



---

Holtek 32-bit Microcontroller with ARM® Cortex™-M3 Core

# **HT32F1653/HT32F1654 Series User Manual**

Revision: V1.00 Date: January 14, 2015

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

## Table of Contents

<b>1 Introduction .....</b>	<b>26</b>
Overview .....	26
Features.....	27
Device Information .....	31
Block Diagram .....	32
<b>2 Document Conventions .....</b>	<b>33</b>
<b>3 System Architecture .....</b>	<b>34</b>
ARM® Cortex™-M3 Processor .....	34
Bus Architecture.....	35
Memory Organization .....	36
Memory Map.....	37
Embedded Flash Memory.....	40
Embedded SRAM Memory .....	40
AHB Peripherals .....	40
APB Peripherals .....	40
<b>4 Flash Memory Controller (FMC) .....</b>	<b>41</b>
Introduction .....	41
Features.....	41
Function Descriptions .....	42
Flash Memory Map .....	42
Flash Memory Architecture .....	43
Wait State Setting .....	43
Booting Configuration .....	44
Page Erase .....	45
Mass Erase.....	46
Word Programming.....	47
Option Byte Description .....	48
Page Erase/Program Protection .....	48
Security Protection.....	50
Register Map .....	51
Register Descriptions.....	52
Flash Target Address Register – TADR .....	52
Flash Write Data Register – WRDR .....	53
Flash Operation Command Register – OCMR .....	54
Flash Operation Control Register – OPCR.....	55
Flash Operation Interrupt Enable Register – OIER .....	56
Flash Operation Interrupt and Status Register – OISR .....	57
Flash Page Erase/Program Protection Status Register – PPSR.....	59
Flash Security Protection Status Register – CPSR .....	60

Flash Vector Mapping Control Register – VMCR .....	61
Flash Pre-fetch Control Register – CFCR .....	62
SRAM Booting Vector Register n – SBVTn, n = 0 ~ 3 .....	64
<b>5 Power Control Unit (PWRCU) .....</b>	<b>65</b>
Introduction .....	65
Features.....	65
Functional Descriptions .....	66
Backup Domain .....	66
3.3V Power Domain.....	67
1.8V Power Domain.....	68
Operation Modes .....	68
Register Map .....	71
Register Descriptions.....	72
Backup Domain Status Register – BAKSR.....	72
Backup Domain Control Register – BAKCR.....	73
Backup Domain Test Register – BAKTEST .....	75
HSI Ready Counter Control Register – HSIRCR.....	76
Low Voltage/Brown Out Detect Control and Status Register – LVDCSR .....	77
Backup Register n – BAKREGn, n = 0 ~ 9 .....	79
<b>6 Clock Control Unit (CKCU) .....</b>	<b>80</b>
Introduction .....	80
Features.....	82
Functional Descriptions .....	82
High Speed External Crystal Oscillator – HSE .....	82
High Speed Internal RC Oscillator – HSI .....	83
Auto Trimming of High Speed Internal RC Oscillator – HSI.....	83
Phase Locked Loop – PLL.....	85
Low Speed External Crystal Oscillator – LSE.....	87
Low Speed Internal RC Oscillator – LSI .....	87
Clock Ready Flag .....	87
System Clock Selection – CK_SYS.....	88
HSE Clock Monitor .....	88
Clock Output Capability .....	89
Register Map .....	89
Register Descriptions.....	90
Global Clock Configuration Register – GCFGR.....	90
Global Clock Control Register – GCCR.....	92
Global Clock Status Register – GCSR .....	94
Global Clock Interrupt Register – GCIR.....	95
PLL Configuration Register – PLLCFG.....	97
PLL Control Register – PLLCR.....	98
AHB Configuration Register – AHBCFGR .....	99

AHB Clock Control Register – AHCCR .....	100
APB Configuration Register – APBCFGR.....	102
APB Clock Control Register 0 – APBCCR0.....	103
APB Clock Control Register 1 – APBCCR1.....	105
Clock Source Status Register – CKST .....	107
APB Peripheral Clock Selection Register 0 – APBPCSR0 .....	108
APB Peripheral Clock Selection Register 1 – APBPCSR1 .....	110
HSI Control Register – HSICR.....	112
HSI Auto Trimming Counter Register – HSIATCR.....	114
Low Power Control Register – LPCR .....	115
MCU Debug Control Register – MCUDBGCR.....	116
<b>7 Reset Control Unit (RSTCU).....</b>	<b>119</b>
Introduction .....	119
Functional Descriptions .....	120
Power On Reset .....	120
System Reset .....	120
AHB and APB Unit Reset.....	120
Register Map .....	121
Register Descriptions.....	121
Global Reset Status Register – GRSR.....	121
AHB Peripheral Reset Register – AHBPRSTR.....	122
APB Peripheral Reset Register 0 – APBPRSTR0 .....	123
APB Peripheral Reset Register 1 – APBPRSTR1 .....	125
<b>8 General Purpose I/O (GPIO).....</b>	<b>127</b>
Introduction .....	127
Features .....	128
Functional Descriptions .....	128
Default GPIO Pin Configuration.....	128
General Purpose I/O – GPIO .....	128
GPIO Locking Mechanism.....	130
Register Map .....	130
Register Descriptions.....	131
Port A Data Direction Control Register – PADIRCR .....	131
Port A Input Function Enable Control Register – PAINER.....	132
Port A Pull-Up Selection Register – PAPUR .....	133
Port A Pull-Down Selection Register – PAPDR .....	134
Port A Open Drain Selection Register – PAODR .....	135
Port A Output Current Drive Selection Register – PADRVR .....	136
Port A Lock Register – PALOCKR .....	137
Port A Data Input Register – PADINR.....	138
Port A Output Data Register – PADOUTR .....	139
Port A Output Set/Reset Control Register – PASRR .....	140

Port A Output Reset Register – PARR .....	141
Port B Data Direction Control Register – PBDIRCR .....	142
Port B Input Function Enable Control Register – PBINER .....	143
Port B Pull-Up Selection Register – PBPUR .....	144
Port B Pull-Down Selection Register – PBPDR .....	145
Port B Open Drain Selection Register – PBODR .....	146
Port B Output Current Drive Selection Register – PBDRV .....	147
Port B Lock Register – PBLOCKR .....	148
Port B Data Input Register – PBDINR .....	149
Port B Output Data Register – PBDOUTR .....	150
Port B Output Set/Reset Control Register – PBSRR .....	151
Port B Output Reset Register – PBRR .....	152
Port C Data Direction Control Register – PCDIRCR .....	153
Port C Input Function Enable Control Register – PCINER .....	154
Port C Pull-Up Selection Register – PCPUR .....	155
Port C Pull-Down Selection Register – PCPDR .....	156
Port C Open Drain Selection Register – PCODR .....	157
Port C Output Current Drive Selection Register – PCDRV .....	158
Port C Lock Register – PCLOCKR .....	159
Port C Data Input Register – PCDINR .....	160
Port C Output Data Register – PCDOUTR .....	161
Port C Output Set/Reset Control Register – PCSRR .....	162
Port C Output Reset Register – PCRR .....	163
Port D Data Direction Control Register – PDDIRCR .....	164
Port D Input Function Enable Control Register – PDINER .....	165
Port D Pull-Up Selection Register – PDPUR .....	166
Port D Pull-Down Selection Register – PDPDR .....	167
Port D Open Drain Selection Register – PDODR .....	168
Port D Lock Register – PDLOCKR .....	169
Port D Data Input Register – PDDINR .....	170
Port D Output Data Register – PDDOUTR .....	171
Port D Output Set/Reset Control Register – PDSRR .....	172
Port D Output Reset Register – PDRR .....	173
<b>9 Alternate Function Input/Output Control Unit (AFIO).....</b>	<b>174</b>
Introduction .....	174
Features .....	175
Functional Descriptions .....	175
External Interrupt Pin Selection .....	175
Alternate Function .....	176
Lock Mechanism .....	176
Register Map .....	176
Register Descriptions .....	177
EXTI Source Selection Register 0 – ESSR0 .....	177
EXTI Source Selection Register 1 – ESSR1 .....	178

GPIO Alternate Function Configuration Register 0 – GPxCFG0 (x= A, B, C, D).....	179
GPIO Alternate Function Configuration Register 1 – GPxCFG1 (x= A, B, C, D).....	180
<b>10 Nested Vectored Interrupt Controller (NVIC).....</b>	<b>181</b>
Introduction .....	181
Features.....	183
Function Descriptions .....	184
SysTick Calibration .....	184
Register Map .....	184
<b>11 External Interrupt/Event Controller (EXTI).....</b>	<b>186</b>
Introduction .....	186
Features.....	186
Function Descriptions .....	187
Wakeup Event Management.....	187
External Interrupt/Event Line Mapping .....	188
Interrupt and Debounce .....	188
Register Map .....	189
Register Descriptions.....	190
EXTI Interrupt Configuration Register n – EXTICFGn, n = 0 ~ 15 .....	190
EXTI Interrupt Control Register – EXTICR .....	191
EXTI Interrupt Edge Flag Register – EXTIEDGEFLGR.....	192
EXTI Interrupt Edge Status Register – EXTIEDGESR .....	193
EXTI Interrupt Software Set Command Register – EXTISSCR.....	194
EXTI Interrupt Wakeup Control Register – EXTIWAKUPCR .....	195
EXTI Interrupt Wakeup Polarity Register – EXTIWAKUPPOLR .....	196
EXTI Interrupt Wakeup Flag Register – EXTIWAKUPFLG .....	197
<b>12 Analog to Digital Converter (ADC) .....</b>	<b>198</b>
Introduction .....	198
Features.....	199
Functional Description .....	200
ADC Clock Setup.....	200
Regular and High Priority Channel Selection .....	200
Conversion Modes.....	200
Start Conversion on External Event.....	206
High Priority Group Management .....	206
Sampling Time Setting.....	208
Data Format and Alignment .....	208
Analog Watchdog.....	209
Interrupts.....	209
PDMA Request .....	210
Register Map .....	210
Register Descriptions.....	212

ADC Reset Register – ADCRST .....	212
ADC Regular Conversion Mode Register – ADCCONV .....	213
ADC High Priority Conversion Mode Register – ADCHCONV .....	214
ADC Regular Conversion List Register 0 – ADCLST0 .....	215
ADC Regular Conversion List Register 1 – ADCLST1 .....	216
ADC Regular Conversion List Register 2 – ADCLST2 .....	217
ADC Regular Conversion List Register 3 – ADCLST3 .....	218
ADC High Priority Conversion List Register – ADCHLST .....	219
ADC Input Offset Register n – ADCOFRn, n = 0 ~ 15 .....	220
ADC Input Sampling Time Register n – ADCSTRn, n = 0 ~ 15 .....	221
ADC Regular Conversion Data Register y – ADCDRy, y = 0 ~ 15 .....	222
ADC High Priority Conversion Data Register y – ADCHDRy, y = 0 ~ 3 .....	223
ADC Regular Trigger Control Register – ADCTCR .....	224
ADC Regular Trigger Source Register – ADCTS .....	225
ADC High Priority Trigger Control Register – ADCHTCR .....	226
ADC High Priority Trigger Source Register – ADCHTS .....	227
ADC Watchdog Control Register – ADCWCR .....	228
ADC Watchdog Lower Threshold Register – ADCLTR .....	229
ADC Watchdog Upper Threshold Register – ADCUTR .....	230
ADC Interrupt Enable Register – ADCIER .....	231
ADC Interrupt Raw Status Register – ADCIRAW .....	232
ADC Interrupt Status Register – ADCISR .....	233
ADC Interrupt Clear Register – ADCICLR .....	235
ADC DMA Request Register – ADCDMAR .....	236
<b>13 Comparator (CMP) .....</b>	<b>237</b>
Introduction .....	237
Features .....	237
Function Descriptions .....	238
Comparator Inputs and Output .....	238
Comparator Voltage Reference .....	238
Interrupts and Wake-up .....	239
Power Mode and Hysteresis .....	240
Comparator Write-Protected mechanism .....	240
Register Map .....	240
Register Descriptions .....	241
Comparator Control Register n – CMPCRn, n = 0 or 1 .....	241
Comparator Voltage Reference Value Register n – CVRVALRn, n = 0 or 1 .....	243
Comparator Interrupt Enable Register n – CMPIERn, n = 0 or 1 .....	244
Comparator Transition Flag Register n – CMPTFRn, n = 0 or 1 .....	245
<b>14 General-Purpose Timers (GPTM) .....</b>	<b>246</b>
Introduction .....	246
Features .....	247

Functional Descriptions .....	248
Counter Mode .....	248
Clock Controller .....	251
Trigger Controller .....	252
Slave Controller .....	254
Master Controller .....	257
Channel Controller .....	258
Input Stage .....	261
Output Stage .....	262
Update Management .....	266
Quadrature Decoder .....	267
Digital Filter .....	269
Clearing the CHxOREF when ETIF is high .....	269
Single Pulse Mode .....	270
Asymmetric PWM Mode .....	272
Timer Interconnection .....	273
Trigger ADC Start .....	276
PDMA Request .....	276
Register Map .....	277
Register Descriptions .....	278
Timer Counter Configuration Register – CNTCFR .....	278
Timer Mode Configuration Register – MDCFR .....	280
Timer Trigger Configuration Register – TRCFR .....	283
Timer Counter Register – CTR .....	285
Channel 0 Input Configuration Register – CH0ICFR .....	286
Channel 1 Input Configuration Register – CH1ICFR .....	288
Channel 2 Input Configuration Register – CH2ICFR .....	290
Channel 3 Input Configuration Register – CH3ICFR .....	292
Channel 0 Output Configuration Register – CH0OCFR .....	294
Channel 1 Output Configuration Register – CH1OCFR .....	296
Channel 2 Output Configuration Register – CH2OCFR .....	298
Channel 3 Output Configuration Register – CH3OCFR .....	300
Channel Control Register – CHCTR .....	302
Channel Polarity Configuration Register – CHPOLR .....	303
Timer PDMA/Interrupt Control Register – DICTR .....	304
Timer Event Generator Register – EVGR .....	306
Timer Interrupt Status Register – INTSR .....	308
Timer Counter Register – CNTR .....	311
Timer Prescaler Register – PSCR .....	312
Timer Counter Reload Register – CRR .....	313
Channel 0 Capture/Compare Register – CH0CCR .....	314
Channel 1 Capture/Compare Register – CH1CCR .....	315
Channel 2 Capture/Compare Register – CH2CCR .....	316
Channel 3 Capture/Compare Register – CH3CCR .....	317
Channel 0 Asymmetric Compare Register – CH0ACR .....	318

Channel 1 Asymmetric Compare Register – CH1ACR.....	319
Channel 2 Asymmetric Compare Register – CH2ACR.....	320
Channel 3 Asymmetric Compare Register – CH3ACR.....	321
<b>15 Basic Function Timer (BFTM).....</b>	<b>322</b>
Introduction .....	322
Features.....	322
Functional Description .....	323
Repetitive Mode.....	323
One Shot Mode.....	324
Trigger ADC Start.....	324
Register Map .....	325
Register Descriptions.....	325
BFTM Control Register – BFTMCR.....	325
BFTM Status Register – BFTMSR.....	326
BFTM Counter Register – BFTMCNTR .....	327
BFTM Compare Value Register – BFTMCMPR .....	328
<b>16 Motor Control Timer (MCTM) .....</b>	<b>329</b>
Introduction .....	329
Features.....	330
Functional Descriptions .....	331
Counter Mode .....	331
Clock Controller .....	335
Trigger Controller .....	336
Slave Controller .....	338
Master Controller .....	340
Channel Controller.....	341
Input Stage .....	344
Output Stage.....	345
Update Management .....	356
Quadrature Decoder .....	358
Digital Filter.....	360
Clearing CHxOREF when ETIF is high.....	360
Single Pulse Mode.....	361
Asymmetric PWM Mode .....	363
Timer Interconnection .....	364
Trigger ADC Start.....	368
Lock Level Table .....	368
PDMA Request .....	369
Register Map .....	370
Register Descriptions.....	371
Timer Counter Configuration Register – CNTCFR .....	371
Timer Mode Configuration Register – MDCFR .....	373

Timer Trigger Configuration Register – TRCFR.....	376
Timer Counter Register – CTR .....	378
Channel 0 Input Configuration Register – CH0ICFR .....	379
Channel 1 Input Configuration Register – CH1ICFR.....	381
Channel 2 Input Configuration Register – CH2ICFR .....	383
Channel 3 Input Configuration Register – CH3ICFR.....	385
Channel 0 Output Configuration Register – CH0OCFR .....	387
Channel 1 Output Configuration Register – CH1OCFR .....	389
Channel 2 Output Configuration Register – CH2OCFR .....	391
Channel 3 Output Configuration Register – CH3OCFR .....	393
Channel Control Register – CHCTR .....	395
Channel Polarity Configuration Register – CHPOLR.....	397
Channel Break Configuration Register – CHBRKCFR .....	399
Channel Break Control Register – CHBRKCTR .....	400
Timer PDMA/Interrupt Control Register – DICTR .....	403
Timer Event Generator Register – EVGR.....	405
Timer Interrupt Status Register – INTSR.....	407
Timer Counter Register – CNTR.....	410
Timer Prescaler Register – PSCR .....	411
Timer Counter Reload Register – CRR .....	412
Timer Repetition Register – REPR .....	413
Channel 0 Capture/Compare Register – CH0CCR .....	414
Channel 1 Capture/Compare Register – CH1CCR .....	415
Channel 2 Capture/Compare Register – CH2CCR .....	416
Channel 3 Capture/Compare Register – CH3CCR .....	417
Channel 0 Asymmetric Compare Register – CH0ACR.....	418
Channel 1 Asymmetric Compare Register – CH1ACR.....	419
Channel 2 Asymmetric Compare Register – CH2ACR.....	420
Channel 3 Asymmetric Compare Register – CH3ACR.....	421
<b>17 Real Time Clock (RTC) .....</b>	<b>422</b>
Introduction .....	422
Features.....	422
Functional Descriptions .....	423
RTC Related Register Reset .....	423
Reading RTC Register.....	423
Low Speed Clock Configuration .....	423
RTC Counter Operation.....	424
Interrupt and Wakeup Control.....	424
RTCOUT Output Pin Configuration.....	425
Register Map .....	426
Register Descriptions.....	426
RTC Counter Register – RTCCNT.....	426
RTC Compare Register – RTCCMP .....	427

RTC Control Register – RTCCR .....	428
RTC Status Register – RTCCSR .....	430
RTC Interrupt and Wakeup Enable Register – RTCIWEN .....	431
<b>18 Watchdog Timer (WDT) .....</b>	<b>432</b>
Introduction .....	432
Features .....	432
Functional Descriptions .....	433
Register Map .....	435
Register Descriptions .....	435
Watchdog Timer Control Register – WDTCR .....	435
Watchdog Timer Mode Register 0 – WDTMR0 .....	436
Watchdog Timer Mode Register 1 – WDTMR1 .....	437
Watchdog Timer Status Register – WDTSR .....	438
Watchdog Timer Protection Register – WDTPR .....	439
Watchdog Timer Counter Register – WDTCNTR .....	440
Watchdog Timer Clock Selection Register – WDTCSR .....	441
<b>19 Inter-Integrated Circuit (I<sup>2</sup>C) .....</b>	<b>442</b>
Introduction .....	442
Features .....	443
Functional Descriptions .....	443
Two Wire Serial Interface .....	443
START and STOP Conditions .....	443
Data Validity .....	444
Addressing Format .....	445
Data Transfer and Acknowledge .....	447
Clock Synchronization .....	448
Arbitration .....	448
General Call Address .....	449
Bus Error .....	449
Address Mask Enable .....	449
Address Snoop .....	449
Operation Mode .....	449
Conditions of Holding SCL Line .....	455
I <sup>2</sup> C Timeout Function .....	456
PDMA Interface .....	456
Register Map .....	457
Register Descriptions .....	458
I <sup>2</sup> C Control Register – I2CCR .....	458
I <sup>2</sup> C Interrupt Enable Register – I2CIER .....	460
I <sup>2</sup> C Address Register – I2CADDR .....	462
I <sup>2</sup> C Status Register – I2CSR .....	463
I <sup>2</sup> C SCL High Period Generation Register – I2CSHPGR .....	466

I <sup>2</sup> C SCL Low Period Generation Register – I2CSLPGR .....	467
I <sup>2</sup> C Data Register – I2CDR .....	468
I <sup>2</sup> C Target Register – I2CTAR .....	469
I <sup>2</sup> C Address Mask Register – I2CADDMR .....	470
I <sup>2</sup> C Address Snoop Register – I2CADDSSR .....	471
I <sup>2</sup> C Timeout Register – I2CTOUT .....	472
<b>20 Serial Peripheral Interface (SPI) .....</b>	<b>473</b>
Introduction .....	473
Features .....	474
Function Descriptions .....	474
Master Mode .....	474
Slave Mode .....	474
SPI Serial Frame Format .....	475
Status Flags .....	479
Register Map .....	482
Register Descriptions .....	482
SPI Control Register 0 – SPICR0 .....	482
SPI Control Register 1 – SPICR1 .....	484
SPI Interrupt Enable Register – SPIIER .....	486
SPI Clock Prescaler Register – SPICPR .....	487
SPI Data Register – SPIDR .....	488
SPI Status Register – SPISR .....	489
SPI FIFO Control Register – SPIFCR .....	491
SPI FIFO Status Register – SPIFSR .....	492
SPI FIFO Time Out Counter Register – SPIFTOCR .....	493
<b>21 Universal Synchronous Asynchronous Receiver Transmitter (USART) ....</b>	<b>494</b>
Introduction .....	494
Features .....	495
Function Descriptions .....	496
Serial Data Format .....	496
Baud Rate Generation .....	497
IrDA .....	498
RS485 Mode .....	500
Synchronous Mode .....	502
Hardware Flow Control .....	504
Interrupts and Status .....	505
PDMA Interface .....	505
Register Map .....	506
Register Descriptions .....	507
USART Receiver Buffer Register – RBR .....	507
USART Transmitter Buffer Register – TBR .....	508
USART Interrupt Enable Register – IER .....	509

USART Interrupt Identification Register – IIR .....	510
USART FIFO Control Register – FCR .....	512
USART Line Control Register – LCR.....	513
USART Modem Control Register – MODCR .....	514
USART Line Status Register – LSR .....	515
USART Modem Status Register – MODSR.....	517
USART Timing Parameter Register – TPR.....	518
USART Mode Register – MDR .....	519
USART IrDA Control Register – IrDACKR .....	520
USART RS485 Control Register – RS485CR.....	521
USART Synchronous Control Register – SYNCNR .....	522
USART FIFO Status Register – FSR.....	523
USART Divider Latch Register– DLR .....	524
USART Debug/Test Register – DEGTSTR.....	525
<b>22 Universal Asynchronous Receiver Transmitter (UART).....</b>	<b>526</b>
Introduction .....	526
Features.....	527
Function Descriptions .....	527
Serial Data Format.....	527
Baud Rate Generation .....	528
Interrupts and Status .....	529
PDMA Interface.....	529
Register Map .....	529
Register Descriptions.....	530
UART Receiver Buffer Register – RBR .....	530
UART Transmitter Buffer Register – TBR .....	531
UART Interrupt Enable Register – IER .....	532
UART Interrupt Identification Register – IIR.....	533
UART FIFO Control Register – FCR.....	534
UART Line Control Register – LCR .....	535
UART Line Status Register – LSR.....	536
UART Timing Parameter Register – TPR .....	538
UART Mode Register – MDR.....	539
UART FIFO Status Register – FSR .....	540
UART Divider Latch Register – DLR.....	541
UART Debug/Test Register – DEGTSTR .....	542
<b>23 Smart Card Interface (SCI) .....</b>	<b>543</b>
Introduction .....	543
Features.....	544
Functional Descriptions .....	544
Elementary Time Unit Counter.....	544
Guard Time Counter .....	546

Waiting Time Counter .....	547
Card Clock and Data Selection.....	548
Card Detection .....	548
SCI Data Transfer Mode.....	549
Interrupt Generator .....	551
PDMA Interface.....	552
Register Map .....	552
Register Descriptions.....	553
SCI Control Register – CR.....	553
SCI Status Register – SR .....	555
SCI Contact Control Register – CCR.....	557
SCI Elementary Time Unit Register – ETUR .....	558
SCI Guard Time Register – GTR.....	559
SCI Waiting Time Register – WTR.....	560
SCI Interrupt Enable Register – IER.....	561
SCI Interrupt Pending Register – IPR.....	563
SCI Transmit Buffer – TXB.....	565
SCI Receive Buffer – RXB.....	566
SCI Prescaler Register – PSCR .....	567
<b>24 USB Device Controller (USB) .....</b>	<b>568</b>
Introduction .....	568
Features.....	568
Functional Descriptions .....	569
Endpoints.....	569
EP-SRAM .....	569
Serial Interface Engine – SIE.....	570
Double-Buffering .....	570
Suspend Mode and Wake-up .....	572
Remote Wake-up .....	572
Register Map .....	572
Register Descriptions.....	574
USB Control and Status Register – USBCSR .....	574
USB Interrupt Enable Register – USBIER .....	576
USB Interrupt Status Register – USBISR .....	577
USB Frame Count Register – USBFCR .....	579
USB Device Address Register – USBDEVA .....	580
USB Endpoint 0 Control and Status Register – USBEP0CSR .....	581
USB Endpoint 0 Interrupt Enable Register – USBEP0IER .....	583
USB Endpoint 0 Interrupt Status Register – USBEP0ISR .....	585
USB Endpoint 0 Transfer Count Register – USBEP0TCR .....	587
USB Endpoint 0 Configuration Register – USBEP0CFGREG .....	588
USB Endpoint 1 ~ 3 Control and Status Register – USBEPnCSR, n = 1 ~ 3.....	589
USB Endpoint 1 ~ 3 Interrupt Enable Register – USBEPnIER, n = 1 ~ 3.....	591

USB Endpoint 1 ~ 3 Interrupt Status Register – USBEPnISR, n = 1 ~ 3.....	592
USB Endpoint 1 ~ 3 Transfer Count Register – USBEPnTCR, n = 1 ~ 3 .....	593
USB Endpoint 1 ~ 3 Configuration Register – USBEPnCFGR, n = 1 ~ 3.....	594
USB Endpoint 4 ~ 7 Control and Status Register – USBEPnCSR, n = 4 ~ 7.....	595
USB Endpoint 4 ~ 7 Interrupt Enable Register – USBEPnIER, n = 4 ~ 7.....	598
USB Endpoint 4 ~ 7 Interrupt Status Register – USBEPnISR, n = 4 ~ 7.....	599
USB Endpoint 4 ~ 7 Transfer Count Register – USBEPnTCR, n = 4 ~ 7 .....	600
USB Endpoint 4 ~ 7 Configuration Register – USBEPnCFGR, n = 4 ~ 7.....	601
<b>25 Peripheral Direct Memory Access (PDMA).....</b>	<b>602</b>
Introduction .....	602
Features.....	602
Functional Description .....	603
AHB Master .....	603
PDMA Channel .....	603
PDMA Request Mapping .....	603
Channel transfer .....	605
Channel Priority .....	605
Transfer Request .....	606
Address Mode.....	606
Auto-Reload.....	606
Transfer Interrupt .....	607
Register Map .....	607
Register Descriptions.....	609
PDMA Channel n Control Register – PDMACHnCR, n=0~7 .....	609
PDMA Channel n Source Address Register – PDMACHnSADR, n=0~7.....	611
PDMA Channel n Destination Address Register – PDMACHnDADR, n=0~7.....	612
PDMA Channel n Current Address Register – PDMACHnCADR, n=0~7.....	613
PDMA Channel n Transfer Size Register – PDMACHnTSR, n=0~7.....	614
PDMA Channel n Current Transfer Size Register – PDMACHnCTS, n=0~7 .....	615
PDMA Interrupt Status Register 0 – PDMAISR0 .....	616
PDMA Interrupt Status Register 1 – PDMAISR1 .....	618
PDMA Interrupt Status Clear Register 0 – PDMAISCR0 .....	620
PDMA Interrupt Status Clear Register 1 – PDMAISCR1 .....	621
PDMA Interrupt Enable Register 0 – PDMAIER0 .....	622
PDMA Interrupt Enable Register 1 – PDMAIER1 .....	623
<b>26 Extend Bus Interface (EBI).....</b>	<b>624</b>
Introduction .....	624
Features.....	624
Functional Descriptions .....	625
Non-multiplexed 8-bit Data 8-bit Address Mode .....	626
Non-multiplexed 16-bit Data N-bit Address Mode.....	627
Multiplexed 16-bit Data, 16-bit Address Mode .....	628

Multiplexed 8-bit Data, 24-bit Address Mode .....	629
Page Read Operation .....	630
Write Buffer and EBI Status .....	633
Bus Turn-around and Idle Cycles .....	633
AHB Transaction Width Conversion .....	634
EBI Bank Access .....	636
EBI Ready .....	637
PDMA Request .....	637
Register Map .....	637
Register Descriptions .....	638
EBI Control Register – EBICR .....	638
EBI Page Control Register – EBIPCR .....	641
EBI Status Register – EBISR .....	642
EBI Address Timing Register n – EBIATRn, n = 0 ~ 3 .....	643
EBI Read Timing Register n – EBIRTRn, n = 0 ~ 3 .....	644
EBI Write Timing Register n – EBIWTRn, n = 0 ~ 3 .....	645
EBI Parity Register n – EBIPR, n = 0 ~ 3 .....	646
EBI Interrupt Enable Register – EBIENR .....	647
EBI Interrupt Flag Register – EBIIFR .....	648
EBI Interrupt Clear Register – EBIIFCR .....	649
<b>27 Inter-IC Sound (I<sup>2</sup>S) .....</b>	<b>650</b>
Introduction .....	650
Features .....	650
Functional Description .....	651
I <sup>2</sup> S Master and Slave Mode .....	651
I <sup>2</sup> S Clock Rate Generator .....	652
I <sup>2</sup> S Interface Format .....	654
FIFO Control and Arrangement .....	661
PDMA and Interrupt .....	662
Register Map .....	662
Register Descriptions .....	663
I <sup>2</sup> S Control Register – I2SCR .....	663
I <sup>2</sup> S Interrupt Enable Register – I2SIER .....	665
I <sup>2</sup> S Clock Divider Register – I2SCDR .....	666
I <sup>2</sup> S Tx Data Register – I2STXDR .....	667
I <sup>2</sup> S Rx Data Register – I2SRXDR .....	667
I <sup>2</sup> S FIFO Control Register – I2SFCR .....	668
I <sup>2</sup> S Status Register – I2SSR .....	669
I <sup>2</sup> S Rate Counter Value Register – I2SRCNTR .....	671
<b>28 Cyclic Redundancy Check (CRC) .....</b>	<b>672</b>
Introduction .....	672
Features .....	673

Function Descriptions .....	673
CRC Computation.....	673
Byte and Bit Reversal for CRC Computation.....	673
CRC with PDMA .....	674
Register Map .....	674
Register Descriptions.....	675
CRC Control Register – CRCCR .....	675
CRC Seed Register – CRCSDR.....	676
CRC Checksum Register – CRCCSR .....	677
CRC Data Register – CRCDR .....	678

## List of Tables

Table 1. HT32F1653/1654 Series Features and Peripheral List.....	31
Table 2. Document Conventions .....	33
Table 3. HT32F1653/1654 Register Map.....	38
Table 4. Flash Memory and Option Byte.....	43
Table 5. Relationship between wait state cycle and HCLK.....	43
Table 6. Boot Modes .....	44
Table 7. Option Byte Memory Map .....	48
Table 8. Access Permission of Protected Main Flash Page.....	49
Table 9. Access Permission When Security Protection is Enabled.....	50
Table 10. FMC Register Map .....	51
Table 11. Operation Mode Definitions .....	68
Table 12. Enter/Exit Power Saving Modes.....	69
Table 13. Power Status After System Reset.....	70
Table 14. PWRCU Register Map .....	71
Table 15. Output Divider 2 Setting .....	86
Table 16. Feedback Divider 2 Value Setting .....	86
Table 17. CKOUT Clock Source .....	89
Table 18. CKCU Register Map.....	89
Table 19. RSTCU Register Map .....	121
Table 20. AFIO, GPIO and IO Pad Control Signal True Table.....	129
Table 21. GPIO Register Map .....	130
Table 22. AFIO Selection for Peripheral Map.....	176
Table 23. AFIO Register Map.....	176
Table 24. Exception Types .....	181
Table 25. NVIC Register Map .....	184
Table 26. EXTI Register Map.....	189
Table 27. Data Format in ADCDRy[15:0] (y= 0~15) and ADCHDRy[15:0] (y= 0~3) .....	208
Table 28. A/D Converter Register Map .....	210
Table 29. CMP Register Map .....	240
Table 30. Counting Direction and Encoding Signals.....	268
Table 31. GPTM Register Map .....	277
Table 32. GPTM Internal Trigger Connection .....	284
Table 33. BFTM Register Map .....	325
Table 34. Compare Match Output Setup .....	346
Table 35. Output Control Bits for Complementary Output with a Break Event Occurrence .....	355
Table 36. Counting Direction and Encoding Signals.....	359
Table 37. Lock Level Table.....	368
Table 38. MCTM Register Map .....	370
Table 39. MCTM Internal Trigger Connection .....	377

Table 40. LSE Startup Mode Operating Current and Startup Time.....	423
Table 41. RTCOUT Output Mode and Active Level Setting .....	425
Table 42. RTC Register Map.....	426
Table 43. WDT Register Map.....	435
Table 44. Conditions of Holding SCL line.....	455
Table 45. I <sup>2</sup> C Register Map.....	457
Table 46. I <sup>2</sup> C Clock Setting Example .....	467
Table 47. SPI Interface Format Setup.....	475
Table 48. SPI Mode Fault Trigger Conditions .....	480
Table 49. SPI Master Mode SEL Pin Status .....	480
Table 50. SPI Register Map.....	482
Table 51. Baud Rate Error calculation – CK_USART = 72 MHz.....	497
Table 52. USART Register Map.....	506
Table 53. USART Interrupt Control Function .....	511
Table 54. Baud Rate Error Calculation – CK_UART = 72 MHz .....	528
Table 55. UART Register Map .....	529
Table 56. UART Interrupt Control Function.....	533
Table 57. DI Field Based D <sub>i</sub> Encoded Decimal Values.....	545
Table 58. FI Field Based F <sub>i</sub> Encoded Decimal Values .....	545
Table 59. Possible ETU Values Obtained with the F/D <sub>i</sub> Ratio.....	545
Table 60. SCI Register Map .....	552
Table 61. Endpoint Characteristics .....	569
Table 62. USB Data Types and Buffer Size .....	569
Table 63. USB Register Map .....	572
Table 64. Resume Event Detection .....	575
Table 65. PDMA Channel Assignments .....	604
Table 66. PDMA Address Modes .....	606
Table 67. PDMA Register Map.....	607
Table 68. The EBI Maps The AHB Transactions Width To External Device Transactions .....	634
Table 69. The EBI Maps The AHB Transactions Width To External Device Transactions Width Using Byte Lane EBI_BL[1:0].....	635
Table 70. Register map of EBI .....	637
Table 71. Recommend FS List @ 8 MHz PCLK .....	653
Table 72. Recommend FS List @ 48 MHz PCLK .....	653
Table 73. Recommend FS List @ 72 MHz PCLK .....	653
Table 74. I <sup>2</sup> S Register Map .....	662
Table 75. CRC Register Map .....	674

## List of Figures

Figure 1. HT32F1653/1654 Block Diagram .....	32
Figure 2. Cortex™-M3 Block Diagram .....	35
Figure 3. HT32F1653/1654 Bus Architecture .....	36
Figure 4. HT32F1653/1654 Memory Map.....	37
Figure 5. Flash Memory Controller Block Diagram.....	41
Figure 6. Flash Memory Map.....	42
Figure 7. Vector Remapping.....	44
Figure 8. Page Erase Operation Flowchart .....	45
Figure 9. Mass Erase Operation Flowchart .....	46
Figure 10. Word Programming Operation Flowchart .....	47
Figure 11. PWRCU Block Diagram.....	65
Figure 12. CKCU Block Diagram .....	81
Figure 13. External HSE Crystal, Ceramic, and Resonator .....	82
Figure 14. The Block Diagram of HSI Auto Trimming .....	84
Figure 15. PLL Block Diagram .....	85
Figure 16. External Crystal, Ceramic, and Resonators for LSE .....	87
Figure 17. RSTCU Block Diagram.....	119
Figure 18. Power On Reset Sequence .....	120
Figure 19. GPIO Block Diagram .....	127
Figure 20. AFIO(GPIO Control Signal.....	129
Figure 21. AFIO Block Diagram .....	174
Figure 22. EXTI Channel Input Selection .....	175
Figure 23. EXTI Block Diagram .....	186
Figure 24. EXTI Wake-up Event Management .....	187
Figure 25. EXTI Interrupt Debounce Function .....	188
Figure 26. ADC Block Diagram .....	198
Figure 27. One Shot Conversion Mode .....	201
Figure 28. Continuous Conversion Mode .....	202
Figure 29. Regular Group Discontinuous Conversion Mode .....	204
Figure 30. High Priority Group Discontinuous Conversion Mode .....	205
Figure 31. High Priority Group Management .....	207
Figure 32. Comparator Block Diagram .....	237
Figure 33. Comparator Voltage Reference Scaler Block Diagram .....	238
Figure 34. Comparator Output Interrupt Structure .....	239
Figure 35. Comparator Output Wake-up Structure.....	239
Figure 36. GPTM Block Diagram.....	246
Figure 37. Up-counting Example .....	248
Figure 38. Down-counting Example.....	249
Figure 39. Center-aligned Counting Example.....	250

Figure 40. GPTM Clock Selection Source .....	252
Figure 41. Trigger Control Block .....	253
Figure 42. Slave Controller Diagram .....	254
Figure 43. GPTM in Restart Mode .....	254
Figure 44. GPTM in Pause Mode .....	255
Figure 45. GPTM in Trigger Mode .....	256
Figure 46. Master GPTM <sub>n</sub> and Slave GPTM <sub>m</sub> /MCTM <sub>m</sub> Connection .....	257
Figure 47. MTO Selection .....	257
Figure 48. Capture/Compare Block Diagram .....	258
Figure 49. Input Capture Mode .....	259
Figure 50. PWM Pulse Width Measurement Example .....	260
Figure 51. Channel 0 and Channel 1 Input Stage .....	261
Figure 52. Channel 2 and Channel 3 Input Stage .....	262
Figure 53. Output Stage Block Diagram .....	262
Figure 54. Toggle Mode Channel Output Reference Signal (CH <sub>x</sub> PRE = 0) .....	263
Figure 55. Toggle Mode Channel Output Reference Signal (CH <sub>x</sub> PRE = 1) .....	264
Figure 56. PWM Mode Channel Output Reference Signal and Counter in Up-counting Mode .....	264
Figure 57. PWM Mode Channel Output Reference Signal and Counter in Down-counting Mode .....	265
Figure 58. PWM Mode Channel Output Reference Signal and Counter in Centre-align Mode .....	265
Figure 59. Update Event Setting Diagram .....	266
Figure 60. Input Stage and Quadrature Decoder Block Diagram .....	267
Figure 61. Both TI0 and TI1 Quadrature Decoder Counting .....	268
Figure 62. GT_ETI Pin Digital Filter Diagram with N = 2 .....	269
Figure 63. Clearing CH <sub>x</sub> OREF by ETIF .....	269
Figure 64. Single Pulse Mode .....	270
Figure 65. Immediate Active Mode Minimum Delay .....	271
Figure 66. Asymmetric PWM Mode versus Center Align Counting Mode .....	272
Figure 67. Pausing GPTM1 using the GPTM0 CH0OREF Signal .....	273
Figure 68. Triggering GPTM1 with GPTM0 Update Event .....	274
Figure 69. Trigger GPTM0 and GPTM1 with the GPTM0 CH0 Input .....	275
Figure 70. GPTM PDMA Mapping Diagram .....	276
Figure 71. BFTM Block Diagram .....	322
Figure 72. BFTM – Repetitive Mode .....	323
Figure 73. BFTM – One Shot Mode .....	324
Figure 74. BFTM – One Shot Mode Counter Updating .....	324
Figure 75. MCTM Block Diagram .....	329
Figure 76. Up-counting Example .....	331
Figure 77. Down-counting Example .....	332
Figure 78. Center-aligned Counting Example .....	333
Figure 79. Update Event Dependent Repetition Mechanism Example .....	334
Figure 80. MCTM Clock Selection Source .....	336

Figure 81. Trigger Control Block .....	337
Figure 82. Slave Controller Diagram .....	338
Figure 83. MCTM in Restart Mode .....	338
Figure 84. MCTM in Pause Mode .....	339
Figure 85. MCTM in Trigger Mode .....	340
Figure 86. Master MCTM <sub>n</sub> and Slave GPTM <sub>m</sub> Connection .....	340
Figure 87. MTO Selection .....	341
Figure 88. Capture/Compare Block Diagram .....	342
Figure 89. Input Capture Mode .....	342
Figure 90. PWM Pulse Width Measurement Example .....	343
Figure 91. Channel 0 and Channel 1 Input Stages .....	344
Figure 92. Channel 2 and Channel 3 Input Stages .....	344
Figure 93. Output Stage Block Diagram .....	345
Figure 94. Toggle Mode Channel Output Reference Signal – CHxPRE = 0 .....	346
Figure 95. Toggle Mode Channel Output Reference Signal – CHxPRE = 1 .....	347
Figure 96. PWM Mode Channel Output Reference Signal and Counter in Up-counting Mode .....	347
Figure 97. PWM Mode Channel Output Reference Signal and Counter in Down-counting Mode .....	348
Figure 98. PWM Mode 1 Channel Output Reference Signal and Counter in Centre-aligned Counting Mode .....	348
Figure 99. Dead-time Insertion Performed for Complementary Outputs .....	349
Figure 100. Break Signal Bolck Diagram for MCTM .....	350
Figure 101. MT_BRK Pin Digital Filter Diagram with N = 2 .....	351
Figure 102. Channel 3 Output with a Break Event Occurrence .....	352
Figure 103. Channel 0 ~ 2 Complementary Outputs with a Break Event Occurrence .....	352
Figure 104. Channel 0 ~ 2 Only One Output Enabled when Fault Event Occurs .....	353
Figure 105. Hardware Protection When Both CHxO and CHxNO Are in Active Condition .....	354
Figure 106. Update Event 1 Setup Diagram .....	356
Figure 107. CHxE, CHxNE and CHxOM Updated by Update Event 2 .....	357
Figure 108. Update Event 2 Setup Diagram .....	357
Figure 109. Input Stage and Quadature Decoder Block Diagram .....	358
Figure 110. Both TI0 and TI1 Quadrature Decoder Counting .....	359
Figure 111. MT_ETI Pin Digital Filter Diagram with N = 2 .....	360
Figure 112. Clearing CHxOREF by ETIF .....	360
Figure 113. Single Pulse Mode .....	361
Figure 114. Immediate Active Mode Minimum Delay .....	362
Figure 115. Asymmetric PWM Mode versus Center-aligned Counting Mode .....	363
Figure 116. Pausing GPTM0 using the MCTM0 CH0OREF Signal .....	364
Figure 117. Triggering GPTM0 with MCTM0 Update Event 1 .....	365
Figure 118. Trigger MCTM0 and GPTM0 with the MCTM0 CH0 Input .....	366
Figure 119. CH1XOR Input as Hall Sensor Interface .....	367
Figure 120. MCTM PDMA Mapping Diagram .....	369

Figure 121. RTC Block Diagram .....	422
Figure 122. Watchdog Timer Block Diagram .....	432
Figure 123. Watchdog Timer Behavior .....	434
Figure 124. I <sup>2</sup> C Module Block Diagram.....	442
Figure 125. START and STOP Condition .....	444
Figure 126. Data Validity.....	444
Figure 127. 7-bit Addressing Mode .....	445
Figure 128. 10-bit Addressing Write Transmit Mode .....	446
Figure 129. 10-bit Addressing Read Receive Mode .....	446
Figure 130. I <sup>2</sup> C Bus Acknowledge .....	447
Figure 131. Clock Synchronization during Arbitration.....	448
Figure 132. Two Master Arbitration Procedure .....	448
Figure 133. Master Transmitter Timing Diagram .....	450
Figure 134. Master Receiver Timing Diagram .....	452
Figure 135. Slave Transmitter Timing Diagram .....	453
Figure 136. Slave Receiver Timing Diagram .....	454
Figure 137. SCL Timing Diagram.....	467
Figure 138. SPI Block Diagram .....	473
Figure 139. SPI Single Byte Transfer Timing Diagram – CPOL = 0, CPHA = 0.....	475
Figure 140. SPI Continuous Data Transfer Timing Diagram – CPOL = 0, CPHA = 0 .....	476
Figure 141. SPI Single Byte Transfer Timing Diagram – CPOL = 0, CPHA = 1.....	476
Figure 142. SPI Continuous Transfer Timing Diagram – CPOL = 0, CPHA = 1.....	477
Figure 143. SPI Single Byte Transfer Timing Diagram – CPOL = 1, CPHA = 0.....	477
Figure 144. SPI Continuous Transfer Timing Diagram – CPOL = 1, CPHA = 0.....	478
Figure 145. SPI Single Byte Transfer Timing Diagram – CPOL = 1, CPHA = 1.....	478
Figure 146. SPI Continuous Transfer Timing Diagram – CPOL = 1, CPHA = 1.....	478
Figure 147. SPI Multi-Master Slave Environment.....	480
Figure 148. USART Block Diagram .....	494
Figure 149. USART Serial Data Format .....	496
Figure 150. USART Clock CK_USART and Data Frame Timing .....	497
Figure 151. USART I/O and IrDA Block Diagram .....	499
Figure 152. IrDA Modulation and Demodulation.....	499
Figure 153. RS485 Interface and Waveform .....	500
Figure 154. USART Synchronous Transmission Example .....	502
Figure 155. 8-bit Format USART Synchronous Waveform.....	503
Figure 156. USART RTS Flow Control .....	504
Figure 157. USART CTS Flow Control .....	504
Figure 158. UART Block Diagram.....	526
Figure 159. UART Serial Data Format.....	527
Figure 160. UART Clock CK_UART and Data Frame Timing .....	528
Figure 161. SCI Block Diagram .....	543

Figure 162. Character Frame and Compensation Mode .....	546
Figure 163. Guard Time Duration .....	547
Figure 164. Character and Block Waiting Time Duration – CWT and BWT.....	548
Figure 165. SCI Card Detection Diagram .....	549
Figure 166. SCI Interrupt Structure.....	551
Figure 167. USB Block Diagram.....	568
Figure 168. Endpoint Buffer Allocation Example.....	570
Figure 169. Double-buffering Operation Example .....	571
Figure 170. PDMA Block Diagram .....	602
Figure 171. PDMA Request Mapping Architecture .....	604
Figure 172. PDMA Channel Arbitration and Scheduling Example .....	605
Figure 173. EBI Block Diagram .....	625
Figure 174. EBI Non-multiplexed 8-bit Data, 8-bit Address Read Operation.....	626
Figure 175. EBI Non-multiplexed 8-bit Data, 8-bit Address Write Operation .....	626
Figure 176. EBI Non-multiplexed 16-bit Data, N-bit Address Read Operation .....	627
Figure 177. EBI Non-multiplexed 16-bit Data, N-bit Address Write Operation.....	627
Figure 178. An EBI Address Latch Setup Diagram.....	628
Figure 179. EBI Multiplexed 16-bit Data, 16-bit Address Read Operation.....	628
Figure 180. EBI Multiplexed 16-bit Data, 16-bit Address Write Operation.....	629
Figure 181. EBI Multiplexed 8-bit Data, 24-bit Address Read Operation.....	629
Figure 182. EBI Multiplexed 8-bit Data, 24-bit Address Write Operation.....	630
Figure 183. EBI Non-multiplexed 8-bit Data, 8-bit Address Mode For Page Read Operation .....	631
Figure 184. EBI Non-multiplexed 16-bit Data, N-bit Address Mode For Page Read Operation .....	631
Figure 185. EBI Multiplexed 16-bit Data, 16-bit Address Mode For Page Read Operation.....	631
Figure 186. EBI Multiplexed 8-bit Data, 24-bit Address Mode For Page Read Operation.....	632
Figure 187. EBI Page Close Example .....	632
Figure 188. EBI Inserts An IDLE Cycle Between Transactions In The Same Bank (NOIDLE = 0).....	633
Figure 189. EBI De-asserts An IDLE Cycle Between Transactions In The Same Bank (NOIDLE=1) ..	634
Figure 190. EBI Bank Memory Map.....	636
Figure 191. I <sup>2</sup> S Block Diagram.....	650
Figure 192. Simple I <sup>2</sup> S Master/Slave Configuration.....	651
Figure 193. I <sup>2</sup> S Clock Generator Diagram .....	652
Figure 194. I <sup>2</sup> S-justified Stereo Mode Waveforms .....	654
Figure 195. I <sup>2</sup> S-justified Stereo Mode Waveforms (32-bit Channel Enabled).....	654
Figure 196. Left-justified Stereo Mode Waveforms.....	655
Figure 197. Left-justified Stereo Mode Waveforms (32-bit Channel Enabled).....	655
Figure 198. Right-justified Stereo Mode Waveforms .....	656
Figure 199. Right-justified Stereo Mode Waveforms (32-bit Channel Enabled) .....	656
Figure 200. I <sup>2</sup> S-justified Mono Mode Waveforms.....	657
Figure 201. I <sup>2</sup> S-justified Mono Mode Waveforms (32-bit Channel Enabled).....	657
Figure 202. Left-justified Mono Mode Waveforms .....	658

Figure 203. Left-justified Mono Mode Waveforms (32-bit Channel Enabled) .....	658
Figure 204. Right-justified Mono Mode Waveforms .....	659
Figure 205. Right-justified Mono Mode Waveforms (32-bit Channel Enabled).....	659
Figure 206. I <sup>2</sup> S-justified Repeat Mode Waveforms .....	660
Figure 207. I <sup>2</sup> S-justified Repeat Mode Waveforms (32-bit Channel Enabled).....	660
Figure 208. FIFO Data Content Arrangement for Various Modes.....	661
Figure 209. CRC Block Diagram .....	672
Figure 210. CRC Data Bit and Byte Reversal Example.....	674

# 1 Introduction

## Overview

This user manual provides detailed information including how to use the HT32F1653/1654 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 HT32F1653/1654 datasheet.

The Holtek HT32F1653/1654 devices are high performance and low power consumption 32-bit microcontrollers based around an ARM® Cortex™-M3 processor core. The Cortex™-M3 is a next-generation processor core which is tightly coupled with Nested Vectored Interrupt Controller (NVIC), SysTick timer, and including advanced debug support.

The HT32F1653/1654 devices operate at a frequency of up to 72 MHz with a Flash accelerator to obtain maximum efficiency. It provides up to 64 KB of embedded Flash memory for code/data storage and up to 16 KB of embedded SRAM memory for system operation and application program usage. A variety of peripherals, such as ADC, Comparator, I<sup>2</sup>C, USART, UART, SPI, I<sup>2</sup>S, PDMA, GPTM, MCTM, SCI, EBI, CRC-16/32, 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 HT32F1653/1654 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™-M3 processor core
  - Up to 72 MHz operation frequency
  - Single-cycle multiplication and hardware division
  - Built-in Nested Vectored Interrupt Controller (NVIC)
  - 24-bit SysTick timer
- On-chip memory
  - 64 Kbytes on-chip Flash memory for instruction/data and options storage
  - Up to 16 Kbytes on-chip SRAM
  - Flexible booting mode supports bootloader
- Flash Memory Controller – FMC
  - Flash accelerator to obtain maximum efficiency
  - 32-bit word programming with In System Programming Interface (ISP)
  - Fast parallel writer programming interface for external programming tool
  - Flash protection capability to prevent illegal access
- Reset and Clock Control Units – RSTCU, CKCU
  - Supply supervisor: Power On Reset / Power Down Reset (POR/PDR) and Programmable Low Voltage Detector (LVD)
  - External 4 to 16 MHz crystal oscillator
  - External 32,768 Hz crystal oscillator
  - Internal 8 MHz RC oscillator trimmed to ±2% accuracy at 3.3 V operating voltage and 25°C operating temperature
  - Internal 32 kHz RC oscillator
  - Integrated system clock PLL
  - Independent clock gating bits for peripheral clock sources
- Power management – PWRCU
  - Single 3.3 V power supply: 2.7 V to 3.6 V
  - Integrated 1.8 V LDO regulator for core and peripheral power supply
  - VBAT battery power supply for RTC and backup registers
  - Three power domains: 3.3 V, 1.8 V and Backup
  - Four power saving modes: Sleep, Deep-Sleep1, Deep-Sleep2, Power-Down
- External Interrupt/Event Controller – EXTI
  - Up to 16 EXTI lines with configurable trigger source and type
  - All GPIO pins can be selected as EXTI trigger source
  - Source trigger type includes high level, low level, negative edge, positive edge, or both edge
  - Individual interrupt enable, wakeup enable and status bits for each EXTI line
  - Software interrupt trigger mode for each EXTI line
  - Integrated deglitch filter for short pulse blocking

- Analog to Digital Converter – ADC
  - 12-bit SAR ADC engine
  - Up to 1 MSPS conversion rate – 1  $\mu$ s at 56 MHz, 1.17  $\mu$ s at 72 MHz
  - Up to 16 external analog input channels
  - External reference voltage input possibility
  - Conversion range:  $V_{REF+} \sim V_{REF-}$
  - 2.5 V ~ 3.6 V conversion range
- Analog Comparator – CMP
  - Two rail-to-rail comparator
  - Each comparator has configurable negative inputs used for flexible voltage selection
  - Dedicated I/O pin or internal voltage reference provided by 6-bit scaler.
  - Programmable hysteresis
  - Programming speed and consumption
  - Comparator output can be output to I/O or to timers or ADC trigger inputs
  - 6-bit scaler can be configurable to dedicated I/O for voltage reference.
  - Comparator has interrupt generation capability with wakeup from Sleep or Deep Sleep modes through the EXTI controller.
  - Supply voltage range: 2.7 V ~ 3.6 V
- IO ports
  - Up to 51 GPIOs
  - Port A ~ D are mapped on 16 external interrupts (EXTI)
  - Almost I/O pins are configurable output driving current
- PWM Generation and Capture Timers – GPTM
  - One 16-bit up, down, up/down auto-reload counters
  - 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
- Motor Control Timer – MCTM
  - One 16-bit up, down, up/down auto-reload counters
  - 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

- Basic Function Timer – BFTM
  - Two 32-bit compare/match count-up counters – no I/O control features
  - One shot mode – counting stops after a match condition
  - Repetitive mode – restart counter after a match condition
- Watchdog Timer – WDT
  - 12-bit down counter with 3-bit prescaler
  - Interrupt or reset event for the system
  - Programmable watchdog timer window function
  - Registers write protection function
- Real Time Clock – RTC
  - 32-bit up-counter with a programmable prescaler
  - Alarm function
  - Interrupt and Wake-up event
- Inter-integrated Circuit – 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: 16 × 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
  - FIFO Depth: 16 × 9 bits for both receiver and transmitter

- Smart Card Interface – SCI
  - Supports ISO 7816-3 Standard
  - Character mode
  - Single transmit buffer and single receive buffer
  - 11-bit ETU (elementary time unit) counter
  - 9-bit guard time counter
  - 24-bit general purpose waiting time counter
  - Parity generation and checking
  - Automatic character retry on parity error detection in transmission and reception modes
- Inter-IC Sound – I<sup>2</sup>S
  - Master or slave mode
  - Mono and stereo
  - I<sup>2</sup>S-justified, Left-justified, and Right-justified mode
  - 8/16/24/32-bit sample size with 32-bit channel extended
  - 8 × 32-bit Tx & Rx FIFO with PDMA supported
  - 8-bit Fractional Clock Divider with rate control
- Cyclic Redundancy Check – CRC
  - Support CRC16 polynomial: 0x8005,  $X^{16} + X^{15} + X^2 + 1$
  - Support CCITT CRC16 polynomial: 0x1021,  $X^{16} + X^{12} + X^5 + 1$
  - Support IEEE-802.3 CRC32 polynomial: 0x04C11DB7,  $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
  - Support 1's complement, byte reverse & bit reverse operation on data and checksum
  - Support byte, half-word & word data size
  - Programmable CRC initial seed value
  - CRC computation done in 1 AHB clock cycle for 8-bit data and 4 AHB clock cycles for 32-bit data
  - Support PDMA to complete a CRC computation of a block of memory
- Peripheral Direct Memory Access – PDMA
  - 8 channels with trigger source grouping
  - 8/16/32-bit width data transfer
  - Supports Address increment, decrement or fixed mode
  - 4-level programmable channel priority
  - Auto reload mode
  - Supported trigger source:  
ADC, SPI, EBI, CRC, USART, UART, I<sup>2</sup>C, I<sup>2</sup>S, GPTM, MCTM, SCI and software request
- External Bus Interface – EBI
  - Programmable interface for various memory types
  - Individual chip select signal for per memory bank
  - Accurate control of setup, strobe, hold and turn-around timing per memory bank
  - Support page mode read
  - Both multiplexed and non-multiplexed address and data line configurations
  - Up to 21 address lines
  - Up to 16-bit data bus width
  - Automatic translation when AHB transaction width and external memory interface width is different
  - Write buffer to decrease the stalling of the AHB write burst transaction

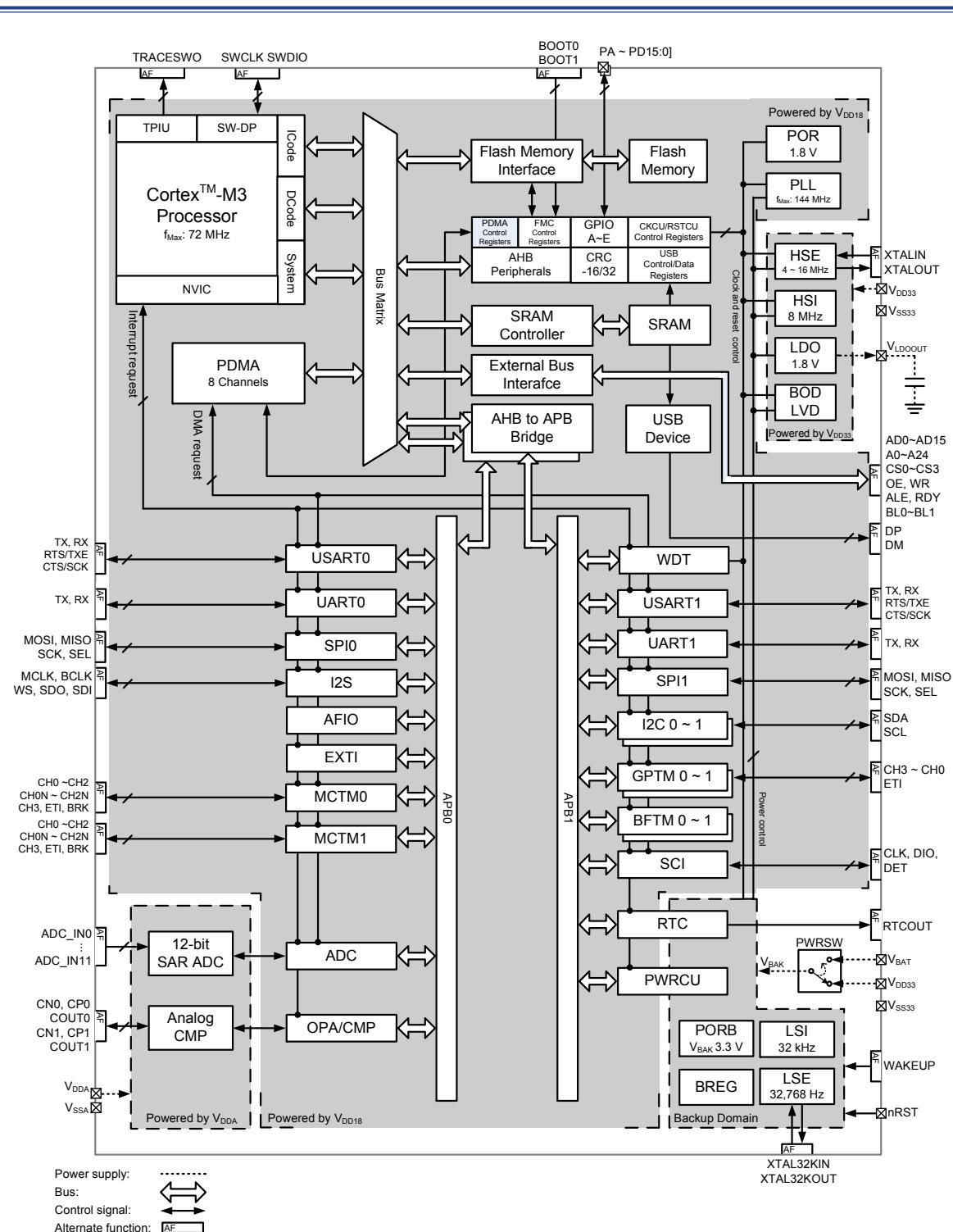
- Universal Serial Bus Device Controller – USB
  - Complies with USB 2.0 full-speed (12Mbps) specification
  - On-chip USB full-speed transceiver
  - 1 control endpoint (EP0) for control transfer
  - 3 single-buffered endpoint (EP1~EP3) for bulk and interrupt transfer
  - 4 double-buffered endpoint (EP4~EP7) for bulk, interrupt and isochronous transfer
  - 1KB EP-SRAM used as the endpoint data buffers
- Debug support
  - Serial Wire Debug Port (SW-DP)
  - 6 instruction comparator and 2 literal comparator for hardware breakpoint/watchpoint or code patch
  - 1-bit asynchronous trace (TRACESWO)
- Package and Operation Temperature
  - 48/64-pin LQFP package
  - Operation temperature range: -40 °C to +85 °C

## Device Information

Table 1. HT32F1653/1654 Series Features and Peripheral List

	Peripherals	HT32F1654	HT32F1653
Main Flash (KB)	63	32	
Option Bytes Flash	1	1	
SRAM (KB)	16	8	
Timers	MCTM	2	
	GPTM	2	
	BFTM	2	
	RTC	1	
	WDT	1	
Communication	CSIF	1	
	USB	1	
	SCI	1	
	USART	2	
	UART	2	
	SPI	2	
	I <sup>2</sup> C	2	
	I <sup>2</sup> S	1	
	EBI	1	
	CRC	1	
GPIO	Up to 49		
EXTI	16		
12-bit ADC	1		
Number of channels	Max. 16 Channels		
OPA/Comparator	2		
CPU frequency	Up to 72 MHz		
Operating voltage	2.7 V ~ 3.6 V		
Operating temperature	-40 °C ~ +85 °C		
Package	48/64LQFP		

## Block Diagram



Introduction

Figure 1. HT32F1653/1654 Block Diagram

## 2 Document Conventions

Unless otherwise specified, this document uses the conventions which showed as follows.

**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="margin-left: auto; margin-right: auto;"> <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</td> <td>RW</td> </tr> <tr> <td>0</td> <td>0</td> </tr> </table>	19	18	SERDYIE	PLLRDYIE	RW	RW	0	0	Software can read and write to this bit.
19	18									
SERDYIE	PLLRDYIE									
RW	RW									
0	0									
RO	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">3</td> <td style="text-align: center;">2</td> </tr> <tr> <td>HSIRDY</td> <td>HSERDY</td> </tr> <tr> <td>RO</td> <td>RO</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </table>	3	2	HSIRDY	HSERDY	RO	RO	1	0	Software can only read this bit. Write operation has no effort.
3	2									
HSIRDY	HSERDY									
RO	RO									
1	0									
RC	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> </tr> <tr> <td>PDF</td> <td>BAK_PORF</td> </tr> <tr> <td>RC</td> <td>RC</td> </tr> <tr> <td>0</td> <td>1</td> </tr> </table>	1	0	PDF	BAK_PORF	RC	RC	0	1	Software can read this bit. Read operation clears it to 0 automatically.
1	0									
PDF	BAK_PORF									
RC	RC									
0	1									
WC	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">3</td> <td style="text-align: center;">2</td> </tr> <tr> <td>SERDYF</td> <td>PLLRDYF</td> </tr> <tr> <td>WC</td> <td>WC</td> </tr> <tr> <td>0</td> <td>0</td> </tr> </table>	3	2	SERDYF	PLLRDYF	WC	WC	0	0	Software can read this bit or clear it by writing 1. Write 0 to it has no effort.
3	2									
SERDYF	PLLRDYF									
WC	WC									
0	0									
WO	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">31</td> <td style="text-align: center;">30</td> </tr> <tr> <td></td> <td>DB_CKSRC</td> </tr> <tr> <td>WO</td> <td>WO</td> </tr> <tr> <td>0</td> <td>0</td> </tr> </table>	31	30		DB_CKSRC	WO	WO	0	0	Software can only write to this bit. Read operation always returns 0.
31	30									
	DB_CKSRC									
WO	WO									
0	0									
Reserved	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> </tr> <tr> <td>LLRDY</td> <td>Reserved</td> </tr> <tr> <td>RO</td> <td>0</td> </tr> </table>	1	0	LLRDY	Reserved	RO	0	Reserved bit(s) for future use. Software should not rely on the value of the reserved bit. In general case, reserved bits are set to 0. Note that reserved bit must be kept at reset value.		
1	0									
LLRDY	Reserved									
RO	0									
Word		Data length of word is 32-bit.								
Half-word		Data length of half-word is 16-bit.								
Byte		Data length of byte is 8-bit.								

## 3 System Architecture

The system architecture of the HT32F1653/1654 series of devices that includes the ARM® Cortex™-M3 processor, bus architecture and memory organization will be described in the following sections. The Cortex™-M3 is a next generation processor core which offers many new features. Integrated and advanced features make the Cortex™-M3 processor suitable for high performance and low power microcontroller market. In brief, Cortex™-M3 processor includes three AHB-Lite buses, ICode, DCode, and System bus. All memory accesses of Cortex™-M3 are performed on those three buses according to the different purpose and target memory space. The memory organization with Harvard architecture, pre-defined memory map, and up to 4 Gbytes memory space makes the system flexible and extendable.

### ARM® Cortex™-M3 Processor

Cortex™-M3 is a general purpose 32-bit processor core which very suitable for high performance and low power microcontroller market. It offers many new features including a Thumb-2 instruction sets, hardware divide, low latency interrupt respond time, atomic bit-banding access, and multiple buses for simultaneous accesses. Cortex™-M3 is based on ARMv7 architecture and supports both Thumb and Thumb-2 instruction sets. Some system peripherals are also provided by the Cortex™-M3 including:

- Internal Bus Matrix connected with ICode bus, DCode bus, System bus, Private Peripheral Bus (PPB), and debug accesses (AHB-AP)
- Nested Vectored Interrupt Controller (NVIC)
- Flash Patch and Breakpoint (FPB)
- Data Watchpoint and Trace (DWT)
- Instrument Trace Macrocell (ITM)
- Memory Protection Unit (MPU)
- Serial Wire JTAG Debug Port (SWJ-DP)
- Embedded Trace Macrocell (ETM)
- Trace Port Interface Unit (TPIU)

The following figure shows the block diagram of Cortex™-M3. For more information, please refer to ARM® Cortex™-M3 Technical Reference Manual.

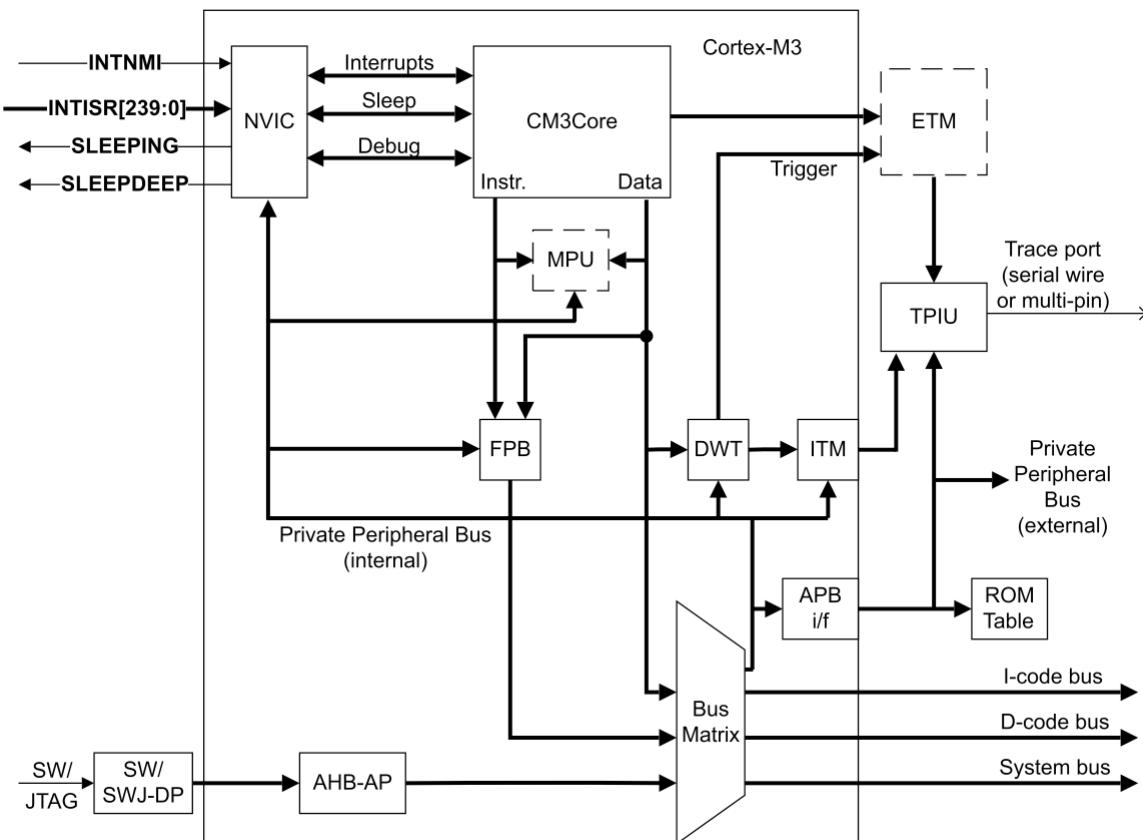
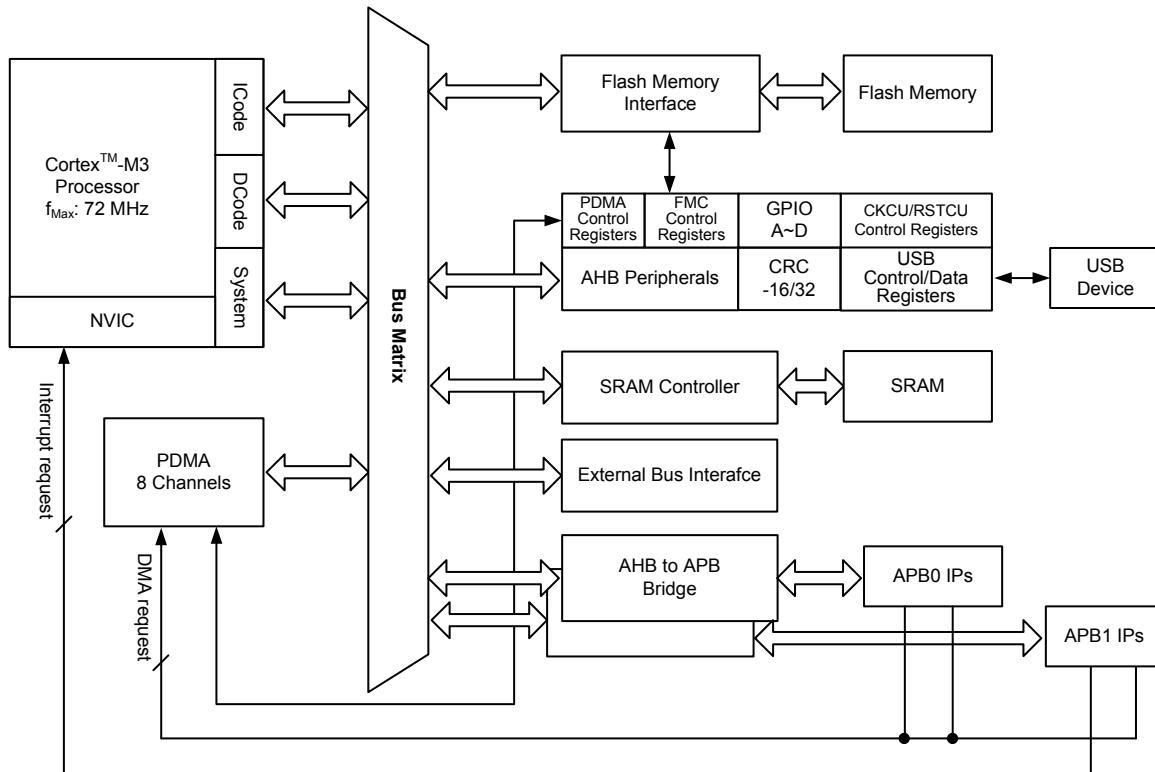


Figure 2. Cortex™-M3 Block Diagram

## Bus Architecture

HT32F1653/1654 consists of four master and six slaves in the bus architecture. Cortex™-M3 ICode, DCode, System bus, and Peripheral Direct Memory Access (PDMA) are the masters, internal SRAM, internal Flash memory, AHB peripherals, external bus interface and two AHB to APB bridges are the slaves. ICode bus is for instruction and vector fetches from Code region (0x0000\_0000 ~ 0x1FFF\_FFFF) to Cortex™-M3 core. DCode bus is for data load/stores and debugging accesses of Code region. In similar, System bus is for instruction/vector fetches, data load/stores, and debugging accesses of system regions. System regions include internal SRAM region and Peripheral region. All of these buses are based on 32-bit Advanced High-performance Bus-Lite (AHB-Lite) protocol. Following figure shows the bus architecture of HT32F1653/1654.



**Figure 3. HT32F1653/1654 Bus Architecture**

## Memory Organization

ARM® Cortex™-M3 is in Harvard architecture which can use separate buses to fetch instructions and load/store data. The instruction and data bus share the same memory address space. The maximum addressing range of Cortex™-M3 is 4 GB since it has 32-bit bus address width. Besides, a pre-define memory map is provided by Cortex™-M3 to reduce the software complexity between different implementation by chip vendors. However, some regions are used by ARM® Cortex™-M3 system peripherals. Please refer to ARM® Cortex™-M3 Technical Reference Manual for more information. Following figure shows the memory map of HT32F1653/1654 including Code, SRAM, peripheral, and other pre-define regions.

## Memory Map

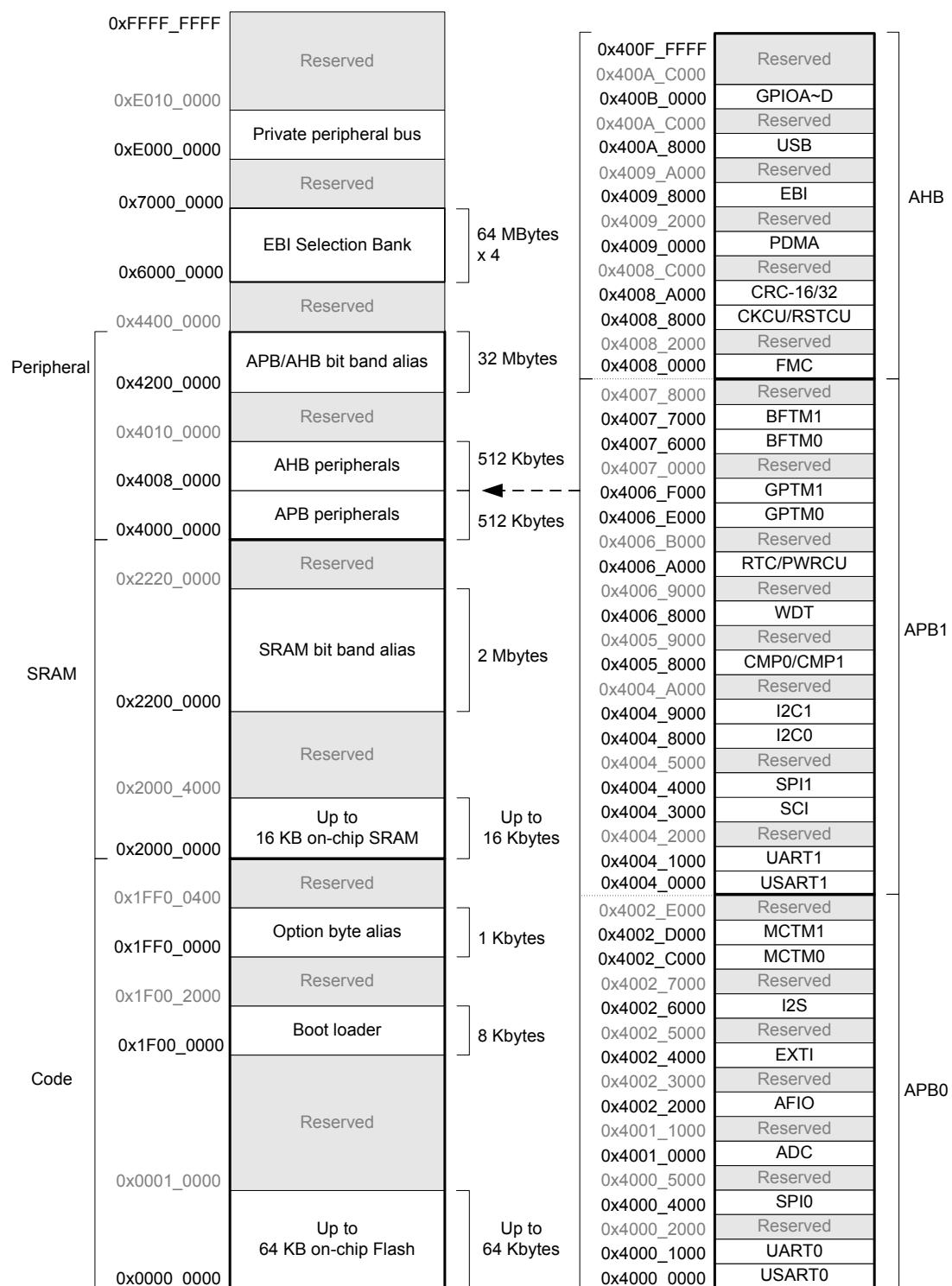


Figure 4. HT32F1653/1654 Memory Map

**Table 3. HT32F1653/1654 Register Map**

Start Address	End Address	Peripheral	Bus	Register map
0x4000_0000	0x4000_0FFF	USART0	APB0	
0x4000_1000	0x4000_1FFF	UART0		
0x4000_2000	0x4000_3FFF	Reserved		
0x4000_4000	0x4000_4FFF	SPI0		
0x4000_5000	0x4000_FFFF	Reserved		
0x4001_0000	0x4001_0FFF	ADC		
0x4001_1000	0x4002_1FFF	Reserved		
0x4002_2000	0x4002_2FFF	AFIO		
0x4002_3000	0x4002_3FFF	Reserved		
0x4002_4000	0x4002_4FFF	EXTI		
0x4002_5000	0x4002_5FFF	Reserved		
0x4002_6000	0x4002_6FFF	I2S		
0x4002_7000	0x4002_BFFF	Reserved		
0x4002_C000	0x4002_CFFF	MCTM0		
0x4002_D000	0x4002_DFFF	MCTM1		
0x4002_E000	0x4003_FFFF	Reserved		
0x4004_0000	0x4004_0FFF	USART1		
0x4004_1000	0x4004_1FFF	UART1		
0x4004_2000	0x4004_2FFF	Reserved		
0x4004_3000	0x4004_3FFF	SCI		
0x4004_4000	0x4004_4FFF	SPI1		
0x4004_5000	0x4004_7FFF	Reserved		
0x4004_8000	0x4004_8FFF	I <sup>2</sup> C0		
0x4004_9000	0x4004_9FFF	I <sup>2</sup> C1		
0x4004_A000	0x4005_7FFF	Reserved		
0x4005_8000	0x4005_8FFF	Comparator	APB1	
0x4005_9000	0x4006_7FFF	Reserved		
0x4006_8000	0x4006_8FFF	WDT		
0x4006_9000	0x4006_9FFF	Reserved		
0x4006_A000	0x4006_AFFF	RTC/PWRCU		
0x4006_B000	0x4006_DFFF	Reserved		
0x4006_E000	0x4006_EFFF	GPTM0		
0x4006_F000	0x4006_FFFF	GPTM1		
0x4007_0000	0x4007_5FFF	Reserved		
0x4007_6000	0x4007_6FFF	BFTM0		
0x4007_7000	0x4007_7FFF	BFTM1		
0x4007_8000	0x4007_FFFF	Reserved		

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

## Embedded Flash Memory

HT32F1653/1654 provides up to 64 Kbytes on-chip Flash memory which is located at address 0x0000\_0000. It supports bytes, halt-words, and word access. Note that Flash memory only supported read operation for Cortex™-M3 ICode or DCode bus access. Any write operation on Flash memory (via DCode) causes bus fault exception occurred. The Flash memory is consisted of 64 pages. Each 1 KB page can be erased independently. A 32-bit programming interface provides the capability to change bits from 1 to 0. The data storage or firmware upgrade can be performed by several ways such as In System Programming (ISP), In Application Programming (IAP), In Circuit Programming (ICP), or parallel writer programming. The above programming methods provide flexibility to user for data storage and firmware upgrade purpose. For more information please refer to the Flash Memory Controller section.

## Embedded SRAM Memory

HT32F1653/1654 contains up to 16 KB on-chip SRAM which is located at address 0x2000\_0000. It supports bytes, halt-words (16 bits), and word access (32 bits). The read and write operations are performed at CPU speed with 0 wait state. In order to reduce the time spent of read-modify-write operations, Cortex™-M3 provides bit-banding function to perform single atomic bit operation. User can modify a single bit in SRAM bit-band region by accessing the corresponding bit-band alias. For more information about bit-binding, please reference to the ARM® Cortex™-M3 Technical Reference Manual. The following formulas and examples show how to access a bit in the bit-band region by calculate the bit-band alias.

$$\text{Bit-band alias} = \text{Bit-band base} + (\text{byte offset} * 32) + (\text{bit number} * 4)$$

For example, if you want to access bit 7 of address 0x2000\_0200, the bit-band alias is:

$$\text{Bit-band alias} = 0x2200\_0000 + (0x200 * 32) + (7 * 4) = 0x2200\_401C$$

Write to address 0x2200\_401C causes the bit 7 of address 0x2000\_0200 changed. On the contrary, read address 0x2200\_401C returns 0x01 or 0x00 according to the value of bit 7 at SRAM address 0x2000\_0200.

## AHB Peripherals

The AHB peripherals address range is from 0x4008\_0000 to 0x400F\_FFFF. Some peripherals are connected to the AHB bus directly such as Clock Control Unit, Reset Control Unit, and Flash Memory Controller. The clocks of AHB peripherals are always enabled after system reset. Access to registers for these peripherals can be achieved directly via the AHB bus. Note that all peripheral registers in AHB bus only supported word access.

## APB Peripherals

The APB peripherals address range is from 0x4000\_0000 to 0x4007\_FFFF. An APB to AHB bridge provides the access capability between the Cortex™-M3 and APB peripherals. Moreover, the clocks of each APB peripherals are disabled after system reset. Software must enable peripheral clock by setting APBCCR register in Clock Control Unit before access peripheral register. Note that the APB to AHB bridge duplicates the half-word or byte data to word width when a half-word or byte access is performed on APB peripheral register. In other words, the access result of half-word or byte access on APB peripheral register is various depending on the peripheral register definition.

# 4

# Flash Memory Controller (FMC)

## Introduction

The Flash Memory Controller, FMC, provides all the necessary functions, pre-fetch buffer and branch cache for the embedded on-chip Flash Memory. Since the access speed of the Flash Memory is slower than the CPU, a wide access interface with a pre-fetch buffer is provided for the Flash Memory in order to reduce the CPU waiting time which will cause CPU instruction execution delays. Flash Memory word program/page erase functions are also provided.

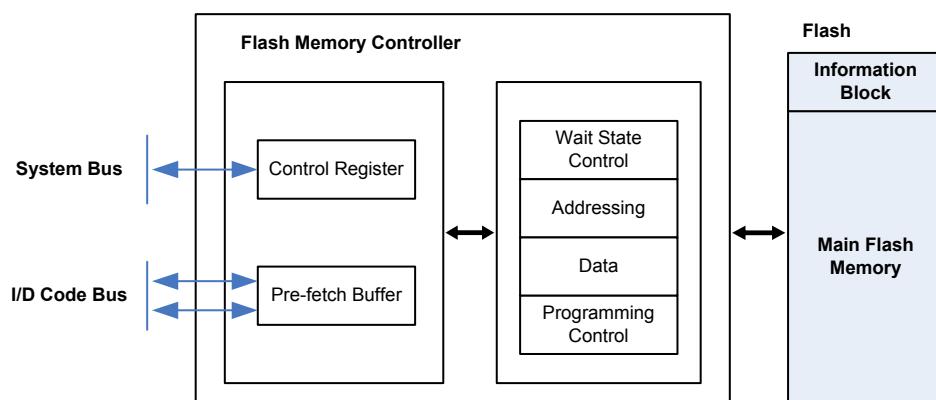


Figure 5. Flash Memory Controller Block Diagram

## Features

- Up to 64 KB on-chip Flash memory for storing instruction/data and options
  - 64 KB: 63 KB + 1 KB (instruction/data + option byte)
  - 32 KB: 32 KB + 1 KB (instruction/data + option byte)
- Page size of 1 KB
- Wide access interface with pre-fetch buffer and branch cache to reduce instruction gaps
- Page erase and mass erase capability
- 32-bit word programming
- Interrupt function to indicate end of Flash memory operations or an error occurs
- Flash read protection to prevent illegal code/data access
- Page erase/program protection to prevent unexpected operation

## Function Descriptions

### Flash Memory Map

The following figure is the Flash memory map of the system in which the address ranges from 0x0000\_0000 to 0x1FFF\_FFFF (0.5 GB). The address from 0x1F00\_0000 to 0x1F00\_1FFF is mapped to the Boot Loader with a capacity of 8 KB. Additionally, the region addressed from 0x1FF0\_0000 to 0x1FF0\_03FF is the Option Byte with a capacity of 1 KB. The memory mapping on system view is shown as below.

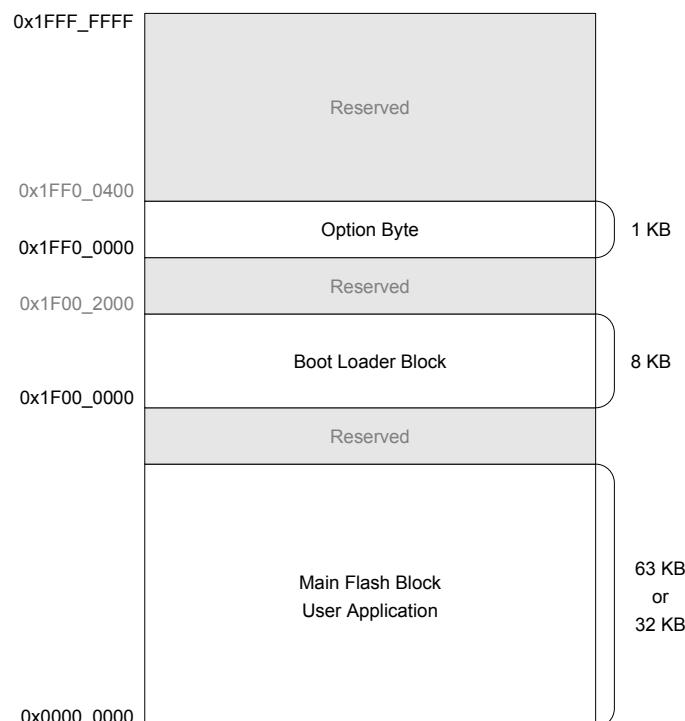


Figure 6. Flash Memory Map

## Flash Memory Architecture

The Flash memory consists of up to 64 KB main Flash with 1 KB capacity per page and a 8 KB Information Block for the Boot Loader. The main Flash memory contains a total of 64 pages (or 33 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. Flash Memory and Option Byte**

Block	Name	Address	Page Protection Bit	Size
Main Flash	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 61	0x0000_F400 ~ 0x0000_F7FF	OB_PP [61]	1 KB
	Page 62	0x0000_F800 ~ 0x0000_FBFF	OB_PP [62]	1 KB
	Option Byte	0x1FF0_0000 ~ 0x1FF0_03FF	OB_CP [1]	1 KB
	Information Block	Boot Loader	0x1F00_0000 ~ 0x1F00_1FFF	NA
				8 KB

**NOTE:** The Information Block stores the boot loader - this block can not be programmed or erased by user.

## Wait State Setting

When the CPU clock, HCLK, is greater than access speed of the Flash memory, then 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 the WAIT [2:0] bits of the Flash Cache and Pre-fetch Control Register, CFCR. In order to match the wait state requirement, the following two rules should be considered.

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

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

**Table 5. Relationship between wait state cycle and HCLK**

Wait State Cycle	HCLK
0	1 MHz < HCLK ≤ 24 MHz
1	24 MHz < HCLK ≤ 48 MHz
2	48 MHz < HCLK ≤ 72 MHz

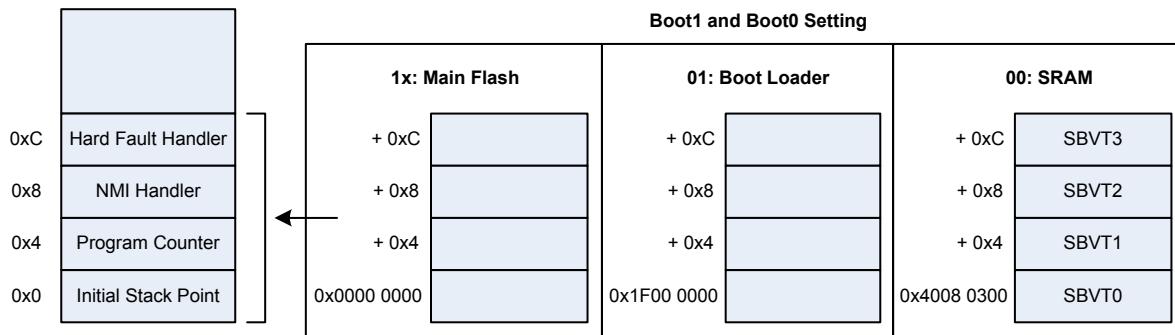
## Booting Configuration

The system provides three kinds of boot modes which can be selected using the BOOT0 and BOOT1 pins. The BOOT0 and BOOT1 pins are sampled during a power-on reset or a system reset. Once the logic value on these pins has been determined, the first 4 words of vector will be remapped to the corresponding source according to the boot modes. The boot modes are shown in the following table.

**Table 6. Boot Modes**

Boot modes selection pins		Mode	Descriptions
BOOT1	BOOT0		
0	0	SRAM	Vector source is SBVT0 ~ 3
0	1	Boot Loader	Vector source is Boot Loader
1	X	Main Flash	Vector source is main Flash memory

The Vector Mapping Control Register, VMCR, is provided to change the vector remapping setting temporarily after a device reset. The initial reset value of the VMCR register is determined by the BOOT0 and BOOT1 pins which will be sampled during the reset duration.



**Figure 7. Vector Remapping**

## Page Erase

The FMC provides a page erase function which is used to initialize the contents of a Flash memory page to a high state. Each page can be erased independently without affecting the contents of other pages. The following steps show the access sequence of the register for a page erase operation.

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

Note that a correct target page address must be confirmed. The software may run out of control if the target erase page is being used to fetch codes or to access data. The FMC will not provide any notification when this occurs. Additionally, the page erase operation will be ignored on protected pages. A Flash operation error interrupt will be triggered by the FMC if the OREIEN bit in the OIER register is set. The software can check the PPEF bit in the OISR register to detect this condition in the interrupt handler. The following figure shows the page erase operation flow.

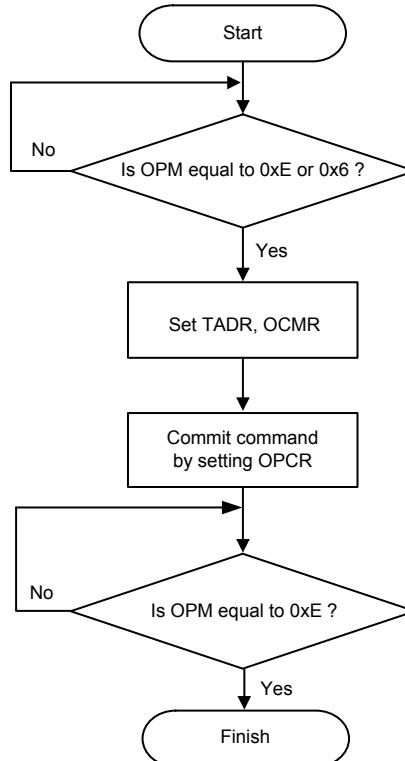


Figure 8. Page Erase Operation Flowchart

## Mass Erase

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

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

Since all Flash contents will be reset to the value of 0xFFFF\_FFFF, the mass erase operation can be implemented by an application that runs in SRAM or by the debug tool that accesses the FMC registers directly. An application executes on the Flash memory shall not trigger a mass erase operation. The following figure shows the mass erase operation flow.

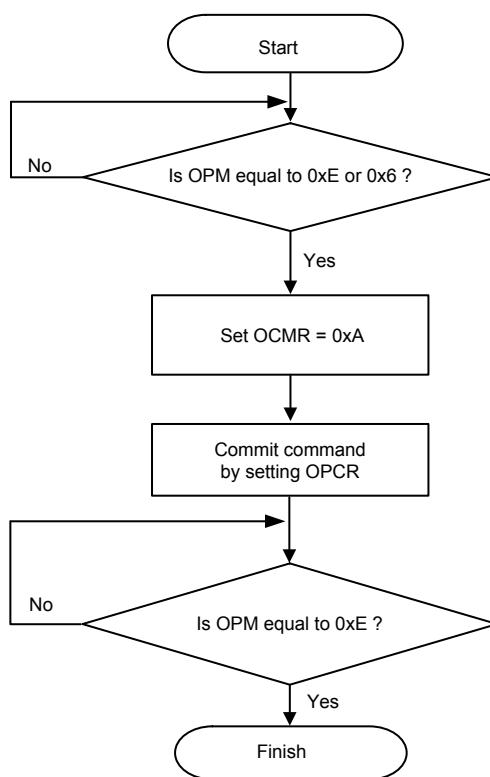


Figure 9. Mass Erase Operation Flowchart

## Word Programming

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

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

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

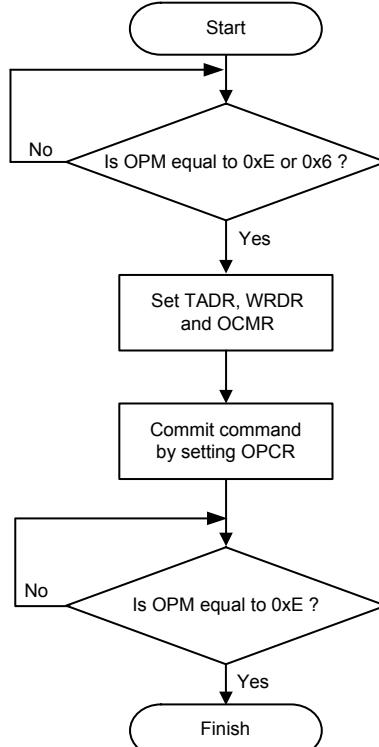


Figure 10. Word Programming Operation Flowchart

## Option Byte Description

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

**Table 7. Option Byte Memory Map**

Option Byte	Offset	Description	Reset Value
<b>Option Byte Base Address = 0x1FF0_0000</b>			
OB_PP	0x000 0x004 0x008 0x00C	OB_PP [n]: Main Flash Page Erase/Program Protection (n = 0 ~ 62 for page 0 ~ page 62) 0: Main Flash Page n Erase/Program Protection is enabled 1: Main Flash Page n Erase/Program Protection is disabled OB_PP [127:63]: Reserved	0xFFFF_FFFF 0xFFFF_FFFF 0xFFFF_FFFF 0xFFFF_FFFF
OB_CP	0x010	OB_CP [0]: Flash Security Protection 0: Flash Security protection is enabled 1: Flash Security protection is disabled OB_CP [1]: Option Byte Erase/Program Protection 0: Option Byte protection is enabled 1: Option Byte protection is disabled OB_CP [31:2]: Reserved	0xFFFF_FFFF
OB_CK	0x020	OB_CK [31:0]: Option Byte Checksum OB_CK should be set as the sum of the 5 words Option Byte content, of which the address offset ranges from 0x000 to 0x010 (0x000 + 0x004 + 0x008 + 0x00C + 0x010), when the content of the OB_PP or OB_CP register is not equal to 0xFFFF_FFFF. Otherwise, both pages erase/program protection and security protection will be enabled.	0xFFFF_FFFF

## Page Erase/Program Protection

The FMC provides page erase/program protection function to prevent inadvertent operation of the Flash memory. The page erase or word programming command will not be accepted by the FMC on protected pages. When the page erase or word programming command is sent to the FMC on a protected page, the PPEF bit in the OISR register will then set by the FMC. If the OREIEN bit in the OIER register is also set to 1 then the Flash operation error interrupt will be triggered by the FMC. The page protection function can be individually enabled for each page by configuring the OB\_PP [62:0] bit field in the Option Byte. If a page erase operation is executed on the Option Byte region, all the Flash memory page protection functions will be disabled. The page protection function of the Option Byte is enabled by clearing the OB\_CP [1] bit to 0. Once the Option Byte has been protected, the only way to disable its protection function is to execute a mass erase operation. The following table shows the access permission of the main Flash page when the page protection is enabled.

**Table 8. Access Permission of Protected Main Flash Page**

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

**NOTE:** The write protection is based on specific pages. The above access permission only affects the pages that protection function has been enabled. Other pages are not affected.

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

- Check the OPCR register to confirm that no Flash memory operation is in progress (OPM [3:0] equal to 0xE or 0x6). Otherwise, wait until the previous operation has been finished.
- Write the OB\_PP address into the TADR register (TADR=0x1FF0\_0000~0x1FF0\_000C).
- Write the data which indicates the protection function of the corresponding page is enabled or disabled into the WRDR register (0: Enabled, 1: Disabled).
- Write the word programming command into the OCMR register (CMD [3:0] = 0x4).
- Send the word programming command to the FMC by setting the OPCR register (set OPM [3:0] = 0xA).
- Wait until all operations have been finished by checking the value of the OPCR register (OPM [3:0] equal to 0xE).
- Read and verify the OB\_PP if required using DCODE access.
- Before to activate the new OB\_PP setting, the OB\_CK must be updated according to the Option Byte checksum rule.
- Apply a system reset to active the new setting.

## Security Protection

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

**Table 9. Access Permission When Security Protection is Enabled**

Mode Operation	User application <sup>(Note 1)</sup>	ICP/Debug mode	Boot from SRAM
DCODE Read	O	X <sup>(1)</sup>	X <sup>(1)</sup>
Program	O <sup>(1)</sup>	X	X
Page Erase	O <sup>(1)</sup>	X	X
Mass Erase	O	O	O

- NOTES:**
1. User application means the software that is executed or booted from the main Flash memory with the JTAG/SW debugger disconnected. However the Option Byte and the page 0 are still protected in which programming and Page Erase operations can not be executed.
  2. When the JTAG/SW debugger is connected or the executing application is booted from SRAM, all the DCODE read operation to all Flash memory regions will return with the value of 0.

The following steps show the security protection procedure register access sequence:

- Check the OPCR register to confirm that no Flash memory operation is in progress (OPM [3:0] equal to 0xE or 0x6). Otherwise, wait until the previous operation has been finished.
- Write the OB\_CP address into the TADR register (TADR = 0x1FF0\_0010).
- Write the data into the WRDR register to clear the OB\_CP [0] bit to 0.
- Write the word programming command into the OCMR register (CMD [3:0] = 0x4).
- Send the word programming command to the FMC by setting the OPCR register (set OPM = 0xA).
- Wait until all operations have been finished by checking the value of the OPCR register (OPM [3:0] equal to 0xE).
- Read and verify the OB\_CP if required using DCODE access.
- Before to activate the security protection function, the OB\_CK must be updated according to the Option Byte checksum rule.
- Apply a system reset to active the new setting.

## Register Map

The following table shows the FMC registers and reset values.

**Table 10. 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	0xXXXX_XXXX
	0x024		0xXXXX_XXXX
	0x028		0xXXXX_XXXX
	0x02C		0xXXXX_XXXX
CPSR	0x030	Flash Security Protection Status Register	0xXXXX_XXXX
VMCR	0x100	Flash Vector Mapping Control Register	0x0000_000X
CFCR	0x200	Flash Cache and Pre-fetch Control Register	0x0000_13D1
SBVT0	0x300	SRAM Booting Vector 0 (Stack Point)	0x2000_X000
SBVT1	0x304	SRAM Booting Vector 1 (Program Counter)	0x2000_0155
SBVT2	0x308	SRAM Booting Vector 2 (NMI Handler)	0x0000_0000
SBVT3	0x30C	SRAM Booting Vector 3 (Hard Fault Handler)	0x0000_0000

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

## Register Descriptions

### Flash Target Address Register – TADR

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

Offset: 0x000

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
TADB								
Type/Reset	RW	0						
23 22 21 20 19 18 17 16								
TADB								
Type/Reset	RW	0						
15 14 13 12 11 10 9 8								
TADB								
Type/Reset	RW	0						
7 6 5 4 3 2 1 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-bit, TADR shall be set as word-aligned (4-byte). 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, TADB [9:0] will be ignored in order to limit the target address as 1 KB-aligned.</p>

## Flash Write Data Register – WRDR

This register stores the data for programming operation.

Offset: 0x004

Reset value: 0x0000\_0000

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

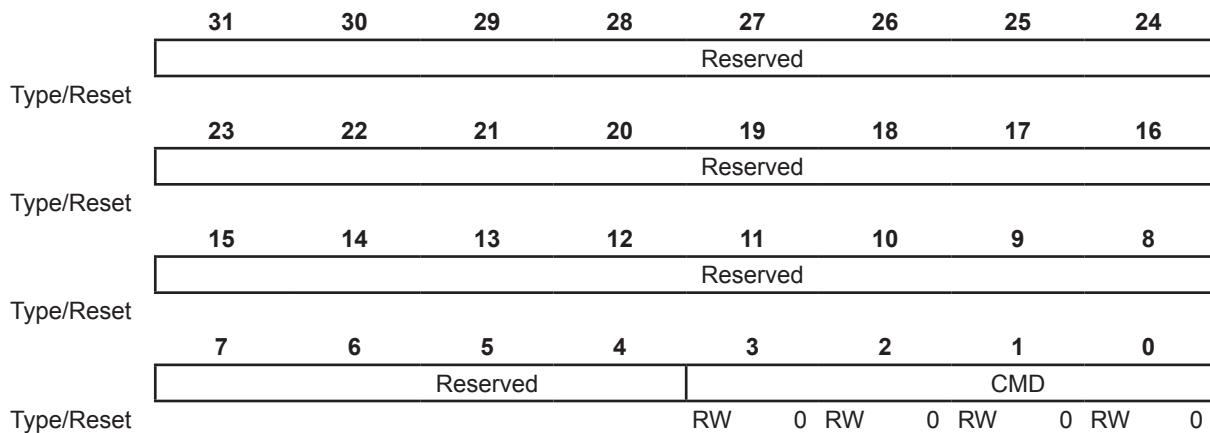
Bits	Field	Descriptions
[31:0]	WRDB	Flash Write Data Bits The data value for programming operation.

## Flash Operation Command Register – OCMR

This register is used to specify the Flash operation commands that include 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 the definitions of the operation command bits CMD [3:0] which determine the Flash memory operation. If an invalid command is set and the IOCMIEN bit is set to 1, an Invalid Operation Command interrupt will occur.</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

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			OPM			Reserved		
	RW	0	RW	0	RW	0	RW	0	

Bits	Field	Descriptions										
[4:1]	OPM	<p>Operation Mode</p> <p>The following table shows the operation modes of the FMC. User can commit the command which is set by the OCMR register for the FMC according to the address alias setting in the TADR register. The contents of the TADR, WRDR, and OCMR registers should be prepared before setting this register. After all the operations have been finished, the OPM field will be set as 0xE by the FMC hardware. The Idle mode can be set when all the operations have been finished for power saving purposes. Note that the operation status should be checked before the next operation is executed on the FMC. The contents of the 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 the FMC interrupt function. The FMC generates interrupts to the controller when corresponding interrupt enable bits are set.

Offset: 0x014

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			OREIEN	IOCMIEN	OBEIEN	ITADIEN	ORFIEN		
	RW			0	RW	0	RW	0	RW	

Bits	Field	Descriptions
[4]	OREIEN	Operation Error Interrupt Enable 0: Operation error interrupt is disabled 1: Operation error interrupt is enabled
[3]	IOCMIEN	Invalid Operation Command Interrupt Enable 0: Invalid operation command interrupt is disabled 1: Invalid operation command interrupt is enabled
[2]	OBEIEN	Option Byte Checksum Error Interrupt Enable 0: Option Byte checksum error interrupt is disabled 1: Option Byte checksum error interrupt is enabled
[1]	ITADIEN	Invalid Target Address Interrupt Enable 0: Invalid target address interrupt is disabled 1: Invalid target address interrupt is enabled
[0]	ORFIEN	Operation Finished Interrupt Enable 0: Operation finish interrupt is disabled 1: Operation finish interrupt is enabled

## Flash Operation Interrupt and Status Register – OISR

This register indicates the FMC interrupt status which is used to check if a Flash operation has been finished otherwise an error occurs. The status bits are available when the corresponding enable bits in the OIER register are set.

Offset: 0x018

Reset value: 0x0001\_0000

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

Bits	Field	Descriptions
[17]	PPEF	Page Erase/Program Protected Error Flag 0: Page Erase/Program Protected Error does not occur 1: Operation error occurs due to an invalid erase/program operation being 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 has not yet finished 1: The last Flash operation command has finished This bit is directly connected to the 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 Flash operation error occurs such as an invalid command, program error and erase error, etc. 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 has been written to the OCMR register The IOCM interrupt will occur if the IOCMIEN bit in the OIER register is set. Reset this bit by writing 1.
[2]	OBEF	Option Byte Checksum Error Flag 0: Option Byte checksum is correct 1: Option Byte checksum is incorrect The OBE interrupt will occur if the OBEIEN bit in the OIER register is set. Reset this bit by writing 1.

<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>
[1]	ITADF	<p>Invalid Target Address Flag</p> <p>0: The target address is valid 1: The target address is invalid</p> <p>The data in the TADR field must have a range from 0x0000_0000 to 0x1FFF_FFFF. An ITAD interrupt will occur if the ITADIEN bit in the OIER register is set. Reset this bit by writing 1.</p>
[0]	ORFF	<p>Operation Finished Flag</p> <p>0: Flash operation has not finished 1: Last Flash operation has finished</p> <p>The ORF interrupt will occur if the ORFIEN bit in the OIER register is set. Reset this bit by writing 1.</p>

## Flash Page Erase/Program Protection Status Register – PPSR

This register indicates the page protection status of the Flash memory.

Offset: 0x020 ~ 0x02C

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						

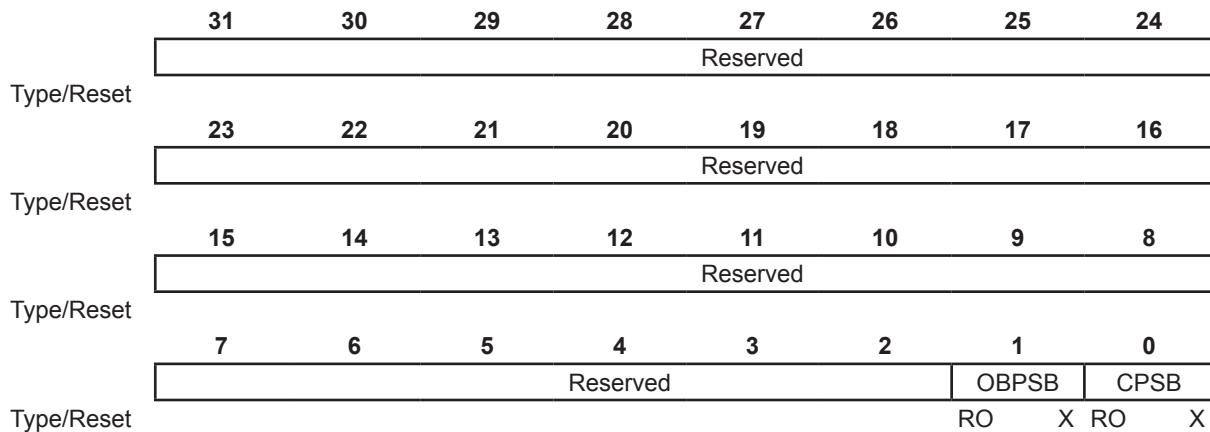
Bits	Field	Descriptions
[127:0]	PPSBn	<p>Page n Erase/Program Protection Status Bits (n = 0 ~ 62)</p> <p>0: The corresponding page n is protected 1: The corresponding page n is not protected</p> <p>The content of this register is not dynamically updated and will only be reloaded by the Option Byte loader which is activated when any kind of reset occurs. The erase or program function of the specific pages is not allowed when the corresponding bits of the PPSR registers are reset. The reset value of the bits PPSR [62:0] is determined by the Option Byte, OB_PP [62:0]. The other bits of the OB_PP and PPSR registers are reserved for future usage.</p>

## Flash Security Protection Status Register – CPSR

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

Offset: 0x030

Reset value: 0xFFFF\_FFFF



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

## Flash Vector Mapping Control Register – VMCR

This register is used to control the vector mapping. The reset value of the VMCR register is determined by the status of the external booting pins, BOOT0 and BOOT1 during the power-on reset period.

Offset: 0x100

Reset value: 0x0000\_000X

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						VMCB			
							RW	X	RW	X

Bits	Field	Descriptions									
[1:0]	VMCB	Vector Mapping Control Bit									
		The VMCB bits are used to control the mapping source of the first 4-word vectors addressed from 0x0 to 0xC. The following table shows the vector mapping setting setup.									
		<b>BOOT1</b>	<b>BOOT0</b>	<b>VMCB [1:0]</b>	Descriptions						
		Low	Low	00	SRAM booting mode The vector mapping source is SBVT0~3.						
		Low	High	01	Boot Loader mode The vector mapping source is the Boot Loader area.						
		High	Low	10	Main Flash mode The vector mapping source is the main Flash memory area.						
		High	High	11							

The reset value of the VMCR register is determined by the pins status of the external booting pins BOOT1 and BOOT0 during a power-on reset and system reset. However, when the application program is executed, the vector mapping setting can be temporarily changed by configuring the VMCB bits to correctly access the first 4-word vectors in the Flash memory, especially when the CPU is booted from the Boot Loader or the SRAM region.

## Flash Pre-fetch Control Register – CFCR

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

Offset: 0x200

Reset value: 0x0000\_13D1

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

Bits	Field	Descriptions
[16]	FZWPSEN	Flash Zero Wait-state Power Saving Enable Bit 0: Flash zero wait-state power saving is disabled 1: Flash zero wait-state power saving is enabled This bit can only be set to 1 to reduce the Flash memory operating current when the Flash memory is operated in zero wait-state. This bit has no effect when the wait-state setting is changed to non-zero.
[15]	FHLAEN	Flash Memory Half-cycle Access Enable Bit 0: Half-cycle access is disabled 1: Half-cycle access is enabled This bit can only be set when the Flash memory is operated in zero wait-state. Setting this bit to 1 will reduce the Flash memory operation current further when the system frequency, HCLK, is less than a frequency of 12MHz. This bit has no effect when the wait-state setting is changed to non-zero.
[12]	CE	Branch Cache Enable Bit 0: Branch cache is disabled 1: Branch cache is enabled The branch cache is enabled in default state.
[7]	DCDB	DCODE data Cacheable Enable Bit 0: DCODE data is Cacheable 1: DCODE data is Non-Cacheable The DCODE data is non-cacheable in default state.
[4]	PFBE	Pre-fetch Buffer Enable Bit 0: Pre-fetch buffer is disabled 1: Pre-fetch buffer is enabled The pre-fetch buffer is enabled in default state. When the pre-fetch buffer is disabled, the instruction and data are provided directly by the Flash memory.

Bits	Field	Descriptions															
[2:0]	WAIT	<p>Flash Wait-state Setting</p> <p>These bits are used to set the HCLK wait clock during a non-sequential Flash access. The actual value of the wait clocks is given by (WAIT [2:0] – 1). Since a wide access interface with a pre-fetch buffer and branch cache is provided, the wait-state of sequential Flash access is very close to zero.</p> <table border="1"><thead><tr><th>WAIT [2:0]</th><th>Wait-state cycle</th><th>HCLK</th></tr></thead><tbody><tr><td>001</td><td>0</td><td>1 MHz &lt; HCLK ≤ 24 MHz</td></tr><tr><td>010</td><td>1</td><td>24 MHz &lt; HCLK ≤ 48 MHz</td></tr><tr><td>011</td><td>2</td><td>48 MHz &lt; HCLK ≤ 72 MHz</td></tr><tr><td>Others</td><td>Reserved</td><td>Reserved</td></tr></tbody></table>	WAIT [2:0]	Wait-state cycle	HCLK	001	0	1 MHz < HCLK ≤ 24 MHz	010	1	24 MHz < HCLK ≤ 48 MHz	011	2	48 MHz < HCLK ≤ 72 MHz	Others	Reserved	Reserved
WAIT [2:0]	Wait-state cycle	HCLK															
001	0	1 MHz < HCLK ≤ 24 MHz															
010	1	24 MHz < HCLK ≤ 48 MHz															
011	2	48 MHz < HCLK ≤ 72 MHz															
Others	Reserved	Reserved															

## SRAM Booting Vector Register n – SBVTn, n = 0 ~ 3

These registers specify the initial values of the Stack Point, Program Counter, NMI Handler address, and hard fault handler address for the SRAM boot mode.

Offset: 0x300 ~ 0x30C

Reset value: Various depending on the address offset

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

Bits	Field	Descriptions
[31:0]	SBVTn	<p>SRAM Booting Vector n ( n = 0 ~ 3)</p> <p>The SRAM Booting Vector 0 ~ 3 provide a SRAM booting capability for application debugging. The contents of the SBVTn registers are re-mapped into the addresses 0x0 to 0xC of the Flash memory CODE area under the SRAM boot mode Refer to the description of the VCMR register and BOOT1/BOOT0 boot pins. The following table shows the purpose and reset value of the SBVTn register. The reset value provides a fixed setting for program execution during the SRAM boot mode. These registers can be modified by the debugging tool in order to change the program execution setting. The reset values of the SBVTn register will be reloaded only by power-on reset. Other reset sources will have no effect.</p>

Name	Address Offset	Purpose Descriptions	Reset Value
SBVT0	0x300	Stack point	16 KB SRAM: 0x2000_4000 8 KB SRAM: 0x2000_2000
SBVT1	0x304	Program counter	0x2000_0155
SBVT2	0x308	NMI handler address	0x0000_0000
SBVT3	0x30C	Hard fault handler address	0x0000_0000

This access width of the SBVTn register must be 32-bit (Word access). 8 or 16-bit (Byte or Half-Word) access is not allowed.

## 5 Power Control Unit (PWRCU)

### Introduction

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

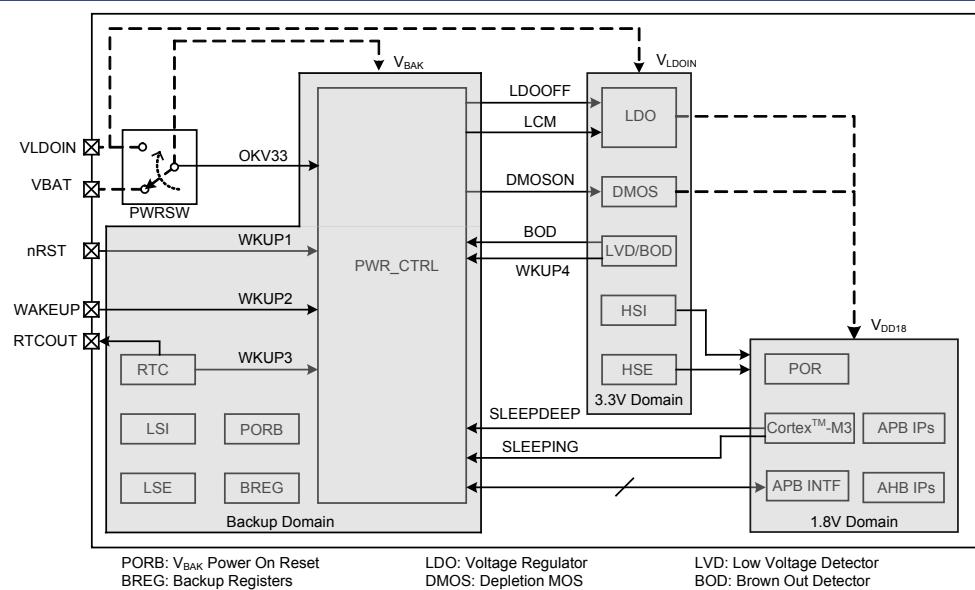


Figure 11. PWRCU Block Diagram

### Features

- Three power domains: Backup, 3.3V and 1.8V power domains.
- Four power saving modes: Sleep, Deep-Sleep1, Deep-Sleep2 and Power-Down modes.
- Internal Voltage regulator supplies 1.8 V voltage source.
- Additional Depletion MOS supplies 1.8 V voltage source with low leakage and low operating current.
- Brown Out Detector can issue a system reset or an interrupt when 3.3 V power source ( $V_{LDOIN}$ ) is lower than 2.6 V.
- Low Voltage Detector can issue an interrupt or wakeup event when  $V_{LDOIN}$  is lower than a programmable threshold voltage ranging from 2.7 V to 3.5 V (except 3.3V).
- Battery power ( $V_{BAT}$ ) for backup domain when  $V_{LDOIN}$  is shut down.
- 40 bytes of backup registers powered by  $V_{BAK}$  for data storage of user application data when in the Power-Down mode.

## Functional Descriptions

### Backup Domain

#### Power Switch

The Backup Domain is powered by the 3.3V power source,  $V_{LDOIN}$ , or the battery power source,  $V_{BAT}$ , which is selected by the power switch PWRSW. The operating voltage of the power switch ranges from 2.7V to 3.6V. If  $V_{LDOIN}$  is lower than  $V_{BAT}$ , then the power source will be switched from  $V_{LDOIN}$  to  $V_{BAT}$ . Therefore, even if  $V_{LDOIN}$  is powered down, all the circuitry in the backup domain can operate normally. This means that the backup register contents will be retained, the RTC circuitry will operate normally and the low speed oscillators can keep running.

#### Backup Domain Reset

The Backup Domain reset sources include the Backup Domain power-on-reset (PORB) and the Backup Domain software reset which is activated by setting the BAKRST bit in the BAKCR register. The PORB signal forces the device to stay in the reset mode until the  $V_{BAK}$  is greater than 2.1V. The slew rate of PORB signal is approximately  $V_{BAK}/100ms$ . Also the application software can trigger Backup Domain software reset by setting the BAKRST bit in the BAKCR register. All registers of PWRCU and RTC will be reset only by the Backup Domain reset.

#### LSE, LSI and RTC

The Real Time Clock circuitry clock source can be derived from either the Low Speed Internal RC oscillator, LSI, or the Low Speed External Crystal oscillator, LSE. Before entering the power saving mode by executing WFI/WFE instruction, the Cortex™-M3 needs to setup the compare register with an expected wakeup time and enable the wakeup function to achieve the RTC timer wakeup event. After entering the power saving mode for a certain amount of time, the Compare Match flag, CMFLAG, will be asserted to wakeup the device when the compare match event occurs. The details of the RTC configuration for wakeup timer will be described in the RTC chapter.

#### Backup Registers and Isolation Cells

Ten 32-bit registers, up to 40 bytes, are located in the Backup Domain for user application data storage. These registers are powered by  $V_{BAK}$  which constantly supplies power when the 1.8V power is switched off. The Backup Registers are only reset by the Backup Domain power-on-reset, PORB, or the Backup Domain software reset, BAKRST. When the device resumes operation from the 1.8V power, either by Hardware or Software, access to the Backup registers and the RTC registers are disabled by the isolation cells which protect these registers against possible parasitic write accesses. To resume access operations, users can disable these isolation cells by setting the BKISO bit in the LPCR register of the Clock Control Unit to 1.

### 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  $V_{LDOIN} \leq 2.6V$

The LDO will be automatically switched on by hardware if any of the following conditions occurs:

- Resume operation from the power saving mode - RTC wakeup, LVD wakeup or WAKEUP pin rising edge
- Detect a falling edge on the external reset pin (nRST)
- The control bit BODEN = 1 and the supply power  $V_{LDOIN} > 2.6V$

To enter the Deep-Sleep1 mode, the PWRCU will request the LDO to operate in a low current mode, LCM. To enter the Deep-Sleep2 mode, the PWRCU will turn off the LDO and turn on the DMOS to supply an alternative 1.8V power.

## 3.3V Power Domain

### Voltage Regulator

The voltage regulator, LDO, Depletion MOS, DMOS, Low voltage Detector, LVD, the Brown out Detector, BOD and High Speed Internal oscillator, HSI, all operate under the 3.3 V power domain. The LDO can be configured to operate in either normal mode (LDOOFF=0, SLEEPDEEP=0, I=200mA) or low current mode (LDOOFF=0, SLEEPDEEP=1, I= 100 mA) to supply the 1.8 V power. An alternative 1.8 V power source is the output of the DMOS which has low leakage and drive current characteristics. It is controlled using the DMOSON bit in the BAKCR register. The DMOS output has weak drive current capability and can operate only in the Deep-Sleep2 mode for data retention purposes in the  $V_{DD18}$  power domain.

### Low Voltage Detector/Brown Out Detector

The Brown Out Detector, BOD, is used to detect if the 3.3 V supply voltage is equal to or lower than 2.6V. When the BODEN bit in the LVDCSR register is set to 1 and the 3.3 V supply voltage is lower than 2.6 V 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 Cortex™-M3 to execute a power down procedure when the BODRIS bit is set to 1. The Low Voltage Detector, LVD, can also detect whether the supply voltage is lower than a programmable threshold voltage ranging from 2.7 V to 3.5 V. It is selected by the LVDS bits in the LVDCSR register. When a low voltage on the  $V_{LDOIN}$  power pin is detected, the LVDF flag will be active and an interrupt will be generated and sent to the Cortex™-M3 if the LVDEN and LVDIWEN bits in the LVDCSR register are set.

### High Speed Internal Oscillator

The High Speed Internal Oscillator, HSI, is located in the 3.3 V 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 1.8 V power on reset will occur.

## 1.8V Power Domain

The main functions that include the APB interface for the backup domain, 1.8 V power on reset (POR), Cortex™-M3 logic, AHB/APB peripherals, and so on are located in this power domain. Once the 1.8V is powered up, the POR will generate a reset sequence (Refer to PORB) on the 1.8 V power domain. Subsequently, to enter the expected power saving mode, the associated control bits including the LDOOFF, DMOSON, and SLEEPDEEP bits must be configured. Then, once a WFI or WFE instruction is executed, the device will enter an expected power saving mode which will be discussed in the following section.

## Operation Modes

### Run Mode

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

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

**Table 11. Operation Mode Definitions**

Mode name	Hardware Action
Run	After system reset, Cortex™-M3 fetches instructions to execute.
Sleep	1. Cortex™-M3 core clock will be stopped. 2. Peripherals, Flash and SRAM clocks can be stopped.
Deep-Sleep	1. Stop all clocks in the 1.8 V power domain. 2. Disable HSI, HSE, and PLL. 3. Reduce the 1.8 V power domain current by turning on the LDO low current mode or DMOS.
Power-Down	Shut down the 1.8 V power domain

### Sleep Mode

By default, only the Cortex™-M3 clock will be stopped in the Sleep mode. Clearing the FMCEN or SRAMEN bit in the CKCU AHBCCR register to 0 will have the effect of stopping the Flash clock or SRAM clock after the system enters the Sleep mode. If it is not necessary for the CPU to access the Flash memory and SRAM in the Sleep mode, it is recommended to clear the FMCEN and SRAMEN bits in the AHBCCR register to minimize power consumption. To enter the Sleep mode, it is only necessary to clear the SLEEPDEEP bit to 0 and execute a WFI or WFE instruction. The system will exit from the Sleep mode via any interrupt or event trigger. The accompanying table provides more information about the power saving modes.

**Table 12. Enter/Exit Power Saving Modes**

Mode	Mode Entry				Mode Exit
	Cortex™-M3 Instruction	Cortex™-M3 SLEEPDEEP	LDOOFF	DMOSON	
Sleep		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)
Deep-Sleep1	WFI or WFE (Takes effect)	1	0	0	Any EXTI in event mode or WDT timeout or CMP wakeup or RTC wakeup or LVD wakeup <sup>(2)</sup> or WAKEUP pin rising edge USB resume
		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.

### 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 1.8V 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 Cortex™-M3 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 Cortex™-M3 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 microcontroller out of the Power-Down mode. In the Power-Down mode, the 1.8 V power supply will be turned off. The remaining active power supplies are the 3.3 V I/O power ( $V_{DD33}$ ) and the Backup Domain power ( $V_{BAK}$ ).

After a system reset, the PORSTF bit in the RSTCU GRSR register, the PDF and BAKPORF bits in the BAKSR register should be checked by software to confirm if the device is being resumed from the Power-Down mode by a backup domain power on reset, an unexpected loss of the 1.8 V power or other reset events (nRST, WDT,...). If the device has entered the Power-Down mode under the correct firmware procedure, then the PDF bit will be set. The System information could be saved in the Backup Registers and be retrieved when the 1.8 V power domain is powered on again. More information about the PDF and BAKPORF bits in the BAKSR register and PORSTF bit in the RSTCU GRSR register is shown in the following table.

**Table 13. Power Status After System Reset**

BAKPORF	PDF	PORSTF	Description
1	0	1	Power-up for the first time after the backup domain is reset: Power on reset when $V_{BAK}$ is applied for the first time or executing a software reset command on the backup domain.
0	0	1	Restart from unexpected loss of the 1.8 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>BAK</sub> backup power domain.

**Table 14. PWRCU Register Map**

Register	Offset	Description	Reset Value
<b>PWRCU Base Address = 0x4006_A000</b>			
BAKSR	0x100	Backup Domain Status Register	0x0000_0001
BAKCR	0x104	Backup Domain Control Register	0x0000_0000
BAKTEST	0x108	Backup Domain Test Register	0x0000_0027
HSIRCR	0x10C	HSI Ready Counter Control Register	0x0000_0003
LVDCSR	0x110	Low Voltage/Brown Out Detect Control and Status Register	0x0000_0000
BAKREG0	0x200	Backup Register 0	0x0000_0000
BAKREG1	0x204	Backup Register 1	0x0000_0000
BAKREG2	0x208	Backup Register 2	0x0000_0000
BAKREG3	0x20C	Backup Register 3	0x0000_0000
BAKREG4	0x210	Backup Register 4	0x0000_0000
BAKREG5	0x214	Backup Register 5	0x0000_0000
BAKREG6	0x218	Backup Register 6	0x0000_0000
BAKREG7	0x21C	Backup Register 7	0x0000_0000
BAKREG8	0x220	Backup Register 8	0x0000_0000
BAKREG9	0x224	Backup Register 9	0x0000_0000

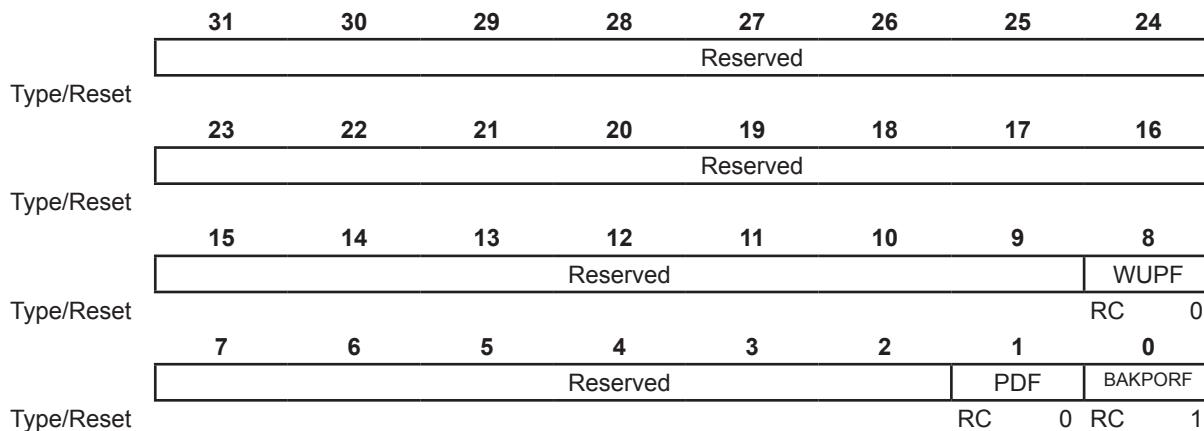
## Register Descriptions

### Backup Domain Status Register – BAKSR

This register indicates backup domain status.

Offset: 0x100

Reset value: 0x0000\_0001 (Reset only by Backup Domain reset)



Bits	Field	Descriptions
[8]	WUPF	<p>External WAKEUP Pin Flag</p> <p>0: The Wakeup pin is not asserted 1: The Wakeup pin is asserted</p> <p>This bit is set by hardware when the WAKEUP pin asserts and is cleared by a software read. Software should read this bit to clear it after a system wake up from the power saving mode.</p>
[1]	PDF	<p>Power Down Flag</p> <p>0: Wakeup from abnormal <math>V_{DD18}</math> shutdown (Loss of <math>V_{DD18}</math> is unexpected) 1: Wakeup from Power-Down mode. The loss of <math>V_{DD18}</math> is under expectation.</p> <p>This bit is set by hardware when the system has successfully entered the Power-Down mode This bit is cleared by a software read.</p>
[0]	BAKPORF	<p>Backup Domain Reset Flag</p> <p>0: Backup Domain reset does not occur 1: Backup Domain reset occurs</p> <p>This bit is set by hardware when Backup Domain reset occurs, either a Backup Domain power on reset or Backup Domain software reset. The bit is cleared by a software read. This bit must be cleared after the system is first powered, otherwise it will be impossible to detect when a Backup Domain reset has been triggered. When this bit is read as 1, a read software loop must be implemented until the bit returns again to 0. This software loop is necessary to confirm that the Backup Domain is ready for access. It must be implemented after the Backup Domain is first powered up.</p>

## Backup Domain Control Register – BAKCR

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

Offset: 0x104

Reset value: 0x0000\_0000 (Reset only by Backup Domain reset)

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	DMOSSTS	Reserved		V18RDYSC	Reserved		WUPIEN	WUPEN	
	RO	0	RW		0	RW		0 RW 0	
Type/Reset	7	6	5	4	3	2	1	0	
	DMOSON	Reserved			LDOOFF	Reserved		BAKRST	
Type/Reset	RW	0	RW		0	WO		0	

Bits	Field	Descriptions
[15]	DMOSSTS	Depletion MOS Status This bit is set to 1 if the DMOSON bit in this register has been set to 1. This bit is cleared to 0 if the DMOSON bit has been set to 0 or if a BOD reset occurred.
[12]	V18RDYSC	$V_{DD18}$ Ready Source Selection Setting this bit to determine what control signal of isolation cells is used to disable the isolation function of the $V_{18}$ to $V_{33}$ power domain level shifter. 0: BKISO bit in the LPCR register located in the CKCU 1: $V_{DD18}$ POR
[9]	WUPIEN	External WAKEUP Pin Interrupt Enable 0: Disable WAKEUP pin interrupt function 1: Enable WAKEUP pin interrupt function The software can set the WUPIEN bit to 1 to assert the LPWUP interrupt in the NVIC unit when both the WUPEN and WUPF bits are set to 1.

Bits	Field	Descriptions
[8]	WUPEN	<p>External WAKEUP Pin Enable</p> <p>0: Disable WAKEUP pin function 1: Enable WAKEUP pin function</p> <p>The Software can set the WUPEN bit as 1 to enable the WAKEUP pin function before entering the power saving mode. When WUPEN = 1, a rising edge on the WAKEUP pin wakes up the system from the power saving mode. As the WAKEUP pin is active high, this bit will set an input pull down mode when the bit is high. The corresponding register bits which should be properly setup are the PCPD [15] to 1 in the PCPDR register, the PCPU [15] to 0 in the PCPUR register and the PCCFG15 field to 0x01 in the GPCCFGH register.</p> <p>Note: This bit is reset by a system reset or a Backup Domain reset. Because this bit is located in the Backup Domain, after reset activity there will be a delay until the bit is active. The bit will not be active until the system reset finished and the Backup Domain ISO signal has been disabled. This means that the bit can not be immediately set by software after a system reset finished and the Backup domain ISO signal disabled. The delay time needed is a minimum of three 32KHz clock periods until the bit reset activity has finished.</p>
[7]	DMOSON	<p>DMOS Control</p> <p>0: DMOS is OFF 1: DMOS is ON</p> <p>A DMOS is implemented to provide an alternative voltage source for the 1.8 V power domain when the Cortex™-M3 enters the Deep-Sleep mode (SLEEPDEEP=1). The control bit DMOSON is set by software and cleared by software or PORB. If the DMOSON bit is set to 1, the LDO will automatically be turned off when the Cortex™-M3 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 Cortex™-M3 enters the Deep-Sleep mode (SLEEPDEEP = 1). The <math>V_{DD18}</math> power is available. 1: The LDO is turned off when the Cortex™-M3 enters the Deep-Sleep mode (SLEEPDEEP=1). The <math>V_{DD18}</math> power is not available.</p> <p>Note: This bit is only available when the DMOSON bit is cleared to 0.</p>
[0]	BAKRST	<p>Backup Domain Software Reset</p> <p>0: No action 1: Backup Domain Software Reset is activated – includes all the related RTC and PWRCU registers</p>

## Backup Domain Test Register – BAKTEST

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

Offset: 0x108

Reset value: 0x0000\_0027

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
	BAKTEST							
	RO	0	RO	0	RO	1	RO	1

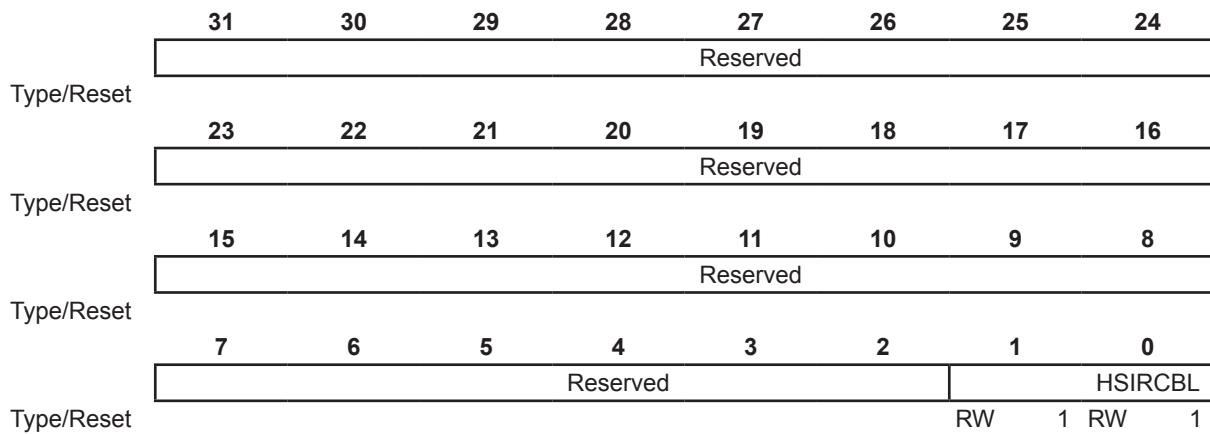
Bits	Field	Descriptions
[7:0]	BAKTEST	Backup Domain Test Bits A constant 0x27 will be read when the Backup Domain is ready for Cortex™-M3 access.

## HSI Ready Counter Control Register – HSIRCR

This register specifies the counter bit length of the HSI ready counter.

Offset: 0x10C

Reset value: 0x0000\_0003 (Reset only by Backup Domain reset)



Bits	Field	Descriptions
[1:0]	HSIRCBL	<p>HSI Ready Counter Bit Length</p> <p>00: 4 bits 01: 5 bits 10: 6 bits 11: 7 bits</p> <p>The HSIRCBL field specifies the bit length of the HSI ready counter. Software can set the HSIRCBL field to shorten the startup waiting time of the HSI oscillator before entering the Deep-Sleep mode or Power-Down mode. (HSIRCBL is reset only by Backup Domain reset)</p>

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

This register specifies flags, enable bits, and option bits for Low-Voltage/Brown-out detector.

Offset: 0x110

Reset value: 0x0000\_0000 (Reset only by Backup Domain reset)

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

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]	LVDIWIN	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 LVDIWIN bit is set to 1, a LVD interrupt will be generated and sent to the Cortex™-M3 NVIC unit.
[19]	LVDF	Low Voltage Detect Status Flag 0: V <sub>DD33</sub> is higher than the specific voltage level 1: V <sub>DD33</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 Cortex™-M3 if the LVDIWIN bit is set to 1. However, if the LVDEWEN bit is set to 1 and the LVDIWIN bit is cleared to 0, only a LVD event will be generated rather than a LVD interrupt when the LVDF flag is asserted.

