



**32-Bit Arm® Cortex®-M0+ 5V Touch Microcontroller**

# **HT32F54231/HT32F54241 HT32F54243/HT32F54253 User Manual**

Revision: V1.20    Date: April 13, 2023

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

## Table of Contents

<b>1 Introduction .....</b>	<b>25</b>
Overview .....	25
Features .....	25
Device Information .....	30
Block Diagram .....	31
<b>2 Document Conventions .....</b>	<b>33</b>
<b>3 System Architecture .....</b>	<b>34</b>
Arm® Cortex®-M0+ Processor .....	34
Bus Architecture .....	35
Memory Organization .....	36
Memory Map .....	37
Embedded Flash Memory .....	42
Embedded SRAM Memory .....	42
AHB Peripherals .....	42
APB Peripherals .....	42
<b>4 Flash Memory Controller (FMC) .....</b>	<b>43</b>
Introduction .....	43
Features .....	43
Functional Descriptions .....	44
Flash Memory Map .....	44
Flash Memory Architecture .....	45
Wait State Setting .....	45
Bootling Configuration .....	46
Page Erase .....	46
Mass Erase .....	48
Word Programming .....	49
Option Byte Description .....	50
Page Erase/Program Protection .....	51
Security Protection .....	52
Register Map .....	53
Register Descriptions .....	54
Flash Target Address Register – TADR .....	54
Flash Write Data Register – WRDR .....	55
Flash Operation Command Register – OCMR .....	56
Flash Operation Control Register – OPCR .....	57
Flash Operation Interrupt Enable Register – OIER .....	58
Flash Operation Interrupt and Status Register – OISR .....	59
Flash Page Erase/Program Protection Status Register – PPSR .....	61
Flash Security Protection Status Register – CPSR .....	62

Flash Vector Mapping Control Register – VMCR .....	63
Flash Manufacturer and Device ID Register – MDID.....	64
Flash Page Number Status Register – PNSR .....	65
Flash Page Size Status Register – PSSR .....	66
Device ID Register – DIDR.....	66
Flash Pre-fetch Control Register – CFCR .....	67
Custom ID Register n – CIDRn (n = 0 ~ 3).....	68
<b>5 Power Control Unit (PWRCU) .....</b>	<b>69</b>
Introduction .....	69
Features.....	69
Functional Descriptions .....	70
V <sub>DD</sub> Power Domain.....	70
V <sub>CORE</sub> Power Domain.....	71
Operation Modes .....	72
Register Map .....	74
Register Descriptions.....	75
Power Control Status Register – PWRSR .....	75
Power Control Register – PWRCR.....	76
Low Voltage / Brown Out Detect Control and Status Register – LVDCSR .....	78
Power Control LDO Status Register – PWRLDOSR .....	79
<b>6 Clock Control Unit (CKCU) .....</b>	<b>80</b>
Introduction .....	80
Features.....	82
Functional Descriptions .....	82
High Speed External Crystal Oscillator – HSE .....	82
High Speed Internal RC Oscillator – HSI.....	83
Auto Trimming of High Speed Internal RC Oscillator – HSI.....	83
Phase Locked Loop – PLL.....	84
Low Speed External Crystal Oscillator – LSE.....	86
Low Speed Internal RC Oscillator – LSI .....	86
Clock Ready Flag .....	86
System Clock (CK_SYS) Selection .....	86
HSE Clock Monitor .....	87
Clock Output Capability .....	87
Register Map .....	88
Register Descriptions.....	89
Global Clock Configuration Register – GCFGR.....	89
Global Clock Control Register – GCCR.....	90
Global Clock Status Register – GCSR .....	91
Global Clock Interrupt Register – GCIR.....	92
PLL Configuration Register – PLLCFGR .....	93
PLL Control Register – PLLCR.....	94

AHB Configuration Register – AHBCFGR .....	95
AHB Clock Control Register – AHBCCR .....	96
APB Configuration Register – APBCFGR.....	98
APB Clock Control Register 0 – APBCCR0.....	99
APB Clock Control Register 1 – APBCCR1.....	101
Clock Source Status Register – CKST .....	103
APB Peripheral Clock Selection Register 0 – APBPCSR0.....	104
APB Peripheral Clock Selection Register 1 – APBPCSR1 .....	106
HSI Control Register – HSICR.....	108
HSI Auto Trimming Counter Register – HSIATCR .....	109
APB Peripheral Clock Selection Register 2 – APBPCSR2.....	110
MCU Debug Control Register – MCUDBGCR.....	111
<b>7 Reset Control Unit (RSTCU) .....</b>	<b>114</b>
Introduction .....	114
Functional Descriptions .....	114
Power-On Reset .....	114
System Reset .....	115
AHB and APB Unit Reset.....	115
Register Map .....	115
Register Descriptions.....	116
Global Reset Status Register – GRSR .....	116
AHB Peripheral Reset Register – AHBPRSTR.....	117
APB Peripheral Reset Register 0 – APBPRSTR0 .....	118
APB Peripheral Reset Register 1 – APBPRSTR1 .....	120
<b>8 General Purpose I/O (GPIO).....</b>	<b>122</b>
Introduction .....	122
Features.....	123
Functional Descriptions .....	123
Default GPIO Pin Configuration.....	123
General Purpose I/O – GPIO .....	123
GPIO Locking Mechanism .....	125
Register Map .....	125
Register Descriptions.....	127
Port A Data Direction Control Register – PADIRCR .....	127
Port A Input Function Enable Control Register – PAINER .....	128
Port A Pull-Up Selection Register – PAPUR .....	129
Port A Pull-Down Selection Register – PAPDR .....	130
Port A Open-Drain Selection Register – PAODR.....	131
Port A Drive Current Selection Register – PADRVR.....	132
Port A Lock Register – PALOCKR .....	133
Port A Data Input Register – PADINR.....	134
Port A Output Data Register – PADOUTR .....	134
Port A Output Set/Reset Control Register – PASRR .....	135

Port A Output Reset Register – PARR.....	136
Port B Data Direction Control Register – PBDIRCR.....	136
Port B Input Function Enable Control Register – PBINER .....	137
Port B Pull-Up Selection Register – PBPUR .....	138
Port B Pull-Down Selection Register – PBPDR.....	139
Port B Open-Drain Selection Register – PBODR .....	140
Port B Drive Current Selection Register – PBDRVR .....	141
Port B Lock Register – PBLOCKR.....	142
Port B Data Input Register – PBDINR .....	143
Port B Output Data Register – PBDOUTR .....	143
Port B Output Set/Reset Control Register – PBSRR.....	144
Port B Output Reset Register – PBRR .....	145
Port C Data Direction Control Register – PCDIRCR .....	146
Port C Input Function Enable Control Register – PCINER .....	147
Port C Pull-Up Selection Register – PCPUR .....	148
Port C Pull-Down Selection Register – PCPDR .....	149
Port C Open-Drain Selection Register – PCODR.....	150
Port C Drive Current Selection Register – PCDRVR .....	151
Port C Lock Register – PCLOCKR .....	152
Port C Data Input Register – PCDINR.....	153
Port C Output Data Register – PCDOUTR .....	154
Port C Output Set/Reset Control Register – PCSRR .....	155
Port C Output Reset Register – PCRR.....	156
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
<b>9 Alternate Function Input / Output Control Unit (AFIO).....</b>	<b>167</b>
Introduction .....	167
Features.....	168
Functional Descriptions .....	168
External Interrupt Pin Selection .....	168
Alternate Function.....	169
Lock Mechanism .....	169
Register Map .....	169
Register Descriptions.....	170
EXTI Source Selection Register 0 – ESSR0 .....	170

EXTI Source Selection Register 1 – ESSR1 .....	171
GPIO Port x Configuration Low Register – GPxCFGLR, x = A, B, C, D .....	172
GPIO Port x Configuration High Register – GPxCFGHR, x = A, B, C, D.....	173
<b>10 Nested Vectored Interrupt Controller (NVIC).....</b>	<b>174</b>
Introduction .....	174
Features.....	175
Functional Descriptions .....	176
SysTick Calibration .....	176
Register Map .....	176
<b>11 External Interrupt/Event Controller (EXTI).....</b>	<b>177</b>
Introduction .....	177
Features.....	177
Functional Descriptions .....	178
Wakeup Event Management.....	178
External Interrupt/Event Line Mapping .....	178
Interrupt and Debounce .....	179
Register Map .....	180
Register Descriptions.....	181
EXTI Interrupt n Configuration Register – EXTICFGRn, n = 0 ~ 15 .....	181
EXTI Interrupt Control Register – EXTICR .....	182
EXTI Interrupt Edge Flag Register – EXTIEDGEFLGR.....	183
EXTI Interrupt Edge Status Register – EXTIEDGESR .....	184
EXTI Interrupt Software Set Command Register – EXTISSCR.....	185
EXTI Interrupt Wakeup Control Register – EXTIWAKUPCR .....	186
EXTI Interrupt Wakeup Polarity Register – EXTIWAKUPPOLR .....	187
EXTI Interrupt Wakeup Flag Register – EXTIWAKUPFLG .....	188
<b>12 Analog to Digital Converter (ADC) .....</b>	<b>189</b>
Introduction .....	189
Features.....	190
Functional Descriptions .....	191
ADC Clock Setup.....	191
Channel Selection.....	191
Conversion Mode.....	191
Start Conversion on External Event.....	194
Sampling Time Setting.....	195
Data Format.....	195
Analog Watchdog.....	195
Interrupts.....	196
PDMA Request (HT32F54243/HT32F54253 only) .....	196
Voltage Reference Generator .....	196
V <sub>DDA</sub> Voltage Monitor .....	197

Register Map .....	197
Register Descriptions.....	198
ADC Conversion Control Register – ADCCR .....	198
ADC Conversion List Register 0 – ADCLST0 .....	200
ADC Conversion List Register 1 – ADCLST1 .....	201
ADC Input Sampling Time Register – ADCSTR .....	202
ADC Conversion Data Register y – ADCDRy, y = 0 ~ 7 .....	203
ADC Trigger Control Register – ADCTCR .....	204
ADC Trigger Source Register – ADCTSR.....	205
ADC Watchdog Control Register – ADCWCR .....	206
ADC Watchdog Threshold Register – ADCTR.....	207
ADC Interrupt Enable Register – ADCIER.....	208
ADC Interrupt Raw Status Register – ADCIRAW .....	209
ADC Interrupt Status Register – ADCISR.....	210
ADC Interrupt Clear Register – ADCICLR .....	211
ADC PDMA Request Register – ADCDMAR .....	212
Voltage Reference Control Register – VREFCR .....	213
Voltage Reference Value Register – VREFVALR .....	214
<b>13 Comparator (CMP).....</b>	<b>215</b>
Introduction .....	215
Features.....	215
Functional Descriptions .....	216
Comparator Inputs and Output .....	216
Comparator Voltage Reference .....	216
Interrupts and Wakeup.....	217
Power Mode and Hysteresis.....	218
Comparator Write-Protected Mechanism .....	218
Register Map .....	218
Register Descriptions.....	219
Comparator Control Register n – CMPCRn, n = 0 or 1 .....	219
Comparator Voltage Reference Value Register n – CVRVALRn, n = 0 or 1 .....	221
Comparator Interrupt Enable Register n – CMPIERn, n = 0 or 1 .....	222
Comparator Transition Flag Register n – CMPTFRn, n = 0 or 1 .....	223
<b>14 General-Purpose Timer (GPTM) .....</b>	<b>224</b>
Introduction .....	224
Features.....	225
Functional Descriptions .....	225
Counter Mode .....	225
Clock Controller .....	227
Trigger Controller.....	229
Slave Controller .....	230
Master Controller .....	232
Channel Controller.....	233

Input Stage .....	235
Quadrature Decoder .....	237
Output Stage.....	238
Update Management .....	242
Single Pulse Mode.....	243
Asymmetric PWM Mode .....	245
Timer Interconnection .....	245
Trigger Peripherals Start.....	248
PDMA Request (HT32F54243/HT32F54253 only) .....	248
Register Map .....	249
Register Descriptions.....	250
Timer Counter Configuration Register – CNTCFR .....	250
Timer Mode Configuration Register – MDCFR.....	251
Timer Trigger Configuration Register – TRCFR.....	254
Timer Control Register – CTR .....	255
Channel 0 Input Configuration Register – CH0ICFR .....	256
Channel 1 Input Configuration Register – CH1ICFR .....	257
Channel 2 Input Configuration Register – CH2ICFR .....	259
Channel 3 Input Configuration Register – CH3ICFR .....	260
Channel 0 Output Configuration Register – CH0OCFR .....	262
Channel 1 Output Configuration Register – CH1OCFR .....	264
Channel 2 Output Configuration Register – CH2OCFR .....	266
Channel 3 Output Configuration Register – CH3OCFR .....	268
Channel Control Register – CHCTR.....	270
Channel Polarity Configuration Register – CHPOLR.....	271
Timer PDMA/Interrupt Control Register – DICTR.....	272
Timer Event Generator Register – EVGR.....	273
Timer Interrupt Status Register – INTSR.....	275
Timer Counter Register – CNTR.....	277
Timer Prescaler Register – PSCR .....	278
Timer Counter-Reload Register – CRR .....	279
Channel 0 Capture/Compare Register – CH0CCR .....	280
Channel 1 Capture/Compare Register – CH1CCR .....	281
Channel 2 Capture/Compare Register – CH2CCR .....	282
Channel 3 Capture/Compare Register – CH3CCR .....	283
Channel 0 Asymmetric Compare Register – CH0ACR.....	284
Channel 1 Asymmetric Compare Register – CH1ACR.....	284
Channel 2 Asymmetric Compare Register – CH2ACR.....	285
Channel 3 Asymmetric Compare Register – CH3ACR.....	285
<b>15 Motor Control Timer (MCTM) .....</b>	<b>286</b>
Introduction .....	286
Features.....	287
Functional Descriptions .....	288
Counter Mode .....	288



Clock Controller .....	292
Trigger Controller .....	293
Slave Controller .....	294
Master Controller .....	296
Channel Controller .....	297
Input Stage .....	299
Output Stage .....	300
Update Management .....	311
Single Pulse Mode .....	313
Asymmetric PWM Mode .....	315
Timer Interconnection .....	315
Trigger Peripheral Start .....	319
Lock Level Table .....	319
PDMA Request (HT32F54243/HT32F54253 only) .....	320
Register Map .....	321
Register Descriptions .....	322
Timer Counter Configuration Register – CNTCFR .....	322
Timer Mode Configuration Register – MDCFR .....	323
Timer Trigger Configuration Register – TRCFR .....	326
Timer Control Register – CTR .....	327
Channel 0 Input Configuration Register – CH0ICFR .....	328
Channel 1 Input Configuration Register – CH1ICFR .....	330
Channel 2 Input Configuration Register – CH2ICFR .....	332
Channel 3 Input Configuration Register – CH3ICFR .....	334
Channel 0 Output Configuration Register – CH0OCFR .....	336
Channel 1 Output Configuration Register – CH1OCFR .....	338
Channel 2 Output Configuration Register – CH2OCFR .....	340
Channel 3 Output Configuration Register – CH3OCFR .....	342
Channel Control Register – CHCTR .....	344
Channel Polarity Configuration Register – CHPOLR .....	346
Channel Break Configuration Register – CHBRKCFR .....	347
Channel Break Control Register – CHBRKCTR .....	348
Timer PDMA/Interrupt Control Register – DICTR .....	350
Timer Event Generator Register – EVGR .....	352
Timer Interrupt Status Register – INTSR .....	354
Timer Counter Register – CNTR .....	356
Timer Prescaler Register – PSCR .....	357
Timer Counter-Reload Register – CRR .....	358
Timer Repetition Register – REPR .....	358
Channel 0 Capture/Compare Register – CH0CCR .....	359
Channel 1 Capture/Compare Register – CH1CCR .....	360
Channel 2 Capture/Compare Register – CH2CCR .....	361
Channel 3 Capture/Compare Register – CH3CCR .....	362
Channel 0 Asymmetric Compare Register – CH0ACR .....	363
Channel 1 Asymmetric Compare Register – CH1ACR .....	363

Channel 2 Asymmetric Compare Register – CH2ACR .....	364
Channel 3 Asymmetric Compare Register – CH3ACR .....	364
<b>16 Single-Channel Timer (SCTM) .....</b>	<b>365</b>
Introduction .....	365
Features .....	366
Functional Descriptions .....	366
Counter Mode .....	366
Clock Controller .....	367
Trigger Controller .....	368
Slave Controller .....	369
Channel Controller .....	371
Input Stage .....	372
Output Stage .....	373
Update Management .....	375
Register Map .....	376
Register Descriptions .....	377
Timer Counter Configuration Register – CNTCFR .....	377
Timer Mode Configuration Register – MDCFR .....	378
Timer Trigger Configuration Register – TRCFR .....	379
Timer Control Register – CTR .....	380
Channel Input Configuration Register – CHICFR .....	381
Channel Output Configuration Register – CHOCFR .....	383
Channel Control Register – CHCTR .....	384
Channel Polarity Configuration Register – CHPOLR .....	385
Timer Interrupt Control Register – DICTR .....	386
Timer Event Generator Register – EVGR .....	387
Timer Interrupt Status Register – INTSR .....	388
Timer Counter Register – CNTR .....	389
Timer Prescaler Register – PSCR .....	389
Timer Counter-Reload Register – CRR .....	390
Channel Capture/Compare Register – CHCCR .....	391
<b>17 Basic Function Timer (BFTM) .....</b>	<b>392</b>
Introduction .....	392
Features .....	392
Functional Description .....	392
Repetitive Mode .....	392
One Shot Mode .....	393
Trigger ADC Start .....	394
Register Map .....	394
Register Descriptions .....	395
BFTM Control Register – BFTMCR .....	395
BFTM Status Register – BFTMSR .....	396

BFTM Counter Value Register – BFTMCNTR .....	397
BFTM Compare Value Register – BFTMCMPR .....	397
<b>18 Real Time Clock (RTC) .....</b>	<b>398</b>
Introduction .....	398
Features .....	398
Functional Descriptions .....	399
RTC Related Register Reset .....	399
Reading RTC Register .....	399
Low Speed Clock Configuration .....	399
RTC Counter Operation .....	399
Interrupt and Wakeup Control .....	400
RTCOUNT Output Pin Configuration .....	400
Register Map .....	401
Register Descriptions .....	401
RTC Counter Register – RTCCNT .....	401
RTC Compare Register – RTCCMP .....	402
RTC Control Register – RTCCR .....	403
RTC Status Register – RTCSR .....	405
RTC Interrupt and Wakeup Enable Register – RTCIWEN .....	406
<b>19 Watchdog Timer (WDT) .....</b>	<b>407</b>
Introduction .....	407
Features .....	407
Functional Description .....	408
Register Map .....	409
Register Descriptions .....	410
Watchdog Timer Control Register – WDTCR .....	410
Watchdog Timer Mode Register 0 – WDTMR0 .....	411
Watchdog Timer Mode Register 1 – WDTMR1 .....	412
Watchdog Timer Status Register – WDTSR .....	413
Watchdog Timer Protection Register – WDTPR .....	414
Watchdog Timer Clock Selection Register – WDTCSR .....	415
<b>20 Inter-Integrated Circuit (I<sup>2</sup>C) .....</b>	<b>416</b>
Introduction .....	416
Features .....	417
Functional Descriptions .....	417
Two-Wire Serial Interface .....	417
START and STOP Conditions .....	417
Data Validity .....	418
Addressing Format .....	418
Data Transfer and Acknowledge .....	420
Clock Synchronization .....	420

Arbitration .....	421
General Call Addressing .....	421
Bus Error.....	421
Address Mask Enable .....	422
Address Snoop .....	422
Operation Mode .....	422
Conditions of Holding SCL Line.....	426
I <sup>2</sup> C Timeout Function .....	427
PDMA Interface (HT32F54243/HT32F54253 only).....	427
Register Map .....	428
Register Descriptions.....	428
I <sup>2</sup> C Control Register – I2CCR .....	428
I <sup>2</sup> C Interrupt Enable Register – I2CIER .....	430
I <sup>2</sup> C Address Register – I2CADDR.....	431
I <sup>2</sup> C Status Register – I2CSR.....	432
I <sup>2</sup> C SCL High Period Generation Register – I2CSHPGR.....	435
I <sup>2</sup> C SCL Low Period Generation Register – I2CSLPGR .....	436
I <sup>2</sup> C Data Register – I2CDR .....	437
I <sup>2</sup> C Target Register – I2CTAR .....	438
I <sup>2</sup> C Address Mask Register – I2CADDRMR .....	439
I <sup>2</sup> C Address Snoop Register – I2CADDRSR.....	440
I <sup>2</sup> C Timeout Register – I2CTOUT.....	441
<b>21 Serial Peripheral Interface (SPI) .....</b>	<b>442</b>
Introduction .....	442
Features.....	443
Functional Descriptions .....	443
Master Mode.....	443
Slave Mode.....	443
SPI Serial Frame Format.....	444
SPI Dual Mode.....	448
Status Flags.....	450
PDMA Interface (HT32F54243/HT32F54253 only).....	453
Register Map .....	453
Register Descriptions.....	454
SPI Control Register 0 – SPICR0 .....	454
SPI Control Register 1 – SPICR1 .....	456
SPI Interrupt Enable Register – SPIIER .....	458
SPI Clock Prescaler Register – SPICPR .....	459
SPI Data Register – SPIDR.....	460
SPI Status Register – SPISR.....	461
SPI FIFO Control Register – SPIFCR.....	462
SPI FIFO Status Register – SPIFSR .....	463
SPI FIFO Time Out Counter Register – SPIFTOCR.....	464

## **22 Universal Synchronous Asynchronous Receiver Transmitter (USART) ..... 465**

Introduction .....	465
Features .....	466
Functional Descriptions .....	466
Serial Data Format.....	466
Baud Rate Generation .....	467
Hardware Flow Control .....	469
IrDA.....	470
RS485 Mode.....	472
Synchronous Master Mode.....	474
Interrupts and Status .....	476
PDMA Interface (HT32F54243/HT32F54253 only).....	476
Register Map .....	476
Register Descriptions.....	477
USART Data Register – USRDR.....	477
USART Control Register – USRCR.....	478
USART FIFO Control Register – USRFCR.....	480
USART Interrupt Enable Register – USRIER .....	481
USART Status & Interrupt Flag Register – USRSIFR.....	483
USART Timing Parameter Register – USRTPR .....	485
USART IrDA Control Register – IrDACR .....	486
USART RS485 Control Register – RS485CR.....	487
USART Synchronous Control Register – SYNCR .....	488
USART Divider Latch Register – USRDLR.....	489
USART Test Register – USRTSTR.....	490

## **23 Universal Asynchronous Receiver Transmitter (UART)..... 491**

Introduction .....	491
Features.....	492
Functional Descriptions .....	492
Serial Data Format.....	492
Baud Rate Generation .....	493
Interrupts and Status .....	494
PDMA Interface (HT32F54243/HT32F54253 only).....	494
Register Map .....	495
Register Descriptions.....	495
UART Data Register – URDR.....	495
UART Control Register – URCR.....	496
UART Interrupt Enable Register – URIER.....	497
UART Status & Interrupt Flag Register – URSIFR .....	499
UART Divider Latch Register – URDLR .....	500
UART Test Register – URTSTR.....	501

<b>24 Peripheral Direct Memory Access (PDMA).....</b>	<b>502</b>
Introduction .....	502
Features .....	502
Functional Description .....	503
AHB Master .....	503
PDMA Channel .....	503
PDMA Request Mapping .....	503
Channel Transfer .....	504
Channel Priority .....	504
Transfer Request .....	505
Address Mode .....	505
Auto-Reload .....	506
Transfer Interrupt .....	506
Register Map .....	507
Register Descriptions .....	508
PDMA Channel n Control Register – PDMACHnCR (n = 0 ~ 5) .....	508
PDMA Channel n Source Address Register – PDMACHnSADR (n = 0 ~ 5) .....	510
PDMA Channel n Destination Address Register – PDMACHnDADR (n = 0 ~ 5) .....	511
PDMA Channel n Transfer Size Register – PDMACHnTSR (n = 0 ~ 5) .....	512
PDMA Channel n Current Transfer Size Register – PDMACHnCTSR (n = 0 ~ 5) .....	513
PDMA Interrupt Status Register – PDMAISR .....	514
PDMA Interrupt Status Clear Register – PDMAISCR .....	515
PDMA Interrupt Enable Register – PDMAIER .....	516
<b>25 Divider (DIV) .....</b>	<b>518</b>
Introduction .....	518
Features .....	518
Functional Descriptions .....	518
Register Map .....	518
Register Descriptions .....	519
Divider Control Register – CR .....	519
Dividend Data Register – DDR .....	520
Divisor Data Register – DSR .....	520
Quotient Data Register – QTR .....	521
Remainder Data Register – RMR .....	521
<b>26 Cyclic Redundancy Check (CRC).....</b>	<b>522</b>
Introduction .....	522
Features .....	522
Functional Descriptions .....	523
CRC Computation .....	523
Byte and Bit Reversal for CRC Computation .....	523
CRC with PDMA .....	524
Register Map .....	524

