



---

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

**HT32F52231/HT32F52241  
HT32F52331/HT32F52341  
User Manual**

Revision: V1.30 Date: July 27, 2018

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

## Table of Contents

<b>1 Introduction .....</b>	<b>22</b>
Overview .....	22
Features.....	23
Device Information .....	27
Block Diagram .....	28
<b>2 Document Conventions .....</b>	<b>29</b>
<b>3 System Architecture .....</b>	<b>30</b>
Arm® Cortex®-M0+ Processor .....	30
Bus Architecture.....	31
Memory Organization .....	32
Memory Map.....	33
Embedded Flash Memory.....	35
Embedded SRAM Memory .....	35
AHB Peripherals .....	35
APB Peripherals .....	35
<b>4 Flash Memory Controller (FMC) .....</b>	<b>36</b>
Introduction .....	36
Features.....	36
Functional Descriptions .....	37
Flash Memory Map .....	37
Flash Memory Architecture .....	39
Wait State Setting .....	40
Booting Configuration .....	40
Page Erase .....	41
Mass Erase.....	42
Word Programming.....	43
Option Byte Description .....	44
Page Erase/Program Protection .....	45
Security Protection.....	46
Register Map .....	47
Register Descriptions.....	48
Flash Target Address Register – TADR .....	48
Flash Write Data Register – WRDR .....	49
Flash Operation Command Register – OCMR .....	50
Flash Operation Control Register – OPCR.....	51
Flash Operation Interrupt Enable Register – OIER .....	52
Flash Operation Interrupt and Status Register – OISR .....	53
Flash Page Erase/Program Protection Status Register – PPSR.....	54
Flash Security Protection Status Register – CPSR .....	55

Flash Vector Mapping Control Register – VMCR .....	56
Flash Manufacturer and Device ID Register – MDID.....	57
Flash Page Number Status Register – PNSR .....	58
Flash Page Size Status Register – PSSR .....	59
Device ID Register – DID.....	60
Flash Pre-fetch Control Register – CFCR .....	61
Custom ID Register n – CIDRn (n = 0 ~3) .....	62
<b>5 Power Control Unit (PWRCU) .....</b>	<b>63</b>
Introduction .....	63
Features.....	64
Functional Descriptions .....	64
V <sub>DD</sub> Power Domain.....	64
1.5 V Power Domain.....	66
Operation Modes .....	66
Register Map .....	69
Register Descriptions.....	69
Power Control Status Register – PWRSR.....	69
Power Control Register – PWRCR.....	70
V <sub>DD</sub> Power Domain Test Register – PWRTEST.....	72
Low Voltage / Brown Out Detect Control and Status Register – LVDCSR .....	73
<b>6 Clock Control Unit (CKCU) .....</b>	<b>75</b>
Introduction .....	75
Features.....	77
Function Descriptions .....	77
High Speed External Crystal Oscillator – HSE .....	77
High Speed Internal RC Oscillator – HSI.....	78
Auto Trimming of High Speed Internal RC Oscillator – HSI.....	78
Phase Locked Loop – PLL.....	80
Low Speed External Crystal Oscillator – LSE.....	82
Low Speed Internal RC Oscillator – LSI .....	82
Clock Ready Flag .....	82
System Clock (CK_SYS) Selection .....	83
HSE Clock Monitor .....	84
Clock Output Capability .....	84
Register Map .....	85
Register Descriptions.....	86
Global Clock Configuration Register – GCFGR.....	86
Global Clock Control Register – GCCR.....	88
Global Clock Status Register – GCSR .....	90
Global Clock Interrupt Register – GCIR.....	91
PLL Configuration Register – PLLCFG.....	92
PLL Control Register – PLLCR .....	92

AHB Configuration Register – AHBCFGR .....	93
AHB Clock Control Register – AHBCCR .....	94
APB Configuration Register – APBCFGR.....	96
APB Clock Control Register 0 – APBCCR0.....	97
APB Clock Control Register 1 – APBCCR1.....	99
Clock Source Status Register – CKST .....	101
APB Peripheral Clock Selection Register 0 – APBPCSR0 .....	102
APB Peripheral Clock Selection Register 1 – APBPCSR1 .....	104
HSI Control Register – HSICR.....	106
HSI Auto Trimming Counter Register – HSIATCR .....	107
Low Power Control Register – LPCR .....	108
MCU Debug Control Register – MCUDBGCR.....	109
<b>7 Reset Control Unit (RSTCU) .....</b>	<b>112</b>
Introduction .....	112
Functional Descriptions .....	113
Power On Reset .....	113
System Reset .....	113
AHB and APB Unit Reset.....	113
Register Map .....	114
Register Descriptions.....	114
Global Reset Status Register – GRSR .....	114
AHB Peripheral Reset Register – AHBPRSTR.....	115
APB Peripheral Reset Register 0 – APBPRSTRO .....	116
APB Peripheral Reset Register 1 – APBPRSTR1 .....	118
<b>8 General Purpose I/O (GPIO).....</b>	<b>120</b>
Introduction .....	120
Features.....	121
Functional Descriptions .....	121
Default GPIO Pin Configuration.....	121
General Purpose I/O – GPIO.....	121
GPIO Locking Mechanism.....	123
Register Map .....	123
Register Descriptions.....	124
Port A Data Direction Control Register – PADIRCR .....	124
Port A Input Function Enable Control Register – PAINER .....	125
Port A Pull-Up Selection Register – PAPUR .....	126
Port A Pull-Down Selection Register – PAPDR .....	127
Port A Open Drain Selection Register – PAODR .....	128
Port A Output Current Drive Selection Register – PADRVR .....	129
Port A Lock Register – PALOCKR .....	130
Port A Data Input Register – PADINR.....	131
Port A Output Data Register – PADOUTR .....	132

Port A Output Set/Reset Control Register – PASRR .....	133
Port A Output Reset Register – PARR .....	134
Port B Data Direction Control Register – PBDIRCR .....	135
Port B Input Function Enable Control Register – PBINER .....	136
Port B Pull-Up Selection Register – PBPUR .....	137
Port B Pull-Down Selection Register – PBPDR .....	138
Port B Open Drain Selection Register – PBODR .....	139
Port B Output Current Drive Selection Register – PBDRVVR .....	140
Port B Lock Register – PBLOCKR .....	141
Port B Data Input Register – PBDINR .....	142
Port B Output Data Register – PBDOUTR .....	143
Port B Output Set/Reset Control Register – PBSRR .....	144
Port B Output Reset Register – PBRR .....	145
Port C Data Direction Control Register – PCDIRCR .....	146
Port C Input Function Enable Control Register – PCINER .....	147
Port C Pull-Up Selection Register – PCPUR .....	148
Port C Pull-Down Selection Register – PCPDR .....	149
Port C Open Drain Selection Register – PCODR .....	150
Port C Output Current Drive Selection Register – PCDRVVR .....	151
Port C Lock Register – PCLOCKR .....	152
Port C Data Input Register – PCDINR .....	153
Port C Output Data Register – PCDOUTR .....	154
Port C Output Set/Reset Control Register – PCSRR .....	155
Port C Output Reset Register – PCRR .....	156
<b>9 Alternate Function Input/Output Control Unit (AFIO).....</b>	<b>157</b>
Introduction .....	157
Features .....	158
Functional Descriptions .....	158
External Interrupt Pin Selection .....	158
Alternate Function .....	159
Lock Mechanism .....	159
Register Map .....	159
Register Descriptions .....	160
EXTI Source Selection Register 0 – ESSR0 .....	160
EXTI Source Selection Register 1 – ESSR1 .....	161
GPIO x Configuration Low Register – GPxCFGLR, x = A, B, C .....	162
GPIO x Configuration High Register – GPxCFGHR, x = A, B, C .....	163
<b>10 Nested Vectored Interrupt Controller (NVIC).....</b>	<b>164</b>
Introduction .....	164
Features .....	165
Function Descriptions .....	166
SysTick Calibration .....	166

Register Map .....	166
<b>11 External Interrupt/Event Controller (EXTI).....</b>	<b>167</b>
Introduction .....	167
Features.....	167
Function Descriptions .....	168
Wakeup Event Management.....	168
External Interrupt/Event Line Mapping .....	169
Interrupt and Debounce .....	169
Register Map .....	170
Register Descriptions.....	171
EXTI Interrupt Configuration Register n – EXTICFGRn, n = 0 ~ 15 .....	171
EXTI Interrupt Control Register – EXTICR .....	172
EXTI Interrupt Edge Flag Register – EXTIEDGEFLGR.....	173
EXTI Interrupt Edge Status Register – EXTIEDGESR .....	174
EXTI Interrupt Software Set Command Register – EXTISSCR.....	175
EXTI Interrupt Wakeup Control Register – EXTIWAKUPCR .....	176
EXTI Interrupt Wakeup Polarity Register – EXTIWAKUPPOLR .....	177
EXTI Interrupt Wakeup Flag Register – EXTIWAKUPFLG .....	178
<b>12 Analog to Digital Converter (ADC) .....</b>	<b>179</b>
Introduction .....	179
Features.....	180
Function Descriptions .....	181
ADC Clock Setup.....	181
Channel Selection.....	181
Conversion Mode.....	181
Start Conversion on External Event.....	184
Sampling Time Setting.....	185
Data Format.....	185
Analog Watchdog.....	185
Interrupts.....	186
Register Map .....	187
Register Descriptions.....	188
ADC Conversion Control Register – ADCCR .....	188
ADC Conversion List Register 0 – ADCLST0 .....	190
ADC Conversion List Register 1 – ADCLST1 .....	191
ADC Input Sampling Time Register – ADCSTR.....	192
ADC Conversion Data Register y – ADCDRy, y = 0 ~ 7 .....	193
ADC Trigger Control Register – ADCTCR .....	194
ADC Trigger Source Register – ADCTSR.....	195
ADC Watchdog Control Register – ADCWCR .....	196
ADC Watchdog Threshold Register – ADCTR.....	197
ADC Interrupt Enable Register – ADCIER .....	198

ADC Interrupt Raw Status Register – ADCIRAW .....	199
ADC Interrupt Status Register – ADCISR.....	200
ADC Interrupt Clear Register – ADCICLR .....	201
<b>13 General-Purpose Timer (GPTM) .....</b>	<b>202</b>
Introduction .....	202
Features.....	203
Functional Descriptions .....	204
Counter Mode .....	204
Clock Controller .....	207
Trigger Controller .....	208
Slave Controller .....	209
Master Controller .....	212
Channel Controller.....	213
Input Stage .....	216
Quadrature Decoder .....	218
Output Stage.....	220
Update Management .....	224
Single Pulse Mode.....	225
Asymmetric PWM Mode .....	227
Timer Interconnection .....	228
Trigger ADC Start.....	231
Register Map .....	231
Register Descriptions.....	232
Timer Counter Configuration Register – CNTCFR .....	232
Timer Mode Configuration Register – MDCFR.....	234
Timer Trigger Configuration Register – TRCFR.....	237
Timer Counter Register – CTR .....	238
Channel 0 Input Configuration Register – CH0ICFR.....	239
Channel 1 Input Configuration Register – CH1ICFR.....	241
Channel 2 Input Configuration Register – CH2ICFR .....	243
Channel 3 Input Configuration Register – CH3ICFR.....	245
Channel 0 Output Configuration Register – CH0OCFR .....	247
Channel 1 Output Configuration Register – CH1OCFR .....	249
Channel 2 Output Configuration Register – CH2OCFR .....	251
Channel 3 Output Configuration Register – CH3OCFR .....	253
Channel Control Register – CHCTR.....	255
Channel Polarity Configuration Register – CHPOLR.....	256
Timer Interrupt Control Register – DICTR .....	257
Timer Event Generator Register – EVGR.....	258
Timer Interrupt Status Register – INTSR .....	260
Timer Counter Register – CNTR.....	263
Timer Prescaler Register – PSCR .....	264
Timer Counter Reload Register – CRR .....	265

Channel 0 Capture/Compare Register – CH0CCR .....	266
Channel 1 Capture/Compare Register – CH1CCR .....	267
Channel 2 Capture/Compare Register – CH2CCR .....	268
Channel 3 Capture/Compare Register – CH3CCR .....	269
Channel 0 Asymmetric Compare Register – CH0ACR.....	270
Channel 1 Asymmetric Compare Register – CH1ACR.....	271
Channel 2 Asymmetric Compare Register – CH2ACR.....	272
Channel 3 Asymmetric Compare Register – CH3ACR.....	273
<b>14 Basic Function Timer (BFTM).....</b>	<b>274</b>
Introduction .....	274
Features.....	274
Functional Description .....	275
Repetitive Mode.....	275
One Shot Mode.....	276
Trigger ADC Start.....	276
Register Map .....	277
Register Descriptions.....	277
BFTM Control Register – BFTMCR .....	277
BFTM Status Register – BFTMSR.....	278
BFTM Counter Register – BFTMCNTR .....	279
BFTM Compare Value Register – BFTMCMPR .....	280
<b>15 Motor Control Timer (MCTM) .....</b>	<b>281</b>
Introduction .....	281
Features.....	282
Functional Descriptions .....	283
Counter Mode .....	283
Clock Controller .....	287
Trigger Controller .....	288
Slave Controller .....	289
Master Controller .....	291
Channel Controller.....	292
Input Stage .....	295
Output Stage.....	297
Update Management .....	308
Single Pulse Mode.....	310
Asymmetric PWM Mode .....	312
Timer Interconnection .....	313
Trigger ADC Start.....	317
Lock Level Table .....	317
Register Map .....	318
Register Descriptions.....	319
Timer Counter Configuration Register – CNTCFR .....	319

Timer Mode Configuration Register – MDCFR .....	321
Timer Trigger Configuration Register – TRCFR .....	324
Timer Counter Register – CTR .....	325
Channel 0 Input Configuration Register – CH0ICFR .....	326
Channel 1 Input Configuration Register – CH1ICFR .....	328
Channel 2 Input Configuration Register – CH2ICFR .....	330
Channel 3 Input Configuration Register – CH3ICFR .....	332
Channel 0 Output Configuration Register – CH0OCFR .....	334
Channel 1 Output Configuration Register – CH1OCFR .....	336
Channel 2 Output Configuration Register – CH2OCFR .....	338
Channel 3 Output Configuration Register – CH3OCFR .....	340
Channel Control Register – CHCTR .....	342
Channel Polarity Configuration Register – CHPOLR .....	344
Channel Break Configuration Register – CHBRKCFR .....	346
Channel Break Control Register – CHBRKCTR .....	347
Timer Interrupt Control Register – DICTR .....	349
Timer Event Generator Register – EVGR .....	350
Timer Interrupt Status Register – INTSR .....	352
Timer Counter Register – CNTR .....	355
Timer Prescaler Register – PSCR .....	356
Timer Counter Reload Register – CRR .....	357
Timer Repetition Register – REPR .....	358
Channel 0 Capture/Compare Register – CH0CCR .....	359
Channel 1 Capture/Compare Register – CH1CCR .....	360
Channel 2 Capture/Compare Register – CH2CCR .....	361
Channel 3 Capture/Compare Register – CH3CCR .....	362
Channel 0 Asymmetric Compare Register – CH0ACR .....	363
Channel 1 Asymmetric Compare Register – CH1ACR .....	364
Channel 2 Asymmetric Compare Register – CH2ACR .....	365
Channel 3 Asymmetric Compare Register – CH3ACR .....	366
<b>16 Single-Channel Timer (SCTM) .....</b>	<b>367</b>
Introduction .....	367
Features .....	368
Functional Descriptions .....	368
Counter Mode .....	368
Clock Controller .....	369
Trigger Controller .....	370
Slave Controller .....	371
Channel Controller .....	373
Input Stage .....	375
Output Stage .....	376
Update Management .....	378
Register Map .....	379
Register Descriptions .....	380

Timer Counter Configuration Register – CNTCFR .....	380
Timer Mode Configuration Register – MDCFR.....	381
Timer Trigger Configuration Register – TRCFR.....	382
Timer Counter Register – CTR .....	383
Channel Input Configuration Register – CHICFR .....	384
Channel Output Configuration Register – CHOCFR .....	386
Channel Control Register – CHCTR.....	387
Channel Polarity Configuration Register – CHPOLR.....	388
Timer Interrupt Control Register – DICTR .....	389
Timer Event Generator Register – EVGR.....	390
Timer Interrupt Status Register – INTSR .....	391
Timer Counter Register – CNTR.....	392
Timer Prescaler Register – PSCR .....	393
Timer Counter Reload Register – CRR .....	394
Channel Capture/Compare Register – CHCCR .....	395
<b>17 Real Time Clock (RTC) .....</b>	<b>396</b>
Introduction .....	396
Features.....	396
Functional Descriptions .....	397
RTC Related Register Reset .....	397
Reading RTC Register.....	397
Low Speed Clock Configuration .....	397
RTC Counter Operation.....	398
Interrupt and Wakeup Control.....	398
RTCOUT Output Pin Configuration.....	399
Register Map .....	400
Register Descriptions.....	400
RTC Counter Register – RTCCNT.....	400
RTC Compare Register – RTCCMP .....	401
RTC Control Register – RTCCR .....	402
RTC Status Register – RTCSR.....	404
RTC Interrupt and Wakeup Enable Register – RTCIWEN.....	405
<b>18 Watchdog Timer (WDT) .....</b>	<b>406</b>
Introduction .....	406
Features.....	406
Functional Description .....	407
Register Map .....	409
Register Descriptions.....	409
Watchdog Timer Control Register – WDTCR .....	409
Watchdog Timer Mode Register 0 – WDTMR0.....	410
Watchdog Timer Mode Register 1 – WDTMR1.....	411
Watchdog Timer Status Register – WDTSR.....	412

Watchdog Timer Protection Register – WDTPR .....	413
Watchdog Timer Clock Selection Register – WDTCSR .....	414
<b>19 Inter-Integrated Circuit (I<sup>2</sup>C) .....</b>	<b>415</b>
Introduction .....	415
Features .....	416
Functional Descriptions .....	416
Two Wire Serial Interface.....	416
START and STOP Conditions.....	416
Data Validity.....	417
Addressing Format .....	418
Data Transfer and Acknowledge .....	420
Clock Synchronization .....	421
Arbitration .....	421
General Call Addressing .....	422
Bus Error.....	422
Address Mask Enable.....	422
Address Snoop .....	422
Operation Mode .....	422
Conditions of Holding SCL Line.....	428
I <sup>2</sup> C Timeout Function .....	429
Register Map .....	429
Register Descriptions.....	430
I <sup>2</sup> C Control Register – I2CCR .....	430
I <sup>2</sup> C Interrupt Enable Register – I2CIER .....	432
I <sup>2</sup> C Address Register – I2CADDR.....	434
I <sup>2</sup> C Status Register – I2CSR.....	435
I <sup>2</sup> C SCL High Period Generation Register – I2CSHPGR.....	438
I <sup>2</sup> C SCL Low Period Generation Register – I2CSLPGR.....	439
I <sup>2</sup> C Data Register – I2CDR .....	440
I <sup>2</sup> C Target Register – I2CTAR .....	441
I <sup>2</sup> C Address Mask Register – I2CADDMR .....	442
I <sup>2</sup> C Address Snoop Register – I2CADDSSR .....	443
I <sup>2</sup> C Timeout Register – I2CTOUT.....	444
<b>20 Serial Peripheral Interface (SPI) .....</b>	<b>445</b>
Introduction .....	445
Features .....	446
Function Descriptions .....	446
Master Mode.....	446
Slave Mode.....	446
SPI Serial Frame Format.....	447
Status Flags.....	451
Register Map .....	453

Register Descriptions.....	454
SPI Control Register 0 – SPICR0 .....	454
SPI Control Register 1 – SPICR1 .....	456
SPI Interrupt Enable Register – SPIIER .....	458
SPI Clock Prescaler Register – SPICPR .....	459
SPI Data Register – SPIDR .....	460
SPI Status Register – SPISR.....	461
SPI FIFO Control Register – SPIFCR.....	463
SPI FIFO Status Register – SPIFSR .....	464
SPI FIFO Time Out Counter Register – SPIFTOCR .....	465
<b>21 Universal Synchronous Asynchronous Receiver Transmitter (USART) .....</b>	<b>466</b>
Introduction .....	466
Features.....	467
Function Descriptions .....	468
Serial Data Format.....	468
Baud Rate Generation .....	469
Hardware Flow Control.....	470
IrDA.....	472
RS485 Mode.....	474
Synchronous Master Mode .....	477
Interrupts and Status .....	479
Register Map .....	479
Register Descriptions.....	480
USART Data Register – USRDR .....	480
USART Control Register – USRCR .....	481
USART FIFO Control Register – USRFCR.....	483
USART Interrupt Enable Register – USRIER .....	484
USART Status & Interrupt Flag Register – USRSIFR.....	486
USART Timing Parameter Register – USRTPR .....	488
USART IrDA Control Register – IrDACP .....	489
USART RS485 Control Register – RS485CR.....	490
USART Synchronous Control Register – SYNCR .....	491
USART Divider Latch Register – USRDLR.....	492
USART Test Register – USRTSTR .....	493
<b>22 Universal Asynchronous Receiver Transmitter (UART).....</b>	<b>494</b>
Introduction .....	494
Features.....	495
Function Descriptions .....	495
Serial Data Format.....	495
Baud Rate Generation .....	496
Interrupts and Status .....	497
Register Map .....	497

Register Descriptions.....	498
UART Data Register – URDR.....	498
UART Control Register – URCR.....	499
UART Interrupt Enable Register – URIER.....	501
UART Status & Interrupt Flag Register – URSIFR .....	502
UART Divider Latch Register – URDLR .....	504
UART Test Register – URTSTR.....	505
<b>23 Smart Card Interface (SCI) .....</b>	<b>506</b>
Introduction .....	506
Features.....	507
Functional Descriptions .....	507
Elementary Time Unit Counter.....	507
Guard Time Counter .....	510
Waiting Time Counter .....	510
Card Clock and Data Selection.....	511
Card Detection .....	511
SCI Data Transfer Mode.....	512
Interrupt Generator .....	514
Register Map .....	515
Register Descriptions.....	516
SCI Control Register – CR.....	516
SCI Status Register – SR .....	518
SCI Contact Control Register – CCR.....	520
SCI Elementary Time Unit Register – ETUR .....	521
SCI Guard Time Register – GTR .....	522
SCI Waiting Time Register – WTR.....	523
SCI Interrupt Enable Register – IER.....	524
SCI Interrupt Pending Register – IPR.....	526
SCI Transmit Buffer – TXB.....	528
SCI Receive Buffer – RXB .....	528
SCI Prescaler Register – PSCR .....	529
<b>24 USB Device Controller (USB) .....</b>	<b>530</b>
Introduction .....	530
Features.....	530
Functional Descriptions .....	531
Endpoints.....	531
EP_SRAM.....	531
Serial Interface Engine – SIE.....	532
Double-Buffering.....	532
Suspend Mode and Wake-up .....	534
Remote Wake-up .....	534
Register Map .....	534

Register Descriptions.....	536
USB Control and Status Register – USBCSR .....	536
USB Interrupt Enable Register – USBIER .....	538
USB Interrupt Status Register – USBISR .....	539
USB Frame Count Register – USBFCR .....	541
USB Device Address Register – USBDEVA .....	542
USB Endpoint 0 Control and Status Register – USBEP0CSR .....	543
USB Endpoint 0 Interrupt Enable Register – USBEP0IER .....	545
USB Endpoint 0 Interrupt Status Register – USBEP0ISR .....	546
USB Endpoint 0 Transfer Count Register – USBEP0TCR .....	548
USB Endpoint 0 Configuration Register – USBEP0CFG .....	549
USB Endpoint 1 ~ 3 Control and Status Register – USBEPnCSR, n = 1 ~ 3 .....	550
USB Endpoint 1 ~ 3 Interrupt Enable Register – USBEPnIER, n = 1 ~ 3 .....	552
USB Endpoint 1 ~ 3 Interrupt Status Register – USBEPnISR, n = 1 ~ 3 .....	553
USB Endpoint 1 ~ 3 Transfer Count Register – USBEPnTCR, n = 1 ~ 3 .....	554
USB Endpoint 1 ~ 3 Configuration Register – USBEPnCFG .....	555
USB Endpoint 4 ~ 7 Control and Status Register – USBEPnCSR, n = 4 ~ 7 .....	556
USB Endpoint 4 ~ 7 Interrupt Enable Register – USBEPnIER, n = 4 ~ 7 .....	559
USB Endpoint 4 ~ 7 Interrupt Status Register – USBEPnISR, n = 4 ~ 7 .....	560
USB Endpoint 4 ~ 7 Transfer Count Register – USBEPnTCR, n = 4 ~ 7 .....	561
USB Endpoint 4 ~ 7 Configuration Register – USBEPnCFG .....	562
<b>25 Cyclic Redundancy Check (CRC).....</b>	<b>563</b>
Introduction .....	563
Features.....	564
Function Descriptions .....	564
CRC Computation.....	564
Byte and Bit Reversal for CRC Computation .....	564
Register Map .....	565
Register Descriptions.....	566
CRC Control Register – CRCCR .....	566
CRC Seed Register – CRCSDR.....	567
CRC Checksum Register – CRCCSR .....	568
CRC Data Register – CRCDR .....	569

## List of Tables

Table 1. Features and Peripheral List .....	27
Table 2. Document Conventions .....	29
Table 3. Register Map .....	34
Table 4. HT32F52231/52241 Flash Memory and Option Byte .....	39
Table 5. HT32F52331/52341 Flash Memory and Option Byte .....	39
Table 6. Relationship between wait state cycle and HCLK .....	40
Table 7. Booting Modes .....	40
Table 8. Option Byte Memory Map .....	44
Table 9. Access Permission of Protected Main Flash Page .....	45
Table 10. Access Permission When Security Protection is Enabled .....	46
Table 11. FMC Register Map .....	47
Table 12. Table 1 Operation Mode Definitions .....	67
Table 13. Enter/Exit Power Saving Modes .....	67
Table 14. Power Status After System Reset .....	68
Table 15. PWRCU Register Map .....	69
Table 16. Output Divider2 Value Mapping .....	81
Table 17. Feedback Divider2 Value Mapping .....	81
Table 18. CKOUT Clock Source .....	84
Table 19. CKCU Register Map .....	85
Table 20. RSTCU Register Map .....	114
Table 21. AFIO, GPIO and IO Pad Control Signal True Table .....	122
Table 22. GPIO Register Map .....	123
Table 23. AFIO Selection for Peripheral Map Example .....	159
Table 24. AFIO Register Map .....	159
Table 25. Exception Types .....	164
Table 26. NVIC Register Map .....	166
Table 27. EXTI Register Map .....	170
Table 28. Data format in ADCDR [15:0] .....	185
Table 29. A/D Converter Register Map .....	187
Table 30. Counting Direction and Encoding Signals .....	219
Table 31. Compare Match Output Setup .....	220
Table 32. GPTM Register Map .....	231
Table 33. GPTM Internal Trigger Connection .....	237
Table 34. BFTM Register Map .....	277
Table 35. Compare Match Output Setup .....	298
Table 36. Output Control Bits for Complementary Output with a Break Event Occurrence .....	307
Table 37. Lock Level Table .....	317
Table 38. MCTM Register Map .....	318
Table 39. MCTM Internal Trigger Connection .....	324

Table 40. Compare Match Output Setup .....	376
Table 41. SCTM Register Map.....	379
Table 42. LSE Startup Mode Operating Current and Startup Time.....	397
Table 43. RTCOUT Output Mode and Active Level Setting .....	399
Table 44. RTC Register Map.....	400
Table 45. Watchdog Timer Register Map.....	409
Table 46. Conditions of Holding SCL line.....	428
Table 47. I <sup>2</sup> C Register Map .....	429
Table 48. I <sup>2</sup> C Clock Setting Example .....	439
Table 49. SPI Interface Format Setup.....	447
Table 50. SPI Mode Fault Trigger Conditions .....	452
Table 51. SPI Master Mode SEL Pin Status. ....	452
Table 52. SPI Register Map .....	453
Table 53. Baud Rate Deviation Error Calculation – CK_USART = 40 MHz.....	469
Table 54. Baud Rate Deviation Error Calculation – CK_USART = 48 MHz.....	470
Table 55. USART Register Map.....	479
Table 56. Baud Rate Deviation Error Calculation – CK_UART = 40 MHz .....	496
Table 57. Baud Rate Deviation Error Calculation – CK_UART = 48 MHz .....	497
Table 58. UART Register Map .....	497
Table 59. DI Field Based Di Encoded Decimal Values .....	508
Table 60. FI Field Based F <sub>i</sub> Encoded Decimal Values .....	508
Table 61. Possible ETU Values Obtained with the F <sub>i</sub> /Di Ratio.....	508
Table 62. SCI Register Map .....	515
Table 63. Endpoint Characteristics .....	531
Table 64. USB Data Types and Buffer Size .....	531
Table 65. USB Register Map .....	534
Table 66. Resume Event Detection .....	537
Table 67. CRC Register Map .....	565

## List of Figures

Figure 1. Block Diagram .....	28
Figure 2. Cortex™-M0+ Block Diagram.....	31
Figure 3. Bus Architecture .....	32
Figure 4. Memory Map.....	33
Figure 5. Flash Memory Controller Block Diagram.....	36
Figure 6. HT32F52231/52241 Flash Memory Map.....	37
Figure 7. HT32F52331/52341 Flash Memory Map.....	38
Figure 8. Vector Remapping.....	40
Figure 9. Page Erases Operation Flowchart.....	41
Figure 10. Mass Erases Operation Flowchart .....	42
Figure 11. Word Programming Operation Flowchart .....	43
Figure 12. PWRCU Block Diagram.....	63
Figure 13. Power On Reset / Power Down Reset Waveform .....	65
Figure 14. CKCU Block Diagram .....	76
Figure 15. External Crystal, Ceramic, and Resonators for HSE .....	77
Figure 16. HSI Auto Trimming Block Diagram .....	79
Figure 17. PLL Block Diagram .....	80
Figure 18. External Crystal, Ceramic, and Resonators for LSE .....	82
Figure 19. RSTCU Block Diagram.....	112
Figure 20. Power On Reset Sequence .....	113
Figure 21. GPIO Block Diagram .....	120
Figure 22. AFIO(GPIO Control Signal.....	122
Figure 23. AFIO Block Diagram .....	157
Figure 24. EXTI Channel Input Selection .....	158
Figure 25. EXTI Block Diagram .....	167
Figure 26. EXTI Wake-up Event Management .....	168
Figure 27. EXTI Interrupt Debounce Function .....	169
Figure 28. ADC Block Diagram .....	179
Figure 29. One Shot Conversion Mode .....	182
Figure 30. Continuous Conversion Mode .....	182
Figure 31. Discontinuous Conversion Mode.....	184
Figure 32. GPTM Block Diagram.....	202
Figure 33. Up-counting Example .....	204
Figure 34. Down-counting Example.....	205
Figure 35. Center-aligned Counting Example.....	206
Figure 36. GPTM Clock Selection Source .....	207
Figure 37. Trigger Controller Block .....	208
Figure 38. Slave Controller Diagram .....	209
Figure 39. GPTM in Restart Mode.....	209

Figure 40. GPTM in Pause Mode .....	210
Figure 41. GPTM in Trigger Mode .....	211
Figure 42. Master GPTMn and Slave GPTMm/MCTMm Connection.....	212
Figure 43. MTO Selection .....	212
Figure 44. Capture/Compare Block Diagram.....	213
Figure 45. Input Capture Mode .....	214
Figure 46. PWM Pulse Width Measurement Example.....	215
Figure 47. Channel 0 and Channel 1 Input Stages.....	216
Figure 48. Channel 2 and Channel 3 Input Stages.....	217
Figure 49. TI0 Digital Filter Diagram with N = 2.....	217
Figure 50. Input Stage and Quadrature Decoder Block Diagram .....	218
Figure 51. Both TI0 and TI1 Quadrature Decoder Counting.....	219
Figure 52. Output Stage Block Diagram .....	220
Figure 53. Toggle Mode Channel Output Reference Signal – CHxPRE = 0.....	221
Figure 54. Toggle Mode Channel Output Reference Signal – CHxPRE = 1 .....	221
Figure 55. PWM Mode Channel Output Reference Signal and Counter in Up-counting Mode .....	222
Figure 56. PWM Mode Channel Output Reference Signal and Counter in Down-counting Mode .....	222
Figure 57. PWM Mode Channel Output Reference Signal and Counter in Centre-align Mode .....	223
Figure 58. Update Event Setting Diagram .....	224
Figure 59. Single Pulse Mode.....	225
Figure 60. Immediate Active Mode Minimum Delay .....	226
Figure 61. Asymmetric PWM Mode versus Center Align Counting Mode.....	227
Figure 62. Pausing GPTM1 using the GPTM0 CH0OREF Signal .....	228
Figure 63. Triggering GPTM1 with GPTM0 Update Event.....	229
Figure 64. Trigger GPTM0 and GPTM1 with the GPTM0 CH0 Input.....	230
Figure 65. BFTM Block Diagram .....	274
Figure 66. BFTM – Repetitive Mode.....	275
Figure 67. BFTM – One Shot Mode.....	276
Figure 68. BFTM – One Shot Mode Counter Updating .....	276
Figure 69. MCTM Block Diagram .....	281
Figure 70. Up-counting Example .....	283
Figure 71. Down-counting Example.....	284
Figure 72. Center-aligned Counting Example.....	285
Figure 73. Update Event 1 Dependent Repetition Mechanism Example.....	286
Figure 74. MCTM Clock Selection Source.....	287
Figure 75. Trigger Controller Block .....	288
Figure 76. Slave Controller Diagram .....	289
Figure 77. MCTM in Restart Mode .....	289
Figure 78. MCTM in Pause Mode .....	290
Figure 79. MCTM in Trigger Mode .....	290
Figure 80. Master MCTMn and Slave GPTMn Connection .....	291

Figure 81. MTO Selection.....	291
Figure 82. Capture/Compare Block Diagram.....	292
Figure 83. Input Capture Mode.....	293
Figure 84. PWM Pulse Width Measurement Example.....	294
Figure 85. Channel 0 and Channel 1 Input Stages.....	295
Figure 86. Channel 2 and Channel 3 Input Stages.....	295
Figure 87. TI0 Digital Filter Diagram with N = 2.....	296
Figure 88. Output Stage Block Diagram .....	297
Figure 89. Toggle Mode Channel Output Reference Signal – CHxPRE = 0 .....	298
Figure 90. Toggle Mode Channel Output Reference Signal – CHxPRE = 1 .....	299
Figure 91. PWM Mode Channel Output Reference Signal and Counter in Up-counting Mode .....	299
Figure 92. PWM Mode Channel Output Reference Signal and Counter in Down-counting Mode .....	300
Figure 93. PWM Mode 1 Channel Output Reference Signal and Counter in Centre-aligned Counting Mode.....	300
Figure 94. Dead-time Insertion Performed for Complementary Outputs .....	301
Figure 95. MCTM Break Signal Bolck Diagram .....	302
Figure 96. MT_BRK Pin Digital Filter Diagram with N = 2 .....	302
Figure 97. Channel 3 Output with a Break Event Occurrence.....	303
Figure 98. Channel 0 ~2 Complementary Outputs with a Break Event Occurrence.....	304
Figure 99. Channel 0 ~2 Only One Output Enabled when Break Event Occurs .....	305
Figure 100. Hardware Protection When Both CHxO and CHxNO Are in Active Condition.....	306
Figure 101. Update Event 1 Setup Diagram .....	308
Figure 102. CHxE, CHxNE and CHxOM Updated by Update Event 2 .....	309
Figure 103. Update Event 2 Setup Diagram.....	309
Figure 104. Single Pulse Mode.....	310
Figure 105. Immediate Active Mode Minimum Delay .....	311
Figure 106. Asymmetric PWM Mode versus Center-aligned Counting Mode.....	312
Figure 107. Pausing GPTM using the MCTM CH0OREF Signal.....	313
Figure 108. Triggering GPTM with MCTM Update Event 1 .....	314
Figure 109. Trigger MCTM and GPTM with the MCTM CH0 Input .....	315
Figure 110. CH1XOR Input as Hall Sensor Interface.....	316
Figure 111. SCTM Block Diagram.....	367
Figure 112. Up-counting Example.....	368
Figure 113. SCTM Clock Selection Source.....	369
Figure 114. Trigger Control Block .....	370
Figure 115. Slave Controller Diagram.....	371
Figure 116. SCTM in Restart Mode .....	371
Figure 117. SCTM in Pause Mode.....	372
Figure 118. SCTM in Trigger Mode.....	372
Figure 119. Capture/Compare Block Diagram .....	373
Figure 120. Input Capture Mode .....	374

Figure 121. Channel Input Stages .....	375
Figure 122. TI Digital Filter Diagram with N = 2.....	375
Figure 123. Output Stage Block Diagram .....	376
Figure 124. Toggle Mode Channel Output Reference Signal – CHPRE = 0.....	377
Figure 125. Toggle Mode Channel Output Reference Signal – CHPRE = 1.....	377
Figure 126. PWM Mode Channel Output Reference Signal .....	378
Figure 127. Update Event Setting Diagram .....	379
Figure 128. RTC Block Diagram.....	396
Figure 129. Watchdog Timer Block Diagram .....	406
Figure 130. Watchdog Timer Behavior .....	408
Figure 131. I <sup>2</sup> C Module Block Diagram.....	415
Figure 132. START and STOP Condition .....	417
Figure 133. Data Validity.....	417
Figure 134. 7-bit Addressing Mode .....	418
Figure 135. 10-bit Addressing Write Transmit Mode .....	419
Figure 136. 10-bits Addressing Read Receive Mode .....	419
Figure 137. I <sup>2</sup> C Bus Acknowledge .....	420
Figure 138. Clock Synchronization during Arbitration.....	421
Figure 139. Two Master Arbitration Procedure .....	421
Figure 140. Master Transmitter Timing Diagram .....	423
Figure 141. Master Receiver Timing Diagram .....	425
Figure 142. Slave Transmitter Timing Diagram .....	426
Figure 143. Slave Receiver Timing Diagram .....	427
Figure 144. SCL Timing Diagram.....	439
Figure 145. SPI Block Diagram .....	445
Figure 146. SPI Single Byte Transfer Timing Diagram – CPOL = 0, CPHA = 0.....	447
Figure 147. SPI Continuous Data Transfer Timing Diagram – CPOL = 0, CPHA = 0 .....	448
Figure 148. SPI Single Byte Transfer Timing Diagram – CPOL = 0, CPHA = 1.....	448
Figure 149. SPI Continuous Transfer Timing Diagram – CPOL = 0, CPHA = 1.....	449
Figure 150. SPI Single Byte Transfer Timing Diagram – CPOL = 1, CPHA = 0.....	449
Figure 151. SPI Continuous Transfer Timing Diagram – CPOL = 1, CPHA = 0.....	450
Figure 152. SPI Single Byte Transfer Timing Diagram – CPOL = 1, CPHA = 1.....	450
Figure 153. SPI Continuous Transfer Timing Diagram – CPOL = 1, CPHA = 1.....	450
Figure 154. SPI Multi-Master Slave Environment.....	452
Figure 155. USART Block Diagram .....	466
Figure 156. USART Serial Data Format .....	468
Figure 157. USART Clock CK_USART and Data Frame Timing.....	469
Figure 158. Hardware Flow Control between 2 USARTs.....	470
Figure 159. USART RTS Flow Control .....	471
Figure 160. USART CTS Flow Control .....	471
Figure 161. IrDA Modulation and Demodulation.....	472

Figure 162. USART I/O and IrDA Block Diagram .....	474
Figure 163. RS485 Interface and Waveform .....	475
Figure 164. USART Synchronous Transmission Example .....	477
Figure 165. 8-bit Format USART Synchronous Waveform.....	478
Figure 166. UART Block Diagram.....	494
Figure 167. UART Serial Data Format.....	495
Figure 168. UART Clock CK_UART and Data Frame Timing.....	496
Figure 169. SCI Block Diagram .....	506
Figure 170. Character Frame and Compensation Mode .....	509
Figure 171. Guard Time Duration .....	510
Figure 172. Character and Block Waiting Time Duration – CWT and BWT.....	511
Figure 173. SCI Card Detection Diagram .....	512
Figure 174. SCI Interrupt Structure.....	514
Figure 175. USB Block Diagram.....	530
Figure 176. Endpoint Buffer Allocation Example.....	532
Figure 177. Double-buffering Operation Example .....	533
Figure 178. CRC Block Diagram .....	563
Figure 179. CRC Data Bit and Byte Reversal Example.....	565

# 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 40 MHz for HT32F52231/52241 or 48 MHz for HT32F52331/52341 with a Flash accelerator to obtain maximum efficiency. It provides up to 64 KB of embedded Flash memory for code/data storage and 8 KB of embedded SRAM memory for system operation and application program usage. A variety of peripherals, such as ADC, I<sup>2</sup>C, USART, UART, SPI, MCTM, GPTM, SSTM, CRC-16/32, RTC, WDT, SCI, 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 40MHz operating frequency for HT32F52231/52241 or 48MHz for HT32F52331/52341
  - 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 64 KB on-chip Flash memory for instruction/data and options storage
  - Up to 8 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  $\pm 2\%$  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
  - $V_{DD}$  power supply for RTC.
  - Two power domains:  $V_{DD}$  and 1.5 V.
  - Four power saving modes: Sleep, Deep-Sleep1, Deep-Sleep2, Power-Down
- External Interrupt/Event Controller – EXTI
  - Up to 16 EXTI lines with configurable trigger source and type
  - All GPIO pins can be selected as EXTI trigger source
  - Source trigger type 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 – 1  $\mu$ s at 28 MHz, 1.4  $\mu$ s at 40 MHz
  - Up to 12 external analog input channels
- IO ports – GPIO
  - Up to 40 GPIOs
  - Port A, B, C 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
  - 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 PWM Generation and Capture Timers – SCTM
  - One 16-bit up and auto-reload counter
  - One 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
  - Single Pulse Mode Output
- Basic Function Timer – BFTM
  - 32-bit compare/match count-up counter – no I/O control features
  - One shot mode – counting stops after a match condition
  - Repetitive mode – restart counter after a match condition

- Watchdog Timer
  - 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
  - 24-bit up-counter with a programmable prescaler
  - Alarm function
  - Interrupt and Wake-up event
- Inter-integrated Circuit – I<sup>2</sup>C
  - 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 ( $f_{PCLK}/2$ ) MHz for master mode and ( $f_{PCLK}/3$ ) MHz for slave mode
  - FIFO Depth: 8 levels
  - Multi-master and multi-slave operation
- Universal Synchronous Asynchronous Receiver Transmitter – USART
  - Supports both asynchronous and clocked synchronous serial communication modes
  - Asynchronous operating baud rate up to ( $f_{PCLK}/16$ ) MHz and synchronous operating rate up to ( $f_{PCLK}/8$ ) 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 x 9 bits for both receiver and transmitter
- Universal Asynchronous Receiver Transmitter – UART
  - Asynchronous serial communication operating baud-rate up to ( $f_{PCLK}/16$ ) 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 (HT32F52331/HT32F52341 only)
  - 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

- 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,  $X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$
  - 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
- Universal Serial Bus Device Controller – USB (HT32F52331/HT32F52341 only)
  - 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
  - HT32F52231/HT32F52241: 24/28-pin SSOP, 33-pin QFN, 48-pin LQFP
  - HT32F52331/HT32F52341: 33-pin QFN, 48-pin LQFP package
  - Operation temperature range: -40 °C to +85 °C

## Device Information

**Table 1. Features and Peripheral List**

Peripherals	HT32F52231	HT32F52241	HT32F52331	HT32F52341
Main Flash (KB)	32	63	32	63
Option Bytes Flash (KB)	1	1	1	1
SRAM (KB)	4	8	4	8
Timers	MCTM	1		
	GPTM	1		
	SCTM	4		
	BFTM	2		
	RTC	1		
	WDT	1		
Communication	USB	—	1	
	SPI	2		
	USART	1		
	UART	2		
	I <sup>2</sup> C	2		
	SCI (ISO7816-3)	—	1	
CRC-16/32	1			
EXTI	16			
12-bit ADC	1			
Number of channels	12 Channels			
GPIO	Up to 40	Up to 38		
CPU frequency	Up to 40 MHz	Up to 48 MHz		
Operating voltage	2.0 V ~ 3.6 V			
Operating temperature	-40 °C ~ +85 °C			
Package	24/28-pin SSOP 33-pin QFN, 48-pin LQFP		33-pin QFN, 48-pin LQFP	

## Block Diagram

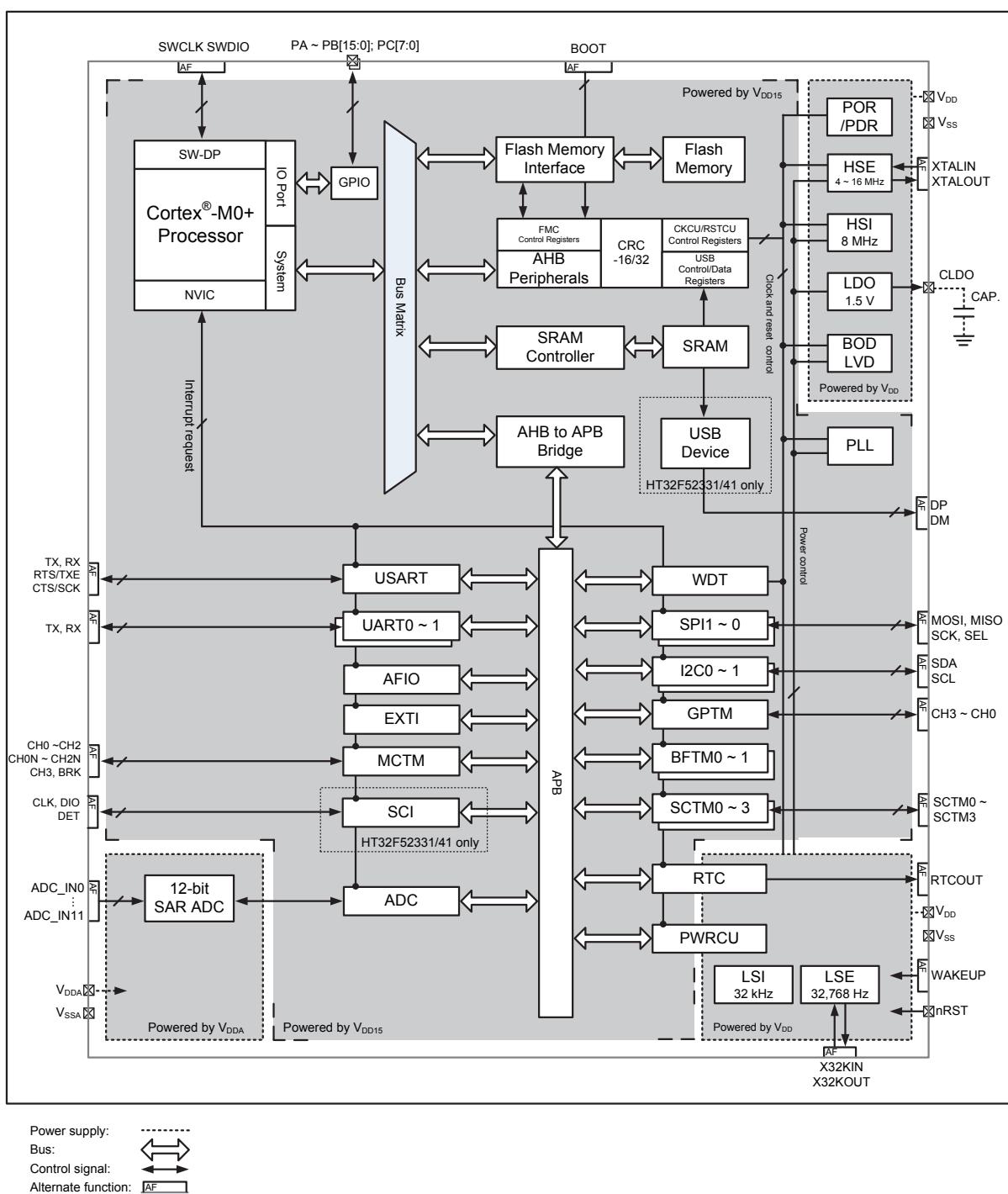


Figure 1. Block Diagram

## 2 Document Conventions

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

**Table 2. Document Conventions**

Notation	Example	Description						
0x	0x5a05	The number string with a 0x prefix indicates a hexadecimal number.						
0xnnnn_nnnn	0x2000_0100	32-bit Hexadecimal address or data.						
b	b0101	The number string with a lowercase b prefix indicates a binary number.						
NAME [n]	ADDR [5]	Specific bit of NAME. NAME can be a register or field of register. For example, ADDR [5] means bit 5 of ADDR register (field).						
NAME [m:n]	ADDR [11:5]	Specific bits of NAME. NAME can be a register or field of register. For example, ADDR [11:5] means bit 11 to 5 of ADDR register (field).						
X	b10X1	Don't care notation which means any value is allowed.						
RW	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="text-align: center;">19</td> <td style="text-align: center;">18</td> </tr> <tr> <td>SERDYIE</td> <td>PLLRDYIE</td> </tr> <tr> <td>RW 0</td> <td>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="display: inline-table; vertical-align: middle;"> <tr> <td style="text-align: center;">3</td> <td style="text-align: center;">2</td> </tr> <tr> <td>HSIRDY</td> <td>HSERDY</td> </tr> <tr> <td>RO 1</td> <td>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="display: inline-table; vertical-align: middle;"> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> </tr> <tr> <td>PDF</td> <td>BAK PORF</td> </tr> <tr> <td>RC 0</td> <td>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="display: inline-table; vertical-align: middle;"> <tr> <td style="text-align: center;">3</td> <td style="text-align: center;">2</td> </tr> <tr> <td>SERDYF</td> <td>PLLRDYF</td> </tr> <tr> <td>WC 0</td> <td>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="display: inline-table; vertical-align: middle;"> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> </tr> <tr> <td>RXCF</td> <td>PARF</td> </tr> <tr> <td>RO 0</td> <td>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="display: inline-table; vertical-align: middle;"> <tr> <td style="text-align: center;">31</td> <td style="text-align: center;">30</td> </tr> <tr> <td></td> <td>DB_CKSRC</td> </tr> <tr> <td>WO 0</td> <td>WO 0</td> </tr> </table>	31	30		DB_CKSRC	WO 0	WO 0	Software can only write to this bit. A read operation always returns 0.
31	30							
	DB_CKSRC							
WO 0	WO 0							
Reserved	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> </tr> <tr> <td>LLRDY</td> <td>Reserved</td> </tr> <tr> <td>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

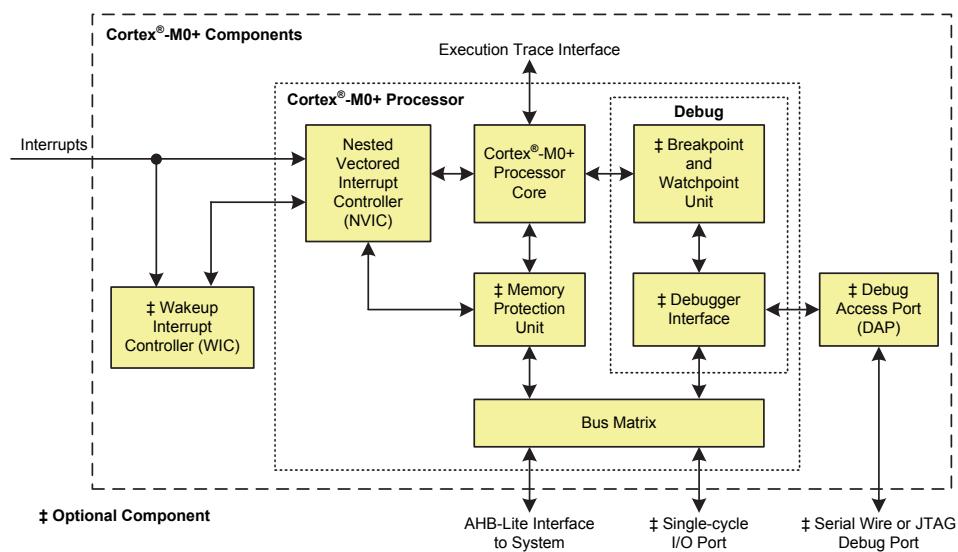
The system architecture of devices that includes the Arm® Cortex®-M0+ processor, bus architecture and memory organization will be described in the following sections. The Cortex®-M0+ is a next generation processor core which offers many new features. Integrated and advanced features make the Cortex®-M0+ 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)

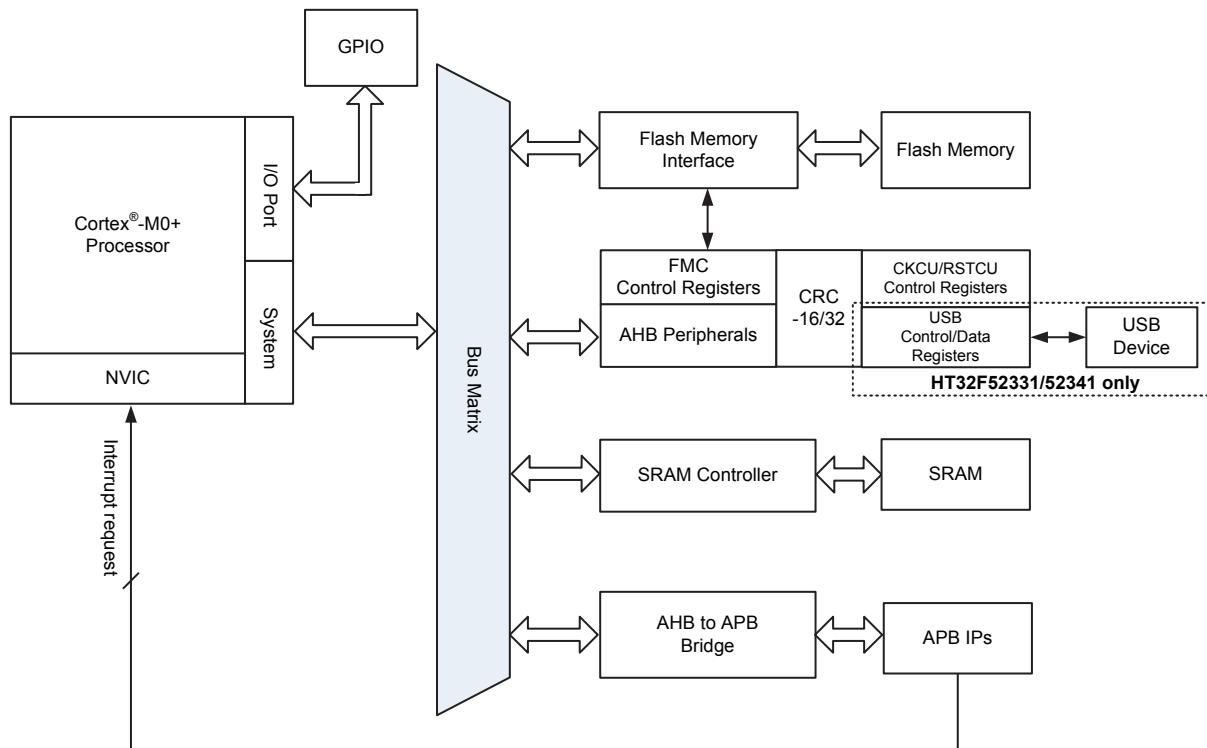
The following figure shows the Cortex®-M0+ processor block diagram. For more information, refer to the Arm® Cortex®-M0+ Technical Reference Manual.



**Figure 2. Cortex™-M0+ Block Diagram**

## Bus Architecture

The HT32F52231/HT32F52241 and HT32F52331/HT32F52341 series consist of one master and four slaves in the bus architecture. The Cortex®-M0+ AHB-Lite bus is the master while the internal SRAM access bus, the internal Flash memory access bus, the AHB peripherals access bus and the AHB to APB bridges are the slaves. The single 32-bit AHB-Lite system interface provides simple integration to all system regions 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 HT32F52231/HT32F52241 and HT32F52331/HT32F52341 series.

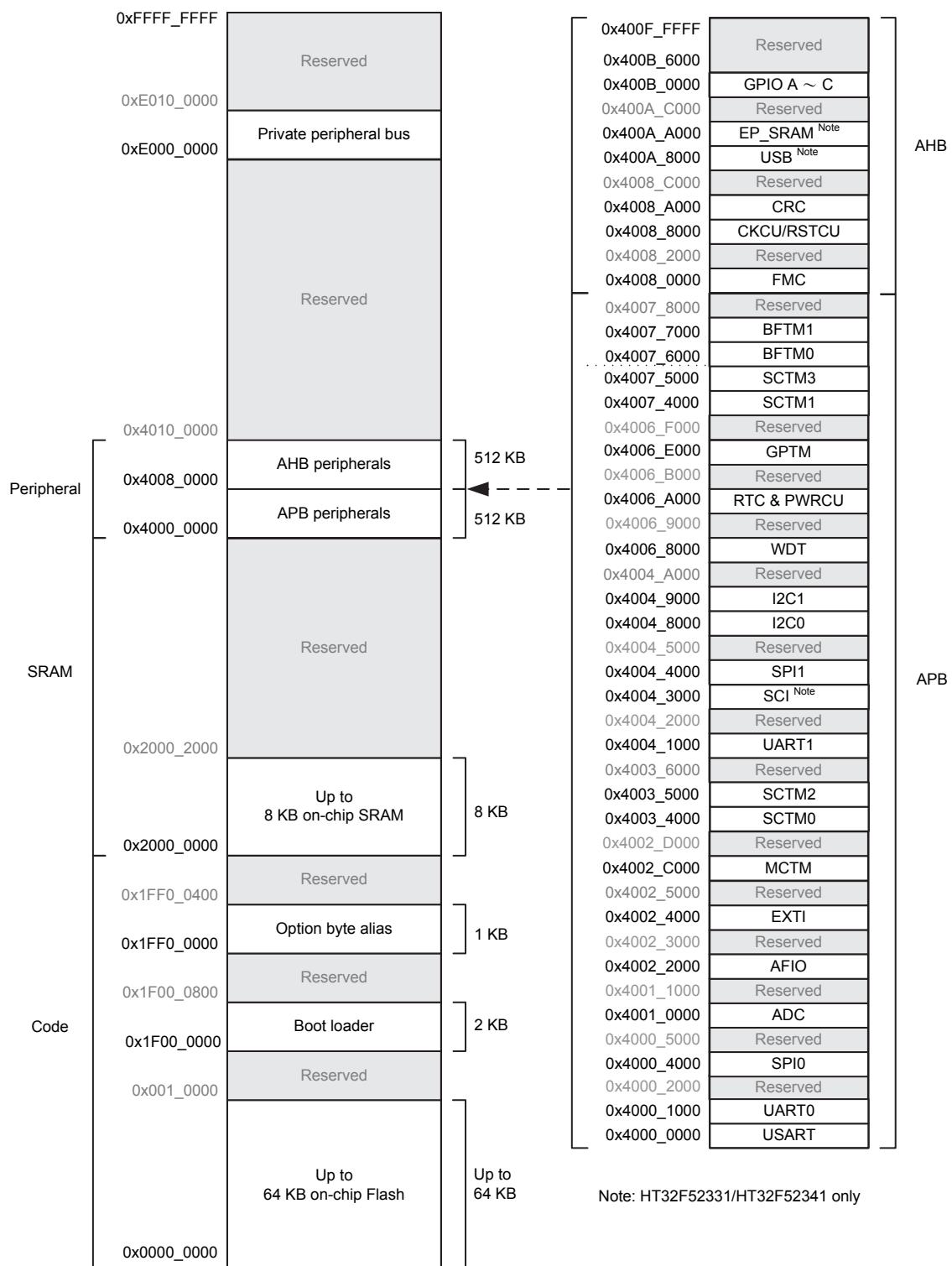


**Figure 3. Bus Architecture**

## 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 HT32F52231/HT32F52241 and HT32F52331/HT32F52341 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	USART	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_BFFF	Reserved	
0x4002_C000	0x4002_CFFF	MCTM	
0x4002_D000	0x4003_3FFF	Reserved	
0x4003_4000	0x4003_4FFF	SCTM0	
0x4003_5000	0x4003_5FFF	SCTM2	
0x4003_6000	0x4004_0FFF	Reserved	
0x4004_1000	0x4004_1FFF	UART1	
0x4004_2000	0x4004_2FFF	Reserved	
0x4004_3000	0x4004_3FFF	SCI <sup>Note</sup>	
0x4004_4000	0x4004_4FFF	SPI1	
0x4004_5000	0x4004_7FFF	Reserved	
0x4004_8000	0x4004_8FFF	I2C0	
0x4004_9000	0x4004_9FFF	I2C1	
0x4004_A000	0x4006_7FFF	Reserved	
0x4006_8000	0x4006_8FFF	WDT	
0x4006_9000	0x4006_9FFF	Reserved	
0x4006_A000	0x4006_AFFF	RTC/PWRCU	
0x4006_B000	0x4006_DFFF	Reserved	
0x4006_E000	0x4006_EFFF	GPTM	
0x4006_F000	0x4007_3FFF	Reserved	
0x4007_4000	0x4007_4FFF	SCTM1	
0x4007_5000	0x4007_5FFF	SCTM3	
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	0x400A_7FFF	Reserved	
0x400A_8000	0x400A_BFFF	USB <small>Note</small>	
0x400A_C000	0x400A_FFFF	Reserved	
0x400B_0000	0x400B_1FFF	GPIOA	
0x400B_2000	0x400B_3FFF	GPIOB	
0x400B_4000	0x400B_5FFF	GPIOC	
0x400B_6000	0x400F_FFFF	Reserved	

**Note:** HT32F52331/HT32F52341 only.

## Embedded Flash Memory

The HT32F52231/HT32F52241 and HT32F52331/HT32F52341 series provide up to 64 KB on-chip Flash memory which is located at address 0x0000\_0000. It supports byte, half-word, and word access operations. Note that the Flash memory only supports read operations for the bus access. Any write operations to the Flash memory will cause a bus fault exception. The Flash memory has up to capacity of 64 pages. Each page has a memory capacity of 1 KB and can be erased independently. A 32-bit programming interface provides the capability of changing bits from 1 to 0. A data storage or firmware upgrade can be implemented using several methods such as In System Programming (ISP), In Application Programming (IAP) or In Circuit Programming (ICP). For more information, refer to the Flash Memory Controller section.

## Embedded SRAM Memory

The HT32F52231/HT32F52241 and HT32F52331/HT32F52341 series contain up to 8 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.

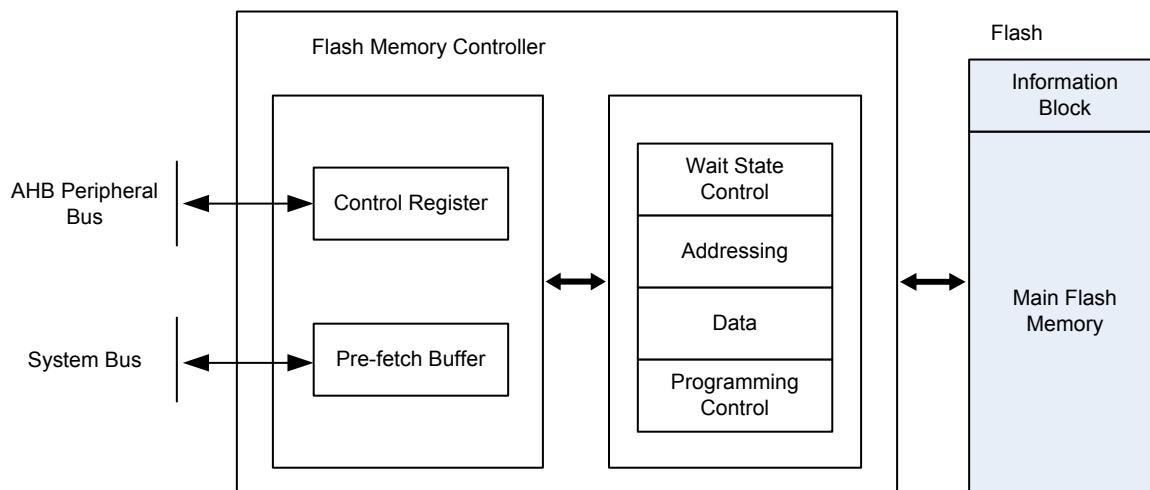


Figure 5. Flash Memory Controller Block Diagram

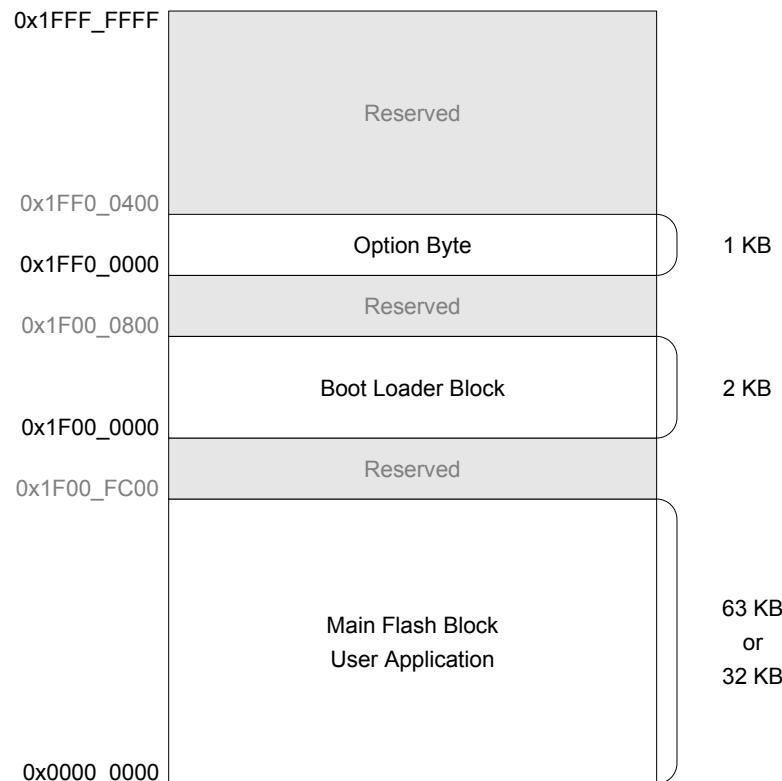
## Features

- Up to 64 KB of on-chip Flash memory for storing instruction/data and options
  - 64 KB (instruction/data + Option Byte)
  - 32 KB (instruction/data + Option Byte)
- Page size of 1K bytes for HT32F52231/52241 and 512 bytes for HT32F52331/52341, totally up to 64/128 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 HT32F52231/52241. The address ranges from 0x0000\_0000 to 0x1FFF\_FFFF (0.5 GB). The address from 0x1F00\_0000 to 0x1F00\_07FF is mapped to Boot Loader Block (2 KB). Besides, address 0xFF0\_0000 to 0xFF0\_03FF is the alias of Option Byte block (1 KB) which locates at the last page of main Flash physically. The memory mapping on system view is shown as below.



**Figure 6. HT32F52231/52241 Flash Memory Map**

The following figure is the Flash memory map of the HT32F52331/52341. 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 (512 bytes) which locates at the last page of main Flash physically. The memory mapping on system view is shown as below.

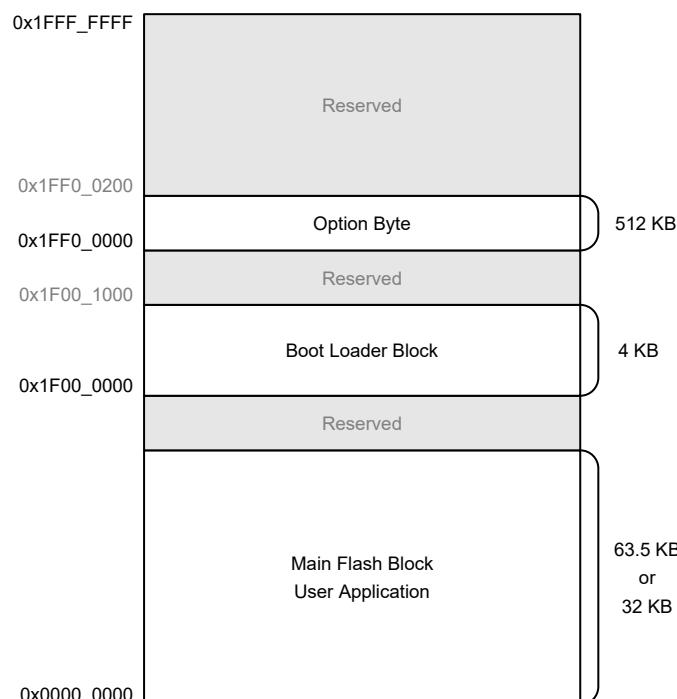


Figure 7. HT32F52331/52341 Flash Memory Map

## Flash Memory Architecture

The HT32F52231/52241 Flash memory consists of up to 64 KB main Flash with 1 KB per page and 2 KB Information Block for Boot Loader. The main Flash memory contains totally 64 pages (or 32 pages for 32 KB device) which can be erased individually. The following table shows the base address, size, and protection setting bit of each page.

**Table 4. HT32F52231/52241 Flash Memory and Option Byte**

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

The HT32F52331/52341 Flash memory consists of up to 64 KB main Flash with 512 bytes per page and 4 KB Information Block for Boot Loader. The main Flash memory contains totally 128 pages (or 64 pages for 32 KB device) which can be erased individually. The following table shows the base address, size, and protection setting bit of each page.

**Table 5. HT32F52331/52341 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	OB_PP [1]	512 bytes
	Page 2	0x0000_0400 ~ 0x0000_05FF	OB_PP [2]	512 bytes
	Page 3	0x0000_0600 ~ 0x0000_07FF	OB_PP [3]	512 bytes
	...	...	...	...
	Page 124	0x0000_F800 ~ 0x0000_F9FF	OB_PP [124]	512 bytes
	Page 125	0x0000_FA00 ~ 0x0000_FBFF	OB_PP [125]	512 bytes
	Page 126	0x0000_FC00 ~ 0x0000_FDFF	OB_PP [126]	512 bytes
	Page 127 (Option Byte)	0x0000_FE00 ~ 0x0000_FFFF Alias: 0x1FF0_0000 ~ 0x1FF0_01FF	OB_CP [1]	512 bytes
	Information Block	Boot Loader	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 [1:0] 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 6. Relationship between wait state cycle and HCLK**

Wait State Cycle	HCLK	
	HT32F52231/52241	HT32F52331/52341
0	0 MHz < HCLK ≤ 20 MHz	0 MHz < HCLK ≤ 24 MHz
1	20 MHz < HCLK ≤ 40 MHz	24 MHz < HCLK ≤ 48 MHz

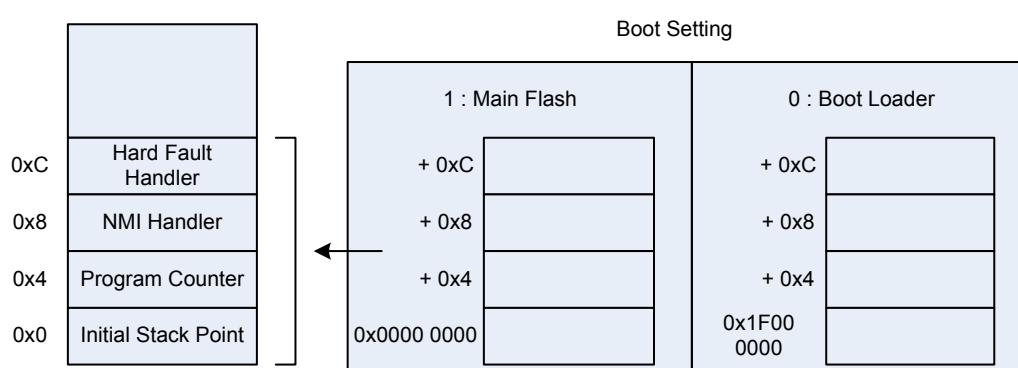
## Booting Configuration

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 7. Booting Modes**

Boot mode selection pin BOOT	Mode	Descriptions	
		0	1
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 8. 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.

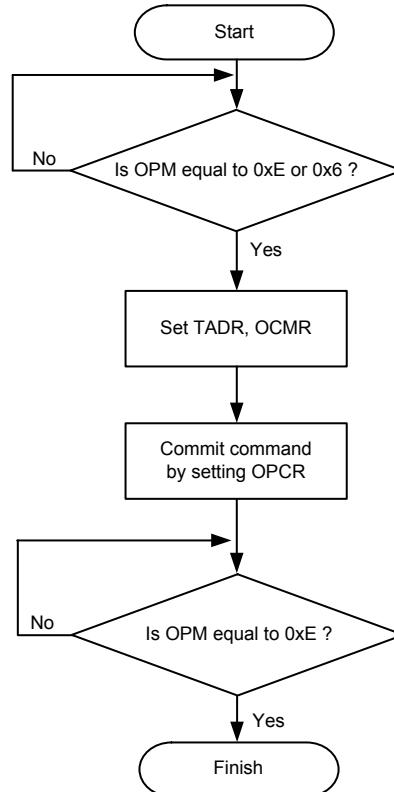


Figure 9. Page Erase Operation Flowchart

## 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.

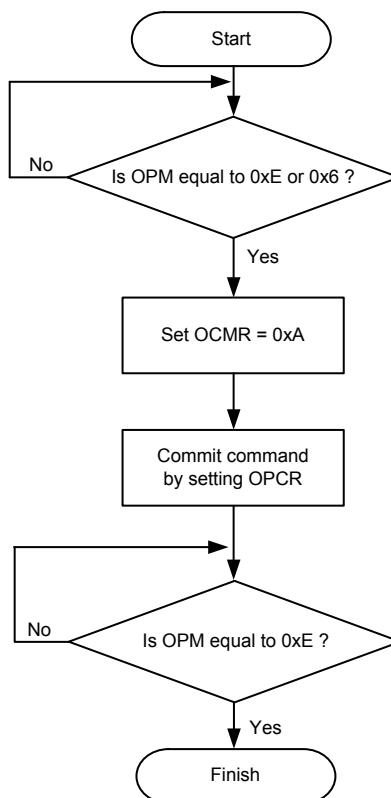


Figure 10. Mass Erase Operation Flowchart

## 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.

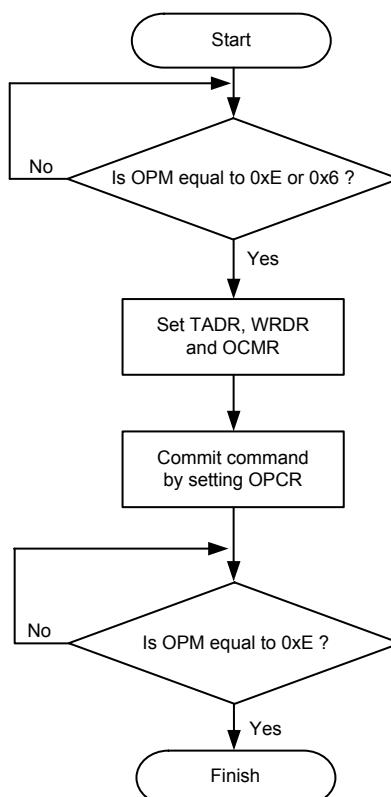


Figure 11. Word Programming Operation Flowchart

## Option Byte Description

The Option Byte can be treated as an independent Flash memory which base address is 0x1FF00\_0000. The following table shows the function description and memory map of Option Byte.

**Table 8. Option Byte Memory Map**

Option Byte	Offset	Description	Reset Value
<b>Option Byte Base Address = 0x1FF0_0000</b>			
OB_PP	0x000	Flash Page Erase/Program Protection (n = 0 ~ 127) For HT32F52231/52241 OB_PP [n] (n = 0 ~ 62) 0: Flash Page n Erase / Program Protection is enabled 1: Flash Page n Erase / Program Protection is disabled	0xFFFF_FFFF
	0x004	OB_PP [n] (n = 63 ~ 127) Reserved	0xFFFF_FFFF
	0x008	For HT32F52331/52341	0xFFFF_FFFF
	0x00C	OB_PP [n] (n = 0 ~ 126) 0: Flash Page n Erase / Program Protection is enabled 1: Flash Page n Erase / Program Protection is disabled OB_PP [n] (n = 127) Reserved	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
	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 9. Access Permission of Protected Main Flash Page**

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

- Notes:**
1. 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.
  2. 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.
  3. The page erase on Option Byte area can disable the page protection of main Flash.
  4. 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 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\_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 10. 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 11. FMC Register Map**

Register	Offset	Description	Reset Value
<b>FMC Base Address = 0x4008_0000</b>			
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	0xFFFF_XXXX
	0x024		0xFFFF_XXXX
	0x028		0xFFFF_XXXX
	0x02C		0xFFFF_XXXX
CPSR	0x030	Flash Security Protection Status Register	0xFFFF_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_0400
CFCR	0x200	Flash Pre-fetch Control Register	0x0000_0051
CIDR0	0x310	Custom ID Register 0	0xFFFF_XXXX
CIDR1	0x314	Custom ID Register 1	0xFFFF_XXXX
CIDR2	0x318	Custom ID Register 2	0xFFFF_XXXX
CIDR3	0x31C	Custom ID Register 3	0xFFFF_XXXX

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

## Register Descriptions

### Flash Target Address Register – TADR

This register specifies the target address of 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						
TADB								
Type/Reset	RW	0						
TADB								
Type/Reset	RW	0						
TADB								
Type/Reset	RW	0						

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 1 KB, the TADB [9:0] will be ignored in order to limit the target address as 1 Kbyte-aligned. For 64 KB main Flash addressing, TADB [31:16] should be zero and TADB [31:15] should be zero for 32 KB. Address from 0x1FF0_0000 to 0x1FF0_03FF is the 1KB Option Byte. This field for available Flash address, it must be under 0xFFFF_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						
WRDB								
Type/Reset	RW	0						
WRDB								
Type/Reset	RW	0						
WRDB								
Type/Reset	RW	0						

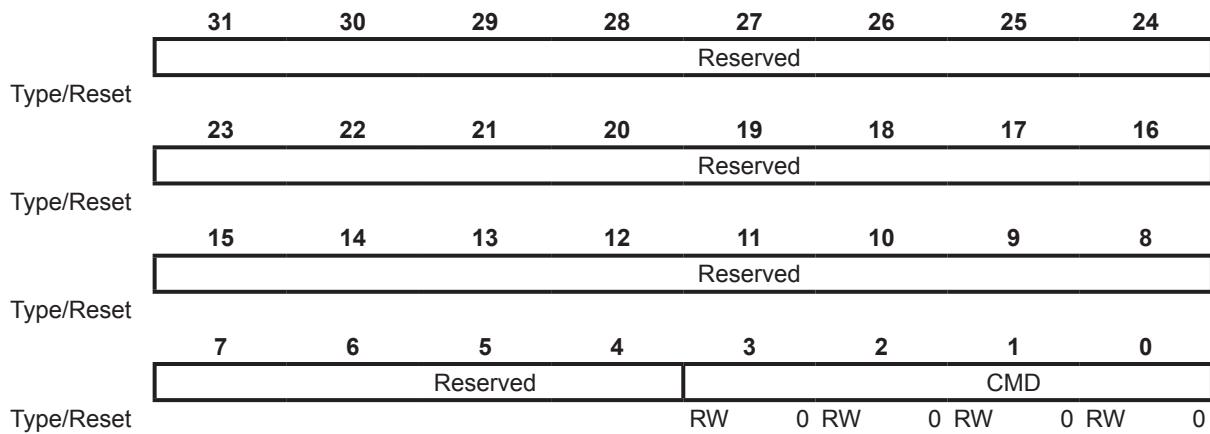
Bits	Field	Descriptions
[31:0]	WRDB	Flash Write Data Bits The data value for programming operation.

## 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	<p>Flash Operation Command</p> <p>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.</p> <table border="1"> <thead> <tr> <th>CMD [3:0]</th><th>Description</th></tr> </thead> <tbody> <tr> <td>0x0</td><td>Idle (default)</td></tr> <tr> <td>0x4</td><td>Word program</td></tr> <tr> <td>0x8</td><td>Page erase</td></tr> <tr> <td>0xA</td><td>Mass erase</td></tr> <tr> <td>Others</td><td>Reserved</td></tr> </tbody> </table>	CMD [3:0]	Description	0x0	Idle (default)	0x4	Word program	0x8	Page erase	0xA	Mass erase	Others	Reserved
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

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

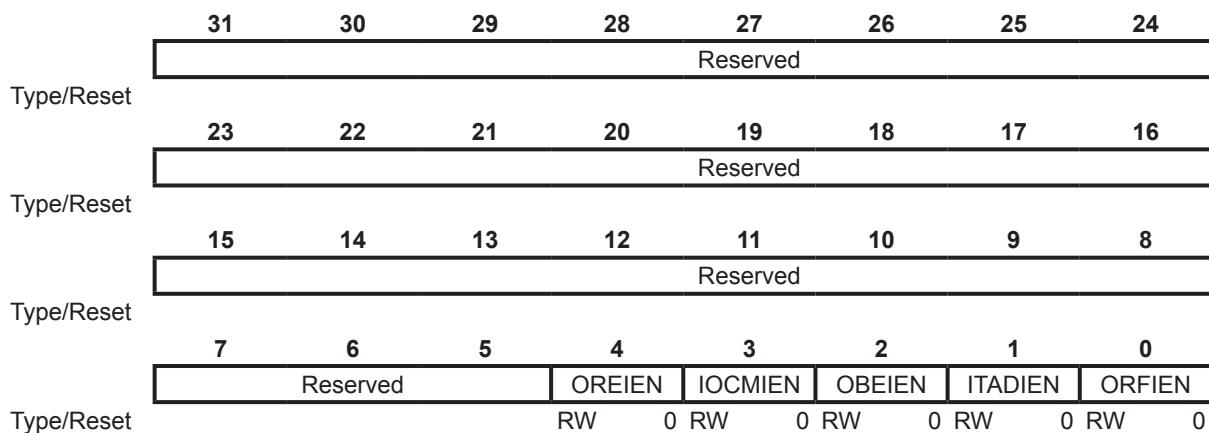
Bits	Field	Descriptions										
[4:1]	OPM	<p>Operation Mode</p> <p>The following table shows the operation mode of FMC. User can commit command which is set by the OCMR register to main flash according to the address alias setting in TADR. The content of TADR, WRDR, and OCMR registers shall be prepared 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> <table border="1"> <thead> <tr> <th>OPM [3:0]</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0x6</td> <td>Idle (default)</td> </tr> <tr> <td>0xA</td> <td>Commit command to main Flash</td> </tr> <tr> <td>0xE</td> <td>All operation finished on main Flash</td> </tr> <tr> <td>Others</td> <td>Reserved</td> </tr> </tbody> </table>	OPM [3:0]	Description	0x6	Idle (default)	0xA	Commit command to main Flash	0xE	All operation finished on main Flash	Others	Reserved
OPM [3:0]	Description											
0x6	Idle (default)											
0xA	Commit command to main Flash											
0xE	All operation finished on main Flash											
Others	Reserved											

## Flash Operation Interrupt Enable Register – OIER

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

Offset: 0x014

Reset value: 0x0000\_0000



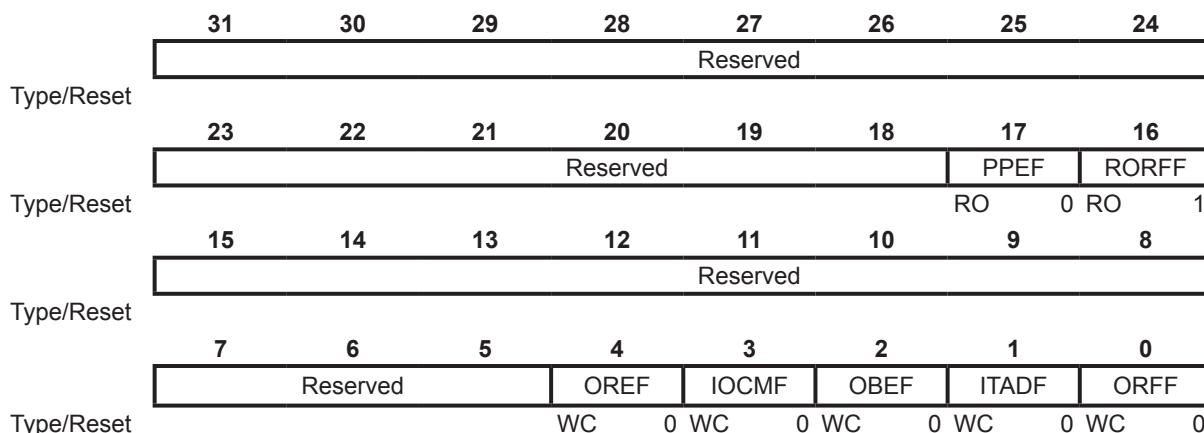
Bits	Field	Descriptions
[4]	OREIEN	Operation Error Interrupt Enable 0: Operation error does not generate an interrupt 1: Operation error generates an interrupt
[3]	IOCMIEN	Invalid Operation Command Interrupt Enable 0: Invalid Operation Command does not generate an interrupt 1: Invalid Operation Command 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



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 OCMD 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: 0xFFFF\_FFFF

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

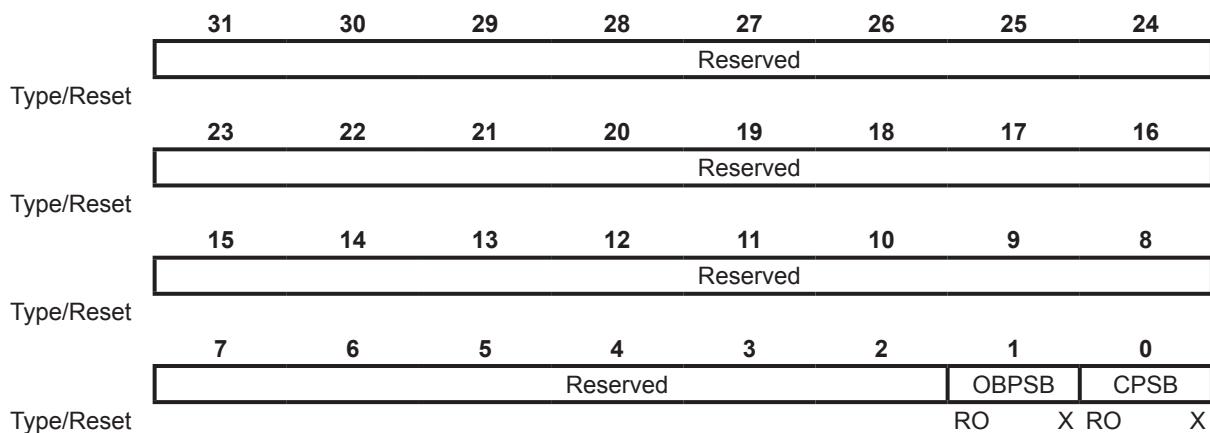
Bits	Field	Descriptions
[127:0]	PPSBn	Page Erase/Program Protection Status Bits (n = 0 ~ 127) PPSB[n] = OB_PP[n] 0: The corresponding pages are protected 1: The corresponding pages are not protected The content of this register is not dynamically updated and will only be reloaded from the Option Byte 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.

## 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\_FFFF



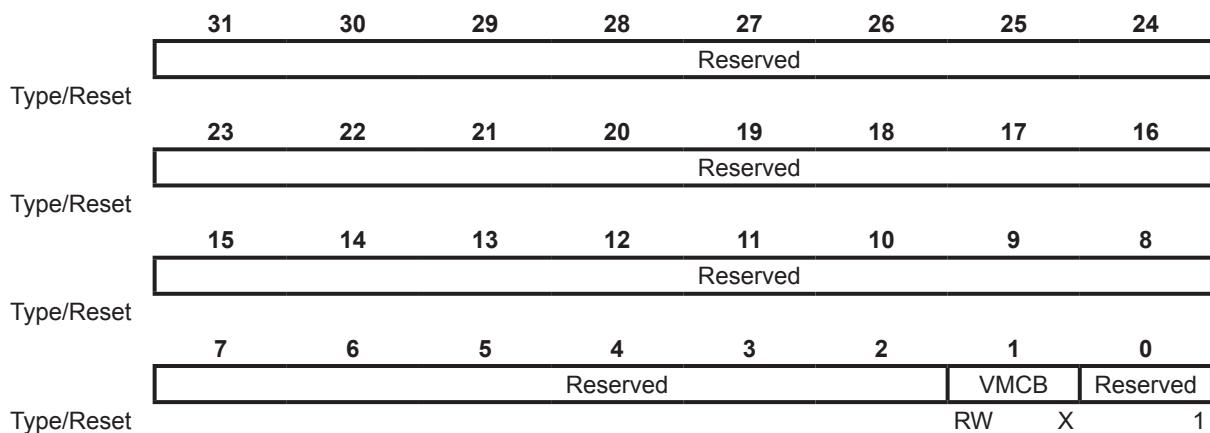
Bits	Field	Descriptions
[1]	OBPSB	Option Byte Page Erase/Program Protection Status Bit 0: The Option Byte page is protected. 1: The Option Byte page is not protected. The reset value of OPBSB is determined by the Option Byte, OB_CP [1].
[0]	CPSB	Flash Security Protection Status Bit 0: Flash Security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



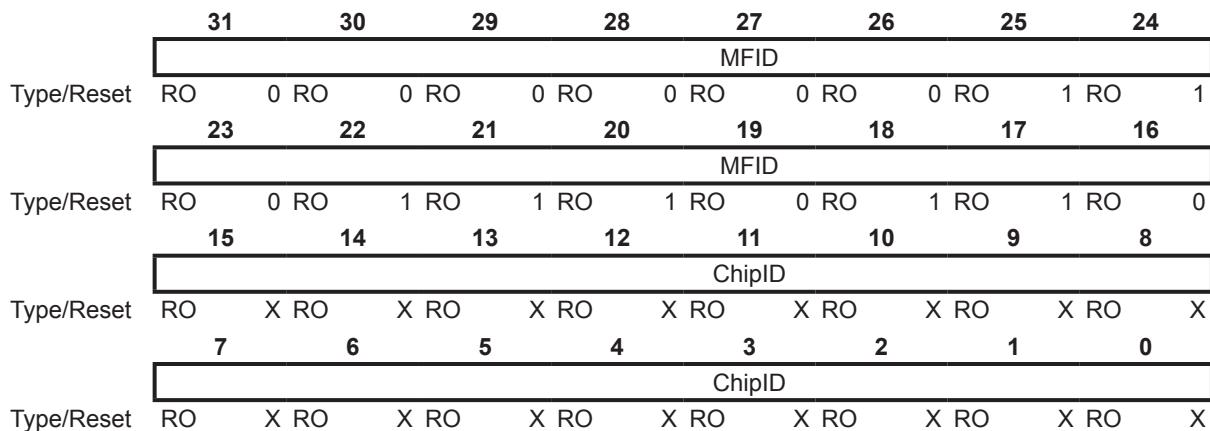
Bits	Field	Descriptions									
[1]	VMCB	<p>Vector Mapping Control Bit</p> <p>The VMCB bits is used to control the mapping source of first 4-word vector (address 0x0 ~ 0xC). The following table shows the vector mapping setting.</p> <table border="1"> <thead> <tr> <th>BOOT</th> <th>VMCB [1]</th> <th>Descriptions</th> </tr> </thead> <tbody> <tr> <td>Low</td> <td>0</td> <td>           Boot Loader mode            The source of the vector mapping is the boot loader area.         </td> </tr> <tr> <td>High</td> <td>1</td> <td>           Main Flash mode            The source of the vector mapping is the main Flash area.         </td> </tr> </tbody> </table>	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.
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.									
<p>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.</p>											

## 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



Bits	Field	Descriptions
[31:16]	MFID	Manufacturer ID Read as 0x0376
[15:0]	ChipID	Chip ID Read the last 4 digital code 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						
23 22 21 20 19 18 17 16								
PNSB								
Type/Reset	RO	0						
15 14 13 12 11 10 9 8								
PNSB								
Type/Reset	RO	0						
7 6 5 4 3 2 1 0								
PNSB								
Type/Reset	RO	X						

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. They indicated the total pages of 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\_0400

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

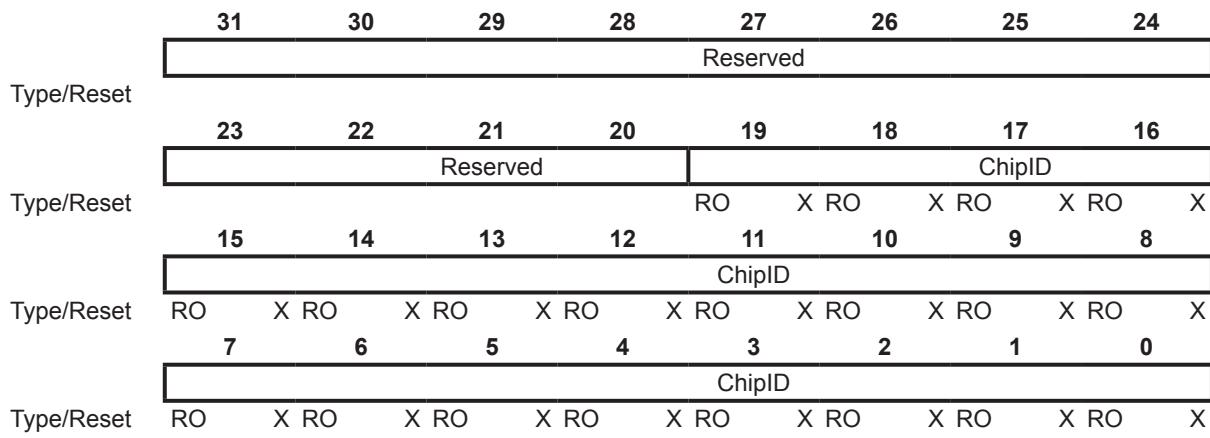
Bits	Field	Descriptions
[31:0]	PSSB	Status Bits of Flash Page Size 0x200: That means the page size is 512 bytes per page. 0x400: That means the page size is 1 KB per page. 0x800: That means the page size is 2 KB per page.

## Device ID Register – DID

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

Offset: 0x18C

Reset value: 0x000X\_XXXX



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

## Flash Pre-fetch Control Register – CFCR

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

Offset: 0x200

Reset value: 0x0000\_0011

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

Bits	Field	Descriptions
[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	
		HT32F52231/52241	HT32F52331/52341
001	0	0 MHz < HCLK ≤ 20 MHz	0 MHz < HCLK ≤ 24 MHz
010	1	20 MHz < HCLK ≤ 40 MHz	20 MHz < HCLK ≤ 48 MHz
Others	Reserved	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
Type/Reset	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
	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
	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

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

## 5 Power Control Unit (PWRCU)

### Introduction

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

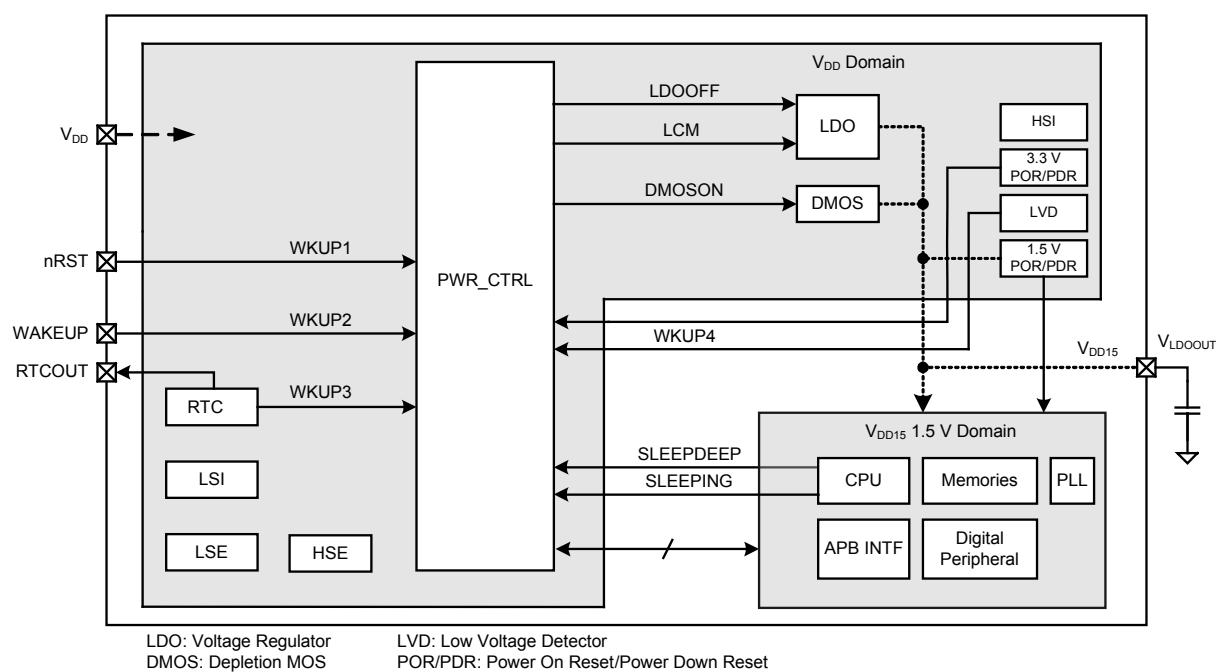


Figure 12. PWRCU Block Diagram

## Features

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

## Functional Descriptions

### $V_{DD}$ Power Domain

#### LDO Power Control

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

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

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

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

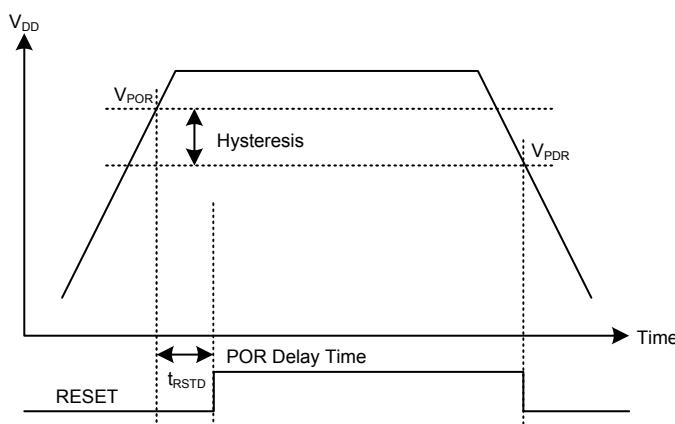
To enter the Deep-Sleep1 mode, the PWRCU will request the LDO to operate in a low current mode, LCM. To enter the Deep-Sleep 2 mode, the PWRCU will turn off the LDO and turn on the DMOS to supply an alternative 1.5 V power.

#### Voltage Regulator

The voltage regulator, LDO, Depletion MOS, DMOS, Low voltage Detector, LVD, High Speed Internal oscillator, HSI, Low Speed Internal RC oscillator, LSI, and the Low Speed External Crystal oscillator, LSE are operated under the  $V_{DD}$  power domain. The LDO can be configured to operate in either normal mode (LDOOFF = 0, LDOLCM = 0,  $I_{OUT}$  = High current mode) or low current mode (LDOOFF = 0, LDOLCM =1,  $I_{OUT}$  = Low current mode) to supply the 1.5 V power. An alternative 1.5 V power source is the output of the DMOS which has low static and driving current characteristics. It is controlled using the DMOSON bit in the PWRCR register. The DMOS output has weak output current and regulation capability and only 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 13. 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<sub>DD</sub> power domain. The HSE crystal oscillator can be switched on or off using the HSEEN bit in the Global Clock Control Register (GCCR). The HSE clock can then be used directly as the system clock source or be used as the PLL input clock.

### LSE, LSI and RTC

The Real Time Clock Timer clock source can be derived from either the Low Speed Internal RC oscillator, LSI, or the Low Speed External Crystal oscillator, LSE. Before entering the power saving mode by executing 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.

### Isolation Cells

When the device resumes operation from the 1.5 V power, either by Hardware or Software, access to the RTC registers in the V<sub>DD</sub> power domain 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 VDDISO bit to 1 in the LPCR register of the Clock Control Unit.

## 1.5 V Power Domain

The main functions that include the APB interface for the V<sub>DD</sub> domain, CPU core logic, AHB/APB peripherals and memories and so on are located in this power domain. Once the 1.5 V is powered up, the POR will generate a reset sequence on 1.5 V power domain. Subsequently, to enter the expected power saving mode, the associated control bits including the LD0OFF, DMOSON and LDOLCM 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 AHBPREG 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 12. Table 1 Operation Mode Definitions**

Mode name	Hardware Action
Run	After system reset, CPU fetches instructions to execute.
Sleep	CPU clock will be stopped. Peripherals, Flash and SRAM clocks can be stopped by setting.
Deep-Sleep1~2	Stop all clocks in the 1.5 V power domain. Disable HSI, HSE, and PLL. 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 FMSEN 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 FMSEN and SRAMEN bits in the AHBCCR register to minimize power consumption. To enter the Sleep mode, it is only CPU executes a WFI or WFE instruction and lets the SLEEPDEEP signal to 0. The system will exit from the Sleep mode via any interrupt or event trigger. The accompanying table provides more information about the power saving modes.

**Table 13. 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 <sup>(1)</sup> or Any interrupt (NVIC on) or Any interrupt with SEVONPEND = 1 (NVIC off)
		1	0	0	Any EXTI in event mode or RTC wakeup or LVD wakeup <sup>(2)</sup> or WAKEUP pin rising edge or USB resume <sup>(3)</sup>
		1	X	1	RTC wakeup or LVD wakeup <sup>(2)</sup> or WAKEUP pin rising edge
		1	1	0	RTC wakeup or LVD wakeup <sup>(2)</sup> 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.
3. The USB resume function is only available for the HT32F52331/HT32F52341 devices.

### 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 ( $V_{DD}/V_{DDA}$ ).

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

**Table 14. Power Status After System Reset**

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

## Register Map

The following table shows the PWRCU registers and reset values. Note all the registers in this unit are located in the V<sub>DD</sub> power domain.

**Table 15. PWRCU Register Map**

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

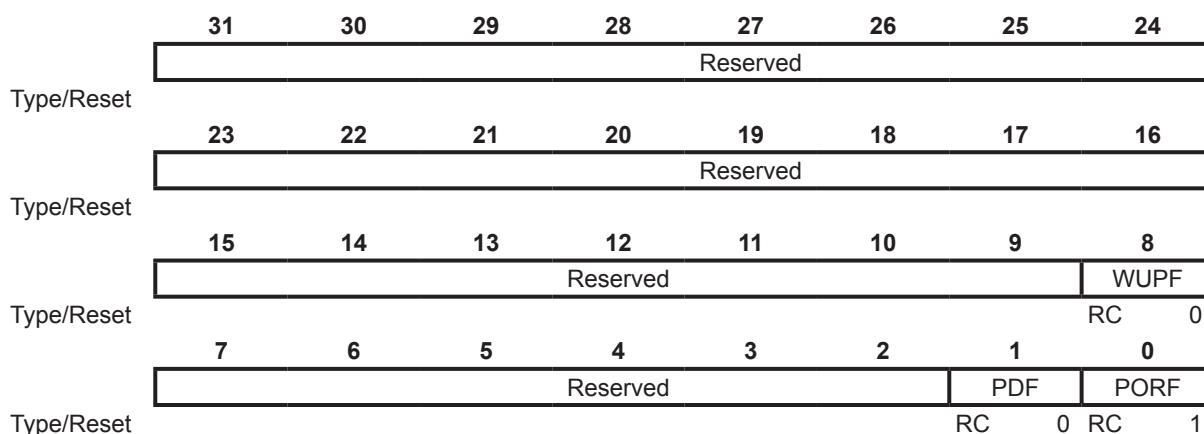
## Register Descriptions

### Power Control Status Register – PWRSR

This register indicates power control status.

Offset: 0x100

Reset value: 0x0000\_0001 (Reset only by V<sub>DD</sub> domain power on reset)



Bits	Field	Descriptions
[8]	WUPF	External WAKEUP Pin Flag 0: The Wakeup pin is not asserted 1: The Wakeup pin is asserted This bit is set by hardware when the WAKEUP pin 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.
[1]	PDF	Power Down Flag 0: Wakeup from abnormal V <sub>DD15</sub> shutdown (Loss of V <sub>DD15</sub> is unexpected) 1: Wakeup from Power-Down mode. The loss of V <sub>DD15</sub> is under expectation. This bit is set by hardware when the system has successfully entered the Power-Down mode This bit is cleared by software read.

Bits	Field	Descriptions
[0]	PORF	<p>Power On Reset Flag</p> <p>0: V<sub>DD</sub> Power Domain reset does not occur 1: V<sub>DD</sub> Power Domain reset occurs</p> <p>This bit is set by hardware when V<sub>DD</sub> power on reset occurs, either a hardware power on reset or software reset. The bit is cleared by software read. This bit must be cleared after the system is first powered on, otherwise it will be impossible to detect when a V<sub>DD</sub> Power Domain reset has been triggered. When this bit is read as 1, a read software loop must be implemented until the bit returns again to 0. This software loop is necessary to confirm that the V<sub>DD</sub> Power Domain is ready for access.</p>

### Power Control Register – PWRCR

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

Offset: 0x104

Reset value: 0x0000\_0000 (Reset only by V<sub>DD</sub> domain power on 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
Type/Reset	DMOSTS		Reserved	V15RDYSC		Reserved	WUPIEN	WUPEN
	RO	0		RW	0		RW	0
Type/Reset	7	6	5	4	3	2	1	0
Type/Reset	DMOSON		Reserved		LDOOFF	LDOLCM	Reserved	PWRST
	RW	0		RW	0	RW	WO	0

Bits	Field	Descriptions
[15]	DMOSTS	<p>Depletion MOS Status</p> <p>This bit is set to 1 if the DMOSON bit in this register has been set to 1.</p> <p>This bit is cleared to 0 if the DMOSON bit has been set to 0 or if a POR/PDR reset occurred.</p>
[12]	V15RDYSC	<p>V<sub>DD</sub>15 Ready Source Selection.</p> <p>0: VDDISO bit in the LPCR register located in the CKCU 1: V<sub>DD</sub>15 POR</p> <p>Setting this bit to determine what control signal of isolation cells is used to disable the isolation function of the V<sub>DD</sub>15 to V<sub>DD</sub> power domain level shifter.</p>
[9]	WUPIEN	<p>External WAKEUP Pin Interrupt Enable</p> <p>0: Disable WAKEUP pin interrupt function 1: Enable WAKEUP pin interrupt function</p> <p>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.</p>

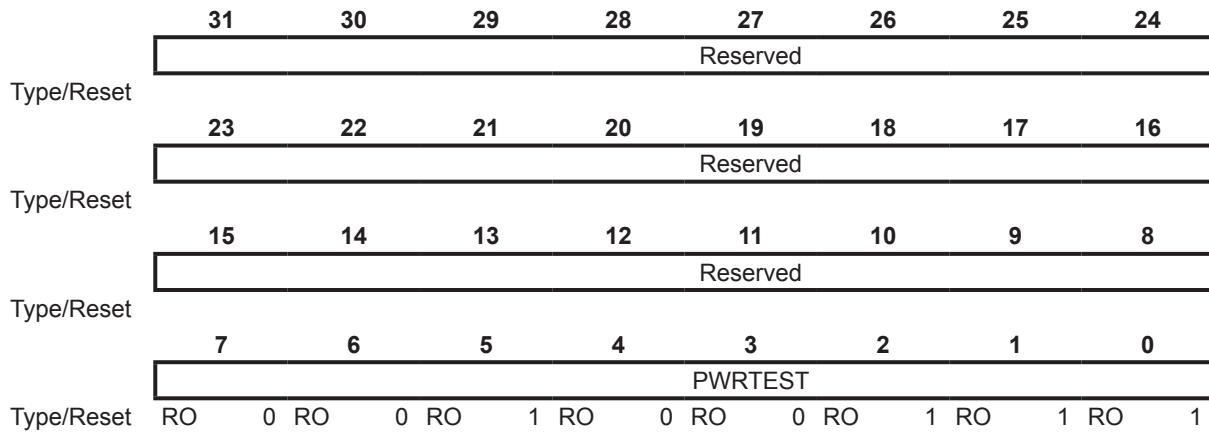
Bits	Field	Descriptions
[8]	WUPEN	<p>External WAKEUP Pin Enable</p> <p>0: Disable WAKEUP pin function.          1: Enable WAKEUP pin function.</p> <p>The Software can set the WUPEN bit as 1 to enable the WAKEUP pin function before entering the power saving mode. When WUPEN = 1, a rising edge on the WAKEUP 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 WAKEUP pin alternate function should first be selected by configuring the PBCFG12 bit field in the GPBCFGHR register to 0x0F before the WAKEUP pin is used. The corresponding pull-up function on the WAKEUP pin should also be disabled by clearing the PBPU[12] bit in the PBPUR register to 0 while the pull-down function should be enabled by setting the PBPD[12] bit in the PBPDR register to 1.</p> <p>Note: This bit is reset by a <math>V_{DD}</math> Power Domain reset. Because this bit is located in the <math>V_{DD}</math> Power 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 <math>V_{DD}</math> Power Domain ISO signal has been disabled. This means that the bit cannot be immediately set by software after a system reset finished and the <math>V_{DD}</math> Power 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 <math>V_{DD}</math> power domain reset. If the DMOSON bit is set to 1, the LDO will automatically be turned off when the CPU enters the Deep-Sleep mode.</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 <math>V_{DD15}</math> power is available.          1: The LDO is turned off when the CPU enters the Deep-Sleep mode (SLEEPDEEP=1). The <math>V_{DD15}</math> 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 <math>V_{DD}</math> power domain reset.</p>
[0]	PWRST	<p><math>V_{DD}</math> Power Domain Software Reset</p> <p>0: No action          1: <math>V_{DD}</math> Power Domain Software Reset is activated.</p> <p>It will reset all the related RTC and PWRCU registers.</p>

## V<sub>DD</sub> Power Domain Test Register – PWRTEST

This register specifies a read-only value for the software to recognize whether V<sub>DD</sub> Power Domain is ready for access.

