



---

**Holtek 32-Bit Microcontroller with Arm® Cortex®-M3 Core**

# **HT32F12364 User Manual**

Revision: V1.10    Date: September 30, 2021

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

## Table of Contents

<b>1 Introduction .....</b>	<b>24</b>
Overview .....	24
Features .....	25
Device Information .....	29
Block Diagram .....	30
<b>2 Document Conventions .....</b>	<b>31</b>
<b>3 System Architecture .....</b>	<b>32</b>
Arm® Cortex®-M3 Processor .....	32
Bus Architecture .....	33
Memory Organization .....	34
Memory Map .....	35
Embedded Flash Memory .....	38
Embedded SRAM Memory .....	38
AHB Peripherals .....	38
APB Peripherals .....	38
<b>4 Flash Memory Controller (FMC) .....</b>	<b>39</b>
Introduction .....	39
Features .....	39
Functional Descriptions .....	40
Flash Memory Map .....	40
Flash Memory Architecture .....	41
Wait State Setting .....	41
Page Erase .....	43
Mass Erase .....	44
Word Programming .....	45
Option Byte Description .....	46
Page Erase/Program Protection .....	47
Security Protection .....	48
Register Map .....	49
Register Descriptions .....	50
Flash Target Address Register – TADR .....	50
Flash Write Data Register – WRDR .....	51
Flash Operation Command Register – OCMR .....	52
Flash Operation Control Register – OPCR .....	53
Flash Operation Interrupt Enable Register – OIER .....	54
Flash Operation Interrupt and Status Register – OISR .....	55
Flash Page Erase/Program Protection Status Register – PPSR .....	57
Flash Security Protection Status Register – CPSR .....	58
Flash Vector Mapping Control Register – VMCR .....	59

Flash Manufacturer and Device ID Register – MDID .....	60
Flash Page Number Status Register – PNSR .....	61
Flash Page Size Status Register – PSSR .....	62
Device ID Register – DID .....	62
Flash Cache and Pre-fetch Control Register – CFCR .....	63
SRAM Booting Vector Register n – SBVTn (n = 0 ~ 3) .....	64
Custom ID Register n – CIDRn (n = 0 ~ 3) .....	65
<b>5 Power Control Unit (PWRCU) .....</b>	<b>66</b>
Introduction .....	66
Features .....	67
Functional Descriptions .....	67
V <sub>DD</sub> Power Domain .....	67
V <sub>CORE</sub> Power Domain .....	69
Operation Modes .....	69
Register Map .....	72
Register Descriptions .....	72
Power Control Status Register – PWRSR .....	72
Power Control Register – PWRCR .....	73
V <sub>DD</sub> Power Domain Test Register – PWRTEST .....	75
Low Voltage / Brown-Out Detect Control and Status Register – LVDCSR .....	76
Power Control LDO Status Register – PWRLDOSR .....	78
<b>6 Clock Control Unit (CKCU) .....</b>	<b>79</b>
Introduction .....	79
Features .....	81
Function Descriptions .....	81
High Speed External Crystal Oscillator – HSE .....	81
High Speed Internal RC Oscillator – HSI .....	82
Auto Trimming of High Speed Internal RC Oscillator – HSI .....	82
Phase Locked Loop – PLL .....	84
USB Phase Locked Loop – USB PLL .....	86
Low Speed External Crystal Oscillator – LSE .....	88
Low Speed Internal RC Oscillator – LSI .....	88
Clock Ready Flag .....	88
System Clock (CK_SYS) Selection .....	89
HSE Clock Monitor .....	90
Clock Output Capability .....	90
Register Map .....	91
Register Descriptions .....	92
Global Clock Configuration Register – GCFGR .....	92
Global Clock Control Register – GCCR .....	93
Global Clock Status Register – GCSR .....	95
Global Clock Interrupt Register – GCIR .....	96

PLL Configuration Register – PLLCFGR .....	98
PLL Control Register – PLLCR .....	99
AHB Configuration Register – AHBCFGR .....	100
AHB Clock Control Register – AHBCCR .....	101
APB Configuration Register – APBCFGR .....	103
APB Clock Control Register 0 – APBCCR0 .....	104
APB Clock Control Register 1 – APBCCR1 .....	105
Clock Source Status Register – CKST .....	106
APB Peripheral Clock Selection Register 0 – APBPCSR0 .....	107
APB Peripheral Clock Selection Register 1 – APBPCSR1 .....	109
HSI Control Register – HSICR .....	110
HSI Auto Trimming Counter Register – HSIATCR .....	112
APB Peripheral Clock Selection Register 2 – APBPCSR2 .....	113
Low Power Control Register – LPCR .....	114
MCU Debug Control Register – MCUDBGCR .....	115
<b>7 Reset Control Unit (RSTCU) .....</b>	<b>117</b>
Introduction .....	117
Functional Descriptions .....	118
Power On Reset .....	118
System Reset .....	118
AHB and APB Unit Reset .....	118
Register Map .....	119
Register Descriptions .....	119
Global Reset Status Register – GRSR .....	119
AHB Peripheral Reset Register – AHBPRSTR .....	120
APB Peripheral Reset Register 0 – APBPRSTR0 .....	121
APB Peripheral Reset Register 1 – APBPRSTR1 .....	123
<b>8 General Purpose I/O (GPIO) .....</b>	<b>125</b>
Introduction .....	125
Features .....	126
Functional Descriptions .....	126
Default GPIO Pin Configuration .....	126
General Purpose I/O – GPIO .....	126
GPIO Locking Mechanism .....	128
Register Map .....	128
Register Descriptions .....	130
Port A Data Direction Control Register – PADIRCR .....	130
Port A Input Function Enable Control Register – PAINER .....	131
Port A Pull-Up Selection Register – PAPUR .....	132
Port A Pull-Down Selection Register – PAPDR .....	133
Port A Open-Drain Selection Register – PAODR .....	134
Port A Drive Current Selection Register – PADRVR .....	135

Port A Lock Register – PALOCKR .....	136
Port A Data Input Register – PADINR .....	137
Port A Output Data Register – PADOUTR .....	137
Port A Output Set/Reset Control Register – PASRR .....	138
Port A Output Reset Register – PARR .....	139
Port B Data Direction Control Register – PBDIRCR .....	139
Port B Input Function Enable Control Register – PBINER .....	140
Port B Pull-Up Selection Register – PBPUR .....	141
Port B Pull-Down Selection Register – PBPDR .....	142
Port B Open-Drain Selection Register – PBODR .....	143
Port B Drive Current Selection Register – PBDRVR .....	144
Port B Lock Register – PBLOCKR .....	145
Port B Data Input Register – PBDINR .....	146
Port B Output Data Register – PBDOUTR .....	146
Port B Output Set/Reset Control Register – PBSRR .....	147
Port B Output Reset Register – PBRR .....	148
Port C Data Direction Control Register – PCDIRCR .....	148
Port C Input Function Enable Control Register – PCINER .....	149
Port C Pull-Up Selection Register – PCPUR .....	150
Port C Pull-Down Selection Register – PCPDR .....	151
Port C Open-Drain Selection Register – PCODR .....	152
Port C Drive Current Selection Register – PCDRVR .....	153
Port C Lock Register – PCLOCKR .....	154
Port C Data Input Register – PCDINR .....	155
Port C Output Data Register – PCDOUTR .....	155
Port C Output Set/Reset Control Register – PCSRR .....	156
Port C Output Reset Register – PCRR .....	157
Port D Data Direction Control Register – PDDIRCR .....	157
Port D Input Function Enable Control Register – PDINER .....	158
Port D Pull-Up Selection Register – PDPUR .....	159
Port D Pull-Down Selection Register – PDPDR .....	160
Port D Open-Drain Selection Register – PDODR .....	161
Port D Drive Current Selection Register – PDDRVR .....	162
Port D Lock Register – PDLOCKR .....	163
Port D Data Input Register – PDDINR .....	164
Port D Output Data Register – PDDOUTR .....	164
Port D Output Set/Reset Control Register – PDSRR .....	165
Port D Output Reset Register – PDRR .....	166
Port F Data Direction Control Register – PFDIRCR .....	166
Port F Input Function Enable Control Register – PFINER .....	167
Port F Pull-Up Selection Register – PFPUR .....	168
Port F Pull-Down Selection Register – PFPDR .....	169
Port F Open-Drain Selection Register – PFODR .....	170
Port F Drive Current Selection Register – PFDRVR .....	171
Port F Lock Register – PFLOCKR .....	172

Port F Data Input Register – PFDINR.....	173
Port F Output Data Register – PFDOUTR.....	173
Port F Output Set/Reset Control Register – PFSRR .....	174
Port F Output Reset Register – PFRR.....	175
<b>9 Alternate Function Input/Output Control Unit (AFIO).....</b>	<b>176</b>
Introduction .....	176
Features.....	177
Functional Descriptions .....	177
External Interrupt Pin Selection .....	177
Alternate Function.....	178
Lock Mechanism .....	178
Register Map .....	178
Register Descriptions.....	179
EXTI Source Selection Register 0 – ESSR0 .....	179
EXTI Source Selection Register 1 – ESSR1 .....	180
GPIO Port x Configuration Low Register – GPxCFGRLR (x = A, B, C, D, F) .....	181
GPIO Port x Configuration High Register – GPxCFGHR (x = A, B, C, D, F) .....	182
<b>10 Nested Vectored Interrupt Controller (NVIC).....</b>	<b>183</b>
Introduction .....	183
Features.....	185
Function Descriptions .....	186
SysTick Calibration .....	186
Register Map .....	186
<b>11 External Interrupt/Event Controller (EXTI).....</b>	<b>188</b>
Introduction .....	188
Features.....	188
Functional Descriptions .....	189
Wakeup Event Management.....	189
External Interrupt/Event Line Mapping .....	190
Interrupt and Debounce .....	191
Register Map .....	192
Register Descriptions.....	193
EXTI Interrupt n Configuration Register – EXTICFGRn (n = 0 ~ 15).....	193
EXTI Interrupt Control Register – EXTICR .....	194
EXTI Interrupt Edge Flag Register – EXTIEDGEFLGR.....	194
EXTI Interrupt Edge Status Register – EXTIEDGESR .....	195
EXTI Interrupt Software Set Command Register – EXTISSCR.....	195
EXTI Interrupt Wakeup Control Register – EXTIWAKUPCR .....	196
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 Descriptions	200
ADC Clock Setup	200
Channel Selection	200
Conversion Mode	200
Start Conversion on External Event	203
Sampling Time Setting	204
Data Format	204
Analog Watchdog	204
Interrupts	205
PDMA Request	205
Voltage Reference Generator	205
V <sub>DDA</sub> Voltage Monitor	206
Register Map	206
Register Descriptions	207
ADC Conversion Control Register – ADCCR	207
ADC Conversion List Register 0 – ADCLST0	209
ADC Conversion List Register 1 – ADCLST1	210
ADC Input Sampling Time Register – ADCSTR	211
ADC Conversion Data Register y – ADCDRy, y = 0 ~ 7	212
ADC Trigger Control Register – ADCTCR	213
ADC Trigger Source Register – ADCTSR	214
ADC Watchdog Control Register – ADCWCR	215
ADC Watchdog Threshold Register – ADCTR	217
ADC Interrupt Enable Register – ADCIER	218
ADC Interrupt Raw Status Register – ADCIRAW	219
ADC Interrupt Status Register – ADCISR	220
ADC Interrupt Clear Register – ADCICLR	221
ADC DMA Request Register – ADCDMAR	222
Voltage Reference Control Register – VREFCR	223
Voltage Reference Value Register – VREFVALR	224
<b>13 General-Purpose Timer (GPTM)</b>	<b>225</b>
Introduction	225
Features	226
Functional Descriptions	226
Counter Mode	226
Clock Controller	228
Trigger Controller	230
Slave Controller	231
Master Controller	233
Channel Controller	234

Input Stage .....	236
Quadrature Decoder .....	238
Output Stage.....	239
Update Management .....	243
Single Pulse Mode.....	244
Asymmetric PWM Mode .....	246
Timer Interconnection .....	246
Trigger Peripherals Start.....	249
PDMA Request .....	249
Register Map .....	250
Register Descriptions.....	251
Timer Counter Configuration Register – CNTCFR .....	251
Timer Mode Configuration Register – MDCFR.....	252
Timer Trigger Configuration Register – TRCFR.....	255
Timer Control Register – CTR .....	256
Channel 0 Input Configuration Register – CH0ICFR .....	257
Channel 1 Input Configuration Register – CH1ICFR .....	258
Channel 2 Input Configuration Register – CH2ICFR .....	260
Channel 3 Input Configuration Register – CH3ICFR .....	261
Channel 0 Output Configuration Register – CH0OCFR .....	263
Channel 1 Output Configuration Register – CH1OCFR .....	265
Channel 2 Output Configuration Register – CH2OCFR .....	267
Channel 3 Output Configuration Register – CH3OCFR .....	269
Channel Control Register – CHCTR.....	271
Channel Polarity Configuration Register – CHPOLR.....	272
Timer PDMA/Interrupt Control Register – DICTR.....	273
Timer Event Generator Register – EVGR.....	275
Timer Interrupt Status Register – INTSR.....	276
Timer Counter Register – CNTR.....	279
Timer Prescaler Register – PSCR .....	280
Timer Counter-Reload Register – CRR .....	281
Channel 0 Capture/Compare Register – CH0CCR .....	282
Channel 1 Capture/Compare Register – CH1CCR .....	283
Channel 2 Capture/Compare Register – CH2CCR .....	284
Channel 3 Capture/Compare Register – CH3CCR .....	285
Channel 0 Asymmetric Compare Register – CH0ACR.....	286
Channel 1 Asymmetric Compare Register – CH1ACR.....	286
Channel 2 Asymmetric Compare Register – CH2ACR.....	287
Channel 3 Asymmetric Compare Register – CH3ACR.....	287
<b>14 Pulse-Width-Modulation Timer (PWM).....</b>	<b>288</b>
Introduction .....	288
Features.....	289
Functional Descriptions .....	289
Counter Mode .....	289



Clock Controller .....	292
Trigger Controller .....	293
Slave Controller .....	294
Master Controller .....	296
Channel Controller .....	297
Output Stage .....	297
Update Management .....	301
Single Pulse Mode .....	302
Asymmetric PWM Mode .....	304
Timer Interconnection .....	304
Trigger Peripherals Start .....	307
PDMA Request .....	307
Register Map .....	308
Register Descriptions .....	309
Timer Counter Configuration Register – CNTCFR .....	309
Timer Mode Configuration Register – MDCFR .....	310
Timer Trigger Configuration Register – TRCFR .....	313
Timer Control Register – CTR .....	314
Channel 0 Output Configuration Register – CH0OCFR .....	315
Channel 1 Output Configuration Register – CH1OCFR .....	317
Channel 2 Output Configuration Register – CH2OCFR .....	319
Channel 3 Output Configuration Register – CH3OCFR .....	321
Channel Control Register – CHCTR .....	323
Channel Polarity Configuration Register – CHPOLR .....	324
Timer PDMA/Interrupt Control Register – DICTR .....	325
Timer Event Generator Register – EVGR .....	326
Timer Interrupt Status Register – INTSR .....	327
Timer Counter Register – CNTR .....	329
Timer Prescaler Register – PSCR .....	329
Timer Counter-Reload Register – CRR .....	330
Channel 0 Compare Register – CH0CR .....	330
Channel 1 Compare Register – CH1CR .....	331
Channel 2 Compare Register – CH2CR .....	331
Channel 3 Compare Register – CH3CR .....	332
Channel 0 Asymmetric Compare Register – CH0ACR .....	332
Channel 1 Asymmetric Compare Register – CH1ACR .....	333
Channel 2 Asymmetric Compare Register – CH2ACR .....	333
Channel 3 Asymmetric Compare Register – CH3ACR .....	334
<b>15 Single-Channel Timer (SCTM) .....</b>	<b>335</b>
Introduction .....	335
Features .....	336
Functional Descriptions .....	336
Counter Mode .....	336
Clock Controller .....	337

Trigger Controller .....	338
Slave Controller .....	339
Channel Controller .....	341
Input Stage .....	342
Output Stage.....	343
Update Management .....	345
Register Map .....	346
Register Descriptions.....	347
Timer Counter Configuration Register – CNTCFR .....	347
Timer Mode Configuration Register – MDCFR.....	348
Timer Trigger Configuration Register – TRCFR.....	349
Timer Control Register – CTR .....	350
Channel Input Configuration Register – CHICFR .....	351
Channel Output Configuration Register – CHOCFR .....	353
Channel Control Register – CHCTR.....	354
Channel Polarity Configuration Register – CHPOLR.....	355
Timer Interrupt Control Register – DICTR .....	356
Timer Event Generator Register – EVGR.....	357
Timer Interrupt Status Register – INTSR.....	358
Timer Counter Register – CNTR.....	359
Timer Prescaler Register – PSCR .....	359
Timer Counter Reload Register – CRR .....	360
Channel Capture/Compare Register – CHCCR .....	361
<b>16 Basic Function Timer (BFTM) .....</b>	<b>362</b>
Introduction .....	362
Features.....	362
Functional Description .....	363
Repetitive Mode.....	363
One Shot Mode.....	364
Trigger ADC Start.....	364
Register Map .....	365
Register Descriptions.....	365
BFTM Control Register – BFTMCR .....	365
BFTM Status Register – BFTMSR.....	366
BFTM Counter Register – BFTMCNTR.....	367
BFTM Compare Value Register – BFTMCMPR .....	367
<b>17 Real Time Clock (RTC) .....</b>	<b>368</b>
Introduction .....	368
Features.....	368
Functional Descriptions .....	369
RTC Related Register Reset .....	369
Reading RTC Register.....	369

Low Speed Clock Configuration .....	369
RTC Counter Operation .....	370
Interrupt and Wakeup Control .....	370
RTCOUNT Output Pin Configuration .....	370
Register Map .....	372
Register Descriptions .....	372
RTC Counter Register – RTCCNT .....	372
RTC Compare Register – RTCCMP .....	373
RTC Control Register – RTCCR .....	374
RTC Status Register – RTCSR .....	376
RTC Interrupt and Wakeup Enable Register – RTCIWEN .....	377
<b>18 Watchdog Timer (WDT) .....</b>	<b>378</b>
Introduction .....	378
Features .....	379
Functional Description .....	379
Register Map .....	381
Register Descriptions .....	381
Watchdog Timer Control Register – WDTCR .....	381
Watchdog Timer Mode Register 0 – WDTMR0 .....	382
Watchdog Timer Mode Register 1 – WDTMR1 .....	383
Watchdog Timer Status Register – WDTSR .....	384
Watchdog Timer Protection Register – WDTPR .....	385
Watchdog Timer Clock Selection Register – WDTCSR .....	386
<b>19 Inter-Integrated Circuit (I<sup>2</sup>C) .....</b>	<b>387</b>
Introduction .....	387
Features .....	388
Functional Descriptions .....	388
Two-Wire Serial Interface .....	388
START and STOP Conditions .....	388
Data Validity .....	389
Addressing Format .....	389
Data Transfer and Acknowledge .....	391
Clock Synchronization .....	391
Arbitration .....	392
General Call Addressing .....	392
Bus Error .....	392
Address Mask Enable .....	393
Address Snoop .....	393
Operation Mode .....	393
Conditions of Holding SCL Line .....	398
I <sup>2</sup> C Timeout Function .....	398
PDMA Interface .....	399

Register Map .....	399
Register Descriptions.....	400
I <sup>2</sup> C Control Register – I2CCR .....	400
I <sup>2</sup> C Interrupt Enable Register – I2CIER .....	402
I <sup>2</sup> C Address Register – I2CADDR.....	403
I <sup>2</sup> C Status Register – I2CSR.....	404
I <sup>2</sup> C SCL High Period Generation Register – I2CSHPGR.....	407
I <sup>2</sup> C SCL Low Period Generation Register – I2CSLPGR .....	408
I <sup>2</sup> C Data Register – I2CDR .....	409
I <sup>2</sup> C Target Register – I2CTAR .....	410
I <sup>2</sup> C Address Mask Register – I2CADDMR .....	411
I <sup>2</sup> C Address Snoop Register – I2CADDSDR .....	412
I <sup>2</sup> C Timeout Register – I2CTOUT.....	413
<b>20 Serial Peripheral Interface (SPI) .....</b>	<b>414</b>
Introduction .....	414
Features.....	415
Functional Descriptions .....	415
Master Mode.....	415
Slave Mode.....	415
SPI Serial Frame Format.....	415
SPI Dual Mode.....	420
Status Flags.....	422
PDMA Interface.....	425
Register Map .....	425
Register Descriptions.....	426
SPI Control Register 0 – SPICR0 .....	426
SPI Control Register 1 – SPICR1 .....	428
SPI Interrupt Enable Register – SPIIER .....	430
SPI Clock Prescaler Register – SPICPR .....	431
SPI Data Register – SPIDR.....	431
SPI Status Register – SPISR.....	432
SPI FIFO Control Register – SPIFCR.....	433
SPI FIFO Status Register – SPIFSR .....	434
SPI FIFO Time Out Counter Register – SPIFTOCR.....	435
<b>21 Universal Synchronous Asynchronous Receiver Transmitter (USART) .....</b>	<b>436</b>
Introduction .....	436
Features.....	437
Functional Descriptions .....	437
Serial Data Format.....	437
Baud Rate Generation .....	438
Hardware Flow Control .....	440
IrDA.....	441

RS485 Mode.....	443
Synchronous Master Mode.....	445
Interrupts and Status .....	447
PDMA Interface.....	447
Register Map .....	447
Register Descriptions.....	448
USART Data Register – USRDR.....	448
USART Control Register – USRCR.....	449
USART FIFO Control Register – USRFCR.....	451
USART Interrupt Enable Register – USRIER .....	452
USART Status & Interrupt Flag Register – USRSIFR.....	454
USART Timing Parameter Register – USRTPR .....	456
USART IrDA Control Register – IrDACR .....	457
USART RS485 Control Register – RS485CR.....	458
USART Synchronous Control Register – SYNCR .....	459
USART Divider Latch Register – USRDLR.....	460
USART Test Register – USRTSTR.....	461
<b>22 Universal Asynchronous Receiver Transmitter (UART).....</b>	<b>462</b>
Introduction .....	462
Features.....	463
Functional Descriptions .....	463
Serial Data Format.....	463
Baud Rate Generation.....	464
Interrupts and Status .....	465
PDMA Interface.....	466
Register Map .....	466
Register Descriptions.....	466
UART Data Register – URDR.....	466
UART Control Register – URCR.....	467
UART Interrupt Enable Register – URIER .....	468
UART Status & Interrupt Flag Register – URSIFR .....	470
UART Divider Latch Register – URDLR .....	471
UART Test Register – URTSTR.....	472
<b>23 Smart Card Interface (SCI) .....</b>	<b>473</b>
Introduction .....	473
Features.....	474
Functional Descriptions .....	474
Elementary Time Unit Counter.....	474
Guard Time Counter .....	476
Waiting Time Counter .....	477
Card Clock and Data Selection.....	478
Card Detection .....	478

SCI Data Transfer Mode .....	479
Interrupt Generator .....	481
PDMA Interface .....	482
Register Map .....	482
Register Descriptions .....	483
SCI Control Register – CR .....	483
SCI Status Register – SR .....	485
SCI Contact Control Register – CCR .....	486
SCI Elementary Time Unit Register – ETUR .....	487
SCI Guard Time Register – GTR .....	488
SCI Waiting Time Register – WTR .....	489
SCI Interrupt Enable Register – IER .....	490
SCI Interrupt Pending Register – IPR .....	491
SCI Transmit Buffer – TXB .....	493
SCI Receive Buffer – RXB .....	493
SCI Prescaler Register – PSCR .....	494
<b>24 USB Device Controller (USB) .....</b>	<b>495</b>
Introduction .....	495
Features .....	495
Functional Descriptions .....	496
Endpoints .....	496
EP_SRAM .....	496
Serial Interface Engine – SIE .....	497
Double-Buffering .....	497
Suspend Mode and Wake-up .....	498
Remote Wake-up .....	499
Register Map .....	499
Register Descriptions .....	500
USB Control and Status Register – USBCSR .....	500
USB Interrupt Enable Register – USBIER .....	502
USB Interrupt Status Register – USBISR .....	503
USB Frame Count Register – USBFCR .....	504
USB Device Address Register – USBDEVAR .....	505
USB Endpoint 0 Control and Status Register – USBEP0CSR .....	506
USB Endpoint 0 Interrupt Enable Register – USBEP0IER .....	507
USB Endpoint 0 Interrupt Status Register – USBEP0ISR .....	509
USB Endpoint 0 Transfer Count Register – USBEP0TCR .....	510
USB Endpoint 0 Configuration Register – USBEP0CFGR .....	511
USB Endpoint 1 ~ 3 Control and Status Register – USBEPnCSR, n = 1 ~ 3 .....	512
USB Endpoint 1 ~ 3 Interrupt Enable Register – USBEPnIER, n = 1 ~ 3 .....	513
USB Endpoint 1 ~ 3 Interrupt Status Register – USBEPnISR, n = 1 ~ 3 .....	514
USB Endpoint 1 ~ 3 Transfer Count Register – USBEPnTCR, n = 1 ~ 3 .....	515
USB Endpoint 1 ~ 3 Configuration Register – USBEPnCFGR, n = 1 ~ 3 .....	516

USB Endpoint 4 ~ 7 Control and Status Register – USBEPnCSR, n = 4 ~ 7 .....	517
USB Endpoint 4 ~ 7 Interrupt Enable Register – USBEPnIER, n = 4 ~ 7 .....	519
USB Endpoint 4 ~ 7 Interrupt Status Register – USBEPnISR, n = 4 ~ 7 .....	520
USB Endpoint 4 ~ 7 Transfer Count Register – USBEPnTCR, n = 4 ~ 7 .....	521
USB Endpoint 4 ~ 7 Configuration Register – USBEPnCFGR, n = 4 ~ 7 .....	522
<b>25 Peripheral Direct Memory Access (PDMA) .....</b>	<b>523</b>
Introduction .....	523
Features .....	523
Functional Description .....	524
AHB Master .....	524
PDMA Channel .....	524
PDMA Request Mapping .....	524
Channel Transfer .....	525
Channel Priority .....	525
Transfer Request .....	526
Address Mode .....	526
Auto-Reload .....	527
Transfer Interrupt .....	527
Register Map .....	528
Register Descriptions .....	529
PDMA Channel n Control Register – PDMACHnCR (n = 0 ~ 5) .....	529
PDMA Channel n Source Address Register – PDMACHnSADR (n = 0 ~ 5) .....	531
PDMA Channel n Destination Address Register – PDMACHnDADR (n = 0 ~ 5) .....	531
PDMA Channel n Transfer Size Register – PDMACHnTSR (n = 0 ~ 5) .....	532
PDMA Channel n Current Transfer Size Register – PDMACHnCTSR (n = 0 ~ 5) .....	533
PDMA Interrupt Status Register – PDMAISR .....	534
PDMA Interrupt Status Clear Register – PDMAISCR .....	535
PDMA Interrupt Enable Register – PDMAIER .....	536
<b>26 External Bus Interface (EBI) .....</b>	<b>538</b>
Introduction .....	538
Features .....	538
Function Descriptions .....	539
Non-Multiplexed 8-Bit Data 8-Bit Address Mode .....	540
Non-Multiplexed 16-Bit Data N-bit Address Mode .....	541
Multiplexed 16-Bit Data, 16-Bit Address Mode .....	542
Multiplexed 8-Bit Data, 20-Bit Address Mode .....	543
Write Buffer and EBI Status .....	544
Bus Turn-around and IDLE Cycles .....	544
AHB Transaction Width Conversion .....	545
EBI Bank Access .....	547
PDMA Request .....	547
Register Map .....	548

Register Descriptions.....	548
EBI Control Register – EBICR.....	548
EBI Status Register – EBISR.....	550
EBI Address Timing Register – EBIATR.....	551
EBI Read Timing Register – EBIRTR.....	552
EBI Write Timing Register – EBIWTR.....	553
EBI Polarity Register – EBIPR.....	554
<b>27 Cyclic Redundancy Check (CRC).....</b>	<b>555</b>
Introduction .....	555
Features.....	555
Function Descriptions .....	556
CRC Computation.....	556
Byte and Bit Reversal for CRC Computation.....	556
CRC with PDMA .....	557
Register Map .....	557
Register Descriptions.....	558
CRC Control Register – CRCCR.....	558
CRC Seed Register – CRCSDR.....	559
CRC Checksum Register – CRCCSR.....	559
CRC Data Register – CRCDR.....	560
<b>28 AES Encrypt/Decrypt Interface (AES).....</b>	<b>561</b>
Introduction .....	561
Features.....	561
Functional Descriptions .....	562
AES Mode Description.....	562
AES Status .....	564
AES PDMA Interface .....	564
AES Interrupt .....	565
AES Initial Vector .....	565
AES Word Swap .....	566
Register Map .....	566
Register Descriptions.....	567
AES Control Register – AESCR .....	567
AES Status Register – AESSR.....	568
AES DMA Register – AESDMAR.....	569
AES Interrupt Status Register – AESISR.....	570
AES Interrupt Enable Register – AESIER.....	571
AES DATA Input Register – AESDINR.....	572
AES DATA Output Register – AESDOUTR.....	572
AES Key Register n – AESKEYRn, n = 0 ~ 7 .....	573
AES Initial Vector Register n – AESIVRn, n = 0 ~ 3 .....	573



## List of Tables

Table 1. Features and Peripheral List .....	29
Table 2. Document Conventions .....	31
Table 3. Register Map .....	36
Table 4. Flash Memory and Option Byte .....	41
Table 5. Relationship Between Wait State Cycle and HCLK .....	41
Table 6. Booting Modes .....	42
Table 7. Option Byte Memory Map .....	46
Table 8. Access Permission of Protected Main Flash Page .....	47
Table 9. Access Permission When Security Protection is Enabled .....	48
Table 10. FMC Register Map .....	49
Table 11. Operation Mode Definitions .....	70
Table 12. Enter/Exit Power Saving Modes .....	70
Table 13. Power Status after System Reset .....	71
Table 14. PWRCU Register Map .....	72
Table 15. Output Divider 2 Value Mapping .....	85
Table 16. Feedback Divider 2 Value Mapping .....	85
Table 17. USB PLL Output Divider 2 Value Mapping .....	87
Table 18. USB PLL Feedback Divider 2 Value Mapping .....	87
Table 19. CKOUT Clock Source .....	90
Table 20. CKCU Register Map .....	91
Table 21. RSTCU Register Map .....	119
Table 22. AFIO, GPIO and I/O Pad Control Signal True Table .....	127
Table 23. GPIO Register Map .....	128
Table 24. AFIO Selection for Peripheral Map Example .....	178
Table 25. AFIO Register Map .....	178
Table 26. Exception types .....	183
Table 27. NVIC Register Map .....	186
Table 28. EXTI Register Map .....	192
Table 29. Data Format in ADCDR [15:0] .....	204
Table 30. A/D Converter Register Map .....	206
Table 31. Counting Direction and Encoding Signals .....	239
Table 32. Compare Match Output Setup .....	240
Table 33. GPTM Register Map .....	250
Table 34. GPTM Internal Trigger Connection .....	255
Table 35. Compare Match Output Setup .....	298
Table 36. PWM Register Map .....	308
Table 37. PWM Internal Trigger Connection .....	313
Table 38. Compare Match Output Setup .....	343
Table 39. SCTM Register Map .....	346

Table 40. BFTM Register Map .....	365
Table 41. LSE Startup Mode Operating Current and Startup Time.....	369
Table 42. RTCOUT Output Mode and Active Level Setting .....	371
Table 43. RTC Register Map.....	372
Table 44. Watchdog Timer Register Map.....	381
Table 45. Conditions of Holding SCL line.....	398
Table 46. I <sup>2</sup> C Register Map .....	399
Table 47. I <sup>2</sup> C Clock Setting Example .....	408
Table 48. SPI Interface Format Setup.....	416
Table 49. SPI Mode Fault Trigger Conditions .....	424
Table 50. SPI Master Mode SPI_SEL Pin Status.....	424
Table 51. SPI Register Map .....	425
Table 52. Baud Rate Deviation Error Calculation – CK_USART = 40 MHz.....	439
Table 53. Baud Rate Deviation Error Calculation – CK_USART = 48 MHz.....	439
Table 54. Baud Rate Deviation Error Calculation – CK_USART = 60 MHz.....	439
Table 55. USART Register Map.....	447
Table 56. Baud Rate Deviation Error Calculation – CK_UART = 40 MHz .....	464
Table 57. Baud Rate Deviation Error Calculation – CK_UART = 48 MHz .....	465
Table 58. Baud Rate Deviation Error Calculation – CK_UART = 60 MHz .....	465
Table 59. UART Register Map .....	466
Table 60. DI Field Based Di Encoded Decimal Values .....	475
Table 61. FI Field Based Fi Encoded Decimal Values .....	475
Table 62. Possible ETU Values Obtained with the Fi/Di Ratio.....	475
Table 63. SCI Register Map .....	482
Table 64. Endpoint Characteristics .....	496
Table 65. USB Data Types and Buffer Size .....	496
Table 66. USB Register Map .....	499
Table 67. Resume Event Detection .....	501
Table 68. PDMA Channel Assignments .....	525
Table 69. PDMA Address Modes .....	526
Table 70. PDMA Register Map.....	528
Table 71. EBI Maps AHB Transactions Width to External Device Transactions .....	546
Table 72. EBI Maps AHB Transactions Width to External Device Transactions Width .....	546
Table 73. EBI Register Map .....	548
Table 74. CRC Register Map .....	557
Table 75. AES Register Map.....	566

## List of Figures

Figure 1. Block Diagram .....	30
Figure 2. Cortex®-M3 Block Diagram.....	33
Figure 3. Bus Architecture .....	34
Figure 4. Memory Map.....	35
Figure 5. Flash Memory Controller Block Diagram.....	39
Figure 6. Flash Memory Map.....	40
Figure 7. Vector Remapping .....	42
Figure 8. Page Erase Operation Flowchart .....	43
Figure 9. Mass Erase Operation Flowchart .....	44
Figure 10. Word Programming Operation Flowchart .....	45
Figure 11. PWRCU Block Diagram .....	66
Figure 12. Power-On Reset / Power-Down Reset Waveform.....	68
Figure 13. CKCU Block Diagram .....	80
Figure 14. External Crystal, Ceramic and Resonators for HSE.....	81
Figure 15. HSI Auto Trimming Block Diagram .....	83
Figure 16. PLL Block Diagram .....	84
Figure 17. USB PLL Block Diagram.....	86
Figure 18. External Crystal, Ceramic and Resonators for LSE .....	88
Figure 19. RSTCU Block Diagram.....	117
Figure 20. Power On Reset Sequence .....	118
Figure 21. GPIO Block Diagram .....	125
Figure 22. AFIO / GPIO Control Signal.....	127
Figure 23. AFIO Block Diagram .....	176
Figure 24. EXTI Channel Input Selection .....	177
Figure 25. EXTI Block Diagram .....	188
Figure 26. EXTI Wake-Up Event Management.....	189
Figure 27. Wake-Up Interrupt Service Routine .....	190
Figure 28. EXTI Interrupt Debounce Function .....	191
Figure 29. ADC with V <sub>REF</sub> Block Diagram.....	198
Figure 30. One Shot Conversion Mode .....	201
Figure 31. Continuous Conversion Mode .....	201
Figure 32. Discontinuous Conversion Mode.....	203
Figure 33. Voltage Reference Generator Block Diagram .....	206
Figure 34. GPTM Block Diagram.....	225
Figure 35. Up-counting Example .....	227
Figure 36. Down-counting Example.....	227
Figure 37. Center-aligned Counting Example.....	228
Figure 38. GPTM Clock Source Selection .....	229
Figure 39. Trigger Control Block.....	230

Figure 40. Slave Controller Diagram .....	231
Figure 41. GPTM in Restart Mode .....	231
Figure 42. GPTM in Pause Mode .....	232
Figure 43. GPTM in Trigger Mode .....	232
Figure 44. Master GPTMn and Slave GPTMm/TMm Connection .....	233
Figure 45. MTO Selection .....	233
Figure 46. Capture/Compare Block Diagram.....	234
Figure 47. Input Capture Mode.....	235
Figure 48. PWM Pulse Width Measurement Example.....	236
Figure 49. Channel 0 and Channel 1 Input Stages.....	236
Figure 50. Channel 2 and Channel 3 Input Stages.....	237
Figure 51. TI0 Digital Filter Diagram with N = 2.....	237
Figure 52. Input Stage and Quadrature Decoder Block Diagram .....	238
Figure 53. Both TI0 and TI1 Quadrature Decoder Counting.....	239
Figure 54. Output Stage Block Diagram .....	239
Figure 55. Toggle Mode Channel Output Reference Signal (CHxPRE = 0) .....	240
Figure 56. Toggle Mode Channel Output Reference Signal (CHxPRE = 1) .....	241
Figure 57. PWM Mode Channel Output Reference Signal and Counter in Up-counting Mode .....	241
Figure 58. PWM Mode Channel Output Reference Signal and Counter in Down-counting Mode .....	242
Figure 59. PWM Mode Channel Output Reference Signal and Counter in Centre-aligned Mode .....	242
Figure 60. Update Event Setting Diagram .....	243
Figure 61. Single Pulse Mode.....	244
Figure 62. Immediate Active Mode Minimum Delay .....	245
Figure 63. Asymmetric PWM Mode versus Center-Aligned Counting Mode .....	246
Figure 64. Pausing PWM0 using the GPTM CH0OREF Signal.....	247
Figure 65. Triggering PWM0 with GPTM Update Event .....	247
Figure 66. Trigger GPTM and PWM0 with the GPTM CH0 Input .....	248
Figure 67. GPTM PDMA Mapping Diagram.....	249
Figure 68. PWM Block Diagram .....	288
Figure 69. Up-counting Example .....	289
Figure 70. Down-counting Example.....	290
Figure 71. Center-aligned Counting Example.....	291
Figure 72. PWM Clock Selection Source.....	292
Figure 73. Trigger Control Block.....	293
Figure 74. Slave Controller Diagram .....	294
Figure 75. PWM in Restart Mode .....	294
Figure 76. PWM in Pause Mode.....	295
Figure 77. PWM in Trigger Mode.....	295
Figure 78. Master PWMn and Slave PWMm/TMm Connection.....	296
Figure 79. MTO Selection .....	296
Figure 80. Compare Block Diagram .....	297

Figure 81. Output Stage Block Diagram .....	297
Figure 82. Toggle Mode Channel Output Reference Signal (CHxPRE = 0) .....	298
Figure 83. Toggle Mode Channel Output Reference Signal (CHxPRE = 1) .....	299
Figure 84. PWM Mode Channel Output Reference Signal and Counter in Up-counting Mode .....	299
Figure 85. PWM Mode Channel Output Reference Signal and Counter in Down-counting Mode .....	300
Figure 86. PWM Mode Channel Output Reference Signal and Counter in Center-aligned Mode .....	300
Figure 87. Update Event Setting Diagram .....	301
Figure 88. Single Pulse Mode .....	302
Figure 89. Immediate Active Mode Minimum Delay .....	303
Figure 90. Asymmetric PWM Mode versus Center-aligned Counting Mode .....	304
Figure 91. Pausing PWM1 using the PWM0 CH0OREF Signal .....	305
Figure 92. Triggering PWM1 with PWM0 Update Event .....	305
Figure 93. Trigger PWM0 and PWM1 with the PWM0 Timer Enable Signal .....	306
Figure 94. PWM PDMA Mapping Diagram .....	307
Figure 95. SCTM Block Diagram .....	335
Figure 96. Up-Counting Example .....	336
Figure 97. SCTM Clock Source Selection .....	337
Figure 98. Trigger Controller Block .....	338
Figure 99. Slave Controller Diagram .....	339
Figure 100. SCTM in Restart Mode .....	339
Figure 101. SCTM in Pause Mode .....	340
Figure 102. SCTM in Trigger Mode .....	340
Figure 103. Capture/Compare Block Diagram .....	341
Figure 104. Input Capture Mode .....	341
Figure 105. Channel Input Stages .....	342
Figure 106. TI Digital Filter Diagram with N = 2 .....	342
Figure 107. Output Stage Block Diagram .....	343
Figure 108. Toggle Mode Channel Output Reference Signal (CHPRE = 0) .....	344
Figure 109. Toggle Mode Channel Output Reference Signal (CHPRE = 1) .....	344
Figure 110. PWM Mode Channel Output Reference Signal .....	345
Figure 111. Update Event Setting Diagram .....	346
Figure 112. BFTM Block Diagram .....	362
Figure 113. BFTM – Repetitive Mode .....	363
Figure 114. BFTM – One Shot Mode .....	364
Figure 115. BFTM – One Shot Mode Counter Updating .....	364
Figure 116. RTC Block Diagram .....	368
Figure 117. Watchdog Timer Block Diagram .....	378
Figure 118. Watchdog Timer Behavior .....	380
Figure 119. I <sup>2</sup> C Module Block Diagram .....	387
Figure 120. START and STOP Condition .....	389
Figure 121. Data Validity .....	389

Figure 122. 7-Bit Addressing Mode .....	390
Figure 123. 10-Bit Addressing Write Transmit Mode .....	390
Figure 124. 10-Bit Addressing Read Receive Mode .....	390
Figure 125. I <sup>2</sup> C Bus Acknowledge .....	391
Figure 126. Clock Synchronization During Arbitration .....	391
Figure 127. Two Master Arbitration Procedure .....	392
Figure 128. Master Transmitter Timing Diagram .....	394
Figure 129. Master Receiver Timing Diagram .....	395
Figure 130. Slave Transmitter Timing Diagram .....	396
Figure 131. Slave Receiver Timing Diagram .....	397
Figure 132. SCL Timing Diagram.....	408
Figure 133. SPI Block Diagram .....	414
Figure 134. SPI Single Byte Transfer Timing Diagram – CPOL = 0, CPHA = 0.....	416
Figure 135. SPI Continuous Data Transfer Timing Diagram – CPOL = 0, CPHA = 0.....	416
Figure 136. SPI Single Byte Transfer Timing Diagram – CPOL = 0, CPHA = 1.....	417
Figure 137. SPI Continuous Transfer Timing Diagram – CPOL = 0, CPHA = 1.....	417
Figure 138. SPI Single Byte Transfer Timing Diagram – CPOL = 1, CPHA = 0.....	418
Figure 139. SPI Continuous Transfer Timing Diagram – CPOL = 1, CPHA = 0.....	418
Figure 140. SPI Single Byte Transfer Timing Diagram – CPOL = 1, CPHA = 1.....	419
Figure 141. SPI Continuous Transfer Timing Diagram – CPOL = 1, CPHA = 1.....	419
Figure 142. SPI Dual Mode Bit Sequence – CPOL = 0, CPHA = 0, DFL = 0x8 (16-bit), MSB Transmitted First.....	420
Figure 143. SPI Dual Mode Bit Sequence – CPOL = 0, CPHA = 1, DFL = 0x8 (16-bit), MSB Transmitted First.....	420
Figure 144. SPI Dual Mode Bit Sequence – CPOL = 1, CPHA = 0, DFL = 0x8 (16-bit) , MSB Transmitted First.....	421
Figure 145. SPI Dual Mode Bit Sequence – CPOL = 1, CPHA = 1, DFL = 0x8 (16-bit), MSB Transmitted First.....	421
Figure 146. SPI Dual Mode Data Read Example – CPOL = 1, CPHA = 1.....	422
Figure 147. SPI Multi-Master Slave Environment.....	423
Figure 148. USART Block Diagram .....	436
Figure 149. USART Serial Data Format .....	438
Figure 150. USART Clock CK_USART and Data Frame Timing .....	438
Figure 151. Hardware Flow Control between 2 USARTs.....	440
Figure 152. USART RTS Flow Control .....	440
Figure 153. USART CTS Flow Control .....	441
Figure 154. IrDA Modulation and Demodulation.....	441
Figure 155. USART I/O and IrDA Block Diagram .....	443
Figure 156. RS485 Interface and Waveform .....	444
Figure 157. USART Synchronous Transmission Example .....	445
Figure 158. 8-Bit Format USART Synchronous Waveform.....	446
Figure 159. UART Block Diagram.....	462

Figure 160. UART Serial Data Format.....	463
Figure 161. UART Clock CK_UART and Data Frame Timing.....	464
Figure 162. SCI Block Diagram .....	473
Figure 163. Character Frame and Compensation Mode .....	476
Figure 164. Guard Time Duration .....	477
Figure 165. Character and Block Waiting Time Duration – CWT and BWT.....	478
Figure 166. SCI Card Detection Diagram .....	479
Figure 167. SCI Interrupt Structure.....	481
Figure 168. USB Block Diagram.....	495
Figure 169. Endpoint Buffer Allocation Example.....	497
Figure 170. Double-buffering Operation Example .....	498
Figure 171. PDMA Block Diagram .....	523
Figure 172. PDMA Request Mapping Architecture .....	524
Figure 173. PDMA Channel Arbitration and Scheduling Example .....	526
Figure 174. EBI Block Diagram .....	539
Figure 175. EBI Non-Multiplexed 8-Bit Data, 8-Bit Address Read Operation .....	540
Figure 176. EBI Non-Multiplexed 8-Bit Data, 8-Bit Address Write Operation .....	540
Figure 177. EBI Non-Multiplexed 16-Bit Data, N-Bit Address Read Operation .....	541
Figure 178. EBI Non-Multiplexed 16-Bit Data, N-Bit Address Write Operation.....	541
Figure 179. An EBI Address Latch Setup Diagram.....	542
Figure 180. EBI Multiplexed 16-Bit Data, 16-Bit Address Read Operation.....	542
Figure 181. EBI Multiplexed 16-Bit Data, 16-Bit Address Write Operation .....	543
Figure 182. EBI Multiplexed 8-Bit Data, 20-Bit Address Read Operation.....	543
Figure 183. EBI Multiplexed 8-Bit Data, 20-Bit Address Write Operation .....	544
Figure 184. EBI Inserts an IDLE Cycle Between Transactions in the Same Bank (NOIDLE = 0) .....	545
Figure 185. EBI De-asserts an IDLE Cycle Between Transactions in the Same Bank (NOIDLE = 1)..	545
Figure 186. EBI Bank Memory Map.....	547
Figure 187. CRC Block Diagram .....	555
Figure 188. CRC Data Bit and Byte Reversal Example.....	556
Figure 189. AES Block Diagram .....	561
Figure 190. AES-ECB Mode .....	562
Figure 191. AES-CBC Mode.....	563
Figure 192. AES-CTR Mode .....	564
Figure 193. AES Interrupt .....	565
Figure 194. Initial Vector for CTR Mode .....	565
Figure 195. AES Word Swap Function .....	566

# 1 Introduction

## Overview

This user manual provides detailed information including how to use the HT32F12364 device, system and bus architecture, memory organization and peripheral instructions. The target audiences for this document are software developers, application developers and hardware developers. For more information regarding pin assignment, package and electrical characteristics, please refer to the HT32F12364 datasheet.

The device is a high performance and low power consumption 32-bit microcontroller 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 advanced debug support.

The device operates at a frequency of up to 72 MHz with a Flash accelerator to obtain maximum efficiency. It provides 256 KB of embedded Flash memory for code/data storage and up to 128 KB of embedded SRAM memory for system operation and application program usage. A variety of peripherals, such as ADC, I<sup>2</sup>C, USART, UART, SPI, SCI, PDMA, GPTM, PWM, SCTM, EBI, CRC-16/32, AES-128/256, USB2.0 FS and SW-DP (Serial Wire Debug Port), etc., are also implemented in the device. Several power saving modes provide the flexibility for maximum optimization between wakeup latency and power consumption, an especially important consideration in low power applications.

The above features ensure that the device is 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, fingerprint recognition smart door lock and so on.



## Features

- Core
  - 32-bit Arm® Cortex®-M3 processor core
  - Up to 72 MHz operation frequency
  - Single-cycle multiplication and hardware division
  - Integrated Nested Vectored Interrupt Controller (NVIC)
  - 24-bit SysTick timer
- On-Chip Memory
  - 256 KB on-chip Flash memory for instruction/data and option storage
  - 128 KB on-chip SRAM
  - Supports multiple booting modes
- Flash Memory Controller – FMC
  - Flash accelerator to obtain maximum efficiency
  - 32-bit word programming with In System Programming Interface (ISP) and In Application Programming (IAP)
  - Flash protection capability to prevent illegal access
- Reset Control Unit – RSTCU
  - Supply supervisor:
    - ◆ Power-On Reset / Power-Down Reset – POR / PDR
    - ◆ Brown-Out Detector – BOD
    - ◆ Programmable Low Voltage Detector – LVD
- Clock Control Unit – CKCU
  - External 4 to 16 MHz crystal oscillator
  - External 32.768 kHz crystal oscillator
  - Internal 8 MHz RC oscillator trimmed to  $\pm 1.5\%$  accuracy at 3.3 V operating voltage and 25 °C operating temperature
  - Internal 32 kHz RC oscillator
  - Integrated system clock PLL and USB PLL
  - Independent clock divider and gating bits for peripheral clock sources
- Power Management – PWRCU
  - Single  $V_{DD}$  power supply: 1.65 V to 3.6 V
  - Integrated 1.5 V LDO regulator for MCU core, peripherals and memories power supply
  - $V_{DD}$  power supply for RTC
  - Two power domains:  $V_{DD}$  and  $V_{CORE}$
  - 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 edges
  - Individual interrupt enable, wakeup enable and status bits for each EXTI line
  - Software interrupt trigger mode for each EXTI line
  - Integrated deglitch filter for short pulse blocking

- Analog to Digital Converter – ADC
  - 12-bit SAR ADC engine
  - Up to 1 MSPS conversion rate
  - Up to 8 external analog input channels
- I/O Ports – GPIO
  - Up to 52 GPIOs
  - Port A, B, C, D, F are mapped on 16 external interrupts (EXTI)
  - Almost all I/O pins have configurable output driving current
- General-Purpose Timer – GPTM
  - 16-bit up/down auto-reload counter
  - Up to 4 independent channels for each timer
  - 16-bit programmable prescaler allowing dividing the counter clock frequency by any factor between 1 and 65536
  - Input Capture function
  - Compare Match Output
  - PWM waveform generation with Edge-aligned and Center-aligned Counting Modes
  - Single Pulse Mode Output
  - Encoder interface controller with two inputs using quadrature decoder
- Pulse-Width-Modulation Timer – PWM
  - 16-bit up/down auto-reload counter
  - Up to 4 independent channels for each timer
  - 16-bit programmable prescaler allowing counter clock frequency division by any factor between 1 and 65536
  - Compare Match Output
  - PWM waveform generation with Edge-aligned and Center-aligned Counting Modes
  - Single Pulse Mode Output
- Single-Channel Timer – SCTM
  - 16-bit auto-reload up-counter
  - One channel for each timer
  - 16-bit programmable prescaler allowing counter clock frequency division by any factor between 1 and 65536
  - Input Capture function
  - Compare Match Output
  - PWM waveform generation with Edge-aligned
  - Single Pulse Mode Output
- Basic Function Timer – BFTM
  - 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 a 3-bit prescaler
  - Reset event for the system
  - Programmable watchdog timer window function
  - Register write protection function

- Real Time Clock – RTC
  - 24-bit up-counter with a programmable prescaler
  - Alarm function
  - Interrupt and Wake-up event
- Inter-integrated Circuit – I<sup>2</sup>C
  - Supports both master and slave modes with a frequency of up to 1 MHz
  - Provides an arbitration function and clock synchronization
  - Supports 7-bit and 10-bit addressing modes and general call addressing
  - Supports slave multi-addressing mode with maskable address
- Serial Peripheral Interface – SPI
  - Supports both master and slave mode
  - Frequency of up to ( $f_{PCLK}/2$ ) MHz for master mode and ( $f_{PCLK}/3$ ) MHz for slave mode
  - FIFO Depth: 8 levels
  - Multi-master and multi-slave operation
- Universal Synchronous Asynchronous Receiver Transmitter – USART
  - Supports both asynchronous and clocked synchronous serial communication modes
  - Asynchronous operating baud rate clock frequency of up to ( $f_{PCLK}/16$ ) MHz and synchronous operating baud rate clock frequency of up to ( $f_{PCLK}/8$ ) MHz
  - Capability of full duplex communication
  - Fully programmable serial communication characteristics including word length, parity bit, stop bit and bit order
  - Error detection: Parity, overrun and frame error
  - Supports Auto hardware flow control mode – RTS, CTS
  - IrDA SIR encoder and decoder
  - RS485 mode with output enable control
  - FIFO Depth: 8-level for both receiver and transmitter
- Universal Asynchronous Receiver Transmitter – UART
  - Asynchronous serial communication operating baud rate clock frequency of up to ( $f_{PCLK}/16$ ) MHz
  - Capability of full duplex communication
  - Fully programmable serial communication characteristics including word length, parity bit, stop bit and bit order
  - Error detection: Parity, overrun and frame error
- Smart Card Interface – SCI
  - Supports ISO 7816-3 Standard
  - Character Transfer 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 check functions
  - Automatic character retry on parity error detection in transmission and reception modes

- Cyclic Redundancy Check – CRC
  - Supports CRC16 polynomial:  $0x8005, X^{16} + X^{15} + X^2 + 1$
  - Supports CCITT CRC16 polynomial:  $0x1021, X^{16} + X^{12} + X^5 + 1$
  - Supports IEEE-802.3 CRC32 polynomial:  $0x04C11DB7, X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
  - Supports 1's complement, byte reverse and bit reverse operation on data and checksum
  - Supports byte, half-word and word data size
  - Programmable CRC initial seed value
  - CRC computation executed in 1 AHB clock cycle for 8-bit data and 4 AHB clock cycles for 32-bit data
  - Supports PDMA to complete a CRC computation of a block of memory
- Peripheral Direct Memory Access – PDMA
  - 6 channels with trigger source grouping
  - 8/16/32-bit width data transfer
  - Supports linear address, circular address and fixed address mode
  - 4-level programmable channel priority
  - Auto reload mode
  - Supports trigger sources:  
ADC, SPI, USART, UART, I<sup>2</sup>C, SCI, GPTM, PWM, AES and software request
- External Bus Interface – EBI
  - Programmable interface for various memory types
  - Translates the AHB transactions into the appropriate external device protocol
  - Individual chip select control for each memory bank
  - Programmable timing to support a wide range of devices
  - Automatic translation when AHB transaction width and external memory interface width is different
  - Write buffer to decrease the stalling of the AHB write burst transaction
  - Both multiplexed and non-multiplexed address and data line configurations
    - ◆ Up to 21 address lines
    - ◆ Up to 16-bit data bus width
- Universal Serial Bus Device Controller – USB
  - Complies with USB 2.0 full-speed (12 Mbps) 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
  - 1024-byte EP\_SRAM used as the endpoint data buffers
- Advanced Encryption Standard – AES
  - Supports AES Encrypt / Decrypt Function
  - Supports AES ECB/CBC/CTR mode
  - Supports Key Sizes of 128, 192 and 256 bits
  - Supports 4 words Initial Vector for CBC and CTR mode
  - $8 \times 32$  bits ( Each IN and OUT FIFO Capacity ) for 2 AES Data blocks
  - Supports Word Data Swap Function
  - Supports PDMA Interface

- Debug Support
  - Serial Wire Debug Port SW-DP
  - 6 instruction comparators and 2 literal comparators for hardware breakpoint or code / literal patches
  - 4 comparators for hardware watchpoints
  - 1-bit asynchronous trace for serial wire debug mode – TRACESWO
- Package and Operation Temperature
  - 40-pin QFN, 48/64-pin LQFP packages
  - Operation temperature range: -40 °C to 85 °C

## Device Information

**Table 1. Features and Peripheral List**

Peripherals		HT32F12364
Main Flash (KB)		255
Option Bytes Flash (KB)		1
SRAM (KB)		128
Timers	GPTM	1
	PWM	1
	SCTM	2
	BFTM	2
	RTC	1
	WDT	1
Communication	USB	1
	USART	1
	UART	2
	SPI	2
	I <sup>2</sup> C	2
	SCI	1
PDMA		6 channels
AES		1
EBI		1
CRC		1
GPIO		Up to 52
EXTI		16
12-bit ADC		1
Number of channels		Max. 8 Channels
CPU Frequency		Up to 72 MHz
Operating Voltage		1.65 V ~ 3.6 V
Operating Temperature		-40 °C ~ 85 °C
Package		40-pin QFN and 48 / 64-pin LQFP

## Block Diagram

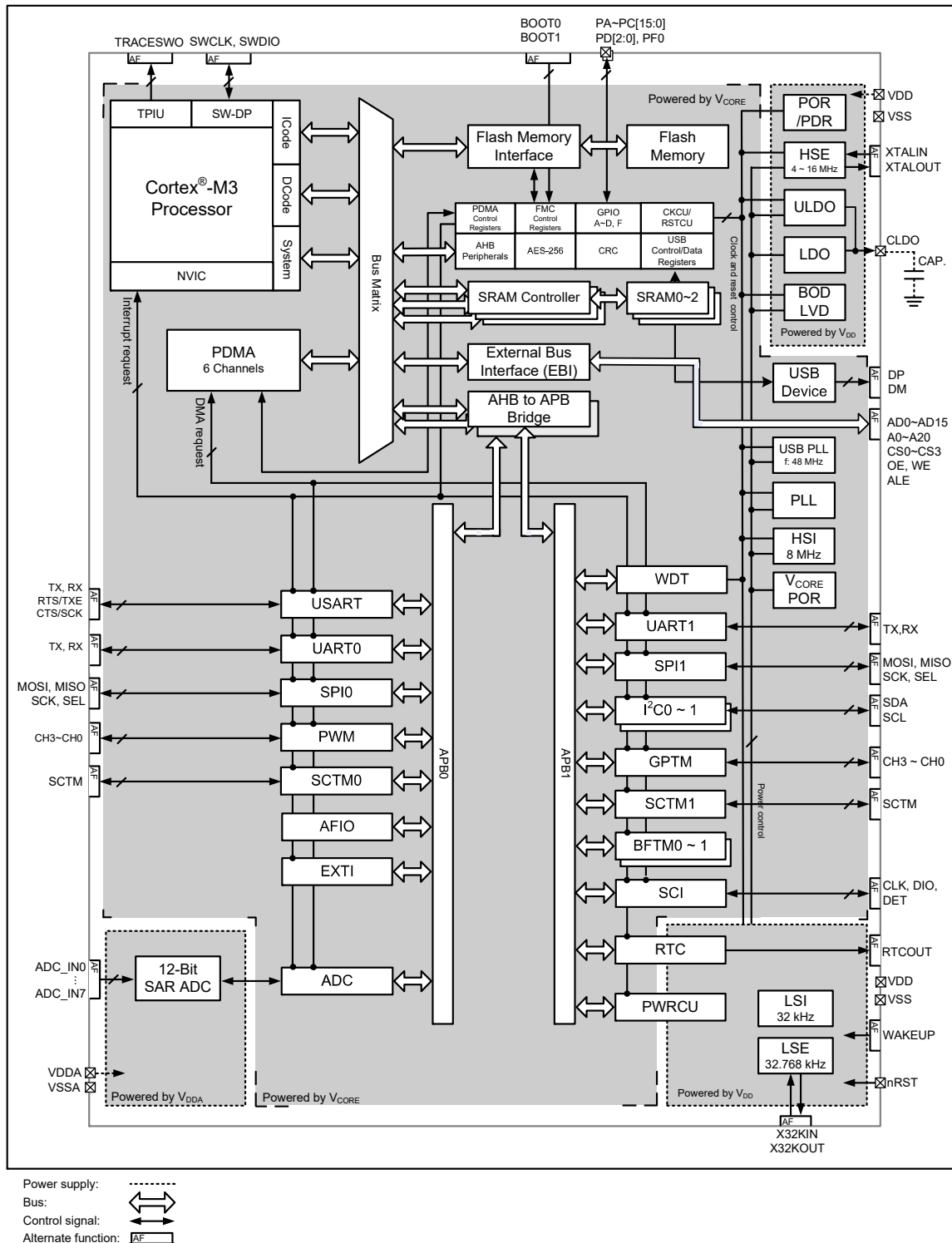


Figure 1. Block Diagram

## 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	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <b>19</b>  <div style="border: 1px solid black; padding: 2px;">SERDYIE</div> RW </div> <div style="text-align: center;"> <b>18</b>  <div style="border: 1px solid black; padding: 2px;">PLLRDYIE</div> 0 RW 0 </div> </div>	Software can read and write to this bit.
RO	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <b>3</b>  <div style="border: 1px solid black; padding: 2px;">HSIRDY</div> RO </div> <div style="text-align: center;"> <b>2</b>  <div style="border: 1px solid black; padding: 2px;">HSERDY</div> 1 RO 0 </div> </div>	Software can only read this bit. Write operation has no effort.
RC	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <b>1</b>  <div style="border: 1px solid black; padding: 2px;">PDF</div> RC </div> <div style="text-align: center;"> <b>0</b>  <div style="border: 1px solid black; padding: 2px;">BAK_PORF</div> 0 RC 1 </div> </div>	Software can read this bit. Read operation clears it to 0 automatically.
WC	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <b>3</b>  <div style="border: 1px solid black; padding: 2px;">SERDYF</div> WC </div> <div style="text-align: center;"> <b>2</b>  <div style="border: 1px solid black; padding: 2px;">PLLRDYF</div> 0 WC 0 </div> </div>	Software can read this bit or clear it by writing 1. Write 0 to it has no effort.
W0C	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <b>1</b>  <div style="border: 1px solid black; padding: 2px;"></div> W0C </div> <div style="text-align: center;"> <b>0</b>  <div style="border: 1px solid black; padding: 2px;">MIF</div> 0 </div> </div>	Software can read this bit or clear it by writing 0. Writing 1 to it will have no effect.
WO	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <b>31</b>  <div style="border: 1px solid black; padding: 2px;">DB_CKSRC</div> WO </div> <div style="text-align: center;"> <b>30</b>  <div style="border: 1px solid black; padding: 2px;"></div> 0 WO 0 </div> </div>	Software can only write to this bit. Read operation always returns 0.
Reserved	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <b>1</b>  <div style="border: 1px solid black; padding: 2px;">LLRDY</div> RO </div> <div style="text-align: center;"> <b>0</b>  <div style="border: 1px solid black; padding: 2px;">Reserved</div> 0 </div> </div>	Reserved bit(s) for future use. Software should not rely on the value of the reserved bit. In general case, reserved bits should be setup to a '0' value. Note that reserved bit must be kept at reset value.
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 device 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 access of Cortex®-M3 are performed on those three buses according to the different purposes and target memory spaces. The memory organization uses a Harvard architecture, pre-defined memory map and up to 4 GB memory of space, making the system flexible and extendable.

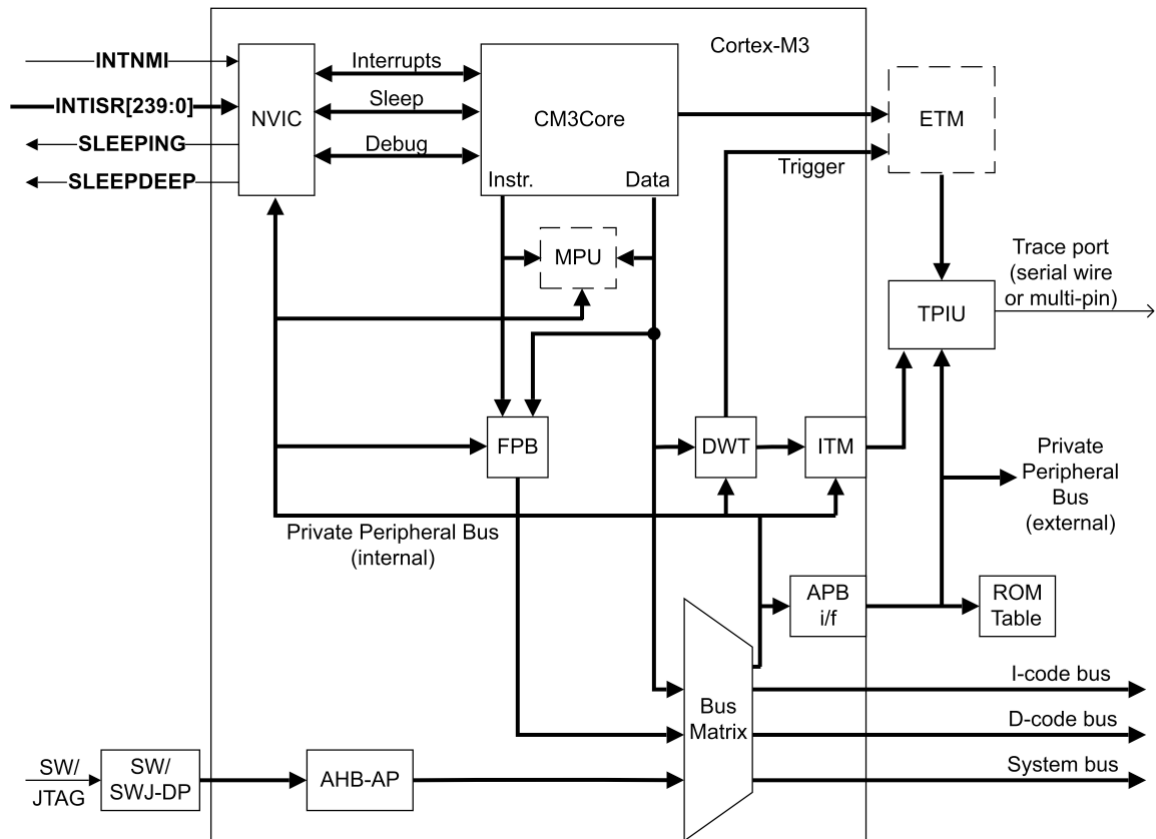
### Arm® Cortex®-M3 Processor

The 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 divider, 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 Debug Port (SW-DP)
- Embedded Trace Macrocell (ETM)
- Trace Port Interface Unit (TPIU)

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

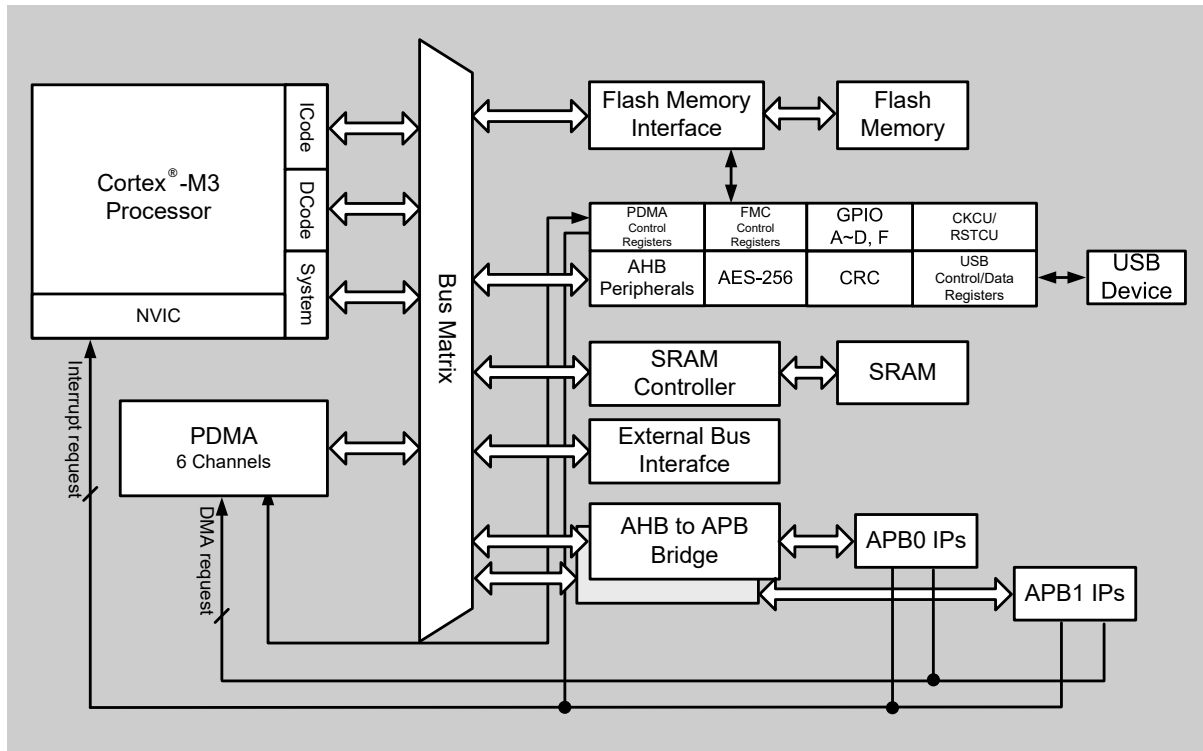




**Figure 2. Cortex®-M3 Block Diagram**

## Bus Architecture

The HT32F12364 device 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. The ICode bus is used for instruction and vector fetches from Code region (0x0000\_0000 ~ 0x1FFF\_FFFF) to Cortex®-M3 core. The DCode bus is used for data load/store and debugging access of Code region. Similarly, the System bus is used for instruction/vector fetch, data load/store, and debugging access of system regions. The system regions include internal SRAM region and peripheral region. All of these master buses are based on 32-bit Advanced High-performance Bus-Lite (AHB-Lite) protocol. The following figure shows the bus architecture of HT32F12364 device.



**Figure 3. Bus Architecture**

## Memory Organization

Arm® Cortex®-M3 is structured in Harvard architecture which can use separate buses to fetch instructions and load/store data. The instruction code and data bus share the same memory address space but in different address ranges. The maximum addressing range of the Cortex®-M3 is 4 GB since it has 32-bit bus address width. Additionally, a pre-defined memory map is provided by the Cortex®-M3 to reduce the software complexity of repeated implementation of different device vendors. However, some regions are used by Arm® Cortex®-M3 system peripherals. Refer to Arm® Cortex®-M3 Technical Reference Manual for more information. The following figure shows the memory map of the HT32F12364 device, including Code, SRAM, peripheral and other pre-defined regions.

## Memory Map

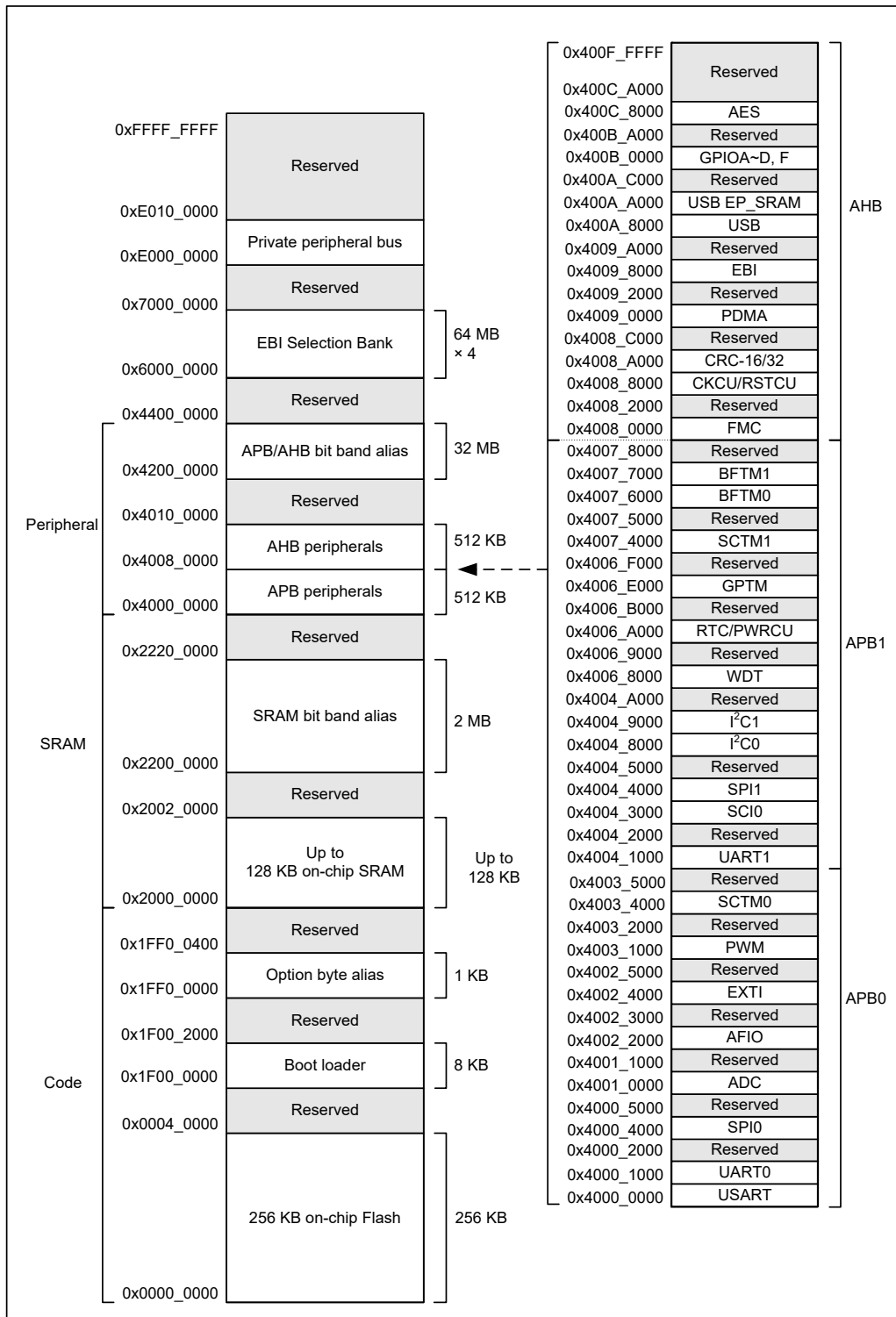


Figure 4. Memory Map

**Table 3. Register Map**

Start Address	End Address	Peripheral	Bus
0x4000_0000	0x4000_0FFF	USART	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	0x4003_0FFF	Reserved	
0x4003_1000	0x4003_1FFF	PWM	
0x4003_2000	0x4003_3FFF	Reserved	
0x4003_4000	0x4003_4FFF	SCTM0	
0x4003_5000	0x4003_FFFF	Reserved	
0x4004_0000	0x4004_0FFF	Reserved	APB1
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	0x4006_7FFF	Reserved	
0x4006_8000	0x4006_8FFF	WDT	
0x4006_9000	0x4006_9FFF	Reserved	
0x4006_A000	0x4006_AFFF	RTC/PWRCU	
0x4006_B000	0x4006_DFFF	Reserved	
0x4006_E000	0x4006_EFFF	GPTM	
0x4006_F000	0x4007_3FFF	Reserved	
0x4007_4000	0x4007_4FFF	SCTM1	
0x4007_5000	0x4007_5FFF	Reserved	
0x4007_6000	0x4007_6FFF	BFTM0	
0x4007_7000	0x4007_7FFF	BFTM1	
0x4007_8000	0x4007_FFFF	Reserved	

Start Address	End Address	Peripheral	Bus
0x4008_0000	0x4008_1FFF	FMC	AHB
0x4008_2000	0x4008_7FFF	Reserved	
0x4008_8000	0x4008_9FFF	CKCU/RSTCU	
0x4008_A000	0x4008_BFFF	CRC-16/32	
0x4008_C000	0x4008_FFFF	Reserved	
0x4009_0000	0x4009_1FFF	PDMA	
0x4009_2000	0x4009_7FFF	Reserved	
0x4009_8000	0x4009_9FFF	EBI	
0x4009_A000	0x400A_7FFF	Reserved	
0x400A_8000	0x400A_9FFF	USB	
0x400A_A000	0x400A_BFFF	USB EP_SRAM	
0x400A_C000	0x400A_FFFF	Reserved	
0x400B_0000	0x400B_1FFF	GPIOA	
0x400B_2000	0x400B_3FFF	GPIOB	
0x400B_4000	0x400B_5FFF	GPIOC	
0x400B_6000	0x400B_7FFF	PIOD	
0x400B_8000	0x400C_9FFF	Reserved	
0x400B_A000	0x400B_BFFF	GPIOF	
0x400B_A000	0x400C_7FFF	Reserved	
0x400C_8000	0x400C_9FFF	AES	
0x400C_A000	0x400F_FFFF	Reserved	

## Embedded Flash Memory

The HT32F12364 device provides a 256 KB on-chip Flash memory which is located at address 0x0000\_0000. It supports bytes, half-words and word access. Note that Flash memory only supports read operation for Cortex®-M3 ICode or DCode bus access. Any write operation to the Flash memory (via DCode bus) will cause a bus fault exception. The Flash memory has a capacity of 256 pages. Each page has a memory capacity of 1 KB and can be erased independently. A 32-bit programming interface provides the capability of changing bits from 1 to 0. A data storage or firmware upgrade can be implemented using several methods such as In System Programming (ISP), In Application Programming (IAP) or In Circuit Programming (ICP). The above programming methods provide flexibility to user for data storage and firmware upgrade purpose. For more information, refer to the Flash Memory Controller section.

## Embedded SRAM Memory

The HT32F12364 device contains a 128 KB on-chip SRAM which is located at address 0x2000\_0000. It supports bytes, half-words and full words access operations. In order to reduce the time of read-modify-write operations, the Cortex®-M3 provides a bit-banding function to perform a single atomic bit operation. Users can modify a single bit in SRAM bit-band region by accessing the corresponding bit-band alias. For more information about bit-banding, refer 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.

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

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

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

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 address of the AHB peripherals ranges from 0x4008\_0000 to 0x400F\_FFFF. Some peripherals such as Clock Control Unit, Reset Control Unit and Flash Memory Controller are connected to the AHB bus directly. The AHB peripheral clocks 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 support only word access.

## APB Peripherals

The address of APB peripherals ranges from 0x4000\_0000 to 0x4007\_FFFF. An APB to AHB bridge provides access capability between the CPU and the APB peripherals. Additionally, the APB peripheral clocks are disabled after a system reset. Software must enable peripheral clock by setting up the APBCCRn registers in Clock Control Unit before accessing the corresponding peripheral register. Note that the APB to AHB bridge will duplicate the half-word or byte data to word width when a half-word or byte access is performed on APB peripheral register. In other words, the access result of half-word or byte access on APB peripheral register will vary depending on the data bit width of the access operation on the peripheral registers.

# 4 Flash Memory Controller (FMC)

## Introduction

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

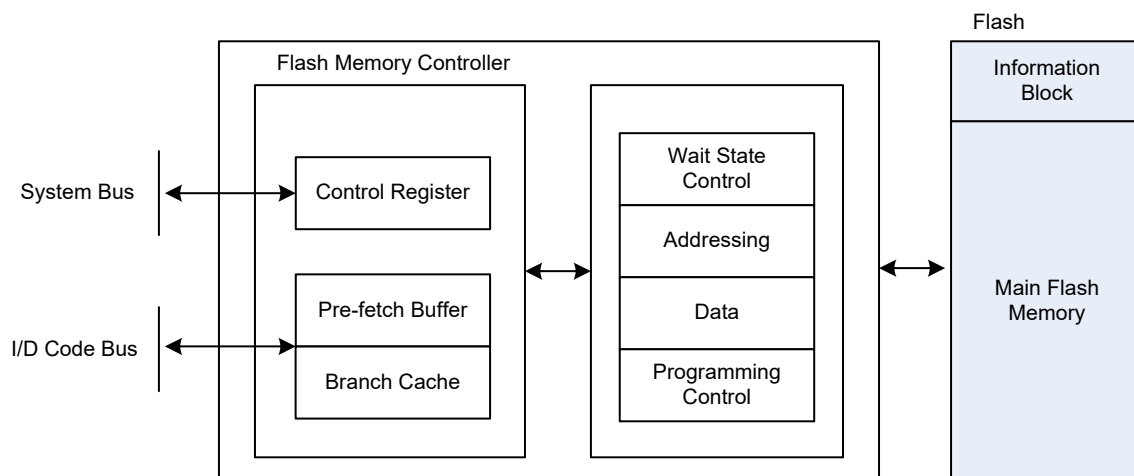


Figure 5. Flash Memory Controller Block Diagram

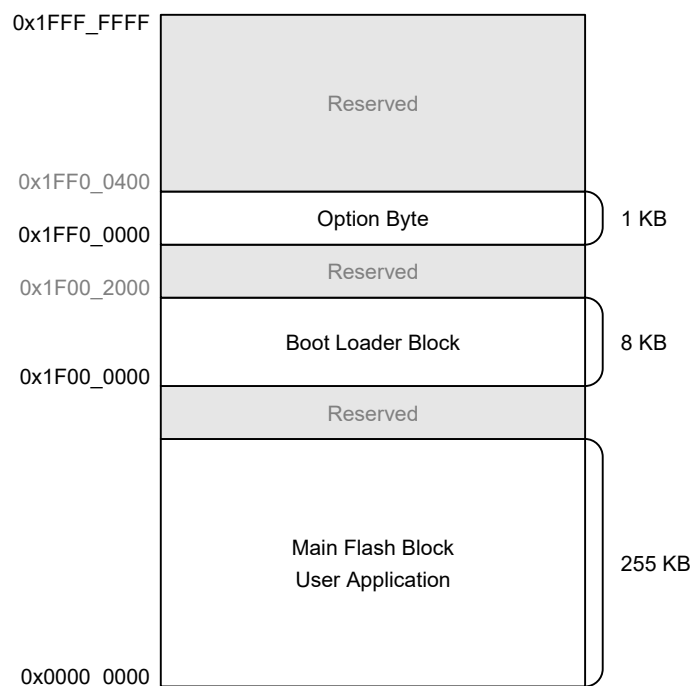
## Features

- 256 KB of on-chip Flash memory for storing instruction/data and option bytes
  - 256 KB: 255 KB + 1 KB (instruction/data + Option Byte)
- Page size of 1 KB, totally 256 pages
- 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 occurrence
- Flash read protection to prevent illegal code/data access
- Page erase/program protection to prevent unexpected operation

## Functional Descriptions

### Flash Memory Map

The following figure is the Flash memory map of the system. The address ranges from 0x0000\_0000 to 0x1FFF\_FFFF (0.5 GB). The address from 0x1F00\_0000 to 0x1F00\_1FFF is mapped to Boot Loader with a capacity of 8 KB. Additionally, the region addressed from 0x1FF0\_0000 to 0x1FF0\_03FF is the alias of Option Byte block 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 256 KB main Flash with 1 KB per page and an 8 KB Information Block for the Boot Loader. The main Flash memory contains a total of 256 pages which can be erased individually. The following table shows the base address, size and protection setting bit of each page.

**Table 4. Flash Memory and Option Byte**

Block	Name	Address	Page Protection Bit	Size
Main Flash Block	Page 0	0x0000_0000 ~ 0x0000_03FF	OB_PP [0]	1 KB
	Page 1	0x0000_0400 ~ 0x0000_07FF		1 KB
	Page 2	0x0000_0800 ~ 0x0000_0BFF	OB_PP [1]	1 KB
	Page 3	0x0000_0C00 ~ 0x0000_0FFF		1 KB
	⋮	⋮	⋮	⋮
	Page 252	0x0003_F000 ~ 0x0003_F3FF	OB_PP [126]	1 KB
	Page 253	0x0003_F400 ~ 0x0003_F7FF		1 KB
	Page 254	0x0003_F800 ~ 0x0003_FBFF	OB_PP [127]	1 KB
	Page 255 (Option Byte)	Physical: 0x0003_FC00 ~ 0x0003_FFFF Alias: 0x1FF0_0000 ~ 0x1FF0_03FF	OB_CP [1]	1 KB
Information Block	Boot Loader	0x1F00_0000 ~ 0x1F00_1FFF	NA	8 KB

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

## Wait State Setting

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

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

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

**Table 5. Relationship Between Wait State Cycle and HCLK**

Wait State Cycle	HCLK
0	0 MHz < HCLK ≤ 20 MHz
1	20 MHz < HCLK ≤ 40 MHz
2	40 MHz < HCLK ≤ 60 MHz
3	60 MHz < HCLK ≤ 72 MHz

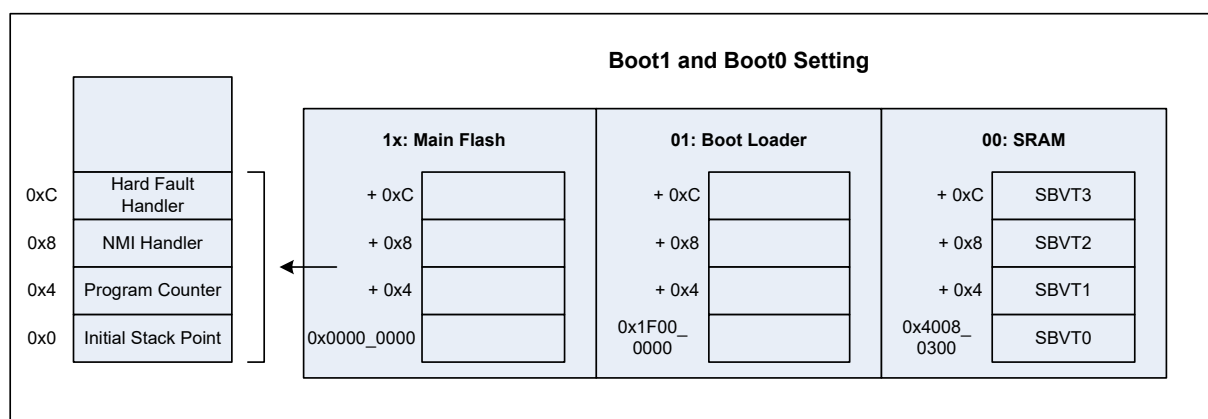
## Booting Configuration

The system provides three kinds of booting mode which can be selected through BOOT0 and BOOT1 pins. The BOOT0 and BOOT1 pin status 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 booting mode. The booting modes are shown in the following table.

**Table 6. Booting Modes**

Booting Mode Selection Pins		Mode	Descriptions
BOOT1	BOOT0		
0	0	SRAM	The source of Vector is SBVT0 ~ 3
0	1	Boot Loader	The source of Vector is Boot Loader
1	X	Main Flash	The source of Vector is main Flash

The Flash Vector Mapping Control Register (VMCR) is provided to change the setting of 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.



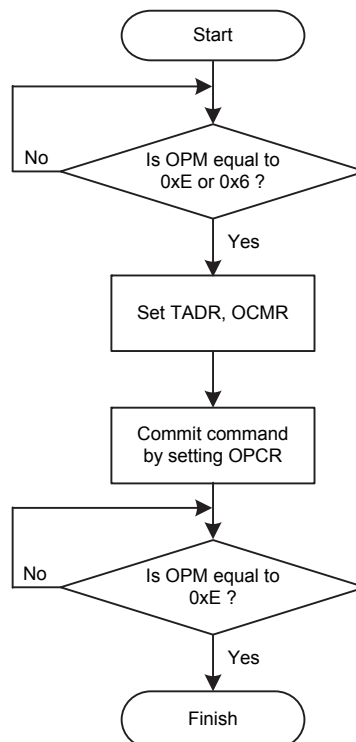
**Figure 7. Vector Remapping**

## Page Erase

The FMC provides a page erase function which is used to partial content of the Flash memory. Any page can be erased independently without affecting others. 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] is equal to 0xE or 0x6). Otherwise, wait until the previous operation has been finished.
- Write the page address to TADR register.
- Write the page erase command to the OCMR register (CMD [3:0] = 0x8).
- Commit the page erase command to 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] is 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 under the code fetching or data accessing status. The FMC will not provide any notification when this occurs. Additionally, the page erase operation will be ignored on the protected pages. When this occurs, the OREF bit will be set by the FMC and then a Flash Operation Error interrupt will be generated if the OREIEN bit in the OIER register is set. 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.



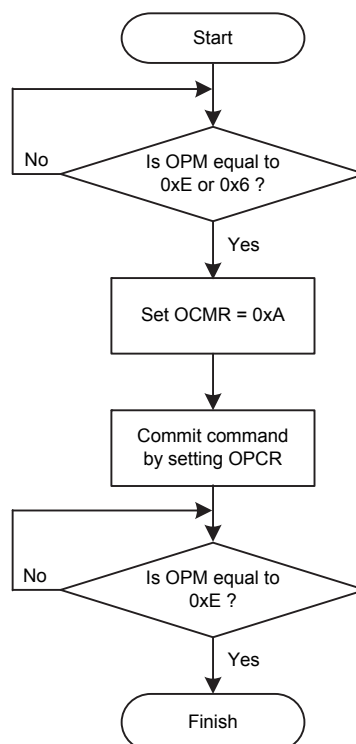
**Figure 8. Page Erase Operation Flowchart**

## 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 sequence.

- Check the OPCR register to confirm that no Flash memory operation is in progress (OPM [3:0] is equal to 0xE or 0x6). Otherwise, wait until the previous operation has been finished.
- Write the mass erase command to the OCMR register (Set CMD [3:0] = 0xA).
- Commit 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] is equal to 0xE).
- Read and verify the Flash memory if required using DCODE access.

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



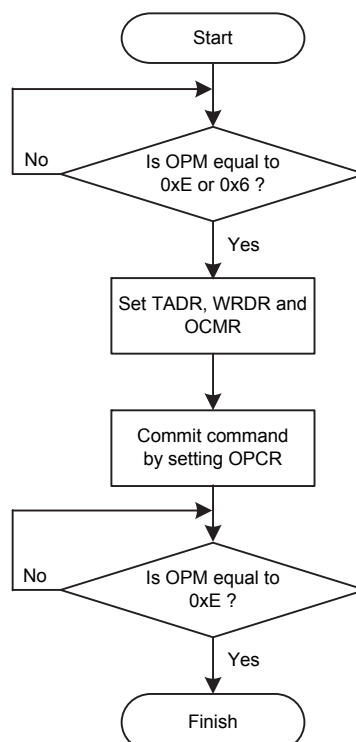
**Figure 9. Mass Erase Operation Flowchart**

## Word Programming

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

- Check the OPCR register to confirm that no Flash memory operation is in progress (OPM [3:0] is equal to 0xE or 0x6). Otherwise, wait until the previous operation has been finished.
- Write the word address to the TADR register. Write data to the WRDR register.
- Write the word programming command to the OCMR register (Set 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] is equal to 0xE).
- Read and verify the Flash memory if required using DCODE access.

Note that the word programming operation cannot be successively applied to the same address twice. Successive word programming operation to the same address must be separated by a page erase operation. Additionally, the word programming operation will be ignored on protected pages. When this occurs, the OREF bit will be set by the FMC and then a Flash Operation Error interrupt will be generated if the OREIEN bit in the OIER register is set. 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 can be treated as an independent Flash memory which base address is 0x1FF0\_0000. The following table shows the functional description and the Option Byte memory map.

**Table 7. Option Byte Memory Map**

Option Byte	Offset	Description	Reset Value
Option Byte Base Address = 0x1FF0_0000			
OB_PP	0x000	OB_PP [n]: Main Flash Page Erase/Program Protection (n = 0 ~ 127) 0: Flash Page 2n and 2n + 1 Erase / Program Protection is enabled 1: Flash Page 2n and 2n + 1 Erase / Program Protection is disabled	0xFFFF_FFFF
	0x004		0xFFFF_FFFF
	0x008		0xFFFF_FFFF
	0x00C		0xFFFF_FFFF
OB_CP	0x010	OB_CP [0]: Flash Security Protection 0: Flash Security protection is enabled 1: Flash Security protection is disabled	0xFFFF_FFFF
		OB_CP [1]: Option Byte Protection 0: Option Byte protection is enabled 1: Option Byte protection is disabled	
		OB_CP [31:2]: Reserved	
OB_CK	0x020	OB_CK [31:0]: Flash Option Byte Checksum OB_CK should be set as the sum of 5 words Option Byte content, of which the offset address ranges from 0x000 to 0x010 (0x000 + 0x004 + 0x008 + 0x00C + 0x010), when the OB_PP or OB_CP register content is not equal to 0xFFFF_FFFF. Otherwise, both page erase/program protection and security protection will be enabled.	0xFFFF_FFFF
OB_WDT	0x02C	Flash Option Watchdog Timer Enable OB_WDT [15:0]: 0x7A92 If the OB_WDT [15:0] is set to 0x7A92, the WDT will be enabled immediately when the MCU power on reset or system reset occurs. The WDT can be disabled by software. OB_WDT [31:16]: Reserved	0xFFFF_FFFF
OB_TOOL	0x030 ~ 0x04C	Reserved for Flash writer tool and boot loader.	0xFFFF_FFFF

## Page Erase/Program Protection

The FMC provides page erase/program protection functions to prevent inadvertent operations on the protected Flash memory area. The page erase (CMD [3:0] = 0x8 in the OCMR register) or word programming (CMD [3:0] = 0x4) command will not be accepted by the FMC on the protected pages. When the page erase or word programming command aimed at the protected pages is sent to the FMC, the PPEF bit in the OISR register will then be set by the FMC and the Flash Operation Error interrupt will be triggered to inform the CPU if the OREIEN bit in the OIER register is set. The page protection function can be individually enabled for each page by configuring the OB\_PP [127:0] bit field in the Option Byte. The following table shows the access permission of the main Flash page when the page protection is enabled.

