RXDMAE bit to 1 in the transmitter or receiver mode respectively. When the data register is empty in the transmitter mode and the TXDMAE bit is set to 1, the PDMA function will be activated to move data from a certain memory location into the I²C data register. Similarly, when the data register is not empty in the receiver mode and the RXDMAE bit is set to 1, the PDMA function will also be activated to move data from the I²C data register to a specific memory location.

The DMA NACK control bit, DMANACK, is used to determine whether the NACK signal is sent or not when the I²C module operates in the master receiver mode and the PDMA function is enabled. If the DMANACK bit is set to 1 and the data has all been received and moved using the PDMA interface, a NACK signal will automatically be sent out to properly terminate the data transfer.

For a more detailed description about the PDMA configurations, refer to the PDMA chapter.

Register Map

The following table shows the I²C registers and reset values.

Table 42. I²C Register Map

Register	Offset	Description	Reset Value
I2CCR	0x000	I ² C Control Register	0x0000_2000
I2CIER	0x004	I ² C Interrupt Enable Register	0x0000_0000
I2CADDR	0x008	I ² C Address Register	0x0000_0000
I2CSR	0x00C	I ² C Status Register	0x0000_0000
I2CSHPGR	0x010	I ² C SCL High Period Generation Register	0x0000_0000
I2CSLPGR	0x014	I ² C SCL Low Period Generation Register	0x0000_0000
I2CDR	0x018	I ² C Data Register	0x0000_0000
I2CTAR	0x01C	I ² C Target Register	0x0000_0000
I2CADDRMR	0x020	I ² C Address Mask Register	0x0000_0000
I2CADDRSR	0x024	I ² C Address Snoop Register	0x0000_0000
I2CTOUT	0x028	I ² C Timeout Register	0x0000_0000

Register Descriptions

I²C Control Register – I2CCR

This register specifies the corresponding I²C function enable control.

Offset: 0x000

Reset value: 0x0000_2000

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

Bits	Field	Descriptions
[15:14]	SEQFILTER	SDA or SCL Input Sequential Filter Configuration Bits 00: Sequential filter is disabled 01: 1 PCLK glitch filter 1x: 2 PCLK glitch filter Note: This setting would affect the frequency of SCL. Details are described in I2CSLPGR register.
[13]	COMBFILTEREN	SDA or SCL Input Combinational Filter Enable Bit 0: Combinational filter is disabled 1: Combinational filter is enabled

Bits	Field	Descriptions
[12]	ENTOUT	<p>I²C Timeout Function Enable Control</p> <p>0: Timeout Function is disabled 1: Timeout Function is enabled</p> <p>This bit is used to enable or disable the I²C timeout function. When the I2CEN bit is cleared to 0, the ENTOUT bit will be automatically cleared to 0 by hardware. It is recommended that users have to properly configure the PSC and TOUT fields in the I2CTOUT register before the timeout counter starts to count by setting the ENOUT bit to 1.</p>
[10]	DMANACK	<p>DMA Mode NACK Control</p> <p>0: No operation 1: The I²C master receiver module sends a NACK signal automatically after receiving the last byte from the slave transmitter in the DMA mode</p> <p>When the I2CEN bit is cleared to 0, the DMANACK bit is automatically cleared to 0 by hardware.</p>
[9]	RXDMAE	<p>DMA Mode RX Request Enable Control</p> <p>0: RX DMA request is disabled 1: RX DMA request is enabled</p> <p>If the data register is not empty in the receiver mode and the RXDMAE bit is set to 1, the relevant PDMA channel will be activated to move the data from the data register to a specific location which is defined in the corresponding PDMA register. When the I2CEN bit is cleared to 0, the RXDMAE bit is automatically cleared to 0 by hardware.</p>
[8]	TXDMAE	<p>DMA Mode TX Request Enable Control</p> <p>0: TX DMA request is disabled 1: TX DMA request is enabled</p> <p>If the data register is empty in the transmitter mode and the TXDMAE bit is set to 1, the relevant PDMA channel will be activated to move the data from a specific location defined in the related PDMA register to the data register. When the I2CEN bit is cleared to 0, the TXDMAE bit is automatically cleared to 0 by hardware.</p>
[7]	ADRM	<p>Addressing Mode</p> <p>0: 7-bit addressing mode 1: 10-bit addressing mode</p> <p>When the I²C master/slave module operates in the 7-bit addressing mode, it can only send out and respond to a 7-bit address and vice versa. When the I2CEN bit is disabled, the ADRM bit is automatically cleared to 0 by hardware.</p>
[3]	I2CEN	<p>I²C Interface Enable</p> <p>0: I²C interface is disabled 1: I²C interface is enabled</p>
[2]	GCEN	<p>General Call Enable</p> <p>0: General call is disabled 1: General call is enabled</p> <p>When the device receives the calling address with a value of 0x00 and if both the GCEN and the AA bits are set to 1, then the I²C interface is addressed as a slave and the GCS bit in the I2CSR register is set to 1. When the I2CEN bit is cleared to 0, the GCEN bit is automatically cleared to 0 by hardware.</p>
[1]	STOP	<p>STOP Condition Control</p> <p>0: No action 1: Send a STOP condition in master mode</p> <p>This bit is set to 1 by software to generate a STOP condition and automatically cleared to 0 by hardware. The STOP bit is only available for the master device.</p>

Bits	Field	Descriptions
[0]	AA	Acknowledge Bit 0: Send a Not-Acknowledge (NACK) signal after a byte is received 1: Send an Acknowledge (ACK) signal after a byte is received When the I2CEN bit is cleared to 0, the AA bit is automatically cleared to 0 by hardware.

I²C Interrupt Enable Register – I2CIER

This register specifies the corresponding I²C interrupt enable bits.

Offset: 0x004

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved					RXBFI	TXDEI	RXDNEI
						RW	0	RW
							0	RW
								0
	15	14	13	12	11	10	9	8
Type/Reset	Reserved				TOUTI	BUSERRI	RXNACKI	ARBLOSI
					RW	0	RW	0
							0	RW
								0
	7	6	5	4	3	2	1	0
Type/Reset	Reserved				GCSI	ADRSI	STOI	STAI
					RW	0	RW	0
							0	RW
								0

Bits	Field	Descriptions
[18]	RXBFI	RX Buffer Full Interrupt Enable Bit 0: Interrupt is disabled 1: Interrupt is enabled When the I2CEN bit in the I2CCR register is cleared to 0, this bit is cleared to 0 by hardware.
[17]	TXDEI	Data Register Empty Interrupt Enable Bit in Transmitter Mode 0: Interrupt is disabled 1: Interrupt is enabled When the I2CEN bit in the I2CCR register is cleared to 0, this bit is cleared to 0 by hardware.
[16]	RXDNEI	Data Register Not Empty Interrupt Enable Bit in Received Mode 0: Interrupt is disabled 1: Interrupt is enabled When the I2CEN bit in the I2CCR register is cleared to 0, this bit is cleared to 0 by hardware.
[11]	TOUTI	Timeout Interrupt Enable Bit 0: Interrupt is disabled 1: Interrupt is enabled When the I2CEN bit in the I2CCR register is cleared to 0, this bit is cleared to 0 by hardware.

Bits	Field	Descriptions
[10]	BUSERRIE	Bus Error Interrupt Enable Bit 0: Interrupt is disabled 1: Interrupt is enabled When the I2CEN bit in the I2CCR register is cleared to 0, this bit is cleared to 0 by hardware.
[9]	RXNACKIE	Received Not-Acknowledge Interrupt Enable Bit 0: Interrupt is disabled 1: Interrupt is enabled When the I2CEN bit in the I2CCR register is cleared to 0, this bit is cleared to 0 by hardware.
[8]	ARBLOSIE	Arbitration Loss Interrupt Enable Bit in the I ² C multi-master mode 0: Interrupt is disabled 1: Interrupt is enabled When the I2CEN bit in the I2CCR register is cleared to 0, this bit is cleared to 0 by hardware.
[3]	GCSIE	General Call Slave Interrupt Enable Bit 0: Interrupt is disabled 1: Interrupt is enabled When the I2CEN bit in the I2CCR register is cleared to 0, this bit is cleared to 0 by hardware.
[2]	ADRSIE	Slave Address Match Interrupt Enable Bit 0: Interrupt is disabled 1: Interrupt is enabled When the I2CEN bit in the I2CCR register is cleared to 0, this bit is cleared to 0 by hardware.
[1]	STOIE	STOP Condition Detected Interrupt Enable Bit 0: Interrupt is disabled 1: Interrupt is enabled When the I2CEN bit in the I2CCR register is cleared to 0, this bit is cleared to 0 by hardware. The bit is used for the I ² C slave mode only.
[0]	STAIE	START Condition Transmit Interrupt Enable Bit 0: Interrupt is disabled 1: Interrupt is enabled When the I2CEN bit in the I2CCR register is cleared to 0, this bit is cleared to 0 by hardware. The bit is used for the I ² C master mode only.

I²C Address Register – I2CADDR

This register specifies the I²C device address.

Offset: 0x008

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
	Reserved								
Type/Reset									
	23	22	21	20	19	18	17	16	
	Reserved								
Type/Reset									
	15	14	13	12	11	10	9	8	
	Reserved						ADDR		
Type/Reset							RW	0	RW
	7	6	5	4	3	2	1	0	
	ADDR								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[9:0]	ADDR	<p>Device Address</p> <p>The register indicates the I²C device address. When the I²C device is used in the 7-bit addressing mode, only the ADDR[6:0] bits will be compared with the received address sent from the I²C master device.</p>

I²C Status Register – I2CSR

This register contains the I²C operation status.

Offset: 0x00C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved		TXNRX	MASTER	BUSBUSY	RXBF	TXDE	RXDNE	
			RO	0	RO	0	RO	0	RO
	15	14	13	12	11	10	9	8	
Type/Reset	Reserved				TOUTF	BUSERR	RXNACK	ARBLOS	
					WC	0	WC	0	WC
	7	6	5	4	3	2	1	0	
Type/Reset	Reserved				GCS	ADRS	STO	STA	
					RC	0	RC	0	RC

Bits	Field	Descriptions
[21]	TXNRX	Transmitter / Receiver Mode 0: Receiver mode 1: Transmitter mode Read only bit.
[20]	MASTER	Master Mode 0: I ² C is in the slave mode or idle 1: I ² C is in the master mode The I ² C interface is switched as a master device on the I ² C bus when the I2CTAR register is assigned and the I ² C bus is idle. The MASTER bit is cleared by hardware when software disables the I ² C bus by clearing the I2CEN bit to 0 or sends a STOP condition to the I ² C bus or the bus error is detected. This bit is set and cleared by hardware and is a read only bit.
[19]	BUSBUSY	Bus Busy 0: I ² C bus is idle 1: I ² C bus is busy The I ² C interface hardware starts to detect the I ² C bus status if the interface is enabled by setting the I2CEN bit to 1. It is set to 1 when the SDA or SCL signal is detected to have a logic low state and cleared when a STOP condition is detected.
[18]	RXBF	Buffer Full Flag in Receiver Mode 0: Data buffer is not full 1: Data buffer is full This bit is set when the data register I2CDR has already stored a data byte and meanwhile the data shift register also has been received a complete new data byte. The RXBF bit is cleared by software reading the I2CDR register.

Bits	Field	Descriptions
[17]	TXDE	<p>Data Register Empty in Transmitter Mode</p> <p>0: Data register I2CDR is not empty 1: Data register I2CDR is empty</p> <p>This bit is set when the I2CDR register is empty in the Transmitter mode. Note that the TXDE bit will be set after the address frame is being transmitted to inform that the data to be transmitted should be loaded into the I2CDR register. The TXDE bit is cleared by software writing data to the I2CDR register in both the master and slave mode or cleared automatically by hardware after setting the STOP signal to terminate the data transfer or setting the I2CTAR register to restart a new data transfer in the master mode.</p>
[16]	RXDNE	<p>Data Register Not Empty in Receiver Mode</p> <p>0: Data register I2CDR is empty 1: Data register I2CDR is not empty</p> <p>This bit is set when the I2CDR register is not empty in the receiver mode. The RXDNE bit is cleared by software reading the data byte from the I2CDR register.</p>
[11]	TOUTF	<p>Timeout Counter Underflow Flag</p> <p>0: No timeout counter underflow has occurred 1: Timeout counter underflow has occurred</p> <p>Writing "1" to this bit will clear the TOUTF flag.</p>
[10]	BUSERR	<p>Bus Error Flag</p> <p>0: No bus error has occurred 1: Bus error has occurred</p> <p>This bit is set by hardware when the I²C interface detects a misplaced START or STOP condition in a transfer process. Writing a "1" to this bit will clear the BUSERR flag.</p> <p>In Master Mode: Once the Bus Error event occurs, both the SDA and SCL lines are released by hardware and the BUSERR flag is asserted. The application software has to clear the BUSERR flag before the next address byte is transmitted.</p> <p>In Slave Mode: Once a misplaced START or STOP condition has been detected by the slave device, the software must clear the BUSERR flag before the next address byte is received.</p>
[9]	RXNACK	<p>Received Not-Acknowledge Flag</p> <p>0: Acknowledge is returned from receiver 1: Not-Acknowledge is returned from receiver</p> <p>The RXNACK bit indicates that the Not-Acknowledge signal is received in master or slave transmitter mode. Writing "1" to this bit will clear the RXNACK flag.</p>
[8]	ARBLOS	<p>Arbitration Loss Flag</p> <p>0: No arbitration loss is detected 1: Bit arbitration loss is detected</p> <p>This bit is set by hardware on the current clock which the I²C interface loses the bus arbitration to another master during the address or data frame transmission. Writing "1" to this bit will clear the ARBLOS flag. Once the ARBLOS flag is asserted by hardware, the ARBLOS flag must be cleared before the next transmission.</p>
[3]	GCS	<p>General Call Slave Flag</p> <p>0: No general call slave occurs 1: I²C interface is addressed by a general call command</p> <p>When the I²C interface receives an address with a value of 0x00 or 0x000 in the 7-bit or 10-bit addressing mode, if both the GCEN and the AA bit are set to 1, then it is switched as a general call slave. This flag is cleared automatically after being read.</p>

Bits	Field	Descriptions
[2]	ADRS	<p>Address Transmit (master mode) / Address Receive (slave mode) Flag</p> <p>Address Sent in Master Mode:</p> <p>0: Address frame has not been transmitted</p> <p>1: Address frame has been transmitted</p> <p>For the 7-bit addressing mode, this bit is set after the master device receives the address frame acknowledge bit sent from the slave device. For the 10-bit addressing mode, this bit is set after receiving the acknowledge bits of the first header byte and the second address. Note that when the second header byte, if exists, is acknowledged, this bit will also be set.</p> <p>Address Matched in Slave Mode:</p> <p>0: I²C interface is not addressed</p> <p>1: I²C interface is addressed as slave</p> <p>When the I²C interface has received the calling address that matches the address defined in the I2CADDR register together with the AA bit being set to 1 in the I2CCR register, it will be switched to a slave mode. This flag is cleared automatically after the I2CSR register has been read.</p>
[1]	STO	<p>STOP Condition Detected Flag</p> <p>0: No STOP condition is detected</p> <p>1: STOP condition is detected in slave mode</p> <p>This bit is only available for the slave mode and is cleared automatically after the I2CSR register is read.</p>
[0]	STA	<p>START Condition Transmit</p> <p>0: No START condition is detected</p> <p>1: START condition is transmitted in master mode</p> <p>This bit is only available for the master mode and is cleared automatically after the I2CSR register is read.</p>

I²C SCL High Period Generation Register – I2CSHPGR

This register specifies the I²C SCL clock high period interval.

Offset: 0x010

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	SHPG								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	SHPG								
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:0]	SHPG	<p>SCL Clock High Period Generation</p> <p>High period duration setting $SCL_{HIGH} = T_{PCLK} \times (SHPG + d)$ where T_{PCLK} is the APB bus peripheral clock (PCLK) period, and d value depends on the setting of SEQFILTER in the I²C Control Register (I2CCR).</p> <p>If SEQFILTER = 00, d = 6</p> <p>If SEQFILTER = 01, d = 8</p> <p>If SEQFILTER = 10 or 11, d = 9</p>

I²C SCL Low Period Generation Register – I2CSLPGR

This register specifies the I²C SCL clock low period interval.

Offset: 0x014

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	SLPG								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	SLPG								
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:0]	SLPG	<p>SCL Clock Low Period Generation</p> <p>Low period duration setting $SCL_{LOW} = T_{PCLK} \times (SLPG + d)$ where T_{PCLK} is the APB bus peripheral clock (PCLK) period, and d value depends on the setting of SEQFILTER in the I²C Control Register (I2CCR).</p> <p>If SEQFILTER = 00, d = 6</p> <p>If SEQFILTER = 01, d = 8</p> <p>If SEQFILTER = 10 or 11, d = 9</p>

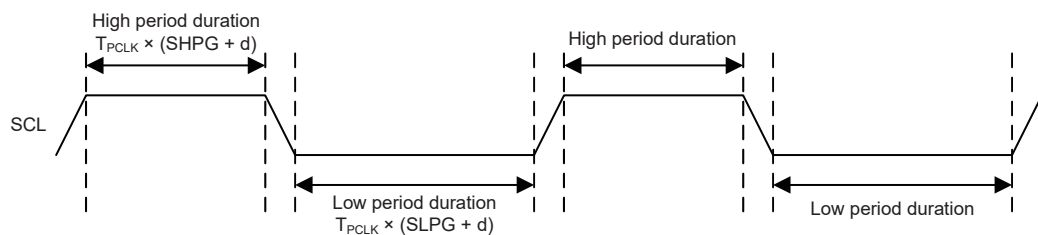


Figure 109. SCL Timing Diagram

Table 43. I²C Clock Setting Example

I ² C Clock	$T_{SCL} = T_{PCLK} \times [(SHPG + d) + (SLPG + d)]$ (where d = 6)					
	SHPG + SLPG Value at PCLK					
	8 MHz	20 MHz	24 MHz	40 MHz	48 MHz	60 MHz
100 kHz (Standard Mode)	68	188	228	388	468	588
400 kHz (Fast Mode)	8	38	48	88	108	138
1 MHz (Fast Mode Plus)	N/A	8	12	28	36	48

I²C Data Register – I2CDR

This register specifies the data to be transmitted or received by the I²C module.

Offset: 0x018

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	DATA							
	RW	0	RW	0	RW	0	RW	0
	0	RW	0	RW	0	RW	0	RW
	0	RW	0	RW	0	RW	0	RW
	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[7:0]	DATA	<p>I²C Data Register</p> <p>For the transmitter mode, a data byte which is transmitted to a slave device can be assigned to these bits. The TXDE flag is cleared if the application software assigns new data to the I2CDR register. For the receiver mode, a data byte is received bit by bit from MSB to LSB through the I²C interface and stored in the data shift register. Once the acknowledge bit is given, the data shift register value is delivered into the I2CDR register if the RXDNE flag is equal to 0.</p>

I²C Target Register – I2CTAR

This register specifies the target device address to be communicated.

Offset: 0x01C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	Reserved					RWD		TAR	
						RW	0	RW	0
	7	6	5	4	3	2	1	0	
Type/Reset	TAR								
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[10]	RWD	Read or Write Direction 0: Write direction to target slave address 1: Read direction from target slave address If this bit is set to 1 in the 10-bit master receiver mode, the I ² C interface will initiate a byte with a value of 11110XX0b in the first header frame and then continue to deliver a byte with a value of 11110XX1b in the second header frame by hardware automatically.
[9:0]	TAR	Target Slave Address The I ² C interface will assign a START signal and send a target slave address automatically once the data is written to this register. When the system wants to send a repeated START signal to the I ² C bus, it is suggested to set the I2CTAR register after a byte transfer is completed. It is not allowed to set TAR in the address frame. I2CTAR[9:7] is not available under the 7-bit addressing mode.

I²C Address Mask Register – I2CADDRMR

This register specifies which bit of the I²C address is masked and not compared with corresponding bit of the received address frame.

Offset: 0x020

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	Reserved						RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	ADDMR								
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[9:0]	ADDMR	<p>Address Mask Control Bit</p> <p>The ADDMR[i] is used to specify whether the ith bit of the ADDR field in the I2CADDR register is masked and is compared with the received address frame or not on the I²C bus. The register is only used for the I²C slave mode only.</p> <p>0: ith bit of the ADDR is compared with the address frame on the I²C bus.</p> <p>1: ith bit of the ADDR is masked and not compared with the address frame on the I²C bus.</p>

I²C Address Snoop Register – I2CADDRSR

This register is used to indicate the address frame value appeared on the I²C bus.

Offset: 0x024

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved						ADDRSR	
	7	6	5	4	3	2	1	0
Type/Reset	ADDRSR							
	RO	0	RO	0	RO	0	RO	0
	RO	0	RO	0	RO	0	RO	0

Bits	Field	Descriptions
[9:0]	ADDRSR	Address Snoop Once the I2CEN bit is enabled, the calling address value on the I ² C bus will automatically be loaded into this ADDSR field.

I²C Timeout Register – I2CTOUT

This register specifies the I²C timeout counter preload value and clock prescaler ratio.

Offset: 0x028

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved					PSC			
	15	14	13	12	11	10	9	8	
Type/Reset	TOUT								
	7	6	5	4	3	2	1	0	
Type/Reset	TOUT								

Bits	Field	Descriptions
[18:16]	PSC	<p>I²C Timeout Counter Prescaler Selection</p> <p>This PSC field is used to specify the I²C timeout counter clock frequency, f_{I2CTO}. The timeout clock frequency is obtained using the following formula.</p> $f_{I2CTO} = \frac{f_{PCLK}}{2^{PSC}}$ <p>PSC = 0 → $f_{I2CTO} = f_{PCLK} / 2^0 = f_{PCLK}$ PSC = 1 → $f_{I2CTO} = f_{PCLK} / 2^1 = f_{PCLK} / 2$ PSC = 2 → $f_{I2CTO} = f_{PCLK} / 2^2 = f_{PCLK} / 4$... PSC = 7 → $f_{I2CTO} = f_{PCLK} / 2^7 = f_{PCLK} / 128$</p>
[15:0]	TOUT	<p>I²C Timeout Counter Preload Value</p> <p>The TOUT field is used to define the counter preloaded value.</p> <p>The counter value is reloaded as any one of the following conditions occurs:</p> <ol style="list-style-type: none"> 1. The RXBF, TXDE, RXDNE, RXNACK, GCS or ADRS flag in the I2CSR register is asserted. 2. The I²C master module sends a START signal. 3. The I²C slave module detects a START signal. <p>The counter stops counting as any one of the following conditions occurs:</p> <ol style="list-style-type: none"> 1. The I²C slave device is not addressed. 2. The I²C master module sends a STOP signal. 3. The I²C slave module detects a STOP signal. 4. The ARBLOS or BUSERR flag in the I2CSR register is asserted.

18 Serial Peripheral Interface (SPI)

Introduction

The Serial Peripheral Interface, SPI, provides an SPI protocol data transmit and receive functions in both master or slave mode. The SPI interface uses 4 pins, among which are the serial data input and output lines MISO and MOSI, the clock line SCK, and the slave select line SEL. One SPI device acts as a master who controls the data flow using the SEL and SCK signals to indicate the start of the data communication and the data sampling rate. To receive the data bits, the streamlined data bits which range from 1 bit to 16 bits specified by the DFL field in the SPICR1 register are latched on a specific clock edge and stored in the data register or in the RX FIFO. Data transmission is carried in a similar way but with the reverse sequence. The mode fault detection provides a capability for multi-master applications.

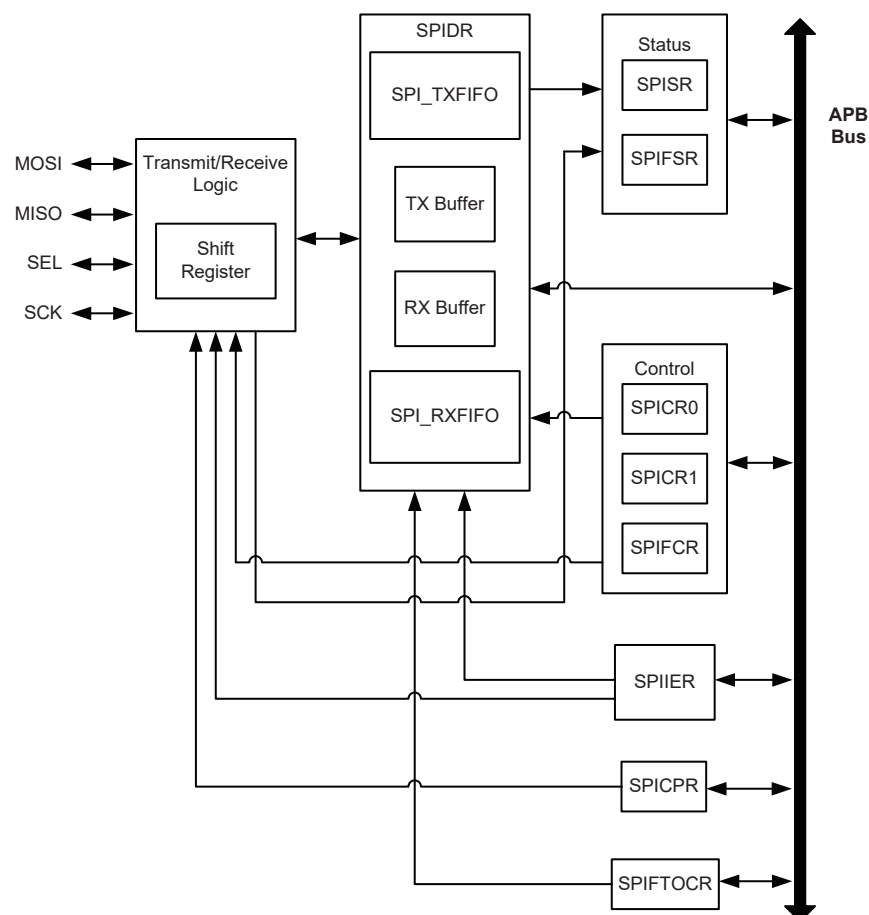


Figure 110. SPI Block Diagram

Features

- Master or slave mode
- Master mode speed up to $f_{PCLK}/2$
- Slave mode speed up to $f_{PCLK}/3$
- Programmable data frame length up to 16 bits
- FIFO Depth: 8 levels
- MSB or LSB first shift selection
- Programmable slave select high or low active polarity
- Multi-master and multi-slave operation
- Master mode supports dual output read mode of SPI series NOR Flash
- Four error flags with individual interrupt
 - Read overrun
 - Write collision
 - Mode fault
 - Slave abort
- Support PDMA interface

Functional Descriptions

Master Mode

Each data frame can range from 1 to 16 bits in data length. The first bit of the transmitted data can be either an MSB or LSB determined by the FIRSTBIT bit in the SPICR1 register. The SPI module is configured as a master or a slave by setting the MODE bit in the SPICR1 register. When the MODE bit is set, the SPI module is configured as a master and will generate the serial clock on the SCK pin. The data stream will transmit data in the shift register to the MOSI pin on the serial clock edge. The SEL pin is active during the full data transmission. When the SELAP bit in the SPICR1 register is set, the SEL pin is active high during the complete data transactions. When the SELM bit in the SPICR1 register is set, the SEL pin will be driven by the hardware automatically and the time interval between the active SEL edge and the first edge of SCK is equal to half an SCK period.

Slave Mode

In the slave mode, the SCK pin acts as an input pin and the serial clock will be derived from the external master device. The SEL pin also acts as an input. When the SELAP bit is cleared to 0, the SEL signal is active low during the full data stream reception. When the SELAP bit is set to 1, the SEL signal will be active high during the full data stream reception.

Note: For the slave mode, the APB clock, known as f_{PCLK} , must be at least 3 times faster than the external SCK clock input frequency.

SPI Serial Frame Format

The SPI interface format is based on the Clock Polarity, CPOL, and the Clock Phase, CPHA, configurations.

- Clock Polarity Bit – CPOL

When the Clock Polarity bit is cleared to 0, the SCK line idle state is low. When the Clock Polarity bit is set to 1, the SCK line idle state is high.

■ Clock Phase Bit – CPHA

When the Clock Phase bit is cleared to 0, the data is sampled on the first SCK clock transition.

When the Clock Phase bit is set to 1, the data is sampled on the second SCK clock transition.

There are four formats contained in the SPI interface. Table 44 shows how to configure these formats by setting the FORMAT field in the SPICR1 register.

Table 44. SPI Interface Format Setup

FORMAT [2:0]	CPOL	CPHA
001	0	0
010	0	1
110	1	0
101	1	1
Others	Reserved	

CPOL = 0, CPHA = 0

In this format, the received data is sampled on the SCK line rising edge while the transmitted data is changed on the SCK line falling edge. In the master mode, the first bit is driven when data is written into the SPIDR Register. In the slave mode, the first bit is driven when the SEL signal goes to an active level. Figure 110 shows the single byte data transfer timing of this format.

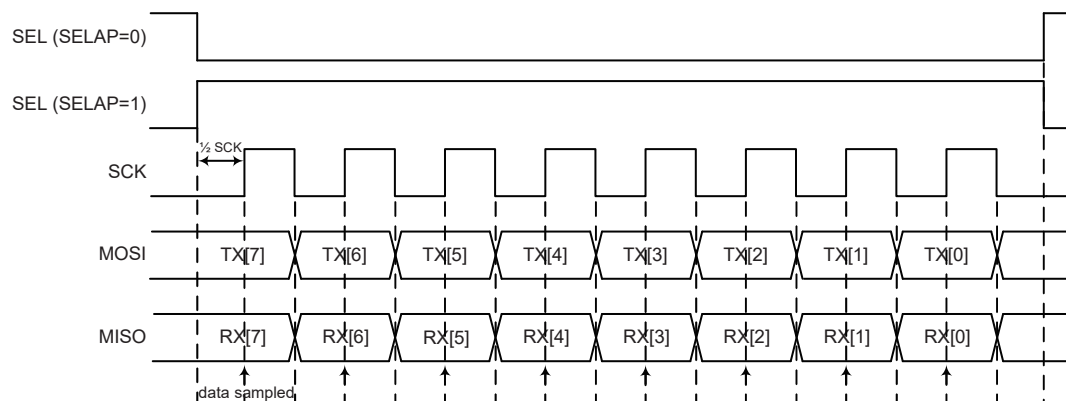


Figure 111. SPI Single Byte Transfer Timing Diagram – CPOL = 0, CPHA = 0

Figure 111 shows the continuous data transfer timing diagram of this format. Note that the SEL signal must change to an inactive level between each data frame.

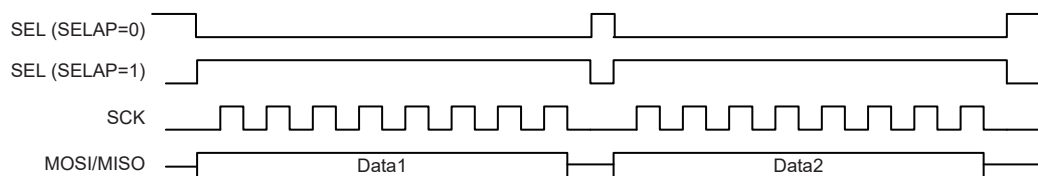


Figure 112. SPI Continuous Data Transfer Timing Diagram – CPOL = 0, CPHA = 0

CPOL = 0, CPHA = 1

In this format, the received data is sampled on the SCK line falling edge while the transmitted data is changed on the SCK line rising edge. In the master mode, the first bit is driven when data is written into the SPIDR register. In the slave mode, the first bit is driven at the first SCK clock rising edge. Figure 112 shows the single data byte transfer timing.

