



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

HT32F52342/HT32F52352 User Manual

Revision: V1.30 Date: September 28, 2018

www.holtek.com

Table of Contents

1 Introduction	26
Overview.....	26
Features.....	27
Device Information.....	32
Block Diagram	33
2 Document Conventions	34
3 System Architecture	35
Arm® Cortex®-M0+ Processor.....	35
Bus Architecture.....	36
Memory Organization	37
Memory Map.....	38
Embedded Flash Memory.....	41
Embedded SRAM Memory.....	41
AHB Peripherals	41
APB Peripherals	41
4 Flash Memory Controller (FMC)	42
Introduction	42
Features.....	42
Functional Descriptions	43
Flash Memory Map.....	43
Flash Memory Architecture	44
Wait State Setting	44
Bootling Configuration	45
Page Erase.....	46
Mass Erase.....	47
Word Programming.....	48
Option Byte Description.....	49
Page Erase/Program Protection.....	49
Security Protection.....	51
Register Map	52
Register Descriptions.....	53
Flash Target Address Register – TADR	53
Flash Write Data Register – WRDR	54
Flash Operation Command Register – OCMR	55
Flash Operation Control Register – OPCR.....	56
Flash Operation Interrupt Enable Register – OIER	57
Flash Operation Interrupt and Status Register – OISR	58
Flash Page Erase/Program Protection Status Register – PPSR.....	60
Flash Security Protection Status Register – CPSR	61

Flash Vector Mapping Control Register – VMCR	62
Flash Manufacturer and Device ID Register – MDID.....	63
Flash Page Number Status Register – PNSR	64
Flash Page Size Status Register – PSSR	65
Flash Cache & Pre-fetch Control Register – CFCR.....	66
Custom ID Register n – CIDRn, n = 0 ~ 3	67
5 Power Control Unit (PWRCU)	68
Introduction	68
Features.....	69
Functional Descriptions	69
Backup Domain	69
V _{DD} Power Domain.....	70
1.5 V Power Domain.....	72
Operation Modes	72
Register Map	74
Register Descriptions.....	75
Backup Domain Status Register – BAKSR.....	75
Backup Domain Control Register – BAKCR	76
Backup Domain Test Register – BAKTEST	78
Low Voltage / Brown Out Detect Control and Status Register – LVDCSR	79
Backup Register n – BAKREGn, n = 0 ~ 9	81
6 Clock Control Unit (CKCU)	82
Introduction	82
Features.....	84
Function Descriptions	84
High Speed External Crystal Oscillator – HSE	84
High Speed Internal RC Oscillator – HSI.....	85
Auto Trimming of High Speed Internal RC Oscillator – HSI.....	85
Phase Locked Loop – PLL.....	87
Low Speed External Crystal Oscillator – LSE.....	89
Low Speed Internal RC Oscillator – LSI	89
Clock Ready Flag	89
System Clock (CK_SYS) Selection	90
HSE Clock Monitor	91
Clock Output Capability	91
Register Map	92
Register Descriptions.....	93
Global Clock Configuration Register – GCFGR.....	93
Global Clock Control Register – GCCR.....	95
Global Clock Status Register – GCSR	97
Global Clock Interrupt Register – GCIR.....	98
PLL Configuration Register – PLLCFGR	99

PLL Control Register – PLLCR	99
AHB Configuration Register – AHBCFGR	100
AHB Clock Control Register – AHBCCR	101
APB Configuration Register – APBCFGR.....	103
APB Clock Control Register 0 – APBCCR0.....	104
APB Clock Control Register 1 – APBCCR1.....	106
Clock Source Status Register – CKST	108
APB Peripheral Clock Selection Register 0 – APBPCSR0	109
APB Peripheral Clock Selection Register 1 – APBPCSR1	111
HSI Control Register – HSICR.....	113
HSI Auto Trimming Counter Register – HSIATCR	114
Low Power Control Register – LPCR	115
MCU Debug Control Register – MCUDBGCR.....	116
7 Reset Control Unit (RSTCU)	119
Introduction	119
Functional Descriptions	120
Power On Reset	120
System Reset	120
AHB and APB Unit Reset.....	120
Register Map	121
Register Descriptions.....	121
Global Reset Status Register – GRSR	121
AHB Peripheral Reset Register – AHBPRSTR.....	122
APB Peripheral Reset Register 0 – APBPRSTR0	123
APB Peripheral Reset Register 1 – APBPRSTR1	125
8 General Purpose I/O (GPIO).....	127
Introduction	127
Features.....	128
Functional Descriptions	128
Default GPIO Pin Configuration.....	128
General Purpose I/O – GPIO.....	128
GPIO Locking Mechanism	130
Register Map	130
Register Descriptions.....	131
Port A Data Direction Control Register – PADIRCR	131
Port A Input Function Enable Control Register – PAINER	132
Port A Pull-Up Selection Register – PAPUR	133
Port A Pull-Down Selection Register – PAPDR	134
Port A Open Drain Selection Register – PAODR.....	135
Port A Output Current Drive Selection Register – PADDRVR	136
Port A Lock Register – PALOCKR	137
Port A Data Input Register – PADINR.....	138

Port A Output Data Register – PADOUTR	139
Port A Output Set/Reset Control Register – PASRR	140
Port A Output Reset Register – PARR	141
Port B Data Direction Control Register – PBDIRCR	142
Port B Input Function Enable Control Register – PBINER	143
Port B Pull-Up Selection Register – PBPUR	144
Port B Pull-Down Selection Register – PBPDR	145
Port B Open Drain Selection Register – PBODR	146
Port B Output Current Drive Selection Register – PBDRVR	147
Port B Lock Register – PBLOCKR	148
Port B Data Input Register – PBDINR	149
Port B Output Data Register – PBDOUTR	150
Port B Output Set/Reset Control Register – PBSRR	151
Port B Output Reset Register – PBRR	152
Port C Data Direction Control Register – PCDIRCR	153
Port C Input Function Enable Control Register – PCINER	154
Port C Pull-Up Selection Register – PCPUR	155
Port C Pull-Down Selection Register – PCPDR	156
Port C Open Drain Selection Register – PCODR	157
Port C Output Current Drive Selection Register – PCDRVR	158
Port C Lock Register – PCLOCKR	159
Port C Data Input Register – PCDINR	160
Port C Output Data Register – PCDOUTR	161
Port C Output Set/Reset Control Register – PCSRR	162
Port C Output Reset Register – PCRR	163
Port D Data Direction Control Register – PDDIRCR	164
Port D Input Function Enable Control Register – PDINER	165
Port D Pull-Up Selection Register – PDPUR	166
Port D Pull-Down Selection Register – PDPDR	167
Port D Open Drain Selection Register – PDODR	168
Port D Output Current Drive Selection Register – PDDRVR	169
Port D Lock Register – PDLOCKR	170
Port D Data Input Register – PDDINR	171
Port D Output Data Register – PDDOUTR	172
Port D Output Set/Reset Control Register – PDSRR	173
Port D Output Reset Register – PDRR	174
9 Alternate Function Input/Output Control Unit (AFIO).....	175
Introduction	175
Features	176
Functional Descriptions	176
External Interrupt Pin Selection	176
Alternate Function	177
Lock Mechanism	177
Register Map	177

Register Descriptions.....	178
EXTI Source Selection Register 0 – ESSR0	178
EXTI Source Selection Register 1 – ESSR1	179
GPIO x Configuration Low Register – GPxCFGLR, x = A, B, C, D.....	180
GPIO x Configuration High Register – GPxCFGHR, x = A, B, C, D	181
10 Nested Vectored Interrupt Controller (NVIC).....	182
Introduction	182
Features.....	183
Function Descriptions	184
SysTick Calibration	184
Register Map	184
11 External Interrupt/Event Controller (EXTI).....	185
Introduction	185
Features.....	185
Function Descriptions	186
Wakeup Event Management.....	186
External Interrupt/Event Line Mapping	187
Interrupt and Debounce	187
Register Map	188
Register Descriptions.....	189
EXTI Interrupt Configuration Register n – EXTICFGRn, n = 0 ~ 15	189
EXTI Interrupt Control Register – EXTICR	190
EXTI Interrupt Edge Flag Register – EXTIEDGEFLGR.....	191
EXTI Interrupt Edge Status Register – EXTIEDGESR	192
EXTI Interrupt Software Set Command Register – EXTISSCR.....	193
EXTI Interrupt Wakeup Control Register – EXTIWAKUPCR	194
EXTI Interrupt Wakeup Polarity Register – EXTIWAKUPPOLR	195
EXTI Interrupt Wakeup Flag Register – EXTIWAKUPFLG	196
12 Analog to Digital Converter (ADC)	197
Introduction	197
Features.....	198
Function Descriptions	199
ADC Clock Setup.....	199
Channel Selection.....	199
Conversion Mode.....	199
Start Conversion on External Event.....	202
Sampling Time Setting.....	203
Data Format.....	203
Analog Watchdog.....	203
Interrupts.....	204
PDMA Request	204

Register Map	205
Register Descriptions.....	206
ADC Conversion Control Register – ADCCR	206
ADC Conversion List Register 0 – ADCLST0	208
ADC Conversion List Register 1 – ADCLST1	209
ADC Input Sampling Time Register – ADCSTR	210
ADC Conversion Data Register y – ADCDRy, y = 0 ~ 7	211
ADC Trigger Control Register – ADCTCR	212
ADC Trigger Source Register – ADCTSR.....	213
ADC Watchdog Control Register – ADCWCR	214
ADC Watchdog Threshold Register – ADCTR.....	215
ADC Interrupt Enable Register – ADCIER.....	216
ADC Interrupt Raw Status Register – ADCIRAW	217
ADC Interrupt Status Register – ADCISR.....	218
ADC Interrupt Clear Register – ADCICLR	219
ADC DMA Request Register – ADCDMAR.....	220
13 Comparator (CMP)	221
Introduction	221
Features.....	221
Function Descriptions	222
Comparator Inputs and Output	222
Comparator Voltage Reference	222
Interrupts and Wakeup.....	223
Power Mode and Hysteresis.....	224
Comparator Write-Protected mechanism	224
Register Map	224
Register Descriptions.....	225
Comparator Control Register n – CMPCRn, n = 0 or 1	225
Comparator Voltage Reference Value Register n – CVRVALRn, n = 0 or 1	227
Comparator Interrupt Enable Register n – CMPIERn, n = 0 or 1	228
Comparator Transition Flag Register n – CMPTFRn, n = 0 or 1	229
14 General-Purpose Timer (GPTM)	230
Introduction	230
Features.....	231
Functional Descriptions	232
Counter Mode	232
Clock Controller	235
Trigger Controller	236
Slave Controller	237
Master Controller	240
Channel Controller.....	241
Input Stage	244

Quadrature Decoder	246
Output Stage.....	248
Update Management	252
Single Pulse Mode.....	253
Asymmetric PWM Mode	255
Timer Interconnection	256
Trigger ADC Start.....	259
PDMA Request	259
Register Map	260
Register Descriptions.....	261
Timer Counter Configuration Register – CNTCFR	261
Timer Mode Configuration Register – MDCFR.....	263
Timer Trigger Configuration Register – TRCFR.....	266
Timer Counter Register – CTR	267
Channel 0 Input Configuration Register – CH0ICFR.....	268
Channel 1 Input Configuration Register – CH1ICFR.....	270
Channel 2 Input Configuration Register – CH2ICFR.....	272
Channel 3 Input Configuration Register – CH3ICFR.....	274
Channel 0 Output Configuration Register – CH0OCFR	276
Channel 1 Output Configuration Register – CH1OCFR	278
Channel 2 Output Configuration Register – CH2OCFR	280
Channel 3 Output Configuration Register – CH3OCFR	282
Channel Control Register – CHCTR.....	284
Channel Polarity Configuration Register – CHPOLR.....	285
Timer PDMA/Interrupt Control Register – DICTR.....	286
Timer Event Generator Register – EVGR.....	288
Timer Interrupt Status Register – INTSR.....	290
Timer Counter Register – CNTR.....	293
Timer Prescaler Register – PSCR	294
Timer Counter Reload Register – CRR	295
Channel 0 Capture/Compare Register – CH0CCR	296
Channel 1 Capture/Compare Register – CH1CCR	297
Channel 2 Capture/Compare Register – CH2CCR	298
Channel 3 Capture/Compare Register – CH3CCR	299
Channel 0 Asymmetric Compare Register – CH0ACR.....	300
Channel 1 Asymmetric Compare Register – CH1ACR.....	301
Channel 2 Asymmetric Compare Register – CH2ACR.....	302
Channel 3 Asymmetric Compare Register – CH3ACR.....	303
15 Basic Function Timer (BFTM).....	304
Introduction	304
Features.....	304
Functional Description	305
Repetitive Mode.....	305
One Shot Mode.....	306

Register Map	307
Register Descriptions.....	307
BFTM Control Register – BFTMCR.....	307
BFTM Status Register – BFTMSR.....	308
BFTM Counter Register – BFTMCNTR	309
BFTM Compare Value Register – BFTMCMPR	310
16 Motor Control Timer (MCTM)	311
Introduction	311
Features.....	312
Functional Descriptions	313
Counter Mode	313
Clock Controller	317
Trigger Controller	318
Slave Controller	319
Master Controller	321
Channel Controller.....	322
Input Stage	325
Output Stage.....	327
Update Management	337
Single Pulse Mode.....	339
Asymmetric PWM Mode	341
Timer Interconnection	342
Trigger ADC Start.....	346
Lock Level Table	346
PDMA Request.....	347
Register Map	348
Register Descriptions.....	349
Timer Counter Configuration Register – CNTCFR	349
Timer Mode Configuration Register – MDCFR.....	351
Timer Trigger Configuration Register – TRCFR.....	354
Timer Counter Register – CTR	355
Channel 0 Input Configuration Register – CH0ICFR.....	356
Channel 1 Input Configuration Register – CH1ICFR.....	358
Channel 2 Input Configuration Register – CH2ICFR.....	360
Channel 3 Input Configuration Register – CH3ICFR.....	362
Channel 0 Output Configuration Register – CH0OCFR	364
Channel 1 Output Configuration Register – CH1OCFR	366
Channel 2 Output Configuration Register – CH2OCFR	368
Channel 3 Output Configuration Register – CH3OCFR	370
Channel Control Register – CHCTR.....	372
Channel Polarity Configuration Register – CHPOLR.....	374
Channel Break Configuration Register – CHBRKCFR	376
Channel Break Control Register – CHBRKCTR.....	377

Timer PDMA/Interrupt Control Register – DICTR	379
Timer Event Generator Register – EVGR	381
Timer Interrupt Status Register – INTSR	383
Timer Counter Register – CNTR	386
Timer Prescaler Register – PSCR	387
Timer Counter Reload Register – CRR	388
Timer Repetition Register – REPR	389
Channel 0 Capture/Compare Register – CH0CCR	390
Channel 1 Capture/Compare Register – CH1CCR	391
Channel 2 Capture/Compare Register – CH2CCR	392
Channel 3 Capture/Compare Register – CH3CCR	393
Channel 0 Asymmetric Compare Register – CH0ACR	394
Channel 1 Asymmetric Compare Register – CH1ACR	395
Channel 2 Asymmetric Compare Register – CH2ACR	396
Channel 3 Asymmetric Compare Register – CH3ACR	397
17 Single-Channel Timer (SCTM)	398
Introduction	398
Features	399
Functional Descriptions	399
Counter Mode	399
Clock Controller	400
Trigger Controller	401
Slave Controller	402
Channel Controller	404
Input Stage	406
Output Stage	407
Update Management	409
Register Map	410
Register Descriptions	411
Timer Counter Configuration Register – CNTCFR	411
Timer Mode Configuration Register – MDCFR	412
Timer Trigger Configuration Register – TRCFR	413
Timer Counter Register – CTR	414
Channel Input Configuration Register – CHICFR	415
Channel Output Configuration Register – CHOCFR	417
Channel Control Register – CHCTR	418
Channel Polarity Configuration Register – CHPOLR	419
Timer Interrupt Control Register – DICTR	420
Timer Event Generator Register – EVGR	421
Timer Interrupt Status Register – INTSR	422
Timer Counter Register – CNTR	423
Timer Prescaler Register – PSCR	424
Timer Counter Reload Register – CRR	425
Channel Capture/Compare Register – CHCCR	426

18 Real Time Clock (RTC)	427
Introduction	427
Features	427
Functional Descriptions	428
RTC Related Register Reset	428
Reading RTC Register	428
Low Speed Clock Configuration	428
RTC Counter Operation	429
Interrupt and Wakeup Control	429
RTCOUT Output Pin Configuration	430
Register Map	431
Register Descriptions	431
RTC Counter Register – RTCCNT	431
RTC Compare Register – RTCCMP	432
RTC Control Register – RTCCR	433
RTC Status Register – RTCSR	435
RTC Interrupt and Wakeup Enable Register – RTCIWEN	436
19 Watchdog Timer (WDT)	437
Introduction	437
Features	437
Functional Description	438
Register Map	440
Register Descriptions	440
Watchdog Timer Control Register – WDTCR	440
Watchdog Timer Mode Register 0 – WDTMR0	441
Watchdog Timer Mode Register 1 – WDTMR1	442
Watchdog Timer Status Register – WDTSR	443
Watchdog Timer Protection Register – WDTPR	444
Watchdog Timer Clock Selection Register – WDTCSR	445
20 Inter-Integrated Circuit (I²C)	446
Introduction	446
Features	447
Functional Descriptions	447
Two Wire Serial Interface	447
START and STOP Conditions	447
Data Validity	448
Addressing Format	449
Data Transfer and Acknowledge	451
Clock Synchronization	452
Arbitration	452
General Call Addressing	453

Bus Error.....	453
Address Mask Enable.....	453
Address Snoop.....	453
Operation Mode.....	453
Conditions of Holding SCL Line.....	459
I ² C Timeout Function.....	460
PDMA Interface.....	460
Register Map.....	461
Register Descriptions.....	462
I ² C Control Register – I2CCR.....	462
I ² C Interrupt Enable Register – I2CIER.....	464
I ² C Address Register – I2CADDR.....	466
I ² C Status Register – I2CSR.....	467
I ² C SCL High Period Generation Register – I2CSHPGR.....	470
I ² C SCL Low Period Generation Register – I2CSLPGR.....	471
I ² C Data Register – I2CDR.....	472
I ² C Target Register – I2CTAR.....	473
I ² C Address Mask Register – I2CADDRMR.....	474
I ² C Address Snoop Register – I2CADDRSR.....	475
I ² C Timeout Register – I2CTOUT.....	476
21 Serial Peripheral Interface (SPI).....	477
Introduction.....	477
Features.....	478
Function Descriptions.....	478
Master Mode.....	478
Slave Mode.....	478
SPI Serial Frame Format.....	479
Status Flags.....	483
Register Map.....	486
Register Descriptions.....	486
SPI Control Register 0 – SPICR0.....	486
SPI Control Register 1 – SPICR1.....	488
SPI Interrupt Enable Register – SPIIER.....	490
SPI Clock Prescaler Register – SPICPR.....	491
SPI Data Register – SPIDR.....	492
SPI Status Register – SPISR.....	493
SPI FIFO Control Register – SPIFCR.....	495
SPI FIFO Status Register – SPIFSR.....	496
SPI FIFO Time Out Counter Register – SPIFTOCR.....	497
22 Universal Synchronous Asynchronous Receiver Transmitter (USART).....	498
Introduction.....	498
Features.....	499

Function Descriptions	500
Serial Data Format.....	500
Baud Rate Generation	501
Hardware Flow Control	502
IrDA.....	504
RS485 Mode.....	506
Synchronous Master Mode.....	509
Interrupts and Status	511
PDMA Interface.....	511
Register Map	511
Register Descriptions.....	512
USART Data Register – USRDR.....	512
USART Control Register – USRCR.....	513
USART FIFO Control Register – USRFCR.....	515
USART Interrupt Enable Register – USRIER.....	516
USART Status & Interrupt Flag Register – USRSIFR.....	518
USART Timing Parameter Register – USRTPR.....	520
USART IrDA Control Register – IrDACR.....	521
USART RS485 Control Register – RS485CR.....	522
USART Synchronous Control Register – SYNCR.....	523
USART Divider Latch Register – USRDLR.....	524
USART Test Register – USRTSTR.....	525
23 Universal Asynchronous Receiver Transmitter (UART).....	526
Introduction	526
Features.....	527
Function Descriptions	527
Serial Data Format.....	527
Baud Rate Generation	528
Interrupts and Status	529
PDMA Interface.....	529
Register Map	530
Register Descriptions.....	530
UART Data Register – URDR.....	530
UART Control Register – URCR.....	531
UART Interrupt Enable Register – URIER.....	533
UART Status & Interrupt Flag Register – URSIFR.....	534
UART Divider Latch Register – URDLR.....	536
UART Test Register – URTSTR.....	537
24 Smart Card Interface (SCI).....	538
Introduction	538
Features.....	539
Functional Descriptions	539

Elementary Time Unit Counter.....	539
Guard Time Counter	541
Waiting Time Counter	542
Card Clock and Data Selection.....	543
Card Detection	543
SCI Data Transfer Mode.....	544
Interrupt Generator	546
PDMA Interface.....	547
Register Map	547
Register Descriptions.....	548
SCI Control Register – CR.....	548
SCI Status Register – SR	550
SCI Contact Control Register – CCR.....	552
SCI Elementary Time Unit Register – ETUR	553
SCI Guard Time Register – GTR	554
SCI Waiting Time Register – WTR.....	555
SCI Interrupt Enable Register – IER.....	556
SCI Interrupt Pending Register – IPR.....	558
SCI Transmit Buffer – TXB.....	560
SCI Receive Buffer – RXB.....	560
SCI Prescaler Register – PSCR	561
25 USB Device Controller (USB)	562
Introduction	562
Features.....	562
Functional Descriptions	563
Endpoints.....	563
EP-SRAM	563
Serial Interface Engine – SIE.....	564
Double-Buffering.....	564
Suspend Mode and Wake-up	566
Remote Wake-up	566
Register Map	566
Register Descriptions.....	568
USB Control and Status Register – USBCSR	568
USB Interrupt Enable Register – USBIER	570
USB Interrupt Status Register – USBISR.....	571
USB Frame Count Register – USBFCR	573
USB Device Address Register – USBDEVA	574
USB Endpoint 0 Control and Status Register – USBEP0CSR	575
USB Endpoint 0 Interrupt Enable Register – USBEP0IER	576
USB Endpoint 0 Interrupt Status Register – USBEP0ISR	578
USB Endpoint 0 Transfer Count Register – USBEP0TCR	579
USB Endpoint 0 Configuration Register – USBEP0CFG	580

USB Endpoint 1 ~ 3 Control and Status Register – USBEPnCSR, n = 1 ~ 3.....	581
USB Endpoint 1 ~ 3 Interrupt Enable Register – USBEPnIER, n = 1 ~ 3.....	582
USB Endpoint 1 ~ 3 Interrupt Status Register – USBEPnISR, n = 1 ~ 3.....	583
USB Endpoint 1 ~ 3 Transfer Count Register – USBEPnTCR, n = 1 ~ 3.....	584
USB Endpoint 1 ~ 3 Configuration Register – USBEPnCFGR, n = 1 ~ 3.....	585
USB Endpoint 4 ~ 7 Control and Status Register – USBEPnCSR, n = 4 ~ 7.....	586
USB Endpoint 4 ~ 7 Interrupt Enable Register – USBEPnIER, n = 4 ~ 7.....	589
USB Endpoint 4 ~ 7 Interrupt Status Register – USBEPnISR, n = 4 ~ 7.....	590
USB Endpoint 4 ~ 7 Transfer Count Register – USBEPnTCR, n = 4 ~ 7.....	591
USB Endpoint 4 ~ 7 Configuration Register – USBEPnCFGR, n = 4 ~ 7.....	592
26 Peripheral Direct Memory Access (PDMA).....	593
Introduction	593
Features.....	593
Functional Description	594
AHB Master	594
PDMA Channel	594
PDMA Request Mapping	594
Channel transfer	596
Channel Priority	596
Transfer Request	597
Address Mode.....	597
Auto-Reload.....	597
Transfer Interrupt	598
Register Map	598
Register Descriptions.....	600
PDMA Channel n Control Register – PDMACHnCR, n = 0 ~ 5.....	600
PDMA Channel n Source Address Register – PDMACHnSADR, n = 0 ~ 5.....	602
PDMA Channel n Destination Address Register – PDMACHnDADR, n=0~5.....	603
PDMA Channel n Transfer Size Register – PDMACHnTSR, n = 0 ~ 5.....	604
PDMA Channel n Current Transfer Size Register – PDMACHnCTSR, n=0~5.....	605
PDMA Interrupt Status Register – PDMAISR	606
PDMA Interrupt Status Clear Register – PDMAISCR	607
PDMA Interrupt Enable Register – PDMAIER	609
27 Extend Bus Interface (EBI).....	610
Introduction	610
Features.....	610
Function Descriptions	611
Non-multiplexed 8-bit Data 8-bit Address Mode	612
Non-multiplexed 16-bit Data N-bit Address Mode.....	613
Multiplexed 16-bit Data, 16-bit Address Mode	614
Multiplexed 8-bit Data, 20-bit Address Mode	615
Write Buffer and EBI Status.....	616

Bus Turn-around and Idle Cycles	616
AHB Transaction Width Conversion	617
EBI Bank Access	619
PDMA Request	620
Register Map	620
Register Descriptions.....	620
EBI Control Register – EBICR.....	620
EBI Status Register – EBISR.....	622
EBI Address Timing Register – EBIATR	623
EBI Read Timing Register – EBIRTR	624
EBI Write Timing Register – EBIWTR.....	625
EBI Parity Register – EBIPR.....	626
28 Inter-IC Sound (I²S)	627
Introduction	627
Features.....	627
Functional Description	628
I ² S Master and Slave Mode	628
I ² S Clock Rate Generator	629
I ² S Interface Format.....	631
FIFO Control and Arrangement	638
PDMA and Interrupt	639
Register Map	639
Register Descriptions.....	640
I ² S Control Register – I2SCR.....	640
I ² S Interrupt Enable Register – I2SIER.....	642
I ² S Clock Divider Register – I2SCDR	643
I ² S TX Data Register – I2STXDR	644
I ² S RX Data Register – I2SRXDR.....	644
I ² S FIFO Control Register – I2SFCR	645
I ² S Status Register – I2SSR	646
I ² S Rate Counter Value Register – I2SRCNTR	648
29 Cyclic Redundancy Check (CRC).....	649
Introduction	649
Features.....	650
Function Descriptions	650
CRC Computation.....	650
Byte and Bit Reversal for CRC Computation.....	650
CRC with PDMA	651
Register Map	651
Register Descriptions.....	652
CRC Control Register – CRCCR	652
CRC Seed Register – CRCSDR.....	653

CRC Checksum Register – CRCCSR 654
CRC Data Register – CRCDR 655

List of Tables

Table 1. Series Features and Peripheral List.....	32
Table 2. Document Conventions.....	34
Table 3. Register Map.....	39
Table 4. Flash Memory and Option Byte.....	44
Table 5. Relationship Between Wait State Cycle and HCLK.....	44
Table 6. Booting Modes.....	45
Table 7. Option Byte Memory Map.....	49
Table 8. Access Permission of Protected Main Flash Page.....	50
Table 9. Access Permission When Security Protection is Enabled.....	51
Table 10. FMC Register Map.....	52
Table 11. Operation Mode Definitions.....	72
Table 12. Enter/Exit Power Saving Modes.....	73
Table 13. Power Status After System Reset.....	74
Table 14. PWRCU Register Map.....	74
Table 15. Output Divider2 Value Mapping.....	88
Table 16. Feedback Divider2 Value Mapping.....	88
Table 17. CKOUT Clock Source.....	91
Table 18. CKCU Register Map.....	92
Table 19. RSTCU Register Map.....	121
Table 20. AFIO, GPIO and IO Pad Control Signal True Table.....	129
Table 21. GPIO Register Map.....	130
Table 22. AFIO Selection for Peripheral Map Example.....	177
Table 23. AFIO Register Map.....	177
Table 24. Exception Types.....	182
Table 25. NVIC Register Map.....	184
Table 26. EXTI Register Map.....	188
Table 27. Data format in ADCDR [15:0].....	203
Table 28. A/D Converter Register Map.....	205
Table 29. CMP Register Map.....	224
Table 30. Counting Direction and Encoding Signals.....	247
Table 31. Compare Match Output Setup.....	248
Table 32. GPTM Register Map.....	260
Table 33. GPTM Internal Trigger Connection.....	266
Table 34. BFTM Register Map.....	307
Table 35. Compare Match Output Setup.....	328
Table 36. Output Control Bits for Complementary Output with a Break Event Occurrence.....	336
Table 37. Lock Level Table.....	346
Table 38. MCTM Register Map.....	348
Table 39. MCTM Internal Trigger Connection.....	354

Table 40. Compare Match Output Setup	407
Table 41. SCTM Register Map.....	410
Table 42. LSE Startup Mode Operating Current and Startup Time.....	428
Table 43. RTCOUT Output Mode and Active Level Setting	430
Table 44. RTC Register Map.....	431
Table 45. Watchdog Timer Register Map.....	440
Table 46. Conditions of Holding SCL line.....	459
Table 47. I ² C Register Map	461
Table 48. I ² C Clock Setting Example	471
Table 49. SPI Interface Format Setup.....	479
Table 50. SPI Mode Fault Trigger Conditions	484
Table 51. SPI Master Mode SEL Pin Status	484
Table 52. SPI Register Map	486
Table 53. Baud Rate Deviation Error Calculation – CK_USART = 40 MHz.....	501
Table 54. Baud Rate Deviation Error Calculation – CK_USART = 48 MHz.....	502
Table 55. USART Register Map.....	511
Table 56. Baud Rate Deviation Error Calculation – CK_UART = 40 MHz	528
Table 57. Baud Rate Deviation Error Calculation – CK_UART = 48 MHz	529
Table 58. UART Register Map	530
Table 59. DI Field Based Di Encoded Decimal Values	540
Table 60. FI Field Based Fi Encoded Decimal Values	540
Table 61. Possible ETU Values Obtained with the Fi/Di Ratio.....	540
Table 62. SCI Register Map	547
Table 63. Endpoint Characteristics	563
Table 64. USB Data Types and Buffer Size	563
Table 65. USB Register Map	566
Table 66. Resume Event Detection	569
Table 67. PDMA Channel Assignments	595
Table 68. PDMA Address Modes	597
Table 69. PDMA Register Map.....	598
Table 70. EBI Maps AHB Transactions Width to External Device Transactions.....	618
Table 71. EBI Maps AHB Transactions Width to External Device Transactions Width.....	618
Table 72. EBI Register Map	620
Table 73. Recommend FS List @ 8 MHz PCLK	630
Table 74. Recommend FS List @ 48 MHz PCLK	630
Table 75. I ² S Register Map	639
Table 76. CRC Register Map	651

List of Figures

Figure 1. Block Diagram	33
Figure 2. Cortex®-M0+ Block Diagram.....	36
Figure 3. Bus Architecture	37
Figure 4. Memory Map.....	38
Figure 5. Flash Memory Controller Block Diagram	42
Figure 6. Flash Memory Map.....	43
Figure 7. Vector Remapping	45
Figure 8. Page Erase Operation Flowchart	46
Figure 9. Mass Erase Operation Flowchart	47
Figure 10. Word Programming Operation Flowchart	48
Figure 11. PWRCU Block Diagram	68
Figure 12. Power On Reset / Power Down Reset Waveform	71
Figure 13. CKCU Block Diagram	83
Figure 14. External Crystal, Ceramic, and Resonators for HSE.....	84
Figure 15. HSI Auto Trimming Block Diagram	86
Figure 16. PLL Block Diagram	87
Figure 17. External Crystal, Ceramic, and Resonators for LSE	89
Figure 18. RSTCU Block Diagram.....	119
Figure 19. Power On Reset Sequence	120
Figure 20. GPIO Block Diagram	127
Figure 21. AFIO/GPIO Control Signal.....	129
Figure 22. AFIO Block Diagram	175
Figure 23. EXTI Channel Input Selection	176
Figure 24. EXTI Block Diagram	185
Figure 25. EXTI Wake-up Event Management	186
Figure 26. EXTI Interrupt Debounce Function	187
Figure 27. ADC Block Diagram.....	197
Figure 28. One Shot Conversion Mode	200
Figure 29. Continuous Conversion Mode	200
Figure 30. Discontinuous Conversion Mode.....	202
Figure 31. Comparator Block Diagram	221
Figure 32. 6-Bit Scaler for Comparator Voltage Reference Block Diagram.....	222
Figure 33. Interrupt Signals of Comparators.....	223
Figure 34. Wakeup Signals of Comparators	223
Figure 35. GPTM Block Diagram.....	230
Figure 36. Up-counting Example	232
Figure 37. Down-counting Example.....	233
Figure 38. Center-aligned Counting Example.....	234
Figure 39. GPTM Clock Selection Source	235

Figure 40. Trigger Controller Block	236
Figure 41. Slave Controller Diagram	237
Figure 42. GPTM in Restart Mode	237
Figure 43. GPTM in Pause Mode	238
Figure 44. GPTM in Trigger Mode	239
Figure 45. Master GPTMn and Slave GPTMm/MCTMm Connection	240
Figure 46. MTO Selection	240
Figure 47. Capture/Compare Block Diagram	241
Figure 48. Input Capture Mode	242
Figure 49. PWM Pulse Width Measurement Example	243
Figure 50. Channel 0 and Channel 1 Input Stages	244
Figure 51. Channel 2 and Channel 3 Input Stages	245
Figure 52. TI0 Digital Filter Diagram with N = 2	245
Figure 53. Input Stage and Quadrature Decoder Block Diagram	246
Figure 54. Both TI0 and TI1 Quadrature Decoder Counting	247
Figure 55. Output Stage Block Diagram	248
Figure 56. Toggle Mode Channel Output Reference Signal (CHxPRE = 0)	249
Figure 57. Toggle Mode Channel Output Reference Signal (CHxPRE = 1)	249
Figure 58. PWM Mode Channel Output Reference Signal and Counter in Up-counting Mode	250
Figure 59. PWM Mode Channel Output Reference Signal and Counter in Down-counting Mode	250
Figure 60. PWM Mode Channel Output Reference Signal and Counter in Centre-align Mode	251
Figure 61. Update Event Setting Diagram	252
Figure 62. Single Pulse Mode	253
Figure 63. Immediate Active Mode Minimum Delay	254
Figure 64. Asymmetric PWM mode versus center align counting mode	255
Figure 65. Pausing GPTM1 using the GPTM0 CH0OREF Signal	256
Figure 66. Triggering GPTM1 with GPTM0 Update Event	257
Figure 67. Trigger GPTM0 and GPTM1 with the GPTM0 CH0 Input	258
Figure 68. GPTM PDMA Mapping Diagram	259
Figure 69. BFTM Block Diagram	304
Figure 70. BFTM – Repetitive Mode	305
Figure 71. BFTM – One Shot Mode	306
Figure 72. BFTM – One Shot Mode Counter Updating	306
Figure 73. MCTM Block Diagram	311
Figure 74. Up-counting Example	313
Figure 75. Down-counting Example	314
Figure 76. Center-aligned Counting Example	315
Figure 77. Update Event 1 Dependent Repetition Mechanism Example	316
Figure 78. MCTM Clock Selection Source	317
Figure 79. Trigger Controller Block	318
Figure 80. Slave Controller Diagram	319

Figure 81. MCTM in Restart Mode	319
Figure 82. MCTM in Pause Mode.....	320
Figure 83. MCTM in Trigger Mode.....	320
Figure 84. Master MCTMn and Slave GPTMm Connection	321
Figure 85. MTO Selection	321
Figure 86. Capture/Compare Block Diagram.....	322
Figure 87. Input Capture Mode.....	323
Figure 88. PWM Pulse Width Measurement Example.....	324
Figure 89. Channel 0 and Channel 1 Input Stages.....	325
Figure 90. Channel 2 and Channel 3 Input Stages.....	325
Figure 91. TIO Digital Filter Diagram with N = 2	326
Figure 92. Output Stage Block Diagram	327
Figure 93. Toggle Mode Channel Output Reference Signal – CHxPRE = 0.....	328
Figure 94. Toggle Mode Channel Output Reference Signal – CHxPRE = 1.....	329
Figure 95. PWM Mode Channel Output Reference Signal and Counter in Up-counting Mode	329
Figure 96. PWM Mode Channel Output Reference Signal and Counter in Down-counting Mode	330
Figure 97. PWM Mode 1 Channel Output Reference Signal and Counter in Centre-aligned Counting Mode.....	330
Figure 98. Dead-time Insertion Performed for Complementary Outputs	331
Figure 99. MCTM Break Signal Bolck Diagram	332
Figure 100. MT_BRK Pin Digital Filter Diagram with N = 2	332
Figure 101. Channel 3 Output with a Break Event Occurrence.....	333
Figure 102. Channel 0 ~ 2 Complementary Outputs with a Break Event Occurrence.....	334
Figure 103. Channel 0 ~ 2 Only One Output Enabled when Fault Event Occurs.....	334
Figure 104. Hardware Protection When Both CHxO and CHxNO Are in Active Condition.....	335
Figure 105. Update Event 1 Setup Diagram	337
Figure 106. CHxE, CHxNE and CHxOM Updated by Update Event 2	338
Figure 107. Update Event 2 Setup Diagram	338
Figure 108. Single Pulse Mode.....	339
Figure 109. Immediate Active Mode Minimum Delay	340
Figure 110. Asymmetric PWM Mode versus Center-aligned Counting Mode	341
Figure 111. Pausing GPTM using the MCTM CH0OREF Signal	342
Figure 112. Triggering GPTM with MCTM Update Event 1.....	343
Figure 113. Trigger MCTM and GPTM with the MCTM CH0 Input	344
Figure 114. CH1XOR Input as Hall Sensor Interface.....	345
Figure 115. MCTM PDMA Mapping Diagram.....	347
Figure 116. SCTM Block Diagram	398
Figure 117. Up-counting Example.....	399
Figure 118. SCTM Clock Selection Source.....	400
Figure 119. Trigger Controller Block	401
Figure 120. Slave Controller Diagram	402

Figure 121. SCTM in Restart Mode	402
Figure 122. SCTM in Pause Mode	403
Figure 123. SCTM in Trigger Mode	403
Figure 124. Capture/Compare Block Diagram.....	404
Figure 125. Input Capture Mode.....	405
Figure 126. Channel Input Stages	406
Figure 127. TI Digital Filter Diagram with N = 2.....	406
Figure 128. Output Stage Block Diagram	407
Figure 129. Toggle Mode Channel Output Reference Signal (CHPRE = 0)	408
Figure 130. Toggle Mode Channel Output Reference Signal (CHPRE = 1)	408
Figure 131. PWM Mode Channel Output Reference Signal	409
Figure 132. Update Event Setting Diagram	410
Figure 133. RTC Block Diagram.....	427
Figure 134. Watchdog Timer Block Diagram	437
Figure 135. Watchdog Timer Behavior	439
Figure 136. I ² C Module Block Diagram.....	446
Figure 137. START and STOP Condition	448
Figure 138. Data Validity.....	448
Figure 139. 7-bit Addressing Mode.....	449
Figure 140. 10-bit Addressing Write Transmit Mode	450
Figure 141. 10-bit Addressing Read Receive Mode	450
Figure 142. I ² C Bus Acknowledge	451
Figure 143. Clock Synchronization during Arbitration.....	452
Figure 144. Two Master Arbitration Procedure	452
Figure 145. Master Transmitter Timing Diagram	454
Figure 146. Master Receiver Timing Diagram	456
Figure 147. Slave Transmitter Timing Diagram	457
Figure 148. Slave Receiver Timing Diagram	458
Figure 149. SCL Timing Diagram.....	471
Figure 150. SPI Block Diagram	477
Figure 151. SPI Single Byte Transfer Timing Diagram – CPOL = 0, CPHA = 0.....	479
Figure 152. SPI Continuous Data Transfer Timing Diagram – CPOL = 0, CPHA = 0	480
Figure 153. SPI Single Byte Transfer Timing Diagram – CPOL = 0, CPHA = 1.....	480
Figure 154. SPI Continuous Transfer Timing Diagram – CPOL = 0, CPHA = 1.....	481
Figure 155. SPI Single Byte Transfer Timing Diagram – CPOL = 1, CPHA = 0.....	481
Figure 156. SPI Continuous Transfer Timing Diagram – CPOL = 1, CPHA = 0.....	482
Figure 157. SPI Single Byte Transfer Timing Diagram – CPOL = 1, CPHA = 1.....	482
Figure 158. SPI Continuous Transfer Timing Diagram – CPOL = 1, CPHA = 1.....	482
Figure 159. SPI Multi-Master Slave Environment.....	484
Figure 160. USART Block Diagram	498
Figure 161. USART Serial Data Format	500

Figure 162. USART Clock CK_USART and Data Frame Timing	501
Figure 163. Hardware Flow Control between 2 USARTs	502
Figure 164. USART RTS Flow Control	503
Figure 165. USART CTS Flow Control	503
Figure 166. IrDA Modulation and Demodulation	504
Figure 167. USART I/O and IrDA Block Diagram	506
Figure 168. RS485 Interface and Waveform	507
Figure 169. USART Synchronous Transmission Example	509
Figure 170. 8-bit Format USART Synchronous Waveform	510
Figure 171. UART Block Diagram	526
Figure 172. UART Serial Data Format	527
Figure 173. UART Clock CK_UART and Data Frame Timing	528
Figure 174. SCI Block Diagram	538
Figure 175. Character Frame and Compensation Mode	541
Figure 176. Guard Time Duration	542
Figure 177. Character and Block Waiting Time Duration – CWT and BWT	543
Figure 178. SCI Card Detection Diagram	544
Figure 179. SCI Interrupt Structure	546
Figure 180. USB Block Diagram	562
Figure 181. Endpoint Buffer Allocation Example	564
Figure 182. Double-buffering Operation Example	565
Figure 183. PDMA Block Diagram	593
Figure 184. PDMA Request Mapping Architecture	595
Figure 185. PDMA Channel Arbitration and Scheduling Example	596
Figure 186. EBI Block Diagram	611
Figure 187. EBI Non-multiplexed 8-bit Data, 8-bit Address Read Operation	612
Figure 188. EBI Non-multiplexed 8-bit Data, 8-bit Address Write Operation	612
Figure 189. EBI Non-multiplexed 16-bit Data, N-bit Address Read Operation	613
Figure 190. EBI Non-multiplexed 16-bit Data, N-bit Address Write Operation	613
Figure 191. An EBI Address Latch Setup Diagram	614
Figure 192. EBI Multiplexed 16-bit Data, 16-bit Address Read Operation	614
Figure 193. EBI Multiplexed 16-bit Data, 16-bit Address Write Operation	615
Figure 194. EBI Multiplexed 8-bit Data, 20-bit Address Read Operation	615
Figure 195. EBI Multiplexed 8-bit Data, 20-bit Address Write Operation	616
Figure 196. EBI Inserts an IDLE Cycle between Transactions in the Same Bank (NOIDLE = 0)	617
Figure 197. EBI De-asserts an IDLE Cycle between Transactions in the Same Bank (NOIDLE = 1) ..	617
Figure 198. EBI Bank Memory Map	619
Figure 199. I ² S Block Diagram	627
Figure 200. Simple I ² S Master/Slave Configuration	628
Figure 201. I ² S Clock Generator Diagram	629
Figure 202. I ² S-justified Stereo Mode Waveforms	631

Figure 203. I ² S-justified Stereo Mode Waveforms (32-bit Channel Enabled).....	631
Figure 204. Left-justified Stereo Mode Waveforms.....	632
Figure 205. Left-justified Stereo Mode Waveforms (32-bit Channel Enabled).....	632
Figure 206. Right-justified Stereo Mode Waveforms	633
Figure 207. Right-justified Stereo Mode Waveforms (32-bit Channel Enabled)	633
Figure 208. I ² S-justified Mono Mode Waveforms.....	634
Figure 209. I ² S-justified Mono Mode Waveforms (32-bit Channel Enabled).....	634
Figure 210. Left-justified Mono Mode Waveforms	635
Figure 211. Left-justified Mono Mode Waveforms (32-bit Channel Enabled)	635
Figure 212. Right-justified Mono Mode Waveforms.....	636
Figure 213. Right-justified Mono Mode Waveforms (32-bit Channel Enabled).....	636
Figure 214. I ² S-justified Repeat Mode Waveforms	637
Figure 215. I ² S-justified Repeat Mode Waveforms (32-bit Channel Enabled).....	637
Figure 216. FIFO Data Content Arrangement for Various Modes.....	638
Figure 217. CRC Block Diagram	649
Figure 218. CRC Data Bit and Byte Reversal Example.....	651

1 Introduction

Overview

This user manual provides detailed information including how to use the devices, 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 datasheet.

The devices are high performance and low power consumption 32-bit microcontrollers 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 including advanced debug support.

The devices operate at a frequency of up to 48 MHz with a Flash accelerator to obtain maximum efficiency. It provides up to 128 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 ADC, I²C, USART, UART, SPI, I²S, GPTM, MCTM, SCI, CRC-16/32, RTC, WDT, PDMA, EBI, USB2.0 FS, SW-DP (Serial Wire Debug Port), etc., are also implemented in the device series. Several power saving modes provide the flexibility for maximum optimization between wakeup latency and power consumption, an especially important consideration in low power applications.

The above features ensure that the devices are suitable for use in a wide range of applications, especially in areas such as white goods application control, power monitors, alarm systems, consumer products, handheld equipment, data logging applications, motor control and so on.

Features

- Core
 - 32-bit Arm® Cortex®-M0+ processor core
 - Up to 48 MHz operating frequency
 - 0.93 DMIPS/MHz (Dhrystone v2.1)
 - Single-cycle multiplication
 - Integrated Nested Vectored Interrupt Controller (NVIC)
 - 24-bit SysTick timer
- On-chip Memory
 - Up to 128 KB on-chip Flash memory for instruction/data and options storage
 - 16 KB on-chip SRAM
 - Supports multiple boot modes
- Flash Memory Controller – FMC
 - Flash accelerator for 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) and Programmable Low Voltage Detector (LVD)
- Clock Control Unit – CKCU
 - External 4 to 16 MHz crystal oscillator
 - External 32,768 Hz crystal oscillator
 - Internal 8MHz 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
 - Independent clock divider and gating bits for peripheral clock sources
- Power management – PWRCU
 - Single V_{DD} power supply: 2.0 V to 3.6 V
 - Integrated 1.5 V LDO regulator for CPU core, peripherals and memories power supply
 - V_{BAT} battery power supply for RTC and backup registers
 - Three power domains: V_{DD} , 1.5 V and Backup
 - 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 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

- Analog to Digital Converter – ADC
 - 12-bit SAR ADC engine
 - Up to 1 MSPS conversion rate
 - Up to 12 external analog input channels
- Analog Comparator – CMP
 - Two rail-to-rail comparator
 - Each comparator has configurable negative inputs used for flexible voltage selection
 - Dedicated I/O pin or internal voltage reference provided by 6-bit scaler.
 - Programmable hysteresis
 - Programming speed and consumption
 - Comparator output can be output to I/O or to timers or ADC trigger inputs
 - 6-bit scaler can be configurable to dedicated I/O for voltage reference.
 - Comparator has interrupt generation capability with wakeup from Sleep or Deep Sleep modes through the EXTI controller.
- IO ports – GPIO
 - Up to 51 GPIOs
 - Port A, B, C, D are mapped as 16 external interrupts – EXTI
 - Almost I/O pins are configurable output driving current.
- Motor Control Timer – MCTM
 - One 16-bit up, down, up/down auto-reload counter
 - Up to 4 independent channels
 - 16-bit programmable prescaler allowing dividing the counter clock frequency 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
 - Complementary Outputs with programmable dead-time insertion
 - Supports 3-phase motor control and hall sensor interface
 - Break input to force the timer's output signals into a reset or fixed condition
- PWM Generation and Capture Timer – GPTM
 - One 16-bit up, down, up/down auto-reload counter
 - Up to 4 independent channels for each timer
 - 16-bit programmable prescaler allowing dividing the counter clock frequency 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

- Single Channel Generation and Capture Timers – SCTM
 - One 16-bit up and auto-reload counter
 - One channel for each timer
 - 16-bit programmable prescaler allowing dividing the counter clock frequency by any factor between 1 and 65536
 - Input Capture function
 - Compare Match Output
 - PWM waveform generation with Edge-aligned
 - Single Pulse Mode Output
- Basic Function Timer – BFTM
 - One 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
 - 12-bit down counter with 3-bit prescaler
 - Reset event for the system
 - Programmable watchdog timer window function
 - Registers write protection function
- Real Time Clock – RTC
 - 32-bit up-counter with a programmable prescaler
 - Alarm function
 - Interrupt and Wake-up event
- Inter-integrated Circuit – I²C
 - Supports both master and slave modes with a frequency of up to 1 MHz
 - Provide 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 ($f_{\text{PCLK}}/2$) MHz for master mode and ($f_{\text{PCLK}}/3$) 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 up to ($f_{\text{PCLK}}/16$) MHz and synchronous operating rate up to ($f_{\text{PCLK}}/8$) MHz
 - Capability of full duplex communication
 - Fully programmable characteristics of serial communication including: word length, parity bit, stop bit and bit order
 - Error detection: Parity, overrun, and frame error
 - Support Auto hardware flow control mode – RTS, CTS
 - IrDA SIR encoder and decoder
 - RS485 mode with output enable control
 - FIFO Depth: 8×9 bits for both receiver and transmitter

- Universal Asynchronous Receiver Transmitter – UART
 - Asynchronous serial communication operating baud-rate up to ($f_{\text{CLK}}/16$) MHz
 - Capability of full duplex communication
 - Fully programmable characteristics of serial communication including: word length, parity bit, stop bit and bit order
 - Error detection: Parity, overrun, and frame error
- Smart Card Interface – SCI
 - Supports ISO 7816-3 Standard
 - Character mode
 - Single transmit buffer and single receive buffer
 - 11-bit ETU (elementary time unit) counter
 - 9-bit guard time counter
 - 24-bit general purpose waiting time counter
 - Parity generation and checking
 - Automatic character retry on parity error detection in transmission and reception modes
- Inter-IC Sound – I²S
 - Master or slave mode
 - Mono and stereo
 - I²S-justified, Left-justified, and Right-justified mode
 - 8/16/24/32-bit sample size with 32-bit channel extended
 - 8 × 32-bit TX & RX FIFO with PDMA supported
 - 8-bit Fractional Clock Divider with rate control
- Cyclic Redundancy Check – CRC
 - Support CRC16 polynomial: 0x8005, $X^{16}+X^{15}+X^2+1$
 - Support CCITT CRC16 polynomial: 0x1021, $X^{16}+X^{12}+X^5+1$
 - Support IEEE-802.3 CRC32 polynomial: 0x04C11DB7, $X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^3+X^2+X+1$
 - Support 1's complement, byte reverse & bit reverse operation on data and checksum
 - Support byte, half-word & 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
 - Support 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 Address increment, decrement or fixed mode
 - 4-level programmable channel priority
 - Auto reload mode
 - Supports trigger sources:
ADC, SPI, USART, UART, I²C, I²S, EBI, GPTM, MCTM, SCI and software request

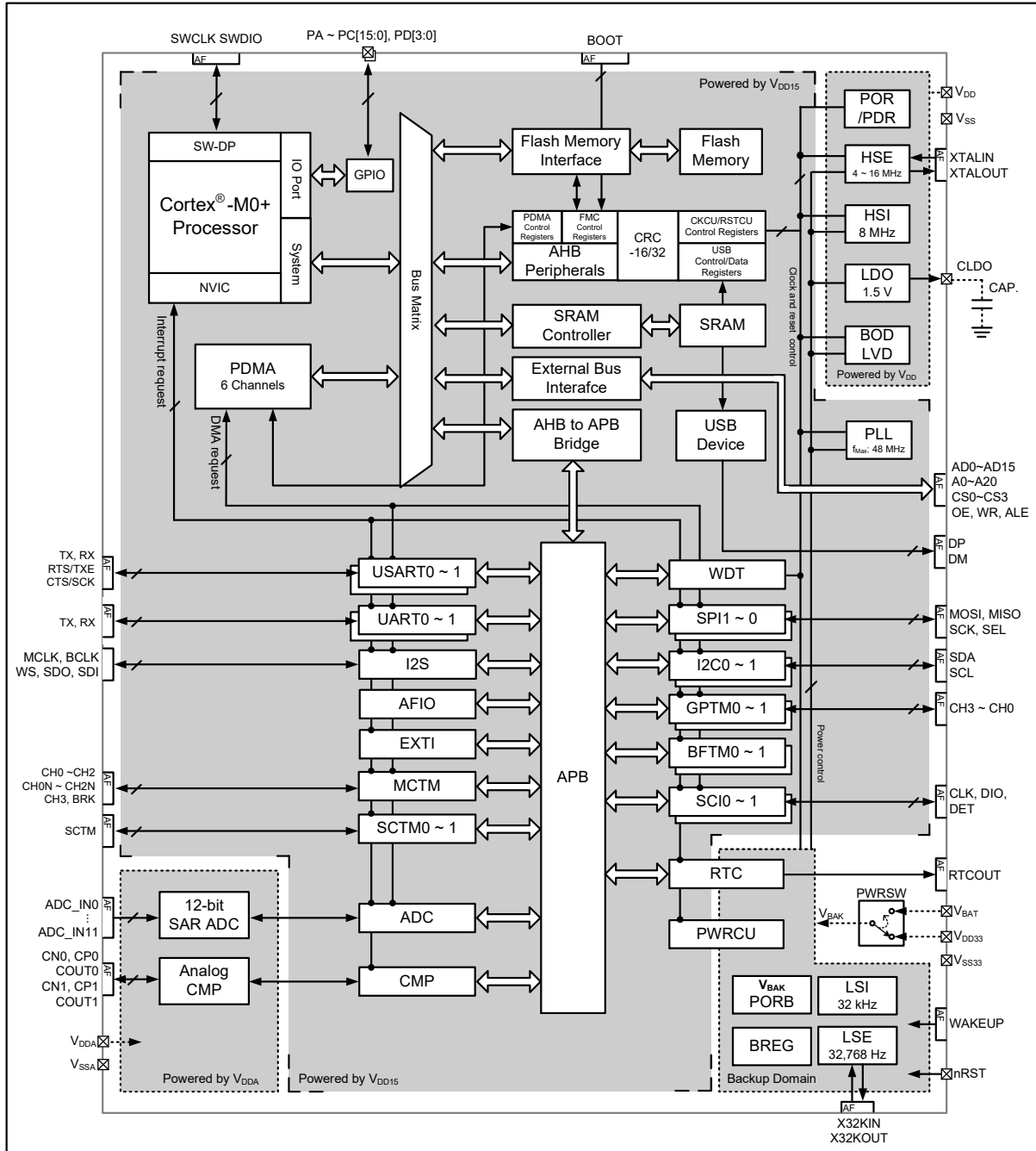
- External Bus Interface – EBI
 - Programmable interface for various memory types
 - Translate the AHB transactions into the appropriate external device protocol
 - Individual chip select signal for per memory bank
 - Programmable timing to support a wide range of devices
 - Automatic translation when AHB transaction width and external memory interface width is different
 - Write buffer to decrease the stalling of the AHB write burst transaction
 - Support multiplexed and non-multiplexed address and data line configurations
 - Multiplexed address and data line configurations
 - Up to 21 address lines
 - Up to 16-bit data bus width
- Universal Serial Bus Device Controller – USB
 - Complies with USB 2.0 full-speed (12Mbps) 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
 - 1KB EP-SRAM used as the endpoint data buffers
- 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
 - 33-pin QFN, 48/64-pin LQFP package
 - Operation temperature range: -40 °C to +85 °C

Device Information

Table 1. Series Features and Peripheral List

Peripherals		HT32F52342	HT32F52352
Main Flash (KB)		64	127.5
Option Bytes Flash (KB)		0.5	0.5
SRAM (KB)		8	16
Timers	MCTM	1	
	GPTM	2	
	SCTM	2	
	BFTM	2	
	RTC	1	
	WDT	1	
Communication	USB	1	
	SPI	2	
	USART	2	
	UART	2	
	I ² C	2	
	I ² S	1	
	SCI (ISO7816-3)	2	
PDMA		6 channels	
EBI		1	
CRC-16/32		1	
EXTI		16	
12-bit ADC		1	
Number of channels		12 Channels	
Comparator		2	
GPIO		Up to 51	
CPU frequency		Up to 48 MHz	
Operating voltage		2.0 V ~ 3.6 V	
Operating temperature		-40 °C ~ +85 °C	
Package		33-pin QFN, 48/64-pin LQFP	

Block Diagram



Power supply:
 Bus:
 Control signal:
 Alternate function:

Figure 1. Block Diagram

Introduction

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.						
0xn...n	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	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">19</td> <td style="text-align: center;">18</td> </tr> <tr> <td style="text-align: center;">SERDYIE</td> <td style="text-align: center;">PLLRDYIE</td> </tr> <tr> <td style="text-align: center;">RW 0</td> <td style="text-align: center;">RW 0</td> </tr> </table>	19	18	SERDYIE	PLLRDYIE	RW 0	RW 0	Software can read and write to this bit.
19	18							
SERDYIE	PLLRDYIE							
RW 0	RW 0							
RO	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">3</td> <td style="text-align: center;">2</td> </tr> <tr> <td style="text-align: center;">HSIRDY</td> <td style="text-align: center;">HSERDY</td> </tr> <tr> <td style="text-align: center;">RO 1</td> <td style="text-align: center;">RO 0</td> </tr> </table>	3	2	HSIRDY	HSERDY	RO 1	RO 0	Software can only read this bit. A write operation will have no effect.
3	2							
HSIRDY	HSERDY							
RO 1	RO 0							
RC	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">PDF</td> <td style="text-align: center;">BAK_PORF</td> </tr> <tr> <td style="text-align: center;">RC 0</td> <td style="text-align: center;">RC 1</td> </tr> </table>	1	0	PDF	BAK_PORF	RC 0	RC 1	Software can only read this bit. Read operation will clear it to 0 automatically.
1	0							
PDF	BAK_PORF							
RC 0	RC 1							
WC	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">3</td> <td style="text-align: center;">2</td> </tr> <tr> <td style="text-align: center;">SERDYF</td> <td style="text-align: center;">PLLRDYF</td> </tr> <tr> <td style="text-align: center;">WC 0</td> <td style="text-align: center;">WC 0</td> </tr> </table>	3	2	SERDYF	PLLRDYF	WC 0	WC 0	Software can read this bit or clear it by writing 1. Writing a 0 will have no effect.
3	2							
SERDYF	PLLRDYF							
WC 0	WC 0							
W0C	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">RXCF</td> <td style="text-align: center;">PARF</td> </tr> <tr> <td style="text-align: center;">RO 0</td> <td style="text-align: center;">W0C 0</td> </tr> </table>	1	0	RXCF	PARF	RO 0	W0C 0	Software can read this bit or clear it by writing 0. Writing a 1 will have no effect.
1	0							
RXCF	PARF							
RO 0	W0C 0							
WO	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">31</td> <td style="text-align: center;">30</td> </tr> <tr> <td style="text-align: center;">DB_CKSRC</td> <td style="text-align: center;">DB_CKSRC</td> </tr> <tr> <td style="text-align: center;">WO 0</td> <td style="text-align: center;">WO 0</td> </tr> </table>	31	30	DB_CKSRC	DB_CKSRC	WO 0	WO 0	Software can only write to this bit. A read operation always returns 0.
31	30							
DB_CKSRC	DB_CKSRC							
WO 0	WO 0							
Reserved	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">LLRDY</td> <td style="text-align: center;">Reserved</td> </tr> <tr> <td style="text-align: center;">RO 0</td> <td></td> </tr> </table>	1	0	LLRDY	Reserved	RO 0		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.
1	0							
LLRDY	Reserved							
RO 0								
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

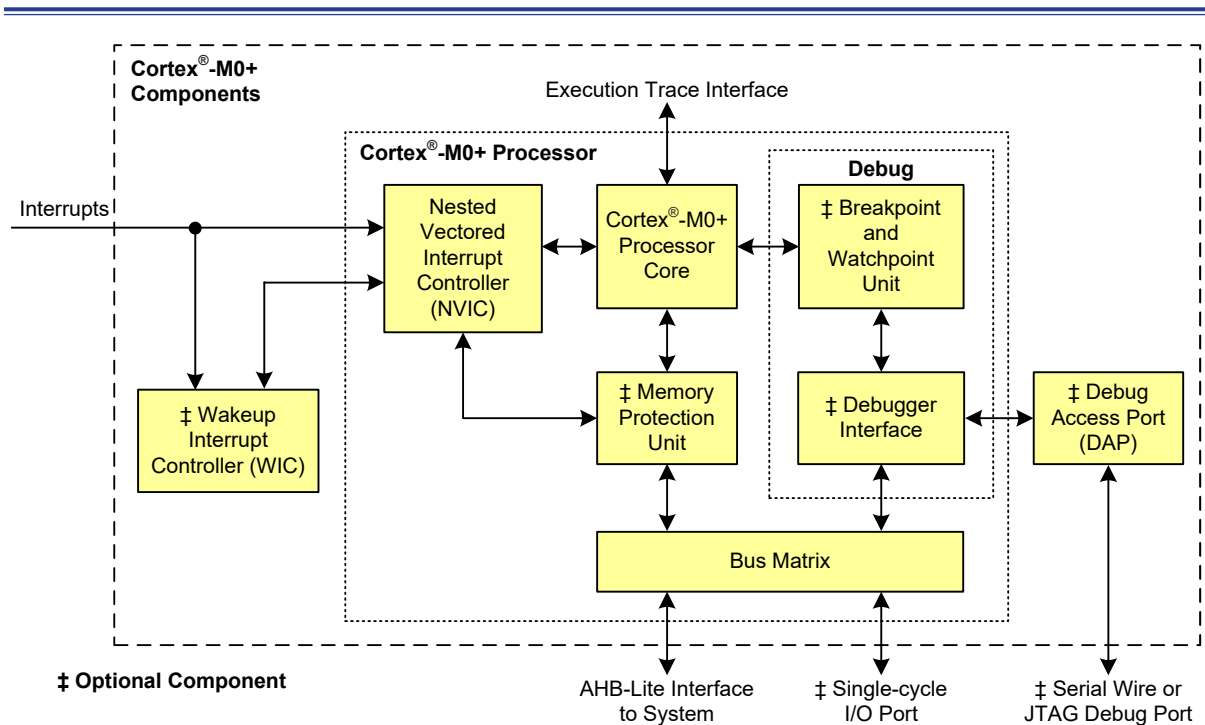
The system architecture of devices that includes the Arm® Cortex®-M0+ processor, bus architecture and memory organization will be described in the following sections. The Cortex®-M0+ is a next generation processor core which offers many new features. Integrated and advanced features make the Cortex®-M0+ processor suitable for market products that require microcontrollers with high performance and low power consumption. In brief, The Cortex®-M0+ processor includes 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 port; 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 port 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.

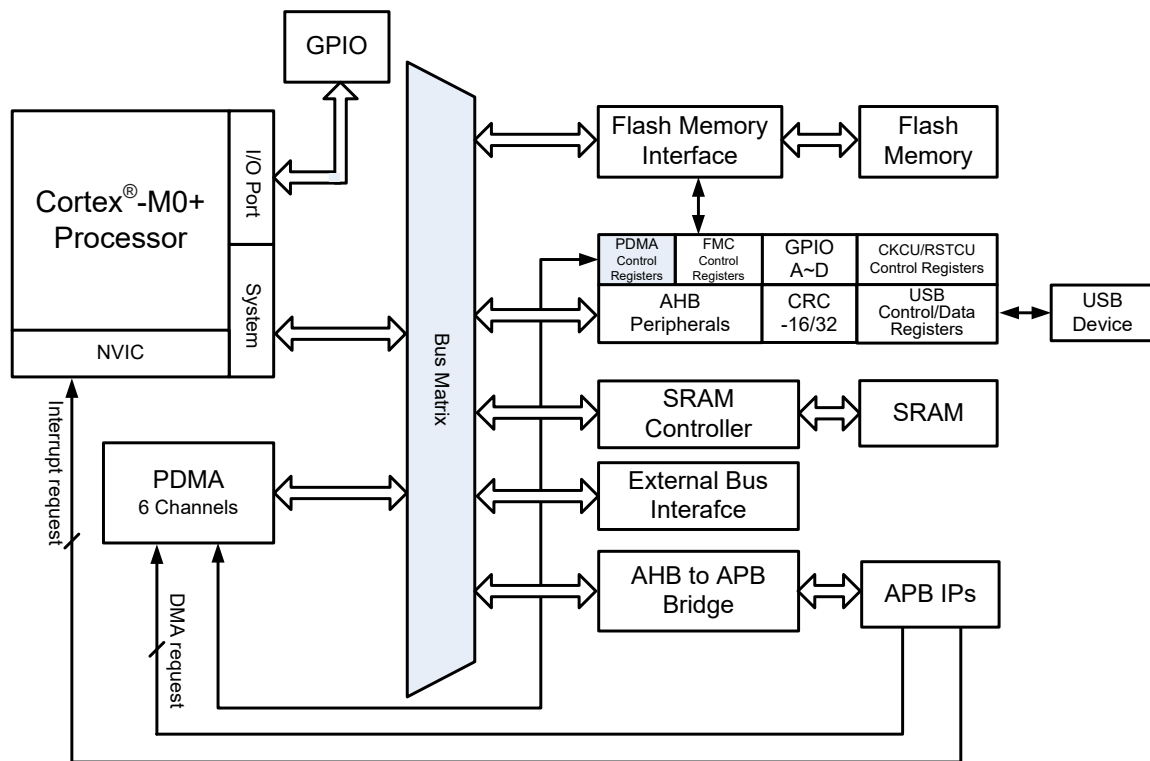


System Architecture

Figure 2. Cortex-M0+ Block Diagram

Bus Architecture

The HT32F52342/52352 series consist of two masters and five slaves in the bus architecture. The Cortex-M0+ AHB-Lite bus and Peripheral Direct Memory Access (PDMA) are the masters while the internal SRAM access bus, the internal Flash memory access bus, the AHB peripherals access bus, External Bus Interface (EBI) and the AHB to APB bridges are the slaves. The single 32-bit AHB-Lite system interface provides simple integration to all system regions include 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 HT32F52342/52352 series.



System Architecture

Figure 3. Bus Architecture

Memory Organization

The Arm® Cortex®-M0+ processor accesses and debug accesses share the single external interface to external AHB peripheral. 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 HT32F52342/52352 series of devices, including Code, SRAM, peripheral, and other pre-defined regions.

Memory Map

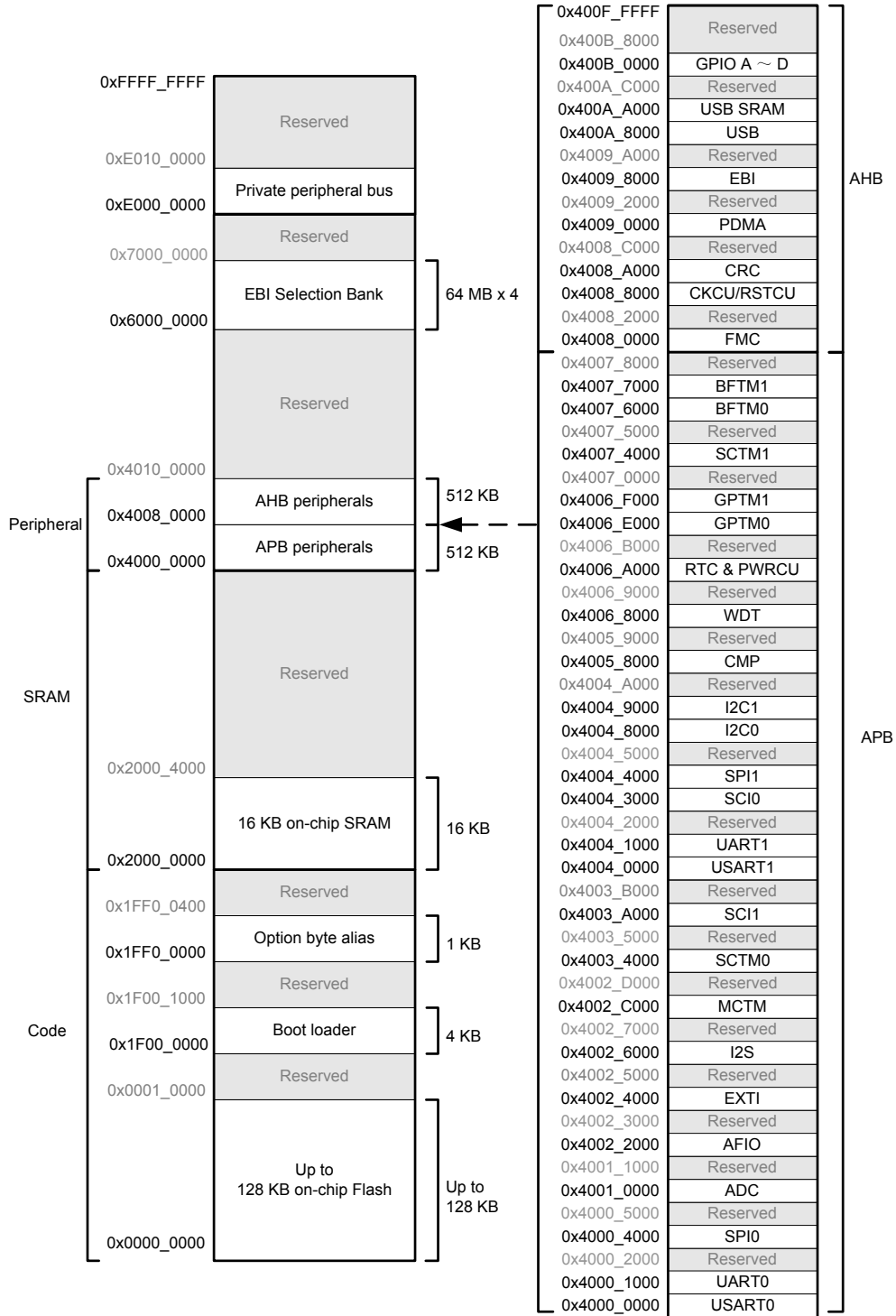


Figure 4. Memory Map

Table 3. Register Map

Start Address	End Address	Peripheral	Bus
0x4000_0000	0x4000_0FFF	USART0	APB
0x4000_1000	0x4000_1FFF	UART0	
0x4000_2000	0x4000_3FFF	Reserved	
0x4000_4000	0x4000_4FFF	SPI0	
0x4000_5000	0x4000_FFFF	Reserved	
0x4001_0000	0x4001_0FFF	ADC	
0x4001_1000	0x4002_1FFF	Reserved	
0x4002_2000	0x4002_2FFF	AFIO	
0x4002_3000	0x4002_3FFF	Reserved	
0x4002_4000	0x4002_4FFF	EXTI	
0x4002_5000	0x4002_5FFF	Reserved	
0x4002_6000	0x4002_6FFF	I ² S	
0x4002_7000	0x4002_BFFF	Reserved	
0x4002_C000	0x4002_CFFF	MCTM	
0x4002_D000	0x4003_3FFF	Reserved	
0x4003_4000	0x4003_4FFF	SCTM0	
0x4003_5000	0x4003_9FFF	Reserved	
0x4003_A000	0x4003_AFFF	SCI1	
0x4003_B000	0x4003_FFFF	Reserved	
0x4004_0000	0x4004_0FFF	USART1	
0x4004_1000	0x4004_1FFF	UART1	
0x4004_2000	0x4004_2FFF	Reserved	
0x4004_3000	0x4004_3FFF	SCI0	
0x4004_4000	0x4004_4FFF	SPI1	
0x4004_5000	0x4004_7FFF	Reserved	
0x4004_8000	0x4004_8FFF	I2C0	
0x4004_9000	0x4004_9FFF	I2C1	
0x4004_A000	0x4005_7FFF	Reserved	
0x4005_8000	0x4005_8FFF	Comparator	
0x4005_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	GPTM0	
0x4006_F000	0x4006_FFFF	GPTM1	
0x4007_0000	0x4007_3FFF	Reserved	
0x4007_4000	0x4007_4FFF	SCTM1	
0x4007_5000	0x4007_5FFF	Reserved	
0x4007_6000	0x4007_6FFF	BFTM0	
0x4007_7000	0x4007_7FFF	BFTM1	
0x4007_8000	0x4007_FFFF	Reserved	

Start Address	End Address	Peripheral	Bus
0x4008_0000	0x4008_1FFF	FMC	AHB
0x4008_2000	0x4008_7FFF	Reserved	
0x4008_8000	0x4008_9FFF	CKCU/RSTCU	
0x4008_A000	0x4008_BFFF	CRC	
0x4008_C000	0x4008_FFFF	Reserved	
0x4009_0000	0x4009_1FFF	PDMA Control Registers	
0x4009_2000	0x4009_7FFF	Reserved	
0x4009_8000	0x4009_9FFF	EBI Control Registers	
0x4009_A000	0x400A_7FFF	Reserved	
0x400A_8000	0x400A_BFFF	USB	
0x400A_C000	0x400A_FFFF	Reserved	
0x400B_0000	0x400B_1FFF	GPIOA	
0x400B_2000	0x400B_3FFF	GPIOB	
0x400B_4000	0x400B_5FFF	GPIOC	
0x400B_6000	0x400B_7FFF	GIOD	
0x400B_8000	0x400F_FFFF	Reserved	

Embedded Flash Memory

The HT32F52342/52352 series provide up to 128 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 up to capacity of 128 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

The HT32F52342/52352 series contain up to 16 KB on-chip SRAM which is located at address 0x2000_0000. It support 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 clock 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 functions of flash operation and pre-fetch buffer for the embedded on-chip Flash memory. 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 Flash memory is slower than the CPU, a wide access interface with pre-fetch buffer is provided to the Flash memory in order to reduce the wait state (which will cause instruction gaps) of the CPU. The functions of word program/page erase are also provided for instruction/data storage of Flash memory.

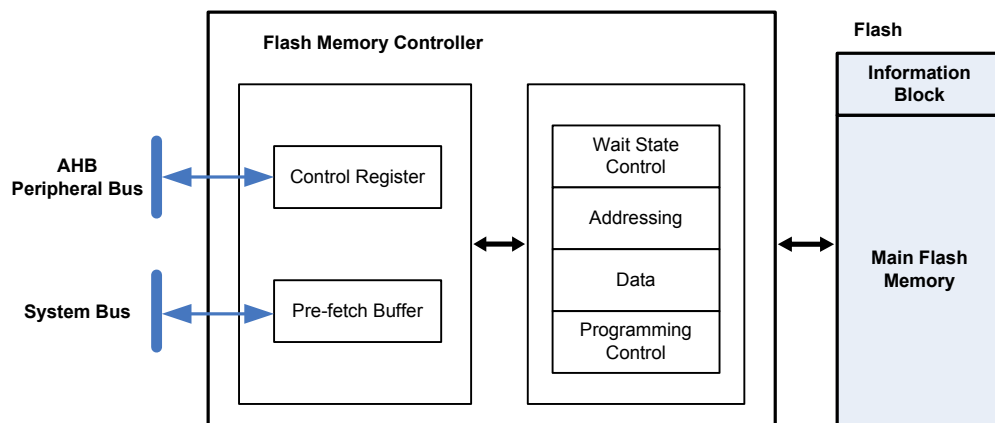


Figure 5. Flash Memory Controller Block Diagram

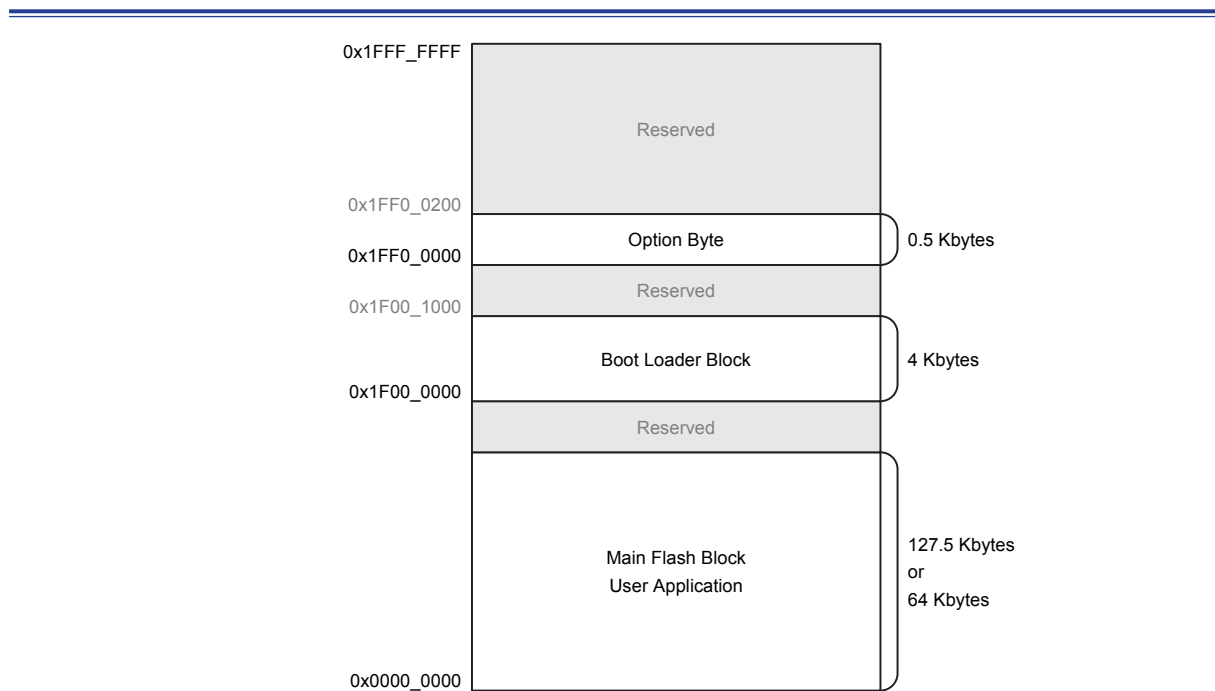
Features

- Up to 128 KB of on-chip Flash memory for storing instruction/data and options
 - 128 KB (instruction/data + Option Byte)
 - 64 KB (instruction/data + Option Byte)
- Page size of 512 Bytes, totally up to 256 pages depending on the main Flash size
- Wide access interface with pre-fetch buffer to reduce instruction gaps
- Page erase and mass erase capability
- 32-bit word programming
- Interrupt capability when ready or error occurs
- Flash read protection to prevent illegal code/data access
- Page erase/program protection to prevent unexpected operation

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_0FFF is mapped to Boot Loader Block (4 KB). Besides, address 0x1FF0_0000 to 0x1FF0_01FF is the alias of Option Byte block (0.5 KB) which locates at the last page of main Flash physically. The memory mapping on system view is shown as below.



Flash Memory Controller (FMC)

Figure 6. Flash Memory Map

Flash Memory Architecture

The Flash memory consists of up to 128 KB main Flash with 512 Bytes per page and 4 KB Information Block for Boot Loader. The main Flash memory contains totally 256 pages (or 128 pages for 64 KB device) 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_01FF	OB_PP [0]	512 Bytes
	Page 1	0x0000_0200 ~ 0x0000_03FF		512 Bytes
	Page 2	0x0000_0400 ~ 0x0000_05FF	OB_PP [1]	512 Bytes
	Page 3	0x0000_0600 ~ 0x0000_07FF		512 Bytes
	⋮	⋮	⋮	⋮
	Page 252	0x0001_F800 ~ 0x0001_F9FF	OB_PP [126]	512 Bytes
	Page 253	0x0001_FA00 ~ 0x0001_FBFF		512 Bytes
	Page 254	0x0001_FC00 ~ 0x0001_FDFF	OB_PP [127]	512 Bytes
	Page 255 (Option Byte)	Physical: 0x0001_FE00 ~ 0x0001_FFFF Alias: 0x1FF0_0000 ~ 0x1FF0_01FF	OB_CP [1]	512 Bytes
Information Block	Boot Loader	0x1F00_0000 ~ 0x1F00_0FFF	NA	4 KB

- Notes:**
1. Information Block stores boot loader, this block can not be programmed or erased by user.
 2. Option Byte is always located at last page of main Flash block.

Wait State Setting

When the HCLK clock is greater than access speed of Flash memory, the wait state cycles must be inserted during the CPU fetch instructions or load data from Flash memory. The wait state can be changed by setting WAIT [2:0] of Flash Cache and Pre-fetch Control Register (CFCR). In order to fit the requirement of wait state, the following two rules shall be considered.

- HCLK clock is changed from lower to higher: Change the wait state setting first and then change the HCLK clock.
- HCLK clock is changed from higher to lower: Change 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 HSI (8 MHz) is selected as HCLK clock source after system reset.

Table 5. Relationship Between Wait State Cycle and HCLK

Wait State Cycle	HCLK
0	0 MHz < HCLK <= 24 MHz
1	24 MHz < HCLK <= 48 MHz

Booting Configuration

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

Table 6. Booting Modes

Booting mode selection pin BOOT	Mode	Descriptions
0	Boot Loader	The source of Vector is Boot Loader
1	Main Flash	The source of Vector is main Flash

The Vector Mapping Control Register (VMCR) is provided to change the setting of the vector remapping temporarily after the chip reset. The reset value of VMCR is determined by the value of BOOT pin which will be sampled during the reset.

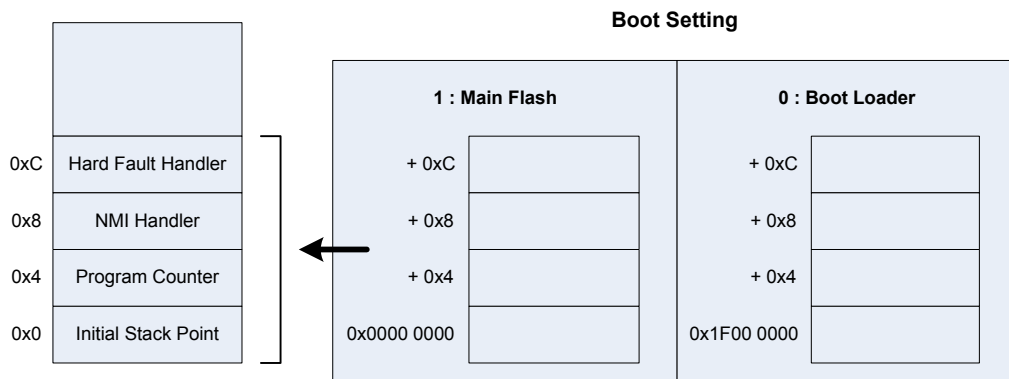


Figure 7. Vector Remapping

Page Erase

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

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

Note that a correct address of the target page must be confirmed. Software may run away if the target erase page is being used for fetching code or accessing data and FMC will not notify when this happens. Besides, the page erase will be ignored on the protected pages. A Flash Operation Error interrupt will be triggered by FMC if the OREIEN bit in the OIER register is set. Software can check the PPEF bit in the OISR register to detect this condition in the interrupt handler. The following figure displays the flow of page erase operation.

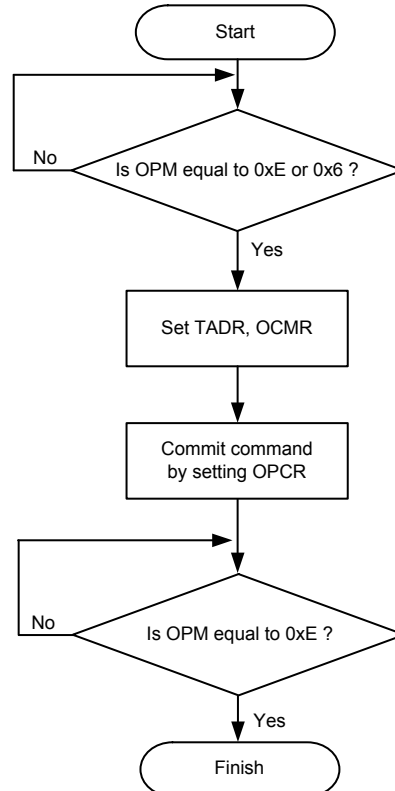


Figure 8. Page Erase Operation Flowchart

Mass Erase

The FMC provides a complete erase function which is used for resetting all the main Flash memory content. The following steps show the sequence of the register access for mass erase.

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

Since all Flash data will be reset as 0xFFFF_FFFF, the mass erase operation can be done by the program that runs in the SRAM or by the debugging tool that access FMC register directly. The software function that is executed on the Flash memory shall not trigger a mass erase operation. The following figure displays the flow of mass erase operation.

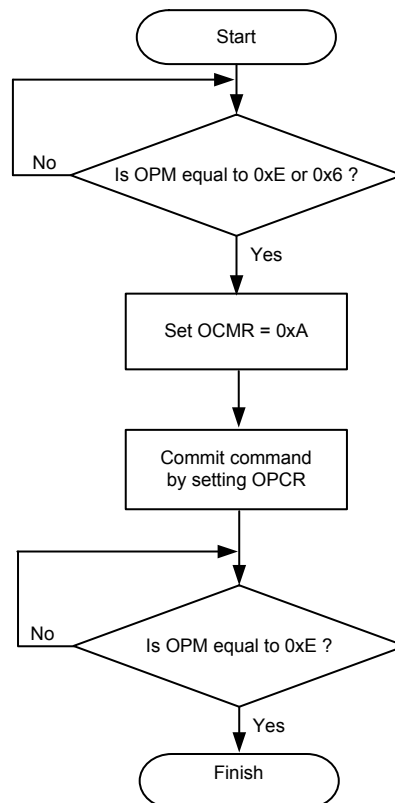


Figure 9. Mass Erase Operation Flowchart

Word Programming

The FMC provides a 32 bits word programming function which is used for modifying the Flash memory content. The following steps show the sequence of register access for word programming.

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

Note that the word programming operation can not be applied to the same address twice. Successive word programming operation to the same address must be separated by a page erase operation. Besides, the word program will be ignored on protected pages. A Flash operation error interrupt will be triggered by FMC if the OREIEN bit in the OIER register is set. Software can check the PPEF bit in the OISR register to detect this condition in the interrupt handler. The following figure displays the flow of word programming operation.

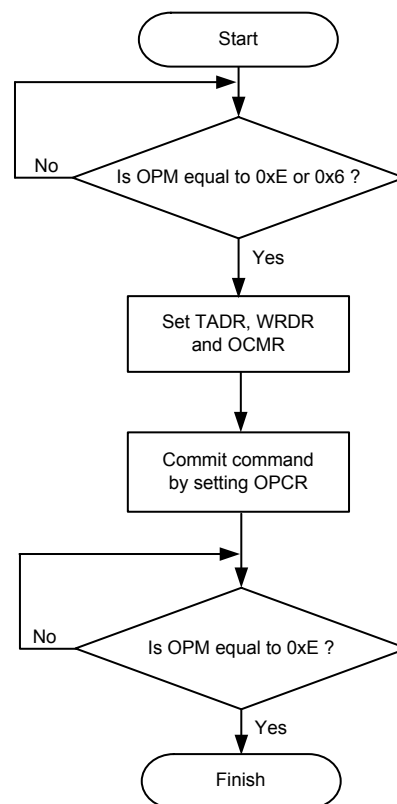


Figure 10. Word Programming Operation Flowchart

Option Byte Description

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

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 254 ~ page 0) OB_PP [n] (n = 0 ~ 127) 0: Flash Page 2n and 2n+1 Erase/Program Protection is enabled 1: Flash Page 2n and 2n+1 Erase/Program Protection is disabled	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 content value sum of 5 registers which offset address is form 0x000 to 0x010 in Option Byte (0x000 + 0x004 + 0x008 + 0x00C + 0x010) when the OB_PP or OB_CP register's content is not equal to 0xFFFF_FFFF. Otherwise, both page erase/program protection and security protection will be enabled.	0xFFFF_FFFF

Page Erase/Program Protection

FMC provides functions of page erase/program protection to prevent unexpected operation of Flash memory. The page erase (CMD [3:0] = 0x8 in the OCMR register) or word program (CMD [3:0] = 0x4) command will not be accepted by FMC on the protected pages, the PPEF bit in the OISR register will then be set by FMC and the Flash operation error interrupt will be triggered to CPU by FMC if the OREIEN bit in the OIER register is set. The page protection function can be enabled for each page independently by setting the OB_PP registers of the Option Byte. 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

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

- Notes:**
- Note that the setting of write protection is based on page. The above access permission only affects the pages that enable protection function. Other pages are not affected.
 - Main Flash page protection is configured by OB_PP [126:0]. Option Byte is physically located at the last page of main Flash. Option Byte page protection is configured by the OB_CP [1] bit.
 - The page erase on Option Byte area can disable the page protection of main Flash.
 - The page protection of Option Byte can only be disabled by a mass erase operation.

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

- Check the OPCR register to confirm that no Flash memory operation is in progress (OPM [3:0] equals to 0xE or 0x6). Otherwise, wait until the previous operation has been finished.
- Write OB_PP address to TADR register (TADR = 0x1FF0_0000).
- Write WRDR register which indicates the protection function of corresponding page is enabled or disabled (0: Enabled, 1: Disabled).
- Write word program command to OCMR register (CMD [3:0] = 0x4).
- Commit word program command to FMC by setting OPCR register (set OPM [3:0] = 0xA).
- Wait until all operations have been finished by checking the value of OPCR register (OPM [3:0] equals to 0xE).
- Read and verify the Option Byte if required.
- Program the OB_CK Option Byte as sum of 5 words 0x000 ~ 0x010 according to the checksum rule of Option Byte.
- Apply a system reset to active the new OB_PP setting.

Security Protection

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

Table 9. Access Permission When Security Protection is Enabled

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

Notes: 1. User application means the software that is executed or booted from main Flash memory with the JTAG/SW debugger being disconnected. But Option Byte block and page 0 are still in protection and can not Program/Page Erase.

2. Mass erase operation can erase Option Byte block and disable security protection.

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

- Check OPCR register to confirm that no Flash memory operation is ongoing (OPM [3:0] equal to 0xE or 0x6). Otherwise, wait until the previous operation has been finished.
- Write OB_CP address to TADR register (TADR = 0x1FF0_0010).
- Write WRDR register, set OB_CP [0] as 0.
- Write word program command to OCMR register (CMD [3:0] = 0x4).
- Commit word program command to FMC by setting OPCR register (set OPM = 0xA).
- Wait until all operations have been finished by checking the value of OPCR register (OPM [3:0] equals to 0xE).
- Read and verify the Option Byte if required.
- Program the OB_CK Option Byte as sum of 5 words 0x000 ~ 0x010 according to the checksum rule of Option Byte.
- 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
FMC Base Address = 0x4008_0000			
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	Flash Page Erase/Program Protection Status Register	0xXXXX_XXXX
	0x024		0xXXXX_XXXX
	0x028		0xXXXX_XXXX
	0x02C		0xXXXX_XXXX
CPSR	0x030	Flash Security Protection Status Register	0xXXXX_XXXX
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_00X0
PSSR	0x188	Flash Page Size Status Register	0x0000_0200
CFCR	0x200	Flash Cache & Pre-fetch Control Register	0x0000_0051
CIDR0	0x310	Custom ID Register 0	0xXXXX_XXXX
CIDR1	0x314	Custom ID Register 1	0xXXXX_XXXX
CIDR2	0x318	Custom ID Register 2	0xXXXX_XXXX
CIDR3	0x31C	Custom ID Register 3	0xXXXX_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 page erase and word programming operation.

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. Since the programming length is 32 bits, the TADR shall be set as word-aligned (4 bytes). The TADB [1:0] will be ignored during programming operations. For page erase operations, the TADR register contains the page address which is going to be erased. Since the page size is 512 Byte, the TADB [8:0] will be ignored in order to limit the target address as 512 Byte-aligned. For 128 KB main Flash addressing, TADB [31:17] should be zero and TADB [31:16] should be zero for 64 KB. Address from 0x1FF0_0000 to 0x1FF0_03FF is the 1KB Option Byte. This field for available Flash address, it must be under 0x1FFF_FFFF. Otherwise, the Invalid Target Address interrupt will be occurred if the corresponding interrupt enable bit is set.</p>

Flash Write Data Register – WRDR

This register specifies the data to be written for 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

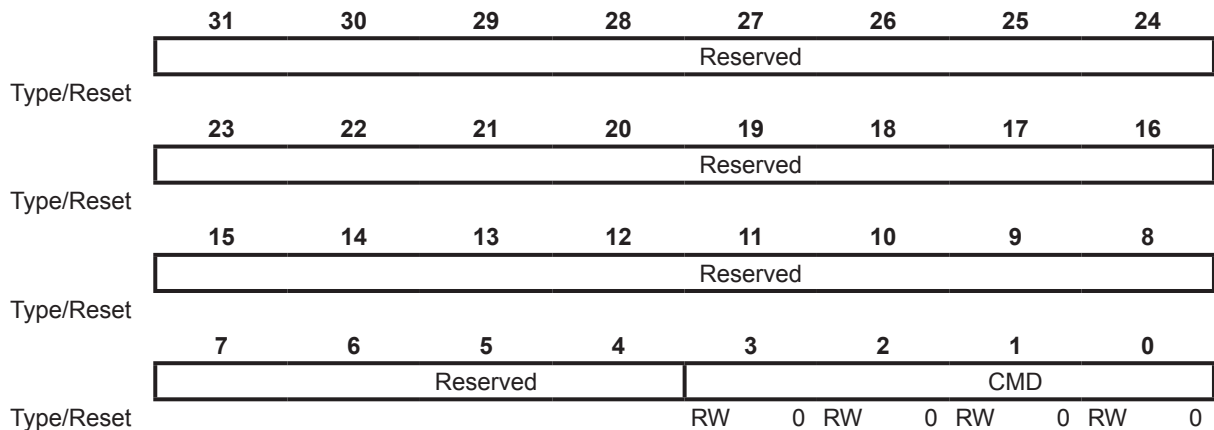
Bits	Field	Descriptions
[31:0]	WRDB	Flash Write Data Bits The data value for programming operation.

Flash Operation Command Register – OCMR

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

Offset: 0x00C

Reset value: 0x0000_0000



Bits	Field	Descriptions
[3:0]	CMD	Flash Operation Command The following table shows definitions of CMD [3:0] which specify the Flash operation. If an invalid command is set and IOCMIEN = 1, the Invalid Operation Command interrupt will be occurred.

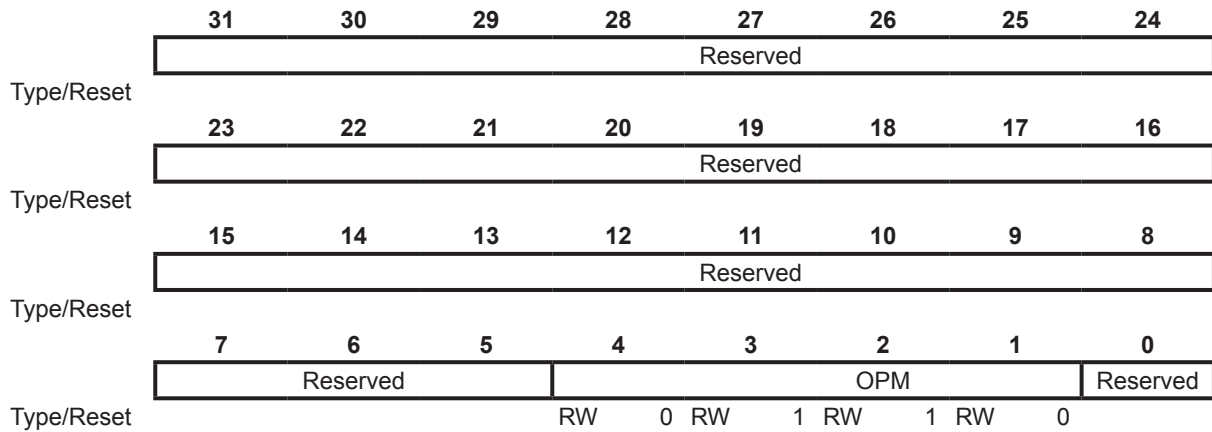
CMD [3:0]	Description
0x0	Idle (default)
0x4	Word program
0x8	Page erase
0xA	Mass erase
Others	Reserved

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



Bits	Field	Descriptions
[4:1]	OPM	<p>Operation Mode</p> <p>The following table shows the operation mode of FMC. User can commit command which is set by the OCMR register to main flash according to the address alias setting in TADR. The content of TADR, WRDR, and OCMR registers shall be prepared before setting this register. After all the operation has been finished, the OPM field will be set as 0xE or 0xF by the FMC hardware. The Idle mode can be set when all the operations have been finished for power saving. Note that the operation status should be checked before the next action is applied to the FMC. The content of TADR, WRDR, OCMR, and OPCR registers should not be changed until the previous operation has been finished.</p>

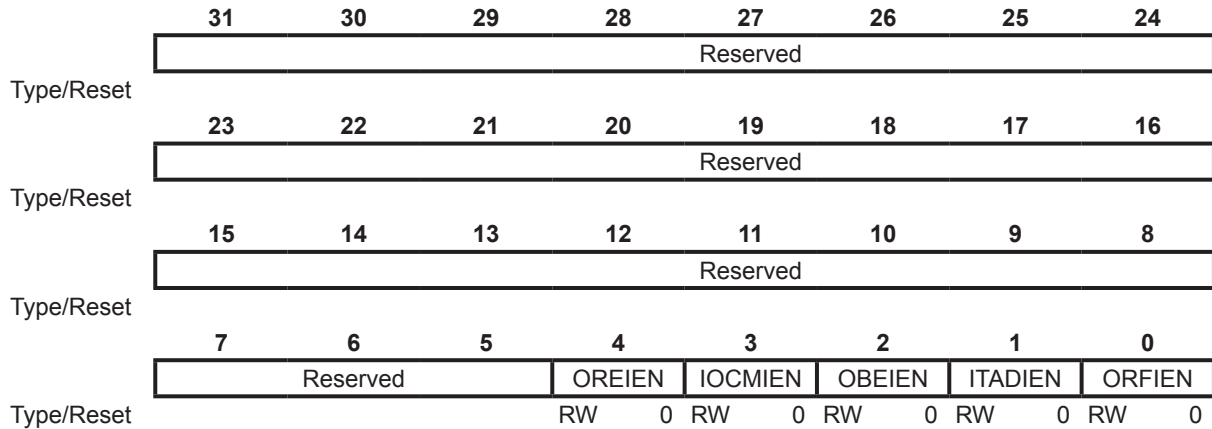
OPM [3:0]	Description
0x6	Idle (default)
0xA	Commit command to main Flash
0xE	All operation finished on main Flash
Others	Reserved

Flash Operation Interrupt Enable Register – OIER

This register is used to enable or disable interrupt function of FMC. The FMC generates interrupt to the controller when corresponding interrupt enable bits are set.

Offset: 0x014

Reset value: 0x0000_0000



Bits	Field	Descriptions
[4]	OREIEN	Operation Error Interrupt Enable 0: Operation error does not generate an interrupt 1: Operation error generates an interrupt
[3]	IOCM IEN	Invalid Operation Command Interrupt Enable 0: Invalid Operation Command does not generate an interrupt 1: Invalid Operation Command generates an interrupt
[2]	OBEIEN	Option Byte Check Sum Error Interrupt Enable 0: Option Byte Check Sum Error does not generate an interrupt 1: Option Byte Check Sum Error generates an interrupt
[1]	ITADIEN	Invalid Target Address Interrupt Enable 0: Invalid Target Address does not generate an interrupt 1: Invalid Target Address generates an interrupt
[0]	ORFIEN	Operation Finished Interrupt Enable 0: Operation Finish does not generate an interrupt 1: Operation Finish generates an interrupt

Flash Operation Interrupt and Status Register – OISR

This register indicates the status of the FMC interrupt to check if an operation has been finished or an error occurs. The status bits (bit [4:0]) are available when the corresponding bits in the OIER register are set.

Offset: 0x018

Reset value: 0x0001_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved						PPEF	RORFF	
	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

Bits	Field	Descriptions
[17]	PPEF	Page Erase/Program Protected Error Flag 0: Page Erase/Program Protected Error is not occurred 1: Operation error since an invalid erase/program operation is applied to a protected page This bit is reset by hardware once a new flash operation command is committed.
[16]	RORFF	Raw Operation Finished Flag 0: The last flash operation command is not finished 1: The last flash operation command is finished RORFF is directly connected from Flash memory for debugging purpose.
[4]	OREF	Operation Error Flag 0: No flash operation error occurred 1: The last flash operation is failed This bit will be set when any error of flash operation such as invalid command, program error and erase error, etc. is occurred. The ORE interrupt occurs if the OREIEN bit in the OIER register is set. Reset this bit by writing 1.
[3]	IOCMF	Invalid Operation Command Flag 0: No invalid flash operation command was set 1: An invalid flash operation command is set into the OCMR register. The IOCM interrupt will be occurred if the IOCMIEN bit in the OIER register is set. Reset this bit by writing 1.
[2]	OBEF	Option Byte Check Sum Error Flag 0: Check sum of Option Byte is correct 1: Check sum of Option Byte is incorrect The OBE interrupt will be occurred if the OBEIEN bit in the OIER register is set. But the Option Byte Check Sum Error Flag has to wait the interrupt condition is cleared then reset this bit by software writes 1, that means the Option Byte check sum value has to modified to correct. Otherwise, the interrupt will be continually kept or the software disables the interrupt enable bit to release the interrupt request.

Bits	Field	Descriptions
[1]	ITADF	Invalid Target Address Flag 0: The target address is valid 1: The target address TADR is invalid TADR field must be below 0x1FFF_FFFF. The ITAD interrupt will be occurred if the ITADIEN bit in the OIER register is set. Reset this bit by writing 1.
[0]	ORFF	Operation Finished Flag 0: No operation finished interrupt occurred 1: Last flash operation command is finished The ORF interrupt will be occurred if the ORFIEN bit in the OIER register is set. Reset this bit by writing 1.

Flash Page Erase/Program Protection Status Register – PPSR

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

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

Reset value: 0xXXXX_XXXX

	31	30	29	28	27	26	25	24	
	PPSBn								
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	
	PPSBn								
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	
	PPSBn								
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	
	PPSBn								
Type/Reset	RO	X RO	X RO	X RO	X RO	X RO	X RO	X RO	X

Bits	Field	Descriptions
[127:0]	PPSBn	<p>Page Erase/Program Protection Status Bits (n = 0 ~ 127) PPSB[n] = OB_PP[n]</p> <p>0: The corresponding pages are protected 1: The corresponding pages are 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 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]. 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 and dependent on the chip specification. The other remained bits of OB_PP and PPSR registers are reserved.</p>

Flash Security Protection Status Register – CPSR

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

Offset: 0x030

Reset value: 0xFFFF_XXXX

	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 OPBSB is determined by the Option Byte, OB_CP [1].
[0]	CPSB	Flash Security Protection Status Bit 0: Flash Security protection is enabled 1: Flash Security protection is not enabled The reset value of CPSB is determined by the Option Byte, OB_CP [0].

Flash Vector Mapping Control Register – VMCR

This register is used to control the mapping of vector. The reset value of VMCR is determined by booting power on the reset setting BOOT pin.

Offset: 0x100

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						RW	X	1

Bits	Field	Descriptions
------	-------	--------------

[1] VMCB Vector Mapping Control Bit
The VMCB bits is used to control the mapping source of first 4-word vector (address 0x0 ~ 0xC). The following table shows the vector mapping setting.

BOOT	VMCB [1]	Descriptions
Low	0	Boot Loader mode The source of the vector mapping is the boot loader area.
High	1	Main Flash mode The source of the vector mapping is the main Flash area.

The reset value of VMCB is determined by the pins status of BOOT during power on reset and system reset. The setting of the vector mapping can be changed temporarily by setting the VMCB bit when the application is running.

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	0	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 Read the last 4 digital codes of the MCU device part number.

Flash Page Number Status Register – PNSR

This register specifies the page number of Flash memory.

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 specifies the page size in bytes.

Offset: 0x188

Reset value: 0x0000_0200

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

Bits	Field	Descriptions
[31:0]	PSSB	Status Bits of Flash Page Size 0x200: That means the page size is 512 Byte per page. 0x400: That means the page size is 1 KB per page. 0x800: That means the page size is 2 KB per page.

Flash Cache & Pre-fetch Control Register – CFCR

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

Offset: 0x200

Reset value: 0x0000_13D1

	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		CE		Reserved			
	7	6	5	4	3	2	1	0
Type/Reset	Reserved		PFBE		Reserved	WAIT		
	1	1	0	RW	1	RW	0	RW
							0	RW
								1

Bits	Field	Descriptions
[12]	CE	Branch Cache Enable Bit 0 = Cache is disabled. 1 = Cache is enabled (default).
[4]	PFBE	Pre-fetch Buffer Enable Bit 0: Pre-fetch buffer is disabled. The Instruction/Data is provided by Flash memory directly. 1: Pre-fetch buffer is enabled (default).
[2:0]	WAIT	Flash Wait State Setting The WAIT [2:0] is used to set the count of the HCLK wait clock during non-sequential address Flash access. The actual wait clock is (WAIT [2:0] - 1). Since the wide access interface with 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 ≤ 24 MHz
010	1	24 MHz < HCLK ≤ 48 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: 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 three digital power domains.

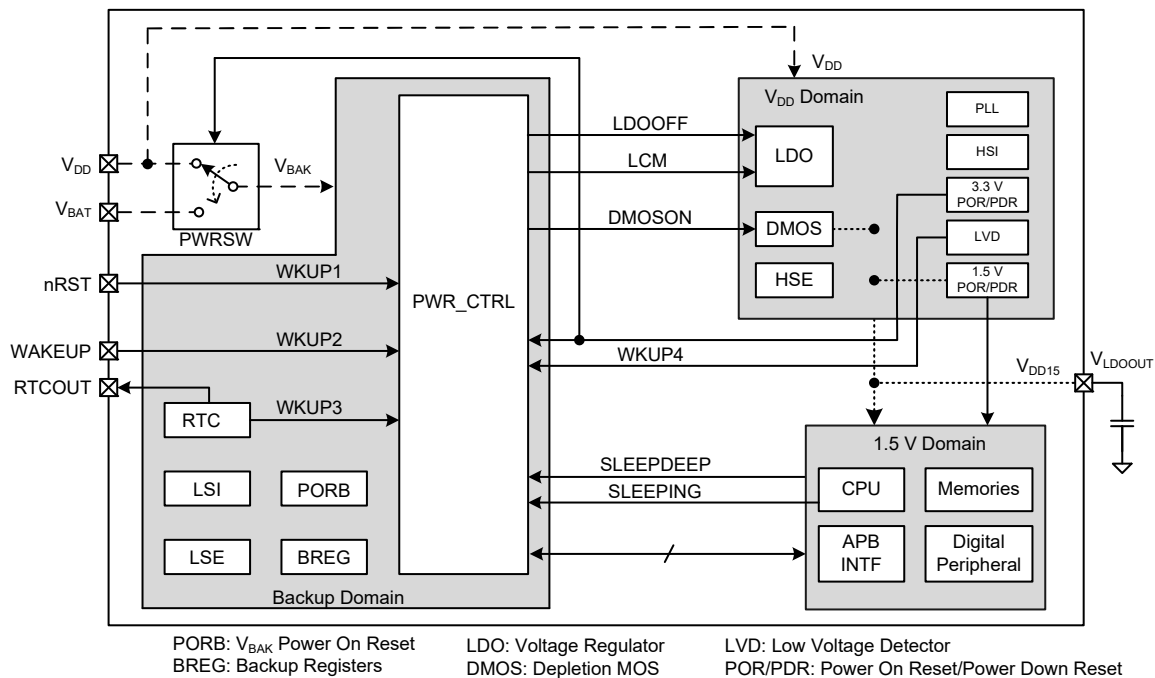


Figure 11. PWRCU Block Diagram

Features

- Three power domains: Backup, V_{DD} and 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 $V_{DD} \leq V_{BOD}$.
- BOD 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} .
- LVD Low Voltage Detector can issue an interrupt or wakeup event when V_{DD} is lower than a programmable threshold voltage V_{LVD} .
- Switch Battery power (V_{BAT}) for backup domain when V_{DD} is lower than V_{PDR} voltage.
- 40 bytes of backup registers powered by V_{BAK} for data storage of user application data when in the Power-Down mode.

Functional Descriptions

Backup Domain

Power Switch

The Backup Domain is powered by the V_{DD} power source or the battery power source, V_{BAT} , which is selected by the power switch PWRSW. The operating voltage range of the Back Domain is from 2.0 V to 3.6 V. If V_{DD} is lower than V_{PDR} , then the power source of the Back Domain will be automatically switched from V_{DD} to V_{BAT} . Therefore, even if V_{DD} is powered down, all the circuitry in the backup domain can operate normally. This means that the backup register contents will be retained, the RTC circuitry will operate normally and the low speed oscillators can keep running.

Backup Domain Reset

The Backup Domain reset sources include the Backup Domain Power-On-Reset (PORB) and the Backup Domain software reset which is activated by setting the BAKRST bit in the BAKCR register. The PORB signal forces the device to stay in the reset mode until the V_{BAK} is greater than V_{PORB} . The application software can set the PORBDN bit in the BAKCR register to disable PORB circuit to save the current consumption in the Backup Domain. Also the application software can trigger Backup Domain software reset by setting the BAKRST bit in the BAKCR register. All registers of PWRCU and RTC will be reset only by the Backup Domain reset.

LSE, LSI and RTC

The Real Time Clock circuitry 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 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 wakeup the device when the compare match event occurs. The details of the RTC configuration for wakeup timer will be described in the RTC chapter.

Backup Registers and Isolation Cells

Ten 32-bit registers, up to 40 bytes, are located in the Backup Domain for user application data storage. These registers are powered by V_{BAK} which constantly supplies power when the 1.5 V core power is switched off. The Backup Registers are only reset by the Backup Domain power-on-reset, PORB, or the Backup Domain software reset, BAKRST. When the device resumes operation from the 1.5 V power, either by Hardware or Software, access to the Backup registers and the RTC registers are disabled by the isolation cells which protect these registers against possible parasitic write accesses. To resume access operations, users must disable these isolation cells by setting the BKISO bit to 1 in the LPCR register of the Clock Control Unit.

LDO Power Control

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

- The Power-Down or Deep-Sleep 2 mode is entered.
- The control bits $BODEN = 1$, $BODRIS=0$ and the supply power $V_{DD} \leq V_{BOD}$.
- The supply power $V_{DD33} \leq V_{PDR}$

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 a low current mode, LCM. To enter the Deep-Sleep 2 mode, the PWRCU will turn off the LDO and turn on the DMOS to supply an alternative 1.5 V power.

V_{DD} Power Domain

Voltage Regulator

The voltage regulator, LDO, Depletion MOS, DMOS, Low voltage Detector, LVD and High Speed Internal oscillator, HSI are operated under the V_{DD} power domain. The LDO can be configured to operate in either normal mode ($LDOOFF = 0$, $SLEEPDEEP = 0$, $I_{OUT} =$ High current mode) or low current mode ($LDOOFF = 0$, $SLEEPDEEP=1$, $I_{OUT} =$ 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 BAKCR register. The DMOS output has weak output current and regulation capability and only operate in the Deep-Sleep 2 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 2.0 V. The device remains in Power-Down mode when V_{DD} is below a specified threshold V_{PDR} , without the need for an external reset circuit. For more details the power on / power down reset threshold voltage, refer to the electrical characteristics of the corresponding datasheet.

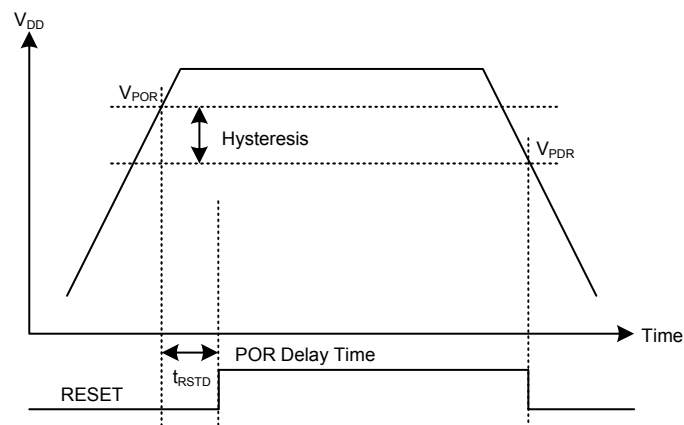


Figure 12. Power On Reset / Power Down Reset Waveform

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 bits in the LVDCSR register. When a low voltage on the V_{DD} power pin is detected, the LVDF flag will be active and an interrupt will be generated and sent to the MCU 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 will 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 wake up 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.

1.5 V Power Domain

The main functions that include the APB interface for the backup 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 (Refer to PORB) on 1.5 V power domain. Subsequently, to enter the expected power saving mode, the associated control bits including the LDOOFF, DMOSON, and SLEEPDEEP bits must be configured. Then, once a WFI or WFE instruction is executed, the device will enter an 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 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	1. CPU clock will be stopped. 2. Peripherals, Flash and SRAM clocks can be stopped by setting.
Deep-Sleep1 ~ 2	1. Stop all clocks in the 1.5 V power domain. 2. Disable HSI, HSE, and PLL. 3. Turning 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 0 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 clear the SLEEPDEEP bit to 0 and execute a WFI or WFE instruction. 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 CMP Wakeup or LVD wakeup ⁽²⁾ or WAKEUP pin rising edge or USB resume
Deep-Sleep2		1	X	1	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 in event mode, RTC, LVD, and 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 waked up by a LVD event and then the LDO regulator can be turned on when system is woken up from the Deep-Sleep2 and Power-Down modes.

Deep-Sleep Mode

To enter 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 PLL and high speed oscillator, known as HSI and HSE, will be stopped. In addition, Deep-Sleep1 turns the LDO into low current mode while Deep-Sleep2 turns off the LDO and uses a DMOS to keep 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 oscillator will be enabled. Finally, the CPU will return to 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 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. A RTC wakeup trigger event, a 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}) and the Backup Domain power (V_{BAK}).

After a system reset, the PORSTF bit in the RSTCU GRSR register, the PDF and BAKPORF bits in the BAKSR register should be checked by software to confirm if the device is being resumed from the Power-Down mode by a backup domain power on reset, an unexpected loss of the 1.5 V power or other reset events (nRST, WDT,...). If the device has entered the Power-Down mode under the correct firmware procedure, then the PDF bit will be set. The System information could be saved in the Backup Registers and be retrieved when the 1.5 V power domain is powered on again. More information about the PDF and BAKPORF bits in the BAKSR register and PORSTF bit in the RSTCU GRSR register is shown in the following table.

Table 13. Power Status After System Reset

BAKPORF	PDF	PORSTF	Description
1	0	1	Power-up for the first time after the backup domain is reset: Power on reset when V_{BAK} is applied for the first time or executing software reset command on the backup 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_{BAK} backup power domain.

Table 14. PWRCU Register Map

Register	Offset	Description	Reset Value
BAKSR	0x100	Backup Domain Status Register	0x0000_0001
BAKCR	0x104	Backup Domain Control Register	0x0000_0000
BAKTEST	0x108	Backup Domain Test Register	0x0000_0027
LVDCSR	0x110	Low Voltage/Brown Out Detect Control and Status Register	0x0000_0000
BAKREG0	0x200	Backup Register 0	0x0000_0000
BAKREG1	0x204	Backup Register 1	0x0000_0000
BAKREG2	0x208	Backup Register 2	0x0000_0000
BAKREG3	0x20C	Backup Register 3	0x0000_0000
BAKREG4	0x210	Backup Register 4	0x0000_0000
BAKREG5	0x214	Backup Register 5	0x0000_0000
BAKREG6	0x218	Backup Register 6	0x0000_0000
BAKREG7	0x21C	Backup Register 7	0x0000_0000
BAKREG8	0x220	Backup Register 8	0x0000_0000
BAKREG9	0x224	Backup Register 9	0x0000_0000

Register Descriptions

Backup Domain Status Register – BAKSR

This register indicates backup domain status.

Offset: 0x100

Reset value: 0x0000_0001 (Reset only by Backup Domain 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	
									RC 0
	7	6	5	4	3	2	1	0	
Type/Reset	Reserved						PDF	BAKPORF	
							RC	0	RC 1

Bits	Field	Descriptions
[8]	WUPF	<p>External WAKEUP Pin Flag</p> <p>0: The Wakeup pin is not asserted 1: The Wakeup pin is asserted</p> <p>This bit is set by hardware when the WAKEUP pin asserts and is cleared by software read. Software should read this bit to clear it after a system wake up from the power saving mode.</p>
[1]	PDF	<p>Power Down Flag</p> <p>0: Wakeup from abnormal V_{DD15} shutdown (Loss of V_{DD15} is unexpected) 1: Wakeup from Power-Down mode. The loss of V_{DD15} is under expectation.</p> <p>This bit is set by hardware when the system has successfully entered the Power-Down mode This bit is cleared by software read.</p>
[0]	BAKPORF	<p>Backup Domain Reset Flag</p> <p>0: Backup Domain reset does not occur 1: Backup Domain reset occurs</p> <p>This bit is set by hardware when Backup Domain reset occurs, either a Backup Domain power on reset or Backup Domain software reset. The bit is cleared by software read. This bit must be cleared after the system is first powered, otherwise it will be impossible to detect when a Backup 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 Backup Domain is ready for access. It must be implemented after the Backup Domain is first powered up.</p>

Backup Domain Control Register – BAKCR

This register provides power control bits for the Deep-Sleep and Power-Down modes.

Offset: 0x104

Reset value: 0x0000_0000 (Reset only by Backup Domain 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	D MOSSTS	Reserved		V15RDYSC	Reserved		WUPIEN	WUPEN		
	RO	0		RW	0		RW	0	RW	0
	7	6	5	4	3	2	1	0		
Type/Reset	D MOSON	Reserved			LDOOFF	LDOLCM	Reserved	BAKRST		
	RW	0			RW	0	RW	0	WO	0

Bits	Field	Descriptions
[15]	D MOSSTS	Depletion MOS Status This bit is set to 1 if the D MOSON bit in this register has been set to 1. This bit is cleared to 0 if the D MOSON bit has been set to 0 or if a POR/PDR reset occurred.
[12]	V15RDYSC	V_{DD15} Ready Source Selection. 0: BKISO bit in the LPCR register located in the CKCU 1: V_{DD15} POR Setting this bit to determine what control signal of isolation cells is used to disable the isolation function of the V_{DD15} to V_{DD} power domain level shifter.
[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 LPWUP interrupt in the NVIC unit when both the WUPEN and WUPF bits are set to 1.

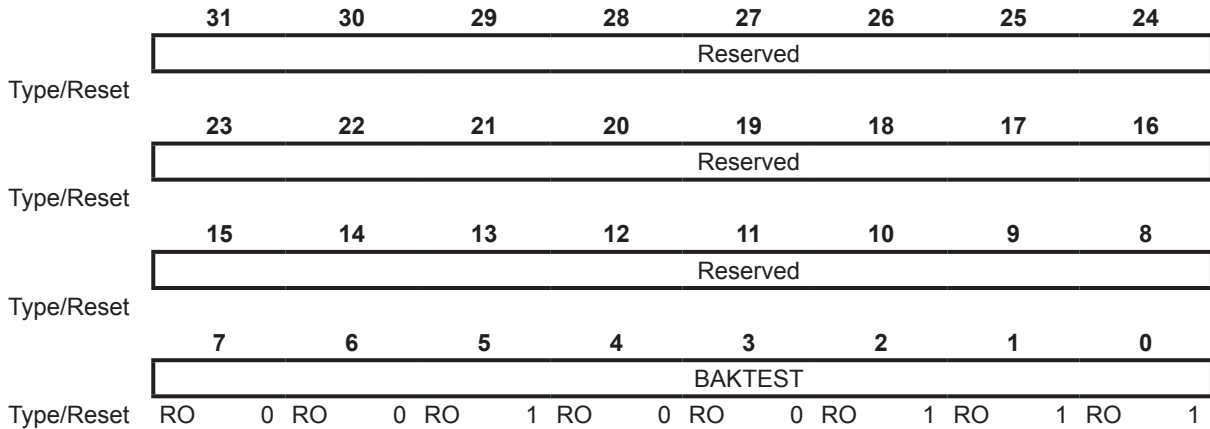
Bits	Field	Descriptions
[8]	WUPEN	<p>External WAKEUP Pin Enable</p> <p>0: Disable WAKEUP pin function. 1: Enable WAKEUP pin function.</p> <p>The Software can set the WUPEN bit as 1 to enable the WAKEUP pin function before entering the power saving mode. When WUPEN = 1, a rising edge on the WAKEUP pin wakes up the system from the power saving mode. As the WAKEUP pin is active high, this bit will set an input pull down mode when the bit is high. The corresponding register bits which should be properly setup are the PBPDP[12] to 1 in the PBPDR register, the PBPU [12] to 0 in the PBPUR register and the PBCFG12 [3:0] field to 0x0F in the GPBCFGHR register.</p> <p>Note: This bit is reset by a system reset or a Backup Domain reset. Because this bit is located in the Backup Domain, after 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 Backup 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 Backup domain ISO signal disabled. The delay time needed is a minimum of three 32KHz clock periods until the bit reset activity has finished.</p>
[7]	DMOSON	<p>DMOS Control</p> <p>0: DMOS is OFF 1: DMOS is ON</p> <p>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 PORB. If the DMOSON bit is set to 1, the LDO will automatically be turned off when the CPU enters the Deep-Sleep mode.</p>
[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. 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. 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 10mA below and lower static current when the LDOLCM bit is set. It is suitable for CPU is operated at lower speed system clock to get a lower current consumption. This bit will be clear to 0 when the LDO is power down or V_{DD} power domain reset.</p>
[0]	BAKRST	<p>Backup Domain Software Reset</p> <p>0: No action 1: Backup Domain Software Reset is activated - includes all the related RTC and PWRCU registers.</p>

Backup Domain Test Register – BAKTEST

This register specifies a read-only value for the software to recognize whether backup domain is ready for access.

Offset: 0x108

Reset value: 0x0000_0027



Bits	Field	Descriptions
[7:0]	BAKTEST	Backup Domain Test Bits A constant 0x27 will be read when the Backup 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 Backup Domain reset)

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

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

Bits	Field	Descriptions
[16]	LV DEN	Low Voltage Detect Enable 0: Disable Low Voltage Detect 1: Enable Low Voltage Detect Setting this bit to 1 will generate a LVD event when the V_{DDA} power is lower than the voltage set by LVDS bits. Therefore when the LVD function is enabled before the system is into the Deep-Sleep2 (DMOS is turn on and LDO is power down) or Power-Down mode (DMOS and LDO is power down), the LVDEWEN bit has to be enabled to avoid 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

Backup Register n – BAKREGn, n = 0 ~ 9

This register specifies backup register n for storing data during the VDD15 power-off period.