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

Operation \ Mode	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

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

The following steps show the 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] is equal to 0xE or 0x6). Otherwise, wait until the previous operation has been finished.
- Write the OB\_PP address to the TADR register (Set TADR = 0x1FF0\_0000).
- Write the desired data, which indicates the protection function of the corresponding page is to be enabled or disabled, into the WRDR register (0: Enabled, 1: Disabled).
- Write the word programming command to the OCMR register (Set CMD [3:0] = 0x4).
- Commit 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] is equal to 0xE).
- Read and verify the Option Byte 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 activate the new setting.

## Security Protection

The FMC provides a security protection function to prevent illegal code/data access of the Flash memory. This function is useful for protecting the software / firmware from the 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**

Operation	Mode	User Application <sup>(1)</sup>	ICP/Debug Mode	Boot from SRAM
DCODE Read		O	X (read as 0)	X (read as 0)
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 SW debugger being disconnected. However, the Option Byte block and page 0 are still protected in which Programming and Page Erase operations cannot be executed.

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

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] is equal to 0xE or 0x6). Otherwise, wait until the pervious operation has been finished.
- Write the OB\_CP address to the TADR register (Set TADR = 0x1FF0\_0010).
- Write the data into the WRDR register to clear OB\_CP [0] bit to 0.
- Write the word programming command to the OCMR register (Set 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] equals to 0xE).
- Read and verify the Option Byte if required using DCODE access.
- Before to activate the security protection function, the OB\_CK field must be update according to the Option Byte checksum rule.
- Apply a system reset to activate 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
TADR	0x000	Flash Target Address Register	0x0000_0000
WRDR	0x004	Flash Write Data Register	0x0000_0000
OCMR	0x00C	Flash Operation Command Register	0x0000_0000
OPCR	0x010	Flash Operation Control Register	0x0000_000C
OIER	0x014	Flash Operation Interrupt Enable Register	0x0000_0000
OISR	0x018	Flash Operation Interrupt and Status Register	0x0001_0000
PPSR	0x020 0x024 0x028 0x02C	Flash Page Erase/Program Protection Status Register	0xFFFF_XXXX 0xFFFF_XXXX 0xFFFF_XXXX 0xFFFF_XXXX
CPSR	0x030	Flash Security Protection Status Register	0x0000_000X
VMCR	0x100	Flash Vector Mapping Control Register	0x0000_000X
MDID	0x180	Flash Manufacturer and Device ID Register	0x0376_XXXX
PNSR	0x184	Flash Page Number Status Register	0x0000_00FF
PSSR	0x188	Flash Page Size Status Register	0x0000_0400
DID	0x18C	Device ID Register	0x000X_XXXX
CFCR	0x200	Flash Cache and Pre-fetch Control Register	0x0000_1091
SBVT0	0x300	SRAM Booting Vector 0 (Stack Point)	0x200X_0000
SBVT1	0x304	SRAM Booting Vector 1 (Program Counter)	0x2000_0159
SBVT2	0x308	SRAM Booting Vector 2 (NMI Handler)	0x0000_0000
SBVT3	0x30C	SRAM Booting Vector 3 (Hard Fault Handler)	0x0000_0000
CIDR0	0x310	Custom ID Register 0	0xFFFF_XXXX
CIDR1	0x314	Custom ID Register 1	0xFFFF_XXXX
CIDR2	0x318	Custom ID Register 2	0xFFFF_XXXX
CIDR3	0x31C	Custom ID Register 3	0xFFFF_XXXX

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

## Register Descriptions

### Flash Target Address Register – TADR

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

Offset: 0x000

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[31:0]	TADB	<p>Flash Target Address Bits</p> <p>For programming operations, the TADR register specifies the address where the data is written. Since the programming length is 32-bit, the TADR should be set as word-aligned (4 bytes). The TADB [1:0] will be ignored during programming operations. For page erase operations, the TADR register contains the page address which is going to be erased. Since the page size is 1 KB, the TADB [9:0] will be ignored in order to limit the target address as 1 Kbyte-aligned. For 256 KB main Flash addressing, TADB [31:18] should be zero. The Option Byte which has a 1 KB capacity ranges from 0x1FF0_0000 to 0x1FF0_03FF. This field is used to specify the Flash address which must be within the range from 0x0000_0000 to 0x1FFF_FFFF. Otherwise, an Invalid Target Address interrupt will be generated if the corresponding interrupt enable bit is set.</p>

## Flash Write Data Register – WRDR

This register stores the data to be written into the TADR register for programming operation.

Offset: 0x004

Reset value: 0x0000\_0000

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

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

## Flash Operation Command Register – OCMR

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

Offset: 0x00C

Reset value: 0x0000\_0000

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

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

## Flash Operation Control Register – OPCR

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

Offset: 0x010

Reset value: 0x0000\_000C

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

Bits	Field	Descriptions
[4:1]	OPM	<p>Operation Mode</p> <p>The following table shows the FMC operation modes. Users can commit 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 to 0xE by the FMC hardware. The Idle mode can be set when all the operations have been finished for power saving purpose. Note that the operation status should be checked before executing next operation. The content of the TADR, WRDR, OCMR and OPCR registers should not be changed until the previous operation has been finished.</p>

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

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

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			OREIEN	IOCMIEN	OBEIEN	ITADIEN	ORFIEN
			RW	0	RW	0	RW	0

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

## 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 or if an error has occurred. The status bits, bit [4:0], if set high, are available to trigger the interrupts when the corresponding enable bits in the OIER register are set high.

Offset: 0x018

Reset value: 0x0001\_0000

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

Bits	Field	Descriptions
[17]	PPEF	Page Erase/Program Protected Error Flag 0: Page Erase/Program Protected Error does not occur 1: Operation error occurs due to an invalid page 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 is not finished 1: The last Flash operation command is 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 Operation Error interrupt will be generated 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 has been written into the OCMR register 1: An invalid Flash operation command has been written into the OCMR register This bit will be set when an invalid Flash operation command has been written into the OCMR register. Then the Invalid Operation Command interrupt will be generated if the IOCMIEN bit in the OIER register is set. Reset this bit by writing 1.

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



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

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

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

Reset value: 0xFFFF\_FFFF

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

Bits	Field	Descriptions
[127:0]	PPSBn	<p>Page Erase/Program Protection Status Bits (n = 0 ~ 127)</p> <p>PPSB[n] = OB_PP[n]</p> <p>0: The corresponding pages are protected</p> <p>1: The corresponding pages are not protected</p> <p>The content of this register is not dynamically updated and will only be reloaded from the Option Byte when any kind of reset occurs. The erase or program function of the specific pages is not allowed when the corresponding bits of the PPSR registers are reset. The reset value of PPSR [127:0] is determined by the Option Byte OB_PP [127:0] bits. Since the maximum page number of the main flash is 256 KB in this device. Therefore, each page erase/program protection status bit protect two pages. The other remained bits of OB_PP and PPSR registers are reserved.</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 active when any kind of reset occurs.

Offset: 0x030

Reset value: 0x0000\_000X

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

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

## Flash Vector Mapping Control Register – VMCR

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

Offset: 0x100

Reset value: 0x0000\_000X

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

Bits	Field	Descriptions																				
[1:0]	VMCB	<p>Vector Mapping Control Bit</p> <p>The VMCB bits are used to control the mapping source of first 4-word vector addressed from 0x0 to 0xC. The following table shows the vector mapping setting.</p> <table><tr><th>BOOT1</th><th>BOOT0</th><th>VMCB [1:0]</th><th>Description</th></tr><tr><td>Low</td><td>Low</td><td>00</td><td>SRAM booting mode The vector mapping source is SBVT0 ~ 3.</td></tr><tr><td>Low</td><td>High</td><td>01</td><td>Boot Loader mode The vector mapping source is the boot loader area.</td></tr><tr><td>High</td><td>Low</td><td>10</td><td>Main Flash mode</td></tr><tr><td>High</td><td>High</td><td>11</td><td>The vector mapping source is the main Flash area.</td></tr></table> <p>The reset value of the VMCB register is determined by the pins status of the external booting pins BOOT1 and BOOT0 during power on reset and system reset. The vector mapping setting can be changed temporarily by configuring the VMCB bits when the application program is running.</p>	BOOT1	BOOT0	VMCB [1:0]	Description	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	High	High	11	The vector mapping source is the main Flash area.
BOOT1	BOOT0	VMCB [1:0]	Description																			
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																			
High	High	11	The vector mapping source is the main Flash area.																			

## Flash Manufacturer and Device ID Register – MDID

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

Offset: 0x180

Reset value: 0x0376\_XXXX

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

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

## Flash Page Number Status Register – PNSR

This register is used to indicate the Flash memory page number.

Offset: 0x184

Reset value: 0x0000\_00FF

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

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

## Flash Page Size Status Register – PSSR

This register is used to indicate the page size in bytes.

Offset: 0x188

Reset value: 0x0000\_0400

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

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

## Device ID Register – DID

This register is used to store the device part number information which can be used as the product identity.

Offset: 0x18C

Reset value: 0x000X\_XXXX

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

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

## Flash Cache and Pre-fetch Control Register – CFCCR

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

Offset: 0x200

Reset value: 0x0000\_1091

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

Bits	Field	Descriptions
[12]	CE	Branch Cache Enable Bit 0: Cache is disabled 1: Cache is enabled
[7]	DCDB	DCODE Data Cacheable Control Bit 0: DCODE Data Access is Cacheable 1: DCODE Data Access is Non-Cacheable
[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 by default setting. When the pre-fetch buffer is disabled, the instruction and Data are provided by the Flash memory directly.
[2:0]	WAIT	Flash Wait State Setting The WAIT [2:0] field is used to set the HCLK wait clock during non-sequential address 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.

WAIT [2:0]	Wait Status	Allowed HCLK Range
001	0	0 MHz < HCLK <= 20 MHz
010	1	20 MHz < HCLK <= 40 MHz
011	2	40 MHz < HCLK <= 60 MHz
100	3	60 MHz < HCLK <= 72 MHz
Others	Reserved	Reserved

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

These registers specify the initial values of Stack Point, Program Counter, NMI Handler address and Hard Fault Handler address for the SRAM Booting mode.

Offset: 0x300 (0) ~ 0x30C (3)

Reset value: Various depending on the address offset

	31	30	29	28	27	26	25	24	
	SBVTn								
Type/Reset	RW	X RW	X RW	X RW	X RW	X RW	X RW	X RW	X
	23	22	21	20	19	18	17	16	
	SBVTn								
Type/Reset	RW	X RW	X RW	X RW	X RW	X RW	X RW	X RW	X
	15	14	13	12	11	10	9	8	
	SBVTn								
Type/Reset	RW	X RW	X RW	X RW	X RW	X RW	X RW	X RW	X
	7	6	5	4	3	2	1	0	
	SBVTn								
Type/Reset	RW	X RW	X RW	X RW	X RW	X RW	X RW	X 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 an SRAM booting capability for applications debugging. The contents of the SBVTn registers are re-mapped into addresses 0x0 ~ 0xC of the Flash memory CODE area under SRAM booting 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 booting mode. Those registers can be modified by the debugging tool in order to change the program execution setting. The reset values of SBVTn will be reloaded only by a power-on reset. Other reset sources will have no effect.</p> <table><tr><th>Name</th><th>Address Offset</th><th>Purpose Descriptions</th><th>Reset Value</th></tr><tr><td>SBVT0</td><td>0x300</td><td>Stack point</td><td>128 KB SRAM: 0x2002_0000 64 KB SRAM: 0x2001_0000</td></tr><tr><td>SBVT1</td><td>0x304</td><td>Program counter</td><td>0x2000_0159</td></tr><tr><td>SBVT2</td><td>0x308</td><td>NMI handler address</td><td>0x0000_0000</td></tr><tr><td>SBVT3</td><td>0x30C</td><td>Hard fault handler address</td><td>0x0000_0000</td></tr></table>	Name	Address Offset	Purpose Descriptions	Reset Value	SBVT0	0x300	Stack point	128 KB SRAM: 0x2002_0000 64 KB SRAM: 0x2001_0000	SBVT1	0x304	Program counter	0x2000_0159	SBVT2	0x308	NMI handler address	0x0000_0000	SBVT3	0x30C	Hard fault handler address	0x0000_0000
Name	Address Offset	Purpose Descriptions	Reset Value																			
SBVT0	0x300	Stack point	128 KB SRAM: 0x2002_0000 64 KB SRAM: 0x2001_0000																			
SBVT1	0x304	Program counter	0x2000_0159																			
SBVT2	0x308	NMI handler address	0x0000_0000																			
SBVT3	0x30C	Hard fault handler address	0x0000_0000																			

This access width of the registers SBVT0 ~SBVT3 must be 32 bits (Word access). 8 or 16 bits (Byte or Half-Word) access is not allowed.



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

This register is used to store the custom ID information which can be used as the custom identity.

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

Reset value: Various depending on Flash Manufacture Privilege Information Block.

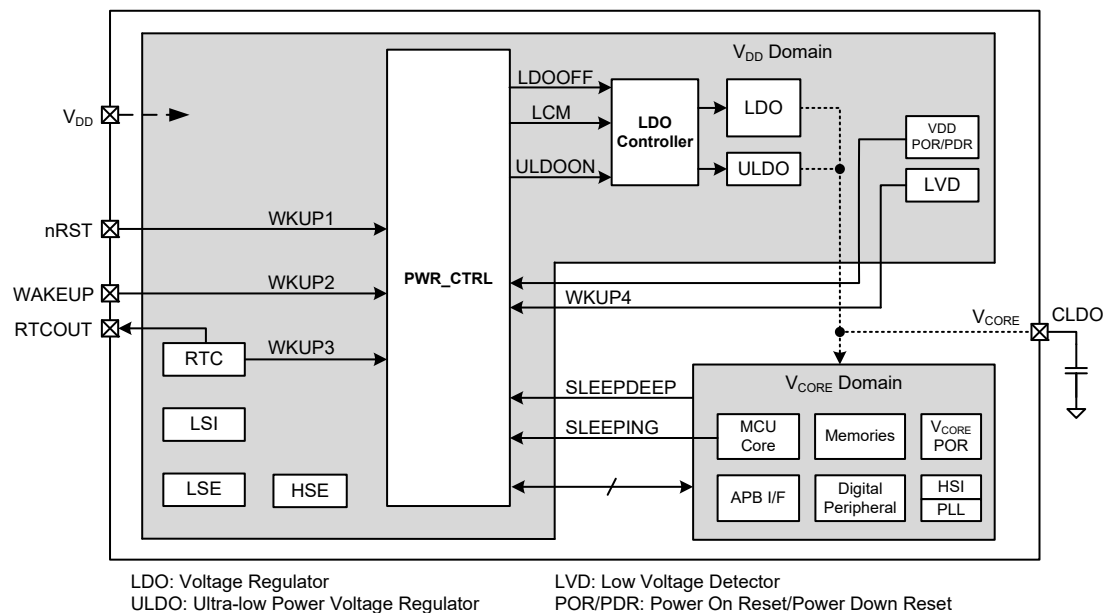
	31	30	29	28	27	26	25	24	
	CID								
Type/Reset	RO	X RO	X RO	X RO	X RO	X RO	X RO	X RO	X
	23	22	21	20	19	18	17	16	
	CID								
Type/Reset	RO	X RO	X RO	X RO	X RO	X RO	X RO	X RO	X
	15	14	13	12	11	10	9	8	
	CID								
Type/Reset	RO	X RO	X RO	X RO	X RO	X RO	X RO	X RO	X
	7	6	5	4	3	2	1	0	
	CID								
Type/Reset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

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

## 5 Power Control Unit (PWRCU)

# Introduction

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



### Figure 11. PWRCU Block Diagram

## Features

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

## Functional Descriptions

### $V_{DD}$ Power Domain

#### LDO Power Control

The main LDO and ultra-low power LDO (ULDO) will be automatically switched off when one of the following conditions occurs:

- The Power-Down mode is entered
- The supply power  $V_{DD} \leq V_{PDR}$

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

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

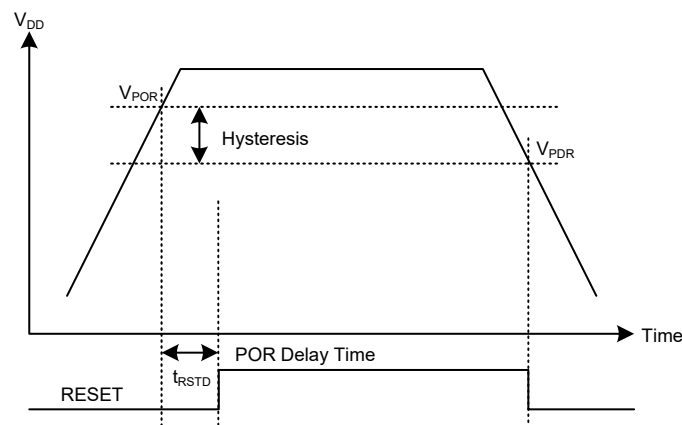
To enter the Deep-Sleep1 or Deep-Sleep2 mode, the PWRCU will turn off the main LDO and request the ULDO to operate in the low standby current mode to supply an alternative  $V_{CORE}$  power.

#### Voltage Regulator

The main voltage regulator, LDO, ultra-low power LDO, ULDO, Low voltage Detector, LVD, Low Speed Internal RC oscillator, LSI, High Speed External Crystal oscillator, HSE, and the Low Speed External Crystal oscillator, LSE, are operated under the  $V_{DD}$  power domain. The main LDO can be configured to operate in normal mode ( $LDOOFF = 0$ ,  $LDOLCM = 0$ ,  $I_{OUT} = \text{high current mode}$ ) and the ultra-low power LDO can be configured to operate in low current mode ( $LDOOFF = 0$ ,  $LDOLCM = 1$ ,  $I_{OUT} = \text{low current mode}$ ) to supply the  $V_{CORE}$  power. The ULDO output has ultra-low static current characteristics and can be configured to operate in the Deep-Sleep2 mode for data retention purposes in the  $V_{CORE}$  power domain. It is controlled using the ULDOON bit in the PWRCR register.

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

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



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

### Low Voltage Detector / Brown-Out Detector

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

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

### High Speed External Oscillator

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

### LSE, LSI and RTC

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

### V<sub>CORE</sub> Power Domain

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

#### High Speed Internal Oscillator

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

### Operation Modes

#### Run Mode

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

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

**Table 11. Operation Mode Definitions**

Mode Name	Hardware Action
Run	After system reset, CPU fetches instructions to execute.
Sleep	CPU clock will be stopped. Peripherals, Flash and SRAM clocks can be stopped by setting.
Deep-Sleep1 ~ 2	Stop all clocks in the V <sub>CORE</sub> power domain. Disable HSI, HSE and PLL. Turn on the ultra-low power ULDO to reduce the V <sub>CORE</sub> power domain current.
Power-Down	Shut down the V <sub>CORE</sub> power domain.

### Sleep Mode

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

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

Mode	Mode Entry				Mode Exit
	CPU Instruction	CPU SLEEPDEEP	LDOOFF	ULDOON	
Sleep	WFI or WFE (Takes effect)	0	X	X	WFI: Any interrupt WFE: Any wakeup event <sup>(1)</sup> or Any interrupt (NVIC on) or Any interrupt with SEVONPEND = 1 (NVIC off)
Deep-Sleep1		1	0	0	Any EXTI in event mode or RTC wakeup or LVD wakeup <sup>(2)</sup> or WAKEUP pin rising edge or USB resume
Deep-Sleep2		1	X	1	Any EXTI in event mode or RTC wakeup or LVD wakeup <sup>(2)</sup> or WAKEUP pin rising edge
Power-Down		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 trigger event, RTC event, LVD event or WAKEUP pin rising edge.  
2. If the system allows the LVD activity to wake it up after the system has entered the power saving mode, the LVDEWEN and LVDEN bits in the LVDCSR register must be set to 1 to make sure that the system can be woken up by an LVD event and then the main LDO can be turned on when the system is woken up from the Deep-Sleep2 mode and Power-Down mode.

### Deep-Sleep Mode

To enter the Deep-Sleep mode, configure the registers as shown in the preceding table and execute the WFI or WFE instruction. In the Deep-Sleep mode, all clocks including the PLL and high speed oscillators, known as HSI and HSE, will be stopped. In addition, the Deep-Sleep1 and Deep-Sleep2 mode will turn off the main LDO and use an ultra-low power LDO, ULDO, to keep the  $V_{CORE}$  power. Once the PWRCU receives a wakeup event or an interrupt as shown in the preceding Mode-Exiting table, the main LDO will then operate in normal mode and the high speed oscillators will be enabled. Finally, the CPU will return to the Run mode to handle the wakeup interrupt if required. A Low Voltage Detection also can be regarded as a wakeup event if the corresponding wakeup control bit LVDEWEN in the LVDCSR register is enabled. The last wakeup event is a transition from low to high on the external WAKEUP pin sent to the PWRCU to resume from the Deep-Sleep mode. During the Deep-Sleep mode, retaining the register and memory contents will shorten the wakeup latency.

### Power-Down Mode

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

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

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

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

## Register Map

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

**Table 14. PWRCU Register Map**

Register	Offset	Description	Reset Value
PWRSR	0x100	Power Control Status Register	0x0000_0001
PWRCR	0x104	Power Control Register	0x0000_0000
PWRTEST	0x108	$V_{DD}$ Power Domain Test Register	0x0000_0027
LVDCSR	0x110	Low Voltage / Brown-Out Detect Control and Status Register	0x0000_0000
PWRLDOSR	0x11C	Power Control LDO Status Register	0x30XX_XXXX

## Register Descriptions

### Power Control Status Register – PWRSR

This register indicates the power control status.

Offset: 0x100

Reset value: 0x0000\_0001 (Reset only by  $V_{DD}$  domain power-on reset)

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

Bits	Field	Descriptions
[8]	WUPF	External WAKEUP Pin Flag 0: The WAKEUP pin is not asserted 1: The WAKEUP pin is asserted This bit is set by hardware when the WAKEUP pin is asserted and is cleared by software read. Software should read this bit to clear it after a system wakeup from the power saving mode.
[1]	PDF	Power-Down Flag 0: Wakeup from abnormal $V_{CORE}$ shutdown (loss of $V_{CORE}$ is unexpected) 1: Wakeup from Power-Down mode (loss of $V_{CORE}$ is under expectation) This bit is set by hardware when the system has successfully entered the Power-Down mode under the correct firmware procedure. This bit is cleared by software read.



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

## Power Control Register – PWRCR

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

Offset: 0x104

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

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

Bits	Field	Descriptions
[15]	ULDOSTS	<p>Ultra-low power regulator Status</p> <p>If the ULDOON bit in this register has been set to 1, this bit will be set high after the MCU is waken up from the Deep-Sleep2 mode.</p> <p>This bit is cleared to 0 if the ULDOON bit has been cleared to 0 or if a POR/PDR reset has occurred.</p>
[9]	WUPIEN	<p>External WAKEUP Pin Interrupt Enable</p> <p>0: Disable WAKEUP pin interrupt function 1: Enable WAKEUP pin interrupt function</p> <p>The software can set the WUPIEN bit to 1 to assert the WKUP interrupt in the NVIC unit when both the WUPEN and WUPF bits are set to 1.</p>

Bits	Field	Descriptions
[8]	WUPEN	<p>External WAKEUP Pin Enable</p> <p>0: Disable WAKEUP pin function 1: Enable WAKEUP pin function</p> <p>The software can set the WUPEN bit as 1 to enable the WAKEUP pin function before entering the power saving mode. When WUPEN = 1, a rising edge on the WAKEUP pin wakes up the system from the power saving mode. As the WAKEUP pin is active high, this pin should be configured to an input and pull-down state. Note: This bit is reset by a <math>V_{DD}</math> Power Domain reset. Because this bit is located in the <math>V_{DD}</math> Power Domain, after the reset activity there will be a delay until the bit is active. The bit will not be active until the system reset finished and the <math>V_{DD}</math> 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 <math>V_{DD}</math> domain ISO signal disabled. The delay time needed is a minimum of three 32 kHz clock periods until the bit reset activity has finished.</p>
[7]	ULDOON	<p>Ultra-low power regulator Control</p> <p>0: ULDO is OFF 1: ULDO is ON</p> <p>An ultra-low power regulator ULDO is implemented to provide an alternative voltage source for the <math>V_{CORE}</math> power domain when the MCU enters the Deep-Sleep mode (SLEEPDEEP = 1). The control bit ULDOON is set by software and cleared by software or <math>V_{DD}</math> power domain reset. If the ULDOON bit is set to 1, the main LDO will automatically be turned off when the MCU enters the Deep-Sleep mode.</p>
[5:4]	LDOFTRM	<p>LDO Output Voltage Fine Trim</p> <p>00: The LDO default output voltage 01: The LDO default output voltage offset - 5 % 10: The LDO default output voltage offset + 3 % 11: The LDO default output voltage offset + 7 %</p> <p>These bits will be clear to 0 when the LDO is power down or <math>V_{DD}</math> power domain reset.</p>
[3]	LDOOFF	<p>Main regulator Operating Mode Control</p> <p>0: The LDO operates in a low current mode when CPU enters the Deep-Sleep mode (SLEEPDEEP = 1). That means MCU <math>V_{CORE}</math> power is available and supplied by the ultra-low power regulator ULDO. 1: The LDO is turned off when the MCU enters the Deep-Sleep mode (SLEEPDEEP = 1). That means MCU <math>V_{CORE}</math> power is not available and enters the Power Down mode.</p> <p>Note: This bit is only available when the ULDOON bit is cleared to 0.</p>
[2]	LDOLCM	<p>LDO Low Current Mode</p> <p>0: The <math>V_{CORE}</math> power domain is supplied by main regulator LDO in normal current mode. 1: The <math>V_{CORE}</math> power domain is supplied by ultra-low power regulator ULDO in low current mode.</p> <p>Note: This bit is only available when MCU is in the Run mode. The ULDO output current capability will be limited at 5 mA below and lower static current when the LDOLCM bit is set. It is suitable for MCU which is operated at lower speed system clock to get a lower current consumption. This bit will be cleared to 0 when the MCU enters the Power Down mode or a <math>V_{DD}</math> power domain reset occurs.</p>
[0]	PWRST	<p><math>V_{DD}</math> Power Domain Software Reset</p> <p>0: No action 1: <math>V_{DD}</math> Power Domain Software Reset is activated</p> <p>When this bit is set high, it will reset all RTC and PWRCU related registers.</p>

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

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

Offset: 0x108

Reset value: 0x0000\_0027

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

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

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

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

Offset: 0x110

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

	31	30	29	28	27	26	25	24		
Type/Reset	Reserved									
	23	22	21	20	19	18	17	16		
Type/Reset	Reserved	LVDS [2]	LVDEWEN	LVDIWEN	LVDF	LVDS [1:0]		LVDEN		
		RW	0	RW	0	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				BODF	Reserved	BODRIS	BODEN		
					RO	0	RW	0	RW	0

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

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

## Power Control LDO Status Register – PWRLDOSR

This register specifies the LDO and the ULDO progress status.

Offset: 0x11C

Reset value: 0x30XX\_XXXX

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

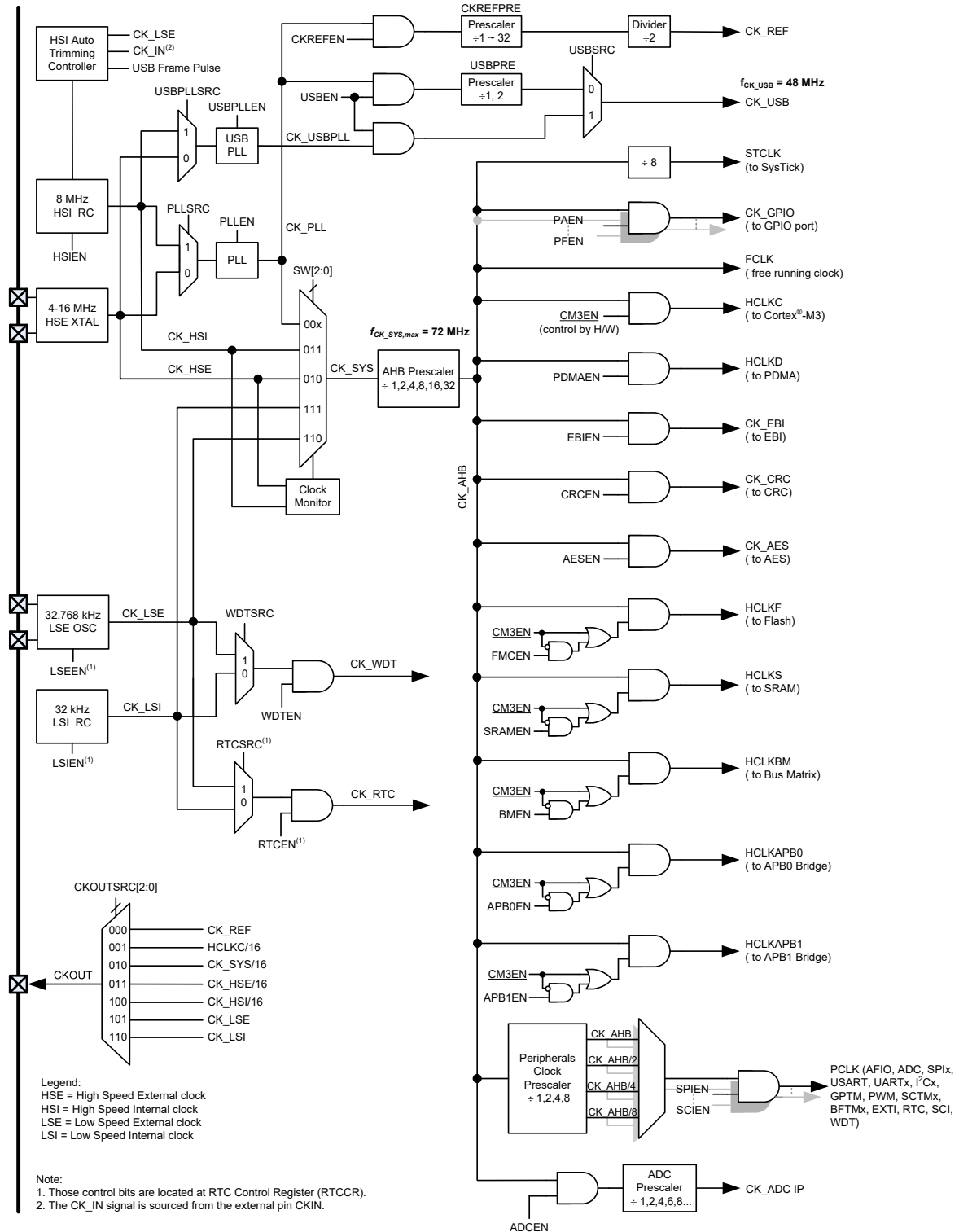
Bits	Field	Descriptions
[29:28]	LDOPST	Main regulator Progress Status 00: LDO off 01: LDO on in progress 10: LDO off in progress 11: LDO on
[25:24]	ULDOPST	Ultra-low power regulator Progress Status 00: ULDO off 01: ULDO on in progress 10: ULDO off in progress 11: ULDO on

## 6 Clock Control Unit (CKCU)

### Introduction

The Clock Control unit (CKCU) provides functions of high speed internal RC oscillator (HSI), High speed external crystal oscillator (HSE), Low speed internal RC oscillator (LSI), Low speed external crystal oscillator (LSE), Phase Lock Loop (PLL), HSE clock monitor, clock prescaler, clock multiplexer and clock gating. The clock of AHB, APB, and CPU are derived from system clock (CK\_SYS) which can come from HSI, HSE, LSI, LSE or PLL. Watchdog Timer and Real Time Clock (RTC) use either LSI or LSE as their clock source. The maximum operating frequency of system clock  $f_{CK\_AHB}$  can be up to 72 MHz.

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



**Figure 13. CKCU Block Diagram**



## Features

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

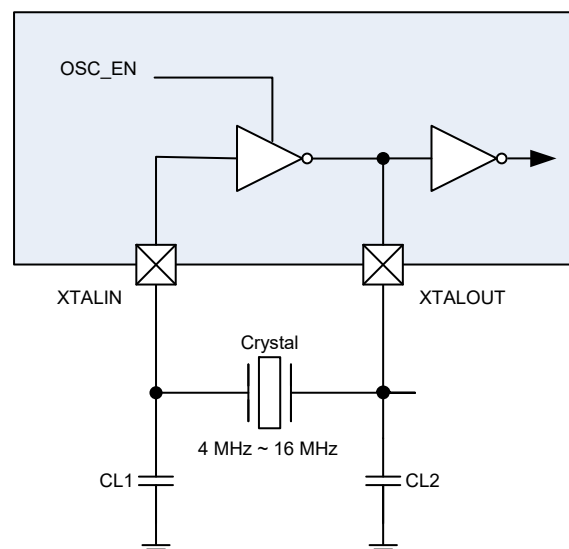
## Function Descriptions

### High Speed External Crystal Oscillator – HSE

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

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

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



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

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

## High Speed Internal RC Oscillator – HSI

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

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

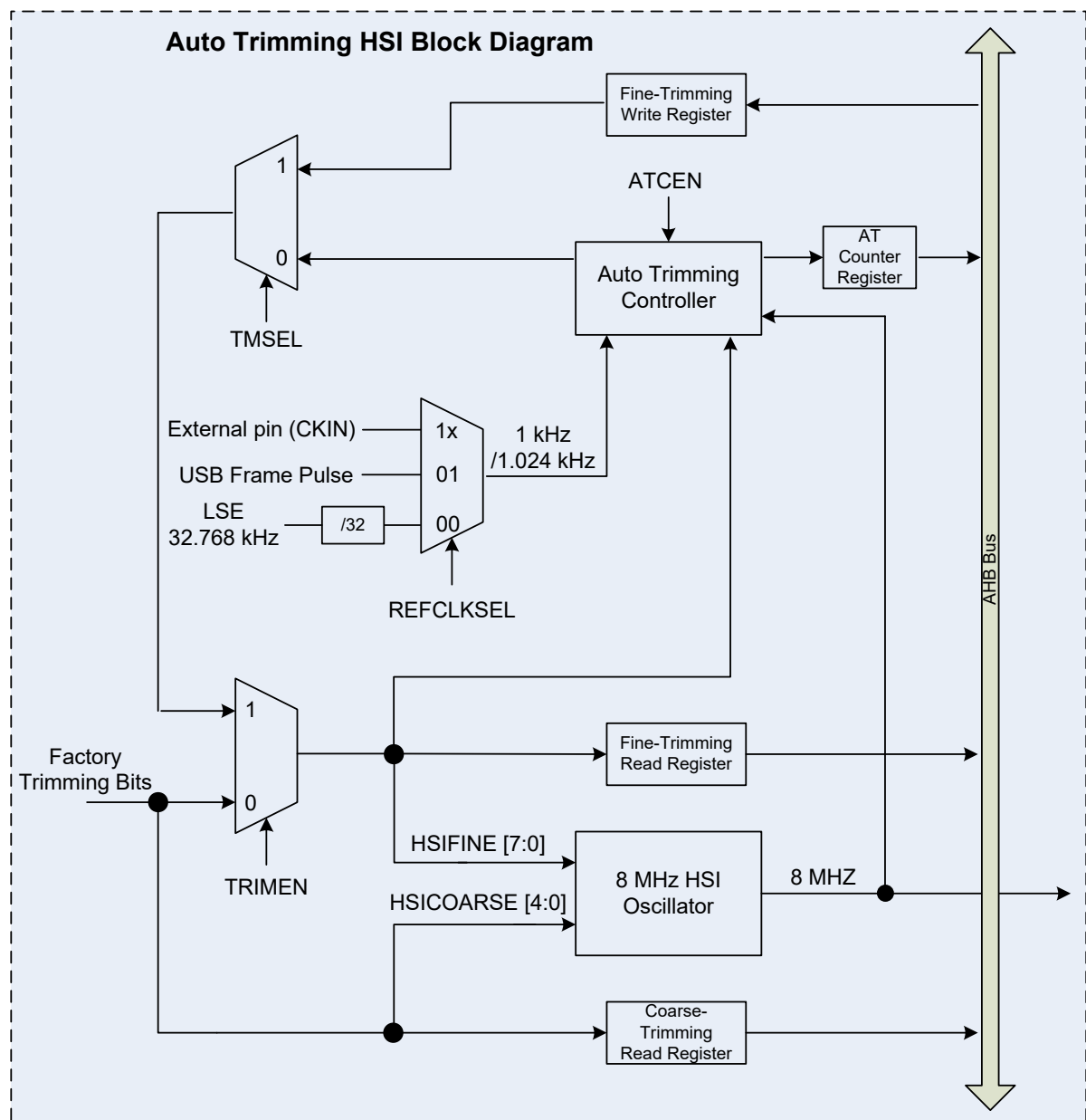
Software could configure the Power Saving Wakeup RC Clock Enable bit PSRCEN to 1 to force HSI clock to be system clock when the system initially wake-up. Subsequently, the system clock is back to the original clock source if the original clock source ready flag is asserted. This function can reduce the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

## Auto Trimming of High Speed Internal RC Oscillator – HSI

The frequency accuracy of the high speed internal RC oscillator HSI can vary from one chip to another due to manufacturing process variations, this is why each device is factory calibrated for  $\pm 1.5\%$  accuracy at  $V_{DD} = 3.3\text{ V}$  and  $T_A = 25\text{ }^{\circ}\text{C}$ . But the accuracy is not enough for some applications and environments requirement. Therefore, this device provides the trimming mechanism for HSI frequency calibration using more accurate external reference clock. The detail block diagram is shown as the following figure.

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

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



**Figure 15. HSI Auto Trimming Block Diagram**

## Phase Locked Loop – PLL

This PLL can provide 8 ~ 80 MHz clock output which is 2 ~ 32 multiples of a fundamental reference frequency of 4 ~ 16 MHz. The rationale of the clock synthesizer relies on the digital Phase Locked Loop (PLL) which includes a reference divider, a feedback divider, a digital phase frequency detector (PFD), a current-controlled charge pump, a built-in loop filter and a voltage-controlled oscillator (VCO) to achieve a stable phase-locked state.

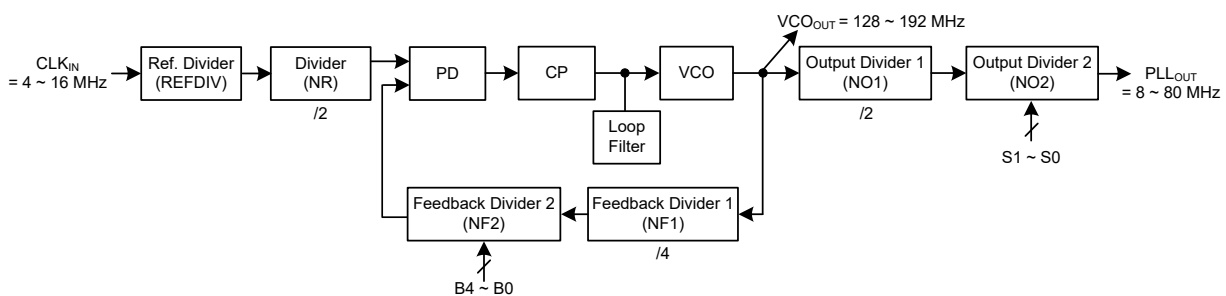


Figure 16. PLL Block Diagram

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

$$\begin{aligned} \text{PLL}_{\text{OUT}} &= \text{CLK}_{\text{IN}} \times \frac{\text{NF1} \times \text{NF2}}{\text{REFDIV} \times \text{NR} \times \text{NO1} \times \text{NO2}} = \text{CLK}_{\text{IN}} \times \frac{4 \times \text{NF2}}{\text{REFDIV} \times 2 \times 2 \times \text{NO2}} \\ &= \text{CLK}_{\text{IN}} \times \frac{\text{NF2}}{\text{REFDIV} \times \text{NO2}} \end{aligned}$$

where REFDIV = 1 or 2, NR = 2, NF1 = Feedback Divider 1 = 4, NF2 = Feedback Divider 2 = 1 ~ 32, NO1 = Output Divider 1 = 2, NO2 = Output Divider 2 = 1, 2, 4, or 8

Considering the duty cycle with 50%, both input frequency and output frequency is divided by 2. Assume that a given CLK<sub>in</sub> frequency as the PLL input generates a specific PLL output frequency; it is recommended to load a larger value into the NF2 field to increase the PLL stability and reduce the jitter with the expense of the settling time. The output and feedback divider 2 setup value are described in Table 15 and Table 16. All the configuration bits (S1 ~ S0, B4 ~ B0) in Table 15 and Table 16 are defined in the PLL Configuration Register (PLLCFGR) and PLL Control Register (PLLCR) in the section of Register Definition. Note that VCO<sub>OUT</sub> frequency should be in the range from 128 MHz to 192 MHz. If the selected configuration exceeds this range, the PLL output frequency cannot be guaranteed to match the above PLL<sub>OUT</sub> formula.

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

**Table 15. Output Divider 2 Value Mapping**

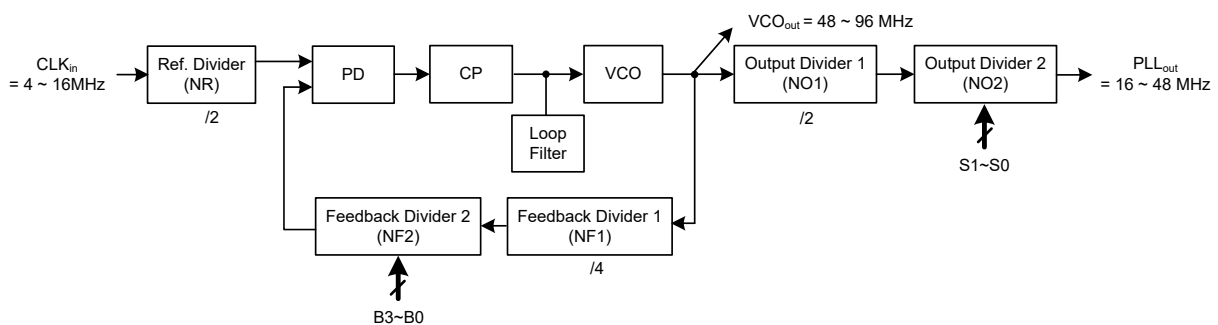
Output Divider 2 Setup Bits S1 ~ S0 (POTD Field in the PLLCFGR Register)	NO2 (Output Divider 2 Value)
00	1
01	2
10	4
11	8

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

Feedback Divider 2 Setup Bits B4 ~ B0 (PFBD Field in the PLLCFGR Register)	NF2 (Feedback Divider 2 Value)
00000	32
00001	1
00010	2
00011	3
00100	4
00101	5
00110	6
00111	7
01000	8
01001	9
01010	10
01011	11
⋮	⋮
⋮	⋮
11110	30
11111	31

## USB Phase Locked Loop – USB PLL

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



**Figure 17. USB PLL Block Diagram**

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

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

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

Considering the duty cycle with 50%, both input frequency and output frequency is divided by 2. Assume that a given  $CLK_{in}$  frequency as USB PLL input generates a specific USB PLL output frequency; it is recommended to load a larger value into the NF2 field to increase the PLL stability and reduce the jitter with the expense of the settling time. The output and feedback divider 2 value are described in Table 17 and Table 18. All the configuration bits (S1 ~ S0, B3 ~ B0) in Table 17 and Table 18 are defined in the PLL Configuration Register (PLLCFGR) and PLL Control Register (PLLCR) in the section of Register Definition. Note that  $VCO_{OUT}$  is ranged from 48 MHz to 96 MHz. If your configurations exceed this range, the output frequency of USB PLL will not be promised to match the above  $PLL_{OUT}$  formula.

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

**Table 17. USB PLL Output Divider 2 Value Mapping**

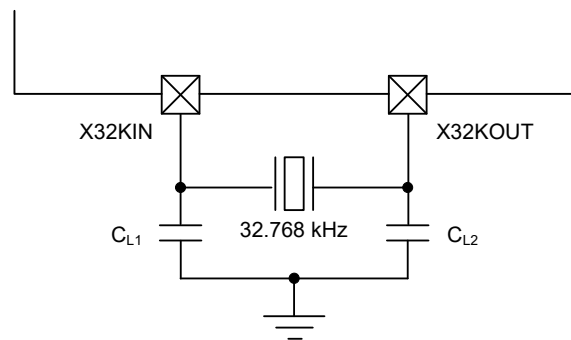
Output Divider 2 Setup Bits S1 ~ S0 (USBPOTD Field in the PLLCFGR Register)	NO2 (Output Divider 2 Value)
00	1
01	2
10	4
11	8

**Table 18. USB PLL Feedback Divider 2 Value Mapping**

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

## Low Speed External Crystal Oscillator – LSE

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



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

## Low Speed Internal RC Oscillator – LSI

The low speed internal RC oscillator with frequency of about 32 kHz produces a low power clock source for the circuits of Real-Time-Clock peripheral, Watchdog Timer or system clock. The LSI offers a low clock source because no external component is required to make it oscillates. The LSI frequency accuracy is shown in the Datasheet. The LSIRDY flag in the Global Clock Status Register (GCSR) will indicate if the LSI clock is stable. An interrupt can be generated if the related interrupt enable bit LSIRDYIE in the Global Clock Interrupt Register (GCIR) is set as the LSI becomes stable.

## Clock Ready Flag

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



## System Clock (CK\_SYS) Selection

After a system reset occurs, the high speed internal RC oscillator HSI is selected as the system clock (CK\_SYS). The CK\_SYS may come from the HSI, HSE, LSE, LSI or PLL output clock and it can be switched from one clock source to another via the System Clock Switch field, SW, in the Global Clock Control Register (GCCR). The system will still run under the original clock until the destination clock gets ready when the SW value is changed. The corresponding clock ready status bits in the Global Clock Status Register (GCSR) will indicate whether the selected clock is ready to use or not. The CKCU also contains the clock source status bits in the Clock Source Status Register CKST to indicate which clock is currently used as system clock. If a clock source or the PLL output clock is used as system clock, it is not possible to stop it. More details about clock enable function is described below.

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

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

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

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

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

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

The system clock selection programming guide is listed below.

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

## HSE Clock Monitor

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

## Clock Output Capability

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

**Table 19. CKOUT Clock Source**

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

## Register Map

The following table shows the CKCU register and reset value.

**Table 20. CKCU Register Map**

Register	Offset	Description	Reset Value
GCFGR	0x000	Global Clock Configuration Register	0x0000_0302
GCCR	0x004	Global Clock Control Register	0x0000_0803
GCSR	0x008	Global Clock Status Register	0x0000_0028
GCIR	0x00C	Global Clock Interrupt Register	0x0000_0000
PLLCFGR	0x018	PLL Configuration Register	0x0000_0000
PLLCR	0x01C	PLL Control Register	0x0000_0000
AHBCFGR	0x020	AHB Configuration Register	0x0000_0000
AHBCCR	0x024	AHB Clock Control Register	0x0000_00E5
APBCFGR	0x028	APB Configuration Register	0x0000_0000
APBCCR0	0x02C	APB Clock Control Register 0	0x0000_0000
APBCCR1	0x030	APB Clock Control Register 1	0x0000_0000
CKST	0x034	Clock Source Status Register	0x0100_0003
APBPCR0	0x038	APB Peripheral Clock Selection Register 0	0x0000_0000
APBPCR1	0x03C	APB Peripheral Clock Selection Register 1	0x0000_0000
HSICR	0x040	HSI Control Register	0xFFFF_0000 where X is undefined
HSIATCR	0x044	HSI Auto Trimming Counter Register	0x0000_0000
APBPCR2	0x048	APB Peripheral Clock Selection Register 2	0x0000_0000
LPCR	0x300	Low Power Control Register	0x0000_0000
MCUDBGCR	0x304	MCU Debug Control Register	0x0000_0000

## Register Descriptions

### Global Clock Configuration Register – GCFGR

This register specifies the clock source for PLL/USB/CKOUT.

Offset: 0x000

Reset value: 0x0000\_0302

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

Bits	Field	Descriptions
[31:29]	LPMOD	Lower Power Mode Status 000: When chip is in running mode 001: When chip once entered Sleep mode 010: When chip once entered Deep-Sleep1 mode 011: When chip once entered Deep-Sleep2 mode 100: When chip once entered Power-Down mode Others: Reserved Set by hardware. Reset by software writing b110.
[23:22]	USBPRE	USB Clock Prescaler Selection 00: CK_USB = CK_PLL 01: CK_USB = CK_PLL / 2 Others: Reserved Set and reset by software to control USB clock prescaler setting.
[15:11]	CKREFPRE	CK_REF Clock Prescaler Selection $CK\_REF = CK\_PLL / (CKREFPRE + 1) / 2$ 00000: CK_REF = CK_PLL / 2 00001: CK_REF = CK_PLL / 4 ... 11111: CK_REF = CK_PLL / 64 Set and reset by software to control CK_REF clock prescaler setting.
[10]	USBSRC	USB Clock Source Selection 0: CK_PLL clock is selected 1: CK_USBPLL clock is selected Set and reset by software to control USB clock source.
[9]	USBPLLSRC	USB PLL Clock Source Selection 0: External 4 ~ 16 MHz crystal oscillator clock is selected (HSE) 1: Internal 8 MHz RC oscillator clock is selected (HSI) Set and reset by software to control USB PLL clock source.

Bits	Field	Descriptions
[8]	PLLSRC	PLL Clock Source Selection 0: External 4 ~ 16 MHz crystal oscillator clock is selected (HSE) 1: Internal 8 MHz RC oscillator clock is selected (HSI) Set and reset by software to control PLL clock source.
[2:0]	CKOUTSRC	CKOUT Clock Source Selection 000: CK_REF is selected – Where $CK\_REF = CK\_PLL / (CKREFPRE + 1) / 2$ 001: (HCLKC / 16) is selected 010: (CK_SYS / 16) is selected 011: (CK_HSE / 16) is selected 100: (CK_HSI / 16) is selected 101: CK_LSE is selected 110: CK_LSI is selected 111: Reserved Set and reset by software.

## Global Clock Control Register – GCCR

This register specifies the clock enable bits.

Offset: 0x004

Reset value: 0x0000\_0803

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

Bits	Field	Descriptions
[17]	PSRCEN	Power Saving Wakeup RC Clock Enable 0: No action 1: Use Internal 8 MHz RC clock (HSI) as system clock after Deep-Sleep1/2 wakeup  The software can set the PSRCEN bit high before entering the Deep-Sleep1 or Deep-Sleep2 mode. In order to reduce the waiting time after a wakeup. When the PSRCEN bit is set to 1, the HSI will be used as the CK_SYS clock source after waking up from the Deep-Sleep1 or Deep-Sleep2 mode. This means that the instruction can be executed before the original CK_SYS source is stable since the HSI clock is provided to CPU. After the original clock source is ready, the CK_SYS clock will automatically be switched back to the original.

Bits	Field	Descriptions
[16]	CKMEN	<p>HSE Clock Monitor Enable</p> <p>0: Disable External 4 ~ 16 MHz crystal oscillator clock monitor</p> <p>1: Enable External 4 ~ 16 MHz crystal oscillator clock monitor</p> <p>When the hardware detects that the HSE clock stuck at a low or high state, the internal hardware will switch the system clock to the internal high speed HSI RC clock.</p>
[11]	HSIEN	<p>Internal High Speed Clock Enable</p> <p>0: Internal 8 MHz RC oscillator clock is disabled</p> <p>1: Internal 8 MHz RC oscillator clock is enabled</p> <p>Set and reset by software. This bit cannot be reset if HSI clock is used as system clock.</p>
[10]	HSEEN	<p>External High Speed Clock Enable</p> <p>0: External 4 ~ 16 MHz crystal oscillator clock is disabled</p> <p>1: External 4 ~ 16 MHz crystal oscillator clock is enabled</p> <p>Set and reset by software. This bit cannot be reset if the HSE clock is used as system clock or the PLL input clock.</p>
[9]	PLLEN	<p>PLL Enable</p> <p>0: PLL is disabled</p> <p>1: PLL is enabled</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>
[8]	HSEGAIN	<p>External High Speed Clock Gain Selection</p> <p>0: HSE is in low gain mode</p> <p>1: HSE is in high gain mode</p>
[3]	USBPLLEN	<p>USB PLL Enable</p> <p>0: USB PLL is disabled</p> <p>1: USB PLL is enabled</p> <p>Set and reset by software to enable USB PLL. This bit cannot be reset if the PLL clock is used as system clock.</p>
[2:0]	SW	<p>System Clock Switch</p> <p>00x: CK_PLL clock out as system clock</p> <p>010: CK_HSE as system clock</p> <p>011: CK_HSI as system clock</p> <p>110: CK_LSE as system clock</p> <p>111: CK_LSI as system clock</p> <p>Others: CK_HSI as system clock</p> <p>This bit field is set and reset by software to select the CK_SYS source. The HSI oscillator will be forced as the system clock when the HSE oscillator clock failure is detected, where the HSE is used directly or indirectly as system clock, as the clock monitor is enabled.</p> <p>Note: When switching the system clock using the SW bits, the system clock will not be immediately switched and a certain delay is necessary. The system clock source selected by the SW bits can be indicated in the CKSWST bits in the clock source status register CKST to make sure which clock is currently used as the system clock.</p>

## Global Clock Status Register – GCSR

This register indicates the clock ready status.

Offset: 0x008

Reset value: 0x0000\_0028

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

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

## Global Clock Interrupt Register – GCIR

This register specifies interrupt enable and flag bits.

Offset: 0x00C

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[22]	LSIRDYIE	LSI Ready Interrupt Enable 0: Disable LSI stabilization interrupt 1: Enable LSI stabilization interrupt Set and reset by software to enable/disable interrupt caused by LSI stabilization.
[21]	LSERDYIE	LSE Ready Interrupt Enable 0: Disable LSE stabilization interrupt 1: Enable LSE stabilization interrupt Set and reset by software to enable/disable interrupt caused by LSE stabilization.
[20]	HSIRDYIE	HSI Ready Interrupt Enable 0: Disable HSI stabilization interrupt 1: Enable HSI stabilization interrupt Set and reset by software to enable/disable interrupt caused by HSI stabilization.
[19]	HSERDYIE	HSE Ready Interrupt Enable 0: Disable HSE stabilization interrupt 1: Enable HSE stabilization interrupt Set and reset by software to enable/disable interrupt caused by HSE stabilization.
[18]	PLLRDYIE	PLL Ready Interrupt Enable 0: Disable PLL stabilization interrupt 1: Enable PLL stabilization interrupt Set and reset by software to enable/disable interrupt caused by PLL stabilization.
[17]	USBPLLRDYIE	USB PLL Ready Interrupt Enable 0: Disable USB PLL stabilization interrupt 1: Enable USB PLL stabilization interrupt Set and reset by software to enable/disable interrupt caused by USB PLL stabilization.
[16]	CKSIE	Clock Stuck Interrupt Enable 0: Disable clock failure interrupt 1: Enable clock failure interrupt Set and reset by software to enable or disable the clock failure interrupt caused by clock monitor.



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

## PLL Configuration Register – PLLCFGR

This register specifies the PLL configurations.

Offset: 0x018

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[28]	REFDIV	PLL Input Reference Clock Divider 0: Reference divider NR = 2 1: Reference divider NR = 4
[27:23]	PFBD	PLL VCO Output Clock Feedback Divider (B4 ~ B0) Feedback Divider divides the output clock from VCO of PLL.
[22:21]	POTD	PLL Output Clock Divider (S1 ~ S0)
[10:7]	USBPFBD	USB PLL VCO Output Clock Feedback Divider (B3 ~ B0) Feedback Divider divides the output clock from VCO of PLL.
[6:5]	USBPOTD	USB PLL Output Clock Divider (S1 ~ S0)

## PLL Control Register – PLLCR

This register specifies the PLL Bypass mode.

Offset: 0x01C

Reset value: 0x0000\_0000

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

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

## AHB Configuration Register – AHBCFGR

This register specifies the system clock frequency.

Offset: 0x020

Reset value: 0x0000\_0000

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

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

## AHB Clock Control Register – AHBCCR

This register specifies the AHB clock enable bits.

Offset: 0x024

Reset value: 0x0000\_00E5

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

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

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

## APB Configuration Register – APBCFGR

This register specifies the ADC conversion clock frequency.

Offset: 0x028

Reset value: 0x0001\_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved					ADCDIV		
	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
[18:16]	ADCDIV	ADC Clock Frequency Divide Selection 000: CK_ADC = (CK_AHB / 1) 001: CK_ADC = (CK_AHB / 2) 010: CK_ADC = (CK_AHB / 4) 011: CK_ADC = (CK_AHB / 8) 100: CK_ADC = (CK_AHB / 16) 101: CK_ADC = (CK_AHB / 32) 110: CK_ADC = (CK_AHB / 64) 111: CK_ADC = (CK_AHB / 6) Set and reset by software to control ADC conversion clock division factor.

## APB Clock Control Register 0 – APBCCR0

This register specifies the APB peripherals clock enable bits.

Offset: 0x02C

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	Reserved							SCIEN	
Type/Reset								RW	0
	23	22	21	20	19	18	17	16	
	Reserved								
Type/Reset									
	15	14	13	12	11	10	9	8	
	EXTIEN	AFIOEN	Reserved		UR1EN	UR0EN	Reserved	USREN	
Type/Reset	RW	0	RW	0		RW	0	RW	0
	7	6	5	4	3	2	1	0	
	Reserved		SPI1EN	SPI0EN	Reserved		I2C1EN	I2C0EN	
Type/Reset			RW	0	RW	0		RW	0

Bits	Field	Descriptions
[24]	SCIEN	Smart Card Interface Clock Enable 0: SCI clock is disabled 1: SCI clock is enabled Set and reset by software.
[15]	EXTIEN	External Interrupt Clock Enable 0: EXTI clock is disabled 1: EXTI clock is enabled Set and reset by software.
[14]	AFIOEN	Alternate Function I/O Clock Enable 0: AFIO clock is disabled 1: AFIO clock is enabled Set and reset by software.
[11]	UR1EN	UART1 Clock Enable 0: UART1 clock is disabled 1: UART1 clock is enabled Set and reset by software.
[10]	UR0EN	UART0 Clock Enable 0: UART0 clock is disabled 1: UART0 clock is enabled Set and reset by software.
[8]	USREN	USART Clock Enable 0: USART clock is disabled 1: USART clock is enabled Set and reset by software.
[5]	SPI1EN	SPI1 Clock Enable 0: SPI1 clock is disabled 1: SPI1 clock is enabled Set and reset by software.



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

### APB Clock Control Register 1 – APBCCR1

This register specifies the APB peripherals clock enable bits.

Offset: 0x030

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[29]	SCTM1EN	SCTM1 Clock Enable 0: SCTM1 clock is disabled 1: SCTM1 clock is enabled Set and reset by software.
[28]	SCTM0EN	SCTM0 Clock Enable 0: SCTM0 clock is disabled 1: SCTM0 clock is enabled Set and reset by software.
[24]	ADCCEN	ADC Controller Clock Enable 0: ADC clock is disabled 1: ADC clock is enabled Set and reset by software.
[17]	BFTM1EN	BFTM1 Clock Enable 0: BFTM1 clock is disabled 1: BFTM1 clock is enabled Set and reset by software.

Bits	Field	Descriptions
[16]	BFTM0EN	BFTM0 Clock Enable 0: BFTM0 clock is disabled 1: BFTM0 clock is enabled Set and reset by software.
[12]	PWMEN	PWM Clock Enable 0: PWM clock is disabled 1: PWM clock is enabled Set and reset by software.
[8]	GPTMEN	GPTM Clock Enable 0: GPTM clock is disabled 1: GPTM clock is enabled Set and reset by software.
[6]	VDDREN	V <sub>DD</sub> Domain Clock Enable for Register Access 0: Register access clock is disabled 1: Register access clock is enabled Set and reset by software.
[4]	WDTREN	Watchdog Timer Clock Enable for Register Access 0: Watchdog Timer clock is disabled 1: Watchdog Timer clock is enabled Set and reset by software.

### Clock Source Status Register – CKST

This register specifies clock source status.

Offset: 0x034

Reset value: 0x0100\_0003

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

Bits	Field	Descriptions
[27:24]	HSIST	Internal High Speed Clock Occupation Status (CK_HSI) xxx1: HSI is used by System Clock (CK_SYS) (SW = 0x3) xx1x: HSI is used by PLL x1xx: HSI is used by Clock Monitor 1xxx: HSI is used by USB PLL

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

## APB Peripheral Clock Selection Register 0 – APBPCSR0

This register specifies the APB peripheral clock prescaler selection.

Offset: 0x038

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[31:30]	UR1PCLK	UART1 Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[29:28]	UR0PCLK	UART0 Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock

Bits	Field	Descriptions
[25:24]	USRPCLK	USART Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[21:20]	GPTMPCLK	GPTM Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[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				SCTM1PCLK		SCTM0PCLK	
Type/Reset					RW	0	RW	0
	23	22	21	20	19	18	17	16
	Reserved						SCIPCLK	
Type/Reset							RW	0
	15	14	13	12	11	10	9	8
	VDDRCLK		WDTRPCLK		Reserved			
Type/Reset	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
	Reserved		ADCCPCLK		EXTIPCLK		AFIOPCLK	
Type/Reset			RW	0	RW	0	RW	0

Bits	Field	Descriptions
[27:26]	SCTM1PCLK	SCTM1 Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[25:24]	SCTM0PCLK	SCTM0 Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[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]	VDDRCLK	VDD Domain Register Access Clock Selection 00: PCLK = CK_AHB / 4 01: PCLK = CK_AHB / 8 10: PCLK = CK_AHB / 16 11: PCLK = CK_AHB / 32 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[13:12]	WDTRPCLK	WDT Register Access Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock

Bits	Field	Descriptions
[5:4]	ADCCPCLK	ADC Controller Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[3:2]	EXTIPCLK	EXTI Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[1:0]	AFIOPCLK	AFIO Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock

## HSI Control Register – HSICR

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

Offset: 0x040

Reset value: 0xFFFF\_0000 where X is undefined

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

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

Bits	Field	Descriptions
[7]	FLOCK	Frequency Lock 0: HSI frequency is not trimmed into target range 1: HSI frequency is trimmed into target range
[6:5]	REFCLKSEL	Reference Clock Selection 00: Select 32.768 kHz external low speed clock source (LSE) 01: Select 1 kHz USB frame pulse 1x: Select external pin (CKIN) 1 kHz pulse This bit is used to select the reference clock for the HSI Auto Trimming Controller.
[4]	TMSEL	Trimming Mode Selection 0: Automatic by Auto Trimming Controller 1: Manual by user program This bit is used to select the HSI RC oscillator trimming function by the ATC hardware or user program via the HSIFINE[7:0] bits in the HSI Control Register.
[3]	ATMSEL	Automatic Trimming Mode Selection 0: Auto Trimming Controller is used binary search to approach the target range 1: Auto Trimming Controller is used linear search to approach the target range This bit is used to select the automatic trimming method by ATC hardware for the HSI RC oscillator.
[2]	LTRSEL	Lock Target Range Selection 0: 0.1 % variation 1: 0.2 % variation This bit is used to select the lock target range of the internal HSI RC oscillator trimming function for 0.1 % or 0.2 % variation.
[1]	ATCEN	Auto Trimming Controller Enable 0: Disable Auto Trimming Controller 1: Enable Auto Trimming Controller
[0]	TRIMEN	Trimming Enable 0: HSI Trimming is disabled 1: HSI Trimming is enabled The bit enables the HSI RC oscillator trimming function by the ATC hardware or user program.

## HSI Auto Trimming Counter Register – HSIATCR

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

Offset: 0x044

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
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		ATCNT						
	7	6	5	4	3	2	1	0	
Type/Reset	ATCNT								
	RO	0	RO	0	RO	0	RO	0	RO
	0	RO	0	RO	0	RO	0	RO	0

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



## APB Peripheral Clock Selection Register 2 – APBPCSR2

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

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						PWMPCLK	
	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
[17:16]	PWMPCLK	PWM Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock.

## Low Power Control Register – LPCR

This register specifies the low power control.

Offset: 0x300

Reset value: 0x0000\_0000

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

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

## MCU Debug Control Register – MCUDBGCR

This register specifies the MCU debug control.

Offset: 0x304

Reset value: 0x0000\_0000 (Reset by VDD domain reset only)

	31	30	29	28	27	26	25	24
	Reserved	DBPWM	Reserved					
Type/Reset		RW 0						
	23	22	21	20	19	18	17	16
	DBSCTM1	DBSCTM0	Reserved	DBTRACE	DBUR1	DBUR0	DBBFTM1	DBBFTM0
Type/Reset	RW 0	RW 0		RW 0	RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	DBSCI	DBDSLP2	DBI2C1	DBI2C0	DBSPI1	DBSPI0	Reserved	DBUSR
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0		RW 0
	7	6	5	4	3	2	1	0
	Reserved	DBGPTM	Reserved		DBWDT	DBPD	DBDSLP1	DBSLP
Type/Reset		RW 0			RW 0	RW 0	RW 0	RW 0

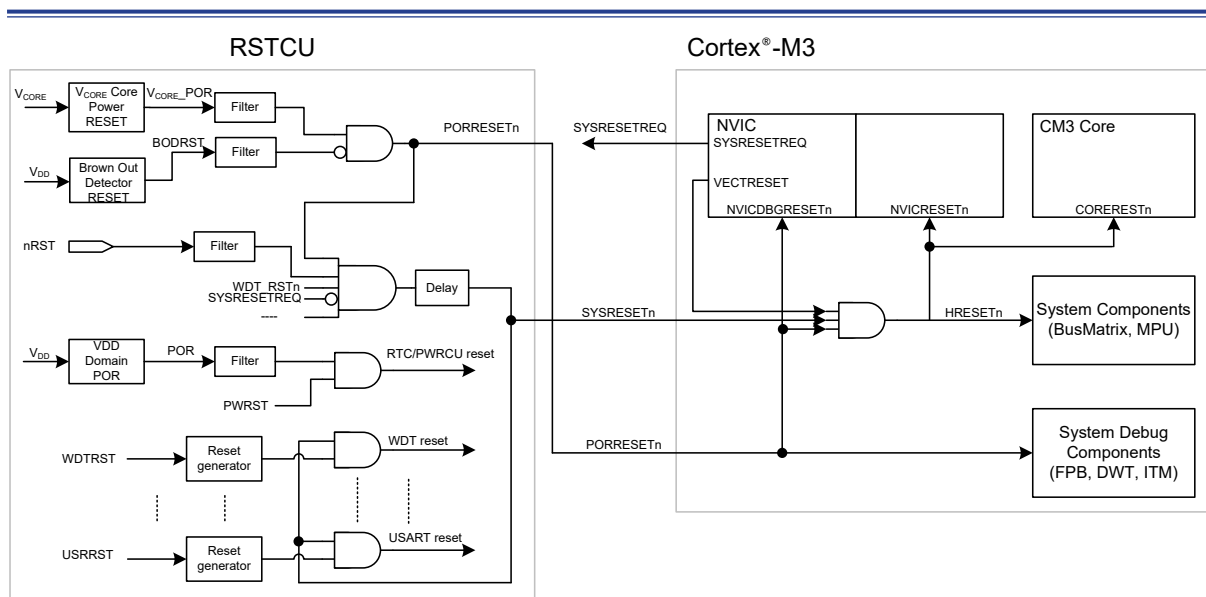
Bits	Field	Descriptions
[30]	DBPWM	PWM Debug Mode Enable 0: PWM counter continues to count even if the core is halted 1: PWM counter is stopped when the core is halted Set and reset by software.
[23]	DBSCTM1	SCTM1 Debug Mode Enable 0: SCTM1 counter continues to count even if the core is halted 1: SCTM1 counter is stopped when the core is halted Set and reset by software.
[22]	DBSCTM0	SCTM0 Debug Mode Enable 0: SCTM0 counter continues to count even if the core is halted 1: SCTM0 counter is stopped when the core is halted Set and reset by software.
[20]	DBTRACE	TRACESWO Debug Mode Enable 0: Disable TRACESWO output 1: Enable TRACESWO output Set and reset by software.
[19]	DBUR1	UART1 Debug Mode Enable 0: Same behavior as in normal mode 1: UART1 FIFO timeout is frozen when the core is halted Set and reset by software.
[18]	DBUR0	UART0 Debug Mode Enable 0: Same behavior as in normal mode 1: UART0 FIFO timeout is frozen when the core is halted Set and reset by software.
[17]	DBBFTM1	BFTM1 Debug Mode Enable 0: BFTM1 counter continues to count even if the core is halted 1: BFTM1 counter is stopped when the core is halted Set and reset by software.

Bits	Field	Descriptions
[16]	DBBFTM0	BFTM0 Debug Mode Enable 0: BFTM0 counter continues to count even if the core is halted 1: BFTM0 counter is stopped when the core is halted Set and reset by software.
[15]	DBSCI	SCI Debug Mode Enable 0: Same behavior as in normal mode 1: SCI timeout is frozen Set and reset by software.
[14]	DBDSLP2	Debug Deep-Sleep2 0: LDO = Off (but turn on DMOS), FCLK = Off and CM3EN = Off in Deep-Sleep2 1: LDO = On, FCLK = On and CM3EN = On in Deep-Sleep2 Set and reset by software.
[13]	DBI2C1	I <sup>2</sup> C1 Debug Mode Enable 0: Same behavior as in normal mode 1: I <sup>2</sup> C1 timeout is frozen when the core is halted Set and reset by software.
[12]	DBI2C0	I <sup>2</sup> C0 Debug Mode Enable 0: Same behavior as in normal mode 1: I <sup>2</sup> C0 timeout is frozen when the core is halted Set and reset by software.
[11]	DBSPI1	SPI1 Debug Mode Enable 0: Same behavior as in normal mode 1: SPI1 FIFO timeout is frozen when the core is halted Set and reset by software.
[10]	DBSPI0	SPI0 Debug Mode Enable 0: Same behavior as in normal mode 1: SPI0 FIFO timeout is frozen when the core is halted Set and reset by software.
[8]	DBUSR	USART Debug Mode Enable 0: Same behavior as in normal mode 1: USART FIFO timeout is frozen when the core is halted Set and reset by software.
[6]	DBGPTM	GPTM Debug Mode Enable 0: GPTM counter continues to count even if the core is halted 1: GPTM counter is stopped when the core is halted Set and reset by software.
[3]	DBWDT	Watchdog Timer Debug Mode Enable 0: Watchdog Timer counter continues to count even if the core is halted 1: Watchdog Timer counter is stopped when the core is halted Set and reset by software.
[2]	DBPD	Debug Power-Down Mode 0: LDO = Off, FCLK = Off and CM3EN = Off in Power-Down mode 1: LDO = On, FCLK = On and CM3EN = On in Power-Down mode Set and reset by software.
[1]	DBDSLP1	Debug Deep-Sleep1 0: LDO = Low power mode, FCLK = Off and CM3EN = Off in Deep-Sleep1 1: LDO = On, FCLK = On and CM3EN = On in Deep-Sleep1 Set and reset by software.
[0]	DBSLP	Debug Sleep Mode 0: LDO = On, FCLK = On and CM3EN = Off in Sleep mode 1: LDO = On, FCLK = On and CM3EN = On in Sleep mode Set and reset by software.

# 7 Reset Control Unit (RSTCU)

## Introduction

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

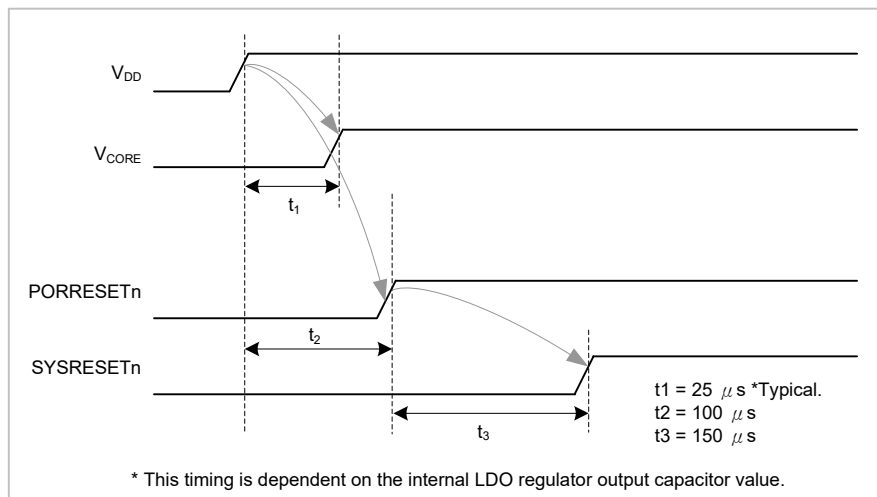


**Figure 19. RSTCU Block Diagram**

## Functional Descriptions

### Power On Reset

The Power on reset, POR, is generated by either an external reset or the internal reset generator. Both types have an internal filter to prevent glitches from causing erroneous reset operations. By referring to Figure 20, the  $V_{CORE\_POR}$  active low signal will be de-asserted when the internal LDO voltage regulator is ready to provide  $V_{CORE}$  power. In addition to the  $V_{CORE\_POR}$  signal, the Power Control Unit, PWRCU, will assert the BODF signal as a Power Down Reset, PDR, when the BODEN bit in the LVDCSR register is set and the brown-out event occurs. For more details about the PWRCU function, refer to the PWRCU chapter.



**Figure 20. Power On Reset Sequence**

### System Reset

A system reset is generated by a power on reset (PORRESETn), a Watchdog Timer reset (WDT\_RSTn), nRST pin or a software reset (SYSRESETREQ) event. For more information about SYSRESETREQ 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 USART reset via the USRRST bit in the APBPRSTR0 register.

## Register Map

The following table shows the RSTCU registers and reset values.

**Table 21. RSTCU Register Map**

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

## Register Descriptions

### Global Reset Status Register – GRSR

This register specifies a variety of reset status conditions.

Offset: 0x100

Reset value: 0x0000\_0008

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

Bits	Field	Descriptions
[3]	PORSTF	Core Power On Reset Flag 0: No POR occurred 1: POR occurred This bit is set by hardware when a power on reset occurs and reset by writing 1 into it.
[2]	WDTRSTF	Watchdog Timer Reset Flag 0: No Watchdog Timer reset occurred 1: Watchdog Timer occurred This bit is set by hardware when a watchdog timer reset occurs and reset by writing 1 into it or by hardware when a power on reset occurs.
[1]	EXTRSTF	External Pin Reset Flag 0: No pin reset occurred 1: Pin reset occurred This bit is set by hardware when an external pin reset occurs and reset by writing 1 into it or by hardware when a power on reset occurs.
[0]	SYSRSTF	System Reset Flag 0: No NVIC asserting system reset occurred 1: NVIC asserting system reset occurred This bit is set by hardware when a system reset occurs and reset by writing 1 into it or by hardware when a power on reset occurs.

## AHB Peripheral Reset Register – AHBPRSTR

This register specifies several AHB peripheral software reset control bits.

Offset: 0x104

Reset value: 0x0000\_0000

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

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



Bits	Field	Descriptions
[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 peripheral software reset control bits.

Offset: 0x108

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[24]	SCIRST	Smart Card Interface Reset Control 0: No reset 1: Reset SCI This bit is set by software and cleared to 0 by hardware automatically.
[15]	EXTIRST	External Interrupt Controller Reset Control 0: No reset 1: Reset EXTI This bit is set by software and cleared to 0 by hardware automatically.
[14]	AFIORST	Alternate Function I/O Reset Control 0: No reset 1: Reset Alternate Function I/O This bit is set by software and cleared to 0 by hardware automatically.

Bits	Field	Descriptions
[11]	UR1RST	UART1 Reset Control 0: No reset 1: Reset UART1 This bit is set by software and cleared to 0 by hardware automatically.
[10]	UR0RST	UART0 Reset Control 0: No reset 1: Reset UART0 This bit is set by software and cleared to 0 by hardware automatically.
[8]	USRRST	USART Reset Control 0: No reset 1: Reset USART This bit is set by software and cleared to 0 by hardware automatically.
[5]	SPI1RST	SPI1 Reset Control 0: No reset 1: Reset SPI1 This bit is set by software and cleared to 0 by hardware automatically.
[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 peripheral software reset control bits.

Offset: 0x10C

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved		SCTM1RST	SCTM0RST	Reserved		ADCRST	
	RW		0	RW	0			RW
								0
	23	22	21	20	19	18	17	16
Type/Reset	Reserved						BFTM51RST	BFTM0RST
							RW	0
								RW
								0
	15	14	13	12	11	10	9	8
Type/Reset	Reserved			PWMRST				GPTMRST
				RW				RW
				0				0
	7	6	5	4	3	2	1	0
Type/Reset	Reserved			WDTRST	Reserved			
				RW				
				0				

Bits	Field	Descriptions
[29]	SCTM1RST	SCTM1 Reset Control 0: No reset 1: Reset SCTM1 This bit is set by software and cleared to 0 by hardware automatically.
[28]	SCTM0RST	SCTM0 Reset Control 0: No reset 1: Reset SCTM0 This bit is set by software and cleared to 0 by hardware automatically.
[24]	ADCRST	A/D Converter Reset Control 0: No reset 1: Reset A/D Converter This bit is set by software and cleared to 0 by hardware automatically.
[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.
[12]	PWMRST	PWM Reset Control 0: No reset 1: Reset PWM This bit is set by software and cleared to 0 by hardware automatically.

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

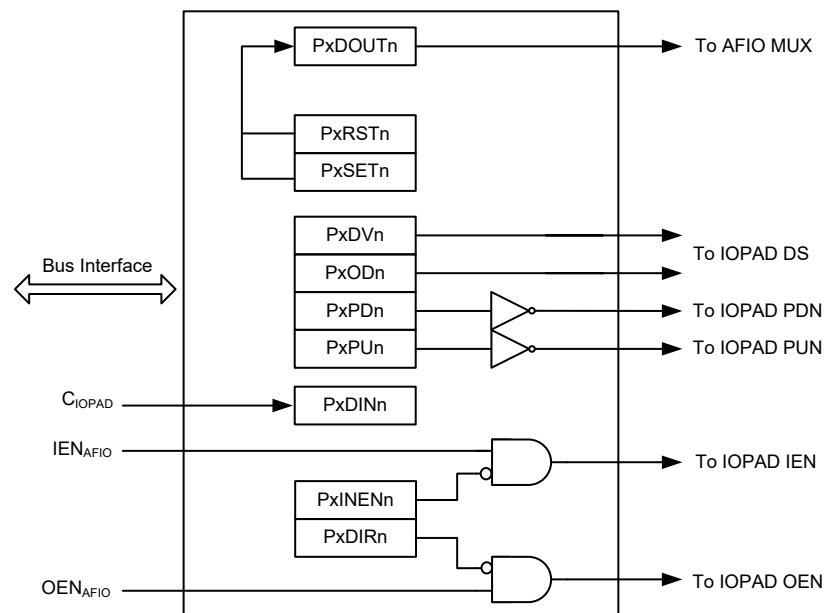
## 8 General Purpose I/O (GPIO)

### Introduction

There are up to 52 General Purpose I/O ports, GPIO, named PA0 ~ PA15, PB0 ~ PB15, PC0 ~ PC15, PD0 ~ PD2 and PF0 for the device to implement the logic input/output functions. Each of the GPIO ports has related control and configuration registers to satisfy the requirement of specific applications. The really available General Purpose I/O port numbers are dependent on the device specification and package type. Please refer the device data sheet for detail information.

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

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



**Figure 21. GPIO Block Diagram**

## Features

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

## Functional Descriptions

### Default GPIO Pin Configuration

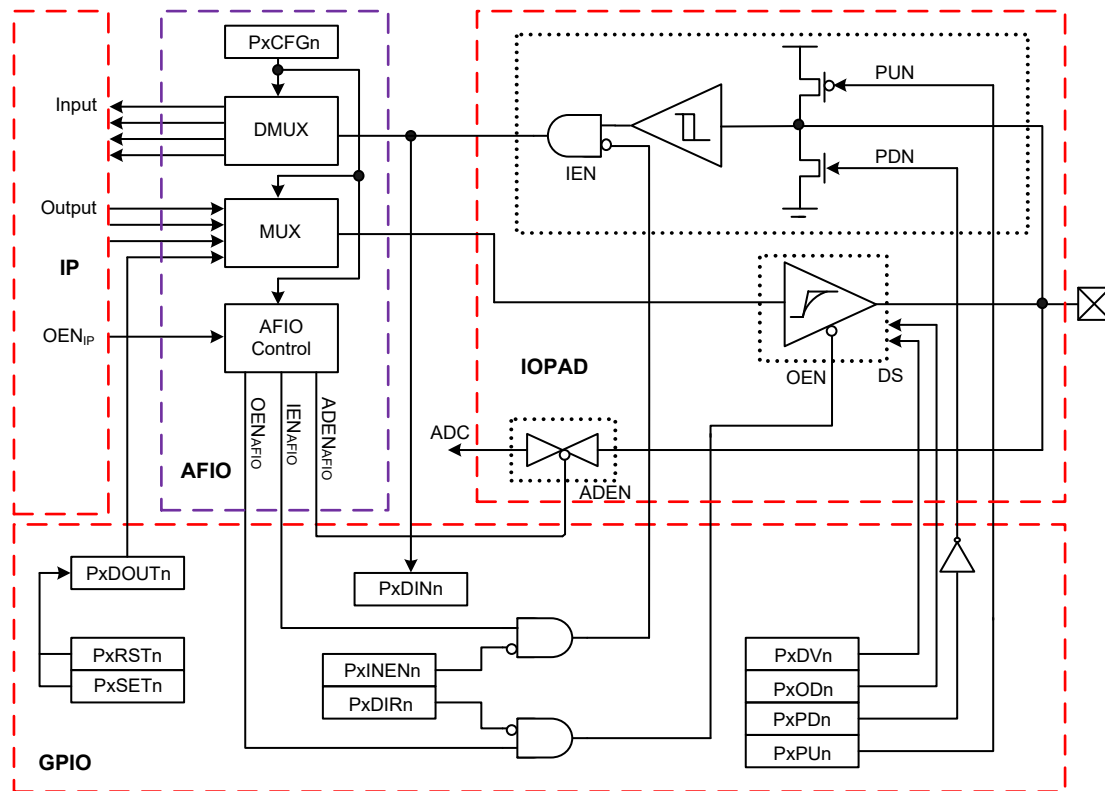
During or just after the reset period, the alternative functions are all inactive and the GPIO ports are configured into the input disable floating mode, i.e. input disabled without pull-up/pull-down resistors. Only the boot and Serial-Wired Debug pins which are pin-shared with the I/O pins are active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

- BOOT0: Input enable with internal pull-up
- BOOT1: Input enable with internal pull-up
- SWCLK: Input enable with internal pull-up
- SWDIO: Input enable with internal pull-up

### General Purpose I/O – GPIO

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

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



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

PxDINn/PxDOUTn (x = A~D, F): Data Input/Data Output

PxRSTn/PxSETn(x = A ~ D, F): Reset/Set

PxDIRn(x = A ~ D, F): Direction

PxDVn(x = A ~ D, F): Output Drive

PxPDn/PxPUn (x = A ~ D, F): Pull-Down/Up

PxINENn(x = A ~ D, F): Input Enable

PxODn(x = A ~ D, F): Open-Drain

PxCFGn(x = A ~ D, F): AFIO Configuration

**Table 22. AFIO, GPIO and I/O Pad Control Signal True Table**

Type	AFIO			GPIO		PAD		
	ADEN <sub>AFIO</sub>	OEN <sub>AFIO</sub>	IEN <sub>AFIO</sub>	PxDIRn	PxINENn	ADEN	OEN	IEN
GPIO Input <sup>(Note)</sup>	1	1	1	0	1	1	1	0
GPIO Output <sup>(Note)</sup>	1	1	1	1	0 (1 if need)	1	0	1 (0)
AFIO Input	1	1	0	0	X	1	1	0
AFIO Output	1	0	1	X	0 (1 if need)	1	0	1 (0)
ADC Input	0	1	1	0	0 (1 if need)	0	1	1 (0)
OSC Output	0	1	1	0	0 (1 if need)	0	1	1 (0)

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

## GPIO Locking Mechanism

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

**Table 23. 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 / 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
PBDRVR	0x014	Port B Drive Current Selection Register	0x0000_0000
PBLOCKR	0x018	Port B Lock Register	0x0000_0000
PBDINR	0x01C	Port B Data Input Register	0x0000_0000
PBDOUTR	0x020	Port B Data Output Register	0x0000_0000
PBSRR	0x024	Port B Output Set / Reset Control Register	0x0000_0000
PBRR	0x028	Port B Output Reset Control Register	0x0000_0000
<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
PCDRVR	0x014	Port C Drive Current Selection Register	0x0000_0000
PCLOCKR	0x018	Port C Lock Register	0x0000_0000
PCDINR	0x01C	Port C Data Input Register	0x0000_0000



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

## Register Descriptions

### Port A Data Direction Control Register – PADIRCR

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

Offset: 0x000

Reset value: 0x0000\_0000

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

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

## Port A Input Function Enable Control Register – PAINER

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

Offset: 0x004

Reset value: 0x0000\_0300

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

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

## Port A Pull-Up Selection Register – PAPUR

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

Offset: 0x008

Reset value: 0x0000\_3300

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

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

## Port A Pull-Down Selection Register – PAPDR

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

Offset: 0x00C

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[15:0]	PAPDn	GPIO Port A 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 A Open-Drain Selection Register – PAODR

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

Offset: 0x010

Reset value: 0x0000\_0000

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

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

## Port A Drive Current Selection Register – PADRVR

This register specifies the GPIO Port A output driving current.

Offset: 0x014

Reset value: 0x0000\_0000

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

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

## Port A Lock Register – PALOCKR

This register specifies the GPIO Port A lock configuration.

Offset: 0x018

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[31:16]	PALKEY	<p>GPIO Port A Lock Key</p> <p>0x5FA0: Port A Lock function is enabled</p> <p>Others: Port A Lock function is disabled</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 register 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

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

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

## Port A Output Data Register – PADOUTR

This register specifies the GPIO Port A output data.

Offset: 0x020

Reset value: 0x0000\_0000

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

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

## Port A Output Set/Reset Control Register – PASRR

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

Offset: 0x024

Reset value: 0x0000\_0000

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

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

## Port A Output Reset Register – PARR

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

Offset: 0x028

Reset value: 0x0000\_0000

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

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

## Port B Data Direction Control Register – PBDIRCR

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

Offset: 0x000

Reset value: 0x0000\_0000

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

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

## Port B Input Function Enable Control Register – PBINER

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

Offset: 0x004

Reset value: 0x0000\_0000

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

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

## Port B Pull-Up Selection Register – PBPUR

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

Offset: 0x008

Reset value: 0x0000\_0000

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

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

## Port B Pull-Down Selection Register – PBPDR

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

Offset: 0x00C

Reset value: 0x0000\_0000

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

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

## Port B Open-Drain Selection Register – PBODR

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

Offset: 0x010

Reset value: 0x0000\_0000

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

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

## Port B Drive Current Selection Register – PBDVR

This register specifies the GPIO Port B output driving current.

Offset: 0x014

Reset value: 0x0000\_0000

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

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



## Port B Lock Register – PBLOCKR

This register specifies the GPIO Port B lock configuration.

Offset: 0x018

Reset value: 0x0000\_0000

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

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

## Port B Data Input Register – PBDINR

This register specifies the GPIO Port B input data.

Offset: 0x01C

Reset value: 0x0000\_0000

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

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

## Port B Output Data Register – PBDOUTR

This register specifies the GPIO Port B output data.

Offset: 0x020

Reset value: 0x0000\_0000

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

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

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

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

Offset: 0x024

Reset value: 0x0000\_0000

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

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

## Port B Output Reset Register – PBRR

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

Offset: 0x028

Reset value: 0x0000\_0000

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

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

## Port C Data Direction Control Register – PCDIRCR

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

Offset: 0x000

Reset value: 0x0000\_0000

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

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

## Port C Input Function Enable Control Register – PCINER

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

Offset: 0x004

Reset value: 0x0000\_0000

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

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</p> <p>1: Pin n input function is enabled</p> <p>When the pin n input function is disabled, the input Schmitt trigger will be turned off and the Schmitt trigger output will remain at a zero state.</p>

## Port C Pull-Up Selection Register – PCPUR

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

Offset: 0x008

Reset value: 0x0000\_0000

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

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

## Port C Pull-Down Selection Register – PCPDR

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

Offset: 0x00C

Reset value: 0x0000\_0000

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

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

## Port C Open-Drain Selection Register – PCODR

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

Offset: 0x010

Reset value: 0x0000\_0000

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

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



## Port C Drive Current Selection Register – PCDRVR

This register specifies the GPIO Port C output driving current.

Offset: 0x014

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	PCDV15		PCDV14		PCDV13		PCDV12		
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	PCDV11		PCDV10		PCDV9		PCDV8		
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	PCDV7		PCDV6		PCDV5		PCDV4		
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	PCDV3		PCDV2		PCDV1		PCDV0		
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW

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

## Port C Lock Register – PCLOCKR

This register specifies the GPIO Port C lock configuration.

Offset: 0x018

Reset value: 0x0000\_0000

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

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

## Port C Data Input Register – PCDINR

This register specifies the GPIO Port C input data.

Offset: 0x01C

Reset value: 0x0000\_0000

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

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

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

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

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

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

Offset: 0x024

Reset value: 0x0000\_0000

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

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

## Port C Output Reset Register – PCRR

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

Offset: 0x028

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[15:0]	PCRSTn	GPIO Port C pin n Output Reset Control 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

	31	30	29	28	27	26	25	24	
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				PDDIR				
					RW	0	RW	0	RW
									0

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

	31	30	29	28	27	26	25	24
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					PDINEN		
						RW	0	RW
							0	RW
								0

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

## Port D Pull-Up Selection Register – PDPUR

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

Offset: 0x008

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
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					PDCU		
						RW	0	RW
							0	RW
								0

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

## Port D Pull-Down Selection Register – PDPDR

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

Offset: 0x00C

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
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					PDPD		
						RW	0	RW
							0	RW
								0

Bits	Field	Descriptions
[2:0]	PDPDn	GPIO Port D pin n Pull-Down Selection Control Bits (n = 0 ~ 2) 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 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

	31	30	29	28	27	26	25	24
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					PDOD		
						RW	0	RW
							0	RW
								0

Bits	Field	Descriptions
[2:0]	PDODn	<p>GPIO Port D pin n Open-Drain Selection Control Bits (n = 0 ~ 2)</p> <p>0: Pin n Open-Drain output is disabled (The output type is CMOS output)</p> <p>1: Pin n Open-Drain output is enabled (The output type is open-drain output)</p> <p>Note: When the open-drain function is enabled, the pin n internal pull-up or pull-down configuration will be invalid.</p>

## Port D Drive Current Selection Register – PDDRVR

This register specifies the GPIO Port D output driving current.