Bits	Field	Descriptions
[22], [18:17]	LVDS [2:0]	Low Voltage Detect Level Selection 000: 2.7 V (nominal, default) 001: 2.8 V (nominal) 010: 2.9 V (nominal) 011: 3.0 V (nominal) 100: 3.1 V (nominal) 101: 3.2 V (nominal) 110: 3.4 V (nominal) 111: 3.5 V (nominal)
[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 3.3V 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 Detection Flag If $V_{DD33} < 2.6$ V, BODF = 1. Otherwise, BODF = 0.
[1]	BODRIS	BOD Reset or Interrupt Selection 0: Reset the whole device 1: Generate Interrupt
[0]	BODEN	Brown Out Detect Enable 0: Disable Brown Out Detect 1: Enable Brown Out Detect

## Backup Register n – BAKREGn, n = 0 ~ 9

This register specifies backup register n for storing data during the V<sub>DD18</sub> power-off period.

Offset: 0x200 ~ 0x224

Reset value: 0x0000\_0000 (Reset only by Backup Domain reset)

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

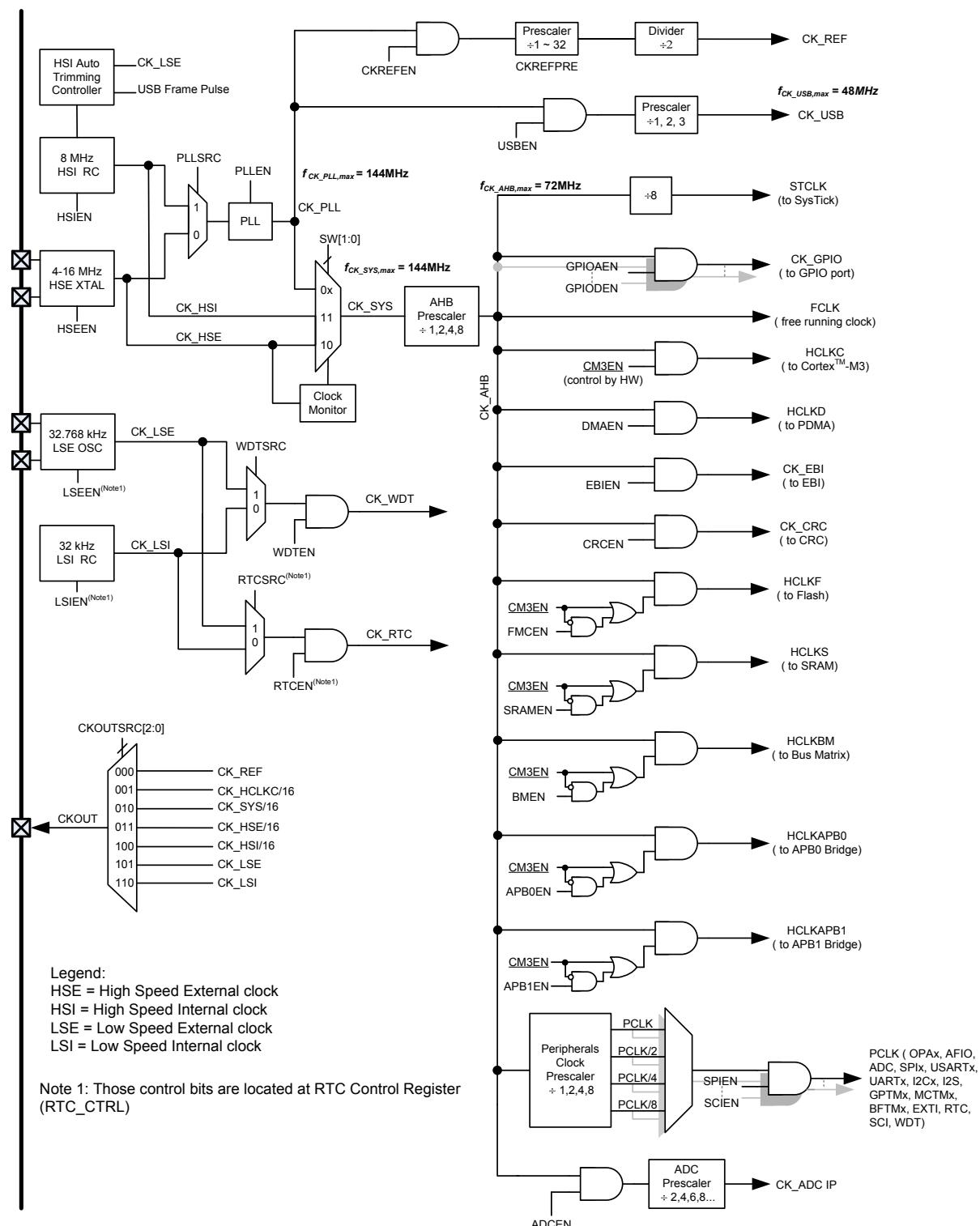
Bits	Field	Descriptions
[31:0]	BAKREGn	Backup Register n (n = 0 ~ 9) These registers are used for data storage in general purpose. The contents of BAKREGn registers will remain even if the V <sub>DD18</sub> power is lost.

## 6 Clock Control Unit (CKCU)

### Introduction

The Clock Control unit, CKCU, provides a range of frequencies and clock functions. These include a High Speed Internal RC oscillator (HSI), a High Speed External crystal oscillator (HSE), a Low Speed Internal RC oscillator (LSI), a Low Speed External crystal oscillator (LSE), a Phase Lock Loop (PLL), a HSE clock monitor, clock prescalers, clock multiplexers and clock gating circuitry. The clocks of the AHB, APB and Cortex™-M3 are derived from the system clock (CK\_SYS) which can source from the HSI, HSE or PLL. The Watchdog Timer (WDT) and Real Time Clock (RTC) use either the LSI or LSE as their clock source. The maximum operating frequency of the system core clock (CK\_AHB) can be up to 72 MHz.

Some of the internal clocks can also be wired out via the CKOUT pin for debugging purposes. The clock monitor circuit can be used to detect an HSE clock failure. Once the HSE clock has ceased to operate, for whatever reason, the CKCU will force the system clock source to switch to the HSI clock to prevent a system halt from occurring.



Legend:  
 HSE = High Speed External clock  
 HSI = High Speed Internal clock  
 LSE = Low Speed External clock  
 LSI = Low Speed Internal clock

Note 1: Those control bits are located at RTC Control Register (RTC\_CTRL)

Figure 12. CKCU Block Diagram

## Features

- 4 to 16 MHz High Speed External crystal oscillator – HSE
  - 8 MHz High Speed Internal RC oscillator – HSI
  - 32,768 Hz Low Speed External crystal oscillator – LSE
  - 32 kHz Low Speed Internal RC oscillator – LSI
  - PLL clock source can be HSE or HSI
  - HSE clock monitor

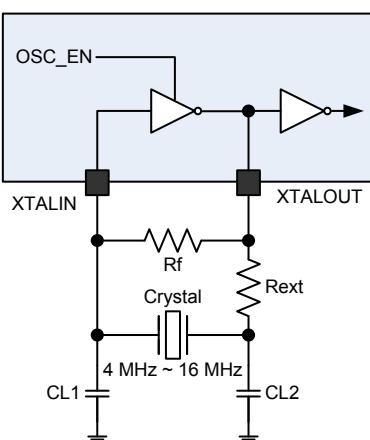
## Functional Descriptions

High Speed External Crystal Oscillator – HSE

The high speed external crystal oscillator, HSE, which has a frequency from 4 to 16 MHz, produces a highly accurate clock source for use as the system clock. The crystal with a specific frequency must be connected and located close to the two HSE pins, XTALIN / XTALOUT. The external resistor and capacitor components connected to the crystal are necessary for proper oscillation.

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 13. External HSE Crystal, Ceramic, and Resonator**

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 indicates if the high-speed external crystal oscillator is stable. When the HSE is powered up, it will not be released for use until this HSERDY bit is set by the hardware. This specific delay period is known as the oscillator “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. At this point the HSE clock can be used directly as the system clock source or the PLL input clock.

### High Speed Internal RC Oscillator – HSI

The high speed internal RC oscillator, HSI, has a fixed frequency of 8 MHz and is the default clock source selection for the CPU when the device is powered up. The HSI oscillator provides a lower cost type clock source as 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 is used to indicate if the internal RC oscillator is stable. The start-up time of the HSI oscillator 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 when the HSI becomes stable. The HSI clock can also be used as the PLL input clock.

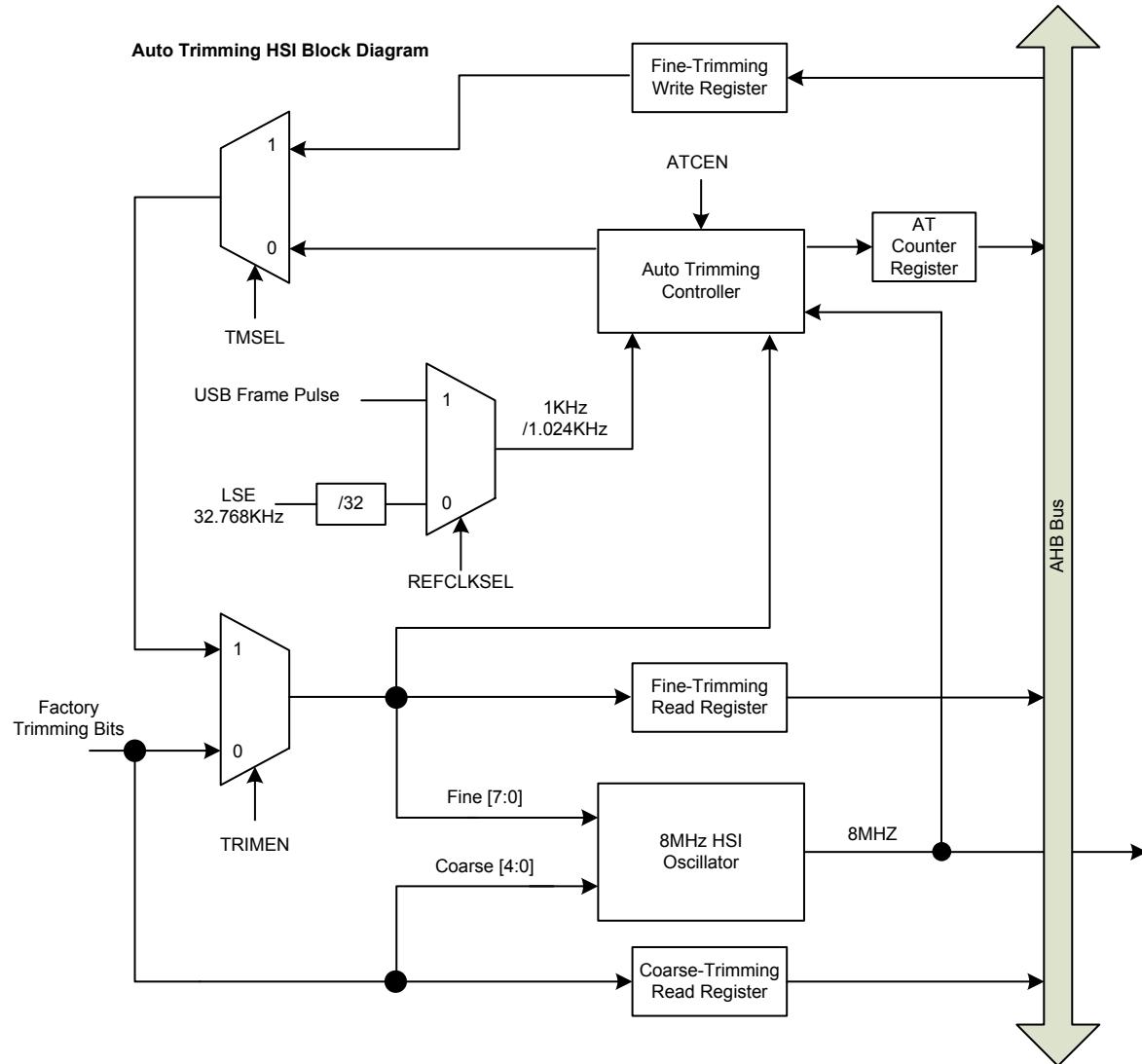
The frequency accuracy of the HSI can be calibrated by the manufacturer, but its operating frequency is still less accurate than HSE. The applications requirements, environment and cost will determine which oscillator type is selected.

If the HSE or PLL is the system clock source, to minimize the time required for the system to recover from the Deep-Sleep Mode, the software can set the PSRCEN bit (Power Saving Wakeup RC Clock Enable) to 1 to force the HSI clock to be system clock when the system initially wake-up. Subsequently, the system clock will automatically be switched back to the original clock source, HSE or PLL, when the original clock source ready flag is asserted. This function will reduce the wakeup time when using the HSE or PLL 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 HT 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 14.

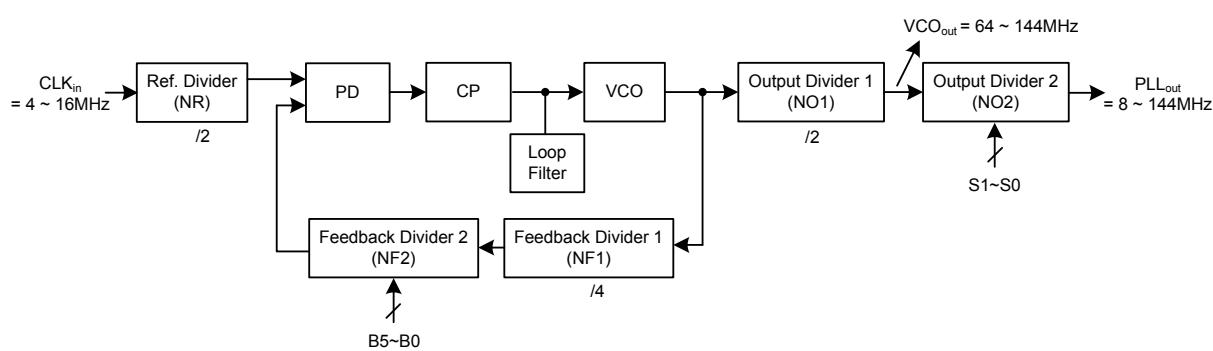
After reset, the factory trimming value is loaded in the HSICOARSE[4:0] and HSIFINE[7:0] bits in the HSI Control Register (HSICR). If the application has to be more accurate HSI frequency that affects by voltage or temperature variations. User can trim manually the HSI frequency using the HSIFINE[7:0] bits in the HSI Control Register (HSICR) or Auto Trimming Controller (ATC) to adjust automatically via a external reference clock in the application. The reference clock can be provided form low speed external crystal or ceramic resonator oscillator LSE with 32,768 Hz or 1ms USB frame synchronous signal.



**Figure 14. The Block Diagram of HSI Auto Trimming**

## Phase Locked Loop – PLL

The internal Phase Locked Loop, PLL, can provide 8 ~ 144 MHz clock output which is 2 ~ 36 multiples of a fundamental reference frequency of 4 ~ 16 MHz. The PLL includes a reference divider, feedback dividers, a digital phase frequency detector (PD), a current-controlled charge pump (CP), an internal loop filter and a voltage-controlled oscillator (VCO) to achieve a stable phase-locked state.



**Figure 15. PLL Block Diagram**

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

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

where NR = Ref divider = 2, NF1 = Feedback Divider 1 = 4, NF2 = Feedback Divider 2 = 1 ~ 64,

NF1 = Output Divider 1 = 2,

NF2 = Output Divider 2 = 1, 2, 4, or 8

Consider a duty cycle of 50% and where both input and output frequencies are divided by 2. If a given clock PLL input clock source frequency,  $CK_{IN}$ , generates a specific PLL output frequency, then it is recommended to use a higher value for NF2 in order to increase PLL stability and reduce jitter at the expense of settling time. The setup bits of the output and feedback divider2 are described in the following two tables. All the setup bits named S1~S0 and B5~B0 in the tables are defined in the PLL Configuration Register PLLCFGR and the PLL Control Register PLLCR in the Register Definition section. Note that the  $VCO_{OUT}$  frequency must have a range from 64 MHz to 144 MHz. If the selected configuration exceeds this range, the PLL output frequency cannot be guaranteed to match the above  $PLL_{OUT}$  formula.

The PLL can be switched on or off by using the PLLEN bit in the Global Clock Control Register GCCR. The PLLRDY flag in the Global Clock Status Register GCSR will indicate if the PLL clock is stable. An interrupt can be generated if the related interrupt enable bit PLLRDYIE in the Global Clock Interrupt Register GCIR is set as the PLL becomes stable.

**Table 15. Output Divider 2 Setting**

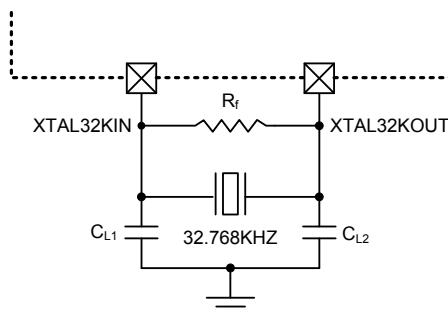
Output divider 2 setup bit S1~S0 (POTD bits in the PLLCFGR register)	NO2 (Output divider 2 value)
00	1
01	2
10	4
11	8

**Table 16. Feedback Divider 2 Value Setting**

Feedback divider 2 setup bits B5~B0 (PFBD bits in the PLLCFGR register)	NF2 (Feedback divider 2 value)
000000	64
000001	1
000010	2
000011	3
000100	4
000101	5
000110	6
000111	7
001000	8
001001	9
001010	10
001011	11
001100	12
001101	13
001110	14
⋮	⋮
111111	63

## Low Speed External Crystal Oscillator – LSE

The low speed external crystal or ceramic resonator oscillator, which has a frequency of 32,768 Hz, produces a low power but highly accurate clock source for the Real-Time-Clock circuit and the Watchdog Timer. The crystal or ceramic resonator must be located close to the two LSE pins, XTAL32KIN and XTAL32KOUT their external resistors and capacitors are necessary for proper oscillation. The LSE oscillator can be switched on or off 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 when the LSE becomes stable.



**Figure 16. External Crystal, Ceramic, and Resonators for LSE**

## Low Speed Internal RC Oscillator – LSI

The low speed internal RC oscillator has a frequency of about 32 kHz and is a low power clock source for the Real Time Clock circuit or the Watchdog Timer. The LSI offers a low cost clock source as no external components are required. The LSI RC oscillator can be switched on or off by using the LSIEN bit in the RTC Control Register, RTCCR. The frequency accuracy can be calibrated using the configuration options. 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 when the LSI becomes stable.

## Clock Ready Flag

The CKCU provides the corresponding clock ready flags of the HSI, HSE, PLL, LSI, and LSE to indicate if these clocks are stable. Before users select the clock 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 bit in the GCSR register. Additionally, the CKCU can trigger an interrupt to notify the 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 in the interrupt service routine.

## System Clock Selection – CK\_SYS

After the system reset, the default CK\_SYS source will be HSI and can be switched to HSE or PLL by changing the System Clock Switch bits, SW, in the Global Clock Control Register, GCCR. When the SW value is changed, the CK\_SYS will continue to operate using the original clock source until the target clock source is stable. The corresponding clock ready status is in the Global Clock Status Register, GCSR, and the clock source usage can be found in the Clock Source Status Register, CKST. When a clock source is used directly by the CK\_SYS or the PLL, it is not possible to stop it.

More detail about function of clock enable is described in below.

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

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

- 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 following action takes effect, the PLL is always under enable state

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

Programming guide of System clock selection is listed in following.

1. Enable any source clock which will become system clock or PLL input clock.
2. Configuring PLLSRC register can not take effect before ready flags of both HSI and HSO are asserted,
3. Configuring SW register to change system clock source will take effect after ready flag of source clock is asserted. Note that system clock will force to HSI if clock monitor is enabled and PLL clock or HSE clock configured as system clock is stuck at 0/1.

## 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. This function should be enabled after the HSE start-up delay and disabled when the HSE is stopped. Once the HSE failure is detected, the HSE will be automatically disabled. The HSE Clock Stuck Flag, CKSF, in the Global Clock Interrupt Register, GCIR, will be set and the HSE failure event will be generated if the Clock Stuck Interrupt Enable bit, CKSIE, in the GCIR register is set. This failure interrupt is connected to the Non-Maskable Interrupt, NMI, of the Cortex-M3. If the HSE is selected as the clock source of CK\_SYS or PLL, the HSE failure will force the CK\_SYS source to HSI and the PLL will be disabled automatically.

## Clock Output Capability

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

**Table 17. CKOUT Clock Source**

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

## Register Map

The following table shows the CKCU register and reset value.

**Table 18. CKCU Register Map**

Register	Offset	Description	Reset Value
<b>CKCU Base Address = 0x4008_8000</b>			
GCFGR	0x000	Global Clock Configuration Register	0x0000_0102
GCCR	0x004	Global Clock Control Register	0x0000_0803
GCSR	0x008	Global Clock Status Register	0x0000_0028
GCIR	0x00C	Global Clock Interrupt Register	0x0000_0000
PLLCFGR	0x018	PLL Configuration Register	0x0000_0000
PLLCR	0x01C	PLL Control Register	0x0000_0000
AHBCFGR	0x020	AHB Configuration Register	0x0000_0000
AHBCCR	0x024	AHB Clock Control Register	0x0000_0005
APBCFGR	0x028	APB Configuration Register	0x0000_0000
APBCCR0	0x02C	APB Clock Control Register 0	0x0000_0000
APBCCR1	0x030	APB Clock Control Register 1	0x0000_0000
CKST	0x034	Clock Source Status Register	0xC100_0000
APBPCSR0	0x038	APB Peripheral Clock Selection Register 0	0x0000_0000
APBPCSR1	0x03C	APB Peripheral Clock Selection Register 1	0x0000_0000
HSICR	0x040	HSI Control Register	0xFFFF_0000 where X is undefined
HSIATCR	0x044	HSI Auto Trimming Counter Register	0x0000_0000
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	
Type/Reset		LPMOD		Reserved					
	RO	0	RO	0	RO	0			
	23	22	21	20	19	18	17	16	
Type/Reset	RW	0	RW	0	Reserved				
	15	14	13	12	11	10	9	8	
Type/Reset	RW	0	RW	0	RW	0	Reserved	PLLSRC	
	7	6	5	4	3	2	1	0	
Type/Reset	Reserved					RW	0	RW	
						CKOUTSRC	1	RW	

Bits	Field	Descriptions
[31:29]	LPMOD	Lower Power Mode Status Set and reset by hardware. 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
[23:22]	USBPRE	USB Clock Prescaler Selection Set and reset by software to control USB clock prescaler setting. 00: CK_USB = CK_PLL 01: CK_USB = CK_PLL / 2 10: CK_USB = CK_PLL / 3 Others: Reserved
[15:11]	CKREFPRE	CK_REF Clock Prescaler Selection Set and reset by software to control CK_REF clock prescaler setting. $CK_{REF} = CK_{PLL} / (CKREFPRE + 1) / 2$ 00000: CK_REF = CK_PLL / 2 00001: CK_REF = CK_PLL / 4 ... 11111: CK_REF = CK_PLL / 64
[8]	PLLSRC	PLL Clock Source Selection Set and reset by software to control PLL clock source. 0: External 4 ~ 16 MHz crystal oscillator clock is selected (HSE) 1: Internal 8 MHz RC oscillator clock is selected (HSI)

<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>
[2:0]	CKOUTSRC	<p>CKOUT Clock Source Selection Set and reset by software.</p> <p>000: CK_REF is selected 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</p>

## Global Clock Control Register – GCCR

This register specifies the clock enable bits.

Offset: 0x004

Reset value: 0x0000\_0803

	31	30	29	28	27	26	25	24	
Reserved									
Type/Reset	23	22	21	20	19	18	17	16	
Reserved									
Type/Reset	15	14	13	12	11	10	9	8	
RW 0 RW 0									
Type/Reset	7	6	5	4	3	2	1	0	
RW 1 RW 1									
Type/Reset	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. Software can set PSRCEN before enter power saving mode in order to reduce the waiting time after wakeup. When PSRCEN = 1, hardware will select HSI as clock source after the system wakeup from power saving mode. Meanwhile, instruction can start execution since the HSI clock is provided to Cortex™-M3. After the original clock source (which selected as CK_SYS before enter power saving mode) is ready, hardware will switch back the clock source as originally.
[16]	CKMEN	HSE Clock Monitor Enable 0: Disable External 4 ~ 16 MHz crystal oscillator clock monitor 1: Enable External 4 ~ 16 MHz crystal oscillator clock monitor When hardware detects HSE clock stuck at low/high state, internal hardware will switch the system clock to internal high speed RC clock (HSI). The only way to recover the system clock is (1) external reset, (2) power on reset or (3) clear CKSF by software.
[11]	HSIEN	Internal High Speed Clock Enable Set and reset by software. This bit can not be reset if HSI clock is used as system clock. 0: Internal 8 MHz RC oscillator clock is set to off 1: Internal 8 MHz RC oscillator clock is set to on
[10]	HSEEN	External High Speed Clock Enable Set and reset by software. This bit can not be reset if the HSE clock is used as system clock. 0: External 4 ~ 16 MHz crystal oscillator clock is set to off 1: External 4 ~ 16 MHz crystal oscillator clock is set to on

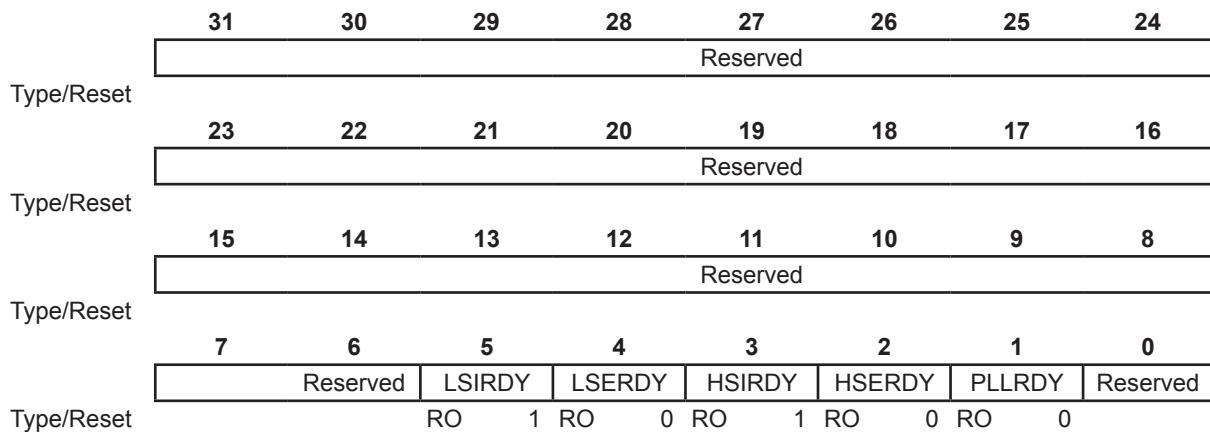
<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>
[9]	PLLEN	<p>PLL Enable</p> <p>Set and reset by software to enable PLL. This bit cannot be reset if the PLL clock is used as system clock.</p> <p>0: PLL off 1: PLL on</p>
[1:0]	SW	<p>System Clock Switch</p> <p>Set and reset by software to select CK_SYS source. Set by hardware to force HSI (b11) when clock failure of the HSE oscillator that is used directly or indirectly as system clock (if the clock monitor is enabled).</p> <p>0x: CK_PLL clock out as system clock 10: CK_HSE as system clock 11: CK_HSI as system clock</p>

## 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 Clock Ready Flag Set by hardware to indicate if the LSI is stable. 0: Internal 32 kHz RC oscillator clock is not ready 1: Internal 32 kHz RC oscillator clock is ready
[4]	LSEIRDY	External Low Speed Clock Ready Flag Set by hardware to indicate if the LSE is stable. 0: External 32,768 Hz crystal oscillator clock is not ready 1: External 32,768 Hz crystal oscillator clock is ready
[3]	HSIRDY	Internal High Speed Clock Ready Flag Set by hardware to indicate if the HSI is stable. 0: Internal 8 MHz RC oscillator clock is not ready 1: Internal 8 MHz RC oscillator clock is ready
[2]	HSEIRDY	External High Speed Clock Ready Flag Set by hardware to indicate if the HSE is stable. 0: External 4 ~ 16 MHz crystal oscillator clock is not ready 1: External 4 ~ 16 MHz crystal oscillator clock is ready
[1]	PLLRDY	PLL Clock Ready Flag Set by hardware to indicate if the PLL is stable. 0: PLL is not ready 1: PLL is ready

## Global Clock Interrupt Register – GCIR

This register specifies interrupt enable and flag bits.

Offset: 0x00C

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[22]	LSIRDYIE	LSI Ready Interrupt Enable Set and reset by software to enable/disable interrupt caused by LSI stabilization. 0: Disable LSI stabilization interrupt 1: Enable LSI stabilization interrupt
[21]	LSERDYIE	LSE Ready Interrupt Enable Set and reset by software to enable/disable interrupt caused by LSE stabilization. 0: Disable LSE stabilization interrupt 1: Enable LSE stabilization interrupt
[20]	HSIRDYIE	HSI Ready Interrupt Enable Set and reset by software to enable/disable interrupt caused by HSI stabilization. 0: Disable HSI stabilization interrupt 1: Enable HSI stabilization interrupt
[19]	HSERDYIE	HSE Ready Interrupt Enable Set and reset by software to enable/disable interrupt caused by HSE stabilization. 0: Disable HSE stabilization interrupt 1: Enable HSE stabilization interrupt
[18]	PLLRDYIE	PLL Ready Interrupt Enable Set and reset by software to enable/disable interrupt caused by PLL stabilization. 0: Disable PLL stabilization interrupt 1: Enable PLL stabilization interrupt
[16]	CKSIE	Clock Stuck Interrupt Enable Set and reset by software to enable/disable interrupt caused by clock monitor. 0: Disable clock fail interrupt 1: Enable clock fail interrupt
[6]	LSIRDYF	LSI Ready Interrupt Flag Reset by software (Write 1 clear). Set by hardware when the Internal 32 kHz RC oscillator clock stabilization and LSIRDYDIE is set. 0: No clock ready interrupt caused by LSI stabilization 1: Clock ready interrupt caused by LSI stabilization

Bits	Field	Descriptions
[5]	LSERDYF	LSE Ready Interrupt Flag Reset by software (Write 1 clear). Set by hardware when the External 32,768 Hz crystal oscillator clock stabilization and LSERDYDIE is set. 0: No clock ready interrupt caused by LSE stabilization 1: Clock ready interrupt caused by LSE stabilization
[4]	HSIRDYF	HSI Ready Interrupt Flag Reset by software (Write 1 clear). Set by hardware when the Internal 8 MHz RC oscillator clock stabilization and HSIRDYDIE is set. 0: No clock ready interrupt caused by HSI stabilization 1: Clock ready interrupt caused by HSI stabilization
[3]	HSERDYF	HSE Ready Interrupt Flag Reset by software (Write 1 clear). Set by hardware when the External 4 ~ 16 MHz crystal oscillator clock stabilization and HSERDYDIE is set. 0: No clock ready interrupt caused by HSE stabilization 1: Clock ready interrupt caused by HSE stabilization
[2]	PLLRDYF	PLL Ready Interrupt Flag Reset by software (Write 1 clear). Set by hardware when the PLL stabilization and PLLRDYDIE is set. 0: Clock not ready interrupt caused by PLL stabilization 1: Clock ready interrupt caused by PLL stabilization
[0]	CKSF	Clock Stuck Interrupt Flag Reset by software (Write 1 clear). Set by hardware when HSE clock stuck and CKSIE is set. 0: Clock works normally 1: HSE clock is stuck

## PLL Configuration Register – PLLCFGR

This register specifies configuration of PLL.

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

Bits	Field	Descriptions
[28:23]	PFBD	PLL VCO Output Clock Feedback Divider (Figure 14 B5 ~ B0) Feedback Divider divides the output clock from VCO of PLL.
[22:21]	POTD	PLL Output Clock Divider (Figure 14 S1 ~ S0)

## PLL Control Register – PLLCR

This register specifies Bypass mode of PLL.

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			

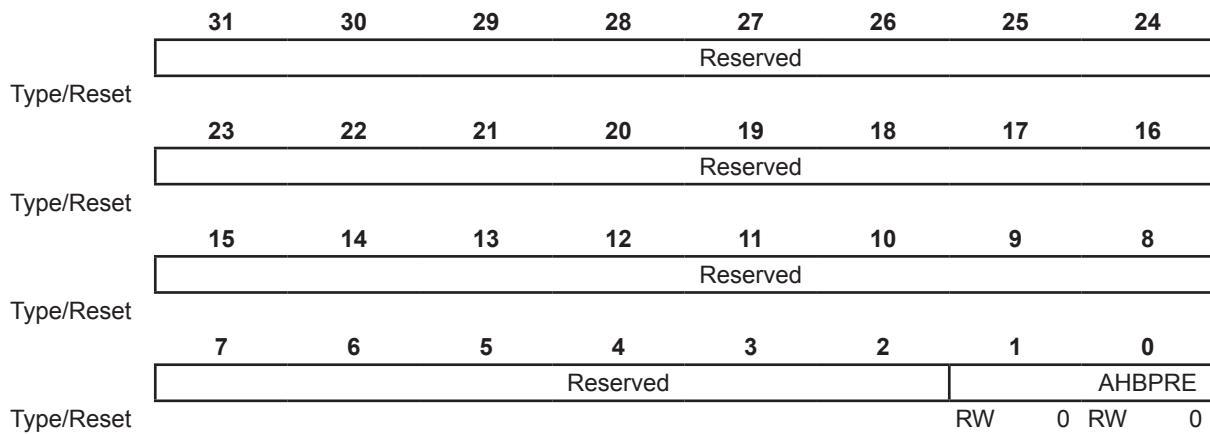
Bits	Field	Descriptions
[31]	PLLbps	PLL Bypass Mode Enable 0: Disable PLL Bypass mode 1: Enable PLL Bypass mode which acts FOUT = FIN

## AHB Configuration Register – AHBCFGR

This register specifies frequency of system clock.

Offset: 0x020

Reset value: 0x0000\_0000



Bits	Field	Descriptions
[1:0]	AHBPRE	<p>AHB Pre-Scalar</p> <p>Set and reset by software to control the division factor of the AHB clock.</p> <ul style="list-style-type: none"> <li>00: CK_AHB = CK_SYS</li> <li>01: CK_AHB = CK_SYS / 2</li> <li>10: CK_AHB = CK_SYS / 4</li> <li>11: CK_AHB = CK_SYS / 8</li> </ul>

## AHB Clock Control Register – AHBCCR

This register specifies clock enable bits of AHB.

Offset: 0x024

Reset value: 0x0000\_0005

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved				PDEN	PCEN	PBEN	PAEN	
	RW 0 RW 0 RW 0 RW 0 RW 0								
Type/Reset	15	14	13	12	11	10	9	8	
	Reserved		CRCEN	EBIEN	CKREFEN	USBEN	Reserved		
Type/Reset	RW 0 RW 0 RW 0 RW 0 RW 0								
Type/Reset	7	6	5	4	3	2	1	0	
	APB1EN	APB0EN	BMEN	PDMAEN	Reserved	SRAMEN	Reserved	FMCEN	
Type/Reset	RW	1	RW	1	RW	0	RW	1	

Bits	Field	Descriptions
[19]	PDEN	GPIO Port D Clock Enable Set and reset by software 0: Port D clock is disabled 1: Port D clock is enabled
[18]	PCEN	GPIO Port C Clock Enable Set and reset by software 0: Port C clock is disabled 1: Port C clock is enabled
[17]	PBEN	GPIO Port B Clock Enable Set and reset by software 0: Port B clock is disabled 1: Port B clock is enabled
[16]	PAEN	GPIO Port A Clock Enable Set and reset by software 0: Port A clock is disabled 1: Port A clock is enabled
[13]	CRCEN	CRC Module Clock Enable Set and reset by software. 0: CRC clock disable 1: CRC clock enable
[12]	EBIEN	EBI Module Clock Enable Set and reset by software. 0: EBI clock disable 1: EBI clock enable
[11]	CKREFEN	CK_REF Clock Enable Set and reset by software 0: CK_REF clock is disabled 1: CK_REF clock is enabled

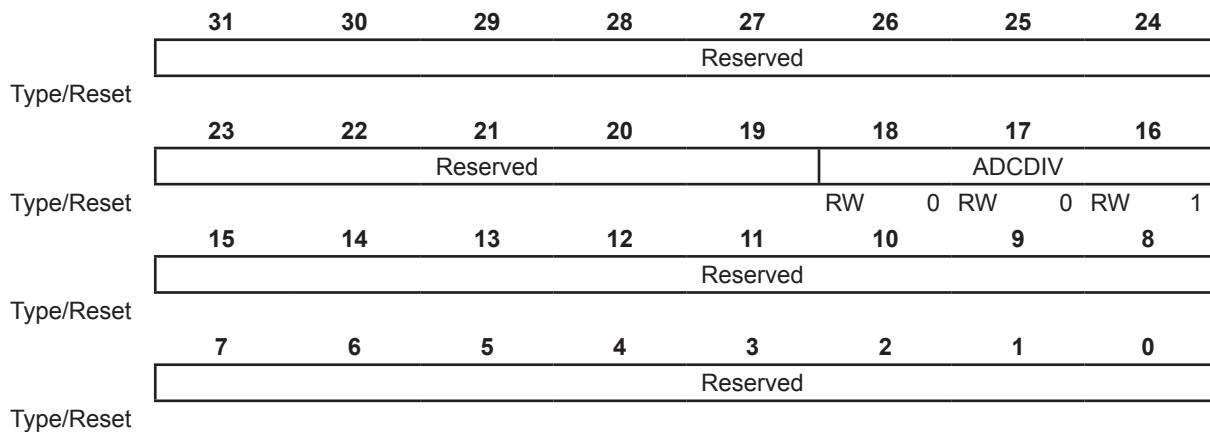
Bits	Field	Descriptions
[10]	USBEN	USB Clock Enable Set and reset by software 0: USB clock disabled 1: USB clock enabled
[7]	APB1EN	APB1 bridge Clock Enable Set and reset by software. User can set APB1EN as 0 to reduce power consumption if the APB1 bridge is unused during Sleep mode. 0: APB1 bridge clock auto disable by hardware during Sleep mode 1: APB1 bridge clock always enable during Sleep mode
[6]	APB0EN	APB0 bridge Clock Enable Set and reset by software. User can set APB0EN as 0 to reduce power consumption if the APB0 bridge is unused during Sleep mode. 0: APB0 bridge clock auto disable by hardware during Sleep mode 1: APB0 bridge clock always enable during Sleep mode
[5]	BMEN	Bus Matrix Clock Enable Set and reset by software. User can set BMEN as 0 to reduce power consumption if the bus matrix is unused during Sleep mode. 0: Bus Matrix clock auto disable by hardware during Sleep mode 1: Bus Matrix clock always enable during Sleep mode
[4]	PDMAEN	Peripheral DMA Clock Enable Set and reset by software. 0: PDMA clock disable 1: PDMA clock enable Note: The PDMA can independently operate when the processor is into the sleep mode. But the relative clock of AHB bus slave or peripherals has to be enabled.
[2]	SRAMEN	SRAM Clock Enable Set and reset by software. User can set SRAMEN as 0 to reduce power consumption if the SRAM is unused during Sleep mode. 0: SRAM clock auto disable by hardware during Sleep mode 1: SRAM clock always enable during Sleep mode
[0]	FMCEN	Flash Memory Controller Clock Enable Set and reset by software. User can set FMCEN as 0 to reduce power consumption if the Flash Memory is unused during Sleep mode. 0: FMC clock auto disable by hardware during Sleep mode 1: FMC clock always enable during Sleep mode

## APB Configuration Register – APBCFGR

This register specifies the frequency of ADC conversion clock.

Offset: 0x028

Reset value: 0x0001\_0000



Bits	Field	Descriptions
[18:16]	ADCDIV	ADC Clock Frequency Divide Selection Set and reset by software to control ADC conversion clock division factor. 000: reserved 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 / 6)

## APB Clock Control Register 0 – APBCCR0

This register specifies clock enable bits of APB peripherals.

Offset: 0x02C

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset				Reserved				
						RW	0	RW 0
	23	22	21	20	19	18	17	16
Type/Reset				Reserved				
						RW	0	RW 0
	15	14	13	12	11	10	9	8
Type/Reset	RW	0	RW 0		RW	0	RW 0	RW 0
	7	6	5	4	3	2	1	0
Type/Reset			Reserved	SPI1EN	SPI0EN		Reserved	I2C1EN I2C0EN
			RW	0	RW 0		RW	0 RW 0

Bits	Field	Descriptions
[25]	I2SEN	I <sup>2</sup> S Interface Clock Enable Set and reset by software. 0: I <sup>2</sup> S clock is disabled 1: I <sup>2</sup> S clock is enabled
[24]	SCIEN	Smart Card Interface Clock Enable Set and reset by software. 0: Smart Card clock is disabled 1: Smart Card clock is enabled
[15]	EXTIEN	External Interrupt Clock Enable Set and reset by software. 0: EXTI clock is disabled 1: EXTI clock is enabled
[14]	AFIOEN	Alternate Function I/O Clock Enable Set and reset by software. 0: AFIO clock is disabled 1: AFIO clock is enabled
[11]	UR1EN	UART1 Clock Enable Set and reset by software. 0: UART1 clock is disabled 1: UART1 clock is enabled
[10]	UR0EN	UART0 Clock Enable Set and reset by software. 0: UART0 clock is disabled 1: UART0 clock is enabled
[9]	USR1EN	USART1 Clock Enable Set and reset by software. 0: USART1 clock is disabled 1: USART1 clock is enabled

<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>
[8]	USR0EN	USART0 Clock Enable Set and reset by software. 0: USART0 clock is disabled 1: USART0 clock is enabled
[5]	SPI1EN	SPI1 Clock Enable Set and reset by software. 0: SPI1 clock is disabled 1: SPI1 clock is enabled
[4]	SPI0EN	SPI0 Clock Enable Set and reset by software. 0: SPI0 clock is disabled 1: SPI0 clock is enabled
[1]	I2C1EN	I <sup>2</sup> C1 Clock Enable Set and reset by software. 0: I <sup>2</sup> C1 clock is disabled 1: I <sup>2</sup> C1 clock is enabled
[0]	I2C0EN	I <sup>2</sup> C0 Clock Enable Set and reset by software. 0: I <sup>2</sup> C0 clock is disabled 1: I <sup>2</sup> C0 clock is enabled

## APB Clock Control Register 1 – APBCCR1

This register specifies clock enable bits APB peripherals.

Offset: 0x030

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved	CMPEN		Reserved		BFTM1EN	BFTM0EN	
	RW	0				RW	0	RW
Type/Reset	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset	RW	0				RW	0	RW
	7	6	5	4	3	2	1	0
Type/Reset	Reserved	BKPREN	Reserved	WDTREN		Reserved	MCTM1EN	MCTM0EN
	RW	0		RW	0		RW	0

Bits	Field	Descriptions
[24]	ADCCEN	ADC Controller Clock Enable Set and reset by software. 0: ADC clock is disabled 1: ADC clock is enabled
[22]	CMPEN	CMP Clock Enable Set and reset by software. 0: CMP clock is disabled 1: CMP clock is enabled
[17]	BFTM1EN	BFTM1 Clock Enable Set and reset by software. 0: BFTM1 clock is disabled 1: BFTM1 clock is enabled
[16]	BFTM0EN	BFTM0 Clock Enable Set and reset by software. 0: BFTM0 clock is disabled 1: BFTM0 clock is enabled
[9]	GPTM1EN	GPTM1 Clock Enable Set and reset by software. 0: GPTM1 clock is disabled 1: GPTM1 clock is enabled
[8]	GPTM0EN	GPTM0 Clock Enable Set and reset by software. 0: GPTM0 clock is disabled 1: GPTM0 clock is enabled
[6]	BKPREN	Backup Domain Clock Enable for Registers Access Set and reset by software. 0: RTC clock is disabled 1: RTC clock is enabled

<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>
[4]	WDTREN	Watchdog Timer Clock Enable for Registers Access Set and reset by software. 0: Watchdog Timer clock is disabled 1: Watchdog Timer clock is enabled
[1]	MCTM1EN	MCTM1 Clock Enable Set and reset by software. 0: MCTM1 clock is disabled 1: MCTM1 clock is enabled
[0]	MCTM0EN	MCTM0 Clock Enable Set and reset by software. 0: MCTM0 clock is disabled 1: MCTM0 clock is enabled

## Clock Source Status Register – CKST

This register specifies status of clock source.

Offset: 0x034

Reset value: 0xC100\_0000

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

Bits	Field	Descriptions
[31:30]	CKSWST	Clock Switch Status 0x: CK_PLL clock out as system clock 10: CK_HSE as system clock 11: CK_HSI as system clock
[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

## APB Peripheral Clock Selection Register 0 – APBPCSR0

This register specifies APB peripheral clock prescaler selection.

Offset: 0x038

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset	RW	0	RW	0	RW	0	RW	0
	UR1PCLK		UR0PCLK		USR1PCLK		USR0PCLK	
	23	22	21	20	19	18	17	16
Type/Reset	RW	0	RW	0	RW	0	RW	0
	GPTM1PCLK		GPTM0PCLK		MCTM1PCLK		MCTM0PCLK	
	15	14	13	12	11	10	9	8
Type/Reset	RW	0	RW	0	RW	0		
	BFTM1PCLK		BFTM0PCLK				Reserved	
	7	6	5	4	3	2	1	0
Type/Reset	RW	0	RW	0	RW	0	RW	0
	SPI1PCLK		SPI0PCLK		I2C1PCLK		I2C0PCLK	

Bits	Field	Descriptions
[31:30]	UR1PCLK	UART1 Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[29:28]	UR0PCLK	UART0 Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[27:26]	USR1PCLK	USART1 Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[25:24]	USR0PCLK	USART0 Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[23:22]	GPTM1PCLK	GPTM1 Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock

Bits	Field	Descriptions
[21:20]	GPTM0PCLK	GPTM0 Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[19:18]	MCTM1PCLK	MCTM1 Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[17:16]	MCTM0PCLK	MCTM0 Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[15:14]	BFTM1PCLK	BFTM1 Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB/2 10: PCLK = CK_AHB/4 11: PCLK = CK_AHB/8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[13:12]	BFTM0PCLK	BFTM0 Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[7:6]	SPI1PCLK	SPI1 Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[5:4]	SPI0PCLK	SPI0 Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[3:2]	I2C1PCLK	I <sup>2</sup> C1 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	I <sup>2</sup> C0 Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock

## APB Peripheral Clock Selection Register 1 – APBPCSR1

This register specifies APB peripheral clock prescaler selection.

Offset: 0x03C

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[21:20]	I2SPCLK	I <sup>2</sup> S Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[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
[15:14]	BKPRCLK	Backup Domain Register Access Clock Selection 00: PCLK = CK_AHB / 4 01: PCLK = CK_AHB / 8 10: PCLK = CK_AHB / 16 11: PCLK = CK_AHB / 32 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[13:12]	WDTRPCLK	WDT Register Access Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[9:8]	CMPPCLK	CMP Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock

<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>
[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 to control the frequency trimming of HSI RC oscillation.

Offset: 0x040

Reset value: 0xXXXX\_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	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	ATCEN	TRIMEN
	RO	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[28:24]	HSICOARSE	HSI Clock Coarse Trimming Value These bits are initialized automatically at startup. They are adjusted by factory trimming and can not trim by program.
[23:16]	HSIFINE	HSI Clock Fine Trimming Value These bits are initialized automatically at startup. They are also adjusted by factory trimming. But these bits provide an additional user-programmable trimming value that is added to the HSICOARSE[4:0] bits to get more accurate or compensate the variations in voltage and temperature that influence the frequency of the HSI. It can be programmed by software or Auto Trimming Controller (ATC) with external reference clock.
[7]	FLOCK	Frequency Lock 0: HSI frequency is not trimmed into target range 1: HSI frequency is trimmed into target range
[5]	REFCLKSEL	Reference Clock Selection 0: Select 32.768KHz external low speed clock source (LSE) 1: Select 1KHz USB frame pulse This bit is selected the reference clock for the HSI Auto Trimming Controller.
[4]	TMSEL	Trimming Mode Selection 0: Automatic by Auto Trimming Controller 1: Manual by user program This bit is selected the HSI RC oscillator trimming function by ATC hardware or user programming via the HSIFINE[7:0] bits in the HSI Control Register.
[1]	ATCEN	ATC Enable 0: Disable Auto Trimming Controller 1: Enable Auto Trimming Controller

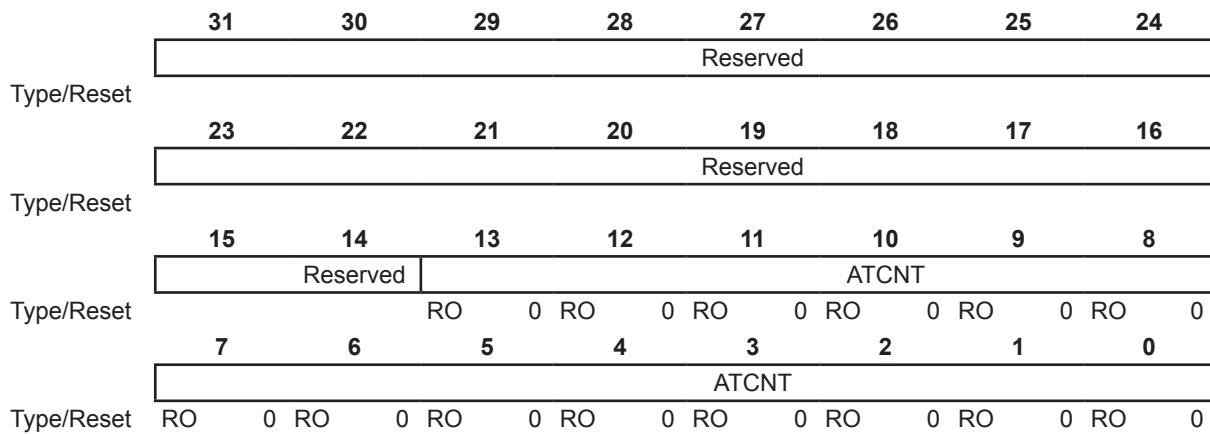
<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>
[0]	TRIMEN	<p>Trimming Enable</p> <p>0: HSI Trimming is disable 1: HSI Trimming is enable</p> <p>The bit enables the HSI RC oscillator trimming function by ATC hardware or user programming.</p>

## HSI Auto Trimming Counter Register – HSIATCR

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

Offset: 0x044

Reset value: 0x0000\_0000



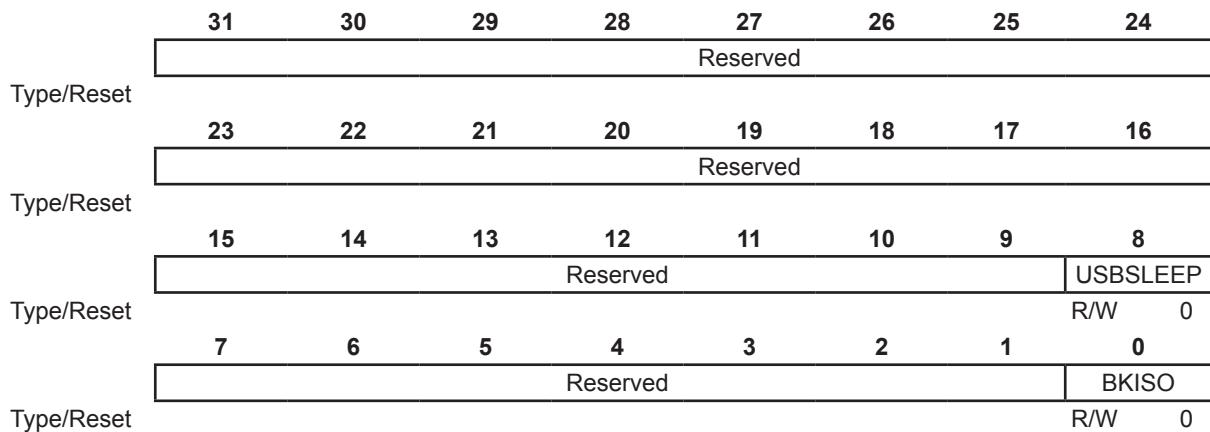
Bits	Field	Descriptions
[13:0]	ATCNT	Auto Trimming Counter This bits contains the counter value of the HSI auto trimming controller

## Low Power Control Register – LPCR

This register specifies low power control.

Offset: 0x300

Reset value: 0x0000\_0000



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

## MCU Debug Control Register – MCUDBGCR

This register specifies debug control of MCU.

Offset: 0x304

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	R/W	0	R/W	0	R/W	0	R/W	0	
Type/Reset	15	14	13	12	11	10	9	8	
	DBSCI	DBDSDLP2	DBI2C1	DBI2C0	DBSPI1	DBSPI0	DBUSR1	DBUSR0	
Type/Reset	R/W	0	R/W	0	R/W	0	R/W	0	
	7	6	5	4	3	2	1	0	
Type/Reset	DBGPTM1	DBGPTM0	DBMCTM1	DBMCTM0	DBWDT	DBPD	DBDSDLP1	DBSLP	
	R/W	0	R/W	0	R/W	0	R/W	0	

Bits	Field	Descriptions
[20]	DBTRACE	TRACESWO Debug Mode Enable Set and reset by software 0: Disable TRACESWO output 1: Enable TRACESWO output
[19]	DBUR1	UART1 Debug Mode Enable Timeout mode of USART3 is stopped when the core is halted. Set and reset by software. 0: Same behavior as in normal mode 1: USART3 FIFO timeout is frozen
[18]	DBUR0	UART0 Debug Mode Enable Timeout mode of USART2 is stopped when the core is halted. Set and reset by software. 0: Same behavior as in normal mode 1: USART2 FIFO timeout is frozen
[17]	DBBFTM1	BFTM1 Debug Mode Enable Counter of BFTM1 is stopped when the core is halted. Set and reset by software. 0: BFTM1 counter continues even if the core is halted 1: BFTM1 counter is stopped when the core is halted
[16]	DBBFTM0	BFTM0 Debug Mode Enable Counter of BFTM0 is stopped when core is halted. Set and reset by software. 0: BFTM0 counter continues even if the core is halted 1: BFTM0 counter is stopped when the core is halted
[15]	DBSCI	SCI Debug Mode Enable Timeout mode of SCI is stopped when the core is halted. Set and reset by software. 0: Same behavior as in normal mode 1: SCI timeout is frozen

Bits	Field	Descriptions
[14]	DBDSL2	Debug Deep-Sleep2 Set and reset by software. 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
[13]	DBI2C1	I <sup>2</sup> C1 Debug Mode Enable Timeout mode of I <sup>2</sup> C1 is stopped when the core is halted. Set and reset by software. 0: Same behavior as in normal mode 1: I <sup>2</sup> C1 timeout is frozen
[12]	DBI2C0	I <sup>2</sup> C0 Debug Mode Enable Timeout mode of I <sup>2</sup> C0 is stopped when the core is halted. Set and reset by software. 0: Same behavior as in normal mode 1: I <sup>2</sup> C0 timeout is frozen
[11]	DBSPI1	SPI1 Debug Mode Enable Timeout mode of SPI1 is stopped when the core is halted. Set and reset by software. 0: Same behavior as in normal mode 1: SPI1 FIFO timeout is frozen
[10]	DBSPI0	SPI0 Debug Mode Enable Timeout mode of SPI0 is stopped when the core is halted. Set and reset by software. 0: Same behavior as in normal mode 1: SPI0 FIFO timeout is frozen
[9]	DBUSR1	USART1 Debug Mode Enable Timeout mode of USART1 is stopped when the core is halted. Set and reset by software. 0: Same behavior as in normal mode 1: USART1 FIFO timeout is frozen
[8]	DBUSR0	USART0 Debug Mode Enable Timeout mode of USART0 is stopped when the core is halted. Set and reset by software. 0: Same behavior as in normal mode 1: USART0 FIFO timeout is frozen
[7]	DBGPTM1	GPTM1 Debug Mode Enable Counter of GPTM1 is stopped when the core is halted. Set and reset by software. 0: GPTM1 counter continues even if the core is halted 1: GPTM1 counter is stopped when the core is halted
[6]	DBGPTM0	GPTM0 Debug Mode Enable Counter of GPTM0 is stopped when core is halted. Set and reset by software. 0: GPTM0 counter continues even if the core is halted 1: GPTM0 counter is stopped when the core is halted
[5]	DBMCTM1	MCTM1 Debug Mode Enable Counter of MCTM1 is stopped when core is halted. Set and reset by software. 0: MCTM1 counter continues even if the core is halted 1: MCTM1 counter is stopped when the core is halted
[4]	DBMCTM0	MCTM0 Debug Mode Enable Counter of MCTM0 is stopped when core is halted. Set and reset by software. 0: MCTM0 counter continues even if the core is halted 1: MCTM0 counter is stopped when the core is halted

Bits	Field	Descriptions
[3]	DBWDT	Watchdog Timer Debug Mode Enable Counter of Watchdog Timer is stopped when the core is halted. Set and reset by software. 0: Watchdog Timer counter continues even if the core is halted 1: Watchdog Timer counter is stopped when the core is halted
[2]	DBPD	Debug Power-Down Mode Set and reset by software. 0: LDO = Off, FCLK = Off, and HCLK = Off in Power-Down mode 1: LDO = On, FCLK = On, and HCLK = On in Power-Down mode
[1]	DBDSLP1	Debug Deep-Sleep1 Set and reset by software. 0: LDO = Low power mode, FCLK = Off, and HCLK = Off in Deep-Sleep1 1: LDO = On, FCLK = On, and HCLK = On in Deep-Sleep1
[0]	DBSLP	Debug Sleep Mode Set and reset by software. 0: LDO = On, FCLK = On, and HCLK = Off in Sleep mode 1: LDO = On, FCLK = On, and HCLK = On in Sleep mode

# 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

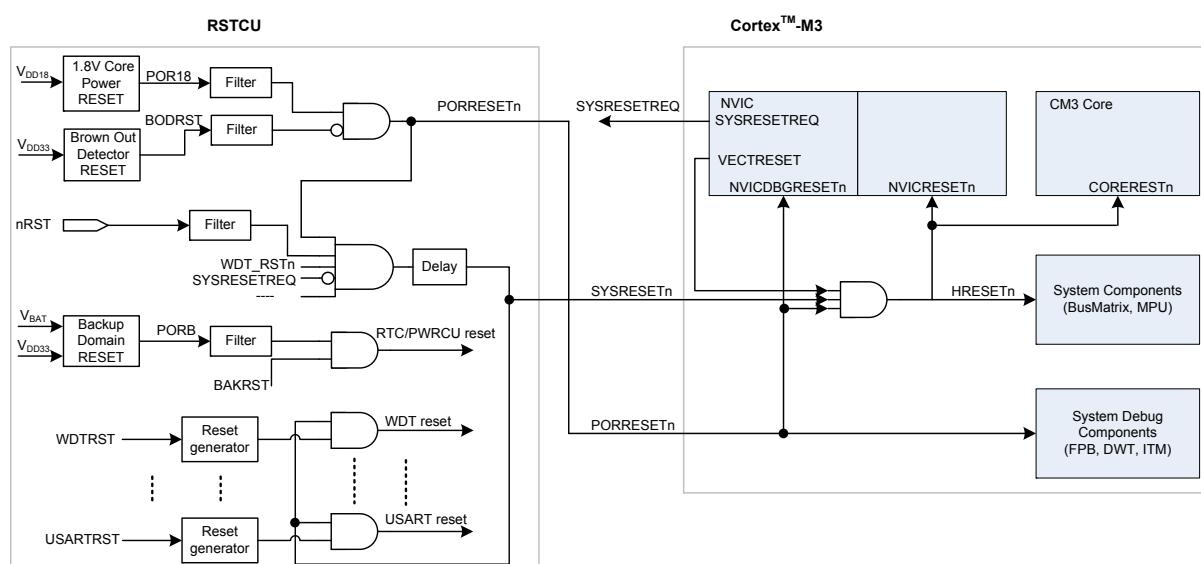
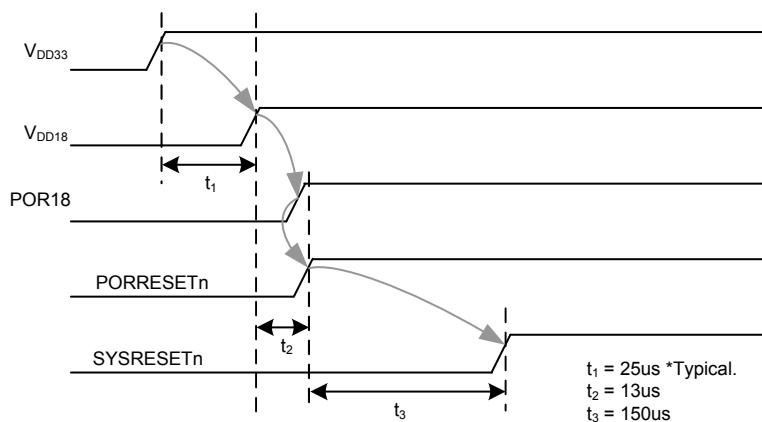


Figure 17. RSTCU Block Diagram

## 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 18, the POR18 active low signal will be de-asserted when the internal LDO voltage regulator is ready to provide 1.8 V power. In addition to the POR18 signal, the Power Control Unit, PWRCU, will assert the BODF signal as a Power Down Reset, PDR, when the BODEN bit in the LVDCSR register is set and the brown-out event occurs. For more details about the PWRCU function, refer to the PWRCU chapter.



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

Figure 18. 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 and VECTRESET events, refer to the related chapter in the Cortex™-M3 reference manual.

### AHB and APB Unit Reset

The AHB and APB unit reset can be divided into hardware and software resets. A hardware reset can be generated by either power on reset or system reset for all AHB and APB units. Each functional IP connected to the AHB and APB buses can be reset individually through the associated software reset bits in the RSTCU. For example, the application software can generate a USART0 reset via the USR0RST bit in the APBPRSTR0 register.

## Register Map

The following table shows the RSTCU registers and reset values.

**Table 19. RSTCU Register Map**

Register	Offset	Description	Reset Value
<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

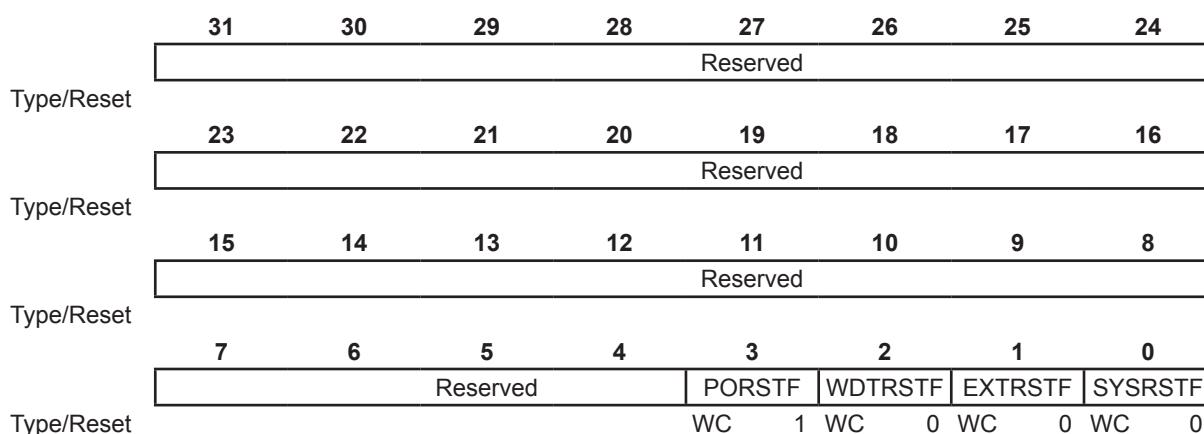
## Register Descriptions

### 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 1.8V 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 reset 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]	SYSRSTF	<p>System Reset Flag</p> <p>0: No NVIC system reset occurred 1: NVIC system reset occurred</p> <p>This bit is set by hardware when a system reset occurs and reset by writing 1 into it or by hardware when a power on reset occurs.</p>

## AHB Peripheral Reset Register – AHBPRSTR

This register specifies several AHB peripherals software reset control bits.

Offset: 0x104

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
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					RW	0	RW	0	RW
PDRST PCRST PBRST PARST									
Type/Reset	7	6	5	4	3	2	1	0	
CRCRST EBIRST USBRST Reserved DMARST									
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW

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

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

## APB Peripheral Reset Register 0 – APBPRSTR0

This register specifies several APB peripherals software reset control bits.

Offset: 0x108

Reset value: 0x0000\_0000

Type/Reset	31	30	29	28	27	26	25	24
	Reserved					I2SRST	SCIRST	
						RW	0	RW
Type/Reset	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset	15	14	13	12	11	10	9	8
Type/Reset	EXTIRST	AFIORST	Reserved		UR1RST	UR0RST	USR1RST	USR0RST
	RW	0	RW	0	RW	0	RW	0
Type/Reset	7	6	5	4	3	2	1	0
	Reserved		SPI1RST	SPI0RST	Reserved		I2C1RST	I2C0RST
			RW	0	RW	0	RW	0

Bits	Field	Descriptions
[25]	I2SRST	I <sup>2</sup> S Reset Control 0: No reset 1: Reset I <sup>2</sup> S This bit is set by software and cleared to 0 by hardware automatically.
[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.
[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.

Bits	Field	Descriptions
[14]	AFIORST	Alternate Function I/O Reset Control 0: No reset 1: Reset Alternate Function I/O This bit is set by software and cleared to 0 by hardware automatically.
[11]	UR1RST	UART1 Reset Control 0: No reset 1: Reset UART1 This bit is set by software and cleared to 0 by hardware automatically.
[10]	UR0RST	UART0 Reset Control 0: No reset 1: Reset UART0 This bit is set by software and cleared to 0 by hardware automatically.
[9]	USR1RST	USART1 Reset Control 0: No reset 1: Reset USART1 This bit is set by software and cleared to 0 by hardware automatically.
[8]	USR0RST	USART0 Reset Control 0: No reset 1: Reset USART0 This bit is set by software and cleared to 0 by hardware automatically.
[5]	SPI1RST	SPI1 Reset Control 0: No reset 1: Reset SPI1 This bit is set by software and cleared to 0 by hardware automatically.
[4]	SPI0RST	SPI0 Reset Control 0: No reset 1: Reset SPI0 This bit is set by software and cleared to 0 by hardware automatically.
[1]	I2C1RST	I <sup>2</sup> C1 Reset Control 0: No reset 1: Reset I <sup>2</sup> C1 This bit is set by software and cleared to 0 by hardware automatically.
[0]	I2C0RST	I <sup>2</sup> C0 Reset Control 0: No reset 1: Reset I <sup>2</sup> C0 This bit is set by software and cleared to 0 by hardware automatically.

## APB Peripheral Reset Register 1 – APBPRSTR1

This register specifies several APB peripherals software reset control bits.

Offset: 0x10C

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved	CMPRST		Reserved			BFTM1RST	BFTM0RST
	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		WDTRST		Reserved		MCTM1RST	MCTM0RST
Type/Reset							RW	0
							RW	0

Bits	Field	Descriptions
[24]	ADCRST	A/D Converter Reset Control 0: No reset 1: Reset A/D Converter This bit is set by software and cleared to 0 by hardware automatically.
[22]	CMPRST	Comparator Controller Reset Control 0: No reset 1: Reset CMP0 and CMP1 This bit is set by software and cleared to 0 by hardware automatically.
[17]	BFTM1RST	BFTM1 Reset Control 0: No reset 1: Reset BFTM1 This bit is set by software and cleared to 0 by hardware automatically.
[16]	BFTM0RST	BFTM0 Reset Control 0: No reset 1: Reset BFTM0 This bit is set by software and cleared to 0 by hardware automatically.
[9]	GPTM1RST	GPTM1 Reset Control 0: No reset 1: Reset GPTM1 This bit is set by software and cleared to 0 by hardware automatically.
[8]	GPTM0RST	GPTM0 Reset Control 0: No reset 1: Reset GPTM0 This bit is set by software and cleared to 0 by hardware automatically.
[4]	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.

<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>
[1]	MCTM1RST	MCTM1 Reset Control 0: No reset 1: Reset MCTM1 This bit is set by software and cleared to 0 by hardware automatically.
[0]	MCTM0RST	MCTM0 Reset Control 0: No reset 1: Reset MCTM0 This bit is set by software and cleared to 0 by hardware automatically.

## 8 General Purpose I/O (GPIO)

### Introduction

There are up to 51 General Purpose I/O port, GPIO, named PA0~PA15, PB0~PB15, PC0~PC15 and PD0~PD2 for the device to implement the logic input/output functions. Each of the GPIO ports has related control and configuration registers to satisfy the requirement of specific applications.

The 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).

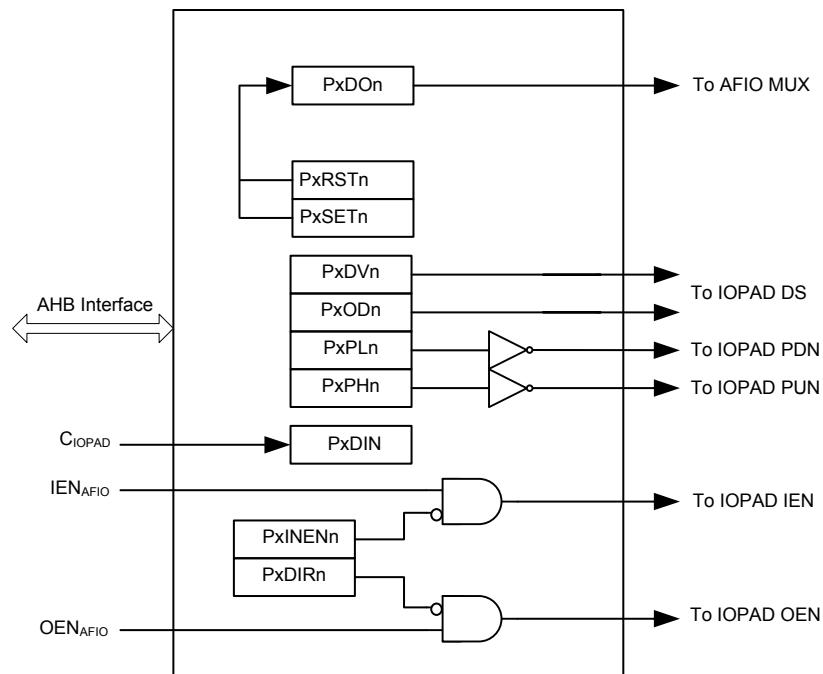


Figure 19. GPIO Block Diagram

## Features

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

## Functional Descriptions

### Default GPIO Pin Configuration

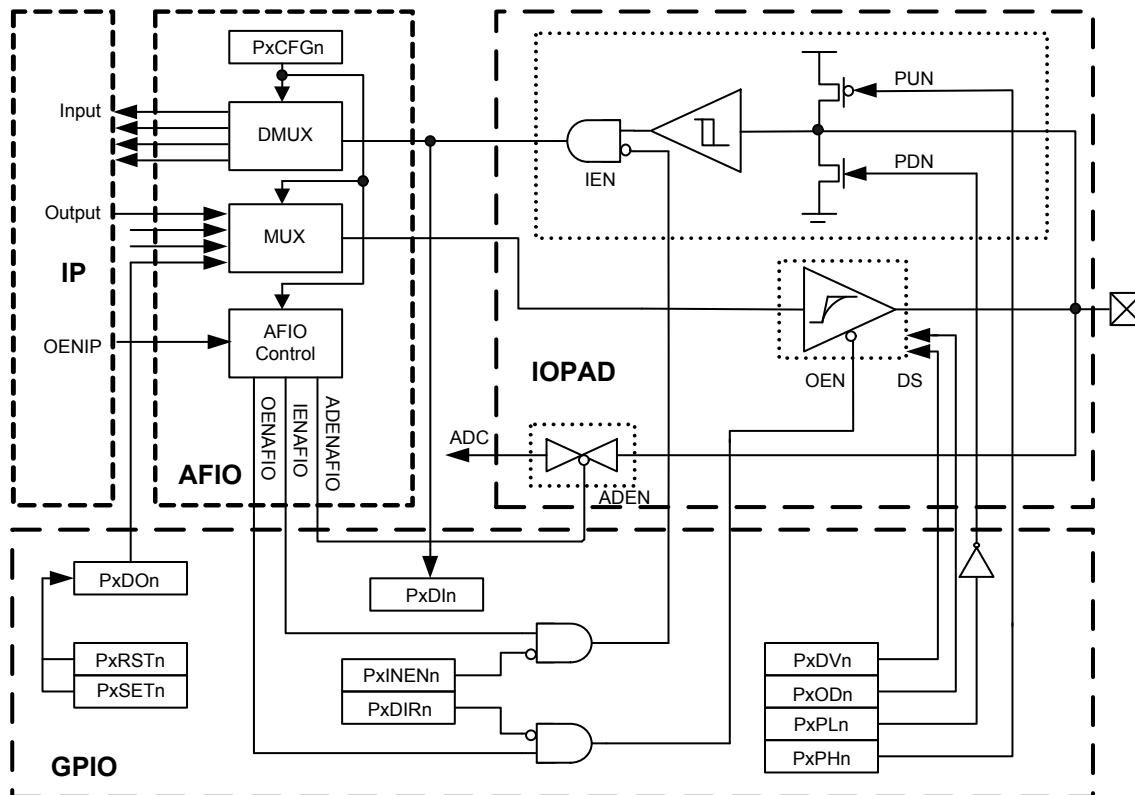
During or just after the reset period, the alternative functions are all inactive and the GPIO ports are configured into the input disable floating mode, i.e. input disabled without pull-up/pull-down resistors. Only the boot and Serial-Wired Debug pins which are pin-shared with the I/O pins PA8, PA9, PA12 and PA13 are active after a device reset.

- PA8/BOOT0: Input enable with internal pull-up
- PA9/BOOT1: Input enable with internal pull-up
- PA12/SWCLK: Input enable with internal pull-up
- PA13/SWDIO: Input enable with internal pull-up

### General Purpose I/O – GPIO

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

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



**Figure 20. AFIO(GPIO Control Signal**

PxDIn/PxDOOn ( $x = A \sim D$ ): Data Input/Data Output

PxRSTn/PxSETn ( $x = A \sim D$ ): Reset/Set

PxDIRn ( $x = A \sim D$ ): Direction

PxINENn ( $x = A \sim D$ ): Input Enable

PxDVn ( $x = A \sim D$ ): Output Drive

PXODn ( $x = A \sim D$ ): Open Drain

PxPLn/PxPHn ( $x = A \sim D$ ): Pull Low/High

PxCFGn ( $x = A \sim D$ ): AFIO Configuration

**Table 20. AFIO, GPIO and IO Pad Control Signal True Table**

Type	AFIO			GPIO		PAD		
	$ADEN_{AFIO}$	$OEN_{AFIO}$	$IEN_{AFIO}$	$PxDIRn$	$PxINENn$	$ADEN$	$OEN$	$IEN$
GPIO Input <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 ( $x = A \sim D$ ) registers are used to lock the port  $x$  and lock control options. The value 0x5FA0 is written into the PxLKEY field in the PxLOCKR registers to freeze the PxDIRCR, PxINER, PxPUR, PxPDR, PxODR, PxDRVVR control and AFIO mode configuration (GPxCFGHR or GPxCFGLR, where  $x = A \sim D$ ). If the value in the PxLOCKR is 0x5FA0\_0001, it means that the Port  $x$  Lock function is enabled and the Port  $x$  pin 0 is frozen.

## Register Map

The following table shows the GPIO registers and reset values of the Port A ~ D.

**Table 21. GPIO Register Map**

Register	Offset	Description	Reset Value
<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_0300
PAPUR	0x008	Port A Pull-Up Selection Register	0x0000_3300
PAPDR	0x00C	Port A Pull-Down Selection Register	0x0000_0000
PAODR	0x010	Port A Open Drain Selection Register	0x0000_0000
PADRVR	0x014	Port A Drive Current Selection Register	0x0000_0000
PALOCKR	0x018	Port A Lock Register	0x0000_0000
PADINR	0x01C	Port A Data Input Register	0x0000_3300
PADOUTR	0x020	Port A Data Output Register	0x0000_0000
PASRR	0x024	Port A Output Set and Reset Control Register	0x0000_0000
PARR	0x028	Port A Output Reset Control Register	0x0000_0000
<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
PBDRVVR	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
PCDRVVR	0x014	Port C Drive Current Selection Register	0x0000_0000
PCLOCKR	0x018	Port C Lock Register	0x0000_0000
PCDINR	0x01C	Port C Data Input Register	0x0000_0000

Register	Offset	Description	Reset Value
PCDOUTR	0x020	Port C Data Output Register	0x0000_0000
PCSRR	0x024	Port C Output Set and Reset Control Register	0x0000_0000
PCRR	0x028	Port C Output Reset Control Register	0x0000_0000
<b>GPIO D Base Address = 0x400B_6000</b>			
PDDIRCR	0x000	Port D Data Direction Control Register	0x0000_0000
PDINER	0x004	Port D Input Function Enable Control Register	0x0000_0000
PDPUR	0x008	Port D Pull-Up Selection Register	0x0000_0000
PDPDR	0x00C	Port D Pull-Down Selection Register	0x0000_0000
PDODR	0x010	Port D Open Drain Selection Register	0x0000_0000
PDLOCKR	0x018	Port D Lock Register	0x0000_0000
PDDINR	0x01C	Port D Data Input Register	0x0000_0000
PDDOUTR	0x020	Port D Data Output Register	0x0000_0000
PDSRR	0x024	Port D Output Set and Reset Control Register	0x0000_0000
PDRR	0x028	Port D Output Reset Control Register	0x0000_0000

## Register Descriptions

### Port A Data Direction Control Register – PADIRCR

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

Offset: 0x000

Reset value: 0x0000\_0000

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

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\_0300

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

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

## Port A Pull-Up Selection Register – PAPUR

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

Offset: 0x008

Reset value: 0x0000\_3300

	31	30	29	28	27	26	25	24
Reserved								
Type/Reset	23	22	21	20	19	18	17	16
Reserved								
Type/Reset	15	14	13	12	11	10	9	8
PAPU								
Type/Reset	RW	0	RW	0	RW	1	RW	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	<p>GPIO Port A 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 A Pull-Down Selection Register – PAPDR

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

Offset: 0x00C

Reset value: 0x0000\_0000

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

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

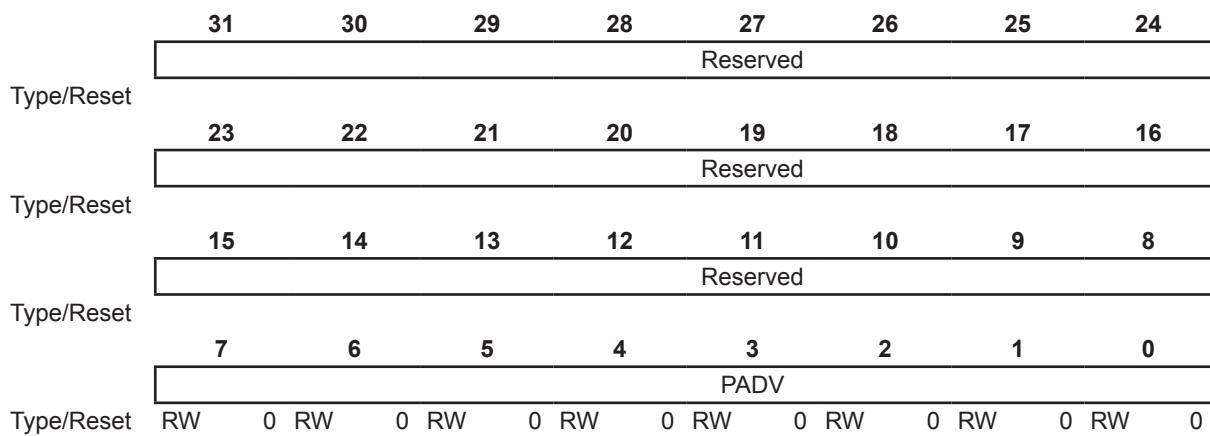
Bits	Field	Descriptions
[15:0]	PAODn	GPIO Port A pin n Open Drain Selection Control Bits (n = 0 ~ 15) 0: Pin n Open Drain output is disabled. (The output type is CMOS output) 1: Pin n Open Drain output is enabled. (The output type is open-drain output)

## Port A Output Current Drive Selection Register – PADRVR

This register specifies the GPIO Port A output driving current.

Offset: 0x014

Reset value: 0x0000\_0000



Bits	Field	Descriptions
[7:0]	PADVn	GPIO Port A pin n Output Current Drive Selection Control Bits (n = 0 ~ 7) 0: 4mA source/sink current 1: 8mA 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						
23 22 21 20 19 18 17 16								
PALKEY								
Type/Reset	RW	0						
15 14 13 12 11 10 9 8								
PALOCK								
Type/Reset	RW	0						
7 6 5 4 3 2 1 0								
PALOCK								
Type/Reset	RW	0						

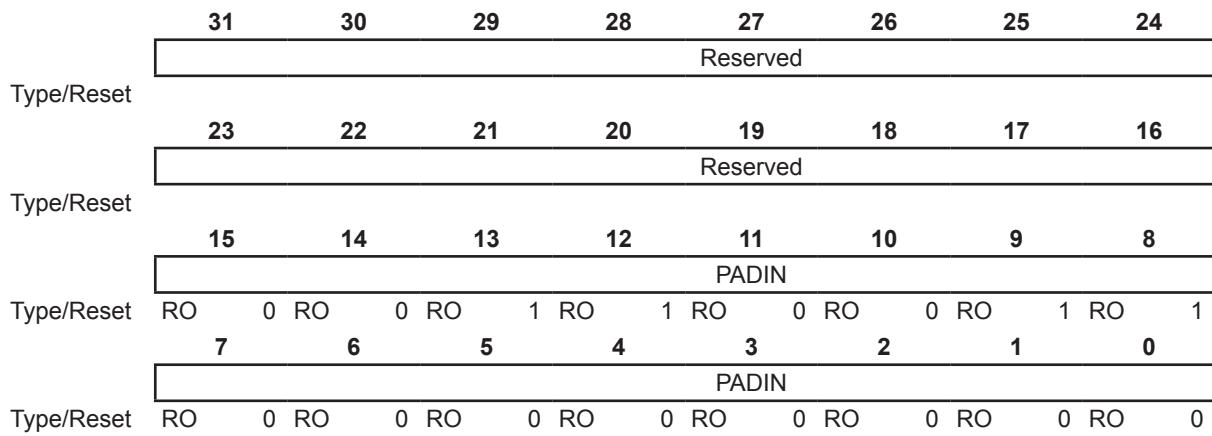
Bits	Field	Descriptions
[31:16]	PALKEY	<p>GPIO Port A Lock Key</p> <p>0x5FA0: Port A Lock function is enable</p> <p>Others: Port A Lock function is disable</p> <p>To lock the Port A function, a value 0x5FA0 should be written into the PALKEY field in this register. To execute a successful write operation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

## Port A Data Input Register – PADINR

This register specifies the GPIO Port A input data.

Offset: 0x01C

Reset value: 0x0000\_3300



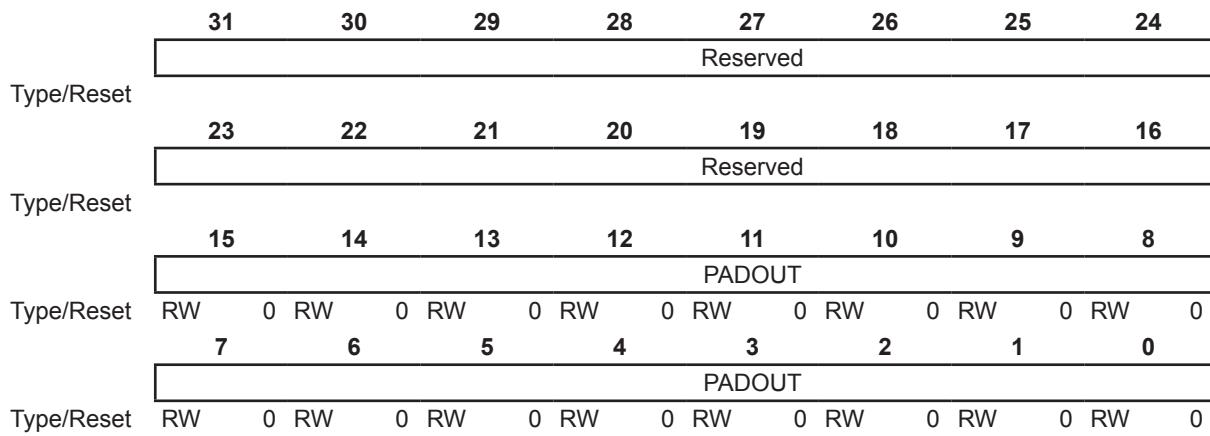
Bits	Field	Descriptions
[15:0]	PADINn	GPIO Port A pin n Data Input Bits (n = 0 ~ 15) 0: The input data of pin is 0 1: The input data of pin is 1

## Port A Output Data Register – PADOUTR

This register specifies the GPIO Port A output data.

Offset: 0x020

Reset value: 0x0000\_0000



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

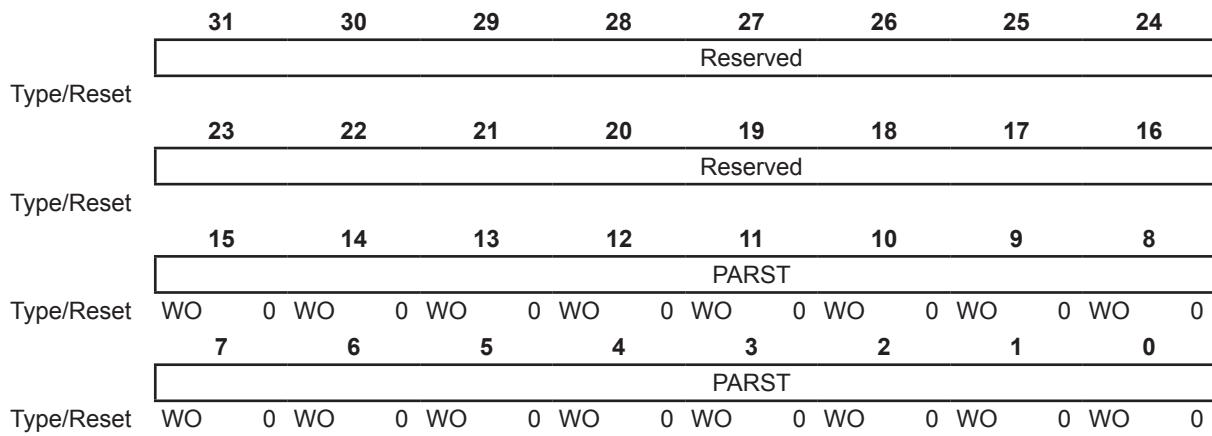
Bits	Field	Descriptions
[31:16]	PARSTn	GPIO Port A pin n Output Reset Control Bits (n = 0 ~ 15) 0: No effect on the PADOUTn bit 1: Reset the PADOUTn bit
[15:0]	PASETn	GPIO Port A pin n Output Set Control Bits (n = 0 ~ 15) 0: No effect on the PADOUTn bit 1: Set the PADOUTn bit Note that the function enabled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



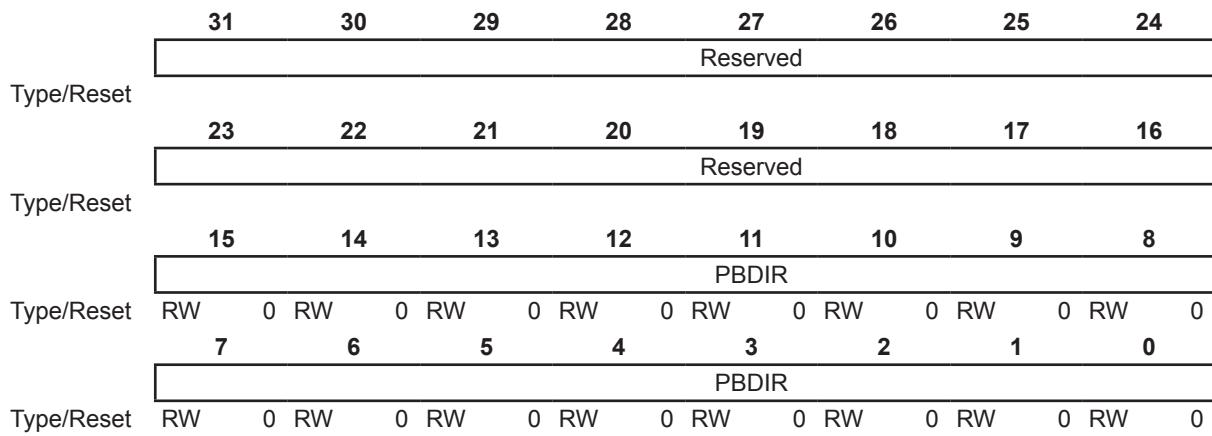
Bits	Field	Descriptions
[15:0]	PARSTn	GPIO Port A pin n Output Reset Bits (n = 0 ~ 15) 0: No effect on the PADOUTn bit 1: Reset the PADOUTn bit

## 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



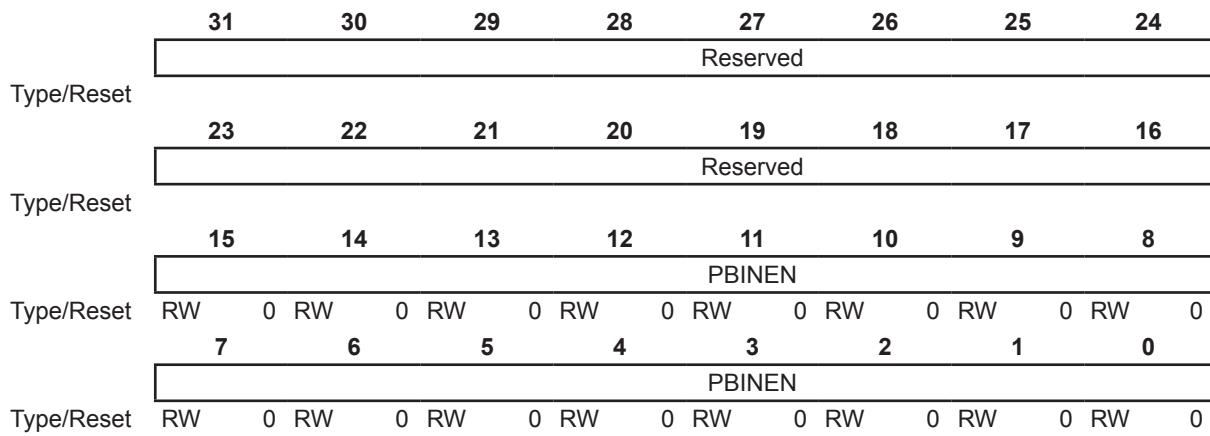
Bits	Field	Descriptions
[15:0]	PBDIRn	GPIO Port B pin n Direction Control Bits (n = 0 ~ 15) 0: Pin n is input mode 1: Pin n is output mode

## 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



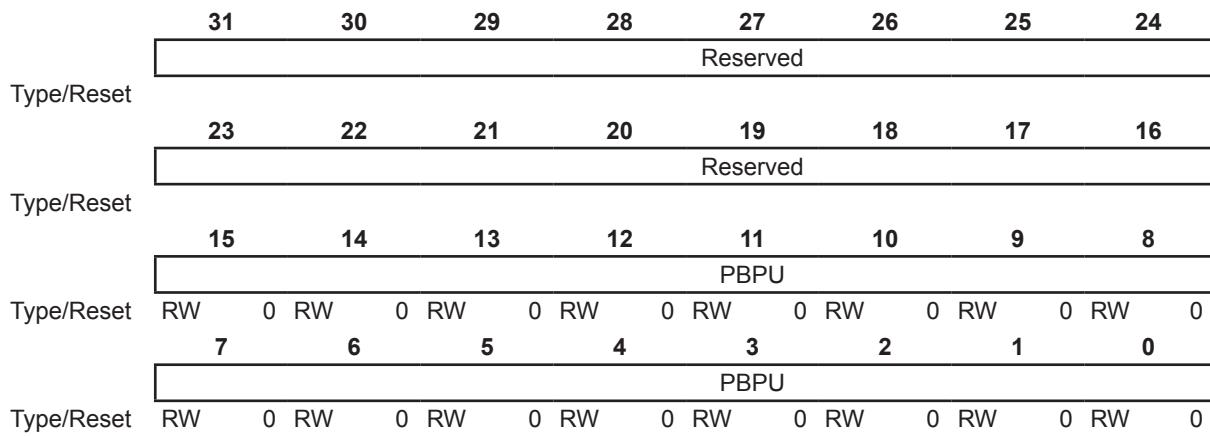
Bits	Field	Descriptions
[15:0]	PBINENn	GPIO Port B pin n Input Enable Control Bits (n = 0 ~ 15) 0: Pin n input function is disabled. 1: Pin n input function is enabled. When the pin n input function is disabled, the input Schmitt trigger will be turned off and the Schmitt trigger output will remain at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



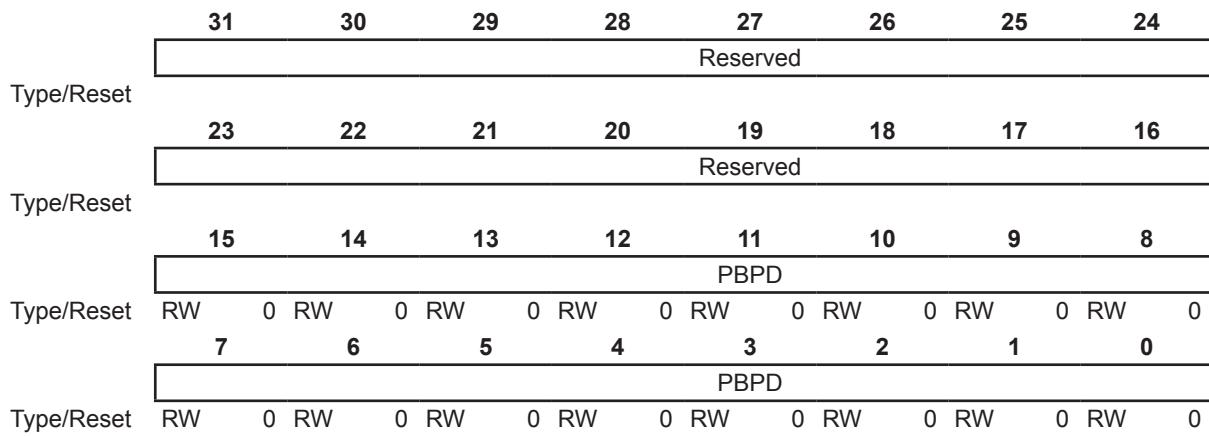
<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>
[15:0]	PBPUn	GPIO Port B pin n Pull-Up Selection Control Bits (n = 0 ~ 15) 0: Pin n pull-up function is disabled 1: Pin n pull-up function is enabled Note: When the pull-up and pull-down functions are both enabled, the pull-up function will have the higher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

## Port 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



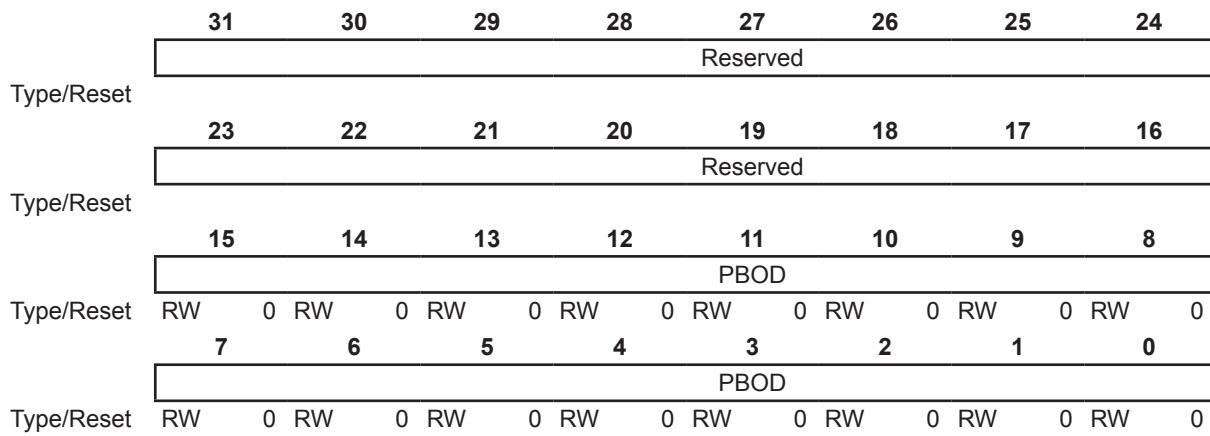
Bits	Field	Descriptions
[15:0]	PBPDn	<p>GPIO Port B pin n Pull-Down Selection Control Bits (n = 0 ~ 15)</p> <p>0: Pin n pull-down function is disabled 1: Pin n pull-down function is enabled</p> <p>Note: When the pull-up and pull-down functions are both enabled, the pull-up function will have the higher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

Bits	Field	Descriptions
[15:14]	PBDVn	GPIO Port B pin n Output Current Drive Selection Control Bits (n = 6 ~ 11, 14~15)
[11:6]		0: 4mA source/sink current 1: 8mA 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						
PBLKEY								
Type/Reset	RW	0						
PBLOCK								
Type/Reset	RW	0						
PBLOCK								
Type/Reset	RW	0						

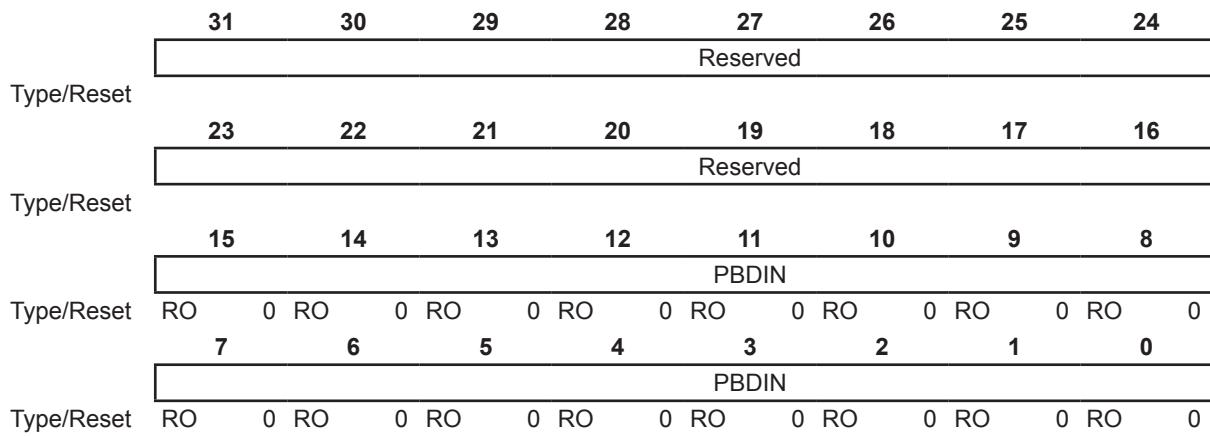
Bits	Field	Descriptions
[31:16]	PBLKEY	<p>GPIO Port Block Key</p> <p>0x5FA0: Port Block function is enable</p> <p>Others: Port B Lock function is disable</p> <p>To lock the Port B function, a value 0x5FA0 should be written into the PBLKEY field in this register. To execute a successful write operation 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</p> <p>1: Port B pin n is locked</p> <p>The PBLOCKn bits are used to lock the configurations of corresponding GPIO Pins when the correct Lock Key is applied to the PBLKEY field. The locked configurations including PBDIRn, PBINENn, PBPUn, PBPDn 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



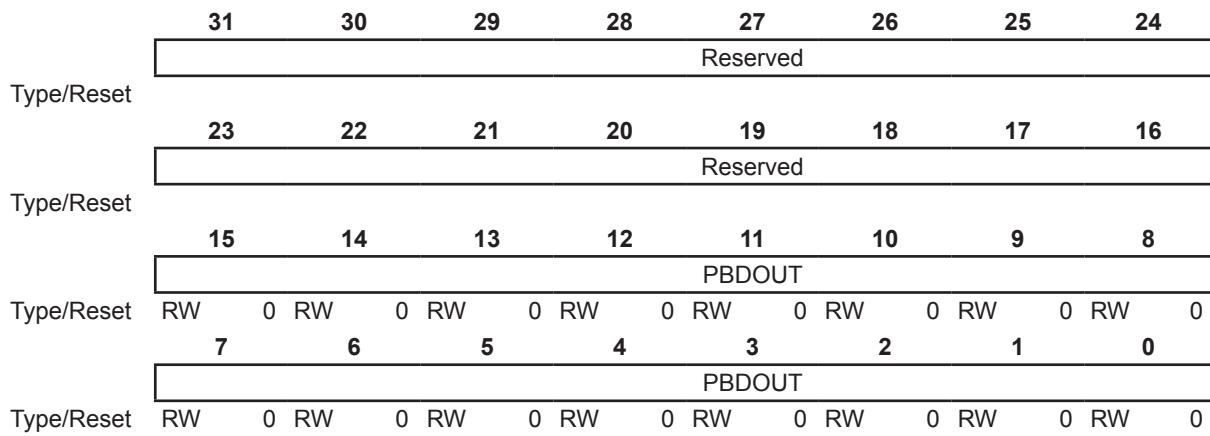
Bits	Field	Descriptions
[15:0]	PBDINn	GPIO Port B pin n Data Input Bits (n = 0 ~ 15) 0: The input data of corresponding pin is 0 1: The input data of corresponding pin is 1

## Port B Output Data Register – PBDOUTR

This register specifies the GPIO Port B output data.

Offset: 0x020

Reset value: 0x0000\_0000



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

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

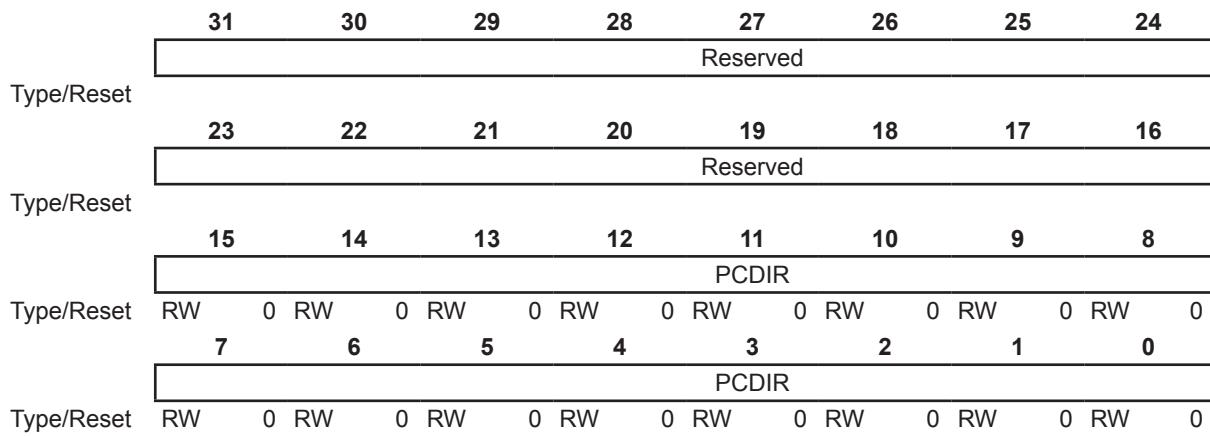
Bits	Field	Descriptions
[15:0]	PBRSTn	GPIO Port B pin n Output Reset Bits (n = 0 ~ 15) 0: No effect on the PBDOUTn bit 1: Reset the PBDOUTn bit

## 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
[15:0]	PCDIRn	GPIO Port C pin n Direction Control Bits (n = 0 ~ 15) 0: Pin n is input mode 1: Pin n is output mode

## Port C Input Function Enable Control Register – PCINER

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

Offset: 0x004

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[15:0]	PCINENn	<p>GPIO Port C pin n Input Enable Control Bits (n = 0 ~ 15)</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>

## Port C Pull-Up Selection Register – PCPUR

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

Offset: 0x008

Reset value: 0x0000\_0000

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

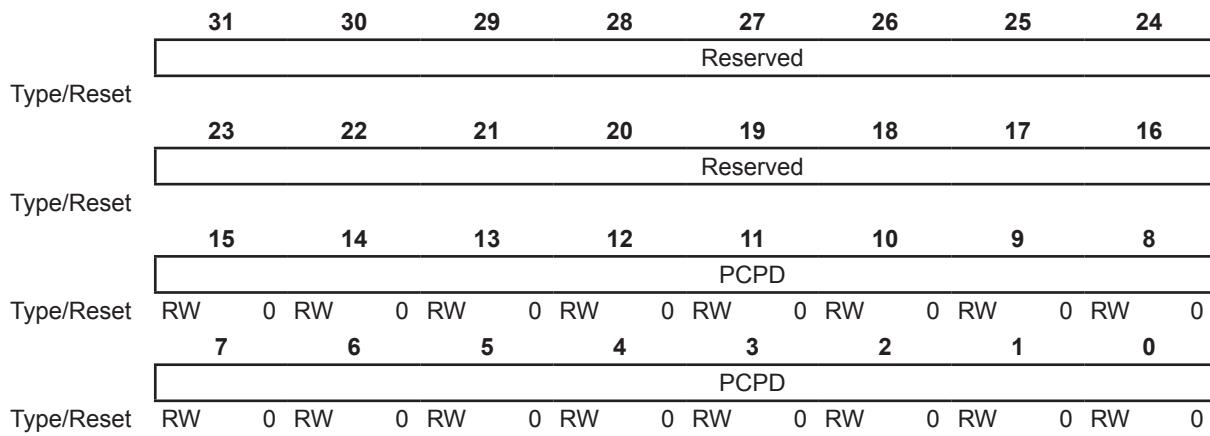
Bits	Field	Descriptions
[15:0]	PCPUn	<p>GPIO Port C pin n Pull-Up Selection Control Bits (n = 0 ~ 15)</p> <p>0: Pin n pull-up function is disabled 1: Pin n pull-up function is enabled</p> <p>Note: When the pull-up and pull-down functions are both enabled, the pull-up function will have the higher priority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

## Port 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



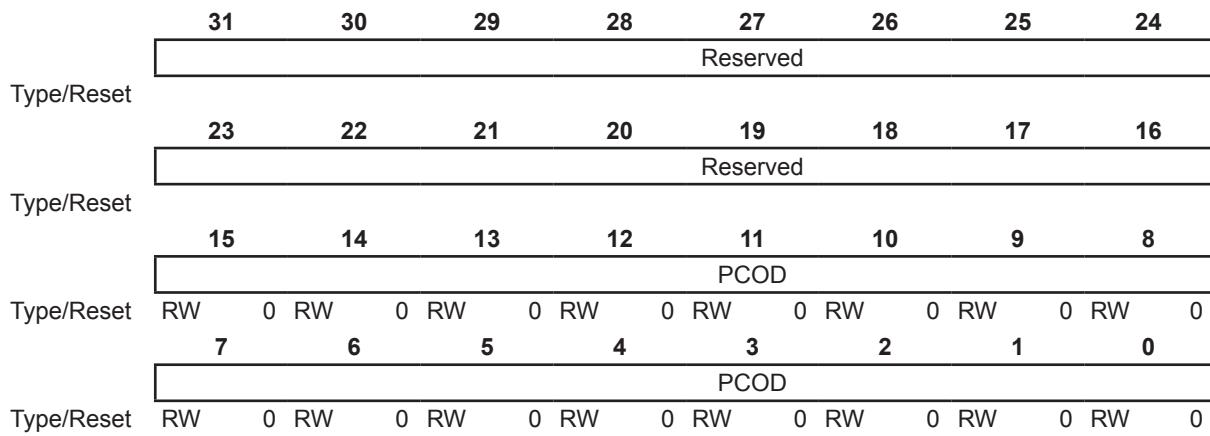
<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>
[15:0]	PCPDn	GPIO Port C pin n Pull-Down Selection Control Bits (n = 0 ~ 15) 0: Pin n pull-down function is disabled 1: Pin n pull-down function is enabled Note: When the pull-up and pull-down functions are both enabled, the pull-up function will have the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

## Port C Open Drain Selection Register – PCODR

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

Offset: 0x010

Reset value: 0x0000\_0000



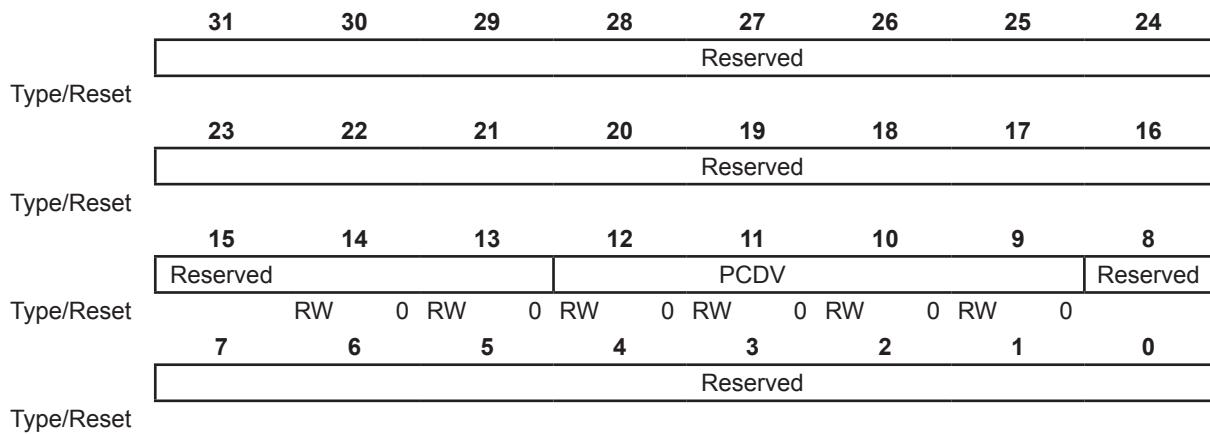
Bits	Field	Descriptions
[15:0]	PCODn	GPIO Port C pin n Open Drain Selection Control Bits (n = 0 ~ 15) 0: Pin n Open Drain output is disabled. (The output type is CMOS output) 1: Pin n Open Drain output is enabled. (The output type is open-drain output)

## Port C Output Current Drive Selection Register – PCDRVR

This register specifies the GPIO Port C output driving current.

Offset: 0x014

Reset value: 0x0000\_0000



Bits	Field	Descriptions
[12:9]	PCDVn	GPIO Port C pin n Output Current Drive Selection Control Bits (n = 9 ~ 12) 0: 4mA source/sink current 1: 8mA source/sink current

## Port C Lock Register – PCLOCKR

This register specifies the GPIO Port C lock configuration.

Offset: 0x018

Reset value: 0x0000\_0000

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

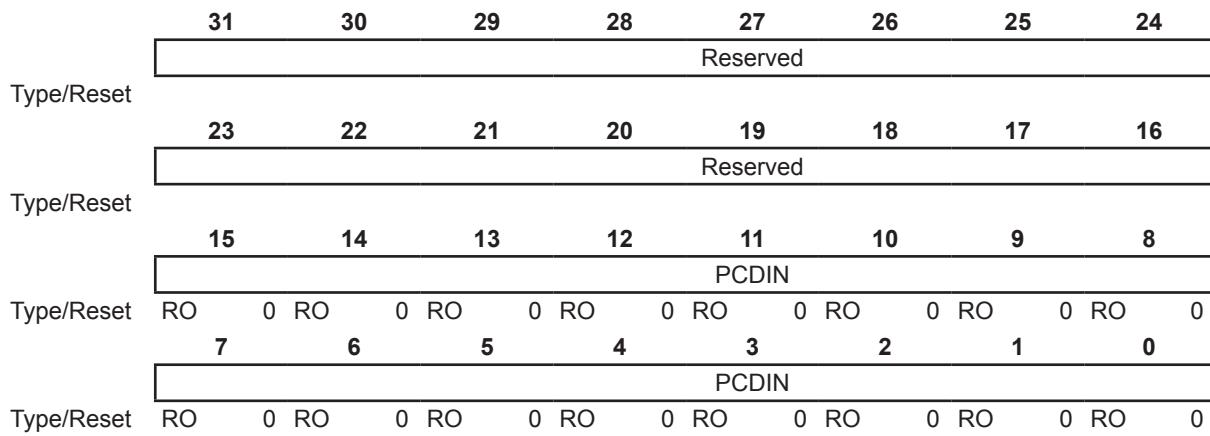
Bits	Field	Descriptions
[31:16]	PCLKEY	<p>GPIO Port C lock Key</p> <p>0x5FA0: Port C Lock function is enable</p> <p>Others: Port C Lock function is disable</p> <p>To lock the Port C function, a value 0x5FA0 should be written into the PCLKEY field in this register. To execute a successful write operation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

## Port C Data Input Register – PCDINR

This register specifies the GPIO Port C input data.

Offset: 0x01C

Reset value: 0x0000\_0000



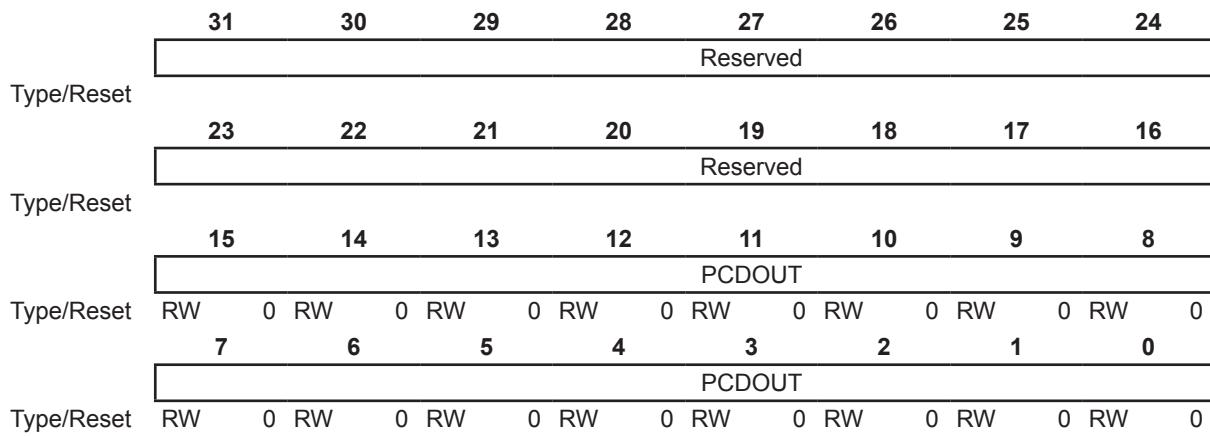
Bits	Field	Descriptions
[15:0]	PCDINn	GPIO Port C pin n Data Input Bits (n = 0 ~ 15) 0: The input data of corresponding pin is 0 1: The input data of corresponding pin is 1

## Port C Output Data Register – PCDOUTR

This register specifies the GPIO Port C output data.

Offset: 0x020

Reset value: 0x0000\_0000



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

## Port C Output Set/Reset Control Register – PCSRR

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

Offset: 0x024

Reset value: 0x0000\_0000

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

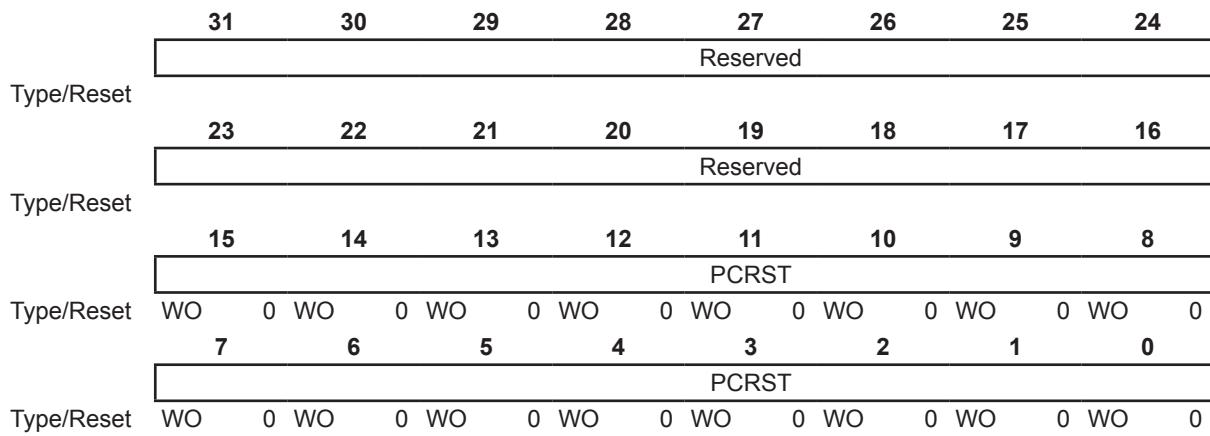
Bits	Field	Descriptions
[31:16]	PCRSTn	GPIO Port C pin n Output Reset Control Bits (n = 0 ~ 15) 0: No effect on the PCDOUTn bit 1: Reset the PCDOUTn bit
[15:0]	PCSETn	GPIO Port C pin n Output Set Control Bits (n = 0 ~ 15) 0: No effect on the PCDOUTn bit 1: Set the PCDOUTn bit Note that the function enabled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

## 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



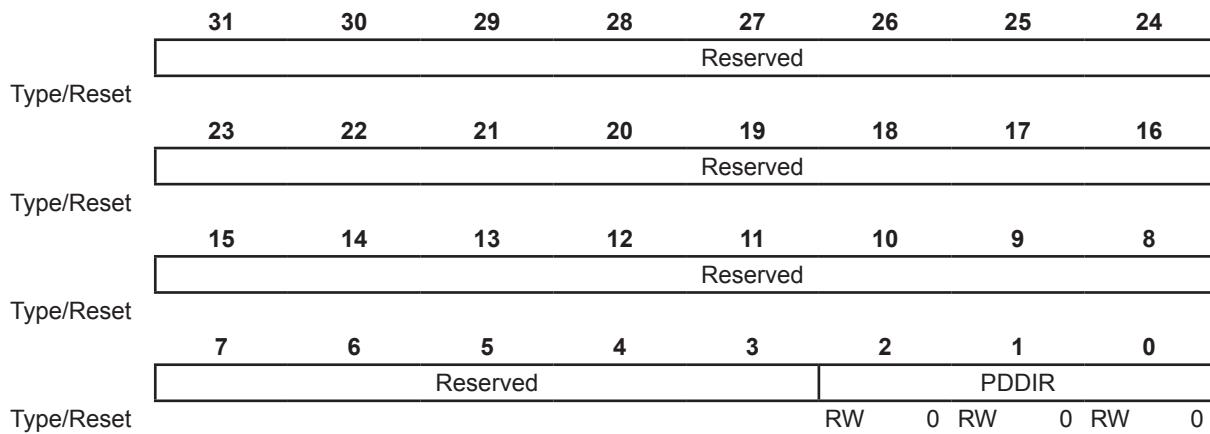
Bits	Field	Descriptions
[15:0]	PCRSTn	GPIO Port C pin n Output Reset Bits (n = 0 ~ 15) 0: No effect on the PCDOUTn bit 1: Reset the PCDOUTn bit

## Port D Data Direction Control Register – PDDIRCR

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

Offset: 0x000

Reset value: 0x0000\_0000



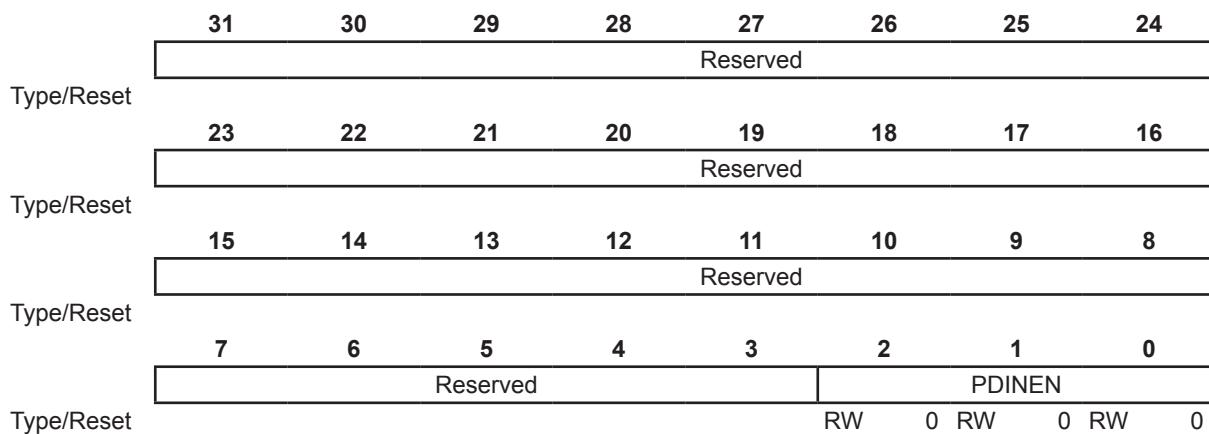
Bits	Field	Descriptions
[2:0]	PDDIRn	GPIO Port D pin n Direction Control Bits (n = 0 ~ 2) 0: Pin n is input mode 1: Pin n is output mode

## Port D Input Function Enable Control Register – PDINER

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

Offset: 0x004

Reset value: 0x0000\_0000



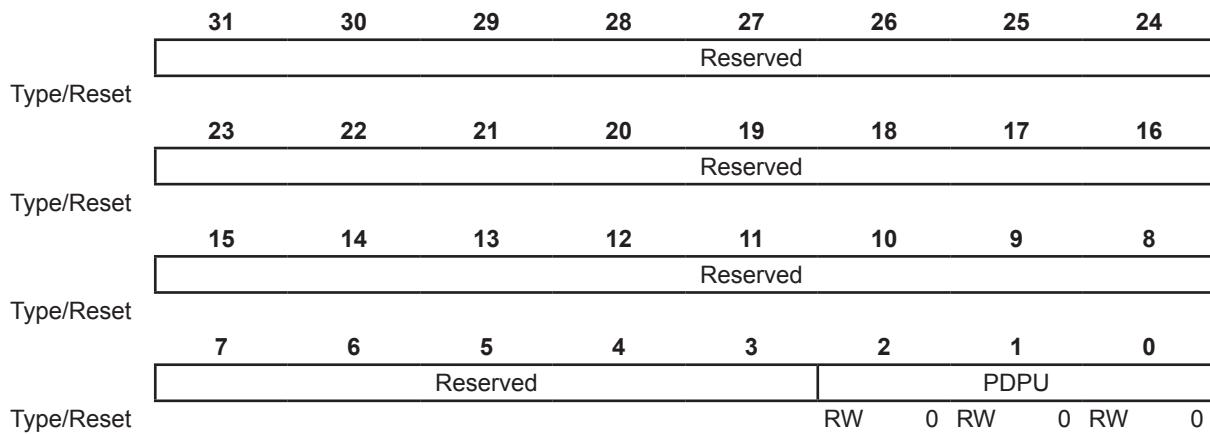
Bits	Field	Descriptions
[2:0]	PDINENn	<p>GPIO Port D pin n Input Enable Control Bits (n = 0 ~ 2)</p> <p>0: Pin n input function is disabled. 1: Pin n input function is enabled.</p> <p>When the pin n input function is disabled, the input Schmitt trigger will be turned off and the Schmitt trigger output will remain at a zero state.</p>

## Port D Pull-Up Selection Register – PDPUR

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

Offset: 0x008

Reset value: 0x0000\_0000



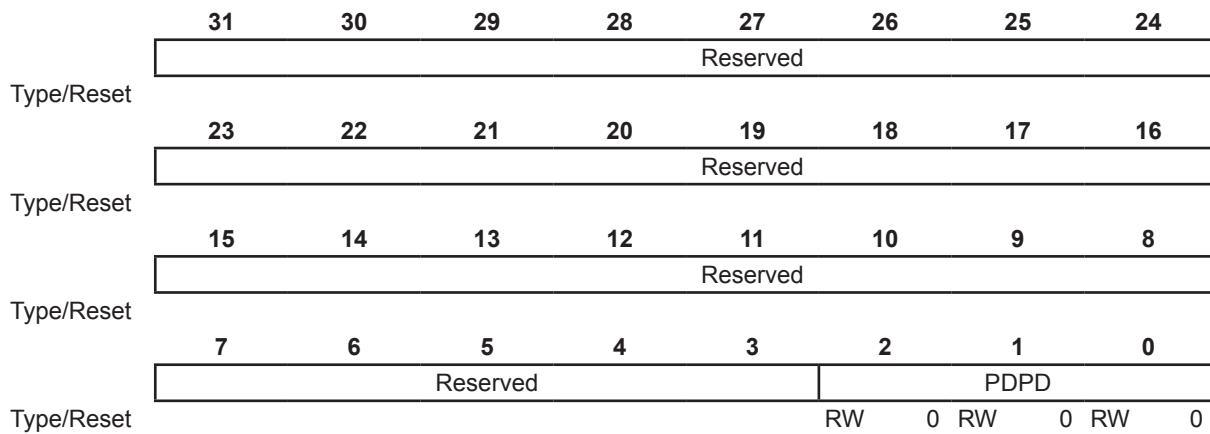
Bits	Field	Descriptions
[2:0]	PDPUn	<p>GPIO Port D pin n Pull-Up Selection Control Bits (n = 0 ~ 2)</p> <p>0: Pin n pull-up function is disabled 1: Pin n pull-up function is enabled</p> <p>Note: When the pull-up and pull-down functions are both enabled, the pull-up function will have the higher priority and therefore the pull-down function will be blocked and disabled.</p>

## Port D Pull-Down Selection Register – PDPDR

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

Offset: 0x00C

Reset value: 0x0000\_0000



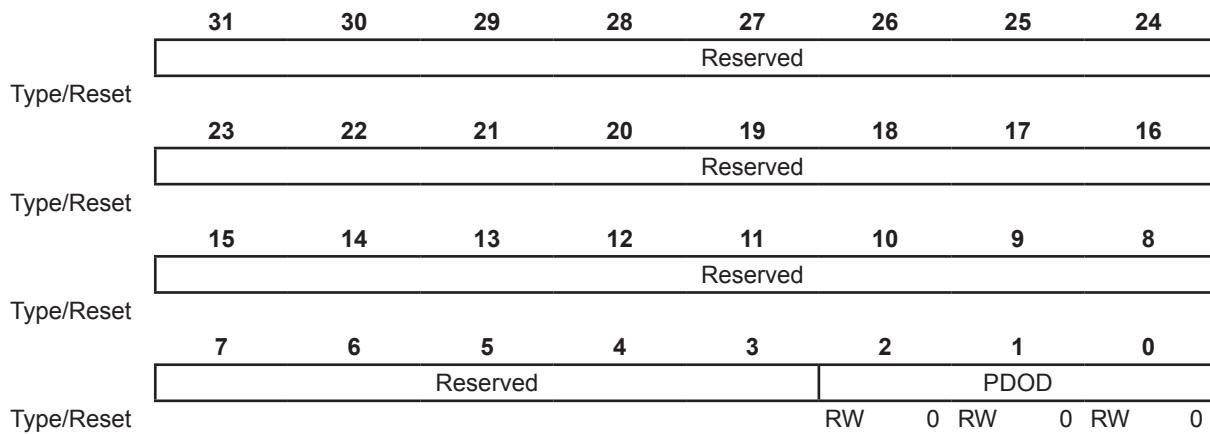
Bits	Field	Descriptions
[2:0]	PDPDn	<p>GPIO Port D pin n Pull-Down Selection Control Bits (n = 0 ~ 2)</p> <p>0: Pin n pull-down function is disabled</p> <p>1: Pin n pull-down function is enabled</p> <p>Note: When the pull-up and pull-down functions are both enabled, the pull-up function will have the higher priority and therefore the pull-down function will be blocked and disabled.</p>

## Port D Open Drain Selection Register – PDODR

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

Offset: 0x010

Reset value: 0x0000\_0000



Bits	Field	Descriptions
[2:0]	PDODn	GPIO Port D pin n Open Drain Selection Control Bits (n = 0 ~ 2) 0: Pin n Open Drain output is disabled. (The output type is CMOS output) 1: Pin n Open Drain output is enabled. (The output type is open-drain output)

## Port D Lock Register – PDLOCKR

This register specifies the GPIO Port D lock configuration.

Offset: 0x018

Reset value: 0x0000\_0000

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

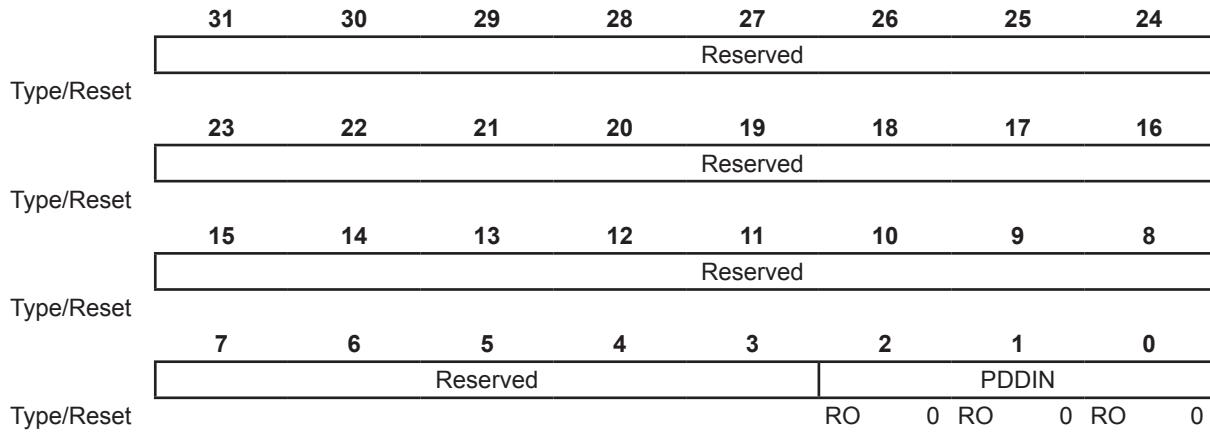
Bits	Field	Descriptions
[31:16]	PDLKEY	<p>GPIO Port D Lock Key</p> <p>0x5FA0: Port D Lock function is enable</p> <p>Others: Port D Lock function is disable</p> <p>To lock the Port D function, a value 0x5FA0 should be written into the PDLKEY field in this register. To execute a successful write operation on this lock register, the value written into the PDLKEY field must be 0x5FA0. If the value written into this field is not equal to 0x5FA0, any write operations on the PDLOCKR register will be aborted. The result of a read operation on the PDLKEY field returns the GPIO Port D Lock Status which indicates whether the GPIO Port D is locked or not. If the read value of the PDLKEY field is 0, this indicates that the GPIO Port D Lock function is disabled. Otherwise, it indicates that the GPIO Port D Lock function is enabled as the read value is equal to 1.</p>
[2:0]	PDLOCKn	<p>GPIO Port D pin n Lock Control Bits (n = 0 ~ 2)</p> <p>0: Port D pin n is not locked</p> <p>1: Port D pin n is locked</p> <p>The PDLOCKn bits are used to lock the configurations of corresponding GPIO Pins when the correct Lock Key is applied to the PDLKEY field. The locked configurations including PDDIRn, PDINENn, PDPUn, PDPDn and PDODn setting in the related GPIO registers. Additionally, the GPDCFGHR or GPDCFGLR field which is used to configure the alternative function of the associated GPIO pin will also be locked. Note that the PDLOCKR can only be written once which means that PDLKEY and PDLOCKn (lock control bit) should be written together and can not be changed until a system reset or GPIO Port D reset occurs.</p>

## Port D Data Input Register – PDDINR

This register specifies the GPIO Port D input data.

Offset: 0x01C

Reset value: 0x0000\_0000



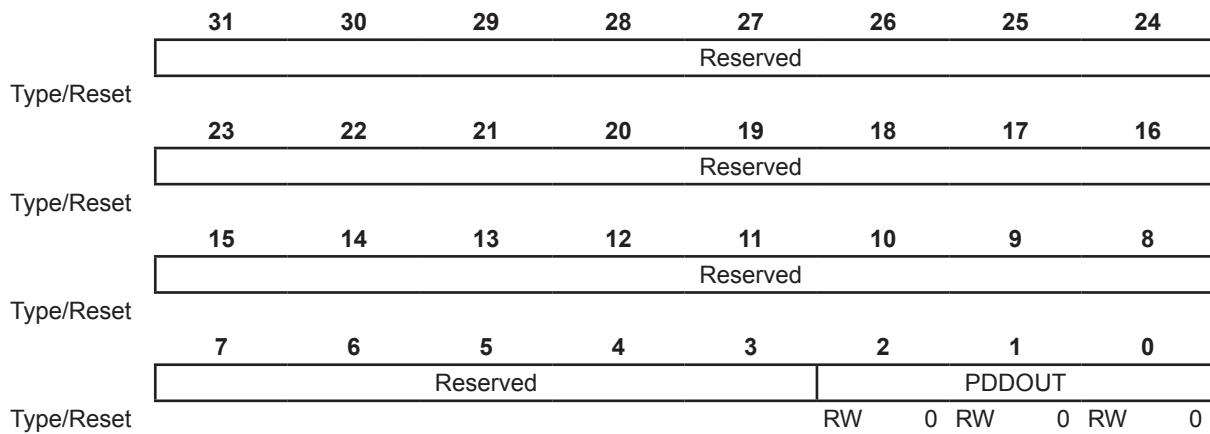
Bits	Field	Descriptions
[2:0]	PDDINn	GPIO Port D pin n Data Input Bits (n = 0 ~ 2) 0: The input data of corresponding pin is 0 1: The input data of corresponding pin is 1

## Port D Output Data Register – PDDOUTR

This register specifies the GPIO Port D output data.

Offset: 0x020

Reset value: 0x0000\_0000



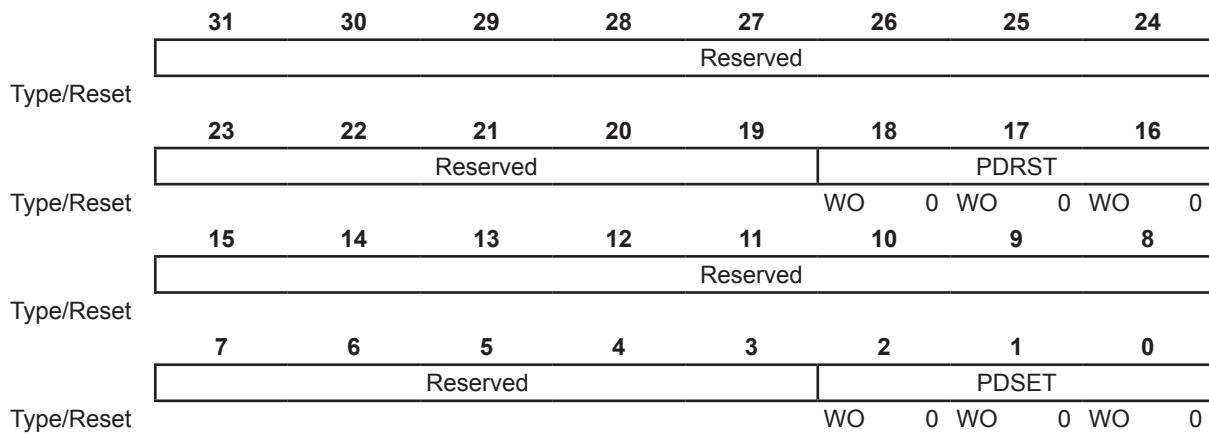
Bits	Field	Descriptions
[2:0]	PDDOUTn	GPIO Port D pin n Data Output Bits (n = 0 ~ 2) 0: Data to be output on pin n is 0 1: Data to be output on pin n is 1

## Port D Output Set/Reset Control Register – PDSRR

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

Offset: 0x024

Reset value: 0x0000\_0000



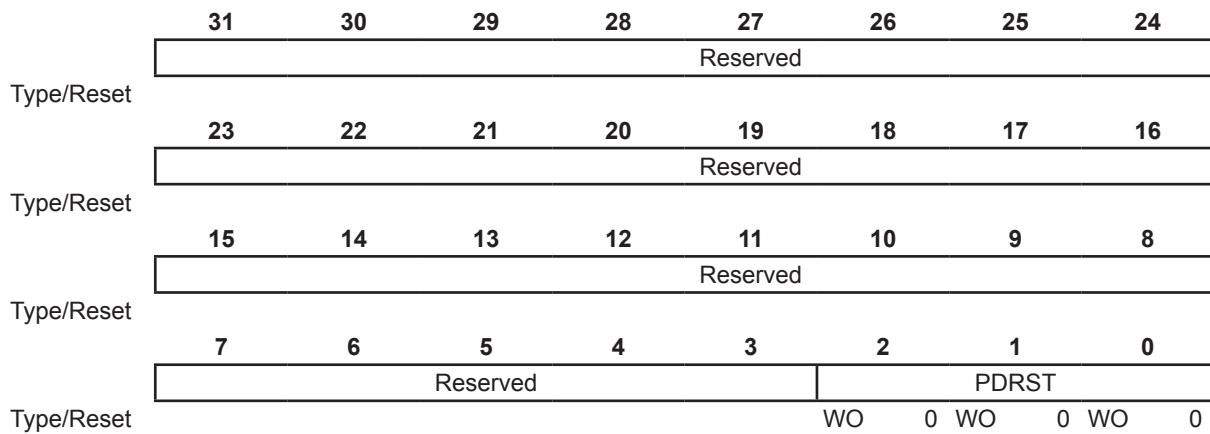
Bits	Field	Descriptions
[18:16]	PDRSTn	GPIO Port D pin n Output Reset Control Bits (n = 0 ~ 2) 0: No effect on the PDDOUTn bit 1: Reset the PDDOUTn bit
[2:0]	PDSETn	GPIO Port D pin n Output Set Control Bits (n = 0 ~ 2) 0: No effect on the PDDOUTn bit 1: Set the PDDOUTn bit Note that the function enabled by the PDSETn bit has the higher priority if both the PDRSTn and PDSETn bits are set at the same time.