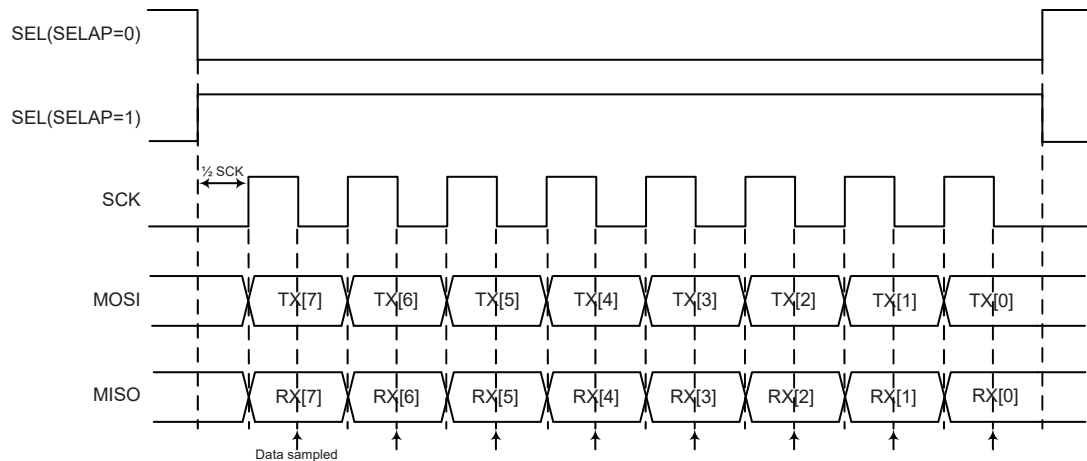


Figure 113. SPI Single Byte Transfer Timing Diagram – CPOL = 0, CPHA = 1

Figure 113 shows the continuous data transfer diagram timing. Note that the SEL signal must remain active until the last data transfer has completed.

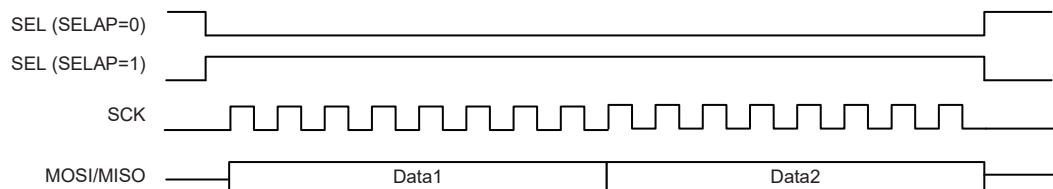


Figure 114. SPI Continuous Transfer Timing Diagram – CPOL = 0, CPHA = 1

CPOL = 1, CPHA = 0

In this format, the received data is sampled on the SCK line falling edge while the transmitted data is changed on the SCK line rising edge. In the master mode, the first bit is driven when data is written into the SPIDR register. In the slave mode, the first bit is driven when the SEL signal changes to an active level. Figure 114 shows the single byte transfer timing of this format.

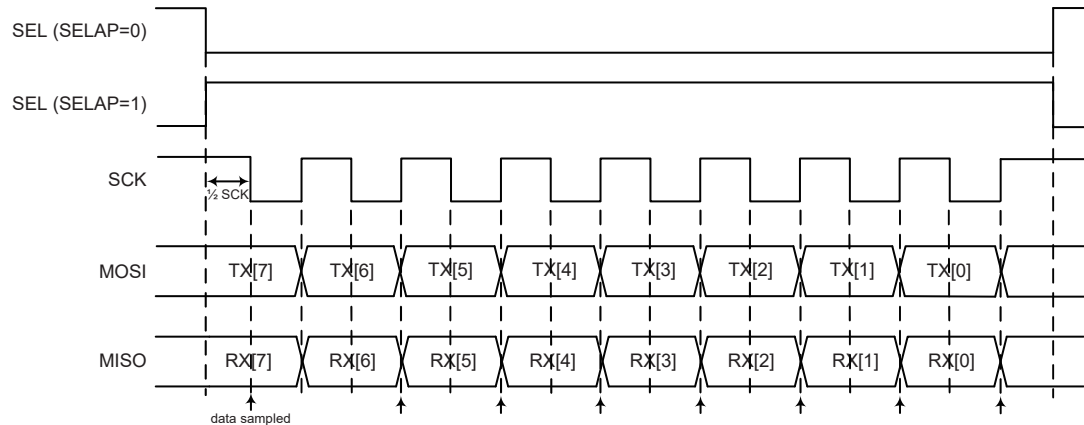


Figure 115. SPI Single Byte Transfer Timing Diagram – CPOL = 1, CPHA = 0

Figure 115 shows the continuous data transfer timing of this format. Note that the SEL signal must change to an inactive level between each data frame.

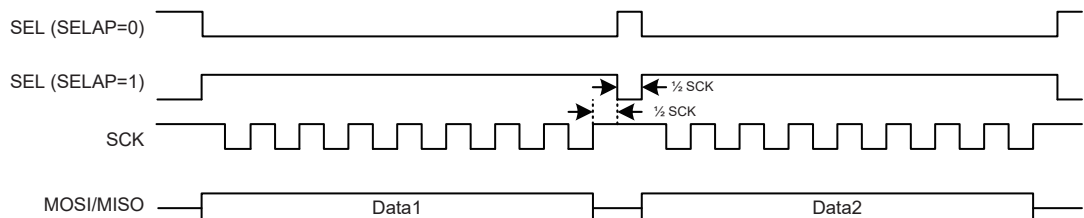


Figure 116. SPI Continuous Transfer Timing Diagram – CPOL = 1, CPHA = 0

CPOL = 1, CPHA = 1

In this format, the received data is sampled on the SCK line rising edge while the transmitted data is changed on the SCK line falling edge. In the master mode, the first bit is driven when data is written into the SPIDR register. In the slave mode, the first bit is driven at the first SCK falling edge. Figure 116 shows the single byte transfer timing of this format.

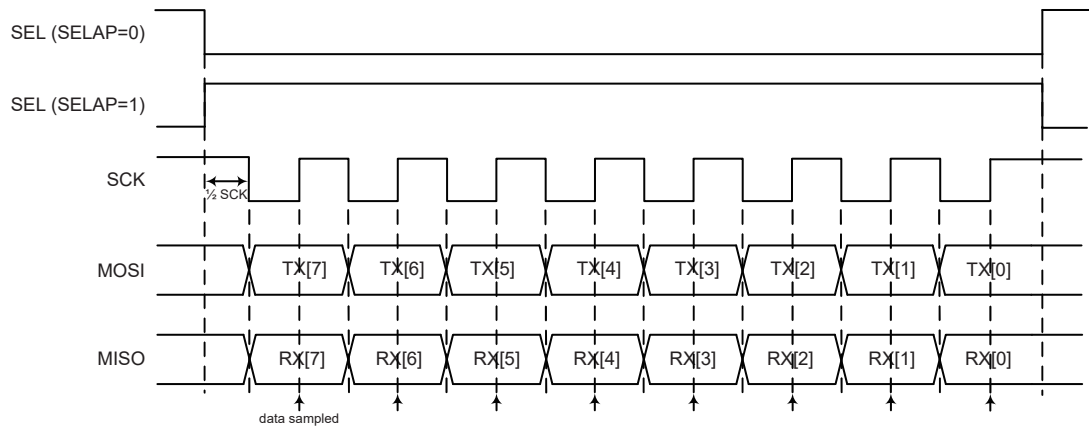


Figure 117. SPI Single Byte Transfer Timing Diagram – CPOL = 1, CPHA = 1

Figure 117 shows the continuous data transfer timing of this format. Note that the SEL signal must remain active until the last data transfer has completed.

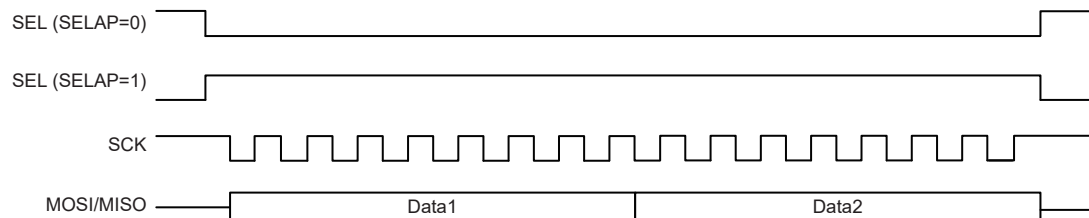


Figure 118. SPI Continuous Transfer Timing Diagram – CPOL = 1, CPHA = 1

Status Flags

TX Buffer Empty – TXBE

This TXBE flag is set when the TX buffer is empty in the non-FIFO mode or when the TX FIFO data length is equal to or less than the TX FIFO threshold level as defined by the TXFTLS field in the SPIFCR register in the FIFO mode. The following data to be transmitted can then be loaded into the buffer again. After this, the TXBE flag will be reset when the TX buffer already contains new data in the non-FIFO mode or when the TX FIFO data length is greater than the TX FIFO threshold level determined by the TXFTLS field in FIFO mode.

Transmission Register Empty – TXE

This TXE flag is set when both the TX buffer and the TX shift registers are empty. It will be reset when the TX buffer or the TX shift register contains new transmitted data.

RX Buffer Not Empty – RXBNE

This RXBNE flag is set when there is valid received data in the RX buffer in the non-FIFO mode or the RX FIFO data length is equal to or greater than the RX FIFO threshold level as defined by the RXFTLS field in the SPIFCR register in the SPI FIFO mode. This flag will be automatically cleared by hardware when the received data have been read out from the RX buffer totally in the non-FIFO mode or when the RX FIFO data length is less than the RX FIFO threshold level set in the RXFTLS field.

Time Out Flag – TO

The time out function is only available in the SPI FIFO mode and is disabled by loading a zero value into the TOC field in the Time Out Counter register. The time out counter will start counting if the SPI RX FIFO is not empty, once data is read from the SPIDR register or new data is received, the time out counter will be reset to 0 and count again. When the time out counter value is equal to the value specified by the TOC field in the SPIFTOCR register, the TO flag will be set. The flag is cleared by writing 1 to this bit.

Mode Fault – MF

The mode fault flag can be used to detect SPI bus usage in the SPI multi-master mode. For the multi-master mode, the SPI module is configured as a master device and the SEL signal is set as an input signal. The mode fault flag is set when the SPI SEL pin is suddenly changed to an active level by another SPI master. This means that another SPI master is requesting to use the SPI bus. Therefore, when an SPI mode fault occurs, it will force the SPI module to operate in the slave mode and also disable all of the SPI interface signals to avoid SPI bus signal collisions. For the same reason, if the SPI master wants to transfer data, it also needs to inform other SPI masters by driving their SEL signals to an active state. The detailed configuration diagram for the SPI multi-master mode is shown in the following figure.

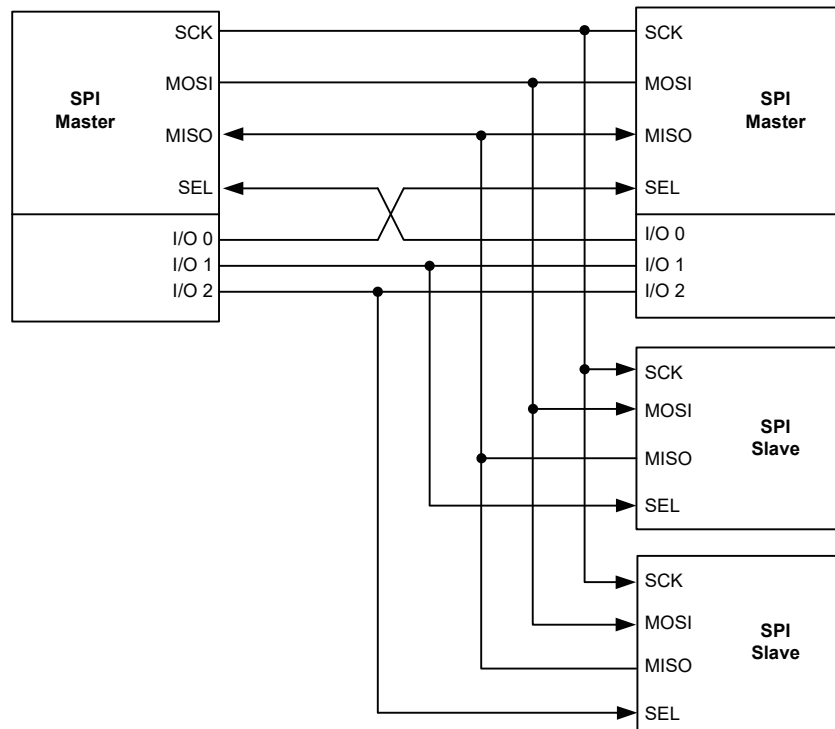


Figure 119. SPI Multi-Master Slave Environment

Table 45. SPI Mode Fault Trigger Conditions

Mode Fault	Descriptions
Trigger Condition	<ol style="list-style-type: none"> 1. SPI Master mode. 2. SELOEN = 0 in the SPICR0 register – SEL pin is configured to be the input mode. 3. SEL signal changes to an active level when driven by the external SPI master.
SPI Behavior	<ol style="list-style-type: none"> 1. Mode fault flag is set. 2. The SPIEN bit in the SPICR0 register is reset. This disables the SPI interface and blocks all output signals from the device. 3. The MODE bit in the SPICR1 register is reset. This forces the device into slave mode.

Table 46. SPI Master Mode SEL Pin Status

	SEL as Input – SELOEN = 0		SEL as Output – SELOEN = 1	
Multi-Master	Supported		Not supported	
SPI SEL Control Signal	Use Another GPIO to replace the SEL pin function		SEL pin in hardware or software control mode – using SELM setting	
Continuous Transfer	Case 1	Case 2	Case 1	Case 2
	Not supported	Supported	Hardware control	Hardware or software control

Case 1: SEL signal must be inactive between each data transfer.

Case 2: SEL signal will not to be inactive until the last data frame has finished.

Note: When the SPI is in the slave mode, the SEL signal is always an input and not affected by the SELOEN bit in the SPICR0 register.

Write Collision – WC

The following conditions will assert the Write Collision Flag.

- The FIFOEN bit in the SPIFCR register is cleared
The write collision flag is asserted when new data is written into the SPIDR register while both the TX buffer and the shift register are already full. Any new data written into the TX buffer will be lost.
- The FIFOEN bit in the SPIFCR register is set
The write collision flag is asserted to indicate that new data is written into the SPIDR register while both the TX FIFO and the TX shift register are already full. Any new data written into the TX FIFO will be lost.

Read Overrun – RO

- The FIFOEN bit in the SPIFCR register is cleared
The read overrun flag is asserted to indicate that both the RX shift register and the RX buffer are already full, if one more data is received. This will result in the newly received data not being shifted into the SPI shift register. As a result the latest received data will be lost.
- The FIFOEN bit in the SPIFCR register is set

The read overrun flag is set to indicate that the RX shift register and the RX FIFO are both full, if one more data is received. This means that the latest received data can not be shifted into the SPI shift register. As a result the latest received data will be lost.

Slave Abort – SA

In the SPI slave mode, the slave abort flag is set to indicate that the SEL pin suddenly changed to an inactive state during the reception of a data frame transfer. The data frame length is set by the DFL field in the SPICR1 register.

PDMA Interface

The PDMA interface is integrated in the SPI module. The PDMA function can be enabled by setting the TXDMAE or RXDMAE bit to 1 in the transmitter or receiver mode respectively. When the transmit buffer empty flag, TXBE, is asserted and the TXDMAE bit is set to 1, the PDMA function will be activated to move data from the memory location that users designated into the SPI data register or the TX FIFO until the TXBE flag is cleared to 0. The TXBE flag will be asserted when the transmit buffer is empty in the non-FIFO mode or the data contained in the TX FIFO is equal to or less than the level defined by the TXFTLS field in the FIFO mode.

Similarly, when the receive buffer not empty flag, RXBNE, is asserted and the RXDMAE bit is set to 1, the PDMA function will be activated to move data from the SPI data register or the RX FIFO to the memory location that users designated until the RXBNE flag is cleared to 0. The RXBNE flag will be asserted when the receive buffer is not empty in the non-FIFO mode or the data contained in the RX FIFO is equal to or greater than the level defined by the RXFTLS field in the FIFO mode.

For a more detailed description about the PDMA configurations, refer to the PDMA chapter.

Register Map

The following table shows the SPI registers and reset values.

Table 47. SPI Register Map

Register	Offset	Description	Reset Value
SPICR0	0x000	SPI Control Register 0	0x0000_0000
SPICR1	0x004	SPI Control Register 1	0x0000_0000
SPIIER	0x008	SPI Interrupt Enable Register	0x0000_0000
SPICPR	0x00C	SPI Clock Prescaler Register	0x0000_0000
SPIDR	0x010	SPI Data Register	0x0000_0000
SPISR	0x014	SPI Status Register	0x0000_0003
SPIFCR	0x018	SPI FIFO Control Register	0x0000_0000
SPIFSR	0x01C	SPI FIFO Status Register	0x0000_0000
SPIFTOCR	0x020	SPI FIFO Time Out Counter Register	0x0000_0000

Register Descriptions

SPI Control Register 0 – SPICR0

This register specifies the SEL control and the SPI enable bits.

Offset: 0x000

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	SELHT				GUADT				
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	GUADTEN	DUALEN	Reserved	SSELC	SELOEN	RXDMAE	TXDMAE	SPIEN	
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:12]	SELHT	Chip Select Hold Time 0x0: 1/2 SCK 0x1: 1 SCK 0x2: 3/2 SCK 0x3: 2 SCK Note that SELHT is for master mode only.

Bits	Field	Descriptions
[11:8]	GUADT	Guard Time GUADTEN = 1 0x0: 1 SCK 0x1: 2 SCK 0x2: 3 SCK ... Note that GUADT is for master mode only.
[7]	GUADTEN	Guard Time Enable 0: Guard Time is 1/2 SCK 1: When this bit is set, guard time can be controlled by GUADT Note that GUADTEN is for master mode only.
[6]	DUALEN	Dual Port Enable 0: Dual port is disabled 1: Dual port is enabled The control bit is used to support the dual output read mode of the series SPI NOR Flash. When this bit is set and the MOSI signal will change the direction from output to input and receive the series data stream. That means the DUALEN control bit is only for master mode.
[4]	SSELC	Software Slave Select Control 0: Set the SEL output to an inactive state 1: Set the SEL output to an active state The application software can set the SEL output to an active or inactive state by configuring the SSELC bit. The active level is configured by the SELAP bit in the SPICR1 register. Note that the SSELC bit is only available when the SELOEN bit is set to 1 for enabling the SEL output meanwhile the SELM bit is cleared to 0 for controlling the SEL signal by software. Otherwise, the SSELC bit has no effect.
[3]	SELOEN	Slave Select Output Enable 0: Set the SEL signal to the input mode for multi-master mode 1: Set the SEL signal to the output mode for slave select The SELOEN is only available in the master mode to set the SEL signal as an input or output signal. When the SEL signal is configured to operate in the output mode, it is used as a slave select signal in either the hardware or software mode according to the SELM bit setting in the SPICR1 register. The SEL signal is used for mode fault detection in the multi-master environment when it is configured to operate in the input mode
[2]	RXDMAE	RX PDMA request enable 0: SPI RX path PDMA request is disabled 1: SPI RX path PDMA request is enabled
[1]	TXDMAE	TX PDMA request enable 0: SPI TX path PDMA request is disabled 1: SPI TX path PDMA request is enabled
[0]	SPIEN	SPI Enable 0: SPI interface is disabled 1: SPI interface is enabled

SPI Control Register 1 – SPICR1

This register specifies the SPI parameters including the data length, the transfer format, the SEL active polarity/mode, the LSB/MSB control, and the master/slave mode.

Offset: 0x004

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved	MODE	SELM	FIRSTBIT	SELAP	FORMAT		
		RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0
Type/Reset	Reserved				DFL			
					RW	0	RW	0

Bits	Field	Descriptions
[14]	MODE	Master or Slave Mode 0: Slave mode 1: Master mode
[13]	SELM	Slave Select Mode 0: SEL signal is controlled by software – asserted or de-asserted by the SSEL bit 1: SEL signal is controlled by hardware – generated automatically by the SPI hardware Note that the SELM bit is available for master mode only, i.e., MODE = 1.
[12]	FIRSTBIT	LSB or MSB Transmitted First 0: MSB is transmitted first 1: LSB is transmitted first
[11]	SELAP	Slave Select Active Polarity 0: SEL signal is active low 1: SEL signal is active high

Bits	Field	Descriptions																		
[10:8]	FORMAT	<p>SPI Data Transfer Format</p> <p>These three bits are used to determine the data transfer format of the SPI interface.</p> <table border="1"> <thead> <tr> <th>FORMAT [2:0]</th><th>CPOL</th><th>CPHA</th></tr> </thead> <tbody> <tr> <td>001</td><td>0</td><td>0</td></tr> <tr> <td>010</td><td>0</td><td>1</td></tr> <tr> <td>110</td><td>1</td><td>0</td></tr> <tr> <td>101</td><td>1</td><td>1</td></tr> <tr> <td>Others</td><td colspan="2">Reserved</td></tr> </tbody> </table> <p>CPOL: Clock Polarity 0: SCK Idle state is low 1: SCK Idle state is high</p> <p>CPHA: Clock Phase 0: Data is captured on the first SCK clock edge 1: Data is captured on the second SCK clock edge</p>	FORMAT [2:0]	CPOL	CPHA	001	0	0	010	0	1	110	1	0	101	1	1	Others	Reserved	
FORMAT [2:0]	CPOL	CPHA																		
001	0	0																		
010	0	1																		
110	1	0																		
101	1	1																		
Others	Reserved																			
[3:0]	DFL	<p>Data Frame Length</p> <p>Selects the data transfer frame from 1 bit to 16 bits.</p> <p>0x1: 1 bit 0x2: 2 bits ... 0xF: 15 bits 0x0: 16 bits</p>																		

SPI Interrupt Enable Register – SPIIER

This register contains the corresponding SPI interrupt enable control bit.

Offset: 0x008

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	TOIEN	SAIEN	MFIEN	ROIEN	WCIEEN	RXBNEIEN	TXEIEEN	TXBEIEN
	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[7]	TOIEN	Time Out Interrupt Enable 0: Disable 1: Enable
[6]	SAIEN	Slave Abort Interrupt Enable 0: Disable 1: Enable
[5]	MFIEN	Mode Fault Interrupt Enable 0: Disable 1: Enable
[4]	ROIEN	Read Overrun Interrupt Enable 0: Disable 1: Enable
[3]	WCIEEN	Write Collision Interrupt Enable 0: Disable 1: Enable
[2]	RXBNEIEN	RX Buffer Not Empty Interrupt Enable 0: Disable 1: Enable An interrupt is generated when the RXBNE flag is set and RXBNEIEN is set. In the FIFO mode, the interrupt being generated depends upon the RX FIFO trigger level setting.
[1]	TXEIEEN	Transmission Register Empty Interrupt Enable 0: Disable 1: Enable The transmission register empty interrupt request will be generated when the TXE flag and the TXEIEEN bit are set.

Bits	Field	Descriptions
[0]	TXBEIEN	TX Buffer Empty Interrupt Enable 0: Disable 1: Enable The TX buffer empty interrupt request will be generated when the TXBE flag and the TXBEIEN bit are set. In the FIFO mode, the interrupt request being generated depends upon the TX FIFO trigger level setting.

SPI Clock Prescaler Register – SPICPR

This register specifies the SPI clock prescaler ratio.

Offset: 0x00C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	CP							
	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
Type/Reset	CP							
	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[15:0]	CP	SPI Clock Prescaler The SPI clock (SCK) is determined by the following equation: $f_{SCK} = f_{PCLK} / (2 \times (CP + 1))$, where the CP ranges is from 0 to 65535 Note: For the SPI slave mode, the system clock (f_{PCLK}) must be at least 3 times faster than the external SPI SCK input.

SPI Data Register – SPIDR

This register stores the SPI received or transmitted Data.

Offset: 0x010

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	DR								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	DR								
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:0]	DR	<p>Data Register</p> <p>The SPI data register is used to store the serial bus transmitted or received data. In the non-FIFO mode, writing data into the SPI data register will also load the data into the data transmission buffer, known as the TX buffer. Reading data from the SPI data register will return the data held in the data received buffer, named RX buffer.</p>

This register contains the relevant SPI status.

Reset value: 0x0000 0003

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	Reserved								BUSY
	7	6	5	4	3	2	1	0	
Type/Reset	TO	SA	MF	RO	WC	RXBNE	TXE	TXBE	
	WC	0	WC	0	WC	0	WC	0	1
	WC	0	WC	0	WC	0	RO	0	RO
	WC	0	WC	0	WC	0	RO	1	RO
	WC	0	WC	0	WC	0	RO	1	RO

18 Serial Peripheral Interface (SPI)

Bits	Field	Descriptions
[2]	RXBNE	RX Buffer Not Empty flag 0: RX buffer is empty 1: RX buffer is not empty This bit indicates the RX buffer status in the non-FIFO mode. It is also used to indicate if the RX FIFO trigger level has been reached in the FIFO mode. This bit will be cleared when the SPI RX buffer is empty in the non-FIFO mode or if the number of data contained in RX FIFO is less than the trigger level which is specified by the RXFTLS field in the SPIFCR register in the SPI FIFO mode.
[1]	TXE	Transmission Register Empty flag 0: TX buffer or TX shift register is not empty 1: TX buffer and TX shift register both are empty
[0]	TXBE	TX Buffer Empty flag 0: TX buffer is not empty 1: TX buffer is empty In the FIFO mode, this bit if set indicates that the number of data contained in TX FIFO is equal to or less than the trigger level specified by the TXFTLS field in the SPIFCR register.

SPI FIFO Control Register – SPIFCR

This register contains the related SPI FIFO control including the FIFO enable control and the FIFO trigger level selections.

Offset: 0x018

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved					FIFOEN	Reserved	
	7	6	5	4	3	2	1	0
Type/Reset	RXFTLS				TXFTLS			
	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[10]	FIFOEN	FIFO Enable 0: FIFO is disabled 1: FIFO is enabled This bit can not be set or reset when the SPI interface is in transmitting.

Bits	Field	Descriptions
[7:4]	RXFTLS	RX FIFO Trigger Level Select 0000: Trigger level is 0 0001: Trigger level is 1 ... 1000: Trigger level is 8 Others: Reserved The RXFTLS field is used to specify the RX FIFO trigger level. When the number of data contained in the RX FIFO is equal to or greater than the trigger level defined by the RXFTLS field, the RXBNE flag will be set
[3:0]	TXFTLS	TX FIFO Trigger Level Select 0000: Trigger level is 0 0001: Trigger level is 1 ... 1000: Trigger level is 8 Others: Reserved The TXFTLS field is used to specify the TX FIFO trigger level. When the number of data contained in the TX FIFO is equal to or less than the trigger level defined by the TXFTLS field, the TXBE flag will be set.

SPI FIFO Status Register – SPIFSR

This register contains the relevant SPI FIFO status.

Offset: 0x01C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	RXFS				TXFS			
	RO	0	RO	0	RO	0	RO	0

Bits	Field	Descriptions
[7:4]	RXFS	RX FIFO Status 0000: RX FIFO empty 0001: RX FIFO contains 1 data ... 1000: RX FIFO contains 8 data Others: Reserved

Bits	Field	Descriptions
[3:0]	TXFS	TX FIFO Status 0000: TX FIFO empty 0001: TX FIFO contains 1 data ... 1000: TX FIFO contains 8 data Others: Reserved

SPI FIFO Time Out Counter Register – SPIFCR

This register stores the SPI RX FIFO time out counter value.

Offset: 0x020

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	TOC								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	TOC								
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:0]	TOC	Time Out Counter The time out counter starts to count from 0 after the SPI RX FIFO receives a data, and the counter value is reset once the data is read from the SPIDR register by software or another new data is received. If the FIFO does not receive new data or the software does not read data from the SPIDR register the time out counter value will continuously increase. When the time out counter value is equal to the TOC setting value, the TO flag in the SPISR register will be set and an interrupt will be generated if the TOIEN bit in the SPIIER register is set. The time out counter will be stopped when the RX FIFO is empty. The SPI FIFO time out function can be disabled by setting the TOC field to zero. The time out counter is driven by the system APB clock, named f_{PCLK} .

19 Universal Synchronous Asynchronous Receiver Transmitter (USART)

Introduction

The Universal Synchronous Asynchronous Receiver Transceiver, USART, provides a flexible full duplex data exchange using synchronous or asynchronous transfer. The USART is used to translate data between parallel and serial interfaces, and is also commonly used for RS232 standard communication. The USART peripheral function supports a variety of interrupts.

The USART module includes an 8-level transmit FIFO, TX FIFO, and a 8-level receive FIFO, RX FIFO. Software can detect a USART error status by reading the USART Status & Interrupt Flag Register, USRSIFR. The status includes the condition of the transfer operations as well as several error conditions resulting from Parity, Overrun, Framing and Break events.

The USART includes a programmable baud rate generator which is capable of dividing the USART clock CK_USART to produce a baud rate clock for the USART transmitter and receiver.

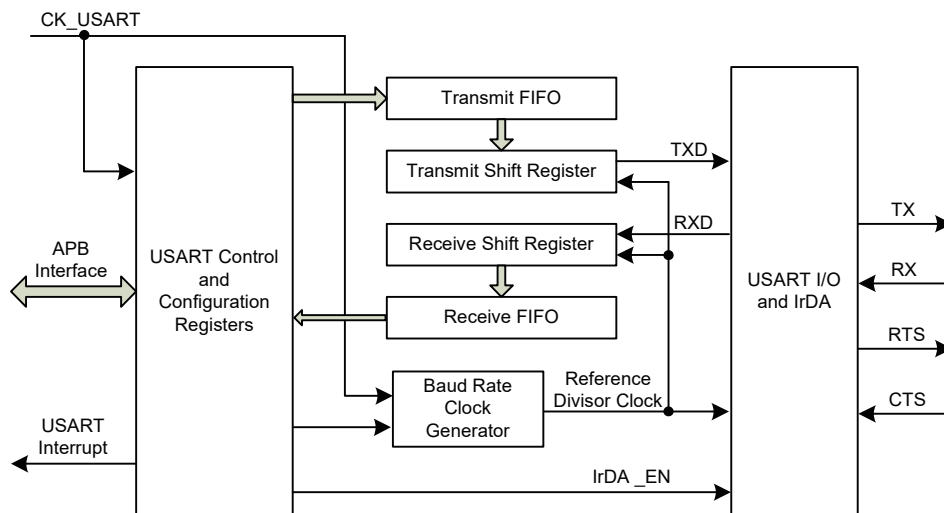


Figure 120. USART Block Diagram

Features

- Supports both asynchronous and clocked synchronous serial communication modes
- Full Duplex Communication Capability
- Programmable baud rate clock frequency up to ($f_{\text{PLCK}}/16$) MHz for asynchronous mode and ($f_{\text{PLCK}}/8$) MHz for synchronous mode
- IrDA SIR encoder and decoder
 - Support normal 3/16 bit duration and low-power (1.41 ~ 2.23 μs) durations
- Supports RS485 mode with output enable
- Auto hardware flow control mode – RTS, CTS
- Fully programmable serial communication functions including:
 - Word length: 7, 8 or 9-bit character
 - Parity: Even, odd, or no-parity bit generation and detection
 - Stop bit: 1 or 2 stop bits generation
 - Bit order: LSB-first or MSB-first transfer
- Error detection: Parity, overrun and frame error
- FIFO:
 - Receive FIFO: 8-level (max. 9 data bits)
 - Transmit FIFO: 8-level (max. 9 data bits)
- Supports PDMA Interface

Functional Descriptions

Serial Data Format

The USART module performs a parallel-to-serial conversion on data that is written to the transmit FIFO registers and then sends the data with the following format: Start bit, 7 ~ 9 LSB/MSB first data bits, optional Parity bit and finally 1 ~ 2 Stop bits. The Start bit has the opposite polarity of the data line idle state. The Stop bit is the same as the data line idle state and provides a delay before the next start situation. Both the Start and Stop bits are used for data synchronization during the asynchronous data transmission.