Register Descriptions.....	524
CRC Control Register – CRCCR.....	524
CRC Seed Register – CRCSDR.....	525
CRC Checksum Register – CRCCSR.....	526
CRC Data Register – CRCDR.....	526
<b>27 LED Controller (LEDC) .....</b>	<b>527</b>
Introduction .....	527
Features.....	528
Functional Description .....	528
LEDC Basic Setting .....	528
LEDC Clock Source Selection .....	528
LEDC Operational Description.....	529
LEDC Frame Interrupt .....	533
LEDC Data Update Method .....	534
Frame Rate Calculation .....	534
Register Map .....	535
Register Descriptions.....	536
LED Control Register – LEDCR.....	536
LED COM Enable Register – LEDCER .....	537
LED Polarity Control Register – LEDPCR .....	538
LED Interrupt Enable Register – LEDIER.....	539
LED Status Register – LEDSR .....	540
LED Dead Time Control Register – LEDDTCR .....	541
LED Data Register n – LEDDRn (n = 0 ~ 11) .....	542
<b>28 Touch Key – TKEY .....</b>	<b>543</b>
Introduction .....	543
Features.....	543
Functional Description .....	543
Manual Mode .....	544
Auto Scan Mode .....	545
Periodic Auto Scan Mode .....	545
Touch Key Interrupts.....	546
Touch Key Scan Operation Flowchart .....	547
Register Map .....	550
Register Descriptions.....	551
Touch Key Control Register – TKCR .....	551
Touch Key Counter Register – TKCNTR .....	554
Touch Key Time Slot Counter Preload Register – TKTSCRR .....	555
Touch Key Interrupt Enable Register – TKIER .....	556
Touch Key Status Register – TKSR.....	557
Touch Key Module n Control Register – TKMnCR .....	559
Touch Key Module n Key Configuration Register – TKMnKCFGR .....	561

Touch Key Module n Status Register – TKMnSR .....	563
Touch Key Module n Reference Oscillator Capacitor Register – TKMnROCPR .....	564
Touch Key Module n Key 3 Capacitor Register – TKMnK3CPR.....	564
Touch Key Module n Key 2 Capacitor Register – TKMnK2CPR.....	565
Touch Key Module n Key 1 Capacitor Register – TKMnK1CPR.....	565
Touch Key Module n Key 0 Capacitor Register – TKMnK0CPR.....	566
Touch Key Module n C/F Counter Register – TKMnCFCNTR.....	567
Touch Key Module n Key 3 Counter Register – TKMnK3CNTR.....	568
Touch Key Module n Key 2 Counter Register – TKMnK2CNTR.....	568
Touch Key Module n Key 1 Counter Register – TKMnK1CNTR.....	569
Touch Key Module n Key 0 Counter Register – TKMnK0CNTR.....	569
Touch Key Module n Key 3 Threshold Register – TKMnK3THR .....	570
Touch Key Module n Key 2 Threshold Register – TKMnK2THR .....	570
Touch Key Module n Key 1 Threshold Register – TKMnK1THR .....	571
Touch Key Module n Key 0 Threshold Register – TKMnK0THR .....	571



## List of Tables

Table 1. Features and Peripheral List .....	30
Table 2. Document Conventions .....	33
Table 3. HT32F54231/HT32F54241 Register Map .....	39
Table 4. HT32F54243/HT32F54253 Register Map .....	40
Table 5. Flash Memory and Option Byte .....	45
Table 6. Relationship between Wait State Cycle and HCLK .....	46
Table 7. Booting Modes .....	46
Table 8. Option Byte Memory Map .....	50
Table 9. Access Permission of Protected Main Flash Page .....	51
Table 10. Access Permission When Security Protection is Enabled .....	52
Table 11. FMC Register Map .....	53
Table 12. Operation Mode Definitions .....	72
Table 13. Enter/Exit Power Saving Modes .....	73
Table 14. Power Status after System Reset .....	73
Table 15. PWRCU Register Map .....	74
Table 16. Output Divider 2 Value Mapping .....	85
Table 17. Feedback Divider 2 Value Mapping .....	85
Table 18. CKOUT Clock Source .....	88
Table 19. CKCU Register Map .....	88
Table 20. RSTCU Register Map .....	115
Table 21. AFIO, GPIO and I/O Pad Control Signal True Table .....	124
Table 22. GPIO Register Map .....	125
Table 23. AFIO Selection for Peripheral Map Example .....	169
Table 24. AFIO Register Map .....	169
Table 25. Exception Types .....	174
Table 26. NVIC Register Map .....	176
Table 27. EXTI Register Map .....	180
Table 28. Data Format in ADCDR [15:0] .....	195
Table 29. A/D Converter Register Map .....	197
Table 30. CMP Register Map .....	218
Table 31. Counting Direction and Encoding Signals .....	238
Table 32. Compare Match Output Setup .....	239
Table 33. GPTM Register Map .....	249
Table 34. GPTM Internal Trigger Connection .....	254
Table 35. Compare Match Output Setup .....	301
Table 36. Output Control Bits for Complementary Output with a Break Event Occurrence .....	310
Table 37. Lock Level Table .....	319
Table 38. MCTM Register Map .....	321
Table 39. MCTM Internal Trigger Connection .....	326

Table 40. Compare Match Output Setup .....	373
Table 41. SCTM Register Map.....	376
Table 42. BFTM Register Map .....	394
Table 43. LSE Startup Mode Operating Current and Startup Time.....	399
Table 44. RTCOUT Output Mode and Active Level Setting .....	400
Table 45. RTC Register Map.....	401
Table 46. Watchdog Timer Register Map.....	409
Table 47. Conditions of Holding SCL line.....	426
Table 48. I <sup>2</sup> C Register Map .....	428
Table 49. I <sup>2</sup> C Clock Setting Example .....	436
Table 50. SPI Interface Format Setup.....	444
Table 51. SPI Mode Fault Trigger Conditions .....	452
Table 52. SPI Master Mode SPI_SEL Pin Status.....	452
Table 53. SPI Register Map .....	453
Table 54. Baud Rate Deviation Error Calculation – CK_USART = 40 MHz.....	468
Table 55. Baud Rate Deviation Error Calculation – CK_USART = 48 MHz.....	468
Table 56. Baud Rate Deviation Error Calculation – CK_USART = 60 MHz.....	468
Table 57. USART Register Map.....	476
Table 58. Baud Rate Deviation Error Calculation – CK_UART = 40 MHz .....	493
Table 59. Baud Rate Deviation Error Calculation – CK_UART = 48 MHz .....	494
Table 60. Baud Rate Deviation Error Calculation – CK_UART = 60 MHz .....	494
Table 61. UART Register Map .....	495
Table 62. PDMA Channel Assignments .....	504
Table 63. PDMA Address Modes .....	505
Table 64. PDMA Register Map.....	507
Table 65. DIV Register Map.....	518
Table 66. CRC Register Map .....	524
Table 67. LED Pixel Data and (SEGx, COMy) Relationship .....	534
Table 68. LEDC Register Map .....	535
Table 69. Touch Key Register Map .....	550

## List of Figures

Figure 1. HT32F54231/HT32F54241 Block Diagram .....	31
Figure 2. HT32F54243/HT32F54253 Block Diagram .....	32
Figure 3. Cortex®-M0+ Block Diagram.....	35
Figure 4. Bus Architecture .....	36
Figure 5. HT32F54231/HT32F54241 Memory Map.....	37
Figure 6. HT32F54243/HT32F54253 Memory Map.....	38
Figure 7. Flash Memory Controller Block Diagram.....	43
Figure 8. Flash Memory Map.....	44
Figure 9. Vector Remapping .....	46
Figure 10. Page Erase Operation Flowchart .....	47
Figure 11. Mass Erase Operation Flowchart.....	48
Figure 12. Word Programming Operation Flowchart.....	49
Figure 13. PWRUCU Block Diagram.....	69
Figure 14. Power-On Reset / Power-Down Reset Waveform.....	70
Figure 15. CKCU Block Diagram .....	81
Figure 16. External Crystal, Ceramic and Resonators for HSE .....	82
Figure 17. HSI Auto Trimming Block Diagram .....	84
Figure 18. PLL Block Diagram .....	84
Figure 19. External Crystal, Ceramic and Resonators for LSE .....	86
Figure 20. RSTCU Block Diagram.....	114
Figure 21. Power-On Reset Sequence .....	115
Figure 22. GPIO Block Diagram .....	122
Figure 23. AFIO/GPIO Control Signal.....	124
Figure 24. AFIO Block Diagram .....	167
Figure 25. EXTI Channel Input Selection .....	168
Figure 26. EXTI Block Diagram .....	177
Figure 27. EXTI Wakeup Event Management .....	178
Figure 28. EXTI Wakeup Interrupt Service Routine Management.....	179
Figure 29. EXTI Interrupt Debounce Function .....	179
Figure 30. ADC with V <sub>REF</sub> Block Diagram.....	189
Figure 31. One Shot Conversion Mode .....	192
Figure 32. Continuous Conversion Mode .....	192
Figure 33. Discontinuous Conversion Mode .....	194
Figure 34. Voltage Reference Generator Block Diagram .....	197
Figure 35. Comparator Block Diagram .....	215
Figure 36. Comparator Voltage Reference Block Diagram.....	216
Figure 37. Comparator Interrupt Signals .....	217
Figure 38. Comparator Wakeup Signals.....	217
Figure 39. GPTM Block Diagram.....	224

Figure 40. Up-counting Example .....	226
Figure 41. Down-counting Example.....	226
Figure 42. Center-aligned Counting Example.....	227
Figure 43. GPTM Clock Source Selection .....	228
Figure 44. Trigger Controller Block .....	229
Figure 45. Slave Controller Diagram .....	230
Figure 46. GPTM in Restart Mode .....	230
Figure 47. GPTM in Pause Mode .....	231
Figure 48. GPTM in Trigger Mode .....	231
Figure 49. Master GPTMn and Slave GPTMm/TMm Connection .....	232
Figure 50. MTO Selection .....	232
Figure 51. Capture/Compare Block Diagram.....	233
Figure 52. Input Capture Mode.....	234
Figure 53. PWM Pulse Width Measurement Example.....	235
Figure 54. Channel 0 and Channel 1 Input Stages.....	235
Figure 55. Channel 2 and Channel 3 Input Stages.....	236
Figure 56. TI0 Digital Filter Diagram with N = 2.....	236
Figure 57. Input Stage and Quadrature Decoder Block Diagram .....	237
Figure 58. Both TI0 and TI1 Quadrature Decoder Counting .....	238
Figure 59. Output Stage Block Diagram .....	238
Figure 60. Toggle Mode Channel Output Reference Signal (CHxPRE = 0) .....	239
Figure 61. Toggle Mode Channel Output Reference Signal (CHxPRE = 1) .....	240
Figure 62. PWM Mode Channel Output Reference Signal and Counter in Up-counting Mode .....	240
Figure 63. PWM Mode Channel Output Reference Signal and Counter in Down-counting Mode .....	241
Figure 64. PWM Mode Channel Output Reference Signal and Counter in Centre-aligned Mode .....	241
Figure 65. Update Event Setting Diagram .....	242
Figure 66. Single Pulse Mode.....	243
Figure 67. Immediate Active Mode Delay .....	244
Figure 68. Asymmetric PWM Mode versus Center-Aligned Counting Mode .....	245
Figure 69. Pausing MCTM Using the GPTM CH0OREF Signal .....	246
Figure 70. Triggering MCTM with GPTM Update Event .....	246
Figure 71. Trigger GPTM and MCTM with the GPTM CH0 Input .....	247
Figure 72. GPTM PDMA Mapping Diagram.....	248
Figure 73. MCTM Block Diagram .....	286
Figure 74. Up-Counting Example .....	288
Figure 75. Down-Counting Example.....	289
Figure 76. Center-Aligned Counting Example .....	290
Figure 77. Update Event 1 Dependent Repetition Mechanism Example.....	291
Figure 78. MCTM Clock Source Selection.....	292
Figure 79. Trigger Controller Block .....	293
Figure 80. Slave Controller Diagram .....	294

Figure 81. MCTM in Restart Mode .....	294
Figure 82. MCTM in Pause Mode.....	295
Figure 83. MCTM in Trigger Mode.....	295
Figure 84. Master MCTMn and Slave GPTMm Connection .....	296
Figure 85. MTO Selection .....	296
Figure 86. Capture/Compare Block Diagram.....	297
Figure 87. Input Capture Mode.....	297
Figure 88. PWM Pulse Width Measurement Example.....	298
Figure 89. Channel 0 and Channel 1 Input Stages.....	299
Figure 90. Channel 2 and Channel 3 Input Stages.....	299
Figure 91. TI0 Digital Filter Diagram with N = 2.....	300
Figure 92. Output Stage Block Diagram .....	300
Figure 93. Toggle Mode Channel Output Reference Signal – CHxPRE = 0.....	301
Figure 94. Toggle Mode Channel Output Reference Signal – CHxPRE = 1 .....	302
Figure 95. PWM Mode Channel Output Reference Signal and Counter in Up-Counting Mode .....	302
Figure 96. PWM Mode Channel Output Reference Signal and Counter in Down-counting Mode .....	303
Figure 97. PWM Mode 1 Channel Output Reference Signal and Counter in Centre-Aligned Counting Mode.....	303
Figure 98. Dead-Time Insertion Performed for Complementary Outputs .....	304
Figure 99. MCTM Break Signal Bolck Diagram .....	305
Figure 100. MT_BRK Pin Digital Filter Diagram with N = 2 .....	305
Figure 101. Channel 3 Output with a Break Event Occurrence .....	306
Figure 102. Channel 0 ~ 2 Complementary Outputs with a Break Event Occurrence.....	307
Figure 103. Channel 0 ~ 2 Only One Output Enabled when Break Event Occurs .....	308
Figure 104. Hardware Protection When Both CHxO and CHxNO Are in Active Condition .....	309
Figure 105. Update Event 1 Setup Diagram.....	311
Figure 106. CHxE, CHxNE and CHxOM Updated by Update Event 2 .....	312
Figure 107. Update Event 2 Setup Diagram.....	312
Figure 108. Single Pulse Mode.....	313
Figure 109. Immediate Active Mode Delay .....	314
Figure 110. Asymmetric PWM Mode versus Center-Aligned Counting Mode.....	315
Figure 111. Pausing GPTM Using the MCTM CH0OREF Signal.....	316
Figure 112. Triggering GPTM with MCTM Update Event 1.....	316
Figure 113. Trigger MCTM and GPTM with the MCTM CH0 Input .....	317
Figure 114. CH0XOR Input as Hall Sensor Interface.....	318
Figure 115. MCTM PDMA Mapping Diagram.....	320
Figure 116. SCTM Block Diagram .....	365
Figure 117. Up-Counting Example.....	366
Figure 118. SCTM Clock Source Selection.....	367
Figure 119. Trigger Controller Block .....	368
Figure 120. Slave Controller Diagram .....	369

Figure 121. SCTM in Restart Mode .....	369
Figure 122. SCTM in Pause Mode .....	370
Figure 123. SCTM in Trigger Mode .....	370
Figure 124. Capture/Compare Block Diagram.....	371
Figure 125. Input Capture Mode.....	371
Figure 126. Channel Input Stages .....	372
Figure 127. TI Digital Filter Diagram with N = 2.....	372
Figure 128. Output Stage Block Diagram .....	373
Figure 129. Toggle Mode Channel Output Reference Signal (CHPRE = 0) .....	374
Figure 130. Toggle Mode Channel Output Reference Signal (CHPRE = 1) .....	374
Figure 131. PWM Mode Channel Output Reference Signal .....	375
Figure 132. Update Event Setting Diagram .....	376
Figure 133. BFTM Block Diagram .....	392
Figure 134. BFTM – Repetitive Mode.....	393
Figure 135. BFTM – One Shot Mode.....	393
Figure 136. BFTM – One Shot Mode Counter Updating .....	394
Figure 137. RTC Block Diagram.....	398
Figure 138. Watchdog Timer Block Diagram .....	407
Figure 139. Watchdog Timer Behavior .....	409
Figure 140. I <sup>2</sup> C Module Block Diagram.....	416
Figure 141. START and STOP Condition .....	418
Figure 142. Data Validity.....	418
Figure 143. 7-bit Addressing Mode.....	419
Figure 144. 10-bit Addressing Write Transmit Mode.....	419
Figure 145. 10-bits Addressing Read Receive Mode .....	419
Figure 146. I <sup>2</sup> C Bus Acknowledge .....	420
Figure 147. Clock Synchronization during Arbitration.....	420
Figure 148. Two Masters Arbitration Procedure.....	421
Figure 149. Master Transmitter Timing Diagram .....	423
Figure 150. Master Receiver Timing Diagram .....	424
Figure 151. Slave Transmitter Timing Diagram .....	425
Figure 152. Slave Receiver Timing Diagram .....	426
Figure 153. SCL Timing Diagram.....	436
Figure 154. SPI Block Diagram .....	442
Figure 155. SPI Single Byte Transfer Timing Diagram – CPOL = 0, CPHA = 0.....	444
Figure 156. SPI Continuous Data Transfer Timing Diagram – CPOL = 0, CPHA = 0 .....	445
Figure 157. SPI Single Byte Transfer Timing Diagram – CPOL = 0, CPHA = 1.....	445
Figure 158. SPI Continuous Transfer Timing Diagram – CPOL = 0, CPHA = 1.....	445
Figure 159. SPI Single Byte Transfer Timing Diagram – CPOL = 1, CPHA = 0.....	446
Figure 160. SPI Continuous Transfer Timing Diagram – CPOL = 1, CPHA = 0.....	446
Figure 161. SPI Single Byte Transfer Timing Diagram – CPOL = 1, CPHA = 1.....	447

Figure 162. SPI Continuous Transfer Timing Diagram – CPOL = 1, CPHA = 1.....	447
Figure 163. SPI Dual Mode Bit Sequence – CPOL = 0, CPHA = 0, DFL = 0x8 (16-bit), MSB Transmitted First.....	448
Figure 164. SPI Dual Mode Bit Sequence – CPOL = 0, CPHA = 1, DFL = 0x8 (16-bit), MSB Transmitted First.....	448
Figure 165. SPI Dual Mode Bit Sequence – CPOL = 1, CPHA = 0, DFL = 0x8 (16-bit), MSB Transmitted First.....	449
Figure 166. SPI Dual Mode Bit Sequence – CPOL = 1, CPHA = 1, DFL = 0x8 (16-bit), MSB Transmitted First.....	449
Figure 167. SPI Dual Mode Data Read Example – CPOL = 1, CPHA = 1.....	450
Figure 168. SPI Multi-Master Slave Environment.....	451
Figure 169. USART Block Diagram .....	465
Figure 170. USART Serial Data Format .....	467
Figure 171. USART Clock CK_USART and Data Frame Timing .....	467
Figure 172. Hardware Flow Control between 2 USARTs.....	469
Figure 173. USART RTS Flow Control .....	469
Figure 174. USART CTS Flow Control .....	470
Figure 175. IrDA Modulation and Demodulation.....	470
Figure 176. USART I/O and IrDA Block Diagram .....	472
Figure 177. RS485 Interface and Waveform .....	473
Figure 178. USART Synchronous Transmission Example .....	474
Figure 179. 8-Bit Format USART Synchronous Waveform.....	475
Figure 180. UART Block Diagram.....	491
Figure 181. UART Serial Data Format.....	492
Figure 182. UART Clock CK_UART and Data Frame Timing.....	493
Figure 183. PDMA Block Diagram .....	502
Figure 184. PDMA Request Mapping Architecture .....	503
Figure 185. PDMA Channel Arbitration and Scheduling Example .....	505
Figure 186. Divider Functional Diagram .....	518
Figure 187. CRC Block Diagram .....	522
Figure 188. CRC Data Bit and Byte Reversal Example.....	523
Figure 189. LEDC Block Diagram.....	527
Figure 190. Common Cathode 8-segment Digital Display Connection .....	529
Figure 191. Common Cathode 8-segment Digital Display Timing .....	530
Figure 192. Common Anode 8-segment Digital Display + NPN BJT Connection.....	530
Figure 193. Common Anode 8-segment Digital Display+ NPN BJT Timing .....	531
Figure 194. Common Cathode 8-segment Digital Display + NPN Transistor Connection.....	531
Figure 195. Common Cathode 8-segment Digital Display + NPN Transistor Timing.....	532
Figure 196. Common Anode 8-segment Digital Display + PNP BJT Connection .....	532
Figure 197. Common Anode 8-segment Digital Display + PNP BJT Timing.....	533
Figure 198. Frame Interrupt Diagram .....	533
Figure 199. Touch Key Manual Scan Mode Timing Diagram.....	544

Figure 200. Touch Key Interrupts.....	547
Figure 201. Touch Key Manual Scan Mode Flowchart – TKMOD[1:0] = 01 .....	547
Figure 202. Touch Key Auto Scan Mode Flowchart – TKMOD[1:0] = 00 .....	548
Figure 203. Touch Key Periodic Auto Scan Mode Flowchart – TKMOD[1:0] = 10 ~ 11 .....	549



# 1 Introduction

## Overview

This user manual provides detailed information including how to use the HT32F54231/HT32F54241/HT32F54243/HT32F54253 devices, system and bus architecture, memory organization and peripheral instructions. The target audiences for this document are software developers, application developers and hardware developers. For more information regarding pin assignment, package and electrical characteristics, please refer to the HT32F54231/HT32F54241/HT32F54243/HT32F54253 datasheet.

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

The devices operate at a frequency of up to 60 MHz with a Flash accelerator to obtain maximum efficiency. It provides up to 128 KB of embedded Flash memory for code/data storage and up to 16 KB of embedded SRAM memory for system operation and application program usage. A variety of peripherals, such as Hardware Divider DIV, PDMA, ADC, I<sup>2</sup>C, USART, UART, SPI, GPTM, SCTM, BFTM, MCTM, CRC-16/32, RTC, WDT, Touch key, LED controller and SW-DP (Serial Wire Debug Port), etc., are also implemented in the device series. Several power saving modes provide the flexibility for maximum optimization between wakeup latency and power consumption, which is an especially important consideration in low power applications.

The above features ensure that the devices are suitable for use in a wide range of applications, especially in areas such as washing machines, refrigerators, electric pressure cookers, high-speed blenders, rice cookers and so on.

## Features

- Core
  - 32-bit Arm® Cortex®-M0+ processor core
  - Up to 60 MHz operating frequency
  - Single-cycle multiplication
  - Integrated Nested Vectored Interrupt Controller (NVIC)
  - 24-bit SysTick timer
- On-Chip Memory
  - Up to 128 KB on-chip Flash memory for instruction/data and option storage
  - Up to 16 KB on-chip SRAM
  - Supports multiple booting modes
- Flash Memory Controller – FMC
  - Flash accelerator to obtain maximum efficiency
  - 32-bit word programming with In System Programming (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 Hz crystal oscillator
  - Internal 8 MHz RC oscillator trimmed to  $\pm 2\%$  accuracy at 5.0 V operating voltage and 25 °C operating temperature
  - Internal 32 kHz RC oscillator
  - Integrated system clock PLL
  - Independent clock divider and gating bits for peripheral clock sources
- Power Management – PWRCU
  - Single  $V_{DD}$  power supply: 2.5 V to 5.5 V
  - Integrated 1.5 V LDO regulator for MCU core, peripheral and memory power supply
  - $V_{DD}$  power supply for RTC
  - Two power domains:  $V_{DD}$  and  $V_{CORE}$
  - Three power saving modes: Sleep, Deep-Sleep1 and Deep-Sleep2 modes
- External Interrupt/Event Controller – EXTI
  - Up to 16 EXTI lines with configurable trigger source and type
  - All GPIO pins can be selected as EXTI trigger source
  - Source trigger type can be high level, low level, negative edge, positive edge or both 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 10 external analog input channels
- Comparator – CMP (HT32F54243/HT32F54253 only)
  - Rail-to-rail comparators
  - Each comparator has configurable negative input used for flexible voltage selection
    - ◆ External CN pin
    - ◆ Internal 8-bit CVR output
  - Programmable hysteresis
  - Programming respond speed and power consumption
  - Comparator output can be routed to I/O or to multiple timers or ADC trigger inputs
  - 8-bit CVR can be configurable to dedicated I/O for voltage reference
  - Comparator has interrupt generation capability with wakeup from Sleep, Deep-Sleep1 or Deep-Sleep2 mode through the EXTI controller
- I/O Ports – GPIO
  - Up to 54 GPIOs
  - Port A, B, C, D are mapped on 16 external interrupts – EXTI
  - Almost all I/O pins have configurable output driving current

- **Motor Control Timer – MCTM**
  - 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
  - Input Capture function
  - Compare Match Output
  - PWM waveform generation with Edge-aligned and Center-aligned Counting Modes
  - Single Pulse Mode Output
  - Complementary Output with programmable dead-time insertion
  - Supports 3-phase motor control and hall sensor interface
  - Break input to force the timer's output signals into a reset or fixed condition
- **General-Purpose Timer – GPTM**
  - 16-bit up/down auto-reload counter
  - Up to 4 independent channels
  - 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
  - Input Capture function
  - Compare Match Output
  - PWM waveform generation with Edge-aligned and Center-aligned Counting Modes
  - Single Pulse Mode Output
  - Encoder interface controller with two inputs using quadrature decoder
- **Single-Channel Timer – SCTM**
  - 16-bit auto-reload up-counter
  - One channel for each timer
  - 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
  - Input Capture function
  - Compare Match Output
  - PWM waveform generation with Edge-aligned
- **Basic Function Timer – BFTM**
  - 32-bit compare/match count-up counter – no I/O control features
  - One shot mode – counting stops after a match condition
  - Repetitive mode – restart counter after a match condition
- **Watchdog Timer – WDT**
  - 12-bit down-counter with a 3-bit prescaler
  - Reset event for the system
  - Programmable watchdog timer window function
  - Register write protection function
- **Real Time Clock – RTC**
  - 24-bit up-counter with a programmable prescaler
  - Alarm function
  - Interrupt and wakeup 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 ( $f_{\text{PCLK}}/2$ ) MHz for master mode and ( $f_{\text{PCLK}}/3$ ) MHz for slave mode
  - FIFO Depth: 8 levels
  - Multi-master and multi-slave operation
- Universal Synchronous Asynchronous Receiver Transmitter – USART
  - Supports both asynchronous and clocked synchronous serial communication modes
  - Asynchronous operating baud rate clock frequency of up to ( $f_{\text{PCLK}}/16$ ) MHz and synchronous operating baud rate clock frequency of up to ( $f_{\text{PCLK}}/8$ ) 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 ( $f_{\text{PCLK}}/16$ ) MHz
  - Capability of full duplex communication
  - Fully programmable serial communication characteristics including word length, parity bit, stop bit and bit order
  - Error detection: Parity, overrun and frame error
- Cyclic Redundancy Check – CRC
  - Supports CRC16 polynomial: 0x8005,  
 $X^{16} + X^{15} + X^2 + 1$
  - Supports CCITT CRC16 polynomial: 0x1021,  
 $X^{16} + X^{12} + X^5 + 1$
  - Supports IEEE-802.3 CRC32 polynomial: 0x04C11DB7,  
 $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
  - Supports 1's complement, byte reverse and bit reverse operation on data and checksum
  - Supports byte, half-word and word data size
  - Programmable CRC initial seed value
  - CRC computation executed in 1 AHB clock cycle for 8-bit data and 4 AHB clock cycles for 32-bit data
  - HT32F54243/HT32F54253 devices support PDMA to complete a CRC computation of a block of memory