## Port D Output Reset Register – PDRR

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

Offset: 0x028

Reset value: 0x0000\_0000



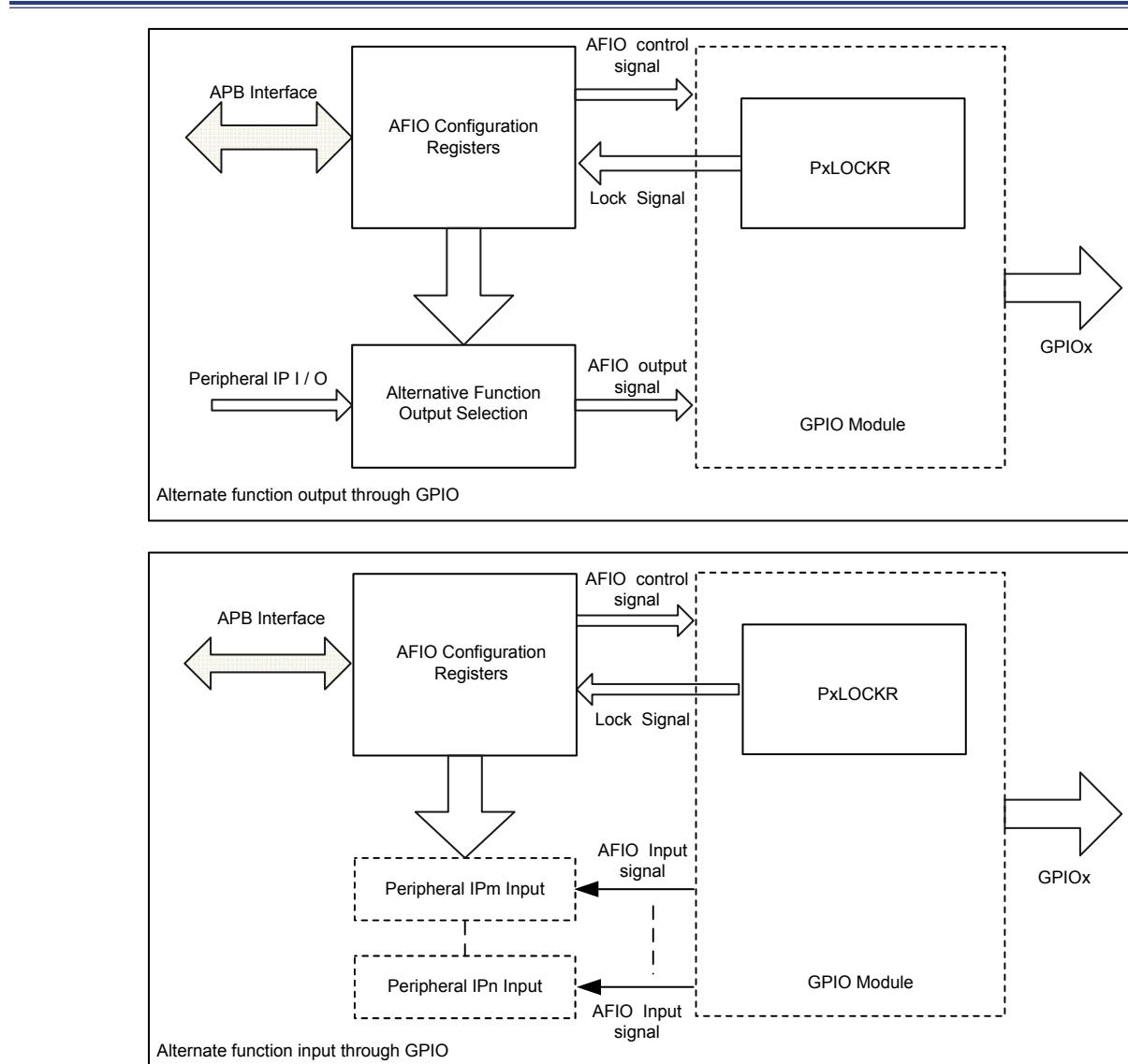
Bits	Field	Descriptions
[2:0]	PDRSTn	GPIO Port D pin n Output Reset Bits (n = 0 ~ 2) 0: No effect on the PDDOUTn bit 1: Reset the PDDOUTn bit

# 9

## Alternate Function Input/Output Control Unit (AFIO)

### Introduction

In order to expand the flexibility of the GPIO or the usage of peripheral functions, each IO pin can be configured to have up to sixteen different functions such as GPIO or IP functions by setting the 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 21. AFIO Block Diagram**

## Features

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

## Functional Descriptions

### External Interrupt Pin Selection

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

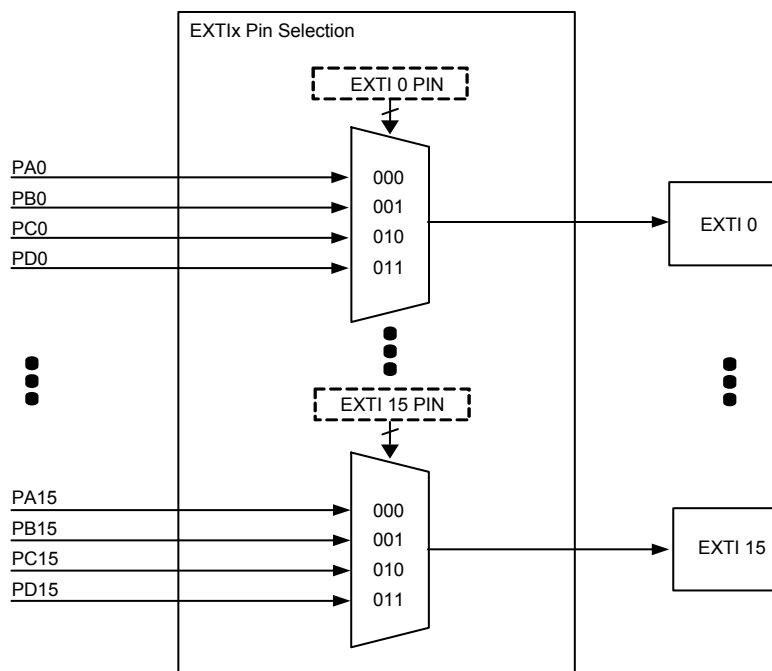


Figure 22. EXTI Channel Input Selection

## Alternate Function

Up to sixteen alternative functions can be chosen for each I/O pad by setting the PxCFGn [3:0] field in the GPxCFGRL or GPxCFGHR (n = 0~15, x = A~ D) registers. 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. Note that if the Operational Amplifier/Comparator is active, then pins PB [8:6] or PB [11:9] can not be set as other AFIO functional pins simultaneously.

- 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] = 0011: Alternate Function 3 (AF3)
- .....
- PxCFGn [3:0] = 1110: Alternate Function 14 (AF14)
- PxCFGn [3:0] = 1111: Alternate Function 15 (AF15)

**Table 22. AFIO Selection for Peripheral Map**

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	N/A	N/A	System Other

## Lock Mechanism

The GPIO PxLOCKR (i.e. x = A ~D) also offer lock key 0x5FA0 to lock AFIO input and output status until Reset.

## Register Map

The following table shows the AFIO register and reset value.

**Table 23. AFIO Register Map**

Register	Offset	Description	Reset Value
<b>AFIO Base Address = 0x4002_2000</b>			
ESSR0	0x000	EXTI Source Selection Register 0	0x0000_0000
ESSR1	0x004	EXTI Source Selection Register 1	0x0000_0000
GPACFGR0	0x020	GPIO Port A Alternate Function Configuration Register 0	0x0000_0000
GPACFGR1	0x024	GPIO Port A Alternate Function Configuration Register 1	0x0000_0000
GPBCFGR0	0x028	GPIO Port B Alternate Function Configuration Register 0	0x0000_0000
GPBCFGR1	0x02C	GPIO Port B Alternate Function Configuration Register 1	0x0000_0000
GPCCFGR0	0x030	GPIO Port C Alternate Function Configuration Register 0	0x0000_0000
GPCCFGR1	0x034	GPIO Port C Alternate Function Configuration Register 1	0x0000_0000
GPDCFGR0	0x038	GPIO Port D Alternate Function Configuration Register 0	0x0000_0000
GPDCFGR1	0x03C	GPIO Port D Alternate Function Configuration Register 1	0x0000_0000

## Register Descriptions

### EXTI Source Selection Register 0 – ESSR0

This register specifies the selection of IO EXTI0 ~ EXTI7.

Offset: 0x000

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
	EXTI7PIN							EXTI6PIN
	EXTI5PIN							EXTI4PIN
	EXTI3PIN							EXTI2PIN
	EXTI1PIN							EXTI0PIN

Bits	Field	Descriptions
[31:0]	EXTInPIN [3:0]	<p>EXTIn Pin Selection (n = 0 ~ 7)</p> <p>0000: PA Bit n is selected as EXTIn source signal            0001: PB Bit n is selected as EXTIn source signal            0010: PC Bit n is selected as EXTIn source signal            0011: PD Bit n is selected as EXTIn source signal            Others: Reserved</p> <p>Note: Since not all GPIO pins are available in all products and package types, please refer to the pin assignment section for detailed pin information. The setting of EXTInPIN [3:0] is invalid when the corresponding pin is not available.</p>

## EXTI Source Selection Register 1 – ESSR1

This register specifies the IO selection of EXTI15~EXTI8.

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
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

Bits	Field	Descriptions
[31:0]	EXTInPIN [3:0]	EXTIn Pin Selection (n = 8 ~ 15)  0000: PA Bit n is selected as EXTIn source signal 0001: PB Bit n is selected as EXTIn source signal 0010: PC Bit n is selected as EXTIn source signal 0011: PD Bit n is selected as EXTIn source signal Others: Reserved  Note: Since not all Port GPIO pins are available in all products and package types, please refer to the pin assignment section for detailed pin information. The setting of EXTInPIN [3:0] is invalid when the corresponding pin is not available.

## GPIO Alternate Function Configuration Register 0 – GPxCFG0 (x= A, B, C, D)

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

Offset: 0x020 (A Port), 0x028 (B Port), 0x030 (C Port), 0x038 (D Port)

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions																				
[31:0]	PxCFGn [3:0]	Port x AFIO Configuration Selection for bit y (y= 0 ~ 7) <table border="1"> <thead> <tr> <th>PxCFGn[3:0]</th> <th>Alternate Function Selection</th> </tr> </thead> <tbody> <tr> <td>0000</td> <td>AF0</td> </tr> <tr> <td>0001</td> <td>AF1</td> </tr> <tr> <td>0010</td> <td>AF2</td> </tr> <tr> <td>0011</td> <td>AF3</td> </tr> <tr> <td>....</td> <td>....</td> </tr> <tr> <td>1100</td> <td>AF12</td> </tr> <tr> <td>1101</td> <td>AF13</td> </tr> <tr> <td>1110</td> <td>AF14</td> </tr> <tr> <td>1111</td> <td>AF15</td> </tr> </tbody> </table>	PxCFGn[3:0]	Alternate Function Selection	0000	AF0	0001	AF1	0010	AF2	0011	AF3	....	....	1100	AF12	1101	AF13	1110	AF14	1111	AF15
PxCFGn[3:0]	Alternate Function Selection																					
0000	AF0																					
0001	AF1																					
0010	AF2																					
0011	AF3																					
....	....																					
1100	AF12																					
1101	AF13																					
1110	AF14																					
1111	AF15																					

## GPIO Alternate Function Configuration Register 1 – GPxCFG1 (x= A, B, C, D)

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

Offset: 0x024 (A Port), 0x02C (B Port), 0x034 (C Port), 0x03C (D Port)

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

Bits	Field	Descriptions																				
[31:0]	PxCFGn [3:0]	Port x AFIO Configuration Selection for bit y (n= 8 ~ 15) <table border="1"> <thead> <tr> <th>PxCFGn[3:0]</th> <th>Alternate Function Selection</th> </tr> </thead> <tbody> <tr> <td>0000</td> <td>AF0</td> </tr> <tr> <td>0001</td> <td>AF1</td> </tr> <tr> <td>0010</td> <td>AF2</td> </tr> <tr> <td>0011</td> <td>AF3</td> </tr> <tr> <td>....</td> <td>....</td> </tr> <tr> <td>1100</td> <td>AF12</td> </tr> <tr> <td>1101</td> <td>AF13</td> </tr> <tr> <td>1110</td> <td>AF14</td> </tr> <tr> <td>1111</td> <td>AF15</td> </tr> </tbody> </table>	PxCFGn[3:0]	Alternate Function Selection	0000	AF0	0001	AF1	0010	AF2	0011	AF3	....	....	1100	AF12	1101	AF13	1110	AF14	1111	AF15
PxCFGn[3:0]	Alternate Function Selection																					
0000	AF0																					
0001	AF1																					
0010	AF2																					
0011	AF3																					
....	....																					
1100	AF12																					
1101	AF13																					
1110	AF14																					
1111	AF15																					

# 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™-M3. 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-M3 for more details.

Additionally, an integrated simple, 24-bit down count timer (SysTick) is provided by the Cortex™-M3 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 24. Exception Types**

Exception type	Priority	Interrupt Number	Exception Number	Vector Address	Description
—	—	—	0	0x000	Initial Stack Pointer value
Reset	-3 (Highest)	—	1	0x004	Reset
NMI	-2	—	2	0x008	Non-Maskable Interrupt. The clock stuck interrupt signal (clock monitor function provided by Clock Control Unit) is connected to the NMI input
Hard Fault	-1	—	3	0x00C	All fault classes
Memory Management	Configurable <sup>(1)</sup>	—	4	0x010	Memory Protection Unit (MPU) mismatch, including access violation and no match
Bus Fault	Configurable <sup>(1)</sup>	—	5	0x014	Pre-fetch fault, memory access fault, and other address/memory related fault
Usage Fault	Configurable <sup>(1)</sup>	—	6	0x018	Usage fault, such as undefined executed instruction or illegal attempt of state transition
—	—	—	7	0x01C	Reserved
—	—	—	8	0x020	Reserved
—	—	—	9	0x024	Reserved
—	—	—	10	0x028	Reserved
SVCCall	Configurable <sup>(1)</sup>	—	11	0x02C	SVC instruction System service call
Debug Monitor	Configurable <sup>(1)</sup>	—	12	0x030	Debug monitor, when not halted
—	Configurable <sup>(1)</sup>	—	13	0x034	Reserved
PendSV	Configurable <sup>(1)</sup>	—	14	0x038	System Service Pendable request
SysTick	Configurable <sup>(1)</sup>	—	15	0x03C	SysTick timer interrupt
CKRDY	Configurable <sup>(2)</sup>	0	16	0x040	Clock ready interrupt (HSE, HSI, LSE, LSI, or PLL)
LVD	Configurable <sup>(2)</sup>	1	17	0x044	Low voltage detection interrupt
BOD	Configurable <sup>(2)</sup>	2	18	0x048	Brown-out detection interrupt
WDT	Configurable <sup>(2)</sup>	3	19	0x04C	Watchdog timer global interrupt

Exception type	Priority	Interrupt Number	Exception Number	Vector Address	Description
RTC	Configurable <sup>(2)</sup>	4	20	0x050	RTC global interrupt
FMC	Configurable <sup>(2)</sup>	5	21	0x054	FMC global interrupt
EVWUP	Configurable <sup>(2)</sup>	6	22	0x058	EXTI event wakeup interrupt
LPWUP	Configurable <sup>(2)</sup>	7	23	0x05C	WAKEUP pin interrupt
EXTI0	Configurable <sup>(2)</sup>	8	24	0x060	EXTI Line 0 interrupt
EXTI1	Configurable <sup>(2)</sup>	9	25	0x064	EXTI Line 1 interrupt
EXTI2	Configurable <sup>(2)</sup>	10	26	0x068	EXTI Line 2 interrupt
EXTI3	Configurable <sup>(2)</sup>	11	27	0x06C	EXTI Line 3 interrupt
EXTI4	Configurable <sup>(2)</sup>	12	28	0x070	EXTI Line 4 interrupt
EXTI5	Configurable <sup>(2)</sup>	13	29	0x074	EXTI Line 5 interrupt
EXTI6	Configurable <sup>(2)</sup>	14	30	0x078	EXTI Line 6 interrupt
EXTI7	Configurable <sup>(2)</sup>	15	31	0x07C	EXTI Line 7 interrupt
EXTI8	Configurable <sup>(2)</sup>	16	32	0x080	EXTI Line 8 interrupt
EXTI9	Configurable <sup>(2)</sup>	17	33	0x084	EXTI Line 9 interrupt
EXTI10	Configurable <sup>(2)</sup>	18	34	0x088	EXTI Line 10 interrupt
EXTI11	Configurable <sup>(2)</sup>	19	35	0x08C	EXTI Line 11 interrupt
EXTI12	Configurable <sup>(2)</sup>	20	36	0x090	EXTI Line 12 interrupt
EXTI13	Configurable <sup>(2)</sup>	21	37	0x094	EXTI Line 13 interrupt
EXTI14	Configurable <sup>(2)</sup>	22	38	0x098	EXTI Line 14 interrupt
EXTI15	Configurable <sup>(2)</sup>	23	39	0x09C	EXTI Line 15 interrupt
COMP	Configurable <sup>(2)</sup>	24	40	0x0A0	Comparator global interrupt
ADC	Configurable <sup>(2)</sup>	25	41	0x0A4	ADC global interrupt
—	—	26	42	0x0A8	Reserved
MCTM0_BRK	Configurable <sup>(2)</sup>	27	43	0x0AC	MCTM0 break interrupt
MCTM0_UP	Configurable <sup>(2)</sup>	28	44	0x0B0	MCTM0 update interrupt
MCTM0_TR_UP2	Configurable <sup>(2)</sup>	29	45	0x0B4	MCTM0 trigger/update event 2 interrupt
MCTM0_CC	Configurable <sup>(2)</sup>	30	46	0x0B8	MCTM0 capture/compare interrupt
MCTM1_BRK	Configurable <sup>(2)</sup>	31	47	0x0BC	MCTM1 break interrupt
MCTM1_UP	Configurable <sup>(2)</sup>	32	48	0x0C0	MCTM1 update interrupt
MCTM1_TR_UP2	Configurable <sup>(2)</sup>	33	49	0x0C4	MCTM1 trigger/update event 2 interrupt
MCTM1_CC	Configurable <sup>(2)</sup>	34	50	0x0C8	MCTM1 capture/compare interrupt
GPTM0	Configurable <sup>(2)</sup>	35	51	0x0CC	GPTM0 global interrupt
GPTM1	Configurable <sup>(2)</sup>	36	52	0x0D0	GPTM1 global interrupt
—	—	37	53	0x0D4	Reserved
—	—	38	54	0x0D8	Reserved
—	—	39	55	0x0DC	Reserved
—	—	40	56	0x0E0	Reserved
BFTM0	Configurable <sup>(2)</sup>	41	57	0x0E4	BFTM0 global interrupt
BFTM1	Configurable <sup>(2)</sup>	42	58	0x0E8	BFTM1 global interrupt
I <sup>2</sup> C0	Configurable <sup>(2)</sup>	43	59	0x0EC	I <sup>2</sup> C0 global interrupt
I <sup>2</sup> C1	Configurable <sup>(2)</sup>	44	60	0x0F0	I <sup>2</sup> C1 global interrupt
SPI0	Configurable <sup>(2)</sup>	45	61	0x0F4	SPI0 global interrupt

Exception type	Priority	Interrupt Number	Exception Number	Vector Address	Description
SPI1	Configurable <sup>(2)</sup>	46	62	0x0F8	SPI1 global interrupt
USART0	Configurable <sup>(2)</sup>	47	63	0x0FC	USART0 global interrupt
USART1	Configurable <sup>(2)</sup>	48	64	0x100	USART1 global interrupt
UART0	Configurable <sup>(2)</sup>	49	65	0x104	UART0 global interrupt
UART1	Configurable <sup>(2)</sup>	50	66	0x108	UART1 global interrupt
SCI	Configurable <sup>(2)</sup>	51	67	0x10C	SCI global interrupt
I <sup>2</sup> S	Configurable <sup>(2)</sup>	52	68	0x110	I <sup>2</sup> S global interrupt
USB	Configurable <sup>(2)</sup>	53	69	0x114	USB global interrupt
—	—	54	70	0x118	Reserved
PDMA_CH0	Configurable <sup>(2)</sup>	55	71	0x11C	PDMA channel 0 global interrupt
PDMA_CH1	Configurable <sup>(2)</sup>	56	72	0x120	PDMA channel 1 global interrupt
PDMA_CH2	Configurable <sup>(2)</sup>	57	73	0x124	PDMA channel 2 global interrupt
PDMA_CH3	Configurable <sup>(2)</sup>	58	74	0x128	PDMA channel 3 global interrupt
PDMA_CH4	Configurable <sup>(2)</sup>	59	75	0x12C	PDMA channel 4 global interrupt
PDMA_CH5	Configurable <sup>(2)</sup>	60	76	0x130	PDMA channel 5 global interrupt
PDMA_CH6	Configurable <sup>(2)</sup>	61	77	0x134	PDMA channel 6 global interrupt
PDMA_CH7	Configurable <sup>(2)</sup>	62	78	0x138	PDMA channel 7 global interrupt
—	—	63	79	0x13C	Reserved
—	—	64	80	0x140	Reserved
—	—	65	81	0x144	Reserved
—	—	66	82	0x148	Reserved
—	—	67	83	0x14C	Reserved
EBI	Configurable <sup>(2)</sup>	68	84	0x150	EBI global interrupt

**NOTES:** 1. The exception priority can be changed using the NVIC System Handler Priority Registers. For more information, refer to the ARM “Technical Reference Manual of Cortex™-M3” document.  
2. The interrupt priority can be changed using the NVIC Interrupt Priority Registers. For more information, refer to the ARM “Technical Reference Manual of Cortex™-M3” document.

## Features

- 16 system Cortex™-M3 exceptions
- Up to 58 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 or AHB 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 for the RTOS tick timer or other purpose. The TENMS field in the SCALIB register has a fixed value of 9000 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 9 MHz (72 MHz divide by 8).

## Register Map

The following table shows the NVIC registers and reset values.

**Table 25. NVIC Register Map**

Register	Offset	Description	Reset Value
<b>NVIC Base Address = 0xE000_E000</b>			
ICTR	0x004	Interrupt Control Type Register	0x0000_0001
SCTRL	0x010	SysTick Control and Status Register	0x0000_0000
SLOAD	0x014	SysTick Reload Value Register	Unpredictable
SVAL	0x018	SysTick Current Value Register	Unpredictable
SCALIB	0x01C	SysTick Calibration Value Register	0x4000_2328
ISER0_31	0x100	Irq 0 to 31 Set Enable Register	0x0000_0000
ISER32_63	0x104	Irq 32 to 63 Set Enable Register	0x0000_0000
ISER64_95	0x108	Irq 64 to 95 Set Enable Register	0x0000_0000
ICER0_31	0x180	Irq 0 to 31 Clear Enable Register	0x0000_0000
ICER32_63	0x184	Irq 32 to 63 Clear Enable Register	0x0000_0000
ICER64_95	0x188	Irq 64 to 95 Clear Enable Register	0x0000_0000
ISPR0_31	0x200	Irq 0 to 31 Set Pending Register	0x0000_0000
ISPR32_63	0x204	Irq 32 to 63 Set Pending Register	0x0000_0000
ISPR64_95	0x208	Irq 64 to 95 Set Pending Register	0x0000_0000
ICPR0_31	0x280	Irq 0 to 31 Clear Pending Register	0x0000_0000
ICPR32_63	0x284	Irq 32 to 63 Clear Pending Register	0x0000_0000
ICPR64_95	0x288	Irq 64 to 95 Clear Pending Register	0x0000_0000
IABR0_31	0x300	Irq 0 to 31 Active Bit Register	0x0000_0000
IABR32_63	0x304	Irq 32 to 63 Active Bit Register	0x0000_0000
IABR64_95	0x308	Irq 64 to 95 Active Bit Register	0x0000_0000
IRQ0_3	0x400	Irq 0 to 3 Priority Register	0x0000_0000
IRQ4_7	0x404	Irq 4 to 7 Priority Register	0x0000_0000
IRQ8_11	0x408	Irq 8 to 11 Priority Register	0x0000_0000
IRQ12_15	0x40C	Irq 12 to 15 Priority Register	0x0000_0000
IRQ16_19	0x410	Irq 16 to 19 Priority Register	0x0000_0000
IRQ20_23	0x414	Irq 20 to 23 Priority Register	0x0000_0000
IRQ24_27	0x418	Irq 24 to 27 Priority Register	0x0000_0000
IRQ28_31	0x41C	Irq 28 to 31 Priority Register	0x0000_0000
IRQ32_35	0x420	Irq 32 to 35 Priority Register	0x0000_0000
IRQ36_39	0x424	Irq 36 to 39 Priority Register	0x0000_0000
IRQ40_43	0x428	Irq 40 to 43 Priority Register	0x0000_0000
IRQ44_47	0x42C	Irq 44 to 47 Priority Register	0x0000_0000

Register	Offset	Description	Reset Value
IRQ48_51	0x430	Irq 48 to 51 Priority Register	0x0000_0000
IRQ52_55	0x434	Irq 52 to 55 Priority Register	0x0000_0000
IRQ56_59	0x438	Irq 56 to 59 Priority Register	0x0000_0000
IRQ60_63	0x43C	Irq 60 to 63 Priority Register	0x0000_0000
IRQ64_67	0x440	Irq 64 to 67 Priority Register	0x0000_0000
ICSR	0xD04	Interrupt Control State Register	0x0000_0000
VTOR	0xD08	Vector Table Offset Register	0x0000_0000
AIRCR	0xD0C	Application Interrupt/Reset Control Register	0xFA05_0000
SCR	0xD10	System Control Register	0x0000_0000
CCR	0xD14	Configuration Control Register	0x0000_0000
SHPR4-7	0xD18	System Handlers 4-7 Priority Register	0x0000_0000
SHPR8_11	0xD1C	System Handlers 8-11 Priority Register	0x0000_0000
SHPR12_15	0xD20	System Handlers 12-15 Priority Register	0x0000_0000
SHCSR	0xD24	System Handler Control and State Register	0x0000_0000
CFSR	0xD28	Configurable Fault Status Registers	0x0000_0000
HFSR	0xD2C	Hard Fault Status Register	0x0000_0000
DFSR	0xD30	Debug Fault Status Register	0x0000_0000
MMFAR	0xD34	Mem Manage Address Register	Unpredictable
BFAR	0xD38	Bus Fault Address Register	Unpredictable
AFSR	0xD3C	Auxiliary Fault Status Register	0x0000_0000
STIR	0xF00	Software Trigger Interrupt Register	0x0000_0000

**NOTE:** For more information of the detail descriptions of the above registers, please refer to the “Technical Reference Manual of Cortex™-M3” document from ARM.

# 11

## External Interrupt/Event Controller (EXTI)

### Introduction

The External Interrupt/Event Controller, EXTI, comprises 16 edge detectors which can generate a wake-up event or interrupt requests independently. In interrupt mode there are five trigger types which can be selected as the external interrupt trigger type, low level, high level, negative edge, positive edge and both edges, selectable using the SRCnTYPE field in the EXTICFGRn register. In the wake-up event mode, the wake-up event polarity can be configured by setting the EXTInWPOL 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.

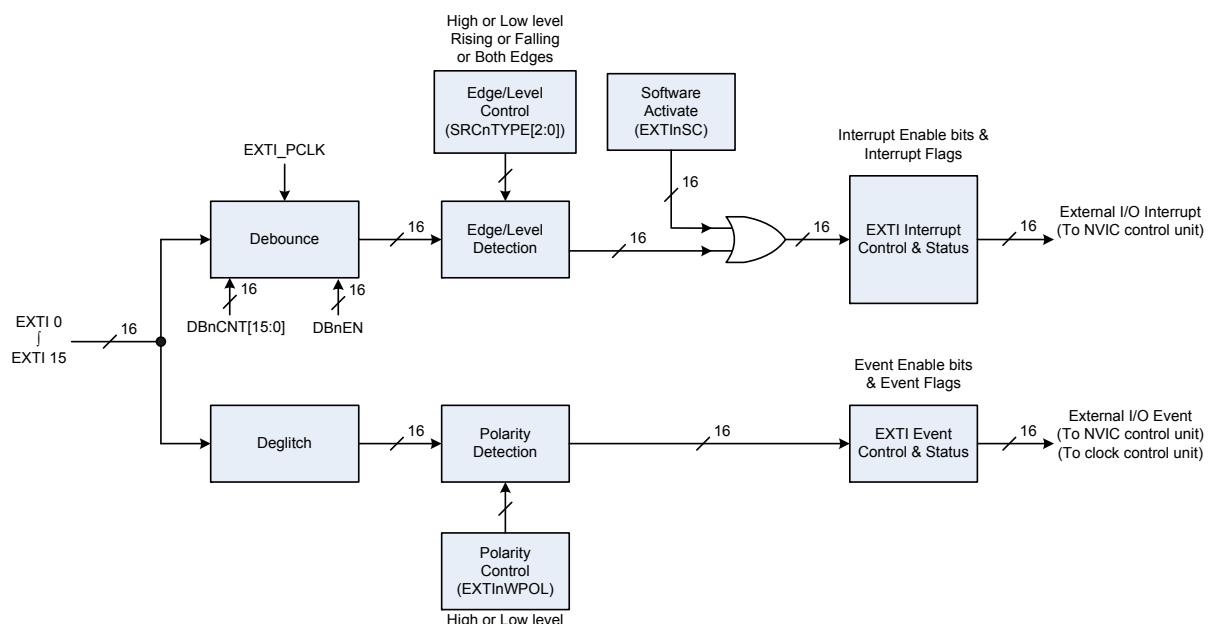


Figure 23. EXTI Block Diagram

### 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 Cortex™-M3 core and the Clock Control Unit, CKCU. These external events include EXTI events, Low Voltage Detection, WAKEUP input pin, Comparator and RTC wakeup functions. By configuring the wakeup event enable bit in the corresponding peripheral, the wakeup signal will be sent to the Cortex™-M3 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

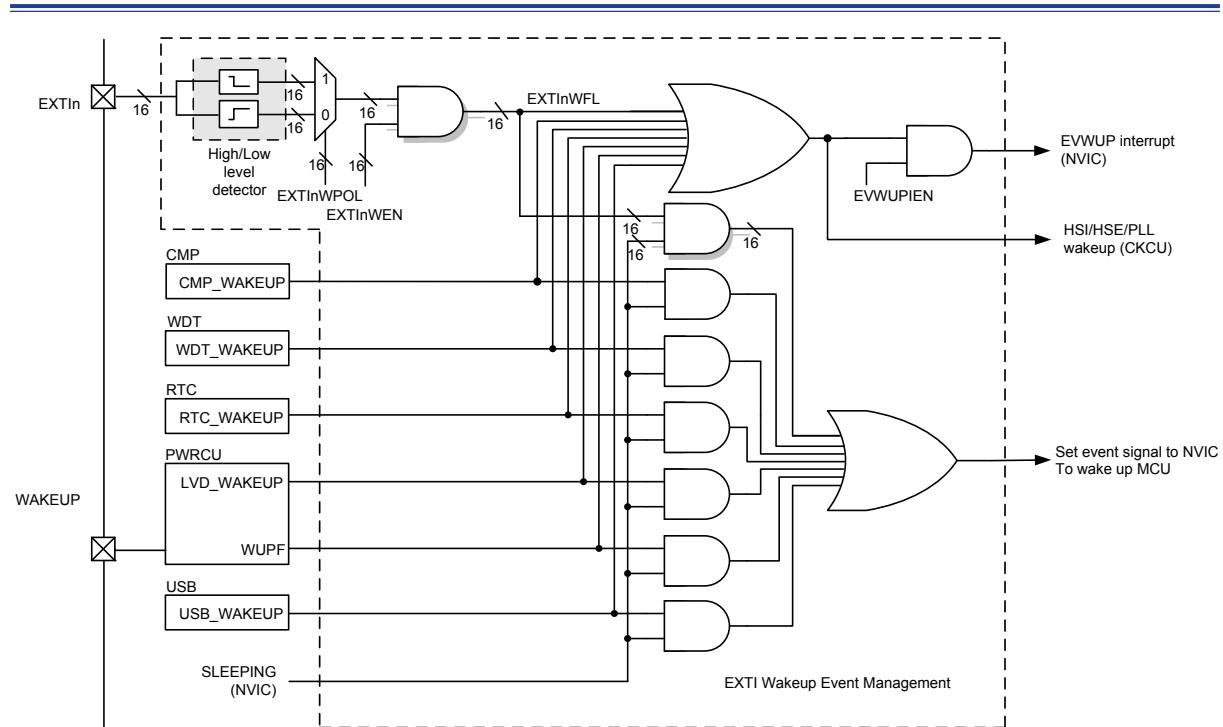


Figure 24. EXTI Wake-up Event Management

## External Interrupt/Event Line Mapping

All GPIO pins can be selected as EXTI trigger sources by configuring the EXTInPIN [3:0] field in the AFIO ESSRn 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 EXTICFGn to enable the corresponding pin de-bounce function and configure the DBnCNT field in the EXTICFGn 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.

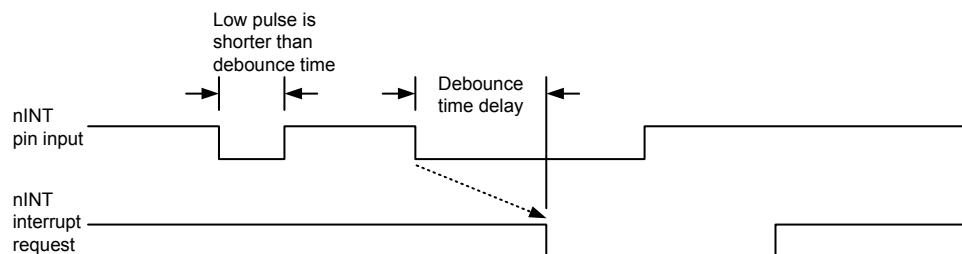


Figure 25. EXTI Interrupt Debounce Function

## Register Map

The following table shows the EXTI registers and reset values.

**Table 26. EXTI Register Map**

Register	Offset	Description	Reset Value
<b>EXTI Base Address = 0x4002_4000</b>			
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
EXTISCR	0x04C	EXTI Interrupt Software Set Command Register	0x0000_0000
EXTIWAKUPCR	0x050	EXTI Interrupt Wakeup Control Register	0x0000_0000
EXTIWAKUPPOLR	0x054	EXTI Interrupt Wakeup Polarity Register	0x0000_0000
EXTIWAKUPFLG	0x058	EXTI Interrupt Wakeup Flag Register	0x0000_0000

## Register Descriptions

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

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

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

Reset value: 0x0000\_0000

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

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) No matter if the interrupt source is configured to be active by the EXTI, it should always be configured as level-sensitive.
[15:0]	DBnCNT	EXTIn De-bounce Counter (n = 0 ~ 15) The de-bounce time is calculated with DBnCNT x APB clock (EXTI_PCLK) period and should be long enough to take effect on the input signal.

## EXTI Interrupt Control Register – EXTICR

This register is used to control the EXTI interrupt.

Offset: 0x040

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Reserved								
Type/Reset								
Reserved								
Type/Reset								
	15	14	13	12	11	10	9	8
Type/Reset	EXTI15EN	EXTI14EN	EXTI13EN	EXTI12EN	EXTI11EN	EXTI10EN	EXTI9EN	EXTI8EN
	RW	0	RW	0	RW	0	RW	0
Type/Reset								
	7	6	5	4	3	2	1	0
Type/Reset	EXTI7EN	EXTI6EN	EXTI5EN	EXTI4EN	EXTI3EN	EXTI2EN	EXTI1EN	EXTI0EN
	RW	0	RW	0	RW	0	RW	0

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

## EXTI Interrupt Edge Flag Register – EXTIEDGEFLGR

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

Offset: 0x044

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Reserved								
Type/Reset								
Reserved								
Type/Reset								
	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
Type/Reset								
	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	EXTIn Both Edge Detection Flag (n = 0 ~ 15) 0: No edge is detected 1: Positive or negative edge is detected This bit is set by the hardware circuitry when a positive or negative edge is detected on the corresponding EXTI line. Software should write 1 to clear it.

## EXTI Interrupt Edge Status Register – EXTIEDGESR

This register indicates the polarity of a detected EXTI edge.

Offset: 0x048

Reset value: 0x0000\_0000

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

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

## EXTI Interrupt Software Set Command Register – EXTISSCR

This register is used to activate the EXTI interrupt.

Offset: 0x04C

Reset value: 0x0000\_0000

	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	EXTI15SC	EXTI14SC	EXTI13SC	EXTI12SC	EXTI11SC	EXTI10SC	EXTI9SC	EXTI8SC
	RW	0	RW	0	RW	0	RW	0
Type/Reset	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		
	EVWUPIEN Reserved									
Type/Reset	RW	0	Reserved							
	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		
Type/Reset	7	6	5	4	3	2	1	0		
	EXTI7WEN	EXTI6WEN	EXTI5WEN	EXTI4WEN	EXTI3WEN	EXTI2WEN	EXTI1WEN	EXTI0WEN		
Type/Reset	RW	0	RW	0	RW	0	RW	0		

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

## EXTI Interrupt Wakeup Polarity Register – EXTIWAKUPPOLR

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

Offset: 0x054

Reset value: 0x0000\_0000

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

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

## EXTI Interrupt Wakeup Flag Register – EXTIWAKUPFLG

This register is the EXTI interrupt wake flag register.

Offset: 0x058

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[15:0]	EXTInWFL	EXTIn Wakeup Flag (n = 0 ~ 15) 0: No wakeup occurs 1: System is 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 18 multiplexed channels including 16 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 modes. A left-aligned or right-aligned 16-bit data register is provided to store the data after conversion. Since the products with different pin counts are provided, please refer to the datasheet for more information about available input channels.

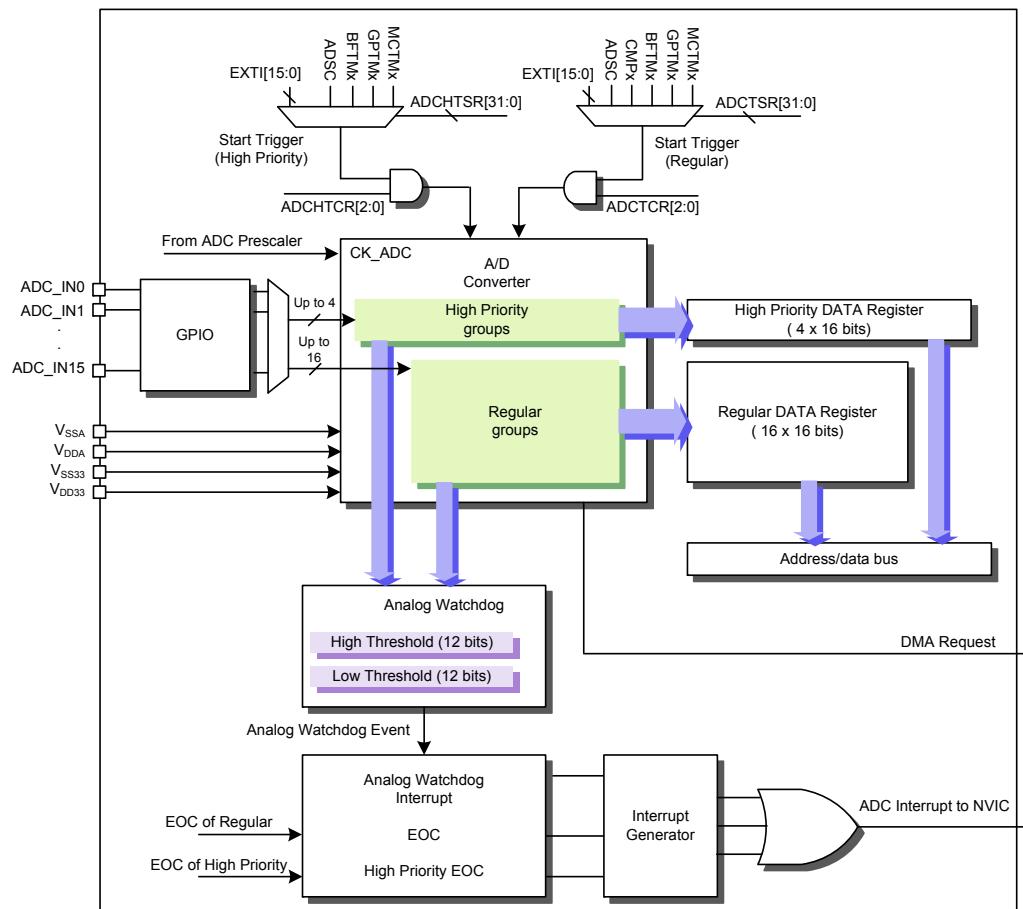


Figure 26. ADC Block Diagram

## Features

- 12-bit SAR ADC engine
- Up to 1 MSPS conversion rate
  - 1  $\mu$ s at 56 MHz, 1.17  $\mu$ s at 72 MHz
- 16 external analog input channels
- 2 internal reference voltage detect analog input channels
- Individual programmable sampling time for each channel
- Three conversion modes
  - One shot conversion mode
  - Continuous conversion mode
  - Discontinuous conversion mode
- Two level conversion priority
  - Regular – can be interrupted by higher priority conversion
  - High priority
- Up to 16 dedicated sequencers and data registers for regular conversion
- Up to 4 dedicated sequencers and data registers for high priority conversion
- Data alignment adjustment and offset cancellation
  - Right/left alignment
  - Signed/unsigned
  - 16 offset registers for each channel
- Analog watchdog for predefined voltage range monitor
  - Lower/upper threshold registers
  - Interrupt generation
- Various trigger start sources for both regular and high priority conversion modes
  - Software trigger
  - EXTI – external interrupt input pin
  - GPTM0/GPTM1 trigger
  - MCTM0/MCTM1 trigger
  - BFTM0/BFTM1 trigger
  - CMP0/CMP1 trigger – for regular conversion only
- Multiple generated interrupts
  - End of single conversion
  - End of subgroup conversion
  - End of cycle conversion
  - Analog Watchdog
  - Data register overwrite
- PDMA request at the end of conversion

## Functional Description

### ADC Clock Setup

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

### Regular and High Priority Channel Selection

The A/D converter supports 16 multiplexed channels and organizes the conversion results into two groups: a regular group and a high priority group. A regular group can organize a conversion sequence which can be implemented on the channels arranged in a specific conversion sequence length from 1 to 16. 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 regular group is composed of up to 16 conversions. The selected channels of the regular group conversion can be specified in the ADCLST0~ADCLST3 registers. The total conversion sequence length is setup using the ADSEQL[3:0] bits in the ADCCONV register.

A high priority group is composed of up to 4 conversions. The selected channels of the high priority group conversions can be set in the ADCHLST register. The total conversion sequence length of the high priority group can be set in the ADHSEQL[1:0] bits in the ADCHCONV 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 Modes

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

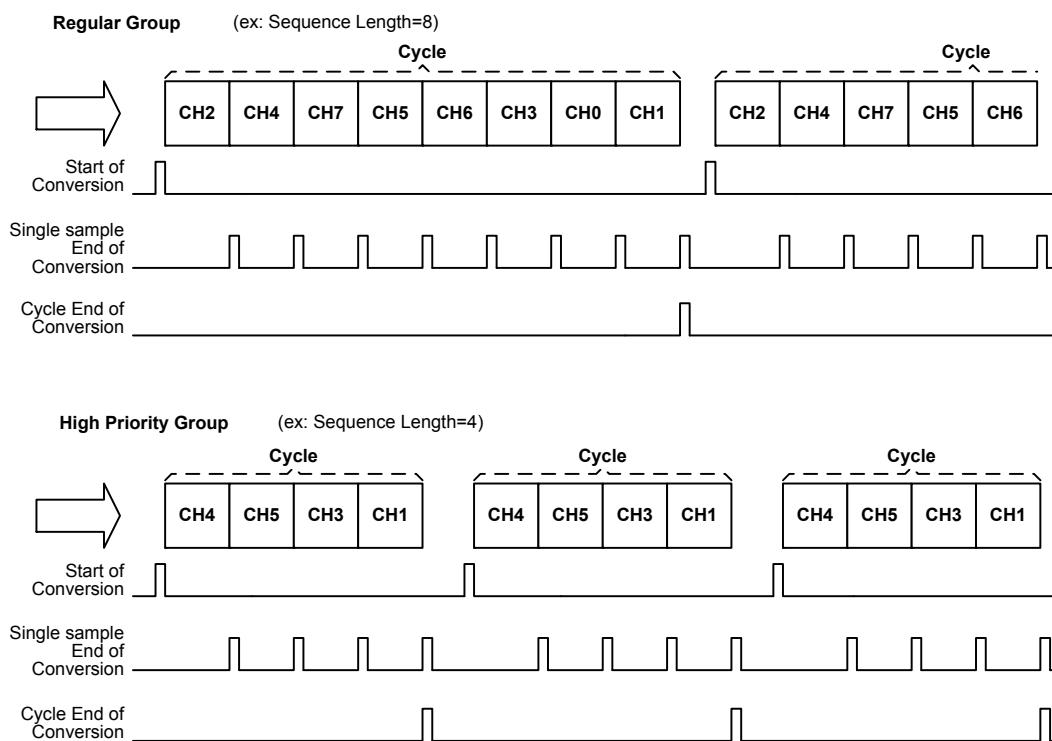
In the One Shot Conversion mode, the ADC will perform conversion cycles on the channels specified in the A/D conversion list registers ADCLSTn or ADCHLST with a specific sequence when an A/D start trigger event occurs. When the A/D conversion mode field AD(H)MODE [1:0] is set to 0x0, the A/D converter will operate in the One Shot Conversion Mode. This mode can be started by a software trigger, a comparator output transition event, an external EXTI event or a TM event for a regular or high priority group determined by the Trigger Control Register ADCTCR or ADCHTCR, and the Trigger Source Register ADCTS or ADCHTSR.

#### Regular Conversion:

- The converted data will be stored in the 16-bit ADCDRy (y = 0~15) registers.
- The ADC regular 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 regular group cycle end of conversion if the ADIEC bit in the ADCIER register is enabled.

### High Priority Conversion:

- The converted data will be stored in the 16-bit ADCHDRy ( $y = 0\sim 3$ ) registers.
- The ADC high priority single sample end of conversion event raw status flag, ADIRAWHS, in the ADCIRAW register will be set when the single sample conversion is finished.
- An interrupt will be generated after a high priority single sample end of conversion if the ADIEHS bit in the ADCIER register is enabled.
- An interrupt will be generated after a high priority group cycle end of conversion if the ADIEHC bit in the ADCIER register is enabled.



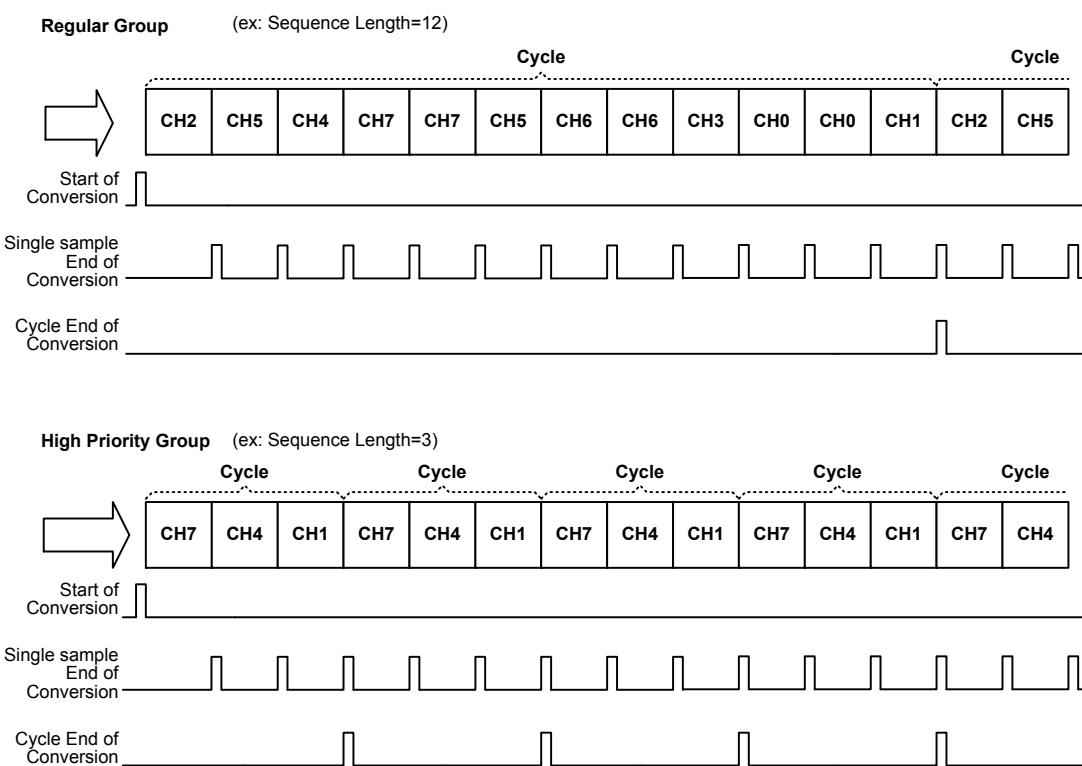
**Figure 27. One Shot Conversion Mode**

### Continuous Conversion Mode

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

#### After each conversion

- The converted data will be stored in the 16-bit ADCDRy ( $y = 0\sim15$ ) or ADCHDRy ( $y = 0\sim3$ ) registers.
- The ADC regular group or high priority group cycle end of conversion event raw status flag, ADIRAWC or ADIRAWHC, in the ADCIRAW register will be set when the conversion cycle is finished.
- An interrupt will be generated after a regular or high priority group cycle end of conversion if the ADIEC or ADIEHC bit in the ADCIER register is enabled.



**Figure 28. Continuous Conversion Mode**

## Discontinuous Conversion Mode

### Regular group

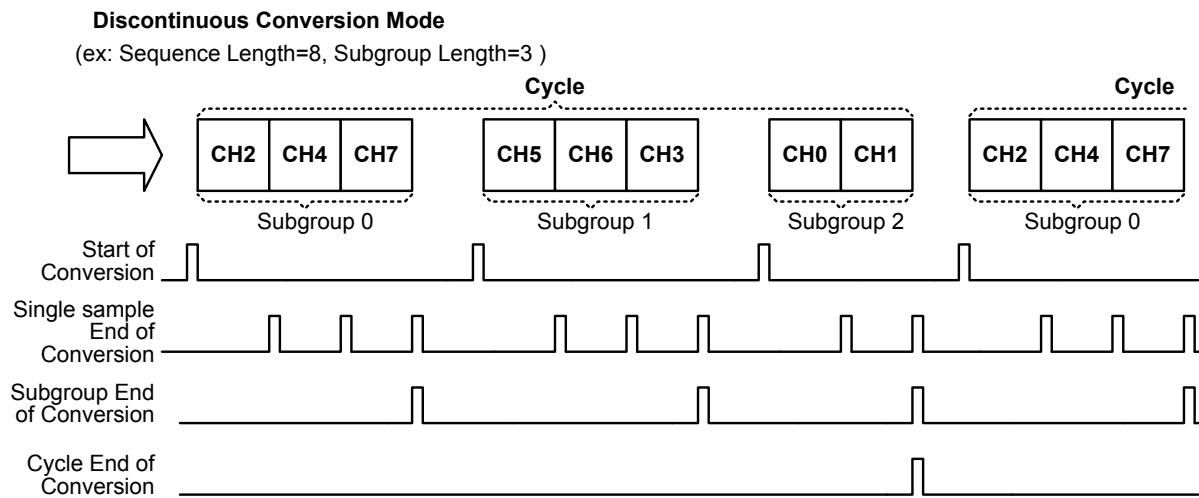
The A/D converter will operate in the Discontinuous Conversion Mode for regular groups when the A/D conversion mode bit field ADMODE [1:0] in the ADCCONV register is set to 0x3. The regular group to be converted can have up to 16 channels and can be arranged in a specific sequence by configuring the ADCLSTn registers where n ranges from 0 to 3. This mode is provided to convert data for the regular group with a short sequence, named as the A/D regular conversion subgroup, each time a trigger event occurs. The subgroup length is defined in the ADSUBL [3:0] field to specify the subgroup length. In the Discontinuous Conversion Mode the A/D converter can be started by a software trigger, a comparator output transition event, an external EXTI event or a TM event for regular 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 regular 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 regular group channels, of which the total number is defined by the ADSEQL[3:0] bits in the ADCCONV register, have finished their conversion, at which point the regular 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 29. Regular Group Discontinuous Conversion Mode**

### High priority group

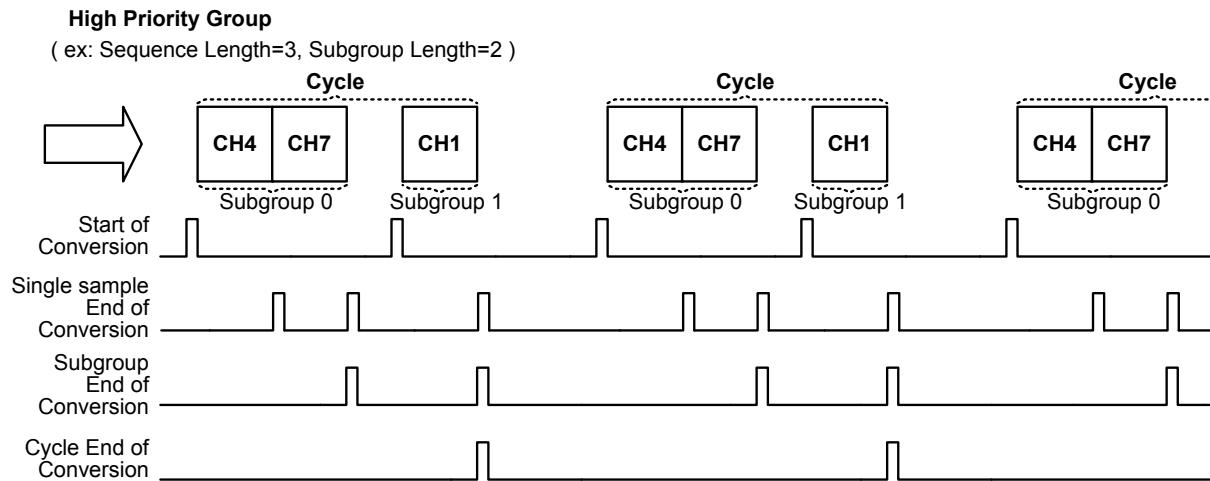
The A/D converter will operate in the Discontinuous Conversion Mode for the high priority group when the A/D high priority conversion mode bit field ADHMODE [1:0] in the ADCHCONV register is set to 0x3. The high priority group to be converted can be up to 4 channels and can be arranged in a specific sequence by configuring the ADCHLST register. This mode is provided to convert data for the high priority group with a short sequence, named as the A/D high priority conversion subgroup, each time a trigger event occurs. The subgroup length is defined in the ADHSUBL [1:0] field to specify the high priority subgroup length. In the Discontinuous Conversion Mode the A/D converter can be started by a software trigger, a comparator output transition event, an external EXTI event or a TM functional event for high priority group determined by the high priority Trigger Control Register ADCHTCR and the high priority Trigger Source Register ADCHTSR.

In the Discontinuous Conversion Mode, the A/D Converter will start to convert the next n conversions when a trigger event occurs. Here the number n is the subgroup length defined by the ADHSUBL field. The channels to be converted with a specific sequence are specified in the ADCHLST register. After n conversions have finished, the high priority subgroup EOC interrupt raw flag ADIRAWHG in the ADCIRAW register will be asserted. The A/D converter will then not continue to perform the next n conversions until the next trigger event occurs. The conversion cycle will finish after all the high priority group channels of which the total number is defined by the ADHSEQ[1:0] bits in the ADCHCONV register have finished conversion and the high priority cycle EOC interrupt raw flag ADIRAWHC in the ADCIRAW register will be asserted. If a new trigger event occurs after all the subgroup channels have been converted, i.e., a complete conversion cycle has been finished, the conversion will restart from the first subgroup.

Example:

A/D subgroup length = 2 (ADHSUBL=1) and sequence length = 3 (ADHSEQL=2), channels to be converted = 4, 7 and 1 - specific converting sequence as defined in the ADCHLST registers.

- Trigger 1: subgroup channels to be converted are CH4 and CH7 with the ADIRAWHG flag being asserted after subgroup EOC.
- Trigger 2: subgroup channel to be converted is CH1 with the ADIRAWHG flag being asserted after subgroup EOC. Also a Cycle end of conversion (EOC) interrupt raw flag ADIRAWHC will be asserted.
- Trigger 3: subgroup channels to be converted are CH4 and CH7 with the ADIRAWHG flag being asserted after subgroup EOC - conversion sequence restarts from the beginning.



**Figure 30. High Priority Group Discontinuous Conversion Mode**

## Start Conversion on External Event

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

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

The A/D converter can also be triggered to start a regular or high priority channel 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 TM 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 EXTIIn. If the external trigger enable bit, ADEXTI or ADHEXTI, is set to 1 and the corresponding EXTI line is selected by configuring the ADEXTIS or ADHEXTIS bit for regular or high priority channels respectively, the A/D converter will start a conversion when an EXTI line rising edge occurs.

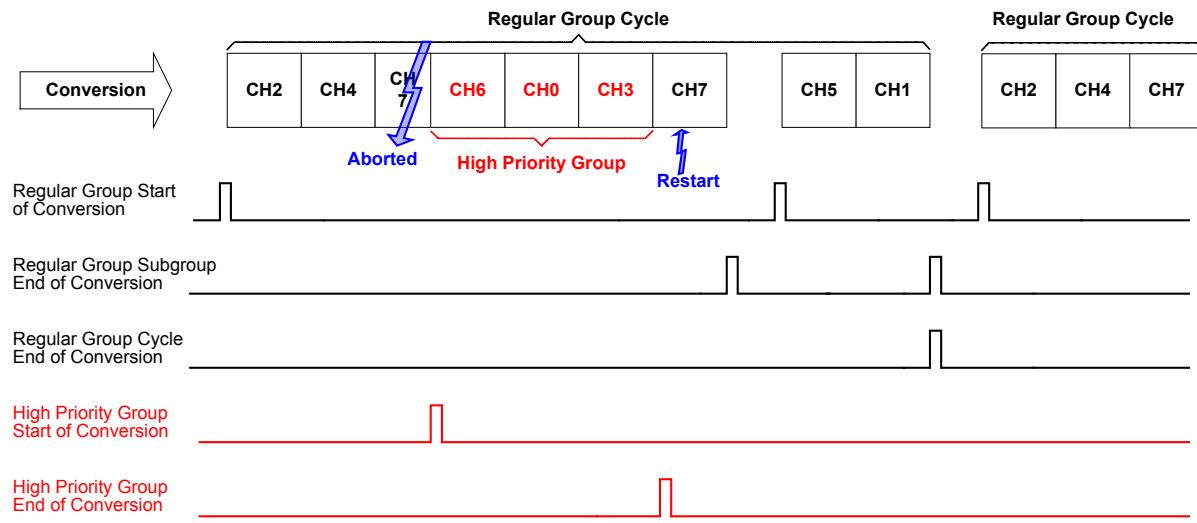
## High Priority Group Management

The high priority channels have a higher priority for A/D conversion. If, during a regular conversion process, a high priority channel trigger event occurs, then the current regular channel conversion will be aborted and the high priority channel conversion will be initiated.

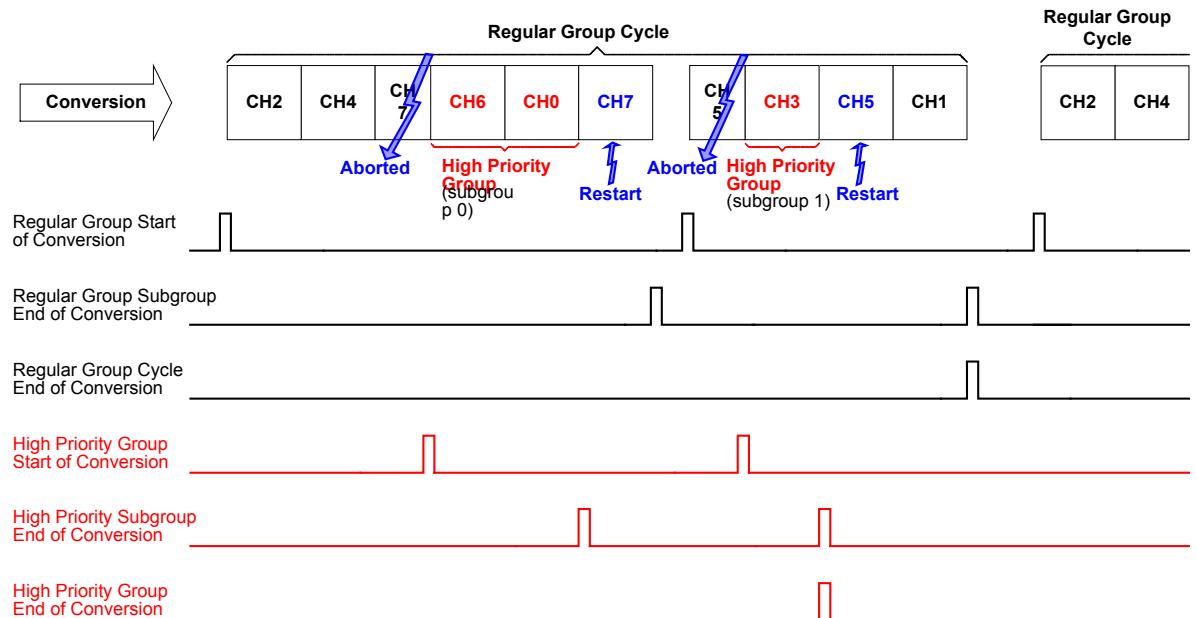
The high priority channel length to be converted depends upon the high priority group conversion mode as the high priority start trigger occurs. When a high priority start trigger occurs, the high priority channel length to be converted is defined by the ADHSQL field for the one shot conversion mode. For the discontinuous conversion mode, the length to be converted is the high priority subgroup length defined by ADHSUBL field. If the high priority group is configured to be operated in the continuous conversion mode, the high priority conversion will keep converting each channel in the high priority group repeatedly after a high priority start trigger occurs unless the high priority conversion mode has been changed.

When the high priority conversion has finished, the regular group conversion will then restart from the aborted channel location. Note that no matter what conversion mode the regular group operates in, the high priority conversion will always interrupt the current regular group conversion when a high priority start trigger occurs regardless of the high priority group conversion mode.

- ◆ Regular Group : Sequence Length=5, Subgroup Length=3, discontinuous conversion mode
- ◆ High Priority Group : Sequence Length=3, one shot conversion mode



- ◆ Regular Group : Sequence Length=5, Subgroup Length=3, discontinuous conversion mode
- ◆ High Priority Group : Sequence Length=3, Subgroup Length=2, discontinuous conversion mode



**Figure 31. High Priority Group Management**

## Sampling Time Setting

The conversion channel sampling time can be individually 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 can be sampled with a different sampling time. By modifying the ADSTn[7:0] bits in the ADCSTRn (n = 0~15) registers, 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 and Alignment

The ADC converted result has different output data formats by configuring the ADOFE and ADAL bits in the ADCOFRn (n = 0~15) registers. The ADOFE bit is used to control the offset cancellation function while the ADAL bit is used to select the data alignment format.

Each channel has a dedicated offset subtraction function whose offset value specified in the ADCOFRn (n = 0~15) registers can be defined by user. The original ADC conversion data located in the data register will always be an unsigned number between 0x0FFF and 0x0000 and only twelve bits are significant. If the ADOFEn bit in the ADCOFRn (n = 0~15) register is set to 1 then the offset subtraction is enabled. The significant conversion data in the ADCDRy (y= 0~15) or ADCHDRy (y= 0~3) registers will be a thirteen-bit format and the most significant bit (MSB) of the data stream is the sign bit.

**Table 27. Data Format in ADCDRy[15:0] (y= 0~15) and ADCHDRy[15:0] (y= 0~3)**

ADOFE	ADAL	Description	Data Format
0	0	Right aligned and unsigned	"0_0_0_0_d11_d10_d9_d8_d7_d6_d5_d4_d3_d2_d1_d0"
0	1	Left aligned and unsigned	"d11_d10_d9_d8_d7_d6_d5_d4_d3_d2_d1_d0_0_0_0"
1	0	Right aligned and signed after offset subtraction	"0_0_0_Sign bit_d11_d10_d9_d8_d7_d6_d5_d4_d3_d2_d1_d0"
1	1	Left aligned and signed after offset subtraction	"Sign bit_d11_d10_d9_d8_d7_d6_d5_d4_d3_d2_d1_d0_0_0_0"

## 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 ADIEL or ADIEU bit in the ADCIER 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 both regular and high priority groups. A single sample EOC event will occur and the single sample EOC interrupt raw flag, ADIRAWS or ADIRAWHS bit 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 or ADIRAWHG bit 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 or ADIRAWHC bit 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 bits, ADIEC, ADIEHC, ADIEG, ADIEHG, ADIES or ADIEHS 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 or ADCHDRy register and the value of the data valid flag named as ADVLDy or ADHVLDy will be changed from low to high. The converted data should be read by the application program, after which the data valid flag ADVLDy or ADHVLDy will be automatically changed from high to low. Otherwise, a data overwrite event will occur and the data overwrite interrupt raw flag ADIRAWO or ADIRAWHO 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 or ADIEHO 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 flags 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 flags and interrupt status bits. These bits are automatically cleared to 0 by hardware after being set to 1.

## PDMA Request

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

## Register Map

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

**Table 28. A/D Converter Register Map**

Register	Offset	Description	Reset Value
<b>ADC Base Address = 0x4001_0000</b>			
ADCRST	0x004	ADC Reset Register	0x0000_0000
ADCCONV	0x008	ADC Regular Conversion Mode Register	0x0000_0000
ADCHCONV	0x00C	ADC High Priority Conversion Mode Register	0x0000_0000
ADCLST0	0x010	ADC Regular Conversion List Register 0	0x0000_0000
ADCLST1	0x014	ADC Regular Conversion List Register 1	0x0000_0000
ADCLST2	0x018	ADC Regular Conversion List Register 2	0x0000_0000
ADCLST3	0x01C	ADC Regular Conversion List Register 3	0x0000_0000
ADCHLST	0x020	ADC High Priority Conversion List Register	0x0000_0000
ADCOFR0	0x030	ADC Input 0 Offset Register	0x0000_0000
ADCOFR1	0x034	ADC Input 1 Offset Register	0x0000_0000
ADCOFR2	0x038	ADC Input 2 Offset Register	0x0000_0000
ADCOFR3	0x03C	ADC Input 3 Offset Register	0x0000_0000
ADCOFR4	0x040	ADC Input 4 Offset Register	0x0000_0000
ADCOFR5	0x044	ADC Input 5 Offset Register	0x0000_0000
ADCOFR6	0x048	ADC Input 6 Offset Register	0x0000_0000
ADCOFR7	0x04C	ADC Input 7 Offset Register	0x0000_0000
ADCOFR8	0x050	ADC Input 8 Offset Register	0x0000_0000
ADCOFR9	0x054	ADC Input 9 Offset Register	0x0000_0000
ADCOFR10	0x058	ADC Input 10 Offset Register	0x0000_0000
ADCOFR11	0x05C	ADC Input 11 Offset Register	0x0000_0000
ADCOFR12	0x060	ADC Input 12 Offset Register	0x0000_0000
ADCOFR13	0x064	ADC Input 13 Offset Register	0x0000_0000
ADCOFR14	0x068	ADC Input 14 Offset Register	0x0000_0000
ADCOFR15	0x06C	ADC Input 15 Offset Register	0x0000_0000
ADCSTR0	0x070	ADC Input 0 Sampling Time Register	0x0000_0000
ADCSTR1	0x074	ADC Input 1 Sampling Time Register	0x0000_0000
ADCSTR2	0x078	ADC Input 2 Sampling Time Register	0x0000_0000
ADCSTR3	0x07C	ADC Input 3 Sampling Time Register	0x0000_0000
ADCSTR4	0x080	ADC Input 4 Sampling Time Register	0x0000_0000
ADCSTR5	0x084	ADC Input 5 Sampling Time Register	0x0000_0000
ADCSTR6	0x088	ADC Input 6 Sampling Time Register	0x0000_0000

Register	Offset	Description	Reset Value
ADCSTR7	0x08C	ADC Input 7 Sampling Time Register	0x0000_0000
ADCSTR8	0x090	ADC Input 8 Sampling Time Register	0x0000_0000
ADCSTR9	0x094	ADC Input 9 Sampling Time Register	0x0000_0000
ADCSTR10	0x098	ADC Input 10 Sampling Time Register	0x0000_0000
ADCSTR11	0x09C	ADC Input 11 Sampling Time Register	0x0000_0000
ADCSTR12	0x0A0	ADC Input 12 Sampling Time Register	0x0000_0000
ADCSTR13	0x0A4	ADC Input 13 Sampling Time Register	0x0000_0000
ADCSTR14	0x0A8	ADC Input 14 Sampling Time Register	0x0000_0000
ADCSTR15	0x0AC	ADC Input 15 Sampling Time Register	0x0000_0000
ADCDR0	0x0B0	ADC Regular Conversion Data Register 0	0x0000_0000
ADCDR1	0x0B4	ADC Regular Conversion Data Register 1	0x0000_0000
ADCDR2	0x0B8	ADC Regular Conversion Data Register 2	0x0000_0000
ADCDR3	0x0BC	ADC Regular Conversion Data Register 3	0x0000_0000
ADCDR4	0x0C0	ADC Regular Conversion Data Register 4	0x0000_0000
ADCDR5	0x0C4	ADC Regular Conversion Data Register 5	0x0000_0000
ADCDR6	0x0C8	ADC Regular Conversion Data Register 6	0x0000_0000
ADCDR7	0x0CC	ADC Regular Conversion Data Register 7	0x0000_0000
ADCDR8	0x0D0	ADC Regular Conversion Data Register 8	0x0000_0000
ADCDR9	0x0D4	ADC Regular Conversion Data Register 9	0x0000_0000
ADCDR10	0x0D8	ADC Regular Conversion Data Register 10	0x0000_0000
ADCDR11	0x0DC	ADC Regular Conversion Data Register 11	0x0000_0000
ADCDR12	0x0E0	ADC Regular Conversion Data Register 12	0x0000_0000
ADCDR13	0x0E4	ADC Regular Conversion Data Register 13	0x0000_0000
ADCDR14	0x0E8	ADC Regular Conversion Data Register 14	0x0000_0000
ADCDR15	0x0EC	ADC Regular Conversion Data Register 15	0x0000_0000
ADCHDR0	0x0F0	ADC High Priority Conversion Data Register 0	0x0000_0000
ADCHDR1	0x0F4	ADC High Priority Conversion Data Register 1	0x0000_0000
ADCHDR2	0x0F8	ADC High Priority Conversion Data Register 2	0x0000_0000
ADCHDR3	0x0FC	ADC High Priority Conversion Data Register 3	0x0000_0000
ADCTCR	0x100	ADC Regular Trigger Control Register	0x0000_0000
ADCTS	0x104	ADC Regular Trigger Source Register	0x0000_0000
ADCHTCR	0x110	ADC High Priority Trigger Control Register	0x0000_0000
ADCHTSR	0x114	ADC High Priority Trigger Source Register	0x0000_0000
ADCWCR	0x120	ADC Watchdog Control Register	0x0000_0000
ADCLTR	0x124	ADC Watchdog Lower Threshold Register	0x0000_0000
ADCUTR	0x128	ADC Watchdog Upper Threshold Register	0x0000_0000
ADCIER	0x130	ADC Interrupt Enable Register	0x0000_0000
ADCIRAW	0x134	ADC Interrupt Raw Status Register	0x0000_0000
ADCISR	0x138	ADC Interrupt Status Register	0x0000_0000
ADCICLR	0x13C	ADC Interrupt Clear Register	0x0000_0000
ADCDMAR	0x140	ADC DMA Request Register	0x0000_0000

## Register Descriptions

### ADC Reset Register – ADCRST

This register is used to reset the A/D Converter by software.

Offset: 0004

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							ADRST RW 0	

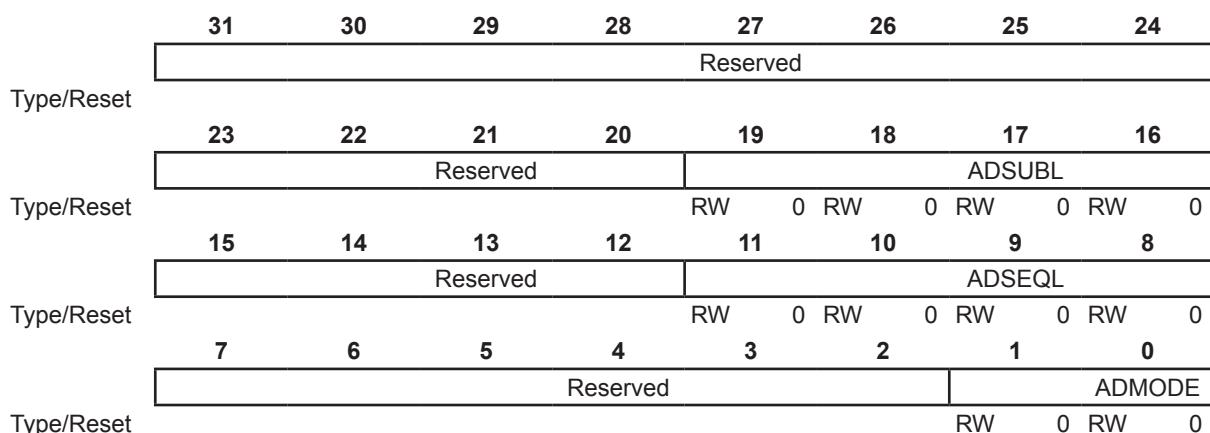
Bits	Field	Descriptions
[0]	ADRST	ADC Software Reset 0: No reset 1: Reset A/D converter except for the A/D Converter controller.

## ADC Regular Conversion Mode Register – ADCCONV

This register specifies the mode setting, sequence length and subgroup length of the A/D Converter regular group conversion. Note that once the content of the ADCCONV register is changed, any regular conversion presently in progress will be aborted and A/D Converter will be reset. The application program has to wait for at least one A/D converter clock CK\_ADC before issuing the next command.

Offset: 0008

Reset value: 0x0000\_0000



Bits	Field	Descriptions															
[19:16]	ADSUBL	<p>ADC Regular Conversion Subgroup Length</p> <p>The ADSUBL field specifies the conversion channel length of each subgroup for the regular group in the discontinuous mode. The subgroup length is equal to the ADSUBL field value plus 1. If the regular sequence length (ADSEQL [3:0] + 1) is not a multiple of the regular subgroup length (ADSUBL [3:0] + 1), the last subgroup will be the rest of the regular group channels that have not been converted.</p>															
[11:8]	ADSEQL	<p>ADC Regular Conversion Sequence Length</p> <p>0x00: The channel specified by the ADSEQ0 field in the ADCLST0 register will be converted.</p> <p>Others: Sequence Length = ADSEQL [3:0] + 1.</p> <p>The ADSEQL field specifies the conversion length of the whole sequence for the regular group. The sequence length is equal to the ADSEQL value plus 1.</p>															
[1:0]	ADMODE	<p>ADC Regular Conversion Mode Select</p> <table border="1"> <thead> <tr> <th>ADMODE [1:0]</th> <th>Mode</th> <th>Descriptions</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>One shot mode</td> <td>After a start trigger, the conversion will be executed on the regular channels for the whole sequence once.</td> </tr> <tr> <td>01</td> <td>Reserved</td> <td></td> </tr> <tr> <td>10</td> <td>Continuous mode</td> <td>After a start trigger, the conversion will be executed on the regular channels for the whole sequence continuously until the conversion mode is changed.</td> </tr> <tr> <td>11</td> <td>Discontinuous mode</td> <td>After a start trigger, the conversion will be executed on the current regular subgroup. When the last subgroup is finished, the conversion restarts from the first subgroup if a start trigger occurs.</td> </tr> </tbody> </table>	ADMODE [1:0]	Mode	Descriptions	00	One shot mode	After a start trigger, the conversion will be executed on the regular channels for the whole sequence once.	01	Reserved		10	Continuous mode	After a start trigger, the conversion will be executed on the regular channels for the whole sequence continuously until the conversion mode is changed.	11	Discontinuous mode	After a start trigger, the conversion will be executed on the current regular subgroup. When the last subgroup is finished, the conversion restarts from the first subgroup if a start trigger occurs.
ADMODE [1:0]	Mode	Descriptions															
00	One shot mode	After a start trigger, the conversion will be executed on the regular channels for the whole sequence once.															
01	Reserved																
10	Continuous mode	After a start trigger, the conversion will be executed on the regular channels for the whole sequence continuously until the conversion mode is changed.															
11	Discontinuous mode	After a start trigger, the conversion will be executed on the current regular subgroup. When the last subgroup is finished, the conversion restarts from the first subgroup if a start trigger occurs.															

## ADC High Priority Conversion Mode Register – ADCHCONV

This register specifies the mode setting, sequence length and subgroup length of the A/D Converter high priority conversion. Note that once the content of the ADCHCONV register is changed, the high priority conversion presently in progress will be aborted and the A/D Converter will be reset. The application program has to wait for at least one A/D converter clock CK\_ADC before issuing the next command.

Offset: 000C

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions															
[17:16]	ADHSUBL	ADC High Priority Conversion Subgroup Length The ADHSUBL field specifies the conversion channel length of each subgroup for the high priority group in the discontinuous mode. The subgroup length is equal to the ADHSUBL field value plus 1. If the high priority sequence length (ADHSEQL [1:0] + 1) is not a multiple of the high priority subgroup length (ADHSUBL [1:0] + 1), the last subgroup will be the rest of the high priority group channels that have not been converted.															
[9:8]	ADHSEQL	ADC High Priority Conversion Length 0x00: The channel specified by the ADHSEQ0 field in the ADCHLST register will be converted. Others: Sequence Length = ADHSEQL [1:0] + 1. The ADHSEQL field specifies the conversion length of the whole sequence for the high priority group. The sequence length is equal to the ADHSEQL value plus 1.															
[1:0]	ADHMODE	ADC High Priority Conversion Mode Select <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>ADHMODE [1:0]</th> <th>Mode</th> <th>Descriptions</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>One shot mode</td> <td>After a start trigger, the conversion will be executed on the whole sequence of the high priority channels once.</td> </tr> <tr> <td>01</td> <td>Reserved</td> <td></td> </tr> <tr> <td>10</td> <td>Continuous mode</td> <td>After a start trigger, the conversion will be executed on the high priority channels for the whole sequence continuously until the conversion mode is changed.</td> </tr> <tr> <td>11</td> <td>Discontinuous mode</td> <td>After a trigger, the conversion will be executed on the current high priority subgroup. When the last subgroup is finished, the conversion restarts from the first subgroup if a start trigger occurs.</td> </tr> </tbody> </table>	ADHMODE [1:0]	Mode	Descriptions	00	One shot mode	After a start trigger, the conversion will be executed on the whole sequence of the high priority channels once.	01	Reserved		10	Continuous mode	After a start trigger, the conversion will be executed on the high priority channels for the whole sequence continuously until the conversion mode is changed.	11	Discontinuous mode	After a trigger, the conversion will be executed on the current high priority subgroup. When the last subgroup is finished, the conversion restarts from the first subgroup if a start trigger occurs.
ADHMODE [1:0]	Mode	Descriptions															
00	One shot mode	After a start trigger, the conversion will be executed on the whole sequence of the high priority channels once.															
01	Reserved																
10	Continuous mode	After a start trigger, the conversion will be executed on the high priority channels for the whole sequence continuously until the conversion mode is changed.															
11	Discontinuous mode	After a trigger, the conversion will be executed on the current high priority subgroup. When the last subgroup is finished, the conversion restarts from the first subgroup if a start trigger occurs.															

## ADC Regular Conversion List Register 0 – ADCLST0

This register specifies the conversion sequence order No.0~No.3 of the A/D Converter regular group.

Offset: 0010

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[28:24]	ADSEQ3	ADC Regular Conversion Sequence Selector 3 Define the 3 <sup>rd</sup> A/D Converter input channel of the regular conversion sequence. 0x00: ADC_IN0 0x01: ADC_IN1 0x02: ADC_IN2 0x03: ADC_IN3 0x04: ADC_IN4 0x05: ADC_IN5 0x06: ADC_IN6 0x07: ADC_IN7 0x08: ADC_IN8 0x09: ADC_IN9 0x0A: ADC_IN10 0x0B: ADC_IN11 0x0C: ADC_IN12 0x0D: ADC_IN13 0x0E: ADC_IN14 0x0F: ADC_IN15 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 abnormal A/D Converter operations.
[20:16]	ADSEQ2	ADC Regular Conversion Sequence Selector 2
[12:8]	ADSEQ1	ADC Regular Conversion Sequence Selector 1
[4:0]	ADSEQ0	ADC Regular Conversion Sequence Selector 0

## ADC Regular Conversion List Register 1 – ADCLST1

This register specifies the conversion sequence order No.4~No.7 of the A/D Converter regular group.

Offset: 0014

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[28:24]	ADSEQ7	ADC Regular Conversion Sequence Selector 7 Define the 7 <sup>th</sup> A/D Converter input channel of the regular conversion sequence. 0x00: ADC_IN0 0x01: ADC_IN1 0x02: ADC_IN2 0x03: ADC_IN3 0x04: ADC_IN4 0x05: ADC_IN5 0x06: ADC_IN6 0x07: ADC_IN7 0x08: ADC_IN8 0x09: ADC_IN9 0x0A: ADC_IN10 0x0B: ADC_IN11 0x0C: ADC_IN12 0x0D: ADC_IN13 0x0E: ADC_IN14 0x0F: ADC_IN15 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 abnormal A/D Converter operations.
[20:16]	ADSEQ6	ADC Regular Conversion Sequence Selector 6
[12:8]	ADSEQ5	ADC Regular Conversion Sequence Selector 5
[4:0]	ADSEQ4	ADC Regular Conversion Sequence Selector 4

## ADC Regular Conversion List Register 2 – ADCLST2

This register specifies the conversion sequence order No.8~No.11 of the A/D Converter regular group.

Offset: 0018

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24		
Type/Reset		Reserved				ADSEQ11				
	23	22	21	20	19	18	17	16		
Type/Reset		Reserved				ADSEQ10				
	15	14	13	12	11	10	9	8		
Type/Reset		Reserved				ADSEQ9				
	7	6	5	4	3	2	1	0		
Type/Reset		Reserved				ADSEQ8				

Bits	Field	Descriptions
[28:24]	ADSEQ11	ADC Regular Conversion Sequence Selector 11 Define the 11 <sup>th</sup> A/D converter input channel of the regular conversion sequence. 0x00: ADC_IN0 0x01: ADC_IN1 0x02: ADC_IN2 0x03: ADC_IN3 0x04: ADC_IN4 0x05: ADC_IN5 0x06: ADC_IN6 0x07: ADC_IN7 0x08: ADC_IN8 0x09: ADC_IN9 0x0A: ADC_IN10 0x0B: ADC_IN11 0x0C: ADC_IN12 0x0D: ADC_IN13 0x0E: ADC_IN14 0x0F: ADC_IN15 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 abnormal A/D Converter operations.
[20:16]	ADSEQ10	ADC Regular Conversion Sequence Selector 10
[12:8]	ADSEQ9	ADC Regular Conversion Sequence Selector 9
[4:0]	ADSEQ8	ADC Regular Conversion Sequence Selector 8

## ADC Regular Conversion List Register 3 – ADCLST3

This register specifies the conversion sequence order No.12~No.15 of the A/D Converter regular group.

Offset: 001C

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24		
Type/Reset		Reserved				ADSEQ15				
	23	22	21	20	19	18	17	16		
Type/Reset		Reserved			ADSEQ14					
	15	14	13	12	11	10	9	8		
Type/Reset		Reserved			ADSEQ13					
	7	6	5	4	3	2	1	0		
Type/Reset		Reserved			ADSEQ12					

Bits	Field	Descriptions
[28:24]	ADSEQ15	ADC Regular Conversion Sequence Selector 15 Define the 15 <sup>th</sup> A/D Converter input channel of the regular conversion sequence. 0x00: ADC_IN0 0x01: ADC_IN1 0x02: ADC_IN2 0x03: ADC_IN3 0x04: ADC_IN4 0x05: ADC_IN5 0x06: ADC_IN6 0x07: ADC_IN7 0x08: ADC_IN8 0x09: ADC_IN9 0x0A: ADC_IN10 0x0B: ADC_IN11 0x0C: ADC_IN12 0x0D: ADC_IN13 0x0E: ADC_IN14 0x0F: ADC_IN15 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 abnormal A/D Converter operations.
[20:16]	ADSEQ14	ADC Regular Conversion Sequence Selector 14
[12:8]	ADSEQ13	ADC Regular Conversion Sequence Selector 13
[4:0]	ADSEQ12	ADC Regular Conversion Sequence Selector 12

## ADC High Priority Conversion List Register – ADCHLST

This register specifies the conversion sequence order No.0~No.3 of the A/D Converter high priority group.

Offset: 0020

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset		Reserved		ADHSEQ3				
				RW	0	RW	0	RW
Type/Reset	23	22	21	20	19	18	17	16
		Reserved		ADHSEQ2				
Type/Reset				RW	0	RW	0	RW
	15	14	13	12	11	10	9	8
Type/Reset		Reserved		ADHSEQ1				
				RW	0	RW	0	RW
Type/Reset	7	6	5	4	3	2	1	0
		Reserved		ADHSEQ0				
Type/Reset				RW	0	RW	0	RW

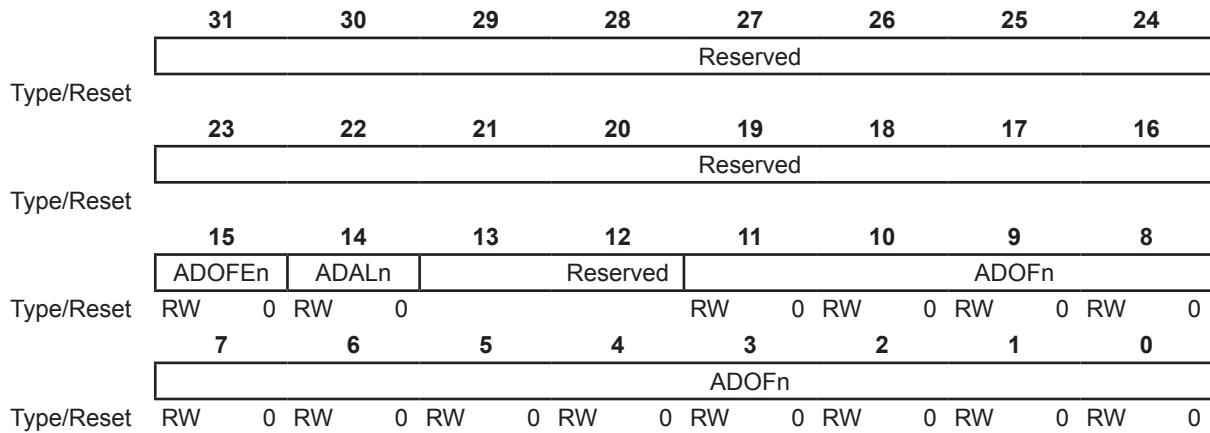
Bits	Field	Descriptions
[28:24]	ADHSEQ3	ADC High Priority Conversion Sequence Selector 3 Define the 1 <sup>st</sup> A/D Converter input channel of the high priority conversion sequence. 0x0: ADC_IN0 0x1: ADC_IN1 0x2: ADC_IN2 0x3: ADC_IN3 0x4: ADC_IN4 0x5: ADC_IN5 0x6: ADC_IN6 0x7: ADC_IN7 0x8: ADC_IN8 0x9: ADC_IN9 0xA: ADC_IN10 0xB: ADC_IN11 0xC: ADC_IN12 0xD: ADC_IN13 0xE: ADC_IN14 0xF: ADC_IN15 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 abnormal A/D Converter operations.
[20:16]	ADHSEQ2	ADC High Priority Conversion Sequence Selector 2
[12:8]	ADHSEQ1	ADC High Priority Conversion Sequence Selector 1
[4:0]	ADHSEQ0	ADC High Priority Conversion Sequence Selector 0

## ADC Input Offset Register n – ADCOFRn, n = 0 ~ 15

This register specifies the A/D converter input channel n offset value together with the offset cancellation function enable control.

Offset: 0030 ~ 006C

Reset value: 0x0000\_0000



Bits	Field	Descriptions
[15]	ADOFEn	ADC Input Channel n Offset Cancellation Enable (n = 0 ~ 15) 0: ADC_INn offset cancellation is disabled 1: ADC_INn offset cancellation is enabled
[14]	ADALn	ADC Input Channel n Data Alignment Select (n = 0 ~ 15) 0: Right aligned 1: Left aligned This bit is used to determine the data aligned format of the input channel n conversion data.
[11:0]	ADOFn	ADC Input Channel n Offset Value (n = 0 ~ 15) The data value read from the ADC data register, ADCDRn, will be the result which is obtained by subtracting this offset value on Channel n, ADC_INn, from the raw A/D conversion data if the offset cancellation is enabled.

## ADC Input Sampling Time Register n – ADCSTRn, n = 0 ~ 15

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

Offset: 0070 ~ 00AC

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[7:0]	ADSTn	ADC Input Channel n Sampling Time (n = 0 ~ 15) Sampling time = (ADSTn [7:0] + 1.5) × A/D converter clock period.

### ADC Regular Conversion Data Register y – ADCDRy, y = 0 ~ 15

This register is used to store the conversion data of the regular conversion sequence order No.n which is specified in the ADSEQn field.

Offset: 00B0 ~ 00EC

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[31]	ADVLDy	ADC Regular Conversion Data of the Sequence Order No.n Valid Bit (y = 0 ~ 15) 0: Data is invalid or have been read 1: New data is valid
[15:0]	ADDy	ADC Regular Conversion Data of the Sequence Order No.n (y = 0 ~ 15) The data is the regular conversion result of the sequence order No.n specified in the ADSEQn field (n=0 ~ 15).

## ADC High Priority Conversion Data Register y – ADCHDRy, y = 0 ~ 3

This register is used to store the conversion data of the high priority conversion sequence order No.n which is specified in the ADHSEQn field.

Offset: 00F0 ~ 00FC

Reset value: 0x0000\_0000

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

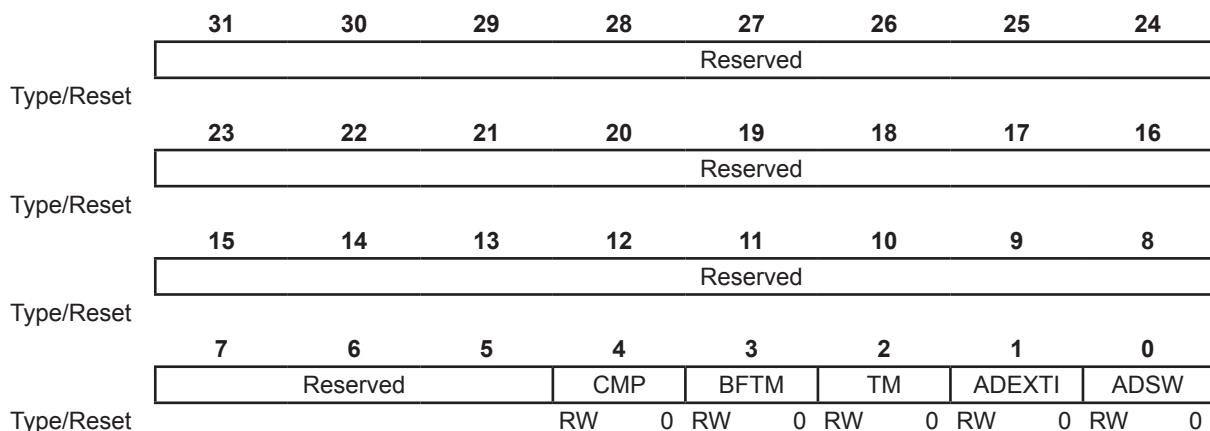
Bits	Field	Descriptions
[31]	ADHVLDy	ADC High Priority Conversion Data of the Sequence Order No.n Valid Bit (y = 0 ~ 3) 0: Data is invalid or have been read 1: New data is valid
[15:0]	ADHDy	ADC High Priority Conversion Data of the Sequence Order No.n (y = 0 ~ 3) The data is the high priority conversion result of the sequence order No.n specified in ADHSEQn field (n = 0 ~ 3).

## ADC Regular Trigger Control Register – ADCTCR

This register contains the A/D converter start conversion trigger enable bits of the regular conversion.

Offset: 0100

Reset value: 0x0000\_0000



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

## ADC Regular Trigger Source Register – ADCTSR

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

Offset: 0104

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							TME
						RW	0	RW
Type/Reset	23	22	21	20	19	18	17	16
	Reserved			CMPS	BFTMS	TMS		
Type/Reset				RW	0	RW	0	RW
Type/Reset	15	14	13	12	11	10	9	8
	Reserved				ADEXTIS			
Type/Reset					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 Regular 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
[20]	CMPS	CMP Trigger Selection of ADC Regular Conversion 0: CMP0 1: CMP1
[19]	BFTMS	BFTM Trigger Timer Selection of ADC Regular Conversion 0: BFTM0 1: BFTM1
[18:16]	TMS	GPTM or MCTM Trigger Timer Selection of ADC Regular Conversion 000: MCTM0 001: MCTM1 010: GPTM0 011: GPTM1 Others: Reserved – Should not be used to avoid unpredictable results
[11:8]	ADEXTIS	EXTI Trigger Source Selection of ADC Regular Conversion 0x00: EXTI line 0 0x01: EXTI line 1 ... 0x0F: EXTI line 15
[0]	ADSC	ADC Regular Conversion Software Trigger Bit 0: No operation 1: Start the regular conversion immediately This bit is set by software to start the regular conversion manually and then cleared by hardware automatically after conversion started.

## ADC High Priority Trigger Control Register – ADCHTCR

This register contains the A/D Converter start conversion trigger enable bits of the high priority conversion.

Offset: 0110

Reset value: 0x0000\_0000



Bits	Field	Descriptions
[3]	HBFTM	ADC High Priority Conversion BFTM Event Trigger Enable control 0: Disable high priority conversion trigger by BFTM events 1: Enable high priority conversion trigger by BFTM events
[2]	HTM	ADC High Priority Conversion Trigger by GPTM or MCTM Event 0: Disable high priority conversion trigger by GPTM or MCTM events 1: Enable high priority conversion trigger by GPTM or MCTM events
[1]	ADHEXTI	ADC High Priority Conversion Trigger by EXTI Event 0: Disable high priority conversion trigger by EXTI lines 1: Enable high priority conversion trigger by EXTI lines
[0]	ADHSW	ADC High Priority Conversion Trigger by Software 0: Disable high priority conversion trigger by software trigger bit 1: Enable high priority conversion trigger by software trigger bit

## ADC High Priority Trigger Source Register – ADCHTSR

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

Offset: 0114

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[26:24]	HTME	GPTM or MCTM Trigger Event Selection of ADC High Priority 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]	HBFTMS	BFTM Trigger Timer Selection of ADC High Priority Conversion 0: BFTM0 1: BFTM1
[18:16]	HTMS	GPTM or MCTM Trigger Timer Selection of ADC High Priority Conversion 000: MCTM0 001: MCTM1 010: GPTM0 011: GPTM1 Others: Reserved – Should not be used to avoid unpredictable results
[11:8]	ADHEXTIS	EXTI Trigger Source Selection of ADC High Priority Conversion 0x00: EXTI line 0 0x01: EXTI line 1 ... 0x0F: EXTI line 15
[0]	ADHSC	ADC High Priority Conversion Software Trigger Bit 0: No operation 1: Start the high priority conversion immediately This bit is set by software to start the high priority conversion manually and then 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: 0120

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[27:24]	ADUCH	<p>Upper Threshold Channel Status</p> <p>0000: ADC_IN0 converted data is higher than the upper threshold</p> <p>0001: ADC_IN1 converted data is higher than the upper threshold</p> <p>...</p> <p>1111: ADC_IN15 converted data is higher than the upper threshold</p> <p>If one of these status bits is set to 1 by the watchdog monitor function, the status field 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.</p>
[19:16]	ADLCH	<p>Lower Threshold Channel Status</p> <p>0000: ADC_IN0 converted data is lower than the lower threshold</p> <p>0001: ADC_IN1 converted data is lower than the lower threshold</p> <p>...</p> <p>1111: ADC_IN15 converted data is lower than the lower threshold</p> <p>If one of these status bits is set to 1 by the watchdog monitor function, the status field 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.</p>
[11:8]	ADWCH	<p>ADC Watchdog Specific Channel Selection for watchdog function</p> <p>0000: ADC_IN0 is monitored</p> <p>0001: ADC_IN1 is monitored</p> <p>...</p> <p>1111: ADC_IN15 is monitored</p>
[2]	ADWALL	<p>ADC Watchdog Specific or All Channel Selection for watchdog function</p> <p>0: Only the channel which specified by the ADWCH field is monitored</p> <p>1: All channels are monitored</p>
[1]	ADWUE	<p>ADC Watchdog Upper Threshold Monitor Enable Bit</p> <p>0: Disable upper threshold monitor function</p> <p>1: Enable upper threshold monitor function</p>

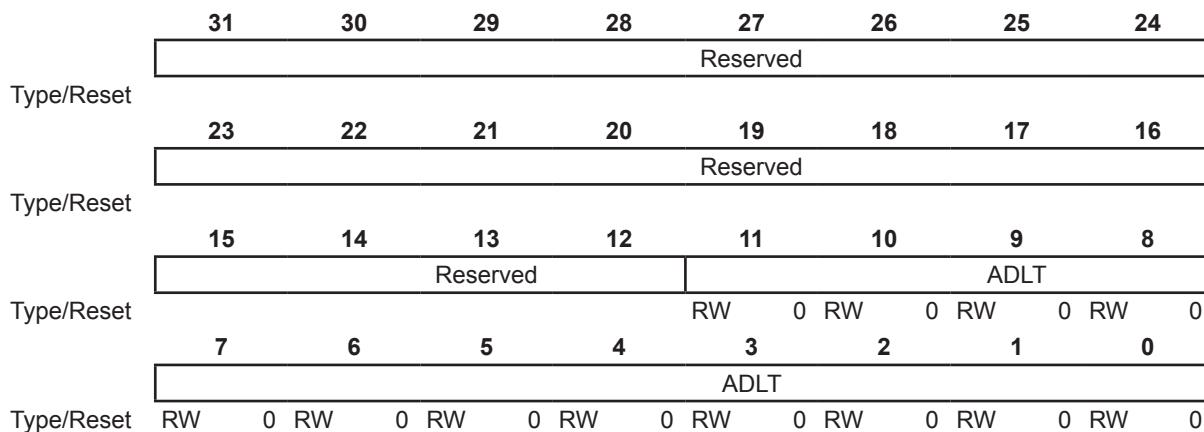
Bits	Field	Descriptions
[0]	ADWLE	ADC Watchdog Lower Threshold Monitor Enable Bit 0: Disable lower threshold monitor function 1: Enable lower threshold monitor function

### ADC Watchdog Lower Threshold Register – ADCLTR

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

Offset: 0124

Reset value: 0x0000\_0000



Bits	Field	Descriptions
[11:0]	ADLT	ADC Watchdog Lower Threshold Value Specify the lower threshold for the ADC watchdog low threshold monitor function.

## ADC Watchdog Upper Threshold Register – ADCUTR

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

Offset: 0128

Reset value: 0x0000\_0000

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

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

## ADC Interrupt Enable Register – ADCIER

This register contains the A/D Converter interrupt enable bits.

Offset: 0130

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved					ADIEHO	ADIEO	
						RW	0	RW 0
Type/Reset	Reserved					ADIEU	ADIEL	
						RW	0	RW 0
Type/Reset	Reserved					ADIEHC	ADIEHG	ADIEHS
						RW	0	RW 0
Type/Reset	Reserved					ADIEC	ADIEG	ADIES
						RW	0	RW 0

Bits	Field	Descriptions
[25]	ADIEHO	ADC High Priority Data Register Overwrite Interrupt Enable 0: ADC high priority data register overwrite interrupt is disabled 1: ADC high priority data register overwrite interrupt is enabled
[24]	ADIEO	ADC Regular Data Register Overwrite Interrupt Enable 0: ADC regular data register overwrite interrupt is disabled 1: ADC regular 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
[10]	ADIEHC	ADC High Priority Cycle EOC Interrupt Enable 0: ADC high priority cycle end of conversion interrupt is disabled 1: ADC high priority cycle end of conversion interrupt is enabled
[9]	ADIEHG	ADC High Priority Subgroup EOC Interrupt Enable 0: ADC high priority subgroup end of conversion interrupt is disabled 1: ADC high priority subgroup end of conversion interrupt is enabled
[8]	ADIEHS	ADC High Priority Single EOC Interrupt Enable 0: ADC high priority single end of conversion interrupt is disabled 1: ADC high priority single end of conversion interrupt is enabled
[2]	ADIEC	ADC Regular Cycle EOC Interrupt Enable 0: ADC regular cycle end of conversion interrupt is disabled 1: ADC regular cycle end of conversion interrupt is enabled
[1]	ADIEG	ADC Regular Subgroup EOC Interrupt Enable 0: ADC regular subgroup end of conversion interrupt is disabled 1: ADC regular subgroup end of conversion interrupt is enabled
[0]	ADIES	ADC Regular Single EOC Interrupt Enable 0: ADC regular single end of conversion interrupt is disabled 1: ADC regular single end of conversion interrupt is enabled

## ADC Interrupt Raw Status Register – ADCIRAW

This register contains the A/D Converter interrupt raw status bits.

Offset: 0134

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved					ADIRAWHO	ADIRAWO	
						RO	0	RO
Type/Reset	Reserved					ADIRAWU	ADIRAWL	
						RO	0	RO
Type/Reset	Reserved					ADIRAWHC	ADIRAWHG	ADIRAWSHS
						RO	0	RO
Type/Reset	Reserved					ADIRAWC	ADIRAWG	ADIRAWS
						RO	0	RO

Bits	Field	Descriptions
[25]	ADIRAWHO	ADC High Priority Data Register Overwrite Interrupt Raw Status 0: ADC high priority data register overwrite event does not occur 1: ADC high priority data register overwrite event occurs
[24]	ADIRAWO	ADC Regular Data Register Overwrite Interrupt Raw Status 0: ADC regular data register overwrite event does not occur 1: ADC regular 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
[10]	ADIRAWHC	ADC High Priority Cycle EOC Interrupt Raw Status 0: ADC high priority cycle end of conversion event does not occur 1: ADC high priority cycle end of conversion event occurs
[9]	ADIRAWHG	ADC High Priority Subgroup EOC Interrupt Raw Status 0: ADC high priority subgroup end of conversion event does not occur 1: ADC high priority subgroup end of conversion event occurs
[8]	ADIRAWSHS	ADC High Priority Single EOC Interrupt Raw Status 0: ADC high priority single end of conversion event does not occur 1: ADC high priority single end of conversion event occurs
[2]	ADIRAWC	ADC Regular Cycle EOC Interrupt Raw Status 0: ADC regular cycle end of conversion event does not occur 1: ADC regular cycle end of conversion event occurs
[1]	ADIRAWG	ADC Regular Subgroup EOC Interrupt Raw Status 0: ADC regular subgroup end of conversion event does not occur 1: ADC regular subgroup end of conversion event occurs
[0]	ADIRAWS	ADC Regular Single EOC Interrupt Raw Status 0: ADC regular single end of conversion event does not occur 1: ADC regular single end of conversion event occurs

## ADC Interrupt Status Register – ADCISR

This register contains the A/D Converter interrupt 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: 0138

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[25]	ADISRHO	ADC High Priority Data Register Overwrite Interrupt Status 0: ADC high priority data register overwrite event does not occur or the related interrupt control is disabled. 1: ADC high priority data register overwrite event occurs as the related interrupt control is enabled.
[24]	ADISRO	ADC Regular Data Register Overwrite Interrupt Status 0: ADC regular data register overwrite event does not occur or the related interrupt control is disabled. 1: ADC regular data register overwrite event occurs as the related interrupt control is enabled.
[17]	ADISRU	ADC Watchdog Upper Threshold Interrupt Status 0: ADC watchdog upper threshold interrupt does not occur or the related interrupt control is disabled. 1: ADC watchdog upper threshold interrupt occurs as the related interrupt control is enabled.
[16]	ADISRL	ADC Watchdog Lower Threshold Interrupt Status 0: ADC watchdog lower threshold interrupt does not occur or the related interrupt control is disabled. 1: ADC watchdog lower threshold interrupt occurs as the related interrupt control is enabled.
[10]	ADISRHC	ADC High Priority Cycle EOC Interrupt Status 0: ADC high priority cycle end of conversion interrupt does not occur or the related interrupt control is disabled. 1: ADC high priority cycle end of conversion interrupt occurs as the related interrupt control is enabled.
[9]	ADISRHG	ADC High Priority Subgroup EOC Interrupt Status 0: ADC high priority subgroup end of conversion interrupt does not occur or the related interrupt control is disabled. 1: ADC high priority subgroup end of conversion interrupt occurs as the related interrupt control is enabled.

<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>
[8]	ADISRHS	ADC High Priority Single EOC Interrupt Status 0: ADC high priority single end of conversion interrupt does not occur or the related interrupt control is disabled. 1: ADC high priority single end of conversion interrupt occurs as the related interrupt control is enabled.
[2]	ADISRC	ADC Regular Cycle EOC Interrupt Status 0: ADC regular cycle end of conversion interrupt does not occur or the related interrupt control is disabled. 1: ADC regular cycle end of conversion interrupt occurs as the related interrupt control is enabled.
[1]	ADISRG	ADC Regular Subgroup EOC Interrupt Status 0: ADC regular subgroup end of conversion interrupt does not occur or the related interrupt control is disabled. 1: ADC regular subgroup end of conversion interrupt occurs as the related interrupt control is enabled.
[0]	ADISRS	ADC Regular Single EOC Interrupt Status 0: ADC regular single end of conversion interrupt does not occur or the related interrupt control is disabled. 1: ADC regular single end of conversion interrupt occurs as the related interrupt control is enabled.

## ADC Interrupt Clear Register – ADCICLRL

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

Offset: 013C

Reset value: 0x0000\_0000

Type/Reset	31	30	29	28	27	26	25	24
	Reserved					ADICLRRHO	ADICLRLRO	
						WO	0	WO
Type/Reset	23	22	21	20	19	18	17	16
	Reserved					ADICLRRU	ADICLRLR	
						WO	0	WO
Type/Reset	15	14	13	12	11	10	9	8
	Reserved					ADICLRRHC	ADICLRRHG	ADICLRRHS
						WO	0	WO
Type/Reset	7	6	5	4	3	2	1	0
	Reserved					ADICLRC	ADICLRG	ADICLRS
						WO	0	WO

Bits	Field	Descriptions
[25]	ADICLRRHO	ADC High Priority Data Register Overwrite Interrupt Status Clear Bit 0: No effect 1: Clear ADISRHO and ADIRAWHO bits
[24]	ADICLRLRO	ADC Regular Data Register Overwrite Interrupt Status Clear Bit 0: No effect 1: Clear ADISRO and ADIRAWO bits
[17]	ADICLRRU	ADC Watchdog Upper Threshold Interrupt Status Clear Bit 0: No effect 1: Clear ADISRU and ADIRAWU bits
[16]	ADICLRLR	ADC Watchdog Lower Threshold Interrupt Status Clear Bit 0: No effect 1: Clear ADISRL and ADIRAWL bits
[10]	ADICLRRHC	ADC High Priority Cycle EOC Interrupt Status Clear Bit 0: No effect 1: Clear ADISRHC and ADIRAWHC bits
[9]	ADICLRRHG	ADC High Priority Subgroup EOC Interrupt Status Clear Bit 0: No effect 1: Clear ADISRHG and ADIRAWHG bits
[8]	ADICLRRHS	ADC High Priority Single EOC Interrupt Status Clear Bit 0: No effect 1: Clear ADISRHS and ADIRAWHS bits
[2]	ADICLRC	ADC Regular Cycle EOC Interrupt Status Clear Bit 0: No effect 1: Clear ADISRC and ADIRAWC bits
[1]	ADICLRG	ADC Regular Subgroup EOC Interrupt Status Clear Bit 0: No effect 1: Clear ADISRG and ADIRAWG bits
[0]	ADICLRS	ADC Regular Single EOC Interrupt Status Clear Bit 0: No effect 1: Clear ADISRS and ADIRAWS bits

## ADC DMA Request Register – ADCDMAR

This register contains the A/D Converter DMA request enable bits.

Offset: 0140

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[10]	ADDMAHC	ADC High Priority Cycle EOC DMA Request Enable Bit 0: ADC high priority cycle end of conversion DMA request is disabled 1: ADC high priority cycle end of conversion DMA request is enabled
[9]	ADDMAHG	ADC High Priority Subgroup EOC DMA Request Enable Bit 0: ADC high priority subgroup end of conversion DMA request is disabled 1: ADC high priority subgroup end of conversion DMA request is enabled
[8]	ADDMAHS	ADC High Priority Single EOC DMA Request Enable Bit 0: ADC high priority single end of conversion DMA request is disabled 1: ADC high priority single end of conversion DMA request is enabled
[2]	ADDMAC	ADC Regular Cycle EOC DMA Request Enable Bit 0: ADC regular cycle end of conversion DMA request is disabled 1: ADC regular cycle end of conversion DMA request is enabled
[1]	ADDMAH	ADC Regular Subgroup EOC DMA Request Enable Bit 0: ADC regular subgroup end of conversion DMA request is disabled 1: ADC regular subgroup end of conversion DMA request is enabled
[0]	ADDMAS	ADC Regular Single EOC DMA Request Enable Bit 0: ADC regular single end of conversion DMA request is disabled 1: ADC regular single end of conversion DMA request is enabled

# 13 Comparator (CMP)

## Introduction

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

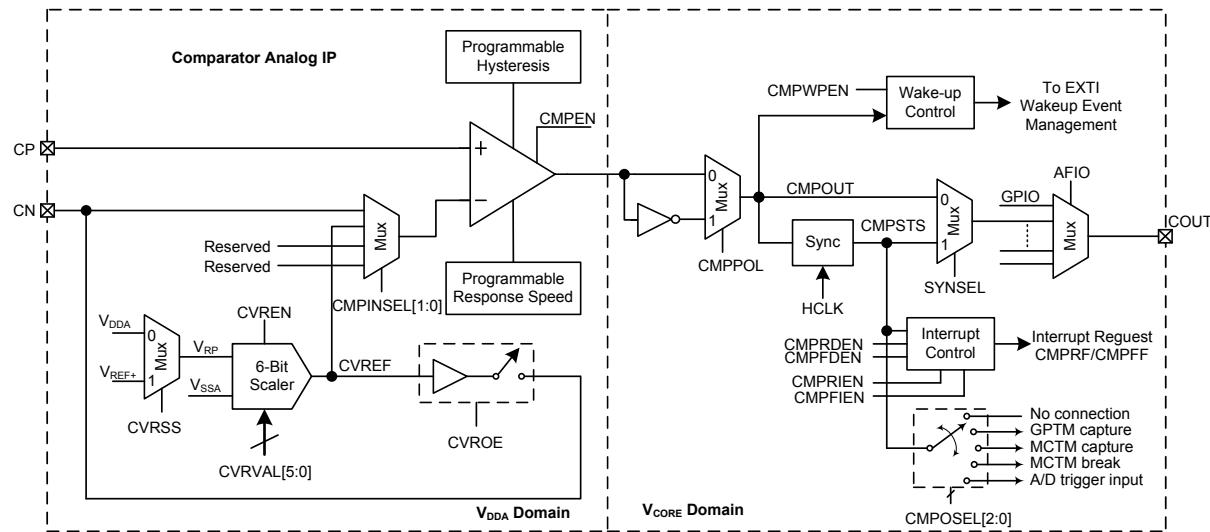


Figure 32. Comparator Block Diagram

## Features

- Rail-to-rail comparator
- Configurable negative inputs for flexible voltage selection
  - Dedicated I/O pin
  - Internal voltage reference provided by 6-bit Scaler
- Programmable hysteresis
- Programmable response speed
- Comparator output used as multiple timers or A/D Converter trigger inputs
- Programmable internal voltage reference provided by 6-bit Scaler
- Comparator output transition with interrupt generation and wake-up functions

## Function Descriptions

### Comparator Inputs and Output

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

### Comparator Voltage Reference

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

$$CVREF = CVRVAL * (V_{RP} - V_{SSA}) / 63$$

The 6-bit scaler positive reference voltage  $V_{RP}$  can come from either the  $V_{DDA}$  or the external  $V_{REF+}$  pin. The voltage source is selected by the  $CVRSS$  bit in the Comparator Control Register CMPCR. The  $CVREF$  output is used to provide a reference voltage for the analog comparator. It can be internally used or configured to connect to the CN pin by setting the  $CVROE$  bit in the Comparator Control Register CMPCR. The settling time of the comparator voltage reference must be considered when the  $CVREF$  output voltage is changed.

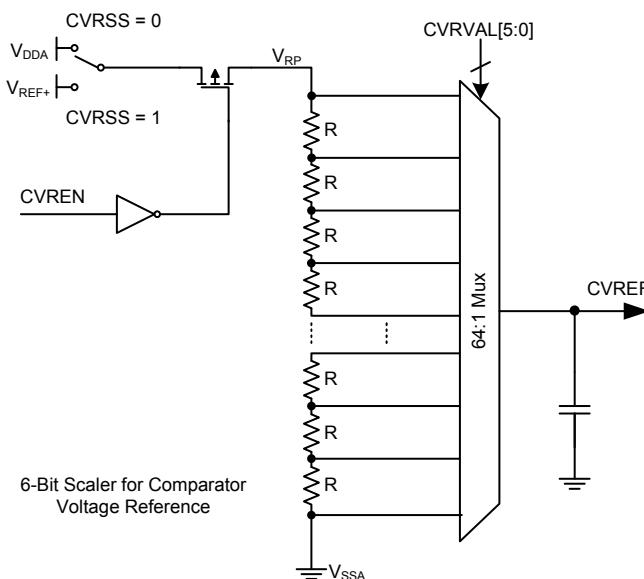
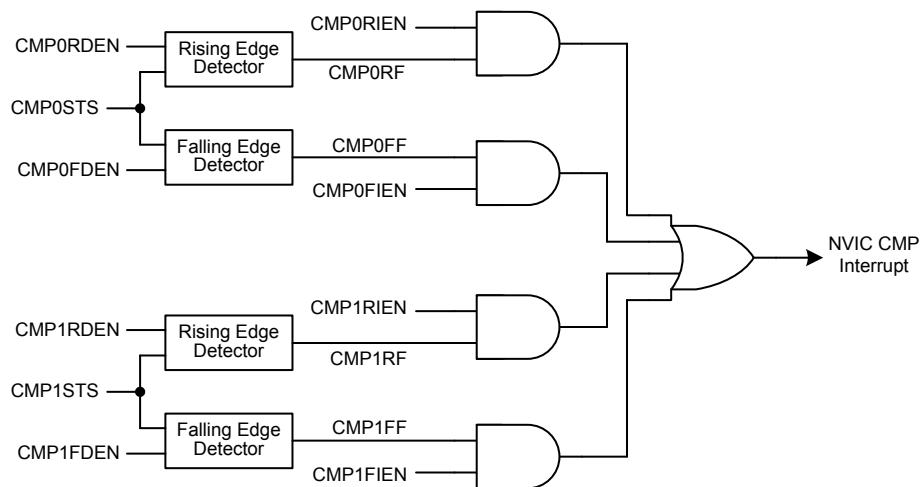


Figure 33. Comparator Voltage Reference Scaler Block Diagram

## Interrupts and Wake-up

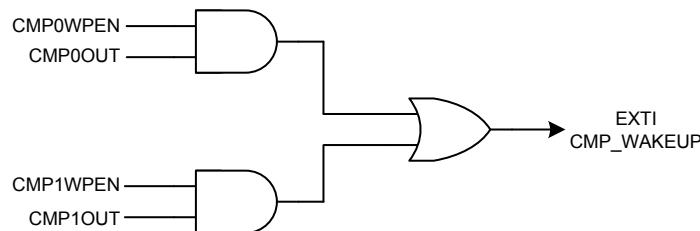
The comparator can generate an interrupt when its output changes to generates a rising or falling edge and its corresponding edge detection and interrupt control bit is enabled.

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



**Figure 34. Comparator Output Interrupt Structure**

The comparator outputs are also internally connected to the EXTI Wake-up Event Management unit. The comparator output rising transition is used to wakeup from the MCU Deep Sleep 1 when the comparator wake-up enable bit CMPWPEN is set in the Comparator Control Register CMPCR.



**Figure 35. Comparator Output Wake-up Structure**

## Power Mode and Hysteresis

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

The comparator supports four hysteresis levels to avoid spurious output transitions in case of noisy signals. The bit CMPHM[1:0] in the CMPCR register can be configured to obtain different hysteresis levels for the comparator.

## Comparator Write-Protected mechanism

As the comparator can be used for safety purposes, it is necessary to ensure that the comparator configurations will not be altered due to spurious register access or program counter corruption. For this purpose, the write protection is provided by writing a specific value into the PROTECT field in the Comparator Control Register CMPCR. Before configuring the bits [15:0] in the Comparator Control Register CMPCR, the pattern, 0x9C3A, must first be written into the register protection bits [31:16] in the CMPCR register. Then the write protection will be disabled and the bits [15:0] can be configured by application program. For the same reason, the comparator input and output can also be locked using the corresponding lock bit in the Port n Lock Register PnLOCKR (n=A~E) in the GPIO unit.

## Register Map

The following table shows the Comparator registers and reset values.

**Table 29. CMP Register Map**

Register	Offset	Description	Reset Value
<b>CMP Base Address = 0x4005_8000</b>			
CMPCR0	0x000	Comparator Control Register 0	0x0001_0000
CVRVALR0	0x004	Comparator Voltage Reference Value Register 0	0x0000_0000
CMPIER0	0x008	Comparator Interrupt Enable Register 0	0x0000_0000
CMPTFR0	0x00C	Comparator Transition Flag Register 0	0x0000_0000
CMPCR1	0x100	Comparator Control Register 1	0x0001_0000
CVRVALR1	0x104	Comparator Voltage Reference Value Register 1	0x0000_0000
CMPIER1	0x108	Comparator Interrupt Enable Register 1	0x0000_0000
CMPTFR1	0x10C	Comparator Transition Flag Register 1	0x0000_0000

## Register Descriptions

### Comparator Control Register n – CMPCRn, n = 0 or 1

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

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

Reset value: 0x0001\_0000

	31	30	29	28	27	26	25	24
PROTECT								
Type/Reset	RW	0	RW	0	RW	0	RW	0
PROTECT								
Type/Reset	RW	0	RW	0	RW	0	RW	1
CMPOSEL								
Type/Reset	R	0	RW/R	0	RW/R	0	RW/R	0
CVRSS CVROE CVREN								
Type/Reset	RW/R	0	RW/R	0	RW/R	0	RW/R	0
CMPINSEL CMPHFM CMPSM CMPEN								
Type/Reset	RW/R	0	RW/R	0	RW/R	0	RW/R	0
SYNSEL CMPPOL								
Type/Reset	RW/R	0	RW/R	0	RW/R	0	RW/R	0

Bits	Field	Descriptions
[31:16]	PROTECT	<p>Register Protection</p> <p>For write operation:</p> <ul style="list-style-type: none"> <li>0x9C3A: Disable the CMPCRn register write protection</li> <li>Others values: Enable the CMPCRn register write protection</li> </ul> <p>For read operation:</p> <ul style="list-style-type: none"> <li>0x0000: CMPCRn register write protection is disabled</li> <li>0x0001: CMPCRn register write protection is enabled</li> </ul> <p>These bits are used to enable or disable the write protection of the filed [14:0] of the CMPCRn register. Enabling the write protection will make the filed [14:0] of the CMPCRn register become read-only to prevent any unexpected write operation. The value read from this field indicates if the write protection is enabled or not.</p>
[15]	CMPSTS	<p>Comparator Output Status</p> <ul style="list-style-type: none"> <li>0: Output is low</li> <li>1: Output is high</li> </ul> <p>This read-only bit is a copy of the comparator output state after the polarity selection and synchronization.</p>
[14]	CMPWPEN	<p>Comparator Wake-up Function Enable</p> <ul style="list-style-type: none"> <li>0: Disable comparator wake-up function</li> <li>1: Enable comparator wake-up function</li> </ul> <p>This bit is used to enable the MCU wake-up function from the Deep-Sleep mode when the comparator output after polarity control changes state from low to high.</p>

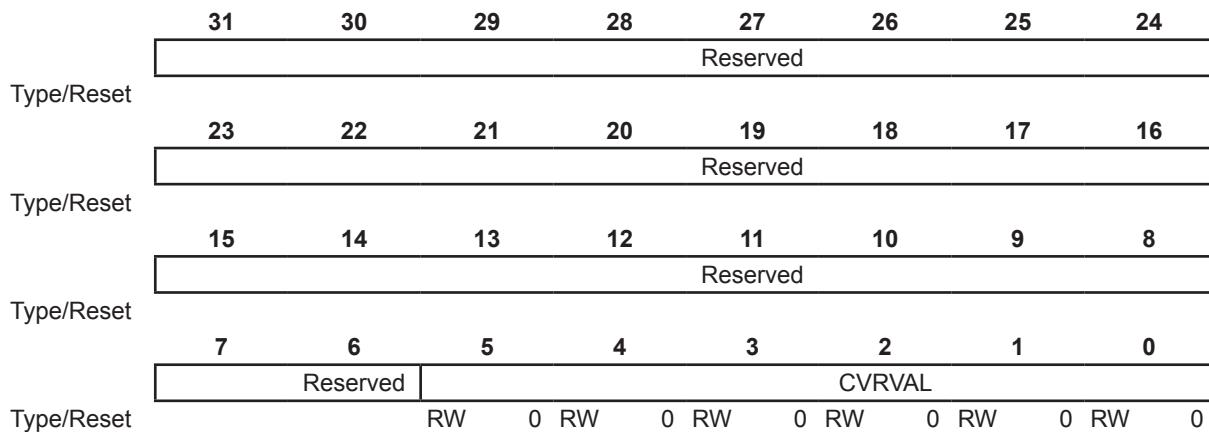
Bits	Field	Descriptions
[13:11]	CMPOSEL	<p>Comparator 0 Output Selection</p> <ul style="list-style-type: none"> <li>000: No selection</li> <li>001: GPTM0 capture channel 3</li> <li>010: MCTM0 capture channel 3</li> <li>011: MCTM0 break input 1</li> <li>100: ADC trigger input</li> <li>Other: Reserved</li> </ul> <p>Comparator 1 Output Selection</p> <ul style="list-style-type: none"> <li>000: No selection</li> <li>001: GPTM1 capture channel 3</li> <li>010: MCTM1 capture channel 3</li> <li>011: MCTM1 break input 1</li> <li>100: ADC trigger input</li> <li>Other: Reserved</li> </ul> <p>The output signal sent to the GPTM, MCTM or A/D converter is the CMPSTS signal.</p>
[10]	CVRSS	<p>Comparator Voltage Reference Source Selection</p> <ul style="list-style-type: none"> <li>0: 6-bit scaler reference voltage comes from <math>V_{DDA}</math></li> <li>1: 6-bit scaler reference voltage comes from <math>V_{REF+}</math></li> </ul>
[9]	CVROE	<p>Comparator Voltage Reference Output Enable</p> <ul style="list-style-type: none"> <li>0: Disable 6-bit scaler output to CN pin</li> <li>1: Enable 6-bit scaler output to CN pin</li> </ul>
[8]	CVREN	<p>Comparator Voltage Reference Enable</p> <ul style="list-style-type: none"> <li>0: Disable 6-bit comparator voltage reference scaler</li> <li>1: Enable 6-bit comparator voltage reference scaler</li> </ul>
[7]	SYNSEL	<p>Synchronization Selection</p> <ul style="list-style-type: none"> <li>0: Asynchronous signal of Comparator output is selected</li> <li>1: Synchronous signal of comparator output is selected</li> </ul> <p>Note that the synchronous comparator output should be selected before being passed to the AFIO unit.</p>
[6]	CMPPOL	<p>Comparator Output Polarity Selection</p> <ul style="list-style-type: none"> <li>0: Comparator output is not inverted</li> <li>1: Comparator output is inverted</li> </ul>
[5:4]	CMPINSEL	<p>Comparator Inverted Input Selection</p> <ul style="list-style-type: none"> <li>00: The external CN pin is selected</li> <li>01: The internal 6-bit reference voltage scaler output is selected</li> <li>1x: Reserved</li> </ul> <p>These bits are used to select the comparator inverted input source.</p>
[3:2]	CMPHM	<p>Comparator Hysteresis Mode Selection</p> <ul style="list-style-type: none"> <li>00: No hysteresis</li> <li>01: Low hysteresis mode</li> <li>10: Middle hysteresis mode</li> <li>11: High hysteresis mode</li> </ul>
[1]	CMPSM	<p>Comparator Response Speed Mode Selection</p> <ul style="list-style-type: none"> <li>0: Low speed mode</li> <li>1: High speed mode</li> </ul>
[0]	CMPEN	<p>Comparator Enable</p> <ul style="list-style-type: none"> <li>0: Disable Comparator – Entering the power down mode</li> <li>1: Enable Comparator</li> </ul>

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

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

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

Reset value: 0x0000\_0000



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

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

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

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

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					CMPRIEN	CMPFIEN			
						RW	0	RW	0	

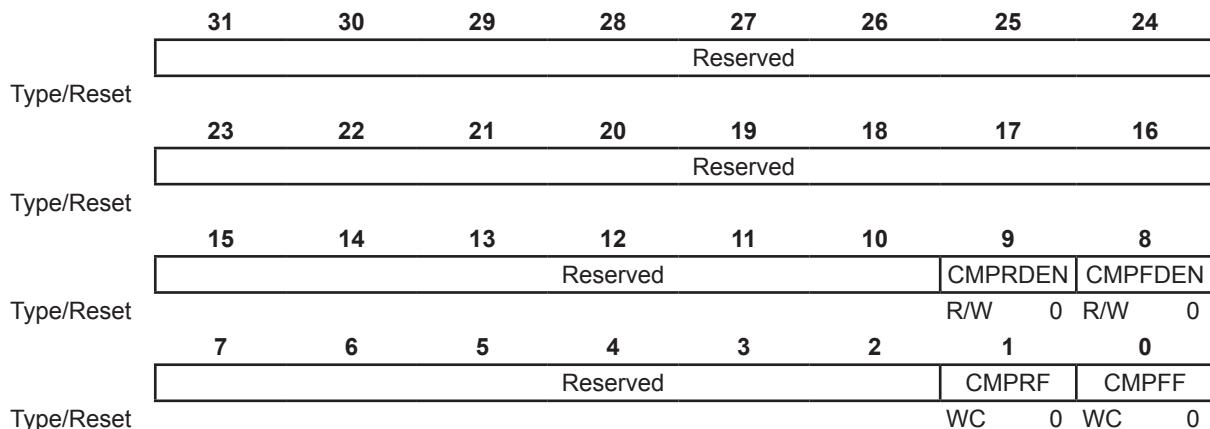
Bits	Field	Descriptions
[1]	CMPRIEN	Comparator Output Rising Edge Interrupt Enable 0: Disable comparator output rising edge interrupt 1: Enable comparator output rising edge interrupt
[0]	CMPFIEN	Comparator Output Falling Edge Interrupt Enable 0: Disable comparator output falling edge interrupt 1: Enable comparator output falling edge interrupt

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

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

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

Reset value: 0x0000\_0000



Bits	Field	Descriptions
[9]	CMPRDEN	Comparator Output Rising Edge Detection Enable 0: Disable comparator output rising edge detection 1: Enable comparator output rising edge detection Note that the signal to be detected is a copy of the comparator output state after the polarity selection and synchronization by the HCLK clock.
[8]	CMPFDEN	Comparator Output Falling Edge Detection Enable 0: Disable comparator output falling edge detection 1: Enable comparator output falling edge detection Note that the signal to be detected is a copy of the comparator output state after the polarity selection and synchronization by the HCLK clock.
[1]	CMPRF	Comparator Output Rising Edge Flag 0: No comparator output rising edge occurs 1: Comparator output rising edge occurs This flag is available when the comparator output rising edge detection function is enabled. This bit is set to 1 by hardware and cleared to 0 by writing a "1" into it.
[0]	CMPFF	Comparator Output Falling Edge Flag 0: No Comparator output falling edge occurs 1: Comparator output falling edge occurs This flag is available when the comparator output falling edge detection function is enabled. This bit is set to 1 by hardware and cleared to 0 by writing a "1" into it.

# 14

## General-Purpose Timers (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.

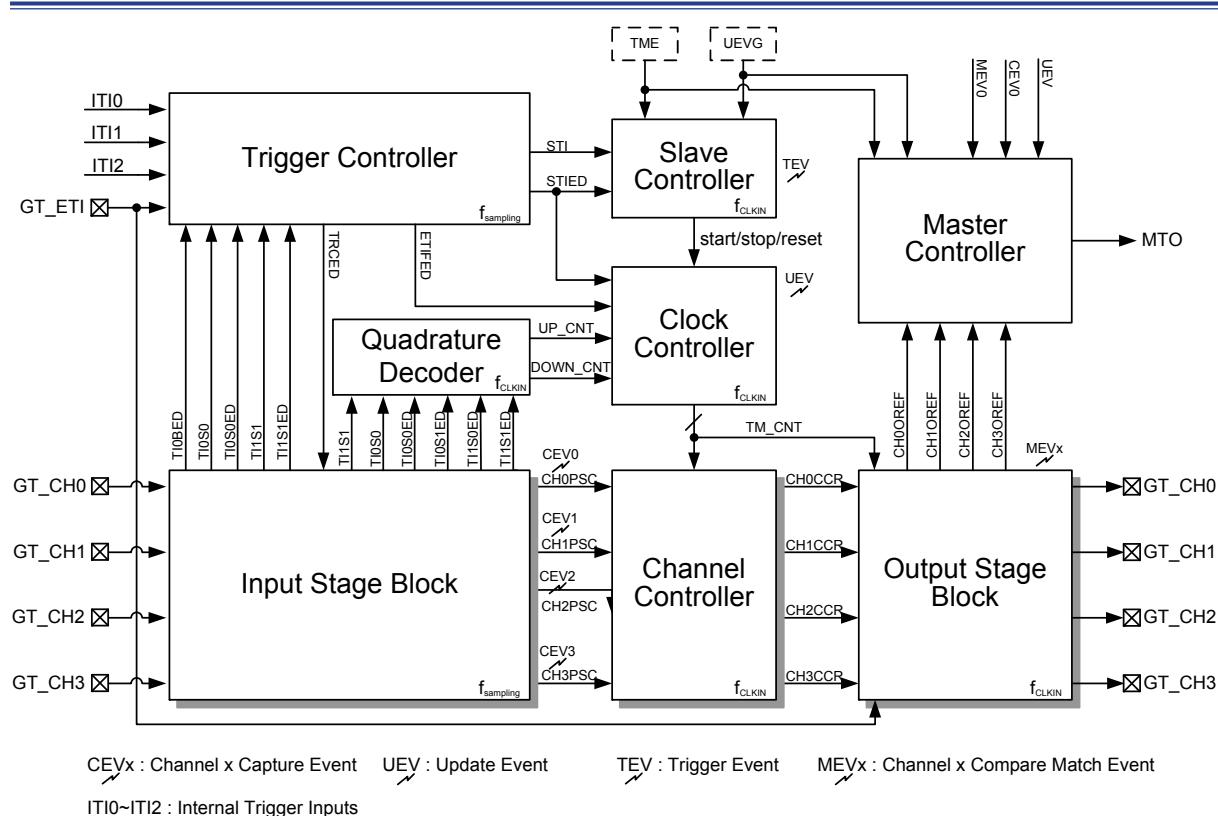


Figure 36. GPTM Block Diagram

## Features

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

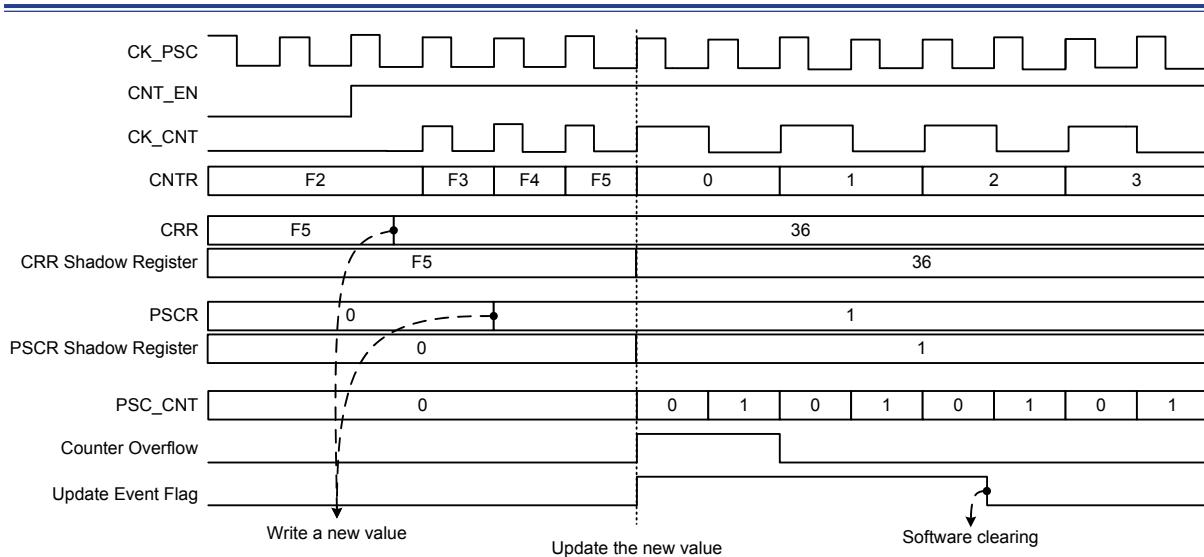
## Functional Descriptions

### Counter Mode

#### Up-Counting

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

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

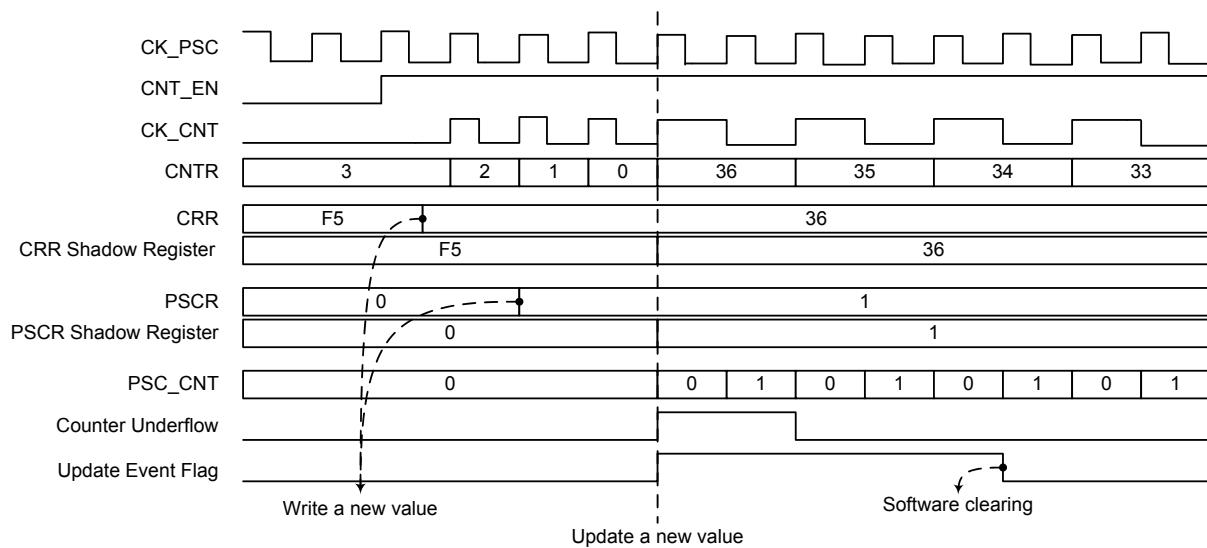


**Figure 37. Up-counting Example**

### 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 38. 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.

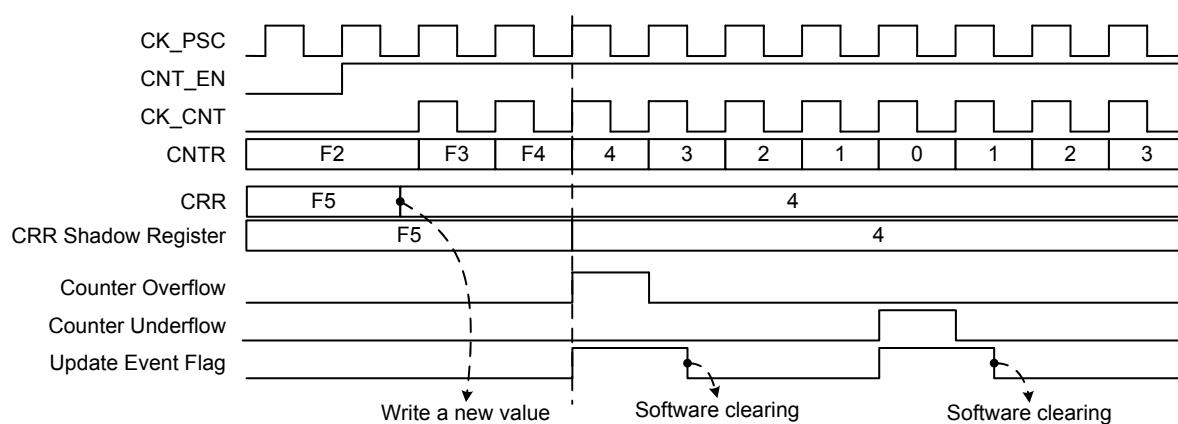


Figure 39. Center-aligned Counting Example

## Clock Controller

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

### Internal APB clock $f_{CLKIN}$

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

### Quadrature Decoder

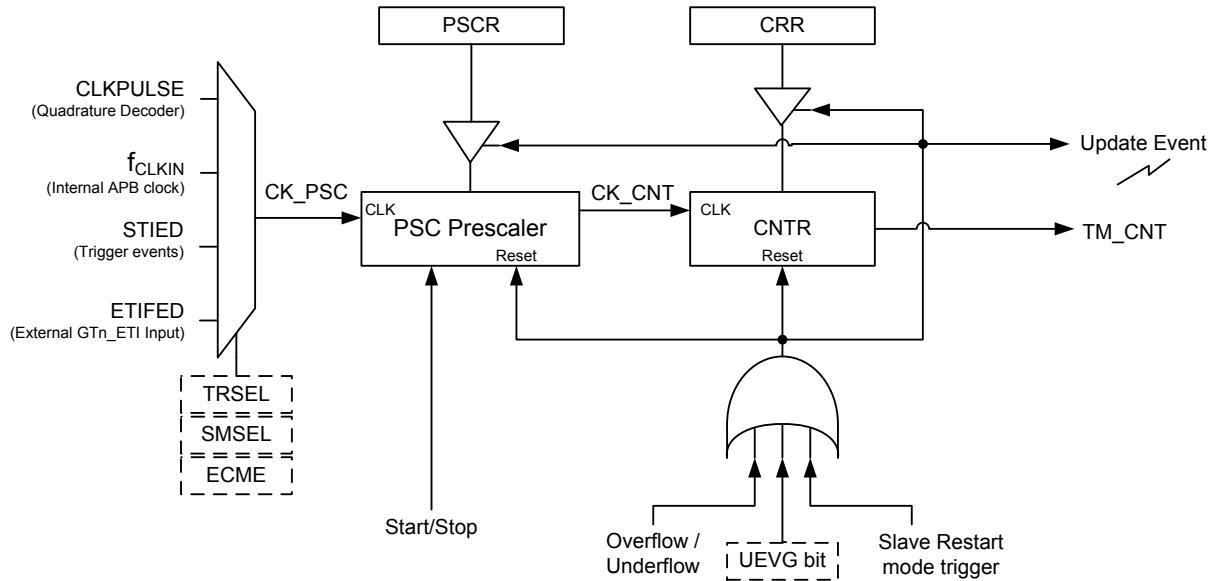
To select Quadrature Decoder mode the SMSEL field should be set to 0x1, 0x2 or 0x3 in the MDCFR register. The Quadrature Decoder function uses two input states of the GT\_CH0 and GT\_CH1 pins to generate the clock pulse to drive the counter prescaler. The counting direction bit DIR is modified by hardware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

### STIED

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

### ETIFED

The counter prescaler can be driven to count during each rising edge on the external pin GT\_ETI. This mode can be selected by setting the ECME bit in the TRCFR register to 1. The other way to select the ETIF signal as the clock source is to set the SMSEL field to 0x7 and the TRSEL field to 0x3 respectively. Note that the ETIF signal is derived from the GT\_ETI pin sampled by a digital filter. When the clock source is selected to come from the ETIF signal, the Trigger Controller including the edge detection circuitry will generate a clock pulse during each ETIF signal rising edge to clock the counter prescaler.