Offset: 0x108

Reset value: 0x0000\_0027



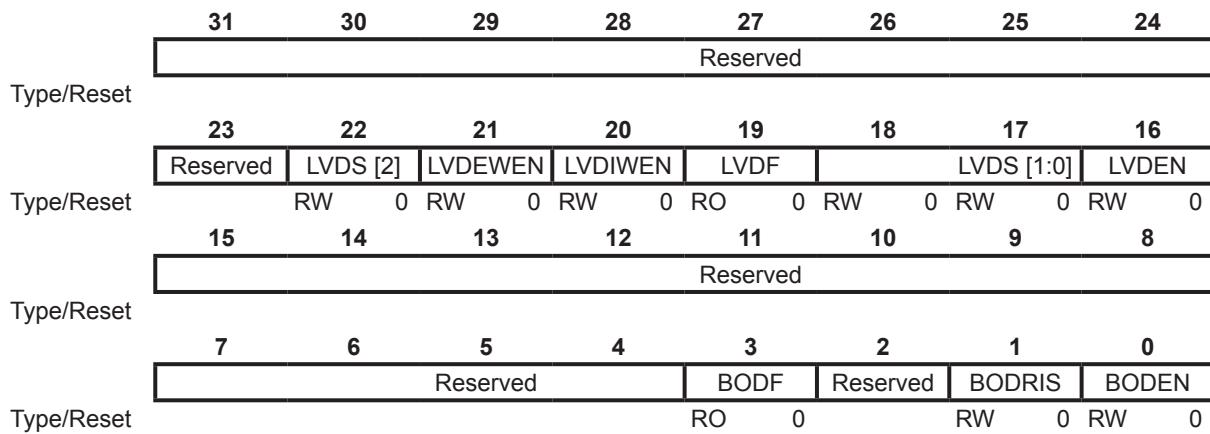
Bits	Field	Descriptions
[7:0]	PWRTEST	V <sub>DD</sub> Power Domain Test Bits A constant 0x27 will be read when the V <sub>DD</sub> Power Domain is ready for CPU access.

## Low Voltage / Brown Out Detect Control and Status Register – LVCSR

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

Offset: 0x110

Reset value: 0x0000\_0000 (Reset only by V<sub>DD</sub> domain power on reset)



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 <sub>DD</sub> is higher than the specific voltage level 1: V <sub>DD</sub> 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]	LVDEN	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 $V_{DD}$ 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

## 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.

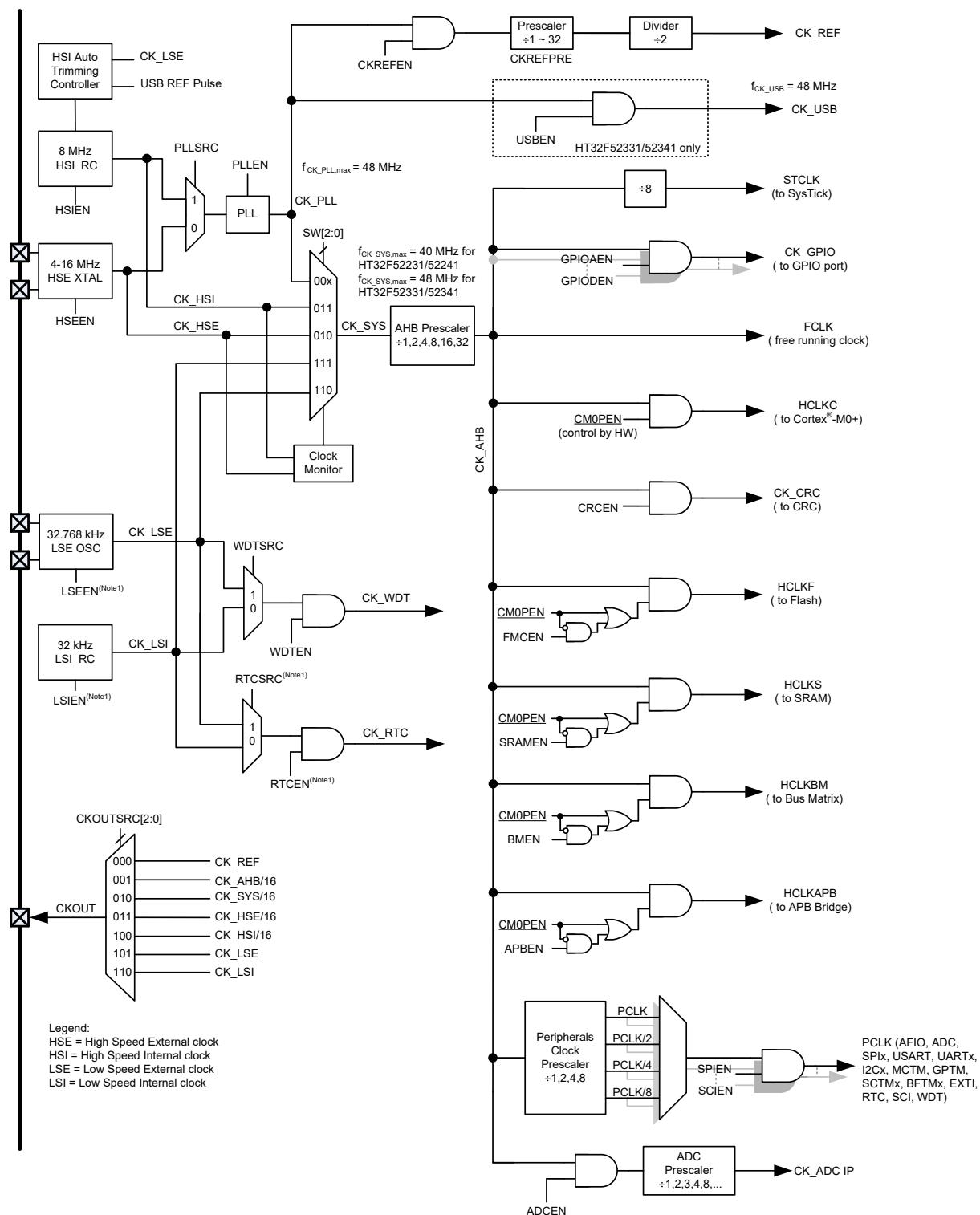


Figure 14. CKCU Block Diagram

## 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.

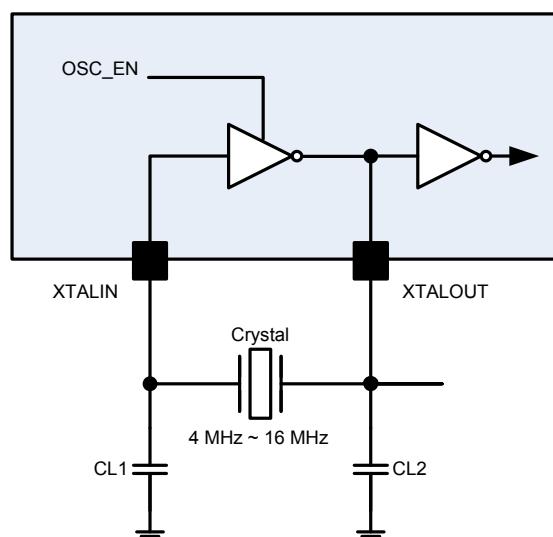


Figure 15. External Crystal, Ceramic, and Resonators for HSE

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  $V_{DD} = 3.3$  V and  $T_A = 25^\circ C$ . 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 a 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.

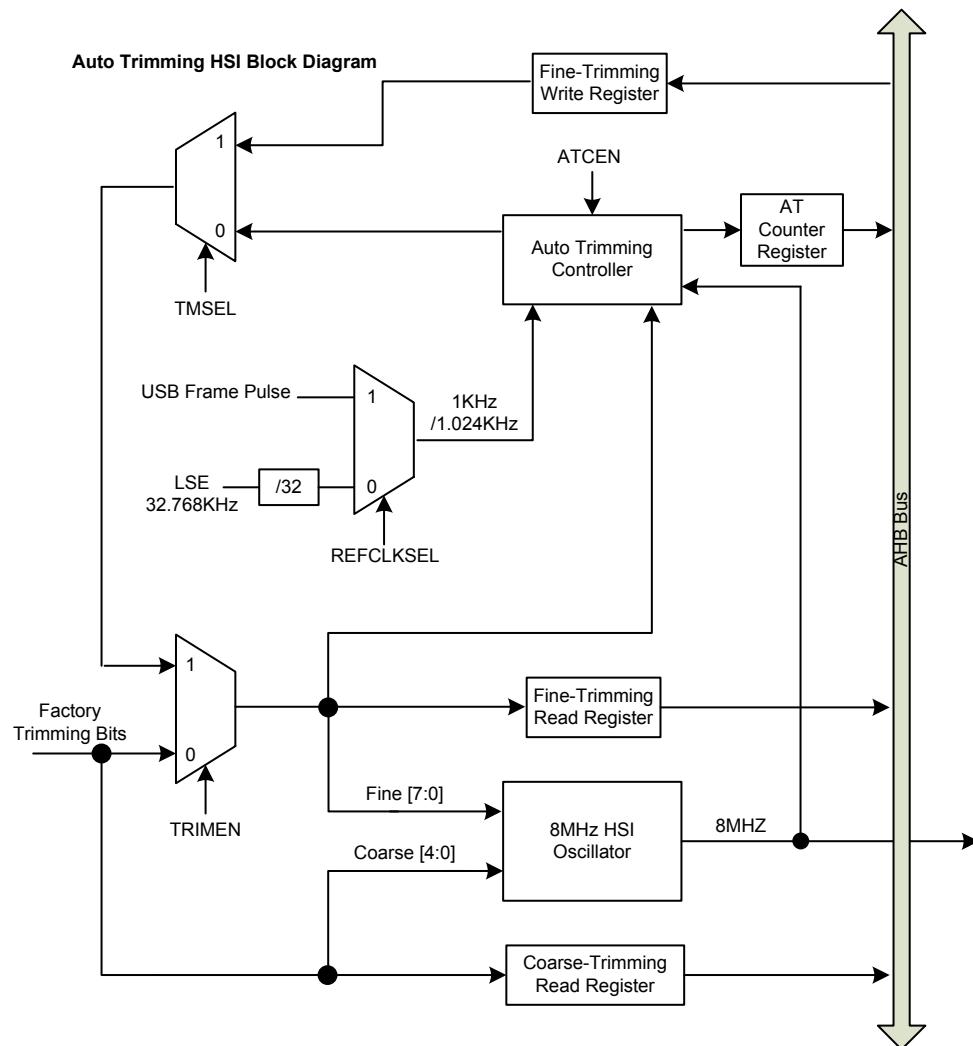
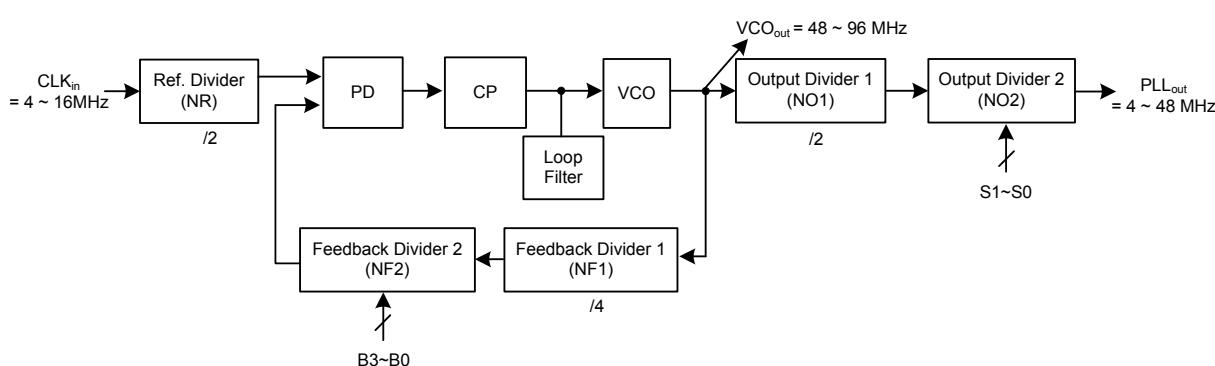


Figure 16. HSI Auto Trimming Block Diagram

## 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 17. PLL Block Diagram**

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

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

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

Considering the duty cycle of 50%, both input and output frequencies are divided by 2. If a given CLK<sub>in</sub> 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 the expense of the 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<sub>OUT</sub> 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<sub>OUT</sub> formula.

The PLL can be switched on or off by using the PLLLEN 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 16. Output Divider2 Value Mapping**

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

**Table 17. Feedback Divider2 Value Mapping**

Feedback divider2 setup bits B[3:0] (PFBD bits in the PLLCFG 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.

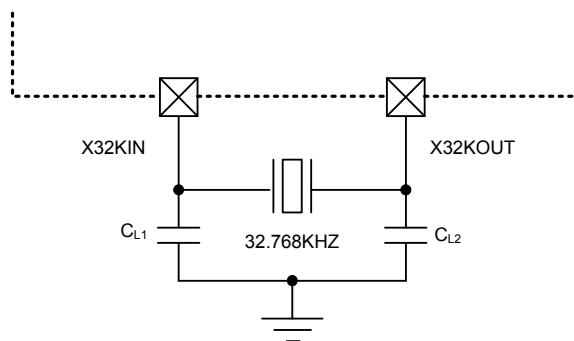


Figure 18. External Crystal, Ceramic, and Resonators for LSE

## Low Speed Internal RC Oscillator – LSI

The low speed internal RC oscillator with a frequency of about 32 kHz produces a low power clock source for the 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 data sheet. 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 oscillators 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 the 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 configuring 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. (PLLEN, 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. (PLLEN, 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. (PLLEN)

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 clock failure 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 section, AFIO, 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 18. CKOUT Clock Source**

CKOUTSRC[2:0]	Clock Source
000	CK_REF = CK_PLL / (CKREFPRE + 1) / 2
001	HCLK / 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 19. CKCU Register Map**

Register	Offset	Description	Reset Value
GCFGGR	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
PLLCFGGR	0x018	PLL Configuration Register	0x0000_0000
PLLCR	0x01C	PLL Control Register	0x0000_0000
AHBCFGGR	0x020	AHB Configuration Register	0x0000_0000
AHBCCR	0x024	AHB Clock Control Register	0x0000_0065
APBCFGGR	0x028	APB Configuration Register	0x0000_0000
APBCCR0	0x02C	APB Clock Control Register 0	0x0000_0000
APBCCR1	0x030	APB Clock Control Register 1	0x0000_0000
CKST	0x034	Clock Source Status Register	0x0100_0003
APBPCSR0	0x038	APB Peripheral Clock Selection Register 0	0x0000_0000
APBPCSR1	0x03C	APB Peripheral Clock Selection Register 1	0x0000_0000
HSICR	0x040	HSI Control Register	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	1
	7	6	5	4	3	2	1	0
	Reserved					RW	0	RW
Type/Reset							1	RW
								0

Bits	Field	Descriptions
[31:29]	LPMOD	<p>Lower Power Mode Status</p> <p>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</p> <p>Set and reset by hardware.</p>
[23:22]	USBPRE	<p>USB Clock Prescaler Selection</p> <p>00: CK_USB = CK_PLL      01: CK_USB = CK_PLL / 2      Others: Reserved</p> <p>Set and reset by software to control the USB clock prescaler setting.      Note that the USB interface is only available for the HT32F52331/ HT32F52341 devices.</p>
[15:11]	CKREFPRE	<p>CK_REF Clock Prescaler Selection</p> <p><math>CK_{REF} = CK_{PLL} / (CKREFPRE + 1) / 2</math></p> <p>00000: <math>CK_{REF} = CK_{PLL} / 2</math>      00001: <math>CK_{REF} = CK_{PLL} / 4</math></p> <p>...      11111: <math>CK_{REF} = CK_{PLL} / 64</math></p> <p>Set and reset by software to control the CK_REF clock prescaler setting.</p>
[8]	PLLSRC	<p>PLL Clock Source Selection</p> <p>0: External 4 ~ 16 MHz crystal oscillator clock is selected (HSE)      1: Internal 8 MHz RC oscillator clock is selected (HSI)</p> <p>Set and reset by software to control the PLL clock source.</p>

Bits	Field	Descriptions
[2:0]	CKOUTSRC	CKOUT Clock Source Selection 000: CK_REF is selected where $CK\_REF = CK\_PLL / (CKREFPRE+1) / 2$ 001: $(CK\_AHB / 16)$ 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	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved						PSRCEN	CKMEN	
	15	14	13	12	11	10	9	8	
Type/Reset	Reserved				HSIEN	HSEEN	PLLEN	HSEGAIN	
	7	6	5	4	3	2	1	0	
Type/Reset	Reserved				SW				

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 a low or high state, the internal hardware will switch the system clock to the internal high speed HSI RC clock. The way to recover the system clock is by an external reset, a power on reset or clearing the CKSF by software.
[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 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.

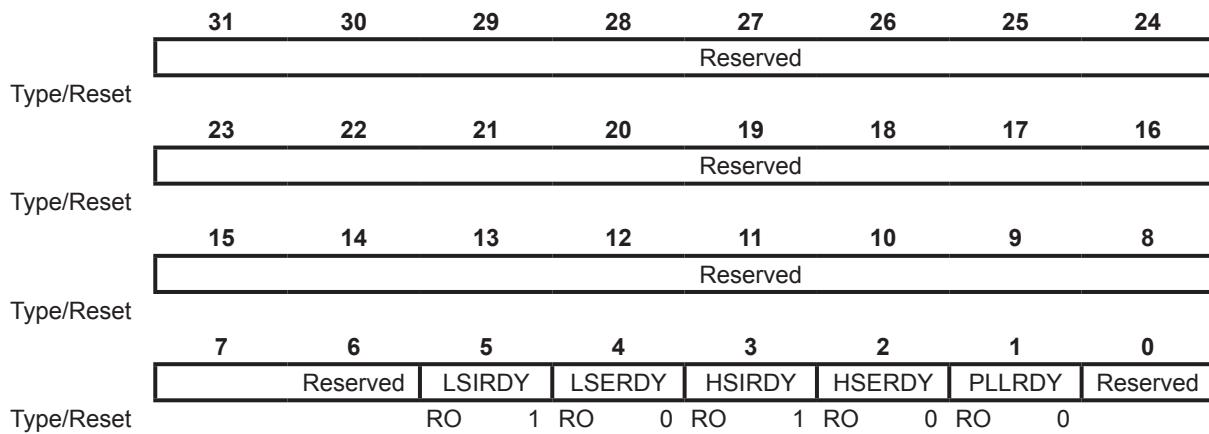
Bits	Field	Descriptions
[9]	PLLEN	PLL Enable 0: PLL is disabled 1: PLL is enabled Set and reset by software. This bit cannot be reset if the PLL clock is used as the system clock.
[8]	HSEGAIN	External High Speed Oscillator Gain Selection 0: HSE is in low gain mode 1: HSE is in high gain mode
[2:0]	SW	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 switching the system clock using the SW bits, the system clock will not be 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



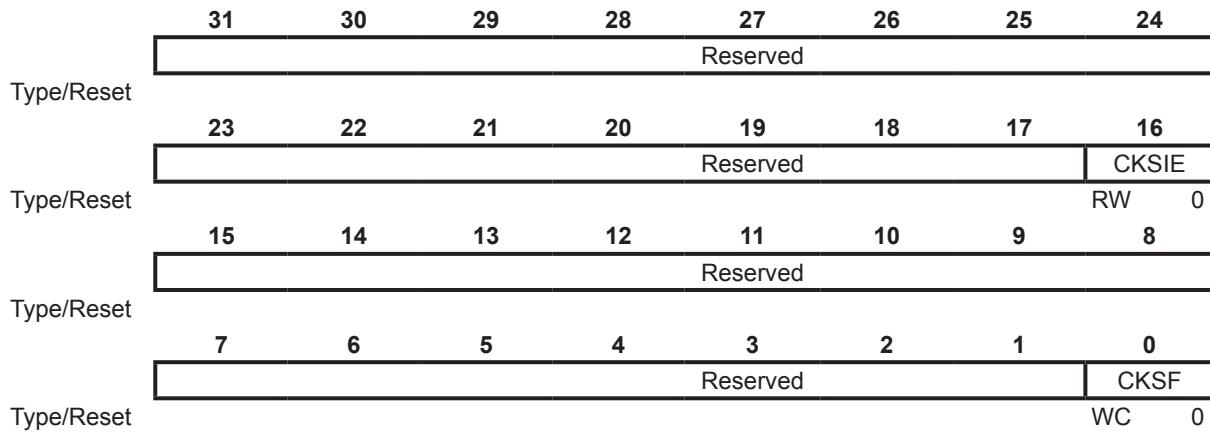
Bits	Field	Descriptions
[5]	LSIRDY	Internal Low Speed Oscillator Ready Flag 0: Internal 32 kHz RC oscillator is not ready 1: Internal 32 kHz RC oscillator is ready Set by hardware to indicate whether the LSI is stable to be used.
[4]	LSEIRDY	External Low Speed Oscillator Ready Flag 0: External 32,768 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 clock is ready Set by hardware to indicate whether the HSI is stable to be used.
[2]	HSEIRDY	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]	PLLIRDY	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



Bits	Field	Descriptions
[16]	CKSIE	<p>Clock Stuck Interrupt Enable</p> <p>0: Disable clock failure interrupt            1: Enable clock failure interrupt</p> <p>Set and reset by software to enable or disable the clock failure interrupt caused by the clock monitor function.</p>
[0]	CKSF	<p>Clock Stuck Interrupt Flag</p> <p>0: Clock works normally            1: HSE clock is stuck</p> <p>Reset by software (Write 1 clear). Set by hardware when the HSE clock is stuck and the CKSIE 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	RW	0
Type/Reset	23	22	21	20	19	18	17	16
	PFBD	POTD			Reserved			
Type/Reset	RW	0	RW	0	RW	0	RW	0
Type/Reset	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset	7	6	5	4	3	2	1	0
	Reserved							

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

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

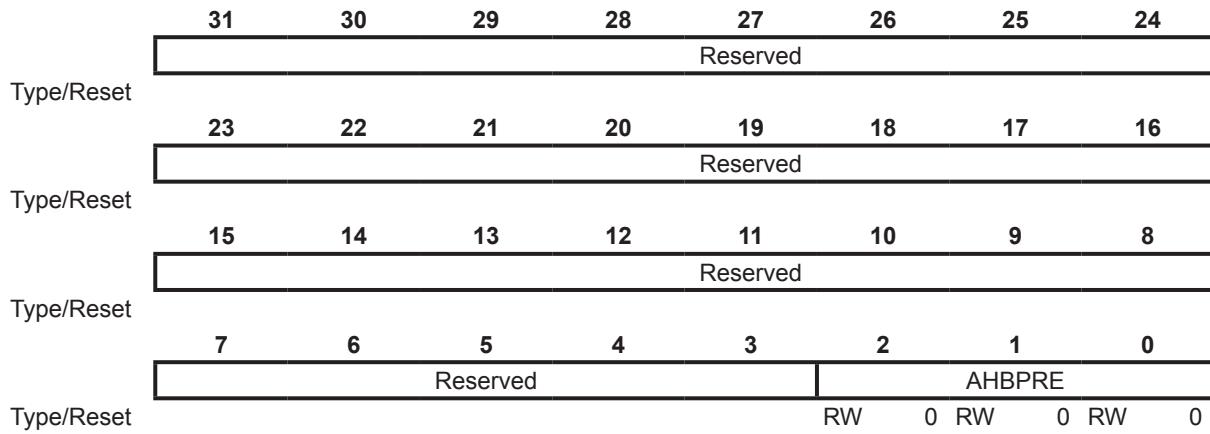
Bits	Field	Descriptions
[31]	PLLBPS	PLL Bypass Mode Enable 0: Disable PLL Bypass mode 1: Enable PLL Bypass mode which acts as FOUT = FIN

## 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\_0065

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

Bits	Field	Descriptions
[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.
[11]	CKREFEN	CK_REF Clock Enable 0: CK_REF clock is disabled 1: CK_REF clock is enabled Set and reset by software
[10]	USBEN	USB Clock Enable 0: USB clock is disabled 1: USB clock is enabled Set and reset by software. Note that the USB interface is only available for the HT32F52331/ HT32F52341 devices.
[6]	APBEN	APB bridge Clock Enable 0: APB bridge clock is automatically disabled by hardware during Sleep mode 1: APB bridge clock is always enabled during Sleep mode Set and reset by software. User can set the APBEN bit to 0 to reduce the power consumption if the APB bridge is unused during Sleep mode.

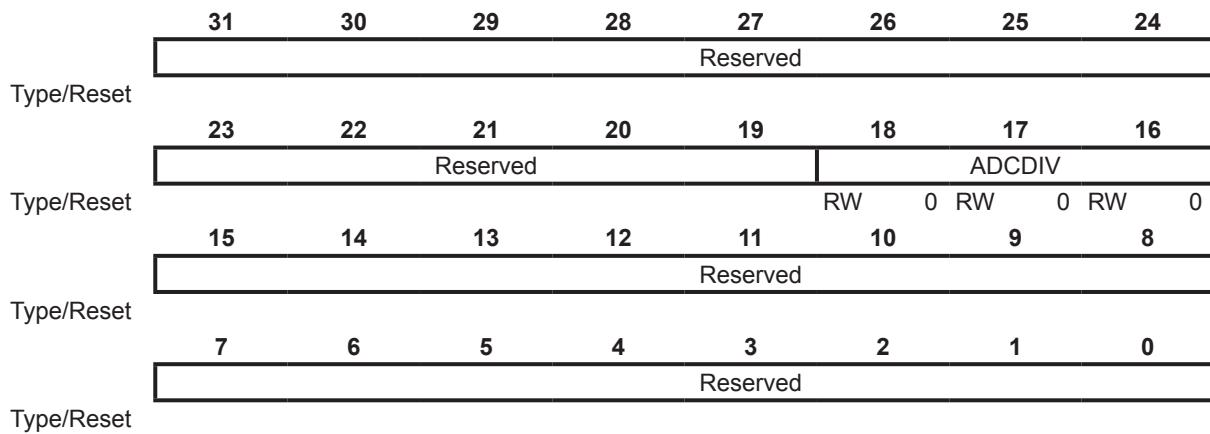
<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>
[5]	BMEN	<p>Bus Matrix Clock Enable</p> <p>0: Bus Matrix clock is automatically disabled by hardware during Sleep mode 1: Bus Matrix clock is always enabled during Sleep mode</p> <p>Set and reset by software. User can set the BMEN bit to 0 to reduce the power consumption if the bus matrix is unused during Sleep mode.</p>
[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 power consumption if the Flash Memory is unused during Sleep mode.</p>

## APB Configuration Register – APBCFGR

This register specifies the ADC conversion clock frequency.

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							
	RW 0							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	RW 0							
Type/Reset	15	14	13	12	11	10	9	8
	EXTIEN	AFIOEN	Reserved		UR1EN	UR0EN	Reserved	USRREN
	RW 0	RW 0	RW 0		RW 0	RW 0	RW 0	
Type/Reset	7	6	5	4	3	2	1	0
	Reserved		SPI1EN	SPI0EN	Reserved		I2C1EN	I2C0EN
Type/Reset	RW 0		RW 0	RW 0	RW 0		RW 0	RW 0

Bits	Field	Descriptions
[24]	SCIEN	Smart Card Interface Clock Enable 0: SCI clock is disabled 1: SCI clock is enabled Set and reset by software. Note that the SCI function is only available for the HT32F52331/ HT32F52341 devices.
[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.
[8]	USRREN	USART Clock Enable 0: USART clock is disabled 1: USART 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.

<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>
[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.

## 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	SCTM3EN	SCTM2EN	SCTM1EN	SCTM0EN		Reserved		ADCCEN
	RW	0	RW	0	RW	0		RW 0
	23	22	21	20	19	18	17	16
Type/Reset					Reserved		BFTM1EN	BFTM0EN
							RW 0	RW 0
	15	14	13	12	11	10	9	8
Type/Reset					Reserved			GPTMEN
								RW 0
Type/Reset	7	6	5	4	3	2	1	0
	Reserved	VDDREN	Reserved	WDTREN		Reserved		MCTMEN
		RW 0		RW 0				RW 0

Bits	Field	Descriptions
[31]	SCTM3EN	SCTM3 Clock Enable 0: SCTM3 clock is disabled 1: SCTM3 clock is enabled Set and reset by software.
[30]	SCTM2EN	SCTM2 Clock Enable 0: SCTM2 clock is disabled 1: SCTM2 clock is enabled Set and reset by software.
[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.
[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.

<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>
[8]	GPTMEN	GPTM Clock Enable 0: GPTM clock is disabled 1: GPTM clock is enabled Set and reset by software.
[6]	VDDREN	$V_{DD}$ Domain Clock Enable for Registers Access 0: $V_{DD}$ Domain Register access clock is disabled 1: $V_{DD}$ Domain Register access clock is enabled Set and reset by software.
[4]	WDTREN	Watchdog Timer Clock Enable for Registers Access 0: Register access clock is disabled 1: Register access clock is enabled Set and reset by software.
[0]	MCTMEN	MCTM Clock Enable 0: MCTM clock is disabled 1: MCTM 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
	23	22	21	20	19	18	17	16
Type/Reset				Reserved			HSEST	
						RO	0	RO
	15	14	13	12	11	10	9	8
Type/Reset				Reserved			PLLST	
					RO	0	RO	0
	7	6	5	4	3	2	1	0
Type/Reset				Reserved			CKSWST	
					RO	0	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
Type/Reset	RW	0	RW	0	RW	0	RW	0
	23	22	21	20	19	18	17	16
Type/Reset	Reserved		GPTMPCLK		Reserved		MCTMPCLK	
		RW	0	RW	0	RW	0	RW
Type/Reset	15	14	13	12	11	10	9	8
	BFTM1PCLK		BFTM0PCLK		Reserved			
Type/Reset	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
Type/Reset	SPI1PCLK		SPI0PCLK		I2C1PCLK		I2C0PCLK	
	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
[25:24]	USRPCCLK	USART Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[21:20]	GPTMPCLK	GPTM Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[17:16]	MCTMPCLK	MCTM 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]	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	RW	0	RW	0	RW	0	RW	0
	23	22	21	20	19	18	17	16
	Reserved						SCIPCLK	
Type/Reset							RW	0
	15	14	13	12	11	10	9	8
Type/Reset	RW	0	RW	0	RW	0	Reserved	
	7	6	5	4	3	2	1	0
Type/Reset	Reserved		ADCCPCLK		EXTIOPCLK		AFIOPCLK	
		RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[31:30]	SCTM3PCLK	SCTM3 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]	SCTM2PCLK	SCTM2 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]	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

Bits	Field	Descriptions
[17:16]	SCIPCLK	SCI 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 Note that the SCI function is only available for the HT32F52331/ HT32F52341 devices.
[15:14]	VDDRPCLK	$V_{DD}$ Domain Register Access Clock Selection 00: PCLK = CK_AHB / 4 01: PCLK = CK_AHB / 8 10: PCLK = CK_AHB / 16 11: PCLK = CK_AHB / 32 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[13:12]	WDTRPCLK	WDT Register Access Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[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: 0XXXX\_0000 where X is undefined

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

Bits	Field	Descriptions
[28:24]	HSICOARSE	HSI Clock Coarse Trimming Value These bits are initialized automatically at startup. They are adjusted by factory trimming and cannot 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: 32.768kHz external low speed clock – LSE 1: 1kHz USB frame pulse This bit is used to select the reference clock for the HSI Auto Trimming Controller, ATC.
[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	Auto Trimming Controller Enable 0: Disable Auto Trimming Controller 1: Enable Auto Trimming Controller

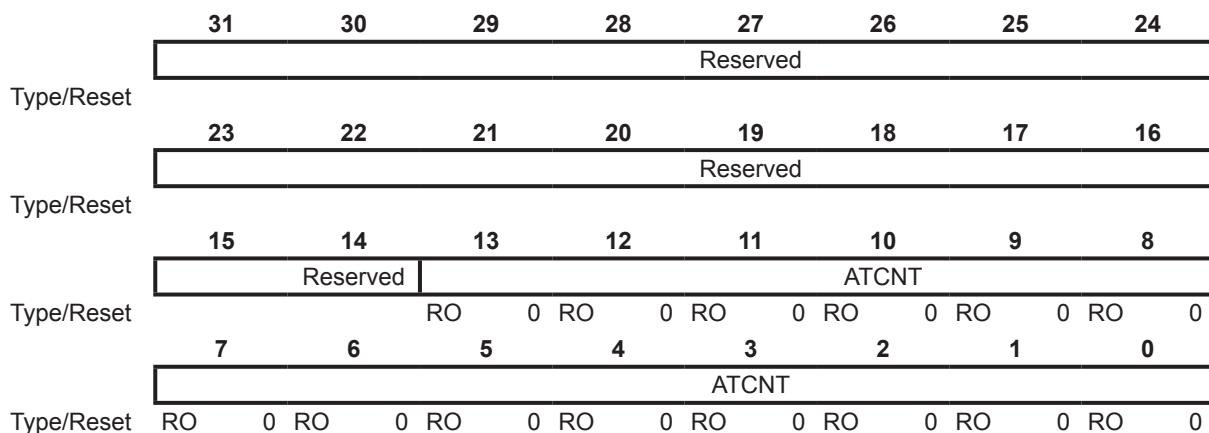
Bits	Field	Descriptions
[0]	TRIMEN	<p>Trimming Enable</p> <p>0: HSI Trimming is disabled 1: HSI Trimming is enabled</p> <p>The bit enables the HSI RC oscillator trimming function by the ATC hardware or user program.</p>

### HSI Auto Trimming Counter Register – HSIATCR

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

Offset: 0x044

Reset value: 0x0000\_0000



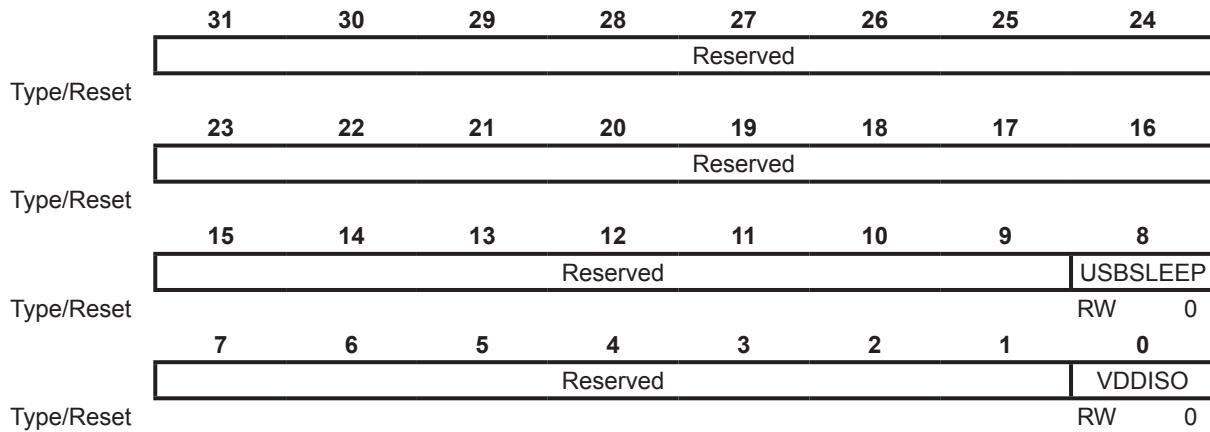
Bits	Field	Descriptions
[13:0]	ATCNT	<p>Auto Trimming Counter</p> <p>These bits are the counter value of the HSI auto trimming controller.</p>

## Low Power Control Register – LPCR

This register specifies the low power control.

Offset: 0x300

Reset value: 0x0000\_0000



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]	VDDISO	VDD Domain Isolation Control 0: VDD domain is isolated from other power domain 1: VDD 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 – MCUDBGCR

This register specifies the MCU debug control.

Offset: 0x304

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved					DBSCTM3	DBSCTM2	
						RW	0	RW
Type/Reset	23	22	21	20	19	18	17	16
	DBSCTM1	DBSCTM0	Reserved	DBTRACE	DBUR1	DBUR0	DBBFTM1	DBBFTM0
Type/Reset	RW	0	RW	0	RW	0	RW	0
Type/Reset	15	14	13	12	11	10	9	8
	DBSCI	DBDSDLP2	DBI2C1	DBI2C0	DBSPI1	DBSPI0	Reserved	DBUSR
Type/Reset	RW	0	RW	0	RW	0	RW	0
Type/Reset	7	6	5	4	3	2	1	0
	Reserved	DBGPTM	Reserved	DBMCTM	DBWDT	DBPD	DBDSDLP1	DBSLP
Type/Reset		RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[25]	DBSCTM3	SCTM3 Debug Mode Enable 0: SCTM3 counter continues to count even if the core is halted 1: SCTM3 counter stops counting when the core is halted Set and reset by software.
[24]	DBSCTM2	SCTM2 Debug Mode Enable 0: SCTM2 counter continues to count even if the core is halted 1: SCTM2 counter stops counting when the core is halted Set and reset by software.
[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.
[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.

Bits	Field	Descriptions
[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.
[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]	DBSCI	SCI Debug Mode Enable 0: Same behavior as in normal mode 1: SCI timeout is frozen when the core is halted Set and reset by software. Note that the SCI function is only available for the HT32F52331/ HT32F52341 devices.
[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.
[8]	DBUSR	USART Debug Mode Enable 0: Same behavior as in normal mode 1: USART FIFO timeout is frozen when the core is halted Set and reset by software.
[6]	DBGPTM	GPTM Debug Mode Enable 0: GPTM counter continues to count even if the core is halted 1: GPTM counter 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 Counter of MCTM is stopped when 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.

<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>
[2]	DBPD	Debug Power-Down Mode 0: LDO = Off, FCLK = Off, and HCLK = Off in Power-Down mode 1: LDO = On, FCLK = On, and HCLK = On in Power-Down mode Set and reset by software.
[1]	DBDSLP1	Debug Deep-Sleep1 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

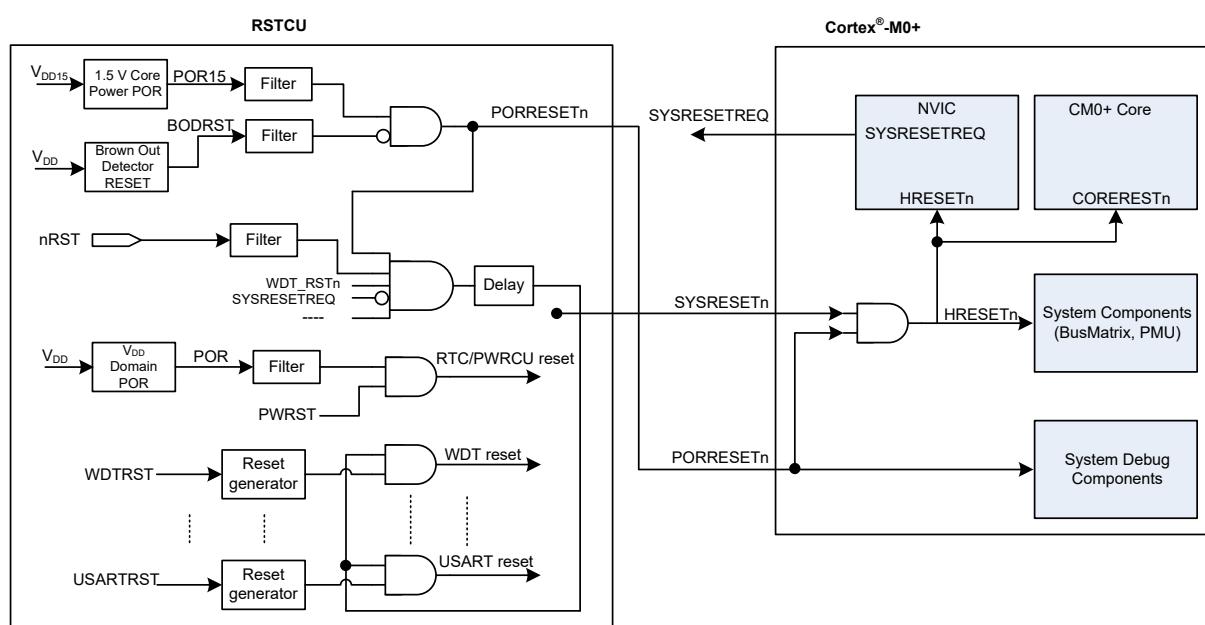


Figure 19. RSTCU Block Diagram

## Functional Descriptions

### Power On Reset

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

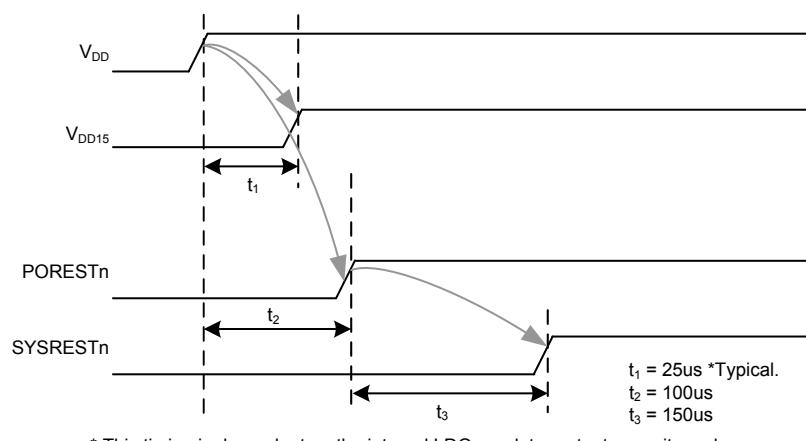


Figure 20. Power On Reset Sequence

### System Reset

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

### AHB and APB Unit Reset

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

## Register Map

The following table shows the RSTCU registers and reset values.

**Table 20. RSTCU Register Map**

Register	Offset	Description	Reset Value
<b>RSTCU Base Address = 0x4008_8000</b>			
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

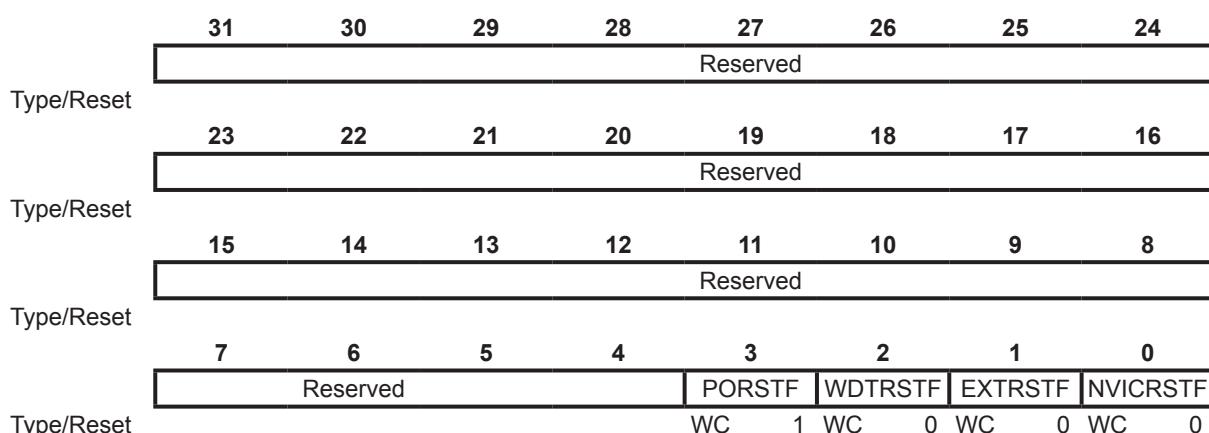
## Register Descriptions

### Global Reset Status Register – GRSR

This register specifies a variety of reset status conditions.

Offset: 0x100

Reset value: 0x0000\_0008



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.

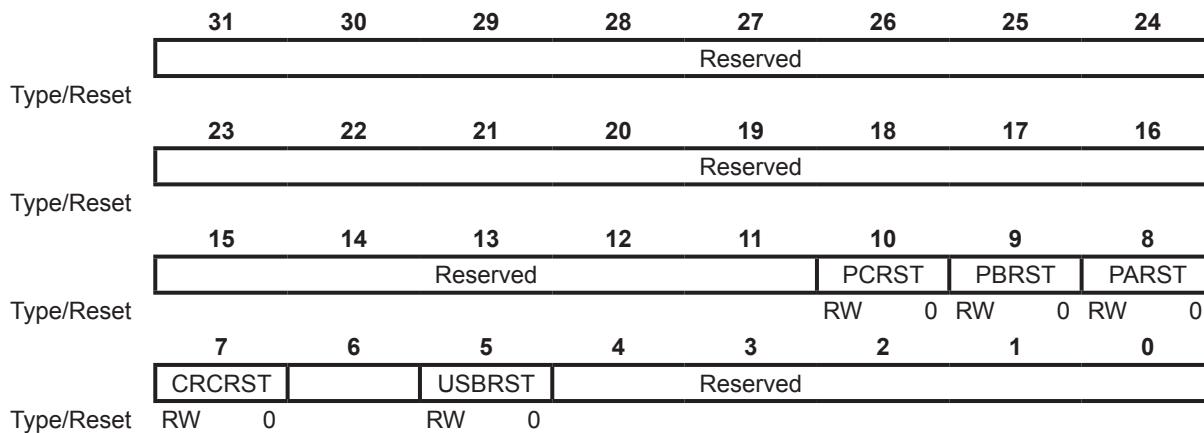
Bits	Field	Descriptions
[0]	NVICRSTF	<p>NVIC Reset Flag</p> <p>0: No NVIC asserting system reset occurred</p> <p>1: NVIC asserting system reset occurred</p> <p>This bit is set by hardware when a system reset occurs and reset by writing 1 into it or by hardware when a power on reset occurs.</p>

## AHB Peripheral Reset Register – AHBPRSTR

This register specifies several AHB peripherals software reset control bits.

Offset: 0x104

Reset value: 0x0000\_0000



Bits	Field	Descriptions
[10]	PCRST	<p>GPIO Port C Reset Control</p> <p>0: No reset</p> <p>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</p> <p>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</p> <p>1: Reset Port A</p> <p>This bit is set by software and cleared to 0 by hardware automatically.</p>
[7]	CRCRST	<p>CRC Reset Control</p> <p>0: No reset</p> <p>1: Reset CRC</p> <p>This bit is set by software and cleared to 0 by hardware automatically.</p>
[5]	USBRST	<p>USB Reset Control</p> <p>0: No reset</p> <p>1: Reset USB</p> <p>This bit is set by software and cleared to 0 by hardware automatically. Note that the USB function is only available for the HT32F52331/ HT32F52341 devices.</p>

## 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			SCIRST	
							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	Reserved	USRrst
	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

Bits	Field	Descriptions
[24]	SCIRST	Smart Card Interface Reset Control 0: No reset 1: Reset Smart Card Interface This bit is set by software and cleared to 0 by hardware automatically. Note that the SCI function is only available for the HT32F52331/ HT32F52341 devices.
[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.
[8]	USRrst	USART Reset Control 0: No reset 1: Reset USART 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.

<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>
[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	SCTM3RST	SCTM2RST	SCTM1RST	SCTM0RST		Reserved		ADCRST
	RW	0	RW	0	RW	0		RW 0
	23	22	21	20	19	18	17	16
Type/Reset					Reserved		BFTM1RST	BFTM0RST
							RW 0	RW 0
Type/Reset	15	14	13	12	11	10	9	8
					Reserved			GPTMRST
Type/Reset							RW 0	
	7	6	5	4	3	2	1	0
Type/Reset			Reserved	WDTRST		Reserved		MCTMRST
					RW 0		RW 0	

Bits	Field	Descriptions
[31]	SCTM3RST	SCTM3 Reset Control 0: No reset 1: Reset SCTM3 This bit is set by software and cleared to 0 by hardware automatically.
[30]	SCTM2RST	SCTM2 Reset Control 0: No reset 1: Reset SCTM2 This bit is set by software and cleared to 0 by hardware automatically.
[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.
[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.

<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>
[8]	GPTMRST	GPTM Reset Control 0: No reset 1: Reset GPTM This bit is set by software and cleared to 0 by hardware automatically.
[4]	WDTRST	Watchdog Timer Reset Control 0: No reset 1: Reset Watchdog Timer This bit is set by software and cleared to 0 by hardware automatically.
[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 40 General Purpose I/O port, GPIO, named PA0 ~ PA15, PB0 ~ PB15 and PC0 ~ PC7 for the HT32F52231/52241 device and up to 38 General Purpose I/O port, GPIO, named PA0 ~ PA15, PB0 ~ PB15 and PC0 ~ PC5 for the HT32F52331/52341 device to implement the logic input/output functions. Each of the GPIO ports has related control and configuration registers to satisfy the requirement of specific applications. The actual available General Purpose I/O port numbers are dependent on the device specification and package type. Refer to the device datasheet for detailed information.

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

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

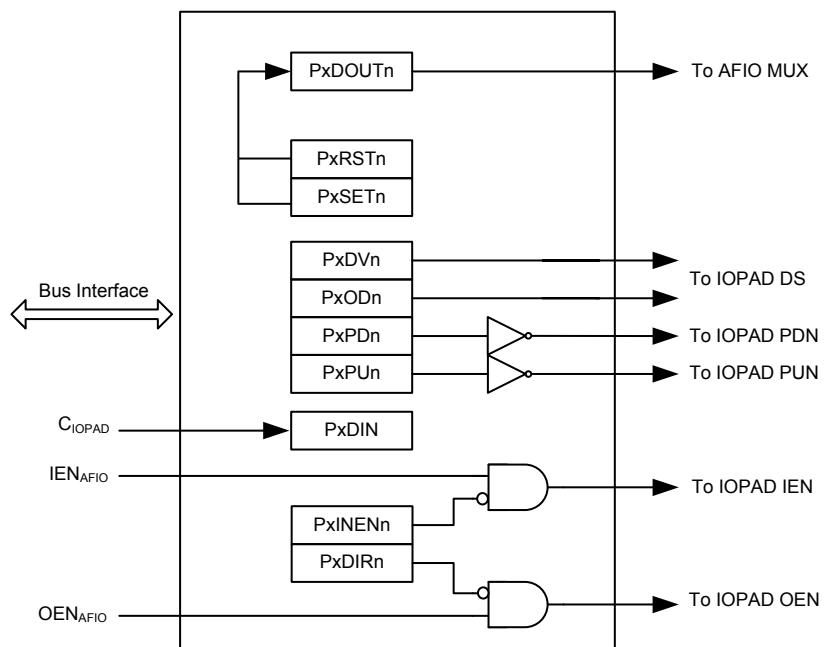


Figure 21. GPIO Block Diagram

## Features

- Input/output direction control
- 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

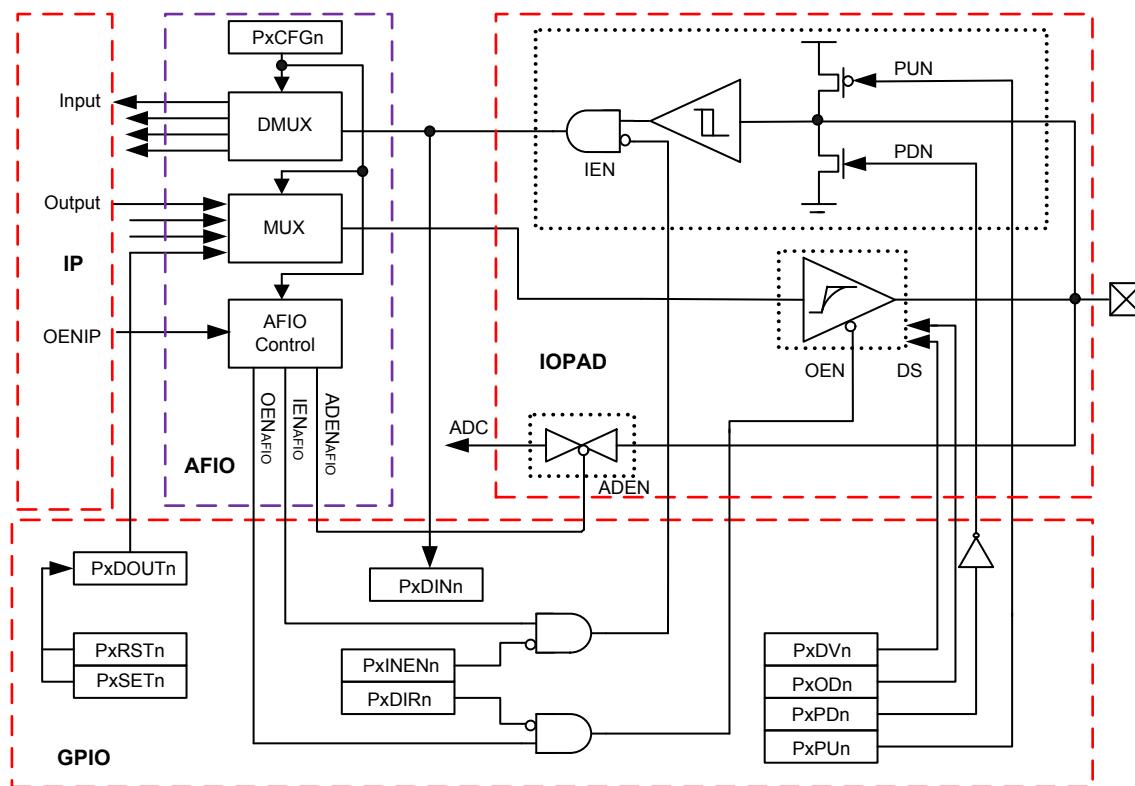
During or just after the reset period, the alternative functions are all inactive and the GPIO ports are configured into the input disable floating mode, i.e. input disabled without pull-up/pull-down resistors. Only the boot and Serial-Wired Debug pins which are pin-shared with the I/O pins are active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

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

### General Purpose I/O – GPIO

The GPIO pins can be configured as inputs or outputs via the data direction control registers PxDIRCR (where x = A ~ C). 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 PxDRVRR.



**Figure 22. AFIO/GPIO Control Signal**

PxDInn/PxDOOutn (x=A ~ C): Data Input/Data Output	PxRSTn/PxSETn (x=A ~ C): Reset/Set
PxDIRn (x=A ~ C): Direction	PxINENN (x=A ~ C): Input Enable
PxDVn (x=A ~ C): Output Drive	PxODn (x=A ~ C): Open Drain
PxPDn/PxPUn (x=A ~ C): Pull Down/Up	PxCFGn (x=A ~ C): AFIO Configuration

**Table 21. AFIO, GPIO and IO Pad Control Signal True Table**

Type	AFIO			GPIO		PAD		
	ADEN <sub>AFIO</sub>	OEN <sub>AFIO</sub>	IEN <sub>AFIO</sub>	PxDIRn	PxINENN	ADEN	OEN	IEN
GPIO Input <sup>(Note)</sup>	1	1	1	0	1	1	1	0
GPIO Output <sup>(Note)</sup>	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 ( $x = A \sim C$ ) 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, PxDRVVR control and AFIO mode configuration (GPxCFGHR or GPxCFGLR, where  $x = A \sim C$ ). 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 ~ C.

**Table 22. GPIO Register Map**

Register	Offset	Description	Reset Value
<b>GPIO A Base Address = 0x400B_0000</b>			
PADIRCR	0x000	Port A Data Direction Control Register	0x0000_0000
PAINER	0x004	Port A Input Function Enable Control Register	0x0000_0200
PAPUR	0x008	Port A Pull-Up Selection Register	0x0000_3200
PAPDR	0x00C	Port A Pull-Down Selection Register	0x0000_0000
PAODR	0x010	Port A Open Drain Selection Register	0x0000_0000
PADDRV	0x014	Port A Drive Current Selection Register	0x0000_0000
PALOCKR	0x018	Port A Lock Register	0x0000_0000
PADINR	0x01C	Port A Data Input Register	0x0000_3200
PADOUTR	0x020	Port A Data Output Register	0x0000_0000
PASRR	0x024	Port A Output Set and Reset Control Register	0x0000_0000
PARR	0x028	Port A Output Reset Control Register	0x0000_0000
<b>GPIO B Base Address = 0x400B_2000</b>			
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
PBDRV	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
<b>GPIO C Base Address = 0x400B_4000</b>			
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
PCDRV	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
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

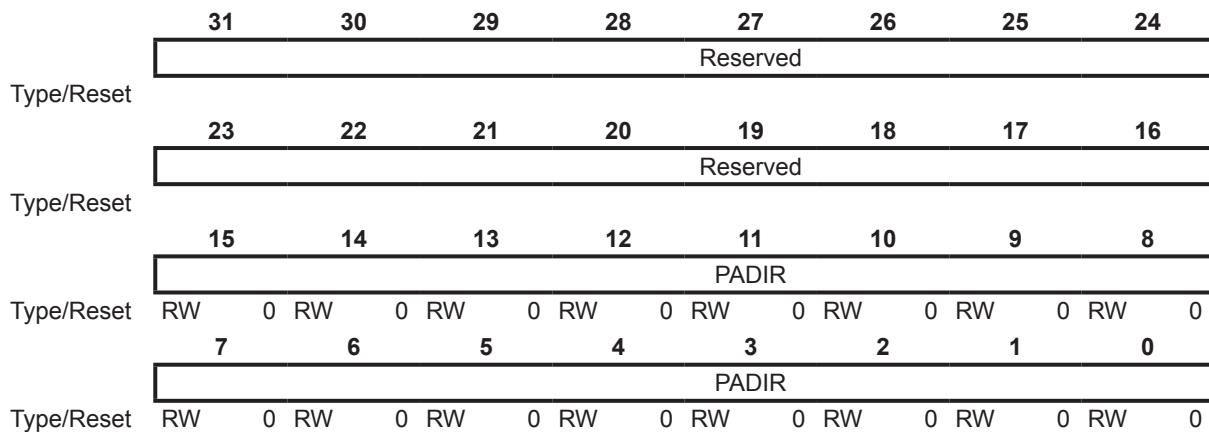
## Register Descriptions

### 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



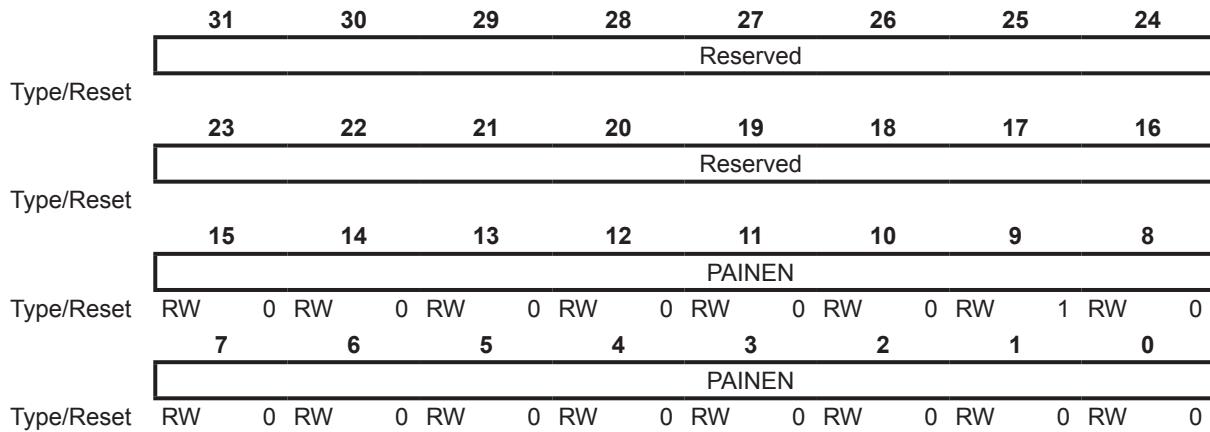
Bits	Field	Descriptions
[15:0]	PADIRn	GPIO Port A pin n Direction Control Bits (n = 0 ~ 15) 0: Pin n is input mode 1: Pin n is output mode

## 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\_0200



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\_3200

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

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

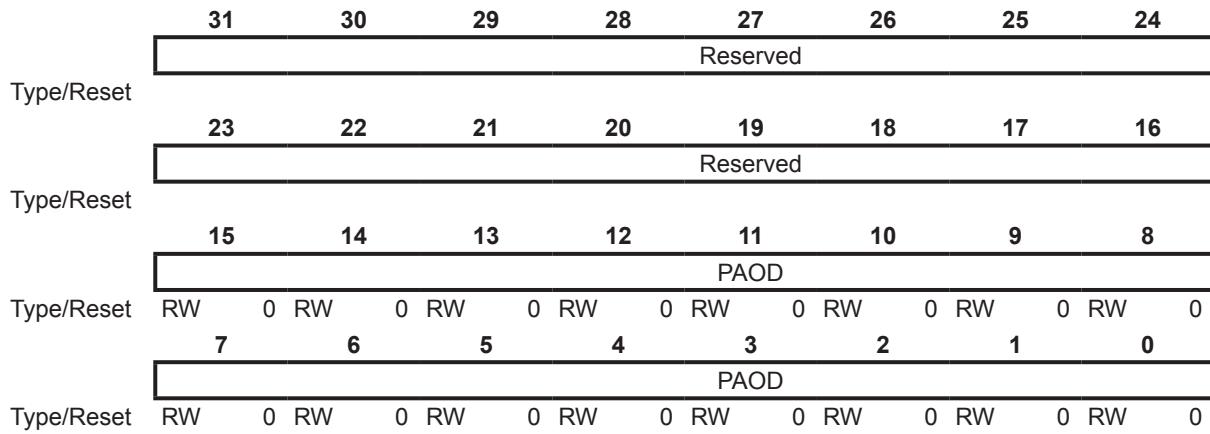
Bits	Field	Descriptions
[15:0]	PAPDn	<p>GPIO Port A pin n Pull-Down Selection Control Bits (n = 0 ~ 15)</p> <p>0: Pin n pull-down function is disabled            1: Pin n pull-down function is enabled</p> <p>Note: When the pull-up and pull-down functions are both enabled, the pull-up function will have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



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

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

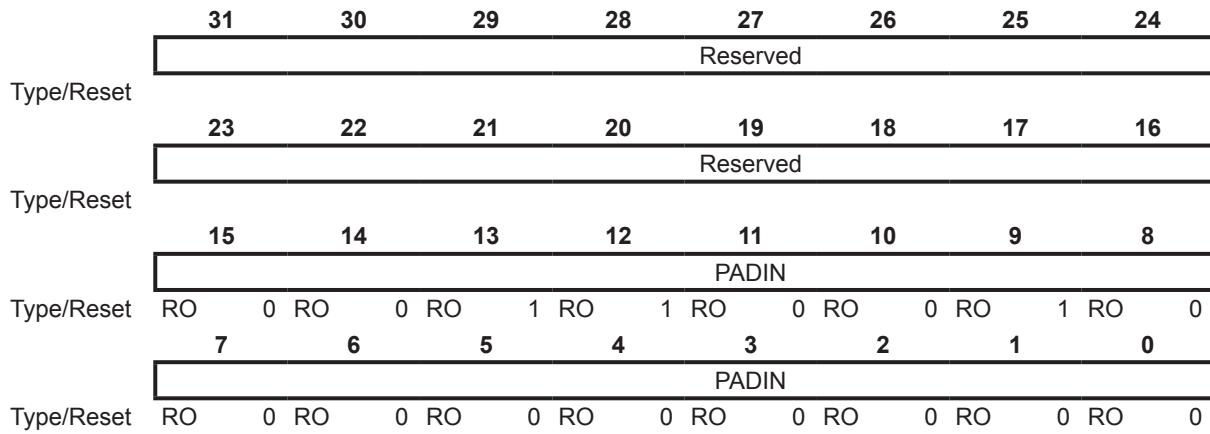
Bits	Field	Descriptions
[31:16]	PALKEY	<p>GPIO Port A Lock Key</p> <p>0x5FA0: Port A Lock function is enable Others: Port A Lock function is disable</p> <p>To lock the Port A function, a value 0x5FA0 should be written into the PALKEY field in this register. To execute a successful write operation on 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\_3200



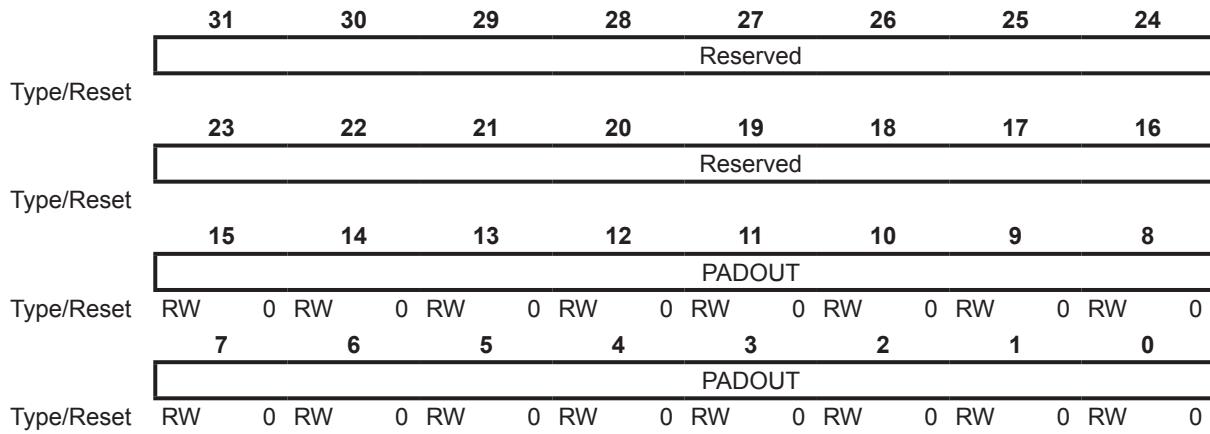
Bits	Field	Descriptions
[15:0]	PADINn	GPIO Port A pin n Data Input Bits (n = 0 ~ 15) 0: The input data of the corresponding pin is 0 1: The input data of the corresponding pin is 1

## Port A Output Data Register – PADOUTR

This register specifies the GPIO Port A output data.

Offset: 0x020

Reset value: 0x0000\_0000



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
23 22 21 20 19 18 17 16								
PARST								
Type/Reset	WO	0	WO	0	WO	0	WO	0
15 14 13 12 11 10 9 8								
PASET								
Type/Reset	WO	0	WO	0	WO	0	WO	0
7 6 5 4 3 2 1 0								
PASET								
Type/Reset	WO	0	WO	0	WO	0	WO	0

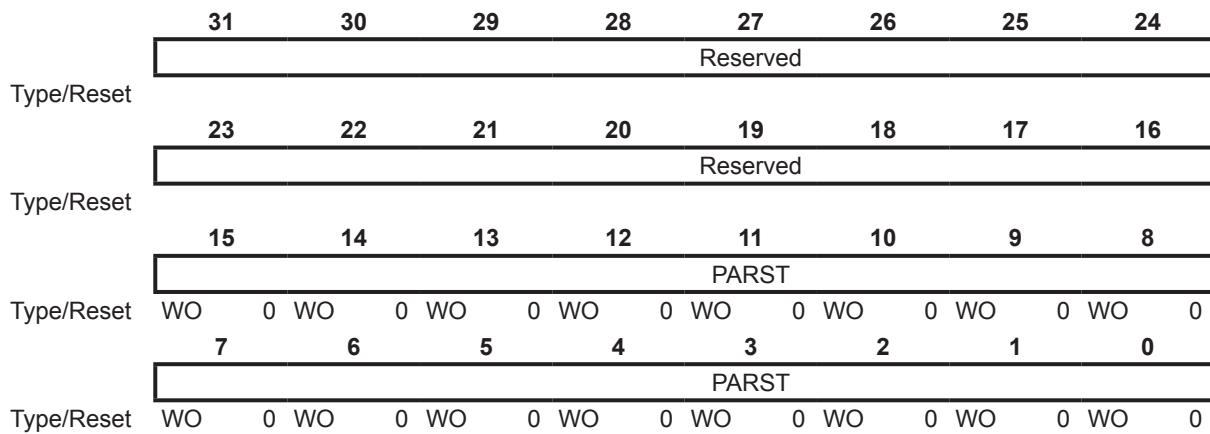
Bits	Field	Descriptions
[31:16]	PARSTn	GPIO Port A pin n Output Reset Control Bits (n = 0 ~ 15) 0: No effect on the PADOUTn bit 1: Reset the PADOUTn bit
[15:0]	PASETn	GPIO Port A pin n Output Set Control Bits (n = 0 ~ 15) 0: No effect on the PADOUTn bit 1: Set the PADOUTn bit Note that the function enabled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



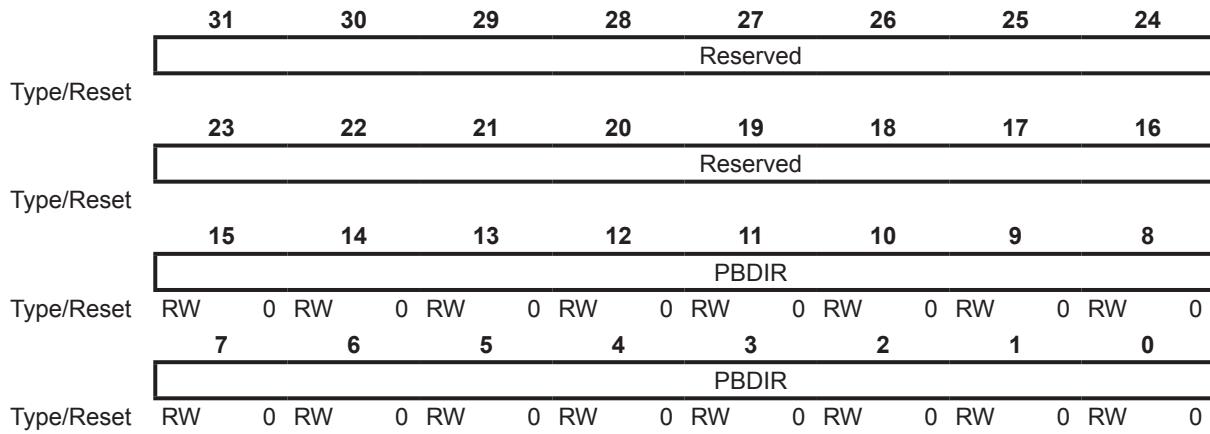
Bits	Field	Descriptions
[15:0]	PARSTn	GPIO Port A pin n Output Reset Bits (n = 0 ~ 15) 0: No effect on the PADOUTn bit 1: Reset the PADOUTn bit

## 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



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

## Port B Input Function Enable Control Register – PBINER

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

Offset: 0x004

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[15:0]	PBINENn	GPIO Port B 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 B Pull-Up Selection Register – PBPUR

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

Offset: 0x008

Reset value: 0x0000\_0000

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

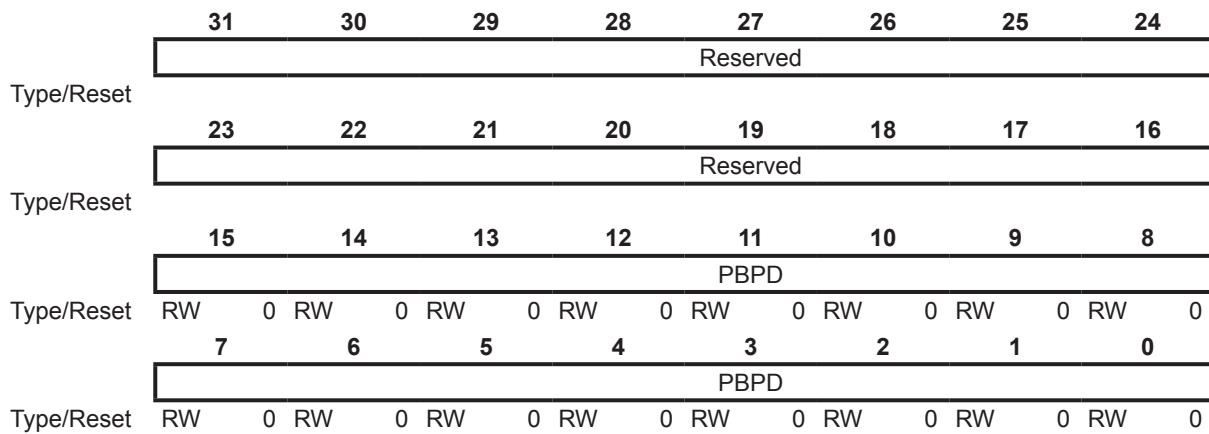
Bits	Field	Descriptions
[15:0]	PBPU <sub>n</sub>	<p>GPIO Port B pin n Pull-Up Selection Control Bits (n = 0 ~ 15)</p> <p>0: Pin n pull-up function is disabled 1: Pin n pull-up function is enabled</p> <p>Note: When the pull-up and pull-down functions are both enabled, the pull-up function will have the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

## Port B Pull-Down Selection Register – PBPDR

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

Offset: 0x00C

Reset value: 0x0000\_0000



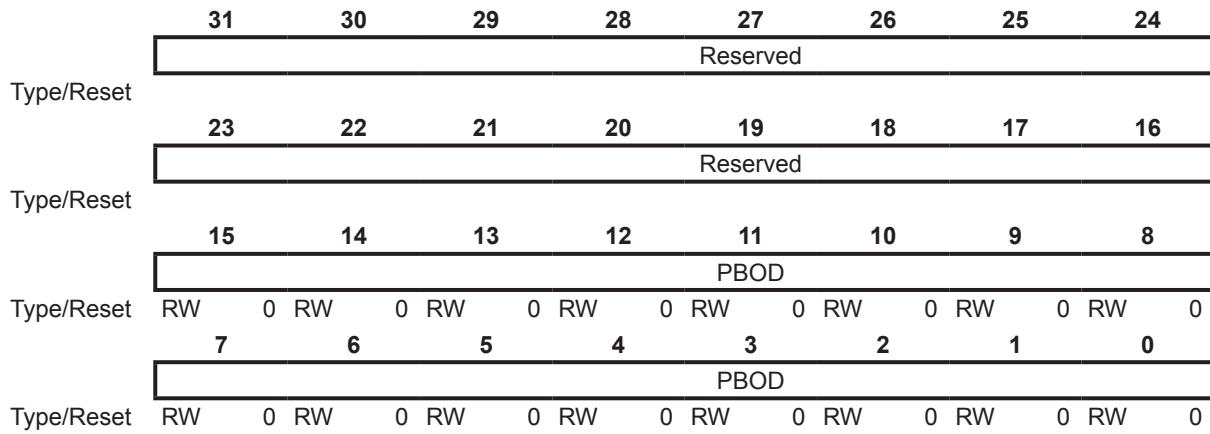
Bits	Field	Descriptions
[15:0]	PBPDn	<p>GPIO Port B pin n Pull-Down Selection Control Bits (n = 0 ~ 15)</p> <p>0: Pin n pull-down function is disabled 1: Pin n pull-down function is enabled</p> <p>Note: When the pull-up and pull-down functions are both enabled, the pull-up function will have the higher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

## Port B Open Drain Selection Register – PBODR

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

Offset: 0x010

Reset value: 0x0000\_0000



Bits	Field	Descriptions
[15:0]	PBODn	GPIO Port B 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 B Output Current Drive Selection Register – PBDRV

This register specifies the GPIO Port B output driving current.

Offset: 0x014

Reset value: 0x0000\_0000

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

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

## Port B Lock Register – 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
23 22 21 20 19 18 17 16								
PBLKEY								
Type/Reset	RW	0	RW	0	RW	0	RW	0
15 14 13 12 11 10 9 8								
PBLOCK								
Type/Reset	RW	0	RW	0	RW	0	RW	0
7 6 5 4 3 2 1 0								
PBLOCK								
Type/Reset	RW	0	RW	0	RW	0	RW	0

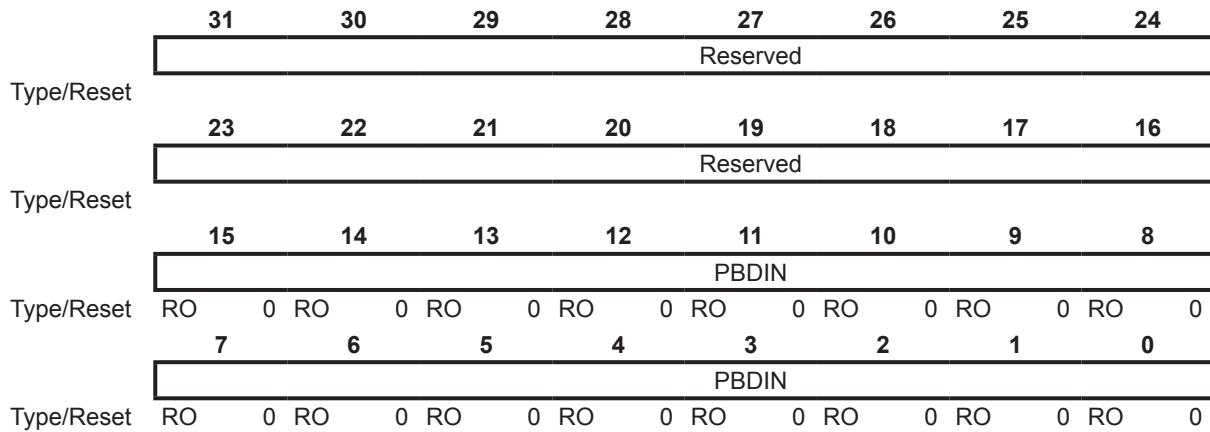
Bits	Field	Descriptions
[31:16]	PBLKEY	<p>GPIO Port Block Key</p> <p>0x5FA0: Port Block function is enable Others: Port B Lock function is disable</p> <p>To lock the Port B function, a value 0x5FA0 should be written into the PBLKEY field in this register. To execute a successful write operation on this 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>
[15:0]	PBLOCKn	<p>GPIO Port B pin n Lock Control Bits (n = 0 ~ 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, PBPUn, PBPDn and PBODn 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



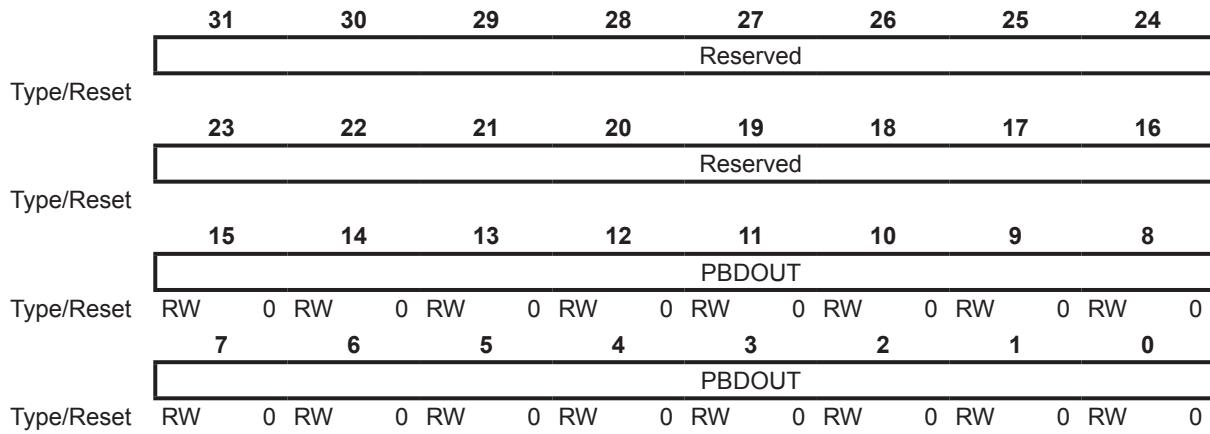
Bits	Field	Descriptions
[15:0]	PBDINn	GPIO Port B pin n Data Input Bits (n = 0 ~ 15) 0: The input data of the corresponding pin is 0 1: The input data of the corresponding pin is 1

## Port B Output Data Register – PBDOUTR

This register specifies the GPIO Port B output data.

Offset: 0x020

Reset value: 0x0000\_0000



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

## Port B Output Set/Reset Control Register – PBSRR

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

Offset: 0x024

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
PBRST								
Type/Reset	WO	0	WO	0	WO	0	WO	0
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								
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

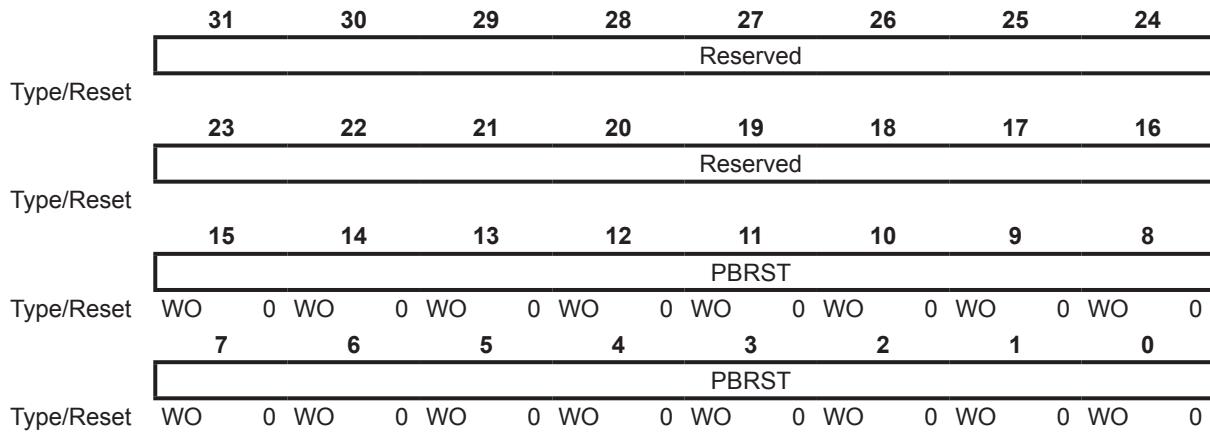
Bits	Field	Descriptions
[31:16]	PBRSTn	GPIO Port B pin n Output Reset Control Bits (n = 0 ~ 15) 0: No effect on the PBDOUTn bit 1: Reset the PBDOUTn bit
[15:0]	PBSETn	GPIO Port B pin n Output Set Control Bits (n = 0 ~ 15) 0: No effect on the PBDOUTn bit 1: Set the PBDOUTn bit Note that the function enabled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

## Port B Output Reset Register – PBRR

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

Offset: 0x028

Reset value: 0x0000\_0000



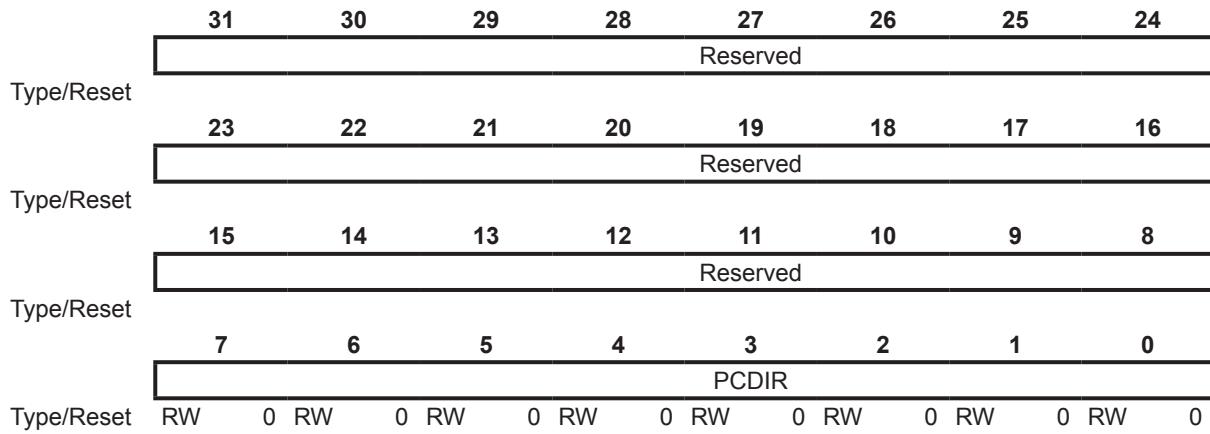
Bits	Field	Descriptions
[15:0]	PBRSTn	GPIO Port B pin n Output Reset Bits (n = 0 ~ 15) 0: No effect on the PBDOUTn bit 1: Reset the PBDOUTn bit

## Port C Data Direction Control Register – PCDIRCR

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

Offset: 0x000

Reset value: 0x0000\_0000



Bits	Field	Descriptions
[n:0]	PCDIRn	<p>GPIO Port C pin n Direction Control Bits (n = 0 ~ 7 or 5)</p> <p>0: Pin n is input mode 1: Pin n is output mode</p> <p>For the HT32F52231/HT32F52241 devices the variable “n” is equal to 7 while the variable “n” is equal to 5 for the HT32F52331/HT32F52341 devices.</p>

## Port C Input Function Enable Control Register – PCINER

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

Offset: 0x004

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[n:0]	PCINENn	<p>GPIO Port C pin n Input Enable Control Bits (n = 0 ~ 7 or 5)</p> <p>0: Pin n input function is disabled. 1: Pin n input function is enabled.</p> <p>When the pin n input function is disabled, the input Schmitt trigger will be turned off and the Schmitt trigger output will remain at a zero state.</p> <p>For the HT32F52231/HT32F52241 devices the variable “n” is equal to 7 while the variable “n” is equal to 5 for the HT32F52331/HT32F52341 devices.</p>

## Port C Pull-Up Selection Register – PCPUR

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

Offset: 0x008

Reset value: 0x0000\_0000

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

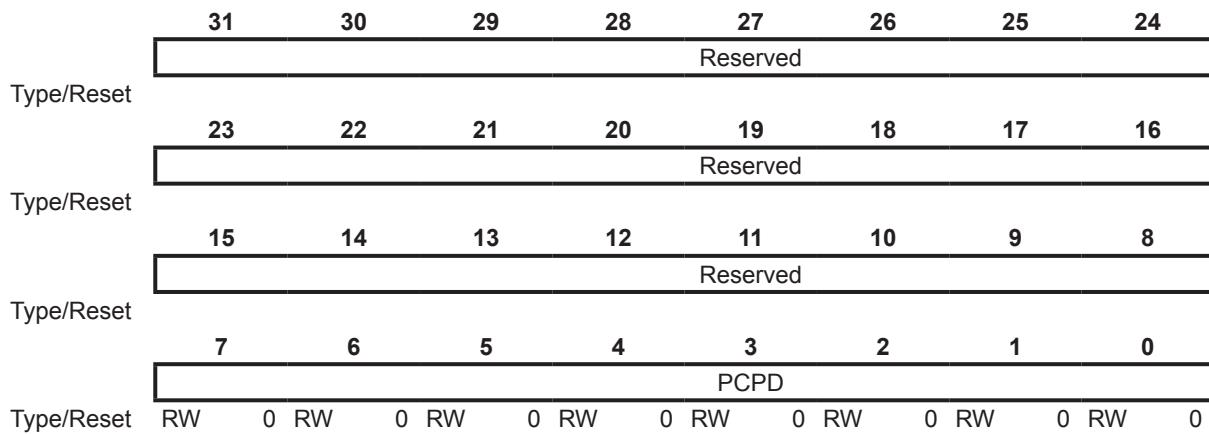
Bits	Field	Descriptions
[n:0]	PCPUn	<p>GPIO Port C pin n Pull-Up Selection Control Bits (n = 0 ~ 7 or 5)</p> <p>0: Pin n pull-up function is disabled                  1: Pin n pull-up function is enabled</p> <p>For the HT32F52231/HT32F52241 devices the variable “n” is equal to 7 while the variable “n” is equal to 5 for the HT32F52331/HT32F52341 devices.</p> <p>Note: When the pull-up and pull-down functions are both enabled, the pull-up function will have the higher priority and therefore the pull-down function will be blocked and disabled.</p>

## Port C Pull-Down Selection Register – PCPDR

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

Offset: 0x00C

Reset value: 0x0000\_0000



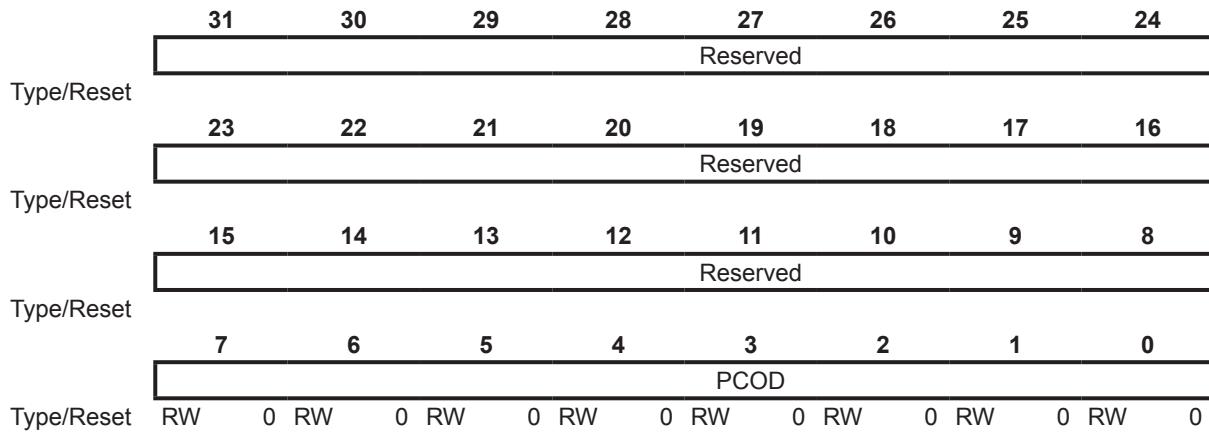
Bits	Field	Descriptions
[n:0]	PCPDn	<p>GPIO Port C pin n Pull-Down Selection Control Bits (n = 0 ~ 7 or 5)</p> <p>0: Pin n pull-down function is disabled 1: Pin n pull-down function is enabled</p> <p>For the HT32F52231/HT32F52241 devices the variable “n” is equal to 7 while the variable “n” is equal to 5 for the HT32F52331/HT32F52341 devices.</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
[n:0]	PCODn	<p>GPIO Port C pin n Open Drain Selection Control Bits (n = 0 ~ 7 or 5)</p> <p>0: Pin n Open Drain output is disabled. (The output type is CMOS output)</p> <p>1: Pin n Open Drain output is enabled. (The output type is open-drain output)</p> <p>For the HT32F52231/HT32F52241 devices the variable “n” is equal to 7 while the variable “n” is equal to 5 for the HT32F52331/HT32F52341 devices.</p>

## 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
Reserved								
Type/Reset	23	22	21	20	19	18	17	16
Reserved								
Type/Reset	15	14	13	12	11	10	9	8
PCDV7   PCDV6   PCDV5   PCDV4								
Type/Reset	RW	0	RW	0	RW	0	RW	0
PCDV3   PCDV2   PCDV1   PCDV0								
Type/Reset	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[15:0] or PCDVn[1:0]		GPIO Port C pin n Output Current Drive Selection Control Bits (n = 0 ~ 7 or 5)
[11:0]		00: 4 mA source/sink current 01: 8 mA source/sink current 10: 12 mA source/sink current 11: 16 mA source/sink current For the HT32F52231/HT32F52241 devices the variable “n” is equal to 7 while the variable “n” is equal to 5 for the HT32F52331/HT32F52341 devices.

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

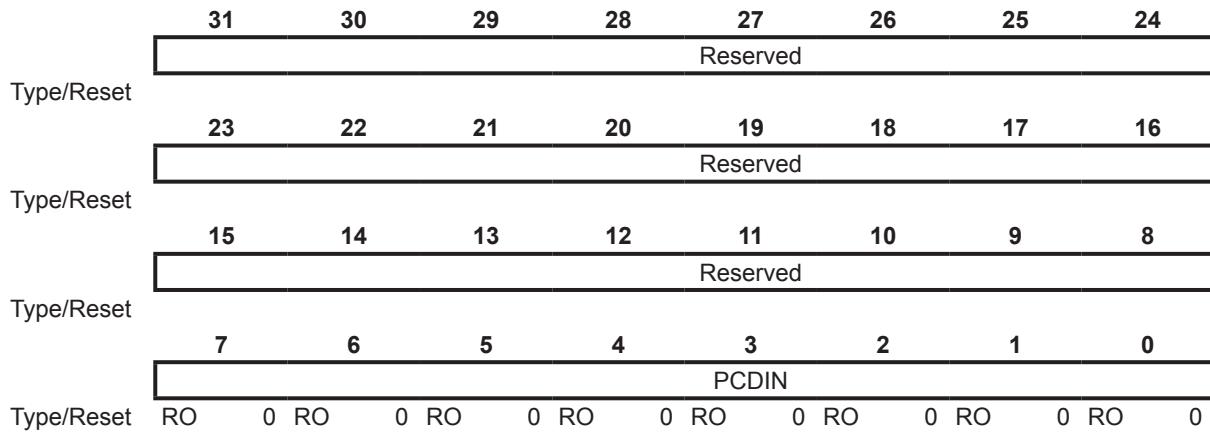
Bits	Field	Descriptions
[31:16]	PCLKEY	<p>GPIO Port C lock Key</p> <p>0x5FA0: Port C Lock function is enable Others: Port C Lock function is disable</p> <p>To lock the Port C function, a value 0x5FA0 should be written into the PCLKEY field in this register. To execute a successful write operation on 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>
[n:0]	PCLOCKKn	<p>GPIO Port C pin n Lock Control Bits (n = 0 ~ 7 or 5)</p> <p>0: Port C pin n is not locked 1: Port C pin n is locked</p> <p>The PCLOCKKn 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 PCLOCKKn (lock control bit) should be written together and can not be changed until a system reset or GPIO Port C reset occurs.</p> <p>For the HT32F52231/HT32F52241 devices the variable “n” is equal to 7 while the variable “n” is equal to 5 for the HT32F52331/HT32F52341 devices.</p>

## Port C Data Input Register – PCDINR

This register specifies the GPIO Port C input data.

Offset: 0x01C

Reset value: 0x0000\_0000



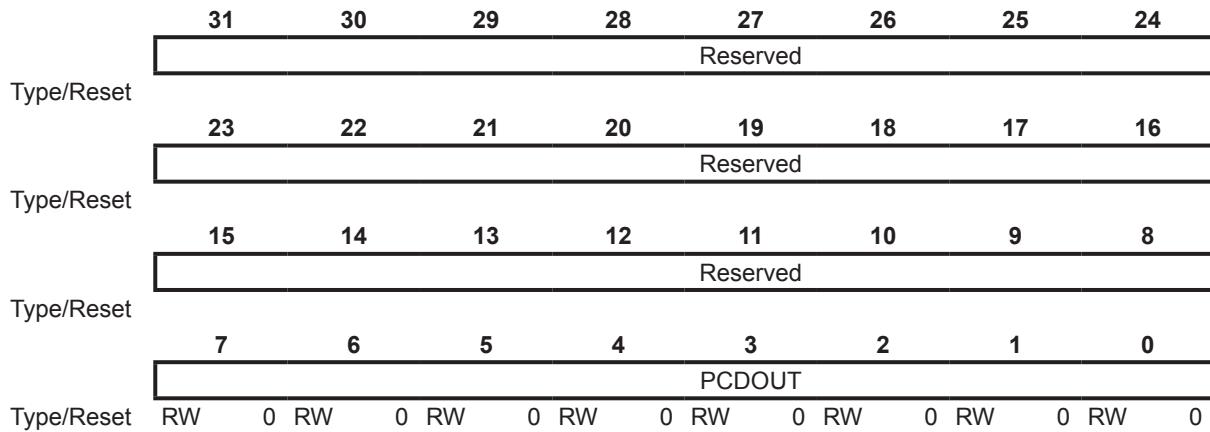
Bits	Field	Descriptions
[n:0]	PCDINn	<p>GPIO Port C pin n Data Input Bits (n = 0 ~ 7 or 5)</p> <p>0: The input data of the corresponding pin is 0 1: The input data of the corresponding pin is 1</p> <p>For the HT32F52231/HT32F52241 devices the variable “n” is equal to 7 while the variable “n” is equal to 5 for the HT32F52331/HT32F52341 devices.</p>

## Port C Output Data Register – PCDOUTR

This register specifies the GPIO Port C output data.

Offset: 0x020

Reset value: 0x0000\_0000



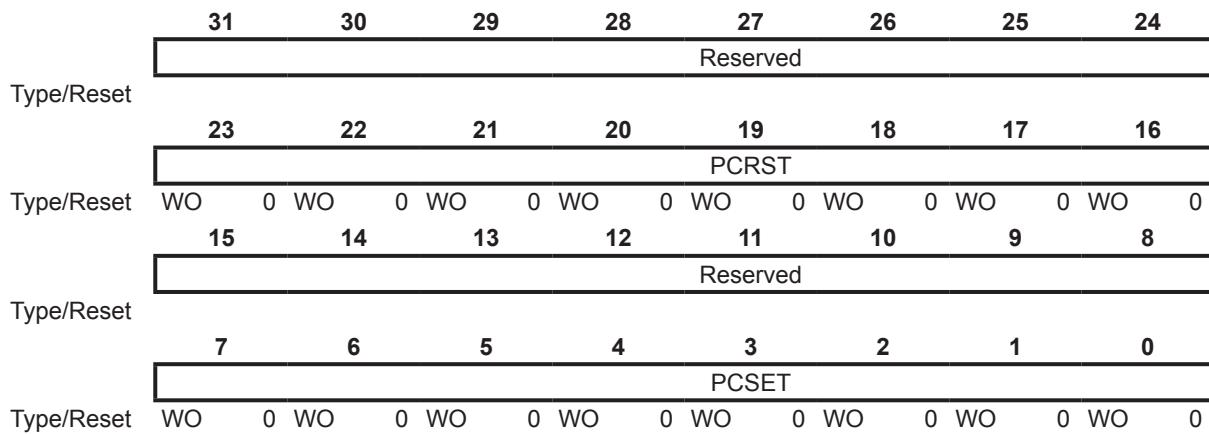
Bits	Field	Descriptions
[n:0]	PCDOUTn	<p>GPIO Port C pin n Data Output Bits (n = 0 ~ 7 or 5)</p> <p>0: Data to be output on pin n is 0 1: Data to be output on pin n is 1</p> <p>For the HT32F52231/HT32F52241 devices the variable “n” is equal to 7 while the variable “n” is equal to 5 for the HT32F52331/HT32F52341 devices.</p>

## 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



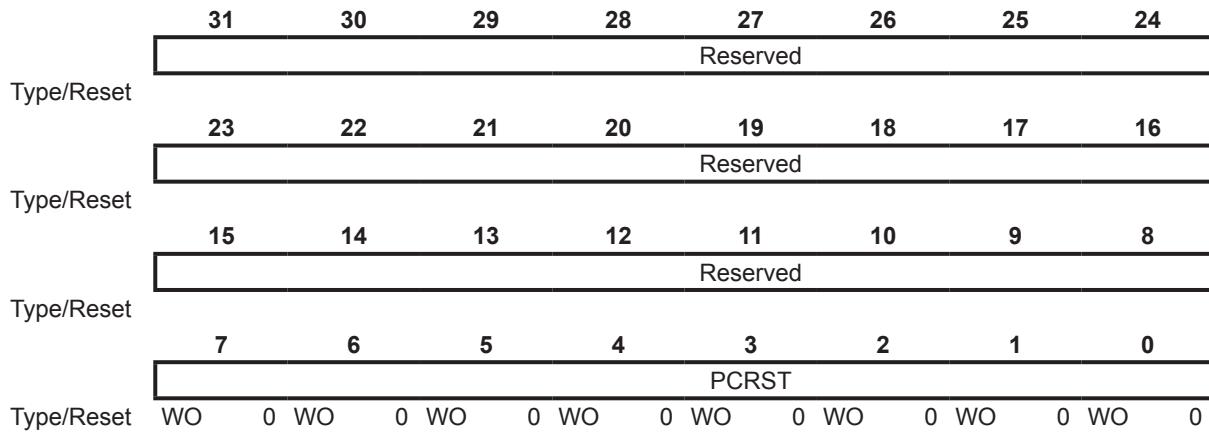
Bits	Field	Descriptions
[23:16] or PCRSTn		GPIO Port C pin n Output Reset Control Bits (n = 0 ~ 7 or 5) [21:16] 0: No effect on the PCDOUTn bit 1: Reset the PCDOUTn bit
[n:0]	PCSETn	GPIO Port C pin n Output Set Control Bits (n = 0 ~ 7 or 5) 0: No effect on the PCDOUTn bit 1: Set the PCDOUTn bit Note 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. For the HT32F52231/HT32F52241 devices the variable “n” is equal to 7 while the variable “n” is equal to 5 for the HT32F52331/HT32F52341 devices.

## 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



Bits	Field	Descriptions
[n:0]	PCRST <sub>n</sub>	GPIO Port C pin n Output Reset Bits (n = 0 ~ 7 or 5) 0: No effect on the PCDOUT <sub>n</sub> bit 1: Reset the PCDOUT <sub>n</sub> bit For the HT32F52231/HT32F52241 devices the variable “n” is equal to 7 while the variable “n” is equal to 5 for the HT32F52331/HT32F52341 devices.

## 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 GPxCFGRL or GPxCFGHR register where x is the different port name. According to the usage of the IP resource and application requirements, suitable pin-out locations can be selected by using the peripheral IO remapping mechanism. Additionally, various GPIO pins can be selected to be the EXTI interrupt line by setting the EXTInPIN [3:0] field in the ESSRn register to trigger an interrupt or event. Please refer to the EXTI section for more details.

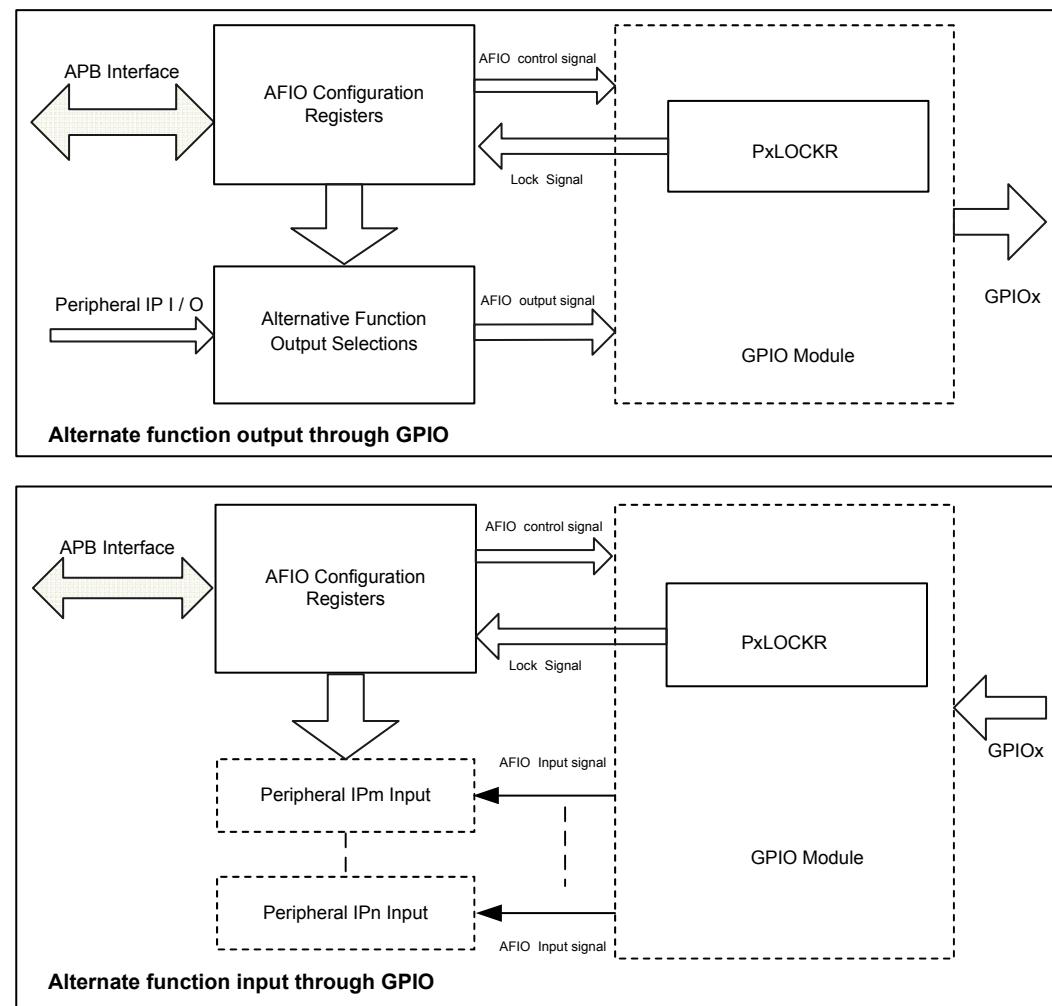


Figure 23. AFIO Block Diagram

## Features

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

## Functional Descriptions

### External Interrupt Pin Selection

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

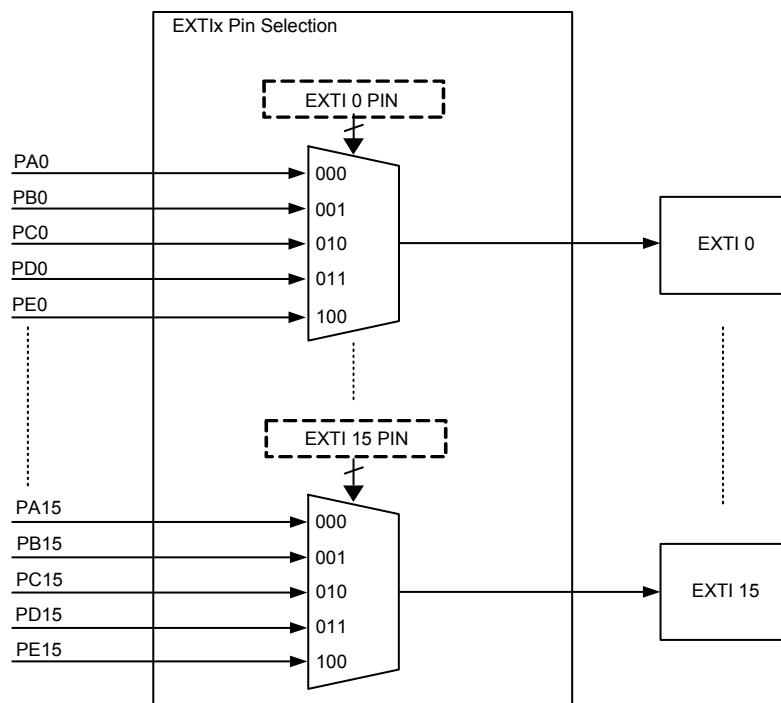


Figure 24. EXTI Channel Input Selection

## Alternate Function

Up to sixteen alternative functions can be chosen for each I/O pad by setting the PxCFGn [3:0] field in the GPxCFGGLR or GPxCFGHHR (n = 0~15, x = A~ C) 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 23. 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 <sup>2</sup> C	SCI	EBI	I <sup>2</sup> 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 24. 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
GPACFGGLR	0x020	GPIO Port A AFIO Configuration Register 0	0x0000_0000
GPACFGHHR	0x024	GPIO Port A AFIO Configuration Register 1	0x0000_0000
GPBCFGGLR	0x028	GPIO Port B AFIO Configuration Register 0	0x0000_0000
GPBCFGHHR	0x02C	GPIO Port B AFIO Configuration Register 1	0x0000_0000
GPCCFGGLR	0x030	GPIO Port C AFIO Configuration Register 0	0x0000_0000
GPCCFGHHR	0x034	GPIO Port C 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	
	23	22	21	20	19	18	17	16	
	EXTI5PIN					EXTI4PIN			
Type/Reset	RW	0	RW	0	RW	0	RW	0	
	15	14	13	12	11	10	9	8	
	EXTI3PIN					EXTI2PIN			
Type/Reset	RW	0	RW	0	RW	0	RW	0	
	7	6	5	4	3	2	1	0	
	EXTI1PIN					EXTI0PIN			
Type/Reset	RW	0	RW	0	RW	0	RW	0	

Bits	Field	Descriptions
[31:0]	EXTInPIN[3:0]	EXTIn Pin Selection (n = 0 ~ 7)  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 Others: Reserved  Note: Since not all GPIO pins are available in all products and package types, refer to the pin assignment section for detailed pin information. The EXTInPIN [3:0] field setting is invalid when the corresponding pin is not available.

## EXTI Source Selection Register 1 – ESSR1

This register specifies the IO selection of EXTI8~EXTI15.