The USART module also performs a serial-to-parallel conversion on the data that is read from the receive FIFO registers. It will first check the Parity bit and will then look for a Stop bit. If the Stop bit is not found, the USART module will consider the entire word transmission as failed and respond with a Framing Error.

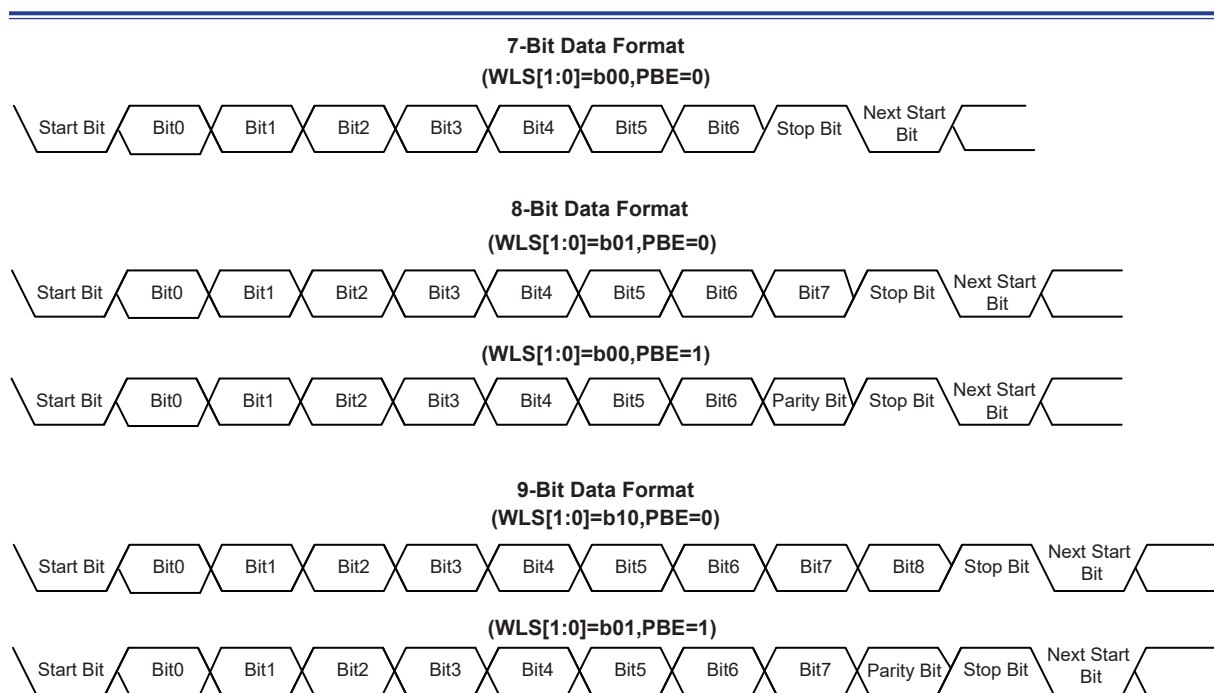


Figure 121. USART Serial Data Format

Baud Rate Generation

The baud rate for the USART receiver and transmitter are both set with the same values. The baud rate divisor, BRD, has the following relationship with the USART clock which is known as CK_USART.

$$\text{Baud Rate Clock} = \text{CK_USART} / \text{BRD}$$

Where the CK_USART clock is the APB clock connected to the USART while the BRD range is from 16 to 65535 for asynchronous mode and 8 to 65535 for synchronous mode.

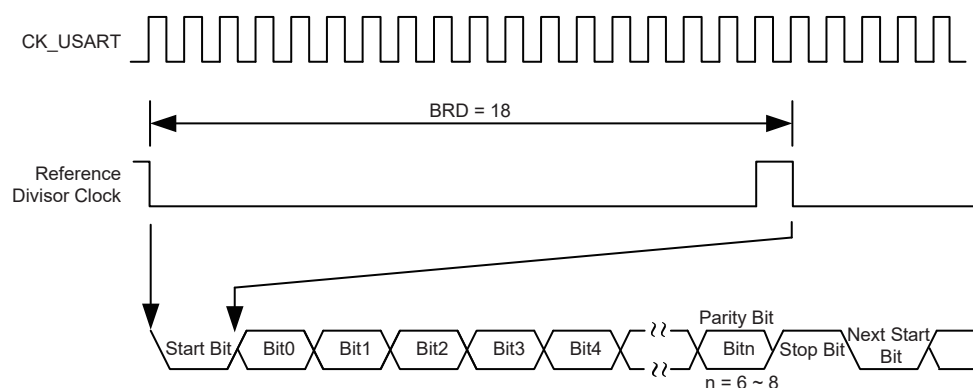


Figure 122. USART Clock CK_USART and Data Frame Timing

Table 48. Baud Rate Deviation Error Calculation – CK_USART = 40 MHz

Baud Rate		CK_USART = 40 MHz		
No.	Kbps	Actual	BRD	Deviation Error Rate
1	2.4	2.4	16667	0.00%
2	9.6	9.6	4167	-0.01%
3	19.2	19.2	2083	0.02%
4	57.6	57.6	694	0.06%
5	115.2	115.3	347	0.06%
6	230.4	229.9	174	-0.22%
7	460.8	459.8	87	-0.22%
8	921.6	930.2	43	0.94%
9	2250	2222.2	18	-1.23%
10	3000	—	—	—

Table 49. Baud Rate Deviation Error Calculation – CK_USART = 48 MHz

Baud Rate		CK_USART = 48 MHz		
No.	Kbps	Actual	BRD	Deviation Error Rate
1	2.4	2.4	20000	0.00%
2	9.6	9.6	5000	0.00%
3	19.2	19.2	2500	0.00%
4	57.6	57.6	833	0.04%
5	115.2	115.1	417	-0.08%
6	230.4	230.8	208	0.16%
7	460.8	461.5	104	0.16%
8	921.6	923.1	52	0.16%
9	2250	2285.7	21	1.59%
10	3000	3000.0	16	0.00%

Table 50. Baud Rate Deviation Error Calculation – CK_USART = 60 MHz

Baud Rate		CK_USART = 60 MHz		
No.	Kbps	Actual	BRD	Deviation Error Rate
1	2.4	2.4	25000	0.00%
2	9.6	9.6	6250	0.00%
3	19.2	19.2	3125	0.00%
4	57.6	57.6	1042	-0.03%
5	115.2	115.2	521	-0.03%
6	230.4	230.8	260	0.16%
7	460.8	461.5	130	0.16%
8	921.6	923.1	65	0.16%
9	2250	2222.2	27	-1.23%
10	3000	3000.0	20	0.00%

Hardware Flow Control

The USART supports the hardware flow control function which is enabled by setting the HFCEN bit in the USRCR register to 1. It is possible to control the serial data flow between two USART devices by using the CTS input and the RTS output. The Figure 122 shows the connection diagram in this mode. The hardware flow control function is categorized into two types. One is the RTS flow control function and the other is the CTS flow control function.

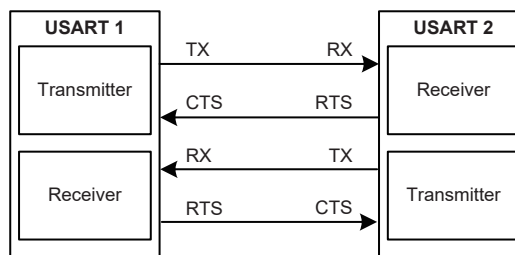


Figure 123. Hardware Flow Control between 2 USARTs

RTS Flow Control

In the RTS flow control, the USART RTS pin is active with a logic low state when the receive data register is empty. It means that the receiver is ready to receive a new data. When the RX FIFO reaches the trigger level which is specified by configuring the RXTL field in the USRFCR register, the USART RTS pin is inactive with a logic high state. Figure 123 shows the example of RTS flow control.

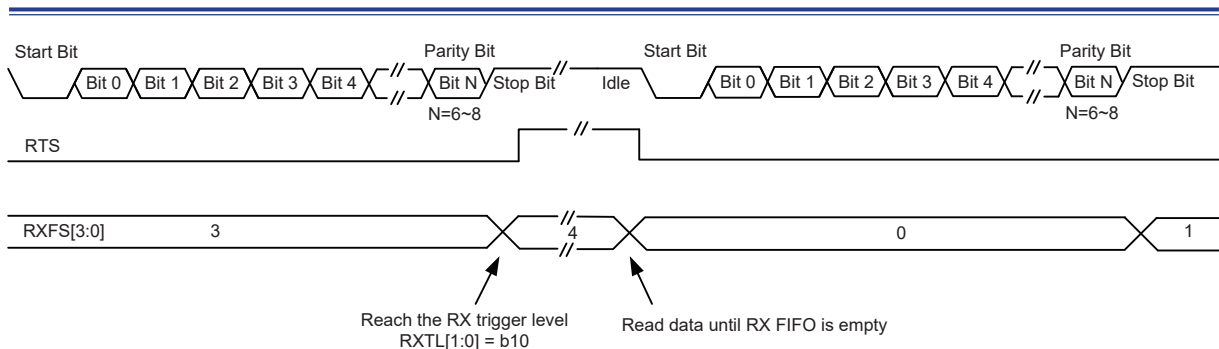


Figure 124. USART RTS Flow Control

CTS Flow Control

If the hardware flow control function is enabled, the URTXEN bit in the USRCR register is controlled by the USART CTS input signal. If the USART CTS pin is forced to a logic low state, the URTXEN bit will automatically be set to 1 to enable the data transmission. However, if the USART CTS pin is forced to a logic high state, the URTXEN bit will be cleared to 0 and then the data transmission will also be disabled.

When the USART CTS pin is forced to a logic high state during a data transmission period, the current data transmission will be continued until the stop bit is completed. The Figure 124 shows an example of communication with CTS flow control.

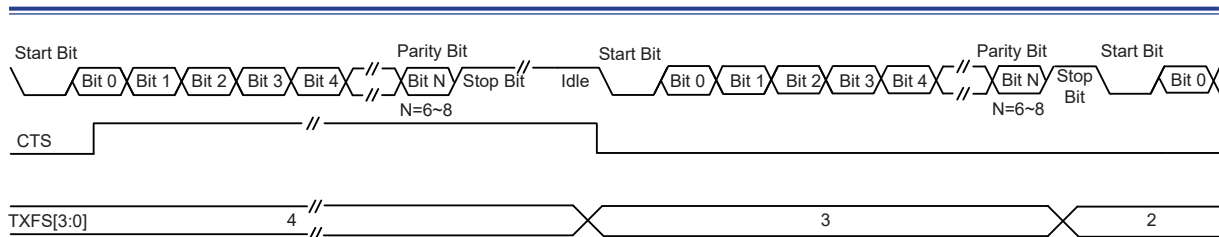


Figure 125. USART CTS Flow Control

IrDA

The USART IrDA mode is provided for half-duplex point-to-point wireless communication.

The USART module includes an integrated modulator and demodulator which allow a wireless communication using infrared transceivers. The transmitter specifies a logic data '0' as a 'high' pulse and a logic data '1' as a 'low' level while the Receiver specifies a logic data '0' as a 'low' pulse and a logic data '1' as 'high' level in the IrDA mode.

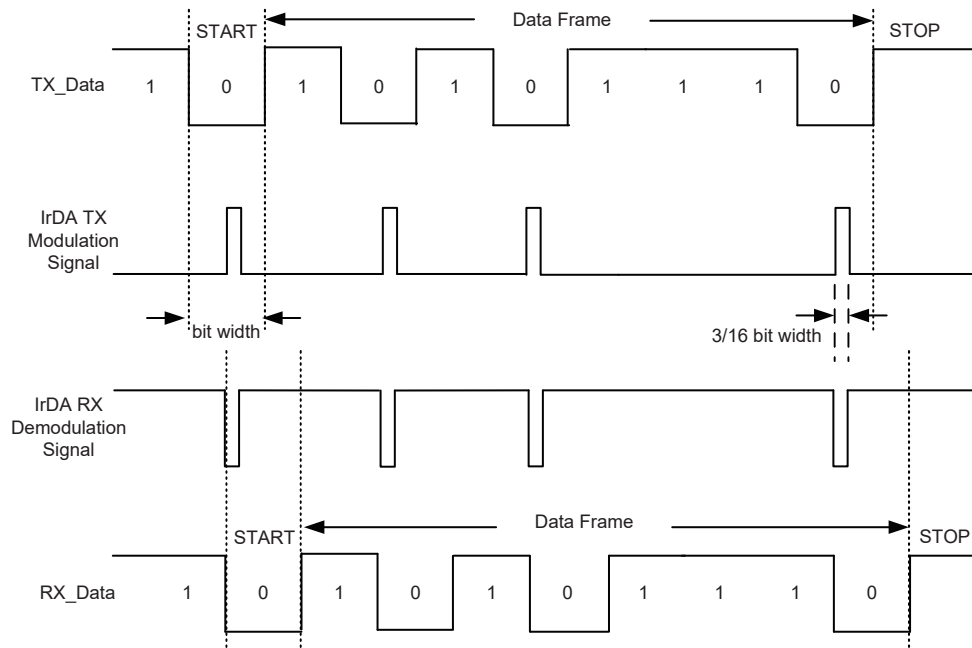


Figure 126. IrDA Modulation and Demodulation

The IrDA mode provides two operation modes, one is the normal mode, and the other is the low-power mode.

IrDA Normal Mode

For the IrDA normal mode, the width of each transmitted pulse generated by the transmitter modulator is specified as 3/16 of the baud rate clock period. The receiver pulse width for the IrDA receiver demodulator is based on the IrDA receive debounce filter which is implemented using an 8-bit down-counting counter. The debounce filter counter value is specified by the IrDAPSC field in the IrDACR register. When a falling edge is detected on the receiver pin, the debounce filter counter starts to count down, driven by the CK_USART clock. If a rising edge is detected on the receiver pin, the counter stops counting and is reloaded with the IrDAPSC value. When a low pulse falling edge on the receiver pin is detected and then before the debounce filter has counted down to zero, a rising edge is also detected, then this low pulse will be considered as glitch noise and will be discarded. If a low pulse falling edge appears on the receiver pin but no rising edge is detected before the debounce counter reaches 0, then the input is regarded as a valid data “0” for this bit duration. The IrDAPSC value must be set to be greater than or equal to 0x01, then the IrDA receiver demodulation operation can function properly. The IrDAPSC value can be adjusted to meet the USART baud rate setting to filter the IrDA received glitch noise of which the width is smaller than the prescaler setting duration.

IrDA Low-Power Mode

In the IrDA low-power mode, the transmitted IrDA pulse width generated by the transmitter modulator is not kept at 3/16 of the baud rate clock period. Instead, the pulse width is fixed and is calculated by the following formula. The transmitted pulse width can be adjusted by the IrDAPSC field to meet the minimum pulse width specification of the external IrDA receiver device.

$$T_{\text{IrDA_L}} = 3 \times \text{IrDAPSC} / \text{CK_USART}$$

Note: $T_{\text{IrDA_L}}$ is the transmitted pulse width in the low-power mode.

The IrDAPSC field is the IrDACR prescaler value in the IrDA Control Register IrDACR.

The debounce behavior in the IrDA low-power receiving mode is similar to the IrDA normal mode. For glitch detection, the low pulse of which the pulse width is shorter than $1 \times (\text{IrDAPSC} / \text{CK_USART})$ should be discarded in the IrDA receiver demodulation. A valid low data is accepted if its low pulse width is greater than $2 \times (\text{IrDAPSC} / \text{CK_USART})$ duration.

The IrDA physical layer specification specifies a minimum delay with a value of 10 ms between the transmission and reception switch; and this IrDA receiver set-up time also should be managed by the software.

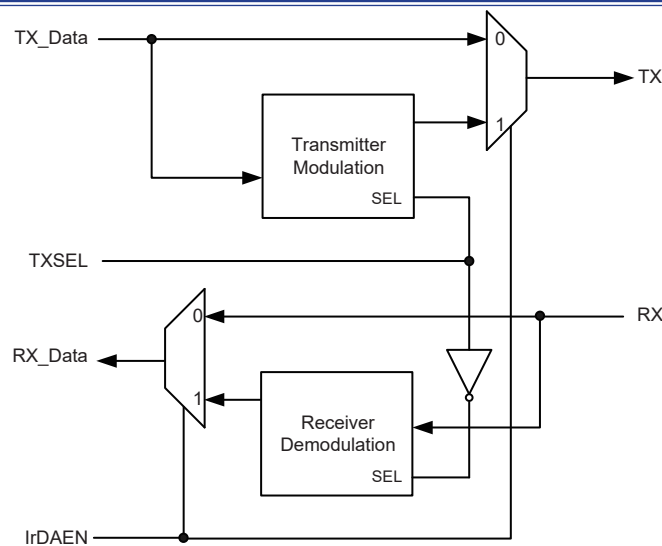


Figure 127. USART I/O and IrDA Block Diagram

RS485 Mode

The RS485 mode of USART provides the data transmission on interface transmitted over a 2-wire twisted pair bus. The RS485 transceiver interprets the voltage levels of the differential signals with respect to a third common voltage. Without this common reference, the transceiver may interpret the differential signals incorrectly. This enhances the noise rejection capabilities of the RS485 interface. The USART RTS pin is used to control the external RS485 transceiver whose polarity can be selected by configuring the TXENP bit in the RS485 Control Register, named RS485CR, when the USART operates in the RS485 mode.

RS485 Auto Direction Mode – AUD

When the RS485 mode is configured as a master transmitter, it will operate in the Auto Direction Mode, AUD. In the AUD mode the polarity of the USART RTS pin is configurable according to the TXENP bit in the RS485 Control Register in the RS485 mode. This pin can be used to control the external RS485 transceiver to enable the transmitter.

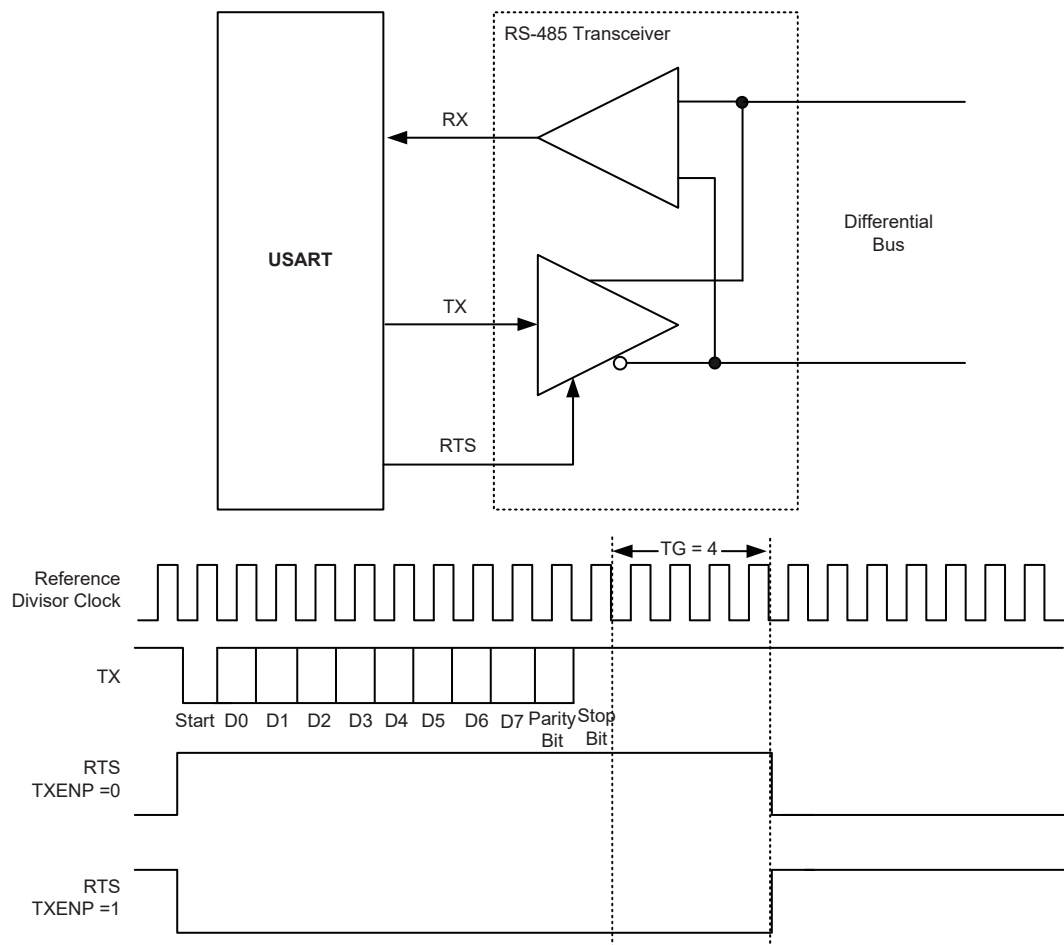


Figure 128. RS485 Interface and Waveform

RS485 Normal Multi-drop Operation Mode – NMM

When the RS485 mode is configured as an addressable slave, it will operate in the Normal Multi-drop Operation Mode, NMM. This mode is enabled when the RSNMM field is set in the RS485CR register. Regardless of the URRXEN value in the USRCR register, all the received data with a parity bit “0” will be ignored until the first address byte is detected with a parity bit “1” and then the received address byte will be stored in the RX FIFO. Once the first address data is detected and stored in the RX FIFO, the RSADD flag in the USRSIFR register will be set and generate an interrupt if the RSADDIE bit in the USRIER register is set to 1. Application software can determine whether the receiver is enabled or disabled to accept the following data by configuring the URRXEN bit. When the receiver is enabled by setting the URRXEN bit to 1, all received data will be stored in the RX FIFO. Otherwise, all received data will be ignored if the receiver is disabled by clearing the URRXEN bit to 0.

RS485 Auto Address Detection Operation Mode – AAD

Except in the Normal Multi-drop Operation Mode, the RS485 mode can operate in the Auto Address Detection Operation Mode, AAD, when it is configured as an addressable slave. This

mode is enabled by setting the RSAAD field to 1 in the RS485CR register. The receiver will detect the address frame with a parity bit “1” and then compare the received address data with the ADDMATCH field value which is a programmable 8-bit address value specified in the RS485CR register. If the address data matches the ADDMATCH value, it will be stored in the RX FIFO and the URRXEN bit will be automatically set. When the receiver is enabled, all received data will be stored in the RX FIFO until the next address frame does not match the ADDMATCH value and then the receiver will be automatically disabled. After the receiver is enabled, software can disable the receiver by setting the URRXEN bit to ‘0’.

Synchronous Master Mode

The data is transmitted in a full-duplex style in the USART Synchronous Master Mode, i.e., data transmission and reception both occur at the same time and only support master mode. The USART CTS pin is the synchronous USART transmitter clock output. In this mode, no clock pulses will be sent to the CTS pin during the start bit, parity bit and stop bit duration. The CPS bit in the Synchronous Control Register SYNCR, can be used to determine whether data is captured on the first or the second clock edge. The CPO bit in the SYNCR can be used to configure the clock polarity in the USART Synchronous Mode idle state. Detailed timing information is shown in the Figure 129.

In the USART synchronous Mode, the USART CTS/SCK clock output pin is only used to transmit the data to slave device. If the transmission data register USRDR, is written with valid data, the USART synchronous mode will automatically transmit this data with the corresponding clock output and the USART receiver will also receive data on the RX pin. Otherwise the receiver will not obtain synchronous data if no data is transmitted.

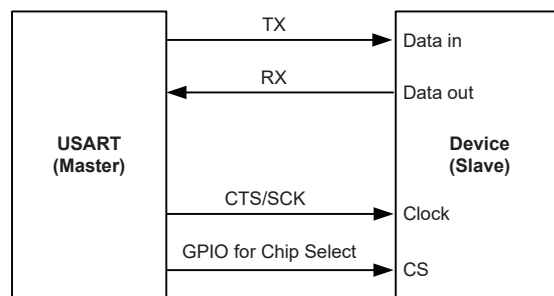


Figure 129. USART Synchronous Transmission Example

Note: The USART supports the synchronous master mode only: it can not receive or send data related to an input clock. The USART CTS/SCK clock is always an output.

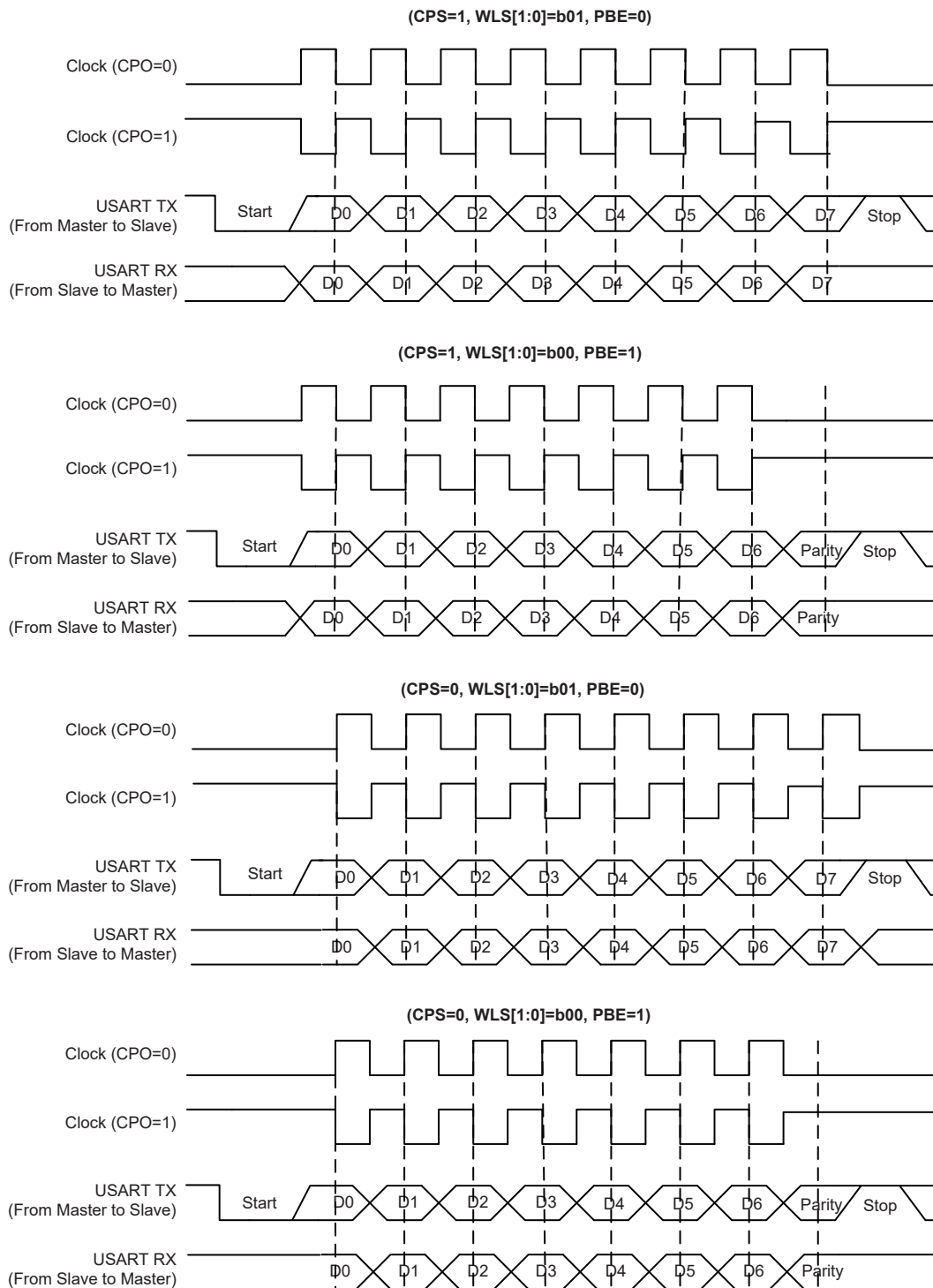


Figure 130. 8-bit Format USART Synchronous Waveform

Interrupts and Status

The USART can generate interrupts when the following events occur and the corresponding interrupt enable bits are set:

- Receive FIFO time-out interrupt: An interrupt is generated when the USART receive FIFO is not empty and does not receive a new data package during the specified time-out interval.
- Receiver line status interrupts: The interrupts are generated when the USART receiver overrun error, parity error, framing error and break events occur.
- Transmit FIFO threshold level interrupt: An interrupt is generated when the data to be transmitted in the USART Transmit FIFO is less than the specified threshold level.
- Transmit complete interrupt: An interrupt is generated when the Transmit FIFO is empty and the content of the transmit shift register (TSR) is also completely shifted.
- Receive FIFO threshold level interrupt: An interrupt is generated when the FIFO received data amount has reached the specified threshold level.

PDMA Interface

The PDMA interface is integrated in the USART. The PDMA function can be enabled by setting the TXDMAEN or RXDMAEN bit in the USRCR register to 1 in the transmit or receive mode respectively. When the data to be transmitted in the USART Transmit FIFO is less than the TX FIFO threshold level specified by the TXTL field in the USRFCR register and the TXDMAEN bit is set to 1, the PDMA function will be activated to move data from a source location into the USART TX FIFO.

Similarly, when the received data amount in the receive FIFO is equal to the RX FIFO threshold level specified by the RXTL field in the USRFCR register and the RXDMAEN bit is set to 1, the PDMA function will be activated to move data from the USART RX FIFO to a specific destination location. For a more detailed description about the PDMA configurations, refer to the PDMA chapter.

Register Map

The following table shows the USART registers and reset values.

Table 51. USART Register Map

Register	Offset	Description	Reset Value
USRDR	0x000	USART Data Register	0x0000_0000
USRCR	0x004	USART Control Register	0x0000_0000
USRFCR	0x008	USART FIFO Control Register	0x0000_0000
USRIER	0x00C	USART Interrupt Enable Register	0x0000_0000
USRSIFR	0x010	USART Status & Interrupt Flag Register	0x0000_0180
USRTPR	0x014	USART Timing Parameter Register	0x0000_0000
IrDACR	0x018	USART IrDA Control Register	0x0000_0000
RS485CR	0x01C	USART RS485 Control Register	0x0000_0000
SYNCR	0x020	USART Synchronous Control Register	0x0000_0000
USRDLR	0x024	USART Divider Latch Register	0x0000_0010
USRTSTR	0x028	USART Test Register	0x0000_0000

Register Descriptions

USART Data Register – USRDR

The register is used to access the USART transmitted and received FIFO data.