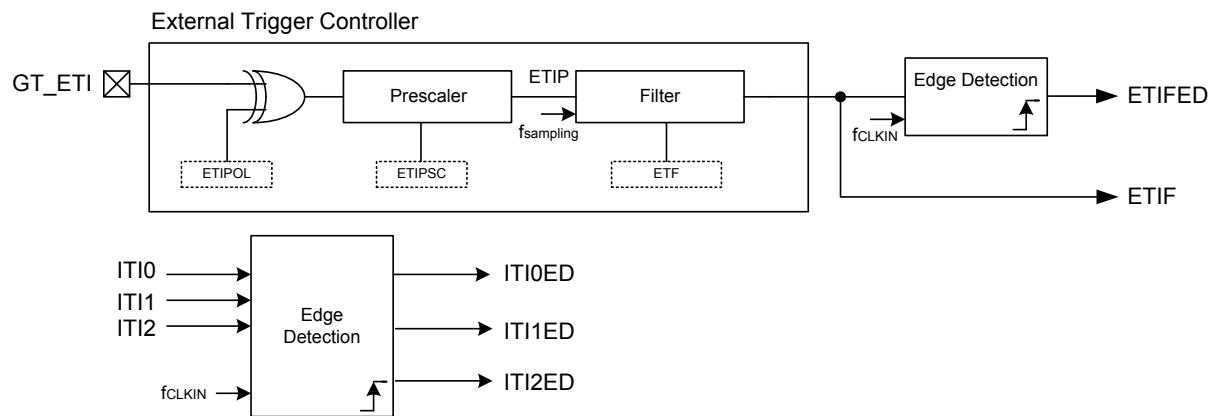


**Figure 40. GPTM Clock Selection Source**

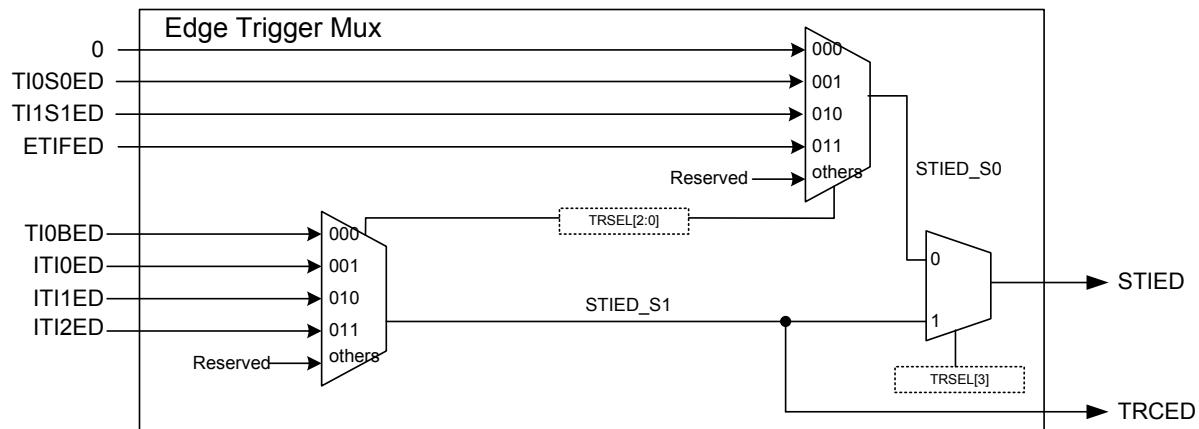
### Trigger Controller

The trigger controller is used to select the trigger source and setup the trigger level or edge trigger condition. The active polarity of the external trigger input signal GT\_ETI can be configured by the External Trigger Polarity control bit ETIPOL in the GPTM Trigger Configuration Register TRCFR. The frequency of the external trigger input can be divided by configuring the related bits, named as External Trigger Prescaler control bits ETIPSC, in the TRCFR register. The trigger signal can also be filtered by configuring the External Trigger Filter ETF selection bits in the TRCFR register if a filtered signal is necessary for specific applications. 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.

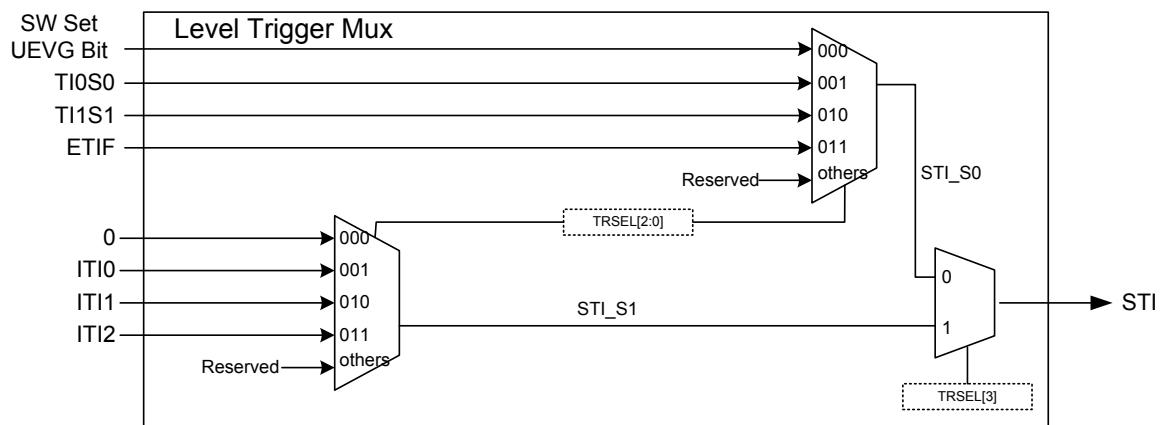
Trigger Controller Block = Edge Trigger Mux + Level Trigger Mux



Edge Trigger = External (ETI)+ Internal (ITIx) + Channel input (CHx) + XOR function



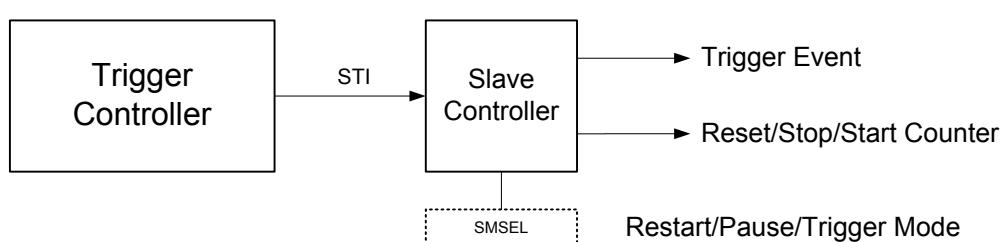
Level Trigger Source = External (ETI)+ Internal (ITIx) + Channel input (CHx) + Software UEV1G bit



**Figure 41. Trigger Control Block**

## Slave Controller

The GPTM can be synchronized with an external trigger in several modes including the Restart mode, the Pause mode and the Trigger mode which 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.

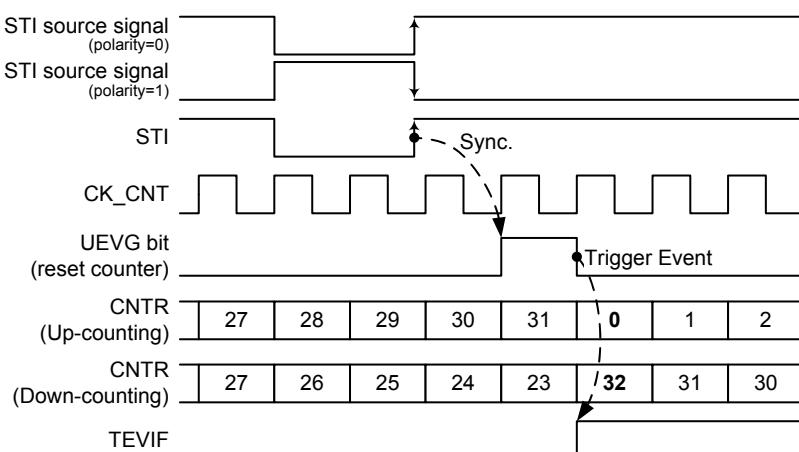


**Figure 42. Slave Controller Diagram**

### 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 43. 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.

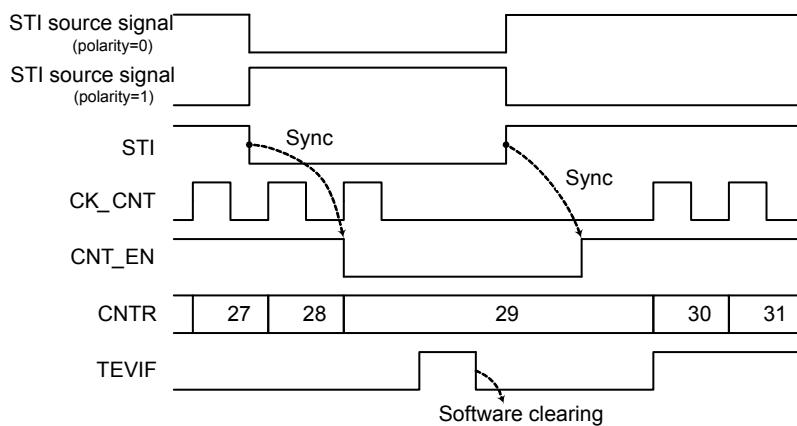


Figure 44. GPTM in Pause Mode

### 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.

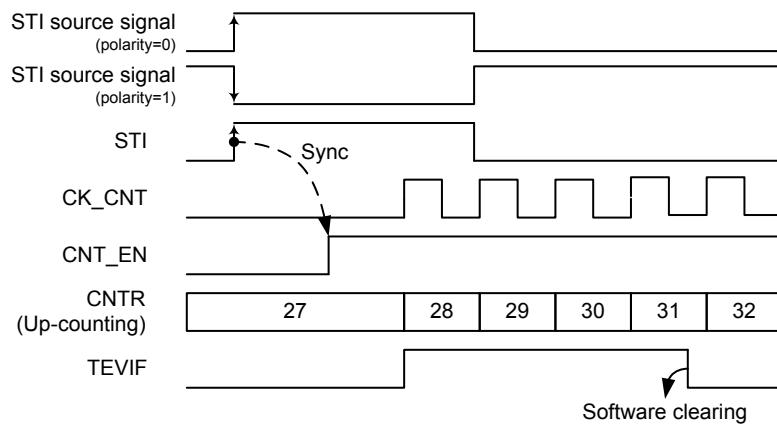


Figure 45. GPTM in Trigger Mode

## 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.

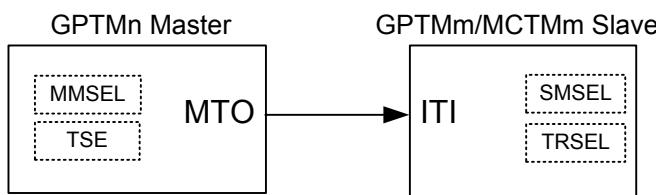


Figure 46. Master GPTMn and Slave GPTMm/MCTMm Connection

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.

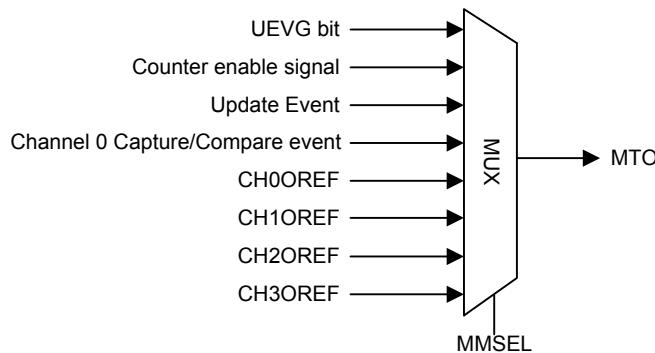


Figure 47. MTO Selection

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.

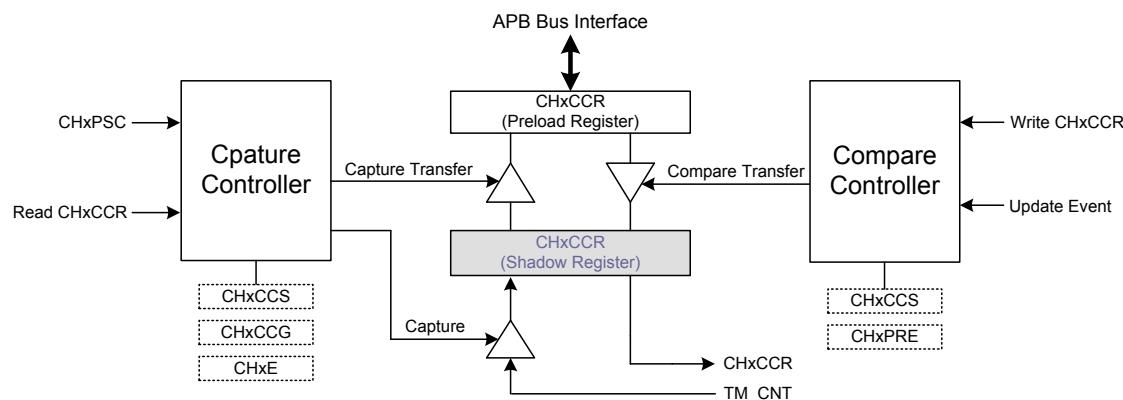
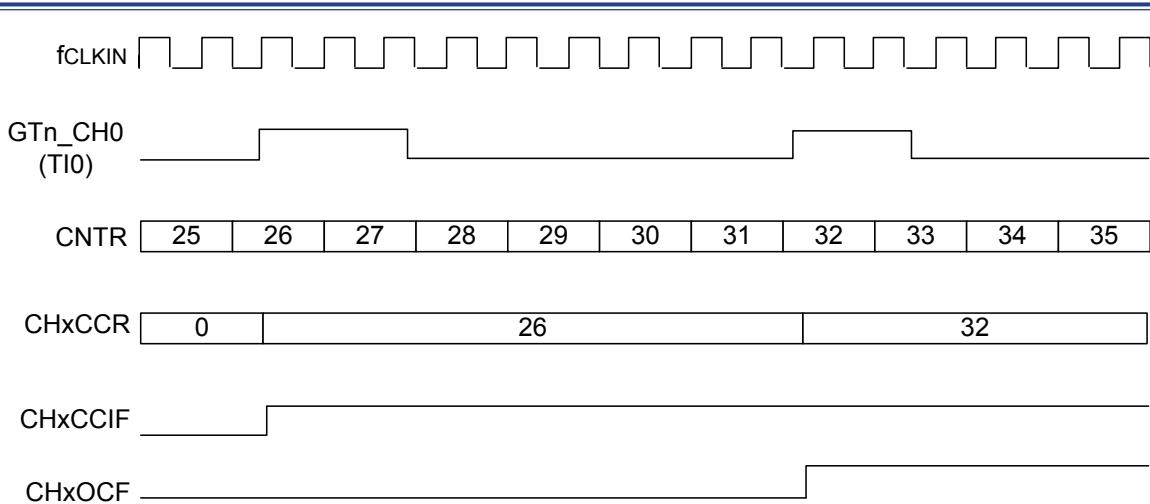


Figure 48. Capture/Compare Block Diagram

### 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.



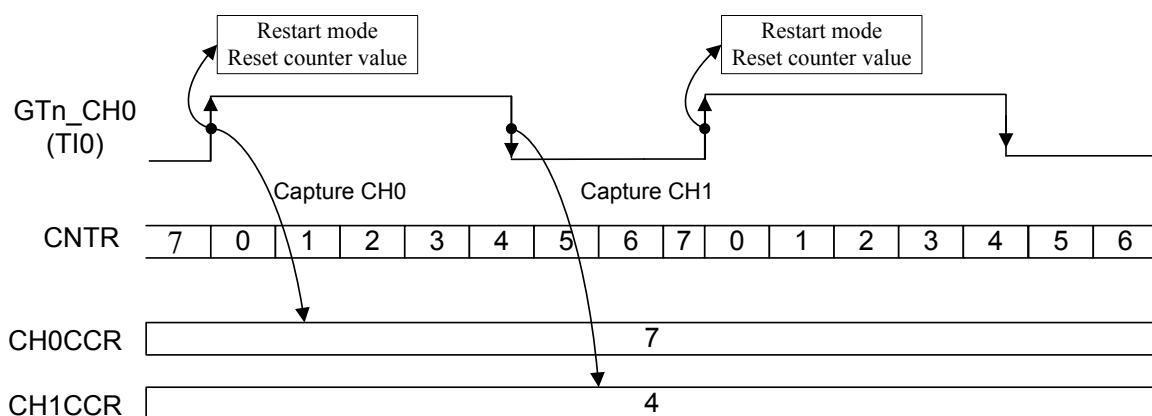
**Figure 49. Input Capture Mode**

### 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 50. 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 TIxF. 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).

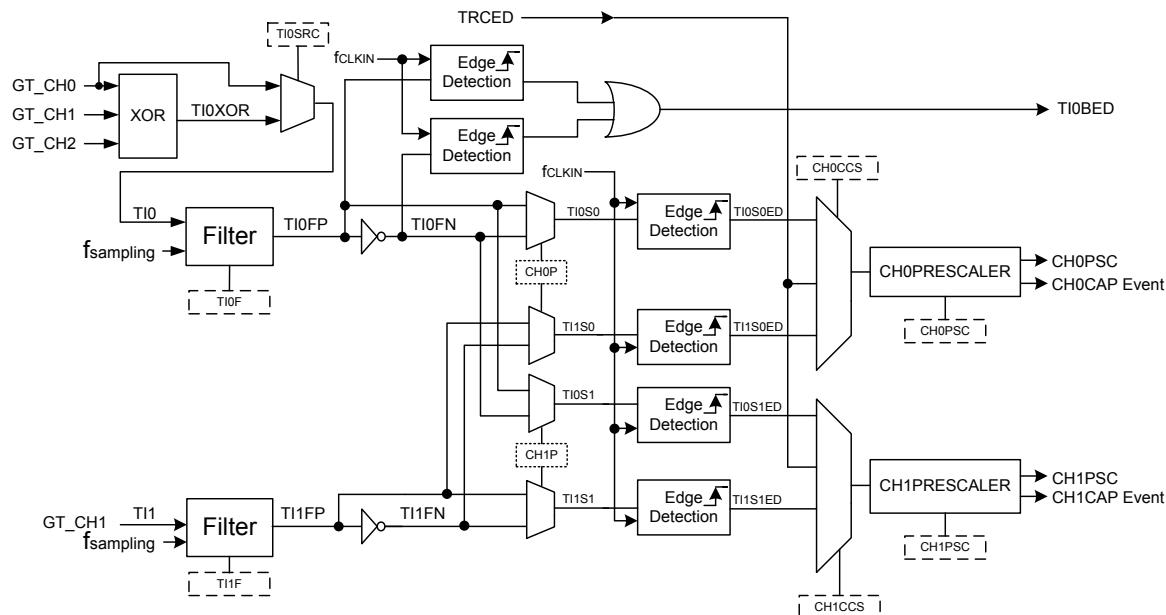
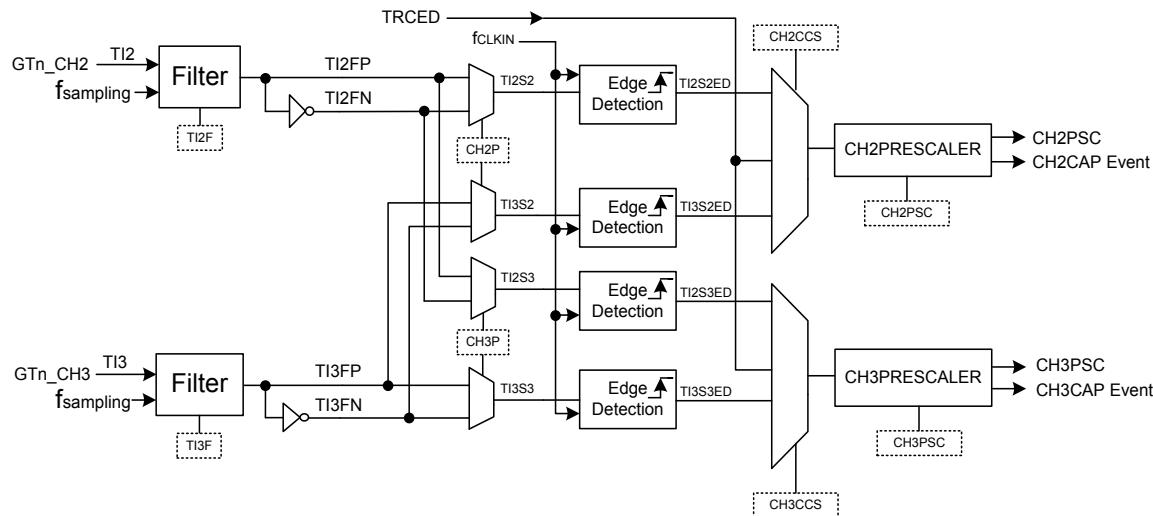


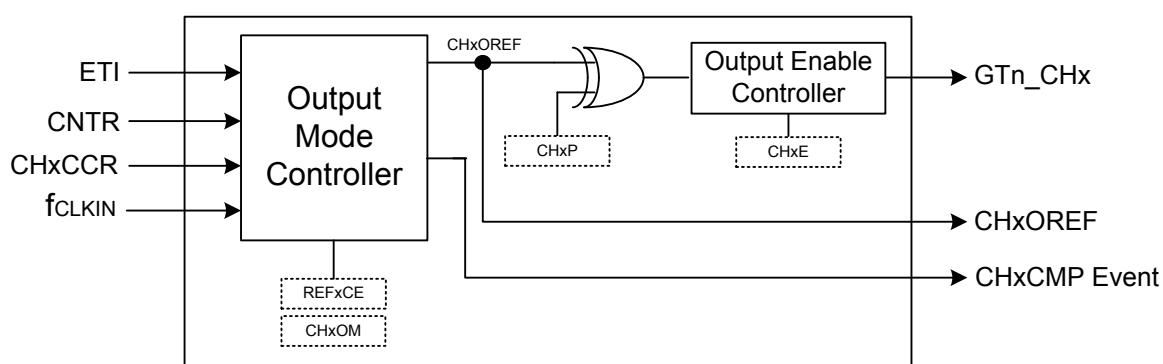
Figure 51. Channel 0 and Channel 1 Input Stage



**Figure 52. Channel 2 and Channel 3 Input Stage**

## Output Stage

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



**Figure 53. 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 setting the CHxOM bits. The CHxOREF signal has several types of output function. These include, keeping the original level by setting the CHxOM field to 0x00, set to 0 by setting the CHxOM field to 0x01, set to 1 by setting the CHxOM field to 0x02 or signal toggle by setting the CHxOM field to 0x03 when the counter value matches the content of the CHxCCR register.

The PWM mode 1 and PWM mode 2 outputs are also another kind of CHxOREF output which is setup by setting the CHxOM field to 0x06/0x07. In these modes, the CHxOREF signal level is changed according to the counting direction and the relationship between the counter value and the CHxCCR content. With regard to a more detailed description refer to the relative bit definition.

Another special function of the CHxOREF signal is a forced output which can be achieved by setting the CHxOM field to 0x04/0x05. Here the output can be forced to an inactive/active level irrespective of the comparison condition between the counter and the CHxCCR values.

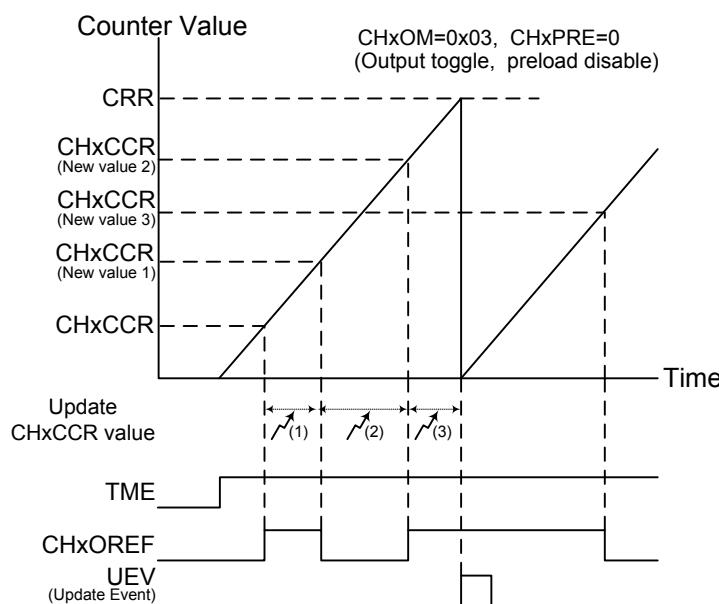


Figure 54. Toggle Mode Channel Output Reference Signal (CHxPRE = 0)

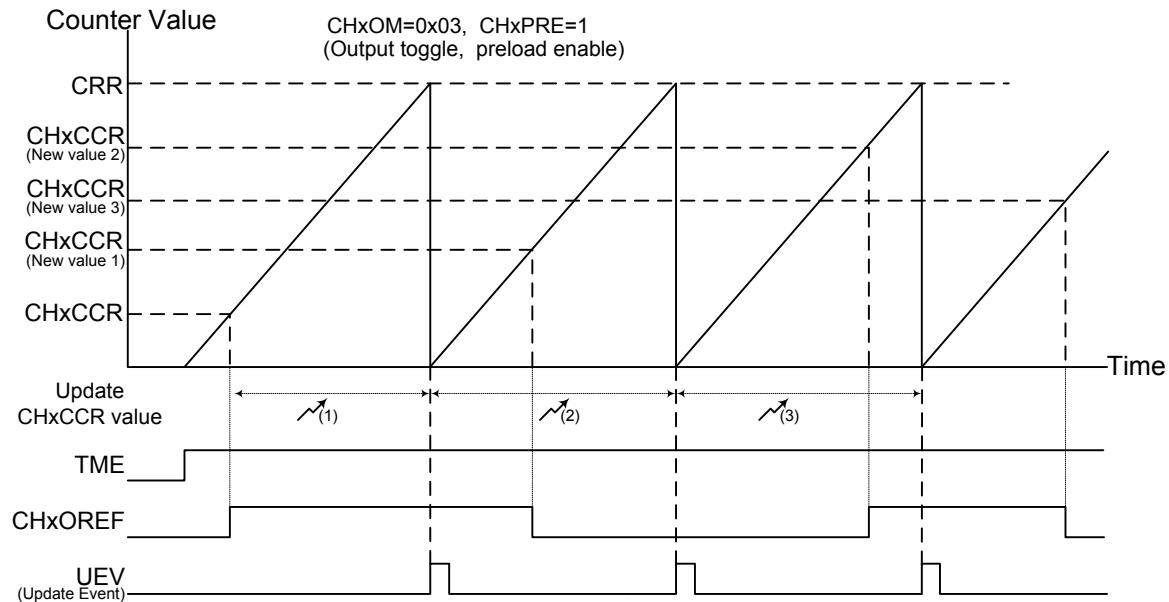


Figure 55. Toggle Mode Channel Output Reference Signal (CHxPRE = 1)

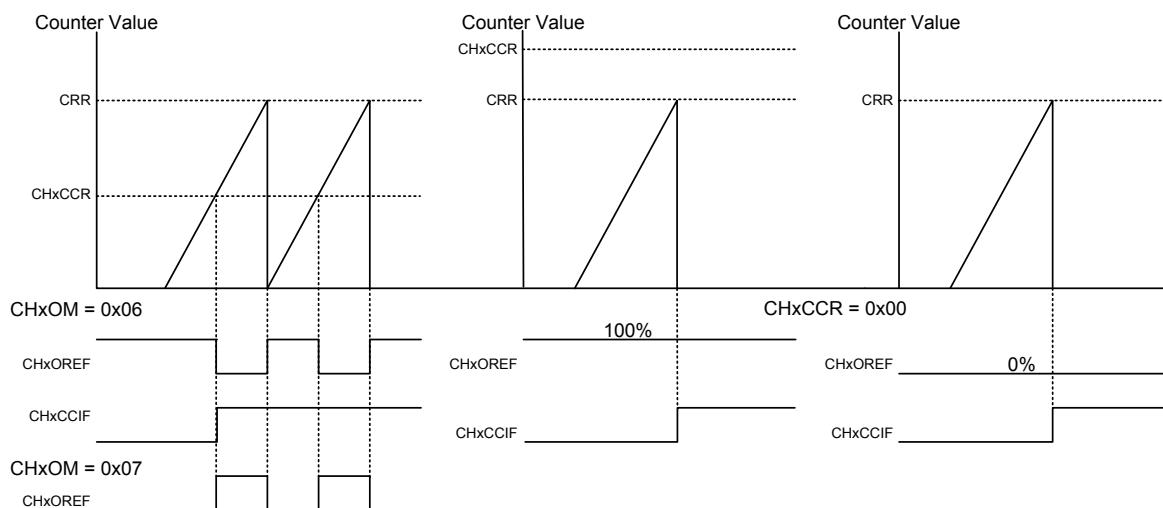


Figure 56. PWM Mode Channel Output Reference Signal and Counter in Up-counting Mode

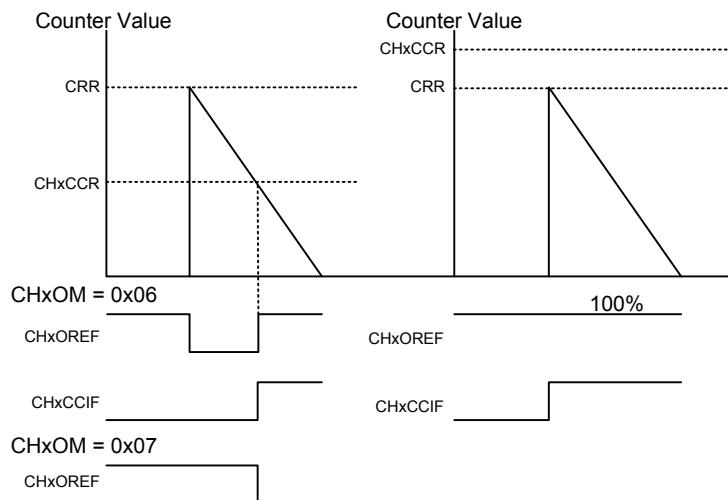


Figure 57. PWM Mode Channel Output Reference Signal and Counter in Down-counting Mode

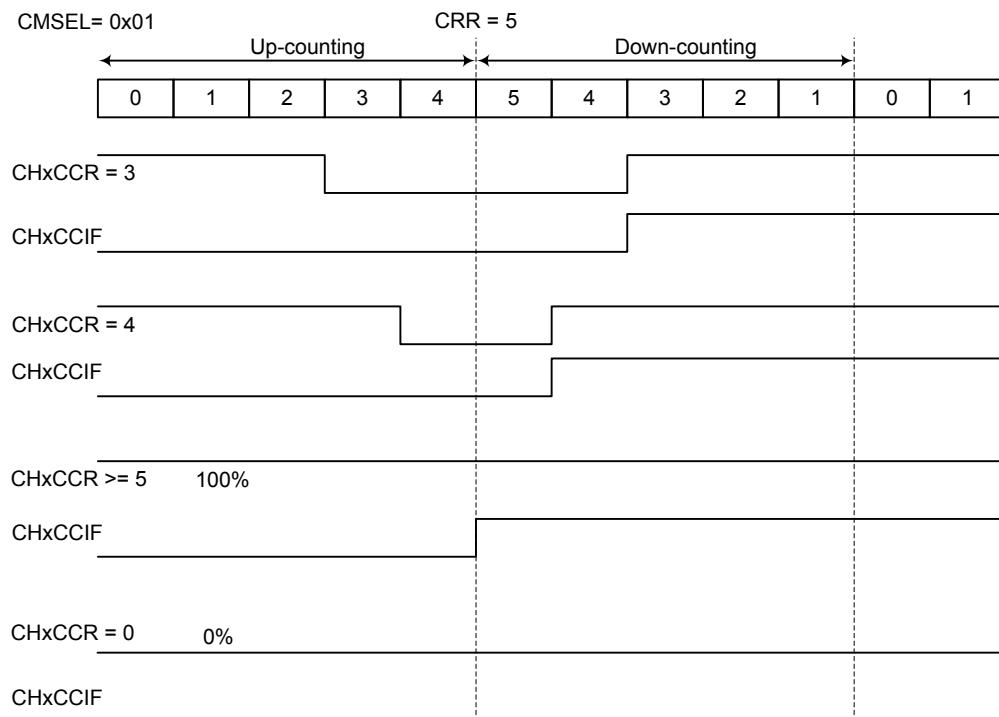


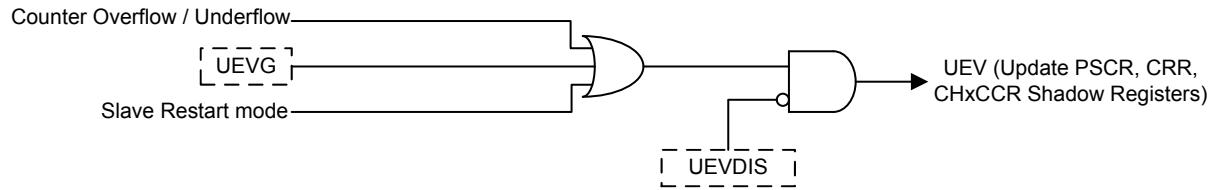
Figure 58. PWM Mode Channel Output Reference Signal and Counter in Centre-align Mode

## Update Management

The Update event is used to update the CRR, the PSCR 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

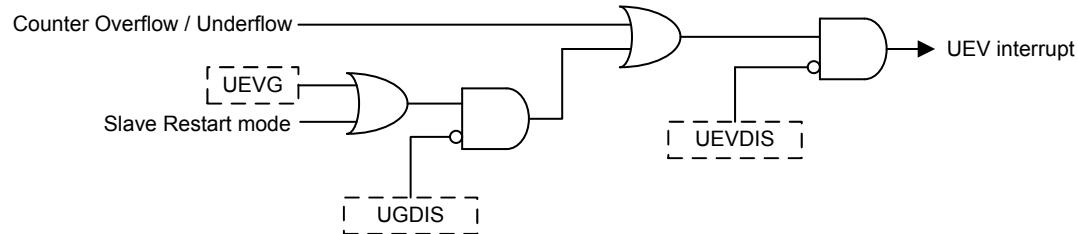
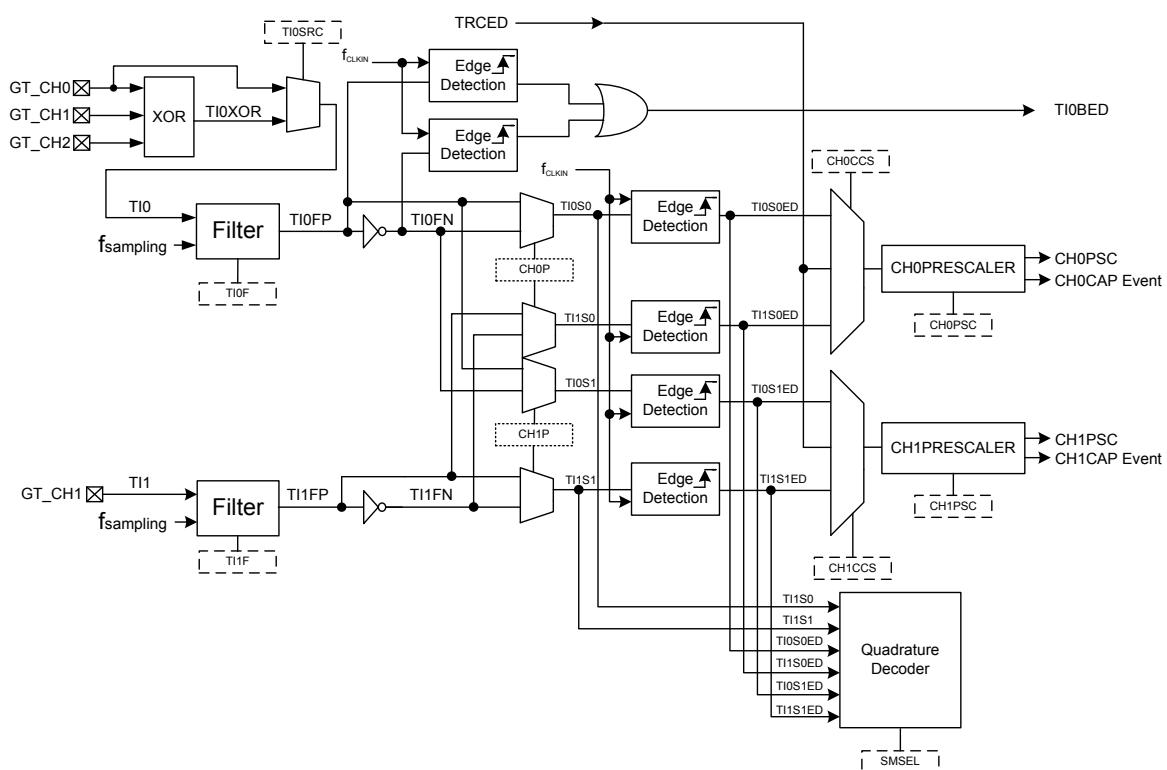


Figure 59. Update Event Setting Diagram

## 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.

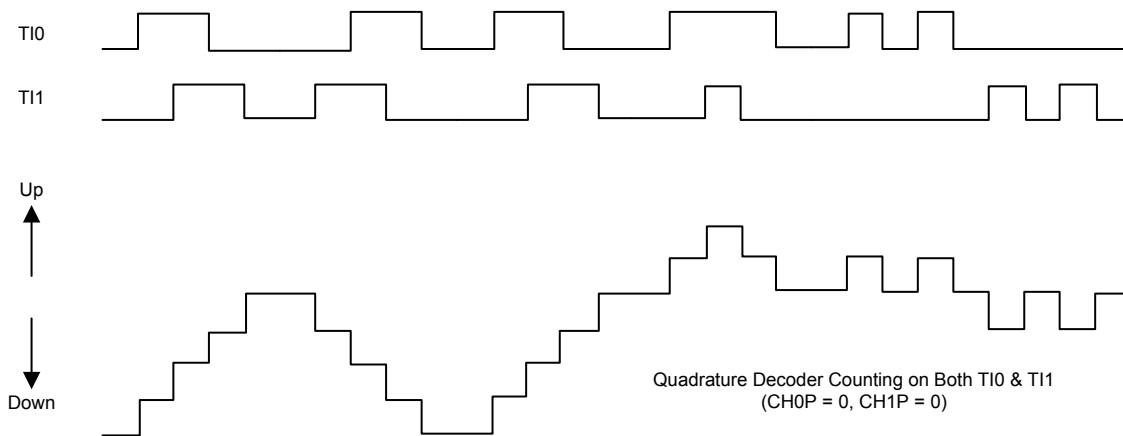


**Figure 60. Input Stage and Quadrature Decoder Block Diagram**

**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 61. Both TI0 and TI1 Quadrature Decoder Counting**

## Digital Filter

The digital filters are embedded in the input stage and clock controller block for the GT\_CH0 ~ GT\_CH3 and GT\_ETI 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.

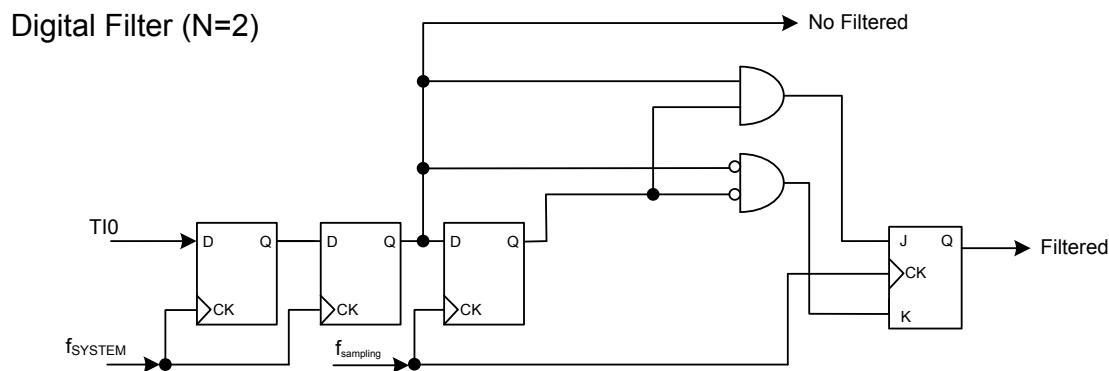


Figure 62. GT\_ETI Pin Digital Filter Diagram with N = 2

## Clearing the CHxOREF when ETIF is high

The CHxOREF signal can be forced to 0 when the ETIF signal is derived from the external GT\_ETI pin and when it is set to a high level by setting the REFxCE bit to 1 in the CHxOCFR register. The CHxOREF signal will not return to its active level until the next update event occurs.

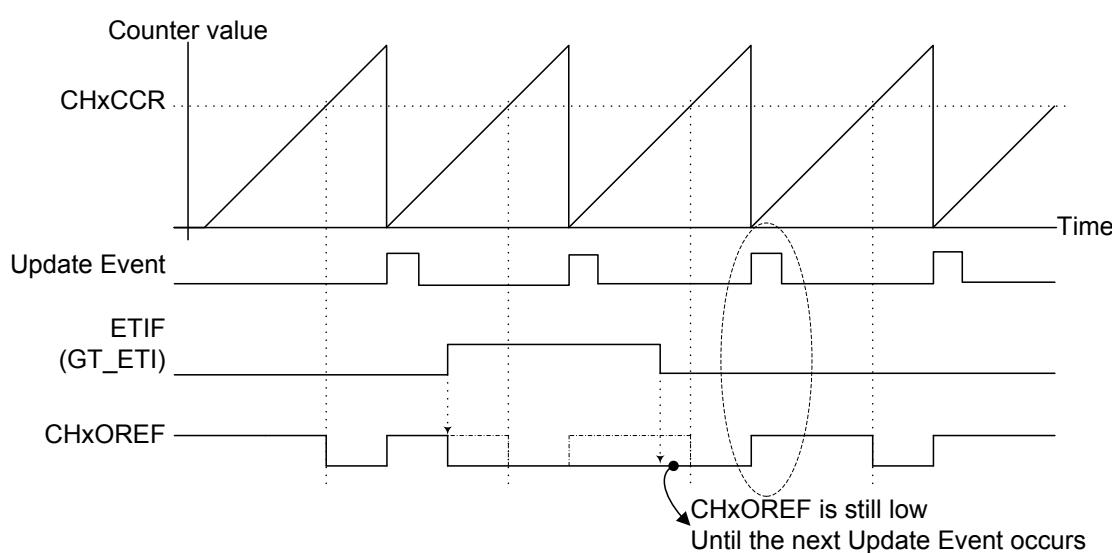
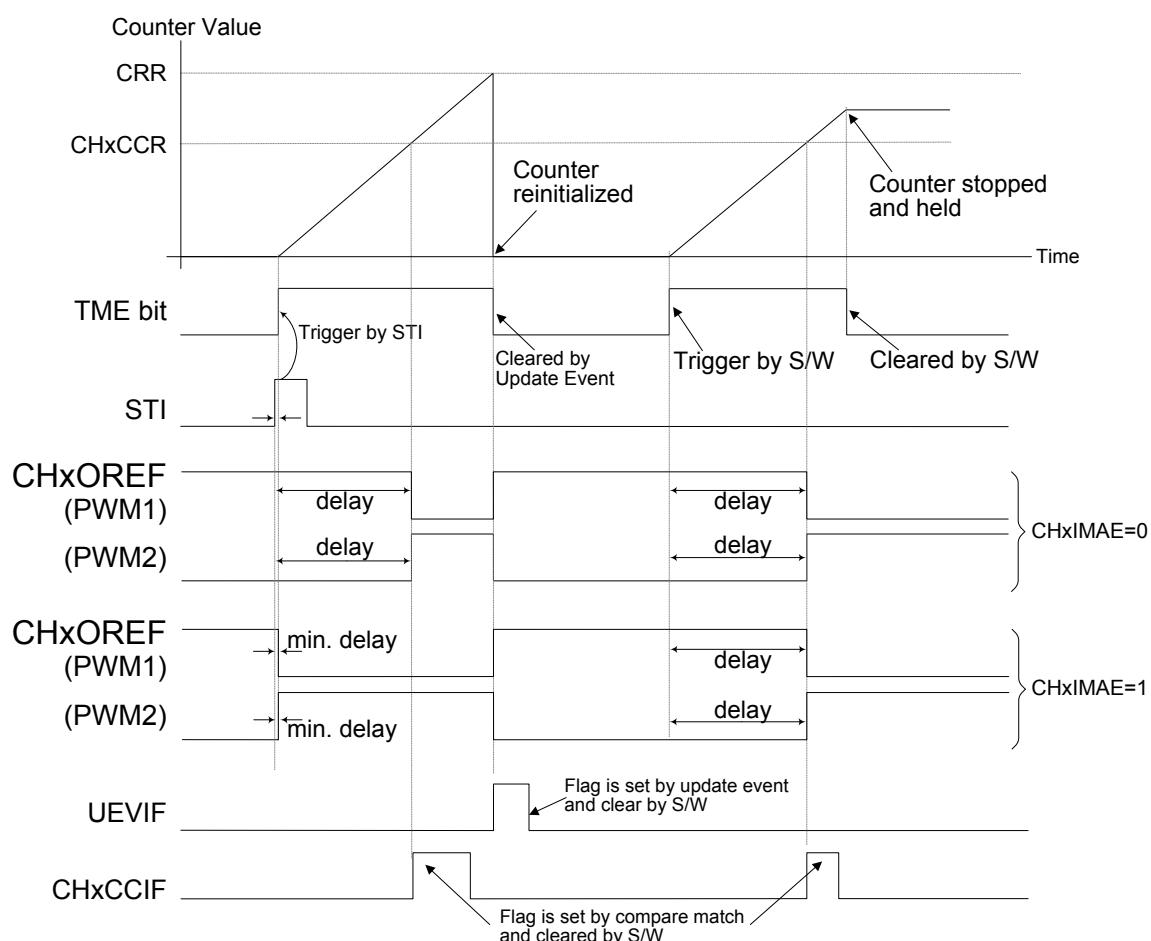


Figure 63. Clearing CHxOREF by ETIF

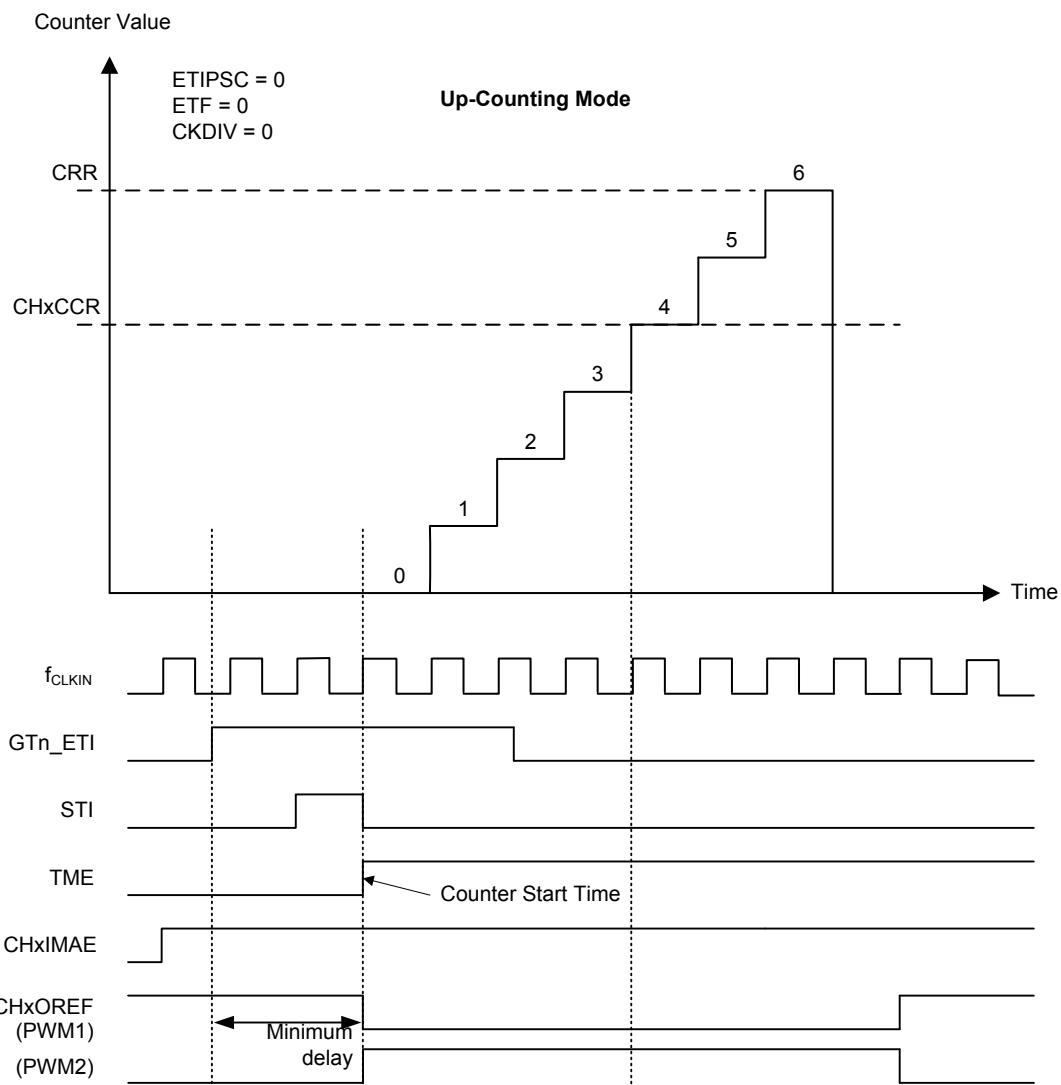
## Single Pulse Mode

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 64. 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.

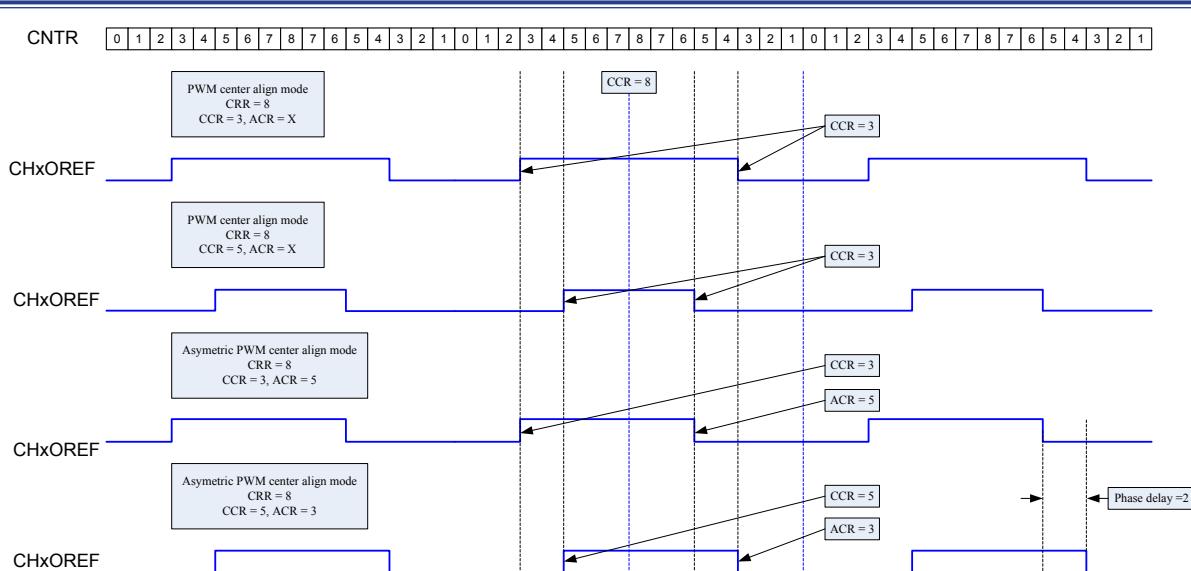


**Figure 65. Immediate Active Mode Minimum Delay**

## Asymmetric PWM Mode

Asymmetric PWM mode allows two center-aligned PWM signals to be generated with a programmable phase shift. While the PWM frequency is determined by the value of the CRR register, the duty cycle and the phase-shift are determined by the CHxCCR and CHxACR register. When the counter is counting up, the PWM using the value in CHxCCR as up-count compare value. When the counter is into counting down stage, the PWM using the value in CHxACR as down-count compare value. The Figure 66 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 66. Asymmetric PWM Mode versus Center Align Counting Mode**

## Timer Interconnection

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

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

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

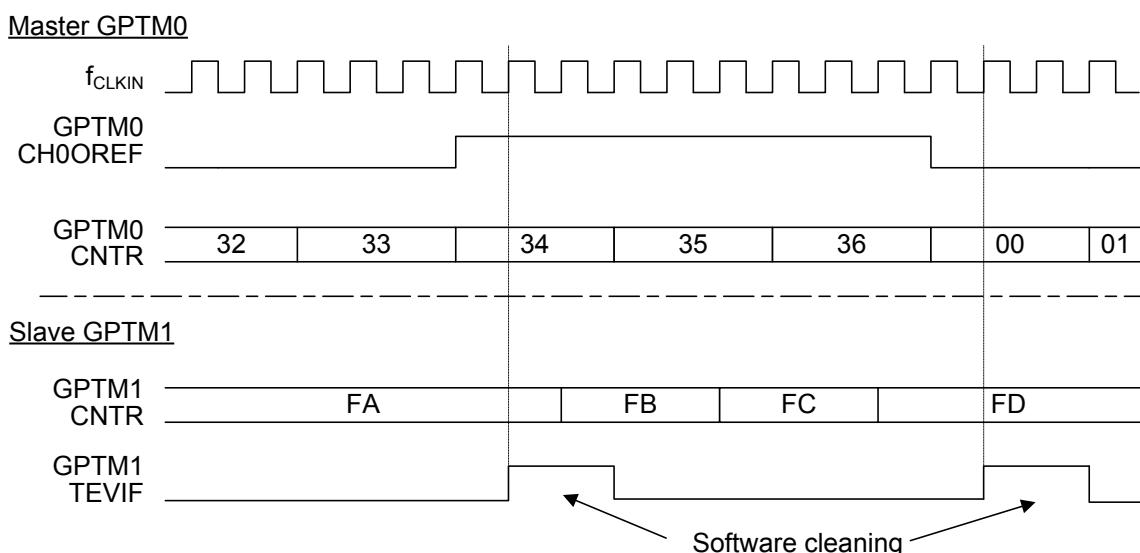


Figure 67. Pausing GPTM1 using the GPTM0 CH0OREF Signal

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

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

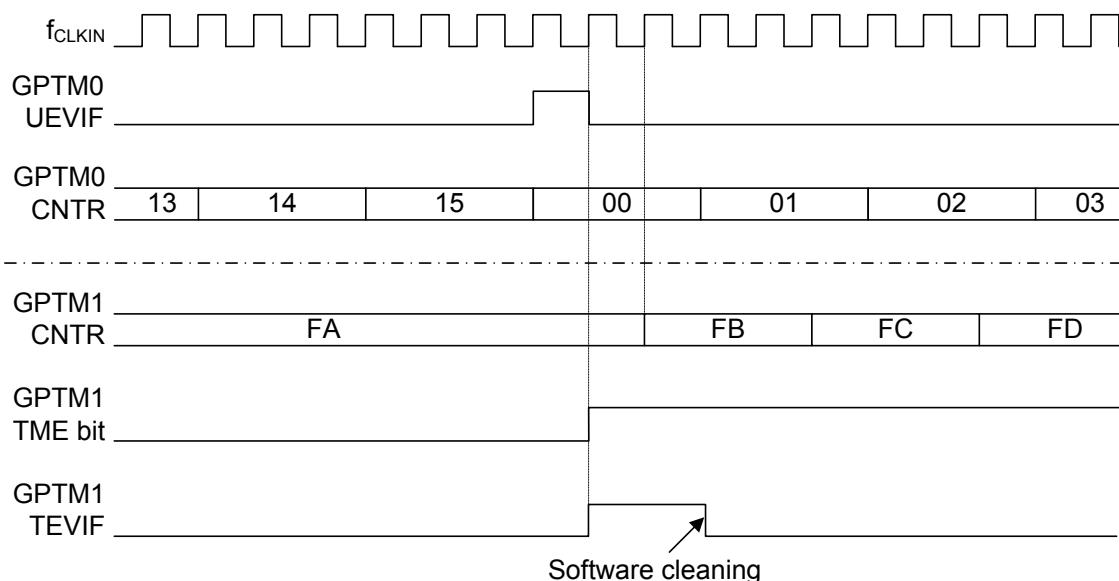
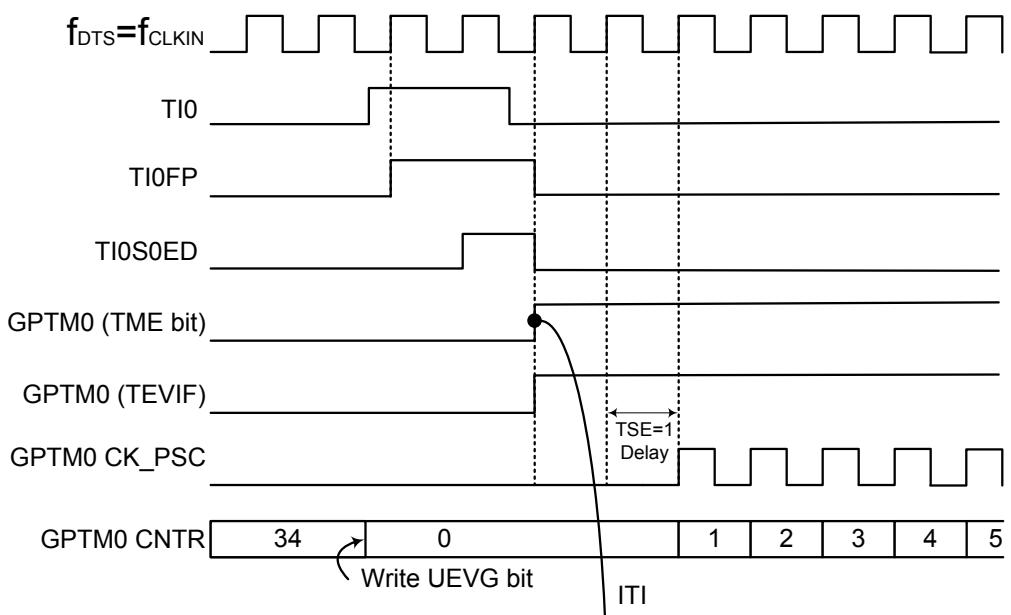


Figure 68. Triggering GPTM1 with GPTM0 Update Event

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

- Configure GPTM0 to operate in the master mode to send its enable signal as a trigger output (MMSEL = 0x01).
- Configure GPTM0 slave mode to receive its input trigger source from GT\_CH0 pin (TRSEL = 0x01).
- Configure GPTM0 to be in the slave trigger mode (SMSEL = 0x06).
- Enable the GPTM0 master timer synchronization function by setting the TSE bit in the MDCFR register to 1 to synchronize the slave timer.
- Configure GPTM1 to receive its input trigger source from the GPTM0 trigger output (TRSEL = 0x09).
- Configure GPTM1 to be in the slave trigger mode (SMSEL = 0x06).

### Master GPTM0



### Slave GPTM1

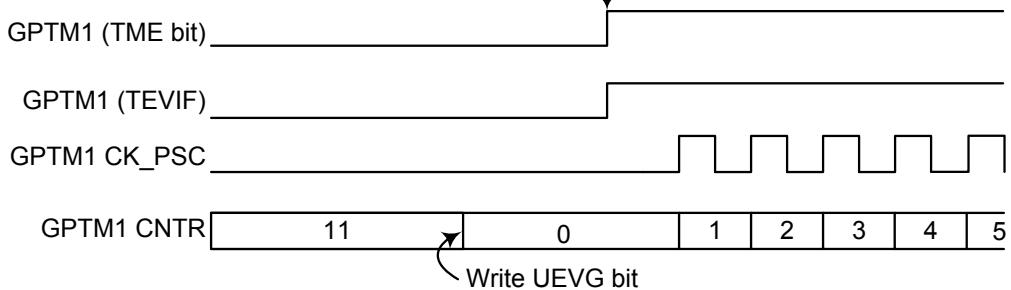


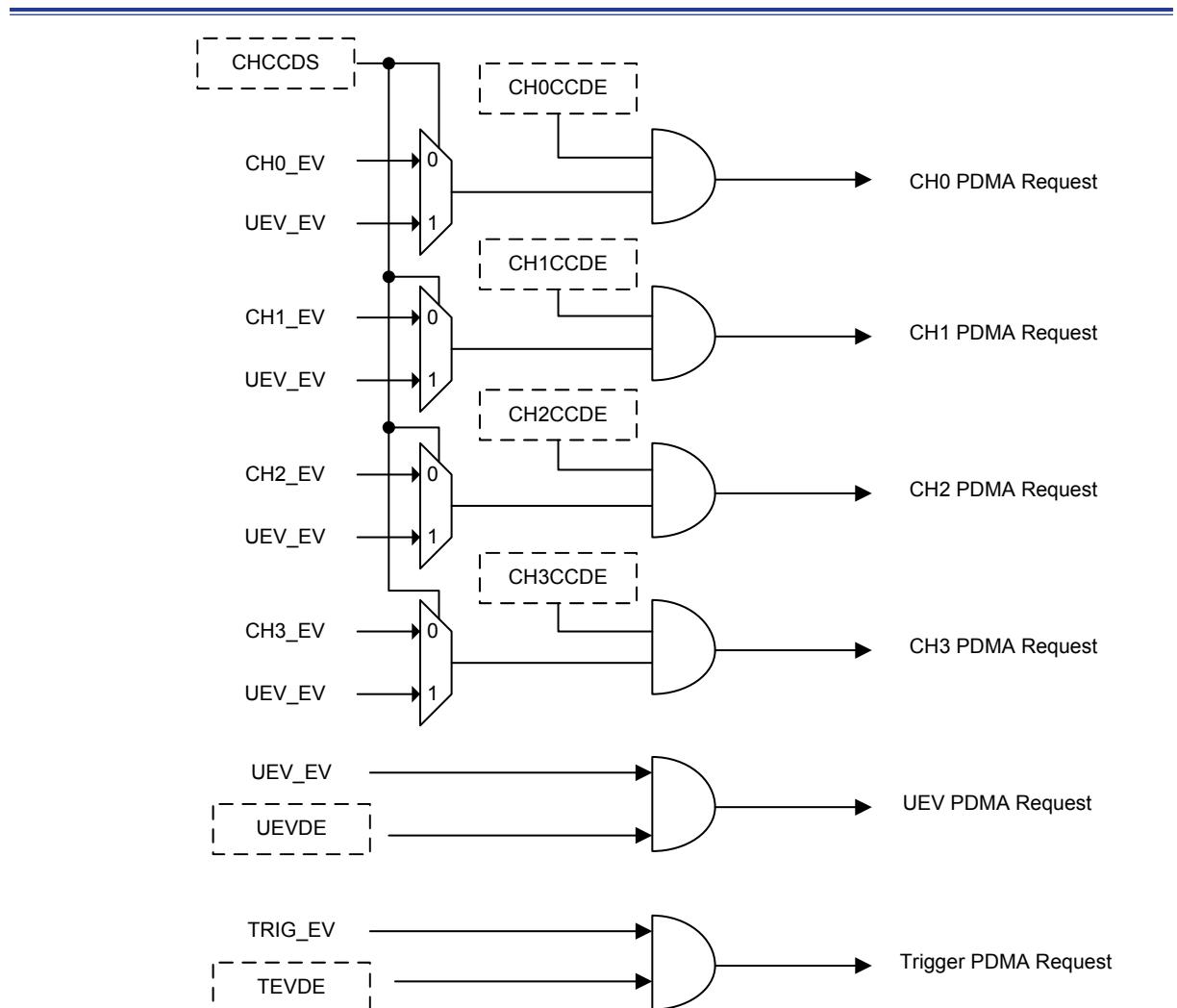
Figure 69. Trigger GPTM0 and GPTM1 with the GPTM0 CH0 Input

## Trigger ADC Start

To interconnect with the Analog-to-Digital Converter, the GPTM can output the MTO signal or the channel output GT\_CHx (x = 0 ~ 3) signal to be used as the Analog-to-Digital Converter input trigger signal.

## PDMA Request

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



**Figure 70. GPTM PDMA Mapping Diagram**

## Register Map

The following table shows the GPTM registers and reset values.

**Table 31. GPTM Register Map**

Register	Offset	Description	Reset Value
<b>GPTM0 Base Address = 0x4006_E000</b>			
<b>GPTM1 Base Address = 0x4006_F000</b>			
CNTCFR	0x000	Timer Counter Configuration Register	0x0000_0000
MDCFR	0x004	Timer Mode Configuration Register	0x0000_0000
TRCFR	0x008	Timer Trigger Configuration Register	0x0000_0000
CTR	0x010	Timer Control Register	0x0000_0000
CH0ICFR	0x020	Channel 0 Input Configuration Register	0x0000_0000
CH1ICFR	0x024	Channel 1 Input Configuration Register	0x0000_0000
CH2ICFR	0x028	Channel 2 Input Configuration Register	0x0000_0000
CH3ICFR	0x02C	Channel 3 Input Configuration Register	0x0000_0000
CH0OCFR	0x040	Channel 0 Output Configuration Register	0x0000_0000
CH1OCFR	0x044	Channel 1 Output Configuration Register	0x0000_0000
CH2OCFR	0x048	Channel 2 Output Configuration Register	0x0000_0000
CH3OCFR	0x04C	Channel 3 Output Configuration Register	0x0000_0000
CHCTR	0x050	Channel Control Register	0x0000_0000
CHPOLR	0x054	Channel Polarity Configuration Register	0x0000_0000
DICTR	0x074	Timer PDMA / Interrupt Control Register	0x0000_0000
EVGR	0x078	Timer Event Generator Register	0x0000_0000
INTSR	0x07C	Timer Interrupt Status Register	0x0000_0000
CNTR	0x080	Timer Counter Register	0x0000_0000
PSCR	0x084	Timer Prescaler Register	0x0000_0000
CRR	0x088	Timer Counter Reload Register	0x0000_FFFF
CH0CCR	0x090	Channel 0 Capture/Compare Register	0x0000_0000
CH1CCR	0x094	Channel 1 Capture/Compare Register	0x0000_0000
CH2CCR	0x098	Channel 2 Capture/Compare Register	0x0000_0000
CH3CCR	0x09C	Channel 3 Capture/Compare Register	0x0000_0000
CH0ACR	0x0A0	Channel 0 Asymmetric Compare Register	0x0000_0000
CH1ACR	0x0A4	Channel 1 Asymmetric Compare Register	0x0000_0000
CH2ACR	0x0A8	Channel 2 Asymmetric Compare Register	0x0000_0000
CH3ACR	0x0AC	Channel 3 Asymmetric Compare Register	0x0000_0000

## Register Descriptions

### Timer Counter Configuration Register – CNTCFR

This register specifies the GPTM counter configuration.

Offset: 0x000

Reset value: 0x0000\_0000

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							

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 mode or when used as a Quadrature decoder.</p>
[17:16]	CMSEL	<p>Counter Mode Selection</p> <p>00: Edge aligned mode. Normal up-counting and down-counting available for this mode. Counting direction is defined by the DIR bit.</p> <p>01: Center aligned 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 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 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 for 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	Update event interrupt generation disable control 0: Any of the following events will generate an update PDMA request or interrupt - Counter overflow/underflow - Setting the UEVG bit - Update generation through the slave mode 1: Only counter overflow/underflow generates an update PDMA request or interrupt
[0]	UEVDIS	Update event Disable control 0: Enable the update event request by one of following events: - Counter overflow/underflow - Setting the UEVG bit - Update generation through the slave mode 1: Disable the update event (However the counter and the prescaler are reinitialized if the UEVG bit is set or if a hardware restart is received from the slave mode)

## Timer Mode Configuration Register – MDCFR

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

Offset: 0x004

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved					SPMSET		
						RW	0	
Type/Reset	23	22	21	20	19	18	17	16
	Reserved					MMSEL		
						RW	0	
Type/Reset	15	14	13	12	11	10	9	8
	Reserved					SMSEL		
						RW	0	
Type/Reset	7	6	5	4	3	2	1	0
	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 GPTM external clock setting and the trigger source selection.

Offset: 0x008

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[24]	ECME	External Clock Mode Enable 0: External clock mode is disabled 1: External clock mode is enabled If the ECME bit is set to 1 and the timer is configured to be in the STI trigger slave mode in which the trigger source is derived from the GT_ETI pin, the external clock input on the GT_ETI pin is used.
[16]	ETIPOL	External Trigger Polarity 0: GT_ETI active at high level or rising edge 1: GT_ETI active at low level or falling edge
[13:12]	ETIPSC	External Trigger Prescaler A prescaler can be enabled to reduce the ETIP frequency. 00: Prescaler OFF 01: ETIP frequency divided by 2 10: ETIP frequency divided by 4 11: ETIP frequency divided by 8

Bits	Field	Descriptions
[11:8]	ETF	<p>External Trigger Filter</p> <p>These bits define the frequency divided ratio that is used to sample the GT_ETI signal. The digital filter in the GPTM is an N-event counter where N means 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_{DTS}</math></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>
[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: External Trigger input (ETIF)</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 32. GPTM Internal Trigger Connection**

Slave Timing Module	ITI0	ITI1	ITI2
GPTM0	GPTM1	MCTM0	MCTM1
GPTM1	GPTM0	MCTM0	MCTM1

## Timer Counter Register – CTR

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

Offset: 0x010

Reset value: 0x0000\_0000

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									
							CRBE	TME		
							RW	0	RW	0

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

## Channel 0 Input Configuration Register – CH0ICFR

This register specifies the channel 0 input mode configuration.

Offset: 0x020

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
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.

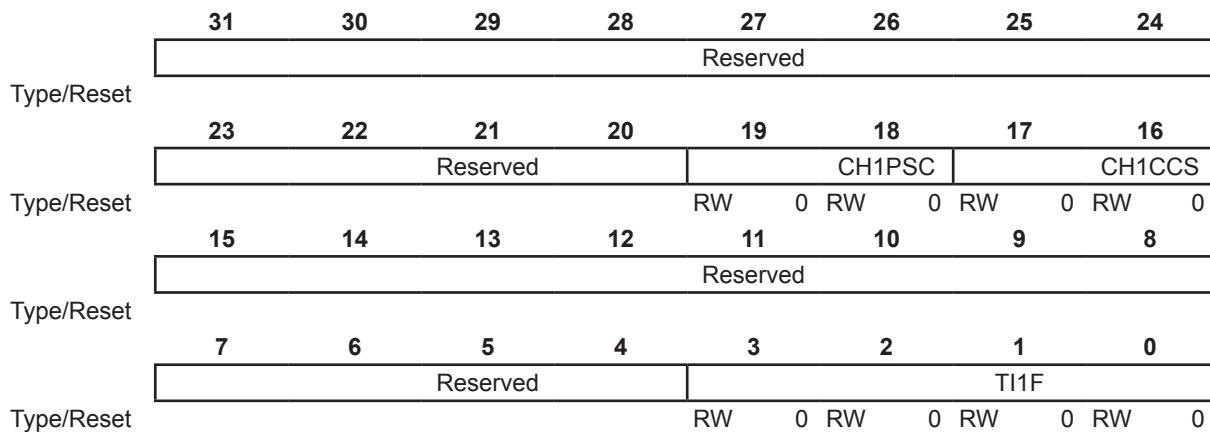
Bits	Field	Descriptions
[3:0]	TI0F	<p>Channel 0 Input Source TI0 Filter Setting</p> <p>These bits define the frequency divided ratio used to sample the TI0 signal. The Digital filter in the GPTM is an N-event counter where N is defined as how many valid transitions are necessary to output a filtered signal.</p> <ul style="list-style-type: none"> <li>0000: No filter, the sampling clock is <math>f_{DTS}</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>

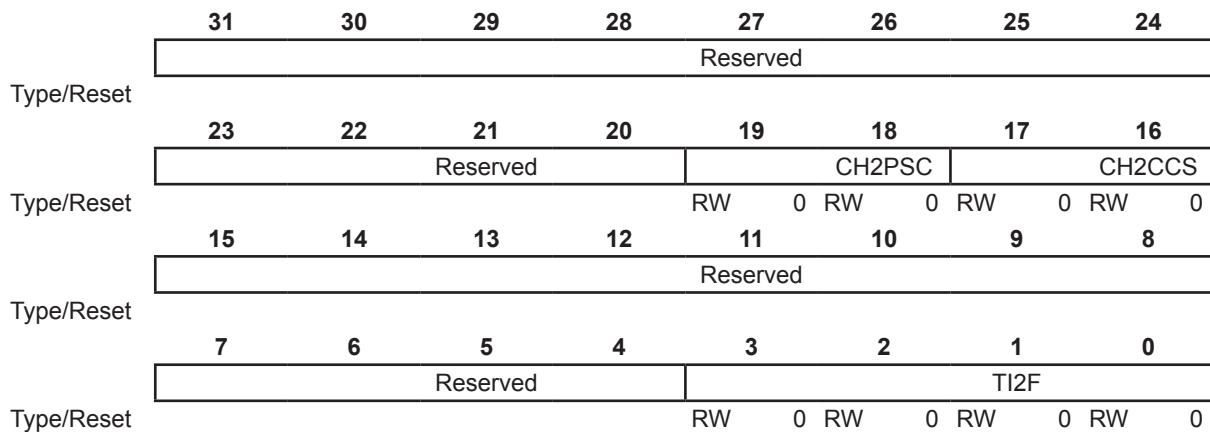
Bits	Field	Descriptions
[3:0]	TI1F	<p>Channel 1 Input Source TI1 Filter Setting</p> <p>These bits define the frequency divided ratio used to sample the TI1 signal. The Digital filter in the GPTM is an N-event counter where N is defined as how many valid transitions are necessary to output a filtered signal.</p> <ul style="list-style-type: none"> <li>0000: No filter, the sampling clock is <math>f_{DTS}</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>

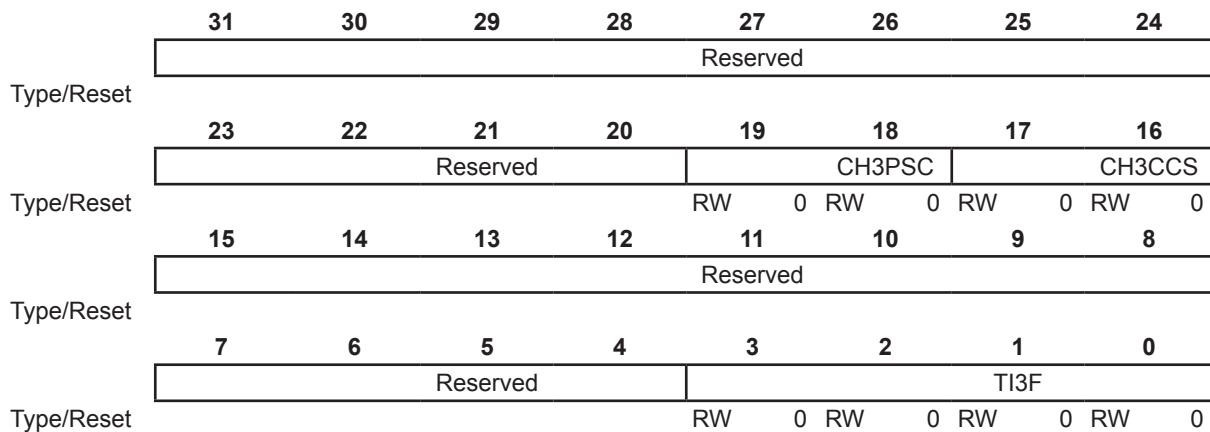
Bits	Field	Descriptions
[3:0]	TI2F	<p>Channel 2 Input Source TI2 Filter Setting</p> <p>These bits define the frequency divided ratio used to sample the TI2 signal. The Digital filter in the GPTM is an N-event counter where N is defined as how many valid transitions are necessary to output a filtered signal.</p> <ul style="list-style-type: none"> <li>0000: No filter, the sampling clock is <math>f_{DTS}</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_{DTS}</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

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							CH0OM[3]
	7	6	5	4	3	2	1	0
Type/Reset	Reserved	CH0IMAE	CH0PRE	REF0CE	CH0OM[2:0]			
	RW	0	RW	0	RW	0	RW	0

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>
[3]	REF0CE	<p>Channel 0 Reference Output Clear Enable</p> <p>0: CH0OREF performed normally and is not affected by the ETIF signal 1: CH0OREF is forced to 0 on the high level of the ETIF signal derived from the GT ETI pin</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>

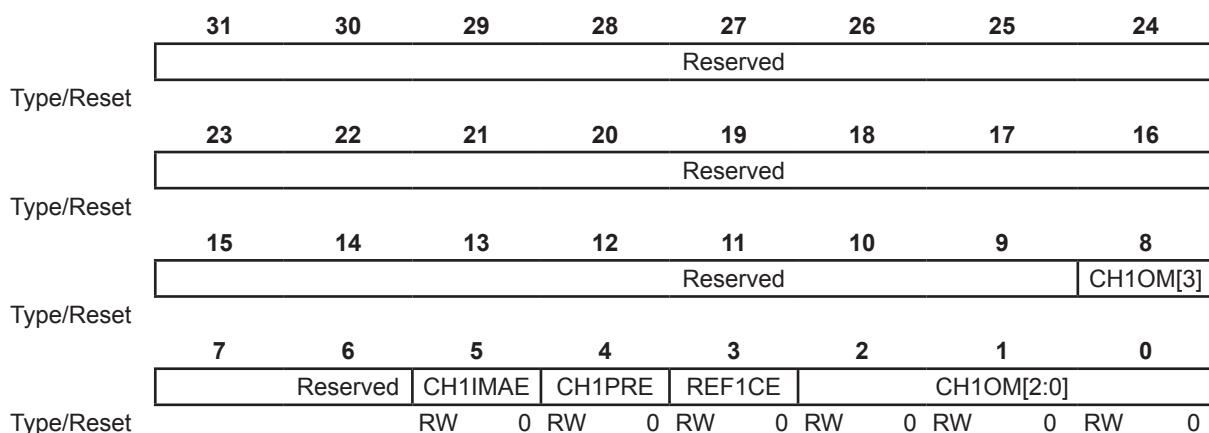
Note: When channel 0 is used as asymmetric PWM output mode, the Counter Mode Selection bit in Counter Configuration Register must be configured as Center-aligned Counting mode (CMSEL = 01/02/03)

## 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</p> <p>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.</p> <p>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</p> <p>The new CH1CCR value will not be transferred to its shadow register until the update event occurs.</p>
[3]	REF1CE	<p>Channel 1 Reference Output Clear Enable</p> <p>0: CH1OREF performed normally and is not affected by the ETIF signal</p> <p>1: CH1OREF is forced to 0 on the high level of the ETIF signal derived from the GT_ETI pin</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

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								CH2OM[3]	
Type/Reset	7	6	5	4	3	2	1	0		
	Reserved	CH2IMAE	CH2PRE	REF2CE	CH2OM[2:0]					
Type/Reset		RW	0	RW	0	RW	0	RW	0	

Bits	Field	Descriptions
[5]	CH2IMAE	<p>Channel 2 Immediate Active Enable</p> <p>0: No action</p> <p>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.</p> <p>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</p> <p>The new CH2CCR value will not be transferred to its shadow register until the update event occurs.</p>
[3]	REF2CE	<p>Channel 2 Reference Output Clear Enable</p> <p>0: CH2OREF performed normally and is not affected by the ETIF signal</p> <p>1: CH2OREF is forced to 0 on the high level of the ETIF signal derived from the GT_ETI pin</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>1100: 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

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								CH3OM[3]	
Type/Reset	7	6	5	4	3	2	1	0		
	Reserved	RW	0	RW	0	RW	0	RW	0	
Type/Reset			CH3IMAE	CH3PRE	REF3CE	CH3OM[2:0]				

Bits	Field	Descriptions
[5]	CH3IMAE	<p>Channel 3 Immediate Active Enable</p> <p>0: No action</p> <p>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.</p> <p>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</p> <p>The new CH3CCR value will not be transferred to its shadow register until the update event occurs.</p>
[3]	REF3CE	<p>Channel 3 Reference Output Clear Enable</p> <p>0: CH3OREF performed normally and is not affected by the ETIF signal</p> <p>1: CH3OREF is forced to 0 on the high level of the ETIF signal derived from the GT_ETI pin</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>1100: 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>

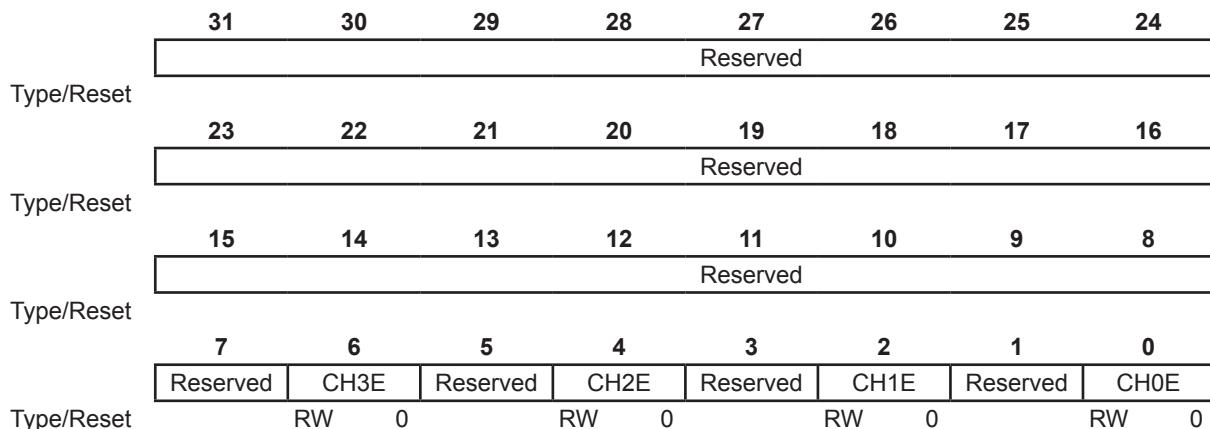
Note: When channel 3 is used as asymmetric PWM output mode, the Counter Mode Selection bit in Counter Configuration Register must be configured as Center-aligned Counting mode (CMSEL = 01/02/03)

## Channel Control Register – CHCTR

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

Offset: 0x050

Reset value: 0x0000\_0000



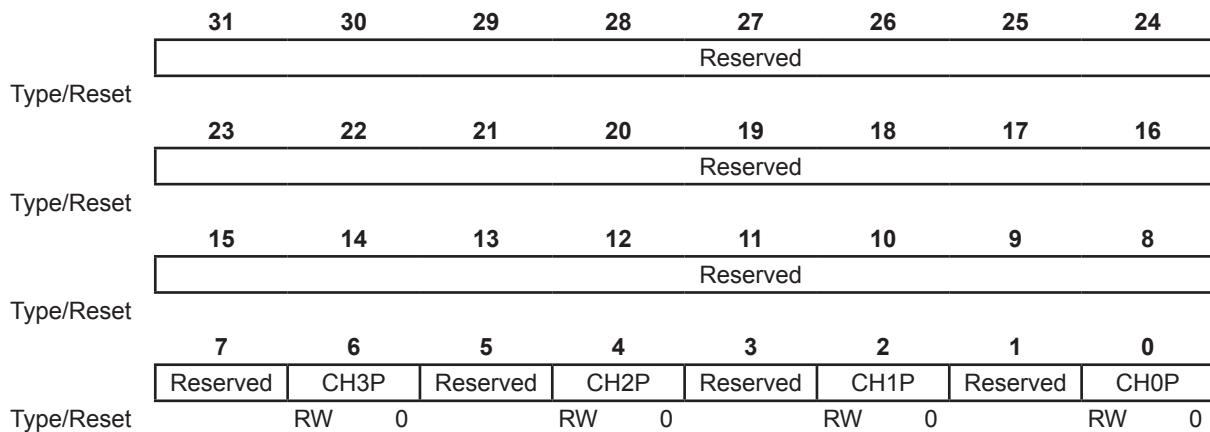
Bits	Field	Descriptions
[6]	CH3E	Channel 3 Capture/Compare Enable - Channel 3 is configured as an input (CH3CCS = 0x01/0x02/0x03) 0: Input Capture Mode disabled 1: Input Capture Mode enabled - Channel 3 is configured as an output (CH3CCS = 0x00) 0: Off – Channel 3 output signal CH3O is not active 1: 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 - When 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 PDMA/Interrupt Control Register – DICTR

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

Offset: 0x074

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[26]	TEVDE	Trigger event PDMA Request Enable 0: Trigger PDMA request disabled 1: Trigger PDMA request enabled
[24]	UEVDE	Update event PDMA Request Enable 0: Update event PDMA request disabled 1: Update event PDMA request enabled
[19]	CH3CCDE	Channel 3 Capture/Compare PDMA Request Enable 0: Channel 3 PDMA request disabled 1: Channel 3 PDMA request enabled
[18]	CH2CCDE	Channel 2 Capture/Compare PDMA Request Enable 0: Channel 2 PDMA request disabled 1: Channel 2 PDMA request enabled
[17]	CH1CCDE	Channel 1 Capture/Compare PDMA Request Enable 0: Channel 1 PDMA request disabled 1: Channel 1 PDMA request enabled
[16]	CH0CCDE	Channel 0 Capture/Compare PDMA Request Enable 0: Channel 0 PDMA request disabled 1: Channel 0 PDMA request enabled
[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

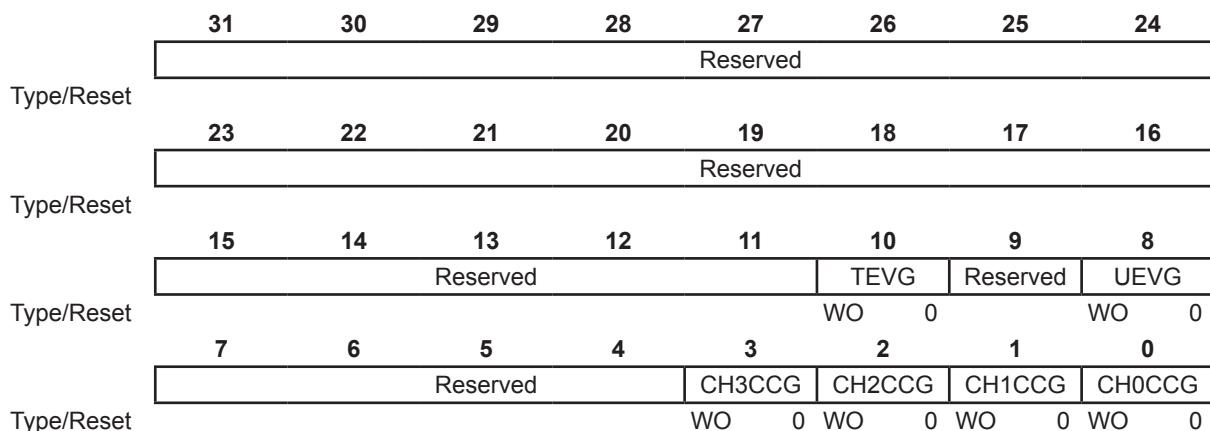
<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>
[2]	CH2CCIE	Channel 2 Capture/Compare Interrupt Enable 0: Channel 2 interrupt disabled 1: Channel 2 interrupt enabled
[1]	CH1CCIE	Channel 1 Capture/Compare Interrupt Enable 0: Channel 1 interrupt disabled 1: Channel 1 interrupt enabled
[0]	CH0CCIE	Channel 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	
							RW	0	
Type/Reset	7	6	5	4	3	2	1	0	
	CH3OCF	CH2OCF	CH1OCF	CH0OCF	CH3CCIF	CH2CCIF	CH1CCIF	CH0CCIF	
Type/Reset	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.

Bits	Field	Descriptions
[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 style="list-style-type: none"> <li>0: No match event occurs</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 except in the center-aligned mode. It is cleared by software.</p> <li>- Channel 3 is configured as an input:</li> <ul style="list-style-type: none"> <li>0: No input capture occurs</li> <li>1: Input capture occurs</li> </ul> <p>This bit is set by hardware on a capture event. It is cleared by software or by reading the CH3CCR register.</p> </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:</li> <ul style="list-style-type: none"> <li>0: No match event occurs</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 except in the center-aligned mode. It is cleared by software.</p> <li>- Channel 2 is configured as an input:</li> <ul style="list-style-type: none"> <li>0: No input capture occurs</li> <li>1: Input capture occurs.</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> </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:</li> <ul style="list-style-type: none"> <li>0: No match event occurs</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 except in the center-aligned mode. It is cleared by software.</p> <li>- Channel 1 is configured as an input:</li> <ul style="list-style-type: none"> <li>0: No input capture occurs</li> <li>1: Input capture occurs</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> </ul>

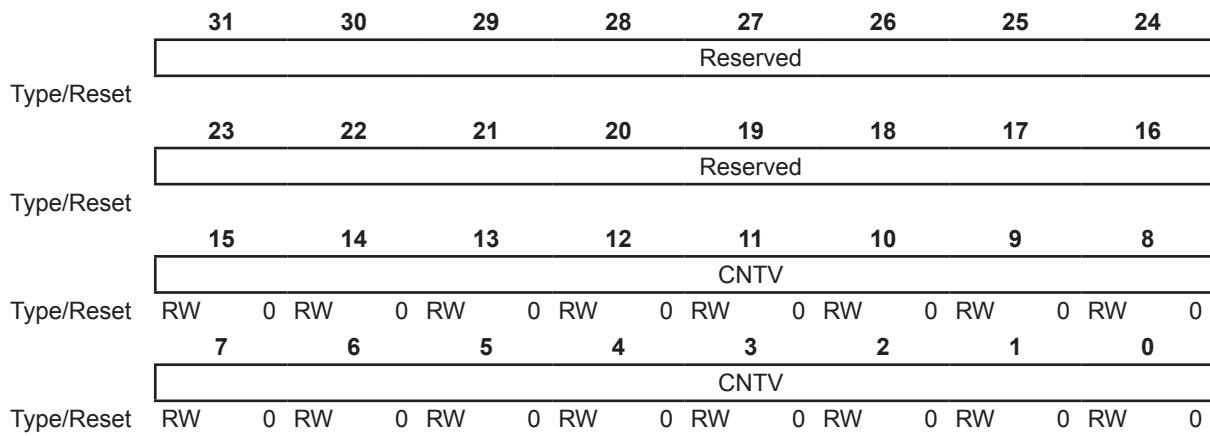
Bits	Field	Descriptions
[0]	CH0CCIF	<p>Channel 0 Capture/Compare Interrupt Flag</p> <p>- Channel 0 is configured as an output:</p> <p>0: No match event occurs</p> <p>1: The contents of the counter CNTR have matched the content of the CH0CCR register</p> <p>This flag is set by hardware when the counter 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

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

Bits	Field	Descriptions
[15:0]	PSCV	<p>Prescaler Value</p> <p>These bits are used to specify the prescaler value to generate the counter clock frequency <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

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

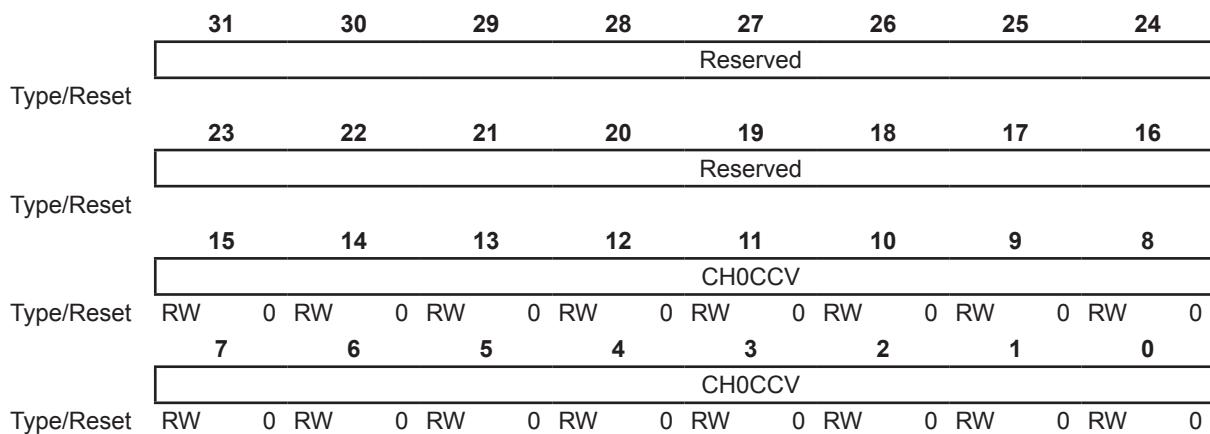
Bits	Field	Descriptions
[15:0]	CRV	Counter Reload Value The CRV is the reload value which is loaded into the actual counter register.

## 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
Reserved								
Type/Reset	23	22	21	20	19	18	17	16
Reserved								
Type/Reset	15	14	13	12	11	10	9	8
CH1CCV								
Type/Reset	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
Type/Reset	RW	0	RW	0	RW	0	RW	0
CH1CCV								
Type/Reset	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

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

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

Bits	Field	Descriptions
[15:0]	CH3CCV	<p>Channel 3 Capture/Compare Value</p> <ul style="list-style-type: none"> <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
Reserved								
Type/Reset	23	22	21	20	19	18	17	16
Reserved								
Type/Reset	15	14	13	12	11	10	9	8
CH0ACV								
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								
Type/Reset	RW	0	RW	0	RW	0	RW	0

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

	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
CH1ACV								
Type/Reset	RW	0	RW	0	RW	0	RW	0
CH1ACV								
Type/Reset	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[15:0]	CH1ACV	Channel 1 Asymmetric Compare Value When channel 1 is configured as asymmetric PWM mode and the counter is counting down, the value written is 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
Reserved								
Type/Reset	23	22	21	20	19	18	17	16
Reserved								
Type/Reset	15	14	13	12	11	10	9	8
CH2ACV								
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								
Type/Reset	RW	0	RW	0	RW	0	RW	0

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

## Channel 3 Asymmetric Compare Register – CH3ACR

This register specifies the timer channel 3 asymmetric compare value.

Offset: 0x0AC

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[15:0]	CH3ACV	<p>Channel 3 Asymmetric Compare Value</p> <p>When channel 3 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>

# 15 Basic Function Timer (BFTM)

## Introduction

The Basic Function Timer Module, BFTM, is a 32-bit up-counting counter designed to measure time intervals, generate one shot 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.

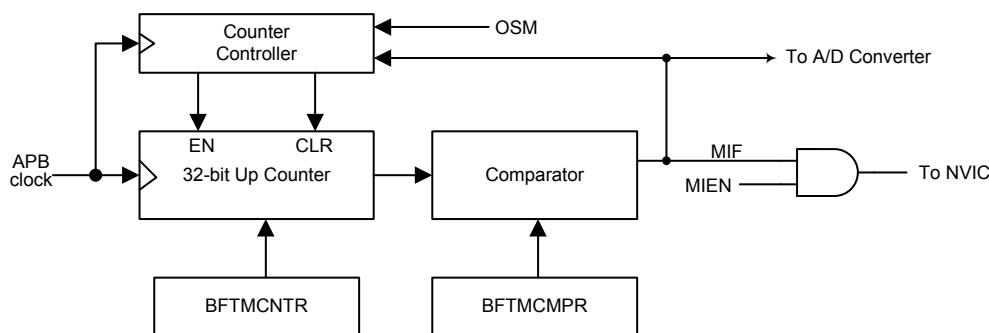


Figure 71. BFTM Block Diagram

## Features

- 32-bit up-counting counter
- Compare Match function
- Includes debug mode
- Clock source: 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 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.

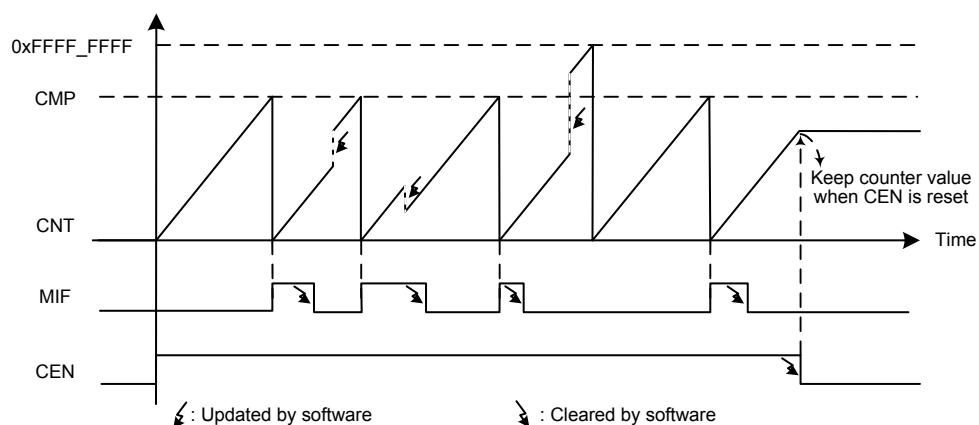


Figure 72. BFTM – Repetitive Mode

## 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.

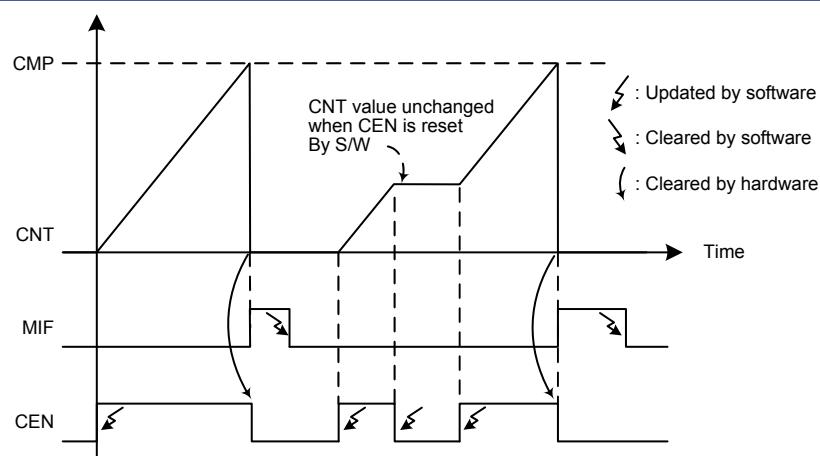


Figure 73. BFTM – One Shot Mode

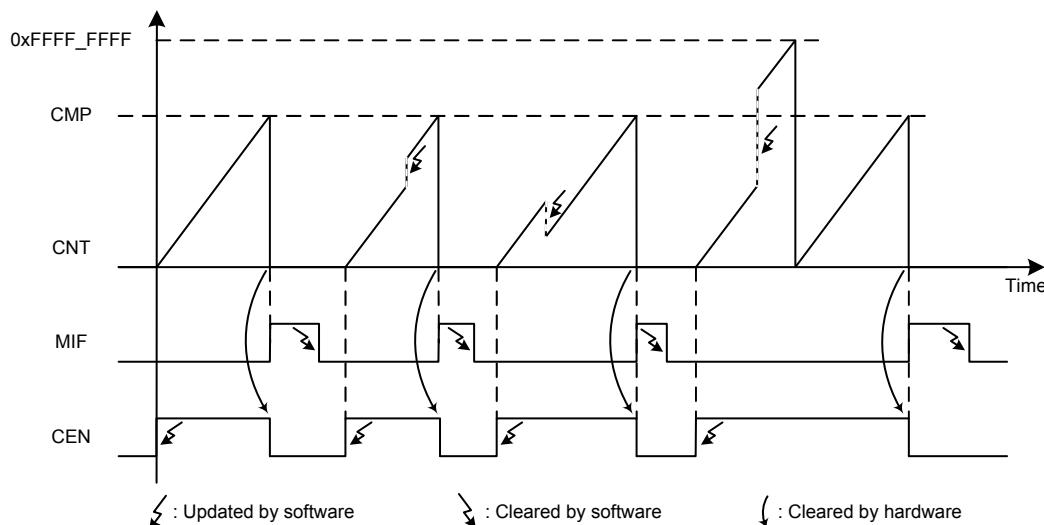


Figure 74. BFTM – One Shot Mode Counter Updating

## 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 33. BFTM Register Map**

Register	Offset	Description	Reset Value
<b>BFTM0 Base Address = 0x4007_6000</b>			
<b>BFTM1 Base Address = 0x4007_7000</b>			
BFTMCR	0x000	BFTM Control Register	0x0000_0000
BFTMSR	0x004	BFTM Status Register	0x0000_0000
BFTMCNTR	0x008	BFTM Counter Value Register	0x0000_0000
BFTMCMPPR	0x00C	BFTM Compare Value Register	0xFFFF_FFFF

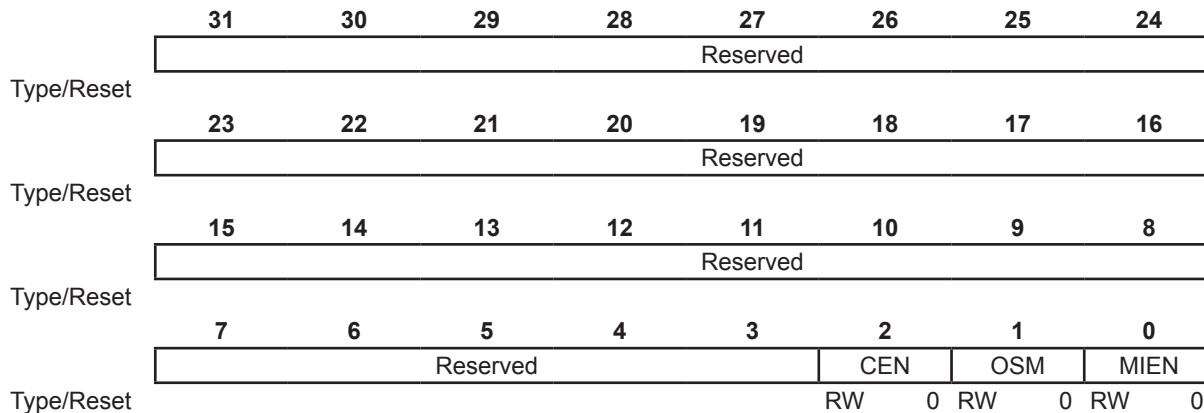
## Register Descriptions

### BFTM Control Register – BFTMCR

This register specifies the overall BFTM control bits.

Offset: 0x000

Reset value: 0x0000\_0000



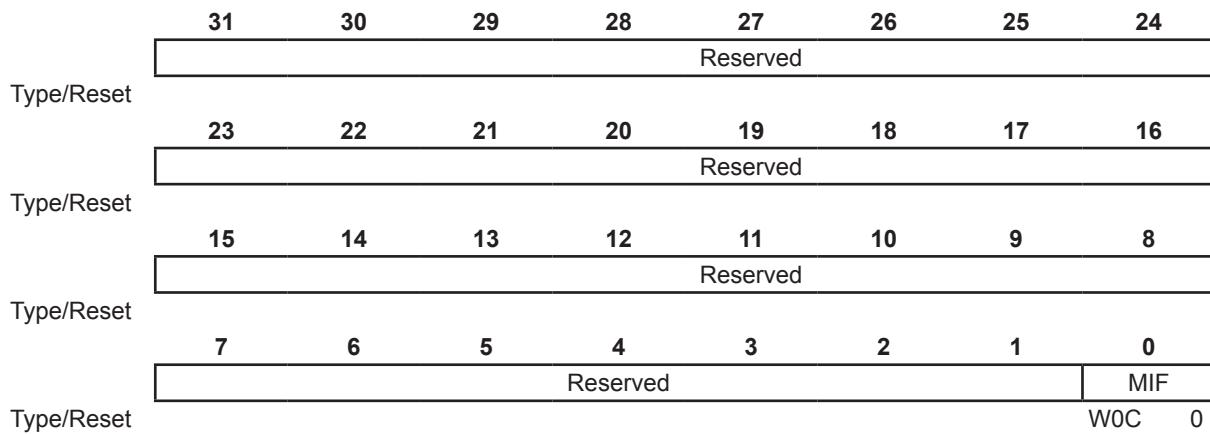
Bits	Field	Descriptions
[2]	CEN	BFTM Counter Enable Control 0: BFTM is disabled 1: BFTM is enabled  When this bit is set to 1, the BFTM counter will start to count. The counter will stop counting and the counter value will remain unchanged when the CEN bit is cleared to 0 by the application program regardless 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\_0004



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

## BFTM Counter Register – BFTMCNTR

This register specifies the BFTM counter value.

Offset: 0x008

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

# 16 Motor Control Timer (MCTM)

## Introduction

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

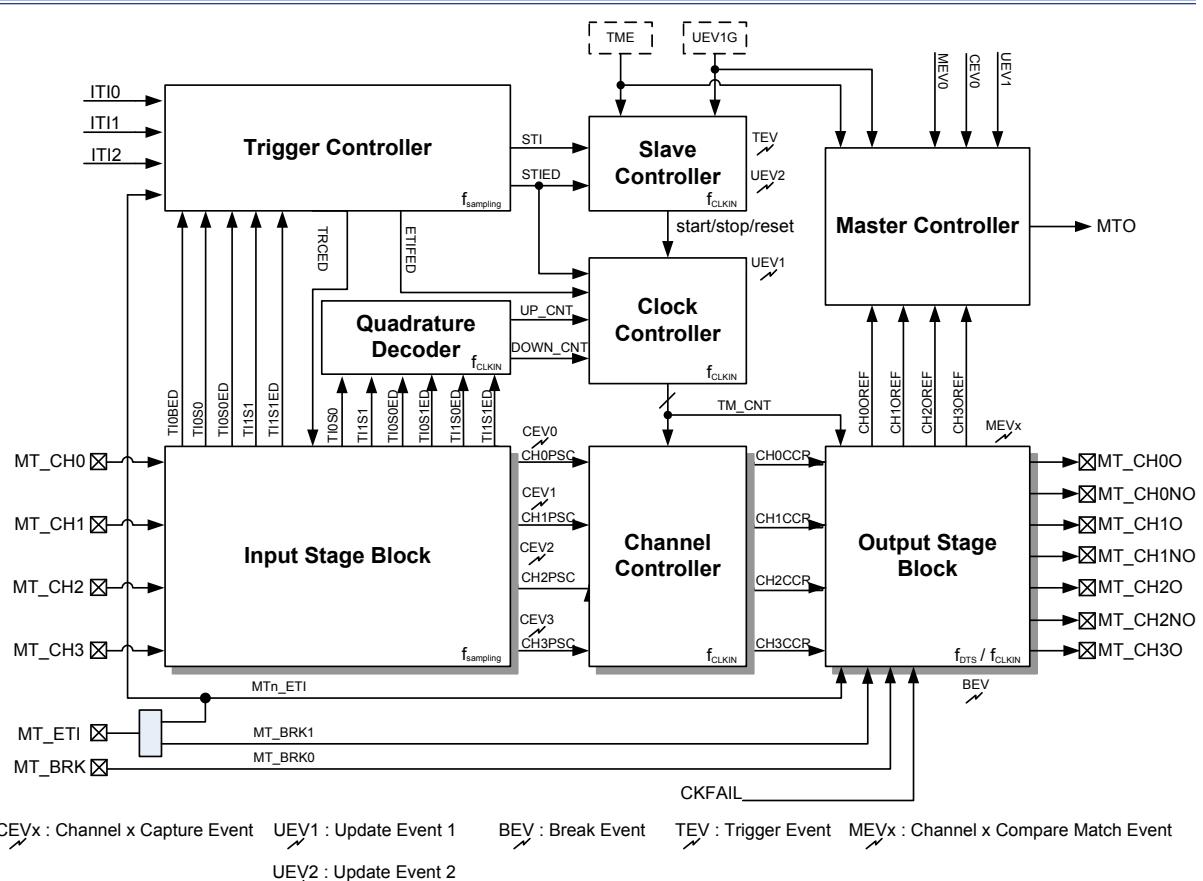


Figure 75. MCTM Block Diagram

## 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
- Encoder interface controller with two inputs using quadrature decoder
- Repetition counter updates timer registers only after a given number of counter cycles.
- Synchronization circuit controls the timer with external signals and can interconnect several timers together.
- Interrupt/PDMA generation on the following events:
  - Update event 1
  - Update event 2
  - Trigger event
  - Input capture event
  - Output compare match
  - Break event - only interrupt
- MCTM Master/Slave mode controller
- Supports 3-phase motor control and hall sensor interface
- Maximum 2 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.

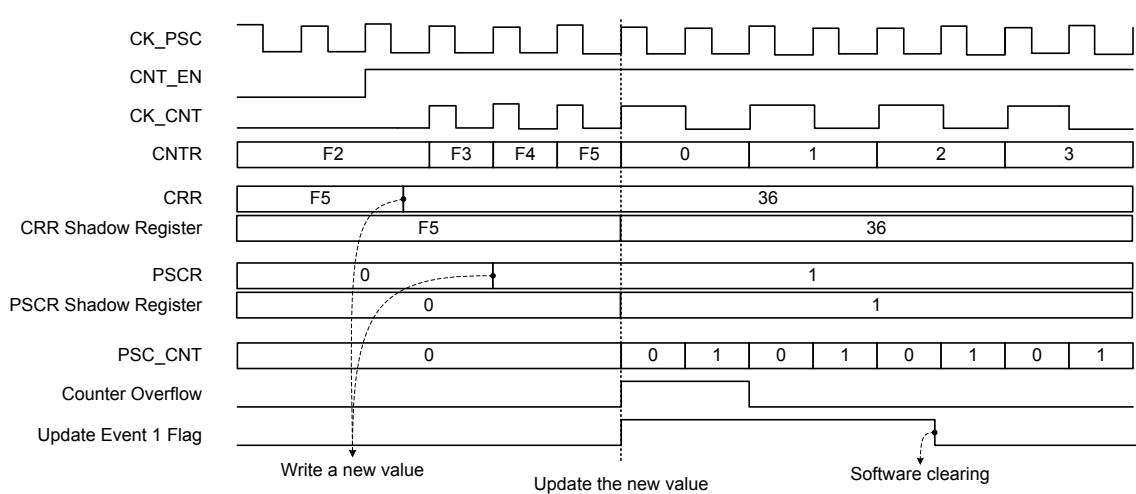
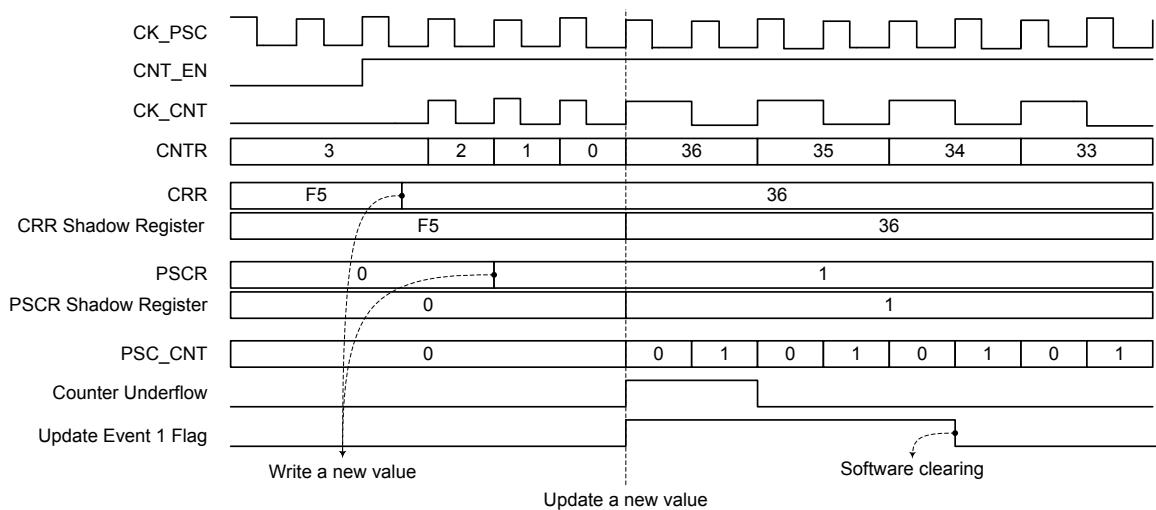


Figure 76. Up-counting Example

### 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.



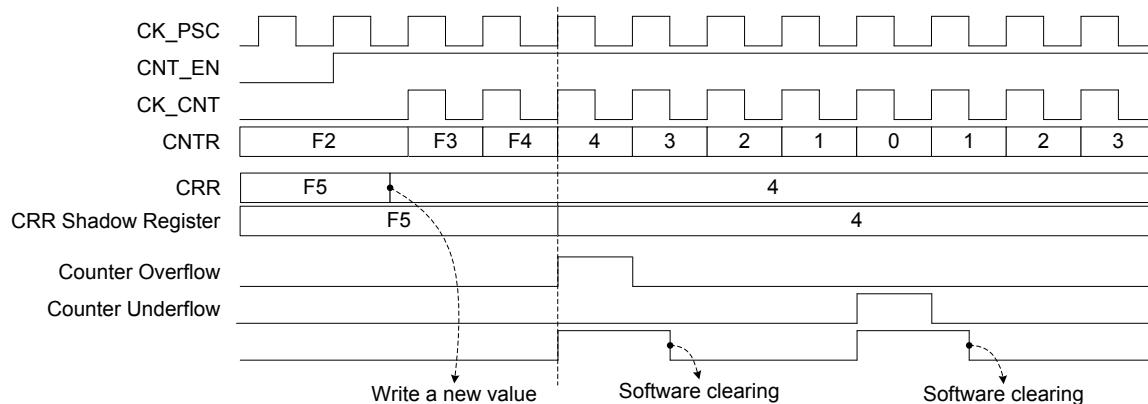
**Figure 77. Down-counting Example**

### 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.

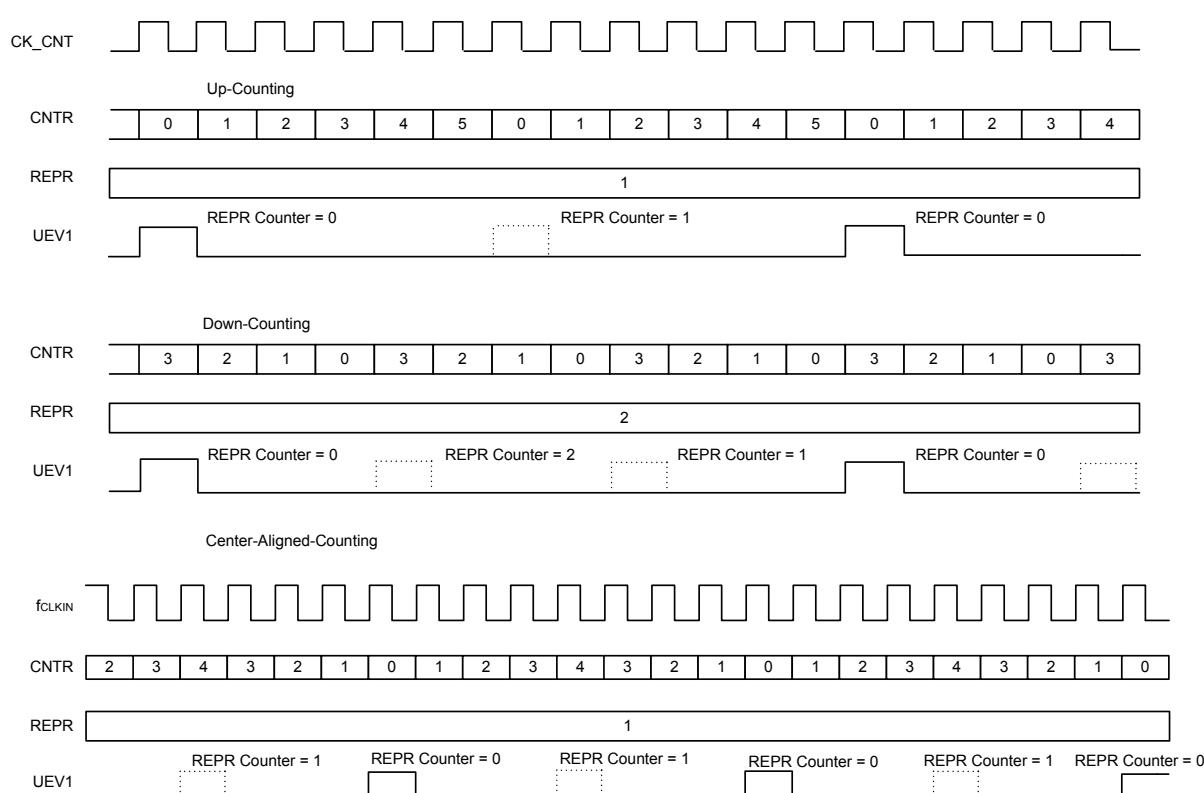


**Figure 78. Center-aligned Counting Example**

### 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 at each counter underflow in the center-aligned counting mode



**Figure 79. Update Event 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.

### Quadrature Decoder

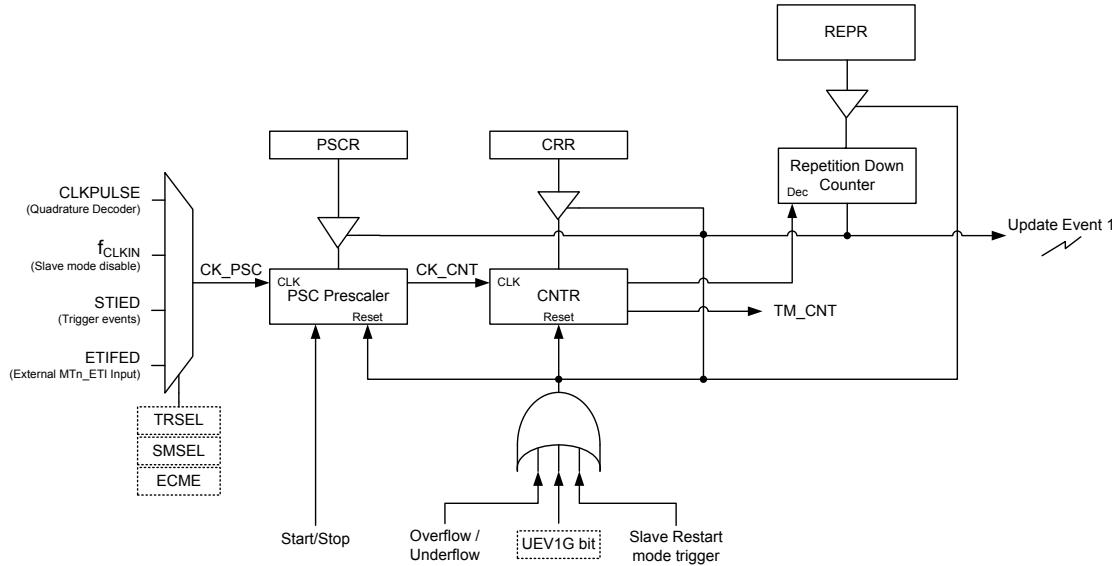
To select the Quadrature Decoder mode the SMSEL field should be set to 0x1, 0x2 or 0x3 in the MDCFR register. The Quadrature Decoder function uses the two input conditions of the MT\_CH0 and MT\_CH1 pins to generate the clock pulses to drive the counter prescaler. The counting direction bit DIR is modified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

### 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.

### ETIFED

The counter prescaler can be driven to count during each rising edge on ETIF. This mode can be selected by setting the ECME bit in the TRCFR register to 1. The other way to select the ETIF signal as the clock source is to set the SMSEL field to 0x7 and the TRSEL field to 0x3 respectively. When the clock source is selected to come from the ETIF signal, the Trigger Controller including the edge detection circuitry will generate a clock pulse during each ETIF signal rising edge to clock the counter prescaler.

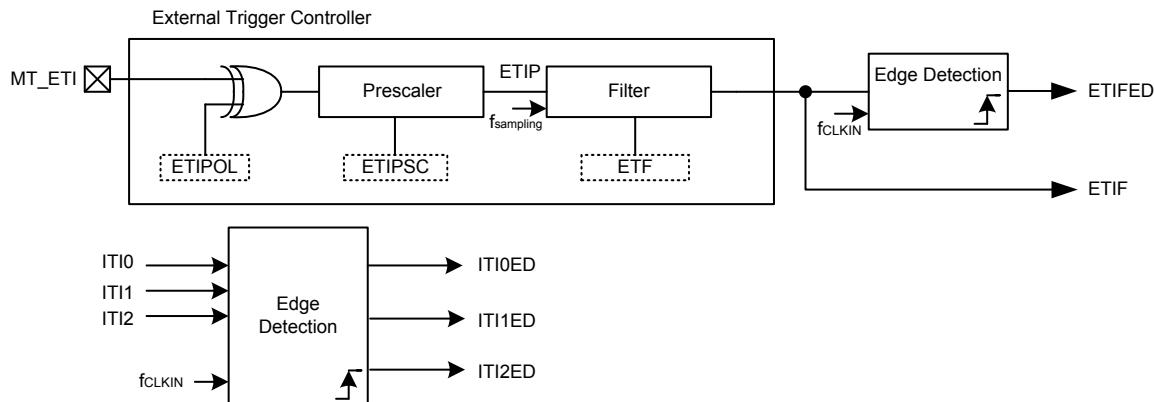


**Figure 80. MCTM Clock Selection Source**

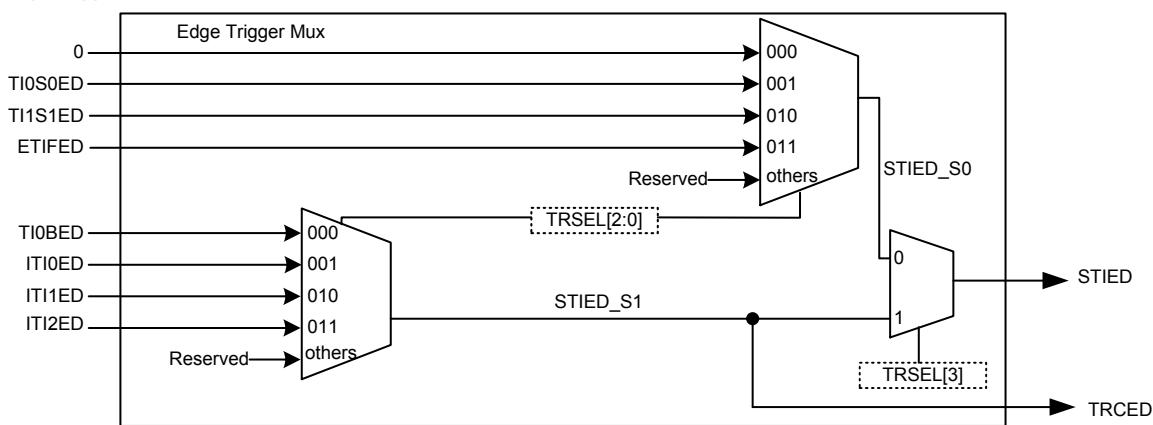
## Trigger Controller

The trigger controller is used to select the trigger source and setup the trigger level and edge trigger conditions. The active polarity of the external trigger input signal MT\_ETI can be configured by the External Trigger Polarity control bit, ETIPOL, in the MCTM Trigger Configuration Register TRCFR. The frequency of the external trigger input can be divided by configuring the related bits, which are the External Trigger Prescaler control bits, ETIPSC, in the TRCFR register. The trigger signal can also be filtered by configuring the External Trigger Filter ETF selection bits in the TRCFR register if a filtered signal is necessary for specific applications. For the internal trigger input, it can be selected by the Trigger Selection bits, TRSEL, in the TRCFR register. For all the trigger sources except the UEV1G bit software trigger, the internal edge detection circuitry will generate a clock pulse at each trigger signal rising edge to activate some MCTM functions which are triggered by a trigger signal rising edge.

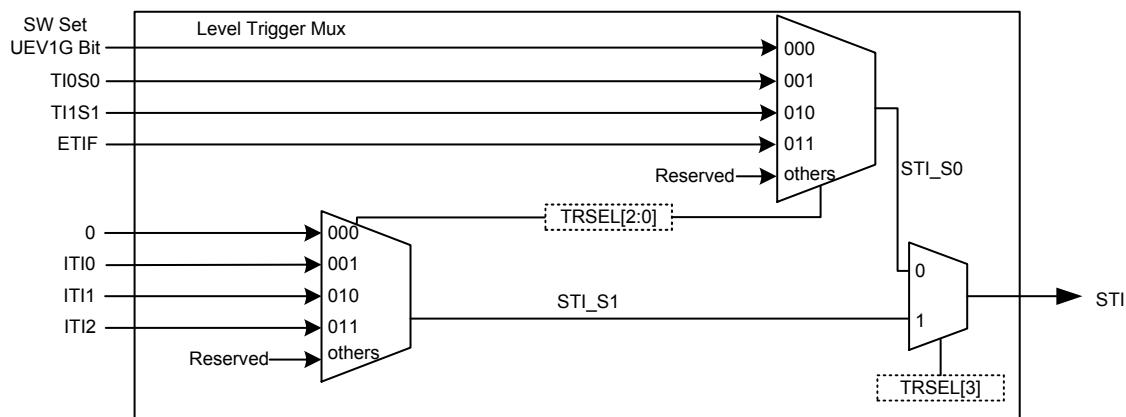
**Trigger Controller Block = Edge Trigger Mux + Level Trigger Mux**



**Edge Trigger = External (ETI)+ Internal (ITIx) + Channel input (CHn) + XOR function**



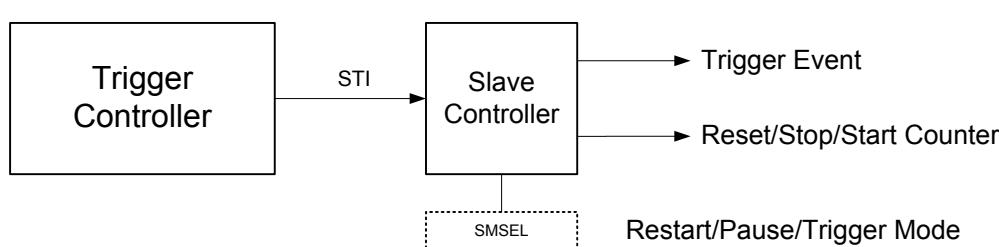
**Level Trigger Source = External (ETI)+ Internal (ITIx) + Channel input (CHn) + Software UEV1G bit**



**Figure 81. Trigger Control Block**

## Slave Controller

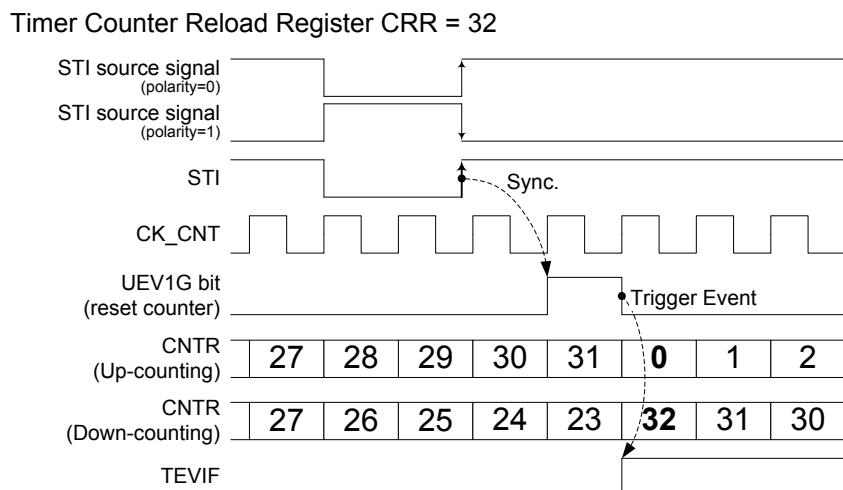
The MCTM can be synchronised with an internal/external trigger in several modes including the Restart mode, the Pause mode and the Trigger mode which are selected by the SMSEL field in the MDCFR register. The trigger input of these modes comes from the STI signal which is selected by the TRSEL field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



**Figure 82. 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 reinitialised when an STI rising edge occurs. If the UEV1DIS bit in the CNTCFR register is cleared to enable the update event, then an update event will be generated together with the STI rising edge and all the preloaded registers will be updated.



**Figure 83. 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.

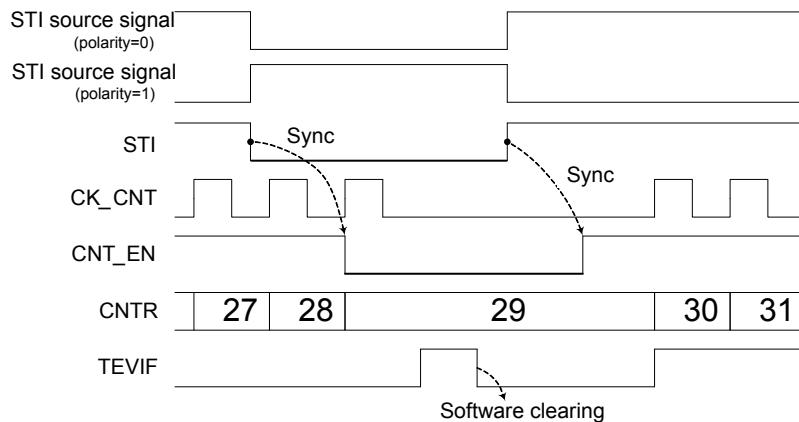
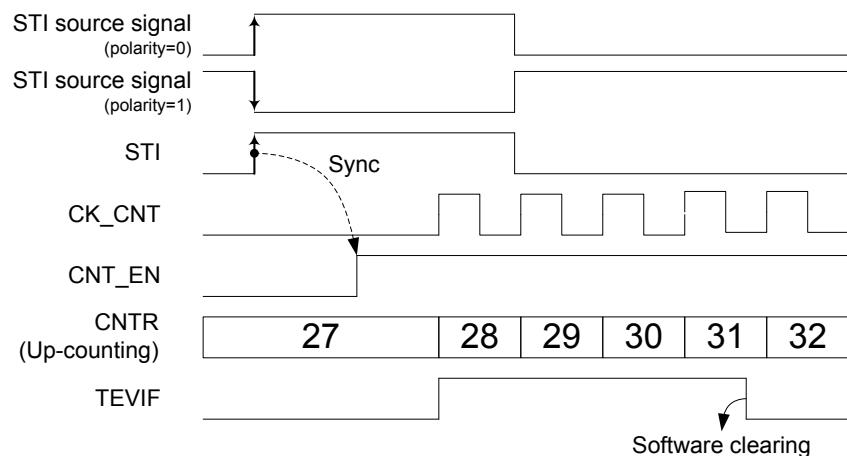


Figure 84. MCTM in Pause Mode

### Trigger Mode

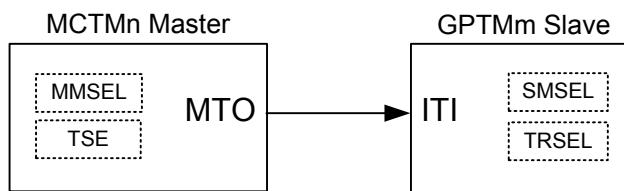
After the counter is disabled to count, the counter can resume counting when an STI rising edge signal occurs. When an STI rising edge occurs, the counter will start to count from the current value in the counter. Note that if the STI signal is selected to be sourced from the UEV1G bit software trigger, 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 85. MCTM in Trigger Mode**

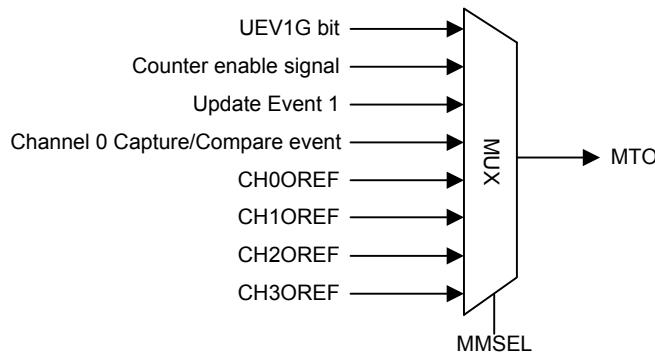
### Master Controller

The MCTMs and GPTMs can be linked together internally for timer synchronisation or chaining. When one MCTM is configured to be in the Master Mode, the MCTM Master Controller will generate a Master Trigger Output (MTO) signal which can reset, start, 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 86. 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 87. MTO Selection**

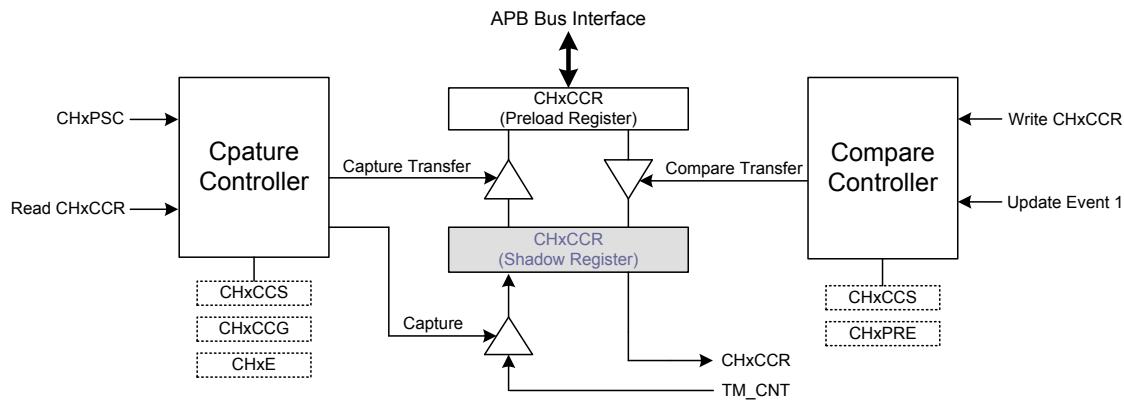
For example, setting the MMSEL field to 0x5 is to select the CH1OREF signal as the MTO signal to synchronise another slave MCTM or GPTM. For a more detailed description, refer to the related MMSEL field definitions in the MDCFR register.

### 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 CH<sub>x</sub>CCR shadow register first and then transferred into the CH<sub>x</sub>CCR preload register when the capture event occurs.