Offset: 0x004

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset	RW	0	RW	0	RW	0	RW	0
	23	22	21	20	19	18	17	16
	EXTI15PIN					EXTI14PIN		
Type/Reset	RW	0	RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8
	EXTI13PIN					EXTI12PIN		
Type/Reset	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
	EXTI11PIN					EXTI10PIN		
Type/Reset	RW	0	RW	0	RW	0	RW	0
	EXTI9PIN					EXTI8PIN		
Type/Reset	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[31:0]	EXTInPIN[3:0]	<p>EXTIn Pin Selection (n = 8 ~ 15)</p> <p>0000: PA Bit n is selected as EXTIn source signal            0001: PB Bit n is selected as EXTIn source signal            0010: PC 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 EXTInPIN [3:0] field setting is invalid when the corresponding pin is not available.</p>

## GPIO x Configuration Low Register – GPxCFGRLR, x = A, B, C

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

Offset: 0x020, 0x028, 0x030

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[31:0]	PxCFGn[3:0]	<p>Port x Pin n Alternate function selection (n = 0~7)</p> <p>0000: Port x pin n is selected as AF0.</p> <p>0001: Port x pin n is selected as AF1.</p> <p>.</p> <p>.</p> <p>1110: Port x pin n is selected as AF14</p> <p>1111: Port x pin n is selected as AF15</p> <p>If the pin is selected as 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.</p>

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

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

Offset: 0x024, 0x02C, 0x034

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[31:0]	PxCFGn[3:0]	<p>Port x Pin n Alternate function selection (n = 8~15)</p> <p>0000: Port x pin n is selected as AF0.</p> <p>0001: Port x pin n is selected as AF1.</p> <p>.</p> <p>.</p> <p>1110: Port x pin n is selected as AF14</p> <p>1111: Port x pin n is selected as AF15</p> <p>If the pin is selected as 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.</p>

# 10 Nested Vectored Interrupt Controller (NVIC)

## Introduction

In order to reduce the latency and increase the interrupt processing efficiency, a tightly coupled integrated section, which is named as Nested Vectored Interrupt Controller (NVIC) is provided by the Cortex®-M0+. The NVIC controls the system exceptions and the peripheral 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 Timer 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 25. 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 <sup>(1)</sup>	0x02C	SVC instruction System service call
-	12-13	Reserved	—	—	—
-2	14	PendSV	Configurable <sup>(1)</sup>	0x038	System Service Pendable request
-1	15	SysTick	Configurable <sup>(1)</sup>	0x03C	SysTick timer decremented to zero
0	16	LVD	Configurable <sup>(2)</sup>	0x040	Low voltage detection interrupt
1	17	RTC	Configurable <sup>(2)</sup>	0x044	RTC global interrupt
2	18	FMC	Configurable <sup>(2)</sup>	0x048	FMC global interrupt
3	19	WKUP	Configurable <sup>(2)</sup>	0x04C	EXTI event wakeup or external WAKEUP pin interrupt
4	20	EXTI0 ~ 1	Configurable <sup>(2)</sup>	0x050	EXTI Line 0 & 1 interrupt
5	21	EXTI2 ~ 3	Configurable <sup>(2)</sup>	0x054	EXTI Line 2 & 3 interrupt
6	22	EXTI4 ~ 15	Configurable <sup>(2)</sup>	0x058	EXTI Line 4 ~ 15 interrupt
7	23	Reserved	—	0x05C	—
8	24	ADC	Configurable <sup>(2)</sup>	0x060	ADC global interrupt
9	25	Reserved	—	0x064	—
10	26	MCTM	Configurable <sup>(2)</sup>	0x068	MCTM global interrupt
11	27	Reserved	—	0x06C	—
12	28	GPTM	Configurable <sup>(2)</sup>	0x070	GPTM global interrupt

Interrupt Number	Exception Number	Exception type	Priority	Vector Address	Description
13	29	SCTM0	Configurable <sup>(2)</sup>	0x074	SCTM0 global interrupt
14	30	SCTM1	Configurable <sup>(2)</sup>	0x078	SCTM1 global interrupt
15	31	SCTM2	Configurable <sup>(2)</sup>	0x07C	SCTM2 global interrupt
16	32	SCTM3	Configurable <sup>(2)</sup>	0x080	SCTM3 global interrupt
17	33	BFTM0	Configurable <sup>(2)</sup>	0x084	BFTM0 global interrupt
18	34	BFTM1	Configurable <sup>(2)</sup>	0x088	BFTM1 global interrupt
19	35	I2C0	Configurable <sup>(2)</sup>	0x08C	I2C0 global interrupt
20	36	I2C1	Configurable <sup>(2)</sup>	0x090	I2C1 global interrupt
21	37	SPI0	Configurable <sup>(2)</sup>	0x094	SPI0 global interrupt
22	38	SPI1	Configurable <sup>(2)</sup>	0x098	SPI1 global interrupt
23	39	USART	Configurable <sup>(2)</sup>	0x09C	USART global interrupt
24	40	Reserved	—	0x0A0	—
25	41	UART0	Configurable <sup>(2)</sup>	0x0A4	UART0 global interrupt
26	42	UART1	Configurable <sup>(2)</sup>	0x0A8	Configurable <sup>(2)</sup>
27	43	SCI <sup>(3)</sup>	Configurable <sup>(2)</sup>	0x0AC	SCI global interrupt
28	44	Reserved	—	0x0B0	—
29	45	USB <sup>(3)</sup>	Configurable <sup>(2)</sup>	0x0B4	USB global interrupt
30	46	Reserved	—	0x0B8	—
31	47	Reserved	—	0x0BC	—

- 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.
  3. SCI and USB IP are only available for HT32F52331/52341 devices.

## 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 or 5000 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 or 5 MHz (48 or 40 MHz divide by 8).

## Register Map

The following table shows the NVIC registers and reset values.

**Table 26. NVIC Register Map**

Register	Offset	Description	Reset Value
<b>NVIC Base Address = 0xE000_E000</b>			
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_2328
NVIC_ISET	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 EXTCFGRn (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.

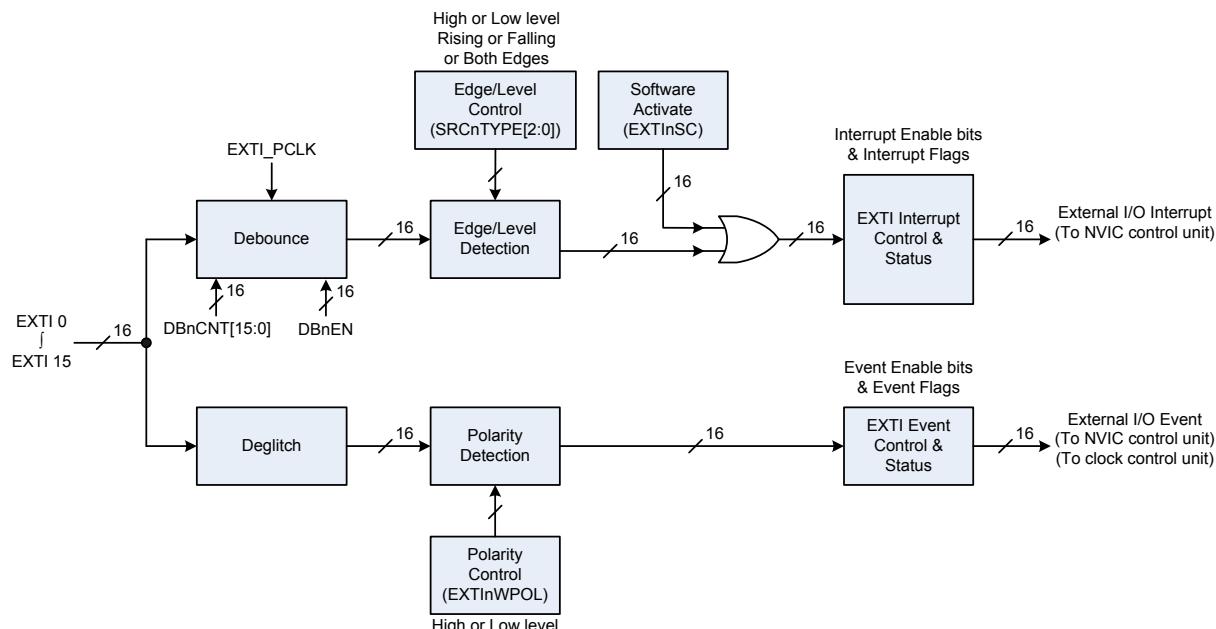


Figure 25. EXTI Block Diagram

## Features

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

## 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, USB and RTC wakeup functions. Note that the USB wakeup function is only available for the HT32F52331/HT32F52341 devices. 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

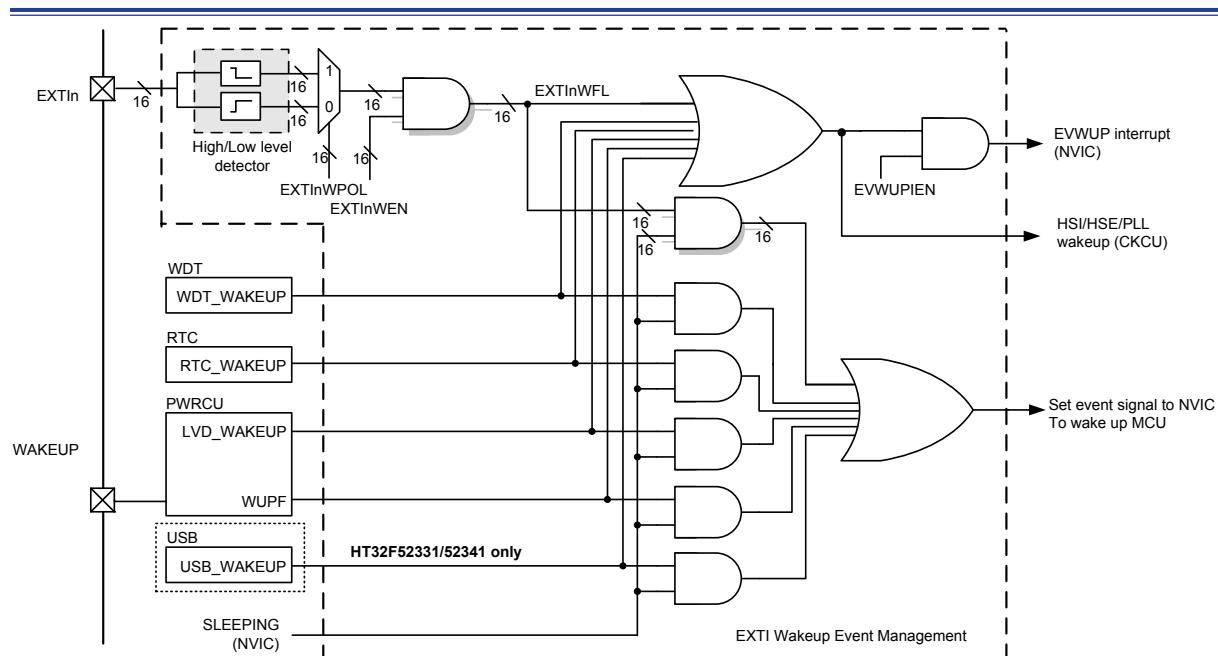


Figure 26. EXTI Wake-up Event Management

## 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 ESSR $n$  ( $n = 0 \sim 1$ ) register to trigger an interrupt or event. Refer to the AFIO section for more details.

## Interrupt and Debounce

The application software can set the DBnEN bit in the EXTIn Interrupt Configuration Register EXTICFGR $n$  ( $n = 0 \sim 15$ ) to enable the corresponding pin de-bounce function and configure the DBnCNT field in the EXTICFGR $n$  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.

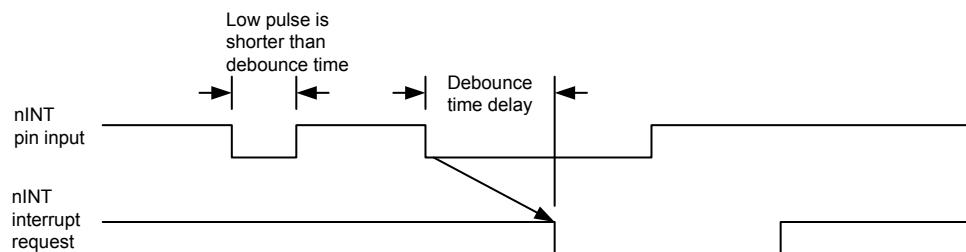


Figure 27. EXTI Interrupt Debounce Function

## Register Map

The following table shows the EXTI registers and reset values.

**Table 27. 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

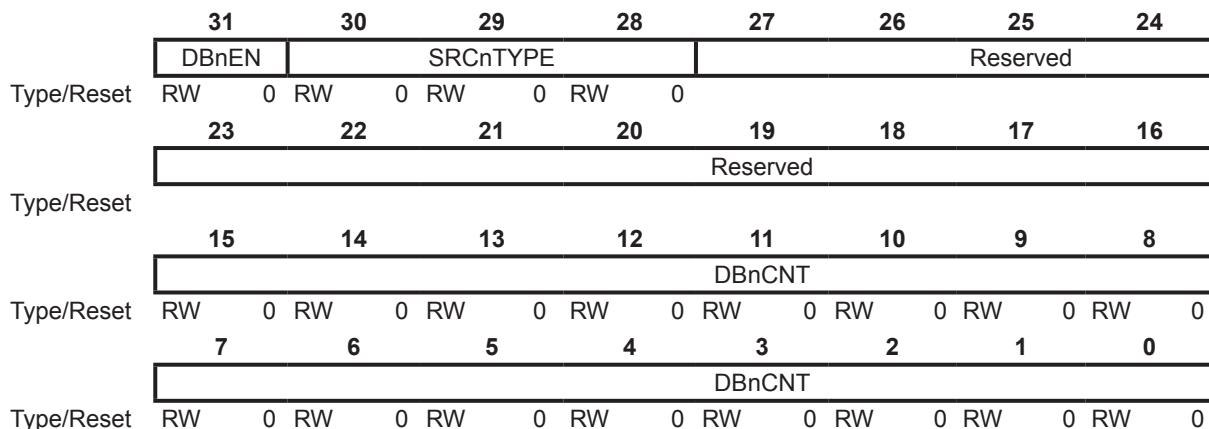
## Register Descriptions

### 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



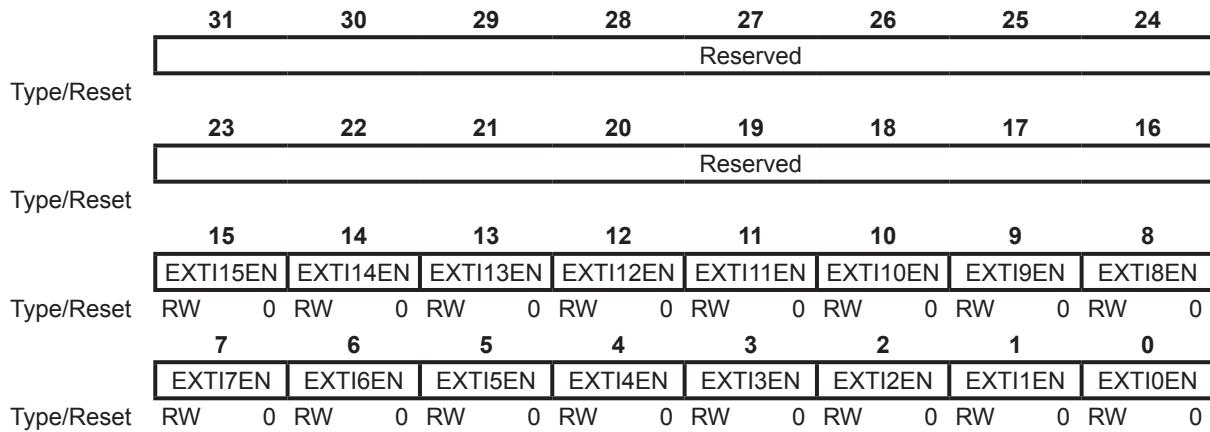
Bits	Field	Descriptions																								
[31]	DBnEN	EXTIn De-bounce Circuit Enable Bit (n = 0 ~ 15) 0: De-bounce circuit is disabled 1: De-bounce circuit is enabled																								
[30:28]	SRCnTYPE	EXTIn Interrupt Source Trigger Type (n = 0 ~ 15)																								
		<table border="1"> <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



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

## EXTI Interrupt Edge Flag Register – EXTIEDGEFLGR

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

Offset: 0x044

Reset value: 0x0000\_0000

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

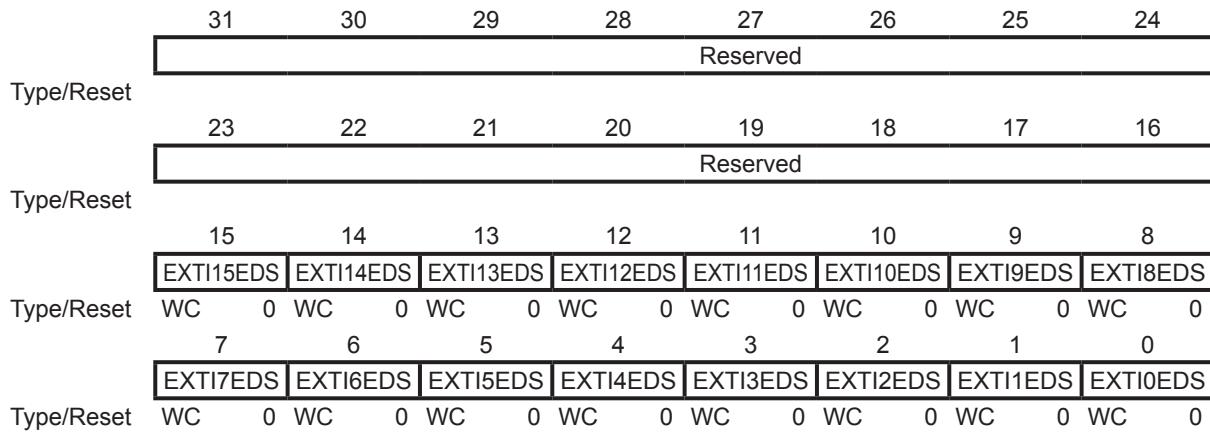
Bits	Field	Descriptions
[15:0]	EXTInEDF	<p>EXTIn Both Edge Detection Flag (n = 0 ~ 15)</p> <p>0: No edge is detected</p> <p>1: Positive or negative edge is detected</p> <p>This bit is set by the hardware circuitry when a positive or negative edge is detected on the corresponding EXTI line. Software should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

## EXTI Interrupt Edge Status Register – EXTIEDGESR

This register indicates the polarity of a detected EXTI edge.

Offset: 0x048

Reset value: 0x0000\_0000



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

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

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
Type/Reset	EVWUPIEN				Reserved			
	RW	0						
	23	22	21	20	19	18	17	16
Type/Reset					Reserved			
	15	14	13	12	11	10	9	8
Type/Reset	EXTI15WEN	EXTI14WEN	EXTI13WEN	EXTI12WEN	EXTI11WEN	EXTI10WEN	EXTI9WEN	EXTI8WEN
	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
Type/Reset	EXTI7WEN	EXTI6WEN	EXTI5WEN	EXTI4WEN	EXTI3WEN	EXTI2WEN	EXTI1WEN	EXTI0WEN
	RW	0	RW	0	RW	0	RW	0

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

## EXTI Interrupt Wakeup Polarity Register – EXTIWAKUPPOLR

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

Offset: 0x054

Reset value: 0x0000\_0000

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

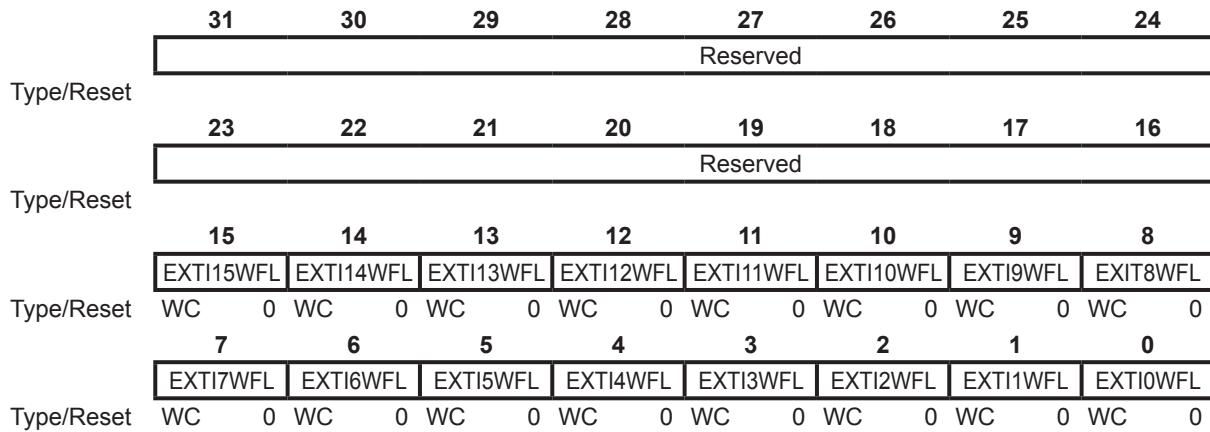
Bits	Field	Descriptions
[15:0]	EXTInWPOL	EXTIn Wakeup Polarity (n = 0 ~ 15) 0: EXTIn wakeup is high level active 1: EXTIn wakeup is low level active

## EXTI Interrupt Wakeup Flag Register – EXTIWAKUPFLG

This register is the EXTI interrupt wake flag register.

Offset: 0x058

Reset value: 0x0000\_0000



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.

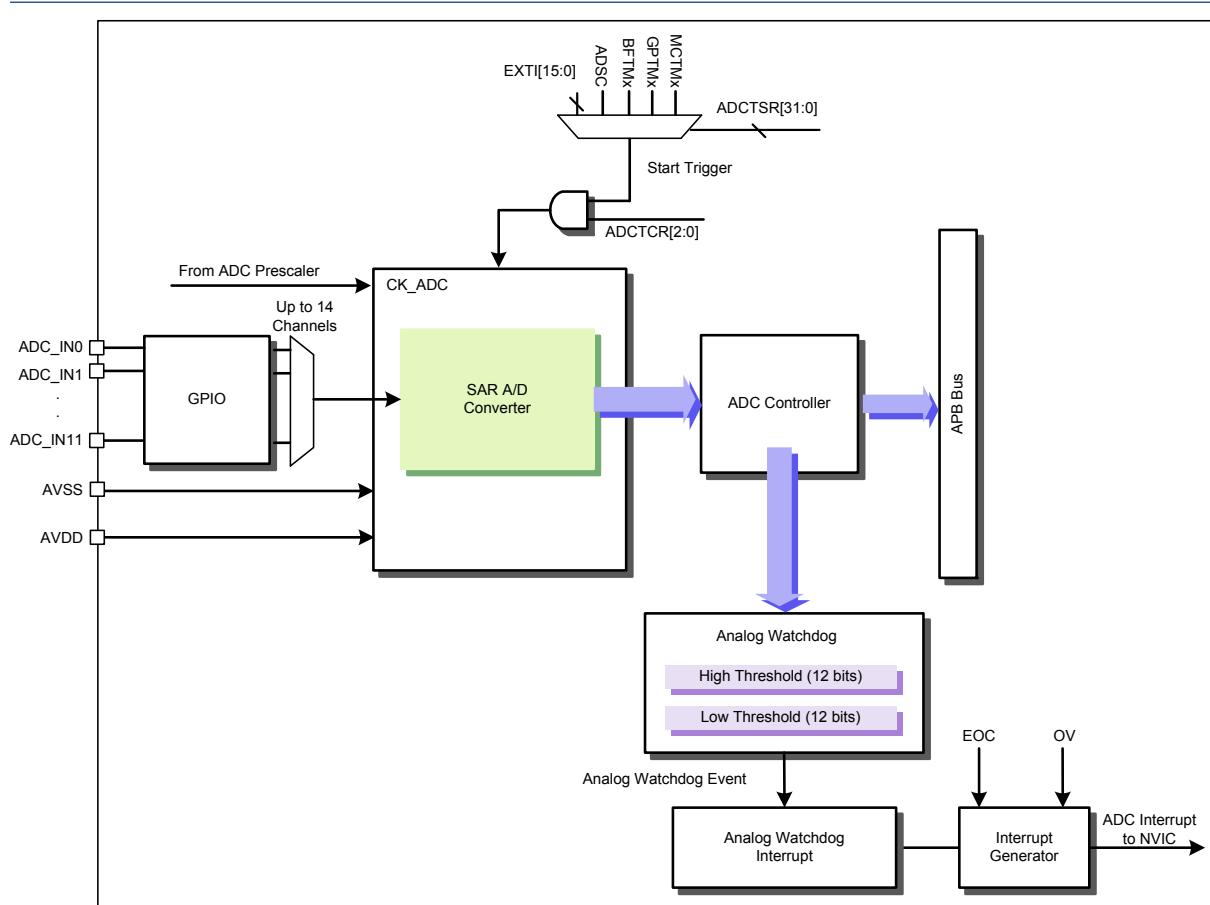


Figure 28. ADC Block Diagram

## 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
  - GPTM trigger
  - MCTM trigger
  - BFTM0 / BFTM1 trigger
- Multiple generated interrupts
  - End of single conversion
  - End of subgroup conversion
  - End of cycle conversion
  - Analog Watchdog
  - Data register overwriting

## 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 peripheral needs keeping 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 ADSEQL[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

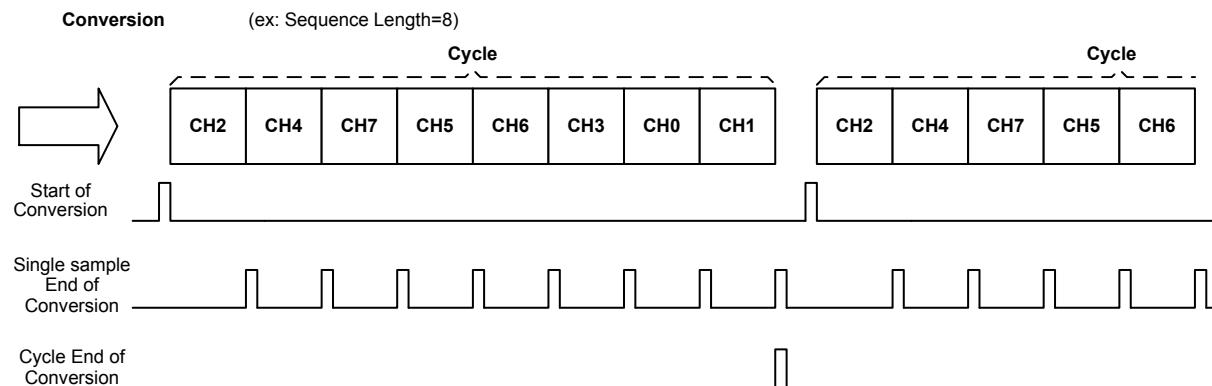
The A/D has three operating conversion modes. The conversion modes are One Shot Conversion Mode, Continuous Conversion Mode, and Discontinuous Conversion mode. Details are provided later.

#### 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, an external EXTI event or a TM event determined by the Trigger Control Register ADCTCR and the Trigger Source Register ADCTS.

#### 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.



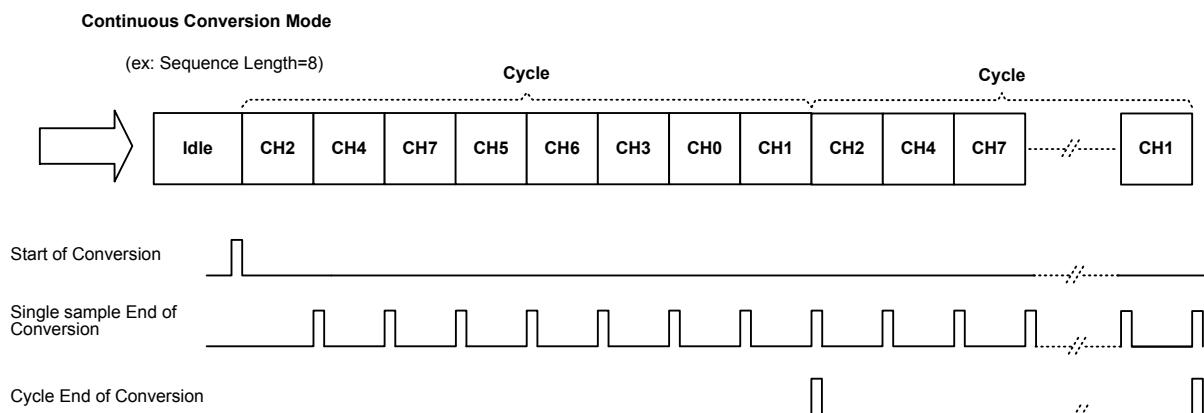
**Figure 29. One Shot Conversion Mode**

### Continuous Conversion Mode

In the Continuous Conversion Mode, repeated conversion cycle will start automatically without requiring additional A/D start trigger signals after a channels group conversion has completed. When the A/D conversion mode field ADMODE[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, an external EXTI event or a TM event determined by the Trigger Control Register ADCTCR and the Trigger Source Register ADCTS.R.

#### After conversion:

- The converted data will be stored in the 16-bit ADCDRy ( $y = 0\sim 7$ ) 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 30. Continuous Conversion Mode**

### Discontinuous Conversion Mode

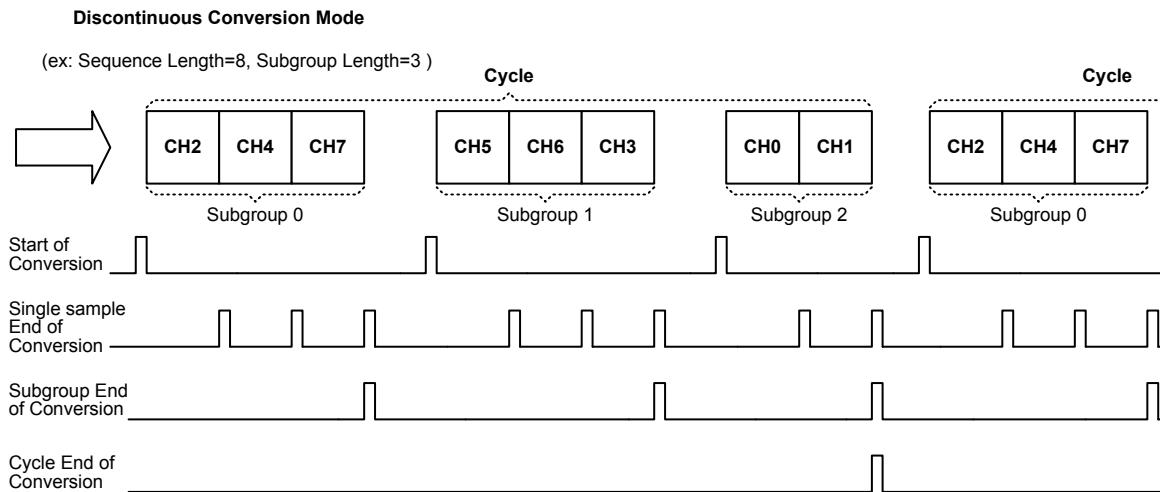
The A/D converter will operate in the Discontinuous Conversion Mode for channels group when the A/D conversion mode bit field ADMODE [1:0] in the ADCCONV register is set to 0x3. The group to be converted can have up to 8 channels and can be arranged in a specific sequence by configuring the ADCLSTn registers 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, an external EXTI event or a TM event for the groups determined by the Trigger Control Register ADCTCR and the Trigger Source Register ADCTS.R.

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 31. Discontinuous Conversion Mode**

### Start Conversion on External Event

An A/D converter conversion can be initiated by a software trigger, 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 ADCTS register to start a group channel conversion.

An A/D converter conversion can be started by setting the software trigger bit, ADSC, in the ADCTS 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, EXITn. 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 ADCTS register, the A/D converter will start a conversion when an EXTI line active edge determined in the EXIT 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 the A/D 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 28. 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.

## Register Map

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

**Table 29. 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
ADCTS	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

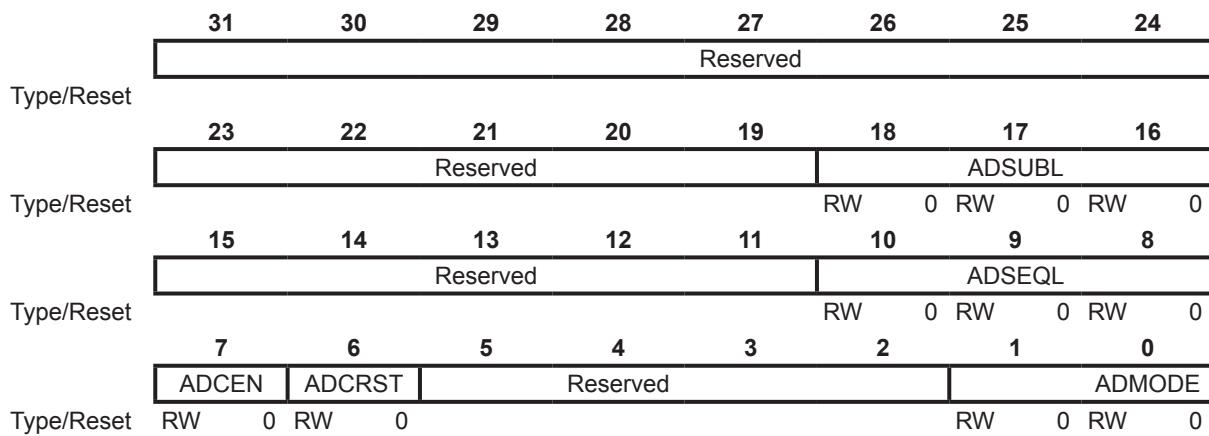
## Register Descriptions

### 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 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



Bits	Field	Descriptions
[18:16]	ADSUBL	ADC Conversion Subgroup Length 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.
[10:8]	ADSEQL	ADC Conversion Length 0x00: The channel specified by the ADSEQ0 field in the ADCLST0 register will be converted. Others: Sequence length = ADSEQL [2:0] + 1.
[7]	ADCEN	ADC Enable 0: ADC disable 1: ADC enable When this bit is cleared to 0, the A/D converter will be disabled and the CK_ADC clock will also be switched off.
[6]	ADCRST	ADC Reset 0: No effect 1: Reset A/D converter except for the A/D Converter controller.

Bits	Field	Descriptions	
[1:0]	ADMODE	ADC Conversion Mode	
		<b>ADMODE [1:0]</b>	<b>Mode</b>
		00	One shot mode
		01	Reserved
		10	Continuous mode
		11	Discontinuous mode

## 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
Type/Reset		Reserved				ADSEQ3		
				RW	0 RW	0 RW	0 RW	0
	23	22	21	20	19	18	17	16
Type/Reset		Reserved				ADSEQ2		
				RW	0 RW	0 RW	0 RW	0
	15	14	13	12	11	10	9	8
Type/Reset		Reserved				ADSEQ1		
				RW	0 RW	0 RW	0 RW	0
	7	6	5	4	3	2	1	0
Type/Reset		Reserved				ADSEQ0		
				RW	0 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 <sup>rd</sup> 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
Type/Reset		Reserved				ADSEQ7		
				RW	0 RW	0 RW	0 RW	0
	23	22	21	20	19	18	17	16
Type/Reset		Reserved				ADSEQ6		
				RW	0 RW	0 RW	0 RW	0
	15	14	13	12	11	10	9	8
Type/Reset		Reserved				ADSEQ5		
				RW	0 RW	0 RW	0 RW	0
	7	6	5	4	3	2	1	0
Type/Reset		Reserved				ADSEQ4		
				RW	0 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 <sup>th</sup> 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 conversion 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	RW	0	RW	0	RW	0	RW	0	
	ADST								

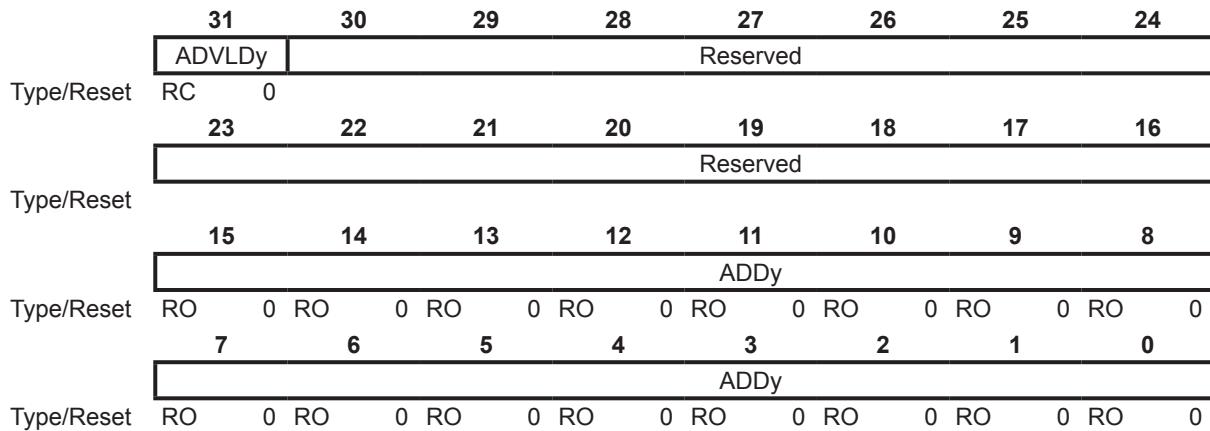
Bits	Field	Descriptions
[7:0]	ADST	ADC Input Channel Sampling Time Sampling time = (ADST[7:0] + 1.5) × CK_ADC clocks.

## 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



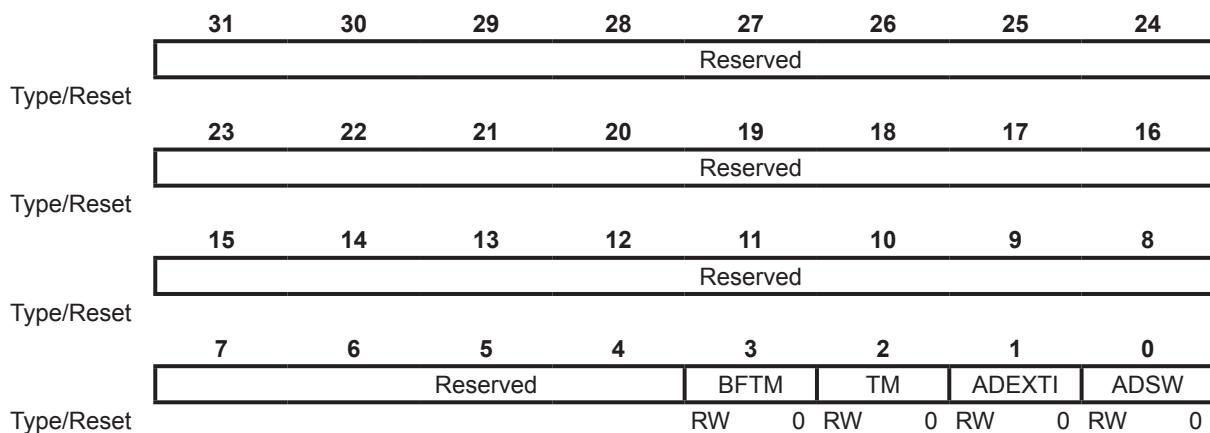
Bits	Field	Descriptions
[31]	ADVLDy	ADC Conversion Data of Sequence Order No.y Valid Bit (y = 0 ~ 7) 0: Data are invalid or have been read 1: New data is valid
[15:0]	ADDy	ADC Conversion Data of Sequence Order No.y (y = 0 ~ 7) The conversion result of Sequence Order No.y in the ADCLSTn (n=0 ~1) registers

## ADC Trigger Control Register – ADCTCR

This register contains the ADC start conversion trigger enable bits.

Offset: 0x070

Reset value: 0x0000\_0000



Bits	Field	Descriptions
[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
Type/Reset	23	22	21	20	19	18	17	16
	Reserved					BFTMS		
					RW	0	RW	0
Type/Reset	15	14	13	12	11	10	9	8
	Reserved					ADEXTIS		
					RW	0	RW	0
Type/Reset	7	6	5	4	3	2	1	0
	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 CH0O event 010: GPTM or MCTM CH1O event 011: GPTM or MCTM CH2O event 100: GPTM or MCTM CH3O event Others: Reserved – Should not be used to avoid unpredictable results
[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: GPTM 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 cleared by hardware automatically after conversion started.

## ADC Watchdog Control Register – ADCWCR

This register provides the control bits and status of the ADC watchdog function.

Offset: 0x078

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[27:24]	ADUCH	Upper Threshold Channel Status 0000: ADC_IN0 converted data is higher than the upper threshold 0001: ADC_IN1 converted data is higher than the upper threshold ... 1011: ADC_IN11 converted data is higher than the upper threshold Others: Reserved If one of these status bits is set to 1 by the watchdog monitor function, the status field value should first be stored in the user-defined memory location in the corresponding ISR. Otherwise, the ADUCH field will be changed if another input channel converted data is higher than the upper threshold.
[19:16]	ADLCH	Lower Threshold Channel Status 0000: ADC_IN0 converted data is lower than the lower threshold 0001: ADC_IN1 converted data is lower than the lower threshold ... 1011: ADC_IN11 converted data is lower than the lower threshold Others: Reserved If one of these status bits is set to 1 by the watchdog monitor function, the status field value should first be stored in the user-defined memory location in the corresponding ISR. Otherwise, the ADLCH field will be changed if another input channel converted data is lower than the lower threshold.
[11:8]	ADWCH	ADC Watchdog Specific Channel Selection 0000: ADC_IN0 is monitored 0001: ADC_IN1 is monitored ... 1011: ADC_IN11 is monitored Others: Reserved
[2]	ADWALL	ADC Watchdog Specific or All Channel Setting 0: Only the channel which specified by the ASWCH field is monitored 1: All channels are monitored

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

	31	30	29	28		27	26	25	24
Type/Reset					Reserved			ADUT	
	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					Reserved			ADLT	
	7	6	5	4		3	2	1	0
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW

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	
							RW	0
	23	22	21	20	19	18	17	16
Type/Reset				Reserved			ADIEU	ADIEL
							RW	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
						RW	0	RW

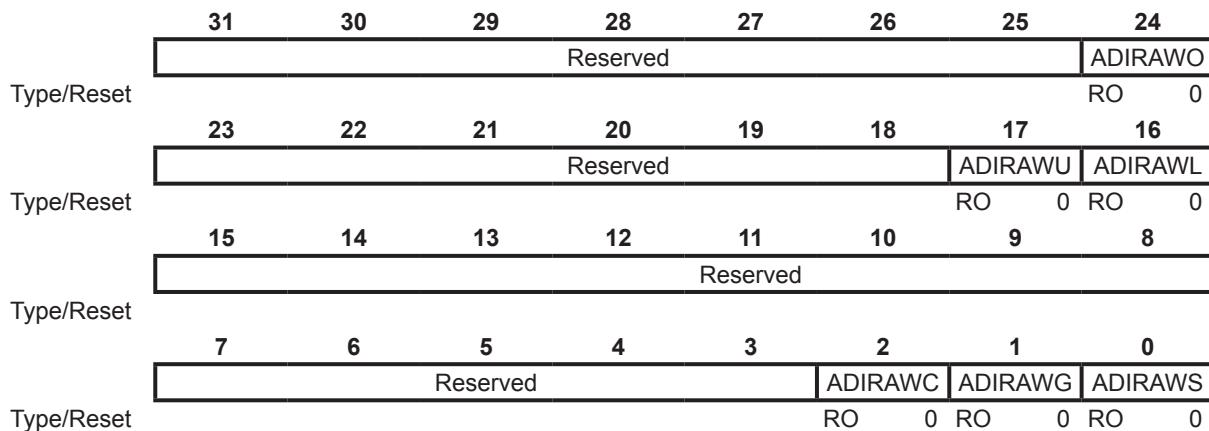
Bits	Field	Descriptions
[24]	ADIEO	ADC Data Register Overwrite Interrupt enable 0: ADC data register overwrite interrupt is disabled 1: ADC data register overwrite interrupt is enabled
[17]	ADIEU	ADC Watchdog Upper Threshold Interrupt enable 0: ADC watchdog upper threshold interrupt is disabled 1: ADC 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



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 occur 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

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

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

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							
Type/Reset	7	6	5	4	3	2	1	0
	Reserved					ADICLRC	ADICLRG	ADICLRS
						WO	0	WO
						WO	0	WO

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

# 13 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.

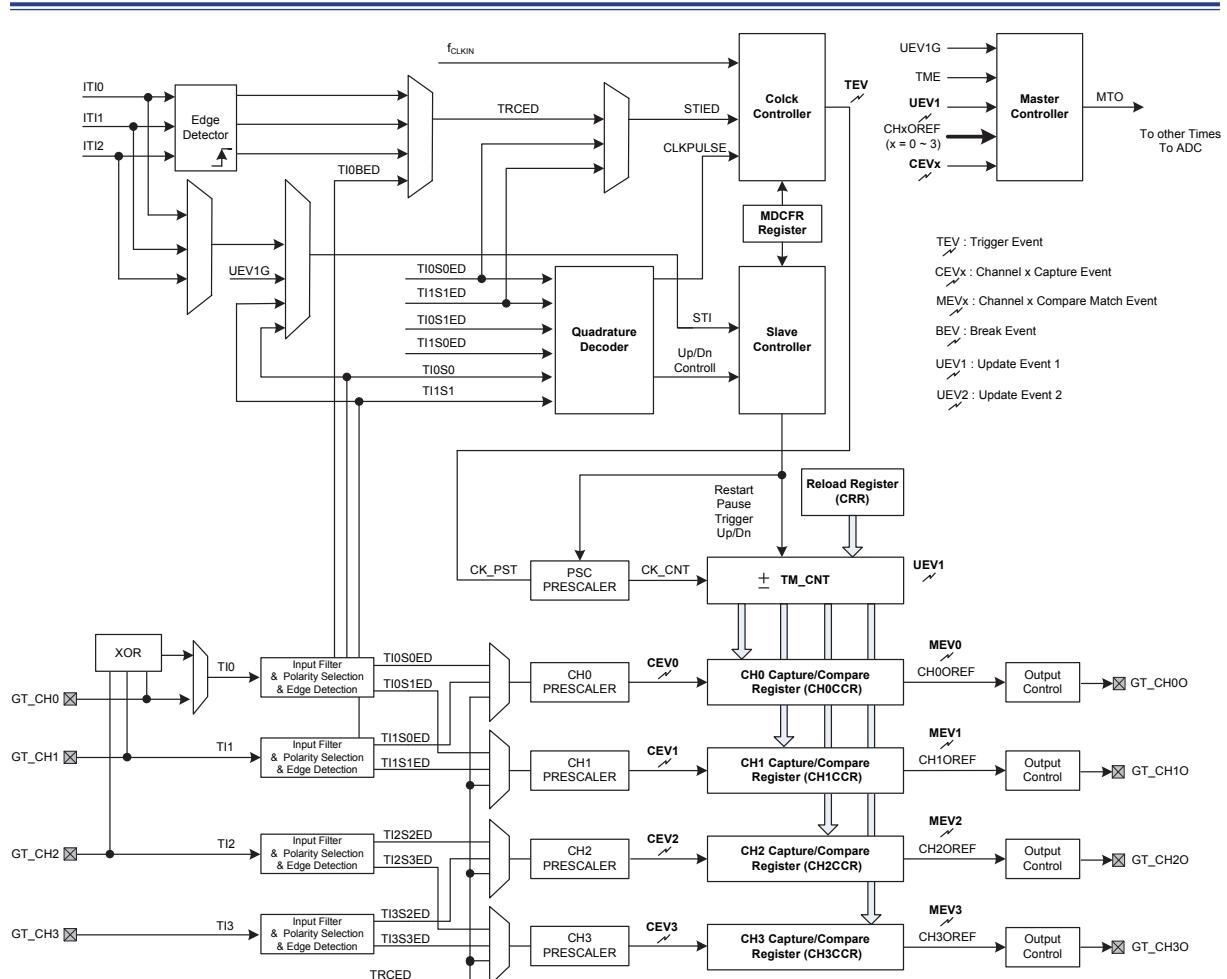


Figure 32. GPTM Block Diagram

## 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 generation with the following events:
  - Update event
  - Trigger event
  - Input capture event
  - Output compare match event
- GPTM Master/Slave mode controller

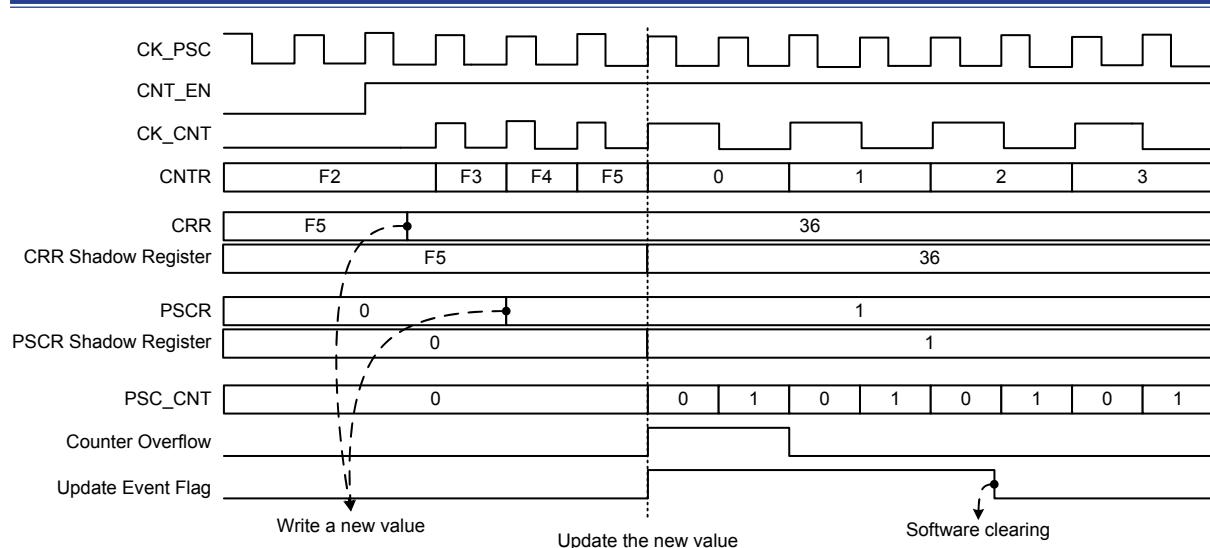
## Functional Descriptions

### Counter Mode

#### Up-Counting

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

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

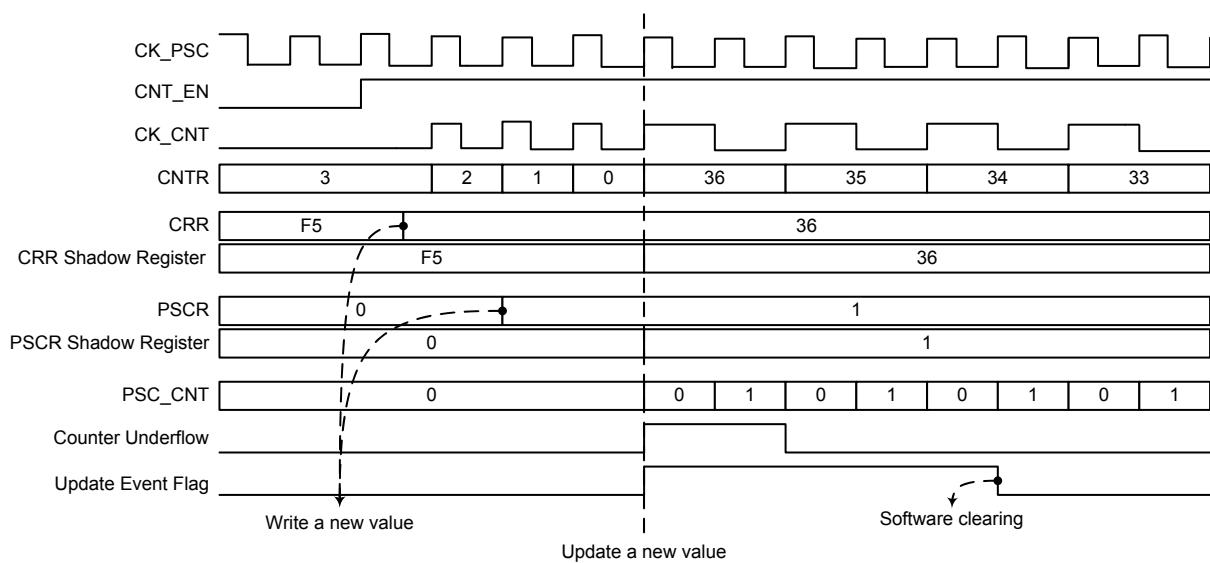


**Figure 33. Up-counting Example**

### 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.



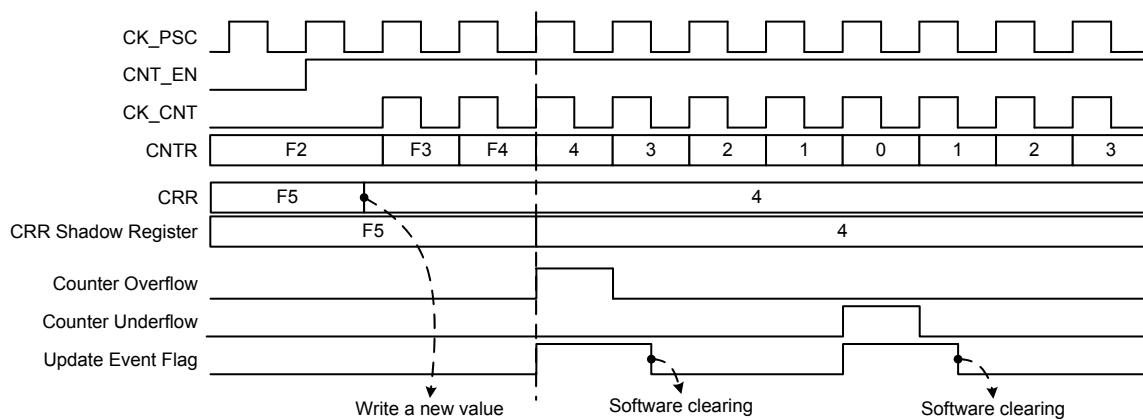
**Figure 34. Down-counting Example**

### 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 35. Center-aligned Counting Example**

## Clock Controller

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

■ Internal APB clock  $f_{CLKIN}$ :

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

■ Quadrature Decoder:

To select Quadrature Decoder mode the SMSEL field should be set to 0x1, 0x2 or 0x3 in the MDCFR register. The Quadrature Decoder function uses two input states of the GT\_CH0 and GT\_CH1 pins to generate the clock pulse to drive the counter prescaler. The counting direction bit DIR is modified by hardware automatically at each transition on the input source signal. The input source signal can be derived from the GT\_CH0 pin only, the GT\_CH1 pin only or both GT\_CH0 and GT\_CH1 pins.

■ STIED:

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

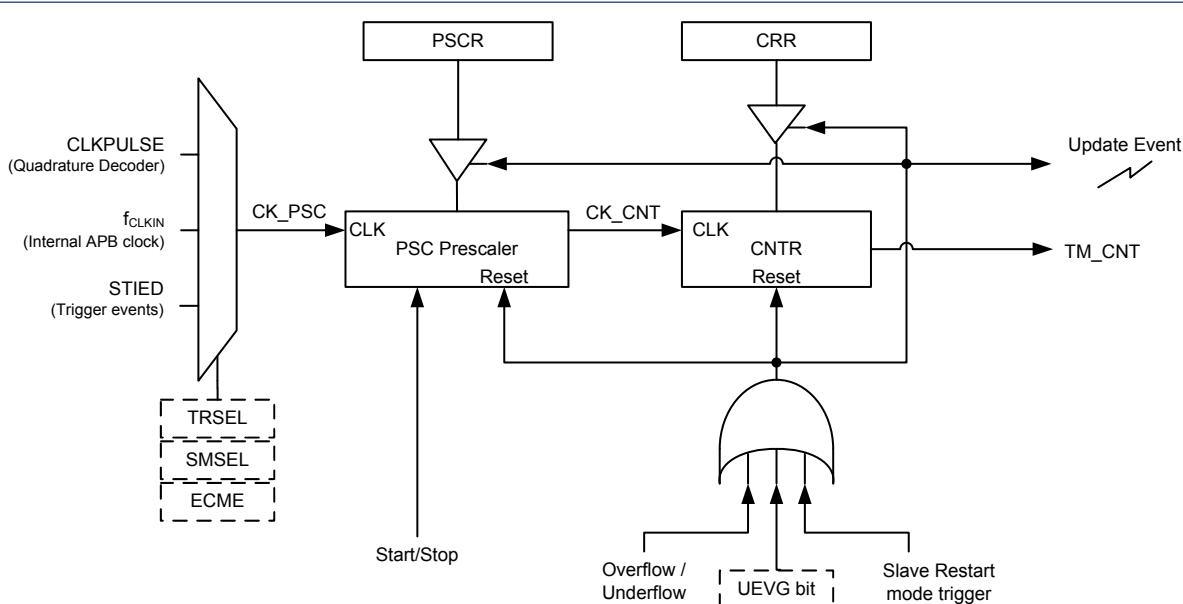
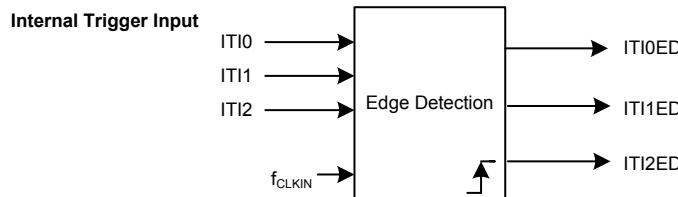


Figure 36. GPTM Clock Selection Source

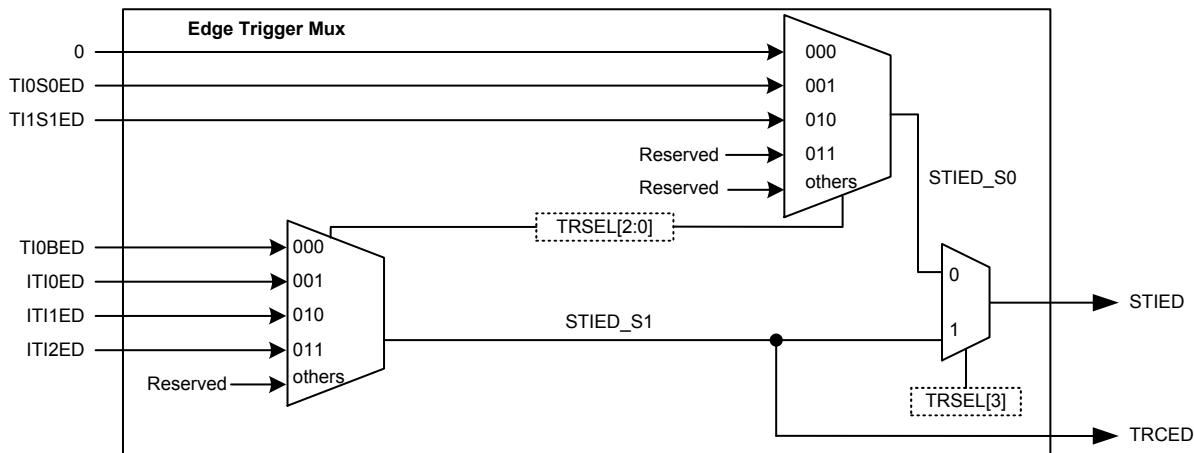
## Trigger Controller

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 UEVG bit software trigger, the internal edge detection circuitry will generate a clock pulse at each trigger signal rising edge to stimulate some GPTM functions which are triggered by a trigger signal rising edge.

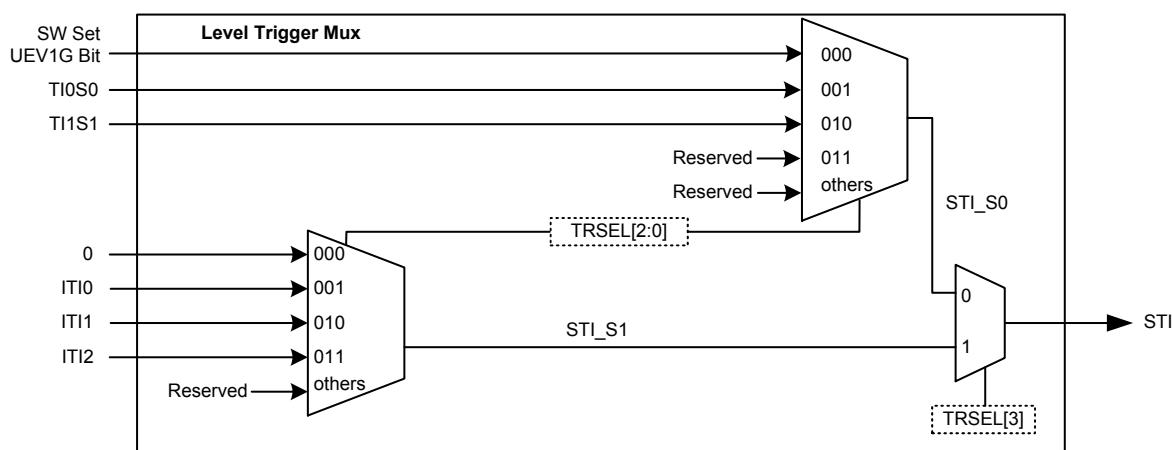
Trigger Controller Block = Edge Trigger Mux + Level Trigger Mux



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



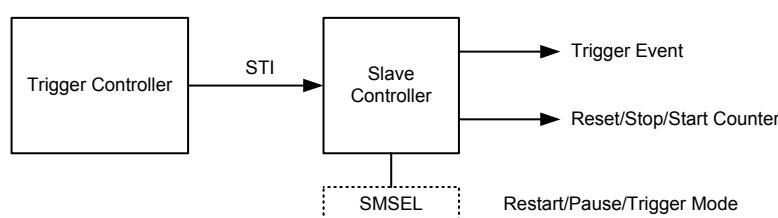
Level Trigger Source = Internal (ITIx) + Channel input (TIn) + Software UEV1G bit



**Figure 37. Trigger Controller Block**

## Slave Controller

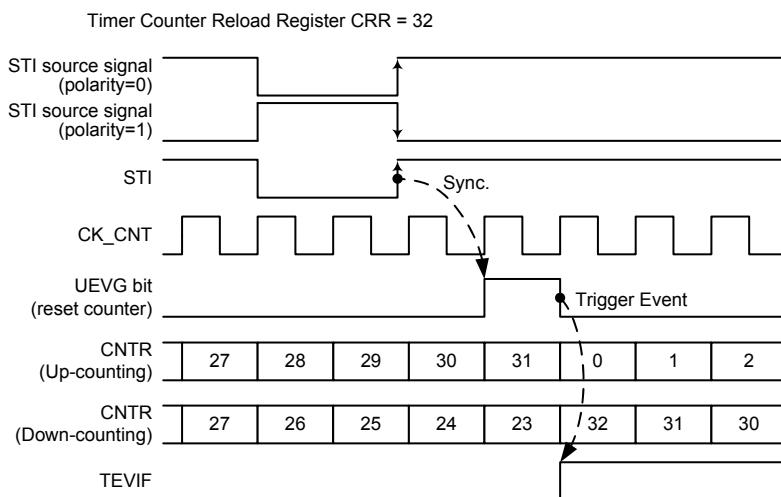
The GPTM can be synchronized with an external trigger in several modes including the Restart mode, the Pause mode and the Trigger mode which is selected by the SMSEL field in the MDCFR register. The trigger input of these modes comes from the STI signal which is selected by the TRSEL field in the TRCFR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



**Figure 38. Slave Controller Diagram**

### Restart Mode

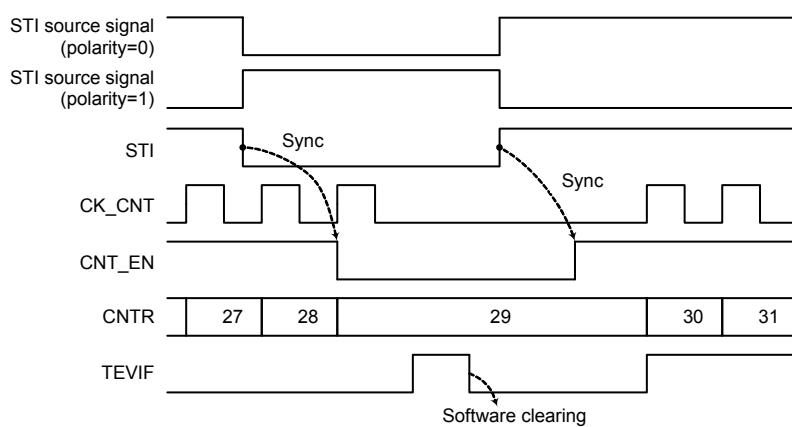
The counter and its prescaler can be reinitialized in response to a rising edge of the STI signal. When a STI rising edge occurs, the update event software generation bit named UEVG will automatically be asserted by hardware and the trigger event flag will also be set. Then the counter and prescaler 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 39. 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 40. 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.

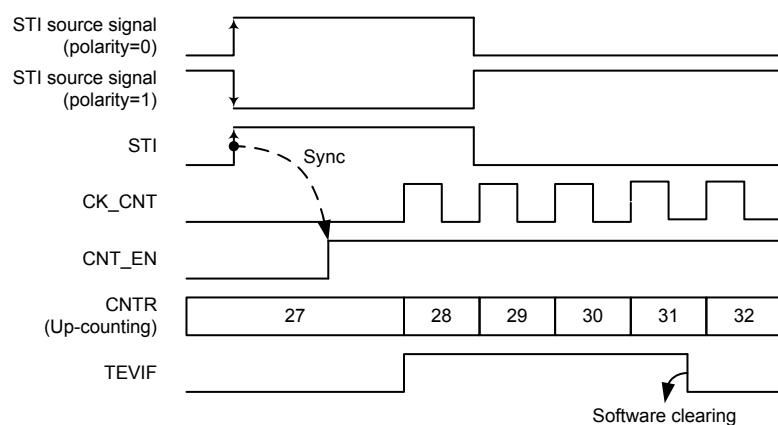


Figure 41. GPTM in Trigger Mode

## 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.

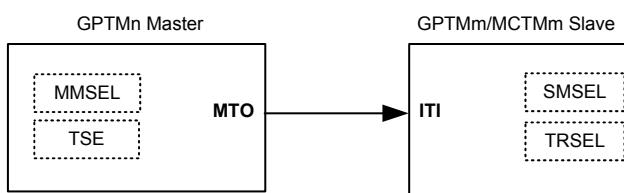


Figure 42. Master GPTMn and Slave GPTMm/MCTMm Connection

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.

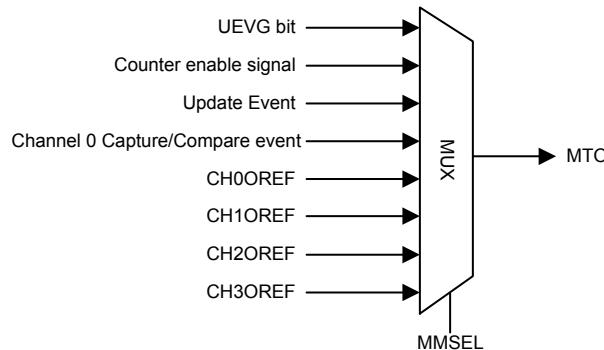


Figure 43. MTO Selection

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.

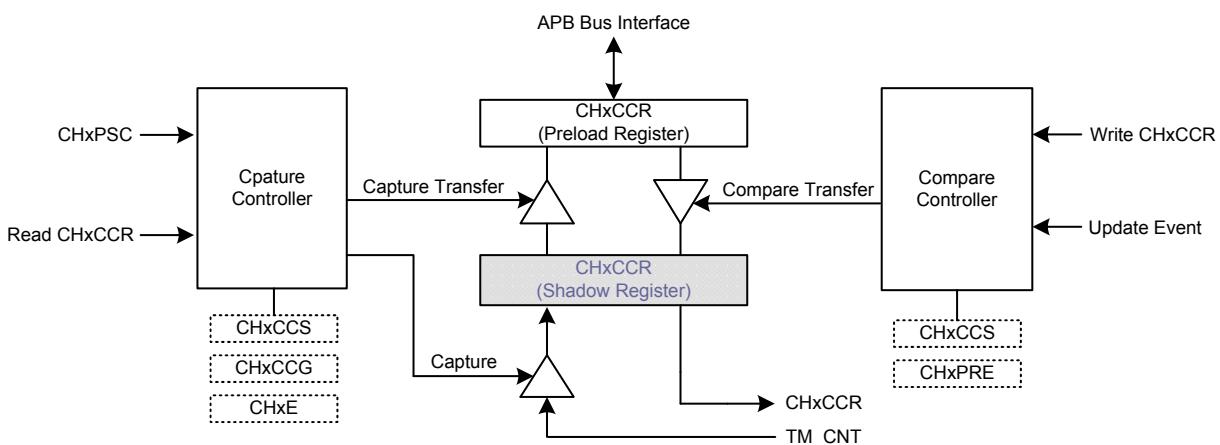
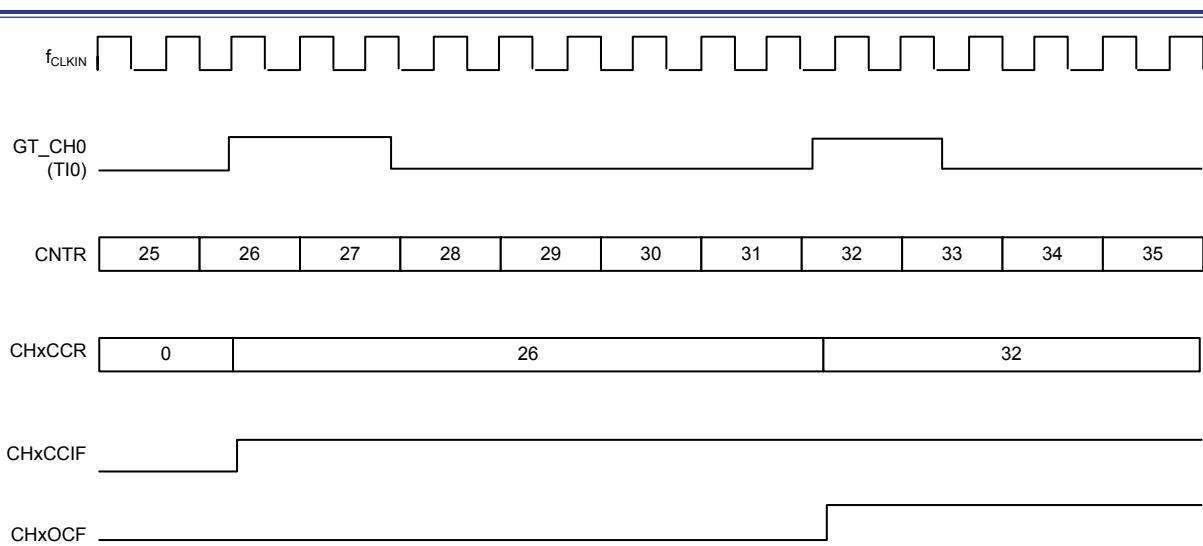


Figure 44. Capture/Compare Block Diagram

### 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.



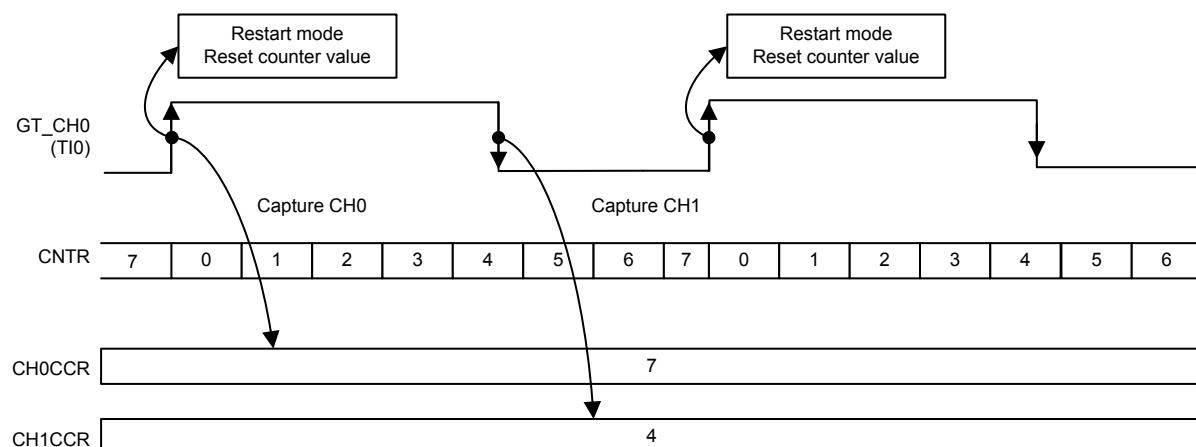
**Figure 45. Input Capture Mode**

### 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.

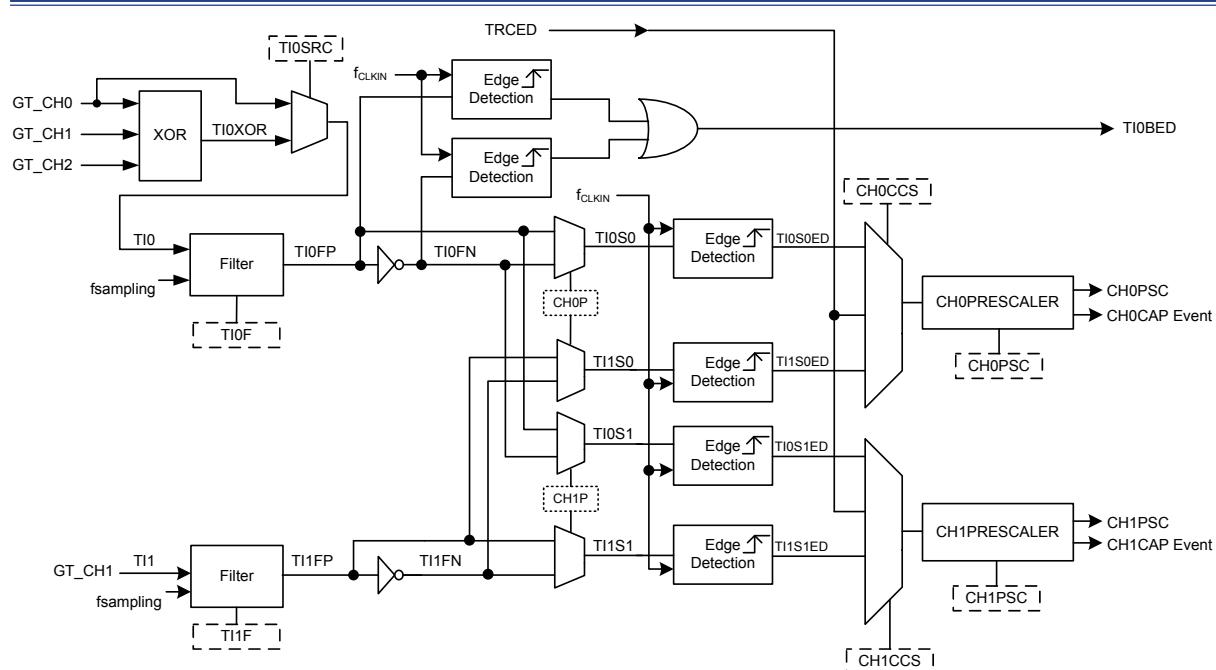
As the following diagram shows, the high pulse width on the GT\_CH0 pin will be captured into the CH1CCR register while the input period will be captured into the CH0CCR register after input capture operation.



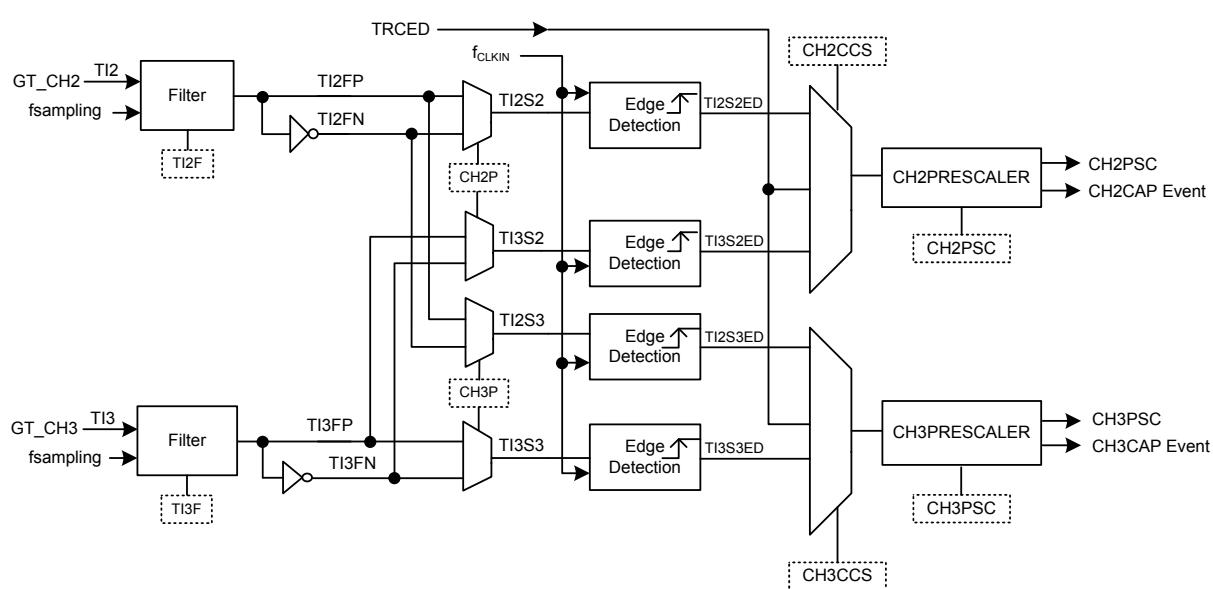
**Figure 46. PWM Pulse Width Measurement Example**

## 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 (TI0) 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 register (CHxPSC).



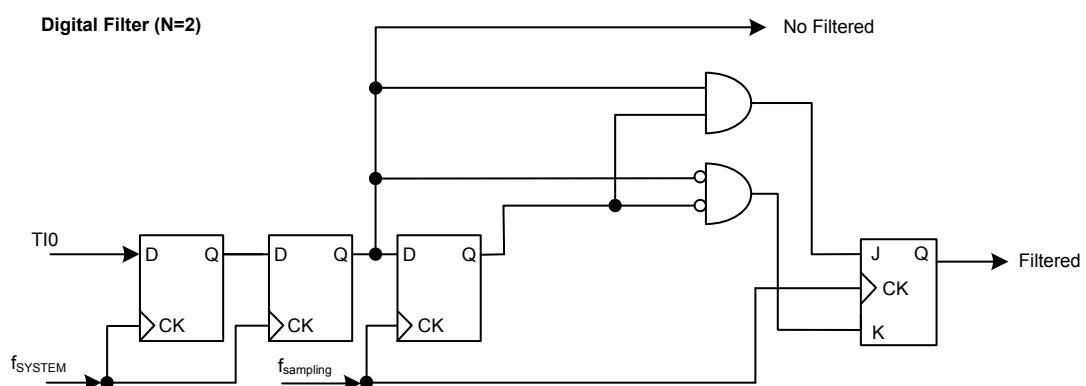
**Figure 47. Channel 0 and Channel 1 Input Stages**



**Figure 48. Channel 2 and Channel 3 Input Stages**

### 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.



**Figure 49. TI0 Digital Filter Diagram with N = 2**

## Quadrature Decoder

The Quadrature Decoder function uses two quadrantal inputs TI0 and TI1 derived from the GT\_CH0 and GT\_CH1 pins respectively to interact to generate the counter value. The DIR bit is modified by hardware automatically during each input source transition. The input source can be either TI0 only, TI1 only or both TI0 and TI1, the selection made by setting the SMSEL field to 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.

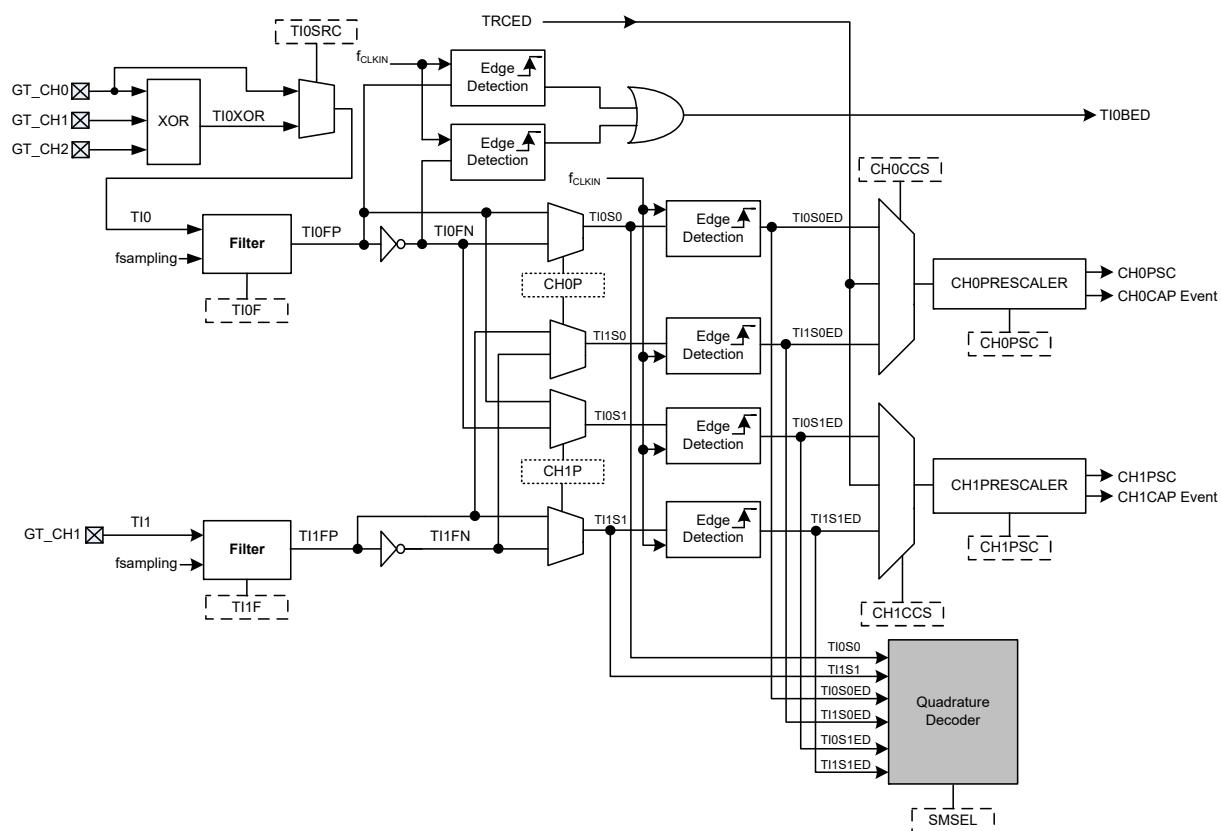
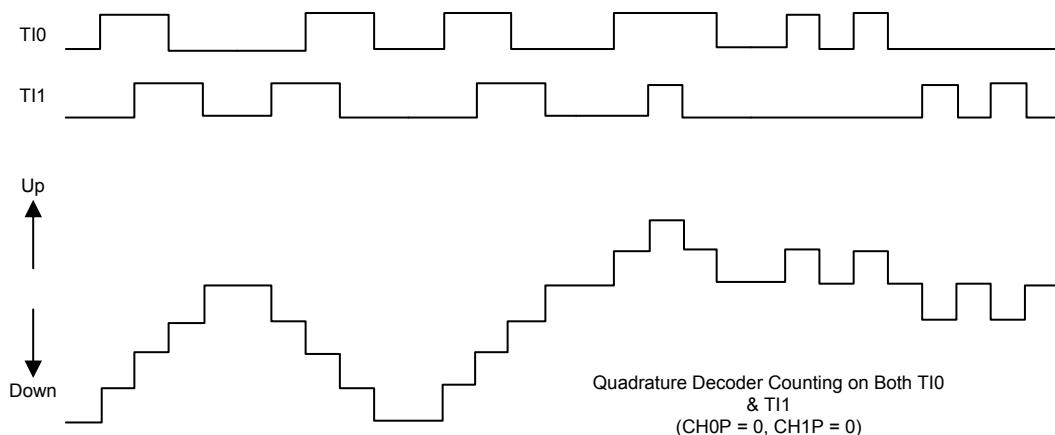


Figure 50. Input Stage and Quadrature Decoder Block Diagram

**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

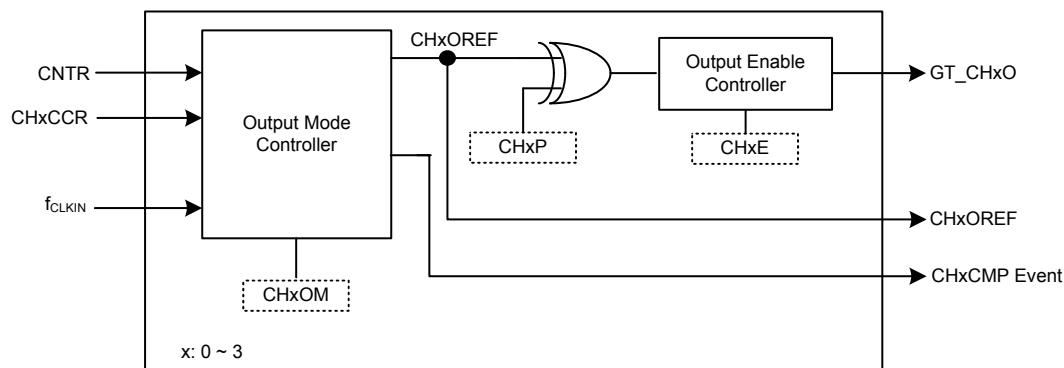
Note: “—” → means “no counting”; “X” → impossible



**Figure 51. 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.



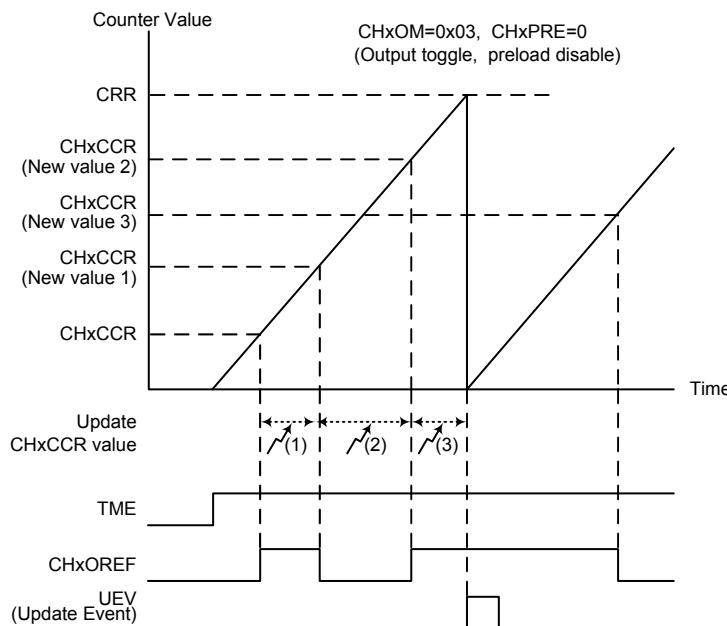
**Figure 52. Output Stage Block Diagram**

### Channel Output Reference Signal

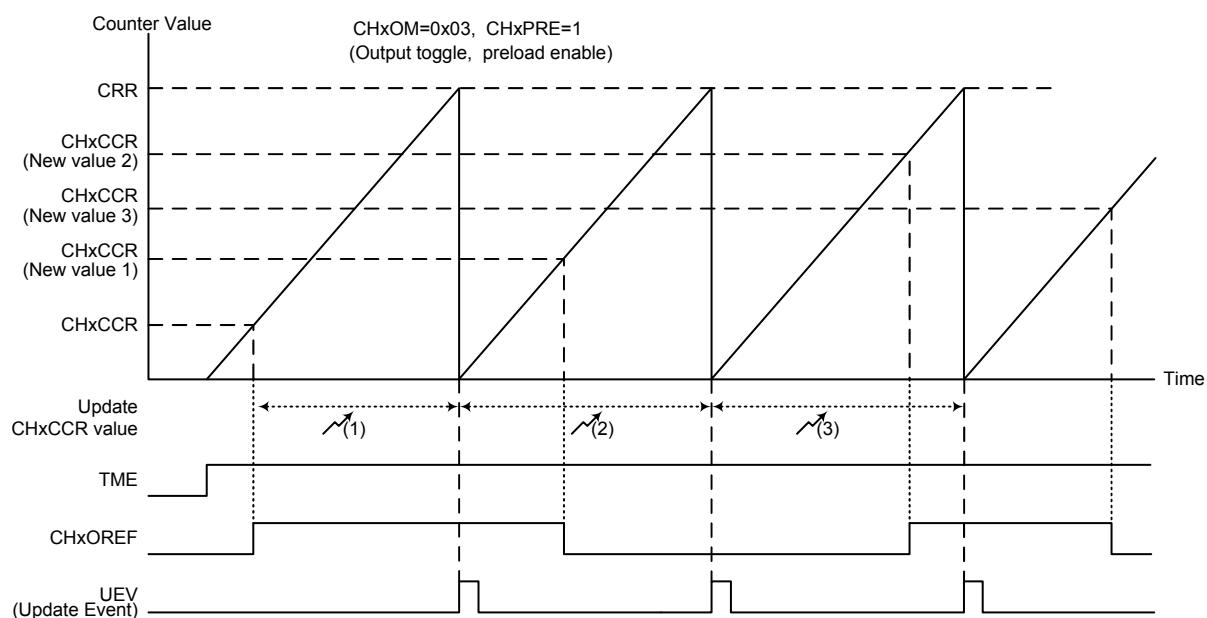
When the GPTM is used in the compare match output mode, the CHxOREF signal (Channel x Output Reference signal) is defined by the CHxOM bit setup. The CHxOREF signal has several types of output function which defines what happens to the output when the counter value matches the contents of the CHxCCR 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 30 shows a summary of the output type setup.

**Table 31. Compare Match Output Setup**

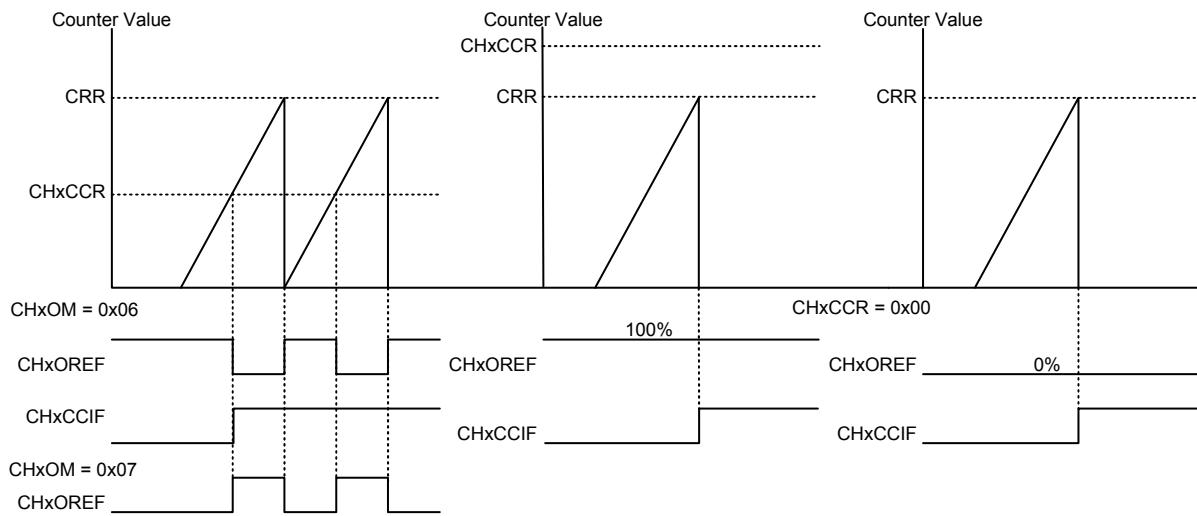
CHxOM value	Compare Match Level
0x00	No change
0x01	Clear Output to 0
0x02	Set Output to 1
0x03	Toggle Output
0x04	Force Inactive Level
0x05	Force Active Level
0x06	PWM Mode 1
0x07	PWM Mode 2



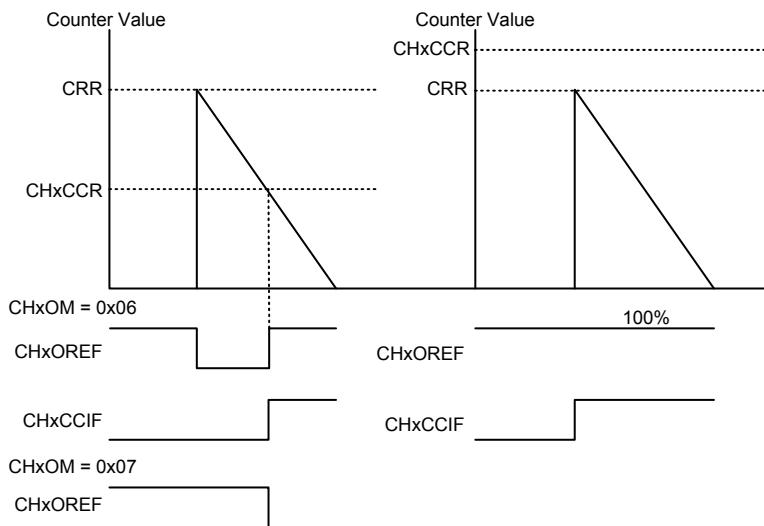
**Figure 53. Toggle Mode Channel Output Reference Signal – CHxPRE = 0**



**Figure 54. Toggle Mode Channel Output Reference Signal – CHxPRE = 1**



**Figure 55. PWM Mode Channel Output Reference Signal and Counter in Up-counting Mode**



**Figure 56. PWM Mode Channel Output Reference Signal and Counter in Down-counting Mode**

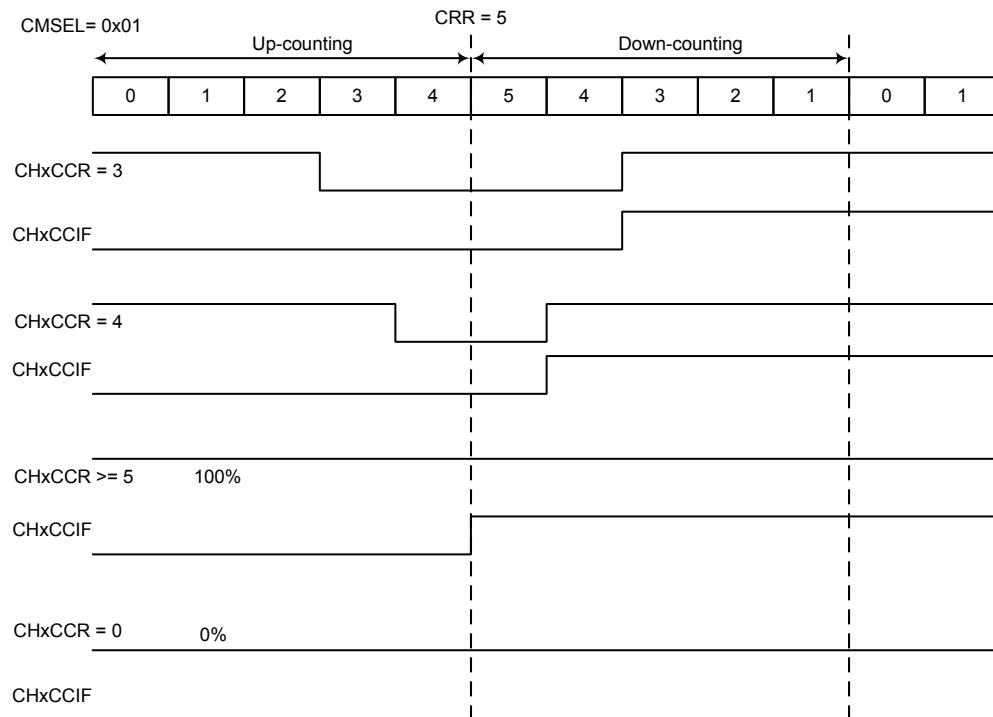


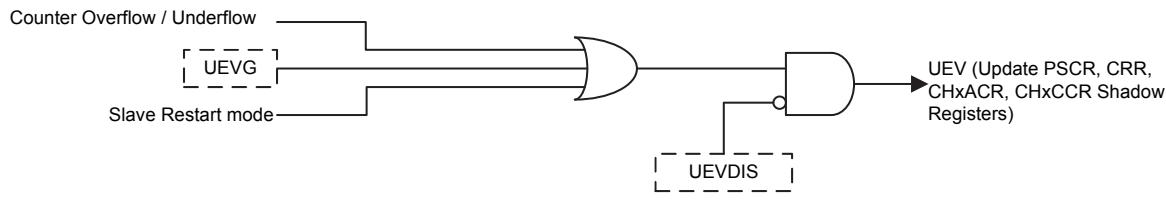
Figure 57. PWM Mode Channel Output Reference Signal and Counter in Centre-align Mode

## Update Management

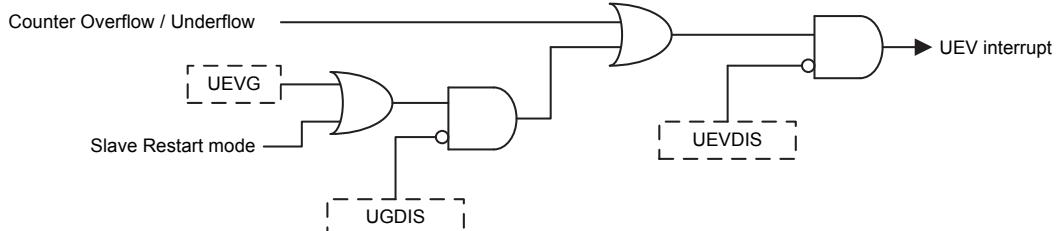
The Update event is used to update the CRR, the PSCR, the CHxACR and the CHxCCR values from the actual registers to the corresponding shadow registers. An update event occurs when the counter overflows or underflows, the software update control bit is triggered or an update event from the slave controller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



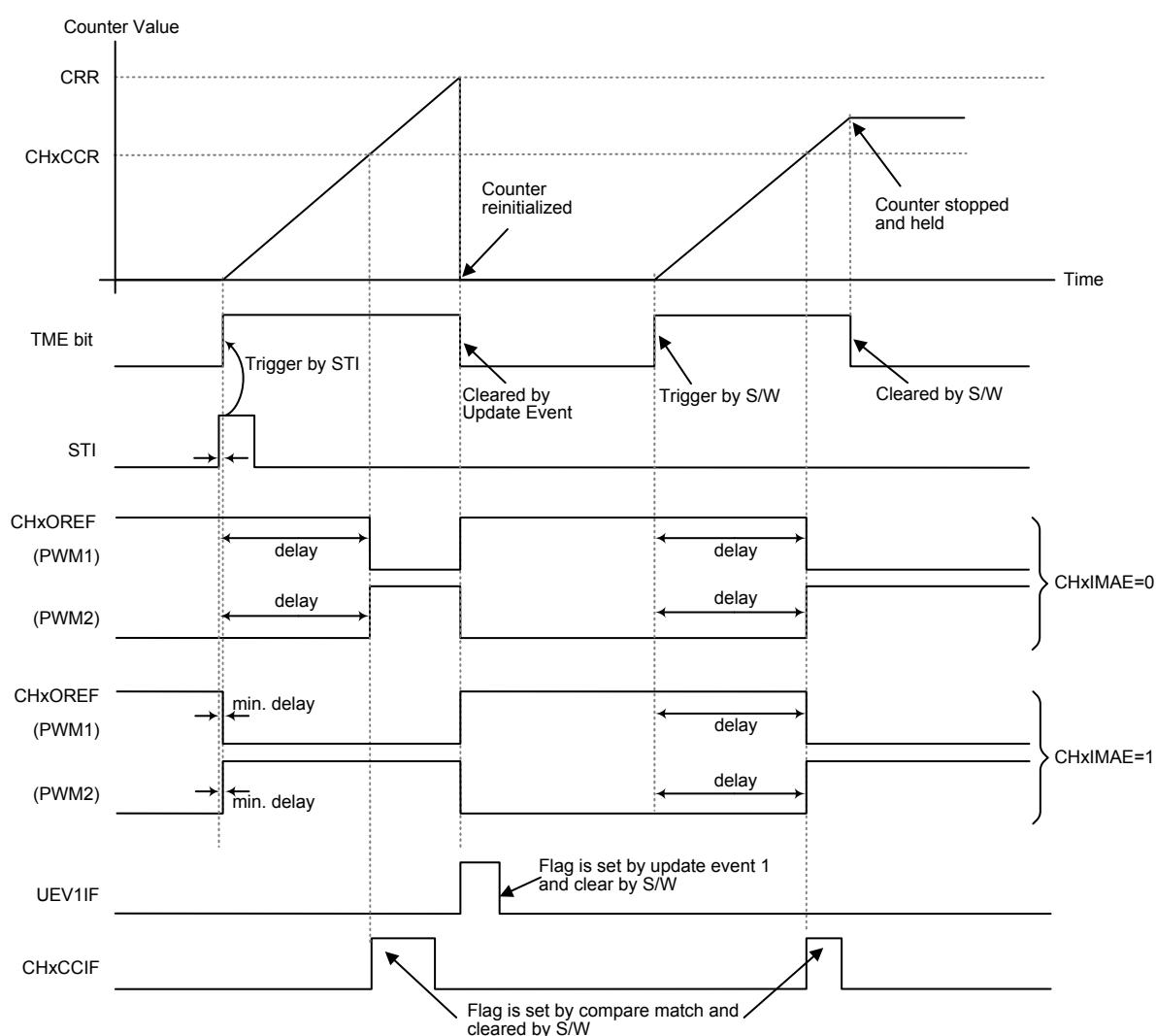
### Update Event Interrupt Management



**Figure 58. Update Event Setting Diagram**

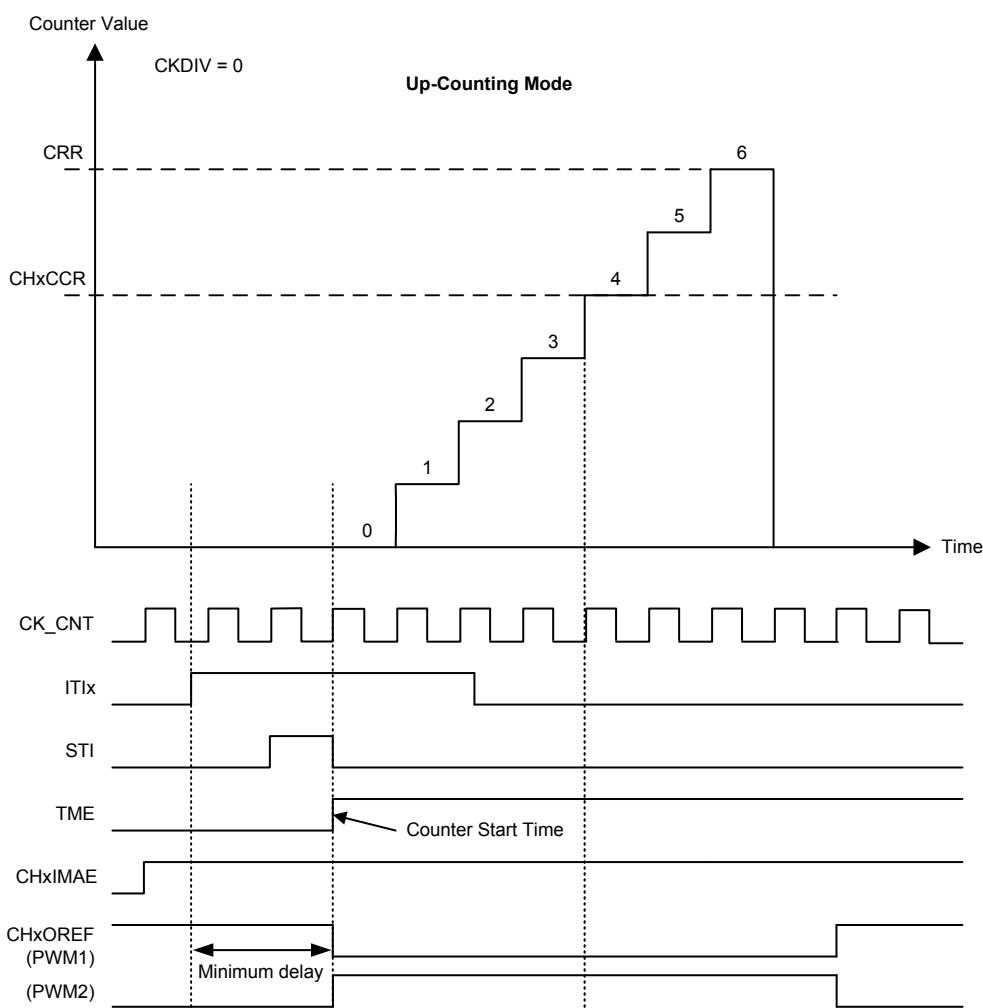
## Single Pulse Mode

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 59. Single Pulse Mode**

In the Single Pulse mode, the STI active edge which sets the TME bit to 1 will enable the counter. However, there exist several clock delays to perform the comparison result between the counter value and the CHxCCR value. In order to reduce the delay to a minimum value, the user can set the CHxIMAE bit in each CHxOCFR register. After 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.

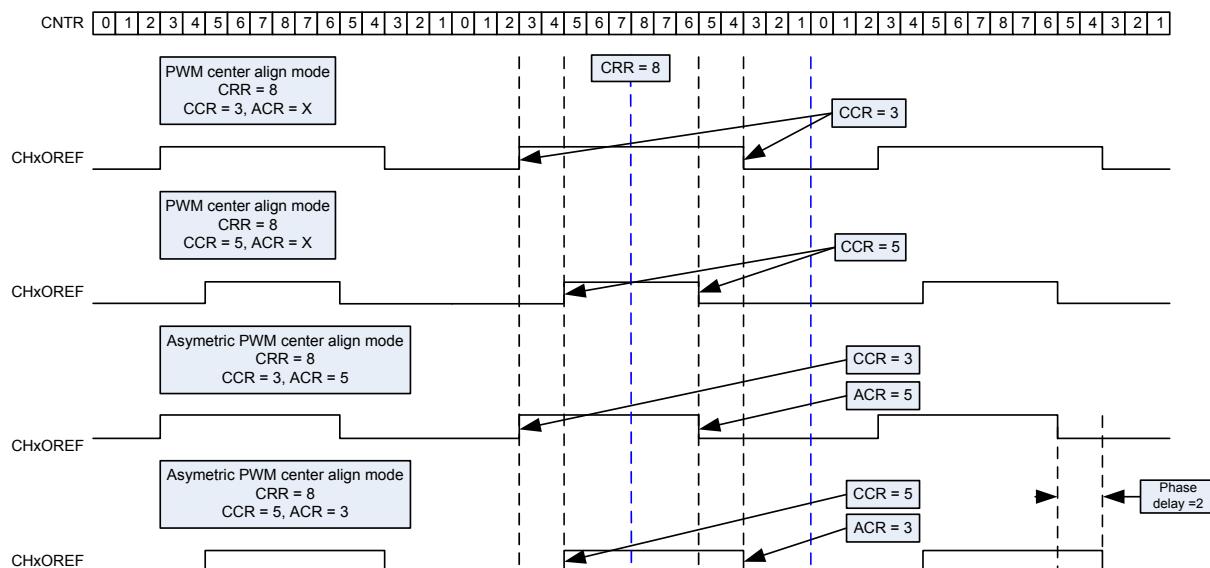


**Figure 60. Immediate Active Mode Minimum Delay**

## 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 in counting down stage, the value in the CHxACR register is used as the down-count compare value. The Figure 60 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.