Offset: 0x014

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved		PDDV2		PDDV1		PDDV0	
	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[5:0]	PDDVn[1:0]	GPIO Port D pin n Output Current Drive Selection Control Bits (n = 0 ~ 2) 00: 4 mA source / sink current 01: 8 mA source / sink current 10: 12 mA source / sink current 11: 16 mA source / sink current

## Port D Lock Register – PDLOCKR

This register specifies the GPIO Port D lock configuration.

Offset: 0x018

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[31:16]	PDLKEY	<p>GPIO Port D Lock Key</p> <p>0x5FA0: Port D Lock function is enabled</p> <p>Others: Port D Lock function is disabled</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, PDODn and PDDVn setting in the related GPIO registers. Additionally, the GPDCFGHR or GPDCFGLR register 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

	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				PDDIN			
Type/Reset					RO	0	RO	0

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

	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				PDDOUT			
Type/Reset					RW	0	RW	0

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

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

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 Note that when the PDRSTn bit in this register or (and) the PDRSTn bit in the PDRR register is enabled, the reset function on the PDDOUTn bit will take effect.
[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 PDSETn and PDRSTn bits are set at the same time.

## Port D Output Reset Register – PDRR

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

Offset: 0x028

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
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				PDRST			
					WO	0	WO	0

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

## Port F Data Direction Control Register – PFDIRCR

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

Offset: 0x000

Reset value: 0x0000\_0000

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

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

## Port F Input Function Enable Control Register – PFINER

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

Offset: 0x004

Reset value: 0x0000\_0000

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

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

## Port F Pull-Up Selection Register – PFPUR

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

Offset: 0x008

Reset value: 0x0000\_0000

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

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



## Port F Pull-Down Selection Register – PFPDR

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

Offset: 0x00C

Reset value: 0x0000\_0000

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

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

## Port F Open-Drain Selection Register – PFODR

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

Offset: 0x010

Reset value: 0x0000\_0000

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

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

## Port F Drive Current Selection Register – PFDRVR

This register specifies the GPIO Port F output driving current.

Offset: 0x014

Reset value: 0x0000\_0000

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

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

## Port F Lock Register – PFLOCKR

This register specifies the GPIO Port F lock configuration.

Offset: 0x018

Reset value: 0x0000\_0000

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

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

## Port F Data Input Register – PFDINR

This register specifies the GPIO Port F input data.

Offset: 0x01C

Reset value: 0x0000\_0000

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

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

## Port F Output Data Register – PFDOUTR

This register specifies the GPIO Port F output data.

Offset: 0x020

Reset value: 0x0000\_0000

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

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

## Port F Output Set/Reset Control Register – PFSRR

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

Offset: 0x024

Reset value: 0x0000\_0000

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

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

## Port F Output Reset Register – PFRR

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

Offset: 0x028

Reset value: 0x0000\_0000

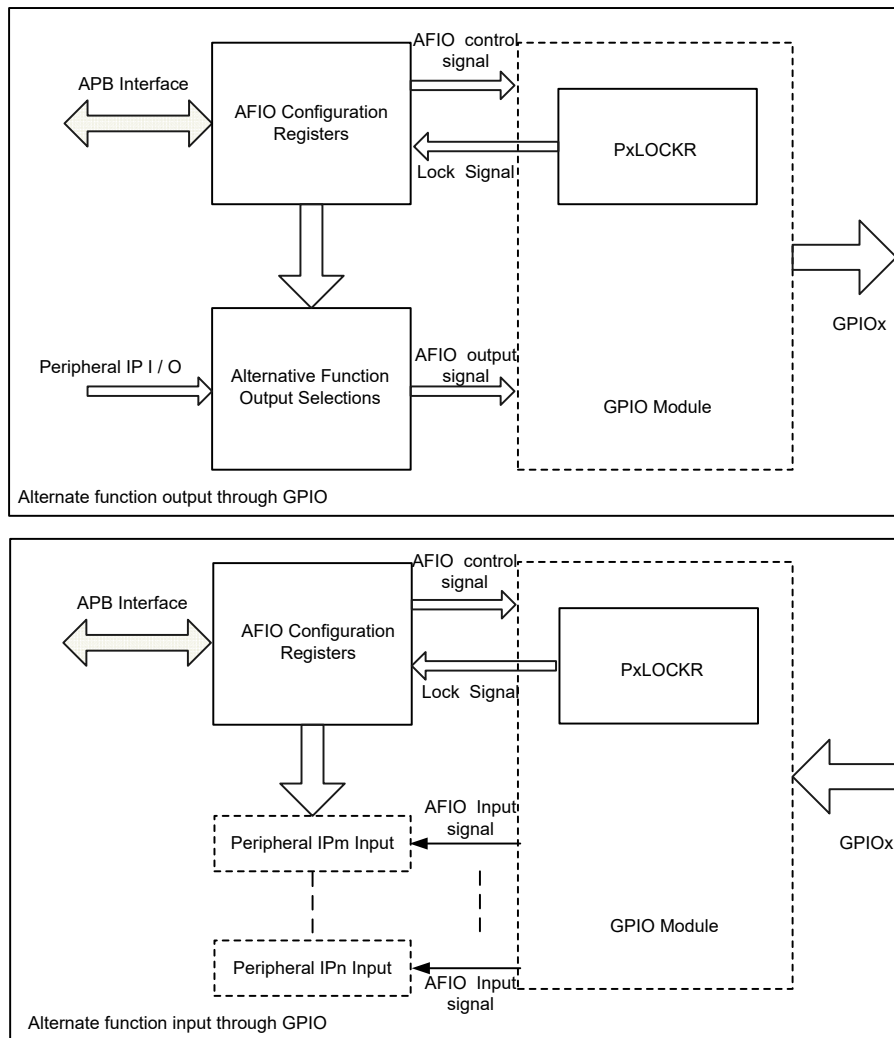
	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved							PFRST
								WO 0

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

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

## Introduction

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



**Figure 23. AFIO Block Diagram**



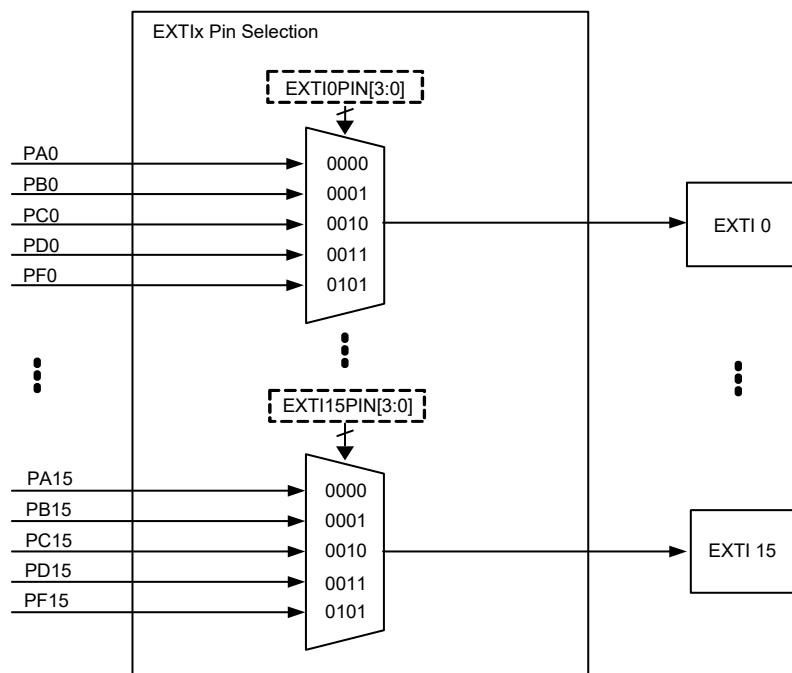
## Features

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

## Functional Descriptions

### External Interrupt Pin Selection

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



**Figure 24. EXTI Channel Input Selection**

## Alternate Function

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

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

**Table 24. AFIO Selection for Peripheral Map Example**

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

## Lock Mechanism

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

## Register Map

The following table shows the AFIO registers and reset value.

**Table 25. AFIO Register Map**

Register	Offset	Description	Reset Value
ESSR0	0x000	EXTI Source Selection Register 0	0x0000_0000
ESSR1	0x004	EXTI Source Selection Register 1	0x0000_0000
GPACFGLR	0x020	GPIO Port A Configuration Low Register	0x0000_0000
GPACFGHR	0x024	GPIO Port A Configuration High Register	0x0000_0000
GPBCFGLR	0x028	GPIO Port B Configuration Low Register	0x0000_0000
GPBCFGHR	0x02C	GPIO Port B Configuration High Register	0x0000_0000
GPCCFGLR	0x030	GPIO Port C Configuration Low Register	0x0000_0000
GPCCFGHR	0x034	GPIO Port C Configuration High Register	0x0000_0000
GPDCFGLR	0x038	GPIO Port D Configuration Low Register	0x0000_0000
GPDCFGHR	0x03C	GPIO Port D Configuration High Register	0x0000_0000
GPFCFGLR	0x048	GPIO Port F Configuration Low Register	0x0000_0000
GPFCFGHR	0x04C	GPIO Port F Configuration High Register	0x0000_0000

## Register Descriptions

### EXTI Source Selection Register 0 – ESSR0

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

Offset: 0x000

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[31:0]	EXTInPIN[3:0]	<p>EXTIn Pin Selection (n = 0 ~ 7)</p> <p>0000: PA Bit n is selected as EXTIn source signal</p> <p>0001: PB Bit n is selected as EXTIn source signal</p> <p>0010: PC Bit n is selected as EXTIn source signal</p> <p>0011: PD Bit n is selected as EXTIn source signal</p> <p>0101: PF Bit n is selected as EXTIn source signal</p> <p>Others: Reserved</p> <p>Note: Since not all GPIO pins are available in all products and package types, refer to the pin assignment section for detailed pin information. The EXTInPIN [3:0] field setting is invalid when the corresponding pin is not available.</p>

## EXTI Source Selection Register 1 – ESSR1

This register specifies the I/O selection of EXTI8 ~ EXTI15.

Offset: 0x004

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[31:0]	EXTInPIN[3:0]	<p>EXTIn Pin Selection (n = 8 ~ 15)</p> <p>0000: PA Bit n is selected as EXTIn source signal</p> <p>0001: PB Bit n is selected as EXTIn source signal</p> <p>0010: PC Bit n is selected as EXTIn source signal</p> <p>0011: PD Bit n is selected as EXTIn source signal</p> <p>0101: PF Bit n is selected as EXTIn source signal</p> <p>Others: Reserved</p> <p>Note: Since not all GPIO pins are available in all products and package types, refer to the pin assignment section for detailed pin information. The EXTInPIN [3:0] field setting is invalid when the corresponding pin is not available.</p>

## GPIO Port x Configuration Low Register – GPxCFGxLR (x = A, B, C, D, F)

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

Offset: 0x020, 0x028, 0x030, 0x038, 0x048

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[31:0]	PxCFGn[3:0]	Alternate function selection for port x pin n (n = 0 ~ 7) 0000: Port x pin n is selected as AF0 0001: Port x pin n is selected as AF1 : : 1110: Port x pin n is selected as AF14 1111: Port x pin n is selected as AF15 Refer to the "Alternate function mapping" table in the device datasheet for the detailed mapping of the alternate function I/O pins.

## GPIO Port x Configuration High Register – GPxCFGHR (x = A, B, C, D, F)

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

Offset: 0x024, 0x02C, 0x034, 0x03C, 0x04C

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[31:0]	PxCFGn[3:0]	<p>Alternate function selection for port x pin n (n = 8 ~ 15)</p> <p>0000: Port x pin n is selected as AF0</p> <p>0001: Port x pin n is selected as AF1</p> <p>⋮</p> <p>1110: Port x pin n is selected as AF14</p> <p>1111: Port x pin n is selected as AF15</p> <p>Refer to the "Alternate function mapping" table in the device datasheet for the detailed mapping of the alternate function I/O pins.</p>

# 10 Nested Vectored Interrupt Controller (NVIC)

## Introduction

In order to reduce the latency and increase the interrupt processing efficiency, a tightly coupled integrated section, which is named as Nested Vectored Interrupt Controller (NVIC) is provided by the Cortex®-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 Time Operation System (RTOS) or as a simple counter. The SysTick counts down from the reloaded value and generates a system interrupt when it reached zero.

The accompanying table lists the system exceptions types and a variety of peripheral interrupts.

**Table 26. Exception types**

Exception Type	Priority	Interrupt Number	Exception Number	Vector Address	Description
	—	—	0	0x000	Initial Stack Point 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
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
—	—	—	13	0x034	Reserved
PendSV	Configurable <sup>(1)</sup>	—	14	0x038	System Service Pendable request
SysTick	Configurable <sup>(1)</sup>	—	15	0x03C	SysTick timer decreases to zero
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

Exception Type	Priority	Interrupt Number	Exception Number	Vector Address	Description
—	—	3	19	0x04C	Reserved
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
—	—	24	40	0x0A0	Reserved
ADC	Configurable <sup>(2)</sup>	25	41	0x0A4	ADC global interrupt
—	—	26	42	0x0A8	Reserved
—	—	27	43	0x0AC	Reserved
—	—	28	44	0x0B0	Reserved
—	—	29	45	0x0B4	Reserved
—	—	30	46	0x0B8	Reserved
—	—	31	47	0x0BC	Reserved
—	—	32	48	0x0C0	Reserved
—	—	33	49	0x0C4	Reserved
—	—	34	50	0x0C8	Reserved
GPTM	Configurable <sup>(2)</sup>	35	51	0x0CC	GPTM global interrupt
—	—	36	52	0x0D0	Reserved
SCTM0	Configurable <sup>(2)</sup>	37	53	0x0D4	SCTM0 global interrupt
SCTM1	Configurable <sup>(2)</sup>	38	54	0x0D8	SCTM1 global interrupt
PWM	Configurable <sup>(2)</sup>	39	55	0x0DC	PWM global interrupt
—	—	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
SPI1	Configurable <sup>(2)</sup>	46	62	0x0F8	SPI1 global interrupt



Exception Type	Priority	Interrupt Number	Exception Number	Vector Address	Description
USART	Configurable <sup>(2)</sup>	47	63	0x0FC	USART global interrupt
—	—	48	64	0x100	Reserved
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
—	—	52	68	0x110	Reserved
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
—	—	61	77	0x134	Reserved
—	—	62	78	0x138	Reserved
—	—	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
AES	Configurable <sup>(2)</sup>	69	85	0x154	AES 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 64 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 decreases 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 of 1ms for the RTOS tick timer or other purpose. The TENMS field in the SCALIB register has a fixed value of 10000 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 10 MHz (80 MHz divide by 8).

## Register Map

The following table shows the NVIC registers and reset values.

**Table 27. NVIC Register Map**

Register	Offset	Description	Reset Value
<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_2710
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

Register	Offset	Description	Reset Value
IRQ44_47	0x42C	Irq 44 to 47 Priority Register	0x0000_0000
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 above detail register descriptions, 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 (n = 0 ~ 15) register. In the wake-up event mode, the wake-up event polarity can be configured by setting the EXTInWPOL (n = 0 ~ 15) field in the EXTIWAKUPPOLR register. If the EVWUPIEN bit in the EXTIWAKUPCR Register is set, the EVWUP interrupt can be generated when the associated wake-up event occurs and the corresponding EXTI wake-up enable bit is set. Each EXTI line can also be masked independently.

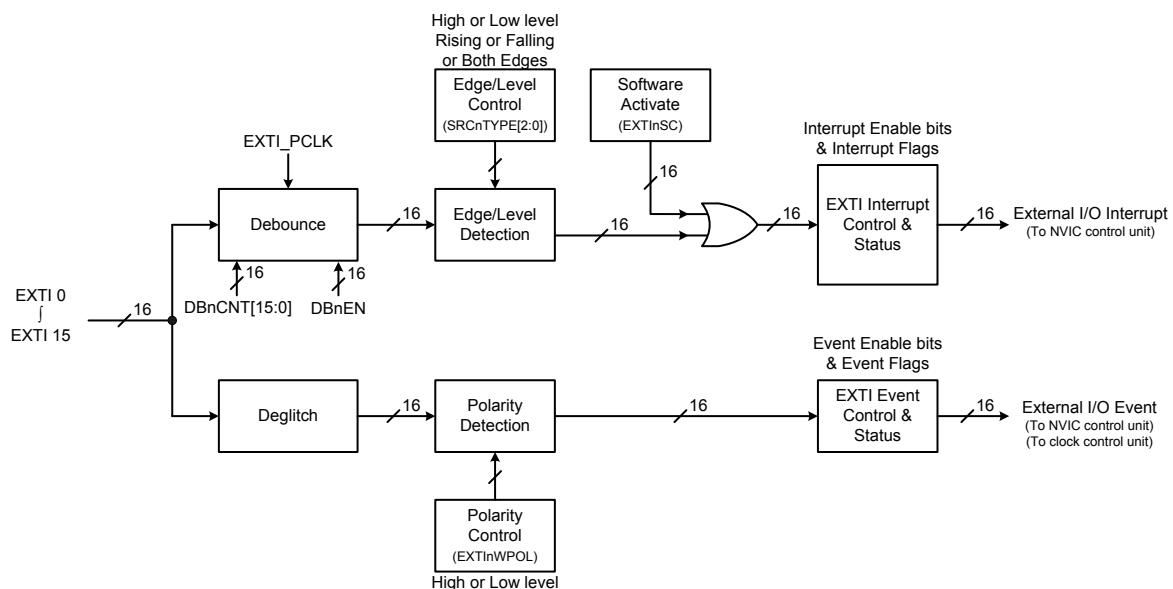


Figure 25. EXTI Block Diagram

## Features

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

## Functional Descriptions

### Wakeup Event Management

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

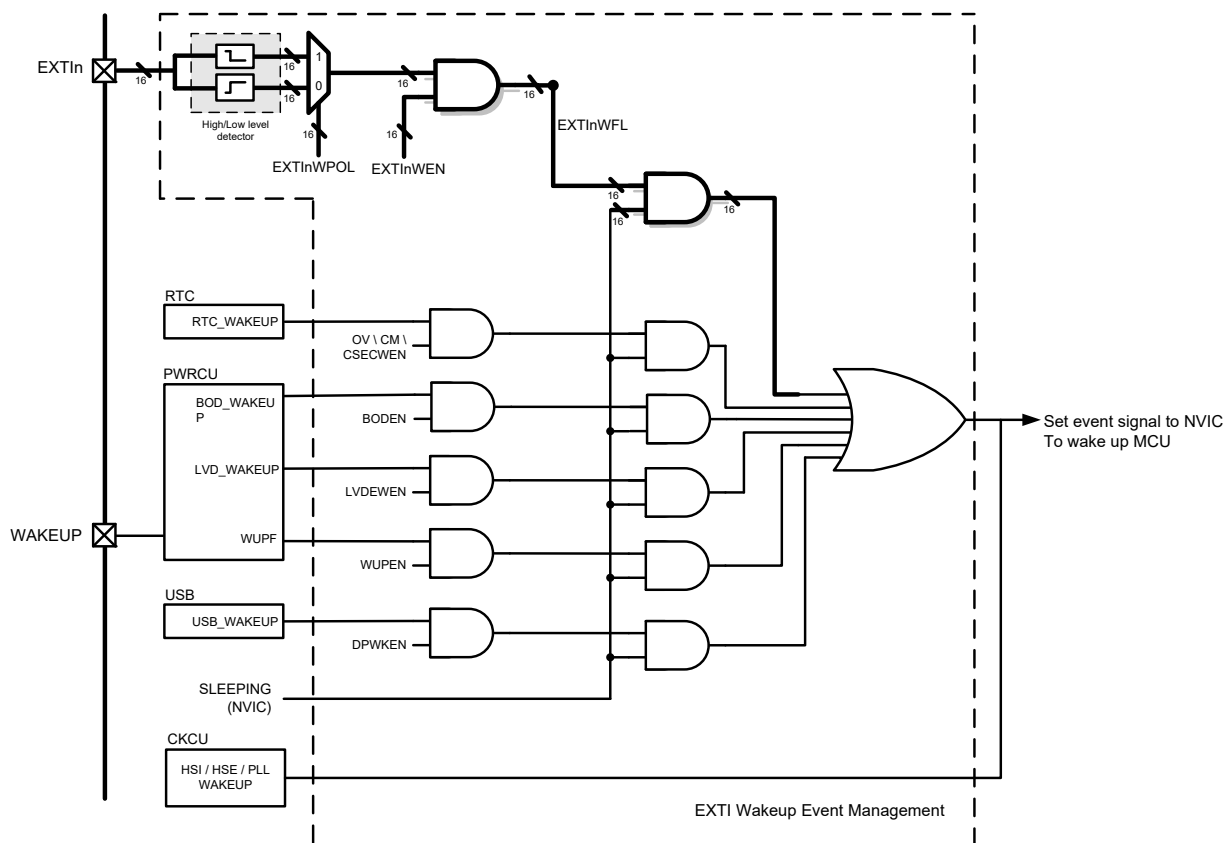
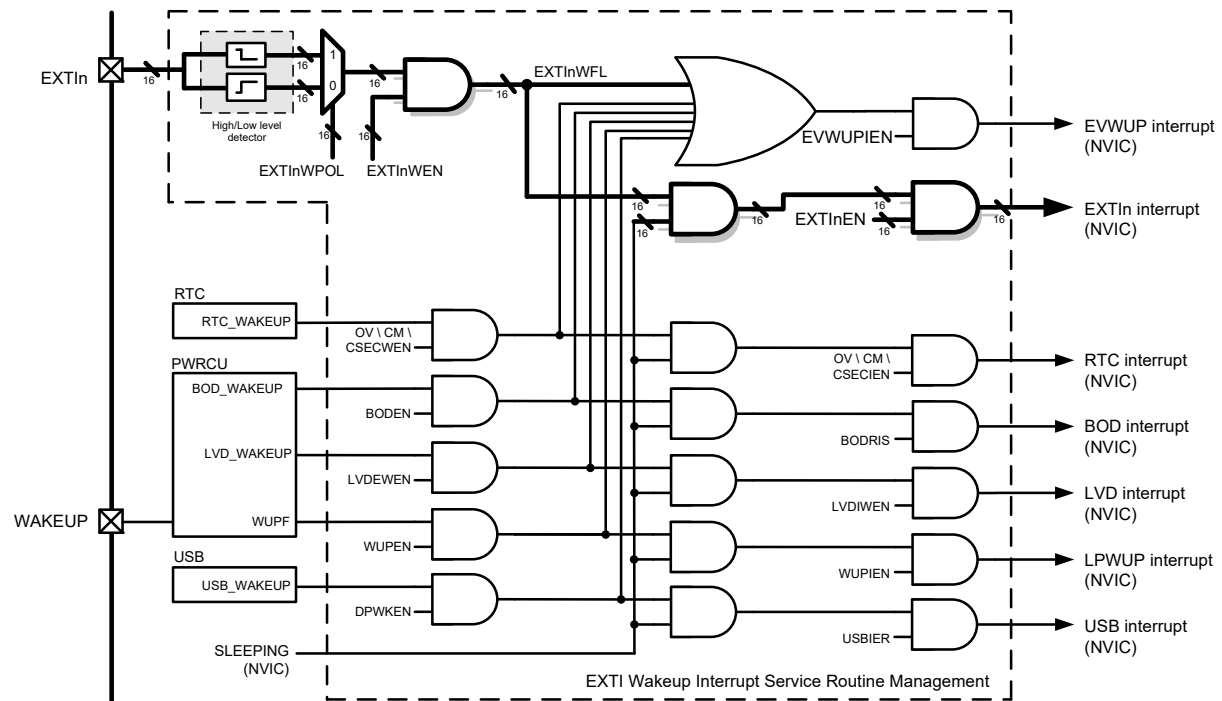


Figure 26. EXTI Wake-Up Event Management

## External Interrupt/Event Line Mapping

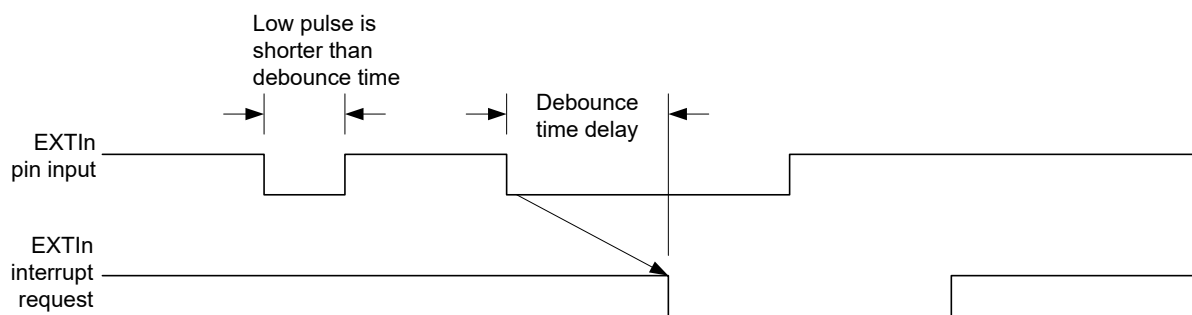
All GPIO pins can be selected as EXTI trigger sources by configuring the EXTIInPIN [3:0] field in the AFIO ESSRn (n = 0 ~ 1) register to trigger an interrupt or event. Refer to the AFIO section for more details.



**Figure 27. Wake-Up Interrupt Service Routine**

## Interrupt and Debounce

The application software can set the DBnEN bit in the EXTIIn Interrupt Configuration Register EXTICFGRn ( $n = 0 \sim 15$ ) to enable the corresponding pin de-bounce function and configure the DBnCNT field in the EXTICFGRn so as to select an appropriate de-bounce time for specific applications. The interrupt signal will however be delayed due to the de-bounce function. When the device is woken up from the power saving mode by an external interrupt, an interrupt request will be generated by the EXTI wakeup flag. After the device has been woken up and the clock has recovered, the EXTI wake-up flag that was triggered by the EXTI line must be read and then cleared by application software. The accompanying diagram shows the relationship between the EXTI input signal and the EXTI interrupt/event request signal.



**Figure 28. EXTI Interrupt Debounce Function**

## Register Map

The following table shows the EXTI registers and reset values.

**Table 28. EXTI Register Map**

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



## Register Descriptions

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

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

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

Reset value: 0x0000\_0000

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

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

## EXTI Interrupt Control Register – EXTICR

This register is used to control the EXTI interrupt.

Offset: 0x040

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	EXTI15EN	EXTI14EN	EXTI13EN	EXTI12EN	EXTI11EN	EXTI10EN	EXTI9EN	EXTI8EN	
	7	6	5	4	3	2	1	0	
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	EXTI7EN	EXTI6EN	EXTI5EN	EXTI4EN	EXTI3EN	EXTI2EN	EXTI1EN	EXTI0EN	

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

## EXTI Interrupt Edge Flag Register – EXTIEDGEFLGR

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

Offset: 0x044

Reset value: 0x0000\_0000

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

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

## EXTI Interrupt Edge Status Register – EXTIEDGESR

This register indicates the polarity of a detected EXTI edge.

Offset: 0x048

Reset value: 0x0000\_0000

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

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

## EXTI Interrupt Software Set Command Register – EXTISSCR

This register is used to activate the EXTI interrupt.

Offset: 0x04C

Reset value: 0x0000\_0000

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

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

## EXTI Interrupt Wakeup Control Register – EXTIWAKUPCR

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

Offset: 0x050

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24										
	EVWUPIEN		Reserved															
Type/Reset	RW	0																
	23	22	21	20	19	18	17	16										
	Reserved																	
Type/Reset																		
	15	14	13	12	11	10	9	8										
	EXTI15WEN		EXTI14WEN		EXTI13WEN		EXTI12WEN		EXTI11WEN		EXTI10WEN		EXTI9WEN		EXTI8WEN			
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0		
	7	6	5	4	3	2	1	0										
	EXTI7WEN		EXTI6WEN		EXTI5WEN		EXTI4WEN		EXTI3WEN		EXTI2WEN		EXTI1WEN		EXTI0WEN			
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0		

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

## EXTI Interrupt Wakeup Polarity Register – EXTIWAKUPPOLR

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

Offset: 0x054

Reset value: 0x0000\_0000

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

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

## EXTI Interrupt Wakeup Flag Register – EXTIWAKUPFLG

This register is the EXTI interrupt wakeup flag register.

Offset: 0x058

Reset value: 0x0000\_0000

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

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 (ADC) with a Voltage Reference Generator ( $V_{REF}$ ) is integrated in the devices. There are a total of 11 multiplexed channels including 8 external channels on which the external analog signal can be supplied and 3 internal channels. If the input voltage is required to remain within a specific threshold window, the Analog Watchdog function will monitor and detect the signal. An interrupt will then be generated to inform that the input voltage is higher or lower than the set thresholds. There are three conversion modes to convert an analog signal to digital data. The A/D conversion can be operated in one shot, continuous and discontinuous conversion mode. A 16-bit data register is provided to store the data after conversion.

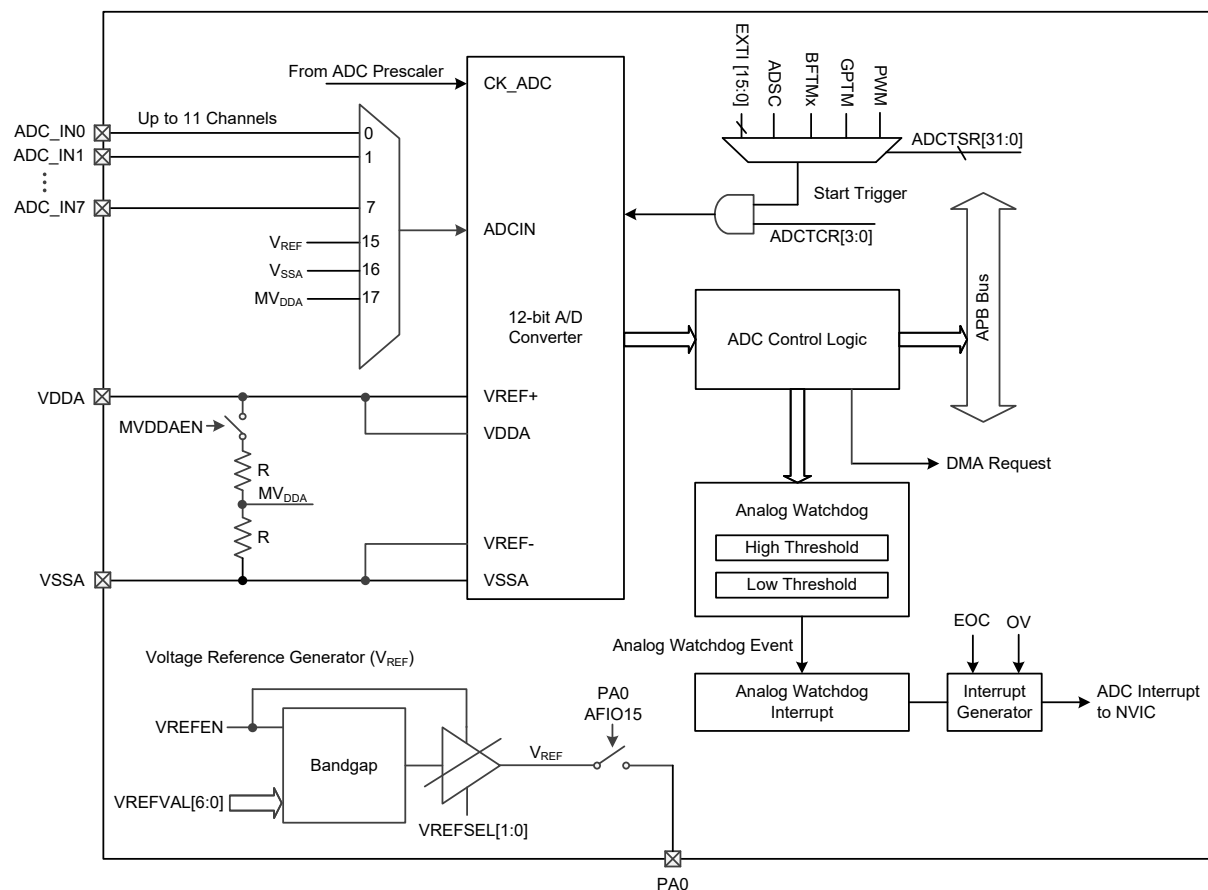


Figure 29. ADC with  $V_{REF}$  Block Diagram

## Features

- 12-bit SAR ADC engine
- Up to 1 MSPS conversion rate
- up to 8 external analog input channels
- 1 channel for internal voltage reference ( $V_{REF}$ )
- 1 channel for monitor external  $V_{DDA}$  power support pin
- Programmable sampling time for conversion channel
- Up to 8 programmable conversion channel sequence and dedicated data registers for conversion result
- Three conversion mode
  - One shot conversion mode
  - Continuous conversion mode
  - Discontinuous conversion mode
- Analog watchdog for predefined voltage range monitor
  - Lower/upper threshold register
  - Interrupt generation
- Various trigger start sources for conversion modes
  - Software trigger
  - EXTI – external interrupt input pin
  - GPTM trigger
  - PWM trigger
  - BFTM0 / BFTM1 trigger
- Multiple generated interrupts
  - End of single conversion
  - End of subgroup conversion
  - End of cycle conversion
  - Analog Watchdog
  - Data register overwriting
- PDMA request on end of conversion occurred

## Functional Descriptions

### ADC Clock Setup

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

### Channel Selection

The A/D converter supports 11 multiplexed channels and organizes the conversion results into a specific group. A conversion group can organize a sequence which can be implemented on the channels arranged in a specific conversion sequence length from 1 to 8. For example, conversion can be carried out with the following channel sequence: CH2, CH4, CH7, CH5, CH6, CH3, CH0 and CH1 one after another.

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

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

### Conversion Mode

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

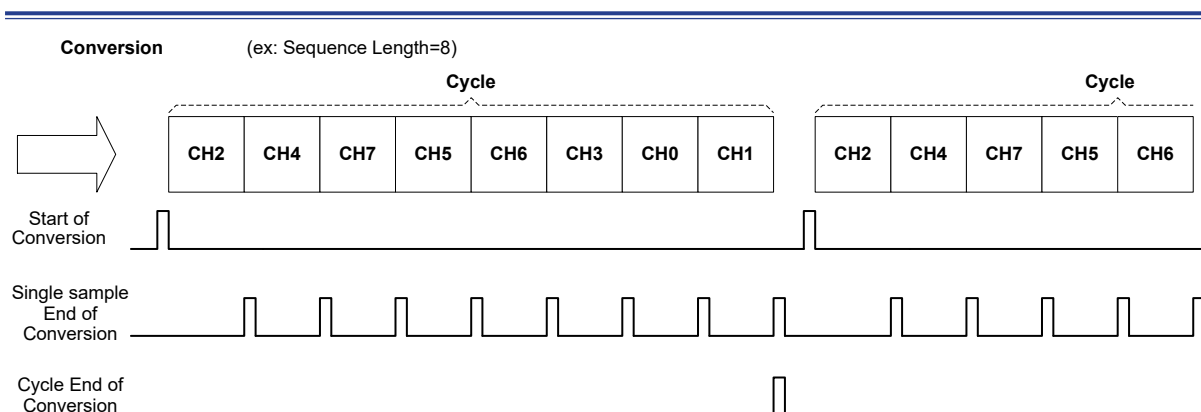
#### One Shot Conversion Mode

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

#### After Conversion:

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





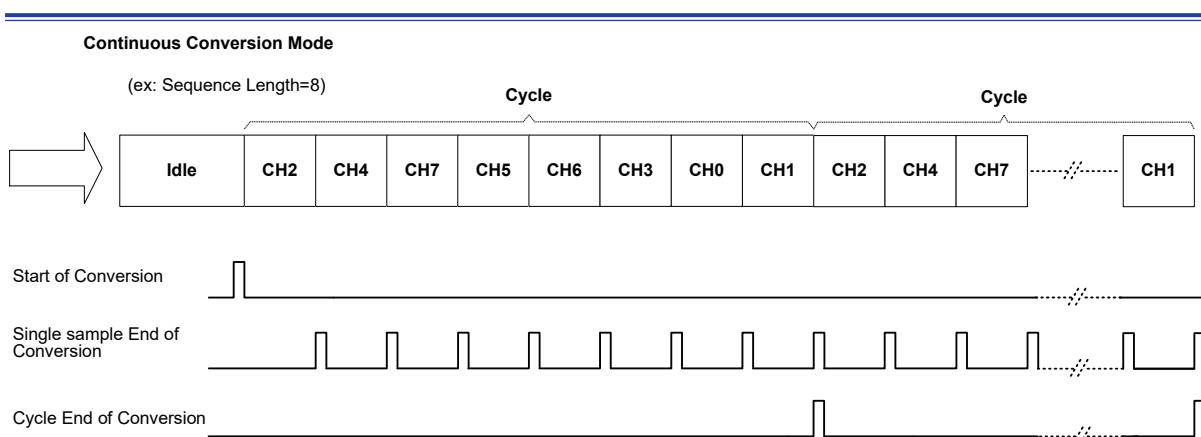
**Figure 30. One Shot Conversion Mode**

### Continuous Conversion Mode

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

#### After conversion:

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



**Figure 31. Continuous Conversion Mode**

### Discontinuous Conversion Mode

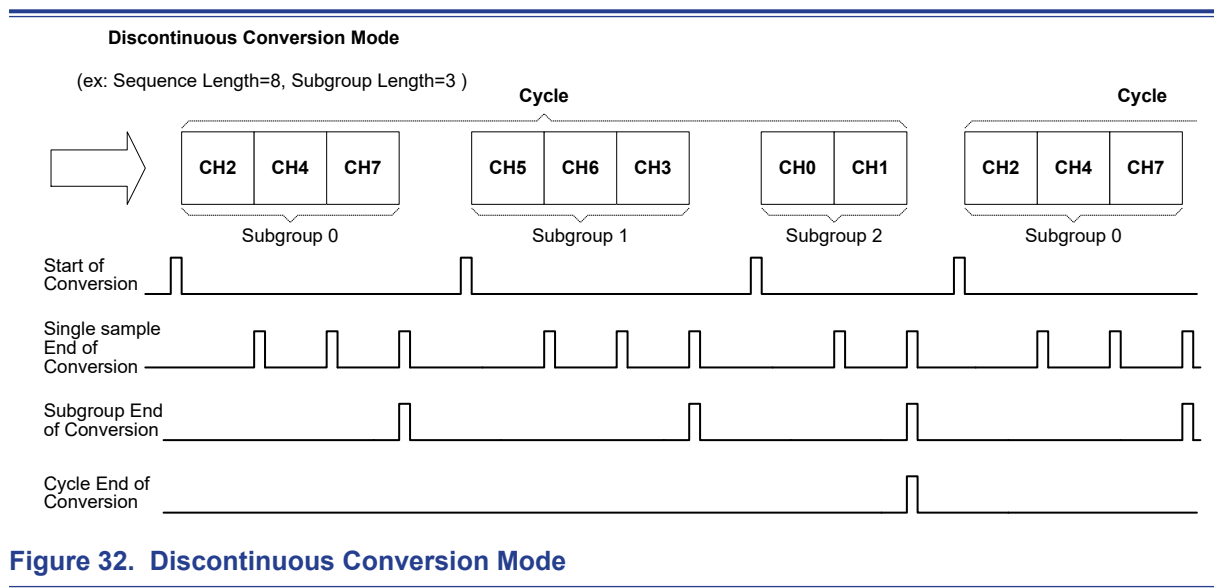
The A/D converter will operate in the Discontinuous Conversion Mode for channels group when the A/D conversion mode bit field ADMODE [1:0] in the ADCCR register is set to 0x3. The group to be converted can have up to 8 channels and can be arranged in a specific sequence by configuring the ADCLSTn registers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

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

Example:

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

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



**Figure 32. Discontinuous Conversion Mode**

## Start Conversion on External Event

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

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

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

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

## Sampling Time Setting

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

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

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

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

Example:

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

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

## Data Format

The ADC conversion result can be read in the ADCDRy register and the data format is shown in the following tData format in ADCDR [15:0].

**Table 29. Data Format in ADCDR [15:0]**

Description	ADCDR register Data Format
Right aligned	"0_0_0_0_d11_d10_d9_d8_d7_d6_d5_d4_d3_d2_d1_d0"

## Analog Watchdog

The A/D converter includes a watchdog function to monitor the converted data. There are two kinds of thresholds for the watchdog monitor function, known as the watchdog lower threshold and watchdog upper threshold, which are specified by the ADLT bit field and ADUT bit field in the ADCTR register 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 by the ADLT bit field and ADUT bit field in the ADCTR register 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 A/D conversion. A single sample EOC event will occur and the single sample EOC interrupt raw flag, ADIRAWS bits in the ADCIRAW register, will be asserted when a single channel conversion has completed. A subgroup EOC event will occur and the subgroup EOC interrupt raw flag, ADIRAWG in the ADCIRAW register, will be asserted when a subgroup conversion has completed. A cycle EOC event will occur and the cycle EOC interrupt raw flag, ADIRAWC bits in the ADCIRAW register, will be asserted when a cycle conversion is finished. When a single sample EOC, a subgroup EOC or a cycle EOC raw flag is asserted and the corresponding interrupt enable bit, ADIES, ADIEG or ADIEC bit in the ADCIER register, is set to 1, the associated interrupt will be generated.

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

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

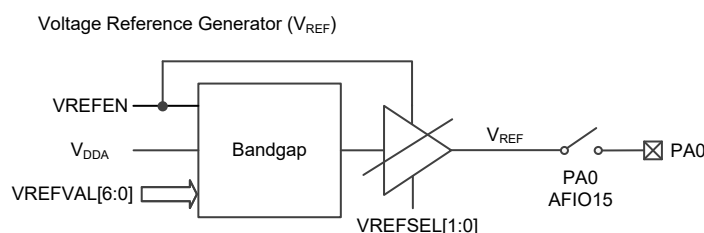
The A/D Converter interrupt clear bits are used to clear the associated A/D converter interrupt raw and interrupt status bits. Writing a 1 into the specific A/D converter interrupt clear bit in the A/D converter interrupt clear register ADCICLR will clear the corresponding A/D converter interrupt raw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

## PDMA Request

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

## Voltage Reference Generator

The internal voltage reference generator ( $V_{REF}$ ) provides a stable voltage output for the ADC. The  $V_{REF}$  is internally connected to the ADC input channel. The precise voltage of the  $V_{REF}$  is individually measured and calculated for each part by manufacture during production test and stored in the Flash Manufacture Privilege Information Block. It can be accessed using the VREFVALR register in the read-only mode. Refer to the datasheet for additional information. When not in use or using the external voltage reference from the VREF pin, the internal  $V_{REF}$  generator can be configured in the power down mode to save power consumption.



**Figure 33. Voltage Reference Generator Block Diagram**

### V<sub>DDA</sub> Voltage Monitor

The MVDDAEN bit in the VREFCR register allows the applications to measure the V<sub>DDA</sub> voltage on the VDDA pin. As the V<sub>DDA</sub> voltage could be higher than the ADC reference voltage, in order to ensure the correct operation of the ADC, the VDDA pin is internally connected to a bridge divider by 2. This bridge is automatically enabled when the MVDDAEN bit is set, to connect the V<sub>DDA</sub>/2 to the ADC input channel. As a consequence, the converted digital value is half of the V<sub>DDA</sub> voltage. To prevent any unwanted consumption on the battery, it is recommended to enable the V<sub>DDA</sub> power divider only when the ADC conversion is required

## Register Map

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

**Table 30. A/D Converter Register Map**

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

Register	Offset	Description	Reset Value
VREFCR	0x0A0	Voltage Reference Control Register	0x0000_0000
VREFVALR	0x0A4	Voltage Reference Value Register	0x0000_00XX

## Register Descriptions

### ADC Conversion Control Register – ADCCR

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

Offset: 0x000

Reset value: 0x0000\_0000

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

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

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



## ADC Conversion List Register 0 – ADCLST0

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

Offset: 0x004

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[28:24]	ADSEQ3	ADC Conversion Sequence Select 3 Select the ADC input channel for the 3 <sup>rd</sup> ADC conversion sequence. 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 ~ 0x0E: Reserved 0x0F: Internal Voltage Reference ( $V_{REF}$ ) 0x10: Analog ground $V_{SSA}$ 0x11: Analog power $MV_{DDA}$ ( $V_{DDA}/2$ ) 0x12 ~ 0x1F: Invalid setting. These values must not be selected as it may cause the ADC abnormal operations.
[20:16]	ADSEQ2	ADC Conversion Sequence Select 2
[12:8]	ADSEQ1	ADC Conversion Sequence Select 1
[4:0]	ADSEQ0	ADC Conversion Sequence Select 0

## ADC Conversion List Register 1 – ADCLST1

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

Offset: 0x008

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[28:24]	ADSEQ7	ADC Conversion Sequence Select 7 Select the ADC input channel for the 7 <sup>th</sup> ADC conversion sequence. 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 ~ 0x0E: Reserved 0x0F: Internal Voltage Reference ( $V_{REF}$ ) 0x10: Analog ground $V_{SSA}$ 0x11: Analog power $MV_{DDA}$ ( $V_{DDA}/2$ ) 0x12 ~ 0x1F: Invalid setting. These values must not be selected as it may cause the ADC abnormal operations.
[20:16]	ADSEQ6	ADC Conversion Sequence Select 6
[12:8]	ADSEQ5	ADC Conversion Sequence Select 5
[4:0]	ADSEQ4	ADC Conversion Sequence Select 4

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

Reset value: 0x0000\_0000

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

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

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

Offset: 0x030 ~ 0x04C

Reset value: 0x0000\_0000

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

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

## ADC Trigger Control Register – ADCTCR

This register contains the ADC start conversion trigger enable bits.

Offset: 0x070

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved				TM1	TM0	ADEXTI	ADSW
					RW	0	RW	0
						RW	0	RW
							RW	0

Bits	Field	Descriptions
[3]	TM1	ADC Conversion BFTM or PWM Event Trigger enable control 0: Disable conversion trigger by BFTM or PWM events 1: Enable conversion trigger by BFTM or PWM events
[2]	TM0	ADC Conversion GPTM Event Trigger enable control 0: Disable conversion trigger by GPTM events 1: Enable conversion trigger by GPTM events
[1]	ADEXTI	ADC Conversion EXTI Event Trigger enable control 0: Disable conversion trigger by EXTI lines 1: Enable conversion trigger by EXTI lines
[0]	ADSW	ADC Conversion Software Trigger enable control 0: Disable conversion trigger by software trigger bit 1: Enable conversion trigger by software trigger bit

## ADC Trigger Source Register – ADCTSR

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

Offset: 0x074

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
	Reserved		TM1E				TM0E	
Type/Reset			RW	0	RW	0	RW	0
	23	22	21	20	19	18	17	16
	TM1S[2:1]		Reserved		TM1S[0]	TM0S		
Type/Reset	RW	0	RW	0		RW	0	RW
	15	14	13	12	11	10	9	8
	Reserved				ADEXTIS			
Type/Reset					RW	0	RW	0
	7	6	5	4	3	2	1	0
	Reserved							ADSC
Type/Reset								RW

Bits	Field	Descriptions
[29:27]	TM1E	PWM Trigger Event Selection of ADC Conversion 000: PWM MTO event 001: PWM CH0O event 010: PWM CH1O event 011: PWM CH2O event 100: PWM CH3O event Others: Reserved – Should not be used to avoid unpredictable results
[26:24]	TM0E	GPTM Trigger Event Selection of ADC Conversion 000: GPTM MTO event 001: GPTM CH0O event 010: GPTM CH1O event 011: GPTM CH2O event 100: GPTM CH3O event Others: Reserved – Should not be used to avoid unpredictable results
[23:22], [19]	TM1S	BFTM or PWM Trigger Timer Selection of ADC Conversion 000: BFTM0 001: BFTM1 010: PWM 011: Reserved Others: Reserved – Should not be used to avoid unpredictable results
[18:16]	TM0S	GPTM Trigger Timer Selection of ADC Conversion 010: GPTM Others: Reserved - Should not be used to avoid unpredictable results
[11:8]	ADEXTIS	EXTI Trigger Source Selection of ADC Conversion 0x0: EXTI line 0 0x1: EXTI line 1 ... 0xF: EXTI line 15 Note that the EXTI line active edge to start an A/D conversion is determined in the External Interrupt/Event Control Unit, EXTI.

Bits	Field	Descriptions
[0]	ADSC	ADC Conversion Software Trigger Bit 0: No operation 1: Start conversion immediately This bit is set by software to start a conversion manually and then cleared by hardware automatically after conversion started.

## ADC Watchdog Control Register – ADCWCR

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

Offset: 0x078

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[27:24]	ADUCH	Upper Threshold Channel Status 0000: ADC_IN0 converted data is higher than the upper threshold 0001: ADC_IN1 converted data is higher than the upper threshold ... 0101: ADC_IN5 converted data is higher than the upper threshold 0110: ADC_IN6 converted data is higher than the upper threshold 0111: ADC_IN7 converted data is higher than the upper threshold Others: Reserved If one of these status bits is set to 1 by the watchdog monitor function, the status field value should first be stored in the user-defined memory location in the corresponding ISR. Otherwise, the ADUCH field will be changed if another input channel converted data is higher than the upper threshold.

Bits	Field	Descriptions
[19:16]	ADLCH	<p>Lower Threshold Channel Status</p> <p>0000: ADC_IN0 converted data is lower then the lower threshold 0001: ADC_IN1 converted data is lower then the lower threshold ... 0101: ADC_IN5 converted data is lower then the lower threshold 0110: ADC_IN6 converted data is lower than the lower threshold 0111: ADC_IN7 converted data is lower than the lower threshold Others: Reserved</p> <p>If one of these status bits is set to 1 by the watchdog monitor function, the status field value should first be stored in the user-defined memory location in the corresponding ISR. Otherwise, the ADLCH field will be changed if another input channel converted data is lower than the lower threshold.</p>
[11:8]	ADWCH	<p>ADC Watchdog Specific Channel Selection</p> <p>0000: ADC_IN0 is monitored 0001: ADC_IN1 is monitored ... 0101: ADC_IN5 is monitored 0110: ADC_IN6 is monitored 0111: ADC_IN7 is monitored Others: Reserved</p>
[2]	ADWALL	<p>ADC Watchdog Specific or All Channel Setting</p> <p>0: Only the channel which specified by the ADWCH field is monitored 1: All channels are monitored</p>
[1]	ADWUE	<p>ADC Watchdog Upper Threshold Enable Bit</p> <p>0: Disable upper threshold monitor function 1: Enable upper threshold monitor function</p>
[0]	ADWLE	<p>ADC Watchdog Lower Threshold Enable Bit</p> <p>0: Disable lower threshold monitor function 1: Enable lower threshold monitor function</p>



## ADC Watchdog Threshold Register – ADCTR

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

Offset: 0x07C

Reset value: 0x0000\_0000

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

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

## ADC Interrupt Enable Register – ADCIER

This register contains the ADC interrupt enable bits.

Offset: 0x080

Reset value: 0x0000\_0000

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

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

## ADC Interrupt Raw Status Register – ADCIRAW

This register contains the ADC interrupt raw status bits.

Offset: 0x084

Reset value: 0x0000\_0000

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

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

## ADC Interrupt Status Register – ADCISR

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

Offset: 0x088

Reset value: 0x0000\_0000

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

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

## ADC Interrupt Clear Register – ADCICLR

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

Offset: 0x08C

Reset value: 0x0000\_0000

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

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

## ADC DMA Request Register – ADCDMAR

This register contains the ADC DMA request enable bits.

Offset: 0x090

Reset value: 0x0000\_0000

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

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

## Voltage Reference Control Register – VREFCR

This register contains the internal voltage reference control bits.

Offset: 0x0A0

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[8]	MVDDAEN	Measurement $V_{DDA}/2$ power Enable 0: Disable 1: Enable measurement $V_{DDA}/2$ power
[5:4]	VREFSEL	Voltage Reference Output Selection 00: 1.215 V 01: 2.0 V 10: 2.5 V 11: 2.7 V These bits select the Voltage Reference output level.
[0]	VREFEN	Voltage Reference Enable 0: Disable Voltage Reference 1: Enable Voltage Reference

## Voltage Reference Value Register – VREFVALR

This register contains the internal voltage reference trim value.

Offset: 0x0A4

Reset value: 0x0000\_00XX (Various depending on Flash Manufacture Privilege Information Block)

	31	30	29	28	27	26	25	24
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	VREFVAL						
		RO	X RO	X RO	X RO	X RO	X RO	X RO

Bits	Field	Descriptions
[6:0]	VREFVAL	Voltage Reference Calibration Value During the manufacturing process, the calibration data of the internal voltage reference is stored in the Flash Manufacture Privilege Information Block and downloaded to this field when the system is powered on.



### 13 General-Purpose Timer (GPTM)

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.



## Features

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

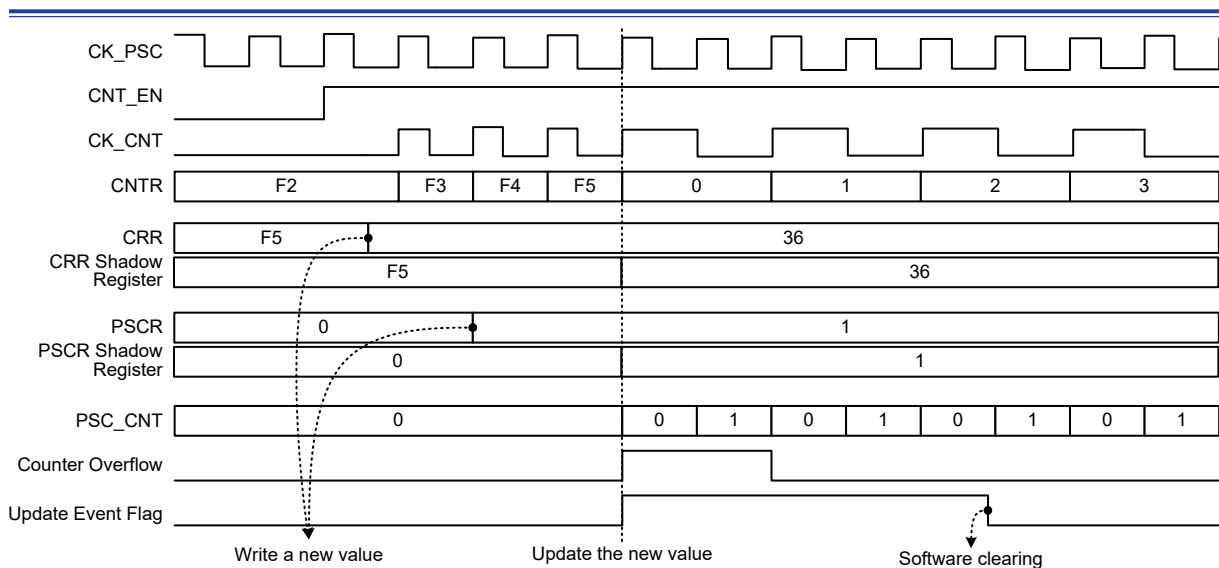
## Functional Descriptions

### Counter Mode

#### Up-Counting

In this mode the counter counts continuously from 0 to the counter-reload value, which is defined in the CRR register, in a count-up direction. Once the counter reaches the counter-reload value, then restarts from 0 and generates a counter overflow event. 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.

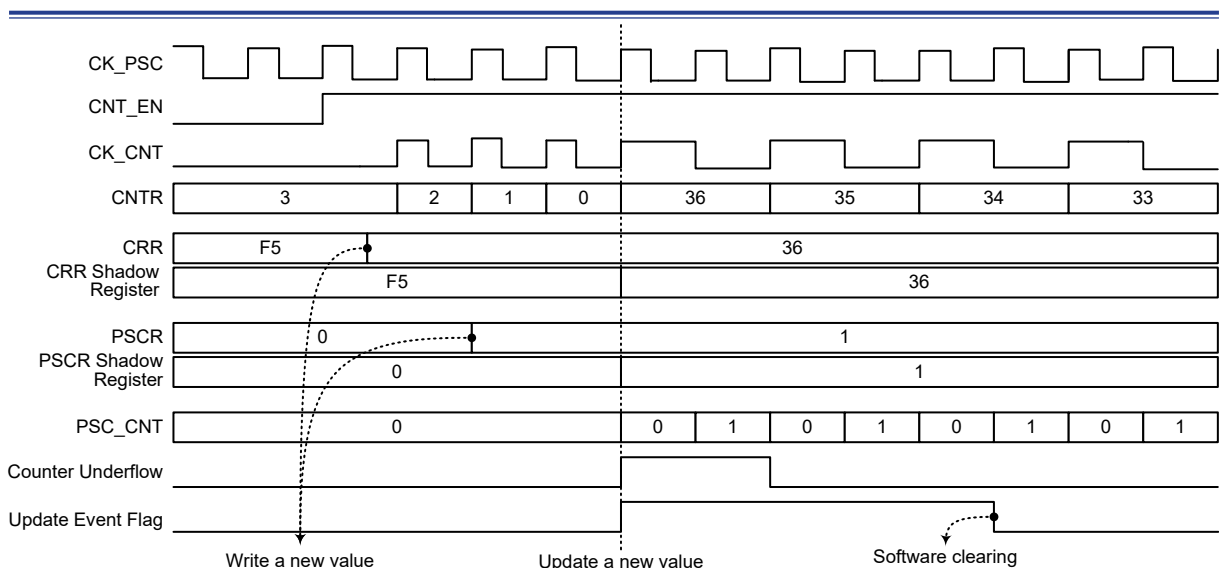


**Figure 35. Up-counting Example**

### 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, then restarts from the counter-reload value and generates a counter underflow event. 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.



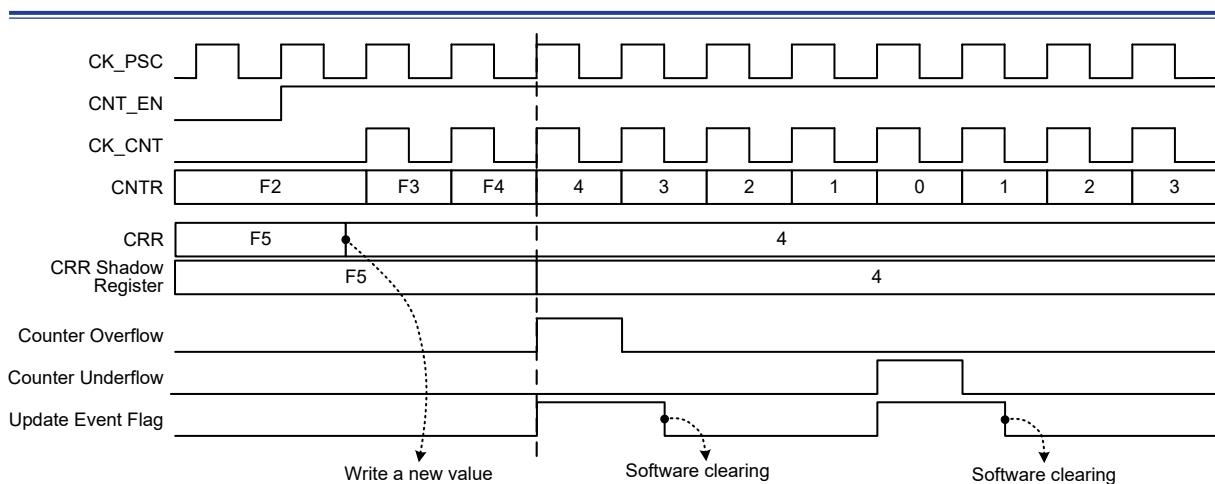
**Figure 36. Down-counting Example**

### Center-Aligned Counting

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

The update event interrupt flag bit in the INTSR register will be set to 1, when an overflow or underflow event occurs.



**Figure 37. Center-aligned Counting Example**

### Clock Controller

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

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

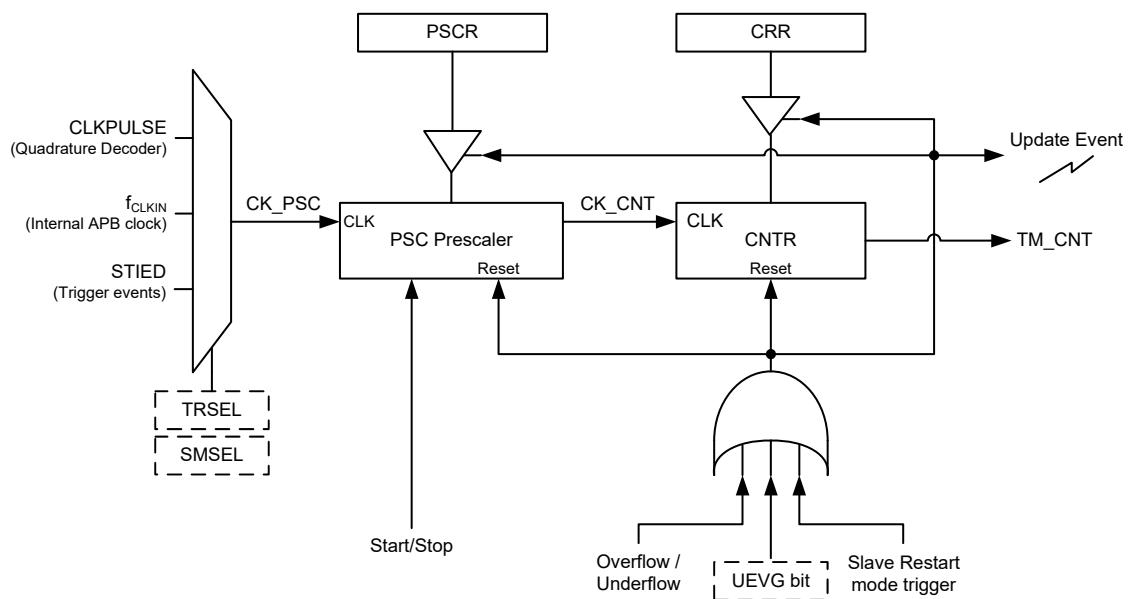
The default internal clock source is the APB clock  $f_{CLKIN}$  used to drive the counter prescaler when the slave mode is disabled. When the slave mode selection bits SMSEL in the MDCFR register are set to 0x4, 0x5 or 0x6, the internal APB clock  $f_{CLKIN}$  is the counter prescaler driving clock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

#### ■ Quadrature Decoder:

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

■ STIED:

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



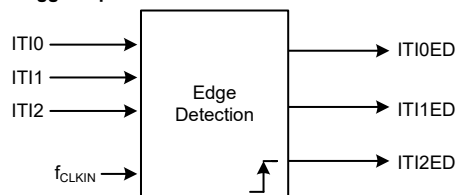
**Figure 38. GPTM Clock Source Selection**

## Trigger Controller

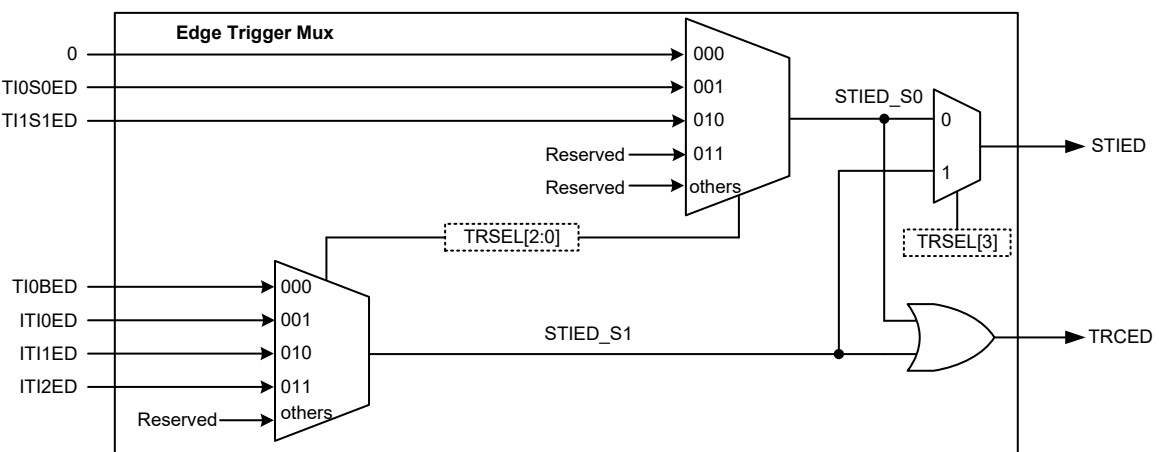
The trigger controller is used to select the trigger source and setup the trigger level or trigger edge condition. For the internal trigger input, it can be selected by the Trigger Selection bits TRSEL in the TRCFR register. For all the trigger sources except the UEVG bit software trigger, the internal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

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

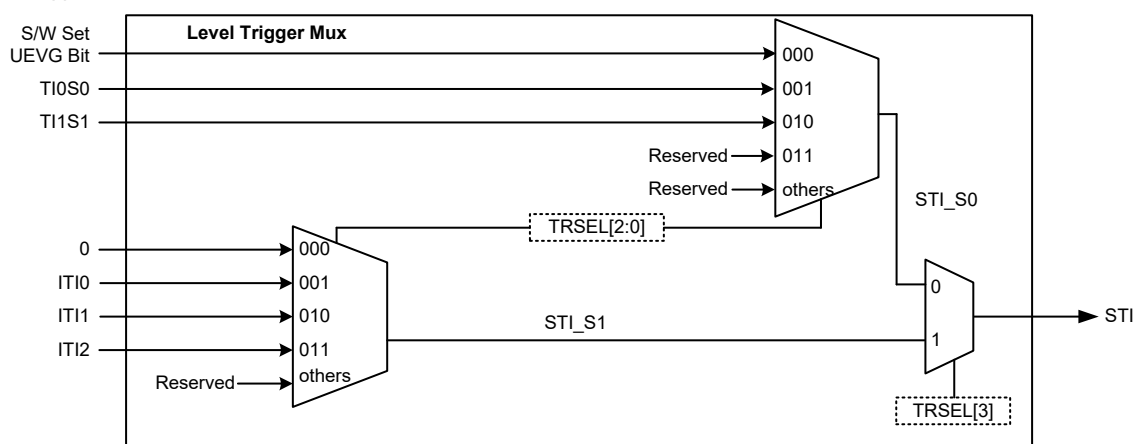
**Internal Trigger Input**



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



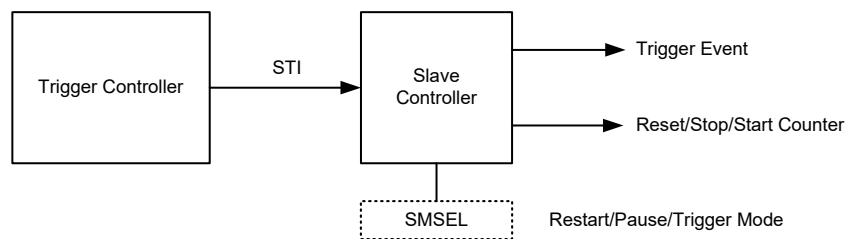
**Level Trigger Source = Internal (ITIx) + Channel input (TIn) + Software UEVG bit**



**Figure 39. Trigger Control Block**

## Slave Controller

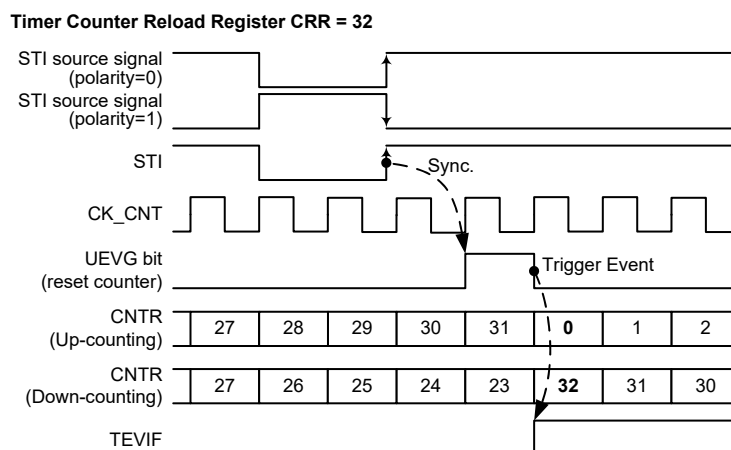
The GPTM can be synchronized with an external trigger in several modes including the Restart mode, the Pause mode and the Trigger mode which can be selected by the SMSEL field in the MDCFR register. The trigger input of these modes comes from the STI signal which is selected by the TRSEL field in the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



**Figure 40. Slave Controller Diagram**

### Restart Mode

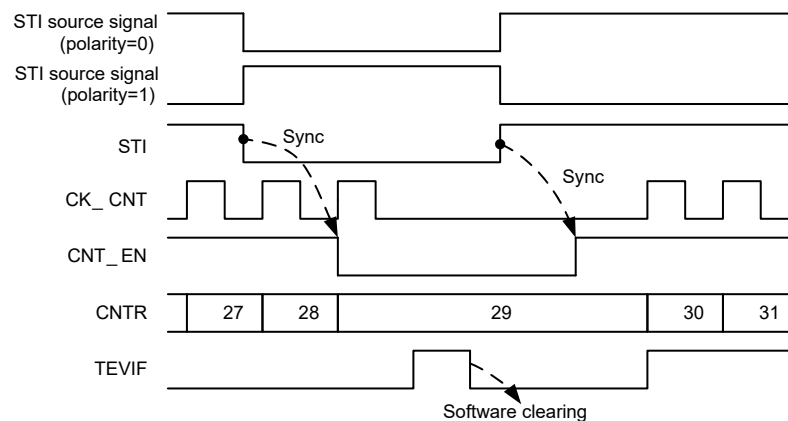
The counter and its prescaler can be reinitialized in response to a rising edge of the STI signal. When an STI rising edge occurs, the update event software generation bit named UEVG will automatically be asserted by hardware and the trigger event flag will also be set. Then the counter and prescaler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



**Figure 41. GPTM in Restart Mode**

### Pause Mode

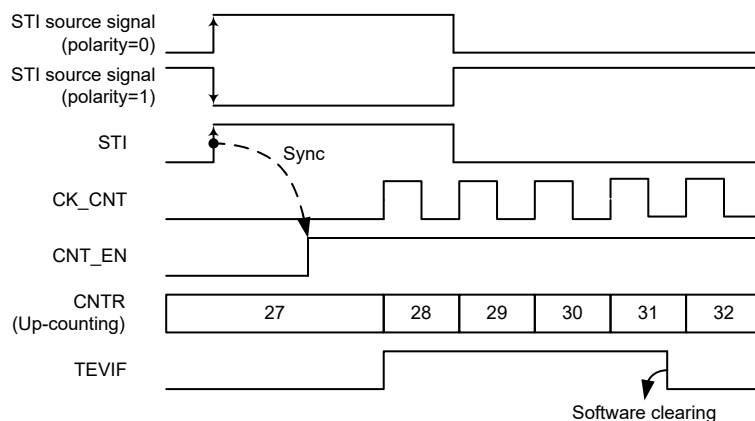
In the Pause Mode, the selected STI input signal level is used to control the counter start/stop operation. The counter starts to count when the selected STI signal is at a high level and stops counting when the STI signal is changed to a low level, here the counter will maintain its present value and 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 42. GPTM 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 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.

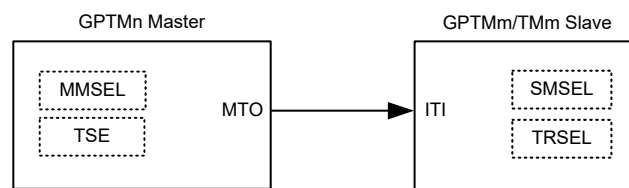


**Figure 43. GPTM in Trigger Mode**



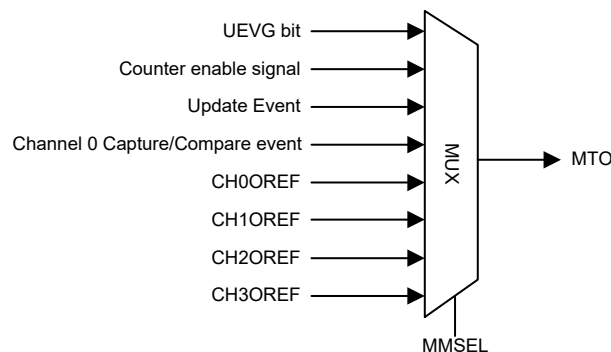
## Master Controller

The GPTMs and TMs 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 TM, if exists, which is configured in the Slave Mode.



**Figure 44. Master GPTMn and Slave GPTMm/TMm Connection**

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



**Figure 45. MTO Selection**

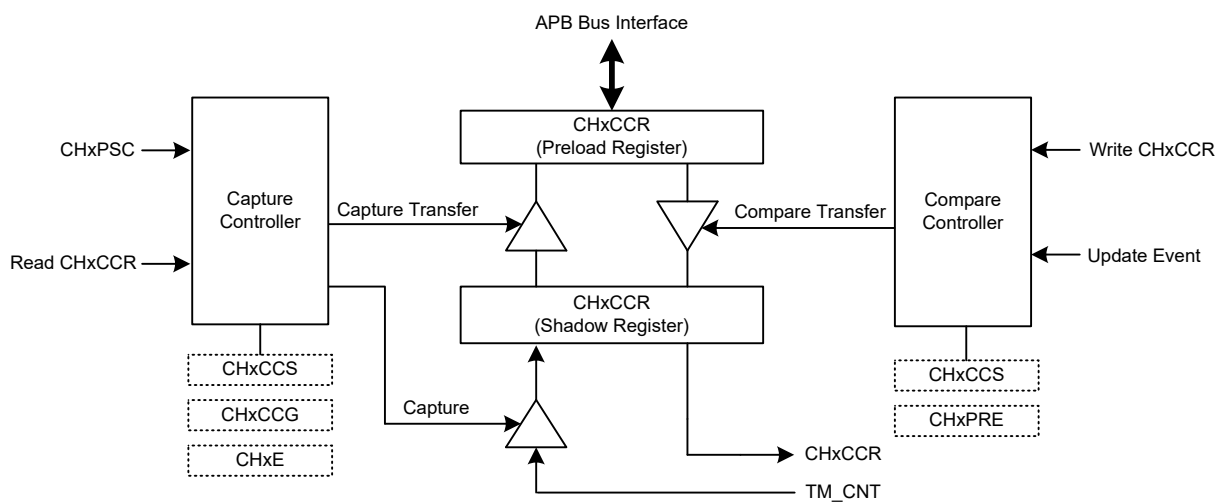
For example, setting the MMSEL field to 0x5 is to select the CH1OREF signal as the MTO signal to synchronize another slave GPTM or TM. For a more detailed description, refer to the related MMSEL field definitions in the MDCFR register.

## Channel Controller

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

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

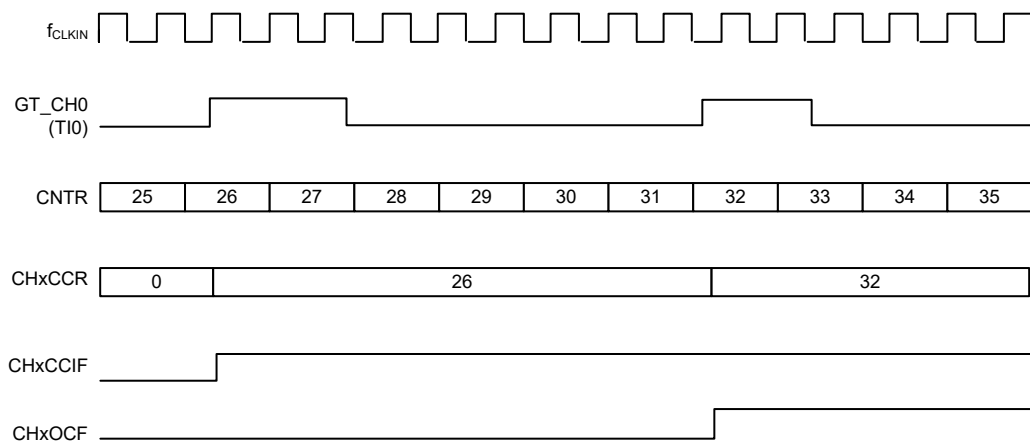
When used in the compare match output mode, the contents of the CHxCCR preload register is copied into the associated shadow register, the counter value is then compared with the register value.



**Figure 46. 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.



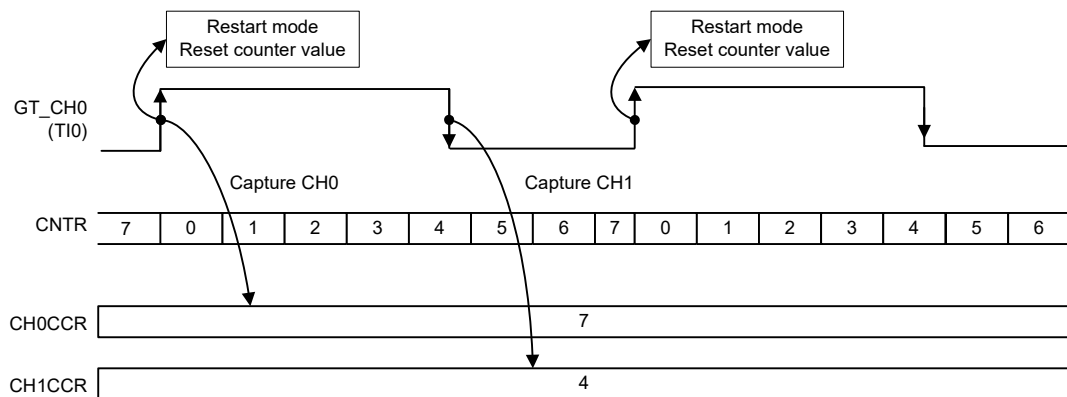
**Figure 47. Input Capture Mode**

### Pulse Width Measurement

The input capture mode can be also used for pulse width measurement from signals on the  $GT\_CHx$  pins,  $TIx$ . The following example shows how to configure the GPTM operated in the input capture mode to measure the high pulse width and the input period on the  $GT\_CH0$  pin using channel 0 and channel 1. The basic steps are shown as follows.

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

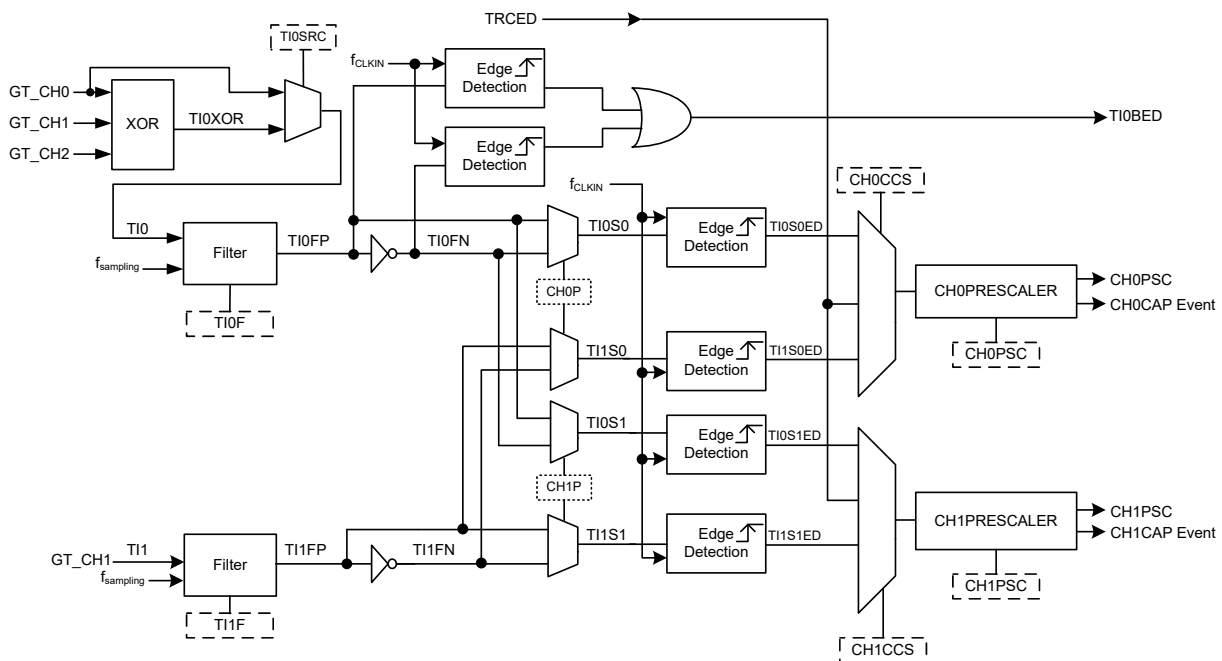
As the following diagram shows, the high pulse width on the  $GT\_CH0$  pin will be captured into the  $CH1CCR$  register while the input period will be captured into the  $CH0CCR$  register after input capture operation.



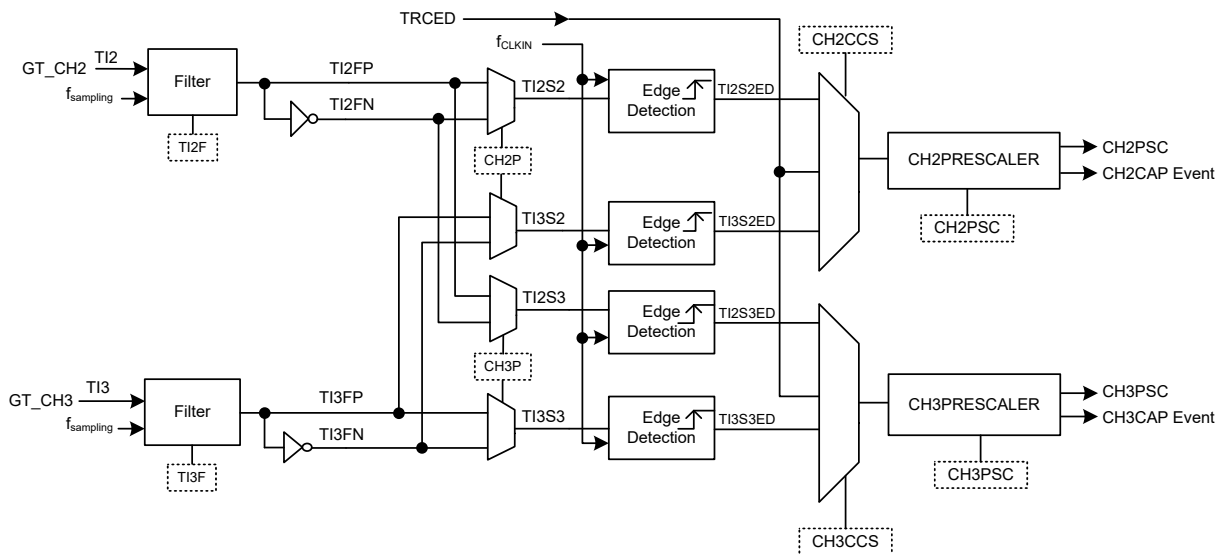
**Figure 48. PWM Pulse Width Measurement Example**

## Input Stage

The input stage consists of a digital filter, a channel polarity selection, edge detection and a channel prescaler. The channel 0 input signal (TIO) can be chosen to come from the GT\_CH0 signal or the Exclusive-OR function of the GT\_CH0, GT\_CH1 and GT\_CH2 signals. The channel input signal (TIX) is sampled by a digital filter to generate a filtered input signal TIXFP. Then the channel polarity and the edge detection block can generate a TIXS0ED or TIXS1ED signal for the input capture function. The effective input event number can be set by the channel capture input source prescaler setting field, CHxPSC.



**Figure 49. Channel 0 and Channel 1 Input Stages**

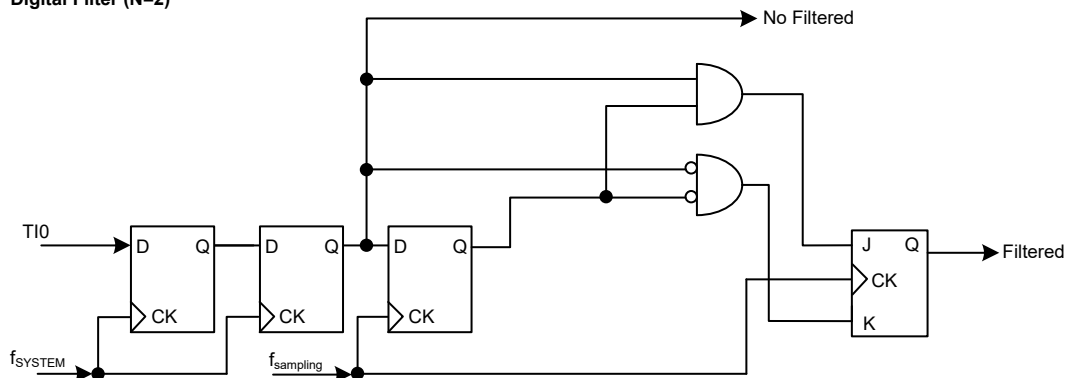


**Figure 50. Channel 2 and Channel 3 Input Stages**

### Digital Filter

The digital filters are embedded in the input stage for the GT\_CH0 ~ GT\_CH3 pins respectively. The digital filter in the GPTM is an N-event counter where N refers to how many valid transitions are necessary to output a filtered signal. The N value can be 0, 2, 4, 5, 6 or 8 according to the user selection for each filter by setting the T1xF field in the CHxICFR register.

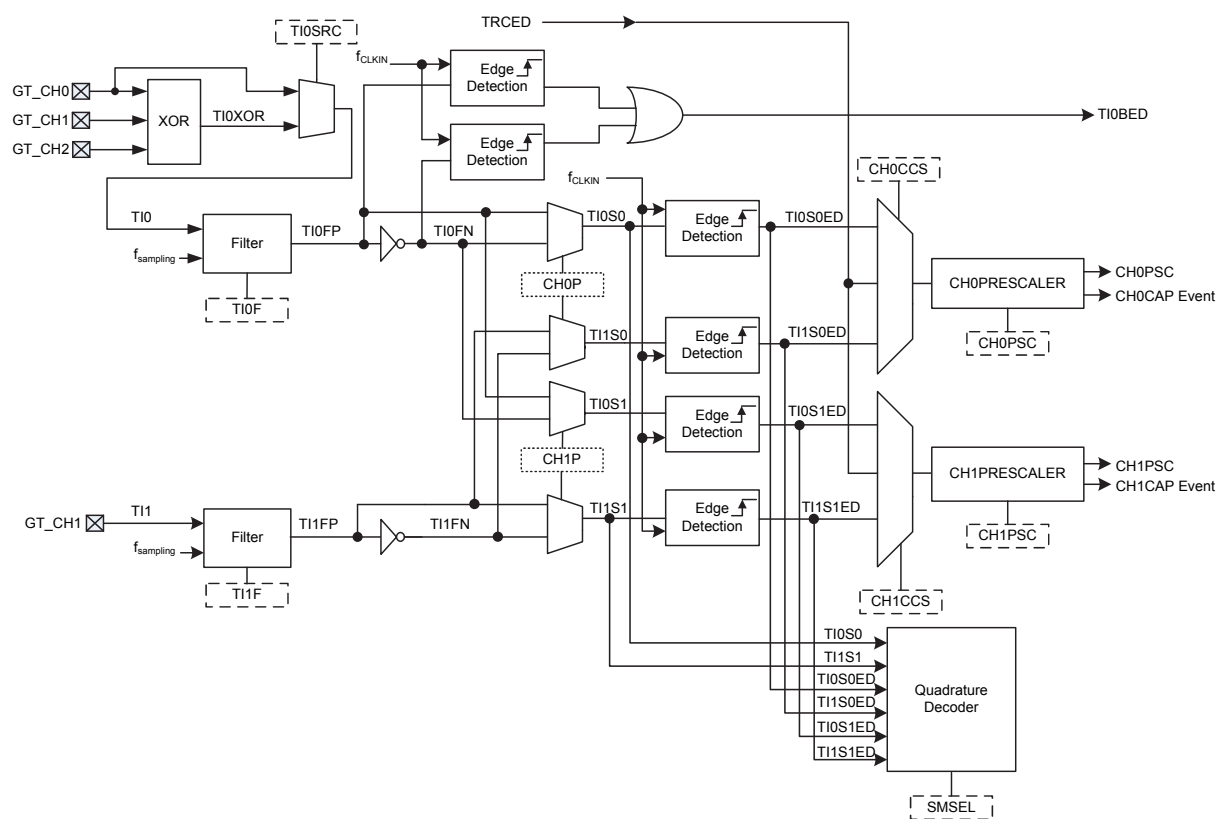
#### Digital Filter (N=2)



**Figure 51. T10 Digital Filter Diagram with N = 2**

## Quadrature Decoder

The Quadrature Decoder function uses two quadrantal inputs TI0 and TI1 derived from the GT\_CH0 and GT\_CH1 pins respectively to interact to generate the counter value. The DIR bit is modified by hardware automatically during each input source transition. The input source can be either TI0 only, TI1 only or both TI0 and TI1, the selection made by setting the SMSEL field to 0x1, 0x2 or 0x3. The mechanism for changing the counter direction is shown in the following table. The Quadrature decoder can be regarded as an external clock with a directional selection. This means that the counter counts continuously in the interval between 0 and the counter-reload value. Therefore, users must configure the CRR register before the counter starts to count.