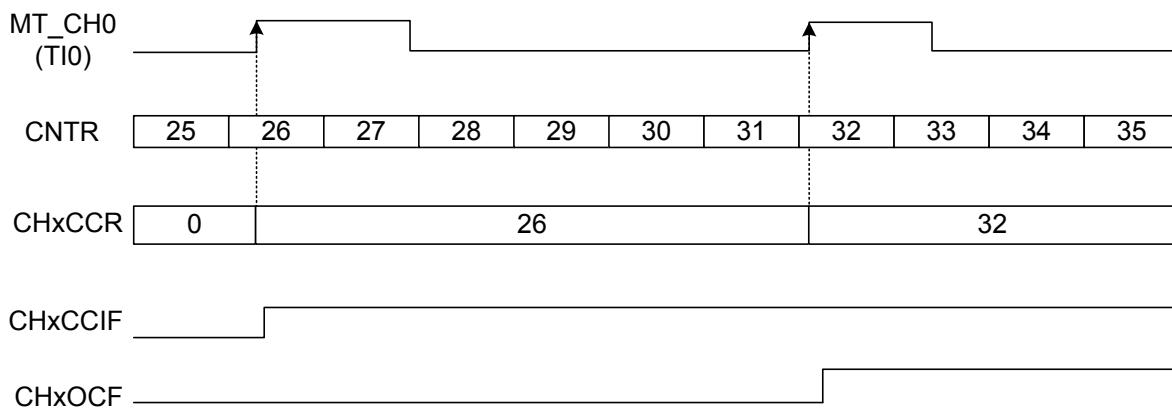
When used in the compare match output mode, the contents of the CH<sub>x</sub>CCR preload register is copied into the associated shadow register, the counter value is then compared with the register value.



**Figure 88. 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.



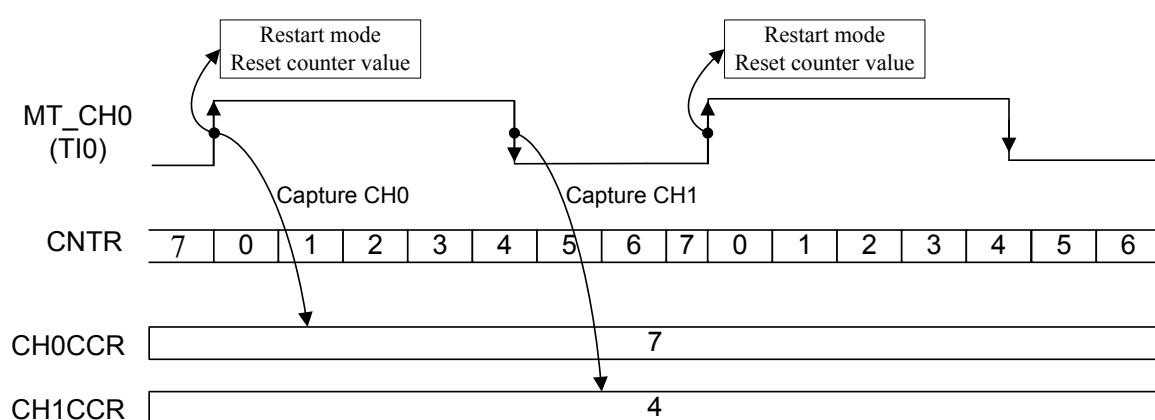
**Figure 89. Input Capture Mode**

### 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 90. 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.

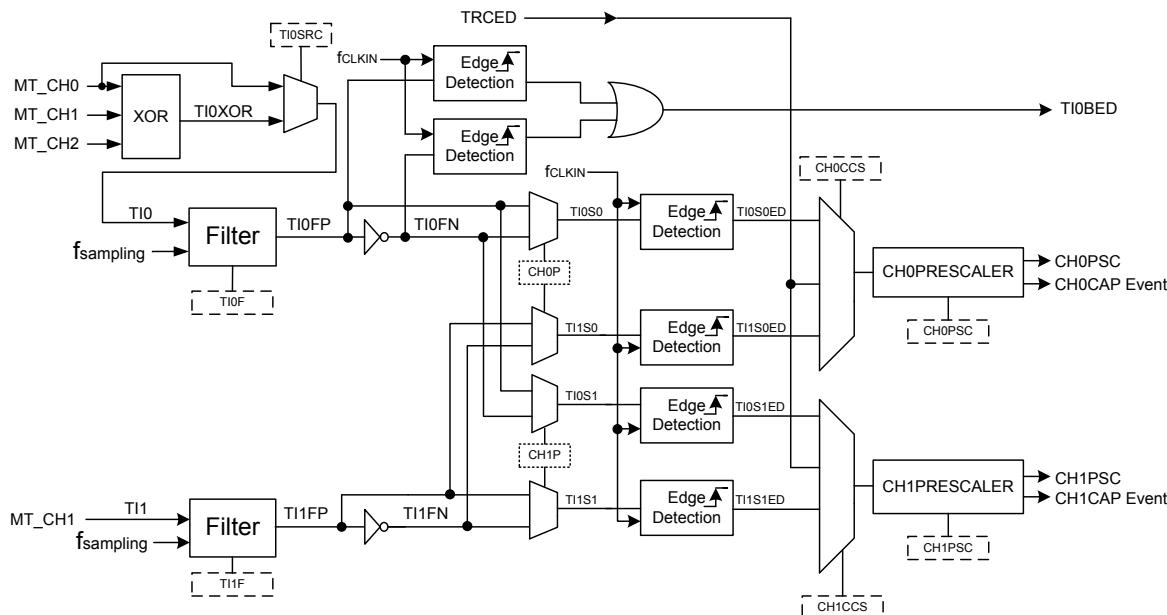


Figure 91. Channel 0 and Channel 1 Input Stages

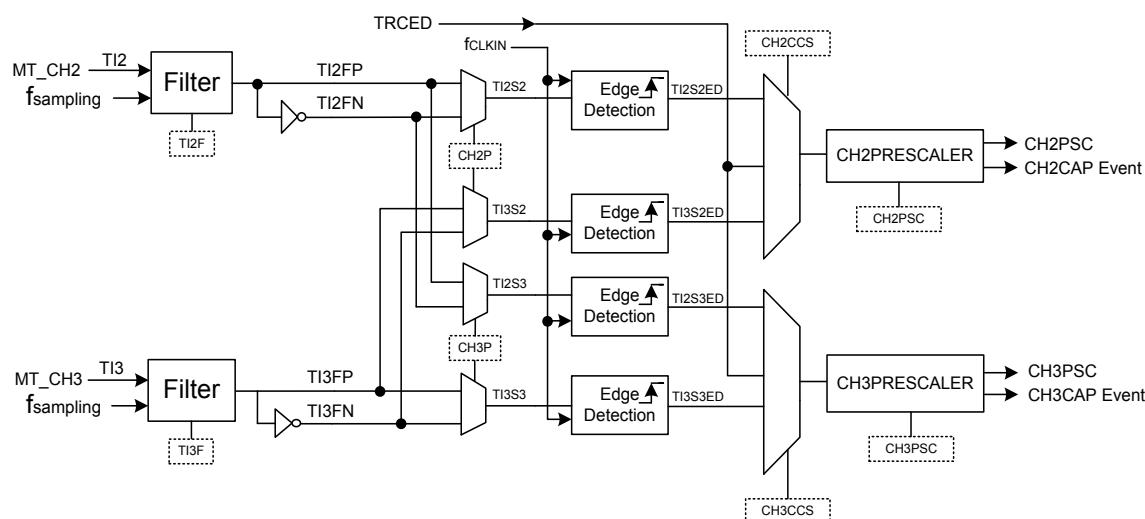
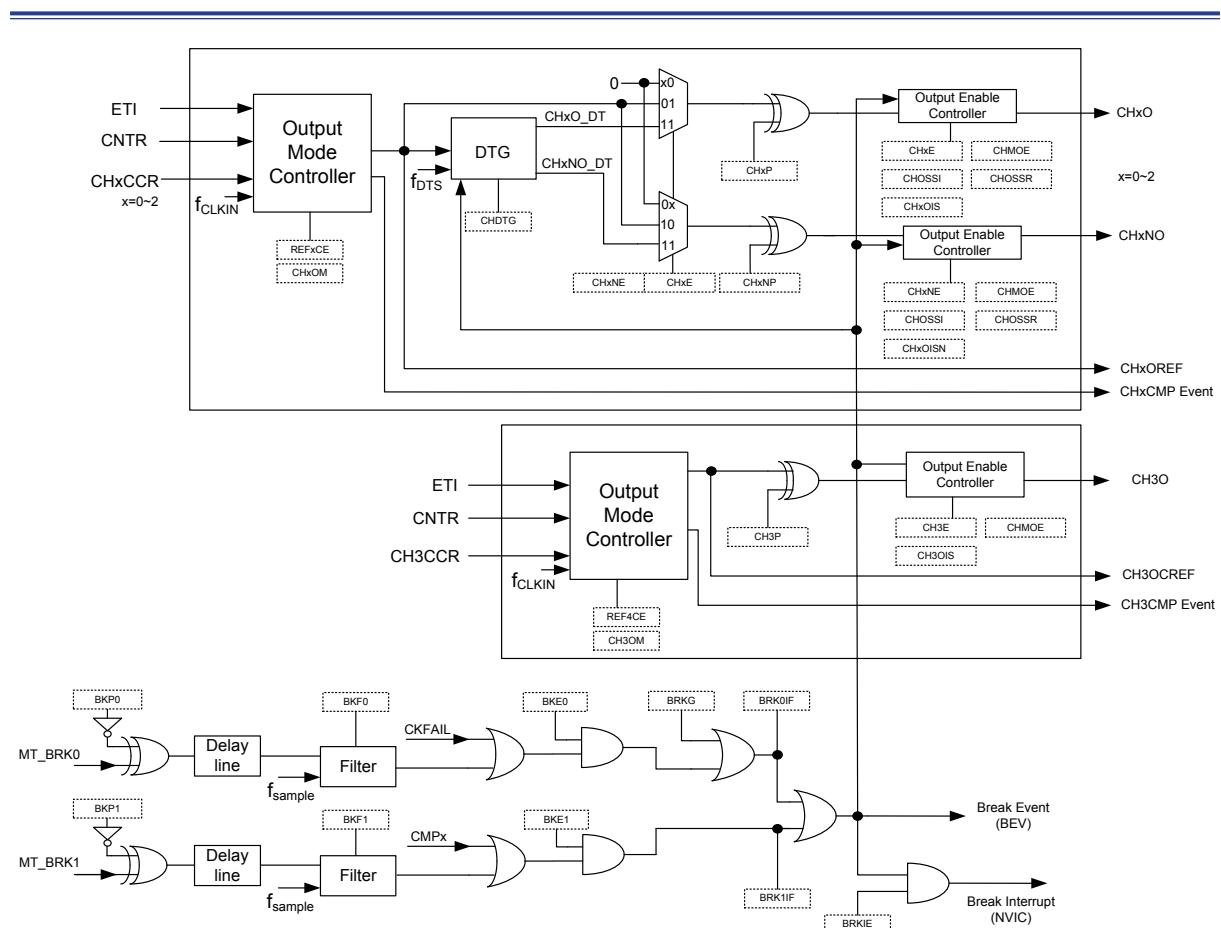


Figure 92. Channel 2 and Channel 3 Input Stages

## 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 93. Output Stage Block Diagram**

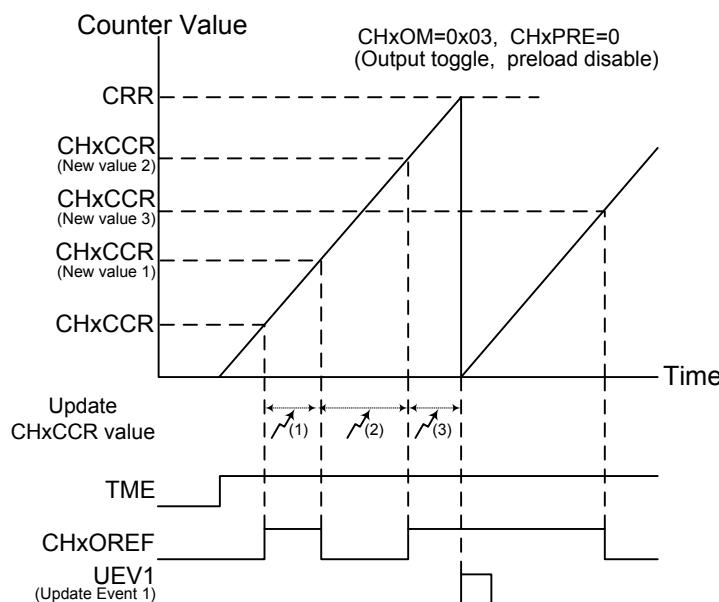
### Channel Output Reference Signal

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 shows a summary of the output type setup.

**Table 34. 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 94. Toggle Mode Channel Output Reference Signal – CHxPRE = 0**

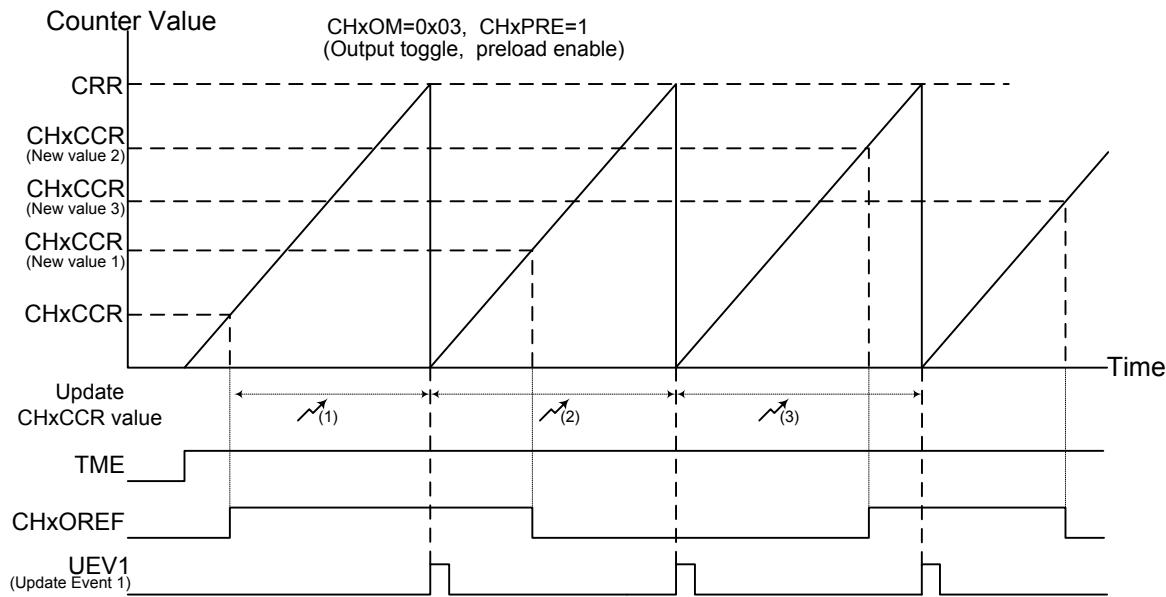


Figure 95. Toggle Mode Channel Output Reference Signal – CH<sub>x</sub>PRE = 1

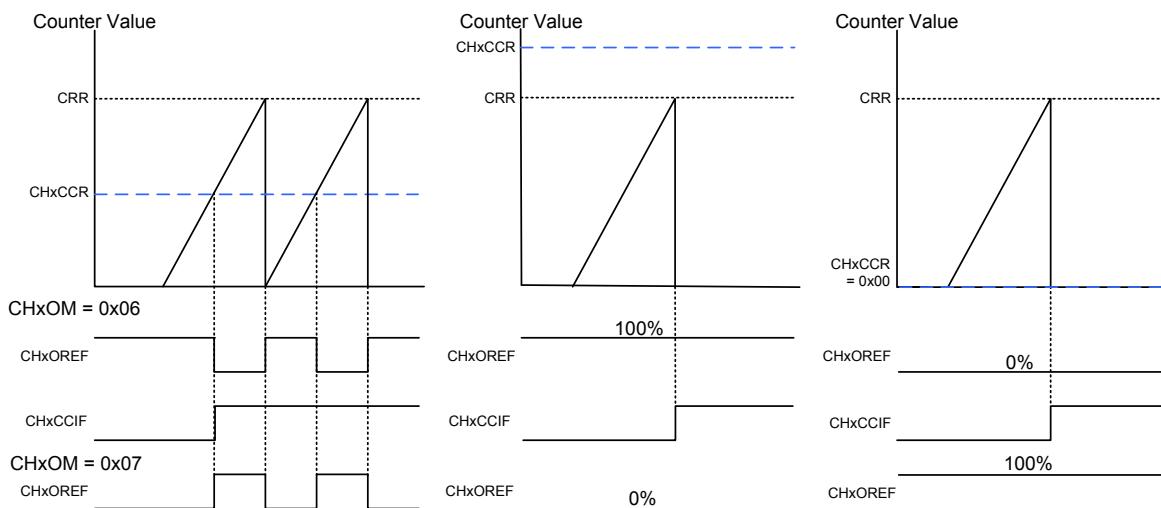
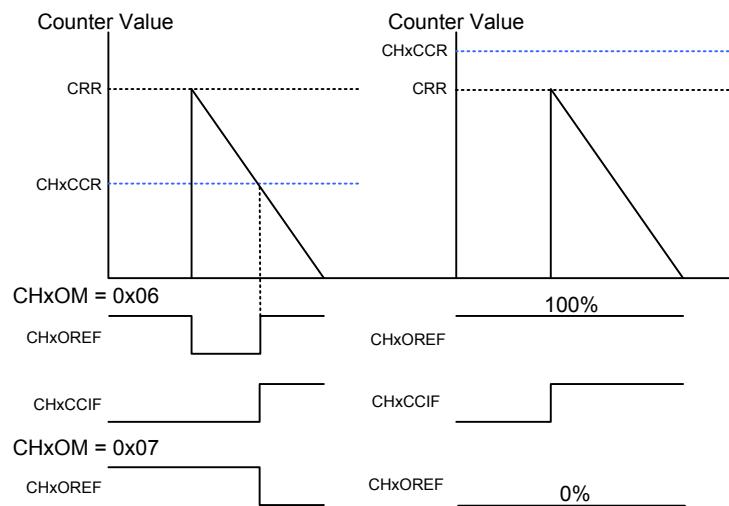
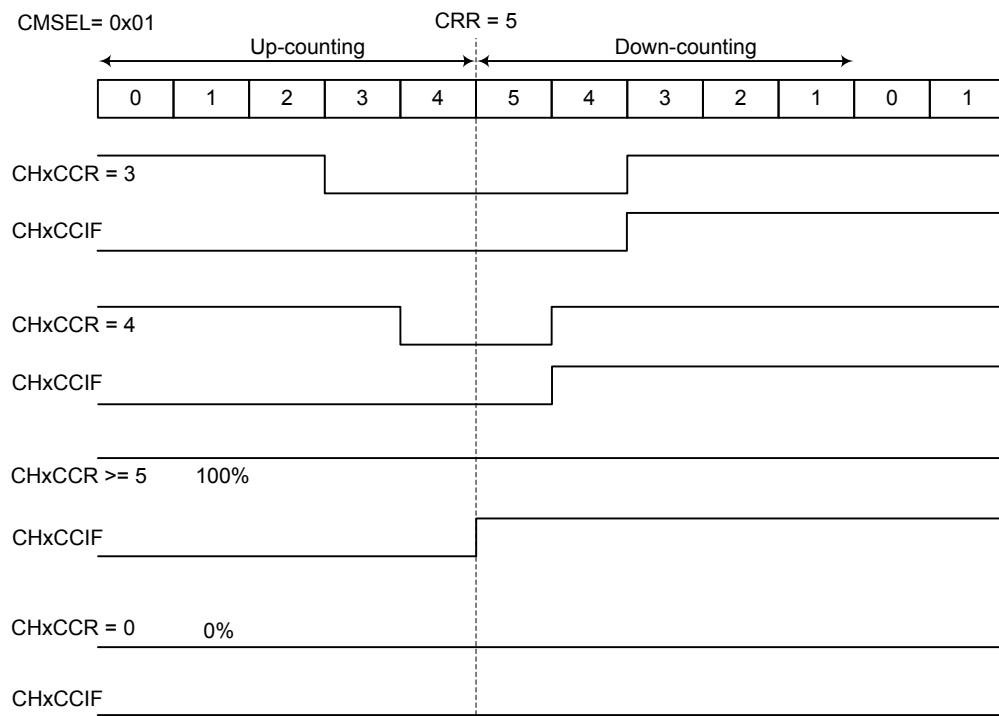


Figure 96. PWM Mode Channel Output Reference Signal and Counter in Up-counting Mode



**Figure 97. PWM Mode Channel Output Reference Signal and Counter in Down-counting Mode**

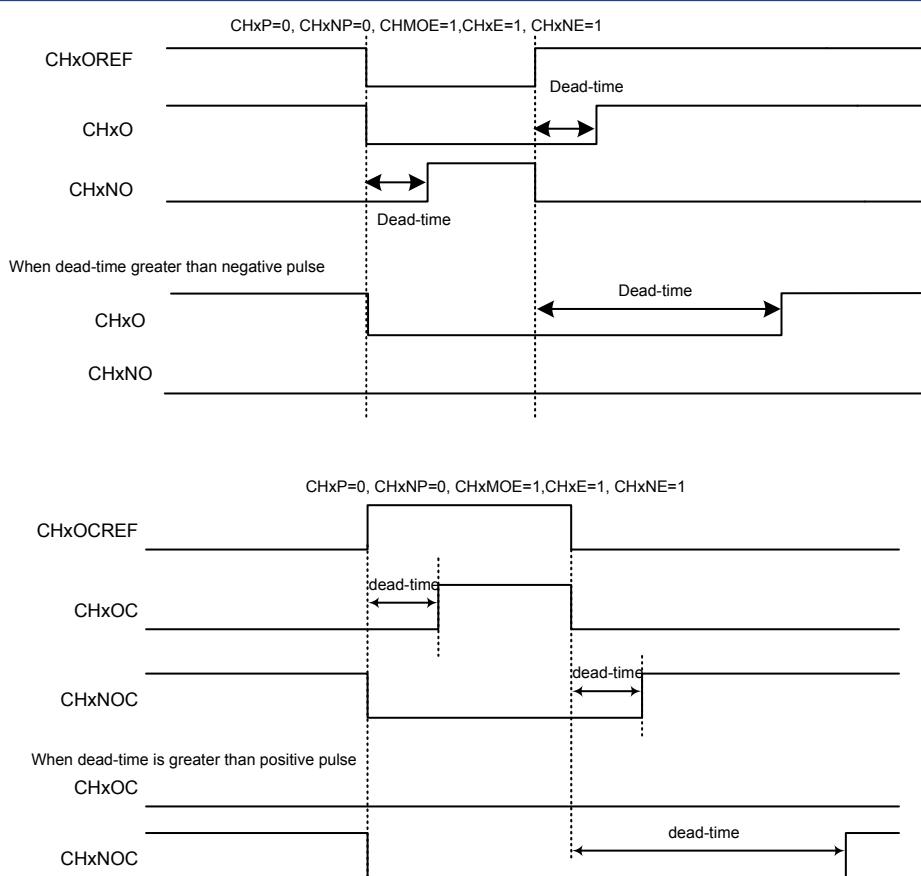


**Figure 98. PWM Mode 1 Channel Output Reference Signal and Counter in Centre-aligned Counting Mode**

### 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.

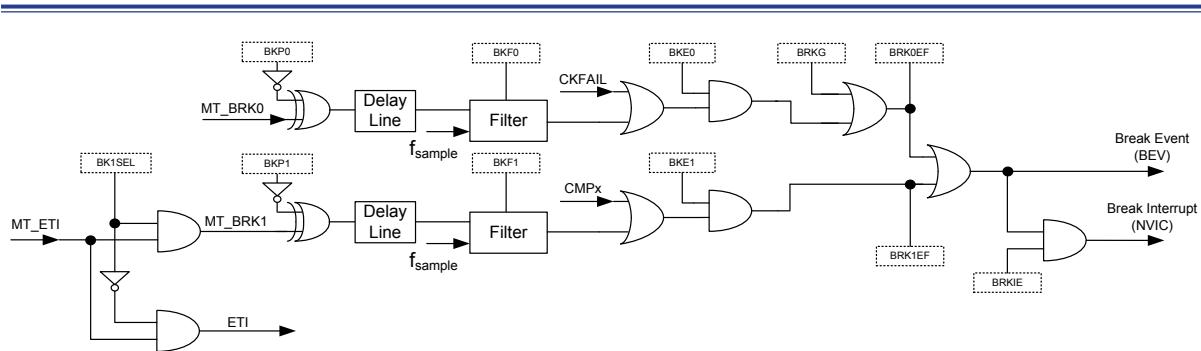


**Figure 99. Dead-time Insertion Performed for Complementary Outputs**

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 maximum two input signals for MCTM break. The MT\_BRK0 is default function and from external MTn\_BRK pin. But the second break signal MT\_BRK1 is share with the MT\_ETI pin and default is disabled. It can set the BRK1SEL bit in CHBRKCTR register to select the MT\_ETI pin for extra break signal MT\_BRK1. The detail block diagram is shown as below figure.



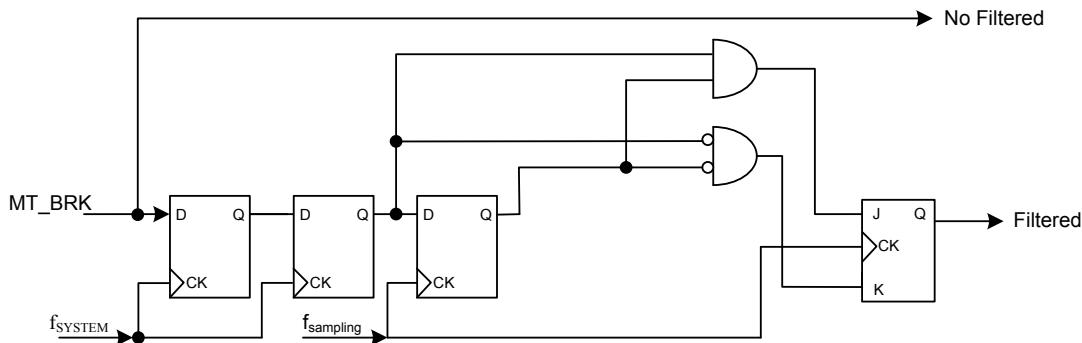
**Figure 100. Break Signal Block Diagram for MCTM**

When the MT\_BRK input has an active level, the Clock Monitor Circuitry detects a clock failure event or comparator transition, 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.

### Digital Filter (N=2)

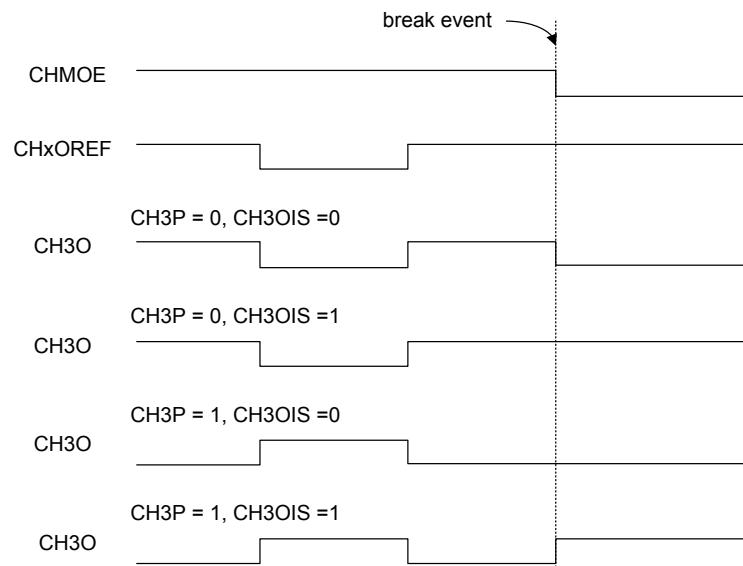


**Figure 101. MT\_BRK Pin Digital Filter Diagram with N = 2**

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:

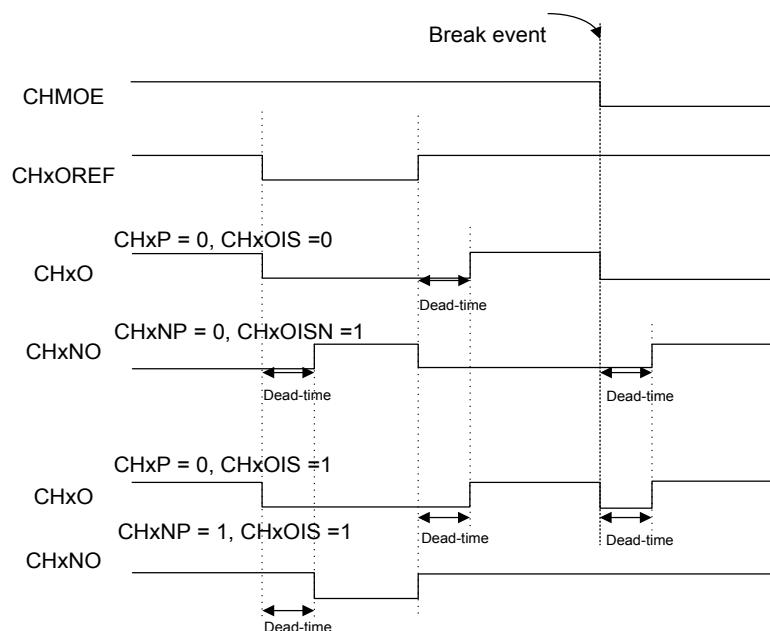
- If complementary outputs are used, the channel outputs a level signal first which can be selected to be either a disable or inactive level, selected by configuring the CHOSSI bit in the CHBRKCTR register. After the dead-time duration, the outputs will be changed to the idle state. The idle state is 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.



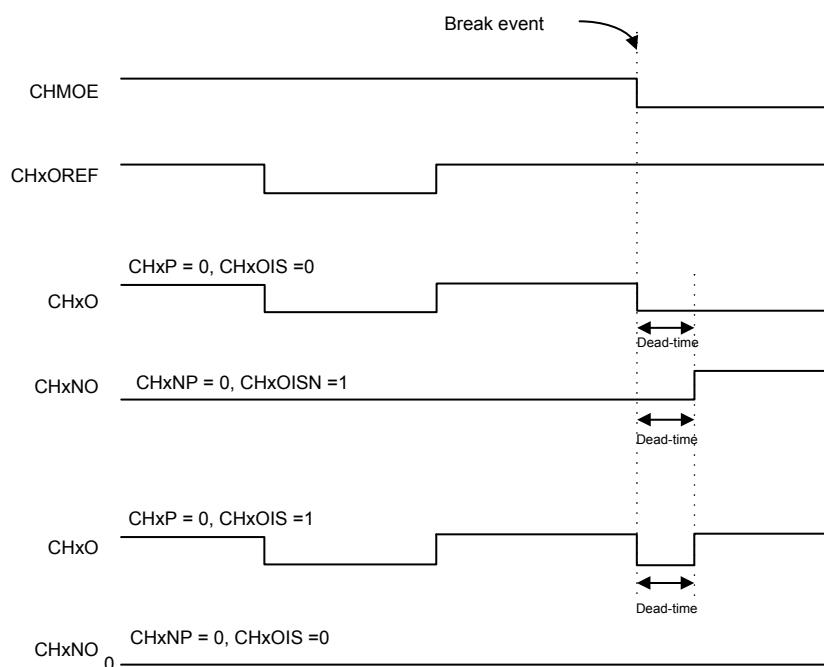
**Figure 102. Channel 3 Output with a Break Event Occurrence**

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

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

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

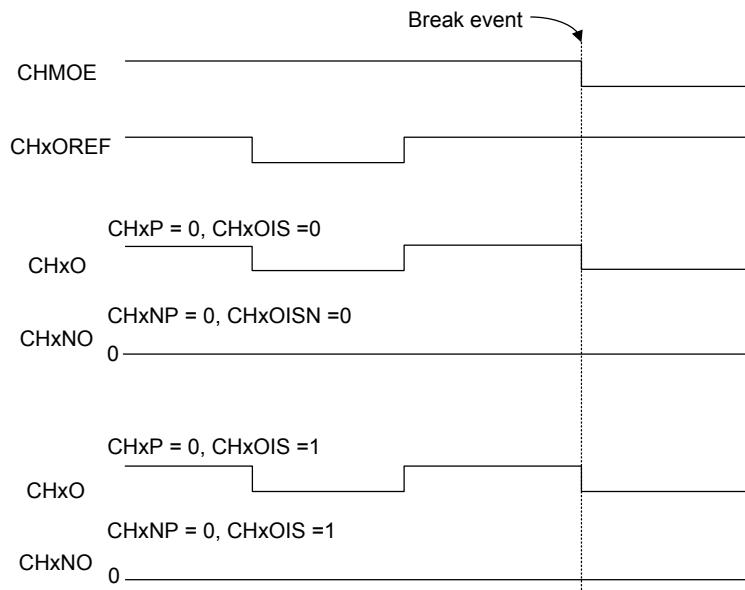


Figure 105. Hardware Protection When Both CHxO and CHxNO Are in Active Condition

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 35. 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	After dead-time: Output enabled MT_CHx = CHxOIS, MT_CHxN = CHxOISN MT_CHx_OEN =0, MT_CHxN_OEN = 0	
		1	0	1			
		1	1	0			
		1	1	1			

**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.

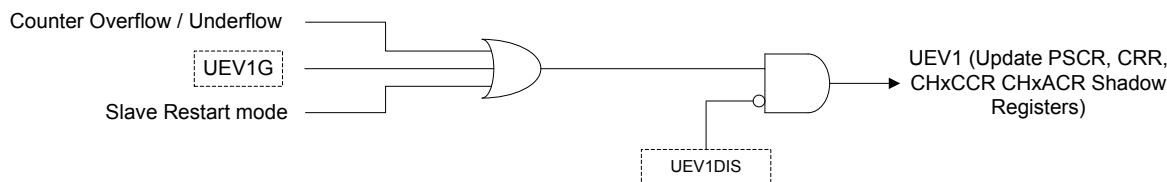
## Update Management

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.

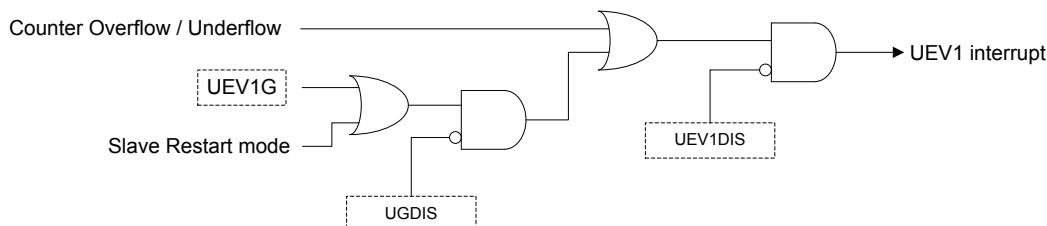
### Update Event 1

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



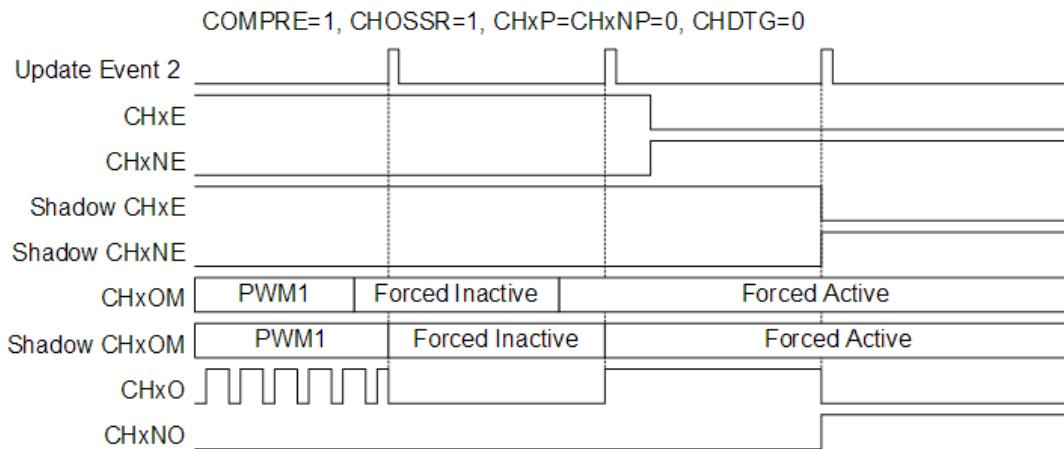
#### Update Event 1 Interrupt Management



**Figure 106. Update Event 1 Setup Diagram**

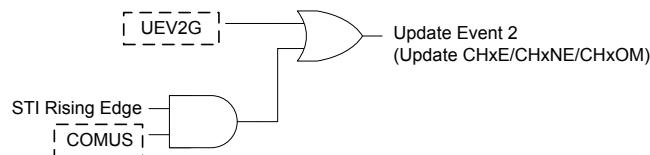
### Update Event 2

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, CHxOM bits will be updated when an update event 2 occurs.



**Figure 107. 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 108. Update Event 2 Setup Diagram**

## Quadrature Decoder

The Quadrature Decoder function uses two quadrantal inputs TI0 and TI1 derived from the MT\_CH0 and MT\_CH1 pins respectively which interact to generate the counter value. The DIR bit is modified by hardware automatically during each input source transition. The counter is counting on TI0 edges only, TI1 edges only or both TI0 and TI1 edges. The selection is 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. The application program must therefore configure the CRR register before the counter starts to count.

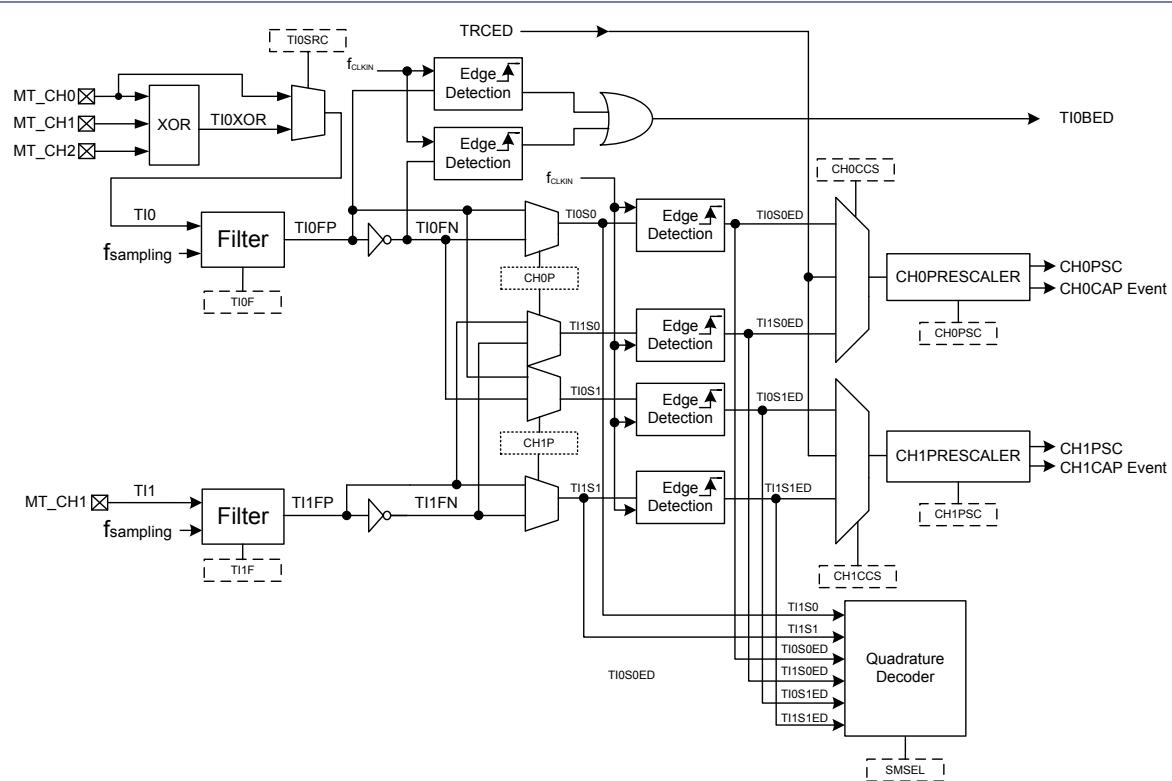
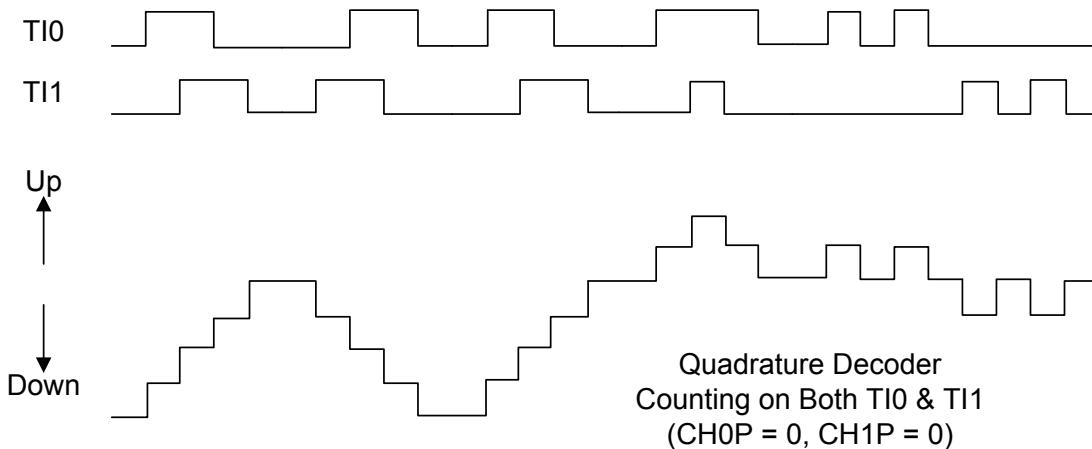


Figure 109. Input Stage and Quadrature Decoder Block Diagram

**Table 36. Counting Direction and Encoding Signals**

<b>Counting mode</b>	<b>Level</b>	<b>TI0S0</b>		<b>TI1S1</b>	
		<b>Rising</b>	<b>Falling</b>	<b>Rising</b>	<b>Falling</b>
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 110. Both TI0 and TI1 Quadrature Decoder Counting**

## Digital Filter

The digital filters are embedded in the input stage and clock controller block for the MT\_CH0 ~ MT\_CH3 and MT\_ETI 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.

Digital Filter (N=2)

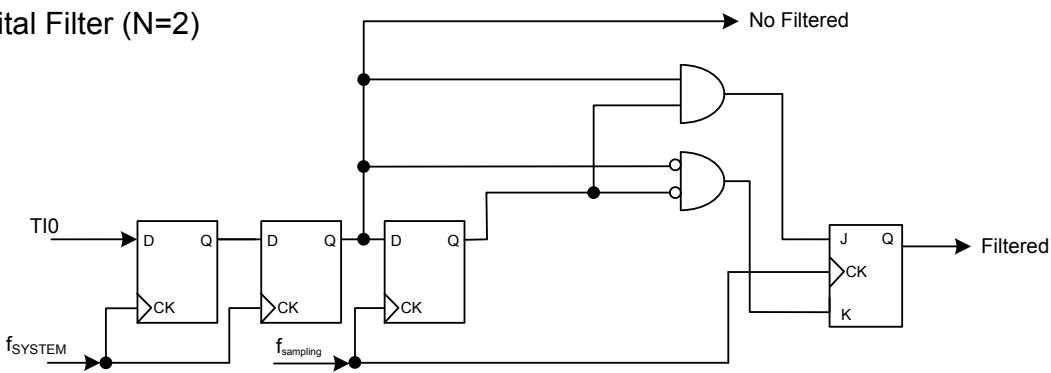


Figure 111. MT\_ETI Pin Digital Filter Diagram with N = 2

## Clearing CHxOREF when ETIF is high

The CHxOREF signal can be forced to 0 when the ETIF signal is set to a high level by setting the REFxCE bit to 1 in the CHxOCFR register. The CHxOREF signal will not return to its active level until the next update event 1 occurs.

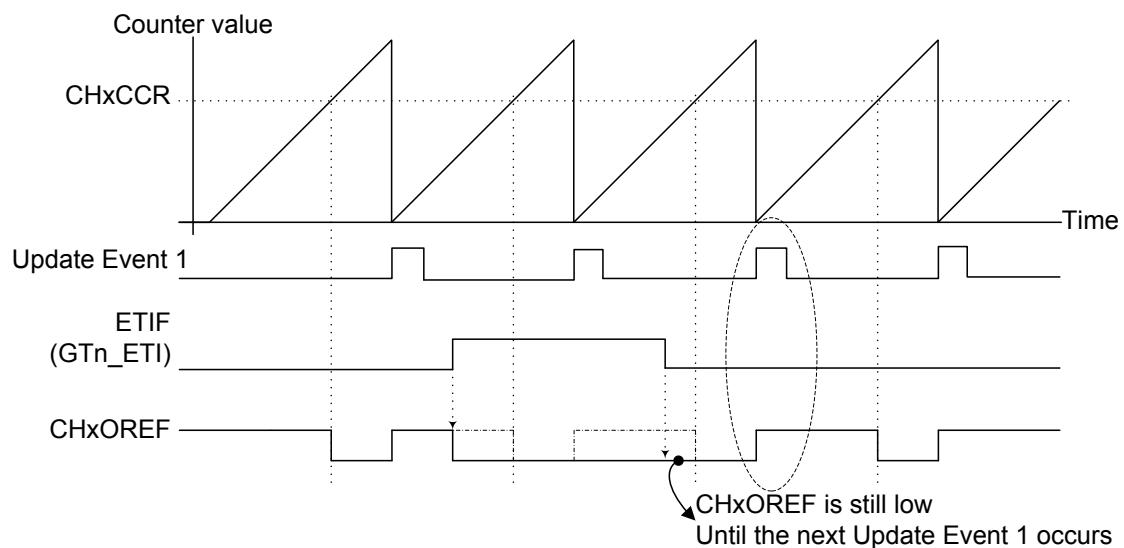
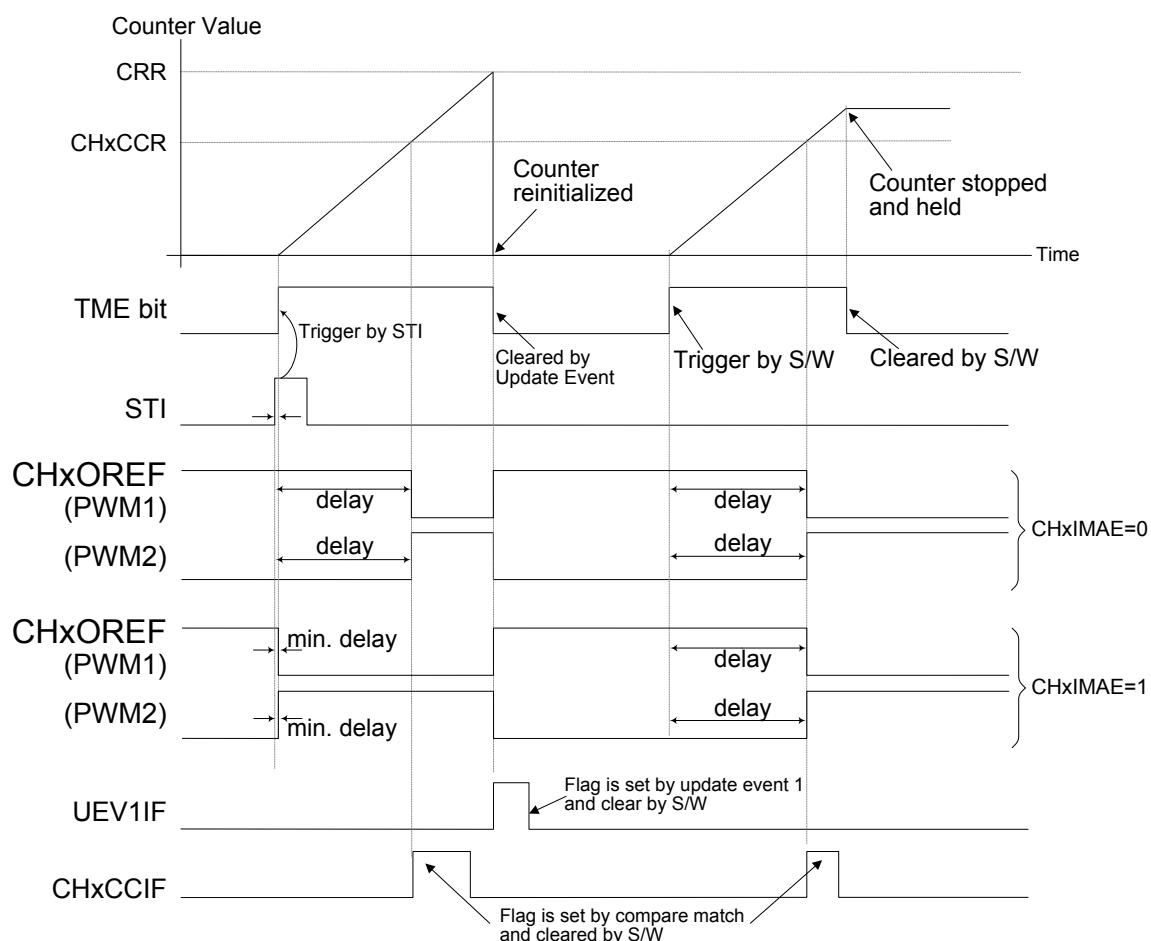


Figure 112. Clearing CHxOREF by ETIF

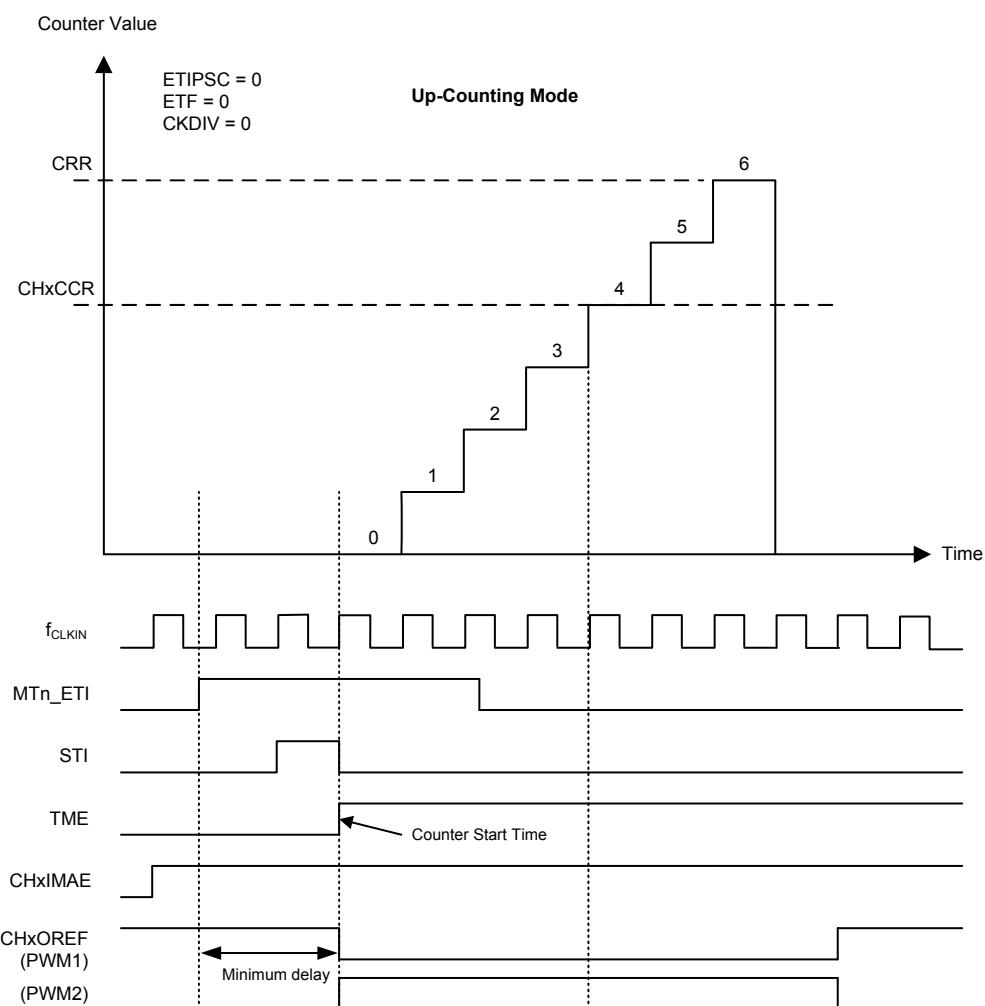
## Single Pulse Mode

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 113. 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.

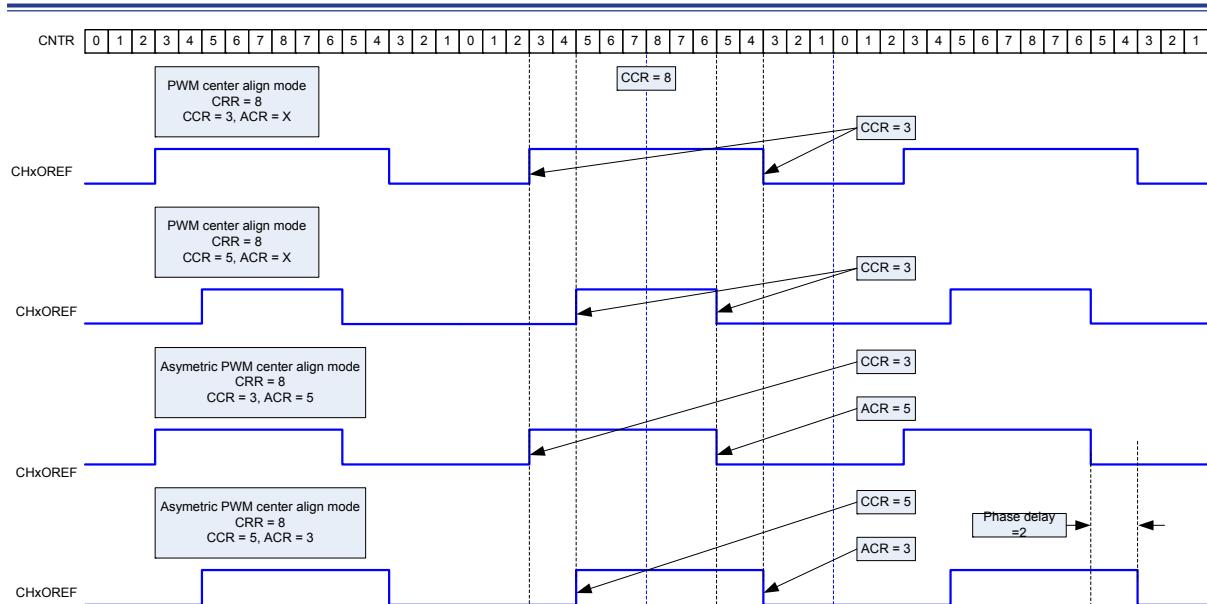


**Figure 114. Immediate Active Mode Minimum Delay**

## 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 register as up-count compare value and the counter is counting down, the PWM using the value in CHxACR register as down-count compare value. The Figure 41 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 115. Asymmetric PWM Mode versus Center-aligned Counting Mode**

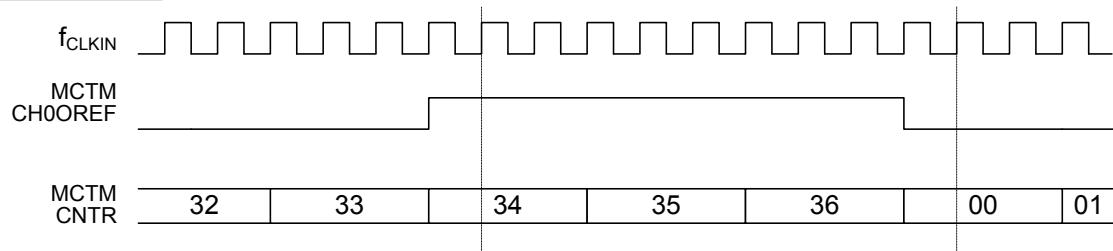
## Timer Interconnection

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.

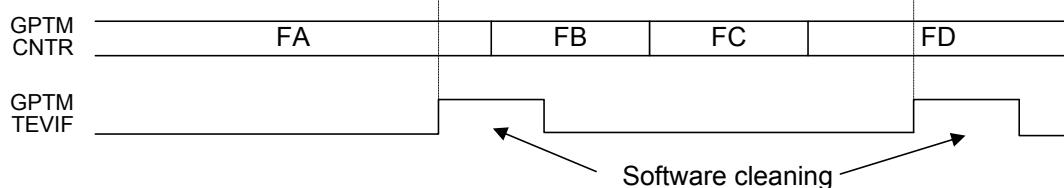
### Using one timer to trigger another timer to start or stop counting

- Configure MCTM0 to be in the master mode and to send its channel 0 Output Reference signal CH0OREF as a trigger output (MMSEL = 0x04).
- Configure the MCTM0 CH0OREF waveform.
- Configure the GPTM0 to receive its input trigger source from the MCTM0 trigger output (TRSEL = 0x0A).
- Configure GPTM0 to operate in the pause mode (SMSEL = 0x05).
- Enable GPTM0 by writing ‘1’ to the TME bit.
- Enable MCTM0 by writing ‘1’ to the TME bit.

#### Master MCTM0



#### Slave GPTM0



**Figure 116. Pausing GPTM0 using the MCTM0 CH0OREF Signal**

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

- Configure MCTM0 to operate in the master mode and to send its Update Event UEV as the trigger output (MMSEL = 0x02).
- Configure the MCTM0 period by setting the CHxCRR register.
- Configure GPTM0 to get the input trigger source from the MCTM0 trigger output (TRSEL = 0x0A).
- Configure GPTM0 to be in the slave trigger mode (SMSEL = 0x06).
- Start MCTM0 by writing ‘1’ to the TME bit.

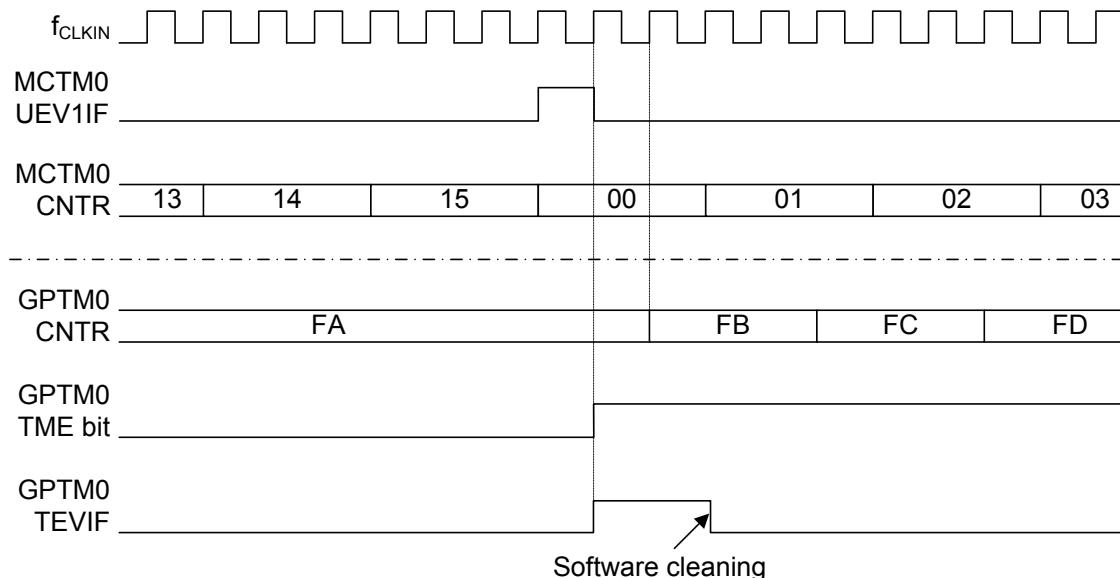
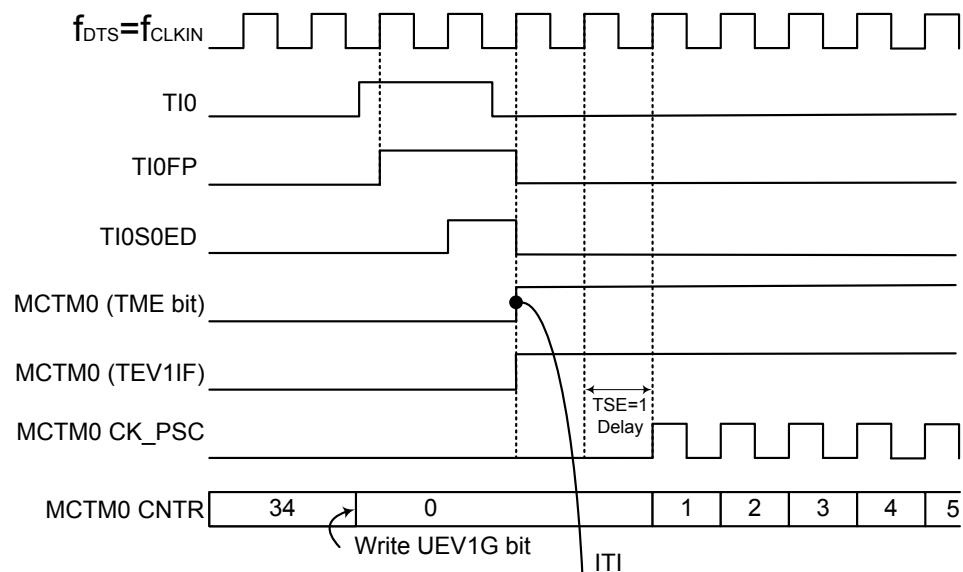


Figure 117. Triggering GPTM0 with MCTM0 Update Event 1

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

- Configure MCTM0 to operate in the master mode to send its enable signal as a trigger output (MMSEL = 0x01).
- Configure MCTM0 slave mode to receive its input trigger source from MT\_CH0 pin (TRSEL = 0x01).
- Configure MCTM0 to be in the slave trigger mode (SMSEL = 0x06).
- Enable the MCTM0 master timer synchronisation function by setting the TSE bit in the MDCFR register to 1 to synchronise the slave timer.
- Configure GPTM0 to receive its input trigger source from the MCTM0 trigger output (TRSEL = 0x0A).
- Configure GPTM0 to be in the slave trigger mode (SMSEL = 0x06).

### Master MCTM0



### Slave GPTM0

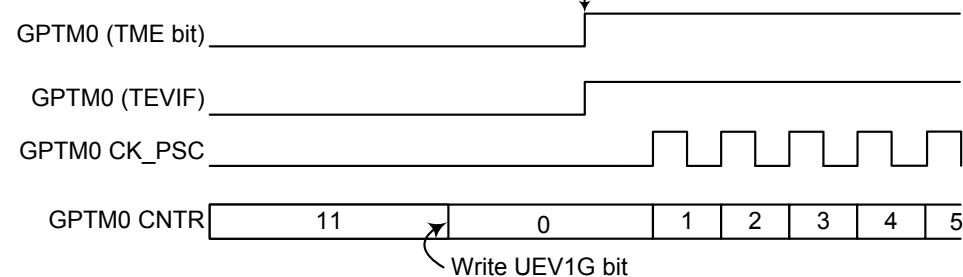


Figure 118. Trigger MCTM0 and GPTM0 with the MCTM0 CH0 Input

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

#### GPTM0:

- 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 GPTM0 (TME=1)

#### MCTM0:

- Select GPTM0 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

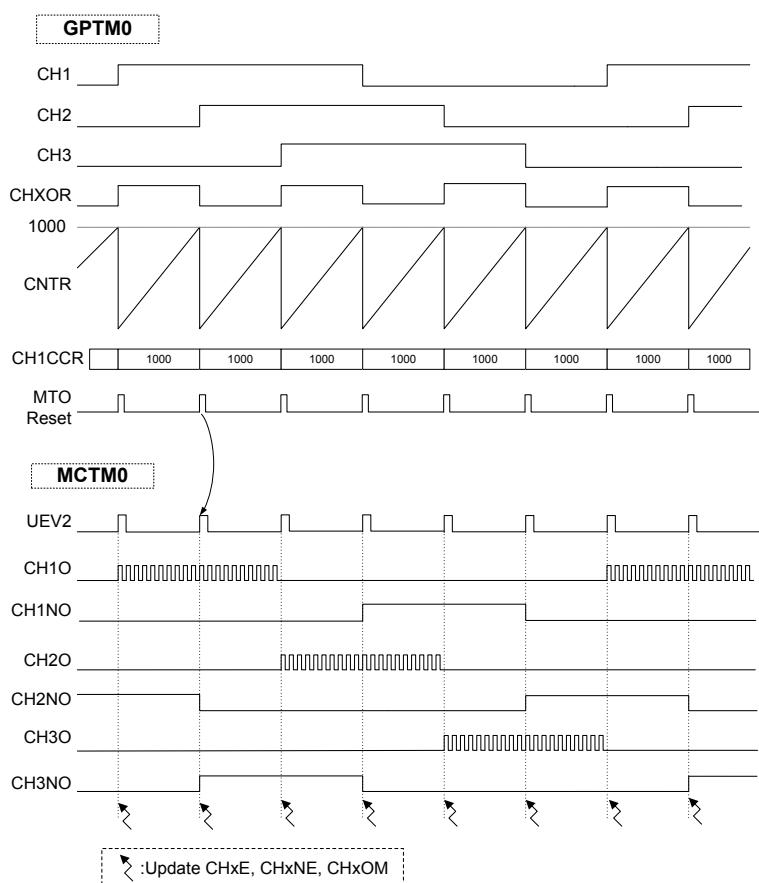


Figure 119. CH1XOR Input as Hall Sensor Interface

## 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.

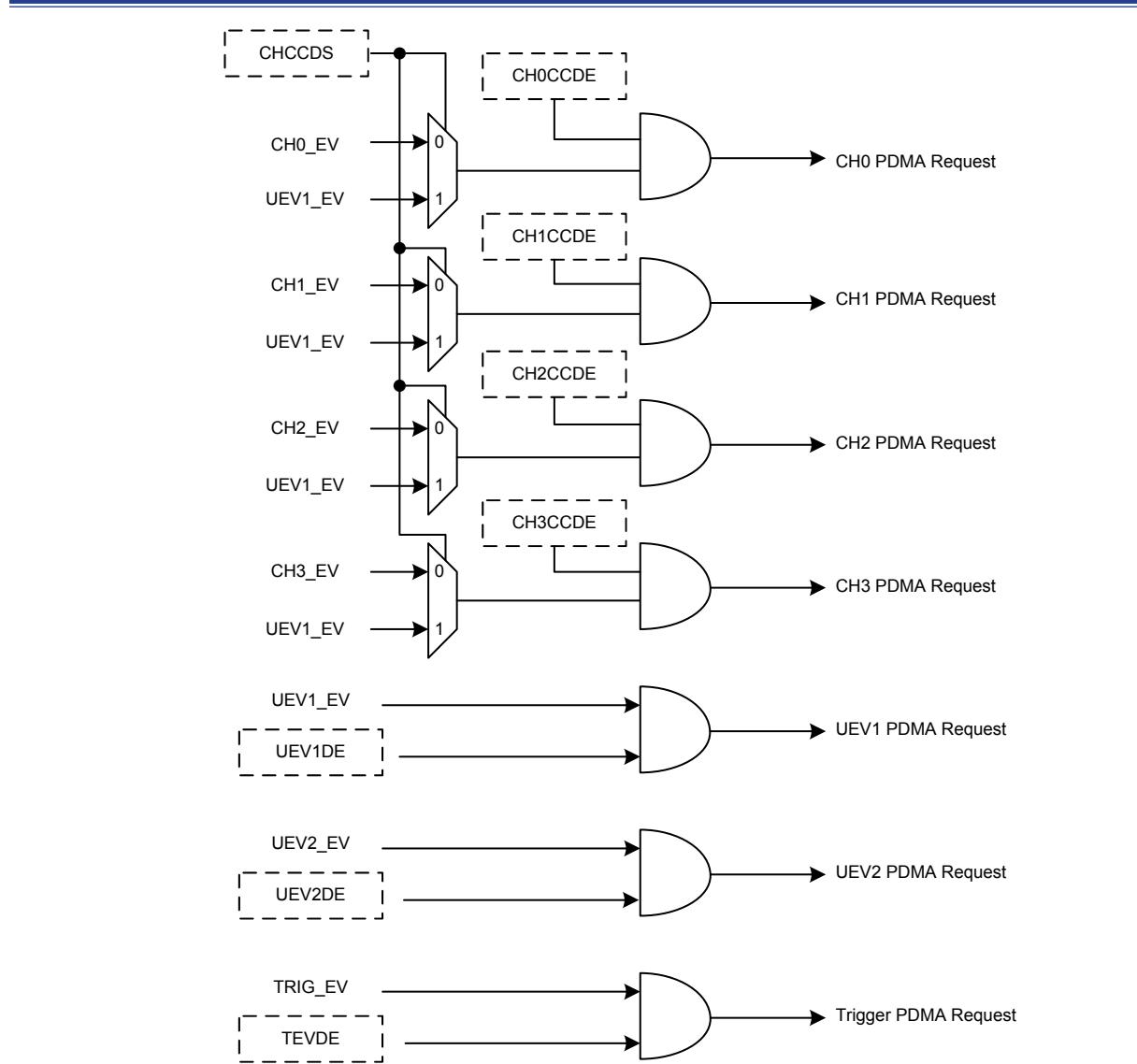
**Table 37. Lock Level Table**

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				

- (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 safty.
- (3) When the MCTMEN and CKMEN control bits of the CKCU lock protection mode is enabled in the MCTM unit, the bits will be allowed to enable only and inhibited to disable again.

## PDMA Request

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



**Figure 120. MCTM PDMA Mapping Diagram**

## Register Map

The following table shows the MCTM registers and reset values.

**Table 38. MCTM Register Map**

Register	Offset	Description	Reset Value
<b>MCTM0 Base Address = 0x4002_C000</b>			
<b>MCTM1 Base Address = 0x4002_D000</b>			
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_000A
DICTR	0x074	Timer PDMA/Interrupt Control Register	0x0000_0000
EVGR	0x078	Timer Event Generator Register	0x0000_0000
INTSR	0x07C	Timer Interrupt Status Register	0x0000_0000
CNTR	0x080	Timer Counter Register	0x0000_0000
PSCR	0x084	Timer Prescaler Register	0x0000_0000
CRR	0x088	Timer Counter Reload Register	0x0000_FFFF
REPR	0x08C	Timer Repetition Register	0x0000_0000
CH0CCR	0x090	Channel 0 Capture/Compare Register	0x0000_0000
CH1CCR	0x094	Channel 1 Capture/Compare Register	0x0000_0000
CH2CCR	0x098	Channel 2 Capture/Compare Register	0x0000_0000
CH3CCR	0x09C	Channel 3 Capture/Compare Register	0x0000_0000
CH0ACR	0x0A0	Channel 0 Asymmetric Compare Register	0x0000_0000
CH1ACR	0x0A4	Channel 1 Asymmetric Compare Register	0x0000_0000
CH2ACR	0x0A8	Channel 2 Asymmetric Compare Register	0x0000_0000
CH3ACR	0x0AC	Channel 3 Asymmetric Compare Register	0x0000_0000

## Register Descriptions

### Timer Counter Configuration Register – CNTCFR

This register specifies the MCTM counter configuration.

Offset: 0x000

Reset value: 0x0000\_0000

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							

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	Update event 1 interrupt generation disable control 0: Any of the following events will generate an update PDMA request or interrupt - Counter overflow/underflow - Setting the UEV1G bit - Update generation through the slave mode 1: Only counter overflow/underflow generates an update PDMA request or interrupt
[0]	UEV1DIS	Update event 1 Disable control 0: Enable the update event 1 request by one of following events - Counter overflow/underflow - Setting the UEV1G bit - Update generation through the slave mode 1: Disable the update event 1 - however the counter and the prescaler are reinitialised if the UEV1G bit is set or if a hardware restart is received from the slave mode

## Timer Mode Configuration Register – MDCFR

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

Offset: 0x004

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved					SPMSET		
						RW	0	
	23	22	21	20	19	18	17	16
Type/Reset	Reserved					MMSEL		
						RW	0	
	15	14	13	12	11	10	9	8
Type/Reset	Reserved					RW	0	
						RW	0	
	7	6	5	4	3	2	1	0
Type/Reset	Reserved					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.	
<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 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	Quadrature Decoder mode 1	The counter uses the clock pulses generated from the interaction between the TI0 and TI1 signals to drive the counter prescaler. A transition of the TI0 edge is used in this mode depending upon the TI1 level.
		010	Quadrature Decoder mode 2	The counter uses the clock pulse generated from the interaction between the TI0 and TI1 signals to drive the counter prescaler. A transition of the TI1 edge is used in this mode depending upon the TI0 level.
		011	Quadrature Decoder mode 3	The counter uses the clock pulse generated from the interaction between the TI0 and TI1 signals to drive the counter prescaler. A transition of one channel edge is used in the quadrature decoder mode 3 depending upon the other channel level.
		100	Restart Mode	The counter value restarts from 0 or the CRR shadow register value depending upon the counter mode on the rising edge of the STI signal. The registers will also be updated.
		101	Pause Mode	The counter starts to count when the selected trigger input STI is high. The counter stops counting on the instant, not being reset, when the STI signal changes its state to a low level. Both the counter start and stop control are determined by the STI signal.
		110	Trigger Mode	The counter starts to count from the original value in the counter on the rising edge of the selected trigger input STI. Only the start of counter is controlled.
		111	STIED	The rising edge of the selected trigger signal STI will clock the counter.
[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 MCTM external clock setting and the trigger source selection.

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

Bits	Field	Descriptions
[24]	ECME	External Clock Mode Enable 0: External clock mode is disabled. 1: External clock mode is enabled. The following two setting have the same effect: 1. Setting the ECME bit to 1 2. Setting SMSEL=0x111 with STI connected to ETIF (TRSEL=0x011)
[16]	ETIPOL	External Trigger Polarity 0: MTn_ETI active at high level or rising edge 1: MTn_ETI active at low level or falling edge.
[13:12]	ETIPSC	External Trigger Prescaler A prescaler can be enabled to reduce the ETIP frequency 00: Prescaler OFF 01: ETIP frequency divided by 2 10: ETIP frequency divided by 4 11: ETIP frequency divided by 8

Bits	Field	Descriptions
[11:8]	ETF	<p>External Trigger Filter</p> <p>These bits define the frequency divided ratio that is used to sample the MTn_ETI signal. The digital filter in the MCTM is an N-event counter where N means 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_{DTS}</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>
[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 – T10S0</li> <li>0010: Channel 1 filtered input – T11S1</li> <li>0011: External Trigger input – ETIF</li> <li>1000: Channel 0 Edge Detector – T10BED</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 39. MCTM Internal Trigger Connection**

Slave Timing Module	ITI0	ITI1	ITI2
MCTM0	MCTM1	GPTM0	GPTM1
MCTM1	MCTM0	GPTM0	GPTM1

## Timer Counter Register – CTR

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

Offset: 0x010

Reset value: 0x0000\_0000

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									
Type/Reset	RW	0	RW	0	RW	0	CRBE	TME		

Bits	Field	Descriptions
[16]	CHCCDS	Channel Capture/Compare PDMA Selection 0: Channel PDMA request derived from the channel capture/compare event. 1: Channel PDMA request derived from the update event 1.
[9]	COMUS	Capture/Compare Control Update Selection 0: Updated by setting the UEV2G bit only 1: Updated by setting the UEV2G bit or when a STI signal rising edge occurs This bit is only available when the capture/compare preload function is enabled by setting the 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			TIOF		
					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 TIO 1: The XOR operation output of the MT_CH0, MT_CH1, and MT_CH2 pins are connected to the channel 0 input TIO
[19:18]	CH0PSC	Channel 0 Capture Input Source Prescaler Setting These bits define the effective events of the channel 0 capture input. Note that the prescaler is reset once the Channel 0 Capture/Compare Enable bit, CH0E, in the Channel Control register named CHCTR is cleared to 0. 00: No prescaler, channel 0 capture input signal is chosen for each active event 01: Channel 0 Capture input signal is chosen for every 2 events 10: Channel 0 Capture input signal is chosen for every 4 events 11: Channel 0 Capture input signal is chosen for every 8 events
[17:16]	CH0CCS	Channel 0 Capture/Compare Selection 00: Channel 0 is configured as an output 01: Channel 0 is configured as an input derived from the TIO signal 10: Channel 0 is configured as an input derived from the TI1 signal 11: Channel 0 is configured as an input which comes from the TRCED signal derived from the Trigger Controller Note: The CH0CCS field can be accessed only when the CH0E bit is cleared to 0.

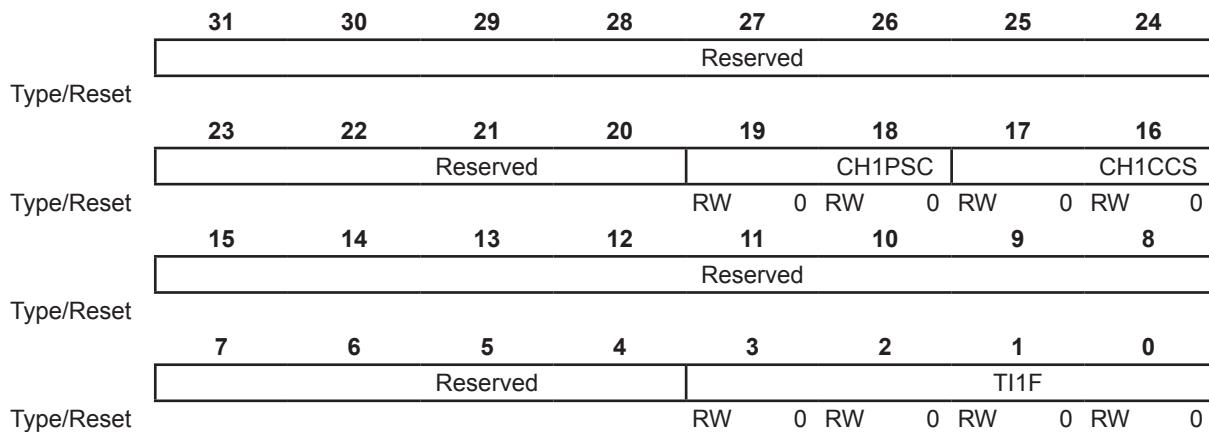
Bits	Field	Descriptions
[3:0]	TI0F	<p>Channel 0 Input Source TI0 Filter Setting</p> <p>These bits define the frequency divided ratio used to sample the TI0 signal. The Digital filter in the MCTM is an N-event counter where N is defined as how many valid transitions are necessary to output a filtered signal.</p> <ul style="list-style-type: none"> <li>0000: No filter, the sampling clock is <math>f_{DTS}</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>

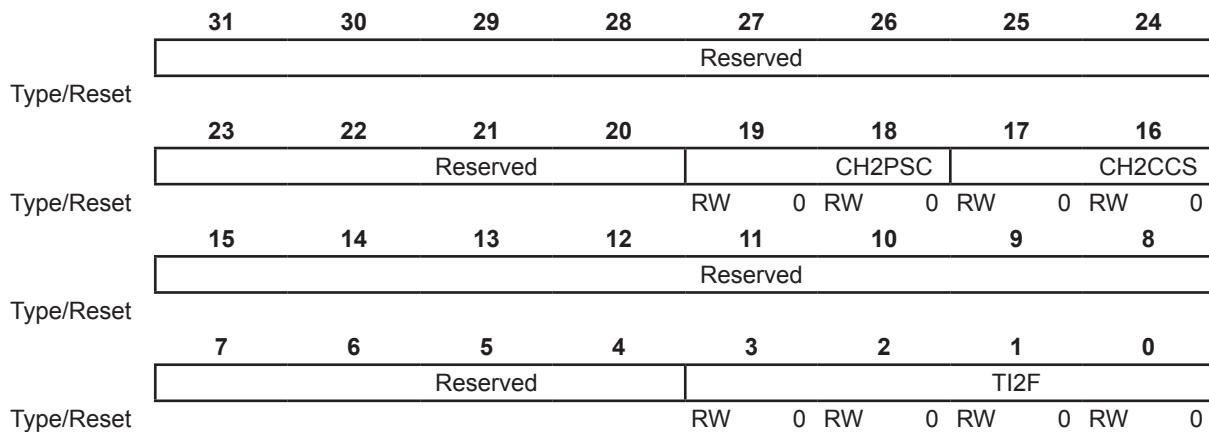
Bits	Field	Descriptions
[3:0]	TI1F	<p>Channel 1 Input Source TI1 Filter Setting</p> <p>These bits define the frequency divide ratio used to sample the TI1 signal. The Digital filter in the MCTM is an N-event counter where N is defined as how many valid transitions are necessary to output a filtered signal.</p> <ul style="list-style-type: none"> <li>0000: No filter, the sampling clock is <math>f_{DTS}</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>

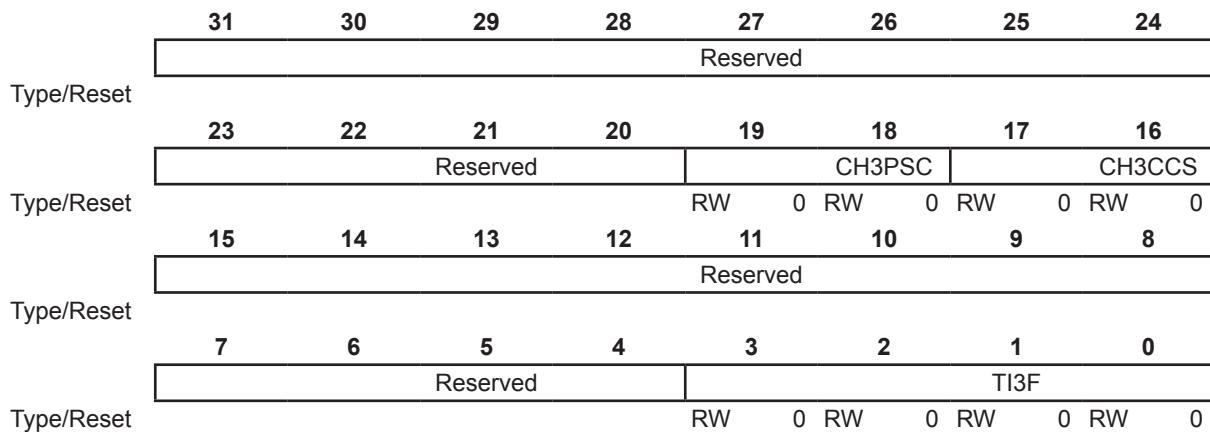
Bits	Field	Descriptions
[3:0]	TI2F	<p>Channel 2 Input Source TI2 Filter Setting</p> <p>These bits define the frequency divide ratio used to sample the TI2 signal. The Digital filter in the MCTM is an N-event counter where N is defined as how many valid transitions are necessary to output a filtered signal.</p> <ul style="list-style-type: none"> <li>0000: No filter, the sampling clock is <math>f_{DTS}</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_{DTS}</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

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								CH0OM[3]	
Type/Reset	7	6	5	4	3	2	1	0		
	Reserved	CH0IMAE	CH0PRE	REF0CE	CH0OM[2:0]					
Type/Reset		RW	0	RW	0	RW	0	RW	0	

Bits	Field	Descriptions
[5]	CH0IMAE	<p>Channel 0 Immediate Active Enable</p> <p>0: No action</p> <p>1: Single pulse Immediate Active Mode is enabled</p> <p>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 channel 0 is configured operate in PWM mode 1 or PWM mode 2.</p>
[4]	CH0PRE	<p>Channel 0 Capture/Compare Register (CH0CCR) Preload Enable</p> <p>0: CH0CCR preload function is disabled</p> <p>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</p> <p>The new CH0CCR value will not be transferred to its shadow register until an update event 1 occurs.</p>
[3]	REF0CE	<p>Channel 0 Reference Output Clear Enable</p> <p>0: CH0OREF operates normally and is not affected by the ETIF signal</p> <p>1: CH0OREF is forced to 0 on the high level of the ETIF signal derived from the MTn_ETI pin</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

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							CH1OM[3]	
Type/Reset	7	6	5	4	3	2	1	0	
	Reserved	RW	0	RW	0	RW	0	RW	
Type/Reset		CH1IMAE	CH1PRE	REF1CE		CH1OM[2:0]			
		RW	0	RW	0	RW	0	RW	

Bits	Field	Descriptions
[5]	CH1IMAE	<p>Channel 1 Immediate Active Enable</p> <p>0: No action</p> <p>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.</p> <p>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</p> <p>The new CH1CCR value will not be transferred to its shadow register until an update event 1 occurs.</p>
[3]	REF1CE	<p>Channel 1 Reference Output Clear Enable</p> <p>0: CH1OREF performed normally and is not affected by the ETIF signal</p> <p>1: CH1OREF is forced to 0 on the high level of the ETIF signal derived from the MTn_ETI pin.</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

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	Reserved								CH2OM[3]
	7	6	5	4	3	2	1	0	
Type/Reset	Reserved	CH2IMAE	CH2PRE	REF2CE	CH2OM[2:0]				RW 0
		RW	0 RW	0 RW	0 RW	0 RW	0 RW	0 RW	0

Bits	Field	Descriptions
[5]	CH2IMAE	<p>Channel 2 Immediate Active Enable</p> <p>0: No action</p> <p>1: Single pulse Immediate Active Mode enabled</p> <p>The CH2OREF will be forced to the compare matched level immediately after an available trigger event occurs irrespective of the result of the comparison between the CNTR and the CH2CCR values.</p> <p>The effective duration ends automatically at the next overflow or underflow event.</p> <p>Note: The CH2IMAE bit is available only if the channel 2 is configured to be operated in PWM mode 1 or PWM mode 2.</p>
[4]	CH2PRE	<p>Channel 2 Capture/Compare Register (CH2CCR) Preload Enable</p> <p>0: CH2CCR preload function is disabled.</p> <p>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</p> <p>The new CH2CCR value will not be transferred to its shadow register until an update event 1 occurs.</p>
[3]	CH3OCCE	<p>Channel 2 Reference Output Clear Enable</p> <p>0: CH2OREF operates normally and is not affected by the ETIF signal</p> <p>1: CH2OREF is forced to 0 during a high level of the ETIF signal derived from the MTn_ETI pin</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

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							CH3OM[3]	
Type/Reset	7	6	5	4	3	2	1	0	
	Reserved	RW	0	RW	0	RW	0	RW	
Type/Reset		CH3IMAE	CH3PRE	REF3CE	CH3OM[2:0]				
					RW	0	RW	0	

Bits	Field	Descriptions
[5]	CH3IMAE	<p>Channel 3 Immediate Active Enable</p> <p>0: No action</p> <p>1: Single pulse Immediate Active Mode enabled</p> <p>The CH3OREF will be forced to the compare matched level immediately after an available trigger event occurs irrespective of the result of the comparison between the CNTR and the CH3CCR values.</p> <p>The effective duration ends automatically at the next overflow or underflow event.</p> <p>Note: The CH3IMAE bit is available only if channel 3 is configured to be operated in PWM mode 1 or PWM mode 2.</p>
[4]	CH3PRE	<p>Channel 3 Capture/Compare Register (CH3CCR) Preload Enable</p> <p>0: CH3CCR preload function is disabled.</p> <p>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</p> <p>The new CH3CCR value will not be transferred to its shadow register until an update event 1 occurs.</p>
[3]	REF3CE	<p>Channel 3 Reference Output Clear Enable</p> <p>0: CH3OREF operates normally and is not affected by the ETIF signal</p> <p>1: CH3OREF is forced to 0 during the high level of the ETIF signal derived from the MTn_ETI pin</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>

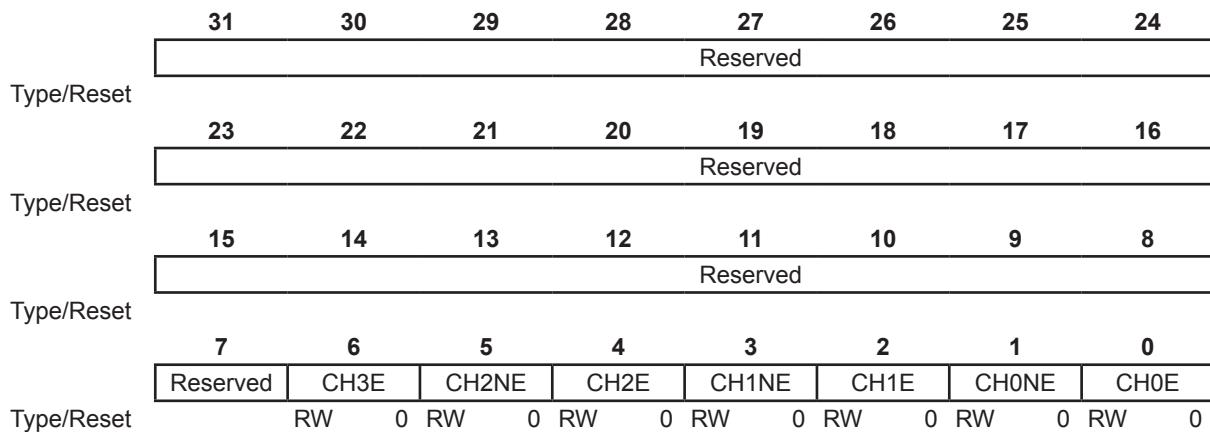
Note: When channel 3 is used as asymmetric PWM output mode, the Counter Mode Selection bit in Counter Configuration Register must be configured as Center-aligned Counting mode (CMSEL = 01/02/03)

## 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	<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>

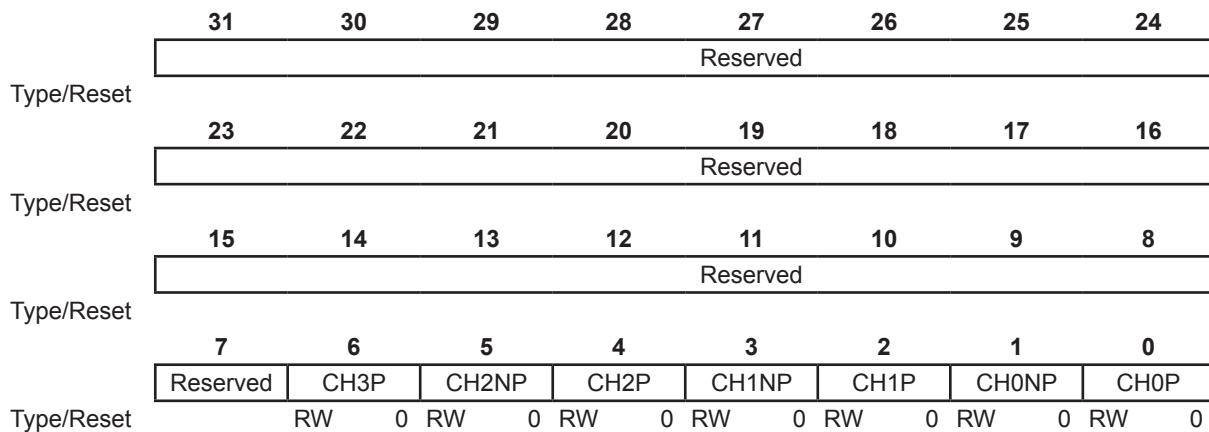
Bits	Field	Descriptions
[3]	CH1NE	Channel 1 Capture/Compare Complementary Enable 0: Off – Channel 1 complementary output CH1NO is not active. The CH1NO level is then determined by the condition of the CHMOE, CHOSSI, CHOSSR, CH1OIS, CH1OISN and CH1E bits. 1: On – Channel 1 complementary output CH1NO is generated on the corresponding output pin determined by the condition of the CHMOE, CHOSSI, CHOSSR, CH1OIS, CH1OISN and CH1E bits.
[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. The CH1O level is then determined by the condition of the CHMOE, CHOSSI, CHOSSR, CH1OIS, CH1OISN and CH1NE bits. 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.
[1]	CH0NE	Channel 0 Capture/Compare Complementary Enable 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. 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.
[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. The CH0O level is then determined by the condition of the CHMOE, CHOSSI, CHOSSR, CH0OIS, CH0OISN and CH0NE bits. 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.

## 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	0

Bits	Field	Descriptions
[6]	CH3OIS	MTx_CH3O Output Idle State 0: Channel 3 output CH3O = 0 when CHMOE = 0 1: Channel 3 output CH3O = 1 when CHMOE = 0
[5]	CH2OISN	MTx_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	MTx_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	MTx_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	MTx_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	MTx_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	MTx_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\_000A

	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
	Reserved	CHOSSR	CHOSSI	GFSEL1	GFSEL0			LOCKLV
RW								
Type/Reset	15	14	13	12	11	10	9	8
BKF1								
Type/Reset	RW	0	RW	0	RW	0	RW	0
Reserved								
Type/Reset	7	6	5	4	3	2	1	0
	BK1SEL	CHAOE	CHMOE	BKP1	BKE1	BKP0	BKE0	
Type/Reset	RW	0	RW	0	RW	1	RW	1

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.
[19]	GFSEL1	Deglitch Filter Selction for Break 1 0: no input deglitch filter 1: 50ns deglitch filter
[18]	GFSEL0	Deglitch Filter Selction for Break 0 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
[15:12]	BKF1	<p>Break 1 Input Filter Setting</p> <p>These bits define the frequency ratio used to sample the MT_BRK1 signal. The digital filter in the MCTM is an N-event counter where N is defined as how many valid transitions are necessary to output a filtered signal.</p> <p>0000: No filter – don't need sample clock.</p> <p>0001: <math>f_{SAMPLING} = f_{CLKIN}</math>, N = 2.      0010: <math>f_{SAMPLING} = f_{CLKIN}</math>, N = 4.      0011: <math>f_{SAMPLING} = f_{CLKIN}</math>, N = 8.      0100: <math>f_{SAMPLING} = f_{DTS}/2</math>, N = 6.      0101: <math>f_{SAMPLING} = f_{DTS}/2</math>, N = 8.      0110: <math>f_{SAMPLING} = f_{DTS}/4</math>, N = 6.      0111: <math>f_{SAMPLING} = f_{DTS}/4</math>, N = 8.      1000: <math>f_{SAMPLING} = f_{DTS}/8</math>, N = 6.      1001: <math>f_{SAMPLING} = f_{DTS}/8</math>, N = 8.      1010: <math>f_{SAMPLING} = f_{DTS}/16</math>, N = 5.      1011: <math>f_{SAMPLING} = f_{DTS}/16</math>, N = 6.      1100: <math>f_{SAMPLING} = f_{DTS}/16</math>, N = 8.      1101: <math>f_{SAMPLING} = f_{DTS}/32</math>, N = 5.      1110: <math>f_{SAMPLING} = f_{DTS}/32</math>, N = 6.      1111: <math>f_{SAMPLING} = f_{DTS}/32</math>, N = 8.</p>
[11:8]	BKF0	<p>Break 0 Input Filter Setting</p> <p>These bits define the frequency ratio used to sample the MT_BRK0 signal. The digital filter in the MCTM is an N-event counter where N is defined as how many valid transitions are necessary to output a filtered signal.</p> <p>0000: No filter – don't need sample clock.</p> <p>0001: <math>f_{SAMPLING} = f_{CLKIN}</math>, N = 2.      0010: <math>f_{SAMPLING} = f_{CLKIN}</math>, N = 4.      0011: <math>f_{SAMPLING} = f_{CLKIN}</math>, N = 8.      0100: <math>f_{SAMPLING} = f_{DTS}/2</math>, N = 6.      0101: <math>f_{SAMPLING} = f_{DTS}/2</math>, N = 8.      0110: <math>f_{SAMPLING} = f_{DTS}/4</math>, N = 6.      0111: <math>f_{SAMPLING} = f_{DTS}/4</math>, N = 8.      1000: <math>f_{SAMPLING} = f_{DTS}/8</math>, N = 6.      1001: <math>f_{SAMPLING} = f_{DTS}/8</math>, N = 8.      1010: <math>f_{SAMPLING} = f_{DTS}/16</math>, N = 5.      1011: <math>f_{SAMPLING} = f_{DTS}/16</math>, N = 6.      1100: <math>f_{SAMPLING} = f_{DTS}/16</math>, N = 8.      1101: <math>f_{SAMPLING} = f_{DTS}/32</math>, N = 5.      1110: <math>f_{SAMPLING} = f_{DTS}/32</math>, N = 6.      1111: <math>f_{SAMPLING} = f_{DTS}/32</math>, N = 8.</p>
[6]	BK1SEL	<p>Break 1 Selection</p> <p>0: ETI signal is selected for the MT_ETI pin.      1: MT_BRK1 signal is selected for the MT_ETI pin.</p> <p>Note: The Break 1 and ETI signal are shared with the same MT_ETI pin. This control bit can switch the pin function for the second break signal of the MCTM.</p>
[5]	CHAOE	<p>Channel Automatic Output Enable</p> <p>0: CHMOE can be set only by software.      1: CHMOE can be set by software or automatically by an update event</p>
[4]	CHMOE	<p>Channel Main Output Enable</p> <p>Cleared asynchronously by hardware on a break event occurrence.</p> <p>0: MT_CHxO and MT_CHxNO are disabled or forced to idle states.      1: MT_CHxO and MT_CHxNO are enabled if the enable bits (CHxE, CHxNE) are set.</p>

<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>
[3]	BKP1	Break 1 Input Polarity 0: Break input active low 1: Break input active high
[2]	BKE1	Break 1 Enable 0: Break inputs disabled 1: Break inputs enabled
[1]	BKP0	Break 0 Input Polarity 0: Break input active low 1: Break input active high
[0]	BKE0	Break 0 Enable 0: Break inputs disabled 1: Break inputs enabled

## Timer PDMA/Interrupt Control Register – DICTR

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

Offset: 0x074

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved					TEVDE	UEV2DE	UEV1DE
						RW	0	RW
Type/Reset	23	22	21	20	19	18	17	16
	Reserved					CH3CCDE	CH2CCDE	CH1CCDE
						RW	0	RW
Type/Reset	15	14	13	12	11	10	9	8
	Reserved					BRKIE	TEVIE	UEV2IE
						RW	0	RW
Type/Reset	7	6	5	4	3	2	1	0
	Reserved					CH3CCIE	CH2CCIE	CH1CCIE
						RW	0	RW
Type/Reset								

Bits	Field	Descriptions
[26]	TEVDE	Trigger event PDMA Request Enable 0: Trigger PDMA request disabled 1: Trigger PDMA request enabled
[25]	UEV2DE	Update event 2 PDMA Request Enable 0: Update event 2 PDMA request disabled 1: Update event 2 PDMA request enabled
[24]	UEV1DE	Update event 1 DMA Request Enable 0: Update event 1 DMA request disabled 1: Update event 1 DMA request enabled
[19]	CH3CCDE	Channel 3 Capture/Compare PDMA Request Enable 0: Channel 3 PDMA request disabled 1: Channel 3 PDMA request enabled
[18]	CH2CCDE	Channel 2 Capture/Compare PDMA Request Enable 0: Channel 2 PDMA request disabled 1: Channel 2 PDMA request enabled
[17]	CH1CCDE	Channel 1 Capture/Compare PDMA Request Enable 0: Channel 1 PDMA request disabled 1: Channel 1 PDMA request enabled
[16]	CH0CCDE	Channel 0 Capture/Compare PDMA Request Enable 0: Channel 0 PDMA request disabled 1: Channel 0 PDMA request enabled
[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

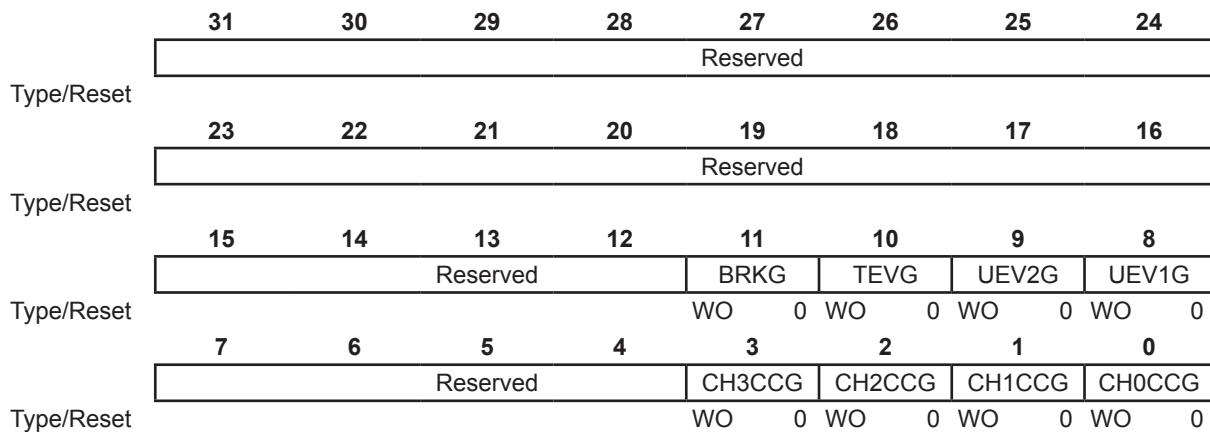
<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>
[9]	UEV2IE	Update event 2 Interrupt Enable 0: Update event 2 interrupt disabled 1: Update event 2 interrupt enabled
[8]	UEV1IE	Update event 1 Interrupt Enable 0: Update event 1 interrupt disabled 1: Update event 1 interrupt enabled
[3]	CH3CCIE	Channel 3 Capture/Compare 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 BRK0IF flag is set and then the CHMOE bit will be cleared
[10]	TEVG	Trigger Event Generation The trigger event TEV can be generated by setting this bit. It is cleared by hardware automatically. 0: No action 1: The TEVIF flag is set
[9]	UEV2G	Update Event 2 Generation The update event 2 UEV2 can be generated by setting this bit. It is cleared by hardware automatically. 0: No action 1: Update the CHxE, CHxNE, and CHxOM bits when COMPRE bit in CTR Register is set to 1
[8]	UEV1G	Update Event 1 Generation The update event 1 UEV1 can be generated by setting this bit. It is cleared by hardware automatically. 0: No action 1: Reinitialise the counter The counter value returns to 0 or the CRR preload value, depending on the counter mode in which the current timer is being used. An update operation on any related registers will also be executed. For a more detailed description, refer to the corresponding section.

Bits	Field	Descriptions
[3]	CH3CCG	<p>Channel 3 Capture/Compare Generation</p> <p>A Channel 3 capture/compare event can be generated by setting this bit. It is cleared by hardware automatically.</p> <ul style="list-style-type: none"> <li>0: No action</li> <li>1: Capture/compare event is generated on channel 3</li> </ul> <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> <ul style="list-style-type: none"> <li>0: No action</li> <li>1: Capture/compare event is generated on channel 2</li> </ul> <p>If Channel 2 is configured as an input, the counter value is captured into the CH2CCR register and then the CH2CCIF bit is set. If Channel 2 is configured as an output, the CH2CCIF bit is set.</p>
[1]	CH1CCG	<p>Channel 1 Capture/Compare Generation</p> <p>A Channel 1 capture/compare event can be generated by setting this bit. It is cleared by hardware automatically.</p> <ul style="list-style-type: none"> <li>0: No action</li> <li>1: Capture/compare event is generated on channel 1</li> </ul> <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> <ul style="list-style-type: none"> <li>0: No action</li> <li>1: Capture/compare event is generated on channel 0</li> </ul> <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			BRK1IF	BRK0IF	TEVIF	UEV2IF	UEV1IF	
	RW 0								
Type/Reset	7	6	5	4	3	2	1	0	
	CH3OCF	CH2OCF	CH1OCF	CH0OCF	CH3CCIF	CH2CCIF	CH1CCIF	CH0CCIF	
Type/Reset	RW	0	RW	0	RW	0	RW	0	

Bits	Field	Descriptions
[12]	BRK1IF	Break 1 Event Interrupt Flag This flag is set by hardware when a break 1 event occurs and is cleared by software. 0: No break 1 event occurred 1: Break 1 event occurred
[11]	BRK0IF	Break 0 Event Interrupt Flag This flag is set by hardware when a break 0 event occurs and is cleared by software. 0: No break 0 event occurred 1: Break 0 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.

Bits	Field	Descriptions
[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>

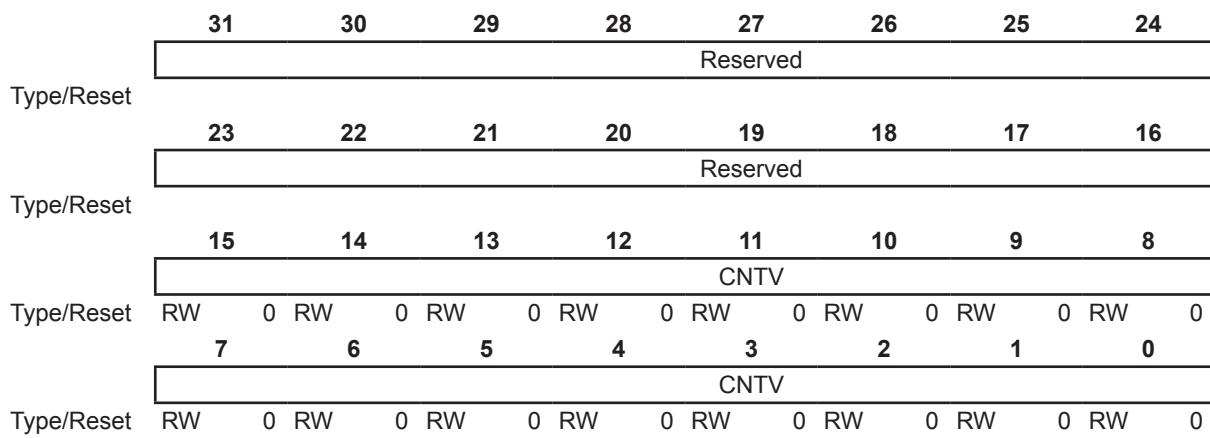
Bits	Field	Descriptions
[0]	CH0CCIF	<p>Channel 0 Capture/Compare Interrupt Flag</p> <ul style="list-style-type: none"><li>- Channel 0 is configured as an output<ul style="list-style-type: none"><li>0: No match event occurs</li><li>1: The contents of the counter CNTR have matched the content of the CH0CCR 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



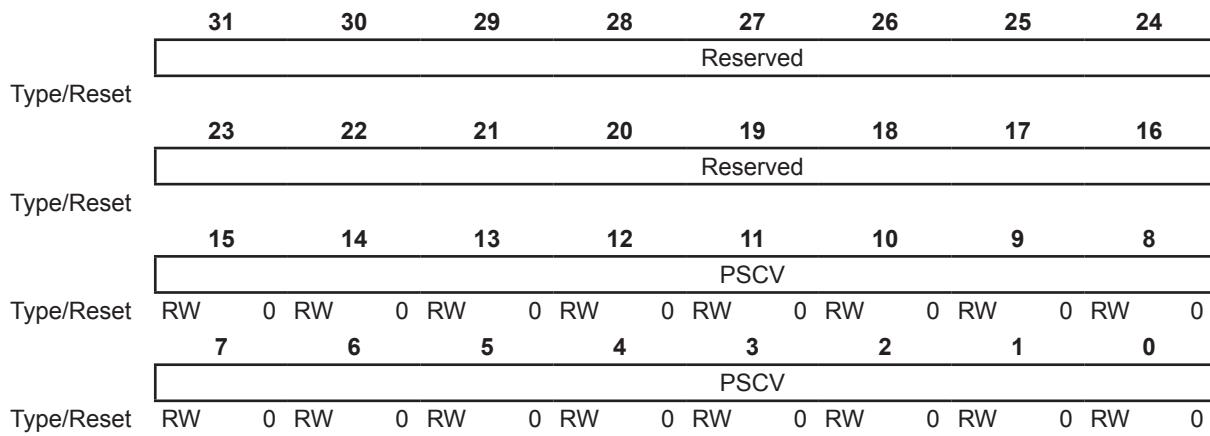
Bits	Field	Descriptions
[15:0]	CNTV	Counter Value.

## Timer Prescaler Register – PSCR

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

Offset: 0x084

Reset value: 0x0000\_0000



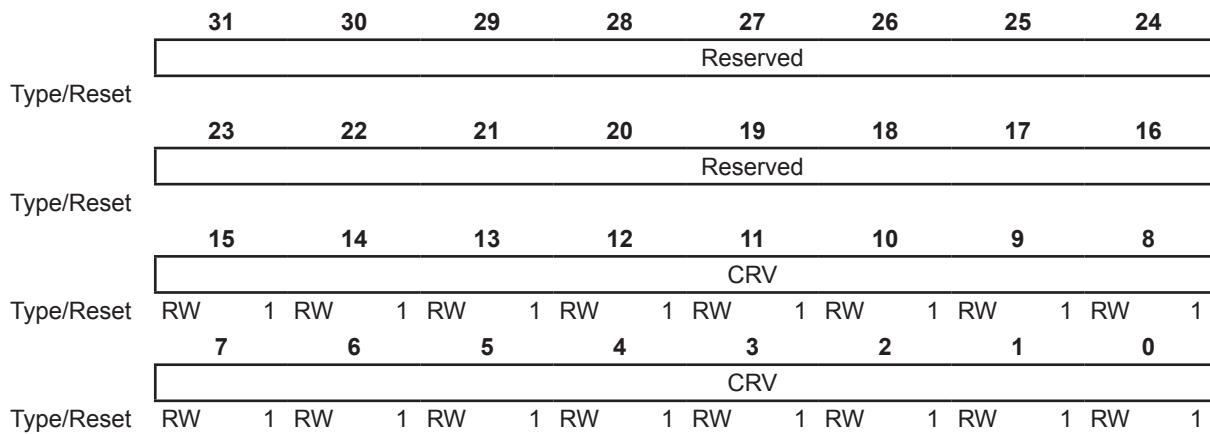
<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>
[15:0]	PSCV	Prescaler Value These bits are used to specify the prescaler value to generate the counter clock frequency $f_{CK\_CNT}$ . $f_{CK\_CNT} = \frac{f_{CK\_PSC}}{PSCV[15:0] + 1}$ , where the $f_{CK\_PSC}$ is the prescaler clock source.

## Timer Counter Reload Register – CRR

This register specifies the timer counter reload value.

Offset: 0x088

Reset value: 0x0000\_FFFF



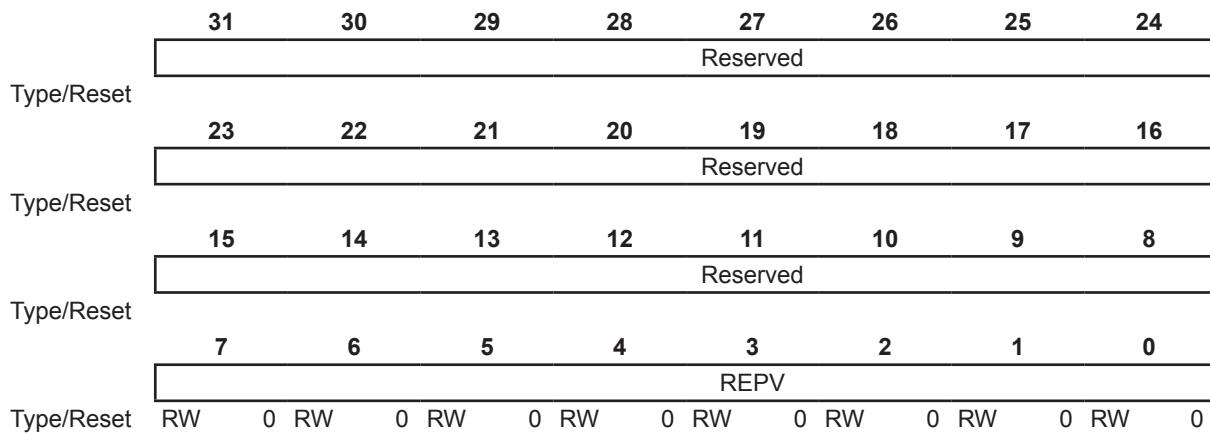
Bits	Field	Descriptions
[15:0]	CRV	Counter Reload Value The CRV is the reload value which is loaded into the actual counter register.

## Timer Repetition Register – REPR

This register specifies the timer repetition counter value.

Offset: 0x08C

Reset value: 0x0000\_0000



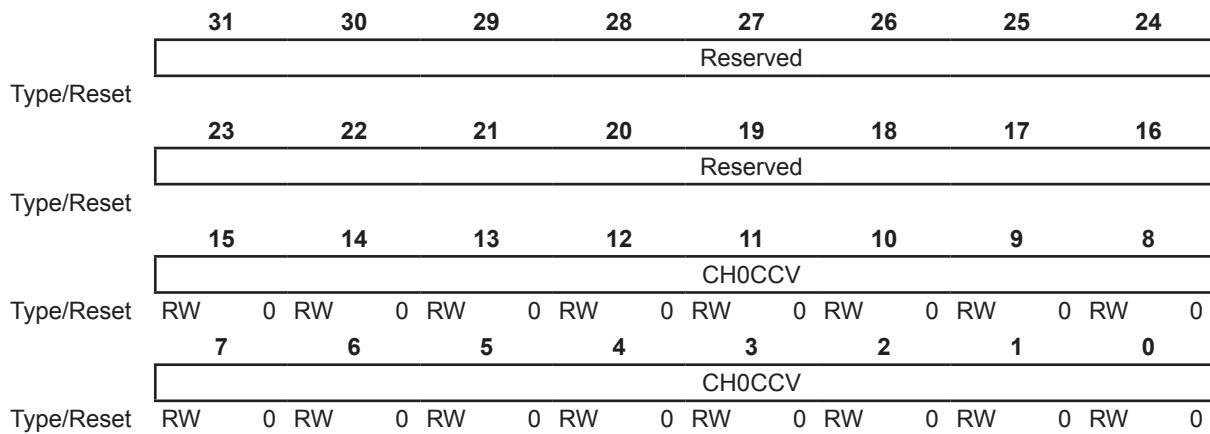
Bits	Field	Descriptions
[7:0]	REPV	Repetition Counter Value. These bits allow the user to specify the update rate of the compare registers.

## Channel 0 Capture/Compare Register – CH0CCR

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

Offset: 0x090

Reset value: 0x0000\_0000



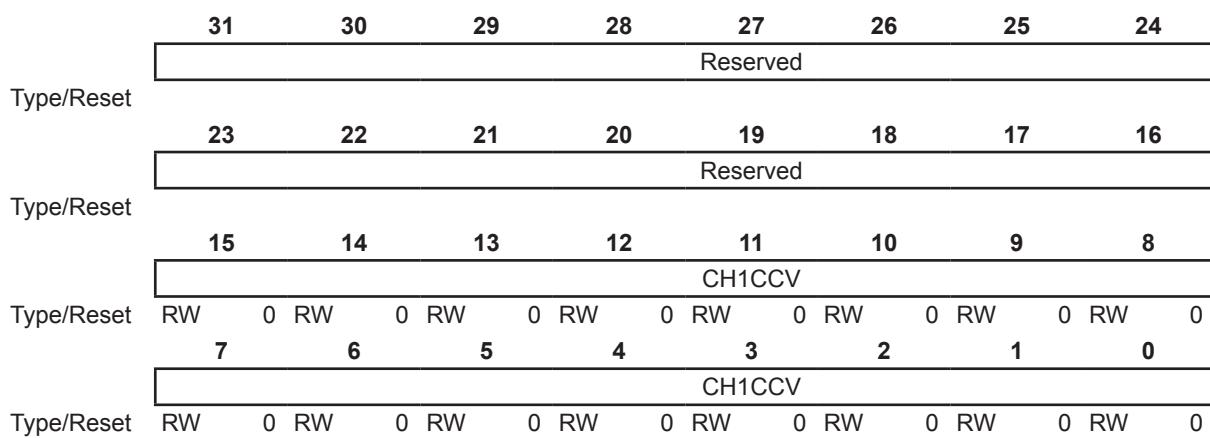
Bits	Field	Descriptions
[15:0]	CH0CCV	<p>Channel 0 Capture/Compare Value</p> <ul style="list-style-type: none"> <li>- When Channel 0 is configured as an output</li> <li>The CH0CCR value is compared with the counter value and the comparison result is used to trigger the CH0OREF output signal.</li> <li>- 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
Reserved								
Type/Reset	23	22	21	20	19	18	17	16
Reserved								
Type/Reset	15	14	13	12	11	10	9	8
CH2CCV								
Type/Reset	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
Type/Reset	RW	0	RW	0	RW	0	RW	0
CH2CCV								
Type/Reset	RW	0	RW	0	RW	0	RW	0

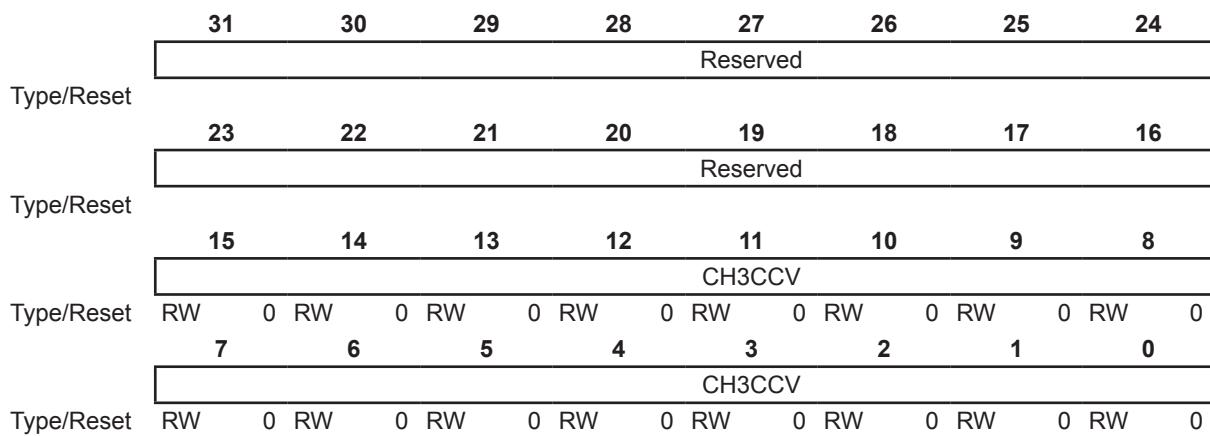
Bits	Field	Descriptions
[15:0]	CH2CCV	<p>Channel 2 Capture/Compare Value</p> <ul style="list-style-type: none"> <li>- When Channel 2 is configured as an output</li> <li>The CH2CCR value is compared with the counter value and the comparison result is used to trigger the CH2OREF output signal.</li> <li>-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

	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
CH0ACV								
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								
Type/Reset	RW	0	RW	0	RW	0	RW	0

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 is this register will be compared to the counter.

## Channel 1 Asymmetric Compare Register – CH1ACR

This register specifies the timer channel 1 asymmetric compare value.

Offset: 0x0A4

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[15:0]	CH1ACV	Channel 1 Asymmetric Compare Value When channel 1 is configured as asymmetric PWM mode and the counter is counting down, the value written is 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
Reserved								
Type/Reset	23	22	21	20	19	18	17	16
Reserved								
Type/Reset	15	14	13	12	11	10	9	8
CH2ACV								
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								
Type/Reset	RW	0	RW	0	RW	0	RW	0

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

## Channel 3 Asymmetric Compare Register – CH3ACR

This register specifies the timer channel 3 asymmetric compare value.

Offset: 0x0AC

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[15:0]	CH3ACV	<p>Channel 3 Asymmetric Compare Value</p> <p>When channel 3 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>

# 17 Real Time Clock (RTC)

## Introduction

The Real Time Clock, RTC, circuitry includes the APB interface, a 32-bit up-counter, a control register, a prescaler, a compare register and a status register. Most of the RTC circuits are located in the Backup Domain, as shown shaded in the accompanying figure, except for the APB interface. The APB interface is located in the V<sub>DD18</sub> domain. Therefore, it is necessary to be isolated from the ISO signal that comes from the power control unit when the V<sub>DD18</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.

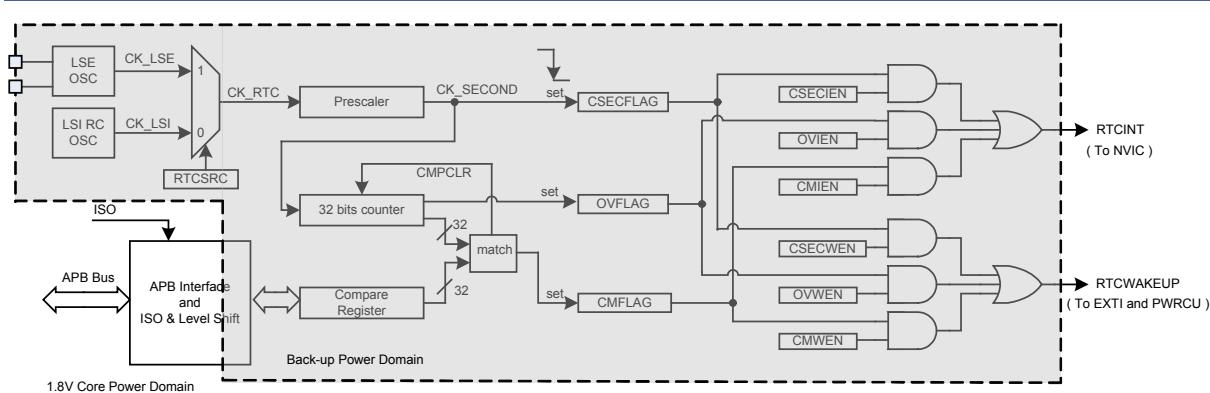


Figure 121. RTC Block Diagram

## Features

- 32-bit up counter for counting elapsed time
- Programmable clock prescaler
  - Division factor: 1, 2, 4, 8..., 32768
- 32-bit compare register for alarm usage
- RTC clock source could be any of the following
  - 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 modes

## Functional Descriptions

### RTC Related Register Reset

The RTC registers can only be reset by either a Backup Domain power on reset, PORB, or by a Backup Domain software reset by setting the BAKRST bit in the BAKCR register. Other reset events have no effect to clear the RTC registers.

### Reading RTC Register

The RTC control logic and the related registers are powered by the  $V_{BAK}$  supply voltage. Therefore, the RTC circuit remains operational in the Power-Down mode where  $V_{DD18}$  is powered off. Only the APB bus, which is located in  $V_{DD18}$  domain, is interconnected to the circuits located in the  $V_{BAK}$  domain using level shift circuitry and isolated by the ISO signals when the  $V_{DD18}$  supply voltage is powered off. The isolation function must be disabled by setting BAKISO bit to 1 in the LPCR register as described in the Clock Control Unit before accessing the RTC registers using the APB bus.

### Low Speed Clock Configuration

The default RTC clock source, CK\_RTC, is derived from the LSI oscillator. The CK\_RTC clock can be derived from either the external 32,768 Hz crystal oscillator, named the LSE oscillator, or the internal 32K RC oscillator, named the LSI oscillator, by setting the RTCSRC bit in the RTCCR register. A prescaler is provided to divide the CK\_RTC by a ratio ranged from  $2^0$  to  $2^{15}$  determined by the RPRES[3:0] field. For instance, setting the prescaler value RPRES[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 lower power consumption. It is a trade-off between startup time and power consumption 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 40. LSE Startup Mode Operating Current and Startup Time**

Startup mode	LSECM Setting in the RTCCR register	Operating Current	Startup time
Normal startup	0	3 $\mu$ A	Above 200 ms
Fast startup	1	8 $\mu$ A	Below 200 ms

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

## RTC Counter Operation

The RTC provides a 32-bit up-counter which increments at the falling edge of the CK\_SECOND clock and whose value can be read from the RTCCNT register asynchronously via the APB bus. A 32-bit compare register, RTCCMP, is provided to store the specific value to be compared with the RTCCNT content. This is used to define a pre-determined time interval. When the RTCCNT register content is equal to the RTCCMP register value, the match flag CMFLAG in the RTCSR register will be set by hardware and an interrupt or wakeup event can be sent according to the corresponding enable bits in the RTCIWEN register. The RTC counter will be either reset to zero or keep counting when the compare match event occurs, dependent upon the CMPCLR bit in the RTCCR register. For example, if the 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 Cortex™-M3 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 41. RTCOUT Output Mode and Active Level Setting**

ROWM	ROES	RTCOUT Output Waveform				
0 (Pulse mode)	0 Compare match	RTCCMP	4			
		RTCCNT	3	4	5	
		RTCOUT (ROAP = 0)		$T_R$		
		RTCOUT (ROAP = 1)			$T_R$	
		ROLF				
	1 Second clock	RTCCMP	X			
		RTCCNT	3	4	5	
		RTCOUT (ROAP = 0)	$T_R$	$T_R$	$T_R$	
		RTCOUT (ROAP = 1)				
		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 / 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>BAK</sub> backup power domain.

**Table 42. RTC Register Map**

Register	Offset	Description	Reset Value
<b>RTC Base Address = 0x4006_A000</b>			
RTCCNT	0x000	RTC Counter Register	0x0000_0000
RTCCMP	0x004	RTC Compare Register	0x0000_0000
RTCCR	0x008	RTC Control Register	0x0000_0F04
RTCSR	0x00C	RTC Status Register	0x0000_0000
RTCIWEN	0x010	RTC Interrupt and Wakeup Enable Register	0x0000_0000

## Register Descriptions

### RTC Counter Register – RTCCNT

This register defines a 32-bit up counter which is increased by the CK\_SECOND clock.

Address: 0x000

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[31:0]	RTCCNTV	<p>RTC Counter Value</p> <p>The current value of the RTC counter is returned when reading the RTCCNT register. The RTCCNT register is updated during the falling edge of the CK_SECOND. This register is reset by one of the following conditions:</p> <ul style="list-style-type: none"> <li>- Backup Domain software reset - set the BAKRST bit in the BAKCR register</li> <li>- Backup Domain power on reset - PORB</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.

Address: 0x004

Reset value: 0x0000\_0000 (Reset by Backup Domain reset only)

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

Bits	Field	Descriptions
[31:0]	RTCCMPV	<p>RTC Compare Match Value</p> <p>A match condition happens when the value in the RTCCNT register is equal to RTCCMP value. An interrupt can be generated if the CMIEN bit in the RTCIWIN 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 RTCCSR 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 RTCCSR register is set and the RTCCNT register will be reset to zero and then the counter continues to count.</p>

## RTC Control Register – RTCCR

This register specifies a range of RTC circuitry control bits

Address: 0x008

Reset value: 0x0000\_0F04 (Reset by Backup Domain reset only)

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

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.

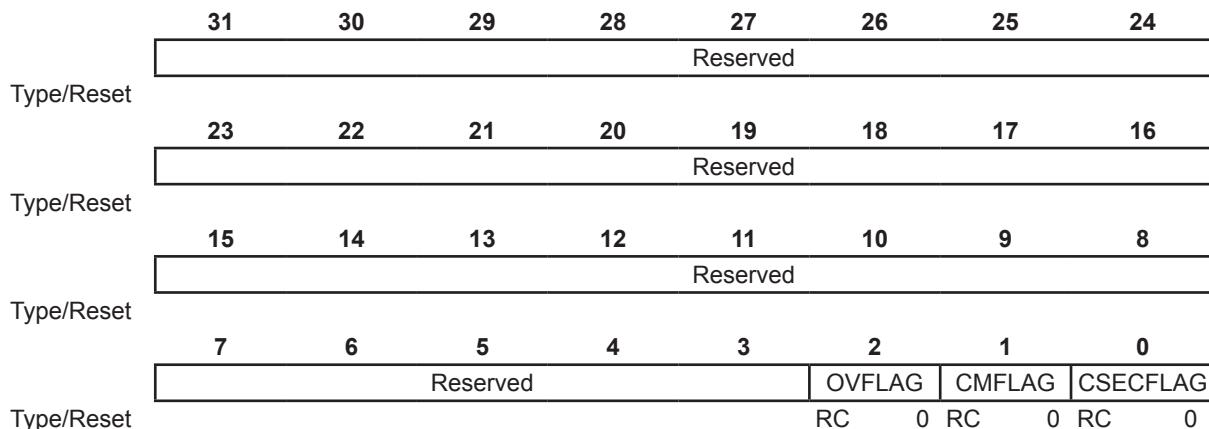
Bits	Field	Descriptions
[11:8]	RPRE	RTC Clock Prescaler Select $CK_{SECOND} = CK_{RTC} / 2^{RPRE}$ 0000: $CK_{SECOND} = CK_{RTC} / 2^0$ 0001: $CK_{SECOND} = CK_{RTC} / 2^1$ 0010: $CK_{SECOND} = CK_{RTC} / 2^2$ ... 1111: $CK_{SECOND} = CK_{RTC} / 2^{15}$
[5]	LSESM	LSE oscillator Startup Mode 0: Normal startup and requires less operating power 1: Fast startup but requires higher operating current
[4]	CMPCLR	Compare Match Counter Clear 0: 32-bit RTC counter is not affected when compare match condition occurs 1: 32-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 Backup Domain powered up. Note: After the backup domain is powered on, the internal LSI RC oscillator will start to oscillate. The frequency range of the LSI oscillator is shown in the LSI oscillator electrical characteristics in the datasheet. The device also provides a production trim value to obtain a more accurate oscillation frequency. The procedure is to disable the LSI oscillator and then enable it again after the backup domain is powered on. After the trimming procedure has completed, the system will automatically load the production trim value to the frequency trimming circuit of the LSI RC oscillator.
[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.

Address: 0x00C

Reset value: 0x0000\_0000 (Reset by Backup Domain reset and RTCEN bit change from 1 to 0)



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

## RTC Interrupt and Wakeup Enable Register – RTCIWEN

This register contains the interrupt and wakeup enable bits.

Address: 0x010

Reset value: 0x0000\_0000 (Reset by Backup Domain reset only)

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

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

# 18 Watchdog Timer (WDT)

## Introduction

The Watchdog timer is a hardware timing device that can be used to detect a system lock-up due to software trapped in a deadlock. The Watchdog timer can be operated in an interrupt mode or a reset mode. The Watchdog timer will generate an interrupt or 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, an interrupt or 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.

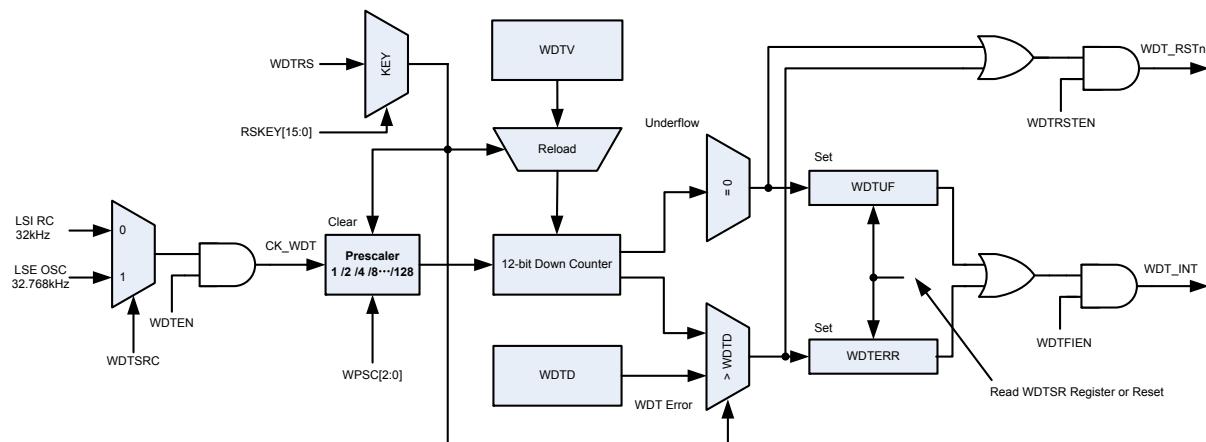


Figure 122. Watchdog Timer Block Diagram

## 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 mode or deep sleep mode 1
- 12-bit down counter with 3-bit prescaler structure
- Can be configured to generate a system reset or interrupt
- Limited reload window setup function for custom Watchdog timer reload times
- Watchdog Timer may be stopped when the processor is in the debug or sleep mode
- Reload lock key to prevent unexpected operation
- Configuration register write protection function for counter value, interrupt enable, reset enable, delta value, and prescaler

## Functional Descriptions

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 32.768kHz slow clock and a 1/128 maximum prescaler value.

The Watchdog timer configuration setup includes a programmable counter reload value, interrupt enable, 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 interrupt or reset. The 12-bit count-down counter can be reloaded with the required Watchdog Timer Reload 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 these situations from occurring, the reload operation must be executed in such a way that the value of the Watchdog timer counter is limited below a specified delta value (WDTD). If the Watchdog timer with counter value higher than the delta value is reloaded, a Watchdog Timer error will be triggered. The Watchdog timer error generates either a Watchdog interrupt or a reset depending upon the related setup. 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 WDTSTS register will be set respectively when the Watchdog timer underflows or when a Watchdog timer error occurs. A system reset or read operation on the WDTSTS register clears the WDTERR and WDTUF flags.

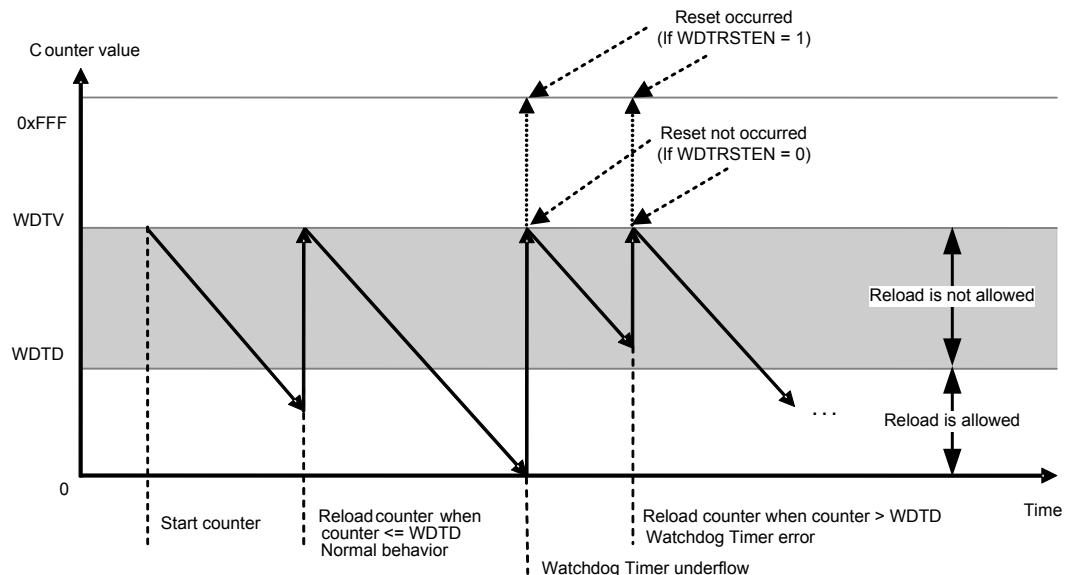
The watchdog timer block uses two clocks: PCLK and CK\_WDT. The PCLK clock is used for APB access to the watchdog registers. The CK\_WDT clock which is sourced from LSE or LSI is used for Watchdog timer counting. There is some synchronisation logic between these two clock domains.

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 or interrupt will occur any time when the Watchdog timer is running and when it has an operating clock source. If a Watchdog interrupt occurs in the Sleep or Deep Sleep mode 1, it will wake up the device.

When the system enters the debug mode, the Watchdog timer counter will either continue to count or stop depending on the DB\_WDT 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/interrupt 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 with the delta value (WDTD).



**Figure 123. Watchdog Timer Behavior**

## Register Map

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

**Table 43. WDT Register Map**

Register	Offset	Description	Reset Value
<b>Watchdog Timer Base Address = 0x4006_8000</b>			
WDTCSR	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
WDTCNTR	0x014	Watchdog Timer Counter Register	0x0000_0FFF
WDTCSR	0x018	Watchdog Timer Clock Selection Register	0x0000_0000

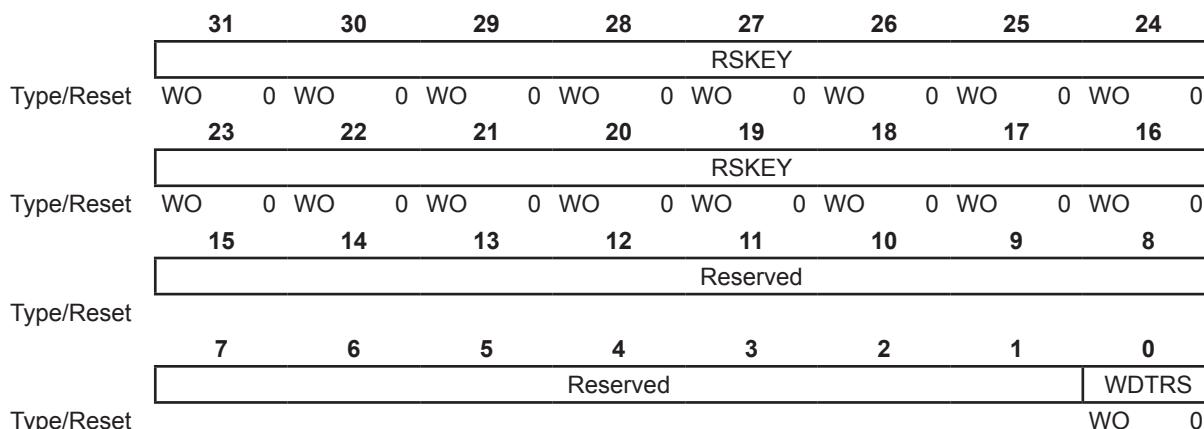
## Register Descriptions

### Watchdog Timer Control Register – WDTCSR

This register is used to reload 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 value of 0x5FA0 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 with a WDTV value 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, interrupt enable and reset enable control.