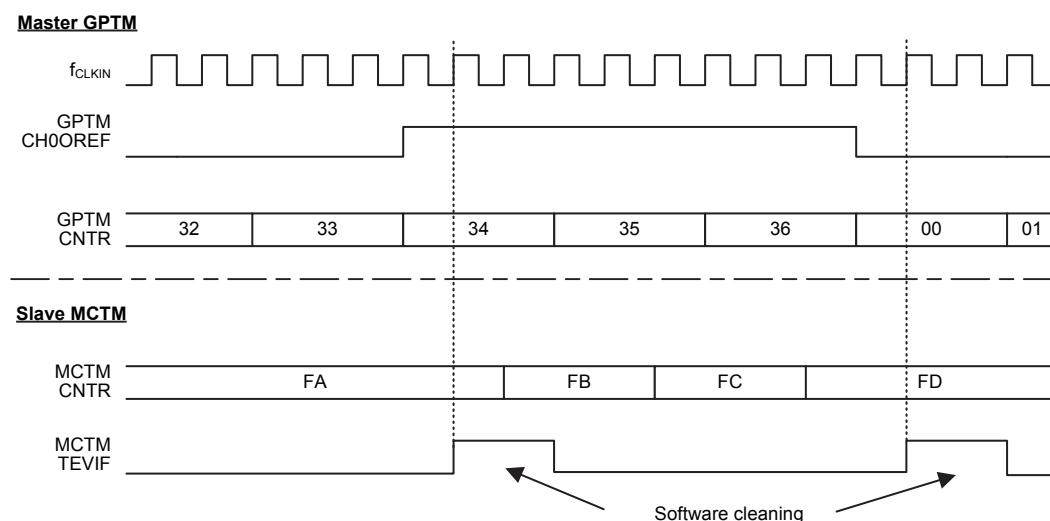
**Figure 61. Asymmetric PWM Mode versus Center Align Counting Mode**

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



**Figure 62. Pausing MCTM using the GPTM CH0OREF Signal**

### Using one timer to trigger another timer start counting

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

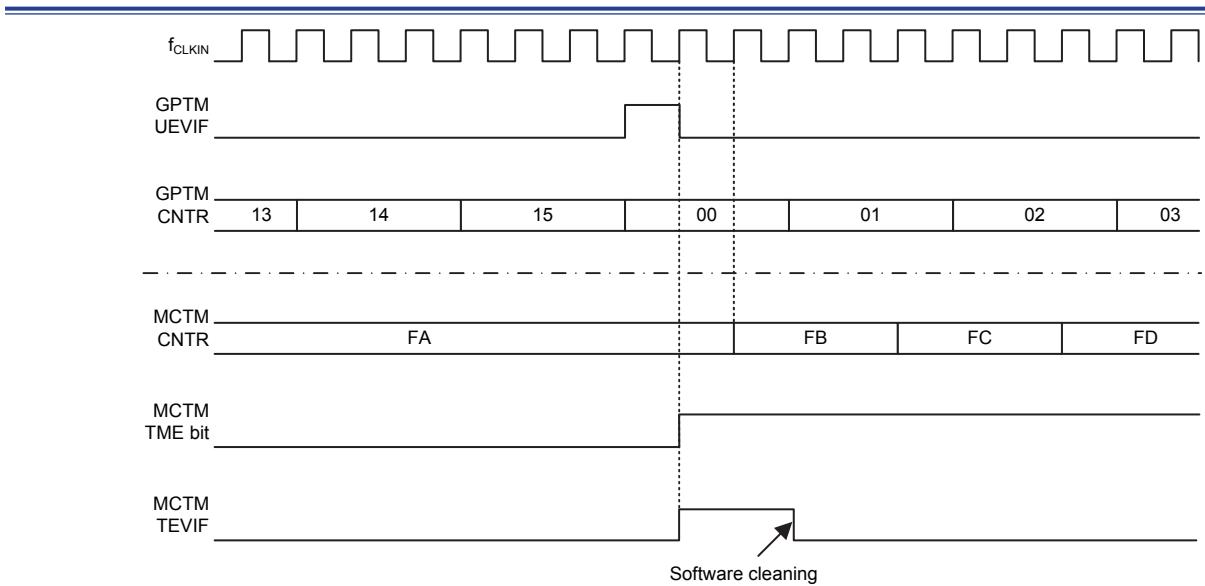
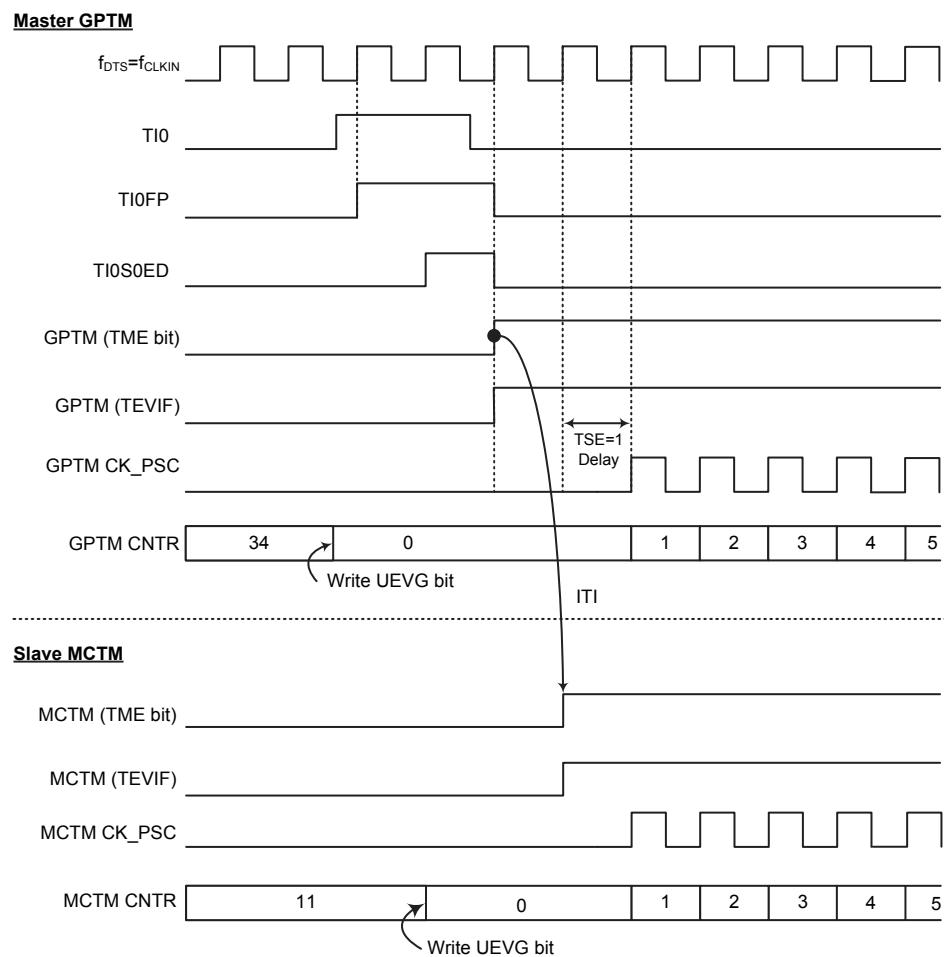


Figure 63. Triggering MCTM with GPTM Update Event

### Starting two timers synchronously in response to an external trigger

- Configure GPTM to operate in the master mode to send its enable signal as a trigger output (MMSEL = 0x01).
- Configure GPTM slave mode to receive its input trigger source from GT\_CH0 pin (TRSEL = 0x01).
- Configure GPTM to be in the slave trigger mode (SMSEL = 0x06).
- Enable the GPTM master timer synchronization function by setting the TSE bit in the MDCFR register to 1 to synchronize the slave timer.
- Configure MCTM to receive its input trigger source from the GPTM trigger output (TRSEL = 0x0A).
- Configure MCTM to be in the slave trigger mode (SMSEL = 0x06).



**Figure 64. Trigger GPTM and MCTM with the GPTM 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 ( $x = 0 \sim 3$ ) to be used as an Analog-to-Digital Converter input trigger signal.

## 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 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

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

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
[1]	UGDIS	Update event interrupt generation disable control 0: Any of the following events will generate an update interrupt - Counter overflow/underflow - Setting the UEVG bit - Update generation through the slave mode 1: Only counter overflow/underflow generates an update interrupt

Bits	Field	Descriptions
[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"><li>- Counter overflow/underflow</li><li>- Setting the UEVG bit</li><li>- Update generation through the slave mode</li></ul> <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 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	
							RW	0
	23	22	21	20	19	18	17	16
Type/Reset				Reserved			MMSEL	
							RW	0
	15	14	13	12	11	10	9	8
Type/Reset				Reserved			SMSEL	
							RW	0
	7	6	5	4	3	2	1	0
Type/Reset				Reserved			TSE	
							RW	0

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

Bits	Field	Descriptions	
[18:16]	MMSEL	Master Mode Selection  Master mode selection is used to select the MTO signal source which is used to synchronize the other slave timer.	
<b>MMSEL [2:0]</b>			
		<b>Mode</b>	<b>Descriptions</b>
	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.

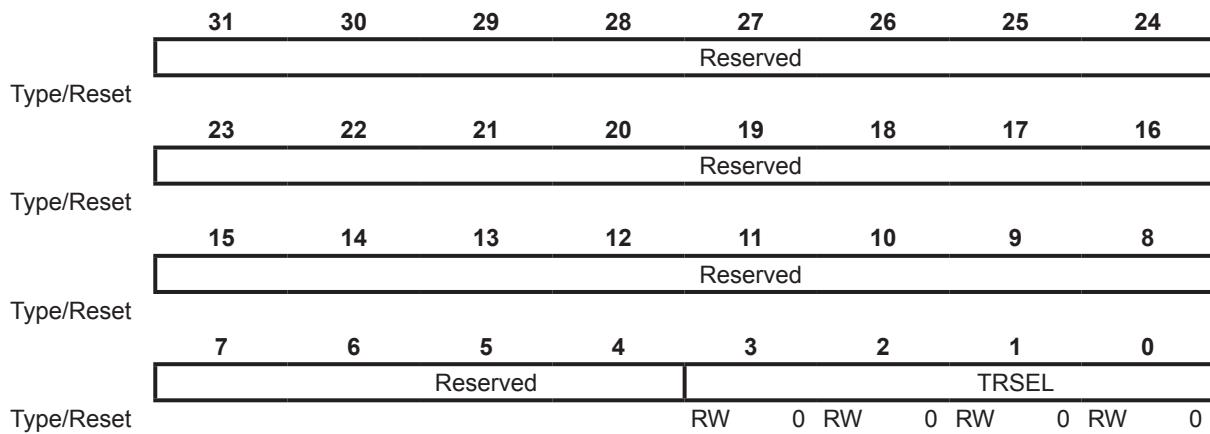
Bits	Field	Descriptions	
[10:8]	SMSEL	Slave Mode Selection	
		<b>SMSEL [2:0]</b>	<b>Mode</b>
		000	Disable mode
		001	Quadrature Decoder mode 1
		010	Quadrature Decoder mode 2
		011	Quadrature Decoder mode 3
		100	Restart Mode
		101	Pause Mode
		110	Trigger Mode
		111	STIED
[0]	TSE	Timer Synchronization Enable	
		0: No action 1: Master timer (current timer) will generate a delay to synchronize its slave timer through the MTO signal.	

## Timer Trigger Configuration Register – TRCFR

This register specifies the trigger source selection of GPTM.

Offset: 0x008

Reset value: 0x0000\_0000



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"> <li>0000: Software Trigger by setting the UEVG bit</li> <li>0001: Filtered input of channel 0 (TI0S0)</li> <li>0010: Filtered input of channel 1 (TI1S1)</li> <li>0011: Reserved</li> <li>1000: Channel 0 Edge Detector (TI0BED)</li> <li>1001: Internal Timing Module Trigger 0 (ITI0)</li> <li>1010: Internal Timing Module Trigger 1 (ITI1)</li> <li>1011: Internal Timing Module Trigger 2 (ITI2)</li> <li>Others: Default 0</li> </ul> <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**

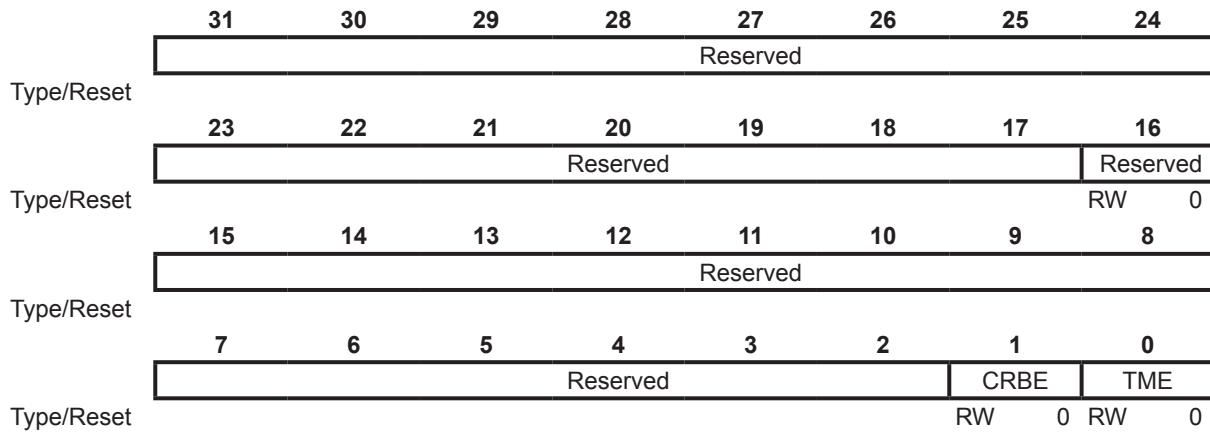
Slave Timing Module	ITI0	ITI1	ITI2
GPTM	X	MCTM	X

## Timer Counter Register – CTR

This register specifies the timer enable bit (TME) and 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 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
Type/Reset	TI0SRC				Reserved			
	RW	0						
Type/Reset	23	22	21	20	19	18	17	16
			Reserved			CH0PSC		CH0CCS
					RW	0	RW	0
Type/Reset	15	14	13	12	11	10	9	8
					Reserved			
Type/Reset	7	6	5	4	3	2	1	0
			Reserved				TI0F	
					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.

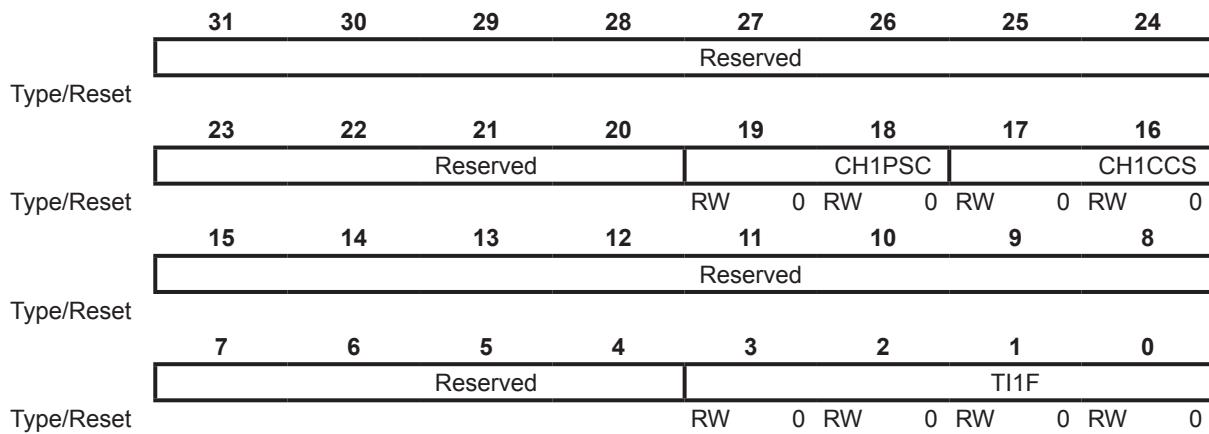
Bits	Field	Descriptions
[3:0]	TIOF	<p>Channel 0 Input Source TIO Filter Setting</p> <p>These bits define the frequency divided ratio used to sample the TIO signal. The Digital filter in the GPTM is an N-event counter where N is defined as how many valid transitions are necessary to output a filtered signal.</p> <ul style="list-style-type: none"> <li>0000: No filter, the sampling clock is <math>f_{SYSTEM}</math>.</li> <li>0001: <math>f_{SAMPLING} = f_{CLKIN}</math>, <math>N = 2</math></li> <li>0010: <math>f_{SAMPLING} = f_{CLKIN}</math>, <math>N = 4</math></li> <li>0011: <math>f_{SAMPLING} = f_{CLKIN}</math>, <math>N = 8</math></li> <li>0100: <math>f_{SAMPLING} = f_{DTS} / 2</math>, <math>N = 6</math></li> <li>0101: <math>f_{SAMPLING} = f_{DTS} / 2</math>, <math>N = 8</math></li> <li>0110: <math>f_{SAMPLING} = f_{DTS} / 4</math>, <math>N = 6</math></li> <li>0111: <math>f_{SAMPLING} = f_{DTS} / 4</math>, <math>N = 8</math></li> <li>1000: <math>f_{SAMPLING} = f_{DTS} / 8</math>, <math>N = 6</math></li> <li>1001: <math>f_{SAMPLING} = f_{DTS} / 8</math>, <math>N = 8</math></li> <li>1010: <math>f_{SAMPLING} = f_{DTS} / 16</math>, <math>N = 5</math></li> <li>1011: <math>f_{SAMPLING} = f_{DTS} / 16</math>, <math>N = 6</math></li> <li>1100: <math>f_{SAMPLING} = f_{DTS} / 16</math>, <math>N = 8</math></li> <li>1101: <math>f_{SAMPLING} = f_{DTS} / 32</math>, <math>N = 5</math></li> <li>1110: <math>f_{SAMPLING} = f_{DTS} / 32</math>, <math>N = 6</math></li> <li>1111: <math>f_{SAMPLING} = f_{DTS} / 32</math>, <math>N = 8</math></li> </ul>

## Channel 1 Input Configuration Register – CH1ICFR

This register specifies the channel 1 input mode configuration.

Offset: 0x024

Reset value: 0x0000\_0000



Bits	Field	Descriptions
[19:18]	CH1PSC	<p>Channel 1 Capture Input Source Prescaler Setting</p> <p>These bits define the effective events of the channel 1 capture input. Note that the prescaler is reset once the Channel 1 Capture/Compare Enable bit, CH1E, in the Channel Control register named CHCTR is cleared to 0.</p> <ul style="list-style-type: none"> <li>00: No prescaler, channel 1 capture input signal is chosen for each active event</li> <li>01: Channel 1 Capture input signal is chosen for every 2 events</li> <li>10: Channel 1 Capture input signal is chosen for every 4 events</li> <li>11: Channel 1 Capture input signal is chosen for every 8 events</li> </ul>
[17:16]	CH1CCS	<p>Channel 1 Capture/Compare Selection</p> <ul style="list-style-type: none"> <li>00: Channel 1 is configured as an output</li> <li>01: Channel 1 is configured as an input derived from the TI1 signal</li> <li>10: Channel 1 is configured as an input derived from the TI0 signal</li> <li>11: Channel 1 is configured as an input which comes from the TRCED signal derived from the Trigger Controller</li> </ul> <p>Note: The CH1CCS field can be accessed only when the CH1E bit is cleared to 0.</p>

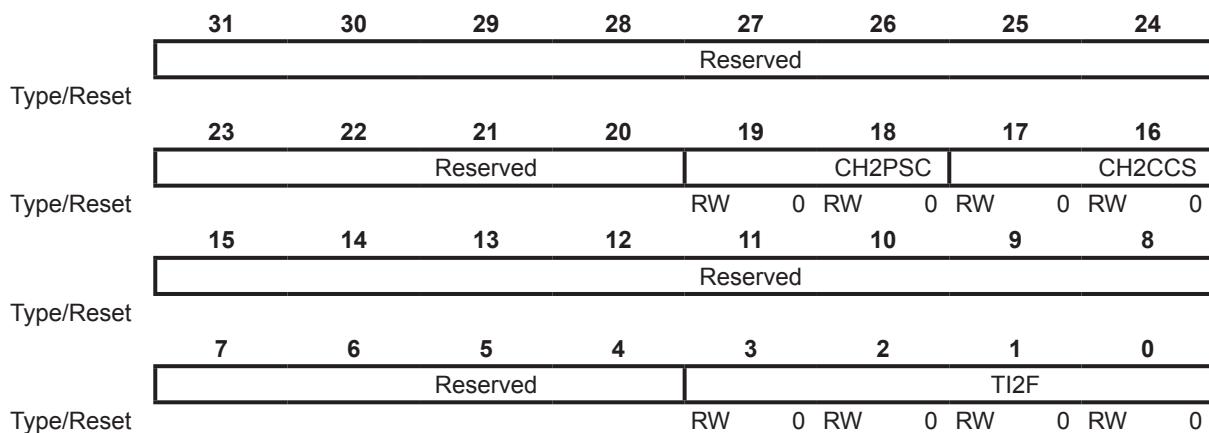
Bits	Field	Descriptions
[3:0]	TI1F	<p>Channel 1 Input Source TI1 Filter Setting</p> <p>These bits define the frequency divided ratio used to sample the TI1 signal. The Digital filter in the GPTM is an N-event counter where N is defined as how many valid transitions are necessary to output a filtered signal.</p> <ul style="list-style-type: none"> <li>0000: No filter, the sampling clock is <math>f_{SYSTEM}</math>.</li> <li>0001: <math>f_{SAMPLING} = f_{CLKIN}</math>, <math>N = 2</math></li> <li>0010: <math>f_{SAMPLING} = f_{CLKIN}</math>, <math>N = 4</math></li> <li>0011: <math>f_{SAMPLING} = f_{CLKIN}</math>, <math>N = 8</math></li> <li>0100: <math>f_{SAMPLING} = f_{DTS} / 2</math>, <math>N = 6</math></li> <li>0101: <math>f_{SAMPLING} = f_{DTS} / 2</math>, <math>N = 8</math></li> <li>0110: <math>f_{SAMPLING} = f_{DTS} / 4</math>, <math>N = 6</math></li> <li>0111: <math>f_{SAMPLING} = f_{DTS} / 4</math>, <math>N = 8</math></li> <li>1000: <math>f_{SAMPLING} = f_{DTS} / 8</math>, <math>N = 6</math></li> <li>1001: <math>f_{SAMPLING} = f_{DTS} / 8</math>, <math>N = 8</math></li> <li>1010: <math>f_{SAMPLING} = f_{DTS} / 16</math>, <math>N = 5</math></li> <li>1011: <math>f_{SAMPLING} = f_{DTS} / 16</math>, <math>N = 6</math></li> <li>1100: <math>f_{SAMPLING} = f_{DTS} / 16</math>, <math>N = 8</math></li> <li>1101: <math>f_{SAMPLING} = f_{DTS} / 32</math>, <math>N = 5</math></li> <li>1110: <math>f_{SAMPLING} = f_{DTS} / 32</math>, <math>N = 6</math></li> <li>1111: <math>f_{SAMPLING} = f_{DTS} / 32</math>, <math>N = 8</math></li> </ul>

## Channel 2 Input Configuration Register – CH2ICFR

This register specifies the channel 2 input mode configuration.

Offset: 0x028

Reset value: 0x0000\_0000



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"> <li>00: No prescaler, channel 2 capture input signal is chosen for each active event</li> <li>01: Channel 2 Capture input signal is chosen for every 2 events</li> <li>10: Channel 2 Capture input signal is chosen for every 4 events</li> <li>11: Channel 2 Capture input signal is chosen for every 8 events</li> </ul>
[17:16]	CH2CCS	<p>Channel 2 Capture/Compare Selection</p> <ul style="list-style-type: none"> <li>00: Channel 2 is configured as an output</li> <li>01: Channel 2 is configured as an input derived from the TI2 signal</li> <li>10: Channel 2 is configured as an input derived from the TI3 signal</li> <li>11: Channel 2 is configured as an input which comes from the TRCED signal derived from the Trigger Controller</li> </ul> <p>Note: The CH2CCS field can be accessed only when the CH2E bit is cleared to 0</p>

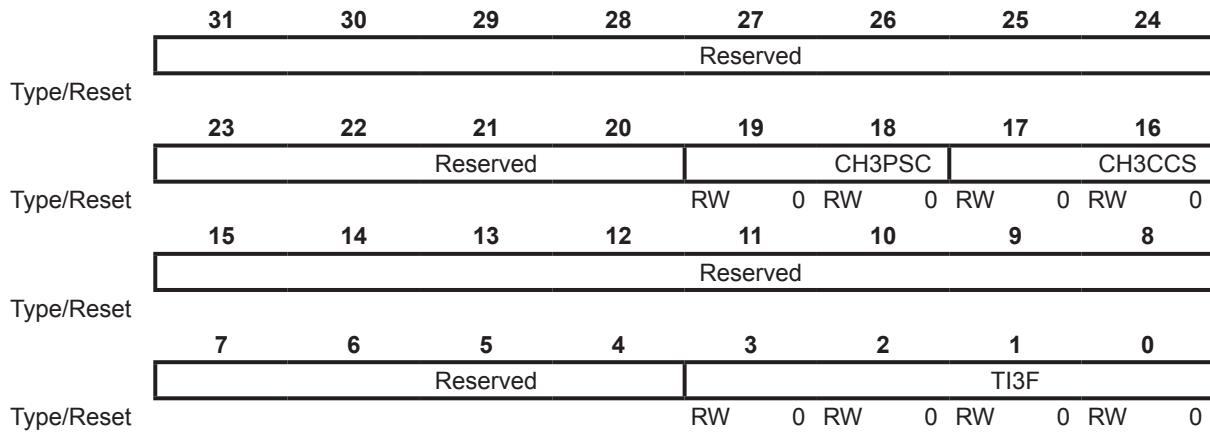
Bits	Field	Descriptions
[3:0]	TI2F	<p>Channel 2 Input Source TI2 Filter Setting</p> <p>These bits define the frequency divided ratio used to sample the TI2 signal. The Digital filter in the GPTM is an N-event counter where N is defined as how many valid transitions are necessary to output a filtered signal.</p> <ul style="list-style-type: none"> <li>0000: No filter, the sampling clock is <math>f_{SYSTEM}</math>.</li> <li>0001: <math>f_{SAMPLING} = f_{CLKIN}</math>, <math>N = 2</math></li> <li>0010: <math>f_{SAMPLING} = f_{CLKIN}</math>, <math>N = 4</math></li> <li>0011: <math>f_{SAMPLING} = f_{CLKIN}</math>, <math>N = 8</math></li> <li>0100: <math>f_{SAMPLING} = f_{DTS} / 2</math>, <math>N = 6</math></li> <li>0101: <math>f_{SAMPLING} = f_{DTS} / 2</math>, <math>N = 8</math></li> <li>0110: <math>f_{SAMPLING} = f_{DTS} / 4</math>, <math>N = 6</math></li> <li>0111: <math>f_{SAMPLING} = f_{DTS} / 4</math>, <math>N = 8</math></li> <li>1000: <math>f_{SAMPLING} = f_{DTS} / 8</math>, <math>N = 6</math></li> <li>1001: <math>f_{SAMPLING} = f_{DTS} / 8</math>, <math>N = 8</math></li> <li>1010: <math>f_{SAMPLING} = f_{DTS} / 16</math>, <math>N = 5</math></li> <li>1011: <math>f_{SAMPLING} = f_{DTS} / 16</math>, <math>N = 6</math></li> <li>1100: <math>f_{SAMPLING} = f_{DTS} / 16</math>, <math>N = 8</math></li> <li>1101: <math>f_{SAMPLING} = f_{DTS} / 32</math>, <math>N = 5</math></li> <li>1110: <math>f_{SAMPLING} = f_{DTS} / 32</math>, <math>N = 6</math></li> <li>1111: <math>f_{SAMPLING} = f_{DTS} / 32</math>, <math>N = 8</math></li> </ul>

## Channel 3 Input Configuration Register – CH3ICFR

This register specifies the channel 3 input mode configuration.

Offset: 0x02C

Reset value: 0x0000\_0000



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"> <li>00: No prescaler, channel 3 capture input signal is chosen for each active event</li> <li>01: Channel 3 Capture input signal is chosen for every 2 events</li> <li>10: Channel 3 Capture input signal is chosen for every 4 events</li> <li>11: Channel 3 Capture input signal is chosen for every 8 events</li> </ul>
[17:16]	CH3CCS	<p>Channel 3 Capture/Compare Selection</p> <ul style="list-style-type: none"> <li>00: Channel 3 is configured as an output</li> <li>01: Channel 3 is configured as an input derived from the TI3 signal</li> <li>10: Channel 3 is configured as an input derived from the TI2 signal</li> <li>11: Channel 3 is configured as an input which comes from the TRCED signal derived from the Trigger Controller</li> </ul> <p>Note: The CH3CCS field can be accessed only when the CH3E bit is cleared to 0</p>

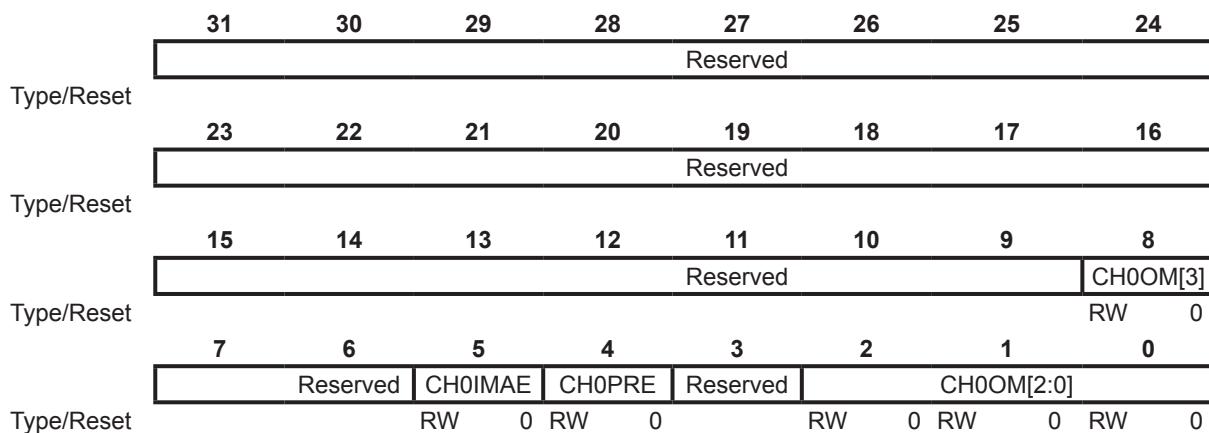
Bits	Field	Descriptions
[3:0]	TI3F	<p>Channel 3 Input Source TI3 Filter Setting</p> <p>These bits define the frequency divided ratio used to sample the TI3 signal. The Digital filter in the GPTM is an N-event counter where N is defined as how many valid transitions are necessary to output a filtered signal.</p> <ul style="list-style-type: none"> <li>0000: No filter, the sampling clock is <math>f_{SYSTEM}</math>.</li> <li>0001: <math>f_{SAMPLING} = f_{CLKIN}</math>, <math>N = 2</math></li> <li>0010: <math>f_{SAMPLING} = f_{CLKIN}</math>, <math>N = 4</math></li> <li>0011: <math>f_{SAMPLING} = f_{CLKIN}</math>, <math>N = 8</math></li> <li>0100: <math>f_{SAMPLING} = f_{DTS} / 2</math>, <math>N = 6</math></li> <li>0101: <math>f_{SAMPLING} = f_{DTS} / 2</math>, <math>N = 8</math></li> <li>0110: <math>f_{SAMPLING} = f_{DTS} / 4</math>, <math>N = 6</math></li> <li>0111: <math>f_{SAMPLING} = f_{DTS} / 4</math>, <math>N = 8</math></li> <li>1000: <math>f_{SAMPLING} = f_{DTS} / 8</math>, <math>N = 6</math></li> <li>1001: <math>f_{SAMPLING} = f_{DTS} / 8</math>, <math>N = 8</math></li> <li>1010: <math>f_{SAMPLING} = f_{DTS} / 16</math>, <math>N = 5</math></li> <li>1011: <math>f_{SAMPLING} = f_{DTS} / 16</math>, <math>N = 6</math></li> <li>1100: <math>f_{SAMPLING} = f_{DTS} / 16</math>, <math>N = 8</math></li> <li>1101: <math>f_{SAMPLING} = f_{DTS} / 32</math>, <math>N = 5</math></li> <li>1110: <math>f_{SAMPLING} = f_{DTS} / 32</math>, <math>N = 6</math></li> <li>1111: <math>f_{SAMPLING} = f_{DTS} / 32</math>, <math>N = 8</math></li> </ul>

## 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> <ul style="list-style-type: none"> <li>0000: No Change</li> <li>0001: Output 0 on compare match</li> <li>0010: Output 1 on compare match</li> <li>0011: Output toggles on compare match</li> <li>0100: Force inactive – CH0OREF is forced to 0</li> <li>0101: Force active – CH0OREF is forced to 1</li> <li>0110: PWM mode 1 <ul style="list-style-type: none"> <li>- During up-counting, channel 0 has an active level when CNTR &lt; CH0CCR or otherwise has an inactive level.</li> <li>- During down-counting, channel 0 has an inactive level when CNTR &gt; CH0CCR or otherwise has an active level.</li> </ul> </li> <li>0111: PWM mode 2 <ul style="list-style-type: none"> <li>- During up-counting, channel 0 has an inactive level when CNTR &lt; CH0CCR or otherwise has an active level.</li> <li>- During down-counting, channel 0 has an active level when CNTR &gt; CH0CCR or otherwise has an inactive level.</li> </ul> </li> <li>1110: Asymmetric PWM mode 1 <ul style="list-style-type: none"> <li>- During up-counting, channel 0 has an active level when CNTR &lt; CH0CCR or otherwise has an inactive level.</li> <li>- During down-counting, channel 0 has an inactive level when CNTR &gt; CH0ACR or otherwise has an active level.</li> </ul> </li> <li>1111: Asymmetric PWM mode 2 <ul style="list-style-type: none"> <li>- During up-counting, channel 0 has an inactive level when CNTR &lt; CH0CCR or otherwise has an active level.</li> <li>- During down-counting, channel 0 has an active level when CNTR &gt; CH0ACR or otherwise has an inactive level</li> </ul> </li> </ul>

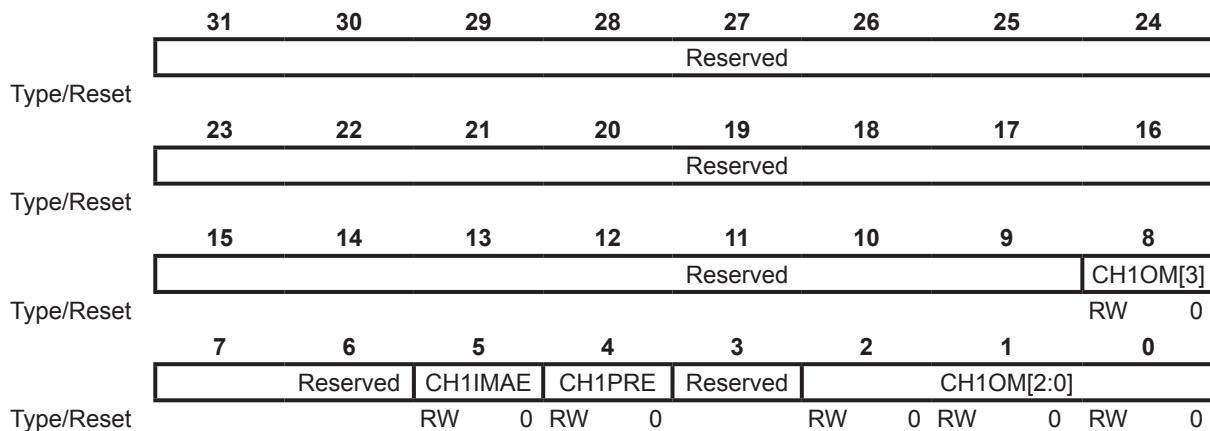
Note: When channel 0 is used as asymmetric PWM output mode, the Counter Mode Selection bit in Counter Configuration Register must be configured as Center-aligned Counting mode (CMSEL = 01/02/03)

## 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> <ul style="list-style-type: none"> <li>0000: No Change</li> <li>0001: Output 0 on compare match</li> <li>0010: Output 1 on compare match</li> <li>0011: Output toggles on compare match</li> <li>0100: Force inactive – CH1OREF is forced to 0</li> <li>0101: Force active – CH1OREF is forced to 1</li> <li>0110: PWM mode 1 <ul style="list-style-type: none"> <li>- During up-counting, channel 1 has an active level when CNTR &lt; CH1CCR or otherwise has an inactive level.</li> <li>- During down-counting, channel 1 has an inactive level when CNTR &gt; CH1CCR or otherwise has an active level.</li> </ul> </li> <li>0111: PWM mode 2 <ul style="list-style-type: none"> <li>- During up-counting, channel 1 has an inactive level when CNTR &lt; CH1CCR or otherwise has an active level.</li> <li>- During down-counting, channel 1 has an active level when CNTR &gt; CH1CCR or otherwise has an inactive level.</li> </ul> </li> <li>1110: Asymmetric PWM mode 1 <ul style="list-style-type: none"> <li>- During up-counting, channel 1 has an active level when CNTR &lt; CH1CCR or otherwise has an inactive level.</li> <li>- During down-counting, channel 1 has an inactive level when CNTR &gt; CH1ACR or otherwise has an active level.</li> </ul> </li> <li>1111: Asymmetric PWM mode 2 <ul style="list-style-type: none"> <li>- During up-counting, channel 1 has an inactive level when CNTR &lt; CH1CCR or otherwise has an active level.</li> <li>- During down-counting, channel 1 has an active level when CNTR &gt; CH1ACR or otherwise has an inactive level</li> </ul> </li> </ul>

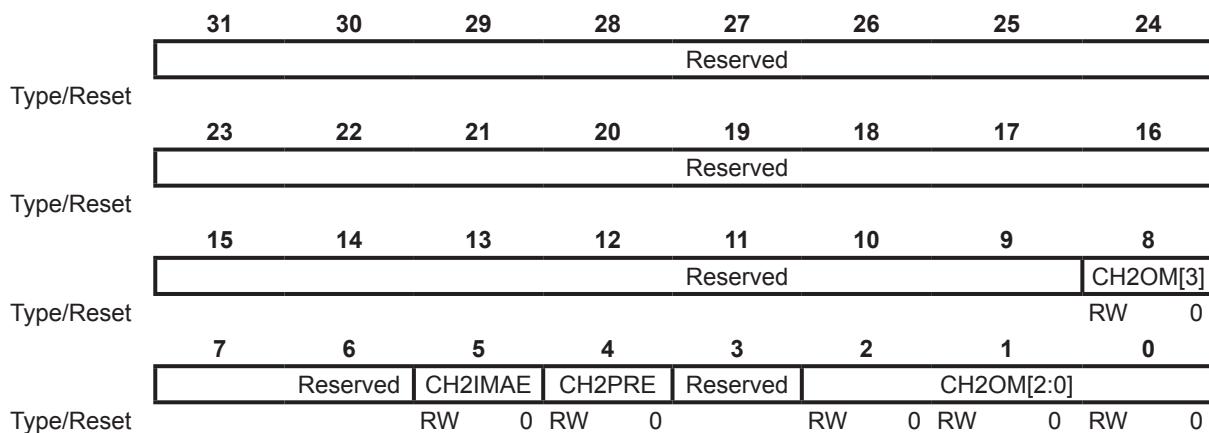
Note: When channel 1 is used as asymmetric PWM output mode, the Counter Mode Selection bit in Counter Configuration Register must be configured as Center-aligned Counting mode (CMSEL = 01/02/03)

## 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> <ul style="list-style-type: none"> <li>0000: No Change</li> <li>0001: Output 0 on compare match</li> <li>0010: Output 1 on compare match</li> <li>0011: Output toggles on compare match</li> <li>0100: Force inactive – CH2OREF is forced to 0</li> <li>0101: Force active – CH2OREF is forced to 1</li> <li>0110: PWM mode 1                     <ul style="list-style-type: none"> <li>- During up-counting, channel 2 has an active level when CNTR &lt; CH2CCR or otherwise has an inactive level.</li> <li>- During down-counting, channel 2 has an inactive level when CNTR &gt; CH2CCR or otherwise has an active level.</li> </ul> </li> <li>0111: PWM mode 2                     <ul style="list-style-type: none"> <li>- During up-counting, channel 2 has an inactive level when CNTR &lt; CH2CCR or otherwise has an active level.</li> <li>- During down-counting, channel 2 has an active level when CNTR &gt; CH2CCR or otherwise has an inactive level.</li> </ul> </li> <li>1110: Asymmetric PWM mode 1                     <ul style="list-style-type: none"> <li>- During up-counting, channel 2 has an active level when CNTR &lt; CH2CCR or otherwise has an inactive level.</li> <li>- During down-counting, channel 2 has an inactive level when CNTR &gt; CH2ACR or otherwise has an active level.</li> </ul> </li> <li>1111: Asymmetric PWM mode 2                     <ul style="list-style-type: none"> <li>- During up-counting, channel 2 has an inactive level when CNTR &lt; CH2CCR or otherwise has an active level.</li> <li>- During down-counting, channel 2 has an active level when CNTR &gt; CH2ACR or otherwise has an inactive level</li> </ul> </li> </ul>

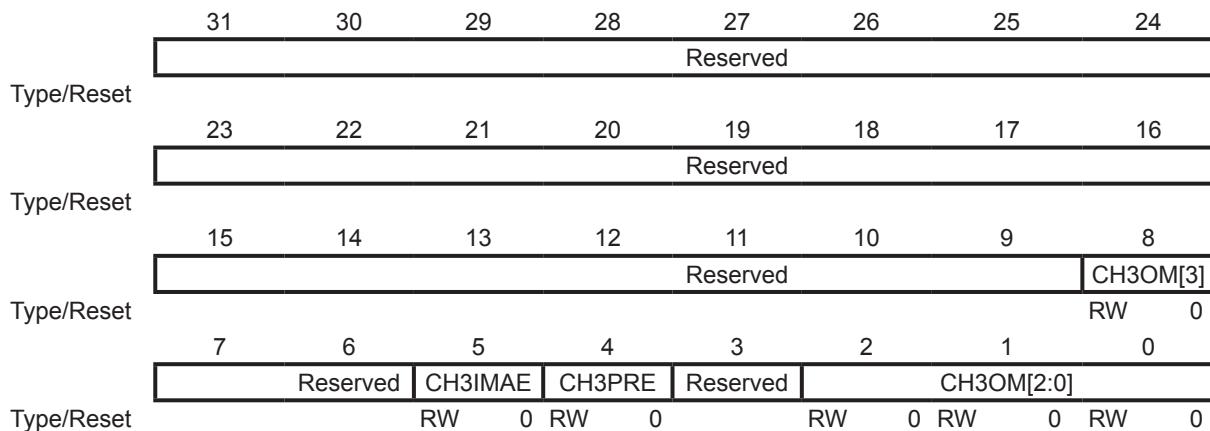
Note: When channel 2 is used as asymmetric PWM output mode, the Counter Mode Selection bit in Counter Configuration Register must be configured as Center-aligned Counting mode (CMSEL = 01/02/03)

## 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> <ul style="list-style-type: none"> <li>0000: No Change</li> <li>0001: Output 0 on compare match</li> <li>0010: Output 1 on compare match</li> <li>0011: Output toggles on compare match</li> <li>0100: Force inactive – CH3OREF is forced to 0</li> <li>0101: Force active – CH3OREF is forced to 1</li> <li>0110: PWM mode 1                     <ul style="list-style-type: none"> <li>- During up-counting, channel 3 has an active level when CNTR &lt; CH3CCR or otherwise has an inactive level.</li> <li>- During down-counting, channel 3 has an inactive level when CNTR &gt; CH3CCR or otherwise has an active level.</li> </ul> </li> <li>0111: PWM mode 2                     <ul style="list-style-type: none"> <li>- During up-counting, channel 3 has an inactive level when CNTR &lt; CH3CCR or otherwise has an active level.</li> <li>- During down-counting, channel 3 has an active level when CNTR &gt; CH3CCR or otherwise has an inactive level</li> </ul> </li> <li>1110: Asymmetric PWM mode 1                     <ul style="list-style-type: none"> <li>- During up-counting, channel 3 has an active level when CNTR &lt; CH3CCR or otherwise has an inactive level.</li> <li>- During down-counting, channel 3 has an inactive level when CNTR &gt; CH3ACR or otherwise has an active level.</li> </ul> </li> <li>1111: Asymmetric PWM mode 2                     <ul style="list-style-type: none"> <li>- During up-counting, channel 3 has an inactive level when CNTR &lt; CH3CCR or otherwise has an active level.</li> <li>- During down-counting, channel 3 has an active level when CNTR &gt; CH3ACR or otherwise has an inactive level</li> </ul> </li> </ul>

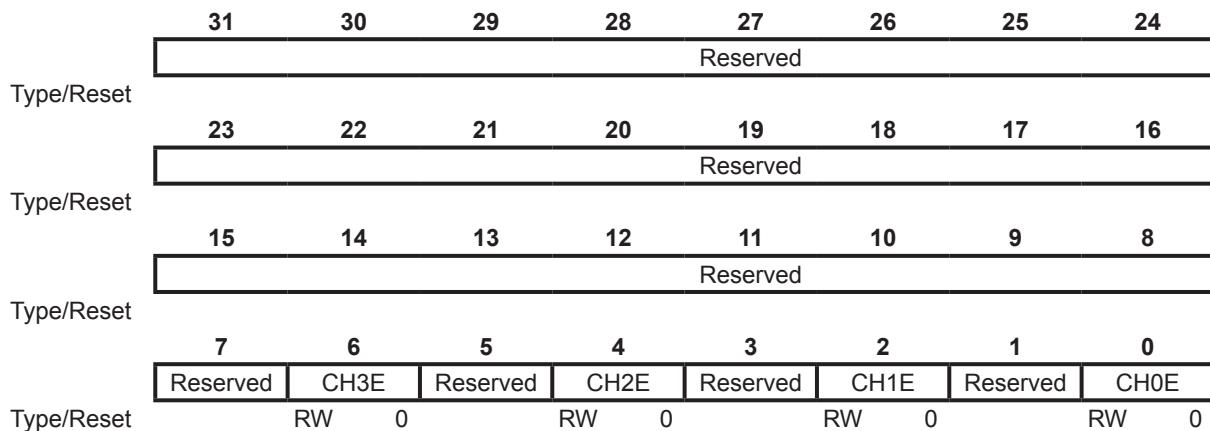
Note: When channel 3 is used as asymmetric PWM output mode, the Counter Mode Selection bit in Counter Configuration Register must be configured as Center-aligned Counting mode (CMSEL = 01/02/03)

## Channel Control Register – CHCTR

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

Offset: 0x050

Reset value: 0x0000\_0000



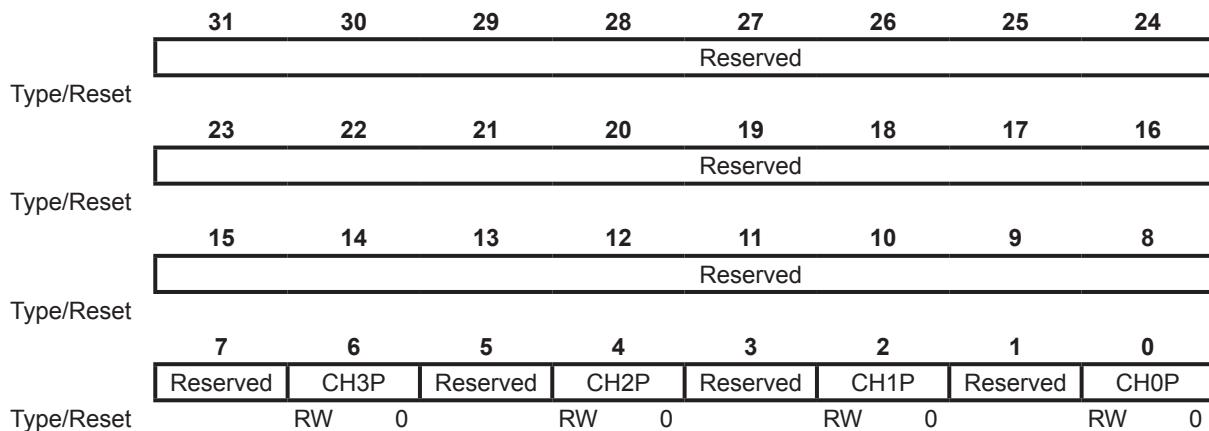
Bits	Field	Descriptions
[6]	CH3E	Channel 3 Capture/Compare Enable - Channel 3 is configured as an input (CH3CCS = 0x01/0x02/0x03) 0: Input Capture Mode disabled 1: Input Capture Mode enabled - Channel 3 is configured as an output (CH3CCS = 0x00) 0: Off – Channel 3 output signal CH3O is not active 1: On – Channel 3 output signal CH3O generated on the corresponding output pin
[4]	CH2E	Channel 2 Capture/Compare Enable - Channel 2 is configured as an input (CH2CCS = 0x01/0x02/0x03) 0: Input Capture Mode disabled 1: Input Capture Mode enabled - Channel 2 is configured as an output (CH2CCS = 0x00) 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	Channel 1 Capture/Compare Enable - Channel 1 is configured as an input (CH1CCS = 0x01/0x02/0x03) 0: Input Capture Mode disabled 1: Input Capture Mode enabled - Channel 1 is configured as an output (CH1CCS = 0x00) 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	Channel 0 Capture/Compare Enable - Channel 0 is configured as an input (CH0CCS = 0x01/0x02/0x03) 0: Input Capture Mode disabled 1: Input Capture Mode enabled - Channel 0 is configured as an output (CH0CCS = 0x00) 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



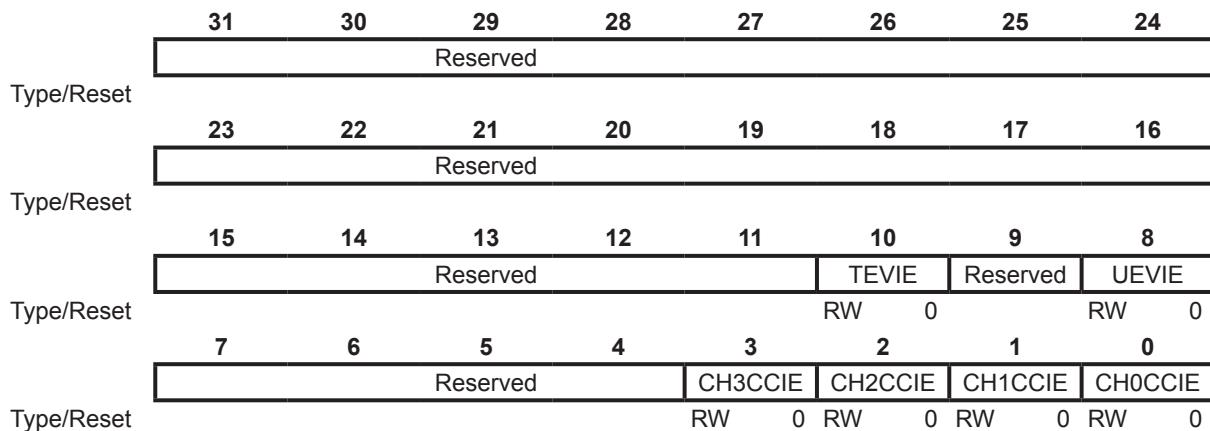
Bits	Field	Descriptions
[6]	CH3P	Channel 3 Capture/Compare Polarity - When Channel 3 is configured as an input 0: capture event occurs on a Channel 3 rising edge 1: capture event occurs on a Channel 3 falling edge - When Channel 3 is configured as an output (CH3CCS = 0x00) 0: Channel 3 Output active high 1: Channel 3 Output 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 active high 1: Channel 2 Output 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 active high 1: Channel 1 Output 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 active high 1: Channel 0 Output active low

## Timer Interrupt Control Register – DICTR

This register contains the timer interrupt enable control bits.

Offset: 0x074

Reset value: 0x0000\_0000



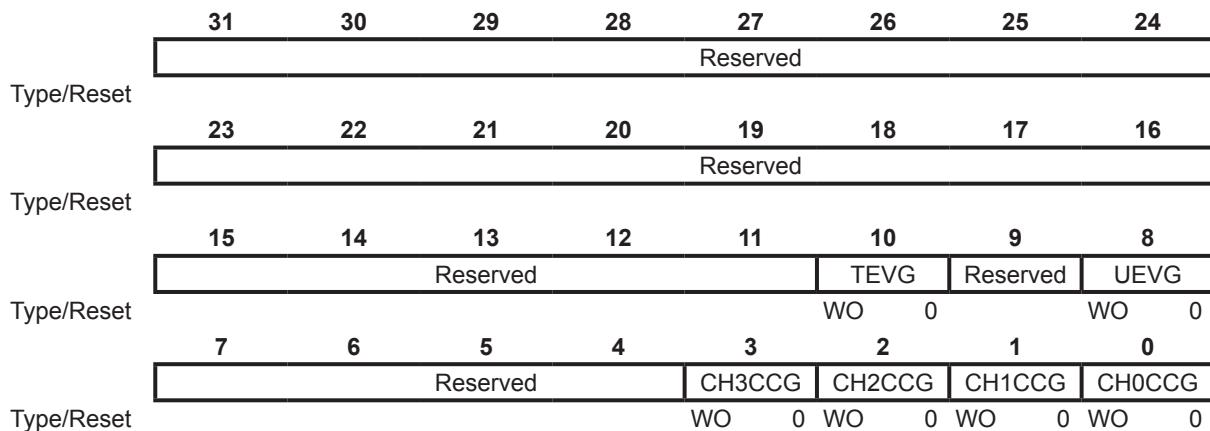
Bits	Field	Descriptions
[10]	TEVIE	Trigger event Interrupt Enable 0: Trigger event interrupt disabled 1: Trigger event interrupt enabled
[8]	UEVIE	Update event Interrupt Enable 0: Update event interrupt disabled 1: Update event interrupt enabled
[3]	CH3CCIE	Channel 3 Capture/Compare Interrupt Enable 0: Channel 3 interrupt disabled 1: Channel 3 interrupt enabled
[2]	CH2CCIE	Channel 2 Capture/Compare Interrupt Enable 0: Channel 2 interrupt disabled 1: Channel 2 interrupt enabled
[1]	CH1CCIE	Channel 1 Capture/Compare Interrupt Enable 0: Channel 1 interrupt disabled 1: Channel 1 interrupt enabled
[0]	CH0CCIE	Channel 0 Capture/Compare Interrupt Enable 0: Channel 0 interrupt disabled 1: Channel 0 interrupt 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>
[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>

<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>
[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					TEVIF	Reserved	UEVIF	
						RW 0	RW 0	RW 0	
	7	6	5	4	3	2	1	0	
Type/Reset	CH3OCF	CH2OCF	CH1OCF	CH0OCF	CH3CCIF	CH2CCIF	CH1CCIF	CH0CCIF	
	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	

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

<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>
[4]	CH0OCF	<p>Channel 0 Over-Capture Flag</p> <p>This flag is set by hardware and cleared by software.</p> <ul style="list-style-type: none"> <li>0: No over-capture event is detected</li> <li>1: Capture event occurs again when the CH0CCIFbit is already set and it is not yet cleared by software.</li> </ul>
[3]	CH3CCIF	<p>Channel 3 Capture/Compare Interrupt Flag</p> <ul style="list-style-type: none"> <li>- Channel 3 is configured as an output:</li> </ul> <p>0: No match event occurs</p> <p>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> <ul style="list-style-type: none"> <li>- Channel 3 is configured as an input:</li> </ul> <p>0: No input capture occurs</p> <p>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> <ul style="list-style-type: none"> <li>- Channel 2 is configured as an output:</li> </ul> <p>0: No match event occurs</p> <p>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> <ul style="list-style-type: none"> <li>- Channel 2 is configured as an input:</li> </ul> <p>0: No input capture occurs</p> <p>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> <ul style="list-style-type: none"> <li>- Channel 1 is configured as an output:</li> </ul> <p>0: No match event occurs</p> <p>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> <ul style="list-style-type: none"> <li>- Channel 1 is configured as an input:</li> </ul> <p>0: No input capture occurs</p> <p>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>

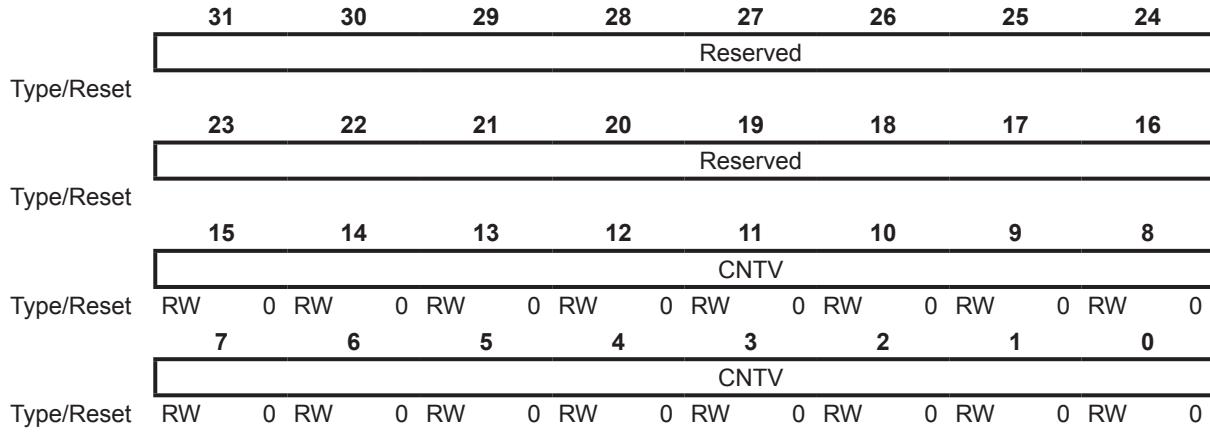
Bits	Field	Descriptions
[0]	CH0CCIF	<p>Channel 0 Capture/Compare Interrupt Flag</p> <p>- Channel 0 is configured as an output:</p> <p>0: No match event occurs</p> <p>1: The contents of the counter CNTR have matched the content of the CH0CCR register</p> <p>This flag is set by hardware when the counter value matches the CH0CCR value except in the center-aligned mode. It is cleared by software.</p> <p>- Channel 0 is configured as an input:</p> <p>0: No input capture occurs</p> <p>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 CH0CCR register.</p>

## Timer Counter Register – CNTR

This register stores the timer counter value.

Offset: 0x080

Reset value: 0x0000\_0000



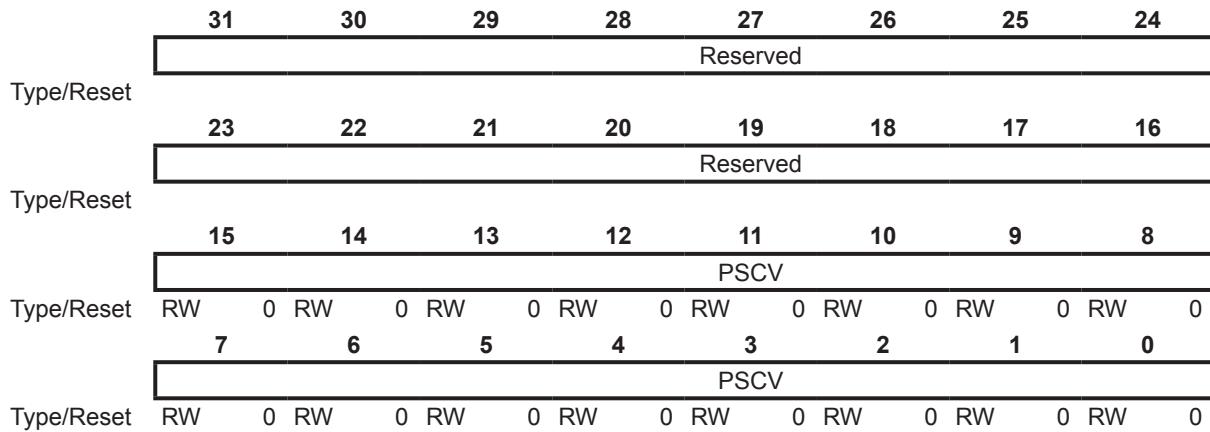
Bits	Field	Descriptions
[15:0]	CNTV	Counter Value.

## Timer Prescaler Register – PSCR

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

Offset: 0x084

Reset value: 0x0000\_0000



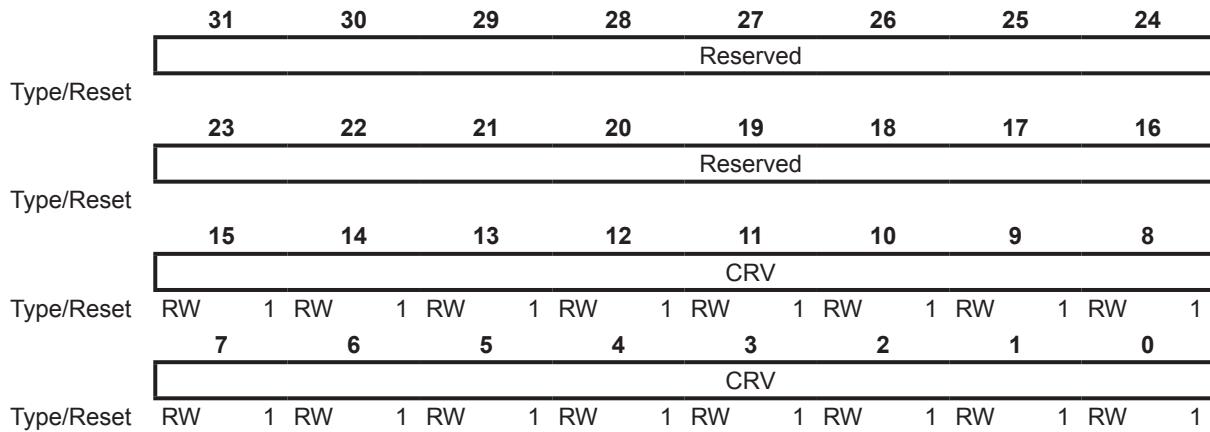
Bits	Field	Descriptions
[15:0]	PSCV	<p>Prescaler Value</p> <p>These bits are used to specify the prescaler value to generate the counter clock frequency <math>f_{CK\_CNT}</math>.</p> $f_{CK\_CNT} = \frac{f_{CK\_PSC}}{PSCV[15:0] + 1}$ , where the $f_{CK\_PSC}$ is the prescaler clock source.

## Timer Counter Reload Register – CRR

This register specifies the timer counter reload value.

Offset: 0x088

Reset value: 0x0000\_FFFF



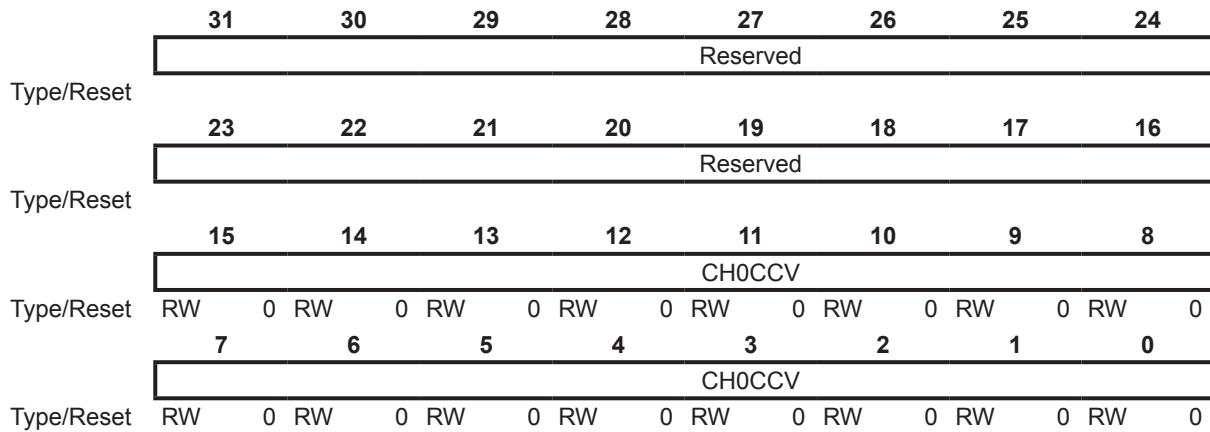
Bits	Field	Descriptions
[15:0]	CRV	Counter Reload Value The CRV is the reload value which is loaded into the actual counter register.

## Channel 0 Capture/Compare Register – CH0CCR

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

Offset: 0x090

Reset value: 0x0000\_0000



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

## Channel 1 Capture/Compare Register – CH1CCR

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

Offset: 0x094

Reset value: 0x0000\_0000

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

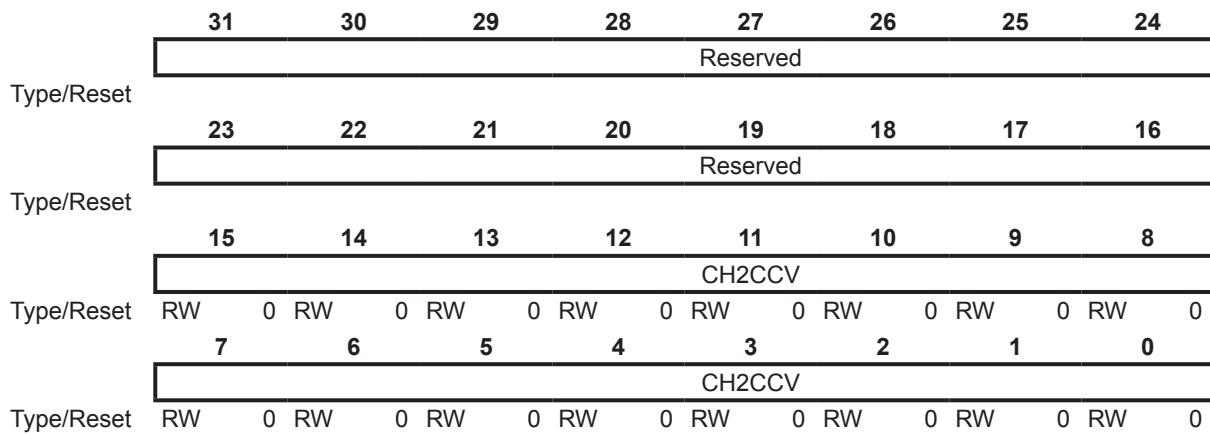
Bits	Field	Descriptions
[15:0]	CH1CCV	<p>Channel 1 Capture/Compare Value</p> <ul style="list-style-type: none"> <li>- When Channel 1 is configured as an output</li> <li>The CH1CCR value is compared with the counter value and the comparison result is used to trigger the CH1OREF output signal.</li> <li>-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

## Channel 2 Capture/Compare Register – CH2CCR

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

Offset: 0x098

Reset value: 0x0000\_0000



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

## Channel 3 Capture/Compare Register – CH3CCR

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

Offset: 0x09C

Reset value: 0x0000\_0000

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

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

## Channel 0 Asymmetric Compare Register – CH0ACR

This register specifies the timer channel 0 asymmetric compare value.

Offset: 0x0A0

Reset value: 0x0000\_0000

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

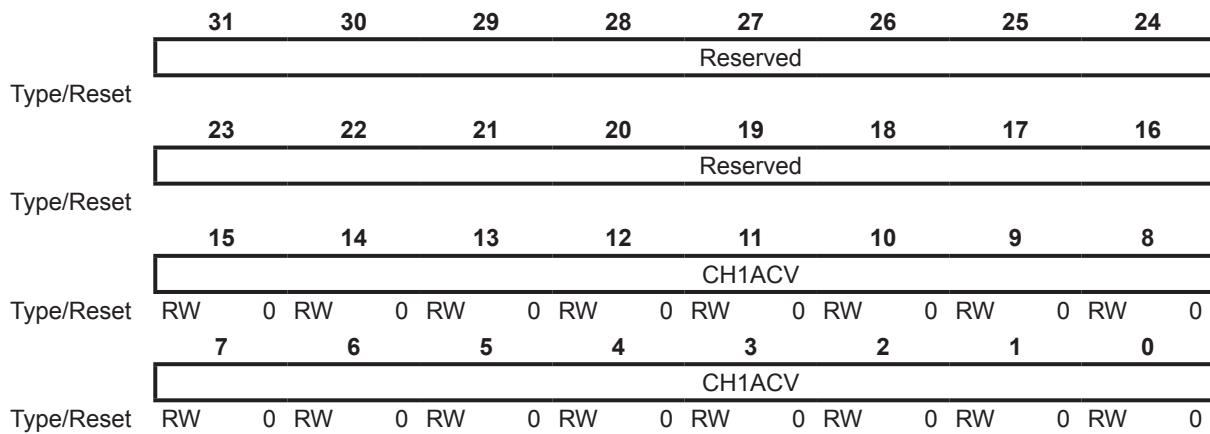
Bits	Field	Descriptions
[15:0]	CH0ACV	<p>Channel 0 Asymmetric Compare Value</p> <p>When channel 0 is configured as asymmetric PWM mode and the counter is counting down, the value written is this register will be compared to the counter.</p>

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

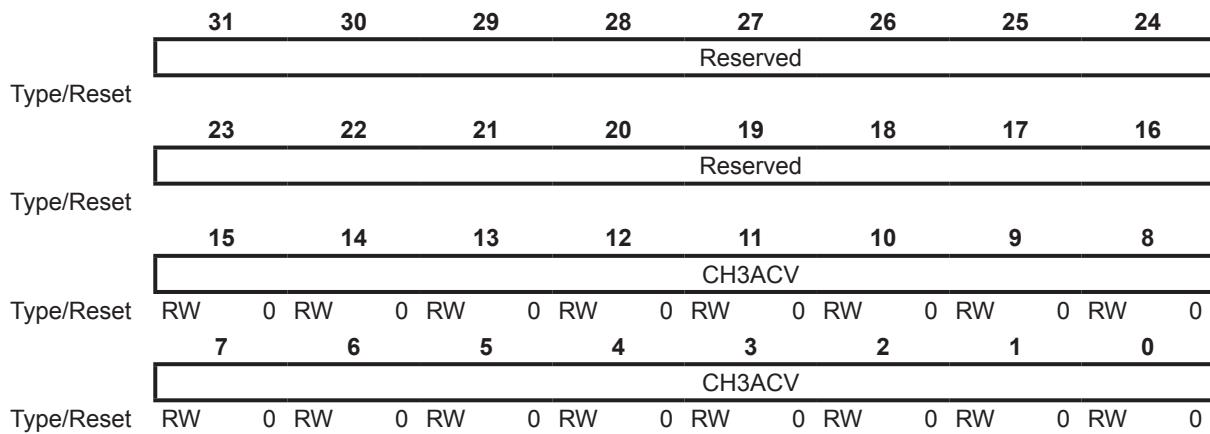
Bits	Field	Descriptions
[15:0]	CH2ACV	Channel 2 Asymmetric Compare Value When channel 2 is configured as asymmetric PWM mode and the counter is counting down, the value written to this register will be compared to the counter.

## 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 to this register will be compared to the counter.

# 14 Basic Function Timer (BFTM)

## Introduction

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

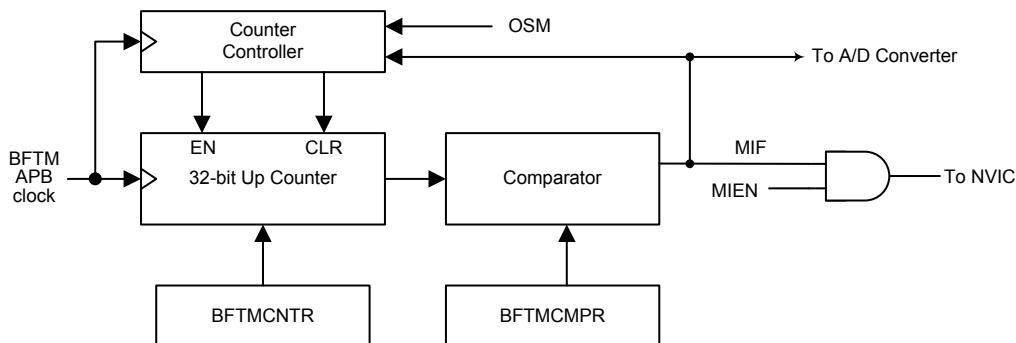


Figure 65. BFTM Block Diagram

## 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.

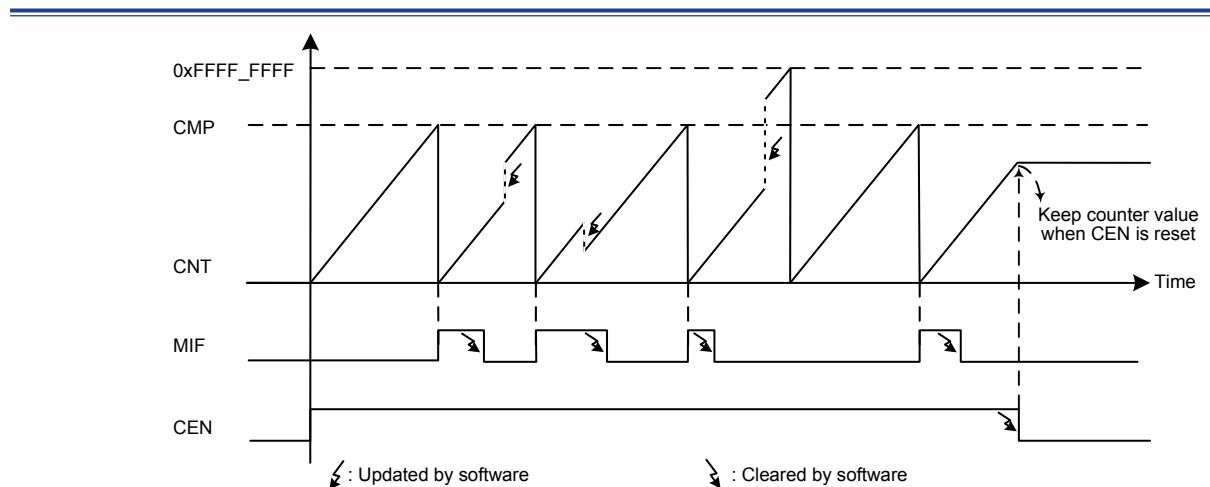


Figure 66. BFTM – Repetitive Mode

## 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.

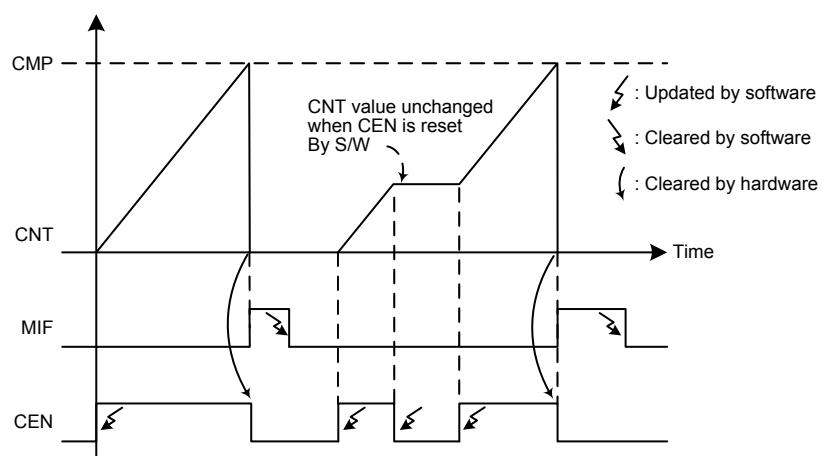


Figure 67. BFTM – One Shot Mode

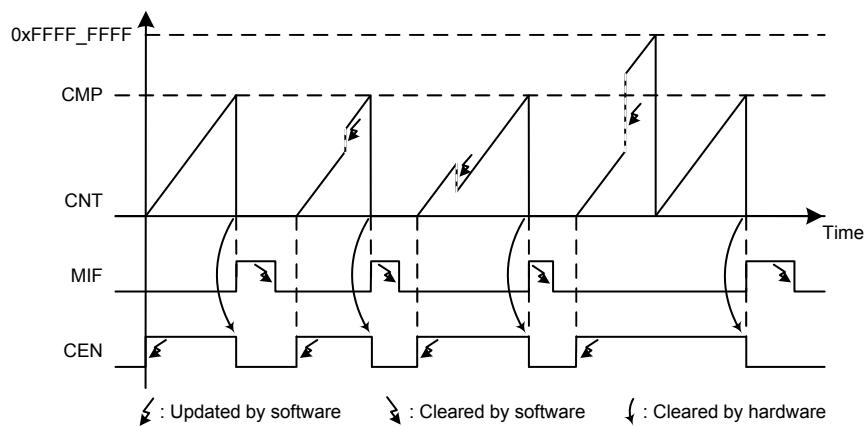


Figure 68. BFTM – One Shot Mode Counter Updating

## 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
BFTMCMPR	0x00C	BFTM Compare Value Register	0xFFFF_FFFF

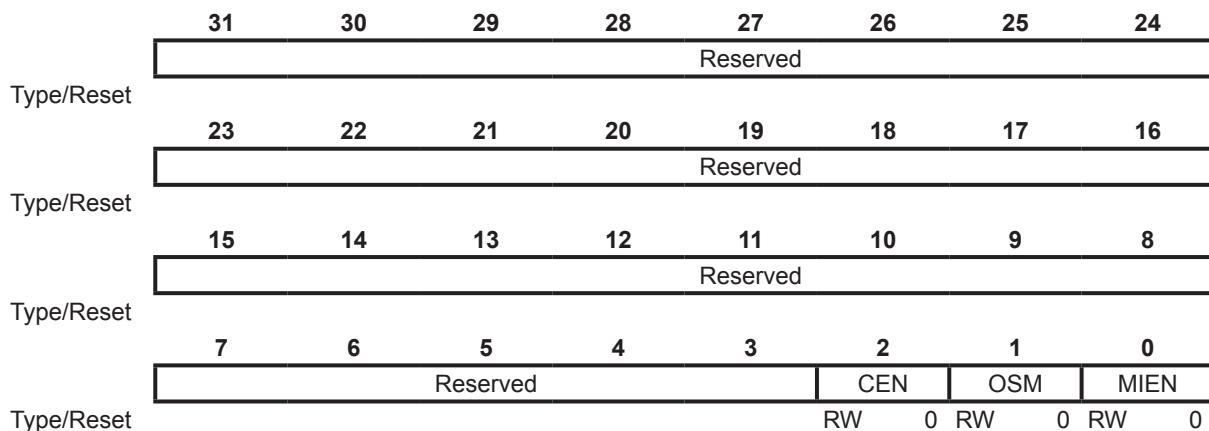
## Register Descriptions

### BFTM Control Register – BFTMCR

This register specifies the overall BFTM control bits.

Offset: 0x000

Reset value: 0x0000\_0000



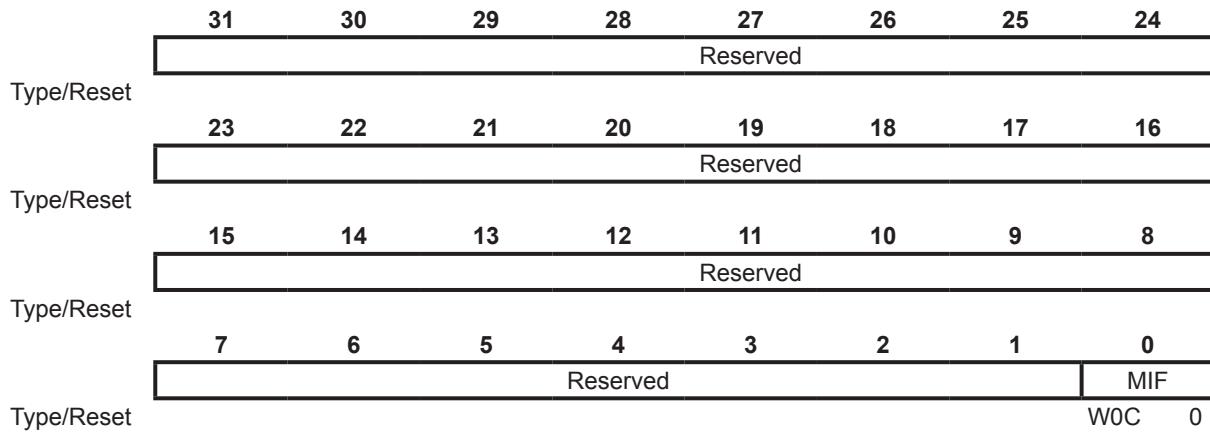
Bits	Field	Descriptions
[2]	CEN	BFTM Counter Enable Control 0: BFTM is disabled 1: BFTM is enabled  When this bit is set to 1, the BFTM counter will start to count. The counter will stop counting and the counter value will remain unchanged when the CEN bit is cleared to 0 by the application program regardless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

## BFTM Status Register – BFTMSR

This register specifies the BFTM status.

Offset: 0x004

Reset value: 0x0000\_0000



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
23 22 21 20 19 18 17 16								
CNT								
Type/Reset	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
7 6 5 4 3 2 1 0								
Type/Reset	RW	0	RW	0	RW	0	RW	0
CNT								

Bits	Field	Descriptions
[31:0]	CNT	BFTM Counter Register 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
23      22      21      20      19      18      17      16								
CMP								
Type/Reset	RW	1	RW	1	RW	1	RW	1
15      14      13      12      11      10      9      8								
CMP								
Type/Reset	RW	1	RW	1	RW	1	RW	1
7      6      5      4      3      2      1      0								
CMP								
Type/Reset	RW	1	RW	1	RW	1	RW	1

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

# 15 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.

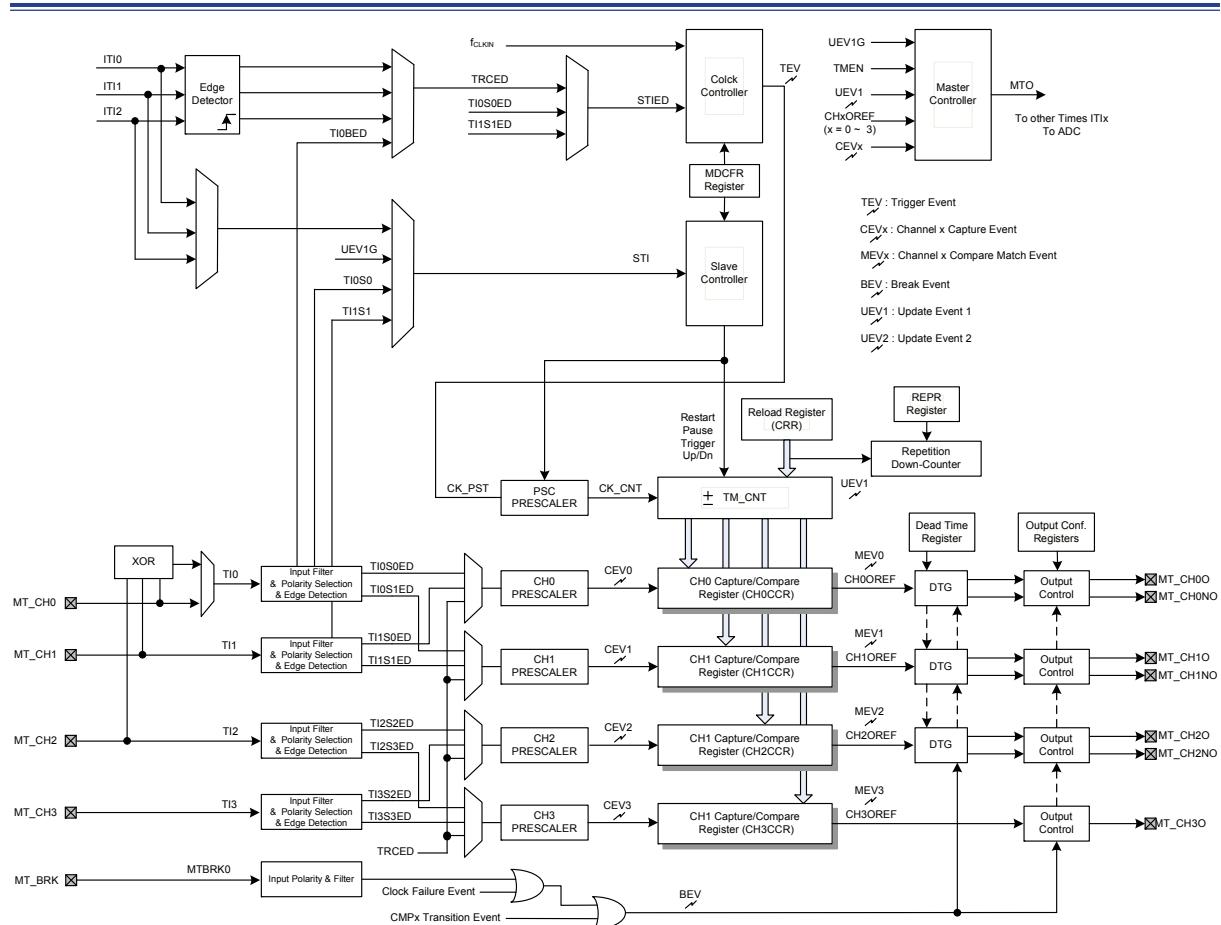


Figure 69. MCTM Block Diagram

## 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 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

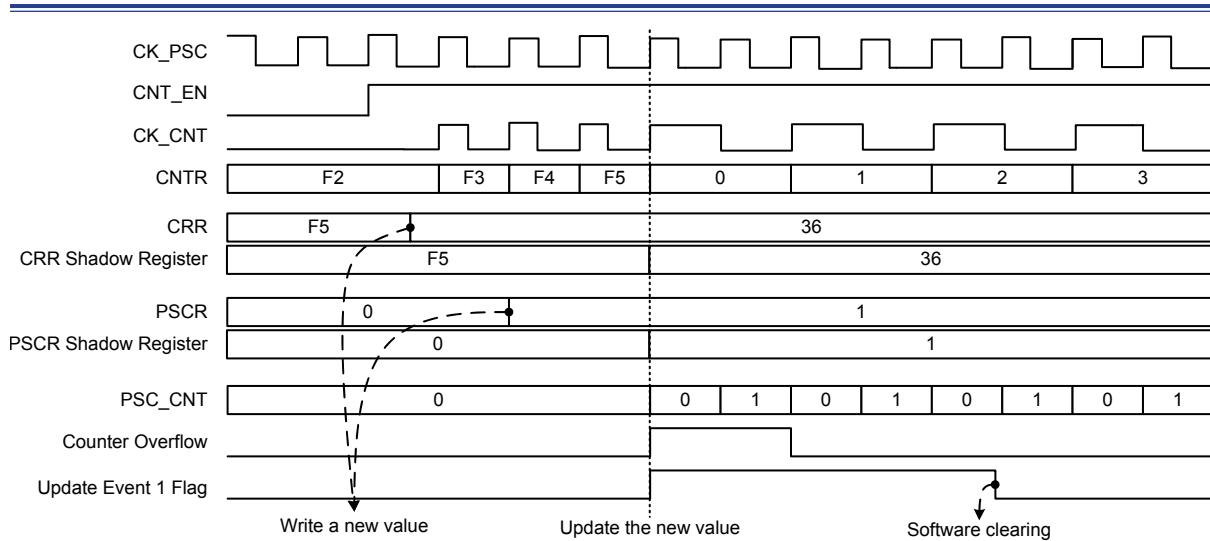
## Functional Descriptions

### 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.

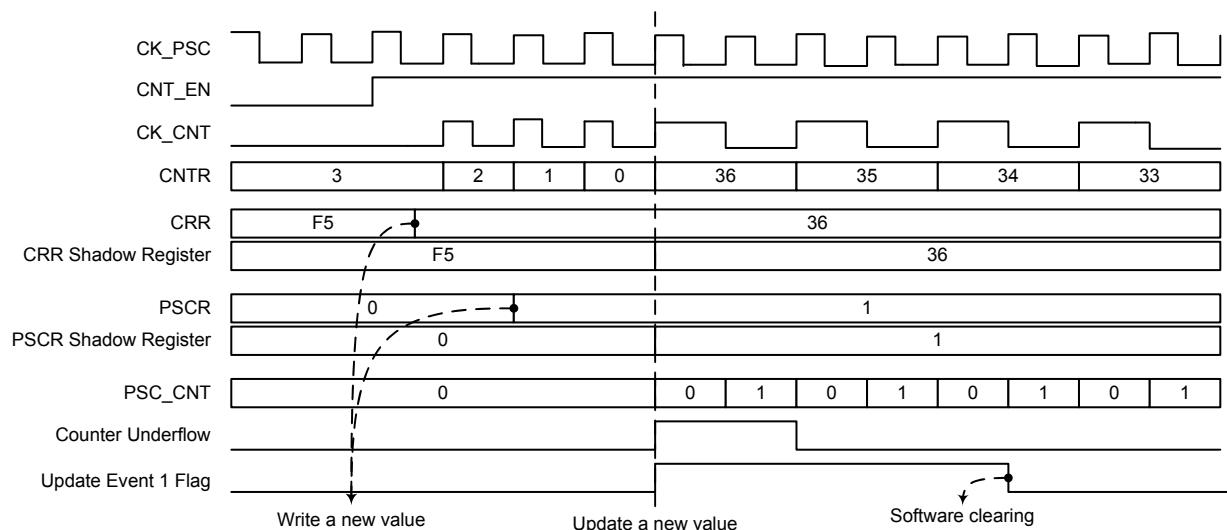


**Figure 70. Up-counting Example**

### 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.



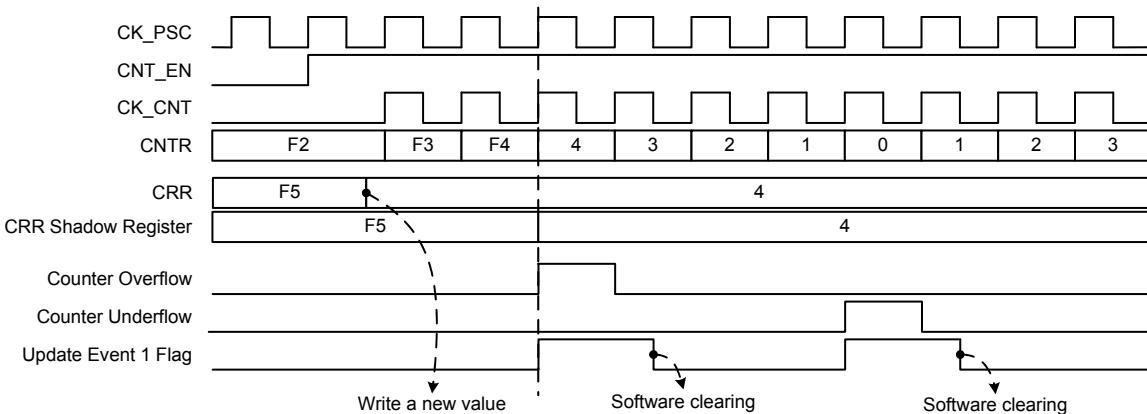
**Figure 71. Down-counting Example**

### 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 UEV1G 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.

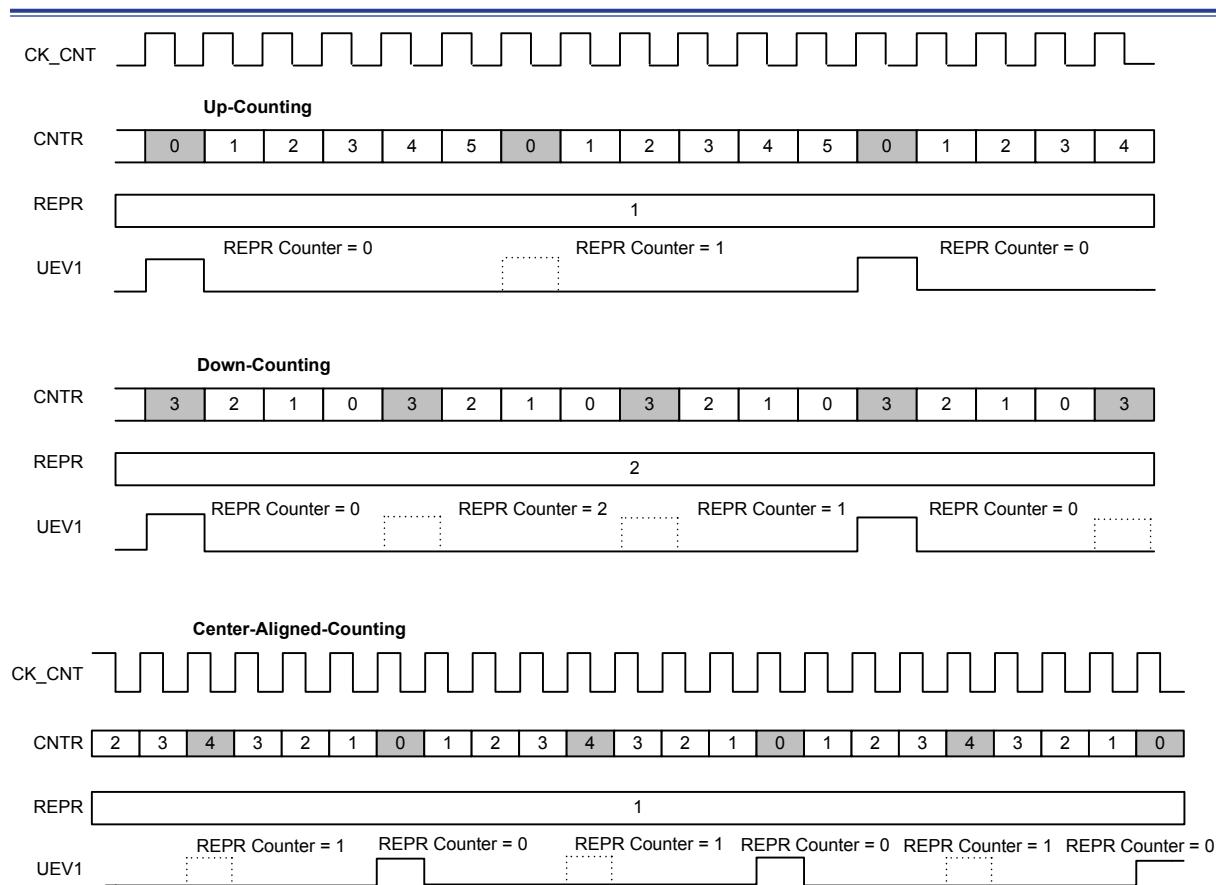


**Figure 72. Center-aligned Counting Example**

### 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 73. Update Event 1 Dependent Repetition Mechanism Example**

## Clock Controller

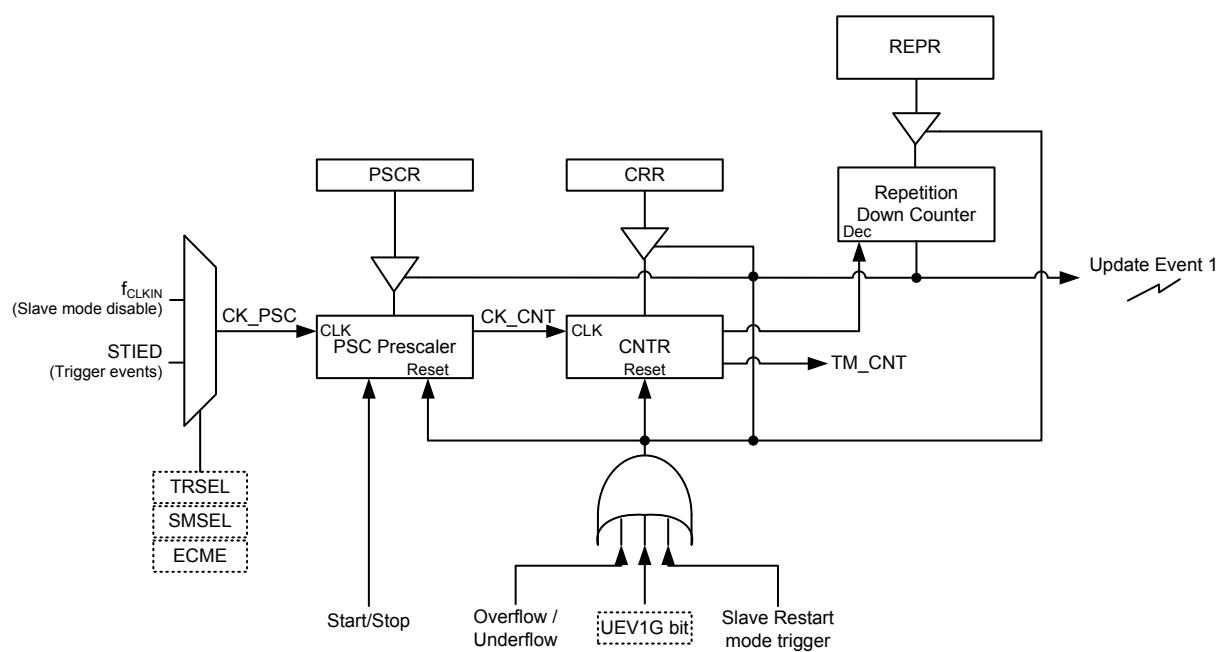
The following describes the Timer Module clock controller which determines the internal prescaler counter clock source.

- 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 UEV1G bit as the trigger source, then when the SMSEL field is set to 0x7, the counter will be updated instead of counting.

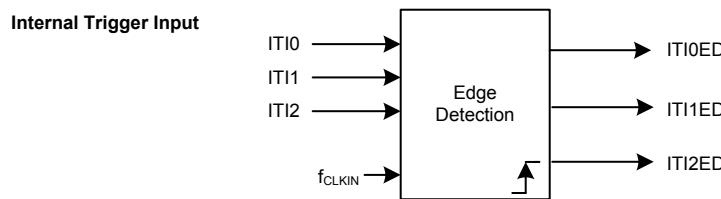


**Figure 74. MCTM Clock Selection Source**

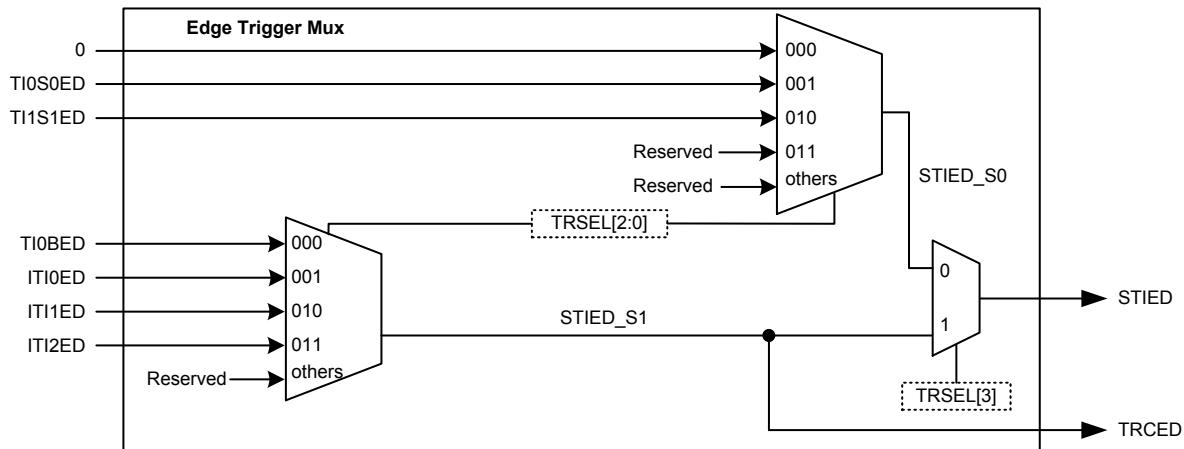
## Trigger Controller

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.

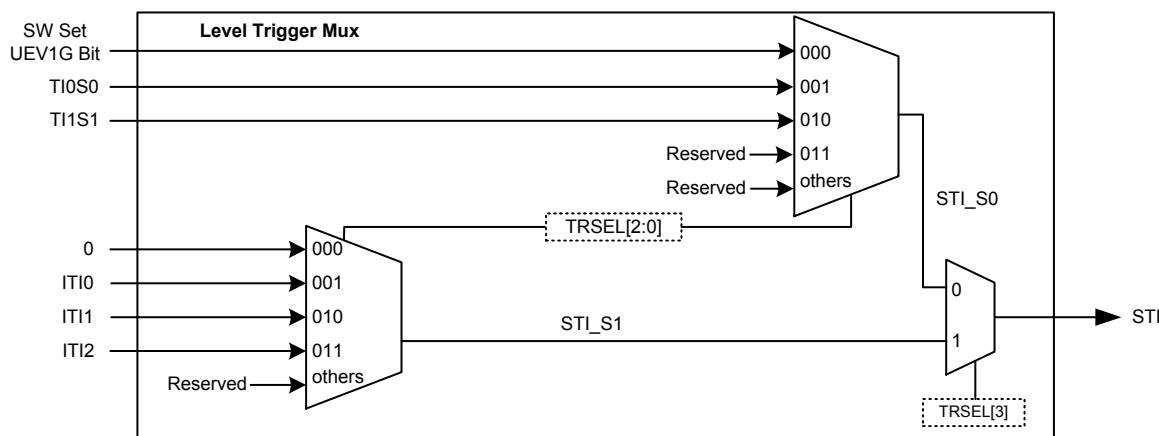
**Trigger Controller Block = Edge Trigger Mux + Level Trigger Mux**



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



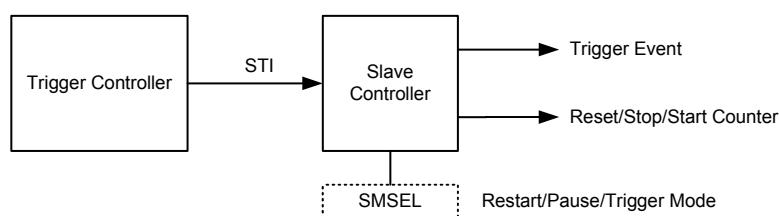
**Level Trigger Source = Internal (ITIx) + Channel input (TIn) + Software UEV1G bit**



**Figure 75. Trigger Controller Block**

## Slave Controller

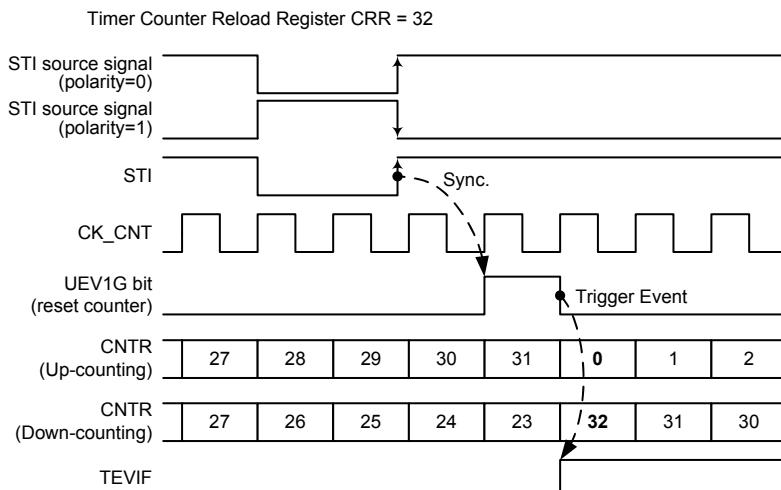
The MCTM can be synchronised with an internal/external trigger in several modes including the Restart mode, the Pause mode and the Trigger mode which are selected by the SMSEL field in the MDCFR register. The trigger input of these modes comes from the STI signal which is selected by the TRSEL field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



**Figure 76. Slave Controller Diagram**

### Restart Mode

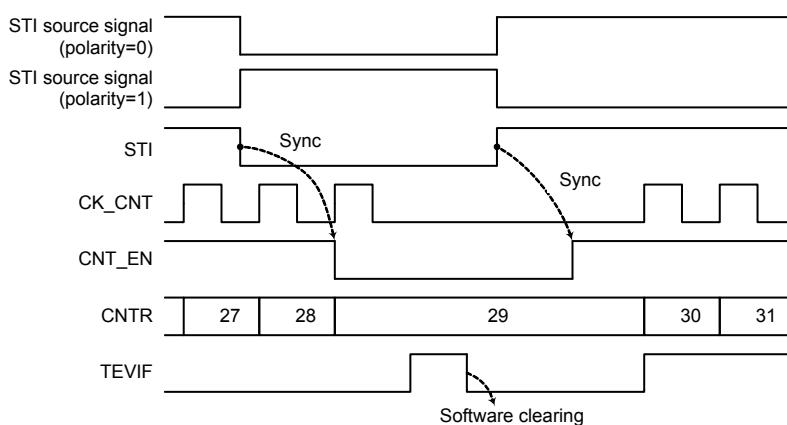
The counter and its prescaler can be reinitialised in response to an STI signal rising edge. If the UEV1DIS bit is set to 1 to disable the update event, then no update event will be generated, however the counter and prescaler are still reinitialized when an STI rising edge occurs. If the UEVIDIS bit 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 77. MCTM in Restart Mode**

### Pause Mode

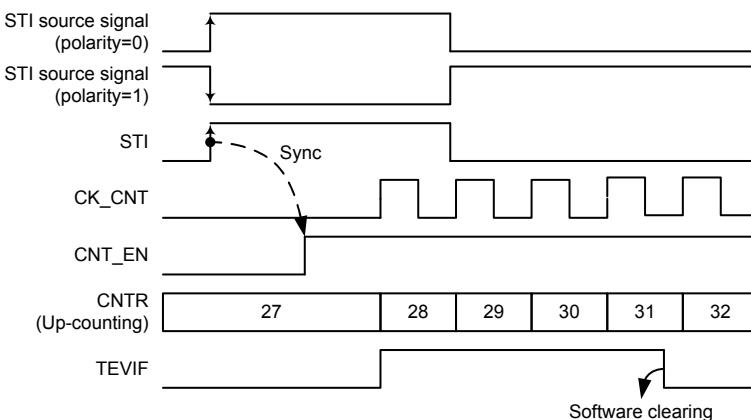
In the Pause Mode, the selected STI input signal level is used to control the counter start/stop operation. The counter starts to count when the selected STI signal is at a high level and stops counting when the STI signal is changed to a low level. When the counter stops, it will maintain its present 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 78. 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 UEV1G bit software trigger, the counter will not resume counting. When software triggering using the UEV1G 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 79. MCTM in Trigger Mode**

## Master Controller

The MCTM and GPTM 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.

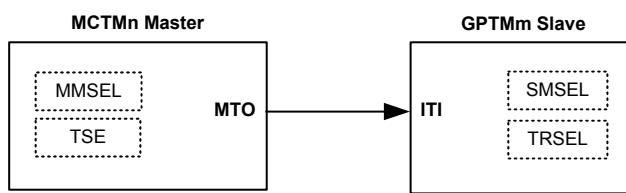


Figure 80. Master MCTMn and Slave GPTMm Connection

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

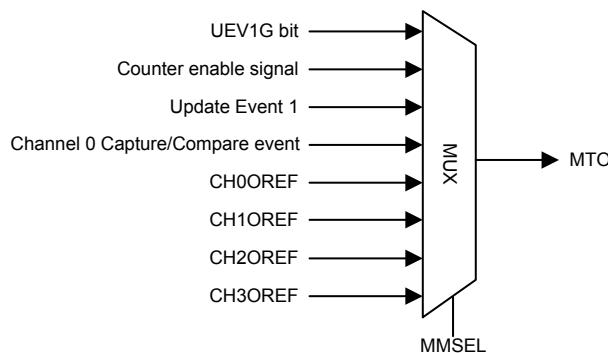


Figure 81. MTO Selection

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.

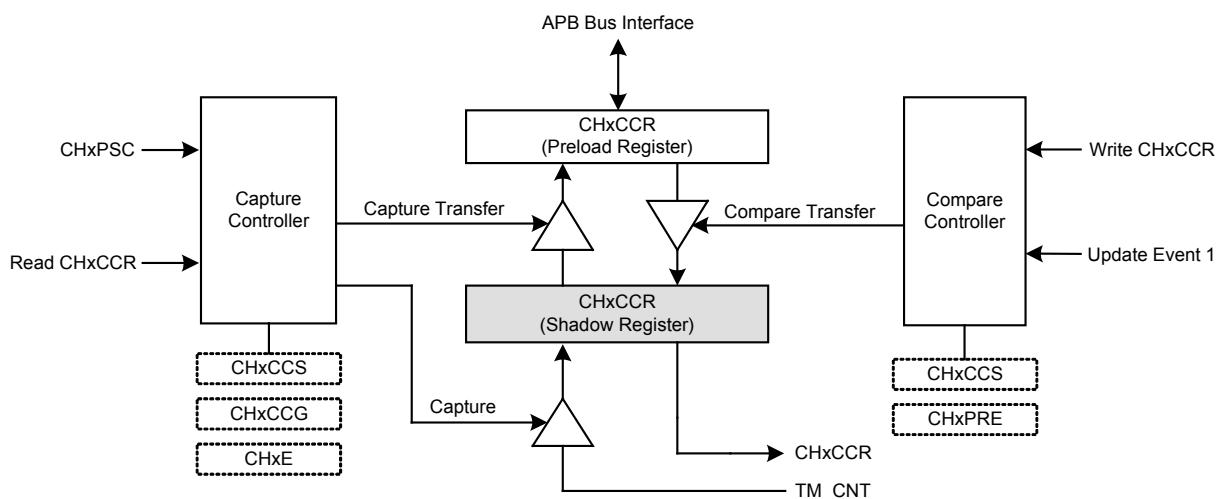
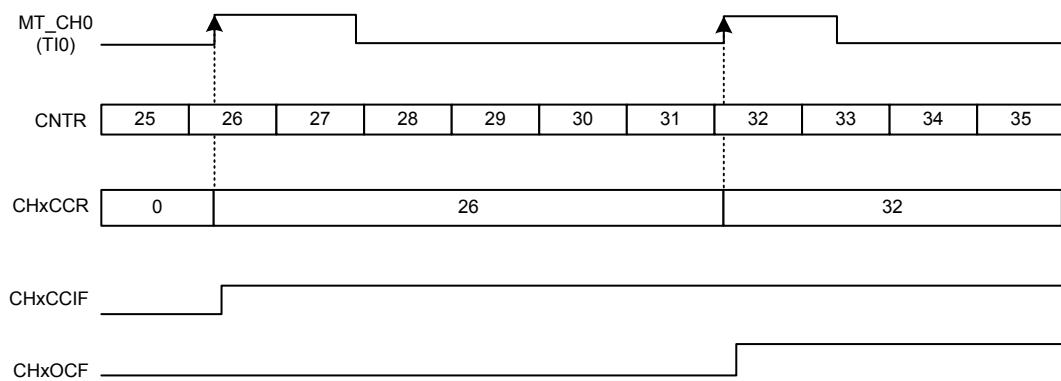


Figure 82. Capture/Compare Block Diagram

### 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.



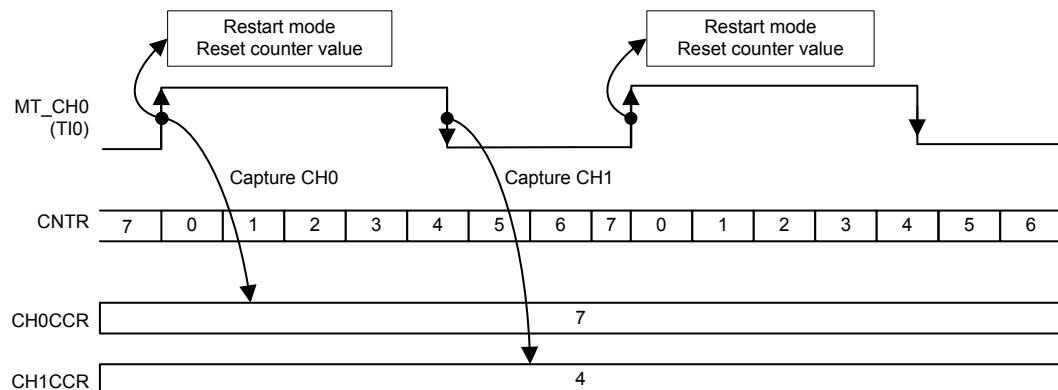
**Figure 83. Input Capture Mode**

### 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 CH1P bit to 1 to choose the falling edge of the TI0 input as the active polarity.
- Setup 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 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 84. 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 TI0 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 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 TIxSxED signal for the input capture function. The effective input event number can be set by the channel input prescaler register CHxPSC.