- Peripheral Direct Memory Access – PDMA (HT32F54243/HT32F54253 only)
  - 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, MCTM, GPTM and software request
- Divider – DIV
  - Signed/unsigned 32-bit divider
  - Operation in 8 clock cycles, load in 1 clock cycle
  - Division by zero error flag
- LED Controller – LEDC
  - Supports 8-segment digital displays up to a maximum of N
    - ◆ For the HT32F54231/HT32F54241, N = 8
    - ◆ For the HT32F54243/HT32F54253, N = 12
  - Supports 8-segment digital displays with common anode or common cathode
  - Supports frame interrupt
  - Three frequency sources: LSI, LSE and PCLK
  - The LED light on/off times can be controlled using the dead time setting
- Touch Key – TKEY
  - Four key oscillator frequencies: 1 MHz / 3 MHz / 7 MHz / 11 MHz
  - 1024 level reference oscillator internal capacitor for frequency matching
  - Single 16-bit C/F Counter
  - Three scan modes: Manual mode, Auto scan mode and Periodic auto scan mode
  - Support detection in the Sleep, Deep-Sleep1 and Deep-Sleep2 modes
  - Hardware upper or lower threshold comparators
  - Keys are organised into several groups, with each group known as a module
    - ◆ For the HT32F54231/HT32F54241, having a module number, M0 to M5
    - ◆ For the HT32F54243/HT32F54253, having a module number, M0 to M6
  - Each module is a fully independent set of four Touch Keys and each Touch Key has its own oscillator
- Debug Support
  - Serial Wire Debug Port – SW-DP
  - 4 comparators for hardware breakpoint or code/literal patch
  - 2 comparators for hardware watchpoints
- Package and Operation Temperature
  - 28-pin SSOP, 32/46-pin QFN and 48/64-pin LQFP packages
  - Operation temperature range: -40 °C to 85 °C

## Device Information

**Table 1. Features and Peripheral List**

Peripherals		HT32F54231	HT32F54241	HT32F54243	HT32F54253
Main Flash (KB)		32	63	64	127
Option Bytes Flash (KB)		1	1	1	1
SRAM (KB)		4	8	8	16
Timers	MCTM	1		1	
	GPTM	1		1	
	SCTM	2		4	
	BFTM	2		2	
	WDT	1		1	
	RTC	1		1	
Communication	SPI	2		2	
	USART	1		2	
	UART	2		4	
	I <sup>2</sup> C	2		3	
Hardware Divider			1		
CRC-16/32			1		
PDMA		—		6 channels	
EXTI			16		
12-bit ADC			1		
Number of channels			Max. 10 channels		
Comparator		—		2	
GPIO		Up to 40		Up to 54	
LED controller		Up to 8 × 8-segment		Up to 12 × 8-segment	
Touch key		24		28	
CPU frequency			Up to 60 MHz		
Operating voltage			2.5 V ~ 5.5 V		
Operating temperature			-40 °C ~ 85 °C		
Package		28-pin SSOP, 32/46-pin QFN, 48-pin LQFP		32/46-pin QFN, 48/64-pin LQFP	

## Block Diagram

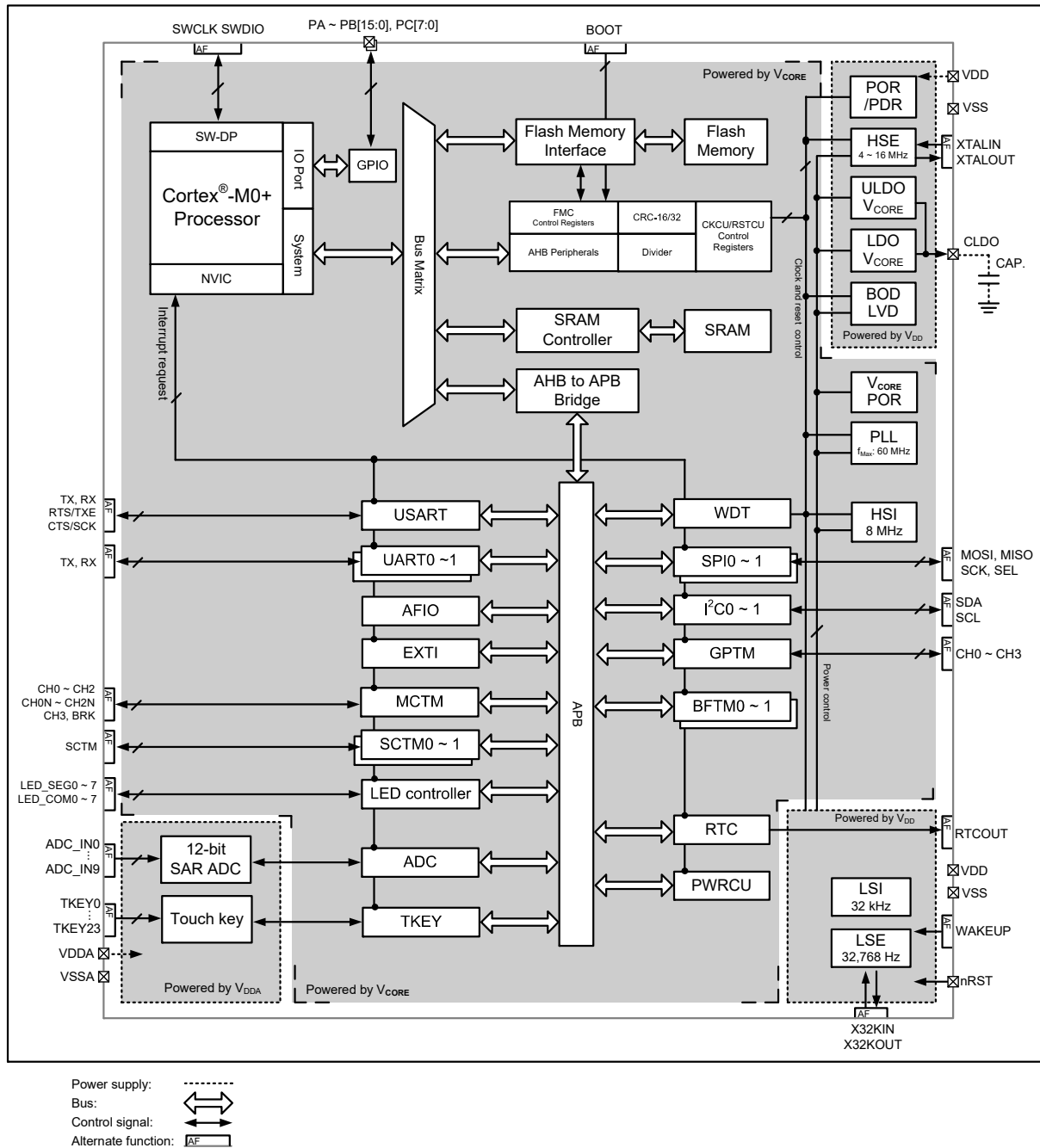


Figure 1. HT32F54231/HT32F54241 Block Diagram





## 2 Document Conventions

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

**Table 2. Document Conventions**

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

## 3 System Architecture

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

### Arm® Cortex®-M0+ Processor

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

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

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

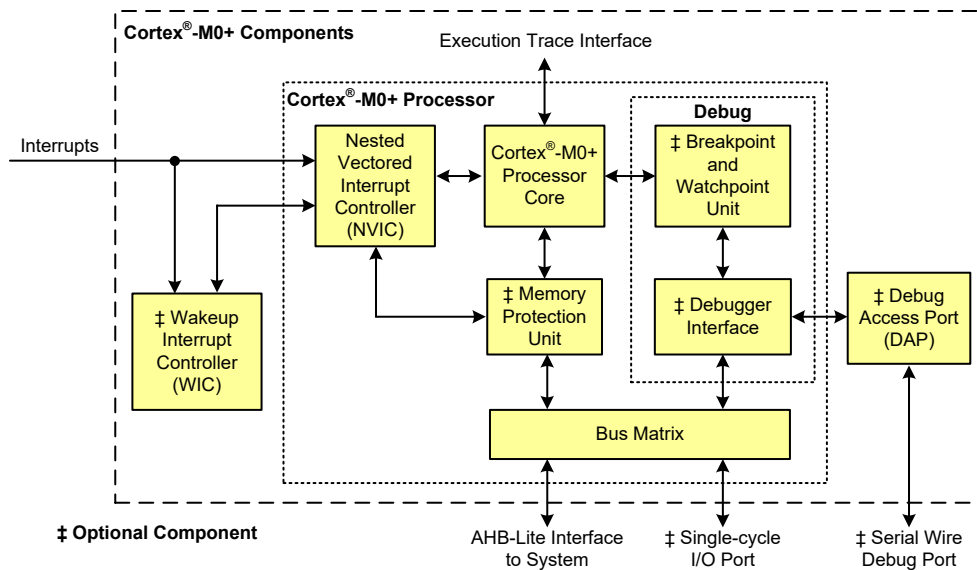
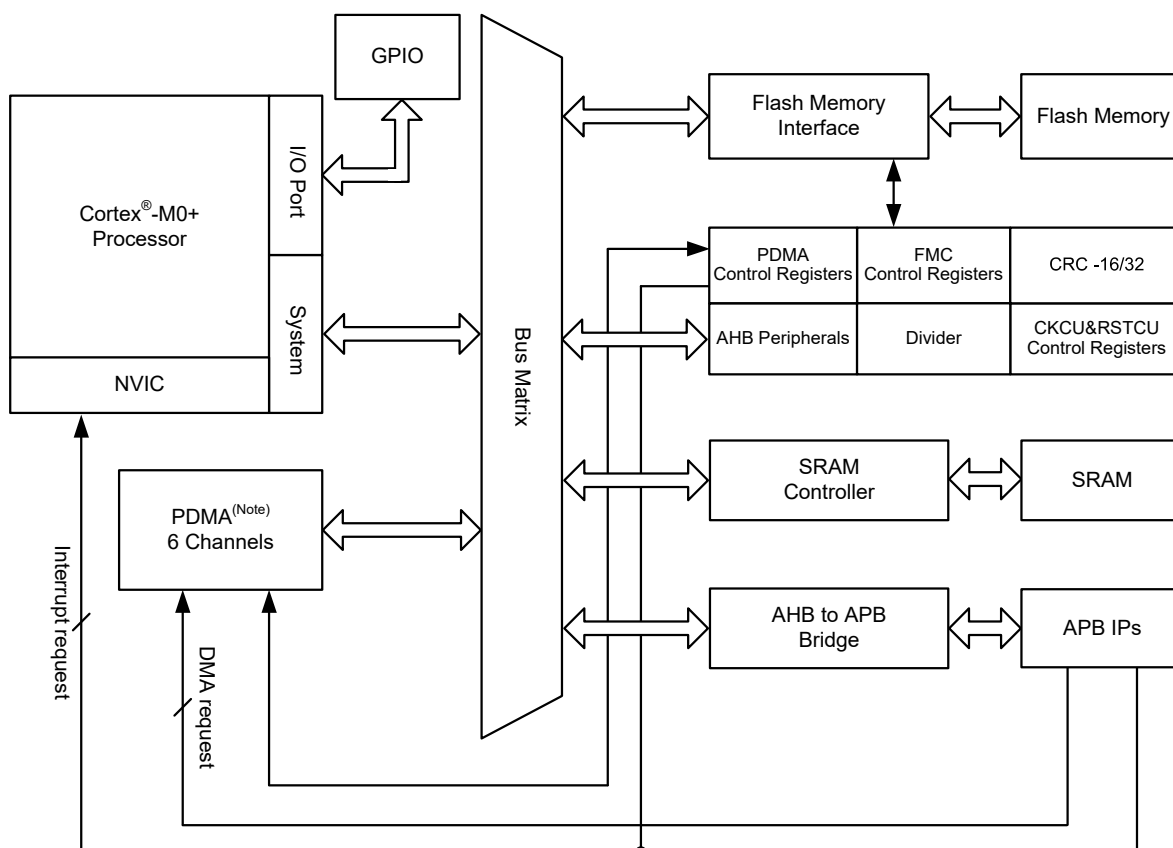


Figure 3. Cortex®-M0+ Block Diagram

## Bus Architecture

The HT32F54231/HT32F54241 series consist of one master and four slaves in the bus architecture. The HT32F54243/HT32F54253 series consist of two masters and four slaves in the bus architecture. The system bus and Peripheral Direct Memory Access (PDMA) are the masters while the internal SRAM access bus, the internal Flash memory access bus, the AHB peripherals access bus and the AHB to APB bridge are the slaves. The PDMA is only available for the HT32F54243/HT32F54253 devices. The single 32-bit AHB-Lite system interface provides simple integration to all system regions including the internal SRAM region and the peripheral region. All of the master buses are based on 32-bit Advanced High-performance Bus-Lite (AHB-Lite) protocol. The following figure shows the bus architecture of the HT32F54231/HT32F54241/HT32F54243/HT32F54253 device series.



Note: The PDMA is only available for the HT32F54243/HT32F54253 devices.

**Figure 4. Bus Architecture**

## Memory Organization

The Arm® Cortex®-M0+ processor access and debug access share the single external interface to external AHB peripherals. The processor access takes priority over debug access. The maximum address range of the Cortex®-M0+ is 4 GB since it has 32-bit bus address width. Additionally, a pre-defined memory map is provided by the Cortex®-M0+ processor to reduce the software complexity of repeated implementation of different device vendors. However, some regions are used by the Arm® Cortex®-M0+ system peripherals. Refer to the Arm® Cortex®-M0+ Technical Reference Manual for more information. The following figure shows the memory map of the HT32F54231/HT32F54241/HT32F54243/HT32F54253 series of devices, including Code, SRAM, peripheral, and other pre-defined regions.

## Memory Map

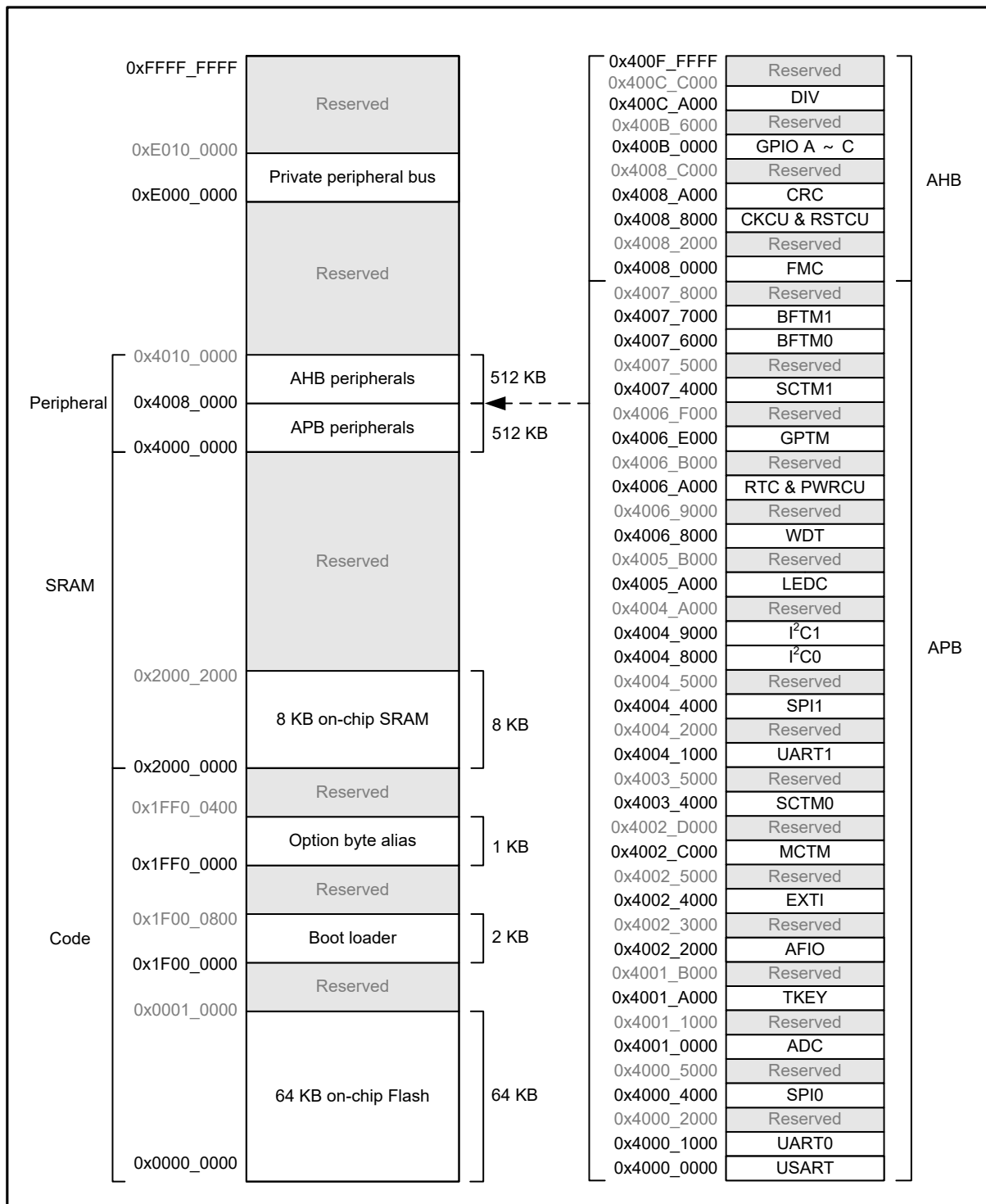


Figure 5. HT32F54231/HT32F54241 Memory Map

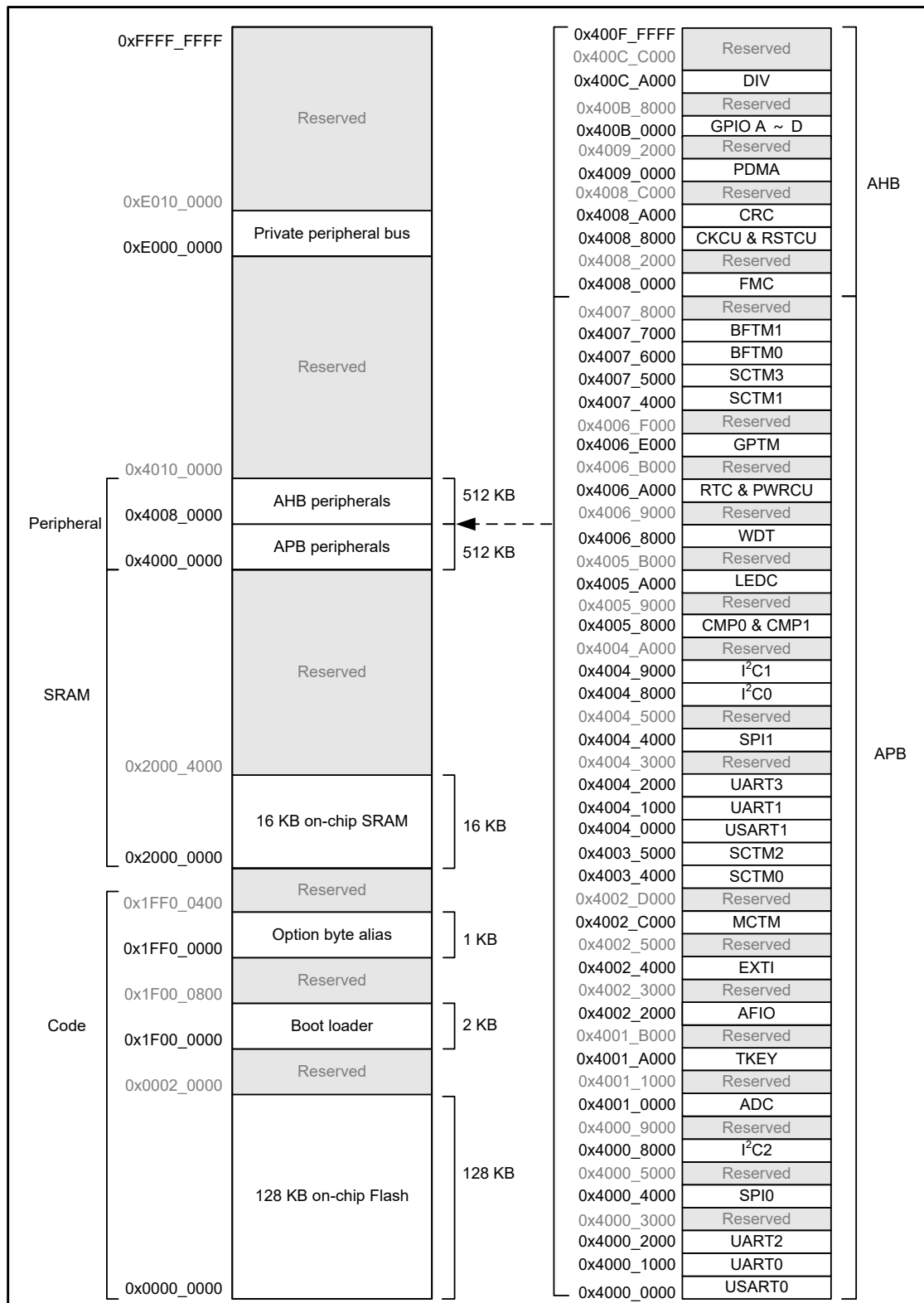


Figure 6. HT32F54243/HT32F54253 Memory Map

**Table 3. HT32F54231/HT32F54241 Register Map**

Start Address	End Address	Peripheral	Bus
0x4000_0000	0x4000_0FFF	USART	APB
0x4000_1000	0x4000_1FFF	UART0	
0x4000_2000	0x4000_3FFF	Reserved	
0x4000_4000	0x4000_4FFF	SPI0	
0x4000_5000	0x4000_FFFF	Reserved	
0x4001_0000	0x4001_0FFF	ADC	
0x4001_1000	0x4001_9FFF	Reserved	
0x4001_A000	0x4001_AFFF	TKEY	
0x4001_B000	0x4002_1FFF	Reserved	
0x4002_2000	0x4002_2FFF	AFIO	
0x4002_3000	0x4002_3FFF	Reserved	
0x4002_4000	0x4002_4FFF	EXTI	
0x4002_5000	0x4002_BFFF	Reserved	
0x4002_C000	0x4002_CFFF	MCTM	
0x4002_D000	0x4003_3FFF	Reserved	
0x4003_4000	0x4003_4FFF	SCTM0	
0x4003_5000	0x4004_0FFF	Reserved	
0x4004_1000	0x4004_1FFF	UART1	
0x4004_2000	0x4004_3FFF	Reserved	
0x4004_4000	0x4004_4FFF	SPI1	
0x4004_5000	0x4004_7FFF	Reserved	
0x4004_8000	0x4004_8FFF	I <sup>2</sup> C0	
0x4004_9000	0x4004_9FFF	I <sup>2</sup> C1	
0x4004_A000	0x4005_9FFF	Reserved	
0x4005_A000	0x4005_AFFF	LEDC	
0x4005_B000	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	
0x4008_C000	0x400A_FFFF	Reserved	
0x400B_0000	0x400B_1FFF	GPIO A	
0x400B_2000	0x400B_3FFF	GPIO B	
0x400B_4000	0x400B_5FFF	GPIO C	
0x400B_6000	0x400C_9FFF	Reserved	
0x400C_A000	0x400C_BFFF	DIV	
0x400C_C000	0x400F_FFFF	Reserved	

**Table 4. HT32F54243/HT32F54253 Register Map**

Start Address	End Address	Peripheral	Bus
0x4000_0000	0x4000_0FFF	USART0	APB
0x4000_1000	0x4000_1FFF	UART0	
0x4000_2000	0x4000_2FFF	UART2	
0x4000_3000	0x4000_3FFF	Reserved	
0x4000_4000	0x4000_4FFF	SPI0	
0x4000_5000	0x4000_7FFF	Reserved	
0x4000_8000	0x4000_8FFF	I <sup>2</sup> C2	
0x4000_9000	0x4000_FFFF	Reserved	
0x4001_0000	0x4001_0FFF	ADC	
0x4001_1000	0x4001_9FFF	Reserved	
0x4001_A000	0x4001_AFFF	TKEY	
0x4001_B000	0x4002_1FFF	Reserved	
0x4002_2000	0x4002_2FFF	AFIO	
0x4002_3000	0x4002_3FFF	Reserved	
0x4002_4000	0x4002_4FFF	EXTI	
0x4002_5000	0x4002_BFFF	Reserved	
0x4002_C000	0x4002_CFFF	MCTM	
0x4002_D000	0x4003_3FFF	Reserved	
0x4003_4000	0x4003_4FFF	SCTM0	
0x4003_5000	0x4003_5FFF	SCTM2	
0x4003_6000	0x4003_FFFF	Reserved	
0x4004_0000	0x4004_0FFF	USART1	
0x4004_1000	0x4004_1FFF	UART1	
0x4004_2000	0x4004_2FFF	UART3	
0x4004_3000	0x4004_3FFF	Reserved	
0x4004_4000	0x4004_4FFF	SPI1	
0x4004_5000	0x4004_7FFF	Reserved	
0x4004_8000	0x4004_8FFF	I <sup>2</sup> C0	
0x4004_9000	0x4004_9FFF	I <sup>2</sup> C1	
0x4004_A000	0x4005_7FFF	Reserved	
0x4005_8000	0x4005_8FFF	CMP0 & CMP1	



Start Address	End Address	Peripheral	Bus
0x4005_9000	0x4005_9FFF	Reserved	APB
0x4005_A000	0x4005_AFFF	LEDC	
0x4005_B000	0x4006_7FFF	Reserved	
0x4006_8000	0x4006_8FFF	WDT	
0x4006_9000	0x4006_9FFF	Reserved	
0x4006_A000	0x4006_AFFF	RTC & PWRCU	
0x4006_B000	0x4006_DFFF	Reserved	
0x4006_E000	0x4006_EFFF	GPTM	
0x4006_F000	0x4007_3FFF	Reserved	
0x4007_4000	0x4007_4FFF	SCTM1	
0x4007_5000	0x4007_5FFF	SCTM3	
0x4007_6000	0x4007_6FFF	BFTM0	
0x4007_7000	0x4007_7FFF	BFTM1	
0x4007_8000	0x4007_FFFF	Reserved	
0x4008_0000	0x4008_1FFF	FMC	AHB
0x4008_2000	0x4008_7FFF	Reserved	
0x4008_8000	0x4008_9FFF	CKCU & RSTCU	
0x4008_A000	0x4008_BFFF	CRC	
0x4008_C000	0x400A_FFFF	Reserved	
0x4009_0000	0x4009_1FFF	PDMA_REG	
0x4009_2000	0x400A_FFFF	Reserved	
0x400B_0000	0x400B_1FFF	GPIO A	
0x400B_2000	0x400B_3FFF	GPIO B	
0x400B_4000	0x400B_5FFF	GPIO C	
0x400B_6000	0x400B_7FFF	GPIO D	
0x400B_8000	0x400C_9FFF	Reserved	
0x400C_A000	0x400C_BFFF	DIV	
0x400C_C000	0x400F_FFFF	Reserved	

## Embedded Flash Memory

The HT32F54231/HT32F54241/HT32F54243/HT32F54253 device series provide an up to 128 KB on-chip Flash memory which is located at address 0x0000\_0000. It supports byte, half-word and word access operations. Note that the Flash memory only supports read operations for the bus access. Any write operations to the Flash memory will cause a bus fault exception. The Flash memory has up to 128 pages. Each page has a memory capacity of 1 KB and can be erased independently. A 32-bit programming interface provides the capability of changing bits from 1 to 0. A data storage or firmware upgrade can be implemented using several methods such as In System Programming (ISP), In Application Programming (IAP) or In Circuit Programming (ICP). For more information, refer to the Flash Memory Controller section.