Offset: 0x004

Reset value: 0x0000\_0FFF

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

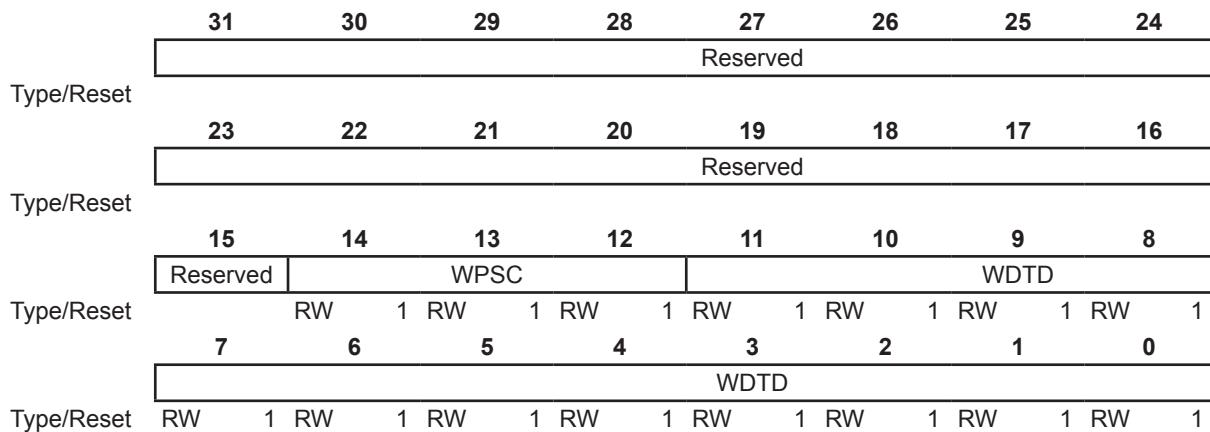
Bits	Field	Descriptions
[16]	WDTEN	Watchdog Timer Running Enable 0: Watchdog timer is disabled 1: Watchdog timer is enabled to run When the Watchdog timer is disabled, the counter will be reset to its hardware default condition. When the WDTEN bit is set, the Watchdog timer will be reloaded with the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

## Watchdog Timer Mode Register 1 – WDTMR1

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

Offset: 0x008

Reset value: 0x0000\_7FFF



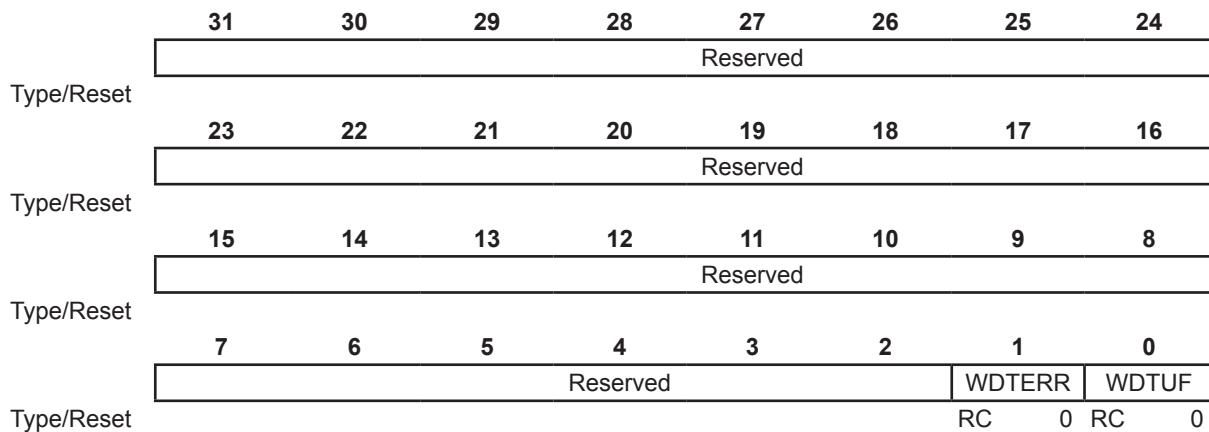
Bits	Field	Descriptions
[14:12]	WPSC	Watchdog Timer Prescaler Selection 000: 1/1 001: 1/2 010: 1/4 011: 1/8 100: 1/16 101: 1/32 110: 1/64 111: 1/128
[11:0]	WDTD	Watchdog Timer Delta Value Define the permitted range to reload the Watchdog timer. If the Watchdog timer counter value is less than or equal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



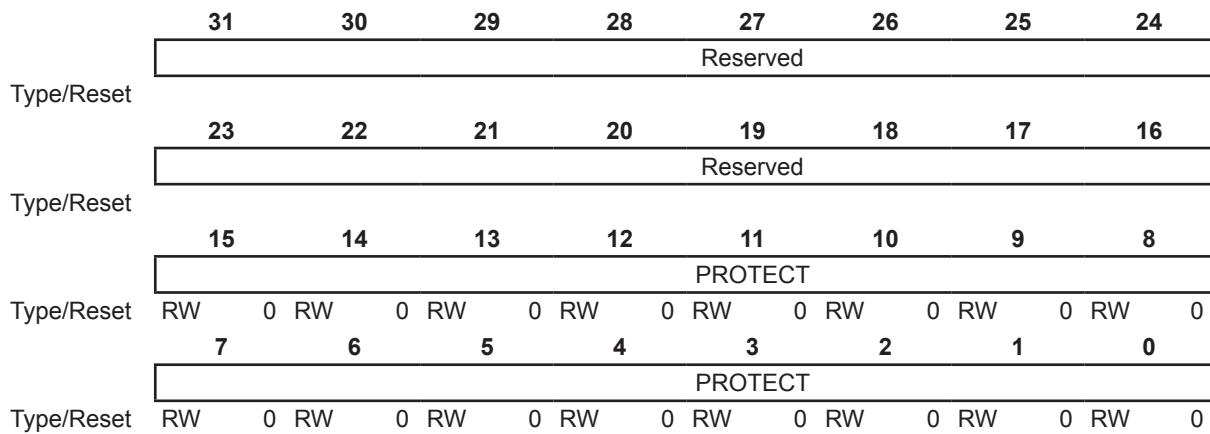
Bits	Field	Descriptions
[1]	WDTERR	<p>Watchdog Timer Error</p> <p>0: No Watchdog timer error occurred since the last read operation on this register</p> <p>1: Watchdog timer error occurred since the last read operation on this register</p> <p>Note: A reload operation will cause a Watchdog Timer error when the Watchdog Timer counter value is greater than or equal to the WDTD value</p>
[0]	WDTUF	<p>Watchdog timer Underflow</p> <p>0: No Watchdog Timer underflow occurred since the last read operation on this register</p> <p>1: Watchdog Timer underflow occurred since the last read operation on this register</p> <p>Note this bit is read clear flag.</p>

## Watchdog Timer Protection Register – WDTPR

This register specifies the Watchdog Timer write protection key configuration.

Offset: 0x010

Reset value: 0x0000\_0000



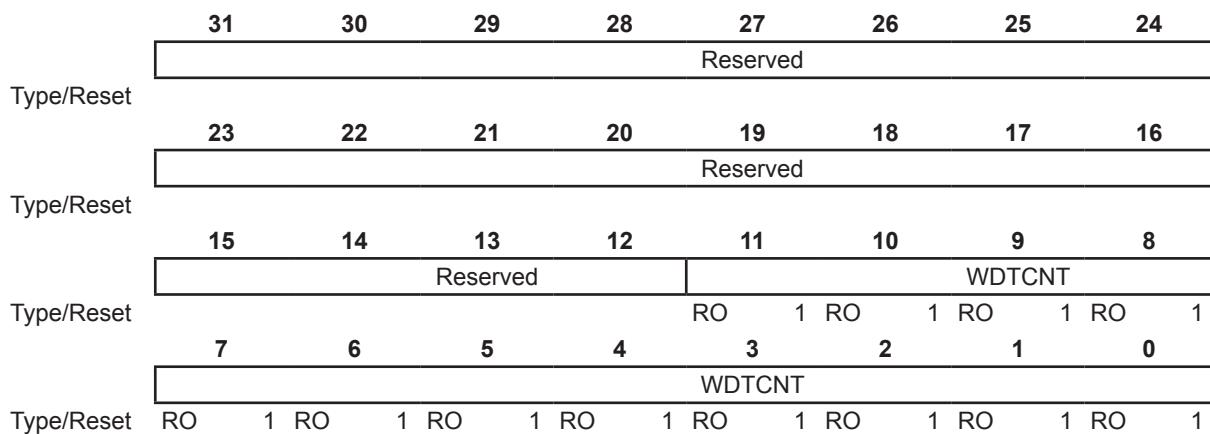
Bits	Field	Descriptions
[15:0]	PROTECT	<p>Watchdog Timer Register Protection</p> <p>For write operation:</p> <ul style="list-style-type: none"> <li>0x35CA: Disable the Watchdog Timer register write protection</li> <li>Others values: Enable the Watchdog Timer register write protection</li> </ul> <p>For 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 Counter Register – WDTCNTR

This register specifies Watchdog Timer counter value configuration.

Offset: 0x014

Reset value: 0x0000\_0FFF



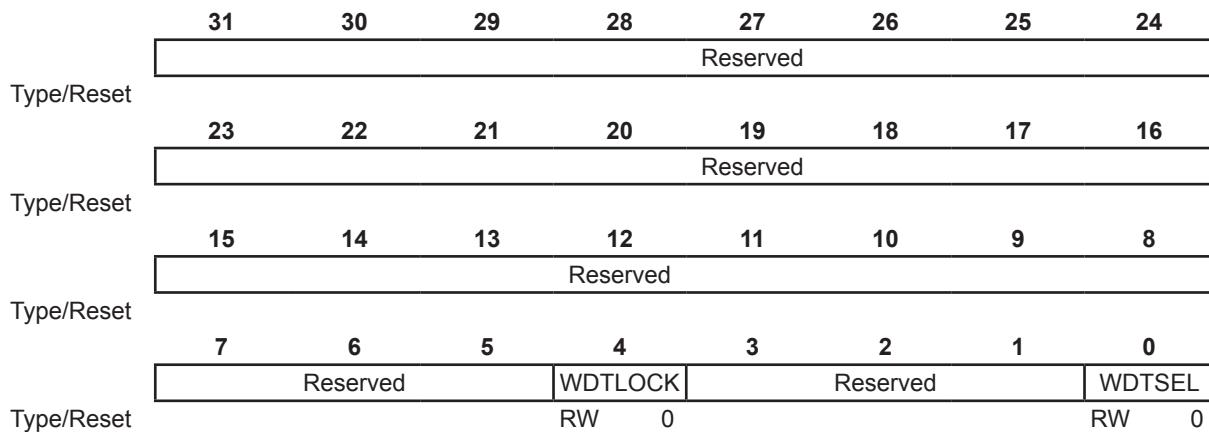
Bits	Field	Descriptions
[11:0]	WDTCNT	<p>Watchdog Timer Counter Value</p> <p>When reading the value of the 12-bit Watchdog timer, as the synchronisation procedure takes several CK_WDT cycles and PCLK cycles, the actual WDTCNT value read will be a previous value rather than the present actual value of the timer if it is read by the CPU.</p>

## Watchdog Timer Clock Selection Register – WDTCSR

This register specifies Watchdog Timer clock source selection and lock configuration.

Offset: 0x018

Reset value: 0x0000\_0FFF



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

# 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.

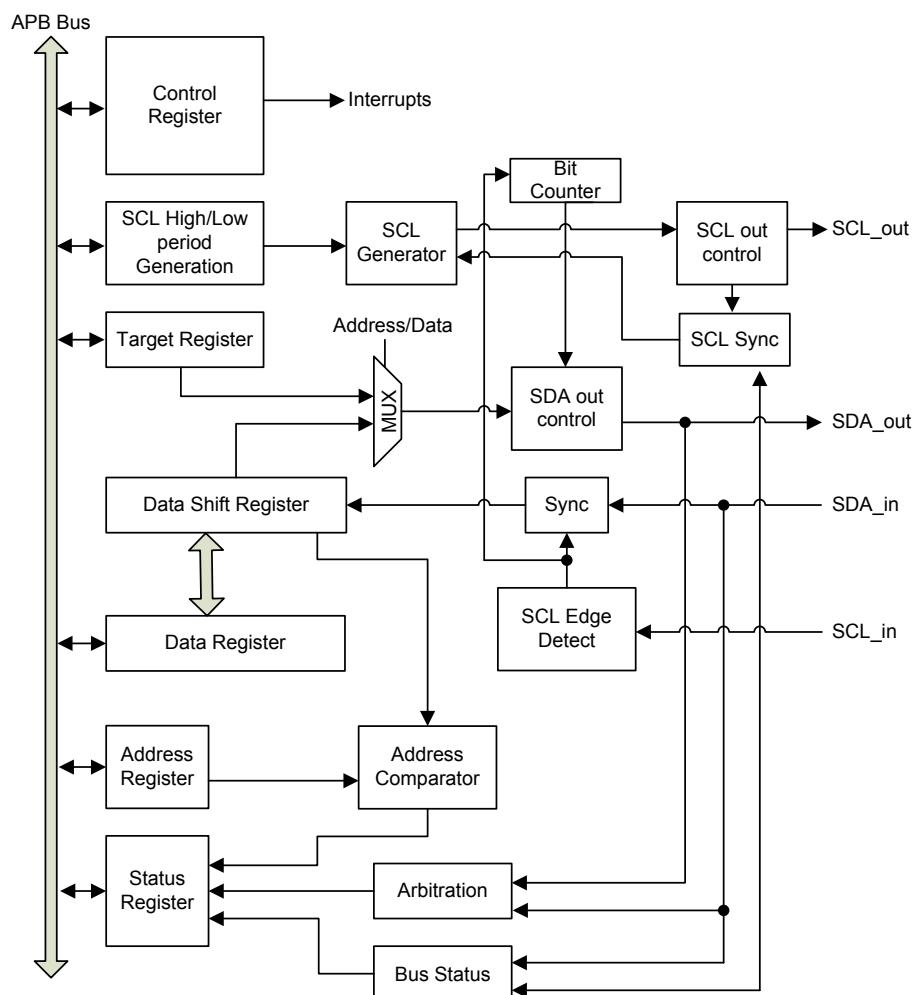


Figure 124. I<sup>2</sup>C Module Block Diagram

## 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
- Supports PDMA Interface

## 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.

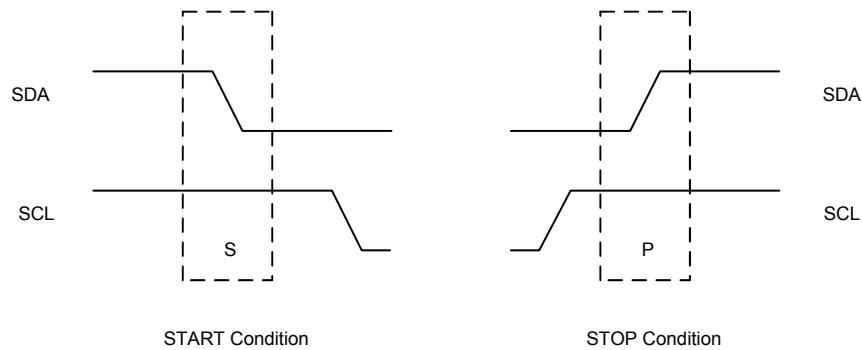


Figure 125. START and STOP Condition

### 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.

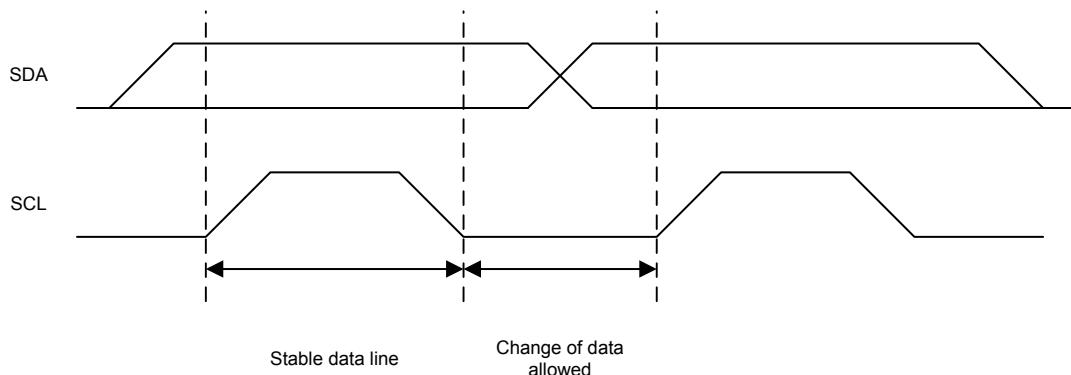


Figure 126. Data Validity

## 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-bit 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.

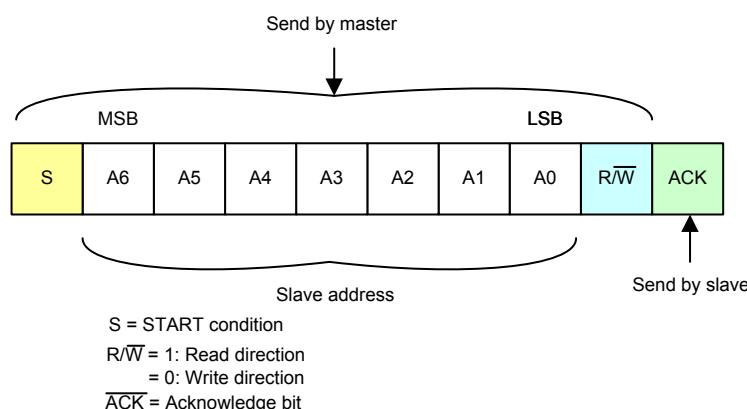
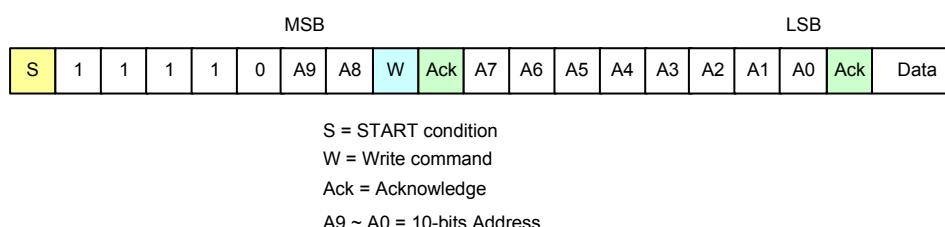


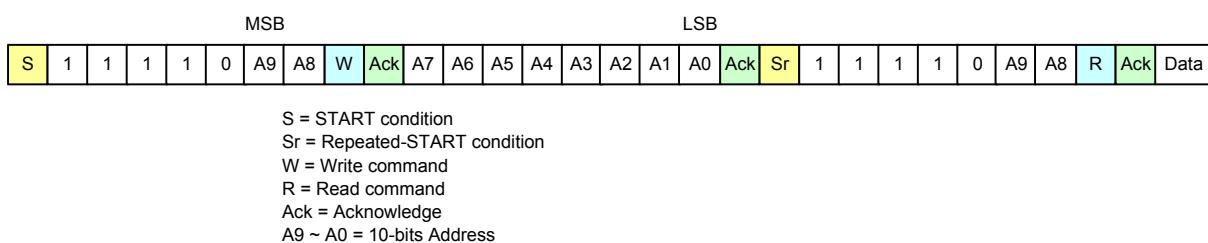
Figure 127. 7-bit Addressing Mode

## 10-bit Address Format

In order to prevent address clashes, due to the limited range of the 7-bit addresses, a new 10-bit address scheme has been introduced. This enhancement can be mixed with the 7-bit addressing mode which increases the available address range about ten times. For the 10-bit addressing mode, the first two bytes after a START signal include a header byte and an address byte that usually determines which slave will be selected by the master. The header byte is composed of a leading “11110”, 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 128. 10-bit Addressing Write Transmit Mode**



**Figure 129. 10-bit Addressing Read Receive Mode**

## Data Transfer and Acknowledge

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

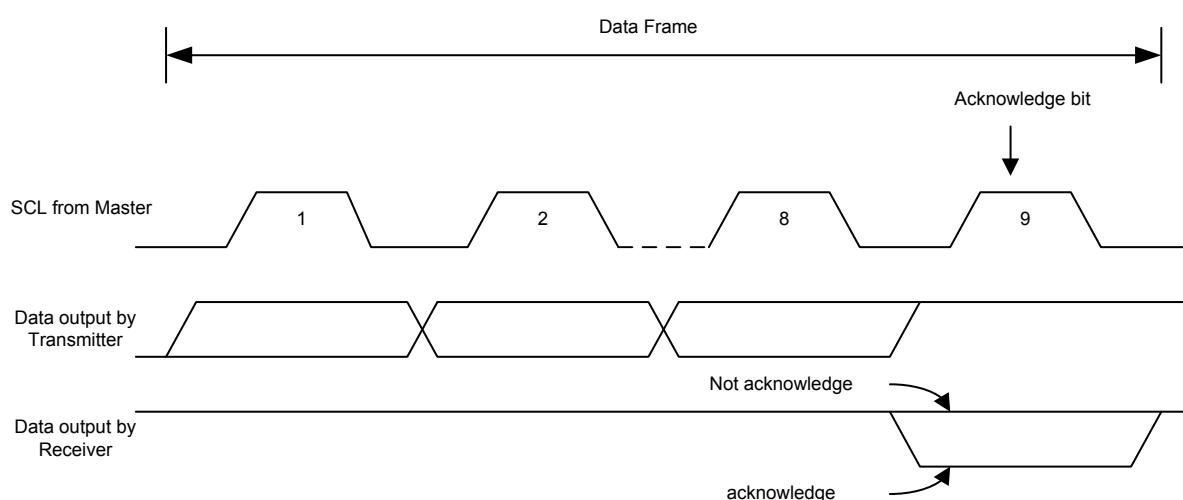


Figure 130. I<sup>2</sup>C Bus Acknowledge

## 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.

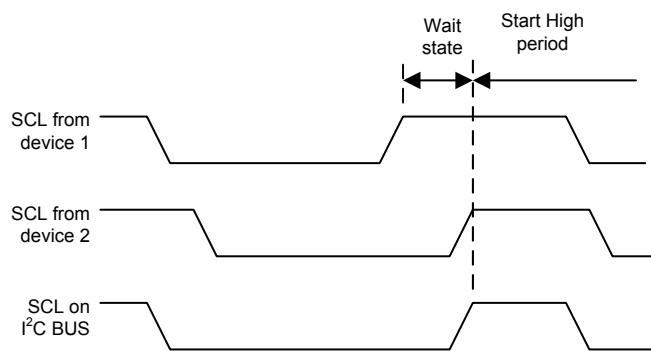


Figure 131. Clock Synchronization during Arbitration

## 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.

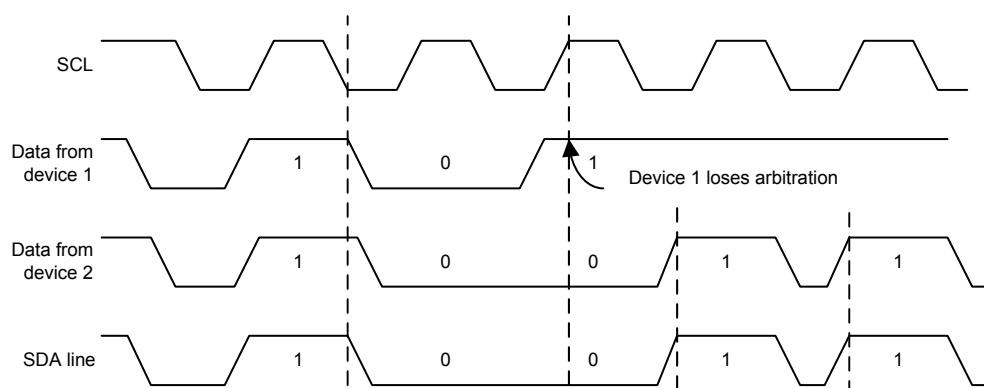


Figure 132. Two Master Arbitration Procedure

## General Call Address

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

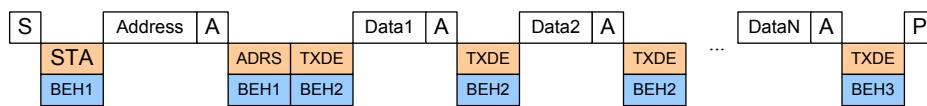
The data to be transmitted to the slave device must be transferred to the I2CDR register.

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.

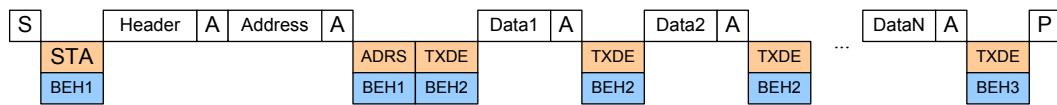
### Close / Continue Transmission

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 133. 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.

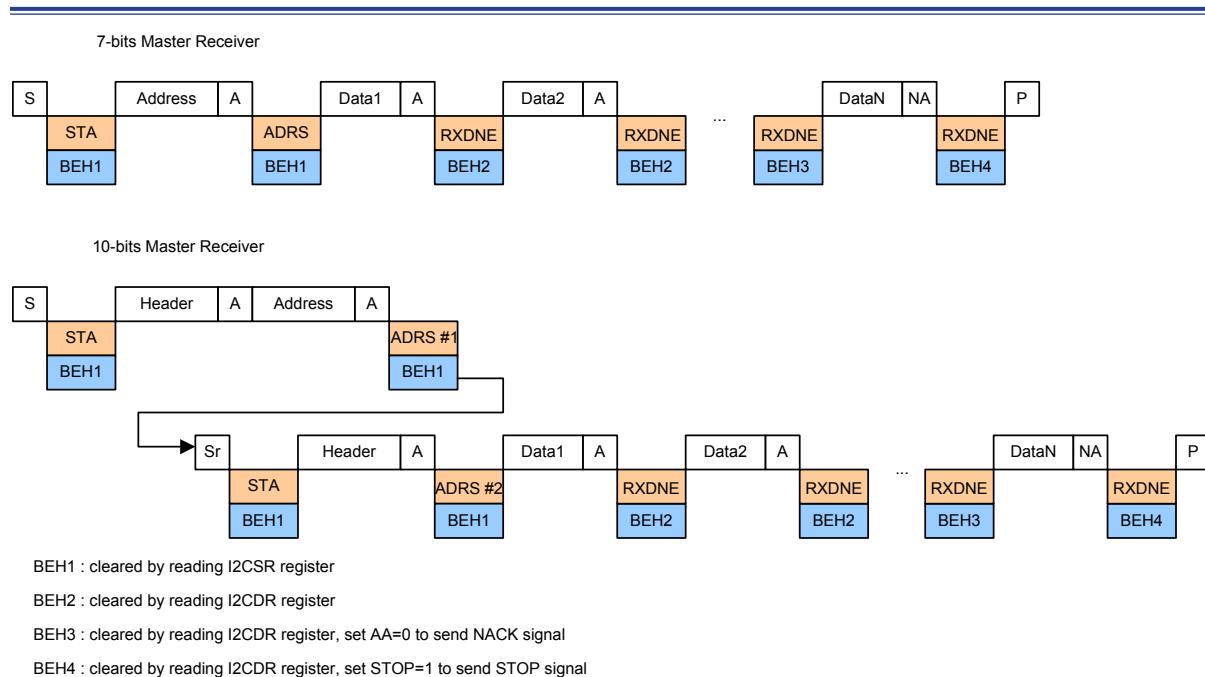
In the 10-bit addressing mode: The ADRS bit in the I2CSR register will be set twice in the 10-bit addressing mode. The first time the ADRS bit is set is when the 10-bit address is sent and the acknowledge signal from the slave device is received. The second time the ADRS bit is set is when the header 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 134. 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

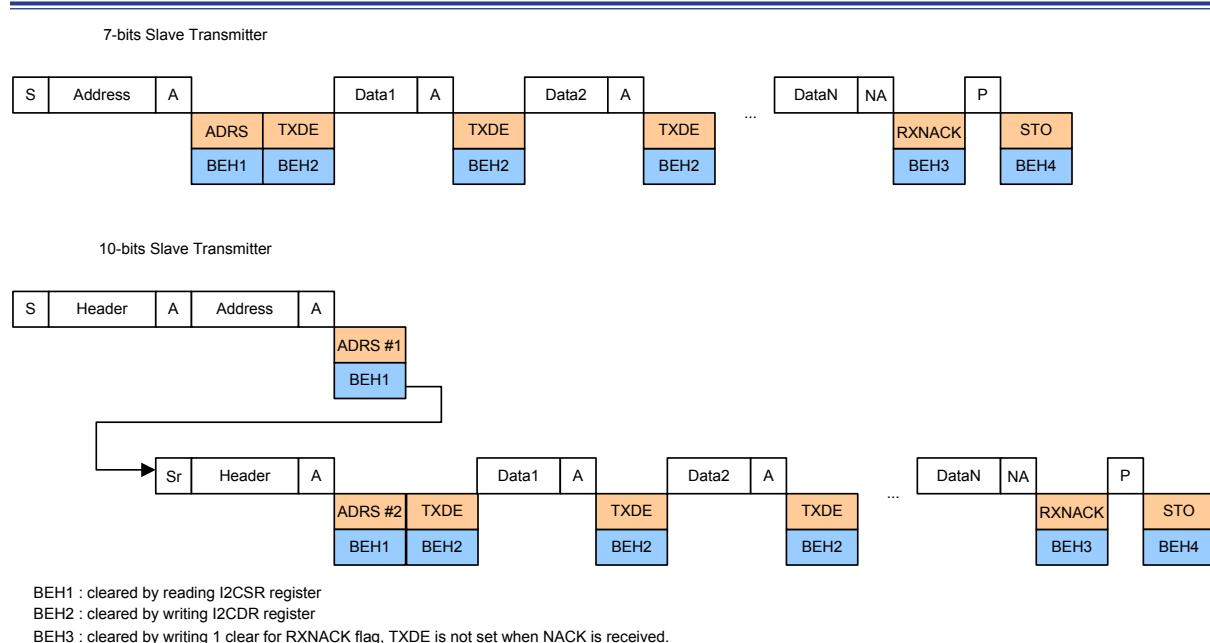
In the Slave transmitter mode, the TXDE bit is set to indicate that the I2CDR is empty, which results in the SCL line being held at a logic low state. New transmission data must then be written into the I2CDR register to continue the data transfer process. Writing a data into the I2CDR register will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



**Figure 135. Slave Transmitter Timing Diagram**

## Slave Receiver Mode

### Address Frame

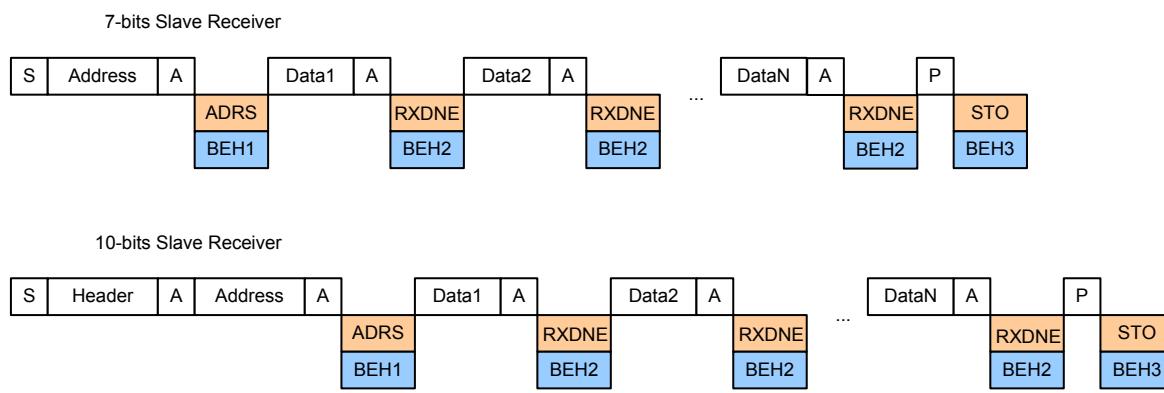
The ADRS bit in the I2CSR register is set after the slave device receives the calling address which matches with the slave device address. After the ADRS bit has been set to 1, it must be cleared to 0 to continue the data transfer process. The ADRS flag is cleared after reading the I2CSR register.

### 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 136. 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 44. 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.11 and Fig.12) Slave: I <sup>2</sup> C is addressed as slave device (Note: Reference Fig.13 and Fig.14)	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.11, 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.

## PDMA Interface

The PDMA interface is integrated in the I<sup>2</sup>C module. The PDMA function can be enabled by setting the TXDMAE or RXDMAE bit to 1 in the transmitter or receiver mode respectively. When the data register is empty in the transmitter mode and the TXDMAE bit is set to 1, the PDMA function will be activated to move data from a certain memory location into the I<sup>2</sup>C data register. Similarly, when the data register is not empty in the receiver mode and the RXDMAE bit is set to 1, the PDMA function will also be activated to move data from the I<sup>2</sup>C data register to a specific memory location.

The DMA NACK control bit, DMANACK, is used to determine whether the NACK signal is sent or not when the I<sup>2</sup>C module operates in the master receiver mode and the PDMA function is enabled. If the DMANACK bit is set to 1 and the data has all been received and moved using the PDMA interface, a NACK signal will automatically be sent out to properly terminate the data transfer.

For a mode detailed description on the PDMA configurations, refer to the PDMA chapter.

## Register Map

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

**Table 45. I<sup>2</sup>C Register Map**

Register	Offset	Description	Reset Value
<b>I<sup>2</sup>C0 Base Address = 0x4004_8000</b>			
<b>I<sup>2</sup>C1 Base Address = 0x4004_9000</b>			
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

	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	RW	0	RW	0	RW	RW	0	RW	
Type/Reset	7	6	5	4	3	2	1	0	
Type/Reset	ADRM	Reserved			I2CEN	GCEN	STOP	AA	
Type/Reset	RW	0			RW	0	RW	0	

Bits	Field	Descriptions
[14:15]	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]	COMBFILTERE	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.
[10]	DMANACK	DMA Mode NACK Control 0: No operation 1: The I <sup>2</sup> C master receiver module sends a NACK signal automatically after receiving the last byte from the slave transmitter in the DMA mode When the I2CEN bit is cleared to 0, the DMANACK bit is automatically cleared to 0 by hardware.

Bits	Field	Descriptions
[9]	RXDMAE	DMA Mode RX Request Enable Control 0: RX DMA request disabled 1: RX DMA request enabled If the data register is not empty in the receiver mode and the RXDMAE bit is set to 1, the relevant PDMA channel will be activated to move the data from the data register to a specific location which is defined in the corresponding PDMA register. When the I2CEN bit is cleared to 0, the RXDMAE bit is automatically cleared to 0 by hardware.
[8]	TXDMAE	DMA Mode TX Request Enable Control 0: TX DMA request disabled 1: TX DMA request enabled If the data register is empty in the transmitter mode and the Txdmae bit is set to 1, the relevant PDMA channel will be activated to move the data from a specific location defined in the related PDMA register to the data register. When the I2CEN bit is cleared to 0, the Txdmae bit is automatically cleared to 0 by hardware.
[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
[2]	GCEN	General Call Enable 0: General call disabled 1: General call enabled 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.
[1]	STOP	STOP Condition Control 0: No action 1: Send a STOP condition in master mode 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.
[0]	AA	Acknowledge Bit 0: Send a Not Acknowledge (NACK) signal after a byte is received 1: Send an Acknowledge (ACK) signal after a byte is received When the I2CEN bit is cleared to 0, the AA bit is automatically cleared to 0 by hardware.

## 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					RXBFIE	TXDEIE	RXDNEIE	
	15	14	13	12	11	10	9	8	
Type/Reset	Reserved			TOUTIE	BUSERRIE	RXNACKIE	ARBLOSIE		
	7	6	5	4	3	2	1	0	
Type/Reset	Reserved			GCSIE	ADRSIE	STOIE	STAIE		

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

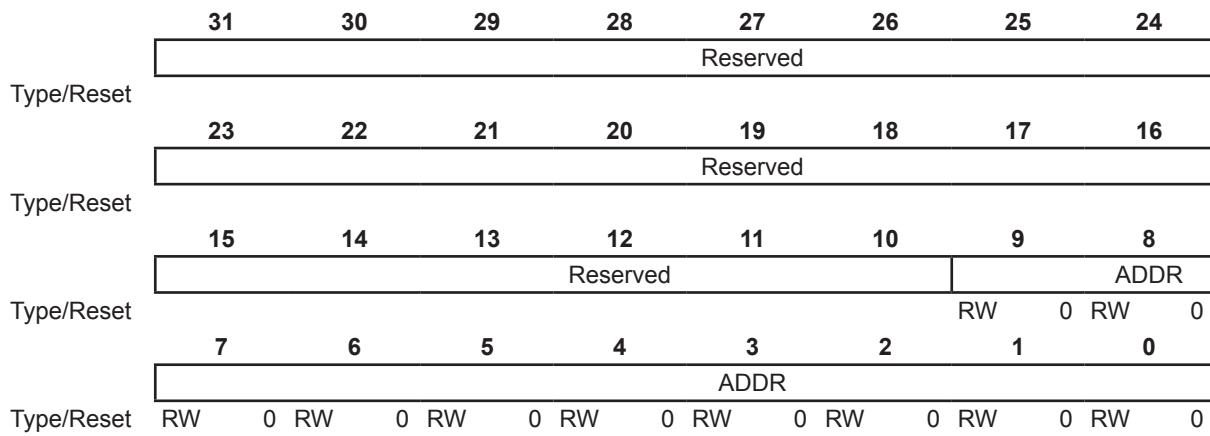
<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>
[8]	ARBLOSIE	Arbitration Loss Interrupt Enable Bit in the I <sup>2</sup> C multi-master mode 0: Interrupt disabled 1: Interrupt enabled When the I2CEN bit in the I2CCR register is cleared to 0, this bit is cleared to 0 by hardware.
[3]	GCSIE	General Call Slave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

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved		TXNRX	MASTER	BUSBUSY	RXBF	TXDE	RXDNE	
Type/Reset	RO		0 RO	0 RO	0 RO	0 RO	0 RO	0	
Type/Reset	15	14	13	12	11	10	9	8	
Type/Reset	Reserved			TOUTF	BUSERR	RXNACK	ARBLOS		
Type/Reset	WC		0 WC	0 WC	0 WC	0 WC	0 WC	0	
Type/Reset	7	6	5	4	3	2	1	0	
Type/Reset	Reserved			GCS	ADRS	STO	STA		
	RC		0 RC	0 RC	0 RC	0 RC	0 RC	0	

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 in Transmitter Mode</p> <ul style="list-style-type: none"> <li>0: Data register I2CDR not empty</li> <li>1: Data register I2CDR empty</li> </ul> <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> <ul style="list-style-type: none"> <li>0: Data register I2CDR empty</li> <li>1: Data register I2CDR not empty</li> </ul> <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> <ul style="list-style-type: none"> <li>0: No timeout counter underflow occurred</li> <li>1: Timeout counter underflow occurred</li> </ul> <p>Writing “1” to this bit will clear the TOUTF flag.</p>
[10]	BUSERR	<p>Bus Error Flag</p> <ul style="list-style-type: none"> <li>0: No bus error has occurs</li> <li>1: Bus error has occurred</li> </ul> <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> <ul style="list-style-type: none"> <li>0: Acknowledge is returned from receiver</li> <li>1: Not Acknowledge is returned from receiver</li> </ul> <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> <ul style="list-style-type: none"> <li>0: No arbitration loss is detected</li> <li>1: Bit arbitration loss is detected</li> </ul> <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> <ul style="list-style-type: none"> <li>0: No general call slave occurs</li> <li>1: I<sup>2</sup>C interface is addressed by a general call command</li> </ul> <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>

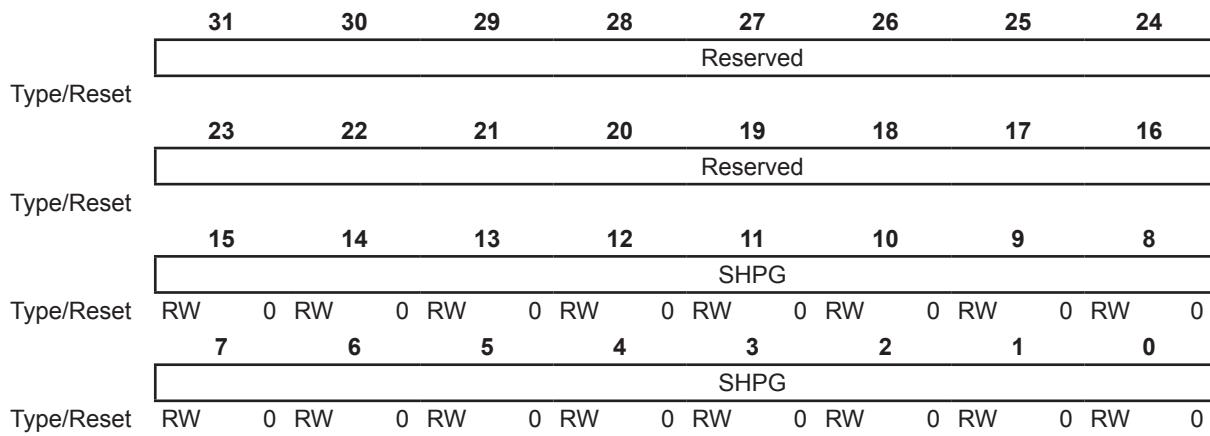
Bits	Field	Descriptions
[2]	ADRS	<p>Address Transmit (master mode) / Address Receive (slave mode) Flag</p> <p>Address Sent in Master Mode</p> <p>0: Address frame has not been transmitted 1: Address frame has been transmitted</p> <p>For the 7-bit addressing mode, this bit is set after the master device 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, and d value depends on the setting of SEQ_FILTER in the I<sup>2</sup>C Control Register (I2CCR).</p> <ul style="list-style-type: none"> <li>If SEQ_FILTER=00, d=6</li> <li>If SEQ_FILTER=01, d=8</li> <li>If SEQ_FILTER=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
Reserved								
Type/Reset	23	22	21	20	19	18	17	16
Reserved								
Type/Reset	15	14	13	12	11	10	9	8
SLPG								
Type/Reset	RW	0	RW	0	RW	0	RW	0
Type/Reset	7	6	5	4	3	2	1	0
SLPG								
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>SCL_{LOW} = T_{PCLK} \times (SLPG + d)</math>          where <math>T_{PCLK}</math> is the APB bus peripheral clock (PCLK) period, and d value depends on the setting of SEQ_FILTER in the I<sup>2</sup>C Control Register (I2CCR).</p> <ul style="list-style-type: none"> <li>If SEQ_FILTER=00, d=6</li> <li>If SEQ_FILTER=01, d=8</li> <li>If SEQ_FILTER=10 or 11, d=9</li> </ul>

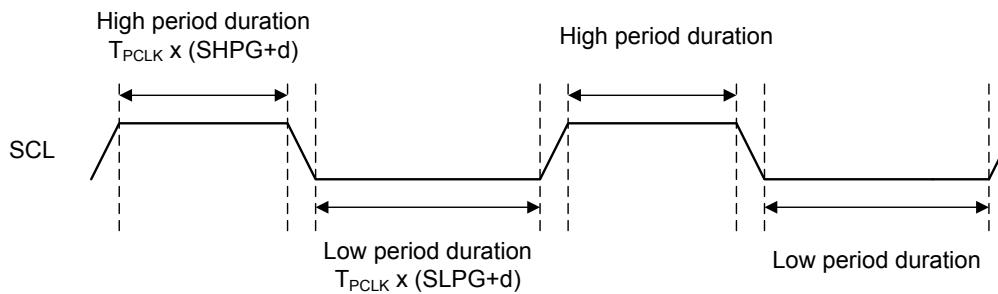


Figure 137. SCL Timing Diagram

Table 46. I<sup>2</sup>C Clock Setting Example

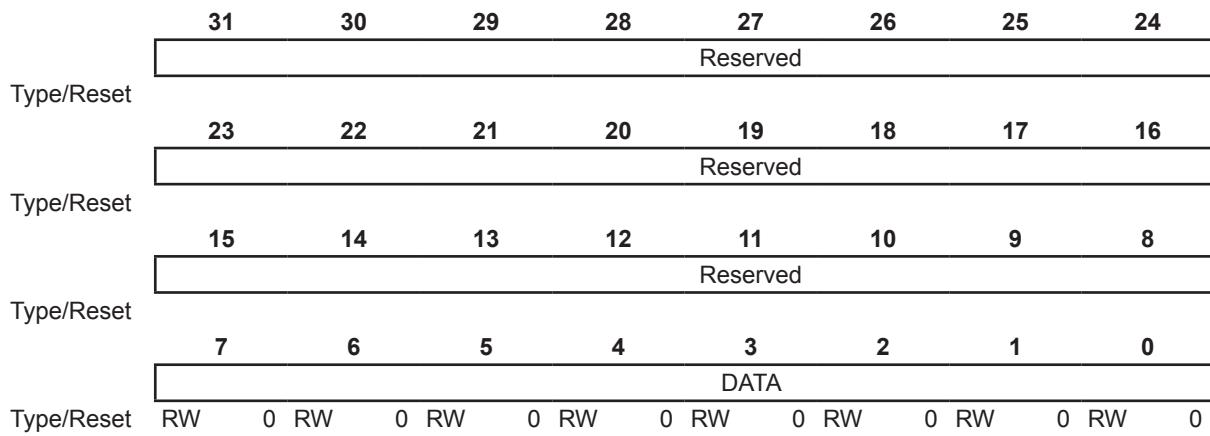
I <sup>2</sup> C Clock	$TSCL = TPCLK \times [ (SHPG + d) + (SLPG + d) ]$ (where d = 6) SHPG + SLPG value at PCLK			
	8MHz	24MHz	48MHz	72MHz
100 kHz (Standard Mode)	68	228	468	708
400 kHz (Fast Mode)	8	48	108	168
1 MHz (Fast Mode Plus)	x	12	36	60

## 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



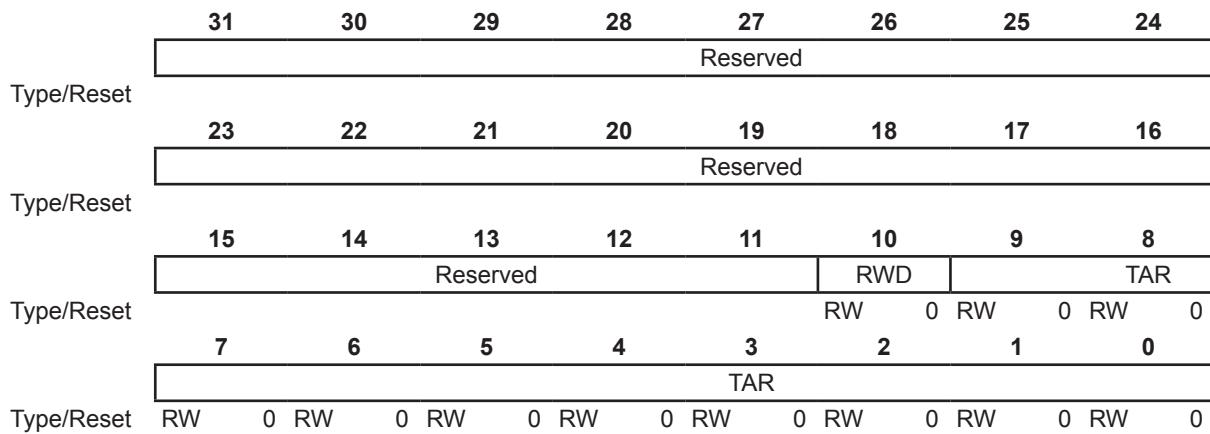
Bits	Field	Descriptions
[7:0]	DATA	<p>I<sup>2</sup>C Data Register</p> <p>For the transmitter mode, a data byte which is transmitted to a slave device can be assigned to these bits. The TXDE flag is cleared if the application software assigns new data to the I2CDR register. For the receiver mode, a data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

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

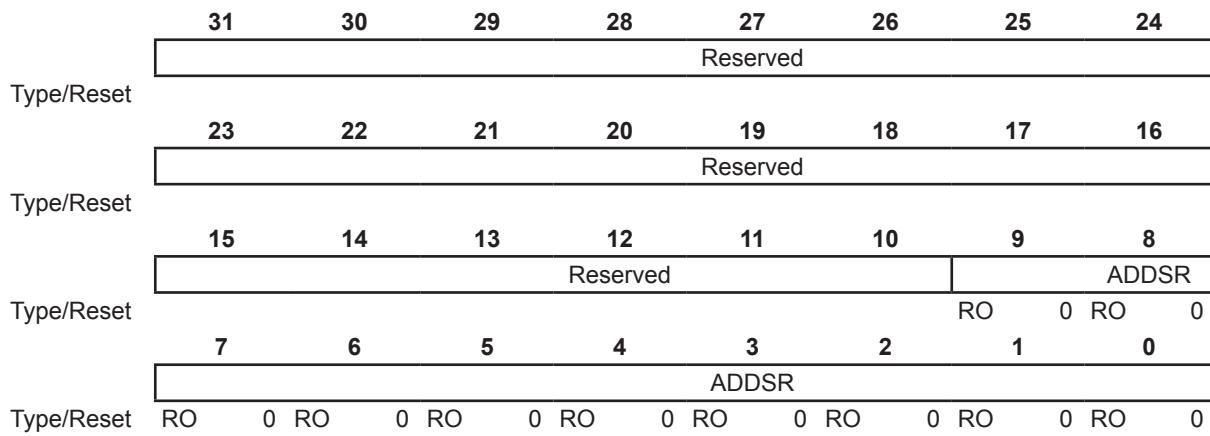
Bits	Field	Descriptions
[9:0]	ADDMR	<p>Address Mask Control Bit</p> <p>The ADDMR[i] is used to specify whether the i<sup>th</sup> bit of the ADDR in the I2CADDR register is masked and is compared with the received address frame or not on the I<sup>2</sup>C bus. The register is only used for the I<sup>2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



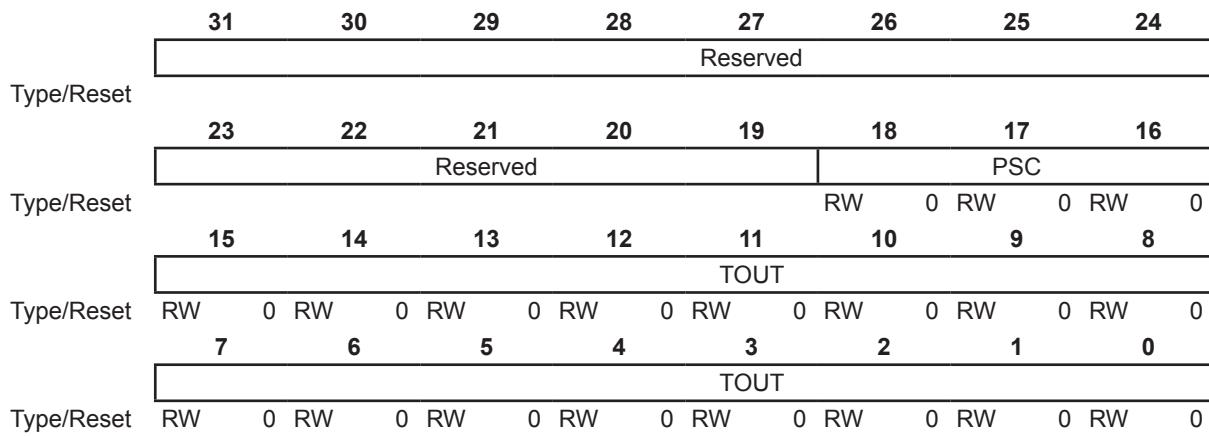
Bits	Field	Descriptions
[9:0]	ADDSR	Address Snoop Once the I2CEN bit is enabled, the calling address value on the I <sup>2</sup> C bus will automatically be loaded into this ADDSR field.

## 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><b>I<sup>2</sup>C Time-out Counter Prescaler Selection</b></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> <math>PSC=0 \rightarrow f_{I2CTO} = f_{PCLK} / 2^0 = f_{PCLK}</math>  <math>PSC=1 \rightarrow f_{I2CTO} = f_{PCLK} / 2^1 = f_{PCLK} / 2</math>  <math>PSC=2 \rightarrow f_{I2CTO} = f_{PCLK} / 2^2 = f_{PCLK} / 4</math>  ... </p> <p> <math>PSC=7 \rightarrow f_{I2CTO} = f_{PCLK} / 2^7 = f_{PCLK} / 128</math> </p>
[15:0]	TOUT	<p><b>I<sup>2</sup>C Timeout Counter Preload Value</b></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 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. The mode fault detection provides a capability for multi-master applications.

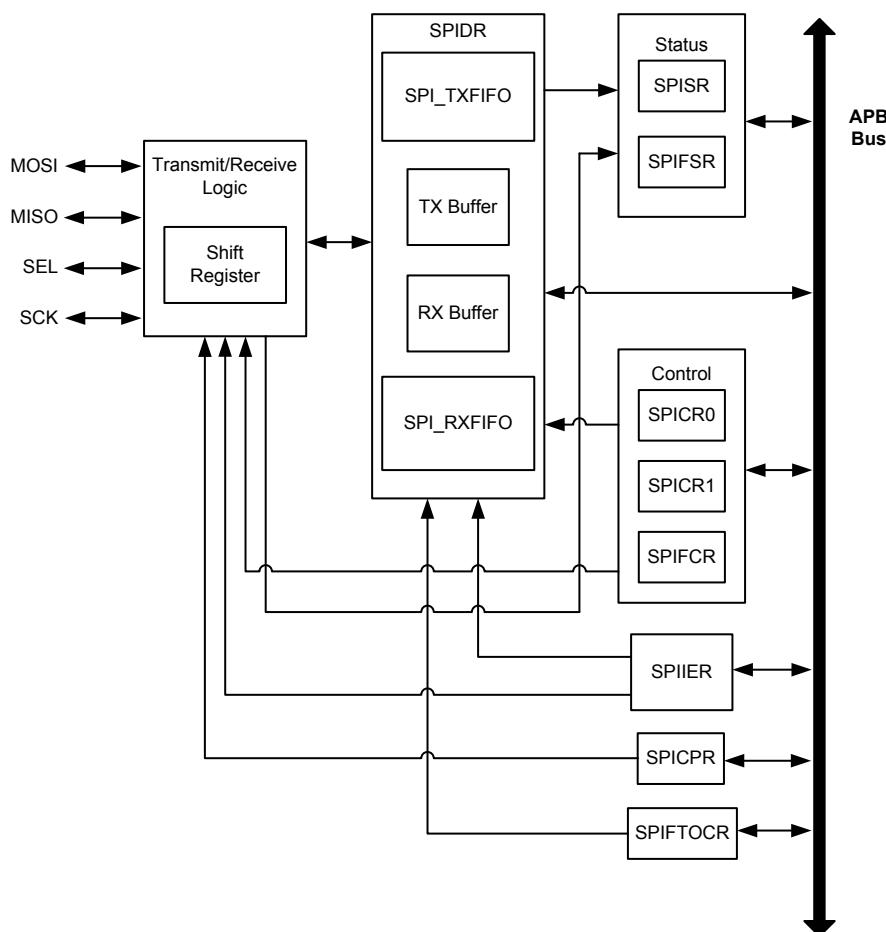


Figure 138. SPI Block Diagram

## 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 from 1 bit 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 serial NOR Flash
- Four error flags with individual interrupt
  - Read overrun
  - Write collision
  - Mode fault
  - Slave abort
- Support PDMA interface

## Function Descriptions

### Master Mode

Each data frame can range from 1 to 16 bits in data length. The first bit of the transmitted data can be either an MSB or LSB, determined by the FIRSTBIT bit in the SPICR1 register. The SPI module is configured as a master or a slave by setting the MODE bit in the SPICR1 register. When the MODE bit is set, the SPI module is configured as a master and will generate the serial clock on the SCK pin. The data stream will transmit data in the shift register to the MOSI pin on the serial clock edge. The SEL pin is active during the full data transmission. When the SELAP bit in the SPICR1 register is set, the SEL pin is active high during the complete data transactions. When the SELM bit in the SPICR1 register is set, the SEL pin will be driven by the hardware automatically and the time interval between the active SEL edge and the first edge of SCK is equal to half a SCK period.

### Slave Mode

In the slave mode, the SCK pin acts as an input pin and the serial clock will be derived from the external master device. The SEL pin also acts as an input. When the SELAP bit is cleared to 0, the SEL signal is active low during the full data stream reception. When the SELAP bit is set to 1, the SEL signal will be active high during the full data stream reception.

Note: For the slave mode, the APB clock, known as  $f_{PCLK}$ , must be at least 3 times faster than the external SCK clock input frequency.

## SPI Serial Frame Format

The SPI interface format is based on the Clock Polarity, CPOL, and the Clock Phase, CPHA, configurations.

- Clock Polarity Bit – CPOL

When the Clock Polarity bit is cleared to 0, the SCK line idle state is LOW. When the Clock Polarity bit is set to 1, the SCK line idle state is HIGH.

- Clock Phase Bit – CPHA

When the Clock Phase bit is cleared to 0, the data is sampled on the first SCK clock transition.

When the Clock Phase bit is set to 1, the data is sampled on the second SCK clock transition.

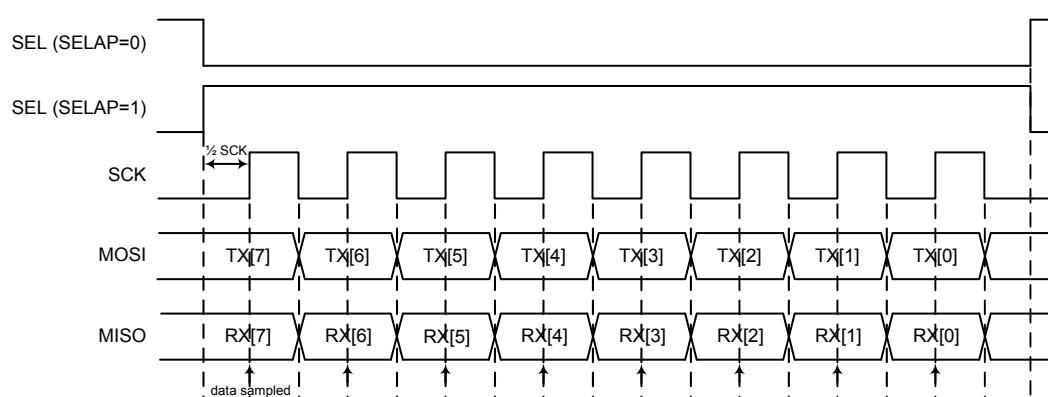
There are four formats contained in the SPI interface. Table 1 shows how to configure these formats by setting the FORMAT field in the SPICR1 register.

**Table 47. SPI Interface Format Setup**

FORMAT [2:0]	CPOL	CPHA
001	0	0
010	0	1
110	1	0
101	1	1
Others	Reserved	

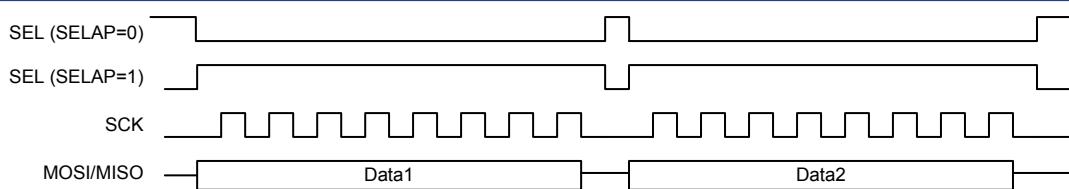
### CPOL = 0, CPHA = 0

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 139 shows the single byte data transfer timing diagram of this format.



**Figure 139. SPI Single Byte Transfer Timing Diagram – CPOL = 0, CPHA = 0**

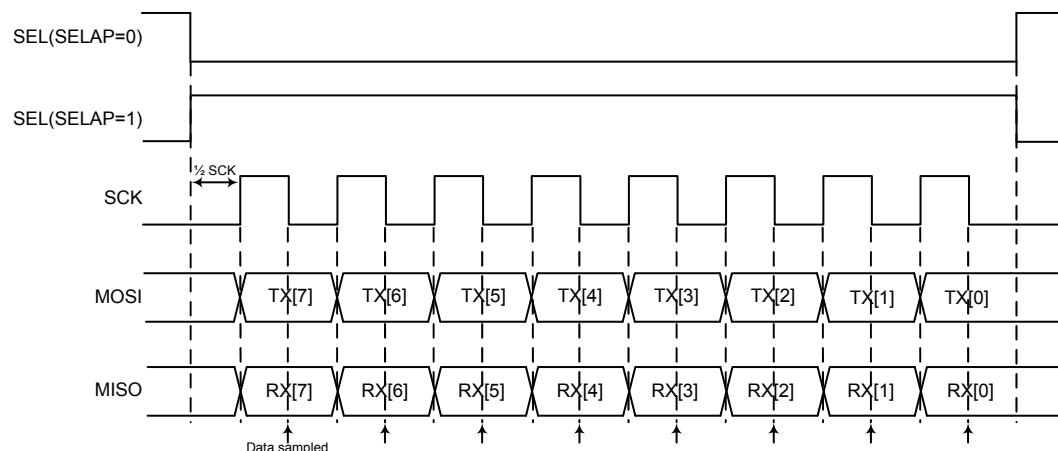
Figure 140 shows the continuous data transfer timing diagram of this format. Note that the SEL signal must change to an inactive level between each data frame.



**Figure 140. SPI Continuous Data Transfer Timing Diagram – CPOL = 0, CPHA = 0**

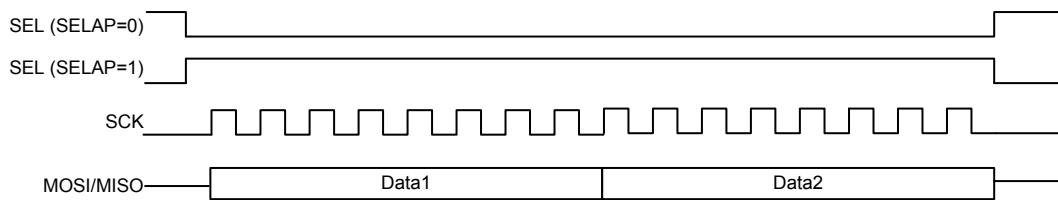
#### CPOL = 0, CPHA = 1

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 141 shows the single data byte transfer timing.



**Figure 141. SPI Single Byte Transfer Timing Diagram – CPOL = 0, CPHA = 1**

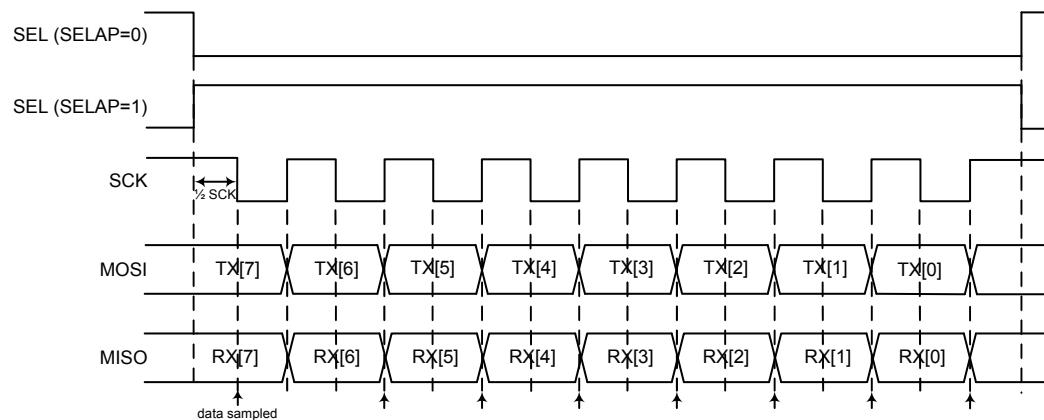
Figure 142 shows the continuous data transfer diagram timing. Note that the SEL signal must remain active until the last data transfer has completed.



**Figure 142. SPI Continuous Transfer Timing Diagram – CPOL = 0, CPHA = 1**

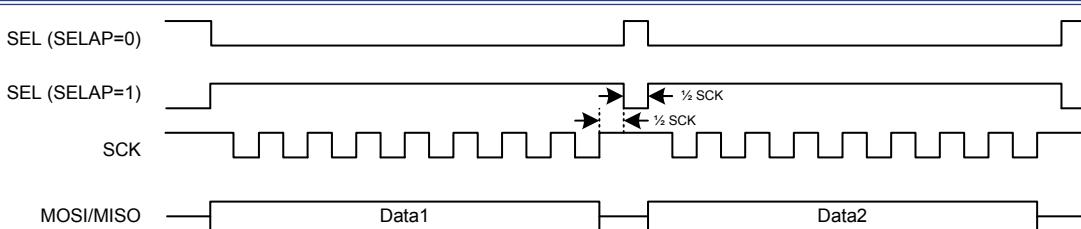
#### CPOL = 1, CPHA = 0

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 143 shows the single byte transfer timing of this format.



**Figure 143. SPI Single Byte Transfer Timing Diagram – CPOL = 1, CPHA = 0**

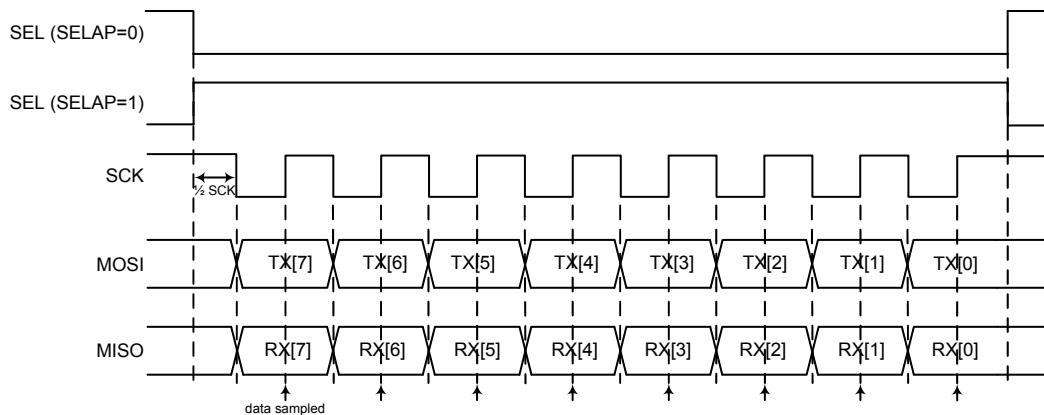
Figure 144 shows the continuous data transfer timing of this format. Note that the SEL signal must change to an inactive level between each data frame.



**Figure 144. SPI Continuous Transfer Timing Diagram – CPOL = 1, CPHA = 0**

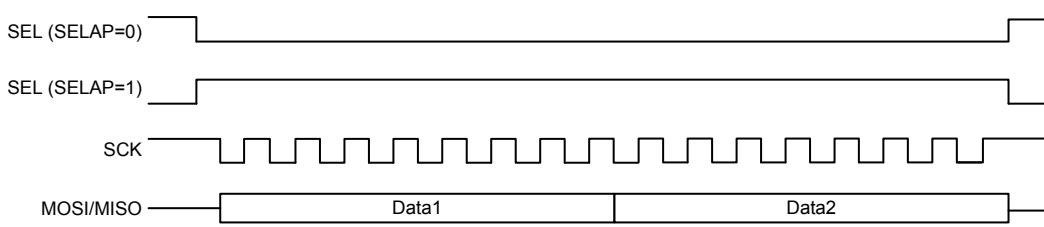
#### CPOL = 1, CPHA = 1

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 145 shows the single byte transfer timing of this format.



**Figure 145. SPI Single Byte Transfer Timing Diagram – CPOL = 1, CPHA = 1**

Figure 146 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 146. 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 the all of 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.

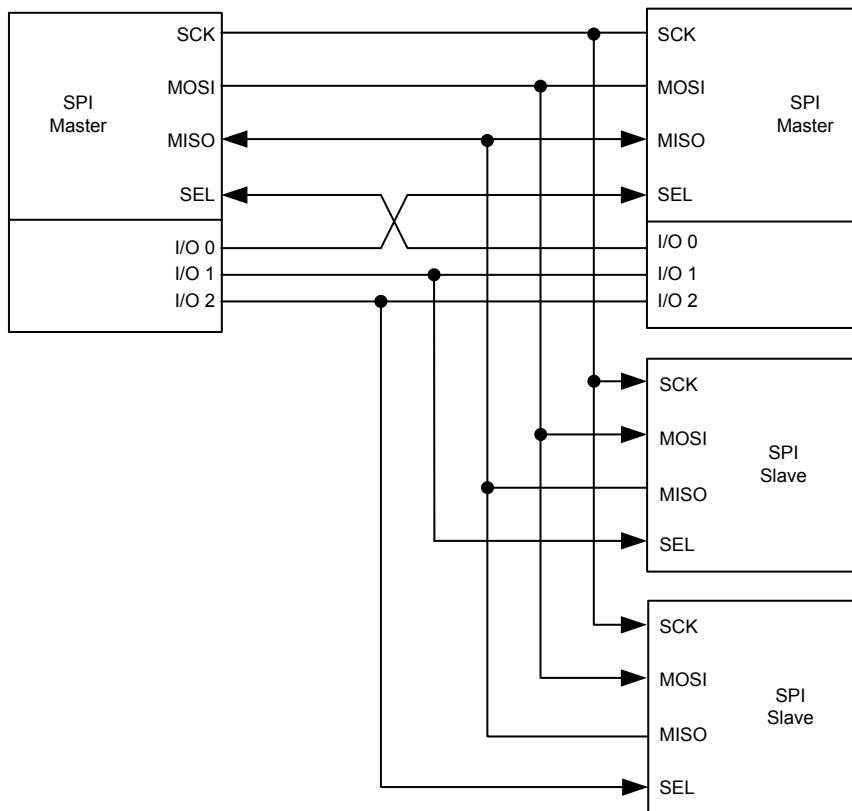


Figure 147. SPI Multi-Master Slave Environment

Table 48. 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 49. 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.

### PDMA Interface

The PDMA interface is integrated in the SPI module. The PDMA function can be enabled by setting the TXDMAE or RXDMAE bit to 1 in the transmitter or receiver mode respectively. When the transmit buffer empty flag, TXBE, is asserted and the TXDMAE bit is set to 1, the PMDA function will be activated to move data from the memory location that users designated into the SPI data register or the TX FIFO until the TXBE flag is cleared to 0. The TXBE flag will be asserted when the transmit buffer is empty in the non-FIFO mode or the data contained in the TX FIFO is equal to or less than the level defined by the TXFTLS field in the FIFO mode.

Similarly, when the receive buffer not empty flag, RXBNE, is asserted and the RXDMAE bit is set to 1, the PDMA function will be activated to move data from the SPI data register or the RX FIFO to the memory location that users designated until the RXBNE flag is cleared to 0. The RXBNE flag will be asserted when the receive buffer is not empty in the non-FIFO mode or the data contained in the RX FIFO is equal to or greater than the level defined by the RXFTLS field in the FIFO mode.

For a mode detailed description on the PDMA configurations, refer to the PDMA chapter.

## Register Map

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

**Table 50. SPI Register Map**

Register	Offset	Description	Reset Value
<b>SPI0 Base Address = 0x4000_4000</b>			
<b>SPI1 Base Address = 0x4004_4000</b>			
SPICR0	0x000	SPI Control Register 0	0x0000_0000
SPICR1	0x004	SPI Control Register 1	0x0000_0000
SPIIER	0x008	SPI Interrupt Enable Register	0x0000_0000
SPICPR	0x00C	SPI Clock Prescaler Register	0x0000_0000
SPIDR	0x010	SPI Data Register	0x0000_0000
SPISR	0x014	SPI Status Register	0x0000_0003
SPIFCR	0x018	SPI FIFO Control Register	0x0000_0000
SPIFSR	0x01C	SPI FIFO Status Register	0x0000_0000
SPIFTOCR	0x020	SPI FIFO Time Out Counter Register	0x0000_0000

## Register Descriptions

### SPI Control Register 0 – SPICR0

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

Offset: 0x000

Reset value: 0x0000\_0000



Bits	Field	Descriptions
[15:12]	SELHT	Chip Select Hold Time 0x0: 1/2 SCK 0x1: 1 SCK 0x2: 3/2 SCK 0x3: 2 SCK ....
Note that SELHT is for master mode only.		

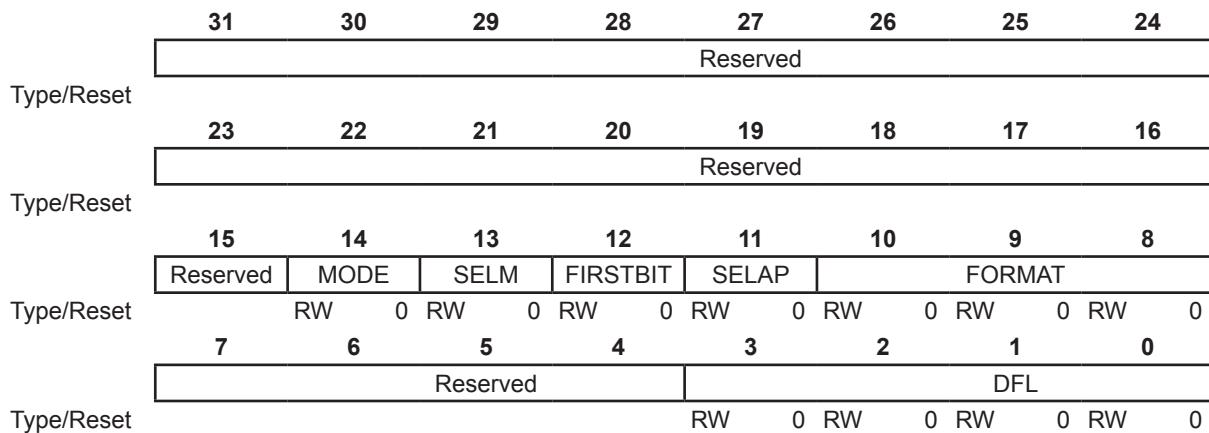
Bits	Field	Descriptions
[11:8]	GUADT	<p>Guard Time</p> <ul style="list-style-type: none"> <li>0x0:1 SCK</li> <li>0x1:2 SCK</li> <li>0x2: 3 SCK</li> </ul> <p>...</p> <p>Note that GUADT is for master mode only.</p>
[7]	GUADTEN	<p>Guard Time Enable</p> <ul style="list-style-type: none"> <li>0: Guard Time is 1/2 SCK</li> <li>1: When set this bit, Guard time can be controlled by GUADT</li> </ul> <p>Note that GUADTEN is for master mode only.</p>
[6]	DUALEN	<p>Dual Port Enable</p> <ul style="list-style-type: none"> <li>0: Dual port is disabled</li> <li>1: Dual port is enabled</li> </ul> <p>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.</p>
[4]	SSELC	<p>Software Slave Select Control</p> <ul style="list-style-type: none"> <li>0: Set the SEL output to an inactive state</li> <li>1: Set the SEL output to an active state</li> </ul> <p>The application Software can setup the SEL output to an active or inactive state by configuring the SSELC bit. The active level is configured by the SELAP bit in the SPICR1 register. Note that the SSELC bit is only available when the SELOEN bit is set to 1 for enabling the SEL output meanwhile the SELM bit is cleared to 0 for controlling the SEL signal by software. Otherwise, the SSELC bit has no effect.</p>
[3]	SELOEN	<p>Slave Select Output Enable</p> <ul style="list-style-type: none"> <li>0: Set the SEL signal to the input mode for Multi-master mode</li> <li>1: Set the SEL signal to the output mode for slave select</li> </ul> <p>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</p>
[2]	RXDMAE	<p>RX PDMA request enable</p> <ul style="list-style-type: none"> <li>0: SPI RX path PDMA request disabled.</li> <li>1: SPI RX path PDMA request enabled.</li> </ul>
[1]	TXDMAE	<p>TX PDMA request enable</p> <ul style="list-style-type: none"> <li>0: SPI TX path PDMA request disabled.</li> <li>1: SPI TX path PDMA request enabled.</li> </ul>
[0]	SPIEN	<p>SPI Enable</p> <ul style="list-style-type: none"> <li>0: SPI interface is disabled</li> <li>1: SPI interface is enabled</li> </ul>

## 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 These three bits are used to determine the data transfer format of the SPI interface</p> <table border="1"> <thead> <tr> <th>FORMAT [2:0]</th><th>CPOL</th><th>CPHA</th></tr> </thead> <tbody> <tr> <td>001</td><td>0</td><td>0</td></tr> <tr> <td>010</td><td>0</td><td>1</td></tr> <tr> <td>110</td><td>1</td><td>0</td></tr> <tr> <td>101</td><td>1</td><td>1</td></tr> <tr> <td>Others</td><td>Reserved</td><td></td></tr> </tbody> </table> <p>CPOL: Clock Polarity 0: SCK Idle state is low 1: SCK Idle state is high CPHA: Clock Phase 0: Data is captured on the first SCK clock edge 1: Data is captured on the second SCK clock edge</p>	FORMAT [2:0]	CPOL	CPHA	001	0	0	010	0	1	110	1	0	101	1	1	Others	Reserved	
FORMAT [2:0]	CPOL	CPHA																		
001	0	0																		
010	0	1																		
110	1	0																		
101	1	1																		
Others	Reserved																			
[3:0]	DFL	<p>Data Frame Length Selects the data transfer frame from 1 bit to 16 bits.</p> <ul style="list-style-type: none"> <li>0x1: 1 bit</li> <li>0x2: 2 bits</li> <li>...</li> <li>0xF: 15 bits</li> <li>0x0: 16 bits</li> </ul>																		

## 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

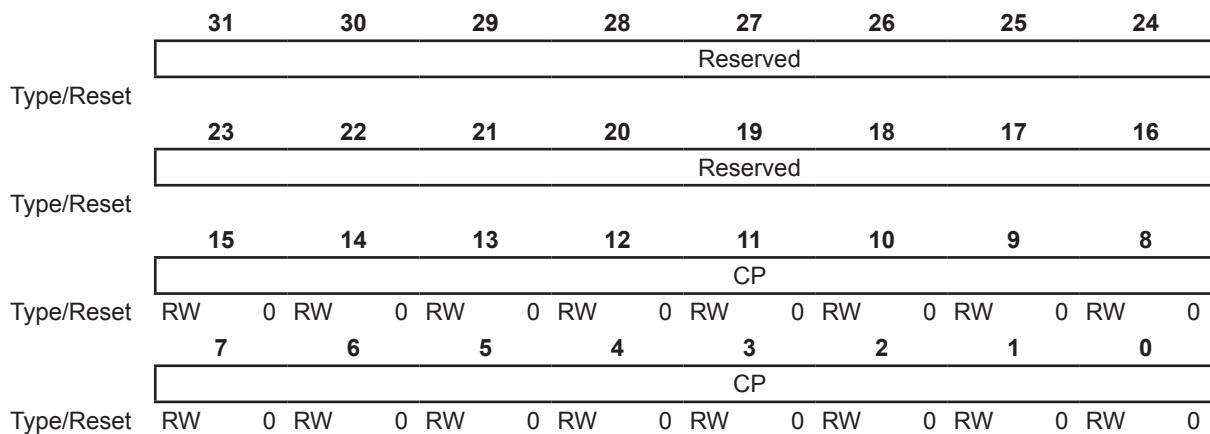
Bits	Field	Descriptions
[0]	TXBEIEN	<p>Tx Buffer Empty Interrupt Enable            0: Disable            1: Enable</p> <p>The TX buffer empty interrupt request will be generated when the TXBE flag and the TXBEIEN bit are set. In the FIFO mode, the interrupt request being generated depends upon the TX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

### SPI Clock Prescaler Register – SPICPR

This register specifies the SPI clock prescaler ratio.

Offset: 0x00C

Reset value: 0x0000\_0000



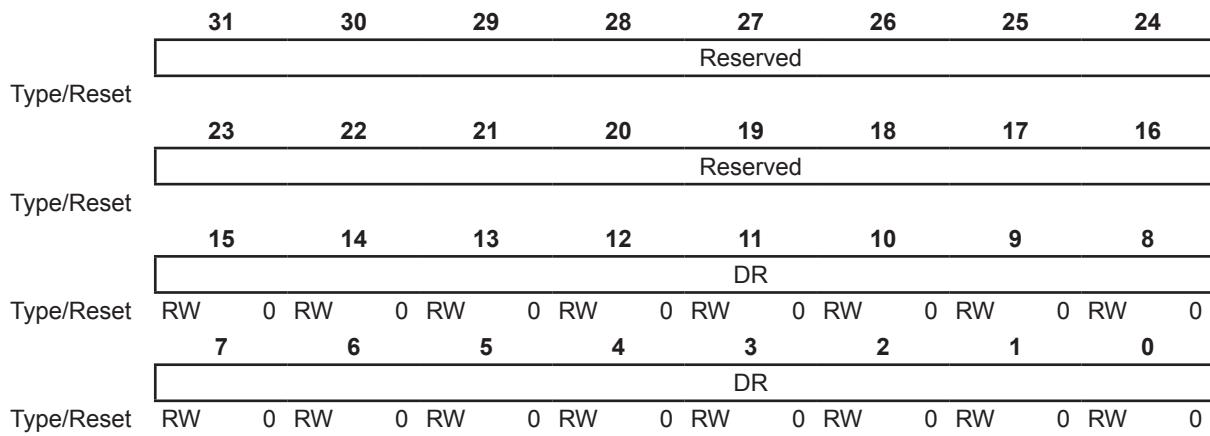
Bits	Field	Descriptions
[15:0]	CP	<p>SPI Clock Prescaler</p> <p>The SPI clock (SCK) is determined by the following equation:  <math>f_{SCK} = f_{PCLK} / (2 * (CP + 1))</math>, where the CP ranges is from 0 to 65535</p> <p>Note: For the SPI slave mode, the system clock (<math>f_{PCLK}</math>) must be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

## SPI Data Register – SPIDR

This register stores the SPI received or transmitted Data.

Offset: 0x010

Reset value: 0x0000\_0000



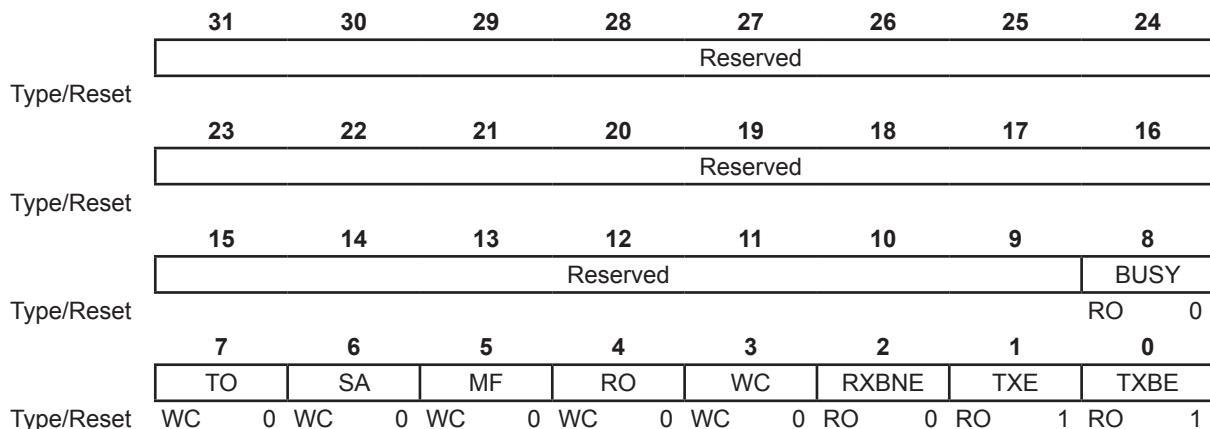
Bits	Field	Descriptions
[15:0]	DR	<p>Data Register</p> <p>The SPI data register is used to store the serial bus transmitted or received data. In the non-FIFO mode, writing data into the SPI data register will also load the data into the data transmission buffer, known as the TX buffer. Reading data from 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
[8]	BUSY	SPI Busy flag 0: SPI not busy 1: SPI busy In the master mode, this flag is reset when the TX buffer and TX shift register are both empty and is set when the TX buffer or the TX shift register are not empty. In the slave mode, this flag is set when SEL changes to an active level and is reset when SEL changes to an inactive level.
[7]	TO	Time Out flag 0: No Rx FIFO time out 1: Rx FIFO time out has occurred. Write 1 to clear it. Once the time out counter value is equal to the TOC field setting in the SPIFTOCR register, the time out flag will be set and an interrupt will be generated if the TOIEN bit in the SPIIER register is enabled. This bit is cleared by writing 1 Note: This Time Out flag function is only available in the SPI FIFO mode.
[6]	SA	Slave Abort flag 0: No slave abort 1: Slave abort has occurred. This bit is set by hardware and cleared by writing 1.
[5]	MF	Mode Fault flag 0: No mode fault 1: Mode fault has occurred This bit is set by hardware and cleared by writing 1.
[4]	RO	Read Overrun flag 0: No read overrun 1: Read overrun has occurred. This bit is set by hardware and cleared by writing 1.

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

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					FIFOEN	Reserved			
Type/Reset	RW	0	RW	0	RW	0	RW	0		
	RXFTLS					TXFTLS				
Type/Reset	RW	0	RW	0	RW	0	RW	0		

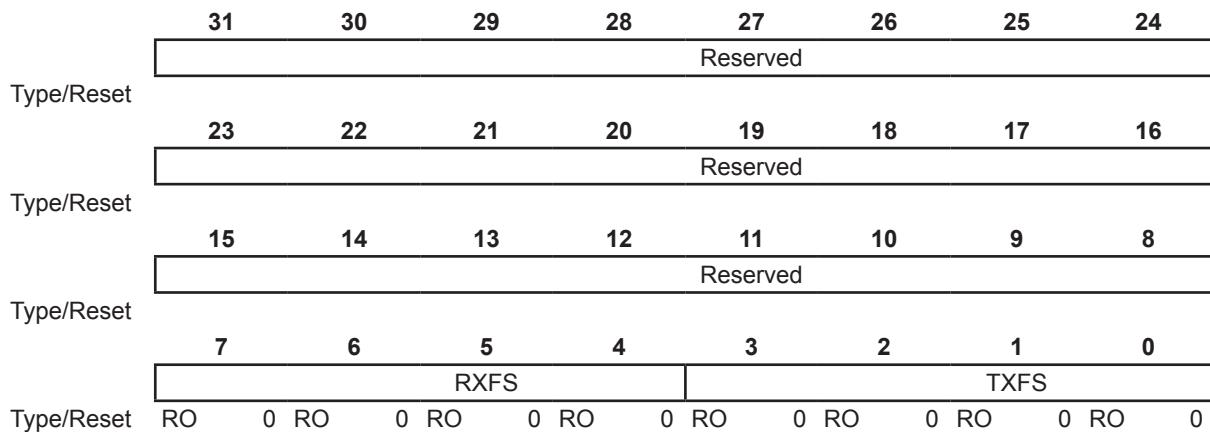
Bits	Field	Descriptions
[10]	FIFOEN	<p>FIFO Enable</p> <p>0: FIFO disable 1: FIFO enable</p> <p>This bit cannot be set or reset when the SPI interface is in transmitting.</p>
[7:4]	RXFTLS	<p>Rx FIFO Trigger Level Select</p> <p>0000: Trigger level is 0 0001: Trigger level is 1 ... 1000: Trigger level is 8 Others: Reserved</p> <p>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</p>
[3:0]	TXFTLS	<p>Tx FIFO Trigger Level Select</p> <p>0000: Trigger level is 0 0001: Trigger level is 1 ... 1000: Trigger level is 8 Others: Reserved</p> <p>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.</p>

## 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

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

Bits	Field	Descriptions
[15:0]	TOC	<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 a 16-byte transmitter FIFO, TX\_FIFO, and a 16-byte receiver FIFO, RX\_FIFO.

Software can detect a USART error status by reading Line Status Register, LSR. The status includes the type and 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 CK\_AHB to produce a clock for the USART transmitter and receiver.

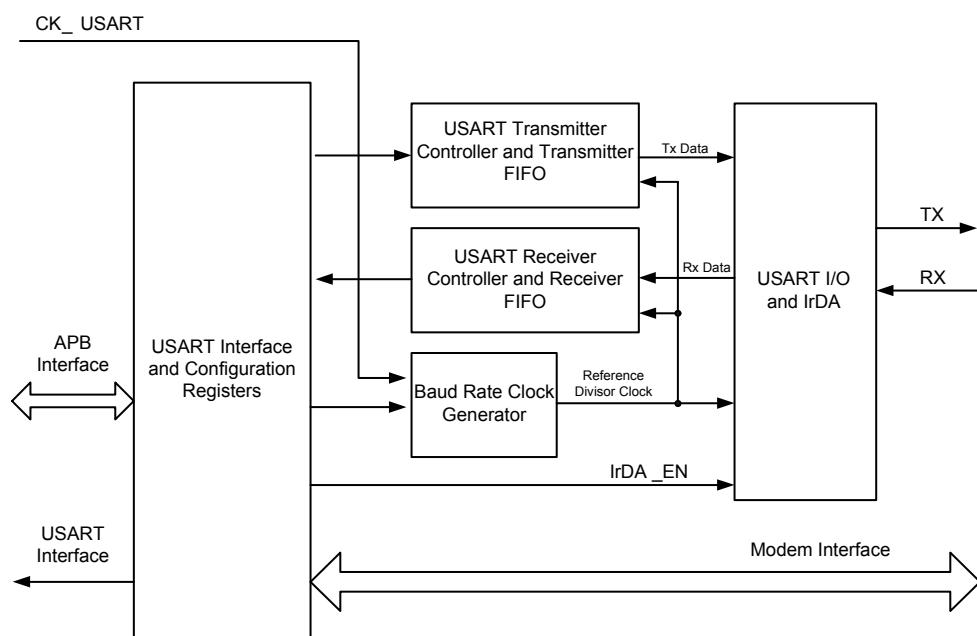


Figure 148. USART Block Diagram

## Features

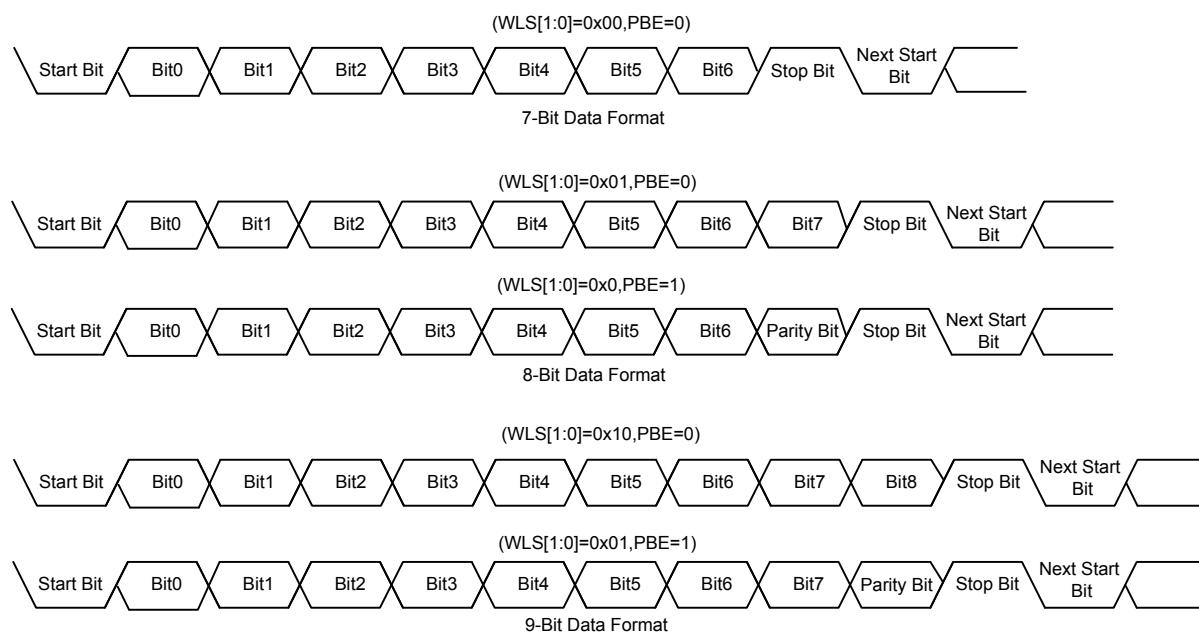
- Supports both asynchronous and clocked synchronous serial communication modes
- Full Duplex Communication Capability
- IrDA SIR encoder and decoder
  - Support of normal 3/16 bit duration and low-power (1.41 ~ 2.23  $\mu$ s) durations
- Supports RS485 mode with output enable
- Auto hardware flow control mode – RTS, CTS
- Full modem function
- 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: 16 × 9 bits (max 9 data bits)
  - Transmit FIFO: 16 × 9 bits (max 9 data bits)
- Supports PDMA Interface

## Function Descriptions

### Serial Data Format

The USART module performs a parallel-to-serial conversion on data that is read from the Transmitter FIFO 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 start and stop bits are both 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 Receiver FIFO. 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 149. 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 system clock connected to the USART module while the BRD range is from 16 to 65535.

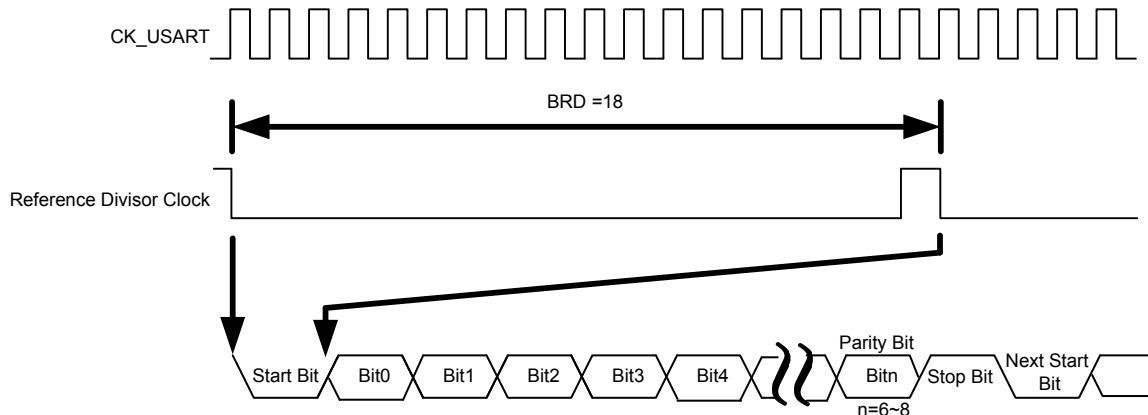


Figure 150. USART Clock CK\_USART and Data Frame Timing

Table 51. Baud Rate Error calculation – CK\_USART = 72 MHz

No	Baud rate		CK_USART = 72 MHz		
	Kbps	Actual	BRD/16	BRD	Error rate
1	2.4	2.4	1875	30000	0%
2	9.6	9.6	468.75	7500	0%
3	19.2	19.2	234.375	3750	0%
4	57.6	57.6	78.125	1250	0%
5	115.2	115.2	39.0625	625	0%
6	230.4	230.769	19.5	312	0.16%
7	460.8	461.538	9.75	156	0.16%
8	921.6	923.076	4.875	78	0.16%
9	2250	2250	2	32	0%
10	4500	4500	1	16	0%

## 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 data ‘0’ as a ‘low’ pulse and a data ‘1’ as a ‘low’ level while the Receiver specifies a data ‘0’ as a ‘low’ pulse and a data ‘1’ as ‘high’ level in the IrDA mode. The IrDA mode provides two operation modes, one is the normal mode, and the other is the low-power mode.

For the IrDA normal mode, the width of each transmitted pulse generated by the transmitter modulator is specified as 3/16th 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 UR\_RX pin, the debounce filter counter starts to count down, driven by the CK\_USART clock. If a rising edge is detected on the UR\_RX pin, the counter stops counting and is reloaded with the IrDAPSC value. When a low pulse falling edge on the UR\_RX 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 UR\_RX pin but no rising edge is detected before the debounce counter reaches 0, then the input on the USART receiver pin 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.

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.

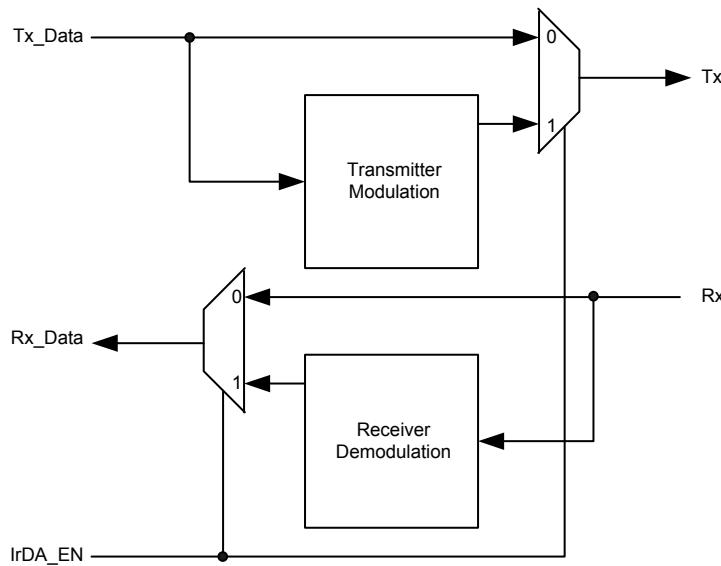
$$T_{IrDA\_L} = 3 \times IrDAPSC / CK\_USART$$

Note:  $T_{IrDA\_L}$  is transmitted pulse width in the low power mode.

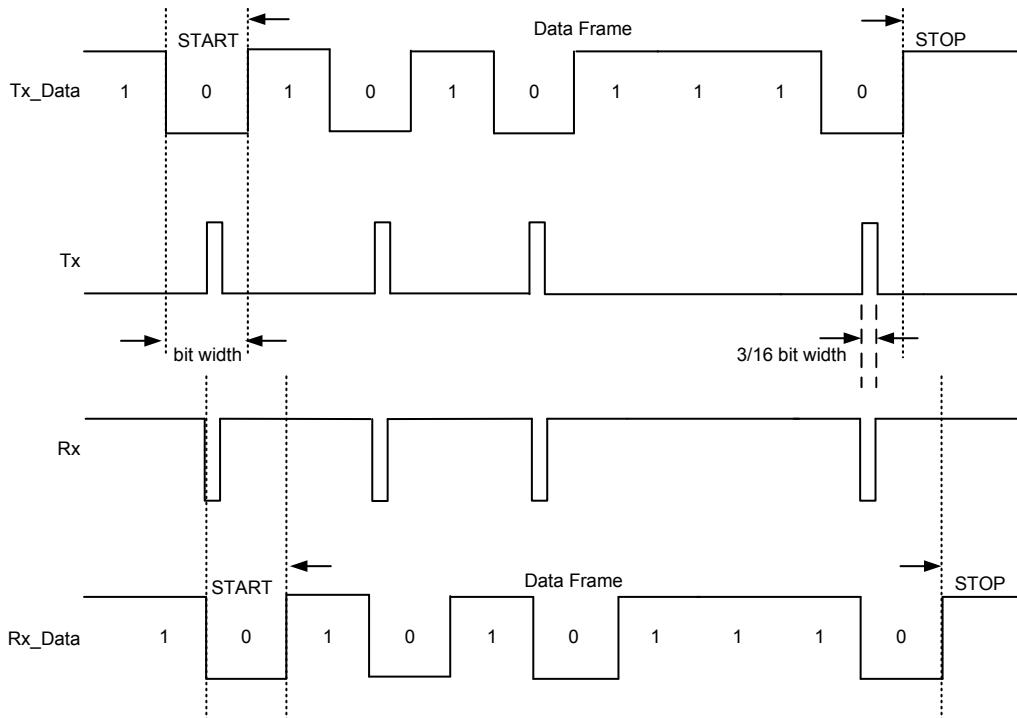
The IrDAPSC field in the IrDACR prescaler value in the IrDA Control Register IrDACR.

The debounce behavior in the IrDA low-power receiving mode is similar to the IrDA normal mode. For glitch detection, the low pulse of which the pulse width is shorter than  $1 \times (IrDAPSC / CK\_USART)$  should be discarded in the IrDA receiver demodulation. A valid low data is accepted if its low pulse width is greater than  $2 \times (IrDAPSC / CK\_USART)$  duration.

The IrDA physical layer specification specifies a minimum delay with a value of 10 ms between the transmission and reception and this IrDA receiver set-up time should be managed by the software.



**Figure 151. USART I/O and IrDA Block Diagram**



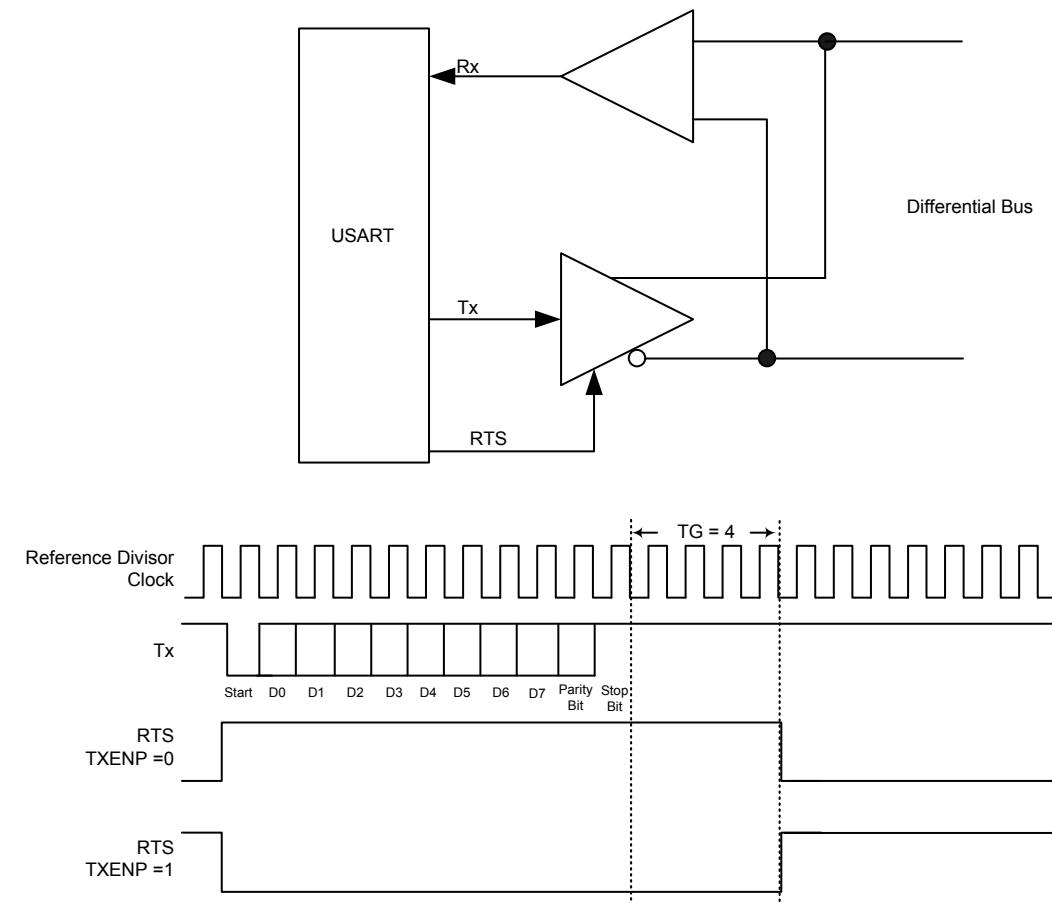
**Figure 152. IrDA Modulation and Demodulation**

## RS485 Mode

The data on RS485 interface is 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 UR\_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 module operates in the RS485 mode.

### RS485 Auto Direction Mode – AUD

When the RS485 interface is configured as a master transmitter, it will operate in the Auto Direction Mode, AUD. In the AUD mode the polarity of the UR\_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 153. RS485 Interface and Waveform**

### RS485 Normal Multi-drop Operation Mode (NMM)

When the RX485 interface 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 FCR 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 RSADDEF flag in the LSR register will be set and generate an interrupt if the RLSIE bit in the IER 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.

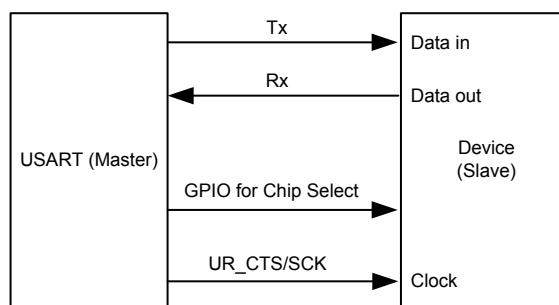
### RS485 Auto Address Detection Operation Mode (AAD)

Except in the Normal Multi-drop Operation Mode, the RS485 interface 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 Mode

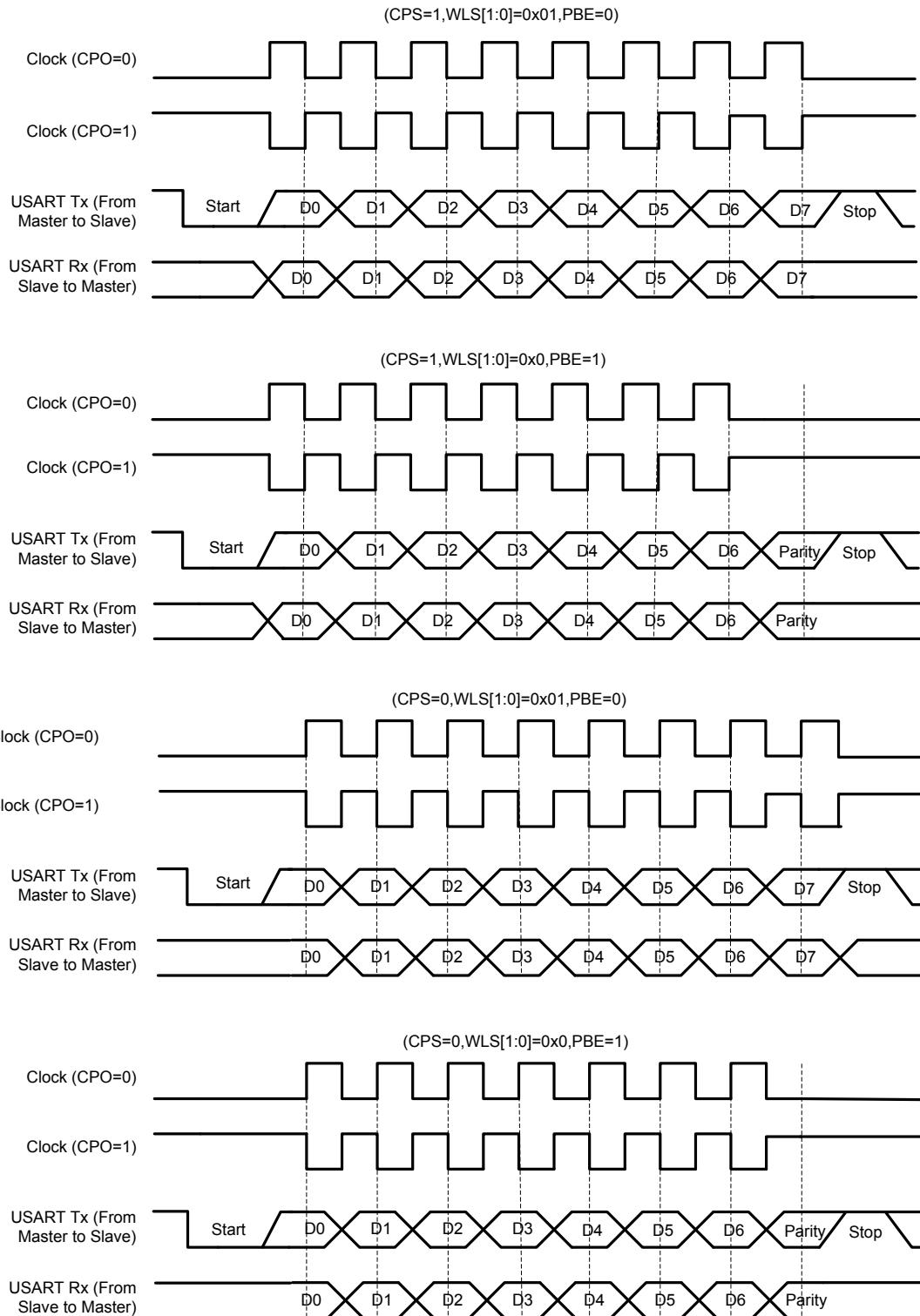
The data is transmitted in a full-duplex style in the USART Synchronous Master Mode, i.e., data transmission and reception both occur at the same time and only support master mode. The USART UR\_CTS/SCK pin is the synchronous USART transmitter clock output. In this mode, no clock pulses will be sent to the UR\_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 the accompanying diagram.

In the USART synchronous Mode, the UR\_CTS/SCK clock output pin is only used to transmit the data to slave device. If the transmission data register TBR, 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 UR\_RX pin. Otherwise the receiver will not obtain synchronous data if no data is transmitted.



**Figure 154. 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 clock is always an output.



**Figure 155. 8-bit Format USART Synchronous Waveform**

## Hardware Flow Control

The RS485 interface supports the hardware flow control function which is enabled by setting the HFCEN bit in the MODCR register to 1. 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.

### RTS Flow Control

In the RTS flow control, the UR\_RTS pin is active with a logic low state when the RX FIFO 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 RFTL field in the FCR register, the UR\_RTS pin is inactive with a logic high state. Figure 156 shows the example of RTS flow control.

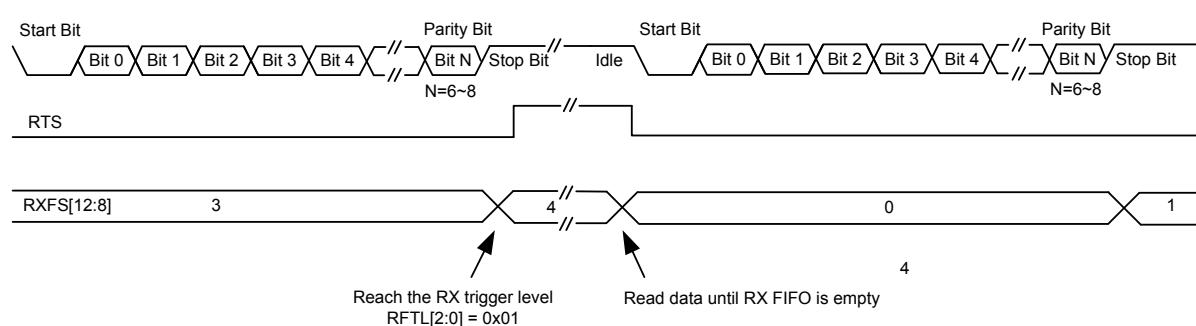


Figure 156. USART RTS Flow Control

### CTS Flow Control

If the hard flow control function is enabled, the URTXEN bit in the FCR register is controlled by the UR\_CTS input signal. If the UR\_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 UR\_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 UR\_CTS pin is forced to a logic high state during a data transmission, the current data transmission will be continued until the stop bit is completed. The Figure 157 shows an example of communication with CTS flow control.

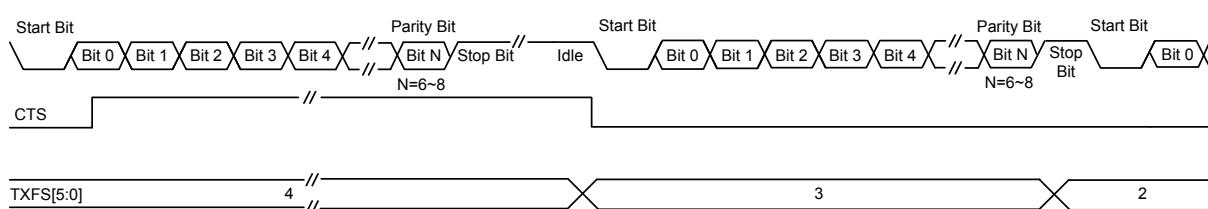


Figure 157. USART CTS Flow Control

## Interrupts and Status

The USART module can generate an interrupt when the following event occurs:

- Receiver Line Status Interrupt (Irpt\_RLSI): Occurrence of USART overrun error, parity error, framing error, or break events.
- Receiver FIFO Threshold Level Interrupt (Irpt\_RFTLI): When the FIFO received data amount has reached the specified threshold level.
- Receiver FIFO Time-out Interrupt (Irpt\_RTOI): When the USART receiver FIFO does not receive a new data package during the specified time-out interval.
- Transmitter FIFO Threshold Level Interrupt (Irpt\_TFTLI): When the data to be transmitted in the USART transmitter FIFO is less than the specified threshold level.
- MODEM Status Interrupt (Irpt\_MODSI): When the DCD, RI, DSR, or CTS bits states in the MODSR register have changed.

## PDMA Interface

The PDMA interface is integrated in the USART module. The PDMA function can be enabled by setting the TXDMAEN or RXDMAEN bit in the MDR register to 1 in the transmitter or receiver mode respectively. When the data to be transmitted in the USART transmitter FIFO is less than the TX FIFO threshold level specified by the TFTL field in the FCR register and the TXDMAEN bit is set to 1, the PDMA function will be activated to move data from a certain memory location into the USART TX FIFO.

Similarly, when the received data amount in the receiver FIFO is equal to the RX FIFO threshold level specified by the RFTL field in the FCR register and the RXDMAEN bit is set to 1, the PDMA function will be activated to move data from the USART RX FIFO to a specific memory location.

For a more detailed description on the PDMA configurations, refer to the PDMA chapter.

## Register Map

The following table shows the USART registers and reset values.

**Table 52. USART Register Map**

Register	Offset	Description	Reset Value
USART0 Base Address = 0x4000_0000			
USART1 Base Address = 0x4004_0000			
RBR	0x000	USART Receiver Buffer Register	0x0000_0000
TBR	0x000	USART Transmitter Buffer Register	0x0000_0000
IER	0x004	USART Interrupt Enable Register	0x0000_0000
IIR	0x008	USART Interrupt Identification Register	0x0000_0001
FCR	0x00C	USART FIFO Control Register	0x0000_0001
LCR	0x010	USART Line Control Register	0x0000_0000
MODCR	0x014	USART Modem Control Register	0x0000_0000
LSR	0x018	USART Line Status Register	0x0000_0060
MODSR	0x01C	USART Modem Status Register	0x0000_0000
TPR	0x020	USART Timing Parameter Register	0x0000_0000
MDR	0x024	USART Mode Register	0x0000_0000
IrDACR	0x028	USART IrDA Control Register	0x0000_0000
RS485CR	0x02C	USART RS485 Control Register	0x0000_0000
SYNCR	0x030	USART Synchronous Control Register	0x0000_0000
FSR	0x034	USART FIFO Status Register	0x0000_0000
DLR	0x038	USART Divider Latch Register	0x0000_0010
DEGTSTR	0x040	USART Debug/Test Register	0x0000_0000

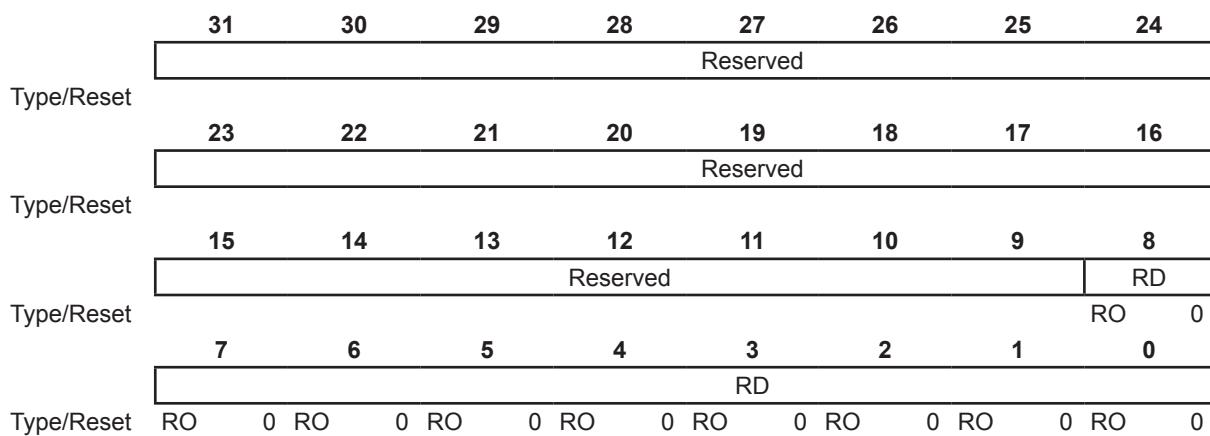
## Register Descriptions

### USART Receiver Buffer Register – RBR

The register is used to store the USART received data.

Offset: 0x000

Reset value: 0x0000\_0000



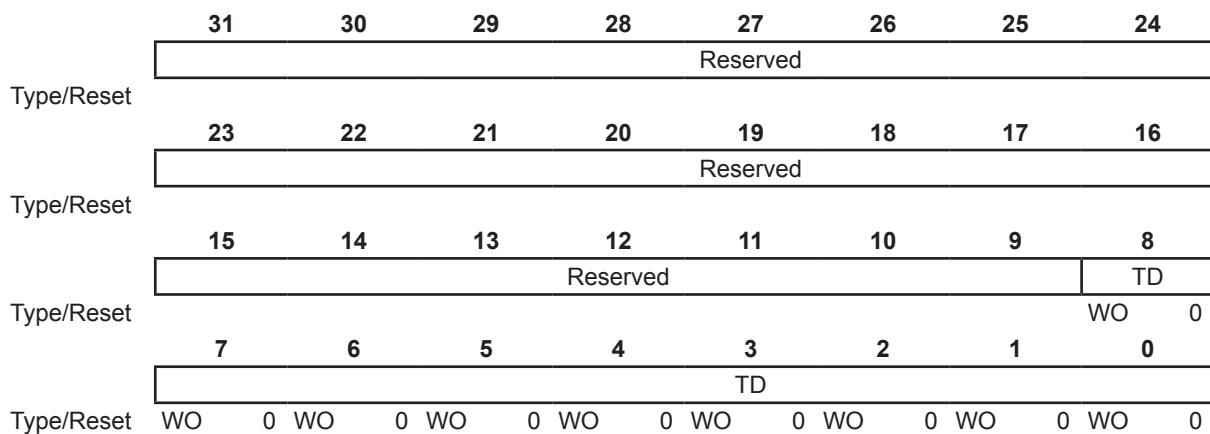
Bits	Field	Descriptions
[8:0]	RD	<p>Reading the data via this receiver buffer register will return the data from the receiver FIFO. The receiver FIFO has a capacity of up to <math>16 \times 9</math> bits.</p> <p>By reading this register, the USART will return a 7, 8 and 9-bit received data. The RD field bit 8 is valid for 9-bit mode only and is fixed at 0 for the 8-bit mode. For the 7-bit mode, the receiver buffer register RD[6:0] contain the available bits.</p>

## USART Transmitter Buffer Register – TBR

The register is used to specify the USART transmitted data.

Offset: 0x000

Reset value: 0x0000\_0000



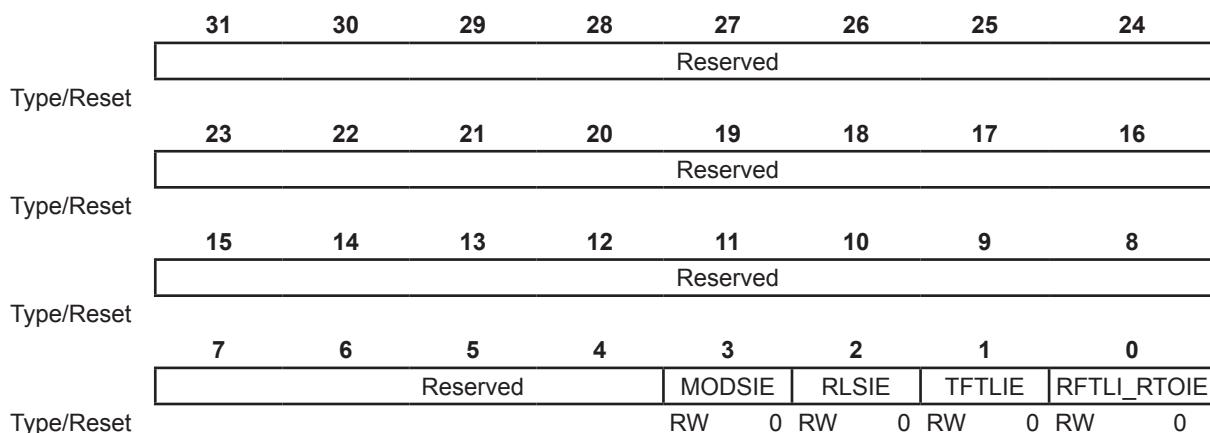
Bits	Field	Descriptions
[8:0]	TD	<p>Writing data to this transmitter buffer register will load data into the transmitter FIFO. The transmitter FIFO has a capacity of up to <math>16 \times 9</math> bits.</p> <p>By writing to this register, the USART will send out 7, 8 or 9-bit transmitted data. The TD 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 transmitter buffer register TD [6:0] contains the available bits.</p>

## USART Interrupt Enable Register – IER

This register is used to enable or disable 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: 0x004

Reset value: 0x0000\_0000



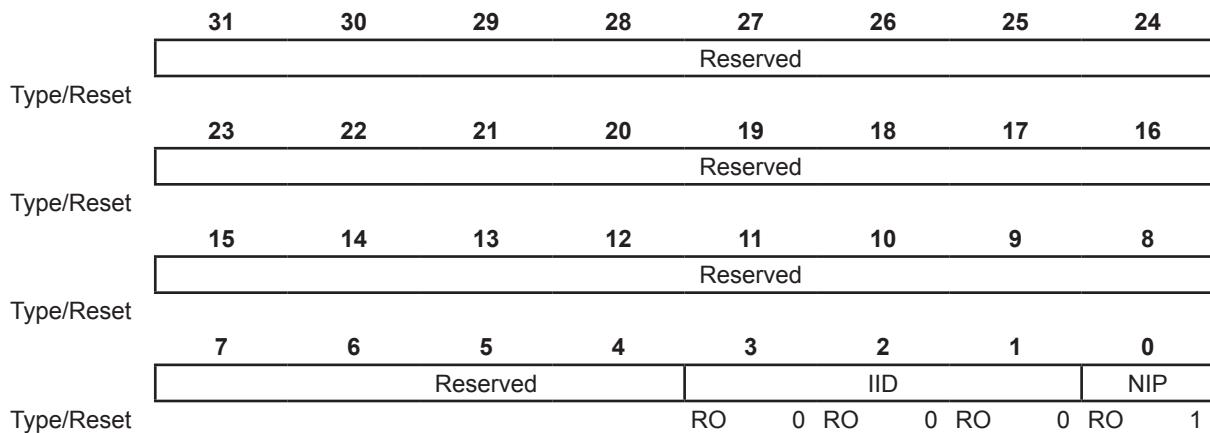
Bits	Field	Descriptions
[3]	MODSIE	MODEM Status Interrupt Enable (Irpt_MODSI) 0: Mask off Irpt_MODSI 1: Enable Irpt_MODSI
[2]	RLSIE	Receive Line Status Interrupt Enable (Irpt_RLSI) 0: Mask off Irpt_RLSI 1: Enable Irpt_RLSI
[1]	TFTLIE	Transmitter FIFO Threshold Level Interrupt (Irpt_TFTLI) Enable 0: Mask off Irpt_TFTLI 1: Enable Irpt_TFTLI
[0]	RFTLI_RTOIE	Receiver FIFO Threshold Level Interrupt Enable (Irpt_RFTLI) or receiver FIFO Time-Out Interrupt Enable (Irpt_RTOI) 0: Mask off Irpt_RFTLI and Irpt_RTOI 1: Enable Irpt_RFTLI or Irpt_RTOI

## USART Interrupt Identification Register – IIR

This register is used to identify the interrupt source including the FIFO threshold and USART Receiver/Transmitter status.

Offset: 0x008

Reset value: 0x0000\_0001



Bits	Field	Descriptions
[3:1]	IID	Interrupt Identification. The detailed interrupt descriptions are shown in the accompanying table. The IID and NIP bits indicate the current USART interrupt request.
[0]	NIP	No Interrupt Pending 1: No pending USART interrupt. 0: Pending USART interrupts.

**Table 53. USART Interrupt Control Function**

IID & NIP	Priority	Interrupt Type	Interrupt Source	Interrupt Reset control
xxx1	NA	None	None	NA
0110	Highest	Receiver Line Status Interrupt (Irpt_RLSI)	USART occurs overrun error, parity error, framing error, or break events. [note] In RS485 mode, it can indicate the address detection.	Reading the LSR register.
0100	Second	Receiver FIFO Threshold Level Interrupt (Irpt_RFTLI)	Receiver FIFO threshold level is reached	Read the RBR register to decrease the receiver FIFO level to less than the specified threshold.
1100	Second	Receiver FIFO Time-out Interrupt (Irpt_RTOI)	Receiver FIFO is not empty and no activities have occurred in the receiver FIFO during the RTOIC time-out duration	Reading the RBR register
0010	Third	Transmitter FIFO Threshold Level Interrupt (Irpt_TFTLI)	Transmitter FIFO level is less than the transmitter FIFO threshold level which is set by the TFTL field in the FIFO Control Register named FCR.	Writing data into the TBR register.
0000	Fourth	MODEM Status Interrupt (Irpt_MODSI)	The DCDS, RIS, DSRS, or CTSS bits in the MODSR register have changing state.	Reading the MODSR register – optional

## USART FIFO Control Register – FCR

This register specifies the USART FIFO control and configurations including threshold level and reset function.

Offset: 0x00C

Reset value: 0x0000\_0001

	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							URRXEN URTXEN	
	7	6	5	4	3	2	1	0	
Type/Reset	RW	0	RW	0	RW	0	WO	0	
	RFTL		TFTL		Reserved	TFR	RFR	FME	
	RW	0	RW	0	RW	0	WO	0	

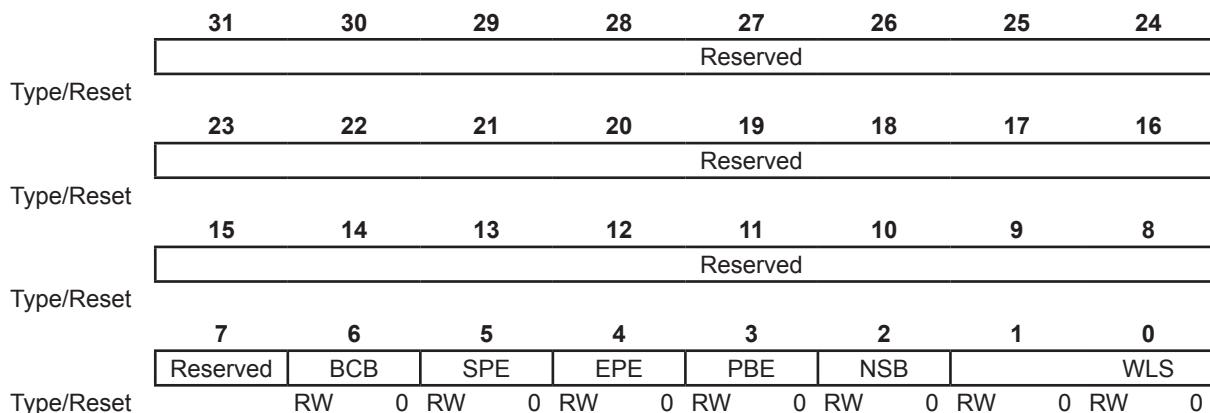
Bits	Field	Descriptions
[9]	URRXEN	USART RX Enable 0: Disable 1: Enable
[8]	URTXEN	USART TX Enable 0: Disable 1: Enable
[7:6]	RFTL	RX FIFO Threshold Level Setting The RFTL field defines the RX FIFO trigger level. 00: 1 byte 01: 4 bytes 10: 8 bytes 11: 14 bytes
[5:4]	TFTL	TX FIFO Threshold Level Setting The TFTL field determines the TX FIFO trigger level. 00: 0 byte 01: 2 bytes 10: 4 bytes 11: 8 bytes
[2]	TFR	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.
[1]	RFR	RX FIFO Reset Setting this bit will generate a reset pulse to reset the RX FIFO which will empty the RX FIFO. i.e., the RX pointer will be reset to 0, after a reset signal. This bit returns to 0 automatically after the reset pulse is generated.
[0]	FME	FIFO Mode Enable Because the USART module is always operated in the FIFO mode, writing to this bit will have no effect.

## USART Line Control Register – LCR

The line control register specifies the serial parameters such as data length, parity, and stop bit for the USART module.

Offset: 0x010

Reset value: 0x0000\_0000



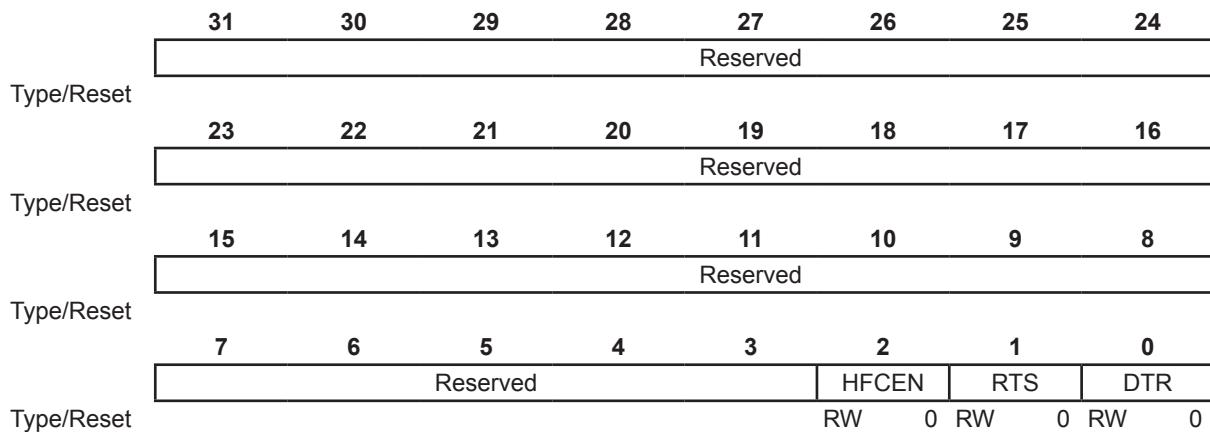
Bits	Field	Descriptions
[6]	BCB	Break Control Bit When this bit is set 1, the serial data output on the UR_TX pin will be forced to the Spacing State (logic 0). This bit acts only on UR_TX output pin and has no effect on the transmitter logic.
[5]	SPE	Stick Parity Enable 0: Disable stick parity 1: Stick 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 set 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 cleared to 0.
[4]	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.
[3]	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.
[2]	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.
[1:0]	WLS	Word Length Select 00: 7 bits 01: 8 bits 10: 9 bits 11: Reserved

## USART Modem Control Register – MODCR

This register contains a related modem control.

Offset: 0x014

Reset value: 0x0000\_0000



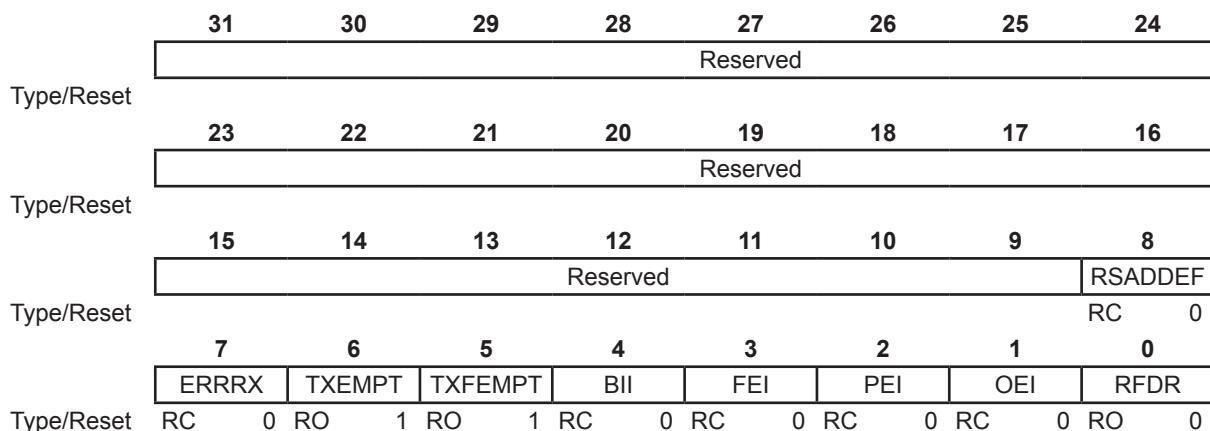
Bits	Field	Descriptions
[2]	HFCEN	Hardware Flow Control Function Enable 0: Disabled 1: Enabled
[1]	RTS	RTS - Request-To-Send Signal 0: Drive UR_RTS pin to logic 1 1: Drive UR_RTS pin to logic 0 Note that the RTS bit is used to control the UR_RTS pin status when the HFCEN bit is set.
[0]	DTR	DTR (Data-Terminal-Ready) Signal 0: Drive DTR pin to logic 1 1: Drive DTR pin to logic 0

## USART Line Status Register – LSR

This register contains the corresponding USART status.

Offset: 0x018

Reset value: 0x0000\_0060



Bits	Field	Descriptions
[8]	RSADDEF	RS485 Address Detection Flag 0: Address is not detected 1: Address is detected This bit is set to 1 when the receiver detects the address and is cleared to 0 when the LSR register is read. Note: This bit is only used in the RS485 mode by setting the MDR field to "b10".
[7]	ERRRX	RX FIFO Error 0: RX FIFO works normally 1: There is at least one parity error (PE), framing error (FE), or break indication (BI) in the receiver FIFO. The ERRRX bit is cleared when the CPU reads the LSR register and if there are no subsequent errors in the RX FIFO
[6]	TXEMPT	Transmitter Empty 0: Either Transmitter FIFO (TX FIFO) or Transmitter Shift Register (TSR) is not empty. 1: Both the TX FIFO and TSR register are empty.
[5]	TXFEMPT	Transmitter FIFO Empty 0: TX FIFO is not empty. 1: TX FIFO is empty. The TXFEMPT bit is set when the last data package in the TX FIFO is transferred to the Transmitter Shift Register (TSR). This bit is reset when the TBR (or TX FIFO) is loaded with new data. This bit also causes the USART to issue an interrupt (Irpt_TFTLI) to the CPU when the TFTLIE bit in the IER register is set to 1 to enable the relevant interrupt and when the TFTL field in the FCR register is set to the value 0.
[4]	BII	Break Interrupt Indicator This bit is set to 1 whenever the received data input is held in the "spacing state" (logic 0) for longer than a full word transmission time, which is the total time of "start bit" + data bits + "parity" + "stop bits" duration, and is reset whenever the CPU reads the contents of this LSR register.

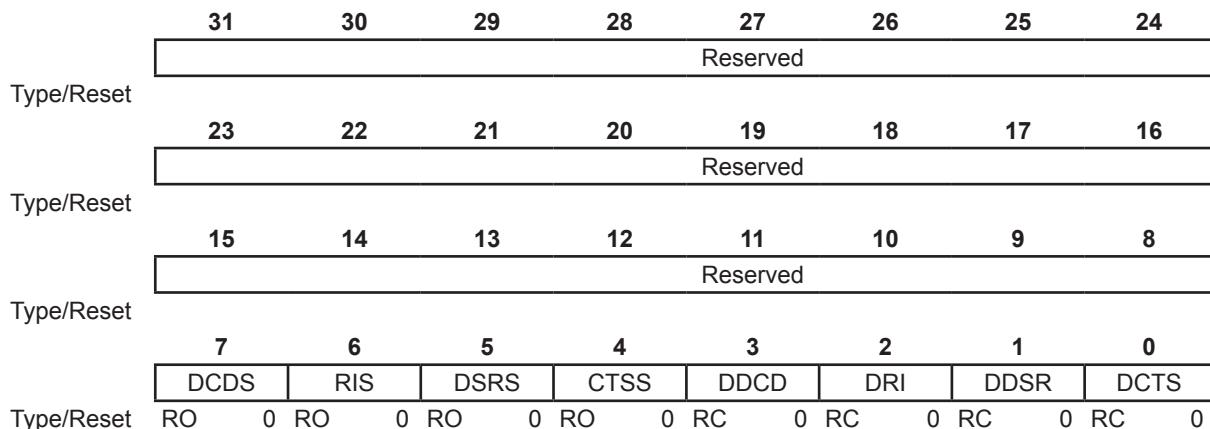
Bits	Field	Descriptions
[3]	FEI	Framing Error Indicator This bit is set 1 whenever the received character does not have a valid "stop bit" , which means, the stop bit following the last data bit or parity bit is detected as a logic 0, and is reset whenever the CPU reads the contents of this LSR register.
[2]	PEI	Parity Error Indicator This bit is set to 1 whenever the received character does not have a valid "parity bit" and is reset whenever the CPU reads the contents of this LSR register.
[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 it is not transferred to the RX FIFO. The OEI bit is used to indicate to the CPU as soon as it happens and is reset whenever the CPU reads the contents of this LSR register.
[0]	RFDR	RX FIFO Data Ready 0: RX FIFO is empty 1: RX FIFO contains at least 1 received data word

## USART Modem Status Register – MODSR

This register contains the USART modem status bits.