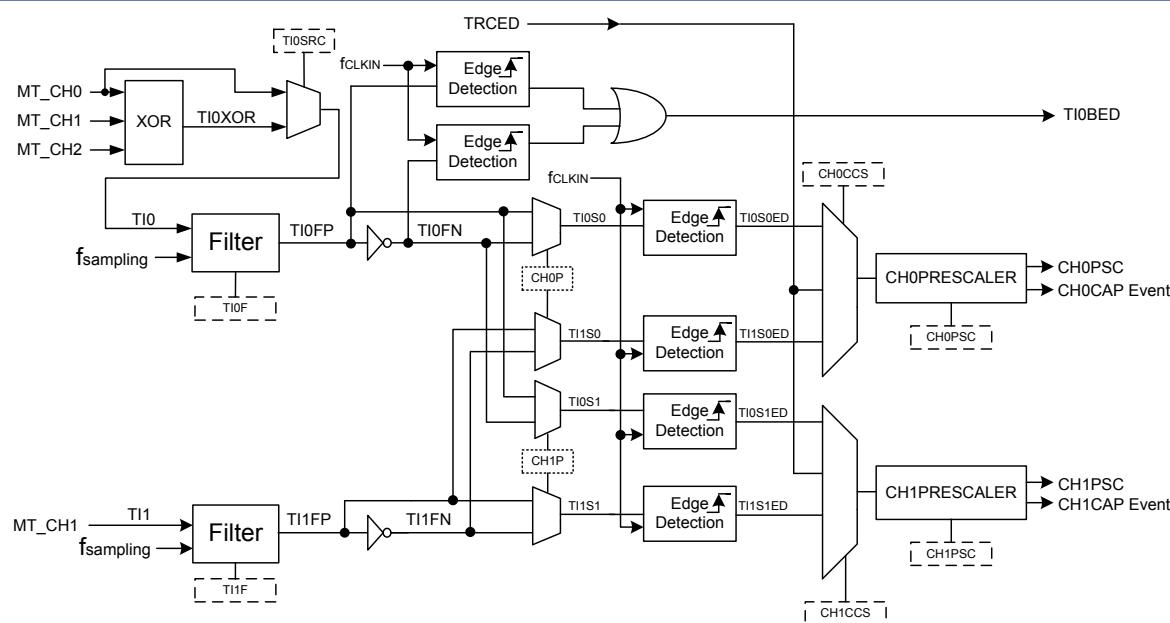


Figure 85. Channel 0 and Channel 1 Input Stages

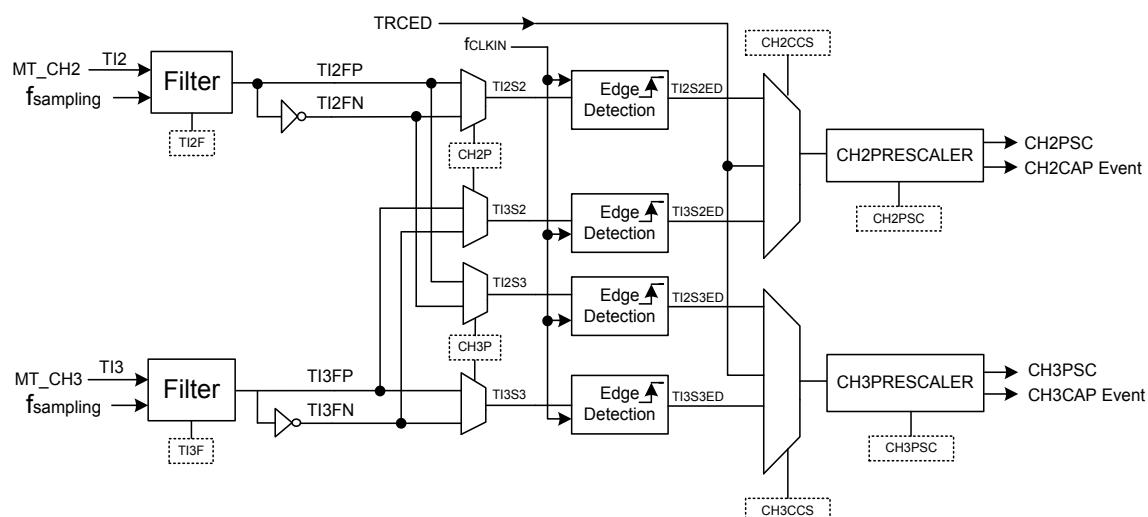


Figure 86. Channel 2 and Channel 3 Input Stages

### 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.

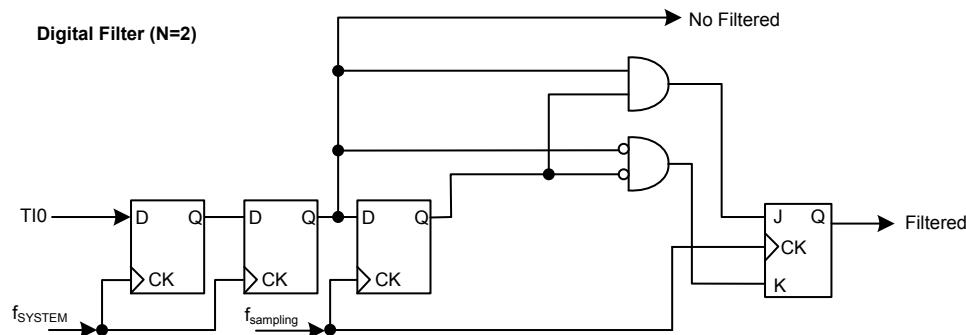


Figure 87. TI0 Digital Filter Diagram with  $N = 2$

## 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.

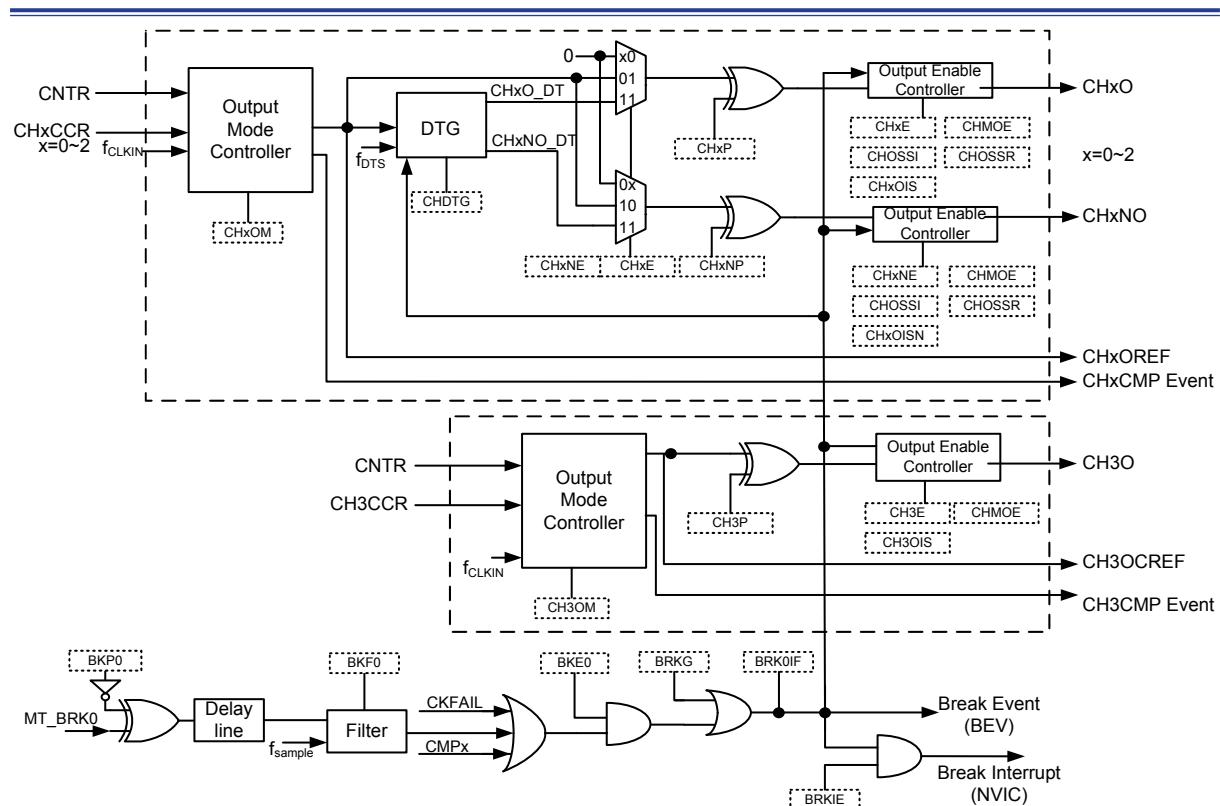


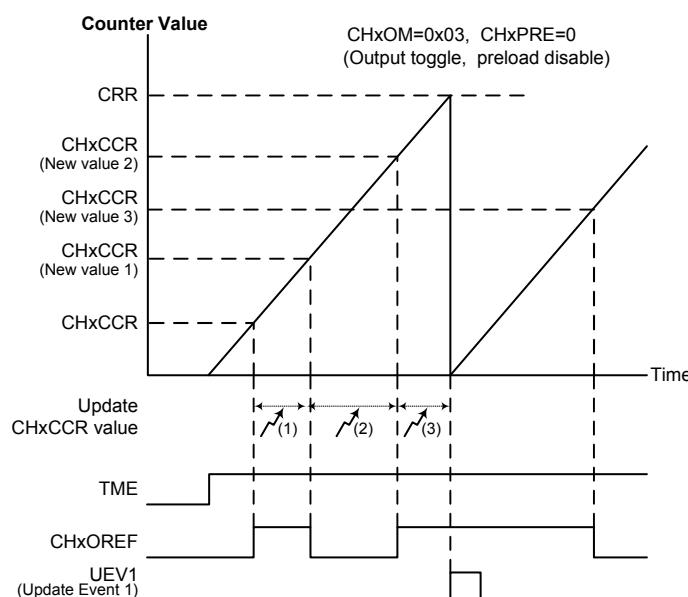
Figure 88. Output Stage Block Diagram

### Channel Output Reference Signal

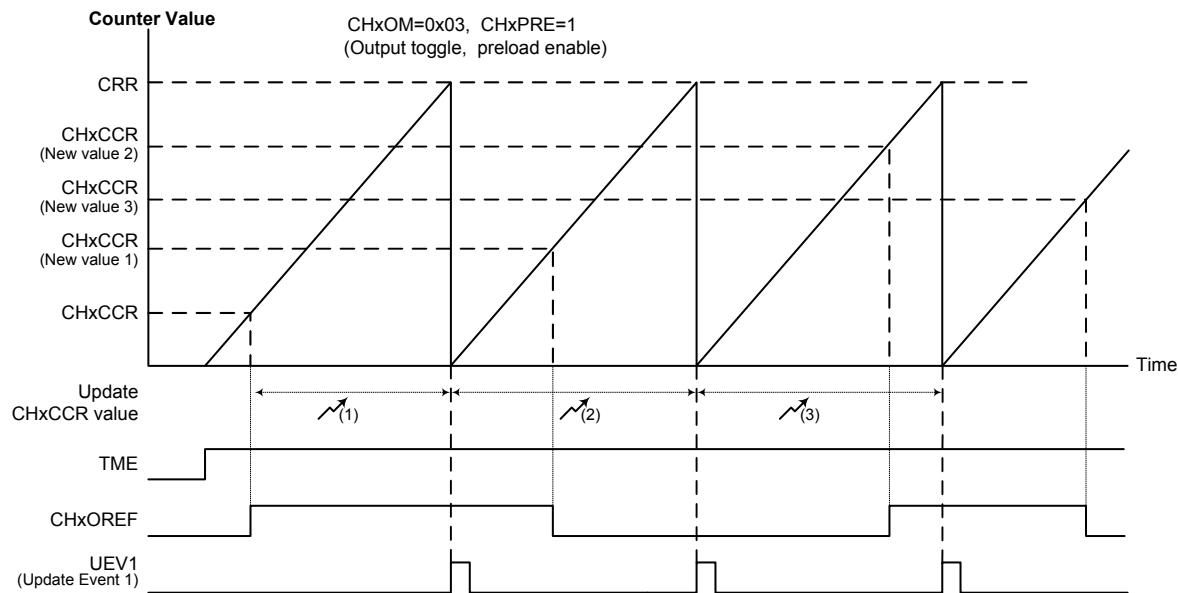
When the MCTM is used in the compare match output mode, the CHxOREF signal (Channel x Output Reference signal) is defined by the CHxOM bit setup. The CHxOREF signal has several types of output function which defines what happens to the output when the counter value matches the contents of the CHxCCR 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 34 shows a summary of the output type setup.

**Table 35. Compare Match Output Setup**

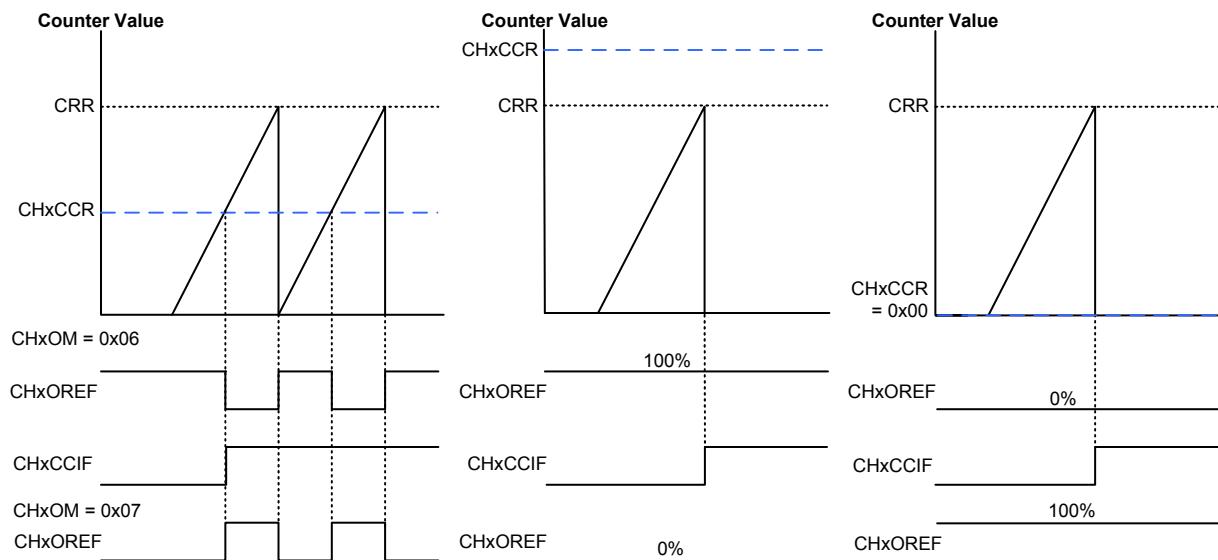
CHxOM value	Compare Match Level
0x00	No change
0x01	Clear Output to 0
0x02	Set Output to 1
0x03	Toggle Output
0x04	Force Inactive Level
0x05	Force Active Level
0x06	PWM Mode 1
0x07	PWM Mode 2



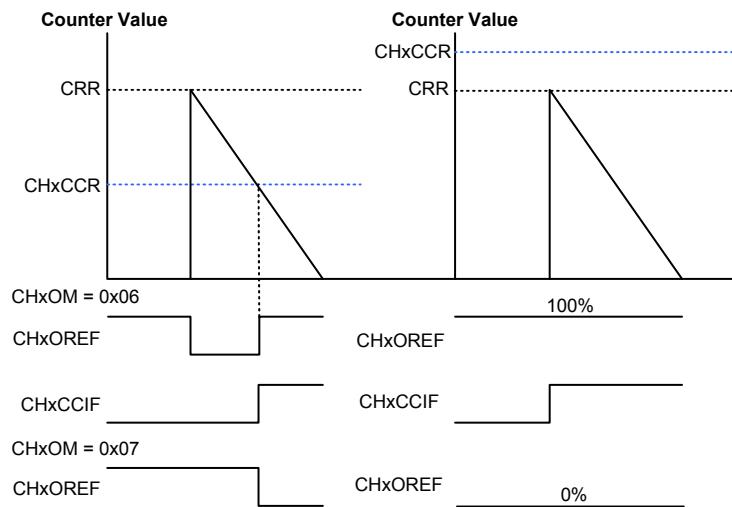
**Figure 89. Toggle Mode Channel Output Reference Signal – CHxPRE = 0**



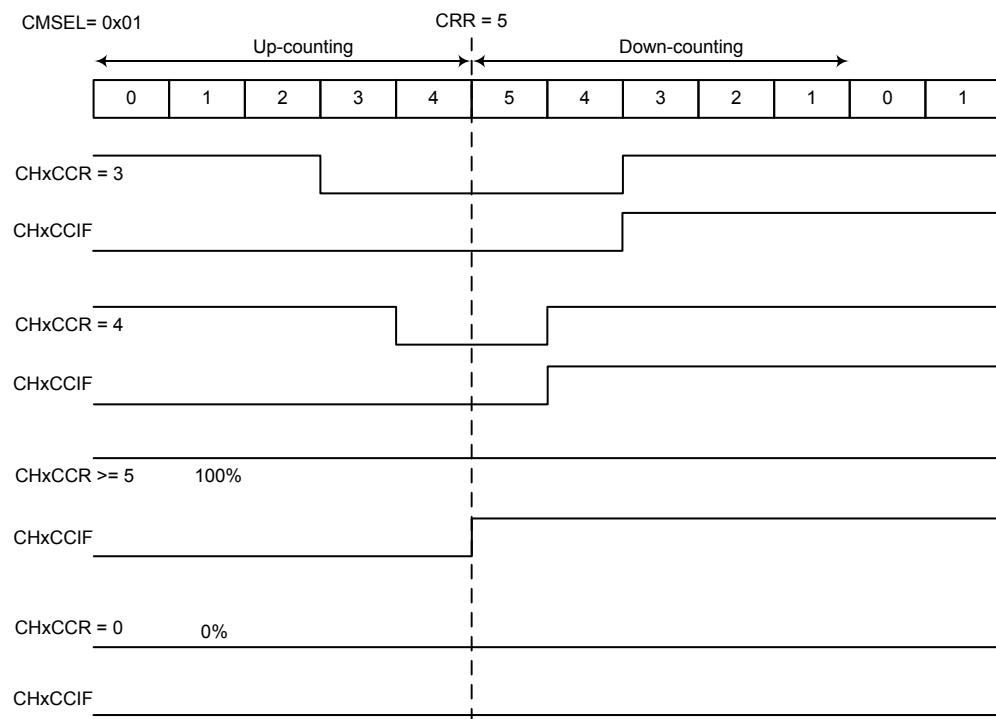
**Figure 90. Toggle Mode Channel Output Reference Signal – CHxPRE = 1**



**Figure 91. PWM Mode Channel Output Reference Signal and Counter in Up-counting Mode**



**Figure 92. PWM Mode Channel Output Reference Signal and Counter in Down-counting Mode**

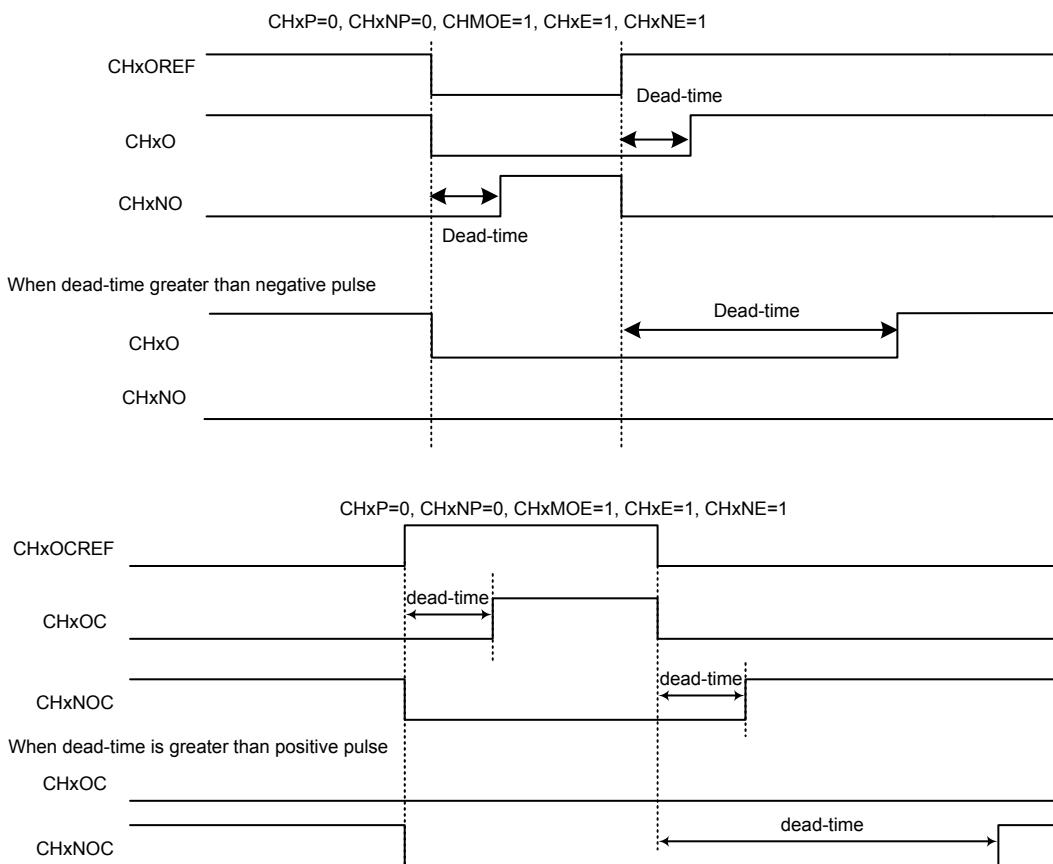


**Figure 93. PWM Mode 1 Channel Output Reference Signal and Counter in Centre-aligned Counting Mode**

### 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.

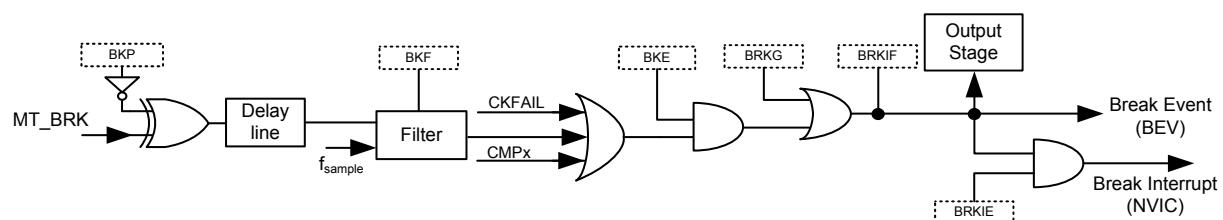


**Figure 94. Dead-time Insertion Performed for Complementary Outputs**

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.

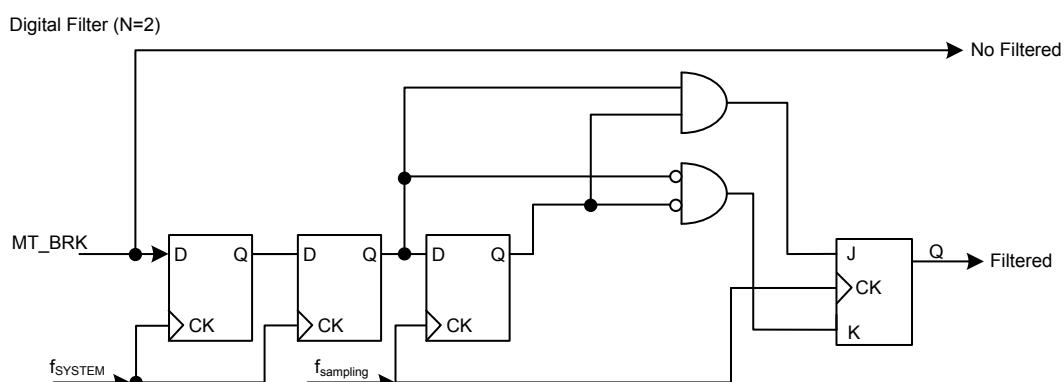


**Figure 95. MCTM Break Signal Block Diagram**

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 internal polarity of break activity function is logic high. So 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 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.



**Figure 96. 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.

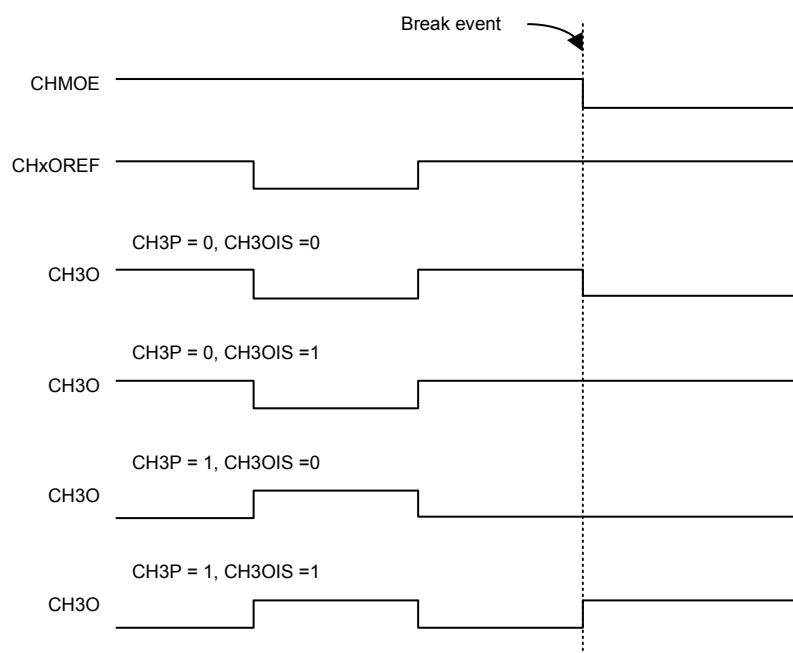


Figure 97. Channel 3 Output with a Break Event Occurrence

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.

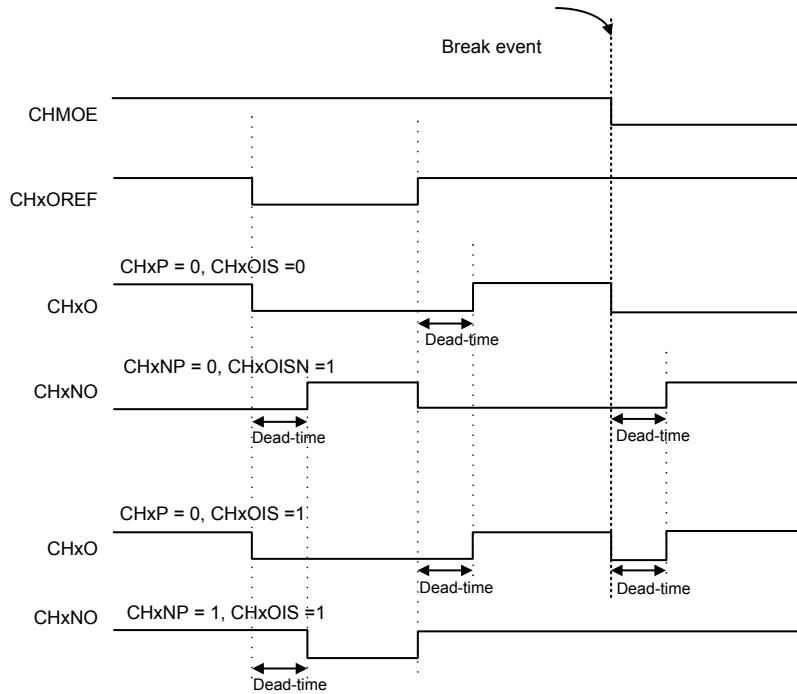


Figure 98. Channel 0 ~2 Complementary Outputs with a Break Event Occurrence

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.

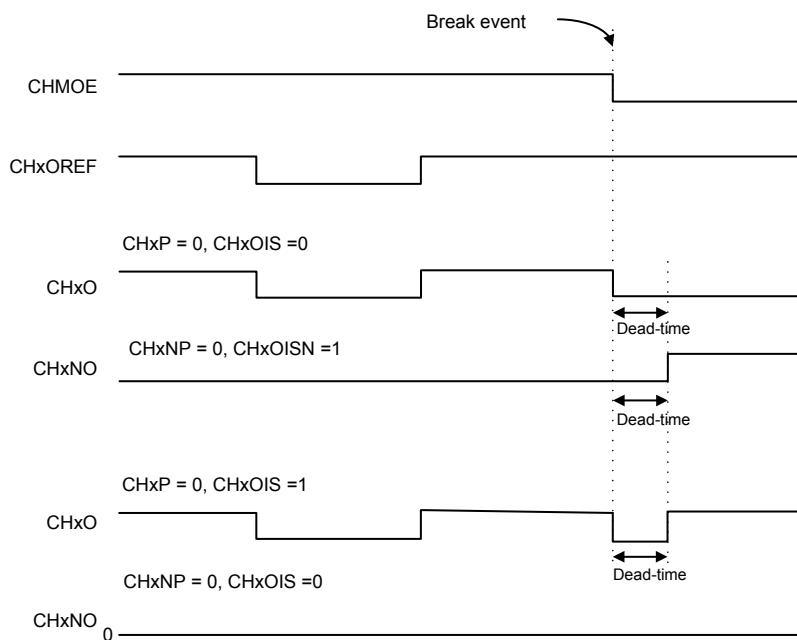
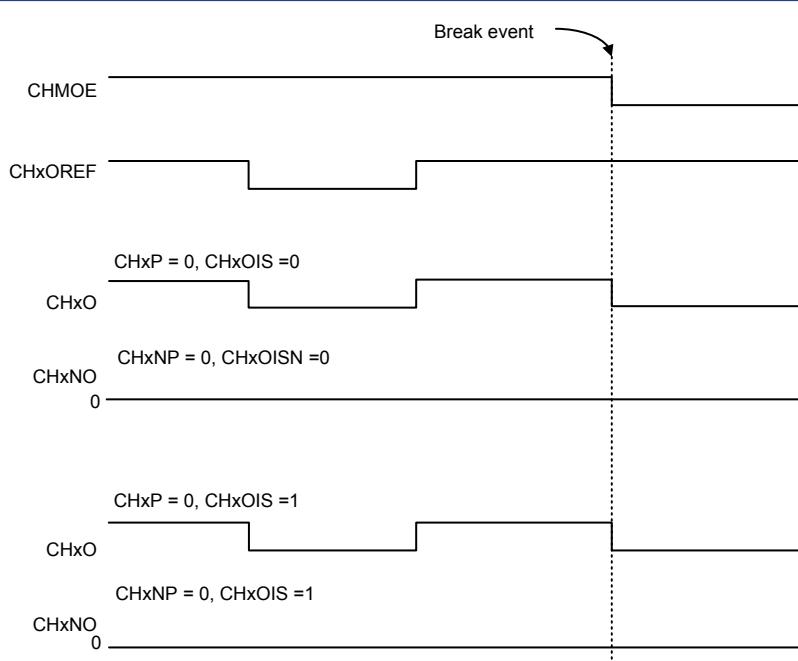


Figure 99. Channel 0 ~2 Only One Output Enabled when Break Event Occurs

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 100. 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	CHOSSI	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 <sup>(Note 1)</sup> = floating MT_CHx_OEN <sup>(Note 2)</sup> = 1	Output disabled - floating - not driven by the timer MT_CHxN = floating MT_CHxN_OEN = 1
		0	0	1	Output disable - 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	
		1	0	1		
		1	1	0		
		1	1	1	After dead-time: Output enabled MT_CHx = CHxOIS, MT_CHxN = CHxOISN MT_CHx_OEN = 0, MT_CHxN_OEN = 0	

**Notes:** 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.

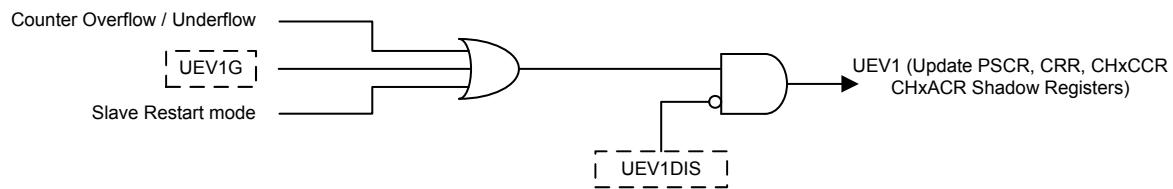
## Update Management

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 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.

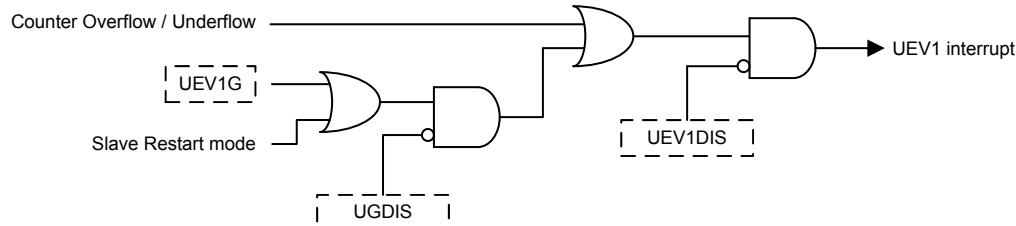
### Update Event 1

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



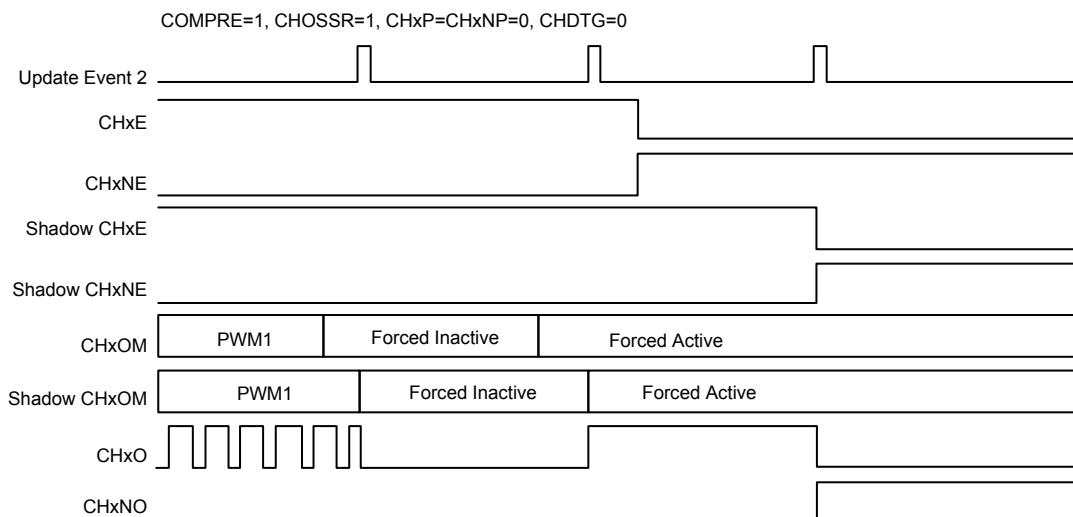
#### Update Event 1 Interrupt Management



**Figure 101. Update Event 1 Setup Diagram**

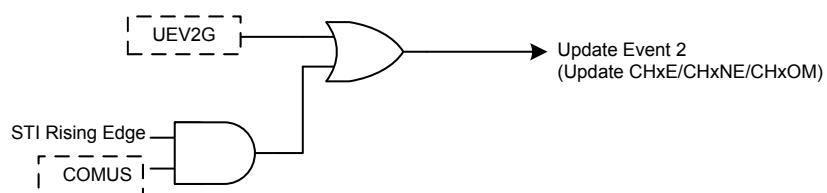
### Update Event 2

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 102. CHxE, CHxNE and CHxOM Updated by Update Event 2**

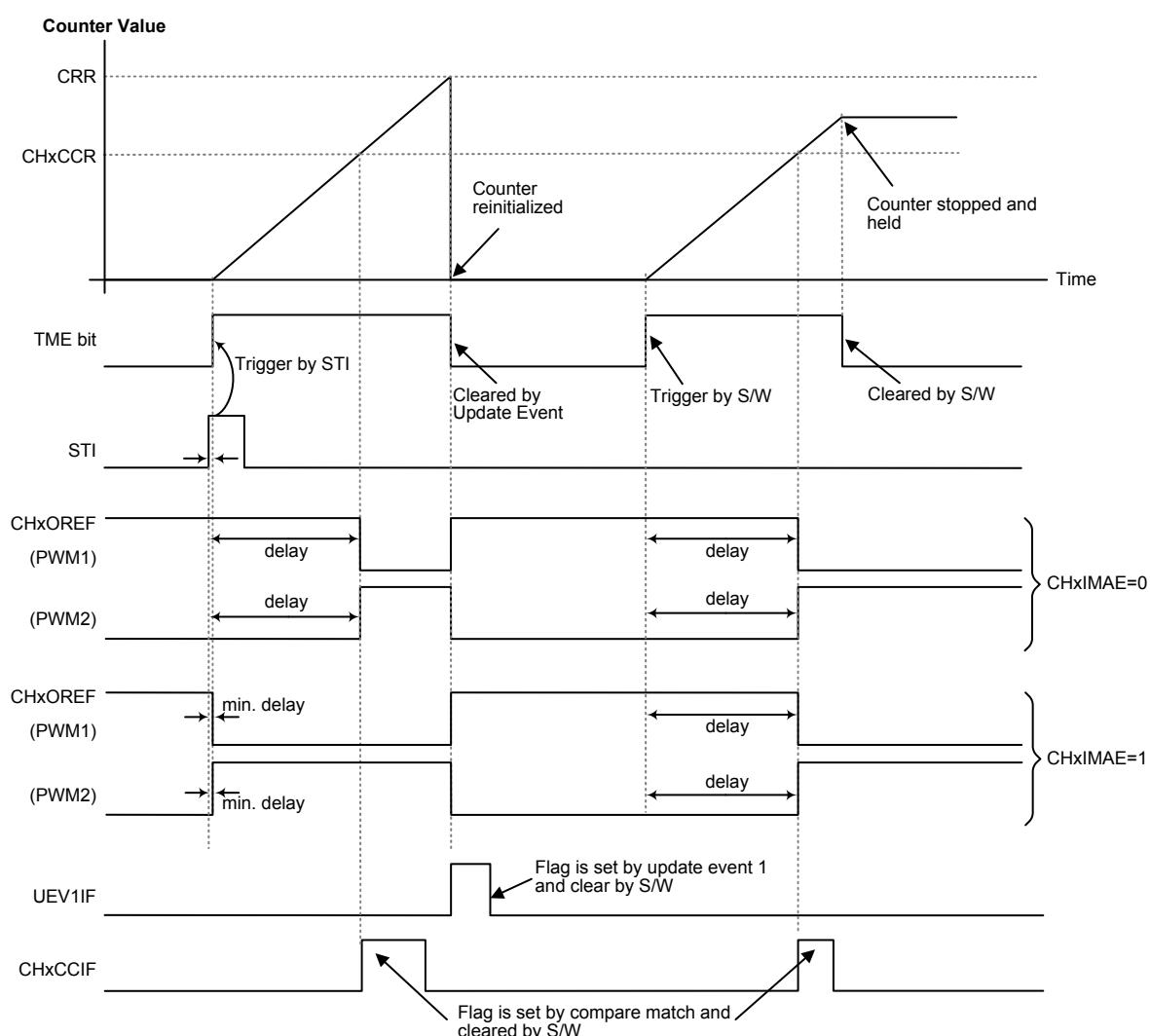
An update event 2 can be generated by setting the software update bit, UEV2G, in the EVGR register or by the rising edge of the STI signal if the COMUS bit is set in the CTR register.



**Figure 103. Update Event 2 Setup Diagram**

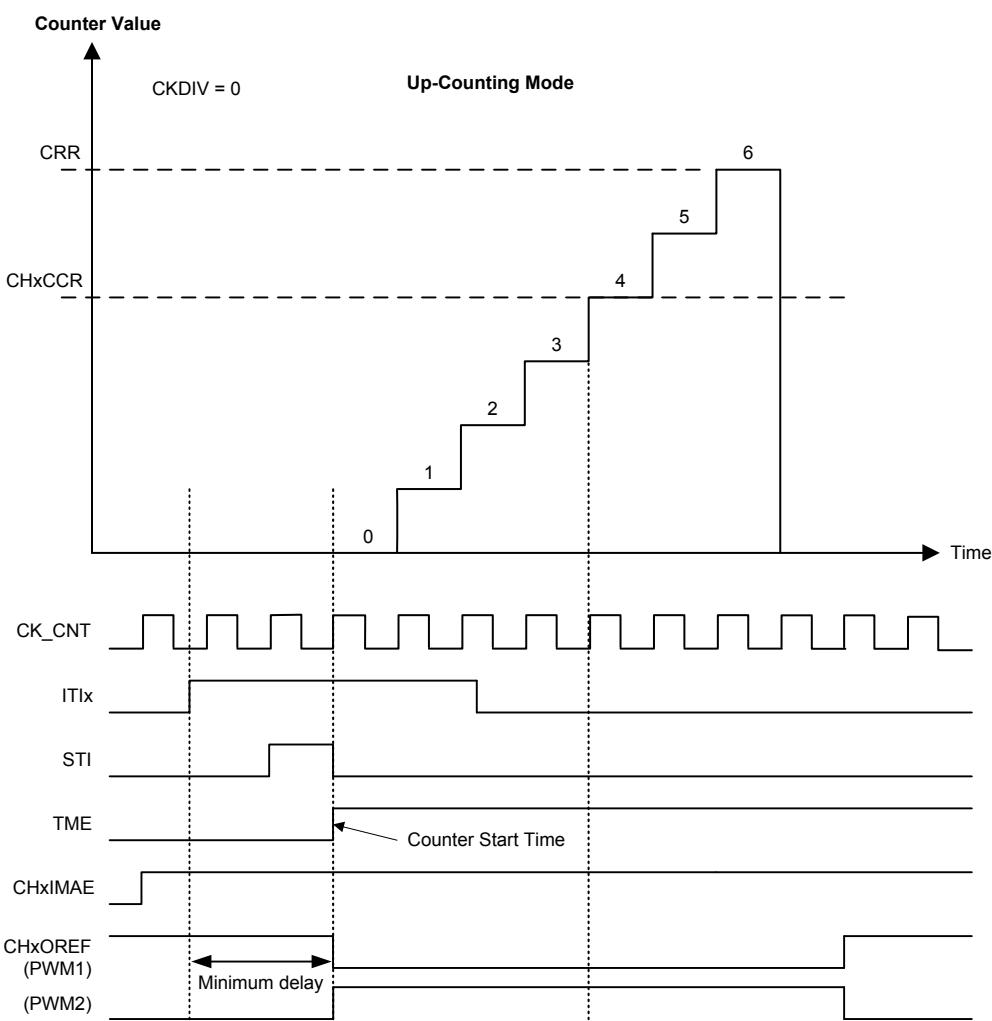
## Single Pulse Mode

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 104. 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.

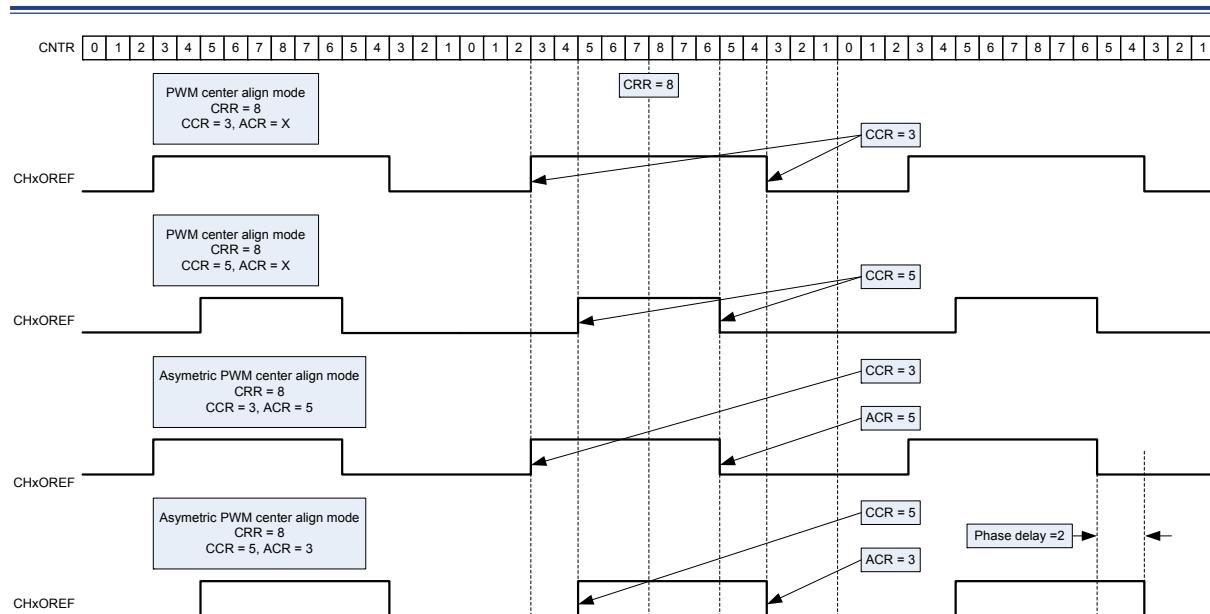


**Figure 105. Immediate Active Mode Minimum Delay**

## 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 105 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.



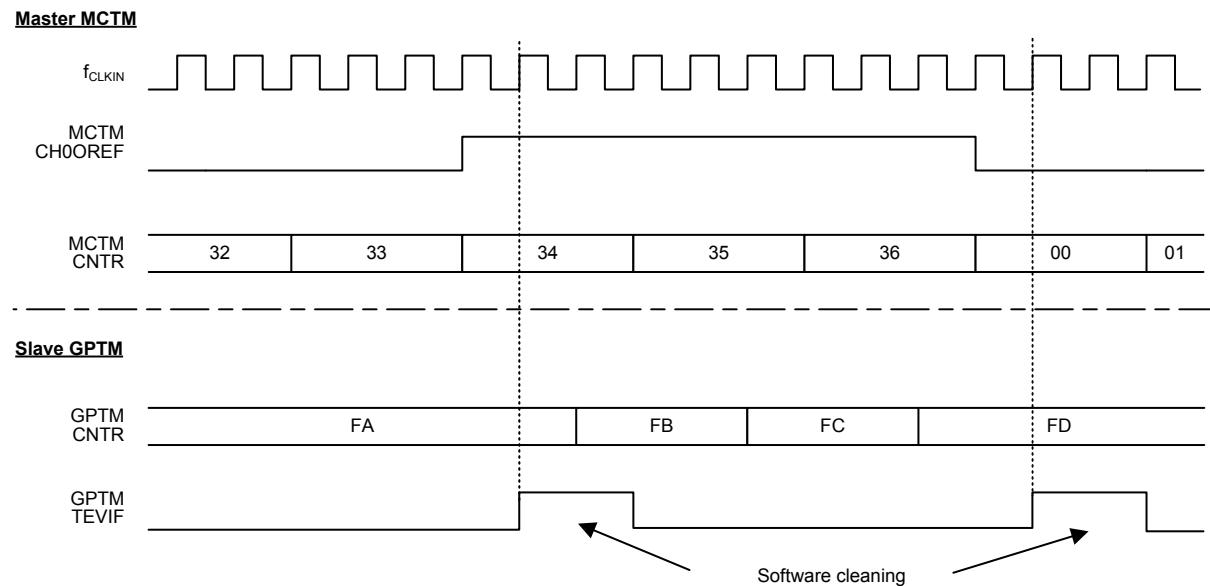
**Figure 106. Asymmetric PWM Mode versus Center-aligned Counting Mode**

## 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

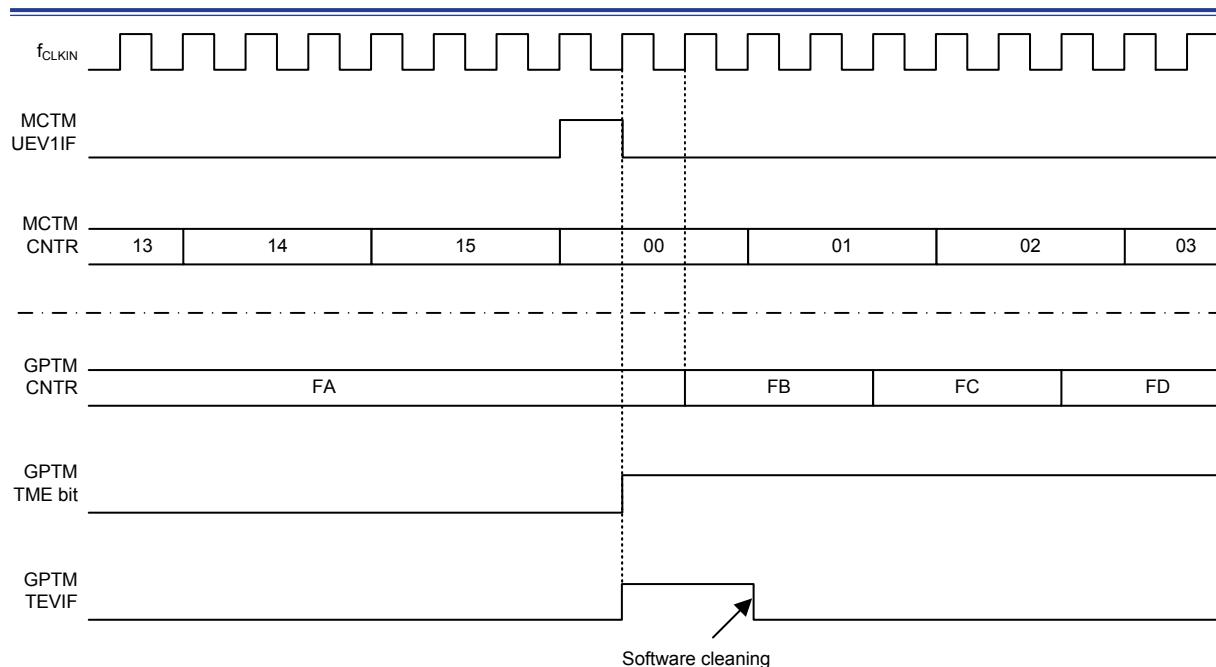
- Configure MCTM to be in the master mode and to send its channel 0 Output Reference signal CH0OREF as a trigger output (MMSEL = 0x04).
- Configure the MCTM CH0OREF waveform.
- Configure the GPTM to receive its input trigger source from the MCTM trigger output (TRSEL = 0x0A).
- Configure GPTM to operate 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 107. 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 108. 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).

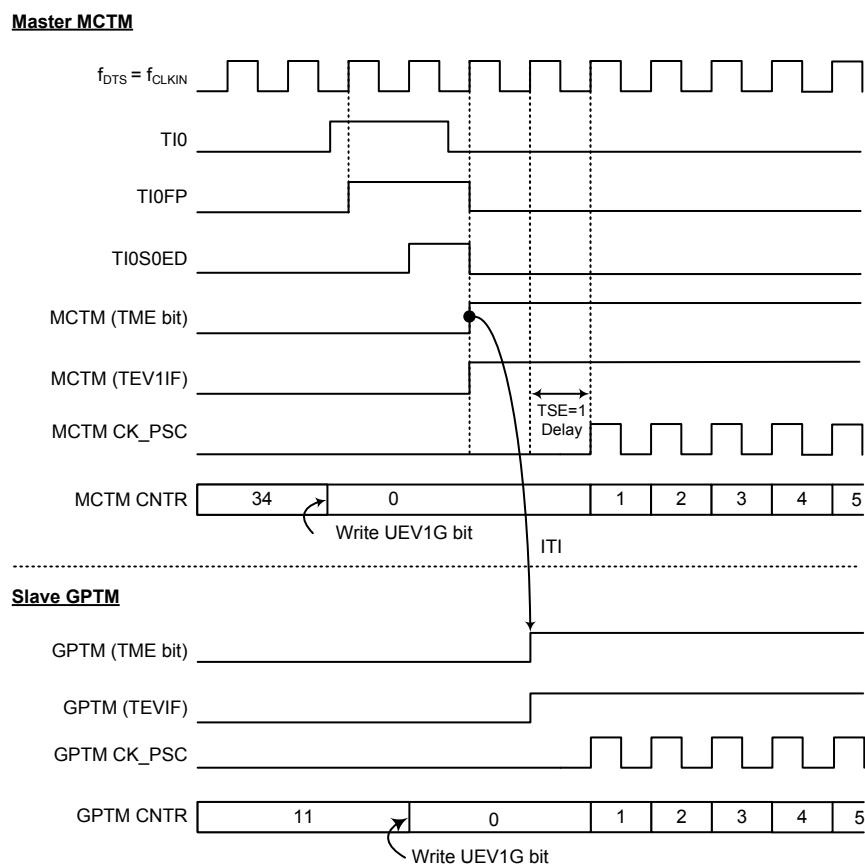


Figure 109. Trigger MCTM and GPTM with the MCTM CH0 Input

### 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 (COMPRE = 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

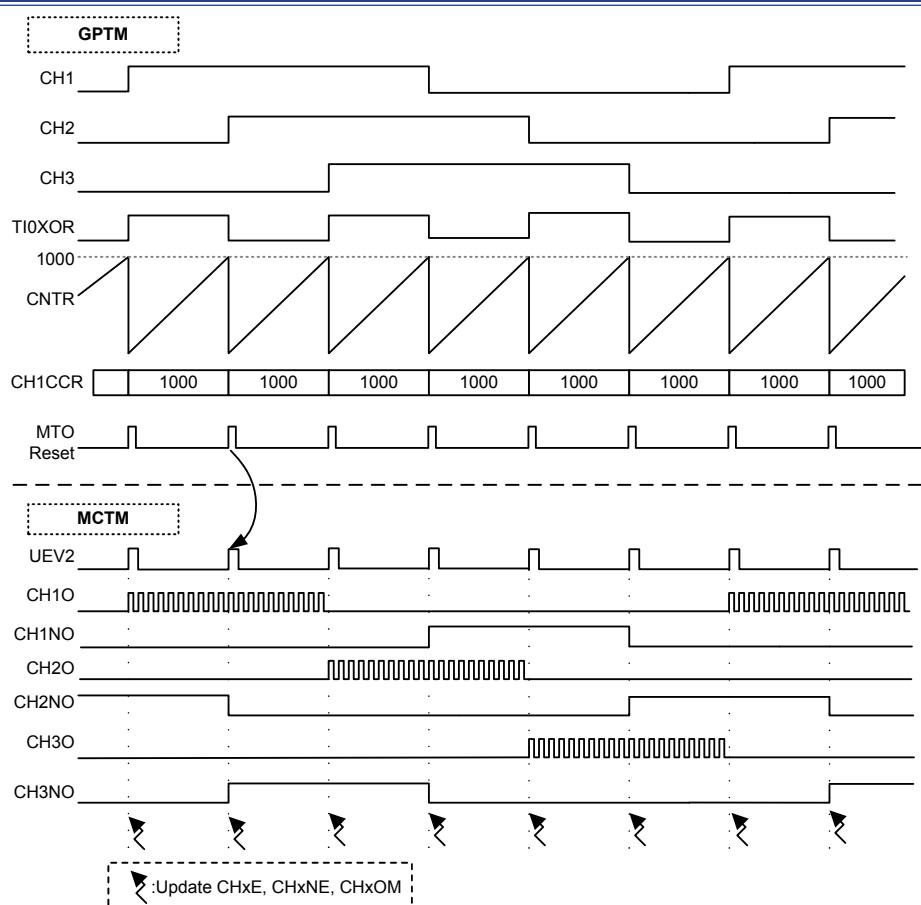


Figure 110. CH1XOR Input as Hall Sensor Interface

## 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 ( $x = 0 \sim 3$ ) 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 <sup>(1)</sup>	CKMEN <sup>(2)</sup>
Lock Level 3 (LOCKLV = '11')	CHDTG	CHxOIS	CHxOISN	BKE	BKP	CHAOE
	CHxP	CHxNP	CHOSSI	CHOSSR	MCTMEN <sup>(1)</sup>	CKMEN <sup>(2)</sup>
	CHxPRE	CHxOM				

- Notes:**
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.

## 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 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	0xA0	Channel 0 Asymmetric Compare Register	0x0000_0000
CH1ACR	0xA4	Channel 1 Asymmetric Compare Register	0x0000_0000
CH2ACR	0xA8	Channel 2 Asymmetric Compare Register	0x0000_0000
CH3ACR	0xAC	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

Type/Reset	31	30	29	28	27	26	25	24	
	Reserved								DIR
									RW 0
Type/Reset	23	22	21	20	19	18	17	16	
	Reserved								CMSEL
									RW 0 RW 0
Type/Reset	15	14	13	12	11	10	9	8	
	Reserved								CKDIV
									RW 0 RW 0
Type/Reset	7	6	5	4	3	2	1	0	
	Reserved								UGDIS UEV1DIS
									RW 0 RW 0

Bits	Field	Descriptions
[24]	DIR	<p>Counting Direction</p> <p>0: Count-up 1: Count-down</p> <p>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</p>
[17:16]	CMSEL	<p>Counter Mode Selection</p> <p>00: Edge-aligned counting mode. Normal up-counting and down-counting available for this mode. Counting direction is defined by the DIR bit.</p> <p>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.</p> <p>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.</p> <p>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.</p>
[9:8]	CKDIV	<p>Clock Division</p> <p>These two bits define the frequency ratio between the timer clock (<math>f_{CLKIN}</math>) and the dead-time clock (<math>f_{DTS}</math>). The dead-time clock is also used as the digital filter sampling clock</p> <p>00: <math>f_{DTS} = f_{CLKIN}</math> 01: <math>f_{DTS} = f_{CLKIN}/2</math> 10: <math>f_{DTS} = f_{CLKIN}/4</math> 11: Reserved</p>

<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>
[1]	UGDIS	<p>Update event 1 interrupt generation disable control</p> <p>0: Any of the following events will generate an update interrupt</p> <ul style="list-style-type: none"><li>- Counter overflow / underflow</li><li>- Setting the UEV1G bit</li><li>- Update generation through the slave mode</li></ul> <p>1: Only counter overflow/underflow generates an update interrupt</p>
[0]	UEV1DIS	<p>Update event 1 Disable control</p> <p>0: Enable the update event 1 request by one of following events</p> <ul style="list-style-type: none"><li>- Counter overflow / underflow</li><li>- Setting the UEV1G bit</li><li>- Update generation through the slave mode</li></ul> <p>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</p>

## 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
	15	14	13	12	11	10	9	8
Type/Reset			Reserved			SMSEL		
						RW	0	RW
	7	6	5	4	3	2	1	0
Type/Reset			Reserved				TSE	
							RW	0

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

Bits	Field	Descriptions	
[18:16]	MMSEL	Master Mode Selection	
Master mode selection is used to select the MTO signal source which is used to synchronise the other slave timer.			
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.	

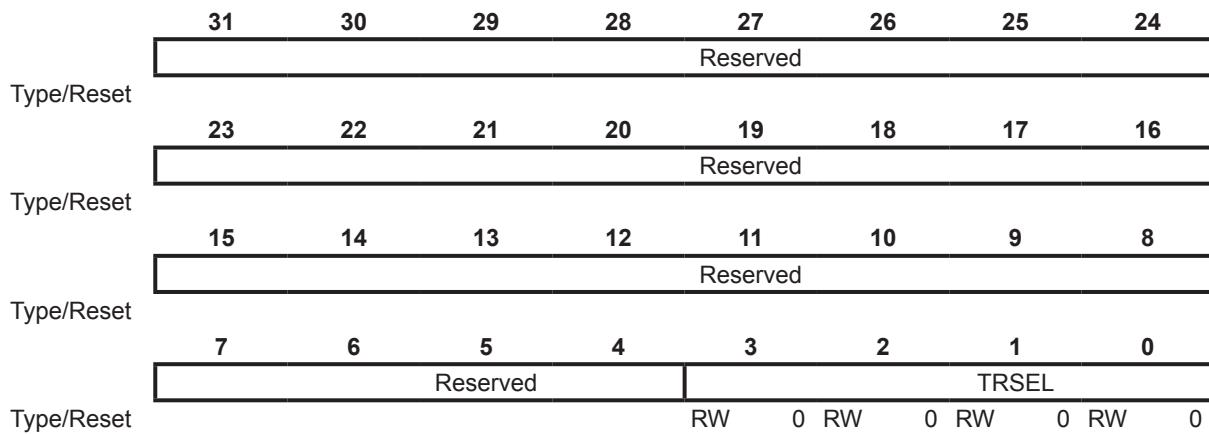
Bits	Field	Descriptions		
[10:8]	SMSEL	Slave Mode Selection		
		<b>SMSEL [2:0]</b>	<b>Mode</b>	<b>Descriptions</b>
		000	Disable mode	The prescaler is clocked directly by the internal clock.
		001	Reserved	
		010	Reserved	
		011	Reserved	
		100	Restart Mode	The counter value restarts from 0 or the CRR 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	Timer Synchronisation Enable		
		0: No action 1: Master timer (current timer) will generate a delay to synchronise its slave timer through the MTO signal.		

## 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"> <li>0000: Software Trigger by setting the UEV1G bit</li> <li>0001: Channel 0 filtered input – TI0S0</li> <li>0010: Channel 1 filtered input – TI1S1</li> <li>0011: Reserved</li> <li>1000: Channel 0 Edge Detector – TI0BED</li> <li>1001: Internal Timer Trigger 0 – ITI0</li> <li>1010: Internal Timer Trigger 1 – ITI1</li> <li>1011: Internal Timer Trigger 2 – ITI2</li> <li>Others: Default 0</li> </ul> <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**

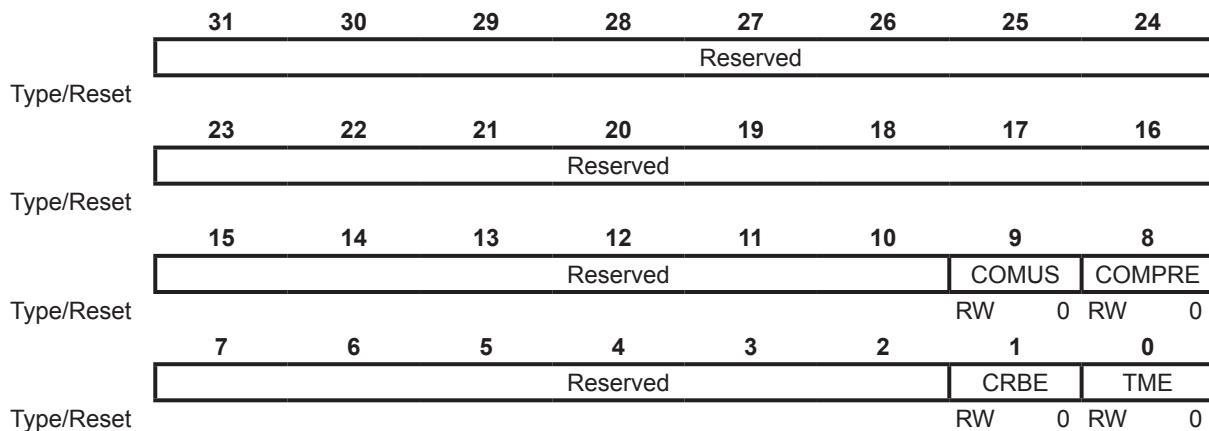
Slave Timing Module	ITI0	ITI1	ITI2
MCTM	X	GPTM	X

## Timer Counter Register – CTR

This register specifies the timer enable bit (TME), CRR buffer enable bit (CRBE) and Capture/compare control bit.

Offset: 0x010

Reset value: 0x0000\_0000



Bits	Field	Descriptions
[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 COMPRE bit to 1.
[8]	COMPRE	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
Type/Reset	TI0SRC				Reserved			
	RW	0						
Type/Reset	23	22	21	20	19	18	17	16
			Reserved			CH0PSC		CH0CCS
					RW	0	RW	0
Type/Reset	15	14	13	12	11	10	9	8
					Reserved			
Type/Reset	7	6	5	4	3	2	1	0
			Reserved				TI0F	
					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.

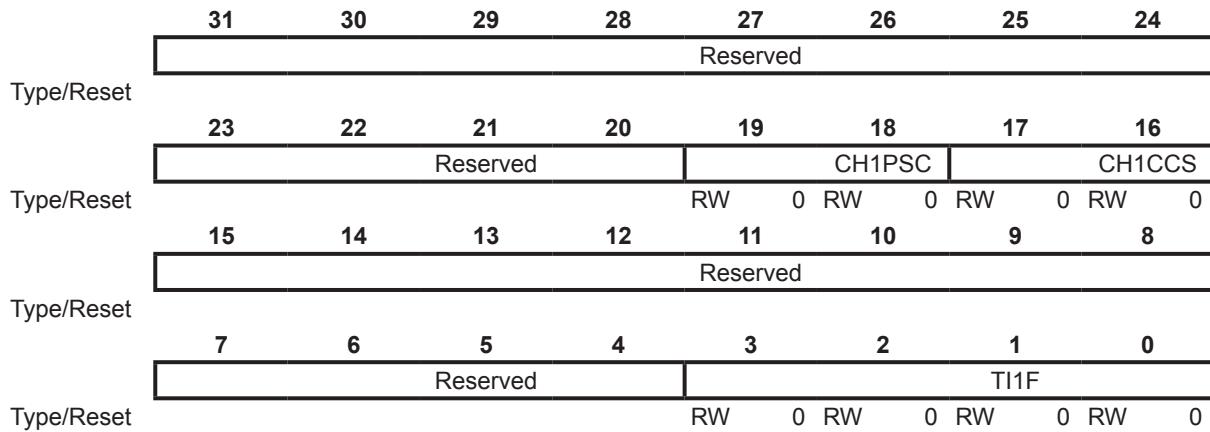
Bits	Field	Descriptions
[3:0]	TIOF	<p>Channel 0 Input Source TIO Filter Setting</p> <p>These bits define the frequency divided ratio used to sample the TIO signal. The Digital filter in the MCTM is an N-event counter where N is defined as how many valid transitions are necessary to output a filtered signal.</p> <ul style="list-style-type: none"> <li>0000: No filter, the sampling clock is <math>f_{SYSTEM}</math>.</li> <li>0001: <math>f_{SAMPLING} = f_{CLKIN}</math>, <math>N = 2</math></li> <li>0010: <math>f_{SAMPLING} = f_{CLKIN}</math>, <math>N = 4</math></li> <li>0011: <math>f_{SAMPLING} = f_{CLKIN}</math>, <math>N = 8</math></li> <li>0100: <math>f_{SAMPLING} = f_{DTS}/2</math>, <math>N = 6</math></li> <li>0101: <math>f_{SAMPLING} = f_{DTS}/2</math>, <math>N = 8</math></li> <li>0110: <math>f_{SAMPLING} = f_{DTS}/4</math>, <math>N = 6</math></li> <li>0111: <math>f_{SAMPLING} = f_{DTS}/4</math>, <math>N = 8</math></li> <li>1000: <math>f_{SAMPLING} = f_{DTS}/8</math>, <math>N = 6</math></li> <li>1001: <math>f_{SAMPLING} = f_{DTS}/8</math>, <math>N = 8</math></li> <li>1010: <math>f_{SAMPLING} = f_{DTS}/16</math>, <math>N = 5</math></li> <li>1011: <math>f_{SAMPLING} = f_{DTS}/16</math>, <math>N = 6</math></li> <li>1100: <math>f_{SAMPLING} = f_{DTS}/16</math>, <math>N = 8</math></li> <li>1101: <math>f_{SAMPLING} = f_{DTS}/32</math>, <math>N = 5</math></li> <li>1110: <math>f_{SAMPLING} = f_{DTS}/32</math>, <math>N = 6</math></li> <li>1111: <math>f_{SAMPLING} = f_{DTS}/32</math>, <math>N = 8</math></li> </ul>

## Channel 1 Input Configuration Register – CH1ICFR

This register specifies the channel 1 input mode configuration.

Offset: 0x024

Reset value: 0x0000\_0000



Bits	Field	Descriptions
[19:18]	CH1PSC	<p>Channel 1 Capture Input Source Prescaler Setting</p> <p>These bits define the effective events of the channel 1 capture input. Note that the prescaler is reset once the Channel 1 Capture/Compare Enable bit, CH1E, in the Channel Control register named CHCTR is cleared to 0.</p> <ul style="list-style-type: none"> <li>00: No prescaler, channel 1 capture input signal is chosen for each active event</li> <li>01: Channel 1 Capture input signal is chosen for every 2 events</li> <li>10: Channel 1 Capture input signal is chosen for every 4 events</li> <li>11: Channel 1 Capture input signal is chosen for every 8 events</li> </ul>
[17:16]	CH1CCS	<p>Channel 1 Capture/Compare Selection</p> <ul style="list-style-type: none"> <li>00: Channel 1 is configured as an output</li> <li>01: Channel 1 is configured as an input derived from the TI1 signal</li> <li>10: Channel 1 is configured as an input derived from the TI0 signal</li> <li>11: Channel 1 is configured as an input which comes from the TRCED signal derived from the Trigger Controller</li> </ul> <p>Note: The CH1CCS field can be accessed only when the CH1E bit is cleared to 0.</p>

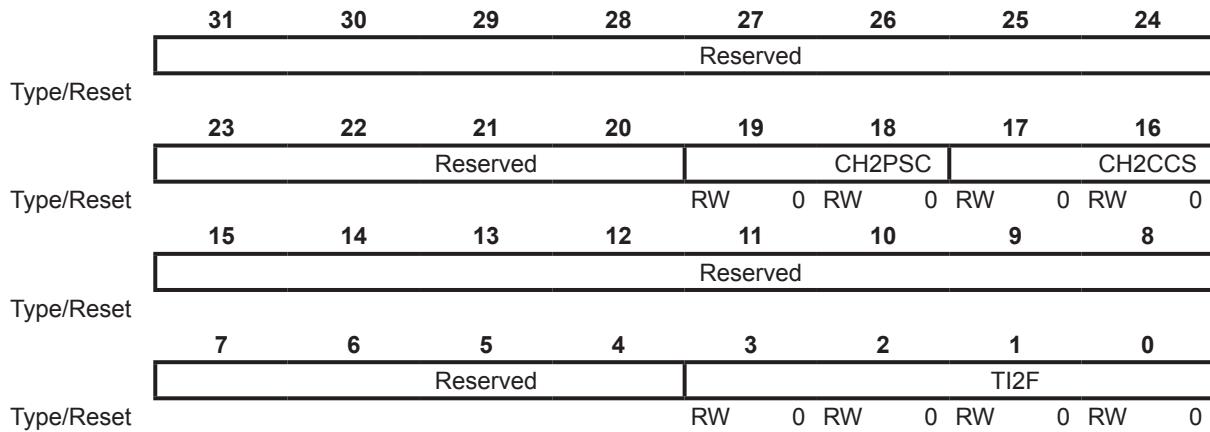
Bits	Field	Descriptions
[3:0]	TI1F	<p>Channel 1 Input Source TI1 Filter Setting</p> <p>These bits define the frequency divide ratio used to sample the TI1 signal. The Digital filter in the MCTM is an N-event counter where N is defined as how many valid transitions are necessary to output a filtered signal</p> <ul style="list-style-type: none"> <li>0000: No filter, the sampling clock is <math>f_{SYSTEM}</math>.</li> <li>0001: <math>f_{SAMPLING} = f_{CLKIN}</math>, <math>N = 2</math></li> <li>0010: <math>f_{SAMPLING} = f_{CLKIN}</math>, <math>N = 4</math></li> <li>0011: <math>f_{SAMPLING} = f_{CLKIN}</math>, <math>N = 8</math></li> <li>0100: <math>f_{SAMPLING} = f_{DTS}/2</math>, <math>N = 6</math></li> <li>0101: <math>f_{SAMPLING} = f_{DTS}/2</math>, <math>N = 8</math></li> <li>0110: <math>f_{SAMPLING} = f_{DTS}/4</math>, <math>N = 6</math></li> <li>0111: <math>f_{SAMPLING} = f_{DTS}/4</math>, <math>N = 8</math></li> <li>1000: <math>f_{SAMPLING} = f_{DTS}/8</math>, <math>N = 6</math></li> <li>1001: <math>f_{SAMPLING} = f_{DTS}/8</math>, <math>N = 8</math></li> <li>1010: <math>f_{SAMPLING} = f_{DTS}/16</math>, <math>N = 5</math></li> <li>1011: <math>f_{SAMPLING} = f_{DTS}/16</math>, <math>N = 6</math></li> <li>1100: <math>f_{SAMPLING} = f_{DTS}/16</math>, <math>N = 8</math></li> <li>1101: <math>f_{SAMPLING} = f_{DTS}/32</math>, <math>N = 5</math></li> <li>1110: <math>f_{SAMPLING} = f_{DTS}/32</math>, <math>N = 6</math></li> <li>1111: <math>f_{SAMPLING} = f_{DTS}/32</math>, <math>N = 8</math></li> </ul>

## Channel 2 Input Configuration Register – CH2ICFR

This register specifies the channel 2 input mode configuration.

Offset: 0x028

Reset value: 0x0000\_0000



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"> <li>00: No prescaler, channel 2 capture input signal is chosen for each active event</li> <li>01: Channel 2 Capture input signal is chosen for every 2 events</li> <li>10: Channel 2 Capture input signal is chosen for every 4 events</li> <li>11: Channel 2 Capture input signal is chosen for every 8 events</li> </ul>
[17:16]	CH2CCS	<p>Channel 2 Capture/Compare Selection</p> <ul style="list-style-type: none"> <li>00: Channel 2 is configured as an output</li> <li>01: Channel 2 is configured as an input derived from the TI2 signal</li> <li>10: Channel 2 is configured as an input derived from the TI3 signal</li> <li>11: Channel 2 is configured as an input which comes from the TRCED signal derived from the Trigger Controller</li> </ul> <p>Note: The CH2CCS field can be accessed only when the CH2E bit is cleared to 0.</p>

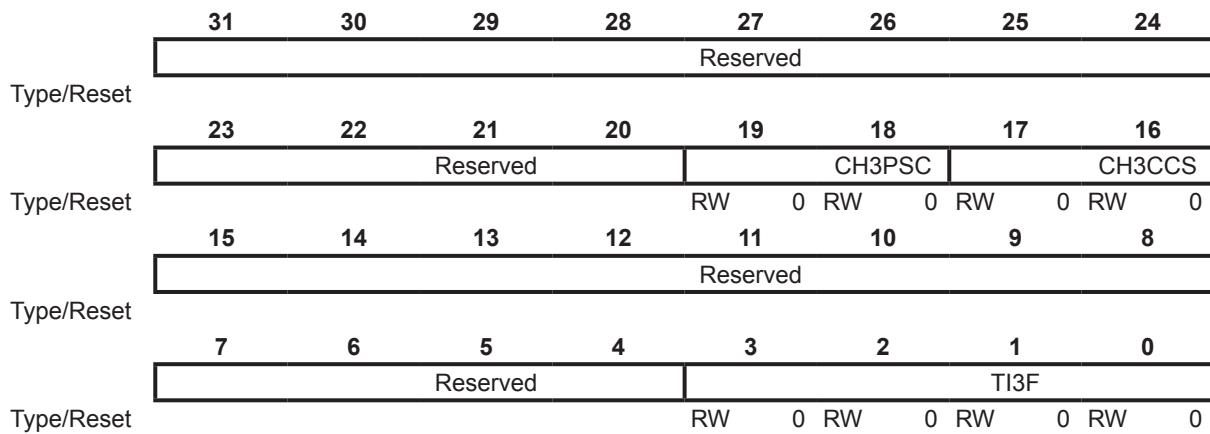
Bits	Field	Descriptions
[3:0]	TI2F	<p>Channel 2 Input Source TI2 Filter Setting</p> <p>These bits define the frequency divide ratio used to sample the TI2 signal. The Digital filter in the MCTM is an N-event counter where N is defined as how many valid transitions are necessary to output a filtered signal.</p> <ul style="list-style-type: none"> <li>0000: No filter, the sampling clock is <math>f_{SYSTEM}</math>.</li> <li>0001: <math>f_{SAMPLING} = f_{CLKIN}</math>, <math>N = 2</math></li> <li>0010: <math>f_{SAMPLING} = f_{CLKIN}</math>, <math>N = 4</math></li> <li>0011: <math>f_{SAMPLING} = f_{CLKIN}</math>, <math>N = 8</math></li> <li>0100: <math>f_{SAMPLING} = f_{DTS} / 2</math>, <math>N = 6</math></li> <li>0101: <math>f_{SAMPLING} = f_{DTS} / 2</math>, <math>N = 8</math></li> <li>0110: <math>f_{SAMPLING} = f_{DTS} / 4</math>, <math>N = 6</math></li> <li>0111: <math>f_{SAMPLING} = f_{DTS} / 4</math>, <math>N = 8</math></li> <li>1000: <math>f_{SAMPLING} = f_{DTS} / 8</math>, <math>N = 6</math></li> <li>1001: <math>f_{SAMPLING} = f_{DTS} / 8</math>, <math>N = 8</math></li> <li>1010: <math>f_{SAMPLING} = f_{DTS} / 16</math>, <math>N = 5</math></li> <li>1011: <math>f_{SAMPLING} = f_{DTS} / 16</math>, <math>N = 6</math></li> <li>1100: <math>f_{SAMPLING} = f_{DTS} / 16</math>, <math>N = 8</math></li> <li>1101: <math>f_{SAMPLING} = f_{DTS} / 32</math>, <math>N = 5</math></li> <li>1110: <math>f_{SAMPLING} = f_{DTS} / 32</math>, <math>N = 6</math></li> <li>1111: <math>f_{SAMPLING} = f_{DTS} / 32</math>, <math>N = 8</math></li> </ul>

## Channel 3 Input Configuration Register – CH3ICFR

This register specifies the channel 3 input mode configuration.

Offset: 0x02C

Reset value: 0x0000\_0000



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"> <li>00: No prescaler, channel 3 capture input signal is chosen for each active event</li> <li>01: Channel 3 Capture input signal is chosen for every 2 events</li> <li>10: Channel 3 Capture input signal is chosen for every 4 events</li> <li>11: Channel 3 Capture input signal is chosen for every 8 events</li> </ul>
[17:16]	CH3CCS	<p>Channel 3 Capture/Compare Selection</p> <ul style="list-style-type: none"> <li>00: Channel 3 is configured as an output</li> <li>01: Channel 3 is configured as an input derived from the TI3 signal</li> <li>10: Channel 3 is configured as an input derived from the TI2 signal</li> <li>11: Channel 3 is configured as an input which comes from the TRCED signal derived from the Trigger Controller</li> </ul> <p>Note: The CH3CCS field can be accessed only when the CH3E bit is cleared to 0.</p>

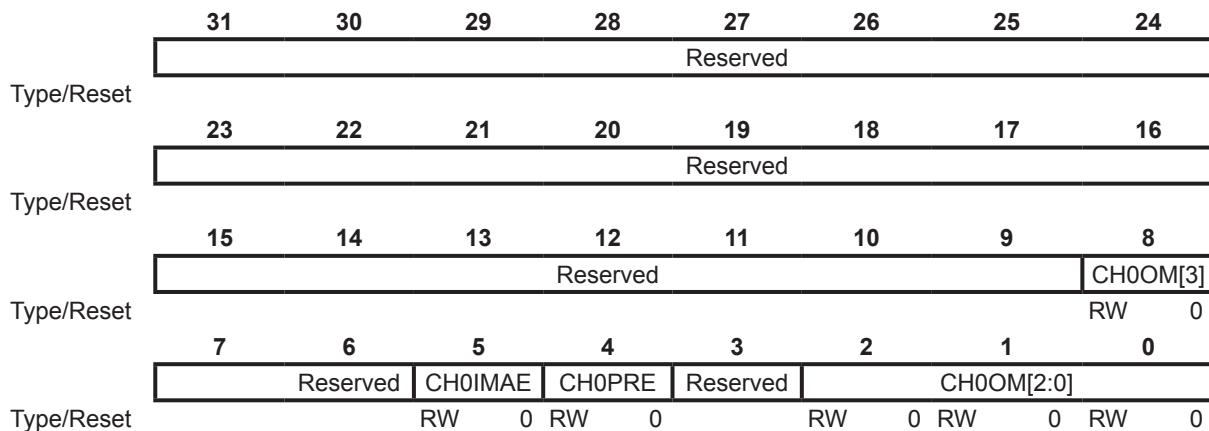
Bits	Field	Descriptions
[3:0]	TI3F	<p>Channel 3 Input Source TI3 Filter Setting</p> <p>These bits define the frequency divide ratio used to sample the TI3 signal. The digital filter in the GPTM is an N-event counter where N is defined as how many valid transitions are necessary to output a filtered signal</p> <ul style="list-style-type: none"> <li>0000: No filter, the sampling clock is <math>f_{SYSTEM}</math>.</li> <li>0001: <math>f_{SAMPLING} = f_{CLKIN}</math>, <math>N = 2</math></li> <li>0010: <math>f_{SAMPLING} = f_{CLKIN}</math>, <math>N = 4</math></li> <li>0011: <math>f_{SAMPLING} = f_{CLKIN}</math>, <math>N = 8</math></li> <li>0100: <math>f_{SAMPLING} = f_{DTS} / 2</math>, <math>N = 6</math></li> <li>0101: <math>f_{SAMPLING} = f_{DTS} / 2</math>, <math>N = 8</math></li> <li>0110: <math>f_{SAMPLING} = f_{DTS} / 4</math>, <math>N = 6</math></li> <li>0111: <math>f_{SAMPLING} = f_{DTS} / 4</math>, <math>N = 8</math></li> <li>1000: <math>f_{SAMPLING} = f_{DTS} / 8</math>, <math>N = 6</math></li> <li>1001: <math>f_{SAMPLING} = f_{DTS} / 8</math>, <math>N = 8</math></li> <li>1010: <math>f_{SAMPLING} = f_{DTS} / 16</math>, <math>N = 5</math></li> <li>1011: <math>f_{SAMPLING} = f_{DTS} / 16</math>, <math>N = 6</math></li> <li>1100: <math>f_{SAMPLING} = f_{DTS} / 16</math>, <math>N = 8</math></li> <li>1101: <math>f_{SAMPLING} = f_{DTS} / 32</math>, <math>N = 5</math></li> <li>1110: <math>f_{SAMPLING} = f_{DTS} / 32</math>, <math>N = 6</math></li> <li>1111: <math>f_{SAMPLING} = f_{DTS} / 32</math>, <math>N = 8</math></li> </ul>

## Channel 0 Output Configuration Register – CH0OCFR

This register specifies the channel 0 output mode configuration.

Offset: 0x040

Reset value: 0x0000\_0000



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> <ul style="list-style-type: none"> <li>0000: No Change</li> <li>0001: Output 0 on compare match</li> <li>0010: Output 1 on compare match</li> <li>0011: Output toggles on compare match</li> <li>0100: Force inactive – CH0OREF is forced to 0</li> <li>0101: Force active – CH0OREF is forced to 1</li> <li>0110: PWM mode 1                     <ul style="list-style-type: none"> <li>- During up-counting, channel 0 has an active level when CNTR &lt; CH0CCR or otherwise has an inactive level.</li> <li>- During down-counting, channel 0 has an inactive level when CNTR &gt; CH0CCR or otherwise has an active level.</li> </ul> </li> <li>0111: PWM mode 2                     <ul style="list-style-type: none"> <li>- During up-counting, channel 0 has an inactive level when CNTR &lt; CH0CCR or otherwise has an active level.</li> <li>- During down-counting, channel 0 has an active level when CNTR &gt; CH0CCR or otherwise has an inactive level.</li> </ul> </li> <li>1110: Asymmetric PWM mode 1                     <ul style="list-style-type: none"> <li>- During up-counting, channel 0 has an active level when CNTR &lt; CH0CCR or otherwise has an inactive level.</li> <li>- During down-counting, channel 0 has an inactive level when CNTR &gt; CH0ACR or otherwise has an active level.</li> </ul> </li> <li>1111: Asymmetric PWM mode 2                     <ul style="list-style-type: none"> <li>- During up-counting, channel 0 has an inactive level when CNTR &lt; CH0CCR or otherwise has an active level.</li> <li>- During down-counting, channel 0 has an active level when CNTR &gt; CH0ACR or otherwise has an inactive level</li> </ul> </li> </ul>

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)

## 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 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> <ul style="list-style-type: none"> <li>0000: No Change</li> <li>0001: Output 0 on compare match</li> <li>0010: Output 1 on compare match</li> <li>0011: Output toggles on compare match</li> <li>0100: Force inactive – CH1OREF is forced to 0</li> <li>0101: Force active – CH1OREF is forced to 1</li> <li>0110: PWM mode 1                     <ul style="list-style-type: none"> <li>- During up-counting, channel 1 has an active level when CNTR &lt; CH1CCR or otherwise has an inactive level.</li> <li>- During down-counting, channel 1 has an inactive level when CNTR &gt; CH1CCR or otherwise has an active level.</li> </ul> </li> <li>0111: PWM mode 2                     <ul style="list-style-type: none"> <li>- During up-counting, channel 1 has an inactive level when CNTR &lt; CH1CCR or otherwise has an active level.</li> <li>- During down-counting, channel 1 has an active level when CNTR &gt; CH1CCR or otherwise has an inactive level.</li> </ul> </li> <li>1110: Asymmetric PWM mode 1                     <ul style="list-style-type: none"> <li>- During up-counting, channel 1 has an active level when CNTR &lt; CH1CCR or otherwise has an inactive level.</li> <li>- During down-counting, channel 1 has an inactive level when CNTR &gt; CH1ACR or otherwise has an active level.</li> </ul> </li> <li>1111: Asymmetric PWM mode 2                     <ul style="list-style-type: none"> <li>- During up-counting, channel 1 has an inactive level when CNTR &lt; CH1CCR or otherwise has an active level.</li> <li>- During down-counting, channel 1 has an active level when CNTR &gt; CH1ACR or otherwise has an inactive level</li> </ul> </li> </ul>

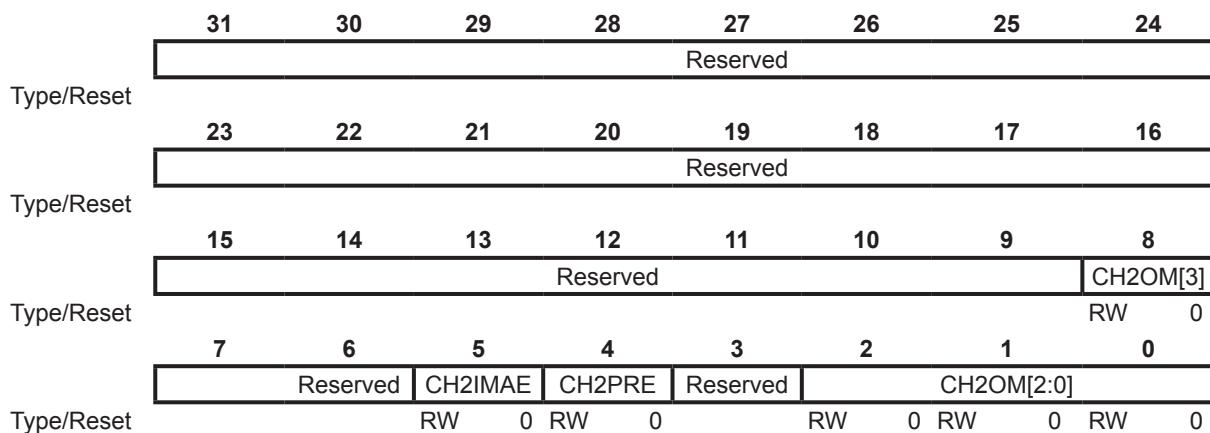
Note: When channel 1 is used as asymmetric PWM output mode, the Counter Mode Selection bit in Counter Configuration Register must be configured as Center-aligned Counting mode (CMSEL = 01/02/03)

## 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	Channel 2 Immediate Active Enable 0: No action 1: Single pulse Immediate Active Mode enabled 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. The effective duration ends automatically at the next overflow or underflow event. Note: The CH2IMAE bit is available only if the channel 2 is configured to be operated in PWM mode 1 or PWM mode 2.
[4]	CH2PRE	Channel 2 Capture/Compare Register (CH2CCR) Preload Enable 0: CH2CCR preload function is disabled. The CH2CCR register can be immediately assigned a new value when the CH2PRE bit is cleared to 0 and the updated CH2CCR value is used immediately. 1: CH2CCR preload function is enabled The new CH2CCR value will not be transferred to its shadow register until an update event 1 occurs.

Bits	Field	Descriptions
[8][2:0]	CH2OM[3:0]	<p>Channel 2 Output Mode Setting</p> <p>These bits define the functional types of the output reference signal CH2OREF.</p> <ul style="list-style-type: none"> <li>0000: No Change</li> <li>0001: Output 0 on compare match</li> <li>0010: Output 1 on compare match</li> <li>0011: Output toggles on compare match</li> <li>0100: Force inactive – CH2OREF is forced to 0</li> <li>0101: Force active – CH2OREF is forced to 1</li> <li>0110: PWM mode 1                     <ul style="list-style-type: none"> <li>- During up-counting, channel 2 has an active level when CNTR &lt; CH2CCR or otherwise has an inactive level.</li> <li>- During down-counting, channel 2 has an inactive level when CNTR &gt; CH2CCR or otherwise has an active level.</li> </ul> </li> <li>0111: PWM mode 2                     <ul style="list-style-type: none"> <li>- During up-counting, channel 2 has an inactive level when CNTR &lt; CH2CCR or otherwise has an active level.</li> <li>- During down-counting, channel 2 has an active level when CNTR &gt; CH2CCR or otherwise has an inactive level.</li> </ul> </li> <li>1110: Asymmetric PWM mode 1                     <ul style="list-style-type: none"> <li>- During up-counting, channel 2 has an active level when CNTR &lt; CH2CCR or otherwise has an inactive level.</li> <li>- During down-counting, channel 2 has an inactive level when CNTR &gt; CH2ACR or otherwise has an active level.</li> </ul> </li> <li>1111: Asymmetric PWM mode 2                     <ul style="list-style-type: none"> <li>- During up-counting, channel 2 has an inactive level when CNTR &lt; CH2CCR or otherwise has an active level.</li> <li>- During down-counting, channel 2 has an active level when CNTR &gt; CH2ACR or otherwise has an inactive level</li> </ul> </li> </ul>

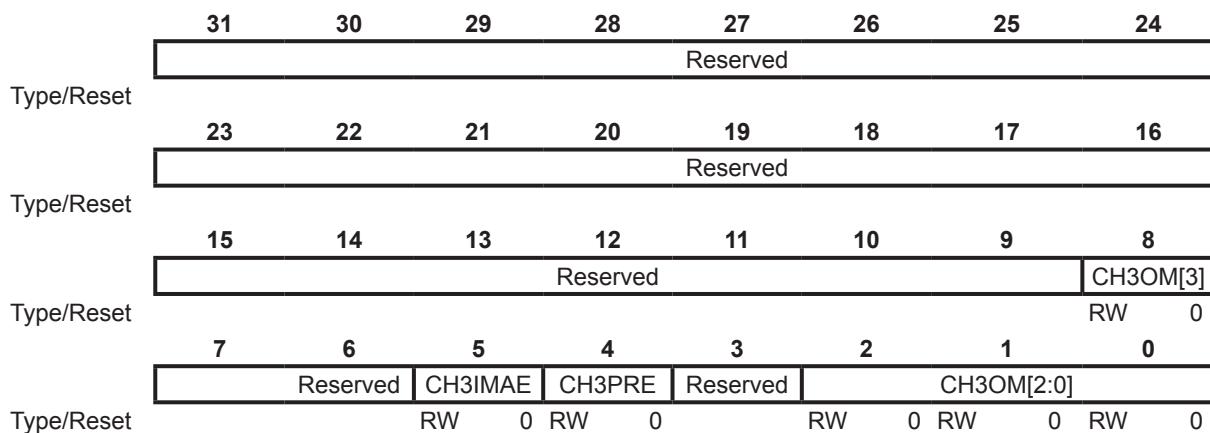
Note: When channel 2 is used as asymmetric PWM output mode, the Counter Mode Selection bit in Counter Configuration Register must be configured as Center-aligned Counting mode (CMSEL = 01/02/03)

## 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	Channel 3 Immediate Active Enable 0: No action 1: Single pulse Immediate Active Mode enabled 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. The effective duration ends automatically at the next overflow or underflow event. Note: The CH3IMAE bit is available only if channel 3 is configured to be operated in PWM mode 1 or PWM mode 2.
[4]	CH3PRE	Channel 3 Capture/Compare Register (CH3CCR) Preload Enable 0: CH3CCR preload function is disabled. The CH3CCR register can be immediately assigned a new value when the CH3PRE bit is cleared to 0 and the updated CH3CCR value is used immediately. 1: CH3CCR preload function is enabled The new CH3CCR value will not be transferred to its shadow register until an update event 1 occurs.

Bits	Field	Descriptions
[8][2:0]	CH3OM[3:0]	<p>Channel 3 Output Mode Setting</p> <p>These bits define the functional types of the output reference signal CH3OREF</p> <ul style="list-style-type: none"> <li>0000: No Change</li> <li>0001: Output 0 on compare match</li> <li>0010: Output 1 on compare match</li> <li>0011: Output toggles on compare match</li> <li>0100: Force inactive – CH3OREF is forced to 0</li> <li>0101: Force active – CH3OREF is forced to 1</li> <li>0110: PWM mode 1                     <ul style="list-style-type: none"> <li>- During up-counting, channel 3 has an active level when CNTR &lt; CH3CCR or otherwise has an inactive level.</li> <li>- During down-counting, channel 3 has an inactive level when CNTR &gt; CH3CCR or otherwise has an active level.</li> </ul> </li> <li>0111: PWM mode 2                     <ul style="list-style-type: none"> <li>- During up-counting, channel 3 has an inactive level when CNTR &lt; CH3CCR or otherwise has an active level.</li> <li>- During down-counting, channel 3 has an active level when CNTR &gt; CH3CCR or otherwise has an inactive level</li> </ul> </li> <li>1110: Asymmetric PWM mode 1                     <ul style="list-style-type: none"> <li>- During up-counting, channel 3 has an active level when CNTR &lt; CH3CCR or otherwise has an inactive level.</li> <li>- During down-counting, channel 3 has an inactive level when CNTR &gt; CH3ACR or otherwise has an active level.</li> </ul> </li> <li>1111: Asymmetric PWM mode 2                     <ul style="list-style-type: none"> <li>- During up-counting, channel 3 has an inactive level when CNTR &lt; CH3CCR or otherwise has an active level.</li> <li>- During down-counting, channel 3 has an active level when CNTR &gt; CH3ACR or otherwise has an inactive level</li> </ul> </li> </ul>

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)

## 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	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[6]	CH3E	<p>Channel 3 Capture/Compare Enable</p> <ul style="list-style-type: none"> <li>- Channel 3 is configured as an input (CH3CCS = 0x01/0x02/0x03)           <ul style="list-style-type: none"> <li>0: Input Capture Mode disabled</li> <li>1: Input Capture Mode enabled</li> </ul> </li> <li>- Channel 3 is configured as an output (CH3CCS = 0x00)           <ul style="list-style-type: none"> <li>0: Off – Channel 3 output signal CH3O is not active</li> <li>1: On – Channel 3 output signal CH3O is generated on the corresponding output pin depending on the condition of the CHMOE, CHOSSI, CHOSSR and CH3OIS bits.</li> </ul> </li> </ul>
[5]	CH2NE	<p>Channel 2 Capture/Compare Complementary Enable</p> <ul style="list-style-type: none"> <li>0: Off – Channel 2 complementary output CH2NO is not active. The CH2NO level is then determined by the CHMOE, CHOSSI, CHOSSR, CH2OIS, CH2OISN and CH2E bits.</li> <li>1: On – Channel 2 complementary output CH2NO is generated on the corresponding output pin depending on the condition of the CHMOE, CHOSSI, CHOSSR, CH2OIS, CH2OISN and CH2E bits.</li> </ul>
[4]	CH2E	<p>Channel 2 Capture/Compare Enable</p> <ul style="list-style-type: none"> <li>- Channel 2 is configured as an input (CH2CCS = 0x01/0x02/0x03)           <ul style="list-style-type: none"> <li>0: Input Capture Mode disabled</li> <li>1: Input Capture Mode enabled</li> </ul> </li> <li>- Channel 2 is configured as an output (CH2CCS = 0x00)           <ul style="list-style-type: none"> <li>0: Off – Channel 2 output signal CH2O is not active. The CH2O level is then determined by the condition of the CHMOE, CHOSSI, CHOSSR, CH2OIS, CH2OISN and CH2NE bits.</li> <li>1: On – Channel 2 output signal CH2O is generated on the corresponding output pin determined by the condition of the CHMOE, CHOSSI, CHOSSR, CH2OIS, CH2OISN and CH2NE bits.</li> </ul> </li> </ul>

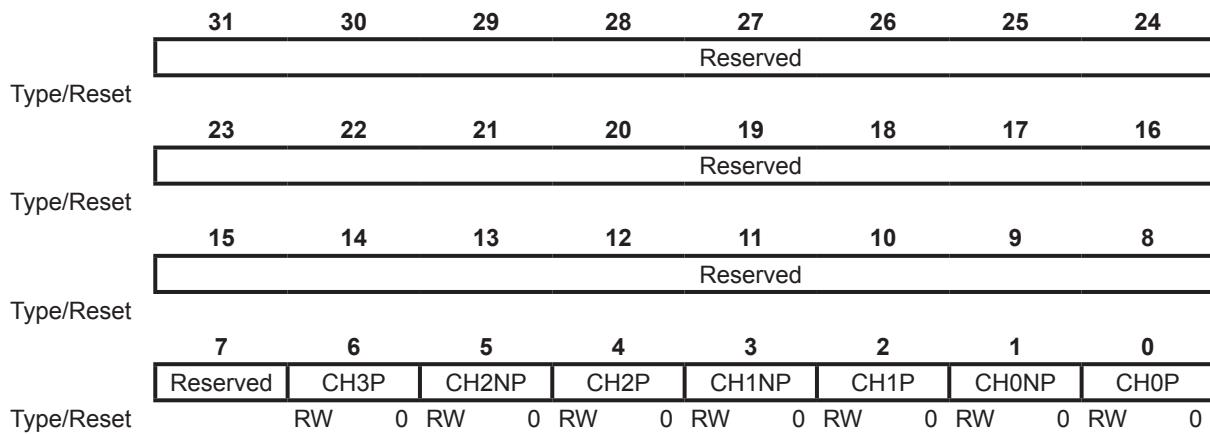
Bits	Field	Descriptions
[3]	CH1NE	<p>Channel 1 Capture/Compare Complementary Enable</p> <p>0: Off – Channel 1 complementary output CH1NO is not active. The CH1NO level is then determined by the condition of the CHMOE, CHOSSI, CHOSSR, CH1OIS, CH1OISN and CH1E bits.</p> <p>1: On – Channel 1 complementary 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> <ul style="list-style-type: none"> <li>- Channel 1 is configured as an input (CH1CCS = 0x01/0x02/0x03)                     <ul style="list-style-type: none"> <li>0: Input Capture Mode disabled</li> <li>1: Input Capture Mode enabled</li> </ul> </li> <li>- Channel 1 is configured as an output (CH1CCS = 0x00)                     <ul style="list-style-type: none"> <li>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.</li> <li>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.</li> </ul> </li> </ul>
[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> <ul style="list-style-type: none"> <li>- Channel 0 is configured as an input (CH0CCS = 0x01/0x02/0x03)                     <ul style="list-style-type: none"> <li>0: Input Capture Mode disabled</li> <li>1: Input Capture Mode enabled</li> </ul> </li> <li>- Channel 0 is configured as an output (CH0CCS = 0x00)                     <ul style="list-style-type: none"> <li>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.</li> <li>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.</li> </ul> </li> </ul>

## 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 active high 1: Channel 3 Output active low
[5]	CH2NP	Channel 2 Capture/Compare Complementary Polarity 0: Channel 2 Output active high. 1: Channel 2 Output 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 active high 1: Channel 2 Output active low
[3]	CH1NP	Channel 1 Capture/Compare Complementary Polarity 0: Channel 1 Output active high. 1: Channel 1 Output 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 active high 1: Channel 1 Output active low
[1]	CH0NP	Channel 0 Capture/Compare Complementary Polarity 0: Channel 0 Output active high. 1: Channel 0 Output active low.

<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>
[0]	CH0P	<p>Channel 0 Capture/Compare Polarity</p> <ul style="list-style-type: none"><li>- When Channel 0 is configured as an input<ul style="list-style-type: none"><li>0: capture event occurs on a Channel 0 rising edge</li><li>1: capture event occurs on a Channel 0 falling edge</li></ul></li><li>- When Channel 0 is configured as an output (CH0CCS = 0x00)<ul style="list-style-type: none"><li>0: Channel 0 Output active high</li><li>1: Channel 0 Output active low</li></ul></li></ul>

## Channel Break Configuration Register – CHBRKCFR

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

Offset: 0x06C

Reset value: 0x0000\_0000

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

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

Bits	Field	Descriptions
[31:24]	CHDTG	Channel Dead Time Duration Definition CHDTG[7:5]=0xx: Channel Dead Time = CHDTG [7:0] × t <sub>dtg</sub> , with t <sub>dtg</sub> = t <sub>DTS</sub> CHDTG[7:5]=10x: Channel Dead Time = (64+CHDTG [5:0])×t <sub>dtg</sub> , with t <sub>dtg</sub> = 2×t <sub>DTS</sub> CHDTG[7:5]=110: Channel Dead Time = (32+CHDTG [4:0])×t <sub>dtg</sub> , with t <sub>dtg</sub> = 8×t <sub>DTS</sub> CHDTG[7:5]=111: Channel Dead Time = (32+CHDTG [4:0])×t <sub>dtg</sub> , with t <sub>dtg</sub> = 16×t <sub>DTS</sub>
[21]	CHOSSR	Channel Off State (CHxE, 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]	CHOSSI	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

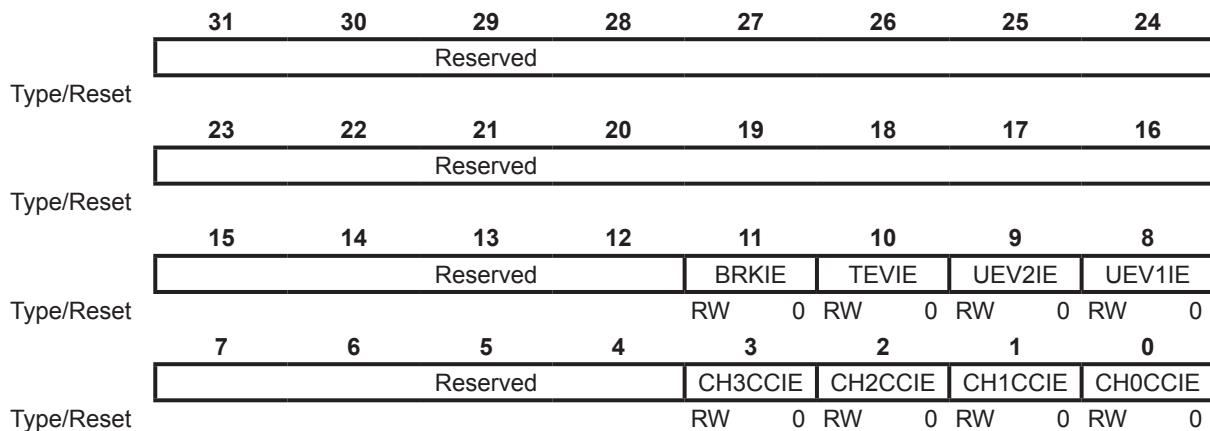
Bits	Field	Descriptions
[11:8]	BKF	<p>Break Input Filter Setting</p> <p>These bits define the frequency ratio used to sample the MT_BRK signal. The digital filter in the MCTM is an N-event counter where N is defined as how many valid transitions are necessary to output a filtered signal.</p> <ul style="list-style-type: none"> <li>0000: No filter – don't need sample clock.</li> <li>0001: <math>f_{SAMPLING} = f_{CLKIN}</math>, N = 2.</li> <li>0010: <math>f_{SAMPLING} = f_{CLKIN}</math>, N = 4.</li> <li>0011: <math>f_{SAMPLING} = f_{CLKIN}</math>, N = 8.</li> <li>0100: <math>f_{SAMPLING} = f_{DTS}/2</math>, N = 6.</li> <li>0101: <math>f_{SAMPLING} = f_{DTS}/2</math>, N = 8.</li> <li>0110: <math>f_{SAMPLING} = f_{DTS}/4</math>, N = 6.</li> <li>0111: <math>f_{SAMPLING} = f_{DTS}/4</math>, N = 8.</li> <li>1000: <math>f_{SAMPLING} = f_{DTS}/8</math>, N = 6.</li> <li>1001: <math>f_{SAMPLING} = f_{DTS}/8</math>, N = 8.</li> <li>1010: <math>f_{SAMPLING} = f_{DTS}/16</math>, N = 5.</li> <li>1011: <math>f_{SAMPLING} = f_{DTS}/16</math>, N = 6.</li> <li>1100: <math>f_{SAMPLING} = f_{DTS}/16</math>, N = 8.</li> <li>1101: <math>f_{SAMPLING} = f_{DTS}/32</math>, N = 5.</li> <li>1110: <math>f_{SAMPLING} = f_{DTS}/32</math>, N = 6.</li> <li>1111: <math>f_{SAMPLING} = f_{DTS}/32</math>, N = 8.</li> </ul>
[5]	CHAOE	<p>Channel Automatic Output Enable</p> <ul style="list-style-type: none"> <li>0: CHMOE can be set only by software.</li> <li>1: CHMOE can be set by software or automatically by an update event</li> </ul>
[4]	CHMOE	<p>Channel Main Output Enable</p> <p>Cleared asynchronously by hardware on a break event occurrence.</p> <ul style="list-style-type: none"> <li>0: MTn_CHxO and MTn_CHxNO are disabled or forced to idle states.</li> <li>1: MTn_CHxO and MTn_CHxNO are enabled if the enable bits (CHxE, CHxNE) are set.</li> </ul>
[1]	BKP	<p>Break Input Polarity.</p> <ul style="list-style-type: none"> <li>0: Break input active low</li> <li>1: Break input active high</li> </ul>
[0]	BKE	<p>Break Enable</p> <ul style="list-style-type: none"> <li>0: Break inputs disabled</li> <li>1: Break inputs enabled</li> </ul>

## Timer Interrupt Control Register – DICTR

This register contains the timer interrupt enable control bits.