Offset: 0x200 ~ 0x224

Reset value: 0x0000_0000 (Reset only by Backup Domain reset)

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

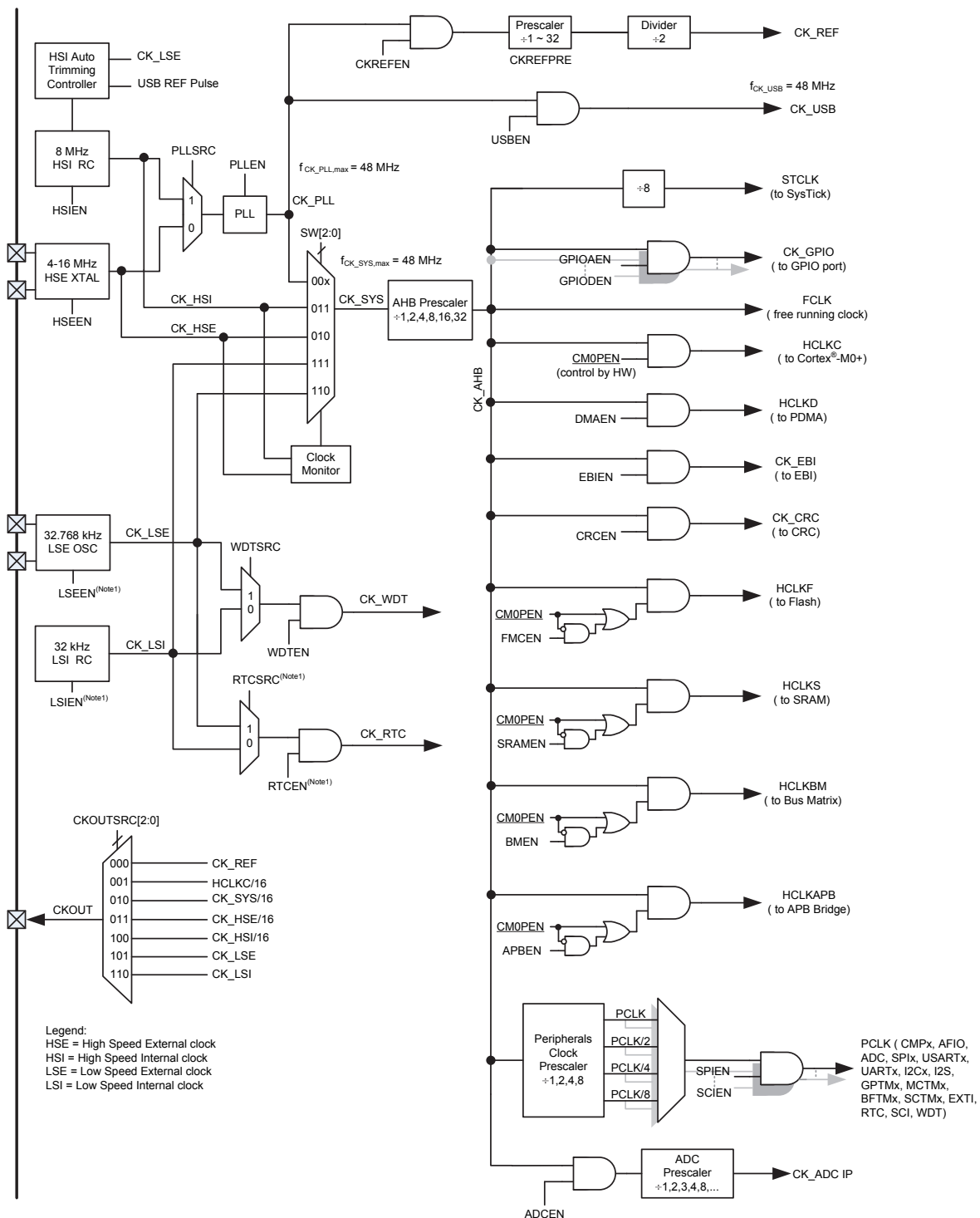
Bits	Field	Descriptions
[31:0]	BAKREGn	Backup Register n (n = 0 ~ 9) These registers are used for data storage in general purpose. The contents of BAKREGn registers will remain even if the V _{DD15} power is lost.

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 clock of AHB, APB, and CPU are derived from system clock (CK_SYS) which can come from HSI, HSE or PLL. Watchdog Timer and Real Time Clock (RTC) use either LSI or LSE as their clock source.

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 (could be broken down or removed or etc), CKCU will force to switch the system clock source to HSI clock to prevent system halt.



Clock Control Unit (CKCU)

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

Function 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.

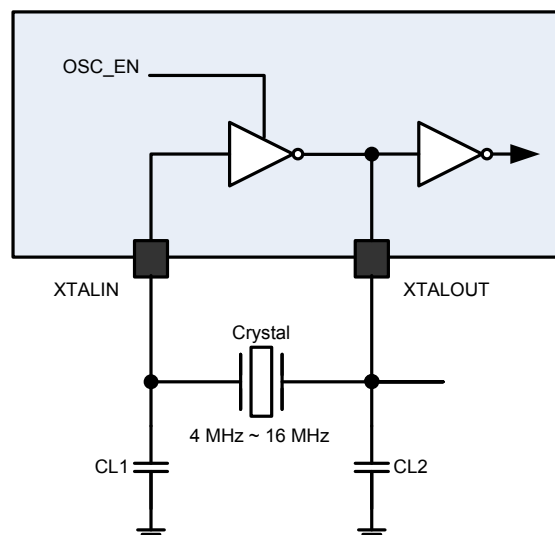


Figure 14. External Crystal, Ceramic, and Resonators for HSE

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. When switching on the HSE oscillator 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, an interrupt will be generated if the related interrupt enable bit HSERDYIE in the Global Clock Interrupt Register (GCIR) is set. 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. An interrupt can be generated if the related interrupt enable bit HSIRDYIE in the Global Clock Interrupt Register (GCIR) is set as the HSI becomes stable. The HSI clock can also be used as the PLL input clock.

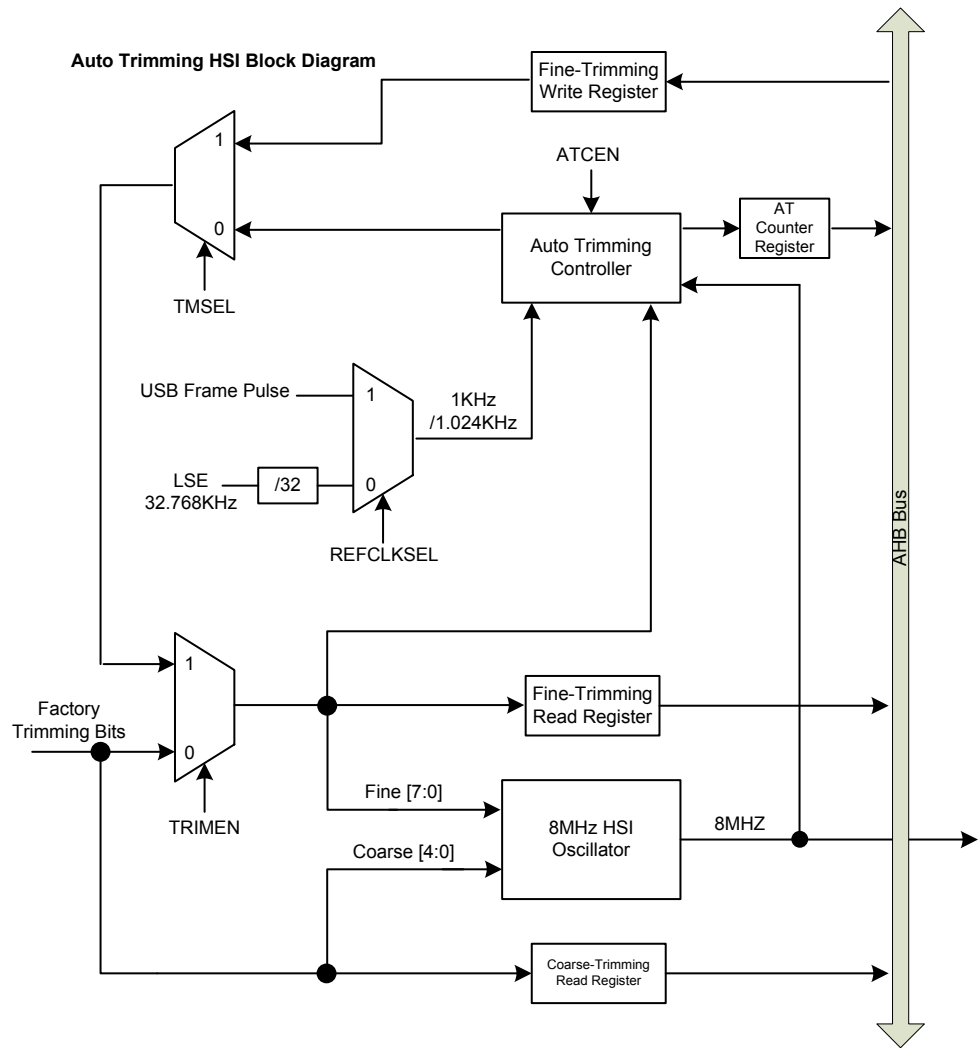
The accuracy of the frequency of the high speed internal RC oscillator HSI can be calibrated via the configuration options, but it is still less accurate than the HSE crystal oscillator. The applications, the environments and the cost will determine the use of the oscillators.

Software could configure PSRCEN bit (Power Saving Wakeup RC Clock Enable) to 1 to force HSI clock to be system clock when wake-up from Deep-Sleep or Power-Down mode. Subsequently, the system clock will be switched 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 the HSE or PLL clock as the 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 $T_A = 25^\circ\text{C}$. But the accuracy is not enough for some applications and environments requirement. Therefore, this device provides the trimming mechanism for HSI frequency calibration using more accurate external reference clock. The detail block diagram is shown as Figure 15.

After reset, the factory trimming value is loaded in the HSICOARSE[4:0] and HSIFINE[7:0] bits in the HSI Control Register (HSICR). The HSI frequency accuracy may be affected by the voltage or temperature variation. If the application has to be driven by more accurate HSI frequency, the HSI frequency can be manually trimmed using the HSIFINE[7:0] bits in the HSI Control Register (HSICR) or automatically adjusted via the Auto Trimming Controller together with an external reference clock in the application. The reference clock can be provided from the low speed external crystal or ceramic resonator oscillator LSE with a 32,768 Hz frequency or a 1ms USB frame synchronous signal.



Clock Control Unit (CCU)

Figure 15. HSI Auto Trimming Block Diagram

Phase Locked Loop – PLL

This PLL can provide 4 ~ 48 MHz clock output which is 1 ~ 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 feedback 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.

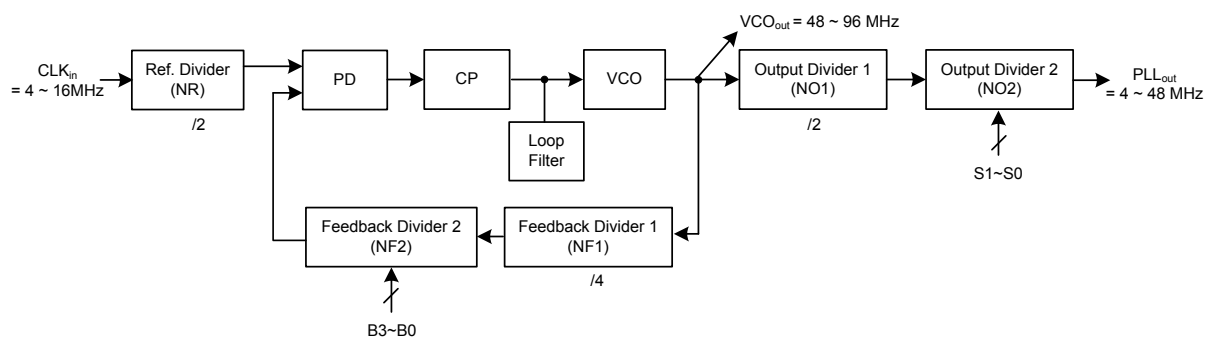


Figure 16. PLL Block Diagram

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

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

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 of 50%, both input and output frequencies are divided by 2. If a given CLK_{in} frequency as the PLL input generates a specific PLL output frequency, it is recommended to load a larger value into the NF2 field to increase the PLL stability and reduce the jitter with but the expense of settling time. The output and feedback divider 2 setup values are described in Table 15 and Table 16. All the configuration bits (S1 ~ S0, B3 ~ B0) in Table 15 and Table 16 are defined in the PLL Configuration Register (PLLCFGR) and PLL Control Register (PLLCR) in the section of Register Definition. Note that the VCO_{OUT} frequency should be in the range from 48 MHz to 96 MHz. If the selected configuration exceeds this range, the PLL output frequency will not be guaranteed 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. An interrupt can be generated if the related interrupt enable bit PLLRDYIE in the Global Clock Interrupt Register (GCIR) is set as the PLL becomes stable.

Table 15. Output Divider2 Value Mapping

Output divider 2 setup bits S[1:0] (POTD bits in the PLLCFGR register)	NO2 (Output divider 2 value)
00	1
01	2
10	4
11	8

Table 16. Feedback Divider2 Value Mapping

Feedback divider2 setup bits B[3:0] (PFBD bits 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 32,768 Hz frequency produces a low power but highly accurate clock source for the 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. An interrupt can be generated if the related interrupt enable bit LSERDYIE in the Global Clock Interrupt Register (GCIR) is set as the LSE becomes stable.

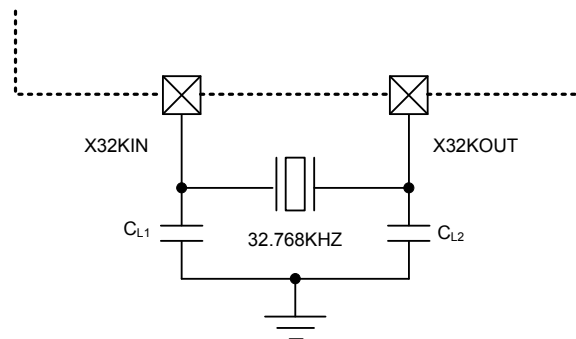


Figure 17. External Crystal, Ceramic, and Resonators for LSE

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 Real-Time-Clock peripheral, Watchdog Timer or system clock. The LSI offers a low cost clock source because no external component is required to make it oscillates. The LSI RC oscillator can be switched on or off by using the LSIEN bit in the RTC Control Register (RTCCR). The LSI frequency accuracy is shown in the datasheet. The LSIRDY flag in the Global Clock Status Register (GCSR) will indicate if the LSI clock is stable. An interrupt can be generated if the related interrupt enable bit LSIRDYIE in the Global Clock Interrupt Register (GCIR) is set as the LSI becomes stable.

Clock Ready Flag

The CKCU provides the corresponding clock ready flags for the HSI, HSE, PLL, LSI, and LSE to indicate whether these clocks are stable. Before using them as the system clock source or other purpose, it is necessary to confirm the specific clock ready flag is set. Software can check the specific clock is ready or not by polling the individual clock ready status bits in GCSR register. Additionally, the CKCU can trigger an interrupt to notify specific clock is ready if the corresponding interrupt enable bit in the GCIR register is set. Software should clear the interrupt status bit in the GCIR register by interrupt service routine.

System Clock (CK_SYS) Selection

After the system reset occurs, the default system clock source CK_SYS will be the high speed internal RC oscillator HSI. The CK_SYS may come from the HSI, HSE, LSE, LSI or PLL output clock and it can be switched from one clock source to another by changing the System Clock Switch bits, SW, in the Global Clock Control Register GCCR. The system will still run under the original clock until the destination clock gets ready when the SW value is changed. 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 source, it is not possible to stop it. More details about the clock enable function is described in the following.

If any event in the following occurs, the HSI will be enabled.

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

If any event in the following occurs, the HSE will be enabled.

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

If any event in the following occurs, the PLL will be enabled.

- Enable USB Enable register. (USBEN)
- Configure clock switch register to PLL (SW)
- Configure PLL enable register to 1. (PLEN)

The system clock selection programming guide is listed in the following.

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

HSE Clock Monitor

The HSE clock monitor function is enabled by the HSE Clock Monitor Enable bit CKMEN in the Global Clock Control Register, GCCR. The HSE clock monitor function should be enabled after the HSE oscillator start-up delay and 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 the HSE oscillator failure event will be generated if the corresponding interrupt enable bit CKSIE in the GCIR is set. This failure interrupt is connected to the CPU Non-Maskable Interrupt, NMI. 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 whose output is used as the system clock, 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, AFIO, section to output the selected clock signal. There are seven output clock signals to be selected via the device clock output source selection bits CKOUTSRC in the Global Clock Configuration Register, GCFGR.

Table 17. 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 register and reset value.

Table 18. CKCU Register Map

Register	Offset	Description	Reset Value
GCFGR	0x000	Global Clock Configuration Register	0x0000_0102
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
APBPCR0	0x038	APB Peripheral Clock Selection Register 0	0x0000_0000
APBPCR1	0x03C	APB Peripheral Clock Selection Register 1	0x0000_0000
HSICR	0x040	HSI Control Register	0xXXXX_0000 where X is undefined
HSIATCR	0x044	HSI Auto Trimming Counter Register	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_0102

	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			
	USBPRE		Reserved								
Type/Reset	RW	0	RW	0							
	15	14	13	12	11	10	9	8			
	CKREFPRE						Reserved		PLLSRC		
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	1	
	7	6	5	4	3	2	1	0			
	Reserved						CKOUTSRC				
Type/Reset						RW	0	RW	1	RW	0

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 Sleep mode1 011: When Chip wants to enter Deep Sleep mode2 100: When Chip wants to enter Power Down mode Others: Reserved Set and reset by hardware.
[23:22]	USBPRE	USB Clock Prescaler Selection 00: CK_USB = CK_PLL 01: CK_USB = CK_PLL / 2 Others: Reserved Set and reset by software to control the USB clock prescaler setting.
[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 the control CK_REF clock prescaler setting.
[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 where $CK_REF = CK_PLL / (CKREFPRE + 1) / 2$ 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.

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
Reserved							
Type/Reset							
23	22	21	20	19	18	17	16
Reserved						PSRCEN	CKMEN
Type/Reset						RW 0	RW 0
15	14	13	12	11	10	9	8
Reserved				HSIEN	HSEEN	PLLEN	HSEGAIN
Type/Reset				RW 1	RW 0	RW 0	RW 0
7	6	5	4	3	2	1	0
Reserved					SW		
Type/Reset					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 power down wakeup. The software can set the PSRCEN bit high before entering the power saving mode in order to reduce the waiting time after a wakeup. When the PSRCEN bit is set to 1, the HSI will be used as the CK_SYS clock source after waking up from the power saving mode. This means that the instruction can be executed early before the original CK_SYS source is stable since the HSI clock is provided to CPU. After the original CK_SYS clock source is ready, the CK_SYS clock will automatically be switched back to the originally selected clock source from the HSI clock.
[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 the hardware detects that the HSE clock is stuck at low or high state, the internal hardware will switch the system clock to internal high speed HSI RC clock.
[11]	HSIEN	Internal High Speed Oscillator Enable 0: Internal 8 MHz RC oscillator is disabled 1: Internal 8 MHz RC oscillator is enabled Set and reset by software. This bit can not be reset if the HSI clock is used as the system clock.
[10]	HSEEN	External High Speed Oscillator Enable 0: External 4 ~ 16 MHz crystal oscillator is disabled 1: External 4 ~ 16 MHz crystal oscillator is enabled Set and reset by software. This bit can not be reset if the HSE clock is used as the system clock or the PLL input clock.
[9]	PLLEN	PLL Enable 0: PLL disabled 1: PLL enabled Set and reset by software. This bit cannot be reset if the PLL clock is used as the system clock.

Bits	Field	Descriptions
[8]	HSEGAIN	External High Speed Oscillator Gain Selection 0: HSE in low gain mode 1: HSE in high gain mode
[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. When switch the system clock using the SW bit, the system clock will be not immediately switched and a certain delay is necessary. The system clock source selected by the SW bits can be indicated in the CKSWST bits in the clock source status register CKST to make sure which clock is currently used as the system clock. Note that the HSI oscillator will be forced as the system clock when the HSE clock failure is detected as the HSE clock monitor function is enabled.

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	Reserved	
			RO	1 RO	0 RO	1 RO	0 RO	0	

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

Global Clock Interrupt Register – GCIR

This register specifies the 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	
									0
									WC

Bits	Field	Descriptions
[16]	CKSIE	<p>Clock Stuck Interrupt Enable</p> <p>0: Disable clock failure interrupt</p> <p>1: Enable clock failure interrupt</p> <p>Set and reset by software to enable or disable the clock failure interrupt caused by clock monitor.</p>
[0]	CKSF	<p>Clock Stuck Interrupt Flag</p> <p>0: Clock works normally</p> <p>1: HSE clock is stuck</p> <p>Reset by software (Write 1 clear). Set by hardware when the HSE clock is stuck and the CKMEN bit is set.</p>

PLL Configuration Register – PLLCFGR

This register specifies the PLL configurations.

Offset: 0x018

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24			
Type/Reset	Reserved					PFBD					
						RW	0	RW	0	RW	0
	23	22	21	20	19	18	17	16			
Type/Reset	PFBD	POTD			Reserved						
	RW	0	RW	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										

Bits	Field	Descriptions
[26:23]	PFBD	PLL VCO Output Clock Feedback Divider (B3 ~ B0 in Figure 16) The feedback Divider divides the output clock from the PLL VCO.
[22:21]	POTD	PLL Output Clock Divider (S1 ~ S0 in Figure 16)

PLL Control Register – PLLCR

This register specifies the PLL Bypass mode.

Offset: 0x01C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	PLLBPS	Reserved						
	RW	0						
	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							

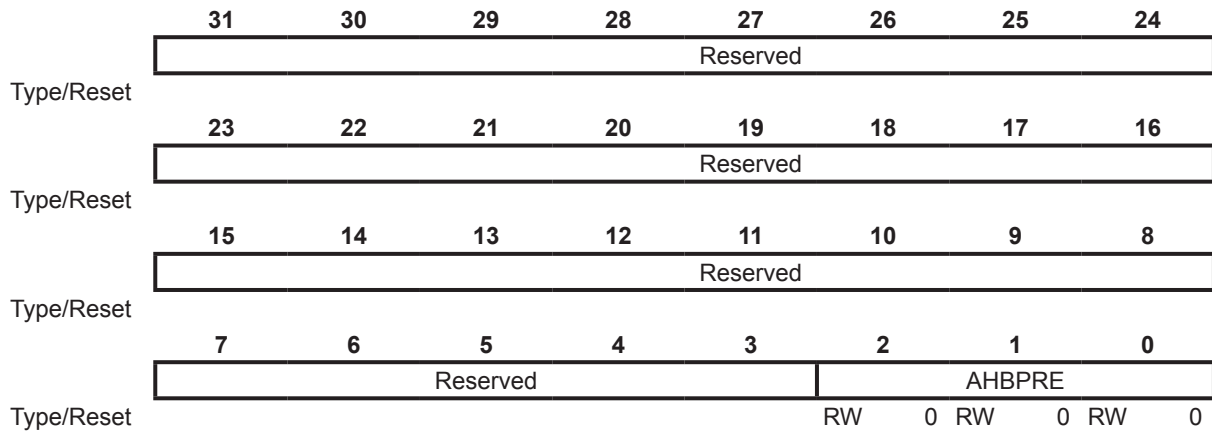
Bits	Field	Descriptions
[31]	PLLBPS	PLL Bypass Mode Enable 0: Disable PLL Bypass mode 1: Enable PLL Bypass mode which acts as FOUT = FIN

AHB Configuration Register – AHBCFGR

This register specifies the system clock frequency.

Offset: 0x020

Reset value: 0x0000_0000



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

AHB Clock Control Register – AHBCCR

This register specifies the AHB clock enable control bits.

Offset: 0x024

Reset value: 0x0000_0005

31	30	29	28	27	26	25	24	
Reserved								
Type/Reset								
23	22	21	20	19	18	17	16	
Reserved				PDEN	PCEN	PBEN	PAEN	
Type/Reset				RW	0	RW	0	
15	14	13	12	11	10	9	8	
Reserved		CRCEN	EBIEN	CKREFEN	USBEN	Reserved		
Type/Reset		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

Bits	Field	Descriptions
[19]	PDEN	GPIO Port D Clock Enable 0: Port D clock is disabled 1: Port D 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
[13]	CRCEN	CRC Module Clock Enable 0: CRC clock is disabled 1: CRC clock is enabled Set and reset by software.
[12]	EBIEN	EBI Module Clock Enable 0: EBI clock is disabled 1: EBI clock is enabled Set and reset by software.
[11]	CKREFEN	CK_REF Clock Enable 0: CK_REF clock is disabled 1: CK_REF clock is enabled Set and reset by software

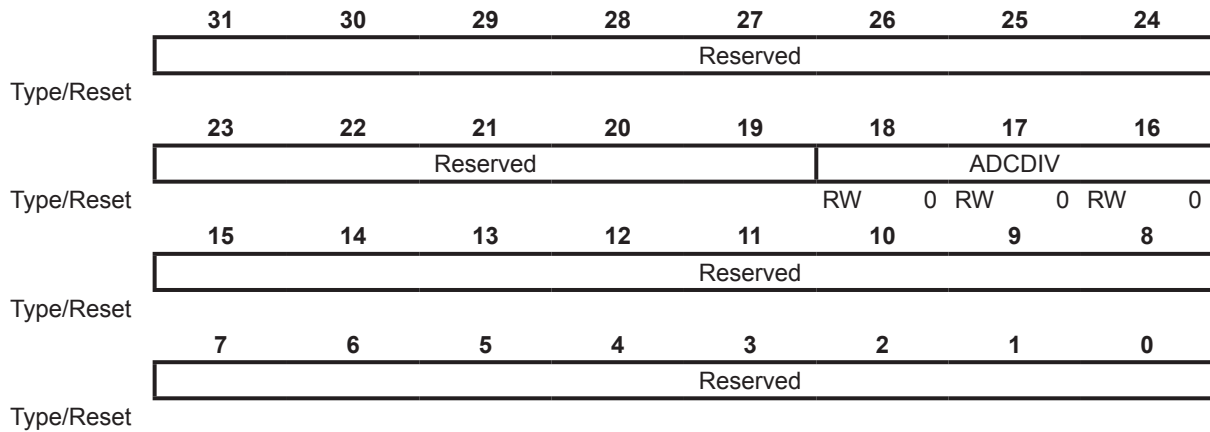
Bits	Field	Descriptions
[10]	USBEN	<p>USB Clock Enable</p> <p>0: USB clock is disabled 1: USB clock is enabled</p> <p>Set and reset by software</p>
[6]	APBEN	<p>APB bridge Clock Enable</p> <p>0: APB bridge clock is automatically disabled by hardware during Sleep mode 1: APB bridge clock is always enabled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

APB Configuration Register – APBCFGR

This register specifies the frequency of ADC conversion clock.

Offset: 0x028

Reset value: 0x0000_0000



Bits	Field	Descriptions
[18:16]	ADCDIV	ADC Clock Frequency Division Selection 000: CK_ADC = (CK_AHB / 1) 001: CK_ADC = (CK_AHB / 2) 010: CK_ADC = (CK_AHB / 4) 011: CK_ADC = (CK_AHB / 8) 100: CK_ADC = (CK_AHB / 16) 101: CK_ADC = (CK_AHB / 32) 110: CK_ADC = (CK_AHB / 64) 111: CK_ADC = (CK_AHB / 3)
		Set and reset by software to control the ADC conversion clock division factor.

APB Clock Control Register 0 – APBCCR0

This register specifies the APB peripherals clock enable bits.

Offset: 0x02C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved				SCI1EN	Reserved	I2SEN	SCI0EN
					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	EXTIEN	AFIOEN	Reserved		UR1EN	UR0EN	USR1EN	USR0EN
	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
Type/Reset	Reserved		SPI1EN	SPI0EN	Reserved		I2C1EN	I2C0EN
			RW	0	RW	0	RW	0

Bits	Field	Descriptions
[27]	SCI1EN	Smart Card Interface 1 Clock Enable 0: SCI1 clock is disabled 1: SCI1 clock is enabled Set and reset by software
[25]	I2SEN	I ² S Interface Clock Enable 0: I ² S clock is disabled 1: I ² S clock is enabled Set and reset by software
[24]	SCI0EN	Smart Card Interface 0 Clock Enable 0: SCI0 clock is disabled 1: SCI0 clock is enabled Set and reset by software
[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.
[11]	UR1EN	UART1 Clock Enable 0: UART1 clock is disabled 1: UART1 clock is enabled Set and reset by software.
[10]	UR0EN	UART0 Clock Enable 0: UART0 clock is disabled 1: UART0 clock is enabled Set and reset by software.

Bits	Field	Descriptions
[9]	USR1EN	USART1 Clock Enable 0: USART1 clock is disabled 1: USART1 clock is enabled Set and reset by software.
[8]	USR0EN	USART0 Clock Enable 0: USART0 clock is disabled 1: USART0 clock is enabled Set and reset by software.
[5]	SPI1EN	SPI1 Clock Enable 0: SPI1 clock is disabled 1: SPI1 clock is enabled Set and reset by software.
[4]	SPI0EN	SPI0 Clock Enable 0: SPI0 clock is disabled 1: SPI0 clock is enabled Set and reset by software.
[1]	I2C1EN	I2C1 Clock Enable 0: I2C1 clock is disabled 1: I2C1 clock is enabled Set and reset by software.
[0]	I2C0EN	I2C0 Clock Enable 0: I2C0 clock is disabled 1: I2C0 clock is enabled Set and reset by software.

Clock Control Unit (CCU)

APB Clock Control Register 1 – APBCCR1

This register specifies the APB peripherals clock enable bits.

Offset: 0x030

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved		SCTM1EN	SCTM0EN	Reserved		ADCCEN		
			RW 0	RW 0				RW 0	
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved	CMPEN	Reserved				BFTM1EN	BFTM0EN	
		RW 0					RW 0	RW 0	
	15	14	13	12	11	10	9	8	
Type/Reset	Reserved						GPTM1EN	GPTM0EN	
							RW 0	RW 0	
	7	6	5	4	3	2	1	0	
Type/Reset	Reserved	BKPREN	Reserved	WDTREN	Reserved		MCTM0EN		
		RW 0		RW 0				RW 0	

Bits	Field	Descriptions
[29]	SCTM1EN	SCTM1 Clock Enable 0: SCTM1 clock is disabled 1: SCTM1 clock is enabled Set and reset by software.
[28]	SCTM0EN	SCTM0 Clock Enable 0: SCTM0 clock is disabled 1: SCTM0 clock is enabled Set and reset by software.
[24]	ADCCEN	ADC Controller Clock Enable 0: ADC clock is disabled 1: ADC clock is enabled Set and reset by software.
[22]	CMPEN	CMP Clock Enable 0: CMP clock is disabled 1: CMP clock is enabled Set and reset by software.
[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.
[9]	GPTM1EN	GPTM1 Clock Enable 0: GPTM1 clock is disabled 1: GPTM1 clock is enabled Set and reset by software.

Bits	Field	Descriptions
[8]	GPTM0EN	GPTM0 Clock Enable 0: GPTM0 clock is disabled 1: GPTM0 clock is enabled Set and reset by software.
[6]	BKPREN	Backup Domain Clock Enable for Registers Access 0: RTC clock is disabled 1: RTC clock is enabled Set and reset by software.
[4]	WDTREN	Watchdog Timer Clock Enable for Registers Access 0: Watchdog Timer clock is disabled 1: Watchdog Timer clock is enabled Set and reset by software.
[0]	MCTM0EN	MCTM0 Clock Enable 0: MCTM0 clock is disabled 1: MCTM0 clock is enabled Set and reset by software.

Clock Source Status Register – CKST

This register specifies the clock source status.

Offset: 0x034

Reset value: 0x0100_0003

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

Bits	Field	Descriptions
[26:24]	HSIST	Internal High Speed Clock Occupation Status (CK_HSI) xx1: HSI is used by System Clock (CK_SYS) (SW = 0x03) x1x: HSI is used by PLL 1xx: HSI is used by Clock Monitor
[17:16]	HSEST	External High Speed Clock Occupation Status (CK_HSE) x1: HSE is used by System Clock (CK_SYS) (SW = 0x02) 1x: HSE is used by 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 the APB peripheral clock prescaler selection.

Offset: 0x038

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	UR1PCLK		UR0PCLK		USR1PCLK		USR0PCLK	
Type/Reset	RW	0	RW	0	RW	0	RW	0
	23	22	21	20	19	18	17	16
	GPTM1PCLK		GPTM0PCLK		Reserved		MCTM0PCLK	
Type/Reset	RW	0	RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8
	BFTM1PCLK		BFTM0PCLK		Reserved			
Type/Reset	RW	0	RW	0	RW	0		
	7	6	5	4	3	2	1	0
	SPI1PCLK		SPI0PCLK		I2C1PCLK		I2C0PCLK	
Type/Reset	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[31:30]	UR1PCLK	UART1 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
[29:28]	UR0PCLK	UART0 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
[27:26]	USR1PCLK	USART1 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]	USR0PCLK	USART0 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
[23:22]	GPTM1PCLK	GPTM1 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
[21:20]	GPTM0PCLK	GPTM0 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]	MCTM0PCLK	MCTM0 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]	BFTM1PCLK	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]	BFTM0PCLK	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
[7:6]	SPI1PCLK	SPI1 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
[5:4]	SPI0PCLK	SPI0 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
[3:2]	I2C1PCLK	I2C1 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]	I2C0PCLK	I2C0 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 the APB peripheral clock prescaler selection.

Offset: 0x03C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24		
Type/Reset	Reserved				RW	0	RW	0	RW	0
	23	22	21	20	19	18	17	16		
Type/Reset	Reserved		I2SPCLK		SCI1PCLK		SCI0PCLK			
		RW	0	RW	0	RW	0	RW	0	
	15	14	13	12	11	10	9	8		
Type/Reset	BKPRPCLK		WDTRPCLK		Reserved		CMPPCLK			
	RW	0	RW	0	RW	0	RW	0	RW	
	7	6	5	4	3	2	1	0		
Type/Reset	Reserved		ADCCPCLK		EXTIPCLK		AFIOPCLK			
	RW	0	RW	0	RW	0	RW	0	RW	

Bits	Field	Descriptions
[27:26]	SCTM1PCLK	SCTM1 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]	SCTM0PCLK	SCTM0 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]	I2SPCLK	I ² S 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
[19:18]	SCI1PCLK	SCI1 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]	SCI0PCLK	SCI0 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
[15:14]	BKPRCLK	Backup 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
[9:8]	CMPPCLK	CMP 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
[5:4]	ADCCPCLK	ADC Controller 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
[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	X	
	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	X	
	15	14	13	12	11	10	9	8		
	Reserved									
Type/Reset										
	7	6	5	4	3	2	1	0		
	FLOCK	Reserved	REFCLKSEL	TMSEL	Reserved	Reserved	ATCEN	TRIMEN		
Type/Reset	RO	0	RW	0	RW	0	RW	0	RW	

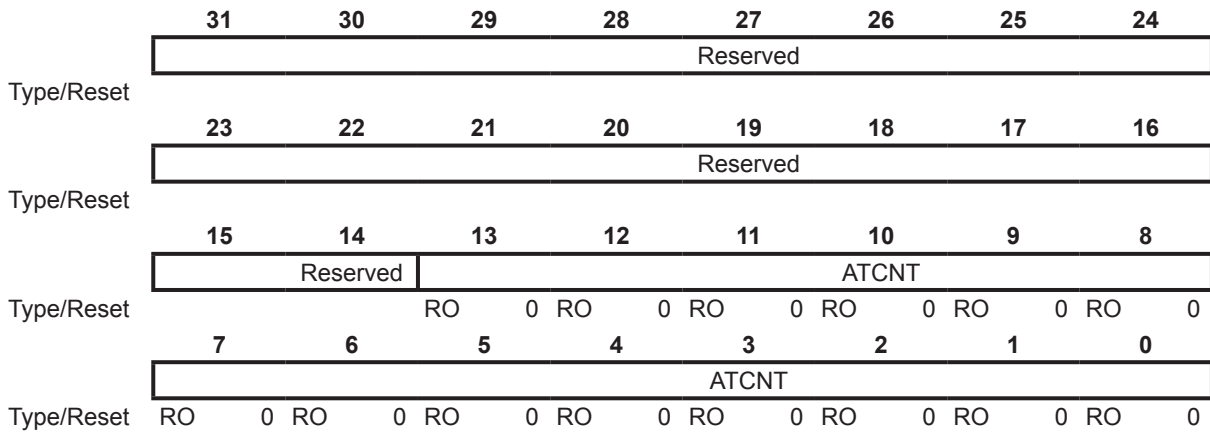
Bits	Field	Descriptions
[28:24]	HSICOARSE	HSI Clock Coarse Trimming Value These bits are initialized automatically at startup. They are adjusted by factory trimming and can not be trimmed by program.
[23:16]	HSIFINE	HSI Clock Fine Trimming Value These bits are initialized automatically at startup. They 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 higher accuracy or compensate the variations in voltage and temperature that influence the HSI frequency. It can be programmed by software or Auto-Trimming Controller (ATC) 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
[5]	REFCLKSEL	Reference Clock Selection 0: Select 32.768KHz external low speed clock source (LSE) 1: Select 1KHz USB frame pulse This bit is 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 the ATC hardware or user program via the HSIFINE[7:0] bits in the HSI Control Register.
[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 The bit enables the HSI RC oscillator trimming function by the 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



Clock Control Unit (CCU)

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

Low Power Control Register – LPCR

This register specifies the 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							RW	0
	7	6	5	4	3	2	1	0	
Type/Reset	Reserved							RW	0

Bits	Field	Descriptions
[8]	USBSLEEP	USB Sleep Software Control Enable 0: Disable 1: Enable USB Software Sleeping Set and reset by software. Please refer to the Power Control Unit chapter for more information.
[0]	BKISO	Backup Domain Isolation Control 0: Backup domain is isolated from other power domain 1: Backup domain is accessible by other power domain Set and reset by software. Please refer to the Power Control Unit chapter for more information.

MCU Debug Control Register – MCUIDBGCR

This register specifies the MCU debug control.

Offset: 0x304

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	RW	0	RW	0	RW	0	RW	0	RW
	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
[23]	DBSCTM1	SCTM1 Debug Mode Enable 0: SCTM1 counter continues to count even if the core is halted 1: SCTM1 counter stops counting when the core is halted Set and reset by software.
[22]	DBSCTM0	SCTM0 Debug Mode Enable 0: SCTM0 counter continues to count even if the core is halted 1: SCTM0 counter stops counting when the core is halted Set and reset by software.
[21]	DBSCI1	SCI1 Debug Mode Enable 0: Same behavior as in normal mode 1: SCI1 timeout is frozen when the core is halted Set and reset by software.
[20]	DBTRACE	TRACESWO Debug Mode Enable 0: Disable TRACESWO output 1: Enable TRACESWO output Set and reset by software.
[19]	DBUR1	UART1 Debug Mode Enable 0: Same behavior as in normal mode 1: UART1 FIFO timeout is frozen when the core is halted Set and reset by software.
[18]	DBUR0	UART0 Debug Mode Enable 0: Same behavior as in normal mode 1: UART0 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 stops counting when the core is halted Set and reset by software.

Bits	Field	Descriptions
[16]	DBBFTM0	BFTM0 Debug Mode Enable 0: BFTM0 counter continues to count even if the core is halted 1: BFTM0 counter stops counting when the core is halted Set and reset by software.
[15]	DBSCIO	SCIO Debug Mode Enable 0: Same behavior as in normal mode 1: SCIO timeout is frozen when the core is halted Set and reset by software.
[14]	DBDSLP2	Debug Deep-Sleep2 0: LDO = Off (but turn on DMOS), FCLK = Off, and HCLK = Off in Deep-Sleep2 1: LDO = On, FCLK = On, and HCLK = On in Deep-Sleep2 Set and reset by software.
[13]	DBI2C1	I2C1 Debug Mode Enable 0: Same behavior as in normal mode 1: I2C1 timeout is frozen when the core is halted Set and reset by software.
[12]	DBI2C0	I2C0 Debug Mode Enable 0: Same behavior as in normal mode 1: I2C0 timeout is frozen when the core is halted Set and reset by software.
[11]	DBSPI1	SPI1 Debug Mode Enable 0: Same behavior as in normal mode 1: SPI1 FIFO timeout is frozen when the core is halted Set and reset by software.
[10]	DBSPI0	SPI0 Debug Mode Enable 0: Same behavior as in normal mode 1: SPI0 FIFO timeout is frozen when the core is halted Set and reset by software.
[9]	DBUSR1	USART1 Debug Mode Enable 0: Same behavior as in normal mode 1: USART1 FIFO timeout is frozen when the core is halted Set and reset by software.
[8]	DBUSR0	USART0 Debug Mode Enable 0: Same behavior as in normal mode 1: USART0 FIFO timeout is frozen when the core is halted Set and reset by software.
[7]	DBGPTM1	GPTM1 Debug Mode Enable 0: GPTM1 counter continues to count even if the core is halted 1: GPTM1 counter stops counting when the core is halted Set and reset by software.
[6]	DBGPTM0	GPTM0 Debug Mode Enable 0: GPTM0 counter continues to count even if the core is halted 1: GPTM0 counter stops counting when the core is halted Set and reset by software.
[4]	DBMCTM	MCTM Debug Mode Enable 0: MCTM counter continues to count even if the core is halted 1: MCTM counter stops counting 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 stops counting when the core is halted Set and reset by software.

Bits	Field	Descriptions
[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]	DBDSP1	Debug Deep-Sleep1 0: LDO = Low power mode, FCLK = Off, and HCLK = Off in Deep-Sleep1 1: LDO = On, FCLK = On, and HCLK = On in Deep-Sleep1 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.

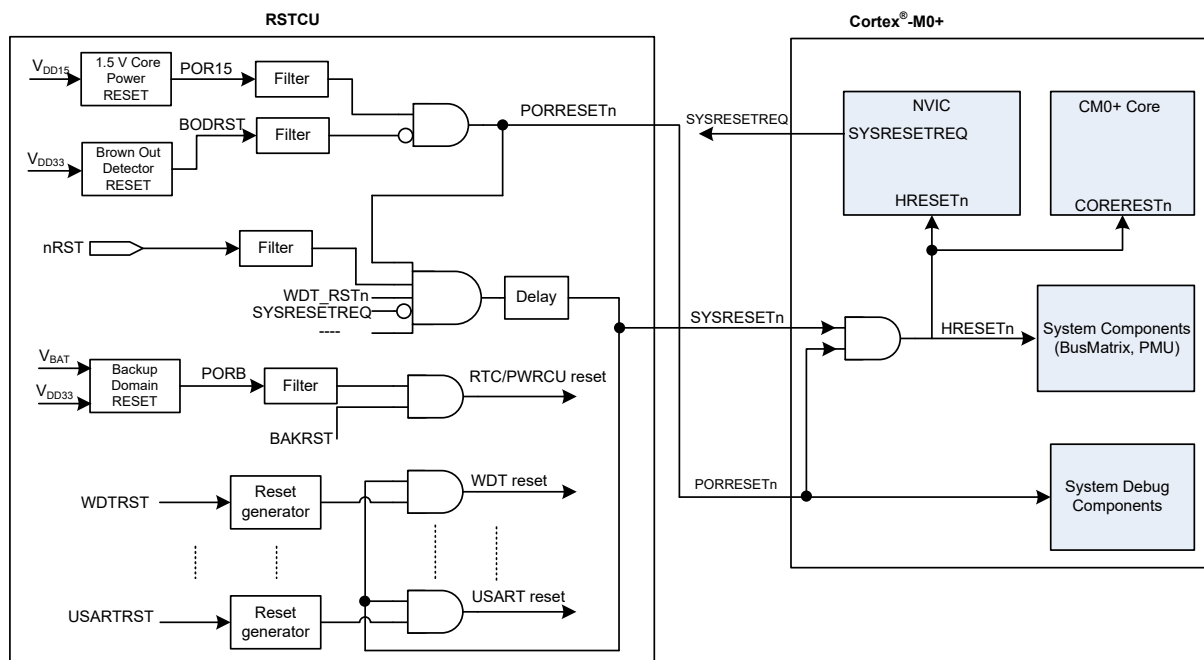
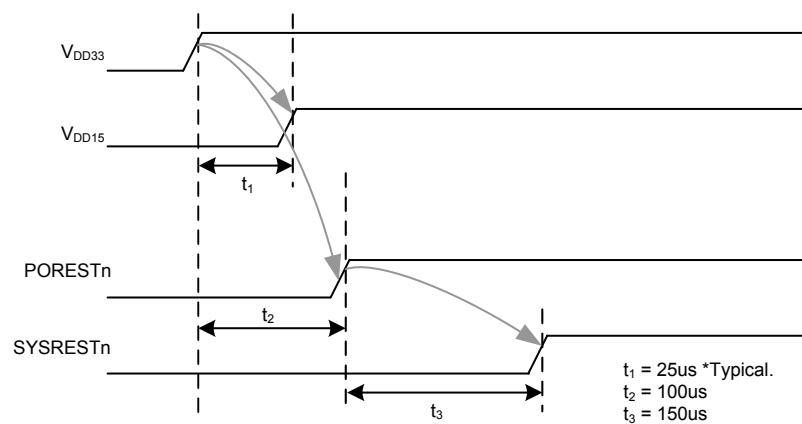


Figure 18. RSTCU Block Diagram

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 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.



* This timing is dependent on the internal LDO regulator output capacitor value.

Figure 19. Power On Reset Sequence

System Reset

A system reset is generated by a power on reset (PORRESESt_n), a Watchdog Timer reset (WDT_RSESt_n), nRST pin or a software reset (SYSRESEStREQ) event. For more information about SYSRESEStREQ 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 USART0 reset via the USR0RST bit in the APBPRSTR0 register.

Register Map

The following table shows the RSTCU registers and reset values.

Table 19. RSTCU Register Map

Register	Offset	Description	Reset Value
RSTCU Base Address = 0x4008_8000			
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				PORSTF	WDTRSTF	EXTRSTF	NVICRSTF	
					WC	1 WC	0 WC	0 WC	0

Bits	Field	Descriptions
[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.

Bits	Field	Descriptions
[0]	NVICRSTF	<p>NVIC Reset Flag</p> <p>0: No NVIC asserting system reset occurred 1: NVIC asserting system reset occurred</p> <p>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.</p>

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	
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			PERST	PDRST	PCRST	PBRST	PARST	
				RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0	
Type/Reset	CRCRST	EBIRST	USBRST	Reserved				DMARST	
	RW	0	RW	0	RW	0		RW	0

Bits	Field	Descriptions
[12]	PERST	<p>GPIO Port E Reset Control</p> <p>0: No reset 1: Reset Port E</p> <p>This bit is set by software and cleared to 0 by hardware automatically.</p>
[11]	PDRST	<p>GPIO Port D Reset Control</p> <p>0: No reset 1: Reset Port D</p> <p>This bit is set by software and cleared to 0 by hardware automatically.</p>
[10]	PCRST	<p>GPIO Port C Reset Control</p> <p>0: No reset 1: Reset Port C</p> <p>This bit is set by software and cleared to 0 by hardware automatically.</p>
[9]	PBRST	<p>GPIO Port B Reset Control</p> <p>0: No reset 1: Reset Port B</p> <p>This bit is set by software and cleared to 0 by hardware automatically.</p>
[8]	PARST	<p>GPIO Port A Reset Control</p> <p>0: No reset 1: Reset Port A</p> <p>This bit is set by software and cleared to 0 by hardware automatically.</p>

Bits	Field	Descriptions
[7]	CRCRST	CRC Reset Control 0: No reset 1: Reset CRC This bit is set by software and cleared to 0 by hardware automatically.
[6]	EBIRST	EBI Reset Control 0: No reset 1: Reset EBI This bit is set by software and cleared to 0 by hardware automatically.
[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				SCI1RST	Reserved	I2SRST	SCI0RST
					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	EXTIRST	AFIORST	Reserved		UR1RST	UR0RST	USR1RST	USR0RST
	RW 0	RW 0			RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
Type/Reset	Reserved		SPI1RST	SPI0RST	Reserved		I2C1RST	I2C0RST
			RW 0	RW 0			RW 0	RW 0

Bits	Field	Descriptions
[27]	SCI1RST	Smart Card Interface 1 Reset Control 0: No reset 1: Reset Smart Card Interface This bit is set by software and cleared to 0 by hardware automatically.
[25]	I2SRST	I2S Reset Control 0: No reset 1: Reset I2S This bit is set by software and cleared to 0 by hardware automatically.

Bits	Field	Descriptions
[24]	SCI0RST	Smart Card Interface 0 Reset Control 0: No reset 1: Reset Smart Card Interface This bit is set by software and cleared to 0 by hardware automatically.
[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.
[11]	UR1RST	UART1 Reset Control 0: No reset 1: Reset UART1 This bit is set by software and cleared to 0 by hardware automatically.
[10]	UR0RST	UART0 Reset Control 0: No reset 1: Reset UART0 This bit is set by software and cleared to 0 by hardware automatically.
[9]	USR1RST	USART1 Reset Control 0: No reset 1: Reset USART1 This bit is set by software and cleared to 0 by hardware automatically.
[8]	USR0RST	USART0 Reset Control 0: No reset 1: Reset USART0 This bit is set by software and cleared to 0 by hardware automatically.
[5]	SPI1RST	SPI1 Reset Control 0: No reset 1: Reset SPI1 This bit is set by software and cleared to 0 by hardware automatically.
[4]	SPI0RST	SPI0 Reset Control 0: No reset 1: Reset SPI0 This bit is set by software and cleared to 0 by hardware automatically.
[1]	I2C1RST	I2C1 Reset Control 0: No reset 1: Reset I2C1 This bit is set by software and cleared to 0 by hardware automatically.
[0]	I2C0RST	I2C0 Reset Control 0: No reset 1: Reset I2C0 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		SCTM1RST	SCTM0RST	Reserved		ADCRST		
			RW 0	RW 0				RW 0	
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved	CMPRST	Reserved				BFTM1RST	BFTM0RST	
		RW 0					RW 0	RW 0	
	15	14	13	12	11	10	9	8	
Type/Reset	Reserved						GPTM1RST	GPTM0RST	
							RW 0	RW 0	
	7	6	5	4	3	2	1	0	
Type/Reset	Reserved			WDTRST	Reserved		MCTMRST		
				RW 0				RW 0	

Bits	Field	Descriptions
[29]	SCTM1RST	SCTM1 Reset Control 0: No reset 1: Reset SCTM1 This bit is set by software and cleared to 0 by hardware automatically.
[28]	SCTM0RST	SCTM0 Reset Control 0: No reset 1: Reset SCTM0 This bit is set by software and cleared to 0 by hardware automatically.
[24]	ADCRST	A/D Converter Reset Control 0: No reset 1: Reset A/D Converter This bit is set by software and cleared to 0 by hardware automatically.
[22]	CMPRST	Comparator Controller Reset Control 0: No reset 1: Reset CMP This bit is set by software and cleared to 0 by hardware automatically.
[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.
[9]	GPTM1RST	GPTM1 Reset Control 0: No reset 1: Reset GPTM1 This bit is set by software and cleared to 0 by hardware automatically.

Bits	Field	Descriptions
[8]	GPTM0RST	GPTM0 Reset Control 0: No reset 1: Reset GPTM0 This bit is set by software and cleared to 0 by hardware automatically.
[4]	WDRST	Watchdog Timer Reset Control 0: No reset 1: Reset Watchdog Timer This bit is set by software and cleared to 0 by hardware automatically.
[0]	MCTMRST	MCTM Reset Control 0: No reset 1: Reset MCTM This bit is set by software and cleared to 0 by hardware automatically.

8 General Purpose I/O (GPIO)

Introduction

There are up to 51 General Purpose I/O port, GPIO, named PA0 ~ PA15, PB0 ~ PB15, PC0 ~ PC15 and PD0 ~ PD3 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 really available General Purpose I/O port numbers are dependent on the device specification and package type. Please refer the device data sheet for detail 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).

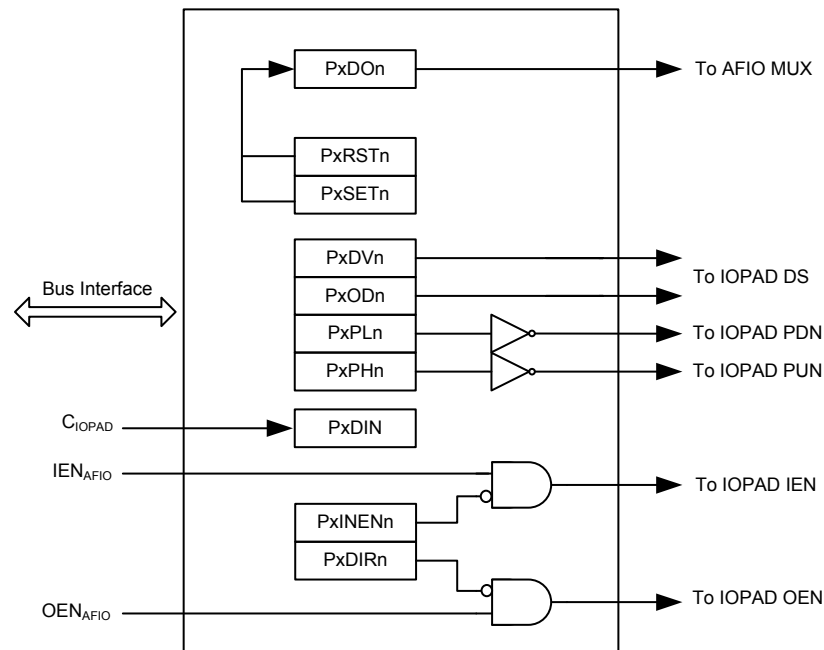


Figure 20. GPIO Block Diagram

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.

- PA8: Input enable with internal pull-up
- 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 ~ D). 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 PxDOUTR. 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 and reset control register PxSRR or the port output reset control 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.

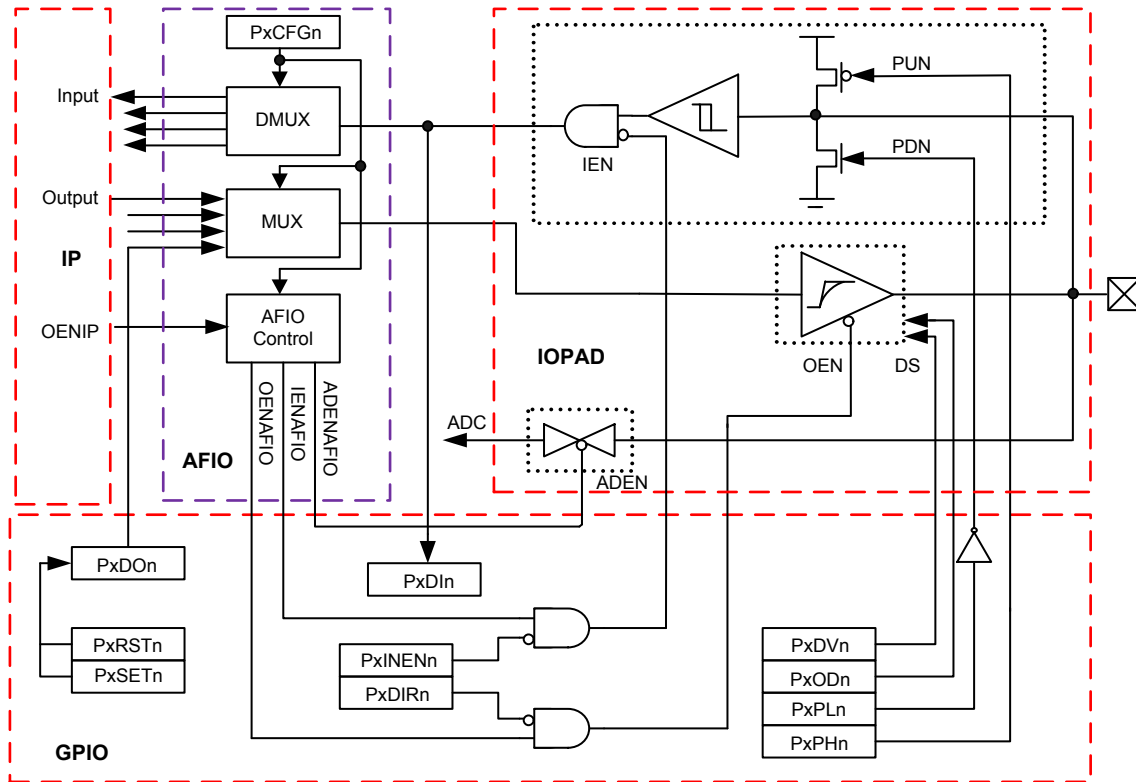


Figure 21. AFIO/GPIO Control Signal

- | | |
|--|--------------------------------------|
| PxDIn/PxDO n (x=A ~ D): Data Input/Data Output | PxRSTn/PxSETn (x=A ~ D): Reset/Set |
| PxDIRn (x=A ~ D): Direction | PxINENn (x=A ~ D): Input Enable |
| PxDVn (x=A ~ D): Output Drive | PxODn (x=A ~ D): Open Drain |
| PxPLn/PxPHn (x=A ~ D): Pull Low/High | PxCFGn (x=A ~ D): AFIO Configuration |

Table 20. 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 ~ D) 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 GPxCFGLR, where x = A ~ D). If the value in the PxLOCKR 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 ~ D.

Table 21. 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_3300
PAPUR	0x008	Port A Pull-Up Selection Register	0x0000_3300
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_3300
PADOUTR	0x020	Port A Data Output Register	0x0000_0000
PASRR	0x024	Port A Output Set and 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 and 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 and Reset Control Register	0x0000_0000
PCRR	0x028	Port C Output Reset Control Register	0x0000_0000
GPIO D Base Address = 0x400B_6000			
PDIRCR	0x000	Port D Data Direction Control Register	0x0000_0000
PDINER	0x004	Port D Input Function Enable Control Register	0x0000_0000
PDPUR	0x008	Port D Pull-Up Selection Register	0x0000_0000
PDPDR	0x00C	Port D Pull-Down Selection Register	0x0000_0000
PDODR	0x010	Port D Open Drain Selection Register	0x0000_0000
PDDVR	0x014	Port D Drive Current Selection Register	0x0000_0000
PDLOCKR	0x018	Port D Lock Register	0x0000_0000
PDDINR	0x01C	Port D Data Input Register	0x0000_0000
PDDOUTR	0x020	Port D Data Output Register	0x0000_0000
PDSRR	0x024	Port D Output Set and Reset Control Register	0x0000_0000
PDRR	0x028	Port D Output Reset Control Register	0x0000_0000

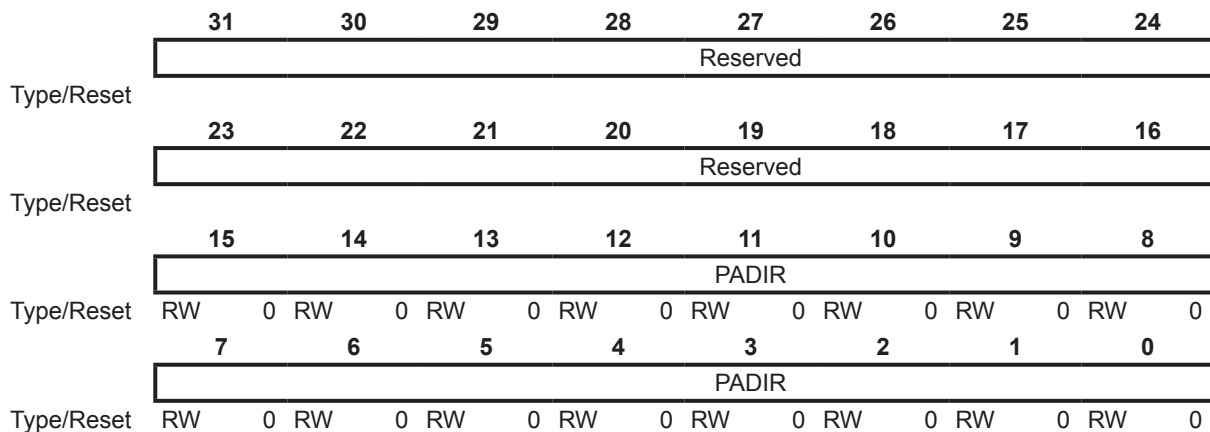
Register Descriptions

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



Bits	Field	Descriptions
[15:0]	PADIRn	GPIO Port A pin n Direction Control Bits (n = 0 ~ 15) 0: Pin n is input mode 1: Pin n is 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_3300

	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	1	RW	1	RW	0	RW	0	RW	1	RW	1
	7	6	5	4	3	2	1	0								
Type/Reset	PAINEN															
	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[15:0]	PAINENn	GPIO Port A pin n Input Enable Control Bits (n = 0 ~ 15) 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 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_3300

	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	PAPU							
Type/Reset	RW	0	RW	0	RW	1	RW	1	RW	0	RW	0	RW	1	RW	1
	7	6	5	4	3	2	1	0	PAPU							
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[15:0]	PAPUn	GPIO Port A pin n Pull-Up Selection Control Bits (n = 0 ~ 15) 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 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	Reserved							
Type/Reset																
	23	22	21	20	19	18	17	16	Reserved							
Type/Reset																
	15	14	13	12	11	10	9	8	PAPD							
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	PAPD							
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

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	Reserved							
Type/Reset																
	23	22	21	20	19	18	17	16	Reserved							
Type/Reset																
	15	14	13	12	11	10	9	8	PAOD							
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	PAOD							
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[15:0]	PAODn	GPIO Port A pin n Open Drain Selection Control Bits (n = 0 ~ 15) 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)

Port A Output Current Drive Selection Register – PADRVR

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

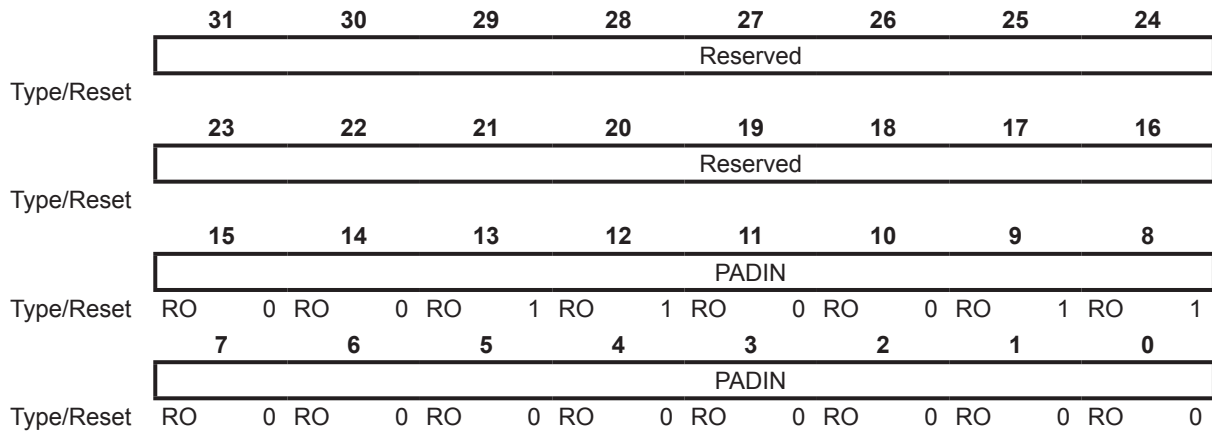
Bits	Field	Descriptions
[31:16]	PALKEY	<p>GPIO Port A Lock Key</p> <p>0x5FA0: Port A Lock function is enable Others: Port A Lock function is disable</p> <p>To lock the Port A function, a value 0x5FA0 should be written into the PALKEY field in this register. To execute a successful write operation on 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 field which is used to configure the alternative function of the associated GPIO pin will also be locked. Note that the PALOCKR 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_3300



Bits	Field	Descriptions
[15:0]	PADINn	GPIO Port A pin n Data Input Bits (n = 0 ~ 15) 0: The input data of pin is 0 1: The input data of 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	Reserved							
Type/Reset																
	23	22	21	20	19	18	17	16	Reserved							
Type/Reset																
	15	14	13	12	11	10	9	8	PADOUT							
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	PADOUT							
Type/Reset	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
	23	22	21	20	19	18	17	16	
	PARST								
Type/Reset	WO	0	WO	0	WO	0	WO	0	WO
	15	14	13	12	11	10	9	8	
	PASET								
Type/Reset	WO	0	WO	0	WO	0	WO	0	WO
	7	6	5	4	3	2	1	0	
	PASET								
Type/Reset	WO	0	WO	0	WO	0	WO	0	WO

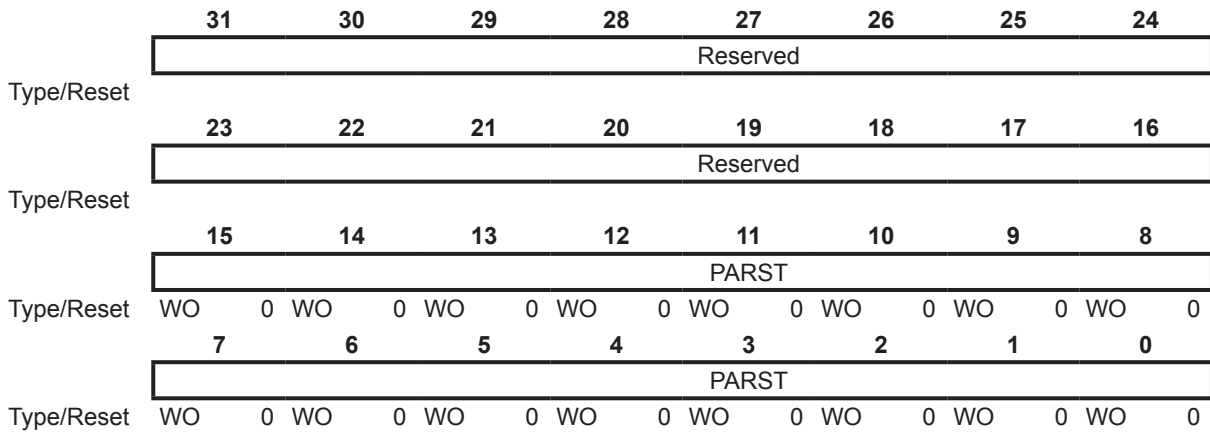
Bits	Field	Descriptions
[31:16]	PARSTn	GPIO Port A pin n Output Reset Control Bits (n = 0 ~ 15) 0: No effect on the PADOUTn bit 1: Reset the PADOUTn bit
[15:0]	PASETn	GPIO Port A pin n Output Set Control Bits (n = 0 ~ 15) 0: No effect on the PADOUTn bit 1: Set the PADOUTn bit Note that the function enabled 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



Bits	Field	Descriptions
[15:0]	PARSTn	GPIO Port A pin n Output Reset 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	Reserved							
Type/Reset																
	23	22	21	20	19	18	17	16	Reserved							
Type/Reset																
	15	14	13	12	11	10	9	8	PBDIR				Reserved	PBDIR		
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	PBDIR							
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[15:10]	PBDIRn	GPIO Port B pin n Direction Control Bits (n = 0 ~ 8, 10 ~ 15) 0: Pin n is input mode 1: Pin n is output mode
[8:0]		

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	Reserved							
Type/Reset																
	23	22	21	20	19	18	17	16	Reserved							
Type/Reset																
	15	14	13	12	11	10	9	8	PBINEN				Reserved	PBINEN		
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	PBINEN							
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[15:10] [8:0]	PBINENn	GPIO Port B pin n Input Enable Control Bits (n = 0 ~ 8, 10 ~ 15) 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 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	Reserved							
Type/Reset																
	23	22	21	20	19	18	17	16	Reserved							
Type/Reset																
	15	14	13	12	11	10	9	8	PBPU				Reserved	PBPU		
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	PBPU							
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[15:10]	PBPUn	GPIO Port B pin n Pull-Up Selection Control Bits (n = 0 ~ 8, 10 ~ 15)
[8:0]		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 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	Reserved							
Type/Reset																
	23	22	21	20	19	18	17	16	Reserved							
Type/Reset																
	15	14	13	12	11	10	9	8	PBPD				Reserved	PBPD		
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	PBPD							
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

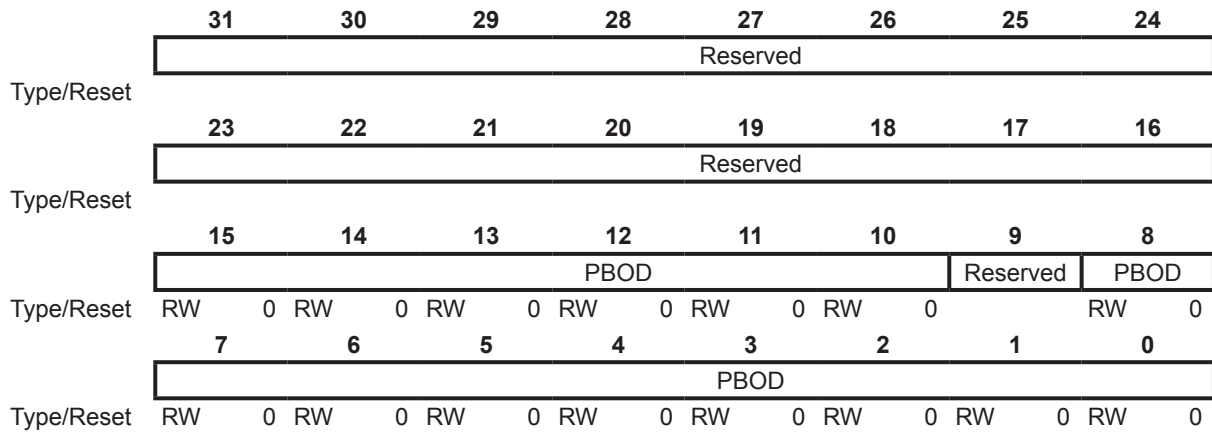
Bits	Field	Descriptions
[15:10]	PBPDn	GPIO Port B pin n Pull-Down Selection Control Bits (n = 0 ~ 8, 10 ~ 15) 0: Pin n pull-down function is disabled 1: Pin n pull-down function is enabled
[8:0]		
<p>Note: When the pull-up and pull-down functions are both enabled, the pull-up function will have the higher 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



Bits	Field	Descriptions
[15:10]	PBODn	GPIO Port B pin n Open Drain Selection Control Bits (n = 0 ~ 8, 10 ~ 15)
[8:0]		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)

Port B Output Current Drive Selection Register – PBDRVR

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		Reserved		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
[31:20], [17:0]	PBDVn[1:0]	GPIO Port B pin n Output Current Drive Selection Control Bits (n = 0 ~ 8, 10 ~ 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 – PBLOCKR

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						Reserved	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 Block Key</p> <p>0x5FA0: Port Block function is enable Others: Port B Lock function is disable</p> <p>To lock the Port B function, a value 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 PBLOCKR 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>
[15:10] [8:0]	PBLOCKn	<p>GPIO Port B pin n Lock Control Bits (n = 0 ~ 8, 10 ~ 15)</p> <p>0: Port B pin n is not locked 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, PBPU_n, PBPD_n and PBOD_n setting in the related GPIO registers. Additionally, the GPBCFGHR or GPBCFGLR field which is used to configure the alternative function of the associated GPIO pin will also be locked. Note that the PBLOCKR 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	Reserved											
Type/Reset																				
	23	22	21	20	19	18	17	16	Reserved											
Type/Reset																				
	15	14	13	12	11	10	9	8	PBDIN								Reserved	PBDIN		
Type/Reset	RO	0	RO	0	RO	0	RO	0	RO	0	RO	0	RO	0	RO	0	RO	0	RO	0
	7	6	5	4	3	2	1	0	PBDIN											
Type/Reset	RO	0	RO	0	RO	0	RO	0	RO	0	RO	0	RO	0	RO	0	RO	0		

Bits	Field	Descriptions
[15:10]	PBDINn	GPIO Port B pin n Data Input Bits (n = 0 ~ 8, 10 ~ 15)
[8:0]		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						Reserved	PBDOUT		
	RW	0	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0		
Type/Reset	PBDOUT									
	RW	0	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[15:10]	PBDOUTn	GPIO Port B pin n Data Output Bits (n = 0 ~ 8, 10 ~ 15) 0: Data to be output on pin n is 0 1: Data to be output on pin n is 1
[8:0]		

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						Reserved	
Type/Reset	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
	15	14	13	12	11	10	9	8
	PBSET						Reserved	PBSET
Type/Reset	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

Bits	Field	Descriptions
[31:26]	PBRSTn	GPIO Port B pin n Output Reset Control Bits (n = 0 ~ 8, 10 ~ 15) 0: No effect on the PBDOUTn bit 1: Reset the PBDOUTn bit
[23:16]		
[15:10]	PBSETn	GPIO Port B pin n Output Set Control Bits (n = 0 ~ 8, 10 ~ 15) 0: No effect on the PBDOUTn bit 1: Set the PBDOUTn bit
[8:0]		

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	Reserved							
Type/Reset																
	23	22	21	20	19	18	17	16	Reserved							
Type/Reset																
	15	14	13	12	11	10	9	8	PBRST				Reserved	PBRST		
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	PBRST							
Type/Reset	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0

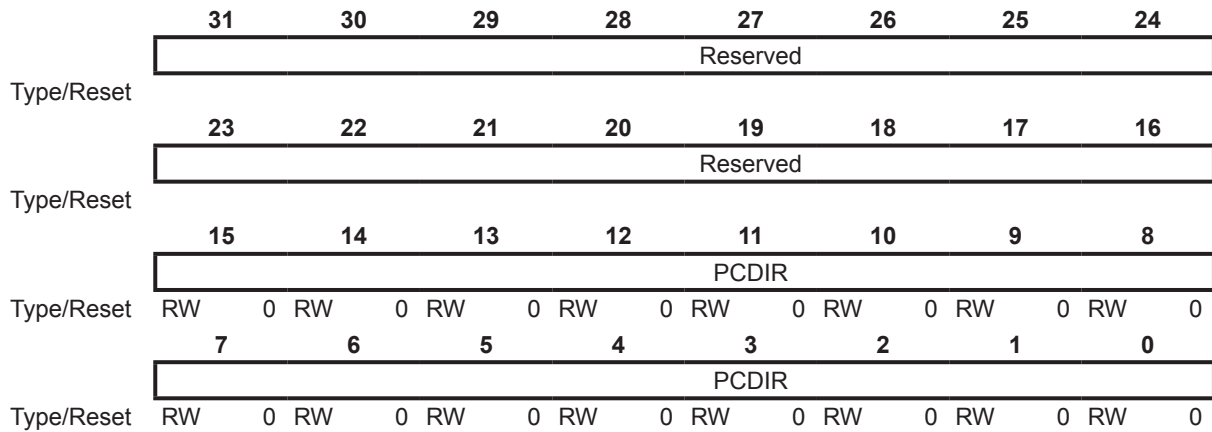
Bits	Field	Descriptions
[15:10]	PBRSTn	GPIO Port B pin n Output Reset Bits (n = 0 ~ 8, 10 ~ 15) 0: No effect on the PBDOUTn bit 1: Reset the PBDOUTn bit
[8:0]		

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



Bits	Field	Descriptions
[15:0]	PCDIRn	GPIO Port C pin n Direction Control Bits (n = 0 ~ 15) 0: Pin n is input mode 1: Pin n is 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	Reserved							
Type/Reset																
	23	22	21	20	19	18	17	16	Reserved							
Type/Reset																
	15	14	13	12	11	10	9	8	PCINEN							
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	PCINEN							
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[15:0]	PCINENn	GPIO Port C pin n Input Enable Control Bits (n = 0 ~ 15) 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 output will remain at a zero state.

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	Reserved							
Type/Reset																
	23	22	21	20	19	18	17	16	Reserved							
Type/Reset																
	15	14	13	12	11	10	9	8	PCPU							
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	PCPU							
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[15:0]	PCPU _n	GPIO Port C pin n Pull-Up Selection Control Bits (n = 0 ~ 15) 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 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	Reserved							
Type/Reset																
	23	22	21	20	19	18	17	16	Reserved							
Type/Reset																
	15	14	13	12	11	10	9	8	PCPD							
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	PCPD							
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

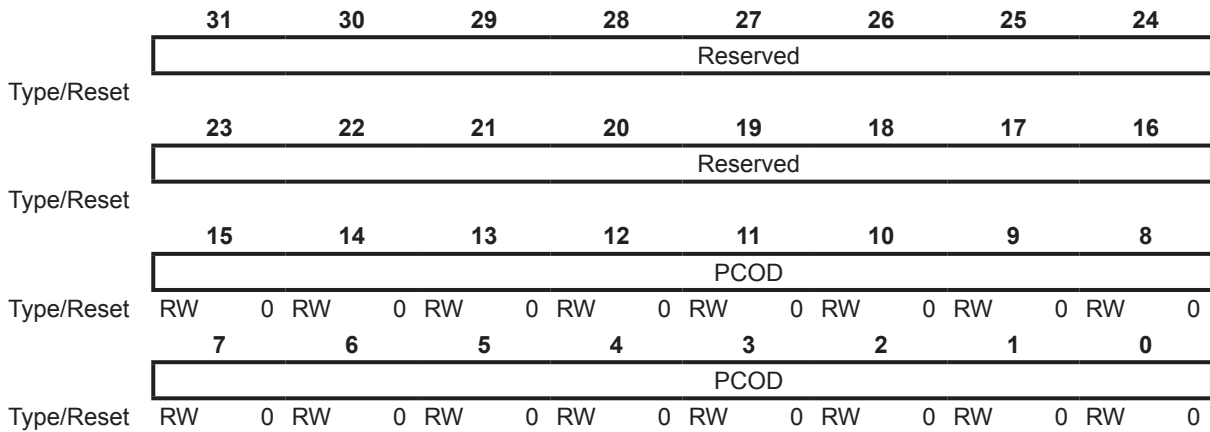
Bits	Field	Descriptions
[15:0]	PCPDn	<p>GPIO Port C pin n Pull-Down Selection Control Bits (n = 0 ~ 15)</p> <p>0: Pin n pull-down function is disabled</p> <p>1: Pin n pull-down function is enabled</p> <p>Note: When the pull-up and pull-down functions are both enabled, the pull-up function will have the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

Port 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



Bits	Field	Descriptions
[15:0]	PCODn	GPIO Port C pin n Open Drain Selection Control Bits (n = 0 ~ 15) 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)

Port C Output Current Drive Selection Register – PCDRVR

This register specifies the GPIO Port C output driving current.

Offset: 0x014

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
	PCDV15		PCDV14		PCDV13		PCDV12		
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	PCDV11		PCDV10		PCDV9		PCDV8		
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	PCDV7		PCDV6		PCDV5		PCDV4		
Type/Reset	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

Bits	Field	Descriptions
[31:0]	PCDVn[1:0]	<p>GPIO Port C pin n Output Current Drive Selection Control Bits (n = 0 ~ 15)</p> <p>00: 4 mA source/sink current 01: 8 mA source/sink current 10: 12 mA source/sink current 11: 16 mA source/sink current</p> <p>Because the PC13 ~ 15 are located at the Backup Domain. Therefore only the sink current capability can be set with PCDV[0] bit and don't care the PCDV[1] bit.</p> <p>x0: 4 mA sink current x1: 8 mA sink current</p> <p>For the source current of this pins are always limited at 2mA.</p>

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	
	PCLOCK								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	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	<p>GPIO Port C lock Key</p> <p>0x5FA0: Port C Lock function is enable Others: Port C Lock function is disable</p> <p>To lock the Port C function, a value 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.</p>
[15:0]	PCLOCKn	<p>GPIO Port C pin n Lock Control Bits (n = 0 ~ 15)</p> <p>0: Port C pin n is not locked 1: Port C pin n is locked</p> <p>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 and PCODn setting in the related GPIO registers. Additionally, the GPCCFGHR or GPCCFGLR field which is used to configure the alternative function of the associated GPIO pin will also be locked. Note that the PCLOCKR 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.</p>

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	PCDIN								
	RO	0	RO	0	RO	0	RO	0	RO
	7	6	5	4	3	2	1	0	
Type/Reset	PCDIN								
	RO	0	RO	0	RO	0	RO	0	RO

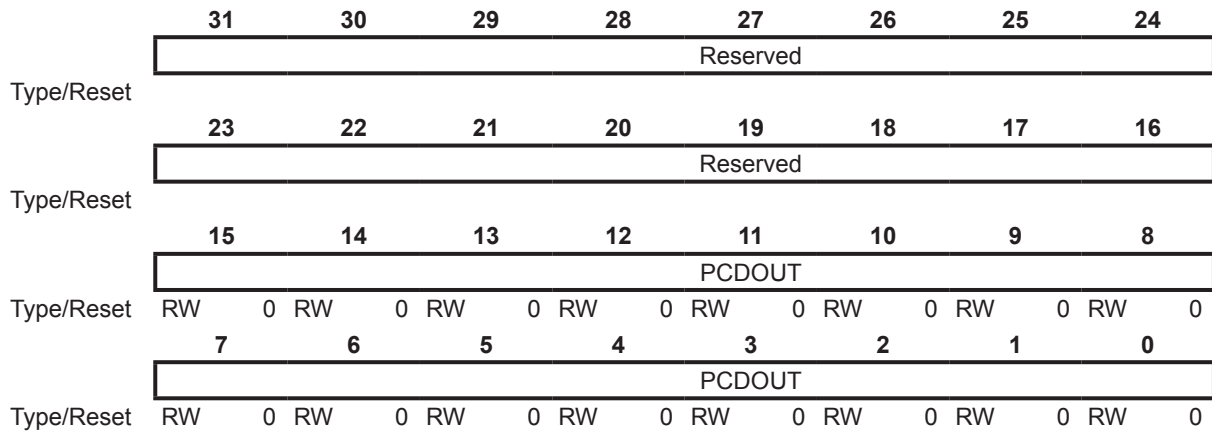
Bits	Field	Descriptions
[15:0]	PCDINn	GPIO Port C pin n Data Input Bits (n = 0 ~ 15) 0: The input data of corresponding pin is 0 1: The input data of corresponding pin is 1

Port C Output Data Register – PCDOUTR

This register specifies the GPIO Port C output data.

Offset: 0x020

Reset value: 0x0000_0000



Bits	Field	Descriptions
[15:0]	PCDOUTn	GPIO Port C 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 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	
	PCRST								
Type/Reset	WO	0	WO	0	WO	0	WO	0	WO
	23	22	21	20	19	18	17	16	
	PCRST								
Type/Reset	WO	0	WO	0	WO	0	WO	0	WO
	15	14	13	12	11	10	9	8	
	PCSET								
Type/Reset	WO	0	WO	0	WO	0	WO	0	WO
	7	6	5	4	3	2	1	0	
	PCSET								
Type/Reset	WO	0	WO	0	WO	0	WO	0	WO

Bits	Field	Descriptions
[31:16]	PCRSTn	GPIO Port C pin n Output Reset Control Bits (n = 0 ~ 15) 0: No effect on the PCDOUn bit 1: Reset the PCDOUn bit
[15:0]	PCSETn	GPIO Port C pin n Output Set Control Bits (n = 0 ~ 15) 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	Reserved							
Type/Reset																
	23	22	21	20	19	18	17	16	Reserved							
Type/Reset																
	15	14	13	12	11	10	9	8	PCRST							
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	PCRST							
Type/Reset	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0

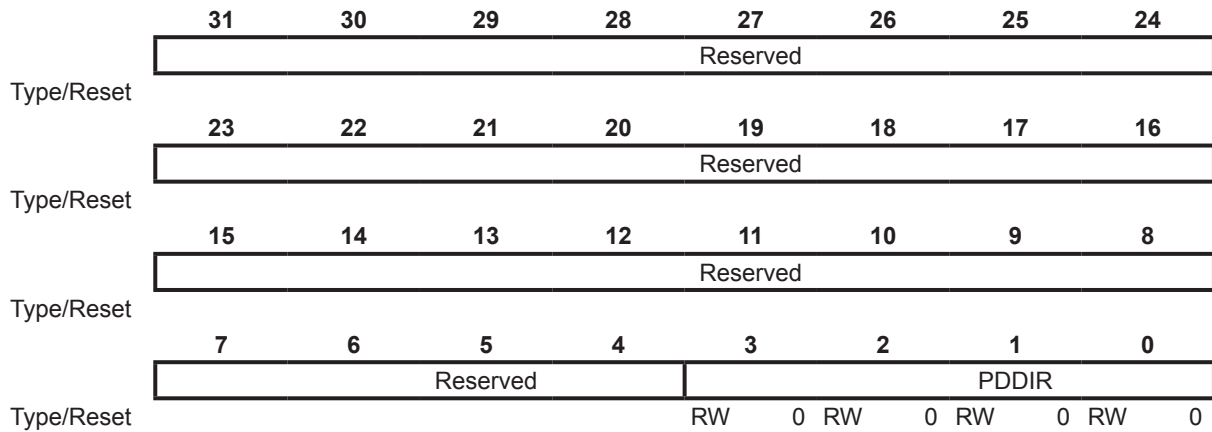
Bits	Field	Descriptions
[15:0]	PCRSTn	GPIO Port C pin n Output Reset Bits (n = 0 ~ 15) 0: No effect on the PCDOUn bit 1: Reset the PCDOUn bit

Port D Data Direction Control Register – PDDIRCR

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

Offset: 0x000

Reset value: 0x0000_0000



Bits	Field	Descriptions
[3:0]	PDDIRn	GPIO Port D pin n Direction Control Bits (n = 0 ~ 3) 0: Pin n is input mode 1: Pin n is output mode

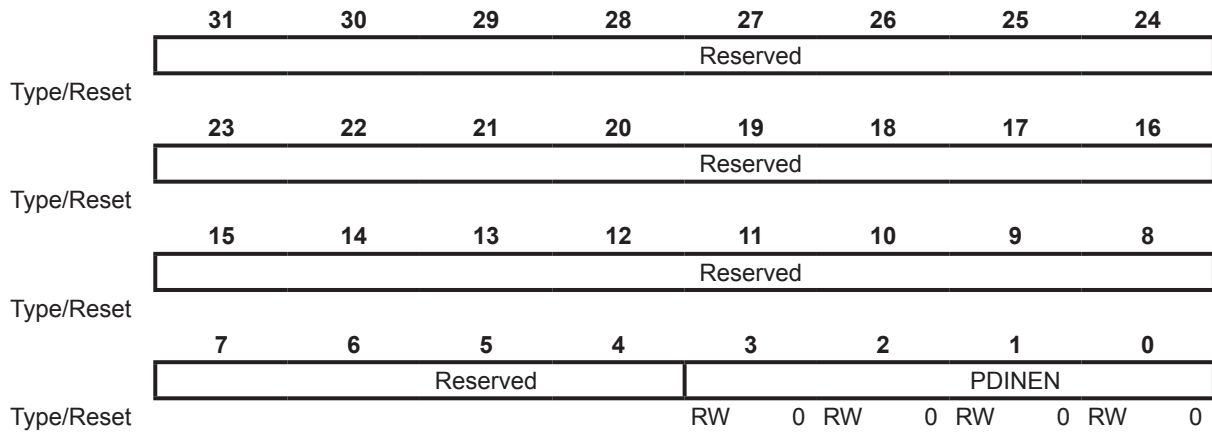
General Purpose I/O (GPIO)

Port D Input Function Enable Control Register – PDINER

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

Offset: 0x004

Reset value: 0x0000_0000



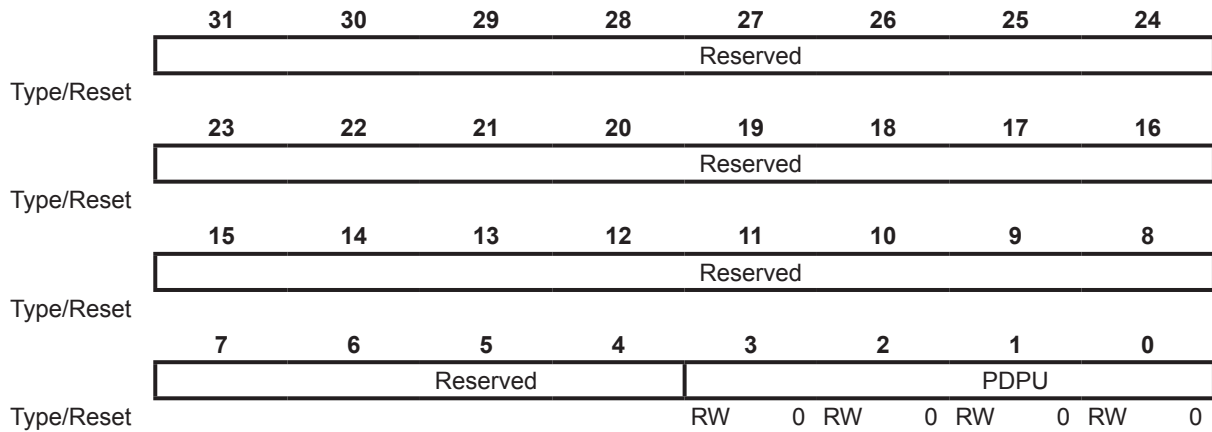
Bits	Field	Descriptions
[3:0]	PDINENn	GPIO Port D pin n Input Enable Control Bits (n = 0 ~ 3) 0: Pin n input function is disabled. 1: Pin n input function is enabled. When the pin n input function is disabled, the input Schmitt trigger will be turned off and the Schmitt trigger output will remain at a zero state.

Port D Pull-Up Selection Register – PDPUR

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

Offset: 0x008

Reset value: 0x0000_0000



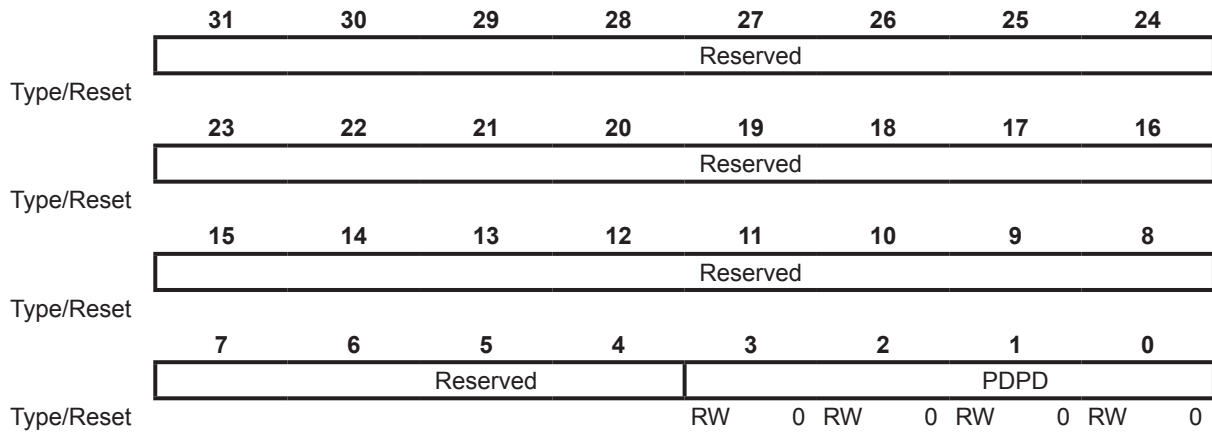
Bits	Field	Descriptions
[3:0]	PDPUn	GPIO Port D pin n Pull-Up Selection Control Bits (n = 0 ~ 3) 0: Pin n pull-up function is disabled 1: Pin n pull-up function is enabled Note: When the pull-up and pull-down functions are both enabled, the pull-up function will have the higher priority and therefore the pull-down function will be blocked and disabled.

Port D Pull-Down Selection Register – PDPDR

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

Offset: 0x00C

Reset value: 0x0000_0000



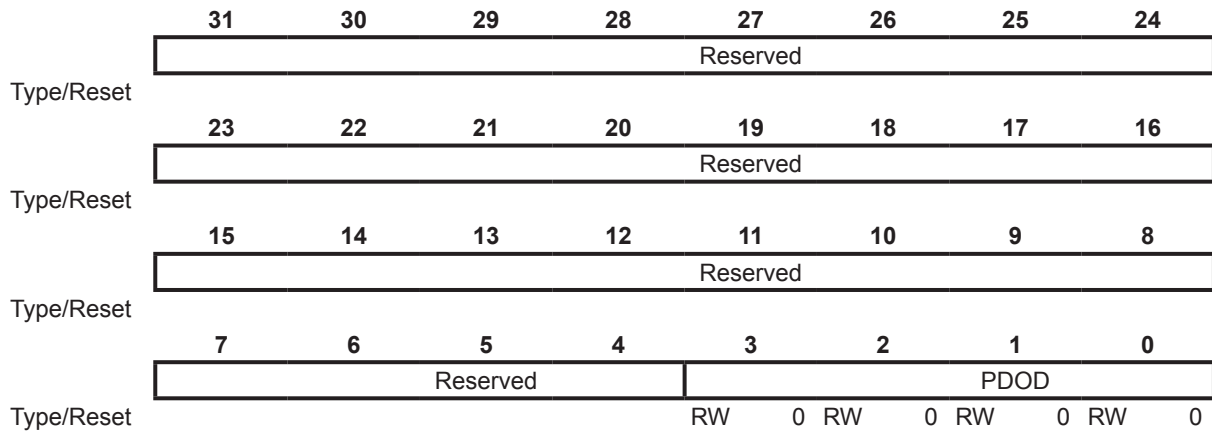
Bits	Field	Descriptions
[3:0]	PDPDn	<p>GPIO Port D pin n Pull-Down Selection Control Bits (n = 0 ~ 3)</p> <p>0: Pin n pull-down function is disabled</p> <p>1: Pin n pull-down function is enabled</p> <p>Note: When the pull-up and pull-down functions are both enabled, the pull-up function will have the higher priority and therefore the pull-down function will be blocked and disabled.</p>

Port D Open Drain Selection Register – PDODR

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

Offset: 0x010

Reset value: 0x0000_0000



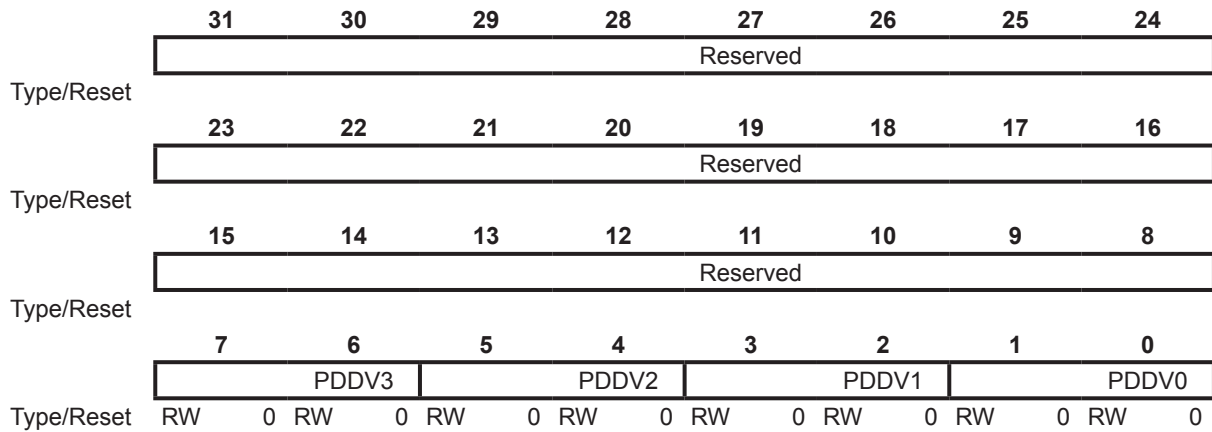
Bits	Field	Descriptions
[3:0]	PDODn	GPIO Port D pin n Open Drain Selection Control Bits (n = 0 ~ 3) 0: Pin n Open Drain output is disabled. (The output type is CMOS output) 1: Pin n Open Drain output is enabled. (The output type is open-drain output)

Port D Output Current Drive Selection Register – PDDRVR

This register specifies the GPIO Port D output driving current.

Offset: 0x014

Reset value: 0x0000_0000



Bits	Field	Descriptions
[7:0]	PDDVn[1:0]	GPIO Port D pin n Output Current Drive Selection Control Bits (n = 0 ~ 3) 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 D Lock Register – PDLOCKR

This register specifies the GPIO Port D lock configuration.

Offset: 0x018

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
	PDLKEY								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	PDLKEY								
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				PDLOCK				
Type/Reset					RW	0	RW	0	RW

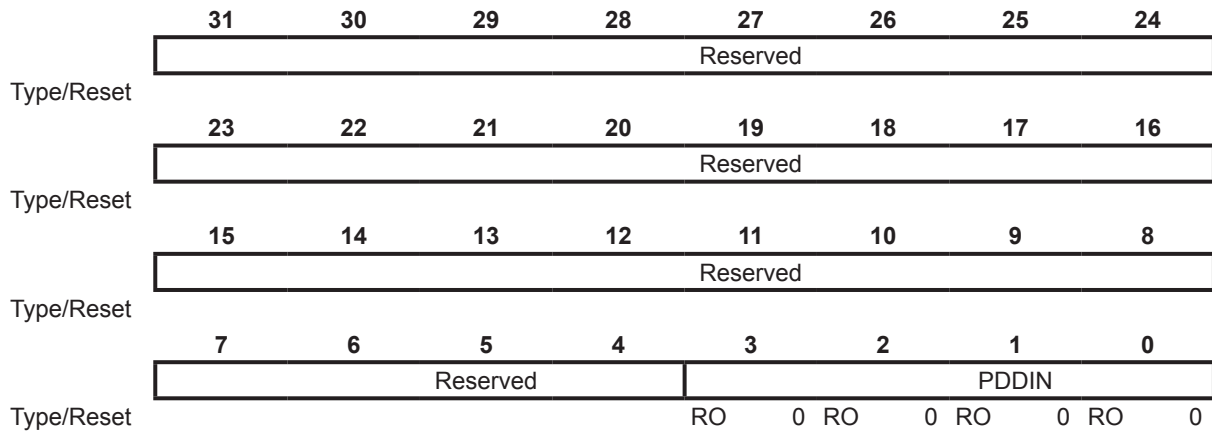
Bits	Field	Descriptions
[31:16]	PDLKEY	<p>GPIO Port D Lock Key</p> <p>0x5FA0: Port D Lock function is enable Others: Port D Lock function is disable</p> <p>To lock the Port D function, a value 0x5FA0 should be written into the PDLKEY field in this register. To execute a successful write operation on this lock register, the value written into the PDLKEY field must be 0x5FA0. If the value written into this field is not equal to 0x5FA0, any write operations on the PDLOCKR register will be aborted. The result of a read operation on the PDLKEY field returns the GPIO Port D Lock Status which indicates whether the GPIO Port D is locked or not. If the read value of the PDLKEY field is 0, this indicates that the GPIO Port D Lock function is disabled. Otherwise, it indicates that the GPIO Port D Lock function is enabled as the read value is equal to 1.</p>
[3:0]	PDLOCKn	<p>GPIO Port D pin n Lock Control Bits (n = 0 ~ 3)</p> <p>0: Port D pin n is not locked 1: Port D pin n is locked</p> <p>The PDLOCKn bits are used to lock the configurations of corresponding GPIO Pins when the correct Lock Key is applied to the PDLKEY field. The locked configurations including PDDIRn, PDINENn, PDPUn, PDPDn and PDODn setting in the related GPIO registers. Additionally, the GPDCFGHR or GPDCFGLR field which is used to configure the alternative function of the associated GPIO pin will also be locked. Note that the PDLOCKR can only be written once which means that PDLKEY and PDLOCKn (lock control bit) should be written together and can not be changed until a system reset or GPIO Port D reset occurs.</p>

Port D Data Input Register – PDDINR

This register specifies the GPIO Port D input data.

Offset: 0x01C

Reset value: 0x0000_0000



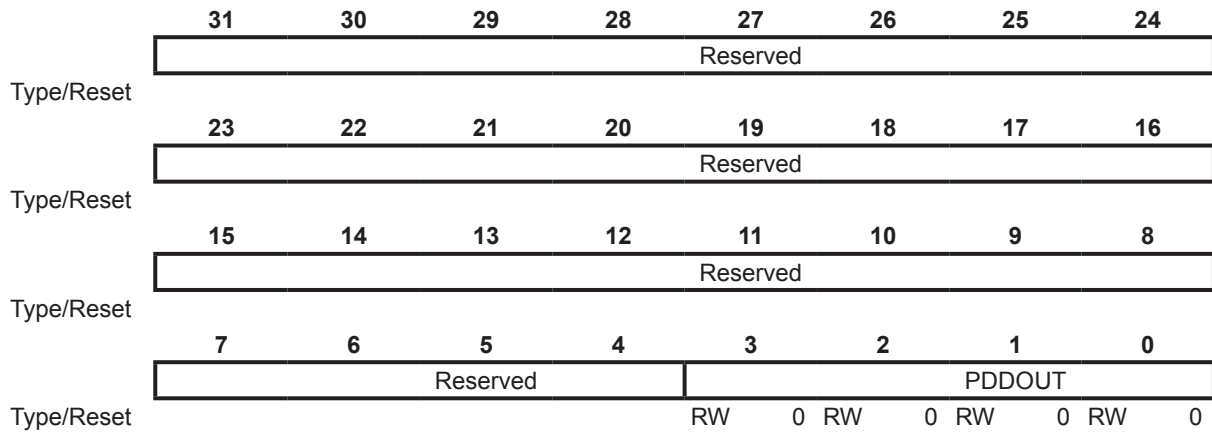
Bits	Field	Descriptions
[3:0]	PDDINn	GPIO Port D pin n Data Input Bits (n = 0 ~ 3) 0: The input data of corresponding pin is 0 1: The input data of corresponding pin is 1

Port D Output Data Register – PDDOUTR

This register specifies the GPIO Port D output data.

Offset: 0x020

Reset value: 0x0000_0000



Bits	Field	Descriptions
[3:0]	PDDOUTn	GPIO Port D pin n Data Output Bits (n = 0 ~ 3) 0: Data to be output on pin n is 0 1: Data to be output on pin n is 1

Port D Output Set/Reset Control Register – PDSRR

This register is used to set or reset the corresponding bit of the GPIO Port D 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				PDRST				
					WO	0	WO	0	WO
					0		WO	0	WO
									0
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				PDSET				
					WO	0	WO	0	WO
					0		WO	0	WO
									0

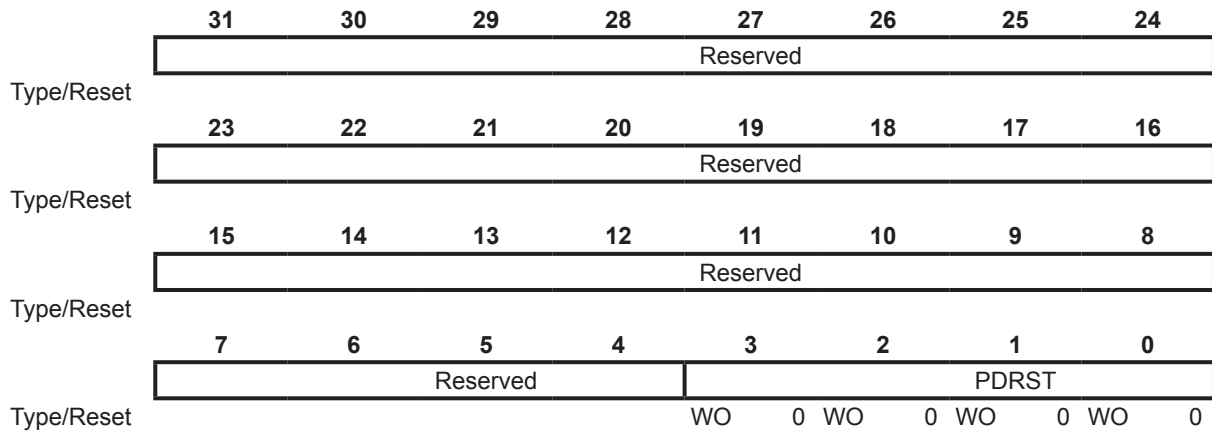
Bits	Field	Descriptions
[19:16]	PDRSTn	GPIO Port D pin n Output Reset Control Bits (n = 0 ~ 3) 0: No effect on the PDDOUTn bit 1: Reset the PDDOUTn bit
[3:0]	PDSETn	GPIO Port D pin n Output Set Control Bits (n = 0 ~ 3) 0: No effect on the PDDOUTn bit 1: Set the PDDOUTn bit Note that the function enabled by the PDSETn bit has the higher priority if both the PDSETn and PDRSTn bits are set at the same time.

Port D Output Reset Register – PDRR

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

Offset: 0x028

Reset value: 0x0000_0000



Bits	Field	Descriptions
[3:0]	PDRSTn	GPIO Port D pin n Output Reset Bits (n = 0 ~ 3) 0: No effect on the PDDOUTn bit 1: Reset the PDDOUTn 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 IO pin can be configured to have up to sixteen different functions such as GPIO or IP functions by setting the GPxCFGxLR or GPxCFGxHR 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 IO 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.

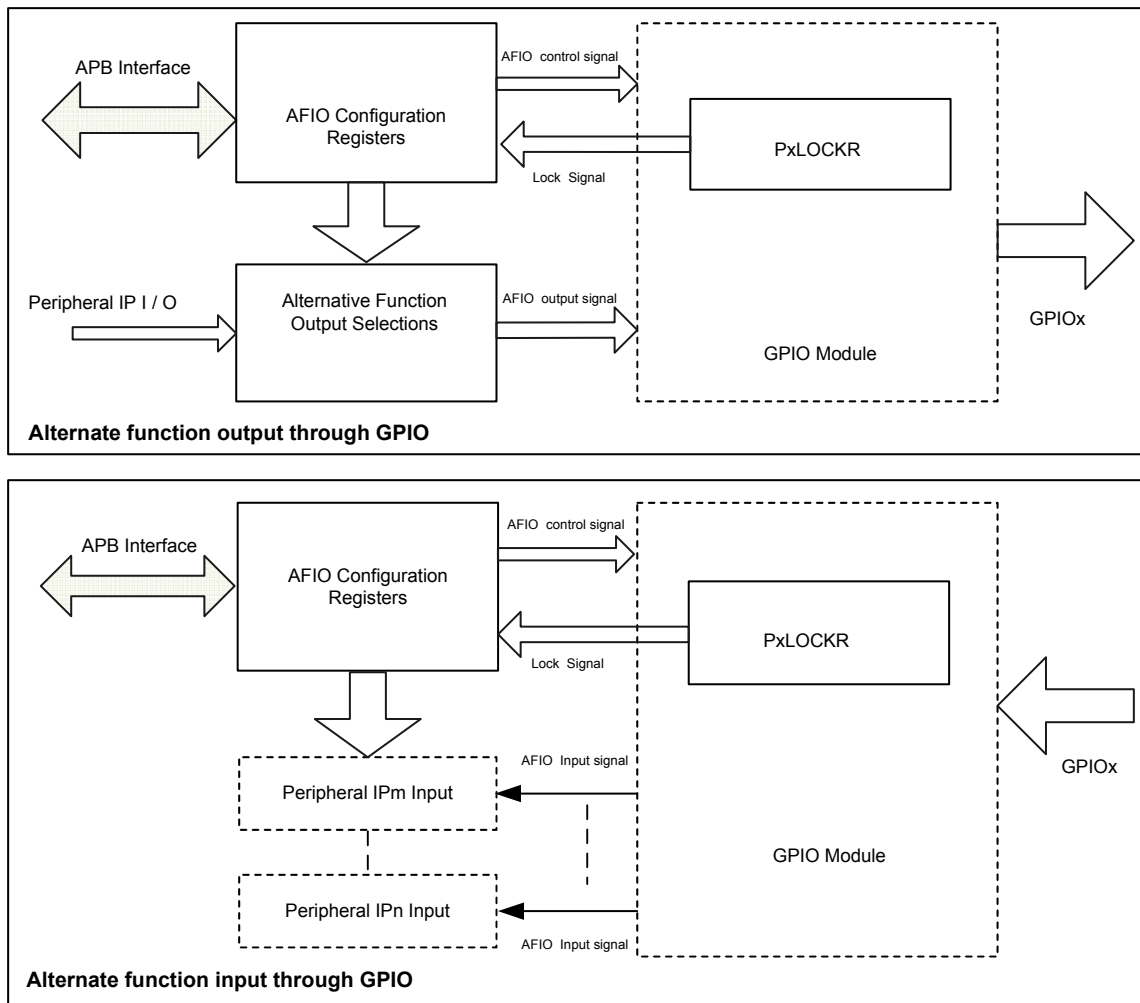


Figure 22. AFIO Block Diagram

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 ~ D pins are available in all package types, please refer to the pin assignment section for detailed pin information. The setting of the EXTIInPIN [3:0] field is invalid when the corresponding pin is not available.

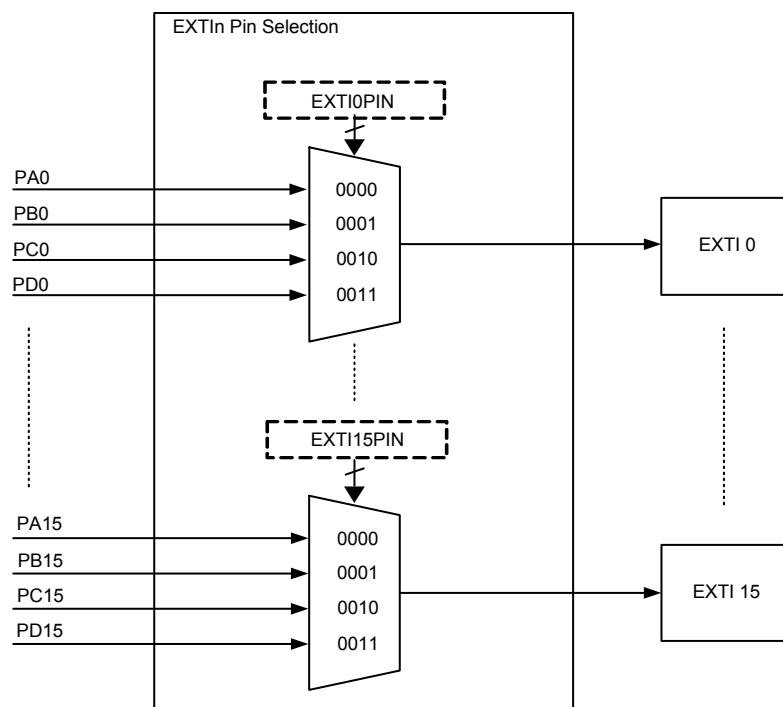


Figure 23. EXTI Channel Input Selection

Alternate Function

Up to sixteen alternative functions can be chosen for each I/O pad by setting the PxCFGn [3:0] field in the GPxCFGLR or GPxCFGHR (n = 0~15, x = A~ D) registers. If the pin is selected as an unavailable item which is noted as a “N/A” item in the “Alternate Function Mapping” table in the device datasheet, this pin will be defined as the default alternate function. Please refer to the “Alternate Function Mapping” table in the device datasheet for the 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 22. 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	ADC	CMP	MCTM /GPTM	SPI	USART /UART	I ² C	SCI	EBI	I ² S	N/A	N/A	SCTM	N/A	System Other

Lock Mechanism

The device also offers a lock function to lock the AFIO configuration using the GPIO lock register, PxLOCKR, 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 value.

Table 23. 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 AFIO Configuration Register 0	0x0000_0000
GPACFGHR	0x024	GPIO Port A AFIO Configuration Register 1	0x0000_0000
GPBCFGLR	0x028	GPIO Port B AFIO Configuration Register 0	0x0000_0000
GPBCFGHR	0x02C	GPIO Port B AFIO Configuration Register 1	0x0000_0000
GPCCFGLR	0x030	GPIO Port C AFIO Configuration Register 0	0x0000_0000
GPCCFGHR	0x034	GPIO Port C AFIO Configuration Register 1	0x0000_0000
GPDCFGLR	0x038	GPIO Port D AFIO Configuration Register 0	0x0000_0000
GPDCFGHR	0x03C	GPIO Port D AFIO Configuration Register 1	0x0000_0000

Register Descriptions

EXTI Source Selection Register 0 – ESSR0

This register specifies the IO 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]	<p>EXTIn Pin Selection (n = 0 ~ 7)</p> <p>0000: PA Bit n is selected as EXTIn source signal 0001: PB Bit n is selected as EXTIn source signal 0010: PC Bit n is selected as EXTIn source signal 0011: PD Bit n is selected as EXTIn source signal 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 EXTIInPIN [3:0] field setting is invalid when the corresponding pin is not available.</p>

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				EXTI14PIN				
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	EXTI13PIN				EXTI12PIN				
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	EXTI11PIN				EXTI10PIN				
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	EXTI9PIN				EXTI8PIN				
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW

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>0011: PD 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 EXTIInPIN [3:0] field setting is invalid when the corresponding pin is not available.</p>

GPIO x Configuration Low Register – GPxCFGLR, x = A, B, C, D

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

Offset: 0x020, 0x028, 0x030, 0x038

Reset value: 0x0000_0000

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

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

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

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

Offset: 0x024, 0x02C, 0x034, 0x03C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
	PxCFG15				PxCFG14				
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	PxCFG13				PxCFG12				
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	PxCFG11				PxCFG10				
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	PxCFG9				PxCFG8				
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW

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

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 interrupt 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 reached zero. The accompanying table lists the 16 system exceptions types and a variety of peripheral interrupts.

Table 24. 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 decremented 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	CMP	Configurable ⁽²⁾	0x05C	Comparator global interrupt
8	24	ADC	Configurable ⁽²⁾	0x060	ADC global interrupt
9	25	Reserved	—	0x064	—
10	26	MCTM	Configurable ⁽²⁾	0x068	MCTM global interrupt
11	27	GPTM1	Configurable ⁽²⁾	0x06C	GPTM1 global interrupt
12	28	GPTM0	Configurable ⁽²⁾	0x070	GPTM0 global interrupt
13	29	SCTM0	Configurable ⁽²⁾	0x074	SCTM0 global interrupt
14	30	SCTM1	Configurable ⁽²⁾	0x078	SCTM1 global interrupt

Interrupt Number	Exception Number	Exception type	Priority	Vector Address	Description
15	31	Reserved	—	0x07C	—
16	32	Reserved	—	0x080	—
17	33	BFTM0	Configurable ⁽²⁾	0x084	BFTM0 global interrupt
18	34	BFTM1	Configurable ⁽²⁾	0x088	BFTM1 global interrupt
19	35	I ² C0	Configurable ⁽²⁾	0x08C	I ² C0 global interrupt
20	36	I ² C1	Configurable ⁽²⁾	0x090	I ² C1 global interrupt
21	37	SPI0	Configurable ⁽²⁾	0x094	SPI0 global interrupt
22	38	SPI1	Configurable ⁽²⁾	0x098	SPI1 global interrupt
23	39	USART0	Configurable ⁽²⁾	0x09C	USART0 global interrupt
24	40	USART1	Configurable ⁽²⁾	0x0A0	USART1 global interrupt
25	41	UART0	Configurable ⁽²⁾	0x0A4	UART0 global interrupt
26	42	UART1	Configurable ⁽²⁾	0x0A8	UART1 global interrupt
27	43	SCI0 ~ 1	Configurable ⁽²⁾	0x0AC	SCI0 & SCI1 global interrupt
28	44	I ² S	Configurable ⁽²⁾	0x0B0	I ² S 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
- 16 programmable priority levels (4 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 decrements to 0
 - SysTick clock source derived from the HCLK clock divided by 8

Function Descriptions

SysTick Calibration

The SysTick Calibration Value Register (SCALIB) is provided by the NVIC to give a reference time base of 1ms for the RTOS tick timer or other purpose. The TENMS field in the SCALIB register has a fixed value of 6000 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 6 MHz (48 MHz divide by 8).

Register Map

The following table shows the NVIC registers and reset values.

Table 25. 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_1770
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 information of the above detail register descriptions, please 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 a wake-up event or interrupt requests independently. In 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 wake-up event mode, the wake-up 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 wake-up event occurs and the corresponding EXTI wake-up enable bit is set. Each EXTI line can also be masked independently.

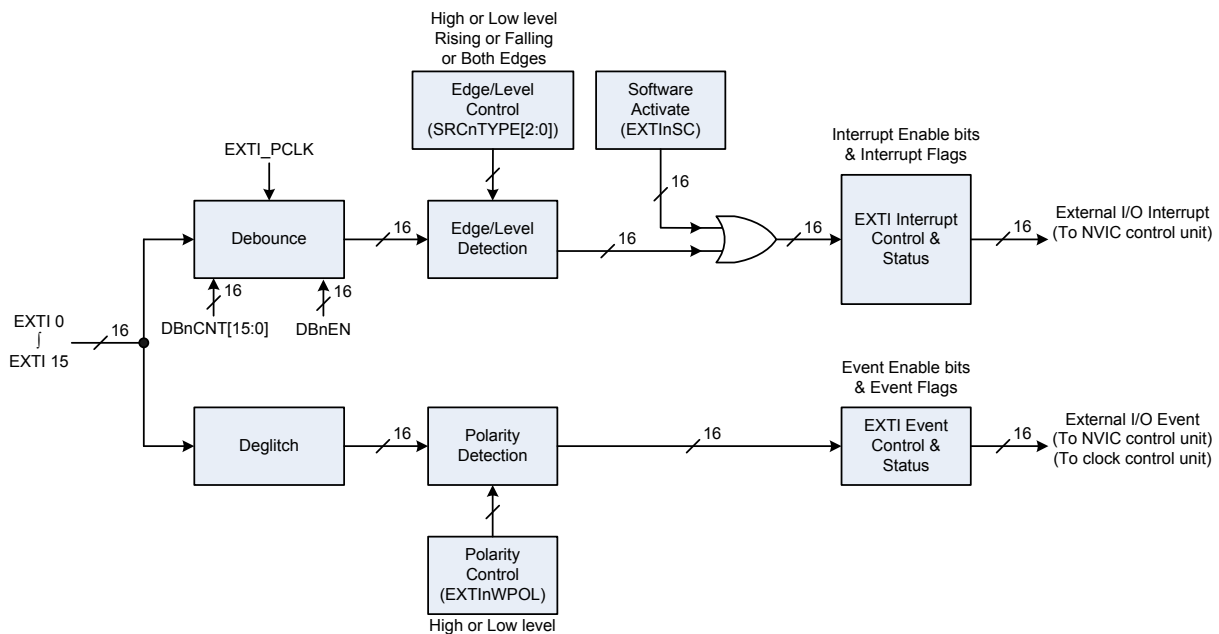


Figure 24. EXTI Block Diagram

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

Function Descriptions

Wakeup Event Management

In order to wakeup 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, Comparator, 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 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

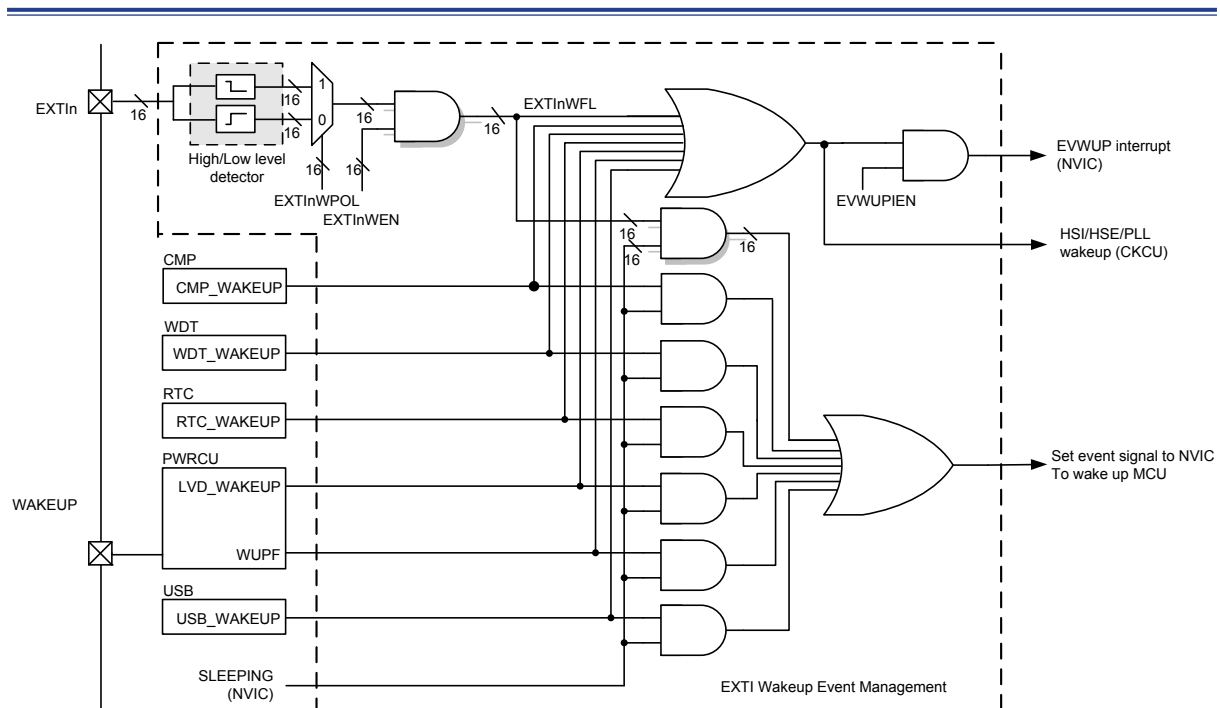


Figure 25. EXTI Wake-up Event Management

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 ~ 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 ~ 15) to enable the corresponding pin de-bounce function and configure the DBnCNT field in the EXTICFGRn so as to select an appropriate de-bounce time for specific applications. The interrupt signal will however be delayed due to the de-bounce 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 wake-up 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.

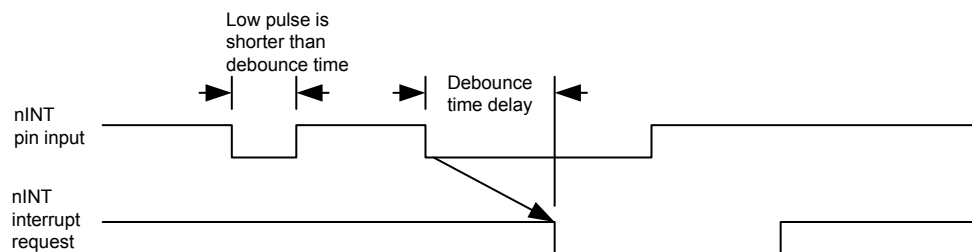


Figure 26. EXTI Interrupt Debounce Function

Register Map

The following table shows the EXTI registers and reset values.