Offset: 0x01C

Reset value: 0x0000\_0000



Bits	Field	Descriptions
[7]	DCDS	DCD - Data-Carrier-Detect Status 0: UR_DCD pin is inactive. 1: UR_DCD pin is active and kept at a "0" state
[6]	RIS	UR_RI Ring-Indicator Status 0: UR_RI pin is inactive 1: UR_RI pin is active and kept at a "0" state
[5]	DSRS	UR_DSR Data-Set-Ready Status 0: UR_DSR pin is inactive 1: UR_DSR pin is active and kept at a "0" state
[4]	CTSS	UR_CTS Clear-To-Send Status 0: UR_CTS pin is inactive 1: UR_CTS pin is active and kept at a "0" state
[3]	DDCD	Detect UR_DCD Status Change This bit is set whenever the UR_DCD input pin status has been changed and will be reset if the CPU reads this MODSR register. When the DDCD bit is set to 1, a modem status interrupt is generated if the MODSIE bit in the IER register is set to 1.
[2]	DRI	Detect UR_RI Status Change This bit is set whenever the UR_RI input pin status has been changed and will be reset if the CPU reads the MODSR register. When the DIR bit is set to 1, a modem status interrupt is generated if the MODSIE bit in the IER register is set to 1.
[1]	DDSR	Detect UR_DSR Status Change This bit is set whenever the UR_DSR input pin status has been changed and will be reset if the CPU reads the MODSR register. When the DDSR bit is set to 1, a modem status interrupt is generated if the MODSIE bit in the IER register is set to 1.
[0]	DCTS	Detect UR_CTS Status Change This bit is set whenever the UR_CTS input pin status has been changed and will be reset if the CPU reads the MODSR register. When the DCTS bit is set to 1, a modem status interrupt is generated if the MODSIE bit in the IER register is set to 1.

## USART Timing Parameter Register – TPR

This register contains the USART timing parameters including the transmitter time guard parameters and the receiver FIFO time-out value together with the RX FIFO time-out interrupt enable control.

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	TG								
	RW	0	RW	0	RW	0	RW	0	
Type/Reset	7	6	5	4	3	2	1	0	
	RTOIE		RTOIC						
Type/Reset	RW	0	RW	0	RW	0	RW	0	

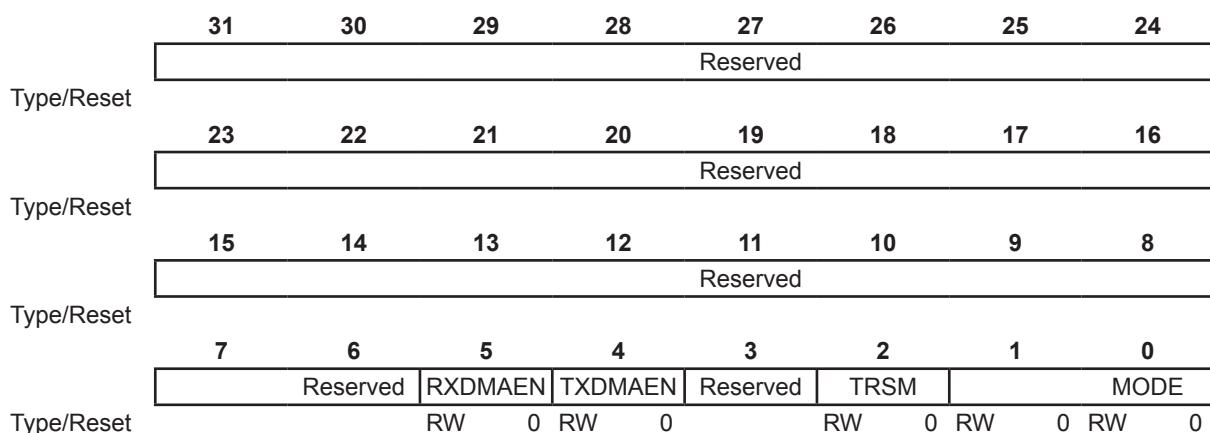
Bits	Field	Descriptions
[15:8]	TG	Transmitter Time Guard The transmitter time guard counter is driven by the baud rate clock. When the TX FIFO transmits data, the counter is reset and then starts to count. Only when the counter content is equal to the TG value, are further word transmission transactions allowed.
[7]	RTOIE	Receiver FIFO Time Out Interrupt Enable The receiver FIFO time-out interrupt is enabled only when the RFTLI_RTOIE bit in the IER register is set to 1.
[6:0]	RTOIC	Receiver FIFO Time-Out Interrupt Compare value The RX FIFO time-out counter, TOUT_CNT, is driven by the baud rate clock. When the RX FIFO receives new data, the counter is reset and then starts to count. Once the time-out counter content is equal to the time-out interrupt compare RTOIC value, a receiver FIFO time-out interrupt, Irpt_RTOI, is generated if the RFTLI_RTOIE bit in the IER 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 Mode Register – MDR

This register specifies the USART mode and the data transfer mode selections.

Offset: 0x024

Reset value: 0x0000\_0000



Bits	Field	Descriptions
[5]	RXDMAEN	USART RX DMA Enable 0: Disabled 1: Enabled
[4]	TXDMAEN	USART TX DMA 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 IrDA Control Register – IrDPCR

This register is the IrDA control register for the USART.

Offset: 0x028

Reset value: 0x0000\_0000

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

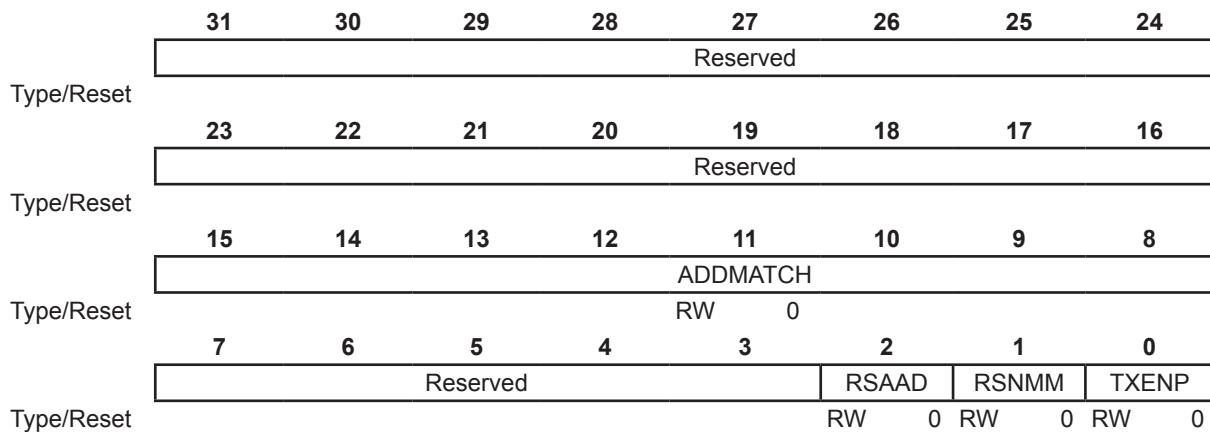
Bits	Field	Descriptions
[15:8]	IrDAPSC	<p>IrDA Prescaler value</p> <p>This field contains the 8-bit debounce prescaler value.</p> <p>The debounce count-down counter is driven by the USART clock, named as CK_USART. The counting period is specified by the IrDAPSC field. The IrDAPSC field must be set to a value equal to or greater than 0x01 to 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 inversed</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 inversed</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>
[0]	IrDAEN	<p>IrDA Enable control</p> <ul style="list-style-type: none"> <li>0: Disable IrDA mode</li> <li>1: Enable IrDA mode</li> </ul>

## USART RS485 Control Register – RS485CR

This register contains the USART UR\_RTS/TXE pin polarity for the RS485 mode.

Offset: 0x02C

Reset value: 0x0000\_0000



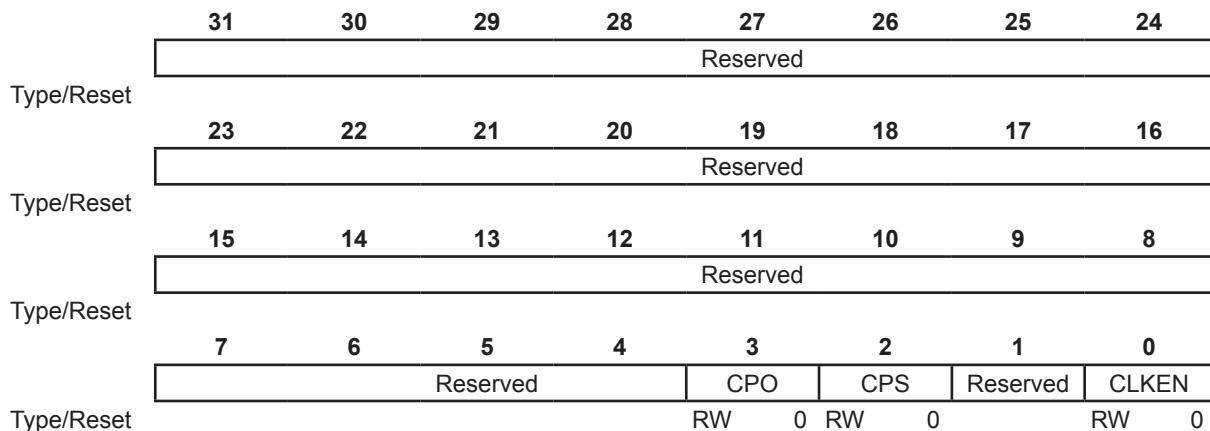
Bits	Field	Descriptions
[15:8]	ADDMATCH	RS485 Auto Address Match value The field contains the address match value for the RS485 auto address detection operation mode.
[2]	RSAAD	RS485 Auto Address Detection Operation Mode Control 0: Disable 1: Enable
[1]	RSNMM	RS485 Normal Multi-drop Operation Mode Control 0: Disable 1: Enable
[0]	TXENP	UR_RTS/TXE Pin Polarity 0: UR_RTS/TXE is active high in the RS485 transmission mode 1: UR_RTS/TXE is active low in the RS485 transmission mode

## USART Synchronous Control Register – SYNCR

This register is used to control the USART synchronous clock pin and the clock polarity together with the clock phase for the synchronous mode.

Offset: 0x030

Reset value: 0x0000\_0000



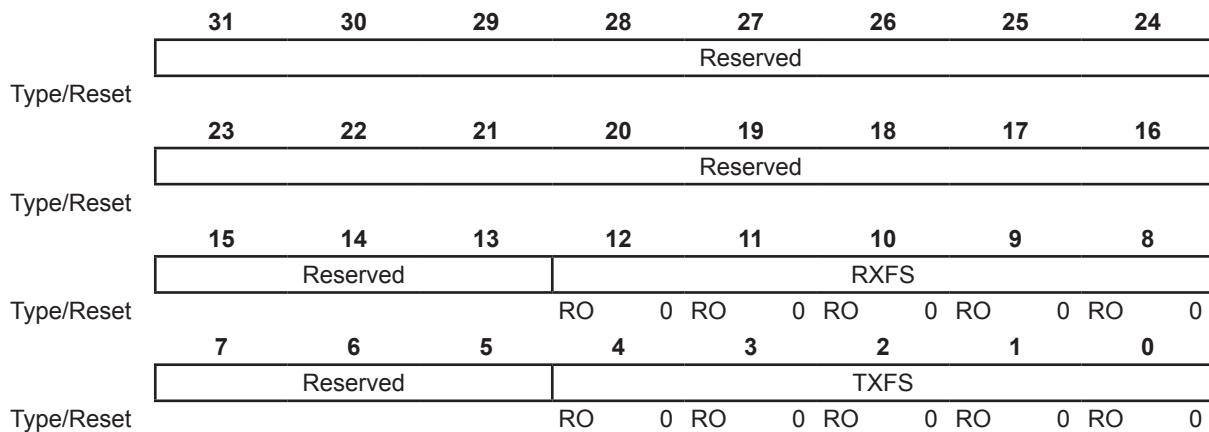
Bits	Field	Descriptions
[3]	CPO	<p>Clock Polarity</p> <p>Selects the polarity of the clock output on the UR_CTS/SCK pin in the synchronous mode. Works in conjunction with the CPS bit to specify the desired clock idle state.</p> <p>0: UR_CTS/SCK pin idle state is low 1: UR_CTS/SCK pin idle state is high</p>
[2]	CPS	<p>Clock Phase</p> <p>This bit allows the user to select the phase of the clock output on the UR_CTS/SCK pin in the synchronous mode. Works in conjunction with the CPO bit to determine the data capture edge.</p> <p>0: Data is captured on the first clock edge 1: Data is captured on the second clock edge</p>
[0]	CLKEN	<p>Clock Enable</p> <p>Enable/disable the UR_CTS/SCK pin.</p> <p>0: UR_CTS/SCK pin disabled 1: UR_CTS/SCK pin enabled</p>

## USART FIFO Status Register – FSR

This register is the USART FIFO status register

Offset: 0x034

Reset value: 0x0000\_0000



Bits	Field	Descriptions
[12:8]	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>00000: Rx FIFO is empty</li> <li>00001: Rx FIFO contains 1 data</li> <li>...</li> <li>10000: Rx FIFO contains 16 data</li> <li>Others: Reserved</li> </ul>
[4:0]	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>00000: TX FIFO is empty</li> <li>00001: TX FIFO contains 1 data</li> <li>...</li> <li>10000: TX FIFO contains 16 data</li> <li>Others: Reserved</li> </ul>

## USART Divider Latch Register—DLR

The register is used to determine the USART clock divided ratio to generate the appropriate baud rate.

Offset: 0x038

Reset value: 0x0000\_0010

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

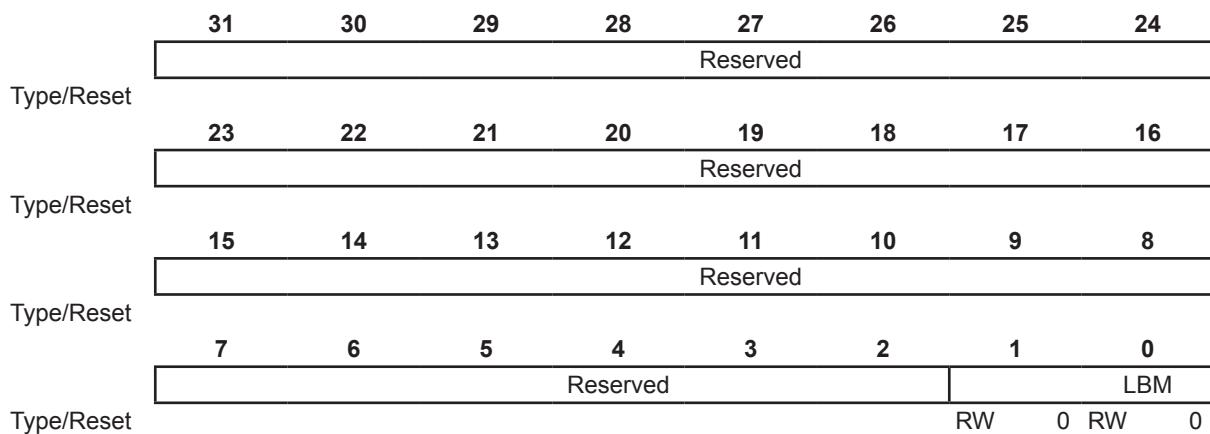
Bits	Field	Descriptions
[15:0]	BRD	Baud Rate Divider The 16 bits define the USART clock divider ratio. Baud Rate = CK_USART / BRD Where the CK_USART clock is the system clock connected to the USART module. The BRD field can be set from 16 to 65535.

## USART Debug/Test Register – DEGTSTR

This register controls the USART debug mode.

Offset: 0x040

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 with 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 peripheral supports four-type interrupts as follows:

- Line Status Interrupt
- Transmitter FIFO Empty Interrupt
- Receiver Threshold Level Reaching Interrupt
- Time Out Interrupt

The UART module includes a 16-byte transmitter FIFO, TX\_FIFO, and a 16-byte receiver FIFO, RX\_FIFO.

Software can detect a UART error status by reading Line Status Register, LSR. The status includes the type and 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 CK\_AHB to produce a clock for the UART transmitter and receiver.

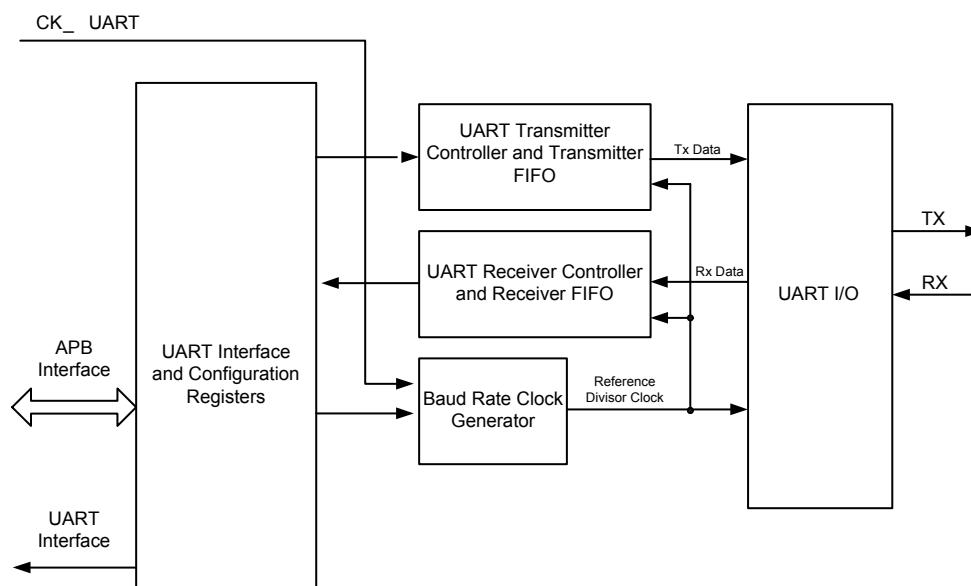


Figure 158. UART Block Diagram

## Features

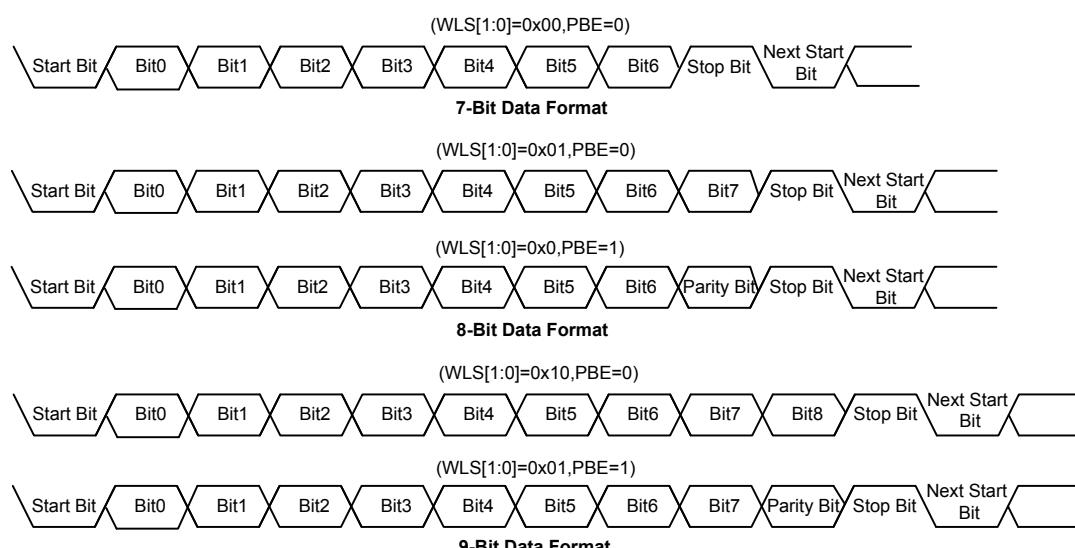
- Asynchronous serial communication mode is supported
- Full Duplex Communication Capability
- Fully programmable serial communication functions including:
  - Word length: 7, 8, or 9-bit character
  - Parity: Even, odd, or no-parity bit generation and detection
  - Stop bit: 1 or 2 stop bit generation
  - Bit order: LSB-first or MSB-first transfer
- Error detection: Parity, overrun, and frame error
- FIFO:
  - Receive FIFO:  $16 \times 9$  bits (max 9 data bits)
  - Transmit FIFO:  $16 \times 9$  bits (max 9 data bits)

## Function Descriptions

### Serial Data Format

The UART module performs a parallel-to-serial conversion on data that is read from the Transmitter FIFO 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 start and stop bits are both 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 Receiver FIFO. 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 159. 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 system clock connected to the UART module while the BRD range is from 16 to 65535.

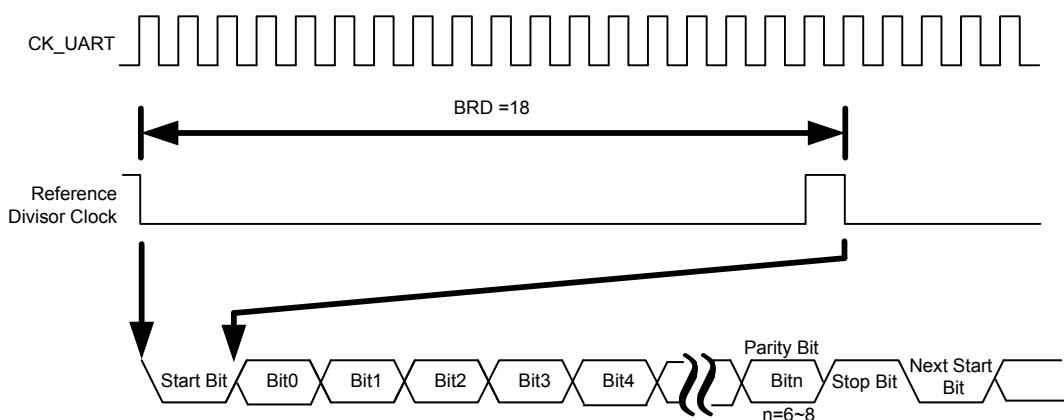


Figure 160. UART Clock CK\_UART and Data Frame Timing

Table 54. Baud Rate Error Calculation – CK\_UART = 72 MHz

Baud rate		CK_UART = 72 MHz			
No	Kbps	Actual	BRD/16	BRD	Error rate
1	2.4	2.4	1875	30000	0%
2	9.6	9.6	468.75	7500	0%
3	19.2	19.2	234.375	3750	0%
4	57.6	57.6	78.125	1250	0%
5	115.2	115.2	39.0625	625	0%
6	230.4	230.769	19.5	312	0.16%
7	460.8	461.538	9.75	156	0.16%
8	921.6	923.076	4.875	78	0.16%
9	2250	2250	2	32	0%
10	4500	4500	1	16	0%

## Interrupts and Status

The UART module can generate an interrupt when the following event occurs:

- Receiver Line Status Interrupt (Irpt\_RLSI): Occurrence of UART overrun error, parity error, framing error, or break events.
- Receiver FIFO Threshold Level Interrupt (Irpt\_RFTLI): When the FIFO received data amount has reached the specified threshold level.
- Receiver FIFO Time-out Interrupt (Irpt\_RTOI): When the UART receiver FIFO does not receive a new data package during the specified time-out interval.
- Transmitter FIFO Threshold Level Interrupt (Irpt\_TFTLI): When the data to be transmitted in the USART transmitter FIFO is less than the specified threshold level.
- MODEM Status Interrupt (Irpt\_MODSI): When the DCD, RI, DSR, or CTS bits states in the MODSR register have changed.

## PDMA Interface

The PDMA interface is integrated in the UART module. The PDMA function can be enabled by setting the TXDMAEN or RXDMAEN bit in the MDR register to 1 in the transmitter or receiver mode respectively. When the data to be transmitted in the UART transmitter FIFO is less than the TX FIFO threshold level specified by the TFTL field in the FCR register and the TXDMAEN bit is set to 1, the PDMA function will be activated to move data from a certain memory location into the UART TX FIFO.

Similarly, when the received data amount in the receiver FIFO is equal to the RX FIFO threshold level specified by the RFTL field in the FCR register and the RXDMAEN bit is set to 1, the PDMA function will be activated to move data from the UART RX FIFO to a specific memory location.

For a more detailed description on the PDMA configurations, refer to the PDMA chapter.

## Register Map

The following table shows the UART registers and reset values.

**Table 55. UART Register Map**

Register	Offset	Description	Reset Value
<b>UART0 Base Address = 0x4000_1000</b>			
<b>UART1 Base Address = 0x4004_1000</b>			
RBR	0x000	UART Receiver Buffer Register	0x0000_0000
TBR	0x000	UART Transmitter Buffer Register	0x0000_0000
IER	0x004	UART Interrupt Enable Register	0x0000_0000
IIR	0x008	UART Interrupt Identification Register	0x0000_0001
FCR	0x00C	UART FIFO Control Register	0x0000_0001
LCR	0x010	UART Line Control Register	0x0000_0000
LSR	0x018	UART Line Status Register	0x0000_0060
TPR	0x020	UART Timing Parameter Register	0x0000_0000
MDR	0x024	UART Mode Register	0x0000_0000
FSR	0x034	UART FIFO Status Register	0x0000_0000
DLR	0x038	UART Divider Latch Register	0x0000_0010
DEGTSTR	0x040	UART Debug/Test Register	0x0000_0000

## Register Descriptions

### UART Receiver Buffer Register – RBR

The register is used to store the UART received data.

Offset: 0x000

Reset value: 0x0000\_0000

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

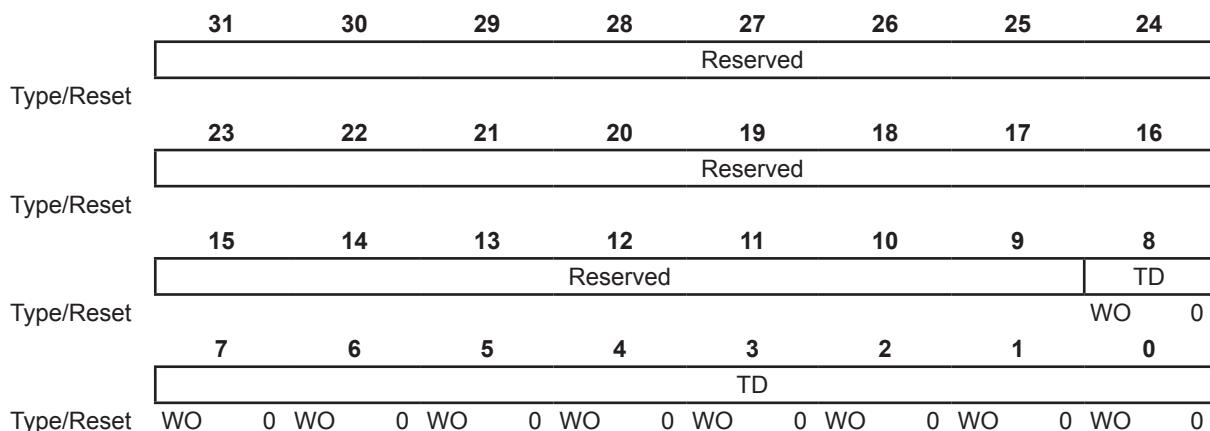
Bits	Field	Descriptions
[8:0]	RD	Reading the data via this receiver buffer register will return the data from the receiver FIFO. The receiver FIFO has a capacity of up to $16 \times 9$ bits. By reading this register, the USART will return a 7, 8 and 9-bit received data. The RD field bit 8 is valid for 9-bit mode only and is fixed at 0 for the 8-bit mode. For the 7-bit mode, the receiver buffer register RD[6:0] contain the available bits.

## UART Transmitter Buffer Register – TBR

The register is used to specify the UART transmitted data.

Offset: 0x000

Reset value: 0x0000\_0000



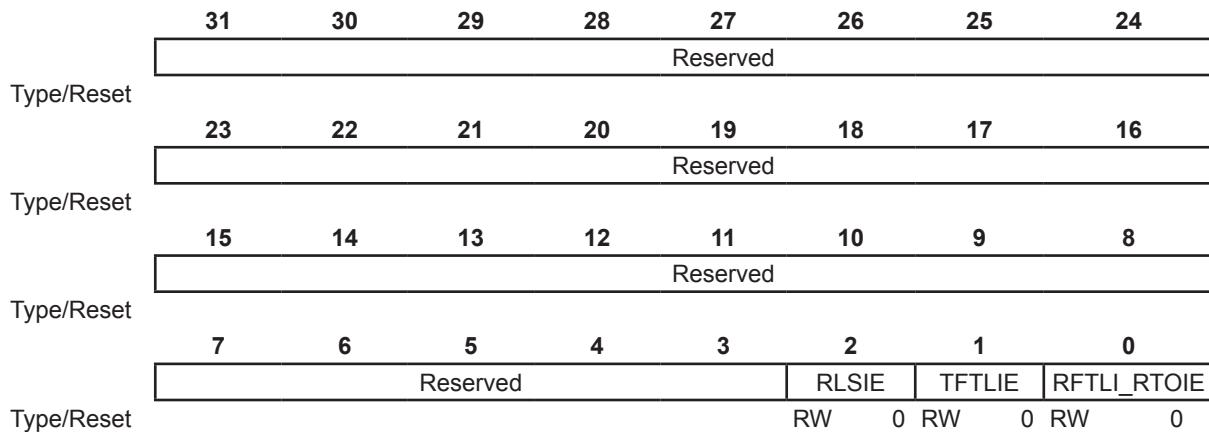
Bits	Field	Descriptions
[8:0]	TD	<p>Writing data to this transmitter buffer register will load data into the transmitter FIFO. The transmitter FIFO has a capacity of up to <math>16 \times 9</math> bits.</p> <p>By writing to this register, the USART will send out 7, 8 or 9-bit transmitted data. The TD 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 transmitter buffer register TD [6:0] contains the available bits.</p>

## UART Interrupt Enable Register – IER

This register is used to enable or disable 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: 0x004

Reset value: 0x0000\_0000



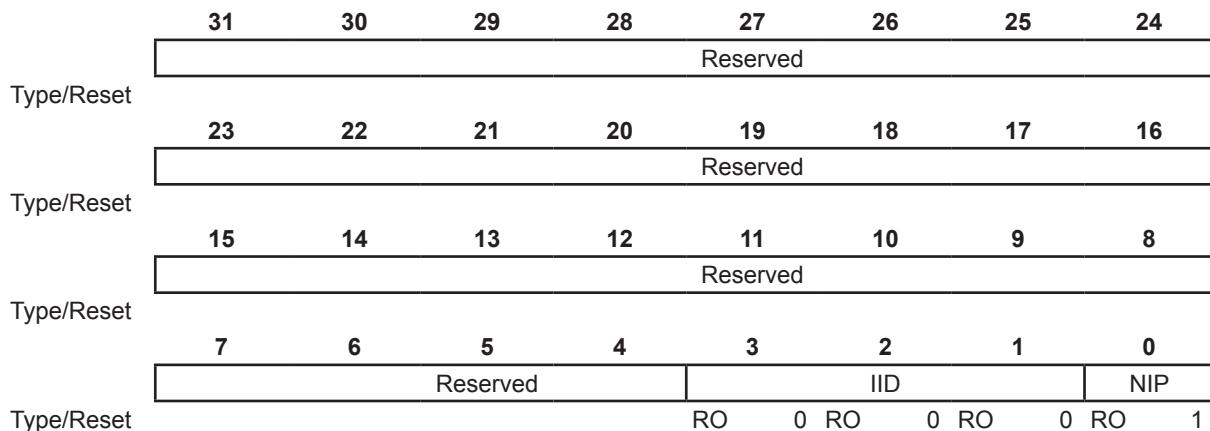
Bits	Field	Descriptions
[2]	RLSIE	Receive Line Status Interrupt Enable (Irpt_RLSI) 0: Mask Irpt_RLSI 1: Enable Irpt_RLSI
[1]	TFTLIE	Transmitter FIFO Threshold Level Interrupt (Irpt_TFTLI) Enable 0: Mask Irpt_TFTLI 1: Enable Irpt_TFTLI
[0]	RFTLI_RTOIE	Receiver FIFO Threshold Level Interrupt Enable (Irpt_RFTLI) or receiver FIFO Time-Out Interrupt Enable (Irpt_RTOI) 0: Mask Irpt_RFTLI and Irpt_RTOI 1: Enable Irpt_RFTLI or Irpt_RTOI

## UART Interrupt Identification Register – IIR

This register is used to identify the interrupt source including the FIFO threshold and UART Receiver/Transmitter status.

Offset: 0x008

Reset value: 0x0000\_0001



Bits	Field	Descriptions
[3:1]	IID	Interrupt Identification The detailed interrupt descriptions are shown in the accompanying table. The IID and NIP bits indicate the current UART interrupt request.
[0]	NIP	No Interrupt Pending 1: No pending URAT interrupt 0: Pending UART interrupts

Table 56. UART Interrupt Control Function

IIR[3:1] & NIP[0]	Priority	Interrupt Type	Interrupt Source	Interrupt Reset control
xxx1	NA	None	None	NA
0110	Highest	Receiver Line Status Interrupt (Irpt_RLSI)	UART occurs overrun error, parity error, framing error, or break events. [note] In RS485 mode, it can indicate the address detection.	Reading the LSR register.
0100	Second	Receiver FIFO Threshold Level Interrupt (Irpt_RFTLI)	Receiver FIFO threshold level is reached	Read the RBR register to decrease the receiver FIFO level to less than the specified threshold.
1100	Second	Receiver FIFO Time-out Interrupt (Irpt_RTOI)	Receiver FIFO is not empty and no activities have occurred in the receiver FIFO during the RTOIC time-out duration	Reading the RBR register
0010	Third	Transmitter FIFO Threshold Level Interrupt (Irpt_TFTLI)	Transmitter FIFO level is less than the transmitter FIFO threshold level which is set by the TFTL field in the FIFO Control Register named FCR.	Writing data into the TBR register.

## UART FIFO Control Register – FCR

This register specifies the UART FIFO control and configurations including threshold level and reset function.

Offset: 0x00C

Reset value: 0x0000\_0001

	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						URRXEN	URTXEN	
	7	6	5	4	3	2	1	0	
Type/Reset	RW	0	RW	0	RW	0	WO	0	
	RFTL		TFTL		Reserved	TFR	RFR	FME	
	RW	0	RW	0	RW	0	WO	0	

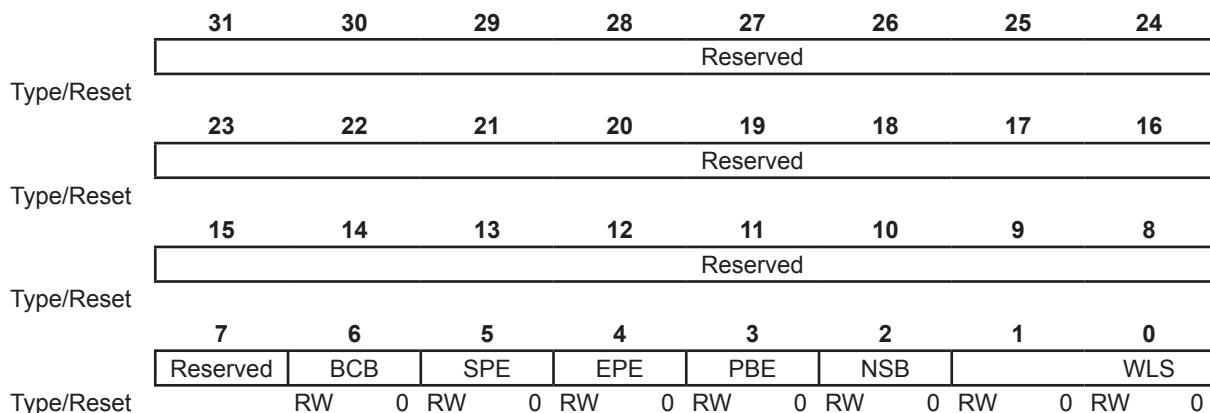
Bits	Field	Descriptions
[9]	URRXEN	UART RX Enable 0: Disable 1: Enable
[8]	URTXEN	UART TX Enable 0: Disable 1: Enable
[7:6]	RFTL	RX FIFO Threshold Level Setting The RFTL field defines the RX FIFO trigger level. 00: 1 byte 01: 4 bytes 10: 8 bytes 11: 14 bytes
[5:4]	TFTL	TX FIFO Threshold Level Setting The TFTL field determines the TX FIFO trigger level. 00: 0 byte 01: 2 bytes 10: 4 bytes 11: 8 bytes
[2]	TFR	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.
[1]	RFR	RX FIFO Reset Setting this bit will generate a reset pulse to reset the RX FIFO which will empty the RX FIFO. i.e., the RX pointer will be reset to 0, after a reset signal. This bit returns to 0 automatically after the reset pulse is generated.
[0]	FME	FIFO Mode Enable Because the UART module is always operated in the FIFO mode, writing to this bit will have no effect.

## UART Line Control Register – LCR

The line control register specifies the serial parameters such as data length, parity, and stop bit for the UARTn module.

Offset: 0x010

Reset value: 0x0000\_0000



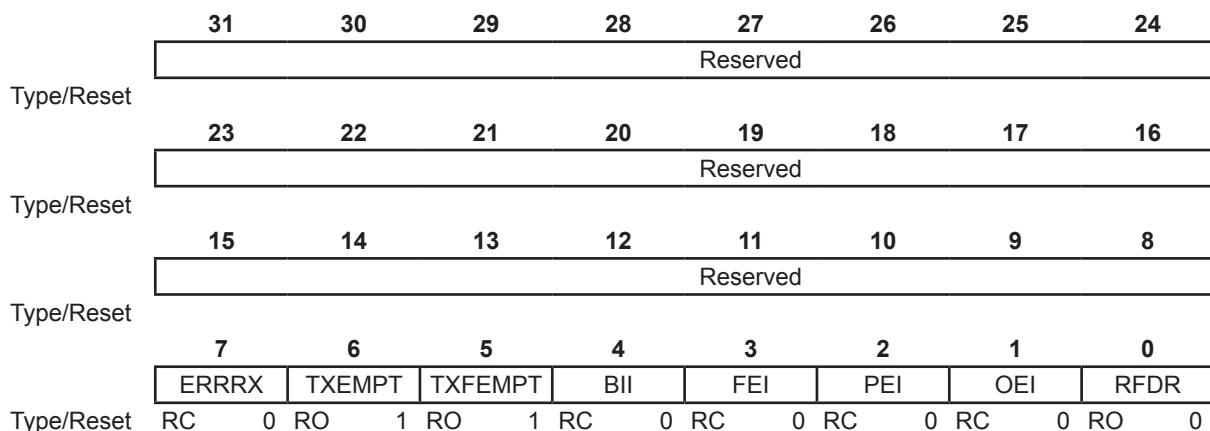
Bits	Field	Descriptions
[6]	BCB	Break Control Bit When this bit is set 1, the serial data output on the UR_TX pin will be forced to the Spacing State (logic 0). This bit acts only on UR_TX output pin and has no effect on the transmitter logic.
[5]	SPE	Stick Parity Enable 0: Disable stick parity 1: Stick 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 set 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 cleared to 0.
[4]	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.
[3]	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.
[2]	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.
[1:0]	WLS	Word Length Select 00: 7 bits 01: 8 bits 10: 9 bits 11: Reserved

## UART Line Status Register – LSR

This register contains the corresponding UART status.

Offset: 0x018

Reset value: 0x0000\_0060



Bits	Field	Descriptions
[7]	ERRRX	RS485 Address Detection Flag 0: Address is not detected 1: Address is detected This bit is set to 1 when the receiver detects the address and is cleared to 0 when the LSR register is read. Note: This bit is only used in the RS485 mode by setting the MDR field to "b10".
[6]	TXEMPT	RX FIFO Error 0: RX FIFO works normally 1: There is at least one parity error (PE), framing error (FE), or break indication (BI) in the receiver FIFO. The ERRRX bit is cleared when the CPU reads the LSR register and if there are no subsequent errors in the RX FIFO
[5]	TXFEMPT	Transmitter Empty 0: Either Transmitter FIFO (TX FIFO) or Transmitter Shift Register (TSR) is not empty 1: Both the TX FIFO and TSR register are empty
[4]	BII	Transmitter FIFO Empty 0: TX FIFO is not empty 1: TX FIFO is empty The TXFEMPT bit is set when the last data package in the TX FIFO is transferred to the Transmitter Shift Register (TSR). This bit is reset when the TBR (or TX FIFO) is loaded with new data. This bit also causes the USART to issue an interrupt (Irpt_TFTLI) to the CPU when the TFTLIE bit in the IER register is set to 1 to enable the relevant interrupt and when the TFTL field in the FCR register is set to the value 0.
[3]	FEI	Break Interrupt Indicator This bit is set to 1 whenever the received data input is held in the "spacing state" (logic 0) for longer than a full word transmission time, which is the total time of "start bit" + data bits + "parity" + "stop bits" duration, and is reset whenever the CPU reads the contents of this LSR register.

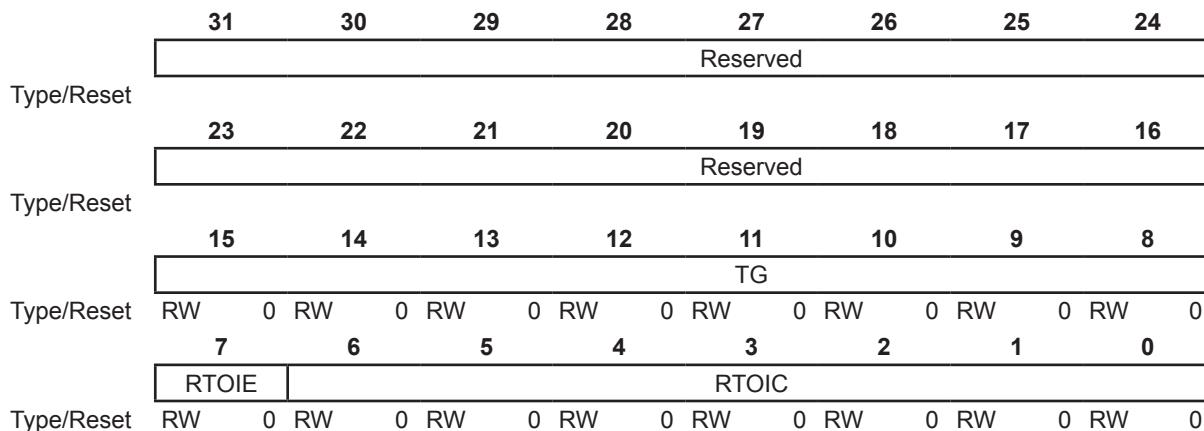
<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>
[2]	PEI	Framing Error Indicator This bit is set 1 whenever the received character does not have a valid "stop bit" , which means, the stop bit following the last data bit or parity bit is detected as a logic 0, and is reset whenever the CPU reads the contents of this LSR register.
[1]	OEI	Parity Error Indicator This bit is set to 1 whenever the received character does not have a valid "parity bit" and is reset whenever the CPU reads the contents of this LSR register.
[0]	RFDR	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 it is not transferred to the RX FIFO. The OEI bit is used to indicate to the CPU as soon as it happens and is reset whenever the CPU reads the contents of this LSR register.

## UART Timing Parameter Register – TPR

This register contains the UART timing parameters including the transmitter time guard parameters and the receiver FIFO time-out value together with the RX FIFO time-out interrupt enable control.

Offset: 0x020

Reset value: 0x0000\_0000



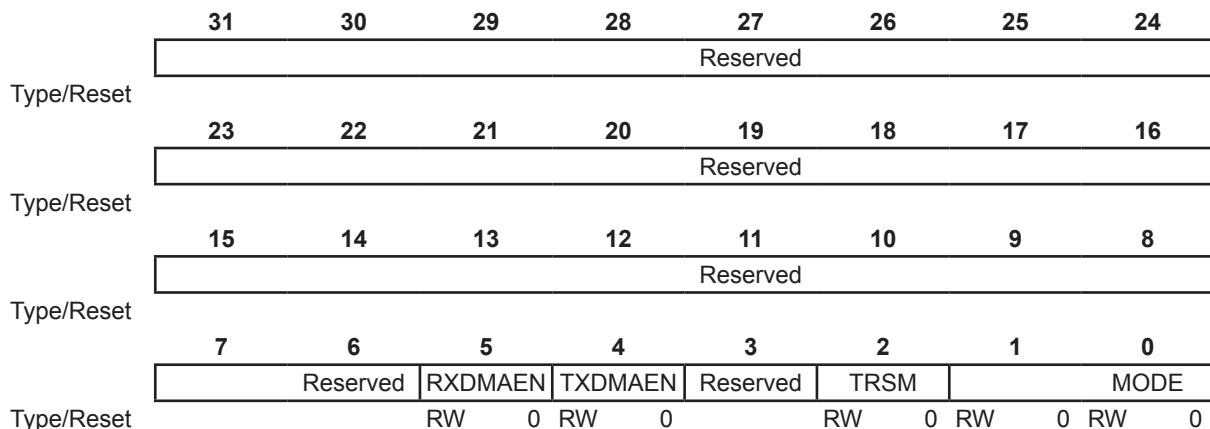
Bits	Field	Descriptions
[15:8]	TG	Transmitter Time Guard The transmitter time guard counter is driven by the baud rate clock. When the TX FIFO transmits data, the counter is reset and then starts to count. Only when the counter content is equal to the TG value, are further word transmission transactions allowed.
[7]	RTOIE	Receiver FIFO Time Out Interrupt Enable The receiver FIFO time-out interrupt is enabled only when the RFTLI_RTOIE bit in the IER register is set to 1.
[6:0]	RTOIC	Receiver FIFO Time-Out Interrupt Compare value The RX FIFO time-out counter, TOUT_CNT, is driven by the baud rate clock. When the RX FIFO receives new data, the counter is reset and then starts to count. Once the time-out counter content is equal to the time-out interrupt compare RTOIC value, a receiver FIFO time-out interrupt, Int_RTOI, is generated if the RFTLI_RTOIE bit in the IER register is set to 1. New received data or the empty RX FIFO after being read will clear the RX FIFO time-out counter.

## UART Mode Register – MDR

This register specifies the UART mode and the data transfer mode selections.

Offset: 0x024

Reset value: 0x0000\_0000



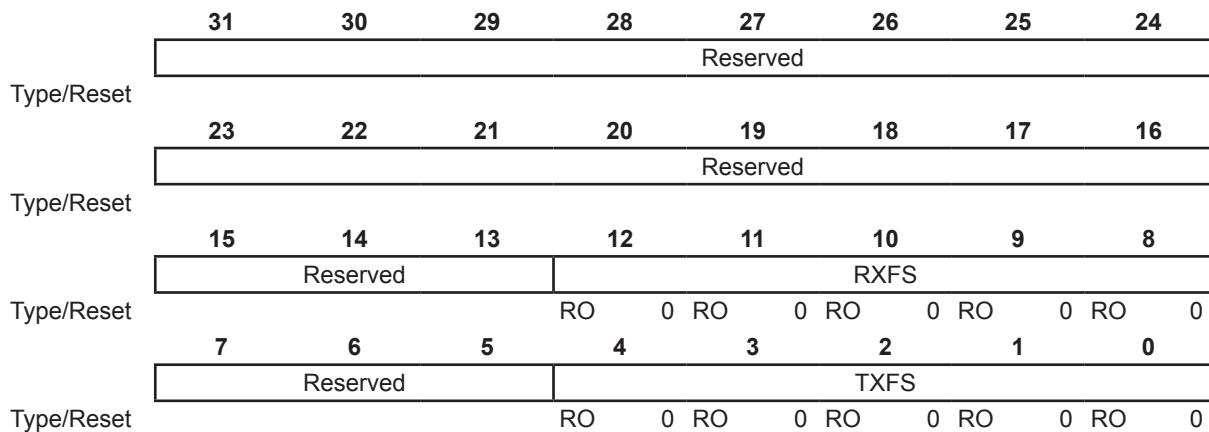
Bits	Field	Descriptions
[5]	RXDMAEN	UART RX DMA Enable 0: Disabled 1: Enabled
[4]	TXDMAEN	UART TX DMA Enable 0: Disabled 1: Enabled
[2]	TRSM	Transfer Mode Selection This bit is used to select the data transfer protocol. 0: LSB first 1: MSB first
[1:0]	MODE	UART Mode Selection 00: Normal operation 01: IrDA 10: RS485 11: Synchronous

## UART FIFO Status Register – FSR

This register is the UART FIFO status register

Offset: 0x034

Reset value: 0x0000\_0000



Bits	Field	Descriptions
[12:8]	RXFS	<p><b>RX FIFO Status</b></p> <p>The RXFS field shows the current number of data contained in the RX FIFO.</p> <ul style="list-style-type: none"> <li>00000: Rx FIFO is empty.</li> <li>00001: Rx FIFO contains 1 data</li> <li>...</li> <li>10000: Rx FIFO contains 16 data</li> <li>Others: Reserved</li> </ul>
[4:0]	TXFS	<p><b>TX FIFO Status</b></p> <p>The TXFS field shows the current number of data contained in the TX FIFO.</p> <ul style="list-style-type: none"> <li>00000: TX FIFO is empty</li> <li>00001: TX FIFO contains 1 data</li> <li>...</li> <li>10000: TX FIFO contains 16 data</li> <li>Others: Reserved</li> </ul>

## UART Divider Latch Register – DLR

The register is used to determine the UART clock divided ratio to generate the appropriate baud rate.

Offset: 0x038

Reset value: 0x0000\_0010

	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
BRD								
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
BRD								

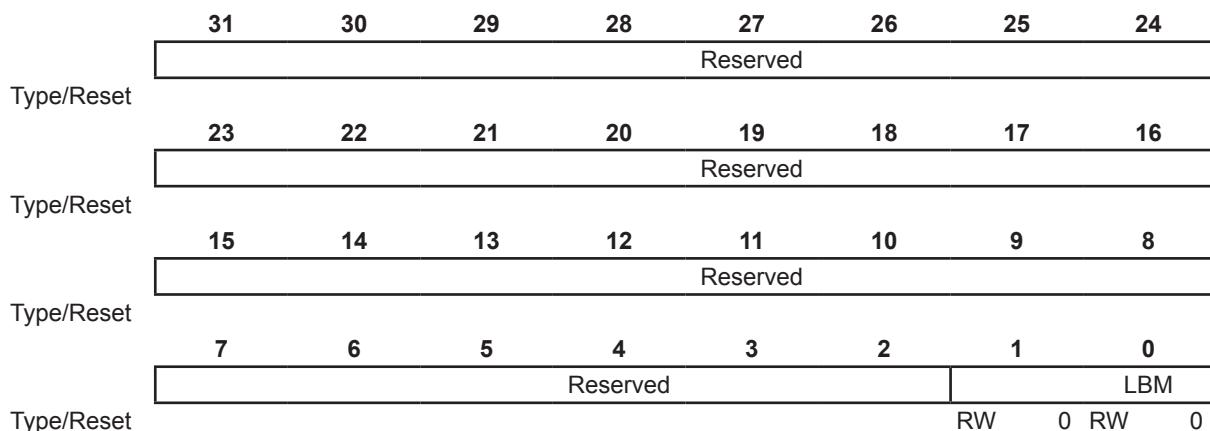
Bits	Field	Descriptions
[15:0]	BRD	<p>Baud Rate Divider</p> <p>The 16 bits define the UART clock divider ratio.</p> <p>Baud Rate = CK_UART / BRD</p> <p>Where the CK_UART clock is the system clock connected to the UART module. The BRD field can be set from 16 to 65535.</p>

## UART Debug/Test Register – DEGTSTR

This register controls the UART debug mode.

Offset: 0x040

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.

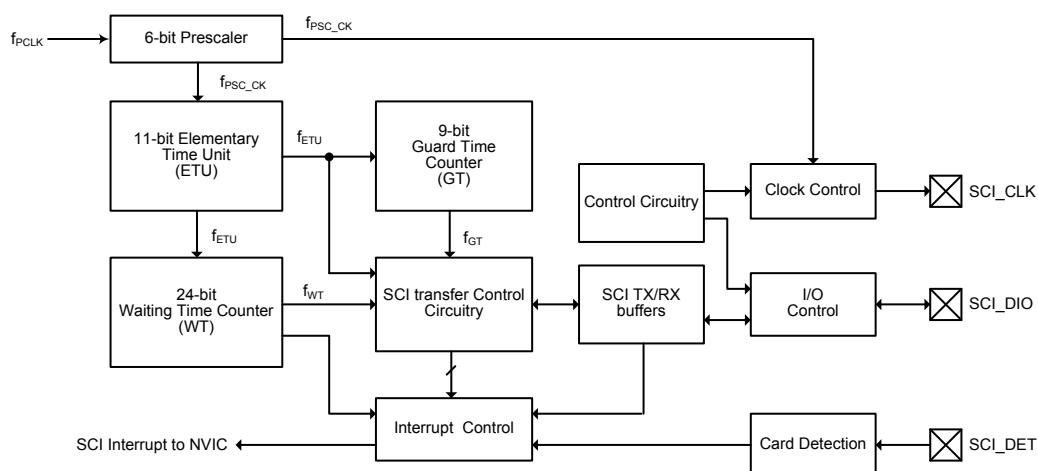


Figure 161. SCI Block Diagram

## Features

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

## Functional Descriptions

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

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

where:

- **etu** is the nominal duration of the data bit on the signal SCI\_DIO provided to the card by the interface
- **D<sub>i</sub>** is the bit-rate adjustment factor
- **F<sub>i</sub>** 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 57. DI Field Based D<sub>i</sub> Encoded Decimal Values**

DI field	0001	0010	0011	0100	0101	0110	1000	1001
D <sub>i</sub> (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 58. FI Field Based F<sub>i</sub> Encoded Decimal Values**

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

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

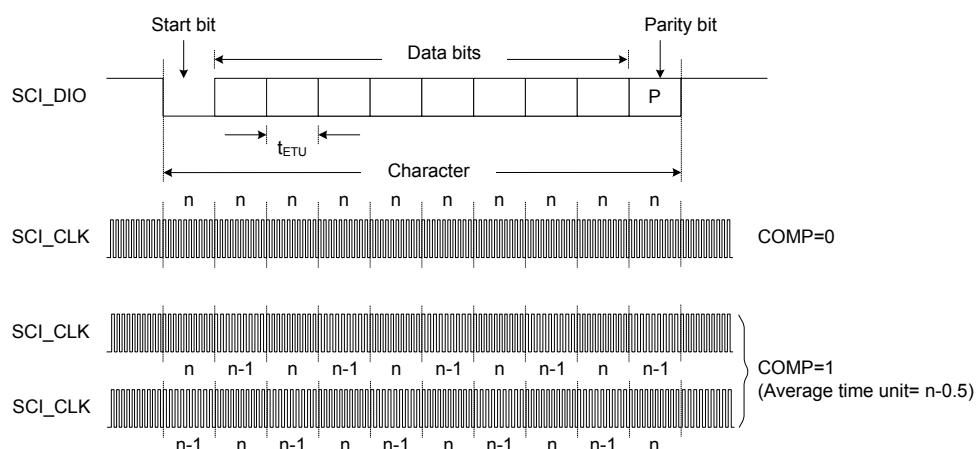
**Table 59. Possible ETU Values Obtained with the F<sub>i</sub>/D<sub>i</sub> Ratio**

<b>F<sub>i</sub> D<sub>i</sub></b>	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	11.62	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.

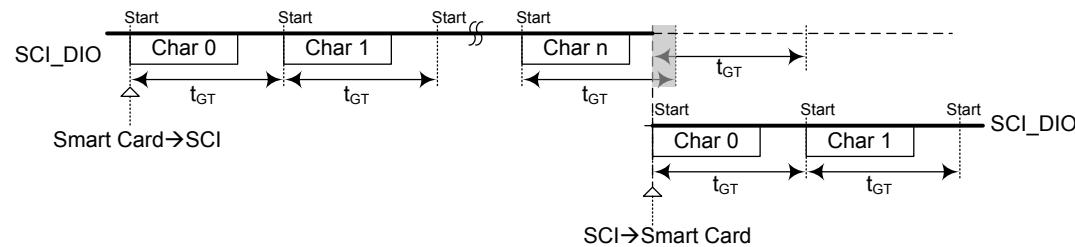
If the calculation result is close to the value of (n-0.5), the compensation mode should be enabled by setting the compensation enable control bit, COMP, in the ETUR register to 1 for successful data transfer. When the result is close to the value of (n-0.5) and the compensation mode is enabled, the 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 162. 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 163. Guard Time Duration**

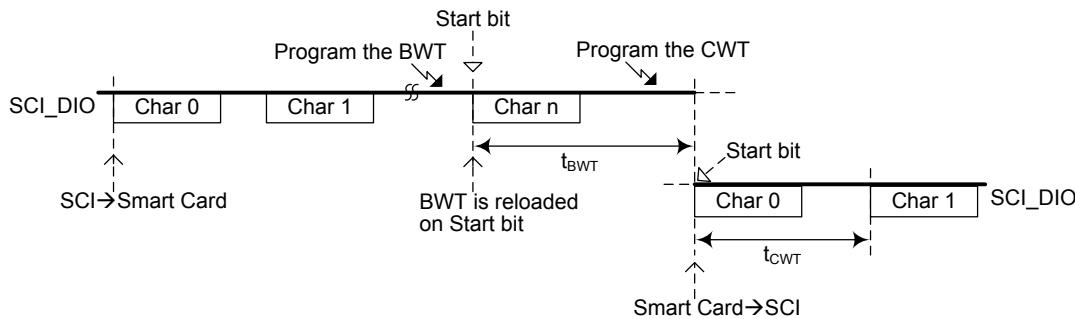
### Waiting Time Counter

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 164. 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.

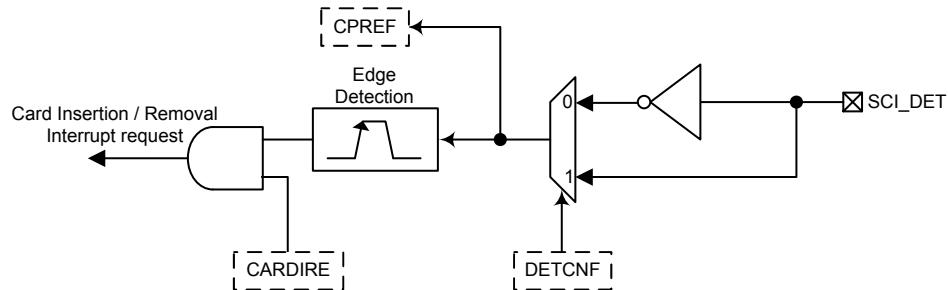
The SCI serial data pin, named SCI\_DIO, can be controlled by the SCI hardware circuitry or the software control bits depending upon whether the SCI is operated in the SCI Transfer Mode or in the Manual Mode. The mode selection is determined by the SCIM bit in the CR register. The SCI\_DIO pin status 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 165. 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

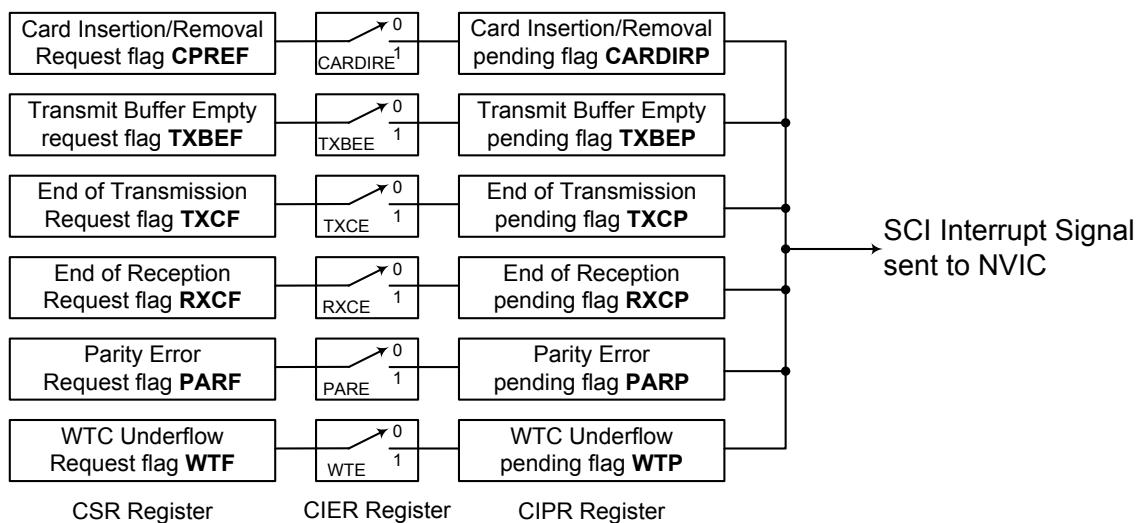
If the direction convention used by the Smart Card is the same as the convention used by the SCI, the SCI will generate a reception interrupt if the reception interrupt is enabled without a parity error flag. Otherwise, the SCI will generate a reception interrupt and the parity error flag will be asserted. 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 166. SCI Interrupt Structure**

## PDMA Interface

The PDMA interface is integrated in the SCI module. The PDMA function can be enabled by setting the TXDMA or RXDMA bit to 1 in the transmitter or receiver mode respectively. When the transmit buffer is empty which results in the transmit buffer empty flag, TXBEF, being asserted and the TXDMA bit is set to 1, the PDMA function will be activated to move data from a certain memory location into the SCI Transmit buffer. Similarly, when the SCI receives a character which results in the character received flag, RXCF, being asserted and the RXDMA bit is set to 1, the PDMA function will be activated to move data from the SCI Receive buffer to a specific memory location.

For a more detailed descriptions on the PDMA configurations, refer to the PDMA chapter.

## Register Map

There are several registers associated with the Smart Card function. Some of these registers control the SCI overall function as well as the interrupts, while some of the registers contain the status bits which indicate the Smart Card data transfer situation and error conditions. Also there are two registers for the SCI transmission and reception respectively to store the data received from or to be transmitted to the external Smart Card. The following table shows the SCI register list and reset values.

**Table 60. SCI Register Map**

Register	Offset	Description	Reset Value
<b>SCI Base Address = 0x4004_3000</b>			
CR	0x000	SCI Control Register	0x0000_0000
SR	0x004	SCI Status Register	0x0000_0080
CCR	0x008	SCI Contact Control Register	0x0000_0008
ETUR	0x00C	SCI Elementary Time Unit Register	0x0000_0174
GTR	0x010	SCI Guard Time Register	0x0000_000C
WTR	0x014	SCI Waiting Time Register	0x0000_2580
IER	0x018	SCI Interrupt Enable Register	0x0000_0000
IPR	0x01C	SCI Interrupt Pending Register	0x0000_0000
TXB	0x020	SCI Transmit Buffer	0x0000_0000
RXB	0x024	SCI Receive Buffer	0x0000_0000
PSCR	0x028	SCI Prescaler Register	0x0000_0000

## Register Descriptions

### SCI Control Register – CR

This register contains the SCI control bits.

Offset: 0x000

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions															
[9]	RXDMA	SCI reception DMA request enable 0: SCI reception DMA request is disabled 1: SCI reception DMA request is enabled															
[8]	TXDMA	SCI transmission DMA request enable control 0: SCI transmission DMA request is disabled 1: SCI transmission DMA request is enabled															
[6]	DETCKNF	Card switch type selection 0: Switch is normally opened if no card is present 1: Switch is normally closed if no card is present															
		<table border="1"> <thead> <tr> <th>DETCNT</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>	DETCNT	SCI_DET pin	STATUS	0	1	No card insert	0	0	Card insert	1	1	Card insert	1	0	No card insert
DETCNT	SCI_DET pin	STATUS															
0	1	No card insert															
0	0	Card insert															
1	1	Card insert															
1	0	No card insert															

This bit is set and cleared by the application program to configure the card detector switch type.

[5]	ENSCI	SCI finite state machine enable bit 0: SCI FSM is disabled and forced to its initial state 1: SCI FSM is enabled
-----	-------	--

[4]	RETRY	Character transfer repetition time selection for a parity error condition 0: Data transfer 5 times when parity error occurs 1: Data transfer 4 times when parity error occurs
-----	-------	---

The bit is available only when the CREP bit is set to 1. When this bit is set to 1, the data will be transmitted or received 4 times once a parity error occurs. If the bit is cleared to 0, the data will be transferred 5 times if a parity error occurs.

Bits	Field	Descriptions
[3]	SCIM	<p>SCI Mode Selection</p> <p>0: SCI data transfer in manual mode 1: SCI data transfer in SCI mode</p> <p>This bit is set and cleared by the application program to select the SCI data Transfer Mode. If it is cleared to 0, the SCI_DIO pin status is the same as the value of the CDIO bit in the CCR register. If it is set to 1, the SCI_DIO pin is driven by the internal SCI control circuitry. Before the data transfer type is switched from the Manual Mode to the SCI Mode, the CDIO bit must be set to 1 to avoid an SCI malfunction.</p>
[2]	WTEN	<p>Waiting Time Counter enable control</p> <p>0: Waiting Time Counter stops counting 1: Waiting Time Counter starts counting</p> <p>The WTEN bit is set and cleared by the application program. When the WTEN bit is cleared to 0, a write access to the WTR register will load the value into the waiting time counter. If it is set to 1, the waiting time counter is enabled and automatically reloaded with the value at each start bit occurrence.</p>
[1]	CREP	<p>Automatic character repetition enable control for a parity error condition</p> <p>0: No retry on parity error 1: Automatic retry on parity error</p> <p>The CREP bit is set and cleared by the application program. When the CREP bit is cleared to 0, both the RXCF and PARF flags will be set when a parity error occurs in the reception mode after the data is received. However, in the Transmission Mode, the PARF flag will be set but the TXCF flag will not be set when a parity error occurs. If the CREP bit is set to 1, a character transfer will automatically be activated 4 or 5 times depending upon the RETRY bit value. In the Transmission Mode the character will be re-transmitted if the transmitted data has a parity error. Here the parity error flag, PARF, will be set at the end of the 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

	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									
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 12.5%;">TXBEF</td> <td style="width: 12.5%;">CPREF</td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;">reserved</td> <td style="width: 12.5%;">WTF</td> <td style="width: 12.5%;">TXCF</td> <td style="width: 12.5%;">RXCF</td> <td style="width: 12.5%;">PARF</td> </tr> </table>									TXBEF	CPREF			reserved	WTF	TXCF	RXCF	PARF
TXBEF	CPREF			reserved	WTF	TXCF	RXCF	PARF									
Type/Reset	RO	1	RO	0	RO	0	W0C	0	RO	0	W0C	0					

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 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

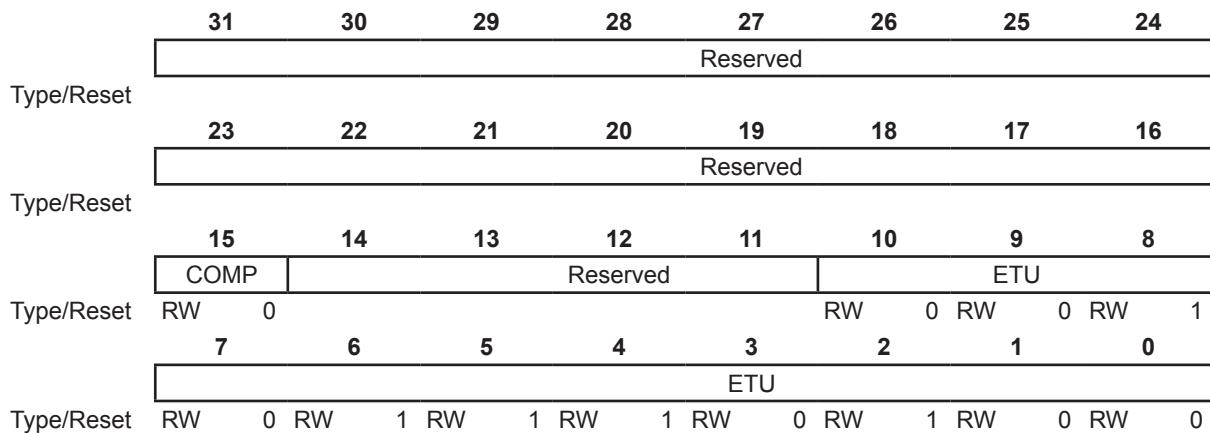
Bits	Field	Descriptions
[7]	CLKSEL	<p>Card Clock Selection</p> <p>0: The CCLK bit content is present on the external SCI_CLK pin</p> <p>1: The clock output on the external SCI_CLK pin is sourced from the <math>f_{PSC\_CK}</math> clock</p> <p>This bit is used to select the external SCI_CLK pin clock source. 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.</p>
[3]	CDIO	<p>SCI_DIO pin control</p> <p>0: SCI_DIO pin is logic level 0</p> <p>1: SCI_DIO pin in open-drain condition</p> <p>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.</p>
[2]	CCLK	<p>SCI_CLK pin control</p> <p>0: SCI_CLK pin is logic level 0</p> <p>1: SCI_CLK pin is logic level 1</p> <p>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.</p>

## 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



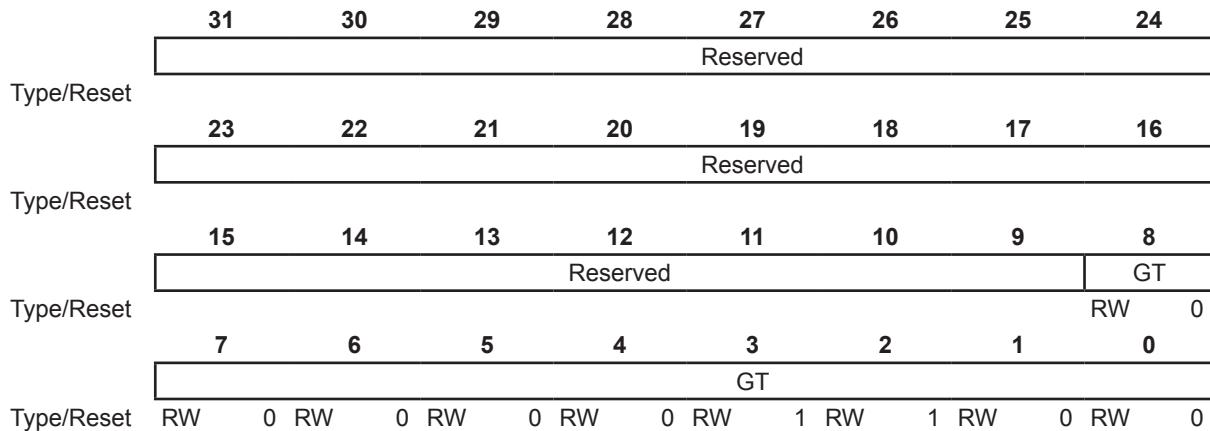
Bits	Field	Descriptions
[15]	COMP	<p>Elementary Time Unit Compensation mode enable control</p> <p>0: Compensation mode is disabled</p> <p>1: Compensation mode is enabled</p> <p>This bit is set and cleared by application program and used to control the ETU compensation function. For more details regarding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

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

Bits	Field	Descriptions
[23:0]	WT	Character Waiting Time value expressed in ETU (0/16777215). This field is configured by the application program to modify the waiting time duration. The reload conditions of the updated waiting time counter value are described in the waiting time counter section. Refer to the waiting time counter section for more details. Note that the WT value can range from 0x00_0000 to 0xFF_FFFF.

## SCI Interrupt Enable Register – IER

This register specifies the interrupt enable control bits for all of the interrupt events in the SCI.

Offset: 0x018

Reset value: 0x0000\_0000

	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.

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

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		
	TXBEP	CARDIRP		Reserved	WTP	TXCP	RXCP	PARP		
RC	0	RC	0		RC	0	RC	0	RC	

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>

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

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
TB								
Type/Reset	RW	0	RW	0	RW	0	RW	0

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

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
					RB			
Type/Reset	RW	0	RW	0	RW	0	RW	0

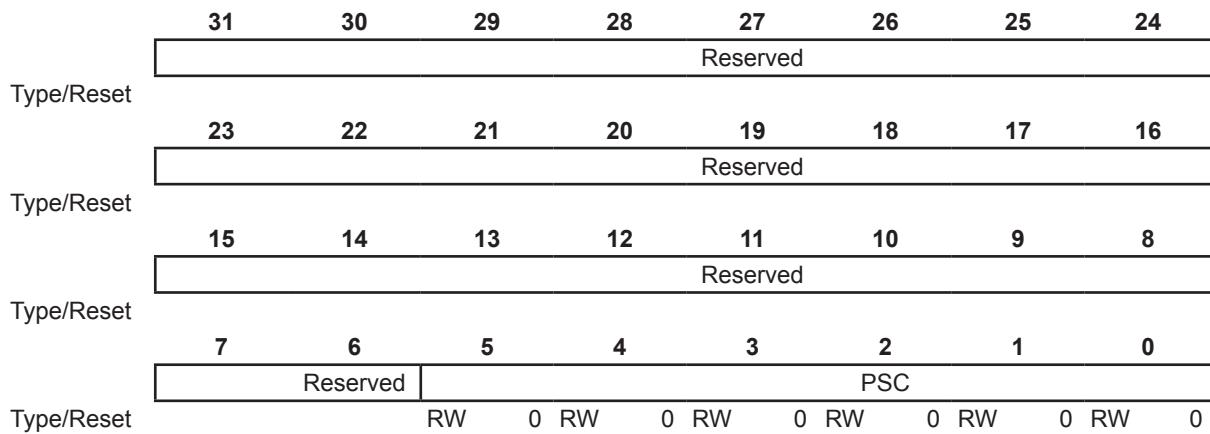
Bits	Field	Descriptions
[7:0]	RB	SCI Received data byte

## 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.

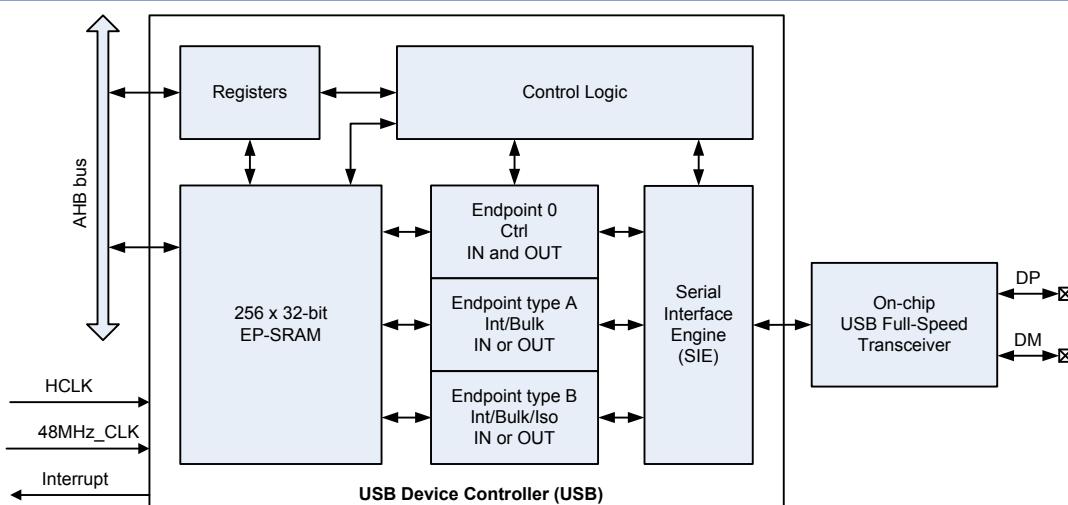


Figure 167. USB Block Diagram

## 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 61. Endpoint Characteristics**

Endpoint Number	Number Address	Transfer Type	Direction	Buffer Type
0	Fixed	Control	IN and OUT	Single buffering
1~3	Configurable	Interrupt/Bulk	IN or OUT	Single buffering
4~7	Configurable	Interrupt/Bulk/Isochronous	IN or OUT	Single or Double buffering

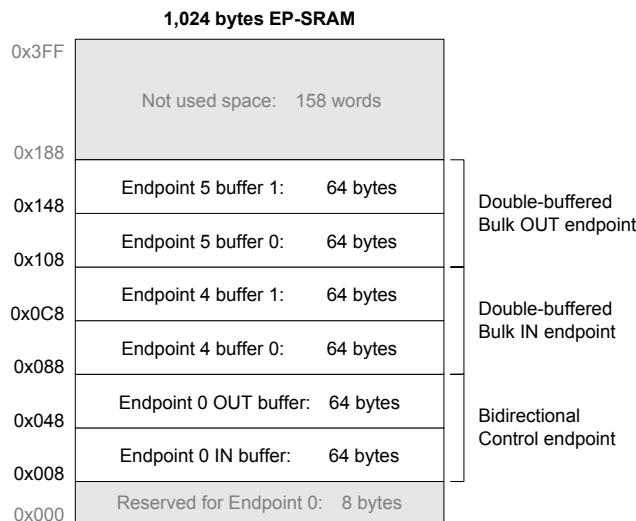
### EP-SRAM

The USB controller contains a dedicated memory space, EP-SRAM, which is used for the USB endpoint buffers. The EP-SRAM, which is connected to the APB bus, can be accessed by the CPU and PDMA. The EP-SRAM base address is 0x400A\_A000 with an offset which ranges from 0x000 to 0x3FF. The EP-SRAM first two words are reserved for Endpoint 0 to temporarily store the 8-byte SETUP data. Therefore the valid start address of the endpoint buffer should start from 0x008 and align to a 4-byte boundary. Each endpoint buffer size is programmable. The following table lists the maximum USB endpoint buffer size which is compliant with USB 2.0 full-speed device specification.

**Table 62. 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.



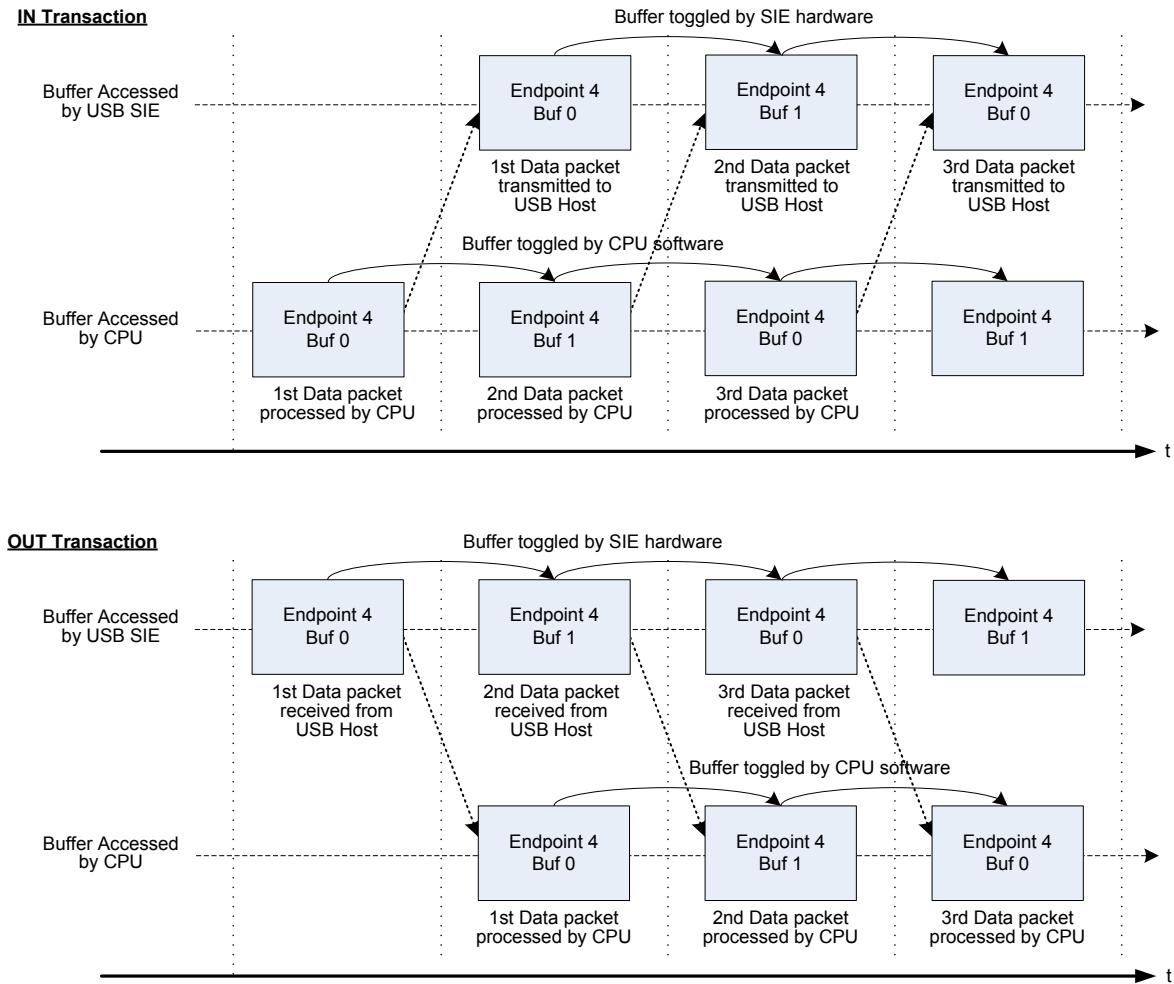
**Figure 168. Endpoint Buffer Allocation Example**

## 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 SEO 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 169. 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 uA 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 63. 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
USBEP3CFG	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
USBEP4CFG	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
USBEP5CFG	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
USBEP6CFG	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
USBEP7CFG	0x0B0	USB Endpoint 7 Configuration Register	0x1000_03FF

## Register Descriptions

### USB Control and Status Register – USBCSR

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

Offset: 0x000

Reset value: 0x0000\_00X6

	31	30	29	28	27	26	25	24	
	Reserved								
Type/Reset	23	22	21	20	19	18	17	16	
	Reserved								
Type/Reset	15	14	13	12	11	10	9	8	
	Reserved				DPWKEN	DPPUEN	SRAMRSTC	ADRSET	
Type/Reset					RW	0	RW	0	
	7	6	5	4	3	2	1	0	
Type/Reset	RXDM	RXDP	GENRSM	Reserved	LPMODE	PDWN	FRES	Reserved	
	RO	X	RO	X	RW	0	RW	1	

Bits	Field	Descriptions
[11]	DPWKEN	DP Wake Up Enable 0: Disable DP wake up 1: Enable DP wake up
[10]	DPPUEN	DP Pull Up Enable 0: Disable DP pull up 1: Enable DP pull up
[9]	SRAMRSTC	USB SRAM reset condition 0: Reset USB SRAM when (DP, DM) = (0,0) 1: User can access USB SRAM in spite of (DP, DM) state
[8]	ADRSET	Device Address Setting Control This bit is used to determine when the USB SIE updates the device address with the value of the USBDEVA 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.

Bits	Field	Descriptions
[5]	GENRSM	<p>Resume Request Generation Control</p> <p>This bit is used to generate a resume request which is sent to the USB host by writing 1 into this bit location. The USB remote wakeup function is always enabled. This bit will be cleared to 0 after a resume signal, sent by the USB host, has been received.</p>
[3]	LPMODE	<p>Low-power Mode Control</p> <p>This bit is used to determine the USB operating mode. Setting this bit will force the USB to enter the low-power mode. When USB bus traffic, known as a wakeup event, is detected by the hardware, this bit should be cleared by software.</p> <p>0: Exit the Low-power mode. 1: Enter the Low-power mode.</p>
[2]	PDWN	<p>Power Down Mode Control</p> <p>Setting this bit will power down the full-speed USB PHY transceiver. This will disconnect the USB PHY transceiver from the USB bus.</p> <p>0: Exit the Power-Down. 1: Enter the Power-Down mode.</p>
[1]	FRES	<p>Force USB Reset Control</p> <p>This bit is used to reset the USB circuitry. Setting this bit will force the USB into a reset state until the software clears it. A USB reset interrupt will be generated if the corresponding interrupt enable bit in the USBIER register is set to 1. All related USB registers are reset to their default values.</p> <p>0: Release USB reset. 1: Force USB reset.</p>

**Table 64. Resume Event Detection**

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

## USB Interrupt Enable Register – USBIER

This register specifies the USB interrupt enable control.

Offset: 0x004

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	EP7IE	EP6IE	EP5IE	EP4IE	EP3IE	EP2IE	EP1IE	EPOIE	
	RW	0	RW	0	RW	0	RW	0	
	7	6	5	4	3	2	1	0	
Type/Reset	Reserved								
		RW	0	RW	0	RW	0	RW	

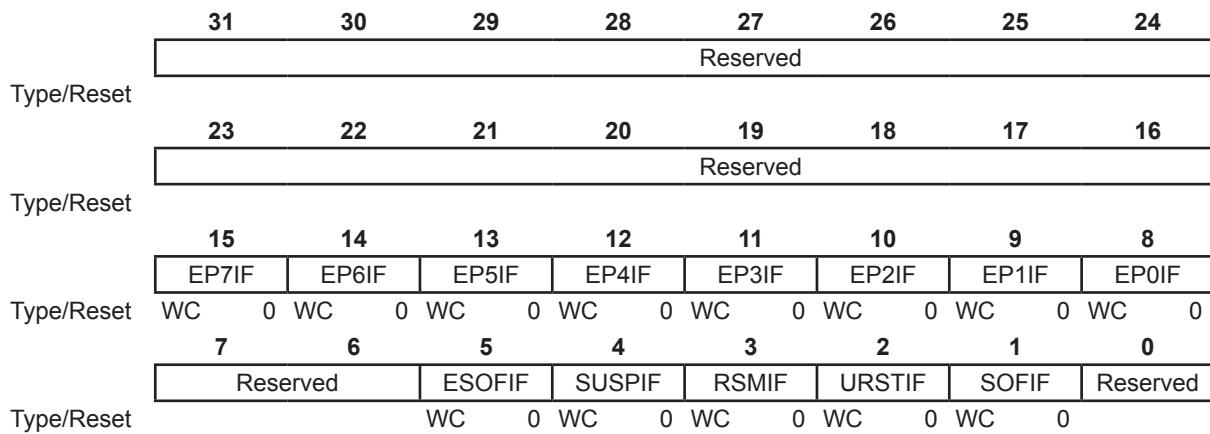
Bits	Field	Descriptions
[15:8]	EPnIE	Endpoint n Interrupt Enable Control (n = 0 ~ 7) 0: Disable interrupt 1: Enable interrupt
[5]	ESOFIE	Expected Start Of Frame (ESOF) Interrupt Enable Control 0: Disable ESOF interrupt 1: Enable ESOF interrupt
[4]	SUSPIE	Suspend Interrupt Enable Control 0: Disable suspend 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 still be asserted.

## USB Interrupt Status Register – USBISR

This register specifies the USB interrupt status.

Offset: 0x008

Reset value: 0x0000\_0000



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 1, this means that a device resume has occurred. This bit is cleared to 0 by writing 1.

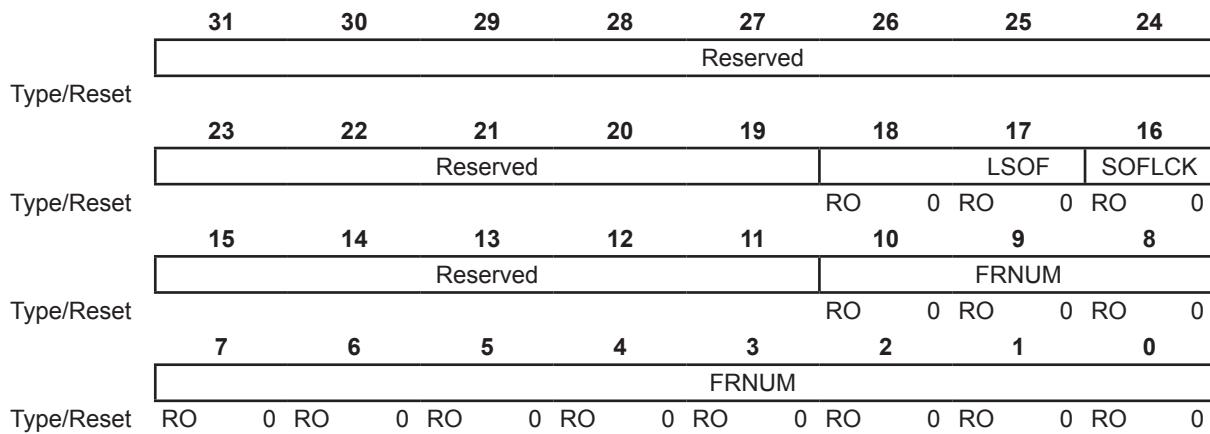
Bits	Field	Descriptions
[2]	URSTIF	<p>USB Reset Interrupt Flag</p> <p>This bit is set by the hardware when the USB reset has been detected. When a USB reset occurs, the internal protocol state machine will be reset and an USB reset interrupt will be generated if the URSTIE bit in the USBIER register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



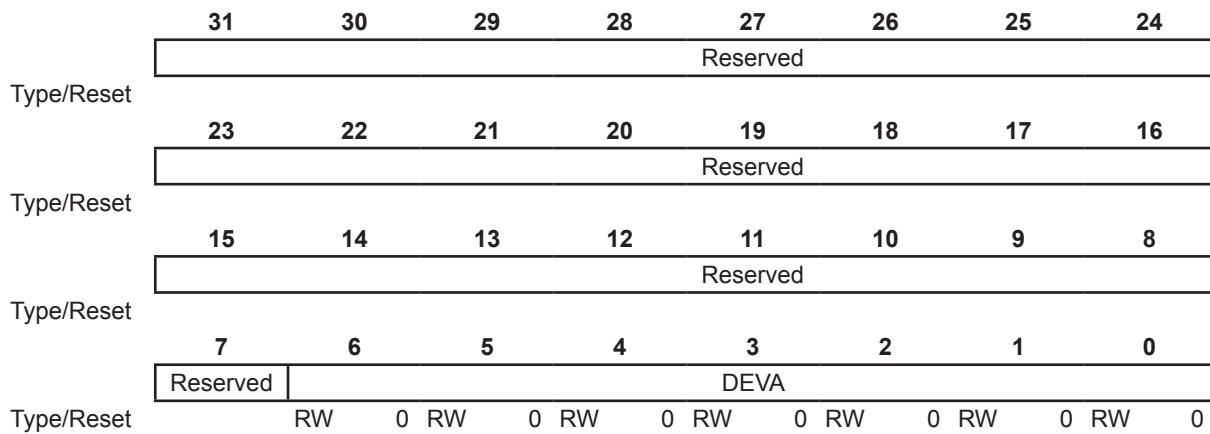
Bits	Field	Descriptions
[18:17]	LSOF	<p>Lost Start-of-Frame number</p> <p>These bits are written and incremented by 1 by the hardware each time the ESOFIF bit is set. It is used to count the number of lost SOF packets. When a SOF packet has been received, these bits are cleared.</p>
[16]	SOFLCK	<p>Start-of-Frame Lock Flag</p> <p>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.</p>
[10:0]	FRNUM	<p>Frame Number</p> <p>This field stores the frame number received from the USB host.</p>

## USB Device Address Register – USBDEVA

This register specifies the USB device address.

Offset: 0x010

Reset value: 0x0000\_0000



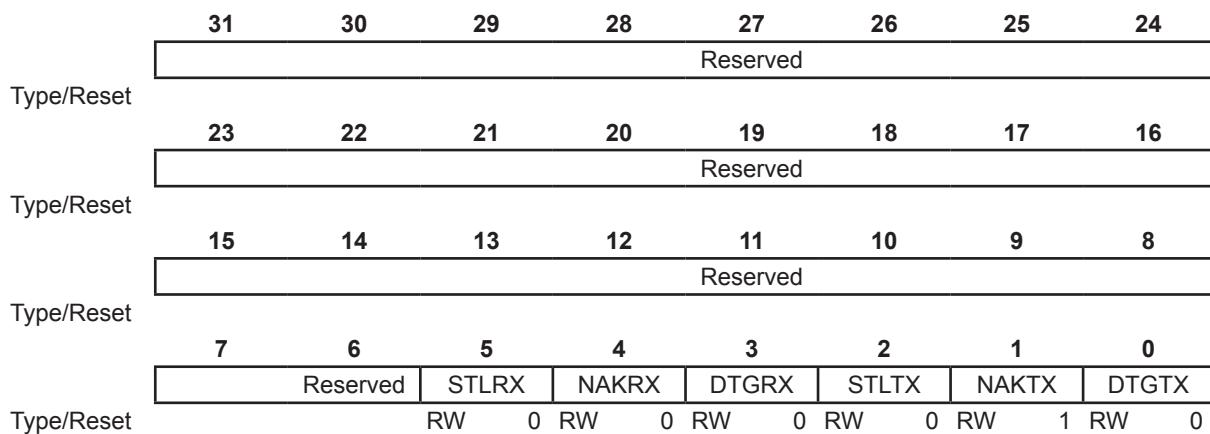
Bits	Field	Descriptions
[6:0]	DEVA	Device Address This field is used to specify the USB device address. This field is cleared when a USB reset event occurs.

## USB Endpoint 0 Control and Status Register – USBEP0CSR

This register specifies the Endpoint 0 control and status.

Offset: 0x014

Reset value: 0x0000\_0002



Bits	Field	Descriptions
[5]	STLRX	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.

Bits	Field	Descriptions
[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							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved				ZLRXIE	SDERIE	SDRXIE	STRXIE
	RW				0	RW	0	RW
Type/Reset	7	6	5	4	3	2	1	0
	UERIE	STLIE	NAKIE	IDTXIE	ITRXIE	ODOVIE	ODRXIE	OTRXIE
Type/Reset	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[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

<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>
[2]	ODOVIE	OUT Data Buffer Overrun Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[1]	ODRXIE	OUT Data Received Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[0]	OTRXIE	OUT Token Received Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt

## USB Endpoint 0 Interrupt Status Register – USBEP0ISR

This register specifies the Endpoint 0 interrupt status.

Offset: 0x01C

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	Reserved								
Type/Reset	23	22	21	20	19	18	17	16	
	Reserved								
Type/Reset	15	14	13	12	11	10	9	8	
	Reserved				ZLRXIF	SDERIF	SDRXIF	STRXIF	
Type/Reset				WC	0	WC	0	WC	
	7	6	5	4	3	2	1	0	
Type/Reset	WC	0	WC	0	WC	0	WC	0	
	UERIF	STLIF	NAKIF	IDTXIF	ITRXIF	ODOVIF	ODRXIF	OTRXIF	
	WC	0	WC	0	WC	0	WC	0	

Bits	Field	Descriptions
[11]	ZLRXIF	Zero Length Data Received Interrupt Flag This bit is set by the hardware when a zero length data packet is received. This bit is cleared by hardware when a SETUP Token is received or by writing 1.
[10]	SDERIF	SETUP Data Error Interrupt Flag This bit is set by the hardware when the SETUP data packet length is not 8 bytes. This bit is cleared by hardware when a SETUP Token is received or by writing 1.
[9]	SDRXIF	SETUP Data Received Interrupt Flag This bit is set by the hardware when a SETUP data packet from the USB host has been received. This bit is cleared by the hardware when a SETUP Token is received or by writing 1. If the received SETUP data is not accessed by the application software before the next SETUP packet is received, the SETUP data buffer will be overwritten.
[8]	STRXIF	SETUP Token Received Interrupt Flag This bit is set by the hardware when a SETUP token is received and is cleared by writing 1.
[7]	UERIF	USB Error Interrupt Flag This bit is set by the hardware when an error occurs during the Endpoint 0 transaction. This bit is cleared by hardware when a SETUP Token is received or by writing 1.
[6]	STLIF	STALL Transmitted Interrupt Flag This bit is set by the hardware when a STALL signal is sent in response to an IN or OUT transaction. This bit is cleared by hardware when a SETUP Token is received or by writing 1.
[5]	NAKIF	NAK Transmitted Interrupt Flag This bit is set by the hardware when a NAK signal is sent in response to an IN or OUT transaction. This bit is cleared by hardware when a SETUP Token is received or by writing 1.

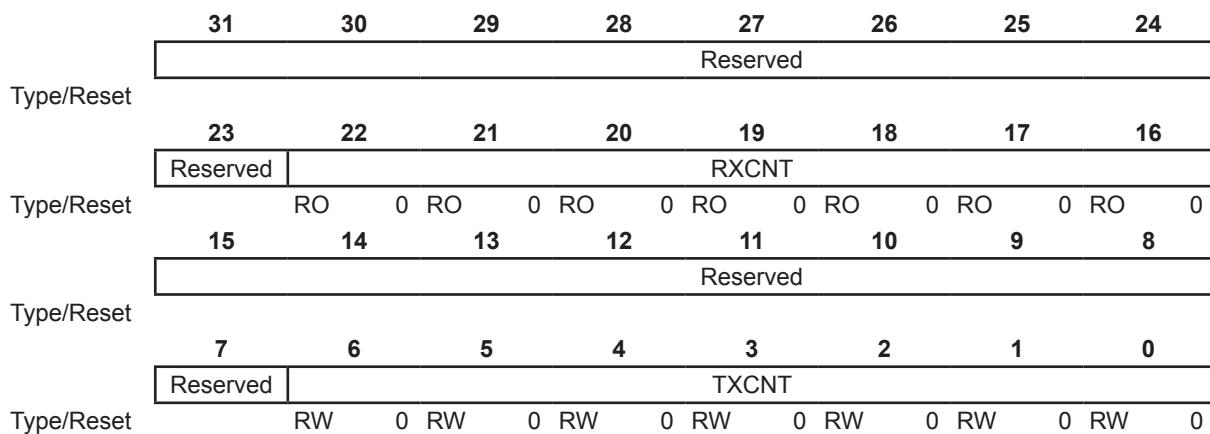
Bits	Field	Descriptions
[4]	IDTXIF	IN Data Transmitted Interrupt Flag This bit is set by the hardware when a data packet is transmitted to and then an ACK signal is received from the USB host. This bit is cleared by hardware when a SETUP Token is received or by writing 1.
[3]	ITRXIF	IN Token Received Interrupt Flag This bit is set by the hardware when the IN token is received from the USB host. This bit is cleared by hardware when a SETUP Token is received or by writing 1.
[2]	ODOVIF	OUT Data Buffer Overrun Interrupt Flag This bit is set by the hardware when the number of received data bytes is larger than the endpoint buffer size. This bit is cleared by hardware when a SETUP Token is received or by writing 1.
[1]	ODRXIF	OUT Data Received Interrupt Flag This bit is set by the hardware when a data packet is successfully received from and then an ACK signal is sent to the USB host. This bit is cleared by hardware when a SETUP Token is received or by writing 1.
[0]	OTRXIF	OUT Token Received Interrupt Flag This bit is set by the hardware when the OUT token is received from the USB host. This bit is cleared by hardware when a SETUP Token is received or by writing 1.

## USB Endpoint 0 Transfer Count Register – USBEP0TCR

This register specifies the Endpoint 0 data transfer byte count.

Offset: 0x020

Reset value: 0x0000\_0000



Bits	Field	Descriptions
[22:16]	RXCNT	Reception Byte Count The bit field contains the number of data bytes received by Endpoint 0 in the preceding SETUP transaction.
[6:0]	TXCNT	Transmission Byte Count The bit field contains the number of data bytes to be transmitted by Endpoint 0 in the next IN token. If the value of this field is zero, it indicates that a zero length packet will be sent.

## USB Endpoint 0 Configuration Register – USBEP0CFGR

This register specifies the Endpoint 0 configurations.

Offset: 0x024

Reset value: 0x8000\_0002

	31	30	29	28	27	26	25	24
Type/Reset	EPEN	Reserved			EPADR			
	RO	1			RO	0	RO	0
Type/Reset	23	22	21	20	19	18	17	16
	Reserved							EPLN
Type/Reset								
	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	EPBUFA							
	RW	0	RW	0	RW	0	RW	0

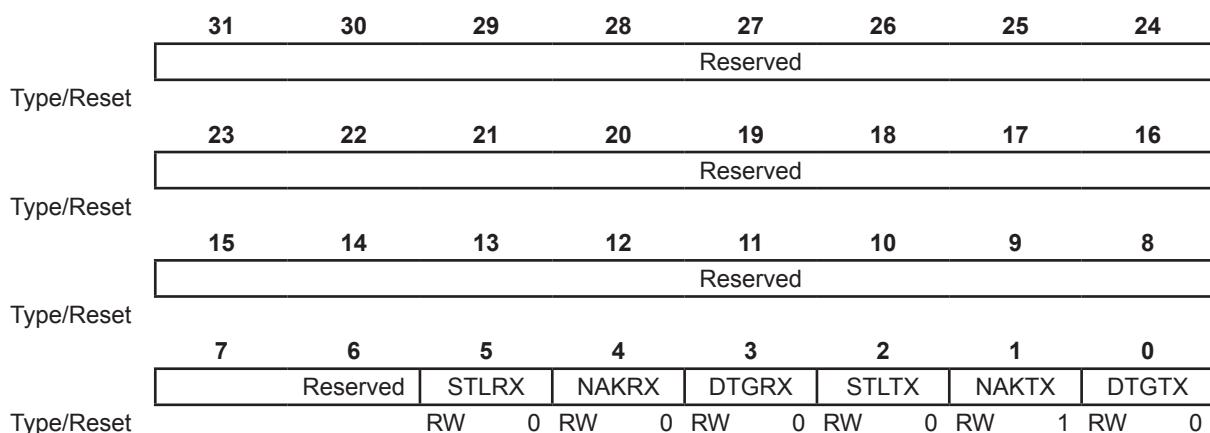
Bits	Field	Descriptions
[31]	EPEN	Endpoint Enable Control This bit is always set to 1 by the hardware circuitry to always enable Endpoint 0.
[27:24]	EPADR	Endpoint Address This field is always set to 0 by the hardware circuitry.
[16:10]	EPLN	Endpoint Buffer Length This field is used to specify the control 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 + EPLN

## USB Endpoint 1 ~ 3 Control and Status Register – USBEPnCSR, n = 1 ~ 3

This register specifies the Endpoint 1 ~ 3 control and status bit.

Offset: 0x028 (n = 1), 0x03C (n = 2), 0x050 (n = 3)

Reset value: 0x0000\_0002



Bits	Field	Descriptions
[5]	STLRX	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 – USBEPnIER, n = 1 ~ 3

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

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

Reset value: 0x0000\_0000

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

	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	WC	0 WC	0 WC	0 WC	0 WC	0 WC	0 WC	0 WC	
	UERIF	STLIF	NAKIF	IDTXIF	ITRXIF	ODOVIF	ODRXIF	OTRXIF	

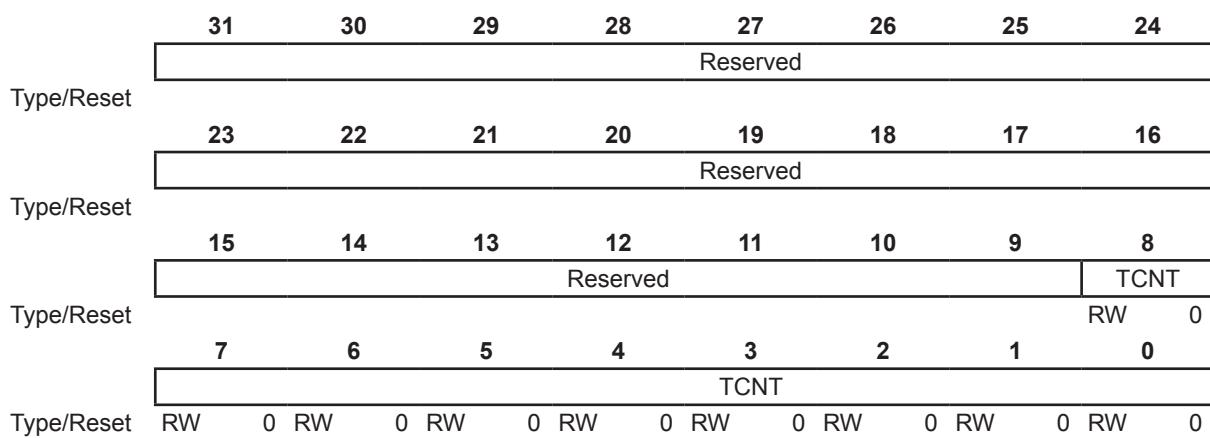
Bits	Field	Descriptions
[7]	UERIF	USB Error Interrupt Flag. This bit is set by the hardware when an error occurs during the transaction. Writing 1 into this status bit will clear it to 0.
[6]	STLIF	STALL Transmitted Interrupt Flag. This bit is set by hardware circuitry when a STALL-token is sent in response 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 – USBE<sub>n</sub>CFGR, 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			
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 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	0	EP_BUFI	EP_BUFO
	1	1	None*	EP_BUFI

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	0	EP_BUFI	EP_BUFO
	1	1	None*	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.

Bits	Field	Descriptions
[0]	DTGTX	<p>Data Toggle bit for transmission transfers.</p> <p>If the endpoint is not used for Isochronous transfer, this bit is available for usage. This bit contains the required value of the data toggle bit (0 = DATA0, 1 = DATA1) for the next data packet to be transmitted. When the current data packet is transmitted by the USB device and the corresponding ACK signal sent from the USB host is received, the hardware circuitry will toggle this bit and then the next data packet will be transmitted. If the endpoint is used for Isochronous transfer, this bit is not used since no data toggling is used and only the DATA0 packet will be transferred for normal Isochronous transfer.</p> <p>This bit can be read and written and can only be toggled by writing 1. It can also be toggled by the software to initialize its value under certain conditions.</p>

## USB Endpoint 4 ~ 7 Interrupt Enable Register – USBEPnIER, n = 4 ~ 7

This register specifies the Endpoint 4 ~ 7 interrupt enable control bits.

Offset: 0x068 (n = 4), 0x07C (n = 5), 0x090 (n = 6), 0x0A4 (n = 7)

Reset value: 0x0000\_0000

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		
	UERIE	STLIE	NAKIE	IDTXIE	ITRXIE	ODOVIE	ODRXIE	OTRXIE		
Type/Reset	RW	0	RW	0	RW	0	RW	0	0	

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

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		
	UERIF	STLIF	NAKIF	IDTXIF	ITRXIF	ODOVIF	ODRXIF	OTRXIF		
Type/Reset	WC	0 WC	0 WC	0 WC	0 WC	0 WC	0 WC	0 WC	0	

Bits	Field	Descriptions
[7]	UERIF	USB Error Interrupt flag This bit is set by the hardware 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
	23	22	21	20	19	18	17	16
Type/Reset	TCNT1					RW	0	RW
	RW	0	RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8
Type/Reset	Reserved					TCNT0		
						RW	0	RW
	7	6	5	4	3	2	1	0
Type/Reset	TCNT0					RW	0	RW
	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[25:16]	TCNT1	<p>Buffer 1 Transfer Byte Count</p> <p>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.</p>
[9:0]	TCNT0	<p>Buffer 0 Transfer Byte Count</p> <p>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.</p>

## 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	EPEN	Reserved	EPTYPE	EPDIR			EPADR	
	RW	0	RW	1	RW	0	RW	0
	23	22	21	20	19	18	17	16
Type/Reset	SDBS		Reserved				EPLN	
	RW	0	RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8
Type/Reset			EPLN				EPBUFA	
	RW	0	RW	0	RW	0	RW	1
	7	6	5	4	3	2	1	0
Type/Reset			EPBUFA					
	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 EPLN value: - Endpoint Buffer 0 start address is EPBUFA - Endpoint Buffer 1 start address is (EPBUFA + EPLN)
[19:10]	EPLN	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 Peripheral Direct Memory Access (PDMA)

## Introduction

The Peripheral Direct Memory Access circuitry, PDMA, provides 8 unidirectional channels for dedicated peripherals to implement the peripheral-to-memory and memory-to-peripheral data transfer. The memory-to-memory data transfer such as the FLASH-to-SRAM or SRAM-to-SRAM type is also supported and requested by the application program. Each PDMA channel configuration is independent. The PDMA channel transfer is split into multiple block transactions and the size of a block is equal to the block length multiplied by the data width.

## Features

- 8 unidirectional PDMA channels
- Memory-to-peripheral, peripheral-to-memory and memory-to-memory data transfer
- 8-bit, 16-bit and 32-bit width data transfer
- Software and hardware requested data transfer with configurable channel priority
- Linear, circular and non-increment address modes
- 4 transfer event flags - Transfer complete, Half Transfer, Block End and Transfer Error
- Auto-Reload function

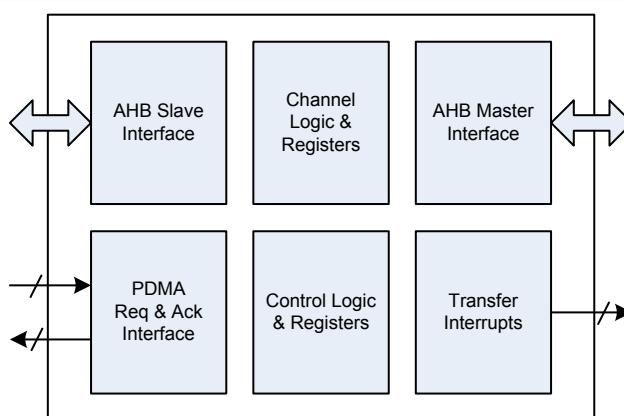


Figure 170. PDMA Block Diagram

## Functional Description

### AHB Master

The PDMA is an AHB master connected to other AHB peripherals such as the FLASH memory, the SRAM memory and the AHB-to-APB bridges through the bus-matrix. The Cortex™-M3 and PDMA can access different AHB slaves at the same time via the bus-matrix.

### PDMA Channel

There are 8 unidirectional PDMA channels used to support data transfer between the peripherals and the memory. The configuration and operation of each PDMA channel is independent. For a bidirectional transfer application, two PDMA channels are required. Each PDMA channel is designed to support the dedicated multiple peripherals with the same registers. Therefore, one PDMA channel only can service one peripheral at the same time. The related registers of the PDMA channel are limited to be accessed with 32-bit operation, otherwise a system hard fault event will occur.

### PDMA Request Mapping

The multiple requests from the peripherals (ADC, SPI0 , I2C, USART and so on) are simply logically ANDed before entering the PDMA, that means that only one request must be enabled at a time in each PDMA channel. Refer to Figure 171: PDMA request mapping architecture and detail peripheral IP requests mapping table is show as the Table 1. The peripheral DMA requests can be independently activated/de-activated by programming the DMA control bit in the registers of the corresponding peripheral.

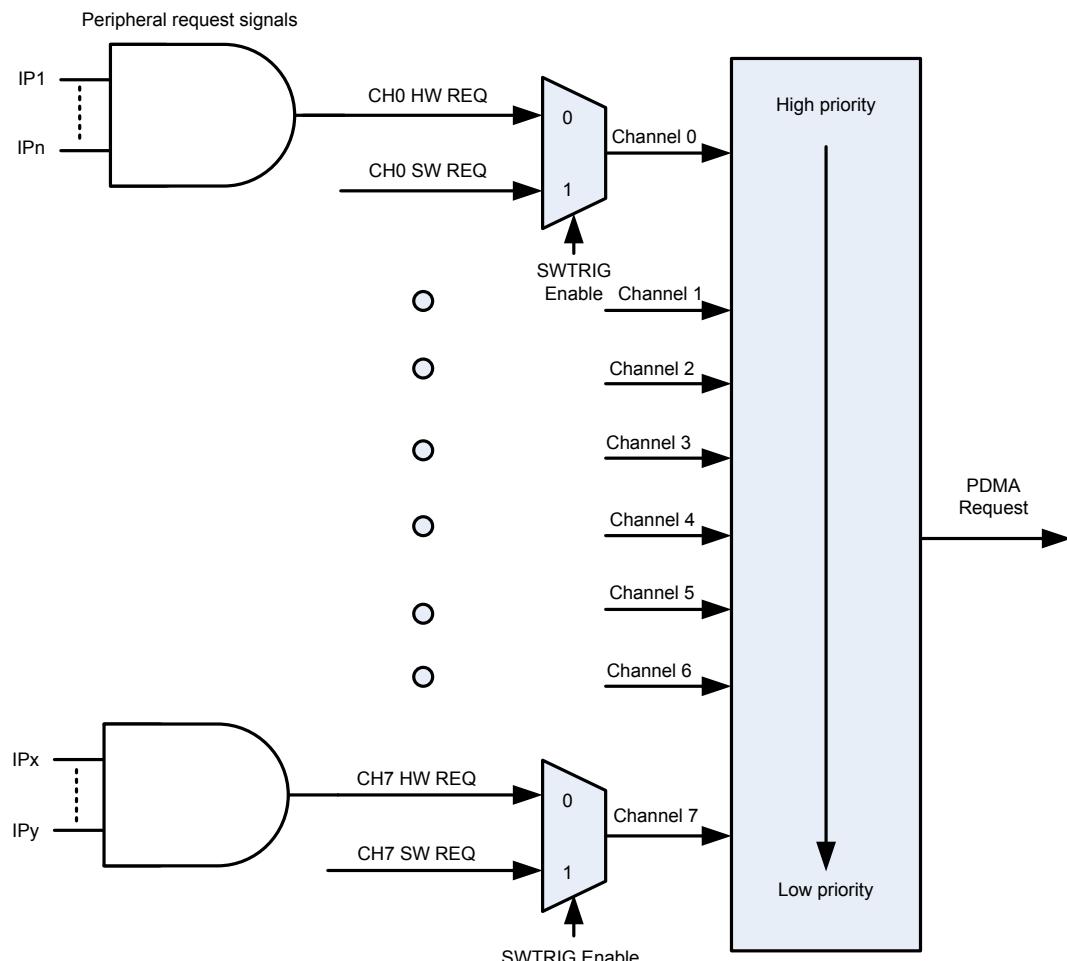


Figure 171. PDMA Request Mapping Architecture

Table 65. PDMA Channel Assignments

IP (x=0,1)	PDMA Channel Number							
	CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7
ADC	ADC							
SPIx	SPI0_RX	SPI0_TX			SPI1_RX	SPI1_TX		
USARTx			USR0_RX	USR0_TX			USR1_RX	USR1_RX
UARTx	UR0_RX	UR0_TX			UR1_RX	UR1_TX		
SCI							SCI_RX	SCI_TX
I2Cx			I2C0_RX		I2C0_TX		I2C1_RX	I2C1_TX
MCTMx	MT0_CH0	MT1_CH0	MT0_CH1	MT0_CH3	MT0_CH2	MT0_UEV1	MT1_UEV2	MT1_TRIG
		MT0_TRIG	MT1_CH2	MT1_UEV1	MT1_CH1	MT1_CH3		
GPTMx	GT0_CH1	GT0_UEV	GT0_CH2	GT0_CH0	GT1_CH0	GT1_CH1	GT1_CH2	GT1_CH3
	GT0_CH3			GT0_TRIGGER		GT1_TRIGGER		
I2S			I2S_RX	I2S_TX				

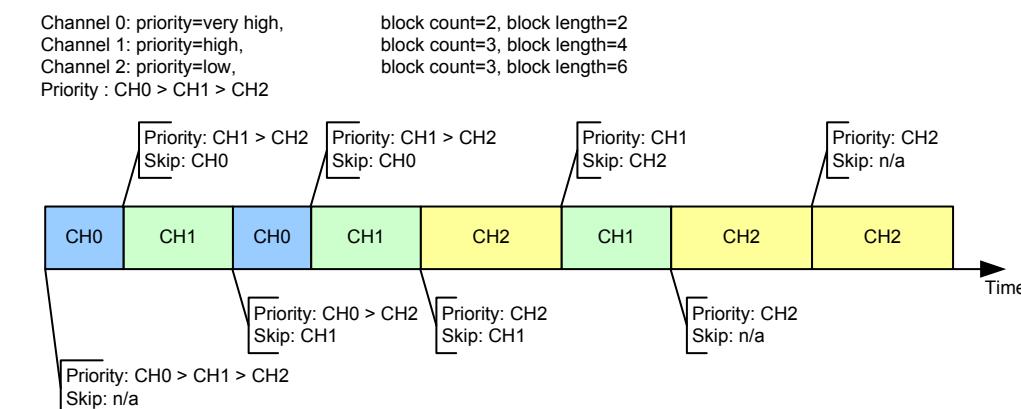
## Channel transfer

A PDMA channel transfer is split into multiple block transactions with PDMA arbitration occurring at the end of each block transaction. Although these 8 channel transfers can all be activated, there is only one block transaction being transferred through the bus at a time. The channel transfer sequence depends upon the channel priority setting of each PDMA channel. The total transfer size is calculated from the block transaction count and block size. The block size is equal to the product of the block length and data bit width. For an efficient transfer, it is recommended that the block length is set as a multiple of 4.

## Channel Priority

The PDMA provides four priority levels, known as very high, high, medium and low, which can be configured by the application software. The PDMA also provides two methods to determine the channel priority. One is determined by application software configuration and the other is determined by the fixed hardware channel number. The PDMA arbitration processor will first check the software configuring channel priority level used to request the PDMA to provide the data transfer services. If more than one channel has the same priority, the channel with a smaller channel number will have priority over one with a larger channel number after arbitration.

Note that the highest priority channel will not occupy the PDMA service all the time when other lower priority channel requests are pending. The highest priority channel will be skipped for one block transaction time duration after one block transaction is complete. Then a block transaction requested by the second priority channel will be performed. After a block transaction of the second priority channel is complete, the PDMA arbitration processor will re-check all of the requested channel priority with the exception of the second priority channel since the second priority channel will be excluded after the end of a block transaction. Therefore, a block data transaction of the higher priority channel will be serviced and this channel will be excluded from the priority arbitration at the end of the block transaction. The PDMA will keep transferring the data using the method described above until all of the requested channel data transfer is complete. Refer to the accompanying figure for an example which shows the PDMA channel arbitration and scheduling.



**Figure 172. PDMA Channel Arbitration and Scheduling Example**

## Transfer Request

For a peripheral-to-memory or memory-to-peripheral transfer, one peripheral hardware request will trigger one block transaction of the dedicated PDMA channel. However, a complete data transfer of the relevant dedicated PDMA channel will be triggered when a software request occurs. It is recommended that the PDMA channel is configured to have a lower priority level and a smaller block length which is requested by the software for memory-to-memory data copy applications.

## Address Mode

The PDMA provides three kinds of address modes which are the linear address, circular address and fixed address modes. These different address modes are used to support different kinds of source and destination address arrangements. The following table shows the detailed address mode combinations.

**Table 66. PDMA Address Modes**

Source Address Mode	Destination Address Mode
Linear Increment / Decrement Address	Linear Increment / Decrement Address
Linear Increment / Decrement Address	Circular Increment / Decrement Address
Linear Increment / Decrement Address	Fixed Address
Circular Increment / Decrement Address	Linear Increment / Decrement Address
Circular Increment / Decrement Address	Circular Increment / Decrement Address
Fixed Address	Linear Increment / Decrement Address
Fixed Address	Fixed Address

### Linear Address Mode

After data is transferred, the current address will be increased or decreased by 1, 2 or 4 depending upon the data bit width setting.

### Circular Address Mode

After data is transferred, the current address will be increased or decreased by 1, 2 or 4 depending upon the data bit width setting. When a block transaction is complete, the current address is loaded with the configured start address.

### Fixed Address Mode

After data is transferred, the current address remains unchanged.

## Auto-Reload

When the auto-reload control bit, AUTORLn, in the PDMA channel n control register named PDMACHnCR is set, both the channel n current address stored in the PDMACHnCADR register and the channel n current transfer size in the PDMACHnCTSR register will be automatically reloaded with the corresponding start value after the current PDMA channel data transfer has totally completed. The channel n will still be activated and the next relative PDMA request can be serviced without any re-configuration using the application software.

## Transfer Interrupt

There are five transfer events during which the interrupts can be asserted for each PDMA channel. These are the block transaction end (BE), half-transfer (HT), transfer complete (TC), transfer error (TE) and global transfer event (GE). Setting the corresponding control bits in the PDMA interrupt enable register named PDMAIER will enable the relevant interrupt events. The global interrupt event, GE, will be generated if any of the four interrupt events including the BE, HT, TC or TE occurs. Clearing the BE, HT, TC or TE event flags will also clear the GE flag. Clearing the GE flag will automatically clear all other event flags. The TE interrupt event will occur when the PDMA accesses a system reserved address space or the PDMA receives a request but when the corresponding transfer size setting is equal to zero.

## Register Map

The following table shows the PDMA registers and the reset values.

**Table 67. PDMA Register Map**

Register	Offset	Description	Reset Value
<b>PDMA Base Address = 0x4009_0000</b>			
<b>PDMA Channel 0 Registers</b>			
PDMACH0CR	0x000	PDMA Channel 0 Control Register	0x0000_0000
PDMACH0SADR	0x004	PDMA Channel 0 Source Address Register	0x0000_0000
PDMACH0DADR	0x008	PDMA Channel 0 Destination Address Register	0x0000_0000
PDMACH0CADR	0x00C	PDMA Channel 0 Current Address Register	0x0000_0000
PDMACH0TSR	0x010	PDMA Channel 0 Transfer Size Register	0x0000_0000
PDMACH0CTSR	0x014	PDMA Channel 0 Current Transfer Size Register	0x0000_0000
<b>PDMA Channel 1 Registers</b>			
PDMACH1CR	0x018	PDMA Channel 1 Control Register	0x0000_0000
PDMACH1SADR	0x01C	PDMA Channel 1 Source Address Register	0x0000_0000
PDMACH1DADR	0x020	PDMA Channel 1 Destination Address Register	0x0000_0000
PDMACH1CADR	0x024	PDMA Channel 1 Current Address Register	0x0000_0000
PDMACH1TSR	0x028	PDMA Channel 1 Transfer Size Register	0x0000_0000
PDMACH1CTSR	0x02C	PDMA Channel 1 Current Transfer Size Register	0x0000_0000
<b>PDMA Channel 2 Registers</b>			
PDMACH2CR	0x030	PDMA Channel 2 Control Register	0x0000_0000
PDMACH2SADR	0x034	PDMA Channel 2 Source Address Register	0x0000_0000
PDMACH2DADR	0x038	PDMA Channel 2 Destination Address Register	0x0000_0000
PDMACH2CADR	0x03C	PDMA Channel 2 Current Address Register	0x0000_0000
PDMACH2TSR	0x040	PDMA Channel 2 Transfer Size Register	0x0000_0000
PDMACH2CTSR	0x044	PDMA Channel 2 Current Transfer Size Register	0x0000_0000
<b>PDMA Channel 3 Registers</b>			
PDMACH3CR	0x048	PDMA Channel 3 Control Register	0x0000_0000
PDMACH3SADR	0x04C	PDMA Channel 3 Source Address Register	0x0000_0000
PDMACH3DADR	0x050	PDMA Channel 3 Destination Address Register	0x0000_0000
PDMACH3CADR	0x054	PDMA Channel 3 Current Address Register	0x0000_0000
PDMACH3TSR	0x058	PDMA Channel 3 Transfer Size Register	0x0000_0000
PDMACH3CTSR	0x05C	PDMA Channel 3 Current Transfer Size Register	0x0000_0000

Register	Offset	Description	Reset Value
<b>PDMA Channel 4 Registers</b>			
PDMACH4CR	0x060	PDMA Channel 4 Control Register	0x0000_0000
PDMACH4SADR	0x064	PDMA Channel 4 Source Address Register	0x0000_0000
PDMACH4DADR	0x068	PDMA Channel 4 Destination Address Register	0x0000_0000
PDMACH4CADR	0x06C	PDMA Channel 4 Current Address Register	0x0000_0000
PDMACH4TSR	0x070	PDMA Channel 4 Transfer Size Register	0x0000_0000
PDMACH4CTSR	0x074	PDMA Channel 4 Current Transfer Size Register	0x0000_0000
<b>PDMA Channel 5 Registers</b>			
PDMACH5CR	0x078	PDMA Channel 5 Control Register	0x0000_0000
PDMACH5SADR	0x07C	PDMA Channel 5 Source Address Register	0x0000_0000
PDMACH5DADR	0x080	PDMA Channel 5 Destination Address Register	0x0000_0000
PDMACH5CADR	0x084	PDMA Channel 5 Current Address Register	0x0000_0000
PDMACH5TSR	0x088	PDMA Channel 5 Transfer Size Register	0x0000_0000
PDMACH5CTSR	0x08C	PDMA Channel 5 Current Transfer Size Register	0x0000_0000
<b>PDMA Channel 6 Registers</b>			
PDMACH6CR	0x090	PDMA Channel 6 Control Register	0x0000_0000
PDMACH6SADR	0x094	PDMA Channel 6 Source Address Register	0x0000_0000
PDMACH6DADR	0x098	PDMA Channel 6 Destination Address Register	0x0000_0000
PDMACH6CADR	0x09C	PDMA Channel 6 Current Address Register	0x0000_0000
PDMACH6TSR	0x0A0	PDMA Channel 6 Transfer Size Register	0x0000_0000
PDMACH6CTSR	0x0A4	PDMA Channel 6 Current Transfer Size Register	0x0000_0000
<b>PDMA Channel 7 Registers</b>			
PDMACH7CR	0x0A8	PDMA Channel 7 Control Register	0x0000_0000
PDMACH7SADR	0x0AC	PDMA Channel 7 Source Address Register	0x0000_0000
PDMACH7DADR	0x0B0	PDMA Channel 7 Destination Address Register	0x0000_0000
PDMACH7CADR	0x0B4	PDMA Channel 7 Current Address Register	0x0000_0000
PDMACH7TSR	0x0B8	PDMA Channel 7 Transfer Size Register	0x0000_0000
PDMACH7CTSR	0x0BC	PDMA Channel 7 Current Transfer Size Register	0x0000_0000
<b>PDMA Global Register</b>			
PDMAISR0	0x120	PDMA Interrupt Status Register 0	0x0000_0000
PDMAISR1	0x124	PDMA Interrupt Status Register 1	0x0000_0000
PDMAICLR0	0x128	PDMA Interrupt Status Clear Register 0	0x0000_0000
PDMAICLR1	0x12C	PDMA Interrupt Status Clear Register 1	0x0000_0000
PDMAIER0	0x130	PDMA Interrupt Enable Register 0	0x0000_0000
PDMAIER1	0x134	PDMA Interrupt Enable Register 1	0x0000_0000

## Register Descriptions

### PDMA Channel n Control Register – PDMACHnCR, n=0~7

This register is used to specify the PDMA channel n data transfer configuration.

Offset: 0x000 (0), 0x018 (1), 0x030 (2), 0x048 (3), 0x060 (4), 0x078 (5), 0x090 (6), 0x0A8 (7)

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	Reserved								
Type/Reset		23	22	21	20	19	18	17	16
	Reserved								
Type/Reset		15	14	13	12	11	10	9	8
	Reserved					AUTORLn	FIXAENn	CHnPRI	
Type/Reset					RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	SRCAMODn	SRCAINCn	DSTAMODn	DSTAINCn	DWIDTHn		SWTRIGN	CHnEN	
	RW	0	RW	0	RW	0	RW	0	0

Bits	Field	Descriptions
[11]	AUTORLn	Channel n Auto Reload Enable Control 0: Disable Auto Reload function 1: Enable Auto Reload function  If this bit is set to 1 to enable the auto-reload function, the PDMACHnCADDR and PDMACHnCTSR registers will be reloaded with the relevant start value and the PDMA channel n will be activated when a transfer is complete. If this bit is cleared to 0, the PDMACHnCADDR and PDMACHnCTSR registers will remain unchanged and the PDMA channel n will be disabled after a transfer completion.
[10]	FIXAENn	Channel n Fixed Address Enable control 0: Disable fixed address function in the circular address mode 1: Enable fixed address function in the circular address mode  Note that this bit is only available when the source or destination address mode is set to be in the circular address mode. For example, the source address mode is set as in the linear address mode and the destination address mode is set as in the circular mode. If this bit is set to enable the fixed address function, then the source address mode will still be in the linear address but the destination address mode will be in the fixed address mode instead of the circular address mode.
[9:8]	CHnPRI	Channel n Priority 00: Low 01: Medium 10: High 11: Very high  The CHnPRI field is used to configure the channel priority using the application program. If there are more than one channels which have the same software configured priority level, the channel with the smaller channel number will have priority to transfer one block of data after the arbitration.

Bits	Field	Descriptions
[7]	SRCAMODn	Channel n Source Address Mode selection 0: Linear address mode 1: Circular address mode In the linear address mode, the current source address field value, CSADR, in the PDMACHnCADR register can be incremented or decremented, determined by the SRCAINCn bit value during a complete transfer. In the circular address mode, the CSADR field value can be incremented or decremented which is also determined by the SRCAINCn bit value during a block transfer and will be loaded with the lower 16-bit value of the PDMACHnSADR register when a block transaction has completed.
[6]	SRCAINCn	Channel n Source Address Increment control 0: Increment 1: Decrement This bit is used to determine whether the current source address in the CSADR field is increased or decreased during a complete transfer in the linear address mode or a block transfer in the circular address mode.
[5]	DSTAMODn	Channel n Destination Address Mode selection 0: Linear address mode 1: Circular address mode In linear address mode, the current destination address field value, CDADR, in the PDMACHnCADR register can be incremented or decremented, determined by the DSTAINCn bit value during a complete transfer. In the circular address mode, the CDADR field value can be incremented or decremented which is also determined by the DSTAINCn bit value during a block transfer and will be loaded with the lower 16-bit value of the PDMACHnDADR register when a block transfer has completed.
[4]	DSTAINCn	Channel n Destination Address Increment Control 0: Increment 1: Decrement This bit is used to determine if the current destination address in the CDADR field is increased or decreased during a complete transfer in the linear address mode or a block transfer in the circular address mode.
[3:2]	DWIDTHn	Data Bit Width selection 00: 8-bit 01: 16-bit 10: 32-bit 11: Reserved The field is used to select the data bit width of the corresponding PDMA channel n.
[1]	SWTRIGn	Software Trigger control 0: No operation 1: Software triggered transfer request Setting this bit will generate a memory-to-memory software transfer request on the corresponding PDMA channel n. It is automatically cleared when a transfer has completely finished.
[0]	CHnEN	Channel n Enable control 0: Disable the PDMA channel n 1: Enable the PDMA channel n Setting this bit will enable a software or hardware transfer request on the PDMA channel n. It is automatically cleared by hardware when a transfer has completed with the auto-reload function being disabled. However, if the AUTORLn bit is set to 1 to enable the auto-reload function, this bit will remain high to enable the PDMA channel n function for the next transfer request instead of automatically being cleared by hardware after a transfer has finished.

## PDMA Channel n Source Address Register – PDMACHnSADR, n=0~7

This register specifies the source address of the PDMA channel n.

Offset: 0x004 (0), 0x01C (1), 0x034 (2), 0x04C (3), 0x064 (4), 0x07C (5), 0x094 (6), 0x0AC (7)

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[31:0]	SADRn	Channel n Source Address The register is used to specify the 32-bit source address of the PDMA channel n.

## PDMA Channel n Destination Address Register – PDMACHnDADR, n=0~7

This register specifies the destination address of the PDMA channel n.

Offset: 0x008 (0), 0x020 (1), 0x038 (2), 0x050 (3), 0x068 (4), 0x080 (5), 0x098 (6), 0x0B0 (7)

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[31:0]	DADRn	Channel n Destination Address The register is used to specify the 32-bit destination address of the PDMA channel n.

## PDMA Channel n Current Address Register – PDMACHnCADR, n=0~7

This register is used to indicate the lower 16-bit of the PDMA channel n current source and destination address.

Offset: 0x00C (0), 0x024 (1), 0x03C (2), 0x054 (3), 0x06C (4), 0x084 (5), 0x09C (6), 0x0B4 (7)

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[31:16]	CSADRn	<p>Channel n Current Source Address</p> <p>The CSADRn field is a read only 16-bit address indicating the current address of the data being read. After the data has been read, the CSADRn field contains the next address which the data is to be read from. Writing a new value to the PDMACHnSADR register will update the CSADRn field with the lower 16-bit content of the PDMACHnSADR register rather than the complete 32-bit value in the PDMACHnSADR register. The CSADRn field will have 16-bit or 32-bit alignment according to the DWIDTHn field setting in the PDMACHnCR register.</p> <p>For example:</p> <p>If the DWIDTHn field is set as 16-bit, the last bit of the CSADRn field will be 0.</p> <p>If the DWIDTHn field is set as 32-bit, the last 2 bits of the CSADRn field will be 0.</p>
[15:0]	CDADRN	<p>Channel n Current Destination Address</p> <p>The CDADRN field is a read only 16-bit address indicating the current address of the data being written. After the data has been written, the CDADRN field contains the next address where the data is to be written. Writing a new value to the PDMACHnDADR register will update the CDADRN field with the lower 16-bit content of the PDMACHnDADR register rather than the whole 32-bit value in the PDMACHnDADR register. The CDADRN field will have 16-bit or 32-bit alignment according to the DWIDTHn field setting in the PDMACHnCR register.</p> <p>For example:</p> <p>If the DWIDTHn field is set as 16-bit, the last bit of the CDADRN field will be 0.</p> <p>If the DWIDTHn field is set as 32-bit, the last 2 bits of the CDADRN field will be 0.</p>

## PDMA Channel n Transfer Size Register – PDMACHnTSR, n=0~7

This register is used to specify the block transaction count and block transaction length.

Offset: 0x010 (0), 0x028 (1), 0x040 (2), 0x058 (3), 0x070 (4), 0x088 (5), 0x0A0 (6), 0x0B8 (7)

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
BLKCNTn[15:8]								
Type/Reset	RW	0	RW	0	RW	0	RW	0
BLKCNTn[7:0]								
Type/Reset	RW	0	RW	0	RW	0	RW	0
BLKLENn[15:8]								
Type/Reset	RW	0	RW	0	RW	0	RW	0
BLKLENn[7:0]								
Type/Reset	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[31:16]	BLKCNTn	Channel n Block Transaction Count BLKCNTn represents the number of block transactions for a channel n complete transfer. The capacity of a complete transfer is the product of the BLKCNTn and BLKLENn values. The maximum BLKCNTn value is 65,535.
[15:0]	BLKLENn	Channel n Block Length The BLKLENn represents the length of a data block. The data width is defined by the DWIDTHn field in the PDMACHnCR register. The maximum BLKLENn value is 65,535.

## PDMA Channel n Current Transfer Size Register – PDMACHnCTSR, n=0~7

This register is used to indicate the current block transaction count and current block length.

Offset: 0x014 (0), 0x02C (1), 0x044 (2), 0x05C (3), 0x074 (4), 0x08C (5), 0x0A4 (6), 0x0BC (7)

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
CBLKCNTn[15:8]								
Type/Reset	RO	0	RO	0	RO	0	RO	0
CBLKCNTn[7:0]								
Type/Reset	RO	0	RO	0	RO	0	RO	0
CBLKLENn[15:8]								
Type/Reset	RO	0	RO	0	RO	0	RO	0
CBLKLENn[7:0]								
Type/Reset	RO	0	RO	0	RO	0	RO	0

Bits	Field	Descriptions
[31:16]	CBLKCNTn	Channel n Current Block Count The CBLKCNTn field is a 16-bit read-only value indicating the number of data blocks that remain to be transferred. After a data block has transferred completely, the CBLKCNTn value will be decremented by 1. Writing a new value to the BLKCNTn field in the PDMACHnTSR register will update the CBLKCNTn field value.
[15:0]	CBLKLENn	Channel n Current Block Length The CBLKLENn field is a 16-bit read-only value that indicates how much data in the block remains to be transferred. After a block of data has transferred completely, the CBLKLENn field will be reloaded with the BLKLENn value. Writing a new value to the BLKLENn field in the PDMACHnTSR register will update the CBLKLENn field value.

## PDMA Interrupt Status Register 0 – PDMAISR0

This register is used to indicate the corresponding interrupt status of the PDMA channel 0~5.

Offset: 0x120

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved		TEISTA5	TCISTA5	HTISTA5	BEISTA5	GEISTA5	TEISTA4
	RO	0	RO	0	RO	0	RO	0
Type/Reset	23	22	21	20	19	18	17	16
	TCISTA4	HTISTA4	BEISTA4	GEISTA4	TEISTA3	TCISTA3	HTISTA3	BEISTA3
Type/Reset	RO	0	RO	0	RO	0	RO	0
Type/Reset	15	14	13	12	11	10	9	8
	GEISTA3	TEISTA2	TCISTA2	HTISTA2	BEISTA2	GEISTA2	TEISTA1	TCISTA1
Type/Reset	RO	0	RO	0	RO	0	RO	0
Type/Reset	7	6	5	4	3	2	1	0
	HTISTA1	BEISTA1	GEISTA1	TEISTA0	TCISTA0	HTISTA0	BEISTA0	GEISTA0
Type/Reset	RO	0	RO	0	RO	0	RO	0

Bits	Field	Descriptions
[29], [24], [19], [14], [9], [4]	TEISTAn	Channel n Transfer Error Interrupt Status (n = 0 ~ 5) 0: No Transfer Error occurs 1: Transfer Error occurs This bit is set by hardware and is cleared by writing a “1” into the corresponding interrupt status clear bit in the PDMAISR0 register. A Transfer error will occur when the PDMA accesses a system reserved address space or the PDMA receives a request but when the corresponding transfer capacity is equal to zero.
[28], [23], [18], [13], [8], [3]	TCISTAn	Channel n Transfer Complete Interrupt Status (n= 0 ~ 5) 0: No Transfer Completion Occurs 1: Transfer Completion Occurs This bit is set by hardware and is cleared by writing a “1” into the corresponding interrupt status clear bit in the PDMAISR0 register. The Transfer Completion event will occur when the PDMA has completed a data transfer task.
[27], [22], [17], [12], [7], [2]	HTISTAn	Channel n Half Transfer Interrupt Status (n= 0 ~ 5) 0: No Half Transfer Event Occurs 1: Half Transfer Event Occurs This bit is set by hardware and is cleared by writing a “1” into the corresponding interrupt status clear bit in the PDMAISR0 register. A Half Transfer event will occur when the PDMA has completed half of the data transfer task.
[26], [21], [16], [11], [6], [1]	BEISTAn	Channel n Block Transaction End Interrupt Status (n= 0 ~ 5) 0: No Block Transaction End Event Occurs 1: Block Transaction End Event Occurs This bit is set by hardware and is cleared by writing a “1” into the corresponding interrupt status clear bit in the PDMAISR0 register. A Block Transaction End event will occur when the PDMA completes a data block transaction task.