## Embedded SRAM Memory

The HT32F54231/HT32F54241/HT32F54243/HT32F54253 device series contain an up to 16 KB on-chip SRAM which is located at address 0x2000\_0000. It supports byte, half-word and word access operations.

## AHB Peripherals

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

## APB Peripherals

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

# 4 Flash Memory Controller (FMC)

## Introduction

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

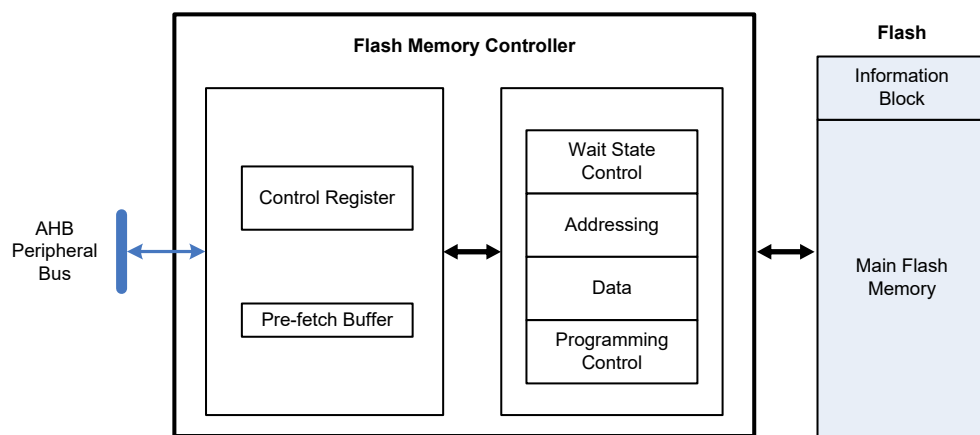


Figure 7. Flash Memory Controller Block Diagram

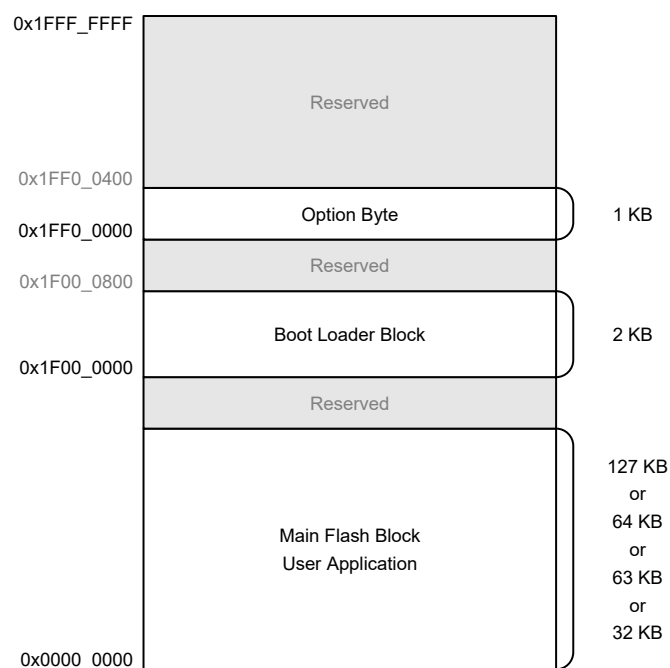
## Features

- Up to 128 KB of on-chip Flash memory for storing instruction/data and option bytes
  - 128 KB (instruction/data + Option Byte) for the HT32F54253
  - 64 KB (instruction/data + Option Byte) for the HT32F54241 and HT32F54243
  - 32 KB (instruction/data + Option Byte) for the HT32F54231
- Page size of 1 KB, totally up to 128 pages depending on the main Flash size
- Wide access interface with a pre-fetch buffer to reduce instruction gaps
- Page erase and mass erase capability
- 32-bit word programming
- Interrupt function to indicate end of Flash memory operations or an error occurrence
- Flash read protection to prevent illegal code/data access
- Page erase/program protection to prevent unexpected operations

## Functional Descriptions

### Flash Memory Map

The following figure is the Flash memory map of the system. The address ranges from 0x0000\_0000 to 0x1FFF\_FFFF (0.5 GB). The address from 0x1F00\_0000 to 0x1F00\_07FF is mapped to the Boot Loader Block with a capacity of 2 KB. Additionally, the region addressed from 0x1FF0\_0000 to 0x1FF0\_03FF is the alias of the Option Byte block with a capacity of 1 KB, which physically locates at the last page of the main Flash. The memory mapping on system view is shown below.



**Figure 8. Flash Memory Map**

## Flash Memory Architecture

The Flash memory consists of up to 128 KB main Flash Block with 1 KB per page and 2 KB Information Block for Boot Loader. The main Flash memory contains a total of 128 pages (or 64 pages for 64 KB device and so on) which can be erased individually. The following table shows the base address, size, and protection setting bit of each page.

**Table 5. Flash Memory and Option Byte**

Block	Name	Address	Page Protection Bit	Size
Main Flash Block	Page 0	0x0000_0000 ~ 0x0000_03FF	OB_PP [0]	1 KB
	Page 1	0x0000_0400 ~ 0x0000_07FF	OB_PP [1]	1 KB
	Page 2	0x0000_0800 ~ 0x0000_0BFF	OB_PP [2]	1 KB
	Page 3	0x0000_0C00 ~ 0x0000_0FFF	OB_PP [3]	1 KB
	:	:	:	:
	:	:	:	:
	Page 124	0x0001_F000 ~ 0x0001_F3FF	OB_PP [124]	1 KB
	Page 125	0x0001_F400 ~ 0x0001_F7FF	OB_PP [125]	1 KB
	Page 126	0x0001_F800 ~ 0x0001_FBFF	OB_PP [126]	1 KB
	Page 127 (Option Byte)	Physical address: 0x0001_FC00 ~ 0x0001_FFFF Alias address: 0x1FF0_0000 ~ 0x1FF0_03FF	OB_CP [1]	1 KB
Information Block	Boot Loader	0x1F00_0000 ~ 0x1F00_07FF	NA	2 KB

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

## Wait State Setting

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

- Before changing the Flash wait state, set the PFBE bit in the CFCR register to 0 to disable the Flash pre-fetch function.
- The HCLK clock is switched from low to high frequency:  
Change the wait state setting first and then switch the HCLK clock.
- The HCLK clock is switched from high to low frequency:  
Switch the HCLK clock first and then change the wait state setting.
- The PFBE bit needs to be restored to 1, if the Flash pre-fetch buffer function is to continue to be used after wait state has been changed.

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

**Table 6. Relationship between Wait State Cycle and HCLK**

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

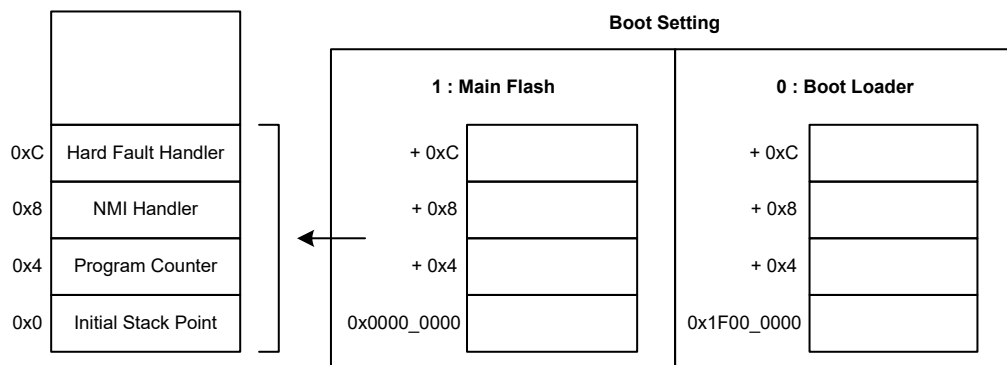
## Booting Configuration

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

**Table 7. Booting Modes**

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

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



**Figure 9. Vector Remapping**

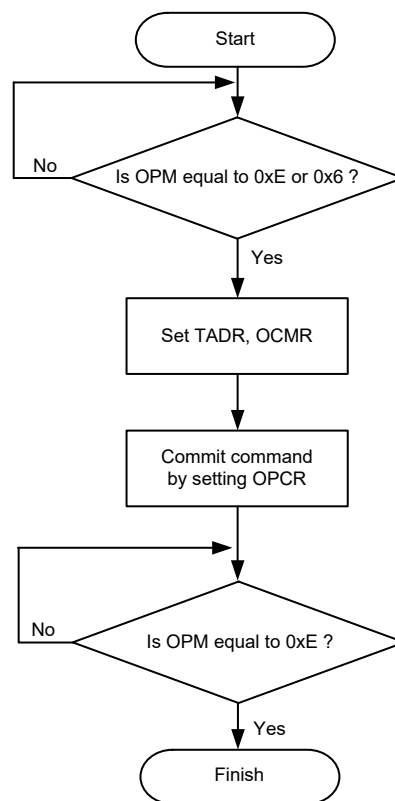
## Page Erase

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

1. Check the OPCR register to confirm that no Flash memory operation is in progress (OPM [3:0] is equal to 0xE or 0x6). Otherwise, wait until the previous operation has been finished.

2. Write the page address to the TADR register.
3. Write the page erase command to the OCMR register (Set CMD [3:0] = 0x8).
4. Commit the page erase command to the FMC by setting the OPCR register (Set OPM [3:0] = 0xA).
5. Wait until all the operations have been completed by checking the value of the OPCR register (OPM [3:0] is equal to 0xE).
6. Read and verify the page if required.

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



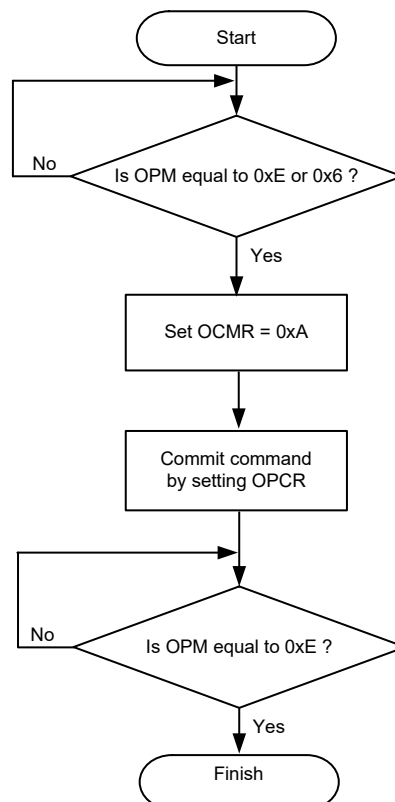
**Figure 10. Page Erase Operation Flowchart**

## Mass Erase

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

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

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



**Figure 11. Mass Erase Operation Flowchart**

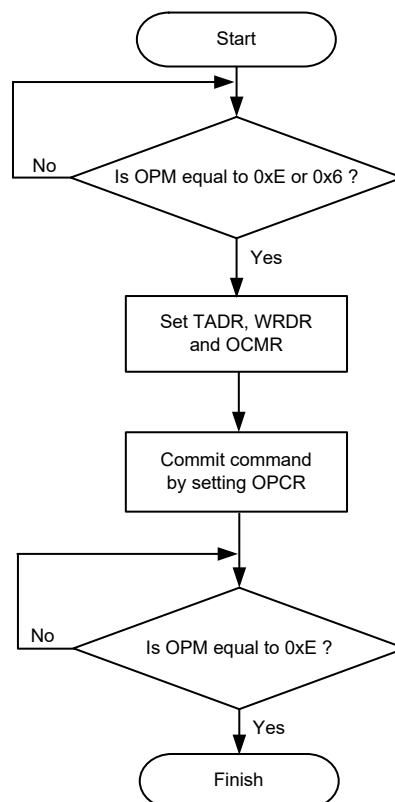


## Word Programming

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

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

Note that the word programming operation 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 the protected pages. When this occurs, the OREF bit will be set by the FMC and then a Flash Operation Error interrupt will be generated if the OREIEN bit in the OIER register is set. The software can check the PPEF bit in the OISR register to detect this condition in the interrupt handler. The following figure shows the word programming operation flow.



**Figure 12. Word Programming Operation Flowchart**

## Option Byte Description

The Option Byte area can be treated as an independent Flash memory of which the base address is 0x1FF0\_0000. The following table shows the functional description and the Option Byte memory map.

**Table 8. Option Byte Memory Map**

Option Byte	Offset	Description	Reset Value
<b>Option Byte Base Address = 0x1FF0_0000</b>			
OB_PP	0x000 0x004 0x008 0x00C	Flash Page Erase/Program Protection OB_PP [n] (n = 0 ~ X) 0: Flash Page n Erase / Program Protection is enabled 1: Flash Page n Erase / Program Protection is disabled OB_PP [n] (n = (X + 1) ~ 127) Reserved For the HT32F54231 the variable "X" is equal to 31. For the HT32F54241 the variable "X" is equal to 62. For the HT32F54243 the variable "X" is equal to 63. For the HT32F54253 the variable "X" is equal to 126.	0xFFFF_FFFF 0xFFFF_FFFF 0xFFFF_FFFF 0xFFFF_FFFF
OB_CP	0x010	Flash Security Protection OB_CP [0] 0: Flash Security protection is enabled 1: Flash Security protection is disabled Option Byte Protection OB_CP [1] 0: Option Byte protection is enabled 1: Option Byte protection is disabled OB_CP [31:2]: Reserved	0xFFFF_FFFF
OB_CK	0x020	Flash Option Byte Checksum OB_CK [31:0] OB_CK should be set as the sum of 5 words of Option Byte content, of which the offset address ranges from 0x000 to 0x010 (0x000 + 0x004 + 0x008 + 0x00C + 0x010), when the OB_PP or OB_CP register content is not equal to 0xFFFF_FFFF. Otherwise, both page erase/program protection and security protection will be enabled.	0xFFFF_FFFF
OB_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 a page erase/program protection function to prevent unexpected operations on the protected Flash memory area. The page erase (CMD [3:0] = 0x8 in the OCMR register) or word programming (CMD [3:0] = 0x4) command will not be accepted by the FMC on the protected pages. When the page erase or word programming command aimed at the protected pages is sent to the FMC, the PPEF bit in the OISR register will then be set by the FMC and the Flash Operation Error interrupt will be triggered to inform the CPU if the OREIEN bit in the OIER register is set. The page protection function can be individually enabled for each page by configuring the OB\_PP registers in the Option Byte area. The following table shows the access permission of the main Flash page when the page protection is enabled.

**Table 9. Access Permission of Protected Main Flash Page**

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

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

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

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

## Security Protection

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

**Table 10. Access Permission When Security Protection is Enabled**

Operation \ Mode	User Application <sup>(1)</sup>	ICP/Debug Mode
Read	O	X (read as 0)
Programming	O <sup>(1)</sup>	X
Page Erase	O <sup>(1)</sup>	X
Mass Erase	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 area and page 0 are still under protection, where the Programming/Page Erase operations are not accepted.

2. The Mass Erase operation can erase the Option Byte area and disable the security protection.

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

1. Check the OPCR register to confirm that no Flash memory operation is in progress (OPM [3:0] is equal to 0xE or 0x6). Otherwise, wait until the previous operation has been finished.
2. Write the OB\_CP address to the TADR register (Set TADR = 0x1FF0\_0010).
3. Write data to the WRDR register to set OB\_CP [0] to 0.
4. Write the word programming command to the OCMR register (Set CMD [3:0] = 0x4).
5. Commit the word programming command to the FMC by setting the OPCR register (Set OPM = 0xA).
6. Wait until all the operations have been finished by checking the value of the OPCR register (OPM [3:0] is equal to 0xE).
7. Read and verify the Option Byte if required.
8. The OB\_CK field in the Option Byte area must be updated according to the Option Byte checksum rule.
9. Apply a system reset to active the new OB\_CP setting.

## Register Map

The following table shows the FMC registers and reset values.

**Table 11. FMC Register Map**

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

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

## Register Descriptions

### Flash Target Address Register – TADR

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

Offset: 0x000

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[31:0]	TADB	<p>Flash Target Address Bits</p> <p>For programming operations, the TADR register specifies the address where the data is written to. Since the programming length is 32-bit, the TADR register should be set as word-aligned (4 bytes). The TADB [1:0] bits will be ignored during programming operations. For page erase operations, the TADR register contains the page address which is to be erased. Since the page size is 1 KB, the TADB [9:0] bits will be ignored in order to limit the target address as 1 Kbyte-aligned. For 128 KB main Flash addressing, TADB [31:17] should be zero and TADB [31:16] should be zero for 64 KB and so on. The region of which the address ranges from 0x1FF0_0000 to 0x1FF0_03FF is the 1 KB Option Byte. This field for available Flash address must be within the range of 0x0000_0000 to 0x1FFF_FFFF. Otherwise, an Invalid Target Address interrupt will be generated if the corresponding interrupt enable bit is set.</p>

## Flash Write Data Register – WRDR

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

Offset: 0x004

Reset value: 0x0000\_0000

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

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

## Flash Operation Command Register – OCMR

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

Offset: 0x00C

Reset value: 0x0000\_0000

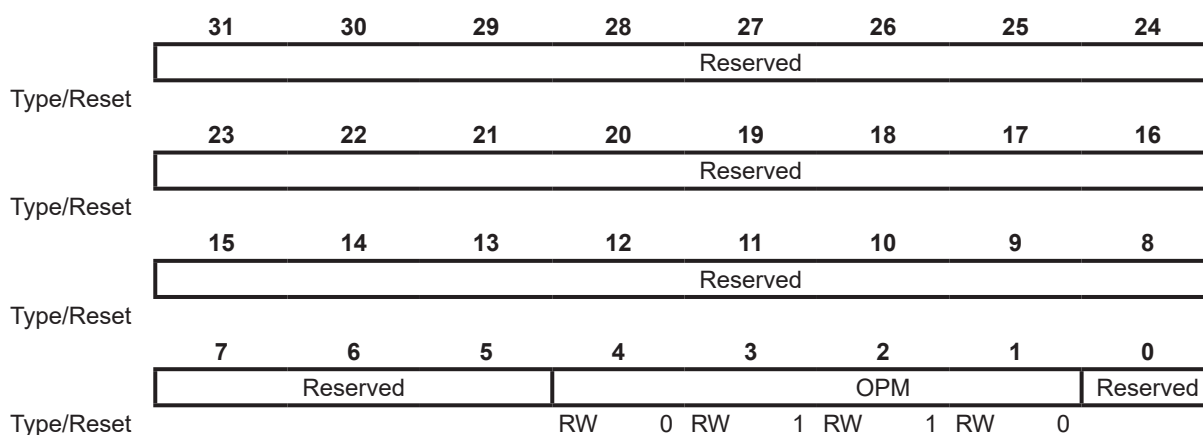
	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved				CMD			
					RW	0	RW	0
						RW	0	RW
							RW	0
								RW
								0

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



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

Reset value: 0x0000 000C



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

## Flash Operation Interrupt Enable Register – OIER

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

Offset: 0x014

Reset value: 0x0000\_0000

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

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

## 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 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 The RORFF 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 high 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 IOCMIE bit in the OIER register is set. Reset this bit by writing 1.

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

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

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

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

Reset value: 0xFFFF\_FFFF

	31	30	29	28	27	26	25	24	
	PPSBn								
Type/Reset	RO	X RO	X RO	X RO	X RO	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 page is protected</p> <p>1: The corresponding page is not protected</p> <p>The content of this register is not dynamically updated and will only be reloaded from the Option Byte when any kind of reset occurs. The erase or program function of the specific pages is not allowed when the corresponding bits of the PPSR registers are reset. The reset value of PPSR [127:0] is determined by the Option Byte OB_PP [127:0] bits. Since the maximum page number of the main Flash is various and dependent on the chip specification. Therefore, the every page erase/program protection status bit may protect one or two pages, depending on the chip specification. Other bits of the OB_PP and PPSR registers are reserved for future use.</p>

## Flash Security Protection Status Register – CPSR

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

Offset: 0x030

Reset value: 0x0000\_000X

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

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

## Flash Vector Mapping Control Register – VMCR

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

Offset: 0x100

Reset value: 0x0000\_000X

	31	30	29	28	27	26	25	24
Type/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	Reserved
							RW	X

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

## Flash Manufacturer and Device ID Register – MDID

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

Offset: 0x180

Reset value: 0x0376\_XXXX

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

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



## Flash Page Number Status Register – PNSR

This register indicates the Flash memory page number.

Offset: 0x184

Reset value: 0x0000\_00XX

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

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

## Flash Page Size Status Register – PSSR

This register indicates the page size in bytes.

Offset: 0x188

Reset value: 0x0000\_0400

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

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

## Device ID Register – DIDR

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

Offset: 0x18C

Reset value: 0x000X\_XXXX

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

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

### Flash Pre-fetch Control Register – CFCR

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

Offset: 0x200

Reset value: 0x0000\_0011

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

Bits	Field	Descriptions
[4]	PFBE	Pre-fetch Buffer Enable Bit 0: Pre-fetch buffer is disabled 1: Pre-fetch buffer is enabled The pre-fetch buffer is enabled by default setting. When the pre-fetch buffer is disabled, the instruction and data are directly provided by the Flash memory.
[2:0]	WAIT	Flash Wait State Setting The WAIT[2:0] field is used to set the HCLK wait clock during a non-sequential Flash access. The actual wait clock is given by (WAIT[2:0] - 1). It must disable pre-fetch buffer function before changing flash wait state, then restore the pre-fetch buffer presetting after wait state has been changed. Since a wide access interface with a pre-fetch buffer is provided, the wait state of sequential Flash access is very close to zero.

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

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

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

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

Reset value: 0XXXXX\_XXXX – 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 and Deep-Sleep2 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 shadow in the following figure indicates the power supply source of two digital power domains.

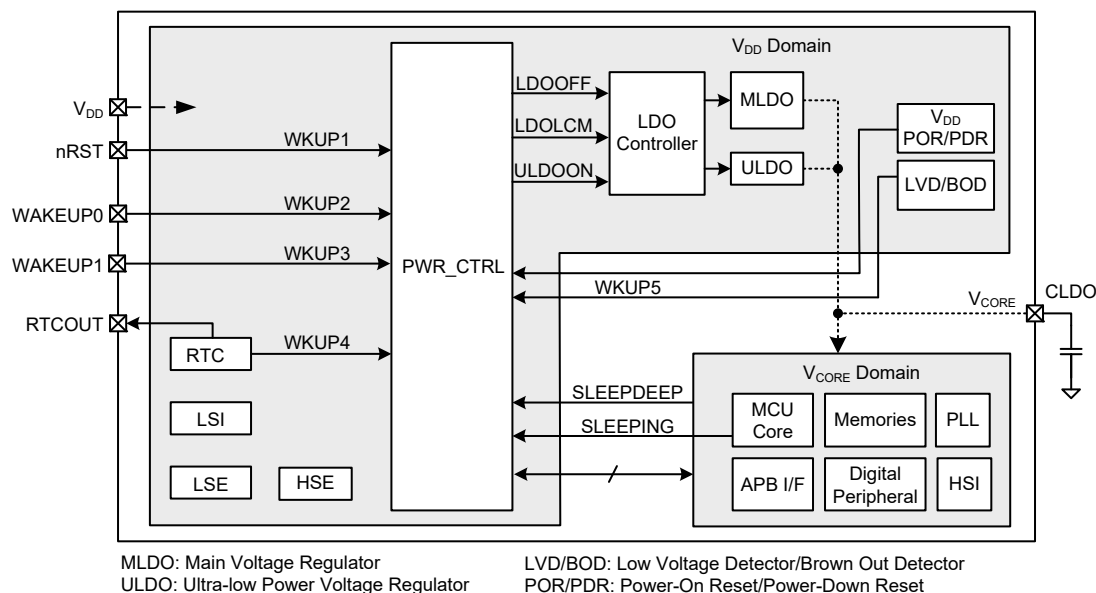


Figure 13. PWRCU Block Diagram

### Features

- Two power domains:  $V_{DD}$  and  $V_{CORE}$  power domains
- Three power saving modes: Sleep, Deep-Sleep1 and Deep-Sleep2 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)
  - 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<sub>DD</sub> Power Domain

#### LDO Power Control

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

- 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, EXTI wakeup and WAKEUPn pin wakeup
- 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<sub>CORE</sub> power.