Offset: 0x000

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							DB
	7	6	5	4	3	2	1	0
Type/Reset	DB							
	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[8:0]	DB	<p>Reading data from this receiver buffer register will return the data from the receive FIFO. The receive FIFO has a capacity of up to 8×9 bits. By reading this register, the USART will return a 7, 8 and 9-bit received data. The DB field bit 8 is valid for the 9-bit mode only and is fixed at 0 for the 8-bit mode. For the 7-bit mode, the DB[6:0] field contains the available bits.</p> <p>Writing data to this buffer register will load data into the Transmit FIFO. The Transmit FIFO has a capacity of up to 8×9 bits. By writing to this register, the USART will send out 7, 8 or 9-bit transmitted data. The DB field bit 8 is valid for the 9-bit mode only and will be ignored for the 8-bit mode. For the 7-bit mode, the DB[6:0] field contains the available bits.</p>

USART Control Register – USRCR

The register specifies the serial parameters such as data length, parity and stop bit for the USART. It also contains the USART enable control bits together with the USART mode and data transfer mode selection.

Offset: 0x004

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	RTS	BCB	SPE	EPE	PBE	NSB	WLS	
	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
Type/Reset	RXDMAEN	TXDMAEN	URRXEN	URTXEN	HFCEN	TRSM	MODE	
	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[15]	RTS	Request-To-Send Signal 0: Drive USART RTS pin to logic 1 1: Drive USART RTS pin to logic 0 Note that the RTS bit is used to control the USART RTS pin status when the HFCEN bit is reset. When the HFCEN bit is set, this RTS bit is read only and indicates the pin status that is controlled by hardware flow control function.
[14]	BCB	Break Control Bit When this bit is set 1, the serial data output on the USART TX pin will be forced to the Spacing State (logic 0). This bit acts only on the USART TX output pin and has no effect on the transmitter logic.
[13]	SPE	Stick Parity Enable 0: Disable stick parity 1: Stick Parity bit is transmitted This bit is only available when the PBE bit is set to 1. If both the PBE and SPE bits are set to 1 and the EPE bit is cleared to 0, the transmitted parity bit will be stuck to 1. However, when the PBE and SPE bits are set to 1 and also the EPE bit is set to 1, the transmitted parity bit will be stuck to 0.
[12]	EPE	Even Parity Enable 0: Odd number of logic 1's are transmitted or checked in the data word and parity bits 1: Even number of logic 1's are transmitted or checked in the data word and parity bits This bit is only available when the PBE bit is set to 1.
[11]	PBE	Parity Bit Enable 0: Parity bit is not generated (transmitted data) or checked (received data) during transfer 1: Parity bit is generated or checked during transfer Note: When the WLS field is set to "10" to select the 9-bit data format, writing to the PBE bit has no effect.

Bits	Field	Descriptions
[10]	NSB	Number of STOP bit 0: One STOP bit is generated in the transmitted data 1: Two STOP bits are generated when 8-bit or 9-bit word length is selected
[9:8]	WLS	Word Length Select 00: 7 bits 01: 8 bits 10: 9 bits 11: Reserved
[7]	RXDMAEN	USART RX DMA Enable 0: Disable 1: Enable
[6]	TXDMAEN	USART TX DMA Enable 0: Disable 1: Enable
[5]	URRXEN	USART RX Enable 0: Disable 1: Enable
[4]	URTXEN	USART TX Enable 0: Disable 1: Enable
[3]	HFCEN	Hardware Flow Control Function Enable 0: Disable 1: Enable
[2]	TRSM	Transfer Mode Selection This bit is used to select the data transfer protocol. 0: LSB first 1: MSB first
[1:0]	MODE	USART Mode Selection 00: Normal operation 01: IrDA 10: RS485 11: Synchronous

USART FIFO Control Register – USRFCR

This register specifies the USART FIFO control and configurations including threshold level and reset function together with the USART FIFO status.

Offset: 0x008

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved				RXFS			
Type/Reset					RO	0	RO	0
	23	22	21	20	19	18	17	16
	Reserved				TXFS			
Type/Reset					RO	0	RO	0
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	RXTL		TXTL		Reserved		RXR	TXR
Type/Reset	RW	0	RW	0	RW	0	WO	0

Bits	Field	Descriptions
[27:24]	RXFS	RX FIFO Status The RXFS field shows the current number of data contained in the RX FIFO. 0000: RX FIFO is empty 0001: RX FIFO contains 1 data ... 1000: RX FIFO contains 8 data Others: Reserved
[19:16]	TXFS	TX FIFO Status The TXFS field shows the current number of data contained in the TX FIFO. 0000: TX FIFO is empty 0001: TX FIFO contains 1 data ... 1000: TX FIFO contains 8 data Others: Reserved
[7:6]	RXTL	RX FIFO Threshold Level Setting 00: 1 data 01: 2 data 10: 4 data 11: 6 data The RXTL field defines the RX FIFO trigger level.
[5:4]	TXTL	TX FIFO Threshold Level Setting 00: 0 data 01: 2 data 10: 4 data 11: 6 data The TXTL field determines the TX FIFO trigger level.
[1]	RXR	RX FIFO Reset Setting this bit will generate a reset pulse to reset the RX FIFO which will empty the RX FIFO, i.e., the RX pointer will be reset to 0 after a reset signal. This bit returns to 0 automatically after the reset pulse is generated.

Bits	Field	Descriptions
[0]	TXR	TX FIFO Reset Setting this bit will generate a reset pulse to reset the TX FIFO which will empty the TX FIFO, i.e., the TX pointer will be reset to 0 after a reset signal. This bit returns to 0 automatically after the reset pulse is generated.

USART Interrupt Enable Register – USRIER

This register is used to enable the related USART interrupt function. The USART module generates interrupts to the controller when the corresponding events occur and the corresponding interrupt enable bits are set.

Offset: 0x00C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
	Reserved								
Type/Reset									
	23	22	21	20	19	18	17	16	
	Reserved								
Type/Reset									
	15	14	13	12	11	10	9	8	
	Reserved						CTSIE	RXTOIE	
Type/Reset							RW	0	RW
	7	6	5	4	3	2	1	0	
	RSADDIE	BIE	FEIE	PEIE	OEIE	TXCIE	TXDEIE	RXDRIE	
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[9]	CTSIE	CTS Clear-To-Send Interrupt Enable 0: Disable 1: Enable If this bit is set, an interrupt will be generated when the CTSC bit in the USRSIFR register is set.
[8]	RXTOIE	Receive FIFO Time-Out Interrupt Enable 0: Disable 1: Enable If this bit is set, an interrupt will be generated when the RXTOF bit in the USRSIFR register is set.
[7]	RSADDIE	RS485 Address Detection Interrupt Enable 0: Disable 1: Enable If this bit is set, an interrupt will be generated when the RSADD bit in the USRSIFR register is set.
[6]	BIE	Break Interrupt Enable 0: Disable 1: Enable If this bit is set, an interrupt will be generated when the BII bit in the USRSIFR register is set.

Bits	Field	Descriptions
[5]	FEIE	Framing Error Interrupt Enable 0: Disable 1: Enable If this bit is set, an interrupt will be generated when the FEI bit in the URSIFR register is set.
[4]	PEIE	Parity Error Interrupt Enable 0: Disable 1: Enable If this bit is set, an interrupt will be generated when the PEI bit in the USRSIFR register is set.
[3]	OEIE	Overrun Error Interrupt Enable 0: Disable 1: Enable If this bit is set, an interrupt will be generated when the OEI bit in the USRSIFR register is set.
[2]	TXCIE	Transmit Complete Interrupt Enable 0: Disable 1: Enable If this bit is set, an interrupt will be generated when the TXC bit in the USRSIFR register is set.
[1]	TXDEIE	Transmit Data Empty Interrupt Enable 0: Disable 1: Enable If this bit is set, an interrupt will be generated when the TXDE bit in the USRSIFR register is set.
[0]	RXDRIE	Receive Data Ready Interrupt Enable 0: Disable 1: Enable If this bit is set, an interrupt will be generated when the RXDR bit in the USRSIFR register is set.

This register contains the corresponding USART status.

Reset value: 0x0000 0180

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	Reserved				CTSS	CTSC	RSADDE	TXC	
					RO	0 WC	0 WC	0 RO	1
	7	6	5	4	3	2	1	0	
Type/Reset	TXDE	RXTOF	RXDR	BII	FEI	PEI	OEI	RXDNE	
	RO	1 WC	0 RO	0 WC	0 WC	0 WC	0 WC	0 RO	0

Bits	Field	Descriptions
[11]	CTSS	CTS Clear-To-Send Status 0: CTS pin is inactive 1: CTS pin is active and kept at a logic low state
[10]	CTSC	CTS Status Change Flag This bit will be set whenever the CTS input pin status has been changed and an Interrupt will be generated if the CTSIE bit in the USRIER register is set. Writing 1 to this bit clears the flag.
[9]	RSADD	RS485 Address Detection 0: Address is not detected 1: Address is detected This bit will be set to 1 when the receiver detects the address. An interrupt will be generated if the RSADDIE bit in the USRIER register is set. Writing 1 to this bit clears the flag. Note: This bit is only used in the RS485 mode by setting the MODE field in the USRCR register.
[8]	TXC	Transmit Complete 0: Either transmit FIFO (TX FIFO) or transmit shift register (TSR) is not empty 1: Both the TX FIFO and TSR register are empty When this bit is set, an interrupt will be generated if the TXCIE bit in the USRIER register is set. This bit is cleared by a write to the USRDR register with new data.
[7]	TXDE	Transmit Data FIFO Empty 0: TX FIFO level is higher than threshold 1: TX FIFO level is equal to or less than threshold The TXDE bit will be set when the transmit FIFO level is less than the transmit FIFO threshold level specified by the TXTL field in the USRFCR register. This bit will be cleared when the USRDR register is written with new data and the TX FIFO level is higher than the threshold setting.

Bits	Field	Descriptions
[6]	RXTOF	<p>Receive FIFO Time-Out Flag</p> <p>0: RX FIFO Time-Out does not occur 1: RX FIFO Time-Out occurs</p> <p>The RXTOF bit will be set if the RX FIFO is not empty and no activities have occurred in the RX FIFO during the time-out duration specified by the RXTOC field. If an RX FIFO time-out condition has occurred, this flag must be cleared before reading the RX FIFO. Writing 1 to this bit clears the flag.</p>
[5]	RXDR	<p>Receive FIFO Ready Flag</p> <p>0: RX FIFO level is less than threshold 1: RX FIFO level is equal to or higher than threshold</p> <p>The RXDR bit will be set when the FIFO received data amount has reached the threshold level specified by the RXTL field in the USRFCR register. This bit will be cleared when the data is read from the USRDR register and the RX FIFO level is less than threshold setting.</p>
[4]	BII	<p>Break Interrupt Indicator</p> <p>This bit will be set to 1 whenever the received data input is held in the “spacing state” (logic 0) for longer than a full word transmission time, which is the total time of “start bit” + data bits + “parity” + “stop bits” duration. Writing 1 to this bit clears the flag.</p>
[3]	FEI	<p>Framing Error Indicator</p> <p>This bit will be set 1 whenever the next character to be read in the RX FIFO does not have a valid “stop bit”, which means the stop bit following the last data bit or parity bit is detected as logic 0. Writing 1 to this bit clears the flag.</p>
[2]	PEI	<p>Parity Error Indicator</p> <p>This bit will be set to 1 whenever the received character does not have a valid “parity bit”. Writing 1 to this bit clears the flag.</p>
[1]	OEI	<p>Overrun Error Indicator</p> <p>An overrun error will occur only after the RX FIFO is full and when the next character has been completely received in the RX shift register. The character in the shift register will be overwritten if a new character is received in the RX shift register after an overrun event occurs, but the data in the RX FIFO will not be overwritten. The OEI bit is used to indicate the overrun event as soon as it happens. Writing 1 to this bit clears the flag.</p>
[0]	RXDNE	<p>RX FIFO Data Not Empty</p> <p>0: RX FIFO is empty 1: RX FIFO contains at least 1 received data word</p>

USART Timing Parameter Register – USRTPR

This register contains the USART timing parameters including the transmitter time guard parameters and the receive FIFO time-out value together with the RX FIFO time-out interrupt enable control.

Offset: 0x014

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24		
Type/Reset	Reserved									
	23	22	21	20	19	18	17	16		
Type/Reset	Reserved									
	15	14	13	12	11	10	9	8		
Type/Reset	TG									
	RW	0	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0		
Type/Reset	RXTOEN	RXTOC								
	RW	0	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[15:8]	TG	Transmitter Time Guard The transmitter time guard counter is driven by the baud rate clock. When the TX FIFO transmits data, the counter is reset and then starts to count after a word transmission has completed. Only when the counter content is equal to the TG value, are further word transmission transactions allowed.
[7]	RXTOEN	Receive FIFO Time-Out Counter Enable 0: RX FIFO Time-Out Counter is disabled 1: RX FIFO Time-Out Counter is enabled
[6:0]	RXTOC	Receive FIFO Time-Out Counter Compare Value The RX FIFO time-out counter is driven by the baud rate clock. When the RX FIFO receives new data, the counter is reset and then starts to count. Once the time-out counter content is equal to the time-out counter compare value RXTOC, a receive FIFO time-out interrupt, RXTOI, will be generated if the RXTOIE bit in the USRIER register is set to 1. New received data or the empty RX FIFO after being read will clear the RX FIFO time-out counter.

USART IrDA Control Register – IrDACR

This register is used to control the IrDA mode of USART.

Offset: 0x018

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	IrDAPSC								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	Reserved		RXINV	TXINV	LB	TXSEL	IrDALP	IrDAEN	
			RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:8]	IrDAPSC	<p>IrDA Prescaler value</p> <p>This field contains the 8-bit debounce prescaler value.</p> <p>The debounce count-down counter is driven by the USART clock, named as CK_USART. The counting period is specified by the IrDAPSC field. The IrDAPSC field must be set to a value equal to or greater than 0x01 for normal debounce counter operation. If the pulse width is less than the duration specified by the IrDAPSC field, the pulse will be considered as glitch noise and discarded.</p> <p>00000000: Reserved – can not be used 00000001: CK_USART clock divided by 1 00000010: CK_USART clock divided by 2 00000011: CK_USART clock divided by 3 ...</p>
[5]	RXINV	<p>RX Signal Inverse Control</p> <p>0: No inversion 1: RX input signal is inverted</p>
[4]	TXINV	<p>TX Signal Inverse Control</p> <p>0: No inversion 1: TX output signal is inverted</p>
[3]	LB	<p>IrDA Loop Back Mode</p> <p>0: Disable IrDA loop back mode 1: Enable IrDA loop back mode for self-testing</p>
[2]	TXSEL	<p>Transmit Select</p> <p>0: Enable IrDA receiver 1: Enable IrDA transmitter</p>
[1]	IrDALP	<p>IrDA Low-Power Mode</p> <p>Selects the IrDA operation mode.</p> <p>0: Normal mode 1: IrDA low-power mode</p>

Bits	Field	Descriptions
[0]	IrDAEN	IrDA Enable control 0: Disable IrDA mode 1: Enable IrDA mode

USART RS485 Control Register – RS485CR

This register is used to control the RS485 mode of USART.

Offset: 0x01C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	ADDMATCH								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	Reserved					RSAAD	RSNMM	TXENP	
						RW	0	RW	0

Bits	Field	Descriptions
[15:8]	ADDMATCH	RS485 Auto Address Match value The field contains the address match value for the RS485 auto address detection operation mode.
[2]	RSAAD	RS485 Auto Address Detection Operation Mode Control 0: Disable 1: Enable
[1]	RSNMM	RS485 Normal Multi-drop Operation Mode Control 0: Disable 1: Enable
[0]	TXENP	USART RTS/TXE Pin Polarity 0: RTS/TXE is active high in the RS485 transmission mode 1: RTS/TXE is active low in the RS485 transmission mode

USART Synchronous Control Register – SYNCR

This register is used to control the USART synchronous mode.

Offset: 0x020

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved				CPO	CPS	Reserved	CLKEN
					RW	0	RW	0

Bits	Field	Descriptions
[3]	CPO	<p>Clock Polarity</p> <p>0: CTS/SCK pin idle state is low</p> <p>1: CTS/SCK pin idle state is high</p> <p>Selects the polarity of the clock output on the USART CTS/SCK pin in the synchronous mode. Works in conjunction with the CPS bit to specify the desired clock idle state.</p>
[2]	CPS	<p>Clock Phase</p> <p>0: Data is captured on the first clock edge</p> <p>1: Data is captured on the second clock edge</p> <p>This bit allows the user to select the phase of the clock output on the USART CTS/SCK pin in the synchronous mode. Works in conjunction with the CPO bit to determine the data capture edge.</p>
[0]	CLKEN	<p>Clock Enable</p> <p>0: CTS/SCK pin is disabled</p> <p>1: CTS/SCK pin is enabled</p> <p>Enable/disable the USART CTS/SCK pin.</p>

USART Divider Latch Register – USRDLR

The register is used to determine the USART clock divided ratio to generate the appropriate baud rate.

Offset: 0x024

Reset value: 0x0000_0010

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	BRD								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	BRD								
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:0]	BRD	<p>Baud Rate Divider</p> <p>The 16 bits define the USART clock divider ratio.</p> <p>Baud Rate = CK_USART / BRD</p> <p>Where the CK_USART clock is the clock connected to the USART module.</p> <p>BRD = 16 ~ 65535 for asynchronous mode;</p> <p>BRD = 8 ~ 65535 for synchronous mode.</p>

USART Test Register – USRTSTR

This register controls the USART debug mode.

Offset: 0x028

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved						LBM	
							RW	0 RW 0

Bits	Field	Descriptions
[1:0]	LBM	Loopback Test Mode Select 00: Normal Operation 01: Reserved 10: Automatic Echo Mode 11: Loopback Mode

20 Universal Asynchronous Receiver Transmitter (UART)

Introduction

The Universal Asynchronous Receiver Transceiver, UART, provides a flexible full duplex data exchange using asynchronous transfer. The UART is used to translate data between parallel and serial interfaces, and is also commonly used for RS232 standard communication. The UART peripheral function supports a variety of interrupts.

The UART module includes a transmit data register TDR and transmit shift register TSR, and a receive data register RDR and receive shift register RSR. Software can detect a UART error status by reading the UART Status & Interrupt Flag Register, URSIFR. The status includes the condition of the transfer operations as well as several error conditions resulting from Parity, Overrun, Framing and Break events.

The UART includes a programmable baud rate generator which is capable of dividing the UART clock CK_APB (CK_UART) to produce a baud rate clock for the UART transmitter and receiver.

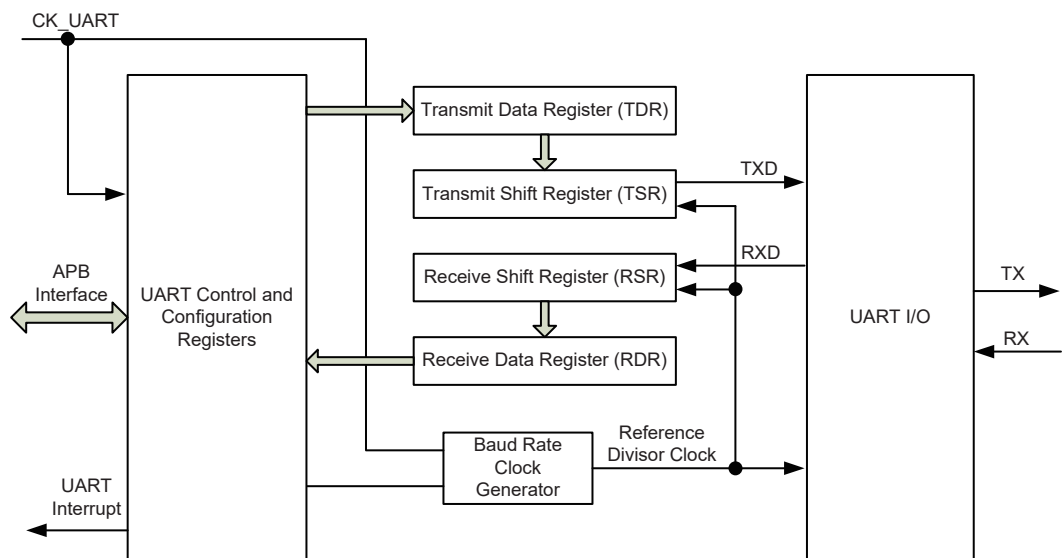


Figure 131. UART Block Diagram

Features

- Supports asynchronous serial communication modes
- Full Duplex Communication Capability
- Programming baud rate clock frequency up to ($f_{PCLK}/16$) MHz
- Fully programmable serial communication functions including:
 - Word length: 7, 8, or 9-bit character
 - Parity: Even, odd, or no-parity bit generation and detection
 - Stop bit: 1 or 2 stop bits generation
 - Bit order: LSB-first or MSB-first transfer
- Error detection: Parity, overrun, and frame error
- Supports PDMA Interface

Functional Descriptions

Serial Data Format

The UART module performs a parallel-to-serial conversion on data that is written to the transmit data register and then sends the data with the following format: Start bit, 7 ~ 9 LSB/MSB first data bits, optional Parity bit and finally 1 ~ 2 Stop bits. The Start bit has the opposite polarity of the data line idle state. The Stop bit is the same as the data line idle state and provides a delay before the next start situation. Both the Start and Stop bits are used for data synchronization during the asynchronous data transmission.

The UART module also performs a serial-to-parallel conversion on the data that is read from the receive data register. It will first check the Parity bit and will then look for a Stop bit. If the Stop bit is not found, the UART module will consider the entire word transmission as failed and respond with a Framing Error.

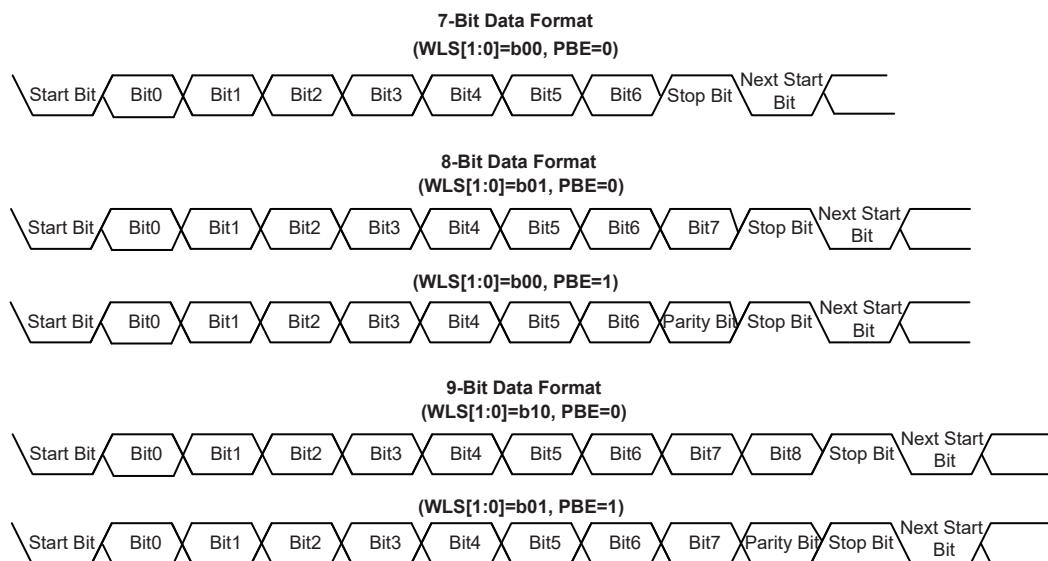


Figure 132. UART Serial Data Format

Baud Rate Generation

The baud rate for the UART receiver and transmitter are both set with the same values. The baud rate divisor, BRD, has the following relationship with the UART clock which is known as CK_UART.

$$\text{Baud Rate Clock} = \text{CK_UART} / \text{BRD}$$

Where the CK_UART clock is the APB clock connected to the UART while the BRD range is from 16 to 65535.

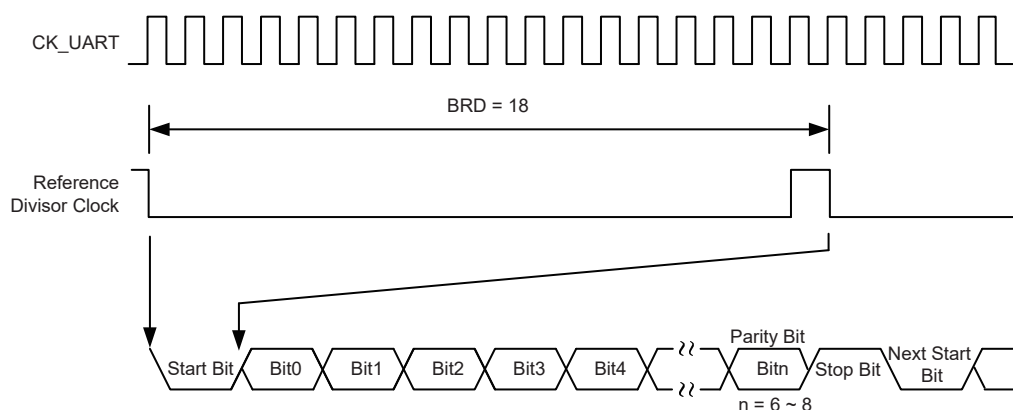


Figure 133. UART Clock CK_UART and Data Frame Timing

Table 52. Baud Rate Deviation Error Calculation – CK_USART = 40 MHz

Baud Rate		CK_USART = 40 MHz		
No.	Kbps	Actual	BRD	Deviation Error Rate
1	2.4	2.4	16667	0.00%
2	9.6	9.6	4167	-0.01%
3	19.2	19.2	2083	0.02%
4	57.6	57.6	694	0.06%
5	115.2	115.3	347	0.06%
6	230.4	229.9	174	-0.22%
7	460.8	459.8	87	-0.22%
8	921.6	930.2	43	0.94%
9	2250	2222.2	18	-1.23%
10	3000	—	—	—

Table 53. Baud Rate Deviation Error Calculation – CK_USART = 48 MHz

Baud Rate		CK_USART = 48 MHz		
No.	Kbps	Actual	BRD	Deviation Error Rate
1	2.4	2.4	20000	0.00%
2	9.6	9.6	5000	0.00%
3	19.2	19.2	2500	0.00%
4	57.6	57.6	833	0.04%
5	115.2	115.1	417	-0.08%
6	230.4	230.8	208	0.16%
7	460.8	461.5	104	0.16%
8	921.6	923.1	52	0.16%
9	2250	2285.7	21	1.59%
10	3000	3000.0	16	0.00%

Table 54. Baud Rate Deviation Error Calculation – CK_USART = 60 MHz

Baud Rate		CK_USART = 60 MHz		
No.	Kbps	Actual	BRD	Deviation Error Rate
1	2.4	2.4	25000	0.00%
2	9.6	9.6	6250	0.00%
3	19.2	19.2	3125	0.00%
4	57.6	57.6	1042	-0.03%
5	115.2	115.2	521	-0.03%
6	230.4	230.8	260	0.16%
7	460.8	461.5	130	0.16%
8	921.6	923.1	65	0.16%
9	2250	2222.2	27	-1.23%
10	3000	3000.0	20	0.00%

Interrupts and Status

The UART can generate interrupts when the following events occur and the corresponding interrupt enable bits are set:

- Receiver line status interrupts: The interrupts are generated when the UART receiver overrun error, parity error, framing error and break event occur.
- Transmit data register empty interrupt: An interrupt is generated when the content of the transmit data register is transferred to the transmit shift register (TSR).
- Transmit complete interrupt: An interrupt is generated when the transmit data register (TDR) is empty and the content of the transmit shift register (TSR) is also completely shifted.
- Receive data ready interrupt: An interrupt is generated when the content of the receive shift register (RSR) has been transferred to the URDR register and is ready to read.

PDMA Interface

The PDMA interface is integrated in the UART. The PDMA function can be enabled by setting the TXDMAEN or RXDMAEN bit in the URCR register to 1 in the transmit or receive mode respectively. When the UART transmit data register (TDR) is empty and the TXDMAEN bit is

set to 1, the PDMA function will be activated to move data from a source location into the UART transmit data register (TDR).

Similarly, when the received data has been in the UART receive data register (RDR) and the RXDMAEN bit is set to 1, the PDMA function will be activated to move data from the UART receive data register (RDR) to a specific destination location. For a more detailed description about the PDMA configurations, refer to the PDMA chapter.

Register Map

The following table shows the UART registers and reset values.

Table 55. UART Register Map

Register	Offset	Description	Reset Value
URDR	0x000	UART Data Register	0x0000_0000
URCR	0x004	UART Control Register	0x0000_0000
URIER	0x00C	UART Interrupt Enable Register	0x0000_0000
URSIFR	0x010	UART Status & Interrupt Flag Register	0x0000_0180
URDLR	0x024	UART Divider Latch Register	0x0000_0010
URTSTR	0x028	UART Test Register	0x0000_0000

Register Descriptions

UART Data Register – URDR

The register is used to access the UART transmitted and received data.

Offset: 0x000

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
	Reserved								
Type/Reset									
	23	22	21	20	19	18	17	16	
	Reserved								
Type/Reset									
	15	14	13	12	11	10	9	8	
	Reserved							DB	
Type/Reset								RW	0
	7	6	5	4	3	2	1	0	
	DB								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[8:0]	DB	<p>By reading this register, the UART will return a 7, 8 and 9-bit received data. The DB field bit 8 is valid for the 9-bit mode only and is fixed at 0 for the 8-bit mode. For the 7-bit mode, the DB[6:0] field contains the available bits.</p> <p>By writing to this register, the UART will send out 7, 8 or 9-bit transmitted data. The DB field bit 8 is valid for the 9-bit mode only and will be ignored for the 8-bit mode. For the 7-bit mode, the DB[6:0] field contains the available bits.</p>

UART Control Register – URCR

The register specifies the serial parameters such as data length, parity and stop bit for the UART.

Offset: 0x004

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24					
Type/Reset	Reserved												
	23	22	21	20	19	18	17	16					
Type/Reset	Reserved												
	15	14	13	12	11	10	9	8					
Type/Reset	Reserved	BCB	SPE	EPE	PBE	NSB	WLS						
		RW	0	RW	0	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0					
Type/Reset	RXDMAEN	TXDMAEN	URRXEN	URTXEN	Reserved	TRSM	Reserved						
	RW	0	RW	0	RW	0	RW	0					

Bits	Field	Descriptions
[14]	BCB	Break Control Bit When this bit is set 1, the serial data output on the UART TX pin will be forced to the Spacing State (logic 0). This bit acts only on the UART TX output pin and has no effect on the transmitter logic.
[13]	SPE	Stick Parity Enable 0: Disable stick parity 1: Stick Parity bit is transmitted This bit is only available when the PBE bit is set to 1. If both the PBE and SPE bits are set to 1 and the EPE bit is cleared to 0, the transmitted parity bit will be stuck to 1. However, when the PBE and SPE bits are set to 1 and also the EPE bit is set to 1, the transmitted parity bit will be stuck to 0.
[12]	EPE	Even Parity Enable 0: Odd number of logic 1's are transmitted or checked in the data word and parity bits 1: Even number of logic 1's are transmitted or checked in the data word and parity bits This bit is only available when the PBE bit is set to 1.
[11]	PBE	Parity Bit Enable 0: Parity bit is not generated (transmitted data) and checked (receive data) during transfer 1: Parity bit is generated and checked during transfer Note: When the WLS field is set to "10" to select the 9-bit data format, writing to the PBE bit has no effect.
[10]	NSB	Number of STOP bit 0: One STOP bit is generated in the transmitted data 1: Two STOP bits are generated when 8-bit or 9-bit word length is selected
[9:8]	WLS	Word Length Select 00: 7 bits 01: 8 bits 10: 9 bits 11: Reserved

Bits	Field	Descriptions
[7]	RXDMAEN	UART RX DMA Enable 0: Disable 1: Enable
[6]	TXDMAEN	UART TX DMA Enable 0: Disable 1: Enable
[5]	URRXEN	UART RX Enable 0: Disable 1: Enable
[4]	URTXEN	UART TX Enable 0: Disable 1: Enable
[2]	TRSM	Transfer Mode Selection This bit is used to select the data transfer protocol. 0: LSB first 1: MSB first

UART Interrupt Enable Register – URIER

This register is used to enable the related UART interrupt function. The UART module generates interrupts to the controller when the corresponding events occur and the corresponding interrupt enable bits are set.