Table 26. 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 Configuration Register n – 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								
	7	6	5	4	3	2	1	0	
Type/Reset	DBnCNT								

Bits	Field	Descriptions																								
[31]	DBnEN	EXTIn De-bounce Circuit Enable Bit (n = 0 ~ 15) 0: De-bounce circuit is disabled 1: De-bounce circuit is enabled																								
[30:28]	SRCnTYPE	EXTIn Interrupt Source Trigger Type (n = 0 ~ 15)																								
		<table border="1"> <thead> <tr> <th colspan="3">SRCnTYPE [2:0]</th> <th>Interrupt Source Type</th> </tr> </thead> <tbody> <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> </tbody> </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 De-bounce Counter (n = 0 ~ 15) The de-bounce time is calculated with DBnCNT x 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	Reserved							
Type/Reset																
	23	22	21	20	19	18	17	16	Reserved							
Type/Reset																
	15	14	13	12	11	10	9	8	EXTI15EN	EXTI14EN	EXTI13EN	EXTI12EN	EXTI11EN	EXTI10EN	EXTI9EN	EXTI8EN
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	EXTI7EN	EXTI6EN	EXTI5EN	EXTI4EN	EXTI3EN	EXTI2EN	EXTI1EN	EXTI0EN
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

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	Reserved							
Type/Reset																
	23	22	21	20	19	18	17	16	Reserved							
Type/Reset																
	15	14	13	12	11	10	9	8	EXTI15EDF	EXTI14EDF	EXTI13EDF	EXTI12EDF	EXTI11EDF	EXTI10EDF	EXTI9EDF	EXTI8EDF
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	EXTI7EDF	EXTI6EDF	EXTI5EDF	EXTI4EDF	EXTI3EDF	EXTI2EDF	EXTI1EDF	EXTI0EDF
Type/Reset	WC	0	WC	0	WC	0	WC	0	WC	0	WC	0	WC	0	WC	0

Bits	Field	Descriptions
[15:0]	EXTInEDF	EXTIn Both Edge Detection Flag (n = 0 ~ 15) 0: No edge is detected 1: Positive or negative edge is detected 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.

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
	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

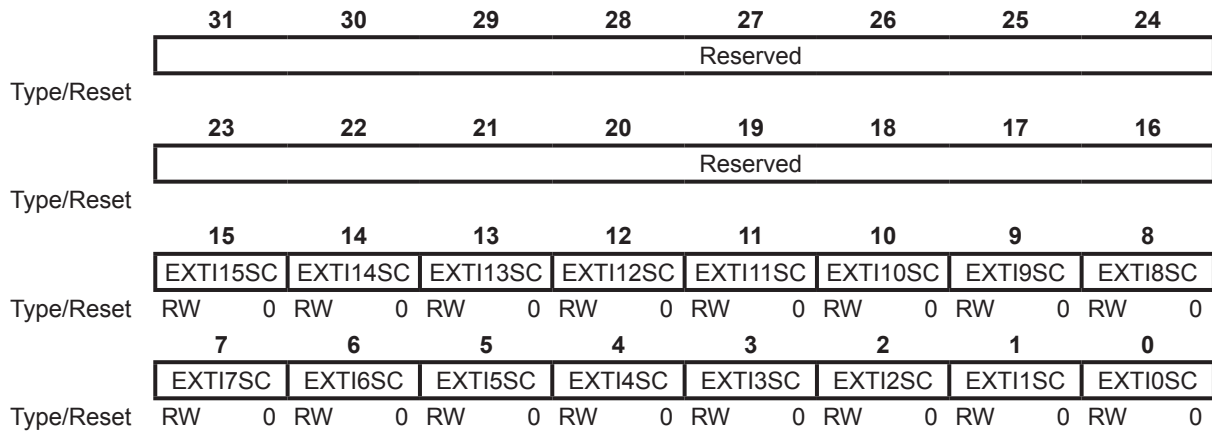
Bits	Field	Descriptions
[15:0]	EXTInEDS	EXTIn Both Edge Detection Status (n = 0 ~ 15) 0: Negative edge is detected 1: Positive edge is detected Software should write 1 to clear it.

EXTI Interrupt Software Set Command Register – EXTISSCR

This register is used to activate the EXTI interrupt.

Offset: 0x04C

Reset value: 0x0000_0000



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	EXTI11WEN	EXTI10WEN	EXTI9WEN	EXTI8WEN								
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								
	EXTI7WEN	EXTI6WEN	EXTI5WEN	EXTI4WEN	EXTI3WEN	EXTI2WEN	EXTI1WEN	EXTI0WEN								
Type/Reset	RW	0	RW	0	RW	0	RW	0	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	Reserved							
Type/Reset																
	23	22	21	20	19	18	17	16	Reserved							
Type/Reset																
	15	14	13	12	11	10	9	8	EXTI15WPOL	EXTI14WPOL	EXTI13WPOL	EXTI12WPOL	EXTI11WPOL	EXTI10WPOL	EXTI9WPOL	EXTI8WPOL
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	EXTI7WPOL	EXTI6WPOL	EXTI5WPOL	EXTI4WPOL	EXTI3WPOL	EXTI2WPOL	EXTI1WPOL	EXTI0WPOL
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

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 wake flag register.

Offset: 0x058

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	EXTI15WFL	EXTI14WFL	EXTI13WFL	EXTI12WFL	EXTI11WFL	EXTI10WFL	EXTI9WFL	EXTI8WFL
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	EXTI7WFL	EXTI6WFL	EXTI5WFL	EXTI4WFL	EXTI3WFL	EXTI2WFL	EXTI1WFL	EXTI0WFL
Type/Reset	WC	0	WC	0	WC	0	WC	0	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 waken up by EXTIn Software should write 1 to clear it.

12 Analog to Digital Converter (ADC)

Introduction

A 12-bit multi-channel Analog to Digital Converter is integrated in the device. There are a total of 14 multiplexed channels including 12 external channels on which the external analog signal can be supplied and 2 internal channels. If the input voltage is required to remain within a specific threshold window, the Analog Watchdog function will monitor and detect the signal. An interrupt will then be generated to inform that the input voltage is higher or lower than the set thresholds. There are three conversion modes to convert an analog signal to digital data. The A/D conversion can be operated in one shot, continuous and discontinuous conversion mode. A 16-bit data register is provided to store the data after conversion.

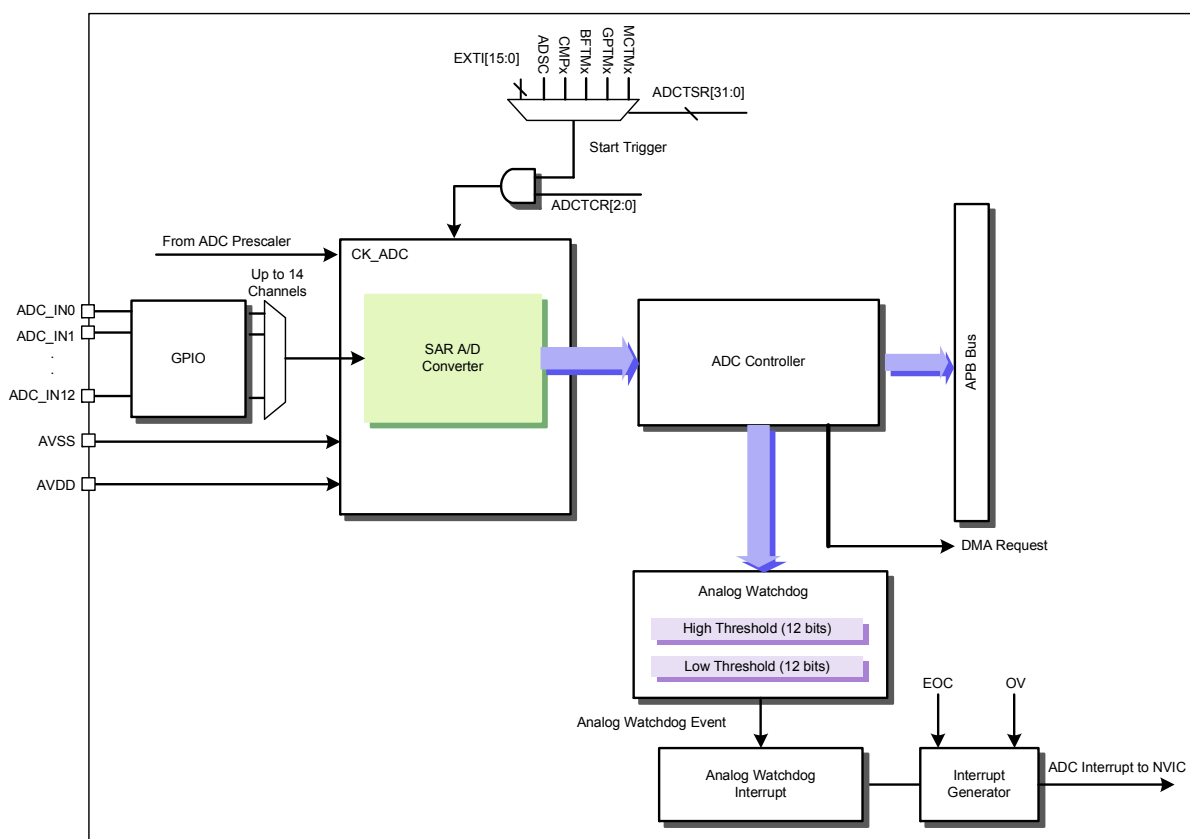


Figure 27. ADC Block Diagram

Features

- 12-bit SAR ADC engine
- Up to 1 MSPS conversion rate
- 12 external analog input channels
- 2 internal analog input channels for reference voltage detection
- Programmable sampling time for conversion channel
- Up to 8 programmable conversion channel sequence and dedicated data registers for conversion result
- Three conversion mode
 - One shot conversion mode
 - Continuous conversion mode
 - Discontinuous conversion mode.
- Analog watchdog for predefined voltage range monitor
 - Lower/upper threshold register
 - Interrupt generation
- Various trigger start source for conversion modes
 - Software trigger
 - EXTI - external interrupt input pin
 - GPTM0 / GPTM1 trigger
 - MCTM trigger
 - BFTM0 / BFTM1 trigger
 - CMP0 / CMP1 trigger
- Multiple generated interrupts
 - End of single conversion
 - End of subgroup conversion
 - End of cycle conversion
 - Analog Watchdog
 - Data register overwriting
- PDMA request on end of conversion occurred

Function Descriptions

ADC Clock Setup

The ADC clock, CK_ADC is provided by the Clock Controller which is synchronous and divided by with the AHB clock known as HCLK. Refer to the Clock Control Unit chapter for more details. Notes that the ADC requires at least two ADC clock cycles to switch between power-on and power-off conditions (ADEN bit = '0').

Channel Selection

The A/D converter supports 12 multiplexed channels and organizes the conversion results into a specific group. A conversion group can organize a sequence which can be implemented on the channels arranged in a specific conversion sequence length from 1 to 8. For example, conversion can be carried out with the following channel sequence: CH2, CH4, CH7, CH5, CH6, CH3, CH1 and CH0 one after another.

A group is composed of up to 8 conversions. The selected channels of the group conversion can be specified in the ADCLST0~ADCLST1 registers. The total conversion sequence length is setup using the ADSEQ[2:0] bits in the ADCCONV register.

Modifying the ADCCONV or ADCHCONV register during a conversion process will reset the current conversion, after which a new start pulse is required to restart a new conversion.

Conversion Mode

The A/D has three operating conversion modes. The conversion modes are One Shot Conversion Mode, Continuous Conversion Mode, and Discontinuous Conversion mode. Details are provided later.

One Shot Conversion Mode

In the One Shot Conversion mode, the ADC will perform conversion cycles on the channels specified in the A/D conversion list registers ADCLSTn with a specific sequence when an A/D converter trigger event occurs. When the A/D conversion mode field ADMODE [1:0] in the ADCCONV register is set to 0x0, the A/D converter will operate in the One Shot Conversion Mode. This mode can be started by a software trigger, a comparator output transition event, an external EXTI event or a TM event determined by the Trigger Control Register ADCTCR and the Trigger Source Register ADCTSR.

After Conversion:

- The converted data will be stored in the 16-bit ADCDRy (y = 0~7) registers.
- The ADC single sample end of conversion event raw status flag, ADIRAWS, in the ADCIRAW register will be set when the single sample conversion is finished.
- An interrupt will be generated after a single sample end of conversion if the ADIES bit in the ADCIER register is enabled.
- An interrupt will be generated after a group cycle end of conversion if the ADIEC bit in the ADCIER register is enabled.

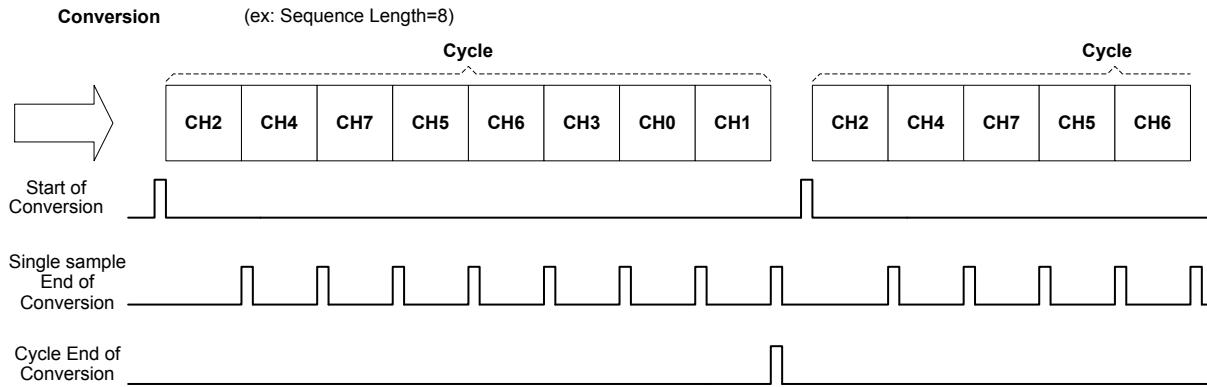


Figure 28. One Shot Conversion Mode

Continuous Conversion Mode

In the Continuous Conversion Mode, repeated conversion cycle will restart automatically without requiring additional A/D start trigger signals after a channel group conversion has completed. When the A/D conversion mode field `ADM0DE[1:0]` is set to `0x2`, the A/D converter will operate in the Continuous Conversion Mode which can be started by a software trigger, a comparator output transition event, an external `EXTI` event or a `TM` event determined by the Trigger Control Register `ADCTCR` and the Trigger Source Register `ADCTSR`.

After conversion:

- The converted data will be stored in the 16-bit `ADCDRy` ($y = 0\sim7$) registers.
- The ADC group cycle end of conversion event raw status flag, `ADIRAWC`, in the `ADCIRAW` register will be set when the conversion cycle is finished.
- An interrupt will be generated after a group cycle end of conversion if the `ADIEC` bit in the `ADCIER` register is enabled.

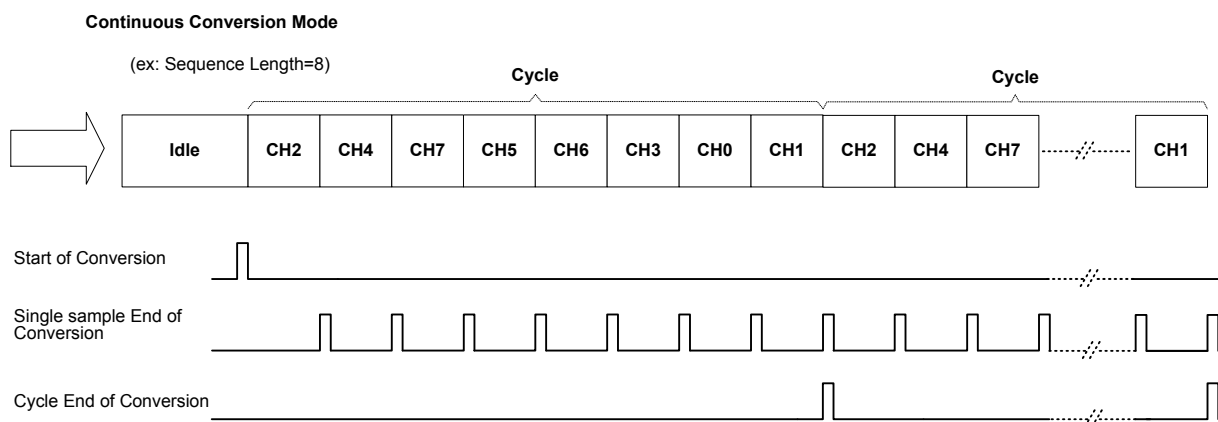


Figure 29. Continuous Conversion Mode

Discontinuous Conversion Mode

The A/D converter will operate in the Discontinuous Conversion Mode for channels group when the A/D conversion mode bit field ADMODE [1:0] in the ADCCONV register is set to 0x3. The group to be converted can have up to 8 channels and can be arranged in a specific sequence by configuring the ADCLSTn registers where n ranges from 0 to 1. This mode is provided to convert data for the group with a short sequence, named as the A/D conversion subgroup, each time a trigger event occurs. The subgroup length is defined by the ADSUBL [2:0] field in the ADCCONV register to specify the subgroup length. In the Discontinuous Conversion Mode the A/D converter can be started by a software trigger, a comparator output transition event, an external EXTI event or a TM event for the groups determined by the Trigger Control Register ADCTCR and the Trigger Source Register ADCTSR.

In the Discontinuous Conversion Mode, the A/D Converter will start to convert the next n conversions where the number n is the subgroup length defined by the ADSUBL field. When a trigger event occurs, the channels to be converted with a specific sequence are specified in the ADCLSTn registers. After n conversions have completed, the subgroup EOC interrupt raw flag ADIRAWG in the ADCIRAW register will be asserted. The A/D converter will now not continue to perform the next n conversions until the next trigger event occurs. The conversion cycle will end after all the group channels, of which the total number is defined by the ADSEQL[2:0] bits in the ADCCONV register, have finished their conversion, at which point the cycle EOC interrupt raw flag ADIRAWC in the ADCIRAW register will be asserted. If a new trigger event occurs after all the subgroup channels have all been converted, i.e., a complete conversion cycle has been finished, the conversion will restart from the first subgroup.

Example:

A/D subgroup length = 3 (ADSUBL=2) and sequence length = 8 (ADSEQL=7), channels to be converted = 2, 4, 7, 5, 6, 3, 0 and 1 - specific converting sequence as defined in the ADCLSTn registers,

- Trigger 1: subgroup channels to be converted are CH2, CH4 and CH7 with the ADIRAWG flag being asserted after subgroup EOC.
- Trigger 2: subgroup channels to be converted are CH5, CH6 and CH3 with the ADIRAWG flag being asserted after subgroup EOC.
- Trigger 3: subgroup channels to be converted are CH0 and CH1 with the ADIRAWG flag being asserted after subgroup EOC. Also a Cycle end of conversion (EOC) interrupt raw flag ADIRAWC will be asserted.
- Trigger 4: subgroup channels to be converted are CH2, CH4 and CH7 with the ADIRAWG flag being asserted - conversion sequence restarts from the beginning.

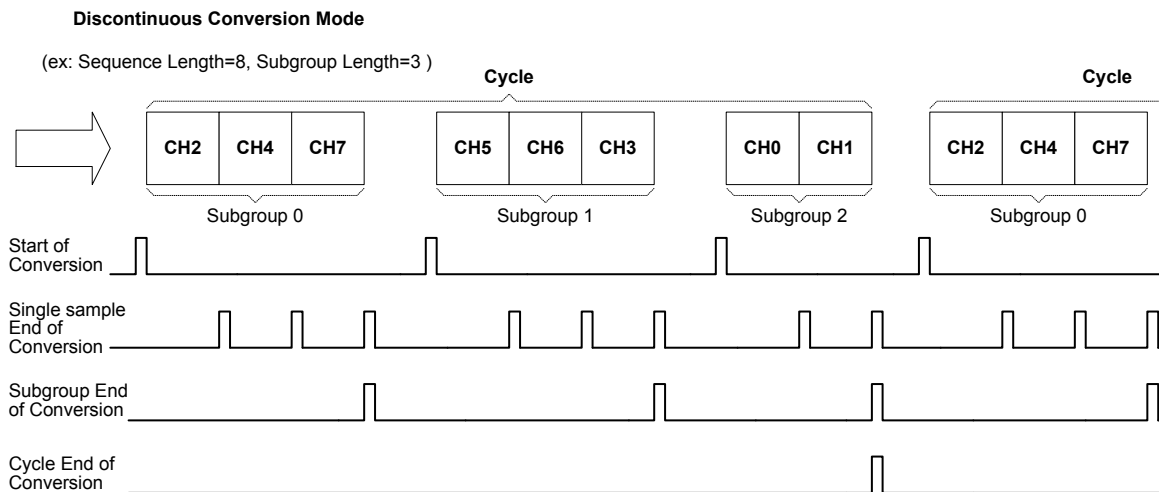


Figure 30. Discontinuous Conversion Mode

Start Conversion on External Event

An A/D conversion can be initiated by a software trigger, a comparator output transition event, a General-Purpose Timer Module (GPTM) event, a Motor Control Timer Module (MCTM) event, a Basic Function Timer Module (BFTM) event or an external trigger. Each trigger source can be enabled by setting the corresponding enable control bit in the ADCTCR register and then selected by configuring the associated selection bits in the ADCTSR register to start a group channel conversion.

An A/D converter conversion can be started by setting the software trigger bit, ADSC, in the ADCTSR register for the group channel when the software trigger enable bit, ADSW, in the ADCTCR register is set to 1. After the A/D converter starts converting the analog data, the corresponding enable bit ADSC will be cleared to 0 automatically.

The A/D converter can also be triggered to start a group conversion by a TM event. The TM events include a GPTM or MCTM master trigger output MTO, four GPTM or MCTM channel outputs CH0~CH3 and a BFTM trigger output. If the corresponding Timer trigger enable bit is set to 1 and the trigger output or the TM channel event is selected via the relevant TM event selection bits, the A/D converter will start a conversion when a rising edge of the selected trigger event occurs.

In addition to the internal trigger sources, the A/D converter can be triggered to start a conversion by an external trigger event. The external trigger event is derived from the external lines, EXTI_n. If the external trigger enable bit ADEXTI is set to 1 and the corresponding EXTI line is selected by configuring the ADEXTIS field in the ADCTSR register, the A/D converter will start a conversion when an EXTI line active edge determined in the EXTI Unit occurs.

Sampling Time Setting

The conversion channel sampling time can be programmed according to the input resistance of the input voltage source. This sampling time must be enough for the input voltage source to charge the internal sample and hold capacitor in A/D the converter to the input voltage level. Each conversion channel is sampled with the same sampling time. By modifying the ADST[7:0] bits in the ADCSTR register, the sampling time of the analog input signal can be determined.

The total conversion time (T_{conv}) is calculated using the following formula:

$$T_{conv} = T_{Sampling} + T_{Latency}$$

Where the minimum sampling time $T_{Sampling} = 1.5$ cycles (when ADST[7:0] = 0) and the minimum channel conversion latency $T_{Latency} = 12.5$ cycles.

Example:

With the A/D Converter clock $CK_ADC = 14$ MHz and a sampling time = 1.5 cycles:

$$T_{conv} = 1.5 + 12.5 = 14 \text{ cycles} = 1 \mu\text{s}$$

Data Format

The ADC conversion result can be read in the ADCDRy register and the data format is shown in the following Table 1.

Table 27. Data format in ADCDR [15:0]

Description	ADCDR register Data Format
Right aligned	"0_0_0_0_d11_d10_d9_d8_d7_d6_d5_d4_d3_d2_d1_d0"

Analog Watchdog

The A/D converter includes a watchdog function to monitor the converted data. There are two kinds of thresholds for the watchdog monitor function, known as the watchdog upper threshold and watchdog lower threshold, which are specified in the Watchdog Upper and Lower Threshold Registers respectively. The watchdog monitor function is enabled by setting the watchdog upper and lower threshold monitor function enable bits, ADWUE and ADWLE, in the watchdog control register ADCWCR. The channel to be monitored can be specified by configuring the ADWCH and ADWALL bits. When the converted data is less or higher than the lower or upper threshold, as defined in the ADCLTR or ADCUTR registers respectively, the watchdog lower or upper threshold interrupt raw flags, ADIRAWL or ADIRAWU in the ADCIRAW register, will be asserted if the watchdog lower or upper threshold monitor function is enabled. If the lower or upper threshold interrupt raw flag is asserted and the corresponding interrupt is enabled by setting the ADIML or ADIMU bit in the ADCIME register, the A/D watchdog lower or upper threshold interrupt will be generated.

Interrupts

When an A/D conversion is completed, an End of Conversion EOC event will occur. There are three kinds of EOC events which are known as single sample EOC, subgroup EOC and cycle EOC for A/D conversion. A single sample EOC event will occur and the single sample EOC interrupt raw flag, ADIRAWS bits in the ADCIRAW register, will be asserted when a single channel conversion has completed. A subgroup EOC event will occur and the subgroup EOC interrupt raw flag, ADIRAWG in the ADCIRAW register, will be asserted when a subgroup conversion has completed. A cycle EOC event will occur and the cycle EOC interrupt raw flag, ADIRAWC bits in the ADCIRAW register, will be asserted when a cycle conversion is finished. When a single sample EOC, a subgroup EOC or a cycle EOC raw flag is asserted and the corresponding interrupt enable bit, ADIMC, ADIEG or ADIES bit in the ADCIER register, is set to 1, the associated interrupt will be generated.

After a conversion has completed, the 12-bit digital data will be stored in the associated ADCDRy registers and the value of the data valid flag named as ADVLDy will be changed from low to high. The converted data should be read by the application program, after which the data valid flag ADVLDy will be automatically changed from high to low. Otherwise, a data overwrite event will occur and the data overwrite interrupt raw flag ADIRAWO bit in the ADCIRAW register will be asserted. When the related data overwrite raw flag is asserted, the data overwrite interrupt will be generated if the interrupt enable bit ADIEO in the ADCIER register is set to 1.

If the A/D watchdog monitor function is enabled and the data after a channel conversion is less than the lower threshold or higher than the upper threshold, the watchdog lower or upper threshold interrupt raw flag ADIRAWL or ADIRAWU in the ADCIRAW register will be asserted. When the ADIRAWL or ADIRAWU flag is asserted and the corresponding interrupt enable bit, ADIEL or ADIEU in the ADCIER register, is set a watchdog lower or upper threshold interrupt will be generated.

The A/D Converter interrupt clear bits are used to clear the associated A/D converter interrupt raw and interrupt status bits. Writing a 1 into the specific A/D converter interrupt clear bit in the A/D converter interrupt clear register ADCICLR will clear the corresponding A/D converter interrupt raw and interrupt status bits. These bits are automatically cleared to 0 by hardware after being set to 1.

PDMA Request

The converted channel value will be stored in the corresponding data register. The A/D Converter can inform the MCU using the A/D Converter EOC interrupt if a new conversion data is already stored in the ADCDRn register. Users also can determine the PDMA request is asserted by setting the ADDMAC, ADDMAG or ADDMAS bits in the ADCDMAR register. A PDMA request will be automatically generated at the relevant end of A/D conversion. The detail description will be introduced in the ADCDMAR register description.

Register Map

The following table shows the A/D Converter registers and reset values.

Table 28. A/D Converter Register Map

Register	Offset	Description	Reset Value
ADCCR	0x000	ADC Conversion Control Register	0x0000_0000
ADCLST0	0x004	ADC Conversion List Register 0	0x0000_0000
ADCLST1	0x008	ADC Conversion List Register 1	0x0000_0000
ADCSTR	0x020	ADC Input Sampling Time Register	0x0000_0000
ADCDR0	0x030	ADC Conversion Data Register 0	0x0000_0000
ADCDR1	0x034	ADC Conversion Data Register 1	0x0000_0000
ADCDR2	0x038	ADC Conversion Data Register 2	0x0000_0000
ADCDR3	0x03C	ADC Conversion Data Register 3	0x0000_0000
ADCDR4	0x040	ADC Conversion Data Register 4	0x0000_0000
ADCDR5	0x044	ADC Conversion Data Register 5	0x0000_0000
ADCDR6	0x048	ADC Conversion Data Register 6	0x0000_0000
ADCDR7	0x04C	ADC Conversion Data Register 7	0x0000_0000
ADCTCR	0x070	ADC Trigger Control Register	0x0000_0000
ADCTSR	0x074	ADC Trigger Source Register	0x0000_0000
ADCWCR	0x078	ADC Watchdog Control Register	0x0000_0000
ADCTR	0x07C	ADC Watchdog Threshold Register	0x0000_0000
ADCIER	0x080	ADC Interrupt Enable register	0x0000_0000
ADCIRAW	0x084	ADC Interrupt Raw Status Register	0x0000_0000
ADCISR	0x088	ADC Interrupt Status Register	0x0000_0000
ADCICLR	0x08C	ADC Interrupt Clear Register	0x0000_0000
ADCDMAR	0x090	ADC DMA Request Register	0x0000_0000

Register Descriptions

ADC Conversion Control Register – ADCCR

This register specifies the mode setting, sequence length, and subgroup length of the ADC conversion mode. Note that once the content of ADCCR is changed, the conversion in progress will be aborted and the A/D converter will return to an idle state. The application program has to wait for at least one CK_ADC clock before issuing the next command.

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					ADSUBL				
						RW	0	RW	0	
	15	14	13	12	11	10	9	8		
Type/Reset	Reserved					ADSEQL				
						RW	0	RW	0	
	7	6	5	4	3	2	1	0		
Type/Reset	ADCEN	ADCRST	Reserved				ADMODE			
	RW	0	RW	0			RW	0	RW	0

Bits	Field	Descriptions
[18:16]	ADSUBL	<p>ADC Conversion Subgroup Length</p> <p>The ADSUBL field specifies the conversion channel length of each subgroup in the Discontinuous Conversion Mode. Subgroup length = ADSUBL [2:0] + 1. If the sequence length (ADSEQL [2:0] + 1) is not a multiple of the subgroup length (ADSUBL [2:0] + 1), the last subgroup will be the rest of the group channels that have not been converted.</p>
[10:8]	ADSEQL	<p>ADC Conversion Length</p> <p>0x00: The channel specified by the ADSEQ0 field in the ADCLST0 register will be converted.</p> <p>Others: Sequence length = ADSEQL [2:0] + 1.</p> <p>The ADSEQL field specifies the whole conversion sequence length for the conversion group.</p>
[7]	ADCEN	<p>ADC Enable</p> <p>0: ADC disable 1: ADC enable</p> <p>When this bit is cleared to 0, the A/D converter will be disabled and the CK_ADC clock will also be switched off.</p>
[6]	ADCRST	<p>ADC Reset</p> <p>0: No effect 1: Reset A/D converter except for the A/D Converter controller.</p>

Bits	Field	Descriptions															
[1:0]	ADMODE	ADC Conversion Mode															
		<table border="1"> <thead> <tr> <th>ADMODE [1:0]</th> <th>Mode</th> <th>Descriptions</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>One shot mode</td> <td>After a start trigger, the conversion will be executed on the specific channels for the whole conversion sequence once.</td> </tr> <tr> <td>01</td> <td>Reserved</td> <td></td> </tr> <tr> <td>10</td> <td>Continuous mode</td> <td>After a start trigger, the conversion will be executed on the specific channels for the whole sequence continuously until conversion mode is changed.</td> </tr> <tr> <td>11</td> <td>Discontinuous mode</td> <td>After a start trigger, the conversion will be executed on the current subgroup. When the last subgroup is finished, the conversion will restart from the first subgroup if another start trigger occurs.</td> </tr> </tbody> </table>	ADMODE [1:0]	Mode	Descriptions	00	One shot mode	After a start trigger, the conversion will be executed on the specific channels for the whole conversion sequence once.	01	Reserved		10	Continuous mode	After a start trigger, the conversion will be executed on the specific channels for the whole sequence continuously until conversion mode is changed.	11	Discontinuous mode	After a start trigger, the conversion will be executed on the current subgroup. When the last subgroup is finished, the conversion will restart from the first subgroup if another start trigger occurs.
ADMODE [1:0]	Mode	Descriptions															
00	One shot mode	After a start trigger, the conversion will be executed on the specific channels for the whole conversion sequence once.															
01	Reserved																
10	Continuous mode	After a start trigger, the conversion will be executed on the specific channels for the whole sequence continuously until conversion mode is changed.															
11	Discontinuous mode	After a start trigger, the conversion will be executed on the current subgroup. When the last subgroup is finished, the conversion will restart from the first subgroup if another start trigger occurs.															

ADC Conversion List Register 0 – ADCLST0

This register specifies the conversion sequence order No.0 ~ No.3 of the ADC.

Offset: 0x004

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
	Reserved				ADSEQ3				
Type/Reset				RW	0	RW	0	RW	0
	23	22	21	20	19	18	17	16	
	Reserved				ADSEQ2				
Type/Reset				RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8	
	Reserved				ADSEQ1				
Type/Reset				RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0	
	Reserved				ADSEQ0				
Type/Reset				RW	0	RW	0	RW	0

Bits	Field	Descriptions
[28:24]	ADSEQ3	ADC Conversion Sequence Select 3 Select the ADC input channel for the 3 rd ADC conversion sequence. 0x0: ADC_IN0 0x1: ADC_IN1 0x2: ADC_IN2 0x3: ADC_IN3 0x4: ADC_IN4 0x5: ADC_IN5 0x6: ADC_IN6 0x7: ADC_IN7 0x8: ADC_IN8 0x9: ADC_IN9 0xA: ADC_IN10 0xB: ADC_IN11 0xC ~ 0xF: Reserved 0x10: Analog ground, AVSS (V _{REF-}) 0x11: Analog power, AVDD (V _{REF+}) 0x12 ~ 0x1F: Invalid setting. These values must not be selected as it may cause the ADC abnormal operations.
[20:16]	ADSEQ2	ADC Conversion Sequence Select 2
[12:8]	ADSEQ1	ADC Conversion Sequence Select 1
[4:0]	ADSEQ0	ADC Conversion Sequence Select 0

ADC Conversion List Register 1 – ADCLST1

This register specifies the conversion sequence order No.4 ~ No.7 of the ADC.

Offset: 0x008

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24		
	Reserved			ADSEQ7						
Type/Reset				RW	0	RW	0	RW	0	
	23	22	21	20	19	18	17	16		
	Reserved			ADSEQ6						
Type/Reset				RW	0	RW	0	RW	0	
	15	14	13	12	11	10	9	8		
	Reserved			ADSEQ5						
Type/Reset				RW	0	RW	0	RW	0	
	7	6	5	4	3	2	1	0		
	Reserved			ADSEQ4						
Type/Reset				RW	0	RW	0	RW	0	

Bits	Field	Descriptions
[28:24]	ADSEQ7	ADC Conversion Sequence Select 7 Select the ADC input channel for the 7 th ADC conversion sequence. 0x0: ADC_IN0 0x1: ADC_IN1 0x2: ADC_IN2 0x3: ADC_IN3 0x4: ADC_IN4 0x5: ADC_IN5 0x6: ADC_IN6 0x7: ADC_IN7 0x8: ADC_IN8 0x9: ADC_IN9 0xA: ADC_IN10 0xB: ADC_IN11 0xC ~ 0xF: Reserved 0x10: Analog ground, AVSS (V _{REF-}) 0x11: Analog power, AVDD (V _{REF+}) 0x12 ~ 0x1F: Invalid setting. These values must not be selected as it may cause the ADC abnormal operations.
[20:16]	ADSEQ6	ADC Conversion Sequence Select 6
[12:8]	ADSEQ5	ADC Conversion Sequence Select 5
[4:0]	ADSEQ4	ADC Conversion Sequence Select 4

ADC Input Sampling Time Register – ADCSTR

This register specifies the A/D converter input channel sampling time.

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	ADST								
	RW	0	RW	0	RW	0	RW	0	RW
									0

Bits	Field	Descriptions
[7:0]	ADST	ADC Input Channel Sampling Time Sampling time = (ADST[7:0] + 1.5) × CK_ADC clocks.

ADC Conversion Data Register y – ADCDRy, y = 0 ~ 7

This register is used to store the conversion data of the conversion sequence order No.y which is specified by the ADSEQy field in the ADCLSTn (n= 0 ~ 1) registers.

Offset: 0x030 ~ 0x04C

Reset value: 0x0000_0000

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

Bits	Field	Descriptions
[31]	ADVLDy	ADC Conversion Data of Sequence Order No.y Valid Bit (y = 0 ~ 7) 0: Data are invalid or have been read 1: New data is valid
[15:0]	ADDy	ADC Conversion Data of Sequence Order No.y (y = 0 ~ 7) The conversion result of Sequence Order ADSEQy in the ADCLSTn (n=0 ~1) registers

ADC Trigger Control Register – ADCTCR

This register contains the ADC start conversion trigger enable bits.

Offset: 0x070

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			CMP	BFTM	TM	ADEXTI	ADSW	
				RW	0	RW	0	RW	0
									0

Bits	Field	Descriptions
[4]	CMP	ADC Conversion CMP Event Trigger enable control 0: Disable conversion trigger by CMP output transition 1: Enable conversion trigger by CMP output transition
[3]	BFTM	ADC Conversion BFTM Event Trigger enable control 0: Disable conversion trigger by BFTM events 1: Enable conversion trigger by BFTM events
[2]	TM	ADC Conversion GPTM or MCTM Event Trigger enable control 0: Disable conversion trigger by GPTM or MCTM events 1: Enable conversion trigger by GPTM or MCTM events
[1]	ADEXTI	ADC Conversion EXTI Event Trigger enable control 0: Disable conversion trigger by EXTI lines 1: Enable conversion trigger by EXTI lines
[0]	ADSW	ADC Conversion Software Trigger enable control 0: Disable conversion trigger by software trigger bit 1: Enable conversion trigger by software trigger bit

ADC Trigger Source Register – ADCTSR

This register contains the trigger source selection and the software trigger bit of the conversion.

Offset: 0x074

Reset value: 0x0000_0000

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

Bits	Field	Descriptions
[26:24]	TME	GPTM or MCTM Trigger Event Selection of ADC Conversion 000: GPTM or MCTM MTO event 001: GPTM or MCTM CH00 event 010: GPTM or MCTM CH10 event 011: GPTM or MCTM CH20 event 100: GPTM or MCTM CH30 event Others: Reserved - Should not be used to avoid unpredictable results
[20]	CMPS	CMP Trigger Selection of ADC Conversion 0: CMP0 1: CMP1
[19]	BFTMS	BFTM Trigger Timer Selection of ADC Conversion 0: BFTM0 1: BFTM1
[18:16]	TMS	GPTM or MCTM Trigger Timer Selection of ADC Conversion 000: MCTM 001: Reserved 010: GPTM0 011: GPTM1 Others: Reserved - Should not be used to avoid unpredictable results
[11:8]	ADEXTIS	EXTI Trigger Source Selection of ADC Conversion 0x00: EXTI line 0 0x01: EXTI line 1 ... 0x0F: EXTI line 15 Note that the EXTI line active edge to start an A/D conversion is determined in the External Interrupt/Event Control Unit, EXTI.
[0]	ADSC	ADC Conversion Software Trigger Bit 0: No operation 1: Start conversion immediately This bit is set by software to start a conversion manually and then cleared by hardware automatically after conversion started.

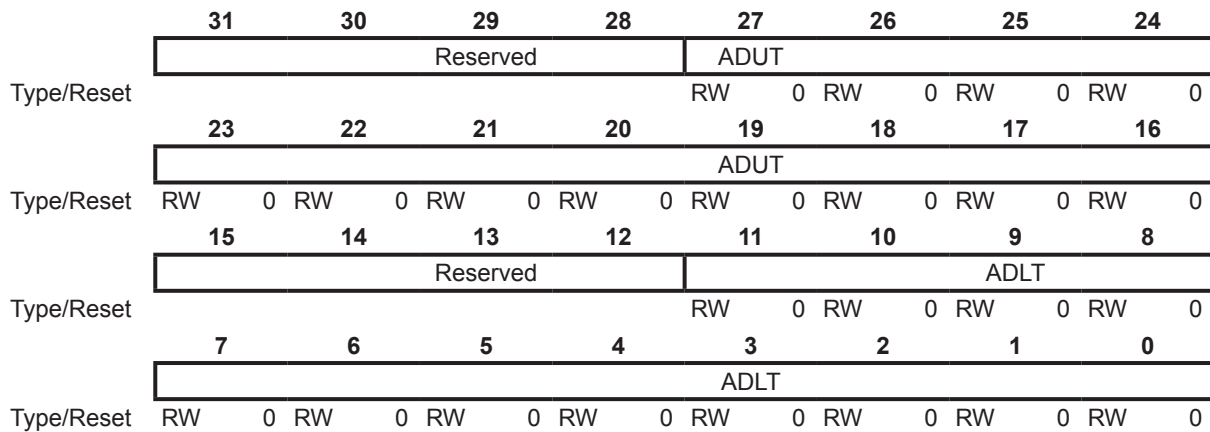
Bits	Field	Descriptions
[1]	ADWUE	ADC Watchdog Upper Threshold Enable Bit 0: Disable upper threshold monitor function 1: Enable upper threshold monitor function
[0]	ADWLE	ADC Watchdog Lower Threshold Enable Bit 0: Disable lower threshold monitor function 1: Enable lower threshold monitor function

ADC Watchdog Threshold Register – ADCTR

This register specifies the upper and lower threshold of the ADC watchdog function.

Offset: 0x07C

Reset value: 0x0000_0000



Bits	Field	Descriptions
[27:16]	ADUT	ADC Watchdog Upper Threshold Value Specify the upper threshold for the ADC watchdog monitor function.
[11:0]	ADLT	ADC Watchdog Lower Threshold Value Specify the lower threshold for the ADC watchdog monitor function.

ADC Interrupt Enable Register – ADCIER

This register contains the ADC interrupt enable bits.

Offset: 0x080

Reset value: 0x0000_0000

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

Bits	Field	Descriptions
[24]	ADIEO	ADC Data Register Overwrite Interrupt enable 0: ADC data register overwrite interrupt is disabled 1: ADC data register overwrite interrupt is enabled
[17]	ADIEU	ADC Watchdog Upper Threshold Interrupt enable 0: ADC watchdog upper threshold interrupt is disabled 1: ADC watchdog upper threshold interrupt is enabled
[16]	ADIEL	ADC Watchdog Lower Threshold Interrupt enable 0: ADC watchdog lower threshold interrupt is disabled 1: ADC watchdog lower threshold interrupt is enabled
[2]	ADIEC	ADC Cycle EOC Interrupt enable 0: ADC cycle end of conversion interrupt is disabled 1: ADC cycle end of conversion interrupt is enabled
[1]	ADIEG	ADC Subgroup EOC Interrupt enable 0: ADC subgroup end of conversion interrupt is disabled 1: ADC subgroup end of conversion interrupt is enabled
[0]	ADIES	ADC Single EOC Interrupt enable 0: ADC single end of conversion interrupt is disabled 1: ADC single end of conversion interrupt is enabled

ADC Interrupt Raw Status Register – ADCIRAW

This register contains the ADC interrupt raw status bits.

Offset: 0x084

Reset value: 0x0000_0000

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

Bits	Field	Descriptions
[24]	ADIRAWO	ADC Data Register Overwrite Interrupt Raw Status 0: ADC data register overwrite event does not occur 1: ADC data register overwrite event occurs
[17]	ADIRAWU	ADC Watchdog Upper Threshold Interrupt Raw Status 0: ADC watchdog upper threshold event does not occur 1: ADC watchdog upper threshold event occurs
[16]	ADIRAWL	ADC Watchdog Lower Threshold Interrupt Raw Status 0: ADC watchdog lower threshold event does not occurs 1: ADC watchdog lower threshold event occurs
[2]	ADIRAWC	ADC Cycle EOC Interrupt Raw Status 0: ADC cycle end of conversion event does not occur 1: ADC cycle end of conversion event occurs
[1]	ADIRAWG	ADC Subgroup EOC Interrupt Raw Status 0: ADC subgroup end of conversion event does not occur 1: ADC subgroup end of conversion event occurs
[0]	ADIRAWS	ADC Single EOC Interrupt Raw Status 0: ADC single end of conversion event does not occur 1: ADC single end of conversion event occurs

ADC Interrupt Status Register – ADCISR

This register contains the ADC interrupt masked status bits. The corresponding interrupt status will be set to 1 if the associated interrupt event occurs and the related enable bit is set to 1.

Offset: 0x088

Reset value: 0x0000_0000

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

Bits	Field	Descriptions
[24]	ADISRO	ADC Data Register Overwrite Interrupt Status 0: ADC data register overwrite interrupt does not occur or the data register overwrite interrupt is disabled. 1: ADC data register overwrite interrupt occurs as the data register overwrite interrupt is enabled.
[17]	ADISRU	ADC Watchdog Upper Threshold Interrupt Status 0: ADC watchdog upper threshold interrupt does not occur or the watchdog upper threshold interrupt is disabled. 1: ADC watchdog upper threshold interrupt occurs as the watchdog upper threshold interrupt is enabled.
[16]	ADISRL	ADC Watchdog Lower Threshold Interrupt Status 0: ADC watchdog lower threshold interrupt does not occur or the watchdog lower threshold interrupt is disabled. 1: ADC watchdog lower threshold interrupt occurs as the watchdog lower threshold interrupt is enabled.
[2]	ADISRC	ADC Cycle EOC Interrupt Status 0: ADC cycle end of conversion interrupt does not occur or the cycle end of conversion interrupt is disabled. 1: ADC cycle end of conversion interrupt occurs as the cycle end of conversion interrupt is enabled.
[1]	ADISRG	ADC Subgroup EOC Interrupt Status 0: ADC subgroup end of conversion interrupt does not occur or the subgroup end of conversion interrupt is disabled. 1: ADC subgroup end of conversion interrupt occurs as the subgroup end of conversion interrupt is enabled.
[0]	ADISRS	ADC Single EOC Interrupt Status 0: ADC single end of conversion interrupt does not occur or the single end of conversion interrupt is disabled. 1: ADC single end of conversion interrupt occurs as the single end of conversion interrupt is enabled.

ADC Interrupt Clear Register – ADCICLR

This register provides the clear bits used to clear the interrupt raw and masked status of the ADC. These bits are set to 1 by software to clear the interrupt status and automatically cleared to 0 by hardware after being set to 1.

Offset: 0x08C

Reset value: 0x0000_0000

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

Bits	Field	Descriptions
[24]	ADICLRO	ADC Data Register Overwrite Interrupt Status Clear Bit 0: No effect 1: Clear ADISRO and ADIRAWO bits
[17]	ADICLRU	ADC Watchdog Upper Threshold Interrupt Status Clear Bit 0: No effect 1: Clear ADISRU and ADIRAWU bits
[16]	ADICLRL	ADC Watchdog Lower Threshold Interrupt Status Clear Bit 0: No effect 1: Clear ADISRL and ADIRAWL bits
[2]	ADICLRC	ADC Cycle EOC Interrupt Status Clear Bit 0: No effect 1: Clear ADISRC and ADIRAWC bits
[1]	ADICLRG	ADC Subgroup EOC Interrupt Status Clear Bit 0: No effect 1: Clear ADISRG and ADIRAWG bits
[0]	ADICLRS	ADC Single EOC Interrupt Status Clear Bit 0: No effect 1: Clear ADISRS and ADIRAWS bits

ADC DMA Request Register – ADCDMAR

This register contains the ADC DMA request enable bits.

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	Reserved								
	7	6	5	4	3	2	1	0	
Type/Reset	Reserved					ADDMAC	ADDMAG	ADDMAS	
						RW 0	RW 0	RW 0	

Bits	Field	Descriptions
[2]	ADDMAC	ADC Cycle EOC DMA Request Enable Bit 0: ADC cycle end of conversion DMA request is disabled 1: ADC cycle end of conversion DMA request is enabled
[1]	ADDMAG	ADC Subgroup EOC DMA Request Enable Bit 0: ADC subgroup end of conversion DMA request is disabled 1: ADC subgroup end of conversion DMA request is enabled
[0]	ADDMAS	ADC Single EOC DMA Request Enable Bit 0: ADC single end of conversion DMA request is disabled 1: ADC single end of conversion DMA request is enabled

13 Comparator (CMP)

Introduction

The two general purpose comparators (CMP) are implemented within the device. They can be configured either as standalone comparators or combined with the different kinds of peripheral IP. Each comparator is capable of asserting interrupts to the NVIC or wakeup the CPU Deep Sleep mode through EXTI wakeup event management unit.

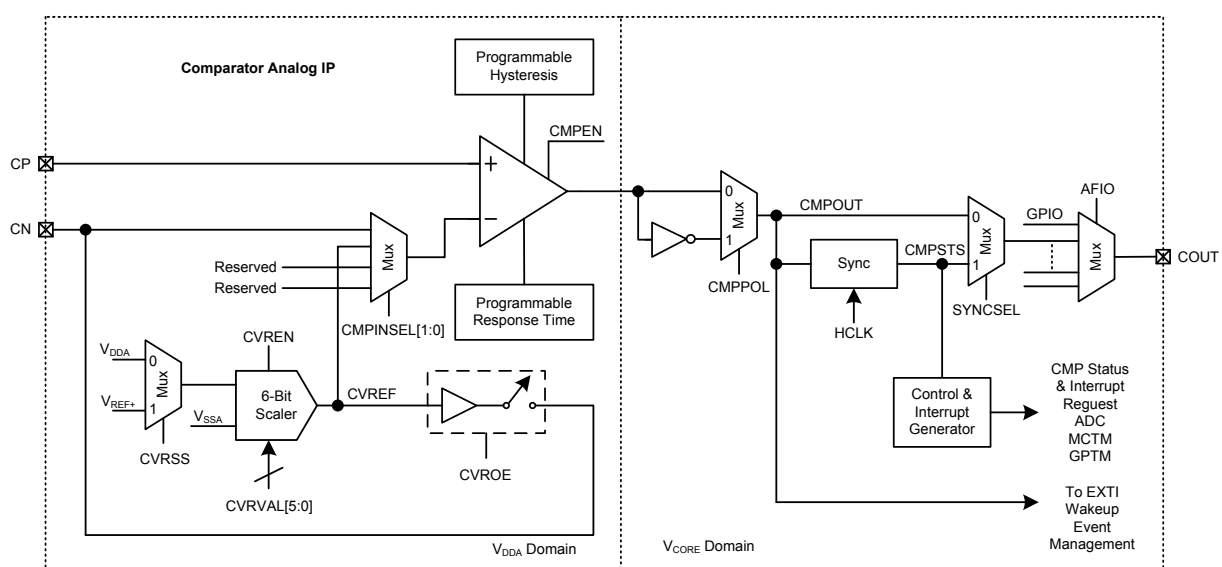


Figure 31. Comparator Block Diagram

Features

- Rail-to-rail comparator
- Configurable negative inputs used for flexible voltage selection
 - Dedicated I/O pin
 - Internal voltage reference provided by 6-bit scaler
- Programmable hysteresis
- Programming response speed and consumption
- Comparator output used as multiple timers or ADC trigger inputs
- Programmable internal voltage reference provided by 6-bit scaler.
- Comparator output transition with interrupt generation and wake-up functions from Sleep or Deep Sleep modes through the EXTI controller.

Function Descriptions

Comparator Inputs and Output

The I/O pins used as comparator inputs or output must be configured in the AFIO controller registers. The detail comparator I/Os information will be referred in the pin assignment table in the datasheet. The output can also be internally connected to a variety of timer or ADC for trigger purpose. The comparator output can be used for both internal and external functions simultaneously.

Comparator Voltage Reference

The comparator voltage reference is a 64-tap resistor ladder network that provides a selectable reference voltage. It also has a power-down function to conserve the power when the reference is not used. The comparator voltage reference provides 64 distinct levels. The equation used to calculate the comparator voltage reference is as follows:

$$CVREF = CVRVAL \times (V_{RP} - V_{SSA}) / 63,$$

The comparator voltage reference power source, V_{RP} , can come from either the V_{DDA} or external V_{REF+} pin. The voltage source is selected by the CVRSS bit in the Comparator Control Register CMPCR. The primary purpose of the CVREF output is to provide a reference voltage for the analog comparator. It may also be used independently of them and configured output to CN pin by setting the CVROE bit in the Comparator Control Register CVRCR. The settling time of the comparator voltage reference must be considered when changing the CVREF output.

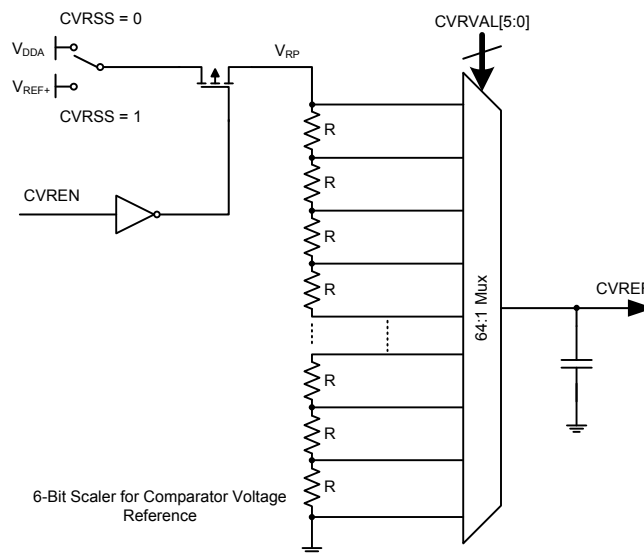


Figure 32. 6-Bit Scaler for Comparator Voltage Reference Block Diagram

Interrupts and Wakeup

The comparator can generate an interrupt when its output waveform generates a rising or falling edge and its corresponding interrupt enables control bit is also set.

For example, when a comparator output rising edge occurs, the comparator rising edge flag bit CMPRF in the Comparator Transition Flag Register CMPTFR will be set. If the comparator output rising edge interrupt enable control bit CMPRIEN in the Comparator Interrupt Enable Register CMPIER is enabled, an interrupt will then be generated and sent to the NVIC unit. Writing “1” into the comparator rising edge flag bit CMPRF in the Comparator Transition Flag Register CMPTFR will clear the CMPRF bit. The comparator output falling edge interrupt also has the same corresponding interrupt setting. A block diagram of interrupt signal for comparators is shown in Figure 33.

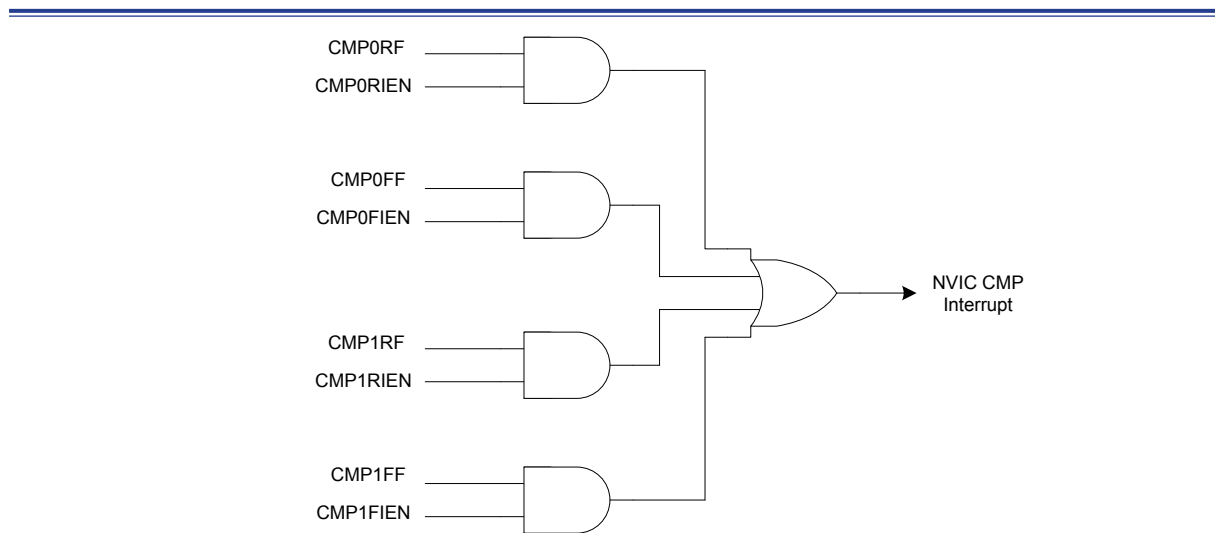


Figure 33. Interrupt Signals of Comparators

The comparator outputs are also internally connected to the EXTI Wake-up Event Management unit. The comparator output rising transition is used to wake up the MCU from the Deep Sleep 1 or 2 modes when the comparator wake-up enable bit CMPWPEN is set in the Comparator Control Register CMPCR. A block diagram of wakeup signal for comparators is shown in Figure 34.

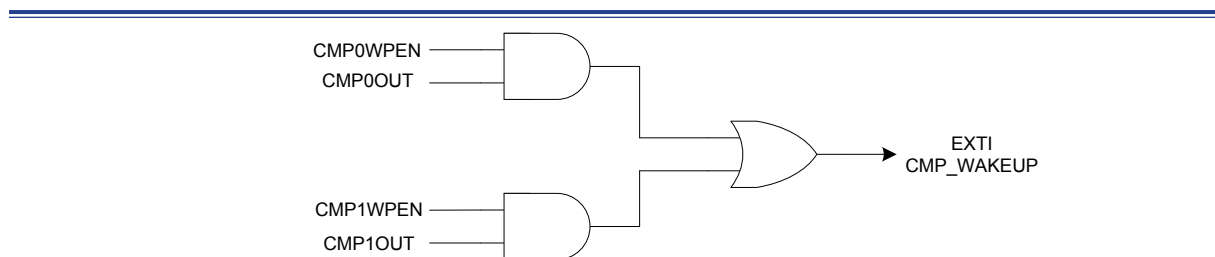


Figure 34. Wakeup Signals of Comparators

Power Mode and Hysteresis

The comparator response time can be programmed to meet the trade-off between the power consumption and application requirement. The bit CMPSM in CMPCR register can be programmed as “1” to get the comparator in the low speed mode with low power consumption.

The comparator also has four hysteresis levels to avoid spurious output transitions in case of noisy signals. The bit CMPHM[1:0] in CMPCR register can be programmed to get the different hysteresis level for the comparator.

Comparator Write-Protected mechanism

As the comparator can be used for safety purposes, it is necessary to insure that the comparator configurations cannot be altered in case of spurious register access or program counter corruption. For this purpose, the write protection is provided by writing a specific value into the PROTECT filed in the Comparator Control Register CMPCR. The write protection function is enabled in default. Before configuring the bits [15:0] in the Comparator Control Register CMPCR, the register protection bits [31:16] of the CMPCR register has to be written into with the 0x9C3A pattern first. Then the write protection mode is disabled and the CMPCR register becomes programmable. As the same reason, the comparator input and output also can be locked with the corresponding configuration lock bit in the Port n Lock Register PnLOCKR (n = A ~ E) in the GPIO unit.

Register Map

The following table shows the CMP registers and reset values.

Table 29. CMP Register Map

Register	Offset	Description	Reset Value
CMPCR0	0x000	Comparator Control Register 0	0x0001_0000
CVRVALR0	0x004	Comparator Voltage Reference Value Register 0	0x0000_0000
CMPIER0	0x008	Comparator Interrupt Enable Register 0	0x0000_0000
CMPTFR0	0x00C	Comparator Transition Flag Register 0	0x0000_0000
CMPCR1	0x100	Comparator Control Register 1	0x0001_0000
CVRVALR1	0x104	Comparator Voltage Reference Value Register 1	0x0000_0000
CMPIER1	0x108	Comparator Interrupt Enable Register 1	0x0000_0000
CMPTFR1	0x10C	Comparator Transition Flag Register 1	0x0000_0000

Register Descriptions

Comparator Control Register n – CMPCRN, n = 0 or 1

This register contains the comparator function and comparator voltage reference control bits.

Offset: 0x000 (n = 0), 0x100 (n = 1)

Reset value: 0x0001_0000

	31	30	29	28	27	26	25	24	PROTECT										
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0					
	23	22	21	20	19	18	17	16	PROTECT										
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	1					
	15	14	13	12	11	10	9	8	CMPSTS		CMPWPEN		CMPOSEL			CVRSS	CVROE	CVREN	
Type/Reset	R	0	RW/R	0	RW/R	0	RW/R	0	RW/R	0	RW/R	0	RW/R	0	RW/R	0	0		
	7	6	5	4	3	2	1	0	SYNCSEL		CMPPOL		CMPINSEL			CMPHM		CMPSM	CMPEN
Type/Reset	RW/R	0	RW/R	0	RW/R	0	RW/R	0	RW/R	0	RW/R	0	RW/R	0	RW/R	0	0		

Bits	Field	Descriptions
[31:16]	PROTECT	Register Protection For write operation: 0x9C3A: Disable the CMPCR register write protection Others values: Enable the CMPCR register write protection For read operation: 0x0000: CMPCR register write protection is disabled 0x0001: CMPCR register write protection is enabled These bits are used to enable or disable the write protection of the filed [14:0] of the CMPCR register. Enabling the write protection will make the filed [14:0] of the CMPCR register become read-only to prevent any unexpected write operation. The value read from this field will indicate if the write protection is enabled or not.
[15]	CMPSTS	Comparator Output Status 0: Output is low 1: Output is high This read-only bit is a copy of the comparator output status after the polarity selection.
[14]	CMPWPEN	Comparator Wake-up Enable 0: Disable comparator wake-up function. 1: Enable comparator wake-up function. This bit is enabled to wake up the MCU from the Deep-Sleep when the comparator output after the polarity selection changes state from low to high.

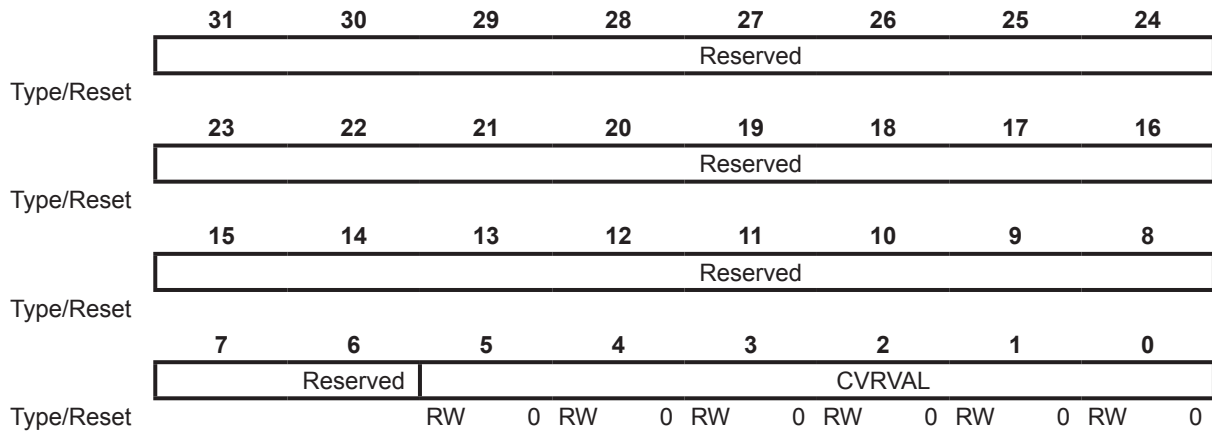
Bits	Field	Descriptions
[13:11]	CMPOSEL	<p>Comparator 0 Output Selection</p> <p>000: No selection 001: GPTM0 capture channel 3 010: MCTM capture channel 3 011: MCTM break input 1 100: ADC trigger input Other: Reserved</p> <p>Comparator 1 Output Selection</p> <p>000: No selection 001: GPTM1 capture channel 3 010: MCTM capture channel 3 011: MCTM break input 1 100: ADC trigger input Other: Reserved</p> <p>These bits are used to select the destination for the comparator output after the polarity selection.</p>
[10]	CVRSS	<p>Comparator Voltage Reference Source Selection</p> <p>0: 6-bit scaler reference voltage source comes from V_{DDA} 1: 6-bit scaler reference voltage source comes from V_{REF+}</p>
[9]	CVROE	<p>Comparator Voltage Reference Output Enable</p> <p>0: Disable 6-bit scaler output to CN pin 1: Enable 6-bit scaler output to CN pin</p>
[8]	CVREN	<p>Comparator Voltage Reference Enable</p> <p>0: Disable 6-bit scaler for comparator voltage reference 1: Enable 6-bit scaler for comparator voltage reference</p>
[7]	SYNSEL	<p>Synchronization Selection</p> <p>0: Asynchronous signal of Comparator output is selected 1: Synchronous signal of comparator output is selected</p> <p>The synchronous comparator output should be selected before being passed to the AFIO unit.</p>
[6]	CMPPOL	<p>Comparator Output Polarity Selection</p> <p>0: Comparator output is not inverted 1: Comparator output is inverted</p>
[5:4]	CMPINSEL	<p>Comparator Inverted Input Selection</p> <p>00: Comparator external CN pin is selected 01: Comparator internal 6-bit reference voltage scaler output is selected 1x: Reserved</p> <p>These bits are used to select the comparator inverted input source.</p>
[3:2]	CMPHM	<p>Comparator Hysteresis Mode Selection</p> <p>00: No hysteresis 01: Low hysteresis mode 10: Middle hysteresis mode 11: High hysteresis mode</p>
[1]	CMPSM	<p>Comparator Speed Mode Selection</p> <p>0: High speed mode 1: Low speed mode</p>
[0]	CMPEN	<p>Comparator Enable</p> <p>0: Disable Comparator (entering the power down mode) 1: Enable Comparator</p>

Comparator Voltage Reference Value Register n – CVRVALRn, n = 0 or 1

The register is used to set the comparator voltage reference level.

Offset: 0x004 (n = 0), 0x104 (n = 1)

Reset value: 0x0000_0000



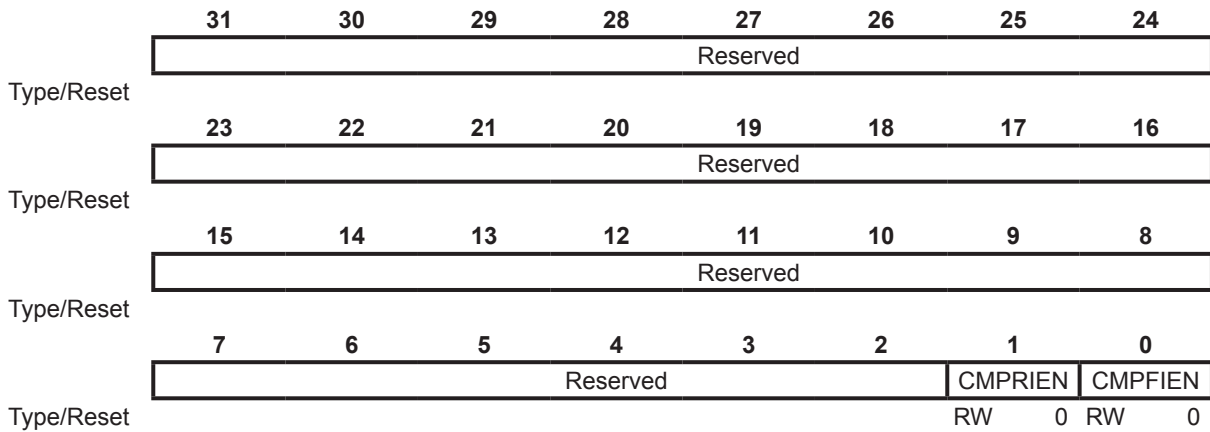
Bits	Field	Descriptions
[5:0]	CVRVAL	<p>Comparator Voltage Reference Value</p> <p>There are 64 levels of the comparator voltage reference which is set using the CVRVAL bits. The relationship between the CVRVAL register value and the comparator voltage reference CVREF is given by the following equation: $CVREF = CVRVAL \times (V_{RP} - V_{SSA}) / 63$, where the V_{RP} can come from the V_{DDA} or V_{REF+}.</p>

Comparator Interrupt Enable Register n – CMPIERn, n = 0 or 1

The register is used to enable the comparator n interrupt when the comparator output transition event occurs.

Offset: 0x008 (n = 0), 0x108 (n = 1)

Reset value: 0x0000_0000



Comparator (CMP)

Bits	Field	Descriptions
[1]	CMPRIEN	Comparator Output Rising Edge Interrupt Enable 0: Comparator output rising edge interrupt is disabled. 1: Comparator output rising edge interrupt is enabled.
[0]	CMPFIEN	Comparator Output Falling Edge Interrupt Enable 0: Comparator output falling edge interrupt is disabled. 1: Comparator output falling edge interrupt is enabled.

Comparator Transition Flag Register n – CMPTFRn, n = 0 or 1

This register contains the comparator n transition detection enable bits and flags.

Offset: 0x00C (n = 0), 0x10C (n = 1)

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							CMPRDEN	CMPFDEN	
	7	6	5	4	3	2	1	0		
Type/Reset	Reserved							CMPRF	CMPFF	
							RW	0	RW	0
							WC	0	WC	0

Bits	Field	Descriptions
[9]	CMPRDEN	<p>Comparator Output Rising Edge Detection Enable</p> <p>0: Disable comparator output rising edge detection</p> <p>1: Enable comparator output rising edge detection</p> <p>Note: The detected comparator transition signal is a copy of the comparator output state after the polarity selection and synchronized by the HCLK clock.</p>
[8]	CMPFDEN	<p>Comparator Output Falling Edge Detection Enable</p> <p>0: Disable comparator output falling edge detection</p> <p>1: Enable comparator output falling edge detection</p> <p>Note: The detected comparator transition signal is a copy of the comparator output state after the polarity selection and synchronized by the HCLK clock.</p>
[1]	CMPRF	<p>Comparator Output Rising Edge Flag</p> <p>0: No comparator output rising edge occurs</p> <p>1: Comparator output rising edge occurs</p> <p>This flag is available when the comparator output rising edge detection is enabled. This bit is set to 1 by hardware and cleared by software written a “1” into it.</p>
[0]	CMPFF	<p>Comparator Output Falling Edge Flag</p> <p>0: No Comparator output falling edge occurs</p> <p>1: Comparator output falling edge occurs</p> <p>This flag is available when the comparator output falling edge detection is enabled. This bit is set to 1 by hardware and cleared by software written a “1” into it.</p>

14 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.

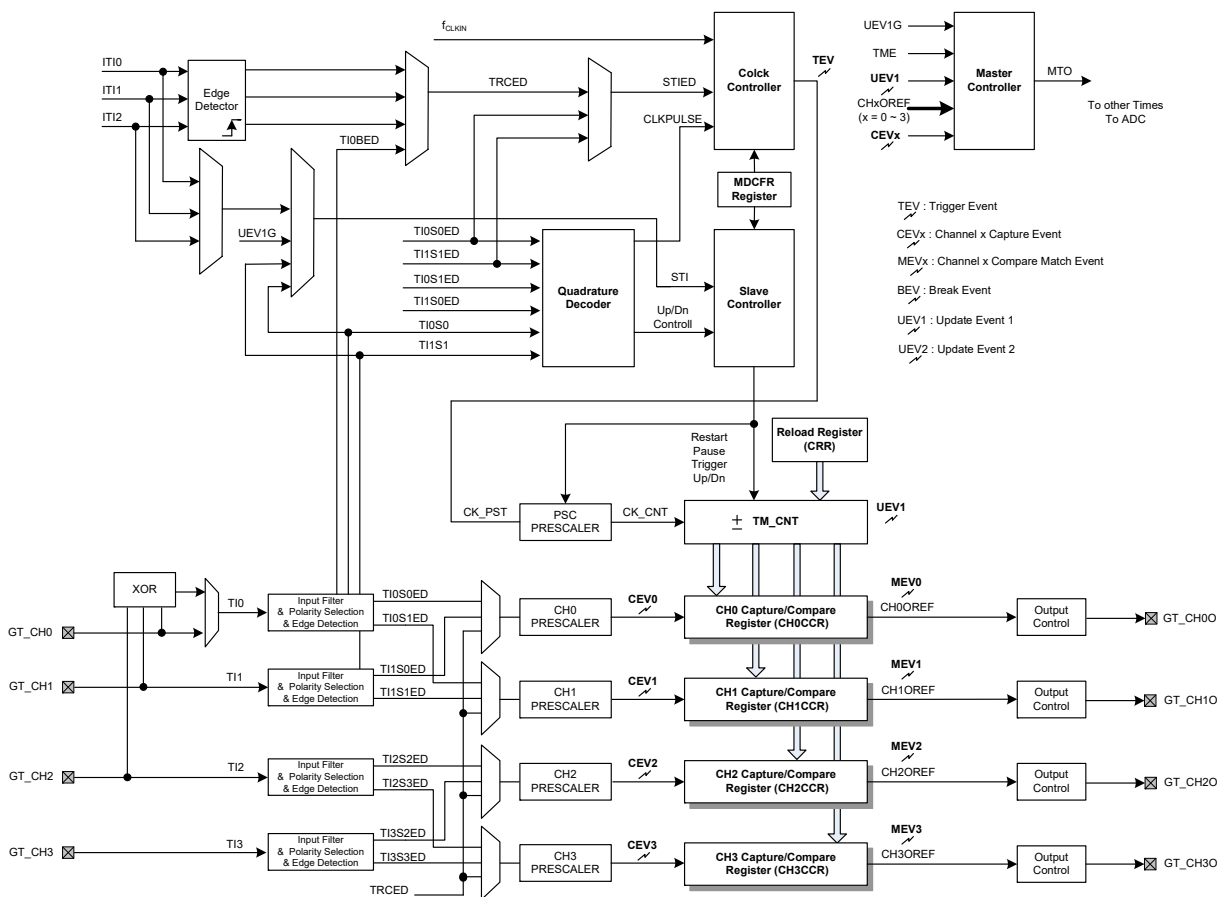


Figure 35. GPTM Block Diagram

Features

- 16-bit up/down auto-reload counter
- 16-bit programmable prescaler that allows division of the counter clock frequency by any factor between 1 and 65536
- 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 UEVGR bit in the EVGR register to 1, the counter value will also be initialized to 0.

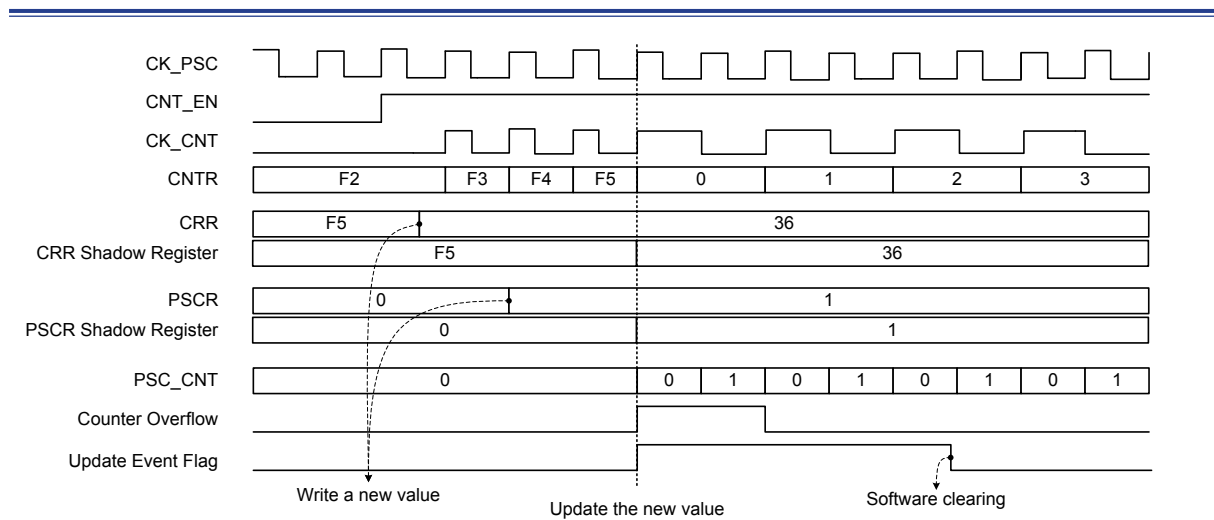


Figure 36. Up-counting Example

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.

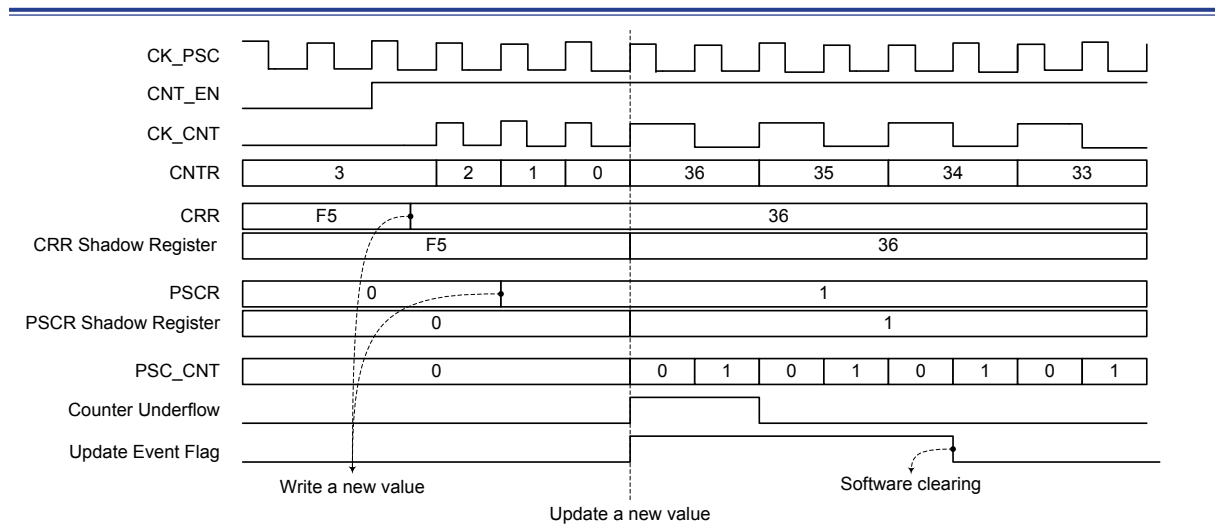


Figure 37. Down-counting Example

Center-Align 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.

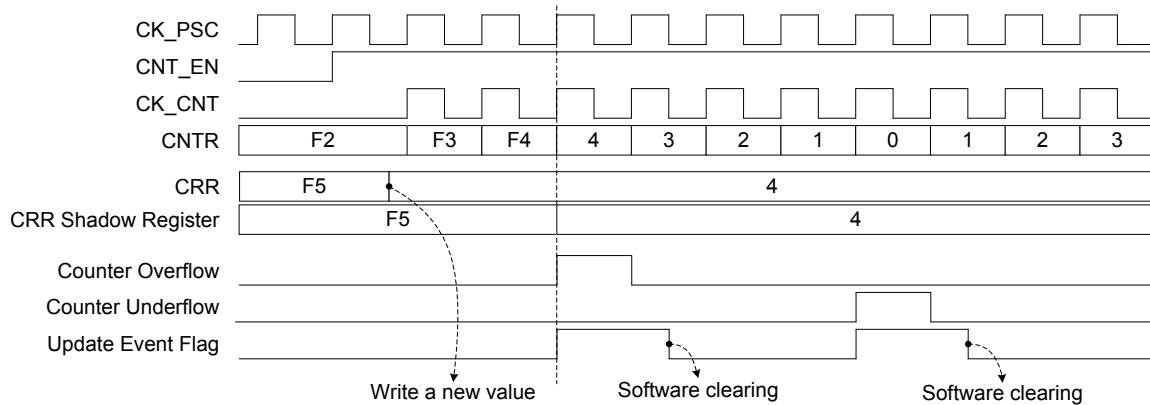


Figure 38. Center-aligned Counting Example

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. 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. When the slave mode selection bits SMSEL are set to 0x4, 0x5 or 0x6, the internal APB clock f_{CLKIN} is the counter prescaler driving clock source.

- 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.

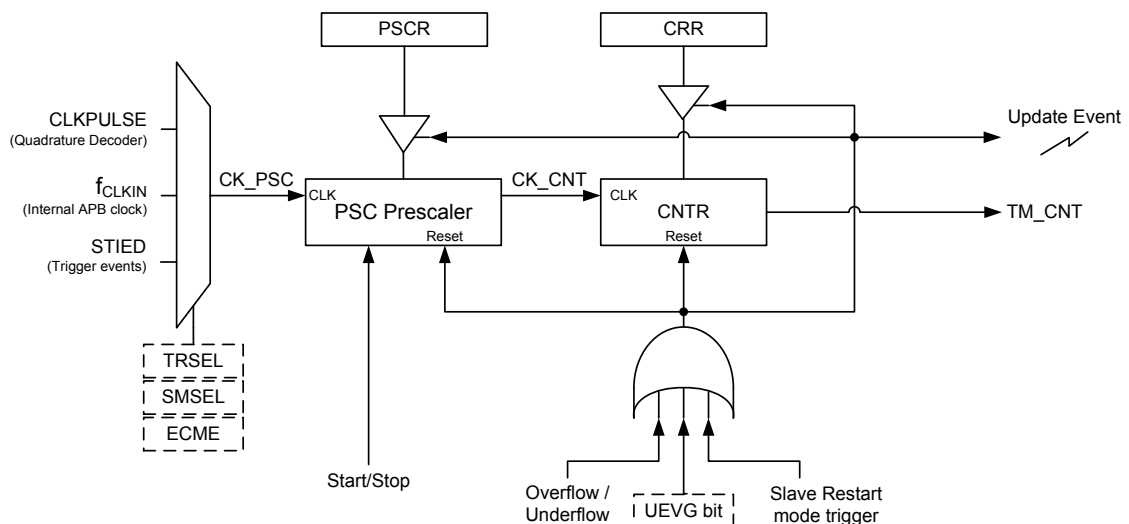


Figure 39. GPTM Clock Selection Source

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 UEV1G 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

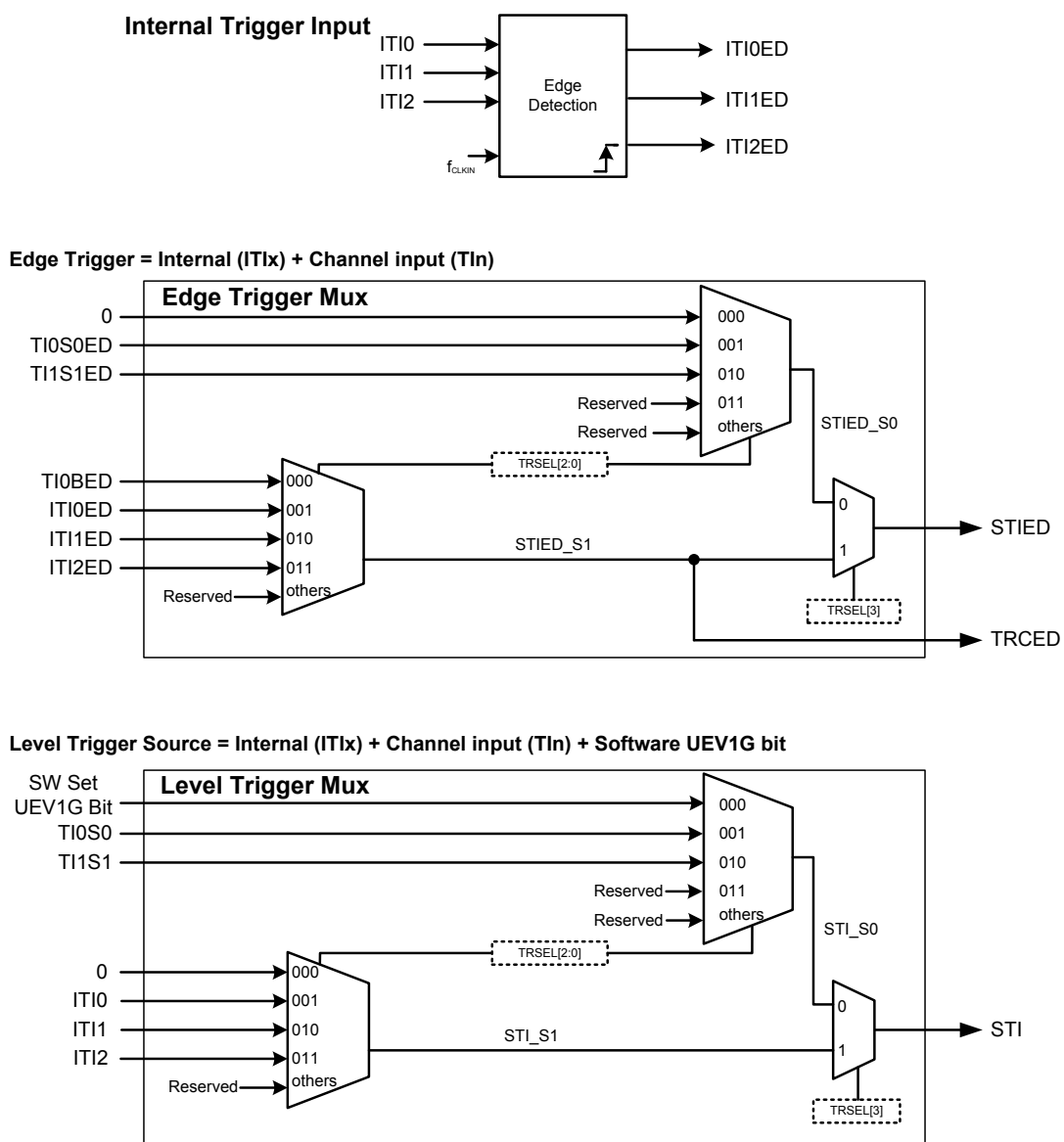


Figure 40. Trigger Controller Block

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 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.

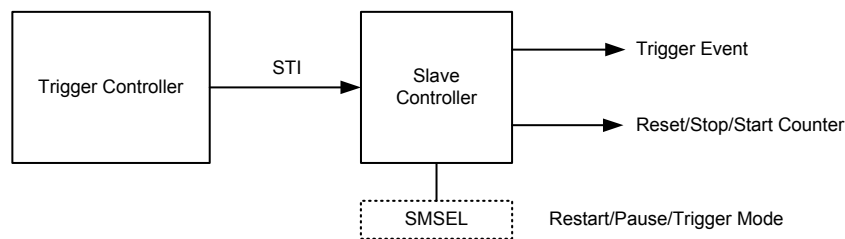


Figure 41. Slave Controller Diagram

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.

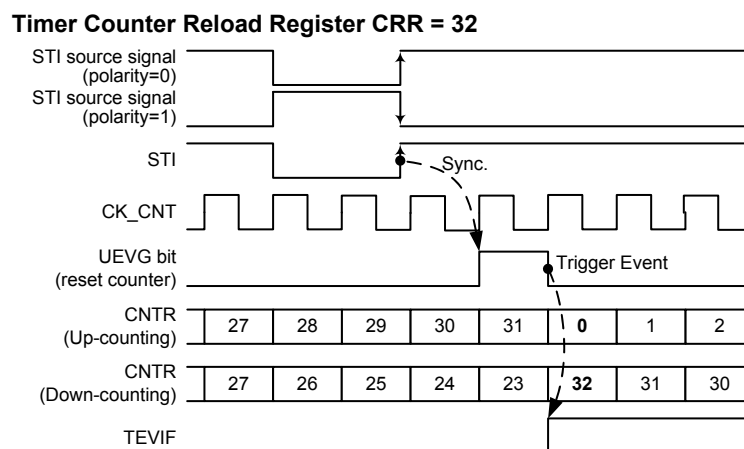


Figure 42. GPTM in Restart Mode

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.

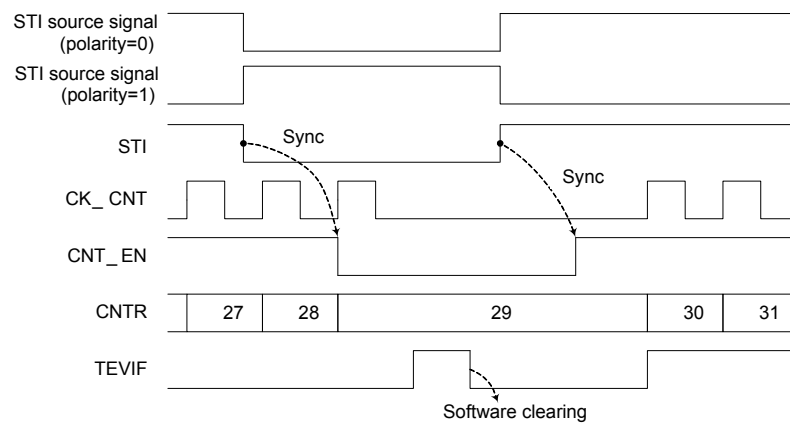


Figure 43. GPTM in Pause Mode

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.

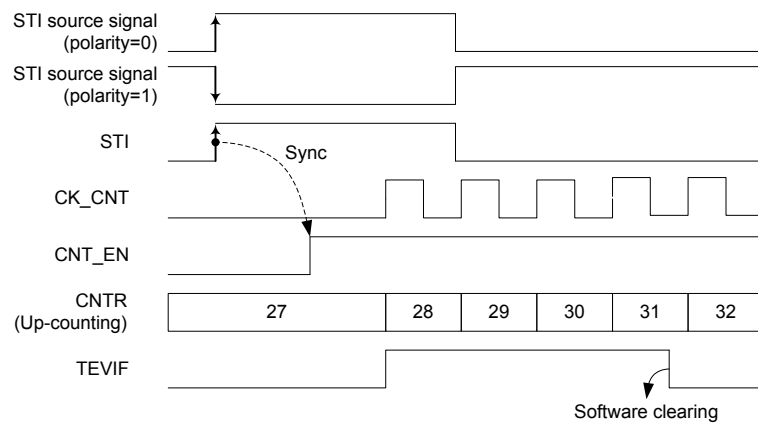


Figure 44. GPTM in Trigger Mode

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.

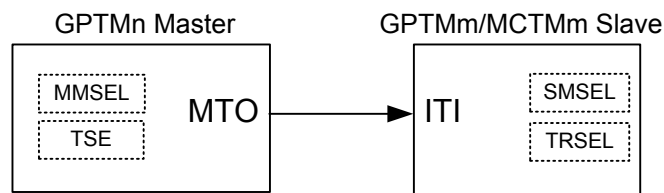


Figure 45. Master GPTMn and Slave GPTMm/MCTMm Connection

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

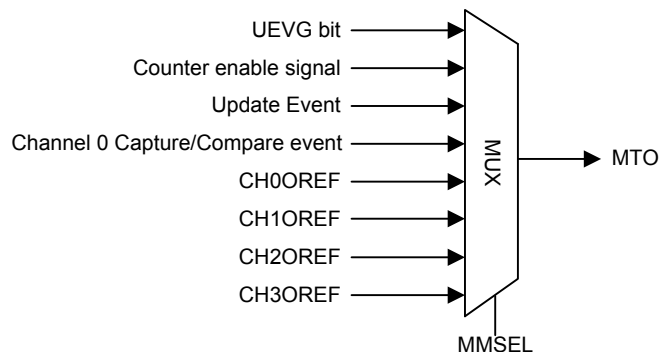


Figure 46. MTO Selection

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 through the read/write 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.

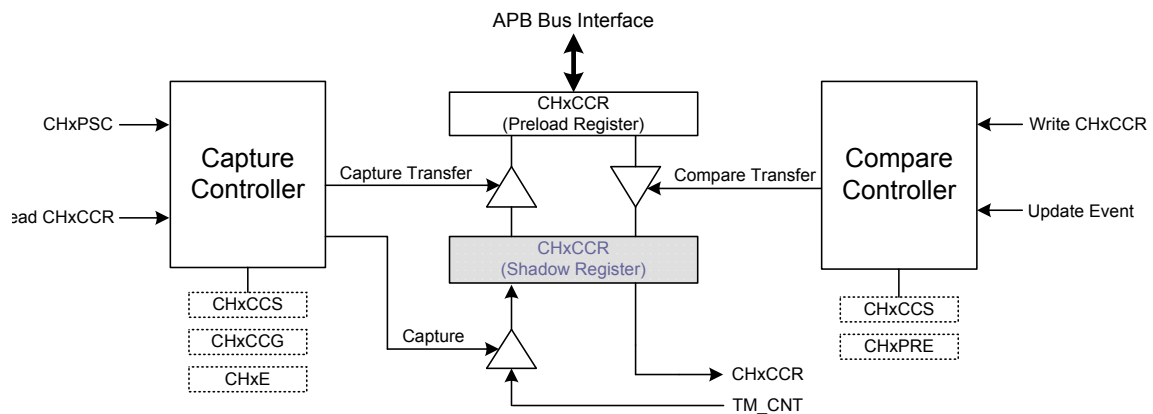


Figure 47. Capture/Compare Block Diagram

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.

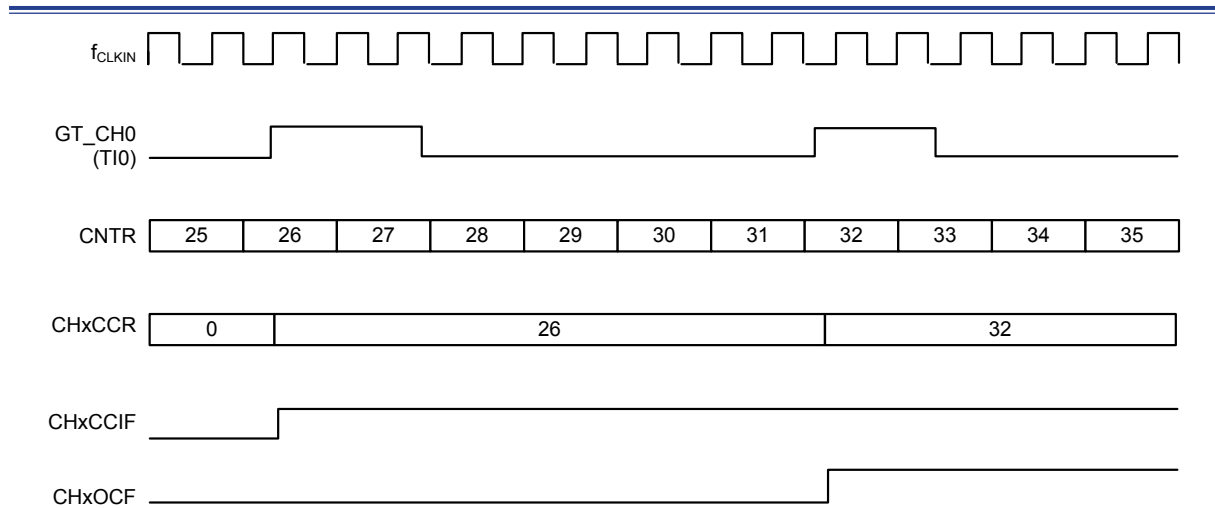


Figure 48. Input Capture Mode

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 0x0001 to select TI0S0 as the trigger input.
- Configure the Slave controller to operate in the Restart mode by setting the SMSEL 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.

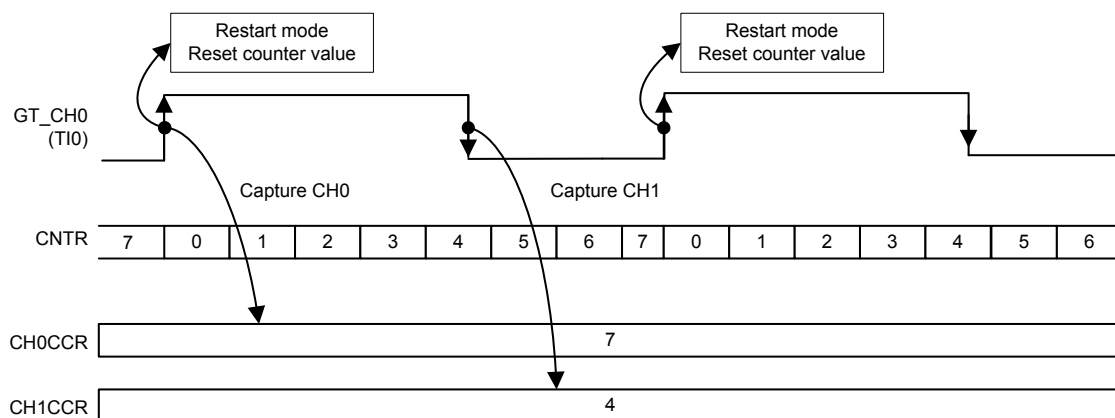


Figure 49. PWM Pulse Width Measurement Example

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 input prescaler (CHxPSC).

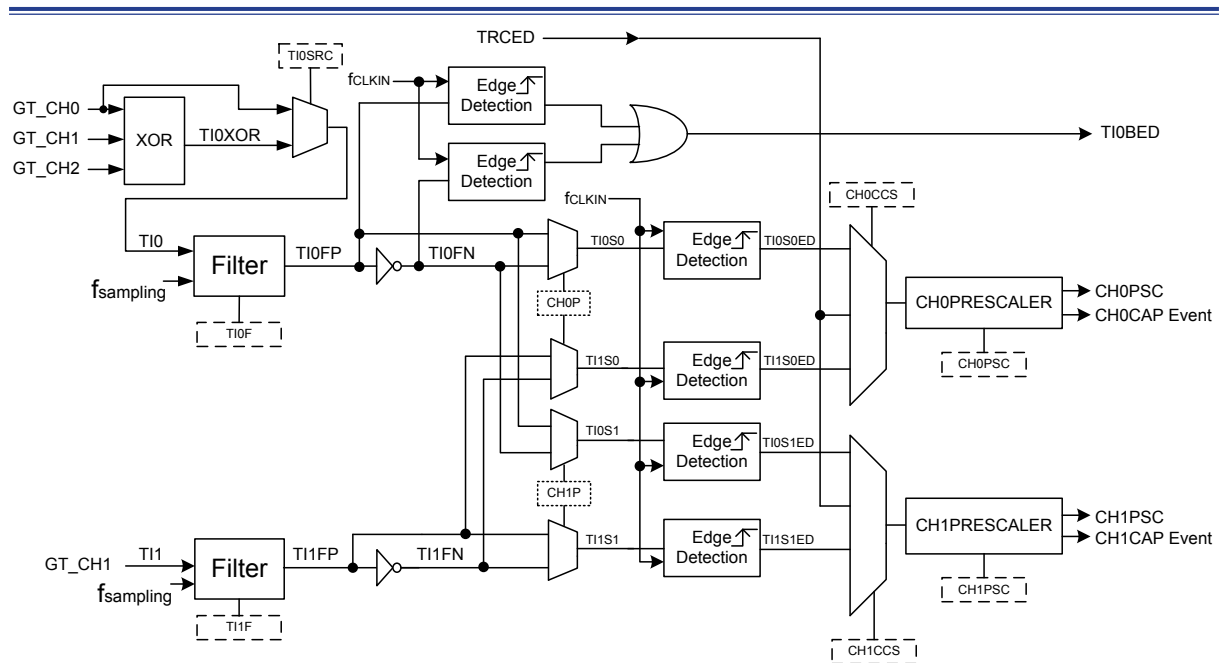


Figure 50. Channel 0 and Channel 1 Input Stages

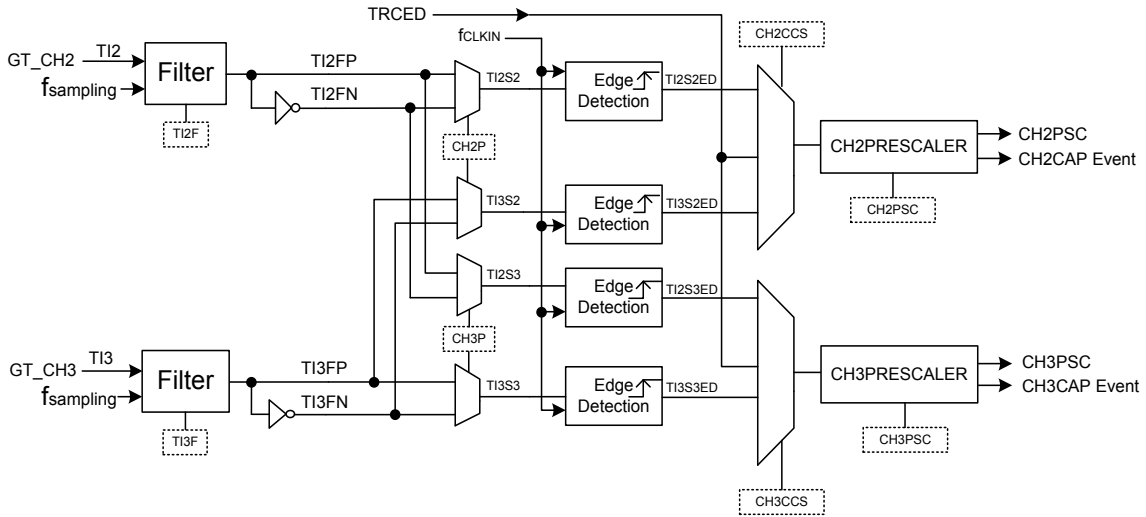


Figure 51. Channel 2 and Channel 3 Input Stages

General-Purpose Timer (GPTM)

Digital Filter

The digital filters are embedded in the input stage for the GT_CH0 ~ GT_CH3 pins respectively. The digital filter in the GPTM is an N-event counter where N refers to how many valid transitions are necessary to output a filtered signal. The N value can be 0, 2, 4, 5, 6 or 8 according to the user selection for each filter.

Digital Filter (N=2)

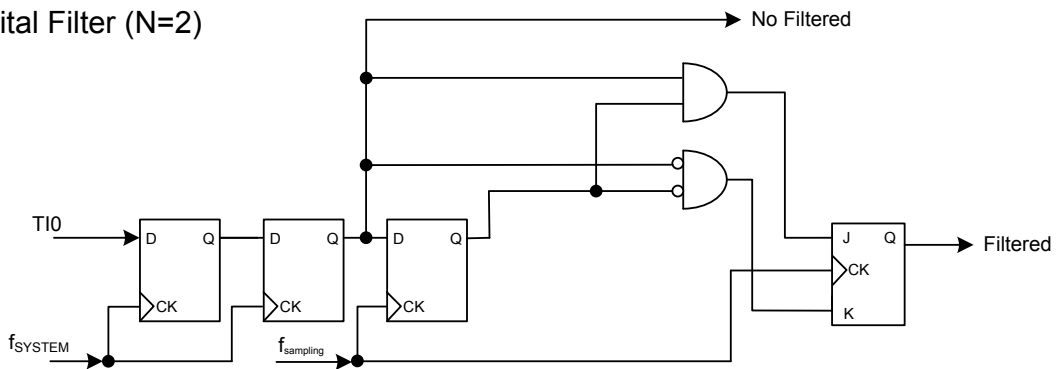


Figure 52. T10 Digital Filter Diagram with N = 2

Quadrature Decoder

The Quadrature Decoder function uses two quadrantal inputs TIO 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 TIO only, TI1 only or both TIO and TI1, the selection made by setting the SMSEL field to 0x01, 0x02 or 0x03. 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.

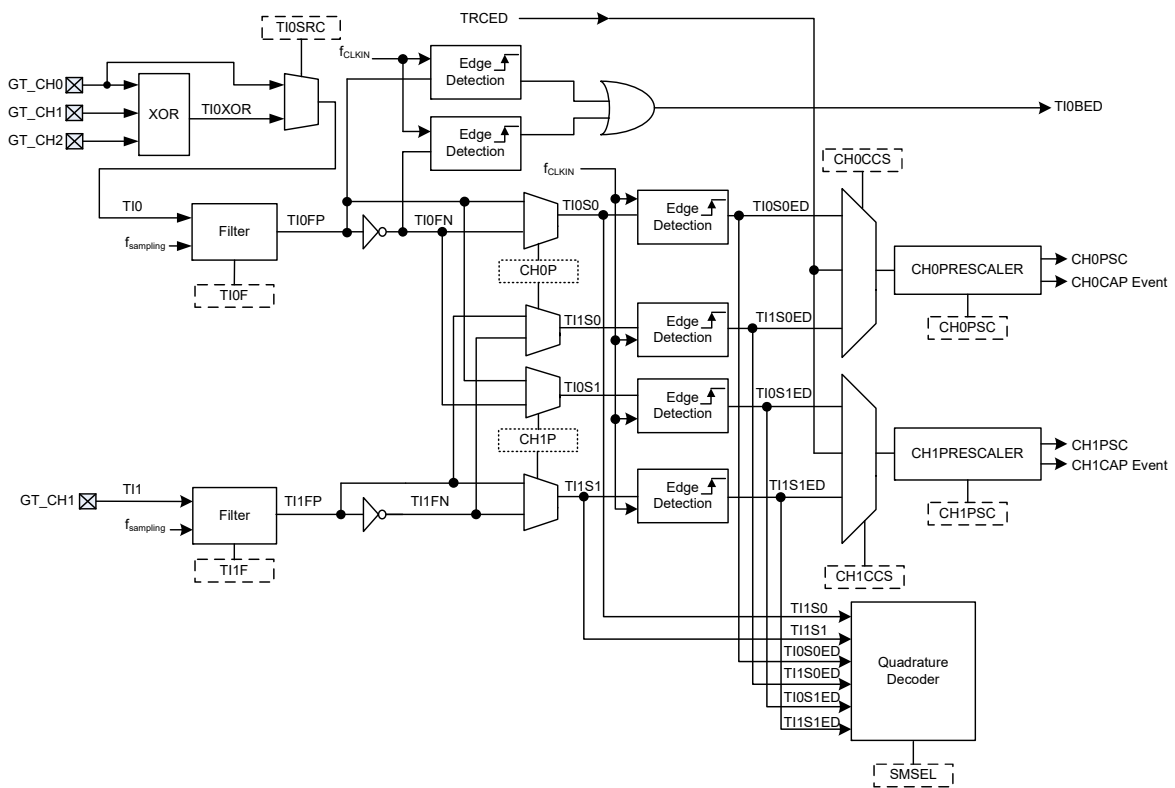


Figure 53. Input Stage and Quadrature Decoder Block Diagram

Table 30. Counting Direction and Encoding Signals

Counting mode	Level	TI0S0		TI1S1	
		Rising	Falling	Rising	Falling
Counting on TI0 only (SMSEL = 0x01)	TI1S1 = High	Down	Up	—	—
	TI1S1 = Low	Up	Down	—	—
Counting on TI1 only (SMSEL = 0x02)	TI0S0 = High	—	—	Up	Down
	TI0S0 = Low	—	—	Down	Up
Counting on TI0 and TI1 (SMSEL = 0x03)	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

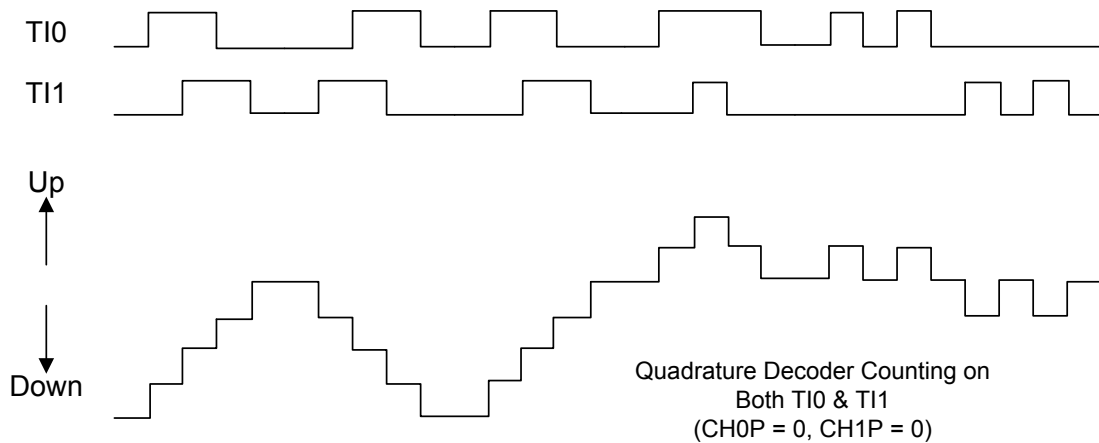


Figure 54. Both TI0 and TI1 Quadrature Decoder Counting

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.

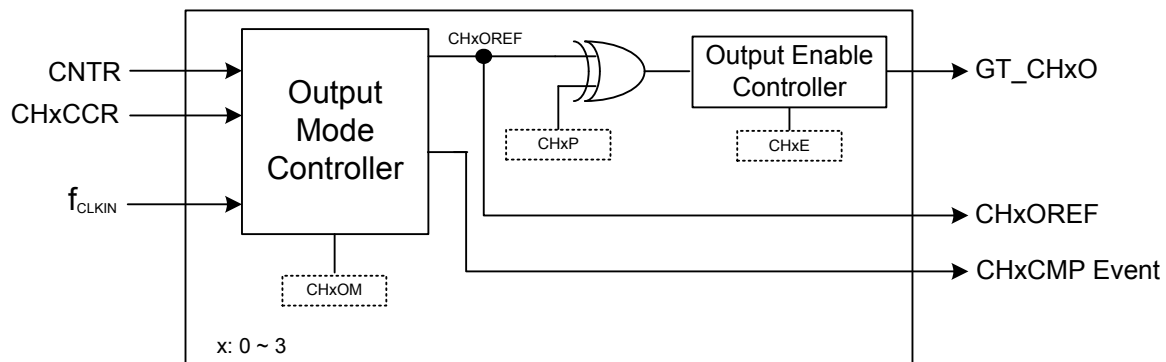


Figure 55. Output Stage Block Diagram

Channel Output Reference Signal

When the GPTM is used in the compare match output mode, the CHxOREF signal (Channel x Output Reference signal) is defined by the CHxOM bit 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 31 shows a summary of the output type setup.

Table 31. Compare Match Output Setup

CHxOM value	Compare Match Level
0x00	No change
0x01	Clear Output to 0
0x02	Set Output to 1
0x03	Toggle Output
0x04	Force Inactive Level
0x05	Force Active Level
0x06	PWM Mode 1
0x07	PWM Mode 2

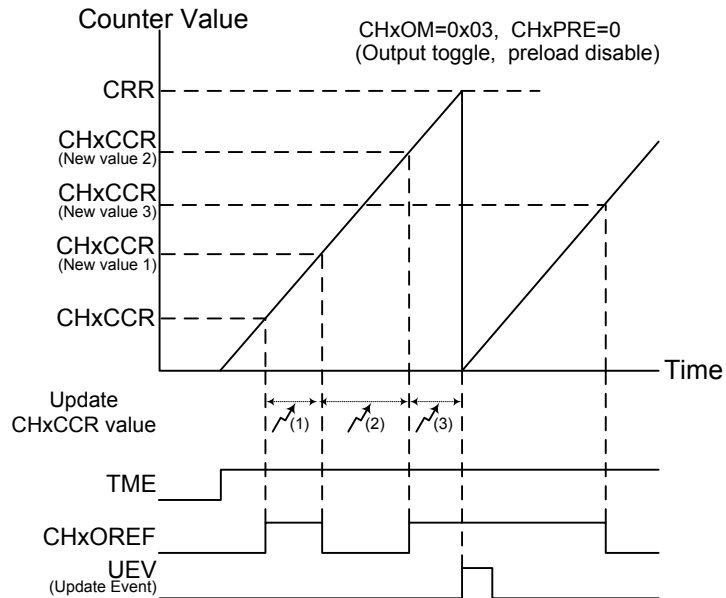


Figure 56. Toggle Mode Channel Output Reference Signal (CHxPRE = 0)

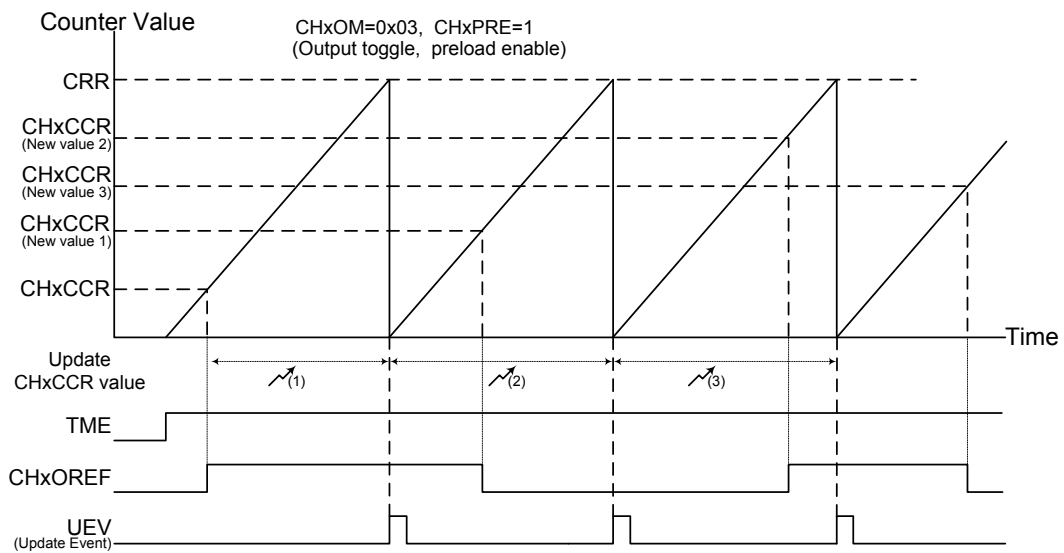


Figure 57. Toggle Mode Channel Output Reference Signal (CHxPRE = 1)

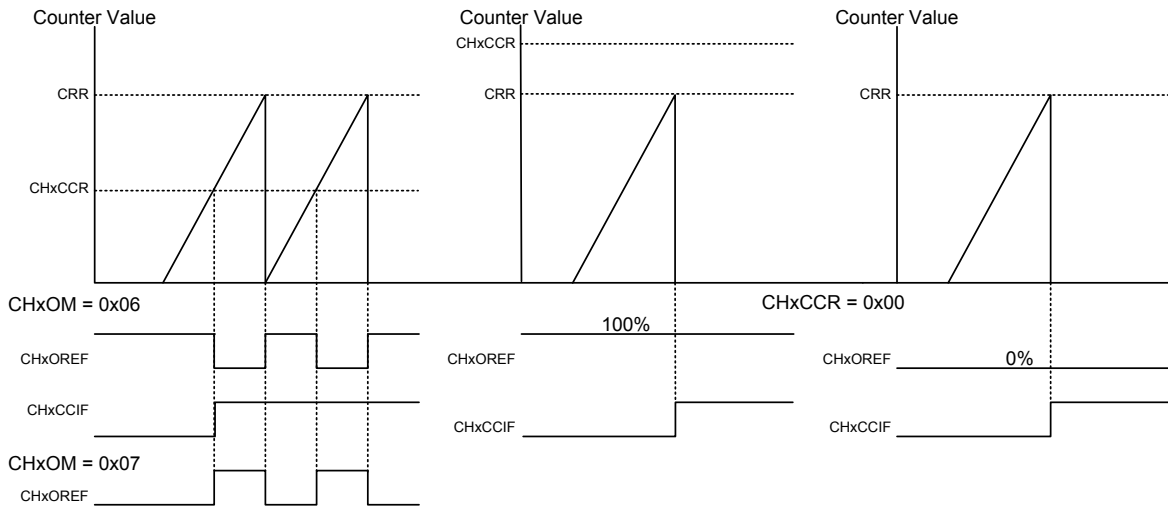


Figure 58. PWM Mode Channel Output Reference Signal and Counter in Up-counting Mode

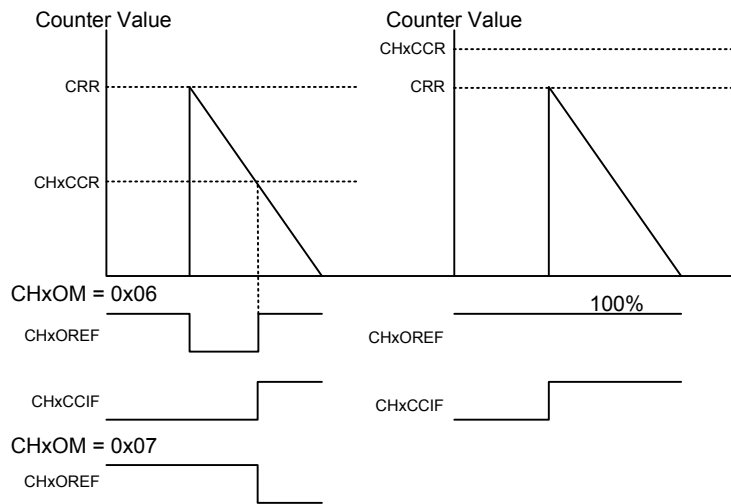


Figure 59. PWM Mode Channel Output Reference Signal and Counter in Down-counting Mode

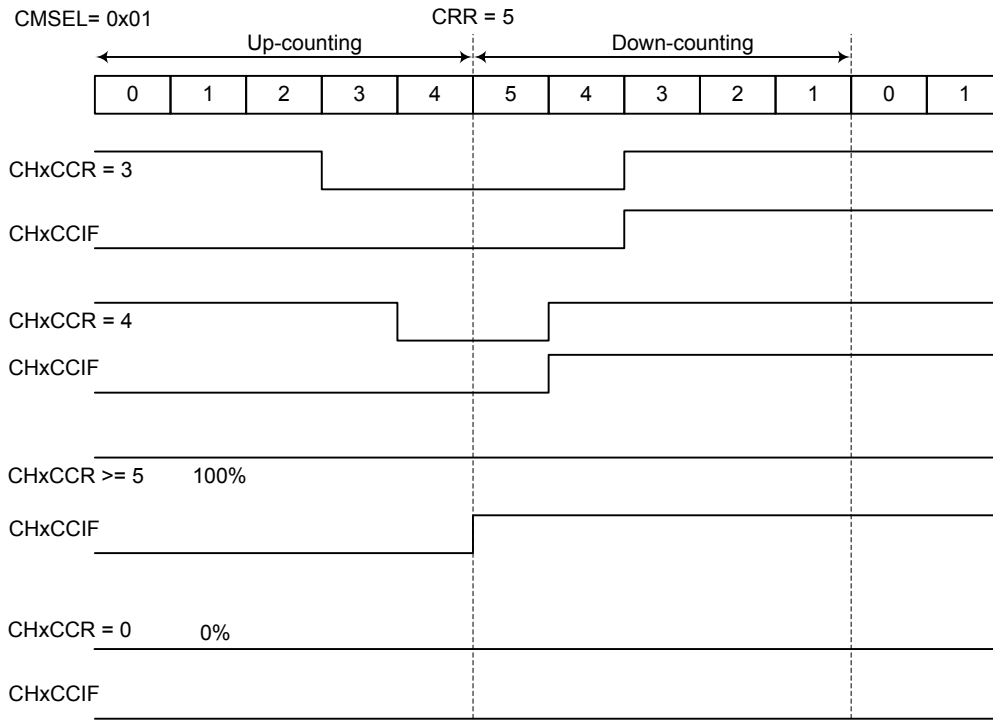


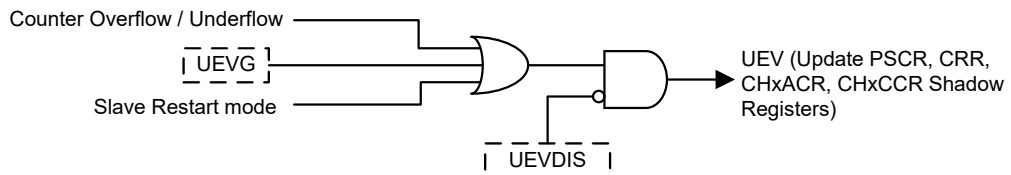
Figure 60. PWM Mode Channel Output Reference Signal and Counter in Centre-align Mode

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 detail description, refer to the UEVDIS and UGDIS bit definition in the CNTCFR register.

Update Event Management



Update Event Interrupt Management

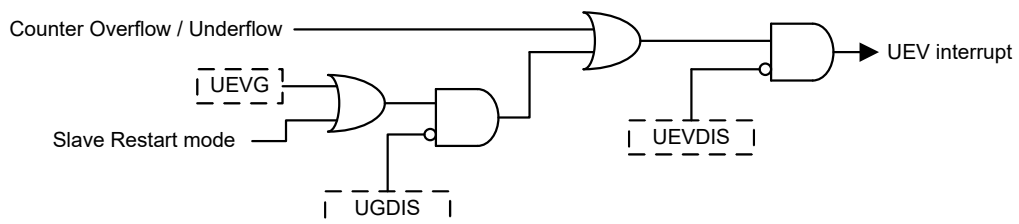


Figure 61. Update Event Setting Diagram

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.

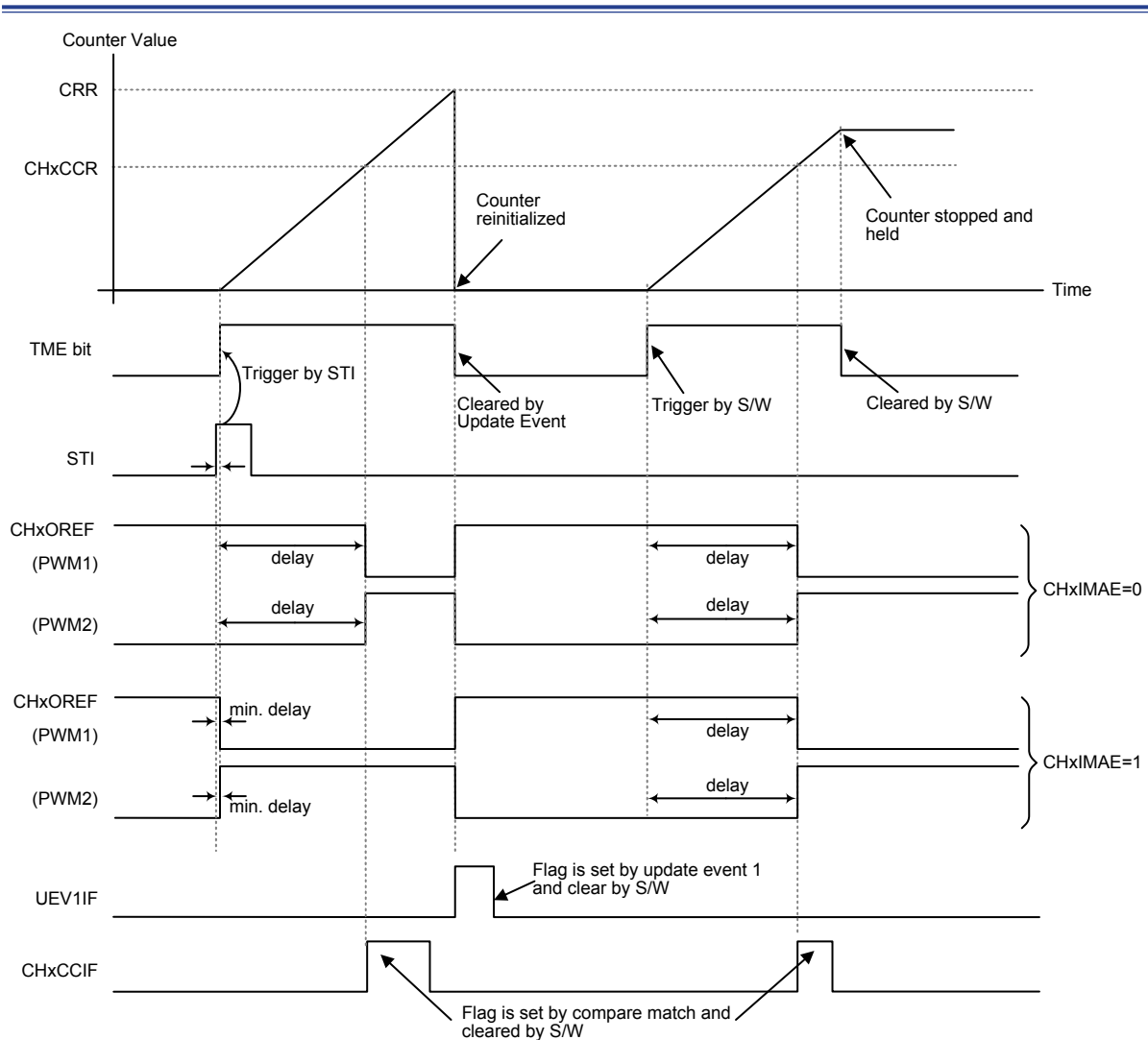


Figure 62. Single Pulse Mode

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 CHxO CFR 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 PWM1 or PWM2 output mode and the trigger source is derived from the STI signal.