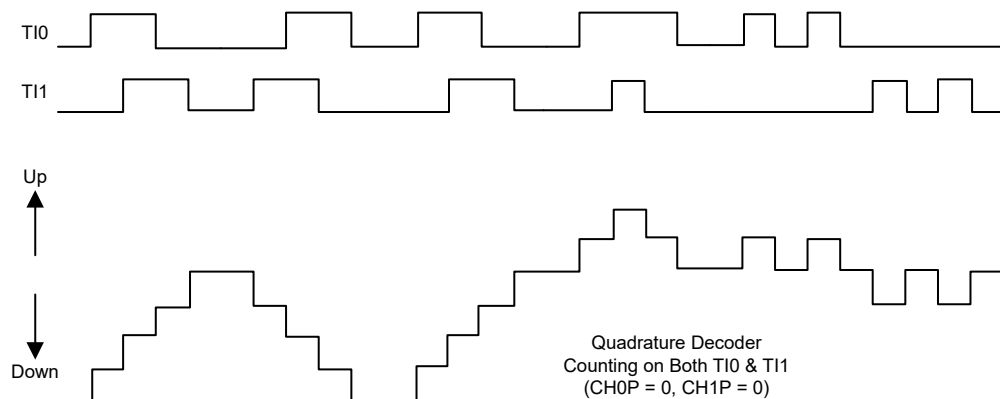


**Figure 52. Input Stage and Quadrature Decoder Block Diagram**

**Table 31. Counting Direction and Encoding Signals**

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

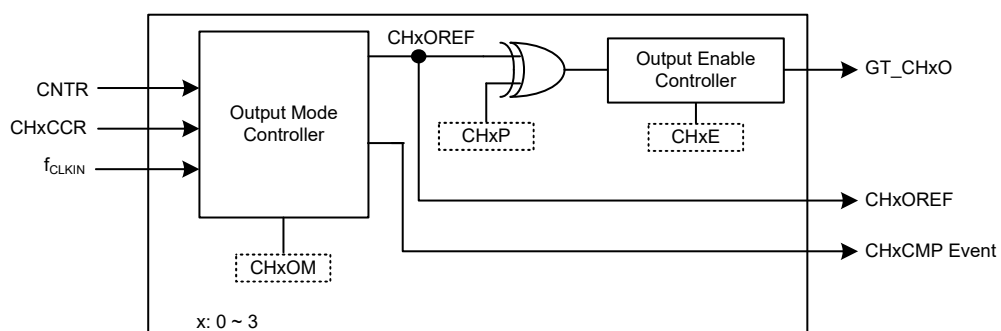
**Note:** “—” → means “no counting”; “X” → impossible



**Figure 53. Both TI0 and TI1 Quadrature Decoder Counting**

## Output Stage

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



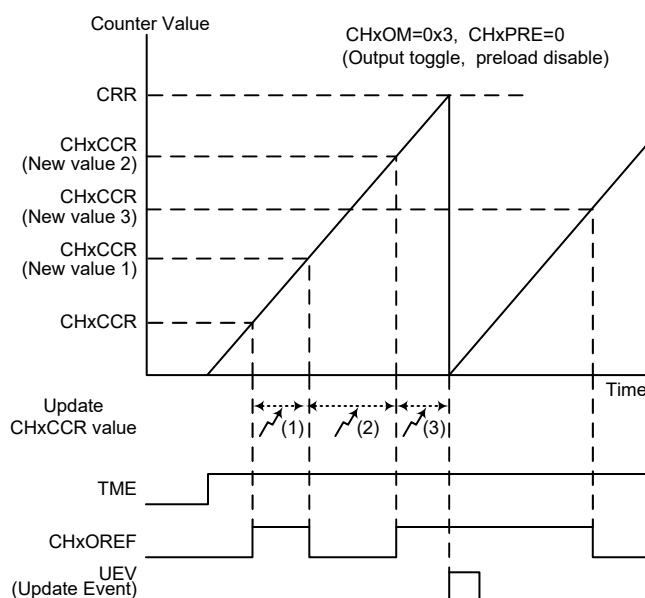
**Figure 54. Output Stage Block Diagram**

### Channel Output Reference Signal

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

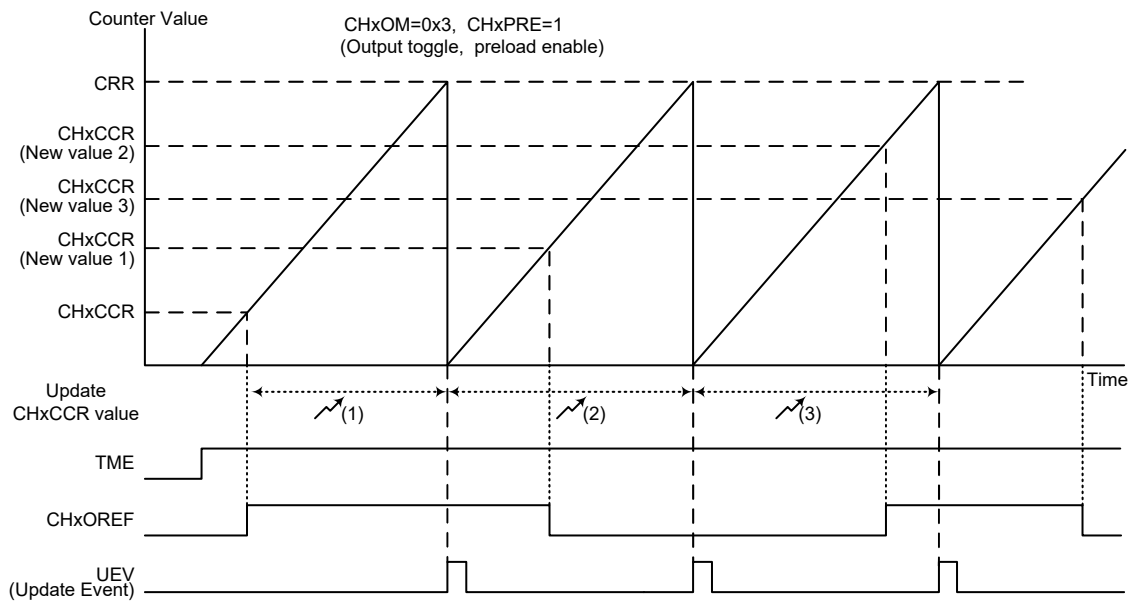
### Table 32. Compare Match Output Setup

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

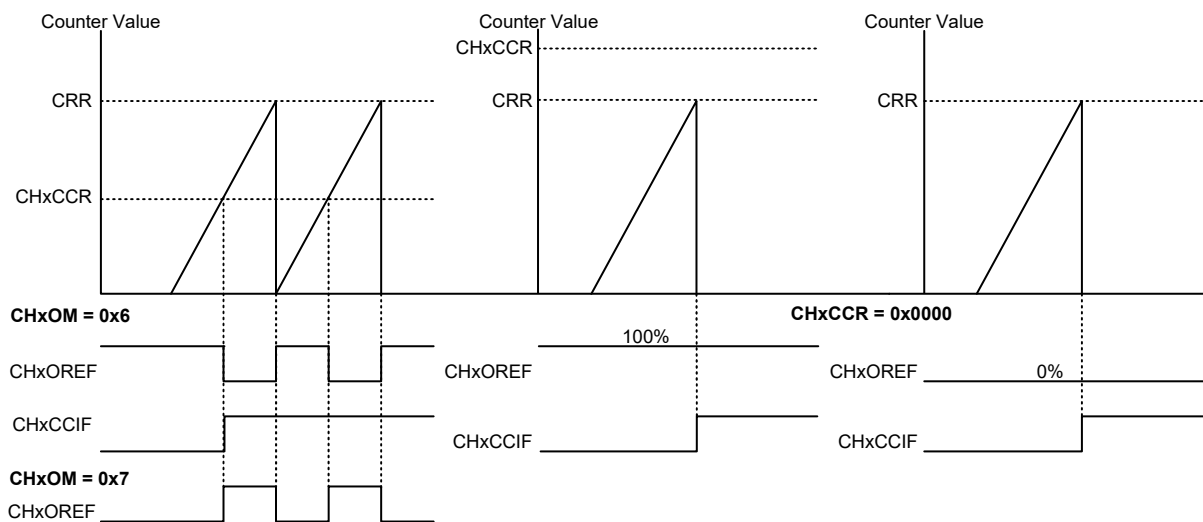


**Figure 55. Toggle Mode Channel Output Reference Signal (CHxPRE = 0)**





**Figure 56. Toggle Mode Channel Output Reference Signal (CHxPRE = 1)**



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

The diagram illustrates the timing of CHxCCR and CHxCCIF signals during up-counting and down-counting. A vertical dashed line marks the transition point where CRR = 5.

**Up-counting (Left):** The counter values are 0, 1, 2, 3, 4. The CHxCCR signal is high for values 0, 1, and 2, and low for 3 and 4. The CHxCCIF signal is low for all values.

**Down-counting (Right):** The counter values are 5, 4, 3, 2, 1, 0, 1. The CHxCCR signal is low for values 5, 4, and 3, and high for 2, 1, 0, and 1. The CHxCCIF signal is high for all values.

**CHxCCR = 3:** The signal is high for counter values 0, 1, and 2, and low for 3, 4, 5, 4, 3, 2, 1, 0, and 1.

**CHxCCIF:** The signal is low for counter values 0, 1, 2, 3, and 4, and high for 5, 4, 3, 2, 1, 0, and 1.

**CHxCCR = 4:** The signal is high for counter values 0, 1, 2, and 3, and low for 4, 5, 4, 3, 2, 1, 0, and 1.

**CHxCCIF:** The signal is low for counter values 0, 1, 2, 3, and 4, and high for 5, 4, 3, 2, 1, 0, and 1.

**CHxCCR >= 5:** The signal is low for counter values 0, 1, 2, 3, and 4, and high for 5, 4, 3, 2, 1, 0, and 1.

**CHxCCIF:** The signal is low for counter values 0, 1, 2, 3, and 4, and high for 5, 4, 3, 2, 1, 0, and 1.

**CHxCCR = 0:** The signal is high for counter values 0, 1, 2, 3, and 4, and low for 5, 4, 3, 2, 1, 0, and 1.

**CHxCCIF:** The signal is low for counter values 0, 1, 2, 3, and 4, and high for 5, 4, 3, 2, 1, 0, and 1.

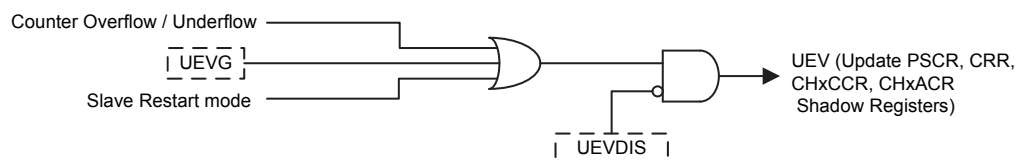
September 30, 2021

## Update Management

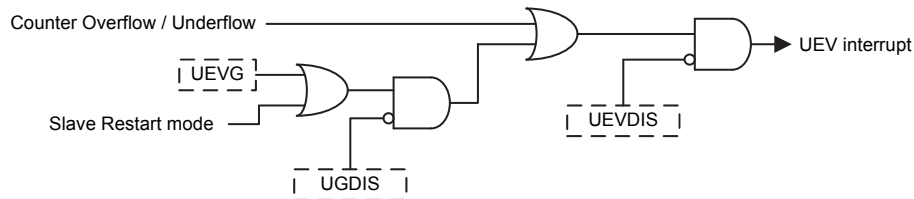
The Update event is used to update the CRR, the PSCR, the CHxACR and the CHxCCR values from the actual registers to the corresponding shadow registers. An update event occurs when the counter overflows or underflows, the software update control bit is triggered or an update event from the slave controller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

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

### Update Event Management



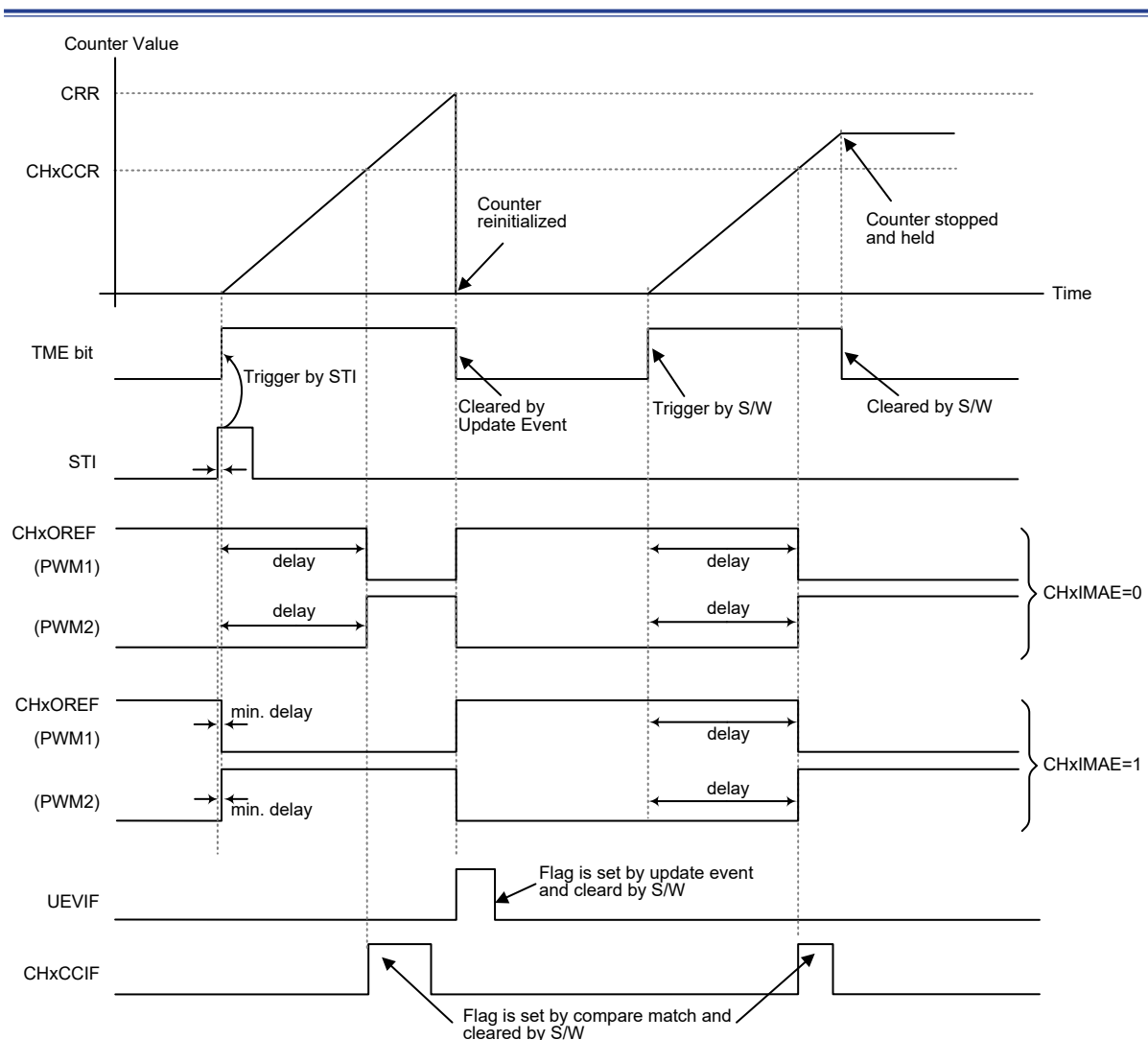
### Update Event Interrupt Management



**Figure 60. Update Event Setting Diagram**

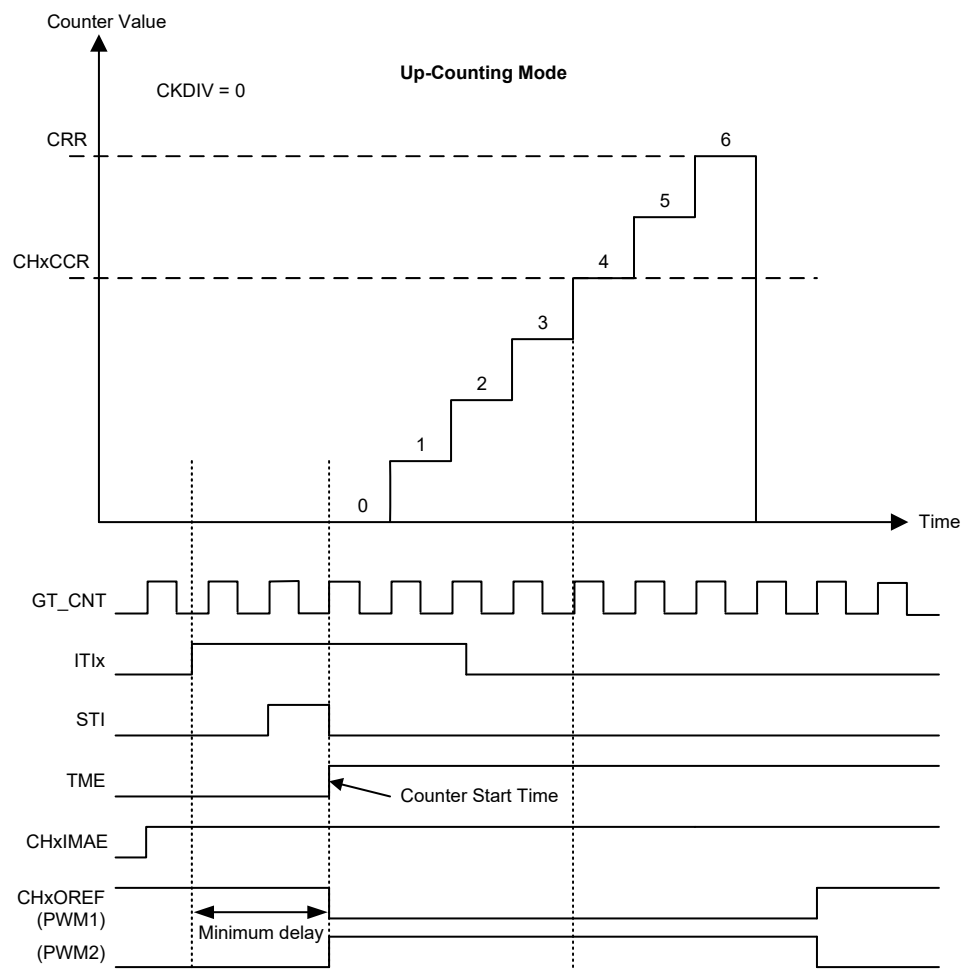
## Single Pulse Mode

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



**Figure 61. Single Pulse Mode**

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

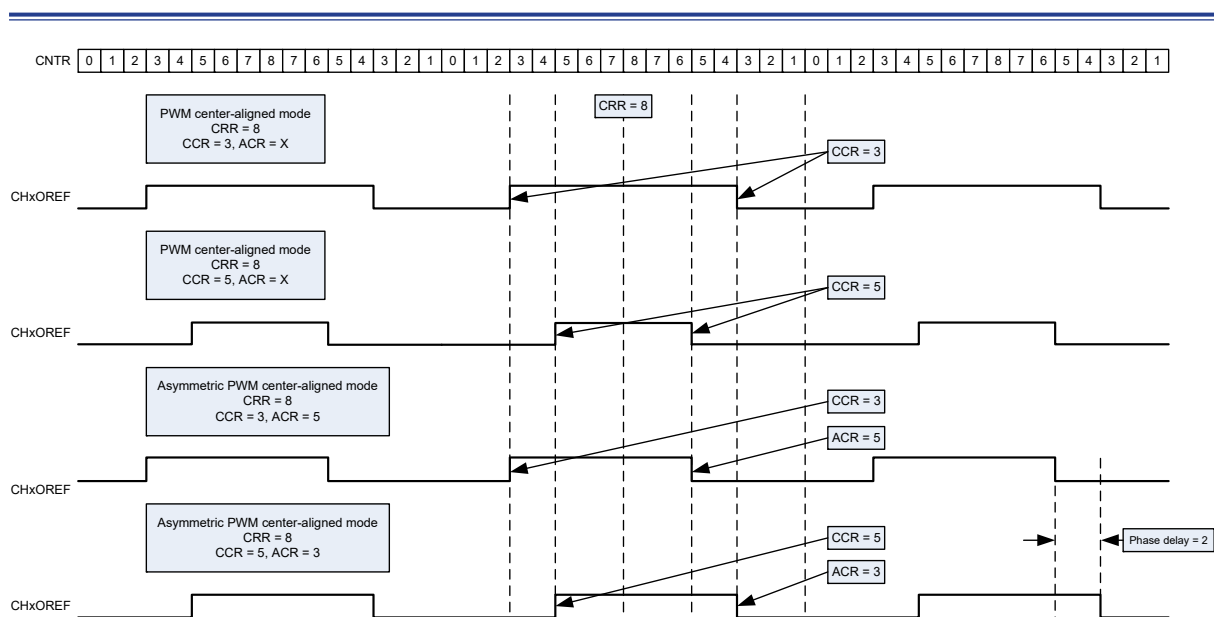


**Figure 62. Immediate Active Mode Minimum Delay**

## Asymmetric PWM Mode

Asymmetric PWM mode allows two center-aligned PWM signals to be generated with a programmable phase shift. While the PWM frequency is determined by the value of the CRR register, the duty cycle and the phase-shift are determined by the CHxCCR and CHxACR register. When the counter is counting up, the PWM uses the value in CHxCCR as up-count compare value. When the counter is into counting down stage, the PWM uses the value in CHxACR as down-count compare value. The Figure 63 is shown as an example for asymmetric PWM mode in center-aligned counting mode.

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



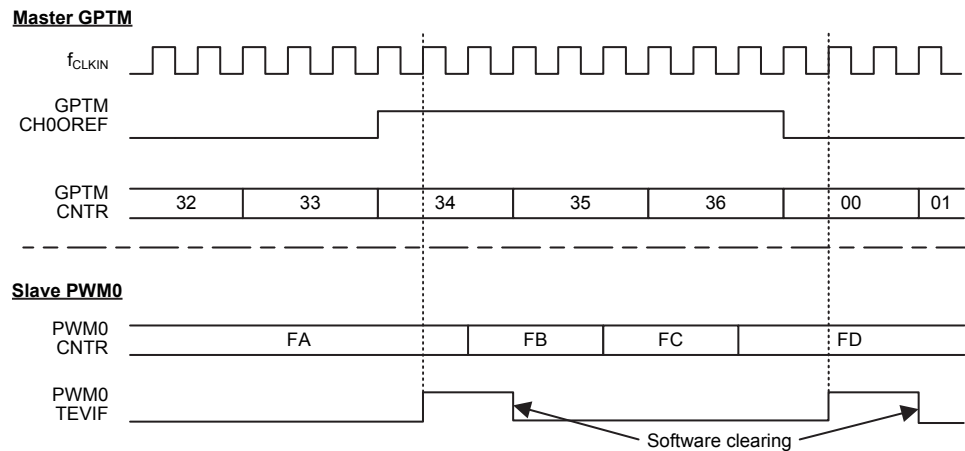
**Figure 63. Asymmetric PWM Mode versus Center-Aligned Counting Mode**

## Timer Interconnection

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

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

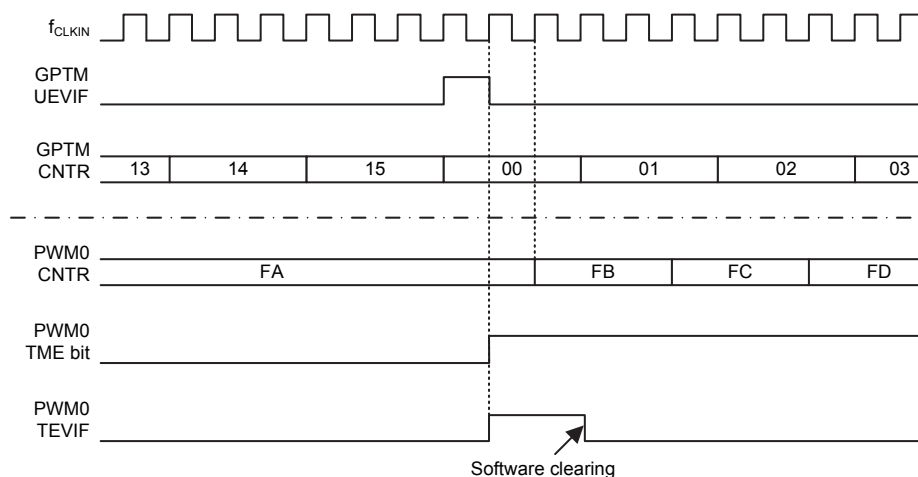
- Configure GPTM as the master mode to send its channel 0 Output Reference signal CH0OREF as a trigger output (MMSEL = 0x4).
- Configure GPTM CH0OREF waveform.
- Configure PWM0 to receive its input trigger source from the GPTM trigger output (TRSEL = 0xA).
- Configure PWM0 to operate in the pause mode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



**Figure 64. Pausing PWM0 using the GPTM CH0OREF Signal**

#### Using one Timer to Trigger another Timer Start Counting

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

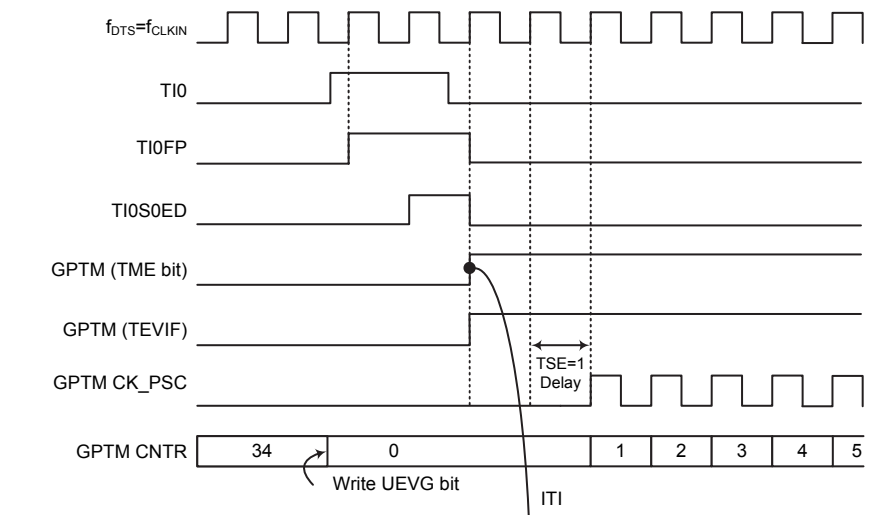


**Figure 65. Triggering PWM0 with GPTM Update Event**

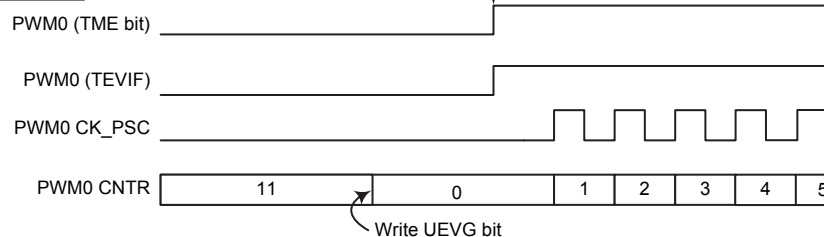
### Starting Two Timers Synchronously in Response to an External Trigger

- Configure GPTM to operate in the master mode to send its enable signal as a trigger output (MMSEL = 0x1).
- Configure GPTM slave mode to receive its input trigger source from GT\_CH0 pin (TRSEL = 0x1).
- Configure GPTM to be in the slave trigger mode (SMSEL = 0x6).
- Enable the GPTM master timer synchronization function by setting the TSE bit in the MDCFR register to 1 to synchronize the slave timer.
- Configure PWM0 to receive its input trigger source from the GPTM trigger output (TRSEL = 0xA).
- Configure PWM0 to be in the slave trigger mode (SMSEL = 0x6).

#### Master GPTM



#### Slave PWM0



**Figure 66. Trigger GPTM and PWM0 with the GPTM CH0 Input**

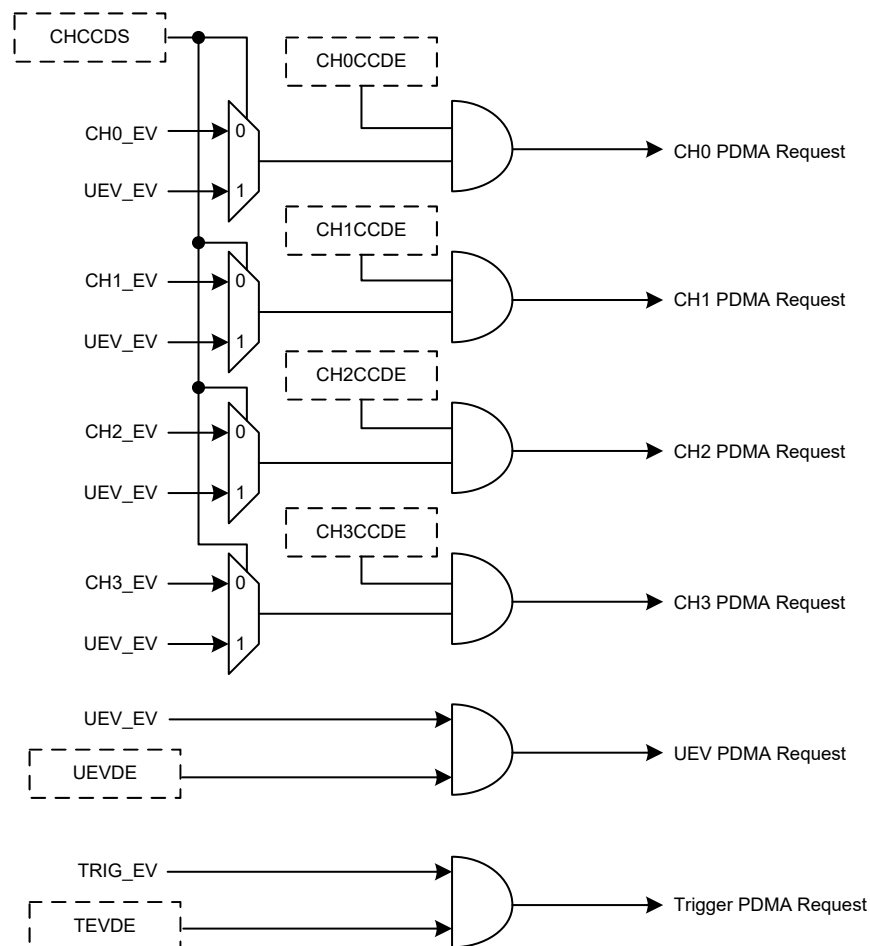


## Trigger Peripherals Start

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

## 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 67. GPTM PDMA Mapping Diagram**

## Register Map

The following table shows the GPTM registers and reset values.

**Table 33. GPTM Register Map**

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

## Register Descriptions

### Timer Counter Configuration Register – CNTCFR

This register specifies the GPTM counter configuration.

Offset: 0x000

Reset value: 0x0000\_0000

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

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

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

### Timer Mode Configuration Register – MDCFR

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

Offset: 0x004

Reset value: 0x0000\_0000

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

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

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

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

## Timer Trigger Configuration Register – TRCFR

This register specifies the trigger source selection of GPTM.

Offset: 0x008

Reset value: 0x0000\_0000

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

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

**Table 34. GPTM Internal Trigger Connection**

Slave Timing Module	ITI0	ITI1	ITI2
GPTM	PWM0	—	PWM1

## Timer Control Register – CTR

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

Offset: 0x010

Reset value: 0x0000\_0000

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

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



## Channel 0 Input Configuration Register – CH0ICFR

This register specifies the channel 0 input mode configuration.

Offset: 0x020

Reset value: 0x0000\_0000

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

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

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

### Channel 1 Input Configuration Register – CH1ICFR

This register specifies the channel 1 input mode configuration.

Offset: 0x024

Reset value: 0x0000\_0000

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

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

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

## Channel 2 Input Configuration Register – CH2ICFR

This register specifies the channel 2 input mode configuration.

Offset: 0x028

Reset value: 0x0000\_0000

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

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

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

### Channel 3 Input Configuration Register – CH3ICFR

This register specifies the channel 3 input mode configuration.

Offset: 0x02C

Reset value: 0x0000\_0000

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

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

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

## Channel 0 Output Configuration Register – CH0OCFR

This register specifies the channel 0 output mode configuration.

Offset: 0x040

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[5]	CH0IMAE	<p>Channel 0 Immediate Active Enable</p> <p>0: No action</p> <p>1: Single pulse Immediate Active Mode is enabled</p> <p>The CH0OREF signal will be forced to the compare matched level immediately after an available trigger event occurs irrespective of the result of the comparison between the CNTR and the CH0CCR values.</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 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 the update event occurs.</p>

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



## Channel 1 Output Configuration Register – CH1OCFR

This register specifies the channel 1 output mode configuration.

Offset: 0x044

Reset value: 0x0000\_0000

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

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

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

## Channel 2 Output Configuration Register – CH2OCFR

This register specifies the channel 2 output mode configuration.

Offset: 0x048

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[5]	CH2IMAE	<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 signal will be forced to the compare matched level immediately after an available trigger event occurs irrespective of the result of the comparison between the CNTR and the CH2CCR values.</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 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>

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

## Channel 3 Output Configuration Register – CH3OCFR

This register specifies the channel 3 output mode configuration.

Offset: 0x04C

Reset value: 0x0000\_0000

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

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 signal will be forced to the compare matched level immediately after an available trigger event occurs irrespective of the result of the comparison between the CNTR and the CH3CCR values.</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 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>

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

## Channel Control Register – CHCTR

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

Offset: 0x050

Reset value: 0x0000\_0000

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

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

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

### Channel Polarity Configuration Register – CHPOLR

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

Offset: 0x054

Reset value: 0x0000\_0000

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

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



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

### Timer PDMA/Interrupt Control Register – DICTR

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

Offset: 0x074

Reset value: 0x0000\_0000

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

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

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

## Timer Event Generator Register – EVGR

This register contains the software event generation bits.

Offset: 0x078

Reset value: 0x0000\_0000

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

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

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

### Timer Interrupt Status Register – INTSR

This register stores the timer interrupt status.

Offset: 0x07C

Reset value: 0x0000\_0000

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

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

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

Bits	Field	Descriptions
[1]	CH1CCIF	<p>Channel 1 Capture/Compare Interrupt Flag</p> <ul style="list-style-type: none"> <li>- Channel 1 is configured as an output: <ul style="list-style-type: none"> <li>0: No match event occurs</li> <li>1: The content of the counter CNTR has matched the content 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> <li>- Channel 1 is configured as an input: <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> </li> </ul>
[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 content of the counter CNTR has matched the content of the CH0CCR register</li> </ul> <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> </li> <li>- Channel 0 is configured as an input: <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 CH0CCR register.</p> </li> </ul>

## Timer Counter Register – CNTR

This register stores the timer counter value.

Offset: 0x080

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[15:0]	CNTV	Counter Value

## Timer Prescaler Register – PSCR

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

Offset: 0x084

Reset value: 0x0000\_0000

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

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



## Timer Counter-Reload Register – CRR

This register specifies the timer counter-reload value.

Offset: 0x088

Reset value: 0x0000\_FFFF

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

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

## Channel 0 Capture/Compare Register – CH0CCR

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

Offset: 0x090

Reset value: 0x0000\_0000

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

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

## Channel 1 Capture/Compare Register – CH1CCR

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

Offset: 0x094

Reset value: 0x0000\_0000

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

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

## Channel 2 Capture/Compare Register – CH2CCR

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

Offset: 0x098

Reset value: 0x0000\_0000

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

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

## Channel 3 Capture/Compare Register – CH3CCR

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

Offset: 0x09C

Reset value: 0x0000\_0000

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

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

## Channel 0 Asymmetric Compare Register – CH0ACR

This register specifies the timer channel 0 asymmetric compare value.

Offset: 0x0A0

Reset value: 0x0000\_0000

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

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

## Channel 1 Asymmetric Compare Register – CH1ACR

This register specifies the timer channel 1 asymmetric compare value.

Offset: 0x0A4

Reset value: 0x0000\_0000

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

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

## Channel 2 Asymmetric Compare Register – CH2ACR

This register specifies the timer channel 2 asymmetric compare value.

Offset: 0x0A8

Reset value: 0x0000\_0000

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

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

## Channel 3 Asymmetric Compare Register – CH3ACR

This register specifies the timer channel 3 asymmetric compare value.

Offset: 0x0AC

Reset value: 0x0000\_0000

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

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

# 14 Pulse-Width-Modulation Timer (PWM)

## Introduction

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

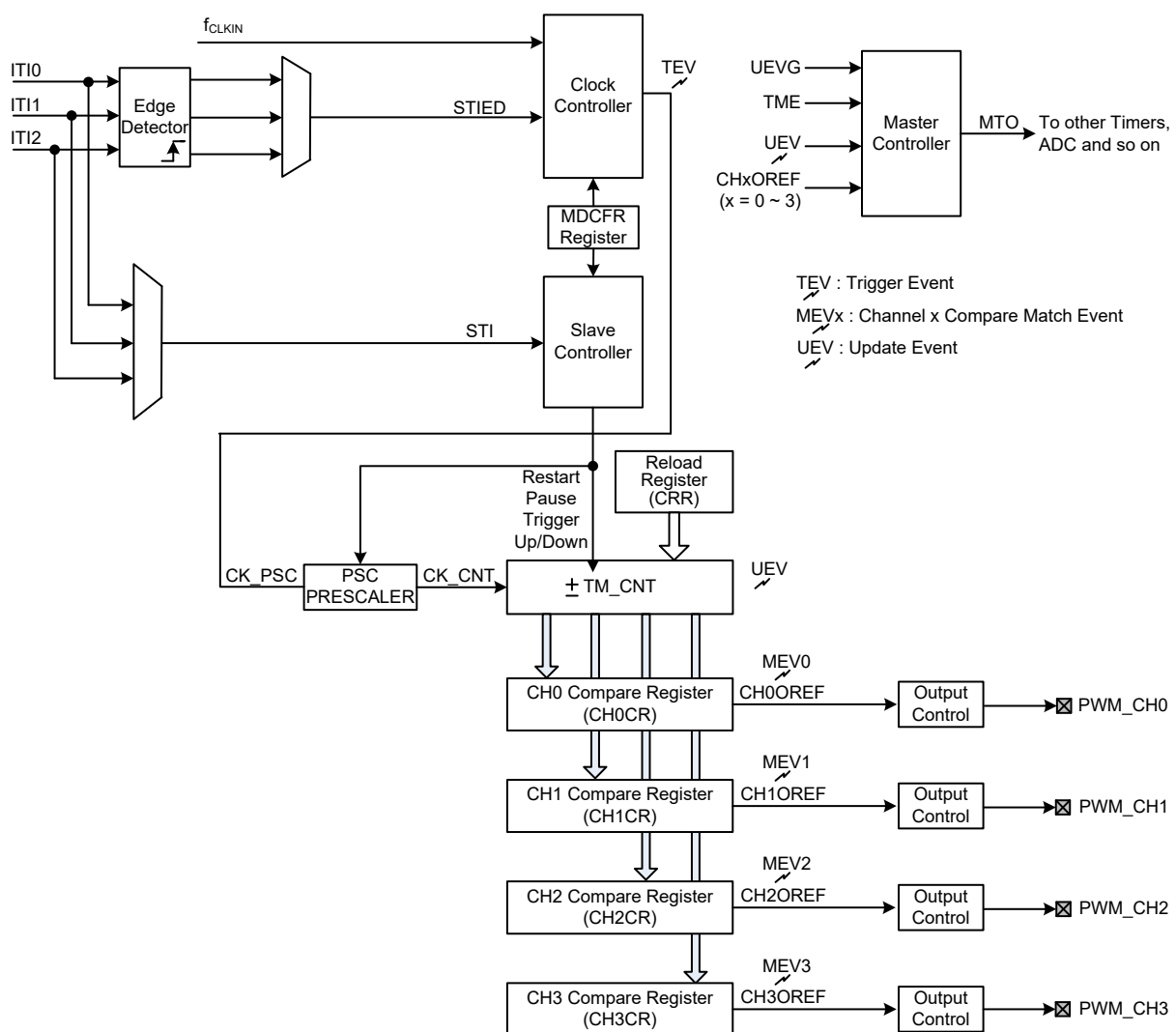


Figure 68. PWM Block Diagram



## Features

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

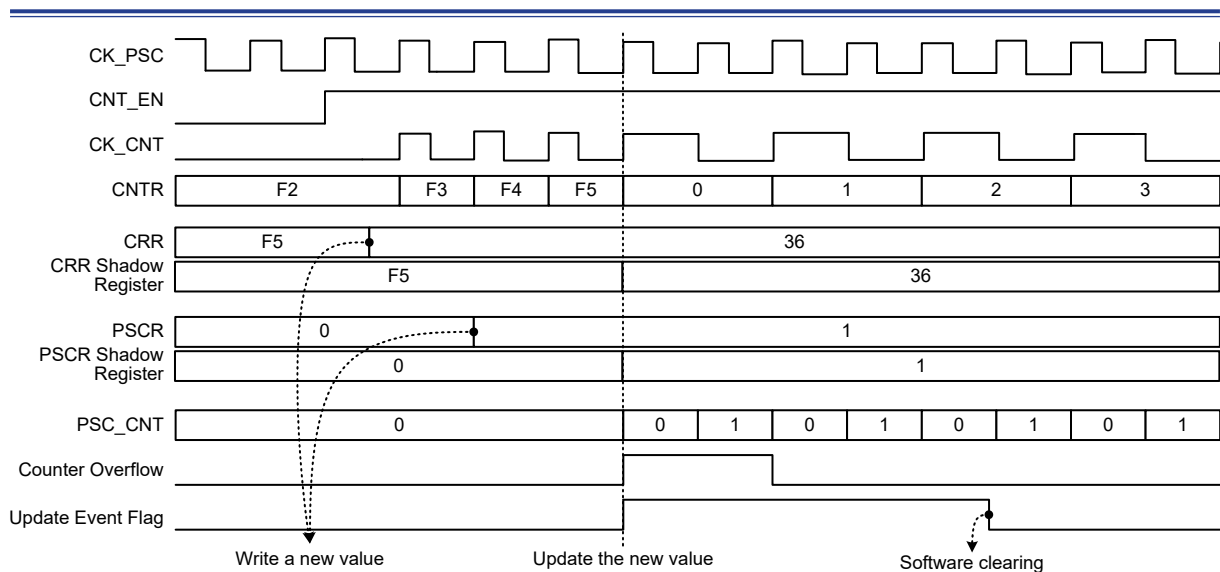
## Functional Descriptions

### Counter Mode

#### Up-Counting

In this mode the counter counts continuously from 0 to the counter-reload value, which is defined in the CRR register, in a count-up direction. Once the counter reaches the counter-reload value, then restarts from 0 and generates a counter overflow event. 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.

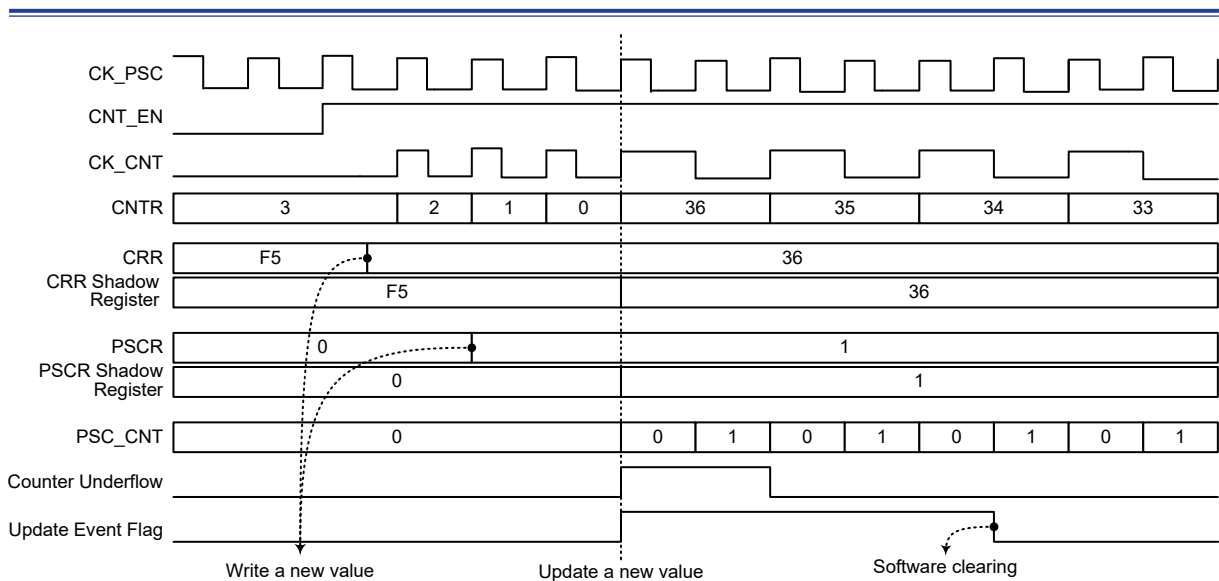


**Figure 69. Up-counting Example**

### 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, then restarts from the counter-reload value and generates a counter underflow event. 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.



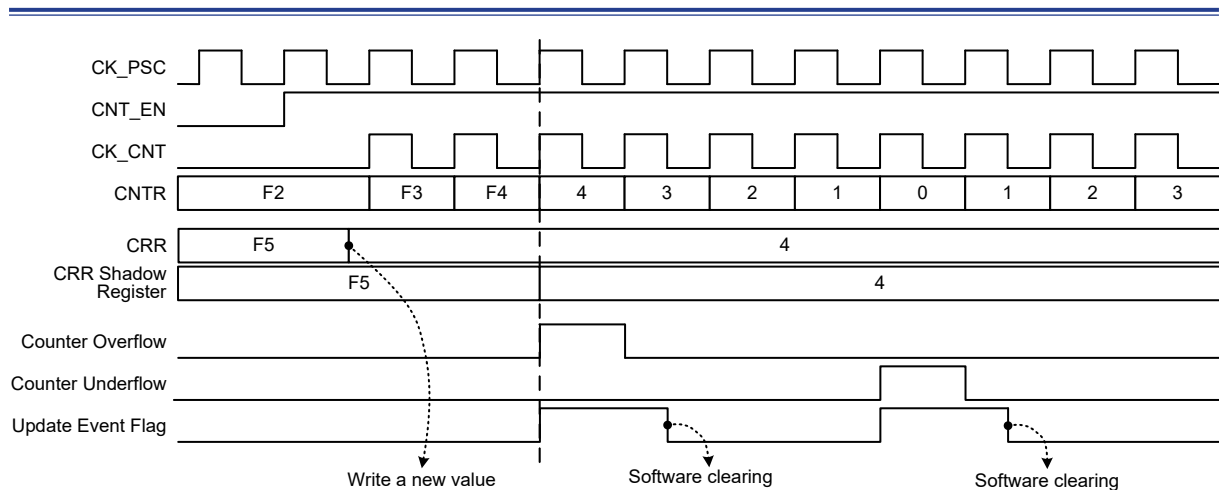
**Figure 70. Down-counting Example**

### Center-Aligned Counting

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

The update event interrupt flag bit in the INTSR register will be set to 1, when an overflow or underflow event occurs.



**Figure 71. Center-aligned Counting Example**

## Clock Controller

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

■ Internal APB clock  $f_{CLKIN}$ :

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

■ STIED:

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

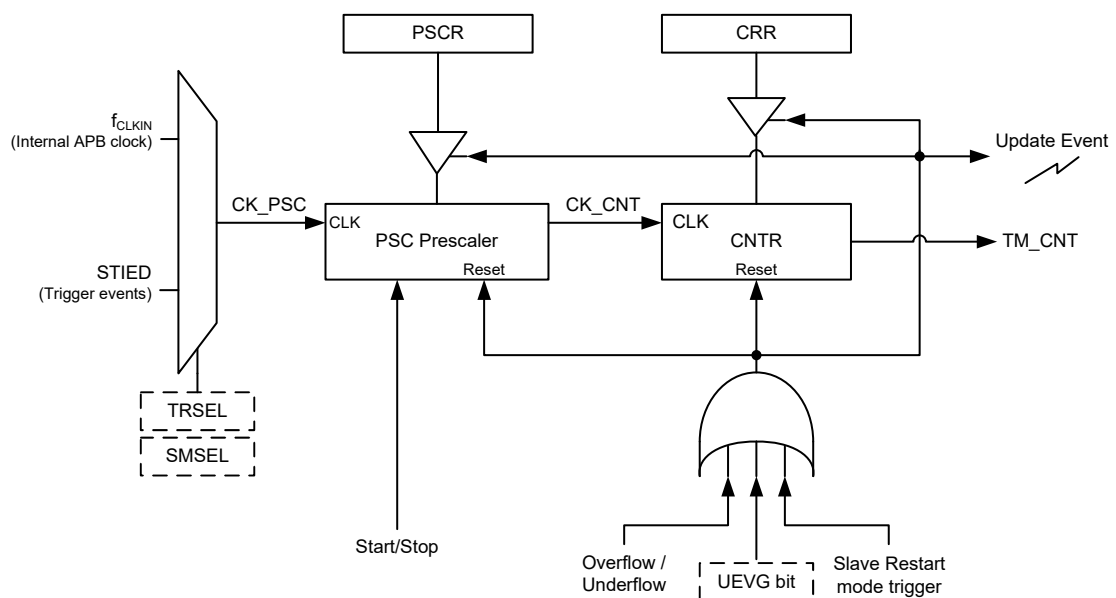
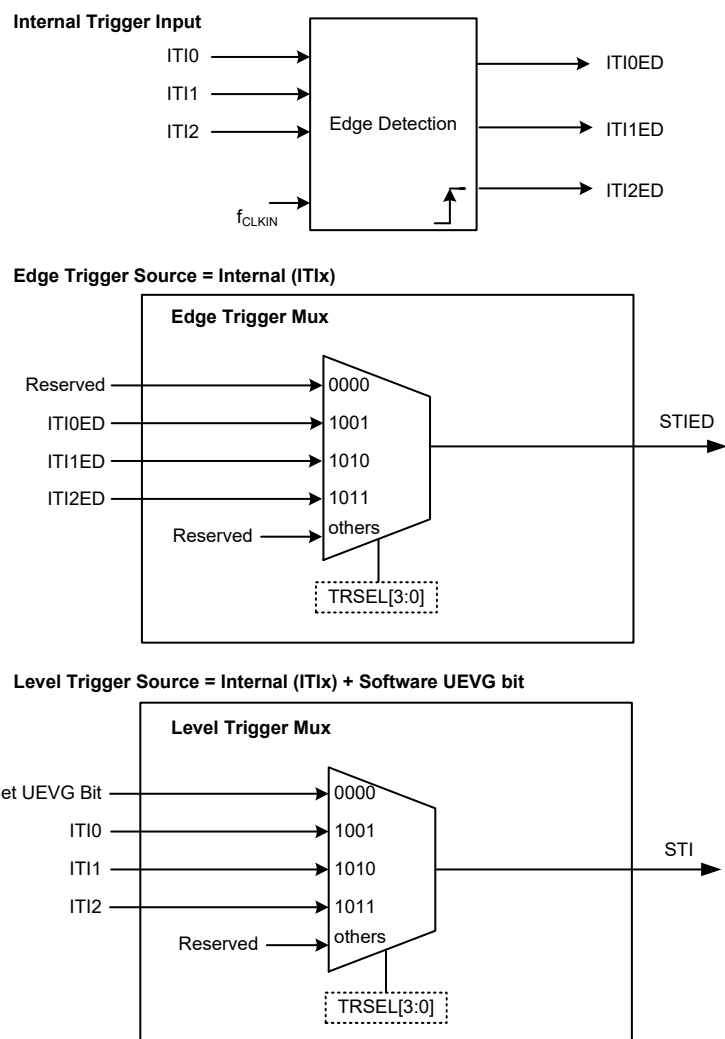


Figure 72. PWM Clock Selection Source

## Trigger Controller

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

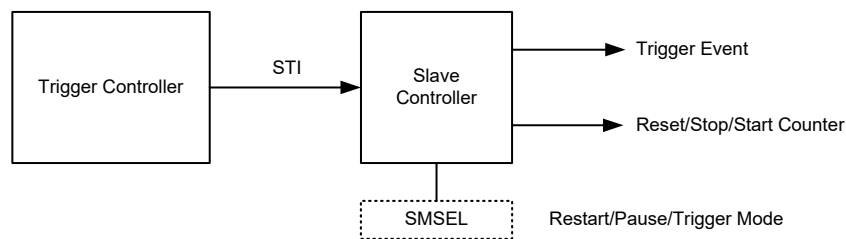
**Trigger Controller Block = Edge Trigger Mux + Level Trigger Mux**



**Figure 73. Trigger Control Block**

## Slave Controller

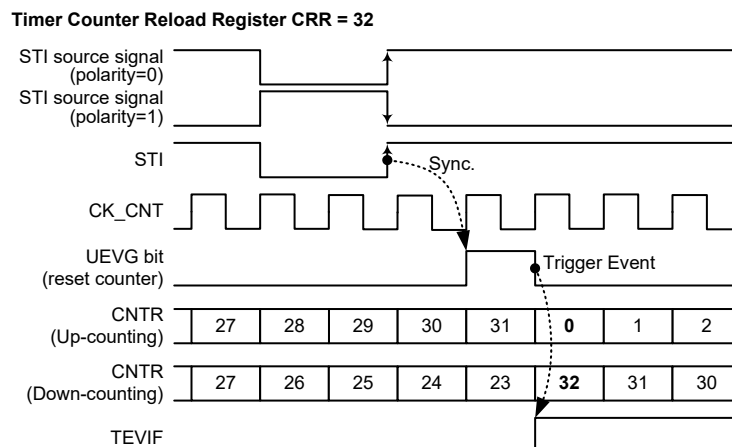
The PWM can be synchronized with an external trigger in several modes including the Restart mode, the Pause mode and the Trigger mode which is selected by the SMSEL field in the MDCFR register. The trigger input of these modes comes from the STI signal which is selected by the TRSEL field in the TRCFR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



**Figure 74. Slave Controller Diagram**

### Restart Mode

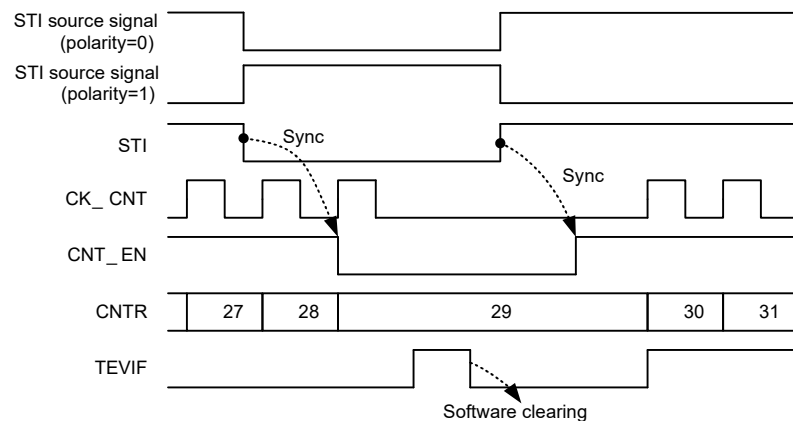
The counter and its prescaler can be reinitialized in response to a rising edge of the STI signal. When an STI rising edge occurs, the update event software generation bit named UEVG will automatically be asserted by hardware and the trigger event flag will also be set. Then the counter and prescaler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



**Figure 75. PWM in Restart Mode**

### Pause Mode

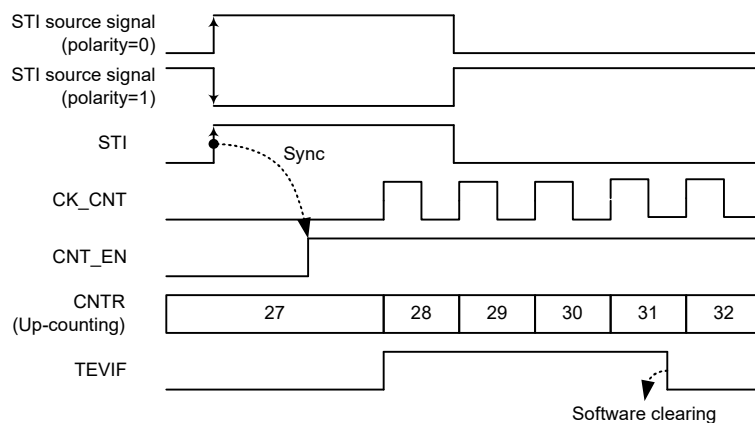
In the Pause Mode, the selected STI input signal level is used to control the counter start/stop operation. The counter starts to count when the selected STI signal is at a high level and stops counting when the STI signal is changed to a low level, here the counter will maintain its present value and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



**Figure 76. PWM in Pause Mode**

### Trigger Mode

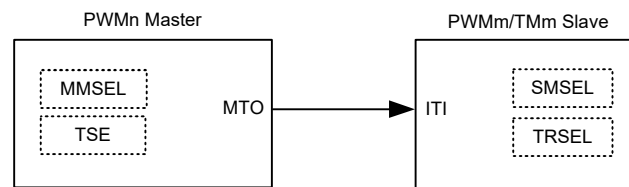
After the counter is disabled to count, the counter can resume counting when an STI rising edge signal occurs. When an STI rising edge occurs, the counter will start to count from the current value in the counter. Note that if the STI signal is selected to be derived from the UEVG bit software trigger, 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 77. PWM in Trigger Mode**

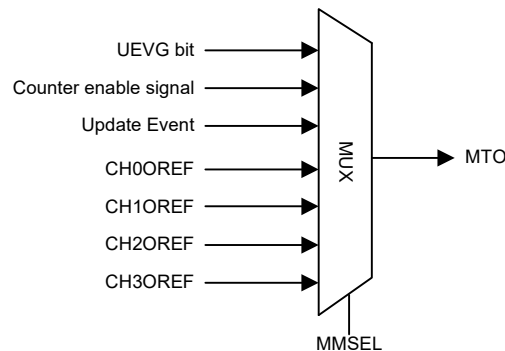
## Master Controller

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



**Figure 78. Master PWMn and Slave PWMm/TMm Connection**

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



**Figure 79. MTO Selection**

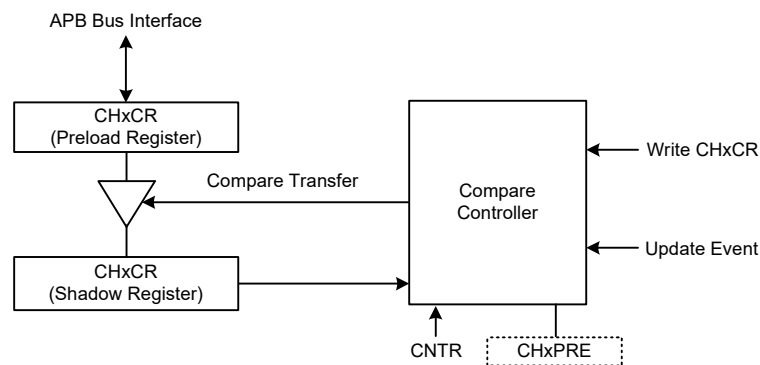
For example, setting the MMSEL field to 0x5 is to select the CH1OREF signal as the MTO signal to synchronize another slave PWM or TM. For a more detailed description, refer to the related MMSEL field definitions in the MDCFR register.



## Channel Controller

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

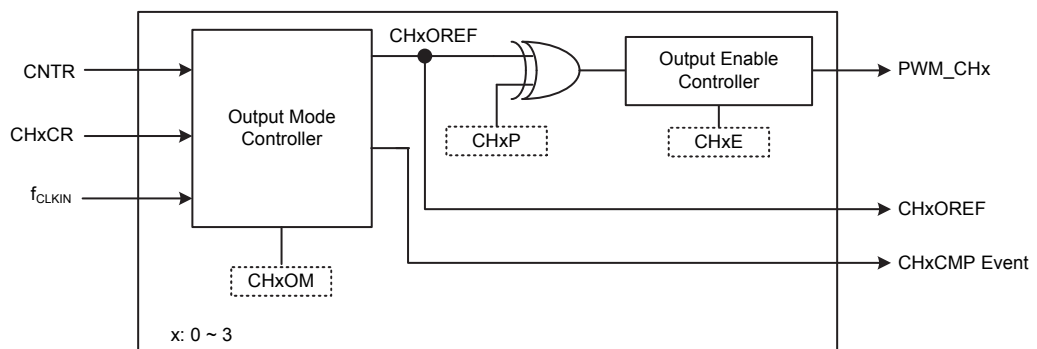
When used in the compare match output mode, the contents of the CHxCR preload register is copied into the associated shadow register; the counter value is then compared with the register value.



**Figure 80. Compare Block Diagram**

## Output Stage

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



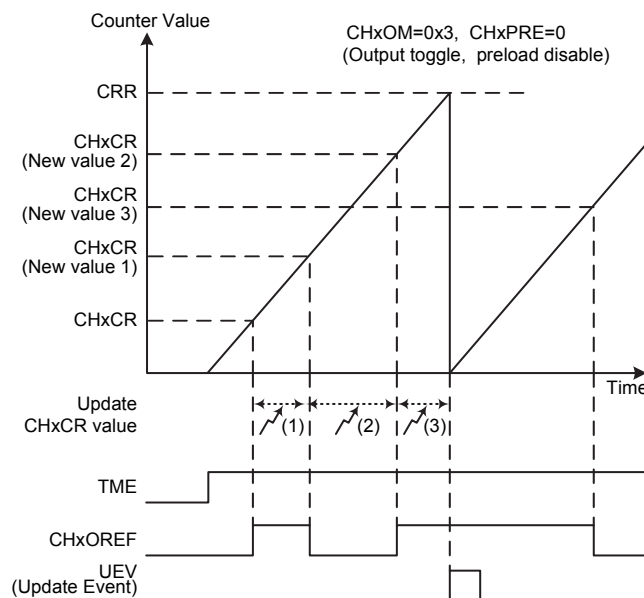
**Figure 81. Output Stage Block Diagram**

### Channel Output Reference Signal

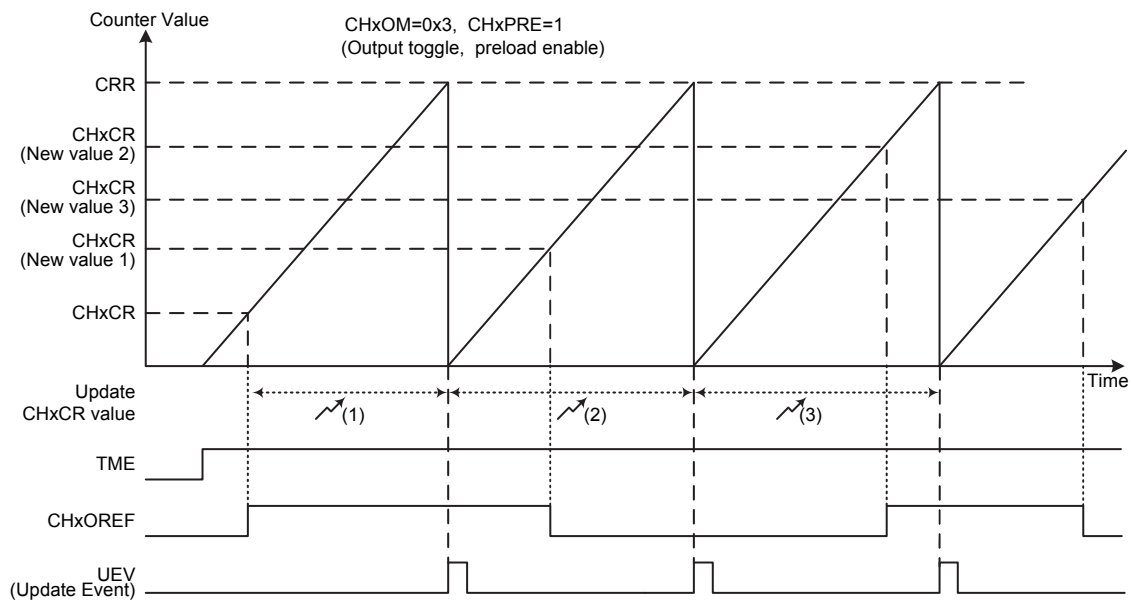
When the PWM is used in the compare match output mode, the Channel x Output Reference signal, CHxOREF, is defined by the CHxOM field setup. The CHxOREF signal has several types of output function which defines what happens to the output when the counter value matches the contents of the CHxCR register. In addition to the low, high and toggle CHxOREF output types; there are also PWM mode 1 and PWM mode 2 outputs. In these modes, the CHxOREF signal level is changed according to the count direction and the relationship between the counter value and the CHxCR content. There are also two modes which will force the output into an inactive or active state irrespective of the CHxCR content or counter values. With regard to a more detailed description refer to the relative bit definition. The accompanying Table 36 shows a summary of the output type setup.

**Table 35. Compare Match Output Setup**

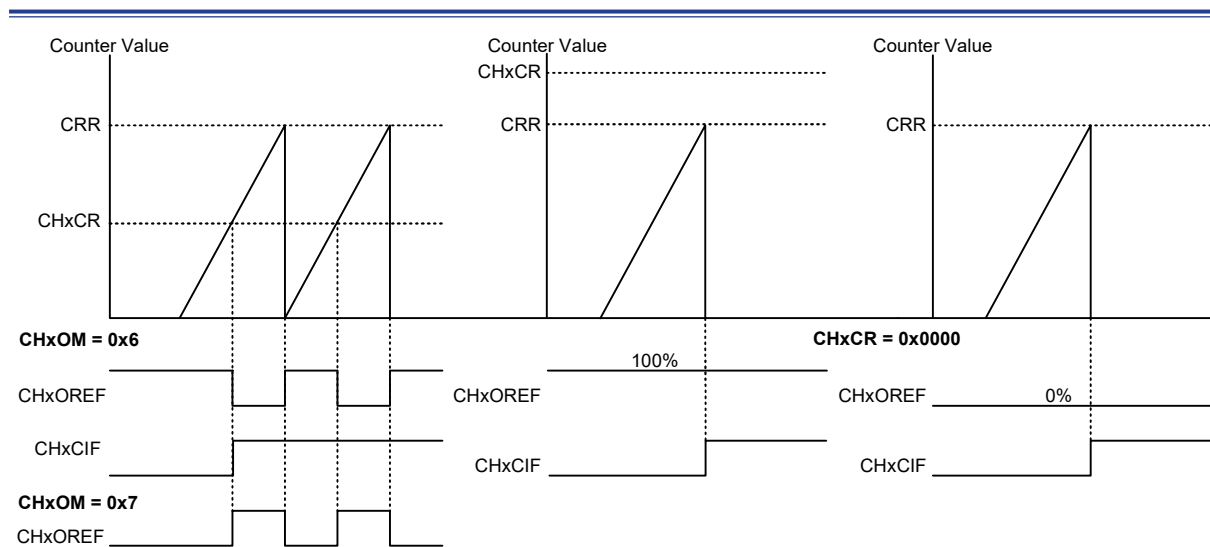
CHxOM Value	Compare Match Level
0x0	No change
0x1	Clear Output to 0
0x2	Set Output to 1
0x3	Toggle Output
0x4	Force Inactive Level
0x5	Force Active Level
0x6	PWM Mode 1
0x7	PWM Mode 2
0xE	Asymmetric PWM mode 1
0xF	Asymmetric PWM mode 2



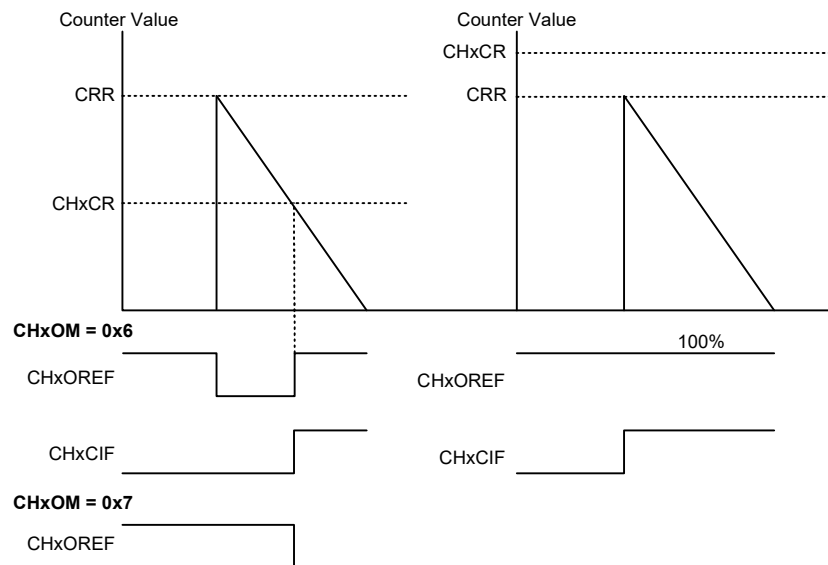
**Figure 82. Toggle Mode Channel Output Reference Signal (CHxPRE = 0)**



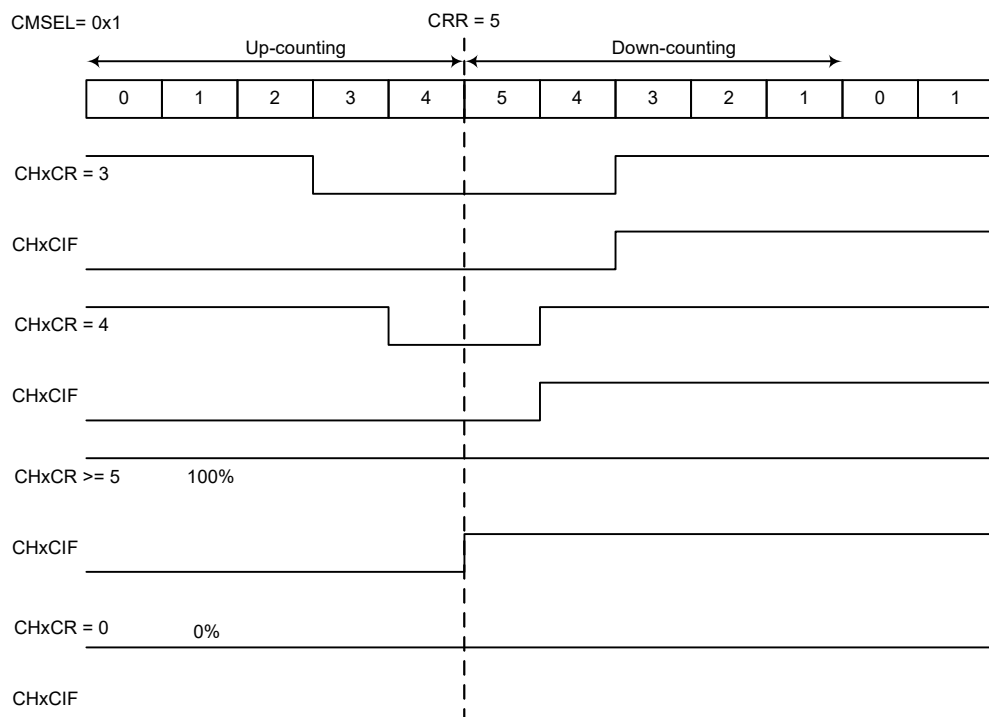
**Figure 83. Toggle Mode Channel Output Reference Signal (CHxPRE = 1)**



**Figure 84. PWM Mode Channel Output Reference Signal and Counter in Up-counting Mode**



**Figure 85. PWM Mode Channel Output Reference Signal and Counter in Down-counting Mode**



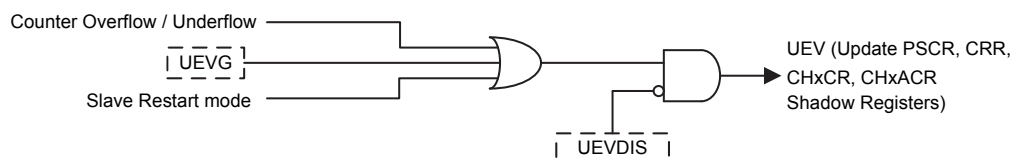
**Figure 86. PWM Mode Channel Output Reference Signal and Counter in Center-aligned Mode**

## Update Management

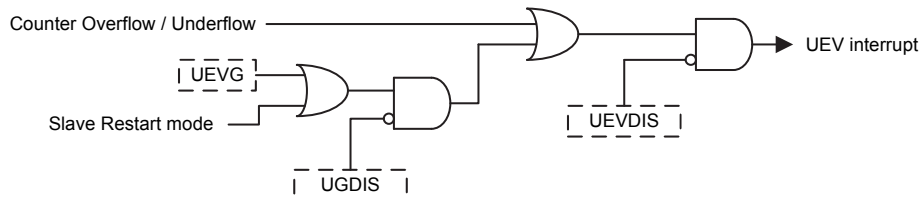
The Update event is used to update the CRR, the PSCR, the CHxACR and the CHxCR values from the actual registers to the corresponding shadow registers. An update event occurs when the counter overflows or underflows, the software update control bit is triggered or an update event from the slave controller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

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

### Update Event Management



### Update Event Interrupt Management



**Figure 87. Update Event Setting Diagram**

## Single Pulse Mode

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

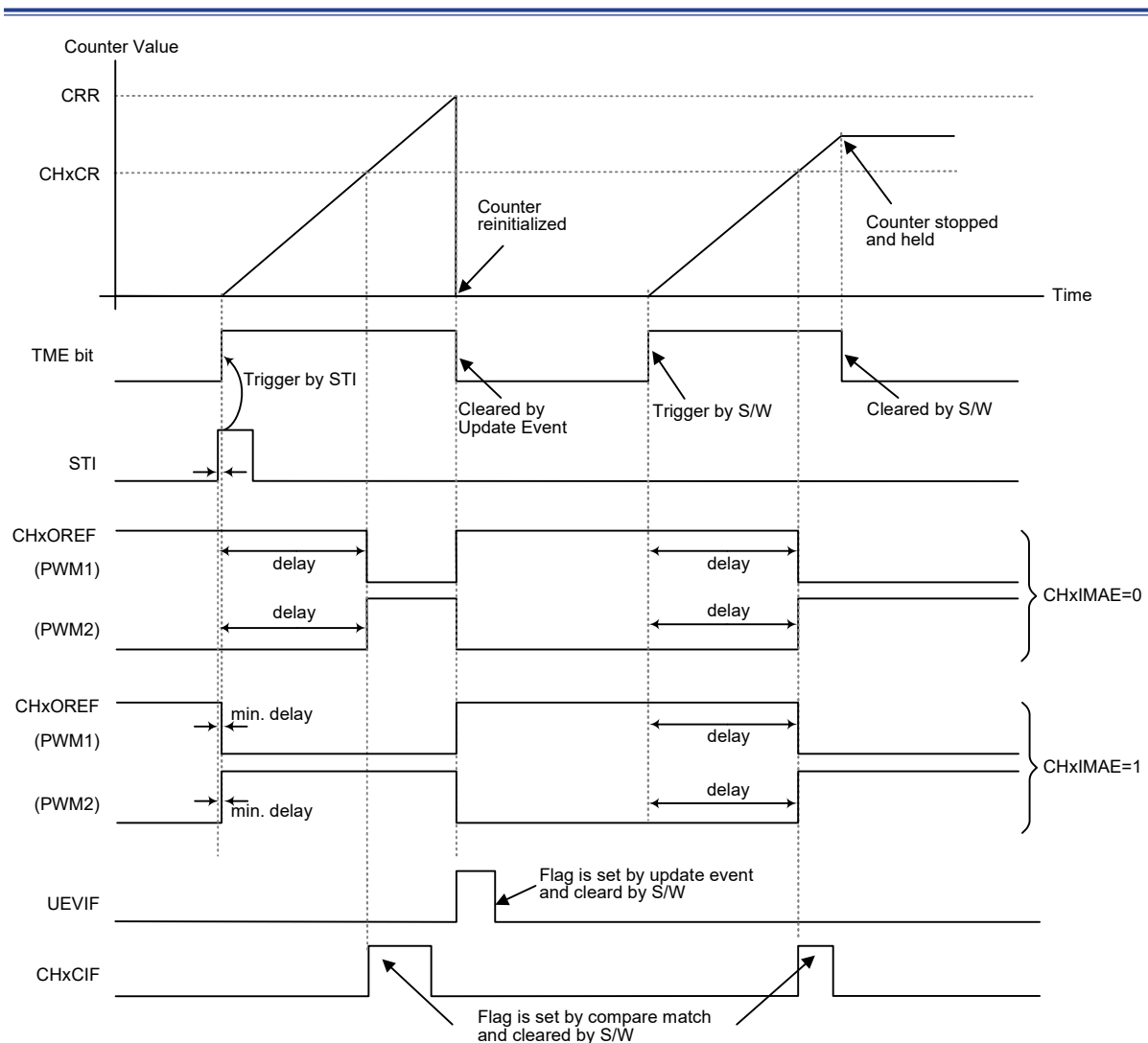
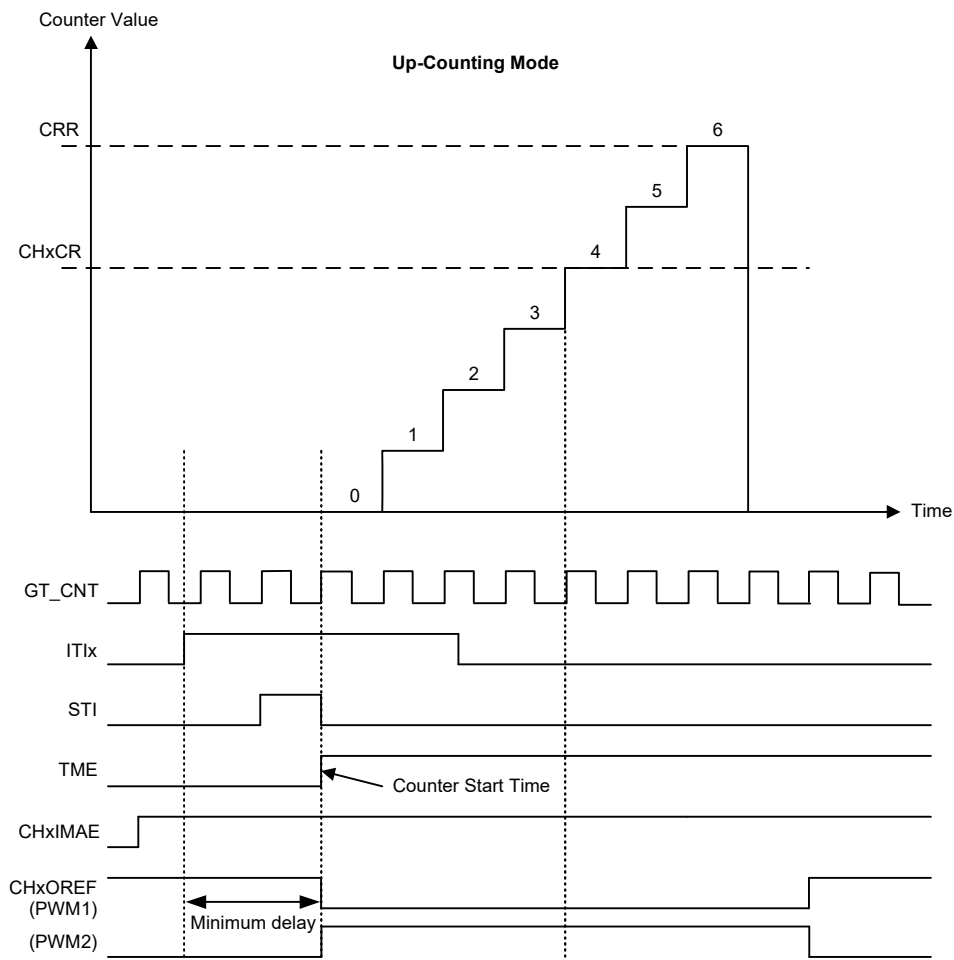


Figure 88. Single Pulse Mode

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

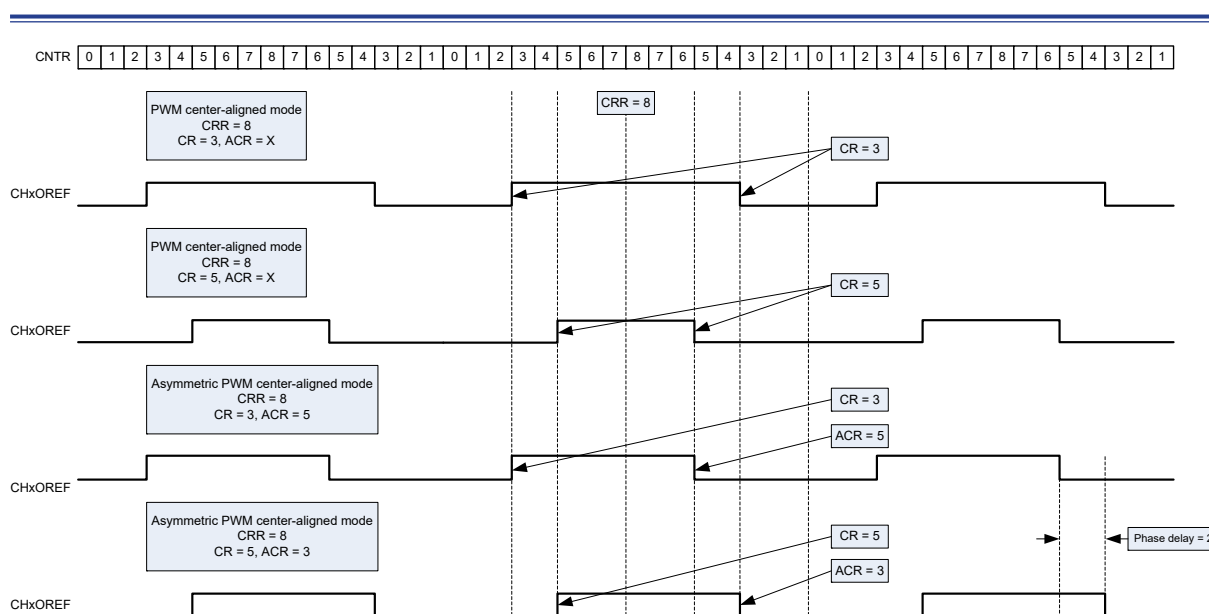


**Figure 89. Immediate Active Mode Minimum Delay**

## Asymmetric PWM Mode

Asymmetric PWM mode allows two center-aligned PWM signals to be generated with a programmable phase shift. While the PWM frequency is determined by the value of the CRR register, the duty cycle and the phase-shift are determined by the CHxCR and CHxACR register. When the counter is counting up, the PWM uses the value in CHxCR as up-count compare value. When the counter is into counting down stage, the PWM uses the value in CHxACR as down-count compare value. The Figure 90 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 90. Asymmetric PWM Mode versus Center-aligned Counting Mode**

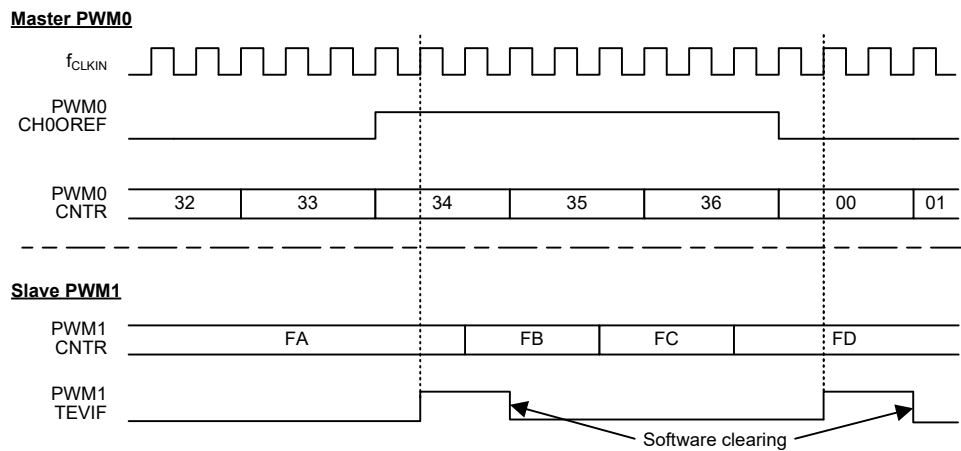
## Timer Interconnection

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

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

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

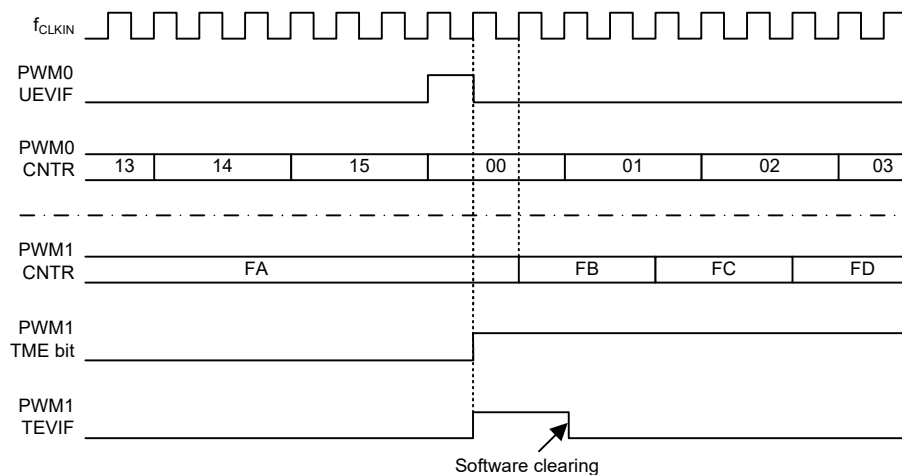




**Figure 91. Pausing PWM1 using the PWM0 CH0OREF Signal**

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

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

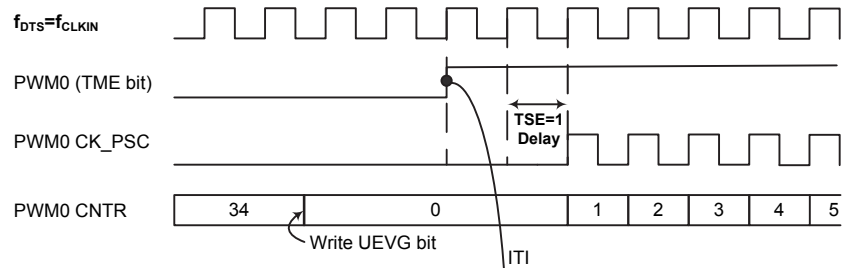


**Figure 92. Triggering PWM1 with PWM0 Update Event**

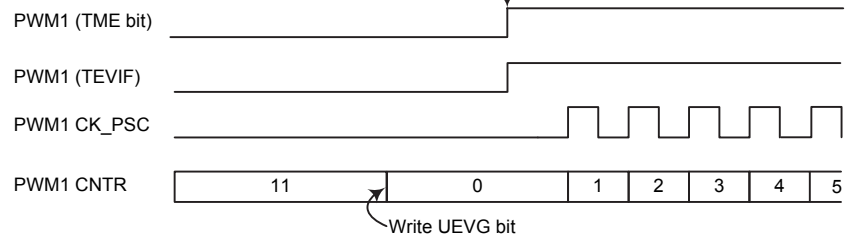
### Starting two timers synchronously with the master enable MTO signal trigger

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

#### Master PWM0



#### Slave PWM1



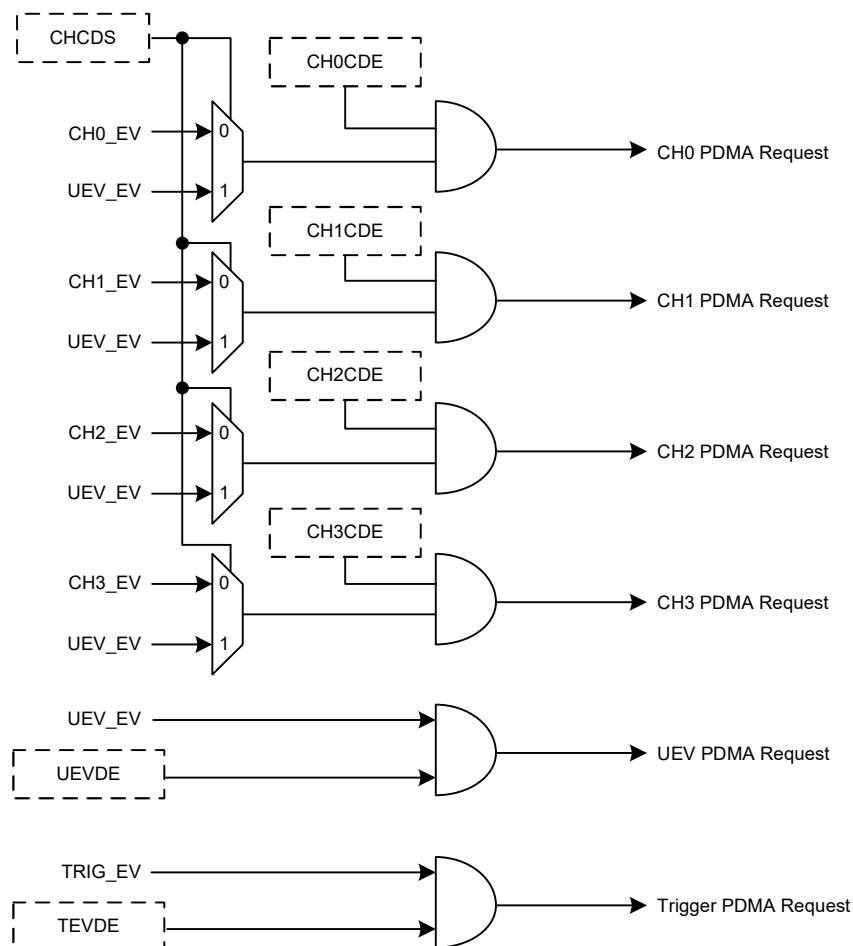
**Figure 93. Trigger PWM0 and PWM1 with the PWM0 Timer Enable Signal**

## Trigger Peripherals Start

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

## PDMA Request

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



**Figure 94. PWM PDMA Mapping Diagram**

## Register Map

The following table shows the PWM registers and reset values.

**Table 36. PWM Register Map**

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

## Register Descriptions

### Timer Counter Configuration Register – CNTCFR

This register specifies the PWM counter configuration.

Offset: 0x000

Reset value: 0x0000\_0000

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

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

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

### Timer Mode Configuration Register – MDCFR

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

Offset: 0x004

Reset value: 0x0000\_0000

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

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

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

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



## Timer Trigger Configuration Register – TRCFR

This register specifies the trigger source selection of PWM.

Offset: 0x008

Reset value: 0x0000\_0000

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

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

**Table 37. PWM Internal Trigger Connection**

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

## Timer Control Register – CTR

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

Offset: 0x010

Reset value: 0x0000\_0000

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

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

## Channel 0 Output Configuration Register – CH0OCFR

This register specifies the channel 0 output mode configuration.

Offset: 0x040

Reset value: 0x0000\_0000

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

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

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

## Channel 1 Output Configuration Register – CH1OCFR

This register specifies the channel 1 output mode configuration.

Offset: 0x044

Reset value: 0x0000\_0000

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

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

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

## Channel 2 Output Configuration Register – CH2OCFR

This register specifies the channel 2 output mode configuration.

Offset: 0x048

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[5]	CH2IMAE	<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 signal will be forced to the compare matched level immediately after an available trigger event occurs irrespective of the result of the comparison between the CNTR and the CH2CR values.</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 PWM mode 2.</p>
[4]	CH2PRE	<p>Channel 2 Compare Register (CH2CR) Preload Enable</p> <p>0: CH2CR preload function is disabled.</p> <p>The CH2CR register can be immediately assigned a new value when the CH2PRE bit is cleared to 0 and the updated CH2CR value is used immediately.</p> <p>1: CH2CR preload function is enabled</p> <p>The new CH2CR value will not be transferred to its shadow register until the update event occurs.</p>

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



## Channel 3 Output Configuration Register – CH3OCFR

This register specifies the channel 3 output mode configuration.

Offset: 0x04C

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[5]	CH3IMAE	<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 signal will be forced to the compare matched level immediately after an available trigger event occurs irrespective of the result of the comparison between the CNTR and the CH3CR values.</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 PWM mode 2.</p>
[4]	CH3PRE	<p>Channel 3 Compare Register (CH3CR) Preload Enable</p> <p>0: CH3CR preload function is disabled.</p> <p>The CH3CR register can be immediately assigned a new value when the CH3PRE bit is cleared to 0 and the updated CH3CR value is used immediately.</p> <p>1: CH3CR preload function is enabled</p> <p>The new CH3CR value will not be transferred to its shadow register until the update event occurs.</p>

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

## Channel Control Register – CHCTR

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

Offset: 0x050

Reset value: 0x0000\_0000

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

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

## Channel Polarity Configuration Register – CHPOLR

This register contains the channel compare output polarity control.

Offset: 0x054

Reset value: 0x0000\_0000

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

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

## Timer PDMA/Interrupt Control Register – DICTR

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

Offset: 0x074

Reset value: 0x0000\_0000

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

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

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

## Timer Event Generator Register – EVGR

This register contains the software event generation bits.

Offset: 0x078

Reset value: 0x0000\_0000

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

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

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

### Timer Interrupt Status Register – INTSR

This register stores the timer interrupt status.

Offset: 0x07C

Reset value: 0x0000\_0000

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

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

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



## Timer Counter Register – CNTR

This register stores the timer counter value.

Offset: 0x080

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[15:0]	CNTV	Counter Value

## Timer Prescaler Register – PSCR

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

Offset: 0x084

Reset value: 0x0000\_0000

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

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

## Timer Counter-Reload Register – CRR

This register specifies the timer counter-reload value.

Offset: 0x088

Reset value: 0x0000\_FFFF

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

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

## Channel 0 Compare Register – CH0CR

This register specifies the timer channel 0 compare value.

Offset: 0x090

Reset value: 0x0000\_0000

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

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

## Channel 1 Compare Register – CH1CR

This register specifies the timer channel 1 compare value.

Offset: 0x094

Reset value: 0x0000\_0000

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

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

## Channel 2 Compare Register – CH2CR

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

Offset: 0x098

Reset value: 0x0000\_0000

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

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

## Channel 3 Compare Register – CH3CR

This register specifies the timer channel 3 compare value.

Offset: 0x09C

Reset value: 0x0000\_0000

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

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

## Channel 0 Asymmetric Compare Register – CH0ACR

This register specifies the timer channel 0 asymmetric compare value.

Offset: 0x0A0

Reset value: 0x0000\_0000

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

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

## Channel 1 Asymmetric Compare Register – CH1ACR

This register specifies the timer channel 1 asymmetric compare value.