Offset: 0x074

Reset value: 0x0000\_0000



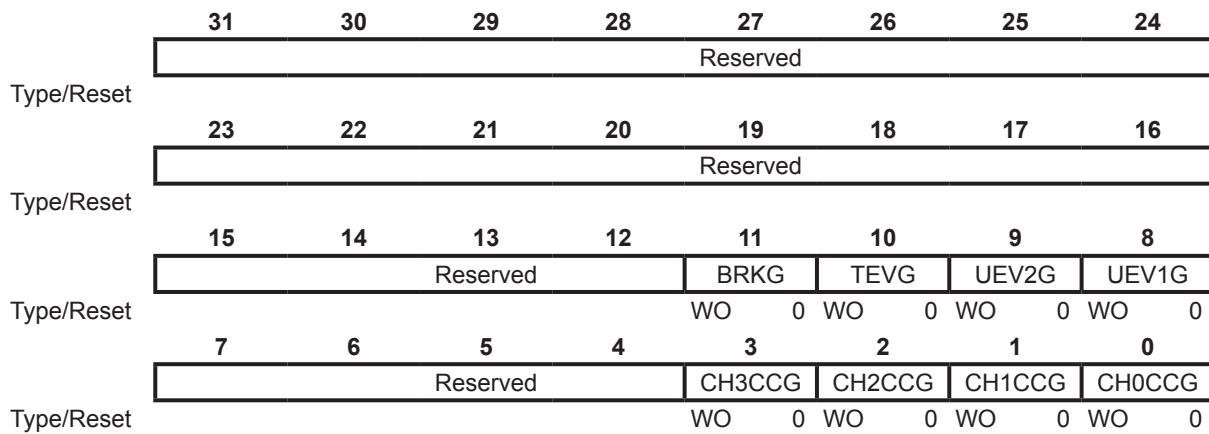
Bits	Field	Descriptions
[11]	BRKIE	Break event Interrupt Enable 0: Break event interrupt disabled 1: Break event interrupt enabled
[10]	TEVIE	Trigger event Interrupt Enable 0: Trigger event interrupt disabled 1: Trigger event interrupt enabled
[9]	UEV2IE	Update event 2 Interrupt Enable 0: Update event 2 interrupt disabled 1: Update event 2 interrupt enabled
[8]	UEV1IE	Update event 1 Interrupt Enable 0: Update event 1 interrupt disabled 1: Update event 1 interrupt enabled
[3]	CH3CCIE	Channel 3 Capture/Compare Interrupt Enable 0: Channel 3 interrupt disabled 1: Channel 3 interrupt enabled
[2]	CH2CCIE	Channel 2 Capture/Compare Interrupt Enable 0: Channel 2 interrupt disabled 1: Channel 2 interrupt enabled
[1]	CH1CCIE	Channel 1 Capture/Compare Interrupt Enable 0: Channel 1 interrupt disabled 1: Channel 1 interrupt enabled
[0]	CH0CCIE	Channel 0 Capture/Compare Interrupt Enable 0: Channel 0 interrupt disabled 1: Channel 0 interrupt enabled

## Timer Event Generator Register – EVGR

This register contains the software event generation bits.

Offset: 0x078

Reset value: 0x0000\_0000



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.

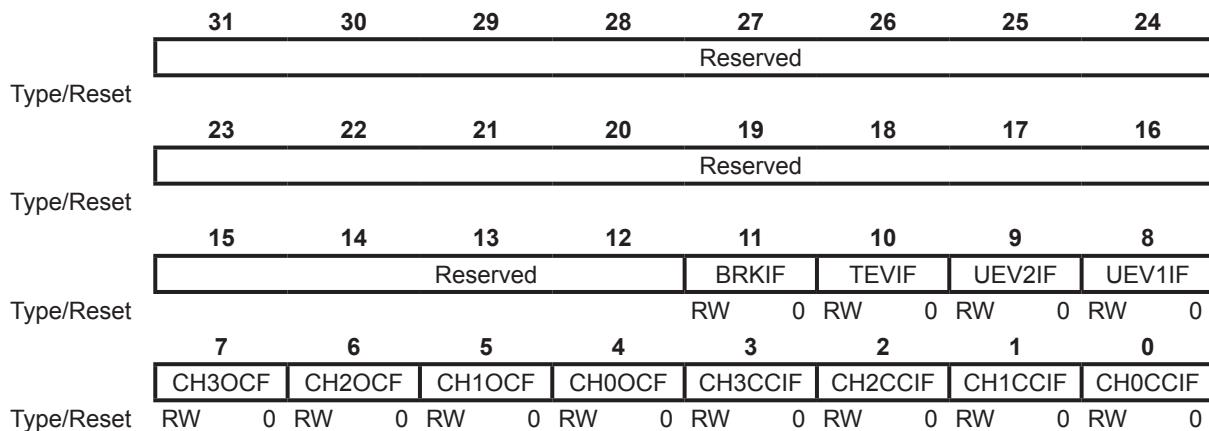
<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>
[3]	CH3CCG	Channel 3 Capture/Compare Generation A Channel 3 capture/compare event can be generated by setting this bit. It is cleared by hardware automatically. 0: No action 1: Capture/compare event is generated on channel 3 If Channel 3 is configured as an 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.
[2]	CH2CCG	Channel 2 Capture/Compare Generation A Channel 2 capture/compare event can be generated by setting this bit. It is cleared by hardware automatically. 0: No action 1: Capture/compare event is generated on channel 2 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.
[1]	CH1CCG	Channel 1 Capture/Compare Generation A Channel 1 capture/compare event can be generated by setting this bit. It is cleared by hardware automatically. 0: No action 1: Capture/compare event is generated on channel 1 If Channel 1 is configured as an input, the counter value is captured into the CH1CCR register and then the CH1CCIF bit is set. If Channel 1 is configured as an output, the CH1CCIF bit is set.
[0]	CH0CCG	Channel 0 Capture/Compare Generation A Channel 0 capture/compare event can be generated by setting this bit. It is cleared by hardware automatically. 0: No action 1: Capture/compare event is generated on channel 0 If Channel 0 is configured as an input, the counter value is captured into the CH0CCR register and then the CH0CCIF bit is set. If Channel 0 is configured as an output, the CH0CCIF bit is set.

## Timer Interrupt Status Register – INTSR

This register stores the timer interrupt status.

Offset: 0x07C

Reset value: 0x0000\_0000



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 UEVDIS=0 - A STI rising edge is received in slave restart mode with UEVDIS=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.

<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>
[6]	CH2OCF	<p>Channel 2 Over-capture Flag</p> <p>This flag is set by hardware and cleared by software.</p> <ul style="list-style-type: none"> <li>0: No over-capture event is detected</li> <li>1: Capture event occurs again when the CH2CCIF bit is already set and it is not cleared yet by software.</li> </ul>
[5]	CH1OCF	<p>Channel 1 Over-capture Flag</p> <p>This flag is set by hardware and cleared by software.</p> <ul style="list-style-type: none"> <li>0: No over-capture event is detected</li> <li>1: Capture event occurs again when the CH1CCIF bit is already set and it is not cleared yet by software.</li> </ul>
[4]	CH0OCF	<p>Channel 0 Over-capture Flag</p> <p>This flag is set by hardware and cleared by software.</p> <ul style="list-style-type: none"> <li>0: No over-capture event is detected</li> <li>1: Capture event occurs again when the CH0CCIF bit is already set and it is not yet cleared by software</li> </ul>
[3]	CH3CCIF	<p>Channel 3 Capture/Compare Interrupt Flag</p> <ul style="list-style-type: none"> <li>- Channel 3 is configured as an output           <ul style="list-style-type: none"> <li>0: No match event occurred</li> <li>1: The contents of the counter CNTR have matched the contents of the CH3CCR register.</li> </ul> <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> </li> <li>- Channel 3 is configured as an input           <ul style="list-style-type: none"> <li>0: No input capture occurred</li> <li>1: Input capture occurred</li> </ul> <p>This bit is set by hardware when a capture event occurs. It is cleared by software or by reading the CH3CCR register.</p> </li> </ul>
[2]	CH2CCIF	<p>Channel 2 Capture/Compare Interrupt Flag</p> <ul style="list-style-type: none"> <li>- Channel 2 is configured as an output           <ul style="list-style-type: none"> <li>0: No match event occurred</li> <li>1: The contents of the counter CNTR have matched the contents of the CH2CCR register</li> </ul> <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> </li> <li>- Channel 2 is configured as an input           <ul style="list-style-type: none"> <li>0: No input capture occurred</li> <li>1: Input capture occurred.</li> </ul> <p>This bit is set by hardware on a capture event. It is cleared by software or by reading the CH2CCR register.</p> </li> </ul>
[1]	CH1CCIF	<p>Channel 1 Capture/Compare Interrupt Flag</p> <ul style="list-style-type: none"> <li>- Channel 1 is configured as an output           <ul style="list-style-type: none"> <li>0: No match event occurred</li> <li>1: The contents of the counter CNTR have matched the contents of the CH1CCR register</li> </ul> <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> </li> <li>- Channel 1 is configured as an input           <ul style="list-style-type: none"> <li>0: No input capture occurred</li> <li>1: Input capture occurred</li> </ul> <p>This bit is set by hardware on a capture event. It is cleared by software or by reading the CH1CCR register.</p> </li> </ul>

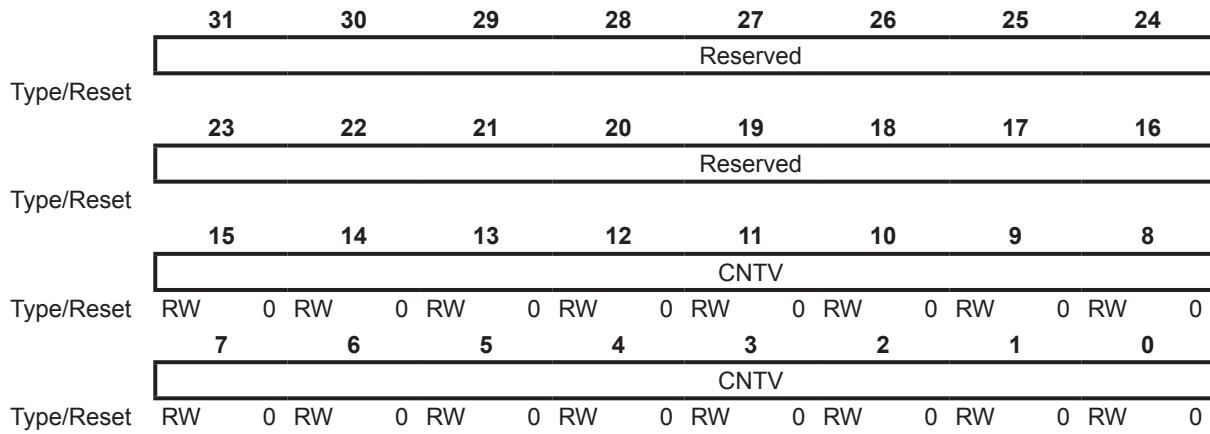
Bits	Field	Descriptions
[0]	CH0CCIF	<p>Channel 0 Capture/Compare Interrupt Flag</p> <ul style="list-style-type: none"><li>- Channel 0 is configured as an output<ul style="list-style-type: none"><li>0: No match event occurs</li><li>1: The contents of the counter CNTR have matched the content of the CH0CCR register</li></ul></li><li>This 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.</li><li>- Channel 0 is configured as an input<ul style="list-style-type: none"><li>0: No input capture occurred</li><li>1: Input capture occurred</li></ul></li><li>This bit is set by hardware on a capture event. It is cleared by software or by reading the CH0CCR register.</li></ul>

## Timer Counter Register – CNTR

This register stores the timer counter value.

Offset: 0x080

Reset value: 0x0000\_0000



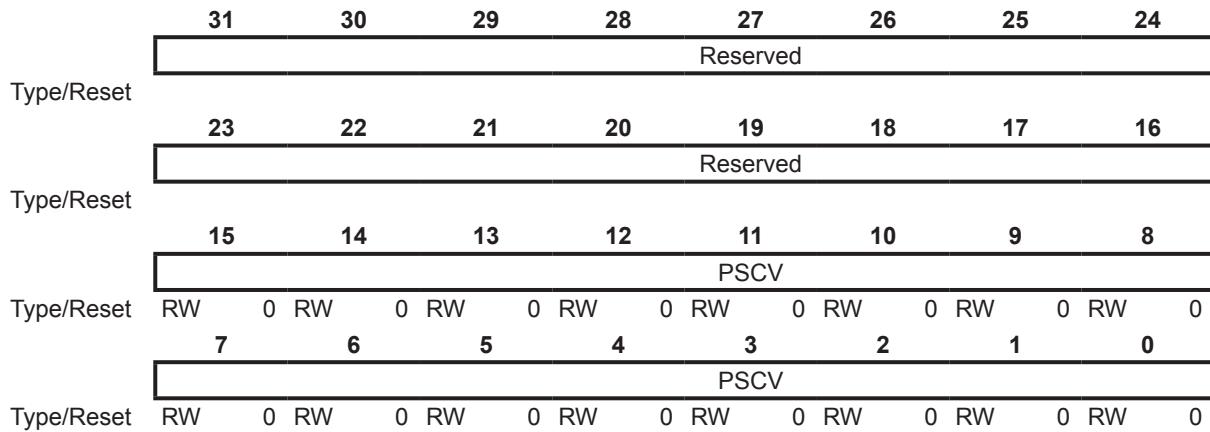
Bits	Field	Descriptions
[15:0]	CNTV	Counter Value.

## Timer Prescaler Register – PSCR

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

Offset: 0x084

Reset value: 0x0000\_0000



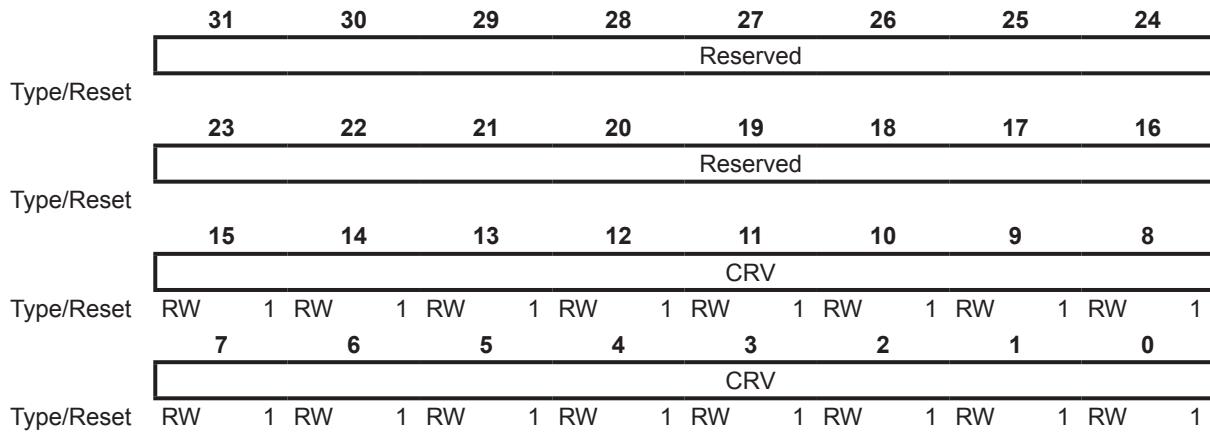
Bits	Field	Descriptions
[15:0]	PSCV	<p>Prescaler Value</p> <p>These bits are used to specify the prescaler value to generate the counter clock frequency <math>f_{CK\_CNT}</math>.</p> $f_{CK\_CNT} = \frac{f_{CK\_PSC}}{PSCV[15:0] + 1}$ , where the $f_{CK\_PSC}$ is the prescaler input clock source.

## 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.

## Timer Repetition Register – REPR

This register specifies the timer repetition counter value.

Offset: 0x08C

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	RW	0	RW	0	RW	0	RW	0
					REPV			

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	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	RW	0	RW	0	RW	0	RW	0	
	7	6	5	4	3	2	1	0	
Type/Reset	CH0CCV								
	RW	0	RW	0	RW	0	RW	0	

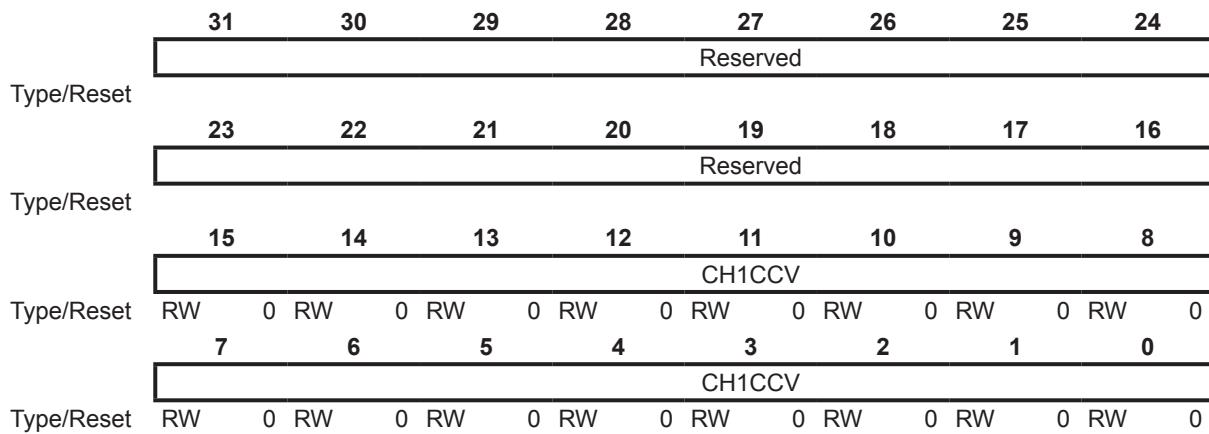
Bits	Field	Descriptions
[15:0]	CH0CCV	<p>Channel 0 Capture/Compare Value</p> <ul style="list-style-type: none"> <li>- When Channel 0 is configured as an output</li> <li>The CH0CCR value is compared with the counter value and the comparison result is used to trigger the CH0OREF output signal.</li> <li>-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

## Channel 1 Capture/Compare Register – CH1CCR

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

Offset: 0x094

Reset value: 0x0000\_0000



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

## Channel 2 Capture/Compare Register – CH2CCR

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

Offset: 0x098

Reset value: 0x0000\_0000

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

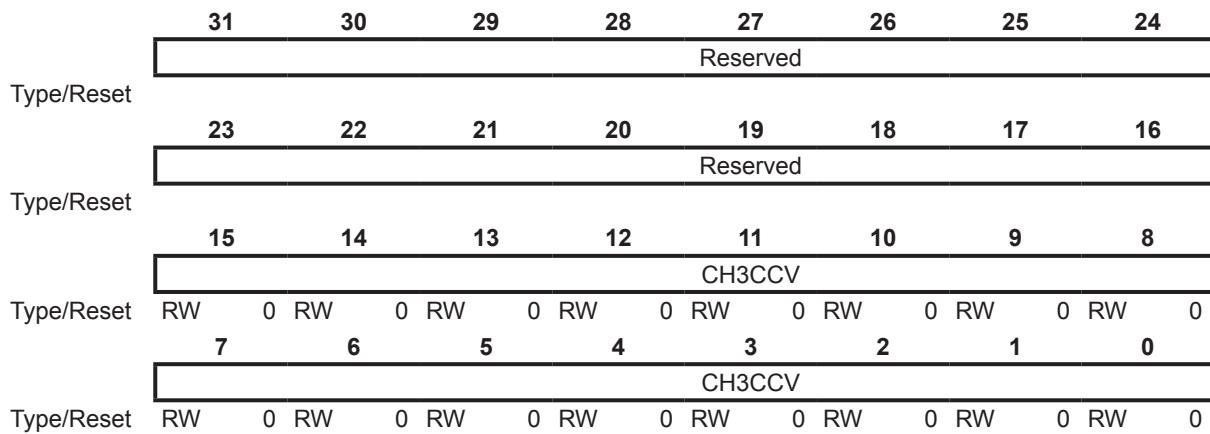
Bits	Field	Descriptions
[15:0]	CH2CCV	<p>Channel 2 Capture/Compare Value</p> <ul style="list-style-type: none"> <li>- When Channel 2 is configured as an output</li> <li>The CH2CCR value is compared with the counter value and the comparison result is used to trigger the CH2OREF output signal.</li> <li>-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

## Channel 3 Capture/Compare Register – CH3CCR

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

Offset: 0x09C

Reset value: 0x0000\_0000



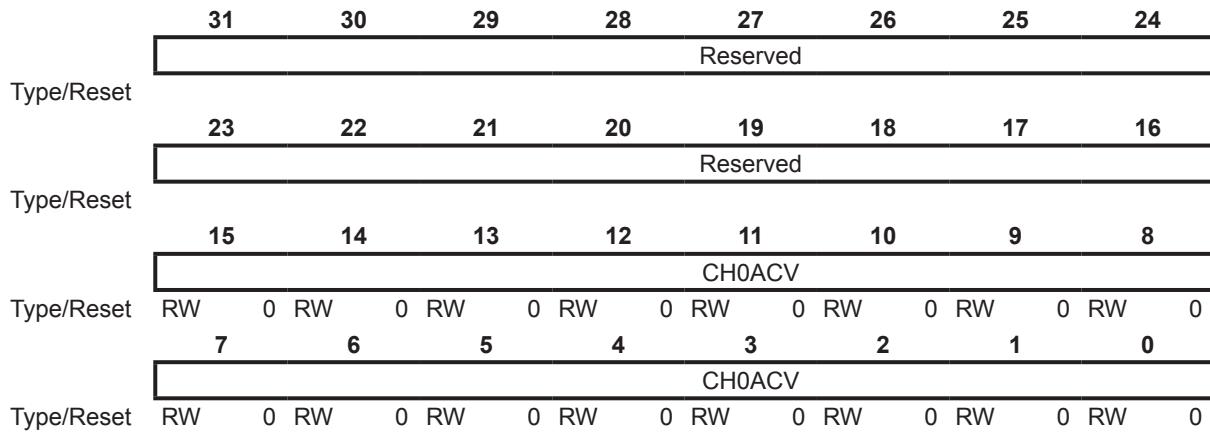
Bits	Field	Descriptions
[15:0]	CH3CCV	<p>Channel 3 Capture/Compare Value</p> <ul style="list-style-type: none"> <li>- When Channel 3 is configured as an output</li> <li>The CH3CCR value is compared with the counter value and the comparison result is used to trigger the CH3OREF output signal.</li> <li>-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

## Channel 0 Asymmetric Compare Register – CH0ACR

This register specifies the timer channel 0 asymmetric compare value.

Offset: 0x0A0

Reset value: 0x0000\_0000



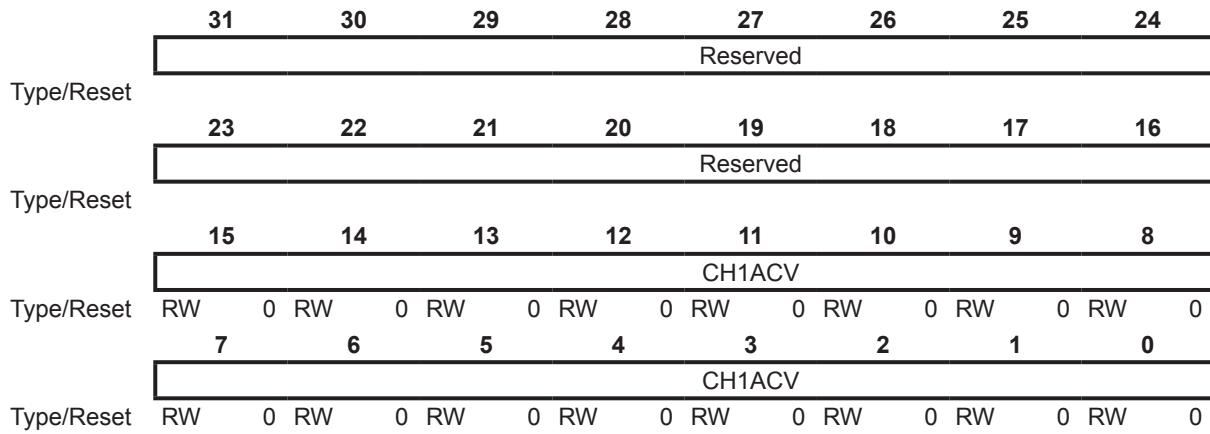
Bits	Field	Descriptions
[15:0]	CH0ACV	Channel 0 Asymmetric Compare Value When channel 0 is configured as asymmetric PWM mode and the counter is counting down, the value written into this register will be compared to the counter.

## Channel 1 Asymmetric Compare Register – CH1ACR

This register specifies the timer channel 1 asymmetric compare value.

Offset: 0x0A4

Reset value: 0x0000\_0000



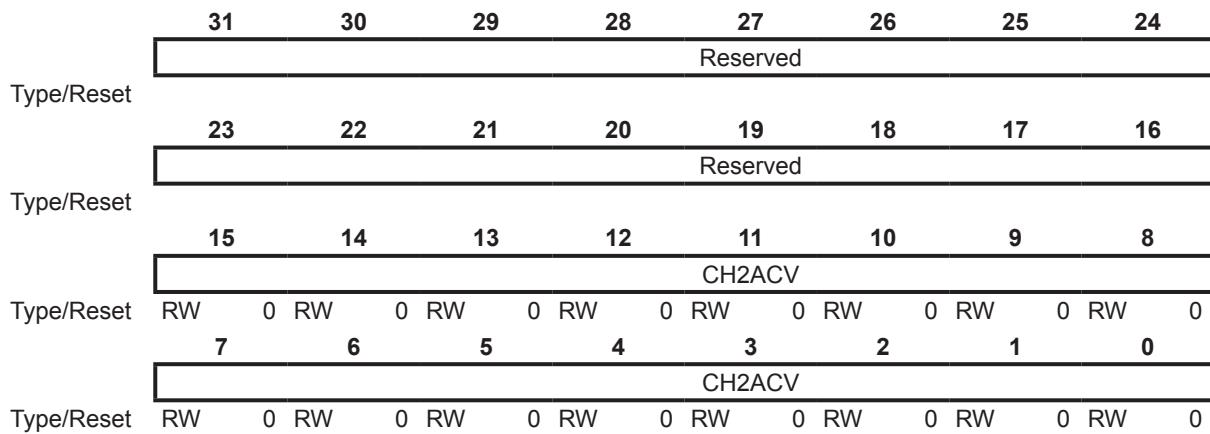
Bits	Field	Descriptions
[15:0]	CH1ACV	Channel 1 Asymmetric Compare Value When channel 1 is configured as asymmetric PWM mode and the counter is counting down, the value written into this register will be compared to the counter.

## Channel 2 Asymmetric Compare Register – CH2ACR

This register specifies the timer channel 2 asymmetric compare value.

Offset: 0x0A8

Reset value: 0x0000\_0000



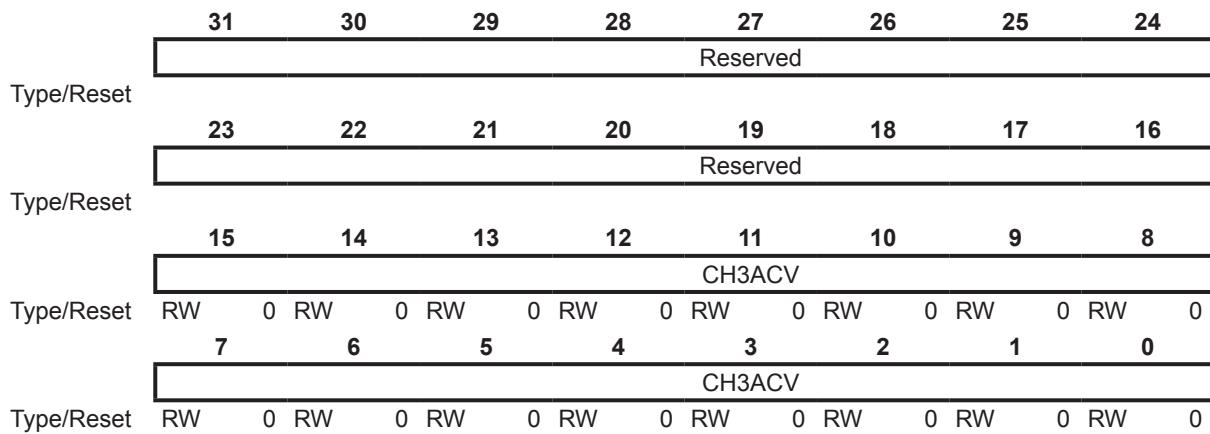
Bits	Field	Descriptions
[15:0]	CH2ACV	Channel 2 Asymmetric Compare Value When channel 2 is configured as asymmetric PWM mode and the counter is counting down, the value written into this register will be compared to the counter.

## 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.

# 16 Single-Channel Timer (SCTM)

## Introduction

The Single-Channel Timer consists of one 16-bit up-counter, one 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.

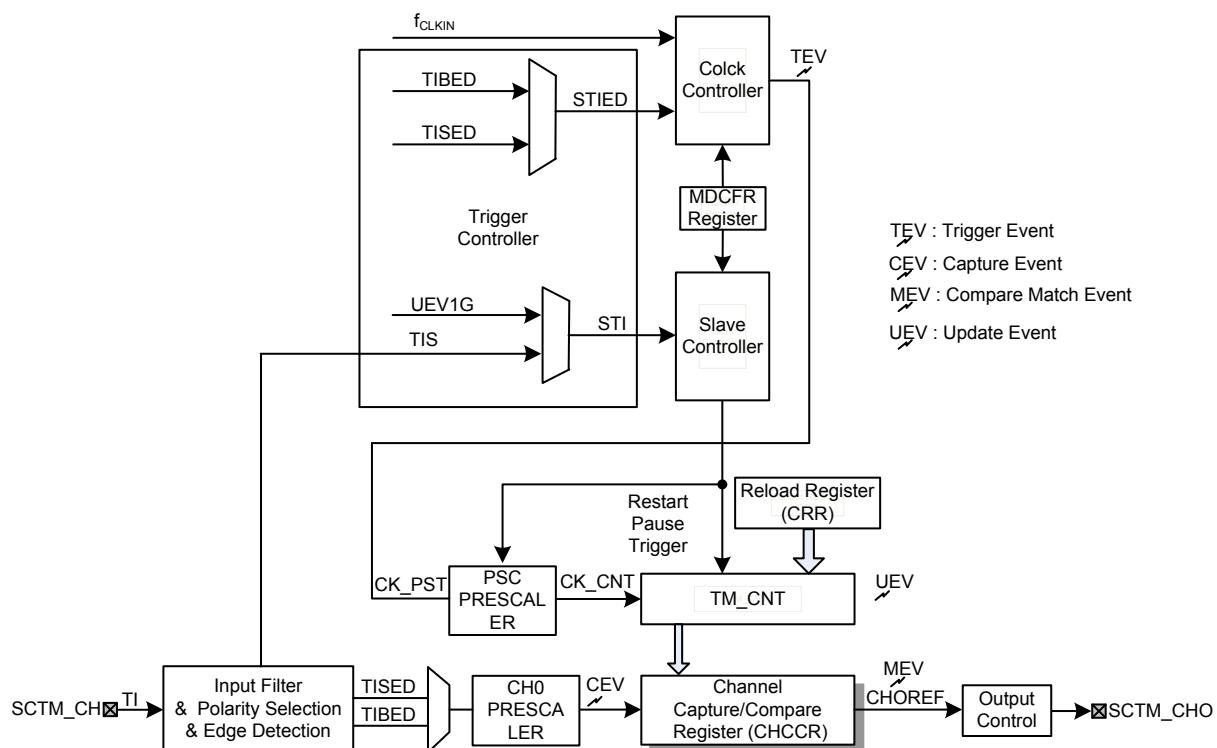


Figure 111. SCTM Block Diagram

## 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.

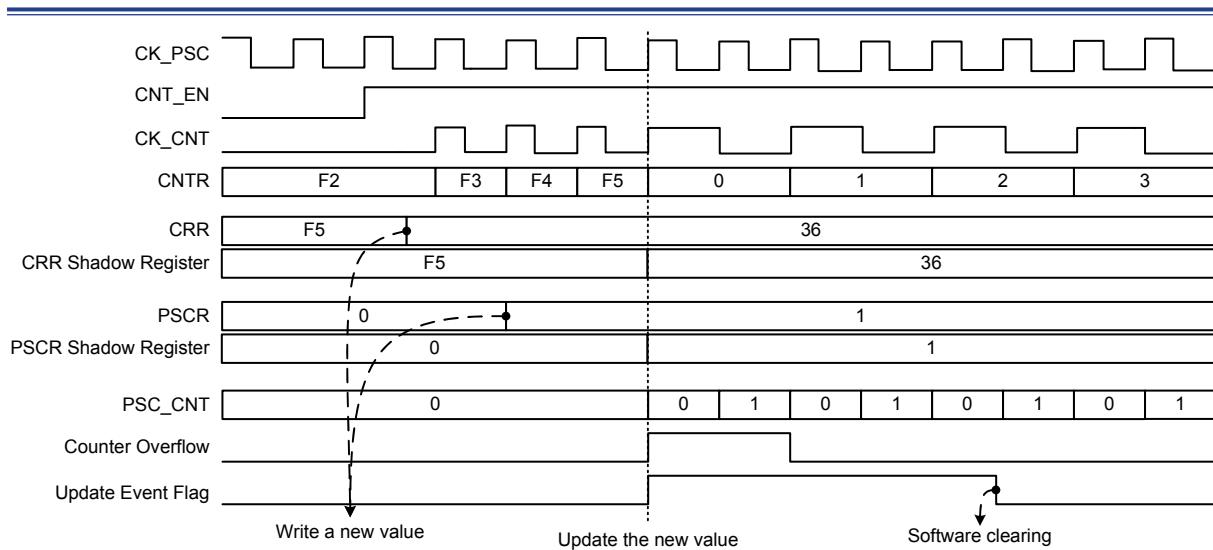


Figure 112. Up-counting Example

## 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.

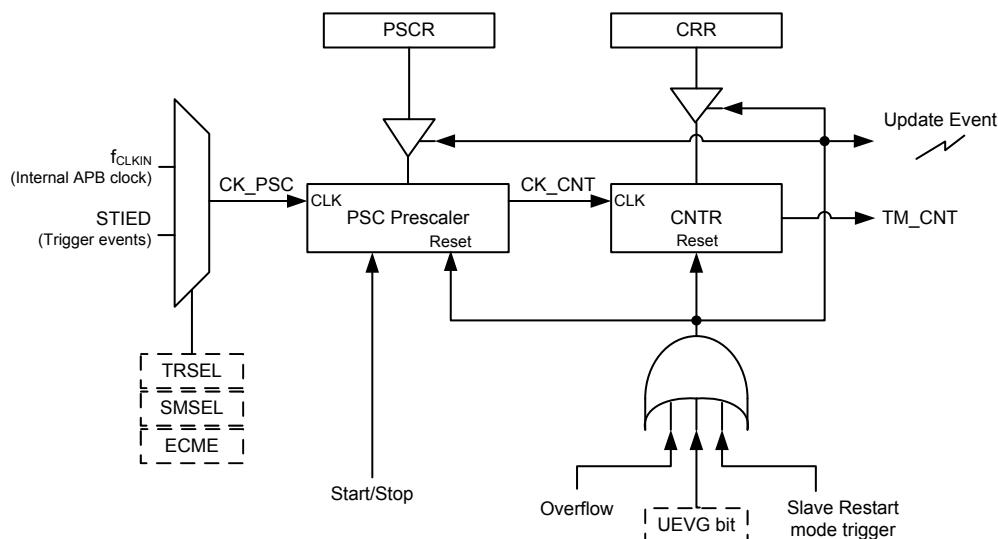


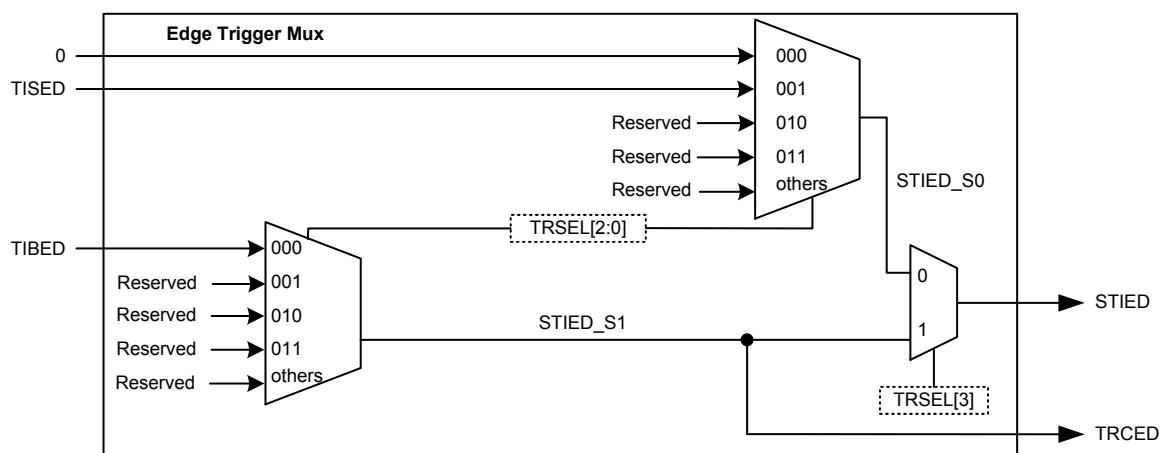
Figure 113. SCTM Clock Selection Source

## Trigger Controller

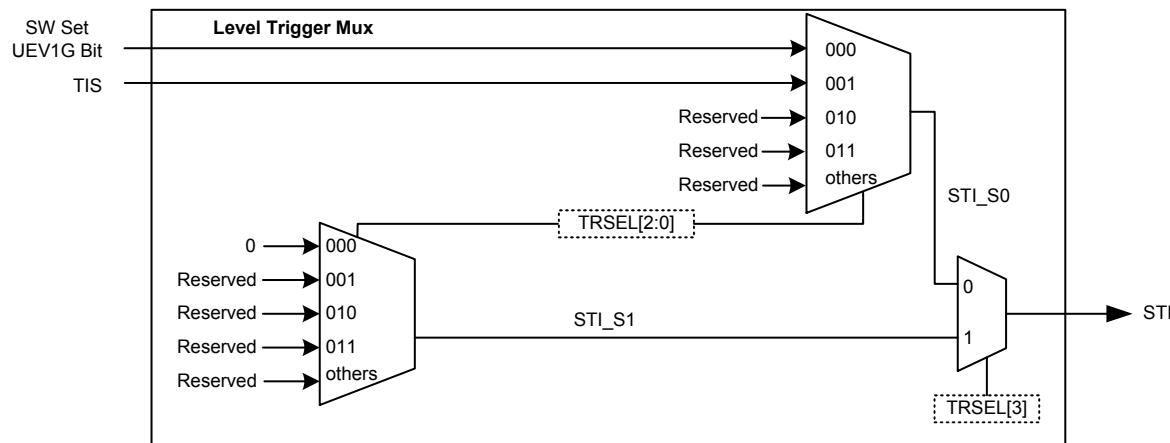
The trigger controller is used to select the trigger source and setup the trigger level or edge trigger condition. For the internal trigger input, it can be selected by the Trigger Selection bits TRSEL in the TRCFR register. For all the trigger sources except the UEVG bit software trigger, the internal 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.

Trigger Controller Block = Edge Trigger Mux + Level Trigger Mux

Edge Trigger = Channel input



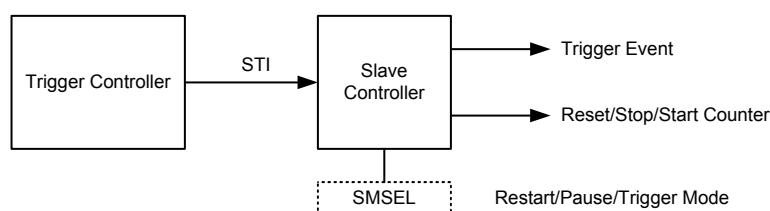
Level Trigger Source = Channel input + Software UEV1G bit



**Figure 114. Trigger Control 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.

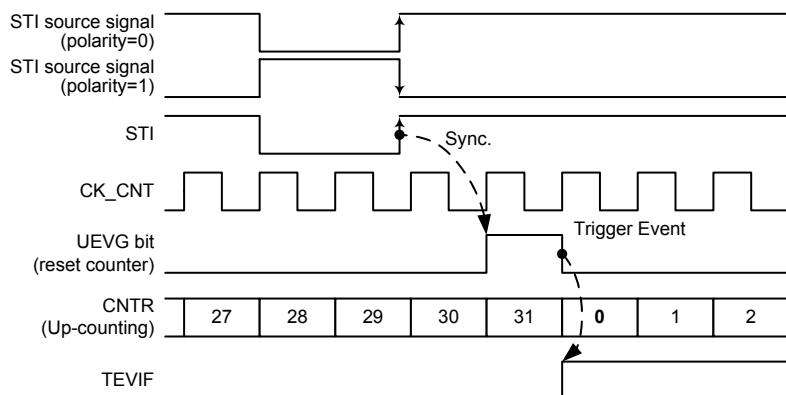


**Figure 115. Slave Controller Diagram**

### 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.

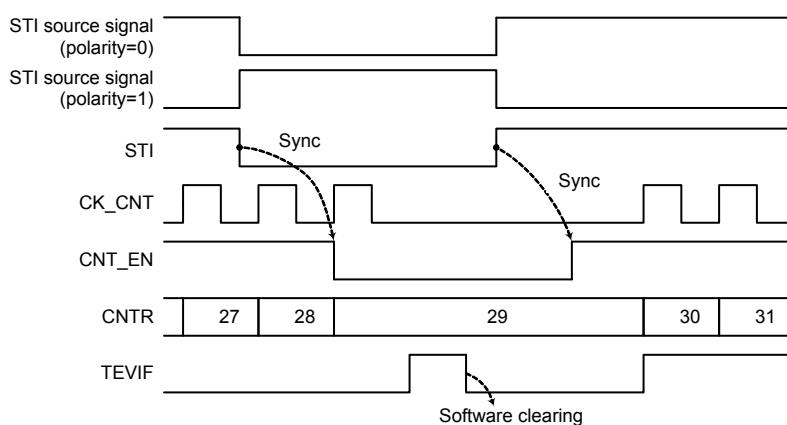
#### Timer Counter Reload Register CRR = 32



**Figure 116. SCTM in Restart Mode**

### Pause Mode

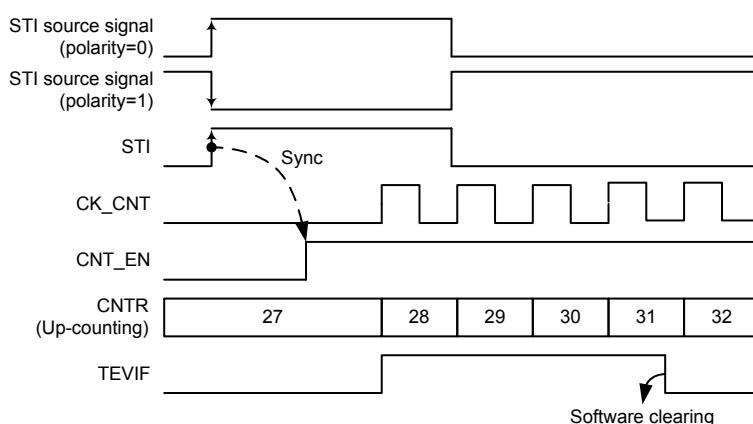
In the Pause Mode, the selected STI input signal level is used to control the counter start/stop operation. The counter starts to count when the selected STI signal is at a high level and stops counting when the STI signal is changed to a low level, here the counter will maintain its present value and 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 117. 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 118. SCTM in Trigger Mode**

## Channel Controller

The SCTM channel can be used as the capture input or compare match output. 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.

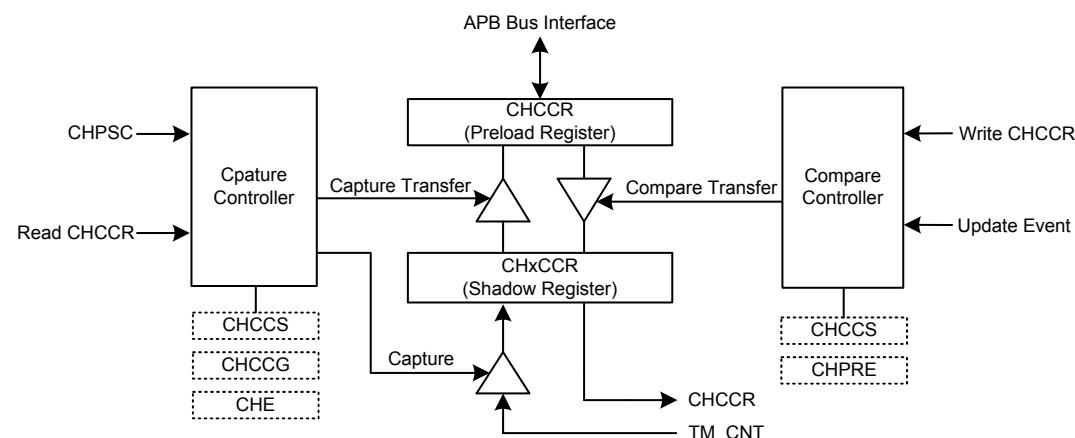
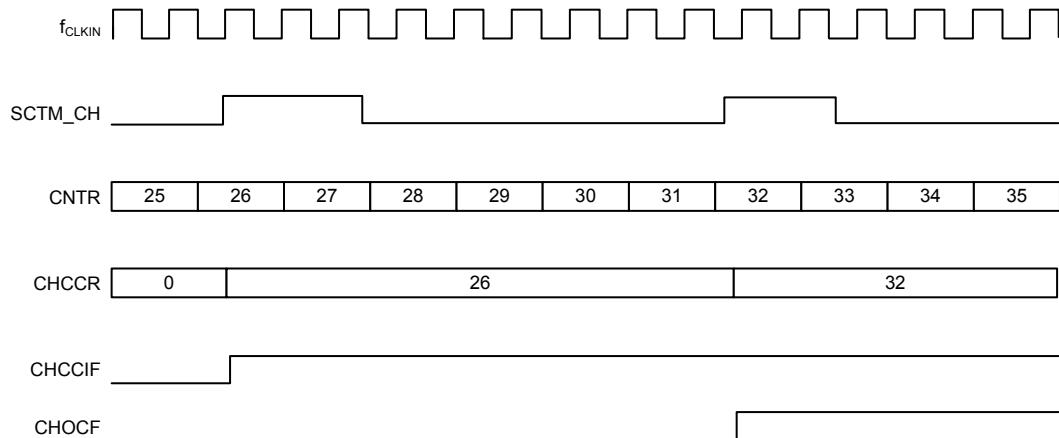


Figure 119. Capture/Compare Block Diagram

### 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 CHOFC, will be set.



**Figure 120. 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).

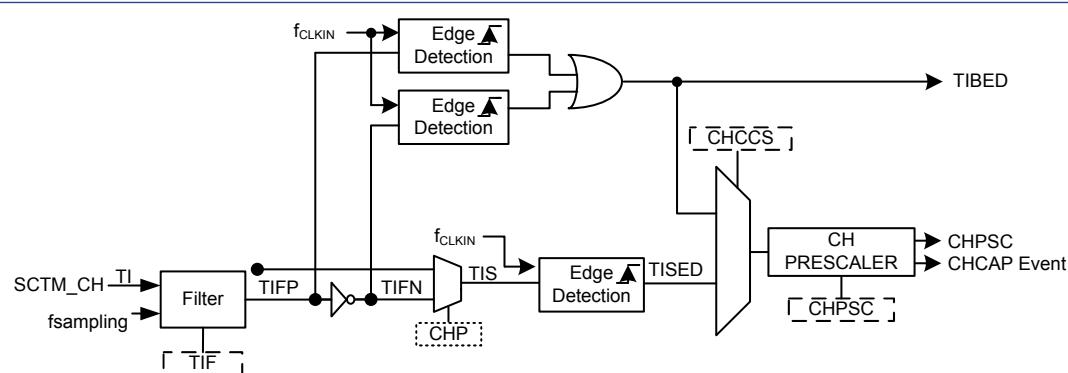


Figure 121. Channel Input Stages

## 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 each filter.

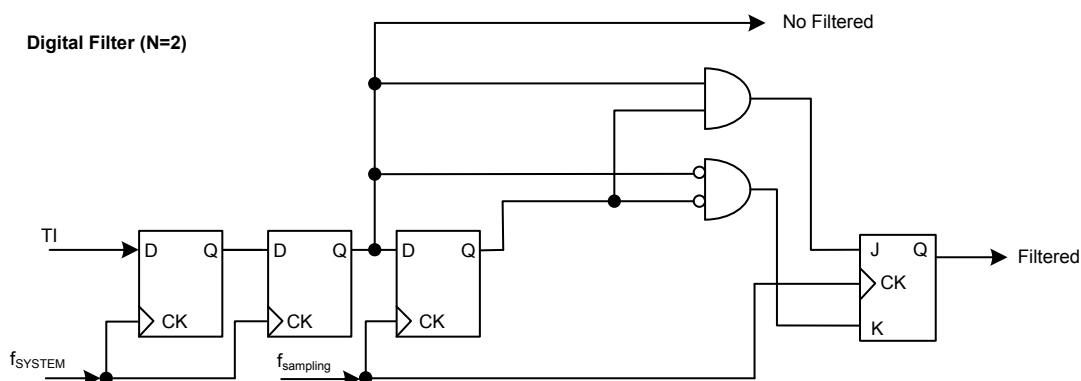
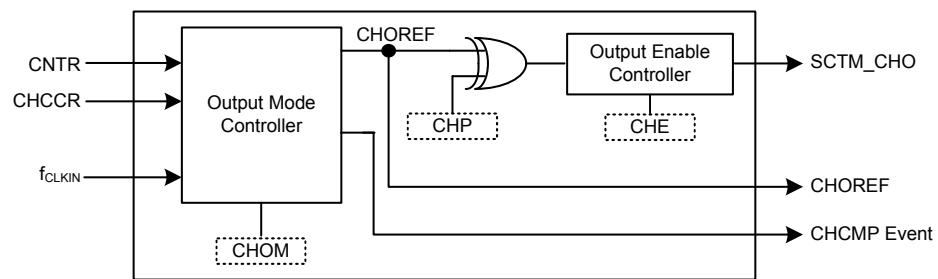


Figure 122. TI Digital Filter Diagram with N = 2

## 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 123. 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 39 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

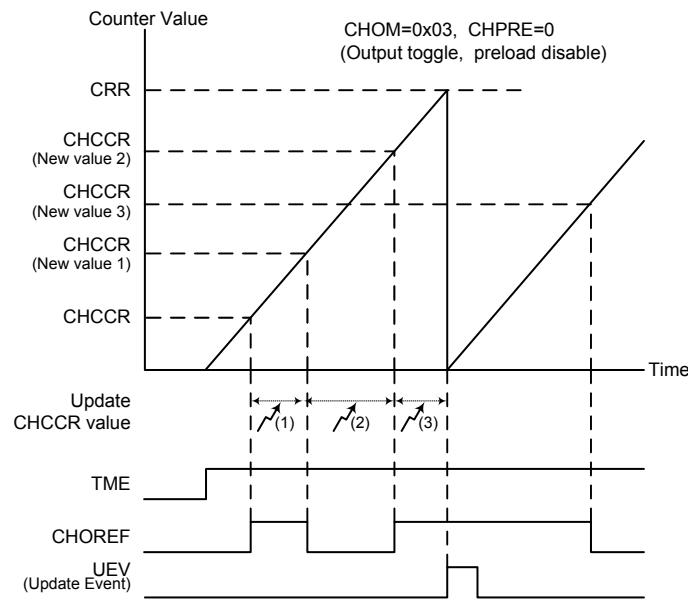


Figure 124. Toggle Mode Channel Output Reference Signal – CHPRE = 0

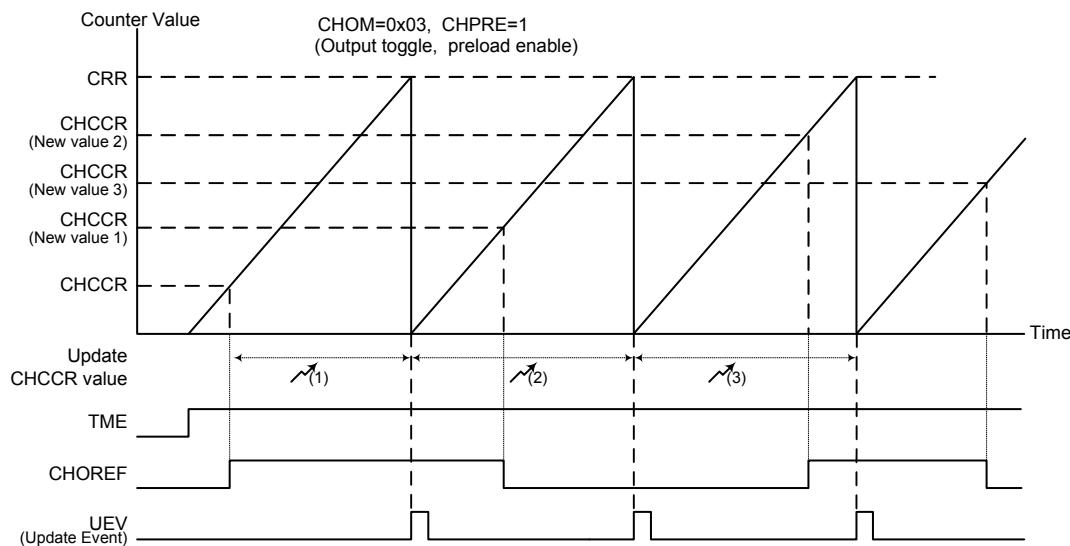
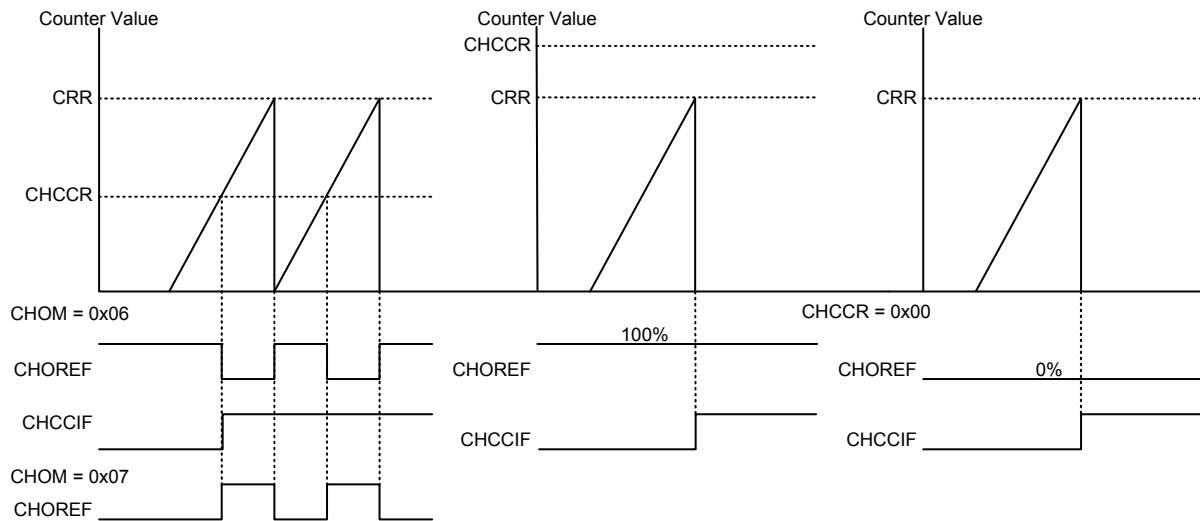


Figure 125. Toggle Mode Channel Output Reference Signal – CHPRE = 1



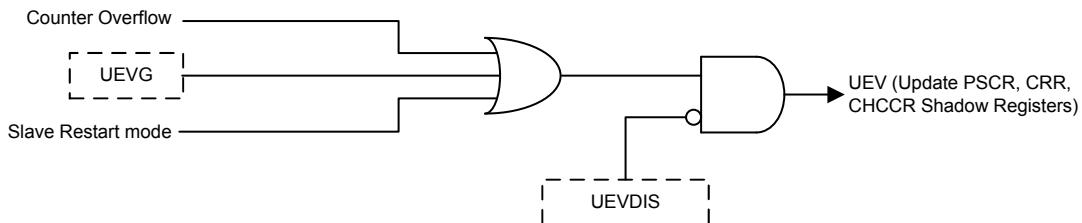
**Figure 126. PWM Mode Channel Output Reference Signal**

## 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**

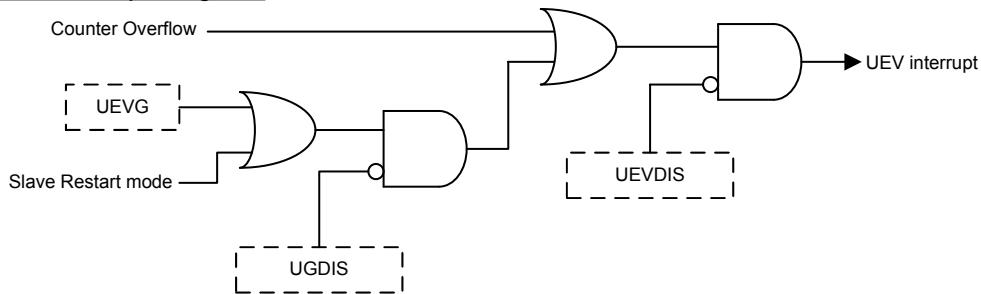


Figure 127. Update Event Setting Diagram

## 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

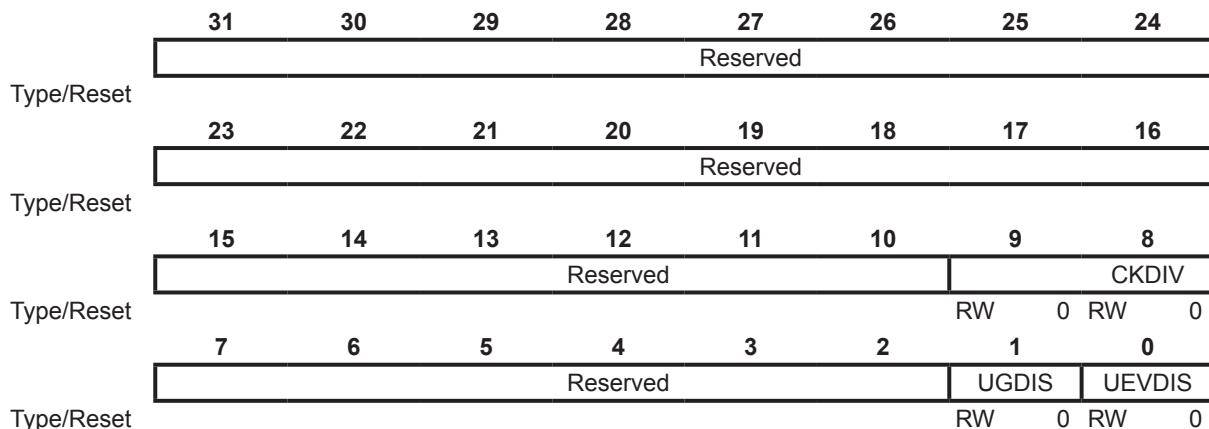
## Register Descriptions

### Timer Counter Configuration Register – CNTCFR

This register specifies the SCTM counter configuration.

Offset: 0x000

Reset value: 0x0000\_0000



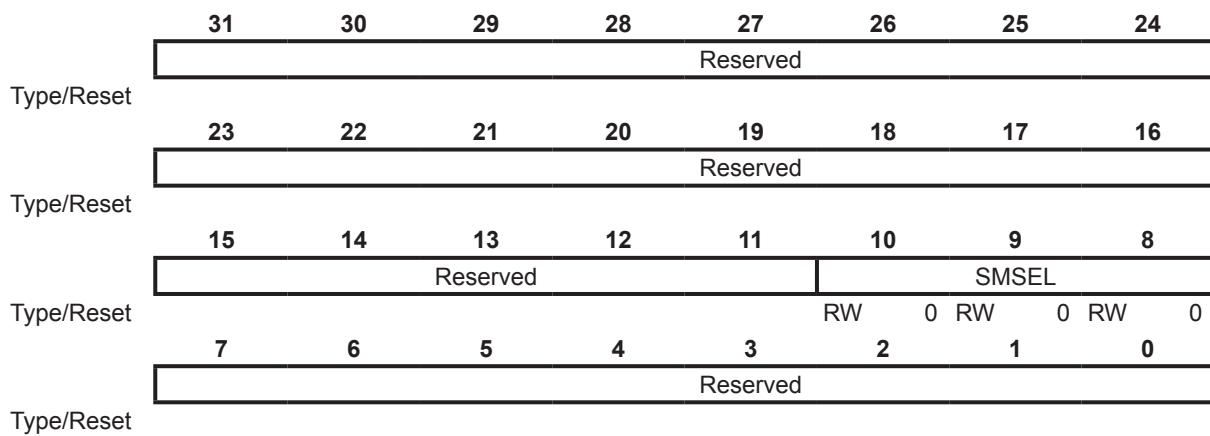
Bits	Field	Descriptions
[9:8]	CKDIV	<p>Clock Division</p> <p>These two bits define the frequency ratio between the timer clock (<math>f_{CLKIN}</math>) and the dead-time clock (<math>f_{DTS}</math>). The dead-time clock is also used for digital filter sampling clock.</p> <ul style="list-style-type: none"> <li>00: <math>f_{DTS} = f_{CLKIN}</math></li> <li>01: <math>f_{DTS} = f_{CLKIN} / 2</math></li> <li>10: <math>f_{DTS} = f_{CLKIN} / 4</math></li> <li>11: Reserved</li> </ul>
[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"> <li>- Counter overflow</li> <li>- Setting the UEVG bit</li> <li>- Update generation through the slave mode</li> </ul> <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"> <li>- Counter overflow</li> <li>- Setting the UEVG bit</li> <li>- Update generation through the slave mode</li> </ul> <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



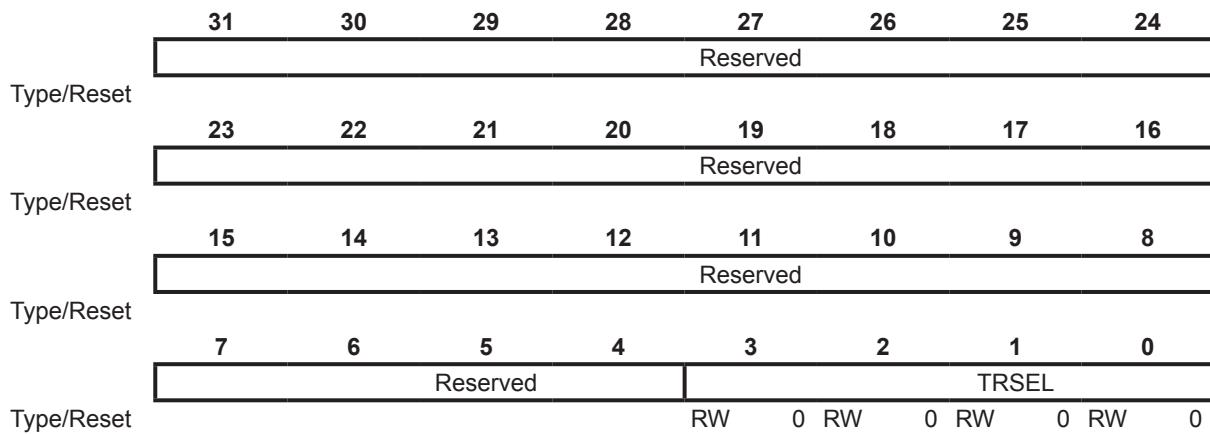
Bits	Field	Descriptions																						
[10:8]	SMSEL	Slave Mode Selection																						
		<table border="1"> <thead> <tr> <th>SMSEL [2:0]</th> <th>Mode</th> <th>Descriptions</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>Disable mode</td> <td>The prescaler is clocked directly by the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



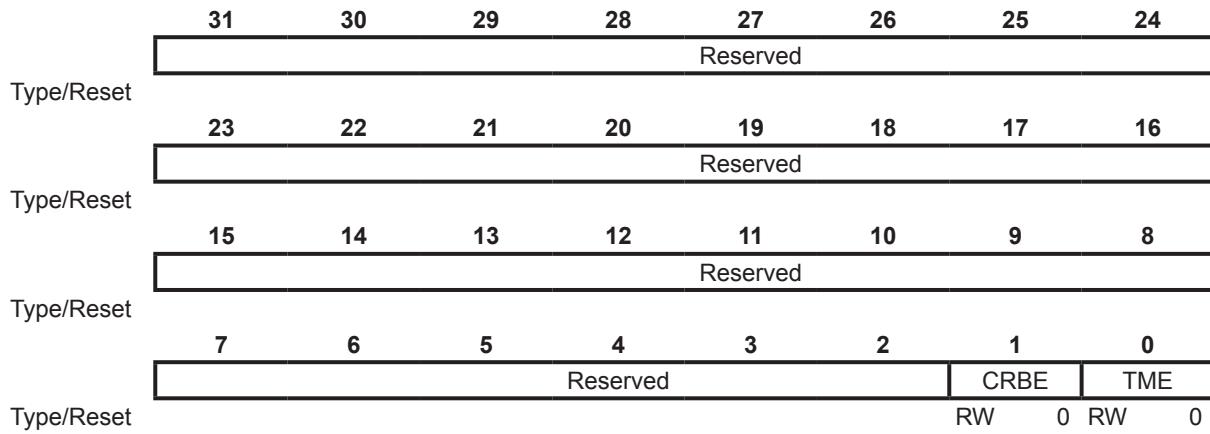
Bits	Field	Descriptions
[3:0]	TRSEL	<p>Trigger Source Selection</p> <p>These bits are used to select the trigger input (STI) for counter synchronizing.</p> <ul style="list-style-type: none"> <li>0000: Software Trigger by setting the UEVG bit</li> <li>0001: Filtered input of channel (TIS)</li> <li>0011: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



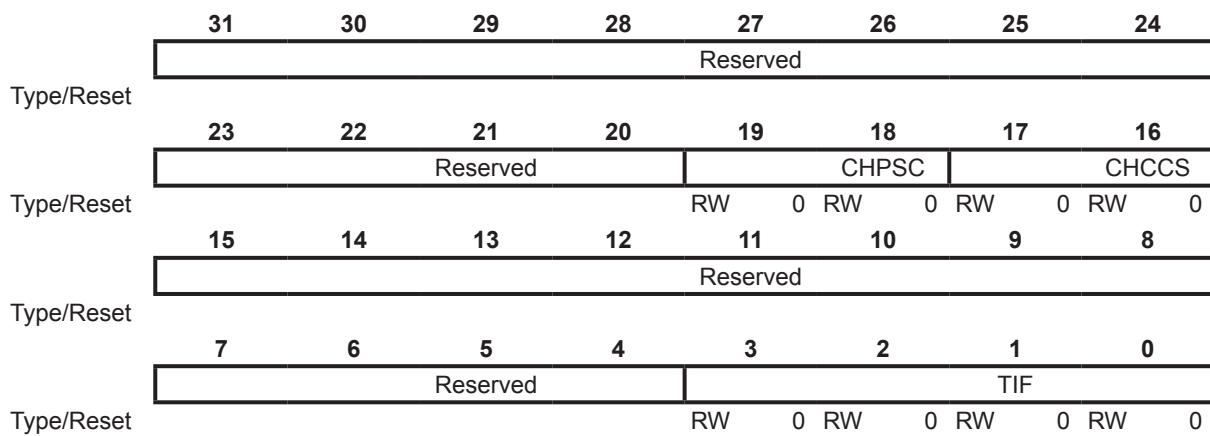
Bits	Field	Descriptions
[1]	CRBE	Counter-Reload register Buffer Enable 0: Counter reload register can be updated immediately 1: Counter reload register cannot be updated until the update event occurs
[0]	TME	Timer Enable bit 0: SCTM off 1: SCTM on – SCTM functions normally When the TME bit is cleared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



Bits	Field	Descriptions
[19:18]	CHPSC	<p>Channel Capture Input Source Prescaler Setting</p> <p>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.</p> <ul style="list-style-type: none"> <li>00: No prescaler, channel capture input signal is chosen for each active event.</li> <li>01: Channel Capture input signal is chosen for every 2 events.</li> <li>10: Channel Capture input signal is chosen for every 4 events.</li> <li>11: Channel Capture input signal is chosen for every 8 events.</li> </ul>
[17:16]	CHCCS	<p>Channel Capture/Compare Selection.</p> <ul style="list-style-type: none"> <li>00: Channel is configured as an output.</li> <li>01: Channel is configured as an input derived from the TI signal.</li> <li>10: Reserved.</li> <li>11: Channel is configured as an input which comes from the TIBED signal.</li> </ul> <p>Note: The CHCCS field can be accessed only when the CHE bit is cleared to 0.</p>

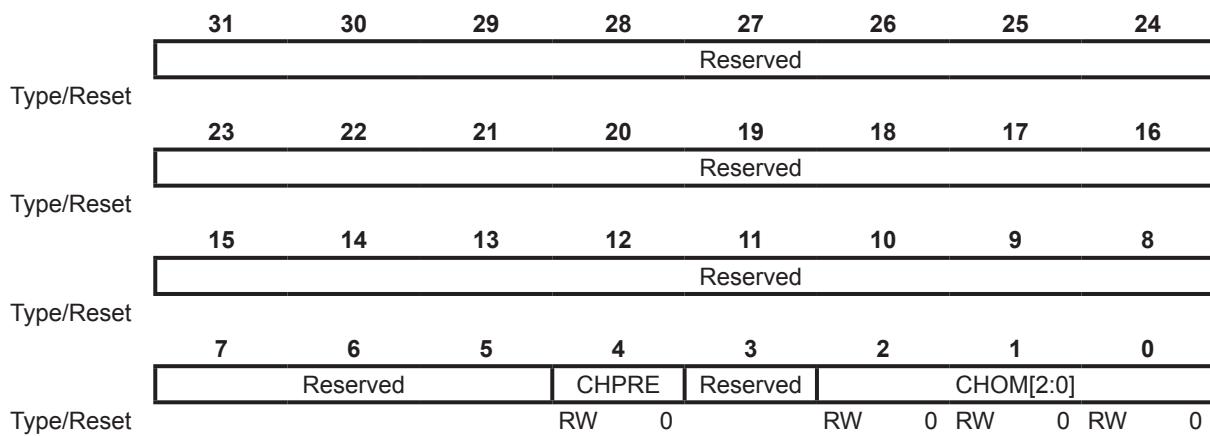
Bits	Field	Descriptions
[3:0]	TIF	<p>Channel Input Source TI Filter Setting</p> <p>These bits define the frequency divided ratio used to sample the TI signal. The Digital filter in the SCTM is an N-event counter where N is defined as how many valid transitions are necessary to output a filtered signal.</p> <ul style="list-style-type: none"> <li>0000: No filter, the sampling clock is <math>f_{SYSTEM}</math>.</li> <li>0001: <math>f_{SAMPLING} = f_{CLKIN}</math>, <math>N = 2</math></li> <li>0010: <math>f_{SAMPLING} = f_{CLKIN}</math>, <math>N = 4</math></li> <li>0011: <math>f_{SAMPLING} = f_{CLKIN}</math>, <math>N = 8</math></li> <li>0100: <math>f_{SAMPLING} = f_{DTS} / 2</math>, <math>N = 6</math></li> <li>0101: <math>f_{SAMPLING} = f_{DTS} / 2</math>, <math>N = 8</math></li> <li>0110: <math>f_{SAMPLING} = f_{DTS} / 4</math>, <math>N = 6</math></li> <li>0111: <math>f_{SAMPLING} = f_{DTS} / 4</math>, <math>N = 8</math></li> <li>1000: <math>f_{SAMPLING} = f_{DTS} / 8</math>, <math>N = 6</math></li> <li>1001: <math>f_{SAMPLING} = f_{DTS} / 8</math>, <math>N = 8</math></li> <li>1010: <math>f_{SAMPLING} = f_{DTS} / 16</math>, <math>N = 5</math></li> <li>1011: <math>f_{SAMPLING} = f_{DTS} / 16</math>, <math>N = 6</math></li> <li>1100: <math>f_{SAMPLING} = f_{DTS} / 16</math>, <math>N = 8</math></li> <li>1101: <math>f_{SAMPLING} = f_{DTS} / 32</math>, <math>N = 5</math></li> <li>1110: <math>f_{SAMPLING} = f_{DTS} / 32</math>, <math>N = 6</math></li> <li>1111: <math>f_{SAMPLING} = f_{DTS} / 32</math>, <math>N = 8</math></li> </ul>

## Channel Output Configuration Register – CHOCFR

This register specifies the channel output mode configuration.

Offset: 0x040

Reset value: 0x0000\_0000



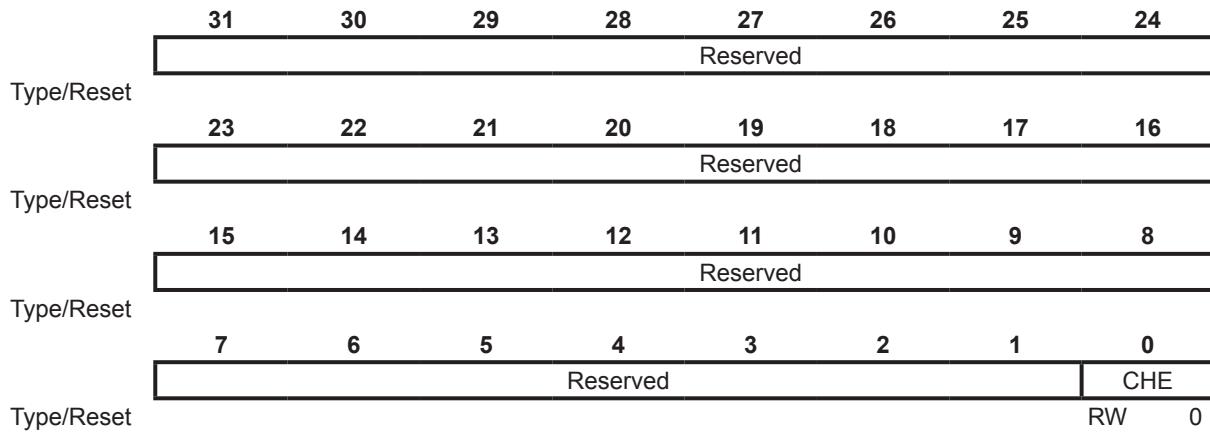
Bits	Field	Descriptions
[4]	CHPRE	<p>Channel Capture/Compare Register (CHCCR) Preload Enable</p> <p>0: CHCCR preload function is disabled.                      The CHCCR register can be immediately assigned a new value when the CHPRE bit is cleared to 0 and the updated CHCCR value is used immediately.</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> <ul style="list-style-type: none"> <li>000: No Change</li> <li>001: Output 0 on compare match</li> <li>010: Output 1 on compare match</li> <li>011: Output toggles on compare match</li> <li>100: Force inactive – CHOREF is forced to 0</li> <li>101: Force active – CHOREF is forced to 1</li> <li>110: PWM mode 1                             <ul style="list-style-type: none"> <li>- During up-counting, channel has an active level when CNTR &lt; CHCCR or otherwise has an inactive level.</li> </ul> </li> <li>111: PWM mode 2                             <ul style="list-style-type: none"> <li>- During up-counting, channel has an inactive level when CNTR &lt; CHCCR or otherwise has an active level.</li> </ul> </li> </ul>

## Channel Control Register – CHCTR

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

Offset: 0x050

Reset value: 0x0000\_0000



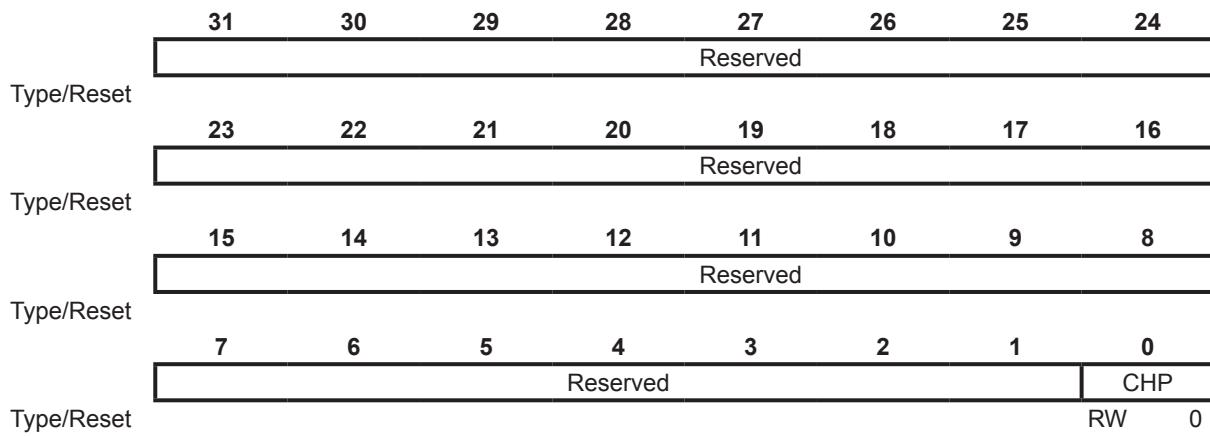
Bits	Field	Descriptions
[0]	CHE	<p>Channel Capture/Compare Enable</p> <ul style="list-style-type: none"> <li>- Channel is configured as an input (CHCCS = 0x01/0x03)                     <ul style="list-style-type: none"> <li>0: Input Capture Mode disabled</li> <li>1: Input Capture Mode enabled</li> </ul>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

## Channel Polarity Configuration Register – CHPOLR

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

Offset: 0x054

Reset value: 0x0000\_0000



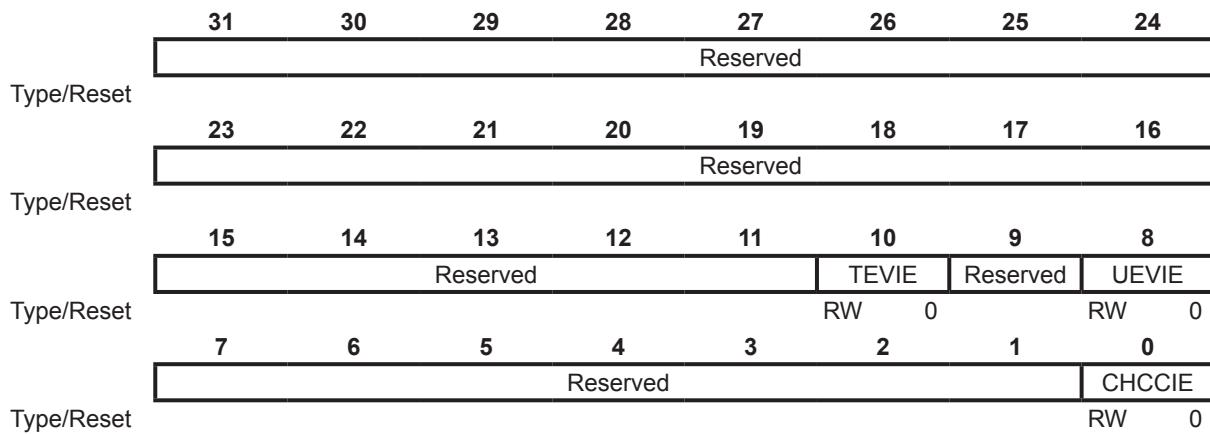
Bits	Field	Descriptions
[0]	CHP	<p>Channel Capture/Compare Polarity</p> <ul style="list-style-type: none"> <li>- When Channel is configured as an input           <ul style="list-style-type: none"> <li>0: capture event occurs on a Channel rising edge</li> <li>1: capture event occurs on a Channel falling edge</li> </ul> </li> <li>- When Channel is configured as an output (CHCCS = 0x00)           <ul style="list-style-type: none"> <li>0: Channel Output active high</li> <li>1: Channel Output active low</li> </ul> </li> </ul>

## Timer Interrupt Control Register – DICTR

This register contains the timer interrupt enable control bits.

Offset: 0x074

Reset value: 0x0000\_0000



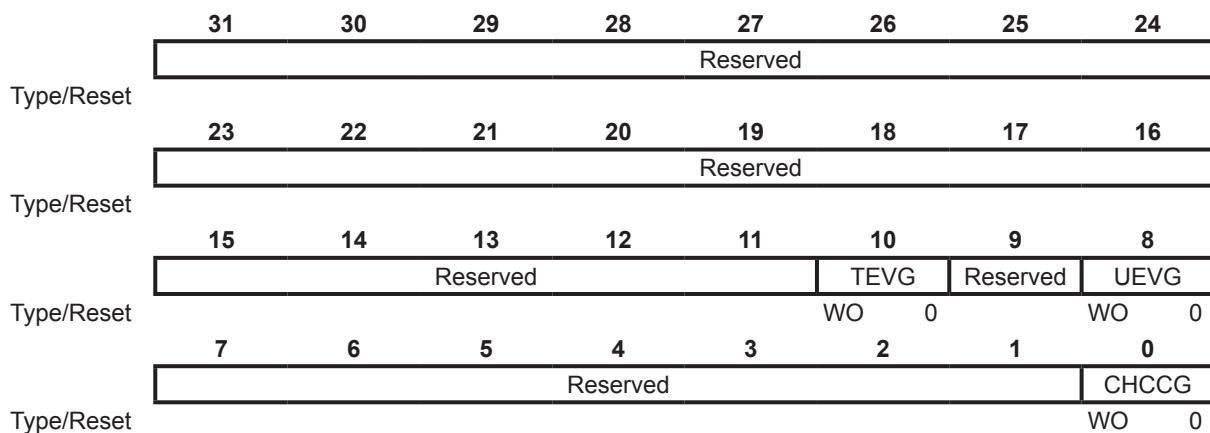
Bits	Field	Descriptions
[10]	TEVIE	Trigger event Interrupt Enable 0: Trigger event interrupt disabled 1: Trigger event interrupt enabled
[8]	UEVIE	Update event Interrupt Enable 0: Update event interrupt disabled 1: Update event interrupt enabled
[0]	CHCCIE	Channel Capture/Compare Interrupt Enable 0: Channel interrupt disabled 1: Channel interrupt 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	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	Reserved					TEVIF	Reserved	UEVIF	
						RW 0	RW 0	RW 0	
Type/Reset	7	6	5	4	3	2	1	0	
	Reserved			CHOCF	Reserved			CHCCIF	
Type/Reset				RW 0				RW 0	

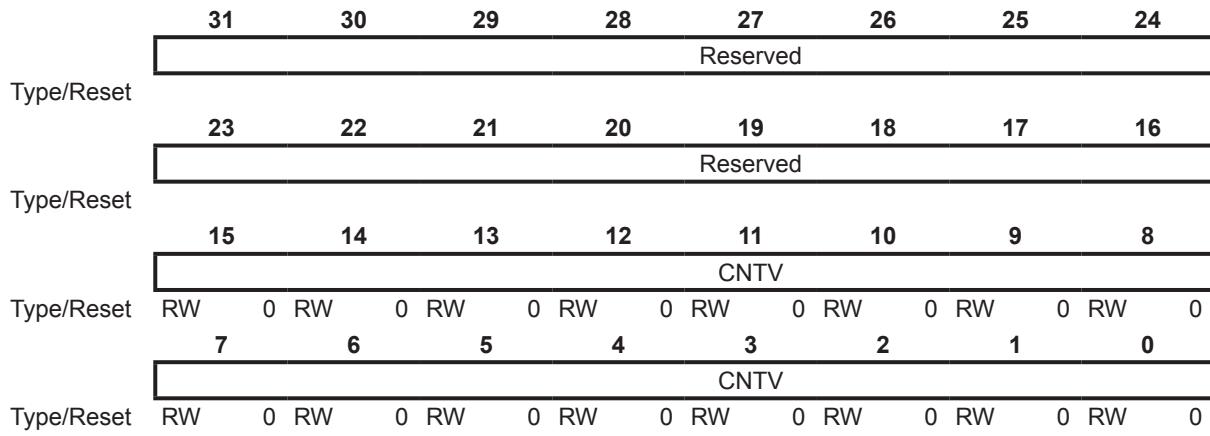
Bits	Field	Descriptions
[10]	TEVIF	Trigger Event Interrupt Flag This flag is set by hardware on a trigger event and is cleared by software. 0: No trigger event occurs 1: Trigger event occurs
[8]	UEVIF	Update Event Interrupt Flag. This bit is set by hardware on an update event and is cleared by software.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



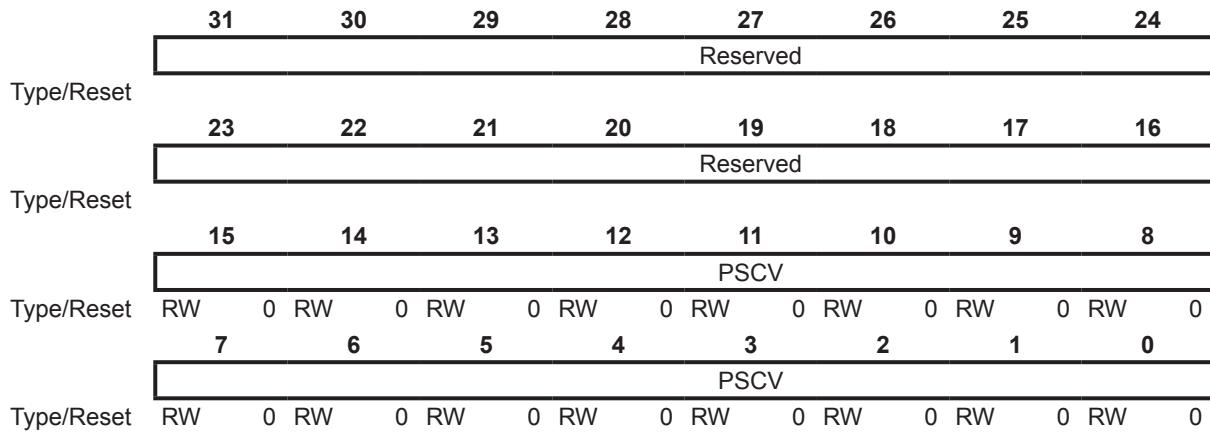
Bits	Field	Descriptions
[15:0]	CNTV	Counter Value.

## Timer Prescaler Register – PSCR

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

Offset: 0x084

Reset value: 0x0000\_0000



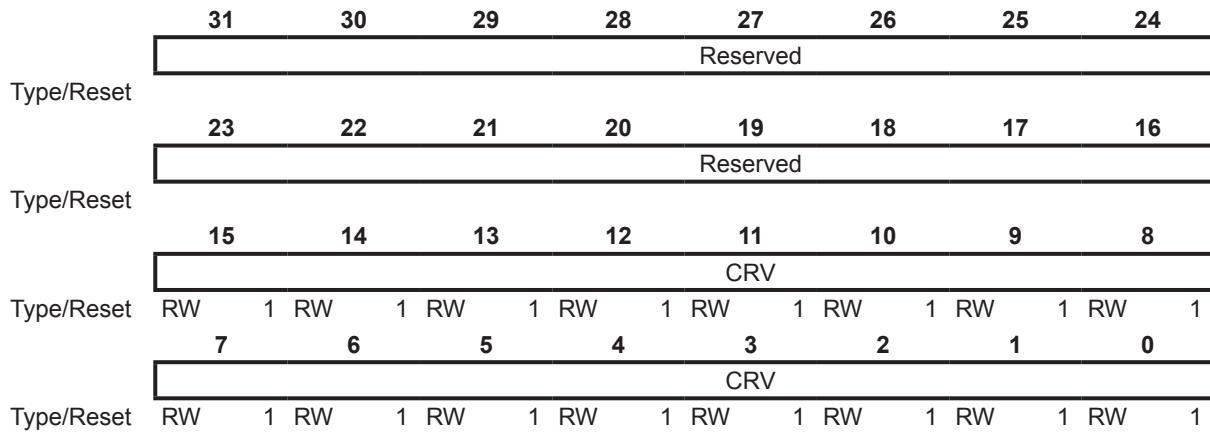
Bits	Field	Descriptions
[15:0]	PSCV	<p>Prescaler Value</p> <p>These bits are used to specify the prescaler value to generate the counter clock frequency <math>f_{CK\_CNT}</math>.</p> $f_{CK\_CNT} = \frac{f_{CK\_PSC}}{PSCV[15:0] + 1}$ , where the $f_{CK\_PSC}$ is the prescaler clock source.

## Timer Counter Reload Register – CRR

This register specifies the timer counter reload value.

Offset: 0x088

Reset value: 0x0000\_FFFF



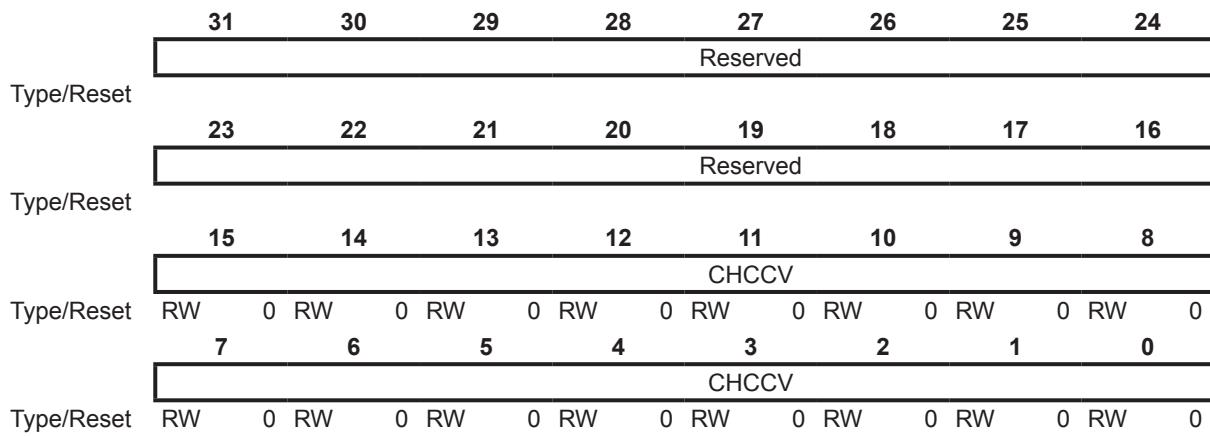
Bits	Field	Descriptions
[15:0]	CRV	Counter Reload Value The CRV is the reload value which is loaded into the actual counter register.

## Channel Capture/Compare Register – CHCCR

This register specifies the timer channel capture/compare value.

Offset: 0x090

Reset value: 0x0000\_0000



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

# 17

## Real Time Clock (RTC)

### Introduction

The Real Time Clock, RTC, circuitry includes the APB interface, a 24-bit up-counter, a control register, a prescaler, a compare register and a status register. Most of the RTC circuits are located in the V<sub>DD</sub> Domain, as shown shaded in the accompanying figure, except for the APB interface. The APB interface is located in the V<sub>DD15</sub> domain. Therefore, it is necessary to be isolated from the ISO signal that comes from the power control unit when the V<sub>DD15</sub> 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.

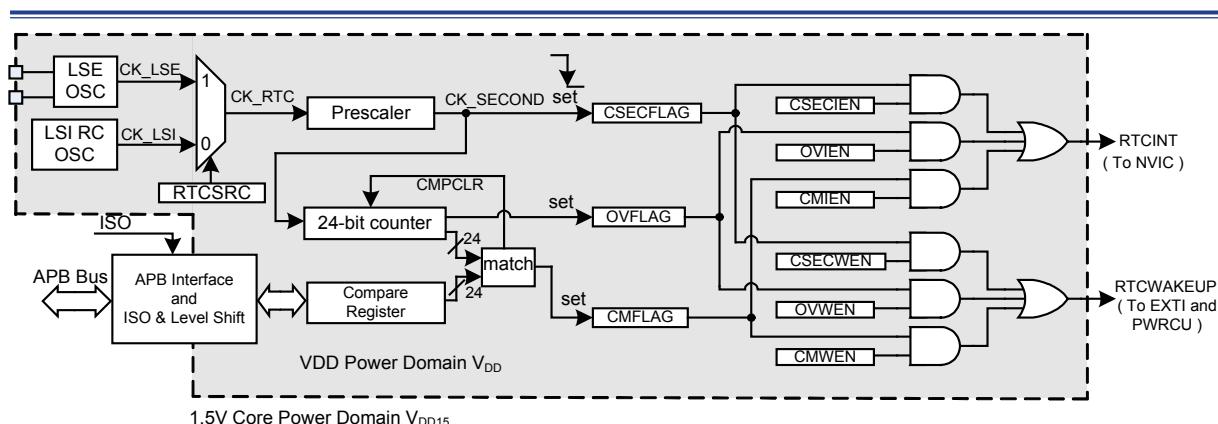


Figure 128. RTC Block Diagram

### Features

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

## Functional Descriptions

### RTC Related Register Reset

The RTC registers can only be reset by either a V<sub>DD</sub> Domain power on reset, POR, or by a V<sub>DD</sub> Domain software reset by setting the PWRST bit in the PWRCR 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<sub>DD</sub> supply voltage. Therefore, the RTC circuitry remains operational in the Power-Down mode where V<sub>DD15</sub> is powered off. Only the APB bus, which is located in the V<sub>DD15</sub> domain, is interconnected to the circuits located in the V<sub>DD</sub> domain using level shift circuitry and isolated by the ISO signals when the V<sub>DD15</sub> supply voltage is powered off. The isolation function must be disabled by setting the VDDISO 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<sup>0</sup> to 2<sup>15</sup> determined by the RPREG [3:0] field. For instance, setting the prescaler value RPREG [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	LSESM Setting in the RTCCR register	Operating Current	Startup time
Normal startup	0	2.0 μA	Above 500 ms
Fast startup	1	3.5 μA	Below 300 ms

@ V<sub>DD</sub> = 3.3 V and LSE clock = 32,768 Hz; these values are only for reference, actual values are dependent on the specification of the external 32.768KHz crystal.

## RTC Counter Operation

The RTC provides a 24-bit up-counter which 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 24-bit compare register, RTCCMP, is provided to store the specific value to be compared with the RTCCNT content. This is used to define a pre-determined time interval. When the RTCCNT register content is equal to the RTCCMP register value, the match flag CMFLAG in the RTCSR register will be set by hardware and an interrupt or wakeup event can be sent according to the corresponding enable bits in the RTCIWEN register. The RTC counter will be either reset to zero or keep counting when the compare match event occurs, dependent upon the CMPCLR bit in the RTCCR register. For example, if the RPREG [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)				
		ROLF				
	1 Second clock	RTCCMP		X		
		RTCCNT	3	4	5	
		RTCOUT (ROAP = 0)	$T_R$	$T_R$	$T_R$	
		RTCOUT (ROAP = 1)				
		ROLF				
1 (Level mode)	0 Compare match	RTCCMP		4		
		RTCCNT	3	4	5	
		RTCOUT (ROAP = 0)				
		RTCOUT (ROAP = 1)				
		ROLF			$\rightarrow$	
	1 Second clock	RTCCMP		X		
		RTCCNT	3	4	5	
		RTCOUT (ROAP = 0)				
		RTCOUT (ROAP = 1)				
		ROLF		$\rightarrow$	$\rightarrow$	

$T_R$ : RTCOUT output pulse time =  $1 / f_{CK\_RTC}$

$\rightarrow$ : Clear by software reading ROLF bit

## Register Map

The following table shows the RTC registers and reset values. Note all the registers in this unit are located at the V<sub>DD</sub> 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_OF04
RTCSR	0x00C	RTC Status Register	0x0000_0000
RTCIWEN	0x010	RTC Interrupt and Wakeup Enable Register	0x0000_0000

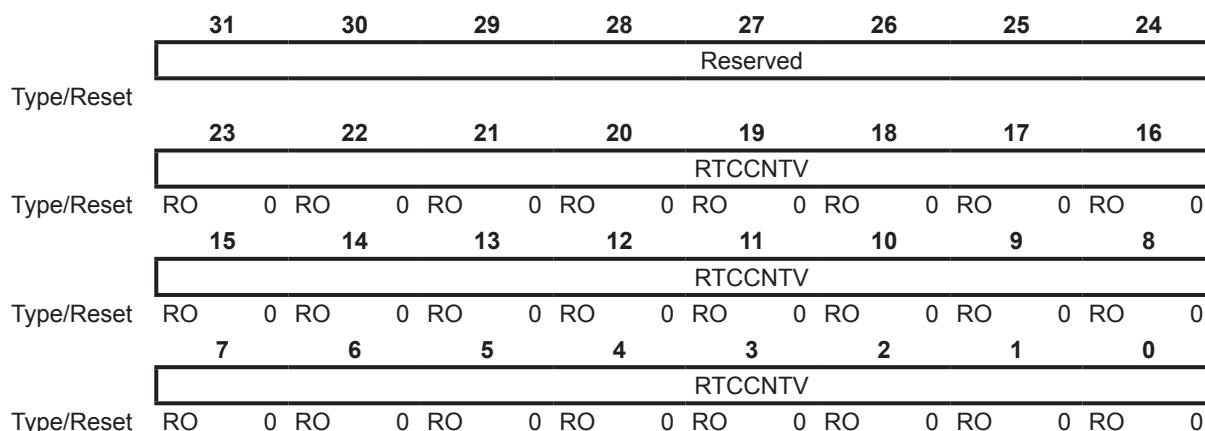
## Register Descriptions

### RTC Counter Register – RTCCNT

This register defines a 24-bit up counter which is incremented by the CK\_SECOND clock.

Offset: 0x000

Reset value: 0x0000\_0000 (Reset by V<sub>DD</sub> Power Domain reset only)



Bits	Field	Descriptions
[23:0]	RTCCNTV	<p>RTC Counter</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"> <li>- V<sub>DD</sub> Domain software reset – set the PWRST bit in the PWRCR register</li> <li>- V<sub>DD</sub> Domain power on reset – POR</li> <li>- Compare match (RTCCNT = RTCCMP) when CMPCLR = 1 (in the RTCCR register)</li> <li>- RTCEN bit changed from 0 to 1</li> </ul>

## 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 V<sub>DD</sub> Power Domain reset only)

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

Bits	Field	Descriptions
[23:0]	RTCCMPV	<p>RTC Compare Match Value</p> <p>A match condition happens when the value in the RTCCNT register is equal to RTCCMP value. An interrupt can be generated if the CMIEN bit in the RTCIEN 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 V<sub>DD</sub> Power Domain reset only)

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

Bits	Field	Descriptions
[20]	ROLF	RTCOUT Level Mode Flag 0: RTCOUT Output is inactive 1: RTCOUT Output is holding as active level 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.
[19]	ROAP	RTCOUT Output Active Polarity 0: Active level is high 1: Active level is low
[18]	ROWM	RTCOUT Output Waveform Mode 0: Pulse mode The output pulse duration is one RTC clock (CK_RTC) period. 1: Level mode The RTCOUT signal will remain at an active level until the ROLF bit is cleared by software reading the ROLF bit.
[17]	ROES	RTCOUT Output Event Selection 0: RTC compare match is selected 1: RTC second clock (CK_SECOND) event is selected 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.
[16]	ROEN	RTCOUT Output Pin Enable 0: Disable RTCOUT output pin 1: Enable RTCOUT output pin 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.

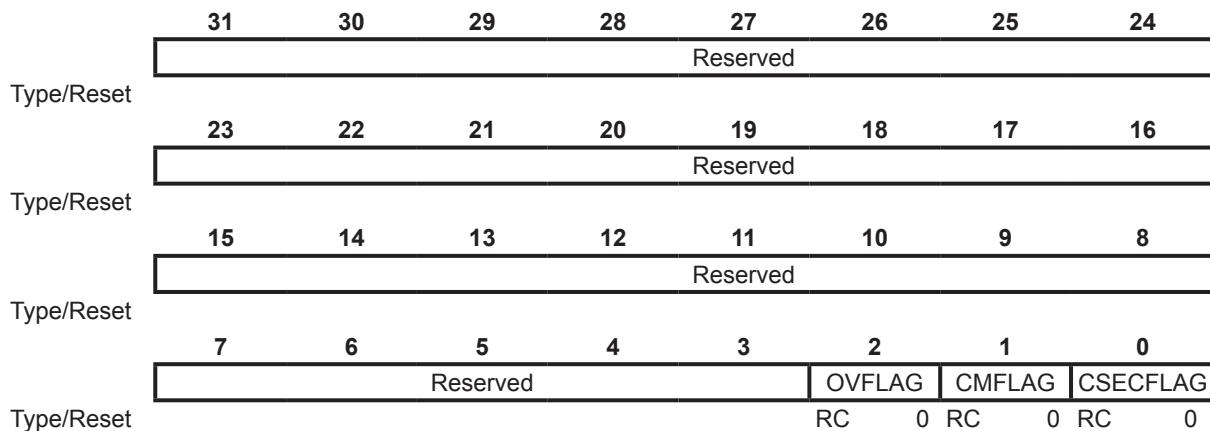
Bits	Field	Descriptions
[11:8]	RPRE	RTC Clock Prescaler Select $CK_{SECOND} = CK_{RTC} / 2^{RPRE}$ 0000: $CK_{SECOND} = CK_{RTC} / 2^0$ 0001: $CK_{SECOND} = CK_{RTC} / 2^1$ 0010: $CK_{SECOND} = CK_{RTC} / 2^2$ ... 1111: $CK_{SECOND} = CK_{RTC} / 2^{15}$
[5]	LSESM	LSE oscillator Startup Mode 0: Normal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

## RTC Status Register – RTCSR

This register stores the counter flags.

Offset: 0x00C

Reset value: 0x0000\_0000 (Reset by V<sub>DD</sub> Power Domain reset and RTCEN bit change from 1 to 0)



## RTC Interrupt and Wakeup Enable Register – RTCIWEN

This register contains the interrupt and wakeup enable bits.

Offset: 0x010

Reset value: 0x0000\_0000 (Reset by V<sub>DD</sub> Power Domain reset only)

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

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

# 18 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.

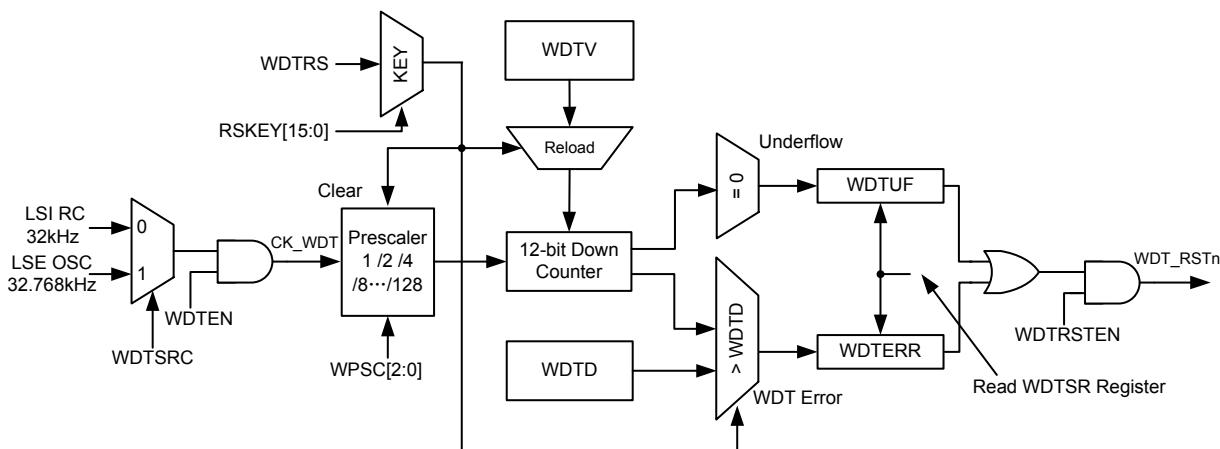


Figure 129. Watchdog Timer Block Diagram

## 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 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 to within a delta value (WDTD). If the Watchdog timer counter value is greater than the delta value and a reload operation is executed, a Watchdog Timer error will occur. The Watchdog timer error will 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.

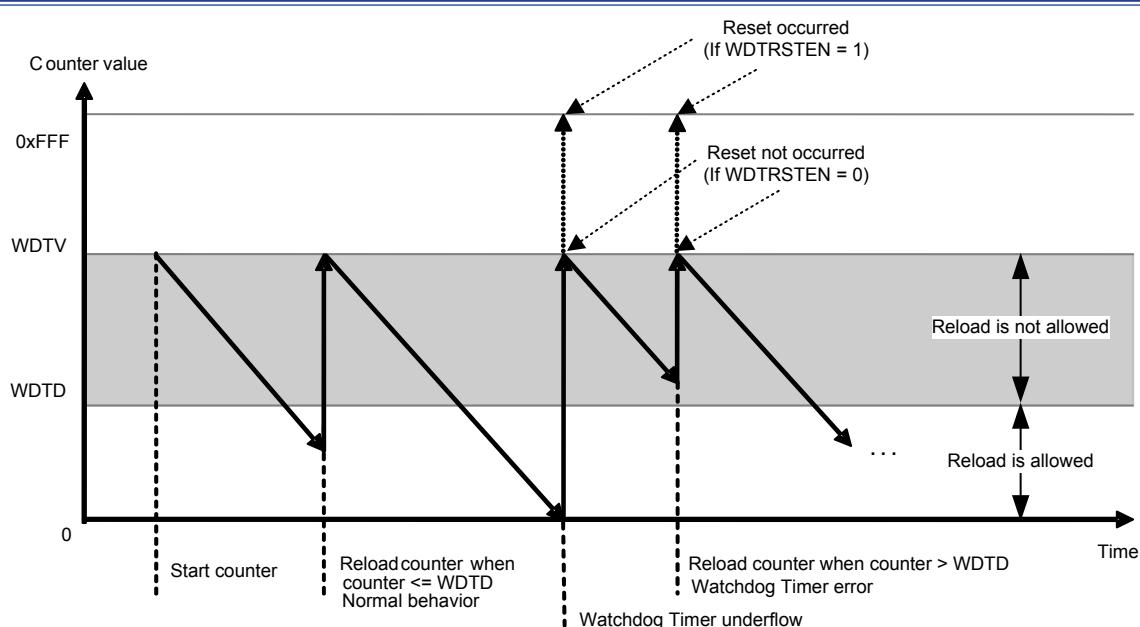
The WDTERR and WDTUF flags in the WDTSR register will be set respectively when the Watchdog timer underflows or when a Watchdog timer error occurs. A system reset or written one operation on the WDTSR register clears the WDTERR and WDTUF flags.

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 MCUDBGCR register) in the Clock Control Unit.

The Watchdog timer should be used in the following manners:

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



**Figure 130. 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

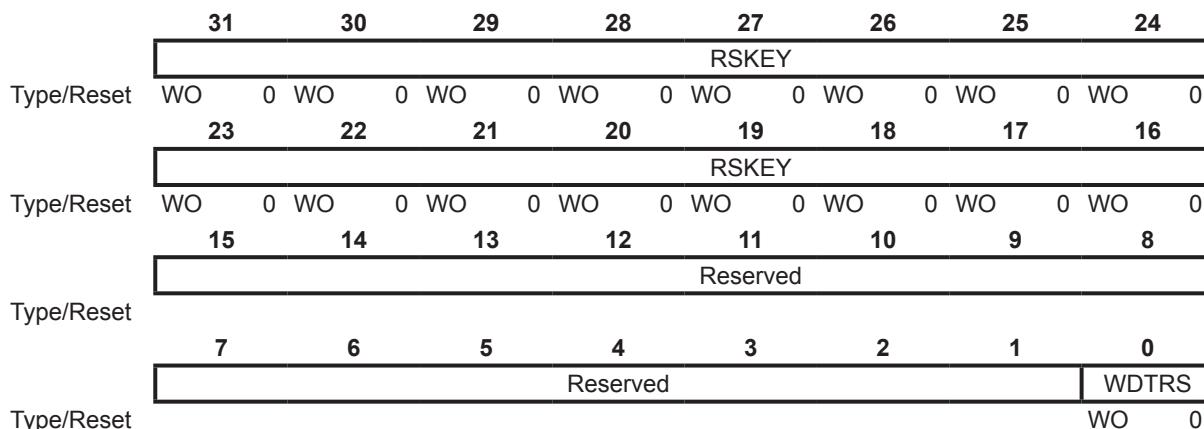
## Register Descriptions

### Watchdog Timer Control Register – WDTCR

This register is used to reload the Watchdog timer.