Offset: 0x00C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24							
	Reserved														
Type/Reset															
	23	22	21	20	19	18	17	16							
	Reserved														
Type/Reset															
	15	14	13	12	11	10	9	8							
	Reserved														
Type/Reset															
	7	6	5	4	3	2	1	0							
	Reserved	BIE	FEIE	PEIE	OEIE	TXCIE	TXDEIE	RXDRIE							
Type/Reset		RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[6]	BIE	Break Interrupt Enable 0: Disable 1: Enable If this bit is set, an interrupt will be generated when the BII bit in the URSIFR register is set.
[5]	FEIE	Framing Error Interrupt Enable 0: Disable 1: Enable If this bit is set, an interrupt will generated when the FEI bit in the URSIFR register is set.

Bits	Field	Descriptions
[4]	PEIE	Parity Error Interrupt Enable 0: Disable 1: Enable If this bit is set, an interrupt will be generated when the PEI bit in the URSIFR register is set.
[3]	OEIE	Overrun Error Interrupt Enable 0: Disable 1: Enable If this bit is set, an interrupt will be generated when the OEI bit in the URSIFR register is set.
[2]	TXCIE	Transmit Complete Interrupt Enable 0: Disable 1: Enable If this bit is set, an interrupt will be generated when the TXC bit in the URSIFR register is set.
[1]	TXDEIE	Transmit Data Register Empty Interrupt Enable 0: Disable 1: Enable If this bit is set, an interrupt will be generated when the TXDE bit in the URSIFR register is set.
[0]	RXDRIE	Receive Data Ready Interrupt Enable 0: Disable 1: Enable If this bit is set, an interrupt will be generated when the RXDR bit in the URSIFR register is set.

UART Status & Interrupt Flag Register – URSIFR

This register contains the corresponding UART status.

Offset: 0x010

Reset value: 0x0000_0180

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	Reserved							TXC	
	7	6	5	4	3	2	1	0	
Type/Reset	TXDE	Reserved	RXDR	BII	FEI	PEI	OEI	Reserved	
	RO	1	RO	0	WC	0	WC	0	WC

Bits	Field	Descriptions
[8]	TXC	Transmit Complete 0: Either the transmit data register (TDR) or transmit shift register (TSR) is not empty 1: Both the transmit data register (TDR) and transmit shift register (TSR) are empty When this bit is set, an interrupt will be generated if the TXCIE bit in the URIER register is set to 1. This bit is cleared by a write to the URDR register with new data.
[7]	TXDE	Transmit Data Register Empty 0: Transmit data register is not empty 1: Transmit data register is empty The TXDE bit is set by hardware when the content of the transmit data register is transferred to the transmit shift register (TSR). An interrupt will be generated if the TXEIE bit in the URIER register is set to 1. This bit is cleared by a write to the URDR register with new data.
[5]	RXDR	RX Data Ready 0: Receive data register is empty 1: The received data in receive data register is ready to read This bit is set by hardware when the content of the receive shift register (RSR) has been transferred to the URDR register. An interrupt will be generated if the RXDRIE bit in the URIER register is set to 1. It is cleared by a read to the URDR register.
[4]	BII	Break Interrupt Indicator This bit is set to 1 whenever the received data input is held in the “spacing state” (logic 0) for longer than a full character transmission time, which is the total time of “start bit” + data bits + “parity” + “stop bits” duration. Writing 1 to this bit clears the flag.
[3]	FEI	Framing Error Indicator This bit is set 1 whenever the received character does not have a valid stop bit, which means, the stop bit following the last data bit or parity bit is detected as logic 0. Writing 1 to this bit clears the flag.

Bits	Field	Descriptions
[2]	PEI	Parity Error Indicator This bit is set to 1 whenever the received character does not have a valid parity bit. Writing 1 to this bit clears the flag.
[1]	OEI	Overrun Error Indicator An overrun error will occur only after the receive data register is full and when the next character has been completely received in the receive shift register. The character in the receive shift register will be overwritten when a new character is received in the receive shift register after an overrun event occurs, but the data in the receive shift register will not be transferred to the receive data register. The OEI bit is used to indicate event as soon as it happens. Writing 1 to this bit clears the flag.

UART Divider Latch Register – URDLR

The register is used to determine the UART clock divided ratio to generate the appropriate baud rate.

Offset: 0x024

Reset value: 0x0000_0010

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	BRD							
	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
Type/Reset	BRD							
	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[15:0]	BRD	Baud Rate Divider The 16 bits define the UART clock divider ratio. Baud Rate = CK_UART / BRD Where the CK_UART clock is the clock connected to the UART module. BRD = 16 ~ 65535 for the UART mode

UART Test Register – URTSTR

This register controls the UART debug mode.

Offset: 0x028

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved						LBM	
							RW	0 RW 0

Bits	Field	Descriptions
[1:0]	LBM	Loopback Test Mode Select 00: Normal Operation 01: Reserved 10: Automatic Echo Mode 11: Loopback Mode

21 USB Device Controller (USB)

Introduction

The USB device controller is compliant with the USB 2.0 full-speed specification. There is one control endpoint known as Endpoint 0 and seven configurable endpoints (EP1 ~ EP7). A 1024-byte EP_SRAM is used for the endpoint buffers. Each endpoint buffer size is programmable by corresponding registers, which provides maximum flexibility for various applications. The integrated USB full-speed transceiver helps to minimize overall system complexity and cost. The USB also contains the suspend and resume features to meet low-power consumption requirement. The accompanying figure shows the USB block diagram.

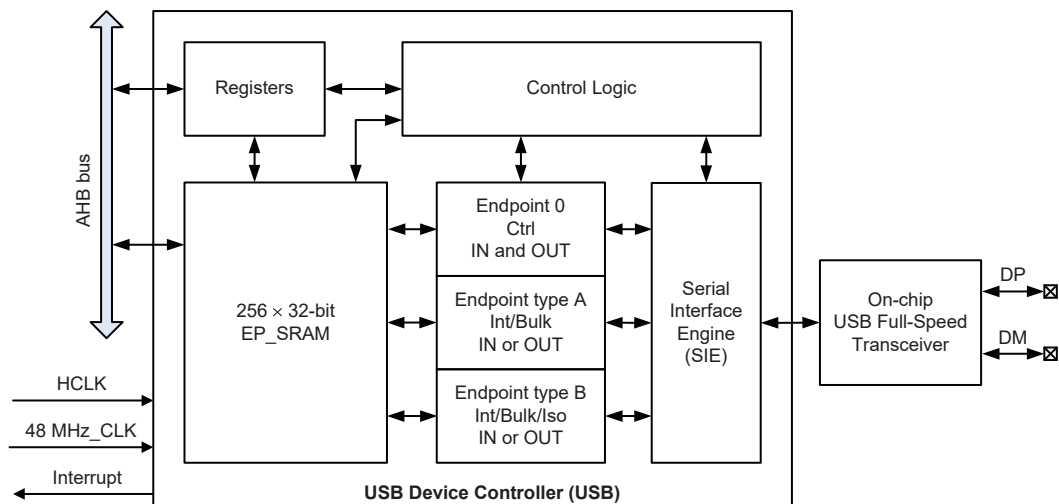


Figure 134. USB Block Diagram

Features

- Complies with USB 2.0 full-speed (12 Mbps) specification
- Fully integrated USB full-speed transceiver
- 1 control endpoint (EP0) for control transfer
- 3 single-buffered endpoints (EP1 ~ EP3) for bulk and interrupt transfer
- 4 double-buffered endpoints (EP4 ~ EP7) for bulk, interrupt and isochronous transfer
- 1,024 bytes EP_SRAM used as endpoint data buffers

Functional Descriptions

Endpoints

The USB Endpoint 0 is the only bidirectional endpoint dedicated to USB control transfer. The device also contains seven unidirectional endpoints for other USB transfer types. There are three endpoints (EP1 ~ EP3) which support a single buffering function which is used for Bulk and Interrupt IN or OUT data transfer. There are four other endpoints (EP4 ~ EP7) which support single or double buffering functions for Bulk, Interrupt and Isochronous IN or OUT data transfer. The address of the seven unidirectional endpoints (EP1 ~ EP7) can be configured by the application software. The following table lists the endpoint characteristics.

Table 56. Endpoint Characteristics

Endpoint Number	Number Address	Transfer Type	Direction	Buffer Type
0	Fixed	Control	IN and OUT	Single buffering
1 ~ 3	Configurable	Interrupt/Bulk	IN or OUT	Single buffering
4 ~ 7	Configurable	Interrupt/Bulk/Isochronous	IN or OUT	Single or Double buffering

EP_SRAM

The USB controller contains a dedicated memory space, EP_SRAM, which is used for the USB endpoint buffers. The EP_SRAM, which is connected to the APB bus, can be accessed by the MCU and PDMA. The EP_SRAM base address is 0x400A_A000 with an offset which ranges from 0x000 to 0x3FF. The EP_SRAM first two words are reserved for Endpoint 0 to temporarily store the 8-byte SETUP data. Therefore the valid start address of the endpoint buffer should start from 0x400A_0x008 and align to a 4-byte boundary. Each endpoint buffer size is programmable. The following table lists the maximum USB endpoint buffer size which is compliant with USB 2.0 full-speed device specification.

Table 57. USB Data Types and Buffer Size

Transfer Type	Direction	Supported Buffer Size	Bandwidth	CRC	Retrying
Control	Bidirectional	8, 16, 32, 64 bytes	Not guaranteed	Yes	Automatic
Bulk	Unidirectional	8, 16, 32, 64 bytes	Not guaranteed	Yes	Yes
Interrupt	Unidirectional	≤ 64 bytes	Not guaranteed	Yes	Yes
Isochronous	Unidirectional	< 512 bytes	Guaranteed	Yes	No

In the following endpoint buffer allocation example, the Endpoint “4” is configured as a double-buffered Bulk IN endpoint while the Endpoint “5” is configured as a double-buffered Bulk OUT endpoint. Each endpoint buffer size is set to 64 bytes.

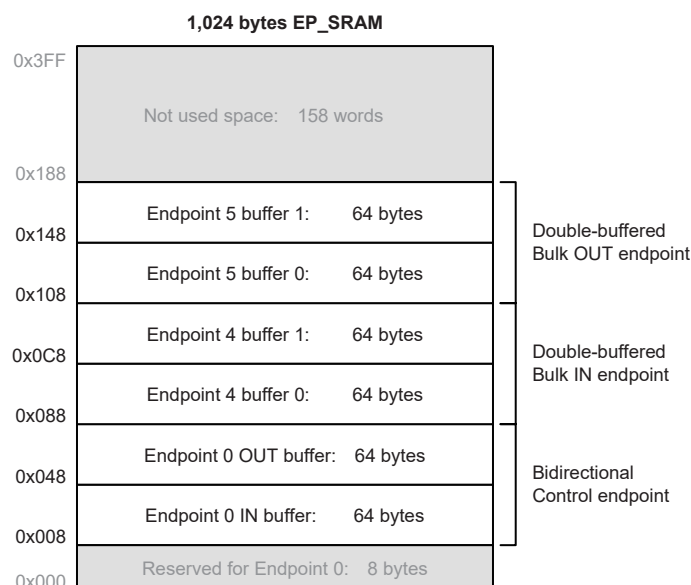


Figure 135. Endpoint Buffer Allocation Example

Serial Interface Engine – SIE

The Serial Interface Engine, SIE, which is connected to the USB full-speed transceiver and internal USB control circuitry, provides a temporal buffer for the transmitted and received data. The SIE also decodes the SE0 signal, SE1 signal, J-state, K-state, USB RESET event and End of Packet event signals, EOP, when the USB module receives data, transmits data or transmits the resume signal for remote control. The SIE detects the number of SOF packets and generates the SOF interrupt signal to the USB control circuitry which includes data format conversion from parallel to serial or serial to parallel. It also includes CRC checking and generation, PID decoder, bit-stuffing and de-bit-stuffing functions.

Double-Buffering

The double buffering function is recommended to be enabled when the corresponding endpoint is specified to be used for Isochronous transfer or high throughput Bulk transfer. The double buffering function stores the preceding data packet sent by the USB host in a simple buffer for the MCU to process and the hardware will ensure that it continues to receive the current data packet in the other buffer during an OUT transaction, and vice versa. Using a double buffering function can achieve the highest possible data transfer rate. The details regarding double buffering usage is provided in the corresponding UDBTG and MDBTG control bit description in the USBEPnCSR register where the denotation n ranges from 4 to 7.

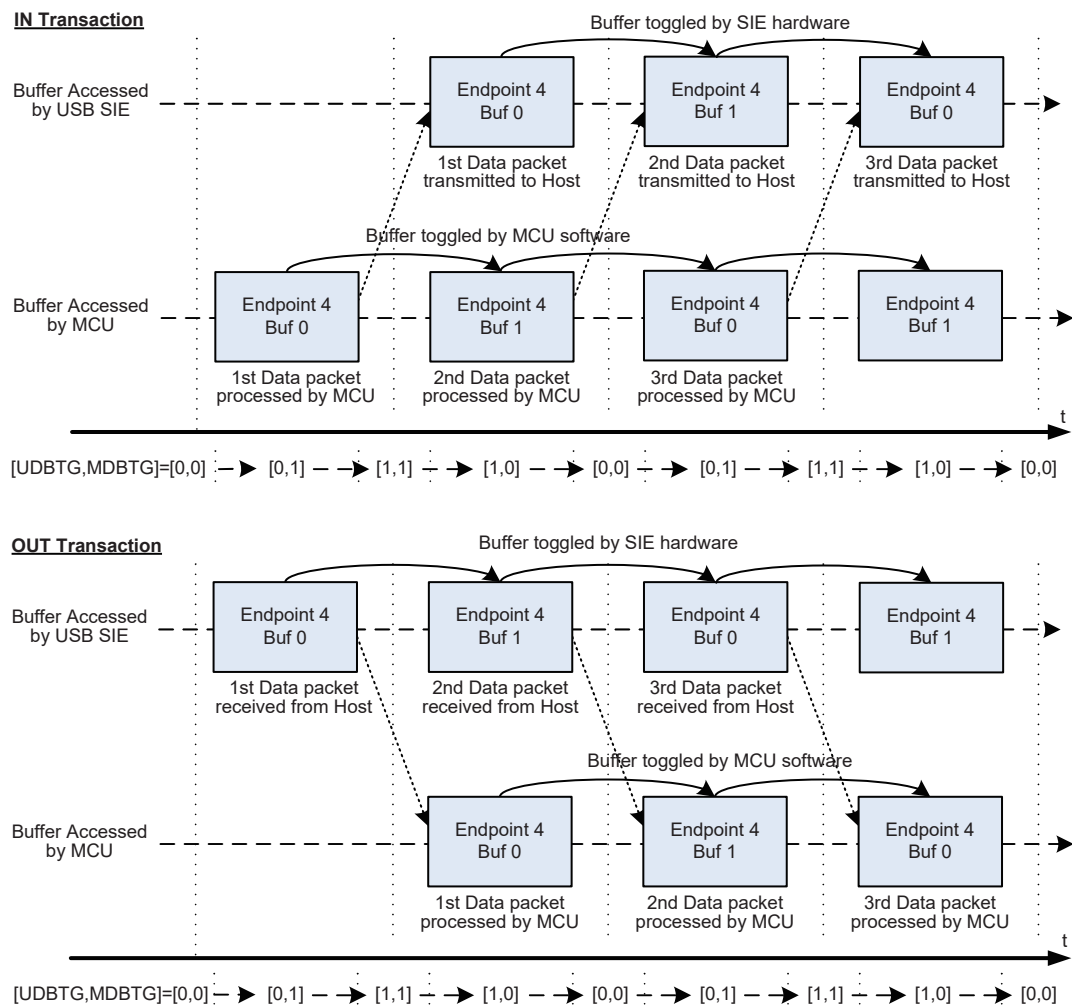


Figure 136. Double-buffering Operation Example

Suspend Mode and Wake-up

According to USB specifications, the device must enter the suspend mode after a 3 ms bus idle time. When the USB device enters the suspend mode, the current from the USB bus must not be greater than 500 μ A to meet the specification suspend mode current requirements. The USB control circuitry will generate a suspend interrupt if the bus is in the idle state for 3 ms. Here the software should set the LPMODE and PDWN bits in the USBCSR register to 1. The LPMODE bit is used to determine whether the USB controller enters the low-power mode or not by holding the USB bus in a reset condition while the PDWN bit is used to determine if the integrated USB full-speed transceiver is turned off or not.

There are two ways for the USB host to wake up the USB device, one is to send a USB reset signal, SE0, and the other is to send a USB resume signal known as the K-state. After a wake-up signal, regardless of whether a SE0 signal or a K-state is detected, the USB device will be woken up.

Remote Wake-up

As the USB device has a remote wake-up function, it can wake up the USB host by sending a resume request signal by setting the GENRSM bit in the USBCSR register to 1. Once the USB host receives the remote wake-up signal from the USB device, it will send a resume signal to the USB device.

Register Map

The following table shows the USB registers and reset values.

Table 58. USB Register Map

Register	Offset	Description	Reset Value
USBCSR	0x000	USB Control and Status Register	0x0000_00X6
USBIER	0x004	USB Interrupt Enable Register	0x0000_0000
USBISR	0x008	USB Interrupt Status Register	0x0000_0000
USBFCR	0x00C	USB Frame Count Register	0x0000_0000
USBDEVAR	0x010	USB Device Address Register	0x0000_0000
USBEP0CSR	0x014	USB Endpoint 0 Control and Status Register	0x0000_0002
USBEP0IER	0x018	USB Endpoint 0 Interrupt Enable Register	0x0000_0000
USBEP0ISR	0x01C	USB Endpoint 0 Interrupt Status Register	0x0000_0000
USBEP0TCR	0x020	USB Endpoint 0 Transfer Count Register	0x0000_0000
USBEP0CFGR	0x024	USB Endpoint 0 Configuration Register	0x8000_0008
USBEP1CSR	0x028	USB Endpoint 1 Control and Status Register	0x0000_0002
USBEP1IER	0x02C	USB Endpoint 1 Interrupt Enable Register	0x0000_0000
USBEP1ISR	0x030	USB Endpoint 1 Interrupt Status Register	0x0000_0000
USBEP1TCR	0x034	USB Endpoint 1 Transfer Count Register	0x0000_0000
USBEP1CFGR	0x038	USB Endpoint 1 Configuration Register	0x1000_03FF
USBEP2CSR	0x03C	USB Endpoint 2 Control and Status Register	0x0000_0002
USBEP2IER	0x040	USB Endpoint 2 Interrupt Enable Register	0x0000_0000
USBEP2ISR	0x044	USB Endpoint 2 Interrupt Status Register	0x0000_0000
USBEP2TCR	0x048	USB Endpoint 2 Transfer Count Register	0x0000_0000
USBEP2CFGR	0x04C	USB Endpoint 2 Configuration Register	0x1000_03FF
USBEP3CSR	0x050	USB Endpoint 3 Control and Status Register	0x0000_0002
USBEP3IER	0x054	USB Endpoint 3 Interrupt Enable Register	0x0000_0000
USBEP3ISR	0x058	USB Endpoint 3 Interrupt Status Register	0x0000_0000
USBEP3TCR	0x05C	USB Endpoint 3 Transfer Count Register	0x0000_0000
USBEP3CFGR	0x060	USB Endpoint 3 Configuration Register	0x1000_03FF
USBEP4CSR	0x064	USB Endpoint 4 Control and Status Register	0x0000_0002
USBEP4IER	0x068	USB Endpoint 4 Interrupt Enable Register	0x0000_0000
USBEP4ISR	0x06C	USB Endpoint 4 Interrupt Status Register	0x0000_0000
USBEP4TCR	0x070	USB Endpoint 4 Transfer Count Register	0x0000_0000
USBEP4CFGR	0x074	USB Endpoint 4 Configuration Register	0x1000_03FF
USBEP5CSR	0x078	USB Endpoint 5 Control and Status Register	0x0000_0002
USBEP5IER	0x07C	USB Endpoint 5 Interrupt Enable Register	0x0000_0000
USBEP5ISR	0x080	USB Endpoint 5 Interrupt Status Register	0x0000_0000
USBEP5TCR	0x084	USB Endpoint 5 Transfer Count Register	0x0000_0000
USBEP5CFGR	0x088	USB Endpoint 5 Configuration Register	0x1000_03FF

Register Descriptions

This register specifies the USB control bits and USB data line status.

Offset: 0x000

Reset value: 0x0000 00X6

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved			FREST	DPWKEN	DPPUEN	SRAMRSTC	ADRSET
			RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
Type/Reset	RXDM	RXDP	GENRSM	Reserved	LPMODE	PDWN	FRES	Reserved
	RO	X RO	X RW	0	RW	0 RW	1 RW	1

Rev. 1.20

Bits	Field	Descriptions
[8]	ADRSET	<p>Device Address Setting Control</p> <p>This bit is used to determine when the USB SIE updates the device address with the value of the USBDEVAR register.</p> <p>0: The SIE updates the device address immediately after an address is written into the USBDEVAR register.</p> <p>1: The SIE updates the device address after the USB Host has successfully read the data from the device by the IN operation. This bit is cleared by the SIE after the device address is updated.</p>
[7]	RXDM	<p>Received DM Line Status</p> <p>This bit is used to observe the status of DM data line status at the end of suspend routines to determine whether a wakeup event has occurred.</p>
[6]	RXDP	<p>Received DP Line Status</p> <p>This bit is used to observe the status of DP data line status at the end of suspend routines to determine whether a wake-up event has occurred.</p>
[5]	GENRSM	<p>Resume Request Generation Control</p> <p>This bit is used to generate a resume request which is sent to the USB host by writing 1 into this bit location. The USB remote wake-up function is always enabled. This bit will be cleared to 0 after a resume signal sent by the USB host has been received.</p>
[3]	LPMODE	<p>Low-power Mode Control</p> <p>This bit is used to determine the USB operating mode. Setting this bit will force the USB to enter the low-power mode. When USB bus traffic, known as a wake-up event, is detected by the hardware, this bit should be cleared by software.</p> <p>0: Exit the Low-power mode</p> <p>1: Enter the Low-power mode</p>
[2]	PDWN	<p>Power-Down Mode Control</p> <p>Setting this bit will power down the full-speed USB PHY transceiver. This will disconnect the USB PHY transceiver from the USB bus.</p> <p>0: Exit the Power-Down</p> <p>1: Enter the Power-Down mode</p>
[1]	FRES	<p>Force USB Reset Control</p> <p>This bit is used to reset the USB circuitry. Setting this bit will force the USB into a reset state until the software clears it. A USB reset interrupt will be generated if the corresponding interrupt enable bit in the USBIER register is set to 1. All related USB registers are reset to their default values.</p> <p>0: Release USB reset</p> <p>1: Force USB reset</p>

Table 59. Resume Event Detection

[RXDP, RXDM] Status	Wake-up Event	Required Resume Software Action
00	Root reset	None
10	None (noise on bus)	Go back to suspend mode
01	Root resume	None
11	Not allowed (noise on bus)	Go back to suspend mode

USB Interrupt Enable Register – USBIER

This register specifies the USB interrupt enable control.

Offset: 0x004

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	EP7IE	EP6IE	EP5IE	EP4IE	EP3IE	EP2IE	EP1IE	EP0IE	
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	Reserved	FRESIE	ESOFIE	SUSPIE	RSMIE	URSTIE	SOFIE	UGIE	
		RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[15:8]	EPnIE	Endpoint n Interrupt Enable Control (n = 0 ~ 7) 0: Disable interrupt 1: Enable interrupt
[6]	FRESIE	Force USB Reset Control (FRES) Interrupt Enable Control 0: Disable FRES interrupt 1: Enable FRES interrupt
[5]	ESOFIE	Expected Start-Of-Frame (ESOF) Interrupt Enable Control 0: Disable ESOF interrupt 1: Enable ESOF interrupt
[4]	SUSPIE	Suspend Interrupt Enable Control 0: Disable Suspend interrupt 1: Enable Suspend interrupt
[3]	RSMIE	Resume Interrupt Enable Control 0: Disable Resume interrupt 1: Enable Resume interrupt
[2]	URSTIE	USB Reset Interrupt Enable Control 0: Disable USB Reset interrupt 1: Enable USB Reset interrupt
[1]	SOFIE	Start-Of-Frame (SOF) Interrupt Enable Control 0: Disable SOF interrupt 1: Enable SOF interrupt
[0]	UGIE	USB Global Interrupt Enable Control 0: Disable USB Global interrupt 1: Enable USB Global interrupt This bit must be set to 1 to enable the corresponding USB interrupt function. If this bit is cleared to 0, the relevant USB interrupt will not be generated, however, the corresponding interrupt flags will still be asserted.

USB Interrupt Status Register – USBISR

This register specifies the USB interrupt status.

Offset: 0x008

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	EP7IF	EP6IF	EP5IF	EP4IF	EP3IF	EP2IF	EP1IF	EP0IF
	WC	0	WC	0	WC	0	WC	0
	7	6	5	4	3	2	1	0
Type/Reset	Reserved	FRESIF	ESOFIF	SUSPIF	RSMIF	URSTIF	SOFIF	Reserved
		WC	0	RW	0	WC	0	WC

Bits	Field	Descriptions
[15:8]	EPnIF	Endpoint n Interrupt Flag (n = 0 ~ 7) This bit is set by the hardware to indicate the generation of relevant endpoint interrupts. Writing 1 into this bit to clear it. It is important to note that the interrupt flag can only be cleared when the endpoint interrupt status bit in the USBEPnISR register is equal to 0.
[6]	FRESIF	Force USB Reset Control (FRES) Interrupt Flag This bit is set by the hardware. When this bit is set 1, this means that Force USB Reset FRES has finished. This bit is cleared to 0 by writing 1.
[5]	ESOFIF	Expected Start-Of-Frame Interrupt Flag This bit is set by the hardware when an SOF packet is expected to be received. The USB host sends an SOF (Start-Of-Frame) packet each millisecond. If the USB device hardware does not receive it properly, an ESOF interrupt will be generated when the ESOFIE bit in the USBIER register is set to 1. If three consecutive ESOF interrupts are generated, which means that the SOF packet has been missed 3 times, the SUSPIF flag will be set to 1. This bit will be set to 1 when the missing SOF packets occur if the timer is not yet locked. This bit can be read or written. However, only 0 can be written into this bit. Writing 1 has no effect.
[4]	SUSPIF	Suspend Interrupt Flag This bit is set by the hardware when no data transfer has occurred for 3ms, indicating that a suspend request has been sent from the USB host. The suspend condition check is enabled immediately after a USB reset. This bit is cleared to 0 by writing 1.
[3]	RSMIF	Resume Interrupt Flag This bit is set by the hardware. When this bit is set 1, this means that a device resume has occurred. This bit is cleared to 0 by writing 1.

Bits	Field	Descriptions
[2]	URSTIF	<p>USB Reset Interrupt Flag</p> <p>This bit is set by the hardware when the USB reset has been detected. When a USB reset occurs, the internal protocol state machine will be reset and an USB reset interrupt will be generated if the URSTIE bit in the USBIER register is set to 1. Data reception and transmission are disabled until the URSTIF bit is cleared to 0. The USB configuration related registers (USBCSR, USBIER, USBISR, USBFCR and USBDEVAR) will not be reset by a USB reset event except for the USB device address (USBDEVAR), this is to ensure that a USB reset interrupt can be safely excited and any data transactions immediately followed by the USB reset can be completely accessed by the software. Therefore the microcontroller must properly reset these registers. The USB endpoint related registers (USBEPnCSR, USBEPnISR and USBEPnTCR) are also reset by a USB reset event, however, the endpoint configuration (USBEPnCFGR) and interrupt enable (USBEPnIER) registers are not affected by the USB reset event and will remain unchanged.</p> <p>This bit is cleared to 0 by writing 1.</p>
[1]	SOFIF	<p>SOF Interrupt Flag</p> <p>This bit is set by the hardware when a start-of-frame packet has been received. This bit is cleared to 0 by writing 1.</p>

USB Frame Count Register – USBFCR

This register specifies the lost Start-of-Frame number and the USB frame count.

Offset: 0x00C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved					LSOF		SOFLCK	
						RO	0	RO	0
	15	14	13	12	11	10	9	8	
Type/Reset	Reserved					FRNUM			
						RO	0	RO	0
	7	6	5	4	3	2	1	0	
Type/Reset	FRNUM								
	RO	0	RO	0	RO	0	RO	0	RO

Bits	Field	Descriptions
[18:17]	LSOF	<p>Lost Start-of-Frame number</p> <p>These bits are written and incremented by 1 by the hardware each time the ESOFIF bit is set. It is used to count the number of lost SOF packets. Once this field equals to 3, which means that the SOF packet has been lost 3 times, the SUSPIF flag will be set to 1. When a SOF packet has been received, these bits are cleared.</p>

Bits	Field	Descriptions
[16]	SOFLCK	Start-of-Frame Lock Flag This bit is set by the hardware when SOF packets have been received before the frame timer times out. Once this flag is set to 1, the frame number which is sent from the USB host will be loaded into the Frame Number field FRNUM in the USBFCR register. If there is no SOF packet has been received during the 1ms frame time duration, this bit will be cleared to 0.
[10:0]	FRNUM	Frame Number This field stores the frame number received from the USB host.

USB Device Address Register – USBDEVAR

This register specifies the USB device address.

Offset: 0x010

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24									
Type/Reset	Reserved																
	23	22	21	20	19	18	17	16									
Type/Reset	Reserved																
	15	14	13	12	11	10	9	8									
Type/Reset	Reserved																
	7	6	5	4	3	2	1	0									
Type/Reset	Reserved	DEVA															
		RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[6:0]	DEVA	Device Address This field is used to specify the USB device address. This field is cleared when a USB reset event occurs.

USB Endpoint 0 Control and Status Register – USBEP0CSR

This register specifies the Endpoint 0 control and status.