Offset: 0x0A4

Reset value: 0x0000\_0000

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

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

## Channel 2 Asymmetric Compare Register – CH2ACR

This register specifies the timer channel 2 asymmetric compare value.

Offset: 0x0A8

Reset value: 0x0000\_0000

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

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

## Channel 3 Asymmetric Compare Register – CH3ACR

This register specifies the timer channel 3 asymmetric compare value.

Offset: 0x0AC

Reset value: 0x0000\_0000

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

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

# 15 Single-Channel Timer (SCTM)

## Introduction

The Single-Channel Timer consists of one 16-bit up-counter, one 16-bit Capture/Compare Register (CCR), one 16-bit Counter-Reload Register (CRR) and several control/status registers. It can be used for a variety of purposes including general timer, input signal pulse width measurement or output waveform generation such as single pulse generation or PWM output.

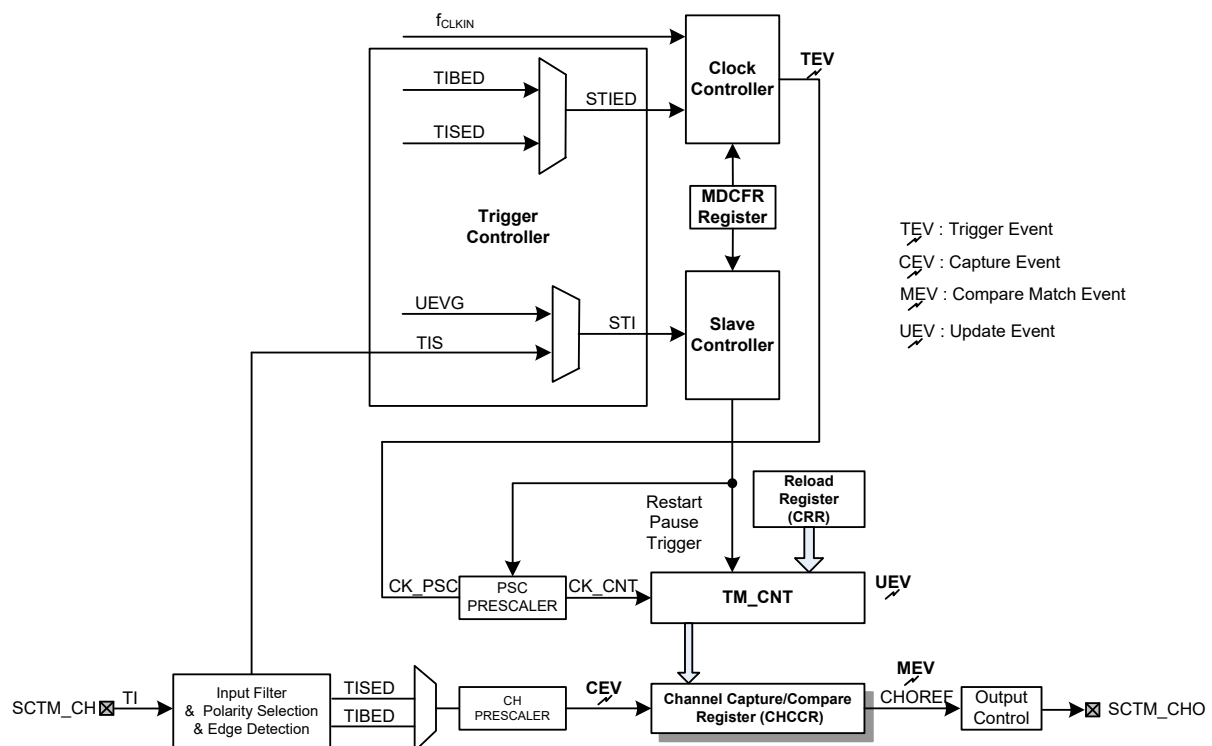


Figure 95. SCTM Block Diagram

## Features

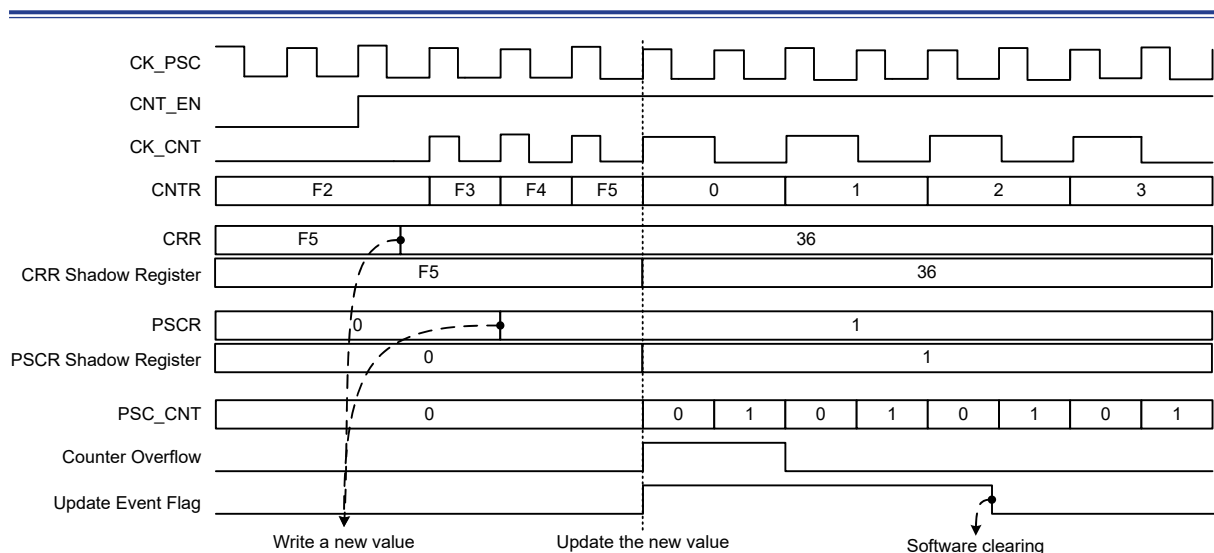
- 16-bit auto-reload up-counter
- 16-bit programmable prescaler that allows division of the counter clock frequency by any factor between 1 and 65536
- Single channel for:
  - Input Capture function
  - Compare Match Output
  - PWM waveform output
- Interrupt generation with the following events:
  -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

## Functional Descriptions

### Counter Mode

#### Up-Counting

The counter counts continuously from 0 to the counter-reload value, which is defined in the CRR register. Once the counter reaches the counter-reload value, then restarts from 0 and generates a counter overflow event. This action will continue repeatedly. When the update event is generated by setting the UEVG bit in the EVGR register to 1, the counter value will also be initialized to 0.



**Figure 96. Up-Counting Example**



## Clock Controller

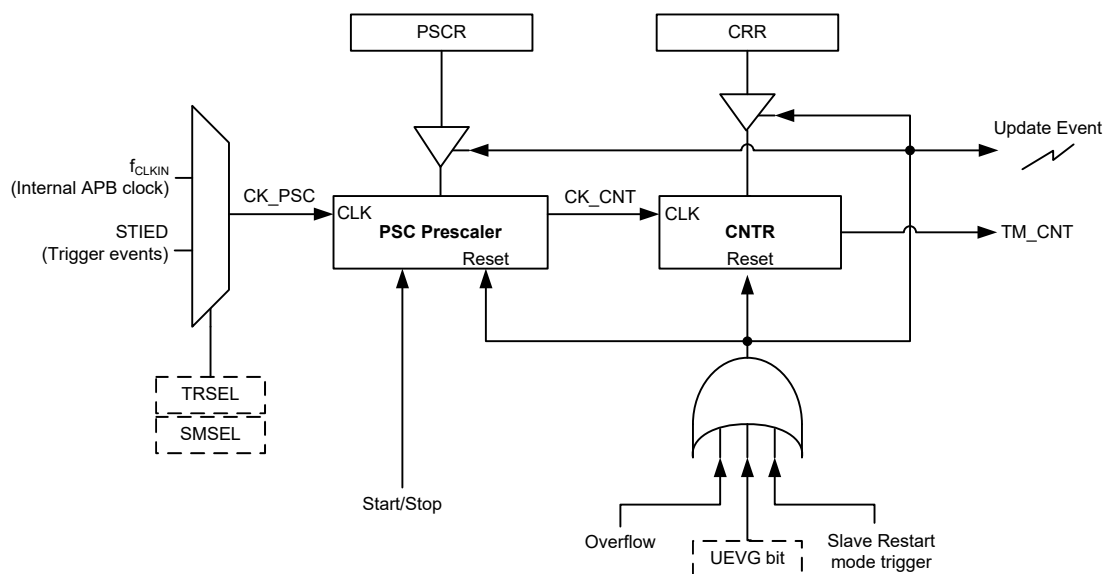
The following describes the Timer Module clock controller which determines the clock source of the internal prescaler counter.

### ■ Internal APB clock $f_{CLKIN}$

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

### ■ STIED

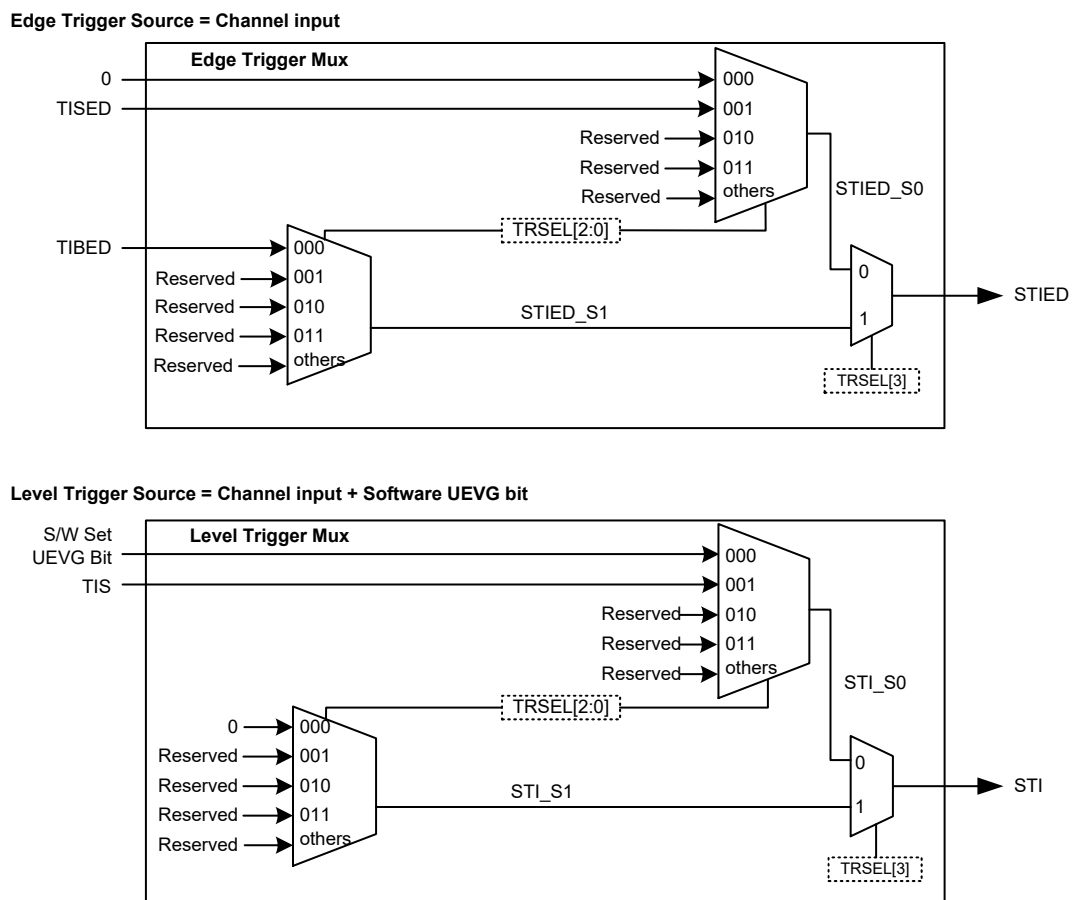
The counter prescaler can count during each rising edge of the STI signal. This mode can be selected by setting the SMSEL field to 0x7 in the MDCFR register. Here the counter will act as an event counter. The input event, known as STI here, can be selected by setting the TRSEL field to an available value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



**Figure 97. SCTM Clock Source Selection**

## Trigger Controller

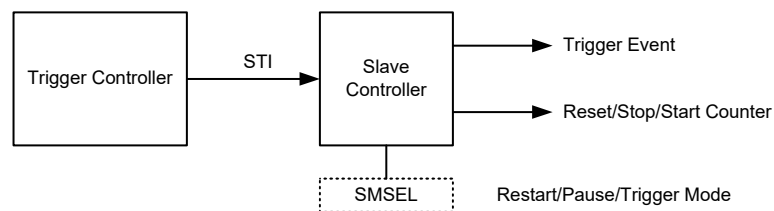
The trigger controller is used to select the trigger source and setup the trigger level or edge trigger condition. For the internal trigger input, it can be selected by the Trigger Selection bits TRSEL in the TRCFR register. For all the trigger sources except the UEVG bit software trigger, the internal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



**Figure 98. Trigger Controller Block**

## Slave Controller

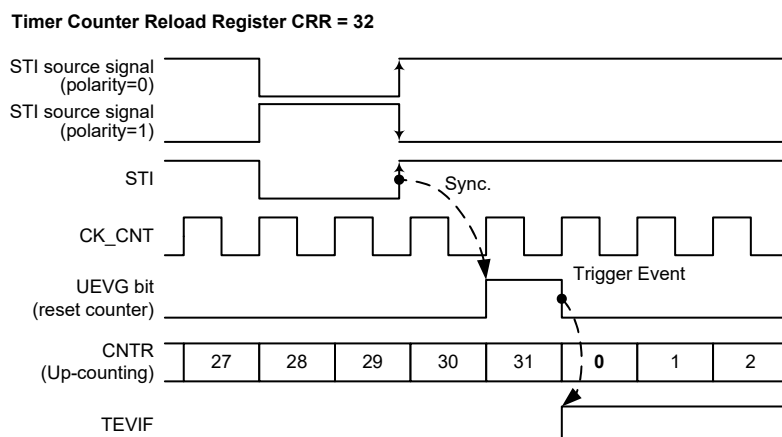
The SCTM can be synchronized with an external trigger in several modes including the Restart mode, the Pause mode and the Trigger mode which is selected by the SMSEL field in the MDCFR register. The trigger input of these modes comes from the STI signal which is selected by the TRSEL field in the TRCFR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



**Figure 99. Slave Controller Diagram**

### Restart Mode

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



**Figure 100. SCTM in Restart Mode**

### Pause Mode

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

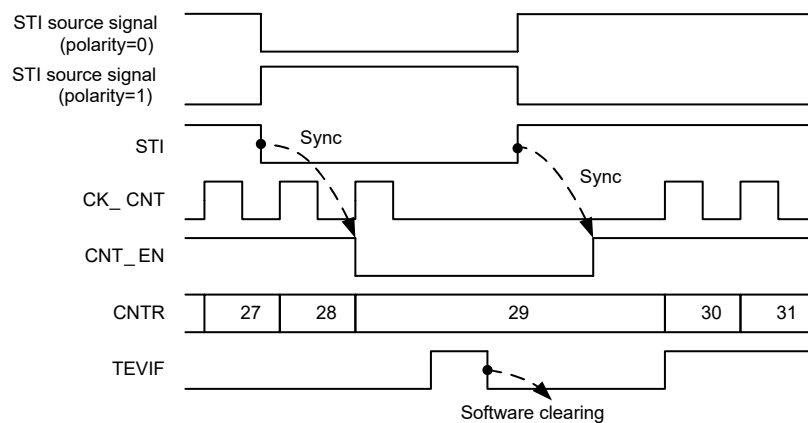


Figure 101. SCTM in Pause Mode

### 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 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.

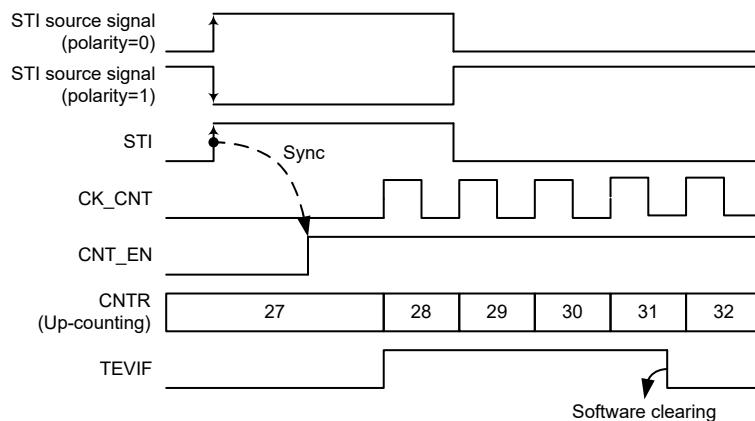


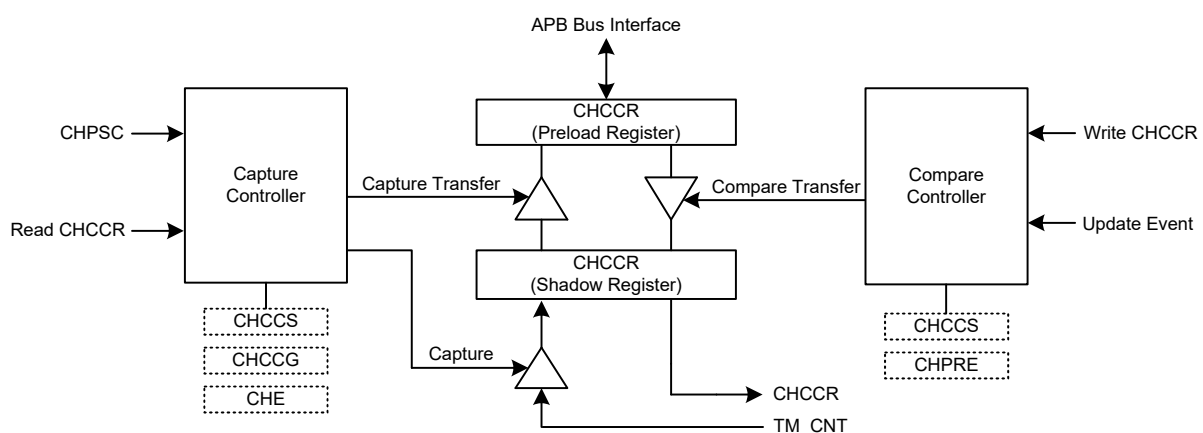
Figure 102. SCTM in Trigger Mode

## Channel Controller

The SCTM channel can be used as the capture input or compare match output. The capture input or compare match output channel is composed of a preload register and a shadow register. Data access of the APB bus is always implemented by reading/writing the preload register.

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

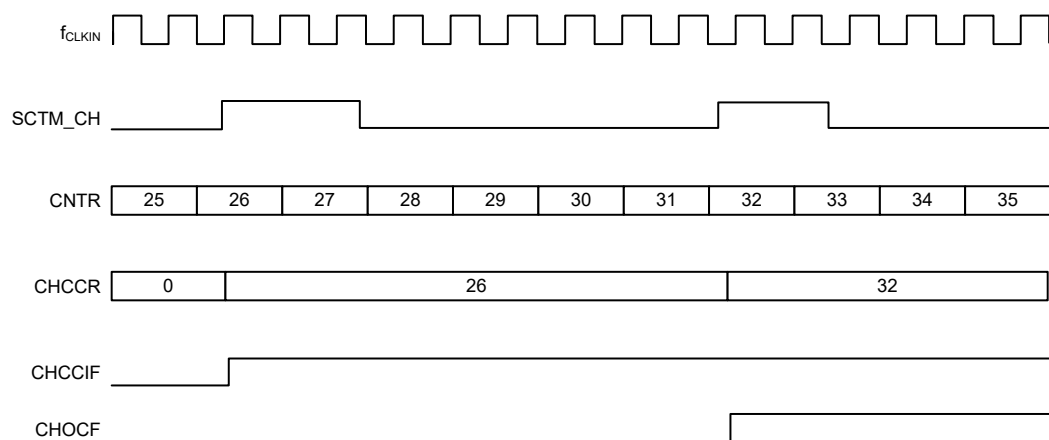
When used in the compare match output mode, the contents of the CHCCR preload register is copied into the associated shadow register, the counter value is then compared with the register value.



### Figure 103. Capture/Compare Block Diagram

### Capture Counter Value Transferred to CHCCR

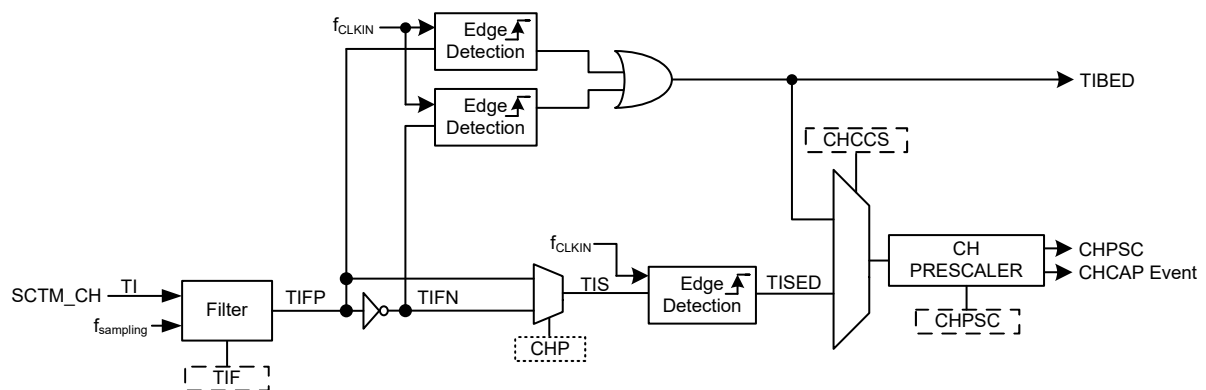
When the channel is used as a capture input, the counter value is captured into the Channel Capture/Compare Register (CHCCR) when an effective input signal transition occurs. Once the capture event occurs, the CHCCIF flag in the INTSR register is set accordingly. If the CHCCIF bit is already set, i.e.,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



### Figure 104. Input Capture Mode

## Input Stage

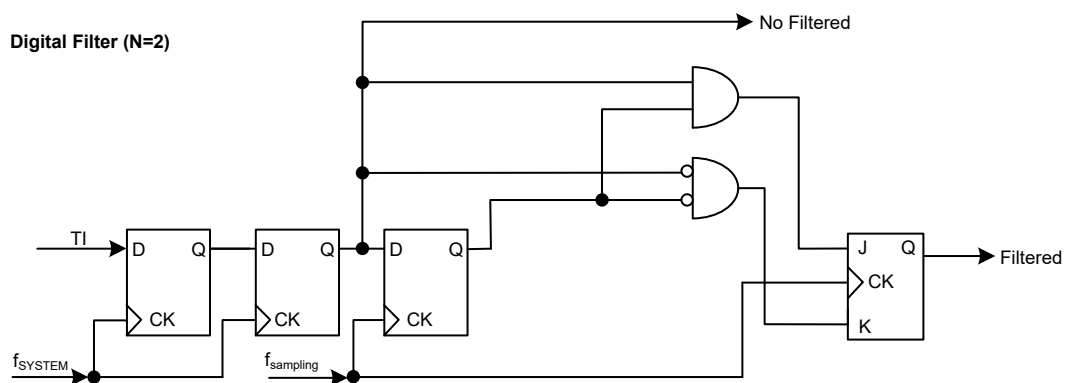
The input stage consists of a digital filter, a channel polarity selection, edge detection and a channel prescaler. The channel input signal (TI) is sampled by a digital filter to generate a filtered input signal TIFP. Then the channel polarity and the edge detection block can generate a TISED signal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



### Figure 105. Channel Input Stages

## Digital Filter

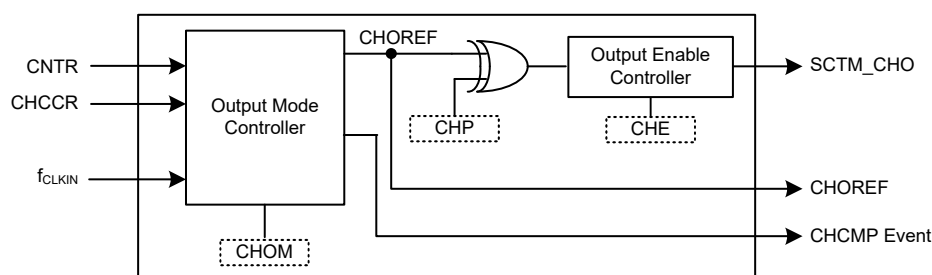
The digital filter is embedded in the channel input stage. The digital filter in the SCTM is an N-event counter where N refers to how many valid transitions are necessary to output a filtered signal. The N value can be 0, 2, 4, 5, 6 or 8 according to the user selection for this digital filter.



**Figure 106. TI Digital Filter Diagram with N = 2**

## Output Stage

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



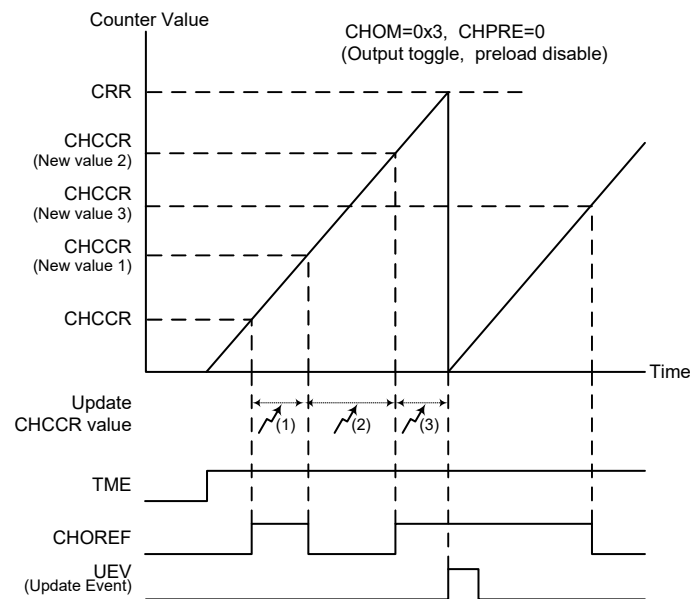
**Figure 107. Output Stage Block Diagram**

### Channel Output Reference Signal

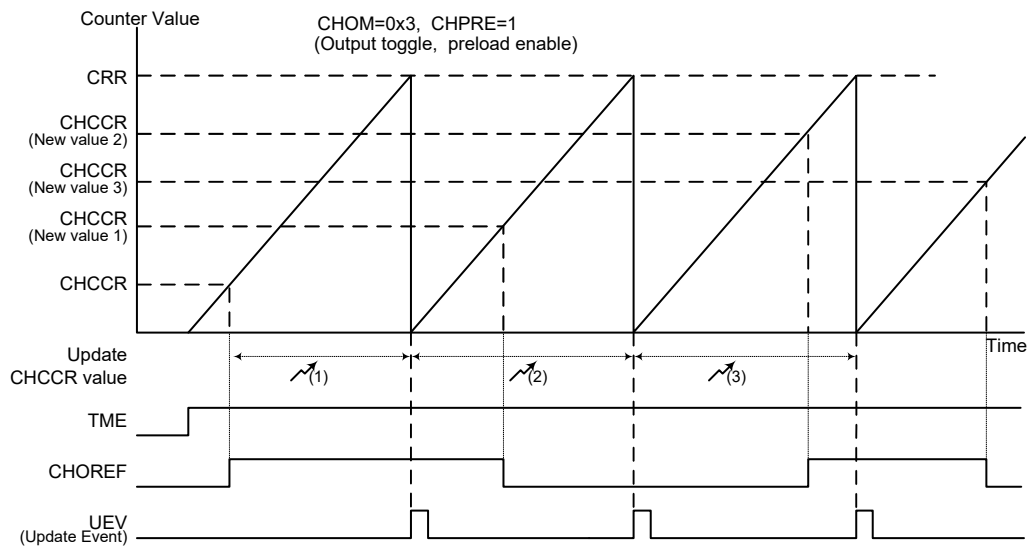
When the SCTM is used in the compare match output mode, the CHOREF signal (Channel Output Reference signal) is defined by the CHOM bit setup. The CHOREF signal has several types of output function which defines what happens to the output when the counter value matches the contents of the CHCCR register. In addition to the low, high and toggle CHOREF output types; there are also PWM mode 1 and PWM mode 2 outputs. In these modes, the CHOREF signal level is changed according to the relationship between the counter value and the CHCCR content. There are also two modes which will force the output into an inactive or active state irrespective of the CHCCR content or counter values. With regard to a more detailed description refer to the relative bit definition. The Table 44 shows a summary of the output type setup.

**Table 38. Compare Match Output Setup**

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

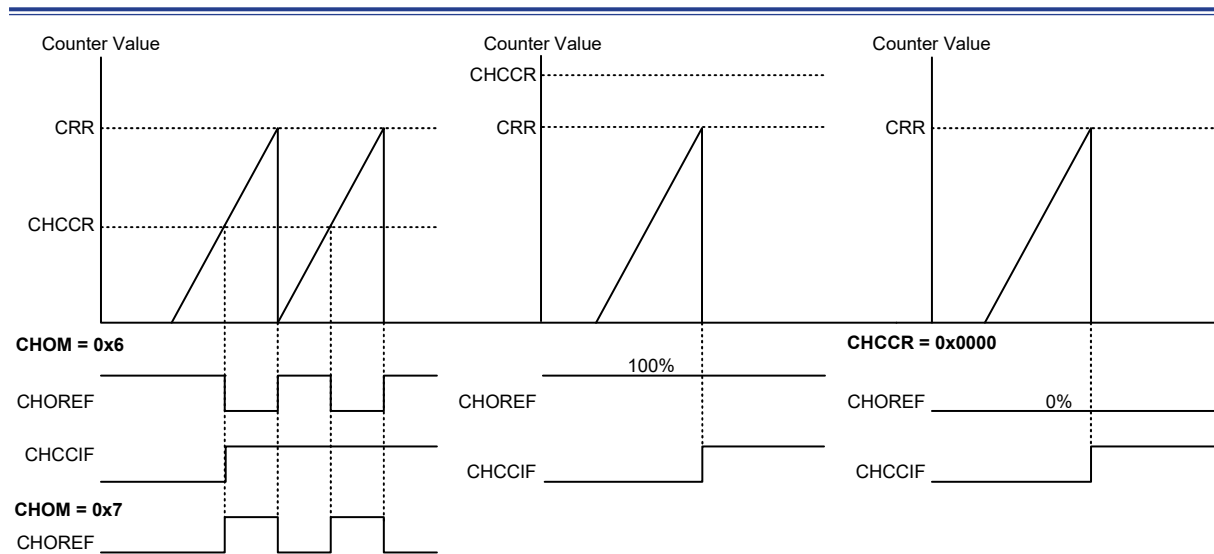


**Figure 108. Toggle Mode Channel Output Reference Signal (CHPRE = 0)**



**Figure 109. Toggle Mode Channel Output Reference Signal (CHPRE = 1)**





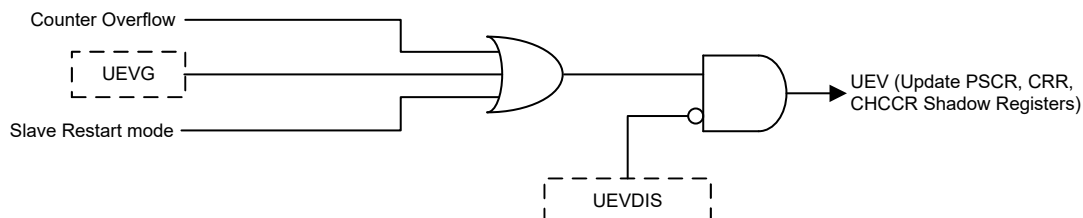
**Figure 110. PWM Mode Channel Output Reference Signal**

## Update Management

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

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

#### Update Event Management



#### Update Event Interrupt Management

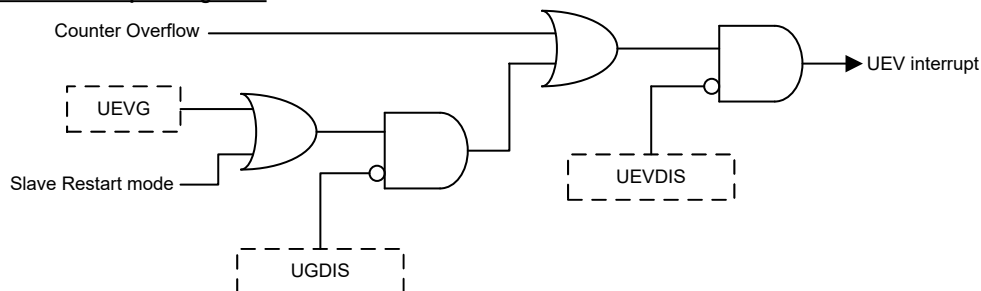


Figure 111. Update Event Setting Diagram

## Register Map

The following table shows the SCTM registers and reset values.

Table 39. SCTM Register Map

Register	Offset	Description	Reset Value
CNTCFR	0x000	Timer Counter Configuration Register	0x0000_0000
MDCFR	0x004	Timer Mode Configuration Register	0x0000_0000
TRCFR	0x008	Timer Trigger Configuration Register	0x0000_0000
CTR	0x010	Timer Control Register	0x0000_0000
CHICFR	0x020	Channel Input Configuration Register	0x0000_0000
CHOCFR	0x040	Channel Output Configuration Register	0x0000_0000
CHCTR	0x050	Channel Control Register	0x0000_0000
CHPOLR	0x054	Channel Polarity Configuration Register	0x0000_0000
DICTR	0x074	Timer Interrupt Control Register	0x0000_0000
EVGR	0x078	Timer Event Generator Register	0x0000_0000
INTSR	0x07C	Timer Interrupt Status Register	0x0000_0000
CNTR	0x080	Timer Counter Register	0x0000_0000
PSCR	0x084	Timer Prescaler Register	0x0000_0000
CRR	0x088	Timer Counter Reload Register	0x0000_FFFF
CHCCR	0x090	Channel Capture/Compare Register	0x0000_0000



## Timer Mode Configuration Register – MDCFR

This register specifies the SCTM slave mode selection.

Offset: 0x004

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions																					
[10:8]	SMSEL	Slave Mode Selection																					
		<table> <tr> <th>SMSEL [2:0]</th><th>Mode</th><th>Descriptions</th></tr> <tr> <td>000</td><td>Disable mode</td><td>The prescaler is clocked directly by the internal clock.</td></tr> <tr> <td>100</td><td>Restart Mode</td><td>The counter value restarts from 0 on the rising edge of the STI signal. The registers will also be updated.</td></tr> <tr> <td>101</td><td>Pause Mode</td><td>The counter starts to count when the selected trigger input STI is high. The counter stops counting on the instant, not being reset, when the STI signal changes its state to a low level. Both the counter start and stop control are determined by the STI signal.</td></tr> <tr> <td>110</td><td>Trigger Mode</td><td>The counter starts to count from the original value in the counter on the rising edge of the selected trigger input STI. Only the counter start control is determined by the STI signal.</td></tr> <tr> <td>111</td><td>STIED</td><td>The rising edge of the selected trigger signal STI will clock the counter.</td></tr> <tr> <td>Others</td><td>Reserved</td><td></td></tr> </table>	SMSEL [2:0]	Mode	Descriptions	000	Disable mode	The prescaler is clocked directly by the internal clock.	100	Restart Mode	The counter value restarts from 0 on the rising edge of the STI signal. The registers will also be updated.	101	Pause Mode	The counter starts to count when the selected trigger input STI is high. The counter stops counting on the instant, not being reset, when the STI signal changes its state to a low level. Both the counter start and stop control are determined by the STI signal.	110	Trigger Mode	The counter starts to count from the original value in the counter on the rising edge of the selected trigger input STI. Only the counter start control is determined by the STI signal.	111	STIED	The rising edge of the selected trigger signal STI will clock the counter.	Others	Reserved	
SMSEL [2:0]	Mode	Descriptions																					
000	Disable mode	The prescaler is clocked directly by the internal clock.																					
100	Restart Mode	The counter value restarts from 0 on the rising edge of the STI signal. The registers will also be updated.																					
101	Pause Mode	The counter starts to count when the selected trigger input STI is high. The counter stops counting on the instant, not being reset, when the STI signal changes its state to a low level. Both the counter start and stop control are determined by the STI signal.																					
110	Trigger Mode	The counter starts to count from the original value in the counter on the rising edge of the selected trigger input STI. Only the counter start control is determined by the STI signal.																					
111	STIED	The rising edge of the selected trigger signal STI will clock the counter.																					
Others	Reserved																						

## Timer Trigger Configuration Register – TRCFR

This register specifies the trigger source selection of SCTM.

Offset: 0x008

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[3:0]	TRSEL	<p>Trigger Source Selection</p> <p>These bits are used to select the trigger input (STI) for counter synchronization.</p> <p>0000: Software Trigger by setting the UEVG bit</p> <p>0001: Filtered input of the channel (TIS)</p> <p>1000: Channel both edge detector (TIBED)</p> <p>Others: Reserved</p> <p>Note: These bits must be updated only when they are not in use, i.e. the slave mode is disabled by setting the SMSEL field to 0x0.</p>

## Timer Control Register – CTR

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

Offset: 0x010

Reset value: 0x0000\_0000

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

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

## Channel Input Configuration Register – CHICFR

This register specifies the channel input mode configuration.

Offset: 0x020

Reset value: 0x0000\_0000

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

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

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



## Channel Output Configuration Register – CHOCFR

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

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

## Channel Control Register – CHCTR

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

Offset: 0x050

Reset value: 0x0000\_0000

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

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

## Channel Polarity Configuration Register – CHPOLR

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

Offset: 0x054

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
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							CHP
								RW 0

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

## Timer Interrupt Control Register – DICTR

This register contains the timer interrupt enable control bits.

Offset: 0x074

Reset value: 0x0000\_0000

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

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

## Timer Event Generator Register – EVGR

This register contains the software event generation bits.

Offset: 0x078

Reset value: 0x0000\_0000

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

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

## Timer Interrupt Status Register – INTSR

This register stores the timer interrupt status.

Offset: 0x07C

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[10]	TEVIF	Trigger Event Interrupt Flag This flag is set by hardware on a trigger event and is cleared by software. 0: No trigger event occurs 1: Trigger event occurs
[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 - The UEVG bit is asserted - A restart trigger event occurs from the slave trigger input
[4]	CHOCF	Channel Over-Capture Flag This flag is set by hardware and cleared by software. 0: No over-capture event is detected 1: Capture event occurs again when the CHCCIF bit is already set and it is not yet cleared by software.
[0]	CHCCIF	Channel Capture/Compare Interrupt Flag - Channel is configured as an output: 0: No match event occurs 1: The contents of the counter CNTR have matched the content of the CHCCR register This flag is set by hardware when the counter value matches the CHCCR value. It is cleared by software. - Channel is configured as an input: 0: No input capture occurs 1: Input capture occurs This bit is set by hardware on a capture event. It is cleared by software or by reading the CHCCR register.

## Timer Counter Register – CNTR

This register stores the timer counter value.

Offset: 0x080

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[15:0]	CNTV	Counter Value

## Timer Prescaler Register – PSCR

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

Offset: 0x084

Reset value: 0x0000\_0000

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

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

## Timer Counter Reload Register – CRR

This register specifies the timer counter reload value.

Offset: 0x088

Reset value: 0x0000\_FFFF

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

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



## Channel Capture/Compare Register – CHCCR

This register specifies the timer channel capture/compare value.

Offset: 0x090

Reset value: 0x0000\_0000

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

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

# 16 Basic Function Timer (BFTM)

## Introduction

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

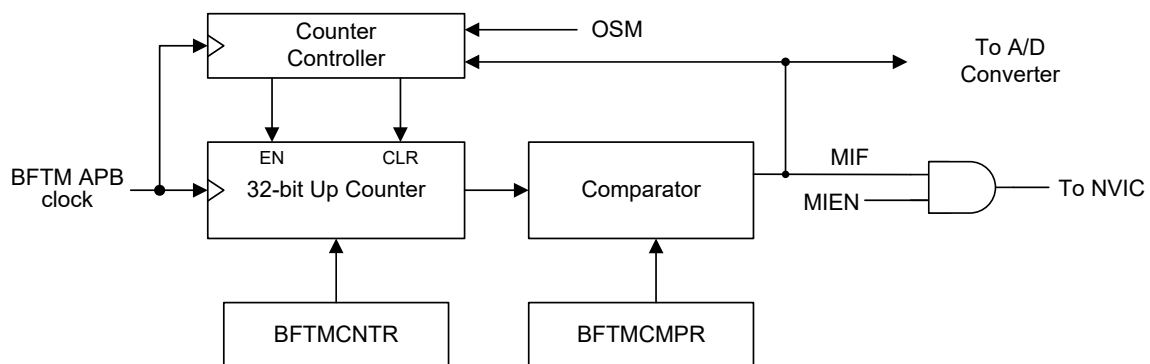


Figure 112. BFTM Block Diagram

## Features

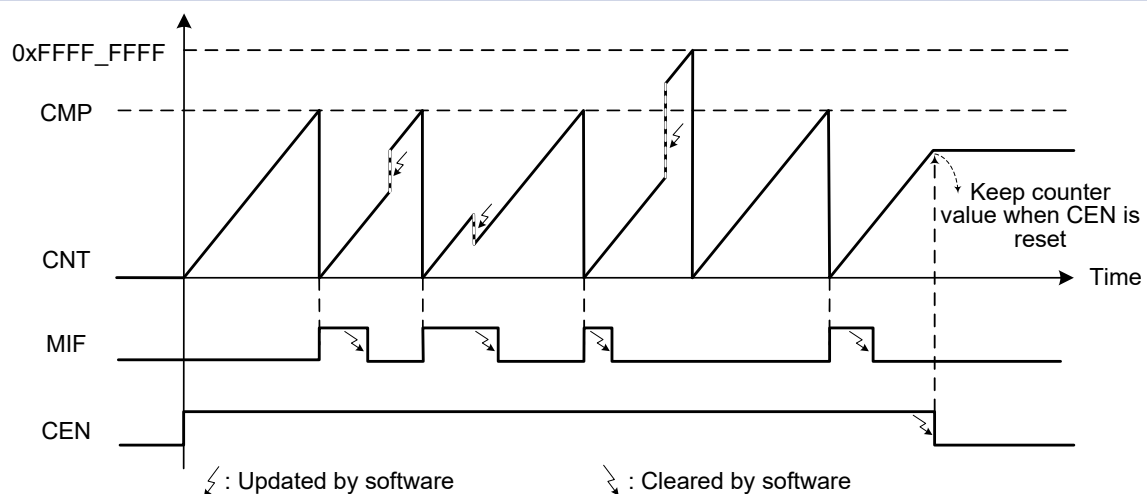
- 32-bit up-counting counter
- Compare Match function
- Includes debug mode
- Clock source: BFTM APB clock
- Counter value can be Read / Written on the fly
- One shot mode: counter stops counting when compare match occurs
- Repetitive mode: counter restarts when compare match occurs
- Compare Match interrupt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

## Functional Description

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

### Repetitive Mode

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



**Figure 113. 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.

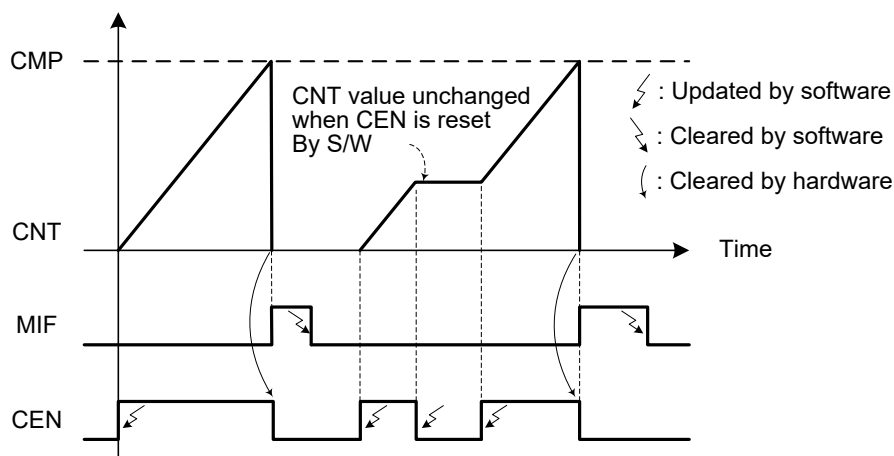


Figure 114. BFTM – One Shot Mode

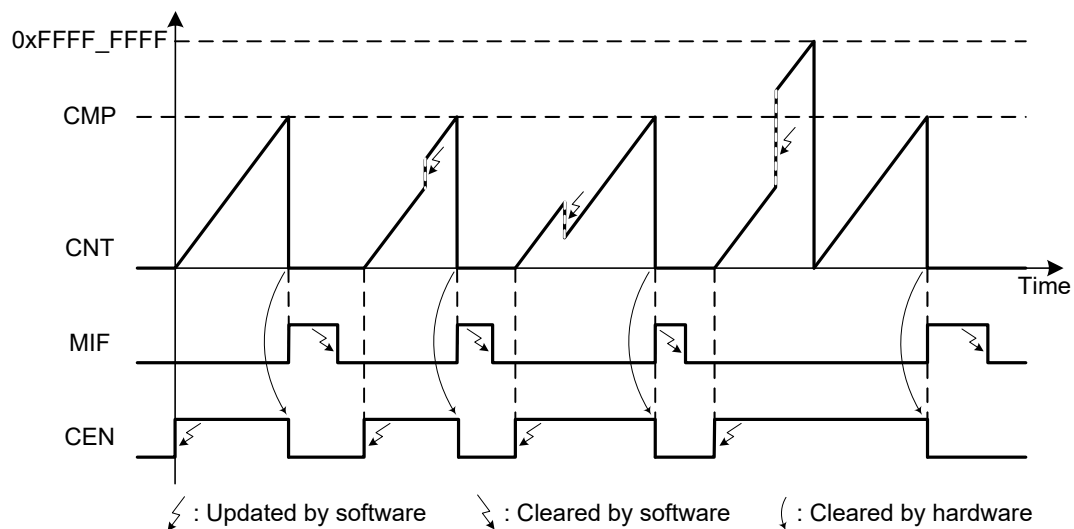


Figure 115. BFTM – One Shot Mode Counter Updating

## 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 40. BFTM Register Map**

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

## Register Descriptions

### BFTM Control Register – BFTMCR

This register specifies the overall BFTM control bits.

Offset: 0x000

Reset value: 0x0000\_0000

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

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

## BFTM Status Register – BFTMSR

This register specifies the BFTM status.

Offset: 0x004

Reset value: 0x0000\_0000

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

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

## BFTM Counter Register – BFTMCNTR

This register specifies the BFTMn counter value.

Offset: 0x008

Reset value: 0x0000\_0000

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

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

## BFTM Compare Value Register – BFTMCMPR

The register specifies the BFTM compare value.

Offset: 0x00C

Reset value: 0xFFFF\_FFFF

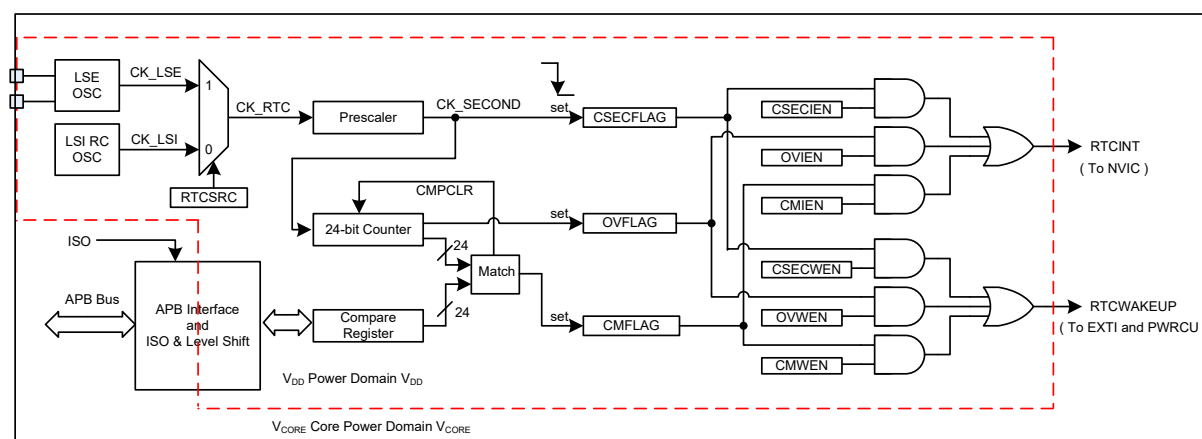
	31	30	29	28	27	26	25	24	
	CMP								
Type/Reset	RW	1	RW	1	RW	1	RW	1	RW
	23	22	21	20	19	18	17	16	
	CMP								
Type/Reset	RW	1	RW	1	RW	1	RW	1	RW
	15	14	13	12	11	10	9	8	
	CMP								
Type/Reset	RW	1	RW	1	RW	1	RW	1	RW
	7	6	5	4	3	2	1	0	
	CMP								
Type/Reset	RW	1	RW	1	RW	1	RW	1	RW

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

## 17 Real Time Clock (RTC)

## Introduction

The Real Time Clock, RTC, circuitry includes the APB interface, a 24-bit up-counter, a control register, a prescaler, a compare register and a status register. Most of the RTC circuits are located in the  $V_{DD}$  Power Domain, as shown in dotted red box in the accompanying figure, except for the APB interface. The APB interface is located in the  $V_{CORE}$  domain. Therefore, it is necessary to be isolated by the ISO signal that comes from the power control unit when the  $V_{CORE}$  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 saving modes. The detailed RTC function will be described in the following sections.



### Figure 116. RTC Block Diagram

## Features

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



## Functional Descriptions

### RTC Related Register Reset

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

### Reading RTC Register

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

### Low Speed Clock Configuration

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

**Table 41. LSE Startup Mode Operating Current and Startup Time**

Startup Mode	LSESM Setting in the RTCCR Register	Operating Current	Startup Time
Normal startup	0	2.0 $\mu$ A	Above 500 ms
Fast startup	1	3.5 $\mu$ A	Below 300 ms

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

## RTC Counter Operation

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





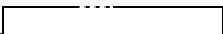





## Interrupt and Wakeup Control

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

## RTCOUNT Output Pin Configuration

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

**Table 42. 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)	-----  -----
		RTCOUT (ROAP = 1)	-----  -----
		ROLF	-----
	1 (Second clock)	RTCCMP	-----X-----
		RTCCNT	-----3-----4-----5-----
		RTCOUT (ROAP = 0)	 -----
		RTCOUT (ROAP = 1)	-----  -----
		ROLF	-----
1 (Level mode)	0 (Compare match)	RTCCMP	-----4-----
		RTCCNT	-----3-----4-----5-----
		RTCOUT (ROAP = 0)	-----  -----
		RTCOUT (ROAP = 1)	-----  -----
		ROLF	-----  -----
	1 (Second clock)	RTCCMP	-----X-----
		RTCCNT	-----3-----4-----5-----
		RTCOUT (ROAP = 0)	 -----
		RTCOUT (ROAP = 1)	-----  -----
		ROLF	-----  -----

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

## Register Map

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

**Table 43. RTC Register Map**

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

## Register Descriptions

### RTC Counter Register – RTCCNT

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

Address: 0x000

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

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

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

## RTC Compare Register – RTCCMP

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

Address: 0x004

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

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

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

This register specifies a range of RTC circuitry control bits.

Reset value: 0x0000 0F04 (Reset by V<sub>DD</sub> Power Domain reset only)

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

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

Bits	Field	Descriptions
[11:8]	RPRE	RTC Clock Prescaler Select $CK\_SECOND = CK\_RTC / 2^{RPRE}$ 0000: $CK\_SECOND = CK\_RTC / 2^0$ 0001: $CK\_SECOND = CK\_RTC / 2^1$ 0010: $CK\_SECOND = CK\_RTC / 2^2$ ... 1111: $CK\_SECOND = CK\_RTC / 2^{15}$
[5]	LSESM	LSE oscillator Startup Mode 0: Normal startup and requires less operating power 1: Fast startup but requires higher operating current
[4]	CMPCLR	Compare Match Counter Clear 0: 24-bit RTC counter is not affected when compare match condition occurs 1: 24-bit RTC counter is cleared when compare match condition occurs
[3]	LSEEN	LSE oscillator Enable Control 0: LSE oscillator is disabled 1: LSE oscillator is enabled
[1]	RTCSRC	RTC Clock Source Selection 0: LSI oscillator is selected as the RTC clock source 1: LSE oscillator is selected as the RTC clock source
[0]	RTCEN	RTC Enable Control 0: RTC is disabled 1: RTC is enabled

## RTC Status Register – RTCSR

This register stores the counter flags.

Address: 0x00C

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

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

Bits	Field	Descriptions
[2]	OVFLAG	Counter Overflow Flag 0: Counter overflow has not occurred since the last RTCSR register read operation 1: Counter overflow has occurred since the last RTCSR register read operation This bit is set by hardware when the counter value in the RTCCNT register changes from 0x00FF_FFFF to 0x0000_0000 and cleared by read operation. This bit is suggested to be read in the RTC IRQ handler and should be taken care when software polling is used.
[1]	CMFLAG	Compare Match Condition Flag 0: Compare match condition has not occurred since the last RTCSR register read operation 1: Compare match condition has occurred since the last RTCSR register read operation. This bit is set by hardware on the CK_SECOND clock falling edge when the RTCCNT register value is equal to the RTCCMP register content. It is cleared by software reading this bit. This bit is suggested for access in the corresponding RTC interrupt routine – do not use software polling during software free running.
[0]	CSECFLAG	CK_SECOND Occurrence Flag 0: CK_SECOND has not occurred since the last RTCSR register read operation 1: CK_SECOND has occurred since the last RTCSR register read operation This bit is set by hardware on the CK_SECOND clock falling edge. It is cleared by software reading this bit. This bit is suggested for access in the corresponding RTC interrupt routine – do not use software polling during software free running.



## RTC Interrupt and Wakeup Enable Register – RTCIWEN

This register contains the interrupt and wakeup enable bits.

Address: 0x010

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

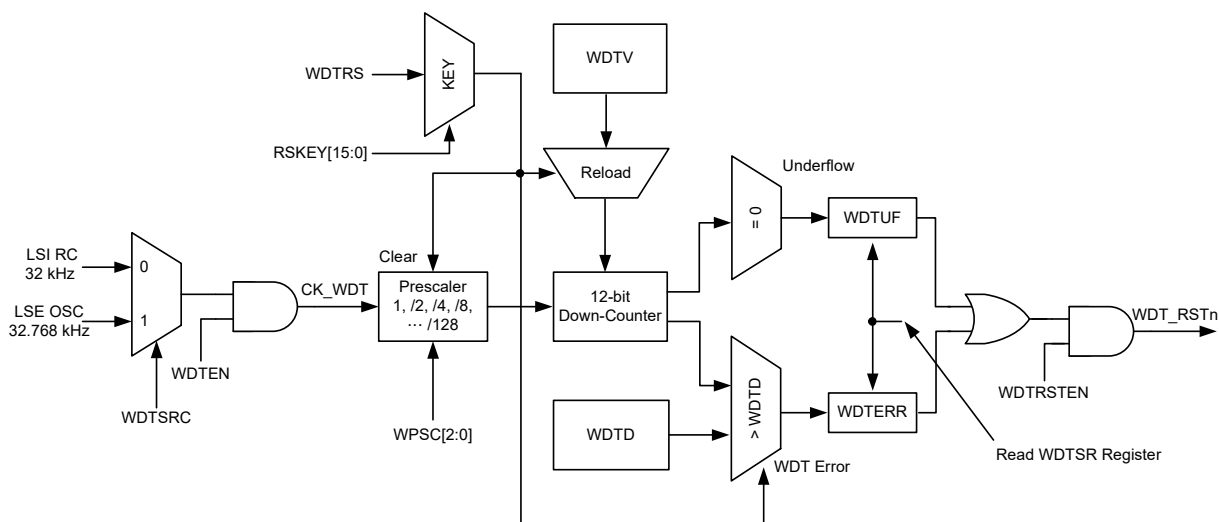
	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved					OVWEN	CMWEN	CSECWEN
	7	6	5	4	3	2	1	0
Type/Reset	Reserved					RW 0	RW 0	RW 0
						OVLEN	CMLEN	CSECLN
						RW 0	RW 0	RW 0

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

# 18 Watchdog Timer (WDT)

## Introduction

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



**Figure 117. Watchdog Timer Block Diagram**

## Features

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

## Functional Description

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

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

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

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

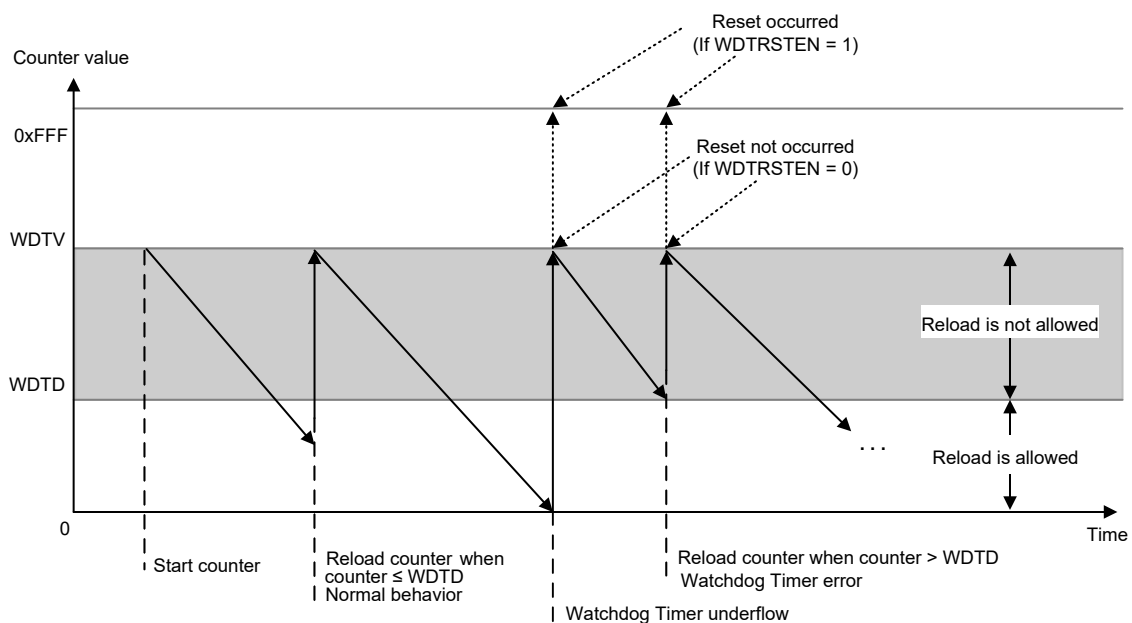
The WDTERR and WDTUF flags in the WDTSR register will be set respectively when the Watchdog Timer error occurs or when a Watchdog Timer underflows. A system reset or writing “1” operation on the WDTSR register will clear the WDTERR and WDTUF flags.

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

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

The Watchdog timer should be used in the following manners:

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



**Figure 118. Watchdog Timer Behavior**

## Register Map

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

**Table 44. Watchdog Timer Register Map**

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

## Register Descriptions

### Watchdog Timer Control Register – WDTCR

This register is used to reload the Watchdog timer.

Offset: 0x000

Reset value: 0x0000\_0000

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

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

## Watchdog Timer Mode Register 0 – WDTMR0

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

Offset: 0x004

Reset value: 0x0000\_0FFF

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

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

## Watchdog Timer Mode Register 1 – WDTMR1

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

Offset: 0x008

Reset value: 0x0000\_7FFF

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

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

## Watchdog Timer Status Register – WDTSR

This register specifies the Watchdog timer status.

Offset: 0x00C

Reset value: 0x0000\_0000

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

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



## Watchdog Timer Protection Register – WDTPR

This register specifies the Watchdog timer protect key configuration.

Offset: 0x010

Reset value: 0x0000\_0000

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

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

## Watchdog Timer Clock Selection Register – WDTCSR

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

Offset: 0x018

Reset value: 0x0000\_0000

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

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

# 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) 1 MHz 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.

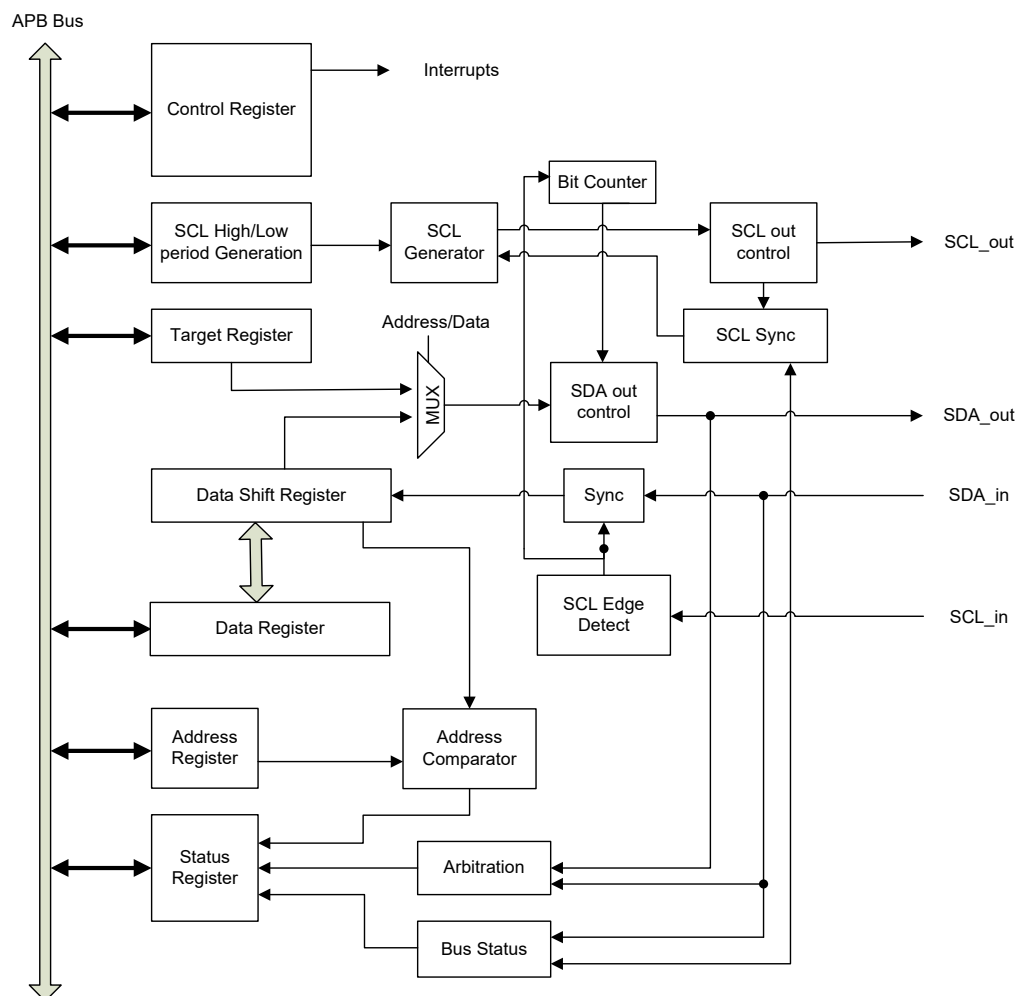


Figure 119. I<sup>2</sup>C Module Block Diagram

## 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 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
- Timeout function
- Supports PDMA Interface

## Functional Descriptions

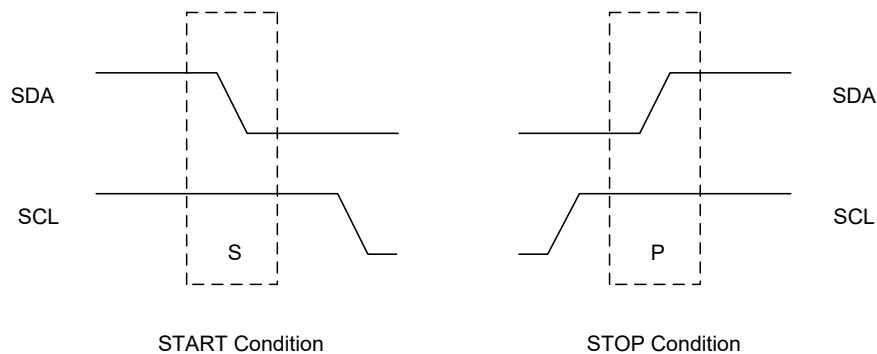
### Two-Wire Serial Interface

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 initiate 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.

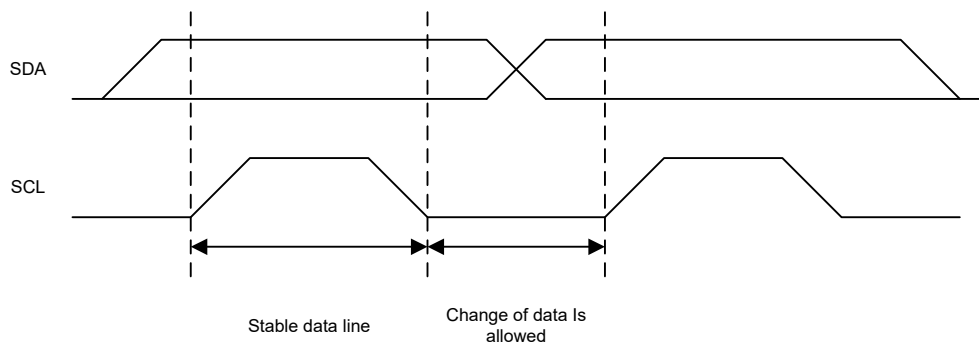
A repeated START, which is denoted as the "Sr" bit, is functionally identical to the normal START condition. A repeated START signal allows the I<sup>2</sup>C interface to communicate with another slave device or with the same device but in a different transfer direction without releasing the I<sup>2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



**Figure 120. 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 121. 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 the master device. The addressing mode selection bit named ADRM in the I2CCR register should be defined to choose either the 7-bit or 10-bit addressing mode.

#### 7-Bit Address Format

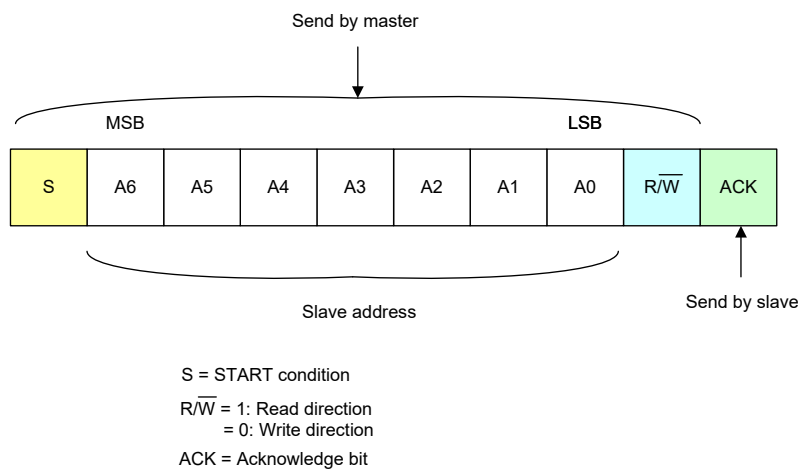
The 7-bit address format is composed of the seven-bit length slave address, which the master device wants to communicate with, an R/ $\overline{W}$  bit and an ACK bit. The R/ $\overline{W}$  bit defines the direction of the data transfer.

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

R/ $\overline{W}$  = 1 (Read): The master receives data from the addressed slave.

The slave address can be assigned through the ADDR field in the I2CADDR register. The slave device sends back the acknowledge bit (ACK) if its slave address matches the transmitted address sent by master.

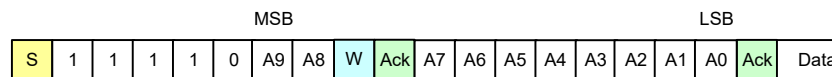
Note that it is forbidden to own the same address for two slave devices.



**Figure 122. 7-Bit Addressing Mode**

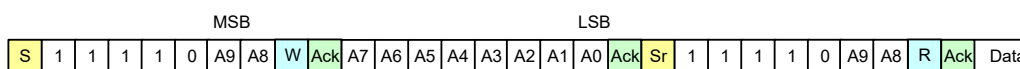
### 10-Bit Address Format

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 bits of the slave device address.



S = START condition  
 W = Write command  
 Ack = Acknowledge  
 A9 ~ A0 = 10-bit Address

**Figure 123. 10-Bit Addressing Write Transmit Mode**



S = START condition  
 Sr = Repeated-START condition  
 W = Write command  
 R = Read command  
 Ack = Acknowledge  
 A9 ~ A0 = 10-bit Address

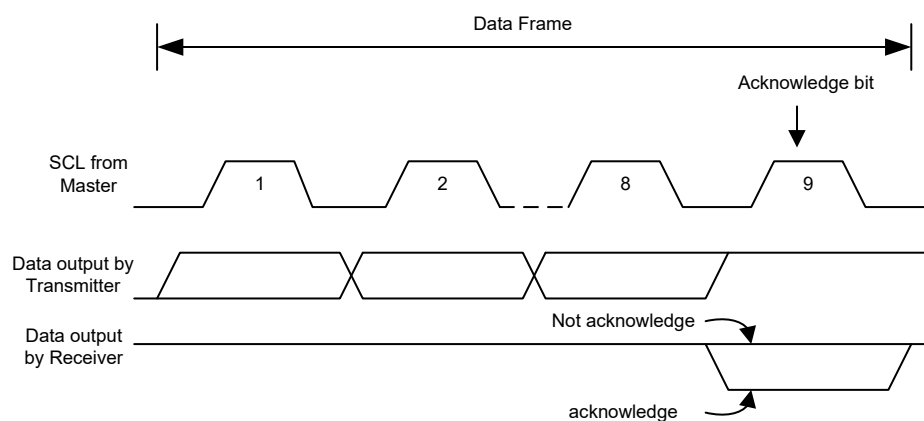
**Figure 124. 10-Bit Addressing Read Receive Mode**

## 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/ $\overline{W}$  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.

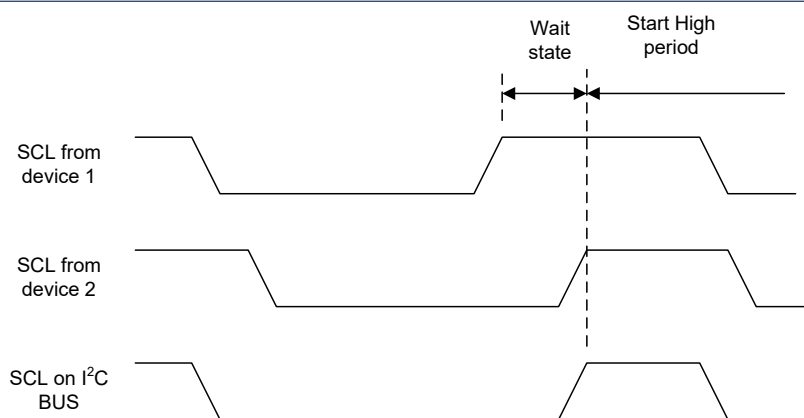
If the master device sends a Not Acknowledge (NACK) signal to the slave device, the slave device should release the SDA line for the master device to generate a STOP signal to terminate the transfer.



**Figure 125. 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.

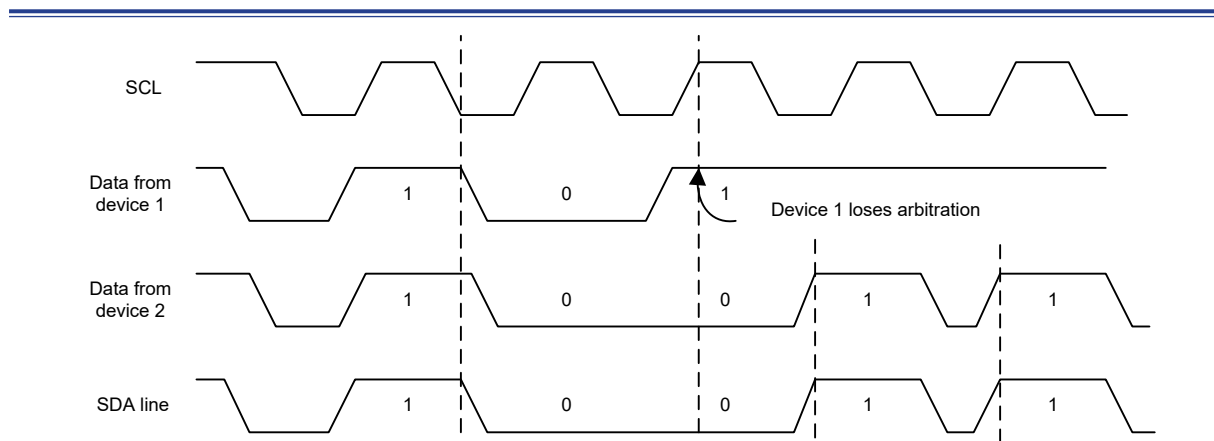


**Figure 126. Clock Synchronization During Arbitration**

## 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, ARBLOSIE, 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 127. Two Master Arbitration Procedure**

## General Call Addressing

The general call addressing function can be used to address all the devices connected to the I<sup>2</sup>C bus. The master device can activate the general call function by configuring the TAR field value to zero and clearing 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 I2CADDR register value as 0x05 and the I2CADDR register value as 0x55, this means if an address which is sent by an I<sup>2</sup>C master on the bus is equal to 0x50, 0x51, 0x54 or 0x55, 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, I2CADDRSR, 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 I2CADDRSR register is a read only register and each calling address on the I<sup>2</sup>C bus will be stored in the I2CADDRSR 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.

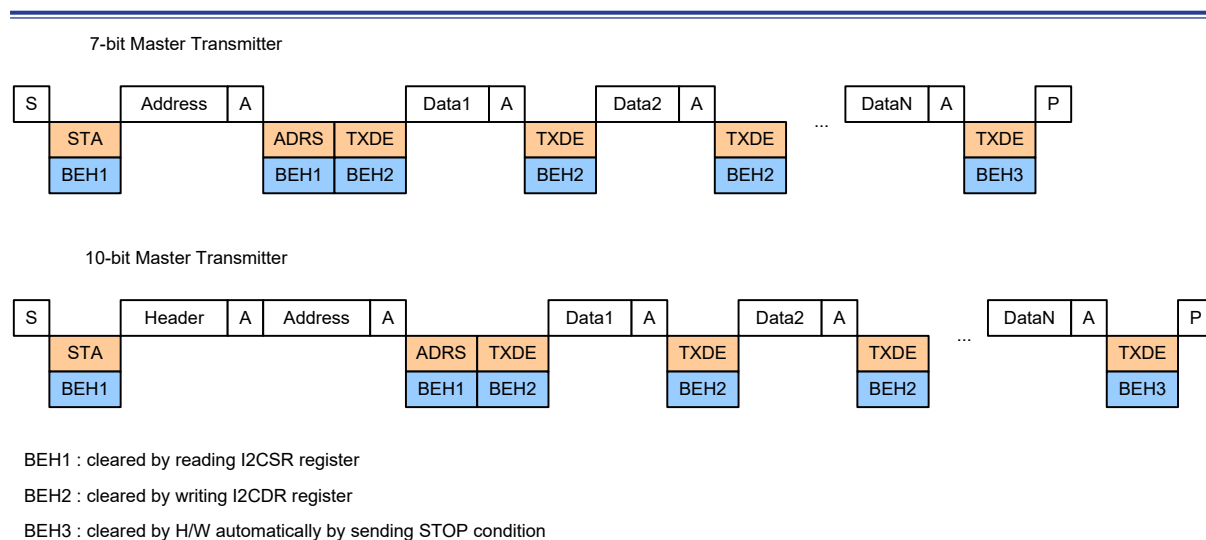
#### Data Frame

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.



**Figure 128. 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.

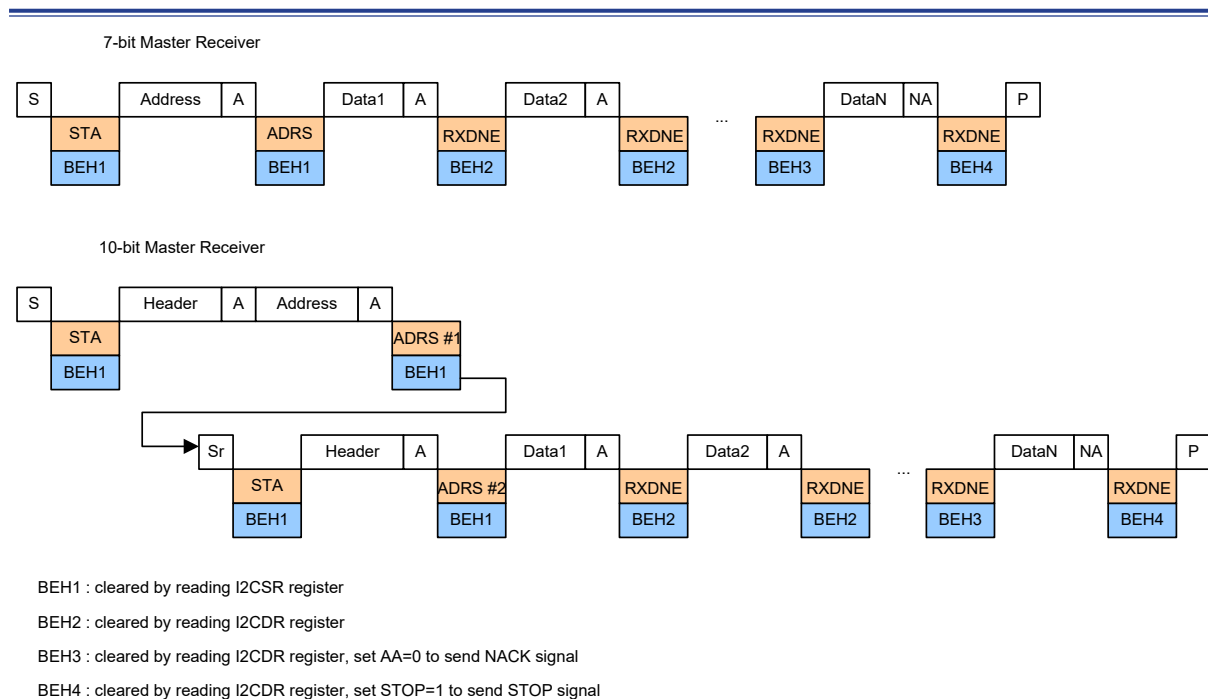
In the 10-bit addressing mode: The ADRS bit in the I2CSR register will be set twice in the 10-bit addressing mode. The first time the ADRS bit is set is when the 10-bit address is sent and the acknowledge signal from the slave device is received. The second time the ADRS bit is set is when the header 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 129. Master Receiver Timing Diagram**

## Slave Transmitter Mode

## Address Frame

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

## Data Frame

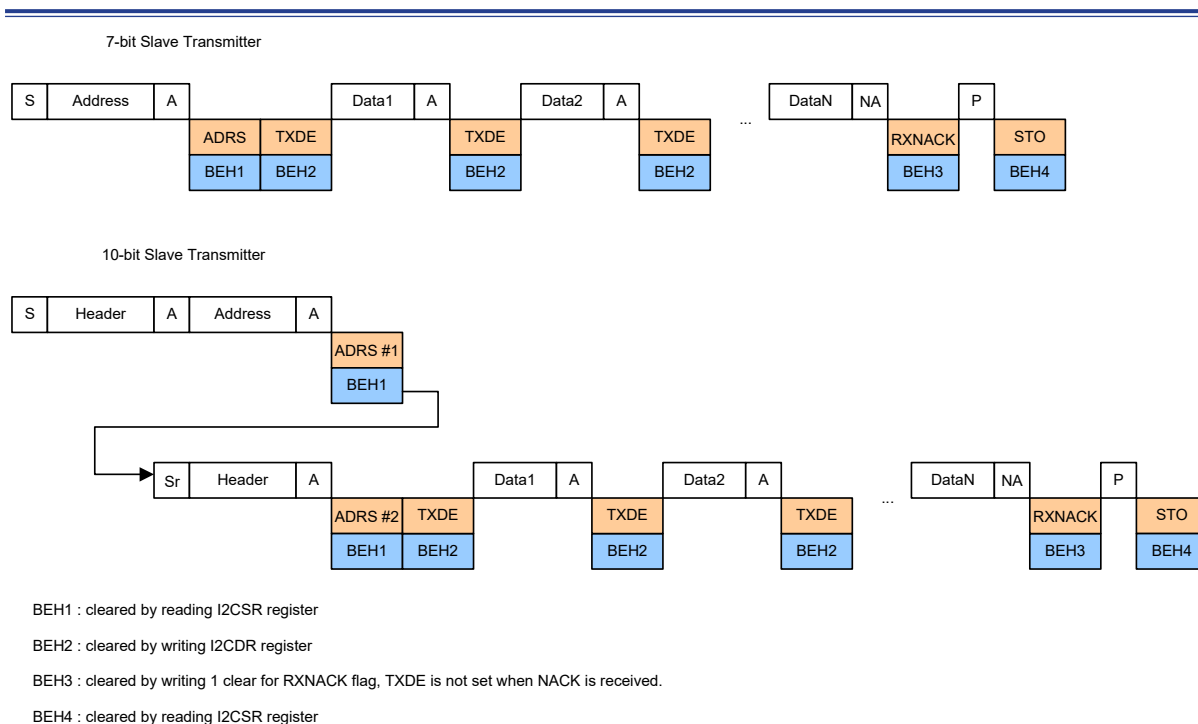
In the Slave transmitter mode, the TXDE bit is set to indicate that the I2CDR is empty, which results in the SCL line being held at a logic low state. New transmission data must then be written into the I2CDR register to continue the data transfer process. Writing a data into the I2CDR register will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

### Receive Not-Acknowledge

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

## STOP Condition

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



### Figure 130. Slave Transmitter Timing Diagram

## Slave Receiver Mode

### Address Frame

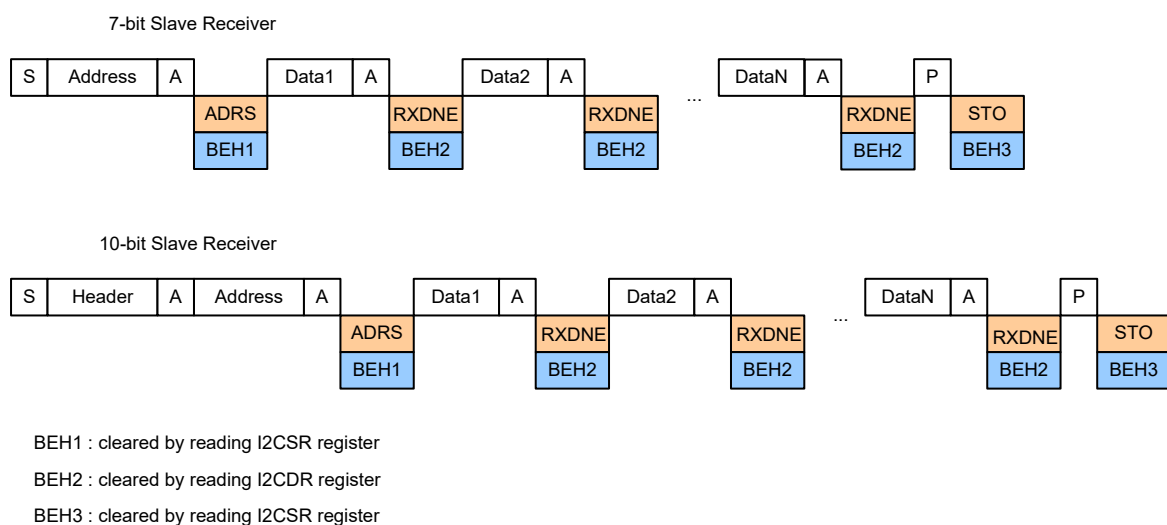
The ADRS bit in the I2CSR register is set after the slave device receives the calling address which matches with the slave device address. After the ADRS bit has been set to 1, it must be cleared to 0 to continue the data transfer process. The ADRS flag is cleared after reading the I2CSR register.

### 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 131. 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 45. Conditions of Holding SCL line**

Type	Condition	Description	Eliminated
Flag	TXDE	I <sup>2</sup> C is used in transmitter 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.128 and Fig.129) Slave: I <sup>2</sup> C is addressed as slave device (Note: Reference Fig.130 and Fig.131)	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 sends NACK used in receiver mode	Occurred when receiving the last data byte in Master received mode (Note: Reference Fig.129, and RXNACK flag won't be asserted 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,  $f_{I2CTO}$ , which is specified by the timeout prescaler field in the I2CTOUT register. The TOUT field in the I2CTOUT register is used to define the timeout counter preload value. The timeout function is enabled by setting the ENTOUT bit in the I2CCR register. The timeout counter will start to count down from the preloaded value if the ENTOUT bit is set to 1 and one of the following conditions occurs:

- The 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 flag in the I2CSR register is asserted.

If the timeout counter underflows, the corresponding timeout flag, TOUTF, in the I2CSR register will be set to 1 and a timeout interrupt will be generated if the relevant interrupt is enabled.

## PDMA Interface

The PDMA interface is integrated in the 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 more 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 46. I<sup>2</sup>C Register Map**

Register	Offset	Description	Reset Value
I2CCR	0x000	I <sup>2</sup> C Control Register	0x0000_2000
I2CIER	0x004	I <sup>2</sup> C Interrupt Enable Register	0x0000_0000
I2CADDR	0x008	I <sup>2</sup> C Address Register	0x0000_0000
I2CSR	0x00C	I <sup>2</sup> C Status Register	0x0000_0000
I2CSHPGR	0x010	I <sup>2</sup> C SCL High Period Generation Register	0x0000_0000
I2CSLPGR	0x014	I <sup>2</sup> C SCL Low Period Generation Register	0x0000_0000
I2CDR	0x018	I <sup>2</sup> C Data Register	0x0000_0000
I2CTAR	0x01C	I <sup>2</sup> C Target Register	0x0000_0000
I2CADDRMR	0x020	I <sup>2</sup> C Address Mask Register	0x0000_0000
I2CADDRSR	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

Reset value: 0x0000\_2000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	SEQFILTER	COMBFILTEREN	ENTOUT	Reserved	DMANACK	RXDMAE	TXDMAE	
	RW 0	RW 0	RW 1	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
Type/Reset	ADRM	Reserved			I2CEN	GCEN	STOP	AA
	RW 0				RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:14]	SEQFILTER	SDA or SCL Input Sequential Filter Configuration Bits 00: Sequential filter is disabled 01: 1 PCLK glitch filter 1x: 2 PCLK glitch filter Note: This setting would affect the frequency of SCL. Details are described in I2CSLPGR register.
[13]	COMBFILTEREN	SDA or SCL Input Combinational Filter Enable Bit 0: Combinational filter is disabled 1: Combinational filter is enabled
[12]	ENTOUT	I <sup>2</sup> C Timeout Function Enable Control 0: Timeout Function is disabled 1: Timeout Function is enabled This bit is used to enable or disable the I <sup>2</sup> C timeout function. It is recommended that users have to properly configure the PSC and TOUT fields in the I2CTOUT register before the timeout counter starts to count by setting the ENOUT bit to 1.
[10]	DMANACK	DMA Mode NACK Control 0: No operation 1: The I <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
[9]	RXDMAE	DMA Mode RX Request Enable Control 0: RX DMA request is disabled 1: RX DMA request is 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.



Bits	Field	Descriptions
[8]	TXDMAE	DMA Mode TX Request Enable Control 0: TX DMA request is disabled 1: TX DMA request is 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.
[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.
[3]	I2CEN	I <sup>2</sup> C Interface Enable 0: I <sup>2</sup> C interface is disabled 1: I <sup>2</sup> C interface is enabled
[2]	GCEN	General Call Enable 0: General call is disabled 1: General call is 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.
[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

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

Bits	Field	Descriptions
[18]	RXBFIE	RX Buffer Full Interrupt Enable Bit 0: Interrupt is disabled 1: Interrupt is enabled
[17]	TXDEIE	Data Register Empty Interrupt Enable Bit in Transmitter Mode 0: Interrupt is disabled 1: Interrupt is enabled
[16]	RXDNEIE	Data Register Not Empty Interrupt Enable Bit in Receiver Mode 0: Interrupt is disabled 1: Interrupt is enabled
[11]	TOUTIE	Timeout Interrupt Enable Bit 0: Interrupt is disabled 1: Interrupt is enabled
[10]	BUSERRIE	Bus Error Interrupt Enable Bit 0: Interrupt is disabled 1: Interrupt is enabled
[9]	RXNACKIE	Received Not Acknowledge Interrupt Enable Bit 0: Interrupt is disabled 1: Interrupt is enabled
[8]	ARBLOSIE	Arbitration Loss Interrupt Enable Bit in the I <sup>2</sup> C multi-master mode 0: Interrupt is disabled 1: Interrupt is enabled
[3]	GCSIE	General Call Slave Interrupt Enable Bit 0: Interrupt is disabled 1: Interrupt is enabled
[2]	ADRSIE	Slave Address Match Interrupt Enable Bit 0: Interrupt is disabled 1: Interrupt is enabled

Bits	Field	Descriptions
[1]	STOIE	STOP Condition Detected Interrupt Enable Bit 0: Interrupt is disabled 1: Interrupt is enabled The bit is used for the I <sup>2</sup> C slave mode only.
[0]	STAIE	START Condition Transmit Interrupt Enable Bit 0: Interrupt is disabled 1: Interrupt is enabled 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

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

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	
			RO	0	RO	0	RO	0	RO
	15	14	13	12	11	10	9	8	
Type/Reset	Reserved				TOUTF	BUSERR	RXNACK	ARBLOS	
					WC	0	WC	0	WC
	7	6	5	4	3	2	1	0	
Type/Reset	Reserved				GCS	ADRS	STO	STA	
					RC	0	RC	0	RC

Bits	Field	Descriptions
[21]	TXNRX	Transmitter / Receiver Mode 0: Receiver mode 1: Transmitter mode Read only bit.
[20]	MASTER	Master Mode 0: I <sup>2</sup> C is in the slave mode or idle 1: I <sup>2</sup> C is in the master mode The I <sup>2</sup> C interface is switched as a master device on the I <sup>2</sup> C bus when the I2CTAR register is assigned and the I <sup>2</sup> C bus is idle. The MASTER bit is cleared by hardware when software disables the I <sup>2</sup> C bus by clearing the I2CEN bit to 0 or sends a STOP condition to the I <sup>2</sup> C bus or the bus error is detected. This bit is set and cleared by hardware and is a read only bit.
[19]	BUSBUSY	Bus Busy 0: I <sup>2</sup> C bus is idle 1: I <sup>2</sup> C bus is busy The I <sup>2</sup> C interface hardware starts to detect the I <sup>2</sup> C bus status if the interface is enabled by setting the I2CEN bit to 1. It is set to 1 when the SDA or SCL signal is detected to have a logic low state and cleared when a STOP condition is detected.
[18]	RXBF	Buffer Full Flag in Receiver Mode 0: Data buffer is not full 1: Data buffer is full This bit is set when the data register I2CDR has already stored a data byte and meanwhile the data shift register also has been received a complete new data byte. The RXBF bit is cleared by software reading the I2CDR register.