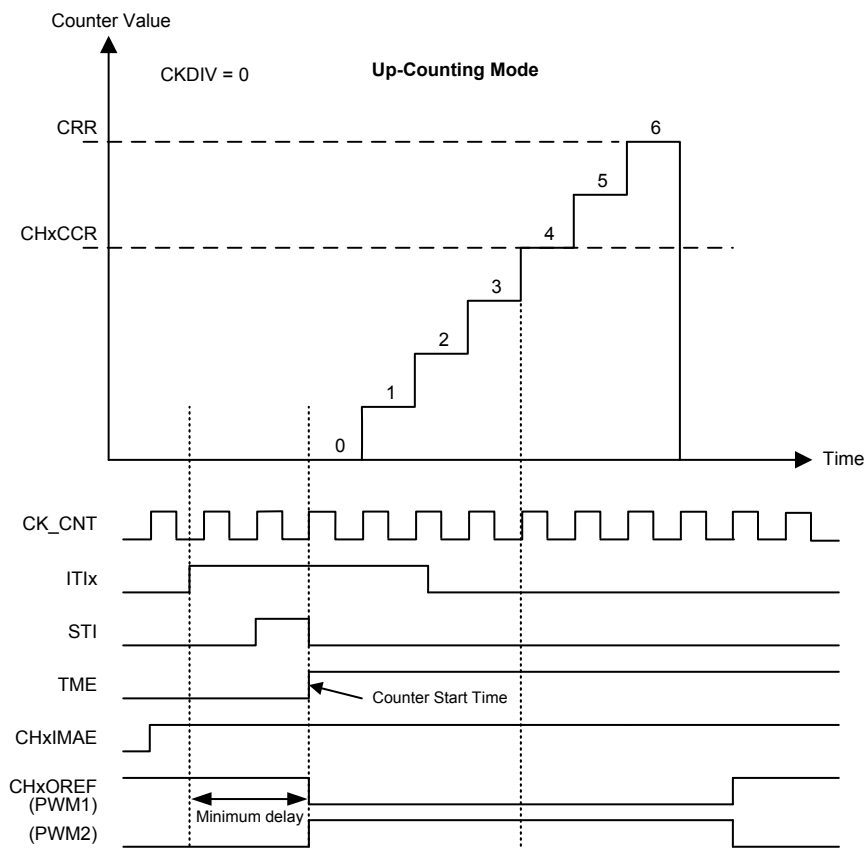


Figure 63. Immediate Active Mode Minimum Delay

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 using the value in CHxCCR as up-count compare value. When the counter is into counting down stage, the PWM using the value in CHxACR as down-count compare value. The Figure 64 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.

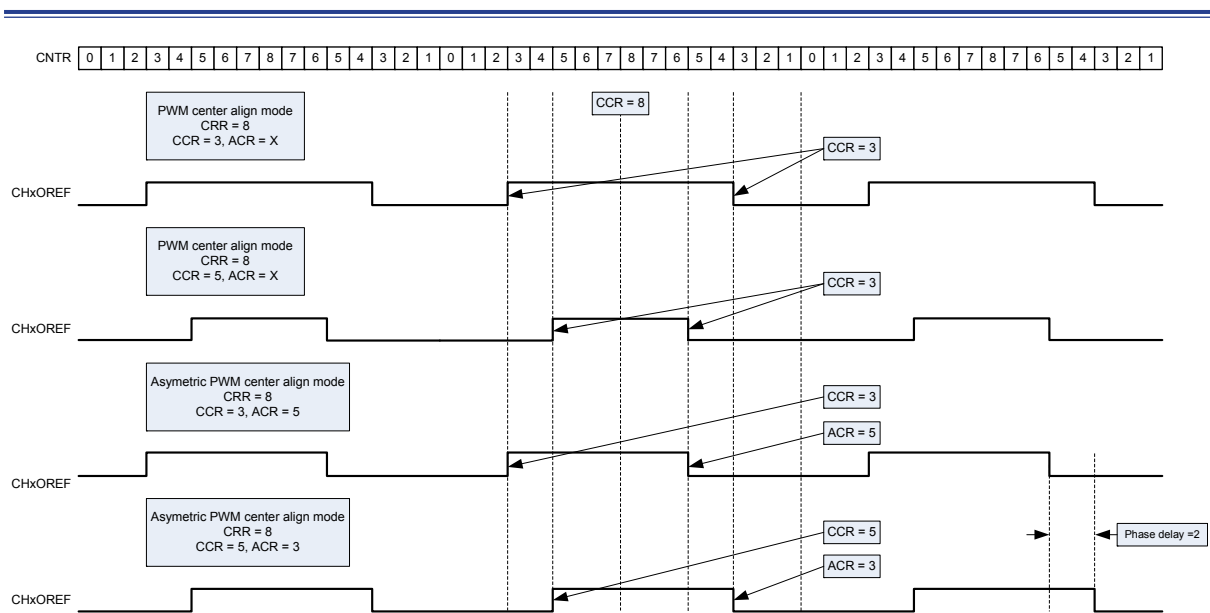


Figure 64. Asymmetric PWM mode versus center align counting mode

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 GPTM0 as the master mode to send its channel 0 Output Reference signal CH0OREF as a trigger output (MMSEL = 0x04).
- Configure GPTM0 CH0OREF waveform.
- Configure GPTM1 to receive its input trigger source from the GPTM0 trigger output (TRSEL=0x09)
- Configure GPTM1 to operate in the pause mode (SMSEL = 0x05).
- Enable GPTM1 by writing "1" to the TME bit.
- Enable GPTM0 by writing "1" to the TME bit.

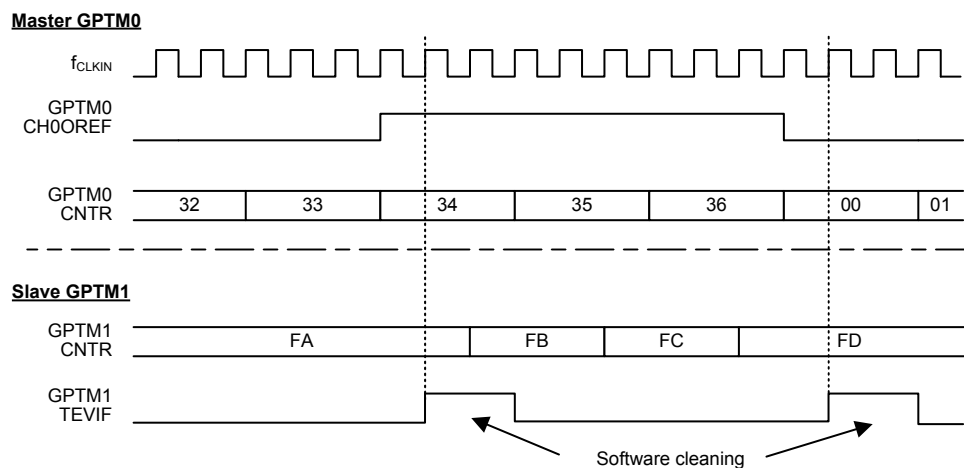


Figure 65. Pausing GPTM1 using the GPTM0 CH0OREF Signal

Using one timer to trigger another timer start counting

- Configure GPTM0 to operate in the master mode to send its Update Event UEV as the trigger output (MMSEL = 0x02).
- Configure the GPTM0 period by setting the CRR register.
- Configure GPTM1 to get the input trigger source from the GPTM0 trigger output (TRSEL=0x09)
- Configure GPTM1 to be in the slave trigger mode (SMSEL = 0x06).
- Start GPTM0 by writing "1" to the TME bit.

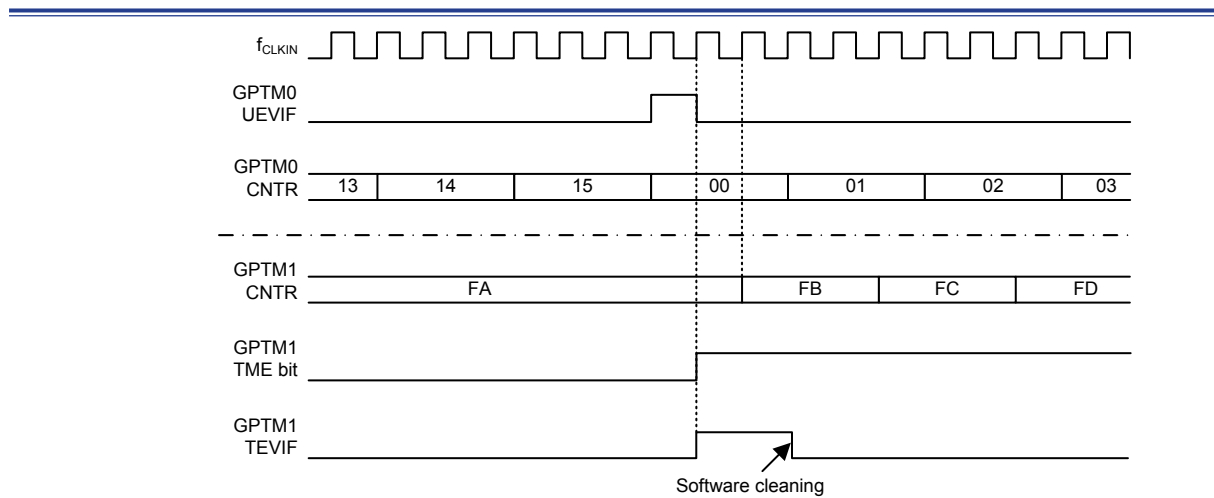
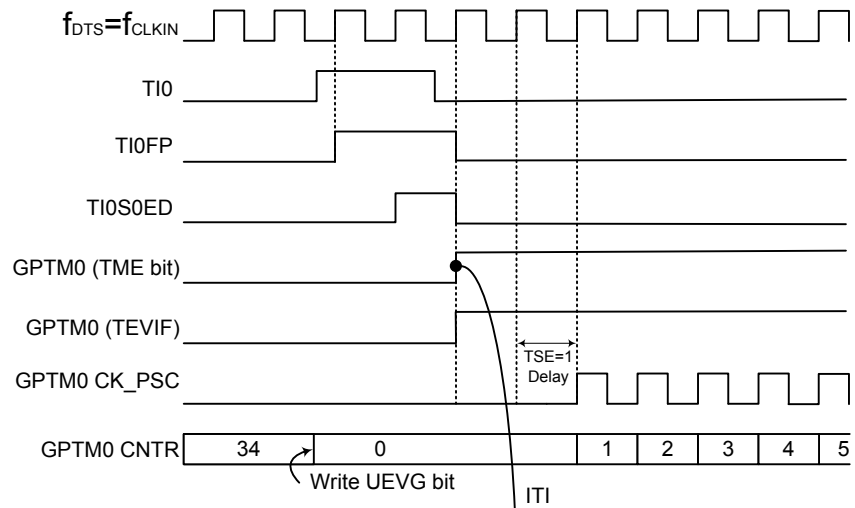


Figure 66. Triggering GPTM1 with GPTM0 Update Event

Starting two timers synchronously in response to an external trigger

- Configure GPTM0 to operate in the master mode to send its enable signal as a trigger output (MMSEL = 0x01).
- Configure GPTM0 slave mode to receive its input trigger source from GTn_CH0 pin (TRSEL= 0x01).
- Configure GPTM0 to be in the slave trigger mode (SMSEL = 0x06).
- Enable the GPTM0 master timer synchronization function by setting the TSE bit in the MDCFR register to 1 to synchronize the slave timer.
- Configure GPTM1 to receive its input trigger source from the GPTM0 trigger output (TRSEL = 0x09).
- Configure GPTM1 to be in the slave trigger mode (SMSEL = 0x06).

Master GPTM0



Slave GPTM1

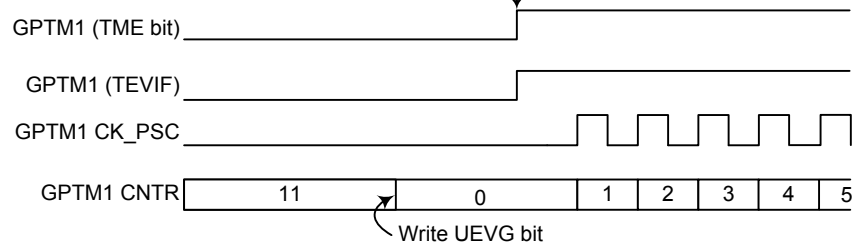


Figure 67. Trigger GPTM0 and GPTM1 with the GPTM0 CH0 Input

Trigger ADC Start

To interconnect to the Analog-to-Digital Converter, the GPTM can output the MTO signal or the channel compare match output signal CHxOREF (x = 0 ~ 3) to be used as an Analog-to-Digital Converter input trigger signal.

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.

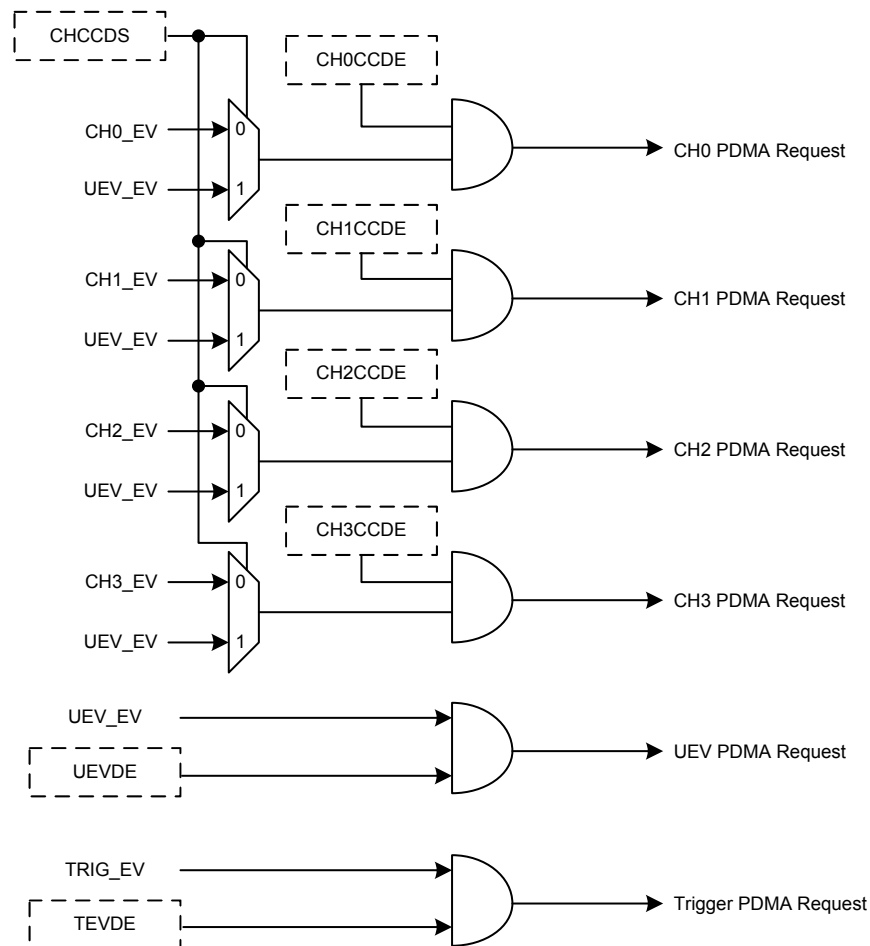


Figure 68. GPTM PDMA Mapping Diagram

Register Map

The following table shows the GPTM registers and reset values.

Table 32. 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	
Type/Reset	Reserved							DIR	0
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved						CMSEL	0	0
	15	14	13	12	11	10	9	8	
Type/Reset	Reserved						CKDIV	0	0
	7	6	5	4	3	2	1	0	
Type/Reset	Reserved						UGDIS	0	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 period.
[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	
Type/Reset	Reserved							SPMSET	0
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved					MMSEL			
						RW	0	RW	0
	15	14	13	12	11	10	9	8	
Type/Reset	Reserved					SMSEL			
						RW	0	RW	0
	7	6	5	4	3	2	1	0	
Type/Reset	Reserved							TSE	0
								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 border="1"> <thead> <tr> <th>MMSEL [2:0]</th> <th>Mode</th> <th>Descriptions</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>Reset Mode</td> <td>The MTO 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>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> </tbody> </table>	MMSEL [2:0]	Mode	Descriptions	000	Reset Mode	The MTO 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	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 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	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.																											

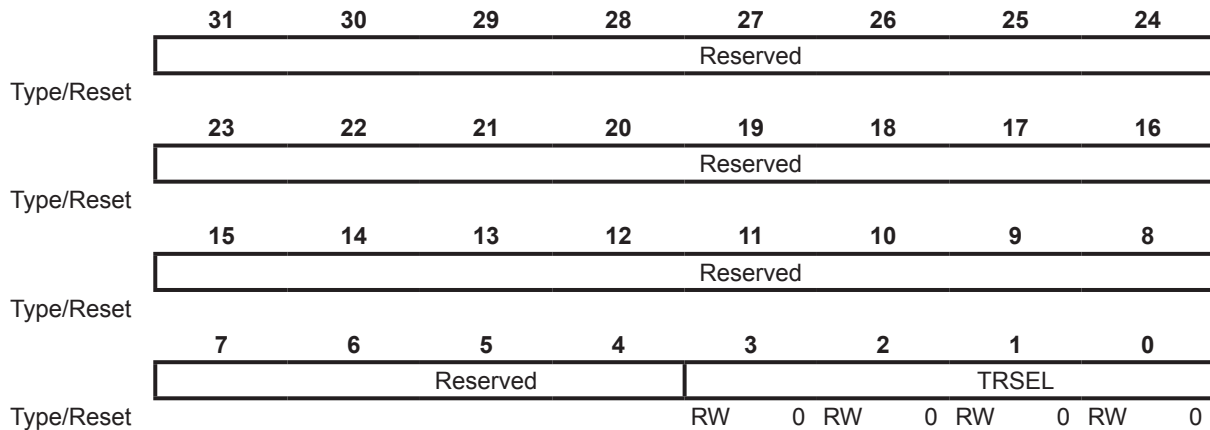
Bits	Field	Descriptions																											
[10:8]	SMSEL	Slave Mode Selection																											
		<table border="1"> <thead> <tr> <th>SMSEL [2:0]</th> <th>Mode</th> <th>Descriptions</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>Disable mode</td> <td>The prescaler is clocked directly 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 T10 and T11 signals to drive the counter prescaler. A transition of the T10 edge is used in this mode depending upon the T11 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 T10 and T11 signals to drive the counter prescaler. A transition of the T11 edge is used in this mode depending upon the T10 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 T10 and T11 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> </tbody> </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 T10 and T11 signals to drive the counter prescaler. A transition of the T10 edge is used in this mode depending upon the T11 level.	010	Quadrature Decoder mode 2	The counter uses the clock pulse generated from the interaction between the T10 and T11 signals to drive the counter prescaler. A transition of the T11 edge is used in this mode depending upon the T10 level.	011	Quadrature Decoder mode 3	The counter uses the clock pulse generated from the interaction between the T10 and T11 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 T10 and T11 signals to drive the counter prescaler. A transition of the T10 edge is used in this mode depending upon the T11 level.																											
010	Quadrature Decoder mode 2	The counter uses the clock pulse generated from the interaction between the T10 and T11 signals to drive the counter prescaler. A transition of the T11 edge is used in this mode depending upon the T10 level.																											
011	Quadrature Decoder mode 3	The counter uses the clock pulse generated from the interaction between the T10 and T11 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	<p>Timer Synchronization Enable</p> <p>0: No action</p> <p>1: Master timer (current timer) will generate a delay to synchronize its slave timer through the MTO signal.</p>																											

Timer Trigger Configuration Register – TRCFR

This register specifies the trigger source selection of GPTM.

Offset: 0x008

Reset value: 0x0000_0000



Bits	Field	Descriptions
[3:0]	TRSEL	<p>Trigger Source Selection</p> <p>These bits are used to select the trigger input (STI) for counter synchronizing.</p> <ul style="list-style-type: none"> 0000: Software Trigger by setting the UEVG bit 0001: Filtered input of channel 0 (TI0S0) 0010: Filtered input of channel 1 (TI1S1) 0011: Reserved 1000: Channel 0 Edge Detector (TI0BED) 1001: Internal Timing Module Trigger 0 (ITI0) 1010: Internal Timing Module Trigger 1 (ITI1) 1011: Internal Timing Module Trigger 2 (ITI2) Others: Default 0 <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 0x00.</p>

Table 33. GPTM Internal Trigger Connection

Slave Timing Module	ITI0	ITI1	ITI2
GPTM0	GPTM1	MCTM	—
GPTM1	GPTM0	MCTM	—

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	
									RW 0
	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]	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
	TI0SRC		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				TI0F			
Type/Reset					RW	0	RW	0

Bits	Field	Descriptions
[31]	TI0SRC	Channel 0 Input Source TI0 Selection 0: The GT_CH0 pin is connected to channel 0 input TI0 1: The XOR operation output of the GT_CH0, GT_CH1, and GT_CH2 pins are connected to the channel 0 input TI0
[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 TI0 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]	TIOF	<p>Channel 0 Input Source TIO Filter Setting</p> <p>These bits define the frequency divided ratio used to sample the TIO 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				T11F			
					RW	0	RW	0
					RW	0	RW	0
					RW	0	RW	0
					RW	0	RW	0

Bits	Field	Descriptions
[19:18]	CH1PSC	Channel 1 Capture Input Source Prescaler Setting 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. 00: No prescaler, channel 1 capture input signal is chosen for each active event 01: Channel 1 Capture input signal is chosen for every 2 events 10: Channel 1 Capture input signal is chosen for every 4 events 11: Channel 1 Capture input signal is chosen for every 8 events
[17:16]	CH1CCS	Channel 1 Capture/Compare Selection 00: Channel 1 is configured as an output 01: Channel 1 is configured as an input derived from the T11 signal 10: Channel 1 is configured as an input derived from the T10 signal 11: Channel 1 is configured as an input which comes from the TRCED signal derived from the Trigger Controller Note: The CH1CCS field can be accessed only when the CH1E bit is cleared to 0.

Bits	Field	Descriptions
[3:0]	T11F	<p>Channel 1 Input Source T11 Filter Setting</p> <p>These bits define the frequency divided ratio used to sample the T11 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 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				CH2PSC		CH2CCS	
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved				TI2F			
					RW	0	RW	0
					RW	0	RW	0
					RW	0	RW	0
					RW	0	RW	0

Bits	Field	Descriptions
[19:18]	CH2PSC	<p>Channel 2 Capture Input Source Prescaler Setting</p> <p>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.</p> <ul style="list-style-type: none"> 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	<p>Channel 2 Capture/Compare Selection</p> <ul style="list-style-type: none"> 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 <p>Note: The CH2CCS field can be accessed only when the CH2E bit is cleared to 0</p>

Bits	Field	Descriptions
[3:0]	T12F	<p>Channel 2 Input Source T12 Filter Setting</p> <p>These bits define the frequency divided ratio used to sample the T12 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
					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 01: Channel 3 Capture input signal is chosen for every 2 events 10: Channel 3 Capture input signal is chosen for every 4 events 11: Channel 3 Capture input signal is chosen for every 8 events</p>
[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>

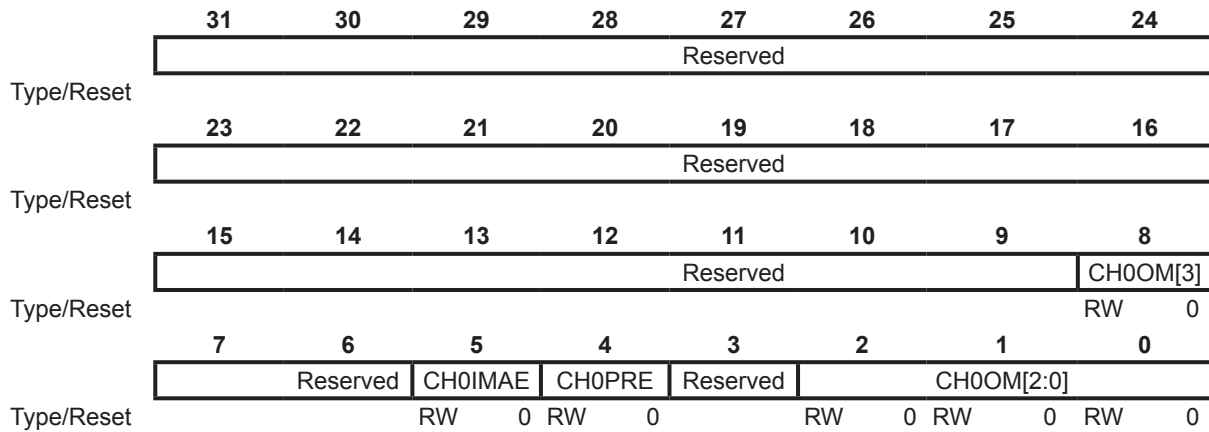
Bits	Field	Descriptions
[3:0]	TI3F	<p>Channel 3 Input Source TI3 Filter Setting</p> <p>These bits define the frequency divided ratio used to sample the TI3 signal. The Digital filter in the GPTM is an N-event counter where N is defined as how many valid transitions are necessary to output a filtered signal.</p> <p>0000: No filter, the sampling clock is f_{SYSTEM}.</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 0 Output Configuration Register – CH0OCFR

This register specifies the channel 0 output mode configuration.

Offset: 0x040

Reset value: 0x0000_0000



Bits	Field	Descriptions
[5]	CH0IMAE	<p>Channel 0 Immediate Active Enable</p> <p>0: No action 1: Single pulse Immediate Active Mode is enabled</p> <p>The CH0OREF 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.</p> <p>The effective duration ends automatically at the next overflow or underflow event.</p> <p>Note: The CH0IMAE bit is available only if the channel 0 is configured to be operated in the PWM mode 1 or the PWM mode 2.</p>
[4]	CH0PRE	<p>Channel 0 Capture/Compare Register (CH0CCR) Preload Enable</p> <p>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.</p> <p>1: CH0CCR preload function is enabled. The new CH0CCR value will not be transferred to its shadow register until the update event occurs.</p>

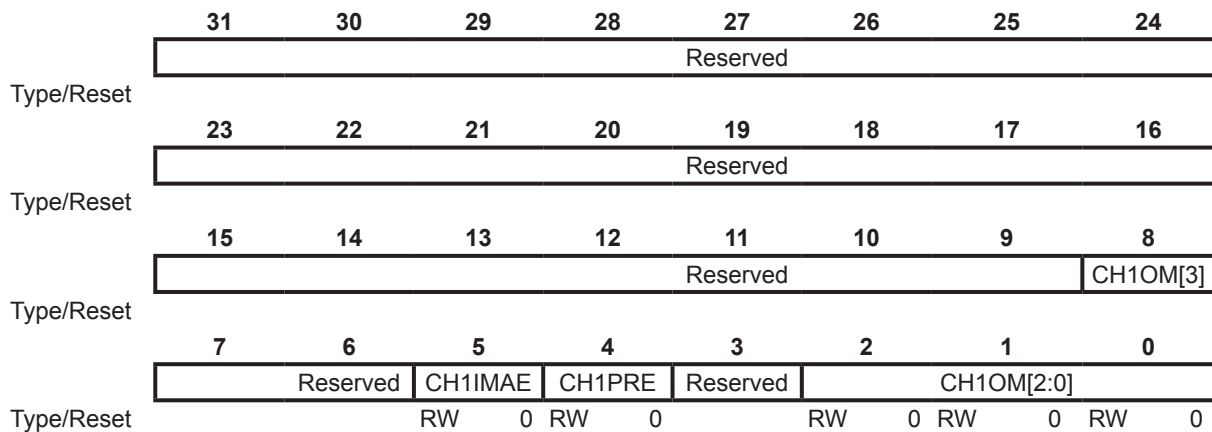
Bits	Field	Descriptions
[8][2:0]	CH0OM[3:0]	<p>Channel 0 Output Mode Setting</p> <p>These bits define the functional types of the output reference signal CH0OREF.</p> <p>0000: No Change</p> <p>0001: Output 0 on compare match</p> <p>0010: Output 1 on compare match</p> <p>0011: Output toggles on compare 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 > 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 < CH0CCR 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 = 01/02/03)</p>

Channel 1 Output Configuration Register – CH1OCFR

This register specifies the channel 1 output mode configuration.

Offset: 0x044

Reset value: 0x0000_0000



Bits	Field	Descriptions
[5]	CH1IMAE	<p>Channel 1 Immediate Active Enable</p> <p>0: No action 1: Single pulse Immediate Active Mode is enabled</p> <p>The CH1OREF 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.</p> <p>The effective duration ends automatically at the next overflow or underflow event.</p> <p>Note: The CH1IMAE bit is available only if the channel 1 is configured to be operated in the PWM mode 1 or the PWM mode 2.</p>
[4]	CH1PRE	<p>Channel 1 Capture/Compare Register (CH1CCR) Preload Enable</p> <p>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.</p> <p>1: CH1CCR preload function is enabled The new CH1CCR value will not be transferred to its shadow register until the update event occurs.</p>

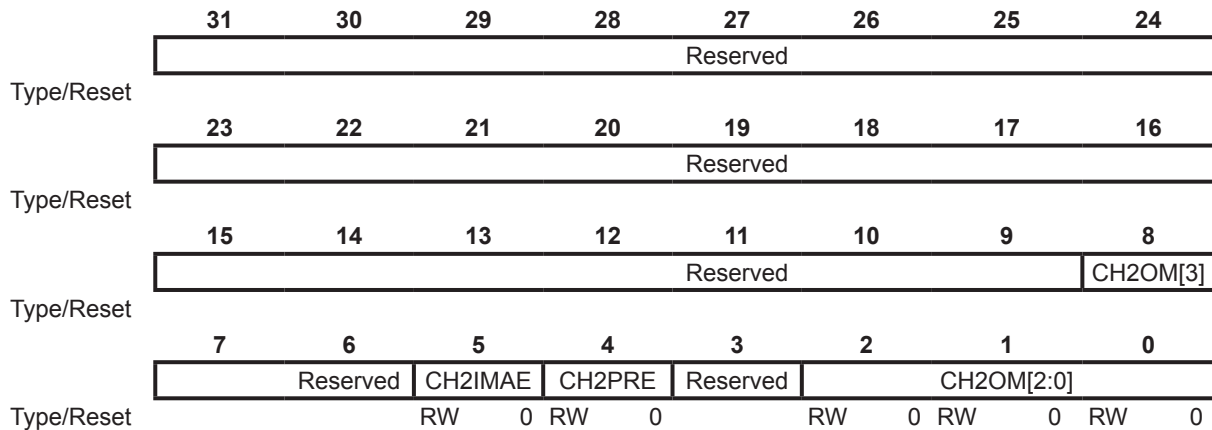
Bits	Field	Descriptions
[8][2:0]	CH1OM[3:0]	<p>Channel 1 Output Mode Setting</p> <p>These bits define the functional types of the output reference signal CH1OREF.</p> <p>0000: No Change</p> <p>0001: Output 0 on compare match</p> <p>0010: Output 1 on compare match</p> <p>0011: Output toggles on compare 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 = 01/02/03)</p>

Channel 2 Output Configuration Register – CH2OCFR

This register specifies the channel 2 output mode configuration.

Offset: 0x048

Reset value: 0x0000_0000



Bits	Field	Descriptions
[5]	CH2IMAE	<p>Channel 2 Immediate Active Enable</p> <p>0: No action 1: Single pulse Immediate Active Mode is enabled</p> <p>The CH2OREF 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.</p> <p>The effective duration ends automatically at the next overflow or underflow event.</p> <p>Note: The CH2IMAE bit is available only if the channel 2 is configured to be operated in the PWM mode 1 or the PWM mode 2.</p>
[4]	CH2PRE	<p>Channel 2 Capture/Compare Register (CH2CCR) Preload Enable</p> <p>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.</p> <p>1: CH2CCR preload function is enabled The new CH2CCR value will not be transferred to its shadow register until the update event occurs.</p>

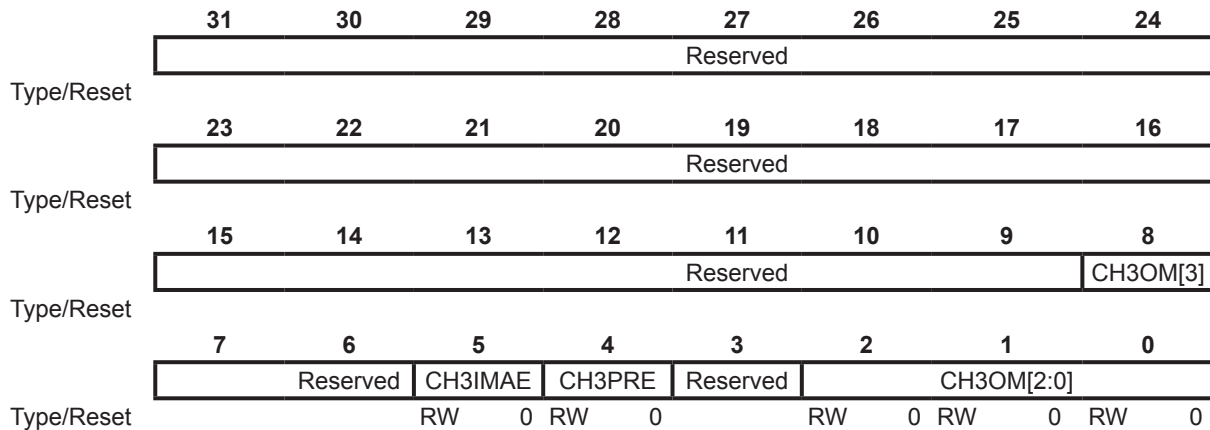
Bits	Field	Descriptions
[8][2:0]	CH2OM[3:0]	<p>Channel 2 Output Mode Setting</p> <p>These bits define the functional types of the output reference signal CH2OREF.</p> <p>0000: No Change</p> <p>0001: Output 0 on compare match</p> <p>0010: Output 1 on compare match</p> <p>0011: Output toggles on compare 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 = 01/02/03)</p>

Channel 3 Output Configuration Register – CH3OCFR

This register specifies the channel 3 output mode configuration.

Offset: 0x04C

Reset value: 0x0000_0000



Bits	Field	Descriptions
[5]	CH3IMAE	<p>Channel 3 Immediate Active Enable</p> <p>0: No action 1: Single pulse Immediate Active Mode is enabled</p> <p>The CH3OREF 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.</p> <p>The effective duration ends automatically at the next overflow or underflow event.</p> <p>Note: The CH3IMAE bit is available only if the channel 3 is configured to be operated in the PWM mode 1 or the PWM mode 2.</p>
[4]	CH3PRE	<p>Channel 3 Capture/Compare Register (CH3CCR) Preload Enable</p> <p>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.</p> <p>1: CH3CCR preload function is enabled The new CH3CCR value will not be transferred to its shadow register until the update event occurs.</p>

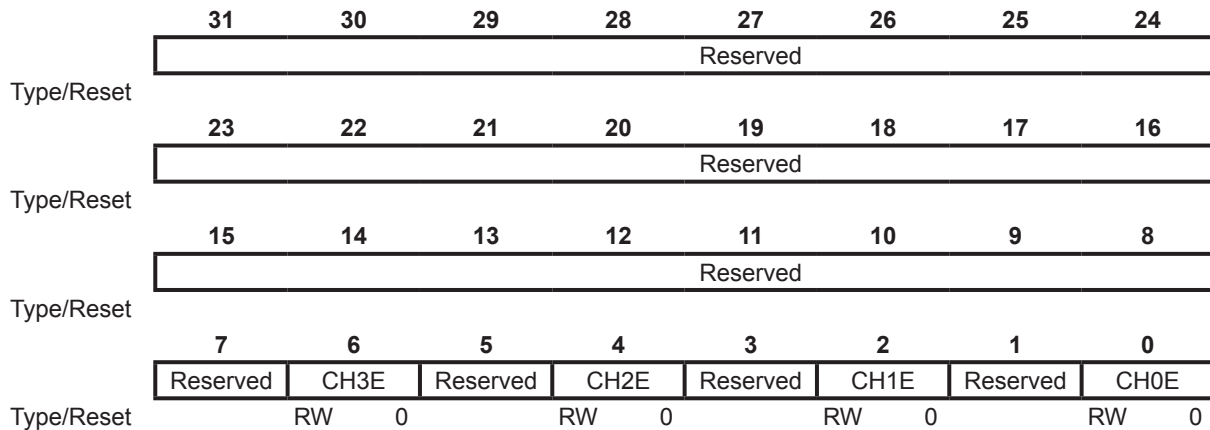
Bits	Field	Descriptions
[8][2:0]	CH3OM[3:0]	<p>Channel 3 Output Mode Setting</p> <p>These bits define the functional types of the output reference signal CH3OREF</p> <p>0000: No Change</p> <p>0001: Output 0 on compare match</p> <p>0010: Output 1 on compare match</p> <p>0011: Output toggles on compare 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 = 01/02/03)</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



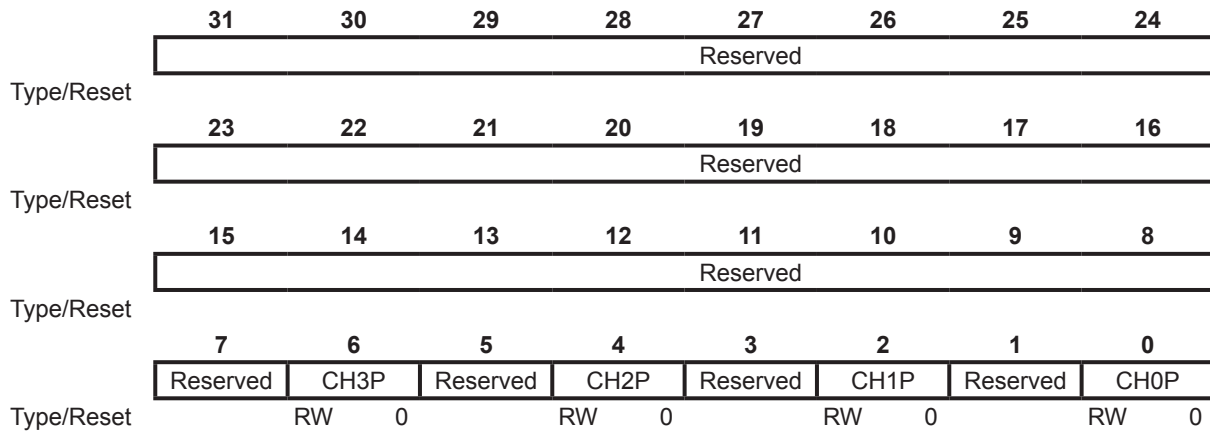
Bits	Field	Descriptions
[6]	CH3E	<p>Channel 3 Capture/Compare Enable</p> <ul style="list-style-type: none"> - Channel 3 is configured as an input (CH3CCS = 0x01/0x02/0x03) <ul style="list-style-type: none"> 0: Input Capture Mode is disabled 1: Input Capture Mode is enabled - Channel 3 is configured as an output (CH3CCS = 0x00) <ul style="list-style-type: none"> 0: Off –Channel 3 output signal CH3O is not active 1: On –Channel 3 output signal CH3O 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 = 0x01/0x02/0x03) <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 = 0x00) <ul style="list-style-type: none"> 0: Off –Channel 2 output signal CH2O is not active 1: On –Channel 2 output signal CH2O 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 = 0x01/0x02/0x03) <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 = 0x00) <ul style="list-style-type: none"> 0: Off –Channel 1 output signal CH1O is not active 1: On –Channel 1 output signal CH1O generated on the corresponding output pin
[0]	CH0E	<p>Channel 0 Capture/Compare Enable</p> <ul style="list-style-type: none"> - Channel 0 is configured as an input (CH0CCS = 0x01/0x02/0x03) <ul style="list-style-type: none"> 0: Input Capture Mode is disabled 1: Input Capture Mode is enabled - Channel 0 is configured as an output (CH0CCS = 0x00) <ul style="list-style-type: none"> 0: Off –Channel 0 output signal CH0O is not active 1: On –Channel 0 output signal CH0O 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



Bits	Field	Descriptions
[6]	CH3P	Channel 3 Capture/Compare Polarity - When Channel 3 is configured as an input 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 = 0x00) 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 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 = 0x00) 0: Channel 2 Output is active high 1: Channel 2 Output is active low
[2]	CH1P	Channel 1 Capture/Compare Polarity - When Channel 1 is configured as an input 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 = 0x00) 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 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 = 0x00) 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
Type/Reset	Reserved					TEVDE	Reserved	UEVDE
						RW 0		RW 0
	23	22	21	20	19	18	17	16
Type/Reset	Reserved				CH3CCDE	CH2CCDE	CH1CCDE	CH0CCDE
					RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
Type/Reset	Reserved					TEVIE	Reserved	UEVIE
						RW 0		RW 0
	7	6	5	4	3	2	1	0
Type/Reset	Reserved				CH3CCIE	CH2CCIE	CH1CCIE	CH0CCIE
					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
[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

Bits	Field	Descriptions
[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					WO	0	Reserved	UEVG	WO
	7	6	5	4	3	2	1	0		
Type/Reset	Reserved				CH3CCG	CH2CCG	CH1CCG	CH0CCG		
					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 detail 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	<p>Channel 1 Capture/Compare Generation</p> <p>A Channel 1 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 1</p> <p>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.</p>
[0]	CH0CCG	<p>Channel 0 Capture/Compare Generation</p> <p>A Channel 0 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 0</p> <p>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.</p>

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
						W0C	0	W0C
						0		0
Type/Reset								
7	6	5	4	3	2	1	0	
CH3OCF	CH2OCF	CH1OCF	CH0OCF	CH3CCIF	CH2CCIF	CH1CCIF	CH0CCIF	
W0C	0	W0C	0	W0C	0	W0C	0	
W0C	0	W0C	0	W0C	0	W0C	0	
Type/Reset								

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
[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.

Bits	Field	Descriptions
[4]	CH0OCF	<p>Channel 0 Over-Capture Flag</p> <p>This flag is set by hardware and cleared by software.</p> <p>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.</p>
[3]	CH3CCIF	<p>Channel 3 Capture/Compare Interrupt Flag</p> <p>- Channel 3 is configured as an output:</p> <p>0: No match event occurs 1: The contents of the counter CNTR have matched the contents of the CH3CCR register.</p> <p>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.</p> <p>- Channel 3 is configured as an input:</p> <p>0: No input capture occurs 1: Input capture occurs</p> <p>This bit is set by hardware on a capture event. It is cleared by software or by reading the CH3CCR register.</p>
[2]	CH2CCIF	<p>Channel 2 Capture/Compare Interrupt Flag</p> <p>- Channel 2 is configured as an output:</p> <p>0: No match event occurs 1: The contents of the counter CNTR have matched the contents of the CH2CCR register</p> <p>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.</p> <p>- Channel 2 is configured as an input:</p> <p>0: No input capture occurs 1: Input capture occurs.</p> <p>This bit is set by hardware on a capture event. It is cleared by software or by reading the CH2CCR register.</p>
[1]	CH1CCIF	<p>Channel 1 Capture/Compare Interrupt Flag</p> <p>- Channel 1 is configured as an output:</p> <p>0: No match event occurs 1: The contents of the counter CNTR have matched the contents of the CH1CCR register</p> <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> <p>- Channel 1 is configured as an input:</p> <p>0: No input capture occurs 1: Input capture occurs</p> <p>This bit is set by hardware on a capture event. It is cleared by software or by reading the CH1CCR register.</p>

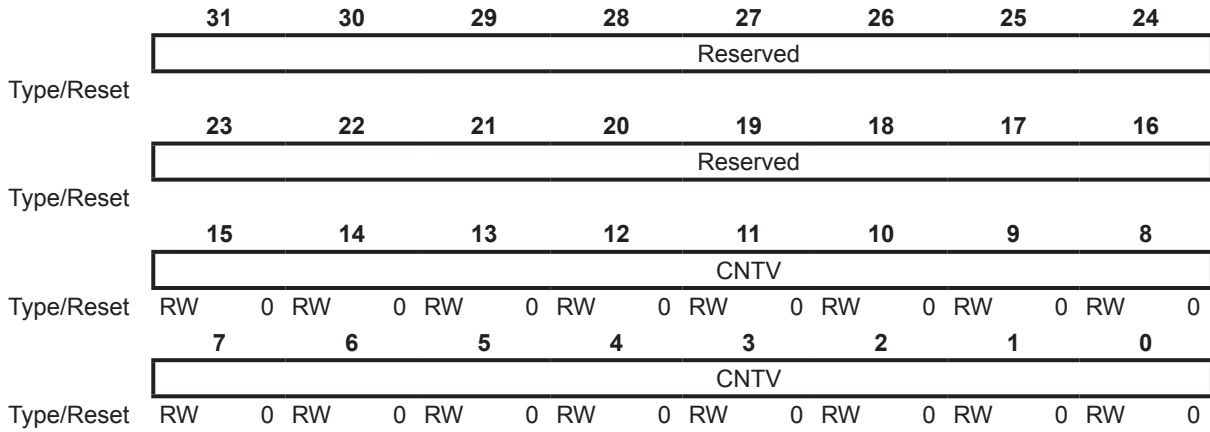
Bits	Field	Descriptions
[0]	CH0CCIF	<p>Channel 0 Capture/Compare Interrupt Flag</p> <ul style="list-style-type: none">- Channel 0 is configured as an output:<ul style="list-style-type: none">0: No match event occurs1: The contents of the counter CNTR have matched the content of the CH0CCR registerThis flag is set by hardware when the counter value matches the CH0CCR value except in the center-aligned mode. It is cleared by software.- Channel 0 is configured as an input:<ul style="list-style-type: none">0: No input capture occurs1: Input capture occursThis bit is set by hardware on a capture event. It is cleared by software or by reading the CH0CCR register.

Timer Counter Register – CNTR

This register stores the timer counter value.

Offset: 0x080

Reset value: 0x0000_0000



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

General-Purpose Timer (GPTM)

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	Reserved							
Type/Reset																
	23	22	21	20	19	18	17	16	Reserved							
Type/Reset																
	15	14	13	12	11	10	9	8	PSCV							
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	PSCV							
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

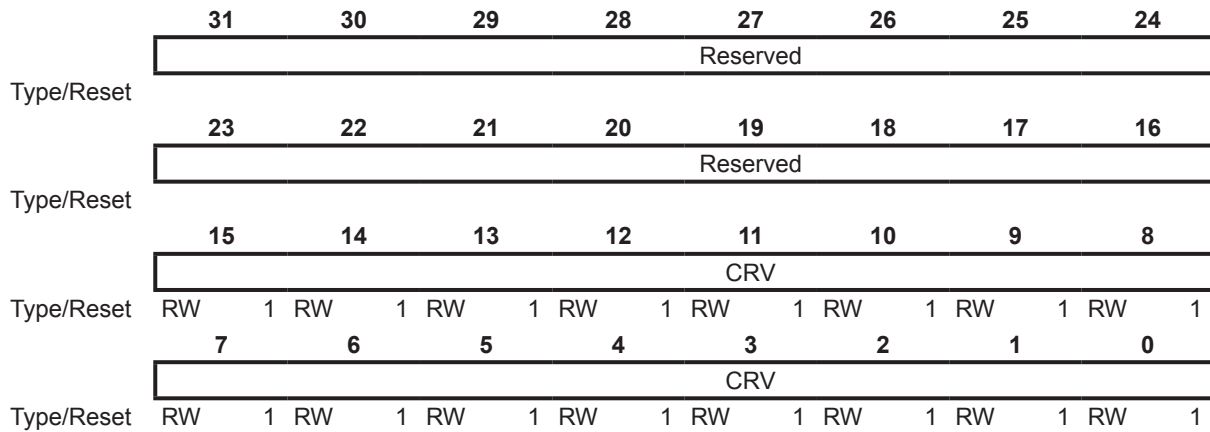
Bits	Field	Descriptions
[15:0]	PSCV	<p>Prescaler Value</p> <p>These bits are used to specify the prescaler value to generate the counter clock frequency f_{CK_CNT}.</p> $f_{CK_CNT} = \frac{f_{CK_PSC}}{PSCV[15:0] + 1}$, where the f_{CK_PSC} is the prescaler clock source.

Timer Counter Reload Register – CRR

This register specifies the timer counter reload value.

Offset: 0x088

Reset value: 0x0000_FFFF



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	Reserved							
Type/Reset																
	23	22	21	20	19	18	17	16	Reserved							
Type/Reset																
	15	14	13	12	11	10	9	8	CH0CCV							
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	CH0CCV							
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

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	Reserved							
Type/Reset																
	23	22	21	20	19	18	17	16	Reserved							
Type/Reset																
	15	14	13	12	11	10	9	8	CH1CCV							
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	CH1CCV							
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

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	Reserved							
Type/Reset																
	23	22	21	20	19	18	17	16	Reserved							
Type/Reset																
	15	14	13	12	11	10	9	8	CH2CCV							
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	CH2CCV							
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[15:0]	CH2CCV	Channel 2 Capture/Compare Value - 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	Reserved							
Type/Reset																
	23	22	21	20	19	18	17	16	Reserved							
Type/Reset																
	15	14	13	12	11	10	9	8	CH3CCV							
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	CH3CCV							
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

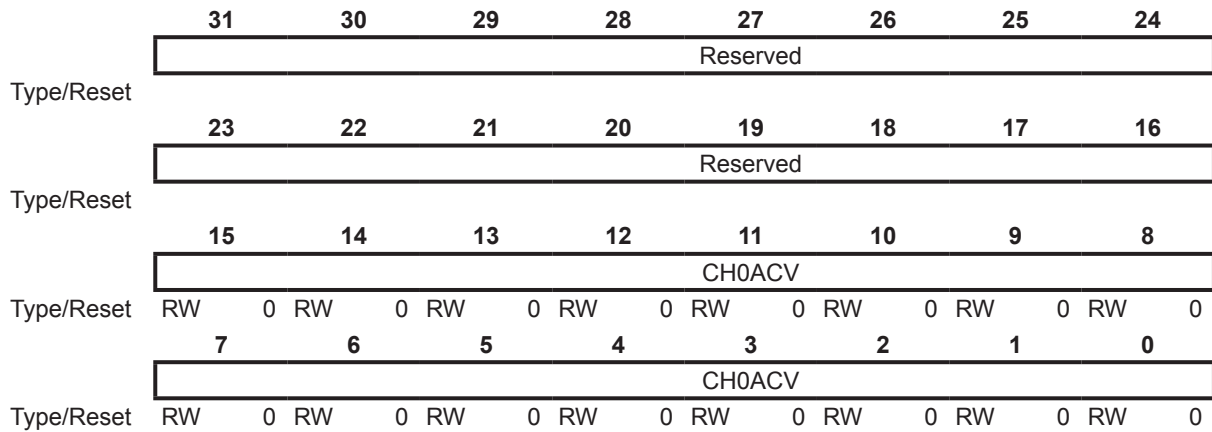
Bits	Field	Descriptions
[15:0]	CH3CCV	<p>Channel 3 Capture/Compare Value</p> <ul style="list-style-type: none"> - When Channel 3 is configured as an output The CH3CCR value is compared with the counter value and the comparison result is used to trigger the CH3OREF output signal. - When Channel 3 is configured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



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	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0								
Type/Reset	CH1ACV															
	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

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	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0								
Type/Reset	CH2ACV															
	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

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	Reserved							
Type/Reset																
	23	22	21	20	19	18	17	16	Reserved							
Type/Reset																
	15	14	13	12	11	10	9	8	CH3ACV							
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	CH3ACV							
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

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.

15 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 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.

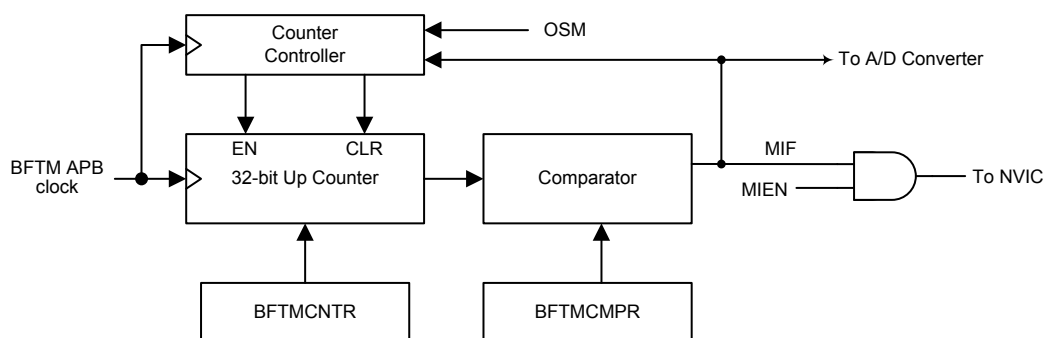


Figure 69. BFTM Block Diagram

Features

- 32-bit up-counting counter
- Compare Match function
- Includes debug mode
- Clock source: BFTM APB clock
- Counter value can be R/W 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.

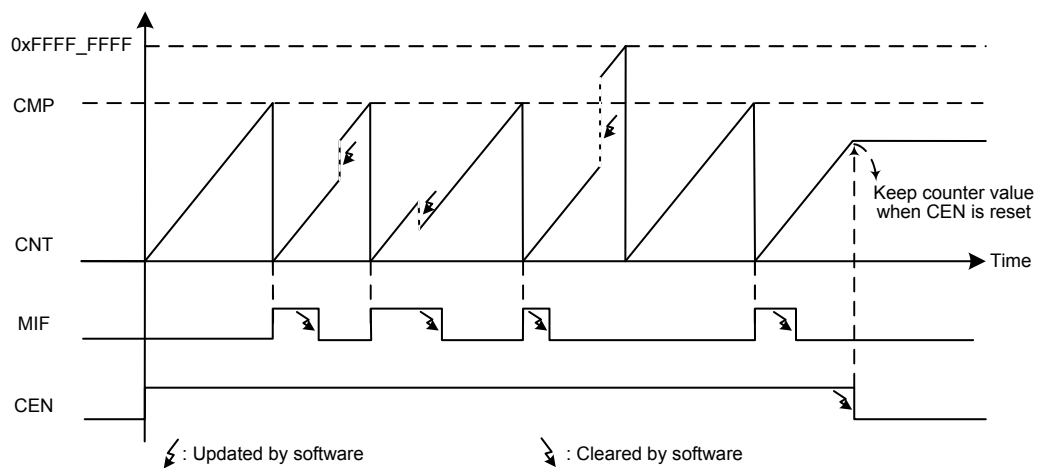


Figure 70. BFTM – Repetitive Mode

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.

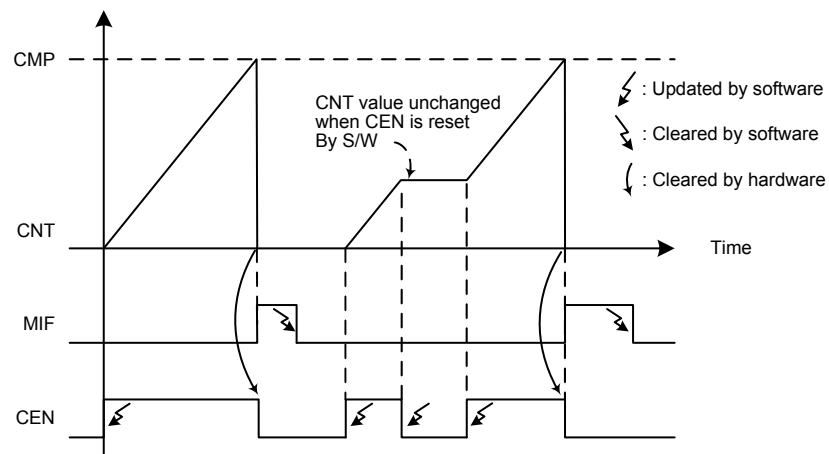


Figure 71. BFTM – One Shot Mode

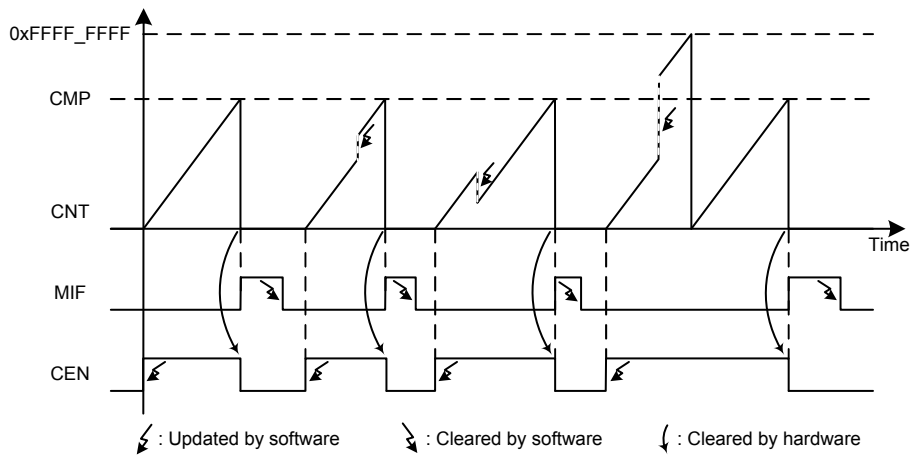


Figure 72. BFTM – One Shot Mode Counter Updating

Trigger ADC Start

When a BFTM compare match event occurs, a compare match interrupt flag, MIF, will be generated which can be used as an A/D Converter input trigger source.

Register Map

The following table shows the BFTM registers and their reset values.

Table 34. 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

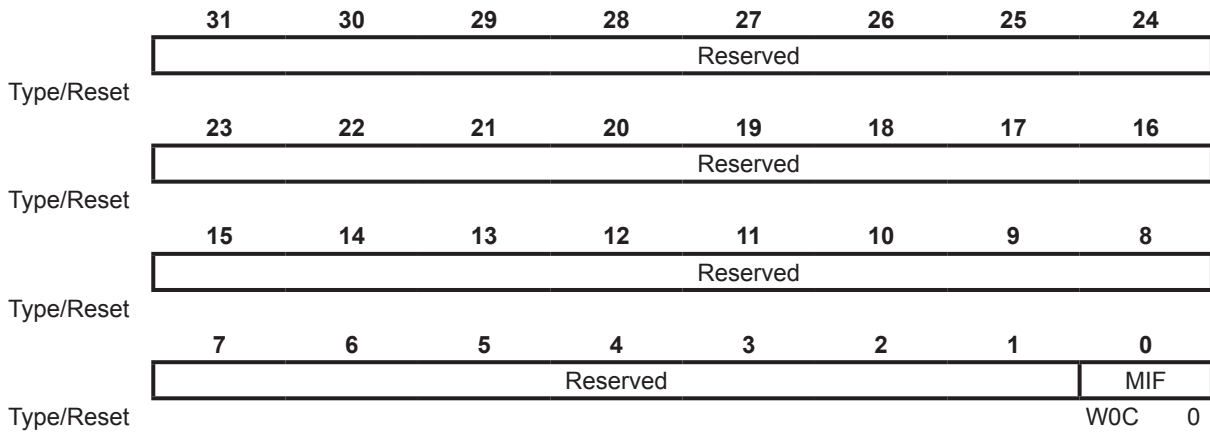
Bits	Field	Descriptions
[2]	CEN	<p>BFTM Counter Enable Control</p> <p>0: BFTM is disabled 1: BFTM is enabled</p> <p>When this bit is set to 1, the BFTM counter will start to count. The counter will stop counting and the counter value will remain unchanged when the CEN bit is cleared to 0 by the application 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.</p>
[1]	OSM	<p>BFTM One Shot Mode Selection</p> <p>0: Counter operates in repetitive mode 1: Counter operates in one shot mode</p>
[0]	MIEN	<p>BFTM Compare Match Interrupt Enable Control</p> <p>0: Compare Match Interrupt is disabled 1: Compare Match Interrupt is enabled</p>

BFTM Status Register – BFTMSR

This register specifies the BFTM status.

Offset: 0x004

Reset value: 0x0000_0004



Bits	Field	Descriptions
[0]	MIF	BFTM Compare Match Interrupt Flag 0: No compare match event occurs 1: Compare match event occurs 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".

BFTM Counter 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	0	RW	0	RW	0
	23	22	21	20	19	18	17	16	CNT					
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8	CNT					
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0	CNT					
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

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.

16 Motor Control Timer (MCTM)

Introduction

The Motor Control Timer consists of one 16-bit up/down-counter, four 16-bit Capture/Compare Registers (CCRs), one 16-bit Counter-Reload Register (CRR), one 8-bit Repetition Counter (REPR) and several control/status registers. It can be used for a variety of purposes which include general time measurement, input signal pulse width measurement, output waveform generation for signals such as single pulse generation or PWM generation, including dead time insertion.

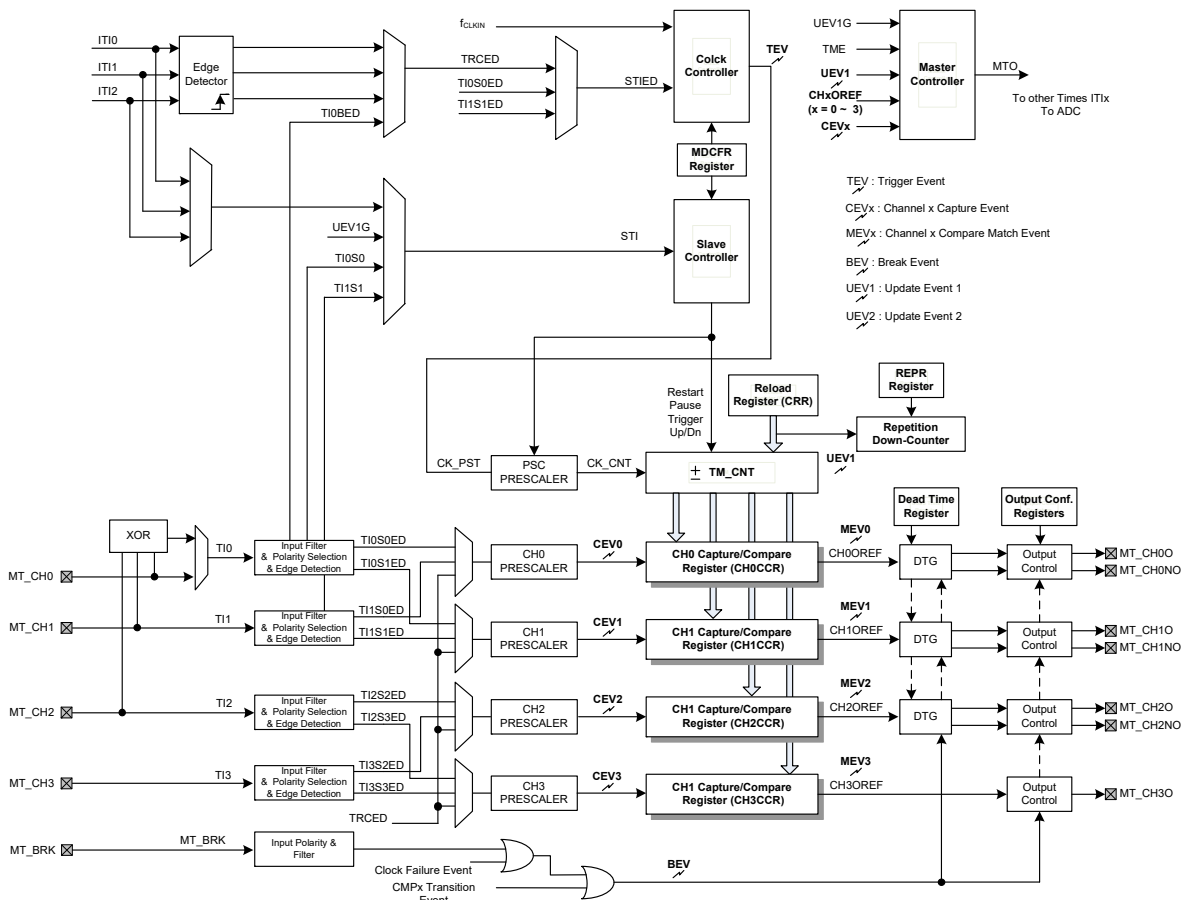


Figure 73. MCTM Block Diagram

Features

- 16-bit up/down auto-reload counter.
- 16-bit programmable prescaler that allows division the counter clock frequency by any factor between 1 and 65536.
- Up to 4 independent channels for:
 - Input Capture function
 - Compare Match Output
 - PWM waveform Generation – Edge and Center-aligned Counting Mode
 - Single Pulse Mode Output
- Complementary Outputs with programmable dead-time insertion
- Repetition counter updates timer registers only after a given number of counter cycles.
- Synchronization circuit controls the timer with external signals and can interconnect several timers together.
- Interrupt/PDMA generation on the following events:
 - Update event 1
 - Update event 2
 - Trigger event
 - Input capture event
 - Output compare match
 - Break event – only interrupt
- MCTM Master/Slave mode controller
- Supports 3-phase motor control and hall sensor interface
- Break input signals to assert the timer output signals in reset state or in a known state

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 an update event 1 is generated by setting the UEV1G bit in the EVGR register to 1, the counter value will also be initialised to 0.

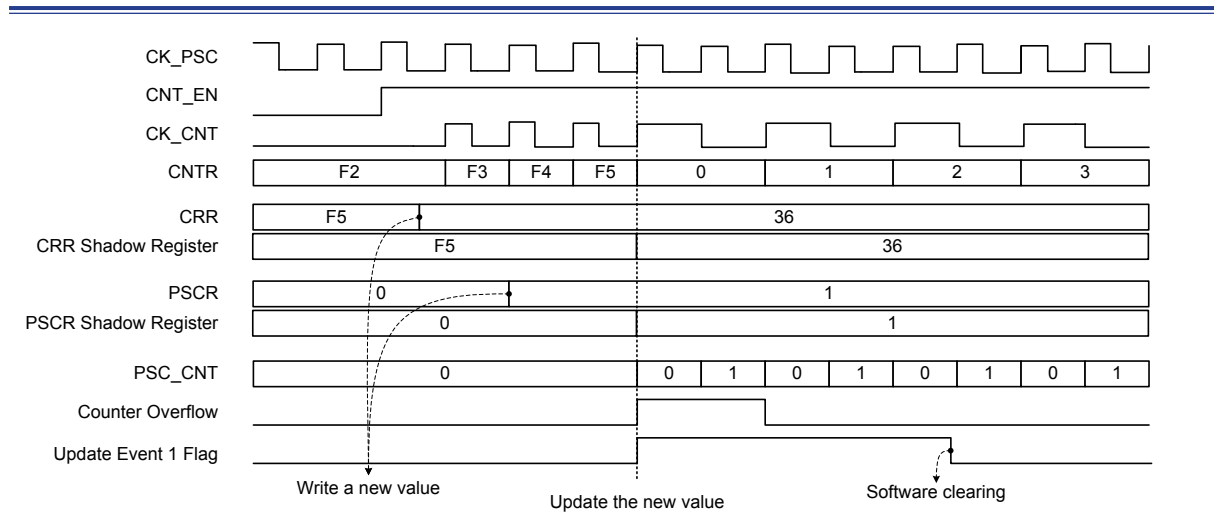


Figure 74. Up-counting Example

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 an update event 1 is generated by setting the UEV1G bit in the EVGR register to 1, the counter value will also be initialised to the counter-reload value.

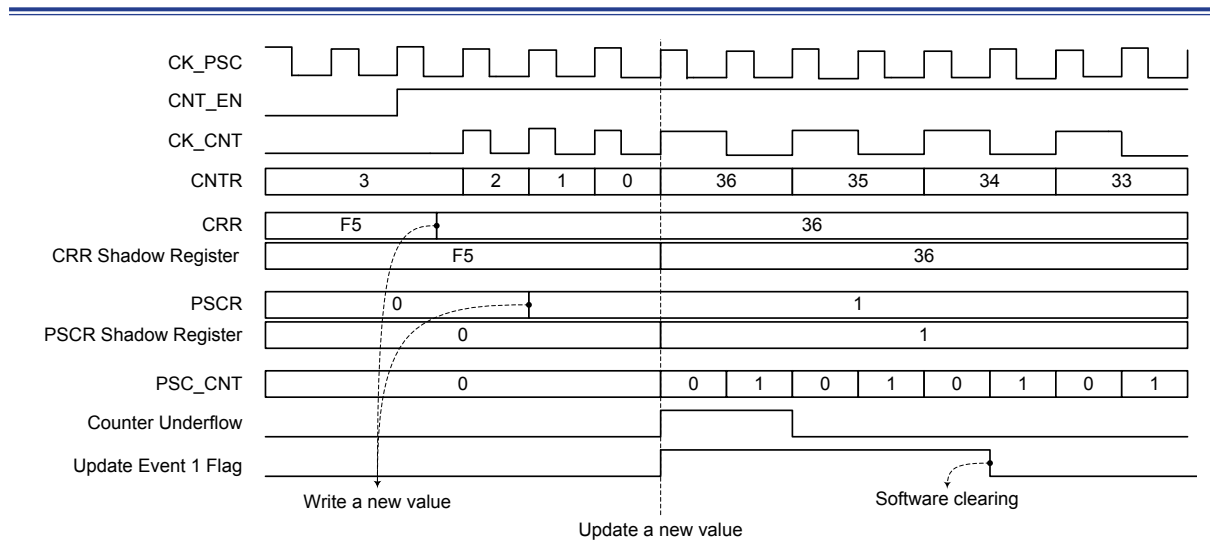


Figure 75. Down-counting Example

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 count direction when in the center-aligned counting mode. The count direction is updated by hardware automatically.

Setting the UEVIG bit in the EVGR register will initialise the counter value to 0 irrespective of whether the counter is counting up or down in the center-aligned counting mode.

The UEV1IF bit in the INTSR register can be set to 1 according to the CMSEL field setting in the CNTCFR register. When CMSEL=0x01, an underflow event will set the UEV1IF bit to 1. When CMSEL=0x10, an overflow event will set the UEV1IF bit to 1. When CMSEL=0x11, either underflow or overflow event will set the UEV1IF bit to 1.

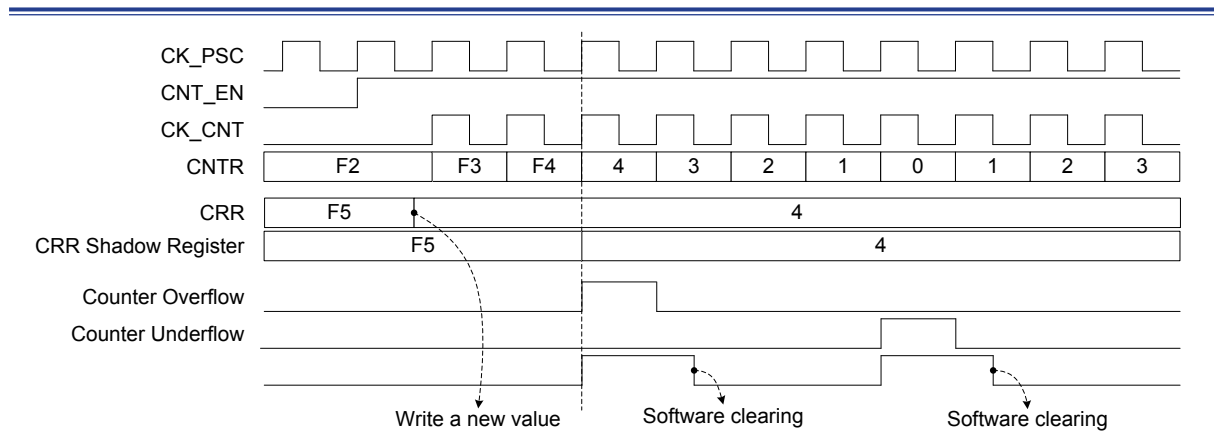


Figure 76. Center-aligned Counting Example

Repetition Down-counter Operation

The update event 1 is usually generated at each overflow or underflow event occurrence. However, when the repetition operation is active by assigning a non-zero value into the REPR register, the update event is only generated if the REPR counter has reached zero. The REPR value is decreased when the following conditions occur:

- At each counter overflow in the up-counting mode
- At each counter underflow in the down-counting mode
- At each counter overflow and underflow in the center-aligned counting mode

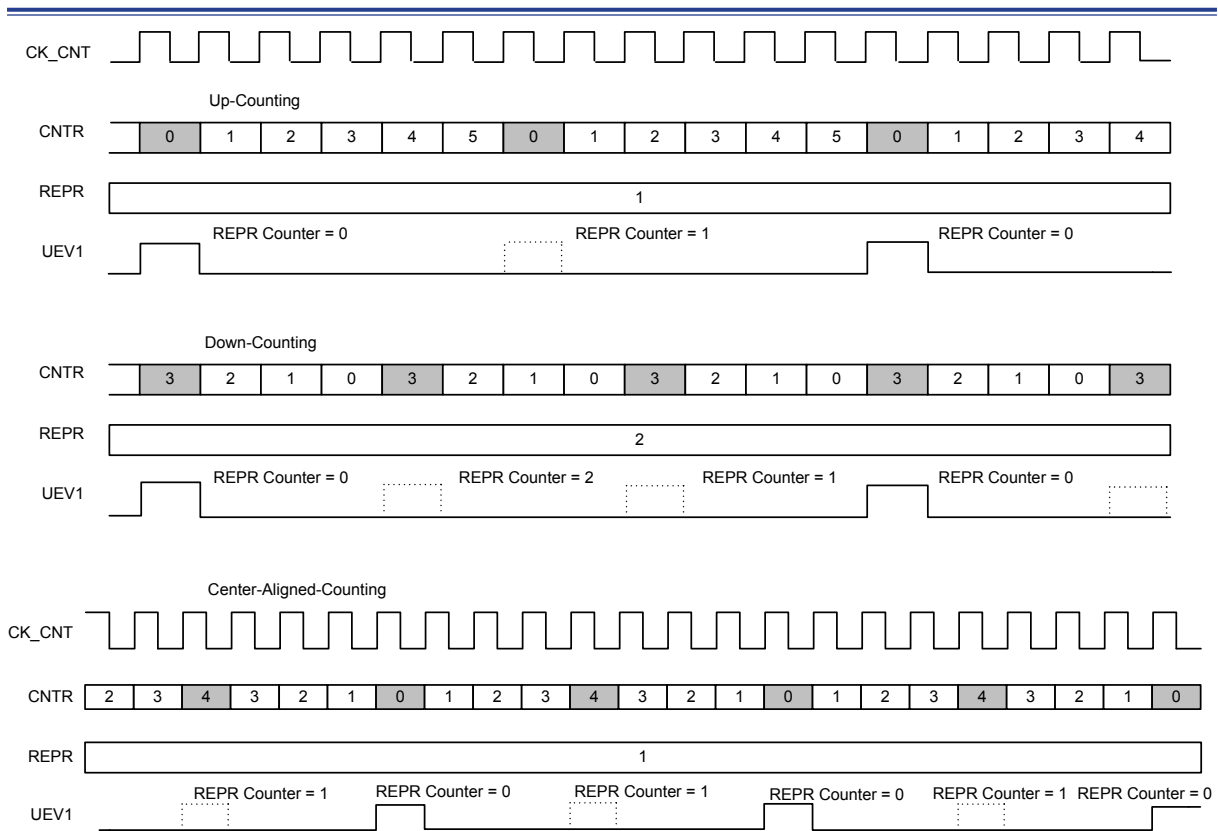


Figure 77. Update Event 1 Dependent Repetition Mechanism Example

Clock Controller

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

- Internal APB clock f_{CLKIN}

The default internal clock source is the APB clock f_{CLKIN} which is used to drive the counter prescaler when the slave mode is disabled. When the slave mode selection bits SMSEL are set to 0x4, 0x5 or 0x6, the internal APB clock f_{CLKIN} is the counter prescaler driving clock source.

- 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 UEVIG bit as the trigger source, then when the SMSEL field is set to 0x7, the counter will be updated instead of counting.

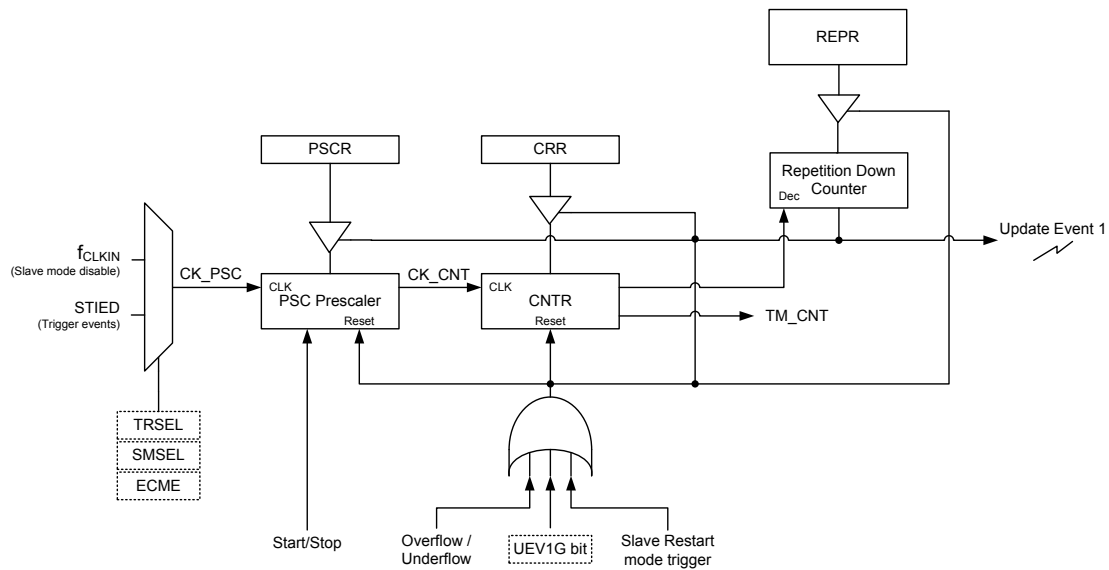
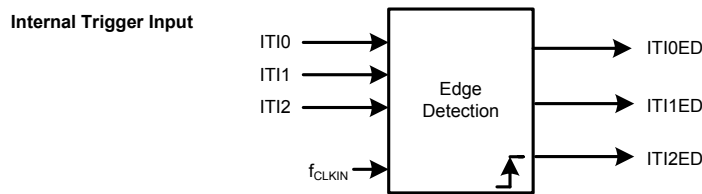


Figure 78. MCTM Clock Selection Source

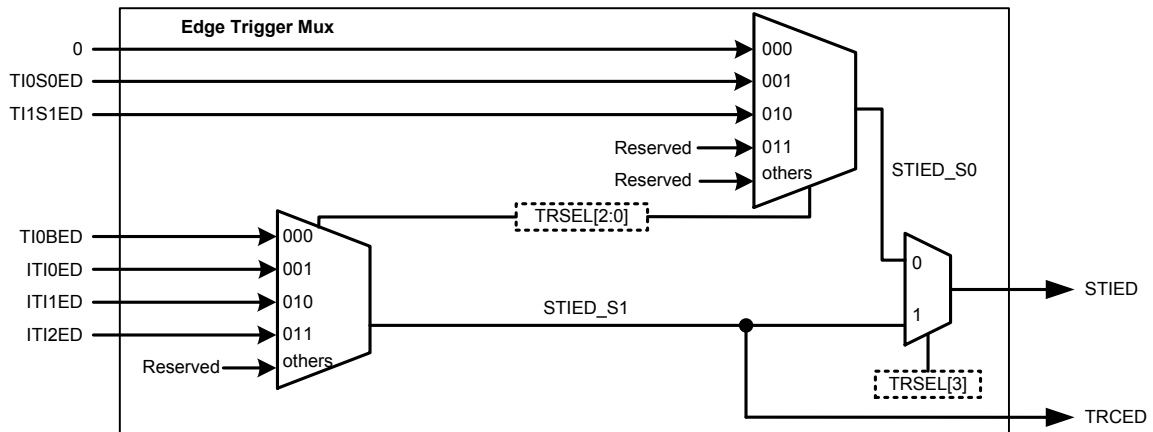
Trigger Controller

The trigger controller is used to select the trigger source and setup the trigger level and edge trigger conditions. For the internal trigger input (ITIx), it can be selected by the Trigger Selection bits, TRSEL, in the TRCFR register. For all the trigger sources except the UEV1G bit software trigger, the internal edge detection circuitry will generate a clock pulse at each trigger signal rising edge to activate some MCTM functions which are triggered by a trigger signal rising edge.

Trigger Controller Block = Edge Trigger Mux + Level Trigger Mux



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



Level Trigger Source = Internal (ITIx) + Channel input (TIn) + Software UEV1G bit

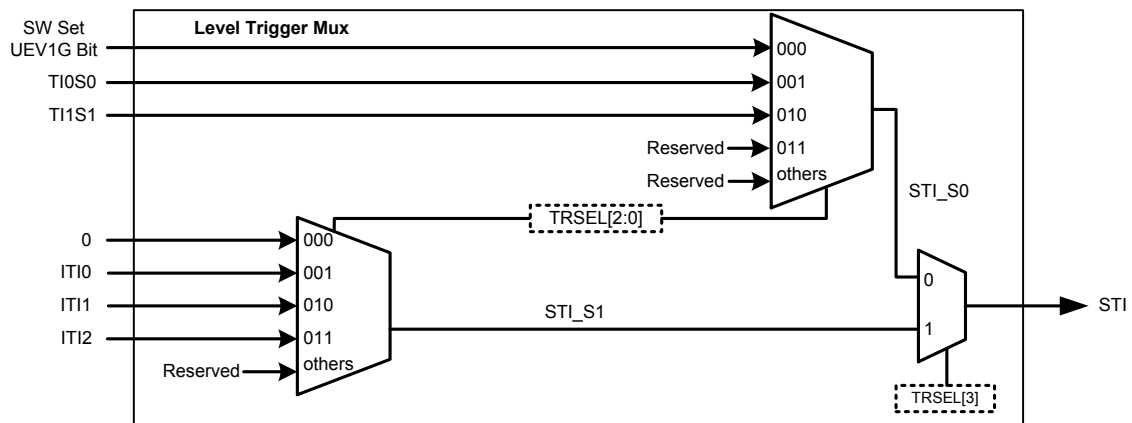


Figure 79. Trigger Controller Block

Slave Controller

The MCTM can be synchronised with an internal/external trigger in several modes including the Restart mode, the Pause mode and the Trigger mode which are 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.

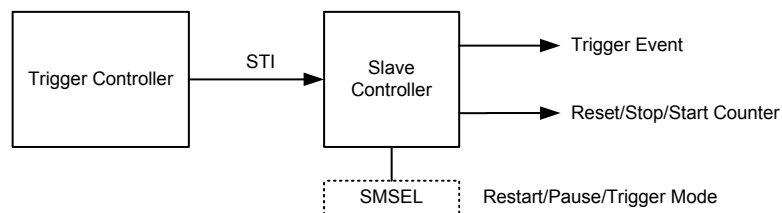


Figure 80. Slave Controller Diagram

Restart Mode

The counter and its prescaler can be reinitialised in response to an STI signal rising edge. If the UEV1DIS bit is set to 1 to disable the update event, then no update event will be generated, however the counter and prescaler are still reinitialized when an STI rising edge occurs. If the UEV1DIS bit in the CNTCFR register is cleared to enable the update event, then an update event will be generated together with the STI rising edge and all the preloaded registers will be updated.

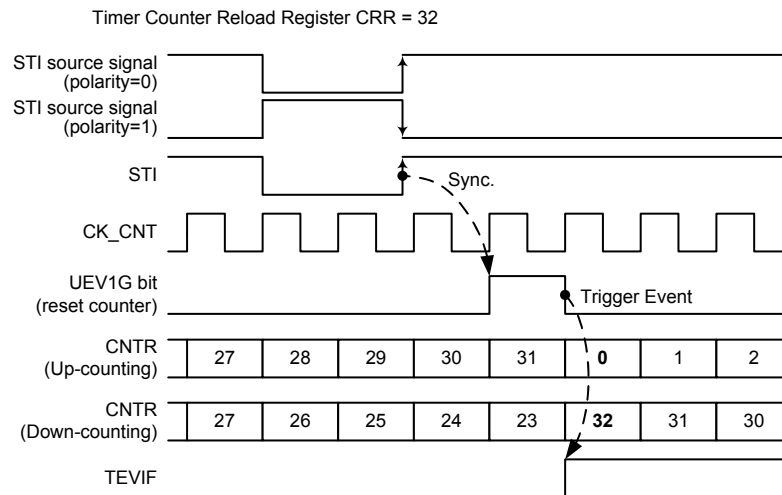


Figure 81. MCTM in Restart Mode

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. When the counter stops, it will maintain its present value and 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.

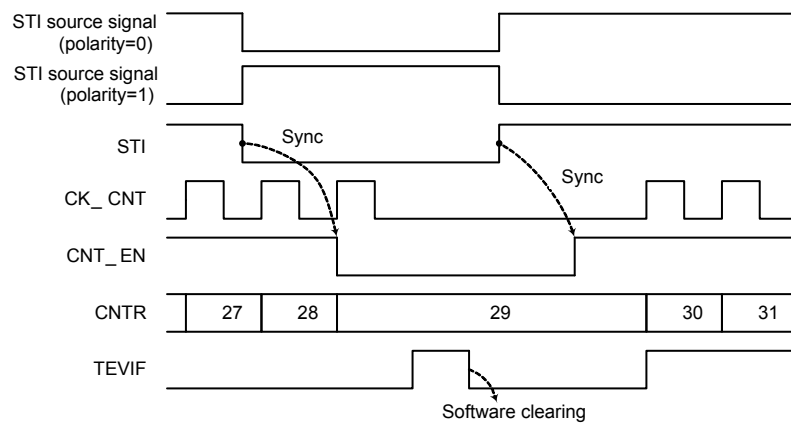


Figure 82. MCTM in Pause Mode

Trigger Mode

After the counter is disabled to count, the counter can resume counting when an 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 sourced from the UEVIG bit software trigger, the counter will not resume counting. When software triggering using the UEVIG 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 to stop counting.

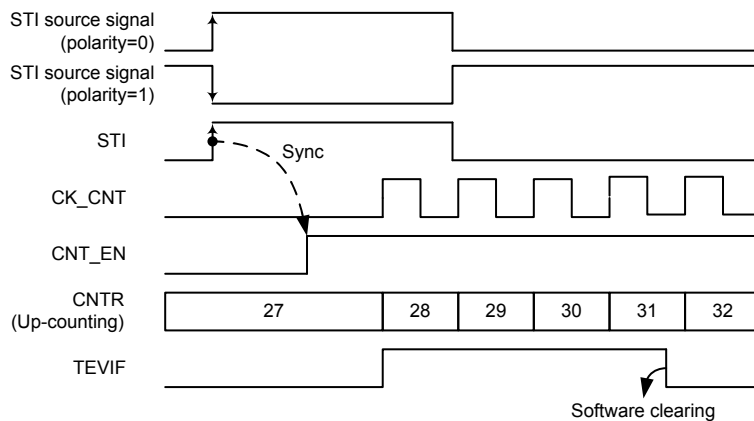


Figure 83. MCTM in Trigger Mode

Master Controller

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

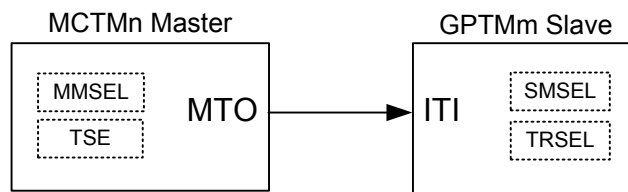


Figure 84. Master MCTMn and Slave GPTMm Connection

The Master Mode Selection bits, MMSEL, in the MDCFR register are used to select the MTO source for synchronising another slave MCTM or GPTM.

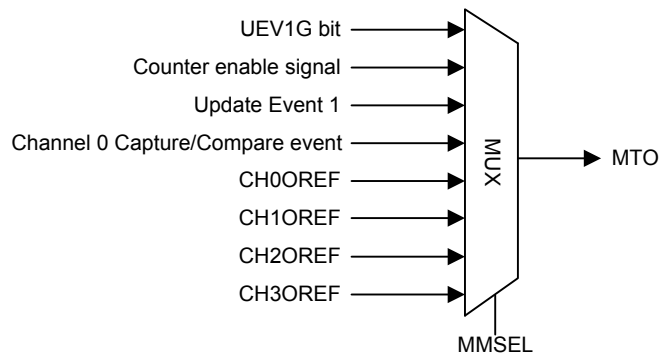


Figure 85. MTO Selection

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

Channel Controller

The MCTM 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 through the read/write 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.

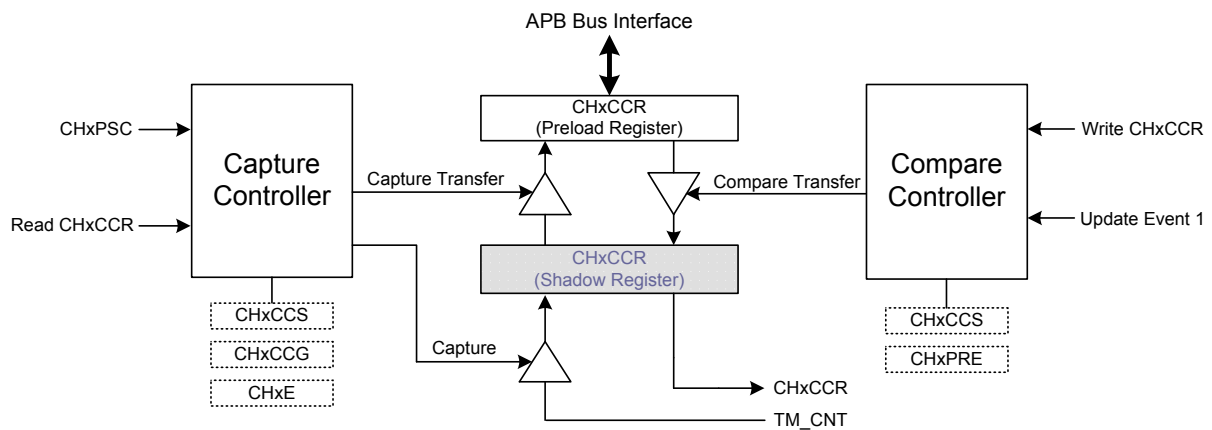


Figure 86. Capture/Compare Block Diagram

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.

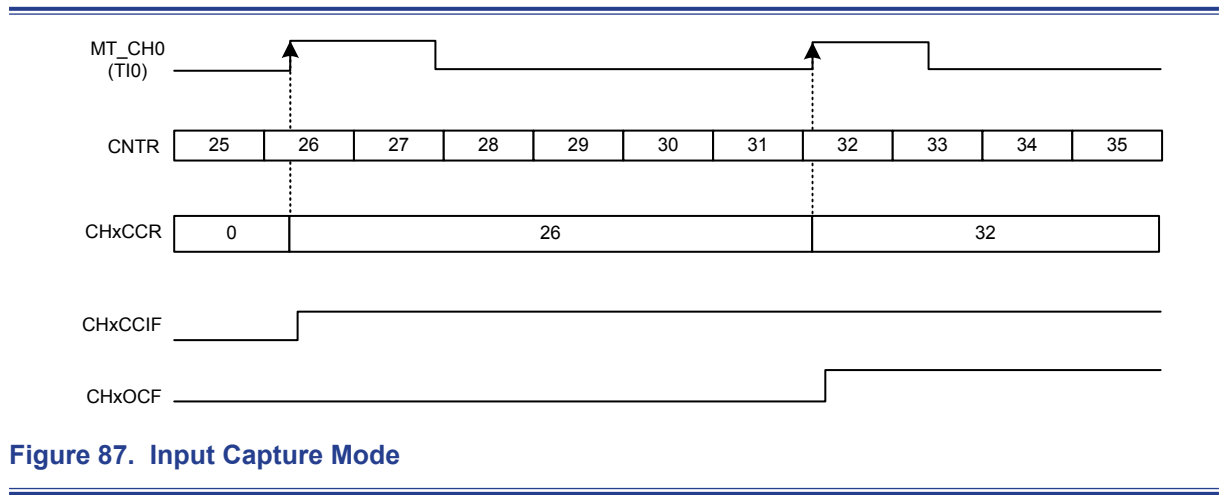


Figure 87. Input Capture Mode

Pulse Width Measurement

The input capture mode can be also used for pulse width measurement from signals on the MT_CHx pins, TIx. The following example shows how to configure the MCTM when operated in the input capture mode to measure the high pulse width and the input period on the MT_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.
- Set the CHIP bit to 1 to choose the falling edge of the TI0 input as the active polarity.
- Setup the TRSEL bits to 0x0001 to select TIOS0 as the trigger input.
- Configure the Slave controller to operate in the Restart mode by setting the SMSEL 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 MT_CH0 pin will be captured into the CH1CCR register while the input period will be captured into the CH0CCR register after an input capture operation.

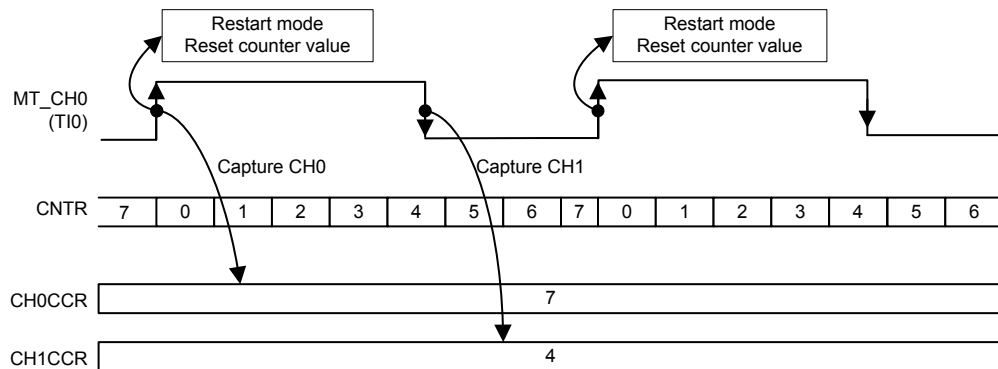


Figure 88. PWM Pulse Width Measurement Example

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 T10 can be chosen to come from the MT_CH0 signal or the Exclusive-OR function of the MT_CH0, MT_CH1 and MT_CH2 signals. The channel input signal T1x is sampled by a digital filter to generate a filtered input signal T1xFP. Then the channel polarity and the edge detection block can generate a T1xSxED signal for the input capture function. The effective input event number can be set by the channel input prescaler register CHxPSC.

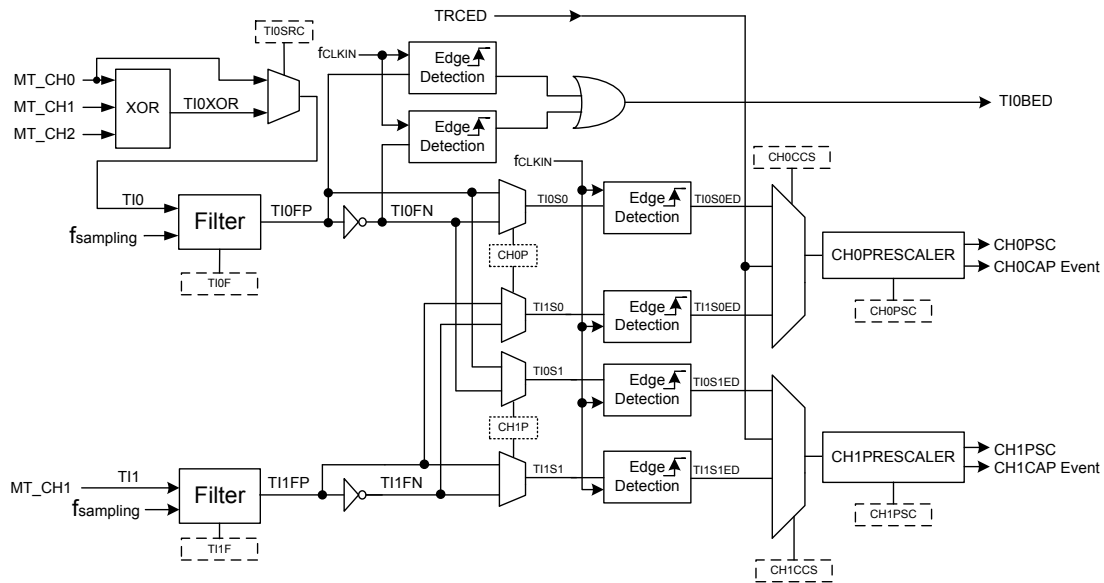


Figure 89. Channel 0 and Channel 1 Input Stages

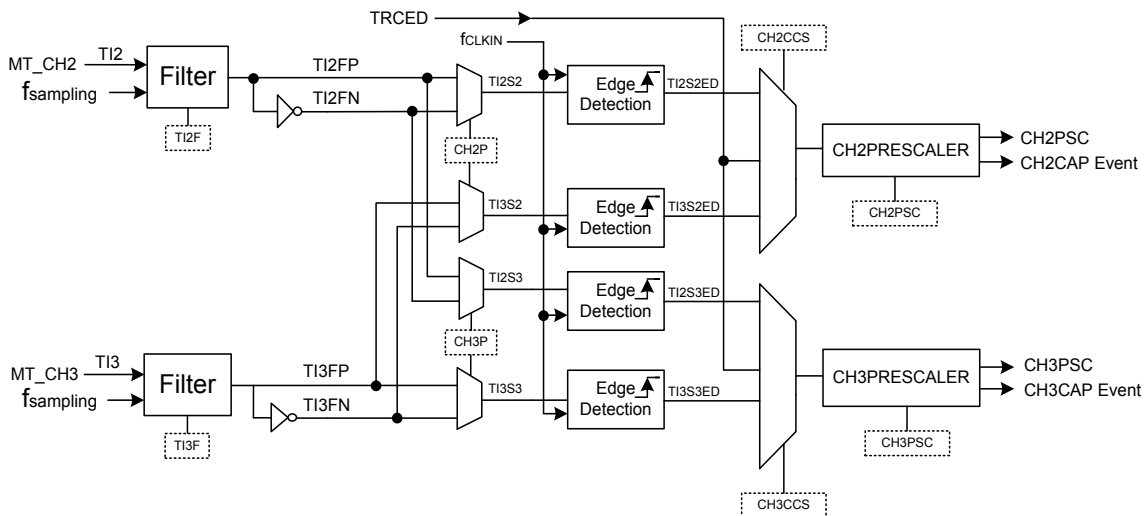


Figure 90. Channel 2 and Channel 3 Input Stages

Digital Filter

The digital filters are embedded in the input stage and clock controller block for the MT_CH0 ~ MT_CH3 pins. The digital filter in the MCTM is an N-event counter where N refers to how many valid transitions are necessary to output a filtered signal. The N value can be 0, 2, 4, 5, 6 or 8 according to the selection for each filter.

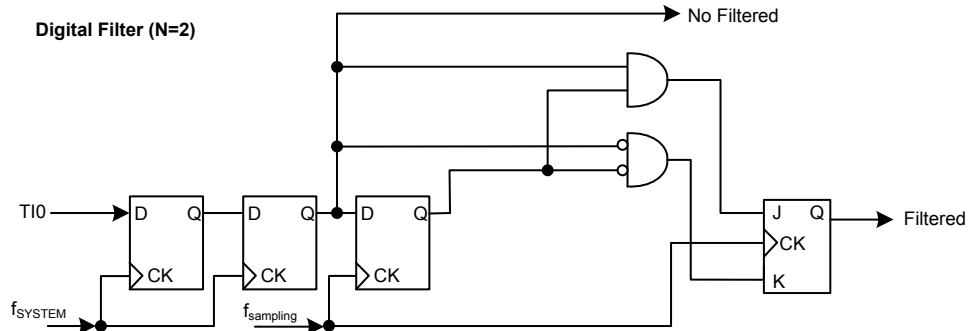


Figure 91. TIO Digital Filter Diagram with N = 2

Output Stage

The MCTM supports complementary outputs for channels 0, 1 and 2 with dead time insertion. The MCTM channel 3 output function is almost the same as that of GPTM channel 3 except for the break function.

The channel outputs, CHxO and CHxNO, are referenced to the CHxOREF signal. These channel outputs generate a wide variety of wide waveforms according to the configuration values of corresponding control bits, as shown by the dashed box in the diagram.

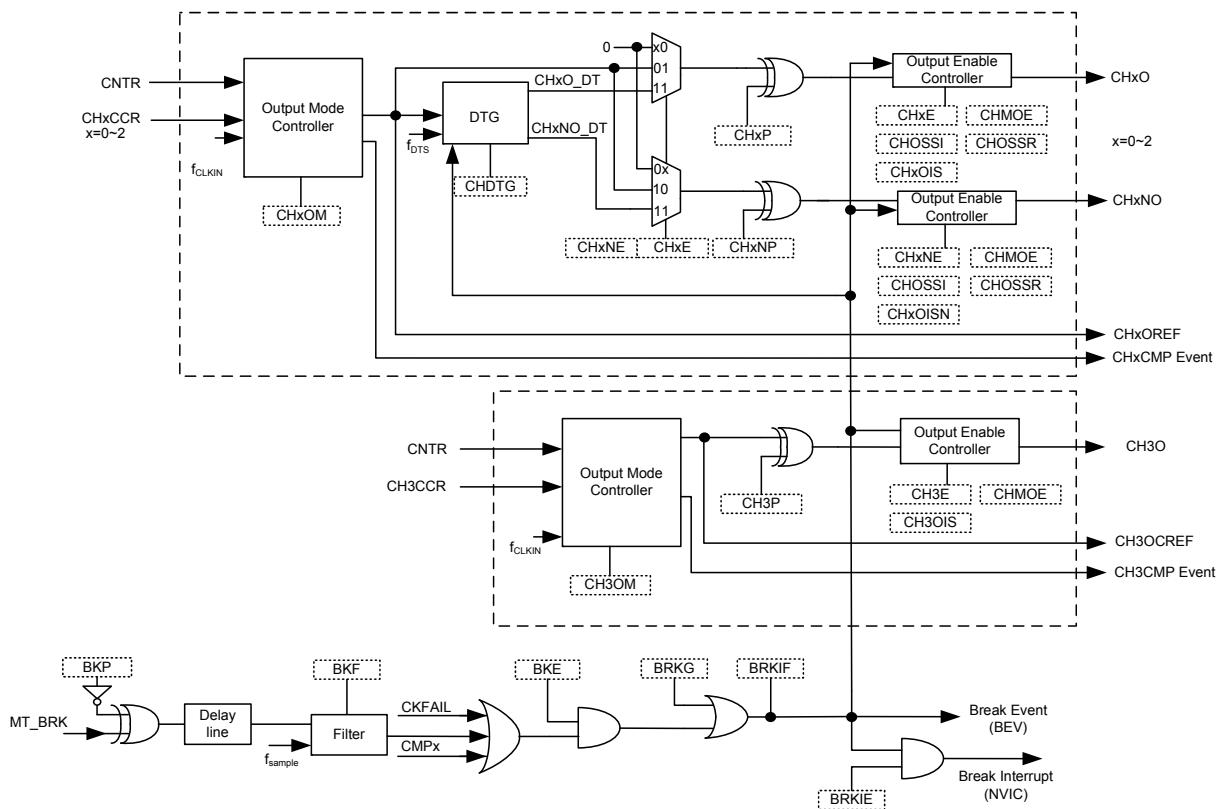


Figure 92. Output Stage Block Diagram

Channel Output Reference Signal

When the MCTM is used in the compare match output mode, the CHxOREF signal (Channel x Output Reference signal) is defined by the CHxOM bit 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 35 shows a summary of the output type setup.

Table 35. Compare Match Output Setup

CHxOM value	Compare Match Level
0x00	No change
0x01	Clear Output to 0
0x02	Set Output to 1
0x03	Toggle Output
0x04	Force Inactive Level
0x05	Force Active Level
0x06	PWM Mode 1
0x07	PWM Mode 2

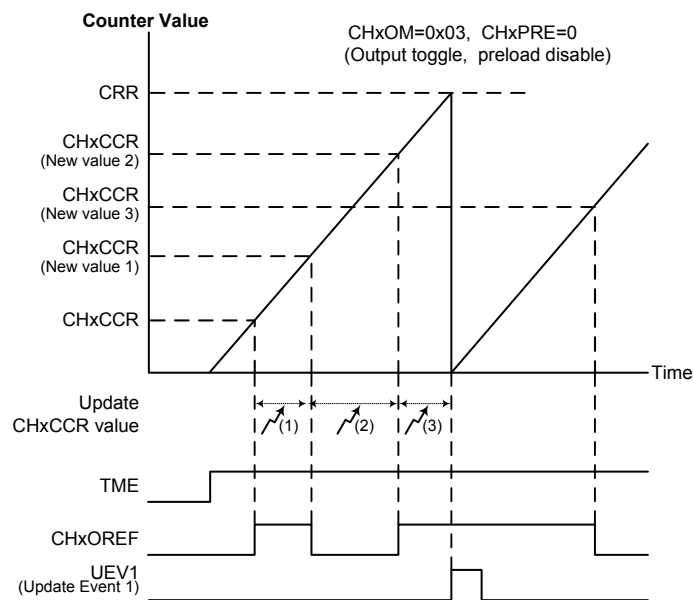


Figure 93. Toggle Mode Channel Output Reference Signal – CHxPRE = 0

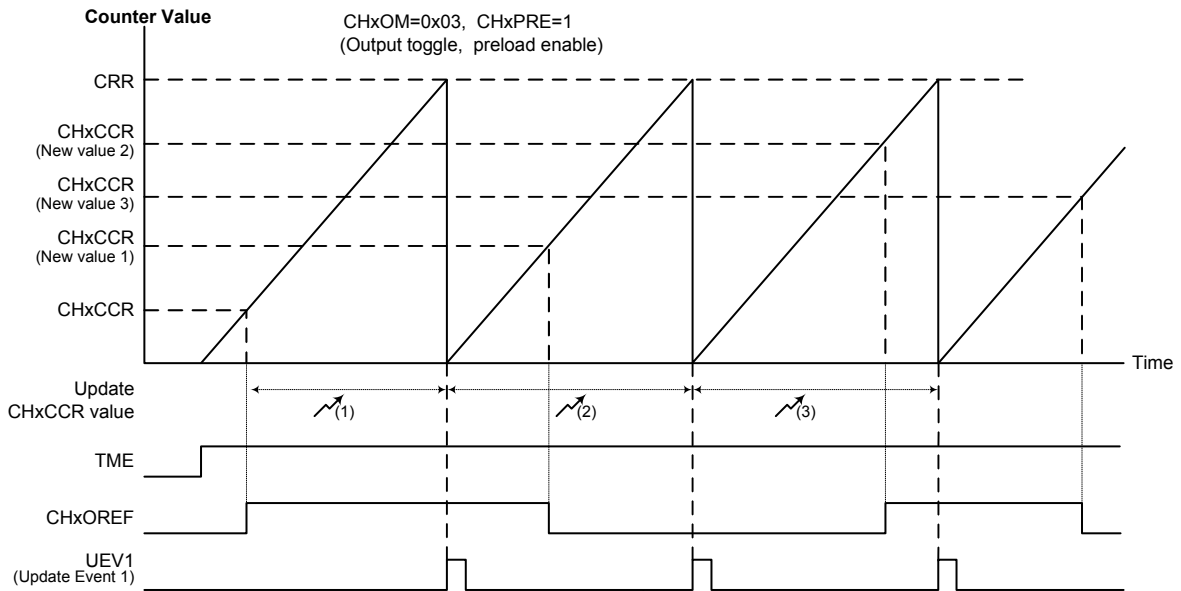


Figure 94. Toggle Mode Channel Output Reference Signal – CHxPRE = 1

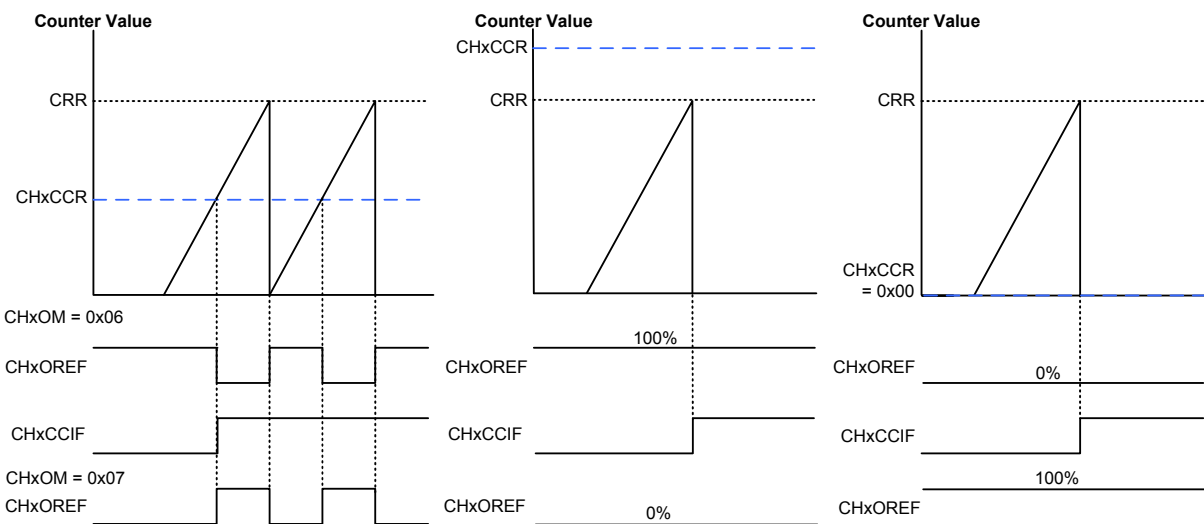


Figure 95. PWM Mode Channel Output Reference Signal and Counter in Up-counting Mode

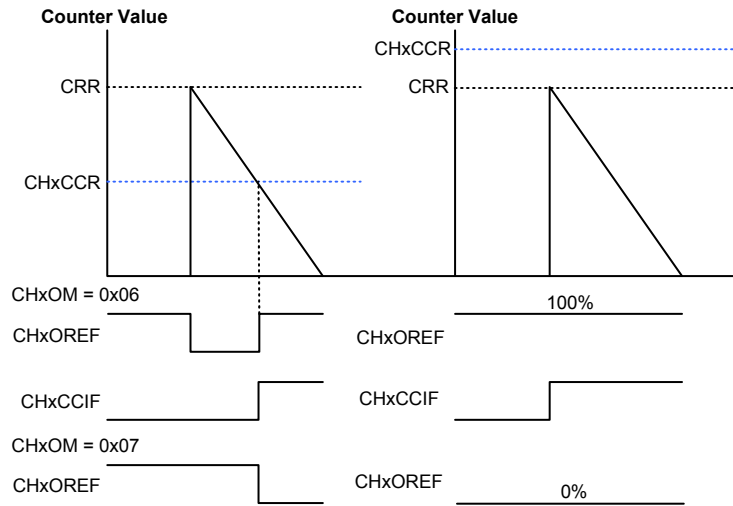


Figure 96. PWM Mode Channel Output Reference Signal and Counter in Down-counting Mode

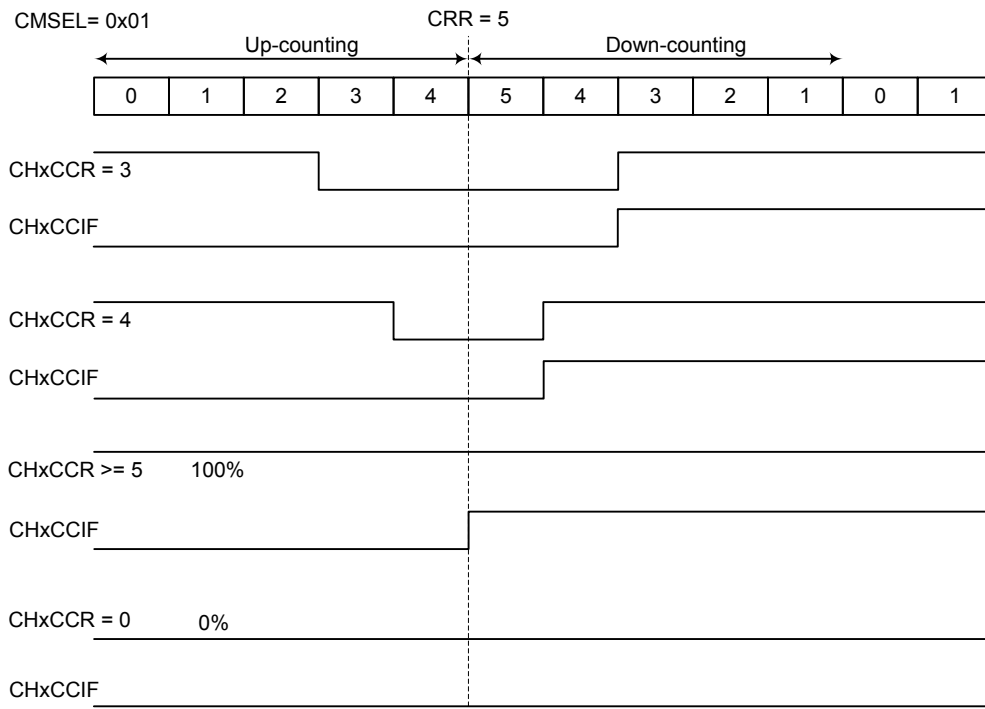


Figure 97. PWM Mode 1 Channel Output Reference Signal and Counter in Centre-aligned Counting Mode

Dead Time Generator

An 8-bit dead time generator function is included for channels 0~2. The dead time insertion is enabled by setting both the CHxE and CHxNE bits. The relationship between the CHxO and CHxNO signals with respect to the CHxOREF signal is as follows:

- The CHxO signal is the same as the CHxOREF signal except for the rising edge which is delayed with a dead time relative to the reference signal rising edge.
- The CHxNO is the opposite of the CHxOREF signal except for the rising edge which is delayed with a dead time relative to the reference signal falling edge.

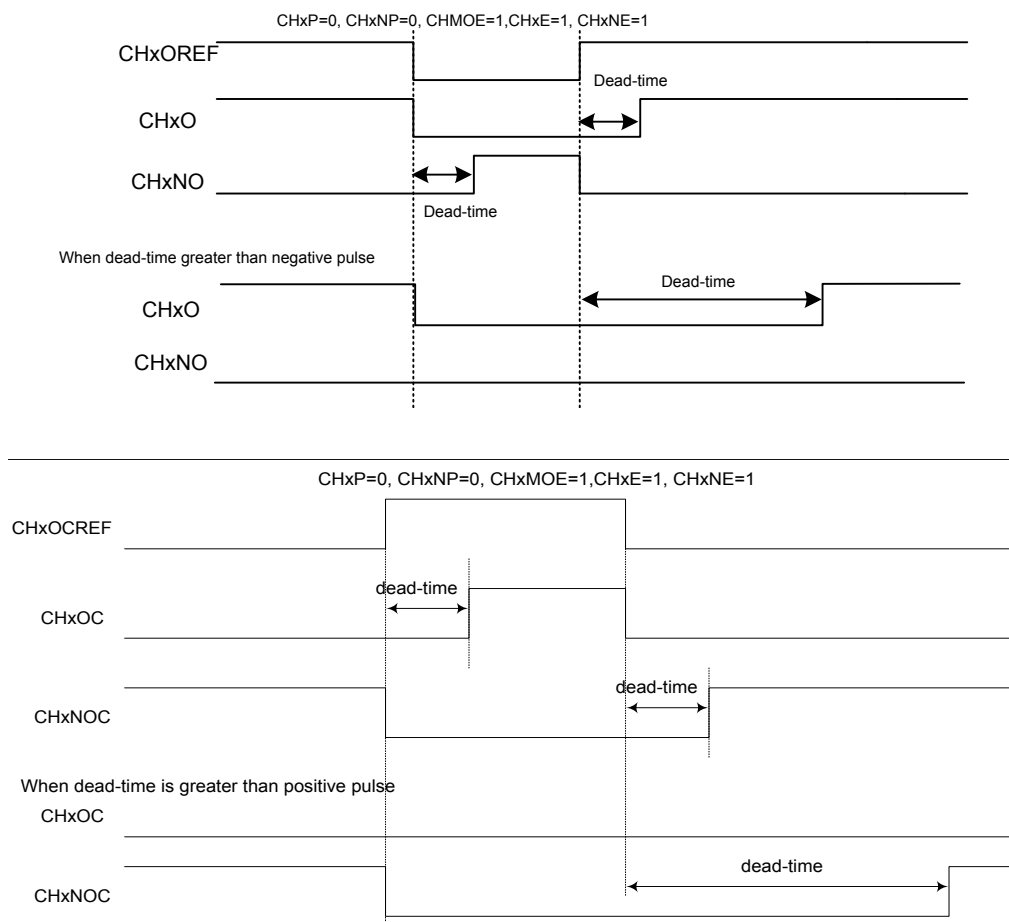


Figure 98. Dead-time Insertion Performed for Complementary Outputs

If the delay is greater than the width of the active output of CHxO or CHxNO, then the corresponding PWM pulses will not be generated.

Break Function

The MCTM includes break function and one input signals for MCTM break. The MT_BRK is default function and from external MT_BRK pin. The detail block diagram is shown as below figure.

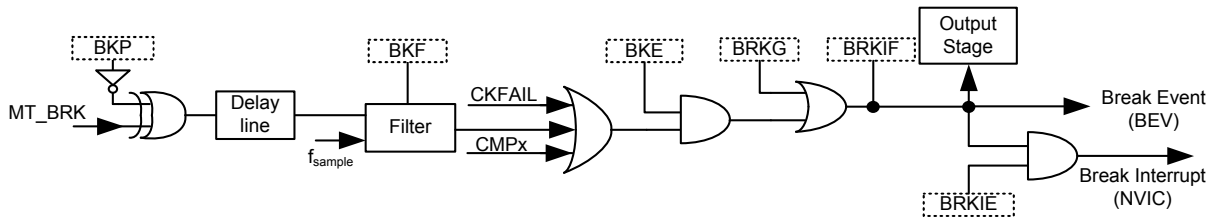


Figure 99. MCTM Break Signal Block Diagram

When the MT_BRK input has an active level or the Clock Monitor Circuitry detects a clock failure event, a break event will be generated if the break function is enabled. Meanwhile, each channel output will be forced to a reset state, an inactive or idle state. Moreover, a break event can also be generated by the software asserting the BRKG bit in the EVGR register even if the break function is disabled.

The MT_BRK input signal can be enabled by setting the BKE bit in the CHBRKCTR register. The break input polarity can be selected by setting the BKP bit in CHBRKCTR register. The BKE and BKP bits can be modified at the same time.

The digital filters are embedded in the input stage and clock controller block for the break signal. The input filter of the MT_BRK signal can be enabled by by setting the BKF bits in the CHBRKCTR register. The digital filter is an N-event counter where N refers to how many valid transitions are necessary to output a filtered signal.

Digital Filter (N=2)

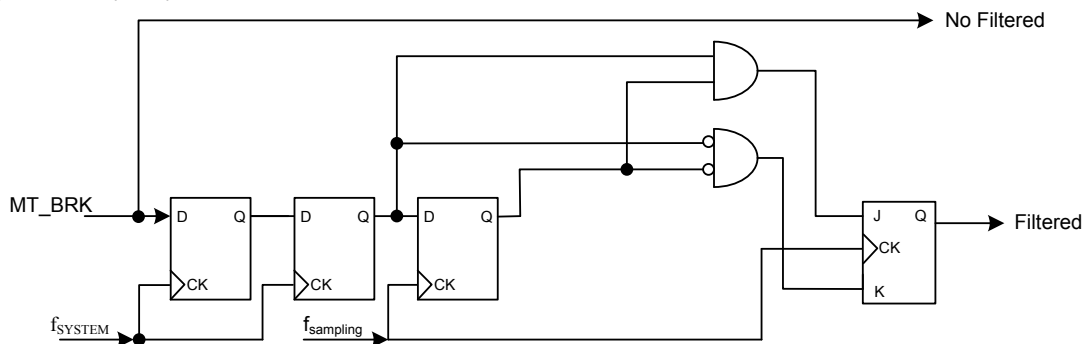


Figure 100. MT_BRK Pin Digital Filter Diagram with N = 2

When using the break function, the channel output enable signals and output levels are changed depending on several control bits which include the CHMOE, CHOSSI, CHOSSR, CHxOIS and CHxOISN bits. Once a break event occurs, the output enable bit CHMOE will be cleared asynchronously. The break interrupt flag, BRKIF, will be set and then an interrupt will be generated if the break function interrupt is enabled by setting the BRKIE bit to 1. The channel output behavior is as described below:

- If complementary outputs are used, the channel outputs a level signal first which can be selected to be either a disable or inactive level, selected by configuring the CHOSSI bit in the CHBRKCTR register. After the dead-time duration, the outputs will be changed to the idle state. The idle state is determined by the CHxOIS/CHxOISN bits in the CHBRKCFR register.
 - If complementary outputs are not used (Channel 3), the channel will output an idle state.
- The main output enable control bit, CHMOE can not be set until the break event is cleared.

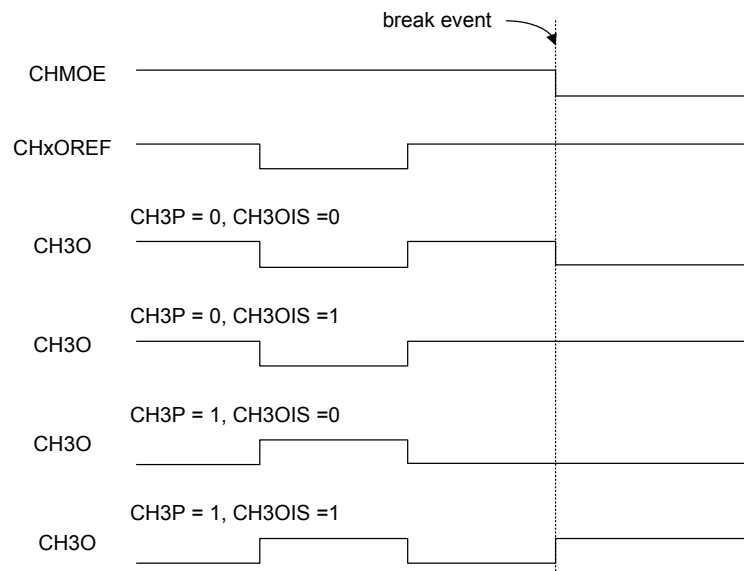


Figure 101. Channel 3 Output with a Break Event Occurrence

The accompanying diagram shows that the complementary output states when a break event occurs where the complementary outputs are enabled by setting both the CHxE and CHxNE bits to 1.