#### Voltage Regulator

The main voltage regulator, LDO, ultra-low power LDO, ULDO, Low voltage Detector, LVD and High Speed External Crystal oscillator, HSE, are operated under the V<sub>DD</sub> power domain. The main LDO can be configured to operate in normal mode (LDOOFF = 0, LDOLCM = 0, I<sub>OUT</sub> = high current mode) and the ultra-low power LDO can be configured to operate in low current mode (LDOOFF = 0, LDOLCM = 1, I<sub>OUT</sub> = low current mode) to supply the V<sub>CORE</sub> 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<sub>CORE</sub> power domain. It is controlled using the ULDOON bit in the PWRCR register.

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

The devices have an integrated POR/PDR circuitry that allows proper operation starting from V<sub>POR</sub> or down to V<sub>PDR</sub>. For more details concerning the power-on/power-down reset threshold voltage, refer to the electrical characteristics of the corresponding datasheet.

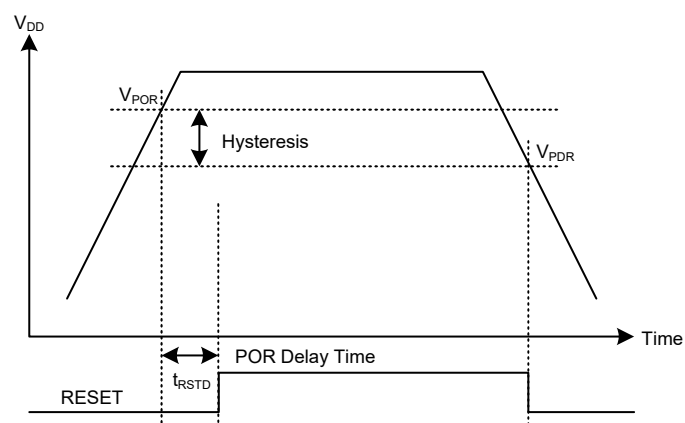


Figure 14. Power-On Reset / Power-Down Reset Waveform

### 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 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.

### $V_{CORE}$ Power Domain

The main functions that include the APB interface for the  $V_{DD}$  domain, MCU core logic, AHB/APB peripherals and memories and so on are located in this power domain. Once the  $V_{CORE}$  is powered up, the POR will generate a reset sequence on the  $V_{CORE}$  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  $V_{CORE}$  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.

## 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 to APBPCSR2 registers to meet the application requirement. Reducing the system clock speed before entering the sleep mode will also help to minimize power consumption.

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

**Table 12. Operation Mode Definitions**

Mode Name	Hardware Action
Run	After system reset, CPU fetches instructions to execute.
Sleep	1. CPU clock will be stopped. 2. Peripherals, Flash and SRAM clocks can be stopped by setting.
Deep-Sleep1 ~ 2	1. Stop all clocks in the V <sub>CORE</sub> power domain. 2. Disable HSI, HSE and PLL. 3. Turn on the ultra-low power ULDO to reduce the V <sub>CORE</sub> power domain current.

### 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 bit 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 13. 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 CMP wakeup <sup>(3)</sup> or LVD wakeup <sup>(2)</sup> or WAKEUPn pin wakeup
Deep-Sleep2		1	X	1	Any EXTI in event mode or RTC wakeup or CMP <sup>(3)</sup> wakeup or LVD wakeup <sup>(2)</sup> or WAKEUPn pin wakeup

- Notes: 1. Wakeup event means EXTI line trigger event, RTC event, LVD event or WAKEUPn pin wakeup.  
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.  
3. CMP wakeup is only available for the HT32F54243/HT32F54253 devices.

### 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 WAKEUPn 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.

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

PORF	PORSTF	Description
1	1	Power-up for the first time after the $V_{DD}$ 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	1	Restart from unexpected loss of the $V_{CORE}$ power or other reset (nRST, WDT, ...)

### WAKEUPn Pin Wakeup

The software can set the WUPnEN bit in register PWRCR to 1 to enable the WAKEUPn pin function before entering the power saving mode, waiting for a wakeup trigger signal occurrence on the WAKEUPn pin to wake up the system from the power saving mode. The external WAKEUPn pin interrupt shares the same exception number with the EXTI event wakeup interrupt. The software can set the EXTI Event Wakeup Interrupt Enable bit (EVWUPIEN) to 1 to assert the WKUP interrupt in the NVIC unit when both the WUPnEN and WUPFn bits are set to 1.

Although the WUPnEN bit is located in the V<sub>DD</sub> domain, it also can be reset by the nRST reset pin. After the reset there will be a delay before the WUPnEN bit is active. This bit will not be active until the system reset is finished and the V<sub>DD</sub> domain ISO signal is disabled. This means that the bit cannot be immediately set by software after a system reset is finished and the V<sub>DD</sub> domain ISO signal is disabled. The delay time requires at least three 32 kHz clock periods after the WUPnEN bit reset has been finished.

## Register Map

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

**Table 15. PWRCU Register Map**

Register	Offset	Description	Reset Value
PWRSR	0x100	Power Control Status Register	0x0000_0010
PWRCR	0x104	Power Control Register	0x0000_0000
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 power control status.

Offset: 0x100

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	Reserved						WUPF1	WUPF0
	7	6	5	4	3	2	1	0
Type/Reset	Reserved			PORF	Reserved			
				RC	1			

Bits	Field	Descriptions
[9]	WUPF1	External WAKEUP1 Pin Flag 0: The WAKEUP1 pin is not asserted 1: The WAKEUP1 pin is asserted This bit is set by hardware when the WAKEUP1 pin asserts and is cleared by software read. Software should read this bit to clear it after a system wake up from the power saving mode.
[8]	WUPF0	External WAKEUP0 Pin Flag 0: The WAKEUP0 pin is not asserted 1: The WAKEUP0 pin is asserted This bit is set by hardware when the WAKEUP0 pin asserts and is cleared by software read. Software should read this bit to clear it after a system wake up from the power saving mode.
[4]	PORF	Power On Reset Flag 0: V <sub>CORE</sub> Power Domain reset does not occur 1: V <sub>CORE</sub> Power Domain reset occurs This bit is set by hardware when V <sub>CORE</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>CORE</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.

## Power Control Register – PWRCR

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

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				WUP1TYPE		WUP0TYPE		
					RW	0	RW	0	
	15	14	13	12	11	10	9	8	
Type/Reset	ULDOSTS	Reserved				WUP1EN	Reserved	WUP0EN	
	RO	0				RW	0	RW	0
	7	6	5	4	3	2	1	0	
Type/Reset	ULDOON	Reserved			LDOOFF	LDOLCM	Reserved	PWCRST	
	RW	0			RW	0		WO	0

Bits	Field	Descriptions															
[19:18]	WUP1TYPE	WAKEUP1 Signal Trigger Type <table border="1"> <thead> <tr> <th colspan="2">WUP1TYPE [1:0]</th><th>WAKEUP1 Signal Trigger Type</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>Positive-edge Triggered</td></tr> <tr> <td>0</td><td>1</td><td>Negative-edge Triggered</td></tr> <tr> <td>1</td><td>0</td><td>High-level Sensitive</td></tr> <tr> <td>1</td><td>1</td><td>Low-level Sensitive</td></tr> </tbody> </table>	WUP1TYPE [1:0]		WAKEUP1 Signal Trigger Type	0	0	Positive-edge Triggered	0	1	Negative-edge Triggered	1	0	High-level Sensitive	1	1	Low-level Sensitive
WUP1TYPE [1:0]		WAKEUP1 Signal Trigger Type															
0	0	Positive-edge Triggered															
0	1	Negative-edge Triggered															
1	0	High-level Sensitive															
1	1	Low-level Sensitive															
[17:16]	WUP0TYPE	WAKEUP0 Signal Trigger Type <table border="1"> <thead> <tr> <th colspan="2">WUP0TYPE [1:0]</th><th>WAKEUP0 Signal Trigger Type</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>Positive-edge Triggered</td></tr> <tr> <td>0</td><td>1</td><td>Negative-edge Triggered</td></tr> <tr> <td>1</td><td>0</td><td>High-level Sensitive</td></tr> <tr> <td>1</td><td>1</td><td>Low-level Sensitive</td></tr> </tbody> </table>	WUP0TYPE [1:0]		WAKEUP0 Signal Trigger Type	0	0	Positive-edge Triggered	0	1	Negative-edge Triggered	1	0	High-level Sensitive	1	1	Low-level Sensitive
WUP0TYPE [1:0]		WAKEUP0 Signal Trigger Type															
0	0	Positive-edge Triggered															
0	1	Negative-edge Triggered															
1	0	High-level Sensitive															
1	1	Low-level Sensitive															
[15]	ULDOSTS	Ultra-low Power Regulator Status This bit is set to 1 if the ULDOON bit in this register has been set to 1 after MCU is wakeup from the Deep-Sleep2 mode. This bit is cleared to 0 if the ULDOON bit has been set to 0 or if a POR/PDR reset occurred.															
[10]	WUP1EN	External WAKEUP1 Pin Enable 0: Disable WAKEUP1 pin function 1: Enable WAKEUP1 pin function The software can set the WUP1EN bit as 1 to enable the WAKEUP1 pin function before entering the power saving mode. When WUP1EN = 1, a change on the WAKEUP1 pin wakes up the system from the power saving mode. If the WAKEUP1 pin is active high, this bit will set an input pull down mode.															

Bits	Field	Descriptions
[8]	WUP0EN	<p>External WAKEUP0 Pin Enable</p> <p>0: Disable WAKEUP0 pin function 1: Enable WAKEUP0 pin function</p> <p>The software can set the WUP0EN bit as 1 to enable the WAKEUP0 pin function before entering the power saving mode. When WUP0EN = 1, a change on the WAKEUP0 pin wakes up the system from the power saving mode. If the WAKEUP0 pin is active high, this bit will set an input pull down mode.</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>
[3]	LDOOFF	<p>Main regulator Operating Mode Control</p> <p>0: The LDO operates in a low current mode when MCU enters the Deep-Sleep mode (SLEEPDEEP = 1). That means MCU <math>V_{CORE}</math> power is available and supplied by 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 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 operated in the run mode. The ULDO output current capability is limited at 5 mA below and lower static current. It is suitable for MCU is operated at lower speed system clock to get a lower current consumption. This bit will be clear to 0 when the MCU is into <math>V_{DD}</math> power domain reset or the nRST pin is occurred reset signal.</p>
[0]	PWCURST	<p>Power Control Unit Software Reset</p> <p>0: No action 1: Power Control Unit Software Reset is activated.</p> <p>When this bit is set, it will reset all the related RTC and PWRCU registers.</p>

## 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

	31	30	29	28	27	26	25	24								
Type/Reset	Reserved															
	23	22	21	20	19	18	17	16								
Type/Reset	Reserved	LVDS [2]	LVDEWEN	LVDIWEN	LVDF	LVDS [1:0]		LVDEN								
		RW	0	RW	0	RO	0	RW	0	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	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 result in the LVDF bit being asserted. If the system requires to be woken up from the Deep-Sleep 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 MCU NVIC unit.
[19]	LVDF	Low Voltage Detect Status Flag 0: $V_{DD}$ is higher than the specific voltage level 1: $V_{DD}$ 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 MCU 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 LVDS bits. Therefore when the LVD function is enabled before the system is into the Deep-Sleep2 (ULDO is turn on and LDO is power down), the LVDEWEN bit has to be enabled to avoid the LDO does not activate in the meantime when the MCU 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

Power Control LDO Status Register – PWRLDOSR

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 RO
	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 clocks of AHB, APB, and CPU are derived from system clock (CK\_SYS) which can come from HSI, HSE, LSI, LSE or PLL. Watchdog Timer and Real Time Clock (RTC) use either LSI or LSE as their clock source. The maximum operating frequency of system clock  $f_{CK\_AHB}$  can be up to 60 MHz.

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



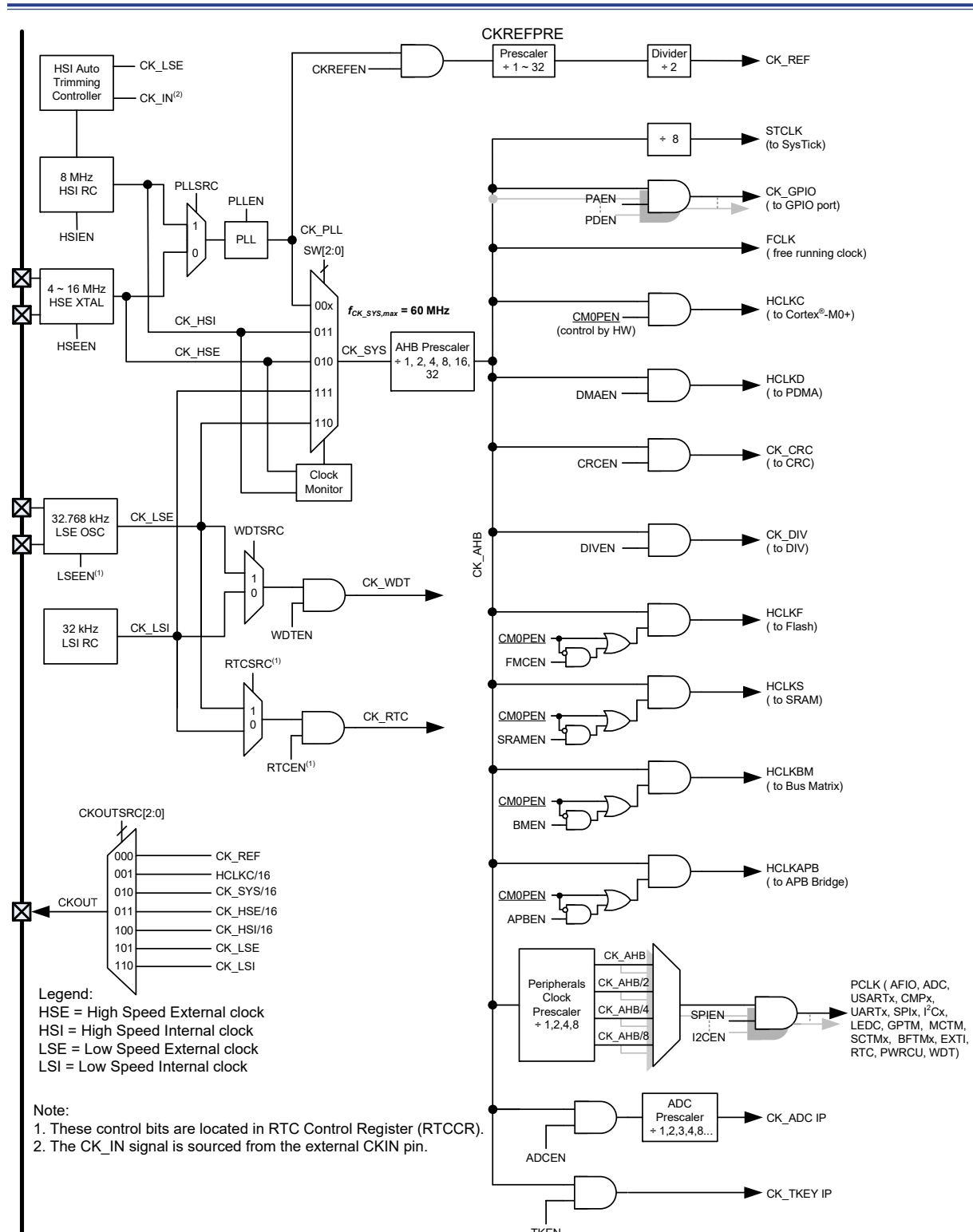


Figure 15. CKCU Block Diagram

## Features

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

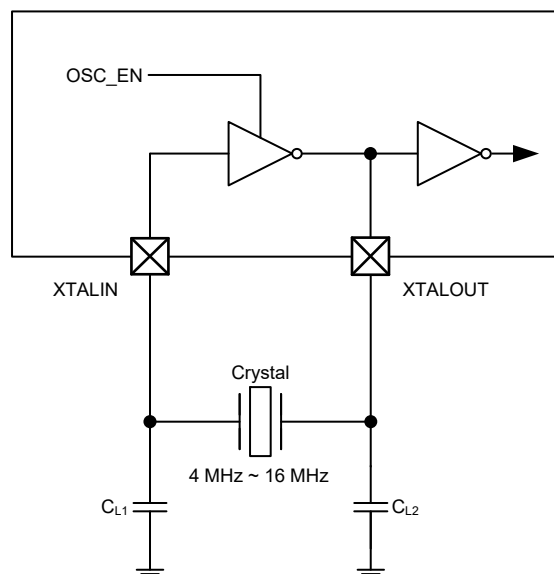
## Functional Descriptions

### High Speed External Crystal Oscillator – HSE

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

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

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



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

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

## High Speed Internal RC Oscillator – HSI

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

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

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

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

The frequency accuracy of the high speed internal RC oscillator HSI can vary from one chip to another due to manufacturing process variations, this is why each device is factory calibrated by Holtek for  $\pm 2\%$  accuracy at  $V_{DD} = 5.0\text{ V}$  and  $T_A = 25\text{ }^{\circ}\text{C}$ . But the accuracy is not enough for some application and environment requirements. Therefore, this device provides the trimming mechanism for HSI frequency calibration using a more accurate external reference clock. The detailed block diagram is shown as the following figure.

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

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

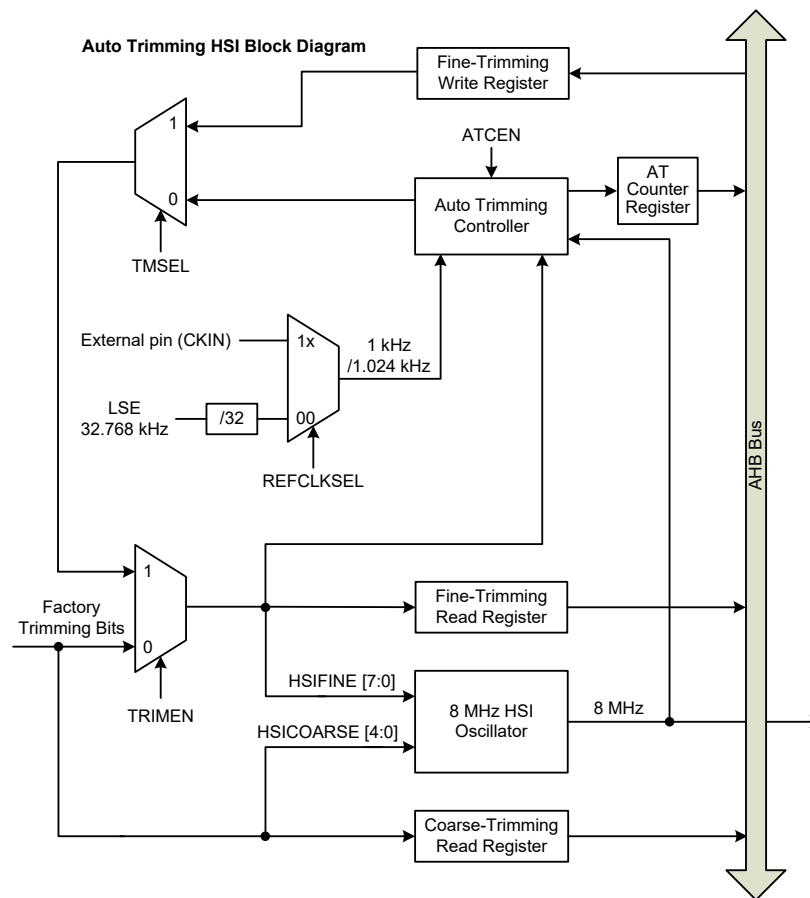


Figure 17. HSI Auto Trimming Block Diagram

## Phase Locked Loop – PLL

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

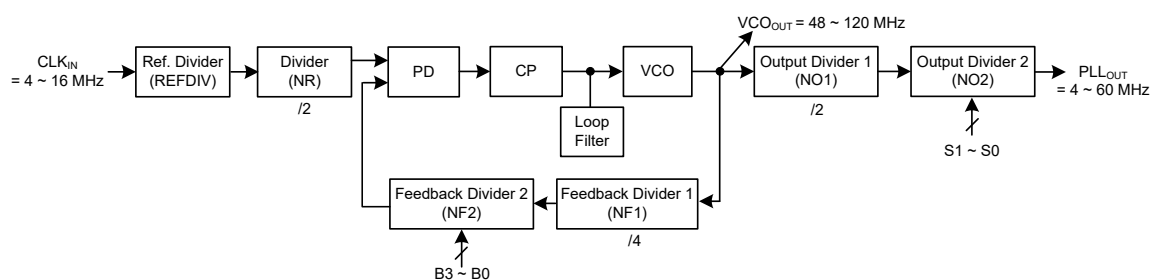


Figure 18. PLL Block Diagram

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 ~ 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<sub>IN</sub> frequency as PLL input generates a specific PLL output frequency; a larger number of NF2 is suggested because it will cause the PLL more stable and less jittered but enlarges the settling time. The output and feedback of divider 2 value are described in Table 16 and Table 17. All the configuration bits (S1 ~ S0, B3 ~ B0) in Table 16 and Table 17 as well as the Bypass mode control are defined in the PLL Configuration Register (PLLCFGR) and PLL Control Register (PLLCR) in the section of Register Definition. Note that VCO<sub>OUT</sub> is ranged from 48 MHz to 120 MHz. If VCO<sub>OUT</sub> by user's configurations exceeds this range, the output frequency of the PLL will not be promised 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.

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

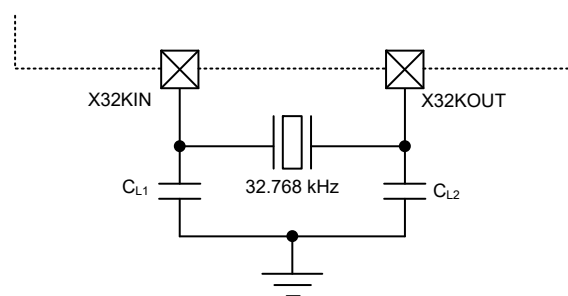
Output Divider 2 Setup Bits S1 ~ S0 (POTD Field in the PLLCFGR Register)	NO2 (Output Divider 2 Value)
00	1
01	2
10	4
11	8

**Table 17. Feedback Divider 2 Value Mapping**

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

## Low Speed External Crystal Oscillator – LSE

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



**Figure 19. External Crystal, Ceramic and Resonators for LSE**

## Low Speed Internal RC Oscillator – LSI

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

## Clock Ready Flag

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

## System Clock (CK\_SYS) Selection

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

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

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

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

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

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

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

Programming guide of system clock selection is listed below.

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

## HSE Clock Monitor

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

## Clock Output Capability

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

**Table 18. CKOUT Clock Source**

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

## Register Map

The following table shows the CKCU registers and reset values.

**Table 19. CKCU Register Map**

Register	Offset	Description	Reset Value
GCFGR	0x000	Global Clock Configuration Register	0x0000_0302
GCCR	0x004	Global Clock Control Register	0x0000_0803
GCSR	0x008	Global Clock Status Register	0x0000_0028
GCIR	0x00C	Global Clock Interrupt Register	0x0000_0000
PLLCFGR	0x018	PLL Configuration Register	0x0000_0000
PLLCR	0x01C	PLL Control Register	0x0000_0000
AHBCFGR	0x020	AHB Configuration Register	0x0000_0000
AHBCCR	0x024	AHB Clock Control Register	0x0000_0065
APBCFGR	0x028	APB Configuration Register	0x0001_0000
APBCCR0	0x02C	APB Clock Control Register 0	0x0000_0000
APBCCR1	0x030	APB Clock Control Register 1	0x0000_0000
CKST	0x034	Clock Source Status Register	0x0100_0003
APBPCSR0	0x038	APB Peripheral Clock Selection Register 0	0x0000_0000
APBPCSR1	0x03C	APB Peripheral Clock Selection Register 1	0x0000_0000
HSICR	0x040	HSI Control Register	0xFFFF_0000 where X is undefined
HSIATCR	0x044	HSI Auto Trimming Counter Register	0x0000_0000
APBPCSR2	0x048	APB Peripheral Clock Selection Register 2	0x0000_0000
MCUDBGCR	0x304	MCU Debug Control Register	0x0000_0000



## Register Descriptions

### Global Clock Configuration Register – GCFGR

This register specifies the low power mode status and the clock source for PLL/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
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	CKREFPRE					Reserved		PLLSRC
Type/Reset	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
	Reserved					CKOUTSRC		
Type/Reset						RW	0	RW
							1	0

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 Others: Reserved Set by hardware. Reset by software writing b11x.
[15:11]	CKREFPRE	CK_REF Clock Prescaler Selection $CK\_REF = CK\_PLL / (CKREFPRE + 1) / 2$ 00000: $CK\_REF = CK\_PLL / 2$ 00001: $CK\_REF = CK\_PLL / 4$ ... 11111: $CK\_REF = CK\_PLL / 64$ Set and reset by software to control the CK_REF clock prescaler setting.
[8]	PLLSRC	PLL Clock Source Selection 0: External 4 ~ 16 MHz crystal oscillator clock is selected (HSE) 1: Internal 8 MHz RC oscillator clock is selected (HSI) Set and reset by software to control the PLL clock source.
[2:0]	CKOUTSRC	CKOUT Clock Source Selection 000: CK_REF is selected, $CK\_REF = CK\_PLL / (CKREFPRE + 1) / 2$ 001: (HCLKC / 16) is selected 010: (CK_SYS / 16) is selected 011: (CK_HSE / 16) is selected 100: (CK_HSI / 16) is selected 101: CK_LSE is selected 110: CK_LSI is selected 111: Reserved Set and reset by software to control the CKOUT clock source.