Bits	Field	Descriptions
[17]	TXDE	<p>Data Register Empty Using in Transmitter Mode</p> <p>0: Data register I2CDR is not empty 1: Data register I2CDR is empty</p> <p>This bit is set when the I2CDR register is empty in the Transmitter mode. Note that the TXDE bit will be set after the address frame is being transmitted to inform that the data to be transmitted should be loaded into the I2CDR register. The TXDE bit is cleared by software writing data to the I2CDR register in both the master and slave mode or cleared automatically by hardware after setting the STOP signal to terminate the data transfer or setting the I2CTAR register to restart a new data transfer in the master mode.</p>
[16]	RXDNE	<p>Data Register Not Empty in Receiver Mode</p> <p>0: Data register I2CDR is empty 1: Data register I2CDR is not empty</p> <p>This bit is set when the I2CDR register is not empty in the receiver mode. The RXDNE bit is cleared by software reading the data byte from the I2CDR register.</p>
[11]	TOUTF	<p>Timeout Counter Underflow Flag</p> <p>0: No timeout counter underflow has occurred 1: Timeout counter underflow has occurred</p> <p>Writing "1" to this bit will clear the TOUTF flag.</p>
[10]	BUSERR	<p>Bus Error Flag</p> <p>0: No bus error has occurred 1: Bus error has occurred</p> <p>This bit is set by hardware when the I<sup>2</sup>C interface detects a misplaced START or STOP condition in a transfer process. Writing a "1" to this bit will clear the BUSERR flag.</p> <p>In Master Mode: Once the Bus Error event occurs, both the SDA and SCL lines are released by hardware and the BUSERR flag is asserted. The application software has to clear the BUSERR flag before the next address byte is transmitted.</p> <p>In Slave Mode: Once a misplaced START or STOP condition has been detected by the slave device, the software must clear the BUSERR flag before the next address byte is received.</p>
[9]	RXNACK	<p>Received Not Acknowledge Flag</p> <p>0: Acknowledge is returned from receiver 1: Not Acknowledge is returned from receiver</p> <p>The RXNACK bit indicates that the not Acknowledge signal is received in master or slave transmitter mode. Writing "1" to this bit will clear the RXNACK flag.</p>
[8]	ARBLOS	<p>Arbitration Loss Flag</p> <p>0: No arbitration loss is detected 1: Bit arbitration loss is detected</p> <p>This bit is set by hardware on the current clock which the I<sup>2</sup>C interface loses the bus arbitration to another master during the address or data frame transmission. Writing "1" to this bit will clear the ARBLOS flag. Once the ARBLOS flag is asserted by hardware, the ARBLOS flag must be cleared before the next transmission.</p>
[3]	GCS	<p>General Call Slave Flag</p> <p>0: No general call slave occurs 1: I<sup>2</sup>C interface is addressed by a general call command</p> <p>When the I<sup>2</sup>C interface receives an address with a value of 0x00 or 0x000 in the 7-bit or 10-bit addressing mode, if both the GCEN and the AA bit are set to 1, then it is switched as a general call slave. This flag is cleared automatically after being read.</p>

Bits	Field	Descriptions
[2]	ADRS	<p>Address Transmit (master mode) / Address Receive (slave mode) Flag</p> <p>Address Sent in Master Mode</p> <p>0: Address frame has not been transmitted</p> <p>1: Address frame has been transmitted</p> <p>For the 7-bit addressing mode, this bit is set after the master device receives the address frame acknowledge bit sent from the slave device. For the 10-bit addressing mode, this bit is set after receiving the acknowledge bit of the first header byte and the second address. Note that when the second header byte, if exists, is acknowledged, this bit will also be set.</p> <p>Address Matched in Slave Mode</p> <p>0: I<sup>2</sup>C interface is not addressed</p> <p>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 is detected</p> <p>1: STOP condition is detected in slave mode</p> <p>This bit is only available for the slave mode and is cleared automatically after the I2CSR register is read.</p>
[0]	STA	<p>START Condition Transmit</p> <p>0: No START condition is detected</p> <p>1: START condition is transmitted in master mode</p> <p>This bit is only available for the master mode and is cleared automatically after the I2CSR register is read.</p>

## 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

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

Bits	Field	Descriptions
[15:0]	SHPG	<p>SCL Clock High Period Generation</p> <p>High period duration setting <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 SEQFILTER in the I<sup>2</sup>C Control Register (I2CCR).</p> <p>If SEQFILTER=00, d=6</p> <p>If SEQFILTER=01, d=8</p> <p>If SEQFILTER=10 or 11, d=9</p>

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

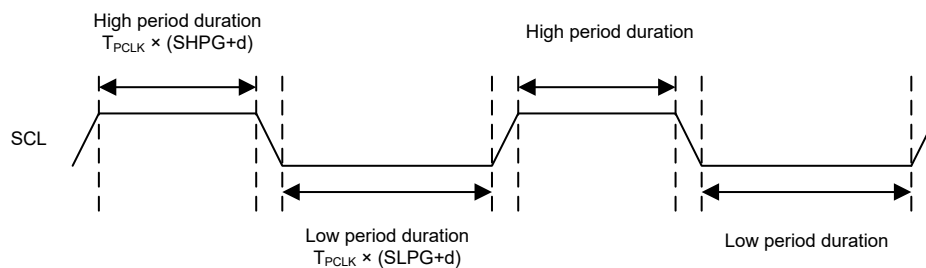
This register specifies the I<sup>2</sup>C SCL clock low period interval.

Offset : 0x014

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	SLPG							
	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
Type/Reset	SLPG							
	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 SEQFILTER in the I<sup>2</sup>C Control Register (I2CCR).</p> <p>If SEQFILTER=00, d=6</p> <p>If SEQFILTER=01, d=8</p> <p>If SEQFILTER=10 or 11, d=9</p>



**Figure 132. SCL Timing Diagram**

**Table 47. I<sup>2</sup>C Clock Setting Example**

I <sup>2</sup> C Clock	$T_{SCL} = T_{PCLK} \times [ (SHPG + d) + (SLPG + d) ]$ (where d = 6) SHPG + SLPG value at PCLK			
	8MHz	24MHz	48MHz	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



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

Reset value: 0x0000 0000

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

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

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

Bits	Field	Descriptions
[10]	RWD	Read or Write Direction 0: Write direction to target slave address 1: Read direction from target slave address If this bit is set to 1 in the 10-bit master receiver mode, the I <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 – I2CADDRMR

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

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

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

## I<sup>2</sup>C Address Snoop Register – I2CADDRSR

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

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

Bits	Field	Descriptions
[9:0]	ADDRSR	Address Snoop Once the I2CEN bit is enabled, the calling address value on the I <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

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

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

# 20 Serial Peripheral Interface (SPI)

## Introduction

The Serial Peripheral Interface, SPI, provides an SPI protocol data transmit and receive functions in both master or slave mode. The SPI interface uses 4 pins, among which are the serial data input and output lines SPI\_MISO and SPI\_MOSI, the clock line SPI\_SCK, and the slave select line SPI\_SEL. One SPI device acts as a master who controls the data flow using the SEL and SCK signals to indicate the start of the data communication and the data sampling rate. To receive the data bits, the streamlined data bits which range from 1 bit to 16 bits specified by the DFL field in the SPICR1 register are latched on a specific clock edge and stored in the data register or in the RX FIFO. Data transmission is carried out in a similar way but with the reverse sequence. The mode fault detection provides a capability for multi-master applications.

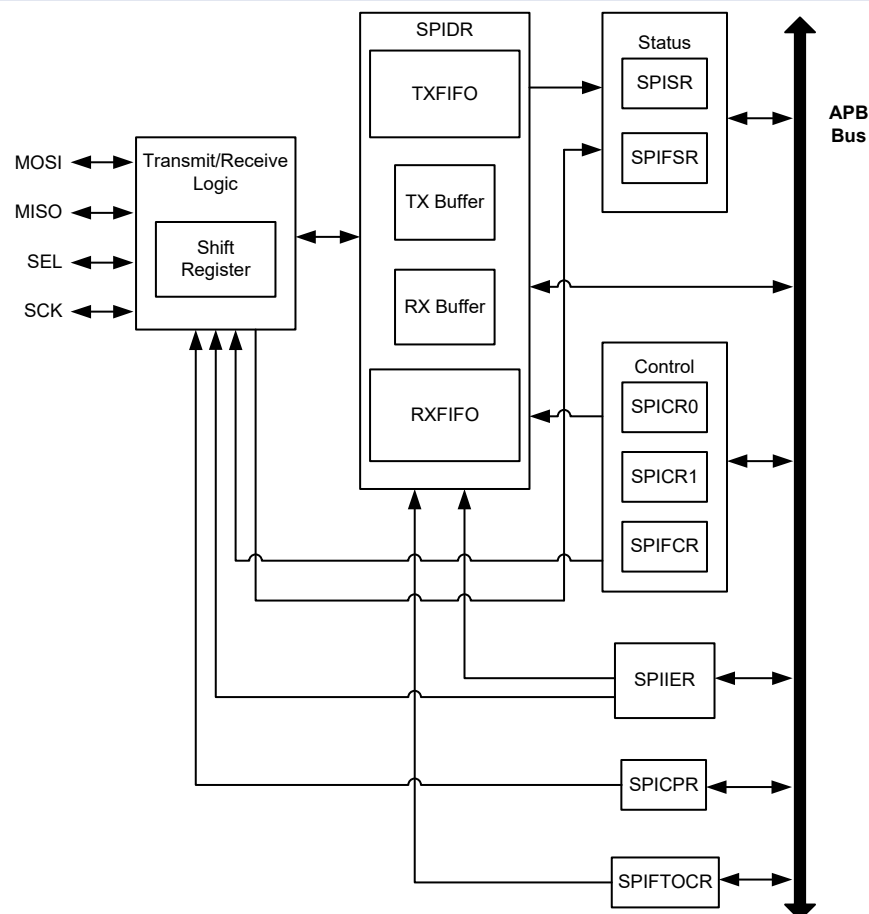


Figure 133. SPI Block Diagram

## Features

- Master or slave mode
- Master mode speed up to  $f_{\text{PCLK}}/2$
- Slave mode speed up to  $f_{\text{PCLK}}/3$
- Programmable data frame length up to 16 bits
- FIFO Depth: 8 levels
- MSB or LSB first shift selection
- Programmable slave select high or low active polarity
- Multi-master and multi-slave operation
- Master mode supports dual output read mode of SPI series NOR Flash
- Four error flags with individual interrupt
  - Read overrun
  - Write collision
  - Mode fault
  - Slave abort
- Support PDMA interface

## Functional Descriptions

### Master Mode

Each data frame can range from 1 to 16 bits in data length. The first bit of the transmitted data can be either an MSB or LSB determined by the FIRSTBIT bit in the SPICR1 register. The SPI module is configured as a master or a slave by setting the MODE bit in the SPICR1 register. When the MODE bit is set, the SPI module is configured as a master and will generate the serial clock on the SPI\_SCK pin. The data stream will transmit data in the shift register to the SPI\_MOSI pin on the serial clock edge. The SPI\_SEL pin is active during the full data transmission. When the SELAP bit in the SPICR1 register is set, the SPI\_SEL pin is active high during the complete data transactions. When the SELM bit in the SPICR1 register is set, the SPI\_SEL pin will be driven by the hardware automatically and the time interval between the active SEL edge and the first edge of SCK is equal to half an SCK period.

### Slave Mode

In the slave mode, the SPI\_SCK pin acts as an input pin and the serial clock will be derived from the external master device. The SPI\_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  $f_{\text{PCLK}}$ , 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.

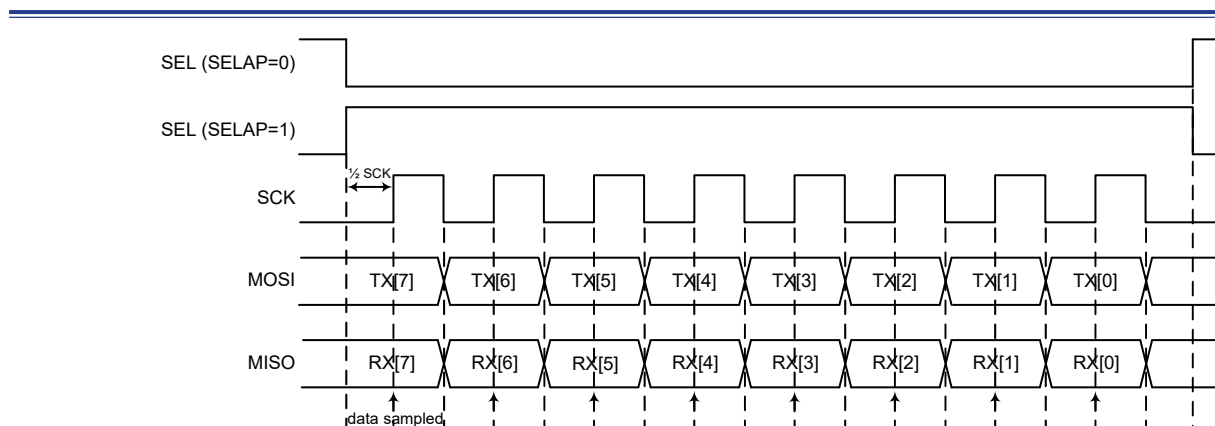
There are four formats contained in the SPI interface. Table 54 shows how to configure these formats by setting the FORMAT field in the SPICR1 register.

**Table 48. SPI Interface Format Setup**

FORMAT [2:0]	CPOL	CPHA
001	0	0
010	0	1
110	1	0
101	1	1
Others	Reserved	

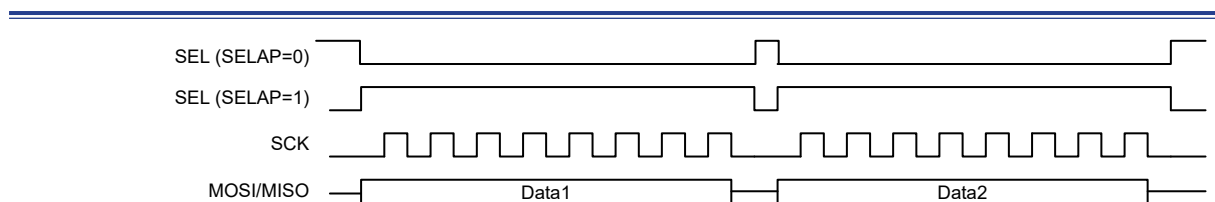
**CPOL = 0, CPHA = 0**

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



**Figure 134. SPI Single Byte Transfer Timing Diagram – CPOL = 0, CPHA = 0**

Figure 181 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.

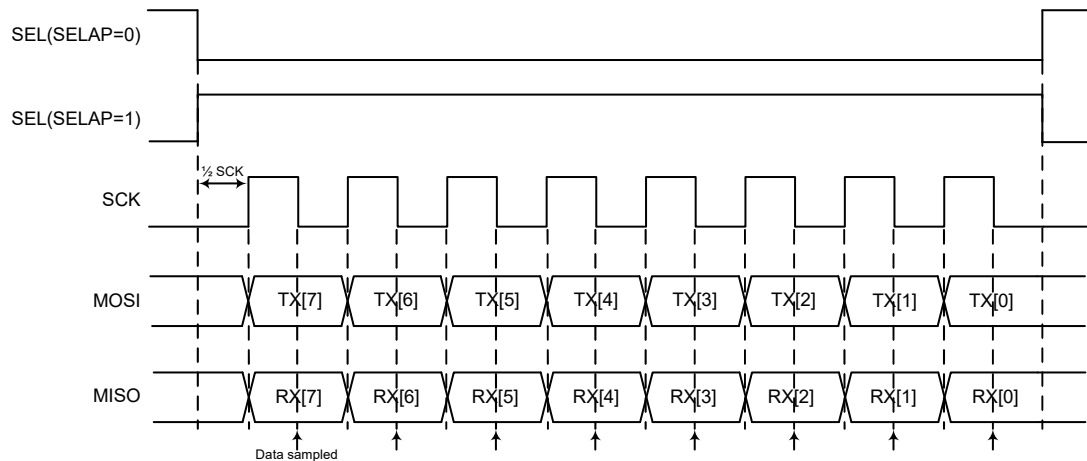


**Figure 135. SPI Continuous Data Transfer Timing Diagram – CPOL = 0, CPHA = 0**



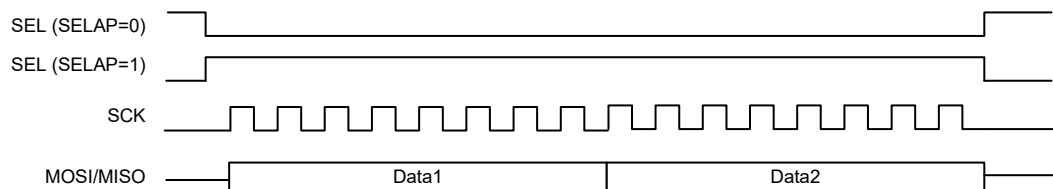
### CPOL = 0, CPHA = 1

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 136 shows the single data byte transfer timing.



**Figure 136. SPI Single Byte Transfer Timing Diagram – CPOL = 0, CPHA = 1**

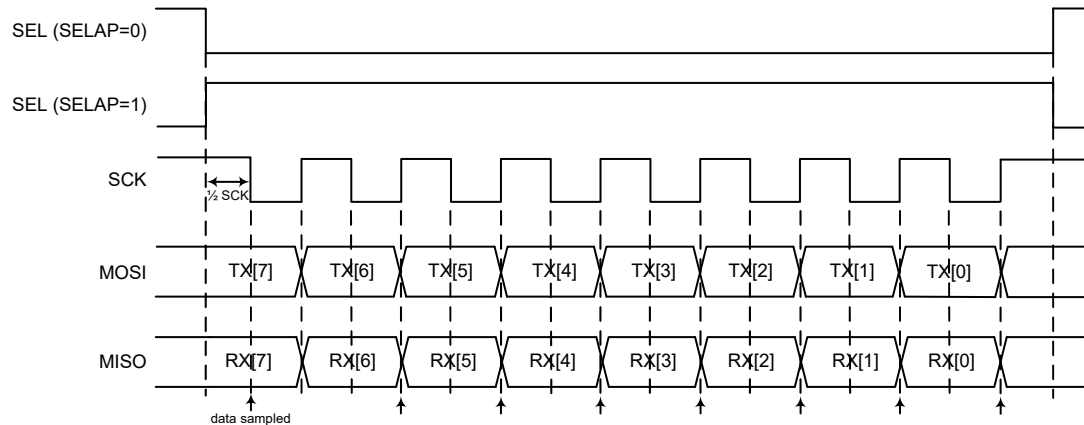
Figure 183 shows the continuous data transfer diagram timing. Note that the SEL signal must remain active until the last data transfer has completed.



**Figure 137. SPI Continuous Transfer Timing Diagram – CPOL = 0, CPHA = 1**

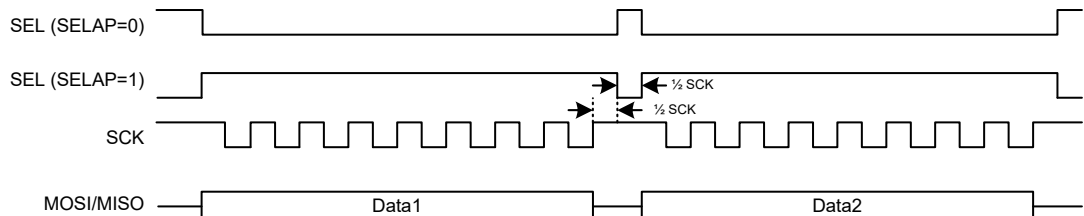
### CPOL = 1, CPHA = 0

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 138 shows the single byte transfer timing of this format.



**Figure 138. SPI Single Byte Transfer Timing Diagram – CPOL = 1, CPHA = 0**

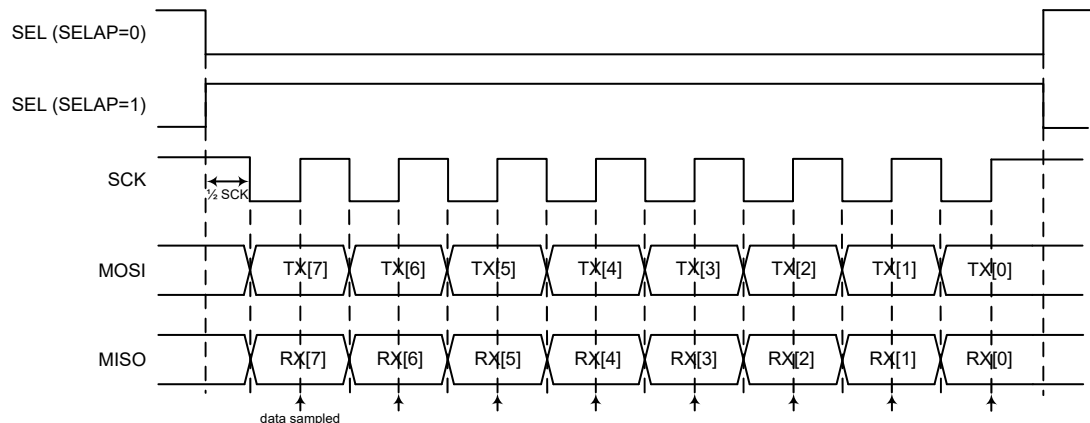
Figure 185 shows the continuous data transfer timing of this format. Note that the SEL signal must change to an inactive level between each data frame.



**Figure 139. SPI Continuous Transfer Timing Diagram – CPOL = 1, CPHA = 0**

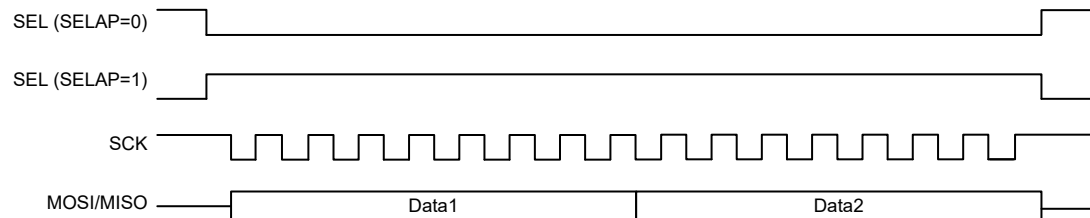
### CPOL = 1, CPHA = 1

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 140 shows the single byte transfer timing of this format.



**Figure 140. SPI Single Byte Transfer Timing Diagram – CPOL = 1, CPHA = 1**

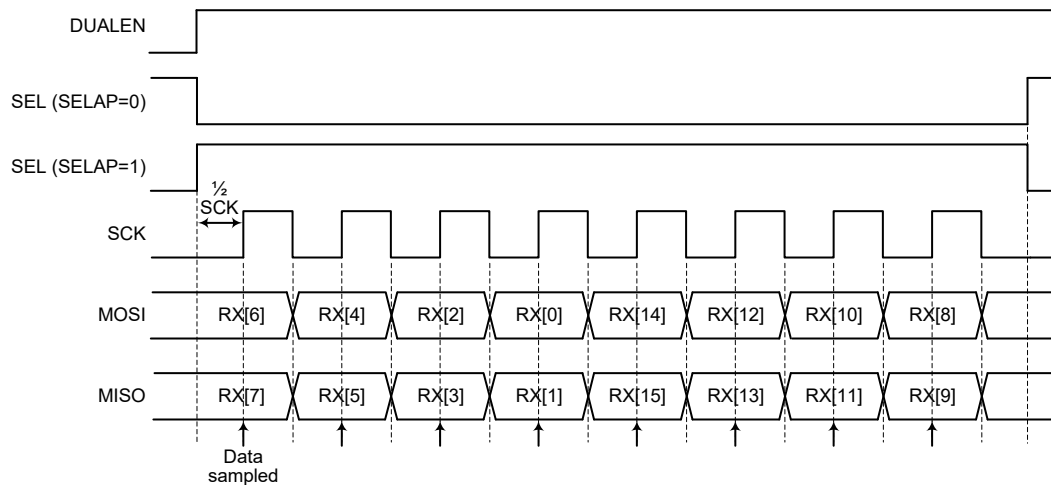
Figure 187 shows the continuous data transfer timing of this format. Note that the SEL signal must remain active until the last data transfer has completed.



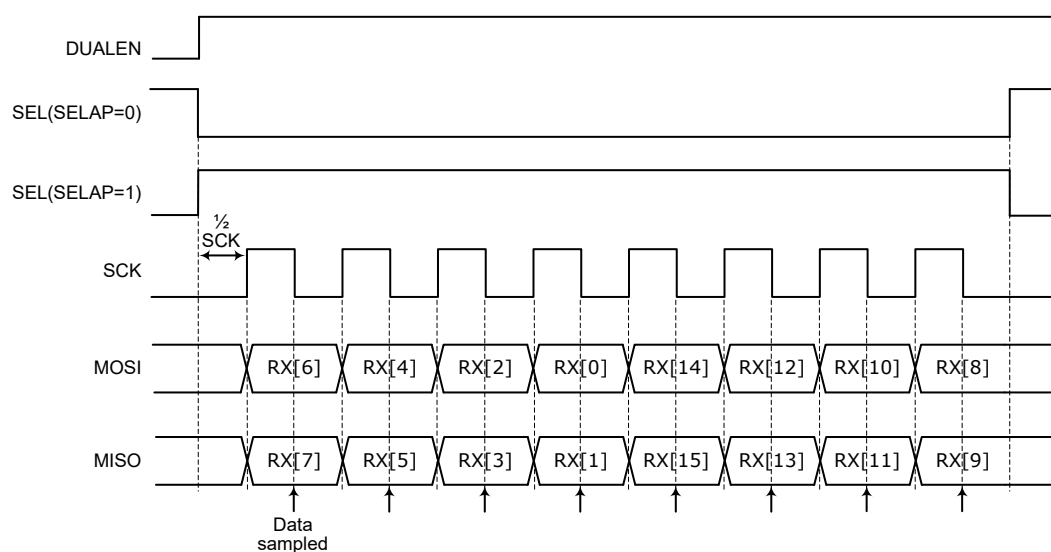
**Figure 141. SPI Continuous Transfer Timing Diagram – CPOL = 1, CPHA = 1**

## SPI Dual Mode

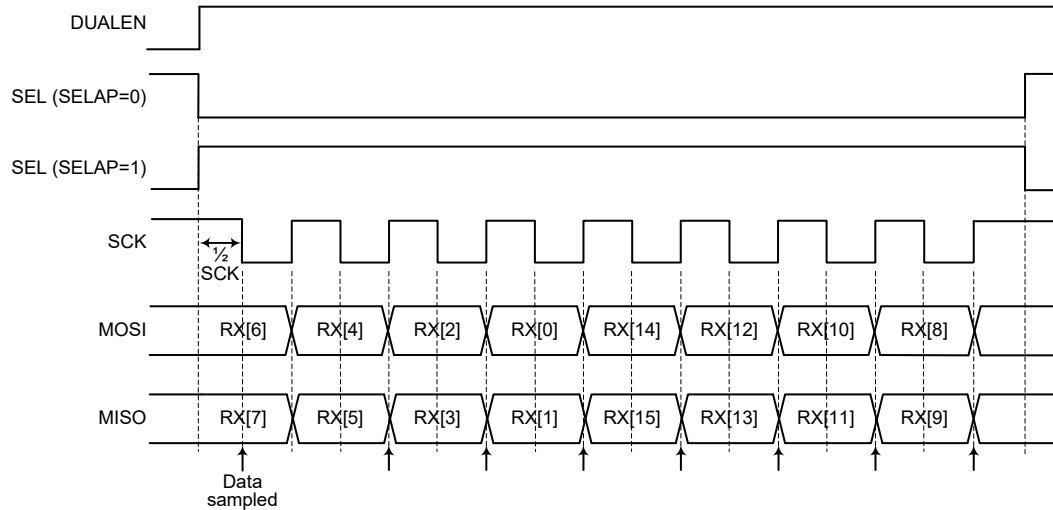
When in the Master mode, the SPI interface operation can be configured to Dual mode. A more efficient data transfer can then be implemented by using this Dual mode together with the four formats described above. In the Dual mode, the SPI data transmission only supports input direction, that is, the SPI\_MOSI pin is also switched from output to an input function. In this way a two-wire transfer method is formed to read data from an external device synchronously. In addition, the Dual mode only supports a data length of 16-bit (DFL=0x8). The Dual mode is commonly used to read data from an external serial SPI Flash. The following figures show the transfer format bit sequences in the SPI Dual mode.



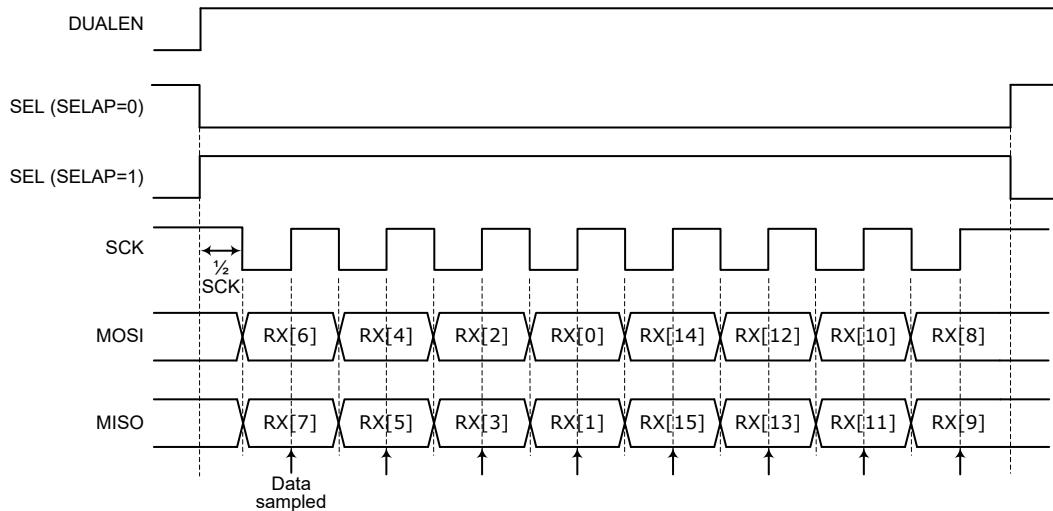
**Figure 142. SPI Dual Mode Bit Sequence – CPOL = 0, CPHA = 0, DFL = 0x8 (16-bit), MSB Transmitted First**



**Figure 143. SPI Dual Mode Bit Sequence – CPOL = 0, CPHA = 1, DFL = 0x8 (16-bit), MSB Transmitted First**

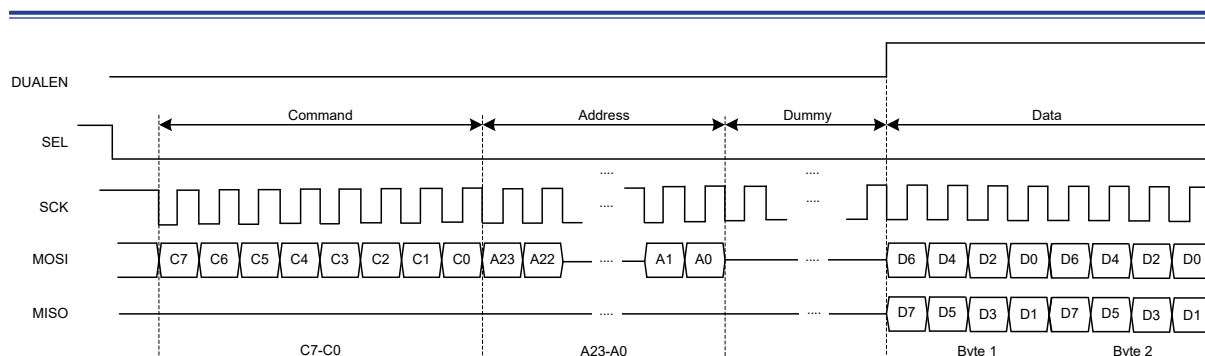


**Figure 144. SPI Dual Mode Bit Sequence – CPOL = 1, CPHA = 0, DFL = 0x8 (16-bit) , MSB Transmitted First**



**Figure 145. SPI Dual Mode Bit Sequence – CPOL = 1, CPHA = 1, DFL = 0x8 (16-bit), MSB Transmitted First**

Figure 192 shows the bit sequence of the SPI Dual mode reading data from an external serial SPI Flash.



**Figure 146. SPI Dual Mode Data Read Example – CPOL = 1, CPHA = 1**

## Status Flags

### TX Buffer Empty – TXBE

This TXBE flag is set when the TX buffer is empty in the non-FIFO mode or when the TX FIFO data length is equal to or less than the TX FIFO threshold level as defined by the TXFTLS field in the SPIFCR register in the FIFO mode. The following data to be transmitted can then be loaded into the buffer again. After this, the TXBE flag will be reset when the TX buffer already contains new data in the non-FIFO mode or when the TX FIFO data length is greater than the TX FIFO threshold level determined by the TXFTLS field in FIFO mode.

### Transmission Register Empty – TXE

This TXE flag is set when both the TX buffer and the TX shift registers are empty. It will be reset when the TX buffer or the TX shift register contains new transmitted data.

### RX Buffer Not Empty – RXBNE

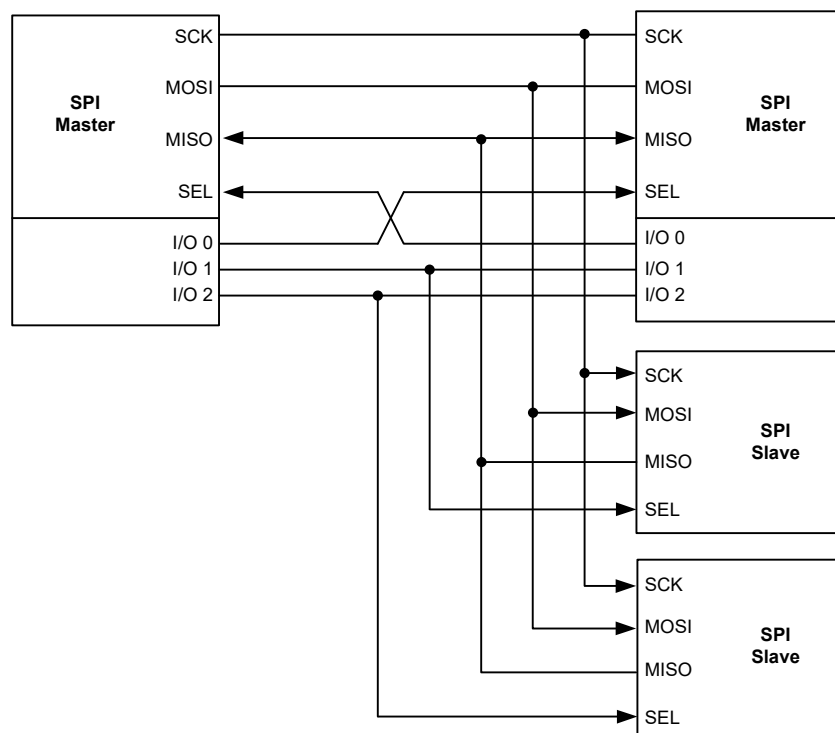
This RXBNE flag is set when there is valid received data in the RX buffer in the non-FIFO mode or the RX FIFO data length is equal to or greater than the RX FIFO threshold level as defined by the RXFTLS field in the SPIFCR register in the SPI FIFO mode. This flag will be automatically cleared by hardware when the received data have been read out from the RX buffer totally in the non-FIFO mode or when the RX FIFO data length is less than the RX FIFO threshold level set in the RXFTLS field.

### Time Out Flag – TO

The time out function is only available in the SPI FIFO mode and is disabled by loading a zero value into the TOC field in the Time Out Counter register. The time out counter will start counting if the SPI RX FIFO is not empty, once data is read from the SPIDR register or new data is received, the time out counter will be reset to 0 and count again. When the time out counter value is equal to the value specified by the TOC field in the SPIFTOCR register, the TO flag will be set. The flag is cleared by writing 1 to this bit.

### Mode Fault – MF

The mode fault flag can be used to detect SPI bus usage in the SPI multi-master mode. For the multi-master mode, the SPI module is configured as a master device and the SEL signal is set as an input signal. The mode fault flag is set when the SPI\_SEL pin is suddenly changed to an active level by another SPI master. This means that another SPI master is requesting to use the SPI bus. Therefore, when an SPI mode fault occurs, it will force the SPI module to operate in the slave mode and also disable all of the SPI interface signals to avoid SPI bus signal collisions. For the same reason, if the SPI master wants to transfer data, it also needs to inform other SPI masters by driving their SEL signals to an active state. The detailed configuration diagram for the SPI multi-master mode is shown in the following figure.



**Figure 147. SPI Multi-Master Slave Environment**

**Table 49. SPI Mode Fault Trigger Conditions**

Mode Fault	Descriptions
Trigger Condition	<ol style="list-style-type: none"> <li>1. SPI Master mode.</li> <li>2. SELOEN = 0 in the SPICR0 register – SPI_SEL pin is configured to be the input mode.</li> <li>3. SEL signal changes to an active level when driven by the external SPI master.</li> </ol>
SPI Behavior	<ol style="list-style-type: none"> <li>1. Mode fault flag is set.</li> <li>2. The SPIEN bit in the SPICR0 register is reset. This disables the SPI interface and blocks all output signals from the device.</li> <li>3. The MODE bit in the SPICR1 register is reset. This forces the device into slave mode.</li> </ol>

**Table 50. SPI Master Mode SPI\_SEL Pin Status**

	SEL as Input – SELOEN = 0		SEL as Output – SELOEN = 1	
Multi-Master	Supported		Not supported	
SPI SEL Control Signal	Use Another GPIO to replace the SPI_SEL pin function		SPI_SEL pin in hardware or software control mode – using SELM setting	
Continuous Transfer	Case 1	Case 2	Case 1	Case 2
	Not supported	Supported	Hardware control	Hardware or software control

**Case 1:** SEL signal must be inactive between each data transfer.

**Case 2:** SEL signal will not to be inactive until the last data frame has finished.

**Note:** When the SPI is in the slave mode, the SEL signal is always an input and not affected by the SELOEN bit in the SPICR0 register.

#### Write Collision – WC

The following conditions will assert the Write Collision Flag.

- The FIFOEN bit in the SPIFCR register is cleared  
The write collision flag is asserted when new data is written into the SPIDR register while both the TX buffer and the shift register are already full. Any new data written into the TX buffer will be lost.
- The FIFOEN bit in the SPIFCR register is set  
The write collision flag is asserted to indicate that new data is written into the SPIDR register while both the TX FIFO and the TX shift register are already full. Any new data written into the TX FIFO will be lost.

#### Read Overrun – RO

- The FIFOEN bit in the SPIFCR register is cleared  
The read overrun flag is asserted to indicate that both the RX shift register and the RX buffer are already full, if one more data is received. This will result in the newly received data not being shifted into the SPI shift register. As a result the latest received data will be lost.
- The FIFOEN bit in the SPIFCR register is set  
The read overrun flag is set to indicate that the RX shift register and the RX FIFO are both full, if one more data is received. This means that the latest received data can not be shifted into the SPI shift register. As a result the latest received data will be lost.

#### Slave Abort – SA

In the SPI slave mode, the slave abort flag is set to indicate that the SPI\_SEL pin suddenly changed to an inactive state during the reception of a data frame transfer. The data frame length is set by the DFL field in the SPICR1 register.



## PDMA Interface

The PDMA interface is integrated in the SPI module. The PDMA function can be enabled by setting the TXDMAE or RxDMAE bit to 1 in the transmitter or receiver mode respectively. When the transmit buffer empty flag, TXBE, is asserted and the TXDMAE bit is set to 1, the PDMA function will be activated to move data from the memory location that users designated into the SPI data register or the TX FIFO until the TXBE flag is cleared to 0. The TXBE flag will be asserted when the transmit buffer is empty in the non-FIFO mode or the data contained in the TX FIFO is equal to or less than the level defined by the TXFTLS field in the FIFO mode.

Similarly, when the receive buffer not empty flag, RXBNE, is asserted and the RxDMAE bit is set to 1, the PDMA function will be activated to move data from the SPI data register or the RX FIFO to the memory location that users designated until the RXBNE flag is cleared to 0. The RXBNE flag will be asserted when the receive buffer is not empty in the non-FIFO mode or the data contained in the RX FIFO is equal to or greater than the level defined by the RXFTLS field in the FIFO mode.

For a more detailed description about the PDMA configurations, refer to the PDMA chapter.

## Register Map

The following table shows the SPI registers and reset values.

**Table 51. SPI Register Map**

Register	Offset	Description	Reset Value
SPICR0	0x000	SPI Control Register 0	0x0000_0000
SPICR1	0x004	SPI Control Register 1	0x0000_0000
SPIIER	0x008	SPI Interrupt Enable Register	0x0000_0000
SPICPR	0x00C	SPI Clock Prescaler Register	0x0000_0000
SPIDR	0x010	SPI Data Register	0x0000_0000
SPISR	0x014	SPI Status Register	0x0000_0003
SPIFCR	0x018	SPI FIFO Control Register	0x0000_0000
SPIFSR	0x01C	SPI FIFO Status Register	0x0000_0000
SPIFTOCR	0x020	SPI FIFO Time Out Counter Register	0x0000_0000

## Register Descriptions

### SPI Control Register 0 – SPICR0

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

Offset: 0x000

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[15:12]	SELHT	Chip Select Hold Time 0x0: 1/2 SCK 0x1: 1 SCK 0x2: 3/2 SCK 0x3: 2 SCK .... Note that SELHT is for master mode only.
[11:8]	GUADT	Guard Time GUADTEN = 1 0x0: 1 SCK 0x1: 2 SCK 0x2: 3 SCK ... Note that GUADT is for master mode only.
[7]	GUADTEN	Guard Time Enable 0: Guard Time is 1/2 SCK 1: When this bit is set, guard time can be controlled by GUADT Note that GUADTEN is for master mode only.
[6]	DUALEN	Dual Mode Enable 0: Dual Mode is disabled 1: Dual Mode is enabled The control bit is used to support the dual output read mode of the series SPI NOR Flash. When this bit is set and the MOSI signal will change the direction from output to input and receive the series data stream. That means the DUALEN control bit is only for master mode.

Bits	Field	Descriptions
[4]	SSELC	Software Slave Select Control 0: Set the SEL output to an inactive state 1: Set the SEL output to an active state The application software can set the SEL output to an active or inactive state by configuring the SSELC bit. The active level is configured by the SELAP bit in the SPICR1 register. Note that the SSELC bit is only available when the SELOEN bit is set to 1 for enabling the SEL output meanwhile the SELM bit is cleared to 0 for controlling the SEL signal by software. Otherwise, the SSELC bit has no effect.
[3]	SELOEN	Slave Select Output Enable 0: Set the SEL signal to the input mode for multi-master mode 1: Set the SEL signal to the output mode for slave select The SELOEN is only available in the master mode to set the SEL signal as an input or output signal. When the SEL signal is configured to operate in the output mode, it is used as a slave select signal in either the hardware or software mode according to the SELM bit setting in the SPICR1 register. The SEL signal is used for mode fault detection in the multi-master environment when it is configured to operate in the input mode
[2]	RXDMAE	RX PDMA request enable 0: SPI RX path PDMA request is disabled 1: SPI RX path PDMA request is enabled
[1]	TXDMAE	TX PDMA request enable 0: SPI TX path PDMA request is disabled 1: SPI TX path PDMA request is enabled
[0]	SPIEN	SPI Enable 0: SPI interface is disabled 1: SPI interface is enabled

## SPI Control Register 1 – SPICR1

This register specifies the SPI parameters including the data length, the transfer format, the SEL active polarity/ mode, the LSB/MSB control and the master/slave mode.

Offset: 0x004

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[15]	SCKDUTY	SCK pulse high/low duty 0: High-Duty is longer than Low-Duty 1: High-Duty is shorter than Low-Duty The frequency of SCK is calculated by $f_{PCLK}/n$ . When n is odd, the SCK high/low duty is not 50%. In this case, the ratio of the high/low duty can be selected by this control bit.
[14]	MODE	Master or Slave Mode 0: Slave mode 1: Master mode
[13]	SELM	Slave Select Mode 0: SEL signal is controlled by software – asserted or de-asserted by the SSEL bit 1: SEL signal is controlled by hardware – generated automatically by the SPI hardware Note that the SELM bit is available for master mode only, i.e., MODE = 1.
[12]	FIRSTBIT	LSB or MSB Transmitted First 0: MSB is transmitted first 1: LSB is transmitted first
[11]	SELAP	Slave Select Active Polarity 0: SEL signal is active low 1: SEL signal is active high

Bits	Field	Descriptions																																							
[10:8]	FORMAT	<p>SPI Data Transfer Format</p> <p>These three bits are used to determine the data transfer format of the SPI interface.</p> <table border="1"> <thead> <tr> <th>FORMAT [2:0]</th><th>CPOL</th><th>CPHA</th></tr> </thead> <tbody> <tr> <td>001</td><td>0</td><td>0</td></tr> <tr> <td>010</td><td>0</td><td>1</td></tr> <tr> <td>110</td><td>1</td><td>0</td></tr> <tr> <td>101</td><td>1</td><td>1</td></tr> <tr> <td>Others</td><td colspan="2">Reserved</td></tr> </tbody> </table> <p>CPOL: Clock Polarity            0: SCK Idle state is low            1: SCK Idle state is high</p> <p>CPHA: Clock Phase            0: Data is captured on the first SCK clock edge            1: Data is captured on the second SCK clock edge</p>	FORMAT [2:0]	CPOL	CPHA	001	0	0	010	0	1	110	1	0	101	1	1	Others	Reserved																						
FORMAT [2:0]	CPOL	CPHA																																							
001	0	0																																							
010	0	1																																							
110	1	0																																							
101	1	1																																							
Others	Reserved																																								
[3:0]	DFL	<p>Data Frame Length</p> <p>Selects the data transfer frame from 1 bit to 16 bits.</p> <table border="1"> <thead> <tr> <th>DFL[3:0]</th><th>SPI Serial Mode</th><th>SPI Dual Mode</th></tr> </thead> <tbody> <tr> <td>0001</td><td>1 bit</td><td>—</td></tr> <tr> <td>0010</td><td>2 bits</td><td>—</td></tr> <tr> <td>0011</td><td>3 bits</td><td>—</td></tr> <tr> <td>0100</td><td>4 bits</td><td>—</td></tr> <tr> <td>0101</td><td>5 bits</td><td>—</td></tr> <tr> <td>0110</td><td>6 bits</td><td>—</td></tr> <tr> <td>0111</td><td>7 bits</td><td>—</td></tr> <tr> <td>1000</td><td>8 bits</td><td>16 bits</td></tr> <tr> <td>1001</td><td>9 bits</td><td>—</td></tr> <tr> <td>...</td><td>...</td><td>—</td></tr> <tr> <td>1111</td><td>15 bits</td><td>—</td></tr> <tr> <td>0000</td><td>16 bits</td><td>—</td></tr> </tbody> </table> <p>Notes: 1. The total number of data bits is determined by the DFL field configuration together with the selected SPI device transmission mode.            2. Taking the 16-bit data transmission for example, the DFL setting can be figured out as follows            In the Serial Mode: Data frame length = 16/1 = 16, DFL = 0x0;            In the Dual SPI Mode: Data frame length = 16/2 = 8, DFL = 0x8;            3. Dual mode only supports 16-bit data length.</p>	DFL[3:0]	SPI Serial Mode	SPI Dual Mode	0001	1 bit	—	0010	2 bits	—	0011	3 bits	—	0100	4 bits	—	0101	5 bits	—	0110	6 bits	—	0111	7 bits	—	1000	8 bits	16 bits	1001	9 bits	—	...	...	—	1111	15 bits	—	0000	16 bits	—
DFL[3:0]	SPI Serial Mode	SPI Dual Mode																																							
0001	1 bit	—																																							
0010	2 bits	—																																							
0011	3 bits	—																																							
0100	4 bits	—																																							
0101	5 bits	—																																							
0110	6 bits	—																																							
0111	7 bits	—																																							
1000	8 bits	16 bits																																							
1001	9 bits	—																																							
...	...	—																																							
1111	15 bits	—																																							
0000	16 bits	—																																							

## SPI Interrupt Enable Register – SPIIER

This register contains the corresponding SPI interrupt enable control bit.

Offset: 0x008

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[7]	TOIEN	Time Out Interrupt Enable 0: Disable 1: Enable
[6]	SAIEN	Slave Abort Interrupt Enable 0: Disable 1: Enable
[5]	MFIEEN	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  An interrupt is generated when the RXBNE flag is set and RXBNEIEN is set. In the FIFO mode, the interrupt being generated depends upon the RX FIFO trigger level setting.
[1]	TXEIEEN	Transmission Register Empty Interrupt Enable 0: Disable 1: Enable  The transmission register empty interrupt request will be generated when the TXE flag and the TXEIEEN bit are set.
[0]	TXBEIEN	TX Buffer Empty Interrupt Enable 0: Disable 1: Enable  The TX buffer empty interrupt request will be generated when the TXBE flag and the TXBEIEN bit are set. In the FIFO mode, the interrupt request being generated depends upon the TX FIFO trigger level setting.

## SPI Clock Prescaler Register – SPICPR

This register specifies the SPI clock prescaler ratio.

Offset: 0x00C

Reset value: 0x0000\_0000

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

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} / (CP + 1)</math>, where the CP ranges has to be from 1 to 65535</p> <p>Note: For the SPI master mode, the APB clock (<math>f_{PCLK}</math>) must be at least 2 times faster than the SPI SCK output.</p>

## SPI Data Register – SPIDR

This register stores the SPI received or transmitted Data.

Offset: 0x010

Reset value: 0x0000\_0000

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

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

## SPI Status Register – SPISR

This register contains the relevant SPI status.

Offset: 0x014

Reset value: 0x0000\_0003

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

Bits	Field	Descriptions
[8]	BUSY	<b>SPI Busy flag</b> 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	<b>Time Out flag</b> 0: No Rx FIFO time out 1: Rx FIFO time out has occurred 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	<b>Slave Abort flag</b> 0: No slave abort 1: Slave abort has occurred This bit is set by hardware and cleared by writing 1.



Bits	Field	Descriptions
[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.
[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	RX Buffer Not Empty flag 0: RX buffer is empty 1: RX buffer is not empty This bit indicates the RX buffer status in the non-FIFO mode. It is also used to indicate if the RX FIFO trigger level has been reached in the FIFO mode. This bit will be cleared when the SPI RX buffer is empty in the non-FIFO mode or if the number of data contained in RX FIFO is less than the trigger level which is specified by the RXFTLS field in the SPIFCR register in the SPI FIFO mode.
[1]	TXE	Transmission Register Empty flag 0: TX buffer or TX shift register is not empty 1: TX buffer and TX shift register both are empty
[0]	TXBE	TX Buffer Empty flag 0: TX buffer is not empty 1: TX buffer is empty In the FIFO mode, this bit if set indicates that the number of data contained in TX FIFO is equal to or less than the trigger level specified by the TXFTLS field in the SPIFCR register.

## SPI FIFO Control Register – SPIFCR

This register contains the related SPI FIFO control including the FIFO enable control and the FIFO trigger level selections.

Offset: 0x018

Reset value: 0x0000\_0000

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

Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0
Bits	Field	Descriptions												
[10]	FIFOEN	FIFO Enable 0: FIFO is disabled 1: FIFO is enabled This bit can not be set or reset when the SPI interface is in transmitting.												
[7:4]	RXFTLS	RX FIFO Trigger Level Select 0000: Trigger level is 0 0001: Trigger level is 1 ... 1000: Trigger level is 8 Others: Reserved The RXFTLS field is used to specify the RX FIFO trigger level. When the number of data contained in the RX FIFO is equal to or greater than the trigger level defined by the RXFTLS field, the RXBNE flag will be set												
[3:0]	TXFTLS	TX FIFO Trigger Level Select 0000: Trigger level is 0 0001: Trigger level is 1 ... 1000: Trigger level is 8 Others: Reserved The TXFTLS field is used to specify the TX FIFO trigger level. When the number of data contained in the TX FIFO is equal to or less than the trigger level defined by the TXFTLS field, the TXBE flag will be set.												

## SPI FIFO Status Register – SPIFSR

This register contains the relevant SPI FIFO status.

Offset: 0x01C

Reset value: 0x0000\_0000

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

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

### SPI FIFO Time Out Counter Register – SPIFCR

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

Offset: 0x020

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[15:0]	TOC	Time Out Counter Compare Value The time out counter starts to count from 0 after the SPI RX FIFO receives a data, and the counter value is reset once the data is read from the SPIDR register by software or another new data is received. If the FIFO does not receive new data or the software does not read data from the SPIDR register the time out counter value will continuously increase. When the time out counter value is equal to the TOC setting value, the TO flag in the SPISR register will be set and an interrupt will be generated if the TOIEN bit in the SPIIER register is set. The time out counter will be stopped when the RX FIFO is empty. The SPI FIFO time out function can be disabled by setting the TOC field to zero. The time out counter is driven by the system APB clock, named $f_{\text{CLK}}$ .

# 21 Universal Synchronous Asynchronous Receiver Transmitter (USART)

## Introduction

The Universal Synchronous Asynchronous Receiver Transceiver, USART, provides a flexible full duplex data exchange using synchronous or asynchronous transfer. The USART is used to translate data between parallel and serial interfaces, and is also commonly used for RS232 standard communication. The USART peripheral function supports a variety of interrupts.

The USART module includes an 8-level transmit FIFO, TX FIFO, and an 8-level receive FIFO, RX FIFO. Software can detect a USART error status by reading the USART Status & Interrupt Flag Register, USRSIFR. The status includes the condition of the transfer operations as well as several error conditions resulting from Parity, Overrun, Framing and Break events.

The USART includes a programmable baud rate generator which is capable of dividing the USART clock CK\_USART to produce a baud rate clock for the USART transmitter and receiver.

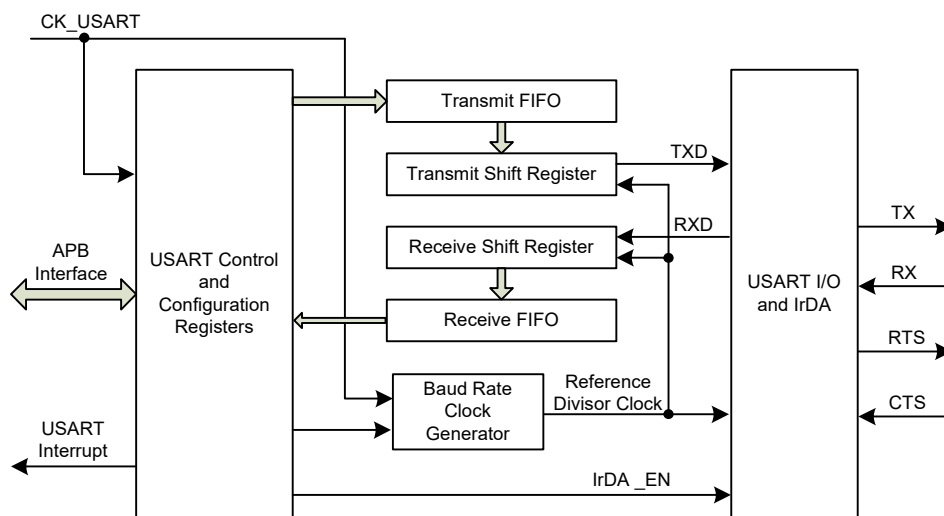


Figure 148. USART Block Diagram

## Features

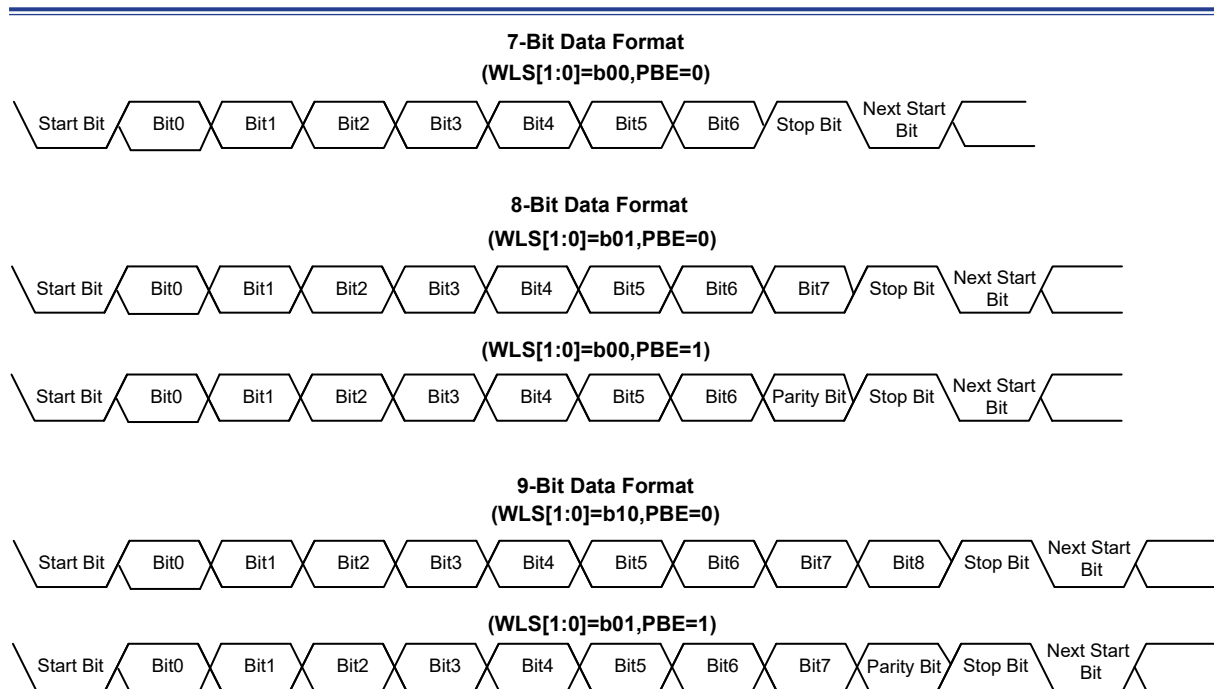
- Supports both asynchronous and clocked synchronous serial communication modes
- Full Duplex Communication Capability
- Programmable baud rate clock frequency up to ( $f_{\text{PLCK}}/16$ ) MHz for asynchronous mode and ( $f_{\text{PLCK}}/8$ ) MHz for synchronous mode
- IrDA SIR encoder and decoder
  - Support normal 3/16 bit duration and low-power (1.41 ~ 2.23  $\mu\text{s}$ ) durations
- Supports RS485 mode with output enable
- Auto hardware flow control mode – RTS, CTS
- Fully programmable serial communication functions including:
  - Word length: 7, 8 or 9-bit character
  - Parity: Even, odd, or no-parity bit generation and detection
  - Stop bit: 1 or 2 stop bits generation
  - Bit order: LSB-first or MSB-first transfer
- Error detection: Parity, overrun and frame error
- FIFO:
  - Receive FIFO: 8-level (max. 9 data bits)
  - Transmit FIFO: 8-level (max. 9 data bits)
- Supports PDMA Interface

## Functional Descriptions

### Serial Data Format

The USART module performs a parallel-to-serial conversion on data that is written to the transmit FIFO registers and then sends the data with the following format: Start bit, 7 ~ 9 LSB/MSB first data bits, optional Parity bit and finally 1 ~ 2 Stop bits. The Start bit has the opposite polarity of the data line idle state. The Stop bit is the same as the data line idle state and provides a delay before the next start situation. Both the Start and Stop bits are used for data synchronization during the asynchronous data transmission.

The USART module also performs a serial-to-parallel conversion on the data that is read from the receive FIFO registers. It will first check the Parity bit and will then look for a Stop bit. If the Stop bit is not found, the USART module will consider the entire word transmission as failed and respond with a Framing Error.



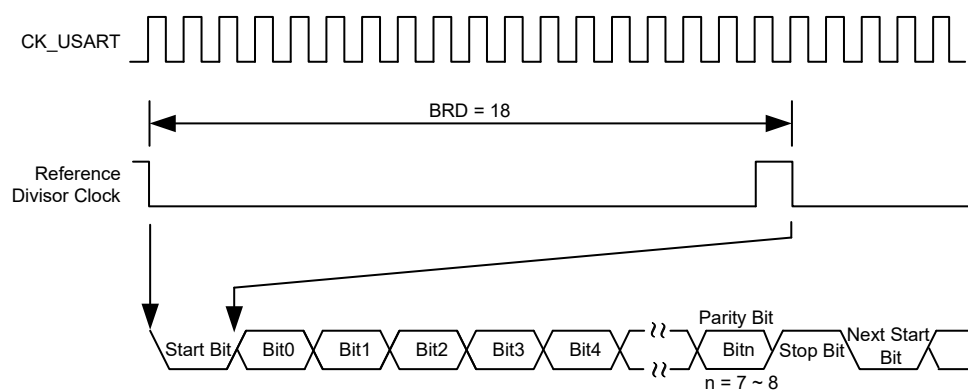
**Figure 149. USART Serial Data Format**

## 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 the CK\_USART clock is the APB clock connected to the USART while the BRD range is from 16 to 65535 for asynchronous mode and 8 to 65535 for synchronous mode.



**Figure 150. USART Clock CK\_USART and Data Frame Timing**

**Table 52. Baud Rate Deviation Error Calculation – CK\_USART = 40 MHz**

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

**Table 53. Baud Rate Deviation Error Calculation – CK\_USART = 48 MHz**

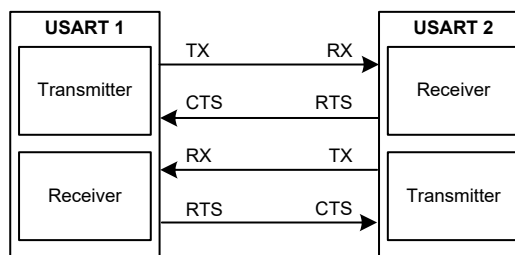
Baud Rate		CK_USART = 48 MHz		
No.	Kbps	Actual	BRD	Deviation Error Rate
1	2.4	2.4	20000	0.00%
2	9.6	9.6	5000	0.00%
3	19.2	19.2	2500	0.00%
4	57.6	57.6	833	0.04%
5	115.2	115.1	417	-0.08%
6	230.4	230.8	208	0.16%
7	460.8	461.5	104	0.16%
8	921.6	923.1	52	0.16%
9	2250	2285.7	21	1.59%
10	3000	3000.0	16	0.00%

**Table 54. Baud Rate Deviation Error Calculation – CK\_USART = 60 MHz**

Baud Rate		CK_USART = 60 MHz		
No.	Kbps	Actual	BRD	Deviation Error Rate
1	2.4	2.4	25000	0.00%
2	9.6	9.6	6250	0.00%
3	19.2	19.2	3125	0.00%
4	57.6	57.6	1042	-0.03%
5	115.2	115.2	521	-0.03%
6	230.4	230.8	260	0.16%
7	460.8	461.5	130	0.16%
8	921.6	923.1	65	0.16%
9	2250	2222.2	27	-1.23%
10	3000	3000.0	20	0.00%

## Hardware Flow Control

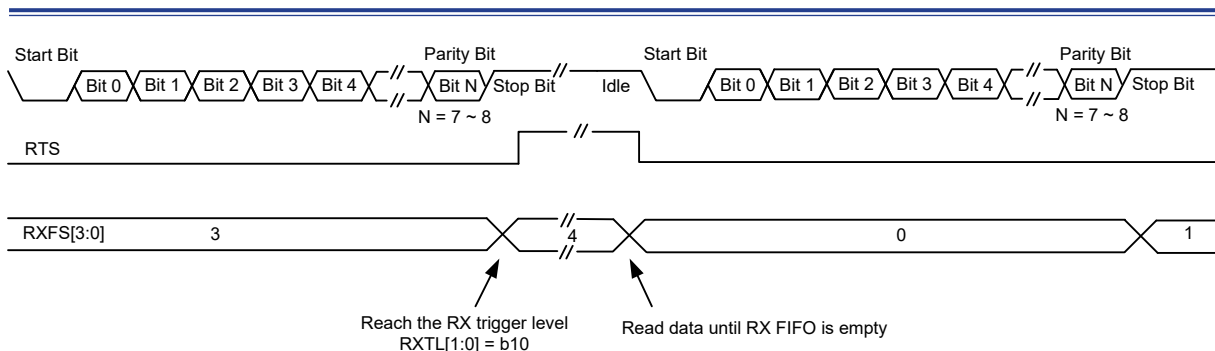
The USART supports the hardware flow control function which is enabled by setting the HFCEN bit in the USRCR register to 1. It is possible to control the serial data flow between two USART devices by using the CTS input and the RTS output. The Figure 151 shows the connection diagram in this mode. The hardware flow control function is categorized into two types. One is the RTS flow control function and the other is the CTS flow control function.



**Figure 151. Hardware Flow Control between 2 USARTs**

### RTS Flow Control

In the RTS flow control, the USART RTS pin is active with a logic low state when the receive data register is empty. It means that the receiver is ready to receive a new data. When the RX FIFO reaches the trigger level which is specified by configuring the RXTL field in the USRFCR register, the USART RTS pin is inactive with a logic high state. Figure 152 shows the example of RTS flow control.



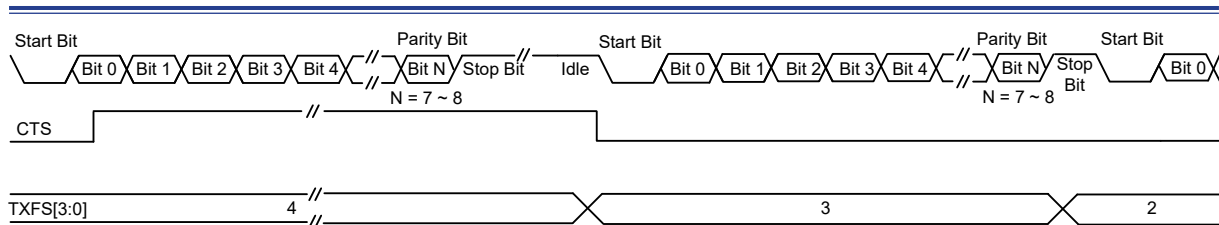
**Figure 152. USART RTS Flow Control**

### CTS Flow Control

If the hardware flow control function is enabled, the URTXEN bit in the USRCR register is controlled by the USART CTS input signal. If the USART CTS pin is forced to a logic low state, the URTXEN bit will automatically be set to 1 to enable the data transmission. However, if the USART CTS pin is forced to a logic high state, the URTXEN bit will be cleared to 0 and then the data transmission will also be disabled.



When the USART CTS pin is forced to a logic high state during a data transmission period, the current data transmission will be continued until the stop bit is completed. The Figure 153 shows an example of communication with CTS flow control.

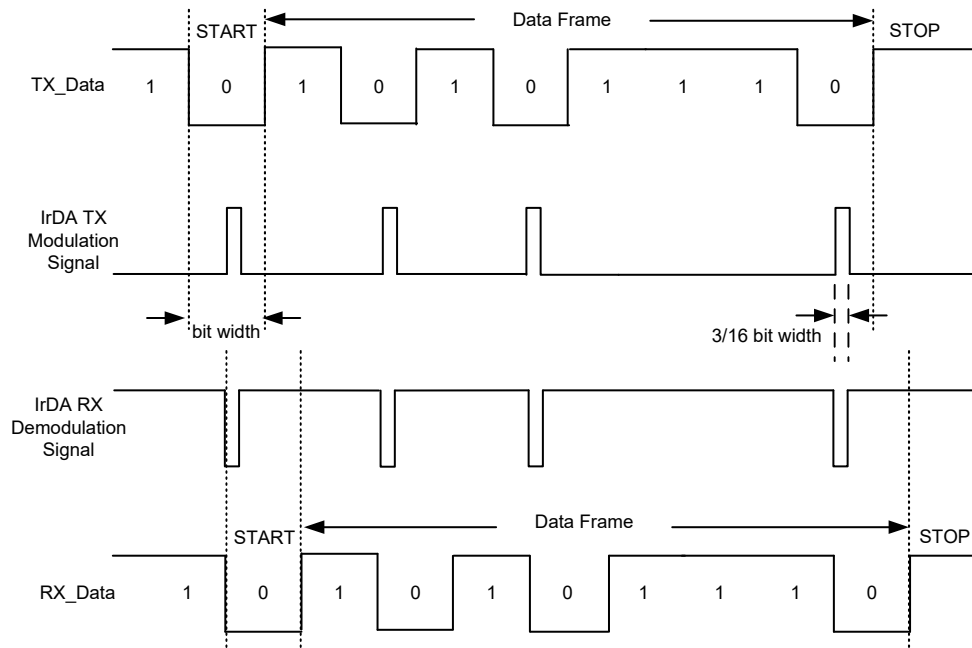


**Figure 153. USART CTS Flow Control**

## IrDA

The USART IrDA mode is provided for half-duplex point-to-point wireless communication.

The USART module includes an integrated modulator and demodulator which allow a wireless communication using infrared transceivers. The transmitter specifies a logic data '0' as a 'high' pulse and a logic data '1' as a 'low' level while the Receiver specifies a logic data '0' as a 'low' pulse and a logic data '1' as a 'high' level in the IrDA mode.



**Figure 154. IrDA Modulation and Demodulation**

The IrDA mode provides two operation modes, one is the normal mode, and the other is the low-power mode.

### IrDA Normal Mode

For the IrDA normal mode, the width of each transmitted pulse generated by the transmitter modulator is specified as 3/16 of the baud rate clock period. The receiver pulse width for the IrDA receiver demodulator is based on the IrDA receive debounce filter which is implemented using an 8-bit down-counting counter. The debounce filter counter value is specified by the IrDAPSC field in the IrDACR register. When a falling edge is detected on the receiver pin, the debounce filter counter starts to count down, driven by the CK\_USART clock. If a rising edge is detected on the receiver pin, the counter stops counting and is reloaded with the IrDAPSC value. When a low pulse falling edge on the receiver pin is detected and then before the debounce filter has counted down to zero, a rising edge is also detected, then this low pulse will be considered as glitch noise and will be discarded. If a low pulse falling edge appears on the receiver pin but no rising edge is detected before the debounce counter reaches 0, then the input is regarded as a valid data “0” for this bit duration. The IrDAPSC value must be set to be greater than or equal to 0x01, then the IrDA receiver demodulation operation can function properly. The IrDAPSC value can be adjusted to meet the USART baud rate setting to filter the IrDA received glitch noise of which the width is smaller than the prescaler setting duration.

### IrDA Low-Power Mode

In the IrDA low-power mode, the transmitted IrDA pulse width generated by the transmitter modulator is not kept at 3/16 of the baud rate clock period. Instead, the pulse width is fixed and is calculated by the following formula. The transmitted pulse width can be adjusted by the IrDAPSC field to meet the minimum pulse width specification of the external IrDA receiver device.

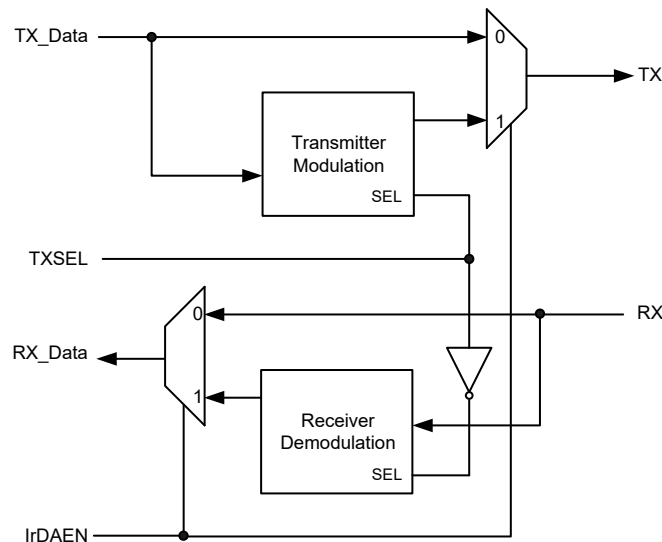
$$T_{\text{IrDA\_L}} = 3 \times \text{IrDAPSC} / \text{CK\_USART}$$

Note:  $T_{\text{IrDA\_L}}$  is the transmitted pulse width in the low-power mode.

The IrDAPSC field is the IrDACR prescaler value in the IrDA Control Register IrDACR.

The debounce behavior in the IrDA low-power receiving mode is similar to the IrDA normal mode. For glitch detection, the low pulse of which the pulse width is shorter than  $1 \times (\text{IrDAPSC} / \text{CK\_USART})$  should be discarded in the IrDA receiver demodulation. A valid low data is accepted if its low pulse width is greater than  $2 \times (\text{IrDAPSC} / \text{CK\_USART})$  duration.

The IrDA physical layer specification specifies a minimum delay with a value of 10 ms between the transmission and reception switch; and this IrDA receiver set-up time also should be managed by the software.



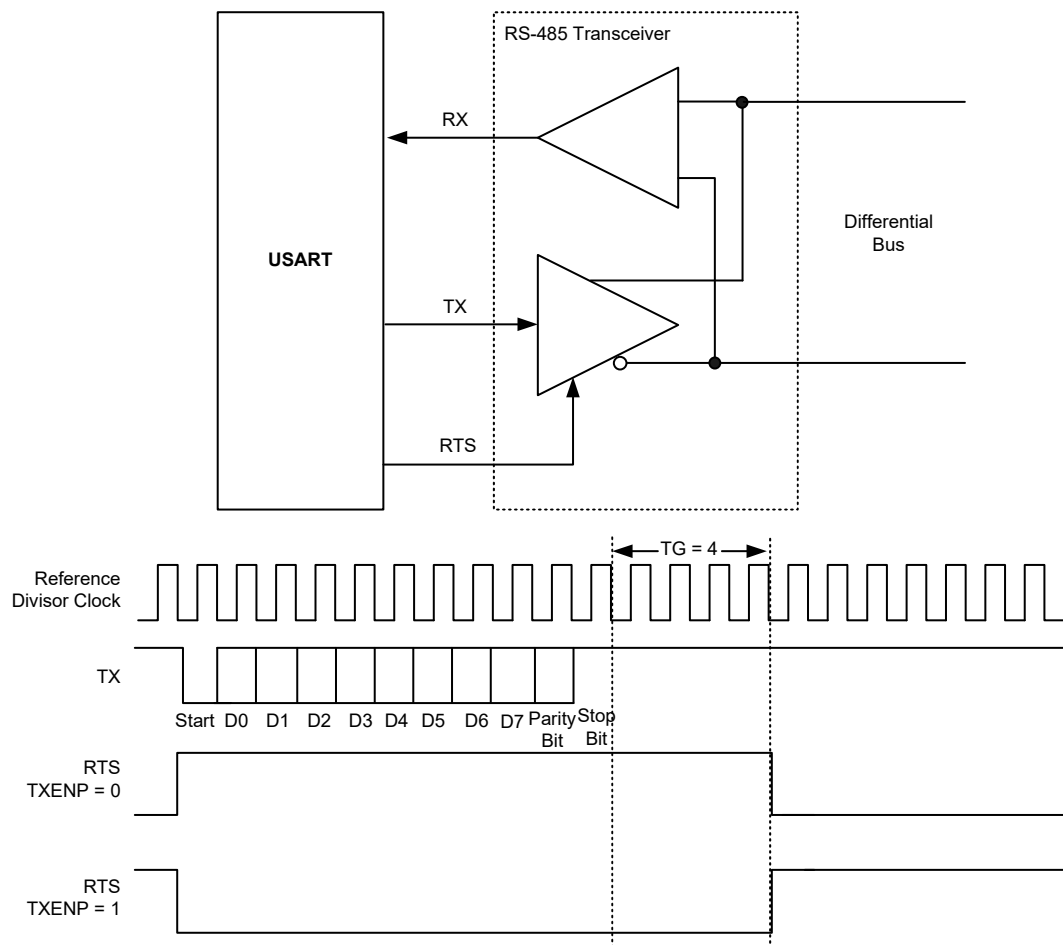
**Figure 155. USART I/O and IrDA Block Diagram**

## RS485 Mode

The RS485 mode of USART provides the data transmission on interface transmitted over a 2-wire twisted pair bus. The RS485 transceiver interprets the voltage levels of the differential signals with respect to a third common voltage. Without this common reference, the transceiver may interpret the differential signals incorrectly. This enhances the noise rejection capabilities of the RS485 interface. The USART RTS pin is used to control the external RS485 transceiver whose polarity can be selected by configuring the TXENP bit in the RS485 Control Register, named RS485CR, when the USART operates in the RS485 mode.

### RS485 Auto Direction Mode – AUD

When the RS485 mode is configured as a master transmitter, it will operate in the Auto Direction Mode, AUD. In the AUD mode the polarity of the USART RTS pin is configurable according to the TXENP bit in the RS485 Control Register in the RS485 mode. This pin can be used to control the external RS485 transceiver to enable the transmitter.



**Figure 156. RS485 Interface and Waveform**

### RS485 Normal Multi-Drop Operation Mode – NMM

When the RS485 mode is configured as an addressable slave, it will operate in the Normal Multi-drop Operation Mode, NMM. This mode is enabled when the RSNMM field is set in the RS485CR register. Regardless of the URRXEN value in the USRCR register, all the received data with a parity bit “0” will be ignored until the first address byte is detected with a parity bit “1” and then the received address byte will be stored in the RX FIFO. Once the first address data is detected and stored in the RX FIFO, the RSADD flag in the USRSIFR register will be set and generate an interrupt if the RSADDIE bit in the USRIER register is set to 1. Application software can determine whether the receiver is enabled or disabled to accept the following data by configuring the URRXEN bit. When the receiver is enabled by setting the URRXEN bit to 1, all received data will be stored in the RX FIFO. Otherwise, all received data will be ignored if the receiver is disabled by clearing the URRXEN bit to 0.

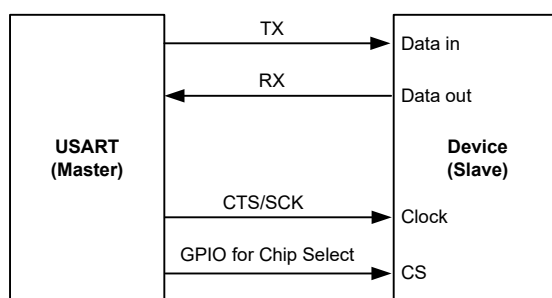
### RS485 Auto Address Detection Operation Mode – AAD

Except in the Normal Multi-drop Operation Mode, the RS485 mode can operate in the Auto Address Detection Operation Mode, AAD, when it is configured as an addressable slave. This mode is enabled by setting the RSAAD field to 1 in the RS485CR register. The receiver will detect the address frame with a parity bit “1” and then compare the received address data with the ADDMATCH field value which is a programmable 8-bit address value specified in the RS485CR register. If the address data matches the ADDMATCH value, it will be stored in the RX FIFO and the URRXEN bit will be automatically set. When the receiver is enabled, all received data will be stored in the RX FIFO until the next address frame does not match the ADDMATCH value and then the receiver will be automatically disabled. After the receiver is enabled, software can disable the receiver by clearing the URRXEN bit to ‘0’.

### Synchronous Master Mode

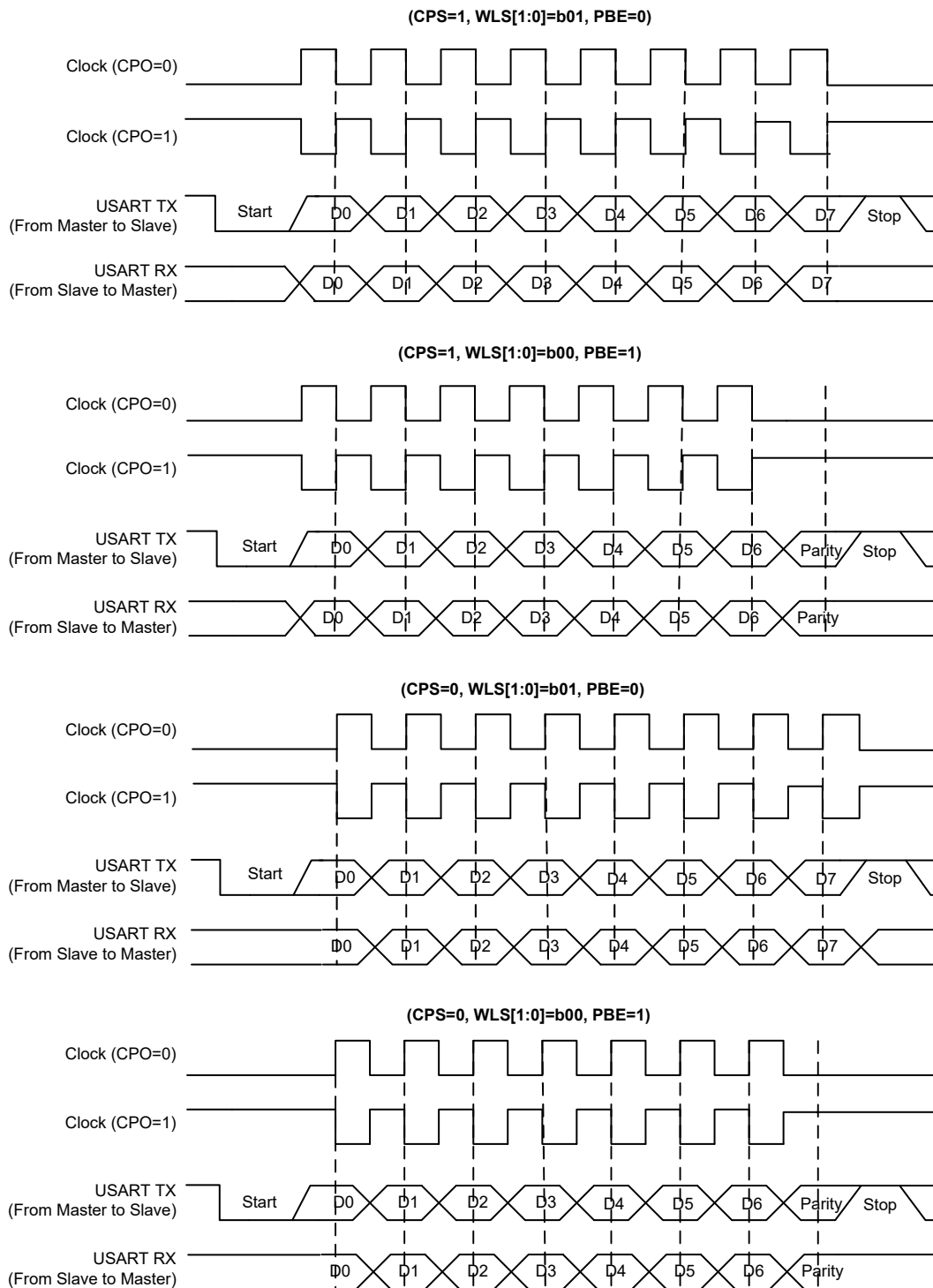
The data is transmitted in a full-duplex style in the USART Synchronous Master Mode, i.e., data transmission and reception both occur at the same time and only support master mode. The USART CTS pin is the synchronous USART transmitter clock output. In this mode, no clock pulses will be sent to the CTS pin during the start bit, parity bit and stop bit duration. The CPS bit in the Synchronous Control Register SYNCR, can be used to determine whether data is captured on the first or the second clock edge. The CPO bit in the SYNCR can be used to configure the clock polarity in the USART Synchronous Mode idle state. Detailed timing information is shown in the Figure 157.

In the USART synchronous Mode, the USART CTS/SCK clock output pin is only used to transmit the data to slave device. If the transmission data register USRDR, is written with valid data, the USART synchronous mode will automatically transmit this data with the corresponding clock output and the USART receiver will also receive data on the RX pin. Otherwise the receiver will not obtain synchronous data if no data is transmitted.



**Figure 157. USART Synchronous Transmission Example**

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



**Figure 158. 8-Bit Format USART Synchronous Waveform**

## Interrupts and Status

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

- Receive FIFO time-out interrupt: An interrupt is generated when the USART receive FIFO is not empty and does not receive a new data package during the specified time-out interval.
- Receiver line status interrupts: The interrupts are generated when the USART receiver overrun error, parity error, framing error and break events occur.
- Transmit FIFO threshold level interrupt: An interrupt is generated when the data to be transmitted in the USART Transmit FIFO is less than the specified threshold level.
- Transmit complete interrupt: An interrupt is generated when the Transmit FIFO is empty and the content of the transmit shift register (TSR) is also completely shifted.
- Receive FIFO threshold level interrupt: An interrupt is generated when the FIFO received data amount has reached the specified threshold level.

## PDMA Interface

The PDMA interface is integrated in the USART. The PDMA function can be enabled by setting the TXDMAEN or RXDMAEN bit in the USRCR register to 1 in the transmit or receive mode respectively. When the data to be transmitted in the USART Transmit FIFO is less than the TX FIFO threshold level specified by the TXTL field in the USRFCR register and the TXDMAEN bit is set to 1, the PDMA function will be activated to move data from a source location into the USART TX FIFO.

Similarly, when the received data amount in the receive FIFO is equal to the RX FIFO threshold level specified by the RXTL field in the USRFCR register and the RXDMAEN bit is set to 1, the PDMA function will be activated to move data from the USART RX FIFO to a specific destination location. For a more detailed description about the PDMA configurations, refer to the PDMA chapter.

## Register Map

The following table shows the USART registers and reset values.

**Table 55. USART Register Map**

Register	Offset	Description	Reset Value
USRDR	0x000	USART Data Register	0x0000_0000
USRCR	0x004	USART Control Register	0x0000_0000
USRFCR	0x008	USART FIFO Control Register	0x0000_0000
USRIER	0x00C	USART Interrupt Enable Register	0x0000_0000
USRSIFR	0x010	USART Status & Interrupt Flag Register	0x0000_0980
USRTPR	0x014	USART Timing Parameter Register	0x0000_0000
IrDACR	0x018	USART IrDA Control Register	0x0000_0000
RS485CR	0x01C	USART RS485 Control Register	0x0000_0000
SYNCR	0x020	USART Synchronous Control Register	0x0000_0000
USRDLR	0x024	USART Divider Latch Register	0x0000_0010
USRTSTR	0x028	USART Test Register	0x0000_0000

## Register Descriptions

### USART Data Register – USRDR

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

Offset: 0x000

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[8:0]	DB	<p>Reading data from this receiver buffer register will return the data from the receive FIFO. The receive FIFO has a capacity of up to <math>8 \times 9</math> bits. By reading this register, the USART will return a 7, 8 and 9-bit received data. The DB field bit 8 is valid for the 9-bit mode only and is fixed at 0 for the 8-bit mode. For the 7-bit mode, the DB[6:0] field contains the available bits.</p> <p>Writing data to this buffer register will load data into the Transmit FIFO. The Transmit FIFO has a capacity of up to <math>8 \times 9</math> bits. By writing to this register, the USART will send out 7, 8 or 9-bit transmitted data. The DB field bit 8 is valid for the 9-bit mode only and will be ignored for the 8-bit mode. For the 7-bit mode, the DB[6:0] field contains the available bits.</p>



## USART Control Register – USRCR

The register specifies the serial parameters such as data length, parity and stop bit for the USART. It also contains the USART enable control bits together with the USART mode and data transfer mode selection.

Offset: 0x004

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	RTS	BCB	SPE	EPE	PBE	NSB	WLS	
	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
Type/Reset	RXDMAEN	TXDMAEN	URRXEN	URTXEN	HFCEN	TRSM	MODE	
	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[15]	RTS	Request-To-Send Signal 0: Drive USART RTS pin to logic 1 1: Drive USART RTS pin to logic 0 Note that the RTS bit is used to control the USART RTS pin status when the HFCEN bit is reset. When the HFCEN bit is set, this RTS bit is read only and indicates the pin status that is controlled by hardware flow control function.
[14]	BCB	Break Control Bit When this bit is set 1, the serial data output on the USART TX pin will be forced to the Spacing State (logic 0). This bit acts only on the USART TX output pin and has no effect on the transmitter logic.
[13]	SPE	Stick Parity Enable 0: Disable stick parity 1: Stick Parity bit is transmitted This bit is only available when the PBE bit is set to 1. If both the PBE and SPE bits are set to 1 and the EPE bit is cleared to 0, the transmitted parity bit will be stuck to 1. However, when the PBE and SPE bits are set to 1 and also the EPE bit is set to 1, the transmitted parity bit will be stuck to 0.
[12]	EPE	Even Parity Enable 0: Odd number of logic 1's are transmitted or checked in the data word and parity bits 1: Even number of logic 1's are transmitted or checked in the data word and parity bits This bit is only available when the PBE bit is set to 1.
[11]	PBE	Parity Bit Enable 0: Parity bit is not generated (transmitted data) or checked (received data) during transfer 1: Parity bit is generated or checked during transfer Note: When the WLS field is set to "10" to select the 9-bit data format, writing to the PBE bit has no effect.

Bits	Field	Descriptions
[10]	NSB	Number of STOP bit 0: One STOP bit is generated in the transmitted data 1: Two STOP bits are generated when 8-bit or 9-bit word length is selected
[9:8]	WLS	Word Length Select 00: 7 bits 01: 8 bits 10: 9 bits 11: Reserved
[7]	RXDMAEN	USART RX DMA Enable 0: Disable 1: Enable
[6]	TXDMAEN	USART TX DMA Enable 0: Disable 1: Enable
[5]	URRXEN	USART RX Enable 0: Disable 1: Enable
[4]	URTXEN	USART TX Enable 0: Disable 1: Enable
[3]	HFCEN	Hardware Flow Control Function Enable 0: Disable 1: Enable
[2]	TRSM	Transfer Mode Selection This bit is used to select the data transfer protocol. 0: LSB first 1: MSB first
[1:0]	MODE	USART Mode Selection 00: Normal operation 01: IrDA 10: RS485 11: Synchronous

## USART FIFO Control Register – USRFCR

This register specifies the USART FIFO control and configurations including threshold level and reset function together with the USART FIFO status.

Offset: 0x008

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[27:24]	RXFS	<b>RX FIFO Status</b> The RXFS field shows the current number of data contained in the RX FIFO. 0000: RX FIFO is empty 0001: RX FIFO contains 1 data ... 1000: RX FIFO contains 8 data Others: Reserved
[19:16]	TXFS	<b>TX FIFO Status</b> The TXFS field shows the current number of data contained in the TX FIFO. 0000: TX FIFO is empty 0001: TX FIFO contains 1 data ... 1000: TX FIFO contains 8 data Others: Reserved
[7:6]	RXTL	<b>RX FIFO Threshold Level Setting</b> 00: 1 data 01: 2 data 10: 4 data 11: 6 data The RXTL field defines the RX FIFO trigger level.
[5:4]	TXTL	<b>TX FIFO Threshold Level Setting</b> 00: 0 data 01: 2 data 10: 4 data 11: 6 data The TXTL field determines the TX FIFO trigger level.
[1]	RXR	<b>RX FIFO Reset</b> Setting this bit will generate a reset pulse to reset the RX FIFO which will empty the RX FIFO, i.e., the RX pointer will be reset to 0 after a reset signal. This bit returns to 0 automatically after the reset pulse is generated.

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

## USART Interrupt Enable Register – USRIER

This register is used to enable the related USART interrupt function. The USART module generates interrupts to the controller when the corresponding events occur and the corresponding interrupt enable bits are set.

Offset: 0x00C

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	Reserved								
Type/Reset									
	23	22	21	20	19	18	17	16	
	Reserved								
Type/Reset									
	15	14	13	12	11	10	9	8	
	Reserved						CTSIE	RXTOIE	
Type/Reset							RW	0	RW
	7	6	5	4	3	2	1	0	
	RSADDIE	BIE	FEIE	PEIE	OEIE	TXCIE	TXDEIE	RXDRIE	
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[9]	CTSIE	CTS Clear-To-Send Interrupt Enable 0: Disable 1: Enable If this bit is set, an interrupt will be generated when the CTSC bit in the USRSIFR register is set.
[8]	RXTOIE	Receive FIFO Time-Out Interrupt Enable 0: Disable 1: Enable If this bit is set, an interrupt will be generated when the RXTOF bit in the USRSIFR register is set.
[7]	RSADDIE	RS485 Address Detection Interrupt Enable 0: Disable 1: Enable If this bit is set, an interrupt will be generated when the RSADD bit in the USRSIFR register is set.
[6]	BIE	Break Interrupt Enable 0: Disable 1: Enable If this bit is set, an interrupt will be generated when the BII bit in the USRSIFR register is set.