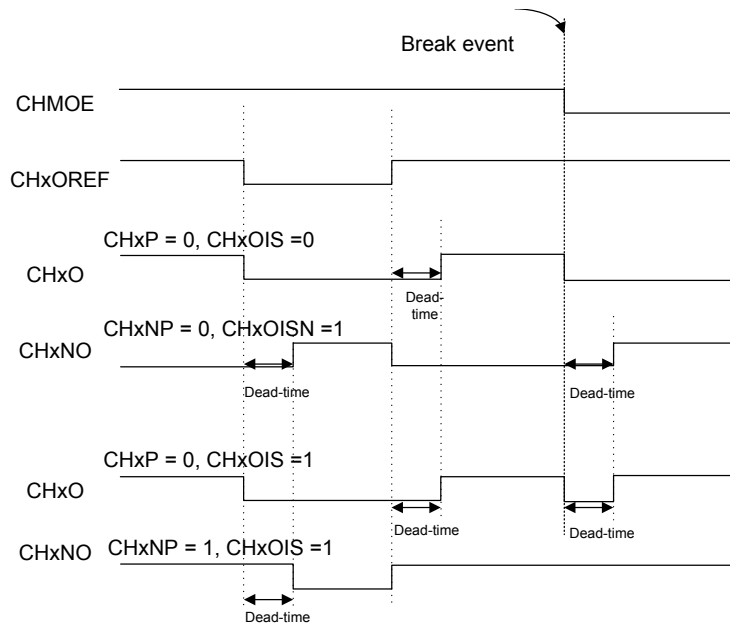


Figure 102. Channel 0 ~ 2 Complementary Outputs with a Break Event Occurrence

The accompanying diagram shows the output states in the case of the output being enabled by setting the CHxE bit to 1 and the complementary output being disabled by clearing the CHxNE to 0 when a break event occurs.

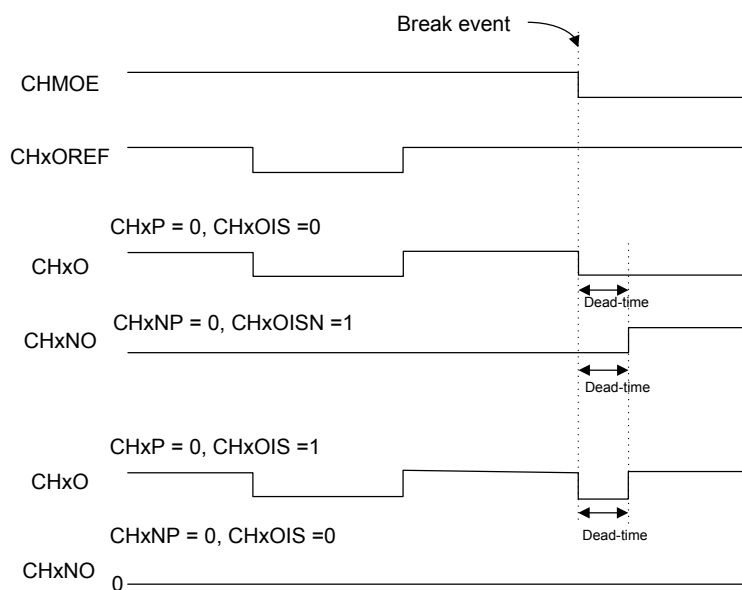


Figure 103. Channel 0 ~ 2 Only One Output Enabled when Fault Event Occurs

The CHxO and CHxNO complementary outputs should not be set to an active level at the same time. The hardware will protect the MCTM circuitry to force only one channel output to be in the active state.

Example: Both CHxOIS and CHxOISN are set to active levels after a break event; only the CHxO waveform is generated.

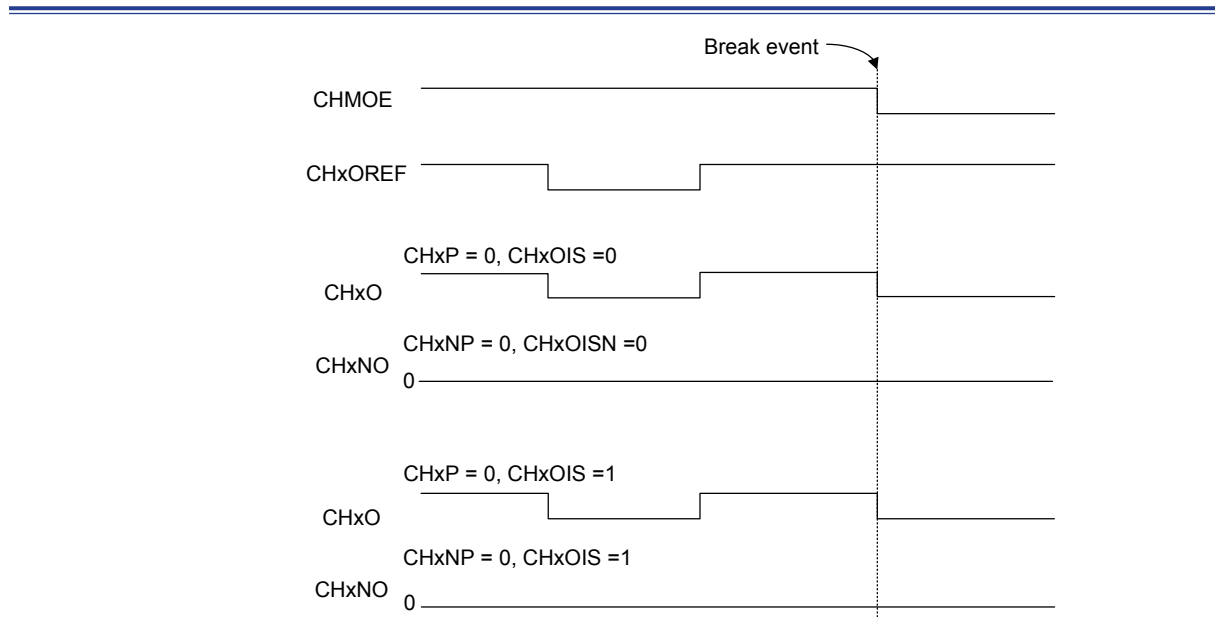


Figure 104. Hardware Protection When Both CHxO and CHxNO Are in Active Condition

CHMOE can be set automatically by update event 1 if the automatic output enable function is enabled by setting the CHAOE bit in the CHBRKCTR register to 1.

Channel Complementary Output with Break Function

The Channel complementary outputs, CHxO and CHxNO, are enabled by a combination of the CHxE, CHxNE, CHMOE, CHOSSR, CHOSSI control bits.

Table 36. Output Control Bits for Complementary Output with a Break Event Occurrence

Control bit					Output status	
CHMOE	CHOSI	CHOSSR	CHxE	CHxNE	MT_CHx Pin output state	MT_CHxN Pin output state
1 (Run)	x	0	0	0	Output disabled - floating - not driven by the timer MT_CHx ^(Note 1) = floating MT_CHx_OEN ^(Note 2) = 1	Output disabled - floating - not driven by the timer MT_CHxN = floating MT_CHxN_OEN = 1
		0	0	1	Output disabled - floating - not driven by the timer MT_CHx_OEN = 1	Output enabled MT_CHxN = CHx_OREF xor CHxNP MT_CHxN_OEN = 0
		0	1	0	Output enabled MT_CHx = CHx_OREF xor CHxP MT_CHx_OEN = 0	Output disabled - floating - not driven by the timer MT_CHxN = floating MT_CHxN_OEN = 1
		0	1	1	Output enabled MT_CHx = CHx_OREF xor CHxP + dead-time MT_CHx_OEN = 0	Output enabled MT_CHxN = not CHx_ OREF xor CHxNP + dead- time MT_CHxN_OEN = 0
		1	0	0	Output disabled - floating - not driven by the timer MT_CHx = floating MT_CHx_OEN = 1	Output disabled floating - not driven by the timer MT_CHxN = floating MT_CHxN_OEN = 1
		1	0	1	Off-State MT_CHx = CHxP MT_CHx_OEN = 0	Output enabled MT_CHxN = CHx_OREF xor CHxNP MT_CHxN_OEN = 0
		1	1	0	Output enabled MT_CHx = CHx_OCREF xor CHxP MT_CHx_OEN = 0	Off-State MT_CHxN = CHxNP MT_CHxN_OEN = 0
		1	1	1	Output enabled MT_CHx = CHx_OREF xor CHxP + dead-time MT_CHx_OEN = 0	Output enabled MT_CHxN = not CHx_ OREF xor CHxNP + dead- time MT_CHxN_OEN = 0
0 (Idle)	x	0	0	0	Before dead-time: Output disabled - floating MT_CHx = floating, MT_CHxN = floating MT_CHx_OEN = 1, MT_CHxN_OEN = 1	
		0	0	1		
		0	1	0		
		0	1	1		
		1	0	0	Before dead-time: Off state MT_CHx = CHxP, MT_CHxN = CHxNP MT_CHx_OEN = 0, MT_CHxN_OEN = 0 After dead-time: Output enabled MT_CHx = CHxOIS, MT_CHxN = CHxOISN MT_CHx_OEN = 0, MT_CHxN_OEN = 0	
		1	0	1		
		1	1	0		
		1	1	1		

Note: 1. The MT_CHx pin is the MCTM I/O Pin.
2. The MT_CHx_OEN and MT_CHxN_OEN signals are the MCTM I/O pin output enable combinational logic control signals which are active low.

Motor Control Timer (MCTM)

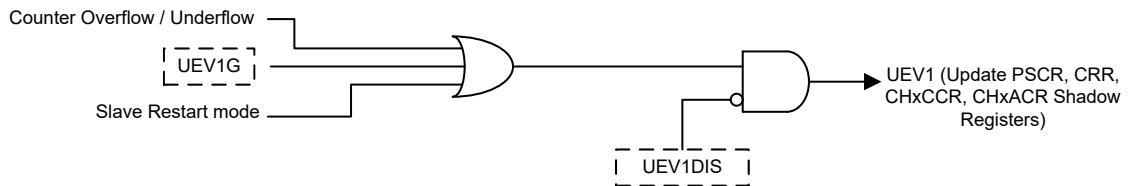
Update Management

The update events are categorised into two different types which are the update event 1, UEV1, and update event 2, UEV2. The update event 1 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 1 occurs when the counter overflows or underflows, the UEV1G bit is set or the slave restart mode is triggered. The update event 2 is used to update the CHxE, CHxNE and CHxOM control bits. An update event 2 is generated when a rising edge on the STI occurs or the corresponding software update control bit is set.

Update Event 1

The UEV1DIS bit in the CNTCFR register can determine whether an update event 1 occurs or not. When the update event 1 occurs, the corresponding update event interrupt will be generated depending upon whether the update event 1 interrupt generation function is enabled or not by configuring the UGDIS bit in the CNTCFR register. For a more detailed description, refer to the UEV1DIS and UGDIS bit definition in the CNTCFR register.

Update Event 1 Management



Update Event 1 Interrupt Management

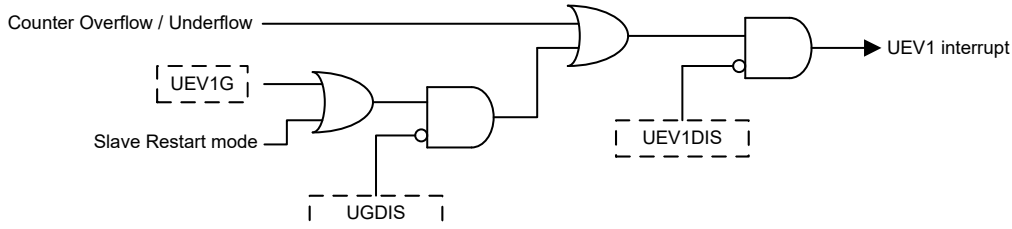


Figure 105. Update Event 1 Setup Diagram

Update Event 2

The CHxE, CHxNE, CHxOM control bits for the complementary outputs can be preloaded by setting the COMPRE bit in the CTR register. Here the shadow bits of the CHxE, CHxNE, and CHxOM bits will be updated when an update event 2 occurs.

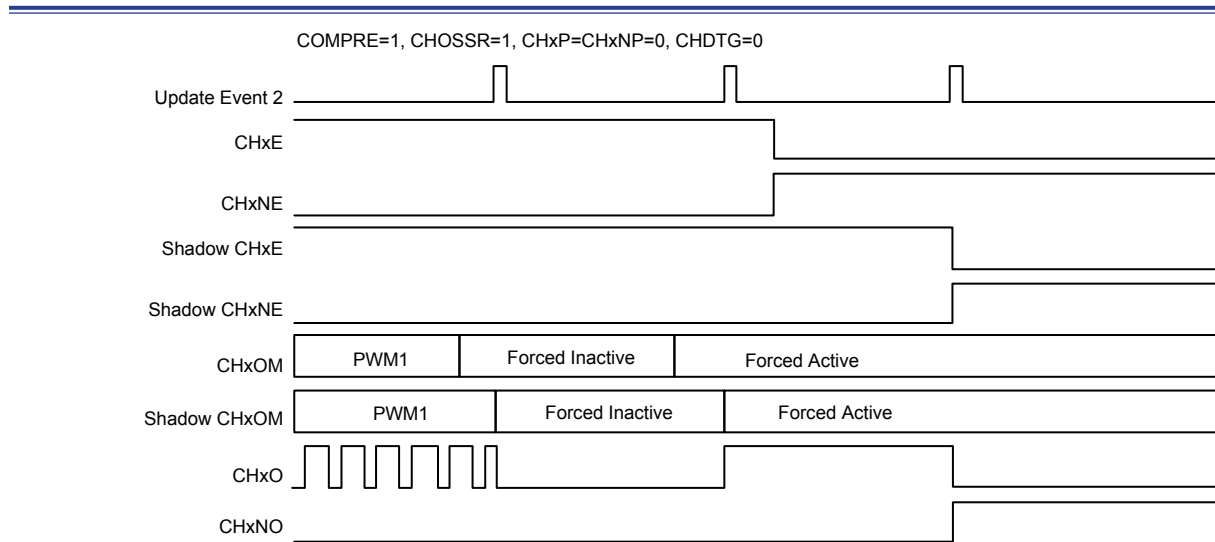


Figure 106. CHxE, CHxNE and CHxOM Updated by Update Event 2

An update event 2 can be generated by setting the software update bit, UEV2G, in the EVGR register or by the rising edge of the STI signal if the COMUS bit is set in the CTR register.

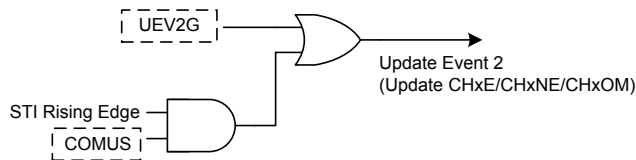


Figure 107. Update Event 2 Setup Diagram

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 1 occurs or the TME bit is cleared 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 1, the counter will be reinitialised.

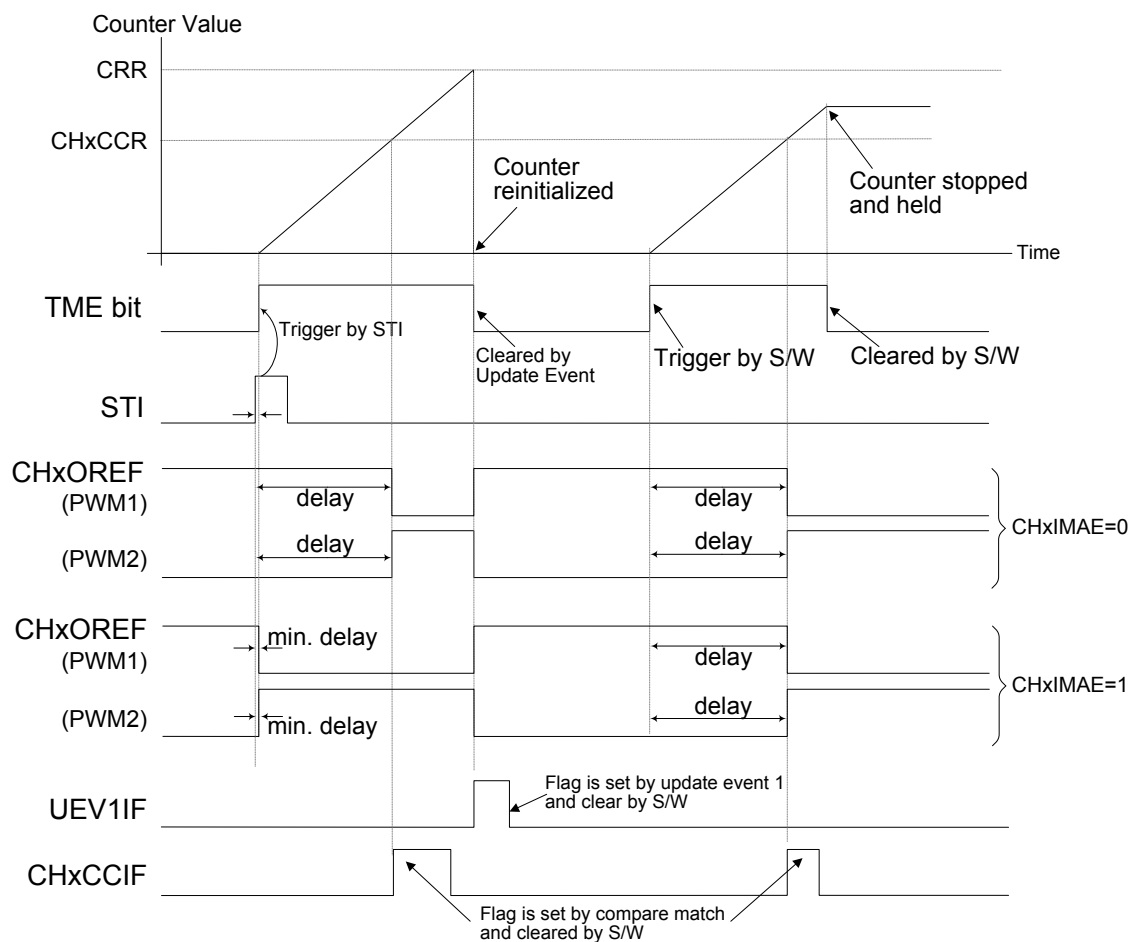


Figure 108. Single Pulse Mode

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, users 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 to 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 PWM1 or PWM2 output mode and the trigger source is derived from the STI signal.

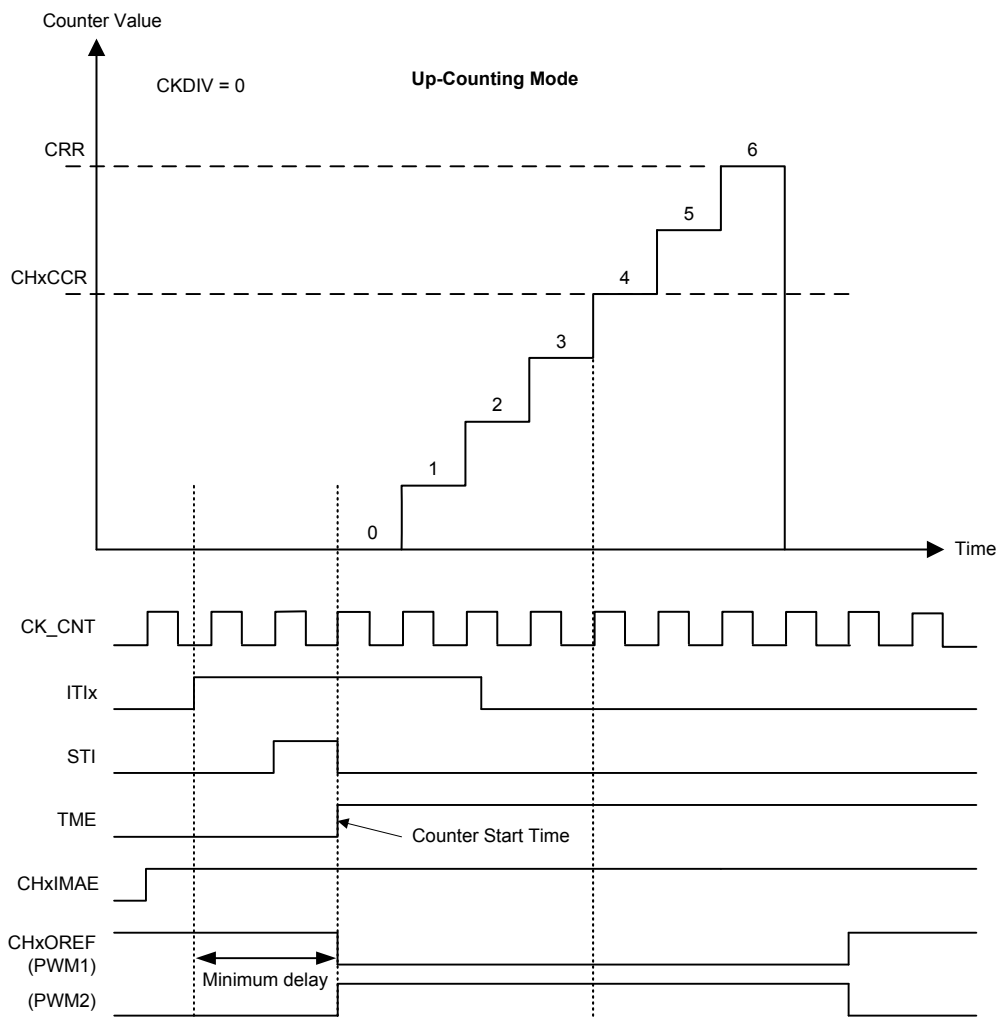


Figure 109. Immediate Active Mode Minimum Delay

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 MCTMx_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 using the value in CHxCCR as up-count compare value. When the counter is in counting down stage, then the value in the CHxACR register is used as the down-count compare value. The Figure 110 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.

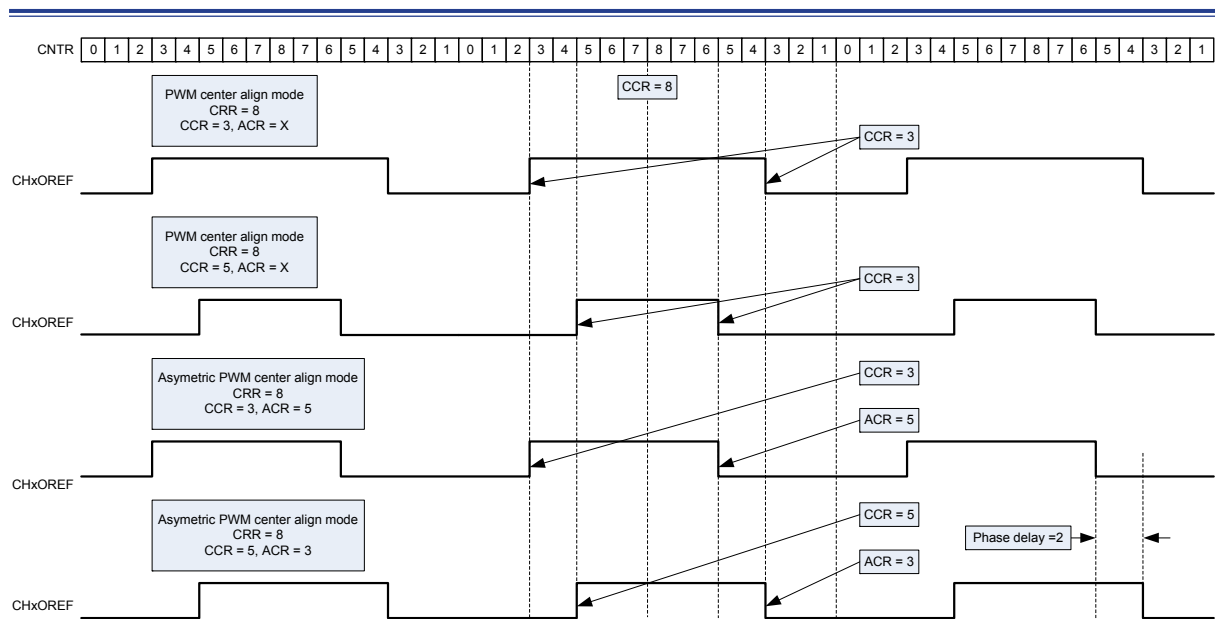


Figure 110. Asymmetric PWM Mode versus Center-aligned Counting Mode

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 trigger another timer to start or stop counting

- Configure MCTM to be in the master mode and to send its channel 0 Output Reference signal CH0OREF as a trigger output (MMSEL = 0x04).
- Configure the MCTM CH0OREF waveform.
- Configure the GPTM to receive its input trigger source from the MCTM trigger output (TRSEL = 0x0A).
- Configure GPTM to operate in the pause mode (SMSEL = 0x05).
- Enable GPTM by writing '1' to the TME bit.
- Enable MCTM by writing '1' to the TME bit.

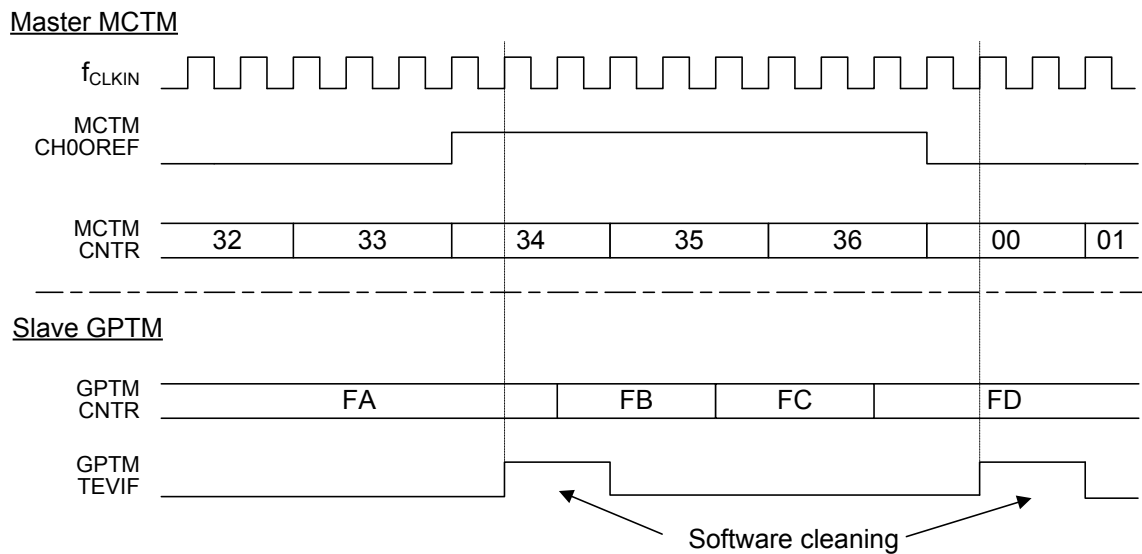


Figure 111. Pausing GPTM using the MCTM CH0OREF Signal

Using one timer to trigger another timer to start counting

- Configure MCTM to operate in the master mode and to send its Update Event UEV as the trigger output (MMSEL = 0x02).
- Configure the MCTM period by setting the CRR register.
- Configure GPTM to get the input trigger source from the MCTM trigger output (TRSEL = 0x0A).
- Configure GPTM to be in the slave trigger mode (SMSEL = 0x06).
- Start MCTM by writing '1' to the TME bit.

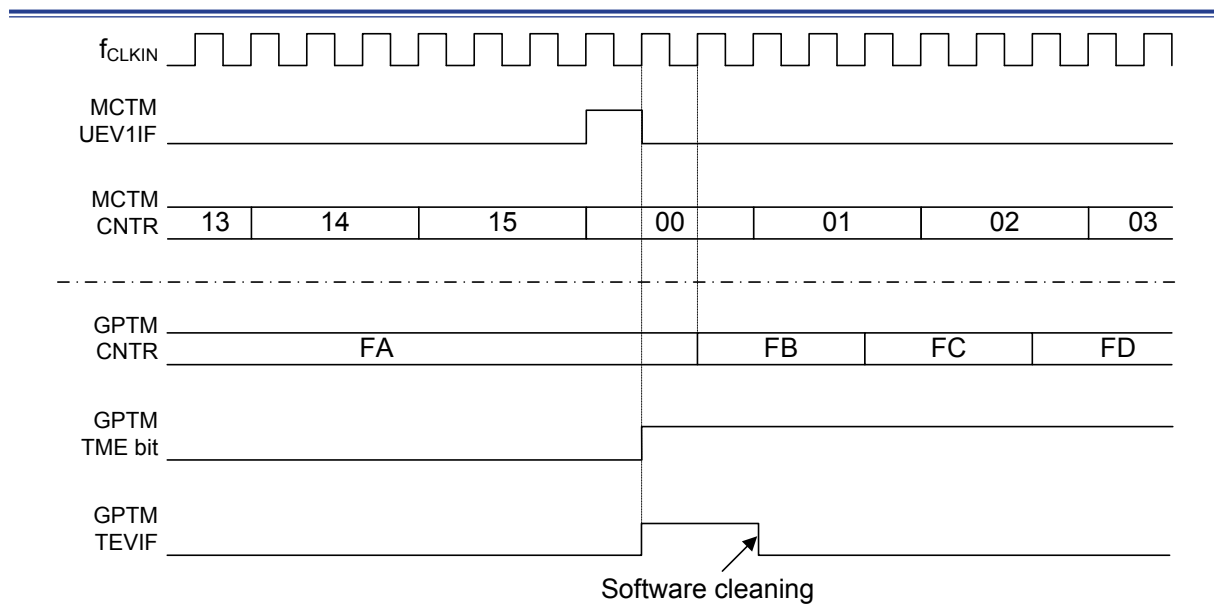
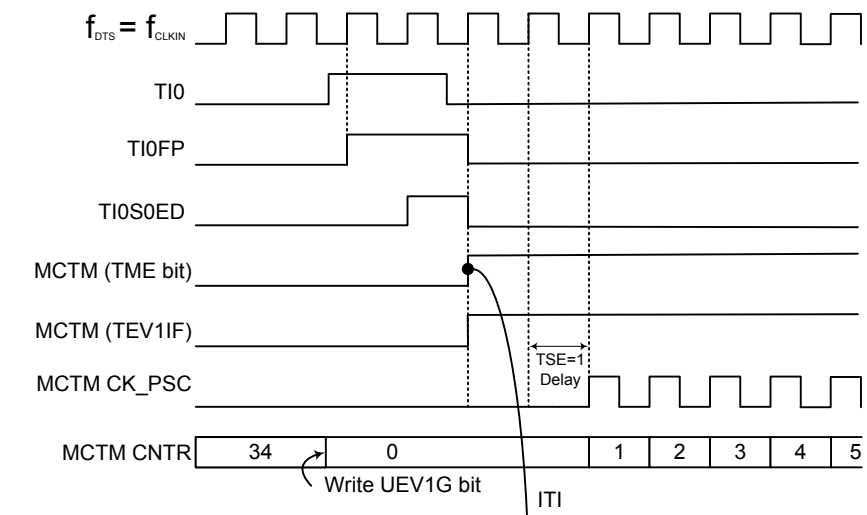


Figure 112. Triggering GPTM with MCTM Update Event 1

Starting two timers synchronously in response to an external trigger

- Configure MCTM to operate in the master mode to send its enable signal as a trigger output (MMSEL = 0x01).
- Configure MCTM slave mode to receive its input trigger source from MT_CH0 pin (TRSEL = 0x01).
- Configure MCTM to be in the slave trigger mode (SMSEL = 0x06).
- Enable the MCTM master timer synchronisation function by setting the TSE bit in the MDCFR register to 1 to synchronise the slave timer.
- Configure GPTM to receive its input trigger source from the MCTM trigger output (TRSEL = 0x0A).
- Configure GPTM to be in the slave trigger mode (SMSEL = 0x06).

Master MCTM



Slave GPTM

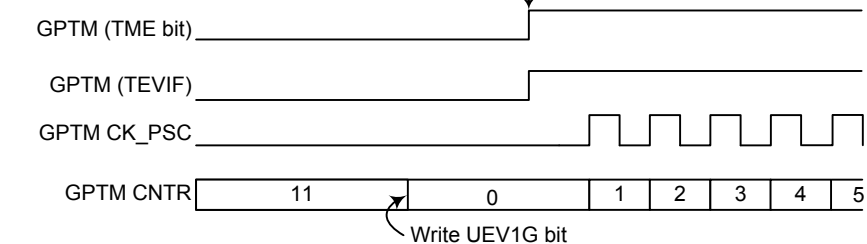


Figure 113. Trigger MCTM and GPTM with the MCTM CH0 Input

Using one timer as a hall sensor interface to trigger another timer with update event 2

GPTM:

- Configure channel 0 to choose an input XOR function (TI0SRC = 1)
- Configure channel 0 to be in the input capture mode and TRCED as capture source (CH0CCS= 0x03) and Enable channel 0 (CH0E=1)
- Configure the UEVG bit as the source of MTO (MMSEL= 0x00)
- Configure TI0BED to be connected to STI (TRSEL = 0x08)
- Configure the counter to be in the slave restart mode (SMSEL = 0x04)
- Enable GPTM (TME=1)

MCTM:

- Select GPTM MTO to be the STI source of MCTM (TRSEL = 0x0A)
- Enable the CHxE, CHxNE and CHxOM preload function (COMPRES = 1)
- Select the rising edge on STI to generate an update event 2 (COMUS = 1)
- Enable the update event 2 interrupt (UEV2IE = 1)
- In the update event 2 ISR: write CHxE, CHxNE and CHxOM register for the next step

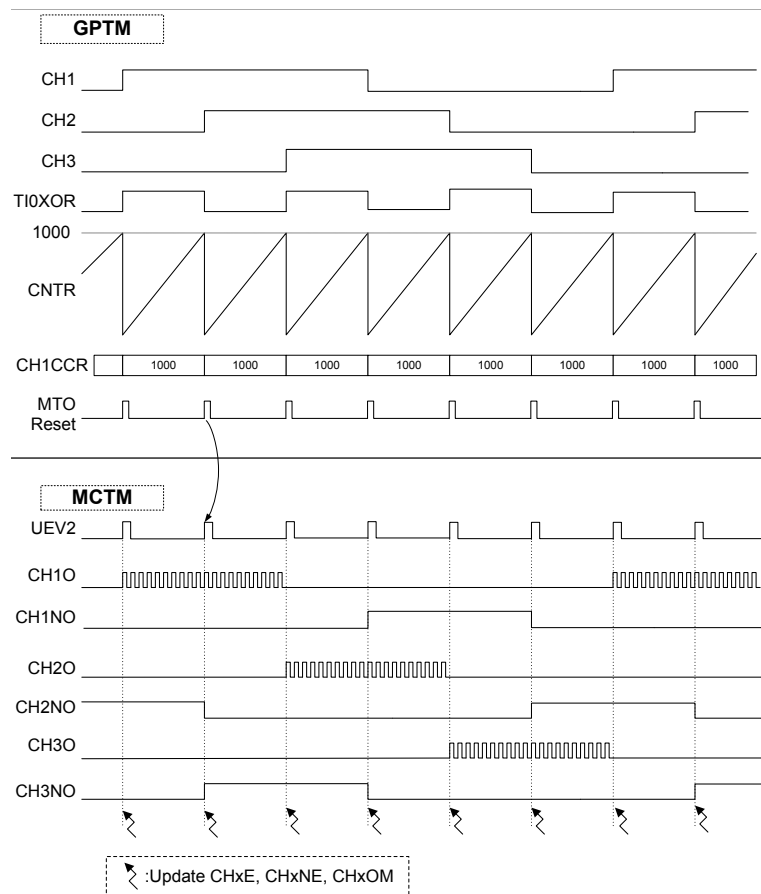


Figure 114. CH1XOR Input as Hall Sensor Interface

Trigger ADC Start

To interconnect to the Analog-to-Digital Converter, the MCTM can output the MTO signal or the channel compare match output signal CHxOREF (x = 0 ~ 3) to be used as an Analog-to-Digital Converter input trigger signal.

Lock Level Table

In addition to the break input and output management, a write protection has been internally implemented in the break circuitry to safeguard the application. Users can choose one protection level selected by the LOCKLV bits to protect the relative control bits of the registers. The LOCKLV bits can only be written once after an MCTM or system reset. Then the protected bits will be locked and can not be changed anymore except by the MCTM reset or when the system is reset.

Table 37. Lock Level Table

Lock Configuration	Protected Bits					
Lock Level 1 (LOCKLV = "01")	CHDTG	CHxOIS	CHxOISN	BKE	BKP	CHAOE
Lock Level 2 (LOCKLV = "10")	CHDTG	CHxOIS	CHxOISN	BKE	BKP	CHAOE
	CHxP	CHxNP	CHOSSI	CHOSSR	MCTMEN ⁽¹⁾	CKMEN ⁽²⁾
Lock Level 3 (LOCKLV = "11")	CHDTG	CHxOIS	CHxOISN	BKE	BKP	CHAOE
	CHxP	CHxNP	CHOSSI	CHOSSR	MCTMEN ⁽¹⁾	CKMEN ⁽²⁾
	CHxPRE	CHxOM				

1. The MCTMEN bit of the APBCCR1 register is located in the CKCU unit and use to control the clock source of the MCTM unit.
2. The CKMEN bit of the GCCR register is located in the CKCU unit and use to monitor the high speed external clock (HSE) source. If the CKMEN bit is enabled and when hardware detects HSE clock stuck at low/high state, internal hardware will automatically switch the system clock to internal high speed RC clock (HSI) to protect the system safety.
3. When the MCTMEN and CKMEN control bits of the CKCU lock protection mode is enabled in the MCTM unit, the bits will be allowed to enable only and inhibited to disable again.

PDMA Request

The MCTM has a PDMA data transfer interface. There are certain events which can generate PDMA requests if the corresponding enable control bits are set to 1 to enable the PDMA access. These events are the MCTM update events, trigger event and channel capture/compare events. When the PDMA request is generated from the MCTM channel, it can be derived from the channel capture/compare event or the MCTM update event 1 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.

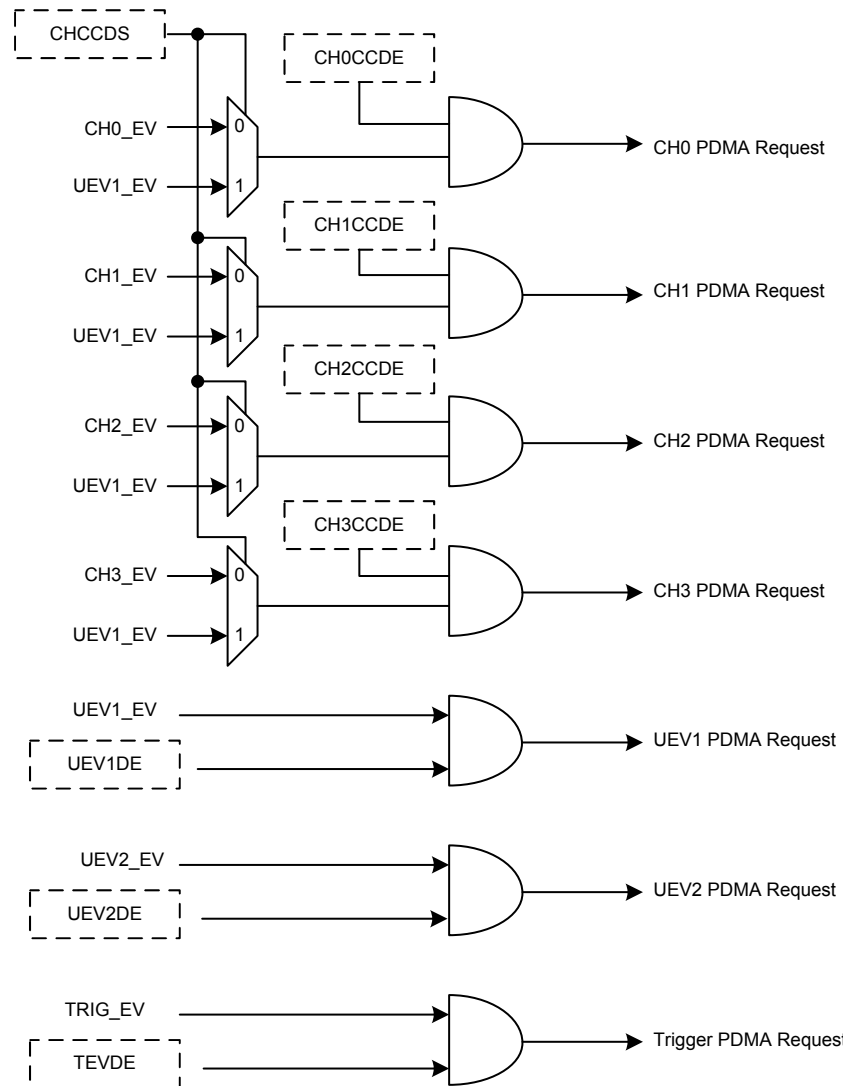


Figure 115. MCTM PDMA Mapping Diagram

Register Map

The following table shows the MCTM registers and reset values.

Table 38. MCTM 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
CHBRKCFR	0x06C	Channel Break Configuration Register	0x0000_0000
CHBRKCTR	0x070	Channel Break Control 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
REPR	0x08C	Timer Repetition Register	0x0000_0000
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 MCTM counter configuration.

Offset: 0x000

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved							DIR	0
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved						CMSEL	0	0
	15	14	13	12	11	10	9	8	
Type/Reset	Reserved						CKDIV	0	0
	7	6	5	4	3	2	1	0	
Type/Reset	Reserved						UGDIS	UEV1DIS	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 counting mode or when used as a Quadrature decoder.
[17:16]	CMSEL	Counter Mode Selection 00: Edge-aligned counting mode. Normal up-counting and down-counting available for this mode. Counting direction is defined by the DIR bit. 01: Center-aligned counting mode 1. The counter counts up and down alternatively. The compare match interrupt flag is set during the count-down period. 10: Center-aligned counting mode 2. The counter counts up and down alternatively. The compare match interrupt flag is set during the count-up period. 11: Center-aligned counting mode 3. The counter counts up and down alternatively. The compare match interrupt flag is set during the count-up and count-down period.
[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 as the 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 1 interrupt generation disable control 0: Any of the following events will generate an update PDMA request or interrupt - Counter overflow / underflow - Setting the UEV1G bit - Update generation through the slave mode 1: Only counter overflow/underflow generates an update PDMA request or interrupt
[0]	UEV1DIS	Update event 1 Disable control 0: Enable the update event 1 request by one of following events - Counter overflow / underflow - Setting the UEV1G bit - Update generation through the slave mode 1: Disable the update event 1 – however the counter and the prescaler are reinitialised if the UEV1G bit is set or if a hardware restart is received from the slave mode

Timer Mode Configuration Register – MDCFR

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

Offset: 0x004

Reset value: 0x0000_0000

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

Bits	Field	Descriptions
[24]	SPMSET	Single Pulse Mode Setting 0: Counter counts normally irrespective of whether a 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 synchronise the other slave timer.</p> <table border="1"> <thead> <tr> <th>MMSEL [2:0]</th> <th>Mode</th> <th>Descriptions</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>Reset Mode</td> <td>The MTO in the Reset mode is an output derived from one of the following cases: 1. Software setting UEV1G bit 2. Slave has trigger input when used in 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 1 is used as the trigger output according to one of the following cases when the UEV1DIS bit is cleared to 0: 1. Counter overflow/underflow 2. Software setting UEV1G 3. Slave has 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 which is used as the master trigger output.</td> </tr> <tr> <td>100</td> <td>Compare output 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 output 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 output 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 output 3</td> <td>The Channel 3 Output reference signal named CH3OREF is used as the trigger output.</td> </tr> </tbody> </table>	MMSEL [2:0]	Mode	Descriptions	000	Reset Mode	The MTO in the Reset mode is an output derived from one of the following cases: 1. Software setting UEV1G bit 2. Slave has trigger input when used in slave restart mode	001	Enable Mode	The Counter Enable signal is used as the trigger output.	010	Update Mode	The update event 1 is used as the trigger output according to one of the following cases when the UEV1DIS bit is cleared to 0: 1. Counter overflow/underflow 2. Software setting UEV1G 3. Slave has 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 which is used as the master trigger output.	100	Compare output 0	The Channel 0 Output reference signal named CH0OREF is used as the trigger output.	101	Compare output 1	The Channel 1 Output reference signal named CH1OREF is used as the trigger output.	110	Compare output 2	The Channel 2 Output reference signal named CH2OREF is used as the trigger output.	111	Compare output 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 in the Reset mode is an output derived from one of the following cases: 1. Software setting UEV1G bit 2. Slave has trigger input when used in slave restart mode																											
001	Enable Mode	The Counter Enable signal is used as the trigger output.																											
010	Update Mode	The update event 1 is used as the trigger output according to one of the following cases when the UEV1DIS bit is cleared to 0: 1. Counter overflow/underflow 2. Software setting UEV1G 3. Slave has 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 which is used as the master trigger output.																											
100	Compare output 0	The Channel 0 Output reference signal named CH0OREF is used as the trigger output.																											
101	Compare output 1	The Channel 1 Output reference signal named CH1OREF is used as the trigger output.																											
110	Compare output 2	The Channel 2 Output reference signal named CH2OREF is used as the trigger output.																											
111	Compare output 3	The Channel 3 Output reference signal named CH3OREF is used as the trigger output.																											

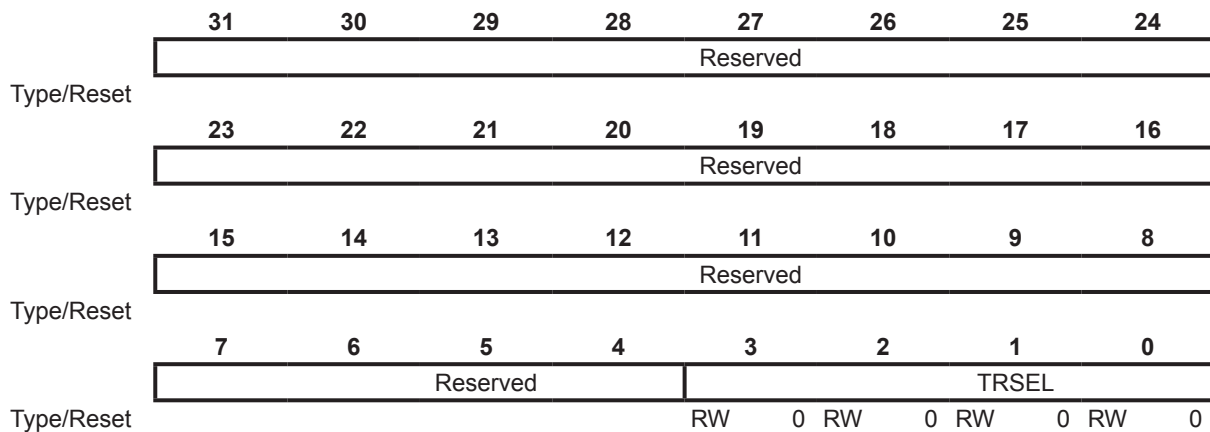
Bits	Field	Descriptions																											
[10:8]	SMSEL	Slave Mode Selection																											
		<table border="1"> <thead> <tr> <th>SMSEL [2:0]</th> <th>Mode</th> <th>Descriptions</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>Disable mode</td> <td>The prescaler is clocked directly by the internal clock.</td> </tr> <tr> <td>001</td> <td>Reserved</td> <td></td> </tr> <tr> <td>010</td> <td>Reserved</td> <td></td> </tr> <tr> <td>011</td> <td>Reserved</td> <td></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 start of counter is controlled.</td> </tr> <tr> <td>111</td> <td>STIED</td> <td>The rising edge of the selected trigger signal STI will be the counter clock.</td> </tr> </tbody> </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 start of counter is controlled.	111	STIED	The rising edge of the selected trigger signal STI will be the counter clock.
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 start of counter is controlled.																											
111	STIED	The rising edge of the selected trigger signal STI will be the counter clock.																											
[0]	TSE	<p>Timer Synchronisation Enable</p> <p>0: No action 1: Master timer (current timer) will generate a delay to synchronise its slave timer through the MTO signal.</p>																											

Timer Trigger Configuration Register – TRCFR

This register specifies the trigger source selection of MCTM.

Offset: 0x008

Reset value: 0x0000_0000



Bits	Field	Descriptions
[3:0]	TRSEL	<p>Trigger Source Selection</p> <p>These bits are used to select the trigger input (STI) for counter synchronising.</p> <ul style="list-style-type: none"> 0000: Software Trigger by setting the UEV1G bit 0001: Channel 0 filtered input – TI0S0 0010: Channel 1 filtered input – TI1S1 0011: Reserved 1000: Channel 0 Edge Detector – TI0BED 1001: Internal Timer Trigger 0 – ITI0 1010: Internal Timer Trigger 1 – ITI1 1011: Internal Timer Trigger 2 – ITI2 Others: Default 0 <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 0x00.</p>

Table 39. MCTM Internal Trigger Connection

Slave Timing Module	ITI0	ITI1	ITI2
MCTM	—	GPTM0	GPTM1

Timer Counter Register – CTR

This register specifies the timer enable bit (TME), CRR buffer enable bit (CRBE), Capture/compare control bit 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	
									RW 0
	15	14	13	12	11	10	9	8	
Type/Reset	Reserved						COMUS	COMPRES	
							RW 0	RW 0	
	7	6	5	4	3	2	1	0	
Type/Reset	Reserved						CRBE	TME	
							RW 0	RW 0	

Bits	Field	Descriptions
[16]	CHCCDS	Channel Capture/Compare PDMA Selection 0: Channel PDMA request derived from the channel capture/compare event. 1: Channel PDMA request derived from the update event 1.
[9]	COMUS	Capture/Compare Control Update Selection 0: Updated by setting the UEV2G bit only 1: Updated by setting the UEV2G bit or when a STI signal rising edge occurs This bit is only available when the capture/compare preload function is enabled by setting the COMPRES bit to 1.
[8]	COMPRES	Capture/Compare Preloaded Enable Control 0: CHxE, CHxNE and CHxOM bits are not preloaded 1: CHxE, CHxNE and CHxOM bits are preloaded If this bit is set to 1, the corresponding capture/compare control bits including the CHxE, CHxNE and CHxOM bits will be updated when the update event 2 occurs.
[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: MCTM off 1: MCTM on – MCTM functions normally When the TME bit is cleared to 0, the counter is stopped and the MCTM consumes no power in any operational 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 MCTM 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
	TI0SRC		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				TI0F			
Type/Reset					RW	0	RW	0

Bits	Field	Descriptions
[31]	TI0SRC	Channel 0 Input Source TI0 Selection 0: The MT_CH0 pin is connected to the channel 0 input TI0 1: The XOR operation output of the MT_CH0, MT_CH1, and MT_CH2 pins are connected to the channel 0 input TI0
[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 TI0 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]	T10F	<p>Channel 0 Input Source T10 Filter Setting</p> <p>These bits define the frequency divided ratio used to sample the T10 signal. The Digital filter in the MCTM 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		
					RW	0	RW	0	RW
					0				0
	15	14	13	12	11	10	9	8	
Type/Reset	Reserved								
	7	6	5	4	3	2	1	0	
Type/Reset	Reserved				T11F				
					RW	0	RW	0	RW
					0				0

Bits	Field	Descriptions
[19:18]	CH1PSC	Channel 1 Capture Input Source Prescaler Setting 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. 00: No prescaler, channel 1 capture input signal is chosen for each active event 01: Channel 1 Capture input signal is chosen for every 2 events 10: Channel 1 Capture input signal is chosen for every 4 events 11: Channel 1 Capture input signal is chosen for every 8 events
[17:16]	CH1CCS	Channel 1 Capture/Compare Selection 00: Channel 1 is configured as an output 01: Channel 1 is configured as an input derived from the T11 signal 10: Channel 1 is configured as an input derived from the T10 signal 11: Channel 1 is configured as an input which comes from the TRCED signal derived from the Trigger Controller Note: The CH1CCS field can be accessed only when the CH1E bit is cleared to 0.

Bits	Field	Descriptions
[3:0]	T11F	<p>Channel 1 Input Source T11 Filter Setting</p> <p>These bits define the frequency divide ratio used to sample the T11 signal. The Digital filter in the MCTM 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 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				CH2PSC		CH2CCS	
					RW	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				TI2F			
					RW	0	RW	0

Bits	Field	Descriptions
[19:18]	CH2PSC	<p>Channel 2 Capture Input Source Prescaler Setting</p> <p>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.</p> <ul style="list-style-type: none"> 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	<p>Channel 2 Capture/Compare Selection</p> <ul style="list-style-type: none"> 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 <p>Note: The CH2CCS field can be accessed only when the CH2E bit is cleared to 0.</p>

Bits	Field	Descriptions
[3:0]	T12F	<p>Channel 2 Input Source T12 Filter Setting</p> <p>These bits define the frequency divide ratio used to sample the T12 signal. The Digital filter in the MCTM 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
					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> <ul style="list-style-type: none"> 00: No prescaler, channel 3 capture input signal is chosen for each active event 01: Channel 3 Capture input signal is chosen for every 2 events 10: Channel 3 Capture input signal is chosen for every 4 events 11: Channel 3 Capture input signal is chosen for every 8 events
[17:16]	CH3CCS	<p>Channel 3 Capture/Compare Selection</p> <ul style="list-style-type: none"> 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>Note: The CH3CCS field can be accessed only when the CH3E bit is cleared to 0.</p>

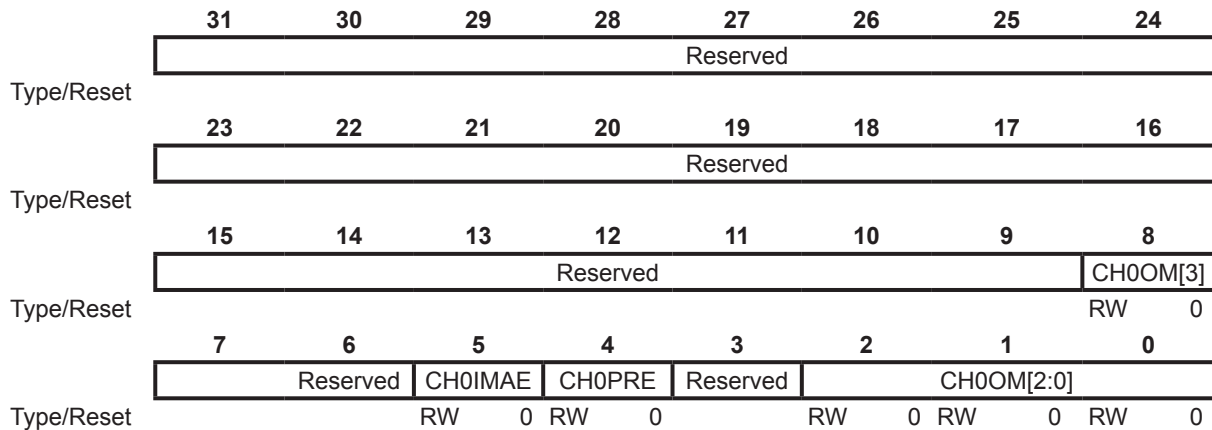
Bits	Field	Descriptions
[3:0]	TI3F	<p>Channel 3 Input Source TI3 Filter Setting</p> <p>These bits define the frequency divide ratio used to sample the TI3 signal. The digital filter in the GPTM is an N-event counter where N is defined as how many valid transitions are necessary to output a filtered signal</p> <p>0000: No filter, the sampling clock is f_{SYSTEM}.</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 0 Output Configuration Register – CH0OCFR

This register specifies the channel 0 output mode configuration.

Offset: 0x040

Reset value: 0x0000_0000



Bits	Field	Descriptions
[5]	CH0IMAE	Channel 0 Immediate Active Enable 0: No action 1: Single pulse Immediate Active Mode is enabled CH0OREF 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 channel 0 is configured operate in 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 an update event 1 occurs.

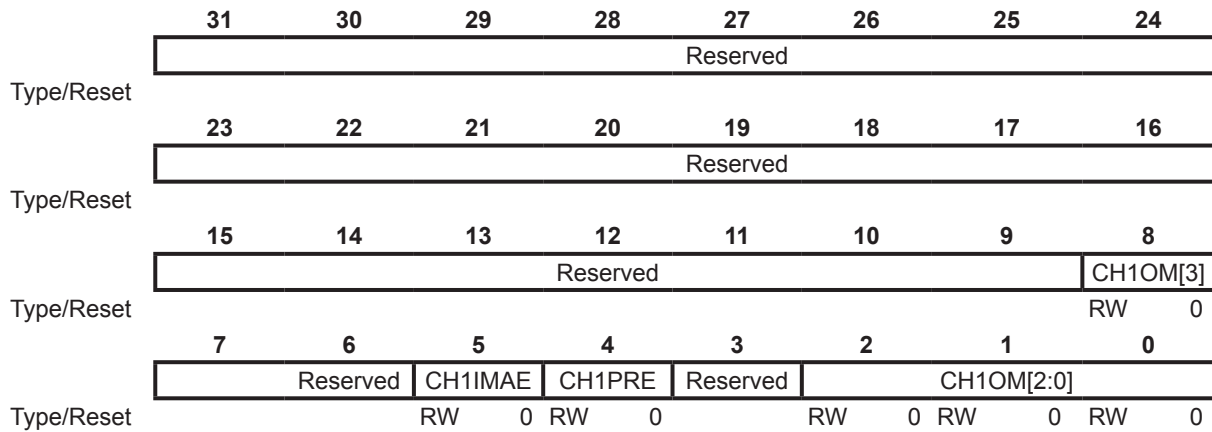
Bits	Field	Descriptions
[8][2:0]	CH0OM[3:0]	<p>Channel 0 Output Mode Setting</p> <p>These bits define the functional types of the output reference signal CH0OREF.</p> <p>0000: No Change</p> <p>0001: Output 0 on compare match</p> <p>0010: Output 1 on compare match</p> <p>0011: Output toggles on compare 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 > 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 < CH0CCR 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 = 01/02/03)</p>

Channel 1 Output Configuration Register – CH1OCFR

This register specifies the channel 1 output mode configuration.

Offset: 0x044

Reset value: 0x0000_0000



Bits	Field	Descriptions
[5]	CH1IMAE	<p>Channel 1 Immediate Active Enable</p> <p>0: No action 1: Single pulse Immediate Active Mode is enabled</p> <p>The CH1OREF 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.</p> <p>The effective duration ends automatically at the next overflow or underflow event.</p> <p>Note: The CH1IMAE bit is available only if channel 1 is configured to be operated in PWM mode 1 or PWM mode 2.</p>
[4]	CH1PRE	<p>Channel 1 Capture/Compare Register (CH1CCR) Preload Enable</p> <p>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.</p> <p>1: CH1CCR preload function is enabled The new CH1CCR value will not be transferred to its shadow register until an update event 1 occurs.</p>

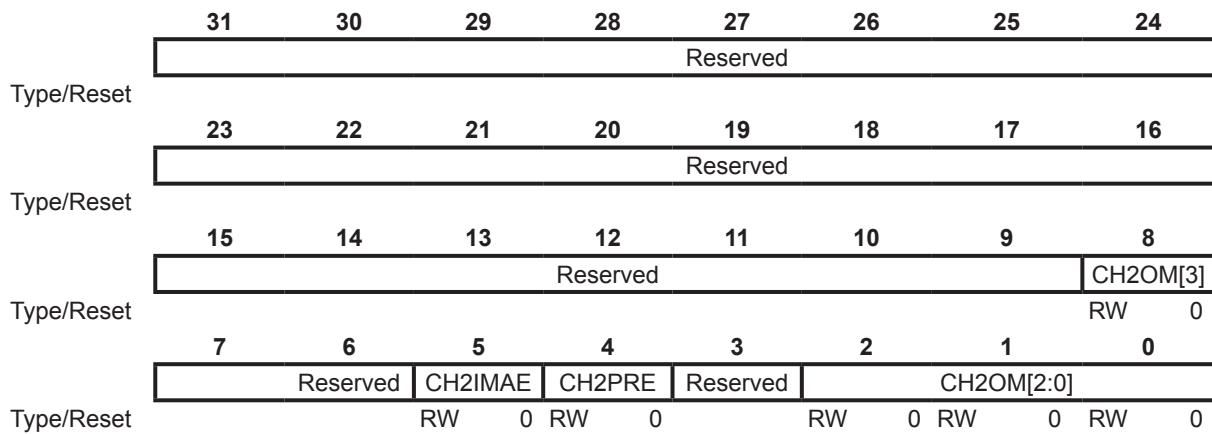
Bits	Field	Descriptions
[8][2:0]	CH1OM[3:0]	<p>Channel 1 Output Mode Setting</p> <p>These bits define the functional types of the output reference signal CH1OREF.</p> <p>0000: No Change</p> <p>0001: Output 0 on compare match</p> <p>0010: Output 1 on compare match</p> <p>0011: Output toggles on compare 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 = 01/02/03)</p>

Channel 2 Output Configuration Register – CH2OCFR

This register specifies the channel 2 output mode configuration.

Offset: 0x048

Reset value: 0x0000_0000



Bits	Field	Descriptions
[5]	CH2IMAE	<p>Channel 2 Immediate Active Enable</p> <p>0: No action 1: Single pulse Immediate Active Mode is enabled</p> <p>The CH2OREF 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.</p> <p>The effective duration ends automatically at the next overflow or underflow event.</p> <p>Note: The CH2IMAE bit is available only if the channel 2 is configured to be operated in PWM mode 1 or PWM mode 2.</p>
[4]	CH2PRE	<p>Channel 2 Capture/Compare Register (CH2CCR) Preload Enable</p> <p>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.</p> <p>1: CH2CCR preload function is enabled The new CH2CCR value will not be transferred to its shadow register until an update event 1 occurs.</p>

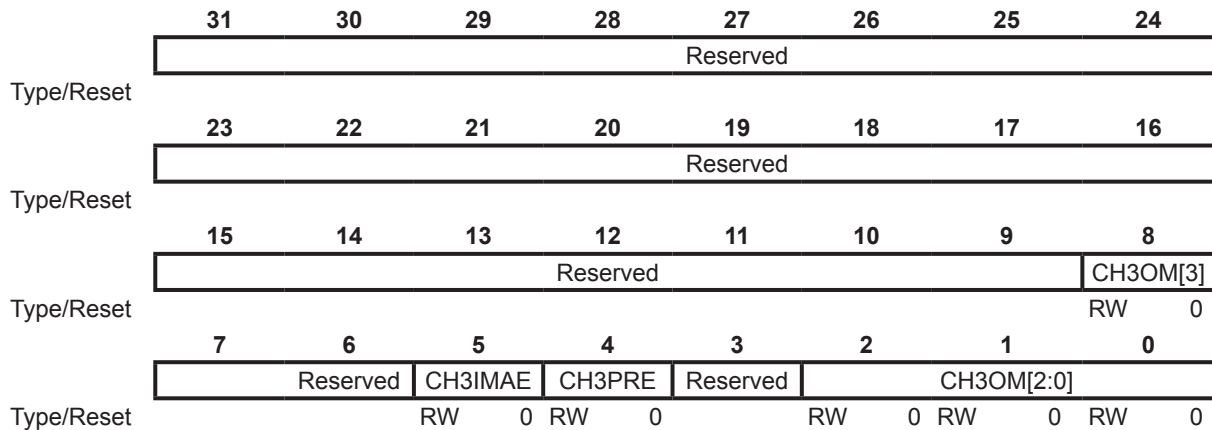
Bits	Field	Descriptions
[8][2:0]	CH2OM[3:0]	<p>Channel 2 Output Mode Setting</p> <p>These bits define the functional types of the output reference signal CH2OREF.</p> <p>0000: No Change</p> <p>0001: Output 0 on compare match</p> <p>0010: Output 1 on compare match</p> <p>0011: Output toggles on compare 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 = 01/02/03)</p>

Channel 3 Output Configuration Register – CH3OCFR

This register specifies the channel 3 output mode configuration.

Offset: 0x04C

Reset value: 0x0000_0000



Bits	Field	Descriptions
[5]	CH3IMAE	<p>Channel 3 Immediate Active Enable</p> <p>0: No action 1: Single pulse Immediate Active Mode is enabled</p> <p>The CH3OREF 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.</p> <p>The effective duration ends automatically at the next overflow or underflow event.</p> <p>Note: The CH3IMAE bit is available only if channel 3 is configured to be operated in PWM mode 1 or PWM mode 2.</p>
[4]	CH3PRE	<p>Channel 3 Capture/Compare Register (CH3CCR) Preload Enable</p> <p>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.</p> <p>1: CH3CCR preload function is enabled The new CH3CCR value will not be transferred to its shadow register until an update event 1 occurs.</p>

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 = 01/02/03)</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	CH2NE	CH2E	CH1NE	CH1E	CH0NE	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 = 0x01/0x02/0x03) <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 = 0x00) <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 depending on the condition of the CHMOE, CHOSS1, CHOSSR and CH3OIS bits.
[5]	CH2NE	<p>Channel 2 Capture/Compare Complementary Enable</p> <ul style="list-style-type: none"> 0: Off – Channel 2 complementary output CH2NO is not active. The CH2NO level is then determined by the CHMOE, CHOSS1, CHOSSR, CH2OIS, CH2OISN and CH2E bits. 1: On – Channel 2 complementary output CH2NO is generated on the corresponding output pin depending on the condition of the CHMOE, CHOSS1, CHOSSR, CH2OIS, CH2OISN and CH2E bits.
[4]	CH2E	<p>Channel 2 Capture/Compare Enable</p> <ul style="list-style-type: none"> - Channel 2 is configured as an input (CH2CCS = 0x01/0x02/0x03) <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 = 0x00) <ul style="list-style-type: none"> 0: Off – Channel 2 output signal CH2O is not active. The CH2O level is then determined by the condition of the CHMOE, CHOSS1, CHOSSR, CH2OIS, CH2OISN and CH2NE bits. 1: On – Channel 2 output signal CH2O is generated on the corresponding output pin determined by the condition of the CHMOE, CHOSS1, CHOSSR, CH2OIS, CH2OISN and CH2NE bits.

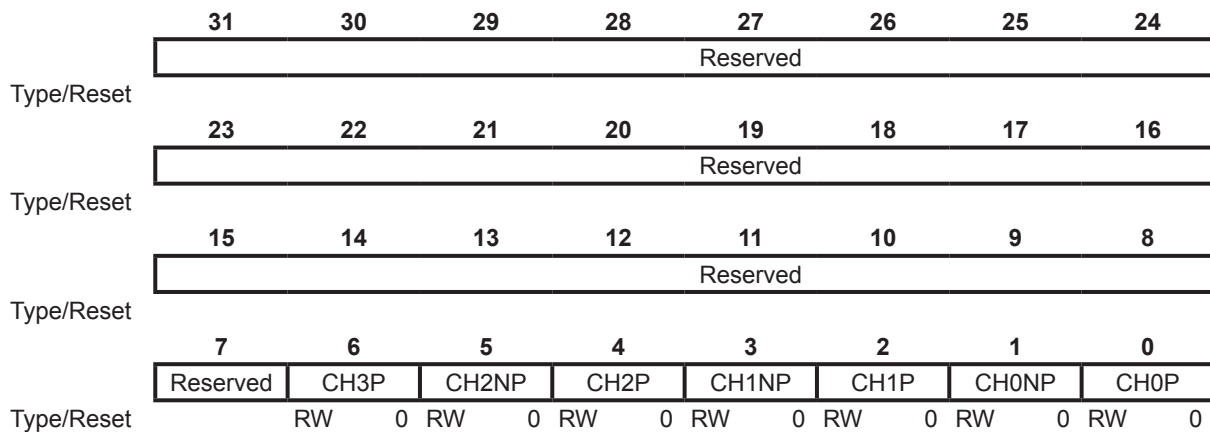
Bits	Field	Descriptions
[3]	CH1NE	<p>Channel 1 Capture/Compare Complementary Enable</p> <p>0: Off – Channel 1 complementary output CH1NO is not active. The CH1NO level is then determined by the condition of the CHMOE, CHOSSI, CHOSSR, CH1OIS, CH1OISN and CH1E bits.</p> <p>1: On – Channel 1 complementary output CH1NO is generated on the corresponding output pin determined by the condition of the CHMOE, CHOSSI, CHOSSR, CH1OIS, CH1OISN and CH1E bits.</p>
[2]	CH1E	<p>Channel 1 Capture/Compare Enable</p> <p>- Channel 1 is configured as an input (CH1CCS = 0x01/0x02/0x03)</p> <p>0: Input Capture Mode is disabled</p> <p>1: Input Capture Mode is enabled</p> <p>- Channel 1 is configured as an output (CH1CCS = 0x00)</p> <p>0: Off – Channel 1 output signal CH1O is not active. The CH1O level is then determined by the condition of the CHMOE, CHOSSI, CHOSSR, CH1OIS, CH1OISN and CH1NE bits.</p> <p>1: On – Channel 1 output signal CH1O is generated on the corresponding output pin depending on the condition of the CHMOE, CHOSSI, CHOSSR, CH1OIS, CH1OISN and CH1NE bits.</p>
[1]	CH0NE	<p>Channel 0 Capture/Compare Complementary Enable</p> <p>0: Off – Channel 0 complementary output CH0NO is not active. The CH0NO level is then determined by the condition of the CHMOE, CHOSSI, CHOSSR, CH0OIS, CH0OISN and CH0E bits.</p> <p>1: On – Channel 0 complementary output CH0NO is generated on the corresponding output pin depending on the condition of the CHMOE, CHOSSI, CHOSSR, CH0OIS, CH0OISN and CH0E bits.</p>
[0]	CH0E	<p>Channel 0 Capture/Compare Enable</p> <p>- Channel 0 is configured as an input (CH0CCS = 0x01/0x02/0x03)</p> <p>0: Input Capture Mode is disabled</p> <p>1: Input Capture Mode is enabled</p> <p>- Channel 0 is configured as an output (CH0CCS = 0x00)</p> <p>0: Off – Channel 0 output signal CH0O is not active. The CH0O level is then determined by the condition of the CHMOE, CHOSSI, CHOSSR, CH0OIS, CH0OISN and CH0NE bits.</p> <p>1: On – Channel 0 output signal CH0O is generated on the corresponding output pin determined by the condition of the CHMOE, CHOSSI, CHOSSR, CH0OIS, CH0OISN and CH0NE bits.</p>

Channel Polarity Configuration Register – CHPOLR

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

Offset: 0x054

Reset value: 0x0000_0000



Bits	Field	Descriptions
[6]	CH3P	Channel 3 Capture/Compare Polarity - When Channel 3 is configured as an input 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 = 0x00) 0: Channel 3 Output is active high 1: Channel 3 Output is active low
[5]	CH2NP	Channel 2 Capture/Compare Complementary Polarity 0: Channel 2 Output is active high. 1: Channel 2 Output is active low
[4]	CH2P	Channel 2 Capture/Compare Polarity - When Channel 2 is configured as an input 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 = 0x00) 0: Channel 2 Output is active high 1: Channel 2 Output is active low
[3]	CH1NP	Channel 1 Capture/Compare Complementary Polarity 0: Channel 1 Output is active high. 1: Channel 1 Output is active low.
[2]	CH1P	Channel 1 Capture/Compare Polarity - When Channel 1 is configured as an input 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 = 0x00) 0: Channel 1 Output is active high 1: Channel 1 Output is active low
[1]	CH0NP	Channel 0 Capture/Compare Complementary Polarity 0: Channel 0 Output is active high. 1: Channel 0 Output is active low.

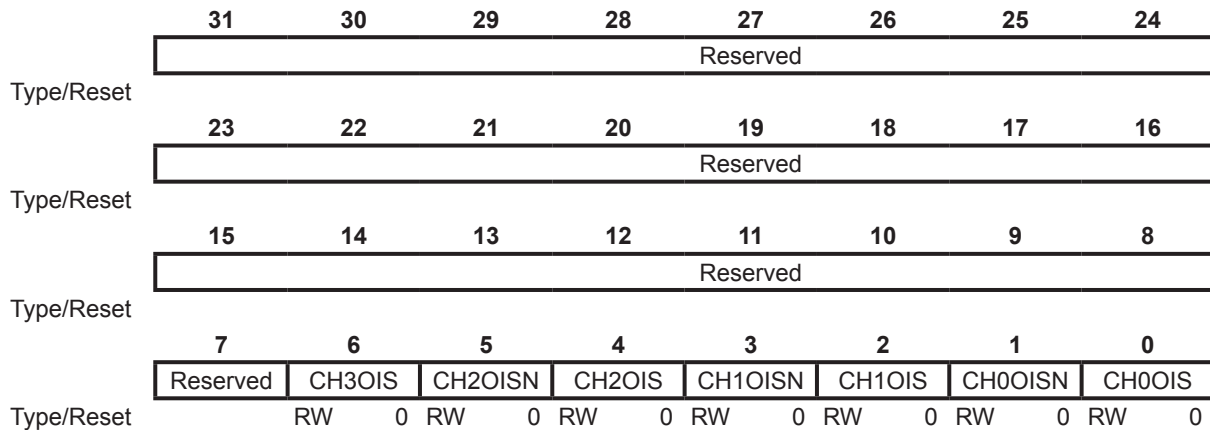
Bits	Field	Descriptions
[0]	CH0P	Channel 0 Capture/Compare Polarity - When Channel 0 is configured as an input 0: capture event occurs on a Channel 0 rising edge 1: capture event occurs on a Channel 0 falling edge - When Channel 0 is configured as an output (CH0CCS = 0x00) 0: Channel 0 Output is active high 1: Channel 0 Output is active low

Channel Break Configuration Register – CHBRKCFR

This register specifies the channel output idle state when using the break function.

Offset: 0x06C

Reset value: 0x0000_0000



Bits	Field	Descriptions
[6]	CH3OIS	MT_CH3O Output Idle State 0: Channel 3 output CH3O = 0 when CHMOE = 0. 1: Channel 3 output CH3O = 1 when CHMOE = 0.
[5]	CH2OISN	MT_CH2NO Output Idle State 0: Channel 2 complementary output CH2NO=0 after a dead time when CHMOE=0 1: Channel 2 complementary output CH2NO=1 after a dead time when CHMOE=0
[4]	CH2OIS	MT_CH2O Output Idle State 0: Channel 2 output CH2O = 0 after a dead time when CHMOE = 0 1: Channel 2 output CH2O = 1 after a dead time when CHMOE = 0
[3]	CH1OISN	MT_CH1NO Output Idle State 0: Channel 1 complementary output CH1NO=0 after a dead time when CHMOE=0 1: Channel 1 complementary output CH1NO=1 after a dead time when CHMOE=0
[2]	CH1OIS	MT_CH1O Output Idle State 0: Channel 1 output CH1O = 0 after a dead time when CHMOE = 0 1: Channel 1 output CH1O = 1 after a dead time when CHMOE = 0
[1]	CH0OISN	MT_CH0NO Output Idle State 0: Channel 0 complementary output CH1NO=0 after a dead time when CHMOE=0 1: Channel 0 complementary output CH1NO=1 after a dead time when CHMOE=0
[0]	CH0OIS	MT_CH0O Output Idle State 0: Channel 0 output CH0O = 0 after a dead time when CHMOE = 0 1: Channel 0 output CH0O = 1 after a dead time when CHMOE = 0

Channel Break Control Register – CHBRKCTR

This register specifies the channel break control bits.

Offset: 0x070

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24	
	CHDTG								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	Reserved		CHOSSR	CHOSSE	Reserved		GFSEL	LOCKLV	
Type/Reset			RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	Reserved				BKF				
Type/Reset					RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	Reserved		CHAOE	CHMOE	Reserved		BKP	BKE	
Type/Reset			RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[31:24]	CHDTG	Channel Dead Time Duration Definition CHDTG[7:5]=0xx: Channel Dead Time=CHDTG [7:0]×t _{dtg} , with t _{dtg} =t _{DTS} CHDTG[7:5]=10x: Channel Dead Time=(64 + CHDTG [5:0])×t _{dtg} , with t _{dtg} =2×t _{DTS} CHDTG[7:5]=110: Channel Dead Time=(32 + CHDTG [4:0])×t _{dtg} , with t _{dtg} =8×t _{DTS} CHDTG[7:5]=111: Channel Dead Time=(32 + CHDTG [4:0])×t _{dtg} , with t _{dtg} =16×t _{DTS}
[21]	CHOSSR	Channel Off State (CHxNE=0) Selection for Normal Run State (CHMOE = 1) 0: When inactive, MT_CHxO/MT_CHxNO output disable - not driven by timer 1: When inactive, MT_CHxO/MT_CHxNO output enabled with their inactive level
[20]	CHOSSE	Channel Off State Selection for Idle Mode (CHMOE = 0) 0: When inactive, MT_CHxO/MT_CHxNO output disable - not driven by timer 1: When inactive, MT_CHxO/MT_CHxNO output enabled with their idle level depending upon the condition of the the CHxOIS and CHxOISN bits.
[18]	GFSEL	Deglitch Filter Selction for Break 0: no input deglitch filter 1: 50ns deglitch filter
[17:16]	LOCKLV	Lock Level Setting These bits offer write protection against software errors. The bits can be written only once after a reset. 00: LOCK OFF. Register write protected function disabled. 01: LOCK Level 1 10: LOCK Level 2 11: LOCK Level 3

Bits	Field	Descriptions
[11:8]	BKF	<p>Break Input Filter Setting</p> <p>These bits define the frequency ratio used to sample the MT_BRK signal. The digital filter in the MCTM 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 – don't need sample clock. 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>
[5]	CHAOE	<p>Channel Automatic Output Enable</p> <p>0: CHMOE can be set only by software. 1: CHMOE can be set by software or automatically by an update event</p>
[4]	CHMOE	<p>Channel Main Output Enable</p> <p>Cleared asynchronously by hardware on a break event occurrence.</p> <p>0: MT_CHxO and MT_CHxNO are disabled or forced to idle states. 1: MT_CHxO and MT_CHxNO are enabled if the enable bits (CHxE, CHxNE) are set.</p>
[1]	BKP	<p>Break Input Polarity.</p> <p>0: Break input active low 1: Break input active high</p>
[0]	BKE	<p>Break Enable</p> <p>0: Break inputs disabled 1: Break inputs enabled</p>

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
Type/Reset	Reserved					TEVDE	UEV2DE	UEV1DE
						RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
Type/Reset	Reserved				CH3CCDE	CH2CCDE	CH1CCDE	CH0CCDE
					RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
Type/Reset	Reserved				BRKIE	TEVIE	UEV2IE	UEV1IE
					RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
Type/Reset	Reserved				CH3CCIE	CH2CCIE	CH1CCIE	CH0CCIE
					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
[25]	UEV2DE	Update event 2 PDMA Request Enable 0: Update event 2 PDMA request is disabled 1: Update event 2 PDMA request is enabled
[24]	UEV1DE	Update event 1 DMA Request Enable 0: Update event 1 PDMA request is disabled 1: Update event 1 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
[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
[11]	BRKIE	Break event Interrupt Enable 0: Break event interrupt is disabled 1: Break event interrupt is enabled
[10]	TEVIE	Trigger event Interrupt Enable 0: Trigger event interrupt is disabled 1: Trigger event interrupt is enabled
[9]	UEV2IE	Update event 2 Interrupt Enable 0: Update event 2 interrupt is disabled 1: Update event 2 interrupt is enabled

Bits	Field	Descriptions
[8]	UEV1IE	Update event 1 Interrupt Enable 0: Update event 1 interrupt is disabled 1: Update event 1 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				BRKG	TEVG	UEV2G	UEV1G	
					WO	0	WO	0	WO
					0		0		0
	7	6	5	4	3	2	1	0	
Type/Reset	Reserved				CH3CCG	CH2CCG	CH1CCG	CH0CCG	
					WO	0	WO	0	WO
					0		0		0

Bits	Field	Descriptions
[11]	BRKG	Software Break Event Generation The break event BEV can be generated by setting this bit. It is automatically cleared by hardware. 0: No action. 1: The BRKIF flag is set and then the CHMOE bit will be cleared.
[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: The TEVIF flag is set
[9]	UEV2G	Update Event 2 Generation The update event 2 UEV2 can be generated by setting this bit. It is cleared by hardware automatically. 0: No action 1: Update the CHxE, CHxNE, and CHxOM bits when COMPRE bit in CTR Register is set to 1.
[8]	UEV1G	Update Event 1 Generation The update event 1 UEV1 can be generated by setting this bit. It is cleared by hardware automatically. 0: No action 1: Reinitialise 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 on any related registers will also be executed. For a more detailed description, refer to the corresponding section.

Bits	Field	Descriptions
[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>
[1]	CH1CCG	<p>Channel 1 Capture/Compare Generation</p> <p>A Channel 1 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 1</p> <p>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.</p>
[0]	CH0CCG	<p>Channel 0 Capture/Compare Generation</p> <p>A Channel 0 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 0</p> <p>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.</p>

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				BRKIF	TEVIF	UEV2IF	UEV1IF	
					W0C	0	W0C	0	W0C
					0	W0C	0	W0C	0
	7	6	5	4	3	2	1	0	
Type/Reset	CH3OCF	CH2OCF	CH1OCF	CH0OCF	CH3CCIF	CH2CCIF	CH1CCIF	CH0CCIF	
	W0C	0	W0C	0	W0C	0	W0C	0	W0C
	0	W0C	0	W0C	0	W0C	0	W0C	0

Bits	Field	Descriptions
[11]	BRKIF	Break Event Interrupt Flag This flag is set by hardware when a break event occurs and is cleared by software. 0: No break event occurred 1: Break event occurred
[10]	TEVIF	Trigger Event Interrupt Flag This flag is set by hardware when a trigger event occurs and is cleared by software. 0: No trigger event occurred 1: Trigger event occurred
[9]	UEV2IF	Update Event 2 Interrupt Flag This bit is set by hardware when an update event 2 occurs and is cleared by software. 0: No update event 2 occurred 1: Update event 2 occurred
[8]	UEV1IF	Update Event 1 Interrupt Flag This bit is set by hardware when a update event 1 occurs and is cleared by software. 0: No update event 1 occurred 1: Update event 1 occurred Note: The update event 1 is sourced from the following conditions: - A counter overflow or underflow - The UEV1G bit is set with UEV1DIS=0 - A STI rising edge is received in slave restart mode with UEV1DIS=0
[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.

Bits	Field	Descriptions
[6]	CH2OCF	<p>Channel 2 Over-capture Flag</p> <p>This flag is set by hardware and cleared by software.</p> <p>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.</p>
[5]	CH1OCF	<p>Channel 1 Over-capture Flag</p> <p>This flag is set by hardware and cleared by software.</p> <p>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.</p>
[4]	CH0OCF	<p>Channel 0 Over-capture Flag</p> <p>This flag is set by hardware and cleared by software.</p> <p>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</p>
[3]	CH3CCIF	<p>Channel 3 Capture/Compare Interrupt Flag</p> <ul style="list-style-type: none"> - Channel 3 is configured as an output <ul style="list-style-type: none"> 0: No match event occurred 1: The contents of the counter CNTR have matched the contents of the CH3CCR register. <p>This flag is set by hardware when the counter value matches the CH3CCR value with exception in the center-aligned counting mode. It is cleared by software.</p> - Channel 3 is configured as an input <ul style="list-style-type: none"> 0: No input capture occurred 1: Input capture occurred <p>This bit is set by hardware when a capture event occurs. It is cleared by software or by reading the CH3CCR register.</p>
[2]	CH2CCIF	<p>Channel 2 Capture/Compare Interrupt Flag</p> <ul style="list-style-type: none"> - Channel 2 is configured as an output <ul style="list-style-type: none"> 0: No match event occurred 1: The contents of the counter CNTR have matched the contents of the CH2CCR register <p>This flag is set by hardware when the counter value matches the CH2CCR value with exception in the center-aligned counting mode. It is cleared by software.</p> - Channel 2 is configured as an input <ul style="list-style-type: none"> 0: No input capture occurred 1: Input capture occurred. <p>This bit is set by hardware on a capture event. It is cleared by software or by reading the CH2CCR register.</p>
[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 occurred 1: The contents of the counter CNTR have matched the contents of the CH1CCR register <p>This flag is set by hardware when the counter value matches the CH1CCR value with exception in the center-aligned counting mode. It is cleared by software.</p> - Channel 1 is configured as an input <ul style="list-style-type: none"> 0: No input capture occurred 1: Input capture occurred <p>This bit is set by hardware on a capture event. It is cleared by software or by reading the CH1CCR register.</p>

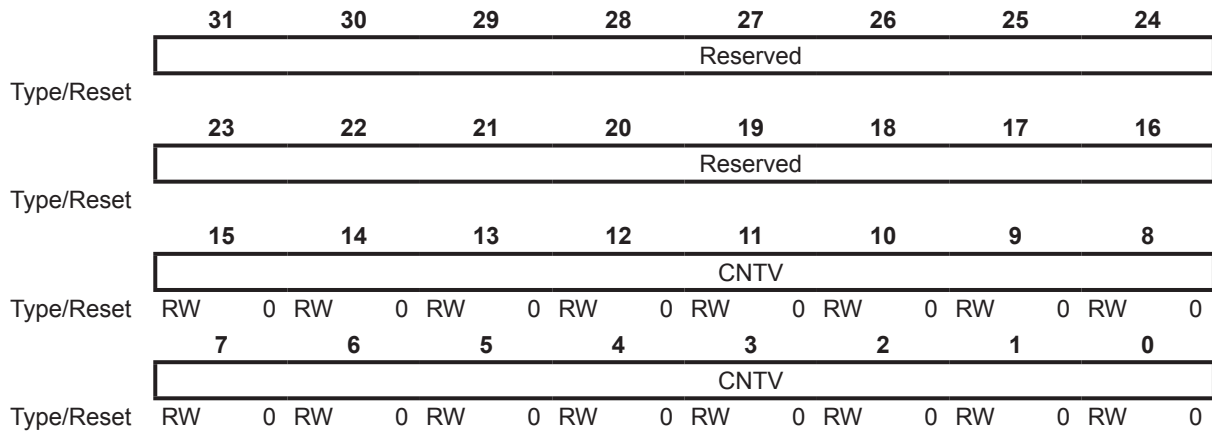
Bits	Field	Descriptions
[0]	CH0CCIF	<p>Channel 0 Capture/Compare Interrupt Flag</p> <ul style="list-style-type: none">- Channel 0 is configured as an output<ul style="list-style-type: none">0: No match event occurs1: The contents of the counter CNTR have matched the content of the CH0CCR registerThis flag is set by hardware when the counter value matches the CH0CCR value with exception in the center-aligned counting mode. It is cleared by software.- Channel 0 is configured as an input<ul style="list-style-type: none">0: No input capture occurred1: Input capture occurredThis bit is set by hardware on a capture event. It is cleared by software or by reading the CH0CCR register.

Timer Counter Register – CNTR

This register stores the timer counter value.

Offset: 0x080

Reset value: 0x0000_0000



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	Reserved							
Type/Reset																
	23	22	21	20	19	18	17	16	Reserved							
Type/Reset																
	15	14	13	12	11	10	9	8	PSCV							
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	PSCV							
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

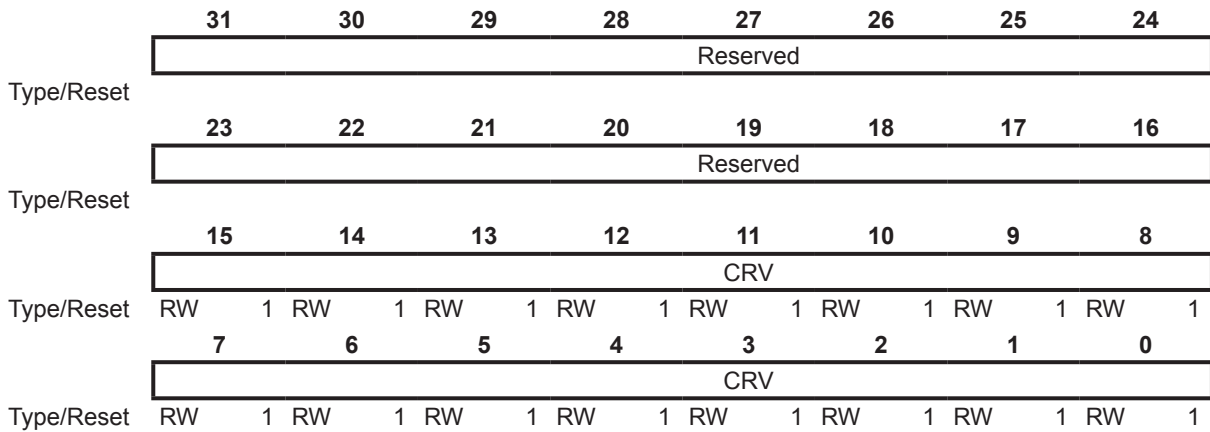
Bits	Field	Descriptions
[15:0]	PSCV	<p>Prescaler Value</p> <p>These bits are used to specify the prescaler value to generate the counter clock frequency f_{CK_CNT}.</p> $f_{CK_CNT} = \frac{f_{CK_PSC}}{PSCV[15:0] + 1}$ <p>where the f_{CK_PSC} is the prescaler input clock source.</p>

Timer Counter Reload Register – CRR

This register specifies the timer counter reload value.

Offset: 0x088

Reset value: 0x0000_FFFF



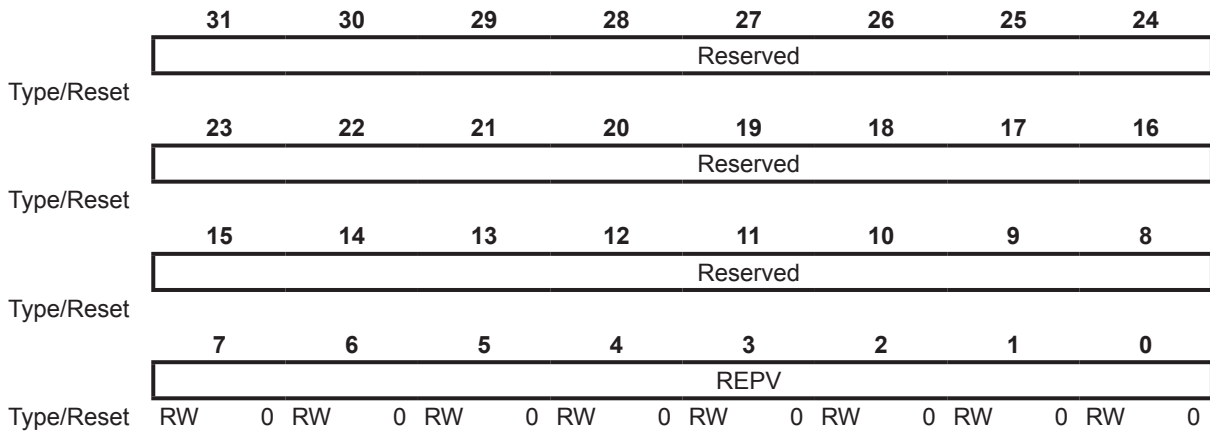
Bits	Field	Descriptions
[15:0]	CRV	Counter Reload Value The CRV is the reload value which is loaded into the actual counter register.

Timer Repetition Register – REPR

This register specifies the timer repetition counter value.

Offset: 0x08C

Reset value: 0x0000_0000



Bits	Field	Descriptions
[7:0]	REPV	Repetition Counter Value. These bits allow the user to specify the update rate of the compare registers.

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	Reserved							
Type/Reset																
	23	22	21	20	19	18	17	16	Reserved							
Type/Reset																
	15	14	13	12	11	10	9	8	CH0CCV							
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	CH0CCV							
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[15:0]	CH0CCV	Channel 0 Capture/Compare Value - 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	Reserved							
Type/Reset																
	23	22	21	20	19	18	17	16	Reserved							
Type/Reset																
	15	14	13	12	11	10	9	8	CH1CCV							
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	CH1CCV							
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

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	Reserved							
Type/Reset																
	23	22	21	20	19	18	17	16	Reserved							
Type/Reset																
	15	14	13	12	11	10	9	8	CH2CCV							
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	CH2CCV							
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

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	Reserved							
Type/Reset																
	23	22	21	20	19	18	17	16	Reserved							
Type/Reset																
	15	14	13	12	11	10	9	8	CH3CCV							
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	CH3CCV							
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

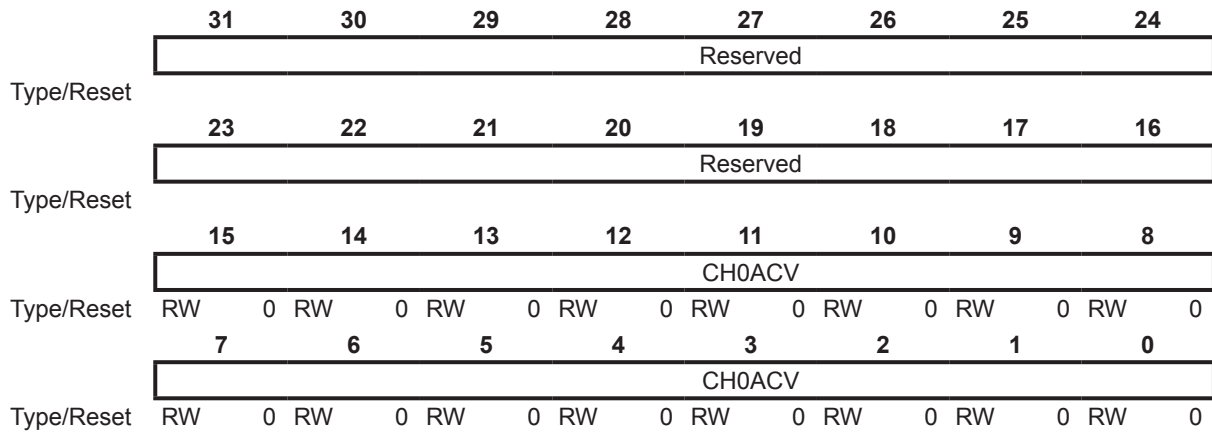
Bits	Field	Descriptions
[15:0]	CH3CCV	<p>Channel 3 Capture/Compare Value</p> <ul style="list-style-type: none"> - When Channel 3 is configured as an output The CH3CCR value is compared with the counter value and the comparison result is used to trigger the CH3OREF output signal. - When Channel 3 is configured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



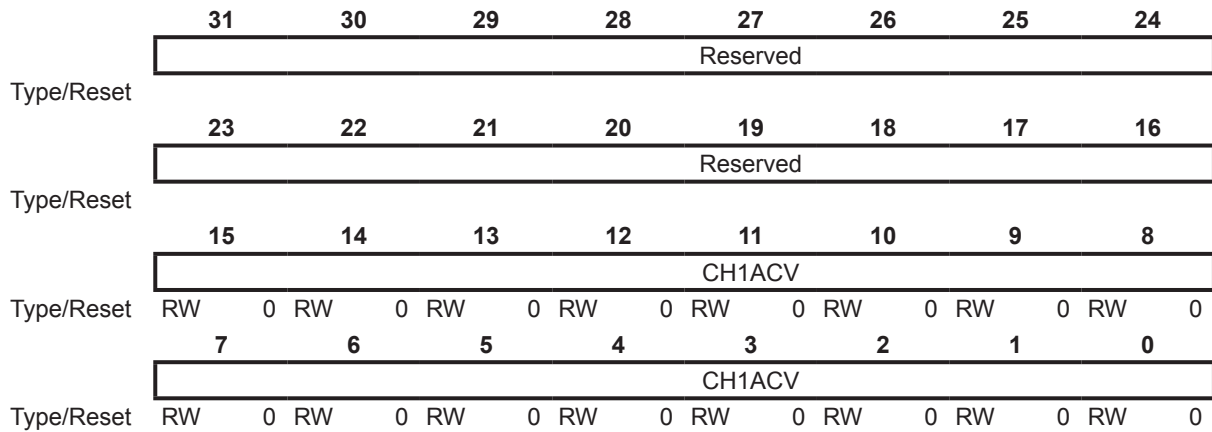
Bits	Field	Descriptions
[15:0]	CH0ACV	Channel 0 Asymmetric Compare Value When channel 0 is configured as asymmetric PWM mode and the counter is counting down, the value written into this register will be compared to the counter.

Channel 1 Asymmetric Compare Register – CH1ACR

This register specifies the timer channel 1 asymmetric compare value.

Offset: 0x0A4

Reset value: 0x0000_0000



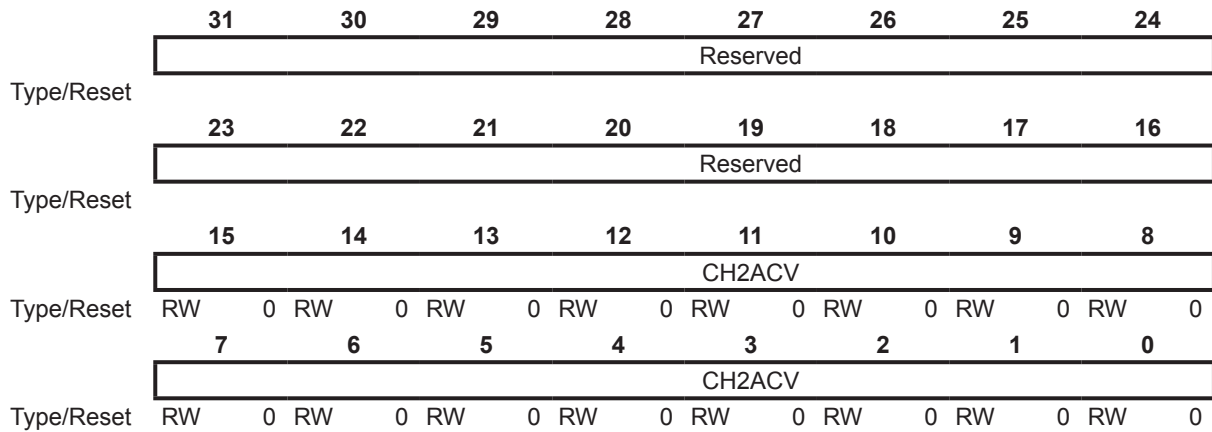
Bits	Field	Descriptions
[15:0]	CH1ACV	Channel 1 Asymmetric Compare Value When channel 1 is configured as asymmetric PWM mode and the counter is counting down, the value written into this register will be compared to the counter.

Channel 2 Asymmetric Compare Register – CH2ACR

This register specifies the timer channel 2 asymmetric compare value.

Offset: 0x0A8

Reset value: 0x0000_0000



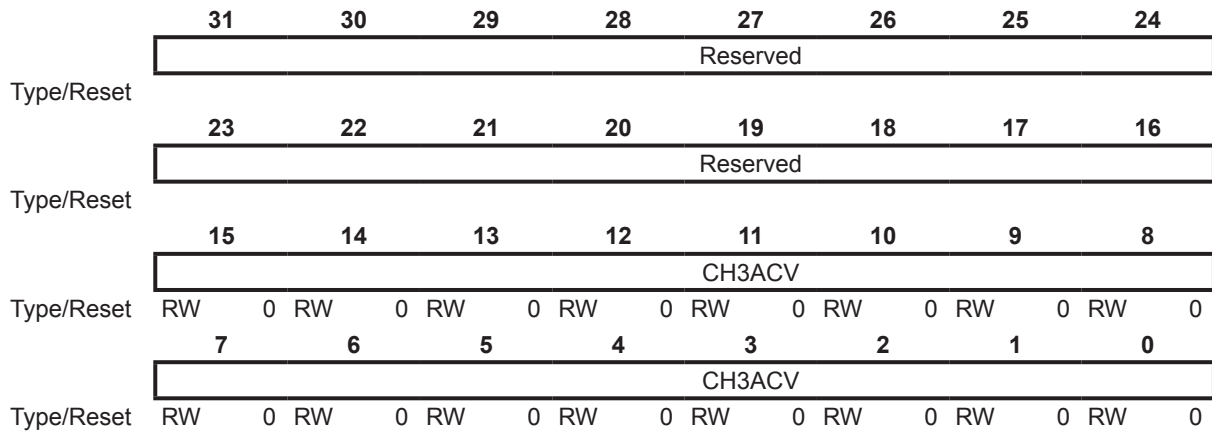
Bits	Field	Descriptions
[15:0]	CH2ACV	Channel 2 Asymmetric Compare Value When channel 2 is configured as asymmetric PWM mode and the counter is counting down, the value written into this register will be compared to the counter.

Channel 3 Asymmetric Compare Register – CH3ACR

This register specifies the timer channel 3 asymmetric compare value.

Offset: 0x0AC

Reset value: 0x0000_0000



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 into this register will be compared to the counter.

17 Single-Channel Timer (SCTM)

Introduction

The Single-Channel Timer consists of one 16-bit up-counter, one 16-bit Capture/Compare Register (CCR), 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.

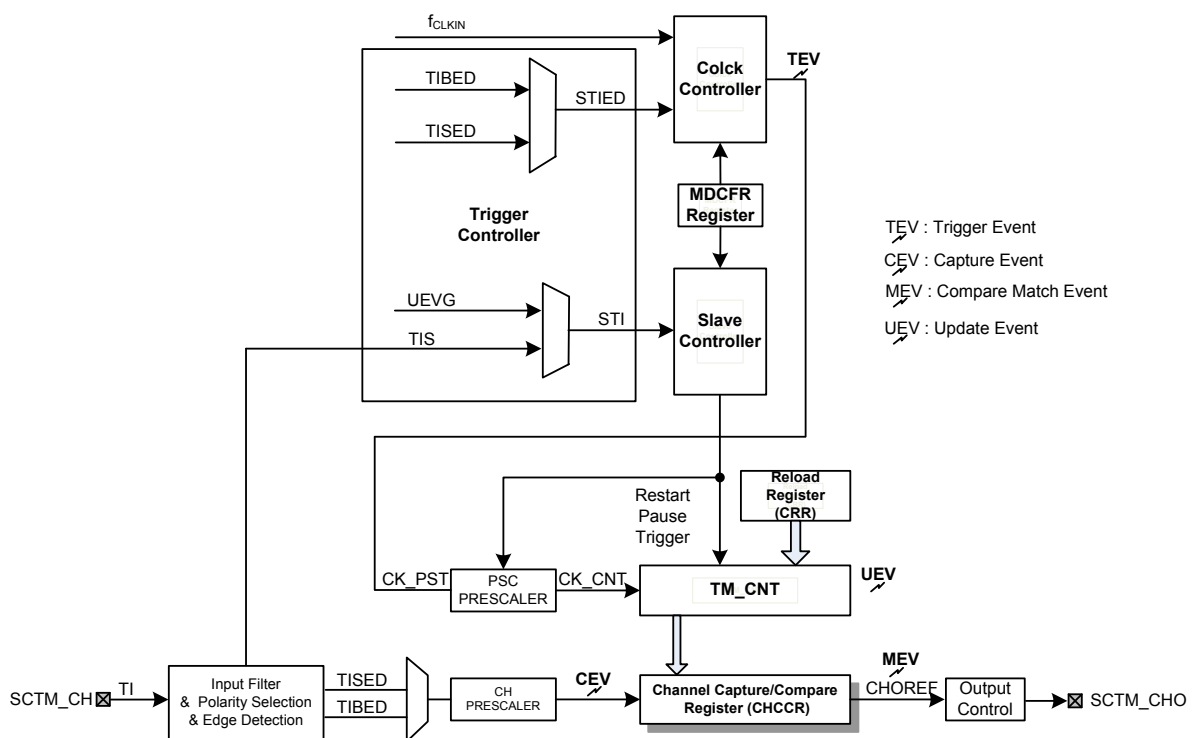


Figure 116. SCTM Block Diagram

Features

- 16-bit auto-reload up counter
- 16-bit programmable prescaler that allows division of the counter clock frequency by any factor between 1 and 65536
- Single channel for:
 - Input Capture function
 - Compare Match Output
 - PWM waveform output
- Interrupt generation with the following events:
 - Update event
 - Trigger event
 - Input capture event
 - Output compare match event

Functional Descriptions

Counter Mode

Up-Counting

The counter counts continuously from 0 to the counter-reload value, which is defined in the CRR register. 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. 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.

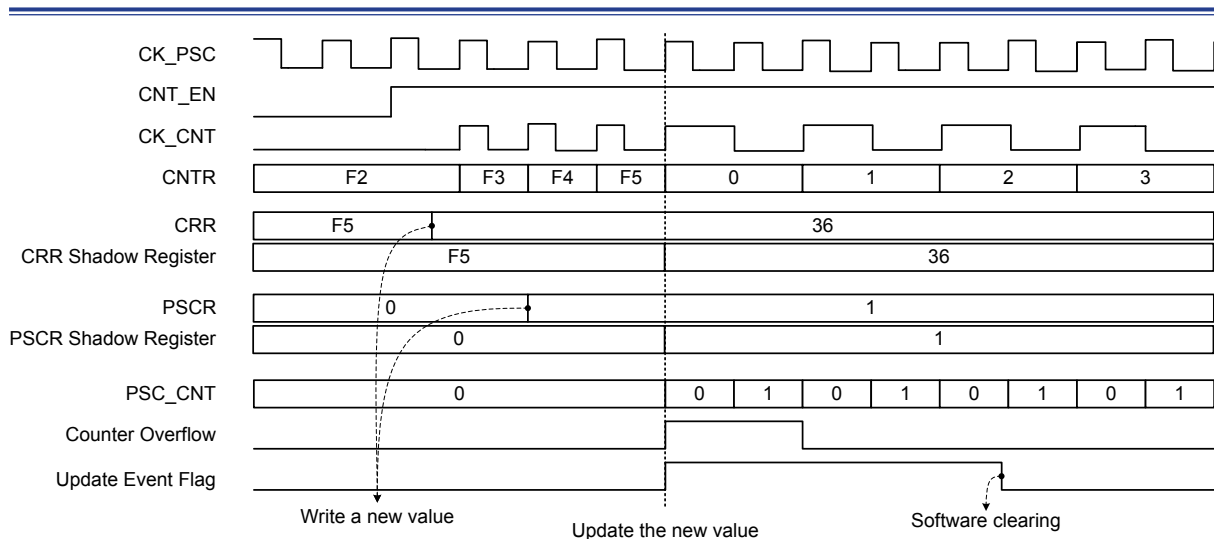


Figure 117. Up-counting Example

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. If the slave mode controller is enabled by setting SMSEL field in the MDCFR register to 0x7, the prescaler is clocked by other clock sources selected by the TRSEL field in the TRCFR register and described as follows. When the slave mode selection bits SMSEL are set to 0x4, 0x5 or 0x6, the internal APB clock f_{CLKIN} is the counter prescaler driving clock source.

- 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.

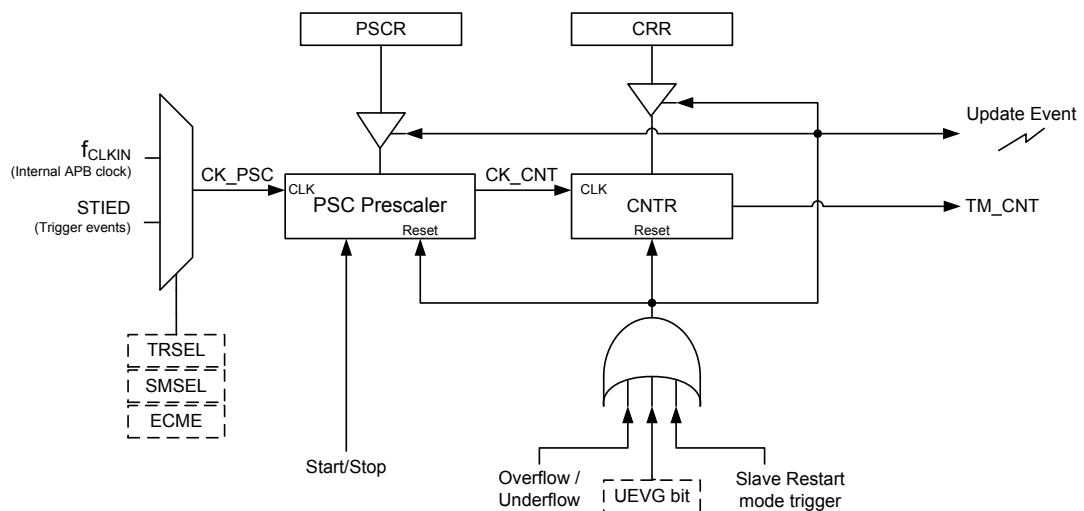
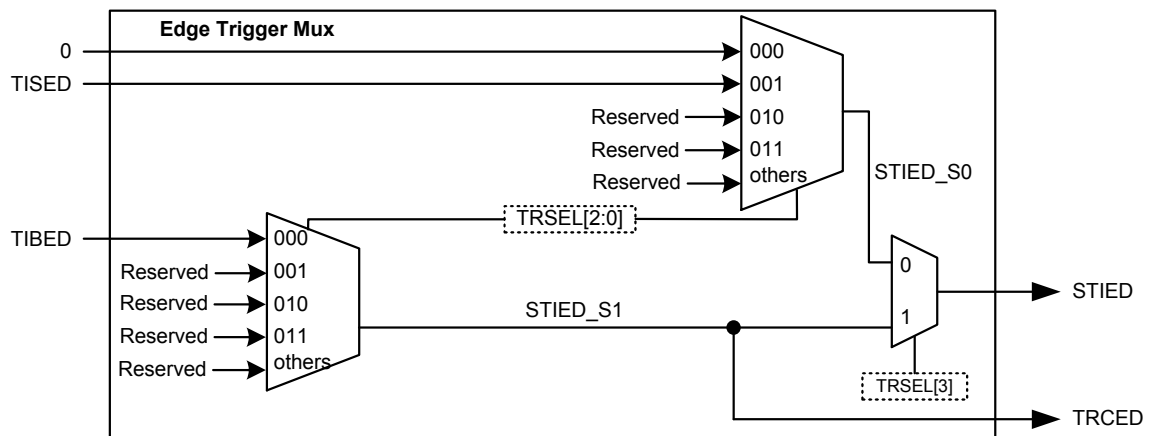


Figure 118. SCTM Clock Selection Source

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 SCTM functions which are triggered by a trigger signal rising edge.

Edge Trigger = Channel input



Level Trigger Source = Channel input + Software UEVG bit

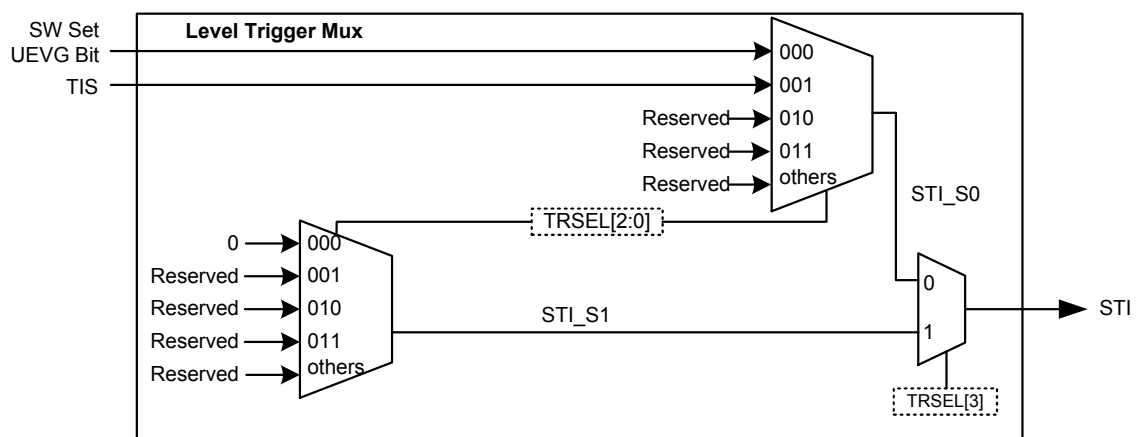


Figure 119. Trigger Controller Block

Slave Controller

The SCTM 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.

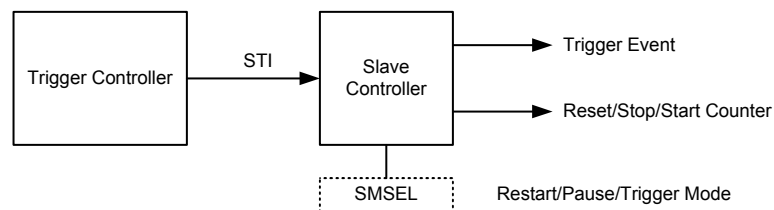


Figure 120. Slave Controller Diagram

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.

Timer Counter Reload Register CRR = 32

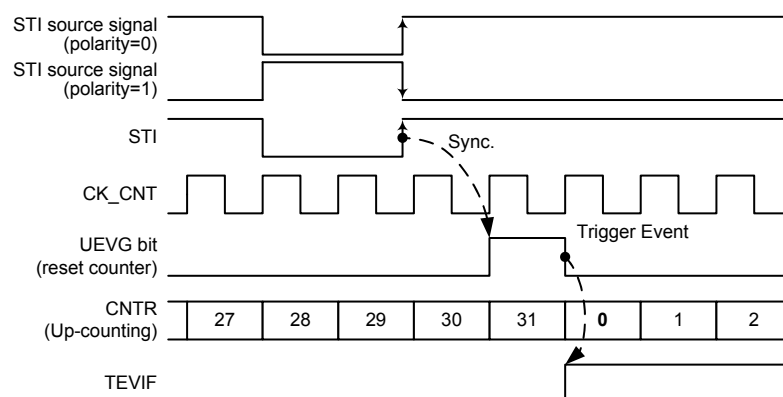


Figure 121. SCTM in Restart Mode

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 TIBED signal.

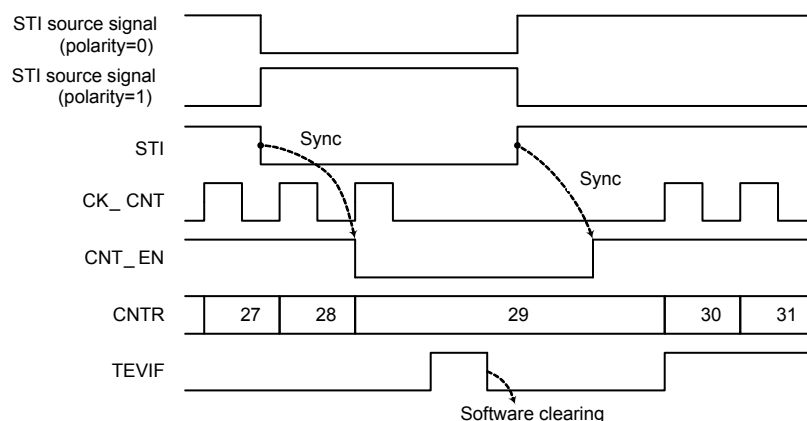


Figure 122. SCTM in Pause Mode

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.

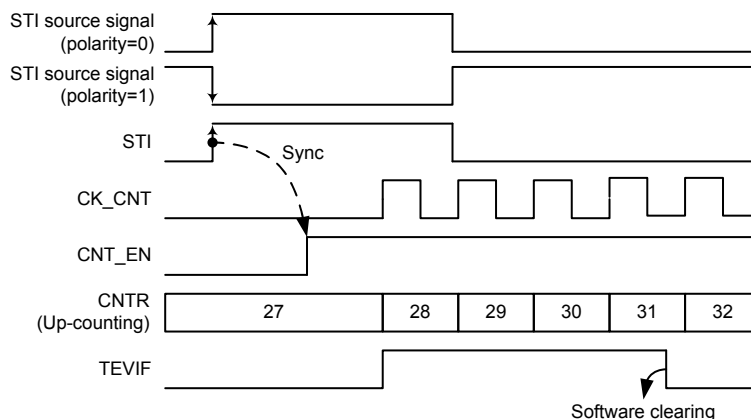


Figure 123. SCTM in Trigger Mode

Channel Controller

The SCTM channel can be used as the capture input or compare match output. The 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 through the read/write preload register.

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

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

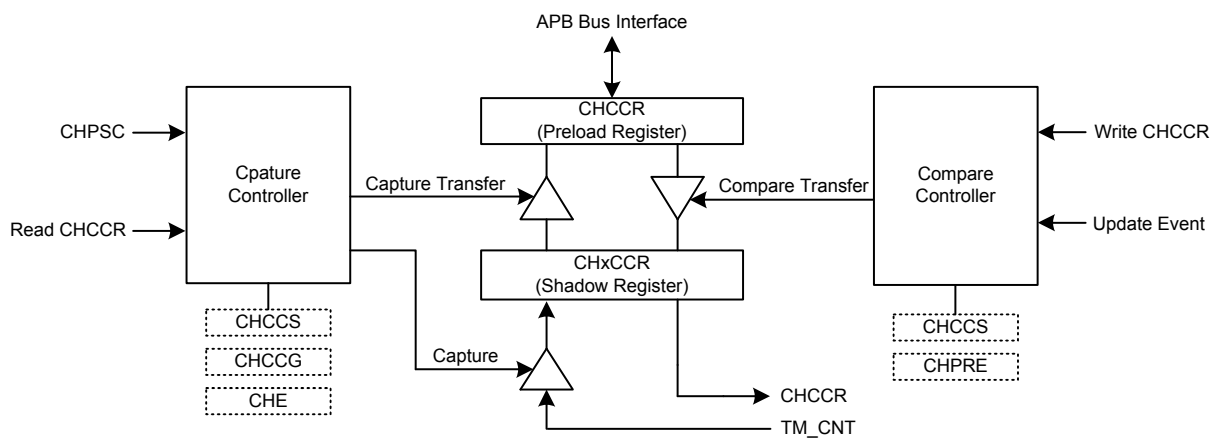


Figure 124. Capture/Compare Block Diagram

Capture Counter Value Transferred to CHCCR

When the channel is used as a capture input, the counter value is captured into the Channel Capture/Compare Register (CHCCR) when an effective input signal transition occurs. Once the capture event occurs, the CHCCIF flag in the INTSR register is set accordingly. If the CHCCIF 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 CHOCF, will be set.

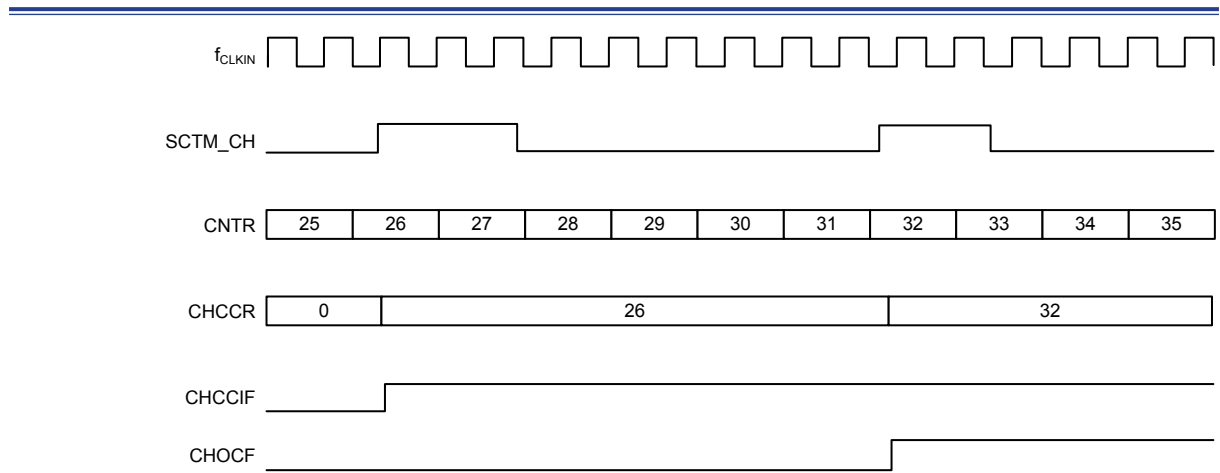


Figure 125. Input Capture Mode

Input Stage

The input stage consists of a digital filter, a channel polarity selection, edge detection and a channel prescaler. The channel input signal (TI) is sampled by a digital filter to generate a filtered input signal TIFP. Then the channel polarity and the edge detection block can generate a TISED signal for the input capture function. The effective input event number can be set by the channel input prescaler register (CHPSC).

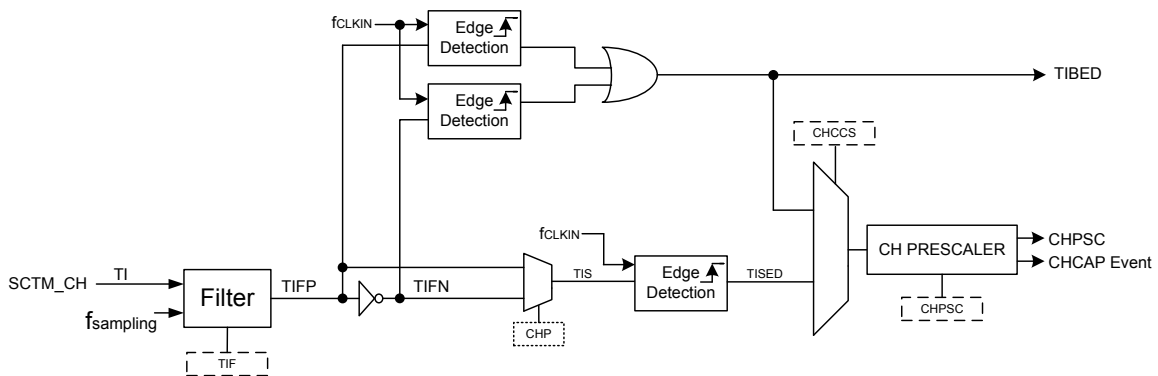


Figure 126. Channel Input Stages

Digital Filter

The digital filters are embedded in the channel input stage. The digital filter in the SCTM is an N-event counter where N refers to how many valid transitions are necessary to output a filtered signal. The N value can be 0, 2, 4, 5, 6 or 8 according to the user selection for this digital filter.

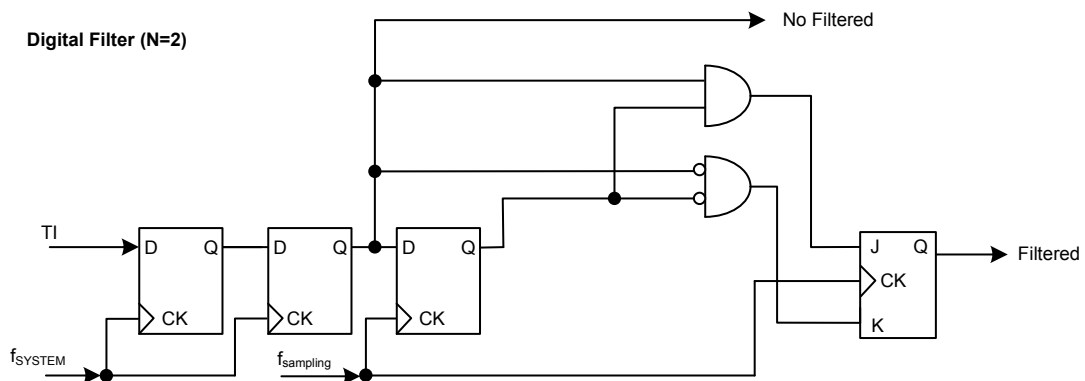


Figure 127. TI Digital Filter Diagram with N = 2

Output Stage

The SCTM output has function for compare match, single pulse or PWM output. The channel output SCTM_CHO is controlled by the CHOM, CHP and CHE bits in the corresponding CHOCFR, CHPOLR and CHCTR registers.