## Global Clock Control Register – GCCR

This register specifies the clock enable bits.

Offset: 0x004

Reset value: 0x0000\_0803

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

Bits	Field	Descriptions
[17]	PSRCEN	Power Saving Wakeup RC Clock Enable 0: No action 1: Use Internal 8 MHz RC clock (HSI) as system clock after a Deep-Sleep1/2 mode wakeup Software can set PSRCEN high before entering the Deep-Sleep1/2 mode in order to reduce the waiting time after wakeup. When PSRCEN = 1, hardware will select HSI as clock source after the system wakeup from Deep-Sleep1/2 mode. Meanwhile, instruction can start execution since the HSI clock is provided to CPU. After the original clock source, which is selected as CK_SYS before entering the Deep-Sleep1/2 mode, is ready, hardware will switch back the clock source as originally.
[16]	CKMEN	HSE Clock Monitor Enable 0: Disable external 4 ~ 16 MHz crystal oscillator clock monitor 1: Enable external 4 ~ 16 MHz crystal oscillator clock monitor When hardware detects the HSE clock stuck at low/high state, internal hardware will switch the system clock to the internal high speed RC clock (HSI).
[11]	HSIEN	Internal High Speed Clock Enable 0: Internal 8 MHz RC oscillator clock is disabled 1: Internal 8 MHz RC oscillator clock is enabled Set and reset by software. This bit cannot be reset if the HSI clock is used as system clock or PLL input clock.
[10]	HSEEN	External High Speed Clock Enable 0: External 4 ~ 16 MHz crystal oscillator clock is disabled 1: External 4 ~ 16 MHz crystal oscillator clock is enabled Set and reset by software. This bit cannot be reset if the HSE clock is used as system clock or PLL input clock.
[9]	PLEN	PLL Enable 0: PLL off 1: PLL on Set and reset by software to control the PLL. This bit cannot be reset if the PLL clock is used as system clock.

Bits	Field	Descriptions
[8]	HSEGAIN	External High Speed Clock Gain Selection 0: HSE low gain mode 1: HSE high gain mode
[2:0]	SW	System Clock Switch 00x: CK_PLL clock out as system clock 010: CK_HSE as system clock 011: CK_HSI as system clock 110: CK_LSE as system clock 111: CK_LSI as system clock Others: CK_HSI as system clock These bits are used to select the CK_SYS source. If the HSE oscillator is used directly or indirectly as the system clock and the HSE clock monitor function is enabled, once the HSE failure is detected, these bits will be set by hardware to force HSI (b011) as the system clock. Note: When switching the system clock using the SW field, the system clock will not be immediately switched and a certain delay is necessary. Software can monitor the CKSWST field in the clock source status register CKST to make sure which clock is currently used as the system clock.

## Global Clock Status Register – GCSR

This register indicates the clock ready status.

Offset: 0x008

Reset value: 0x0000\_0028

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

Bits	Field	Descriptions
[5]	LSIRDY	Internal Low Speed Clock Ready Flag 0: Internal 32 kHz RC oscillator clock is not ready 1: Internal 32 kHz RC oscillator clock is ready Set by hardware to indicate that the LSI is stable to be used.
[4]	LSERDY	External Low Speed Clock Ready Flag 0: External 32,768 Hz crystal oscillator clock is not ready 1: External 32,768 Hz crystal oscillator clock is ready Set by hardware to indicate that the LSE is stable to be used.

Bits	Field	Descriptions
[3]	HSIRDY	Internal High Speed Clock Ready Flag 0: Internal 8 MHz RC oscillator clock is not ready 1: Internal 8 MHz RC oscillator clock is ready Set by hardware to indicate whether the HSI is stable or not.
[2]	HSERDY	External High Speed Clock Ready Flag 0: External 4 ~ 16 MHz crystal oscillator clock is not ready 1: External 4 ~ 16 MHz crystal oscillator clock is ready Set by hardware to indicate that the HSE is stable to be used.
[1]	PLLRDY	PLL Clock Ready Flag 0: PLL is not ready 1: PLL is ready Set by hardware to indicate that the PLL is stable to be used.

### Global Clock Interrupt Register – GCIR

This register specifies interrupt enable and flag bits.

Offset: 0x00C

Reset value: 0x0000\_0000

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

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

## PLL Configuration Register – PLLCFGR

This register specifies PLL configuration.

Offset: 0x018

Reset value: 0x0000\_0000

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

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

## PLL Control Register – PLLCR

This register specifies Bypass mode control of PLL.

Offset: 0x01C

Reset value: 0x0000\_0000

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

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

## AHB Configuration Register – AHBCFGR

This register specifies the system clock frequency.

Offset: 0x020

Reset value: 0x0000\_0000

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

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

## AHB Clock Control Register – AHBCCR

This register specifies clock enable bits of AHB peripherals.

Offset: 0x024

Reset value: 0x0000\_0065

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

Bits	Field	Descriptions
[24]	DIVEN	Divider Clock Enable 0: Divider clock is disabled 1: Divider clock is enabled Set and reset by software.
[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. This bit is only available for the HT32F54243/HT32F54253 devices.
[17]	PBEN	GPIO Port B Clock Enable 0: Port B clock is disabled 1: Port B clock is enabled Set and reset by software.
[16]	PAEN	GPIO Port A Clock Enable 0: Port A clock is disabled 1: Port A clock is enabled Set and reset by software.
[13]	CRCEN	CRC Module Clock Enable 0: CRC clock is disabled 1: CRC clock is enabled Set and reset by software.
[11]	CKREFEN	CK_REF Clock Enable 0: CK_REF clock is disabled 1: CK_REF clock is enabled Set and reset by software.



Bits	Field	Descriptions
[6]	APBEN	APB bridge Clock Enable 0: APB bridge clock is automatically disabled by hardware during Sleep mode 1: APB bridge clock is always enabled during Sleep mode Set and reset by software. Users can set APBEN as 0 to reduce power consumption if the APB bridge is unused during Sleep mode.
[5]	BMEN	Bus Matrix Clock Enable 0: Bus Matrix clock is automatically disabled by hardware during Sleep mode 1: Bus Matrix clock is always enabled during Sleep mode Set and reset by software. Users can set BMEN as 0 to reduce power consumption if the bus matrix is unused during Sleep mode.
[4]	PDMAEN	Peripheral DMA Clock Enable 0: PDMA clock is disabled 1: PDMA clock is enabled Set and reset by software. Note: The PDMA can independently operate when the processor enters the sleep mode. But the relative clock of AHB bus slave or peripherals has to be enabled. This bit is only available for the HT32F54243/HT32F54253 devices.
[2]	SRAMEN	SRAM Clock Enable 0: SRAM clock is automatically disabled by hardware during Sleep mode 1: SRAM clock is always enabled during Sleep mode Set and reset by software. Users can set SRAMEN as 0 to reduce power consumption if the SRAM is unused during Sleep mode.
[0]	FMCEN	Flash Memory Controller Clock Enable 0: FMC clock is automatically disabled by hardware during Sleep mode 1: FMC clock is always enabled during Sleep mode Set and reset by software. Users can set FMCEN as 0 to reduce power consumption if the Flash Memory is unused during Sleep mode.

## APB 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
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved					ADCDIV		
Type/Reset						RW	0 RW	0 RW 1
	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
[18:16]	ADCDIV	ADC Clock Frequency Division Selection 000: CK_ADC = CK_AHB 001: CK_ADC = CK_AHB / 2 010: CK_ADC = CK_AHB / 4 011: CK_ADC = CK_AHB / 8 100: CK_ADC = CK_AHB / 16 101: CK_ADC = CK_AHB / 32 110: CK_ADC = CK_AHB / 64 111: CK_ADC = CK_AHB / 3 Set and reset by software to control the ADC conversion clock division factor.

## APB Clock Control Register 0 – APBCCR0

This register specifies clock enable bits of APB peripherals.

Offset: 0x02C

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
	Reserved		LEDCEN	Reserved				
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	UR3EN	UR2EN	UR1EN	UR0EN	USR1EN	USR0EN
Type/Reset	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
	Reserved		SPI1EN	SPI0EN	Reserved	I2C2EN	I2C1EN	I2C0EN
Type/Reset			RW	0	RW	0	RW	0

Bits	Field	Descriptions
[29]	LEDCEN	LED Controller Clock Enable 0: LEDC clock is disabled 1: LEDC 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.
[13]	UR3EN	UART3 Clock Enable 0: UART3 clock is disabled 1: UART3 clock is enabled Set and reset by software. This bit is only available for the HT32F54243/HT32F54253 devices.
[12]	UR2EN	UART2 Clock Enable 0: UART2 clock is disabled 1: UART2 clock is enabled Set and reset by software. This bit is only available for the HT32F54243/HT32F54253 devices.
[11]	UR1EN	UART1 Clock Enable 0: UART1 clock is disabled 1: UART1 clock is enabled Set and reset by software.
[10]	UR0EN	UART0 Clock Enable 0: UART0 clock is disabled 1: UART0 clock is enabled Set and reset by software.

Bits	Field	Descriptions
[9]	USR1EN	<p>USART1 Clock Enable</p> <p>0: USART1 clock is disabled 1: USART1 clock is enabled</p> <p>Set and reset by software.</p> <p>This bit is only available for the HT32F54243/HT32F54253 devices.</p>
[8]	USR0EN	<p>USART0 Clock Enable</p> <p>0: USART0 clock is disabled 1: USART0 clock is enabled</p> <p>Set and reset by software.</p> <p>Since there is only one USART in the HT32F54231/HT32F54241 devices, the pins, registers and control bits related to the USART do not have the serial number "0".</p>
[5]	SPI1EN	<p>SPI1 Clock Enable</p> <p>0: SPI1 clock is disabled 1: SPI1 clock is enabled</p> <p>Set and reset by software.</p>
[4]	SPI0EN	<p>SPI0 Clock Enable</p> <p>0: SPI0 clock is disabled 1: SPI0 clock is enabled</p> <p>Set and reset by software.</p>
[2]	I2C2EN	<p>I<sup>2</sup>C2 Clock Enable</p> <p>0: I<sup>2</sup>C2 clock is disabled 1: I<sup>2</sup>C2 clock is enabled</p> <p>Set and reset by software. This bit is only available for the HT32F54243/HT32F54253 devices.</p>
[1]	I2C1EN	<p>I<sup>2</sup>C1 Clock Enable</p> <p>0: I<sup>2</sup>C1 clock is disabled 1: I<sup>2</sup>C1 clock is enabled</p> <p>Set and reset by software.</p>
[0]	I2C0EN	<p>I<sup>2</sup>C0 Clock Enable</p> <p>0: I<sup>2</sup>C0 clock is disabled 1: I<sup>2</sup>C0 clock is enabled</p> <p>Set and reset by software.</p>

## APB Clock Control Register 1 – APBCCR1

This register specifies clock enable bits of APB peripherals.

Offset: 0x030

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
	SCTM3EN	SCTM2EN	SCTM1EN	SCTM0EN	Reserved			ADCCEN
Type/Reset	RW 0	RW 0	RW 0	RW 0				RW 0
	23	22	21	20	19	18	17	16
	Reserved	COMPEN	Reserved			TOUCHKEYEN	BFTM1EN	BFTM0EN
Type/Reset		RW 0				RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	Reserved							GPTMEN
Type/Reset								RW 0
	7	6	5	4	3	2	1	0
	Reserved	VDDREN	Reserved	WDTREN	Reserved			MCTMEN
Type/Reset		RW 0		RW 0				RW 0

Bits	Field	Descriptions
[31]	SCTM3EN	SCTM3 Clock Enable 0: SCTM3 clock is disabled 1: SCTM3 clock is enabled Set and reset by software. This bit is only available for the HT32F54243/HT32F54253 devices.
[30]	SCTM2EN	SCTM2 Clock Enable 0: SCTM2 clock is disabled 1: SCTM2 clock is enabled Set and reset by software. This bit is only available for the HT32F54243/HT32F54253 devices.
[29]	SCTM1EN	SCTM1 Clock Enable 0: SCTM1 clock is disabled 1: SCTM1 clock is enabled Set and reset by software.
[28]	SCTM0EN	SCTM0 Clock Enable 0: SCTM0 clock is disabled 1: SCTM0 clock is enabled Set and reset by software.
[24]	ADCCEN	ADC Controller Clock Enable 0: ADC clock is disabled 1: ADC clock is enabled Set and reset by software.
[22]	COMPEN	CMP Controller Clock Enable 0: CMP clock is disabled 1: CMP clock is enabled Set and reset by software. This bit is only available for the HT32F54243/HT32F54253 devices.

Bits	Field	Descriptions
[18]	TOUCHKEYEN	Touch Key Clock Enable 0: Touch Key clock is disabled 1: Touch Key clock is enabled Set and reset by software.
[17]	BFTM1EN	BFTM1 Clock Enable 0: BFTM1 clock is disabled 1: BFTM1 clock is enabled Set and reset by software.
[16]	BFTM0EN	BFTM0 Clock Enable 0: BFTM0 clock is disabled 1: BFTM0 clock is enabled Set and reset by software.
[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: Register access clock is disabled 1: Register access clock is enabled Set and reset by software.
[0]	MCTMEN	MCTM Clock Enable 0: MCTM clock is disabled 1: MCTM clock is enabled Set and reset by software.

## Clock Source Status Register – CKST

This register specifies status of various clock sources.

Offset: 0x034

Reset value: 0x0100\_0003

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

Bits	Field	Descriptions
[26:24]	HSIST	Internal High Speed Clock Occupation Status (CK_HSI) xx1: HSI is used by System Clock (CK_SYS) (SW = 0x3) x1x: HSI is used by System PLL 1xx: HSI is used by Clock Monitor
[17:16]	HSEST	External High Speed Clock Occupation Status (CK_HSE) x1: HSE is used by System Clock (CK_SYS) (SW = 0x2) 1x: HSE is used by System PLL
[11:8]	PLLST	PLL Clock Occupation Status xxx1: PLL is used by System Clock (CK_SYS) xx1x: PLL is used by USART 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 being used as system clock currently.

## APB Peripheral Clock Selection Register 0 – APBPCSR0

This register specifies APB peripheral clock prescaler selection.

Offset: 0x038

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
	UR1PCLK		UR0PCLK		USR1PCLK		USR0PCLK	
Type/Reset	RW	0	RW	0	RW	0	RW	0
	23	22	21	20	19	18	17	16
	Reserved		GPTMPCLK		Reserved		MCTMPCLK	
Type/Reset			RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8
	BFTM1PCLK		BFTM0PCLK		UR3PCLK		UR2PCLK	
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
[27:26]	USR1PCLK	USART1 Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[25:24]	USR0PCLK	These bits are only available for the HT32F54243/HT32F54253 devices. USART0 Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock Since there is only one USART in the HT32F54231/HT32F54241 devices, the pins, registers and control bits related to the USART do not have the serial number "0".



Bits	Field	Descriptions
[21:20]	GPTMPCLK	GPTM Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[17:16]	MCTMPCLK	MCTM Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[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
[11:10]	UR3PCLK	UART3 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 These bits are only available for the HT32F54243/HT32F54253 devices.
[9:8]	UR2PCLK	UART2 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 These bits are only available for the HT32F54243/HT32F54253 devices.
[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

Bits	Field	Descriptions
[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	
	SCTM3PCLK				SCTM2PCLK				SCTM1PCLK				SCTM0PCLK			
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0
	23		22		21		20		19		18		17		16	
	I2C2PCLK				Reserved											
Type/Reset	RW	0	RW	0												
	15		14		13		12		11		10		9		8	
	VDDRPCLK				WDTRPCLK				Reserved				CMPPPCLK			
Type/Reset	RW	0	RW	0	RW	0	RW	0					RW	0	RW	0
	7		6		5		4		3		2		1		0	
	Reserved				ADCPCLK				EXTIPCLK				AFIOPCLK			
Type/Reset					RW	0	RW	0	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[31:30]	SCTM3PCLK	SCTM3 Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock These bits are only available for the HT32F54243/HT32F54253 devices.
[29:28]	SCTM2PCLK	SCTM2 Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock These bits are only available for the HT32F54243/HT32F54253 devices.

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
[23:22]	I2C2PCLK	I <sup>2</sup> C2 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 These bits are only available for the HT32F54243/HT32F54253 devices.
[15:14]	VDDRPCLK	V <sub>DD</sub> Domain Register Access Clock Selection 00: PCLK = CK_AHB / 4 01: PCLK = CK_AHB / 8 10: PCLK = CK_AHB / 16 11: PCLK = CK_AHB / 32 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[13:12]	WDTRPCLK	WDT Register Access Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[9:8]	CMPPCLK	CMP Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock These bits are only available for the HT32F54243/HT32F54253 devices.
[5:4]	ADCPCLK	ADC 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

Bits	Field	Descriptions
[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 RO	X RO	X	
	23	22	21	20	19	18	17	16			
	HSIFINE										
Type/Reset	RW	X RW	X RW	X RW	X RW	X RW	X RW	X RW	X 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 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 automatically adjusted by the Auto Trimming Controller (ATC) together with an external reference clock.
[7]	FLOCK	Frequency Lock 0: HSI frequency is not trimmed into target range 1: HSI frequency is trimmed into target range
[6:5]	REFCLKSEL	Reference Clock Selection 00: Select 32.768 kHz external low speed clock source (LSE) 01: Reserved 1x: Select external pin (CKIN) 1 kHz pulse These bits are used to select the reference clock for the HSI Auto Trimming Controller.

Bits	Field	Descriptions
[4]	TMSEL	Trimming Mode Selection 0: Automatic by Auto Trimming Controller 1: Manual by user program This bit is used to select the HSI RC oscillator trimming function by ATC hardware or user programming via the HSIFINE field in this register.
[3]	ATMSEL	Automatic Trimming Mode Selection 0: Auto Trimming Controller is used binary search to approach the target range 1: Auto Trimming Controller is used linear search to approach the target range This bit is used to select the automatic trimming method by ATC hardware for the HSI RC oscillator.
[2]	LTRSEL	Lock Target Range Selection 0: 0.1 % variation 1: 0.2 % variation This bit is used to select the lock target range of the internal HSI RC oscillator trimming function for 0.1 % or 0.2 % variation.
[1]	ATCEN	ATC Enable 0: Disable Auto Trimming Controller 1: Enable Auto Trimming Controller
[0]	TRIMEN	Trimming Enable 0: HSI Trimming is disabled 1: HSI Trimming is enabled Setting this bit high enables the HSI RC oscillator trimming function by ATC hardware or user programming.

### HSI Auto Trimming Counter Register – HSIATCR

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

Offset: 0x044

Reset value: 0x0000\_0000

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

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

## APB Peripheral Clock Selection Register 2 – APBPCSR2

This register specifies the APB peripheral clock prescaler selection.

Offset: 0x048

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[11:10]	TOUCHKEYPCLK	Touch Key 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]	LEDCPCLK	LED 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

## MCU Debug Control Register – MCUDBGCR

This register specifies the debug control of MCU.

Offset: 0x304

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
	Reserved			DBI2C2	DBUR3	DBUR2	DBSCTM3	DBSCTM2
Type/Reset				RW 0	RW 0	RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	DBSCTM1	DBSCTM0	Reserved		DBUR1	DBUR0	DBBFTM1	DBBFTM0
Type/Reset	RW 0	RW 0			RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	Reserved	DBDSLP2	DBI2C1	DBI2C0	DBSPI1	DBSPI0	DBUSR1	DBUSR0
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	DBMCTM	DBWDT	Reserved	DBDSLP1	DBSLP
Type/Reset		RW 0		RW 0	RW 0		RW 0	RW 0

Bits	Field	Descriptions
[28]	DBI2C2	I <sup>2</sup> C2 Debug Mode Enable 0: Same behavior as in normal mode 1: I <sup>2</sup> C2 timeout is frozen when the core is halted Set and reset by software. This bit is only available for the HT32F54243/HT32F54253 devices.
[27]	DBUR3	UART3 Debug Mode Enable 0: Same behavior as in normal mode 1: UART3 FIFO timeout is frozen when the core is halted Set and reset by software.
[26]	DBUR2	UART2 Debug Mode Enable 0: Same behavior as in normal mode 1: UART2 FIFO timeout is frozen when the core is halted Set and reset by software.
[25]	DBSCTM3	SCTM3 Debug Mode Enable 0: SCTM3 counter continues even if the core is halted 1: SCTM3 counter is stopped when the core is halted Set and reset by software. This bit is only available for the HT32F54243/HT32F54253 devices.
[24]	DBSCTM2	SCTM2 Debug Mode Enable 0: SCTM2 counter continues even if the core is halted 1: SCTM2 counter is stopped when the core is halted Set and reset by software. This bit is only available for the HT32F54243/HT32F54253 devices.
[23]	DBSCTM1	SCTM1 Debug Mode Enable 0: SCTM1 counter continues even if the core is halted 1: SCTM1 counter is stopped when the core is halted Set and reset by software.

Bits	Field	Descriptions
[22]	DBSCTM0	SCTM0 Debug Mode Enable 0: SCTM0 counter continues even if the core is halted 1: SCTM0 counter is stopped when the core is halted 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.
[16]	DBBFTM0	BFTM0 Debug Mode Enable 0: BFTM0 counter continues to count even if the core is halted 1: BFTM0 counter is stopped when the core is halted Set and reset by software.
[14]	DBDSLP2	Debug Deep-Sleep2 Mode 0: LDO = ULDO On, FCLK = Off and CM0PEN = 0 in Deep-Sleep2 mode 1: LDO = MLDO On, FCLK = On and CM0PEN = 1 in Deep-Sleep2 mode 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.
[9]	DBUSR1	USART1 Debug Mode Enable 0: Same behavior as in normal mode 1: USART1 FIFO timerout is frozen when the core is halted Set and reset by software. This bit is only available for the HT32F54243/HT32F54253 devices.
[8]	DBUSR0	USART0 Debug Mode Enable 0: Same behavior as in normal mode 1: USART0 FIFO timeout is frozen when the core is halted Set and reset by software. Since there is only one USART in the HT32F54231/HT32F54241 devices, the pins, registers and control bits related to the USART do not have the serial number "0".



Bits	Field	Descriptions
[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.
[4]	DBMCTM	MCTM Debug Mode Enable 0: MCTM counter continue to count even if the core is halted 1: MCTM 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.
[1]	DBDSLP1	Debug Deep-Sleep1 Mode 0: LDO = ULDO On, FCLK = Off and CM0PEN = 0 in Deep-Sleep1 mode 1: LDO = MLDO On, FCLK = On and CM0PEN = 1 in Deep-Sleep1 mode Set and reset by software.
[0]	DBSLP	Debug Sleep Mode 0: LDO = MLDO On, FCLK = On and CM0PEN = 0 in Sleep mode 1: LDO = MLDO On, FCLK = On and CM0PEN = 1 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.

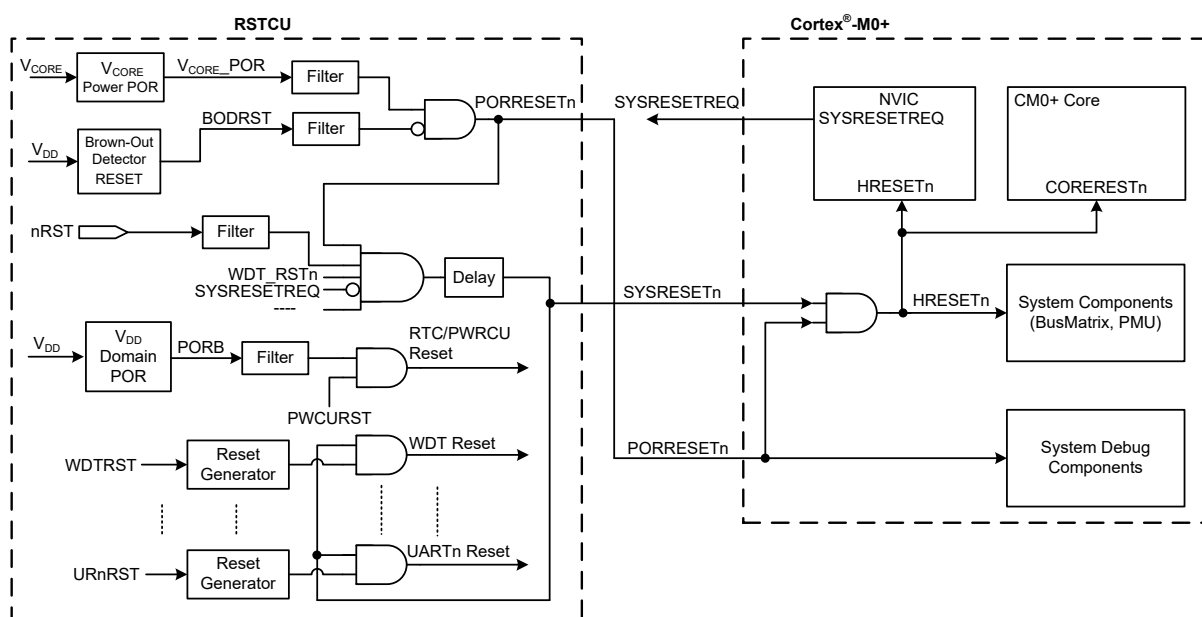
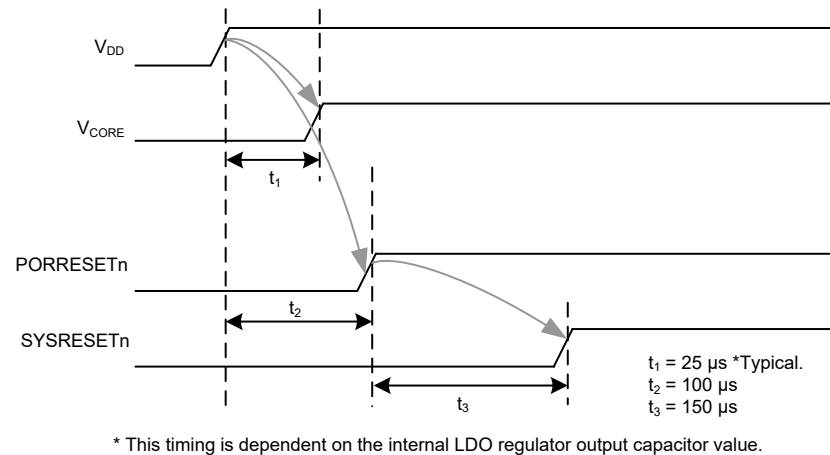


Figure 20. RSTCU Block Diagram

## 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 21, the  $V_{CORE\_POR}$  active low signal will be de-asserted when the internal LDO voltage regulator is ready to provide the  $V_{CORE}$  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 21. Power-On Reset Sequence**

## System Reset

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

## AHB and APB Unit Reset

The AHB and APB unit reset can be divided into hardware and software resets. A hardware reset can be generated by either power on reset or system reset for all AHB and APB units. Each functional IP connected to the AHB and APB buses can be reset individually through the associated software reset bits in the RSTCU. For example, the application software can generate a UART0 reset via the UR0RST bit in the APBPRSTR0 register.