Offset: 0x000

Reset value: 0x0000\_0000



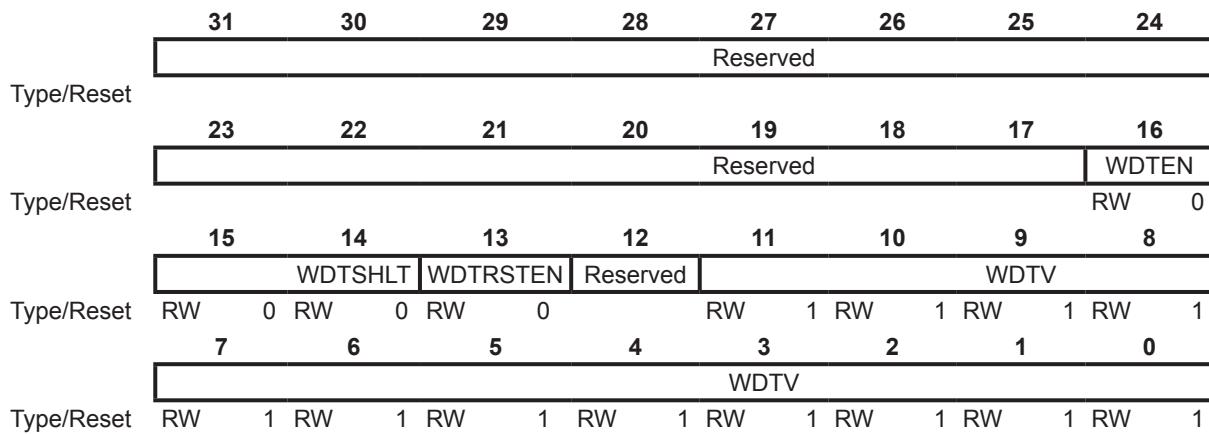
Bits	Field	Descriptions
[31:16]	RSKEY	Watchdog Timer Reload Lock Key The RSKEY [15:0] bits should be written with a 0x5FA0 value to enable the WDT reload operation function. Writing any other value except 0x5FA0 in this field will abort the write operation.
[0]	WDTRS	Watchdog Timer Reload 0: No effect 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

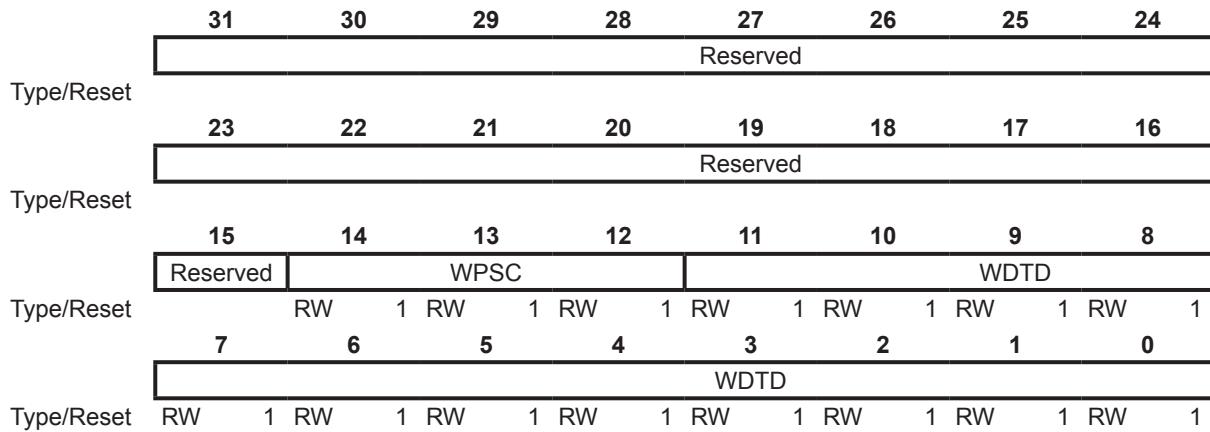


## Watchdog Timer Mode Register 1 – WDTMR1

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

Offset: 0x008

Reset value: 0x0000\_7FFF



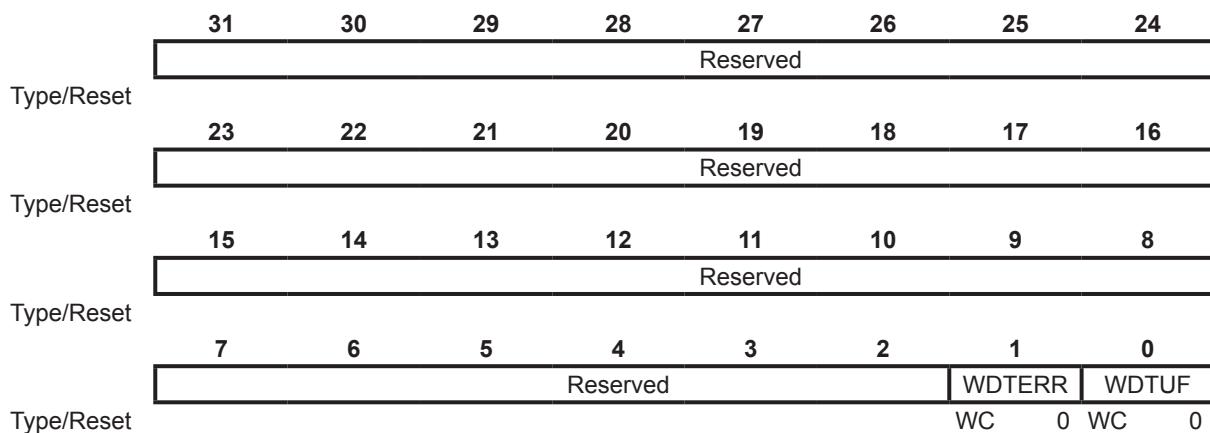
Bits	Field	Descriptions
[14:12]	WPSC	Watchdog Timer Prescaler Selection 000: 1/1 001: 1/2 010: 1/4 011: 1/8 100: 1/16 101: 1/32 110: 1/64 111: 1/128
[11:0]	WDTD	Watchdog Timer Delta Value Define the permitted range to reload the Watchdog timer. If the Watchdog timer counter value is less than or equal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

## Watchdog Timer Status Register – WDTSR

This register specifies the Watchdog timer status.

Offset: 0x00C

Reset value: 0x0000\_0000



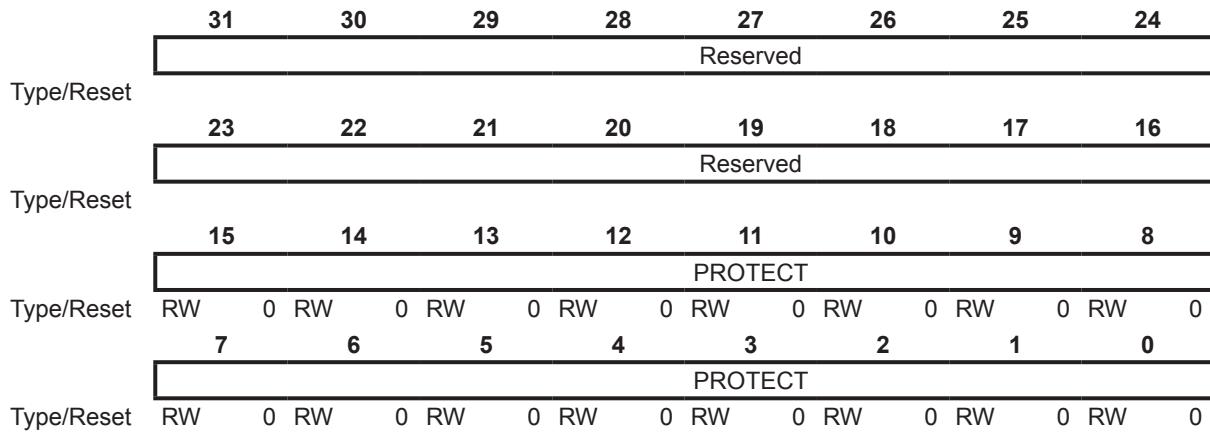
Bits	Field	Descriptions
[1]	WDTERR	<p>Watchdog Timer Error</p> <p>0: No Watchdog timer error has occurred since the last read of this register            1: A Watchdog timer error has occurred since the last read of this register</p> <p>Note: A reload operation when the Watchdog timer counter value is 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            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



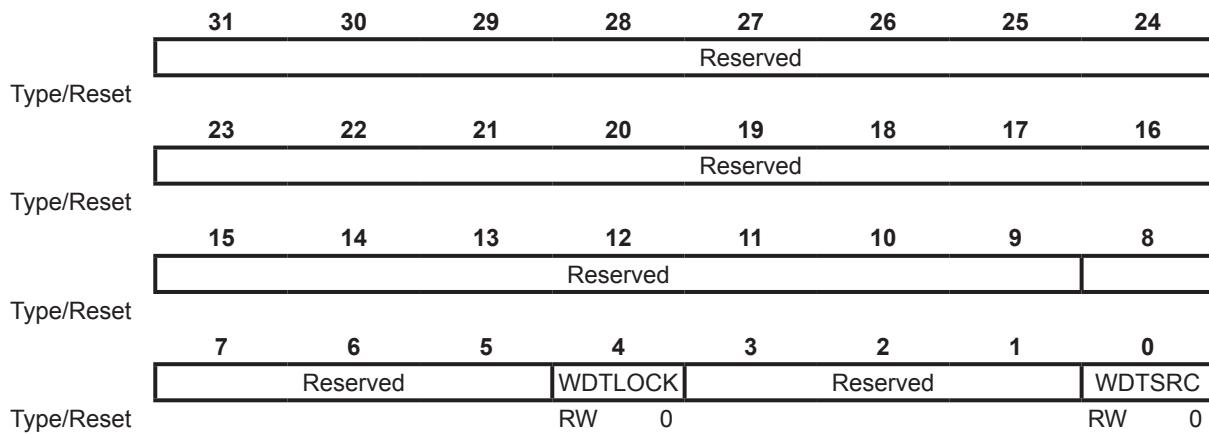
Bits	Field	Descriptions
[15:0]	PROTECT	<p>Watchdog Timer Register Protection</p> <p>For write operation:</p> <ul style="list-style-type: none"> <li>0x35CA: Disable the Watchdog timer register write protection</li> <li>Others: Enable the Watchdog timer register write protection</li> </ul> <p>For read operation:</p> <ul style="list-style-type: none"> <li>0x0000: Watchdog timer register write protection is disabled</li> <li>0x0001: Watchdog timer register write protection is enabled</li> </ul> <p>This register is used to enable/disable the Watchdog timer configuration register write protection function. All configuration registers become read only except for WDTCR and WDTPR when the register write protection is enabled. Additionally, the read operation of PROTECT[0] can obtain the enable/disable status of the register write protection function.</p>

## Watchdog Timer Clock Selection Register – WDTCSR

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

Offset: 0x018

Reset value: 0x0000\_0FFF



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

# 19 Inter-Integrated Circuit (I<sup>2</sup>C)

## Introduction

The I<sup>2</sup>C Module is an internal circuit allowing communication with an external I<sup>2</sup>C 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 I<sup>2</sup>C 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 I<sup>2</sup>C bus is a bi-directional data line between the master and slave devices used for the transmission and reception of data. The I<sup>2</sup>C module also has an arbitration detection function to prevent the situation where more than one master attempts to transmit data on the I<sup>2</sup>C bus at the same time.

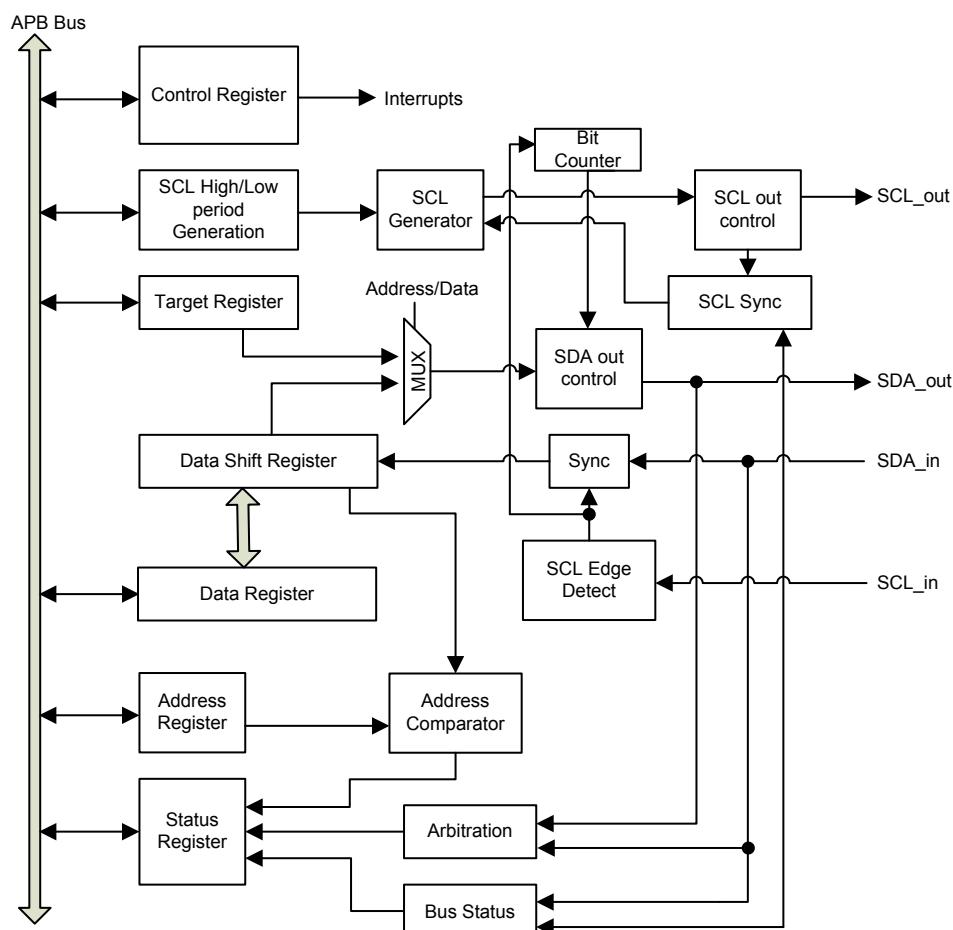


Figure 131. I<sup>2</sup>C Module Block Diagram

## Features

- Two-wire I<sup>2</sup>C 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

## Functional Descriptions

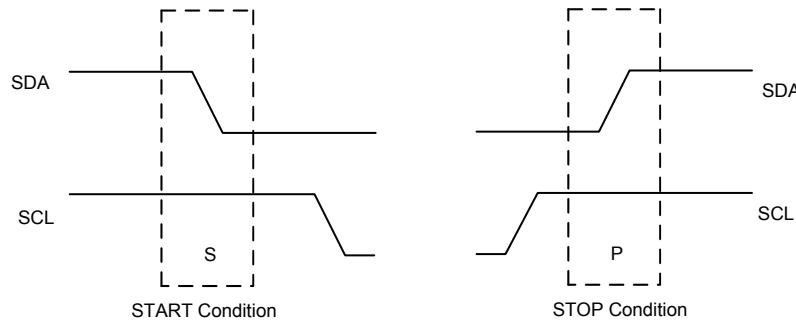
### Two Wire Serial Interface

The I<sup>2</sup>C 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 I<sup>2</sup>C 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.

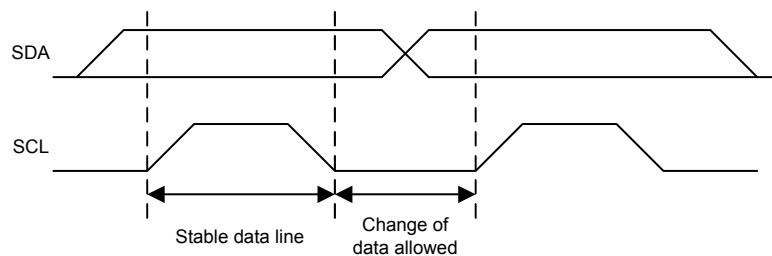
A repeated START, which is denoted as the “Sr” bit, is functionally identical to the normal START condition. A repeated START signal allows the I<sup>2</sup>C interface to communicate with another slave device or with the same device but in a different transfer direction without releasing the I<sup>2</sup>C bus control.



**Figure 132. 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 133. Data Validity**

## Addressing Format

The I<sup>2</sup>C 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/W bit and an ACK bit. The R/W bit defines the direction of the data transfer.

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

R/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.

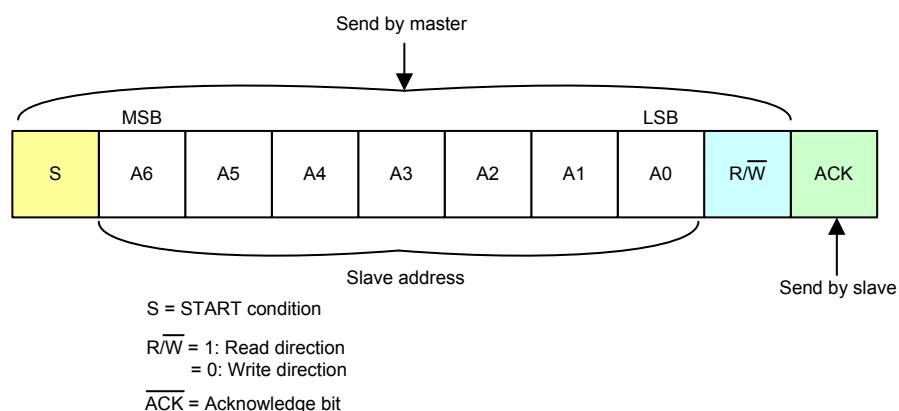
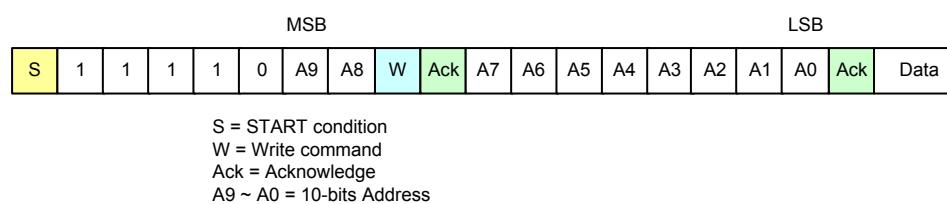


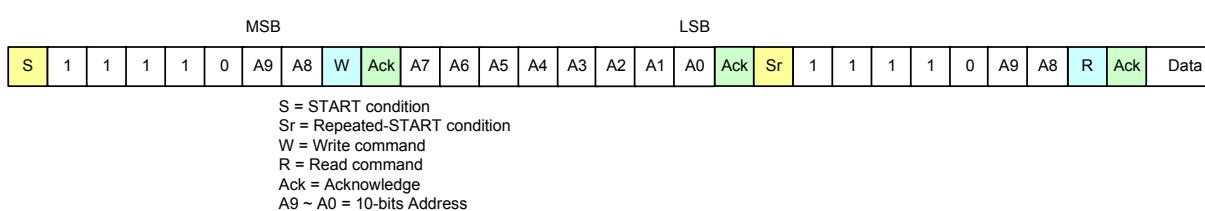
Figure 134. 7-bit Addressing Mode

## 10-bits 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”, 10<sup>th</sup> and 9<sup>th</sup> bits of the slave address. The second byte is the remaining 8 bit address of the slave device.



**Figure 135. 10-bit Addressing Write Transmit Mode**



**Figure 136. 10-bits Addressing Read Receive Mode**

## Data Transfer and Acknowledge

Once the slave device address has been matched, the data can be transmitted to or received from the slave device according to the transfer direction specified by the R/W bit. Each byte is followed by an acknowledge bit on the 9<sup>th</sup> 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.

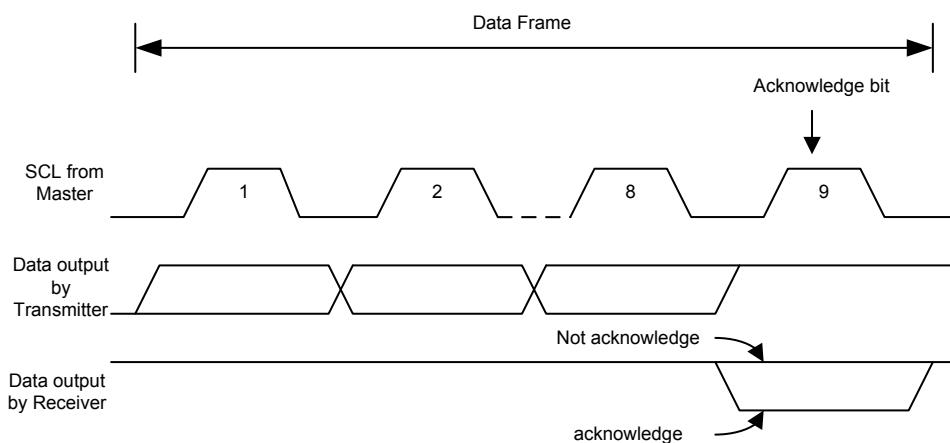


Figure 137. I<sup>2</sup>C Bus Acknowledge

## 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 I<sup>2</sup>C interface to the SCL line.

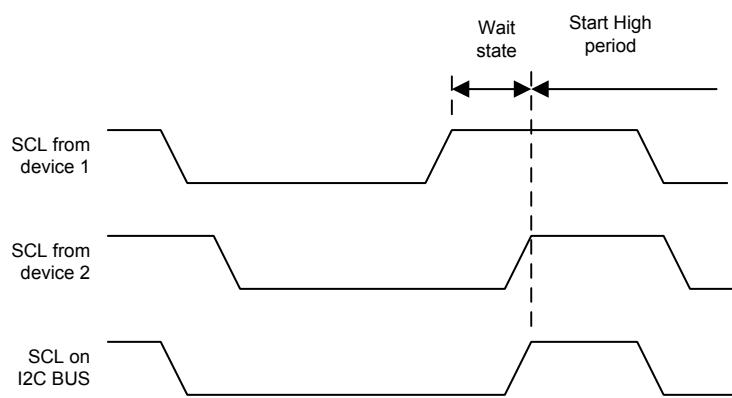


Figure 138. Clock Synchronization during Arbitration

## Arbitration

A master may start a transfer only if the I<sup>2</sup>C 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 I<sup>2</sup>C 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 I<sup>2</sup>C stop signal. When the stop signal is detected, the master which has lost the arbitration may try to access the bus again.

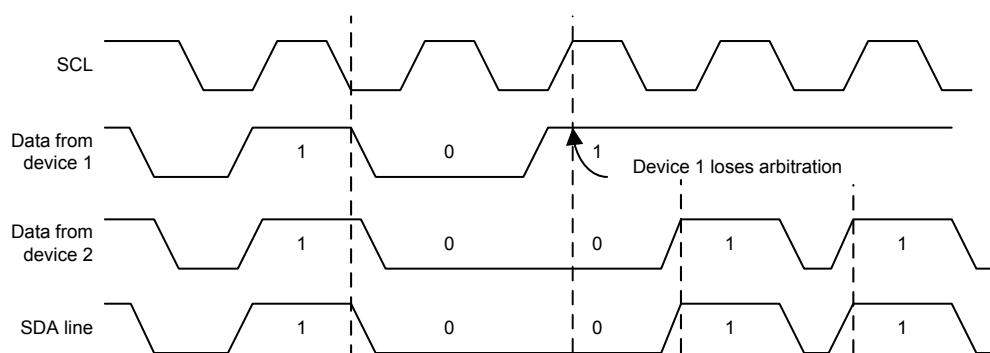


Figure 139. Two Master Arbitration Procedure

## General Call Addressing

The general call addressing function can be used to address all the devices connected to the I<sup>2</sup>C 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 I<sup>2</sup>C 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 I<sup>2</sup>C module to an idle state.

## Address Mask Enable

The I<sup>2</sup>C 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 I2CADDMR register value as 0x05h and the I2CADDR register value as 0x55h, this means if an address which is sent by an I<sup>2</sup>C master on the bus is equal to 0x50h, 0x51h, 0x54h or 0x55h, the I<sup>2</sup>C 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, I2CADDSSR, is used to monitor the calling address on the I<sup>2</sup>C bus during the whole data transfer operation no matter if the I<sup>2</sup>C module operates as a master or a slave device. Note that the I2CADDSSR register is a read only register and each calling address on the I<sup>2</sup>C bus will be stored in the I2CADDSSR register automatically even if the I<sup>2</sup>C device is not addressed.

## Operation Mode

The I<sup>2</sup>C module can operate in one of the following modes:

- Master Transmitter
- Master Receiver
- Slave Transmitter
- Slave Receiver

The I<sup>2</sup>C 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

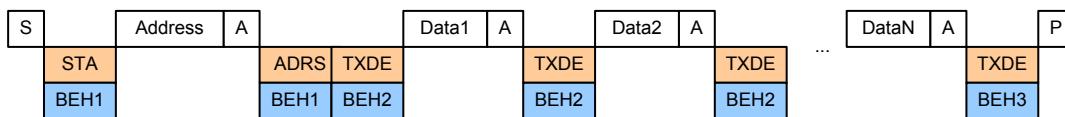
The data to be transmitted to the slave device must be transferred to the I2CDR register.

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.

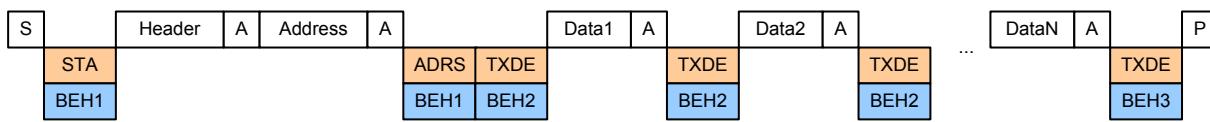
### Close / Continue Transmission

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.

#### 7-bits Master Transmitter



#### 10-bits Master Transmitter



BEH1 : cleared by reading I2CSR register

BEH2 : cleared by writing I2CDR register

BEH3 : cleared by HW automatically by sending STOP condition

**Figure 140. 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.

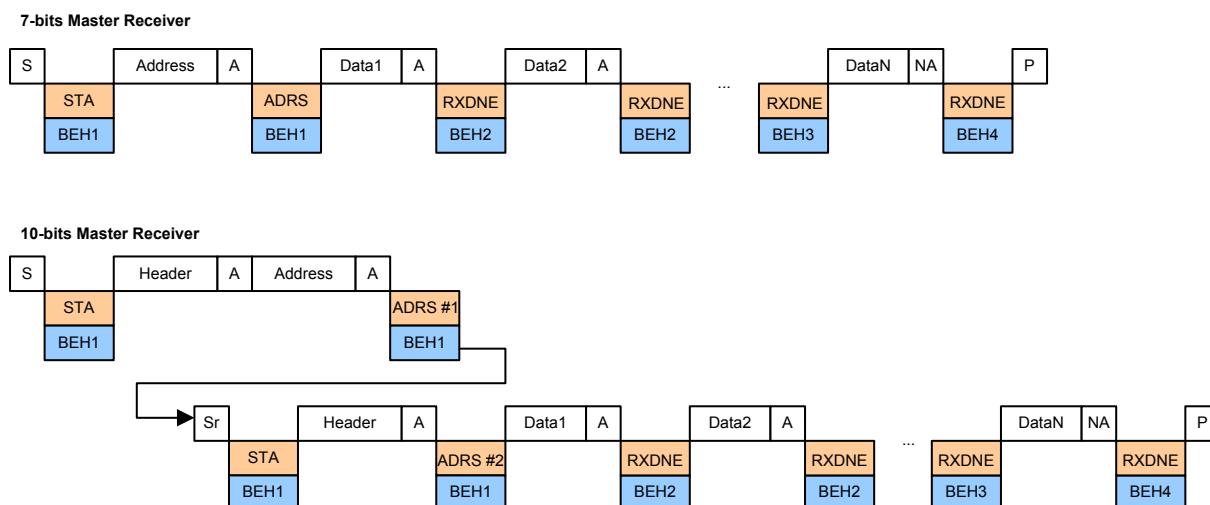
In the 10-bit addressing mode: The ADRS bit in the I2CSR register will be set twice in the 10-bit addressing mode. The first time the ADRS bit is set is when the 10-bit address is sent and the acknowledge signal from the slave device is received. The second time the ADRS bit is set is when the header 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 141. Master Receiver Timing Diagram**

## Slave Transmitter Mode

### Address Frame

In the 7-bit addressing mode, the ADRS bit in the I2CSR register is set after the slave device receives the calling address which matches with the slave device address. In the 10-bit addressing mode, the ADRS bit is set when the first header byte is matched and the second address byte is matched respectively. After the ADRS bit has been set to 1, it must be cleared to 0 to continue the data transfer process. The ADRS bit is cleared after reading the I2CSR register.

### Data Frame

In the Slave transmitter mode, the TXDE bit is set to indicate that the I2CDR is empty, which results in the SCL line being held at a logic low state. New transmission data must then be written into the I2CDR register to continue the data transfer process. Writing a data into the I2CDR register will clear the TXDE bit.

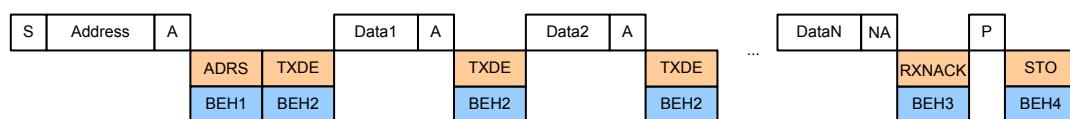
### Receive Not-Acknowledge

When the slave device receives a Not-Acknowledge signal, the RXNACK bit in the I2CSR Register is set but it will not hold the SCL line. Writing “1” to RXNACK will clear the RXNACK flag.

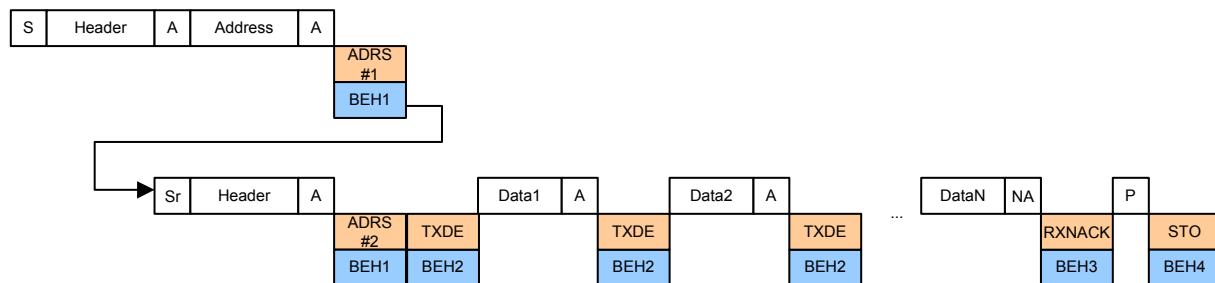
### STOP Condition

When the slave device detects a STOP condition, the STO bit in the I2CSR register is set to indicate that the I<sup>2</sup>C interface transmission is terminated. Reading the I2CSR register can clear the STO flag.

7-bits Slave Transmitter



10-bits Slave Transmitter



BEH1 : cleared by reading I2CSR register

BEH2 : cleared by writing I2CDR register

BEH3 : cleared by writing 1 clear for RXNACK flag, TXDE is not set when NACK is received.

BEH4 : cleared by reading I2CSR register

**Figure 142. Slave Transmitter Timing Diagram**

## Slave Receiver Mode

### Address Frame

The ADRS bit in the I2CSR register is set after the slave device receives the calling address which matches with the slave device address. After the ADRS bit has been set to 1, it must be cleared to 0 to continue the data transfer process. The ADRS flag is cleared after reading the I2CSR register.

### Data Frame

In the slave receiver mode, the data is transmitted from the master device. Once a data byte is received by the slave device, the RXDNE flag in the I2CSR register is set but it will not hold the SCL line. However, if the device receives a complete new data byte and the RXDNE bit has been set to 1, the RXBF bit in the I2CSR register will be set to 1 and the SCL line will be held at a logic low state. When this situation occurs, data from the I2CDR register should be read to continue the data transfer process. The RXDNE flag bit can be cleared after reading the I2CDR register.

### STOP condition

When the slave device detects a STOP condition, the STO flag bit in the I2CSR register is set to indicate that the I<sup>2</sup>C interface transmission is terminated. Reading the I2CSR register can clear the STO flag bit.

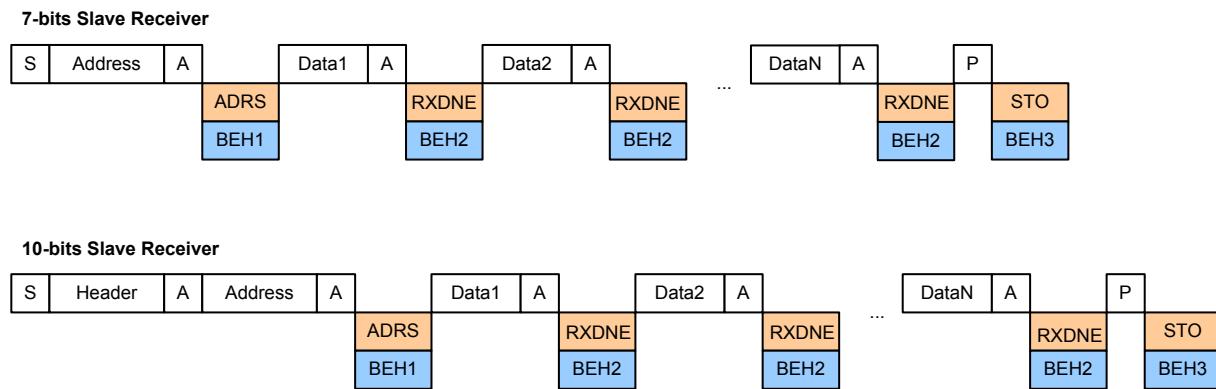


Figure 143. Slave Receiver Timing Diagram

## 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 I<sup>2</sup>C 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 <sup>2</sup> 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 <sup>2</sup> C is addressed as slave through general call	Reading I2CSR register
	ADRS	Master: I <sup>2</sup> C is sent over address frame and is returned an ACK from slave (Note: Reference Fig.139 and Fig.140) Slave: I <sup>2</sup> C is addressed as slave device (Note: Reference Fig.141 and Fig.142)	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.139, and RXNACK flag won't be assert at this case)	Set TAR Set STOP

## I<sup>2</sup>C Timeout Function

In order to reduce the occurrence of I<sup>2</sup>C lockup problem due to the reception of erroneous clock source, a timeout function is provided. If the I<sup>2</sup>C bus clock source is not received for a certain timeout period, then a corresponding I<sup>2</sup>C 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 I<sup>2</sup>C timeout clock, f<sub>I<sup>2</sup>CTO</sub>, 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 I<sup>2</sup>C master module sends a START signal.
- The I<sup>2</sup>C 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 I<sup>2</sup>C slave module is not addressed.
- The I<sup>2</sup>C slave module detects a STOP signal.
- The I<sup>2</sup>C 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.

## Register Map

The following table shows the I<sup>2</sup>C registers and reset values.

**Table 47. I<sup>2</sup>C Register Map**

Register	Offset	Description	Reset Value
I2CCR	0x000	I <sup>2</sup> C Control Register	0x0000_2000
I2CIER	0x004	I <sup>2</sup> C Interrupt Enable Register	0x0000_0000
I2CADDR	0x008	I <sup>2</sup> C Address Register	0x0000_0000
I2CSR	0x00C	I <sup>2</sup> C Status Register	0x0000_0000
I2CSHPGR	0x010	I <sup>2</sup> C SCL High Period Generation Register	0x0000_0000
I2CSLPGR	0x014	I <sup>2</sup> C SCL Low Period Generation Register	0x0000_0000
I2CDR	0x018	I <sup>2</sup> C Data Register	0x0000_0000
I2CTAR	0x01C	I <sup>2</sup> C Target Register	0x0000_0000
I2CADDMR	0x020	I <sup>2</sup> C Address Mask Register	0x0000_0000
I2CADDSSR	0x024	I <sup>2</sup> C Address Snoop Register	0x0000_0000
I2CTOUT	0x028	I <sup>2</sup> C Timeout Register	0x0000_0000

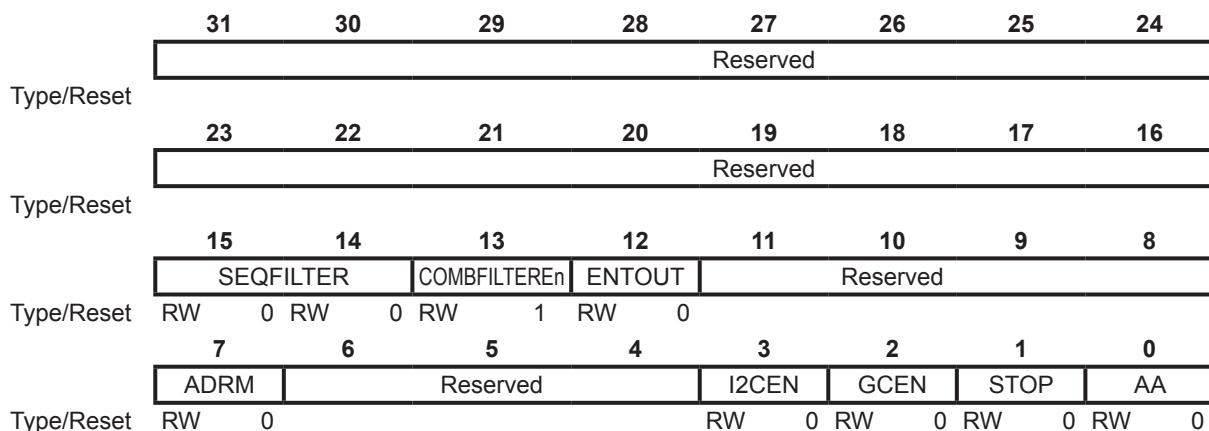
## Register Descriptions

### I<sup>2</sup>C Control Register – I2CCR

This register specifies the corresponding I<sup>2</sup>C function enable control.

Offset: 0x000 (0)

Reset value: 0x0000\_2000



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 <sup>2</sup> C Timeout Function Enable Control 0: Timeout Function disabled 1: Timeout Function enabled This bit is used to enable or disable the I <sup>2</sup> 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 ENTOUT bit to 1.
[7]	ADRM	Addressing Mode 0: 7-bit addressing mode 1: 10-bit addressing mode When the I <sup>2</sup> C 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.
[3]	I2CEN	I <sup>2</sup> C Interface Enable 0: I <sup>2</sup> C interface disabled 1: I <sup>2</sup> C interface enabled

Bits	Field	Descriptions
[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 I<sup>2</sup>C 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>

## I<sup>2</sup>C Interrupt Enable Register – I2CIER

This register specifies the corresponding I<sup>2</sup>C interrupt enable bits.

Offset: 0x004

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[18]	RXBFI	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.

<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>
[8]	ARBLOSIE	Arbitration Loss Interrupt Enable Bit in the I <sup>2</sup> 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 <sup>2</sup> 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 <sup>2</sup> C master mode only.

## I<sup>2</sup>C Address Register – I2CADDR

This register specifies the I<sup>2</sup>C device address.

Offset: 0x008

Reset value: 0x0000\_0000



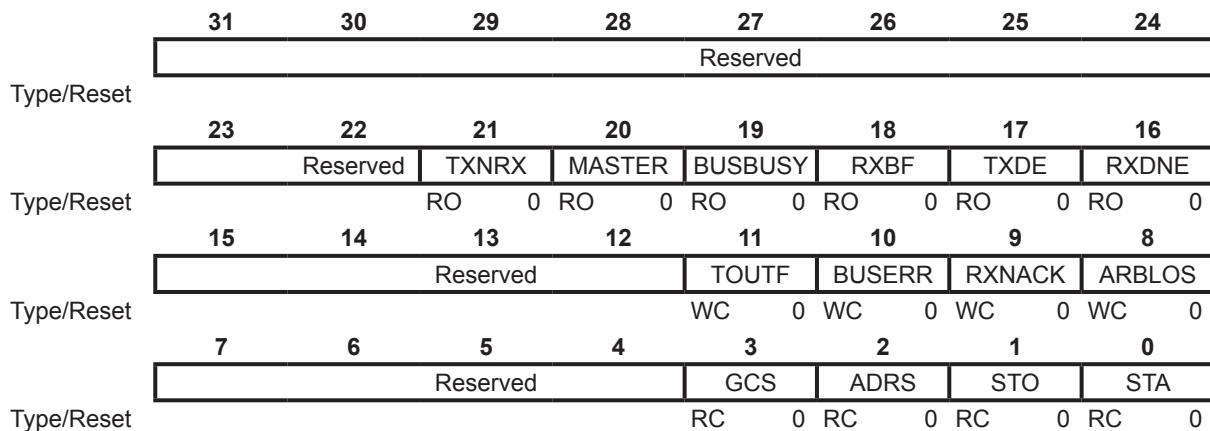
Bits	Field	Descriptions
[9:0]	ADDR	Device Address The register indicates the I <sup>2</sup> C device address. When the I <sup>2</sup> C device is used in the 7-bit addressing mode, only the ADDR[6:0] bits will be compared with the received address sent from the I <sup>2</sup> C master device.

## I<sup>2</sup>C Status Register – I2CSR

This register contains the I<sup>2</sup>C operation status.

Offset: 0x00C

Reset value: 0x0000\_0000



Bits	Field	Descriptions
[21]	TXNRX	Transmitter / Receiver Mode 0: Receiver mode 1: Transmitter mode Read only bit.
[20]	MASTER	Master Mode 0: I <sup>2</sup> C is in the slave mode or idle 1: I <sup>2</sup> C is in the master mode The I <sup>2</sup> C interface is switched as a master device on the I <sup>2</sup> C bus when the I2CTAR register is assigned and the I <sup>2</sup> C bus is idle. The MASTER bit is cleared by hardware when software disables the I <sup>2</sup> C bus by clearing the I2CEN bit to 0 or sends a STOP condition to the I <sup>2</sup> 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 <sup>2</sup> C bus is idle 1: I <sup>2</sup> C bus is busy The I <sup>2</sup> C interface hardware starts to detect the I <sup>2</sup> 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 I<sup>2</sup>C 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 I<sup>2</sup>C 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: I<sup>2</sup>C interface is addressed by a general call command</p> <p>When the I<sup>2</sup>C 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>

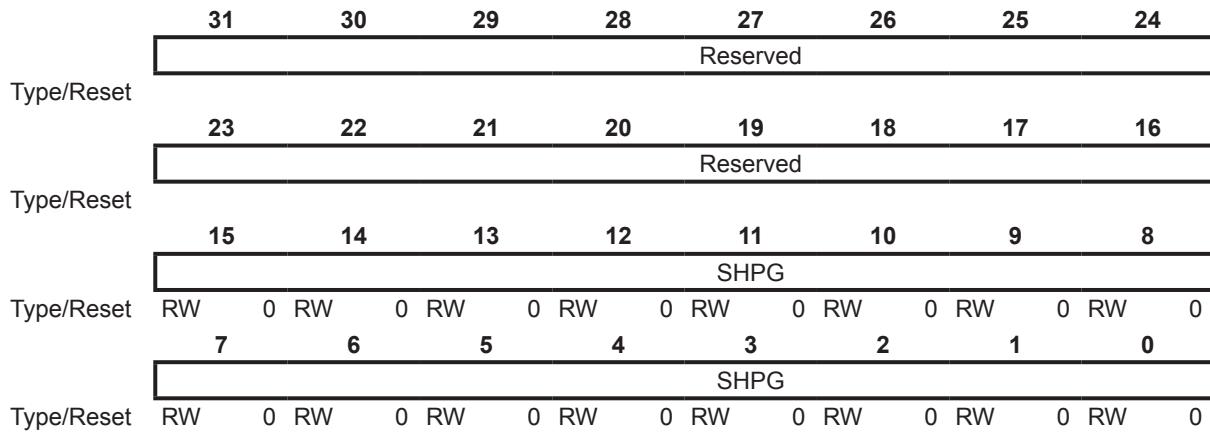
Bits	Field	Descriptions
[2]	ADRS	<p>Address Transmit (master mode) / Address Receive (slave mode) Flag</p> <p>Address Sent in Master Mode</p> <p>0: Address frame has not been transmitted 1: Address frame has been transmitted</p> <p>For the 7-bit addressing mode, this bit is set after the master device 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: I<sup>2</sup>C interface is not addressed 1: I<sup>2</sup>C interface is addressed as slave</p> <p>When the I<sup>2</sup>C 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>

## I<sup>2</sup>C SCL High Period Generation Register – I2CSHPGR

This register specifies the I<sup>2</sup>C SCL clock high period interval.

Offset: 0x010

Reset value: 0x0000\_0000



Bits	Field	Descriptions
[15:0]	SHPG	<p>SCL Clock High Period Generation</p> <p>High period duration setting <math>SCL_{HIGH} = T_{PCLK} \times (SHPG + d)</math> where <math>T_{PCLK}</math> is the APB bus peripheral clock (PCLK) period of the I<sup>2</sup>C, and d value depends on the setting of the SEQFILTER in the I<sup>2</sup>C Control Register (I2CCR).</p> <ul style="list-style-type: none"> <li>If SEQFILTER =00, d=6</li> <li>If SEQFILTER =01, d=8</li> <li>If SEQFILTER =10 or 11, d=9</li> </ul>

## I<sup>2</sup>C SCL Low Period Generation Register – I2CSLPGR

This register specifies the I<sup>2</sup>C SCL clock low period interval.

Offset: 0x014

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	RW	0	RW	0	RW	0	RW	0	
	7	6	5	4	3	2	1	0	
Type/Reset	RW	0	RW	0	RW	0	RW	0	
	SLPG								
	7	6	5	4	3	2	1	0	
Type/Reset	RW	0	RW	0	RW	0	RW	0	
	SLPG								
	7	6	5	4	3	2	1	0	
Type/Reset	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 <math>T_{SCL} = T_{PCLK} \times (SLPG + d)</math> where <math>T_{PCLK}</math> is the APB bus peripheral clock (PCLK) period of I<sup>2</sup>C, and d value depends on the setting of the SEQFILTER in the I<sup>2</sup>C Control Register (I2CCR).</p> <p>If SEQFILTER =00, d=6 If SEQFILTER =01, d=8 If SEQFILTER =10 or 11, d=9</p>

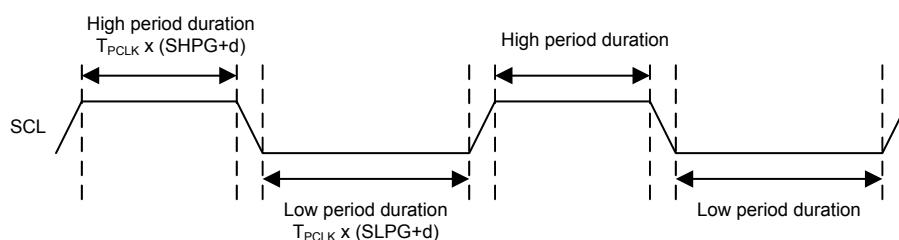


Figure 144. SCL Timing Diagram

Table 48. I<sup>2</sup>C Clock Setting Example

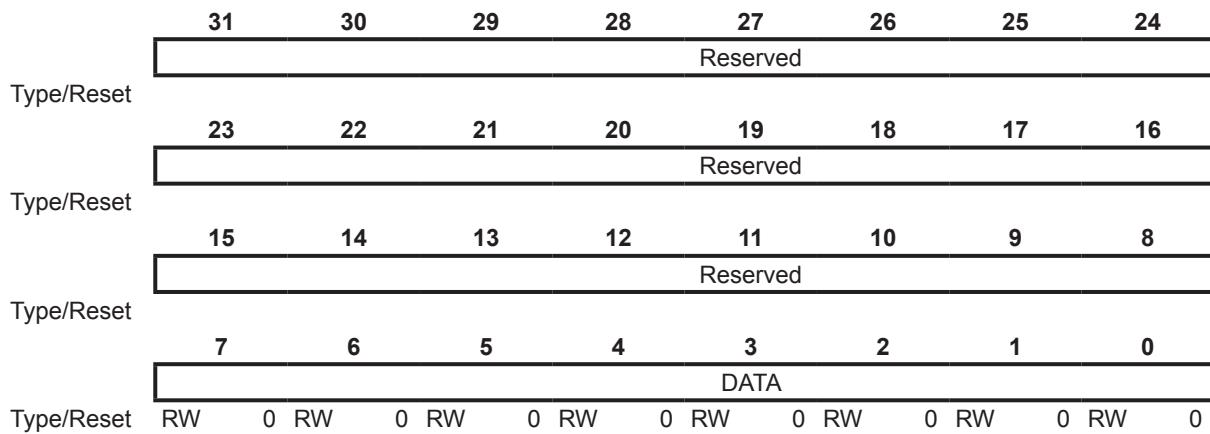
I <sup>2</sup> C Clock	$T_{SCL} = T_{PCLK} \times [ (SHPG + d) + (SLPG + d) ]$ (where d = 6) SHPG + SLPG value at PCLK		
	8MHz	24MHz	48MHz
100 kHz (Standard Mode)	68	228	468
400 kHz (Fast Mode)	8	48	108
1 MHz (Fast-Mode Plus)	x	12	36

## I<sup>2</sup>C Data Register – I2CDR

This register specifies the data to be transmitted or received by the I<sup>2</sup>C module.

Offset: 0x018

Reset value: 0x0000\_0000



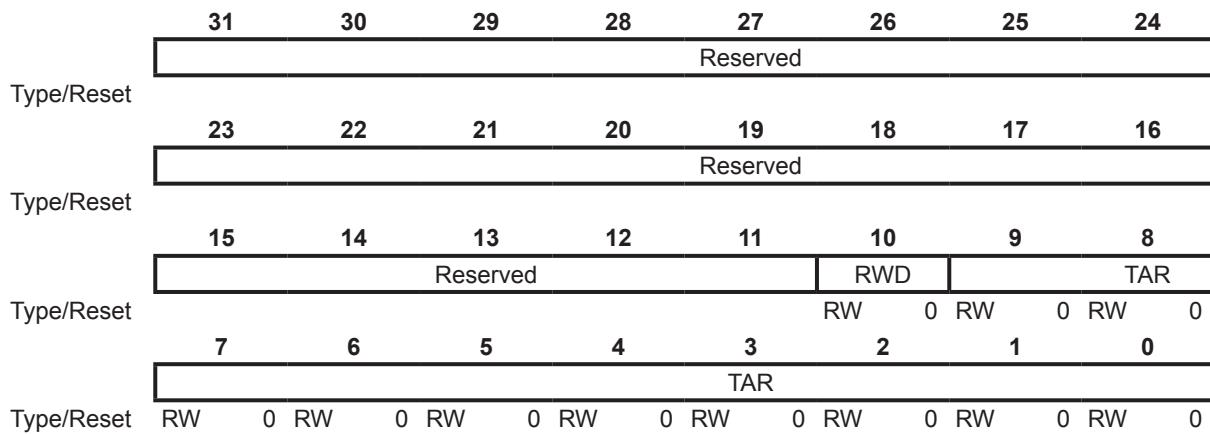
Bits	Field	Descriptions
[7:0]	DATA	<p>I<sup>2</sup>C Data Register</p> <p>For the transmitter mode, a data byte which is transmitted to a slave device can be assigned to these bits. The TXDE flag is cleared if the application software assigns new data to the I2CDR register. For the receiver mode, a data byte is received bit by bit from MSB to LSB through the I<sup>2</sup>C 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>

## I<sup>2</sup>C Target Register – I2CTAR

This register specifies the target device address to be communicated.

Offset: 0x01C

Reset value: 0x0000\_0000



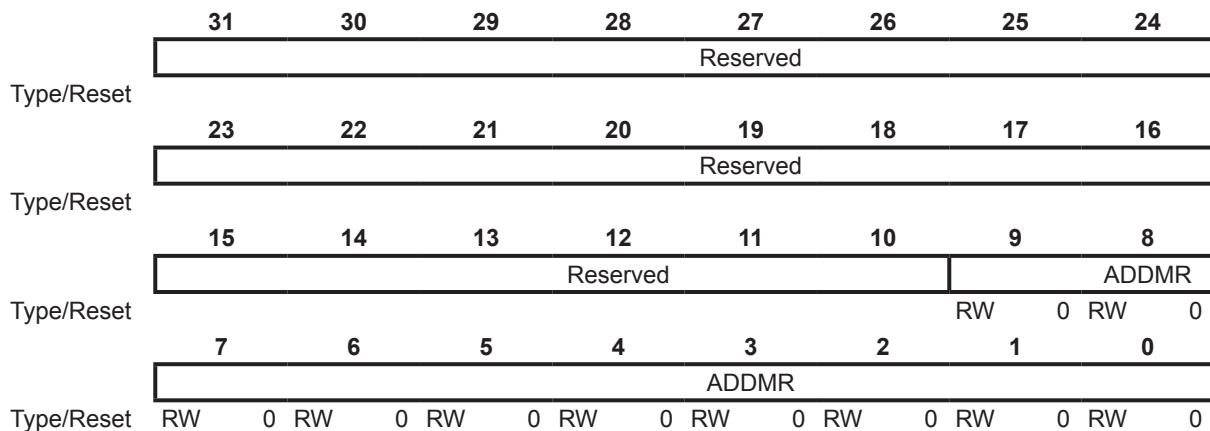
Bits	Field	Descriptions
[10]	RWD	Read or Write Direction  0: Write direction to target slave address 1: Read direction from target slave address  If this bit is set to 1 in the 10-bit master receiver mode, the I <sup>2</sup> C interface will initiate a byte with a value of 11110XX0b in the first header 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 <sup>2</sup> 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 <sup>2</sup> 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.

## I<sup>2</sup>C Address Mask Register – I2CADDMR

This register specifies which bit of the I<sup>2</sup>C address is masked and not compared with corresponding bit of the received address frame.

Offset: 0x020

Reset value: 0x0000\_0000



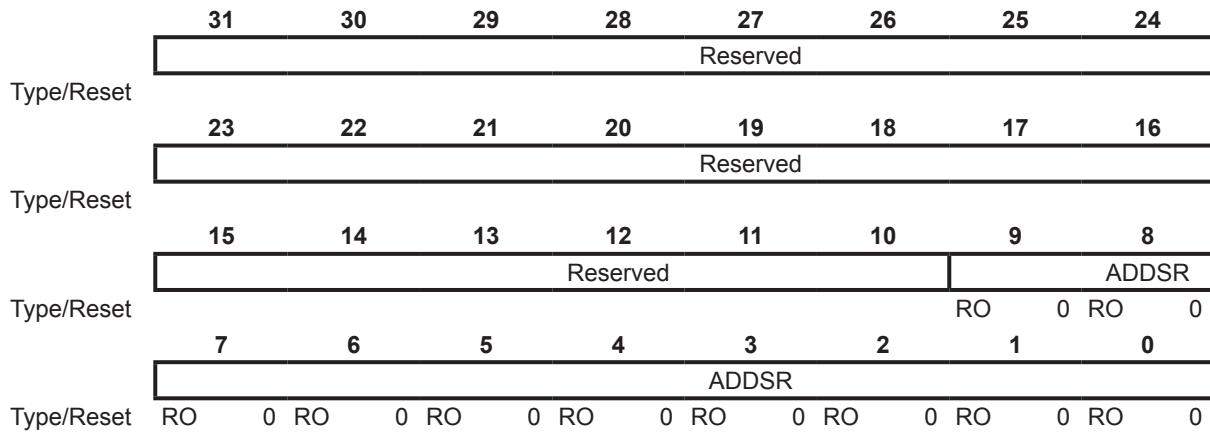
Bits	Field	Descriptions
[9:0]	ADDMR	<p>Address Mask Control Bit</p> <p>The ADDMR[i] is used to specify whether the i<sup>th</sup> bit of the ADDR in the I2CADDR register is masked and is compared with the received address frame or not on the I<sup>2</sup>C bus. The register is only used for the I<sup>2</sup>C slave mode only.</p> <p>0: i<sup>th</sup> bit of the ADDR is compared with the address frame on the I<sup>2</sup>C bus.</p> <p>1: i<sup>th</sup> bit of the ADDR is masked and not compared with the address frame on the I<sup>2</sup>C bus.</p>

## I<sup>2</sup>C Address Snoop Register – I2CADDSSR

This register is used to indicate the address frame value appeared on the I<sup>2</sup>C bus.

Offset: 0x024

Reset value: 0x0000\_0000



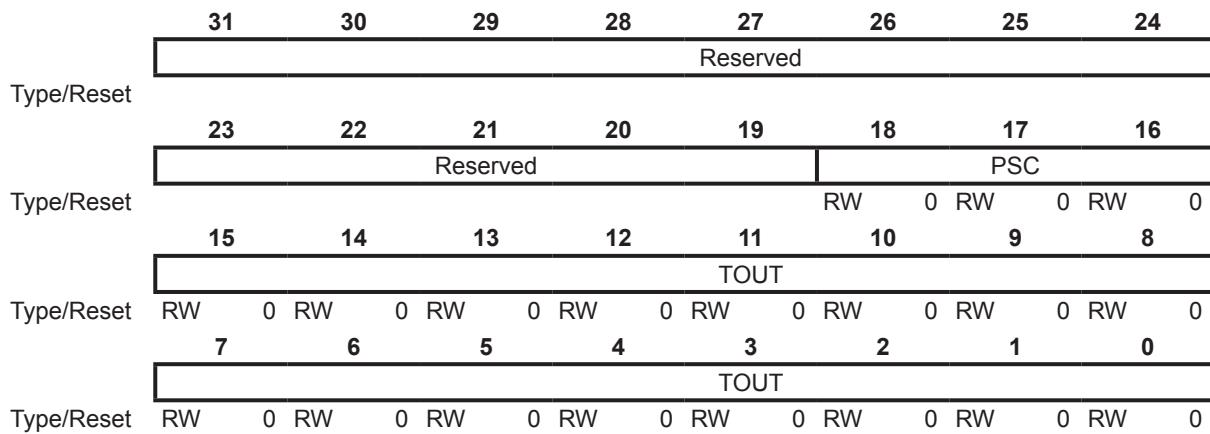
Bits	Field	Descriptions
[9:0]	ADDSR	Address Snoop Once the I2CEN bit is enabled, the calling address value on the I <sup>2</sup> C bus will automatically be loaded into this ADDSR field.

## I<sup>2</sup>C Timeout Register – I2CTOUT

This register specifies the I<sup>2</sup>C Timeout counter preload value and clock prescaler ratio.

Offset: 0x028

Reset value: 0x0000\_0000



Bits	Field	Descriptions
[18:16]	PSC	<p>I<sup>2</sup>C Time-out Counter Prescaler Selection</p> <p>This PSC field is used to specify the I<sup>2</sup>C time-out counter clock frequency, <math>f_{I2CTO}</math>. The time-out clock frequency is obtained using the formula:</p> $f_{I2CTO} = \frac{f_{PCLK}}{2^{PSC}}$ <p>PSC=0 → <math>f_{I2CTO} = f_{PCLK} / 2^0 = f_{PCLK}</math>      PSC=1 → <math>f_{I2CTO} = f_{PCLK} / 2^1 = f_{PCLK} / 2</math>      PSC=2 → <math>f_{I2CTO} = f_{PCLK} / 2^2 = f_{PCLK} / 4</math>      ...      PSC=7 → <math>f_{I2CTO} = f_{PCLK} / 2^7 = f_{PCLK} / 128</math></p>
[15:0]	TOUT	<p>I<sup>2</sup>C 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"> <li>1. The RXBF, TXDE, RXDNE, RXNACK, GCS or ADRS flag in the I2CSR register is asserted.</li> <li>2. The I<sup>2</sup>C master module sends a START signal.</li> <li>3. The I<sup>2</sup>C slave module detects a START signal.</li> </ol> <p>The counter stops counting as the following conditions occur:</p> <ol style="list-style-type: none"> <li>1. The I<sup>2</sup>C slave device is not addressed.</li> <li>2. The I<sup>2</sup>C master module sends a STOP signal.</li> <li>3. The I<sup>2</sup>C slave module detects a STOP signal.</li> <li>4. The ARBLOS or BUSERR flag in the I2CSR register is asserted.</li> </ol>

# 20 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.

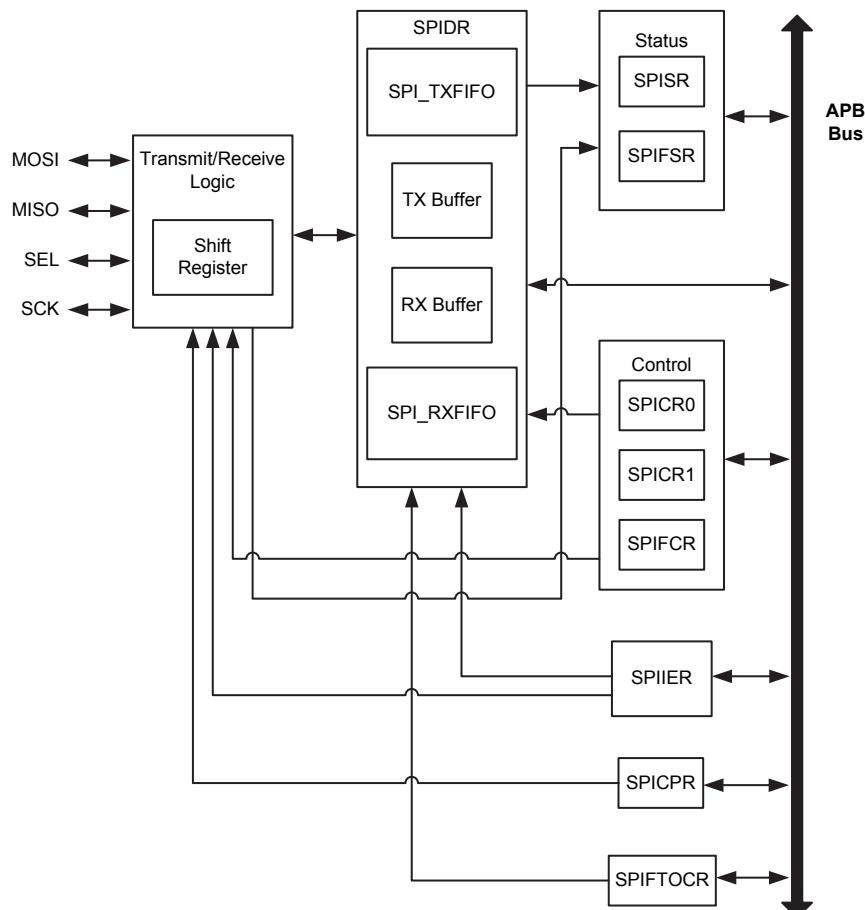


Figure 145. SPI Block Diagram

## 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

## 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 base 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.

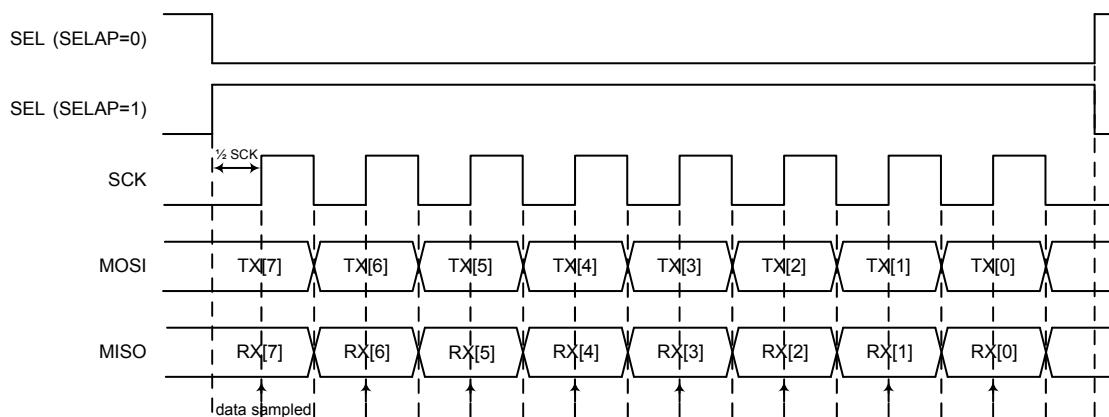
There are four formats contained in the SPI interface. Table 48 shows how to configure these formats by setting the FORMAT field in the SPICR1 register.

**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	

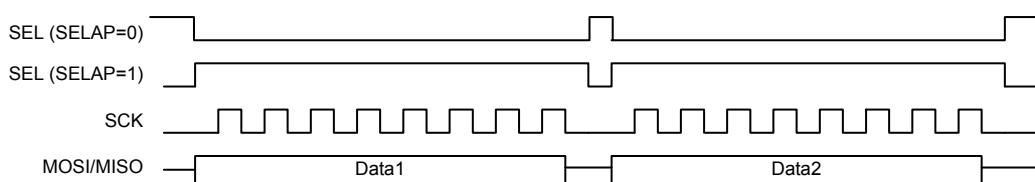
### CPOL = 0, CPHA = 0

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 145 shows the single byte data transfer timing of this format.



**Figure 146. SPI Single Byte Transfer Timing Diagram – CPOL = 0, CPHA = 0**

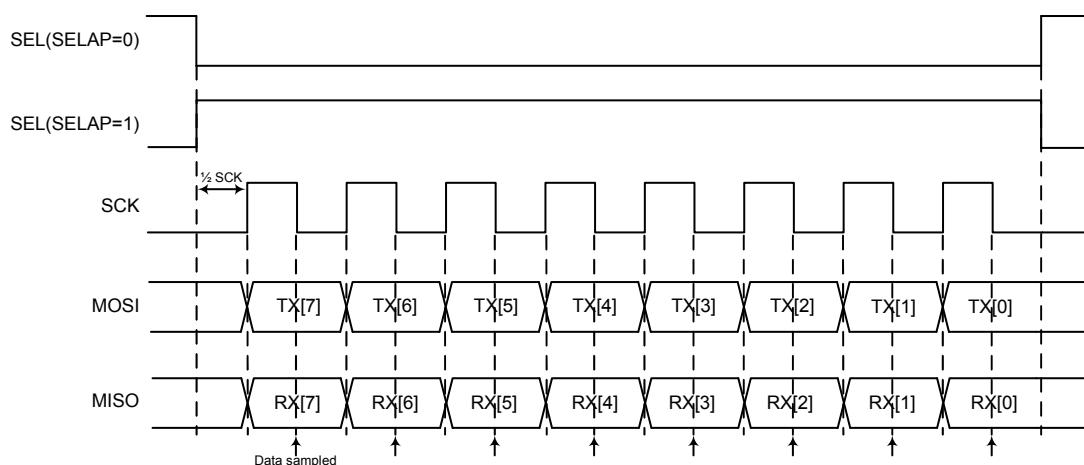
Figure 146 shows the continuous data transfer timing diagram of this format. Note that the SEL signal must change to an inactive level between each data frame.



**Figure 147. SPI Continuous Data Transfer Timing Diagram – CPOL = 0, CPHA = 0**

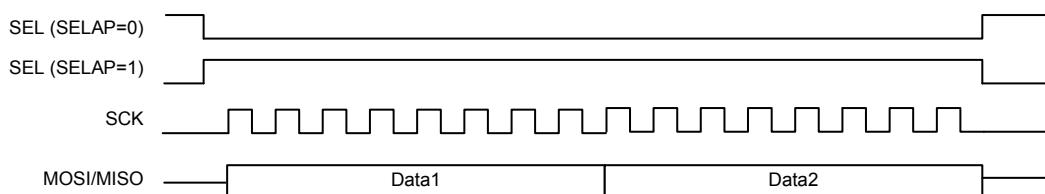
#### CPOL = 0, CPHA = 1

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 147 shows the single data byte transfer timing.



**Figure 148. SPI Single Byte Transfer Timing Diagram – CPOL = 0, CPHA = 1**

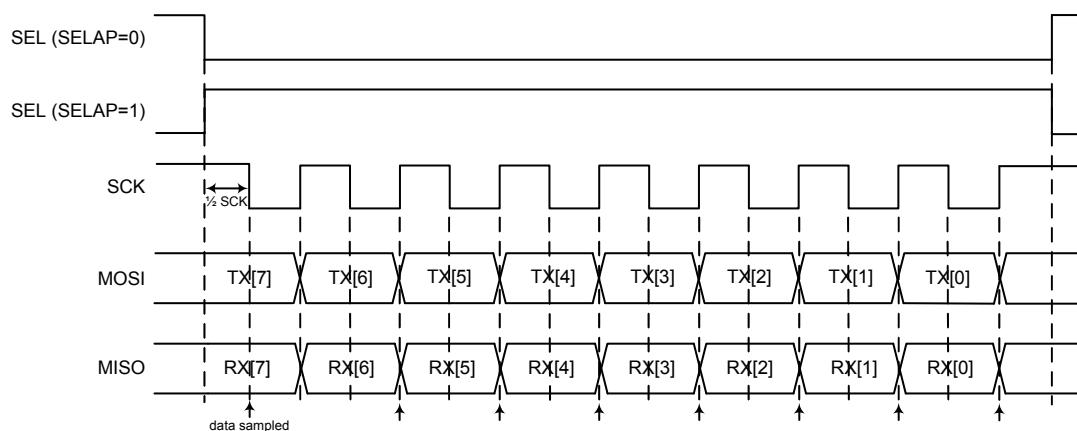
Figure 148 shows the continuous data transfer diagram timing. Note that the SEL signal must remain active until the last data transfer has completed.



**Figure 149. SPI Continuous Transfer Timing Diagram – CPOL = 0, CPHA = 1**

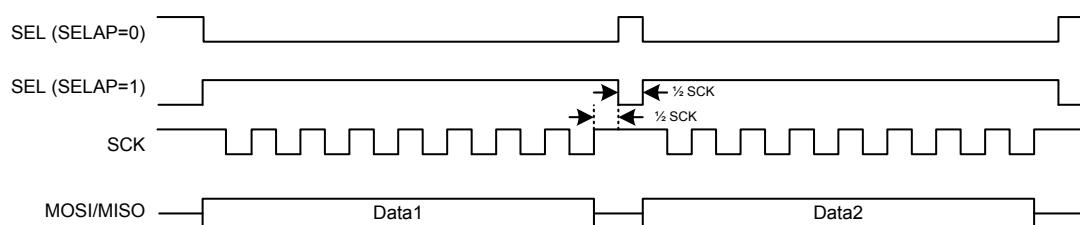
#### CPOL = 1, CPHA = 0

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 149 shows the single byte transfer timing of this format.



**Figure 150. SPI Single Byte Transfer Timing Diagram – CPOL = 1, CPHA = 0**

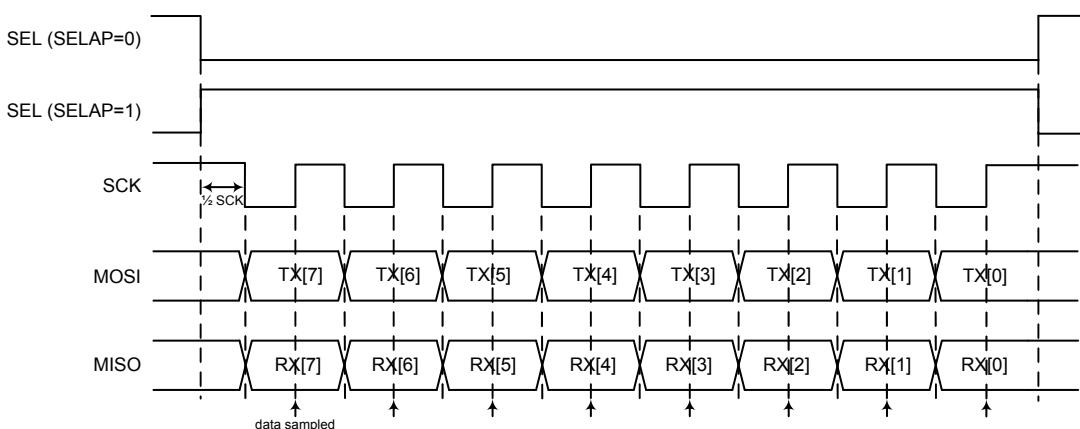
Figure 150 shows the continuous data transfer timing of this format. Note that the SEL signal must change to an inactive level between each data frame.



**Figure 151. SPI Continuous Transfer Timing Diagram – CPOL = 1, CPHA = 0**

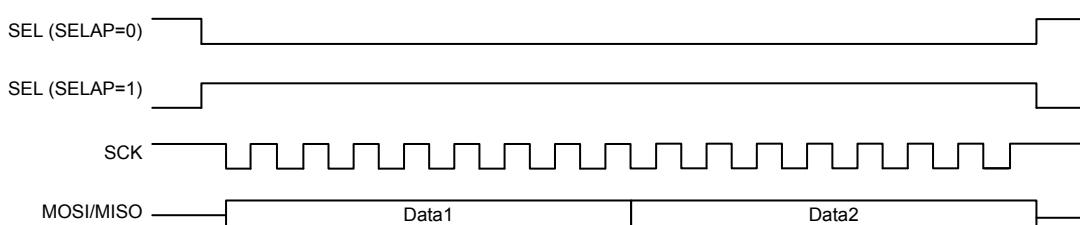
#### CPOL = 1, CPHA = 1

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 151 shows the single byte transfer timing of this format.



**Figure 152. SPI Single Byte Transfer Timing Diagram – CPOL = 1, CPHA = 1**

Figure 152 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 153. 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

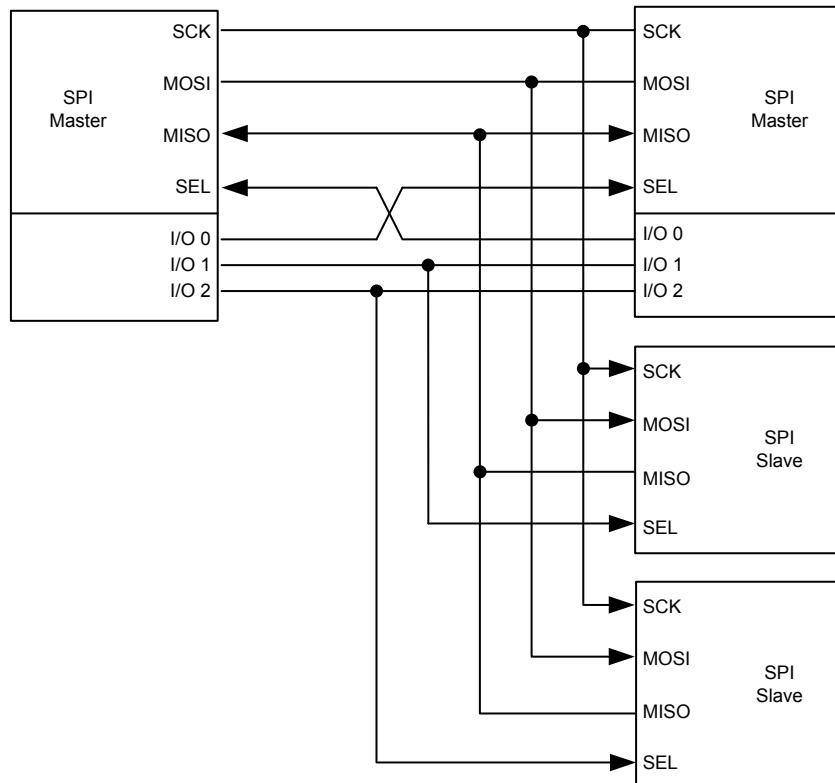
This RXBNE flag is set when there is valid received data in the RX Buffer in the non-FIFO mode or the RX FIFO data length is equal to or greater than the RX FIFO threshold level as defined by the RXFTLS field in the SPIFCR register in the SPI FIFO mode. This flag will be automatically cleared by hardware 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 154. SPI Multi-Master Slave Environment**

**Table 50. SPI Mode Fault Trigger Conditions**

Mode fault	Descriptions
Trigger condition	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	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	
	Case 1	Case 2	Case 1	Case 2
Continuous transfer	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 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.

## 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

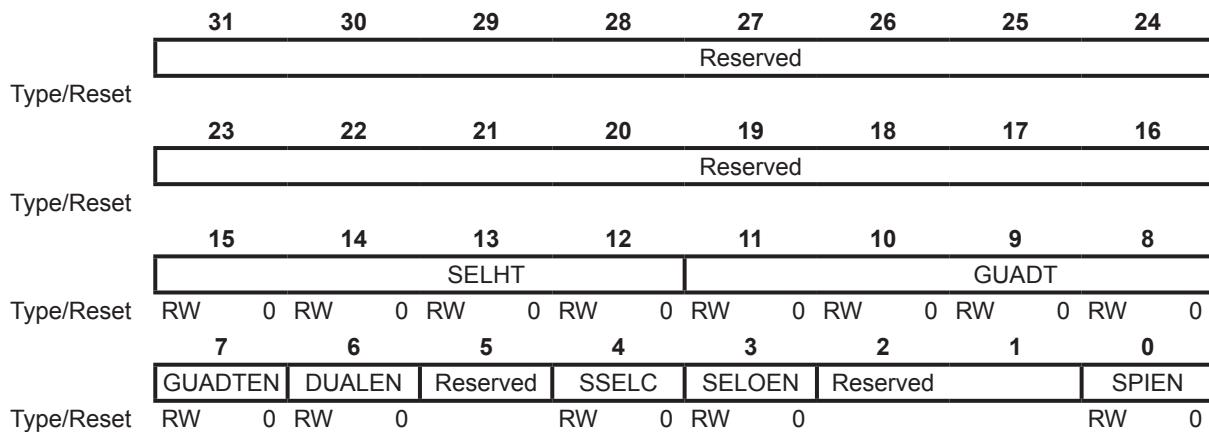
## Register Descriptions

### 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.
[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.

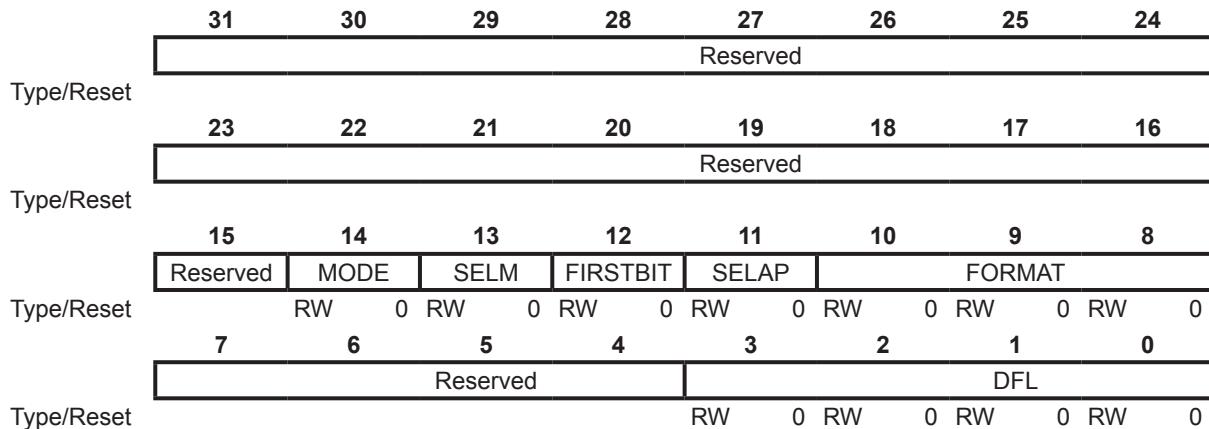
Bits	Field	Descriptions
[4]	SSEL C	Software Slave Select Control 0: Set the SEL output to an inactive state 1: Set the SEL output to an active state The application Software can setup the SEL output to an active or inactive state by configuring the SSEL C bit. The active level is configured by the SELAP bit in the SPICR1 register. Note that the SSEL C 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 SSEL C 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
[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



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 SSELC 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</p> <p>0: SCK Idle state is low 1: SCK Idle state is high</p> <p>CPHA: Clock Phase</p> <p>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	RW	0	RW	0	RW	0	RW	0	
	TOIEN	SAIEN	MFIEN	ROIEN	WCIEN	RXBNEIEN	TXEIEN	TXBEIEN	

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

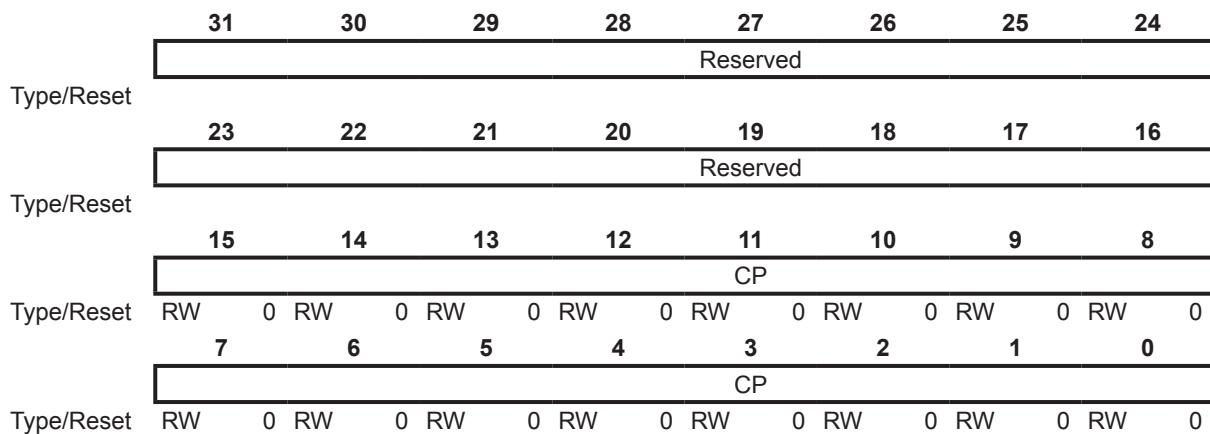
Bits	Field	Descriptions
[0]	TXBEIEN	TX Buffer Empty Interrupt Enable 0: Disable 1: Enable The TX buffer empty interrupt request will be generated when the TXBE flag and the TXBEIEN bit are set. In the FIFO mode, the interrupt request being generated depends upon the TX FIFO trigger level setting.

### SPI Clock Prescaler Register – SPICPR

This register specifies the SPI clock prescaler ratio.

Offset: 0x00C

Reset value: 0x0000\_0000



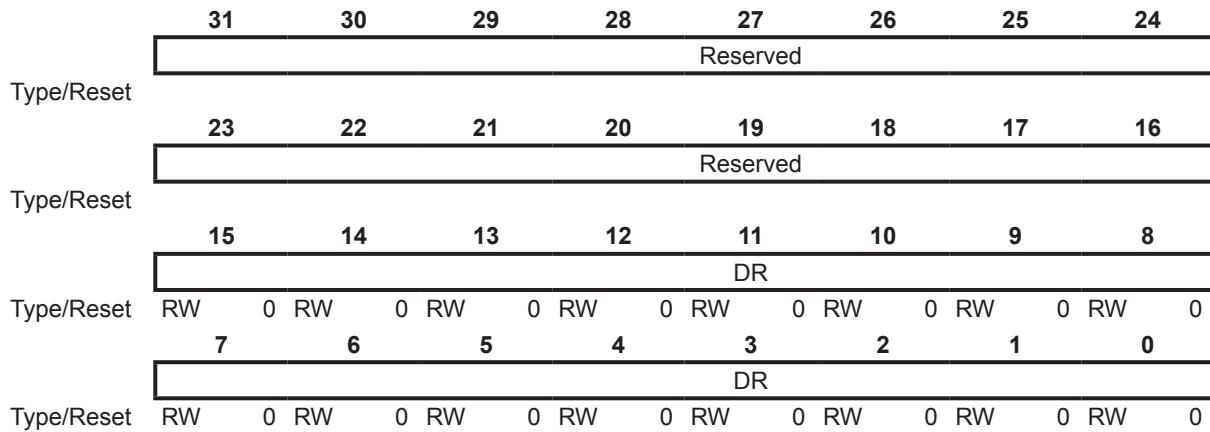
Bits	Field	Descriptions
[15:0]	CP	SPI Clock Prescaler The SPI clock (SCK) is determined by the following equation: $f_{SCK} = f_{PCLK} / (2 \times (CP + 1))$ , where the CP ranges is from 0 to 65535 Note: For the SPI slave mode, the system clock ( $f_{PCLK}$ ) must be at least 3 times faster than the external SPI SCK input.

## SPI Data Register – SPIDR

This register stores the SPI received or transmitted Data.

Offset: 0x010

Reset value: 0x0000\_0000



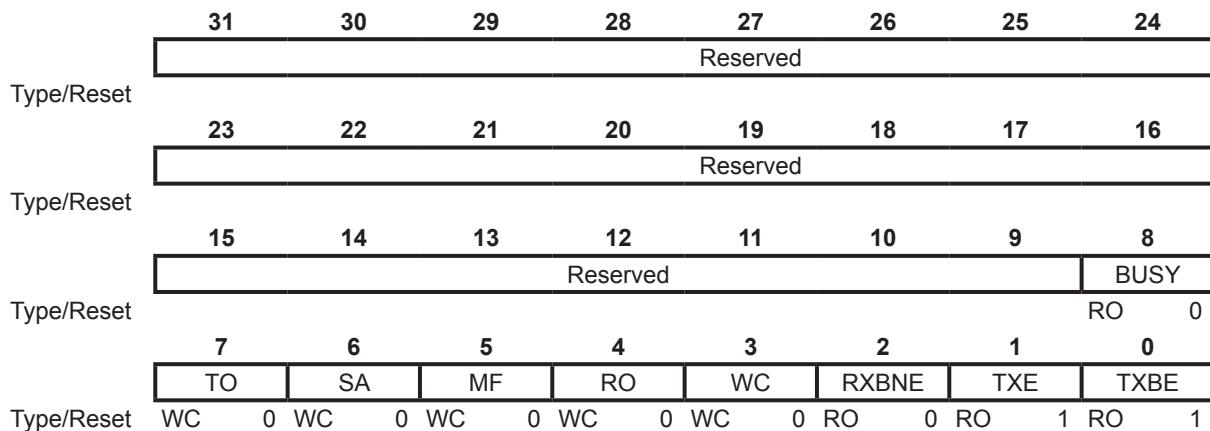
Bits	Field	Descriptions
[15:0]	DR	<p>Data Register</p> <p>The SPI data register is used to store the serial bus transmitted or received data. In the non-FIFO mode, writing data into the SPI data register will also load the data into the data transmission buffer, known as the TX buffer. Reading data from the SPI data register will return the data held in the data received buffer, named RX buffer.</p>

## SPI Status Register – SPISR

This register contains the relevant SPI status.

Offset: 0x014

Reset value: 0x0000\_0003



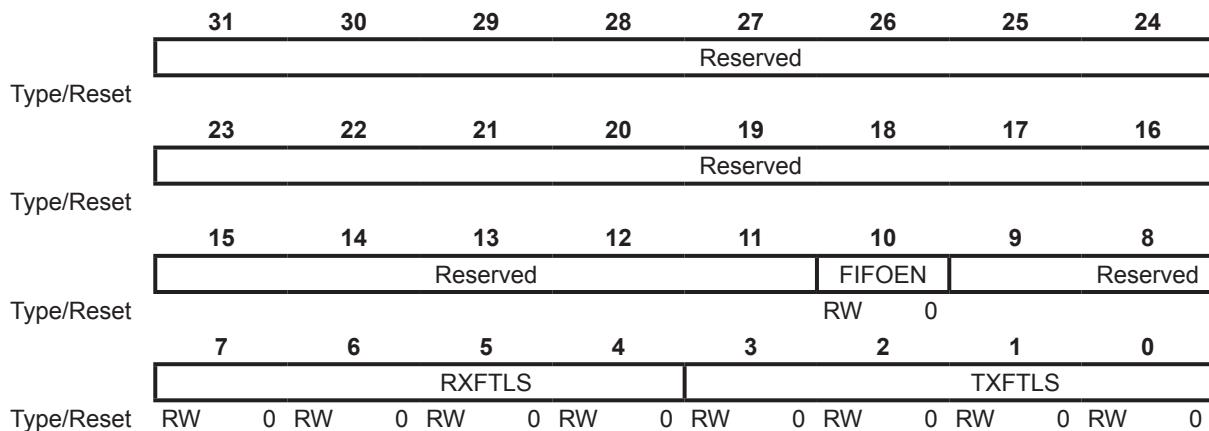
Bits	Field	Descriptions
[3]	WC	Write Collision flag 0: No write collision 1: Write collision has occurred. This bit is set by hardware and cleared by writing 1.
[2]	RXBNE	Receive Buffer Not Empty flag 0: RX buffer empty 1: RX buffer not empty This bit indicates the RX buffer status in the non-FIFO mode.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



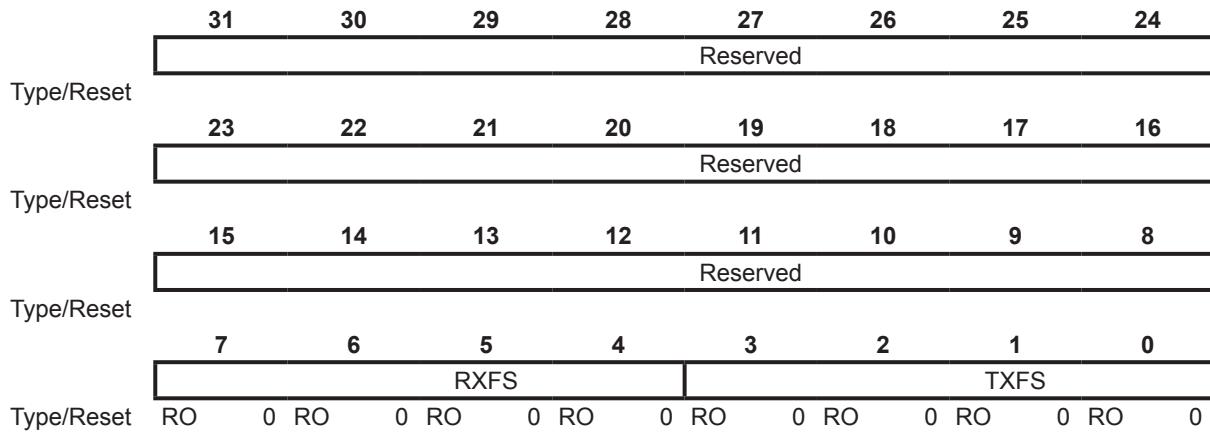
Bits	Field	Descriptions
[10]	FIFOEN	FIFO Enable 0: FIFO disable 1: FIFO enable This bit cannot be set or reset when the SPI interface is in transmitting.
[7:4]	RXFTLS	Rx FIFO Trigger Level Select 0000: Trigger level is 0 0001: Trigger level is 1 ... 1000: Trigger level is 8 Others: Reserved The RXFTLS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



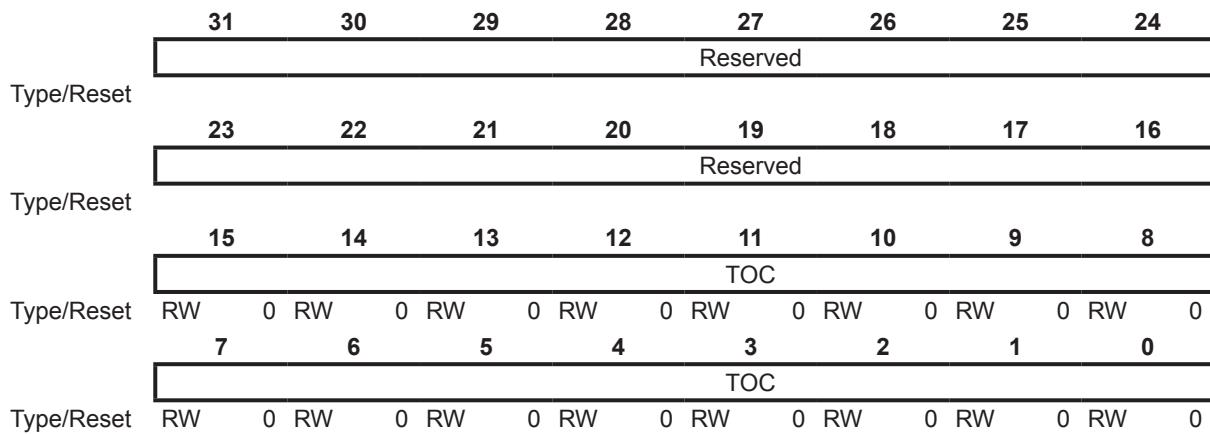
Bits	Field	Descriptions
[7:4]	RXFS	Rx FIFO Status 0000: Rx FIFO empty 0001: Rx FIFO contains 1 data ... 1000: Rx FIFO contains 8 data Others: Reserved
[3:0]	TXFS	Tx FIFO Status 0000: Tx FIFO empty 0001: Tx FIFO contains 1 data ... 1000: Tx FIFO contains 8 data Others: Reserved

## SPI FIFO Time Out Counter Register – SPIFTOCR

This register stores the SPI Rx FIFO time out counter value.

Offset: 0x020

Reset value: 0x0000\_0000



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 SPIIEN 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 <math>f_{PCLK}</math></p>

# 21 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.

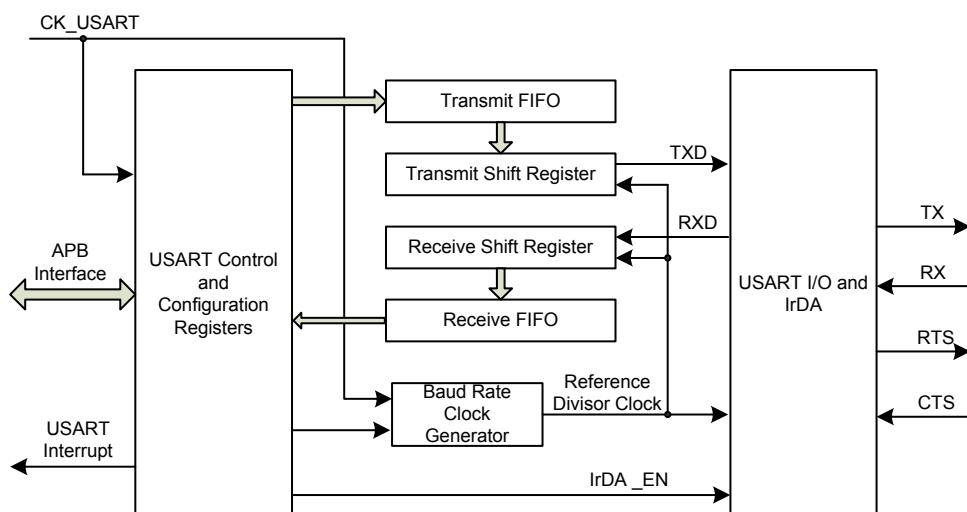


Figure 155. USART Block Diagram

## 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 us) 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 × 9 bits (max 9 data bits)
  - Transmit FIFO: 8 × 9 bits (max 9 data bits)

## 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.

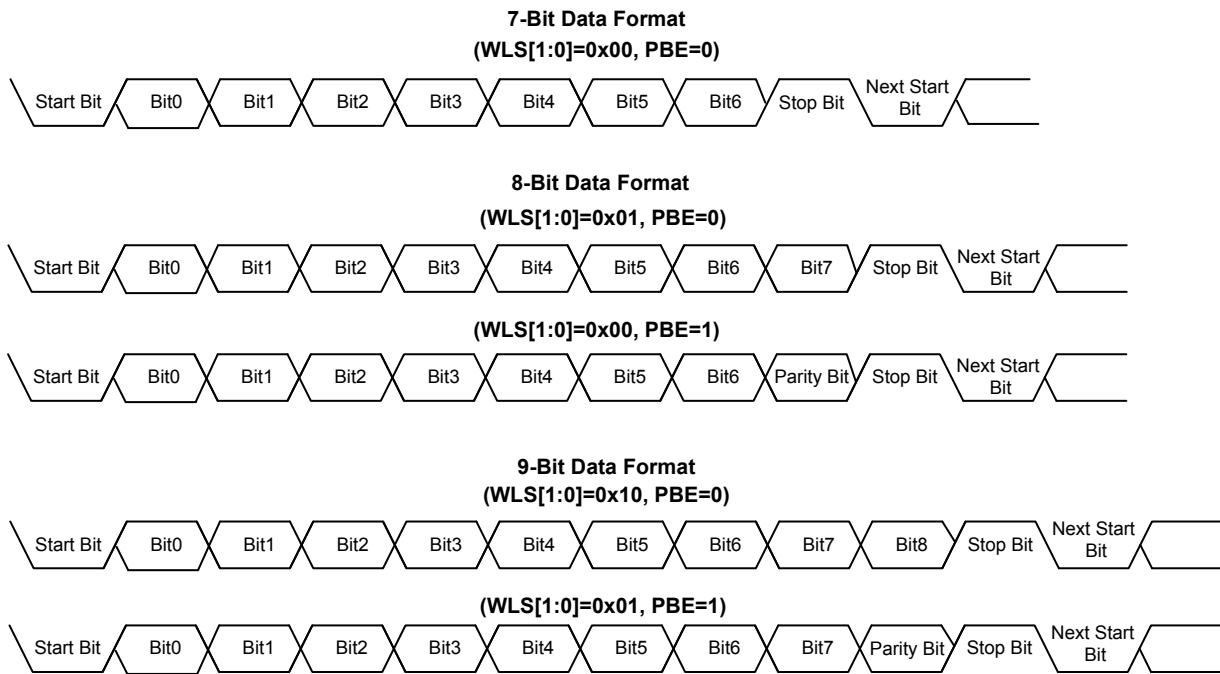


Figure 156. USART Serial Data Format

## 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.

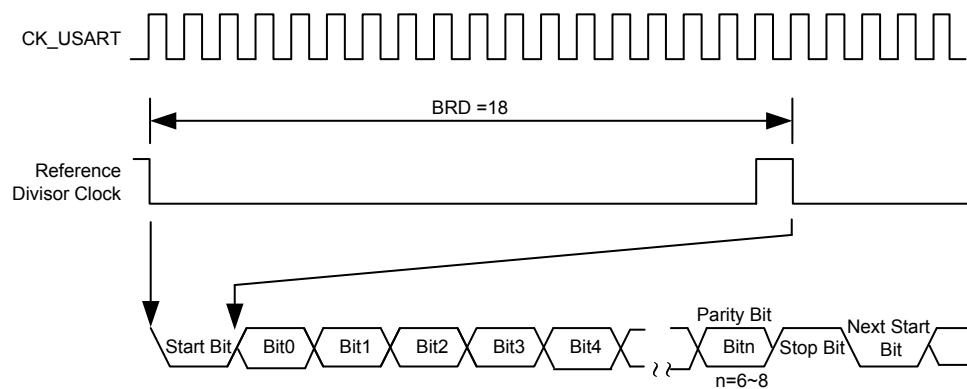


Figure 157. USART Clock CK\_USART and Data Frame Timing

Table 53. Baud Rate Deviation Error Calculation – CK\_USART = 40 MHz

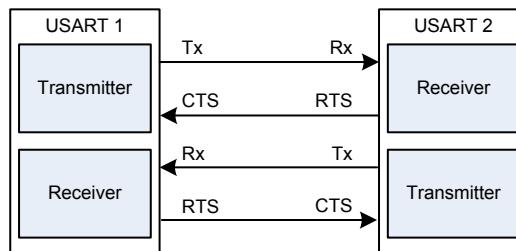
Baud rate		CK_USART = 40 MHz		
No	Kbps	Actual	BRD	Deviation Error rate
1	2.4	2.4	16667	0.00%
2	9.6	9.6	4167	-0.01%
3	19.2	19.2	2083	0.02%
4	57.6	57.6	694	0.06%
5	115.2	115.3	347	0.06%
6	230.4	229.9	174	-0.22%
7	460.8	459.8	87	-0.22%
8	921.6	930.2	43	0.94%
9	2250	2222.2	18	-1.23%
10	3000	3076.9	13	2.56%

**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

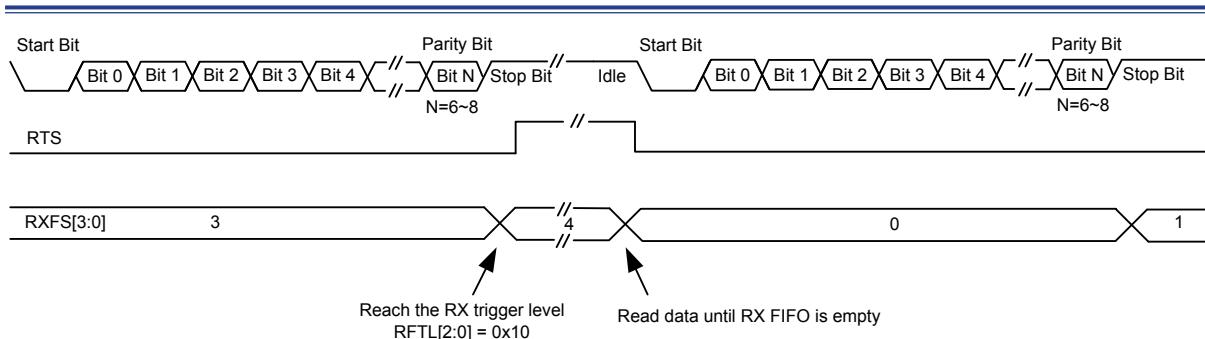
The USART supports the hardware flow control function which is enabled by setting the HFCEN bit in the USRCSR register to 1. It is possible to control the serial data flow between 2 USART devices by using the CTS input and the RTS output. The Figure 157 shows the connection diagram in this mode. The 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 158. 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 158 shows the example of RTS flow control.

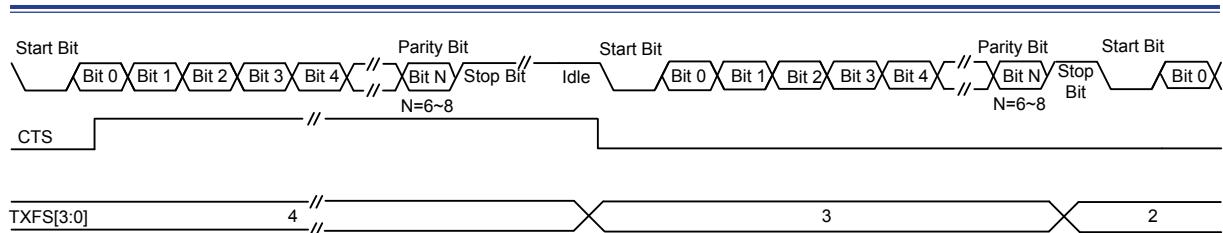


**Figure 159. USART RTS Flow Control**

### 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 159 shows an example of communication with CTS flow control.

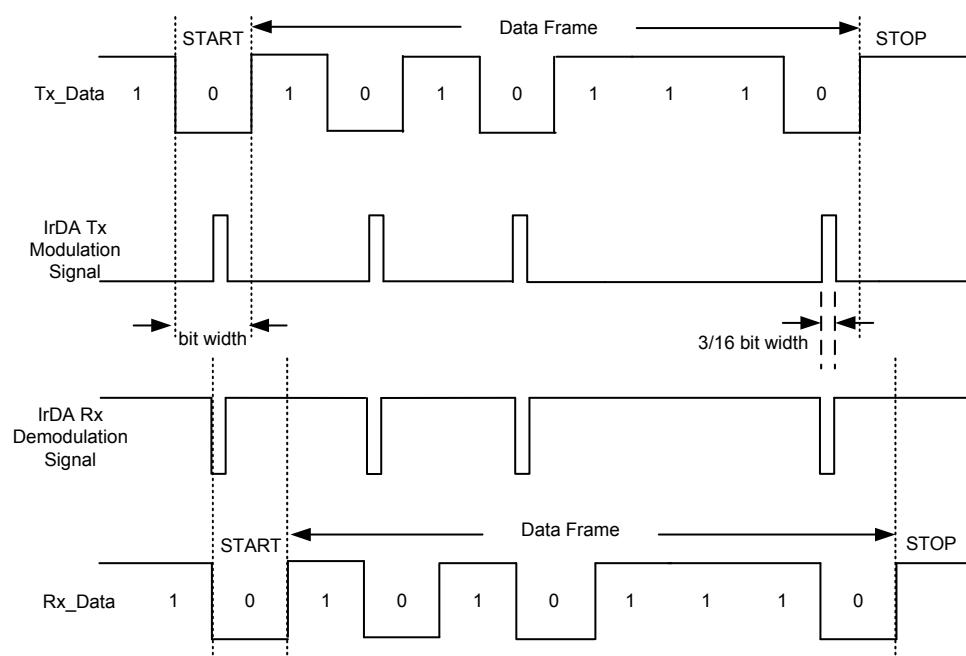


**Figure 160. USART CTS Flow Control**

## 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 ‘high’ level in the IrDA mode.



**Figure 161. 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 in the IrDA Control Register IrDACR is the IrDACR prescaler value.

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.

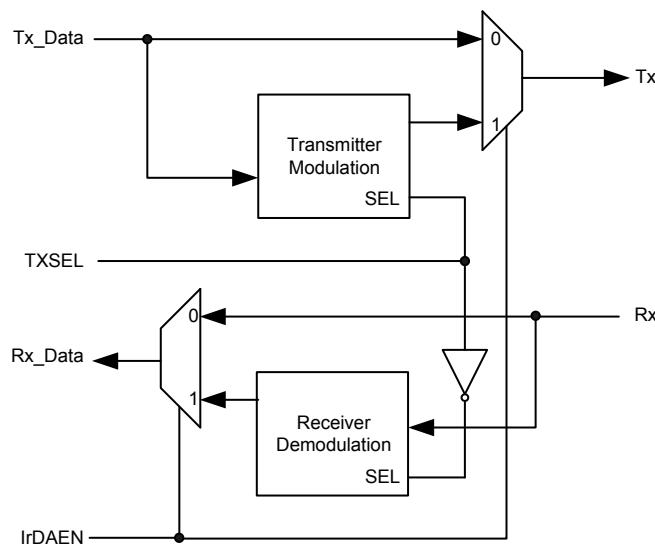


Figure 162. USART I/O and IrDA Block Diagram

## 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.

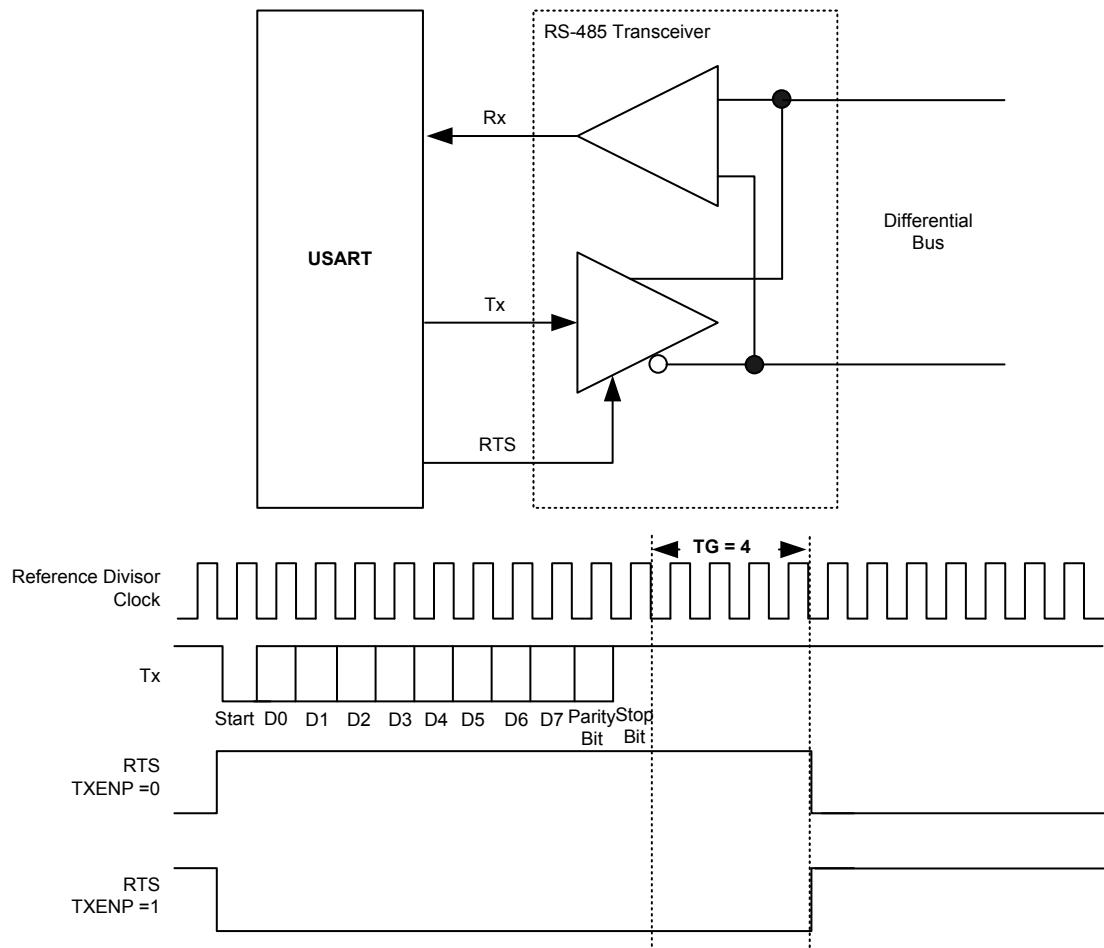


Figure 163. RS485 Interface and Waveform

### 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 USRCSR 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 filed 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 SYNCNR, can be used to determine whether data is captured on the first or the second clock edge. The CPO bit in the SYNCNR can be used to configure the clock polarity in the USART Synchronous Mode idle state. Detailed timing information is shown in Figure 164.

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.