Offset: 0x014

Reset value: 0x0000_0002

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset		Reserved	STLRX	NAKRX	DTGRX	STLTX	NAKTX	DTGTX
			RW	0	RW	0	RW	0
				0	RW	0	RW	1
								RW
								0

Bits	Field	Descriptions
[5]	STLRX	<p>STALL Status for reception (OUT) transfer</p> <p>This bit is set to 1 by the application software and then returns a STALL signal in the handshake phase of an OUT transaction if a functional error is detected. This means that a control request delivered from the USB host is not supported by the USB device. The STALL status is cleared by the hardware circuitry when a new SETUP token is received.</p> <p>This bit can be read and written and can only be toggled by writing 1.</p>
[4]	NAKRX	<p>NAK Status for reception (OUT) transfer</p> <p>This bit is toggled from 0 to 1 by the hardware circuitry, which will result in a NAK signal in the handshake phase of an OUT transaction after an ACK signal has been transmitted. This means that the USB device will be temporarily unable to accept data from the USB host. Therefore, more time will be required for the received data to be properly processed.</p> <p>This bit can be read and written and can only be toggled by writing 1.</p>
[3]	DTGRX	<p>Data Toggle Status for reception (OUT) transfer</p> <p>This bit contains the expected value of the data toggle bit (0 = DATA0, 1 = DTAT1) for the next data packet to be received. When the current valid data packet is received and the corresponding ACK signal is sent to the USB host by the USB device, the hardware circuitry will toggle this bit and the device will be ready to receive the next data packet. For Endpoint 0, the hardware circuitry will toggle this bit to 1 after the SETUP token is received as Endpoint 0 is addressed. This bit can also be toggled by the software to initialize its value for certain applications.</p> <p>This bit can be read and written and can only be toggled by writing 1.</p>
[2]	STLTX	<p>STALL Status for transmission (IN) transfer</p> <p>This bit is set to 1 by the application software and then returns a STALL signal in response to an IN token if a functional error is detected. This means that the USB device is unable to transmit data. The STALL status is cleared by the hardware circuitry when a new SETUP token is received.</p> <p>This bit can be read and written and can only be toggled by writing 1.</p>

Bits	Field	Descriptions
[1]	NAKTX	NAK Status for transmission (IN) transfer This bit is toggled from 0 to 1 by the hardware circuitry, which will result in a NAK signal in the handshake phase of an IN transaction after an ACK signal has been received. It indicates that the USB device is temporarily unable to transmit data to the USB host. Therefore, there will be more time for the application software to properly prepare the data to be transmitted. This bit can be read and written and can only be toggled by writing 1.
[0]	DTGTX	Data Toggle Status for transmission (IN) transfer This bit contains the required value of the data toggle bit (0 = DATA0, 1 = DATA1) for the next data packet to be transmitted. When the current data packet is transmitted by the USB device and the corresponding ACK signal sent by the USB host is received, the hardware circuitry will toggle this bit and the next data packet will be transmitted. For Endpoint 0, the hardware circuitry will toggle this bit to 1 after the SETUP token is received as Endpoint 0 is addressed. This bit can also be toggled by the software to initialize its value for certain applications. This bit can be read and written and can only be toggled by writing 1.

USB Endpoint 0 Interrupt Enable Register – USBEP0IER

This register specifies the Endpoint 0 interrupt control bits.

Offset: 0x018

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved				ZLRXIE	SDERIE	SDRXIE	STRXIE
					RW	0	RW	0
	7	6	5	4	3	2	1	0
Type/Reset	UERIE	STLIE	NAKIE	IDTXIE	ITRXIE	ODOVIE	ODRXIE	OTRXIE
	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[11]	ZLRXIE	Zero Length Data Received Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[10]	SDERIE	SETUP Data Error Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[9]	SDRXIE	SETUP Data Received Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt

Bits	Field	Descriptions
[8]	STRXIE	SETUP Token Received Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[7]	UERIC	USB Error Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[6]	STLIE	STALL Transmitted Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[5]	NAKIE	NAK Transmitted Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[4]	IDTXIE	IN Data Transmitted Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[3]	ITRXIE	IN Token Received Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[2]	ODOVIE	OUT Data Buffer Overrun Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[1]	ODRXIE	OUT Data Received Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[0]	OTRXIE	OUT Token Received Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt

USB Endpoint 0 Interrupt Status Register – USBEP0ISR

This register specifies the Endpoint 0 interrupt status.

Offset: 0x01C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved				ZLRXIF	SDERIF	SDRXIF	STRXIF
					WC	0	WC	0
	7	6	5	4	3	2	1	0
Type/Reset	UERIF	STLIF	NAKIF	IDTXIF	ITRXIF	ODOVIF	ODRXIF	OTRXIF
	WC	0	WC	0	WC	0	WC	0

Bits	Field	Descriptions
[11]	ZLRXIF	Zero Length Data Received Interrupt Flag This bit is set by the hardware when a zero length data packet is received. This bit is cleared by hardware when a SETUP Token is received or by writing 1.
[10]	SDERIF	SETUP Data Error Interrupt Flag This bit is set by the hardware when the SETUP data packet length is not 8 bytes. This bit is cleared by hardware when a SETUP Token is received or by writing 1.
[9]	SDRXIF	SETUP Data Received Interrupt Flag This bit is set by the hardware when a SETUP data packet from the USB host has been received. This bit is cleared by the hardware when a SETUP Token is received or by writing 1. If the received SETUP data is not accessed by the application software before the next SETUP packet is received, the SETUP data buffer will be overwritten.
[8]	STRXIF	SETUP Token Received Interrupt Flag This bit is set by the hardware when a SETUP token is received and is cleared by writing 1.
[7]	UERIF	USB Error Interrupt Flag This bit is set by the hardware when an error occurs during the Endpoint 0 transaction. This bit is cleared by hardware when a SETUP Token is received or by writing 1.
[6]	STLIF	STALL Transmitted Interrupt Flag This bit is set by the hardware when a STALL signal is sent in response to an IN or OUT transaction. This bit is cleared by hardware when a SETUP Token is received or by writing 1.
[5]	NAKIF	NAK Transmitted Interrupt Flag This bit is set by the hardware when a NAK signal is sent in response to an IN or OUT transaction. This bit is cleared by hardware when a SETUP Token is received or by writing 1.

Bits	Field	Descriptions
[4]	IDTXIF	IN Data Transmitted Interrupt Flag This bit is set by the hardware when a data packet is transmitted to and then an ACK signal is received from the USB host. This bit is cleared by hardware when a SETUP Token is received or by writing 1.
[3]	ITRXIF	IN Token Received Interrupt Flag This bit is set by the hardware when the IN token is received from the USB host. This bit is cleared by hardware when a SETUP Token is received or by writing 1.
[2]	ODOVIF	OUT Data Buffer Overrun Interrupt Flag This bit is set by the hardware when the number of received data bytes is larger than the endpoint buffer size. This bit is cleared by hardware when a SETUP Token is received or by writing 1.
[1]	ODRXIF	OUT Data Received Interrupt Flag This bit is set by the hardware when a data packet is successfully received from the USB host and then an ACK signal is sent to the USB host. This bit is cleared by hardware when a SETUP Token is received or by writing 1.
[0]	OTRXIF	OUT Token Received Interrupt Flag This bit is set by the hardware when the OUT token is received from the USB host. This bit is cleared by hardware when a SETUP Token is received or by writing 1.

USB Endpoint 0 Transfer Count Register – USBEP0TCR

This register specifies the Endpoint 0 data transfer byte count.

Offset: 0x020

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved	RXCNT							
	15	14	13	12	11	10	9	8	
Type/Reset	Reserved								
	7	6	5	4	3	2	1	0	
Type/Reset	Reserved	TXCNT							

Bits	Field	Descriptions
[22:16]	RXCNT	Reception Byte Count The bit field contains the number of data bytes received by Endpoint 0 in the preceding SETUP transaction.
[6:0]	TXCNT	Transmission Byte Count The bit field contains the number of data bytes to be transmitted by Endpoint 0 in the next IN token. If the value of this field is zero, it indicates that a zero length packet will be sent.

USB Endpoint 0 Configuration Register – USBEP0CFGR

This register specifies the Endpoint 0 configurations.

Offset: 0x024

Reset value: 0x8000_0008

	31	30	29	28	27	26	25	24
	EPEN	Reserved						EPADR
Type/Reset	RO 1	RO 0 RO 0 RO 0 RO 0						0
	23	22	21	20	19	18	17	16
	Reserved							EPLEN
Type/Reset	RW 0							0
	15	14	13	12	11	10	9	8
	EPLEN						EPBUFA	
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	EPBUFA							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 1	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[31]	EPEN	Endpoint Enable Control This bit is always set to 1 by the hardware circuitry to always enable Endpoint 0.
[27:24]	EPADR	Endpoint Address This field is always set to 0 by the hardware circuitry.
[16:10]	EPLEN	Endpoint Buffer Length This field is used to specify the control transfer packet size which can be 8, 16, 32 or 64 bytes as defined in the USB full-speed standard specification.
[9:0]	EPBUFA	Endpoint Buffer Address This field is used to specify the start address of the Endpoint 0 buffer allocated in the EP_SRAM. It starts from 0x008 and should be aligned to 4-byte boundary. Start address of EP0 IN buffer = EPBUFA Start address of EP0 OUT buffer = EPBUFA + EPLEN

USB Endpoint 1 ~ 3 Control and Status Register – USBEPnCSR, n = 1 ~ 3

This register specifies the Endpoint 1 ~ 3 control and status bit.

Offset: 0x028 (n = 1), 0x03C (n = 2), 0x050 (n = 3)

Reset value: 0x0000_0002

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset		Reserved	STLRX	NAKRX	DTGRX	STLTX	NAKTX	DTGTX
			RW	0	RW	0	RW	0
				0			1	0

Bits	Field	Descriptions
[5]	STLRX	<p>STALL bit for reception transfers</p> <p>This bit is set to 1 by the application software if a functional error has been detected. This bit can be read and written and can only be toggled by writing 1. It can also be toggled by the software to initialize the value under certain conditions.</p>
[4]	NAKRX	<p>NAK bit for reception transfers</p> <p>This bit is toggled from 0 to 1 by the hardware circuitry, which will result in a NAK signal in the handshake phase of an OUT transaction after an ACK signal has been transmitted. It means that the USB device will be temporarily unable to accept data from the USB host until the received data is properly processed.</p> <p>This bit can be read and written and can be only toggled by writing 1.</p>
[3]	DTGRX	<p>Data Toggle bit for reception transfers</p> <p>This bit contains the expected value of the data toggle bit (0 = DATA0, 1 = DATA1) for the next data packet to be received. When the current valid data packet is received and the corresponding ACK signal is sent to the USB host by the USB device, the hardware circuitry will toggle this bit and the device will be ready to receive the next data packet.</p> <p>This bit can be read and written and can only be toggled by writing 1. This bit can also be toggled by the software to initialize its value under certain conditions.</p>
[2]	STLTX	<p>STALL bit for transmission transfers</p> <p>This bit is set to 1 by the application software if a functional error has been detected. This bit can be read and written and can be only toggled by writing 1. It can also be toggled by the software to initialize its value under certain conditions.</p>
[1]	NAKTX	<p>NAK bit for transmission transfers</p> <p>This bit is toggled from 0 to 1 by the hardware circuitry, which will result in a NAK signal in the handshake phase of an IN transaction after an ACK signal has been received. It means that the USB device will be temporarily unable to transmit data packet until the data to be transmitted is appropriately prepared by the application software.</p> <p>This bit can be read and written and can be only toggled by writing 1.</p>

Bits	Field	Descriptions
[0]	DTGTX	<p>Data Toggle bit for transmission transfers</p> <p>This bit contains the required value of the data toggle bit (0 = DATA0, 1 = DATA1) for the next data packet to be transmitted. When the current data packet is transmitted by the USB device and the corresponding ACK signal sent from the USB host is received, the hardware circuitry will toggle this bit and then the next data packet will be transmitted.</p> <p>This bit can be read and written and can only be toggled by writing 1. It can also be toggled by the software to initialize its value under certain conditions.</p>

USB Endpoint 1 ~ 3 Interrupt Enable Register – USBEPnIER, n = 1 ~ 3

This register specifies the Endpoint 1 ~ 3 interrupt enable control bits.

Offset: 0x02C (n = 1), 0x040 (n = 2), 0x054 (n = 3)

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	UERIE	STLIE	NAKIE	IDTXIE	ITRXIE	ODOVIE	ODRXIE	OTRXIE
Type/Reset	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[7]	UERIE	<p>USB Error Interrupt Enable Control</p> <p>0: Disable interrupt</p> <p>1: Enable interrupt</p>
[6]	STLIE	<p>STALL Transmitted Interrupt Enable Control</p> <p>0: Disable interrupt</p> <p>1: Enable interrupt</p>
[5]	NAKIE	<p>NAK Transmitted Interrupt Enable Control</p> <p>0: Disable interrupt</p> <p>1: Enable interrupt</p>
[4]	IDTXIE	<p>IN Data Transmitted Interrupt Enable Control</p> <p>0: Disable interrupt</p> <p>1: Enable interrupt</p>
[3]	ITRXIE	<p>IN Token Received Interrupt Enable Control</p> <p>0: Disable interrupt</p> <p>1: Enable interrupt</p>
[2]	ODOVIE	<p>OUT Data Buffer Overrun Interrupt Enable Control</p> <p>0: Disable interrupt</p> <p>1: Enable interrupt</p>

Bits	Field	Descriptions
[1]	ODRXIE	OUT Data Received Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[0]	OTRXIE	OUT Token Received Interrupt Enable Control. 0: Disable interrupt 1: Enable interrupt

USB Endpoint 1 ~ 3 Interrupt Status Register – USBEPnISR, n = 1 ~ 3

This register specifies the Endpoint 1 ~ 3 interrupt status.

Offset: 0x030 (n = 1), 0x044 (n = 2), 0x058 (n = 3)

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	UERIF	STLIF	NAKIF	IDTXIF	ITRXIF	ODOVIF	ODRXIF	OTRXIF
	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0

Bits	Field	Descriptions
[7]	UERIF	USB Error Interrupt Flag This bit is set by the hardware when an error occurs during the transaction. Writing 1 into this status bit will clear it to 0.
[6]	STLIF	STALL Transmitted Interrupt Flag This bit is set by hardware circuitry when a STALL signal is sent in response to an IN or OUT token and is cleared to 0 by writing 1.
[5]	NAKIF	NAK Transmitted Interrupt Flag This bit is set by hardware circuitry when an NAK signal is sent in response to an IN or OUT token and is cleared to 0 by writing 1.
[4]	IDTXIF	IN Data Transmitted Interrupt Flag This bit is set by hardware circuitry when a data packet is successfully transmitted to the host in response to an IN token and an ACK signal is received. Writing 1 into this status bit will clear it to 0.
[3]	ITRXIF	IN Token Received Interrupt Flag This bit is set by the hardware circuitry when the endpoint receives an IN token from the host and is cleared to 0 by writing 1.

Bits	Field	Descriptions
[2]	ODOVIF	OUT Data Buffer Overrun Interrupt Flag This bit is set by the hardware circuitry when the received data byte count is larger than the corresponding endpoint OUT data buffer size. Writing 1 into this status bit will clear it to 0.
[1]	ODRXIF	OUT Data Received Interrupt Flag This bit is set by the hardware circuitry when a data packet is successfully received from the host for an OUT token and when an endpoint n ACK signal is sent to the host. Writing 1 into this status bit will clear it to 0.
[0]	OTRXIF	OUT Token Received Interrupt Flag This bit is set by the hardware circuitry when the endpoint receives an OUT token from the host and is cleared to 0 by writing 1.

USB Endpoint 1 ~ 3 Transfer Count Register – USBEPnTCR, n = 1 ~ 3

This register specifies the Endpoint 1 ~ 3 transfer byte count.

Offset: 0x034 (n = 1), 0x048 (n = 2), 0x05C (n = 3)

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	Reserved							TCNT	
	7	6	5	4	3	2	1	0	
Type/Reset	TCNT								
	RW	0	RW	0	RW	0	RW	0	
	0	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[8:0]	TCNT	Transfer Byte Count This field contains the number of bytes received by the endpoint n in the preceding OUT transaction or the number of bytes to be transmitted by the endpoint n in the next IN transaction.

USB Endpoint 1 ~ 3 Configuration Register – USBEPnCFGR, n = 1 ~ 3

This register specifies the Endpoint 1 ~ 3 configurations.

Offset: 0x038 (n = 1), 0x04C (n = 2), 0x060 (n = 3)

Reset value: 0x1000_03FF

	31	30	29	28	27	26	25	24
	EPEN	Reserved	EPTYPE	EPDIR	EPADR			
Type/Reset	RW 0		RW 0	RW 1	RW 0	RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	Reserved						EPLEN	
Type/Reset							RW 0	
	15	14	13	12	11	10	9	8
	EPLEN						EPBUFA	
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 1	RW 1
	7	6	5	4	3	2	1	0
	EPBUFA							
Type/Reset	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1

Bits	Field	Descriptions
[31]	EPEN	Enable Control 0: Disable the endpoint n 1: Enable the endpoint n
[29]	EPTYPE	Transfer Type This bit is set to 0 by the hardware circuitry to specify that the endpoint n transfer type is an Interrupt or Bulk transfer type.
[28]	EPDIR	Transfer Direction 0: OUT 1: IN
[27:24]	EPADR	Endpoint Address The EPADR field value can be assigned by the application software to specify the address of the endpoint n. It is important to note that this EPADR field should not be set to 0; otherwise, the endpoint will be disabled.
[16:10]	EPLEN	Buffer Length This field is used to specify the endpoint n data packet size. The field value must be word-aligned to a 4-byte boundary. The maximum size in this field can be 64 bytes which is the maximum payload as defined in the USB full-speed standard specification. Note that the EPLEN value should not be assigned to 0 which will result in the endpoint being disabled.
[9:0]	EPBUFA	Endpoint Buffer Address This field is used to specify the endpoint n data buffer start address which ranges from 0x008 to 0x3FC in the EP_SRAM which has a capacity of 1024 bytes and whose field value must be a multiple of 4.

USB Endpoint 4 ~ 7 Control and Status Register – USBEPnCSR, n = 4 ~ 7

This register specifies the Endpoint 4 ~ 7 control and status bits.

Offset: 0x064 (n = 4), 0x078 (n = 5), 0x08C (n = 6), 0x0A0 (n = 7)

Reset value: 0x0000_0002

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	UDBTG	MDBTG	STLRX	NAKRX	DTGRX	STLTX	NAKTX	DTGTX
	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
------	-------	--------------

[7]	UDBTG	<p>USB Double Buffer Toggle bit</p> <p>The UDBTG and MDBTG bits are used to indicate which data buffer is accessed by the USB SIE hardware and which data buffer is accessed by the MCU software if the double buffering function is enabled. The UDBTG bit will be toggled by the SIE hardware circuitry after the current buffer operation is complete. After the UDBTG bit is toggled by the SIE, an NAK signal will be sent automatically to the USB host by the hardware circuitry. Therefore, the data transfer will be stopped temporarily until the data in the other buffer has been properly setup after which the MDBTG bit is toggled by the MCU application software.</p> <p>The following tables show the double buffering operation and the UDBTG and MDBTG bit status for an IN or OUT transaction.</p>
-----	-------	---

Transaction Type	UDBTG	MDBTG	Buffer Read by SIE	Buffer Written by MCU
IN	0	0	None*	EP_BUF0
	0	1	EP_BUF0	EP_BUF1
	1	1	None*	EP_BUF1
	1	0	EP_BUF1	EP_BUF0

Transaction Type	UDBTG	MDBTG	Buffer Written by SIE	Buffer Read by MCU
OUT	0	0	None*	EP_BUF0
	0	1	EP_BUF0	EP_BUF1
	1	1	None*	EP_BUF1
	1	0	EP_BUF1	EP_BUF0

* Means the USB device sends a NAK signal to the USB host using the hardware circuitry.

The UDBTG and MDBTG bits setting procedure for the double buffering function is shown in the following example:

[UDBTG, MDBTG] = [0, 0] → [0, 1] → [1, 1] → [1, 0] → [0, 0] → [0, 1] → [1, 1] → [1, 0] → ...

Bits	Field	Descriptions
[6]	MDBTG	<p>MCU Double Buffer Toggle bit</p> <p>The MDBTG bit is used to indicate which data buffer is accessed by the MCU if the double buffering function is enabled. It can be toggled to switch to the other buffer by the MCU application software after the data in the current buffer accessed by the MCU has been properly setup. The double buffering operation together with the UDBTG and MDBTG bits are shown in the preceding two tables for the UDBTG bit definition</p>
[5]	STLRX	<p>STALL bit for reception transfers</p> <p>This bit is set to 1 by the application software if a functional error has been detected. This bit can be read and written and can only be toggled by writing 1. It can also be toggled by software to initialize its value under certain conditions.</p>
[4]	NAKRX	<p>NAK bit for reception transfers</p> <p>This bit is toggled from 0 to 1 by the hardware circuitry, which will result in a NAK signal in the handshake phase of an OUT transaction after an ACK signal has been transmitted. It means that the USB device will be temporarily unable to accept data from the USB host until the received data is properly processed. If the endpoint is defined as an Isochronous transfer type, this bit is not available for usage. The hardware will not change the NAKRX bit status after a complete transaction. This bit can be read and written and can be only toggled by writing 1.</p>
[3]	DTGRX	<p>Data Toggle bit for reception transfers</p> <p>If the endpoint is not used for Isochronous transfer, this bit is available for usage. This bit contains the expected value of the data toggle bit (0 = DATA0, 1 = DATA1) for the next data packet to be received. When the current valid data packet is received and the corresponding ACK signal is sent to the USB host by the USB device, the hardware circuitry will toggle this bit and the device will be ready to receive the next data packet. If the endpoint is defined as an Isochronous transfer type, this bit is not used since no data toggling is used and only the DATA0 packet will be transferred for normal Isochronous transfers. This bit can be read and written and can only be toggled by writing 1. This bit can also be toggled by the software to initialize its value under certain conditions.</p>
[2]	STLTX	<p>STALL bit for transmission transfers</p> <p>This bit is set to 1 by the application software if there a functional error has been detected. This bit can be read and written and can be only toggled by writing 1. It can be toggled by the software to initialize its value under certain conditions.</p>
[1]	NAKTX	<p>NAK bit for transmission transfers</p> <p>This bit is toggled from 0 to 1 by the hardware circuitry, which will result in a NAK signal in the handshake phase of an IN transaction after an ACK signal has been received. It means that the USB device will be temporarily unable to transmit a data packet until the data to be transmitted is properly setup by the application software. If the endpoint is defined as an Isochronous transfer type, then this bit is not available for usage. The hardware will not change the NAKTX bit status after a complete transaction. This bit can be read and written and can be only toggled by writing 1. It can also be toggled by the software to initialize its value under certain conditions.</p>

Bits	Field	Descriptions
[0]	DTGTX	Data Toggle bit for transmission transfers If the endpoint is not used for Isochronous transfer, this bit is available for usage. This bit contains the required value of the data toggle bit (0 = DATA0, 1 = DATA1) for the next data packet to be transmitted. When the current data packet is transmitted by the USB device and the corresponding ACK signal sent from the USB host is received, the hardware circuitry will toggle this bit and then the next data packet will be transmitted. If the endpoint is used for Isochronous transfer, this bit is not used since no data toggling is used and only the DATA0 packet will be transferred for normal Isochronous transfer. This bit can be read and written and can only be toggled by writing 1. It can also be toggled by the software to initialize its value under certain conditions.

USB Endpoint 4 ~ 7 Interrupt Enable Register – USBEPnIER, n = 4 ~ 7

This register specifies the Endpoint 4 ~ 7 interrupt enable control bits.

Offset: 0x068 (n = 4), 0x07C (n = 5), 0x090 (n = 6), 0x0A4 (n = 7)

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	URIE	STLIE	NAKIE	IDTXIE	ITRXIE	ODOVIE	ODRXIE	OTRXIE
	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[7]	URIE	USB Error Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[6]	STLIE	STALL Transmitted Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[5]	NAKIE	NAK Transmitted Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[4]	IDTXIE	IN Data Transmitted Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[3]	ITRXIE	IN Token Received Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt

Bits	Field	Descriptions
[2]	ODOVIE	OUT Data Buffer Overrun Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[1]	ODRXIE	OUT Data Received Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[0]	OTRXIE	OUT Token Received Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt

USB Endpoint 4 ~ 7 Interrupt Status Register – USBEPnISR, n = 4 ~ 7

This register specifies the Endpoint 4 ~ 7 interrupt status.

Offset: 0x06C (n = 4), 0x080 (n = 5), 0x094 (n = 6), 0x0A8 (n = 7)

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	UERIF	STLIF	NAKIF	IDTXIF	ITRXIF	ODOVIF	ODRXIF	OTRXIF
	WC	0	WC	0	WC	0	WC	0

Bits	Field	Descriptions
[7]	UERIF	USB Error Interrupt flag This bit is set by the hardware circuitry when an error occurs during the transaction. Writing 1 into this status bit will clear it to 0.
[6]	STLIF	STALL Transmitted Interrupt flag This bit is set by the hardware circuitry when a STALL signal is sent in response to an IN or OUT token and is cleared to 0 by writing 1.
[5]	NAKIF	NAK Transmitted Interrupt flag This bit is set by the hardware circuitry when an NAK signal is sent in response to an IN or OUT token and is cleared to 0 by writing 1.
[4]	IDTXIF	IN Data Transmitted Interrupt flag This bit is set by the hardware circuitry when a data packet is successfully transmitted to the host in response to an IN token and an ACK signal is received. Writing 1 into this status bit will clear it to 0.
[3]	ITRXIF	IN Token Received Interrupt flag This bit is set by the hardware circuitry when the endpoint receives an IN token from the host and is cleared to 0 by writing 1.

Bits	Field	Descriptions
[2]	ODOVIF	OUT Data Buffer Overrun Interrupt flag This bit is set by the hardware circuitry when the received data byte count is larger than the endpoint OUT data buffer size. Writing 1 into this status bit will clear it to 0.
[1]	ODRXIF	OUT Data Received Interrupt flag This bit is set by the hardware circuitry when a data packet is successfully received from the host for an OUT token and an ACK signal is sent to the host. Writing 1 into this status bit will clear it to 0.
[0]	OTRXIF	OUT Token Received Interrupt flag This bit is set by the hardware circuitry when the endpoint receives an OUT token from the host and is cleared to 0 by writing 1.

USB Endpoint 4 ~ 7 Transfer Count Register – USBEPnTCR, n = 4 ~ 7

This register specifies the Endpoint 4 ~ 7 transfer byte count.

Offset: 0x070 (n = 4), 0x084 (n = 5), 0x098 (n = 6), 0x0AC (n = 7)

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
	Reserved						TCNT1		
Type/Reset							RW	0	RW
	23	22	21	20	19	18	17	16	
	TCNT1								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	Reserved						TCNT0		
Type/Reset							RW	0	RW
	7	6	5	4	3	2	1	0	
	TCNT0								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[25:16]	TCNT1	Buffer 1 Transfer Byte Count This bit field contains the number of data bytes received by the endpoint n buffer 1 in the preceding OUT transaction or the number of data bytes to be transmitted by the endpoint n buffer1 in the next IN transaction.
[9:0]	TCNT0	Buffer 0 Transfer Byte Count This bit field contains the number of data bytes received by the endpoint n buffer 0 in the preceding OUT transaction or the number of data bytes to be transmitted by the endpoint n buffer 0 in the next IN transaction. Only the TCNT0 field is used for the endpoint data transfer count when the endpoint is configured as a single-buffering transfer type.

USB Endpoint 4 ~ 7 Configuration Register – USBEPnCFGR, n = 4 ~ 7

This register specifies the Endpoint 4 ~ 7 configurations.

Offset: 0x074 (n = 4), 0x088 (n = 5), 0x09C (n = 6), 0x0B0 (n = 7)

Reset value: 0x1000_03FF

	31	30	29	28	27	26	25	24
	EPEN	Reserved	EPTYPE	EPDIR	EPADR			
Type/Reset	RW 0		RW 0	RW 1	RW 0	RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	SDBS	Reserved			EPLEN			
Type/Reset	RW 0				RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	EPLEN						EPBUFA	
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 1	RW 1
	7	6	5	4	3	2	1	0
	EPBUFA							
Type/Reset	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1

Bits	Field	Descriptions
[31]	EPEN	Enable Control 0: Disable the endpoint n 1: Enable the endpoint n
[29]	EPTYPE	Transfer Type 0: Interrupt or Bulk transfer type 1: Isochronous transfer type
[28]	EPDIR	Transfer Direction 0: OUT 1: IN
[27:24]	EPADR	Endpoint Address The EPADR field can be configured by the application software to specify the address of endpoint n. It is important to note that this EPADR field should not be set to 0; otherwise, the endpoint n will be disabled.
[23]	SDBS	Single-Buffering or Double-Buffering Selection 0: Single-buffering 1: Double-buffering If the SDBS bit is set to 1, the endpoint buffer size is twice that of the EPLEN value: - Endpoint Buffer 0 start address is EPBUFA - Endpoint Buffer 1 start address is (EPBUFA + EPLEN)
[19:10]	EPLEN	Buffer Length This field is used to specify the endpoint n data packet size whose field value must be word-aligned to a 4-byte boundary. Note that the endpoint will be disabled if the EPLEN value is assigned to 0.
[9:0]	EPBUFA	Buffer Address This field is used to specify the endpoint n data buffer start address which ranges from 0x008 to 0x3FC in the EP_SRAM which has a capacity of 1024 bytes where the endpoint transfer data is stored. Note that the buffer start address value must be a multiple of 4.

22 Peripheral Direct Memory Access (PDMA)

Introduction

The Peripheral Direct Memory Access circuitry, PDMA, provides 6 unidirectional channels for dedicated peripherals to implement the peripheral-to-memory and memory-to-peripheral data transfer. The memory-to-memory data transfer such as the FLASH-to-SRAM or SRAM-to-SRAM type is also supported and requested by the application program. Each PDMA channel configuration is independent. The PDMA channel transfer is split into multiple block transactions and the size of a block is equal to the block length multiplied by the data width.

Features

- 6 unidirectional PDMA channels
- Memory-to-peripheral, peripheral-to-memory and memory-to-memory data transfer
- 8-bit, 16-bit and 32-bit width data transfer
- Software and hardware requested data transfer with configurable channel priority
- Linear, circular and fixed address modes
- 4 transfer event flags – Transfer Complete, Half Transfer, Block End and Transfer Error
- Auto-Reload function

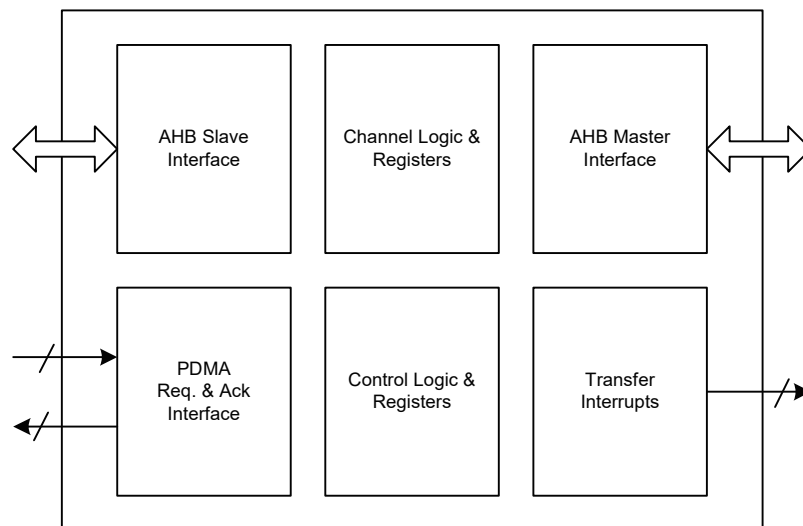


Figure 137. PDMA Block Diagram

Functional Description

AHB Master

The PDMA is an AHB master connected to other AHB peripherals such as the FLASH memory, the SRAM memory and the AHB-to-APB bridges through the bus-matrix. The CPU and PDMA can access different AHB slaves at the same time via the bus-matrix.