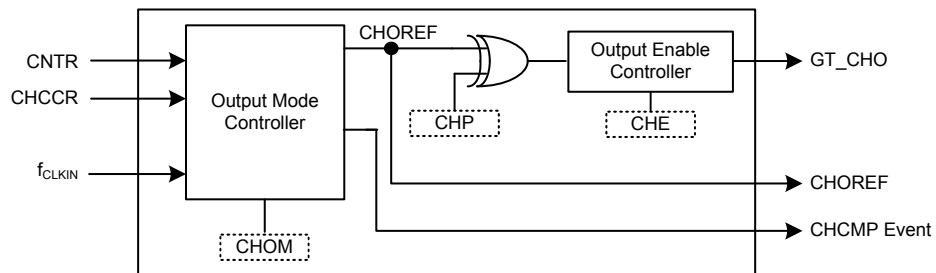


Figure 128. Output Stage Block Diagram

Channel Output Reference Signal

When the SCTM is used in the compare match output mode, the CHOREF signal (Channel Output Reference signal) is defined by the CHOM bit setup. The CHOREF signal has several types of output function which defines what happens to the output when the counter value matches the contents of the CHCCR register. In addition to the low, high and toggle CHOREF output types; there are also PWM mode 1 and PWM mode 2 outputs. In these modes, the CHOREF signal level is changed according to the relationship between the counter value and the CHCCR content. There are also two modes which will force the output into an inactive or active state irrespective of the CHCCR content or counter values. With regard to a more detailed description refer to the relative bit definition. The accompanying Table 40 shows a summary of the output type setup.

Table 40. Compare Match Output Setup

CHOM value	Compare Match Level
0x00	No change
0x01	Clear Output to 0
0x02	Set Output to 1
0x03	Toggle Output
0x04	Force Inactive Level
0x05	Force Active Level
0x06	PWM Mode 1
0x07	PWM Mode 2

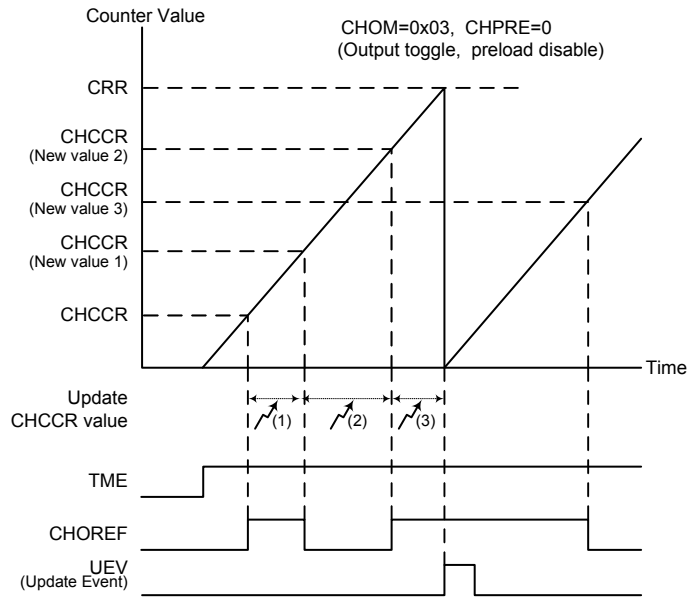


Figure 129. Toggle Mode Channel Output Reference Signal (CHPRE = 0)

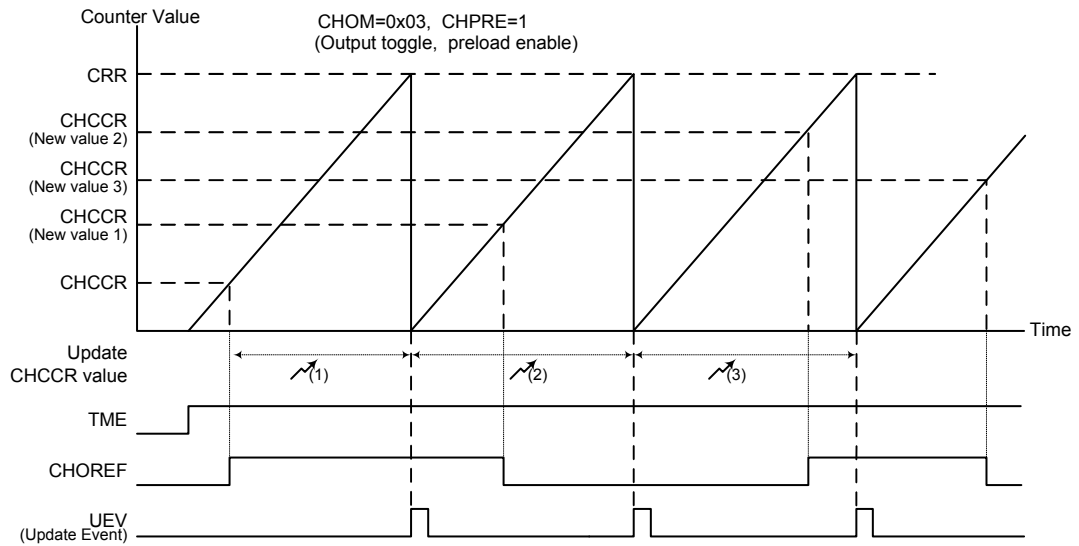


Figure 130. Toggle Mode Channel Output Reference Signal (CHPRE = 1)

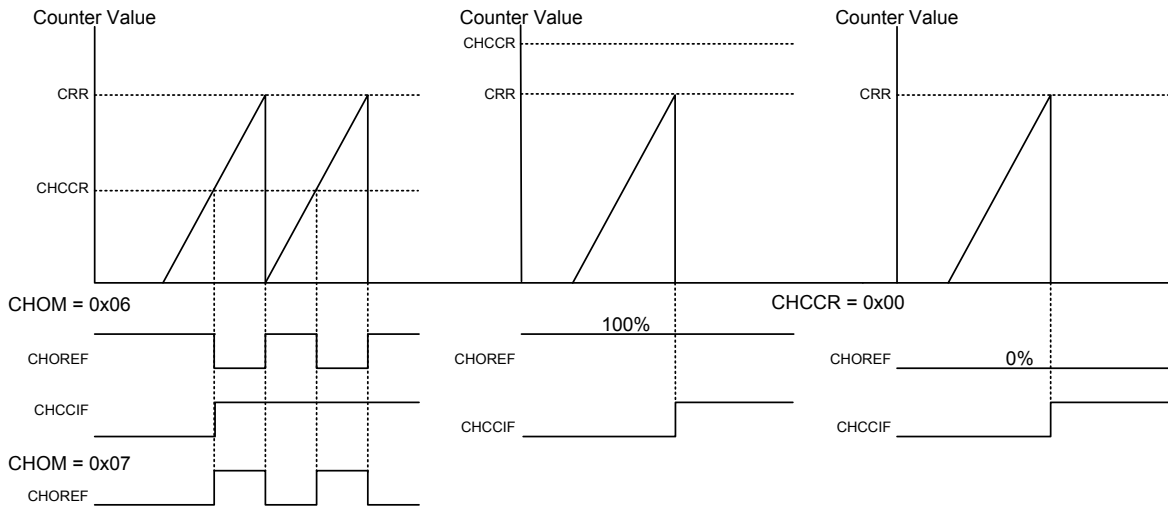


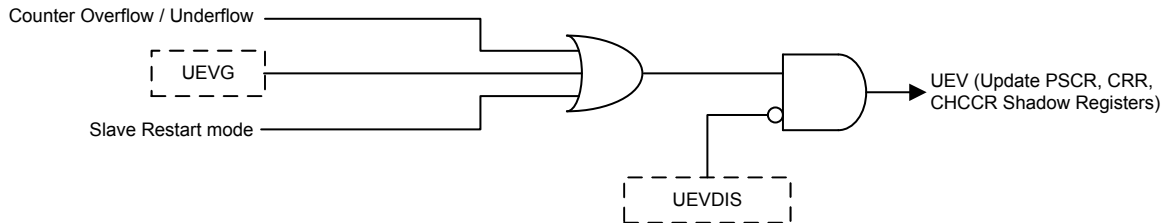
Figure 131. PWM Mode Channel Output Reference Signal

Update Management

The Update event is used to update the CRR, the PSCR and the CHCCR values from the actual registers to the corresponding shadow registers. An update event will occur when the counter overflows, 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

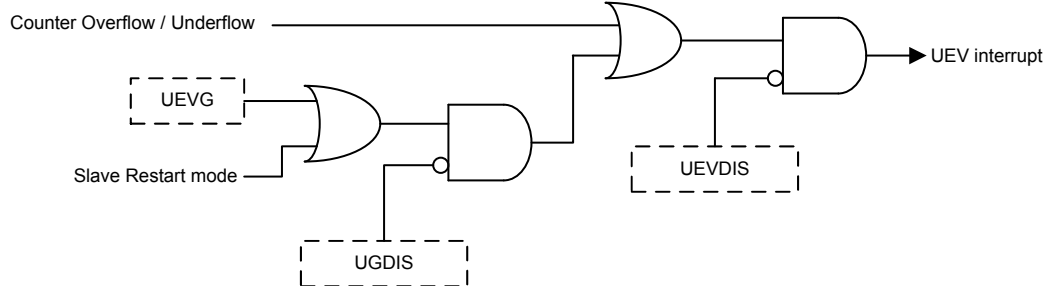


Figure 132. Update Event Setting Diagram

Register Map

The following table shows the SCTM registers and reset values.

Table 41. SCTM 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
CHICFR	0x020	Channel Input Configuration Register	0x0000_0000
CHOCFR	0x040	Channel Output Configuration Register	0x0000_0000
CHCTR	0x050	Channel Control Register	0x0000_0000
CHPOLR	0x054	Channel Polarity Configuration Register	0x0000_0000
DICTR	0x074	Timer 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
CHCCR	0x090	Channel Capture/Compare Register	0x0000_0000

Register Descriptions

Timer Counter Configuration Register – CNTCFR

This register specifies the SCTM counter configuration.

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	0
	7	6	5	4	3	2	1	0		
Type/Reset	Reserved						RW	0	RW	0

Bits	Field	Descriptions
[9:8]	CKDIV	<p>Clock Division</p> <p>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.</p> <p>00: $f_{DTS} = f_{CLKIN}$ 01: $f_{DTS} = f_{CLKIN} / 2$ 10: $f_{DTS} = f_{CLKIN} / 4$ 11: Reserved</p>
[1]	UGDIS	<p>Update event interrupt generation disable control</p> <p>0: Any of the following events will generate an update interrupt</p> <ul style="list-style-type: none"> - Counter overflow - Setting the UEVG bit - Update generation through the slave mode <p>1: Only counter overflow generates an update interrupt</p>
[0]	UEVDIS	<p>Update event Disable control</p> <p>0: Enable the update event request by one of following events:</p> <ul style="list-style-type: none"> - Counter overflow - Setting the UEVG bit - Update generation through the slave mode <p>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)</p>

Timer Mode Configuration Register – MDCFR

This register specifies the SCTM slave 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	Reserved					RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0			
Type/Reset	Reserved										

Bits	Field	Descriptions
------	-------	--------------

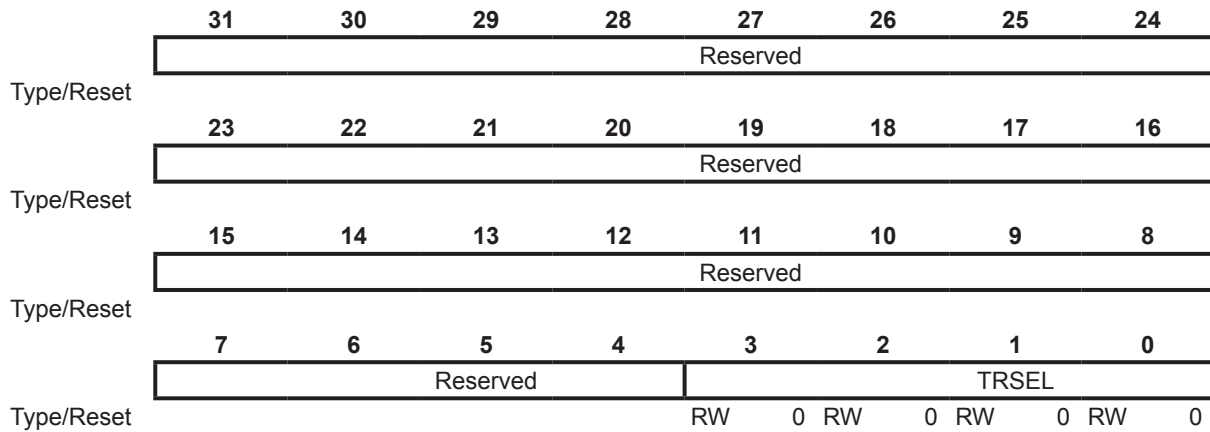
Bits	Field	Descriptions																					
[10:8]	SMSEL	Slave Mode Selection																					
		<table border="1"> <thead> <tr> <th>SMSEL [2:0]</th> <th>Mode</th> <th>Descriptions</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>Disable mode</td> <td>The prescaler is clocked directly by the internal clock.</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> <tr> <td>Others</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	SMSEL [2:0]	Mode	Descriptions	000	Disable mode	The prescaler is clocked directly by the internal clock.	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.	Others	Reserved	
SMSEL [2:0]	Mode	Descriptions																					
000	Disable mode	The prescaler is clocked directly by the internal clock.																					
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.																					
Others	Reserved																						

Timer Trigger Configuration Register – TRCFR

This register specifies the trigger source selection of SCTM.

Offset: 0x008

Reset value: 0x0000_0000



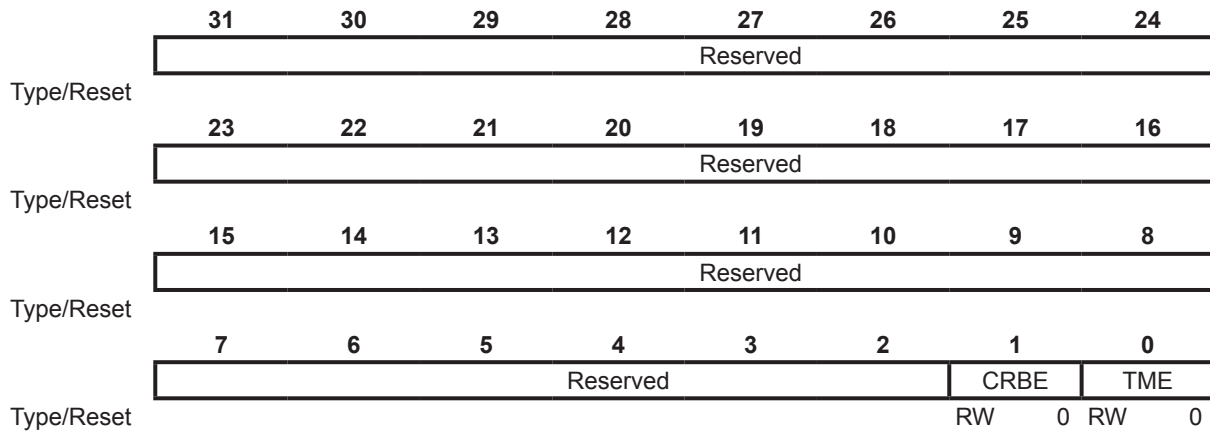
Bits	Field	Descriptions
[3:0]	TRSEL	<p>Trigger Source Selection</p> <p>These bits are used to select the trigger input (STI) for counter synchronizing.</p> <ul style="list-style-type: none"> 0000: Software Trigger by setting the UEVG bit 0001: Filtered input of channel (TIS) 0011: Reserved 1000: Channel both edge detector (TIBED) Others: Default 0 <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 0x00.</p>

Timer Counter Register – CTR

This register specifies the timer enable bit (TME), CRR buffer enable bit (CRBE).

Offset: 0x010

Reset value: 0x0000_0000



Bits	Field	Descriptions
[1]	CRBE	Counter-Reload register Buffer Enable 0: Counter reload register can be updated immediately 1: Counter reload register cannot be updated until the update event occurs
[0]	TME	Timer Enable bit 0: SCTM off 1: SCTM on - SCTM functions normally When the TME bit is cleared to 0, the counter is stopped and the SCTM 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 SCTM registers to function normally.

Channel Input Configuration Register – CHICFR

This register specifies the channel input mode configuration.

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				CHPSC		CHCCS	
					RW	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				TIF			
					RW	0	RW	0

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

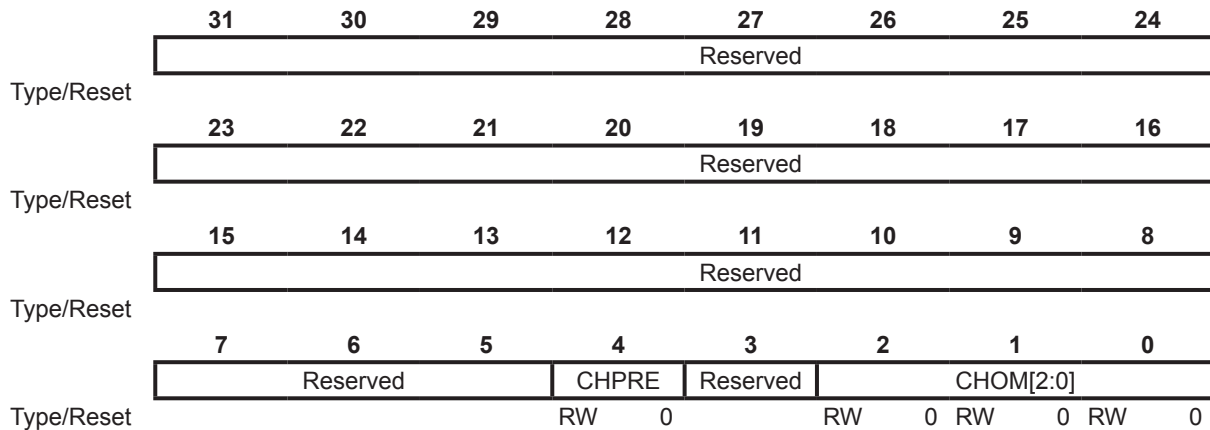
Bits	Field	Descriptions
[3:0]	TIF	<p>Channel Input Source TI Filter Setting</p> <p>These bits define the frequency divided ratio used to sample the TI signal. The Digital filter in the SCTM is an N-event counter where N is defined as how many valid transitions are necessary to output a filtered signal.</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 Output Configuration Register – CHOCFR

This register specifies the channel output mode configuration.

Offset: 0x040

Reset value: 0x0000_0000



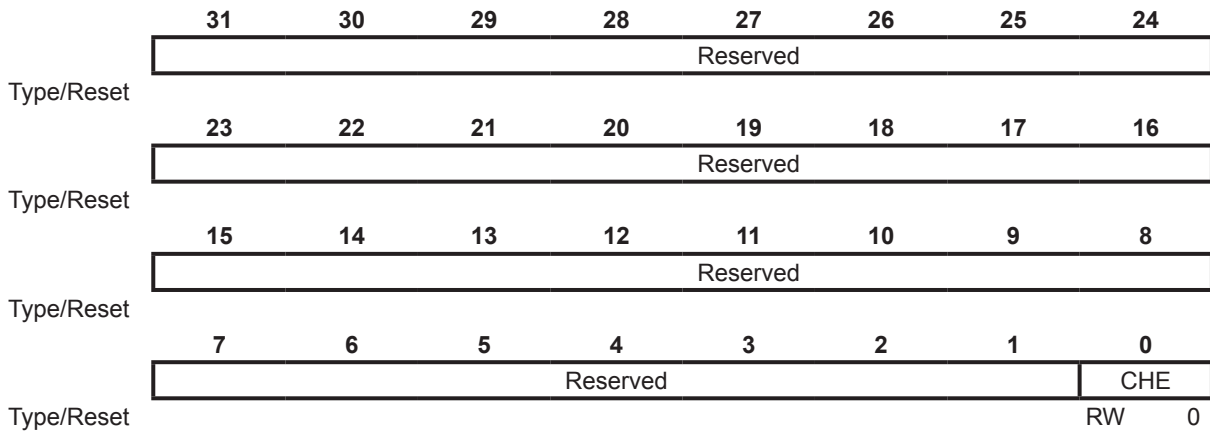
Bits	Field	Descriptions
[4]	CHPRE	<p>Channel Capture/Compare Register (CHCCR) Preload Enable</p> <p>0: CHCCR preload function is disabled. The CHCCR register can be immediately assigned a new value when the CHPRE bit is cleared to 0 and the updated CHCCR value is used immediately.</p> <p>1: CHCCR preload function is enabled. The new CHCCR value will not be transferred to its shadow register until the update event occurs.</p>
[2:0]	CHOM[2:0]	<p>Channel Output Mode Setting</p> <p>These bits define the functional types of the output reference signal CHOREF.</p> <p>000: No Change 001: Output 0 on compare match 010: Output 1 on compare match 011: Output toggles on compare match 100: Force inactive – CHOREF is forced to 0 101: Force active – CHOREF is forced to 1 110: PWM mode 1 - During up-counting, channel has an active level when CNTR < CHCCR or otherwise has an inactive level. 111: PWM mode 2 - During up-counting, channel is has an inactive level when CNTR < CHCCR or otherwise has an active level.</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



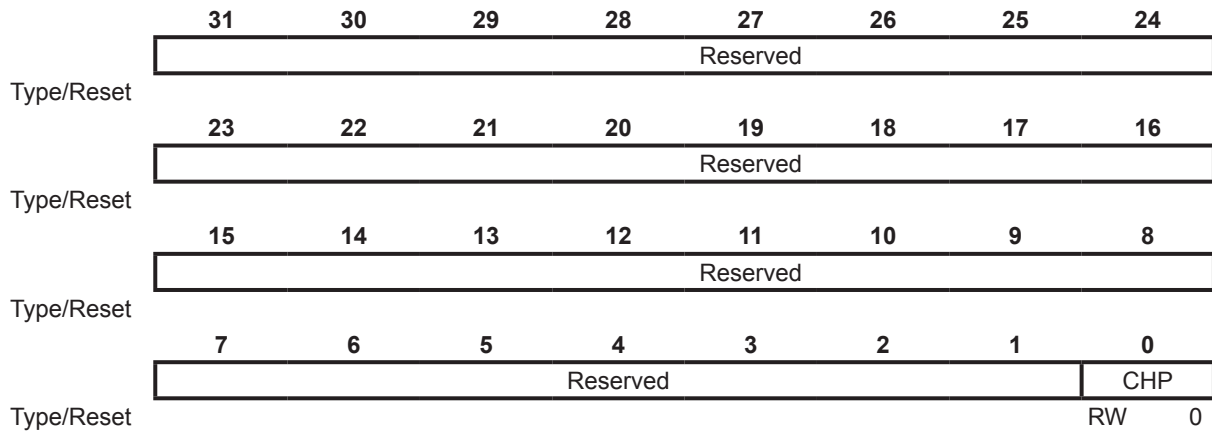
Bits	Field	Descriptions
[0]	CHE	<p>Channel Capture/Compare Enable</p> <ul style="list-style-type: none"> - Channel is configured as an input (CHCCS = 0x01/0x03) <ul style="list-style-type: none"> 0: Input Capture Mode is disabled 1: Input Capture Mode is enabled - Channel is configured as an output (CHCCS = 0x00) <ul style="list-style-type: none"> 0: Off –Channel output signal CHO is not active 1: On –Channel output signal CHO 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



Bits	Field	Descriptions
[0]	CHP	Channel Capture/Compare Polarity - When Channel is configured as an input 0: capture event occurs on a Channel rising edge 1: capture event occurs on a Channel falling edge - When Channel is configured as an output (CHCCS = 0x00) 0: Channel Output is active high 1: Channel Output is active low

Timer Interrupt Control Register – DICTR

This register contains the timer interrupt enable control bits.

Offset: 0x074

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					TEVIE	Reserved	UEVIE	
	7	6	5	4	3	2	1	0	
Type/Reset	Reserved							CHCCIE	
									RW 0
									RW 0

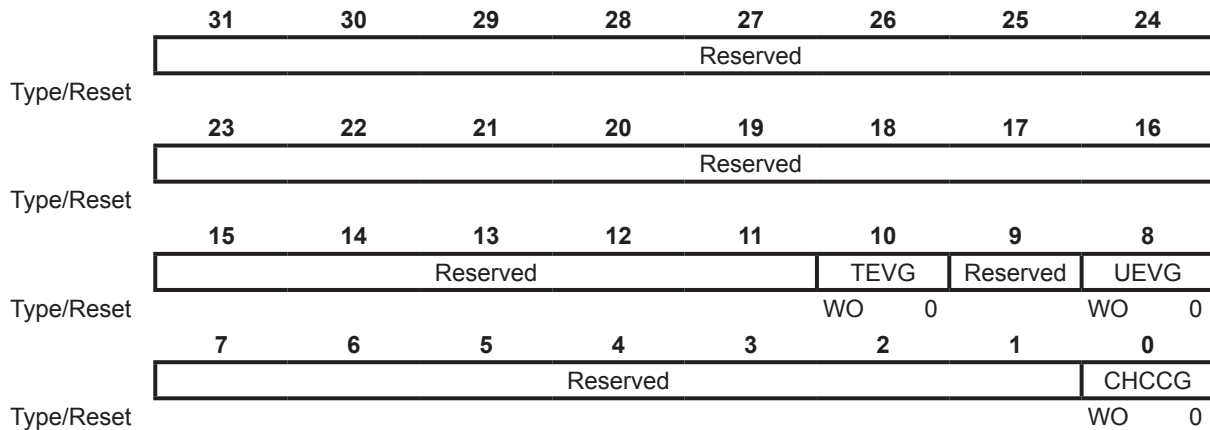
Bits	Field	Descriptions
[10]	TEVIE	Trigger event Interrupt Enable 0: Trigger event interrupt is disabled 1: Trigger event interrupt is enabled
[8]	UEVIE	Update event Interrupt Enable 0: Update event interrupt is disabled 1: Update event interrupt is enabled
[0]	CHCCIE	Channel Capture/Compare Interrupt Enable 0: Channel interrupt is disabled 1: Channel interrupt is enabled

Timer Event Generator Register – EVGR

This register contains the software event generation bits.

Offset: 0x078

Reset value: 0x0000_0000



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 detail descriptions, refer to the corresponding section.</p>
[0]	CHCCG	<p>Channel Capture/Compare Generation</p> <p>A Channel 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.</p> <p>If Channel is configured as an input, the counter value is captured into the CHCCR register and then the CHCCIF bit is set. If Channel is configured as an output, the CHCCIF bit is set.</p>

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
					WOC 0		WOC 0
Type/Reset							
7	6	5	4	3	2	1	0
Reserved			CHOCF	Reserved			CHCCIF
			WOC 0				WOC 0
Type/Reset							

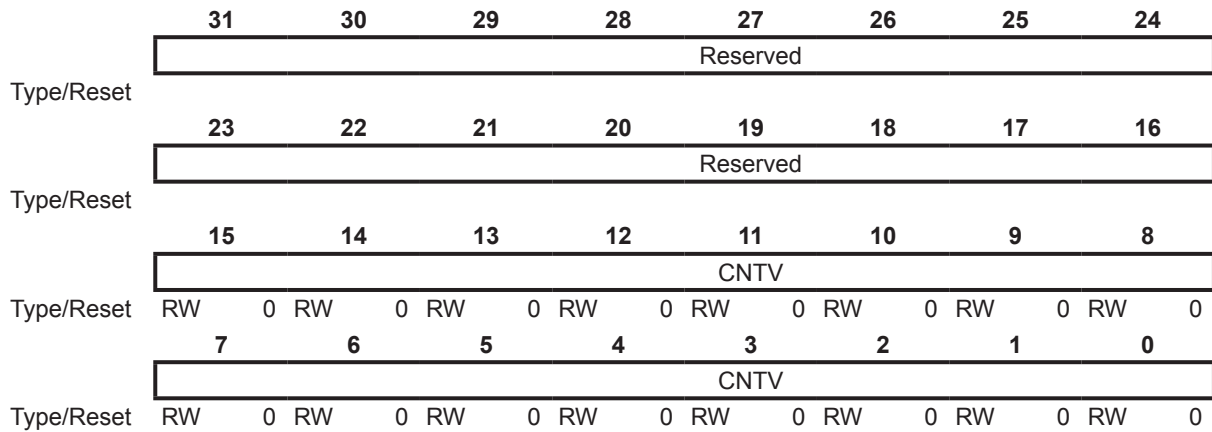
Bits	Field	Descriptions
[10]	TEVIF	Trigger Event Interrupt Flag This flag is set by hardware on a trigger event and is cleared by software. 0: No trigger event occurs 1: Trigger event occurs
[8]	UEVIF	Update Event Interrupt Flag. This bit is set by hardware on an update event and is cleared by software. 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
[4]	CHOCF	Channel 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 CHCCIF bit is already set and it is not yet cleared by software.
[0]	CHCCIF	Channel Capture/Compare Interrupt Flag - Channel is configured as an output: 0: No match event occurs 1: The contents of the counter CNTR have matched the content of the CHCCR register This flag is set by hardware when the counter value matches the CHCCR value. It is cleared by software. - Channel 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 CHCCR register.

Timer Counter Register – CNTR

This register stores the timer counter value.

Offset: 0x080

Reset value: 0x0000_0000



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

Single-Channel Timer (SCTM)

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	Reserved							
Type/Reset																
	23	22	21	20	19	18	17	16	Reserved							
Type/Reset																
	15	14	13	12	11	10	9	8	PSCV							
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	PSCV							
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

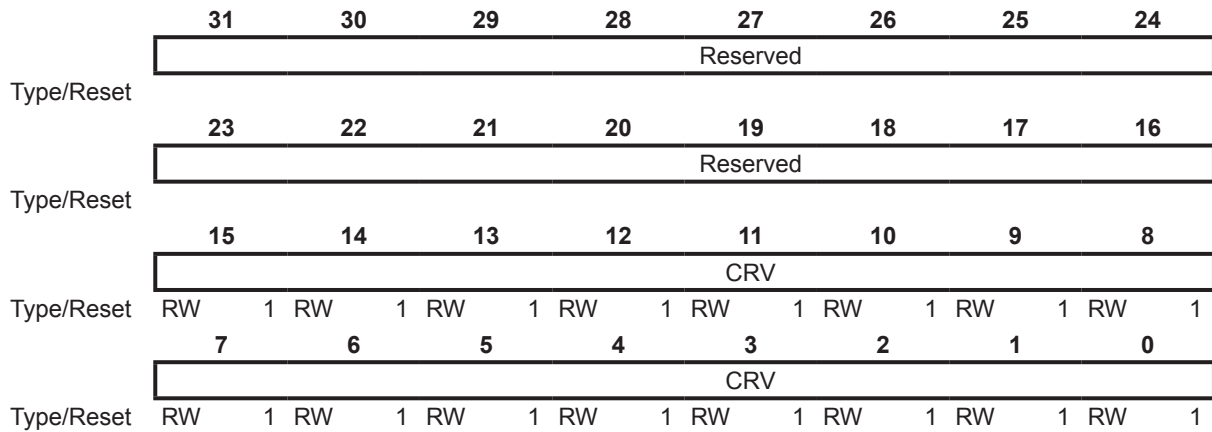
Bits	Field	Descriptions
[15:0]	PSCV	<p>Prescaler Value</p> <p>These bits are used to specify the prescaler value to generate the counter clock frequency f_{CK_CNT}.</p> $f_{CK_CNT} = \frac{f_{CK_PSC}}{PSCV[15:0] + 1}$ <p>where the f_{CK_PSC} is the prescaler clock source.</p>

Timer Counter Reload Register – CRR

This register specifies the timer counter reload value.

Offset: 0x088

Reset value: 0x0000_FFFF



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

Single-Channel Timer (SCTM)

Channel Capture/Compare Register – CHCCR

This register specifies the timer channel capture/compare value.

Offset: 0x090

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	CHCCV							
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	CHCCV							
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[15:0]	CHCCV	Channel Capture/Compare Value - When Channel is configured as an output The CHCCR value is compared with the counter value and the comparison result is used to trigger the CHOREF output signal. - When Channel is configured as an input The CHCCR register stores the counter value captured by the last channel capture event.

18 Real Time Clock (RTC)

Introduction

The Real Time Clock, RTC, circuitry includes the APB interface, a 32-bit up-counter, a control register, a prescaler, a compare register and a status register. Most of the RTC circuits are located in the Backup Domain, as shown shaded 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.

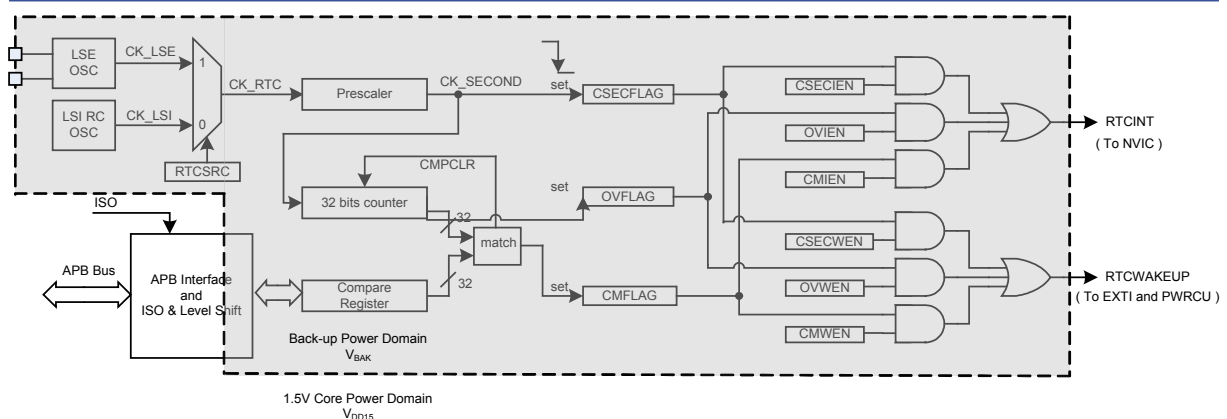


Figure 133. RTC Block Diagram

Features

- 32-bit up counter for counting elapsed time
- Programmable clock prescaler
 - Division factor: 1, 2, 4, 8, ..., 32768
- 32-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 Backup Domain power on reset, PORB, or by a Backup Domain software reset by setting the BAKRST bit in the BAKCR 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_{BAK} 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 circuits located in the VBAK domain using level shift circuitry and isolated by the ISO signals when the V_{DD15} supply voltage is powered off. The isolation function must be disabled by setting the BAKISO bit to 1 in the LPCR register as described in the Clock Control Unit before accessing the RTC registers using the APB bus.

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 0x0F 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 42. 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 32-bit up-counter which increments 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 32-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 0x0F, 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.

RTCOUT Output Pin Configuration

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

Table 43. 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)	----- $\overline{T_R}$ -----
		ROLF	-----
	1 Second clock	RTCCMP	----- X -----
		RTCCNT	----- 3 ----- 4 ----- 5 -----
		RTCOUT (ROAP = 0)	T_R T_R T_R
		RTCOUT (ROAP = 1)	$\overline{T_R}$ $\overline{T_R}$ $\overline{T_R}$
		ROLF	-----
1 (Level mode)	0 Compare match	RTCCMP	----- 4 -----
		RTCCNT	----- 3 ----- 4 ----- 5 -----
		RTCOUT (ROAP = 0)	----- $\overline{1}$ -----
		RTCOUT (ROAP = 1)	----- 1 -----
		ROLF	----- $\overline{1}$ ----- \rightarrow -----
	1 Second clock	RTCCMP	----- X -----
		RTCCNT	----- 3 ----- 4 ----- 5 -----
		RTCOUT (ROAP = 0)	----- 1 ----- $\overline{1}$ ----- 1 -----
		RTCOUT (ROAP = 1)	----- $\overline{1}$ ----- 1 ----- $\overline{1}$ -----
		ROLF	----- $\overline{1}$ ----- \rightarrow ----- $\overline{1}$ ----- \rightarrow -----

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

Real Time Clock (RTC)

Register Map

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

Table 44. 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

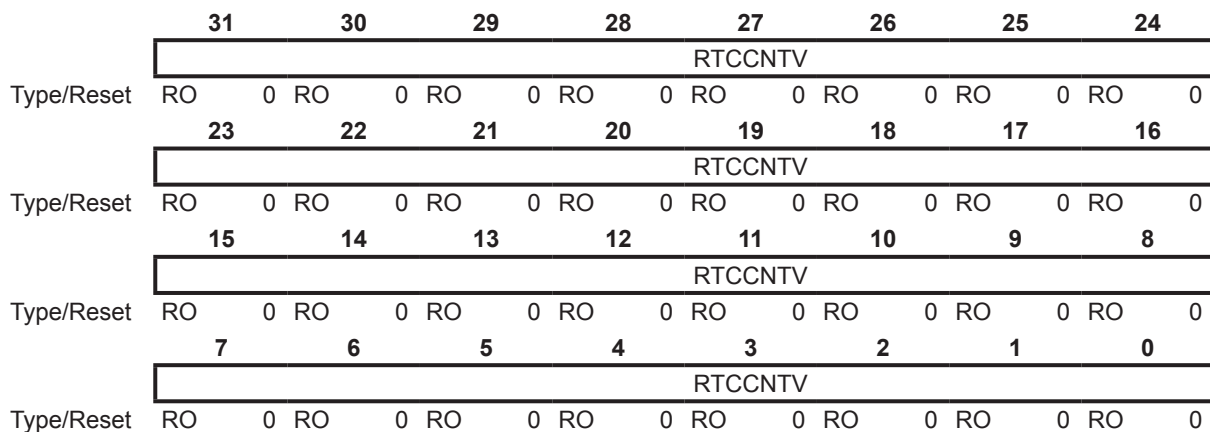
Register Descriptions

RTC Counter Register – RTCCNT

This register defines a 32-bit up counter which is incremented by the CK_SECOND clock.

Offset: 0x000

Reset value: 0x0000_0000



Bits	Field	Descriptions
[31:0]	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. This register is reset by one of the following conditions:</p> <ul style="list-style-type: none"> - Backup Domain software reset - set the BAKRST bit in the BAKCR register - Backup Domain power on reset - PORB - 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.

Offset: 0x004

Reset value: 0x0000_0000 (Reset by Backup Domain reset only)

	31	30	29	28	27	26	25	24	RTCCMPV							
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0
	23	22	21	20	19	18	17	16	RTCCMPV							
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	RTCCMPV							
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	RTCCMPV							
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[31:0]	RTCCMPV	<p>RTC Compare Match Value</p> <p>A match condition happens when the value in the RTCCNT register is equal to 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.

Offset: 0x008

Reset value: 0x0000_0F04 (Reset by Backup Domain reset only)

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

Bits	Field	Descriptions
[20]	ROLF	<p>RTCOUT Level Mode Flag</p> <p>0: RTCOUT Output is inactive</p> <p>1: RTCOUT Output is holding as active level</p> <p>Set by hardware when level mode (ROWM = 1) and a RTCOUT output event occurred. Cleared by software reading this flag. The RTCOUT signal will return to the inactive level after software has read this bit.</p>
[19]	ROAP	<p>RTCOUT Output is active Polarity</p> <p>0: Active level is high</p> <p>1: Active level is low</p>
[18]	ROWM	<p>RTCOUT 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 RTCOUT signal will remain at an active level until the ROLF bit is cleared by software reading the ROLF bit.</p>
[17]	ROES	<p>RTCOUT 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 RTCOUT signal is output on the RTCOUT pin when a RTC compare match event or the RTC second clock (CK_SECOND) event occurs.</p>
[16]	ROEN	<p>RTCOUT Output Pin Enable</p> <p>0: Disable RTCOUT output pin</p> <p>1: Enable RTCOUT output pin</p> <p>When the ROEN bit is set to 1, the RTCOUT signal will be at an active level once a RTC compare match on the RTC second clock (CK_SECOOD) 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 RTCOUT pin will be in a floating state.</p>

Bits	Field	Descriptions
[11:8]	RPRE	<p>RTC Clock Prescaler Select</p> $CK_SECOND = CK_RTC / 2^{RPRE}$ <p>0000: $CK_SECOND = CK_RTC / 2^0$ 0001: $CK_SECOND = CK_RTC / 2^1$ 0010: $CK_SECOND = CK_RTC / 2^2$... 1111: $CK_SECOND = CK_RTC / 2^{15}$</p>
[5]	LSESM	<p>LSE oscillator Startup Mode</p> <p>0: Normal startup and requires less operating power 1: Fast startup but requires higher operating current</p>
[4]	CMPCLR	<p>Compare Match Counter Clear</p> <p>0: 32-bit RTC counter is not affected when compare match condition occurs 1: 32-bit RTC counter is cleared when compare match condition occurs</p>
[3]	LSEEN	<p>LSE oscillator Enable Control</p> <p>0: LSE oscillator disabled 1: LSE oscillator enabled</p>
[2]	LSIEN	<p>LSI oscillator Enable Control</p> <p>0: LSI oscillator disabled 1: LSI oscillator enabled</p> <p>The LSIEN bit default value is 1 which means the LSI oscillator is enabled automatically after the Backup Domain powered up.</p> <p>Note: After the backup 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 backup 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.</p>
[1]	RTC SRC	<p>RTC Clock Source Selection</p> <p>0: LSI oscillator selected as the RTC clock source 1: LSE oscillator selected as the RTC clock source</p>
[0]	RTCEN	<p>RTC Enable Control</p> <p>0: RTC is disabled 1: RTC is enabled</p>

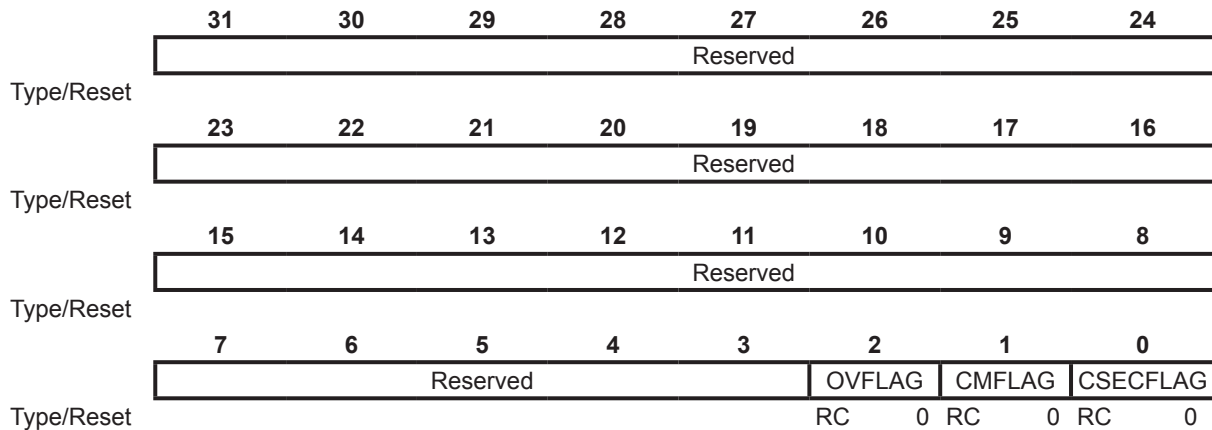
Real Time Clock (RTC)

RTC Status Register – RTCSR

This register stores the counter flags.

Offset: 0x00C

Reset value: 0x0000_0000 (Reset by Backup Domain reset and RTCEN bit change from 1 to 0)



Bits	Field	Descriptions
[2]	OVFLAG	Counter Overflow Flag 0: Counter overflow 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 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 does 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 does 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.

Offset: 0x010

Reset value: 0x0000_0000 (Reset by Backup 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					RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0			
Type/Reset	Reserved					RW	0	RW	0	RW	0

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

19 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.

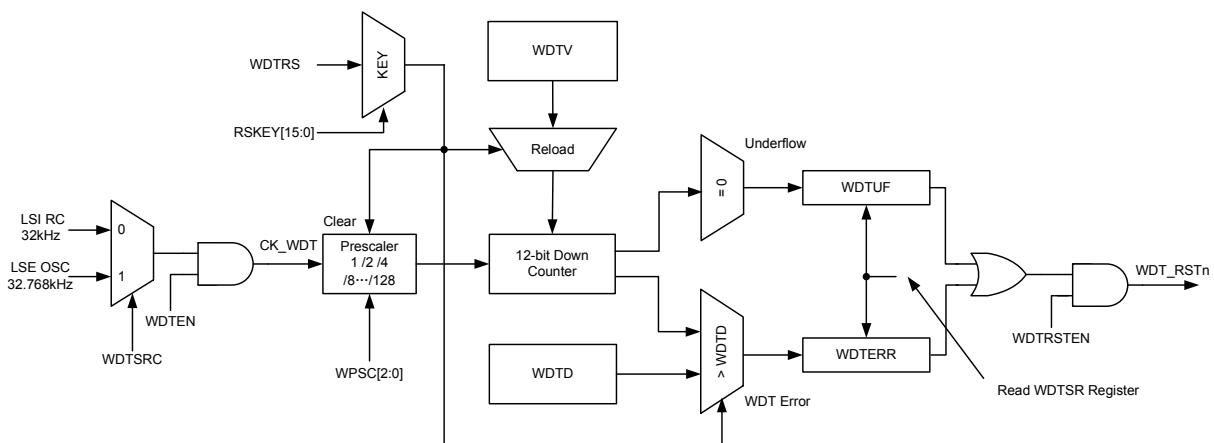


Figure 134. Watchdog Timer Block Diagram

Features

- Clock source from either internal 32 kHz RC oscillator (LSI) or 32,768 Hz oscillator (LSE)
- Can be independently setup to keep running or to stop when entering the sleep or deep sleep mode 1
- 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

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 a programmable counter reload value, reset enable, window value and prescaler value. These configurations are setup 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 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 generate 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 underflows or when a Watchdog timer error occurs. A system reset or write-one operation on the WDTSR register clears 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 or Deep sleep mode 1, the Watchdog timer counter will either continue to count or stop depending on the WDTSHLT bits in the WDTMR0 register. The Watchdog stops counting when the WDTSHLT bits are set in the Sleep mode. The count value is retained so that it continues counting after the system is woken up from the Sleep mode. 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 (in the MCUIDBGCR 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).

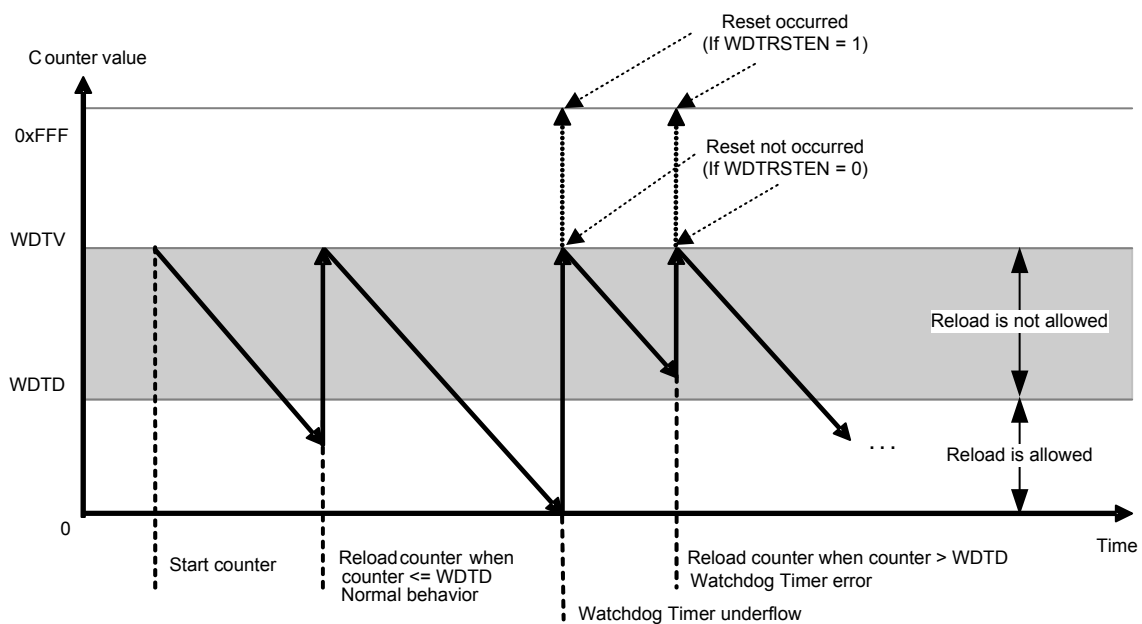


Figure 135. Watchdog Timer Behavior

Register Map

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

Table 45. 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	Reserved											
Type/Reset																				
	23	22	21	20	19	18	17	16	Reserved								WDTEN			
Type/Reset																	RW	0		
	15	14	13	12	11	10	9	8	WDTSHLT		WDTRSTEN		Reserved		WDTV					
Type/Reset	RW	0	RW	0	RW	0	RW		1	RW	1	RW	1	RW	1	RW	1			
	7	6	5	4	3	2	1	0	WDTV											
Type/Reset	RW	1	RW	1	RW	1	RW	1	RW	1	RW	1	RW	1	RW	1				

Bits	Field	Descriptions
[16]	WDTEN	<p>Watchdog Timer Running Enable</p> <p>0: Watchdog timer is disabled</p> <p>1: Watchdog timer is enabled to run.</p> <p>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.</p>
[15:14]	WDTSHLT	<p>Watchdog Timer Sleep Halt</p> <p>00: The Watchdog runs when the system is in the Sleep mode or Deep Sleep mode 1</p> <p>01: The Watchdog runs when the system is in the Sleep mode and halts in Deep Sleep mode 1</p> <p>10 or 11: The Watchdog halts when the system is in the Sleep mode and Deep Sleep mode 1</p> <p>Note that the Watchdog timer always halts when the system is in Deep Sleep mode 2. If a Watchdog interrupt occurs in Sleep or Deep Sleep mode 1, it will wake up the device. The Watchdog stops counting when the WDTSHLT bits are set in the Sleep mode. The count value is retained so that it continues counting after the system wakes up from the Sleep mode.</p>
[13]	WDTRSTEN	<p>Watchdog Timer Reset Enable</p> <p>0: A Watchdog Timer underflow or error has no effect on the reset of system.</p> <p>1: A Watchdog Timer underflow or error triggers a Watchdog timer system reset.</p>
[11:0]	WDTV	<p>Watchdog Timer Counter Value</p> <p>WDTV defines the value loaded into the 12-bit Watchdog down counter.</p>

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	RW	1
	7	6	5	4	3	2	1	0			
Type/Reset	WDTD										
	RW	1	RW	1	RW	1	RW	1	RW	1	

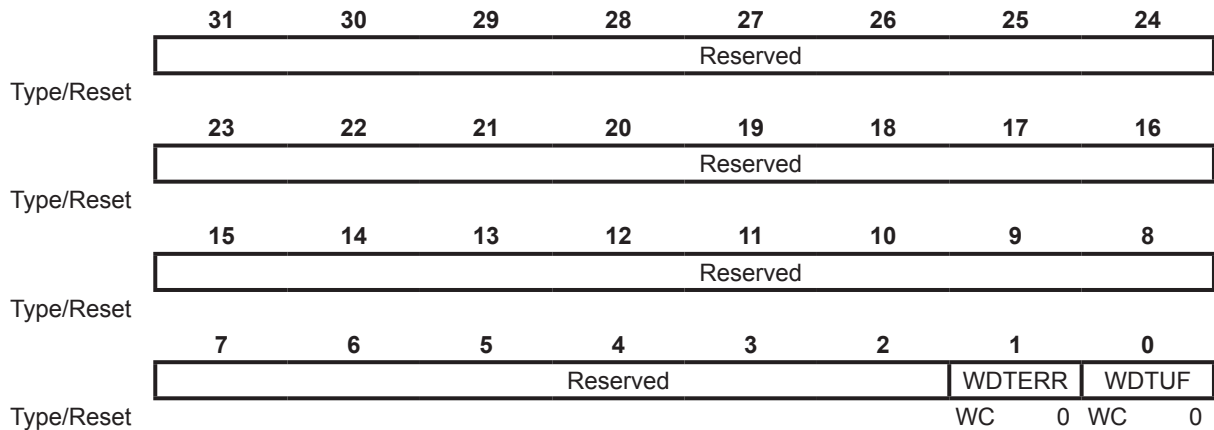
Bits	Field	Descriptions
[14:12]	WPSC	Watchdog Timer Prescaler Selection 000: 1/1 001: 1/2 010: 1/4 011: 1/8 100: 1/16 101: 1/32 110: 1/64 111: 1/128
[11:0]	WDTD	Watchdog Timer Delta Value Define the permitted range to reload the Watchdog timer. If the Watchdog timer counter value is less than or equal 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



Bits	Field	Descriptions
[1]	WDTERR	<p>Watchdog Timer Error</p> <p>0: No Watchdog timer error has occurred since the last read of this register</p> <p>1: A Watchdog timer error has occurred since the last read of this register</p> <p>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.</p>
[0]	WDTUF	<p>Watchdog timer Underflow</p> <p>0: No Watchdog timer underflow since the last read of this register</p> <p>1: A Watchdog timer underflow has occurred since the last read of this register</p> <p>Note that this bit is a write-one-clear flag.</p>

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	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0								
Type/Reset	PROTECT															
	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0		

Watchdog Timer (WDT)

Bits	Field	Descriptions
[15:0]	PROTECT	Watchdog Timer Register Protection For write operation: 0x35CA: Disable the Watchdog timer register write protection Others: Enable the Watchdog timer register write protection For read operation: 0x0000: Watchdog timer register write protection is disabled 0x0001: Watchdog timer register write protection is enabled 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.

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			WDTSEL	
				RW	0			RW	0

Bits	Field	Descriptions
[4]	WDTLOCK	<p>Watchdog Timer Lock Mode</p> <p>0: This bit is only set to 0 on any reset. It cannot be cleared by software.</p> <p>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 cannot be modified or disabled, including the Watchdog timer clock source, and only waits for a system reset to disable the lock mode.</p>
[0]	WDTSRC	<p>Watchdog Timer Clock Source Selection</p> <p>0: Internal 32 kHz RC oscillator clock selected (LSI)</p> <p>1: External 32.768 kHz crystal oscillator clock selected (LSE)</p> <p>Select using software to control the Watchdog timer clock source.</p>

20 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 register is used to setup different kinds of duty cycle implementation for the SCL pulse.

The SDA line which is connected to the whole I²C bus is a bi-directional 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.

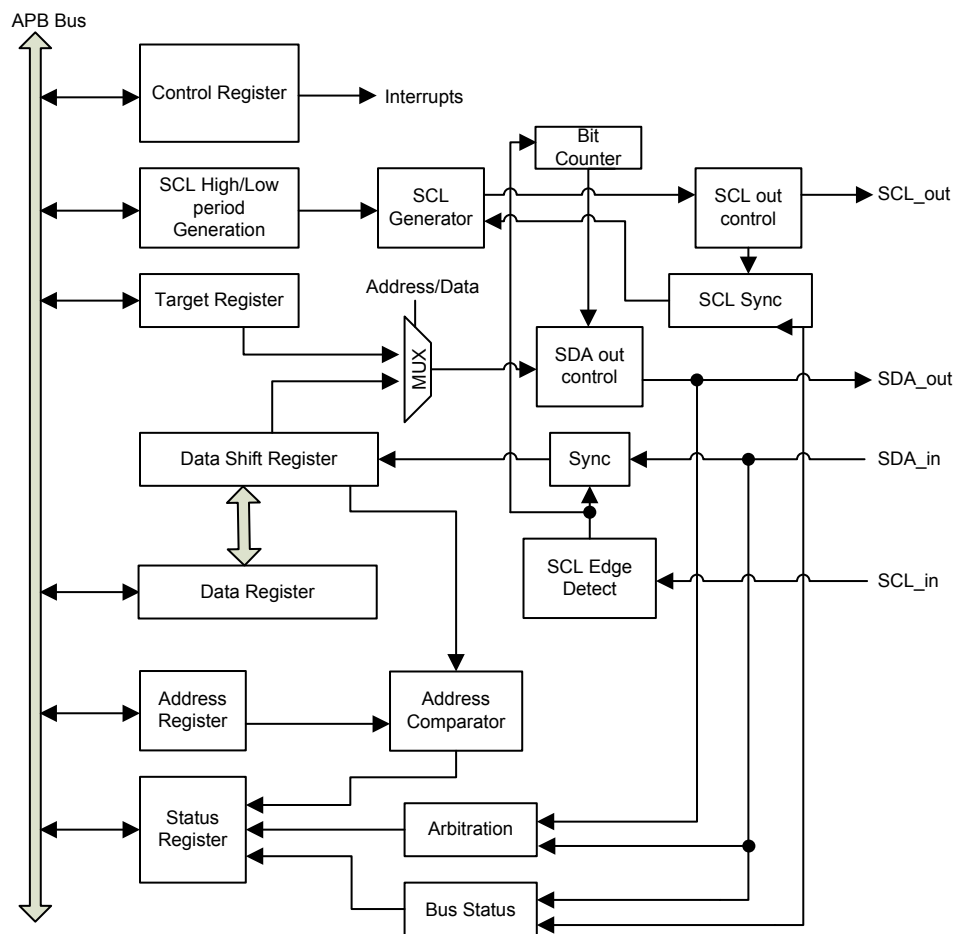


Figure 136. I²C Module Block Diagram

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
- Bi-directional data transfer between master and slave
- Multi-master bus – no central master
 - The same interface can act as Master or Slave
- Arbitration among simultaneously transmitting masters without corrupting of 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 mode and general call addressing.
- Multiple slave addresses using address mask function
- Time-out 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, 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.

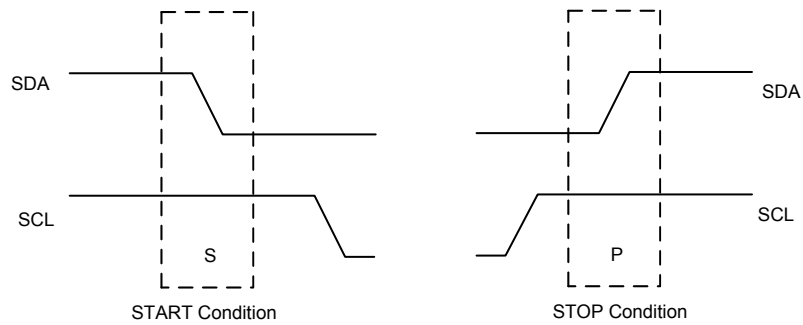


Figure 137. START and STOP Condition

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.

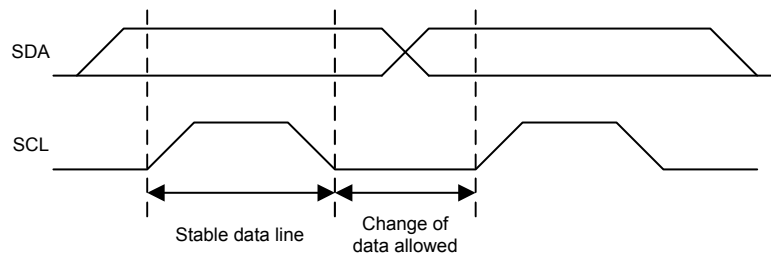


Figure 138. Data Validity

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 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-bits Address Format

The 7-bit address format is composed of the seven-bit length slave address, which the master device wants to communicate with a R/ \bar{W} bit and an ACK bit. The R/ \bar{W} bit defines the direction of the data transfer.

R/ \bar{W} =0 (Write): The master transmits data to the addressed slave.

R/ \bar{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 master.

Note that it is forbidden to own the same address for two slave devices.

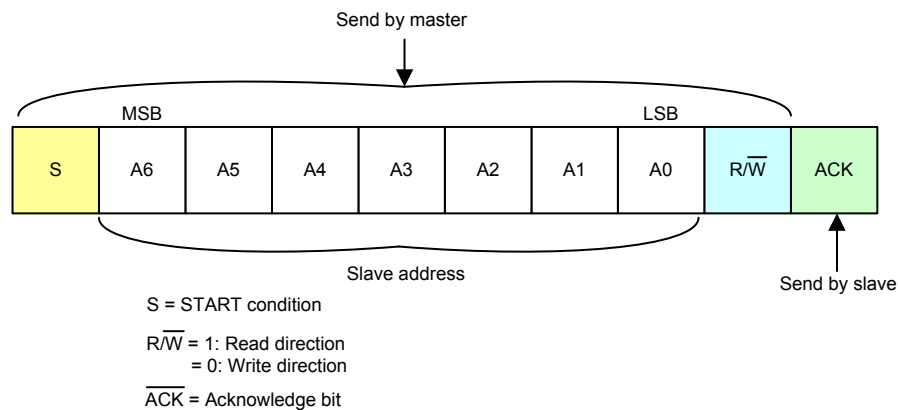


Figure 139. 7-bit Addressing Mode

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”, 10th and 9th bits of the slave address. The second byte is the remaining 8 bit address of the slave device.

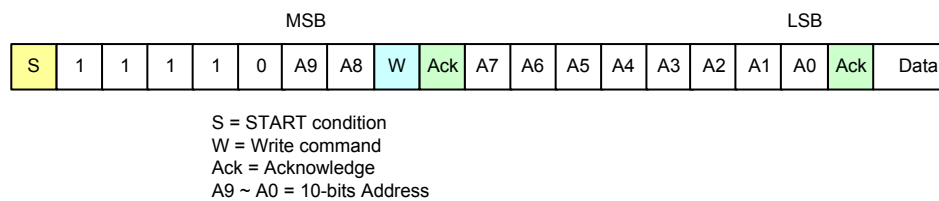


Figure 140. 10-bit Addressing Write Transmit Mode

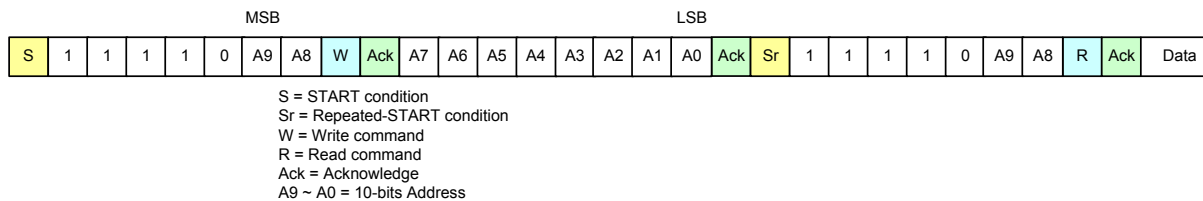


Figure 141. 10-bit Addressing Read Receive Mode

Inter-Integrated Circuit (I2C)

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/ \overline{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.

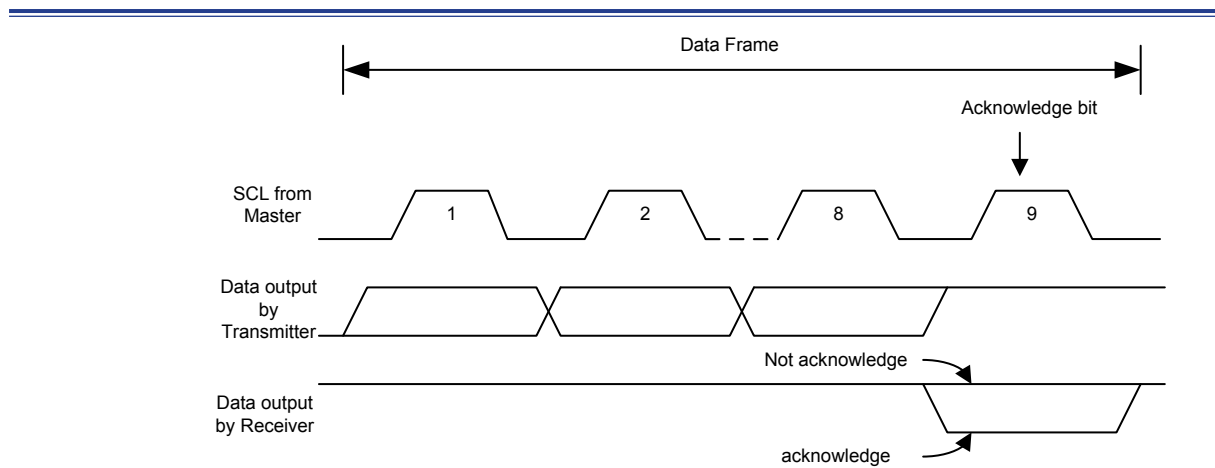


Figure 142. I²C Bus Acknowledge

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.

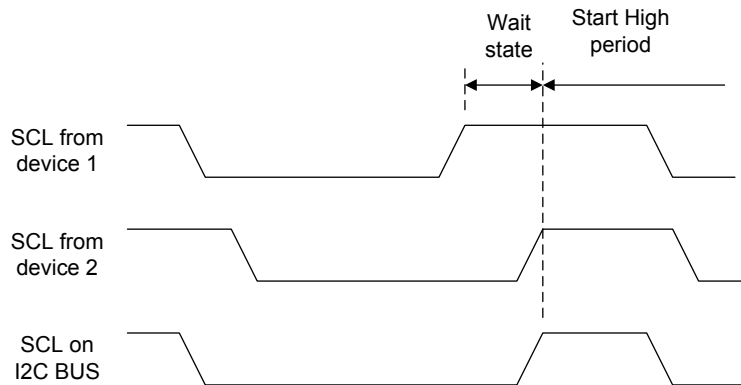


Figure 143. Clock Synchronization during Arbitration

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, ARBLSIEN, 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.

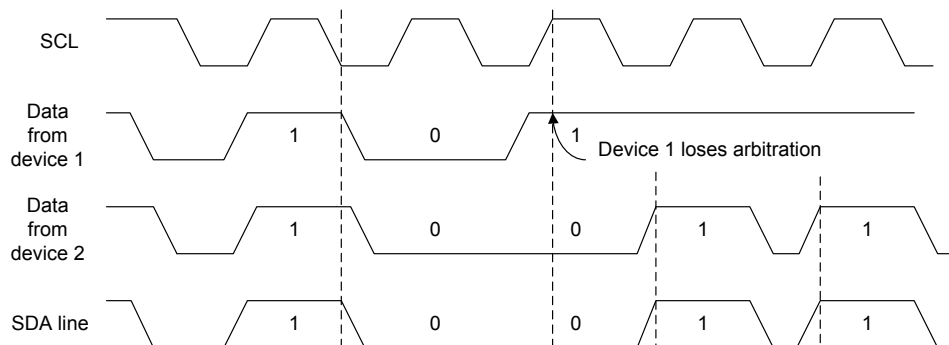


Figure 144. Two Master Arbitration Procedure

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 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 address mask function for user 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 7-bit addressing mode together with the I2CADDR register value as 0x05h and the I2CADDR register value as 0x55h, this means if an address which is sent by an I²C master on the bus is equal to 0x50h, 0x51h, 0x54h or 0x55h, 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 one of 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.

Close / Continue Transmission

After transmitting the last data byte, the STOP bit in the I2CCR register can be set to terminate the transmission or re-assign another slave device by configuring the I2CTAR register to restart a new transfer.

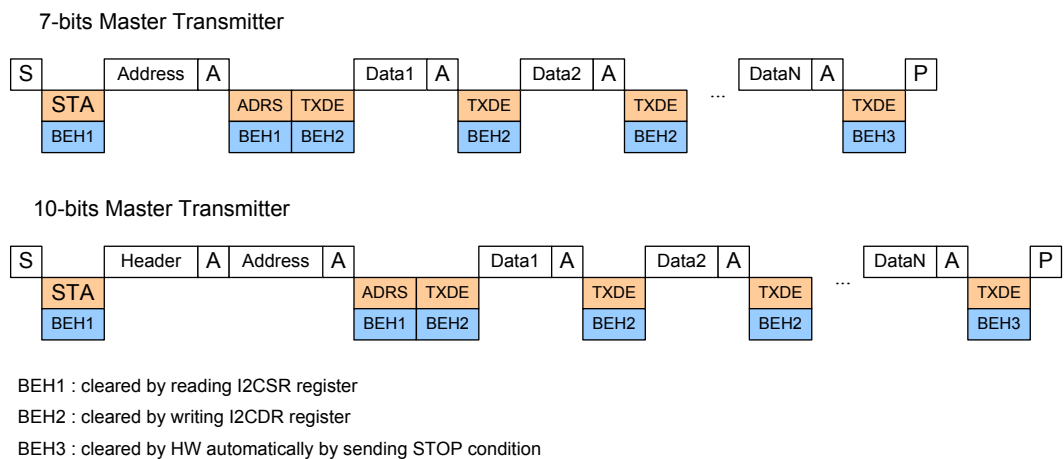


Figure 145. Master Transmitter Timing Diagram

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 10-bit address is sent and the acknowledge signal from the slave device is received. The second time the ADRS bit is set is when the 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 after 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.

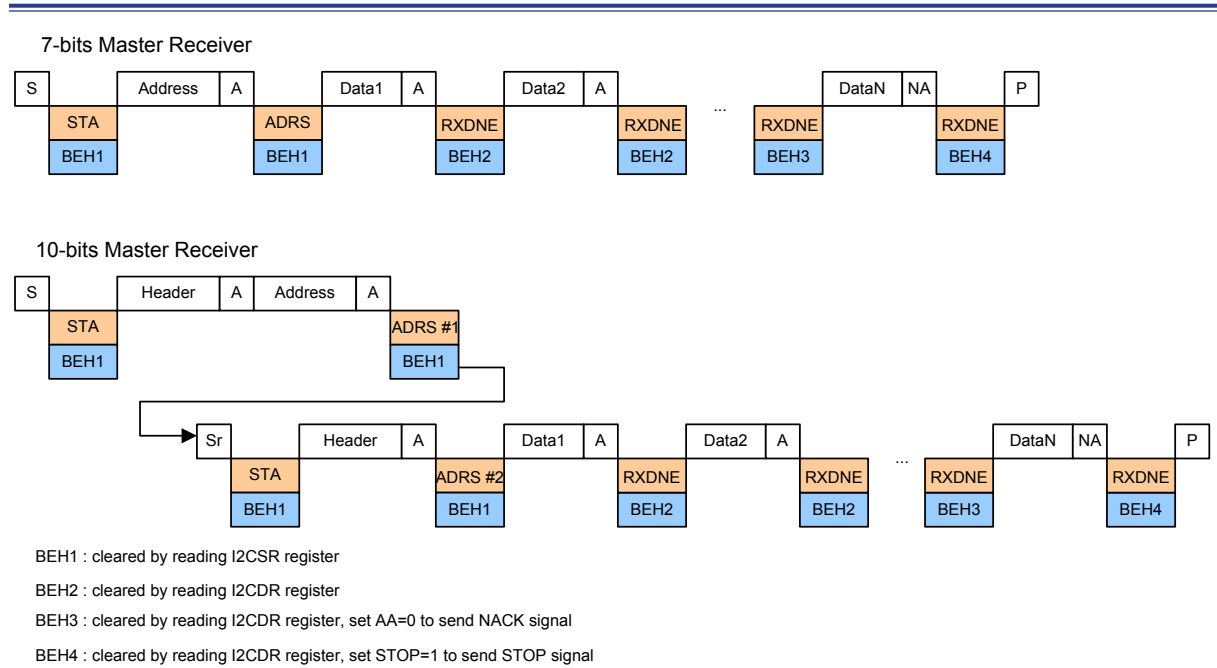


Figure 146. Master Receiver Timing Diagram

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.

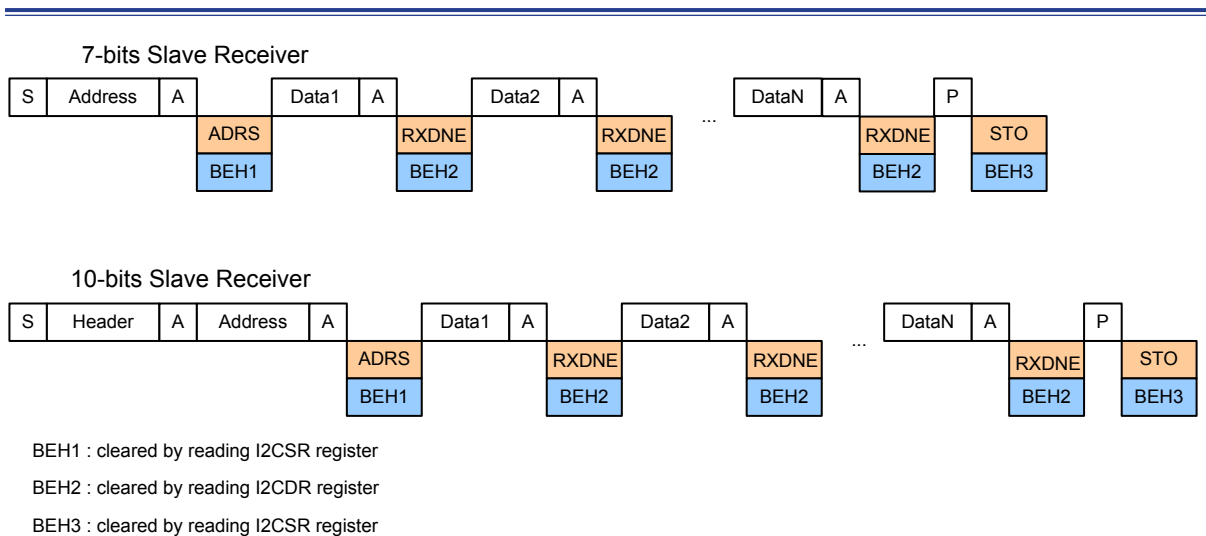


Figure 148. Slave Receiver Timing Diagram

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 46. Conditions of Holding SCL line

Type	Condition	Description	Eliminated
Flag	TXDE	I ² C is used in transmitted mode and I2CDR register needs to have data to transmit. (Note: TXDE won't be assert 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 is sent over address frame and is returned an ACK from slave (Note: Reference Fig.145 and Fig.146) Slave: I ² C is addressed as slave device (Note: Reference Fig.147 and Fig.148)	Reading I2CSR register
	STA	Master send START signal	Reading I2CSR register
	RXBF	Received a complete new data and meanwhile the RXDNE flag has been set already before.	Reading I2CDR register
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 send NACK used in received mode	Occurred when receiving the last data byte in Master received mode. (Note: Reference Fig.145, and RXNACK flag won't be assert at 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 flags are asserted.

The timeout counter will stop counting when the ENTOUT bit is cleared. However, the counter will also stop counting when the conditions, listed as follows occur:

- 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 flags in the I2CSR register are 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 on the PDMA configurations, refer to the PDMA chapter.

Register Map

The following table shows the I²C registers and reset values.

Table 47. 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 (0)

Reset value: 0x0000_2000

	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	SEQFILTER	COMBFILTEREn	ENTOUT	Reserved	DMANACK	RXDMAE	TXDMAE					
Type/Reset	RW	0	RW	0	RW	1	RW	0												
	7	6	5	4	3	2	1	0	ADRM	Reserved			I2CEN	GCEN	STOP	AA				
Type/Reset	RW	0											RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[15:14]	SEQFILTER	SDA or SCL Input Sequential Filter Configuration Bits 00: Sequential filter disable 01: 1 PCLK glitch filter 1x: 2 PCLK glitch filter Note: This setting would affect the frequency of SCL. Detail is described in I2CSLPGR register.
[13]	COMBFILTEREn	SDA or SCL Input Combinational Filter Enable Bit 0: Combinational filter Disable 1: Combinational filter Enable
[12]	ENTOUT	I ² C Timeout Function Enable Control 0: Timeout Function disabled 1: Timeout Function enabled This bit is used to enable or disable the I ² C 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.
[10]	DMANACK	DMA Mode NACK Control 0: No operation 1: The I ² C master receiver module sends a NACK signal automatically after receiving the last byte from the slave transmitter in the DMA mode When the I2CEN bit is cleared to 0, the DMANACK bit is automatically cleared to 0 by hardware.

Bits	Field	Descriptions
[9]	RXDMAE	<p>DMA Mode RX Request Enable Control</p> <p>0: RX DMA request disabled 1: RX DMA request 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 disabled 1: TX DMA request 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 disabled 1: I²C interface enabled</p>
[2]	GCEN	<p>General Call Enable</p> <p>0: General call disabled 1: General call 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>
[0]	AA	<p>Acknowledge Bit</p> <p>0: Send a Not Acknowledge (NACK) signal after a byte is received 1: Send an Acknowledge (ACK) signal after a byte is received</p> <p>When the I2CEN bit is cleared to 0, the AA bit is automatically cleared to 0 by hardware.</p>

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
Reserved							
Type/Reset							
23	22	21	20	19	18	17	16
Reserved					RXBFIE	TXDEIE	RXDNEIE
					RW 0	RW 0	RW 0
Type/Reset							
15	14	13	12	11	10	9	8
Reserved				TOUTIE	BUSERRIE	RXNACKIE	ARBLOSIE
				RW 0	RW 0	RW 0	RW 0
Type/Reset							
7	6	5	4	3	2	1	0
Reserved				GCSIE	ADRSIE	STOIE	STAIE
				RW 0	RW 0	RW 0	RW 0
Type/Reset							

Bits	Field	Descriptions
[18]	RXBFIE	RX Buffer Full Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled When the I2CEN bit in the I2CCR register is cleared to 0, this bit is cleared to 0 by hardware.
[17]	TXDEIE	Data Register Empty Interrupt Enable Bit in Transmitter Mode 0: Interrupt disabled 1: Interrupt enabled When the I2CEN bit in the I2CCR register is cleared to 0, this bit is cleared to 0 by hardware.
[16]	RXDNEIE	Data Register Not Empty Interrupt Enable Bit in Received Mode 0: Interrupt disabled 1: Interrupt enabled When the I2CEN bit in the I2CCR register is cleared to 0, this bit is cleared to 0 by hardware.
[11]	TOUTIE	Timeout Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled When the I2CEN bit in the I2CCR register is cleared to 0, this bit is cleared to 0 by hardware.
[10]	BUSERRIE	Bus Error Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt 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 disabled 1: Interrupt enabled When the I2CEN bit in the I2CCR register is cleared to 0, this bit is cleared to 0 by hardware.

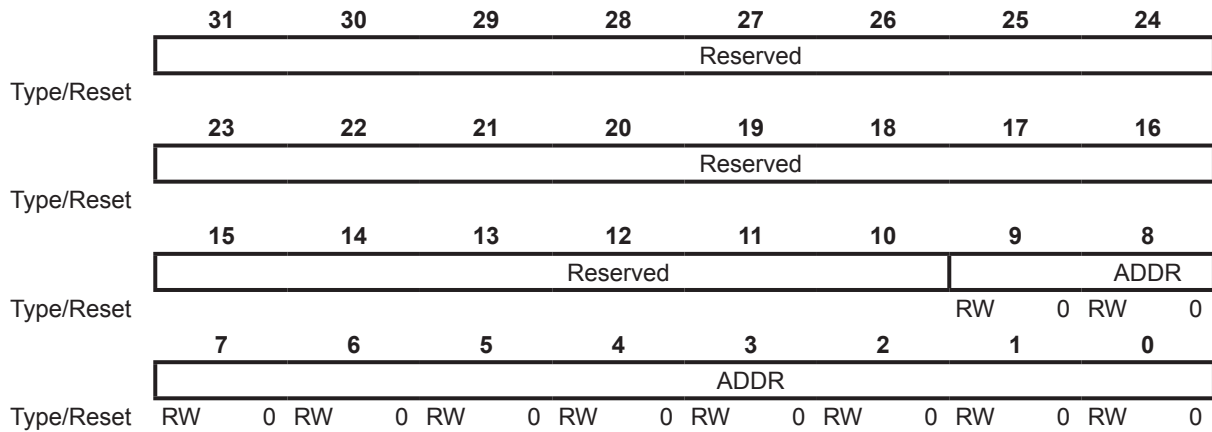
Bits	Field	Descriptions
[8]	ARBLOSIE	Arbitration Loss Interrupt Enable Bit in the I ² C multi-master mode 0: Interrupt disabled 1: Interrupt enabled When the I2CEN bit in the I2CCR register is cleared to 0, this bit is cleared to 0 by hardware.
[3]	GCSIE	General Call Slave Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt 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 disabled 1: Interrupt 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 disabled 1: Interrupt 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 disabled 1: Interrupt 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



Bits	Field	Descriptions
[9:0]	ADDR	Device Address The register indicates the I ² C device address. When the I ² C 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 I ² C master device.

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
Reserved							
Type/Reset							
23	22	21	20	19	18	17	16
Reserved		TXNRX	MASTER	BUSBUSY	RXBF	TXDE	RXDNE
		RO	0 RO	0 RO	0 RO	0 RO	0 RO
Type/Reset							
15	14	13	12	11	10	9	8
Reserved				TOUTF	BUSERR	RXNACK	ARBLOS
				WC	0 WC	0 WC	0 WC
Type/Reset							
7	6	5	4	3	2	1	0
Reserved				GCS	ADRS	STO	STA
				RC	0 RC	0 RC	0 RC
Type/Reset							

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 Using in Transmitter Mode</p> <p>0: Data register I2CDR not empty 1: Data register I2CDR 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 empty 1: Data register I2CDR 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 occurred 1: Timeout counter underflow 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 occurs 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 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 bit of the first header byte and the second address.</p> <p>Address Matched in Slave Mode</p> <p>0: I²C interface is not addressed 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 detected 1: STOP condition 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 detected 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	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0								
Type/Reset	SHPG															
	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[15:0]	SHPG	<p>SCL Clock High Period Generation</p> <p>High period duration setting $SCL_{HIGH} = T_{PCLK} \times (SHPG + d)$</p> <p>where T_{PCLK} is the APB bus peripheral clock (PCLK) period, and d value depends on the setting of the SEQFILTER field 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	Reserved							
Type/Reset																
	23	22	21	20	19	18	17	16	Reserved							
Type/Reset																
	15	14	13	12	11	10	9	8	SLPG							
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	SLPG							
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[15:0]	SLPG	<p>SCL Clock Low Period Generation</p> <p>Low period duration setting $SCL_{LOW} = T_{PCLK} \times (SLPG + d)$</p> <p>where T_{PCLK} is the APB bus peripheral clock (PCLK) period, and d value depends on the setting of the SEQFILTER field 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>

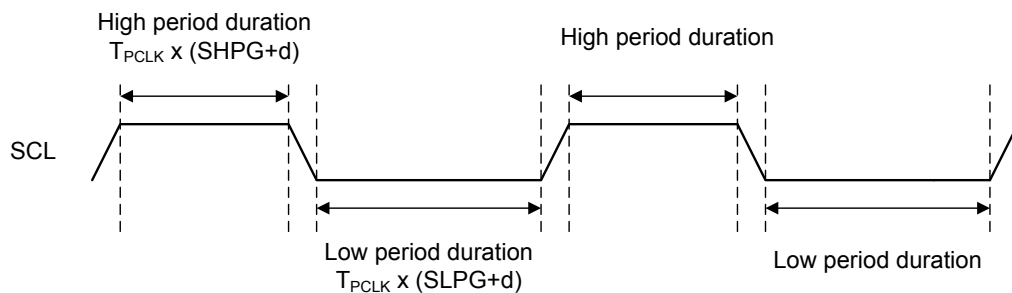


Figure 149. SCL Timing Diagram

Table 48. 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			
	8MHz	24MHz	48MHz	72MHz
100 kHz (Standard Mode)	68	228	468	708
400 kHz (Fast Mode)	8	48	108	168
1 MHz (Fast-Mode Plus)	x	12	36	60

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	RW
									0

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	
	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, the timing 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 – I2CADDR

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	0
	7	6	5	4	3	2	1	0		
Type/Reset	ADDR									
	RW	0	RW	0	RW	0	RW	0	RW	0

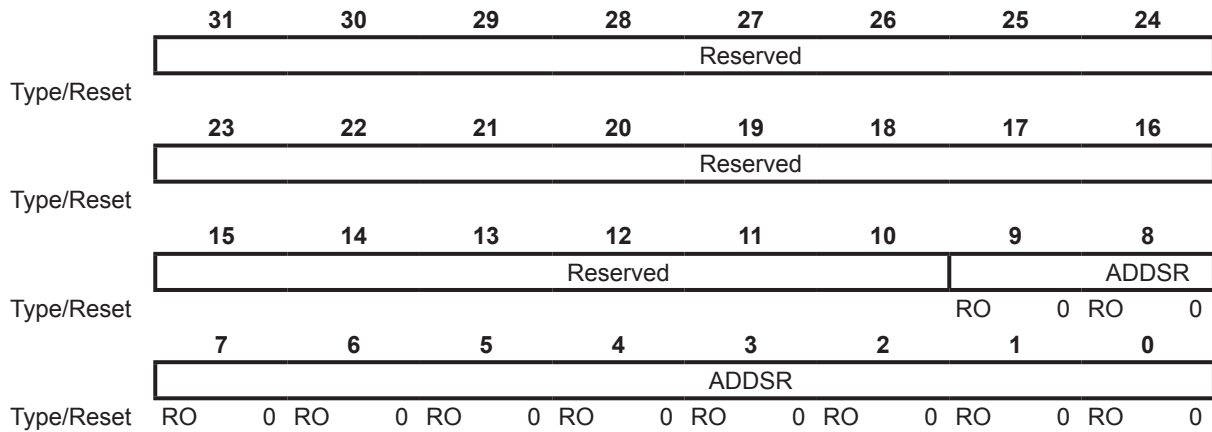
Bits	Field	Descriptions
[9:0]	ADDR	<p>Address Mask Control Bit</p> <p>The ADDR[i] is used to specify whether the ith bit of the ADDR in the I2CADDR register is masked and is compared with the received address frame or not on the I²C bus. The register is only used for the I²C slave mode only.</p> <p>0: 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



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 ADDRSR 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 Time-out Counter Prescaler Selection</p> <p>This PSC field is used to specify the I²C time-out counter clock frequency, f_{I2CTO}. The time-out clock frequency is obtained using the 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 the following conditions occur:</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 the following conditions occur:</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.

21 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 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 in 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.

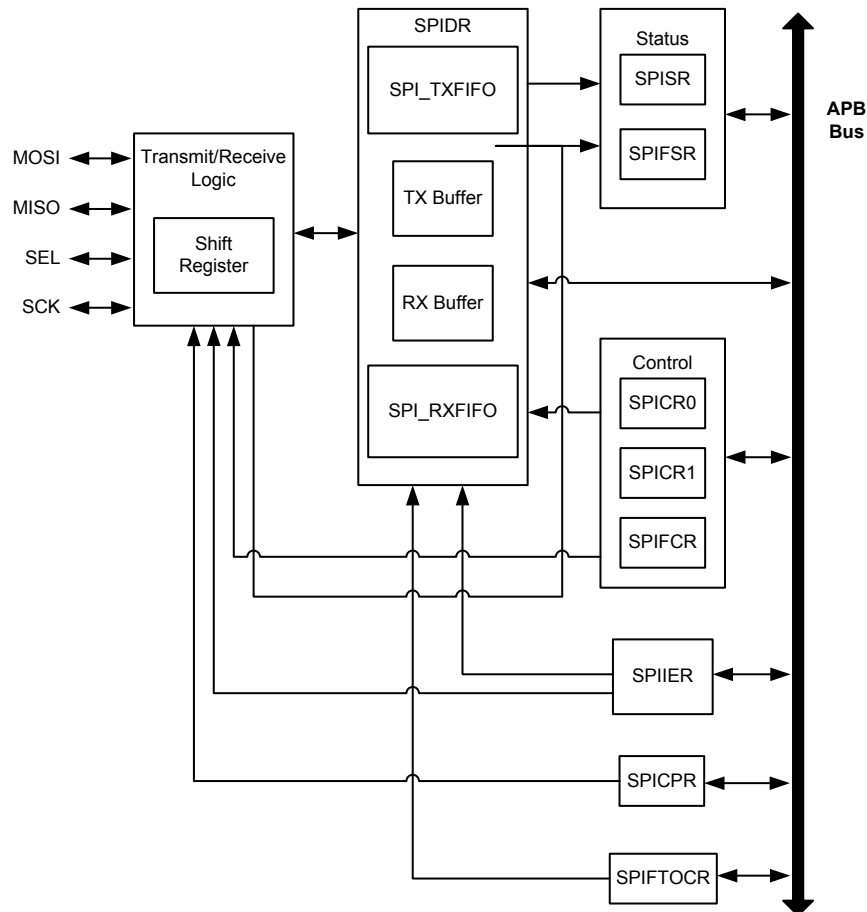


Figure 150. SPI Block Diagram

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 the 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

Function 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 byte 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 49 shows how to configure these formats by setting the FORMAT field in the SPICR1 register.

Table 49. 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 151 shows the single byte data transfer timing of this format.

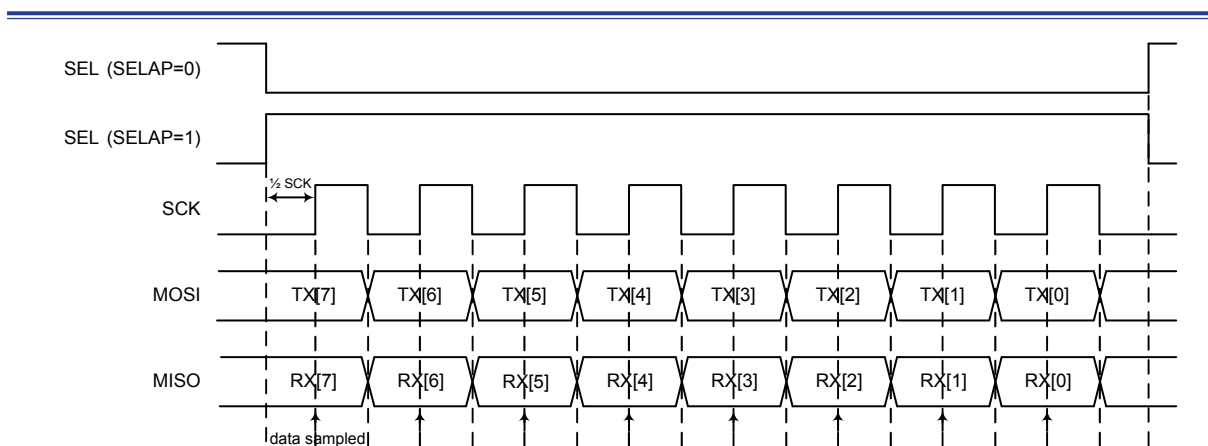


Figure 151. SPI Single Byte Transfer Timing Diagram – CPOL = 0, CPHA = 0

Figure 152 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.

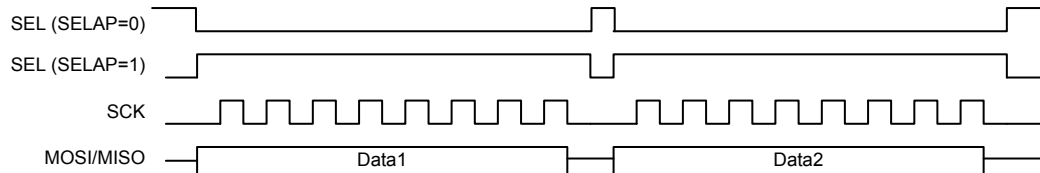


Figure 152. SPI Continuous Data Transfer Timing Diagram – CPOL = 0, CPHA = 0

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 153 shows the single data byte transfer timing.

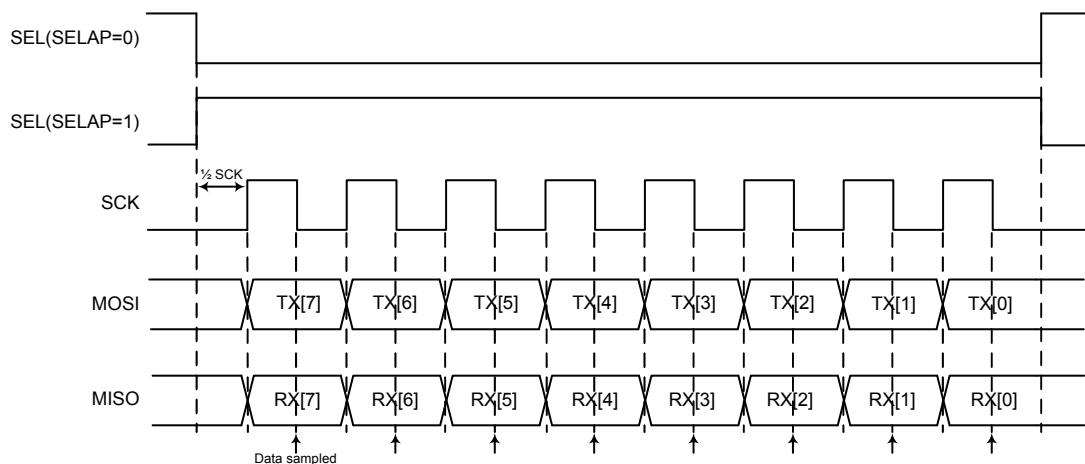


Figure 153. SPI Single Byte Transfer Timing Diagram – CPOL = 0, CPHA = 1

Figure 154 shows the continuous data transfer diagram timing. Note that the SEL signal must remain active until the last data transfer has completed.

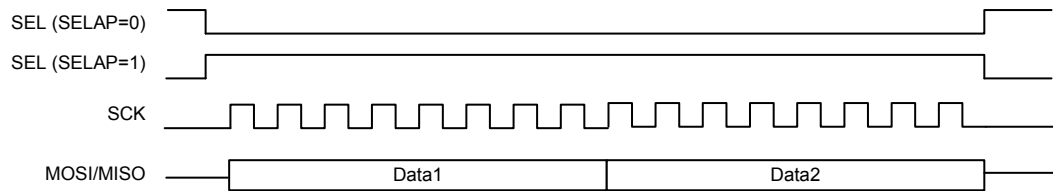


Figure 154. SPI Continuous Transfer Timing Diagram – CPOL = 0, CPHA = 1

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 155 shows the single byte transfer timing of this format.

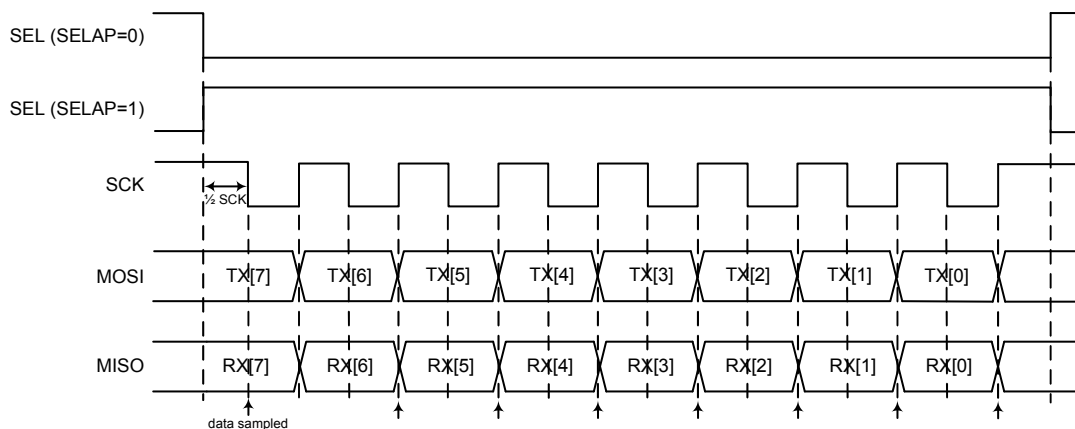


Figure 155. SPI Single Byte Transfer Timing Diagram – CPOL = 1, CPHA = 0

Figure 156 shows the continuous data transfer timing of this format. Note that the SEL signal must change to an inactive level between each data frame.

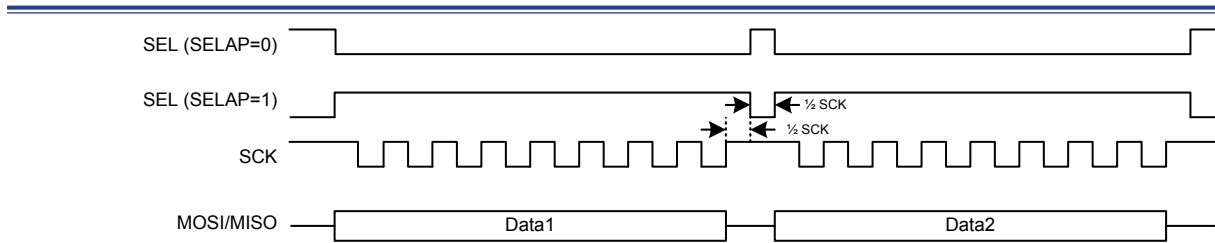


Figure 156. SPI Continuous Transfer Timing Diagram – CPOL = 1, CPHA = 0

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 157 shows the single byte transfer timing of this format.

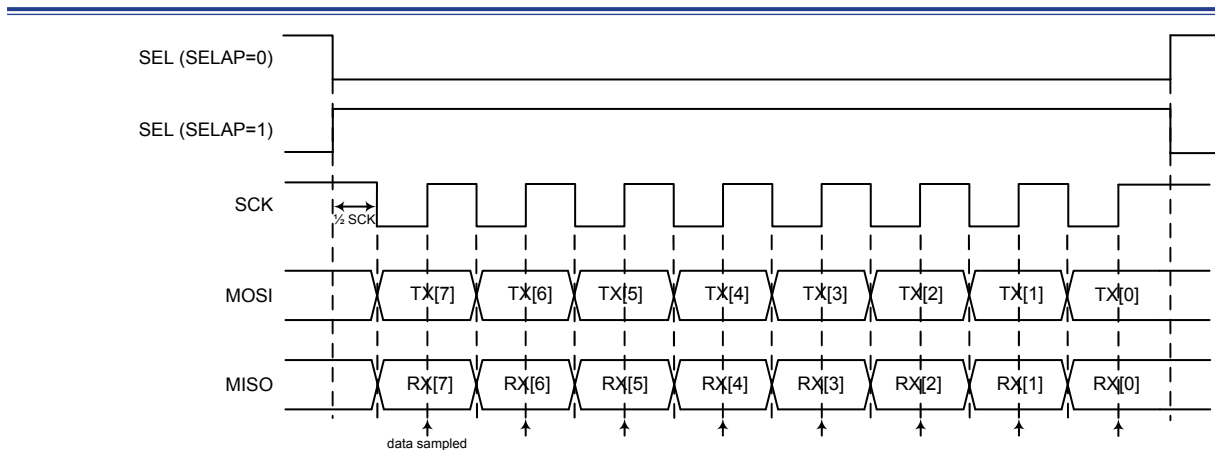


Figure 157. SPI Single Byte Transfer Timing Diagram – CPOL = 1, CPHA = 1

Figure 158 shows the continuous data transfer timing of this format. Note that the SEL signal must remain active until the last data transfer has completed.

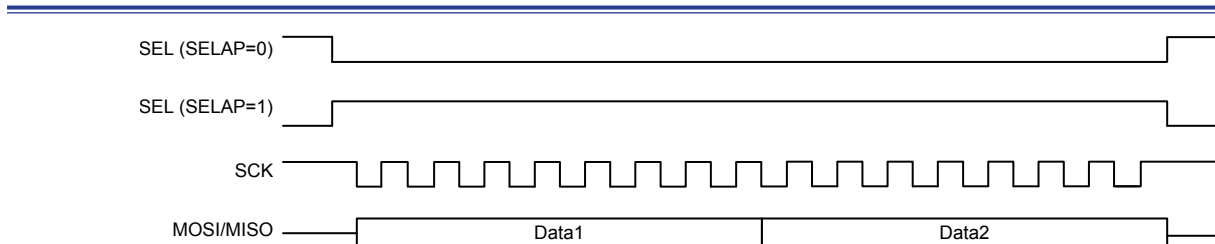


Figure 158. SPI Continuous Transfer Timing Diagram – CPOL = 1, CPHA = 1

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 the TX FIFO data length is greater than the TX FIFO threshold level determined by the TXFTLS bits 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 setup 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 its SEL signal to an active state. The detailed configuration diagram for the SPI multi-master mode is shown in the following figure.

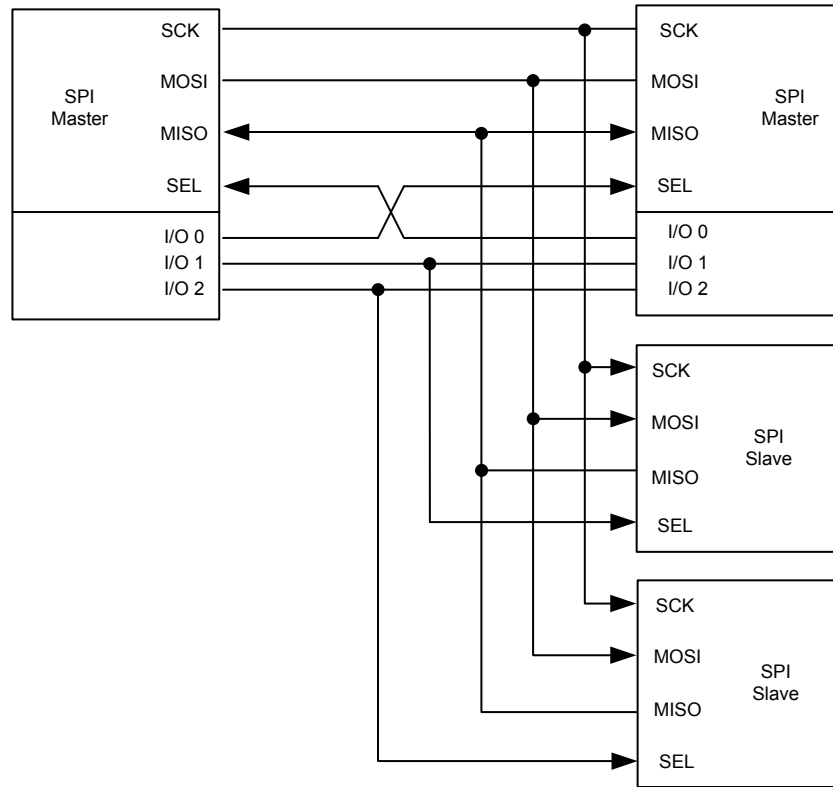


Figure 159. SPI Multi-Master Slave Environment

Table 50. 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 51. SPI Master Mode SEL Pin Status

	SEL as Input - SELOEN = 0		SEL as Output - SELOEN = 1	
Multi-master	Support		Not support	
SPI SEL control signal	Use Another GPIO to replace the SEL pin function		SEL pin in hardware or software mode - using SELM setting	
Continuous transfer	Case 1	Case 2	Case 1	Case 2
	Not supported	Supported	Using 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 active 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 on the PDMA configurations, refer to the PDMA chapter.

Register Map

The following table shows the SPI registers and their reset values.

Table 52. 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

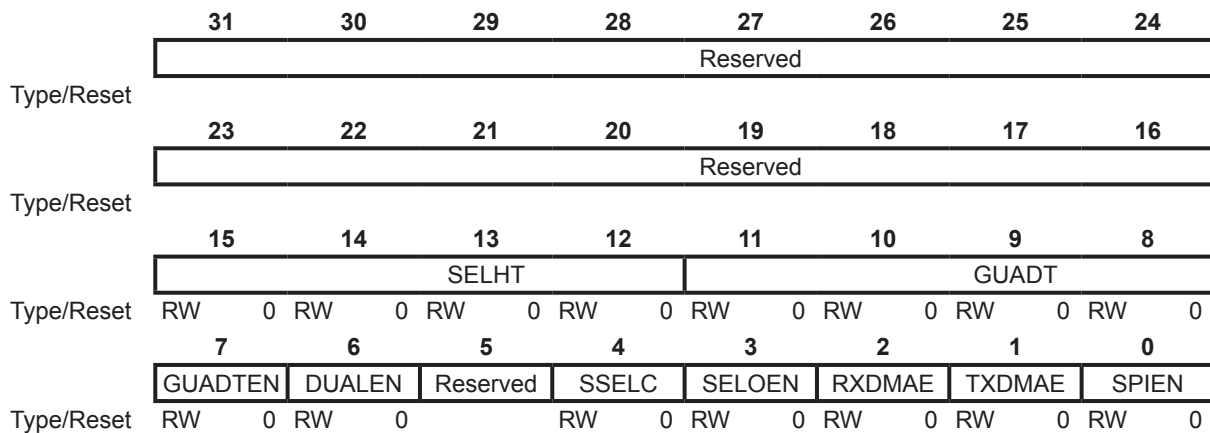
Register Descriptions

SPI Control Register 0 – SPICR0

This register specifies the SEL control and the SPI enable bits.

Offset: 0x000

Reset value: 0x0000_0000



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 set this bit, 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 setup 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 setup 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 disabled. 1: SPI RX path PDMA request enabled.
[1]	TXDMAE	TX PDMA request enable 0: SPI TX path PDMA request disabled. 1: SPI TX path PDMA request 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 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0	
Type/Reset	Reserved				DFL				
					RW 0	RW 0	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_C bit 1: SEL signal is controlled by hardware – generated automatically by the SPI hardware Note that SELM bit is available for master mode only - MODE = 1
[12]	FIRSTBIT	LSB or MSB Transmitted First 0: MSB transmitted first 1: LSB 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>Reserved</td> <td></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	WCIEN	RXBNEIEN	TXEIEN	TXBEIEN	
	RW	0	RW	0	RW	0	RW	0	RW

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]	WCIEN	Write Collision Interrupt Enable 0: Disable 1: Enable
[2]	RXBNEIEN	RX Buffer Not Empty Interrupt Enable 0: Disable 1: Enable Generates an interrupt request when the RXBNE flag is set and when RXBNEIEN is set. In the FIFO mode, the interrupt request being generated depends upon the RX FIFO trigger level setting
[1]	TXEIEN	TX Empty Interrupt Enable 0: Disable 1: Enable The TX register empty interrupt request will be generated when the TXE flag and the TXEIEN bit are set

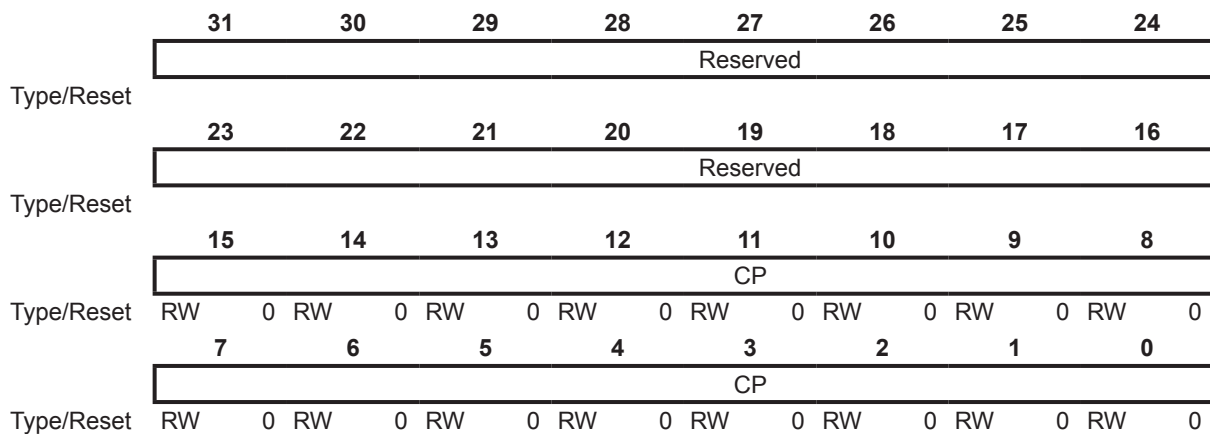
Bits	Field	Descriptions
[0]	TXBEIEN	TX Buffer Empty Interrupt Enable 0: Disable 1: Enable The TX buffer empty interrupt request will be generated when the TXBE flag and the TXBEIEN bit are set. In the FIFO mode, the interrupt request being generated depends upon the TX FIFO trigger level setting.

SPI Clock Prescaler Register – SPICPR

This register specifies the SPI clock prescaler ratio.

Offset: 0x00C

Reset value: 0x0000_0000



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	Reserved							
Type/Reset																
	23	22	21	20	19	18	17	16	Reserved							
Type/Reset																
	15	14	13	12	11	10	9	8	DR							
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	DR							
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[15:0]	DR	Data Register 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.

SPI Status Register – SPISR

This register contains the relevant SPI status.

Offset: 0x014

Reset value: 0x0000_0003

	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								BUSY	
Type/Reset																	RO	0
	7	6	5	4	3	2	1	0	TO	SA	MF	RO	WC	RXBNE	TXE	TXBE		
Type/Reset	WC	0	WC	0	WC	0	WC	0	WC	0	WC	0	RO	0	RO	1	RO	1

Bits	Field	Descriptions
[8]	BUSY	<p>SPI Busy flag</p> <p>0: SPI not busy</p> <p>1: SPI busy</p> <p>In the master mode, this flag is reset when the TX buffer and TX shift register are both empty and is set when the TX buffer or the TX shift register are not empty.</p> <p>In the slave mode, this flag is set when SEL changes to an active level and is reset when SEL changes to an inactive level.</p>
[7]	TO	<p>Time Out flag</p> <p>0: No RX FIFO time out</p> <p>1: RX FIFO time out has occurred.</p> <p>Write 1 to clear it.</p> <p>Once the time out counter value is equal to the TOC field setting in the SPIFCR register, the time out flag will be set and an interrupt will be generated if the TOIEN bit in the SPIIER register is enabled. This bit is cleared by writing 1</p> <p>Note: This Time Out flag function is only available in the SPI FIFO mode.</p>
[6]	SA	<p>Slave Abort flag</p> <p>0: No slave abort</p> <p>1: Slave abort has occurred.</p> <p>This bit is set by hardware and cleared by writing 1.</p>
[5]	MF	<p>Mode Fault flag</p> <p>0: No mode fault</p> <p>1: Mode fault has occurred</p> <p>This bit is set by hardware and cleared by writing 1.</p>
[4]	RO	<p>Read Overrun flag</p> <p>0: No read overrun</p> <p>1: Read overrun has occurred.</p> <p>This bit is set by hardware and cleared by writing 1.</p>

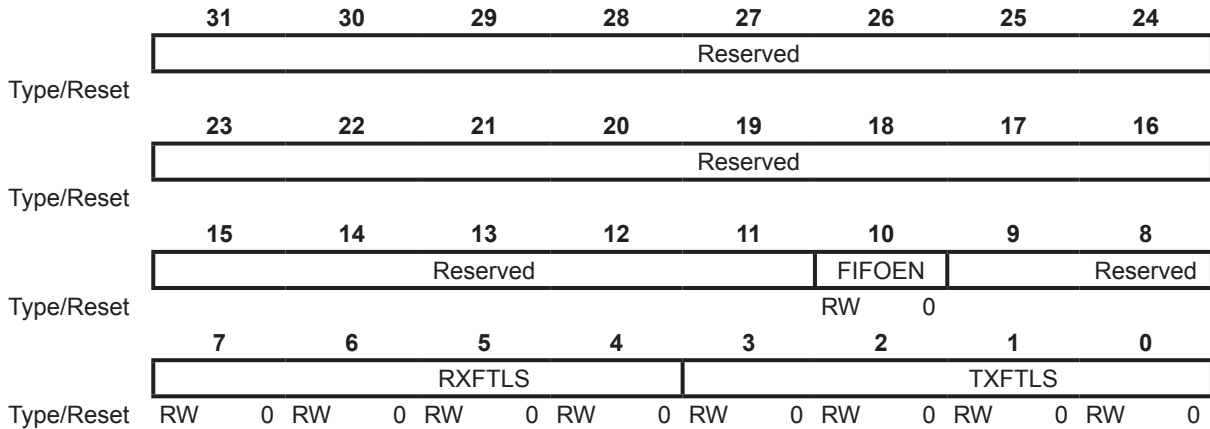
Bits	Field	Descriptions
[3]	WC	Write Collision flag 0: No write collision 1: Write collision has occurred. This bit is set by hardware and cleared by writing 1.
[2]	RXBNE	Receive Buffer Not Empty flag 0: RX buffer empty 1: RX buffer not empty This bit indicates the RX buffer status in the non-FIFO mode. 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	Transmit Buffer Empty flag 0: TX buffer not empty 1: TX buffer empty In the FIFO mode, this bit 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



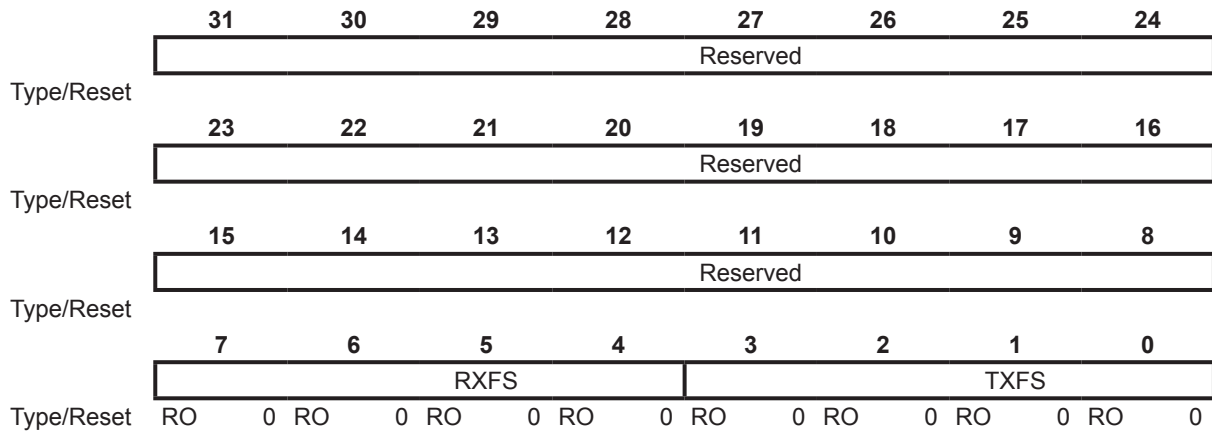
Bits	Field	Descriptions
[10]	FIFOEN	FIFO Enable 0: FIFO disable 1: FIFO enable This bit cannot be set or reset when the SPI interface is in transmitting.
[7:4]	RXFTLS	RX FIFO Trigger Level Select 0000: Trigger level is 0 0001: Trigger level is 1 ... 1000: Trigger level is 8 Others: Reserved The RXFTLS 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



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
[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	Reserved							
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0
	23	22	21	20	19	18	17	16	Reserved							
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	TOC							
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	TOC							
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[15:0]	TOC	<p>Time Out Counter</p> <p>The time out counter starts to count from 0 after the SPI RX FIFO receives a data, and reset the counter value 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 SPIEN 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}.</p>

22 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-byte transmit FIFO, TX FIFO, and a 8-byte receive FIFO, RX FIFO. Software can detect a USART error status by reading 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 of the CK_APB (CK_USART) to produce a baud rate clock for the USART transmitter and receiver.

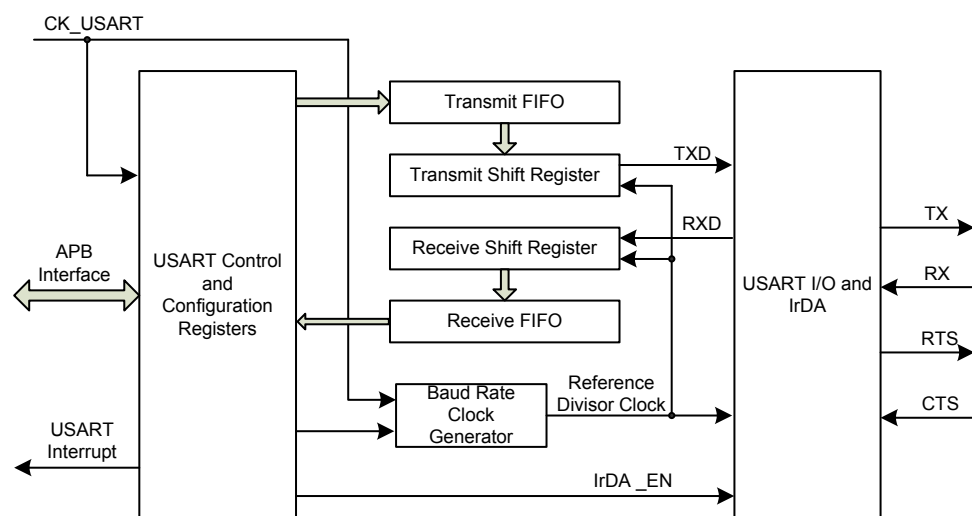


Figure 160. USART Block Diagram

Features

- Supports both asynchronous and clocked synchronous serial communication modes
- Full Duplex Communication Capability
- Programming baud rate up to 3 Mbit/s for asynchronous mode and 6 Mbit/s for synchronous mode
- IrDA SIR encoder and decoder
 - Support of normal 3/16 bit duration and low-power (1.41 ~ 2.23 μ s) 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 bit generation
 - Bit order: LSB-first or MSB-first transfer
- Error detection: Parity, overrun, and frame error
- FIFO:
 - Receive FIFO: 8 \times 9 bits (max 9 data bits)
 - Transmit FIFO: 8 \times 9 bits (max 9 data bits)
- Supports PDMA Interface

Function 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 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. The both 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 to have failed and respond with a Framing Error.

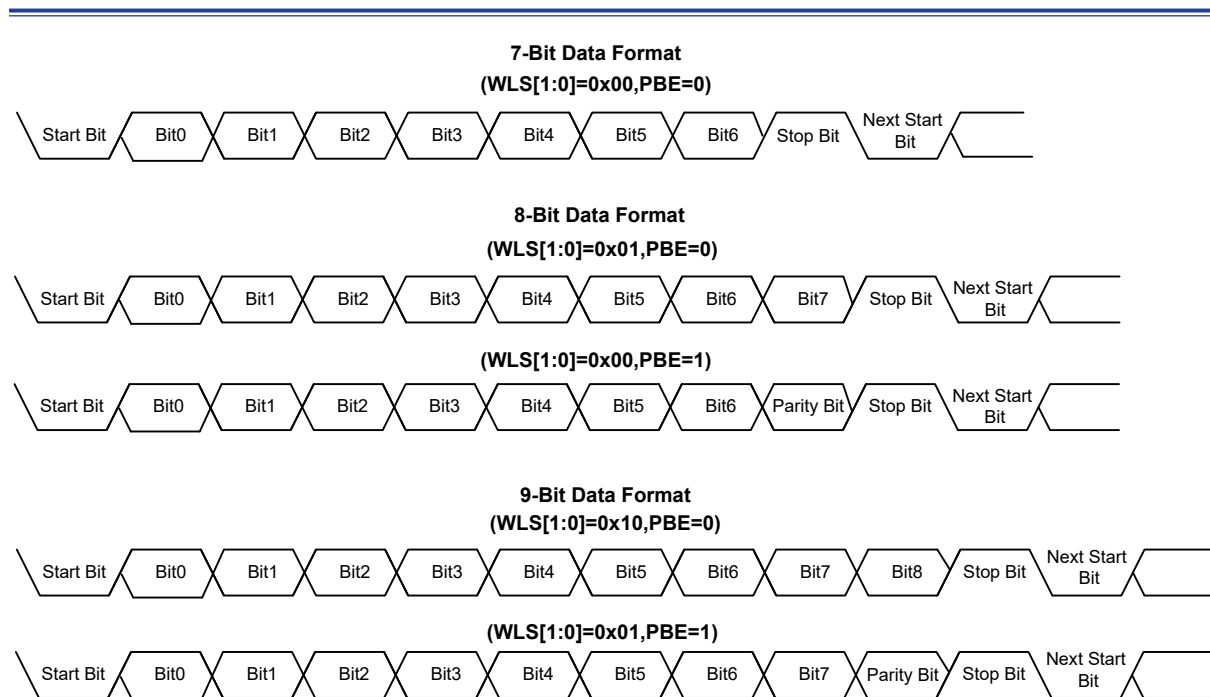


Figure 161. USART Serial Data Format

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 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.

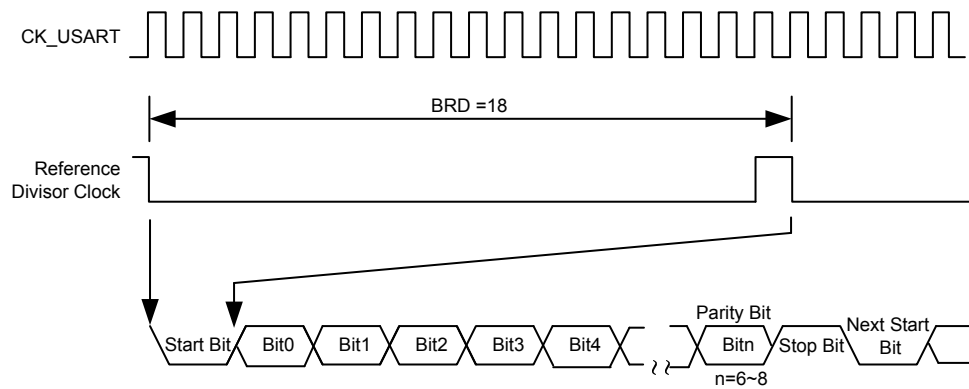


Figure 162. USART Clock CK_USART and Data Frame Timing

Table 53. 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	3076.9	13	2.56%

Table 54. 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	16	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 2 USART devices by using the CTS input and the RTS output. The Figure 163 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.

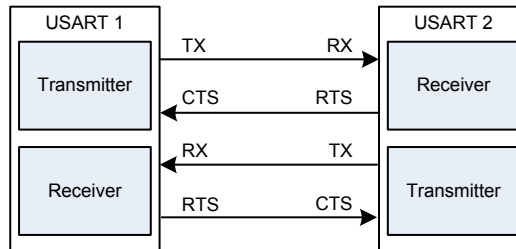


Figure 163. Hardware Flow Control between 2 USARTs

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 164 shows the example of RTS flow control.

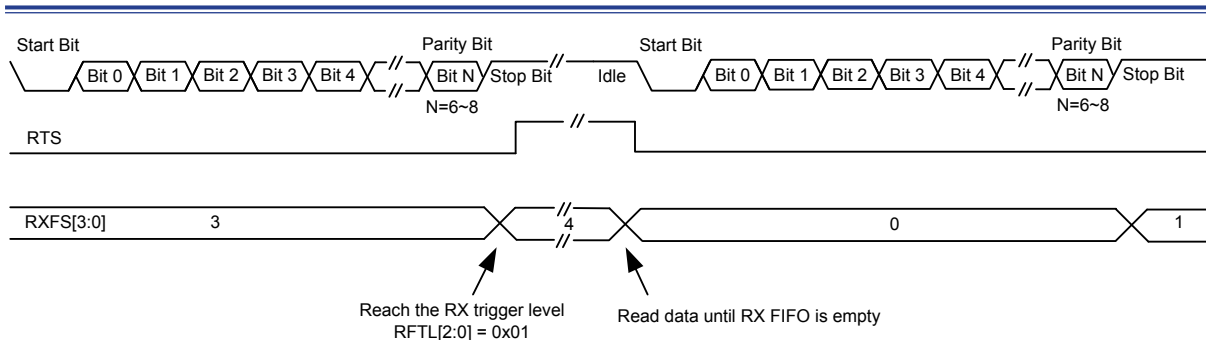


Figure 164. USART RTS Flow Control

CTS Flow Control

If the hard flow control function is enabled, the URTXEN bit in the USRCR register will be 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 165 shows an example of communication with CTS flow control.