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

## USART Status & Interrupt Flag Register – USRSIFR

This register contains the corresponding USART status.

Offset: 0x010

Reset value: 0x0000\_0980

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	Reserved				CTSS	CTSC	RSADD	TXC	
	7	6	5	4	3	2	1	0	
Type/Reset	TXDE	RXTOF	RXDR	BII	FEI	PEI	OEI	RXDNE	
	RO	1	WC	0	RO	0	WC	0	RO
				WC	0	WC	0	WC	0

Bits	Field	Descriptions
[11]	CTSS	CTS Clear-To-Send Status 0: CTS pin is inactive 1: CTS pin is active and kept at a logic low state
[10]	CTSC	CTS Status Change Flag This bit will be set whenever the CTS input pin status has been changed and an Interrupt will be generated if the CTSIE bit in the USRIER register is set. Writing 1 to this bit clears the flag.
[9]	RSADD	RS485 Address Detection 0: Address is not detected 1: Address is detected This bit will be set to 1 when the receiver detects the address. An interrupt will be generated if the RSADDIE bit in the USRIER register is set. Writing 1 to this bit clears the flag. Note: This bit is only used in the RS485 mode by setting the MODE field in the USRCR register.
[8]	TXC	Transmit Complete 0: Either transmit FIFO (TX FIFO) or transmit shift register (TSR) is not empty 1: Both the TX FIFO and TSR register are empty When this bit is set, an interrupt will be generated if the TXCIE bit in the USRIER register is set. This bit is cleared by a write to the USRDR register with new data.
[7]	TXDE	Transmit Data FIFO Empty 0: TX FIFO level is higher than threshold 1: TX FIFO level is equal to or less than threshold The TXDE bit will be set when the transmit FIFO level is less than the transmit FIFO threshold level specified by the TXTL field in the USRFCR register. This bit will be cleared when the USRDR register is written with new data and the TX FIFO level is higher than the threshold setting.

Bits	Field	Descriptions
[6]	RXTOF	<p>Receive FIFO Time-Out Flag</p> <p>0: RX FIFO Time-Out does not occur 1: RX FIFO Time-Out occurs</p> <p>The RXTOF bit will be set if the RX FIFO is not empty and no activities have occurred in the RX FIFO during the time-out duration specified by the RXTOC field. If an RX FIFO time-out condition has occurred, this flag must be cleared before reading the RX FIFO. Writing 1 to this bit clears the flag.</p>
[5]	RXDR	<p>Receive FIFO Ready Flag</p> <p>0: RX FIFO level is less than threshold 1: RX FIFO level is equal to or higher than threshold</p> <p>The RXDR bit will be set when the FIFO received data amount has reached the threshold level specified by the RXTL field in the USRFCR register. This bit will be cleared when the data is read from the USRDR register and the RX FIFO level is less than threshold setting.</p>
[4]	BII	<p>Break Interrupt Indicator</p> <p>This bit will be set to 1 whenever the received data input is held in the “spacing state” (logic 0) for longer than a full word transmission time, which is the total time of “start bit” + data bits + “parity” + “stop bits” duration. Writing 1 to this bit clears the flag.</p>
[3]	FEI	<p>Framing Error Indicator</p> <p>This bit will be set 1 whenever the next character to be read in the RX FIFO does not have a valid “stop bit”, which means the stop bit following the last data bit or parity bit is detected as logic 0. Writing 1 to this bit clears the flag.</p>
[2]	PEI	<p>Parity Error Indicator</p> <p>This bit will be set to 1 whenever the received character does not have a valid “parity bit”. Writing 1 to this bit clears the flag.</p>
[1]	OEI	<p>Overrun Error Indicator</p> <p>An overrun error will occur only after the RX FIFO is full and when the next character has been completely received in the RX shift register. The character in the shift register will be overwritten if a new character is received in the RX shift register after an overrun event occurs, but the data in the RX FIFO will not be overwritten. The OEI bit is used to indicate the overrun event as soon as it happens. Writing 1 to this bit clears the flag.</p>
[0]	RXDNE	<p>RX FIFO Data Not Empty</p> <p>0: RX FIFO is empty 1: RX FIFO contains at least 1 received data word</p>

## USART Timing Parameter Register – USRTPR

This register contains the USART timing parameters including the transmitter time guard parameters and the receive FIFO time-out value together with the RX FIFO time-out interrupt enable control.

Offset: 0x014

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[15:8]	TG	Transmitter Time Guard The transmitter time guard counter is driven by the baud rate clock. When the TX FIFO transmits data, the counter is reset and then starts to count after a word transmission has completed. Only when the counter content is equal to the TG value, are further word transmission transactions allowed.
[7]	RXTOEN	Receive FIFO Time-Out Counter Enable 0: RX FIFO Time-Out Counter is disabled 1: RX FIFO Time-Out Counter is enabled
[6:0]	RXTOC	Receive FIFO Time-Out Counter Compare Value The RX FIFO time-out counter is driven by the baud rate clock. When the RX FIFO receives new data, the counter is reset and then starts to count. Once the time-out counter content is equal to the time-out counter compare value RXTOC, an RX FIFO time-out interrupt, RXTOI, will be generated if the RXTOIE bit in the USRIER register is set to 1. New received data or the empty RX FIFO after being read will clear the RX FIFO time-out counter.



## USART IrDA Control Register – IrDACR

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

Offset: 0x018

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[15:8]	IrDAPSC	<p>IrDA Prescaler value</p> <p>This field contains the 8-bit debounce prescaler value.</p> <p>The debounce count-down counter is driven by the USART clock, named as CK_USART. The counting period is specified by the IrDAPSC field. The IrDAPSC field must be set to a value equal to or greater than 0x01 for normal debounce counter operation. If the pulse width is less than the duration specified by the IrDAPSC field, the pulse will be considered as glitch noise and discarded.</p> <p>00000000: Reserved – can not be used            00000001: CK_USART clock divided by 1            00000010: CK_USART clock divided by 2            00000011: CK_USART clock divided by 3            ...</p>
[5]	RXINV	<p>RX Signal Inverse Control</p> <p>0: No inversion            1: RX input signal is inverted</p>
[4]	TXINV	<p>TX Signal Inverse Control</p> <p>0: No inversion            1: TX output signal is inverted</p>
[3]	LB	<p>IrDA Loop Back Mode</p> <p>0: Disable IrDA loop back mode            1: Enable IrDA loop back mode for self-testing</p>
[2]	TXSEL	<p>Transmit Select</p> <p>0: Enable IrDA receiver            1: Enable IrDA transmitter</p>
[1]	IrDALP	<p>IrDA Low-Power Mode</p> <p>Selects the IrDA operation mode.</p> <p>0: Normal mode            1: IrDA low-power mode</p>

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

## USART RS485 Control Register – RS485CR

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

Offset: 0x01C

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[15:8]	ADDMATCH	RS485 Auto Address Match value The field contains the address match value for the RS485 auto address detection operation mode.
[2]	RSAAD	RS485 Auto Address Detection Operation Mode Control 0: Disable 1: Enable
[1]	RSNMM	RS485 Normal Multi-drop Operation Mode Control 0: Disable 1: Enable
[0]	TXENP	USART RTS/TXE Pin Polarity 0: RTS/TXE is active high in the RS485 transmission mode 1: RTS/TXE is active low in the RS485 transmission mode

## USART Synchronous Control Register – SYNCR

This register is used to control the USART synchronous mode.

Offset: 0x020

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[3]	CPO	<p>Clock Polarity</p> <p>0: CTS/SCK pin idle state is low</p> <p>1: CTS/SCK pin idle state is high</p> <p>This bit can be used to select the polarity of the clock output on the USART CTS/SCK pin in the synchronous mode. Works in conjunction with the CPS bit to specify the desired clock idle state.</p>
[2]	CPS	<p>Clock Phase</p> <p>0: Data is captured on the first clock edge</p> <p>1: Data is captured on the second clock edge</p> <p>This bit can be used to select the phase of the clock output on the USART CTS/SCK pin in the synchronous mode. Works in conjunction with the CPO bit to determine the data capture edge.</p>
[0]	CLKEN	<p>Clock Enable</p> <p>0: CTS/SCK pin is disabled</p> <p>1: CTS/SCK pin is enabled</p> <p>Enable/disable the USART CTS/SCK pin.</p>

## USART Divider Latch Register – USRDLR

The register is used to determine the USART clock divided ratio to generate the appropriate baud rate.

Offset: 0x024

Reset value: 0x0000\_0010

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

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

## USART Test Register – USRTSTR

This register controls the USART debug mode.

Offset: 0x028

Reset value: 0x0000\_0000

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

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

# 22 Universal Asynchronous Receiver Transmitter (UART)

## Introduction

The Universal Asynchronous Receiver Transceiver, UART, provides a flexible full duplex data exchange using asynchronous transfer. The UART is used to translate data between parallel and serial interfaces, and is also commonly used for RS232 standard communication. The UART peripheral function supports a variety of interrupts.

The UART module includes a transmit data register TDR and transmit shift register TSR, and a receive data register RDR and receive shift register RSR. Software can detect a UART error status by reading the UART Status & Interrupt Flag Register, URSIFR. The status includes the condition of the transfer operations as well as several error conditions resulting from Parity, Overrun, Framing and Break events.

The UART includes a programmable baud rate generator which is capable of dividing the UART clock CK\_APB (CK\_UART) to produce a baud rate clock for the UART transmitter and receiver.

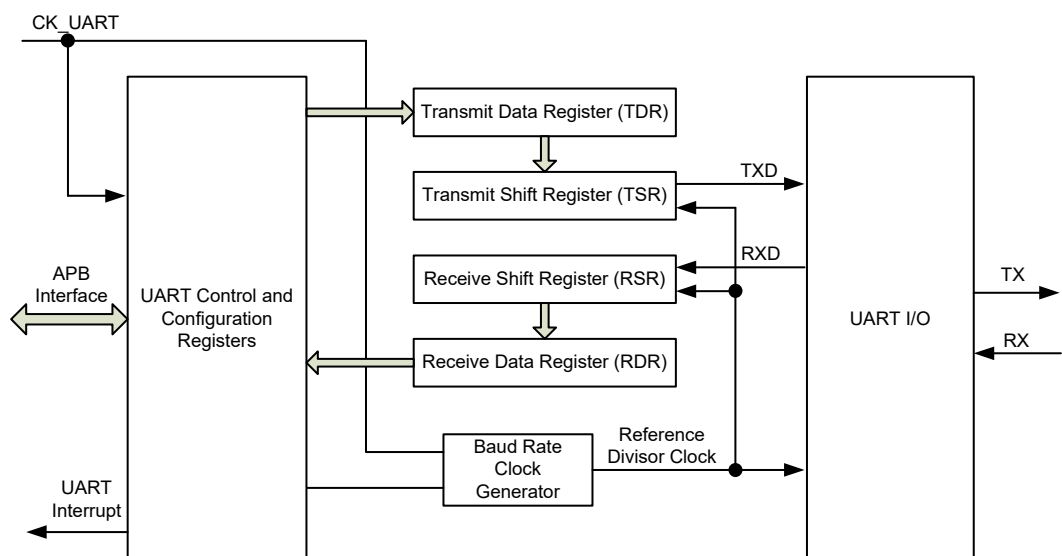


Figure 159. UART Block Diagram

## Features

- Supports asynchronous serial communication modes
- Full Duplex Communication Capability
- Programming baud rate clock frequency up to ( $f_{PCLK}/16$ ) MHz
- Fully programmable serial communication functions including:
  - Word length: 7, 8 or 9-bit character
  - Parity: Even, odd or no-parity bit generation and detection
  - Stop bit: 1 or 2 stop bits generation
  - Bit order: LSB-first or MSB-first transfer
- Error detection: Parity, overrun, and frame error
- Supports PDMA Interface

## Functional Descriptions

### Serial Data Format

The UART module performs a parallel-to-serial conversion on data that is written to the transmit data register and then sends the data with the following format: Start bit, 7 ~ 9 LSB/MSB first data bits, optional Parity bit and finally 1 ~ 2 Stop bits. The Start bit has the opposite polarity of the data line idle state. The Stop bit is the same as the data line idle state and provides a delay before the next start situation. Both the Start and Stop bits are used for data synchronization during the asynchronous data transmission.

The UART module also performs a serial-to-parallel conversion on the data that is read from the receive data register. It will first check the Parity bit and will then look for a Stop bit. If the Stop bit is not found, the UART module will consider the entire word transmission as failed and respond with a Framing Error.

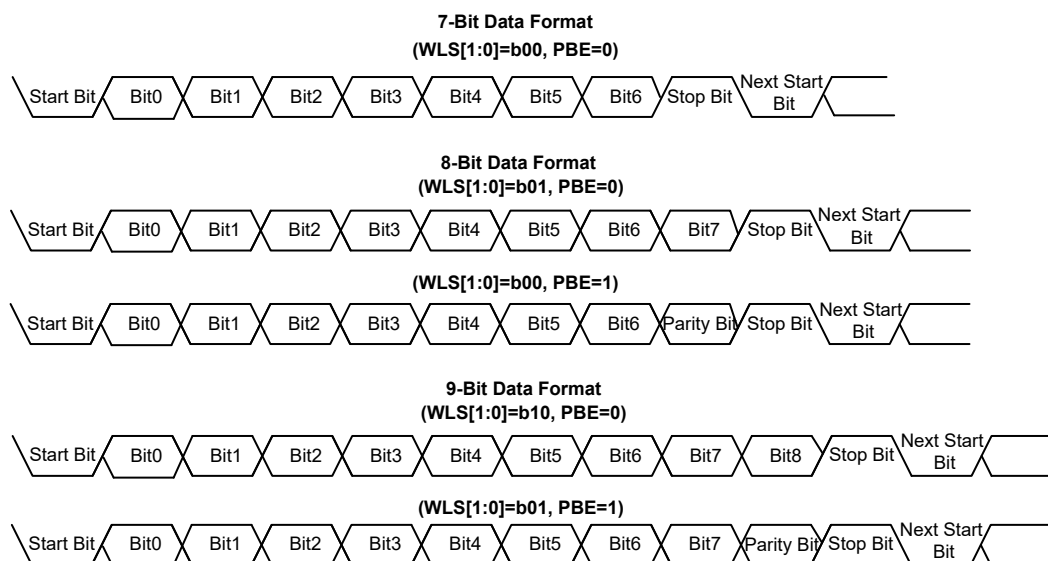


Figure 160. UART Serial Data Format

## Baud Rate Generation

The baud rate for the UART receiver and transmitter are both set with the same values. The baud rate divisor, BRD, has the following relationship with the UART clock which is known as CK\_UART.

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

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

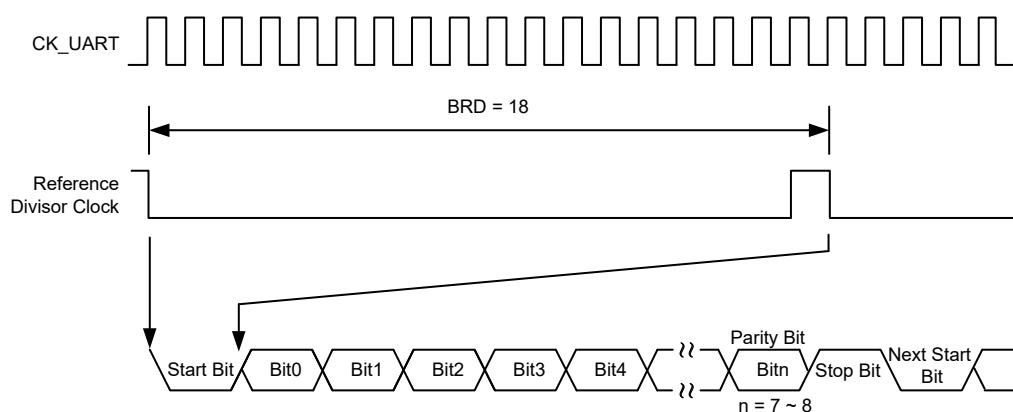


Figure 161. UART Clock CK\_UART and Data Frame Timing

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

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



**Table 57. Baud Rate Deviation Error Calculation – CK\_UART = 48 MHz**

Baud Rate		CK_UART = 48 MHz		
No.	Kbps	Actual	BRD	Deviation Error Rate
1	2.4	2.4	20000	0.00%
2	9.6	9.6	5000	0.00%
3	19.2	19.2	2500	0.00%
4	57.6	57.6	833	0.04%
5	115.2	115.1	417	-0.08%
6	230.4	230.8	208	0.16%
7	460.8	461.5	104	0.16%
8	921.6	923.1	52	0.16%
9	2250	2285.7	21	1.59%
10	3000	3000.0	16	0.00%

**Table 58. Baud Rate Deviation Error Calculation – CK\_UART = 60 MHz**

Baud Rate		CK_UART = 60 MHz		
No.	Kbps	Actual	BRD	Deviation Error Rate
1	2.4	2.4	25000	0.00%
2	9.6	9.6	6250	0.00%
3	19.2	19.2	3125	0.00%
4	57.6	57.6	1042	-0.03%
5	115.2	115.2	521	-0.03%
6	230.4	230.8	260	0.16%
7	460.8	461.5	130	0.16%
8	921.6	923.1	65	0.16%
9	2250	2222.2	27	-1.23%
10	3000	3000.0	20	0.00%

## Interrupts and Status

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

- Receiver line status interrupts: The interrupts are generated when the UART receiver overrun error, parity error, framing error and break event occur.
- Transmit data register empty interrupt: An interrupt is generated when the content of the transmit data register is transferred to the transmit shift register (TSR).
- Transmit complete interrupt: An interrupt is generated when the transmit data register (TDR) is empty and the content of the transmit shift register (TSR) is also completely shifted.
- Receive data ready interrupt: An interrupt is generated when the content of the receive shift register (RSR) has been transferred to the URDR register and is ready to read.

## PDMA Interface

The PDMA interface is integrated in the UART. The PDMA function can be enabled by setting the TXDMAEN or RXDMAEN bit in the URCR register to 1 in the transmit or receive mode respectively. When the UART transmit data register (TDR) is empty and the TXDMAEN bit is set to 1, the PDMA function will be activated to move data from a source location into the UART transmit data register (TDR).

Similarly, when the received data has been in the UART receive data register (RDR) and the RXDMAEN bit is set to 1, the PDMA function will be activated to move data from the UART receive data register (RDR) to a specific destination location. For a more detailed description about the PDMA configurations, refer to the PDMA chapter.

## Register Map

The following table shows the UART registers and reset values.

**Table 59. UART Register Map**

Register	Offset	Description	Reset Value
URDR	0x000	UART Data Register	0x0000_0000
URCR	0x004	UART Control Register	0x0000_0000
URIER	0x00C	UART Interrupt Enable Register	0x0000_0000
URSIFR	0x010	UART Status & Interrupt Flag Register	0x0000_0180
URDLR	0x024	UART Divider Latch Register	0x0000_0010
URTSTR	0x028	UART Test Register	0x0000_0000

## Register Descriptions

### UART Data Register – URDR

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

Offset: 0x000

Reset value: 0x0000\_0000

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

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

## UART Control Register – URCR

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

Offset: 0x004

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24			
	Reserved										
Type/Reset											
	23	22	21	20	19	18	17	16			
	Reserved										
Type/Reset											
	15	14	13	12	11	10	9	8			
	Reserved	BCB	SPE	EPE	PBE	NSB	WLS				
Type/Reset		RW	0	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0			
	RXDMAEN	TXDMAEN	URRXEN	URTXEN	Reserved	TRSM	Reserved				
Type/Reset	RW	0	RW	0	RW	0	RW	0			

Bits	Field	Descriptions
[14]	BCB	Break Control Bit When this bit is set 1, the serial data output on the UART TX pin will be forced to the Spacing State (logic 0). This bit acts only on the UART TX output pin and has no effect on the transmitter logic.
[13]	SPE	Stick Parity Enable 0: Disable stick parity 1: Stick Parity bit is transmitted This bit is only available when the PBE bit is set to 1. If both the PBE and SPE bits are set to 1 and the EPE bit is cleared to 0, the transmitted parity bit will be stuck to 1. However, when the PBE and SPE bits are set to 1 and also the EPE bit is set to 1, the transmitted parity bit will be stuck to 0.
[12]	EPE	Even Parity Enable 0: Odd number of logic 1's are transmitted or checked in the data word and parity bits 1: Even number of logic 1's are transmitted or checked in the data word and parity bits This bit is only available when the PBE bit is set to 1.

Bits	Field	Descriptions
[11]	PBE	Parity Bit Enable 0: Parity bit is not generated (transmitted data) and checked (receive data) during transfer 1: Parity bit is generated and checked during transfer Note: When the WLS field is set to "10" to select the 9-bit data format, writing to the PBE bit has no effect.
[10]	NSB	Number of STOP bit 0: One STOP bit is generated in the transmitted data 1: Two STOP bits are generated when 8-bit or 9-bit word length is selected
[9:8]	WLS	Word Length Select 00: 7 bits 01: 8 bits 10: 9 bits 11: Reserved
[7]	RXDMAEN	UART RX DMA Enable 0: Disable 1: Enable
[6]	TXDMAEN	UART TX DMA Enable 0: Disable 1: Enable
[5]	URRXEN	UART RX Enable 0: Disable 1: Enable
[4]	URTXEN	UART TX Enable 0: Disable 1: Enable
[2]	TRSM	Transfer Mode Selection This bit is used to select the data transfer protocol. 0: LSB first 1: MSB first

## UART Interrupt Enable Register – URIER

This register is used to enable the related UART interrupt function. The UART module generates interrupts to the controller when the corresponding events occur and the corresponding interrupt enable bits are set.

Offset: 0x00C

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
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	BIE	FEIE	PEIE	OEIE	TXCIE	TXDEIE	RXDRIE
		RW	0	RW	0	RW	0	RW
								0

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

## UART Status & Interrupt Flag Register – URSIFR

This register contains the corresponding UART status.

Offset: 0x010

Reset value: 0x0000\_0180

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

Bits	Field	Descriptions
[8]	TXC	Transmit Complete 0: Either the transmit data register (TDR) or transmit shift register (TSR) is not empty 1: Both the transmit data register (TDR) and transmit shift register (TSR) are empty When this bit is set, an interrupt will be generated if the TXCIE bit in the URIER register is set to 1. This bit is cleared by a write to the URDR register with new data.
[7]	TXDE	Transmit Data Register Empty 0: Transmit data register is not empty 1: Transmit data register is empty The TXDE bit is set by hardware when the content of the transmit data register is transferred to the transmit shift register (TSR). An interrupt will be generated if the TXEIE bit in the URIER register is set to 1. This bit is cleared by a write to the URDR register with new data.
[5]	RXDR	RX Data Ready 0: Receive data register is empty 1: The received data in receive data register is ready to read This bit is set by hardware when the content of the receive shift register (RSR) has been transferred to the URDR register. An interrupt will be generated if the RXDRIE bit in the URIER register is set to 1. It is cleared by a read to the URDR register.
[4]	BII	Break Interrupt Indicator This bit is set to 1 whenever the received data input is held in the “spacing state” (logic 0) for longer than a full character transmission time, which is the total time of “start bit” + data bits + “parity” + “stop bits” duration. Writing 1 to this bit clears the flag.
[3]	FEI	Framing Error Indicator This bit is set 1 whenever the received character does not have a valid stop bit, which means, the stop bit following the last data bit or parity bit is detected as logic 0. Writing 1 to this bit clears the flag.

Bits	Field	Descriptions
[2]	PEI	Parity Error Indicator This bit is set to 1 whenever the received character does not have a valid parity bit. Writing 1 to this bit clears the flag.
[1]	OEI	Overrun Error Indicator An overrun error will occur only after the receive data register is full and when the next character has been completely received in the receive shift register. The character in the receive shift register will be overwritten when a new character is received in the receive shift register after an overrun event occurs, but the data in the receive shift register will not be transferred to the receive data register. The OEI bit is used to indicate event as soon as it happens. Writing 1 to this bit clears the flag.

### UART Divider Latch Register – URDLR

The register is used to determine the UART clock divided ratio to generate the appropriate baud rate.

Offset: 0x024

Reset value: 0x0000\_0010

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

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

## UART Test Register – URTSTR

This register controls the UART debug mode.

Offset: 0x028

Reset value: 0x0000\_0000

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

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

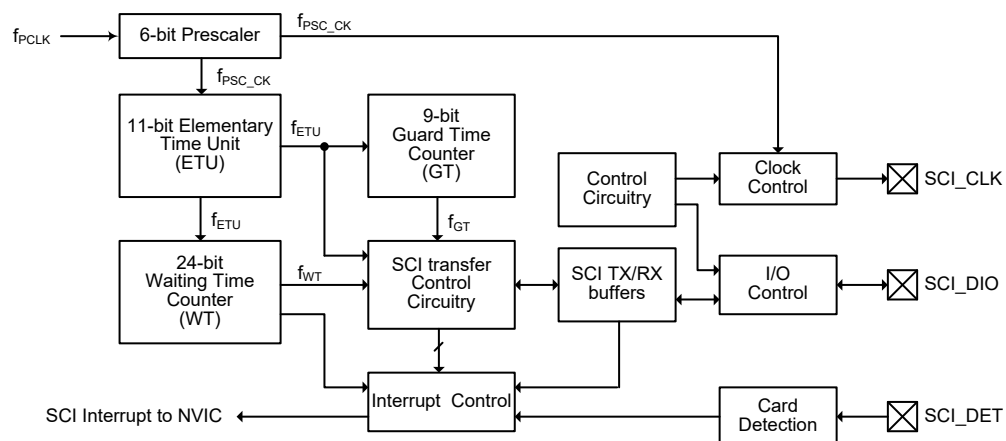


# 23 Smart Card Interface (SCI)

## 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 162. SCI Block Diagram**

## Features

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

## Functional Descriptions

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

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

### Elementary Time Unit Counter

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

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

where:

- $t_{\text{etu}}$  is the nominal duration of the data bit on the signal SCI\_DIO provided to the card by the interface
- $D_i$  is the bit-rate adjustment factor
- $F_i$  is the clock rate conversion factor
- $f$  is the frequency value of the clock signal SCI\_CLK provided to the card by the interface

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

**Table 60. DI Field Based  $D_i$  Encoded Decimal Values**

DI field	0001	0010	0011	0100	0101	0110	0111	1000	1001
$D_i$ (decimal)	1	2	4	8	16	32	64	12	20

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

**Table 61. FI Field Based  $F_i$  Encoded Decimal Values**

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

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

**Table 62. Possible ETU Values Obtained with the  $F_i/D_i$  Ratio**

$F_i/D_i$	372	558	744	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
64	5.81	8.72	11.63	17.44	23.25	29.06	8	12	16	24	32
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.

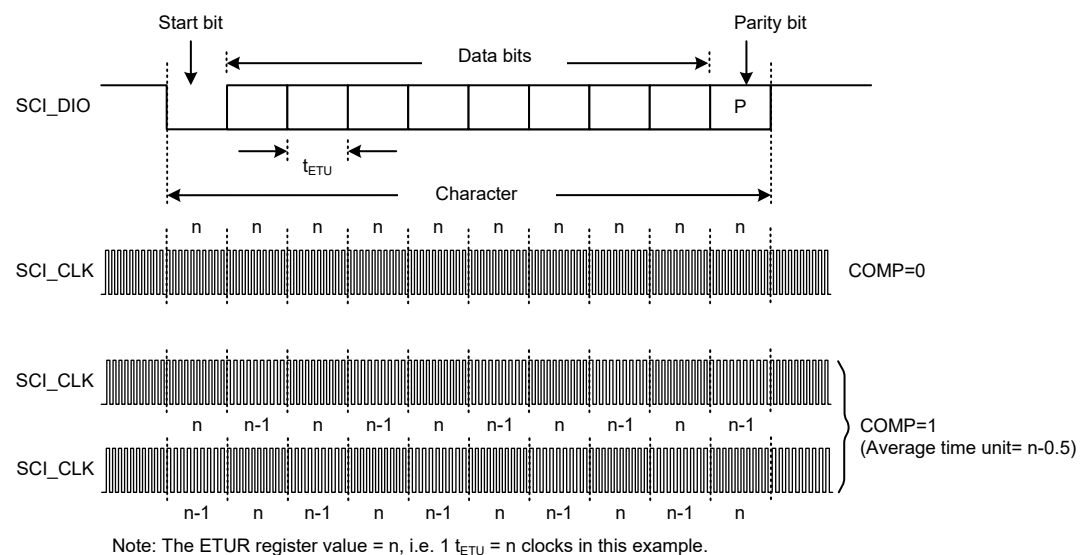
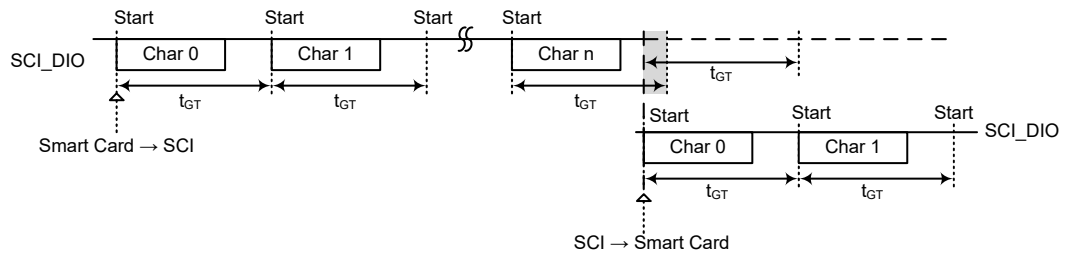


Figure 163. Character Frame and Compensation Mode

### 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 164. Guard Time Duration**

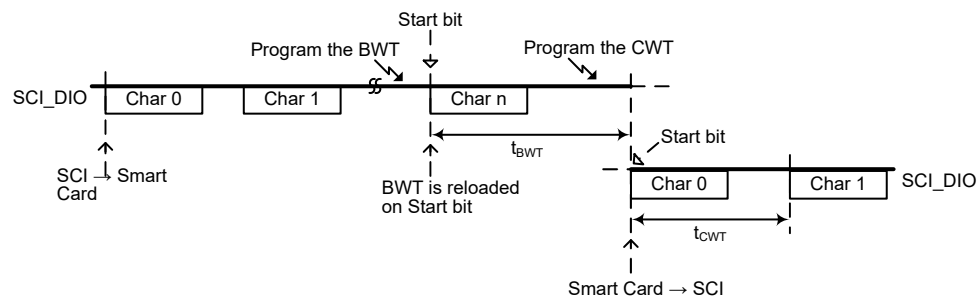
## Waiting Time Counter

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 165. 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.

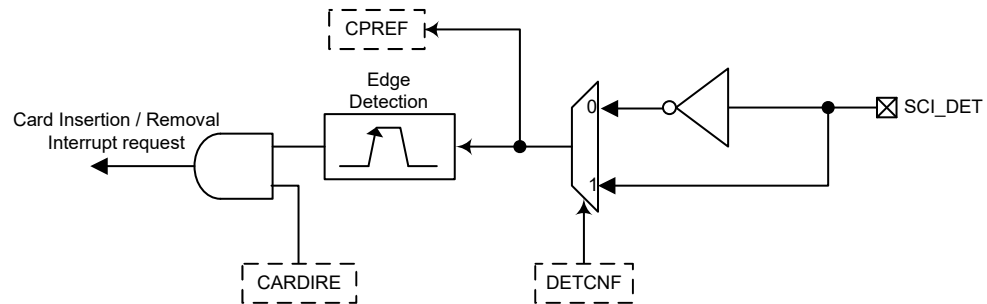
The SCI serial data pin, named SCI\_DIO, can be controlled by the SCI hardware circuitry or the software control bits depending upon whether the SCI is operated in the SCI Transfer Mode or in the Manual Mode. The mode selection is determined by the SCIM bit in the CR register. The SCI\_DIO pin status 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 166. 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 remain zero.

### 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.

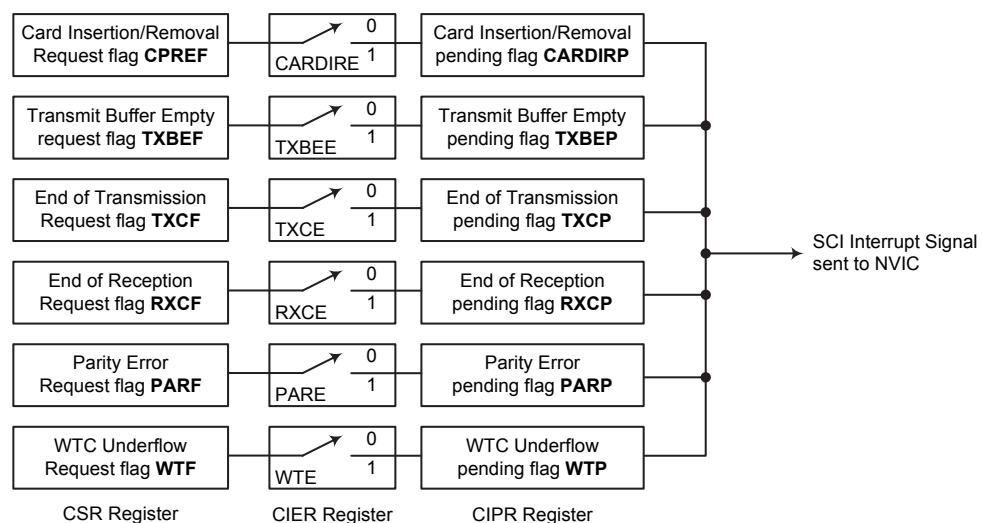


## Interrupt Generator

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 167. 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 63. SCI Register Map**

Register	Offset	Description	Reset Value
CR	0x000	SCI Control Register	0x0000_0000
SR	0x004	SCI Status Register	0x0000_0080
CCR	0x008	SCI Contact Control Register	0x0000_0008
ETUR	0x00C	SCI Elementary Time Unit Register	0x0000_0174
GTR	0x010	SCI Guard Time Register	0x0000_000C
WTR	0x014	SCI Waiting Time Register	0x0000_2580
IER	0x018	SCI Interrupt Enable Register	0x0000_0000
IPR	0x01C	SCI Interrupt Pending Register	0x0000_0000
TXB	0x020	SCI Transmit Buffer	0x0000_0000
RXB	0x024	SCI Receive Buffer	0x0000_0000
PSCR	0x028	SCI Prescaler Register	0x0000_0000

## Register Descriptions

### SCI Control Register – CR

This register contains the SCI control bits.

Offset: 0x000

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions															
[9]	RXDMA	SCI reception DMA request enable 0: SCI reception DMA request is disabled 1: SCI reception DMA request is enabled															
[8]	TXDMA	SCI transmission DMA request enable control 0: SCI transmission DMA request is disabled 1: SCI transmission DMA request is enabled															
[6]	DETCNF	Card switch type selection 0: Switch is normally opened if no card is present 1: Switch is normally closed if no card is present <table border="1"> <thead> <tr> <th>DETCNF</th><th>SCI_DET pin</th><th>STATUS</th></tr> </thead> <tbody> <tr> <td>0</td><td>1</td><td>No card insert</td></tr> <tr> <td>0</td><td>0</td><td>Card insert</td></tr> <tr> <td>1</td><td>1</td><td>Card insert</td></tr> <tr> <td>1</td><td>0</td><td>No card insert</td></tr> </tbody> </table> <p>This bit is set and cleared by the application program to configure the card detector switch type.</p>	DETCNF	SCI_DET pin	STATUS	0	1	No card insert	0	0	Card insert	1	1	Card insert	1	0	No card insert
DETCNF	SCI_DET pin	STATUS															
0	1	No card insert															
0	0	Card insert															
1	1	Card insert															
1	0	No card insert															
[5]	ENSCI	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
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	TXBEF	CPREF	reserved		WTF	TXCF	RXCF	PARF
	RO	1 RO	0		RO	0 W0C	0 RO	0 W0C

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	Parity Error Request Flag. 0: No parity error occurs 1: Parity error has occurred 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.

### SCI Contact Control Register – CCR

This register specifies the SCI pin setting and clock selection.

Offset: 0x008

Reset value: 0x0000\_0008

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

Bits	Field	Descriptions
[7]	CLKSEL	Card Clock Selection 0: The CCLK bit content is present on the external SCI_CLK pin 1: The clock output on the external SCI_CLK pin is sourced from the f <sub>PSC_CLK</sub> clock This bit is used to select the external SCI_CLK pin clock source. It is set and cleared by the application program. It is recommended that to activate the clock at a known level a certain value should be first programmed into the CCLK bit before the CLKSEL bit is switched from 1 to 0.
[3]	CDIO	SCI_DIO pin control 0: SCI_DIO pin is logic level 0 1: SCI_DIO pin in open-drain condition This bit is available only when the SCIM bit in the CR register is cleared to 0 to configure the SCI to operate in the Manual Transfer Mode. It is set and cleared by application program to control the external SCI_DIO pin status in the Manual Mode. Reading this bit will return the present status of the SCI_DIO pin.

Bits	Field	Descriptions
[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 when the SCI operates 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

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

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

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

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



## SCI Waiting Time Register – WTR

This register specifies the waiting time value obtained from the Answer-to-Reset packet described in the Waiting Time Counter section.

Offset: 0x014

Reset value: 0x0000\_2580

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

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.

Bits	Field	Descriptions
[1]	RXCE	Character Reception Completion interrupt enable control 0: Disabled 1: Enabled This bit is set and cleared by the application program and is used to control the Character Reception Completion interrupt. If this bit is set to 1, the Character Reception Completion interrupt will be generated at the end of the character reception.
[0]	PARE	Parity Error interrupt enable control 0: Disabled 1: Enabled This bit is set and cleared by the application program and is used to control the parity error interrupt. If this bit is set to 1, the Parity Error interrupt will be generated when a parity error occurs.

### SCI Interrupt Pending Register – IPR

This register contains the interrupt pending flags for all of the interrupt events in the SCI. These pending flags can be masked by the corresponding interrupt enable control bits.

Offset: 0x01C

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
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	TXBEP	CARDIRP	Reserved		WTP	TXCP	RXCP	PARP
	RC	0	RC	0	RC	0	RC	0

Bits	Field	Descriptions
[7]	TXBEP	Transmit Buffer Empty 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. 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.

Bits	Field	Descriptions
[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>
[2]	TXCP	<p>Character Transmission Completion interrupt pending flag</p> <p>0: No interrupt pending 1: Interrupt pending</p> <p>This bit is set by hardware and cleared by a read access to this register using the application program. It is used to indicate if there is a Character Transmission Completion interrupt pending or not. If a character has been transmitted and the related interrupt enable control bit is set to 1, this bit will be set to 1 to indicate that the character transmission completion interrupt is pending.</p>
[1]	RXCP	<p>Character Reception Completion interrupt pending flag</p> <p>0: No interrupt pending 1: Interrupt pending</p> <p>This bit is set by hardware and cleared by a read access to this register using the application program. It is used to indicate if there is a Character Reception Completion interrupt pending or not. If a character has been received and the relevant interrupt enable control bit is set to 1, this bit will be set to 1 to indicate that the character reception completion interrupt is pending.</p>
[0]	PARP	<p>Parity Error interrupt pending flag</p> <p>0: No interrupt pending 1: Interrupt pending</p> <p>This bit is set by hardware and cleared by a read access to this register using the application program. It is used to indicate if there is a Parity Error interrupt pending or not. If the parity error occurs and its interrupt enable control bit is set to 1, this bit will be set to 1 to indicate that the parity error interrupt is pending.</p>

## SCI Transmit Buffer – TXB

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

Offset: 0x020

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
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	TB							
	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

	31	30	29	28	27	26	25	24
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	RB							
	RO	0	RO	0	RO	0	RO	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

	31	30	29	28	27	26	25	24
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		PSC					
			RW	0	RW	0	RW	0
					RW	0	RW	0
						RW	0	RW
							RW	0

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.

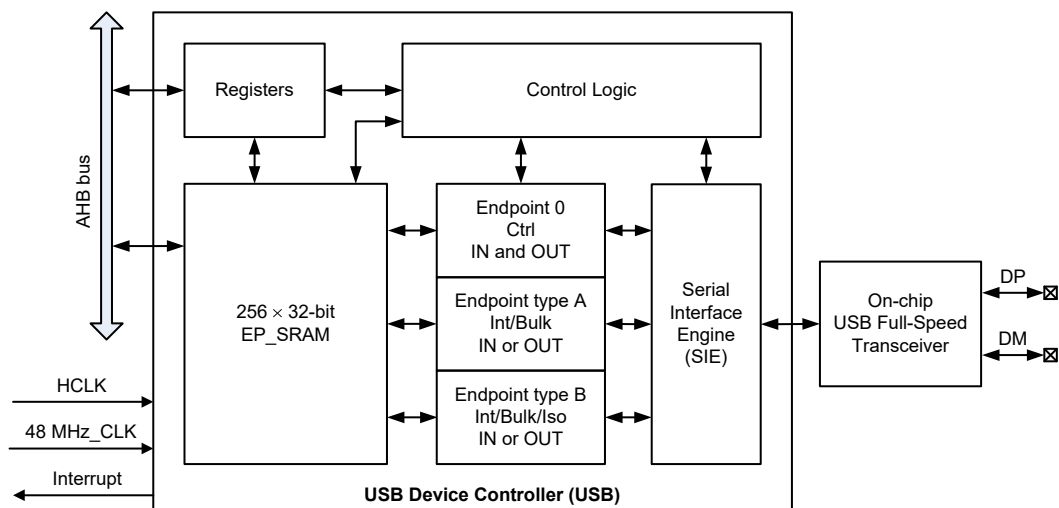


Figure 168. USB Block Diagram

## Features

- Complies with USB 2.0 full-speed (12 Mbps) specification
- Fully integrated USB full-speed transceiver
- 1 control endpoint (EP0) for control transfer
- 3 single-buffered endpoints (EP1 ~ EP3) for bulk and interrupt transfer
- 4 double-buffered endpoints (EP4 ~ EP7) for bulk, interrupt and isochronous transfer
- 1,024 bytes EP\_SRAM used as endpoint data buffers

## Functional Descriptions

### Endpoints

The USB Endpoint 0 is the only bidirectional endpoint dedicated to USB control transfer. The device also contains seven unidirectional endpoints for other USB transfer types. There are three endpoints (EP1 ~ EP3) which support a single buffering function which is used for Bulk and Interrupt IN or OUT data transfer. There are four other endpoints (EP4 ~ EP7) which support single or double buffering functions for Bulk, Interrupt and Isochronous IN or OUT data transfer. The address of the seven unidirectional endpoints (EP1 ~ EP7) can be configured by the application software. The following table lists the endpoint characteristics.

**Table 64. Endpoint Characteristics**

Endpoint Number	Number Address	Transfer Type	Direction	Buffer Type
0	Fixed	Control	IN and OUT	Single buffering
1 ~ 3	Configurable	Interrupt/Bulk	IN or OUT	Single buffering
4 ~ 7	Configurable	Interrupt/Bulk/Isochronous	IN or OUT	Single or Double buffering

### EP\_SRAM

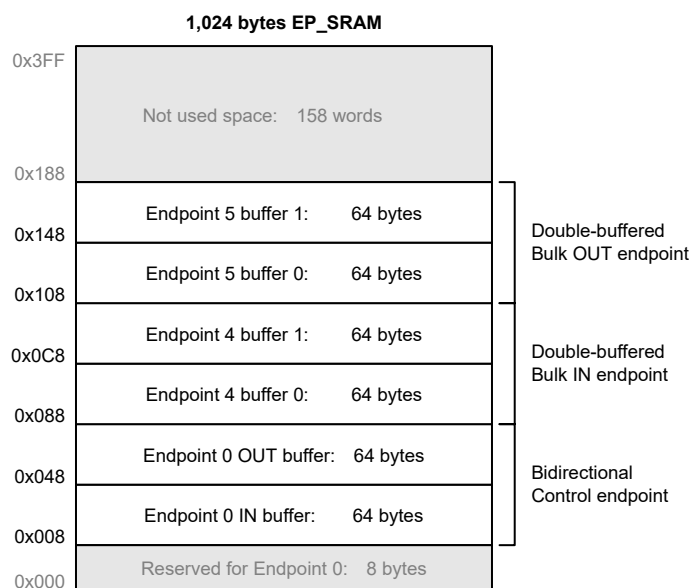
The USB controller contains a dedicated memory space, EP\_SRAM, which is used for the USB endpoint buffers. The EP\_SRAM, which is connected to the APB bus, can be accessed by the MCU and PDMA. The EP\_SRAM base address is 0x400A\_A000 with an offset which ranges from 0x000 to 0x3FF. The EP\_SRAM first two words are reserved for Endpoint 0 to temporarily store the 8-byte SETUP data. Therefore the valid start address of the endpoint buffer should start from 0x400A\_0x008 and align to a 4-byte boundary. Each endpoint buffer size is programmable. The following table lists the maximum USB endpoint buffer size which is compliant with USB 2.0 full-speed device specification.

**Table 65. USB Data Types and Buffer Size**

Transfer Type	Direction	Supported Buffer Size	Bandwidth	CRC	Retrying
Control	Bidirectional	8, 16, 32, 64 bytes	Not guaranteed	Yes	Automatic
Bulk	Unidirectional	8, 16, 32, 64 bytes	Not guaranteed	Yes	Yes
Interrupt	Unidirectional	≤ 64 bytes	Not guaranteed	Yes	Yes
Isochronous	Unidirectional	< 512 bytes	Guaranteed	Yes	No

In the following endpoint buffer allocation example, the Endpoint “4” is configured as a double-buffered Bulk IN endpoint while the Endpoint “5” is configured as a double-buffered Bulk OUT endpoint. Each endpoint buffer size is set to 64 bytes.





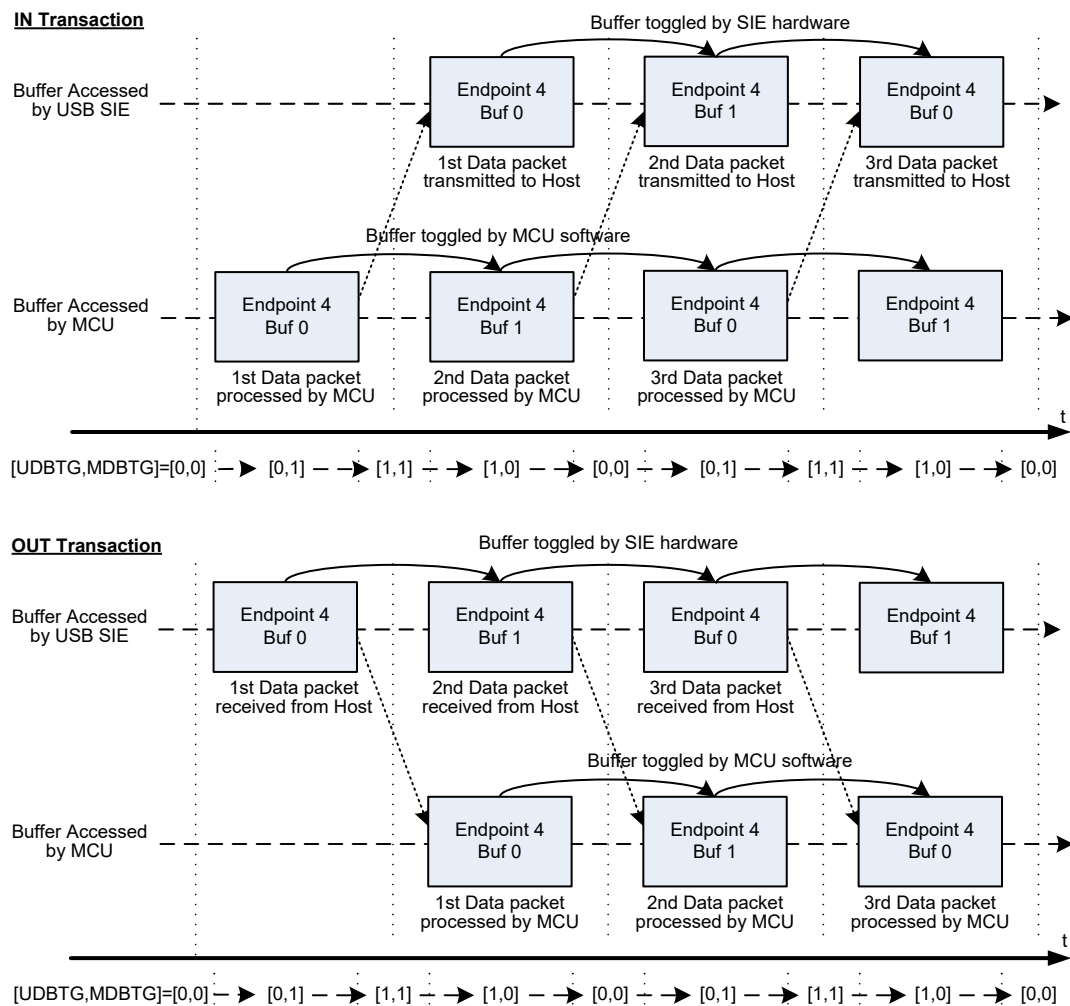
**Figure 169. Endpoint Buffer Allocation Example**

## Serial Interface Engine – SIE

The Serial Interface Engine, SIE, which is connected to the USB full-speed transceiver and internal USB control circuitry, provides a temporal buffer for the transmitted and received data. The SIE also decodes the SE0 signal, SE1 signal, J-state, K-state, USB RESET event and End of Packet event signals, EOP, when the USB module receives data, transmits data or transmits the resume signal for remote control. The SIE detects the number of SOF packets and generates the SOF interrupt signal to the USB control circuitry which includes data format conversion from parallel to serial or serial to parallel. It also includes CRC checking and generation, PID decoder, bit-stuffing and de-bit-stuffing functions.

## Double-Buffering

The double buffering function is recommended to be enabled when the corresponding endpoint is specified to be used for Isochronous transfer or high throughput Bulk transfer. The double buffering function stores the preceding data packet sent by the USB host in a simple buffer for the MCU to process and the hardware will ensure that it continues to receive the current data packet in the other buffer during an OUT transaction, and vice versa. Using a double buffering function can achieve the highest possible data transfer rate. The details regarding double buffering usage is provided in the corresponding UDBTG and MDBTG control bit description in the USBEPnCSR register where the denotation n ranges from 4 to 7.



**Figure 170. Double-buffering Operation Example**

## Suspend Mode and Wake-up

According to USB specifications, the device must enter the suspend mode after a 3 ms bus idle time. When the USB device enters the suspend mode, the current from the USB bus must not be greater than 500  $\mu$ A to meet the specification suspend mode current requirements. The USB control circuitry will generate a suspend interrupt if the bus is in the idle state for 3 ms. Here the software should set the LPMODE and PDWN bits in the USBCSR register to 1. The LPMODE bit is used to determine whether the USB controller enters the low-power mode or not by holding the USB bus in a reset condition while the PDWN bit is used to determine if the integrated USB full-speed transceiver is turned off or not.

There are two ways for the USB host to wake up the USB device, one is to send a USB reset signal, SE0, and the other is to send a USB resume signal known as the K-state. After a wake-up signal, regardless of whether a SE0 signal or a K-state is detected, the USB device will be woken up.

## Remote Wake-up

As the USB device has a remote wake-up function, it can wake up the USB host by sending a resume request signal by setting the GENRSM bit in the USBCSR register to 1. Once the USB host receives the remote wake-up signal from the USB device, it will send a resume signal to the USB device.

## Register Map

The following table shows the USB registers and reset values.

**Table 66. USB Register Map**

Register	Offset	Description	Reset Value
USBCSR	0x000	USB Control and Status Register	0x0000_00X6
USBIER	0x004	USB Interrupt Enable Register	0x0000_0000
USBISR	0x008	USB Interrupt Status Register	0x0000_0000
USBFCR	0x00C	USB Frame Count Register	0x0000_0000
USBDEVAR	0x010	USB Device Address Register	0x0000_0000
USBEP0CSR	0x014	USB Endpoint 0 Control and Status Register	0x0000_0002
USBEP0IER	0x018	USB Endpoint 0 Interrupt Enable Register	0x0000_0000
USBEP0ISR	0x01C	USB Endpoint 0 Interrupt Status Register	0x0000_0000
USBEP0TCR	0x020	USB Endpoint 0 Transfer Count Register	0x0000_0000
USBEP0CFGR	0x024	USB Endpoint 0 Configuration Register	0x8000_0008
USBEP1CSR	0x028	USB Endpoint 1 Control and Status Register	0x0000_0002
USBEP1IER	0x02C	USB Endpoint 1 Interrupt Enable Register	0x0000_0000
USBEP1ISR	0x030	USB Endpoint 1 Interrupt Status Register	0x0000_0000
USBEP1TCR	0x034	USB Endpoint 1 Transfer Count Register	0x0000_0000
USBEP1CFGR	0x038	USB Endpoint 1 Configuration Register	0x1000_03FF
USBEP2CSR	0x03C	USB Endpoint 2 Control and Status Register	0x0000_0002
USBEP2IER	0x040	USB Endpoint 2 Interrupt Enable Register	0x0000_0000
USBEP2ISR	0x044	USB Endpoint 2 Interrupt Status Register	0x0000_0000
USBEP2TCR	0x048	USB Endpoint 2 Transfer Count Register	0x0000_0000
USBEP2CFGR	0x04C	USB Endpoint 2 Configuration Register	0x1000_03FF
USBEP3CSR	0x050	USB Endpoint 3 Control and Status Register	0x0000_0002
USBEP3IER	0x054	USB Endpoint 3 Interrupt Enable Register	0x0000_0000
USBEP3ISR	0x058	USB Endpoint 3 Interrupt Status Register	0x0000_0000
USBEP3TCR	0x05C	USB Endpoint 3 Transfer Count Register	0x0000_0000
USBEP3CFGR	0x060	USB Endpoint 3 Configuration Register	0x1000_03FF
USBEP4CSR	0x064	USB Endpoint 4 Control and Status Register	0x0000_0002
USBEP4IER	0x068	USB Endpoint 4 Interrupt Enable Register	0x0000_0000
USBEP4ISR	0x06C	USB Endpoint 4 Interrupt Status Register	0x0000_0000
USBEP4TCR	0x070	USB Endpoint 4 Transfer Count Register	0x0000_0000
USBEP4CFGR	0x074	USB Endpoint 4 Configuration Register	0x1000_03FF
USBEP5CSR	0x078	USB Endpoint 5 Control and Status Register	0x0000_0002
USBEP5IER	0x07C	USB Endpoint 5 Interrupt Enable Register	0x0000_0000
USBEP5ISR	0x080	USB Endpoint 5 Interrupt Status Register	0x0000_0000
USBEP5TCR	0x084	USB Endpoint 5 Transfer Count Register	0x0000_0000
USBEP5CFGR	0x088	USB Endpoint 5 Configuration Register	0x1000_03FF

Register	Offset	Description	Reset Value
USBEP6CSR	0x08C	USB Endpoint 6 Control and Status Register	0x0000_0002
USBEP6IER	0x090	USB Endpoint 6 Interrupt Enable Register	0x0000_0000
USBEP6ISR	0x094	USB Endpoint 6 Interrupt Status Register	0x0000_0000
USBEP6TCR	0x098	USB Endpoint 6 Transfer Count Register	0x0000_0000
USBEP6CFGR	0x09C	USB Endpoint 6 Configuration Register	0x1000_03FF
USBEP7CSR	0x0A0	USB Endpoint 7 Control and Status Register	0x0000_0002
USBEP7IER	0x0A4	USB Endpoint 7 Interrupt Enable Register	0x0000_0000
USBEP7ISR	0x0A8	USB Endpoint 7 Interrupt Status Register	0x0000_0000
USBEP7TCR	0x0AC	USB Endpoint 7 Transfer Count Register	0x0000_0000
USBEP7CFGR	0x0B0	USB Endpoint 7 Configuration Register	0x1000_03FF

## Register Descriptions

### USB Control and Status Register – USBCSR

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

Offset: 0x000

Reset value: 0x0000\_00X6

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved			FREST	DPWKEN	DPPUEN	SRAMRSTC	ADRSET
	7	6	5	4	3	2	1	0
Type/Reset	RXDM	RXDP	GENRSM	Reserved	LPMODE	PDWN	FRES	Reserved
	RO	X RO	X RW	0	RW	0 RW	1 RW	1

Bits	Field	Descriptions
[12]	FREST	Force USB Reset Control Timing This bit is used to decide the timing to release the USB reset state. 0: When the FRES bit is cleared, immediately release the USB reset state 1: When the FRES bit is set, wait to release the USB reset state until SE0 is detected
[11]	DPWKEN	DP Wake-Up Enable 0: Disable DP wake-up 1: Enable DP wake-up
[10]	DPPUEN	DP Pull-Up Enable 0: Disable DP pull-up 1: Enable DP pull-up
[9]	SRAMRSTC	EP_SRAM Reset Condition 0: Reset EP_SRAM when (DP, DM) = (0,0) 1: Users can access EP_SRAM in spite of (DP, DM) state

Bits	Field	Descriptions
[8]	ADRSET	<p>Device Address Setting Control</p> <p>This bit is used to determine when the USB SIE updates the device address with the value of the USBDEVAR register.</p> <p>0: The SIE updates the device address immediately after an address is written into the USBDEVAR register.</p> <p>1: The SIE updates the device address after the USB Host has successfully read the data from the device by the IN operation. This bit is cleared by the SIE after the device address is updated.</p>
[7]	RXDM	<p>Received DM Line Status</p> <p>This bit is used to observe the status of DM data line status at the end of suspend routines to determine whether a wakeup event has occurred.</p>
[6]	RXDP	<p>Received DP Line Status</p> <p>This bit is used to observe the status of DP data line status at the end of suspend routines to determine whether a wake-up event has occurred.</p>
[5]	GENRSM	<p>Resume Request Generation Control</p> <p>This bit is used to generate a resume request which is sent to the USB host by writing 1 into this bit location. The USB remote wake-up function is always enabled. This bit will be cleared to 0 after a resume signal sent by the USB host has been received.</p>
[3]	LPMODE	<p>Low-power Mode Control</p> <p>This bit is used to determine the USB operating mode. Setting this bit will force the USB to enter the low-power mode. When USB bus traffic, known as a wake-up event, is detected by the hardware, this bit should be cleared by software.</p> <p>0: Exit the Low-power mode</p> <p>1: Enter the Low-power mode</p>
[2]	PDWN	<p>Power-Down Mode Control</p> <p>Setting this bit will power down the full-speed USB PHY transceiver. This will disconnect the USB PHY transceiver from the USB bus.</p> <p>0: Exit the Power-Down</p> <p>1: Enter the Power-Down mode</p>
[1]	FRES	<p>Force USB Reset Control</p> <p>This bit is used to reset the USB circuitry. Setting this bit will force the USB into a reset state until the software clears it. A USB reset interrupt will be generated if the corresponding interrupt enable bit in the USBIER register is set to 1. All related USB registers are reset to their default values.</p> <p>0: Release USB reset</p> <p>1: Force USB reset</p>

**Table 67. Resume Event Detection**

[RXDP, RXDM] Status	Wake-up Event	Required Resume Software Action
00	Root reset	None
10	None (noise on bus)	Go back to suspend mode
01	Root resume	None
11	Not allowed (noise on bus)	Go back to suspend mode

## USB Interrupt Enable Register – USBIER

This register specifies the USB interrupt enable control.

Offset: 0x004

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	EP7IE	EP6IE	EP5IE	EP4IE	EP3IE	EP2IE	EP1IE	EP0IE
	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
Type/Reset	Reserved	FRESIE	ESOFIE	SUSPIE	RSMIE	URSTIE	SOFIE	UGIE
		RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:8]	EPnIE	Endpoint n Interrupt Enable Control (n = 0 ~ 7) 0: Disable interrupt 1: Enable interrupt
[6]	FRESIE	Force USB Reset Control (FRES) Interrupt Enable Control 0: Disable FRES interrupt 1: Enable FRES interrupt
[5]	ESOFIE	Expected Start-Of-Frame (ESOF) Interrupt Enable Control 0: Disable ESOF interrupt 1: Enable ESOF interrupt
[4]	SUSPIE	Suspend Interrupt Enable Control 0: Disable Suspend interrupt 1: Enable Suspend interrupt
[3]	RSMIE	Resume Interrupt Enable Control 0: Disable Resume interrupt 1: Enable Resume interrupt
[2]	URSTIE	USB Reset Interrupt Enable Control 0: Disable USB Reset interrupt 1: Enable USB Reset interrupt
[1]	SOFIE	Start-Of-Frame (SOF) Interrupt Enable Control 0: Disable SOF interrupt 1: Enable SOF interrupt
[0]	UGIE	USB Global Interrupt Enable Control 0: Disable USB Global interrupt 1: Enable USB Global interrupt  This bit must be set to 1 to enable the corresponding USB interrupt function. If this bit is cleared to 0, the relevant USB interrupt will not be generated, however, the corresponding interrupt flags will still be asserted.

## USB Interrupt Status Register – USBISR

This register specifies the USB interrupt status.

Offset: 0x008

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[15:8]	EPnIF	Endpoint n Interrupt Flag (n = 0 ~ 7) This bit is set by the hardware to indicate the generation of relevant endpoint interrupts. Writing 1 into this bit to clear it. It is important to note that the interrupt flag can only be cleared when the endpoint interrupt status bit in the USBEPnISR register is equal to 0.
[6]	FRESIF	Force USB Reset Control (FRES) Interrupt Flag This bit is set by the hardware. When this bit is set 1, this means that Force USB Reset FRES has finished. This bit is cleared to 0 by writing 1.
[5]	ESOFIF	Expected Start-Of-Frame Interrupt Flag This bit is set by the hardware when an SOF packet is expected to be received. The USB host sends an SOF (Start-Of-Frame) packet each millisecond. If the USB device hardware does not receive it properly, an ESOF interrupt will be generated when the ESOFIE bit in the USBIER register is set to 1. If three consecutive ESOF interrupts are generated, which means that the SOF packet has been missed 3 times, the SUSPIF flag will be set to 1. This bit will be set to 1 when the missing SOF packets occur if the timer is not yet locked. This bit can be read or written. However, only 0 can be written into this bit. Writing 1 has no effect.
[4]	SUSPIF	Suspend Interrupt Flag This bit is set by the hardware when no data transfer has occurred for 3ms, indicating that a suspend request has been sent from the USB host. The suspend condition check is enabled immediately after a USB reset. This bit is cleared to 0 by writing 1.
[3]	RSMIF	Resume Interrupt Flag This bit is set by the hardware. When this bit is set 1, this means that a device resume has occurred. This bit is cleared to 0 by writing 1.

Bits	Field	Descriptions
[2]	URSTIF	<p>USB Reset Interrupt Flag</p> <p>This bit is set by the hardware when the USB reset has been detected. When a USB reset occurs, the internal protocol state machine will be reset and an USB reset interrupt will be generated if the URSTIE bit in the USBIER register is set to 1. Data reception and transmission are disabled until the URSTIF bit is cleared to 0. The USB configuration related registers (USBCSR, USBIER, USBISR, USBFCR and USBDEVAR) will not be reset by a USB reset event except for the USB device address (USBDEVAR), this is to ensure that a USB reset interrupt can be safely excited and any data transactions immediately followed by the USB reset can be completely accessed by the software. Therefore the microcontroller must properly reset these registers. The USB endpoint related registers (USBEPnCSR, USBEPnISR and USBEPnTCR) are also reset by a USB reset event, however, the endpoint configuration (USBEPnCFGR) and interrupt enable (USBEPnIER) registers are not affected by the USB reset event and will remain unchanged.</p> <p>This bit is cleared to 0 by writing 1.</p>
[1]	SOFIF	<p>SOF Interrupt Flag</p> <p>This bit is set by the hardware when a start-of-frame packet has been received. This bit is cleared to 0 by writing 1.</p>

### USB Frame Count Register – USBFCR

This register specifies the lost Start-of-Frame number and the USB frame count.

Offset: 0x00C

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[18:17]	LSOF	<p>Lost Start-of-Frame number</p> <p>These bits are written and incremented by 1 by the hardware each time the ESOFIF bit is set. It is used to count the number of lost SOF packets. Once this field equals to 3, which means that the SOF packet has been lost 3 times, the SUSPIF flag will be set to 1. When a SOF packet has been received, these bits are cleared.</p>



Bits	Field	Descriptions
[16]	SOFLCK	Start-of-Frame Lock Flag This bit is set by the hardware when SOF packets have been received before the frame timer times out. Once this flag is set to 1, the frame number which is sent from the USB host will be loaded into the Frame Number field FRNUM in the USBFCR register. If there is no SOF packet has been received during the 1ms frame time duration, this bit will be cleared to 0.
[10:0]	FRNUM	Frame Number This field stores the frame number received from the USB host.

### USB Device Address Register – USBDEVAR

This register specifies the USB device address.

Offset: 0x010

Reset value: 0x0000\_0000

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

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

## USB Endpoint 0 Control and Status Register – USBEP0CSR

This register specifies the Endpoint 0 control and status.

Offset: 0x014

Reset value: 0x0000\_0002

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset		Reserved	STLRX	NAKRX	DTGRX	STLTX	NAKTX	DTGTX
			RW	0	RW	0	RW	0
				0			1	0

Bits	Field	Descriptions
[5]	STLRX	<p>STALL Status for reception (OUT) transfer</p> <p>This bit is set to 1 by the application software and then returns a STALL signal in the handshake phase of an OUT transaction if a functional error is detected. This means that a control request delivered from the USB host is not supported by the USB device. The STALL status is cleared by the hardware circuitry when a new SETUP token is received.</p> <p>This bit can be read and written and can only be toggled by writing 1.</p>
[4]	NAKRX	<p>NAK Status for reception (OUT) transfer</p> <p>This bit is toggled from 0 to 1 by the hardware circuitry, which will result in a NAK signal in the handshake phase of an OUT transaction after an ACK signal has been transmitted. This means that the USB device will be temporarily unable to accept data from the USB host. Therefore, more time will be required for the received data to be properly processed.</p> <p>This bit can be read and written and can only be toggled by writing 1.</p>
[3]	DTGRX	<p>Data Toggle Status for reception (OUT) transfer</p> <p>This bit contains the expected value of the data toggle bit (0 = DATA0, 1 = DTAT1) for the next data packet to be received. When the current valid data packet is received and the corresponding ACK signal is sent to the USB host by the USB device, the hardware circuitry will toggle this bit and the device will be ready to receive the next data packet. For Endpoint 0, the hardware circuitry will toggle this bit to 1 after the SETUP token is received as Endpoint 0 is addressed. This bit can also be toggled by the software to initialize its value for certain applications.</p> <p>This bit can be read and written and can only be toggled by writing 1.</p>
[2]	STLTX	<p>STALL Status for transmission (IN) transfer</p> <p>This bit is set to 1 by the application software and then returns a STALL signal in response to an IN token if a functional error is detected. This means that the USB device is unable to transmit data. The STALL status is cleared by the hardware circuitry when a new SETUP token is received.</p> <p>This bit can be read and written and can only be toggled by writing 1.</p>

Bits	Field	Descriptions
[1]	NAKTX	NAK Status for transmission (IN) transfer This bit is toggled from 0 to 1 by the hardware circuitry, which will result in a NAK signal in the handshake phase of an IN transaction after an ACK signal has been received. It indicates that the USB device is temporarily unable to transmit data to the USB host. Therefore, there will be more time for the application software to properly prepare the data to be transmitted. This bit can be read and written and can only be toggled by writing 1.
[0]	DTGTX	Data Toggle Status for transmission (IN) transfer This bit contains the required value of the data toggle bit (0 = DATA0, 1 = DATA1) for the next data packet to be transmitted. When the current data packet is transmitted by the USB device and the corresponding ACK signal sent by the USB host is received, the hardware circuitry will toggle this bit and the next data packet will be transmitted. For Endpoint 0, the hardware circuitry will toggle this bit to 1 after the SETUP token is received as Endpoint 0 is addressed. This bit can also be toggled by the software to initialize its value for certain applications. This bit can be read and written and can only be toggled by writing 1.

### USB Endpoint 0 Interrupt Enable Register – USBEP0IER

This register specifies the Endpoint 0 interrupt control bits.

Offset: 0x018

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved				ZLRXIE	SDERIE	SDRXIE	STRXIE
	7	6	5	4	3	2	1	0
Type/Reset	UERIE	STLIE	NAKIE	IDTXIE	ITRXIE	ODOVIE	ODRXIE	OTRXIE
	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[11]	ZLRXIE	Zero Length Data Received Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[10]	SDERIE	SETUP Data Error Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[9]	SDRXIE	SETUP Data Received Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt

Bits	Field	Descriptions
[8]	STRXIE	SETUP Token Received Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[7]	UERIE	USB Error Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[6]	STLIE	STALL Transmitted Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[5]	NAKIE	NAK Transmitted Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[4]	IDTXIE	IN Data Transmitted Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[3]	ITRXIE	IN Token Received Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[2]	ODOVIE	OUT Data Buffer Overrun Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[1]	ODRXIE	OUT Data Received Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[0]	OTRXIE	OUT Token Received Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt

## USB Endpoint 0 Interrupt Status Register – USBEP0ISR

This register specifies the Endpoint 0 interrupt status.

Offset: 0x01C

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved				ZLRXIF	SDERIF	SDRXIF	STRXIF
					WC	0	WC	0
	7	6	5	4	3	2	1	0
Type/Reset	UERIF	STLIF	NAKIF	IDTXIF	ITRXIF	ODOVIF	ODRXIF	OTRXIF
	WC	0	WC	0	WC	0	WC	0

Bits	Field	Descriptions
[11]	ZLRXIF	Zero Length Data Received Interrupt Flag This bit is set by the hardware when a zero length data packet is received. This bit is cleared by hardware when a SETUP Token is received or by writing 1.
[10]	SDERIF	SETUP Data Error Interrupt Flag This bit is set by the hardware when the SETUP data packet length is not 8 bytes. This bit is cleared by hardware when a SETUP Token is received or by writing 1.
[9]	SDRXIF	SETUP Data Received Interrupt Flag This bit is set by the hardware when a SETUP data packet from the USB host has been received. This bit is cleared by the hardware when a SETUP Token is received or by writing 1. If the received SETUP data is not accessed by the application software before the next SETUP packet is received, the SETUP data buffer will be overwritten.
[8]	STRXIF	SETUP Token Received Interrupt Flag This bit is set by the hardware when a SETUP token is received and is cleared by writing 1.
[7]	UERIF	USB Error Interrupt Flag This bit is set by the hardware when an error occurs during the Endpoint 0 transaction. This bit is cleared by hardware when a SETUP Token is received or by writing 1.
[6]	STLIF	STALL Transmitted Interrupt Flag This bit is set by the hardware when a STALL signal is sent in response to an IN or OUT transaction. This bit is cleared by hardware when a SETUP Token is received or by writing 1.
[5]	NAKIF	NAK Transmitted Interrupt Flag This bit is set by the hardware when a NAK signal is sent in response to an IN or OUT transaction. This bit is cleared by hardware when a SETUP Token is received or by writing 1.

Bits	Field	Descriptions
[4]	IDTXIF	IN Data Transmitted Interrupt Flag This bit is set by the hardware when a data packet is transmitted to and then an ACK signal is received from the USB host. This bit is cleared by hardware when a SETUP Token is received or by writing 1.
[3]	ITRXIF	IN Token Received Interrupt Flag This bit is set by the hardware when the IN token is received from the USB host. This bit is cleared by hardware when a SETUP Token is received or by writing 1.
[2]	ODOVIF	OUT Data Buffer Overrun Interrupt Flag This bit is set by the hardware when the number of received data bytes is larger than the endpoint buffer size. This bit is cleared by hardware when a SETUP Token is received or by writing 1.
[1]	ODRXIF	OUT Data Received Interrupt Flag This bit is set by the hardware when a data packet is successfully received from the USB host and then an ACK signal is sent to the USB host. This bit is cleared by hardware when a SETUP Token is received or by writing 1.
[0]	OTRXIF	OUT Token Received Interrupt Flag This bit is set by the hardware when the OUT token is received from the USB host. This bit is cleared by hardware when a SETUP Token is received or by writing 1.

### USB Endpoint 0 Transfer Count Register – USBEP0TCR

This register specifies the Endpoint 0 data transfer byte count.

Offset: 0x020

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved	RXCNT						
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved	TXCNT						
	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[22:16]	RXCNT	Reception Byte Count The bit field contains the number of data bytes received by Endpoint 0 in the preceding SETUP transaction.
[6:0]	TXCNT	Transmission Byte Count The bit field contains the number of data bytes to be transmitted by Endpoint 0 in the next IN token. If the value of this field is zero, it indicates that a zero length packet will be sent.

## USB Endpoint 0 Configuration Register – USBEP0CFGR

This register specifies the Endpoint 0 configurations.

Offset: 0x024

Reset value: 0x8000\_0008

	31	30	29	28	27	26	25	24
	EPEN	Reserved						EPADR
Type/Reset	RO 1	RO 0 RO 0 RO 0 RO 0						0
	23	22	21	20	19	18	17	16
	Reserved							EPLEN
Type/Reset								RW 0
	15	14	13	12	11	10	9	8
	EPLEN						EPBUFA	
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	EPBUFA							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 1	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[31]	EPEN	Endpoint Enable Control This bit is always set to 1 by the hardware circuitry to always enable Endpoint 0.
[27:24]	EPADR	Endpoint Address This field is always set to 0 by the hardware circuitry.
[16:10]	EPLEN	Endpoint Buffer Length This field is used to specify the control transfer packet size which can be 8, 16, 32 or 64 bytes as defined in the USB full-speed standard specification.
[9:0]	EPBUFA	Endpoint Buffer Address This field is used to specify the start address of the Endpoint 0 buffer allocated in the EP_SRAM. It starts from 0x008 and should be aligned to 4-byte boundary. Start address of EP0 IN buffer = EPBUFA Start address of EP0 OUT buffer = EPBUFA + EPLEN

## USB Endpoint 1 ~ 3 Control and Status Register – USBEPnCSR, n = 1 ~ 3

This register specifies the Endpoint 1 ~ 3 control and status bit.

Offset: 0x028 (n = 1), 0x03C (n = 2), 0x050 (n = 3)

Reset value: 0x0000\_0002

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset		Reserved	STLRX	NAKRX	DTGRX	STLTX	NAKTX	DTGTX
			RW	0	RW	0	RW	0
				0	RW	0	RW	0
					0	RW	1	RW
								0

Bits	Field	Descriptions
[5]	STLRX	<p>STALL bit for reception transfers</p> <p>This bit is set to 1 by the application software if a functional error has been detected. This bit can be read and written and can only be toggled by writing 1. It can also be toggled by the software to initialize the value under certain conditions.</p>
[4]	NAKRX	<p>NAK bit for reception transfers</p> <p>This bit is toggled from 0 to 1 by the hardware circuitry, which will result in a NAK signal in the handshake phase of an OUT transaction after an ACK signal has been transmitted. It means that the USB device will be temporarily unable to accept data from the USB host until the received data is properly processed.</p> <p>This bit can be read and written and can be only toggled by writing 1.</p>
[3]	DTGRX	<p>Data Toggle bit for reception transfers</p> <p>This bit contains the expected value of the data toggle bit (0 = DATA0, 1 = DATA1) for the next data packet to be received. When the current valid data packet is received and the corresponding ACK signal is sent to the USB host by the USB device, the hardware circuitry will toggle this bit and the device will be ready to receive the next data packet.</p> <p>This bit can be read and written and can only be toggled by writing 1. This bit can also be toggled by the software to initialize its value under certain conditions.</p>
[2]	STLTX	<p>STALL bit for transmission transfers</p> <p>This bit is set to 1 by the application software if a functional error has been detected. This bit can be read and written and can be only toggled by writing 1. It can also be toggled by the software to initialize its value under certain conditions.</p>
[1]	NAKTX	<p>NAK bit for transmission transfers</p> <p>This bit is toggled from 0 to 1 by the hardware circuitry, which will result in a NAK signal in the handshake phase of an IN transaction after an ACK signal has been received. It means that the USB device will be temporarily unable to transmit data packet until the data to be transmitted is appropriately prepared by the application software.</p> <p>This bit can be read and written and can be only toggled by writing 1.</p>



Bits	Field	Descriptions
[0]	DTGTX	<p>Data Toggle bit for transmission transfers</p> <p>This bit contains the required value of the data toggle bit (0 = DATA0, 1 = DATA1) for the next data packet to be transmitted. When the current data packet is transmitted by the USB device and the corresponding ACK signal sent from the USB host is received, the hardware circuitry will toggle this bit and then the next data packet will be transmitted.</p> <p>This bit can be read and written and can only be toggled by writing 1. It can also be toggled by the software to initialize its value under certain conditions.</p>

### USB Endpoint 1 ~ 3 Interrupt Enable Register – USBEPnIER, n = 1 ~ 3

This register specifies the Endpoint 1 ~ 3 interrupt enable control bits.

Offset: 0x02C (n = 1), 0x040 (n = 2), 0x054 (n = 3)

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	UERIE	STLIE	NAKIE	IDTXIE	ITRXIE	ODOVIE	ODRXIE	OTRXIE
Type/Reset	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[7]	UERIE	<p>USB Error Interrupt Enable Control</p> <p>0: Disable interrupt</p> <p>1: Enable interrupt</p>
[6]	STLIE	<p>STALL Transmitted Interrupt Enable Control</p> <p>0: Disable interrupt</p> <p>1: Enable interrupt</p>
[5]	NAKIE	<p>NAK Transmitted Interrupt Enable Control</p> <p>0: Disable interrupt</p> <p>1: Enable interrupt</p>
[4]	IDTXIE	<p>IN Data Transmitted Interrupt Enable Control</p> <p>0: Disable interrupt</p> <p>1: Enable interrupt</p>
[3]	ITRXIE	<p>IN Token Received Interrupt Enable Control</p> <p>0: Disable interrupt</p> <p>1: Enable interrupt</p>
[2]	ODOVIE	<p>OUT Data Buffer Overrun Interrupt Enable Control</p> <p>0: Disable interrupt</p> <p>1: Enable interrupt</p>

Bits	Field	Descriptions
[1]	ODRXIE	OUT Data Received Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[0]	OTRXIE	OUT Token Received Interrupt Enable Control. 0: Disable interrupt 1: Enable interrupt

### USB Endpoint 1 ~ 3 Interrupt Status Register – USBEPnISR, n = 1 ~ 3

This register specifies the Endpoint 1 ~ 3 interrupt status.

Offset: 0x030 (n = 1), 0x044 (n = 2), 0x058 (n = 3)

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	WC	0	WC	0	WC	0	WC	0
	UERIF	STLIF	NAKIF	IDTXIF	ITRXIF	ODOVIF	ODRXIF	OTRXIF
Type/Reset	WC	0	WC	0	WC	0	WC	0

Bits	Field	Descriptions
[7]	UERIF	USB Error Interrupt Flag This bit is set by the hardware when an error occurs during the transaction. Writing 1 into this status bit will clear it to 0.
[6]	STLIF	STALL Transmitted Interrupt Flag This bit is set by hardware circuitry when a STALL signal is sent in response to an IN or OUT token and is cleared to 0 by writing 1.
[5]	NAKIF	NAK Transmitted Interrupt Flag This bit is set by hardware circuitry when an NAK signal is sent in response to an IN or OUT token and is cleared to 0 by writing 1.
[4]	IDTXIF	IN Data Transmitted Interrupt Flag This bit is set by hardware circuitry when a data packet is successfully transmitted to the host in response to an IN token and an ACK signal is received. Writing 1 into this status bit will clear it to 0.
[3]	ITRXIF	IN Token Received Interrupt Flag This bit is set by the hardware circuitry when the endpoint receives an IN token from the host and is cleared to 0 by writing 1.

Bits	Field	Descriptions
[2]	ODOVIF	OUT Data Buffer Overrun Interrupt Flag This bit is set by the hardware circuitry when the received data byte count is larger than the corresponding endpoint OUT data buffer size. Writing 1 into this status bit will clear it to 0.
[1]	ODRXIF	OUT Data Received Interrupt Flag This bit is set by the hardware circuitry when a data packet is successfully received from the host for an OUT token and when an endpoint n ACK signal is sent to the host. Writing 1 into this status bit will clear it to 0.
[0]	OTRXIF	OUT Token Received Interrupt Flag This bit is set by the hardware circuitry when the endpoint receives an OUT token from the host and is cleared to 0 by writing 1.

### USB Endpoint 1 ~ 3 Transfer Count Register – USBEPnTCR, n = 1 ~ 3

This register specifies the Endpoint 1 ~ 3 transfer byte count.

Offset: 0x034 (n = 1), 0x048 (n = 2), 0x05C (n = 3)

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	Reserved							TCNT	
	7	6	5	4	3	2	1	0	
Type/Reset	TCNT								
	RW	0	RW	0	RW	0	RW	0	
	0	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[8:0]	TCNT	Transfer Byte Count This field contains the number of bytes received by the endpoint n in the preceding OUT transaction or the number of bytes to be transmitted by the endpoint n in the next IN transaction.

## USB Endpoint 1 ~ 3 Configuration Register – USBEPnCFGR, n = 1 ~ 3

This register specifies the Endpoint 1 ~ 3 configurations.

Offset: 0x038 (n = 1), 0x04C (n = 2), 0x060 (n = 3)

Reset value: 0x1000\_03FF

	31	30	29	28	27	26	25	24
	EPEN	Reserved	EPTYPE	EPDIR	EPADR			
Type/Reset	RW 0		RW 0	RW 1	RW 0	RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	Reserved							EPLEN
Type/Reset								RW 0
	15	14	13	12	11	10	9	8
	EPLEN						EPBUFA	
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 1	RW 1
	7	6	5	4	3	2	1	0
	EPBUFA							
Type/Reset	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1

Bits	Field	Descriptions
[31]	EPEN	Enable Control 0: Disable the endpoint n 1: Enable the endpoint n
[29]	EPTYPE	Transfer Type This bit is set to 0 by the hardware circuitry to specify that the endpoint n transfer type is an Interrupt or Bulk transfer type.
[28]	EPDIR	Transfer Direction 0: OUT 1: IN
[27:24]	EPADR	Endpoint Address The EPADR field value can be assigned by the application software to specify the address of the endpoint n. It is important to note that this EPADR field should not be set to 0; otherwise, the endpoint will be disabled.
[16:10]	EPLEN	Buffer Length This field is used to specify the endpoint n data packet size. The field value must be word-aligned to a 4-byte boundary. The maximum size in this field can be 64 bytes which is the maximum payload as defined in the USB full-speed standard specification. Note that the EPLEN value should not be assigned to 0 which will result in the endpoint being disabled.
[9:0]	EPBUFA	Endpoint Buffer Address This field is used to specify the endpoint n data buffer start address which ranges from 0x008 to 0x3FC in the EP_SRAM which has a capacity of 1024 bytes and whose field value must be a multiple of 4.

## USB Endpoint 4 ~ 7 Control and Status Register – USBEPnCSR, n = 4 ~ 7

This register specifies the Endpoint 4 ~ 7 control and status bits.

Offset: 0x064 (n = 4), 0x078 (n = 5), 0x08C (n = 6), 0x0A0 (n = 7)

Reset value: 0x0000\_0002

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	UDBTG	MDBTG	STLRX	NAKRX	DTGRX	STLTX	NAKTX	DTGTX
	RW	0	RW	0	RW	0	RW	1
								RW
								0

Bits	Field	Descriptions
------	-------	--------------

[7]	UDBTG	<p>USB Double Buffer Toggle bit</p> <p>The UDBTG and MDBTG bits are used to indicate which data buffer is accessed by the USB SIE hardware and which data buffer is accessed by the MCU software if the double buffering function is enabled. The UDBTG bit will be toggled by the SIE hardware circuitry after the current buffer operation is complete. After the UDBTG bit is toggled by the SIE, an NAK signal will be sent automatically to the USB host by the hardware circuitry. Therefore, the data transfer will be stopped temporarily until the data in the other buffer has been properly setup after which the MDBTG bit is toggled by the MCU application software.</p> <p>The following tables show the double buffering operation and the UDBTG and MDBTG bit status for an IN or OUT transaction.</p>
-----	-------	---

Transaction Type	UDBTG	MDBTG	Buffer Read by SIE	Buffer Written by MCU
IN	0	0	None*	EP_BUF0
	0	1	EP_BUF0	EP_BUF1
	1	1	None*	EP_BUF1
	1	0	EP_BUF1	EP_BUF0

Transaction Type	UDBTG	MDBTG	Buffer Written by SIE	Buffer Read by MCU
OUT	0	0	None*	EP_BUF0
	0	1	EP_BUF0	EP_BUF1
	1	1	None*	EP_BUF1
	1	0	EP_BUF1	EP_BUF0

\* Means the USB device sends a NAK signal to the USB host using the hardware circuitry.

The UDBTG and MDBTG bits setting procedure for the double buffering function is shown in the following example:

[UDBTG, MDBTG] = [0, 0] → [0, 1] → [1, 1] → [1, 0] → [0, 0] → [0, 1] → [1, 1] → [1, 0] → ...

Bits	Field	Descriptions
[6]	MDBTG	<p>MCU Double Buffer Toggle bit</p> <p>The MDBTG bit is used to indicate which data buffer is accessed by the MCU if the double buffering function is enabled. It can be toggled to switch to the other buffer by the MCU application software after the data in the current buffer accessed by the MCU has been properly setup. The double buffering operation together with the UDBTG and MDBTG bits are shown in the preceding two tables for the UDBTG bit definition</p>
[5]	STLRX	<p>STALL bit for reception transfers</p> <p>This bit is set to 1 by the application software if a functional error has been detected. This bit can be read and written and can only be toggled by writing 1. It can also be toggled by software to initialize its value under certain conditions.</p>
[4]	NAKRX	<p>NAK bit for reception transfers</p> <p>This bit is toggled from 0 to 1 by the hardware circuitry, which will result in a NAK signal in the handshake phase of an OUT transaction after an ACK signal has been transmitted. It means that the USB device will be temporarily unable to accept data from the USB host until the received data is properly processed. If the endpoint is defined as an Isochronous transfer type, this bit is not available for usage. The hardware will not change the NAKRX bit status after a complete transaction. This bit can be read and written and can be only toggled by writing 1.</p>
[3]	DTGRX	<p>Data Toggle bit for reception transfers</p> <p>If the endpoint is not used for Isochronous transfer, this bit is available for usage. This bit contains the expected value of the data toggle bit (0 = DATA0, 1 = DATA1) for the next data packet to be received. When the current valid data packet is received and the corresponding ACK signal is sent to the USB host by the USB device, the hardware circuitry will toggle this bit and the device will be ready to receive the next data packet. If the endpoint is defined as an Isochronous transfer type, this bit is not used since no data toggling is used and only the DATA0 packet will be transferred for normal Isochronous transfers. This bit can be read and written and can only be toggled by writing 1. This bit can also be toggled by the software to initialize its value under certain conditions.</p>
[2]	STLTX	<p>STALL bit for transmission transfers</p> <p>This bit is set to 1 by the application software if there a functional error has been detected. This bit can be read and written and can be only toggled by writing 1. It can be toggled by the software to initialize its value under certain conditions.</p>
[1]	NAKTX	<p>NAK bit for transmission transfers</p> <p>This bit is toggled from 0 to 1 by the hardware circuitry, which will result in a NAK signal in the handshake phase of an IN transaction after an ACK signal has been received. It means that the USB device will be temporarily unable to transmit a data packet until the data to be transmitted is properly setup by the application software. If the endpoint is defined as an Isochronous transfer type, then this bit is not available for usage. The hardware will not change the NAKTX bit status after a complete transaction. This bit can be read and written and can be only toggled by writing 1. It can also be toggled by the software to initialize its value under certain conditions.</p>

Bits	Field	Descriptions
[0]	DTGTX	Data Toggle bit for transmission transfers If the endpoint is not used for Isochronous transfer, this bit is available for usage. This bit contains the required value of the data toggle bit (0 = DATA0, 1 = DATA1) for the next data packet to be transmitted. When the current data packet is transmitted by the USB device and the corresponding ACK signal sent from the USB host is received, the hardware circuitry will toggle this bit and then the next data packet will be transmitted. If the endpoint is used for Isochronous transfer, this bit is not used since no data toggling is used and only the DATA0 packet will be transferred for normal Isochronous transfer. This bit can be read and written and can only be toggled by writing 1. It can also be toggled by the software to initialize its value under certain conditions.

### USB Endpoint 4 ~ 7 Interrupt Enable Register – USBEPnIER, n = 4 ~ 7

This register specifies the Endpoint 4 ~ 7 interrupt enable control bits.

Offset: 0x068 (n = 4), 0x07C (n = 5), 0x090 (n = 6), 0x0A4 (n = 7)

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
	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
	URIE	STLIE	NAKIE	IDTXIE	ITRXIE	ODOVIE	ODRXIE	OTRXIE
Type/Reset	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[7]	URIE	USB Error Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[6]	STLIE	STALL Transmitted Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[5]	NAKIE	NAK Transmitted Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[4]	IDTXIE	IN Data Transmitted Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[3]	ITRXIE	IN Token Received Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt

Bits	Field	Descriptions
[2]	ODOVIE	OUT Data Buffer Overrun Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[1]	ODRXIE	OUT Data Received Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[0]	OTRXIE	OUT Token Received Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt

### USB Endpoint 4 ~ 7 Interrupt Status Register – USBEPnISR, n = 4 ~ 7

This register specifies the Endpoint 4 ~ 7 interrupt status.

Offset: 0x06C (n = 4), 0x080 (n = 5), 0x094 (n = 6), 0x0A8 (n = 7)

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	UERIF	STLIF	NAKIF	IDTXIF	ITRXIF	ODOVIF	ODRXIF	OTRXIF
	WC	0	WC	0	WC	0	WC	0

Bits	Field	Descriptions
[7]	UERIF	USB Error Interrupt flag This bit is set by the hardware circuitry when an error occurs during the transaction. Writing 1 into this status bit will clear it to 0.
[6]	STLIF	STALL Transmitted Interrupt flag This bit is set by the hardware circuitry when a STALL signal is sent in response to an IN or OUT token and is cleared to 0 by writing 1.
[5]	NAKIF	NAK Transmitted Interrupt flag This bit is set by the hardware circuitry when an NAK signal is sent in response to an IN or OUT token and is cleared to 0 by writing 1.
[4]	IDTXIF	IN Data Transmitted Interrupt flag This bit is set by the hardware circuitry when a data packet is successfully transmitted to the host in response to an IN token and an ACK signal is received. Writing 1 into this status bit will clear it to 0.
[3]	ITRXIF	IN Token Received Interrupt flag This bit is set by the hardware circuitry when the endpoint receives an IN token from the host and is cleared to 0 by writing 1.



Bits	Field	Descriptions
[2]	ODOVIF	OUT Data Buffer Overrun Interrupt flag This bit is set by the hardware circuitry when the received data byte count is larger than the endpoint OUT data buffer size. Writing 1 into this status bit will clear it to 0.
[1]	ODRXIF	OUT Data Received Interrupt flag This bit is set by the hardware circuitry when a data packet is successfully received from the host for an OUT token and an ACK signal is sent to the host. Writing 1 into this status bit will clear it to 0.
[0]	OTRXIF	OUT Token Received Interrupt flag This bit is set by the hardware circuitry when the endpoint receives an OUT token from the host and is cleared to 0 by writing 1.

### USB Endpoint 4 ~ 7 Transfer Count Register – USBEPnTCR, n = 4 ~ 7

This register specifies the Endpoint 4 ~ 7 transfer byte count.

Offset: 0x070 (n = 4), 0x084 (n = 5), 0x098 (n = 6), 0x0AC (n = 7)

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	Reserved						TCNT1		
Type/Reset							RW	0	RW
	23	22	21	20	19	18	17	16	
	TCNT1								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	Reserved						TCNT0		
Type/Reset							RW	0	RW
	7	6	5	4	3	2	1	0	
	TCNT0								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[25:16]	TCNT1	Buffer 1 Transfer Byte Count This bit field contains the number of data bytes received by the endpoint n buffer 1 in the preceding OUT transaction or the number of data bytes to be transmitted by the endpoint n buffer1 in the next IN transaction.
[9:0]	TCNT0	Buffer 0 Transfer Byte Count This bit field contains the number of data bytes received by the endpoint n buffer 0 in the preceding OUT transaction or the number of data bytes to be transmitted by the endpoint n buffer 0 in the next IN transaction. Only the TCNT0 field is used for the endpoint data transfer count when the endpoint is configured as a single-buffering transfer type.