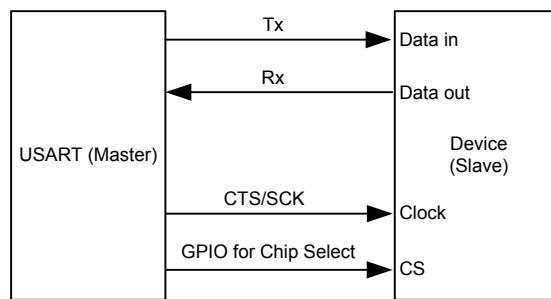
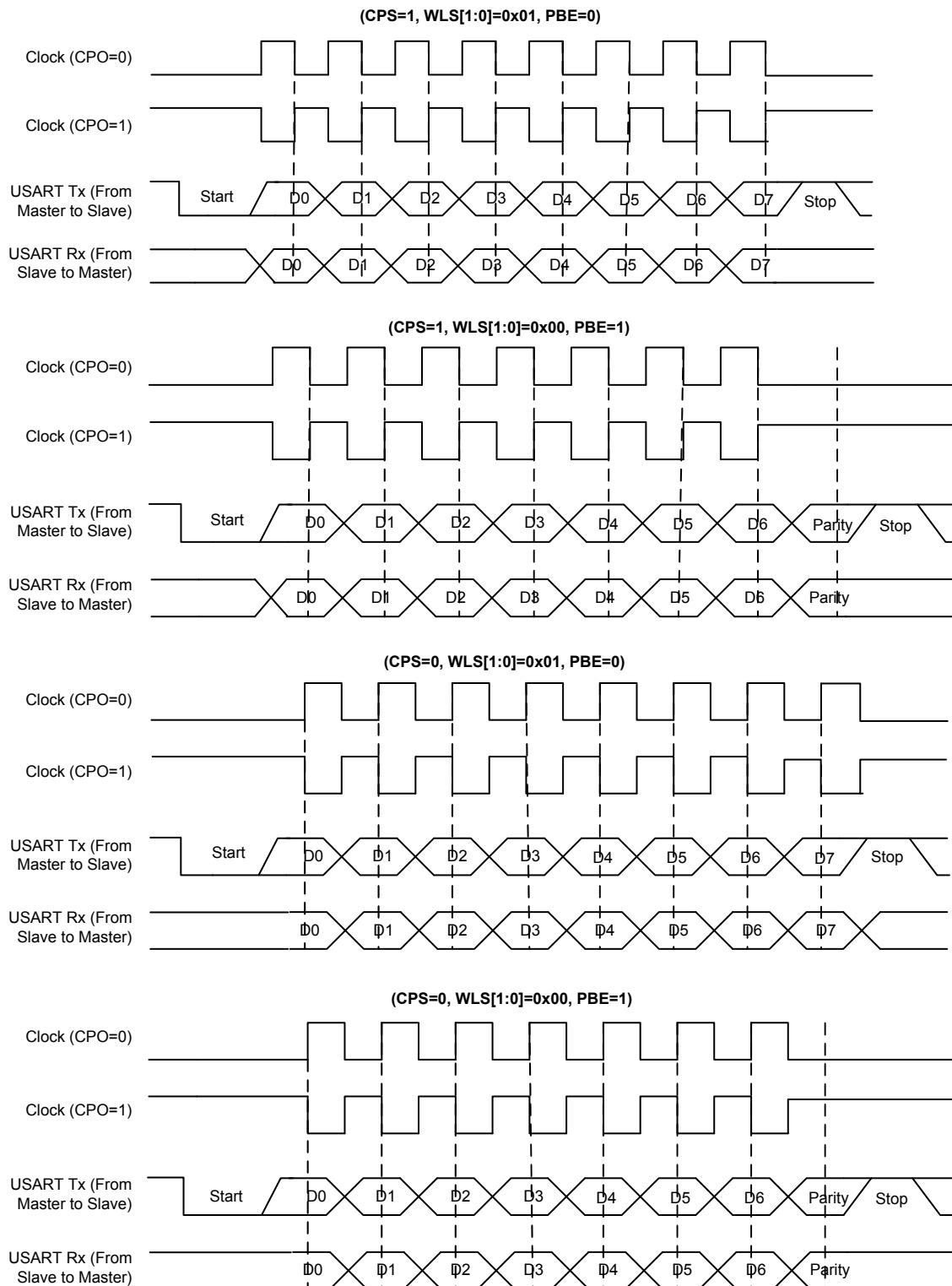


Figure 164. USART Synchronous Transmission Example

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 165. 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 interrupt will be generated when the USART receiver overrun error, parity error, framing error or break events occurs.
- 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.

## 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
USRSTR	0x028	USART Test Register	0x0000_0000

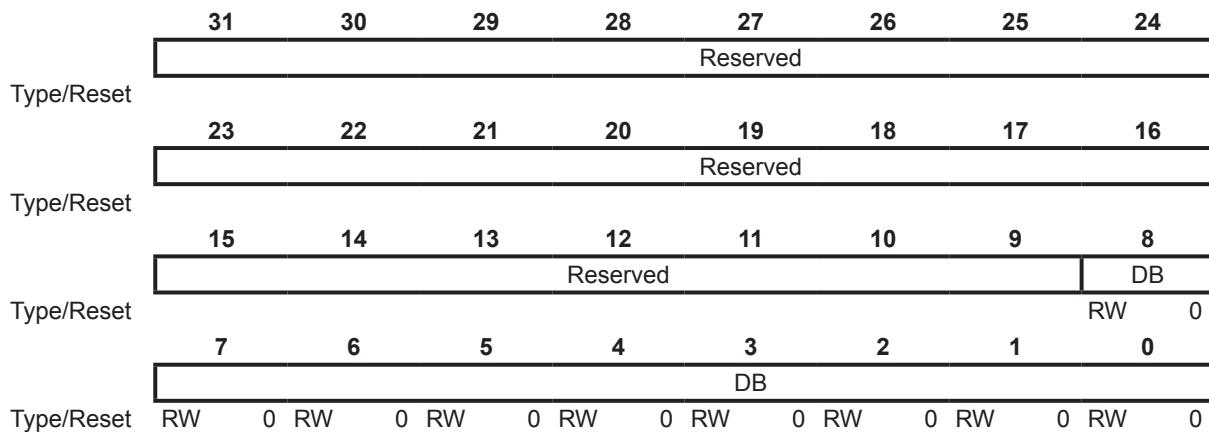
## Register Descriptions

### USART Data Register – USRDR

The register is used to access the USART transmitted and received FIFO data.

Offset: 0x000

Reset value: 0x0000\_0000



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-bits 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 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	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	RW	0	RW	0	RW	0	RW	0	
	RTS	BCB	SPE	EPE	PBE	NSB	WLS		
Type/Reset	RW	0	RW	0	RW	0	RW	0	
	7	6	5	4	3	2	1	0	
Type/Reset	Reserved	URRXEN	URTXEN	HFCEN	TRSM	MODE			
	RW	0	RW	0	RW	0	RW	0	

Bits	Field	Descriptions
[15]	RTS	<p>Request-To-Send Signal</p> <p>0: Drive USART RTS pin to logic 1 1: Drive USART RTS pin to logic 0</p> <p>Note that the RTS bit is used to control the USART RTS pin status when the HFCEN bit is reset.</p> <p>When the HFCEN bit is set, this RTS bit indicates the pin status that is controlled by hardware flow control function.</p>
[14]	BCB	<p>Break Control Bit</p> <p>When this bit is set to 1, the serial data output on the USART TX pin will be forced to the Spacing State (logic 0). This bit acts only on the USART TX output pin and has no effect on the transmitter logic.</p>
[13]	SPE	<p>Stick Parity Enable</p> <p>0: Disable stick parity 1: Stick Parity bit is transmitted</p> <p>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.</p>
[12]	EPE	<p>Even Parity Enable</p> <p>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.</p> <p>This bit is only available when the PBE bit is set to 1.</p>

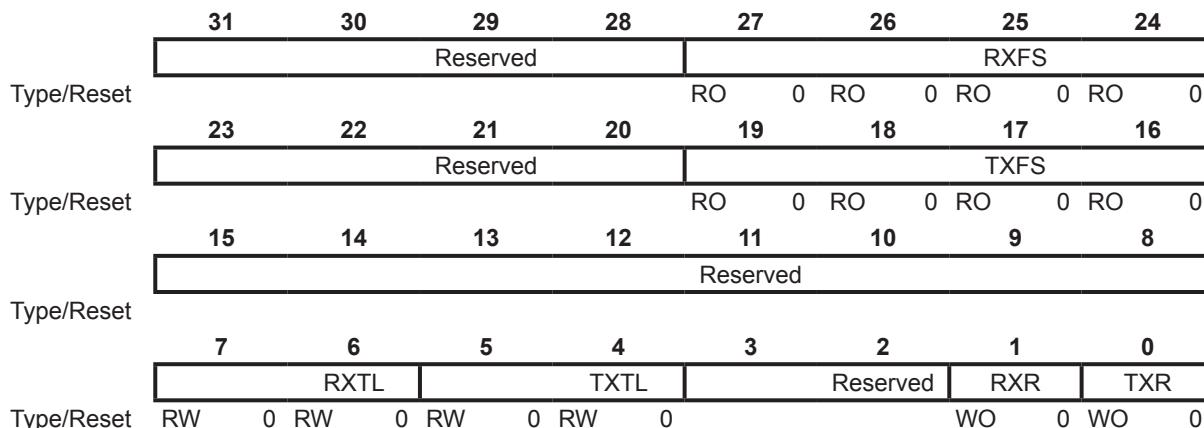
Bits	Field	Descriptions
[11]	PBE	Parity Bit Enable 0: Parity bit is not generated (transmitted data) or checked (receive data) during transfer. 1: Parity bit is generated or checked during transfer. Note: When the WLS field is set to "10" to select the 9-bit data format, writing to the PBE bit has no 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
[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



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> <ul style="list-style-type: none"> <li>0000: RX_FIFO is empty.</li> <li>0001: RX_FIFO contains 1 data</li> <li>...</li> <li>1000: RX_FIFO contains 8 data</li> <li>Others: Reserved</li> </ul>
[19:16]	TXFS	<p>TX_FIFO Status</p> <p>The TXFS field shows the current number of data contained in the TX_FIFO.</p> <ul style="list-style-type: none"> <li>0000: TX_FIFO is empty</li> <li>0001: TX_FIFO contains 1 data</li> <li>...</li> <li>1000: TX_FIFO contains 8 data</li> <li>Others: Reserved</li> </ul>
[7:6]	RXTL	<p>RX_FIFO Threshold Level Setting</p> <ul style="list-style-type: none"> <li>00: 1 byte</li> <li>01: 2 bytes</li> <li>10: 4 bytes</li> <li>11: 6 bytes</li> </ul> <p>The RXTL field defines the RX_FIFO trigger level.</p>
[5:4]	TXTL	<p>TX_FIFO Threshold Level Setting</p> <ul style="list-style-type: none"> <li>00: 0 byte</li> <li>01: 2 bytes</li> <li>10: 4 bytes</li> <li>11: 6 bytes</li> </ul> <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>

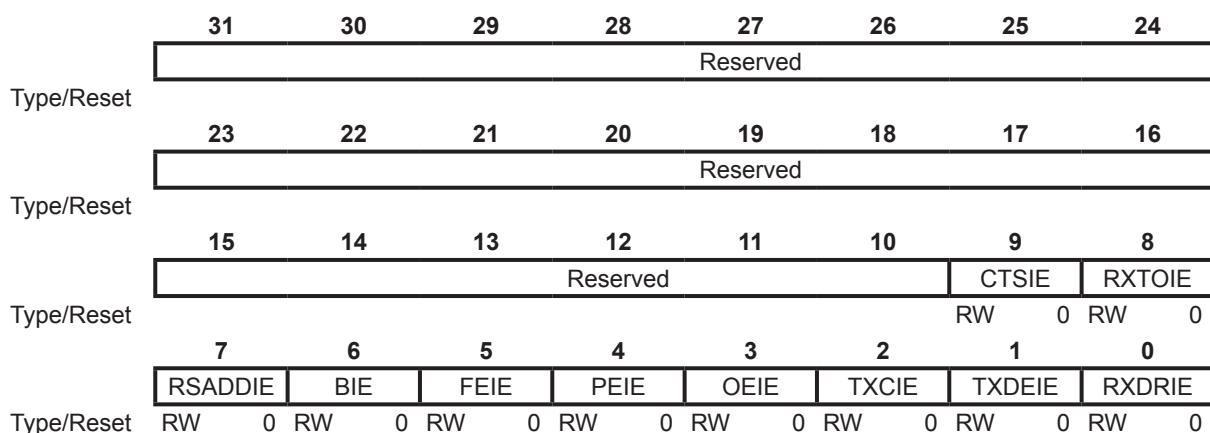
Bits	Field	Descriptions
[0]	TXR	TX_FIFO Reset Setting this bit will generate a reset pulse to reset TX_FIFO which will empty the TX_FIFO. i.e., the TX pointer will be reset to 0, after a reset signal. This bit returns to 0 automatically after the reset pulse is generated.

## 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.

<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>
[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 USRSIFR 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



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 the 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 clear when the new data is written into the USRDR register and the TX_FIFO level is higher than threshold setting.

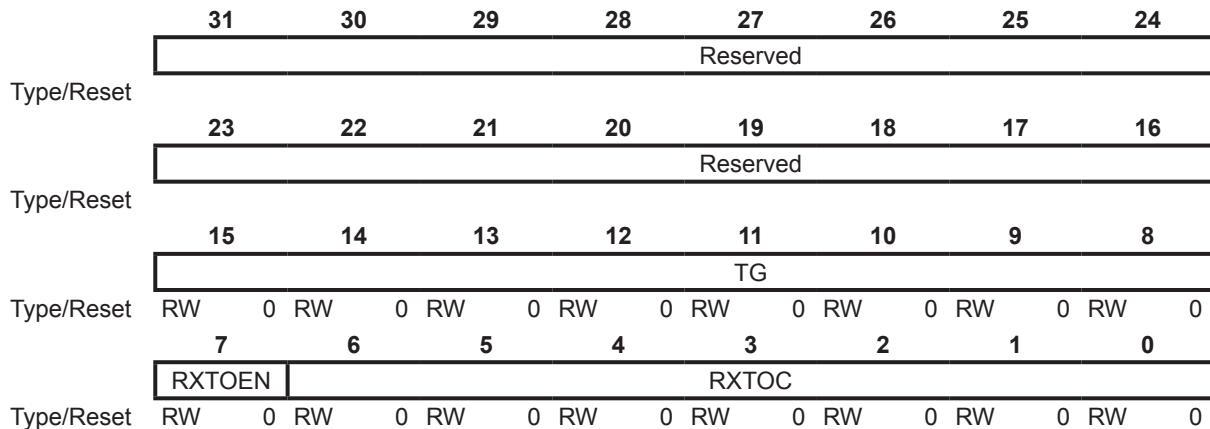
Bits	Field	Descriptions
[6]	RXTOF	Receive FIFO Time-Out Flag 0: RX_FIFO Time-Out does not occur 1: RX_FIFO Time-Out occurs Writing 1 to this bit clears the flag.
[5]	RXDR	Receive FIFO Ready Flag 0: RX_FIFO level is less than threshold 1: RX_FIFO level is higher than threshold The RXDR bit will be set 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 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



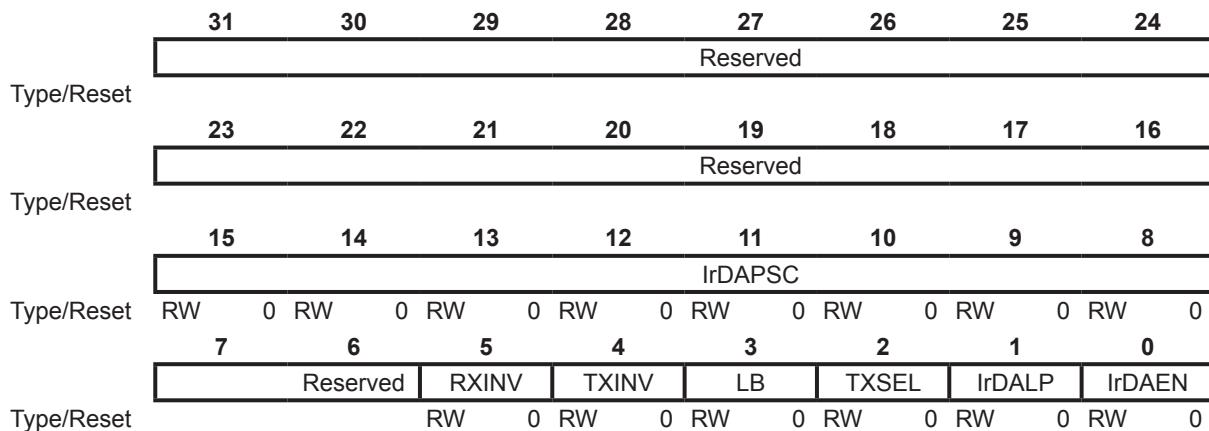
Bits	Field	Descriptions
[15:8]	TG	Transmitter Time Guard The transmitter time guard counter is driven by the baud rate clock. When the TX_FIFO transmits data, the counter will be reset and then starts to count. Only when the counter content is equal to the TG value, are further word transmission transactions 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 – IrDPCR

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

Offset: 0x018

Reset value: 0x0000\_0000



Bits	Field	Descriptions
[15:8]	IrDAPSC	<p>IrDA Prescaler value</p> <p>This field contains the 8-bit debounce prescaler value.</p> <p>The debounce count-down counter is driven by the USART clock, named as CK_USART. The counting period is specified by the IrDAPSC field. The IrDAPSC field must be set to a value equal to or greater than 0x01 for normal debounce counter operation. If the pulse width is less than the duration specified by the IrDAPSC field, the pulse will be considered as glitch noise and discarded.</p> <ul style="list-style-type: none"> <li>00000000: Reserved – can not be used.</li> <li>00000001: CK_USART clock divided by 1</li> <li>00000010: CK_USART clock divided by 2</li> <li>00000011: CK_USART clock divided by 3</li> <li>...</li> </ul>
[5]	RXINV	<p>RX Signal Inverse Control</p> <ul style="list-style-type: none"> <li>0: No inversion</li> <li>1: RX input signal is inverted</li> </ul>
[4]	TXINV	<p>TX Signal Inverse Control</p> <ul style="list-style-type: none"> <li>0: No inversion</li> <li>1: TX output signal is inverted</li> </ul>
[3]	LB	<p>IrDA Loop Back Mode</p> <ul style="list-style-type: none"> <li>0: Disable IrDA loop back mode</li> <li>1: Enable IrDA loop back mode for self testing.</li> </ul>
[2]	TXSEL	<p>Transmit Select</p> <ul style="list-style-type: none"> <li>0: Enable IrDA receiver</li> <li>1: Enable IrDA transmitter</li> </ul>
[1]	IrDALP	<p>IrDA Low Power Mode</p> <p>Select the IrDA operation mode.</p> <ul style="list-style-type: none"> <li>0: Normal mode</li> <li>1: IrDA low power mode</li> </ul>

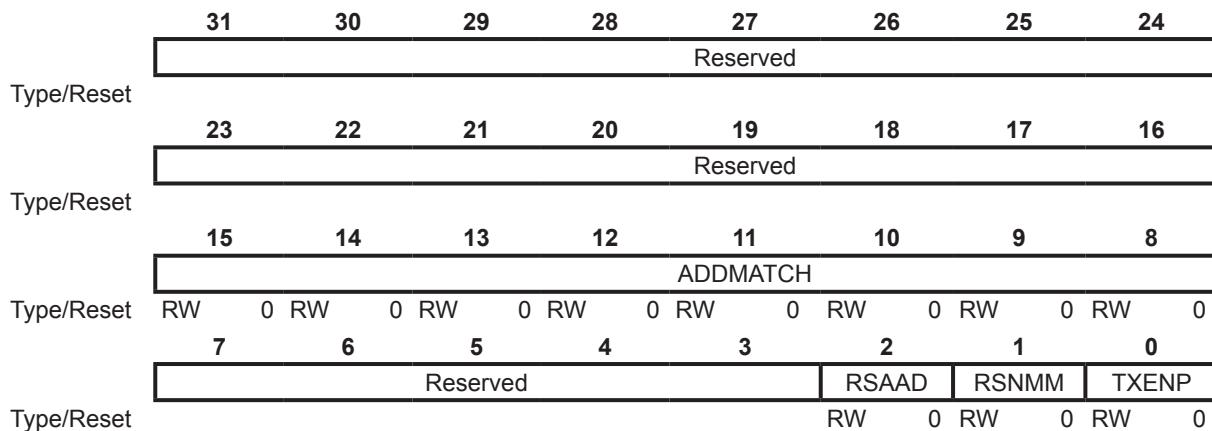
Bits	Field	Descriptions
[0]	IrDAEN	IrDA Enable control 0: Disable IrDA mode 1: Enable IrDA mode

## USART RS485 Control Register – RS485CR

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

Offset: 0x01C

Reset value: 0x0000\_0000



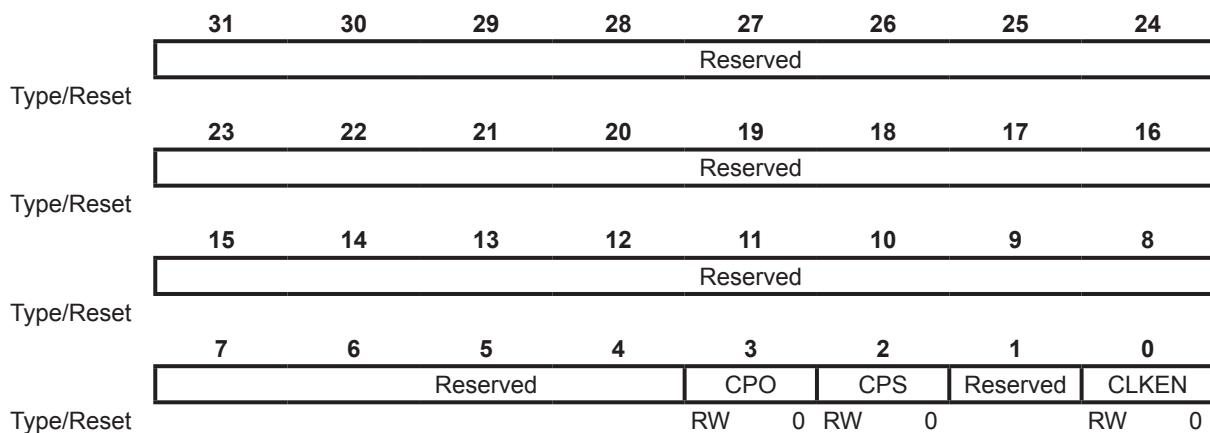
Bits	Field	Descriptions
[15:8]	ADDMATCH	RS485 Auto Address Match value The field contains the address match value for the RS485 auto address detection operation mode.
[2]	RSAAD	RS485 Auto Address Detection Operation Mode Control 0: Disable 1: Enable
[1]	RSNMM	RS485 Normal Multi-drop Operation Mode Control 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	RW	0	RW	0	RW	0	RW	0	
	7	6	5	4	3	2	1	0	
Type/Reset	RW	0	RW	0	RW	1	RW	0	
	BRD								
	RW	0	RW	0	RW	1	RW	0	

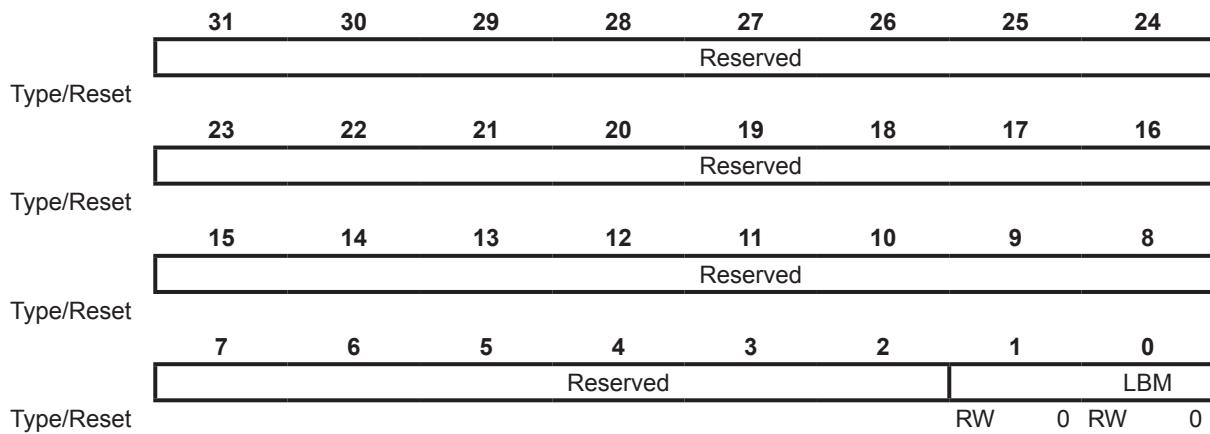
Bits	Field	Descriptions
[15:0]	BRD	<p>Baud Rate Divider</p> <p>The 16 bits define the USART clock divider ratio.</p> <p>Baud Rate = CK_USART / BRD</p> <p>Where the CK_USART clock is the clock connected to the USART module.</p> <p>BRD = 16 ~ 65535 for asynchronous mode</p> <p>BRD = 8 ~ 65535 for synchronous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

## USART Test Register – USRTSTR

This register controls the USART debug mode.

Offset: 0x028

Reset value: 0x0000\_0000



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

## 22 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.

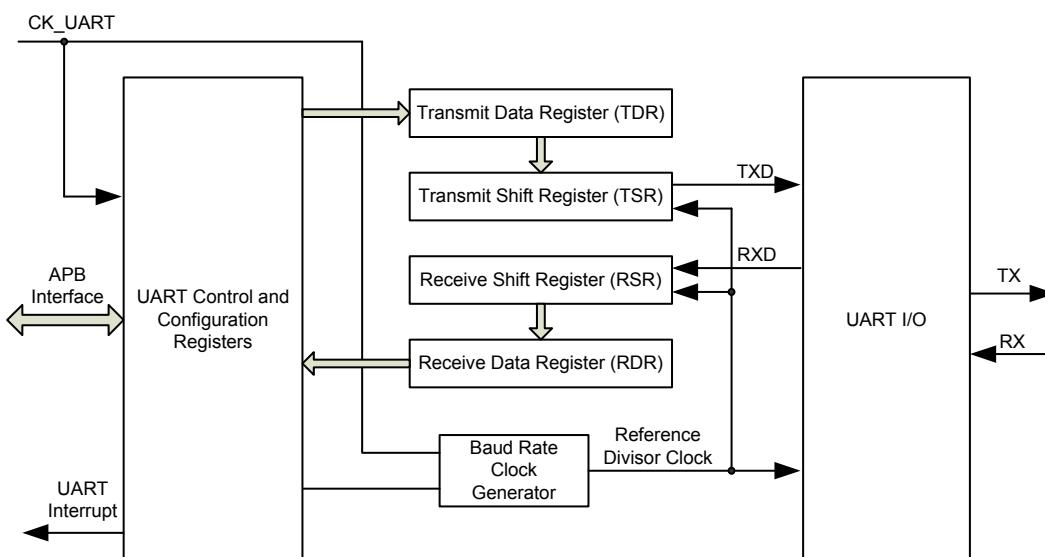


Figure 166. UART Block Diagram

## 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

## 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.

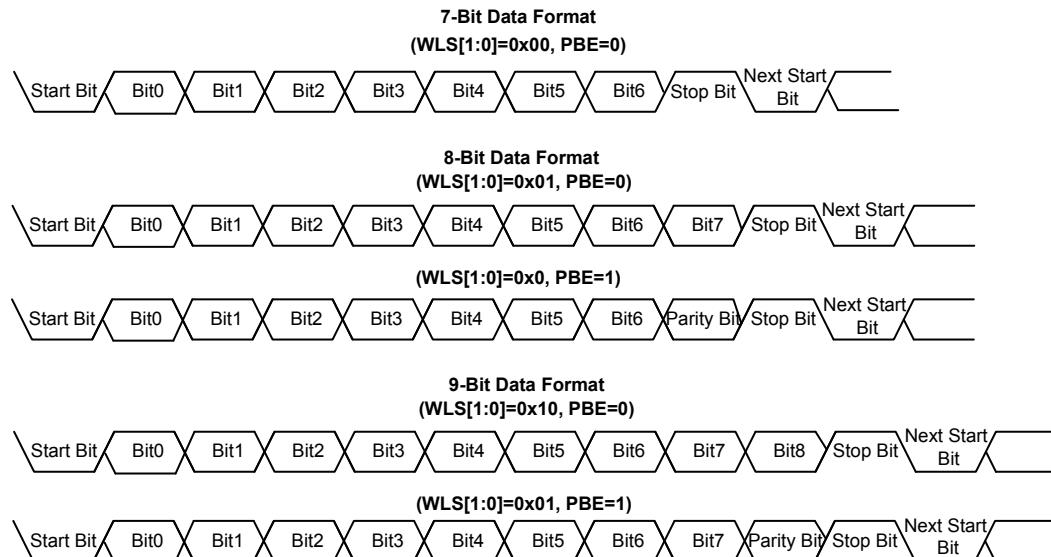


Figure 167. UART Serial Data Format

## 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.

$$\text{Baud Rate Clock} = \text{CK}_\text{UART} / \text{BRD}$$

Where CK\_UART clock is the APB clock connected to the UART while the BRD range is from 16 to 65535.

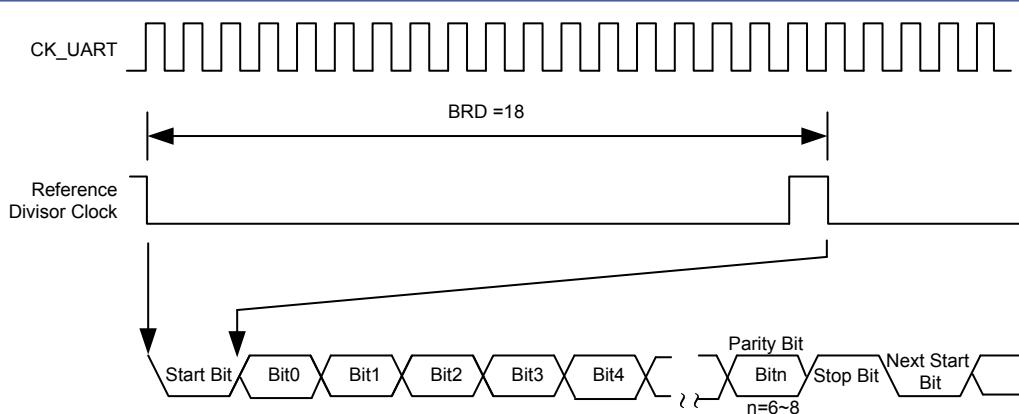


Figure 168. UART Clock CK\_UART and Data Frame Timing

Table 56. Baud Rate Deviation Error Calculation – CK\_UART = 40 MHz

No	Kbps	CK_UART = 40 MHz		
		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 events 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.

### 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

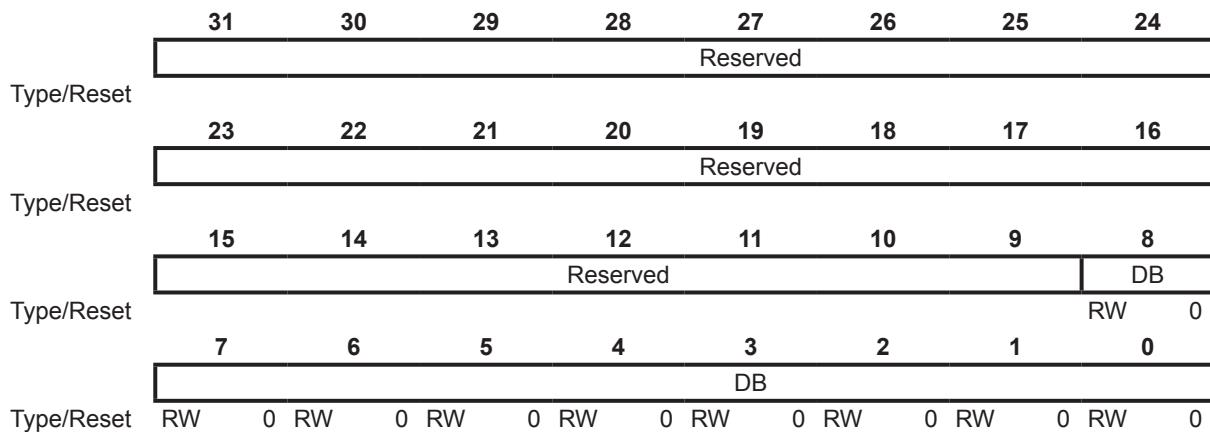
## Register Descriptions

### UART Data Register – URDR

The register is used to access the UART transmitted and received data.

Offset: 0x000

Reset value: 0x0000\_0000



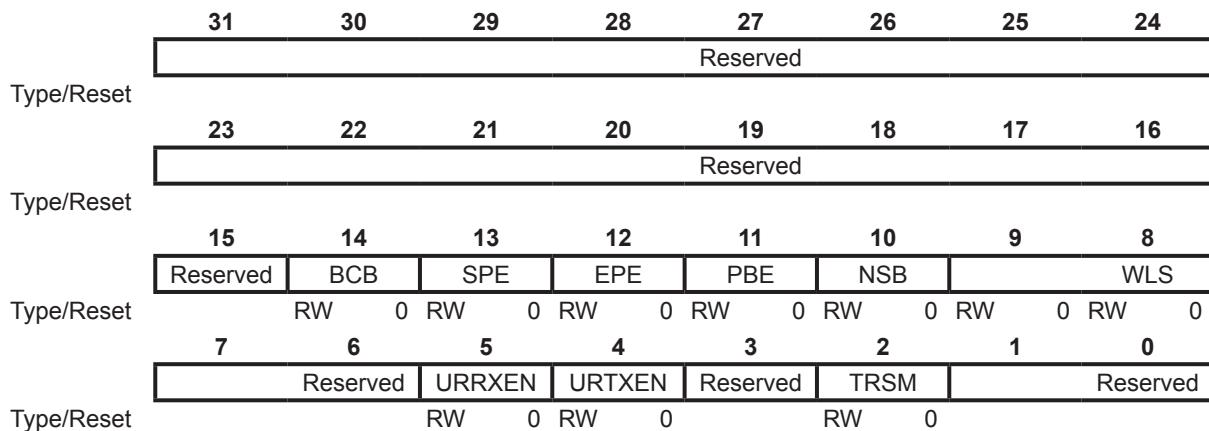
Bits	Field	Descriptions
[8:0]	DB	<p>By reading this register, the UART will return a 7, 8 and 9-bit received data. The DB field bit 8 is valid for 9-bit mode only and is fixed at 0 for the 8-bit mode. For the 7-bits mode, the DB[6:0] contains the available bits.</p> <p>By writing to this register, the UART will send out 7, 8 or 9-bit transmitted data. The DB field bit 8 is valid for the 9-bit mode only and will be ignored for the 8-bit mode. For the 7-bit mode, the DB[6:0] contains the available bits.</p>

## UART Control Register – URCR

The register specifies the serial parameters such as data length, parity, and stop bit for the UART.

Offset: 0x004

Reset value: 0x0000\_0000



Bits	Field	Descriptions
[14]	BCB	Break Control Bit When this bit is set to 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.

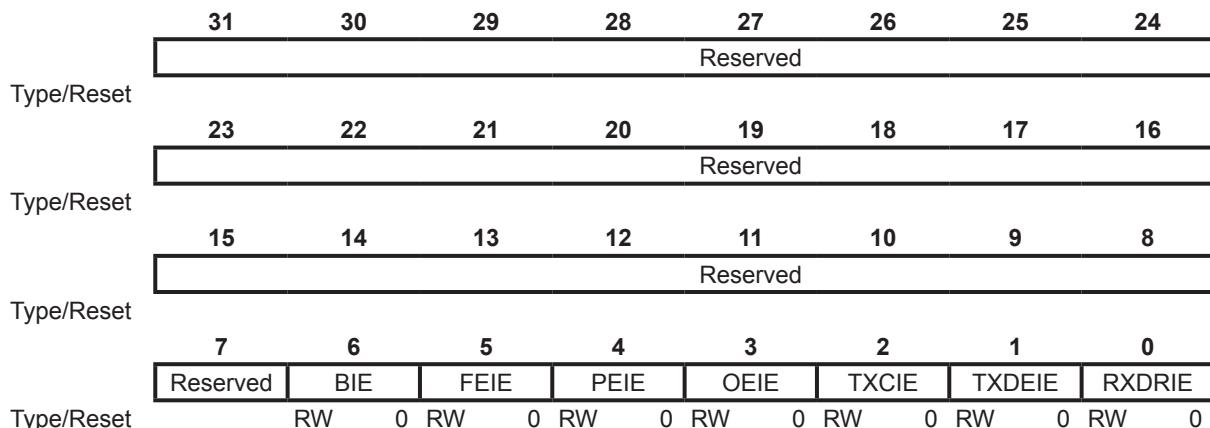
<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>
[9:8]	WLS	Word Length Select 00: 7 bits 01: 8 bits 10: 9 bits 11: Reserved
[5]	URRXEN	UART RX Enable 0: disable 1: enable
[4]	URTXEN	UART TX Enable 0: disable 1: enable
[2]	TRSM	Transfer Mode Selection This bit is used to select the data transfer protocol. 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 transmission complete interrupt is enabled and the TXC bit is set in the URSIFR register.

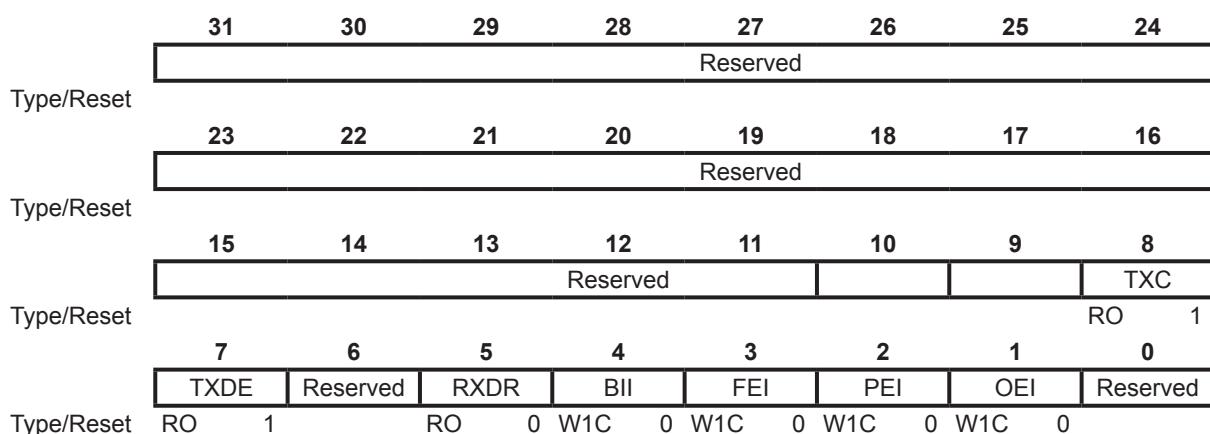
Bits	Field	Descriptions
[1]	TXDEIE	<p>Transmit Data Register Empty Interrupt Enable</p> <p>0: Disable interrupt. 1: Enable interrupt</p> <p>An interrupt is generated when the transmit data register empty interrupt is enabled and the TXDE bit is set in the URSIFR register.</p>
[0]	RXDRIE	<p>Receive Data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

## UART Status & Interrupt Flag Register – URSIFR

This register contains the corresponding UART status.

Offset: 0x010

Reset value: 0x0000\_0180



Bits	Field	Descriptions
[8]	TXC	<p>Transmit Complete</p> <p>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.</p> <p>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.</p>
[7]	TXDE	<p>Transmit Data Register Empty</p> <p>0: Transmit data register is not empty. 1: Transmit data register is empty.</p> <p>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.</p>

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 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 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

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

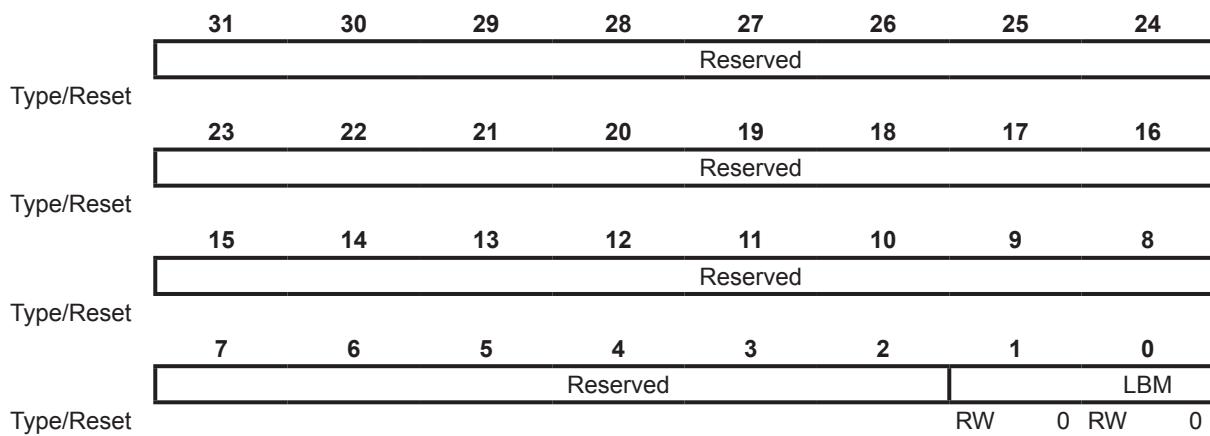
Bits	Field	Descriptions
[15:0]	BRD	<p>Baud Rate Divider</p> <p>The 16 bits define the UART clock divider ratio.</p> <p>Baud Rate = CK_UART / BRD</p> <p>Where the CK_UART clock is the clock connected to the UART module.</p> <p>BRD = 16 ~ 65535 for UART mode</p>

## 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

## 23 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.

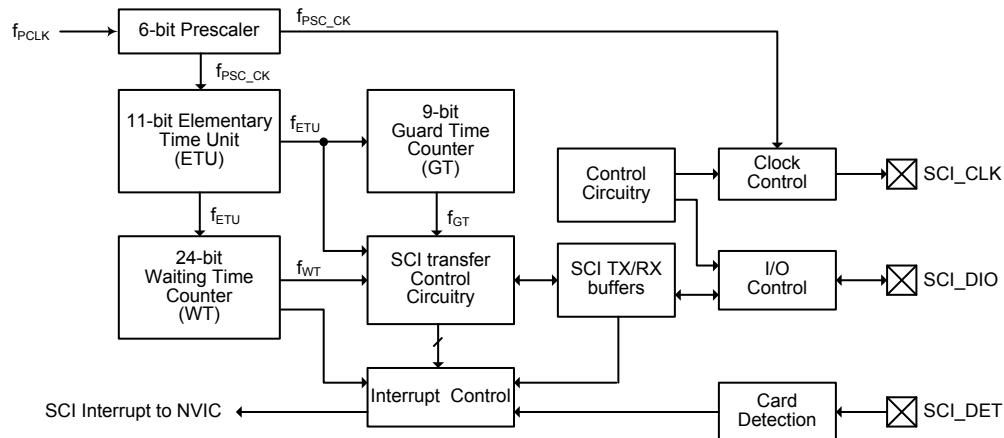


Figure 169. SCI Block Diagram

## 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

## 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. When an insertion or a removal of the external Smart Card is detected, an interrupt signal which is sent to the microcontroller will be generated 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.

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

where:

- etu is the nominal duration of the data bit on the signal SCI\_DIO provided to the card by the interface
- Di is the bit-rate adjustment factor
- Fi 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<sub>i</sub> is an encoded decimal value based on a 4-bit field, named DI, as represented in the accompanying table.

**Table 59. DI Field Based Di Encoded Decimal Values**

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

F<sub>i</sub> 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<sub>i</sub> Encoded Decimal Values**

FI field	0000	0001	0010	0011	0100	0101	0110	1001	1010	1011	1100	1101
Fi (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 Fi and Di values can be obtained by looking up the preceding two tables. After the Fi and Di values are obtained, the value which should be written into the ETUR register can be calculated by Fi/Di. The following table shows the possible ETU values obtained by the F<sub>i</sub>/D<sub>i</sub> ratio.

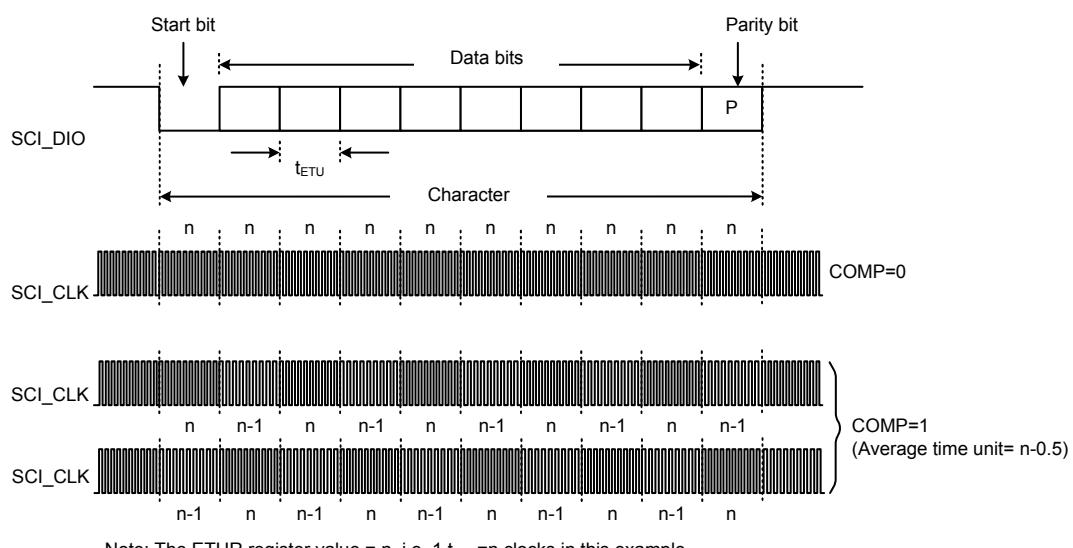
**Table 61. Possible ETU Values Obtained with the Fi/Di Ratio**

Fi Di	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.

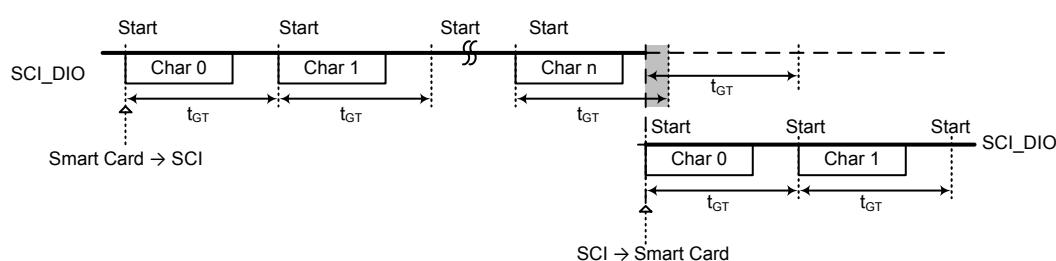
If the calculation result is close to the value of (n-0.5), the compensation mode should be enabled by setting the compensation enable control bit, COMP, in the ETUR register to 1 for successful data transfer. When the result is close to the value of (n-0.5) and the compensation mode is enabled, the 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 170. 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 171. Guard Time Duration**

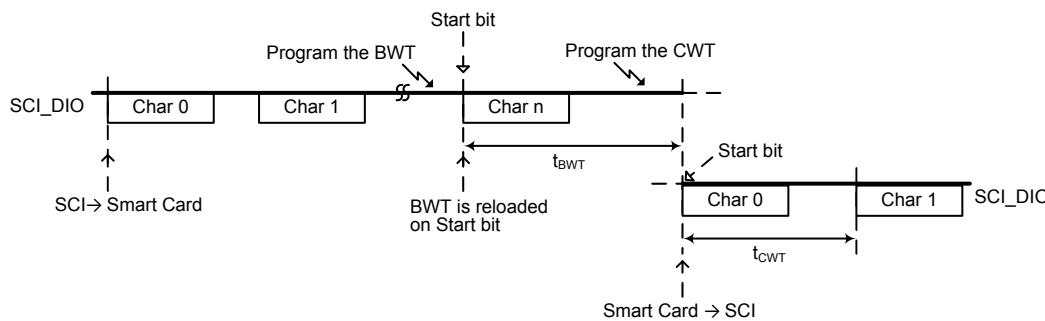
## Waiting Time Counter

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 172. 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.

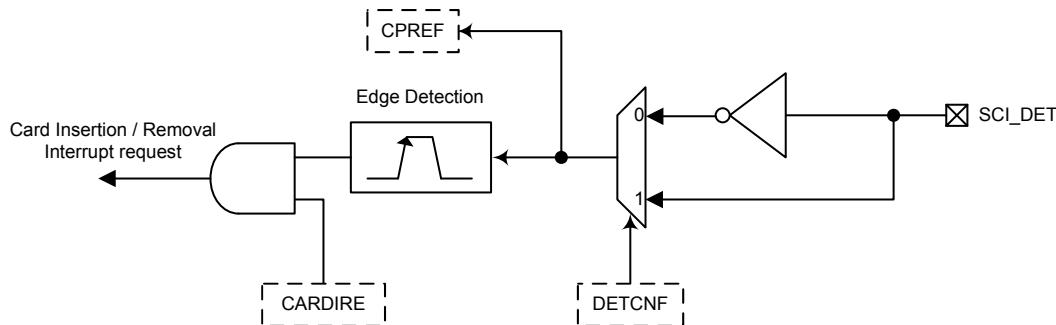
The SCI serial data pin, named SCI\_DIO, can be controlled by the SCI hardware circuitry or the software control bits depending upon whether the SCI is operated in the SCI Transfer Mode or in the Manual Mode. The mode selection is determined by the SCIM bit in the CR register. The SCI\_DIO pin status 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 173. 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

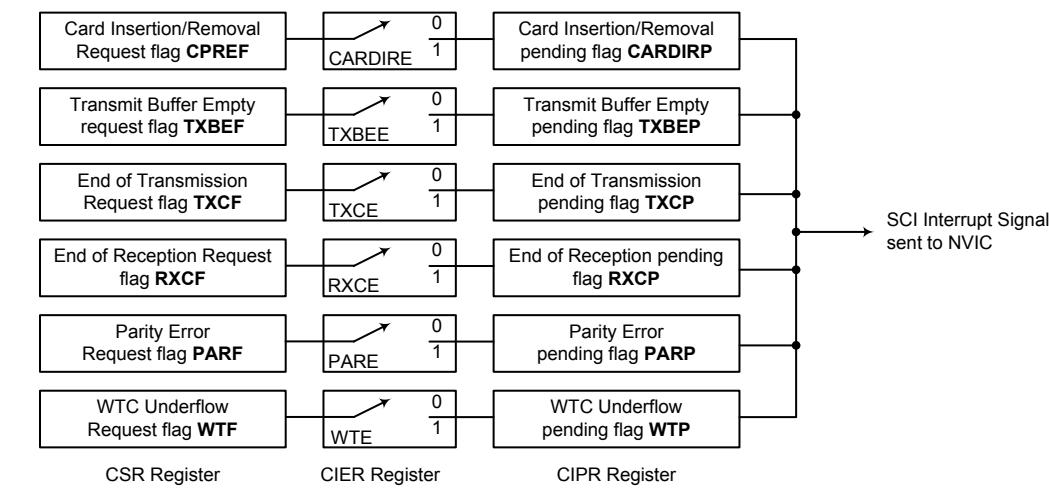
If the direction convention used by the Smart Card is the same as the convention used by the SCI, the SCI will generate a reception interrupt if the reception interrupt is enabled without a parity error flag. Otherwise, the SCI will generate a reception interrupt and the parity error flag will be asserted. 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 174. SCI Interrupt Structure**

## 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

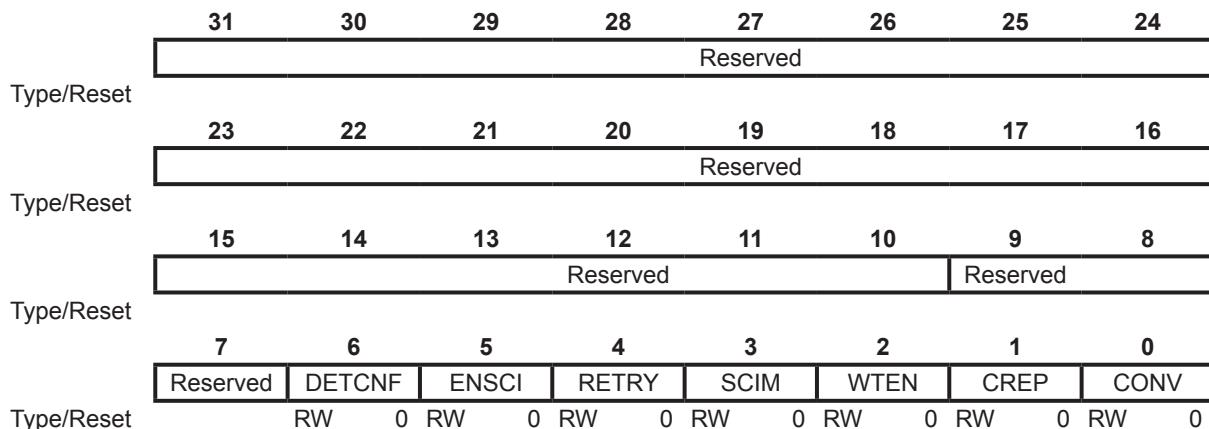
## Register Descriptions

### SCI Control Register – CR

This register contains the SCI control bits.

Offset: 0x000

Reset value: 0x0000\_0000



Bits	Field	Descriptions															
[6]	DETCNF	<p>Card switch type selection</p> <p>0: Switch is normally opened if no card is present 1: Switch is normally closed if no card is present</p> <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> <p>This bit is set and cleared by the application program to configure the card detector switch type.</p>	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															
[5]	ENSCI	<p>SCI finite state machine enable bit</p> <p>0: SCI FSM is disabled and forced to its initial state 1: SCI FSM is enabled</p>															
[4]	RETRY	<p>Character transfer repetition time selection for a parity error condition</p> <p>0: Data transfer 5 times when parity error occurs 1: Data transfer 4 times when parity error occurs</p> <p>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.</p>															
[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>															

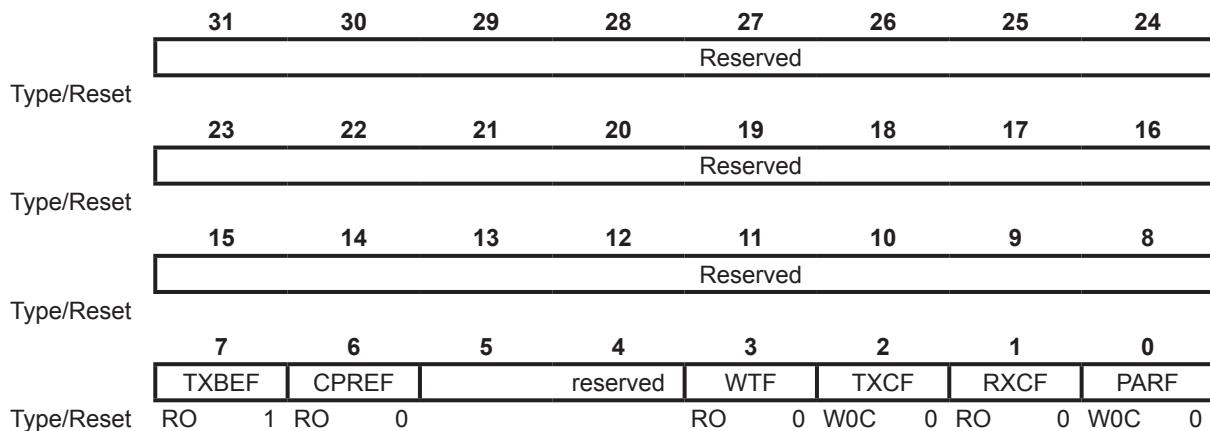
Bits	Field	Descriptions
[2]	WTEN	<p>Waiting Time Counter enable control</p> <p>0: Waiting Time Counter stops counting 1: Waiting Time Counter starts counting</p> <p>The WTEN bit is set and cleared by the application program. When the WTEN bit is cleared to 0, a write access to the WTR register will load 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 4<sup>th</sup> or 5<sup>th</sup> 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 4<sup>th</sup> or 5<sup>th</sup> 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



Bits	Field	Descriptions
[7]	TXBEF	Transmit Buffer Empty Request Flag 0: Transmit buffer is not empty 1: Transmit buffer is empty This bit is used to indicate if the transmit buffer is empty and is set or cleared by hardware automatically.
[6]	CPREF	Card Presence Request Flag 0: No card is present 1: A card is present 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.
[3]	WTF	Waiting Time Counter Underflow Request Flag 0: No Waiting Time Counter underflow 1: The Waiting Time Counter underflows This bit is set and cleared by the application program and indicates if the Waiting Time Counter underflows.
[2]	TXCF	Character Transmission Request Flag. 0: No character transmitted 1: A character has been transmitted This bit is set by hardware and cleared by writing a “0” into it.
[1]	RXCF	Character Received Request Flag. 0: No character received 1: A character has been received 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. 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.

Bits	Field	Descriptions
[0]	PARF	<p>Parity Error Request Flag.</p> <p>0: No parity error occurs</p> <p>1: Parity error has occurred</p> <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	CLKSEL	Reserved			CDIO	CCLK	Reserved		
	RW	0			RW	1	RW	0	

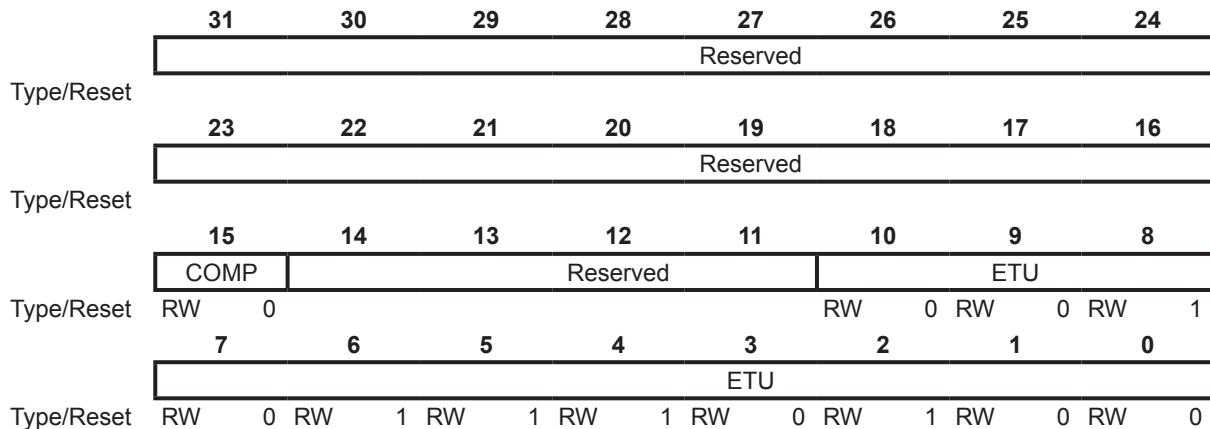
Bits	Field	Descriptions
[7]	CLKSEL	Card Clock Selection 0: The CCLK bit content is present on the external SCI_CLK pin 1: The clock output on the external SCI_CLK pin is sourced from the $f_{PSC\_CK}$ clock This bit is used to select the external SCI_CLK pin clock source. It is set and cleared by the 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



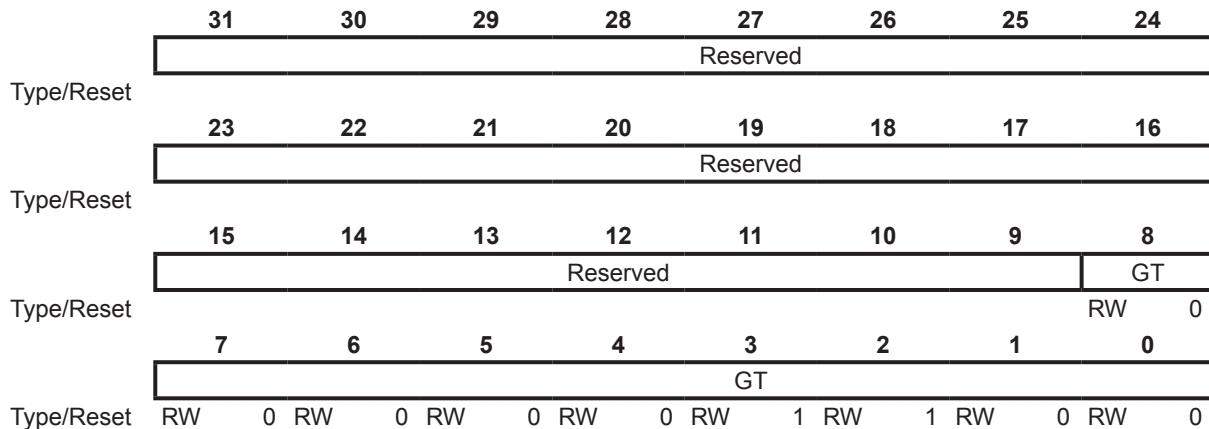
Bits	Field	Descriptions
[15]	COMP	<p>Elementary Time Unit Compensation mode enable control</p> <p>0: Compensation mode is disabled.                  1: Compensation mode is enabled.</p> <p>This bit is set and cleared by application program and used to control the ETU compensation function. For more details regarding the compensation function consult the Elementary Time Unit section.</p>
[10:0]	ETU	<p>ETU value for a character data bit</p> <p>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.</p>

## 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



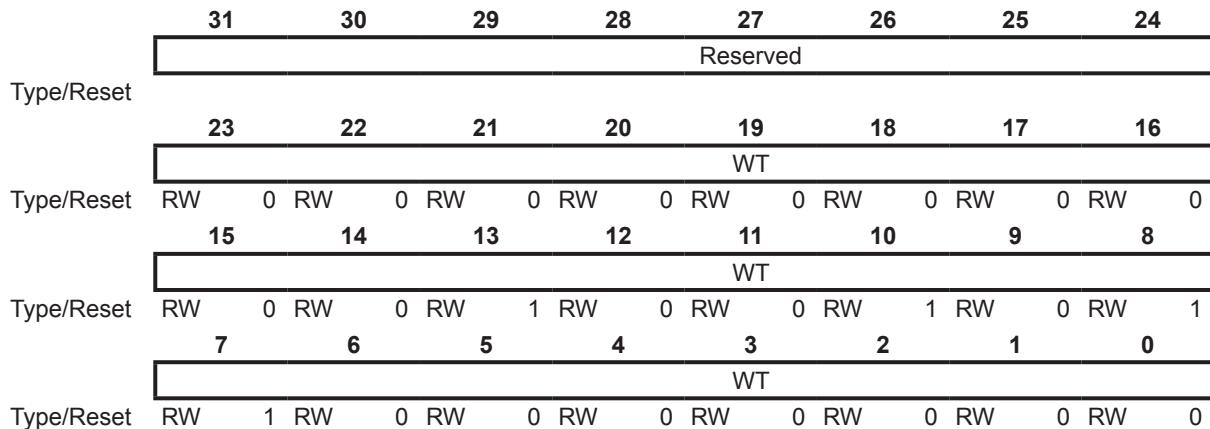
Bits	Field	Descriptions
[8:0]	GT	<p>Character Guard Time value</p> <p>This field is configured by the application program to modify the guard time duration. The updated GT value will be loaded into the GT counter at the end of the current guard time period. Note that the GT value must be in the range from 0x00C to 0x1FF.</p>

## 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	<p>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.</p>

## 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

	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	TXBEE	CARDIRE	Reserved		WTE	TXCE	RXCE	PARE
	RW	0	RW	0	RW	0	RW	0

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.

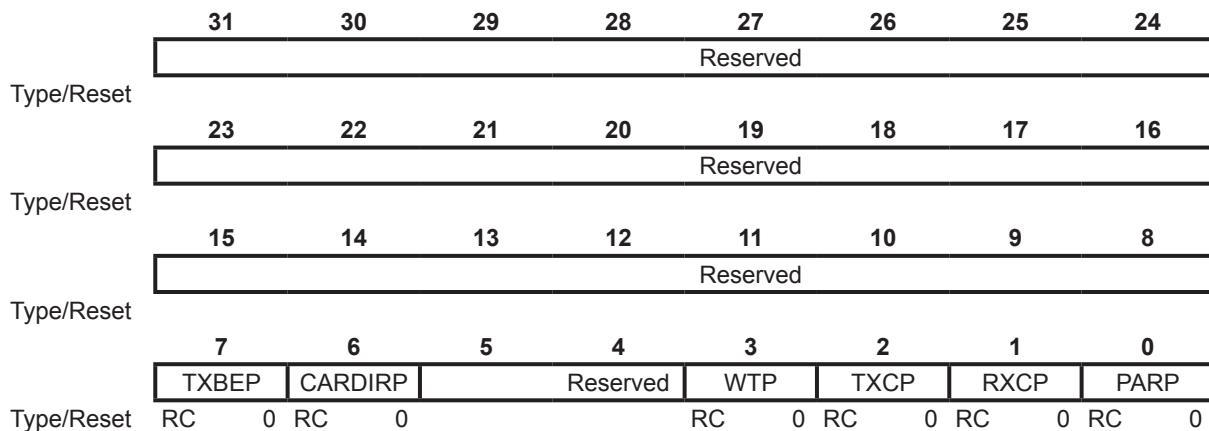
Bits	Field	Descriptions
[1]	RXCE	Character Reception Completion interrupt enable control 0: Disabled 1: Enabled This bit is set and cleared by the application program and is used to control the Character Reception Completion interrupt. If this bit is set to 1, the Character Reception Completion interrupt 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	Character Transmission Completion interrupt pending flag 0: No interrupt pending 1: Interrupt pending 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.
[1]	RXCP	Character Reception Completion interrupt pending flag 0: No interrupt pending 1: Interrupt pending 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.
[0]	PARP	Parity Error interrupt pending flag 0: No interrupt pending 1: Interrupt pending 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.

## SCI Transmit Buffer – TXB

This register is used to store the SCI data to be transmitted.

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	RW	0	RW	0	RW	0	RW	0
					TB			

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

	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	RW	0	RW	0	RW	0
					RB			

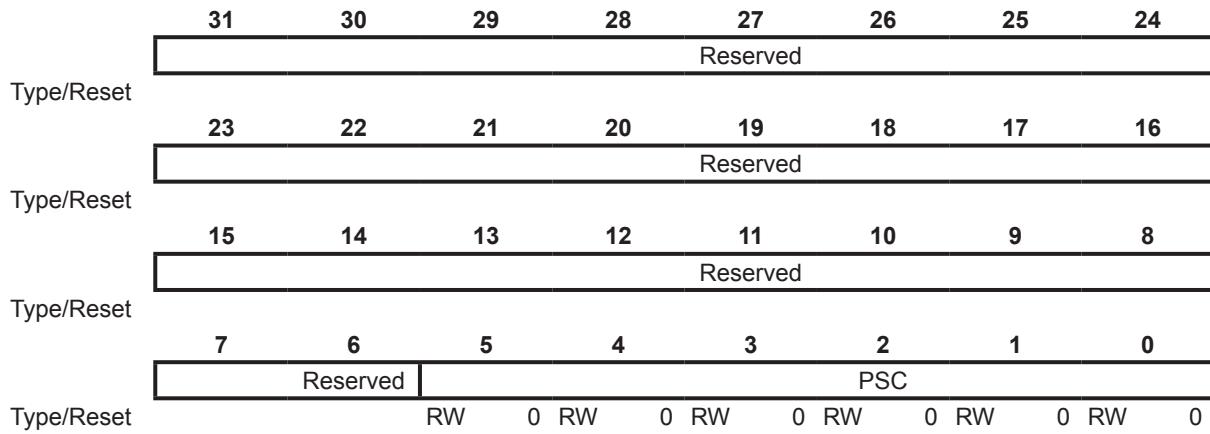
Bits	Field	Descriptions
[7:0]	RB	SCI Received data byte

## 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: $f_{PSC\_CK} = \frac{f_{PCLK}}{2 \times PSC}$

# 24 USB Device Controller (USB)

## Introduction

The USB device controller is compliant with the USB 2.0 full-speed specification. There is one control endpoint known as Endpoint 0 and seven configurable endpoints (EP1~EP7). A 1024-byte EP\_SRAM is used for the endpoint buffers. Each endpoint buffer size is programmable by corresponding registers, which provides maximum flexibility for various applications. The integrated USB full-speed transceiver helps to minimize overall system complexity and cost. The USB also contains the suspend and resume features to meet low-power consumption requirement. The accompanying figure shows the USB block diagram.

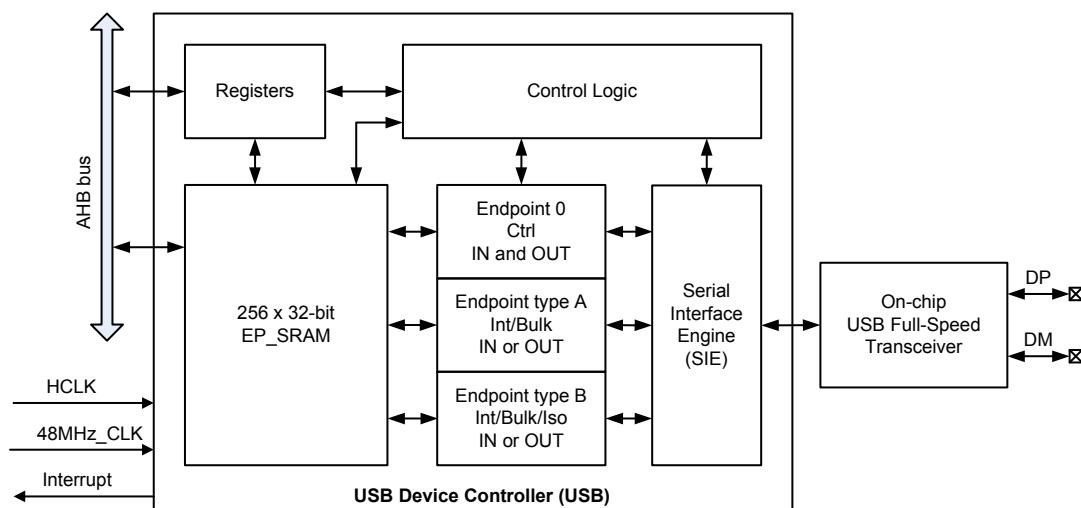


Figure 175. USB Block Diagram

## 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

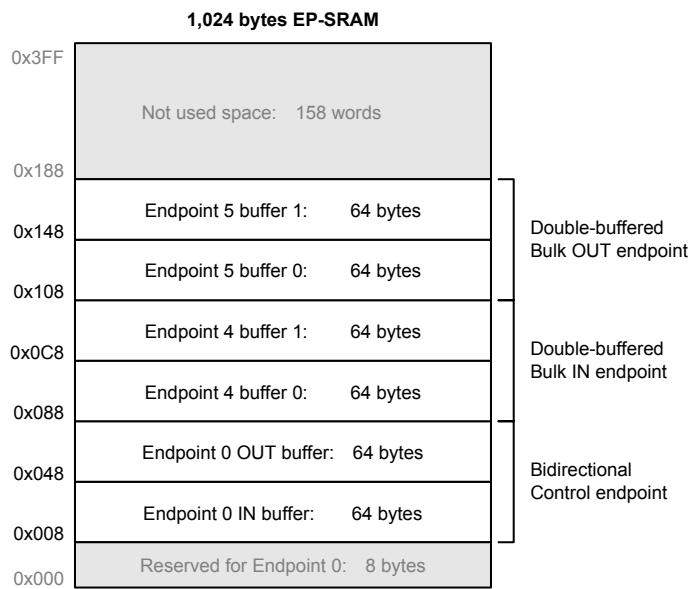
### EP\_SRAM

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. 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.



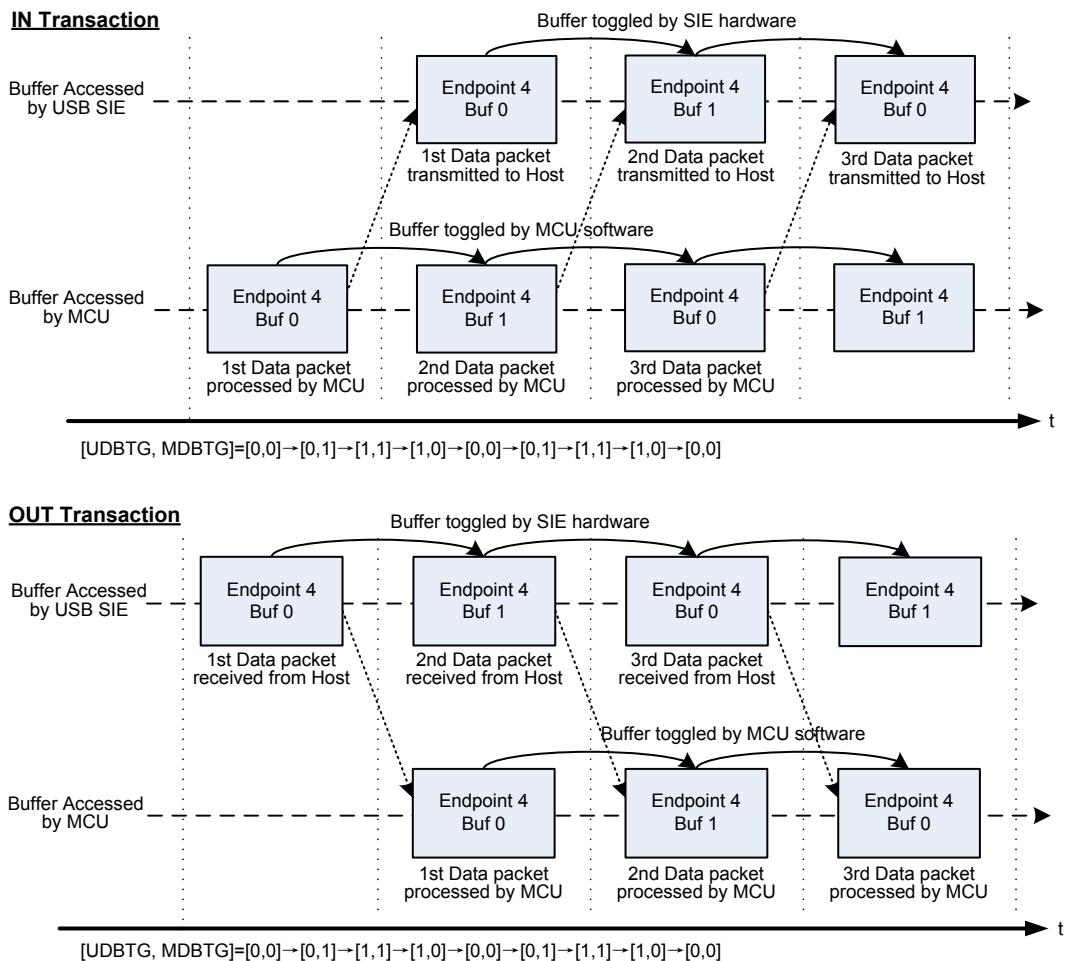
**Figure 176. Endpoint Buffer Allocation Example**

## 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 USBEPEnCSR register where the denotation n ranges from 4 to 7.



**Figure 177. 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  $\mu$ 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
<b>USB Base Address = 0x400A_8000</b>			
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
USBEP0CFG	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
USBEP1CFG	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
USBEP2CFG	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

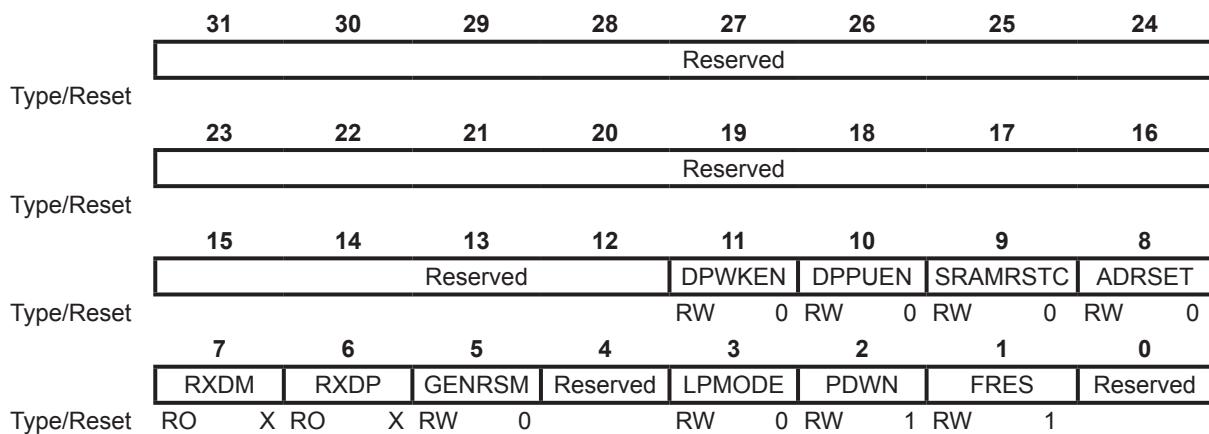
## Register Descriptions

### USB Control and Status Register – USBCSR

This register specifies the USB control bits and USB data line status.

Offset: 0x000

Reset value: 0x0000\_00X6



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	EP_SRAM reset condition 0: Reset EP_SRAM when (DP, DM) = (0,0) 1: User can access EP_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.

<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>
[3]	LPMODE	<b>Low-power Mode Control</b> 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. 0: Exit the Low-power mode 1: Enter the Low-power mode
[2]	PDWN	<b>Power Down Mode Control</b> Setting this bit will power down the full-speed USB PHY transceiver. This will disconnect the USB PHY transceiver from the USB bus. 0: Exit the Power-Down mode 1: Enter the Power-Down mode
[1]	FRES	<b>Force USB Reset Control</b> 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. 0: Release USB reset 1: Force USB reset

**Table 66. Resume Event Detection**

<b>[RXDP, RXDM] Status</b>	<b>Wakeup event</b>	<b>Required resume software action</b>
00	Root reset	None
10	None (noise on bus)	Go back to suspend mode
01	Root resume	None
11	Not allowed (noise on bus)	Go back to suspend mode

## USB Interrupt Enable Register – USBIER

This register specifies the USB interrupt enable control.

Offset: 0x004

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	EP7IE RW	EP6IE 0	EP5IE RW	EP4IE 0	EP3IE RW	EP2IE 0	EP1IE RW	EP0IE 0	
	7	6	5	4	3	2	1	0	
Type/Reset	Reserved 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
Type/Reset	Reserved							
Type/Reset	Reserved							
Type/Reset	Reserved							
Type/Reset	15	14	13	12	11	10	9	8
	EP7IF	EP6IF	EP5IF	EP4IF	EP3IF	EP2IF	EP1IF	EP0IF
	WC	0	WC	0	WC	0	WC	0
Type/Reset	7	6	5	4	3	2	1	0
Type/Reset	Reserved		ESOFIF	SUSPIF	RSMIF	URSTIF	SOFIF	Reserved
	RW		0	WC	0	WC	0	WC

Bits	Field	Descriptions
[15:8]	EPnIF	Endpoint n Interrupt Flag (n = 0 ~ 7) This bit is set by the hardware to indicate the generation of relevant endpoint 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 USBEPIISR 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.

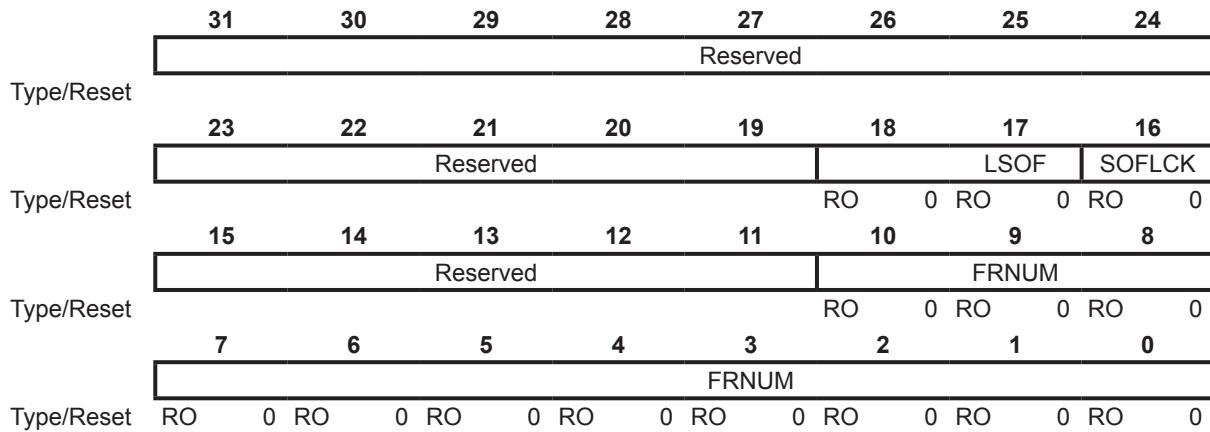
Bits	Field	Descriptions
[2]	URSTIF	<p>USB Reset Interrupt Flag</p> <p>This bit is set by the hardware when the USB reset has been detected. When a USB reset occurs, the internal protocol state machine will be reset and an USB reset interrupt will be generated if the URSTIE bit in the USBIER register 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 (USBEPnCFG) 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



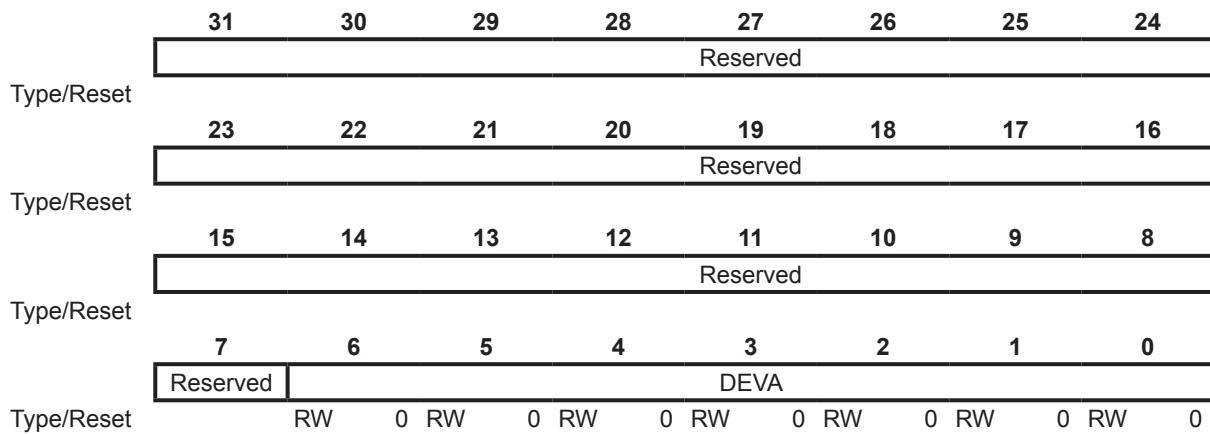
Bits	Field	Descriptions
[18:17]	LSOF	Lost Start-of-Frame number These bits are written and incremented by 1 by the hardware each time the ESOFIF bit is set. It is used to count the number of lost SOF packets. When a SOF packet has been received, these bits are cleared.
[16]	SOFLCK	Start-of-Frame Lock Flag This bit is set by the hardware when SOF packets have been received before the frame timer times out. Once this flag is set to 1, the frame number which is sent from the USB host will be loaded into the Frame Number field 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



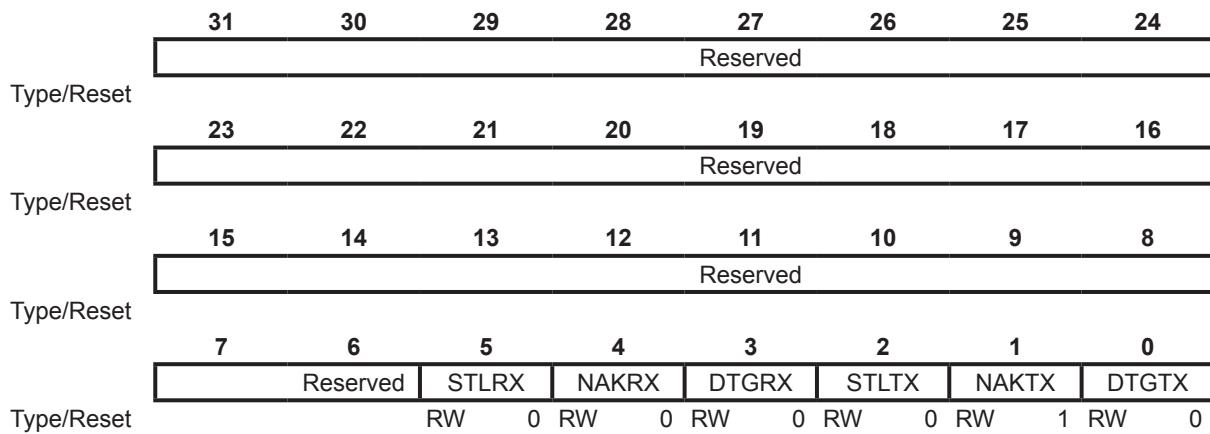
Bits	Field	Descriptions
[6:0]	DEVA	<p>Device Address</p> <p>This field is used to specify the USB device address. This field is cleared when a USB reset event occurs.</p>

## 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	STALL Status for reception (OUT) transfer  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.  This bit can be read and written and can only be toggled by writing 1.
[4]	NAKRX	NAK Status for reception (OUT) transfer  This bit is toggled from 0 to 1 by the hardware circuitry, which will result in a NAK signal in the handshake phase of an 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.  This bit can be read and written and can only be toggled by writing 1.
[3]	DTGRX	Data Toggle Status for reception (OUT) transfer  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.  This bit can be read and written and can only be toggled by writing 1.
[2]	STLTX	STALL Status for transmission (IN) transfer  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.  This bit can be read and written and can only be toggled by writing 1.

<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>
[1]	NAKTX	<p>NAK Status for transmission (IN) transfer This bit is toggled from 0 to 1 by the hardware circuitry, which will result in a NAK signal in the handshake phase of an IN transaction after an ACK signal has been received. It indicates that the USB device is temporarily unable to transmit data to the USB host. Therefore, there will be more time for the application software to properly prepare the data to be transmitted. This bit can be read and written and can only be toggled by writing 1.</p>
[0]	DTGTX	<p>Data Toggle Status for transmission (IN) transfer This bit contains the required value of the data toggle bit (0=DATA0, 1=DATA1) for the next data packet to be transmitted. When the current data packet is transmitted by the USB device and the corresponding ACK signal sent by the USB host is received, the hardware circuitry will toggle this bit and the next data packet will be transmitted. For Endpoint 0, the hardware circuitry will toggle this bit to 1 after the SETUP token is received as Endpoint 0 is addressed. This bit can also be toggled by the software to initialize its value for certain applications. This bit can be read and written and can only be toggled by writing 1.</p>

## USB Endpoint 0 Interrupt Enable Register – USBEP0IER

This register specifies the Endpoint 0 interrupt control bits.

Offset: 0x018

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
Type/Reset	Reserved							
Type/Reset	Reserved							
Type/Reset	15	14	13	12	11	10	9	8
Type/Reset	Reserved					ZLRXIE	SDERIE	SDRXIE
Type/Reset	RW					0	RW	0
Type/Reset	7	6	5	4	3	2	1	0
Type/Reset	UERIE	STLIE	NAKIE	IDTXIE	ITRXIE	ODOVIE	ODRXIE	OTRXIE
Type/Reset	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[11]	ZLRXIE	Zero Length Data Received Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[10]	SDERIE	SETUP Data Error Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[9]	SDRXIE	SETUP Data Received Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[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

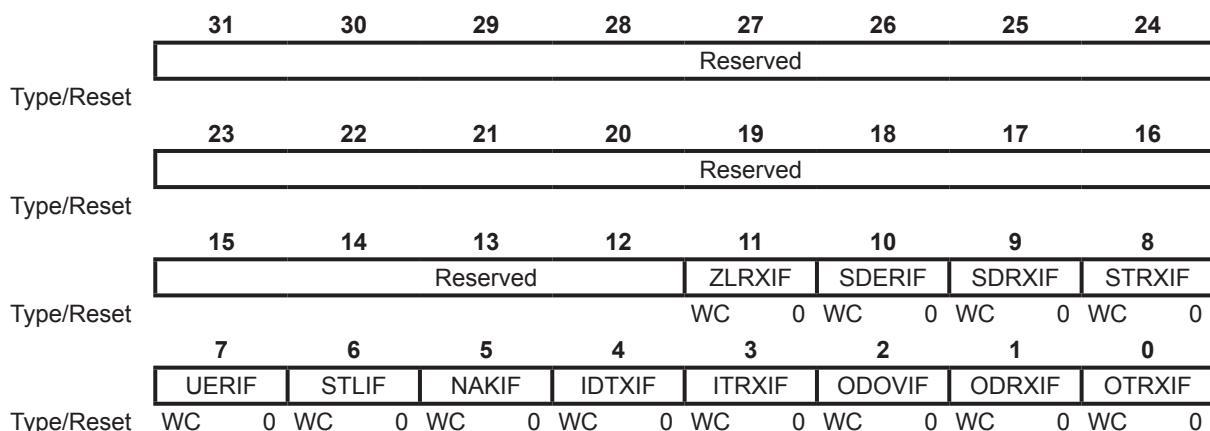
Bits	Field	Descriptions
[1]	ODRXIE	OUT Data Received Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[0]	OTRXIE	OUT Token Received Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt

### USB Endpoint 0 Interrupt Status Register – USBEP0ISR

This register specifies the Endpoint 0 interrupt status.

Offset: 0x01C

Reset value: 0x0000\_0000



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.

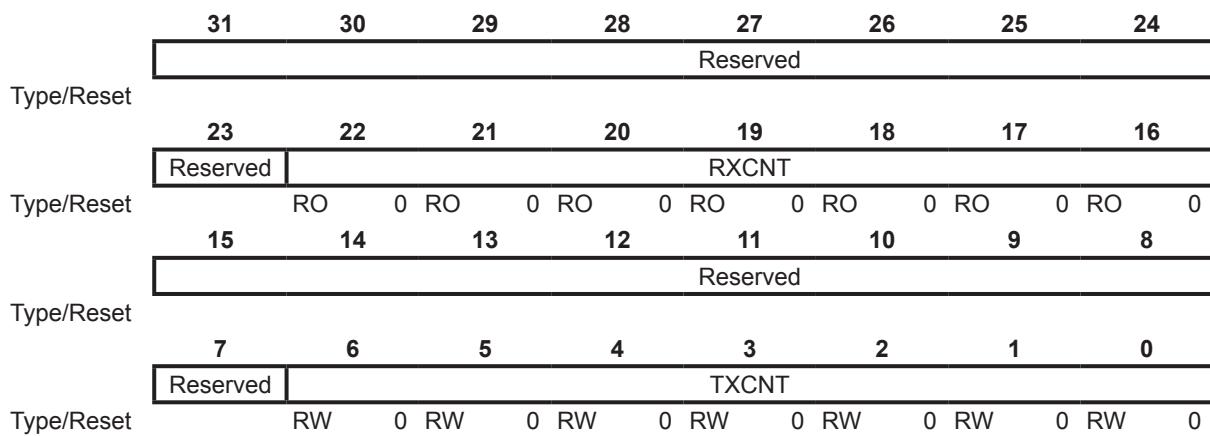
<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>
[6]	STLIF	STALL Transmitted Interrupt Flag This bit is set by the hardware when a STALL signal is sent in response to an IN or OUT transaction. This bit is cleared by hardware when a SETUP Token is received or by writing 1.
[5]	NAKIF	NAK Transmitted Interrupt 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.
[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
Type/Reset	EPEN	Reserved					EPADR	
	RO	1			RO	0	RO	0
Type/Reset	23	22	21	20	19	18	17	16
	Reserved							EPLEN
Type/Reset	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
Type/Reset	EPBUFA							
	RW	0	RW	0	RW	0	RW	0

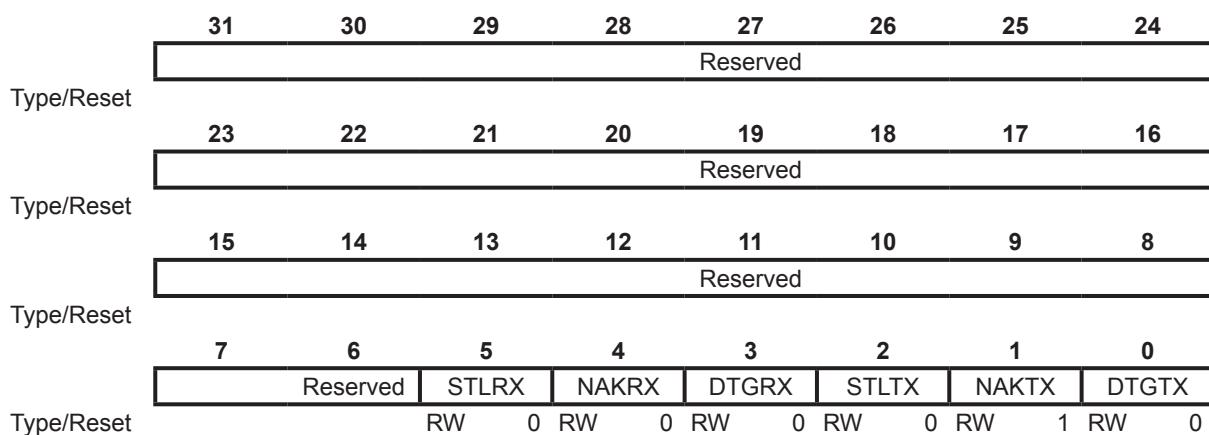
Bits	Field	Descriptions
[31]	EPEN	Endpoint Enable Control This bit is always set to 1 by the hardware circuitry to always enable Endpoint 0.
[27:24]	EPADR	Endpoint Address This field is always set to 0 by the hardware circuitry.
[16:10]	EPLEN	Endpoint Buffer Length This field is used to specify the control transfer packet size which can be 8, 16, 32 or 64 bytes as defined in the USB full-speed standard specification.
[9:0]	EPBUFA	Endpoint Buffer Address This field is used to specify the start address of the Endpoint 0 buffer allocated in the EP_SRAM. It starts from 0x008 and should be aligned to 4-byte boundary. Start address of EP0 IN buffer = EPBUFA Start address of EP0 OUT buffer = EPBUFA + EPLEN

## USB Endpoint 1 ~ 3 Control and Status Register – USBE<sup>n</sup>CSR, 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	STALL bit for reception transfers 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.
[4]	NAKRX	NAK bit for reception transfers 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. This bit can be read and written and can be only toggled by writing 1.
[3]	DTGRX	Data Toggle bit for reception transfers 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. 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.
[2]	STLTX	STALL bit for transmission transfers 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.
[1]	NAKTX	NAK bit for transmission transfers 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. This bit can be read and written and can be only toggled by writing 1.

Bits	Field	Descriptions
[0]	DTGTX	<p>Data Toggle bit for transmission transfers.</p> <p>This bit contains the required value of the data toggle bit (0=DATA0, 1=DATA1) for the next data packet to be transmitted. When the current data packet is transmitted by the USB device and the corresponding ACK signal sent from the USB host is received, the hardware circuitry will toggle this bit and then the next data packet will be transmitted.</p> <p>This bit can be read and written and can only be toggled by writing 1. It can also be toggled by the software to initialize its value under certain conditions.</p>

## USB Endpoint 1 ~ 3 Interrupt Enable Register – USBEPIER, n = 1 ~ 3

This register specifies the Endpoint 1 ~ 3 interrupt enable control bits.

Offset: 0x02C (n = 1), 0x040 (n = 2), 0x054 (n = 3)

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
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	RW	0	RW	0	RW	0	
	UERIE	STLIE	NAKIE	IDTXIE	ITRXIE	ODOVIE	ODRXIE	OTRXIE	

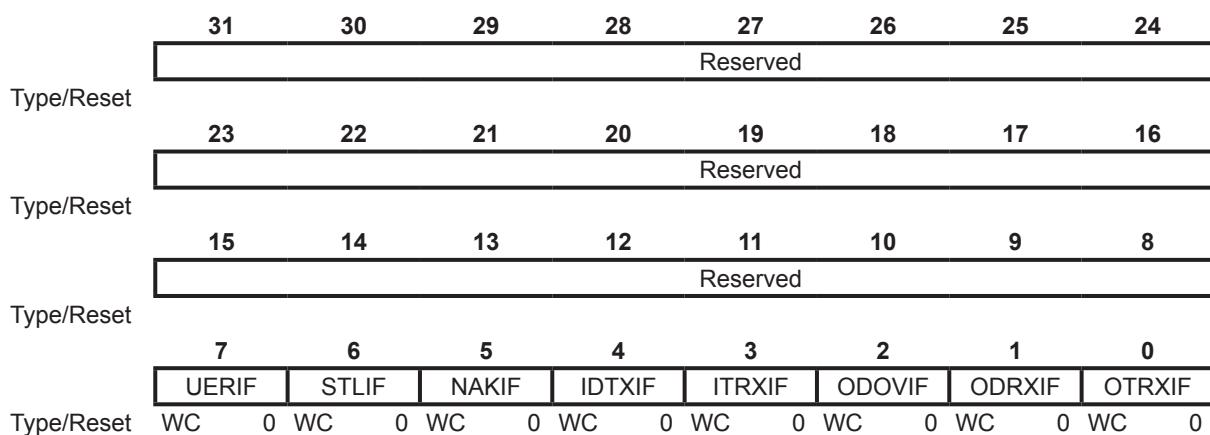
Bits	Field	Descriptions
[7]	UERIE	USB Error Interrupt Enable Control. 1: Enable interrupt 0: Disable interrupt
[6]	STLIE	STALL Transmitted Interrupt Enable Control. 1: Enable interrupt 0: Disable interrupt
[5]	NAKIE	NAK Transmitted Interrupt Enable Control. 1: Enable interrupt 0: Disable interrupt
[4]	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 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



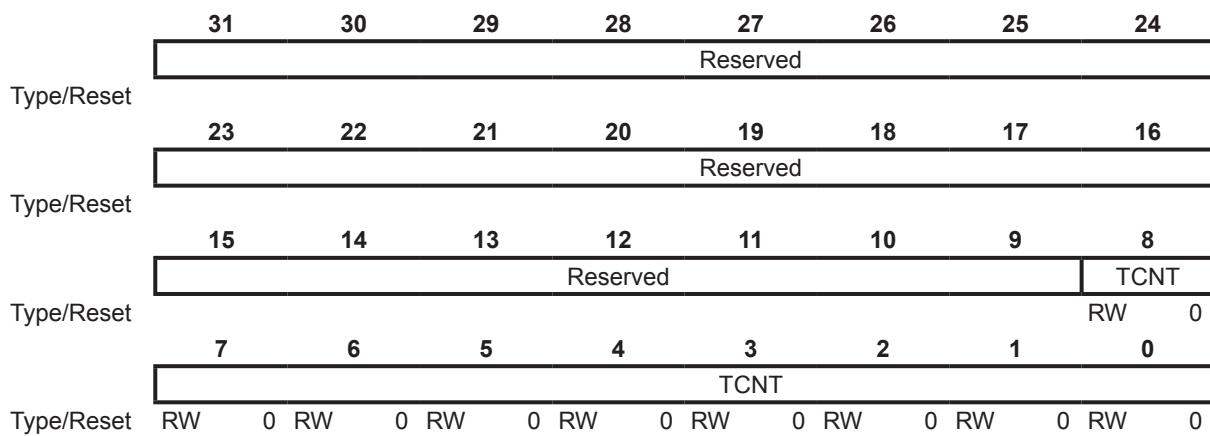
Bits	Field	Descriptions
[7]	UERIF	USB Error Interrupt Flag. This bit is set by the hardware when an error occurs during the transaction. Writing 1 into this status bit will clear it to 0.
[6]	STLIF	STALL Transmitted Interrupt Flag. This bit is set by hardware circuitry when a STALL-token is sent in response 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.
[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	<p>Transfer Byte Count</p> <p>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.</p>

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

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]	EPLN	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 EPLN 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 – USBEPPnCSR, n = 4 ~ 7

This register specifies the Endpoint 4 ~ 7 control and status bits.

Offset: 0x064 (n = 4), 0x078 (n = 5), 0x08C (n = 6), 0x0A0 (n = 7)

Reset value: 0x0000\_0002

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

Bits	Field	Descriptions
[7]	UDBTG	<p>USB Double Buffer Toggle bit</p> <p>The UDBTG and MDBTG bits are used to indicate which data buffer is accessed by the USB SIE hardware and which data buffer is accessed by the 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_BUFO
	0	1	EP_BUFO	EP_BUFI
	1	1	None*	EP_BUFI
	1	0	EP_BUFI	EP_BUFO

Transaction Type	UDBTG	MDBTG	Buffer written by SIE	Buffer read by CPU
OUT	0	0	None*	EP_BUFI
	0	1	EP_BUFO	EP_BUFI
	1	1	None*	EP_BUFO
	1	0	EP_BUFI	EP_BUFO

\* 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]→...

<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>
[6]	MDBTG	CPU Double Buffer Toggle bit  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
[5]	STLRX	STALL bit for reception transfers  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.
[4]	NAKRX	NAK bit for reception transfers  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.
[3]	DTGRX	Data Toggle bit for reception transfers  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.
[2]	STLTX	STALL bit for transmission transfers  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.
[1]	NAKTX	NAK bit for transmission transfers  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.

<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>
[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

	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	RW	0	RW	0	RW	0	
	UERIE	STLIE	NAKIE	IDTXIE	ITRXIE	ODOVIE	ODRXIE	OTRXIE	

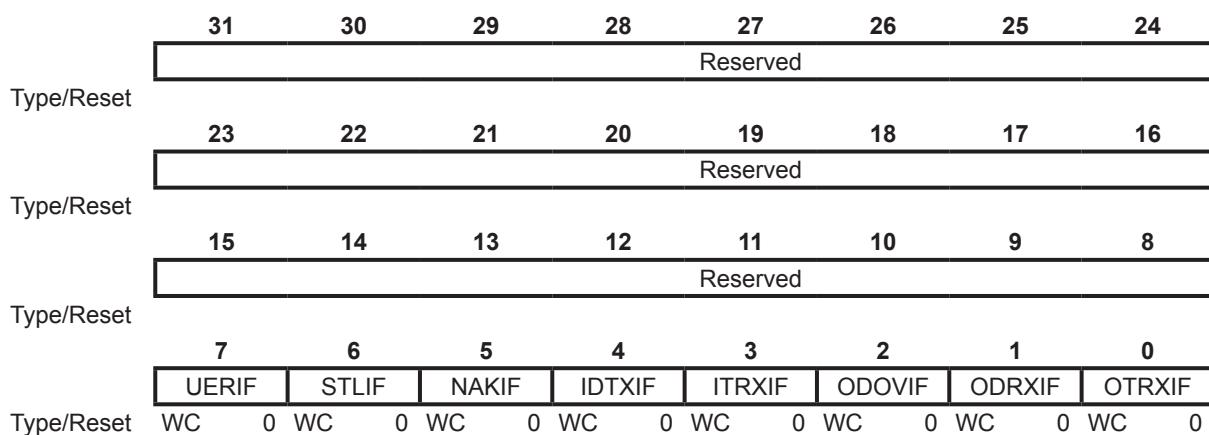
Bits	Field	Descriptions
[7]	UERIE	USB Error Interrupt Enable Control 1: Enable interrupt 0: Disable interrupt
[6]	STLIE	STALL Transmitted Interrupt Enable Control 1: Enable interrupt 0: Disable interrupt
[5]	NAKIE	NAK Transmitted Interrupt Enable Control 1: Enable interrupt 0: Disable interrupt
[4]	IDTXIE	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	RW	0	RW	0	RW	0	RW	0
				TCNT1				
	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	RW	0	RW	0	RW	0	RW	0
				TCNT0				

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

# 25 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-bit or 32-bit output remainder. Ordinarily, a data stream is suffixed by a CRC code and used as a checksum when being sent or stored. Therefore, the received or restored data stream is calculated by the same generator polynomial as described above. If the new CRC code result does not match the one calculated earlier, that means data stream contains a data error.

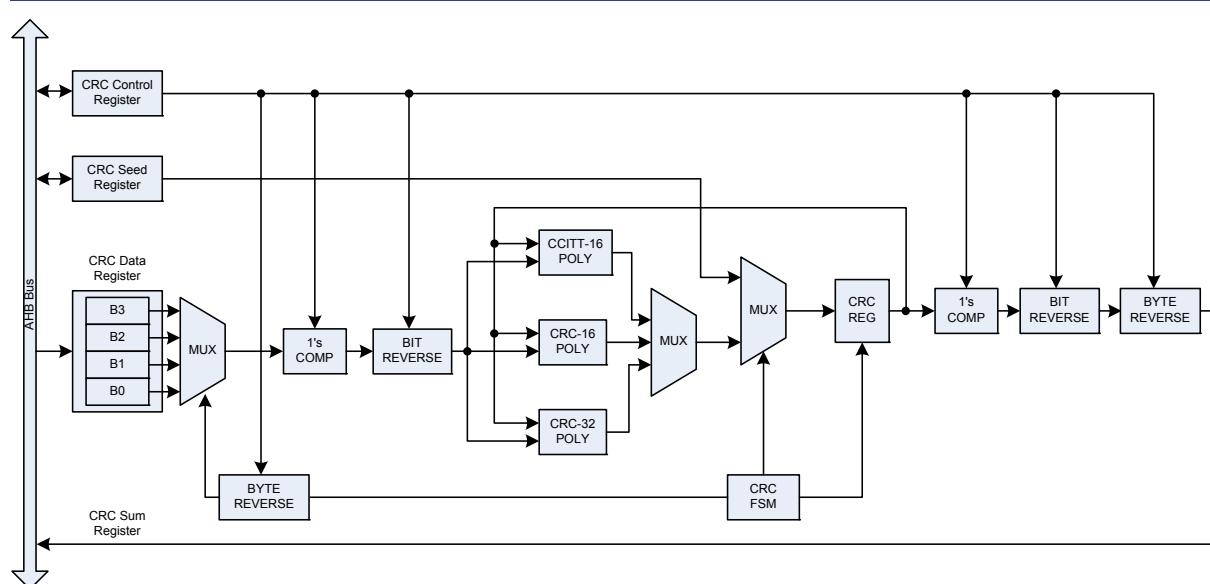


Figure 178. CRC Block Diagram

## 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

## 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 177. 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 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 CRCCR register with any byte writes but the bit reversal setting DATBIRV are still applied to the byte. The Figure 178 is shown the byte and bit reversal operation example.

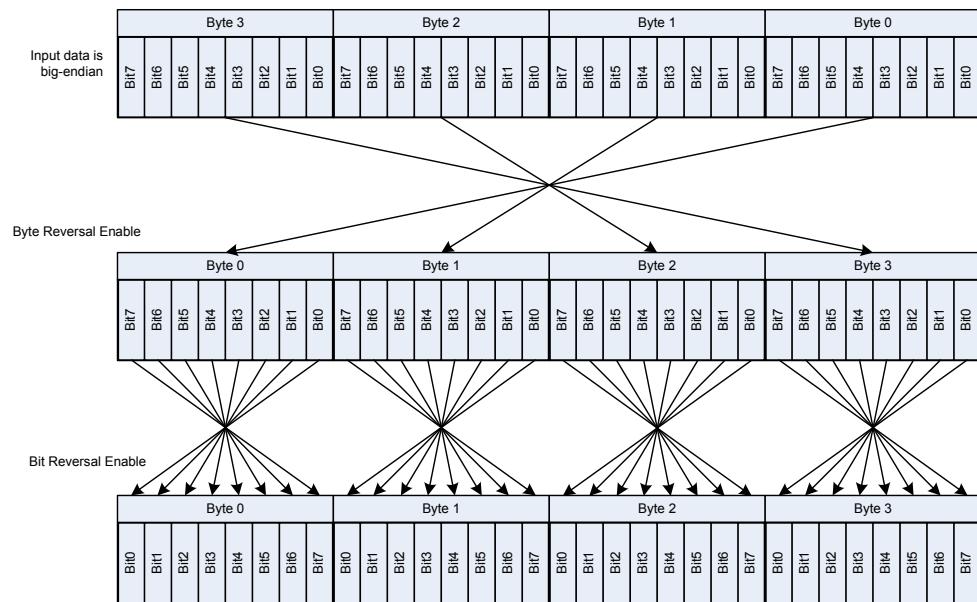


Figure 179. CRC Data Bit and Byte Reversal Example

## Register Map

The following table shows the CRC registers and reset values.

Table 67. CRC Register Map

Register	Offset	Description	Reset Value
<b>CRC Base Address = 0x4008_A000</b>			
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

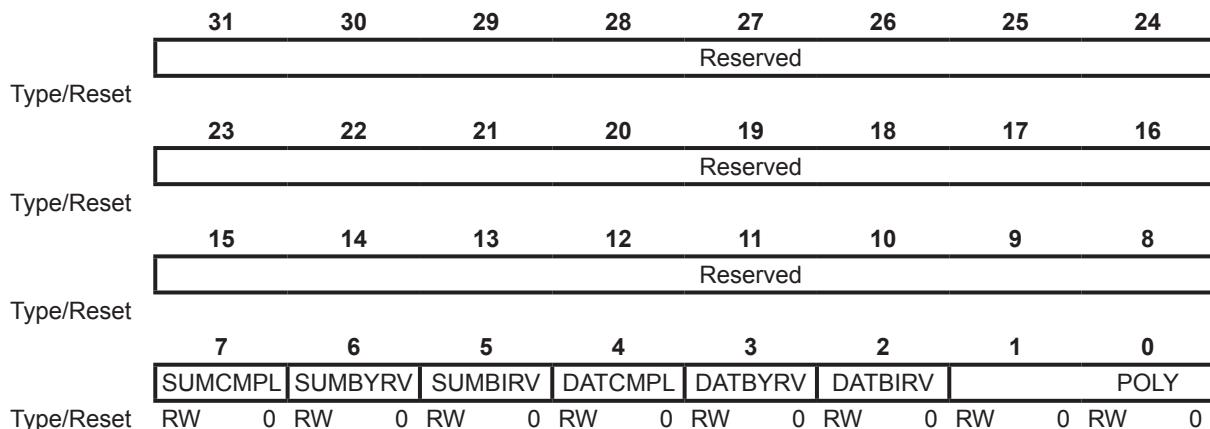
## Register Descriptions

### CRC Control Register – CRCCR

This register specifies the corresponding CRC function enable control.

Offset: 0x000

Reset value: 0x0000\_0000



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
23 22 21 20 19 18 17 16								
SEED								
Type/Reset	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
7 6 5 4 3 2 1 0								
SEED								
Type/Reset	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
Type/Reset	CHKSUM							
	RO	0	RO	0	RO	0	RO	0
	23	22	21	20	19	18	17	16
Type/Reset	CHKSUM							
	RO	0	RO	0	RO	0	RO	0
	15	14	13	12	11	10	9	8
Type/Reset	CHKSUM							
	RO	0	RO	0	RO	0	RO	0
	7	6	5	4	3	2	1	0
Type/Reset	CHKSUM							
	RO	0	RO	0	RO	0	RO	0

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

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

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/en/>.