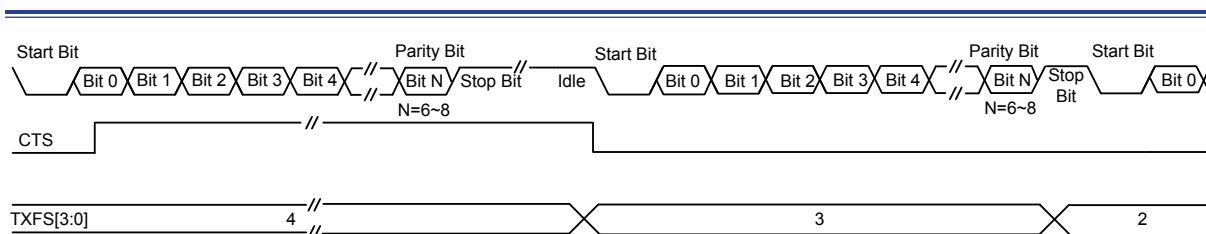


Figure 165. USART CTS Flow Control

IrDA

The USART IrDA mode is provided 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 a 'high' level in the IrDA mode.

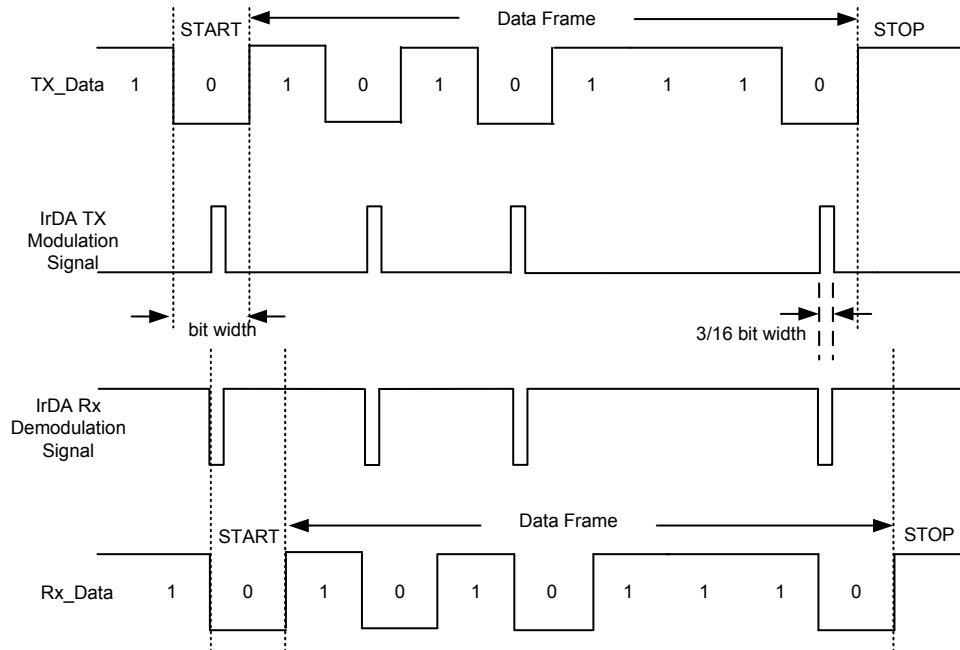


Figure 166. IrDA Modulation and Demodulation

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 transmitted pulse width in the low power mode.

The IrDAPSC field is the IrDA 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.

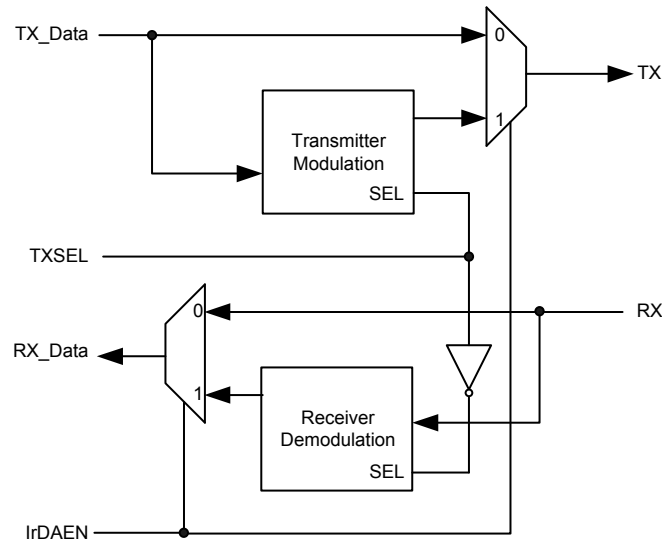


Figure 167. USART I/O and IrDA Block Diagram

RS485 Mode

The RS485 mode of the USART provides the data transmission on the 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.

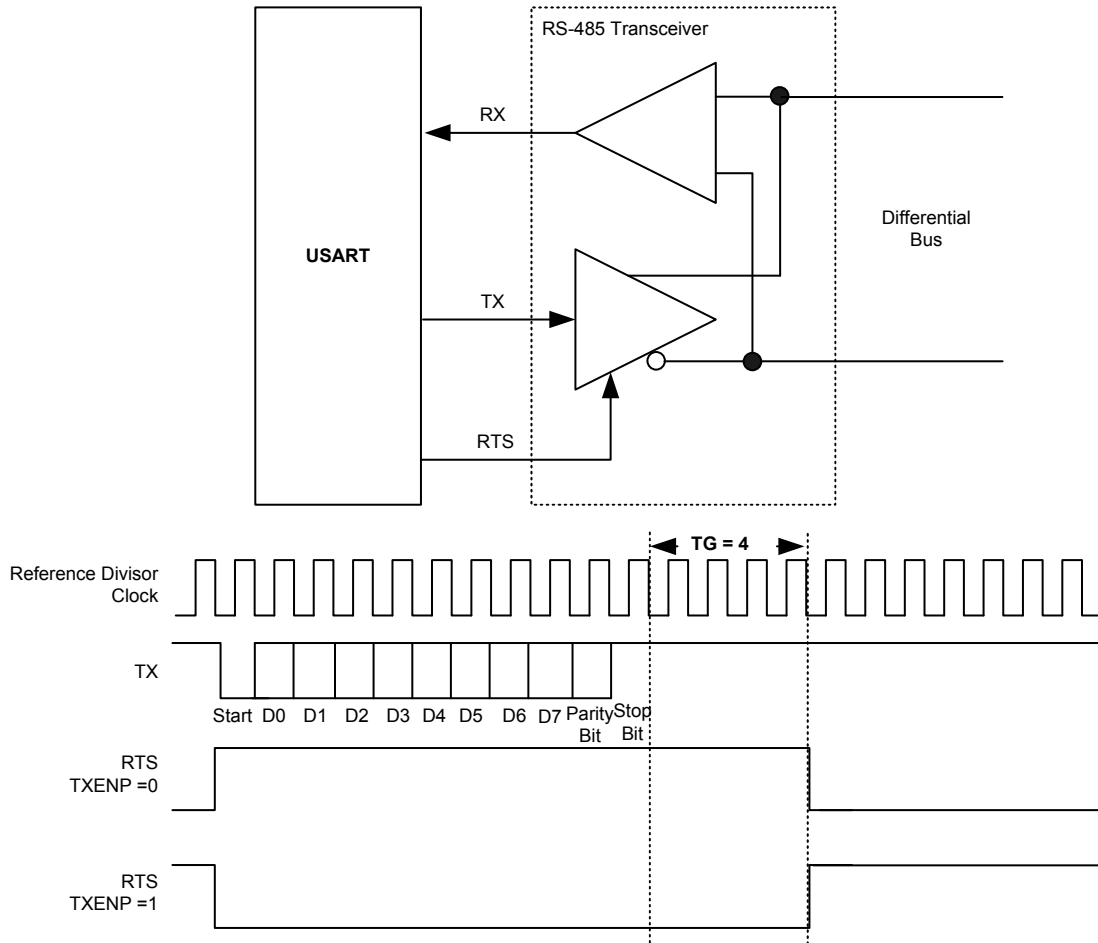


Figure 168. RS485 Interface and Waveform

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 RXFIFO. Once the first address data is detected and stored in the RXFIFO, 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 RXFIFO. 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 RXFIFO and the URRXEN bit will be automatically set. When the receiver is enabled, all received data will be stored in the RXFIFO 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 use to configure the clock polarity in the USART Synchronous Mode idle state. Detailed timing information is shown in Figure 169.

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.

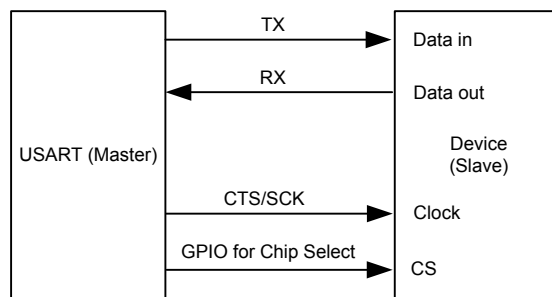


Figure 169. USART Synchronous Transmission Example

Note: The USART supports the synchronous master mode only: it cannot receive or send data related to an input clock. The USART CTS/SCK clock is always an output.

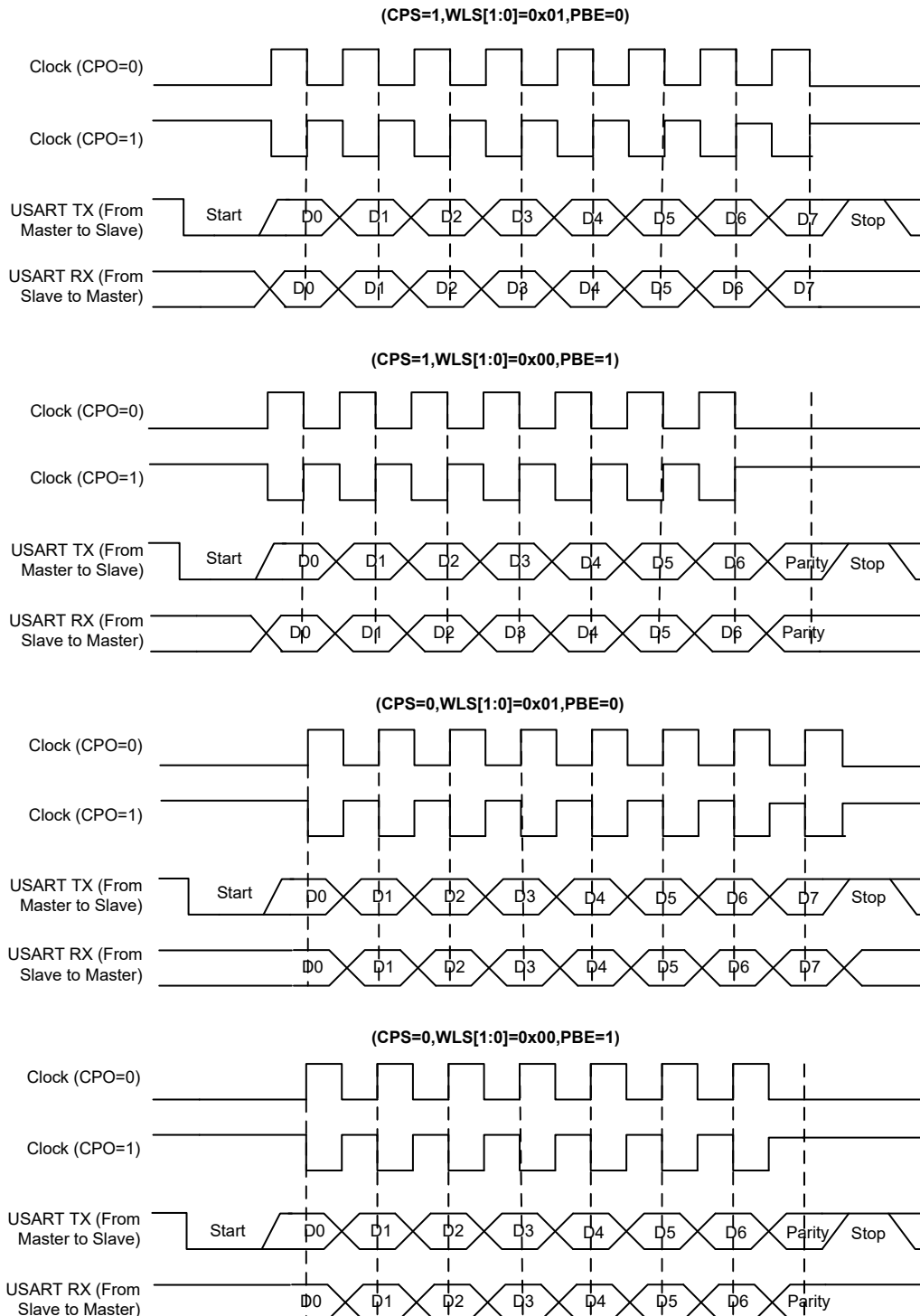


Figure 170. 8-bit Format USART Synchronous Waveform

Interrupts and Status

The USART can generate interrupts when the following event occurs and corresponding interrupt enable bits are set:

- Receive FIFO time-out interrupt: An interrupt will be generated when the USART receive FIFO does not receive a new data package during the specified time-out interval.
- Receiver line status interrupts: The interrupts will be generated when the USART receiver overrun error, parity error, framing error and break events occur.
- Transmit FIFO threshold level interrupt: An interrupt will be generated when the data to be transmitted in the USART Transmit FIFO is less than the specified threshold level.
- Transmit complete interrupt: An interrupt will be 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 will be 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 mode detailed description on the PDMA configurations, refer to the PDMA chapter.

Register Map

The following table shows the USART registers and reset values.

Table 55. 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	
									0
	7	6	5	4	3	2	1	0	
Type/Reset	DB								
	RW	0	RW	0	RW	0	RW	0	RW
									0

Bits	Field	Descriptions
[8:0]	DB	<p>Reading data via this receiver buffer register will return the data from the receive FIFO. The receive FIFO has a capacity of up to 8 x 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 9-bit mode only and is fixed at 0 for the 8-bit mode. For the 7-bit mode, the DB[6:0] 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 x 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] 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 selections.

Offset: 0x004

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				
RTS	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				
RXDMAEN	TXDMAEN	URRXEN	URTXEN	HFCEN	TRSM	MODE					
RW	0	RW	0	RW	0	RW	0	RW	0	RW	0
Type/Reset											

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 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 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 PBE is set to 1.

Bits	Field	Descriptions
[11]	PBE	Parity Bit Enable 0: Parity bit is not generated (transmitted data) or checked (receive 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.
[10]	NSB	Number of "STOP bit" 0: One "STOP bit" is generated in the transmitted data 1: Two "STOP bit" is generated when 8- and 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: Disabled 1: Enabled
[6]	TXDMAEN	USART TX DMA Enable 0: Disabled 1: Enabled
[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: Disabled 1: Enabled
[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	RO	0	RO	0	
23	22	21	20	19	18	17	16					
Reserved				TXFS								
Type/Reset				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					
RXTL		TXTL		Reserved		RXR	TXR					
Type/Reset	RW	0	RW	0	RW	0	RW	0	WO	0	WO	0

Bits	Field	Descriptions
[27:24]	RXFS	<p>RX FIFO Status</p> <p>The RXFS field shows the current number of data contained in the RX FIFO.</p> <p>0000: RX FIFO is empty 0001: RX FIFO contains 1 data ... 1000: RX FIFO contains 8 data Others: Reserved</p>
[19:16]	TXFS	<p>TX FIFO Status</p> <p>The TXFS field shows the current number of data contained in the TX FIFO.</p> <p>0000: TX FIFO is empty 0001: TX FIFO contains 1 data ... 1000: TX FIFO contains 8 data Others: Reserved</p>
[7:6]	RXTL	<p>RX FIFO Threshold Level Setting</p> <p>00: 1 byte 01: 2 bytes 10: 4 bytes 11: 6 bytes</p> <p>The RXTL field defines the RX FIFO trigger level.</p>
[5:4]	TXTL	<p>TX FIFO Threshold Level Setting</p> <p>00: 0 byte 01: 2 bytes 10: 4 bytes 11: 6 bytes</p> <p>The TXTL field determines the TX FIFO trigger level.</p>
[1]	RXR	<p>RX FIFO Reset</p> <p>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.</p>

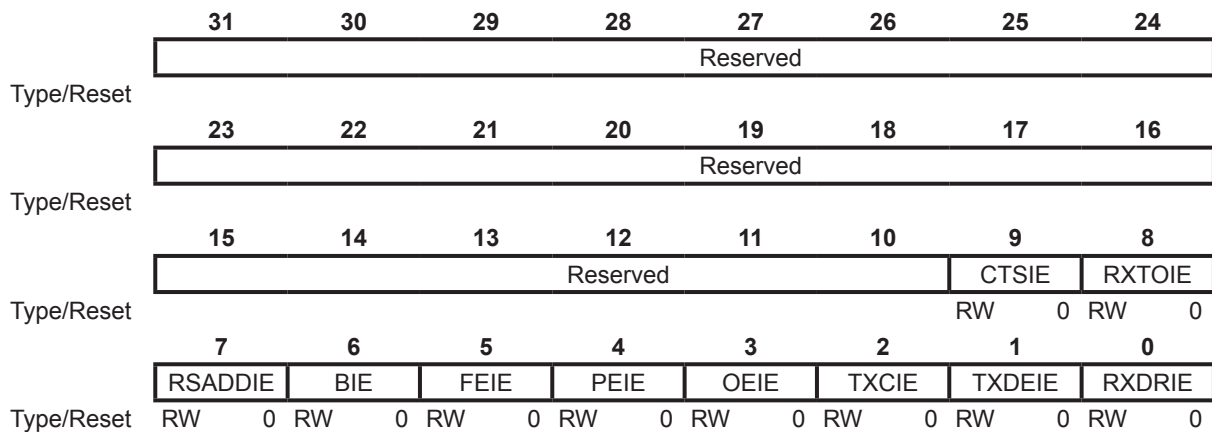
Bits	Field	Descriptions
[0]	TXR	TX FIFO Reset Setting this bit will generate a reset pulse to reset TX FIFO which will empty the TX FIFO, i.e., the TX pointer will be reset to 0 after a reset signal. This bit returns to 0 automatically after the reset pulse is generated.

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



Bits	Field	Descriptions
[9]	CTSIE	CTS Clear-To-Send Interrupt Enable 0: Disable interrupt 1: Enable interrupt An interrupt will be generated when the CTSC bit is set in the USRSIFR register.
[8]	RXTOIE	Receive FIFO Time-Out Interrupt Enable 0: Disable interrupt 1: Enable interrupt Receive FIFO Time-Out Interrupt means that receive FIFO is not empty and no activities have occurred in the receive FIFO during the time-out duration specified by the RXTOC field.
[7]	RSADDIE	RS485 Address Detection Interrupt Enable 0: Disable interrupt 1: Enable interrupt An interrupt will be generated when the RSADD bit is set in the USRSIFR register.
[6]	BIE	Break Interrupt Enable 0: Disable interrupt 1: Enable interrupt An interrupt will be generated when the BII bit is set in the USRSIFR register.

Bits	Field	Descriptions
[5]	FEIE	Framing Error Interrupt Enable 0: Disable interrupt 1: Enable interrupt An interrupt will be generated when the FEI bit is set in the URSIFR register.
[4]	PEIE	Parity Error Interrupt Enable 0: Disable interrupt 1: Enable interrupt An interrupt will be generated when the PEI bit is set in the USRSIFR register.
[3]	OEIE	Overrun Error Interrupt Enable 0: Disable interrupt 1: Enable interrupt An interrupt will be generated when the OEI bit is set in the USRSIFR register.
[2]	TXCIE	Transmit Complete Interrupt Enable 0: Disable interrupt 1: Enable interrupt An interrupt will be generated when the TXC bit is set in the USRSIFR register.
[1]	TXDEIE	Transmit Data Empty Interrupt Enable 0: Disable interrupt 1: Enable interrupt An interrupt will be generated when the TXDE bit is set in the USRSIFR register.
[0]	RXDRIE	Receive Data Ready Interrupt Enable 0: Disable interrupt 1: Enable interrupt An interrupt will be generated when the RXDR bit is set in the USRSIFR register.

USART Status & Interrupt Flag Register – USRSIFR

This register contains the corresponding USART 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				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 is changed and an Interrupt will be generated if the CTSIE = 1 in the USRIER register. 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 RSADDIE = 1 in the USRIER register. 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 An interrupt will be generated if TXCIE=1 in the USRIER register. 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 less than threshold The TXDE bit will be set when transmit FIFO level is less than the transmit FIFO threshold level setting which is set by the TXTL field in the USRFCR register. This bit will be cleared when the new data is written into the USRDR register and the TX FIFO level is higher than threshold setting.

Bits	Field	Descriptions
[6]	RXTOF	Receive FIFO Time-Out Flag 0: RX FIFO Time-Out does not occur. 1: RX FIFO Time-Out occurs. This bit is clear when RX FIFO is empty.
[5]	RXDR	Receive FIFO Ready Flag 0: RX FIFO level is less than threshold 1: RX FIFO level is higher than threshold The RXDR bit will be set when the FIFO received data amount reaches the specified threshold level which is set 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.
[4]	BII	Break Interrupt Indicator 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.
[3]	FEI	Framing Error Indicator This bit will be set to 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 a logic 0. Writing 1 to this bit clears the flag.
[2]	PEI	Parity Error Indicator 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.
[1]	OEI	Overrun Error Indicator 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 is overwritten when an overrun event occurs, but the data in the RX shift register will not be transferred to the RX FIFO. The OEI bit is used to indicate the overrun event as soon as it happens. Writing 1 to this bit clears the flag.
[0]	RXDNE	RX FIFO Data Not Empty 0: RX FIFO is empty 1: RX FIFO contains at least 1 received data word

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 function enable control.

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	TG							
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	RXTOEN RXTOC							
Type/Reset	RW	0	RW	0	RW	0	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 will be reset and then starts to count. 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: Receive FIFO Time-Out Counter is disabled 1: Receive 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 will be 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	Reserved							
Type/Reset																
	23	22	21	20	19	18	17	16	Reserved							
Type/Reset																
	15	14	13	12	11	10	9	8	IrDAPSC							
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	Reserved		RXINV	TXINV	LB	TXSEL	IrDALP	IrDAEN
Type/Reset			RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[15:8]	IrDAPSC	IrDA Prescaler value This field contains the 8-bit debounce prescaler value. 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. 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 ...
[5]	RXINV	RX Signal Inverse Control 0: No inversion 1: RX input signal is inverted
[4]	TXINV	TX Signal Inverse Control 0: No inversion 1: TX output signal is inverted
[3]	LB	IrDA Loop Back Mode 0: Disable IrDA loop back mode 1: Enable IrDA loop back mode for self testing.
[2]	TXSEL	Transmit Select 0: Enable IrDA receiver 1: Enable IrDA transmitter
[1]	IrDALP	IrDA Low Power Mode Select the IrDA operation mode. 0: Normal mode 1: IrDA low power mode

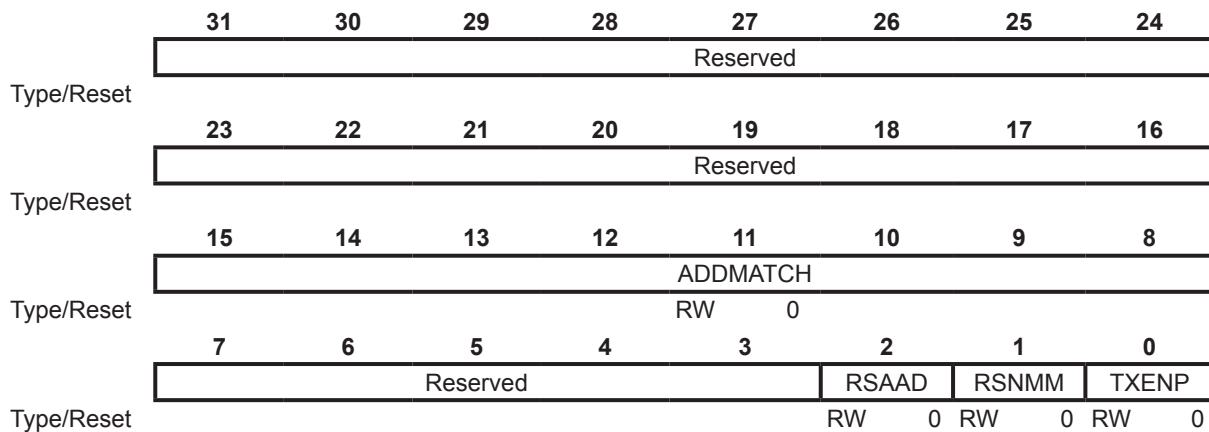
Bits	Field	Descriptions
[0]	IrDAEN	IrDA Enable control 0: Disable IrDA mode 1: Enable IrDA mode

USART RS485 Control Register – RS485CR

This register is used to control the RS485 mode of USART.

Offset: 0x01C

Reset value: 0x0000_0000



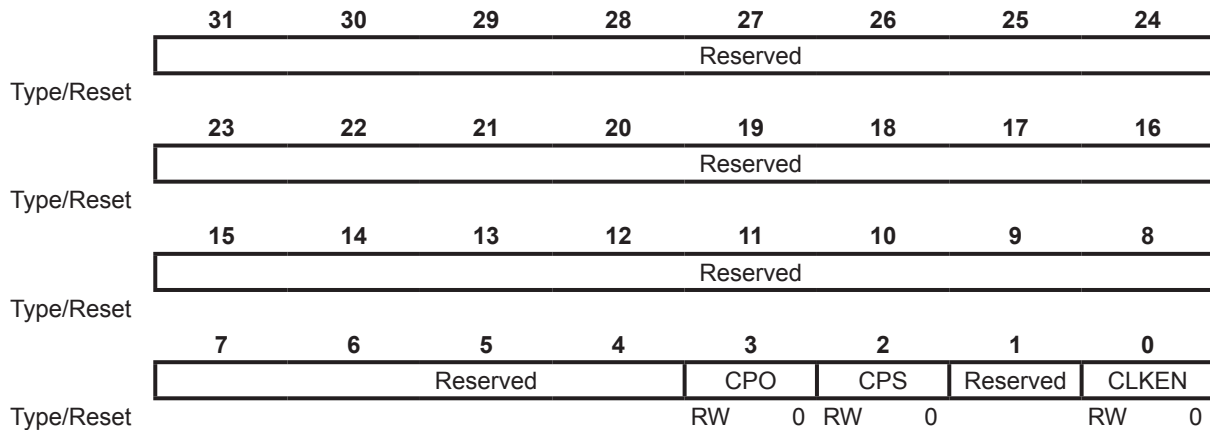
Bits	Field	Descriptions
[15:8]	ADDMATCH	RS485 Auto Address Match value The field contains the address match value for the RS485 auto address detection operation mode.
[2]	RSAAD	RS485 Auto Address Detection Operation Mode Control 0: Disable 1: Enable
[1]	RSNMM	RS485 Normal Multi-drop Operation Mode Control 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



Bits	Field	Descriptions
[3]	CPO	<p>Clock Polarity</p> <p>0: CTS/SCK pin idle state is low. 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. 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 disabled 1: CTS/SCK pin 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

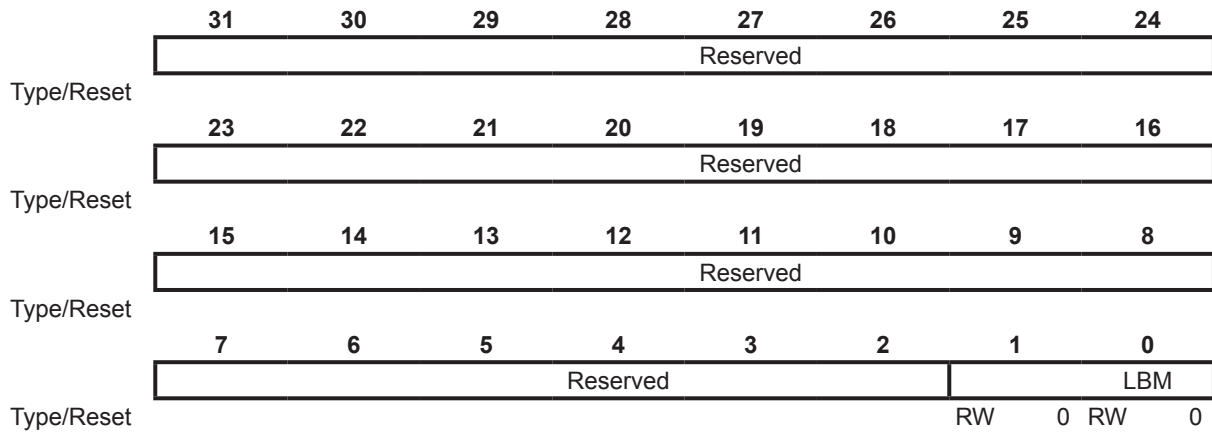
Bits	Field	Descriptions
[15:0]	BRD	Baud Rate Divider The 16 bits define the USART clock divider ratio. $\text{Baud Rate} = \text{CK_USART} / \text{BRD}$ Where the CK_USART clock is the clock connected to the USART module. BRD = 16 ~ 65535 for asynchronous mode BRD = 8 ~ 65535 for synchronous mode.

USART Test Register – USRTSTR

This register controls the USART debug mode.

Offset: 0x028

Reset value: 0x0000_0000



Bits	Field	Descriptions
[1:0]	LBM	Loopback Test Mode Select 00: Normal Operation 01: Reserved 10: Automatic Echo Mode 11: Loopback Mode

23 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 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 of the CK_APB (CK_UART) to produce a baud rate clock for the UART transmitter and receiver.

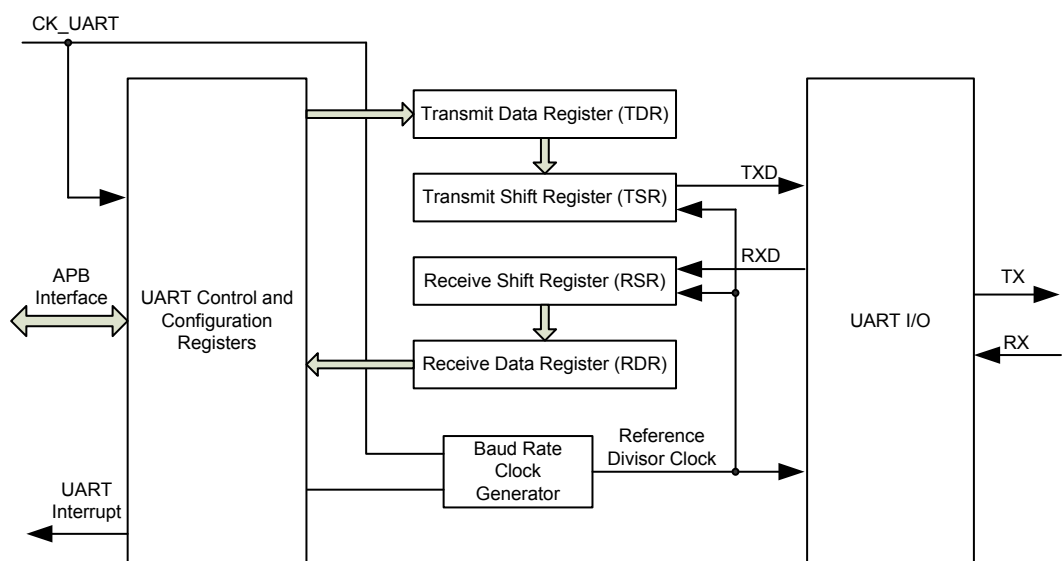


Figure 171. UART Block Diagram

Features

- Supports asynchronous serial communication modes
- Full Duplex Communication Capability
- Programming baud rate up to 3 Mbit/s.
- 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 bit generation
 - Bit order: LSB-first or MSB-first transfer
- Error detection: Parity, overrun, and frame error
- Supports PDMA Interface

Function 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 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. The both 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 to have failed and respond with a Framing Error.

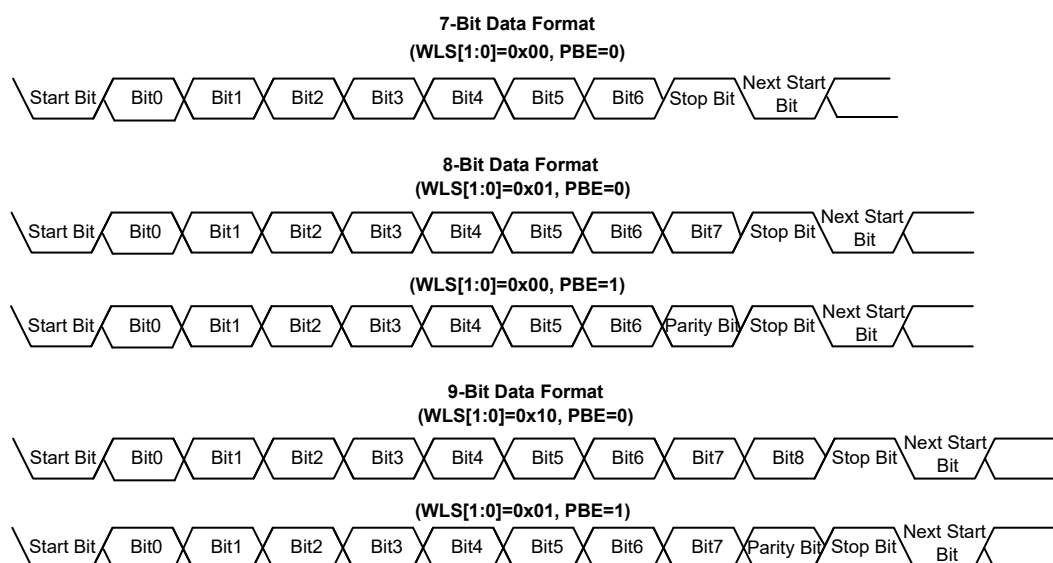


Figure 172. UART Serial Data Format

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 CK_UART clock is the APB clock connected to the UART while the BRD range is from 16 to 65535.

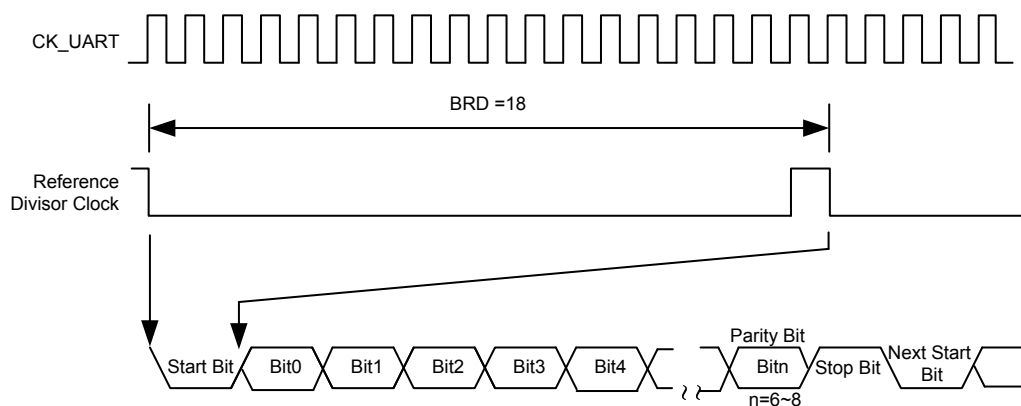


Figure 173. UART Clock CK_UART and Data Frame Timing

Table 56. Baud Rate Deviation Error Calculation – CK_UART = 40 MHz

Baud rate		CK_UART = 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	3076.9	13	2.56%

Table 57. Baud Rate Deviation Error Calculation – CK_UART = 48 MHz

Baud rate		CK_UART = 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	16	0.00%

Interrupts and Status

The UART can generate interrupts when the following event occurs and corresponding interrupt enable bits are set:

- Receiver line status interrupts: The interrupts are generated when the overrun error, parity error, framing error or break event occurs for the UART receiver.
- 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 RDR 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 on the PDMA configurations, refer to the PDMA chapter.

Register Map

The following table shows the UART registers and reset values.

Table 58. 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	
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	
									RW 0
	7	6	5	4	3	2	1	0	
Type/Reset	DB								
	RW	0	RW	0	RW	0	RW	0	RW
									0

Bits	Field	Descriptions
[8:0]	DB	By reading this register, the UART will return a 7, 8 and 9-bit received data. The DB field bit 8 is valid for 9-bit mode only and is fixed at 0 for the 8-bit mode. For the 7-bits mode, the DB[6:0] contains the available bits. 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] contains the available bits.

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					
Reserved												
Type/Reset												
23	22	21	20	19	18	17	16					
Reserved												
Type/Reset												
15	14	13	12	11	10	9	8					
Reserved	BCB	SPE	EPE	PBE	NSB	WLS						
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0
7	6	5	4	3	2	1	0					
RXDMAEN	TXDMAEN	URRXEN	URTXEN	Reserved	TRSM	Reserved						
Type/Reset	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 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 PBE 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 bit" is generated when 8- and 9-bit word length is selected.

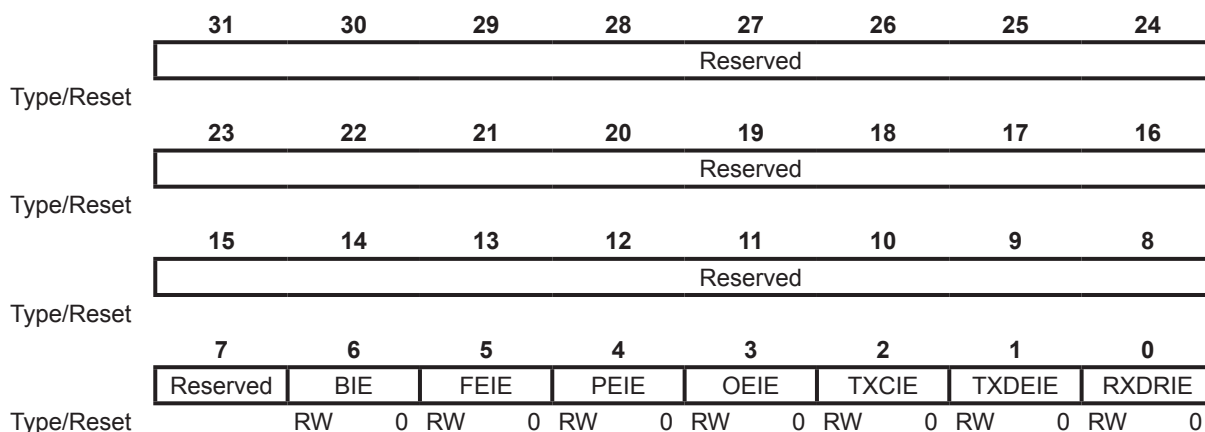
Bits	Field	Descriptions
[9:8]	WLS	Word Length Select 00: 7 bits 01: 8 bits 10: 9 bits 11: Reserved
[7]	RXDMAEN	UART RX DMA Enable 0: Disabled 1: Enabled
[6]	TXDMAEN	UART TX DMA Enable 0: Disabled 1: Enabled
[5]	URRXEN	UART RX Enable 0: Disable 1: Enable
[4]	URTXEN	UART TX Enable 0: Disable 1: Enable
[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



Bits	Field	Descriptions
[6]	BIE	Break Interrupt Enable 0: Disable interrupt 1: Enable interrupt An interrupt is generated when the break interrupt is enabled and the BII bit is set in the URSIFR register.
[5]	FEIE	Framing Error Interrupt Enable 0: Disable interrupt 1: Enable interrupt An interrupt is generated when the framing error interrupt is enabled and the FEI bit is set in the URSIFR register.
[4]	PEIE	Parity Error Interrupt Enable 0: Disable interrupt 1: Enable interrupt An interrupt is generated when the parity error interrupt is enabled and the PEI bit is set in the URSIFR register.
[3]	OEIE	Overrun Error Interrupt Enable 0: Disable interrupt 1: Enable interrupt An interrupt is generated when the overrun error interrupt is enabled and the OEI bit is set in the URSIFR register.
[2]	TXCIE	Transmit Complete Interrupt Enable 0: Disable interrupt 1: Enable interrupt An interrupt is generated when the transmit complete interrupt is enabled and the TXC bit is set in the URSIFR register.

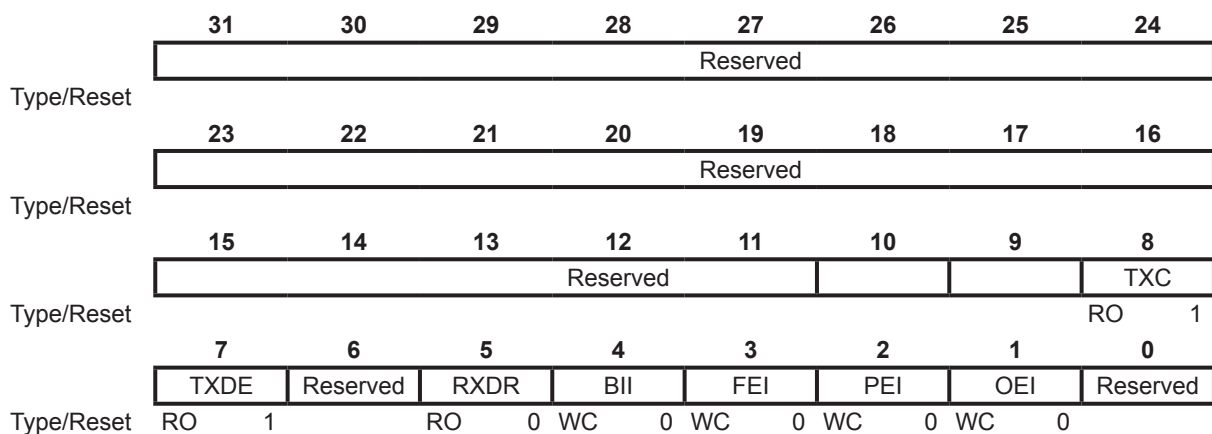
Bits	Field	Descriptions
[1]	TXDEIE	Transmit Data Register Empty Interrupt Enable 0: Disable interrupt 1: Enable interrupt An interrupt is generated when the transmit data register empty interrupt is enabled and the TXDE bit is set in the URSIFR register.
[0]	RXDRIE	Receive Data Ready Interrupt Enable 0: Disable interrupt 1: Enable interrupt An interrupt is generated when the receive data ready interrupt is enabled and the RXDR bit is set in the URSIFR register.

UART Status & Interrupt Flag Register – URSIFR

This register contains the corresponding UART status.

Offset: 0x010

Reset value: 0x0000_0180



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. An interrupt is generated if TXCIE=1 in the URIER register. 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 is generated if TXEIE=1 in the URIER register. This bit is cleared by a write to the URDR register with a new data.

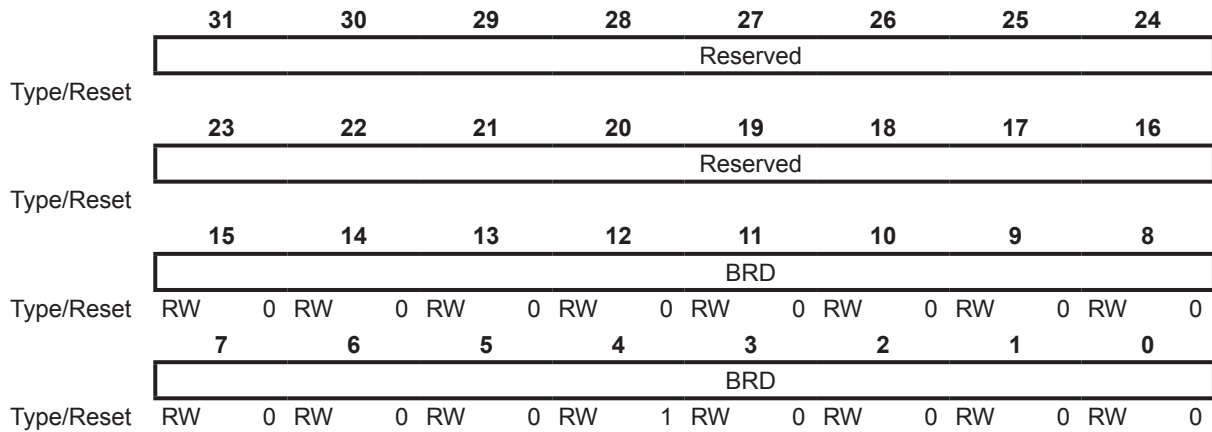
Bits	Field	Descriptions
[5]	RXDR	<p>RX Data Ready</p> <p>0: Receive data register is empty 1: Received data in the receive data register is ready to read.</p> <p>This bit is set by hardware when the content of the receive shift register RDR has been transferred to the URDR register. It is cleared by a read to the URDR register. An interrupt is generated if RXDRIE=1 in the URIER register.</p>
[4]	BII	<p>Break Interrupt Indicator</p> <p>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.</p>
[3]	FEI	<p>Framing Error Indicator</p> <p>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.</p>
[2]	PEI	<p>Parity Error Indicator</p> <p>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.</p>
[1]	OEI	<p>Overrun Error Indicator</p> <p>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 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 the overrun event as soon as it happens. Writing 1 to this bit clears the flag.</p>

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



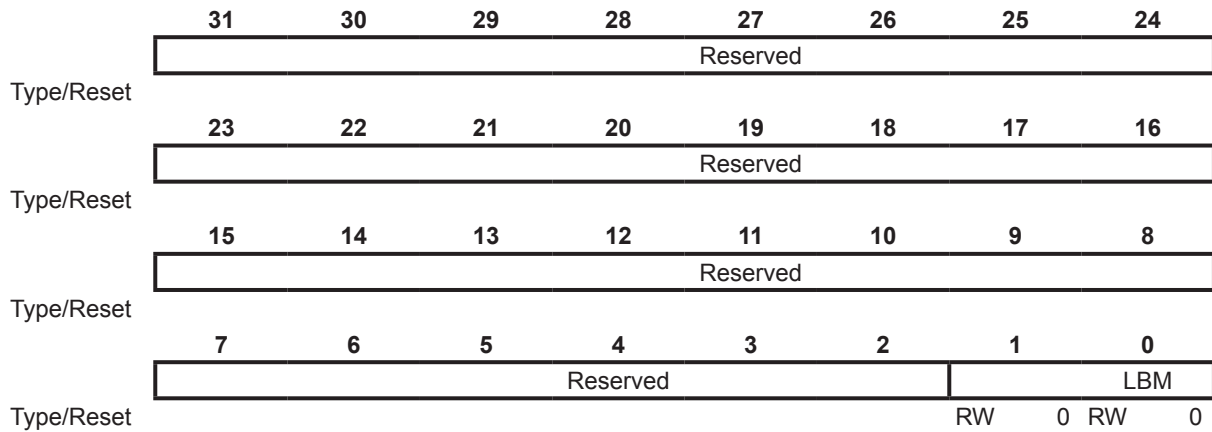
Bits	Field	Descriptions
[15:0]	BRD	Baud Rate Divider The 16 bits define the UART clock divider ratio. $Baud\ Rate = CK_UART / BRD$ Where the CK_UART clock is the clock connected to the UART module. BRD = 16 ~ 65535 for UART mode

UART Test Register – URTSTR

This register controls the UART debug mode.

Offset: 0x028

Reset value: 0x0000_0000



Bits	Field	Descriptions
[1:0]	LBM	Loopback Test Mode Select 00: Normal Operation 01: Reserved 10: Automatic Echo Mode 11: Loopback Mode

24 Smart Card Interface (SCI)

Introduction

The Smart Card Interface, SCI, is compatible with the ISO 7816-3 standard. This interface includes functions for card Insertion/Removal detection, SCI data transfer control logic and data buffers, internal Timer Counters and corresponding control logic circuits to perform the required Smart Card operations. The Smart Card interface acts as a Smart Card Reader to facilitate communication with the external Smart Card. The overall functions of the Smart Card interface are controlled by a series of registers including control and status registers together with several corresponding interrupts which are generated to get the attention of the microcontroller for SCI transfer status.

As the complexity of ISO7816-3 standard data protocol does not permit comprehensive specifications to be provided in this datasheet, the reader should therefore consult other external information for a detailed understanding of this standard.

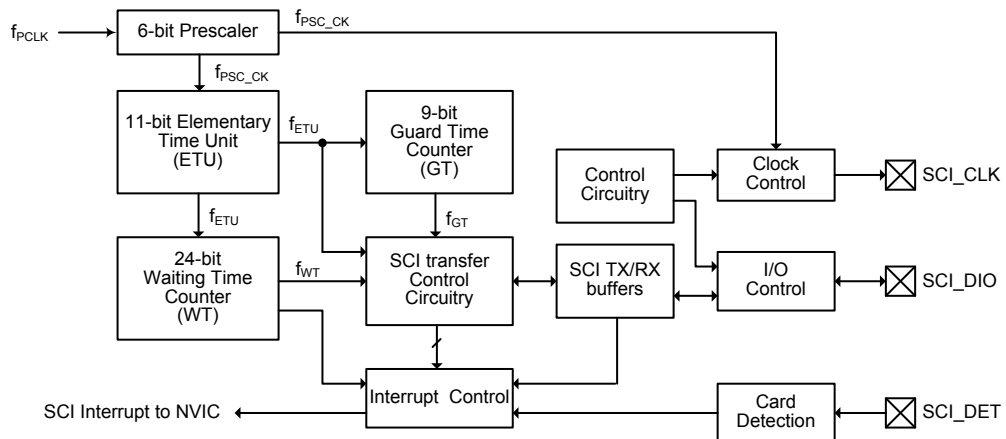


Figure 174. SCI Block Diagram

Features

- Supports ISO 7816-3 standard
- Character Transfer Mode
- 1 transmit buffer and 1 receive buffer
- 11-bit ETU (elementary time unit) counter
- 9-bit guard time counter
- 24-bit general purpose waiting time counter
- Parity generation and checking
- Automatic character repetition on parity error detection in transmission and reception modes
- Supports PDMA access at a transmission or reception completion

Functional Descriptions

To communicate with an external Smart Card, the integrated Smart Card Interface has a series of external pins known as SCI_CLK, SCI_DIO and SCI_DET. The SCI_CLK pin is the clock output signal used to communicate with the external Smart Card together with the serial data pin named SCI_DIO. The operation of the SCI_CLK and SCI_DIO pins can be selected to be the SCI data Transfer Mode which is driven automatically by the SCI control circuits or to be the Manual mode which is controlled by configuring the internal CLK and DIO register bits respectively by the application program. The SCI_DET pin is the external card detection input pin. Insertion or removal of the external Smart Card can be automatically detected and generate an interrupt signal which is sent to the microcontroller if the corresponding interrupt function is enabled.

For proper data transfer, some timing related procedures must be executed before the Smart Card Interface can begin to communicate with the external card. There are three counters named Elementary Time Unit, ETU, Guard Time Counter, GT, and Waiting Time Counter, WT, which are used for the timing related functions in Smart Card Interface data transfer operations.

Elementary Time Unit Counter

The Elementary Time Unit, ETU, is an 11-bit up-counting counter which generates a clock denoted as f_{ETU} to be used as the operating frequency source for the SCI data transmission and reception. The clock source of the ETU comes from the Smart Card clock, named f_{PSC_CK} , which is derived from the 6-bit prescaler. The data transfer of the SCI is a character frame based protocol, which basically consists of a Start bit, 8-bits of data and a Parity bit. The time period, t_{ETU} ($1/f_{ETU}$), generated by the ETU, is the time unit for a character bit. There is a register related to the Elementary Time Unit known as the ETUR register which stores the expected contents of the ETU. Each time the ETUR register is written, the ETU circuitry will reload the new written value and restart counting. The elementary time unit t_{ETU} is obtained from the following formula which defines the bit rate in the ISO 7816-3 standard specification.

$$1\text{etu} = t_{ETU} = \frac{E_i}{D_i} \times \frac{1}{f}$$

where:

- t_{etu} is the nominal duration of the data bit on the signal SCI_DIO provided to the card by the interface
- D_i is the bit-rate adjustment factor
- F_i is the clock rate conversion factor
- f is the frequency value of the clock signal SCI_CLK provided to the card by the interface

D_i is an encoded decimal value based on a 4-bit field, named DI, as represented in the accompanying table.

Table 59. DI Field Based D_i Encoded Decimal Values

DI field	0001	0010	0011	0100	0101	0110	1000	1001
D_i (decimal)	1	2	4	8	16	32	12	20

F_i is an encoded decimal value based on a 4-bit field, named FI, as represented in the following table.

Table 60. FI Field Based F_i Encoded Decimal Values

FI field	0000	0001	0010	0011	0100	0101	0110	1001	1010	1011	1100	1101
F_i (decimal)	372	372	558	744	1116	1488	1860	512	768	1024	1536	2048

The values of FI and DI, as they appear in the preceding tables, will be obtained from the Answer-to-Reset packet sent from the external Smart Card to the Smart Card Interface the first time the external Smart Card is inserted. When the SCI receives the FI and DI information, the F_i and D_i values can be obtained by looking up the preceding two tables. After the F_i and D_i values are obtained, the value which should be written into the ETUR register can be calculated by F_i/D_i . The following table shows the possible ETU values obtained by the F_i/D_i ratio.

Table 61. Possible ETU Values Obtained with the F_i/D_i Ratio

F_i/D_i	372	558	774	1116	1488	1860	512	768	1024	1536	2048
1	372	558	744	1116	1488	1860	512	768	1024	1536	2048
2	186	279	372	558	744	930	256	384	512	768	1024
4	93	139.5	186	279	372	465	128	192	256	384	512
8	46.5	69.75	93	139.5	186	232.5	64	96	128	192	256
16	23.25	34.87	46.5	69.75	93	116.2	32	48	64	96	128
32	22.62	17.43	23.25	34.87	46.5	58.13	16	24	32	48	64
12	31	46.5	62	93	124	155	42.66	64	85.33	128	170.6
20	18.6	27.9	37.2	55.8	74.4	93	25.6	38.4	51.2	76.8	102.4

Compensation mode

As the value of the ETUR register is obtained by the above procedure, the calculation results of the value may not be an integer. If the calculation result is not an integer and is less than the integer n but greater than the integer $(n-1)$, either the integer n or $(n-1)$ should be written into the ETUR register depending upon whether the result is closer to integer n or $(n-1)$. The integer n mentioned here is a decimal.

If the calculation result is close to the value of $(n-0.5)$, the compensation mode should be enabled by setting the compensation enable control bit, COMP, in the ETUR register to 1 for successful data transfer. When the result is close to the value of $(n-0.5)$ and the compensation mode is enabled, the value written into the ETUR register should be n . The ETU circuitry will then generate the time unit sequence with n clock cycles and next $(n-1)$ clock cycles alternately and so on. This results in an average time unit of $(n-0.5)$ clock cycles and allows a time granularity down to a half clock cycle. Note that the ETU will reload the ETUR register value and restart counting at the time when the Start bit appears in the SCI data Transfer Mode.

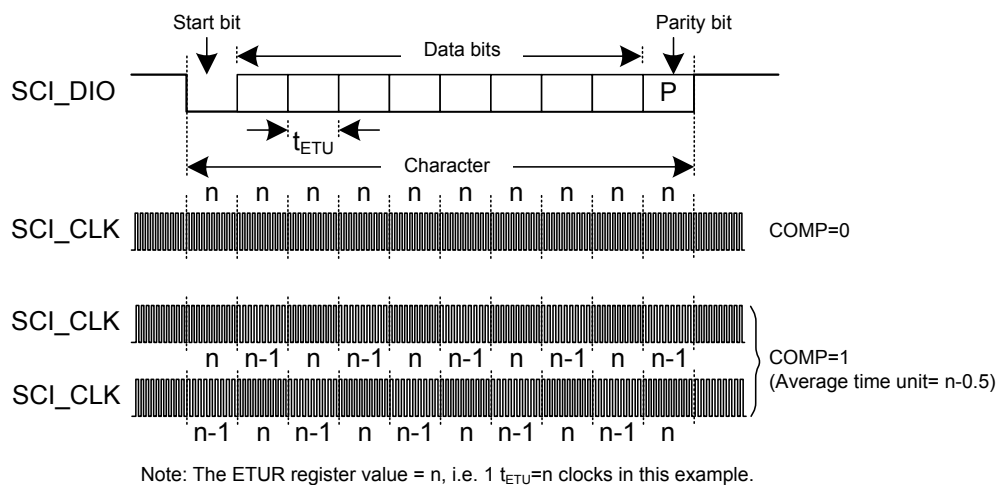


Figure 175. Character Frame and Compensation Mode

Guard Time Counter

The Guard Time Counter, GT, is a 9-bit up-counting counter which generates a minimum time duration known as a character frame, denoted as t_{GT} , between the leading edges of two consecutive characters in the SCI data transfer. The clock source of the guard time counter comes from the ETU, named f_{ETU} in the block diagram. There is a register related to the guard time counter known as the GTR register, which stores the expected value of the guard time counter. The guard time value will be reloaded at the end of the current guard time period. Note that the guard time between the last character received from the Smart Card and the next character transmitted by the SCI circuitry which should be properly managed by the application program. There is no guard time insertion when the first character is transmitted.

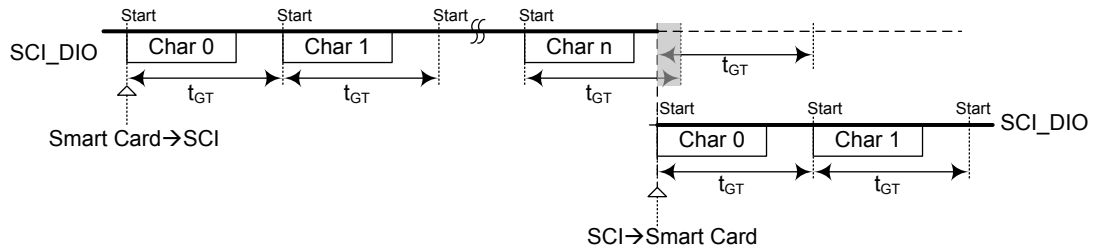


Figure 176. Guard Time Duration

Waiting Time Counter

The Waiting Time counter, WT, is a 24-bit down counting counter which generates a maximum time duration, denoted as t_{WT} , for data transfer. The clock source of the waiting time counter comes from the ETU and is named f_{ETU} .

There is a register for the waiting time counter known as the WTR register which stores the expected waiting time counter value. The waiting time counter can be used in both the SCI data Transfer Mode and manual mode and can reload the value for specific conditions. The function of the waiting time counter is controlled by the WTEN bit in the CR register. When the SCI is configured to be operated in the SCI data Transfer Mode and the waiting time counter is enabled by setting the WTEN bit to 1, the updated WTR register value will be loaded into the waiting time counter when the Start bit is detected. Note that the WTEN bit should not be set to 1 to enable the waiting time counter in the SCI data Transfer Mode until after the external Smart Card is inserted.

If the SCI is configured to operate in the manual mode, the waiting time counter can be used as a general purpose timer and this timer is enabled or disabled by setting or clearing the WTEN bit. The updated WTR register value will not be loaded into the waiting time counter if the waiting time counter is enabled. When the waiting time counter is disabled by setting the WTEN bit to 0 and an updated value is written into the WTR register, the new value will immediately be loaded into the waiting time counter and then the counter will start to count after the WTEN bit is again set to 1.

Software can change the Waiting Time value on-the-fly. For example, in T=1 mode, the value of the Block Waiting Time, t_{BWT} , should be written into the WTR register before the Start bit of the last transmitted character occurs. After the transmission of the last character is completed, software should write the Character Waiting Time value, t_{CWT} , into the WTR register.

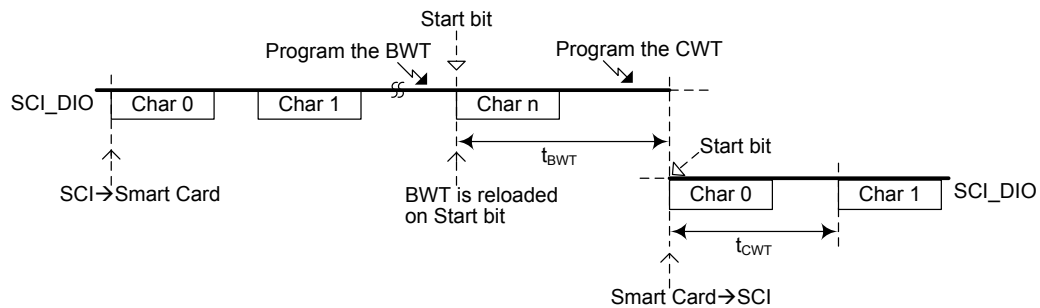


Figure 177. Character and Block Waiting Time Duration – CWT and BWT

Card Clock and Data Selection

The SCI communicates with an external Smart Card using a series of external pins. These are the serial data pin, SCI_DIO, output clock pin, SCI_CLK, and the Card Detection input pin, SCI_DET.

The SCI serial data pin, named SCI_DIO, can be controlled by the SCI hardware circuitry or the software control bits depending upon whether the SCI is operated in the SCI Transfer Mode or in the Manual Mode. The mode selection is determined by the SCIM bit in the CR register. The SCI_DIO pin status is controlled by the CDIO bit in the CCR register when the SCI is configured to operate in the Manual mode by clearing the SCIM bit in the CR register. In the Manual Mode the SCI_DIO pin status is a copy of the CDIO bit. However, when the SCI is configured to operate in the SCI Transfer Mode, the SCI_DIO pin status is determined by the SCI transfer circuitry.

The SCI clock output pin named SCI_CLK can be controlled by the 6-bit SCI prescaler or the software control bits depending upon the condition of the CLKSEL bit in the CCR register. The SCI_CLK pin status is controlled by the CCLK bit in the CCR register when the CLKSEL bit is cleared to 0. The SCI_CLK pin status is a copy of the CCLK bit. However, when the CLKSEL bit is set to 1, the SCI_CLK signal is sourced from the 6-bit prescaler output. The prescaler division ratio is determined by the PSC field in the PSCR register.

Card Detection

When an external Smart Card is inserted, the internal card detector can detect this insertion operation and generate a card insertion interrupt if the corresponding interrupt enable control bit, CARDIRE, in the IER register is set to 1. Similarly, if the card is removed, the internal card detector can also detect the removal and consequently generate a card removal interrupt when the corresponding interrupt function is enabled by setting the control bit, CARDIRE, in the IER register, to 1.

The card detector can support two kinds of card detect switch mechanisms. One is a normally open switch mechanism when the card is not present and the other is a normally closed switch mechanism. After noting which card detect switch mechanism type is used, the card switch selection should be configured by setting the selection bit, DETCNF, in the CR register to correctly detect the card presence. No matter what type of the card switch is selected, by configuring the DETCNF bit, the card Insertion/Removal flag, CPREF, in the SR register will be set to 1 when the card is actually present on the SCI_DET pin. Note that there are no hardware de-bounce circuits in the card detector. Any change of the SCI_DET pin level will cause the CPREF bit to change. The required de-bounce time should be handled by the application program.

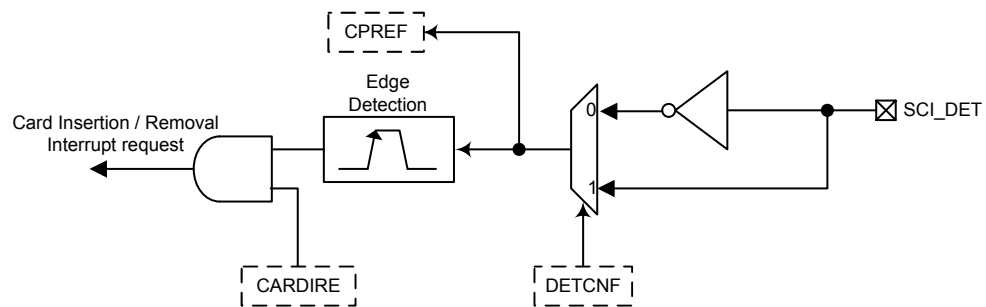


Figure 178. SCI Card Detection Diagram

SCI Data Transfer Mode

The SCI data transfer with the external Smart Card is implemented with two operating modes. One is the SCI mode while the other is the Manual Mode. The data Transfer Mode is selected by the SCI mode selection bit, SCIM, in the CR register. When the SCIM bit is set to 1, the SCI mode is enabled and data will automatically be transferred by the SCI transfer circuitry. Otherwise, data transfer operates in the Manual Mode if the SCIM bit is cleared to 0. The SCI transfer interface is a half-duplex interface and communicates with the external Smart Card via the SCI_CLK and SCI_DIO pins. After a reset condition the SCI transfer interface is in the reception mode but the SCI transfer operation is disabled. When the SCI mode is enabled, data transfer is driven by the SCI transfer circuitry automatically through the SCI_CLK and SCI_DIO pins.

There are two data registers related to data transmission and reception, TXB and RXB, which store the data to be transmitted and received respectively. If a character is written into the TXB register in the SCI Transfer Mode, the SCI transfer interface will automatically switch to the Transmission Mode from the reception mode after a reset. When the SCI transmission or reception has finished, the corresponding request flag, named TXCF or RXCF, in the SR register is set to 1. If the transmit buffer is empty, the transmit buffer empty flag, TXBEF, in the SR register will be set to 1.

Parity Check Function

The SCI transfer interface supports a parity generator and a parity check function. As the parity error occurs during a data transfer, the corresponding request flag, named PARF in the SR register, will be set to 1. Once the PARF bit is set to 1, the parity error pending flag, PARP, in the IPR register will also be set to 1 if the relevant interrupt control bit, PARE, in the IER register is enabled.

If the data transmitted by the SCI is received by the external Smart Card without a parity error, the SCI transmission request flag, TXCF, will be set to 1 and the SCI parity error request flag, PARF, will be cleared to 0. If the data transmitted by the external Smart Card is received by the SCI without a parity error, the SCI reception request flag, RXCF, will be set to 1 and the parity error flag, PARF, will be cleared to 0.

Repetition Function

There is a Character Repetition function supported by the SCI transfer circuitry when a parity error occurs. The Character Repetition function is enabled by setting the CREP bit in the CR register to 1. A repetition function will then be activated when a parity error occurs during a data transfer. The repetition time number can be selected to be 4 or 5 by configuring the RETRY bit in the CR register.

When the CREP bit is set to 1, the character repetition function will be activated. Taking a 4 time repetition as an example, when the CREP bit is set to 1 and the RETRY bit is set to 1, in the Transmission Mode, the SCI will repeatedly transmit the data a maximum of 4 times when an error signal occurs. However, if the SCI is informed that there is still an error signal during the 4 transmissions, the parity error flag PARF will be set to 1 after the same data has been transmitted 4 times but the TXCF flag will not be set. At this time the data in the transmit buffer will be loaded into the transmit shift register and the transmit buffer will be empty which will result in the TXBEF flag being set to 1.

Similarly, when the SCI operates in the reception mode, it will inform the external Smart Card that there is a parity error for a maximum of 4 times if the character repetition function is enabled. If the SCI informs the external Smart Card that there is still an error signal for the 4 receptions, the parity error flag, PARF, will be set to 1 together with the reception request flag, RXCF.

If the CREP bit is cleared to 0, the character repetition function will be disabled. When the SCI operates in the reception mode, both the PARF and RXCF bits will be set to 1 as data with a parity error has been received. If the SCI is informed that there is a parity error in the Transmission Mode, the PARF bit will be set to 1 but the TXCF bit will not be set.

Manual Data Transfer Mode

When the SCIM bit is cleared to 0, data will be transferred in the Manual Mode. In the Manual Mode, the data is controlled by the control bit, CDIO, in the CCR register. The CDIO bit value will be reflected immediately on the SCI_DIO pin in the Manual Mode. Note that in the Manual Mode the character repetition function can not be used as well as the related flags and all the data transfer is handled by the application program. The clock used to drive the external Smart Card that appears on the SCI_CLK pin can be derived from the internal clock source, which is the 6-bit prescaler output, f_{PSC_CK} , or from the control bit, CCLK, in the CCR register. The clock source is selected using the bit, CLKSEL, in the CCR register. When CLKSEL bit is set to 1, the clock used to drive the Smart Card will be sourced from the 6-bit prescaler output, f_{PSC_CK} . If the clock is to be managed manually, the CLKSEL bit should first be cleared to 0 and then the value of the CCLK bit will be present in the SCI_CLK pin.

Data Transfer Direction Convention

If the direction convention used by the Smart Card is the same as the convention used by the SCI, the SCI will generate a reception interrupt if the reception interrupt is enabled without a parity error flag. Otherwise, the SCI will generate a reception interrupt and the parity error flag will be asserted. By checking the parity error flag, the SCI can know if the data direction convention is correct or not.

Interrupt Generator

There are several conditions for the SCI to generate an SCI interrupt. When these conditions are met, an interrupt signal will be generated to obtain the attention of the microcontroller. These conditions are a Smart Card Insertion/Removal, a Waiting Time Counter Underflow, a Parity error, an end of a Character Transmission or Reception and an empty Transmit buffer. When a Smart Card interrupt is generated by any of these conditions, then if the SCI global interrupt and the corresponding SCI interrupt are together enabled, the program will jump to the corresponding interrupt vector where it can be serviced before returning to the main program.

For SCI interrupt events, there are corresponding pending flags which can be masked by the relevant interrupt enable control bit. When the related interrupt enable control is disabled, the corresponding interrupt pending flag will not be affected by the request flag and no interrupt will be generated. If the related interrupt enable control is enabled, the relevant interrupt pending flag will be affected by the request flag and then the interrupt will be generated. The pending flag register, named IPR, is read only and once the pending flag is read by the application program, it will be automatically cleared while the related request flag should be cleared by the application program manually.

For an SCI Interrupt to be serviced, in addition to the bits for the corresponding interrupt enable control in the SCI being set, the SCI global interrupt enable control bit in the NVIC must also be set. If this SCI global interrupt control bit is not set, then no SCI interrupt will be serviced.

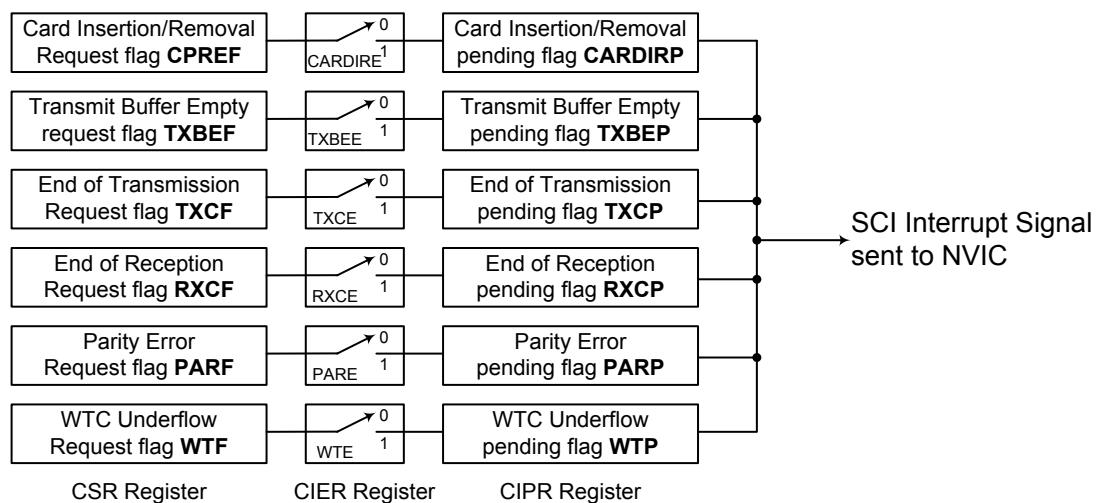


Figure 179. SCI Interrupt Structure

PDMA Interface

The PDMA interface is integrated in the SCI module. The PDMA function can be enabled by setting the TXDMA or RXDMA bit to 1 in the transmitter or receiver mode respectively. When the transmit buffer is empty which results in the transmit buffer empty flag, TXBEF, being asserted and the TXDMA bit is set to 1, the PDMA function will be activated to move data from a certain memory location into the SCI Transmit buffer. Similarly, when the SCI receives a character which results in the character received flag, RXCF, being asserted and the RXDMA bit is set to 1, the PDMA function will be activated to move data from the SCI Receive buffer to a specific memory location.

For a more detailed descriptions on the PDMA configurations, refer to the PDMA chapter.

Register Map

There are several registers associated with the Smart Card function. Some of these registers control the SCI overall function as well as the interrupts, while some of the registers contain the status bits which indicate the Smart Card data transfer situation and error conditions. Also there are two registers for the SCI transmission and reception respectively to store the data received from or to be transmitted to the external Smart Card. The following table shows the SCI register list and reset values.

Table 62. SCI Register Map

Register	Offset	Description	Reset Value
CR	0x000	SCI Control Register	0x0000_0000
SR	0x004	SCI Status Register	0x0000_0080
CCR	0x008	SCI Contact Control Register	0x0000_0008
ETUR	0x00C	SCI Elementary Time Unit Register	0x0000_0174
GTR	0x010	SCI Guard Time Register	0x0000_000C
WTR	0x014	SCI Waiting Time Register	0x0000_2580
IER	0x018	SCI Interrupt Enable Register	0x0000_0000
IPR	0x01C	SCI Interrupt Pending Register	0x0000_0000
TXB	0x020	SCI Transmit Buffer	0x0000_0000
RXB	0x024	SCI Receive Buffer	0x0000_0000
PSCR	0x028	SCI Prescaler Register	0x0000_0000

Register Descriptions

SCI Control Register – CR

This register contains the SCI 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						RXDMA	TXDMA	
							RW 0	RW 0	
	7	6	5	4	3	2	1	0	
Type/Reset	Reserved	DETCNF	ENSCI	RETRY	SCIM	WTEN	CREP	CONV	
		RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	

Bits	Field	Descriptions															
[9]	RXDMA	SCI reception DMA request enable 0: SCI reception DMA request is disabled 1: SCI reception DMA request is enabled															
[8]	TXDMA	SCI transmission DMA request enable control 0: SCI transmission DMA request is disabled 1: SCI transmission DMA request is enabled															
[6]	DETCNF	Card switch type selection 0: Switch is normally opened if no card is present 1: Switch is normally closed if no card is present															
		<table border="1"> <thead> <tr> <th>DETCNF</th> <th>SCI_DET pin</th> <th>STATUS</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> <td>No card insert</td> </tr> <tr> <td>0</td> <td>0</td> <td>Card insert</td> </tr> <tr> <td>1</td> <td>1</td> <td>Card insert</td> </tr> <tr> <td>1</td> <td>0</td> <td>No card insert</td> </tr> </tbody> </table>	DETCNF	SCI_DET pin	STATUS	0	1	No card insert	0	0	Card insert	1	1	Card insert	1	0	No card insert
DETCNF	SCI_DET pin	STATUS															
0	1	No card insert															
0	0	Card insert															
1	1	Card insert															
1	0	No card insert															
		This bit is set and cleared by the application program to configure the card detector switch type.															
[5]	ENSCI	SCI finite state machine enable bit 0: SCI FSM is disabled and forced to its initial state 1: SCI FSM is enabled															
[4]	RETRY	Character transfer repetition time selection for a parity error condition 0: Data transfer 5 times when parity error occurs 1: Data transfer 4 times when parity error occurs The bit is available only when the CREP bit is set to 1. When this bit is set to 1, the data will be transmitted or received 4 times once a parity error occurs. If the bit is cleared to 0, the data will be transferred 5 times if a parity error occurs.															

Bits	Field	Descriptions
[3]	SCIM	<p>SCI Mode Selection</p> <p>0: SCI data transfer in manual mode 1: SCI data transfer in SCI mode</p> <p>This bit is set and cleared by the application program to select the SCI data Transfer Mode. If it is cleared to 0, the SCI_DIO pin status is the same as the value of the CDIO bit in the CCR register. If it is set to 1, the SCI_DIO pin is driven by the internal SCI control circuitry. Before the data transfer type is switched from the Manual Mode to the SCI Mode, the CDIO bit must be set to 1 to avoid an SCI malfunction.</p>
[2]	WTEN	<p>Waiting Time Counter enable control</p> <p>0: Waiting Time Counter stops counting 1: Waiting Time Counter starts counting</p> <p>The WTEN bit is set and cleared by the application program. When the WTEN bit is cleared to 0, a write access to the WTR register will load the value into the waiting time counter. If it is set to 1, the waiting time counter is enabled and automatically reloaded with the value at each start bit occurrence.</p>
[1]	CREP	<p>Automatic character repetition enable control for a parity error condition</p> <p>0: No retry on parity error 1: Automatic retry on parity error</p> <p>The CREP bit is set and cleared by the application program. When the CREP bit is cleared to 0, both the RXCF and PARF flags will be set when a parity error occurs in the reception mode after the data is received. However, in the Transmission Mode, the PARF flag will be set but the TXCF flag will not be set when a parity error occurs. If the CREP bit is set to 1, a character transfer will automatically be activated 4 or 5 times depending upon the RETRY bit value. In the Transmission Mode the character will be re-transmitted if the transmitted data has a parity error. Here the parity error flag, PARF, will be set at the end of the 4th or 5th transmission without the TXCF bit being set. In the reception mode if the received data has a parity error, the SCI will inform the external Smart Card for 4 or 5 times and then the PARF and RXCF flags will both be set at the end of the 4th or 5th reception.</p>
[0]	CONV	<p>Data direction convention select</p> <p>0: LSB is transferred first; a data "1" is a logic high level on the SCI_DIO pin and the parity bit is added after the MSB. 1: MSB is transferred first; a data "1" is a logic low level on the SCI_DIO pin and the parity bit is added after the LSB.</p> <p>This bit is set and cleared by the application program to select if the data is transmitted LSB or MSB first. When the data direction convention is the same as the data direction specified by the external Smart Card, only the RXCF flag will be set to 1 without a parity error. Otherwise, both the RXCF and PARF flags will be set to 1 after the data is received.</p>

SCI Status Register – SR

This register contains the SCI status bits.

Offset: 0x004

Reset value: 0x0000_0080

	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	TXBEF	CPREF	reserved		WTF	TXCF	RXCF	PARF	
	RO	1 RO	0		RO	0 W0C	0 RO	0 W0C	0

Bits	Field	Descriptions
[7]	TXBEF	<p>Transmit Buffer Empty Request Flag</p> <p>0: Transmit buffer is not empty 1: Transmit buffer is empty</p> <p>This bit is used to indicate if the transmit buffer is empty and is set or cleared by hardware automatically.</p>
[6]	CPREF	<p>Card Presence Request Flag</p> <p>0: No card is present 1: A card is present</p> <p>This bit is used to indicate if a card is present and is set or cleared by hardware automatically. The card presence detection function is enabled after the ENSCI bit is set.</p>
[3]	WTF	<p>Waiting Time Counter Underflow Request Flag</p> <p>0: No Waiting Time Counter underflow 1: The Waiting Time Counter underflows</p> <p>This bit is set and cleared by the application program and indicates if the Waiting Time Counter underflows.</p>
[2]	TXCF	<p>Character Transmission Request Flag.</p> <p>0: No character transmitted 1: A character has been transmitted</p> <p>This bit is set by hardware and cleared by writing a “0” into it.</p>
[1]	RXCF	<p>Character Received Request Flag.</p> <p>0: No character received 1: A character has been received</p> <p>This bit is set by hardware and cleared after a read access to the RXB register by the application program. The RXCF bit will be set to 1 when a character is received regardless of the result of the parity check.</p> <p>When the character has been received, the received data stored in the RXB register should be moved to the data memory as specified by the application program. If the contents of the RXB register are not read before the end of the next character to be shifted in, the data stored in the RXB register will be overwritten.</p>

Bits	Field	Descriptions
[0]	PARF	<p>Parity Error Request Flag.</p> <ul style="list-style-type: none">0: No parity error occurs1: Parity error has occurred <p>This bit is set by hardware and cleared by writing a "0" into it. When a character is received, the parity check circuitry will check that the parity is correct or not. If the result of the parity check is not correct, the parity error request flag, PARF, will be set to 1. Otherwise, the PARF bit will remain zero. In the Transmission Mode when the SCI is informed that there is a parity error in the transmitted character by the external Smart Card, the PARF bit will also be set to 1.</p>

SCI Contact Control Register – CCR

This register specifies the SCI pin setting and clock selection.

Offset: 0x008

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	RW	0	Reserved		RW	1	RW	0	Reserved

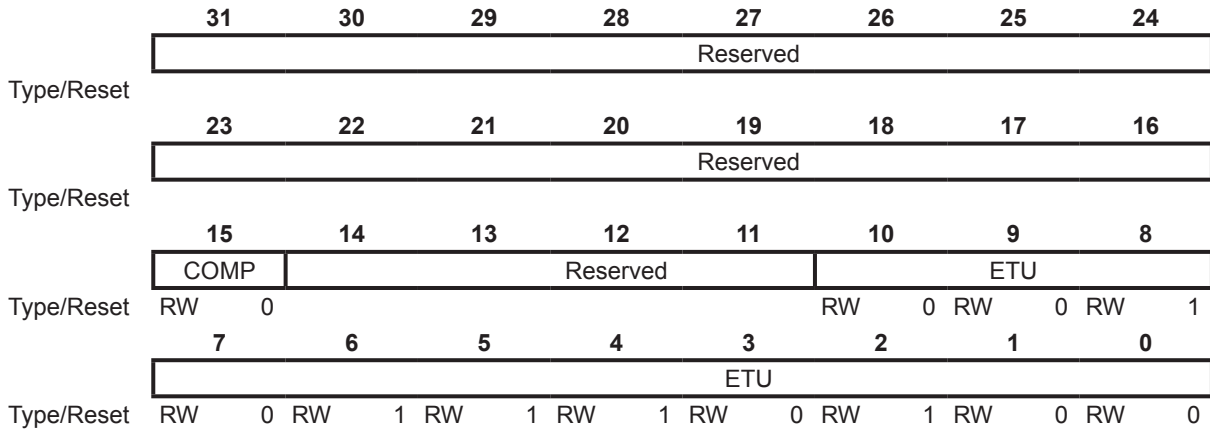
Bits	Field	Descriptions
[7]	CLKSEL	Card Clock Selection 0: The CCLK bit content is present on the external SCI_CLK pin 1: The clock output on the external SCI_CLK pin is sourced from the f_{PSC_CK} clock This bit is used to select the external SCI_CLK pin clock source. It is set and cleared by the application program. It is recommended that to activate the clock at a known level a certain value should be first programmed into the CCLK bit before the CLKSEL bit is switched from 1 to 0.
[3]	CDIO	SCI_DIO pin control 0: SCI_DIO pin is logic level 0 1: SCI_DIO pin in open-drain condition This bit is available only when the SCIM bit in the CR register is cleared to 0 to configure the SCI to operate in the Manual Transfer Mode. It is set and cleared by application program to control the external SCI_DIO pin status in the Manual Mode. Reading this bit will return the present status of the SCI_DIO pin.
[2]	CCLK	SCI_CLK pin control. 0: SCI_CLK pin is logic level 0 1: SCI_CLK pin is logic level 1 This bit is available only when the SCIM bit in the CR register is cleared to 0 to configure the SCI to operate in the Manual Transfer Mode. It is set and cleared by application program to control the external SCI_CLK pin status in the Manual Mode. Reading this bit will return the current value in the register and not the present status of the external SCI_CLK pin. To ensure that the clock remains at a known level a certain value should be first programmed into the CCLK bit before the CLKSEL bit is switched from 1 to 0.

SCI Elementary Time Unit Register – ETUR

The register specifies the value determined by the formula described in the ETU section. It also includes the Compensation function enable control bit for the ETU time granularity.

Offset: 0x00C

Reset value: 0x0000_0174



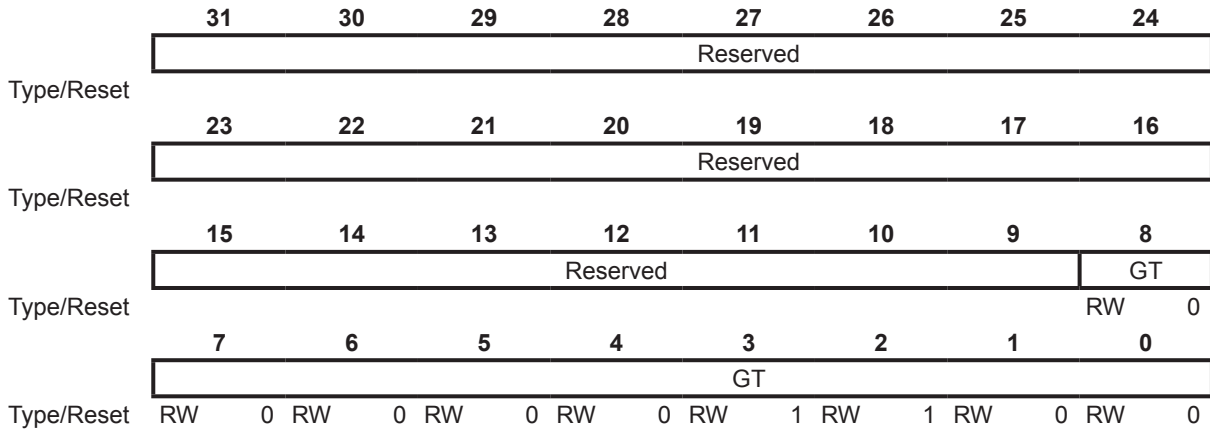
Bits	Field	Descriptions
[15]	COMP	Elementary Time Unit Compensation mode enable control 0: Compensation mode is disabled 1: Compensation mode is enabled This bit is set and cleared by application program and used to control the ETU compensation function. For more details regarding the compensation function consult the Elementary Time Unit section.
[10:0]	ETU	ETU value for a character data bit This field is configured by the application program to modify the ETU time duration. Note that the value of ETU must be in the range of 0x00C to 0x7FF. To obtain the maximum ETU decimal value of 2048, a 0x000 value should be written into this bit field.

SCI Guard Time Register – GTR

This register specifies the guard time value obtained from the Answer-to-Reset packet described in the Guard Time Counter section.

Offset: 0x010

Reset value: 0x0000_000C



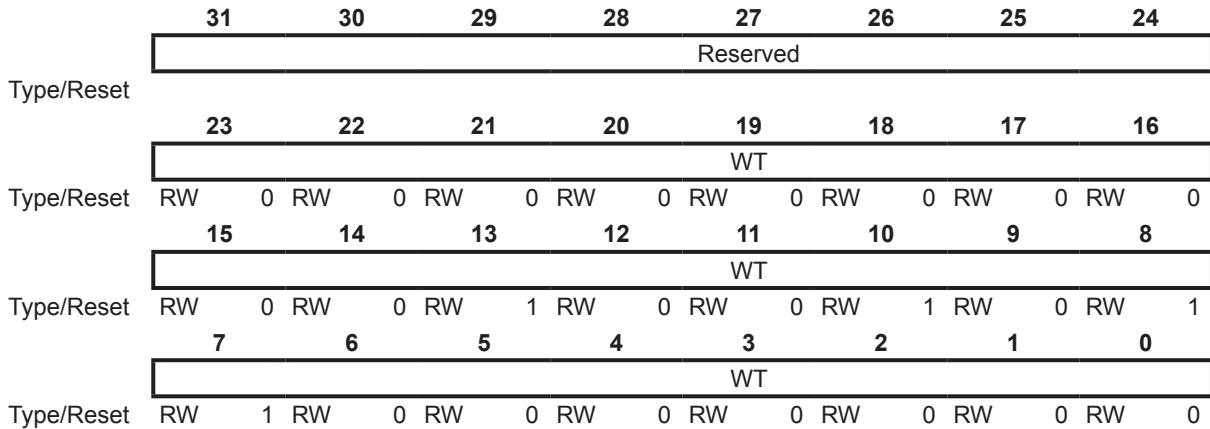
Bits	Field	Descriptions
[8:0]	GT	Character Guard Time value This field is configured by the application program to modify the guard time duration. The updated GT value will be loaded into the GT counter at the end of the current guard time period. Note that the GT value must be in the range from 0x00C to 0x1FF.

SCI Waiting Time Register – WTR

This register specifies the waiting time value obtained from the Answer-to-Reset packet described in the Waiting Time Counter section.

Offset: 0x014

Reset value: 0x0000_2580



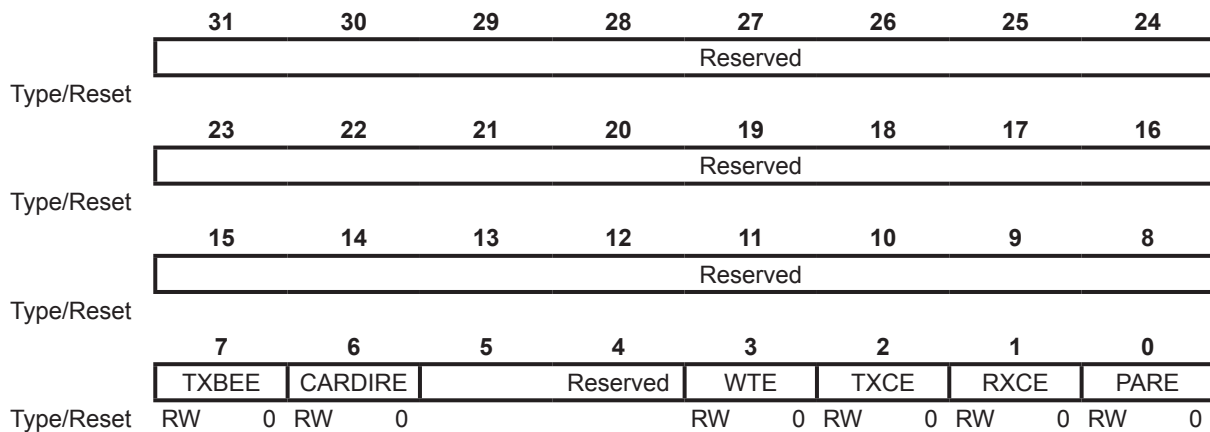
Bits	Field	Descriptions
[23:0]	WT	Character Waiting Time value expressed in ETU (0/16777215). This field is configured by the application program to modify the waiting time duration. The reload conditions of the updated waiting time counter value are described in the waiting time counter section. Refer to the waiting time counter section for more details. Note that the WT value can range from 0x00_0000 to 0xFF_FFFF.

SCI Interrupt Enable Register – IER

This register specifies the interrupt enable control bits for all of the interrupt events in the SCI.

Offset: 0x018

Reset value: 0x0000_0000



Bits	Field	Descriptions
[7]	TXBEE	Transmit buffer empty interrupt enable control 0: Disabled 1: Enabled This bit is set and cleared by application program and is used to control the Transmit Buffer Empty interrupt. If this bit is set to 1, the transmit buffer empty interrupt will be generated when the transmit buffer is empty.
[6]	CARDIRE	Card Insertion / Removal interrupt enable control 0: Disabled 1: Enabled This bit is set and cleared by application program and is used to control the card insertion/removal interrupt. If this bit is set to 1, the card insertion/removal interrupt will be generated when the external Smart Card is inserted or removed.
[3]	WTE	Waiting Timer Underflow interrupt enable control 0: Disabled 1: Enabled This bit is set and cleared by the application program and is used to control the Waiting Timer underflow interrupt. If this bit is set to 1, the waiting time counter underflow interrupt will be generated when the waiting time counter underflows.
[2]	TXCE	Character Transmission Completion interrupt enable control 0: Disabled 1: Enabled This bit is set and cleared by the application program and is used to control the Character Transmission Completion interrupt. If this bit is set to 1, the Character Transmission Completion interrupt will be generated at the end of the character transmission.

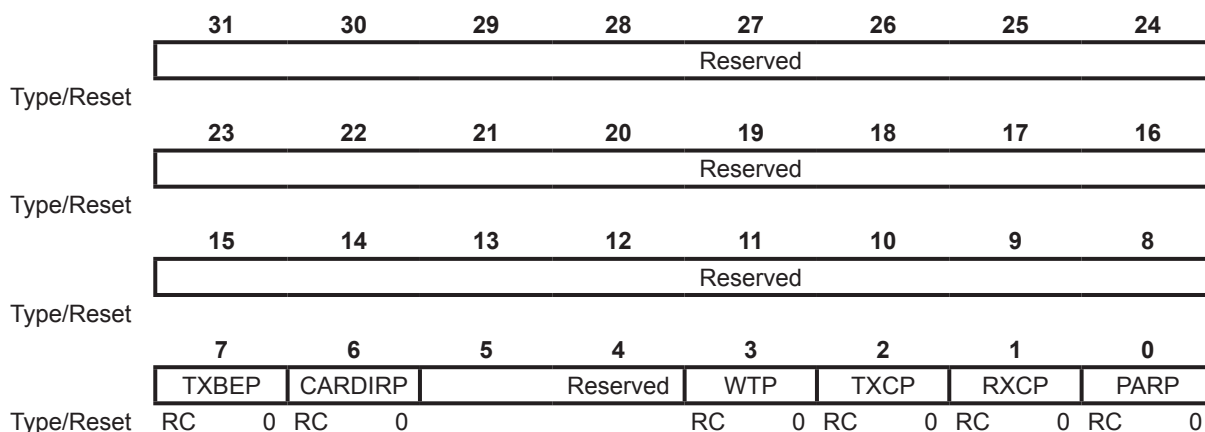
Bits	Field	Descriptions
[1]	RXCE	Character Reception Completion interrupt enable control 0: Disabled 1: Enabled This bit is set and cleared by the application program and is used to control the Character Reception Completion interrupt. If this bit is set to 1, the Character Reception Completion interrupt will be generated at the end of the character reception.
[0]	PARE	Parity Error interrupt enable control 0: Disabled 1: Enabled This bit is set and cleared by the application program and is used to control the parity error interrupt. if this bit is set to 1, the Parity Error interrupt will be generated when a parity error occurs.

SCI Interrupt Pending Register – IPR

This register contains the interrupt pending flags for all of the interrupt events in the SCI. These pending flags can be masked by the corresponding interrupt enable control bits.

Offset: 0x01C

Reset value: 0x0000_0000



Bits	Field	Descriptions
[7]	TXBEP	<p>Transmit Buffer Empty interrupt pending flag</p> <p>0: No interrupt pending 1: Interrupt pending</p> <p>This bit is set by hardware and cleared by a read access to this register using the application program. This bit is used to indicate if there is a Transmit Buffer Empty interrupt pending or not. If the Transmit Buffer is empty and the corresponding interrupt enable control bit is set to 1, this bit will be set to 1 to indicate that the transmit buffer empty interrupt is pending.</p>
[6]	CARDIRP	<p>Card Insertion/Removal interrupt pending flag</p> <p>0: No interrupt pending 1: Interrupt pending</p> <p>This bit is set by hardware and cleared by a read access to this register using the application program. It is used to indicate if there is an external Smart Card insertion/removal interrupt pending or not. If an external Smart Card is inserted or removed and the corresponding interrupt enable control bit is set to 1, this bit will be set to 1 to indicate that the Card insertion/removal interrupt is pending.</p>
[3]	WTP	<p>Waiting Timer Underflow interrupt pending flag</p> <p>0: No interrupt pending 1: Interrupt pending</p> <p>This bit is set by hardware and cleared by a read access to this register using the application program. It is used to indicate if there is a waiting time counter underflow interrupt pending or not. If the waiting time counter underflows and the corresponding interrupt enable control bit is set to 1, this bit will be set to 1 to indicate that the waiting time counter underflow interrupt is pending.</p>

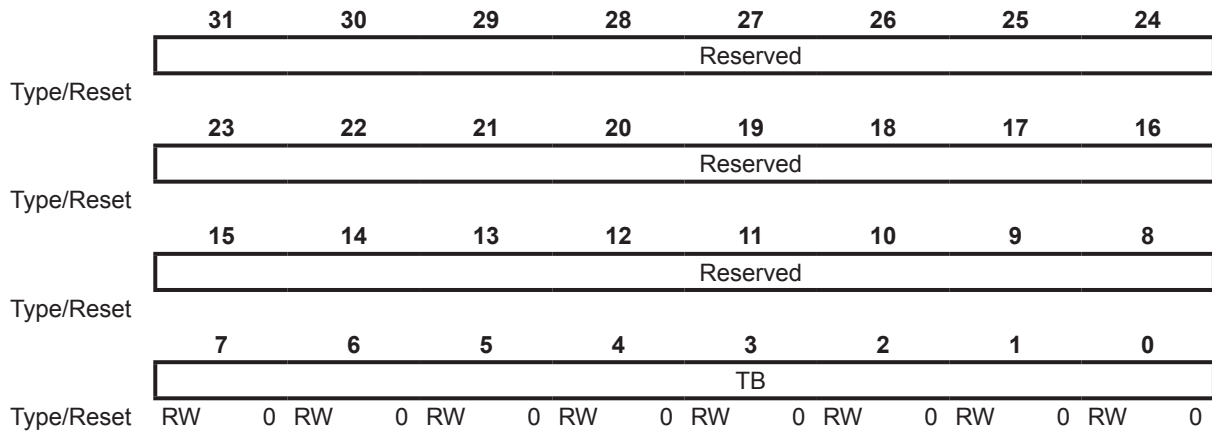
Bits	Field	Descriptions
[2]	TXCP	<p>Character Transmission Completion interrupt pending flag</p> <p>0: No interrupt pending 1: Interrupt pending</p> <p>This bit is set by hardware and cleared by a read access to this register using the application program. It is used to indicate if there is a Character Transmission Completion interrupt pending or not. If a character has been transmitted and the related interrupt enable control bit is set to 1, this bit will be set to 1 to indicate that the character transmission completion interrupt is pending.</p>
[1]	RXCP	<p>Character Reception Completion interrupt pending flag</p> <p>0: No interrupt pending 1: Interrupt pending</p> <p>This bit is set by hardware and cleared by a read access to this register using the application program. It is used to indicate if there is a Character Reception Completion interrupt pending or not. If a character has been received and the relevant interrupt enable control bit is set to 1, this bit will be set to 1 to indicate that the character reception completion interrupt is pending.</p>
[0]	PARP	<p>Parity Error interrupt pending flag</p> <p>0: No interrupt pending 1: Interrupt pending</p> <p>This bit is set by hardware and cleared by a read access to this register using the application program. It is used to indicate if there is a Parity Error interrupt pending or not. If the parity error occurs and its interrupt enable control bit is set to 1, this bit will be set to 1 to indicate that the parity error interrupt is pending.</p>

SCI Transmit Buffer – TXB

This register is used to store the SCI data to be transmitted.

Offset: 0x020

Reset value: 0x0000_0000



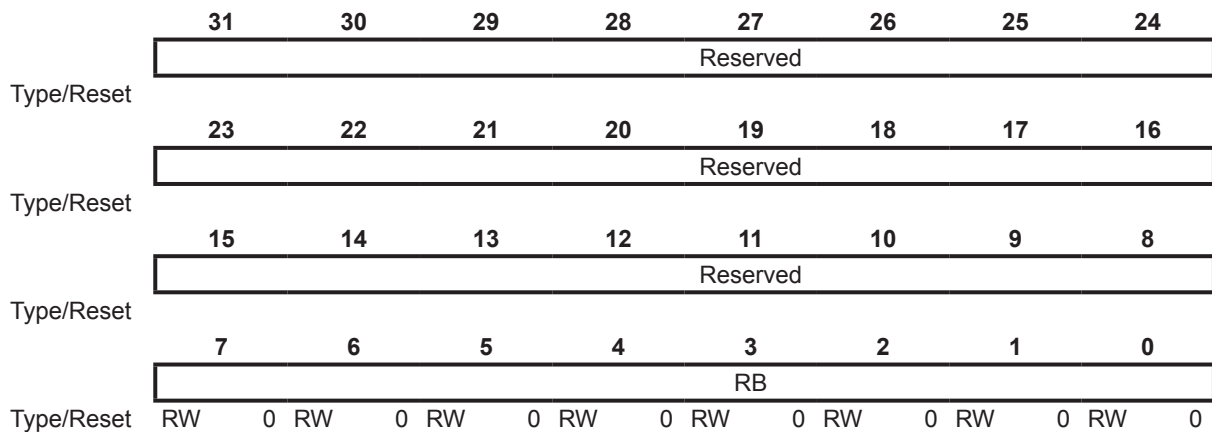
Bits	Field	Descriptions
[7:0]	TB	SCI data byte to be transmitted

SCI Receive Buffer – RXB

This register is used to store the SCI received data.

Offset: 0x024

Reset value: 0x0000_0000



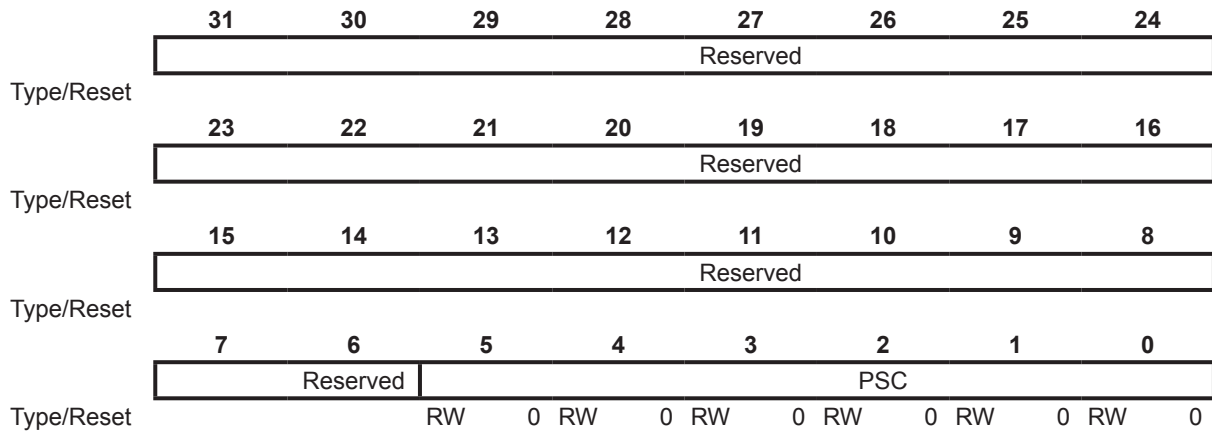
Bits	Field	Descriptions
[7:0]	RB	SCI Received data byte

SCI Prescaler Register – PSCR

This register specifies the prescaler division ratio which is used the SCI internal clock.

Offset: 0x028

Reset value: 0x0000_0000



Bits	Field	Descriptions
[5:0]	PSC	SCI prescaler division ratio 0: $f_{PSC_CK} = f_{PCLK}$ 1~63: $f_{PSC_CK} = \frac{f_{PCLK}}{2 \times PSC}$

25 USB Device Controller (USB)

Introduction

The USB device controller is compliant with the USB 2.0 full-speed specification. There is one control endpoint know 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.

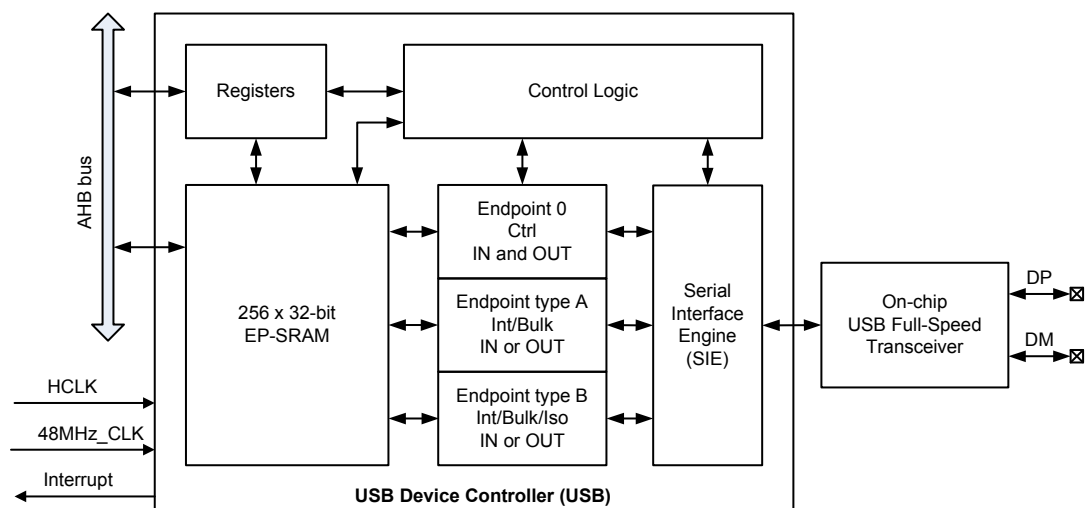


Figure 180. USB Block Diagram

Features

- Complies with USB 2.0 full-speed (12Mbps) specification
- Fully integrated USB full-speed transceiver
- 1 control endpoint (EP0) for control transfer
- 3 single-buffered endpoint (EP1~EP3) for bulk and interrupt transfer
- 4 double-buffered endpoint (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 supports a single buffering function which is used for Bulk and Interrupt IN or OUT data transfer. There are four other endpoints (EP4~EP7) which supports 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 63. 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 CPU 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 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 64. USB Data Types and Buffer Size

Transfer Type	Direction	Supported Buffer Size	Bandwidth	CRC	Retrying
Control	Bidirectional	8, 16, 32, 64	Not guaranteed	Yes	Automatic
Bulk	Unidirectional	8, 16, 32, 64	Not guaranteed	Yes	Yes
Interrupt	Unidirectional	≤ 64	Not guaranteed	Yes	Yes
Isochronous	Unidirectional	< 512	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.

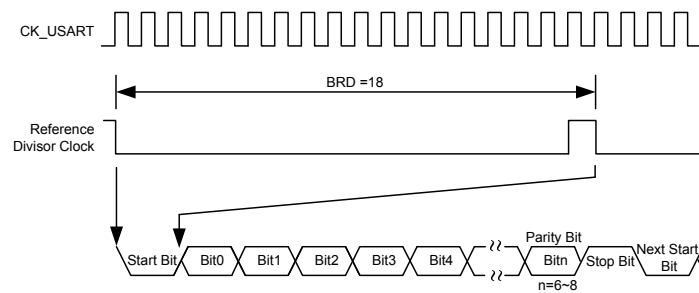


Figure 181. Endpoint Buffer Allocation Example

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 debit-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 CPU to process and the hardware will ensure that it continues to receive the current data packet in the other buffer during a 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.

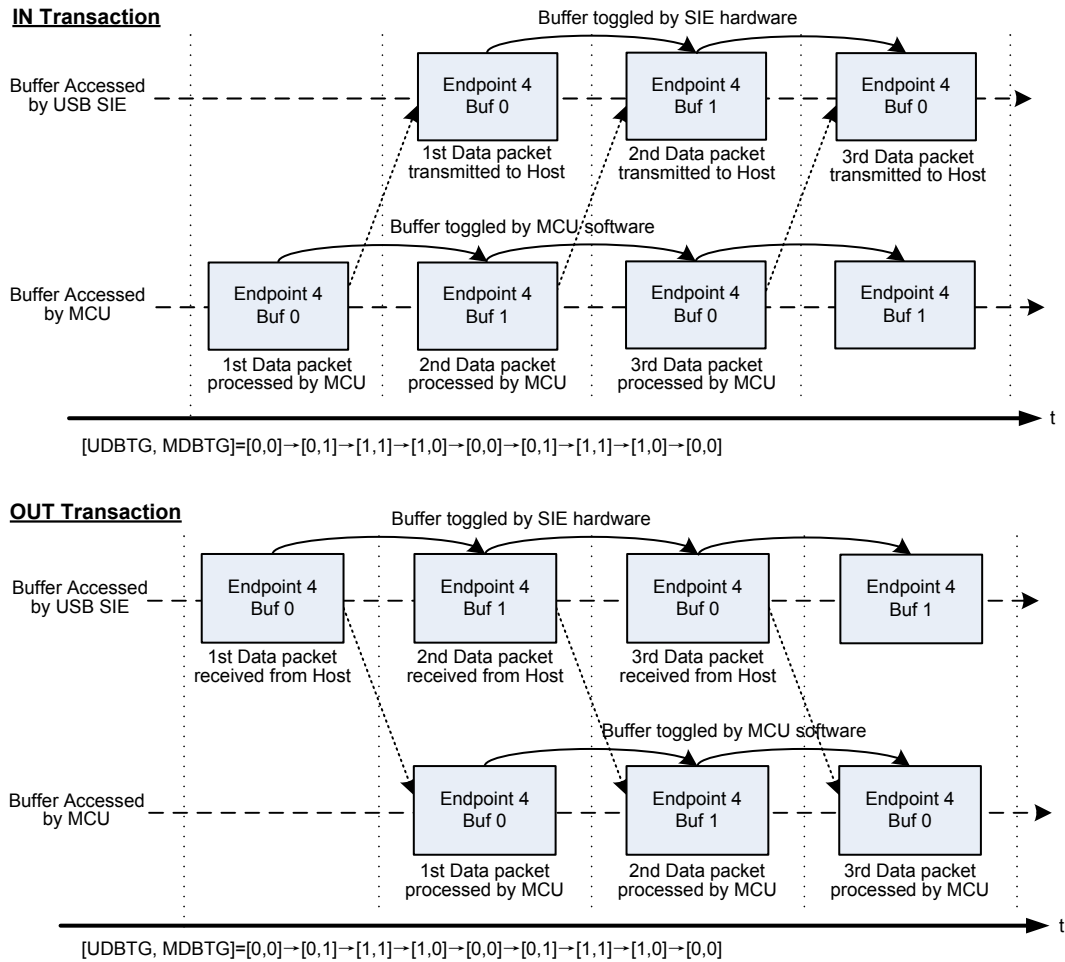


Figure 182. Double-buffering Operation Example

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 65. USB Register Map

Register	Offset	Description	Reset Value
USB Base Address = 0x400A_8000			
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_0002
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

Register	Offset	Description	Reset Value
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
USBEP6CSR	0x08C	USB Endpoint 6 Control and Status Register	0x0000_0002
USBEP6IER	0x090	USB Endpoint 6 Interrupt Enable Register	0x0000_0000
USBEP6ISR	0x094	USB Endpoint 6 Interrupt Status Register	0x0000_0000
USBEP6TCR	0x098	USB Endpoint 6 Transfer Count Register	0x0000_0000
USBEP6CFGR	0x09C	USB Endpoint 6 Configuration Register	0x1000_03FF
USBEP7CSR	0x0A0	USB Endpoint 7 Control and Status Register	0x0000_0002
USBEP7IER	0x0A4	USB Endpoint 7 Interrupt Enable Register	0x0000_0000
USBEP7ISR	0x0A8	USB Endpoint 7 Interrupt Status Register	0x0000_0000
USBEP7TCR	0x0AC	USB Endpoint 7 Transfer Count Register	0x0000_0000
USBEP7CFGR	0x0B0	USB Endpoint 7 Configuration Register	0x1000_03FF

Register Descriptions

USB Control and Status Register – USBCSR

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	Reserved															
Type/Reset																							
23	22	21	20	19	18	17	16	Reserved															
Type/Reset																							
15	14	13	12	11	10	9	8	Reserved				DPWKEN	DPPUEN	SRAMRSTC	ADRSET								
Type/Reset												RW	0	RW	0	RW	0	RW	0				
7	6	5	4	3	2	1	0	RXDM	RXDP	GENRSM	Reserved	LPMODE	PDWN	FRES	Reserved								
Type/Reset												RO	X	RO	X	RW	0	RW	0	RW	1	RW	1

Bits	Field	Descriptions
[11]	DPWKEN	DP Wake Up Enable 0: Disable DP wake up 1: Enable DP wake up
[10]	DPPUEN	DP Pull Up Enable 0: Disable DP pull up 1: Enable DP pull up
[9]	SRAMRSTC	USB SRAM reset condition 0: Reset USB SRAM when (DP, DM) = (0, 0) 1: User can access USB SRAM in spite of (DP, DM) state
[8]	ADRSET	Device Address Setting Control This bit is used to determine when the USB SIE updates the device address with the value of the UDEVA register. 0: The SIE updates the device address immediately after an address is written into the UDEVA register. 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.
[7]	RXDM	Received DM Line Status 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.
[6]	RXDP	Received DP Line Status This bit is used to observe the status of DP data line status at the end of suspend routines to determine whether a wakeup event has occurred.
[5]	GENRSM	Resume Request Generation Control 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 wakeup function is always enabled. This bit will be cleared to 0 after a resume signal, sent by the USB host, has been received.

Bits	Field	Descriptions
[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 wakeup event, is detected by the hardware, this bit should be cleared by software.</p> <p>0: Exit the Low-power mode 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 mode 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 1: Force USB reset</p>

Table 66. Resume Event Detection

[RXDP, RXDM] Status	Wakeup 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	Reserved							
Type/Reset																
	23	22	21	20	19	18	17	16	Reserved							
Type/Reset																
	15	14	13	12	11	10	9	8	EP7IE	EP6IE	EP5IE	EP4IE	EP3IE	EP2IE	EP1IE	EPOIE
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	Reserved		ESOFIE	SUSPIE	RSMIE	URSTIE	SOFIE	UGIE
Type/Reset			RW	0	RW	0	RW	0	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
[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: USB Global interrupt is disabled 1: USB Global interrupt is enabled 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	Reserved							
Type/Reset																
	23	22	21	20	19	18	17	16	Reserved							
Type/Reset																
	15	14	13	12	11	10	9	8	EP7IF	EP6IF	EP5IF	EP4IF	EP3IF	EP2IF	EP1IF	EP0IF
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	Reserved		ESOFIF	SUSPIF	RSMIF	URSTIF	SOFIF	Reserved
Type/Reset			RW	0	WC	0	WC	0	WC	0	WC	0	WC	0		

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 interrupt. 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.
[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 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 to 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.</p> <p>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					RO	0	RO	0	RO	0
	15	14	13	12	11	10	9	8			
Type/Reset	Reserved					FRNUM					
	7	6	5	4	3	2	1	0			
Type/Reset	FRNUM										
	RO	0	RO	0	RO	0	RO	0	RO	0	

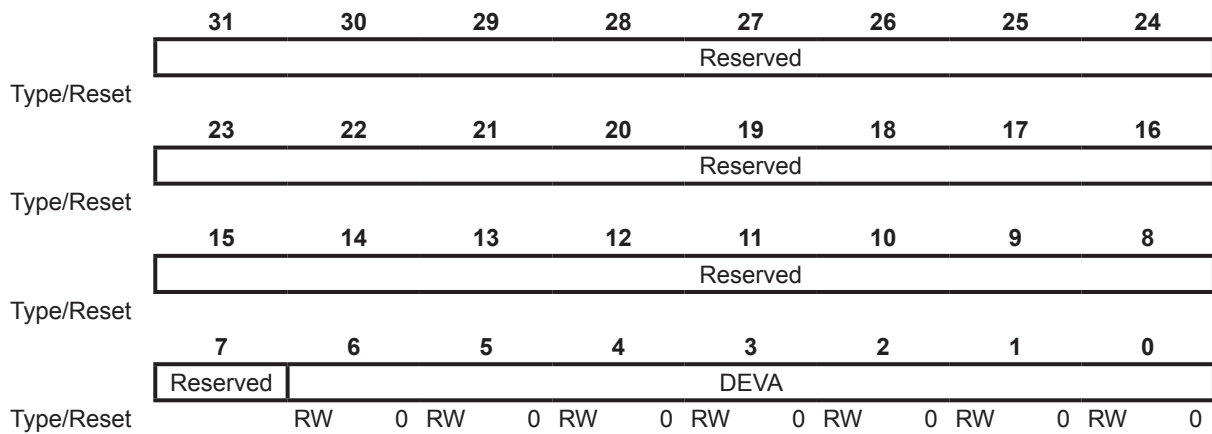
Bits	Field	Descriptions
[18:17]	LSOF	Lost Start-of-Frame number These bits are written and incremented by 1 by the hardware each time the ESOFIF bit is set. It is used to count the number of lost SOF packets. When a SOF packet has been received, these bits are cleared.
[16]	SOFLOCK	Start-of-Frame Lock 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 in the USBFCR register. If there 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 – USBDEVA

This register specifies the USB device address.

Offset: 0x010

Reset value: 0x0000_0000



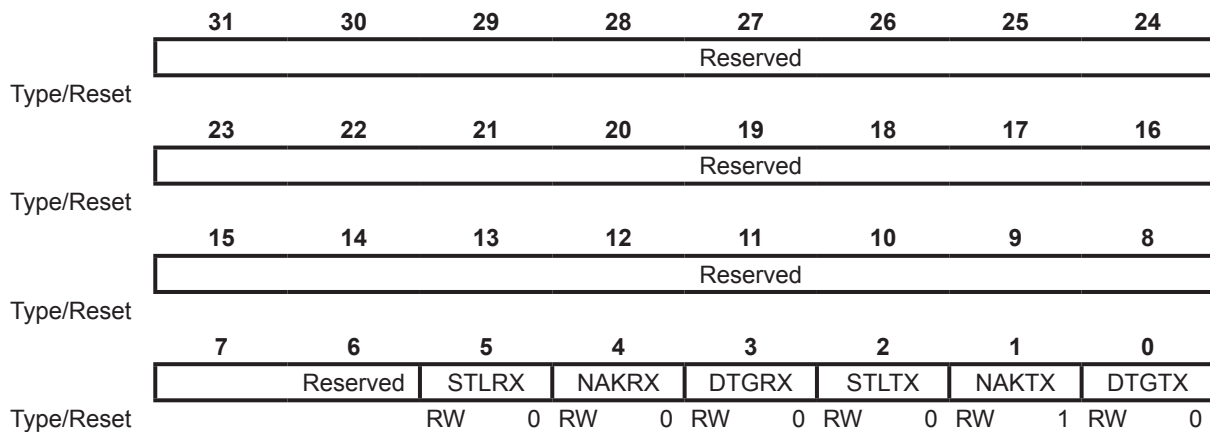
Bits	Field	Descriptions
[6:0]	DEVA	Device Address This field is used to specify the USB device address. This field is cleared when a USB reset event occurs.

USB Endpoint 0 Control and Status Register – USBEP0CSR

This register specifies the Endpoint 0 control and status.

Offset: 0x014

Reset value: 0x0000_0002



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 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 SETUP token is received.</p> <p>This bit can be read and written and can only be toggled by writing 1.</p>

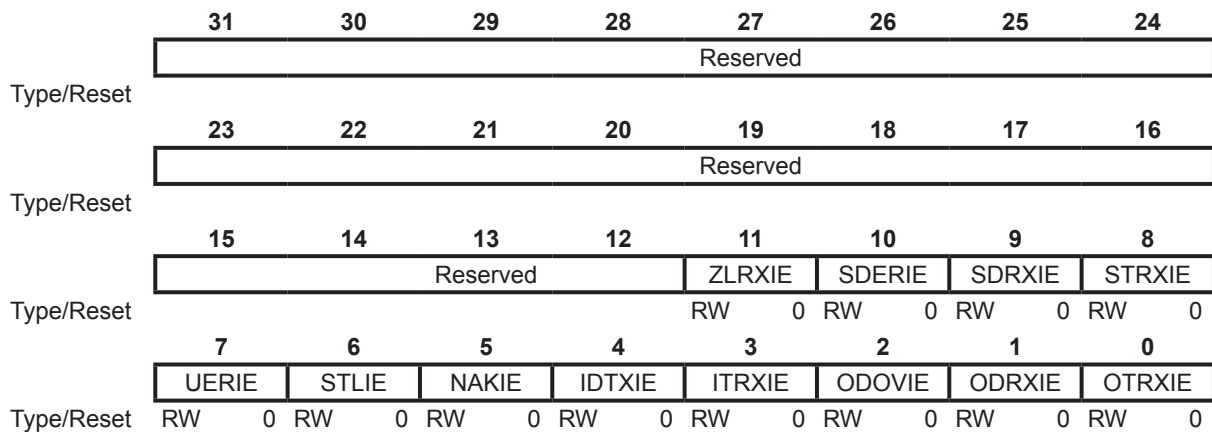
Bits	Field	Descriptions
[1]	NAKTX	<p>NAK Status for transmission (IN) transfer</p> <p>This bit is toggled from 0 to 1 by the hardware circuitry, which will result in a NAK signal in the handshake phase of an IN transaction after an ACK signal has been received. It indicates that the USB device is temporarily 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.</p> <p>This bit can be read and written and can only be toggled by writing 1.</p>
[0]	DTGTX	<p>Data Toggle Status for transmission (IN) transfer</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 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.</p> <p>This bit can be read and written and can only be toggled by writing 1.</p>

USB Endpoint 0 Interrupt Enable Register – USBEP0IER

This register specifies the Endpoint 0 interrupt control bits.

Offset: 0x018

Reset value: 0x0000_0000



Bits	Field	Descriptions
[11]	ZLRXIE	<p>Zero Length Data Received Interrupt Enable Control</p> <p>0: Disable interrupt 1: Enable interrupt</p>
[10]	SDERIE	<p>SETUP Data Error Interrupt Enable Control</p> <p>0: Disable interrupt 1: Enable interrupt</p>
[9]	SDRXIE	<p>SETUP Data Received Interrupt Enable Control</p> <p>0: Disable interrupt 1: Enable interrupt</p>

Bits	Field	Descriptions
[8]	STRXIE	SETUP Token Received Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[7]	UERIE	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								
	7	6	5	4	3	2	1	0	WC	0	WC	0	WC	0	WC	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.