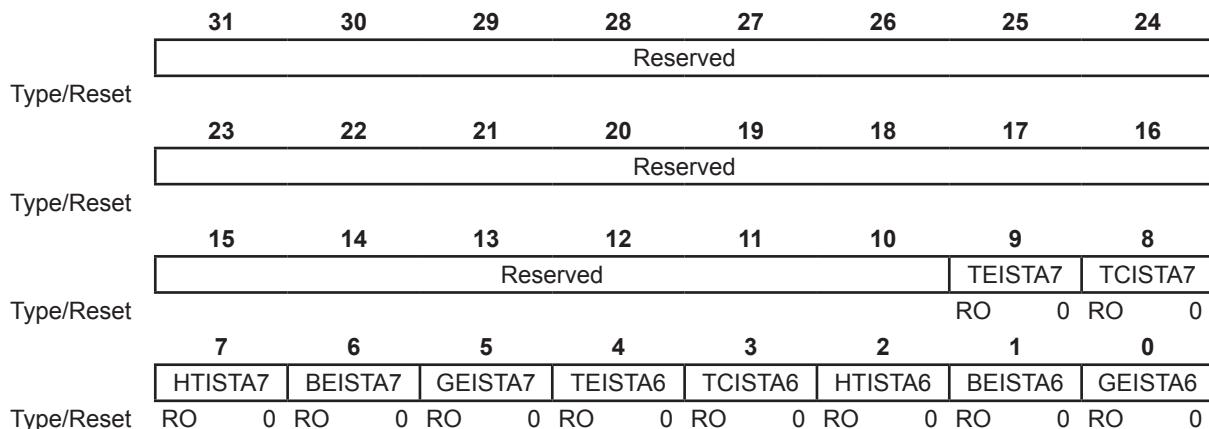
Bits	Field	Descriptions
[25], [20], [15], [10], [5], [0]	GEISTAn	Channel n Global Transfer Interrupt Status ( n= 0 ~ 5) 0: No TE, TC, HT or BE event occurs 1: TE, TC, HT, or BE event occurs
		This bit is set by hardware and is cleared by writing a “1” into the corresponding interrupt status clear bit, GEICLRn, in the PDMAISR0 register. A Global Transfer Event will occur if any of the BE, HT, TC or TE events occur. Also clearing any of the BE, HT, TC or TE event interrupt flags will clear the GE interrupt flag. Note that if a “1” is written into the GEICLRn bit in the PDMAISR0 register to clear the GE interrupt flag, the BE, HT, TC and TE event interrupt flags will also be cleared to 0 together with the GE interrupt status flag.

## PDMA Interrupt Status Register 1 – PDMAISR1

This register is used to indicate the corresponding interrupt status of the PDMA channel 6~7.

Offset: 0x124

Reset value: 0x0000\_0000



Bits	Field	Descriptions
[9], [4]	TEISTA <sub>m</sub>	Channel m Transfer Error Interrupt Status ( $m = 6 \sim 7$ ) 0: No Transfer Error Occurs 1: Transfer Error Occurs This bit is set by hardware and is cleared by writing a “1” into the corresponding interrupt status clear bit in the PDMAISR1 register. A Transfer error will occur when the PDMA accesses a system reserved address space or the PDMA has received a request but the corresponding transfer capacity is equal to zero.
[8], [3]	TCISTA <sub>m</sub>	Channel m Transfer Complete Interrupt Status ( $m = 6 \sim 7$ ) 0: No Transfer Completion occurs 1: Transfer Completion occurs This bit is set by hardware and is cleared by writing a “1” into the corresponding interrupt status clear bit in the PDMAISR1 register. A Transfer Completion event will occur when the PDMA has completed a data transfer task.
[7], [2]	HTISTA <sub>m</sub>	Channel m Half Transfer Interrupt Status ( $m = 6 \sim 7$ ) 0: No Half Transfer Event Occurs 1: Half Transfer Event Occurs This bit is set by hardware and is cleared by writing a “1” into the corresponding interrupt status clear bit in the PDMAISR1 register. A Half Transfer event will occur when the PDMA has completed half of the data transfer task.
[6], [1]	BEISTA <sub>m</sub>	Channel m Block Transaction End Interrupt Status ( $m = 6 \sim 7$ ) 0: No Block Transaction End Event Occurs 1: Block Transaction End Event Occurs This bit is set by hardware and is cleared by writing a “1” into the corresponding interrupt status clear bit in the PDMAISR1 register. A Block Transaction End event will occur when the PDMA has completed a data block transaction task.

Bits	Field	Descriptions
[5], [0]	GEISTAm	<p>Channel m Global Transfer Event Interrupt Status (m = 6 ~ 7)</p> <p>0: No TE, TC, HT or BE Event Occurs</p> <p>1: TE, TC, HT, or BE Event Occurs</p> <p>This bit is set by hardware and is cleared by writing a “1” into the corresponding interrupt status clear bit, GEICLRm, in the PDMAISR1 register. A Global Transfer Event will occur if any of the BE, HT, TC or TE events occur. Also clearing any of the BE, HT, TC or TE event interrupt flags will also clear the GE interrupt flag. Note that if a “1” is written into the GEICLRm bit in the PDMAISR1 register to clear the GE interrupt flag, the BE, HT, TC and TE event interrupt flags will also be cleared to 0 together with the GE interrupt status flag.</p>

## PDMA Interrupt Status Clear Register 0 – PDMAISCR0

This register is used to clear the corresponding interrupt status bits in the PDMAISR0 Register.

Offset: 0x128

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset		Reserved	TEICLR5	TCICLR5	HTICLR5	BEICLR5	GEICLR5	TEICLR4
	RW	0	RW	0	RW	0	RW	0
Type/Reset	23	22	21	20	19	18	17	16
	TCICLR4	HTICLR4	BEICLR4	GEICLR4	TEICLR3	TCICLR3	HTICLR3	BEICLR3
	RW	0	RW	0	RW	0	RW	0
Type/Reset	15	14	13	12	11	10	9	8
	GEICLR3	TEICLR2	TCICLR2	HTICLR2	BEICLR2	GEICLR2	TEICLR1	TCICLR1
	RW	0	RW	0	RW	0	RW	0
Type/Reset	7	6	5	4	3	2	1	0
	HTICLR1	BEICLR1	GEICLR1	TEICLR0	TCICLR0	HTICLR0	BEICLR0	GEICLR0
	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[29], [24], [19], [14], [9], [4]	TEICLRn	Channel n Transfer Error Interrupt Status Clear (n = 0 ~ 5) 0: No Operation 1: Clear the corresponding TEISTAn bit in the PDMAISR0 register Writing a “1” into the TEICLRn bit will clear the TEISTAn status bit in the PDMAISR0 register. This bit will be automatically cleared to 0 after a “1” is written.
[28], [23], [18], [13], [8], [3]	TCICLRn	Channel n Transfer Complete Interrupt Status Clear (n = 0 ~ 5) 0: No Operation 1: Clear the corresponding TCISTAn bit in the PDMAISR0 register Writing a “1” into the TCICLRn bit will clear the TCISTAn status bit in the PDMAISR0 register. This bit will be automatically cleared to 0 after a “1” is written.
[27], [22], [17], [12], [7], [2]	HTRICLRn	Channel n Half Transfer Interrupt Status Clear (n = 0 ~ 5) 0: No Operation 1: Clear the corresponding HTISTAn bit in the PDMAISR0 register Writing a “1” into the HTRICLRn bit will clear the HTISTAn status bit in the PDMAISR0 register. This bit will be automatically cleared to 0 after a “1” is written.
[26], [21], [16], [11], [6], [1]	BEICLRn	Channel n Block Transaction End Interrupt Status Clear (n = 0 ~ 5) 0: No Operation 1: Clear the corresponding BEISTAn bit in the PDMAISR0 register Writing a “1” into the BEICLRn bit will clear the BEISTAn status bit in the PDMAISR0 register. This bit will automatically cleared to 0 after a data “1” is written.
[25], [20], [15], [10], [5], [0]	GEICLRn	Channel n Global Transfer Event Interrupt Status Clear (n = 0 ~ 5) 0: No Operation 1: Clear the corresponding TEISTAn, TCISTAn, HTISTAn, BEISTAn, and GEISTAn bits in the PDMAISR0 register Writing a “1” into the GEICLRn bit will clear the GEISTAn status bit together with the TEISTAn, TCISTAn, HTISTAn, BEISTAn bits in the PDMAISR0 register. This bit will be automatically cleared to 0 after a “1” is written.

## PDMA Interrupt Status Clear Register 1 – PDMAISCR1

This register is used to clear the corresponding interrupt status bits in the PDMAISR1 Register.

Offset: 0x12C

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Reserved								
Type/Reset	23	22	21	20	19	18	17	16
Reserved								
Type/Reset	15	14	13	12	11	10	9	8
Reserved							TEICLR7	TCICLR7
Type/Reset	RW	0	RW	0	RW	0	RW	0
Type/Reset	7	6	5	4	3	2	1	0
	HTICLR7	BEICLR7	GEICLR7	TEICLR6	TCICLR6	HTICLR6	BEICLR6	GEICLR6
Type/Reset	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[9], [4]	TEICLRm	Channel m Transfer Error Interrupt Status Clear (m = 6 ~ 7) 0: No Operation 1: Clear the corresponding TEISTAm bit in the PDMAISR1 register Writing a “1” into the TEICLRm bit will clear the TEISTAm status bit in the PDMAISR1 register. This bit will be automatically cleared to 0 after a “1” is written.
[8], [3]	TCICLRm	Channel m Transfer Complete Interrupt Status Clear (m = 6 ~ 7) 0: No Operation 1: Clear the corresponding TCISTAm bit in the PDMAISR1 register Writing a “1” into the TCICLRm bit will clear the TCISTAm status bit in the PDMAISR1 register. This bit will be automatically cleared to 0 after a “1” is written.
[7], [2]	HTICLRm	Channel m Half Transfer Interrupt Status Clear (m = 6 ~ 7) 0: No Operation 1: Clear the corresponding HTISTAm bit in the PDMAISR1 register Writing a “1” into the HTICLRm bit will clear the HTISTAm status bit in the PDMAISR1 register. This bit will be automatically cleared to 0 after a “1” is written.
[6], [1]	BEICLRm	Channel m Block Transaction End Interrupt Status Clear (m = 6 ~ 7) 0: No Operation 1: Clear the corresponding BEISTAm bit in the PDMAISR1 register Writing a “1” into the BEICLRm bit will clear the BEISTAm status bit in the PDMAISR1 register. This bit will be automatically cleared to 0 after a “1” is written.
[5], [0]	GEICLRm	Channel m Global Transfer Event Interrupt Status Clear (m = 6 ~ 7) 0: No Operation 1: Clear the corresponding TEISTAm, TCISTAm, HTISTAm, BEISTAm, and GEISTAm bits in the PDMAISR1 register Writing a “1” into the GEICLRm bit will clear the GEISTAm status bit together with the TEISTAm, TCISTAm, HTISTAm, BEISTAm bits in the PDMAISR1 register. This bit will be automatically cleared to 0 after a “1” is written.

## PDMA Interrupt Enable Register 0 – PDMAIER0

This register is used to enable or disable the related interrupts of the PDMA channel 0~5.

Offset: 0x130

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved		TEIE5	TCIE5	HTIE5	BEIE5	GEIE5	TEIE4
	RW	0	RW	0	RW	0	RW	0
Type/Reset	23	22	21	20	19	18	17	16
	TCIE4	HTIE4	BEIE4	GEIE4	TEIE3	TCIE3	HTIE3	BEIE3
Type/Reset	RW	0	RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8
Type/Reset	GEIE3	TEIE2	TCIE2	HTIE2	BEIE2	GEIE2	TEIE1	TCIE1
	RW	0	RW	0	RW	0	RW	0
Type/Reset	7	6	5	4	3	2	1	0
	HTIE1	BEIE1	GEIE1	TEIE0	TCIE0	HTIE0	BEIE0	GEIE0
Type/Reset	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[29], [24], [19], [14], [9], [4]	TEIEn	Channel n Transfer Error Interrupt Enable control (n = 0 ~ 5) 0: Transfer Error interrupt is disabled 1: Transfer Error interrupt is enabled This bit is set and cleared by software.
[28], [23], [18], [13], [8], [3]	TCIEn	Channel n Transfer Complete Interrupt Enable control (n = 0 ~ 5) 0: Transfer Completion interrupt is disabled 1: Transfer Completion interrupt is enabled This bit is set and cleared by software.
[27], [22], [17], [12], [7], [2]	HTIEn	Channel n Half Transfer Interrupt Enable control (n = 0 ~ 5) 0: Half Transfer interrupt is disabled 1: Half Transfer interrupt is enabled This bit is set and cleared by software.
[26], [21], [16], [11], [6], [1]	BEIEn	Channel n Block Transaction End Interrupt Enable control (n = 0 ~ 5) 0: Block Transaction End interrupt is disabled 1: Block Transaction End interrupt is enabled This bit is set and cleared by software.
[25], [20], [15], [10], [5], [0]	GEIEn	Channel n Global Transfer Event Interrupt Enable control (n = 0 ~ 5) 0: Global Transfer Event interrupt is disabled 1: Global Transfer Event interrupt is enabled This bit is set and cleared by software.

## PDMA Interrupt Enable Register 1 – PDMAIER1

This register is used to enable or disable the related interrupts of the PDMA channel 6~7.

Offset: 0x134

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							TEIE7	TCIE7
Type/Reset	RW 0 RW 0							
Type/Reset	7	6	5	4	3	2	1	0
HTIE7 BEIE7 GIE7 TEIE6 TCIE6 HTIE6 BEIE6 GEIE6								
Type/Reset	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[9], [4]	TEIEm	Channel m Transfer Error Interrupt Enable (m = 6 ~ 7) 0: Transfer Error interrupt is disabled 1: Transfer Error interrupt is enabled This bit is set and cleared by software.
[8], [3]	TCIEm	Channel m Transfer Complete Interrupt Enable (m = 6 ~ 7) 0: Transfer Completion interrupt is disabled 1: Transfer Completion interrupt is enabled This bit is set and cleared by software.
[7], [2]	HTIEm	Channel m Half Transfer Interrupt Enable (m = 6 ~ 7) 0: Half Transfer interrupt is disabled 1: Half Transfer interrupt is enabled This bit is set and cleared by software.
[6], [1]	BEIEm	Channel m Block Transaction End Interrupt Enable (m = 6 ~ 7) 0: Block Transaction End interrupt is disabled 1: Block Transaction End interrupt is enabled This bit is set and cleared by software.
[5], [0]	GEIEm	Channel m Global Transfer Event Interrupt Enable (m = 6 ~ 7) 0: Global Transfer Event interrupt is disabled 1: Global Transfer Event interrupt is enabled This bit is set and cleared by software.

# 26 Extend Bus Interface (EBI)

## Introduction

The external bus interface is able to access external parallel interface devices such as SRAM, Flash and LCD modules. The interface is memory mapped into the internal address bus of the Cortex-M3. The data and address lines can be multiplexed in order to reduce the number of pins required to connect to external devices. The bus read/write timing can be adjusted to meet the timing specifications of the external devices. Note that the interface only supports asynchronous 8 or 16-bit bus interfaces.

## Features

- Programmable interface for various memory types
  - Asynchronous static random access memory – SRAM
  - Read-only memory – ROM
  - NOR Flash memory
  - 8-bit or 16-bit parallel bus CPU interface device
- Translates AHB transactions into appropriate external device protocol
- 4 memory bank regions and independent chip select control for each memory bank
- Programmable timings to support a wide range of devices
  - Programmable wait states or external asynchronous ready signal control
  - Programmable bus turnaround cycles
  - Programmable output enable and write enable cycles extension for each memory bank
  - Individual active high or low setting of interface control signal for each memory bank
- Supports page read mode
- Automatic translation when AHB transaction width and external memory interface width is different
- Write buffer to decrease stalling of the AHB write burst transactions
- Supports multiplexed and non-multiplexed address and data line configurations
  - Up to 25 address lines
  - Up to 16-bit data bus width

## Functional Descriptions

An overview of the EBI module is shown in Figure 173. The EBI enables internal CPU and other bus matrix master peripherals to access external memories or devices. The EBI automatically translates the internal AHB transactions into the external device protocol. In particular, if the selected external memory is 16 or 8 bits width, then 32-bit wide transactions on the AHB are auto split into consecutive 16 or 8-bit accesses.

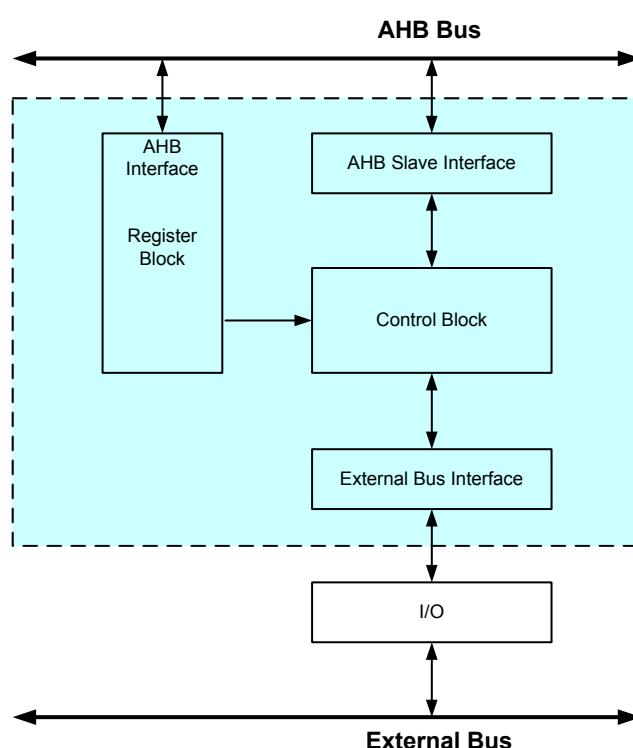


Figure 173. EBI Block Diagram

The EBI supports multiplexed and non-multiplexed addressing modes. The non-multiplexed addressing mode can be operated more efficiently and faster but it requires a higher number of pins. The multiplexed addressing modes are slower and require an external address latch device and a lower number of pins. The functionality of the 16 EBI\_AD pins depends on what kind of the multiplexed addressing mode is used. They are used for both address and data in the multiplexed modes. Also for the non-multiplexed 8-bit address mode, both the address and data fit into these 16 EBI\_AD pins. If more address bits or data bits are needed, an external latch can be used to support up to 24-bit addresses or 16-bit data in the multiplexed addressing modes using only the 16 EBI\_AD pins. Furthermore, independent of the addressing mode, up to 25 non-multiplexed address lines can be enabled on the EBI\_A pin connections. The detailed operation in the supported modes is presented in the following sections. The AHB clock (HCLK) is the reference clock for the EBI.

### Non-multiplexed 8-bit Data 8-bit Address Mode

In this mode, 8-bit address and 8-bit data is supported. The address is located on the higher 8 bits of the EBI\_AD lines and the data uses the lower 8 bits. This mode is set by programming the MODE field in the EBICR register to D8A8. Read and write timing in the 8-bit mode are shown in Figure 174 and Figure 175.

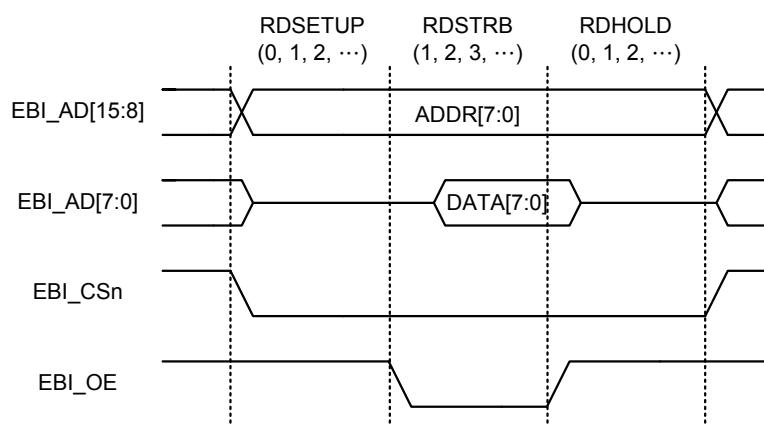


Figure 174. EBI Non-multiplexed 8-bit Data, 8-bit Address Read Operation

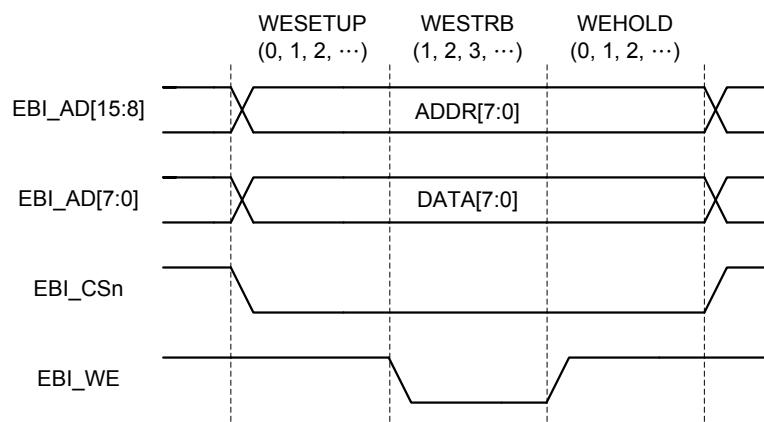


Figure 175. EBI Non-multiplexed 8-bit Data, 8-bit Address Write Operation

### Non-multiplexed 16-bit Data N-bit Address Mode

In this non-multiplexed mode 16-bit data is provided on the 16 EBI\_AD lines. The addresses are provided on the EBI\_A lines. This mode is set by programming the MODE field in the EBICR register to D16. Read and write signals are shown in Figure 176 and Figure 177 for the case in which N address lines on EBI\_A have been enabled.

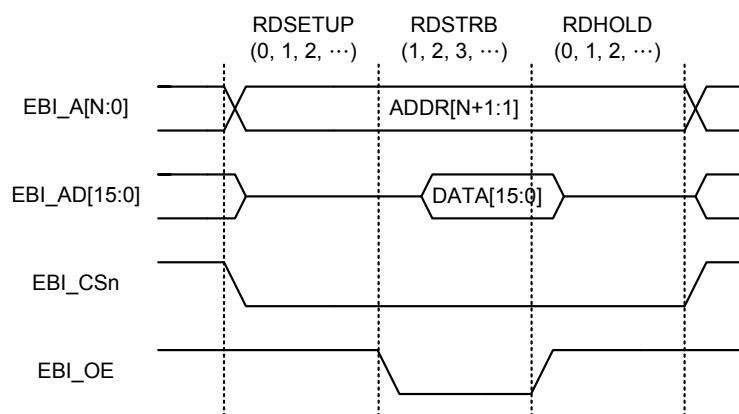


Figure 176. EBI Non-multiplexed 16-bit Data, N-bit Address Read Operation

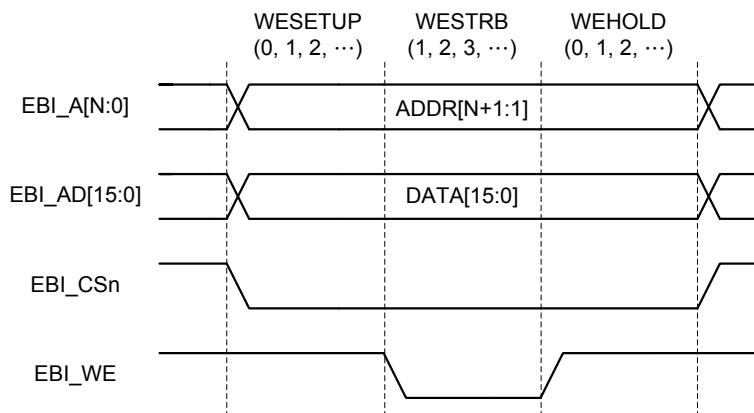


Figure 177. EBI Non-multiplexed 16-bit Data, N-bit Address Write Operation

Since the internal AHB address (HADDR) is a byte (8-bit) address whereas the 16-bit width of external device is addressed in words (16-bit), the address actually issued to the external device varies according to the data width as shown in the following table.

Memory width	Data address issued to the EBI
8-bit	HADDR[N:0] → EBI_A[N:0]
16-bit	HADDR[N+1:1] → EBI_A[N:0]

In case of a 16-bit external device width, the EBI will internally use HADDR[N+1:1] to generate the address EBI\_A[N:0] for external device. Whatever the external memory width (16-bit or 8-bit), EBI\_A[0] should be connected to external device address A[0].

### Multiplexed 16-bit Data, 16-bit Address Mode

In this mode, 16-bit address and 16-bit data is supported, but the utilization of an external latch and an extra signal EBI\_ALE is required. The 16-bit address and 16-bit data bits are multiplexed on the EBI\_AD pins. An EBI address latch setup diagram is shown in Figure 178 . This mode is set by programming the MODE field in the EBICR register to D16A16ALE.

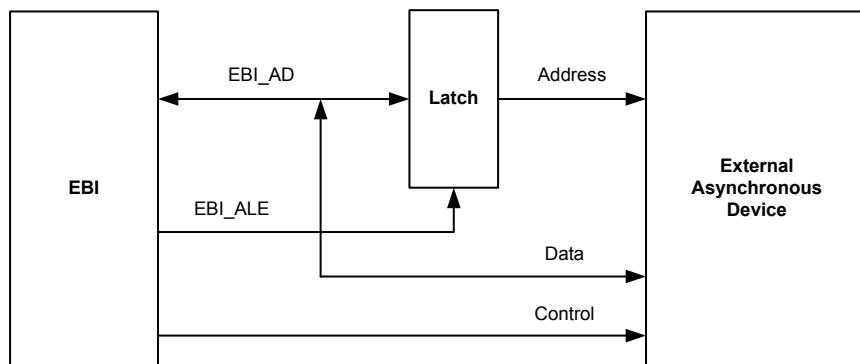


Figure 178. An EBI Address Latch Setup Diagram

At the start of the transaction the address is output on the EBI\_AD lines. The external address latch is controlled by the EBI\_ALE signal and stores the address. Then the data is read or written according to operation. Read and write signals are shown in Figure 179 and Figure 180.

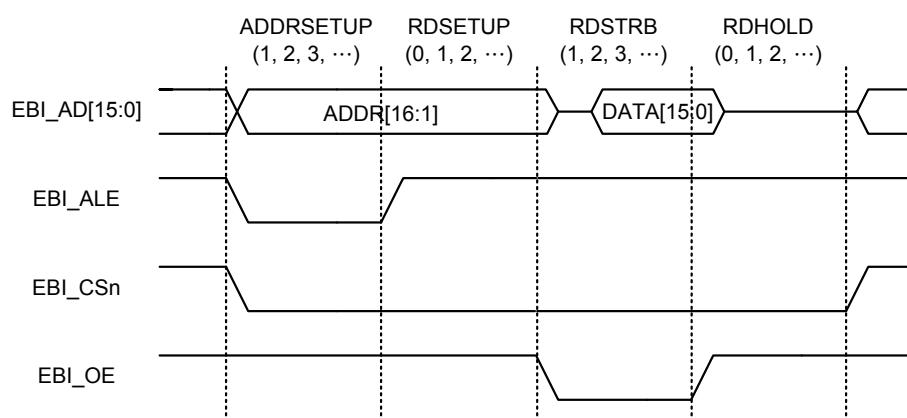
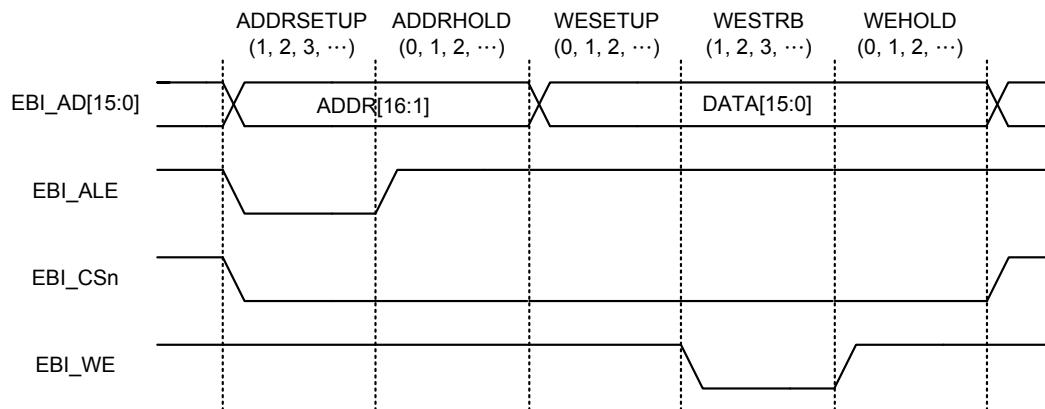


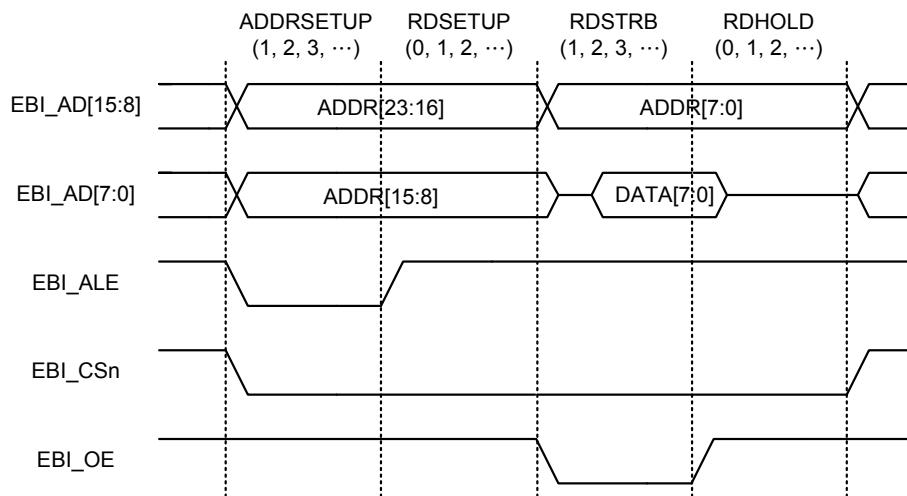
Figure 179. EBI Multiplexed 16-bit Data, 16-bit Address Read Operation



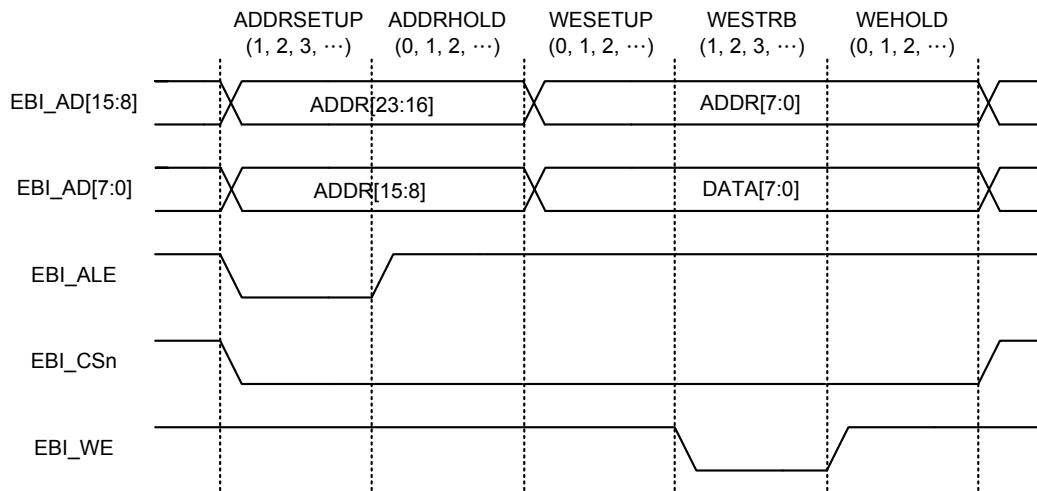
**Figure 180. EBI Multiplexed 16-bit Data, 16-bit Address Write Operation**

### Multiplexed 8-bit Data, 24-bit Address Mode

This mode allows 24-bit address with 8-bit data multiplexed on the EBI\_AD[15:0] lines to reduce the pins utilization and uses the EBI\_ALE signal to decode 8-bit data and 24-bit address. The upper 8 bits of the EBI\_AD lines (EBI\_AD[15:8]) are consecutively used for the highest 8 bits and the lowest 8 bits of the address. The lower 8 bits of the EBI\_AD lines (EBI\_AD[7:0]) are used for the middle 8 address bits and 8-bit data. This mode is set by programming the MODE field in the EBICR register to D8A24ALE. Read and write signals are shown in Figure 181 and Figure 182 respectively.



**Figure 181. EBI Multiplexed 8-bit Data, 24-bit Address Read Operation**

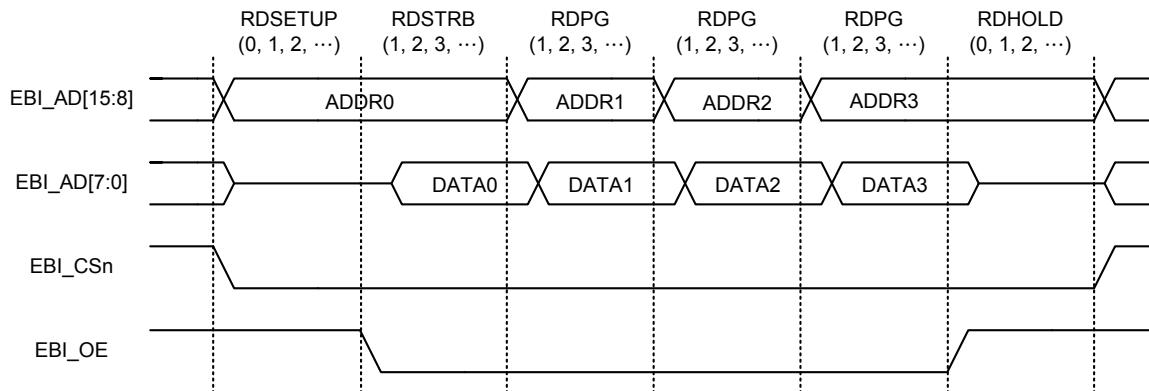


**Figure 182. EBI Multiplexed 8-bit Data, 24-bit Address Write Operation**

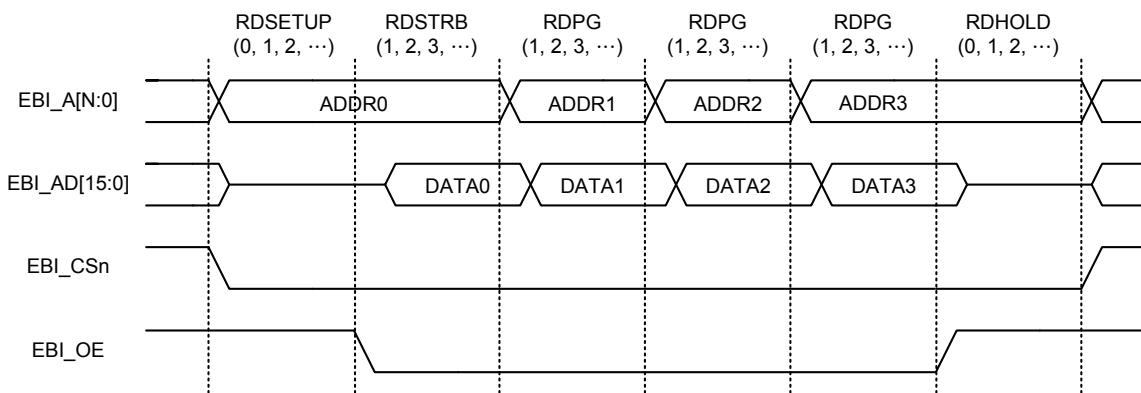
## Page Read Operation

Page mode read operation is a performance-enhancing extension to the legacy asynchronous read transactions. In page-mode-capable devices, an initial asynchronous read access is performed and then adjacent addresses can be read quickly by simply changing the low-order address. For example, Addresses A[3:0] are used to determine the members of the 16-address page mode device. Any change in addresses A[4] or higher will stop the page read and initiate a new asynchronous read access time. Page mode takes advantage of the fact that adjacent addresses can be read faster than random addresses.

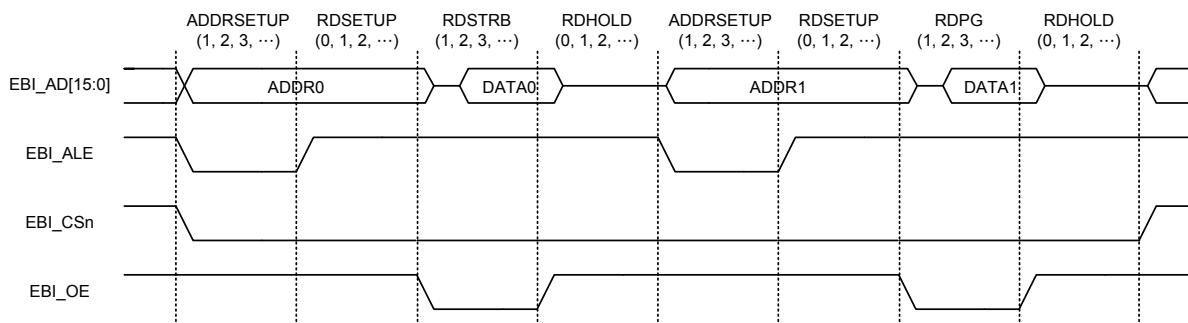
Page mode operation is enabled by setting the PAGEMODE bit in the EBIRTRn register to 1. If enabled, the RDPG field in the EBIPCR register defines the duration of an intrapage access and the PAGELEN field in the EBIPCR register defines the number of addresses members in a page. The INCHIT bit of the EBIPCR register defines whether page hits occur on any addresses member in a page or only on incremental addresses. Page mode reads can be triggered by consecutive reads resulting from wide AHB reads which are automatically translated into multiple narrow external device reads. The following figures show typical page mode read sequences for all addressing modes.



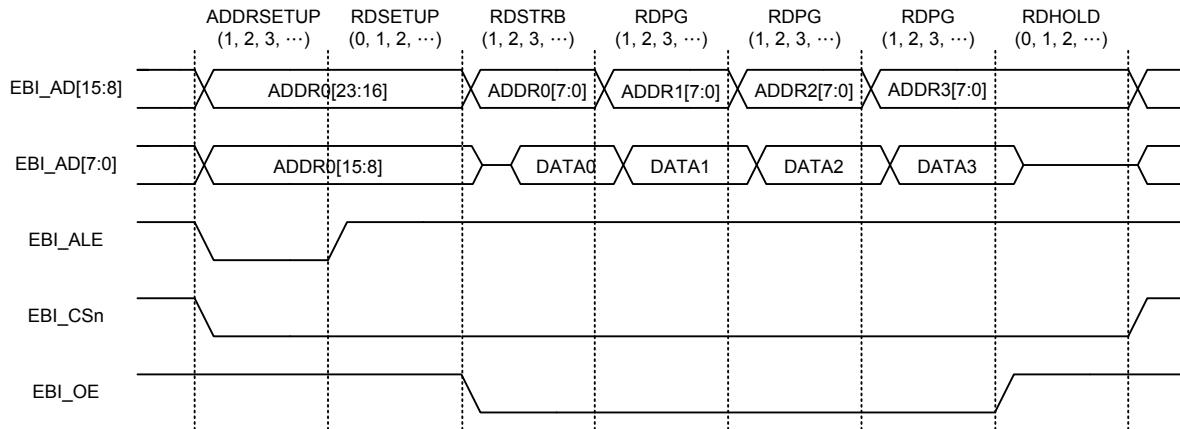
**Figure 183. EBI Non-multiplexed 8-bit Data, 8-bit Address Mode For Page Read Operation**



**Figure 184. EBI Non-multiplexed 16-bit Data, N-bit Address Mode For Page Read Operation**



**Figure 185. EBI Multiplexed 16-bit Data, 16-bit Address Mode For Page Read Operation**

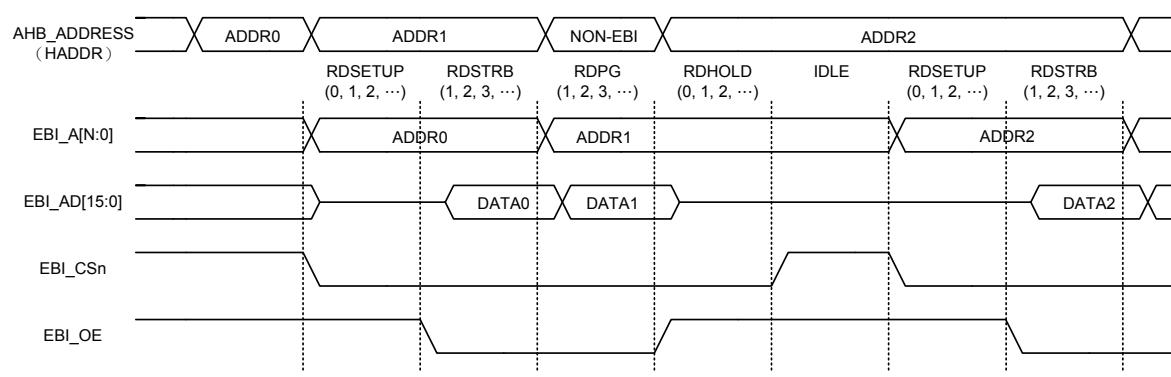


**Figure 186. EBI Multiplexed 8-bit Data, 24-bit Address Mode For Page Read Operation**

The PAGEOPEN field of the EBIPCR defines the maximum duration for which a page read is kept active. New read transactions which hit an open page are started with RDPG timing if the PAGEOPEN time has not been exceeded at the start of such a transaction. Page read transactions are allowed to close the page mode with the following conditions:

- The PAGEOPEN time is exceeded during the continuous read transactions
- The EBI transactions insert a write or a non-intrapage read
- The lack of a new EBI transaction

Figure 187 shows an example in which only ADDR1 benefits from intrapage timing because an unrelated address of AHB transfer is inserted and causes late arrival of ADDR2. The page is considered closed and ADDR2 can therefore not benefit from intrapage timing and it regards it as a normal read access.



Note1: HADDR is AHB bus address input.

**Figure 187. EBI Page Close Example**

## Write Buffer and EBI Status

The EBI has a 32-bit wide write buffer. The write buffer can be used to limit stalling of an AHB write burst transaction which comes from the Cortex-M3 or PDMA to a potentially slow external device.

The EBIBUSY status bit in the EBISR register indicates whether an AHB transaction is still active in the EBI or not. When performing an AHB read or write, the EBIBUSY bit stays 1 until the required transaction(s) with the external device has finished.

## Bus Turn-around and Idle Cycles

The EBI\_AD lines can be driven by either the EBI or the external device depending on the cycle state of EBI bus. The RDHOLD timing parameter is for the bus turn-around time and should be programmed to ensure enough time for the characteristics of an external device. The default setting for the EBI is to insert an IDLE cycle between EBI transactions to the same bank. The IDLE cycle insertion is shown for two back-to-back read transactions in Figure 188. For cases where the IDLE state can also provide the required bus turn-around time, the RDHOLD parameter can be programmed to 0. For increased EBI access performance, the automatic IDLE state insertion can be disabled by setting the NOIDLEn bits in the EBICR register to 1. This example is shown in Figure 189 for two back-to-back reads in a non-multiplexed address mode.

An IDLE cycle will automatically be inserted for the following cases:

- Between two external device transactions to the same bank when the NOIDLEn bit is 0.
- Between two external device transactions to different banks.
- Between a read and a subsequent write on the EBI\_AD lines.
- When no request for an external transaction is available in the EBI.

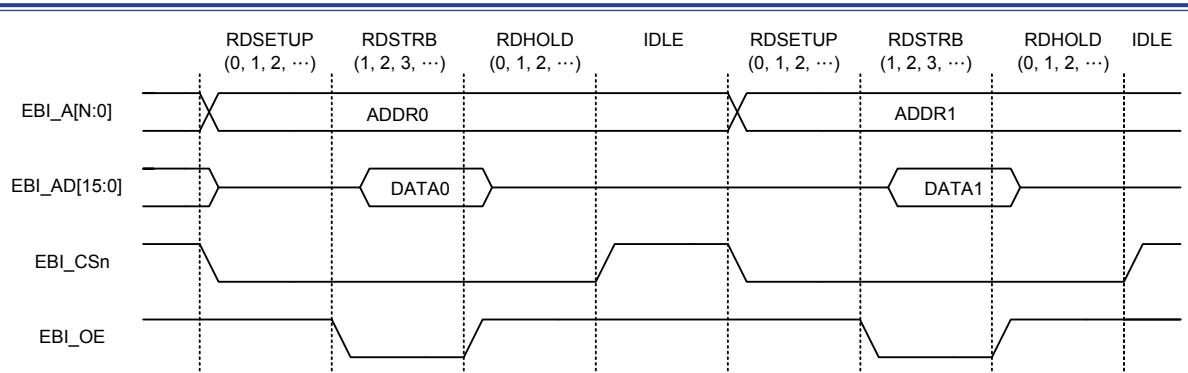


Figure 188. EBI Inserts An IDLE Cycle Between Transactions In The Same Bank (NOIDLE = 0)

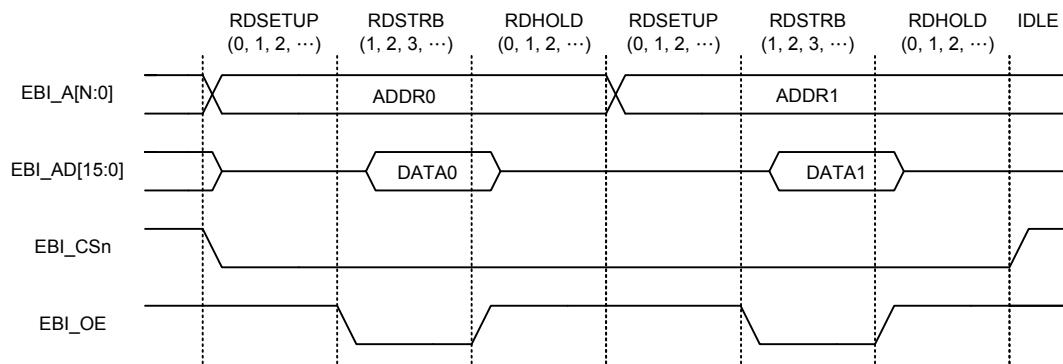


Figure 189. EBI De-asserts An IDLE Cycle Between Transactions In The Same Bank (NOIDLE=1)

### AHB Transaction Width Conversion

The mapping of AHB transactions to an external device depends on the data width of the external device and whether the byte lanes of the external device are supported or not. The Table 1 shows the EBI mapping of AHB transactions to external device transactions. The EBI will automatically translate the different AHB transaction width to external device transactions which matches the external bus capabilities of the device.

- If the AHB master (CPU or PDMA) transaction width is larger than the external bus transaction width. The EBI will split and translate the AHB transaction into consecutive multiple external transactions which have consecutively incrementing the address and start with the least significant data from AHB transaction.
- If the AHB master (CPU or PDMA) transaction width is smaller than the external bus transaction width. The EBI behavior depends on whether the byte lanes are available or not. Reads either use byte lanes to select the required data when it is available, or read according to the full data bus width of the external device and ignore the superfluous data when a byte lane is not available. Writes either uses a byte lane to select the required data when it is available, or EBI automatically perform a read-modify-write sequence when a byte lane is not available.

Table 68. The EBI Maps The AHB Transactions Width To External Device Transactions

AHB Transaction	8-bit External Device Transaction	16-bit External Device Transaction (with byte lanes)	16-bit External Device Transaction (without byte lanes)
8-bit read	1 × 8-bit read	1 × 8-bit read (using byte lane)	1 × 16-bit read (EBI ignore the superfluous data)
16-bit read	2 × 8-bit read	1 × 16-bit read	1 × 16-bit read
32-bit read	4 × 8-bit read	2 × 16-bit read	2 × 16-bit read
8-bit write	1 × 8-bit write	1 × 8-bit write (using byte lane)	1 × 16-bit read; 1 × 16-bit write (EBI read-modify-write)
16-bit write	2 × 8-bit write	1 × 16-bit write	1 × 16-bit write
32-bit write	4 × 8-bit write	2 × 16-bit write	2 × 16-bit write

**Table 69. The EBI Maps The AHB Transactions Width To External Device Transactions Width Using Byte Lane EBI\_BL[1:0]**

External Bus Width	Access from AHB Master		Access to External Bus Interface (EBI)			
	Access type	Address HADDR[1:0] <sup>Note</sup>	Access split	Output value from EBI_A[1:0]	Valid data at EBI_AD[15:0]	Output value from EBI_BL[1:0]
8-bit	Byte (8-bit)	0b00	No split	0b00	EBI_AD[7:0]	0b10
		0b01	No split	0b01		
		0b10	No split	0b10		
		0b11	No split	0b11		
	Half-word (16-bit)	0b00	1/2 access	0b00		
			2/2 access	0b01		
	Half-word (16-bit)	0b10	1/2 access	0b10		
			2/2 access	0b11		
	Word (32-bit)	0b00	1/4 access	0b00		
			2/4 access	0b01		
			3/4 access	0b10		
			4/4 access	0b11		
16-bit	Byte (8-bit)	0b00	No split	0bx0	EBI_AD[7:0]	0b10
		0b01	No split	0bx0	EBI_AD[15:8]	0b01
		0b10	No split	0bx1	EBI_AD[7:0]	0b10
		0b11	No split	0bx1	EBI_AD[15:8]	0b01
	Half-word (16-bit)	0b00	No split	0bx0	EBI_AD[15:0]	0b00
			No split	0bx1	EBI_AD[15:0]	0b00
	Word (32-bit)	0b00	1/2 access	0bx0	EBI_AD[15:0]	0b00
			2/2 access	0bx1	EBI_AD[15:0]	0b00

**NOTES:** 1. HADDR is AHB bus address input.

2. Byte lane polarity is low active in this table.

## EBI Bank Access

The EBI is split into 4 different address regions and each owns an individual EBI\_CS<sub>n</sub> line. When accessing one of the memory regions, the corresponding EBI\_CS<sub>n</sub> line is asserted. This way up to 4 separate devices can share the EBI lines and be identified by the EBI\_CS<sub>n</sub> line. Each bank can individually be enabled or disabled in the EBICR register. And each bank can individually define the external device behavior, including for example data width, timing definitions, page mode operation, and pin polarities. The data space of each bank can be accessed up to 64MB and is shown as Figure 190. The EBI regions address starts at 0x60000000 in the memory map and can also be used for code execution. When running code via EBI regions starting at this address, the Cortex-M3 uses the System bus interface to fetch instructions. This will result in reduced performance because the Cortex-M3 accesses stack, SRAM and peripherals also use the System bus interface.

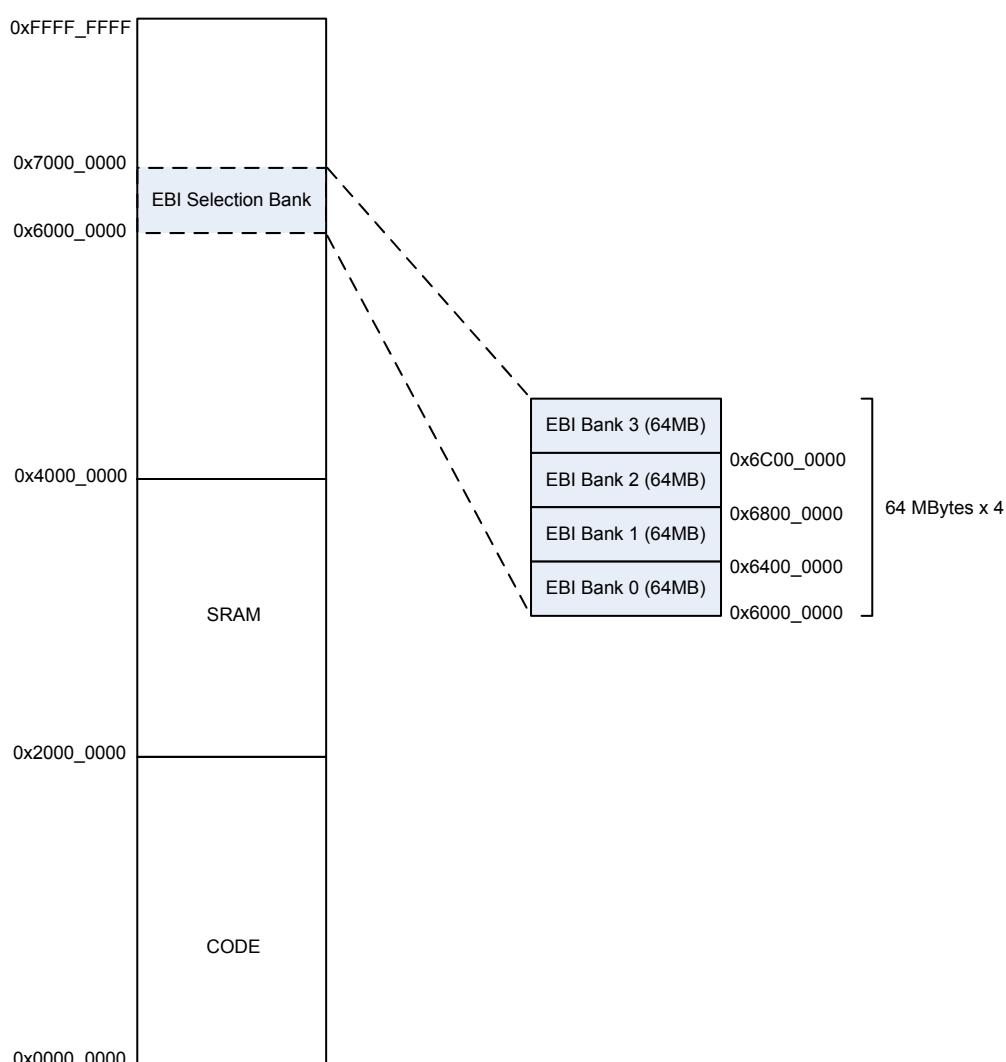


Figure 190. EBI Bank Memory Map

## EBI Ready

Some external devices are able to indicate that they have not finished their write or read operations by asserting the wait signal. The EBI\_ARDY input signal of the EBI is used to extend the read or write cycles for slow external devices when it is enabled by setting the ARDYEN bit in the EBICR register. EBI\_ARDY can be configured by the polarity of this signal with the ARDYPOL bit in the EBIPR register. If the ARDYPOL bit is set to active low, then the read or write cycle is extended while the EBI\_ARDY line is kept high. It also provides a timeout check to prevent a system lock up condition in case where the external device does not de-assert the EBI\_ARDY signal. It will generate a bus asynchronous ready time-out interrupt if EBI\_ARDY is not deasserted within the timeout period. This timeout period has a default value of 32 HLCK clock cycles. Its functionality can be disabled by setting the ARDYTDIS bit in the EBICR register. Note that each memory bank can individually set its wait behavior definition.

## PDMA Request

The EBI only supports using a software trigger for active PDMA service.

## Register Map

The following table shows the EBI register and reset value.

**Table 70. Register map of EBI**

Register	Offset	Description	Reset Value
<b>EBI Base Address = 0x4009_8000</b>			
EBICR	0x000	EBI Control Register	0x0000_0000
EBIPCR	0x004	EBI Page Control Register	0x0000_0F00
EBISR	0x008	EBI Status Register	0x0000_0010
EBIATR0	0x010	EBI Address Timing Register 0	0x0000_0F0F
EBIRTR0	0x014	EBI Read Timing Register 0	0x000F_3F0F
EBIWTR0	0x018	EBI Write Timing Register 0	0x000F_3F0F
EBIPR0	0x01C	EBI Parity Register 0	0x0000_0000
EBIATR1	0x020	EBI Address Timing Register 1	0x0000_0F0F
EBIRTR1	0x024	EBI Read Timing Register 1	0x000F_3F0F
EBIWTR1	0x028	EBI Write Timing Register 1	0x000F_3F0F
EBIPR1	0x02C	EBI Parity Register 1	0x0000_0000
EBIATR2	0x030	EBI Address Timing Register 2	0x0000_0F0F
EBIRTR2	0x034	EBI Read Timing Register 2	0x000F_3F0F
EBIWTR2	0x038	EBI Write Timing Register 2	0x000F_3F0F
EBIPR2	0x03C	EBI Parity Register 2	0x0000_0000
EBIATR3	0x040	EBI Address Timing Register 3	0x0000_0F0F
EBIRTR3	0x044	EBI Read Timing Register 3	0x000F_3F0F
EBIWTR3	0x048	EBI Write Timing Register 3	0x000F_3F0F
EBIPR3	0x04C	EBI Parity Register 3	0x0000_0000
EBIENR	0x050	EBI Interrupt Enable Register	0x0000_0000
EBIIFR	0x054	EBI Interrupt Flag Register	0x0000_0000
EBIIFCR	0x058	EBI Interrupt Clear Register	0x0000_0000

## Register Descriptions

### EBI Control Register – EBICR

This register specifies the control setting for EBI bank.

Offset: 0x000

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset	RW	0	RW	0	RW	0	RW	0
	23	22	21	20	19	18	17	16
Type/Reset	ARDYTDIS3	ARDYEN3	ARDYTDIS2	ARDYEN2	ARDYTDIS1	ARDYEN1	ARDYTDIS0	ARDYEN0
Type/Reset	RW	0	RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8
Type/Reset	NOIDLE3	NOIDLE2	NOIDLE1	NOIDLE0	BANKEN3	BANKEN2	BANKEN1	BANKEN0
Type/Reset	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
Type/Reset	Mode3		Mode2		Mode1		Mode0	
Type/Reset	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[31:28]	IDLET	IDLE Time Sets the number of cycles between EBI transactions. If set to 0, one cycle is inserted by the hardware. The cycle unit is based on the HCLK clock period.
[27]	BLEN3	Byte Lane Enable 3 0: Disable EBI byte lane functionality 1: Enable EBI byte lane functionality Enable or disable byte lane functionality for bank 3.
[26]	BLEN2	Byte Lane Enable 2 0: Disable EBI byte lane functionality 1: Enable EBI byte lane functionality Enable or disable byte lane functionality for bank 2.
[25]	BLEN1	Byte Lane Enable 1 0: Disable EBI byte lane functionality 1: Enable EBI byte lane functionality Enable or disable byte lane functionality for bank 1.
[24]	BLEN0	Byte Lane Enable 0 0: Disable EBI byte lane functionality 1: Enable EBI byte lane functionality Enable or disable byte lane functionality for bank 0.
[23]	ARDYTDIS3	Asynchronous Ready Timeout Disable 3 0: Enable EBI asynchronous ready timeout control functionality 1: Disable EBI asynchronous ready timeout control functionality Enable or disable the asynchronous ready timeout functionality for bank 3. The default asynchronous ready timeout period is 32 HCLK clock cycles and cannot be changed.

Bits	Field	Descriptions
[22]	ARDYEN3	Asynchronous Ready Enable 3 0: Disable EBI asynchronous ready control functionality 1: Enable EBI asynchronous ready control functionality Enable or disable the asynchronous ready functionality for bank 3.
[21]	ARDYTDIS2	Asynchronous Ready Timeout Disable 2 0: Enable EBI asynchronous ready timeout control functionality 1: Disable EBI asynchronous ready timeout control functionality Enable or disable the asynchronous ready timeout functionality for bank 2. The default asynchronous ready timeout period is 32 HCLK clock cycles and cannot be changed.
[20]	ARDYEN2	Asynchronous Ready Enable 2 0: Disable EBI asynchronous ready control functionality 1: Enable EBI asynchronous ready control functionality Enable or disable the asynchronous ready functionality for bank 2.
[19]	ARDYTDIS1	Asynchronous Ready Timeout Disable 1 0: Enable EBI asynchronous ready timeout control functionality 1: Disable EBI asynchronous ready timeout control functionality Enable or disable the asynchronous ready timeout functionality for bank 1. The default asynchronous ready timeout period is 32 HCLK clock cycles and cannot be changed.
[18]	ARDYEN1	Asynchronous Ready Enable 1 0: Disable EBI asynchronous ready control functionality 1: Enable EBI asynchronous ready control functionality Enable or disable the asynchronous ready functionality for bank 1.
[17]	ARDYTDIS0	Asynchronous Ready Timeout Disable 0 0: Enable EBI asynchronous ready timeout control functionality 1: Disable EBI asynchronous ready timeout control functionality Enable or disable the asynchronous ready timeout functionality for bank 0. The default asynchronous ready timeout period is 32 HCLK clock cycles and cannot be changed.
[16]	ARDYEN0	Asynchronous Ready Enable 0 0: Disable EBI asynchronous ready control functionality 1: Enable EBI asynchronous ready control functionality Enable or disable the asynchronous ready functionality for bank 0.
[15]	NOIDLE3	No IDLE 3 0: Enable IDLE state insertion 1: Disable IDLE state insertion Enable or disable the insertion of an idle state between transactions for bank 3.
[14]	NOIDLE2	No IDLE 2 0: Enable IDLE state insertion 1: Disable IDLE state insertion Enable or disable the insertion of an idle state between transactions for bank 2.
[13]	NOIDLE1	No IDLE 1 0: Enable IDLE state insertion 1: Disable IDLE state insertion Enable or disable the insertion of an idle state between transactions for bank 1.
[12]	NOIDLE0	No IDLE 0 0: Enable IDLE state insertion 1: Disable IDLE state insertion Enable or disable the insertion of an idle state between transactions for bank 0.

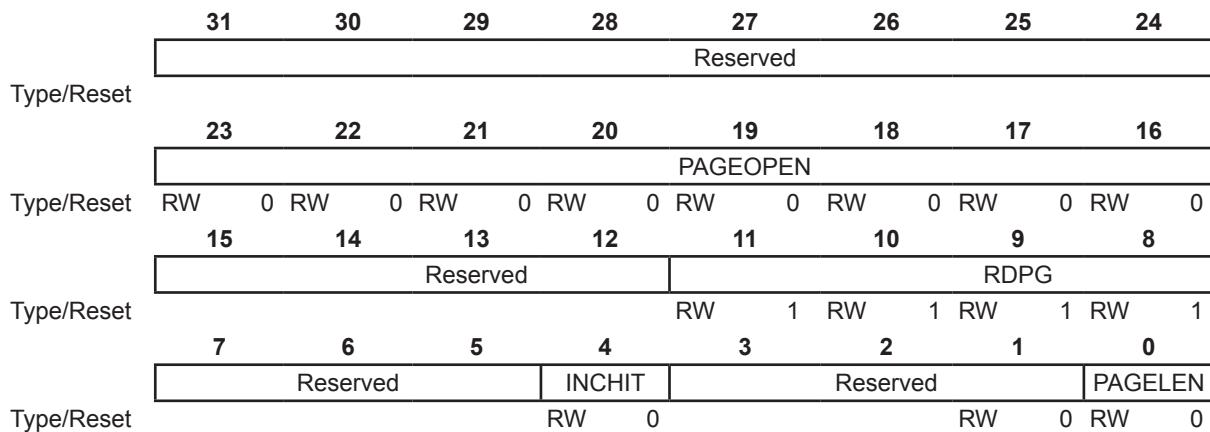
Bits	Field	Descriptions
[11]	BANKEN3	Bank 3 Enable 0: Disable 1: Enable This bit enables or disables bank 3.
[10]	BANKEN2	Bank 2 Enable 0: Disable 1: Enable This bit enables or disables bank 2.
[9]	BANKEN1	Bank 1 Enable 0: Disable 1: Enable This bit enables or disables bank 1.
[8]	BANKEN0	Bank 0 Enable 0: Disable 1: Enable This bit enables or disables bank 0.
[7:6]	MODE3	Set EBI bank 3 access mode 00: D8A8 01: D16A16ALE 10: D8A24ALE 11: D16
[5:4]	MODE2	Set EBI bank 2 access mode 00: D8A8 01: D16A16ALE 10: D8A24ALE 11: D16
[3:2]	MODE1	Set EBI bank 1 access mode 00: D8A8 01: D16A16ALE 10: D8A24ALE 11: D16
[1:0]	MODE0	Set EBI bank 0 access mode 00: D8A8 01: D16A16ALE 10: D8A24ALE 11: D16

## EBI Page Control Register – EBIPCR

This register specifies the EBI page read configuration setting.

Offset: 0x004

Reset value: 0x0000\_0F00



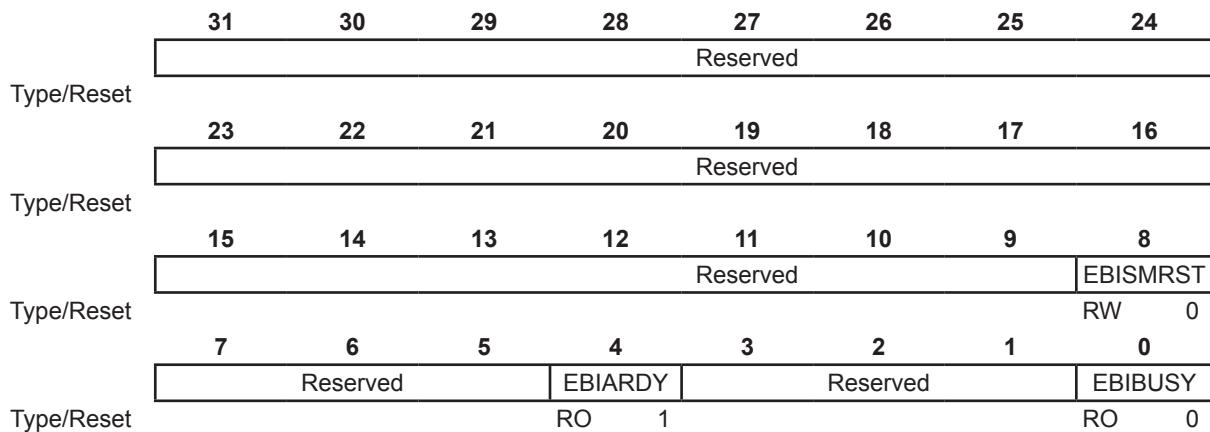
Bits	Field	Descriptions
[23:16]	PAGEOPEN	Maximum Page Open Time Sets the maximum interval of consecutive cycles that a page can be considered open. The cycle unit is basic on an HCLK clock period. Note the PAGEOPEN field should not be set to 0 if the page read is enabled.
[11:8]	RDPG	Page Read Access Time 0000: 1 HCLK clock period 0001: 1 HCLK clock period 0010: 2 HCLK clock periods ..... 1111: 15 HCLK clock periods Sets the number of the cycles for an intrapage page read access time. The cycle unit is basic on an HCLK clock period.
[4]	INCHIT	Incremental Addresses Hit 0: Page hits that occurred on any member in a page 1: Page hits only on incremental addresses Sets the page hits that occurred on any member in a page or only on incremental addresses.
[1:0]	PAGELEN	Page Length 00: 4 members in a page 01: 8 members in a page 10: 16 members in a page 11: 32 members in a page Sets the amount of members in a page.

## EBI Status Register – EBISR

This register indicates the EBI status.

Offset: 0x008

Reset value: 0x0000\_0010



Bits	Field	Descriptions
[8]	EBISMRST	EBI State Machine Reset 0: Normal 1: Reset EBI state machine Write a “1” to reset the EBI internal state machine to its initial state and keep the original the register settings.
[4]	EBIARDY	EBI Asynchronous Ready Status 0: EBI_ARDY is inactive (Device is in busy state) 1: EBI_ARDY is active (Device is in ready state) Indicate the EBI_ARDY line status.
[0]	EBIBUSY	EBI BUSY 0: EBI is idle 1: EBI is busy Indicates the EBI is busy with an AHB transaction.

## EBI Address Timing Register n – EBIATRn, n = 0 ~ 3

This register specifies the address timing setting for bank n. (n = 0 ~ 3)

Offset: 0x010 (n = 0), 0x20 (n = 1), 0x30 (n = 2), 0x40 (n = 3)

Reset value: 0x0000\_0F0F

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

Bits	Field	Descriptions
[11:8]	ADDRHOLD	Address Hold Time Sets the number of cycles the address is held the EBI_AD bus after EBI_ALE is asserted. This field allows setting to 0.
[3:0]	ADDRSETUP	Address Setup Time Sets the number of cycles the address is driven onto the EBI_AD bus before EBI_ALE is asserted. If set to 0, one cycles is inserted by HW. The cycle unit is basic on HCLK clock period.

## EBI Read Timing Register n – EBIRTRn, n = 0 ~ 3

This register specifies the read timing setting for bank n. (n = 0 ~ 3)

Offset: 0x014 (n = 0), 0x24 (n = 1), 0x34 (n = 2), 0x44 (n = 3)

Reset value: 0x000F\_3F0F

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

Bits	Field	Descriptions
[24]	PGEN	Page Enable 0: Page read disable 1: Page read enable This bit is used to enable the page read mode for the corresponding bank.
[19:16]	RDHOLD	Read Hold Time Sets the number of cycles that the EBI_CS <sub>n</sub> is held active for after EBI_OE is de-asserted. This interval is used for bus turnaround.
[13:8]	RDSTRB	Read Strobe Time Sets the number of cycles that the EBI_OE is held active for. After the specified number of cycles, the data is read. If set to 0, one cycle is inserted by the hardware. The cycle unit is based on an HCLK clock period.
[3:0]	RDSETUP	Read Setup Time Sets the number of cycles for the address setup before EBI_OE is asserted. The cycle unit is basic on an HCLK clock period.

### EBI Write Timing Register n – EBIWTRn, n = 0 ~ 3

This register specifies the write timing setting for bank n. (n = 0 ~ 3)

Offset: 0x018 (n = 0), 0x28 (n = 1), 0x38 (n = 2), 0x48 (n = 3)

Reset value: 0x000F\_3F0F

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

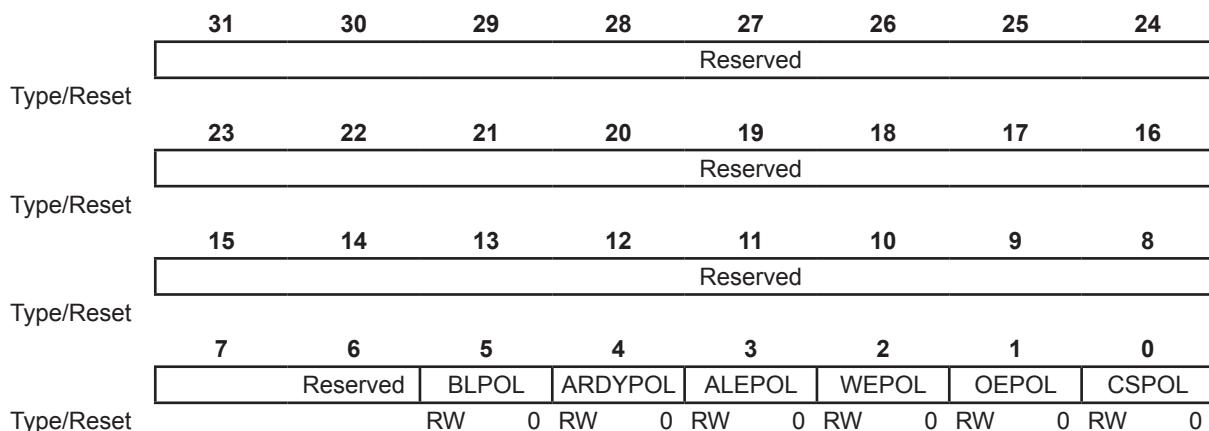
Bits	Field	Descriptions
[19:16]	WEHOLD	Write Hold Time Sets the number of cycles that EBI_CS <sub>n</sub> is held active for after EBI_WE is de-asserted.
[13:8]	WESTRB	Write Strobe Time Sets the number of cycles that EBI_WE is held active for. If set to 0, one cycle is inserted by the hardware. The cycle unit is basic on an HCLK clock period.
[3:0]	WESETUP	Write Setup Time Sets the number of cycles for the address setup before EBI_WE is asserted. The cycle unit is basic on an HCLK clock period.

## EBI Parity Register n – EBIPR, n = 0 ~ 3

This register specifies the polarity of the EBI control signal for bank n. (n = 0 ~ 3)

Offset: 0x01C (n = 0), 0x2C (n = 1), 0x3C (n = 2), 0x4C (n = 3)

Reset value: 0x0000\_0000



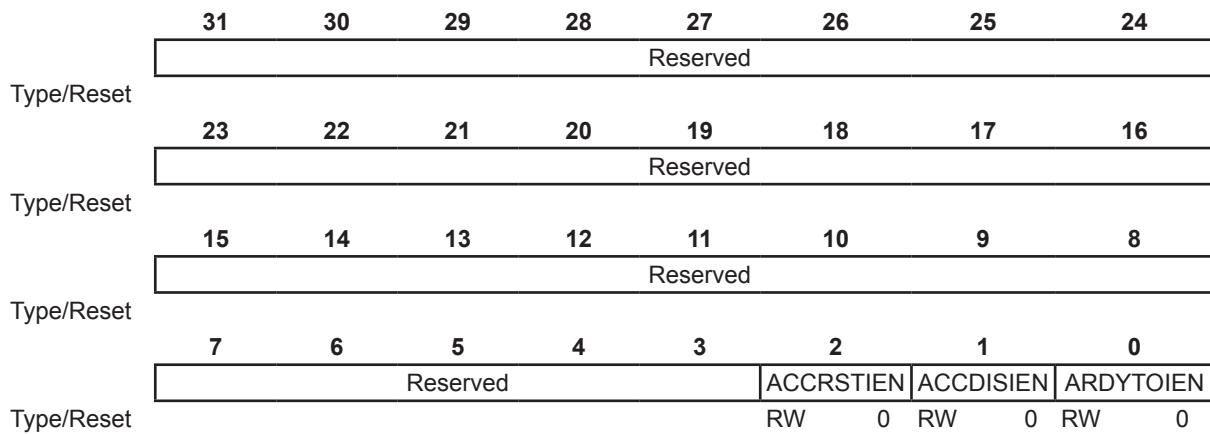
Bits	Field	Descriptions
[5]	BLPOL	Byte Lane Polarity 0: EBI_BL is active low 1: EBI_BL is active high Set the polarity of the EBI_BL line
[4]	ARDYPOL	Asynchronous Ready Polarity 0: EBI_ARDY is active low 1: EBI_ARDY is active high Set the polarity of the EBI_ARDY line
[3]	ALEPOL	Address Latch Polarity 0: EBI_ALE is active low 1: EBI_ALE is active high Set the polarity of the EBI_ALE line
[2]	WEPOL	Write Enable Polarity 0: EBI_WE is active low 1: EBI_WE is active high Set the polarity of the EBI_WE line.
[1]	OEPOL	Output Enable Polarity 0: EBI_OE is active low 1: EBI_OE is active high Set the polarity of the EBI_OE line.
[0]	CSPOL	Chip Selection Polarity 0: EBI_CS is active low 1: EBI_CS is active high Set the polarity of the EBI_CS line.

## EBI Interrupt Enable Register – EBIIENR

This register specifies the EBI interrupt enable.

Offset: 0x050

Reset value: 0x0000\_0000



Bits	Field	Descriptions
[2]	ACCRSTIEN	Enable interrupt for issuing a transaction under an EBI state machine reset (EBISMRST bit is set as “1” in the EBISR register).
[1]	ACCDISIEN	Enable interrupt to access the disabled bank.
[0]	ARDYTOIEN	Enable interrupt for the EBI asynchronous ready time-out.

## EBI Interrupt Flag Register – EBIIFR

This register specifies interrupt indication for EBI.

Offset: 0x054

Reset value: 0x0000\_0000

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

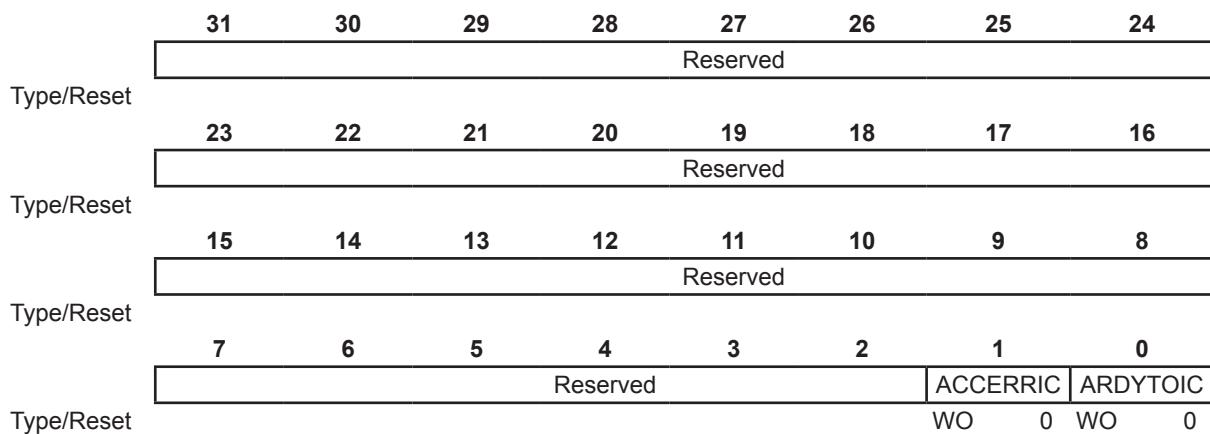
Bits	Field	Descriptions
[1]	ACCERRIF	Set “1” while EBI is accessing the disabled bank or under a EBI state machine reset.
[0]	ARDYTOIF	Set “1” when the EBI asynchronous ready time-out counter reaches a set value.

## EBI Interrupt Clear Register – EBIIFCR

This register specifies interrupt clear for the EBI interrupt.

Offset: 0x058

Reset value: 0x0000\_0000



Bits	Field	Descriptions
[1]	ACCERRIC	Write a “1” to clear the interrupt flag of the accessed disabled bank or under an EBI state machine reset. If software intends to read this bit then only a “0” is returned.
[0]	ARDYTOIC	Write a “1” to clear the interrupt flag of the EBI asynchronous ready time-out interrupt. If software intends to read this bit then only a “0” is returned.

# 27 Inter-IC Sound (I<sup>2</sup>S)

## Introduction

The I<sup>2</sup>S is a synchronous communication interface that can be used as a master or slave to exchange data with other audio peripherals, such as ADCs or DACs. The I<sup>2</sup>S supports a variety of data formats. In addition to the stereo I<sup>2</sup>S-justified, Left-justified and Right-justified modes, there are mono PCM modes with 8/16/24/32-bit sample size. When the I<sup>2</sup>S operates in the master mode, then when using the fractional divider, it can provide an accurate sampling frequency output and support the rate control function and fine-tuning of the output frequency to avoid system problems caused by the cumulative frequency error between different devices.

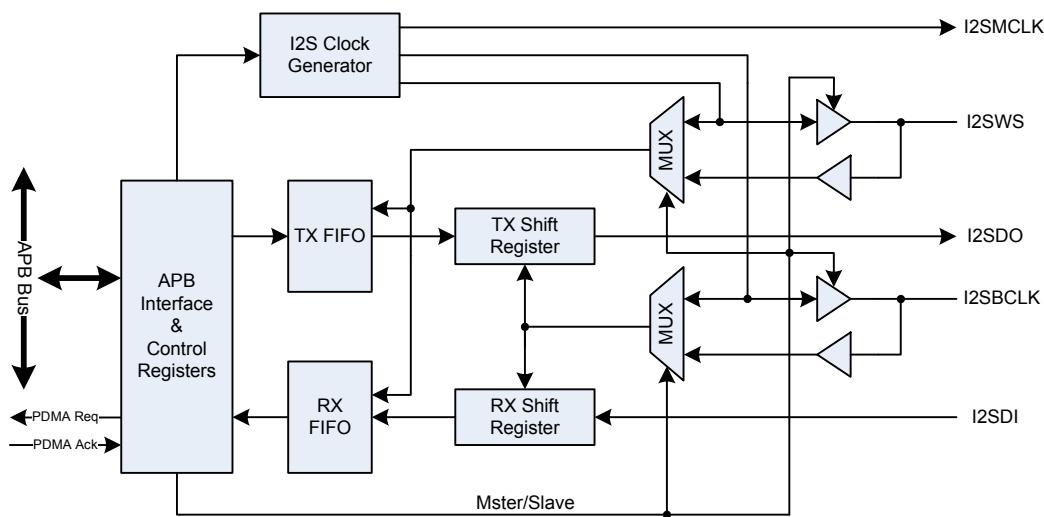


Figure 191. I<sup>2</sup>S Block Diagram

## Features

- Master or slave mode
- Mono and stereo
- I<sup>2</sup>S-justified, Left-justified, and Right-justified mode
- 8/16/24/32-bit sample size with 32-bit channel extended
- 8 × 32-bit Tx & Rx FIFO with PDMA supported
- 8-bit Fractional Clock Divider with rate control

## Functional Description

### I<sup>2</sup>S Master and Slave Mode

The I<sup>2</sup>S can operate in slave or master mode. Within the I<sup>2</sup>S module the difference between these modes lies in the word select (WS) signal which determines the timing of data transmissions.

- In the master mode, the word select signal is generated internally by a clock rate generator.
- In the slave mode, the word select signal is input on the I<sup>2</sup>S\_WS pin.
- When an I<sup>2</sup>S bus is enabled, the word select, bit clock signals are sent continuously by the bus master.
- The mute control bit will place the transmit channel in a mute condition. When the mute mode is enabled, the transmit channel FIFO operates normally, but the output data stream is discarded and replaced by zeroes. This bit does not affect the receive channel so data reception can occur normally.

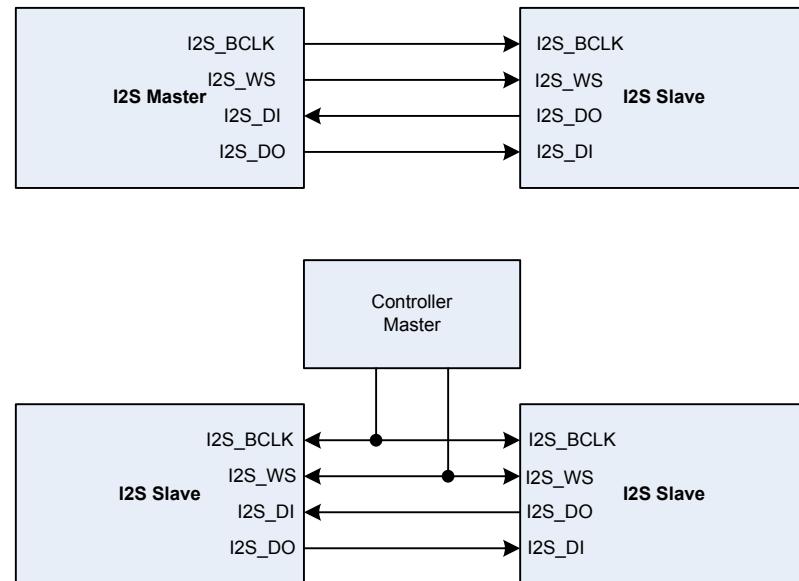


Figure 192. Simple I<sup>2</sup>S Master/Slave Configuration

## I<sup>2</sup>S Clock Rate Generator

The main (I<sup>2</sup>S\_MCLK) and bit clock (I<sup>2</sup>S\_BCLK) rates for the I<sup>2</sup>S are determined by the values in the I2SCDR register. The required I<sup>2</sup>S bit clock rate setting depends on the desired audio sample rate desired, the format (stereo/mono) used, and the data size. The main clock rate (I<sup>2</sup>S\_MCLK) is generated using a fractional rate divider which is a divided down PCLK frequency of the I<sup>2</sup>S. Values of the numerator (X) and the denominator (Y) must be chosen to produce a frequency twice that of the main clock (I<sup>2</sup>S\_MCLK). The output frequency of the divider is divided by 2 in order to get the duty cycle of the output clock more even. The I<sup>2</sup>S clock generator block diagram is shown in Figure 193. The equation for the fractional rate divider is:

$$\begin{aligned} \text{I}^2\text{S\_MCLK} &= 1/2 * \text{PCLK} * (X/Y), \text{ and } 2X \leq Y, X \neq 0, Y \neq 0 \\ \text{I}^2\text{S\_BCLK} &= \text{I}^2\text{S\_MCLK} / N, N = \{1, 2, \dots, 64\} \end{aligned}$$

Because the fractional rate divider is a fully digital implementation function, the divider output clock transitions are synchronous with the input source clock. Therefore, the fractional rate divider will generate some jitter with some divider settings. Users should make note of this phenomenon when choosing the X and Y setup values. It is possible to avoid jitter entirely by choosing fractions such that X divides evenly into Y. For example, 2/4, 2/6, 3/9, 1/N, etc.

The tables below show the recommended setup values to reduce clock jitter for different source clocks and sample rates.

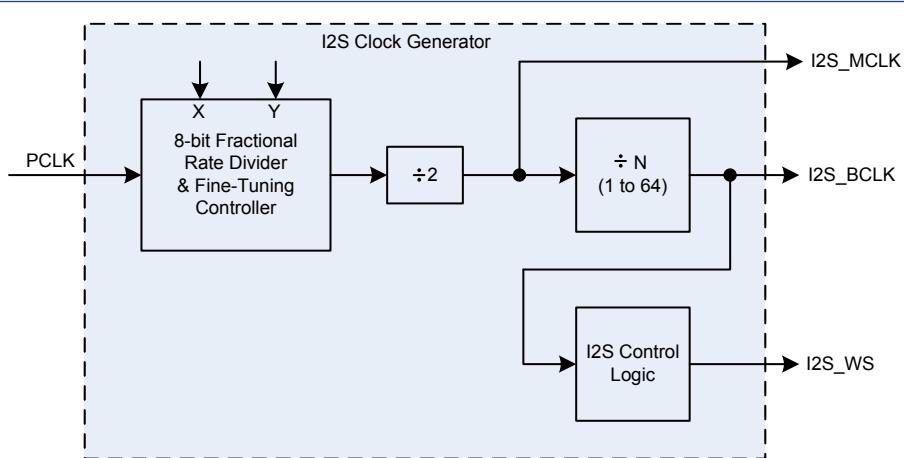


Figure 193. I<sup>2</sup>S Clock Generator Diagram

**Table 71. Recommend FS List @ 8 MHz PCLK**

<b>Fs (Hz)</b>	512 FS		384 FS		256 FS		192 FS		128 FS		64 FS		32 FS		16 FS	
	X	Y	X	Y	X	Y	X	Y	X	Y	X	Y	X	Y	X	Y
8,000	—	—	48	125	32	125	24	125	16	125	8	125	4	125	2	125
11,025	—	—	—	—	85	241	59	223	45	255	21	238	9	204	3	136
12,000	—	—	—	—	48	125	36	125	24	125	12	125	6	125	3	125
16,000	—	—	—	—	—	—	48	125	32	125	16	125	8	125	4	125
22,050	—	—	—	—	—	—	—	—	85	241	45	255	21	238	9	204
24,000	—	—	—	—	—	—	—	—	48	125	24	125	12	125	6	125
32,000	—	—	—	—	—	—	—	—	—	—	32	125	16	125	8	125
44,100	—	—	—	—	—	—	—	—	—	—	85	241	45	255	21	238
48,000	—	—	—	—	—	—	—	—	—	—	48	125	24	125	12	125
96,000	—	—	—	—	—	—	—	—	—	—	—	—	48	125	24	125
192,000	—	—	—	—	—	—	—	—	—	—	—	—	—	—	48	125

**Table 72. Recommend FS List @ 48 MHz PCLK**

<b>Fs (Hz)</b>	512 FS		384 FS		256 FS		192 FS		128 FS		64 FS		32 FS		16 FS	
	X	Y	X	Y	X	Y	X	Y	X	Y	X	Y	X	Y	X	Y
8,000	18	211	8	125	9	211	4	125	5	234	1	94	1	188	-	-
11,025	2	17	3	34	1	17	3	68	1	34	1	68	1	136	-	-
12,000	16	125	12	125	8	125	6	125	4	125	2	125	1	125	1	250
16,000	43	252	16	125	18	211	8	125	9	211	5	234	1	94	1	188
22,050	4	17	3	17	2	17	3	34	1	17	1	34	1	68	1	136
24,000	32	125	24	125	16	125	12	125	8	125	4	125	2	125	1	125
32,000	71	208	32	125	43	252	16	125	18	211	9	211	5	234	1	94
44,100	119	253	85	241	4	17	3	17	2	17	1	17	1	34	1	68
48,000	—	—	48	125	32	125	24	125	16	125	8	125	4	125	2	125
96,000	—	—	—	—	—	—	—	—	—	—	32	125	16	125	8	125
192,000	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

**Table 73. Recommend FS List @ 72 MHz PCLK**

<b>Fs (Hz)</b>	512 FS		384 FS		256 FS		192 FS		128 FS		64 FS		32 FS		16 FS	
	X	Y	X	Y	X	Y	X	Y	X	Y	X	Y	X	Y	X	Y
8,000	12	211	9	211	6	211	5	234	3	211	1	141	—	—	—	—
11,025	4	51	1	17	2	51	1	34	1	51	1	102	1	204	—	—
12,000	18	211	8	125	9	211	4	125	5	234	1	94	1	188	—	—
16,000	19	167	18	211	12	211	9	211	6	211	3	211	1	141	—	—
22,050	37	236	2	17	4	51	1	17	2	51	1	51	1	102	1	204
24,000	43	252	16	125	18	211	8	125	9	211	5	234	1	94	1	188
32,000	38	167	43	252	19	167	18	211	12	211	6	211	3	211	1	141
44,100	53	169	4	17	37	236	2	17	4	51	2	51	1	51	1	102
48,000	71	208	32	125	43	252	16	125	18	211	9	211	5	234	1	94
96,000	—	—	—	—	71	208	32	125	43	252	18	211	9	211	5	234
192,000	—	—	—	—	—	—	—	71	208	43	252	18	211	9	211	—

## I<sup>2</sup>S Interface Format

### I<sup>2</sup>S-justified Stereo Mode

The standard I<sup>2</sup>S-justified mode is where the Most Significant Bit (MSB) of the stereo audio sample data is available on the second rising edge of the BCLK clock following a WS signal transition. In the stereo mode, a low WS state indicates left channel data and a high state indicates right channel data. Figure 194 and Figure 195 show the standard I<sup>2</sup>S-justified stereo mode format.

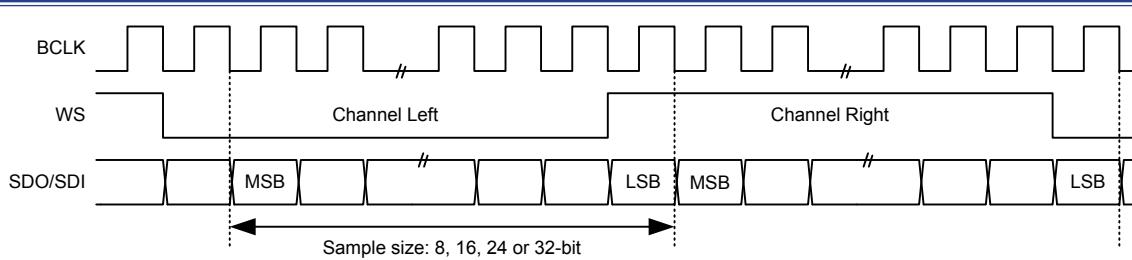


Figure 194. I<sup>2</sup>S-justified Stereo Mode Waveforms

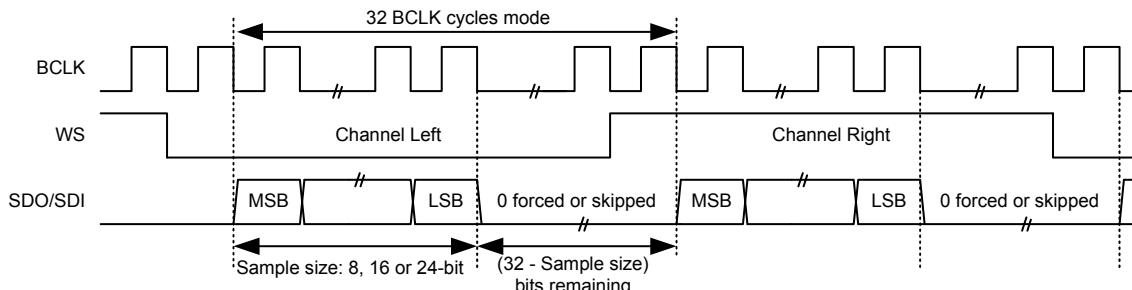


Figure 195. I<sup>2</sup>S-justified Stereo Mode Waveforms (32-bit Channel Enabled)

### Left-justified Stereo Mode

Left-Justified mode is where the Most Significant Bit (MSB) of the stereo audio sample data is available on the first rising edge of BCLK following a WS transition. Figure 196 and Figure 197 are shown with a left I<sup>2</sup>S-justified stereo mode format.

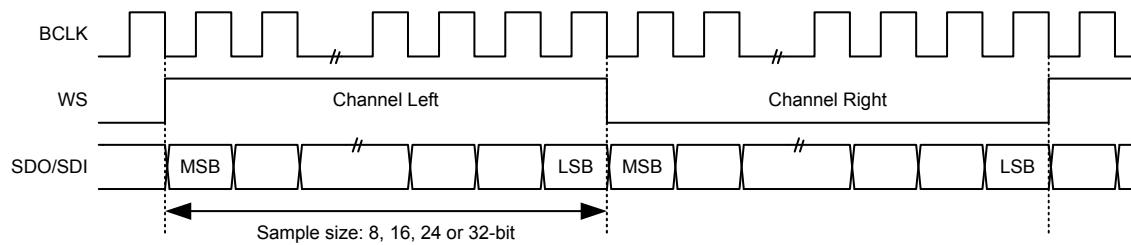


Figure 196. Left-justified Stereo Mode Waveforms

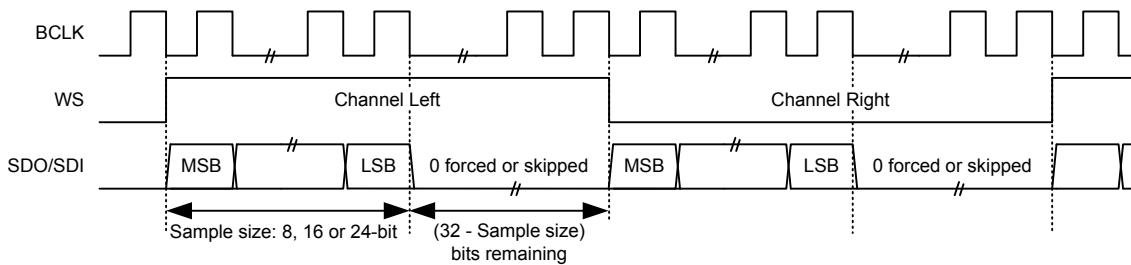


Figure 197. Left-justified Stereo Mode Waveforms (32-bit Channel Enabled)

### Right-justified Stereo Mode

Right-Justified mode is where the Least Significant Bit (LSB) of the stereo audio sample data is available on the rising edge of BCLK preceding a WS transition and where the MSB is transmitted first. Figure 198 and Figure 199 show a right I<sup>2</sup>S-justified stereo mode format.

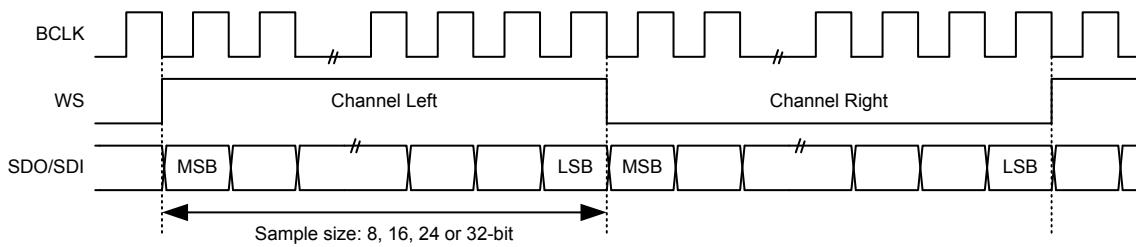


Figure 198. Right-justified Stereo Mode Waveforms

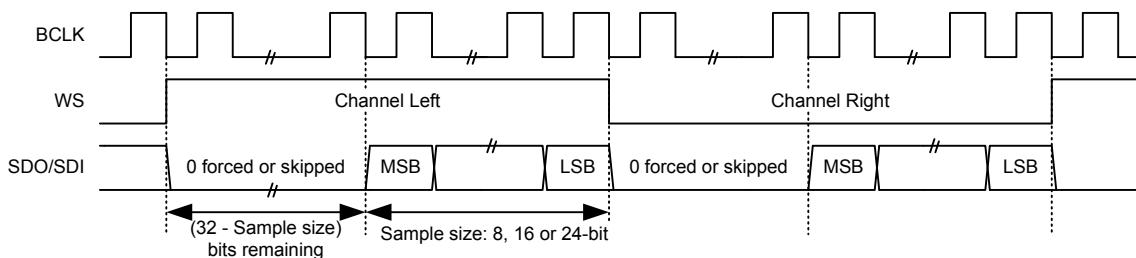
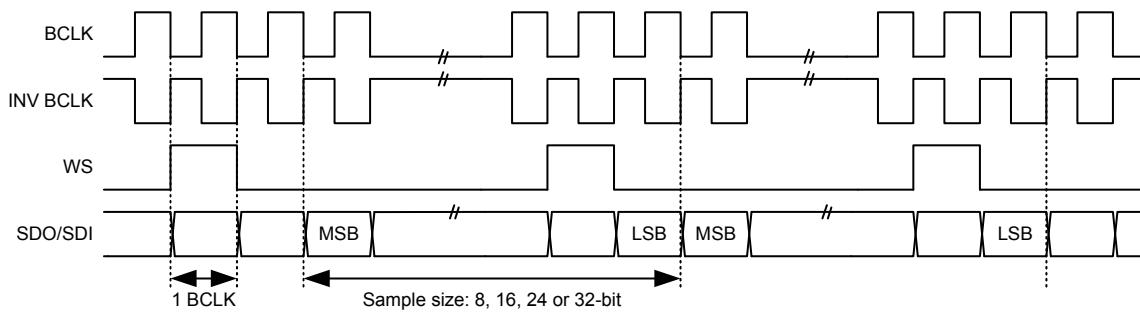


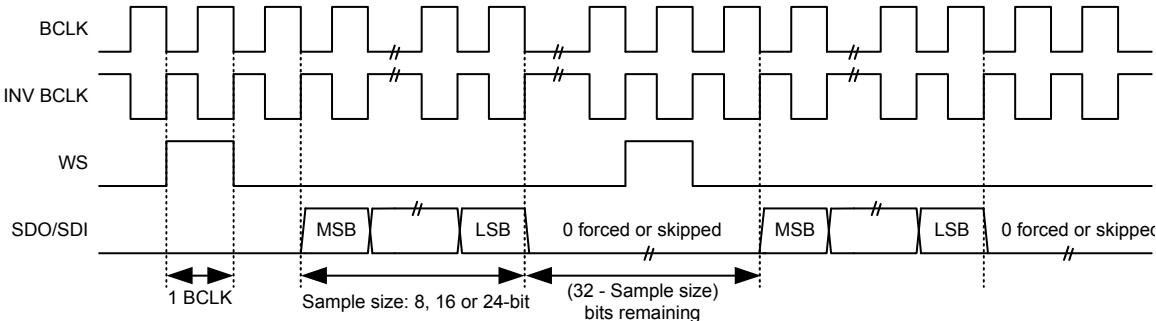
Figure 199. Right-justified Stereo Mode Waveforms (32-bit Channel Enabled)

### I<sup>2</sup>S-justified Mono Mode

In the I<sup>2</sup>S-justified mono mode, the Most Significant Bit (MSB) of the mono audio sample data is available on the second rising edge of the BCLK clock following a falling edge on the WS signal. Figure 200 and Figure 201 show an I<sup>2</sup>S-justified mono mode format.



**Figure 200. I<sup>2</sup>S-justified Mono Mode Waveforms**



**Figure 201. I<sup>2</sup>S-justified Mono Mode Waveforms (32-bit Channel Enabled)**

### Left-justified Mono Mode

In the left-justified mono mode, the Most Significant Bit (MSB) of the mono audio sample data is available on the first rising edge of the BCLK clock following a falling edge on the WS signal. Figure 202 and Figure 203 show a left-justified mono mode format.

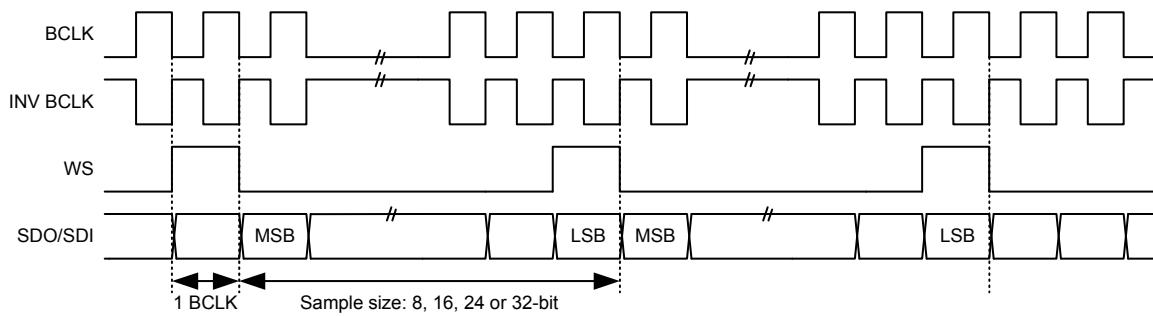


Figure 202. Left-justified Mono Mode Waveforms

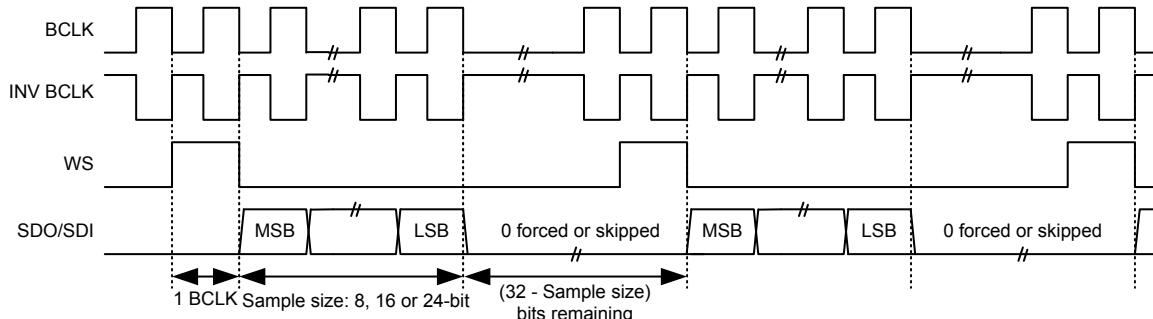


Figure 203. Left-justified Mono Mode Waveforms (32-bit Channel Enabled)

### Right-justified Mono Mode

In the right-justified mono mode, the Least Significant Bit (LSB) of the mono audio sample data is available on the last rising edge of the BCLK clock preceding a rising edge on the WS signal. Figure 204 and Figure 205 show the right-justified mono mode format.

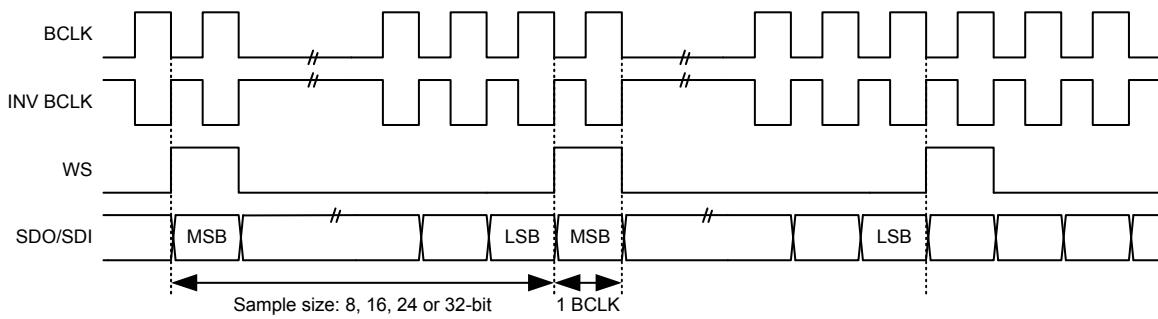


Figure 204. Right-justified Mono Mode Waveforms

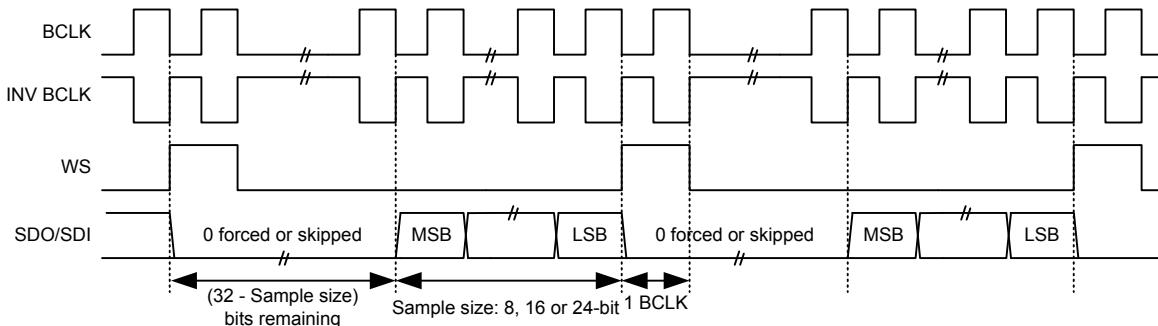


Figure 205. Right-justified Mono Mode Waveforms (32-bit Channel Enabled)

### I<sup>2</sup>S-justified Repeat Mode

In the I<sup>2</sup>S-justified repeat mode, the Most Significant Bit (MSB) of the mono audio sample data is available on the second rising edge of the BCLK clock following a WS signal transition. In this mode the same data is transmitted twice, once when WS is low and again when WS is high. Figure 206 and Figure 207 show the I<sup>2</sup>S-justified repeat mode format.

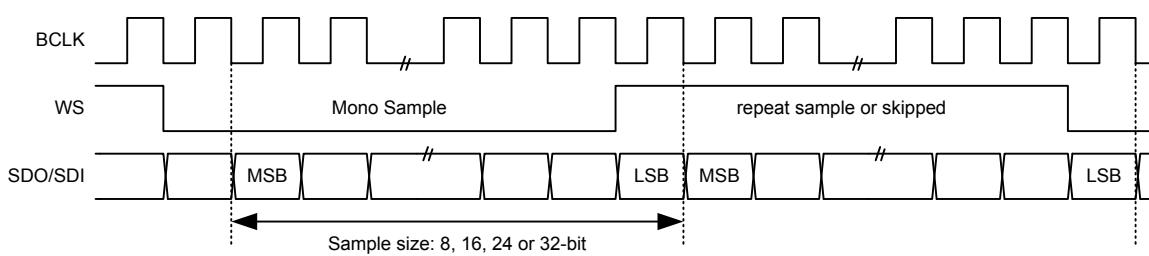


Figure 206. I<sup>2</sup>S-justified Repeat Mode Waveforms

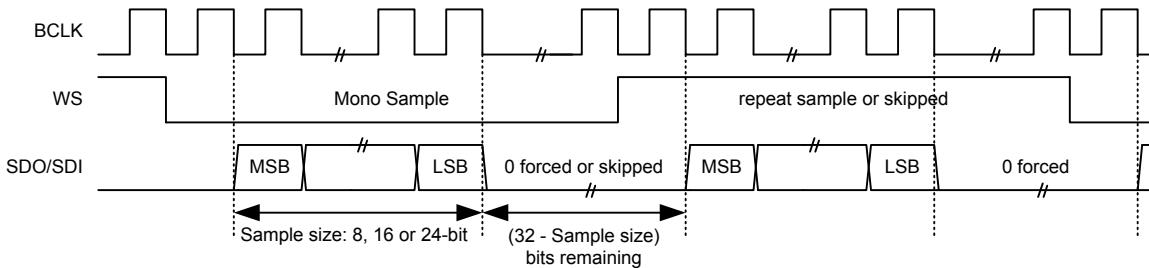
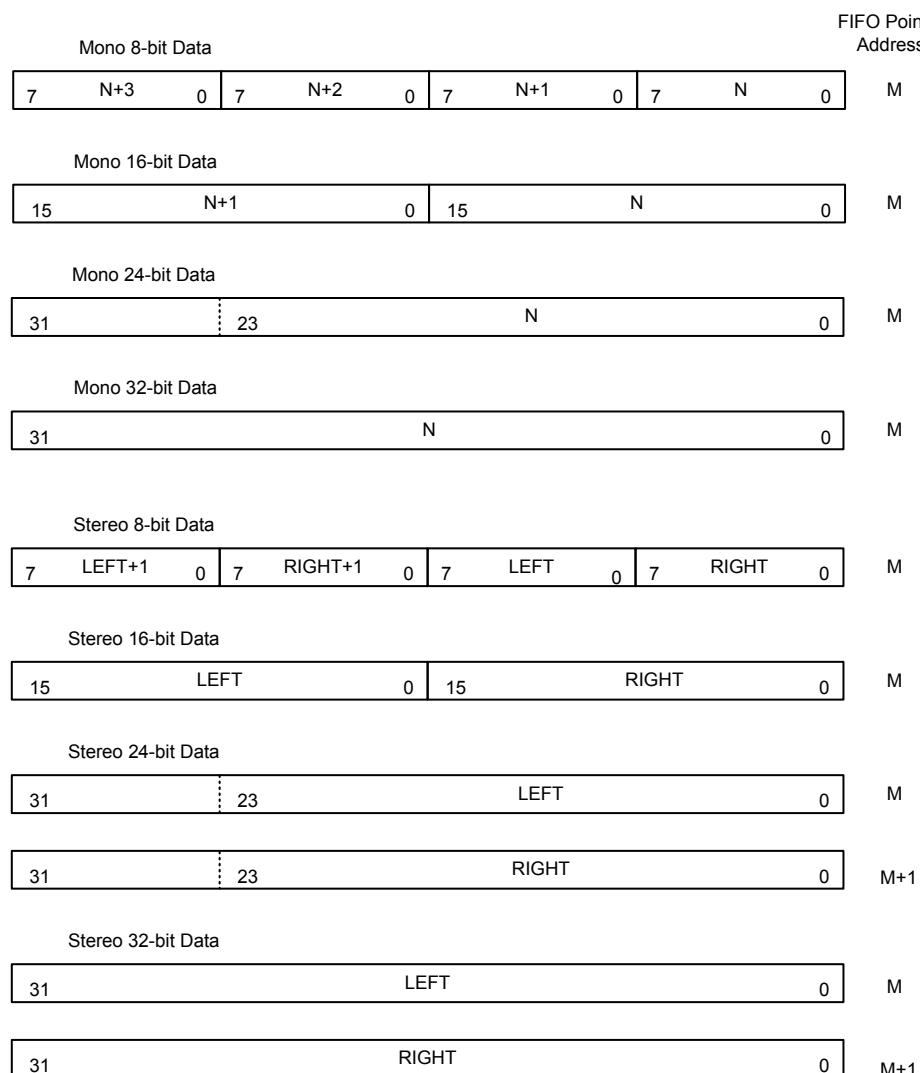


Figure 207. I<sup>2</sup>S-justified Repeat Mode Waveforms (32-bit Channel Enabled)

## FIFO Control and Arrangement

The I<sup>2</sup>S handles audio data for transmission and reception and is performed via the FIFO controller. Each transmitted or received FIFO has a depth of 8 words (8 x 32-bit) and can buffer the data. The format is dependent upon the stereo/mono mode and sample size setting. The detailed FIFO data content format is shown in Figure 208. The FIFO controller consists of comparators which compare the current FIFO levels with configurable depth settings. The current level of the TX or RX FIFO status can be seen in the TXFS and RXFS fields of the I<sup>2</sup>S status register (I2SSR).



**Figure 208. FIFO Data Content Arrangement for Various Modes**

## PDMA and Interrupt

When the level of received data in the RX FIFO is equal to or greater than the level defined by the RXFTLS field in the I<sup>2</sup>S FIFO control register (I2SFCR), the relative RXFTL flag will be set and then an I<sup>2</sup>S RX PDMA request will be generated. A CPU interrupt will be generated if the enable bit of the I<sup>2</sup>S RX PDMA request or the RX FIFO trigger level interrupt is asserted. When the level of transmitted data in the TX FIFO is equal to or less than the level defined by the TXFTLS field in the I<sup>2</sup>S FIFO control register (I2SFCR), the relative TXFTL flag will be set and a I<sup>2</sup>S TX PDMA request will be generated. A CPU interrupt will be generated if the enable bit of the I<sup>2</sup>S TX PDMA request or TX FIFO trigger level interrupt is asserted.

The I<sup>2</sup>S transmitter and receiver have separate PDMA requests and can be assigned to two different PDMA channels. When a PDMA request is enabled for the I<sup>2</sup>S transmitter (TXDMAEN = 1) then this will automatically request that data is transferred to the assigned I<sup>2</sup>S TX PDMA channel whenever TX FIFO space is available and TXFTL is active. When a PDMA request is enabled for the receiver (RXDMAEN = 1) then this will automatically request the data transfers to the I<sup>2</sup>S RX PDMA channel whenever data is present in the receive FIFO and when RXFTL is active.

## Register Map

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

**Table 74. I<sup>2</sup>S Register Map**

Register	Offset	Description	Reset Value
<b>I<sup>2</sup>S Base Address = 0x4002_6000</b>			
I2SCR	0x000	I <sup>2</sup> S Control Register	0x0000_0000
I2SIER	0x004	I <sup>2</sup> S Interrupt Enable Register	0x0000_0000
I2SCDR	0x008	I <sup>2</sup> S Clock Divider Register	0x0000_0000
I2STXDR	0x00C	I <sup>2</sup> S Tx Data Register	0x0000_0000
I2SRXDR	0x010	I <sup>2</sup> S Rx Data Register	0x0000_0000
I2SFCR	0x014	I <sup>2</sup> S FIFO Control Register	0x0000_0000
I2SSR	0x018	I <sup>2</sup> S Status Register	0x0000_0809
I2SRCNTR	0x01C	I <sup>2</sup> S Rate Counter Value Register	0x0000_0000

## Register Descriptions

### I<sup>2</sup>S Control Register – I2SCR

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

Offset: 0x000

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
Reserved									
Type/Reset	23	22	21	20	19	18	17	16	
	Reserved				MCKINV	BCKINV	RCSEL	RCEN	
Type/Reset					RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
Type/Reset	CLKDEN	RXDMAEN	TXDMAEN	TXMUTE	CHANNEL	REPEAT	MCLKEN	BITEXT	
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	FORMAT		SMPSIZE		MS	RXEN	TXEN	I2SEN	
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[19]	MCKINV	MCLK Inverse Enable 0: Disable 1: Enable
[18]	BCKINV	BCLK Inverse Enable 0: Disable 1: Enable
[17]	RCSEL	Rate Control Select (master only) 0: slower 1: faster
[16]	RCEN	Rate Control Enable (master only) 0: Disable 1: Enable
[15]	CLKDEN	Clock Divider Enable (master only) 0: Disable 1: Enable The clock divider can be used to generate the MCLK and BCLK clock of the I <sup>2</sup> S interface for master mode.
[14]	RXDMAEN	RX PDMA Request Enable 0: Disable 1: Enable
[13]	TXDMAEN	TX PDMA Request Enable 0: Disable 1: Enable
[12]	TXMUTE	TX Mute Enable 0: Disable 1: Enable

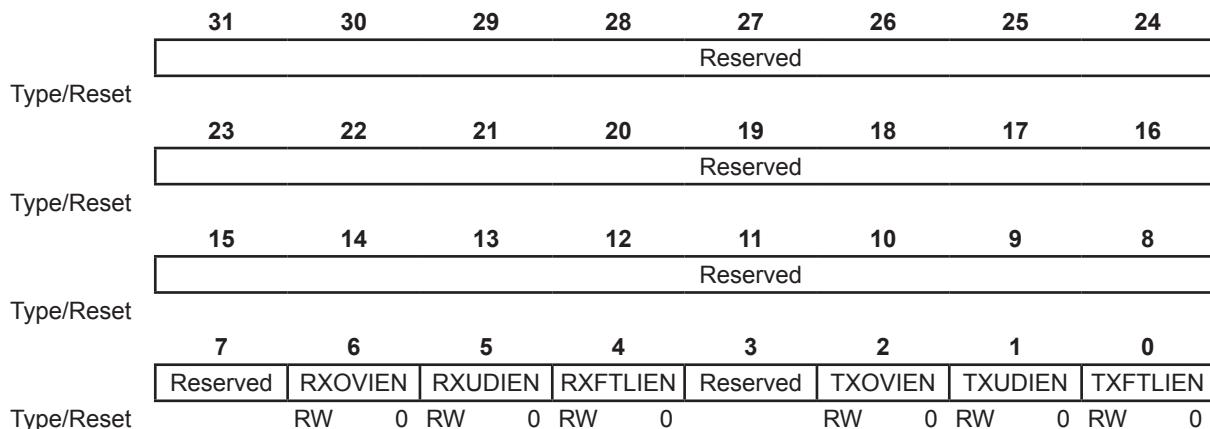
Bits	Field	Descriptions
[11]	CHANNEL	Stereo or Mono 0: Stereo 1: Mono Note: This bit should be configured when I <sup>2</sup> S is disabled.
[10]	REPEAT	Repeat Mode 0: Disable 1: Enable This mode is for I <sup>2</sup> S-justified stereo configuration only, transmitting the mono data on both channels and receiving just the left channel data and ignoring the right. Enabling the repeat mode will reset the CHANNEL bit automatically. Note: This bit should be configured when the I <sup>2</sup> S is disabled.
[9]	MCLKEN	MCLK Output Enable (master only) 0: Disable 1: Enable Note: This bit should be configured when the I <sup>2</sup> S is disabled.
[8]	BITEXT	32-bit Channel Enable 0: Disable 1: Enable Setting this bit will force the channel size to 32-bit. If the sample size is 8/16/24-bit, the remaining bits will be forced to 0 in the TX and ignored in the RX. Note: This bit should be configured when the I <sup>2</sup> S is disabled.
[7:6]	FORMAT	Data Format 00: I <sup>2</sup> S-justified 01: Left-justified 10: Right-justified 11: reserved Note: This bit should be configured when the I <sup>2</sup> S is disabled.
[5:4]	SMPSIZE	Sample Size 00: 8-bit 01: 16-bit 10: 24-bit 11: 32-bit Note: This bit should be configured when the I <sup>2</sup> S is disabled.
[3]	MS	Master or Slave Mode 0: Master 1: Slave Note: This bit should be configured when the I <sup>2</sup> S is disabled.
[2]	RXEN	RX Enable 0: Disable 1: Enable Note: This bit should be configured when the I <sup>2</sup> S is disabled.
[1]	TXEN	TX Enable 0: Disable 1: Enable
[0]	I <sup>2</sup> SEN	I <sup>2</sup> S Enable 0: Disable 1: Enable

## I<sup>2</sup>S Interrupt Enable Register – I2SIER

This register contains the corresponding I<sup>2</sup>S interrupt enable bits.

Offset: 0x004

Reset value: 0x0000\_0000



Bits	Field	Descriptions
[6]	RXOVIEN	RX FIFO Overflow Interrupt Enable 0: Disable 1: Enable
[5]	RXUDIEN	RX FIFO Underflow Interrupt Enable 0: Disable 1: Enable
[4]	RXFTLIEN	RX FIFO Trigger Level Interrupt Enable 0: Disable 1: Enable
[2]	TXOVIEN	TX FIFO Overflow Interrupt Enable 0: Disable 1: Enable
[1]	TXUDIEN	TX FIFO Underflow Interrupt Enable 0: Disable 1: Enable
[0]	TXFTLIEN	TX FIFO Trigger Level Interrupt Enable 0: Disable 1: Enable

## I<sup>2</sup>S Clock Divider Register – I2SCDR

This register specifies the I<sup>2</sup>S clock divider ratio.

Offset: 0x008

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[23:16]	N_DIV	N divider for BCLK Note: This bit should be configured when the I <sup>2</sup> S is disabled.
[15:8]	X_DIV	X divider for MCLK (X != 0) && (Y != 0) && (X/Y >= 1/255) && (X/Y <= 127/255) Note: This bit should be configured when the I <sup>2</sup> S is disabled.
[7:0]	Y_DIV	Y divider for MCLK (X != 0) && (Y != 0) && (X/Y >= 1/255) && (X/Y <= 127/255) Note: This bit should be configured when the I <sup>2</sup> S is disabled.

## I<sup>2</sup>S Tx Data Register – I2STXDR

This register is used to specify the I<sup>2</sup>S transmitted data.

Offset: 0x00C

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[31:0]	TXDR	TX Data Register

## I<sup>2</sup>S Rx Data Register – I2SRXDR

This register is used to store the I<sup>2</sup>S received data.

Offset: 0x010

Reset value: 0x0000\_0000

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

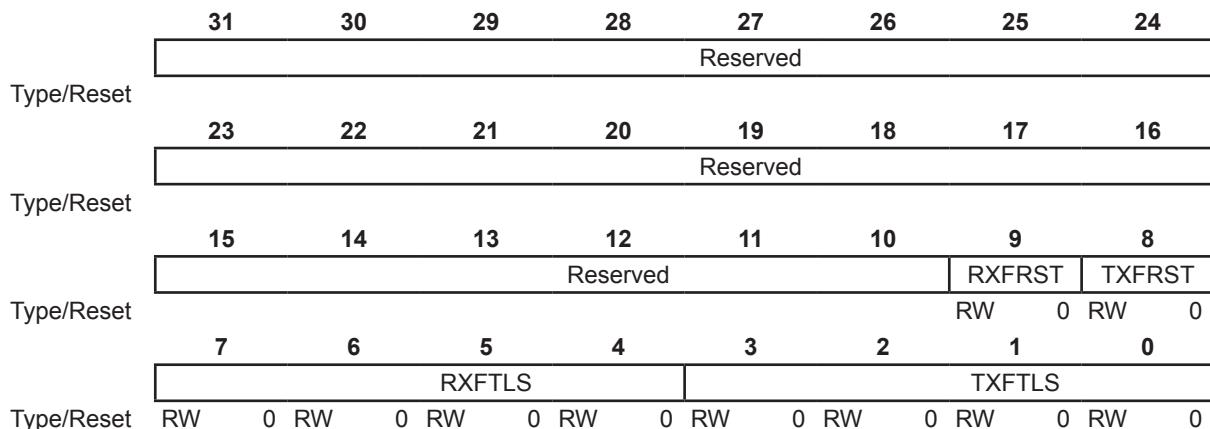
Bits	Field	Descriptions
[31:0]	RXDR	RX Data Register

## I<sup>2</sup>S FIFO Control Register – I2SFCR

This register contains the related I<sup>2</sup>S FIFO control bits.

Offset: 0x014

Reset value: 0x0000\_0000



Bits	Field	Descriptions
[9]	RXFRST	RX FIFO Reset Set this bit to reset the Rx FIFO.
[8]	TXFRST	TX FIFO Reset Set this bit to reset the Tx FIFO.
[7:4]	RXFTLS	RX FIFO Trigger Level Select 0000: Trigger level is 0 0001: Trigger level is 1 ... 0111: Trigger level is 7 1xxx: Trigger level is 8 When the data contained in the Rx FIFO is equal to or greater than the level defined by the RXFTLS field, the RXFTL flag will be set.
[3:0]	TXFTLS	TX FIFO Trigger Level Select 0000: Trigger level is 0 0001: Trigger level is 1 ... 0111: Trigger level is 7 1xxx: Trigger level is 8 When the data contained in the Tx FIFO is equal to or less than the level defined by the TXFTLS field, the TXFTL flag will be set.

## I<sup>2</sup>S Status Register – I2SSR

This register contains the relevant I<sup>2</sup>S status.

Offset: 0x018

Reset value: 0x0000\_0809

	31	30	29	28	27	26	25	24
Type/Reset		R0	0	R0	0	R0	0	R0
	23	22	21	20	19	18	17	16
Type/Reset			Reserved		CLKRDY	TXBUSY	CHS	
					R0	0	R0	0
Type/Reset	15	14	13	12	11	10	9	8
		Reserved		RXFFUL	RXFEMT	RXFOV	RXFUD	RXFTL
Type/Reset				R0	0	R0	1	WC
	7	6	5	4	3	2	1	0
Type/Reset			Reserved	TXFFUL	TXFEMT	TXFOV	TXFUD	TXFTL
				R0	0	R0	1	WC
						1	0	WC
								1

Bits	Field	Descriptions
[31:28]	RXFS	Rx FIFO Status 0000: Rx FIFO empty 0001: Rx FIFO contains 1 data ... 1000: Rx FIFO contains 8 data Others: Reserved
[27:24]	TXFS	Tx FIFO Status 0000: Tx FIFO empty 0001: Tx FIFO contains 1 data ... 1000: Tx FIFO contains 8 data Others: Reserved
[18]	CLKRDY	Clock Divider Output Ready Flag 0: Not ready 1: Ready
[17]	TXBUSY	Tx Busy Flag 0: Not busy 1: Busy
[16]	CHS	Channel Status 0: Left channel 1: Right channel
[12]	RXFFUL	Rx FIFO Full Flag 0: Rx FIFO not full 1: Rx FIFO full
[11]	RXFEMT	Rx FIFO Empty Flag 0: Rx FIFO not empty 1: Rx FIFO empty

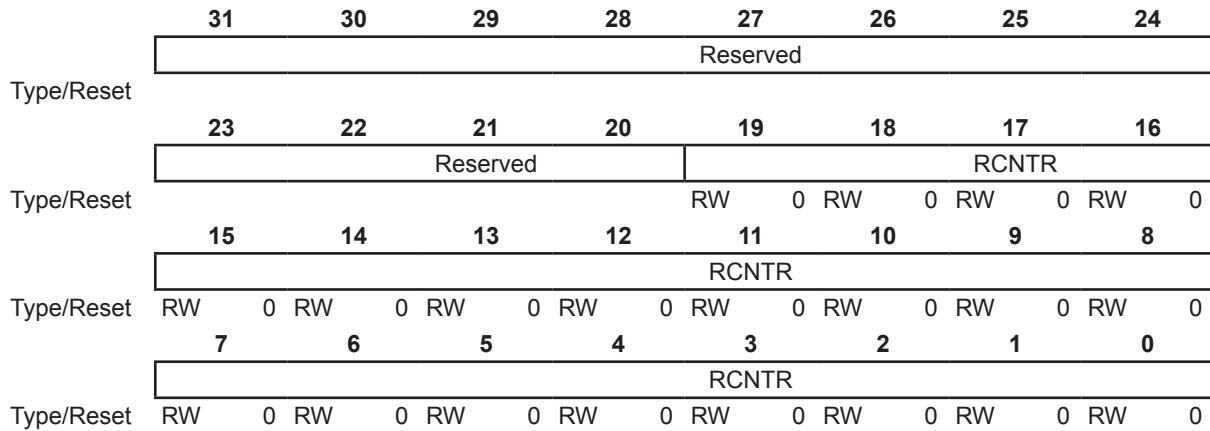
Bits	Field	Descriptions
[10]	RXFOV	Rx FIFO Overflow Flag 0: Rx FIFO not overflow 1: Rx FIFO overflow This bit is set by hardware and cleared by writing 1.
[9]	RXFUD	Rx FIFO Underflow Flag 0: Rx FIFO not underflow 1: Rx FIFO underflow This bit is set by hardware and cleared by writing 1.
[8]	RXFTL	Rx FIFO Trigger Level Flag 0: Data in the Rx FIFO is less than the trigger level 1: Data in the Rx FIFO is equal to or higher than the trigger level This bit is set by hardware and cleared by writing 1.
[4]	TXFFUL	Tx FIFO Full Flag 0: Tx FIFO not full 1: Tx FIFO full
[3]	TXFEMT	Tx FIFO Empty Flag 0: Tx FIFO not empty 1: Tx FIFO empty
[2]	TXFOV	Tx FIFO Overflow Flag 0: Tx FIFO not overflow 1: Tx FIFO overflow This bit is set by hardware and cleared by writing 1.
[1]	TXFUD	Tx FIFO Underflow Flag 0: Tx FIFO not underflow 1: Tx FIFO underflow This bit is set by hardware and cleared by writing 1.
[0]	TXFTL	Tx FIFO Trigger Level Flag 0: Data in the Tx FIFO is higher than the trigger level 1: Data in the Tx FIFO is equal to or less than the trigger level This bit is set by hardware and cleared by writing 1.

## I<sup>2</sup>S Rate Counter Value Register – I2SRCNTR

This register specifics the I<sup>2</sup>S rate control counter value.

Offset: 0x01C

Reset value: 0x0000\_0000



Bits	Field	Descriptions
[19:0]	RCNTR	Rate Counter Value This value must be equal to or higher than Y_DIV for useful rate fine-tuning control.

## 28 Cyclic Redundancy Check (CRC)

### Introduction

The CRC (Cyclic Redundancy Check) calculation unit is an error detection technique test algorithm and uses to verify data transmission or storage data correctness. A CRC calculation takes a data stream or a block of data as input and generates a 16- or 32-bit output remainder. Ordinarily, a data stream is suffixed by a CRC code and used as a checksum when being sent or stored. Therefore, the received or restored data stream is calculated by the same generator polynomial as described above. If the new CRC code result does not match the one calculated earlier, that means data stream contains a data error.

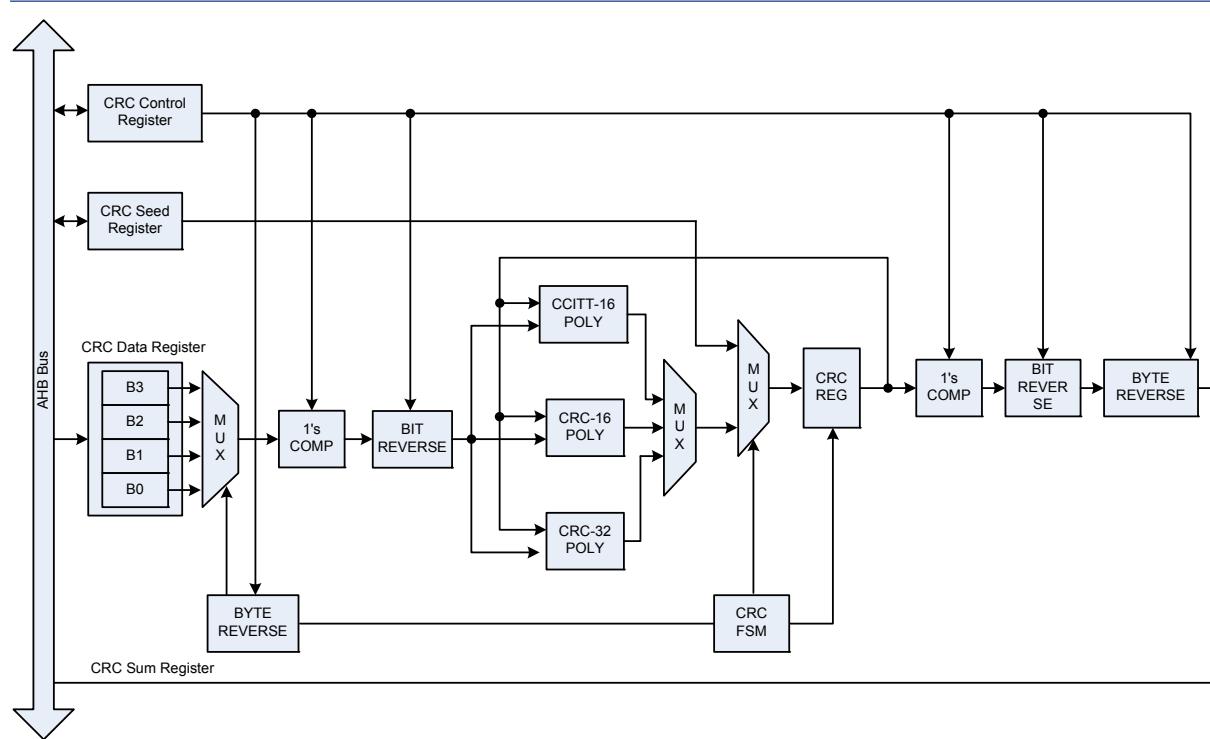


Figure 209. CRC Block Diagram

## Features

- Support CRC16 polynomial: 0x8005,  $X^{16}+X^{15}+X^2+1$
- Support CCITT CRC16 polynomial: 0x1021,  $X^{16}+X^{12}+X^5+1$
- Support IEEE-802.3 CRC32 polynomial: 0x04C11DB7,  $X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$
- Support 1's complement, byte reverse & bit reverse operation on data and checksum
- Support byte, half-word & word data size
- Programmable CRC initial seed value
- CRC computation done in 1 AHB clock cycle for 8-bit data and 4 AHB clock cycles for 32-bit data
- Support PDMA to complete a CRC computation of a block of memory

## Function Descriptions

This unit only enables the calculation in the CRC16, CCITT CRC16 and IEEE-802.3 CRC32 polynomial. In this unit, the generator polynomial is fixed to the numeric values for those modes; therefore, the CRC value based on other generator polynomials cannot be calculated.

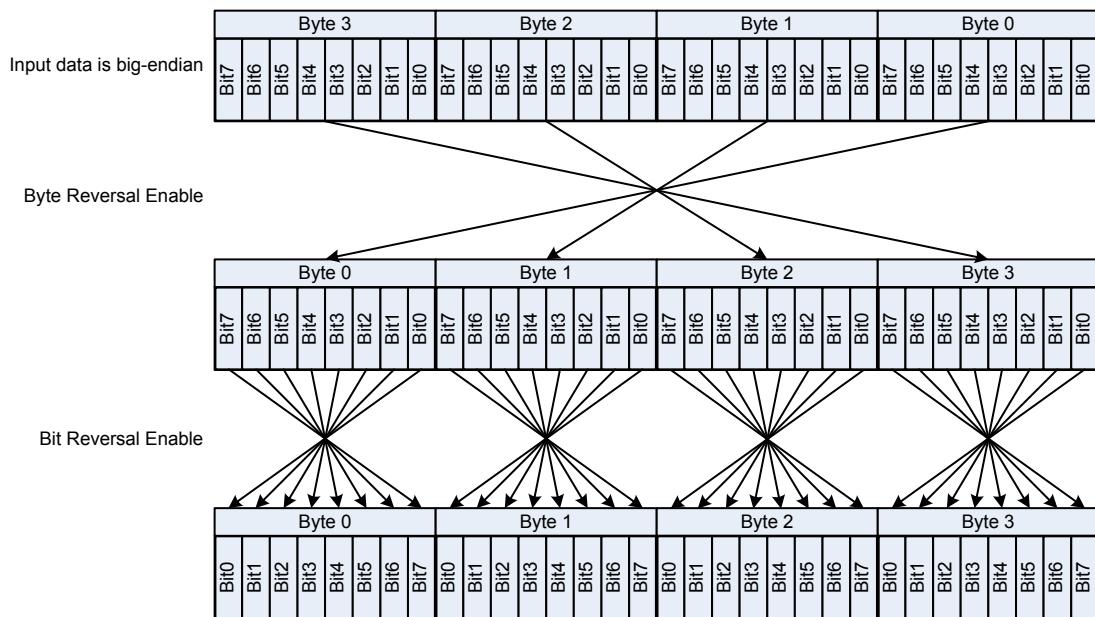
### CRC Computation

The CRC calculation unit has 32-bit write CRC data register (CRCDR) and read CRC checksum register (CRCCSR). The CRCDR register is used to input new data (write access), and the CRCCSR register is used to hold the result of the previous CRC calculation (read access). Each write operation to the CRCDR register creates a combination of the previous CRC value (stored in CRCCSR) and the new one. The CRC block diagram is shown as Figure 209. 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 210 is shown the byte and bit reversal operation example.



**Figure 210. CRC Data Bit and Byte Reversal Example**

## CRC with PDMA

A PDMA channel with software trigger may be used to transfer data into the CRC unit. If a huge block data is needed to calculate. The recommended PDMA model is to use the PDMA to transfer all available words of data and uses software writes to transfer the other remaining bytes. To write data into the CRC unit, the PDMA should be accessed data write by word from the source location of memory to the CRC data register (CRCDR) in non-incrementing address mode. Then software can write any remaining bytes to the CRC data register (CRCDR) and read the CRC calculation result value from the CRC checksum register (CRCCSR).

## Register Map

The following table shows the CRC registers and reset values.

**Table 75. CRC Register Map**

Register	Offset	Description	Reset Value
<b>CRC Base Address = 0x4008A000</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

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

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

Bits	Field	Descriptions
[31:0]	SEED	CRC Seed Data Put the 16/32-bit seed value in this register according to the polynomial setting in the CRCCR register.

## CRC Checksum Register – CRCCSR

This register contains the CRC checksum output.

Offset: 0x008

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
CHKSUM								
Type/Reset	RO	0	RO	0	RO	0	RO	0
23      22      21      20      19      18      17      16								
CHKSUM								
Type/Reset	RO	0	RO	0	RO	0	RO	0
15      14      13      12      11      10      9      8								
CHKSUM								
Type/Reset	RO	0	RO	0	RO	0	RO	0
7      6      5      4      3      2      1      0								
CHKSUM								
Type/Reset	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								
Type/Reset	WO	0	WO	0	WO	0	WO	0
CRCDATA								
Type/Reset	WO	0	WO	0	WO	0	WO	0
15 14 13 12 11 10 9 8								
Type/Reset	WO	0	WO	0	WO	0	WO	0
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© 2015 by HOLTEK SEMICONDUCTOR INC.

The information appearing in this Data Sheet is believed to be accurate at the time of publication. However, Holtek assumes no responsibility arising from the use of the specifications described. The applications mentioned herein are used solely for the purpose of illustration and Holtek makes no warranty or representation that such applications will be suitable without further modification, nor recommends the use of its products for application that may present a risk to human life due to malfunction or otherwise. Holtek's products are not authorized for use as critical components in life support devices or systems. Holtek reserves the right to alter its products without prior notification. For the most up-to-date information, please visit our web site at <http://www.holtek.com.tw>.