PDMA Channel

There are 6 unidirectional PDMA channels used to support data transfer between the peripherals and the memory. The configuration and operation of each PDMA channel is independent. For a bidirectional transfer application, two PDMA channels are required. Each PDMA channel is designed to support the dedicated multiple peripherals with the same registers. Therefore, one PDMA channel only can service one peripheral at the same time. The related registers of the PDMA channel are limited to be accessed with 32-bit operation; otherwise a system hard fault event will occur.

PDMA Request Mapping

The multiple requests from the peripherals (SPI, I²C, USART and so on) are simply logically ANDed before entering the PDMA, which means that only one request must be enabled at a time in each PDMA channel. Refer to Figure 137 – PDMA request mapping architecture and detailed peripheral IP requests mapping table is shown as the Table 60. The peripheral DMA requests can be independently activated/de-activated by programming the DMA control bit in the registers of the corresponding peripheral.

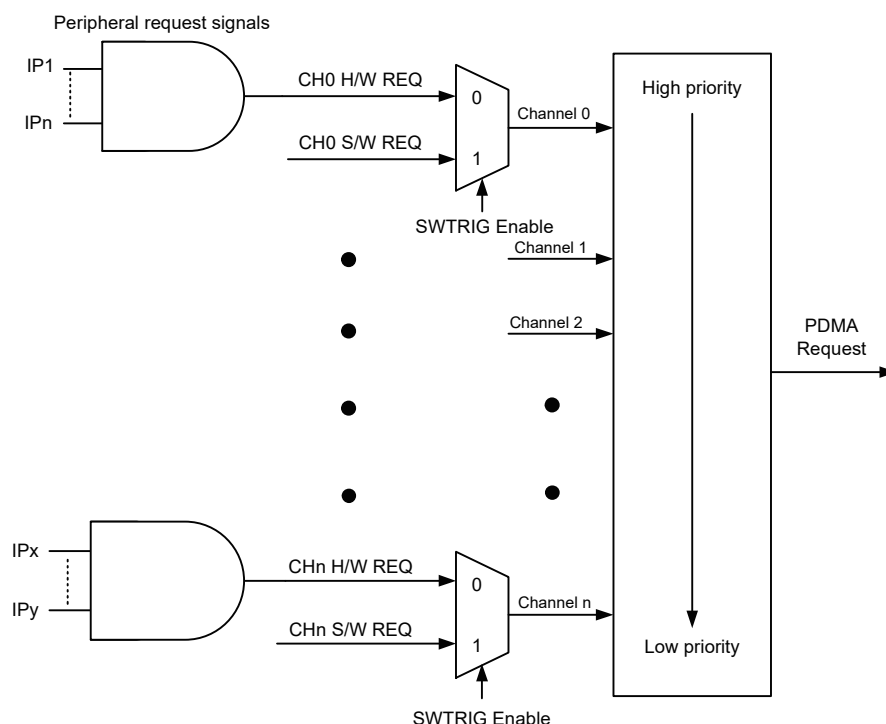


Figure 138. PDMA Request Mapping Architecture

Table 60. PDMA Channel Assignments

IP (x = 0, 1)	PDMA Channel Number					
	CH0	CH1	CH2	CH3	CH4	CH5
SPI	SPI_RX	SPI_TX				
USART	USR_RX	USR_TX				
UART			UR_RX	UR_TX		
I ² C			I2C_RX		I2C_TX	
GPTM	GT_CH1 GT_CH3	GT_CH2 GT_UEV	GT_CH0 GT_TRIG			
PWMx	PWM0_CH1 PWM0_CH3	PWM0_CH2 PWM0_UEV	PWM0_CH0 PWM0_TRIG	PWM1_CH1 PWM1_CH3	PWM1_CH2 PWM1_UEV	PWM1_CH0 PWM1_TRIG
AES					AES_OUT	AES_IN

Channel Transfer

A PDMA channel transfer is split into multiple block transactions with PDMA arbitration occurring at the end of each block transaction. Although these channel transfers can all be activated, there is only one block transaction being transferred through the bus at a time. The channel transfer sequence depends upon the channel priority setting of each PDMA channel. The total transfer size is calculated from the block transaction count and block size. The block size is equal to the product of the block length and data bit width. For an efficient transfer, it is recommended that the block length is set as a multiple of 4.

The total transfer data size calculation is shown as the following equation:

A PDMA channel total transfer data size = Block transaction count × (Block length × Data width)

Channel Priority

The PDMA provides four priority levels, known as very high, high, medium and low, which can be configured by the application software. The PDMA also provides two methods to determine the channel priority. One is determined by application software configuration and the other is determined by the fixed hardware channel number. The PDMA arbitration processor will first check the software configuring channel priority level used to request the PDMA to provide the data transfer services. If more than one channel has the same priority, the channel with a smaller channel number will have priority over one with a larger channel number after arbitration.

Note that the highest priority channel will not occupy the PDMA service all the time when other lower priority channel requests are pending. The highest priority channel will be skipped for one block transaction time duration after one block transaction is complete. Then a block transaction requested by the second priority channel will be performed. After a block transaction of the second priority channel is complete, the PDMA arbitration processor will re-check all of the requested channel priority with the exception of the second priority channel since the second priority channel will be excluded after the end of a block transaction. Therefore, a block data transaction of the higher priority channel will be serviced and this channel will be excluded from the priority arbitration at the end of the block transaction. The PDMA will keep transferring the data using the method described above until all of the requested channel data transfer is complete. Refer to the accompanying figure for an example which shows the PDMA channel arbitration and scheduling.

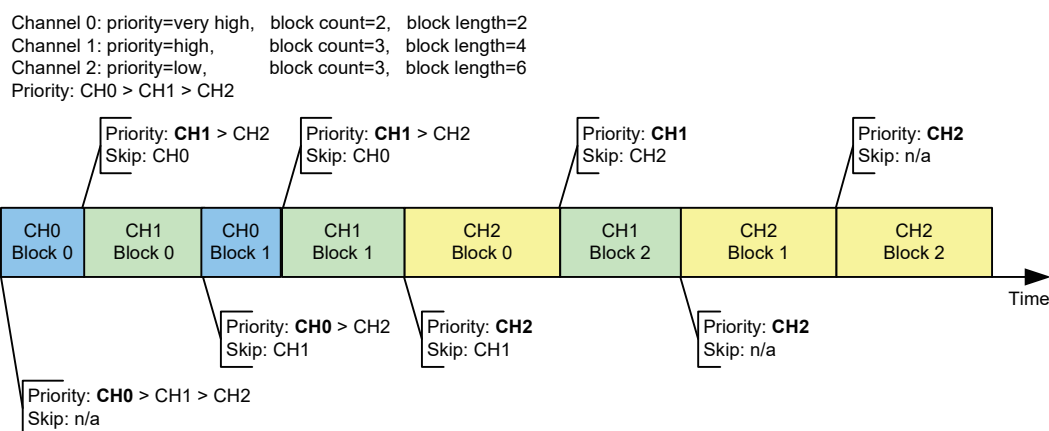


Figure 139. PDMA Channel Arbitration and Scheduling Example

Transfer Request

For a peripheral-to-memory or memory-to-peripheral transfer, one peripheral hardware request will trigger one block transaction of the dedicated PDMA channel. However, a complete data transfer of the relevant dedicated PDMA channel will be triggered when a software request occurs. It is recommended that the PDMA channel is configured to have a lower priority level and a smaller block length which is requested by the software for memory-to-memory data copy applications.

Address Mode

The PDMA provides three kinds of address modes which are the linear address, circular address and fixed address modes. These different address modes are used to support different kinds of source and destination address arrangements. The following table shows the detailed address mode combinations.

Table 61. PDMA Address Modes

Source Address Mode	Destination Address Mode
Linear Increment / Decrement Address	Linear Increment / Decrement Address
Linear Increment / Decrement Address	Circular Increment / Decrement Address
Linear Increment / Decrement Address	Fixed Address
Circular Increment / Decrement Address	Linear Increment / Decrement Address
Circular Increment / Decrement Address	Circular Increment / Decrement Address
Fixed Address	Linear Increment / Decrement Address
Fixed Address	Fixed Address

Linear Address Mode

After data is transferred, the current address will be increased or decreased by 1, 2 or 4 depending upon the data bit width setting.

Circular Address Mode

After data is transferred, the current address will be increased or decreased by 1, 2 or 4 depending upon the data bit width setting. When a block transaction is complete, the current address is loaded with the configured start address.

Fixed Address Mode

After data is transferred, the current address remains unchanged.

Auto-Reload

When the auto-reload control bit, AUTORLn, in the PDMA channel n control register PDMACHnCR is set, both the channel n current address and the channel n current transfer size will be automatically reloaded with the corresponding start value after the current PDMA channel data transfer has totally completed. The channel n will still be activated and the next relative PDMA request can be serviced without any re-configuration using the application software.

Transfer Interrupt

There are five transfer events during which the interrupts can be asserted for each PDMA channel. These are the block transaction end (BE), half transfer (HT), transfer complete (TC), transfer error (TE) and global transfer event (GE). Setting the corresponding control bits in the PDMA interrupt enable register PDMAIER will enable the relevant interrupt events. The global interrupt event, GE, will be generated if any of the four interrupt events including the BE, HT, TC and TE occurs. Clearing the BE, HT, TC or TE event flag will also clear the GE flag. Clearing the GE flag will automatically clear all other event flags. The TE interrupt event will occur when the PDMA accesses a system reserved address space or when the PDMA receives a request but the corresponding transfer size setting is equal to zero.

Register Map

The following table shows the PDMA registers and reset values.

Table 62. PDMA Register Map

Register	Offset	Description	Reset Value
PDMA Channel 0 Registers			
PDMACH0CR	0x000	PDMA Channel 0 Control Register	0x0000_0000
PDMACH0SADR	0x004	PDMA Channel 0 Source Address Register	0x0000_0000
PDMACH0DADR	0x008	PDMA Channel 0 Destination Address Register	0x0000_0000
PDMACH0TSR	0x010	PDMA Channel 0 Transfer Size Register	0x0000_0000
PDMACH0CTSR	0x014	PDMA Channel 0 Current Transfer Size Register	0x0000_0000
PDMA Channel 1 Registers			
PDMACH1CR	0x018	PDMA Channel 1 Control Register	0x0000_0000
PDMACH1SADR	0x01C	PDMA Channel 1 Source Address Register	0x0000_0000
PDMACH1DADR	0x020	PDMA Channel 1 Destination Address Register	0x0000_0000
PDMACH1TSR	0x028	PDMA Channel 1 Transfer Size Register	0x0000_0000
PDMACH1CTSR	0x02C	PDMA Channel 1 Current Transfer Size Register	0x0000_0000
PDMA Channel 2 Registers			
PDMACH2CR	0x030	PDMA Channel 2 Control Register	0x0000_0000
PDMACH2SADR	0x034	PDMA Channel 2 Source Address Register	0x0000_0000
PDMACH2DADR	0x038	PDMA Channel 2 Destination Address Register	0x0000_0000
PDMACH2TSR	0x040	PDMA Channel 2 Transfer Size Register	0x0000_0000
PDMACH2CTSR	0x044	PDMA Channel 2 Current Transfer Size Register	0x0000_0000
PDMA Channel 3 Registers			
PDMACH3CR	0x048	PDMA Channel 3 Control Register	0x0000_0000

Register	Offset	Description	Reset Value
PDMACH3SADR	0x04C	PDMA Channel 3 Source Address Register	0x0000_0000
PDMACH3DADR	0x050	PDMA Channel 3 Destination Address Register	0x0000_0000
PDMACH3TSR	0x058	PDMA Channel 3 Transfer Size Register	0x0000_0000
PDMACH3CTSR	0x05C	PDMA Channel 3 Current Transfer Size Register	0x0000_0000
PDMA Channel 4 Registers			
PDMACH4CR	0x060	PDMA Channel 4 Control Register	0x0000_0000
PDMACH4SADR	0x064	PDMA Channel 4 Source Address Register	0x0000_0000
PDMACH4DADR	0x068	PDMA Channel 4 Destination Address Register	0x0000_0000
PDMACH4TSR	0x070	PDMA Channel 4 Transfer Size Register	0x0000_0000
PDMACH4CTSR	0x074	PDMA Channel 4 Current Transfer Size Register	0x0000_0000
PDMA Channel 5 Registers			
PDMACH5CR	0x078	PDMA Channel 5 Control Register	0x0000_0000
PDMACH5SADR	0x07C	PDMA Channel 5 Source Address Register	0x0000_0000
PDMACH5DADR	0x080	PDMA Channel 5 Destination Address Register	0x0000_0000
PDMACH5TSR	0x088	PDMA Channel 5 Transfer Size Register	0x0000_0000
PDMACH5CTSR	0x08C	PDMA Channel 5 Current Transfer Size Register	0x0000_0000
PDMA Global Register			
PDMAISR	0x120	PDMA Interrupt Status Register	0x0000_0000
PDMAISCR	0x128	PDMA Interrupt Status Clear Register	0x0000_0000
PDMAIER	0x130	PDMA Interrupt Enable Register	0x0000_0000

Register Descriptions

PDMA Channel n Control Register – PDMACHnCR, n = 0 ~ 5

This register is used to specify the PDMA channel n data transfer configuration.

Offset: 0x000 (0), 0x018 (1), 0x030 (2), 0x048 (3), 0x060 (4), 0x078 (5)

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved				AUTORLn	FIXAENn	CHnPRI	
					RW	0	RW	0
	7	6	5	4	3	2	1	0
Type/Reset	SRCAMODn	SRCAINCn	DSTAMODn	DSTAINCn	DWIDTHn		SWTRIGN	CHnEN
	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[11]	AUTORLn	Channel n Auto-Reload Enable Control 0: Disable Auto-Reload function 1: Enable Auto-Reload function If this bit is set to 1 to enable the auto-reload function, both the channel n current address and the channel n current transfer size will be reloaded with the relevant start value and the PDMA channel n will still be activated when a transfer is complete. If this bit is cleared to 0, the channel n current address and the channel n current transfer size will remain unchanged and the PDMA channel n will be disabled after a transfer completion.
[10]	FIXAENn	Channel n Fixed Address Enable control 0: Disable fixed address function in the circular address mode 1: Enable fixed address function in the circular address mode Note that this bit is only available when the source or destination address mode is set to be in the circular address mode. For example, the source address mode is set as in the linear address mode and the destination address mode is set as in the circular mode. If this bit is set to enable the fixed address function, then the source address mode will still be in the linear address but the destination address mode will be in the fixed address mode instead of the circular address mode.
[9:8]	CHnPRI	Channel n Priority 00: Low 01: Medium 10: High 11: Very high The CHnPRI field is used to configure the channel priority using the application program. If there are more than one channel which have the same software configured priority level, the channel with the smaller channel number will have priority to transfer one block of data after the arbitration.

Bits	Field	Descriptions
[7]	SRCAMODn	Channel n Source Address Mode selection 0: Linear address mode 1: Circular address mode In the linear address mode, the current source address value can be increased or decreased, determined by the SRCAINCn bit value during a complete transfer. In the circular address mode, the current source address value can be increased or decreased which is also determined by the SRCAINCn bit value during a block transfer and will be loaded with the lower 16-bit value of the PDMACHnSADR register, which will be regarded as the current source address when a block transaction has completed.
[6]	SRCAINCn	Channel n Source Address Increment control 0: Increment 1: Decrement This bit is used to determine whether the current source address is increased or decreased during a complete transfer in the linear address mode or a block transfer in the circular address mode.
[5]	DSTAMODn	Channel n Destination Address Mode selection 0: Linear address mode 1: Circular address mode In linear address mode, the current destination address value can be increased or decreased, determined by the DSTAINCn bit value during a complete transfer. In the circular address mode, the current destination address value can be increased or decreased which is also determined by the DSTAINCn bit value during a block transfer and will be loaded with the lower 16-bit value of the PDMACHnDADR register, which will be regarded as the current destination address when a block transfer has completed.
[4]	DSTAINCn	Channel n Destination Address Increment Control 0: Increment 1: Decrement This bit is used to determine if the current destination address is increased or decreased during a complete transfer in the linear address mode or a block transfer in the circular address mode.
[3:2]	DWIDTHn	Data Bit Width selection 00: 8-bit 01: 16-bit 10: 32-bit 11: Reserved The field is used to select the data bit width of the corresponding PDMA channel n.
[1]	SWTRIGn	Software Trigger control 0: No operation 1: Software triggered transfer request Setting this bit will generate a memory-to-memory software transfer request on the corresponding PDMA channel n. It is automatically cleared when a transfer has completely finished.

Bits	Field	Descriptions
[0]	CHnEN	Channel n Enable control 0: Disable the PDMA channel n 1: Enable the PDMA channel n Setting this bit will enable a software or hardware transfer request on the PDMA channel n. It is automatically cleared by hardware when a transfer has completed with the auto-reload function being disabled. However, if the AUTORLn bit is set to 1 to enable the auto-reload function, this bit will be remain high to enable the PDMA channel n function for the next transfer request instead of automatically being cleared by hardware after a transfer has finished.

PDMA Channel n Source Address Register – PDMACHnSADR, n = 0 ~ 5

This register specifies the source address of the PDMA channel n.

Offset: 0x004 (0), 0x01C (1), 0x034 (2), 0x04C (3), 0x064 (4), 0x07C (5)

Reset value: 0x0000_0000

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

Bits	Field	Descriptions
[31:0]	SADRn	Channel n Source Address The register is used to specify the 32-bit source address of the PDMA channel n.

PDMA Channel n Destination Address Register – PDMACHnDADR, n = 0 ~ 5

This register specifies the destination address of the PDMA channel n.

Offset: 0x008 (0), 0x020 (1), 0x038 (2), 0x050 (3), 0x068 (4), 0x080 (5)

Reset value: 0x0000_0000

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

Bits	Field	Descriptions
[31:0]	DADRn	Channel n Destination Address The register is used to specify the 32-bit destination address of the PDMA channel n.

PDMA Channel n Transfer Size Register – PDMACHnTSR, n = 0 ~ 5

This register is used to specify the block transaction count and block transaction length.

Offset: 0x010 (0), 0x028 (1), 0x040 (2), 0x058 (3), 0x070 (4), 0x088 (5)

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
	BLKCNTn								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	BLKCNTn								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	Reserved								
Type/Reset									
	7	6	5	4	3	2	1	0	
	BLKLENn								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[31:16]	BLKCNTn	Channel n Block Transaction Count BLKCNTn represents the number of block transactions for a channel n complete transfer. The capacity of a complete transfer is the product of the BLKCNTn and BLKLENn values. The maximum BLKCNTn value is 65,535.
[7:0]	BLKLENn	Channel n Block Length The BLKLENn represents the length of a data block. The data width is defined by the DWIDTHn field in the PDMACHnCR register. The maximum BLKLENn value is 255.

PDMA Channel n Current Transfer Size Register – PDMACHnCTSR, n = 0 ~ 5

This register is used to indicate the current block transaction count.

Offset: 0x014 (0), 0x02C (1), 0x044 (2), 0x05C (3), 0x074 (4), 0x08C (5)

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24								
	CBLKCNTn															
Type/Reset	RO	0	RO	0	RO	0	RO	0	RO	0	RO	0	RO	0	RO	0
	23	22	21	20	19	18	17	16								
	CBLKCNTn															
Type/Reset	RO	0	RO	0	RO	0	RO	0	RO	0	RO	0	RO	0	RO	0
	15	14	13	12	11	10	9	8								
	Reserved															
Type/Reset																
	7	6	5	4	3	2	1	0								
	Reserved															
Type/Reset																

Bits	Field	Descriptions
[31:16]	CBLKCNTn	Channel n Current Block Count The CBLKCNTn field is a 16-bit read-only value indicating the number of data blocks that remain to be transferred. After a data block has transferred completely, the CBLKCNTn value will be decreased by 1. Writing a new value to the BLKCNTn field in the PDMACHnTSR register will update the CBLKCNTn field value.

PDMA Interrupt Status Register – PDMAISR

This register is used to indicate the corresponding interrupt status of the PDMA channel 0 ~ 5.

Offset: 0x120

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved		TEISTA5	TCISTA5	HTISTA5	BEISTA5	GEISTA5	TEISTA4
Type/Reset			RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	23	22	21	20	19	18	17	16
	TCISTA4	HTISTA4	BEISTA4	GEISTA4	TEISTA3	TCISTA3	HTISTA3	BEISTA3
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	15	14	13	12	11	10	9	8
	GEISTA3	TEISTA2	TCISTA2	HTISTA2	BEISTA2	GEISTA2	TEISTA1	TCISTA1
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	7	6	5	4	3	2	1	0
	HTISTA1	BEISTA1	GEISTA1	TEISTA0	TCISTA0	HTISTA0	BEISTA0	GEISTA0
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0

Bits	Field	Descriptions
[29], [24], [19], [14], [9], [4]	TEISTAn	Channel n Transfer Error Interrupt Status (n = 0 ~ 5) 0: No Transfer Error occurs 1: Transfer Error occurs This bit is set by hardware and is cleared by writing a “1” into the corresponding interrupt status clear bit in the PDMAISCR register. A Transfer error will occur when the PDMA accesses a system reserved address space or when the PDMA receives a request but the corresponding transfer capacity is equal to zero.
[28], [23], [18], [13], [8], [3]	TCISTAn	Channel n Transfer Complete Interrupt Status (n = 0 ~ 5) 0: No Transfer Completion Occurs 1: Transfer Completion Occurs This bit is set by hardware and is cleared by writing a “1” into the corresponding interrupt status clear bit in the PDMAISCR register. The Transfer Completion event will occur when the PDMA has completed a data transfer task.
[27], [22], [17], [12], [7], [2]	HTISTAn	Channel n Half Transfer Interrupt Status (n = 0 ~ 5) 0: No Half Transfer Event Occurs 1: Half Transfer Event Occurs This bit is set by hardware and is cleared by writing a “1” into the corresponding interrupt status clear bit in the PDMAISCR register. A Half Transfer event will occur when the PDMA has completed half of the data transfer task.
[26], [21], [16], [11], [6], [1]	BEISTAn	Channel n Block Transaction End Interrupt Status (n = 0 ~ 5) 0: No Block Transaction End Event Occurs 1: Block Transaction End Event Occurs This bit is set by hardware and is cleared by writing a “1” into the corresponding interrupt status clear bit in the PDMAISCR register. A Block Transaction End event will occur when the PDMA completes a data block transaction task.

Bits	Field	Descriptions
[25], [20], [15], [10], [5], [0]	GEISTAn	Channel n Global Transfer Interrupt Status (n = 0 ~ 5) 0: No TE, TC, HT or BE event occurs 1: TE, TC, HT, or BE event occurs This bit is set by hardware and is cleared by writing a “1” into the corresponding interrupt status clear bit, GEICLRn, in the PDMAISR register. A Global Transfer Event will occur if any of the BE, HT, TC and TE events occurs. Also clearing any of the BE, HT, TC and TE event interrupt flags will clear the GE interrupt flag. Note that if a “1” is written into the GEICLRn bit in the PDMAISR register to clear the GE interrupt flag, the BE, HT, TC and TE event interrupt flags will also be cleared to 0 together with the GE interrupt status flag.

PDMA Interrupt Status Clear Register – PDMAISCR

This register is used to clear the corresponding interrupt status bits in the PDMAISR Register.

Offset: 0x128

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
	Reserved		TEICLR5	TCICLR5	HTICLR5	BEICLR5	GEICLR5	TEICLR4	
Type/Reset			WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	
	23	22	21	20	19	18	17	16	
	TCICLR4	HTICLR4	BEICLR4	GEICLR4	TEICLR3	TCICLR3	HTICLR3	BEICLR3	
Type/Reset	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	
	15	14	13	12	11	10	9	8	
	GEICLR3	TEICLR2	TCICLR2	HTICLR2	BEICLR2	GEICLR2	TEICLR1	TCICLR1	
Type/Reset	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	
	7	6	5	4	3	2	1	0	
	HTICLR1	BEICLR1	GEICLR1	TEICLR0	TCICLR0	HTICLR0	BEICLR0	GEICLR0	
Type/Reset	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	

Bits	Field	Descriptions
[29], [24], [19], [14], [9], [4]	TEICLRn	Channel n Transfer Error Interrupt Status Clear (n = 0 ~ 5) 0: No Operation 1: Clear the corresponding TEISTAn bit in the PDMAISR register Writing a “1” into the TEICLRn bit will clear the TEISTAn status bit in the PDMAISR register. This bit will be automatically cleared to 0 after a “1” is written.
[28], [23], [18], [13], [8], [3]	TCICLRn	Channel n Transfer Complete Interrupt Status Clear (n = 0 ~ 5) 0: No Operation 1: Clear the corresponding TCISTAn bit in the PDMAISR register Writing a “1” into the TCICLRn bit will clear the TCISTAn status bit in the PDMAISR register. This bit will be automatically cleared to 0 after a “1” is written.
[27], [22], [17], [12], [7], [2]	HTRICLRn	Channel n Half Transfer Interrupt Status Clear (n = 0 ~ 5) 0: No Operation 1: Clear the corresponding HTISTAn bit in the PDMAISR register Writing a “1” into the HTRICLRn bit will clear the HTISTAn status bit in the PDMAISR register. This bit will be automatically cleared to 0 after a “1” is written.

Bits	Field	Descriptions
[26], [21], [16], [11], [6], [1]	BEICLRn	Channel n Block Transaction End Interrupt Status Clear (n = 0 ~ 5) 0: No Operation 1: Clear the corresponding BEISTAn bit in the PDMAISR register Writing a "1" into the BEICLRn bit will clear the BEISTAn status bit in the PDMAISR register. This bit will automatically cleared to 0 after a data "1" is written.
[25], [20], [15], [10], [5], [0]	GEICLRn	Channel n Global Transfer Event Interrupt Status Clear (n = 0 ~ 5) 0: No Operation 1: Clear the TEISTAn, TCISTAn, HTISTAn, BEISTAn and GEISTAn bits in the PDMAISR register Writing a "1" into the GEICLRn bit will clear the GEISTAn status bit together with the TEISTAn, TCISTAn, HTISTAn and BEISTAn bits in the PDMAISR register. This bit will be automatically cleared to 0 after a "1" is written.

PDMA Interrupt Enable Register – PDMAIER

This register is used to enable or disable the related interrupts of the PDMA channel 0 ~ 5.

Offset: 0x130

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved		TEIE5	TCIE5	HTIE5	BEIE5	GEIE5	TEIE4
Type/Reset			RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	TCIE4	HTIE4	BEIE4	GEIE4	TEIE3	TCIE3	HTIE3	BEIE3
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	GEIE3	TEIE2	TCIE2	HTIE2	BEIE2	GEIE2	TEIE1	TCIE1
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	HTIE1	BEIE1	GEIE1	TEIE0	TCIE0	HTIE0	BEIE0	GEIE0
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[29], [24], [19], [14], [9], [4]	TEIE _n	Channel n Transfer Error Interrupt Enable control (n = 0 ~ 5) 0: Transfer Error interrupt is disabled 1: Transfer Error interrupt is enabled This bit is set and cleared by software.
[28], [23], [18], [13], [8], [3]	TCIE _n	Channel n Transfer Complete Interrupt Enable control (n = 0 ~ 5) 0: Transfer Completion interrupt is disabled 1: Transfer Completion interrupt is enabled This bit is set and cleared by software.
[27], [22], [17], [12], [7], [2]	HTIE _n	Channel n Half Transfer Interrupt Enable control (n = 0 ~ 5) 0: Half Transfer interrupt is disabled 1: Half Transfer interrupt is enabled This bit is set and cleared by software.

Bits	Field	Descriptions
[26], [21], [16], [11], [6], [1]	BEIEn	Channel n Block Transaction End Interrupt Enable control (n = 0 ~ 5) 0: Block Transaction End interrupt is disabled 1: Block Transaction End interrupt is enabled This bit is set and cleared by software.
[25], [20], [15], [10], [5], [0]	GEIEn	Channel n Global Transfer Event Interrupt Enable control (n = 0 ~ 5) 0: Global Transfer Event interrupt is disabled 1: Global Transfer Event interrupt is enabled This bit is set and cleared by software.

23 Divider (DIV)

Introduction

In order to enhance MCU performance, a divider is implemented within the device.

Features

- Signed/unsigned 32-bit divider
- Operation in 8 clock cycles, Load in 1 clock cycle
- Division by zero error flag

Functional Descriptions

The division and modulus functions of the truncated division are related in the following way:

$$A / B = Q \dots R$$

Where “A” is Dividend, “B” is Divisor, “Q” is Quotient and “R” is Remainder. Divider requires a software trigger start signal by controlling the Start bit in the CR register. The divider calculation complete flag will be set to 1 after 8 clock cycles, however, if the divisor register data is zero during the calculation, the division by zero error flag will be set to 1.

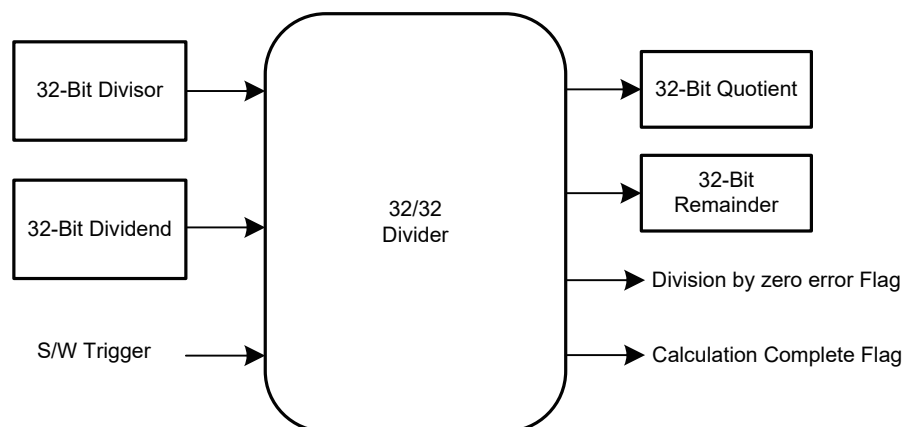


Figure 140. Divider Functional Diagram

Register Map

The following table shows the DIV registers and reset values.

Table 63. DIV Register Map

Register	Offset	Description	Reset Value
CR	0x000	Divider Control Register	0x0000_0008
DDR	0x004	Dividend Data Register	0x0000_0000
DSR	0x008	Divisor Data Register	0x0000_0000
QTR	0x00C	Quotient Data Register	0x0000_0000
RMR	0x010	Remainder Data Register	0x0000_0000

Register Descriptions

Divider Control Register – CR

This register contains the divider calculation complete flag, division by zero error flag and the calculation start control bit.

Offset: 0x000

Reset value: 0x0000_0008

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved				COM	ZEF	Reserved	START
					RO	1	RO	0
								RW
								0

Bits	Field	Descriptions
[3]	COM	Calculation Complete Flag 0: Data are invalid 1: New data are valid When this bit is set to 1 by hardware, it indicates that the divider calculation is completed and data are valid. This bit is cleared to 0 by hardware after the calculation start.
[2]	ZEF	Division By Zero Error Flag 0: Divisor is not zero 1: Divisor is zero This bit is cleared to 0 by hardware after the calculation start.
[0]	START	Calculation Start Control Bit 0: No operation 1: Start the divider calculation Writing 1 to this bit will start the divider calculation.