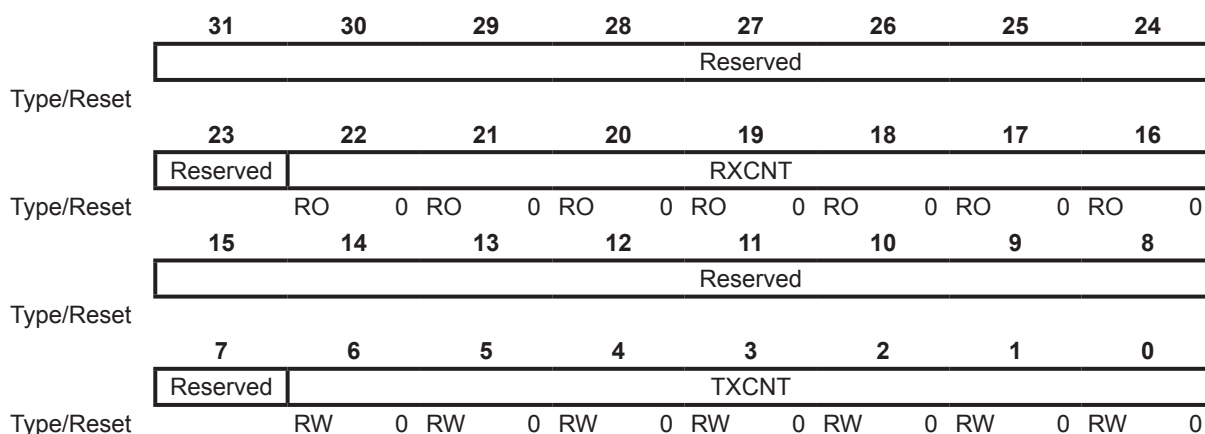
Bits	Field	Descriptions
[4]	IDTXIF	IN Data Transmitted Interrupt Flag This bit is set by the hardware when a data packet is transmitted to and then an ACK signal is received from the USB host. This bit is cleared by hardware when a SETUP Token is received or by writing 1.
[3]	ITRXIF	IN Token Received 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 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



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_0002

	31	30	29	28	27	26	25	24
	EPEN		Reserved				EPADR	
Type/Reset	RO	1			RO	0	RO	0
	23	22	21	20	19	18	17	16
	Reserved							EPLEN
Type/Reset								RW
	15	14	13	12	11	10	9	8
	EPLEN						EPBUFA	
Type/Reset	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	1
								0

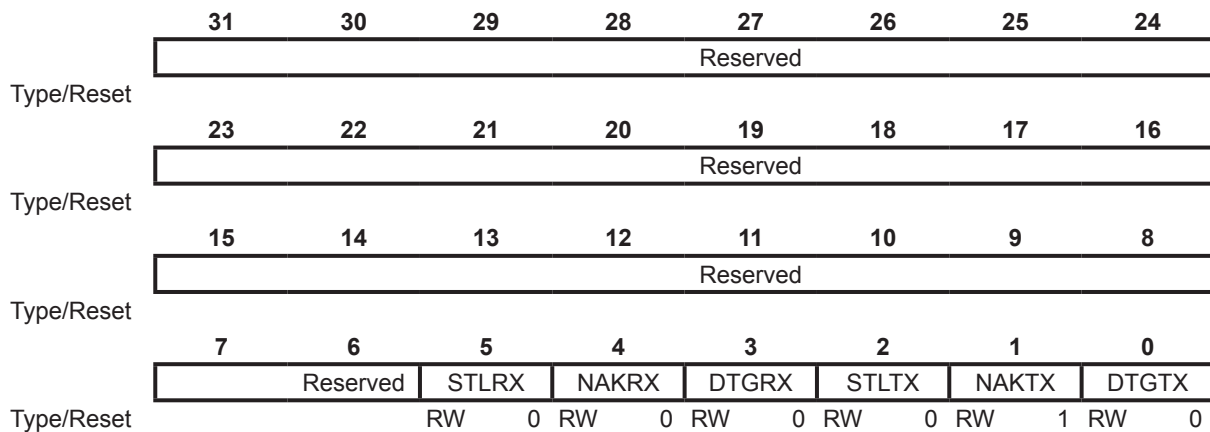
Bits	Field	Descriptions
[31]	EPEN	Endpoint Enable Control This bit is always set to 1 by the hardware circuitry to always enable Endpoint 0.
[27:24]	EPADR	Endpoint Address This field is always set to 0 by the hardware circuitry.
[16:10]	EPLEN	Endpoint Buffer Length This field is used to specify the control 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



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>

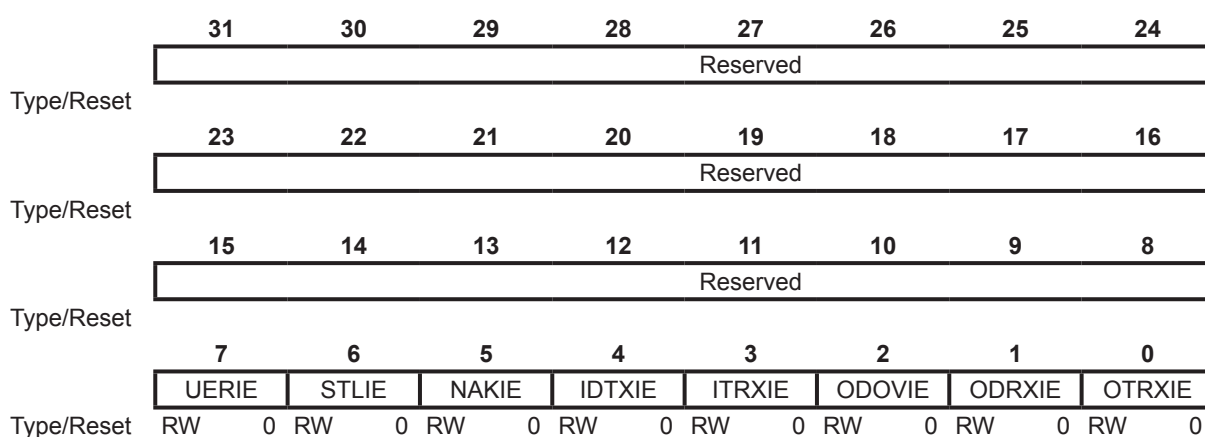
Bits	Field	Descriptions
[0]	DTGTX	Data Toggle bit for transmission transfers. This bit contains the required value of the data toggle bit (0=DATA0, 1=DATA1) for the next data packet to be transmitted. When the current data packet is transmitted by the USB device and the corresponding ACK signal sent from the USB host is received, the hardware circuitry will toggle this bit and then the next data packet will be transmitted. 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 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



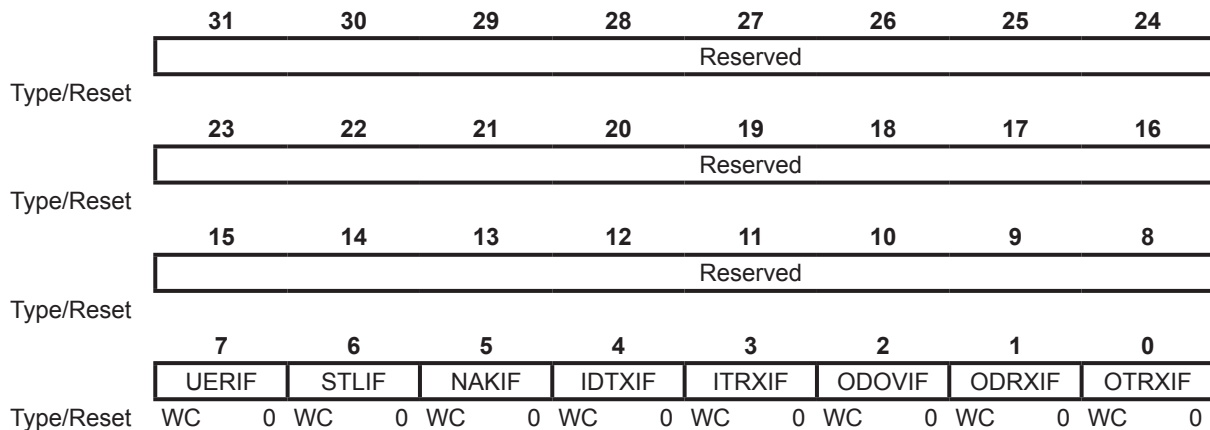
Bits	Field	Descriptions
[7]	UERIE	USB Error Interrupt Enable Control. 1: Enable interrupt 0: Disable interrupt
[6]	STLIE	STALL Transmitted Interrupt Enable Control. 1: Enable interrupt 0: Disable interrupt
[5]	NAKIE	NAK Transmitted Interrupt Enable Control. 1: Enable interrupt 0: Disable interrupt
[4]	IDTXIE	IN Data Transmitted Interrupt Enable Control. 1: Enable interrupt 0: Disable interrupt
[3]	ITRXIE	IN Token Received Interrupt Enable Control. 1: Enable interrupt 0: Disable interrupt
[2]	ODOVIE	OUT Data Buffer Overrun Interrupt Enable Control. 1: Enable interrupt 0: Disable interrupt

Bits	Field	Descriptions
[1]	ODRXIE	OUT Data Received Interrupt Enable Control. 1: Enable interrupt 0: Disable interrupt
[0]	OTRXIE	OUT Token Received Interrupt Enable Control. 1: Enable interrupt 0: Disable 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



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-token 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 a NAK-token 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-token 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.

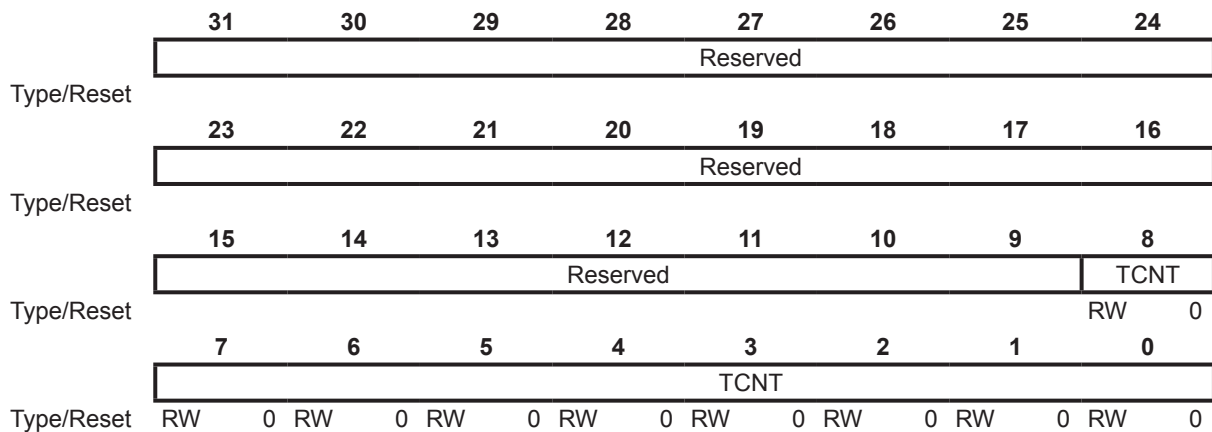
Bits	Field	Descriptions
[2]	ODOVIF	OUT Data Buffer Overrun Interrupt Flag. This bit is set by the hardware circuitry when the received data byte count is larger than the corresponding endpoint OUT data buffersize. Writing 1 into this status bit will clear it to 0.
[1]	ODRXIF	OUT Data Received Interrupt 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



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
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
UDBTG	MDBTG	STLRX	NAKRX	DTGRX	STLTX	NAKTX	DTGTX
RW	0	RW	0	RW	0	RW	1
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 CPU 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, a 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 CPU 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 CPU
IN	0	0	None*	EP_BUF0
	0	1	EP_BUF0	EP_BUF1
	1	0	EP_BUF1	EP_BUF0
	1	1	None*	EP_BUF1

Transaction Type	UDBTG	MDBTG	Buffer written by SIE	Buffer read by CPU
OUT	0	0	None*	EP_BUF1
	0	1	EP_BUF0	EP_BUF1
	1	0	EP_BUF1	EP_BUF0
	1	1	None*	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>CPU Double Buffer Toggle bit</p> <p>The MDBTG bit is used to indicate which data buffer is accessed by the CPU if the double buffering function is enabled. It can be toggled to switch to the other buffer by the CPU application software after the data in the current buffer accessed by the CPU 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>

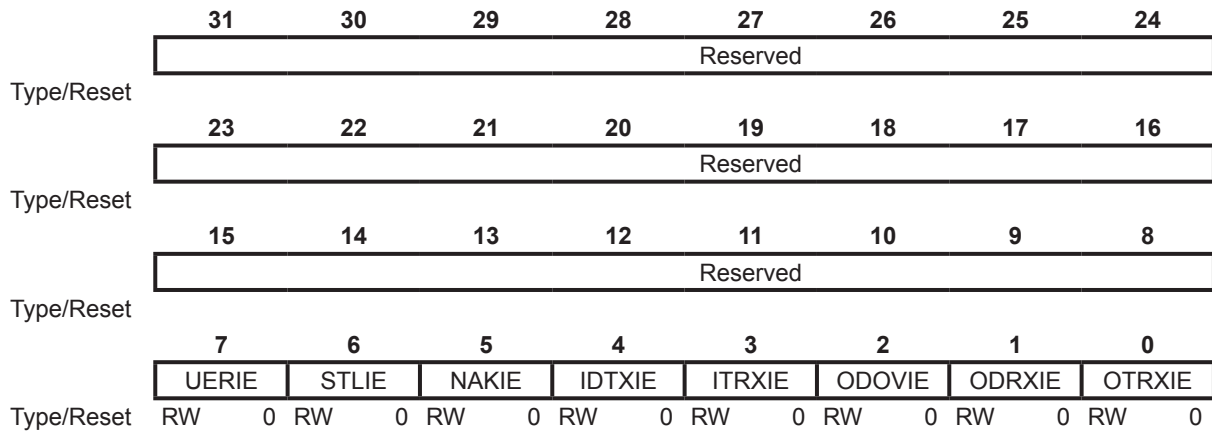
Bits	Field	Descriptions
[0]	DTGTX	<p>Data Toggle bit for transmission transfers.</p> <p>If the endpoint is not used for Isochronous transfer, this bit is available for usage. This bit contains the required value of the data toggle bit (0 = DATA0, 1 = DATA1) for the next data packet to be transmitted. 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.</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 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



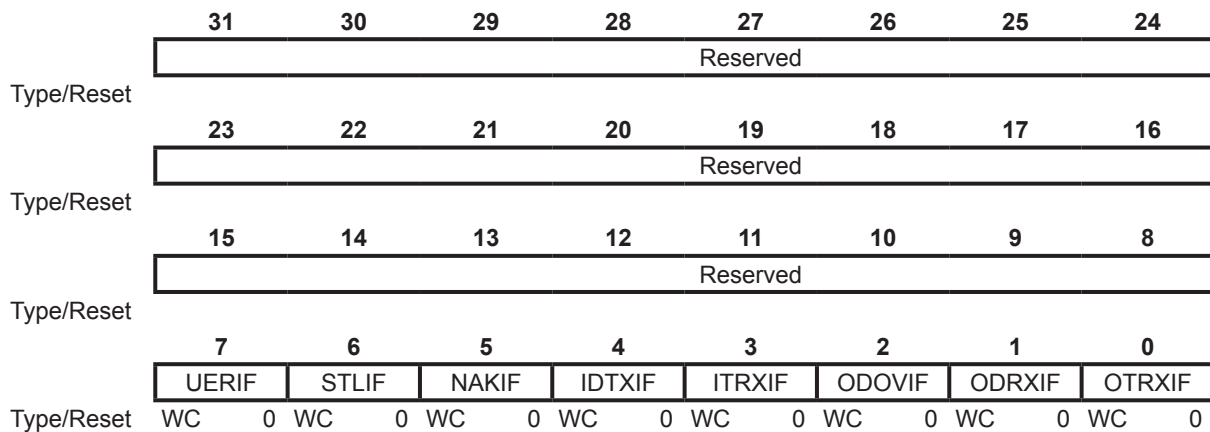
Bits	Field	Descriptions
[7]	UERIE	USB Error Interrupt Enable Control 1: Enable interrupt 0: Disable interrupt
[6]	STLIE	STALL Transmitted Interrupt Enable Control 1: Enable interrupt 0: Disable interrupt
[5]	NAKIE	NAK Transmitted Interrupt Enable Control 1: Enable interrupt 0: Disable interrupt
[4]	IDTXIE	IN Data Transmitted Interrupt Enable Control 1: Enable interrupt 0: Disable interrupt
[3]	ITRXIE	IN Token Received Interrupt Enable Control 1: Enable interrupt 0: Disable interrupt
[2]	ODOVIE	OUT Data Buffer Overrun Interrupt Enable Control 1: Enable interrupt 0: Disable interrupt
[1]	ODRXIE	OUT Data Received Interrupt Enable Control 1: Enable interrupt 0: Disable interrupt
[0]	OTRXIE	OUT Token Received Interrupt Enable Control 1: Enable interrupt 0: Disable 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



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-token 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 a NAK-token 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-token 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.
[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
Type/Reset	Reserved						TCNT1	
							RW	0
							RW	0
	23	22	21	20	19	18	17	16
Type/Reset	TCNT1							
	RW	0	RW	0	RW	0	RW	0
	RW	0	RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8
Type/Reset	Reserved						TCNT0	
							RW	0
							RW	0
	7	6	5	4	3	2	1	0
Type/Reset	TCNT0							
	RW	0	RW	0	RW	0	RW	0
	RW	0	RW	0	RW	0	RW	0

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
	23	22	21	20	19	18	17	16		
	SDBS		Reserved				EPLEN			
Type/Reset	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	1
	7	6	5	4	3	2	1	0		
	EPBUFA									
Type/Reset	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 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 LEN 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.

26 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 non-increment address modes
- 4 transfer event flags – Transfer complete, Half Transfer, Block End and Transfer Error
- Auto-Reload function

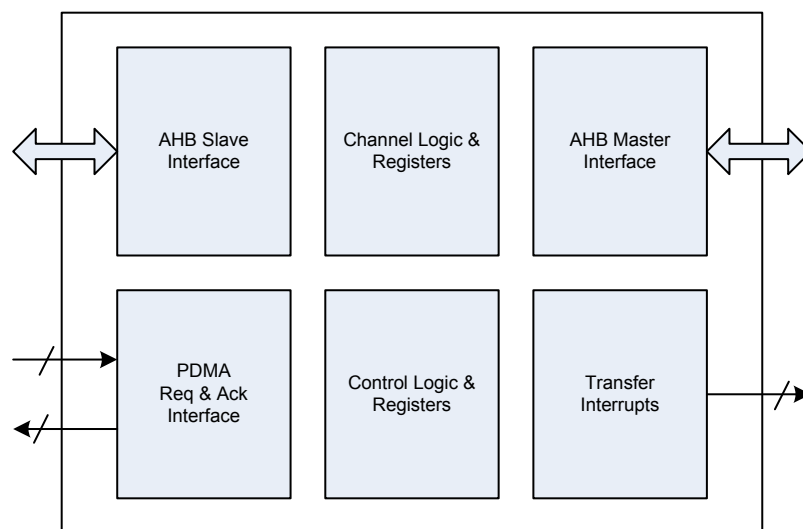


Figure 183. PDMA Block Diagram

Functional Description

AHB Master

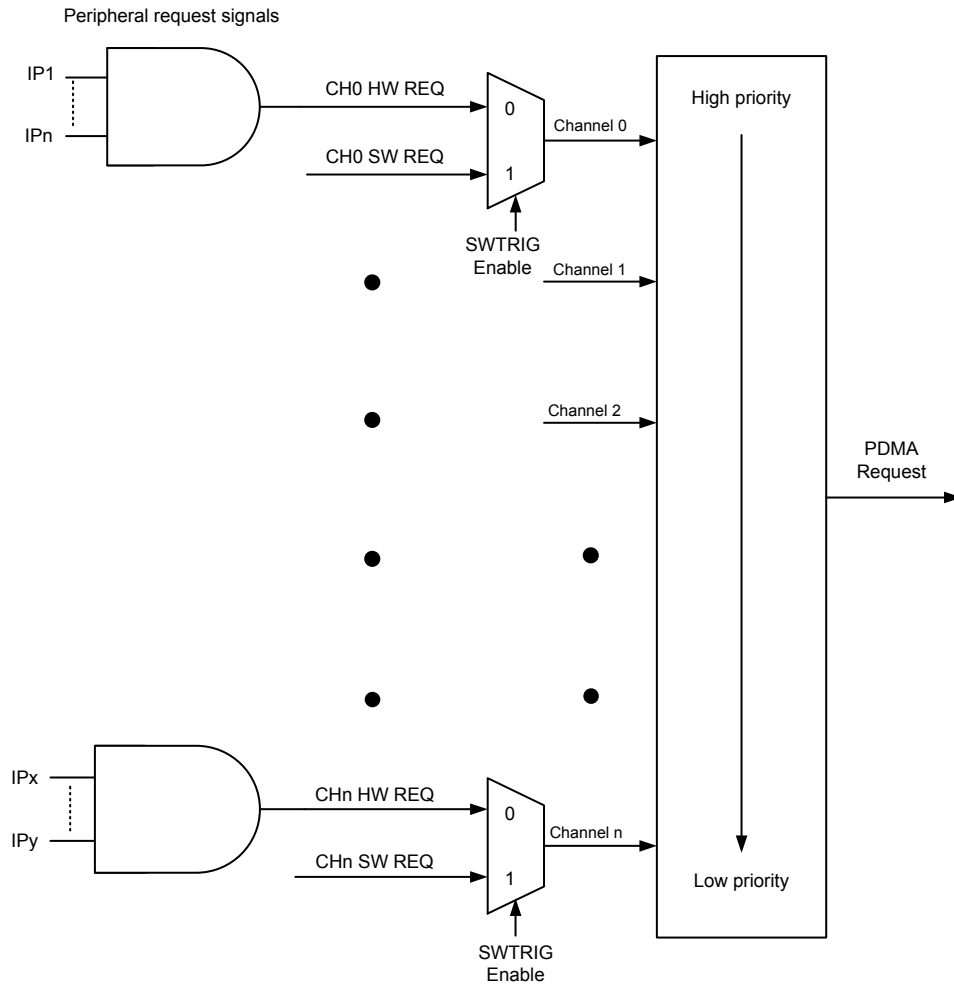
The PDMA is an AHB master connected to other AHB peripherals such as the FLASH memory, the SRAM memory and the AHB-to-APB bridges through the bus-matrix. The CPU and PDMA can access different AHB slaves at the same time via the bus-matrix.

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 (ADC, SPI, I²C, USART and so on) are simply logically ANDed before entering the PDMA, that means that only one request must be enabled at a time in each PDMA channel. Refer to Figure 184: PDMA request mapping architecture and detail peripheral IP requests mapping table is show as the Table 67. The peripheral DMA requests can be independently activated/de-activated by programming the DMA control bit in the registers of the corresponding peripheral.



Peripheral Direct Memory Access (PDMA)

Figure 184. PDMA Request Mapping Architecture

Table 67. PDMA Channel Assignments

IP (x=0,1)	PDMA Channel Number					
	CH0	CH1	CH2	CH3	CH4	CH5
ADC	ADC					
SPIx	SPI0_RX	SPI0_TX			SPI1_RX	SPI1_TX
USARTX	USR0_RX	USR0_TX	USR1_RX	USR1_TX		
UARTX			UR0_RX	UR0_TX	UR1_RX	UR1_TX
SCIx			SCI1_RX	SCI1_TX	SCI0_RX	SCI0_TX
I2Cx			I2C0_RX	I2C1_RX	I2C0_TX	I2C1_TX
MCTM	MT_CH0	MT_TRIG	MT_CH1	MT_CH2	MT_CH3 MT_UEV2	MT_UEV1
GPTMx	GT0_CH1 GT0_CH3	GT0_CH2 GT0_UEV	GT0_CH0 GT0_TRIG	GT1_CH0 GT1_CH3	GT1_CH1 GT1_UEV	GT1_CH2 GT1_TRIG
I ² S		I2S_RX	I2S_TX			

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 below equation:

$$\text{A PDMA channel total transfer data size} = \text{Block transaction count} \times (\text{Block length} \times \text{Data width})$$

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.

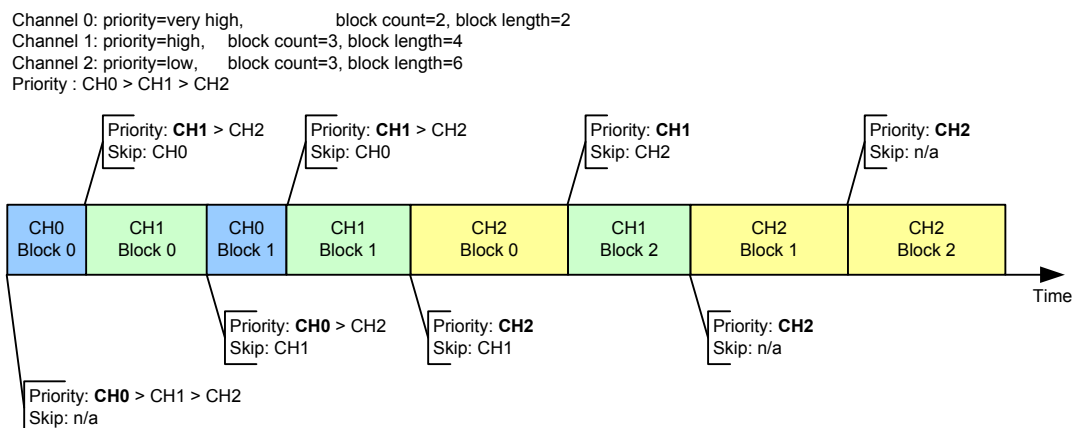


Figure 185. PDMA Channel Arbitration and Scheduling Example

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 68. 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 or TE occurs. Clearing the BE, HT, TC or TE event flags 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 the PDMA receives a request but when the corresponding transfer size setting is equal to zero.

Register Map

The following table shows the PDMA registers and the reset values.

Table 69. PDMA Register Map

Register	Offset	Description	Reset Value
PDMA Base Address = 0x4009_0000			
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
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

Register	Offset	Description	Reset Value
PDMA Base Address = 0x4009_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
PDMAICLR	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

Type/Reset	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				AUTORLn	FIXAENn	CHnPRI											
Type/Reset	7	6	5	4	3	2	1	0	RW	0	RW	0	RW	0	RW	0	RW	0								
Type/Reset	7	6	5	4	3	2	1	0	SRCAMODn	SRCAINCn	DSTAMODn	DSTAINCn	DWIDTHn		SWTRIGn	CHnEN	RW	0	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, PDMACHnCTSR register will be reloaded with the relevant start value and the PDMA channel n will be activated when a transfer is complete. If this bit is cleared to 0, the PDMACHnCTSR register 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	<p>Channel n Source Address Mode selection</p> <p>0: Linear address mode 1: Circular address mode</p> <p>In the linear address mode, the current source address value can be incremented or decremented, determined by the SRCAINCn bit value during a complete transfer. In the circular address mode, the current source address value can be incremented or decremented which is also determined by the SRCAINCn bit value during a block transfer and the lower 16-bit value of the PDMACHnSADR register will be regarded as the current source address when a block transaction has completed.</p>
[6]	SRCAINCn	<p>Channel n Source Address Increment control</p> <p>0: Increment 1: Decrement</p> <p>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.</p>
[5]	DSTAMODn	<p>Channel n Destination Address Mode selection</p> <p>0: Linear address mode 1: Circular address mode</p> <p>In linear address mode, the current destination address value can be incremented or decremented, determined by the DSTAINCn bit value during a complete transfer. In the circular address mode, can be incremented or decremented which is also determined by the DSTAINCn bit value during a block transfer and the lower 16-bit value of the PDMACHnDADR register will be regarded as the current destination address when a block transfer has completed.</p>
[4]	DSTAINCn	<p>Channel n Destination Address Increment Control</p> <p>0: Increment 1: Decrement</p> <p>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.</p>
[3:2]	DWIDTHn	<p>Data Bit Width selection</p> <p>00: 8-bit 01: 16-bit 10: 32-bit 11: Reserved</p> <p>The field is used to select the data bit width of the corresponding PDMA channel n.</p>
[1]	SWTRIGN	<p>Software Trigger control</p> <p>0: No operation 1: Software triggered transfer request</p> <p>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.</p>
[0]	CHnEN	<p>Channel n Enable control</p> <p>0: Disable the PDMA channel n 1: Enable the PDMA channel n</p> <p>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.</p>

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	
	BLKLEn								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW

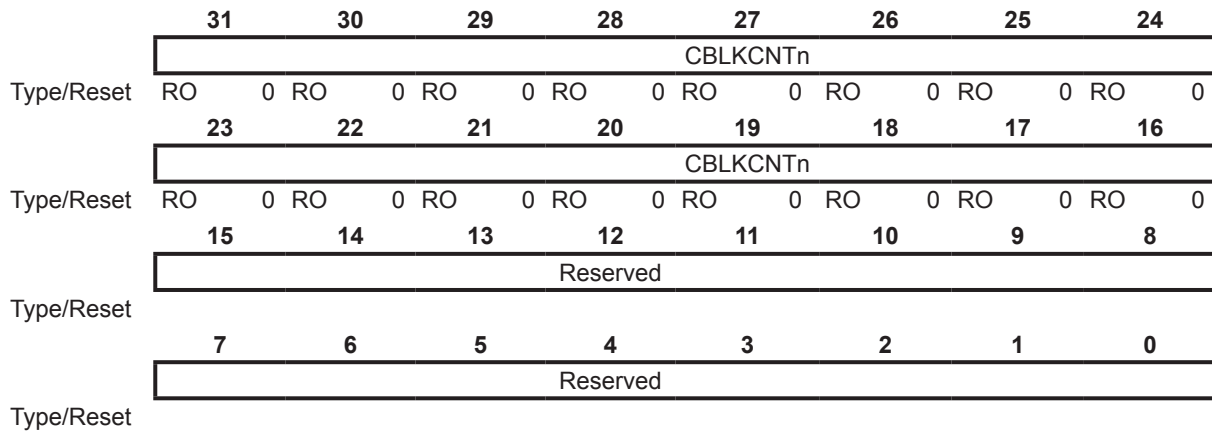
Bits	Field	Descriptions
[31:16]	BLKCNTn	Channel n Block Transaction Count BLKCNTn represents the number of block transactions for a channel n complete transfer. The capacity of a complete transfer is the product of the BLKCNTn and BLKLEn values. The maximum BLKCNTn value is 65,535.
[7:0]	BLKLEn	Channel n Block Length The BLKLEn represents the length of a data block. The data width is defined by the DWIDTHn field in the PDMACHnCR register. The maximum BLKLEn 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



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 decremented 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
	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
	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
	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

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 PDMAISR register. A Transfer error will occur when the PDMA accesses a system reserved address space or the PDMA receives a request but when 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 PDMAISR 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 PDMAISR 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 PDMAISR 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	<p>Channel n Global Transfer Interrupt Status (n= 0 ~ 5)</p> <p>0: No TE, TC, HT or BE event occurs 1: TE, TC, HT, or BE event occurs</p> <p>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 or TE events occur. Also clearing any of the BE, HT, TC or 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.</p>

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
	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
	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
	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

Bits	Field	Descriptions
[29], [24], [19], [14], [9], [4]	TEICLRn	<p>Channel n Transfer Error Interrupt Status Clear (n = 0 ~ 5)</p> <p>0: No Operation 1: Clear the corresponding TEISTAn bit in the PDMAISR register</p> <p>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.</p>
[28], [23], [18], [13], [8], [3]	TCICLRn	<p>Channel n Transfer Complete Interrupt Status Clear (n = 0 ~ 5)</p> <p>0: No Operation 1: Clear the corresponding TCISTAn bit in the PDMAISR register</p> <p>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.</p>
[27], [22], [17], [12], [7], [2]	HTRICLRn	<p>Channel n Half Transfer Interrupt Status Clear (n = 0 ~ 5)</p> <p>0: No Operation 1: Clear the corresponding HTISTAn bit in the PDMAISR register</p> <p>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.</p>

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 corresponding 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, 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.
[26], [21], [16], [11], [6], [1]	BEIE _n	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]	GEIE _n	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.

27 Extend Bus Interface (EBI)

Introduction

The external bus interface is able to access external parallel interface devices such as SRAM, Flash and LCD modules. The interface is memory mapped into the internal address bus of the CPU. The data and address lines can be multiplexed in order to reduce the number of pins required to connect to external devices. The bus read/write timing can be adjusted to meet the timing specifications of the external devices. Note that the interface only supports asynchronous 8 or 16-bit bus interfaces.

Features

- Programmable interface for various memory types
 - Asynchronous static random access memory – SRAM
 - Read-only memory – ROM
 - NOR Flash memory
 - 8-bit or 16-bit parallel bus CPU interface device
- Translates AHB transactions into appropriate external device protocol
- Up to 4 memory bank regions and independent chip select control for each memory bank
- Programmable timings to support a wide range of devices
 - Programmable wait states
 - Programmable bus turnaround cycles
 - Programmable output enable and write enable cycles extension
 - Programmable active high or low setting of interface control signal
- Automatic translation when AHB transaction width and external memory interface width is different
- Write buffer to decrease stalling of the AHB write burst transactions
- Supports multiplexed and non-multiplexed address and data line configurations
 - Up to 21 address lines
 - 8-bit or 16-bit data bus width

Function Descriptions

An overview of the EBI module is shown in Figure 186. The EBI enables internal CPU and other bus matrix master peripherals to access external memories or devices. The EBI automatically translates the internal AHB transactions into the external device protocol. In particular, if the selected external memory is 16 or 8 bits width, then 32-bit wide transactions on the AHB are auto split into consecutive 16 or 8-bit accesses.

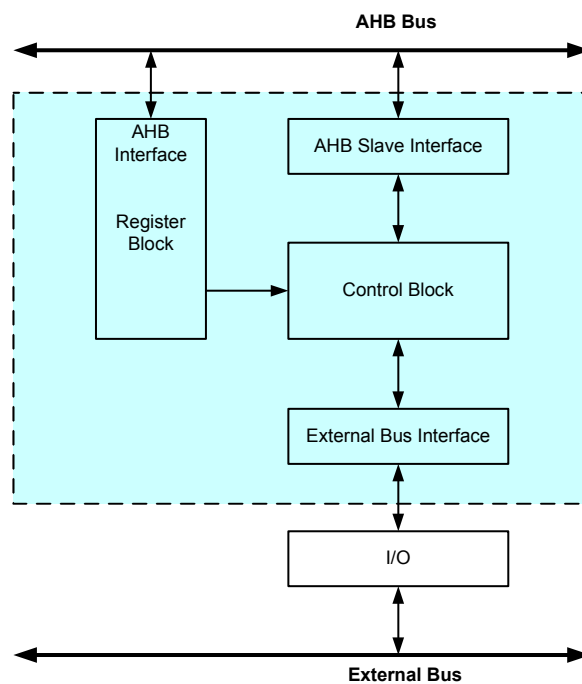


Figure 186. EBI Block Diagram

The EBI supports multiplexed and non-multiplexed addressing modes. The non-multiplexed addressing mode can be operated more efficiently and faster but it requires a higher number of pins. The multiplexed addressing modes are slower and require an external address latch device and a lower number of pins. The functionality of the 16 EBI_AD pins depends on what kind of the multiplexed addressing mode is used. They are used for both address and data in the multiplexed modes. Also for the non-multiplexed 8-bit address mode, both the address and data fit into these 16 EBI_AD pins. If more address bits or data bits are needed, an external latch can be used to support up to 20-bit addresses or 16-bit data in the multiplexed addressing modes using only the 16 EBI_AD pins. Furthermore, independent of the addressing mode, up to 21 non-multiplexed address lines can be enabled on the EBI_A pin connections. The detailed operation in the supported modes is presented in the following sections. The AHB clock (HCLK) is the reference clock for the EBI.

Non-multiplexed 8-bit Data 8-bit Address Mode

In this mode, 8-bit address and 8-bit data is supported. The address is located on the higher 8 bits of the EBI_AD lines and the data uses the lower 8 bits. This mode is set by programming the MODE field in the EBICR register to D8A8. Read and write timing in the 8-bit mode are shown in Figure 187 and Figure 188.

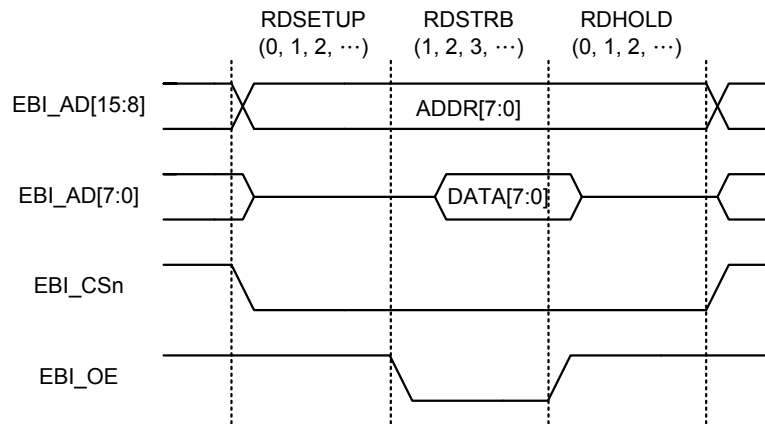


Figure 187. EBI Non-multiplexed 8-bit Data, 8-bit Address Read Operation

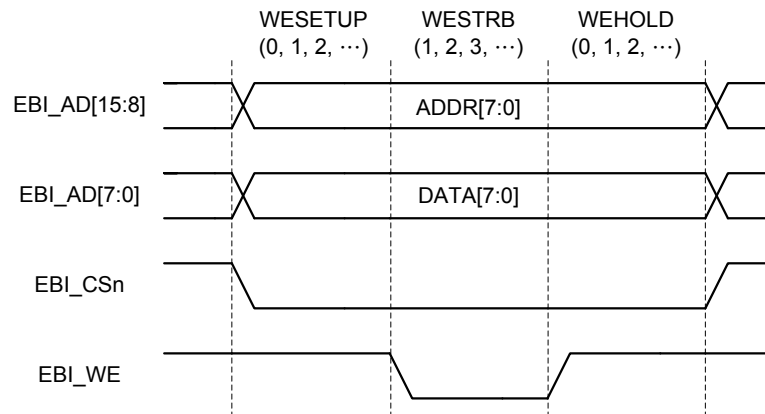


Figure 188. EBI Non-multiplexed 8-bit Data, 8-bit Address Write Operation

Non-multiplexed 16-bit Data N-bit Address Mode

In this non-multiplexed mode 16-bit data is provided on the 16 EBI_AD lines. The addresses are provided on the EBI_A lines. This mode is set by programming the MODE field in the EBICR register to D16. Read and write signals are shown in Figure 189 and Figure 190 for the case in which N address lines on EBI_A have been enabled.

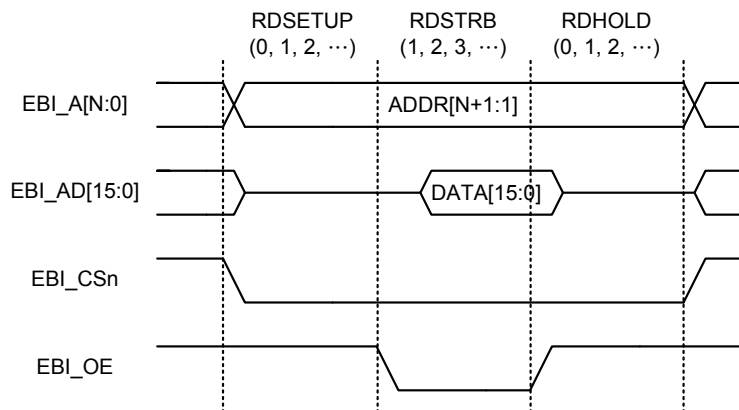


Figure 189. EBI Non-multiplexed 16-bit Data, N-bit Address Read Operation

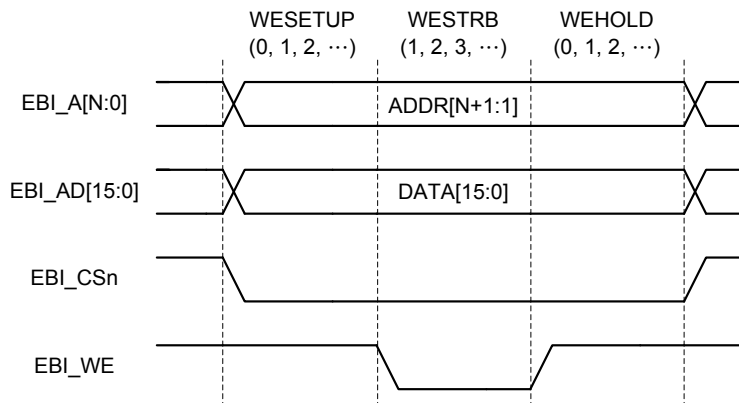


Figure 190. EBI Non-multiplexed 16-bit Data, N-bit Address Write Operation

Since the internal AHB address (HADDR) is a byte (8-bit) address whereas the 16-bit width of external device is addressed in words (16-bit), the address actually issued to the external device varies according to the data width as shown in the following table.

Memory width	Data address issued to the EBI
8-bit	HADDR[N:0] → EBI_A[N:0]
16-bit	HADDR[N+1:1] → EBI_A[N:0]

In case of a 16-bit external device width, the EBI will internally use HADDR[N+1:1] to generate the address EBI_A[N:0] for external device. Whatever the external memory width (16-bit or 8-bit), EBI_A[0] should be connected to external device address A[0].

Multiplexed 16-bit Data, 16-bit Address Mode

In this mode, 16-bit address and 16-bit data is supported, but the utilization of an external latch and an extra signal EBI_ALE is required. The 16-bit address and 16-bit data bits are multiplexed on the EBI_AD pins. An EBI address latch setup diagram is shown in Figure 191. This mode is set by programming the MODE field in the EBICR register to D16A16ALE.

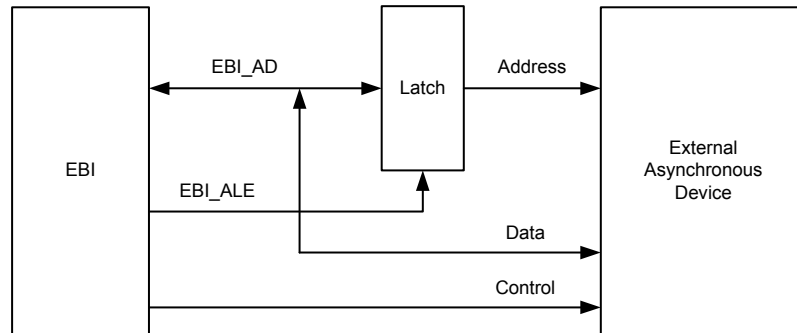


Figure 191. An EBI Address Latch Setup Diagram

At the start of the transaction the address is output on the EBI_AD lines. The external address latch is controlled by the EBI_ALE signal and stores the address. Then the data is read or written according to operation. Read and write signals are shown in Figure 192 and Figure 193.

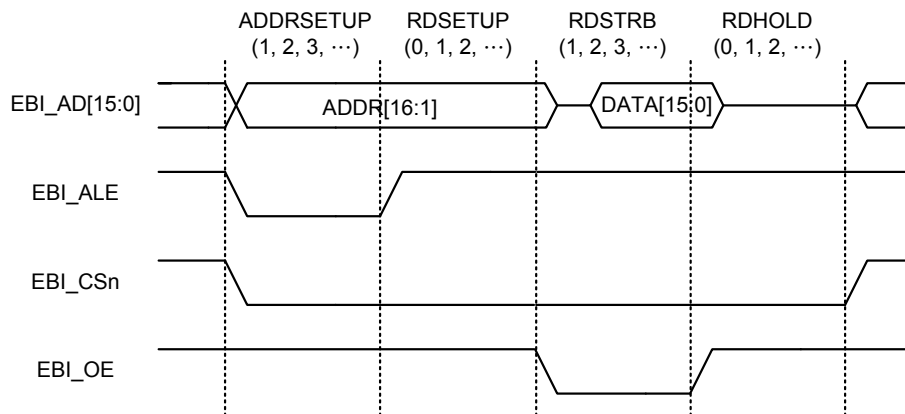


Figure 192. EBI Multiplexed 16-bit Data, 16-bit Address Read Operation

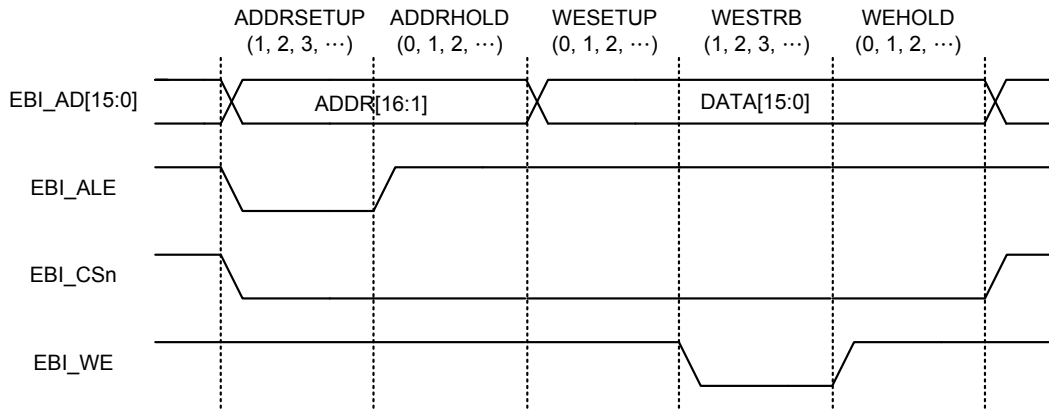


Figure 193. EBI Multiplexed 16-bit Data, 16-bit Address Write Operation

Multiplexed 8-bit Data, 20-bit Address Mode

This mode allows 20-bit address with 8-bit data multiplexed on the EBI_AD[15:0] lines to reduce the pins utilization and uses the EBI_ALE signal to decode 8-bit data and 20-bit address. The upper 8 bits of the EBI_AD lines (EBI_AD[15:8]) are consecutively used for the highest 4 bits and the lowest 8 bits of the address. The lower 8 bits of the EBI_AD lines (EBI_AD[7:0]) are used for the middle 8 address bits and 8-bit data. This mode is set by programming the MODE field in the EBICR register to D8A24ALE. Read and write signals are shown in Figure 194 and Figure 195 respectively.

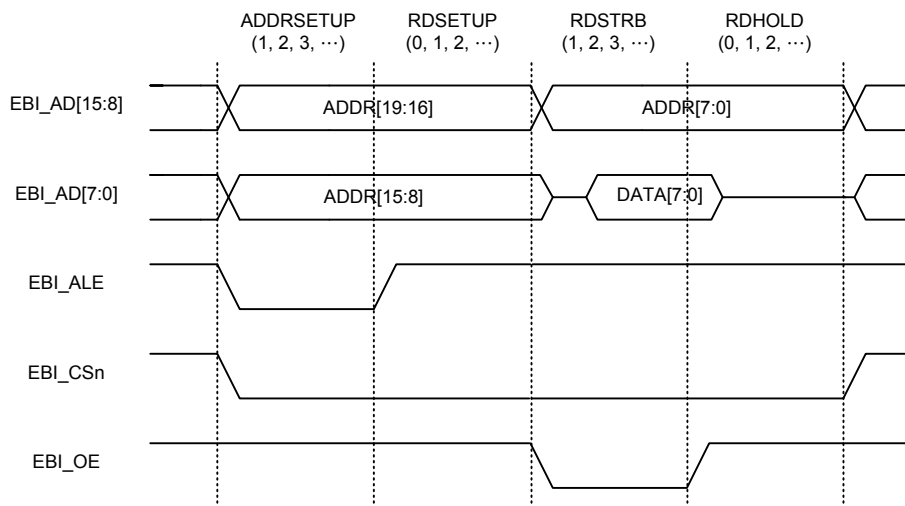


Figure 194. EBI Multiplexed 8-bit Data, 20-bit Address Read Operation

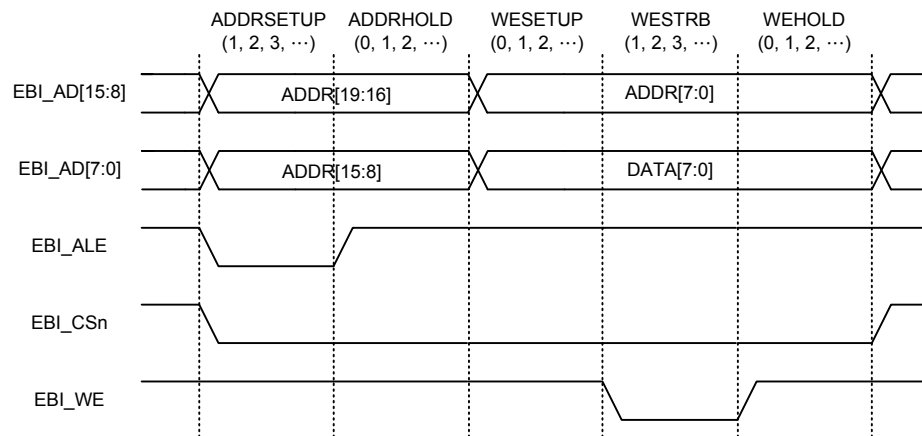


Figure 195. EBI Multiplexed 8-bit Data, 20-bit Address Write Operation

Write Buffer and EBI Status

The EBI has a 32-bit wide write buffer. The write buffer can be used to limit stalling of an AHB write burst transaction which comes from the CPU or PDMA to a potentially slow external device.

The EBIBUSY status bit in the EBISR register indicates whether an AHB transaction is still active in the EBI or not. When performing an AHB read or write, the EBIBUSY bit stays 1 until the required transaction(s) with the external device has finished.

Bus Turn-around and Idle Cycles

The EBI_AD lines can be driven by either the EBI or the external device depending on the cycle state of EBI bus. The RDHOLD timing parameter is for the bus turn-around time and should be programmed to ensure enough time for the characteristics of an external device. The default setting for the EBI is to insert an IDLE cycle between EBI transactions to the same bank. The IDLE cycle insertion is shown for two back-to-back read transactions in Figure 196. For cases where the IDLE state can also provide the required bus turn-around time, the RDHOLD parameter can be programmed to 0. For increased EBI access performance, the automatic IDLE state insertion can be disabled by setting the NOIDLEn (n = 0 ~ 3) bits in the EBICR register to 1. This example is shown in Figure 197 for two back-to-back reads in a non-multiplexed address mode.

An IDLE cycle will automatically be inserted for the following cases:

- Between two external device transactions to the same bank when the NOIDLE bit is 0.
- Between two external device transactions to different banks.
- Between a read and a subsequent write on the EBI_AD lines.
- When no request for an external transaction is available in the EBI.

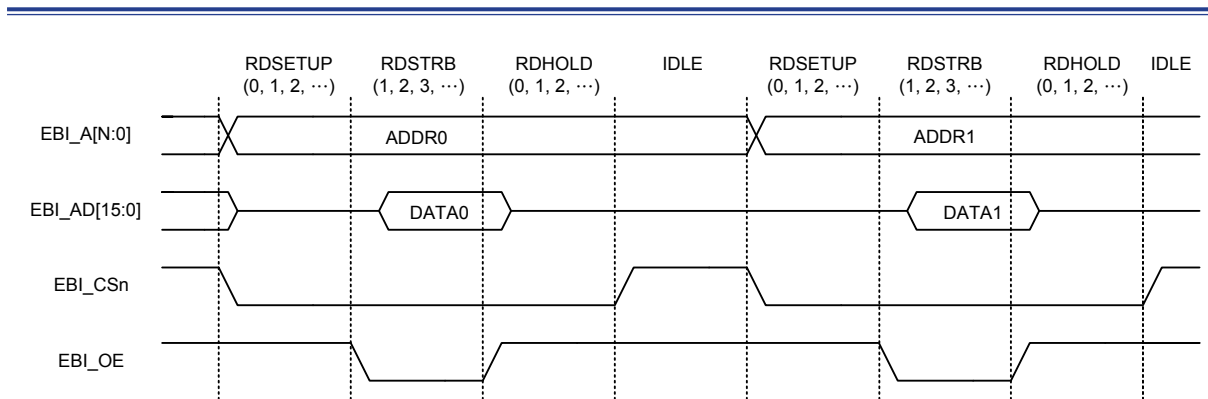


Figure 196. EBI Inserts an IDLE Cycle between Transactions in the Same Bank (NOIDLE = 0)

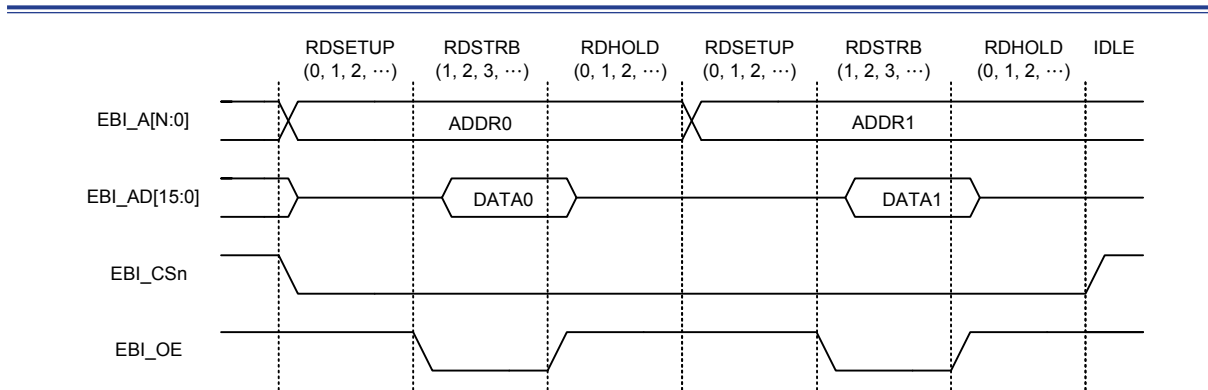


Figure 197. EBI De-asserts an IDLE Cycle between Transactions in the Same Bank (NOIDLE = 1)

AHB Transaction Width Conversion

The mapping of AHB transactions to an external device depends on the data width of the external device as the byte lanes of the external device are supported in the device. The Table 70 shows the EBI mapping of AHB transactions to external device transactions. The EBI will automatically translate the different AHB transaction width to external device transactions which matches the external bus capabilities of the device.

- If the AHB master (CPU or PDMA) transaction width is larger than the external bus transaction width. The EBI will split and translate the AHB transaction into consecutive multiple external transactions which have consecutively incrementing the address and start with the least significant data from AHB transaction.
- If the AHB master (CPU or PDMA) transaction width is smaller than the external bus transaction width. The EBI will read the data according to the full data bus width of the external device and ignore the superfluous data. For write operation the EBI will automatically perform a read-modify-write sequence to write the data.

Table 70. EBI Maps AHB Transactions Width to External Device Transactions.

AHB Transaction	8-bit External Device Transaction	16-bit External Device Transaction
8-bit read	1 x 8-bit read	1 x 16-bit read (EBI ignore the superfluous data)
16-bit read	2 x 8-bit read	1 x 16-bit read
32-bit read	4 x 8-bit read	2 x 16-bit read
8-bit write	1 x 8-bit write	1 x 16-bit read; 1 x 16-bit write (EBI read-modify-write)
16-bit write	2 x 8-bit write	1 x 16-bit write
32-bit write	4 x 8-bit write	2 x 16-bit write

Table 71. EBI Maps AHB Transactions Width to External Device Transactions Width

External Bus Width	Access from AHB Master		Access to External Bus Interface (EBI)		
	Access type	Address HADDR[1:0] ^{Note}	Access split	Output value from EBI_A[1:0]	Valid data at EBI_AD[15:0]
8-bit	Byte (8-bit)	0b00	No split	0b00	EBI_AD[7:0]
		0b01	No split	0b01	
		0b10	No split	0b10	
		0b11	No split	0b11	
	Half-word (16-bit)	0b00	1/2 access	0b00	
			2/2 access	0b01	
	Half-word (16-bit)	0b10	1/2 access	0b10	
			2/2 access	0b11	
	Word (32-bit)	0b00	1/4 access	0b00	
			2/4 access	0b01	
3/4 access			0b10		
4/4 access			0b11		
16-bit	Byte (8-bit)	0b00	No split	0bx0	EBI_AD[7:0]
		0b01	No split	0bx0	EBI_AD[15:8]
		0b10	No split	0bx1	EBI_AD[7:0]
		0b11	No split	0bx1	EBI_AD[15:8]
	Half-word (16-bit)	0b00	No split	0bx0	EBI_AD[15:0]
			No split	0bx1	EBI_AD[15:0]
	Word (32-bit)	0b00	1/2 access	0bx0	EBI_AD[15:0]
			2/2 access	0bx1	EBI_AD[15:0]

Note: HADDR is AHB bus address input.

EBI Bank Access

The EBI is split into 4 different address regions and each owns an individual EBI_CS_n line. When accessing one of the memory regions, the corresponding EBI_CS_n line is asserted. This way up to 4 separate devices can share the EBI lines and be identified by the EBI_CS_n line. Each bank can individually be enabled or disabled in the EBICR register. The data space of each bank can be accessed up to 4MB and is shown as Figure 198.

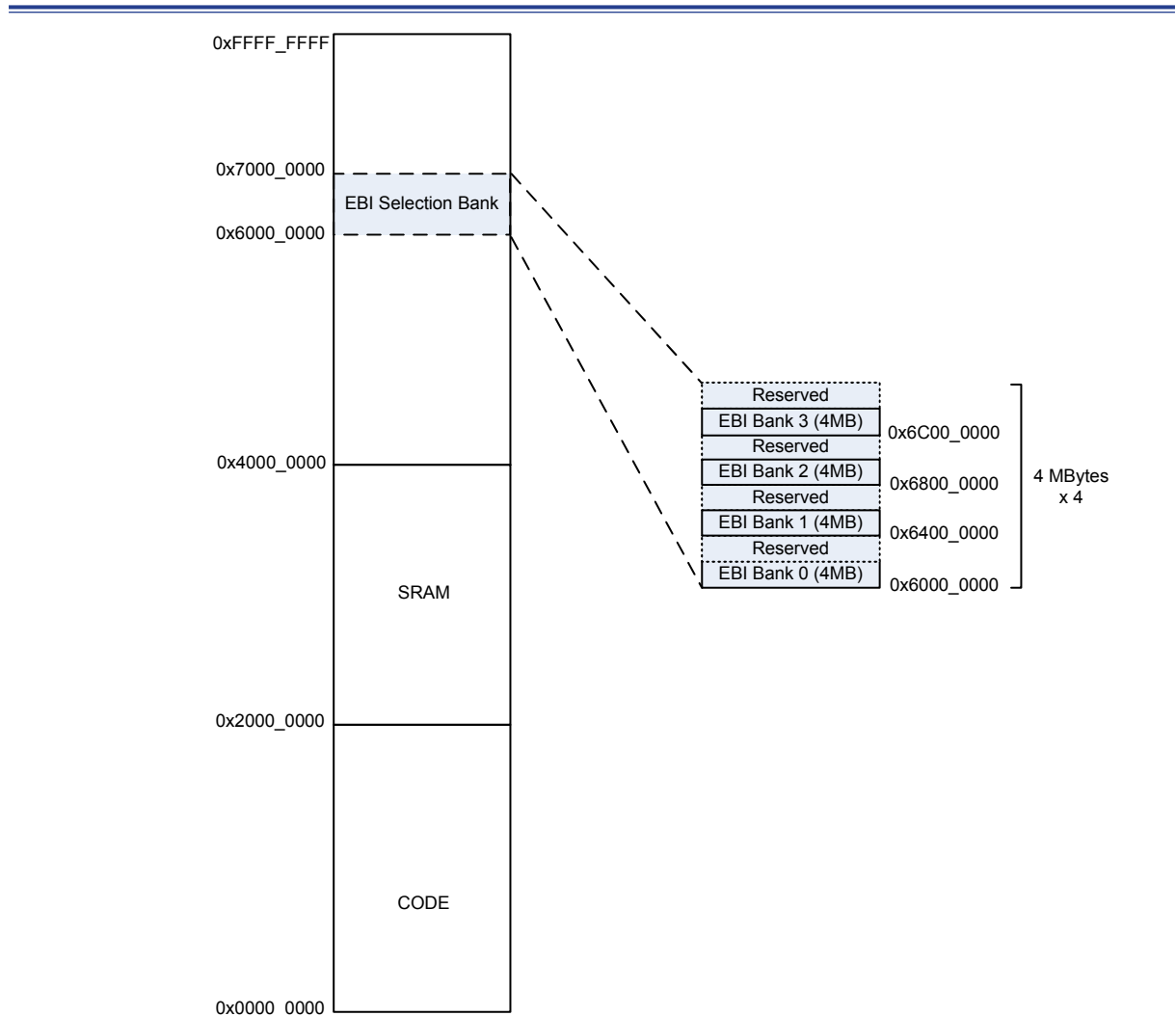


Figure 198. EBI Bank Memory Map

PDMA Request

The EBI only supports a software trigger for activating a PDMA service.

Register Map

The following table shows the EBI register and reset value.

Table 72. EBI Register Map

Register	Offset	Description	Reset Value
EBI Base Address = 0x4009_8000			
EBICR	0x000	EBI Control Register	0x0000_0000
EBISR	0x008	EBI Status Register	0x0000_0000
EBIATR	0x010	EBI Address Timing Register	0x0000_0707
EBIRTR	0x014	EBI Read Timing Register	0x0007_1F07
EBIWTR	0x018	EBI Write Timing Register	0x0007_1F07
EBIPR	0x01C	EBI Parity Register	0x0000_0000

Register Descriptions

EBI Control Register – EBICR

This register specifies the control setting for EBI bank.

Offset: 0x000

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24						
	IDLET				Reserved									
Type/Reset	RW	0	RW	0	RW	0	RW	0						
	23	22	21	20	19	18	17	16						
	Reserved													
Type/Reset														
	15	14	13	12	11	10	9	8						
	NOIDLE3	NOIDLE2	NOIDLE1	NOIDLE0	BANKEN3	BANKEN2	BANKEN1	BANKEN0						
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0						
	Mode3		Mode2		Mode1		Mode0							
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[31:28]	IDLET	IDLE Time Sets the number of cycles between EBI transactions. If set to 0, one cycle is inserted by the hardware. The cycle unit is based on the HCLK clock period.
[15]	NOIDLE3	No IDLE 3 0: Enable IDLE state insertion 1: Disable IDLE state insertion Enable or disable the insertion of an idle state between transactions for bank 3.

Bits	Field	Descriptions
[14]	NOIDLE2	No IDLE 2 0: Enable IDLE state insertion 1: Disable IDLE state insertion Enable or disable the insertion of an idle state between transactions for bank 2.
[13]	NOIDLE1	No IDLE 1 0: Enable IDLE state insertion 1: Disable IDLE state insertion Enable or disable the insertion of an idle state between transactions for bank 1.
[12]	NOIDLE0	No IDLE 0 0: Enable IDLE state insertion 1: Disable IDLE state insertion Enable or disable the insertion of an idle state between transactions for bank 0.
[11]	BANKEN3	Bank 3 Enable 0: Disable 1: Enable This bit enables or disables bank 3.
[10]	BANKEN2	Bank 2 Enable 0: Disable 1: Enable This bit enables or disables bank 2.
[9]	BANKEN1	Bank 1 Enable 0: Disable 1: Enable This bit enables or disables bank 1.
[8]	BANKEN0	Bank 0 Enable 0: Disable 1: Enable This bit enables or disables bank 0.
[7:6]	MODE3	Set EBI bank 3 access mode 00: D8A8 01: D16A16ALE 10: D8A24ALE 11: D16
[5:4]	MODE2	Set EBI bank 2 access mode 00: D8A8 01: D16A16ALE 10: D8A24ALE 11: D16
[3:2]	MODE1	Set EBI bank 1 access mode 00: D8A8 01: D16A16ALE 10: D8A24ALE 11: D16
[1:0]	MODE0	Set EBI bank 0 access mode 00: D8A8 01: D16A16ALE 10: D8A24ALE 11: D16

EBI Status Register – EBISR

This register indicates the EBI 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	Reserved							EBISMRST			
								RW	0		
	7	6	5	4	3	2	1	0			
Type/Reset	Reserved							EBIBUSY			
								RO	0		

Bits	Field	Descriptions
[8]	EBISMRST	EBI State Machine Reset 0: Normal 1: Reset EBI state machine Write a “1” to reset the EBI internal state machine to its initial state and keep the original register settings.
[0]	EBIBUSY	EBI BUSY 0: EBI is idle 1: EBI is busy Indicates the EBI is busy with an AHB transaction.

EBI Address Timing Register – EBIATR

This register specifies the address timing setting for each bank.

Offset: 0x010

Reset value: 0x0000_0707

	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	1	RW	1	RW	1
	7	6	5	4	3	2	1	0			
Type/Reset	Reserved					RW	1	RW	1	RW	1

Bits	Field	Descriptions
[10:8]	ADDRHOLD	Address Hold Time Sets the number of cycles for which the address should be held on the EBI_AD bus after the EBI_ALE signal is asserted. This field is allowed to be set to 0.
[2:0]	ADDRSETUP	Address Setup Time Sets the number of cycles for which the address is driven onto the EBI_AD bus before the EBI_ALE signal is asserted. If set to 0, one cycle is inserted by the hardware. The cycle unit is based on an HCLK clock period.

EBI Read Timing Register – EBIRTR

This register specifies the read timing setting for each bank

Offset: 0x014

Reset value: 0x0007_1F07

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

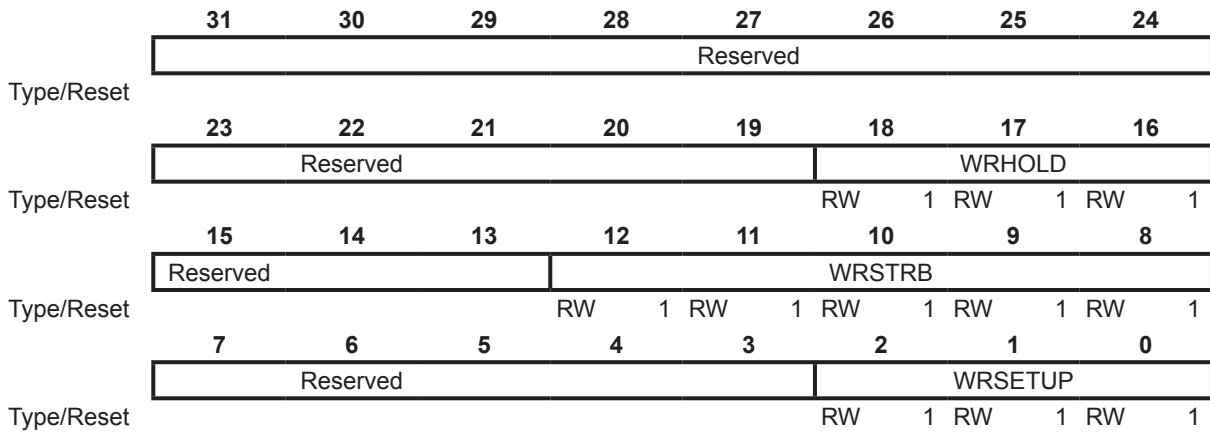
Bits	Field	Descriptions
[18:16]	RDHOLD	Read Hold Time Sets the number of cycles for which the EBI_CS _n signal is held active after the EBI_OE signal is de-asserted. This interval is used for bus turnaround.
[12:8]	RDSTRB	Read Strobe Time Sets the number of cycles in which the data will be read when the EBI_OE is held active. After the specified number of cycles, the data will be read. If set to 0, one cycle is inserted by the hardware. The cycle unit is based on an HCLK clock period.
[2:0]	RDSETUP	Read Setup Time Sets the number of cycles for which the address is setup before the EBI_OE signal is asserted. The cycle unit is based on an HCLK clock period.

EBI Write Timing Register – EBIWTR

This register specifies the write timing setting for each bank.

Offset: 0x018

Reset value: 0x0007_1F07



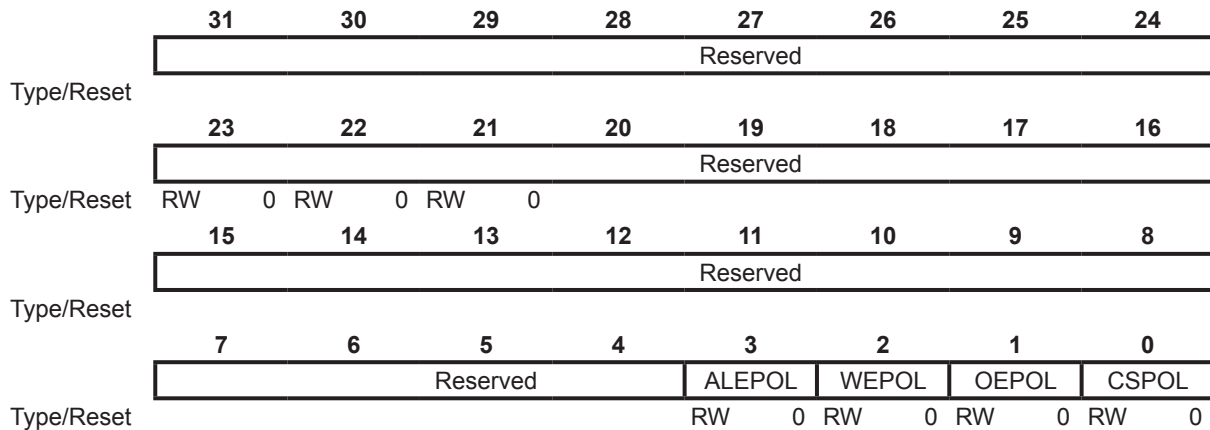
Bits	Field	Descriptions
[18:16]	WEHOLD	Write Hold Time Sets the number of cycles for which the EBI_CS _n signal is held active after the EBI_WE signal is de-asserted.
[12:8]	WESTRB	Write Strobe Time Sets the number of cycles in which the data will be written when the EBI_WE signal is held active. If set to 0, one cycle is inserted by the hardware. The cycle unit is based on an HCLK clock period.
[2:0]	WESETUP	Write Setup Time Sets the number of cycles for which the data is setup before the EBI_WE signal is asserted. The cycle unit is based on an HCLK clock period.

EBI Parity Register – EBIPR

This register specifies the polarity of the EBI control signal for each bank.

Offset: 0x01C

Reset value: 0x0000_0000



Bits	Field	Descriptions
[3]	ALEPOL	Address Latch Polarity 0: EBI_ALE is active low 1: EBI_ALE is active high Set the polarity of the EBI_ALE line
[2]	WEPOL	Write Enable Polarity 0: EBI_WE is active low 1: EBI_WE is active high Set the polarity of the EBI_WE line.
[1]	OEPOL	Output Enable Polarity 0: EBI_OE is active low 1: EBI_OE is active high Set the polarity of the EBI_OE line.
[0]	CSPOL	Chip Selection Polarity 0: EBI_CS is active low 1: EBI_CS is active high Set the polarity of the EBI_CS _n line.

28 Inter-IC Sound (I²S)

Introduction

The I²S is a synchronous communication interface that can be used as a master or slave to exchange data with other audio peripherals, such as ADCs or DACs. The I²S supports a variety of data formats. In addition to the stereo I²S-justified, Left-justified and Right-justified modes, there are mono PCM modes with 8/16/24/32-bit sample size. When the I²S operates in the master mode, then when using the fractional divider, it can provide an accurate sampling frequency output and support the rate control function and fine-tuning of the output frequency to avoid system problems caused by the cumulative frequency error between different devices.

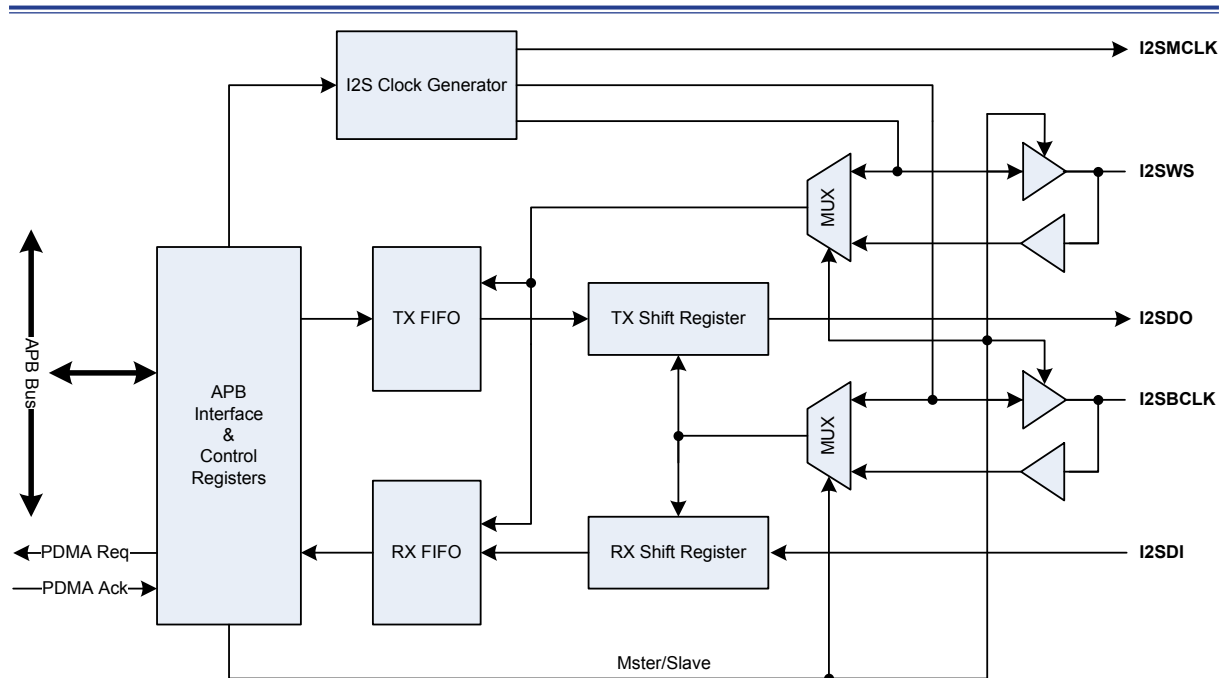


Figure 199. I²S Block Diagram

Features

- Master or slave mode
- Mono and stereo
- I²S-justified, Left-justified, and Right-justified mode
- 8/16/24/32-bit sample size with 32-bit channel extended
- 8 × 32-bit TX & RX FIFO with PDMA supported
- 8-bit Fractional Clock Divider with rate control

Functional Description

I²S Master and Slave Mode

The I²S can operate in slave or master mode. Within the I²S module the difference between these modes lies in the word select (WS) signal which determines the timing of data transmissions.

- In the master mode, the word select signal is generated internally by a clock rate generator.
- In the slave mode, the word select signal is input on the I²S_WS pin.
- When an I²S bus is enabled, the word select, bit clock signals are sent continuously by the bus master.
- The mute control bit will place the transmit channel in a mute condition. When the mute mode is enabled, the transmit channel FIFO operates normally, but the output data stream is discarded and replaced by zeroes. This bit does not affect the receive channel so data reception can occur normally.

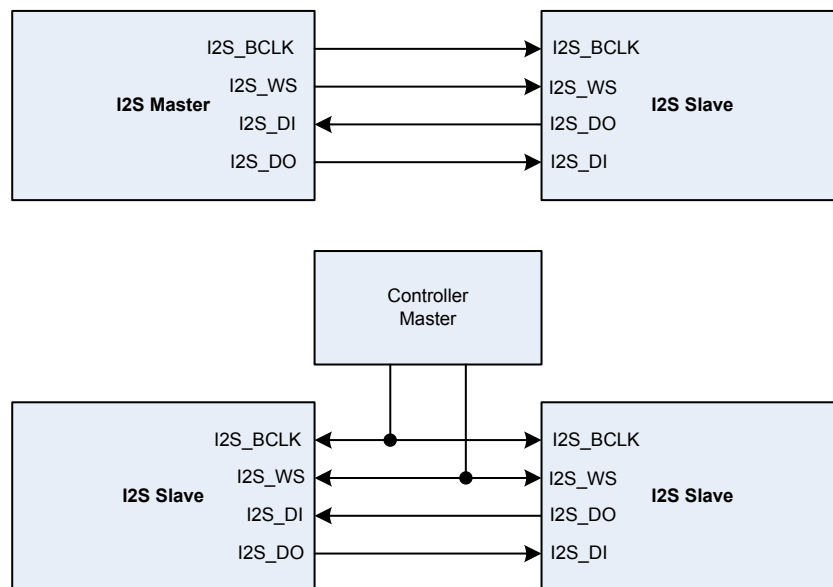


Figure 200. Simple I²S Master/Slave Configuration

I²S Clock Rate Generator

The main (I²S_MCLK) and bit clock (I²S_BCLK) rates for the I²S are determined by the values in the I2SCDR register. The required I²S bit clock rate setting depends on the desired audio sample rate desired, the format (stereo/mono) used, and the data size. The main clock rate (I²S_MCLK) is generated using a fractional rate divider which is a divided down PCLK frequency of the I²S. Values of the numerator (X) and the denominator (Y) must be chosen to produce a frequency twice that of the main clock (I²S_MCLK). The output frequency of the divider is divided by 2 in order to get the duty cycle of the output clock more even. The I²S clock generator block diagram is shown in Figure 201. The equation for the fractional rate divider is:

$$I^2S_MCLK = 1/2 \times PCLK \times (X/Y), \text{ and } X/Y \leq 1, X = 1 \sim 255, Y = 1 \sim 255$$

$$I^2S_BCLK = I^2S_MCLK / (N+1), N = 0 \sim 255$$

Because the fractional rate divider is a fully digital implementation function, the divider output clock transitions are synchronous with the input source clock. Therefore, the fractional rate divider will generate some jitter with some divider settings. Users should make note of this phenomenon when choosing the X and Y setup values. It is possible to avoid jitter entirely by choosing fractions such that X divides evenly into Y. For example, 2/4, 2/6, 3/9, etc.

The tables below show the recommended setup values to reduce clock jitter for different source clocks and sample rates.

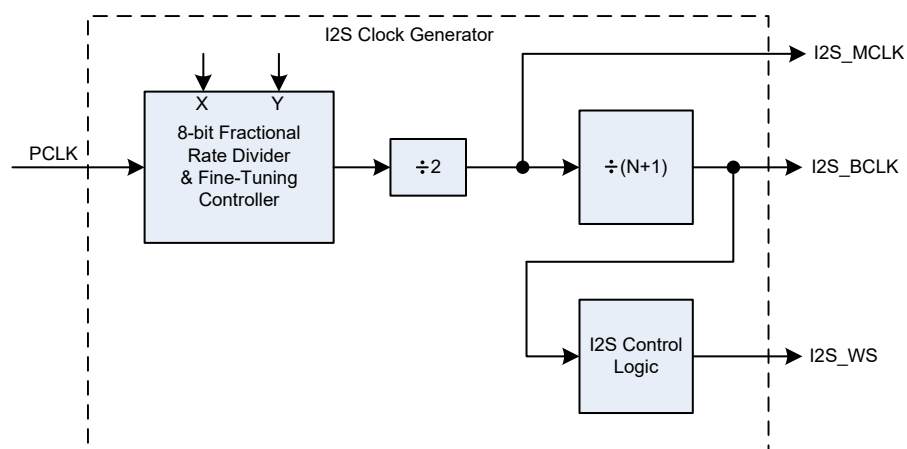


Figure 201. I²S Clock Generator Diagram

Table 73. Recommend FS List @ 8 MHz PCLK

Fs (Hz)	512 Fs		384 Fs		256 Fs		192 Fs		128 Fs		64 Fs	
	X	Y	X	Y	X	Y	X	Y	X	Y	X	Y
8,000	—	—	96	125	64	125	48	125	32	125	16	125
11,025	—	—	—	—	170	241	118	223	90	255	42	238
12,000	—	—	—	—	96	125	72	125	48	125	24	125
16,000	—	—	—	—	—	—	96	125	64	125	32	125
22,050	—	—	—	—	—	—	—	—	170	241	90	255
24,000	—	—	—	—	—	—	—	—	96	125	48	125
32,000	—	—	—	—	—	—	—	—	—	—	64	125
44,100	—	—	—	—	—	—	—	—	—	—	170	241
48,000	—	—	—	—	—	—	—	—	—	—	96	125
96,000	—	—	—	—	—	—	—	—	—	—	—	—
192,000	—	—	—	—	—	—	—	—	—	—	—	—

Table 74. Recommend FS List @ 48 MHz PCLK

Fs (Hz)	512 Fs		384 Fs		256 Fs		192 Fs		128 Fs		64 Fs	
	X	Y	X	Y	X	Y	X	Y	X	Y	X	Y
8,000	36	211	16	125	18	211	8	125	10	234	2	94
11,025	4	17	6	34	2	17	6	68	2	34	2	68
12,000	32	125	24	125	16	125	12	125	8	125	4	125
16,000	86	252	32	125	36	211	16	125	18	211	10	234
22,050	8	17	6	17	4	17	6	34	2	17	2	34
24,000	64	125	48	125	32	125	24	125	16	125	8	125
32,000	142	208	64	125	86	252	32	125	36	211	18	211
44,100	238	253	170	241	8	17	6	17	4	17	2	17
48,000	—	—	96	125	64	125	48	125	32	125	16	125
96,000	—	—	—	—	—	—	96	125	64	125	32	125
192,000	—	—	—	—	—	—	—	—	—	—	64	125

I²S Interface Format

I²S-justified Stereo Mode

The standard I²S-justified mode is where the Most Significant Bit (MSB) of the stereo audio sample data is available on the second rising edge of the BCLK clock following a WS signal transition. In the stereo mode, a low WS state indicates left channel data and a high state indicates right channel data. Figure 202 and Figure 203 show the standard I²S-justified stereo mode format.

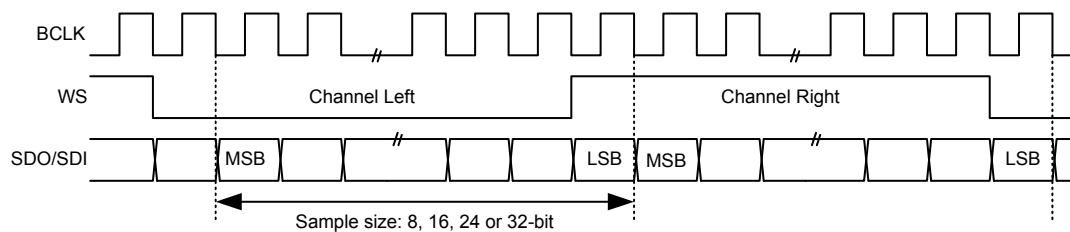


Figure 202. I²S-justified Stereo Mode Waveforms

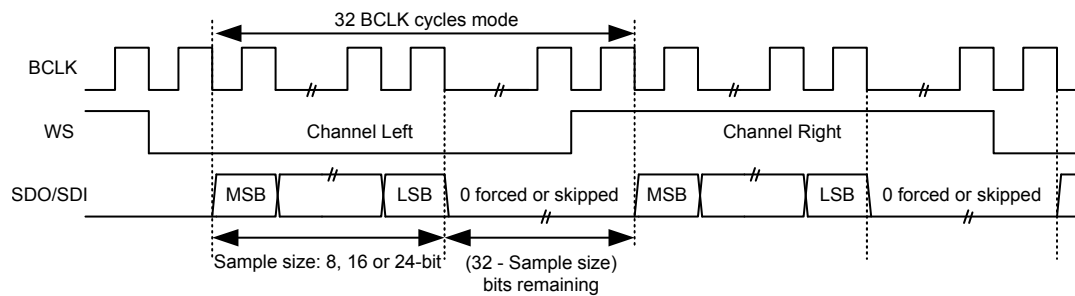


Figure 203. I²S-justified Stereo Mode Waveforms (32-bit Channel Enabled)

Left-justified Stereo Mode

Left-Justified mode is where the Most Significant Bit (MSB) of the stereo audio sample data is available on the first rising edge of BCLK following a WS transition. Figure 204 and Figure 205 are shown with a left I²S-justified stereo mode format.

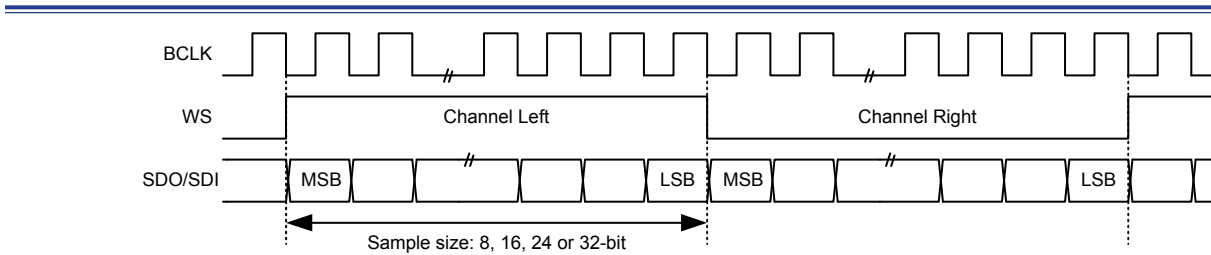


Figure 204. Left-justified Stereo Mode Waveforms

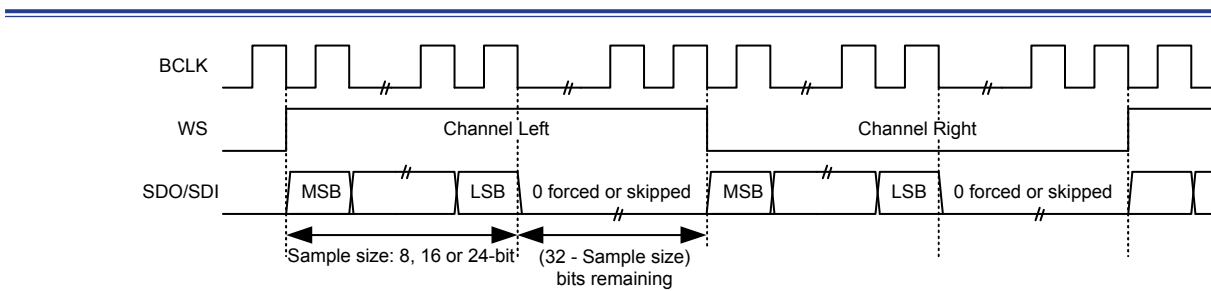


Figure 205. Left-justified Stereo Mode Waveforms (32-bit Channel Enabled)

Right-justified Stereo Mode

Right-Justified mode is where the Least Significant Bit (LSB) of the stereo audio sample data is available on the rising edge of BCLK preceding a WS transition and where the MSB is transmitted first. Figure 206 and Figure 207 show a right I²S-justified stereo mode format.

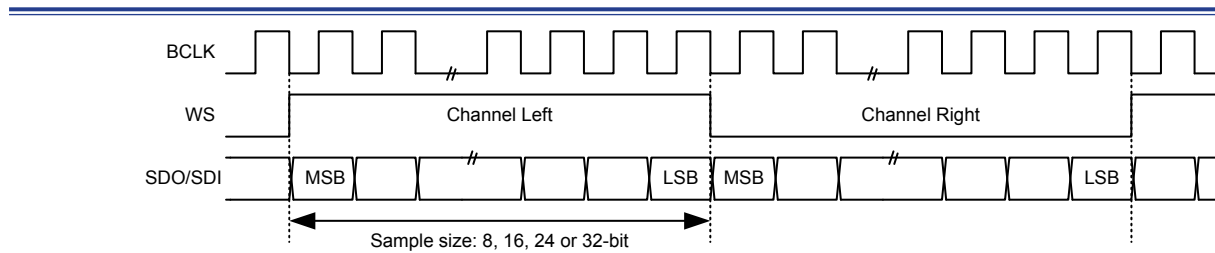


Figure 206. Right-justified Stereo Mode Waveforms

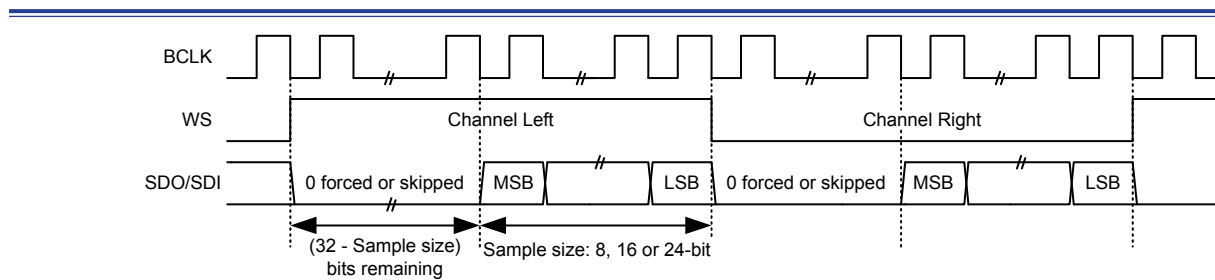


Figure 207. Right-justified Stereo Mode Waveforms (32-bit Channel Enabled)

Inter-IC Sound (I2S)

I²S-justified Mono Mode

In the I²S-justified mono mode, the Most Significant Bit (MSB) of the mono audio sample data is available on the second rising edge of the BCLK clock following a falling edge on the WS signal. Figure 208 and Figure 209 show an I²S-justified mono mode format.

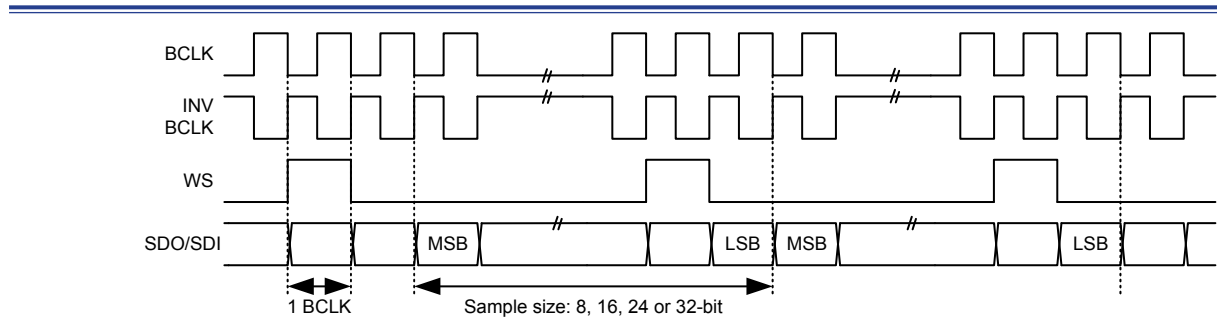


Figure 208. I²S-justified Mono Mode Waveforms

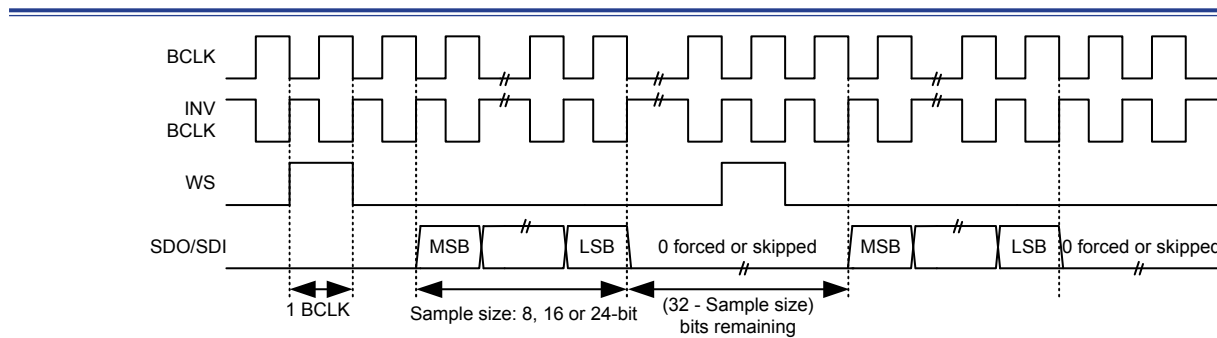


Figure 209. I²S-justified Mono Mode Waveforms (32-bit Channel Enabled)

Left-justified Mono Mode

In the left-justified mono mode, the Most Significant Bit (MSB) of the mono audio sample data is available on the first rising edge of the BCLK clock following a falling edge on the WS signal. Figure 210 and Figure 211 show a left-justified mono mode format.

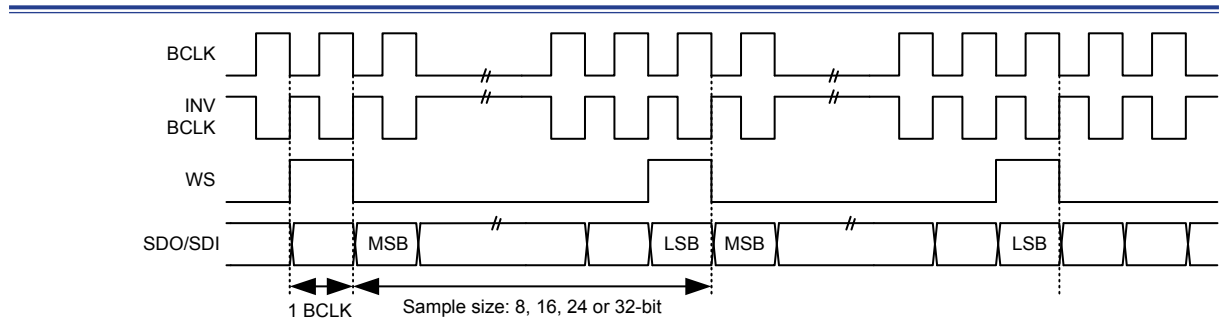


Figure 210. Left-justified Mono Mode Waveforms

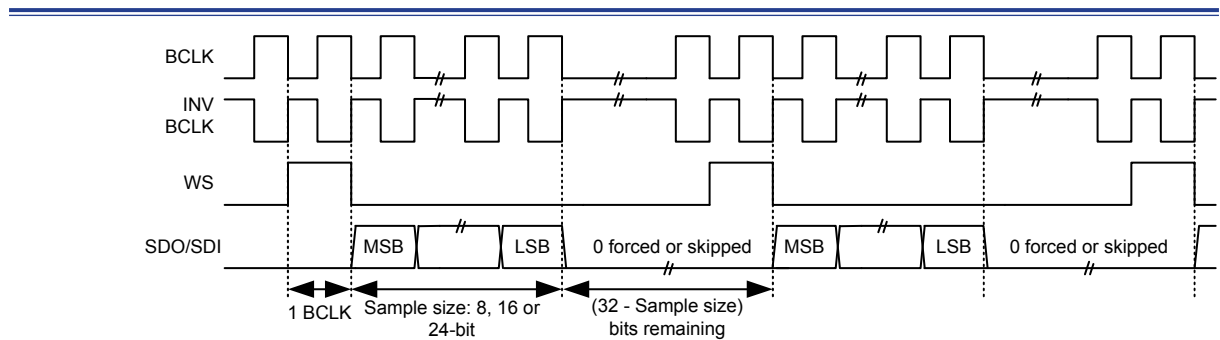


Figure 211. Left-justified Mono Mode Waveforms (32-bit Channel Enabled)

Right-justified Mono Mode

In the right-justified mono mode, the Least Significant Bit (LSB) of the mono audio sample data is available on the last rising edge of the BCLK clock preceding a rising edge on the WS signal. Figure 212 and Figure 213 show the right-justified mono mode format.

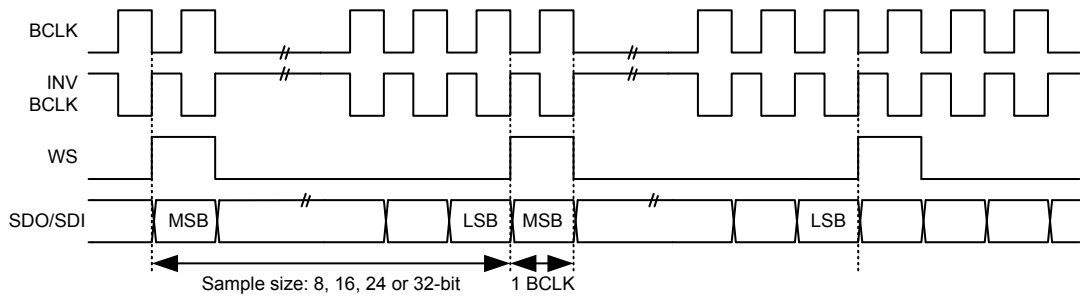


Figure 212. Right-justified Mono Mode Waveforms

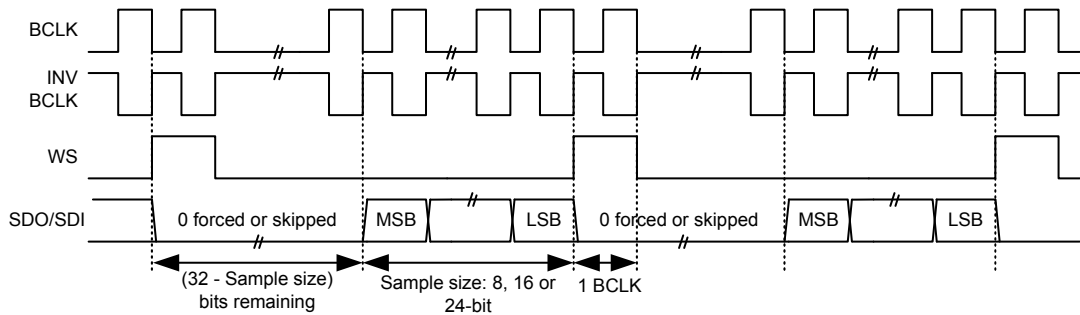


Figure 213. Right-justified Mono Mode Waveforms (32-bit Channel Enabled)

I²S-justified Repeat Mode

In the I²S-justified repeat mode, the Most Significant Bit (MSB) of the mono audio sample data is available on the second rising edge of the BCLK clock following a WS signal transition. In this mode the same data is transmitted twice, once when WS is low and again when WS is high. Figure 214 and Figure 215 show the I²S-justified repeat mode format.

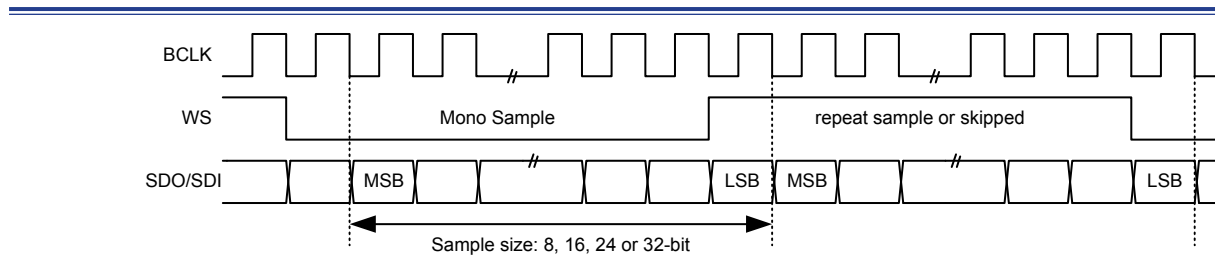


Figure 214. I²S-justified Repeat Mode Waveforms

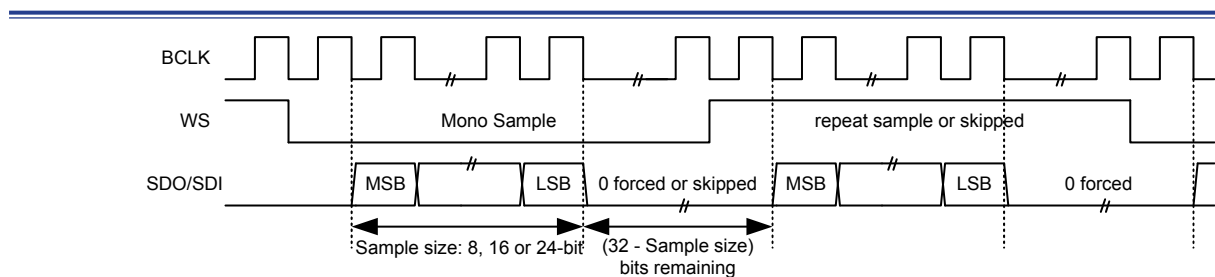


Figure 215. I²S-justified Repeat Mode Waveforms (32-bit Channel Enabled)

FIFO Control and Arrangement

The I²S handles audio data for transmission and reception and is performed via the FIFO controller. Each transmitted or received FIFO has a depth of 8 words (8 × 32-bit) and can buffer the data. The format is dependent upon the stereo/mono mode and sample size setting. The detailed FIFO data content format is shown in Figure 216. The FIFO controller consists of comparators which compare the current FIFO levels with configurable depth settings. The current level of the TX or RX FIFO status can be seen in the TXFS and RXFS fields of the I²S status register (I2SSR).

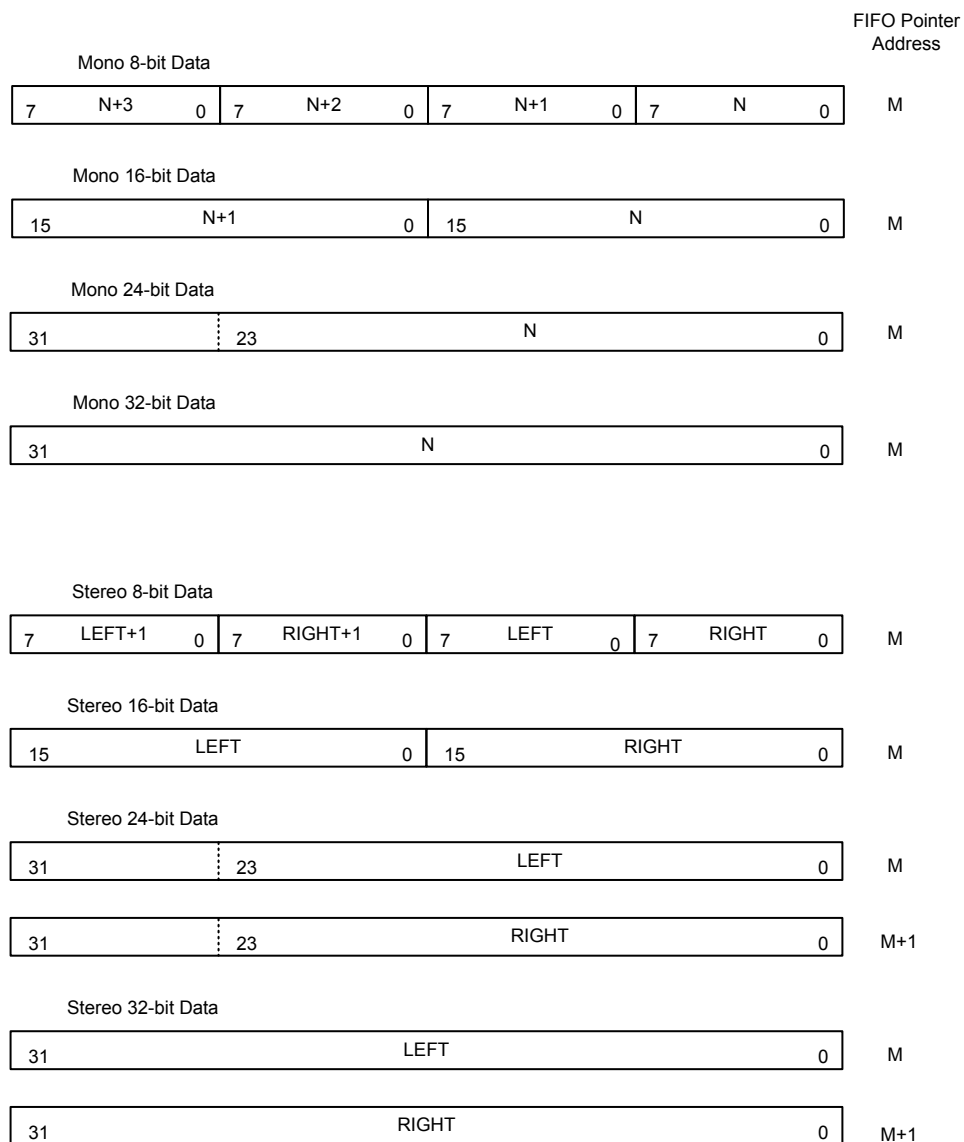


Figure 216. FIFO Data Content Arrangement for Various Modes

PDMA and Interrupt

When the level of received data in the RX FIFO is equal to or greater than the level defined by the RXFTLS field in the I²S FIFO control register (I2SFCR), the relative RXFTL flag will be set and then an I²S RX PDMA request will be generated. A CPU interrupt will be generated if the enable bit of the I²S RX PDMA request or the RX FIFO trigger level interrupt is asserted. When the level of transmitted data in the TX FIFO is equal to or less than the level defined by the TXFTLS field in the I²S FIFO control register (I2SFCR), the relative TXFTL flag will be set and a I²S TX PDMA request will be generated. A CPU interrupt will be generated if the enable bit of the I²S TX PDMA request or TX FIFO trigger level interrupt is asserted.

The I²S transmitter and receiver have separate PDMA requests and can be assigned to two different PDMA channels. When a PDMA request is enabled for the I²S transmitter (TXDMAEN = 1) then this will automatically request that data is transferred to the assigned I²S TX PDMA channel whenever TX FIFO space is available and TXFTL is active. When a PDMA request is enabled for the receiver (RXDMAEN = 1) then this will automatically request the data transfers to the I²S RX PDMA channel whenever data is present in the receive FIFO and when RXFTL is active.

Register Map

The following table shows the I²S registers and reset values.

Table 75. I²S Register Map

Register	Offset	Description	Reset Value
I2SCR	0x000	I ² S Control Register	0x0000_0000
I2SIER	0x004	I ² S Interrupt Enable Register	0x0000_0000
I2SCDR	0x008	I ² S Clock Divider Register	0x0000_0000
I2STXDR	0x00C	I ² S TX Data Register	0x0000_0000
I2SRXDR	0x010	I ² S RX Data Register	0x0000_0000
I2SFCR	0x014	I ² S FIFO Control Register	0x0000_0000
I2SSR	0x018	I ² S Status Register	0x0000_0809
I2SRCNTR	0x01C	I ² S Rate Counter Value Register	0x0000_0000

Register Descriptions

I²S Control Register – I2SCR

This register specifies the corresponding I²S function enable control.

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				MCKINV	BCKINV	RCSEL	RCEN				
Type/Reset								RW	0	RW	0	RW	0	RW	0				
15	14	13	12	11	10	9	8	CLKDEN	RXDMAEN	TXDMAEN	TXMUTE	CHANNEL	REPEAT	MCLKEN	BITEXT				
Type/Reset								RW	0	RW	0	RW	0	RW	0	RW	0	RW	0
7	6	5	4	3	2	1	0	FORMAT		SMPSIZE		MS	RXEN	TXEN	I2SEN				
Type/Reset								RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[19]	MCKINV	MCLK Inverse Enable 0: Disable 1: Enable
[18]	BCKINV	BCLK Inverse Enable 0: Disable 1: Enable
[17]	RCSEL	Rate Control Select (master only) 0: slower 1: faster
[16]	RCEN	Rate Control Enable (master only) 0: Disable 1: Enable
[15]	CLKDEN	Clock Divider Enable (master only) 0: Disable 1: Enable The clock divider can be used to generate the MCLK and BCLK clock of the I ² S interface for master mode.
[14]	RXDMAEN	RX PDMA Request Enable 0: Disable 1: Enable
[13]	TXDMAEN	TX PDMA Request Enable 0: Disable 1: Enable
[12]	TXMUTE	TX Mute Enable 0: Disable 1: Enable

Bits	Field	Descriptions
[11]	CHANNEL	Stereo or Mono 0: Stereo 1: Mono Note: This bit should be configured when I ² S is disabled.
[10]	REPEAT	Repeat Mode 0: Disable 1: Enable This mode is for I ² S-justified stereo configuration only, transmitting the mono data on both channels and receiving just the left channel data and ignoring the right. Enabling the repeat mode will reset the CHANNEL bit automatically. Note: This bit should be configured when the I ² S is disabled.
[9]	MCLKEN	MCLK Output Enable (master only) 0: Disable 1: Enable Note: This bit should be configured when the I ² S is disabled.
[8]	BITEXT	32-bit Channel Enable 0: Disable 1: Enable Setting this bit will force the channel size to 32-bits. If the sample size is 8/16/24-bits, the remaining bits will be forced to 0 in the TX and ignored in the RX. Note: This bit should be configured when the I ² S is disabled.
[7:6]	FORMAT	Data Format 00: I ² S-justified 01: Left-justified 10: Right-justified 11: reserved Note: This bit should be configured when the I ² S is disabled.
[5:4]	SMPSIZE	Sample Size 00: 8-bit 01: 16-bit 10: 24-bit 11: 32-bit Note: This bit should be configured when the I ² S is disabled.
[3]	MS	Master or Slave Mode 0: Master 1: Slave Note: This bit should be configured when the I ² S is disabled.
[2]	RXEN	RX Enable 0: Disable 1: Enable
[1]	TXEN	TX Enable 0: Disable 1: Enable
[0]	I2SEN	I ² S Enable 0: Disable 1: Enable

I²S Interrupt Enable Register – I2SIER

This register contains the corresponding I²S 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								
	15	14	13	12	11	10	9	8	
Type/Reset	Reserved								
	7	6	5	4	3	2	1	0	
Type/Reset	Reserved	RXOVIEN	RXUDIEN	RXFTLIEN	Reserved	TXOVIEN	TXUDIEN	TXFTLIEN	
		RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[6]	RXOVIEN	RX FIFO Overflow Interrupt Enable 0: Disable 1: Enable
[5]	RXUDIEN	RX FIFO Underflow Interrupt Enable 0: Disable 1: Enable
[4]	RXFTLIEN	RX FIFO Trigger Level Interrupt Enable 0: Disable 1: Enable
[2]	TXOVIEN	TX FIFO Overflow Interrupt Enable 0: Disable 1: Enable
[1]	TXUDIEN	TX FIFO Underflow Interrupt Enable 0: Disable 1: Enable
[0]	TXFTLIEN	TX FIFO Trigger Level Interrupt Enable 0: Disable 1: Enable

I²S Clock Divider Register – I2SCDR

This register specifies the I²S clock divider ratio.

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	N_DIV							
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	X_DIV							
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	Y_DIV							
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[23:16]	N_DIV	N divider for BCLK 0x00: divide 1 0x01: divide 2 ... 0xFF: divide 256 Note: This bit should be configured when the I2S is disabled.
[15:8]	X_DIV	X divider for MCLK (X = 1 ~ 255) && (X / Y ≤ 1) Note: This bit should be configured when the I2S is disabled.
[7:0]	Y_DIV	Y divider for MCLK (Y = 1 ~ 255) && (X / Y ≤ 1) Note: This bit should be configured when the I2S is disabled.

I²S TX Data Register – I2STXDR

This register is used to specify the I²S transmitted data.

Offset: 0x00C

Reset value: 0x0000_0000

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

Bits	Field	Descriptions
[31:0]	TXDR	TX Data Register

I²S RX Data Register – I2SRXDR

This register is used to store the I²S received data.

Offset: 0x010

Reset value: 0x0000_0000

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

Bits	Field	Descriptions
[31:0]	RXDR	RX Data Register

I²S FIFO Control Register – I2SFCR

This register contains the related I²S FIFO control bits.

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						RW	0	RW	0		
	7	6	5	4	3	2	1	0				
Type/Reset	RW				RW				RW			

Bits	Field	Descriptions
[9]	RXFRST	RX FIFO Reset Set this bit to reset the RX FIFO.
[8]	TXFRST	TX FIFO Reset Set this bit to reset the TX FIFO.
[7:4]	RXFTLS	RX FIFO Trigger Level Select 0000: Trigger level is 0 0001: Trigger level is 1 ... 0111: Trigger level is 7 1xxx: Trigger level is 8 When the data contained in the RX FIFO is equal to or greater than the level defined by the RXFTLS field, the RXFTL flag will be set.
[3:0]	TXFTLS	TX FIFO Trigger Level Select 0000: Trigger level is 0 0001: Trigger level is 1 ... 0111: Trigger level is 7 1xxx: Trigger level is 8 When the data contained in the TX FIFO is equal to or less than the level defined by the TXFTLS field, the TXFTL flag will be set.

I²S Status Register – I2SSR

This register contains the relevant I²S status.

Offset: 0x018

Reset value: 0x0000_0809

	31	30	29	28	27	26	25	24	
	RXFS				TXFS				
Type/Reset	RO	0	RO	0	RO	0	RO	0	RO
	23	22	21	20	19	18	17	16	
	Reserved					CLKRDY	TXBUSY	CHS	
Type/Reset						RO	0	RO	0
	15	14	13	12	11	10	9	8	
	Reserved			RXFFUL	RXFEMT	RXFOV	RXFUD	RXFTL	
Type/Reset				RO	0	RO	1	WC	0
	7	6	5	4	3	2	1	0	
	Reserved			TXFFUL	TXFEMT	TXFOV	TXFUD	TXFTL	
Type/Reset				RO	0	RO	1	WC	0

Bits	Field	Descriptions
[31:28]	RXFS	RX FIFO Status 0000: RX FIFO empty 0001: RX FIFO contains 1 data ... 1000: RX FIFO contains 8 data Others: Reserved
[27:24]	TXFS	TX FIFO Status 0000: TX FIFO empty 0001: TX FIFO contains 1 data ... 1000: TX FIFO contains 8 data Others: Reserved
[18]	CLKRDY	Clock Divider Output Ready Flag 0: not ready 1: ready
[17]	TXBUSY	TX Busy Flag 0: not busy 1: busy
[16]	CHS	Channel Status 0: left channel 1: right channel
[12]	RXFFUL	RX FIFO Full Flag 0: RX FIFO not full 1: RX FIFO full
[11]	RXFEMT	RX FIFO Empty Flag 0: RX FIFO not empty 1: RX FIFO empty

Bits	Field	Descriptions
[10]	RXFOV	RX FIFO Overflow Flag 0: RX FIFO not overflow 1: RX FIFO overflow This bit is set by hardware and cleared by writing 1.
[9]	RXFUD	RX FIFO Underflow Flag 0: RX FIFO not underflow 1: RX FIFO underflow This bit is set by hardware and cleared by writing 1.
[8]	RXFTL	RX FIFO Trigger Level Flag 0: Data in the RX FIFO is less than the trigger level 1: Data in the RX FIFO is equal to or higher than the trigger level This bit is set by hardware and cleared by writing 1.
[4]	TXFFUL	TX FIFO Full Flag 0: TX FIFO not full 1: TX FIFO full
[3]	TXFEMT	TX FIFO Empty Flag 0: TX FIFO not empty 1: TX FIFO empty
[2]	TXFOV	TX FIFO Overflow Flag 0: TX FIFO not overflow 1: TX FIFO overflow This bit is set by hardware and cleared by writing 1.
[1]	TXFUD	TX FIFO Underflow Flag 0: TX FIFO not underflow 1: TX FIFO underflow This bit is set by hardware and cleared by writing 1.
[0]	TXFTL	TX FIFO Trigger Level Flag 0: Data in the TX FIFO is higher than the trigger level 1: Data in the TX FIFO is equal to or less than the trigger level This bit is set by hardware and cleared by writing 1.

I²S Rate Counter Value Register – I2SRCNTR

This register specifies the I²S rate control counter value.

Offset: 0x01C

Reset value: 0x0000_0000

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

Bits	Field	Descriptions
[19:0]	RCNTR	Rate Counter Value This value must be equal to or higher than Y_DIV for useful rate fine-tuning control.

Inter-IC Sound (I2S)

29 Cyclic Redundancy Check (CRC)

Introduction

The CRC (Cyclic Redundancy Check) calculation unit is an error detection technique test algorithm and uses 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- 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.

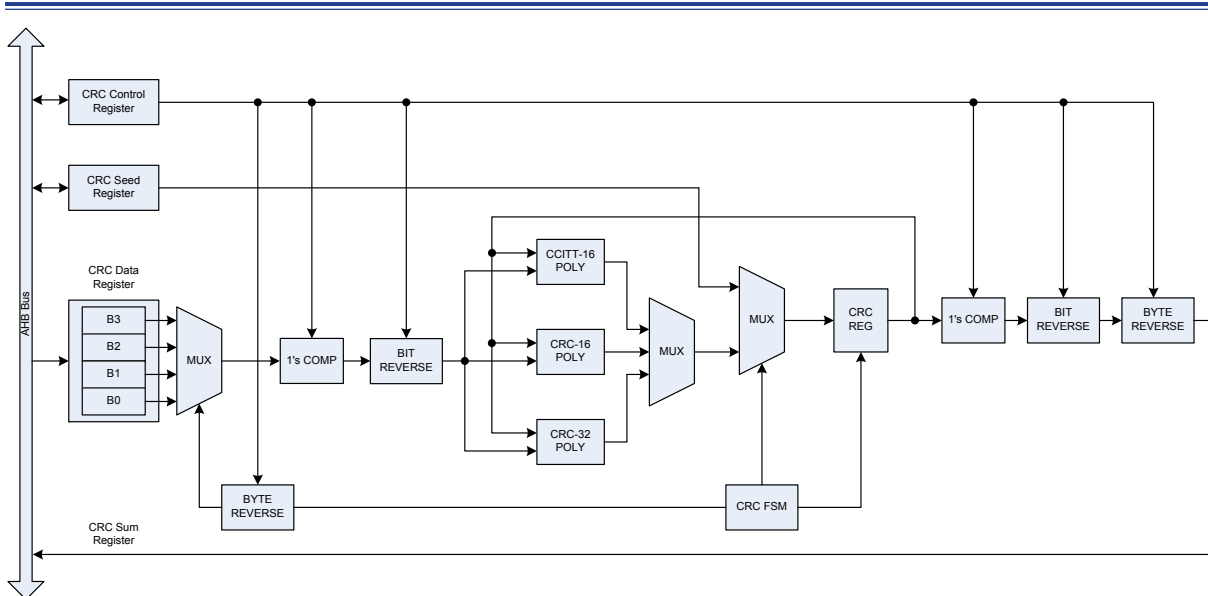


Figure 217. CRC Block Diagram

Features

- Support CRC16 polynomial: $0x8005, X^{16}+X^{15}+X^2+1$
- Support CCITT CRC16 polynomial: $0x1021, X^{16}+X^{12}+X^5+1$
- Support IEEE-802.3 CRC32 polynomial: $0x04C11DB7, X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$
- Support 1's complement, byte reverse & bit reverse operation on data and checksum
- Support byte, half-word & 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
- Support PDMA to complete a CRC computation of a block of memory

Function Descriptions

This unit only enables the calculation in the CRC16, CCITT CRC16 and IEEE-802.3 CRC32 polynomial. 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 32-bit write CRC data register (CRCDR) and 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 217. The CRC unit calculates the CRC data register (CRCDR) value is basic on byte by byte and default byte and bit order is big-endian. The CRCDR register can be accessed write 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 cycles 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 words write. The hardware ignores the DATBYRV bit of the CRCRCR register with any byte writes but the bit reversal setting DATBIRV are still applied to the byte. The Figure 218 is shown the byte and bit reversal operation example.

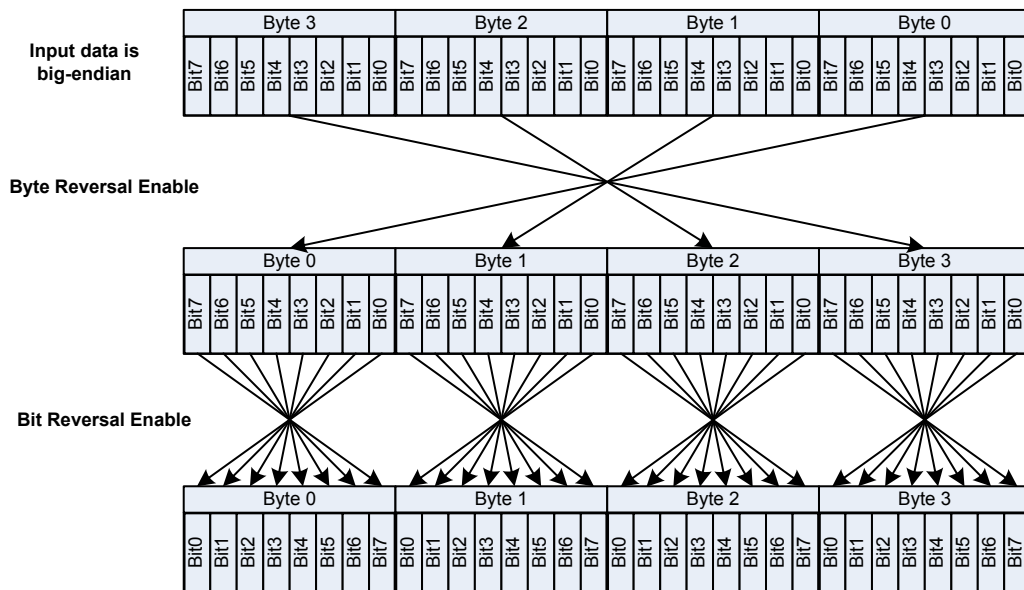


Figure 218. CRC Data Bit and Byte Reversal Example

CRC with PDMA

A PDMA channel with software trigger may be used to transfer data into the CRC unit. If a huge block data is needed to calculate. The recommended PDMA model is to use the PDMA to transfer all available words of data and uses software writes to transfer the other remaining bytes. To write data into the CRC unit, the PDMA should be accessed with data write by word 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 76. CRC Register Map

Register	Offset	Description	Reset Value
CRC Base Address = 0x4008A000			
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	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	SUMCMPL	SUMBYRV	SUMBIRV	DATCMPL	DATBYRV	DATBIRV	POLY	
Type/Reset	RW	0	RW	0	RW	0	RW	0	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
[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	0	WO	0	WO	0
	23	22	21	20	19	18	17	16	SEED					
Type/Reset	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0
	15	14	13	12	11	10	9	8	SEED					
Type/Reset	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0
	7	6	5	4	3	2	1	0	SEED					
Type/Reset	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0	WO	0

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

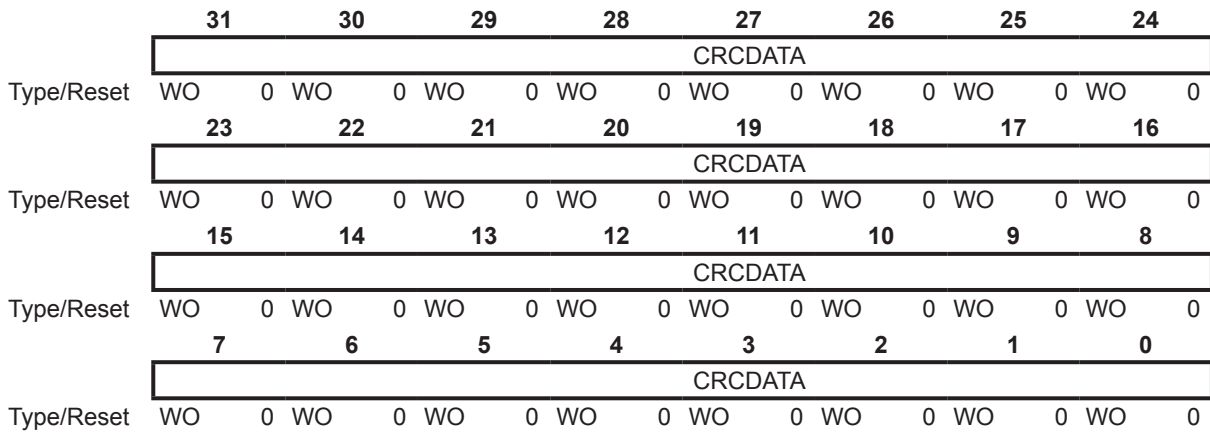
Bits	Field	Descriptions
[31:0]	CHKSUM	CRC Checksum Data Get the CRC 16/32-bit checksum result thought this register according to the polynomial setting in the CRCCR register after all data are written to the CRCDR register.

CRC Data Register – CRCDR

This register is used to specify the CRC input data.

Offset: 0x00C

Reset value: 0x0000_0000



Bits	Field	Descriptions
[31:0]	CRCDATA	CRC Input Data Byte, half-word & word write are allowed. 1's complement, byte reverse and bit reverse operation can be applied.

Copyright© 2018 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.tw>.