## Register Map

The following table shows the RSTCU registers and reset values.

**Table 20. RSTCU Register Map**

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

## Register Descriptions

### Global Reset Status Register – GRSR

This register specifies a variety of reset status conditions.

Offset: 0x100

Reset value: 0x0000\_0008

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

Bits	Field	Descriptions
[3]	PORSTF	<p><math>V_{CORE}</math> Power On Reset Flag</p> <p>0: No POR occurred</p> <p>1: POR occurred</p> <p>This bit is set by hardware when a power-on reset occurs and reset by writing 1 into it.</p>
[2]	WDTRSTF	<p>Watchdog Timer Reset Flag</p> <p>0: No Watchdog Timer reset occurred</p> <p>1: Watchdog Timer occurred</p> <p>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.</p>
[1]	EXTRSTF	<p>External Pin Reset Flag</p> <p>0: No pin reset occurred</p> <p>1: Pin reset occurred</p> <p>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.</p>
[0]	NVICRSTF	<p>NVIC Reset Flag</p> <p>0: No NVIC asserting system reset occurred</p> <p>1: NVIC asserting system reset occurred</p> <p>This bit is set by hardware when a system reset occurs and reset by writing 1 into it or by hardware when a power-on reset occurs.</p>

## AHB Peripheral Reset Register – AHBPRSTR

This register specifies several AHB peripherals software reset control bits.

Offset: 0x104

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[24]	DIVRST	Divider Reset Control 0: No reset 1: Reset Divider This bit is set by software and cleared to 0 by hardware automatically.
[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. This bit is only available for the HT32F54243/HT32F54253 devices.
[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.
[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. This bit is only available for the HT32F54243/HT32F54253 devices.

## APB Peripheral Reset Register 0 – APBPRSTR0

This register specifies several APB peripherals software reset control bits.

Offset: 0x108

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
	Reserved		LEDCRST	Reserved				
Type/Reset			RW	0				
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	EXTIRST	AFIORST	UR3RST	UR2RST	UR1RST	UR0RST	USR1RST	USR0RST
Type/Reset	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
	Reserved		SPI1RST	SPI0RST	Reserved	I2C2RST	I2C1RST	I2C0RST
Type/Reset			RW	0	RW	0	RW	0

Bits	Field	Descriptions
[29]	LEDCRST	LED Controller Reset Control 0: No reset 1: Reset LED Controller 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.
[13]	UR3RST	UART3 Reset Control 0: No reset 1: Reset UART3 This bit is set by software and cleared to 0 by hardware automatically. This bit is only available for the HT32F54243/HT32F54253 devices.
[12]	UR2RST	UART2 Reset Control 0: No reset 1: Reset UART2 This bit is set by software and cleared to 0 by hardware automatically. This bit is only available for the HT32F54243/HT32F54253 devices.
[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.

Bits	Field	Descriptions
[9]	USR1RST	<p>USART1 Reset Control</p> <p>0: No reset 1: Reset USART1</p> <p>This bit is set by software and cleared to 0 by hardware automatically. This bit is only available for the HT32F54243/HT32F54253 devices.</p>
[8]	USR0RST	<p>USART0 Reset Control</p> <p>0: No reset 1: Reset USART0</p> <p>This bit is set by software and cleared to 0 by hardware automatically. Since there is only one USART in the HT32F54231/HT32F54241 devices, the pins, registers and control bits related to the USART do not have the serial number "0".</p>
[5]	SPI1RST	<p>SPI1 Reset Control</p> <p>0: No reset 1: Reset SPI1</p> <p>This bit is set by software and cleared to 0 by hardware automatically.</p>
[4]	SPI0RST	<p>SPI0 Reset Control</p> <p>0: No reset 1: Reset SPI0</p> <p>This bit is set by software and cleared to 0 by hardware automatically.</p>
[2]	I2C2RST	<p>I<sup>2</sup>C2 Reset Control</p> <p>0: No reset 1: Reset I<sup>2</sup>C2</p> <p>This bit is set by software and cleared to 0 by hardware automatically. This bit is only available for the HT32F54243/HT32F54253 devices.</p>
[1]	I2C1RST	<p>I<sup>2</sup>C1 Reset Control</p> <p>0: No reset 1: Reset I<sup>2</sup>C1</p> <p>This bit is set by software and cleared to 0 by hardware automatically.</p>
[0]	I2C0RST	<p>I<sup>2</sup>C0 Reset Control</p> <p>0: No reset 1: Reset I<sup>2</sup>C0</p> <p>This bit is set by software and cleared to 0 by hardware automatically.</p>

## APB Peripheral Reset Register 1 – APBPRSTR1

This register specifies several APB peripherals software reset control bits.

Offset: 0x10C

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
	SCTM3RST	SCTM2RST	SCTM1RST	SCTM0RST	Reserved			ADCRST
Type/Reset	RW 0	RW 0	RW 0	RW 0				RW 0
	23	22	21	20	19	18	17	16
	Reserved	CMPRST	Reserved			TOUCHKEYRST	BFTM1RST	BFTM0RST
Type/Reset		RW 0				RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	Reserved							GPTMRST
Type/Reset								RW 0
	7	6	5	4	3	2	1	0
	Reserved			WDTRST	Reserved		MCTMRST	
Type/Reset				RW 0			RW 0	

Bits	Field	Descriptions
[31]	SCTM3RST	SCTM3 Reset Control 0: No reset 1: Reset SCTM3 This bit is set by software and cleared to 0 by hardware automatically. This bit is only available for the HT32F54243/HT32F54253 devices.
[30]	SCTM2RST	SCTM2 Reset Control 0: No reset 1: Reset SCTM2 This bit is set by software and cleared to 0 by hardware automatically. This bit is only available for the HT32F54243/HT32F54253 devices.
[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	ADC Reset Control 0: No reset 1: Reset A/D Converter This bit is set by software and cleared to 0 by hardware automatically.
[22]	CMPRST	CMP Reset Control 0: No reset 1: Reset CMP This bit is set by software and cleared to 0 by hardware automatically. This bit is only available for the HT32F54243/HT32F54253 devices.
[18]	TOUCHKEYRST	Touch Key Reset Control 0: No reset 1: Reset Touch Key



Bits	Field	Descriptions
[17]	BFTM1RST	BFTM1 Reset Control 0: No reset 1: Reset BFTM1 This bit is set by software and cleared to 0 by hardware automatically.
[16]	BFTM0RST	BFTM0 Reset Control 0: No reset 1: Reset BFTM0 This bit is set by software and cleared to 0 by hardware automatically.
[8]	GPTMRST	GPTM Reset Control 0: No reset 1: Reset GPTM This bit is set by software and cleared to 0 by hardware automatically.
[4]	WDTRST	Watchdog Timer Reset Control 0: No reset 1: Reset Watchdog Timer This bit is set by software and cleared to 0 by hardware automatically.
[0]	MCTMRST	MCTM Reset Control 0: No reset 1: Reset MCTM This bit is set by software and cleared to 0 by hardware automatically.

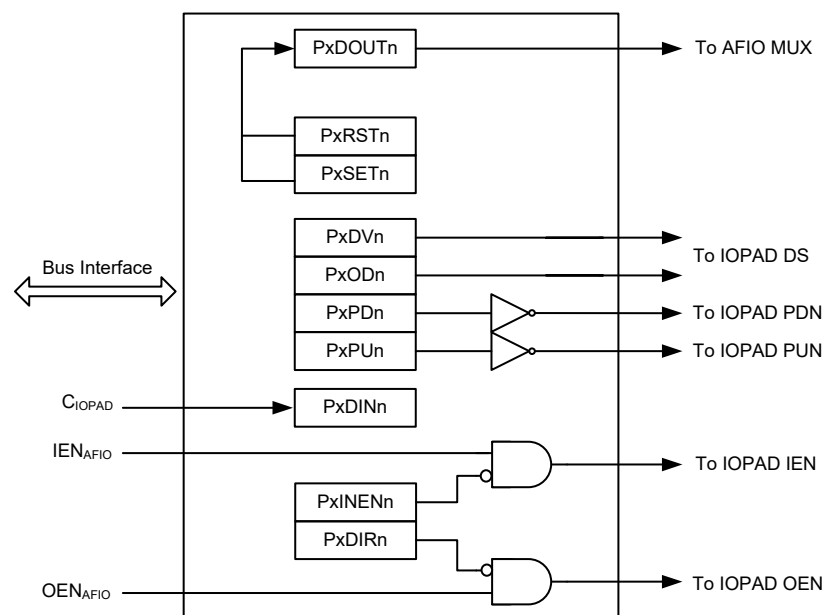
## 8 General Purpose I/O (GPIO)

### Introduction

There are up to 40 General Purpose I/O ports, GPIO, named PA0 ~ PA15, PB0 ~ PB15 and PC0 ~ PC7 for the HT32F54231/HT32F54241 devices and up to 54 General Purpose I/O ports, GPIO, named PA0 ~ PA15, PB0 ~ PB15, PC0 ~ PC15 and PD0 ~ PD5 for the HT32F54243/HT32F54253 devices to implement the logic input/output functions. Each of the GPIO ports has related control and configuration registers to satisfy the requirement of specific applications. The actual available General Purpose I/O port numbers are dependent on the device specification and package type. Refer to the device datasheet for detailed information.

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

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



**Figure 22. GPIO Block Diagram**

## Features

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

## Functional Descriptions

### Default GPIO Pin Configuration

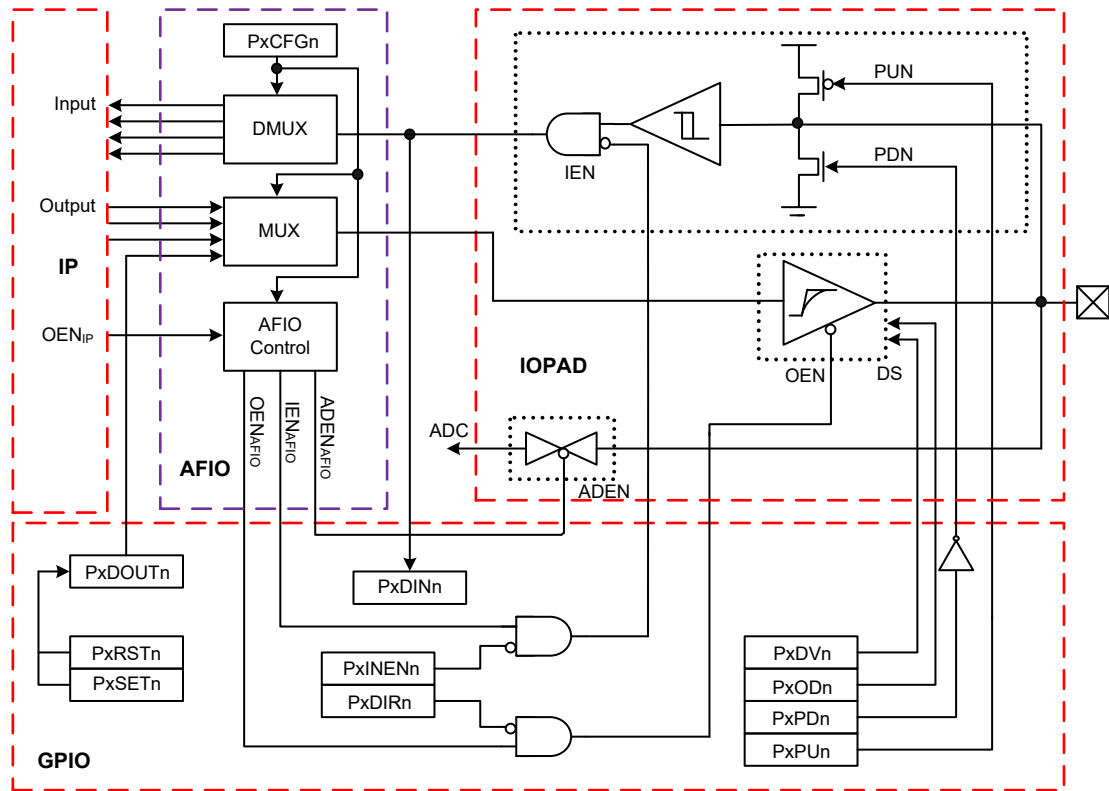
During or just after the reset period, the alternative functions are all inactive and the GPIO ports are configured into the input disable floating mode, i.e. input disabled without pull-up/pull-down resistors. Only the boot and Serial-Wired Debug pins which are pin-shared with the I/O pins are active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

- PA9\_BOOT: Input enable with internal pull-up
- SWCLK: Input enable with internal pull-up
- SWDIO: Input enable with internal pull-up

### General Purpose I/O – GPIO

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

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



**Figure 23. AFIO/GPIO Control Signal**

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

**Table 21. 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) registers are used to lock the port x and lock control options. The value 0x5FA0 is written into the PxLKEY field in the PxLOCKR registers to freeze the PxDIRCR, PxINER, PxPUR, PxPDR, PxODR, PxDRVR control and AFIO mode configuration (GPxCFGHR or GPxCFGLR, where x = A ~ D). If the value in the PxLOCKR register is 0x5FA0\_0001, it means that the Port x Lock function is enabled and the Port x pin 0 is frozen.

## Register Map

The following table shows the GPIO registers and reset values of the Port A ~ D. The port D related registers are only available for the HT32F54243/HT32F54253 devices.

**Table 22. GPIO Register Map**

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

Register	Offset	Description	Reset Value
PCDINR	0x01C	Port C Data Input Register	0x0000_0000
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
PDDRVCR	0x014	Port D Drive Current Selection Register	0x0000_0000
PDLOCKR	0x018	Port D Lock Register	0x0000_0000
PDDINR	0x01C	Port D Data Input Register	0x0000_0000
PDDOUTR	0x020	Port D Data Output Register	0x0000_0000
PDSRR	0x024	Port D Output Set and Reset Control Register	0x0000_0000
PDRR	0x028	Port D Output Reset Control Register	0x0000_0000

## Register Descriptions

### Port A Data Direction Control Register – PADIRCR

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

Offset: 0x000

Reset value: 0x0000\_0000

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

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

## Port A Input Function Enable Control Register – PAINER

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

Offset: 0x004

Reset value: 0x0000\_0200

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	PAINEN								
	RW	0	RW	0	RW	0	RW	0	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	<p>GPIO Port A pin n Input Enable Control Bits (n = 0 ~ 15)</p> <p>0: Pin n input function is disabled</p> <p>1: Pin n input function is enabled</p> <p>When the pin n input function is disabled, the input Schmitt trigger will be turned off and the Schmitt trigger output will remain at a zero state.</p>



## Port A Pull-Up Selection Register – PAPUR

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

Offset: 0x008

Reset value: 0x0000\_3200

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

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

## Port A Pull-Down Selection Register – PAPDR

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

Offset: 0x00C

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[15:0]	PAPDn	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 – PADDRVR

This register specifies the GPIO Port A output driving current.

Offset: 0x014

Reset value: 0x0000\_0000

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

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

## Port A Lock Register – PALOCKR

This register specifies the GPIO Port A lock configuration.

Offset: 0x018

Reset value: 0x0000\_0000

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

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

## Port A Data Input Register – PADINR

This register specifies the GPIO Port A input data.

Offset: 0x01C

Reset value: 0x0000\_3200

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

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

## Port A Output Data Register – PADOUTR

This register specifies the GPIO Port A output data.

Offset: 0x020

Reset value: 0x0000\_0000

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

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

## Port B Data Direction Control Register – PBDIRCR

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

Offset: 0x000

Reset value: 0x0000\_0000

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

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



## Port B Input Function Enable Control Register – PBINER

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

Offset: 0x004

Reset value: 0x0000\_0000

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

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

## Port B Pull-Up Selection Register – PBPUR

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

Offset: 0x008

Reset value: 0x0000\_0000

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

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 – PBDRVR

This register specifies the GPIO Port B output driving current.

Offset: 0x014

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	PBDV15		PBDV14		PBDV13		PBDV12		
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW
	23	22	21	20	19	18	17	16	
	PBDV11		PBDV10		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 Drive Current Selection Control Bits (n = 0 ~ 15) 00: 4 mA source/sink current 01: 8 mA source/sink current 10: 12 mA source/sink current 11: 16 mA source/sink current

## Port B Lock Register – PBLCKR

This register specifies the GPIO Port B lock configuration.

Offset: 0x018

Reset value: 0x0000\_0000

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

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

## Port B Data Input Register – PBDINR

This register specifies the GPIO Port B input data.

Offset: 0x01C

Reset value: 0x0000\_0000

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

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	RW	0	RW	0	RW	0

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

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

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

Offset: 0x024

Reset value: 0x0000\_0000

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

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 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							
	WO	0	WO	0	WO	0	WO	0
	WO	0	WO	0	WO	0	WO	0
	7	6	5	4	3	2	1	0
Type/Reset	PBRST							
	WO	0	WO	0	WO	0	WO	0
	WO	0	WO	0	WO	0	WO	0

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

## Port C Data Direction Control Register – PCDIRCR

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

Offset: 0x000

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[x:0]	PCDIRn	GPIO Port C pin n Direction Control Bits (n = 0 ~ x) 0: Pin n is in input mode 1: Pin n is in output mode For the HT32F54231/HT32F54241 devices the variable “x” is equal to 7 while the variable “x” is equal to 15 for the HT32F54243/HT32F54253 devices.

## 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
[x:0]	PCINEN <sub>n</sub>	<p>GPIO Port C pin n Input Enable Control Bits (n = 0 ~ x)</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> <p>For the HT32F54231/HT32F54241 devices the variable “x” is equal to 7 while the variable “x” is equal to 15 for the HT32F54243/HT32F54253 devices.</p>

## Port C Pull-Up Selection Register – PCPUR

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

Offset: 0x008

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	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
[x:0]	PCPU <sub>n</sub>	<p>GPIO Port C pin n Pull-Up Selection Control Bits (n = 0 ~ x)</p> <p>0: Pin n pull-up function is disabled</p> <p>1: Pin n pull-up function is enabled</p> <p>For the HT32F54231/HT32F54241 devices the variable “x” is equal to 7 while the variable “x” is equal to 15 for the HT32F54243/HT32F54253 devices.</p> <p>Note: When the pull-up and pull-down functions are both enabled, the pull-up function will have the higher priority and therefore the pull-down function will be blocked and disabled.</p>

## Port C Pull-Down Selection Register – PCPDR

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

Offset: 0x00C

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
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
[x:0]	PCPDn	<p>GPIO Port C pin n Pull-Down Selection Control Bits (n = 0 ~ x)</p> <p>0: Pin n pull-down function is disabled</p> <p>1: Pin n pull-down function is enabled</p> <p>For the HT32F54231/HT32F54241 devices the variable “x” is equal to 7 while the variable “x” is equal to 15 for the HT32F54243/HT32F54253 devices.</p> <p>Note: When the pull-up and pull-down functions are both enabled, the pull-up function will have the higher priority and therefore the pull-down function will be blocked and disabled.</p>

## Port C Open-Drain Selection Register – PCODR

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

Offset: 0x010

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
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
[x:0]	PCODn	<p>GPIO Port C pin n Open-Drain Selection Control Bits (n = 0 ~ x)</p> <p>0: Pin n Open-Drain output is disabled (The output type is CMOS output)</p> <p>1: Pin n Open-Drain output is enabled (The output type is open-drain output)</p> <p>For the HT32F54231/HT32F54241 devices the variable “x” is equal to 7 while the variable “x” is equal to 15 for the HT32F54243/HT32F54253 devices.</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 – PCDVR

This register specifies the GPIO Port C output driving current.

Offset: 0x014

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	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
[2x+1:0]	PCDVn[1:0]	GPIO Port C pin n Drive Current Selection Control Bits (n = 0 ~ x) 00: 4 mA source/sink current 01: 8 mA source/sink current 10: 12 mA source/sink current 11: 16 mA source/sink current For the HT32F54231/HT32F54241 devices the variable “x” is equal to 7 while the variable “x” is equal to 15 for the HT32F54243/HT32F54253 devices

## Port C Lock Register – PCLOCKR

This register specifies the GPIO Port C lock configuration.

Offset: 0x018

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[31:16]	PCLKEY	<p>GPIO Port C lock Key</p> <p>0x5FA0: Port C Lock function is enable Others: Port C Lock function is disable</p> <p>To lock the Port C function, a value of 0x5FA0 should be written into the PCLKEY field in this register. To execute a successful write operation on this lock register, the value written into the PCLKEY field must be 0x5FA0. If the value written into this field is not equal to 0x5FA0, any write operations on the PCLOCKR register will be aborted. The result of a read operation on the PCLKEY field returns the GPIO Port C Lock Status which indicates whether the GPIO Port C is locked or not. If the read value of the PCLKEY field is 0, this indicates that the GPIO Port C Lock function is disabled. Otherwise, it indicates that the GPIO Port C Lock function is enabled as the read value is equal to 1.</p>
[x:0]	PCLOCKn	<p>GPIO Port C pin n Lock Control Bits (n = 0 ~ x)</p> <p>0: Port C pin n is not locked 1: Port C pin n is locked</p> <p>The PCLOCKn bits are used to lock the configurations of corresponding GPIO Pins when the correct Lock Key is applied to the PCLKEY field. The locked configurations including PCDIRn, PCINENn, PCPUn, PCPDn, PCODn and PCDVn setting in the related GPIO registers. Additionally, the GPCCFGHR or GPCCFGLR register which is used to configure the alternative function of the associated GPIO pin will also be locked. Note that the PCLOCKR register can only be written once which means that PCLKEY and PCLOCKn (lock control bit) should be written together and cannot be changed until a system reset or GPIO Port C reset occurs.</p> <p>For the HT32F54231/HT32F54241 devices the variable “x” is equal to 7 while the variable “x” is equal to 15 for the HT32F54243/HT32F54253 devices.</p>



## Port C Data Input Register – PCDINR

This register specifies the GPIO Port C input data.

Offset: 0x01C

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[x:0]	PCDINn	<p>GPIO Port C pin n Data Input Bits (n = 0 ~ x)</p> <p>0: The input data of corresponding pin is 0</p> <p>1: The input data of corresponding pin is 1</p> <p>For the HT32F54231/HT32F54241 devices the variable “x” is equal to 7 while the variable “x” is equal to 15 for the HT32F54243/HT32F54253 devices.</p>

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

Bits	Field	Descriptions
[x:0]	PCDOUTn	GPIO Port C pin n Data Output Bits (n = 0 ~ x) 0: Data to be output on pin n is 0 1: Data to be output on pin n is 1 For the HT32F54231/HT32F54241 devices the variable “x” is equal to 7 while the variable “x” is equal to 15 for the HT32F54243/HT32F54253 devices.

## 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
[x+16:16]	PCRSTn	<p>GPIO Port C pin n Output Reset Control Bits (n = 0 ~ x)</p> <p>0: No effect on the PCDOUn bit</p> <p>1: Reset the PCDOUn bit</p> <p>Note that when the PCRSTn bit in this register or the PCRSTn bit in the PCRR register is enabled, the reset function on the PCDOUn bit will take effect.</p> <p>For the HT32F54231/HT32F54241 devices the variable “x” is equal to 7 while the variable “x” is equal to 15 for the HT32F54243/HT32F54253 devices.</p>
[x:0]	PCSETn	<p>GPIO Port C pin n Output Set Control Bits (n = 0 ~ x)</p> <p>0: No effect on the PCDOUn bit</p> <p>1: Set the PCDOUn bit</p> <p>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.</p> <p>For the HT32F54231/HT32F54241 devices the variable “x” is equal to 7 while the variable “x” is equal to 15 for the HT32F54243/HT32F54253 devices.</p>

## 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								

Bits	Field	Descriptions
[x:0]	PCRSTn	GPIO Port C pin n Output Reset Control Bits (n = 0 ~ x) 0: No effect on the PCDOUn bit 1: Reset the PCDOUn bit For the HT32F54231/HT32F54241 devices the variable “x” is equal to 7 while the variable “x” is equal to 15 for the HT32F54243/HT32F54253 devices.