Dividend Data Register – DDR

The register contains the dividend of the divider.

Offset: 0x004

Reset value: 0x0000_0000

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

Bits	Field	Descriptions
[31:0]	DDR	This bit field is used to specify the dividend of the divider calculation.

Divisor Data Register – DSR

The register contains the divisor of the divider.

Offset: 0x008

Reset value: 0x0000_0000

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

Bits	Field	Descriptions
[31:0]	DSR	This bit field is used to specify the divisor of the divider calculation.

Quotient Data Register – QTR

The register contains the quotient of the divider calculation result.

Offset: 0x00C

Reset value: 0x0000_0000

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

Bits	Field	Descriptions
[31:0]	QTR	This bit field is used to store the quotient of the divider calculation result.

Remainder Data Register – RMR

The register contains the remainder of the divider calculation result.

Offset: 0x010

Reset value: 0x0000_0000

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

Bits	Field	Descriptions
[31:0]	RMR	This bit field is used to store the remainder of the divider calculation result.

24 Cyclic Redundancy Check (CRC)

Introduction

The CRC (Cyclic Redundancy Check) calculation unit is an error detection technique test algorithm and is used to verify data transmission or storage data correctness. A CRC calculation takes a data stream or a block of data as input and generates a 16-bit or 32-bit output remainder. Ordinarily, a data stream is suffixed by a CRC code and used as a checksum when being sent or stored. Therefore, the received or restored data stream is calculated by the same generator polynomial as described above. If the new CRC code result does not match the one calculated earlier, that means data stream contains a data error.

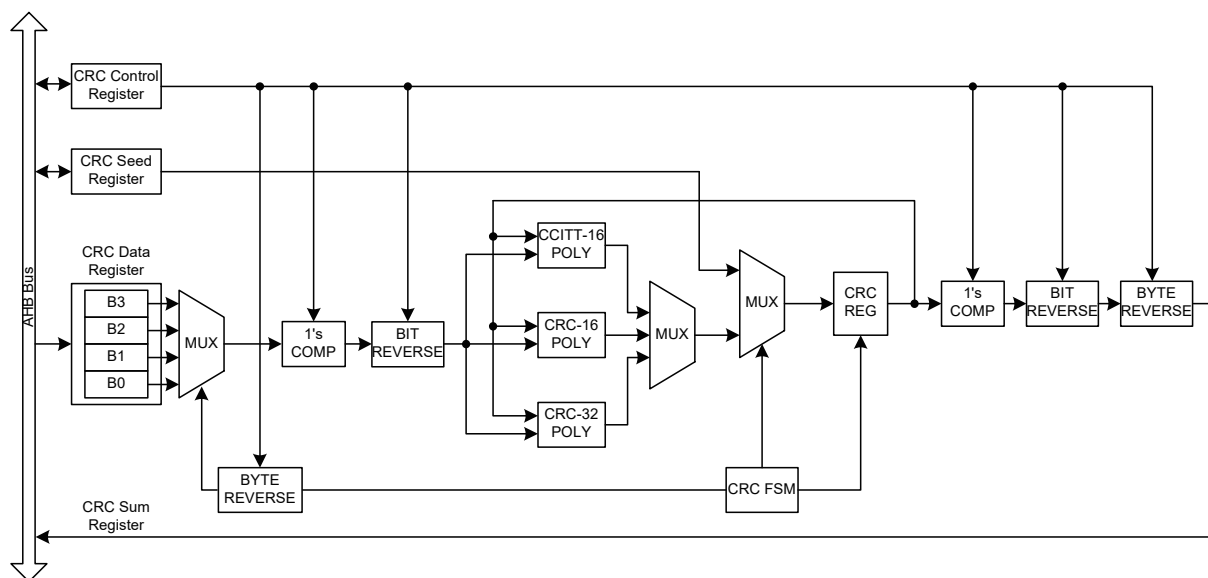


Figure 141. CRC Block Diagram

Features

- Supports CRC16 polynomial: $0x8005$, $X^{16} + X^{15} + X^2 + 1$
- Supports CCITT CRC16 polynomial: $0x1021$, $X^{16} + X^{12} + X^5 + 1$
- Supports IEEE-802.3 CRC32 polynomial: $0x04C11DB7$, $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
- Supports 1's complement, byte reverse and bit reverse operation on data and checksum
- Supports byte, half-word and word data size
- Programmable CRC initial seed value
- CRC computation done in 1 AHB clock cycle for 8-bit data and 4 AHB clock cycles for 32-bit data
- Supports PDMA to complete a CRC computation of a block of memory

Functional Descriptions

This unit only enables the calculation in the CRC16, CCITT CRC16 and IEEE-802.3 CRC32 polynomials. In this unit, the generator polynomial is fixed to the numeric values for those modes; therefore, the CRC value based on other generator polynomials cannot be calculated.

CRC Computation

The CRC calculation unit has a 32-bit write CRC data register (CRCDR) and a read CRC checksum register (CRCCSR). The CRCDR register is used to input new data (write access) and the CRCCSR register is used to hold the result of the previous CRC calculation (read access). Each write operation to the CRCDR register creates a combination of the previous CRC value (stored in CRCCSR) and the new one. The CRC block diagram is shown as Figure 140. The CRC unit calculates the CRC data register (CRCDR) value byte by byte and the default byte and bit order is big-endian. The CRCDR register can be written by word, right-aligned half-word and right-aligned byte. For the other registers only 32-bit access is allowed. The duration of the computation depends on data width:

- 4 AHB clock cycles for 32-bit data input
- 2 AHB clock cycles for 16-bit data input
- 1 AHB clock cycle for 8-bit data input

Byte and Bit Reversal for CRC Computation

The byte reordering and byte-level bit reversal operation can be occurred before the data is used in the CRC calculation or after the CRC checksum output. They are configurable using the corresponding setting field of the CRCCR register. These operations occur on word or half-word writes. The hardware ignores the DATBYRV bit of the CRCCR register during any byte writes but the bit reversal setting DATBIRV are still applied to the byte. Figure 141 shows the byte and bit reversal operation example.

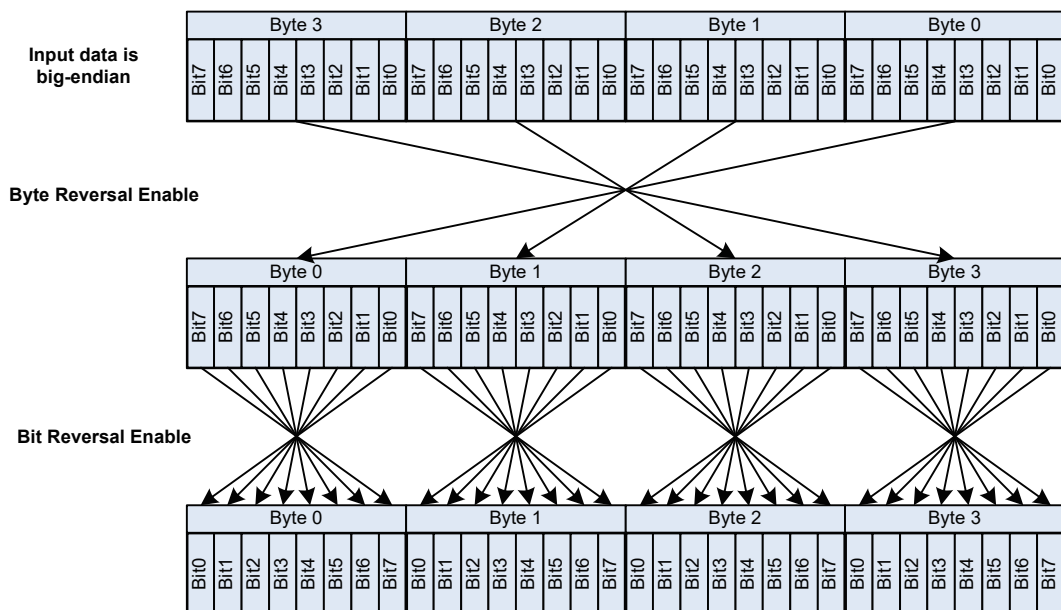


Figure 142. CRC Data Bit and Byte Reversal Example

CRC with PDMA

A PDMA channel with software trigger may be used to transfer data into the CRC unit. If a huge block data needs to be calculated, the recommended PDMA model is to use the PDMA to transfer all available words of data and use software writes to transfer the other remaining bytes. To write data into the CRC unit, the PDMA should use word access method to transfer data from the source location of memory to the CRC data register (CRCDR) in non-incrementing address mode. Then software can write any remaining bytes to the CRC data register (CRCDR) and read the CRC calculation result value from the CRC checksum register (CRCCSR).

Register Map

The following table shows the CRC registers and reset values.

Table 64. CRC Register Map

Register	Offset	Description	Reset Value
CRCCR	0x000	CRC Control Register	0x0000_0000
CRCSDR	0x004	CRC Seed Register	0x0000_0000
CRCCSR	0x008	CRC Checksum Register	0x0000_0000
CRCDR	0x00C	CRC Data Register	0x0000_0000

Register Descriptions

CRC Control Register – CRCCR

This register specifies the corresponding CRC function enable control.

Offset: 0x000

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	SUMCMPL	SUMBYRV	SUMBIRV	DATCMPL	DATBYRV	DATBIRV	POLY	
	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[7]	SUMCMPL	1's Complement operation on Checksum Output 0: Disable 1: Enable
[6]	SUMBYRV	Byte Reverse operation on Checksum Output 0: Disable 1: Enable

Bits	Field	Descriptions
[5]	SUMBIRV	Bit Reverse operation on Checksum Output 0: Disable 1: Enable
[4]	DATCMPL	1's Complement operation on Data 0: Disable 1: Enable
[3]	DATBYRV	Byte Reverse operation on Data 0: Disable 1: Enable
[2]	DATBIRV	Bit Reverse operation on Data 0: Disable 1: Enable
[1:0]	POLY	CRC polynomial 00: CRC-CCITT (0x1021) 01: CRC-16 (0x8005) 1x: CRC-32 (0x04C11DB7)

CRC Seed Register – CRCSDR

This register is used to specify the CRC seed.

Offset: 0x004

Reset value: 0x0000_0000

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

Bits	Field	Descriptions
[31:0]	SEED	CRC Seed Data Put the 16/32-bit seed value in this register according to the polynomial setting in the CRCCR register.

CRC Checksum Register – CRCCSR

This register contains the CRC checksum output.

Offset: 0x008

Reset value: 0x0000_0000

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

Bits	Field	Descriptions
[31:0]	CHKSUM	CRC Checksum Data Get the CRC 16/32-bit checksum result from this register according to the polynomial setting in the CRCCR register after all data are written to the CRCDR register.

CRC Data Register – CRCDR

This register is used to specify the CRC input data.

Offset: 0x00C

Reset value: 0x0000_0000

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

Bits	Field	Descriptions
[31:0]	CRCDATA	CRC Input Data Byte, half-word and word writes are allowed. 1's complement, byte reverse and bit reverse operation can be applied.

25 AES Encrypt/Decrypt Interface (AES)

Introduction

The AES core supports both encryption and decryption functions and supports 128-bit input data. It should be noted that hardware does not pad out any input data bit, therefore users need to do pad action by software at first.

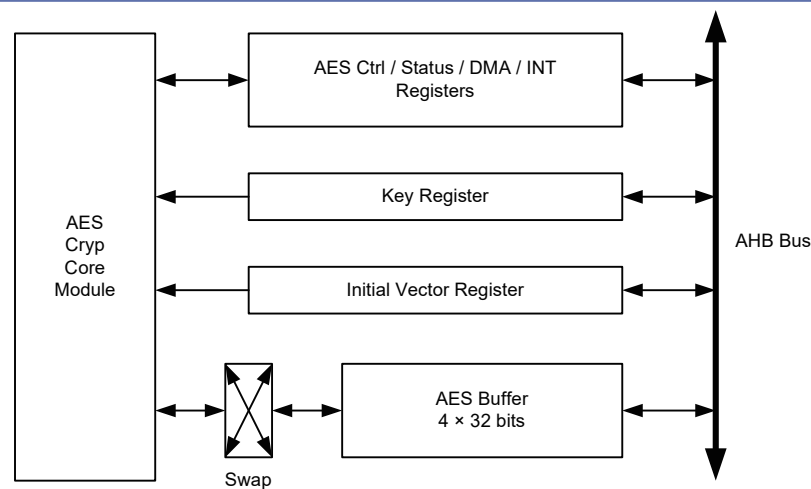


Figure 143. AES Block Diagram

Features

- Supports AES Encrypt / Decrypt functions
- Supports AES ECB/CBC/CTR modes
- Supports Key Size of 128 bits
- Supports 4 words Initial Vector for CBC and CTR modes
- 4 × 32 bits AES data buffer – each IN and OUT FIFO capacity
- Supports Word Data Swap function
- Supports PDMA Interface

Functional Descriptions

AES Mode Description

AES Electronic Codebook (AES-ECB) Mode

The 128-bit plaintext data comes from IN FIFO and will be sent to the AES core to do encryption operation after word swapping operation. The AES core uses a 128-bit key to process the encryption. After encryption, the AES core generates the ciphertext, which will be written into the OUT FIFO after the word swapping operation.

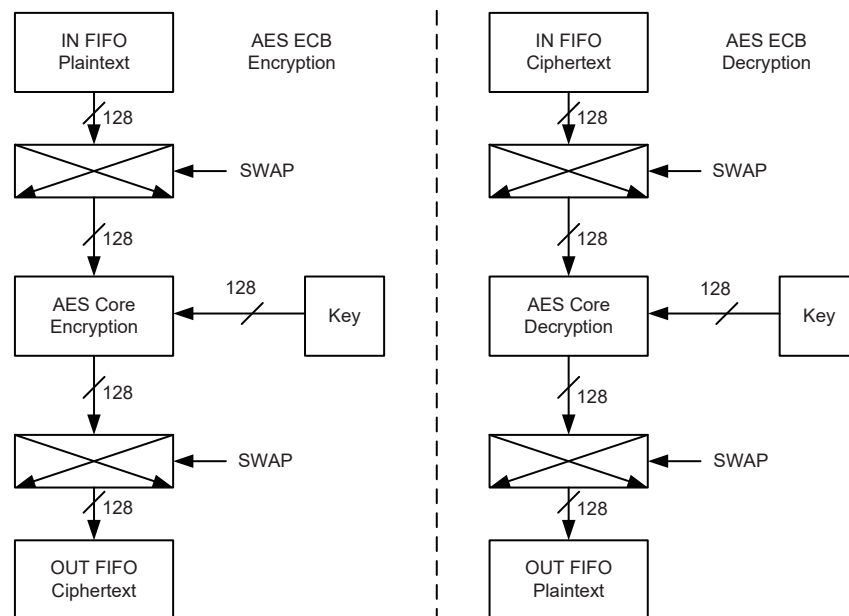


Figure 144. AES-ECB Mode

AES Cipher Block Chaining (AES-CBC) Mode

During encryption in the CBC mode, each block of plaintext is XORed with the previous ciphertext block before being encrypted. The first initial vectors are initialized in the 1st encryption operation. The plaintext after word swapping will be XORed with the initial vectors before encryption. When the encryption output data is pushed into the OUT FIFO, the initial vectors are updated by the encryption output data at the same time.

During decryption in the CBC mode, each block of plaintext is XORed with the previous ciphertext block after being decrypted. The first initial vectors are initialized in the 1st decryption operation. The ciphertext after word swapping and decryption will be XORed with the initial vectors. When the XORed decryption output data is pushed into the OUT FIFO, the initial vectors are updated by the decryption input data at the same time for the next round ciphertext.

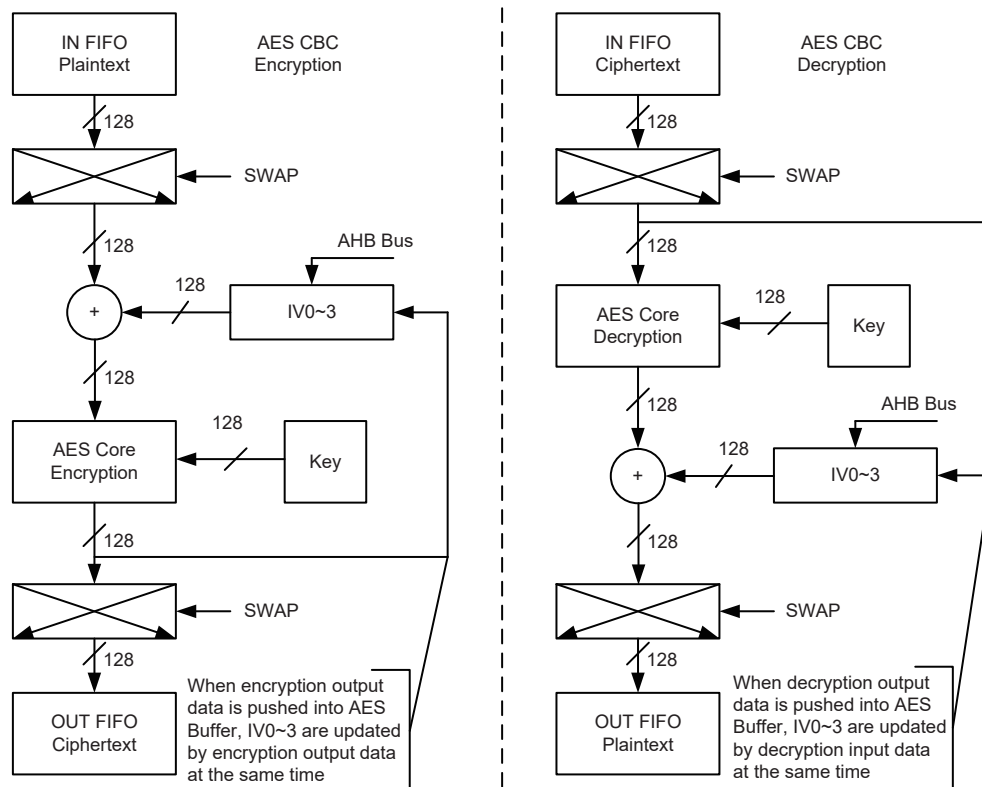


Figure 145. AES-CBC Mode

AES Counter (AES-CTR) Mode

In the CTR mode, the initial vector counter value, after being increased by one, will be sent to the AES core for encryption to generate ciphertext. The AES core uses the same AES direction setting in both encryption and decryption.

During encryption and decryption in the CTR mode, the IN FIFO data after word swapping is XORed with the ciphertext. The XORed data is sent to the OUT FIFO after word swapping. The initial vector counter will be increased by one at the same time for the next round ciphertext.

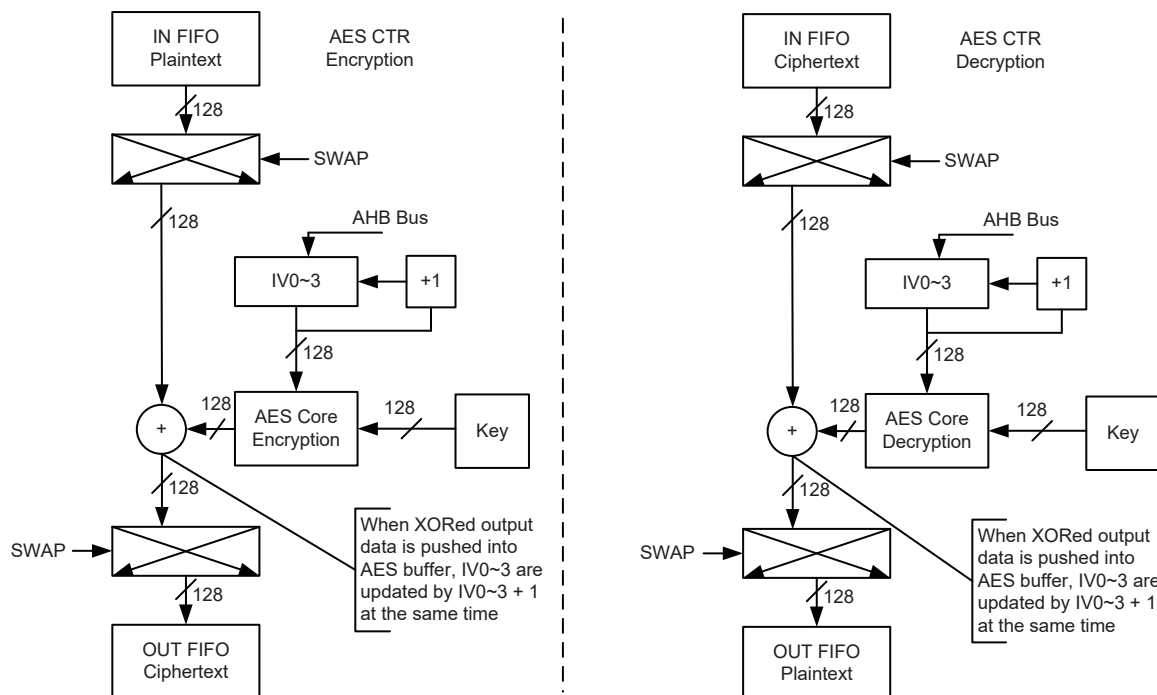


Figure 146. AES-CTR Mode

AES Status

There are five status conditions in the AES for the user to monitor the AES situation. The IFEMPTY bit will be set when the input FIFO is empty while the IFNFULL bit will be set when the input FIFO is not full. The OFNEMPTY bit will be set when there is data in the output FIFO. The OFFULL bit will be set when the output FIFO is full. The BUSY bit will be set when the AES core is executing an encryption/decryption operation or the key is in expansion state.

AES PDMA Interface

The AES supports the 32 bits PDMA data transfer. When the IN FIFO is empty, the AES will send an IN FIFO request to the PDMA. When the OUT FIFO is full, the AES will send an OUT FIFO request to the PDMA.

AES Interrupt

The IFINT request will be generated when the input FIFO is less than 1 AES block (4×32 bits). The OFINT request will be generated when there is data in the AES buffer. When the AES is enabled, an AES interrupt will be generated if the IFINT bit is set with the IFINT interrupt enabled or the OFINT bit is set with the OFINT interrupt enabled.

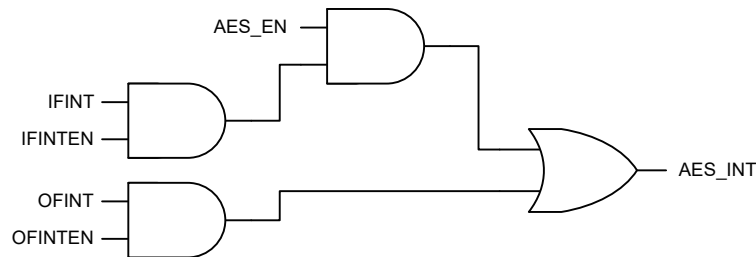


Figure 147. AES Interrupt

AES Initial Vector

The initial vectors (IV0 ~ 3) are not used in the ECB mode. The initial vectors are initialized in the first block of AES input data in the CBC and CTR modes. After the first AES block of input data, the values of the initial vectors will be updated by hardware automatically for the next block of AES input data. The initial vectors in the CTR mode contain nonce, initial vector and counter. The counter will be increased by 1 after every AES data block action.

Nonce (32 bits)	IV0
Initial Vector (64 bits)	IV1, 2
Counter (32 bits)	IV3

Figure 148. Initial Vector for CTR Mode

AES Word Swap

The AES supports a word swap function. The swap action is performed between IN FIFO and AES block data, it is also executed between the AES block data and OUT FIFO. If the word swap function is required, the SWAP bit in the AESCR register should be set high.

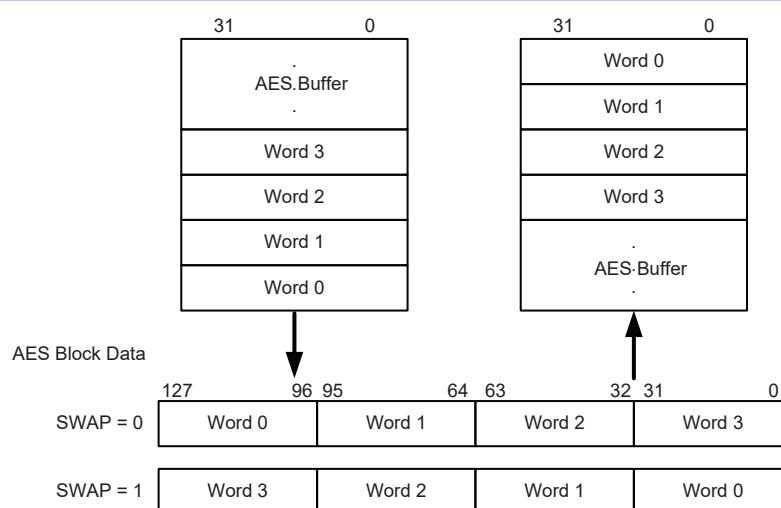


Figure 149. AES Word Swap Function

Register Map

The following table shows the AES registers and reset values.

Table 65. AES Register Map

Register	Offset	Description	Reset Value
AESCR	0x000	AES Control Register	0x0000_0000
AESSR	0x004	AES Status Register	0x0000_0003
AESDMAR	0x008	AES DMA Register	0x0000_0000
AESISR	0x00C	AES Interrupt Status Register	0x0000_0000
AESIER	0x010	AES Interrupt Enable Register	0x0000_0000
AESDINR	0x014	AES Data Input Register	0x0000_0000
AESDOUTR	0x018	AES Data Output Register	0x0000_0000
AESKEYR0	0x01C	AES Key Register 0	0x0000_0000
AESKEYR1	0x020	AES Key Register 1	0x0000_0000
AESKEYR2	0x024	AES Key Register 2	0x0000_0000
AESKEYR3	0x028	AES Key Register 3	0x0000_0000
AESIVR0	0x03C	AES Initial Vector Register 0	0x0000_0000
AESIVR1	0x040	AES Initial Vector Register 1	0x0000_0000
AESIVR2	0x044	AES Initial Vector Register 2	0x0000_0000
AESIVR3	0x048	AES Initial Vector Register 3	0x0000_0000

Register Descriptions

AES Control Register – AESCR

This register specifies the AES control setting.

Offset: 0x000

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved					RW	0	RW
	7	6	5	4	3	2	1	0
Type/Reset	Reserved	KEYSIZE		KEYSTART	MODE		DIR	AESEN
	RO	0	RO	0	RW	0	RW	0

Bits	Field	Descriptions
[10]	FFLUSH	AES IN/OUT FIFO Flush 0: No action 1: Flush FIFO The bit is cleared to 0 by hardware automatically. The bit can be set only in the AES disable state.
[8]	SWAP	AES Data Swap Function 0: No Swap 1: Word Swap
[6:5]	KEYSIZE	AES Key Size 00: 128 bits Others: Reserved
[4]	KEYSTART	AES Key Start 0: Key doesn't Start 1: Key Start It is cleared to 0 by hardware automatically. The bit works when in the AES enable state.
[3:2]	MODE	AES Function Mode 00: ECB mode 01: CBC mode 1x: CTR mode
[1]	DIR	AES Direction 0: Encryption 1: Decryption
[0]	AESEN	AES Enable 0: AES is disabled 1: AES is enabled

AES Status Register – AESSR

This register specifies the AES status.

Offset: 0x004

Reset value: 0x0000_0003

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved			BUSY	OFFULL	OFNEMPTY	IFNFULL	IFEMPTY
				RO	0	RO	0	RO
						0	RO	1
								1

Bits	Field	Descriptions
[4]	BUSY	Busy bit 0: AES is not busy 1: AES is busy AES is busy when AES is executing the encryption/decryption operation or the key is in expansion state.
[3]	OFFULL	Output FIFO is Full 0: Output FIFO is not full 1: Output FIFO is full
[2]	OFNEMPTY	Output FIFO is not Empty 0: Output FIFO is empty 1: Output FIFO is not empty
[1]	IFNFULL	Input FIFO is not Full 0: Input FIFO is full 1: Input FIFO is not full
[0]	IFEMPTY	Input FIFO is Empty 0: Input FIFO is not empty 1: Input FIFO is empty

AES DMA Register – AESDMAR

This register specifies the DMA setting.

Offset: 0x008

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved						OFDMAEN	IFDMAEN
							RW	0 RW 0

Bits	Field	Descriptions
[1]	OFDMAEN	Output FIFO DMA Enable 0: DMA is disabled 1: DMA is enabled
[0]	IFDMAEN	Input FIFO DMA Enable 0: DMA is disabled 1: DMA is enabled

AES Interrupt Status Register – AESISR

The register specifies the interrupt status setting.

Offset: 0x00C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved						OFINT	IFINT
							RO	0 RO 0

Bits	Field	Descriptions
[1]	OFINT	Output FIFO Interrupt Status 0: No Output FIFO Interrupt 1: Output FIFO Interrupt
[0]	IFINT	Input FIFO interrupt Status 0: No Input FIFO Interrupt 1: Input FIFO Interrupt

AES Interrupt Enable Register – AESIER

The register specifies the interrupt enable setting.

Offset: 0x010

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved						OFINTEN	IFINTEN
							RW	0 RW 0

Bits	Field	Descriptions
[1]	OFINTEN	Output FIFO Interrupt Enable bit 0: Interrupt is disabled 1: Interrupt is enabled
[0]	IFINTEN	Input FIFO Interrupt Enable bit 0: Interrupt is disabled 1: Interrupt is enabled

AES DATA Input Register – AESDINR

The register specifies the data input setting.

Offset: 0x014

Reset value: 0x0000_0000

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

Bits	Field	Descriptions
[31:0]	DIN	AES DATA Input 0x0000_0000 ~ 0xFFFF_FFFF

AES DATA Output Register – AESDOUTR

The register specifies the data output setting.

Offset: 0x018

Reset value: 0x0000_0000

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

Bits	Field	Descriptions
[31:0]	DOUT	AES Data Output 0x0000_0000 ~ 0xFFFF_FFFF

AES Key Register n – AESKEYRn, n = 0 ~ 3

The register specifies the data of Key data n.

Offset: 0x01C ~ 0x028

Reset value: 0x0000_0000

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

Bits	Field	Descriptions
[31:0]	KeyData	KeyData 0x0000_0000 ~ 0xFFFF_FFFF

AES Initial Vector Register n – AESIVRn, n = 0 ~ 3

The register specifies the data of Initial Vector data n.

Offset: 0x03C ~ 0x048

Reset value: 0x0000_0000

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

Bits	Field	Descriptions
[31:0]	IVData	Initial Vector Data 0x0000_0000 ~ 0xFFFF_FFFF

Copyright© 2019 by HOLTEK SEMICONDUCTOR INC.

The information appearing in this Data Sheet is believed to be accurate at the time of publication. However, Holtek assumes no responsibility arising from the use of the specifications described. The applications mentioned herein are used solely for the purpose of illustration and Holtek makes no warranty or representation that such applications will be suitable without further modification, nor recommends the use of its products for application that may present a risk to human life due to malfunction or otherwise. Holtek's products are not authorized for use as critical components in life support devices or systems. Holtek reserves the right to alter its products without prior notification. For the most up-to-date information, please visit our web site at <http://www.holtek.com>.