## USB Endpoint 4 ~ 7 Configuration Register – USBEPnCFGR, n = 4 ~ 7

This register specifies the Endpoint 4 ~ 7 configurations.

Offset: 0x074 (n = 4), 0x088 (n = 5), 0x09C (n = 6), 0x0B0 (n = 7)

Reset value: 0x1000\_03FF

	31	30	29	28	27	26	25	24
	EPEN	Reserved	EPTYPE	EPDIR	EPADR			
Type/Reset	RW 0		RW 0	RW 1	RW 0	RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	SDBS	Reserved			EPLEN			
Type/Reset	RW 0				RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	EPLEN						EPBUFA	
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 1	RW 1
	7	6	5	4	3	2	1	0
	EPBUFA							
Type/Reset	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1

Bits	Field	Descriptions
[31]	EPEN	Enable Control 0: Disable the endpoint n 1: Enable the endpoint n
[29]	EPTYPE	Transfer Type 0: Interrupt or Bulk transfer type 1: Isochronous transfer type
[28]	EPDIR	Transfer Direction 0: OUT 1: IN
[27:24]	EPADR	Endpoint Address The EPADR field can be configured by the application software to specify the address of endpoint n. It is important to note that this EPADR field should not be set to 0; otherwise, the endpoint n will be disabled.
[23]	SDBS	Single-Buffering or Double-Buffering Selection 0: Single-buffering 1: Double-buffering If the SDBS bit is set to 1, the endpoint buffer size is twice that of the EPLEN value: - Endpoint Buffer 0 start address is EPBUFA - Endpoint Buffer 1 start address is (EPBUFA + EPLEN)
[19:10]	EPLEN	Buffer Length This field is used to specify the endpoint n data packet size whose field value must be word-aligned to a 4-byte boundary. Note that the endpoint will be disabled if the EPLEN value is assigned to 0.
[9:0]	EPBUFA	Buffer Address This field is used to specify the endpoint n data buffer start address which ranges from 0x008 to 0x3FC in the EP_SRAM which has a capacity of 1024 bytes where the endpoint transfer data is stored. Note that the buffer start address value must be a multiple of 4.

# 25 Peripheral Direct Memory Access (PDMA)

## Introduction

The Peripheral Direct Memory Access circuitry, PDMA, provides 6 unidirectional channels for dedicated peripherals to implement the peripheral-to-memory and memory-to-peripheral data transfer. The memory-to-memory data transfer such as the FLASH-to-SRAM or SRAM-to-SRAM type is also supported and requested by the application program. Each PDMA channel configuration is independent. The PDMA channel transfer is split into multiple block transactions and the size of a block is equal to the block length multiplied by the data width.

## Features

- 6-unidirectional PDMA channels
- Memory-to-peripheral, peripheral-to-memory and memory-to-memory data transfer
- 8-bit, 16-bit and 32-bit width data transfer
- Software and hardware requested data transfer with configurable channel priority
- Linear address, circular address and fixed address modes
- 4 transfer event flags – Transfer complete, Half Transfer, Block End and Transfer Error
- Auto-Reload function

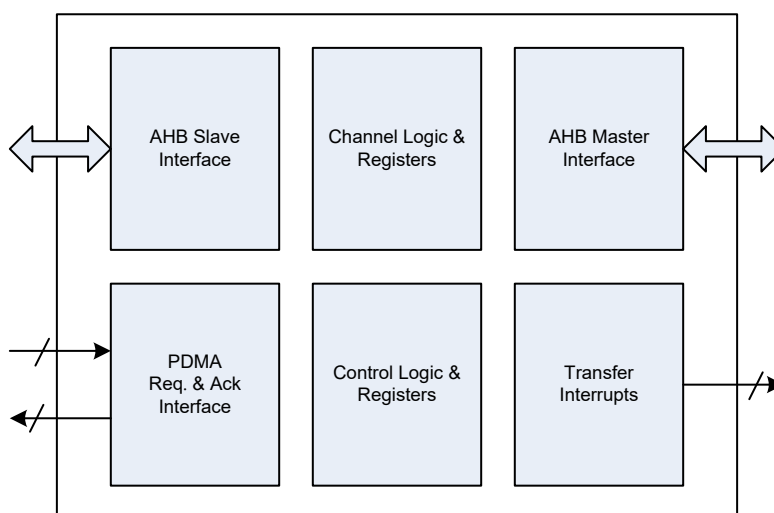


Figure 171. PDMA Block Diagram

## Functional Description

### AHB Master

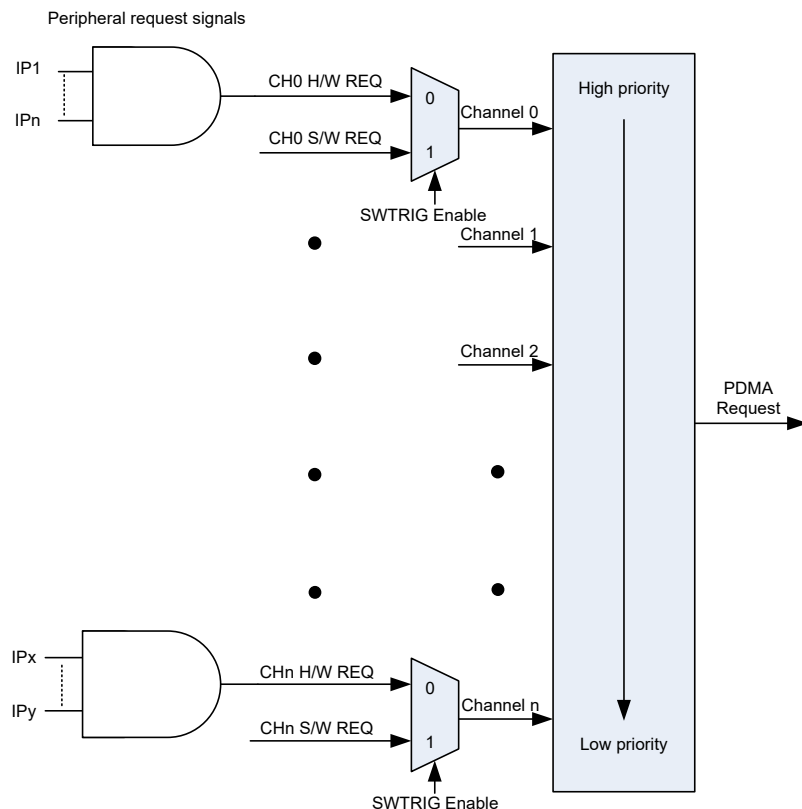
The PDMA is an AHB master connected to other AHB peripherals such as the FLASH memory, the SRAM memory and the AHB-to-APB bridges through the bus-matrix. The CPU and PDMA can access different AHB slaves at the same time via the bus-matrix.

### PDMA Channel

There are 6 unidirectional PDMA channels used to support data transfer between the peripherals and the memory. The configuration and operation of each PDMA channel is independent. For a bidirectional transfer application, two PDMA channels are required. Each PDMA channel is designed to support the dedicated multiple peripherals with the same registers. Therefore, one PDMA channel only can service one peripheral at the same time. The related registers of the PDMA channel are limited to be accessed with 32-bit operation; otherwise a system hard fault event will occur.

### PDMA Request Mapping

The multiple requests from the peripherals (ADC, SPI, I<sup>2</sup>C, 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 172 – PDMA Request Mapping Architecture and detail peripheral IP requests mapping table is shown as the Table 67. The peripheral DMA requests can be independently activated/de-activated by programming the DMA control bit in the registers of the corresponding peripheral.



**Figure 172. PDMA Request Mapping Architecture**

**Table 68. PDMA Channel Assignments**

IP (x=0,1)	PDMA Channel Number					
	CH0	CH1	CH2	CH3	CH4	CH5
ADC	ADC					
SPIx	SPI0_RX	SPI0_TX			SPI1_RX	SPI1_TX
USART	USR0_RX	USR0_TX				
UARTx			UR0_RX	UR0_TX	UR1_RX	UR1_TX
SCI					SCI_RX	SCI_TX
I2Cx			I2C0_RX	I2C1_RX	I2C0_TX	I2C1_TX
GPTM	GT_CH1 GT_CH3	GT_CH2 GT_UEV	GT_CH0 GT_TRIG			
PWM				PWM_CH1 PWM_CH3	PWM_CH2 PWM_UEV	PWM_CH0 PWM_TRIG
AES					AES_OUT	AES_IN

## Channel Transfer

A PDMA channel transfer is split into multiple block transactions with PDMA arbitration occurring at the end of each block transaction. Although these channel transfers can all be activated, there is only one block transaction being transferred through the bus at a time. The channel transfer sequence depends upon the channel priority setting of each PDMA channel. The total transfer size is calculated from the block transaction count and block size. The block size is equal to the product of the block length and data bit width. For an efficient transfer, it is recommended that the block length is set as a multiple of 4.

The total transfer data size calculation is shown as below equation:

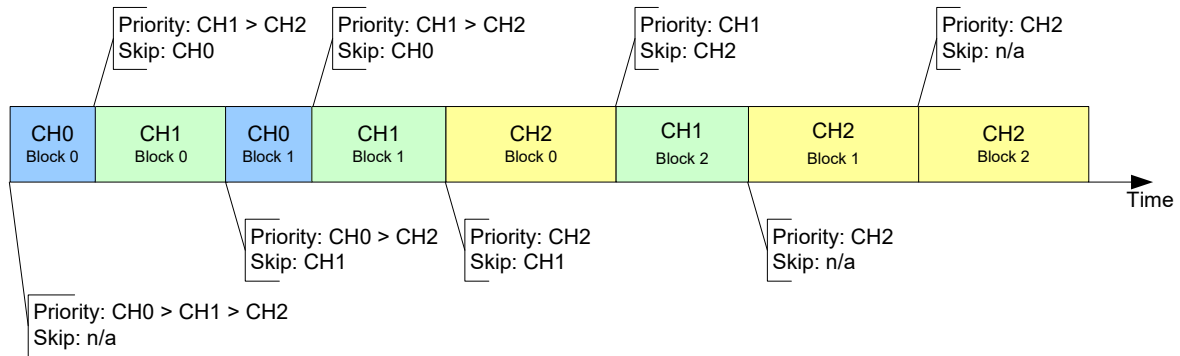
A PDMA channel total transfer data size = Block transaction count × (Block length × Data width)

## Channel Priority

The PDMA provides four priority levels, known as very high, high, medium and low, which can be configured by the application software. The PDMA also provides two methods to determine the channel priority. One is determined by application software configuration and the other is determined by the fixed hardware channel number. The PDMA arbitration processor will first check the software configuring channel priority level used to request the PDMA to provide the data transfer services. If more than one channel has the same priority, the channel with a smaller channel number will have priority over one with a larger channel number after arbitration.

Note that the highest priority channel will not occupy the PDMA service all the time when other lower priority channel requests are pending. The highest priority channel will be skipped for one block transaction time duration after one block transaction is complete. Then a block transaction requested by the second priority channel will be performed. After a block transaction of the second priority channel is complete, the PDMA arbitration processor will re-check all of the requested channel priority with the exception of the second priority channel since the second priority channel will be excluded after the end of a block transaction. Therefore, a block data transaction of the higher priority channel will be serviced and this channel will be excluded from the priority arbitration at the end of the block transaction. The PDMA will keep transferring the data using the method described above until all of the requested channel data transfer is complete. Refer to the accompanying figure for an example which shows the PDMA channel arbitration and scheduling.

Channel 0: priority=very high, block count=2, block length=2  
Channel 1: priority=high, block count=3, block length=4  
Channel 2: priority=low, block count=3, block length=6  
Priority : CH0 > CH1 > CH2



**Figure 173. 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 69. PDMA Address Modes**

Source Address Mode	Destination Address Mode
Linear Increment / Decrement Address	Linear Increment / Decrement Address
Linear Increment / Decrement Address	Circular Increment / Decrement Address
Linear Increment / Decrement Address	Fixed Address
Circular Increment / Decrement Address	Linear Increment / Decrement Address
Circular Increment / Decrement Address	Circular Increment / Decrement Address
Fixed Address	Linear Increment / Decrement Address
Fixed Address	Fixed Address

### Linear Address Mode

After data is transferred, the current address will be increased or decreased by 1, 2 or 4 depending upon the data bit width setting.

### Circular Address Mode

After data is transferred, the current address will be increased or decreased by 1, 2 or 4 depending upon the data bit width setting. When a block transaction is complete, the current address is loaded with the configured start address.

### Fixed Address Mode

After data is transferred, the current address remains unchanged.

## Auto-Reload

When the auto-reload control bit, AUTORLn, in the PDMA channel n control register PDMACHnCR is set, both the channel n current address and the channel n current transfer size will be automatically reloaded with the corresponding start value after the current PDMA channel data transfer has totally completed. The channel n will still be activated and the next relative PDMA request can be serviced without any re-configuration using the application software.

## Transfer Interrupt

There are five transfer events during which the interrupts can be asserted for each PDMA channel. These are the block transaction end (BE), half-transfer (HT), transfer complete (TC), transfer error (TE) and global transfer event (GE). Setting the corresponding control bits in the PDMA interrupt enable register PDMAIER will enable the relevant interrupt events. The global interrupt event, GE, will be generated if any of the four interrupt events including the BE, HT, TC or TE occurs. Clearing the BE, HT, TC or TE event flags will also clear the GE flag. Clearing the GE flag will automatically clear all other event flags. The TE interrupt event will occur when the PDMA accesses a system reserved address space or when the PDMA receives a request but the corresponding transfer size setting is equal to zero.

## Register Map

The following table shows the PDMA registers and the reset values.

**Table 70. PDMA Register Map**

Register	Offset	Description	Reset Value
<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
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
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
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
PDMACH3TSR	0x058	PDMA Channel 3 Transfer Size Register	0x0000_0000
PDMACH3CTSR	0x05C	PDMA Channel 3 Current Transfer Size Register	0x0000_0000
<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
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
PDMACH5TSR	0x088	PDMA Channel 5 Transfer Size Register	0x0000_0000
PDMACH5CTSR	0x08C	PDMA Channel 5 Current Transfer Size Register	0x0000_0000
<b>PDMA Global Register</b>			
PDMAISR	0x120	PDMA Interrupt Status Register	0x0000_0000
PDMAISCR	0x128	PDMA Interrupt Status Clear Register	0x0000_0000
PDMAIER	0x130	PDMA Interrupt Enable Register	0x0000_0000



## Register Descriptions

### PDMA Channel n Control Register – PDMACHnCR (n = 0 ~ 5)

This register is used to specify the PDMA channel n data transfer configuration.

Offset: 0x000 (0), 0x018 (1), 0x030 (2), 0x048 (3), 0x060 (4), 0x078 (5)

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved				AUTORLn	FIXAENn	CHnPRI	
	7	6	5	4	3	2	1	0
Type/Reset	SRCAMODn	SRCAINCn	DSTAMODn	DSTAINCn	DWIDTHn	SWTRIGn	CHnEN	
	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[11]	AUTORLn	Channel n Auto Reload Enable Control 0: Disable Auto Reload function 1: Enable Auto Reload function  If this bit is set to 1 to enable the auto-reload function, the channel n current address and the channel n current transfer size will be reloaded with the relevant start value and the PDMA channel n will still be activated when a transfer is completed. If this bit is cleared to 0, the channel n current address and the channel n current transfer size will remain unchanged and the PDMA channel n will be disabled after a transfer completion.
[10]	FIXAENn	Channel n Fixed Address Enable control 0: Disable fixed address function in the circular address mode 1: Enable fixed address function in the circular address mode  Note that this bit is only available when the source or destination address mode is set to be in the circular address mode. For example, the source address mode is set as in the linear address mode and the destination address mode is set as in the circular mode. If this bit is set to enable the fixed address function, then the source address mode will still be in the linear address but the destination address mode will be in the fixed address mode instead of the circular address mode.
[9:8]	CHnPRI	Channel n Priority 00: Low 01: Medium 10: High 11: Very high  The CHnPRI field is used to configure the channel priority using the application program. If there are more than one channel which have the same software configured priority level, the channel with the smaller channel number will have priority to transfer one block of data after the arbitration.

Bits	Field	Descriptions
[7]	SRCAMODn	<p>Channel n Source Address Mode selection</p> <p>0: Linear address mode 1: Circular address mode</p> <p>In the linear address mode, the current source address value can be increased or decreased, determined by the SRCAINCn bit value during a complete transfer. In the circular address mode, the current source address value can be increased or decreased which is also determined by the SRCAINCn bit value during a block transfer and will be loaded with the lower 16-bit value of the PDMACHnSADR register, which will be regarded as the current source address when a block transaction has completed.</p>
[6]	SRCAINCn	<p>Channel n Source Address Increment control</p> <p>0: Increment 1: Decrement</p> <p>This bit is used to determine whether the current source address is increased or decreased during a complete transfer in the linear address mode or a block transfer in the circular address mode.</p>
[5]	DSTAMODn	<p>Channel n Destination Address Mode selection</p> <p>0: Linear address mode 1: Circular address mode</p> <p>In linear address mode, the current destination address value can be increased or decreased, determined by the DSTAINCn bit value during a complete transfer. In the circular address mode, the current destination address value can be increased or decreased which is also determined by the DSTAINCn bit value during a block transfer and will be loaded with the lower 16-bit value of the PDMACHnDADR register, which will be regarded as the current destination address when a block transfer has completed.</p>
[4]	DSTAINCn	<p>Channel n Destination Address Increment Control</p> <p>0: Increment 1: Decrement</p> <p>This bit is used to determine if the current destination address is increased or decreased during a complete transfer in the linear address mode or a block transfer in the circular address mode.</p>
[3:2]	DWIDTHn	<p>Data Bit Width selection</p> <p>00: 8-bit 01: 16-bit 10: 32-bit 11: Reserved</p> <p>The field is used to select the data bit width of the corresponding PDMA channel n.</p>
[1]	SWTRIGn	<p>Software Trigger control</p> <p>0: No operation 1: Software triggered transfer request</p> <p>Setting this bit will generate a memory-to-memory software transfer request on the corresponding PDMA channel n. It is automatically cleared when a transfer has completely finished.</p>
[0]	CHnEN	<p>Channel n Enable control</p> <p>0: Disable the PDMA channel n 1: Enable the PDMA channel n</p> <p>Setting this bit will enable a software or hardware transfer request on the PDMA channel n. It is automatically cleared by hardware when a transfer has completed with the auto-reload function being disabled. However, if the AUTORLn bit is set to 1 to enable the auto-reload function, this bit will remain high to enable the PDMA channel n function for the next transfer request instead of automatically being cleared by hardware after a transfer has finished.</p>

## PDMA Channel n Source Address Register – PDMACHnSADR (n = 0 ~ 5)

This register specifies the source address of the PDMA channel n.

Offset: 0x004 (0), 0x01C (1), 0x034 (2), 0x04C (3), 0x064 (4), 0x07C (5)

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	SADRn								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	SADRn								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	SADRn								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	SADRn								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[31:0]	SADRn	Channel n Source Address The register is used to specify the 32-bit source address of the PDMA channel n.

## PDMA Channel n Destination Address Register – PDMACHnDADR (n = 0 ~ 5)

This register specifies the destination address of the PDMA channel n.

Offset: 0x008 (0), 0x020 (1), 0x038 (2), 0x050 (3), 0x068 (4), 0x080 (5)

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	DADRn								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	DADRn								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	DADRn								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	DADRn								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[31:0]	DADRn	Channel n Destination Address The register is used to specify the 32-bit destination address of the PDMA channel n.

## PDMA Channel n Transfer Size Register – PDMACHnTSR (n = 0 ~ 5)

This register is used to specify the block transaction count and block transaction length.

Offset: 0x010 (0), 0x028 (1), 0x040 (2), 0x058 (3), 0x070 (4), 0x088 (5)

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	BLKCNTn								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	BLKCNTn								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	Reservd								
Type/Reset									
	7	6	5	4	3	2	1	0	
	BLKLENn								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[31:16]	BLKCNTn	Channel n Block Transaction Count BLKCNTn represents the number of block transactions for a channel n complete transfer. The capacity of a complete transfer is the product of the BLKCNTn and BLKLENn values. The maximum BLKCNTn value is 65,535.
[7:0]	BLKLENn	Channel n Block Length The BLKLENn represents the length of a data block. The data width is defined by the DWIDTHn field in the PDMACHnCR register. The maximum BLKLENn value is 255.

## PDMA Channel n Current Transfer Size Register – PDMACHnCTSR (n = 0 ~ 5)

This register is used to indicate the current block transaction count.

Offset: 0x014 (0), 0x02C (1), 0x044 (2), 0x05C (3), 0x074 (4), 0x08C (5)

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24								
	CBLKCNTn															
Type/Reset	RO	0	RO	0	RO	0	RO	0	RO	0	RO	0	RO	0	RO	0
	23	22	21	20	19	18	17	16								
	CBLKCNTn															
Type/Reset	RO	0	RO	0	RO	0	RO	0	RO	0	RO	0	RO	0	RO	0
	15	14	13	12	11	10	9	8								
	Reserved															
Type/Reset																
	7	6	5	4	3	2	1	0								
	Reserved															
Type/Reset																

Bits	Field	Descriptions
[31:16]	CBLKCNTn	Channel n Current Block Count The CBLKCNTn field is a 16-bit read-only value indicating the number of data blocks that remain to be transferred. After a data block has transferred completely, the CBLKCNTn value will be decreased by 1. Writing a new value to the BLKCNTn field in the PDMACHnTSR register will update the CBLKCNTn field value.

## PDMA Interrupt Status Register – PDMAISR

This register is used to indicate the corresponding interrupt status of the PDMA channel 0 ~ 5.

Offset: 0x120

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	Reserved		TEISTA5	TCISTA5	HTISTA5	BEISTA5	GEISTA5	TEISTA4	
Type/Reset			RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	
	23	22	21	20	19	18	17	16	
	TCISTA4	HTISTA4	BEISTA4	GEISTA4	TEISTA3	TCISTA3	HTISTA3	BEISTA3	
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	
	15	14	13	12	11	10	9	8	
	GEISTA3	TEISTA2	TCISTA2	HTISTA2	BEISTA2	GEISTA2	TEISTA1	TCISTA1	
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	
	7	6	5	4	3	2	1	0	
	HTISTA1	BEISTA1	GEISTA1	TEISTA0	TCISTA0	HTISTA0	BEISTA0	GEISTA0	
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	

Bits	Field	Descriptions
[29], [24], [19], [14], [9], [4]	TEISTAn	Channel n Transfer Error Interrupt Status (n = 0 ~ 5) 0: No Transfer Error occurs 1: Transfer Error occurs This bit is set by hardware and is cleared by writing a "1" into the corresponding interrupt status clear bit in the PDMAISCR register. A Transfer error will occur when the PDMA accesses a system reserved address space or 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 PDMAISCR register. The Transfer Completion event will occur when the PDMA has completed a data transfer task.
[27], [22], [17], [12], [7], [2]	HTISTAn	Channel n Half Transfer Interrupt Status (n= 0 ~ 5) 0: No Half Transfer Event Occurs 1: Half Transfer Event Occurs This bit is set by hardware and is cleared by writing a "1" into the corresponding interrupt status clear bit in the PDMAISCR register. A Half Transfer event will occur when the PDMA has completed half of the data transfer task.
[26], [21], [16], [11], [6], [1]	BEISTAn	Channel n Block Transaction End Interrupt Status (n= 0 ~ 5) 0: No Block Transaction End Event Occurs 1: Block Transaction End Event Occurs This bit is set by hardware and is cleared by writing a "1" into the corresponding interrupt status clear bit in the PDMAISCR register. A Block Transaction End event will occur when the PDMA completes a data block transaction task.

Bits	Field	Descriptions
[25], [20], [15], [10], [5], [0]	GEISTAn	Channel n Global Transfer Interrupt Status (n= 0 ~ 5) 0: No TE, TC, HT or BE event occurs 1: TE, TC, HT or BE event occurs  This bit is set by hardware and is cleared by writing a "1" into the corresponding interrupt status clear bit, GEICLRn, in the PDMAISR register. A Global Transfer Event will occur if any of the BE, HT, TC or TE events occur. Also clearing any of the BE, HT, TC or TE event interrupt flags will clear the GE interrupt flag. Note that if a "1" is written into the GEICLRn bit in the PDMAISR register to clear the GE interrupt flag, the BE, HT, TC and TE event interrupt flags will also be cleared to 0 together with the GE interrupt status flag.

### PDMA Interrupt Status Clear Register – PDMAISCR

This register is used to clear the corresponding interrupt status bits in the PDMAISR Register.

Offset: 0x128

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	Reserved		TEICLR5	TCICLR5	HTICLR5	BEICLR5	GEICLR5	TEICLR4	
Type/Reset			WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	
	23	22	21	20	19	18	17	16	
	TCICLR4	HTICLR4	BEICLR4	GEICLR4	TEICLR3	TCICLR3	HTICLR3	BEICLR3	
Type/Reset	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	
	15	14	13	12	11	10	9	8	
	GEICLR3	TEICLR2	TCICLR2	HTICLR2	BEICLR2	GEICLR2	TEICLR1	TCICLR1	
Type/Reset	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	
	7	6	5	4	3	2	1	0	
	HTICLR1	BEICLR1	GEICLR1	TEICLR0	TCICLR0	HTICLR0	BEICLR0	GEICLR0	
Type/Reset	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	

Bits	Field	Descriptions
[29], [24], [19], [14], [9], [4]	TEICLRn	Channel n Transfer Error Interrupt Status Clear (n = 0 ~ 5) 0: No Operation 1: Clear the corresponding TEISTAn bit in the PDMAISR register Writing a "1" into the TEICLRn bit will clear the TEISTAn status bit in the 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 PDMAISR 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 PDMAISR 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.

Bits	Field	Descriptions
[26], [21], [16], [11], [6], [1]	BEICLRn	Channel n Block Transaction End Interrupt Status Clear (n = 0 ~ 5) 0: No Operation 1: Clear the corresponding BEISTAn bit in the PDMAISR register Writing a "1" into the BEICLRn bit will clear the BEISTAn status bit in the PDMAISR register. This bit will automatically cleared to 0 after a data "1" is written.
[25], [20], [15], [10], [5], [0]	GEICLRn	Channel n Global Transfer Event Interrupt Status Clear (n = 0 ~ 5) 0: No Operation 1: Clear the corresponding TEISTAn, TCISTAn, HTISTAn, BEISTAn, and GEISTAn bits in the PDMAISR register Writing a "1" into the GEICLRn bit will clear the GEISTAn status bit together with the TEISTAn, TCISTAn, HTISTAn, BEISTAn bits in the PDMAISR register. This bit will be automatically cleared to 0 after a "1" is written.

### PDMA Interrupt Enable Register – PDMAIER

This register is used to enable or disable the related interrupts of the PDMA channel 0 ~ 5.

Offset: 0x130

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	Reserved		TEIE5	TCIE5	HTIE5	BEIE5	GEIE5	TEIE4	
Type/Reset			RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	
	23	22	21	20	19	18	17	16	
	TCIE4	HTIE4	BEIE4	GEIE4	TEIE3	TCIE3	HTIE3	BEIE3	
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	
	15	14	13	12	11	10	9	8	
	GEIE3	TEIE2	TCIE2	HTIE2	BEIE2	GEIE2	TEIE1	TCIE1	
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	
	7	6	5	4	3	2	1	0	
	HTIE1	BEIE1	GEIE1	TEIE0	TCIE0	HTIE0	BEIE0	GEIE0	
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	

Bits	Field	Descriptions
[29], [24], [19], [14], [9], [4]	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.



Bits	Field	Descriptions
[26], [21], [16], [11], [6], [1]	BEIEn	Channel n Block Transaction End Interrupt Enable control (n = 0 ~ 5) 0: Block Transaction End interrupt is disabled 1: Block Transaction End interrupt is enabled This bit is set and cleared by software.
[25], [20], [15], [10], [5], [0]	GEIEn	Channel n Global Transfer Event Interrupt Enable control (n = 0 ~ 5) 0: Global Transfer Event interrupt is disabled 1: Global Transfer Event interrupt is enabled This bit is set and cleared by software.

# 26 External Bus Interface (EBI)

## Introduction

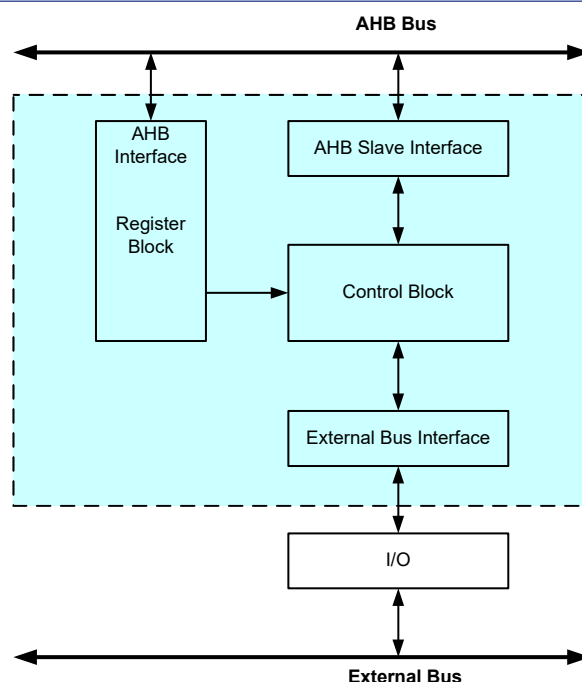
The external bus interface is able to access external parallel interface devices such as SRAM, Flash and LCD modules. The interface is memory mapped into the internal address bus of the CPU. The data and address lines can be multiplexed in order to reduce the number of pins required to connect to external devices. The bus read/write timing can be adjusted to meet the timing specifications of the external devices. Note that the interface only supports asynchronous 8 or 16-bit bus interfaces.

## Features

- Programmable interface for various memory types
  - Asynchronous static random access memory – SRAM
  - Read-only memory – ROM
  - NOR Flash memory
  - 8-bit or 16-bit parallel bus CPU interface device
- Translates AHB transactions into appropriate external device protocol
- Up to 4 memory bank regions and independent chip select control for each memory bank
- Programmable timings to support a wide range of devices
  - Programmable wait states
  - Programmable bus turnaround cycles
  - Programmable output enable and write enable cycles extension
  - Programmable active high or low setting of interface control signal
- Automatic translation when AHB transaction width and external memory interface width are different
- Write buffer to decrease stalling of the AHB write burst transactions
- Supports multiplexed and non-multiplexed address and data line configurations
  - Up to 21 address lines
  - 8-bit or 16-bit data bus width

## Function Descriptions

An overview of the EBI module is shown in Figure 174. 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.

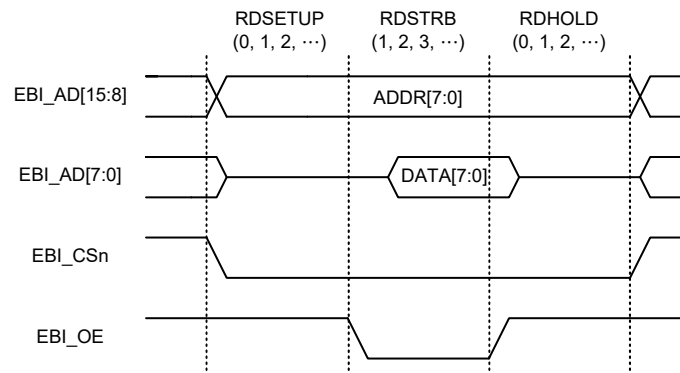


**Figure 174. EBI Block Diagram**

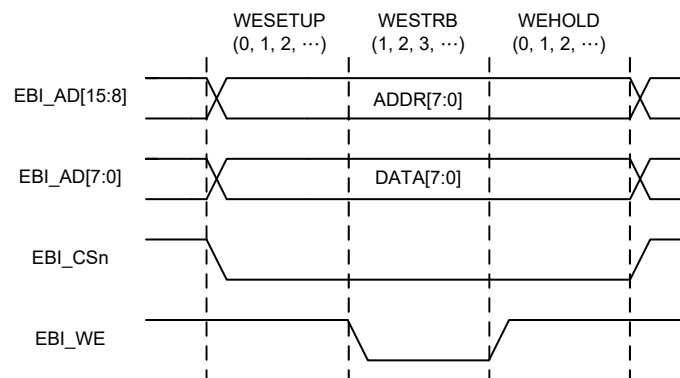
The EBI supports multiplexed and non-multiplexed addressing modes. The non-multiplexed addressing mode can be operated more efficiently and faster but it requires a higher number of pins. The multiplexed addressing mode is slower and requires an external address latch device and a lower number of pins. The functionality of the 16 EBI\_AD pins depends on what kind of the multiplexed addressing mode is used. They are used for both address and data in the multiplexed modes. Also for the non-multiplexed 8-bit address mode, both the address and data fit into these 16 EBI\_AD pins. If more address bits or data bits are needed, an external latch can be used to support up to 20-bit addresses or 16-bit data in the multiplexed addressing modes using only the 16 EBI\_AD pins. Furthermore, independent of the addressing mode, up to 21 non-multiplexed address lines can be enabled on the EBI\_A pin connections. The detailed operation in the supported modes is presented in the following sections. The AHB clock (HCLK) is the reference clock for the EBI.

## Non-Multiplexed 8-Bit Data 8-Bit Address Mode

In this mode, 8-bit address and 8-bit data is supported. The address is located on the higher 8 bits of the EBI\_AD lines and the data uses the lower 8 bits. This mode is set by programming the MODE field in the EBICR register to D8A8. Read and write timing in the 8-bit mode are shown in Figure 175 and Figure 176.



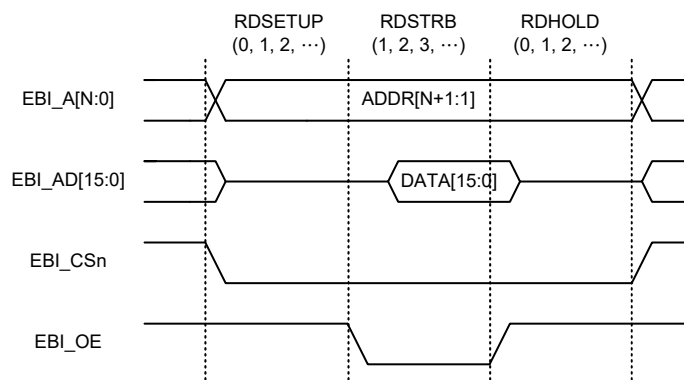
**Figure 175. EBI Non-Multiplexed 8-Bit Data, 8-Bit Address Read Operation**



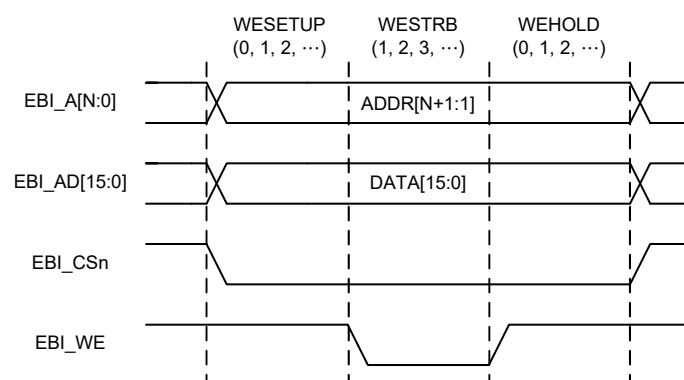
**Figure 176. EBI Non-Multiplexed 8-Bit Data, 8-Bit Address Write Operation**

## 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 177 and Figure 178 for the case in which N address lines on EBI\_A have been enabled.



**Figure 177. EBI Non-Multiplexed 16-Bit Data, N-Bit Address Read Operation**



**Figure 178. EBI Non-Multiplexed 16-Bit Data, N-Bit Address Write Operation**

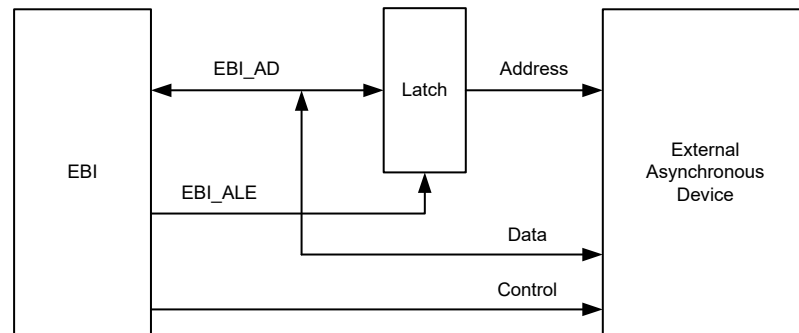
Since the internal AHB address (HADDR) is a byte (8-bit) address whereas the 16-bit width of external device is addressed in words (16-bit), the address actually issued to the external device varies according to the data width as shown in the following table.

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].

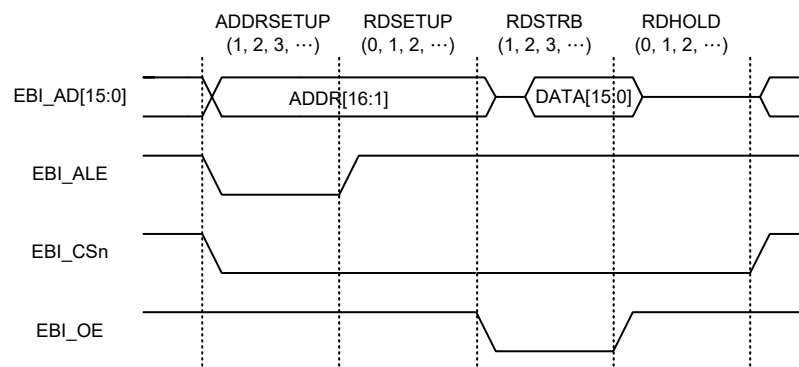
## Multiplexed 16-Bit Data, 16-Bit Address Mode

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 179. This mode is set by programming the MODE field in the EBICR register to D16A16ALE.

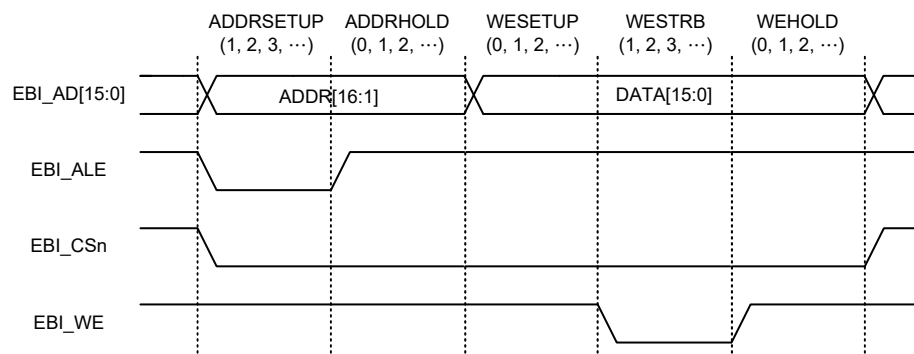


**Figure 179. An EBI Address Latch Setup Diagram**

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 180 and Figure 181.



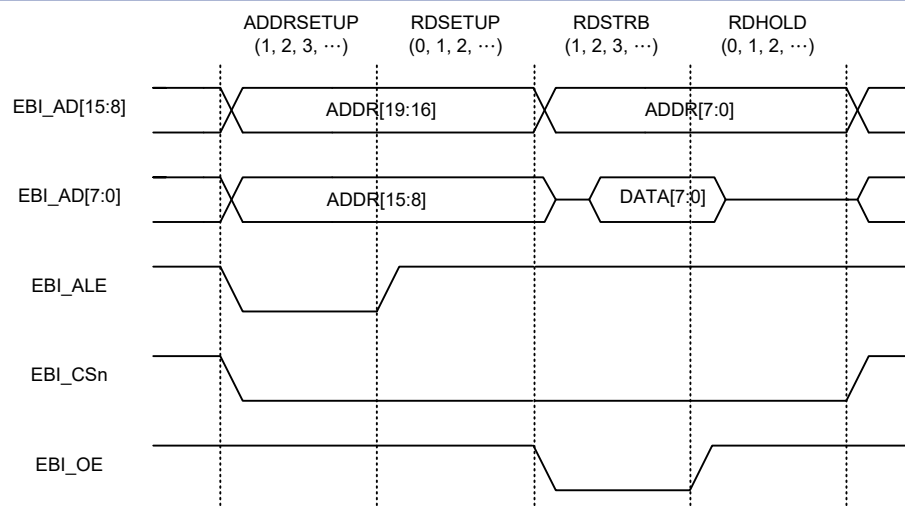
**Figure 180. EBI Multiplexed 16-Bit Data, 16-Bit Address Read Operation**



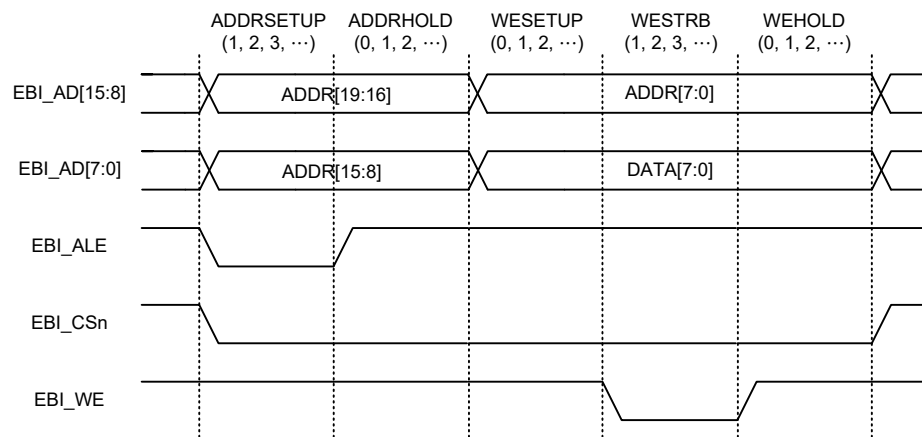
**Figure 181. EBI Multiplexed 16-Bit Data, 16-Bit Address Write Operation**

### Multiplexed 8-Bit Data, 20-Bit Address Mode

This mode allows 20-bit address with 8-bit data multiplexed on the EBI\_AD[15:0] lines to reduce the pins utilization and uses the EBI\_ALE signal to decode 8-bit data and 20-bit address. The upper 8 bits of the EBI\_AD lines (EBI\_AD[15:8]) are consecutively used for the highest 4 bits and the lowest 8 bits of the address. The lower 8 bits of the EBI\_AD lines (EBI\_AD[7:0]) are used for the middle 8 address bits and 8-bit data. This mode is set by programming the MODE field in the EBICR register to D8A24ALE. Read and write signals are shown in Figure 182 and Figure 183 respectively.



**Figure 182. EBI Multiplexed 8-Bit Data, 20-Bit Address Read Operation**



**Figure 183. EBI Multiplexed 8-Bit Data, 20-Bit Address Write Operation**

### Write Buffer and EBI Status

The EBI has a 32-bit wide write buffer. The write buffer can be used to limit stalling of an AHB write burst transaction which comes from the CPU or PDMA to a potentially slow external device.

The EBIBUSY status bit in the EBISR register indicates whether an AHB transaction is still active in the EBI or not. When performing an AHB read or write, the EBIBUSY bit stays 1 until the required transaction(s) with the external device has finished.

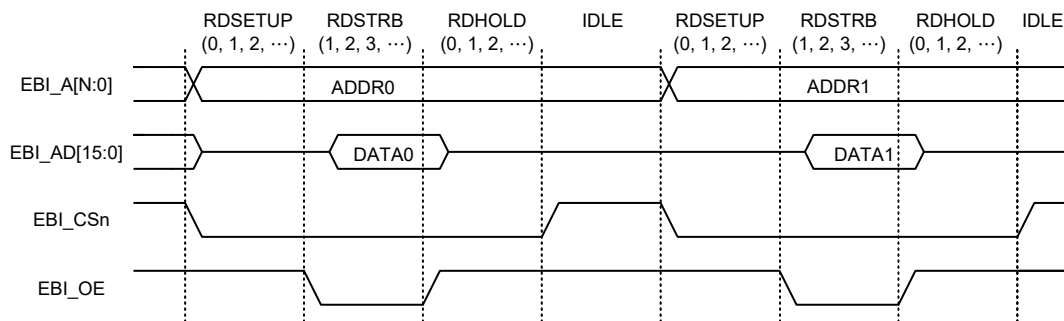
### Bus Turn-around and IDLE Cycles

The EBI\_AD lines can be driven by either the EBI or the external device depending on the cycle state of EBI bus. The RDHOLD timing parameter is for the bus turn-around time and should be programmed to ensure enough time for the characteristics of an external device. The default setting for the EBI is to insert an IDLE cycle between EBI transactions to the same bank. The IDLE cycle insertion is shown for two back-to-back read transactions in Figure 184. For cases where the IDLE state can also provide the required bus turn-around time, the RDHOLD parameter can be programmed to 0. For increased EBI access performance, the automatic IDLE state insertion can be disabled by setting the NOIDLEn (n = 0 ~ 3) bits in the EBICR register to 1. This example is shown in Figure 185 for two back-to-back reads in a non-multiplexed address mode.

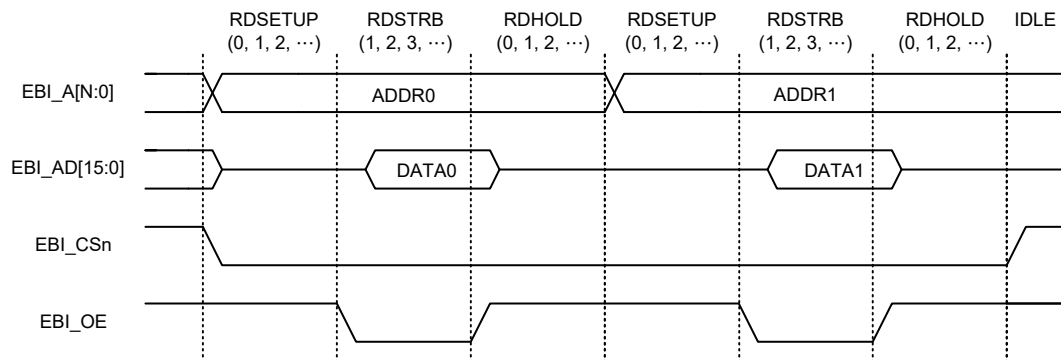
An IDLE cycle will automatically be inserted for the following cases:

- Between two external device transactions to the same bank when the NOIDLE bit is 0.
- Between two external device transactions to different banks.
- Between a read and a subsequent write on the EBI\_AD lines.
- When no request for an external transaction is available in the EBI.





**Figure 184. EBI Inserts an IDLE Cycle Between Transactions in the Same Bank (NOIDLE = 0)**



**Figure 185. EBI De-asserts an IDLE Cycle Between Transactions in the Same Bank (NOIDLE = 1)**

## AHB Transaction Width Conversion

The mapping of AHB transactions to an external device depends on the data width of the external device as the byte lanes of the external device are supported in the device. The Table 84 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 increasing addresses and start with the least significant data from AHB transaction.
- If the AHB master (CPU or PDMA) transaction width is smaller than the external bus transaction width. The EBI will read the data according to the full data bus width of the external device and ignore the superfluous data. For write operation the EBI will automatically perform a read-modify-write sequence to write the data.

**Table 71. EBI Maps AHB Transactions Width to External Device Transactions**

AHB Transaction	8-Bit External Device Transaction	16-Bit External Device Transaction
8-bit read	1 × 8-bit read	1 × 16-bit read (EBI ignore the superfluous data)
16-bit read	2 × 8-bit read	1 × 16-bit read
32-bit read	4 × 8-bit read	2 × 16-bit read
8-bit write	1 × 8-bit write	1 × 16-bit read; 1 × 16-bit write (EBI read-modify-write)
16-bit write	2 × 8-bit write	1 × 16-bit write
32-bit write	4 × 8-bit write	2 × 16-bit write

**Table 72. EBI Maps AHB Transactions Width to External Device Transactions Width**

External Bus Width	Access from AHB Master		Access to External Bus Interface (EBI)		
	Access Type	Address HADDR[1:0] <sup>Note</sup>	Access Split	Output Value from EBI_A[1:0]	Valid Data at EBI_AD[15:0]
8-bit	Byte (8-bit)	b00	No split	b00	EBI_AD[7:0]
		b01	No split	b01	
		b10	No split	b10	
		b11	No split	b11	
	Half-word (16-bit)	b00	1/2 access	b00	
			2/2 access	b01	
	Half-word (16-bit)	b10	1/2 access	b10	
			2/2 access	b11	
	Word (32-bit)	b00	1/4 access	b00	
			2/4 access	b01	
			3/4 access	b10	
			4/4 access	b11	
16-bit	Byte (8-bit)	b00	No split	bx0	EBI_AD[7:0]
		b01	No split	bx0	EBI_AD[15:8]
		b10	No split	bx1	EBI_AD[7:0]
		b11	No split	bx1	EBI_AD[15:8]
	Half-word (16-bit)	b00	No split	bx0	EBI_AD[15:0]
		b10	No split	bx1	EBI_AD[15:0]
	Word (32-bit)	b00	1/2 access	bx0	EBI_AD[15:0]
			2/2 access	bx1	EBI_AD[15:0]

**Note:** HADDR is AHB bus address input.

## EBI Bank Access

The EBI is split into 4 different address regions and each owns an individual EBI\_CS<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. The data space of each bank can be accessed up to 4 MB and is shown as Figure 186.

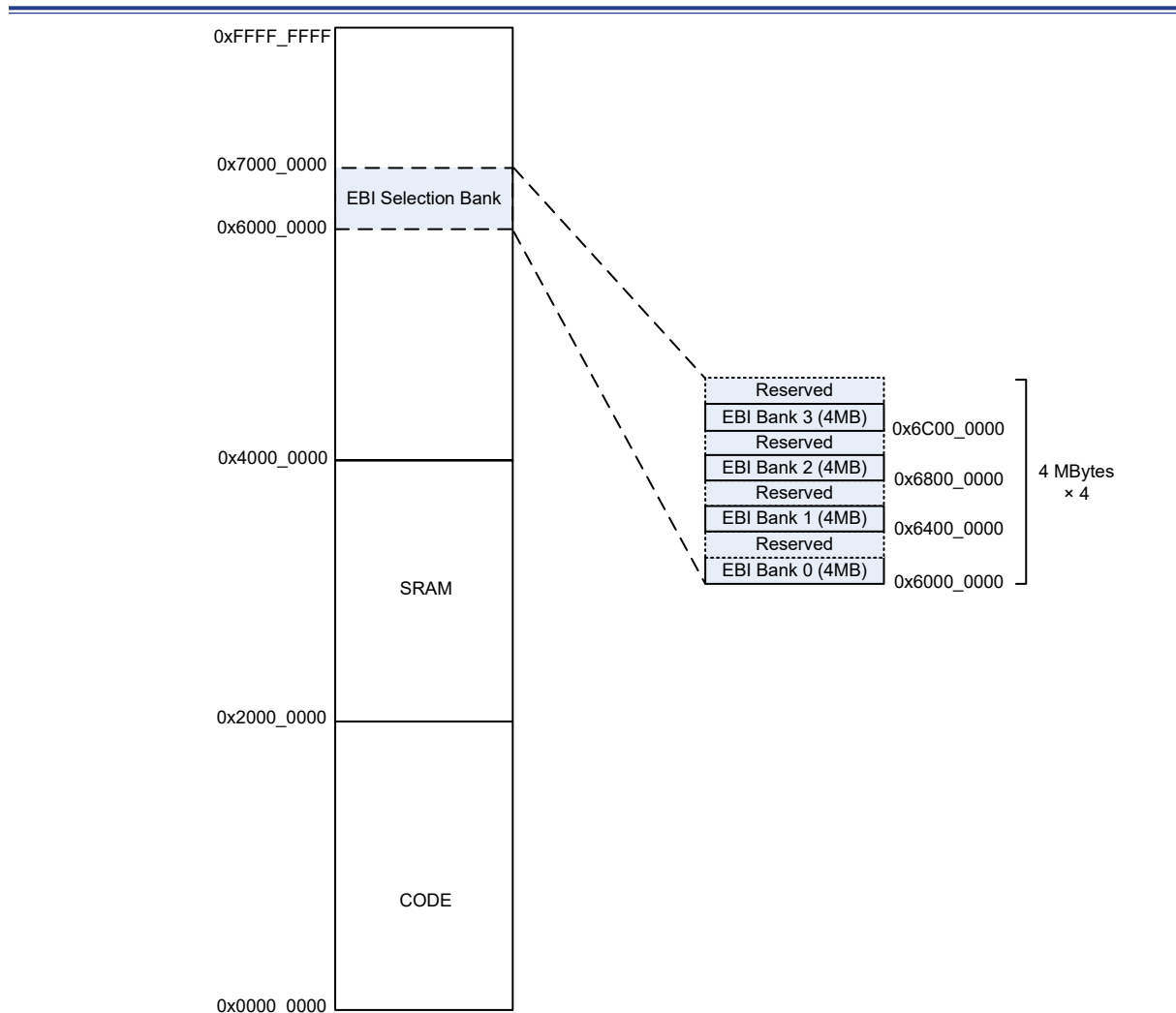


Figure 186. EBI Bank Memory Map

## PDMA Request

The EBI only supports a software trigger for activating a PDMA service.

## Register Map

The following table shows the EBI register and reset value.

**Table 73. EBI Register Map**

Register	Offset	Description	Reset Value
EBICR	0x000	EBI Control Register	0x0000_0000
EBISR	0x008	EBI Status Register	0x0000_0000
EBIATR	0x010	EBI Address Timing Register	0x0000_0707
EBIRTR	0x014	EBI Read Timing Register	0x0007_1F07
EBIWTR	0x018	EBI Write Timing Register	0x0007_1F07
EBIPR	0x01C	EBI Polarity Register	0x0000_0000

## Register Descriptions

### EBI Control Register – EBICR

This register specifies the control setting for EBI bank.

Offset: 0x000

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
	IDLET				Reserved			
Type/Reset	RW	0	RW	0	RW	0	RW	0
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	NOIDLE3	NOIDLE2	NOIDLE1	NOIDLE0	BANKEN3	BANKEN2	BANKEN1	BANKEN0
Type/Reset	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
	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 cleared to 0, one cycle is inserted by the hardware. The cycle unit is based on the HCLK clock period.
[15]	NOIDLE3	No IDLE 3 0: Enable IDLE state insertion 1: Disable IDLE state insertion Enable or disable the insertion of an idle state between transactions for bank 3.
[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.

Bits	Field	Descriptions
[13]	NOIDLE1	No IDLE 1 0: Enable IDLE state insertion 1: Disable IDLE state insertion Enable or disable the insertion of an idle state between transactions for bank 1.
[12]	NOIDLE0	No IDLE 0 0: Enable IDLE state insertion 1: Disable IDLE state insertion Enable or disable the insertion of an idle state between transactions for bank 0.
[11]	BANKEN3	Bank 3 Enable 0: Disable 1: Enable This bit enables or disables bank 3.
[10]	BANKEN2	Bank 2 Enable 0: Disable 1: Enable This bit enables or disables bank 2.
[9]	BANKEN1	Bank 1 Enable 0: Disable 1: Enable This bit enables or disables bank 1.
[8]	BANKEN0	Bank 0 Enable 0: Disable 1: Enable This bit enables or disables bank 0.
[7:6]	MODE3	Set EBI bank 3 access mode 00: D8A8 01: D16A16ALE 10: D8A24ALE 11: D16
[5:4]	MODE2	Set EBI bank 2 access mode 00: D8A8 01: D16A16ALE 10: D8A24ALE 11: D16
[3:2]	MODE1	Set EBI bank 1 access mode 00: D8A8 01: D16A16ALE 10: D8A24ALE 11: D16
[1:0]	MODE0	Set EBI bank 0 access mode 00: D8A8 01: D16A16ALE 10: D8A24ALE 11: D16

This register indicates the EBI status.

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24		
Type/Reset	Reserved									
	23	22	21	20	19	18	17	16		
Type/Reset	Reserved									
	15	14	13	12	11	10	9	8		
Type/Reset	Reserved								EBISMRST	
									RW	0
	7	6	5	4	3	2	1	0		
Type/Reset	Reserved								EBIBUSY	
									RO	0

Bits	Field	Descriptions
[8]	EBISMRST	EBI State Machine Reset 0: Normal 1: Reset EBI state machine Write a “1” to reset the EBI internal state machine to its initial state and keep the original register settings.
[0]	EBIBUSY	EBI BUSY 0: EBI is idle 1: EBI is busy Indicates the EBI is busy with an AHB transaction.

## EBI Address Timing Register – EBIATR

This register specifies the address timing setting for each bank.

Offset: 0x010

Reset value: 0x0000\_0707

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved				ADDRHOLD			
	7	6	5	4	3	2	1	0
Type/Reset	Reserved				ADDRSETUP			
						RW	1	RW
							1	RW
								1

Bits	Field	Descriptions
[10:8]	ADDRHOLD	Address Hold Time Sets the number of cycles for which the address should be held on the EBI_AD bus after the EBI_ALE signal is asserted. This field is allowed to be cleared to 0.
[2:0]	ADDRSETUP	Address Setup Time Sets the number of cycles for which the address is driven onto the EBI_AD bus before the EBI_ALE signal is asserted. If cleared to 0, one cycle is inserted by the hardware. The cycle unit is based on an HCLK clock period.

## EBI Read Timing Register – EBIRTR

This register specifies the read timing setting for each bank

Offset: 0x014

Reset value: 0x0007\_1F07

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved					RDHOLD		
	15	14	13	12	11	10	9	8
Type/Reset	Reserved				RDSTRB			
	7	6	5	4	3	2	1	0
Type/Reset	Reserved					RDSETUP		

Bits	Field	Descriptions
[18:16]	RDHOLD	Read Hold Time Sets the number of cycles for which the EBI_CSn signal is held active after the EBI_OE signal is de-asserted. This interval is used for bus turnaround.
[12:8]	RDSTRB	Read Strobe Time Sets the number of cycles in which the data will be read when the EBI_OE is held active. After the specified number of cycles, the data will be read. If cleared to 0, one cycle is inserted by the hardware. The cycle unit is based on an HCLK clock period.
[2:0]	RDSETUP	Read Setup Time Sets the number of cycles for which the address is setup before the EBI_OE signal is asserted. The cycle unit is based on an HCLK clock period.



## EBI Write Timing Register – EBIWTR

This register specifies the write timing setting for each bank.

Offset: 0x018

Reset value: 0x0007\_1F07

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved					WEHOLD		
	15	14	13	12	11	10	9	8
Type/Reset	Reserved			WESTRB				
	7	6	5	4	3	2	1	0
Type/Reset	Reserved					WESETUP		

Bits	Field	Descriptions
[18:16]	WEHOLD	Write Hold Time Sets the number of cycles for which the EBI_CSn signal is held active after the EBI_WE signal is de-asserted.
[12:8]	WESTRB	Write Strobe Time Sets the number of cycles in which the data will be written when the EBI_WE signal is held active. If cleared to 0, one cycle is inserted by the hardware. The cycle unit is based on an HCLK clock period.
[2:0]	WESETUP	Write Setup Time Sets the number of cycles for which the data is setup before the EBI_WE signal is asserted. The cycle unit is based on an HCLK clock period.

## EBI Polarity Register – EBIPR

This register specifies the polarity of the EBI control signal for each bank.

Offset: 0x01C

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved				ALEPOL	WEPOL	OEPOL	CSPOL
					RW	0	RW	0
						0	RW	0
							0	RW
								0

Bits	Field	Descriptions
[3]	ALEPOL	Address Latch Polarity 0: EBI_ALE is active low 1: EBI_ALE is active high Set the polarity of the EBI_ALE line.
[2]	WEPOL	Write Enable Polarity 0: EBI_WE is active low 1: EBI_WE is active high Set the polarity of the EBI_WE line.
[1]	OEPOL	Output Enable Polarity 0: EBI_OE is active low 1: EBI_OE is active high Set the polarity of the EBI_OE line.
[0]	CSPOL	Chip Selection Polarity 0: EBI_CS is active low 1: EBI_CS is active high Set the polarity of the EBI_CS <sub>n</sub> line.

# 27 Cyclic Redundancy Check (CRC)

## Introduction

The CRC (Cyclic Redundancy Check) calculation unit is an error detection technique test algorithm and is used to verify data transmission or storage data correctness. A CRC calculation takes a data stream or a block of data as input and generates a 16- 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.

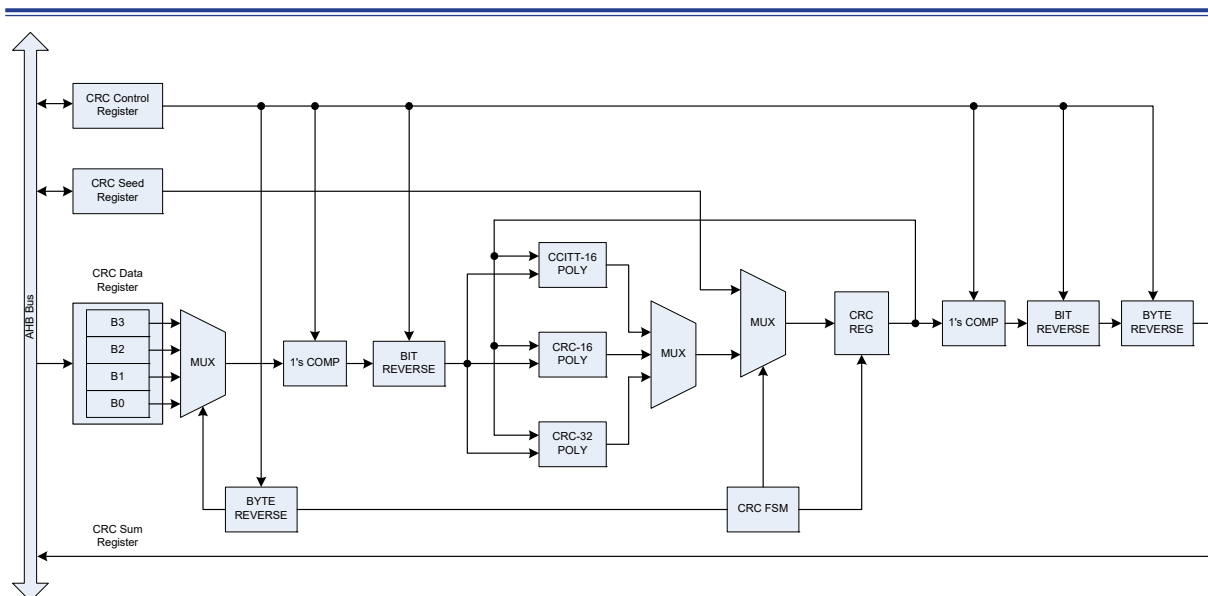


Figure 187. CRC Block Diagram

## Features

- Supports CRC16 polynomial:  $0x8005$ ,  $X^{16} + X^{15} + X^2 + 1$
- Supports CCITT CRC16 polynomial:  $0x1021$ ,  $X^{16} + X^{12} + X^5 + 1$
- Support IEEE-802.3 CRC32 polynomial:  $0x04C11DB7$ ,  $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
- Supports 1's complement, byte reverse & bit reverse operation on data and checksum
- Supports byte, half-word & word data size
- Programmable CRC initial seed value
- CRC computation executed in 1 AHB clock cycle for 8-bit data and 4 AHB clock cycles for 32-bit data
- Supports PDMA to complete a CRC computation of a block of memory

## 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 a 32-bit write CRC data register (CRCDR) and a read CRC checksum register (CRCCSR). The CRCDR register is used to input new data (write access), and the CRCCSR register is used to hold the result of the previous CRC calculation (read access). Each write operation to the CRCDR register creates a combination of the previous CRC value (stored in CRCCSR) and the new one. The CRC block diagram is shown as Figure 187. The CRC unit calculates the CRC data register (CRCDR) value byte by byte and the default byte and bit order is big-endian. The CRCDR register can be written by word, right-aligned half-word and right-aligned byte. For the other registers only 32-bit access is allowed. The duration of the computation depends on data width:

- 4 AHB clock cycles for 32-bit data input
- 2 AHB clock cycles for 16-bit data input
- 1 AHB clock cycle for 8-bit data input

### Byte and Bit Reversal for CRC Computation

The byte reordering and byte-level bit reversal operation can be occurred before the data is used in the CRC calculation or after the CRC checksum output. They are configurable using the corresponding setting field of the CRCCR register. These operations occur on word or half-word 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 188 shows the byte and bit reversal operation example.

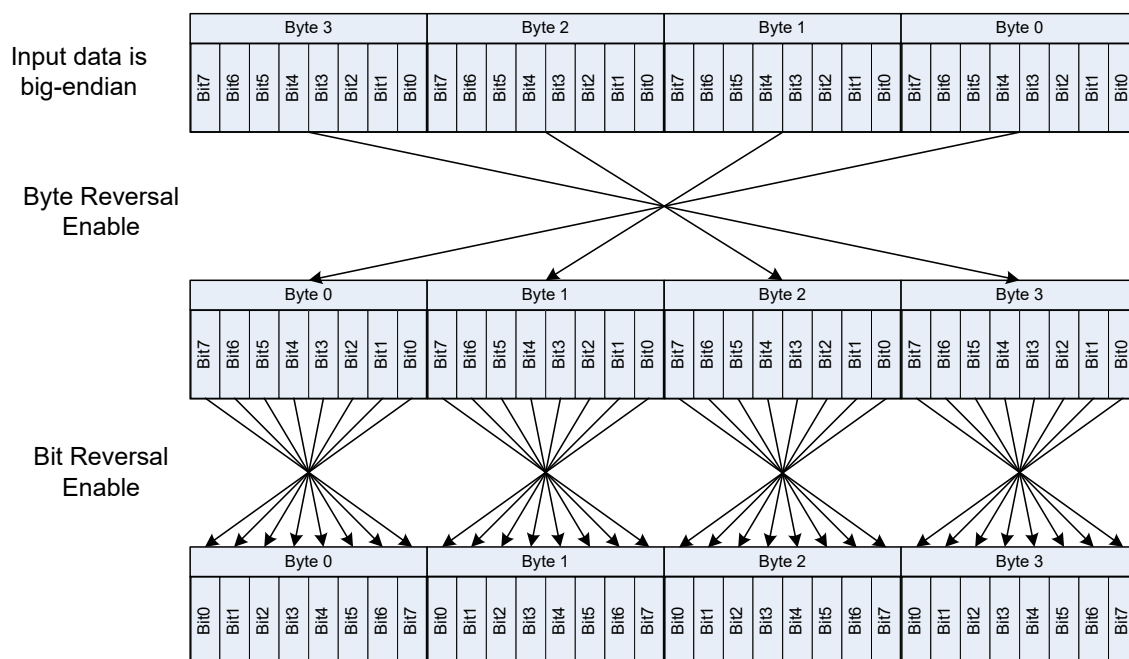


Figure 188. CRC Data Bit and Byte Reversal Example

## 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 use software writes to transfer the other remaining bytes. To write data into the CRC unit, the PDMA should use word access method to transfer data from the source location of memory to the CRC data register (CRCDR) in fixed 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 74. CRC Register Map**

Register	Offset	Description	Reset Value
CRCCR	0x000	CRC Control Register	0x0000_0000
CRCSR	0x004	CRC Seed Register	0x0000_0000
CRCCSR	0x008	CRC Checksum Register	0x0000_0000
CRCDR	0x00C	CRC Data Register	0x0000_0000

## Register Descriptions

### CRC Control Register – CRCCR

This register specifies the corresponding CRC function enable control.

Offset : 0x000

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	SUMCMPL	SUMBYRV	SUMBIRV	DATCMPL	DATBYRV	DATBIRV	POLY	
	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[7]	SUMCMPL	1's Complement operation on Checksum Output 0: Disable 1: Enable
[6]	SUMBYRV	Byte Reverse operation on Checksum Output 0: Disable 1: Enable
[5]	SUMBIRV	Bit Reverse operation on Checksum Output 0: Disable 1: Enable
[4]	DATCMPL	1's Complement operation on Data 0: Disable 1: Enable
[3]	DATBYRV	Byte Reverse operation on Data 0: Disable 1: Enable
[2]	DATBIRV	Bit Reverse operation on Data 0: Disable 1: Enable
[1:0]	POLY	CRC polynomial 00: CRC-CCITT (0x1021) 01: CRC-16 (0x8005) 1X: CRC-32 (0x04C11DB7)

## CRC Seed Register – CRCSDR

This register is used to specify the CRC seed.

Offset : 0x004

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	SEED								
Type/Reset	WO	0	WO	0	WO	0	WO	0	WO
	23	22	21	20	19	18	17	16	
	SEED								
Type/Reset	WO	0	WO	0	WO	0	WO	0	WO
	15	14	13	12	11	10	9	8	
	SEED								
Type/Reset	WO	0	WO	0	WO	0	WO	0	WO
	7	6	5	4	3	2	1	0	
	SEED								
Type/Reset	WO	0	WO	0	WO	0	WO	0	WO

Bits	Field	Descriptions
[31:0]	SEED	CRC Seed Data Put the 16/32-bit seed value in this register according to the polynomial setting in the CRCCR register.

## CRC Checksum Register – CRCCSR

This register contains the CRC checksum output.

Offset : 0x008

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	CHKSUM								
Type/Reset	RO	0	RO	0	RO	0	RO	0	RO
	23	22	21	20	19	18	17	16	
	CHKSUM								
Type/Reset	RO	0	RO	0	RO	0	RO	0	RO
	15	14	13	12	11	10	9	8	
	CHKSUM								
Type/Reset	RO	0	RO	0	RO	0	RO	0	RO
	7	6	5	4	3	2	1	0	
	CHKSUM								
Type/Reset	RO	0	RO	0	RO	0	RO	0	RO

Bits	Field	Descriptions
[31:0]	CHKSUM	CRC Checksum Data Get the CRC 16/32-bit checksum result through this register according to the polynomial setting in the CRCCR register after all data are written to the CRCDR register.

## CRC Data Register – CRCDR

This register is used to specify the CRC input data.

Offset : 0x00C

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	CRCDATA								
Type/Reset	WO	0	WO	0	WO	0	WO	0	WO
	23	22	21	20	19	18	17	16	
	CRCDATA								
Type/Reset	WO	0	WO	0	WO	0	WO	0	WO
	15	14	13	12	11	10	9	8	
	CRCDATA								
Type/Reset	WO	0	WO	0	WO	0	WO	0	WO
	7	6	5	4	3	2	1	0	
	CRCDATA								
Type/Reset	WO	0	WO	0	WO	0	WO	0	WO

Bits	Field	Descriptions
[31:0]	CRCDATA	CRC Input Data Byte, half-word & word write are allowed. 1's complement, byte reverse and bit reverse operation can be applied.



## 28 AES Encrypt/Decrypt Interface (AES)

### Introduction

The AES core supports both encryption and decryption functions and supports 128-bit input data. It should be noted that hardware does not pad out any input data bit, therefore users need to do pad action by software at first.

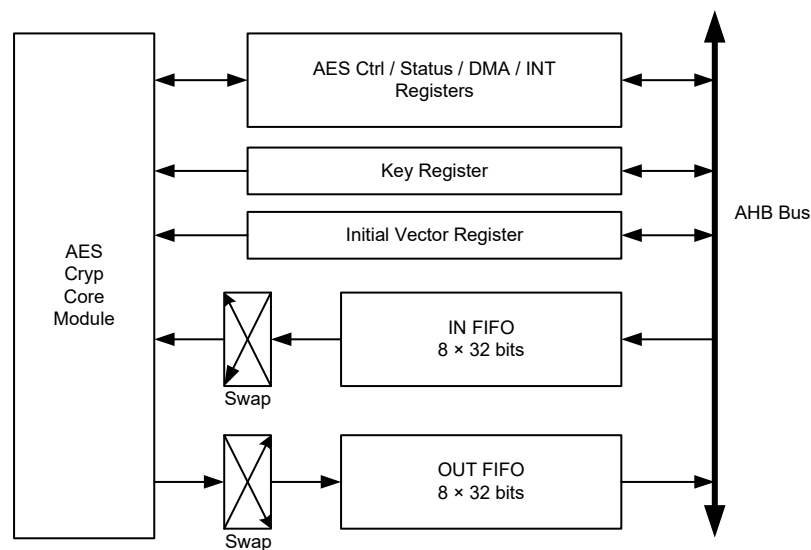


Figure 189. AES Block Diagram

### Features

- Supports AES Encrypt / Decrypt functions
- Supports AES ECB/CBC/CTR modes
- Supports Key Size of 128, 192 and 256 bits
- Supports 4 words Initial Vector for CBC and CTR modes
- 8 × 32 bits (Each IN and OUT FIFO Capacity) for 2 AES Data blocks
- Supports Word Data Swap function
- Supports PDMA Interface

## Functional Descriptions

### AES Mode Description

#### AES Electronic Codebook (AES-ECB) Mode

The 128-bit plaintext data comes from IN FIFO and will be sent to the AES core to do encryption operation after word swapping operation. The AES core uses a 128, 192 or 256-bit key to process the encryption. After encryption, the AES core generates the ciphertext, which will be written into the OUT FIFO after the word swapping operation.

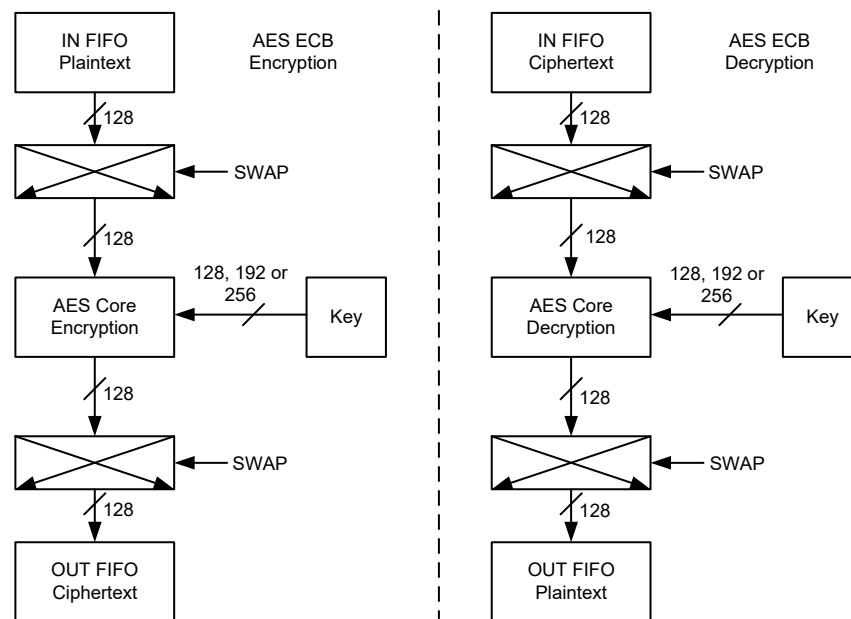


Figure 190. AES-ECB Mode

### AES Cipher Block Chaining (AES-CBC) Mode

During encryption in the CBC mode, each block of plaintext is XORed with the previous ciphertext block before being encrypted. The first initial vectors are initialized in the 1st encryption operation. The plaintext after word swapping will be XORed with the initial vectors before encryption. When the encryption output data is pushed into the OUT FIFO, the initial vectors are updated by the encryption output data at the same time.

During decryption in the CBC mode, each block of plaintext is XORed with the previous ciphertext block after being decrypted. The first initial vectors are initialized in the 1st decryption operation. The ciphertext after word swapping and decryption will be XORed with the initial vectors. When the XORed decryption output data is pushed into the OUT FIFO, the initial vectors are updated by the decryption input data at the same time for the next round ciphertext.

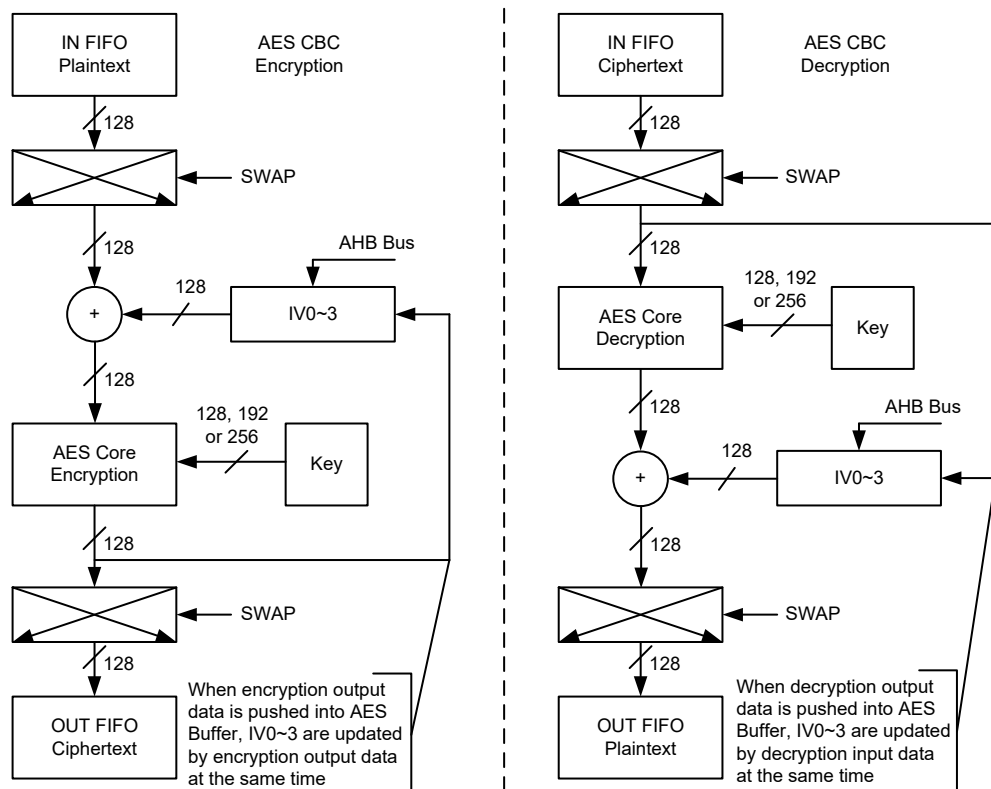


Figure 191. AES-CBC Mode

### AES Counter (AES-CTR) Mode

In the CTR mode, the initial vector counter value, after being increased by one, will be sent to the AES core for encryption to generate ciphertext. The AES core uses the same AES direction setting in both encryption and decryption.

During encryption and decryption in the CTR mode, the IN FIFO data after word swapping is XORed with the ciphertext. The XORed data is sent to the OUT FIFO after word swapping. The initial vector counter will be increased by one at the same time for the next round ciphertext.

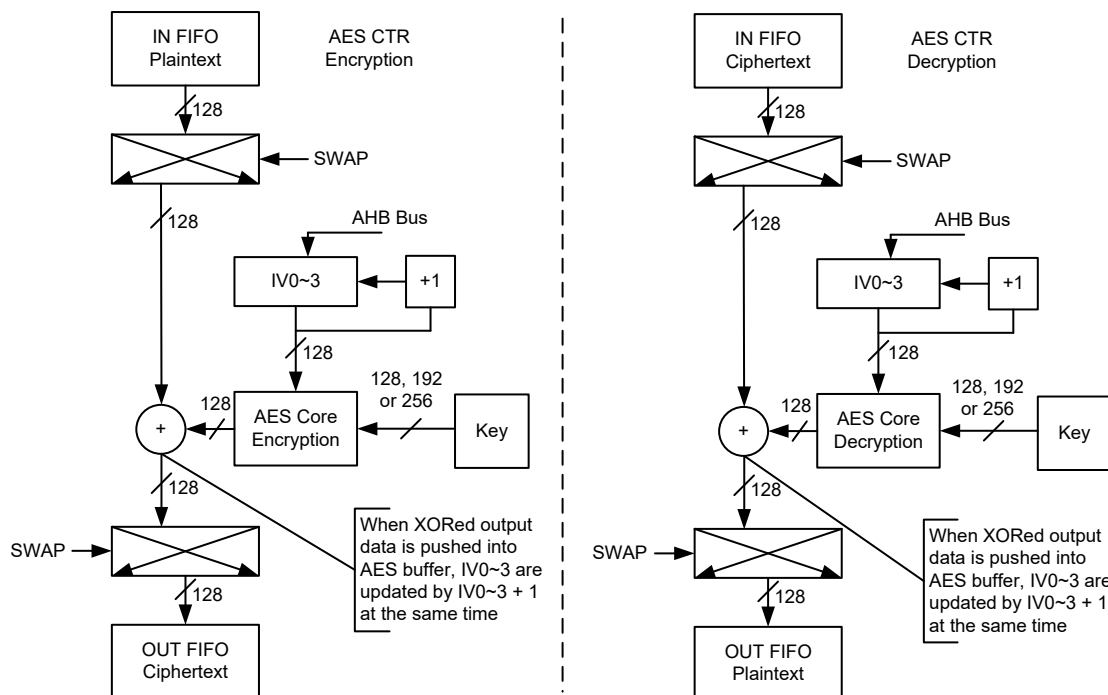


Figure 192. AES-CTR Mode

### AES Status

There are five status conditions in the AES for the user to monitor the AES situation. The IFEMPTY bit will be set when the input FIFO is empty while the IFNFULL bit will be set when the input FIFO is not full. The OFNEMPTY bit will be set when there is data in the output FIFO. The OFFULL bit will be set when the output FIFO is full. The BUSY bit will be set when the AES core is executing an encryption/decryption operation or the key is in expansion state.

### AES PDMA Interface

The AES supports the 32 bits PDMA data transfer. When the IN FIFO is empty, the AES will send an IN FIFO request to the PDMA. When the OUT FIFO is full, the AES will send an OUT FIFO request to the PDMA.

## AES Interrupt

The IFINT request will be generated when the input FIFO is less than or equal to 1 AES block ( $4 \times 32$  bits). The OFINT request will be generated when there is data in the AES buffer. When the IFINT bit is set high, an AES interrupt will be generated if the IFINT interrupt is enabled and the AES remains enabled. When the OFINT bit is set high, an AES interrupt can also be generated if the OFINT interrupt is enabled irrespective of the AES enable/disable condition.

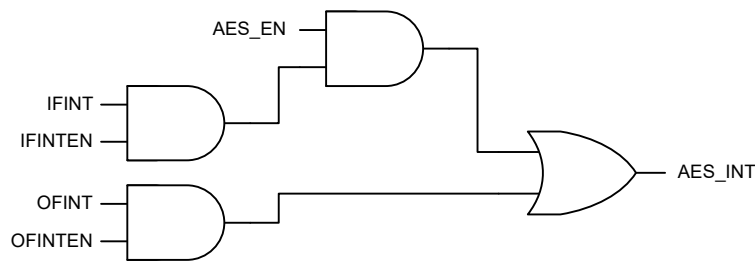


Figure 193. AES Interrupt

## AES Initial Vector

The initial vectors (IV0 ~ 3) are not used in the ECB mode. The initial vectors are initialized in the first block of AES input data in the CBC and CTR modes. After the first AES block of input data, the values of the initial vectors will be updated by hardware automatically for the next block of AES input data. The initial vectors in the CTR mode contain nonce, initial vector and counter. The counter will be increased by 1 after every AES data block action.

Nonce ( 32 bits )	IV0
Initial Vector ( 64 bits )	IV1, 2
Counter ( 32 bits )	IV3

Figure 194. Initial Vector for CTR Mode

## AES Word Swap

The AES supports a word swap function. The swap action is performed between IN FIFO and AES block data, it is also executed between the AES block data and OUT FIFO. If the word swap function is required, the SWAP bit in the AESCR register should be set high.

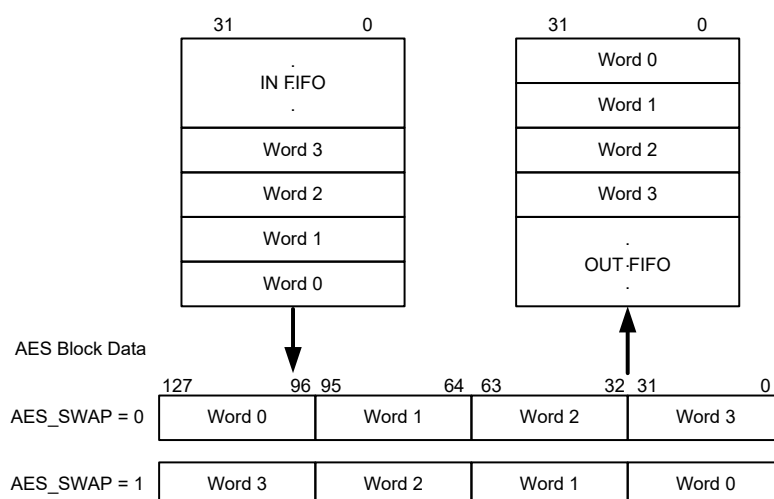


Figure 195. AES Word Swap Function

## Register Map

The following table shows the AES registers and reset values.

Table 75. AES Register Map

Register	Offset	Description	Reset Value
AESCR	0x000	AES Control Register	0x0000_0000
AESSR	0x004	AES Status Register	0x0000_0003
AESDMAR	0x008	AES DMA Register	0x0000_0000
AESISR	0x00C	AES Interrupt Status Register	0x0000_0001
AESIER	0x010	AES Interrupt Enable Register	0x0000_0000
AESDINR	0x014	AES Data Input Register	0x0000_0000
AESDOUTR	0x018	AES Data Output Register	0x0000_0000
AESKEY0	0x01C	AES Key Register 0	0x0000_0000
AESKEY1	0x020	AES Key Register 1	0x0000_0000
AESKEY2	0x024	AES Key Register 2	0x0000_0000
AESKEY3	0x028	AES Key Register 3	0x0000_0000
AESKEY4	0x02C	AES Key Register 4	0x0000_0000
AESKEY5	0x030	AES Key Register 5	0x0000_0000
AESKEY6	0x034	AES Key Register 6	0x0000_0000
AESKEY7	0x038	AES Key Register 7	0x0000_0000
AESIVR0	0x03C	AES Initial Vector Register 0	0x0000_0000
AESIVR1	0x040	AES Initial Vector Register 1	0x0000_0000
AESIVR2	0x044	AES Initial Vector Register 2	0x0000_0000
AESIVR3	0x048	AES Initial Vector Register 3	0x0000_0000

## Register Descriptions

### AES Control Register – AESCR

This register specifies the AES control setting.

Offset: 0x000

Reset value: 0x0000\_0200

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved					FFLUSH	ENDIAN	SWAP
	7	6	5	4	3	2	1	0
Type/Reset	Reserved	KEYSIZE	KEYSTART	MODE		DIR	AESEN	
	RW	0 RW	0 RW	0 RW	0 RW	0 RW	0 RW	0

Bits	Field	Descriptions
[10]	FFLUSH	AES IN/OUT FIFO Flush 0: No action 1: Flush FIFO The bit is cleared to 0 by hardware automatically. The bit can be set only in the AES disable state.
[9]	ENDIAN	Endian Selection 0: Big-endian 1: Little-endian This setting will apply to IN/OUT FIFO data, KEY and IV.
[8]	SWAP	AES Data Swap Function 0: No Swap 1: Word Swap This setting will apply to IN/OUT FIFO data.
[6:5]	KEYSIZE	AES Key Size 00: 128 bits 01: 192 bits 1x: 256 bits
[4]	KEYSTART	AES Key Start 0: Key doesn't Start 1: Key Start It is cleared to 0 by hardware automatically. The bit works when in the AES enable state.
[3:2]	MODE	AES Function Mode 00: ECB mode 01: CBC mode 1x: CTR mode

Bits	Field	Descriptions
[1]	DIR	AES Direction 0: Encryption 1: Decryption
[0]	AESEN	AES Enable 0: AES is disabled 1: AES is enabled

## AES Status Register – AESSR

This register specifies the AES status.

Offset: 0x004

Reset value: 0x0000\_0003

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved		BUSY		OFFULL	OFNEMPTY	IFNFULL	IFEMPTY
			RO		0	RO	0	RO
							1	RO
								1

Bits	Field	Descriptions
[4]	BUSY	Busy bit 0: AES is not busy 1: AES is busy AES is busy when AES is executing the encryption/decryption operation or the key is in expansion state.
[3]	OFFULL	Output FIFO is Full 0: Output FIFO is not full 1: Output FIFO is full
[2]	OFNEMPTY	Output FIFO is not Empty 0: Output FIFO is empty 1: Output FIFO is not empty
[1]	IFNFULL	Input FIFO is not Full 0: Input FIFO is full 1: Input FIFO is not full
[0]	IFEMPTY	Input FIFO is Empty 0: Input FIFO is not empty 1: Input FIFO is empty



## AES DMA Register – AESDMAR

This register specifies the DMA setting.

Offset: 0x008

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved						OFDMAEN	IFDMAEN
							RW	0 RW 0

Bits	Field	Descriptions
[1]	OFDMAEN	Output FIFO DMA Enable 0: DMA is disabled 1: DMA is enabled
[0]	IFDMAEN	Input FIFO DMA Enable 0: DMA is disabled 1: DMA is enabled

## AES Interrupt Status Register – AESISR

The register specifies the interrupt status setting.

Offset: 0x00C

Reset value: 0x0000\_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							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved						OFINT	IFINT
							RO	0 RO 1

Bits	Field	Descriptions
[1]	OFINT	Output FIFO Interrupt Status 0: No Output FIFO Interrupt 1: Output FIFO Interrupt
[0]	IFINT	Input FIFO interrupt Status 0: No Input FIFO Interrupt 1: Input FIFO Interrupt

## AES Interrupt Enable Register – AESIER

The register specifies the interrupt enable setting.

Offset: 0x010

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved						OFINTEN	IFINTEN
							RW	0 RW 0

Bits	Field	Descriptions
[1]	OFINTEN	Output FIFO Interrupt Enable bit 0: Interrupt is disabled 1: Interrupt is enabled
[0]	IFINTEN	Input FIFO Interrupt Enable bit 0: Interrupt is disabled 1: Interrupt is enabled

## AES DATA Input Register – AESDINR

The register specifies the data input setting.

Offset: 0x014

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	DIN								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	DIN								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	DIN								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	DIN								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[31:0]	DIN	AES DATA Input 0x0000_0000 ~ 0xFFFF_FFFF

## AES DATA Output Register – AESDOUTR

The register specifies the data output setting.

Offset: 0x018

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	DOUT								
Type/Reset	RO	0	RO	0	RO	0	RO	0	RO
	23	22	21	20	19	18	17	16	
	DOUT								
Type/Reset	RO	0	RO	0	RO	0	RO	0	RO
	15	14	13	12	11	10	9	8	
	DOUT								
Type/Reset	RO	0	RO	0	RO	0	RO	0	RO
	7	6	5	4	3	2	1	0	
	DOUT								
Type/Reset	RO	0	RO	0	RO	0	RO	0	RO

Bits	Field	Descriptions
[31:0]	DOUT	AES Data Output 0x0000_0000 ~ 0xFFFF_FFFF

## AES Key Register n – AESKEYRn, n = 0 ~ 7

The register specifies the data of Key data n.

Offset: 0x01C ~ 0x038

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	KeyData								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	KeyData								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	KeyData								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	KeyData								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[31:0]	KeyData	KeyData 0x0000_0000 ~ 0xFFFF_FFFF

## AES Initial Vector Register n – AESIVRn, n = 0 ~ 3

The register specifies the data of Initial Vector data n.

Offset: 0x03C ~ 0x048

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	IVData								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	IVData								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	15	14	13	12	11	10	9	8	
	IVData								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
	IVData								
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[31:0]	IVData	Initial Vector Data 0x0000_0000 ~ 0xFFFF_FFFF

Copyright© 2021 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>.