## 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
					RW	0	RW	0
						RW	0	RW
							RW	0
								RW
								0

Bits	Field	Descriptions
[5:0]	PDDIRn	GPIO Port D pin n Direction Control Bits (n = 0 ~ 5) 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
					RW	0	RW	0
						RW	0	RW
							RW	0
								RW
								0

Bits	Field	Descriptions
[5:0]	PDINENn	GPIO Port D pin n Input Enable Control Bits (n = 0 ~ 5) 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		PDPUR					
			RW	0	RW	0	RW	0
					RW	0	RW	0
						RW	0	RW
							RW	0

Bits	Field	Descriptions
[5:0]	PDPURn	GPIO Port D pin n Pull-Up Selection Control Bits (n = 0 ~ 5) 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
					RW	0	RW	0
						RW	0	RW
							RW	0

Bits	Field	Descriptions
[5:0]	PDPDn	GPIO Port D pin n Pull-Down Selection Control Bits (n = 0 ~ 5) 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
					RW	0	RW	0
						RW	0	RW
							RW	0

Bits	Field	Descriptions
[5:0]	PDODn	GPIO Port D pin n Open-Drain Selection Control Bits (n = 0 ~ 5) 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 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				PDDV5		PDDV4	
	7	6	5	4	3	2	1	0
Type/Reset	PDDV3		PDDV2		PDDV1		PDDV0	
	RW	0	RW	0	RW	0	RW	0

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

## Port D Lock Register – PDLOCKR

This register specifies the GPIO Port D lock configuration.

Offset: 0x018

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[31:16]	PDLKEY	<p>GPIO Port D Lock Key</p> <p>0x5FA0: Port D Lock function is enable Others: Port D Lock function is disable</p> <p>To lock the Port D function, a value 0x5FA0 should be written into the PDLKEY field in this register. To execute a successful write operation on this lock register, the value written into the PDLKEY field must be 0x5FA0. If the value written into this field is not equal to 0x5FA0, any write operations on the PDLOCKR register will be aborted. The result of a read operation on the PDLKEY field returns the GPIO Port D Lock Status which indicates whether the GPIO Port D is locked or not. If the read value of the PDLKEY field is 0, this indicates that the GPIO Port D Lock function is disabled. Otherwise, it indicates that the GPIO Port D Lock function is enabled as the read value is equal to 1.</p>
[5:0]	PDLOCKn	<p>GPIO Port D pin n Lock Control Bits (n = 0 ~ 5)</p> <p>0: Port D pin n is not locked 1: Port D pin n is locked</p> <p>The PDLOCKn bits are used to lock the configurations of corresponding GPIO Pins when the correct Lock Key is applied to the PDLKEY field. The locked configurations including PDDIRn, PDINENn, PDPUn, PDPDn, PDODn and PDDVn setting in the related GPIO registers. Additionally, the GPDCFGHR or GPDCFGLR field which is used to configure the alternative function of the associated GPIO pin will also be locked. Note that the PDLOCKR can only be written once which means that PDLKEY and PDLOCKn (lock control bit) should be written together and cannot 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
Type/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		PDDIN					
			RO	0	RO	0	RO	0
					RO	0	RO	0
						RO	0	RO
							RO	0

Bits	Field	Descriptions
[5:0]	PDDINn	GPIO Port D pin n Data Input Bits (n = 0 ~ 5) 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
Type/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		PDDOUT					
			RW	0	RW	0	RW	0
					RW	0	RW	0
						RW	0	RW
							RW	0

Bits	Field	Descriptions
[5:0]	PDDOUTn	GPIO Port D pin n Data Output Bits (n = 0 ~ 5) 0: Data to be output on pin n is 0 1: Data to be output on pin n is 1

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

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

Offset: 0x024

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[21:16]	PDRSTn	GPIO Port D pin n Output Reset Control Bits (n = 0 ~ 5) 0: No effect on the PDDOUTn bit 1: Reset the PDDOUTn bit Note that when the PDRSTn bit in this register or the PDRSTn bit in the PDRR register is enabled, the reset function on the PDDOUTn bit will take effect.
[5:0]	PDSETn	GPIO Port D pin n Output Set Control Bits (n = 0 ~ 5) 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	WO	0
					WO	0	WO	0
						WO	0	WO
							WO	0
								0

Bits	Field	Descriptions
[5:0]	PDRSTn	GPIO Port D pin n Output Reset Bits (n = 0 ~ 5) 0: No effect on the PDDOUTn bit 1: Reset the PDDOUTn bit

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

### Introduction

In order to expand the flexibility of the GPIO or the usage of peripheral functions, each 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. Please refer to the EXTI section for more details.

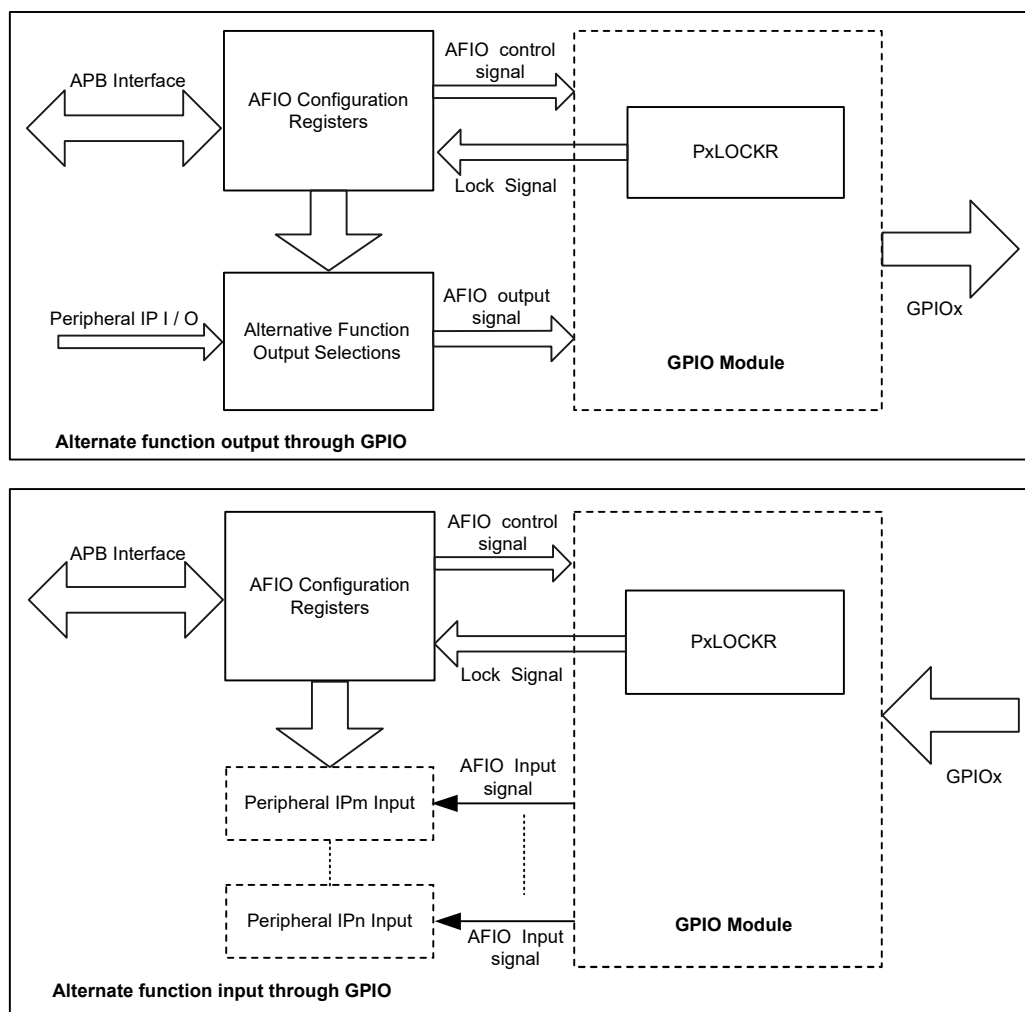


Figure 24. AFIO Block Diagram

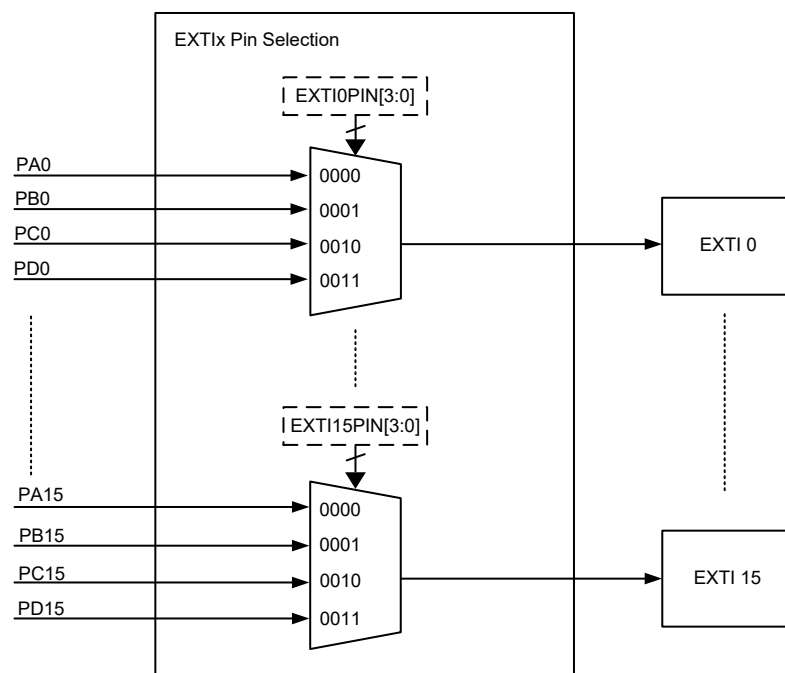
## Features

- 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 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 25. EXTI Channel Input Selection**



## Alternate Function

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

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

**Table 23. AFIO Selection for Peripheral Map Example**

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

Note: CMP is only available for the HT32F54243/HT32F54253 devices.

## Lock Mechanism

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

## Register Map

The following table shows the AFIO registers and reset values.

**Table 24. AFIO Register Map**

Register	Offset	Description	Reset Value
ESSR0	0x000	EXTI Source Selection Register 0	0x0000_0000
ESSR1	0x004	EXTI Source Selection Register 1	0x0000_0000
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

## Register Descriptions

### EXTI Source Selection Register 0 – ESSR0

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

Offset: 0x000

Reset value: 0x0000\_0000

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

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

## EXTI Source Selection Register 1 – ESSR1

This register specifies the 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	0
	23	22	21	20	19	18	17	16	
	EXTI13PIN								
Type/Reset	RW	0	RW	0	RW	0	RW	0	0
	15	14	13	12	11	10	9	8	
	EXTI11PIN								
Type/Reset	RW	0	RW	0	RW	0	RW	0	0
	7	6	5	4	3	2	1	0	
	EXTI9PIN								
Type/Reset	RW	0	RW	0	RW	0	RW	0	0

Bits	Field	Descriptions
[31:0]	EXTInPIN[3:0]	EXTIn Pin Selection (n = 8 ~ 15) 0000: PA Bit n is selected as EXTIn source signal 0001: PB Bit n is selected as EXTIn source signal 0010: PC Bit n is selected as EXTIn source signal 0011: PD Bit n is selected as EXTIn source signal Others: Reserved Note: Since not all 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.

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

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

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

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	PxCFG7								
Type/Reset	RW	0	RW	0	RW	0	RW	0	0
	23	22	21	20	19	18	17	16	
	PxCFG5								
Type/Reset	RW	0	RW	0	RW	0	RW	0	0
	15	14	13	12	11	10	9	8	
	PxCFG3								
Type/Reset	RW	0	RW	0	RW	0	RW	0	0
	7	6	5	4	3	2	1	0	
	PxCFG1								
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 = 0 ~ 7)</p> <p>0000: Port x pin n is selected as AF0</p> <p>0001: Port x pin n is selected as AF1</p> <p>.</p> <p>1110: Port x pin n is selected as AF14</p> <p>1111: Port x pin n is selected as AF15</p> <p>If the pin is selected as unavailable item which is noted as "N/A" in the "Alternate Function Mapping" table of the device datasheet, this pin will be defined as default alternate function. Refer to the "Alternate Function Mapping" table in the device datasheet for detailed mapping of the alternate function I/O pins.</p>

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

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

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

Reset value: 0x0000\_0000

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

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

# 10 Nested Vectored Interrupt Controller (NVIC)

## Introduction

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

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

**Table 25. Exception Types**

Interrupt Number	Exception Number	Exception Type	Priority	Vector Address	Description
—	0	—	—	0x000	Initial Stack Point value
—	1	Reset	-3 (Highest)	0x004	Reset
-14	2	NMI	-2	0x008	Non-Maskable Interrupt. The clock stuck interrupt signal (clock monitor function provided by Clock Control Unit) is connected to the NMI input
-13	3	Hard Fault	-1	0x00C	All fault classes
—	4-10	Reserved	—	—	—
-5	11	SVCALL	Configurable <sup>(1)</sup>	0x02C	SVC instruction System service call
—	12-13	Reserved	—	—	—
-2	14	PendSV	Configurable <sup>(1)</sup>	0x038	System Service Pendable request
-1	15	SysTick	Configurable <sup>(1)</sup>	0x03C	SysTick timer decreased to zero
0	16	LVD	Configurable <sup>(2)</sup>	0x040	Low voltage detection interrupt
1	17	RTC	Configurable <sup>(2)</sup>	0x044	RTC global interrupt
2	18	FMC	Configurable <sup>(2)</sup>	0x048	FMC global interrupt
3	19	WKUP	Configurable <sup>(2)</sup>	0x04C	EXTI event wakeup or external WAKEUPn pin interrupt <sup>(3)</sup>
4	20	EXTI0 ~ 1	Configurable <sup>(2)</sup>	0x050	EXTI Line 0 & 1 interrupt
5	21	EXTI2 ~ 3	Configurable <sup>(2)</sup>	0x054	EXTI Line 2 & 3 interrupt
6	22	EXTI4 ~ 15	Configurable <sup>(2)</sup>	0x058	EXTI Line 4 ~ 15 interrupt
7	23	CMP0 ~ 1 <sup>(4)</sup>	Configurable <sup>(2)</sup>	0x05C	Comparators global interrupt
8	24	ADC	Configurable <sup>(2)</sup>	0x060	ADC global interrupt
9	25	I <sup>2</sup> C2 <sup>(4)</sup>	Configurable <sup>(2)</sup>	0x064	I <sup>2</sup> C2 global interrupt
10	26	MCTM	Configurable <sup>(2)</sup>	0x068	MCTM global interrupt

Interrupt Number	Exception Number	Exception Type	Priority	Vector Address	Description
11	27	TKEY	Configurable <sup>(2)</sup>	0x06C	Touch Key global interrupt
12	28	GPTM	Configurable <sup>(2)</sup>	0x070	GPTM global interrupt
13	29	SCTM0	Configurable <sup>(2)</sup>	0x074	SCTM0 global interrupt
14	30	SCTM1	Configurable <sup>(2)</sup>	0x078	SCTM1 global interrupt
15	31	SCTM2 <sup>(4)</sup>	Configurable <sup>(2)</sup>	0x07C	SCTM2 global interrupt
16	32	SCTM3 <sup>(4)</sup>	Configurable <sup>(2)</sup>	0x080	SCTM3 global interrupt
17	33	BFTM0	Configurable <sup>(2)</sup>	0x084	BFTM0 global interrupt
18	34	BFTM1	Configurable <sup>(2)</sup>	0x088	BFTM1 global interrupt
19	35	I <sup>2</sup> C0	Configurable <sup>(2)</sup>	0x08C	I <sup>2</sup> C0 global interrupt
20	36	I <sup>2</sup> C1	Configurable <sup>(2)</sup>	0x090	I <sup>2</sup> C1 global interrupt
21	37	SPI0	Configurable <sup>(2)</sup>	0x094	SPI0 global interrupt
22	38	SPI1	Configurable <sup>(2)</sup>	0x098	SPI1 global interrupt
23	39	USART0 <sup>(5)</sup>	Configurable <sup>(2)</sup>	0x09C	USART0 global interrupt
24	40	USART1 <sup>(4)</sup>	Configurable <sup>(2)</sup>	0x0A0	USART1 global interrupt
25	41	UART0	Configurable <sup>(2)</sup>	0x0A4	UART0 global interrupt
26	42	UART1	Configurable <sup>(2)</sup>	0x0A8	UART1 global interrupt
27	43	UART2 <sup>(4)</sup>	Configurable <sup>(2)</sup>	0x0AC	UART2 global interrupt
28	44	UART3 <sup>(4)</sup>	Configurable <sup>(2)</sup>	0x0B0	UART3 global interrupt
29	45	LEDC	Configurable <sup>(2)</sup>	0x0B4	LED Controller global interrupt
30	46	PDMA_CH0 ~ 1 <sup>(4)</sup>	Configurable <sup>(2)</sup>	0x0B8	PDMA channel 0 & 1 global interrupt
31	47	PDMA_CH2 ~ 5 <sup>(4)</sup>	Configurable <sup>(2)</sup>	0x0BC	PDMA channel 2 ~ 5 global interrupt

- Notes: 1. The exception priority can be changed using the NVIC System Handler Priority Registers. For more information, refer to the Arm "Cortex®-M0+ Devices Generic User Guide" document.
2. The interrupt priority can be changed using the NVIC Interrupt Priority Registers. For more information, refer to the Arm "Cortex®-M0+ Devices Generic User Guide" document.
3. Refer to the PWRCU chapter for the relevant configuration descriptions about the WAKEUPn pin wakeup interrupt.
4. These exception types are only available for the HT32F54243/HT32F54253 devices.
5. Since there is only one USART in the HT32F54231/HT32F54241 devices, this exception type related to the USART do not have the serial number "0".

## Features

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

## Functional Descriptions

### SysTick Calibration

The SysTick Calibration Value Register (SYST\_CALIB) is provided by the NVIC to give a reference time base of 1ms for the RTOS tick timer or other purposes. The TENMS field in the SYST\_CALIB register has a fixed value of 7500 which is the Counter-Reload value to indicate 1 ms when the clock source comes from the SysTick reference input clock STCLK with a frequency of 7.5 MHz (60 MHz divide by 8).

## Register Map

The following table shows the NVIC registers and reset values.

**Table 26. NVIC Register Map**

Register	Offset	Description	Reset Value
<b>NVIC Base Address = 0xE000_E000</b>			
SYST_CSR	0x010	SysTick Control and Status Register	0x0000_0000
SYST_RVR	0x014	SysTick Reload Value Register	Unpredictable
SYST_CVR	0x018	SysTick Current Value Register	Unpredictable
SYST_CALIB	0x01C	SysTick Calibration Value Register	0x4000_1D4C
NVIC_ISER	0x100	Interrupt Set Enable Register	0x0000_0000
NVIC_ICER	0x180	Interrupt Clear Enable Register	0x0000_0000
NVIC_ISPR	0x200	Interrupt Set Pending Register	0x0000_0000
NVIC_ICPR	0x280	Interrupt Clear Pending Register	0x0000_0000
NVIC_IPR0	0x400	Interrupt 0 ~ 3 Priority Register	0x0000_0000
NVIC_IPR1	0x404	Interrupt 4 ~ 7 Priority Register	0x0000_0000
NVIC_IPR2	0x408	Interrupt 8 ~ 11 Priority Register	0x0000_0000
NVIC_IPR3	0x40C	Interrupt 12 ~ 15 Priority Register	0x0000_0000
NVIC_IPR4	0x410	Interrupt 16 ~ 19 Priority Register	0x0000_0000
NVIC_IPR5	0x414	Interrupt 20 ~ 23 Priority Register	0x0000_0000
NVIC_IPR6	0x418	Interrupt 24 ~ 27 Priority Register	0x0000_0000
NVIC_IPR7	0x41C	Interrupt 28 ~ 31 Priority Register	0x0000_0000
CPUID	0xD00	CPUID register	0x410C_C601
ICSR	0xD04	Interrupt Control and State Register	0x0000_0000
VTOR	0xD08	Vector Table Offset Register	0x0000_0000
AIRCR	0xD0C	Application Interrupt and Reset Control Register	0xFA05_0000
SCR	0xD10	System Control Register	0x0000_0000
CCR	0xD14	Configuration and Control Register	0x0000_0208
SHPR2	0xD1C	System Handlers Priority Register 2	0x0000_0000
SHPR3	0xD20	System Handlers Priority Register 3	0x0000_0000

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



# 11 External Interrupt/Event Controller (EXTI)

## Introduction

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

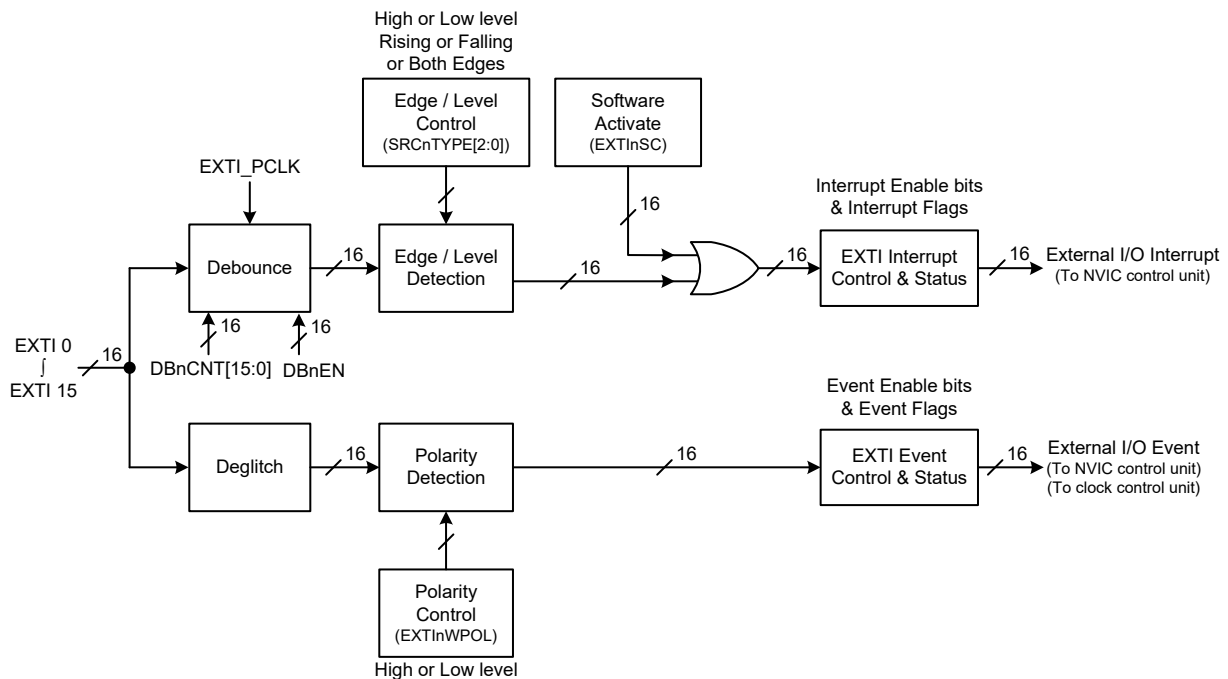


Figure 26. EXTI Block Diagram

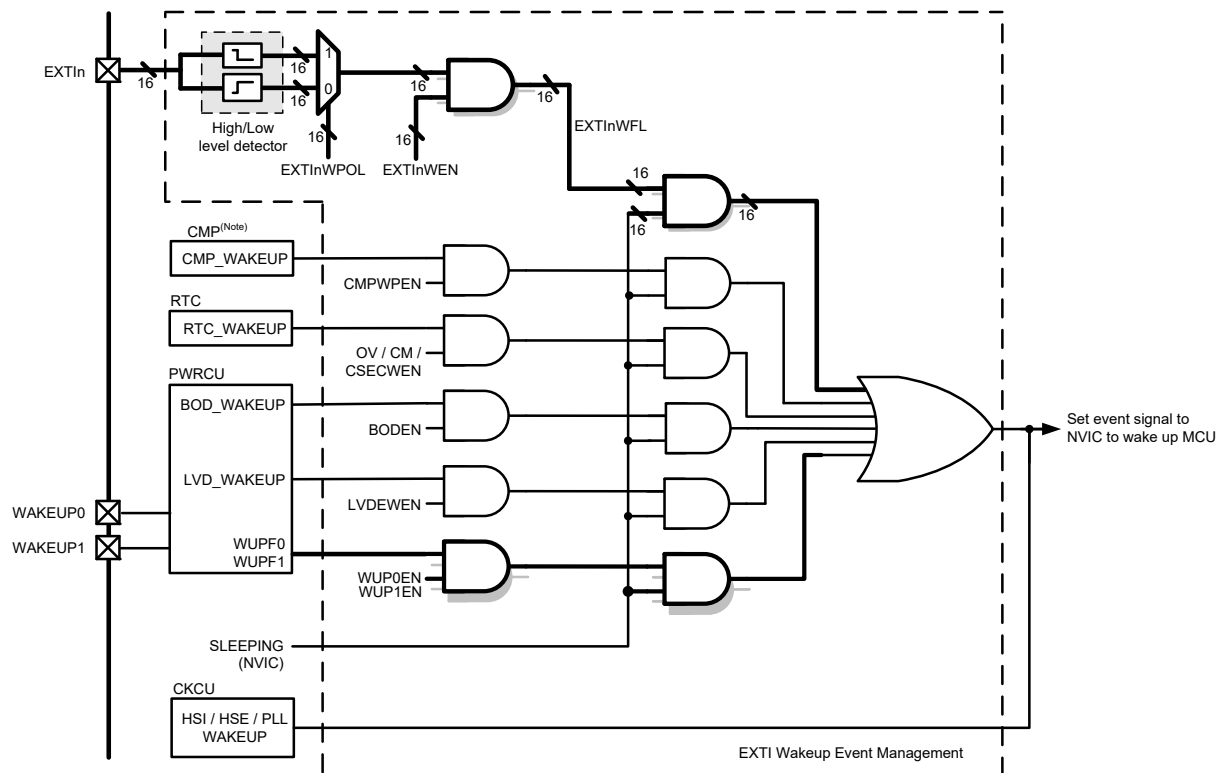
## Features

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

## Functional Descriptions

### Wakeup Event Management

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

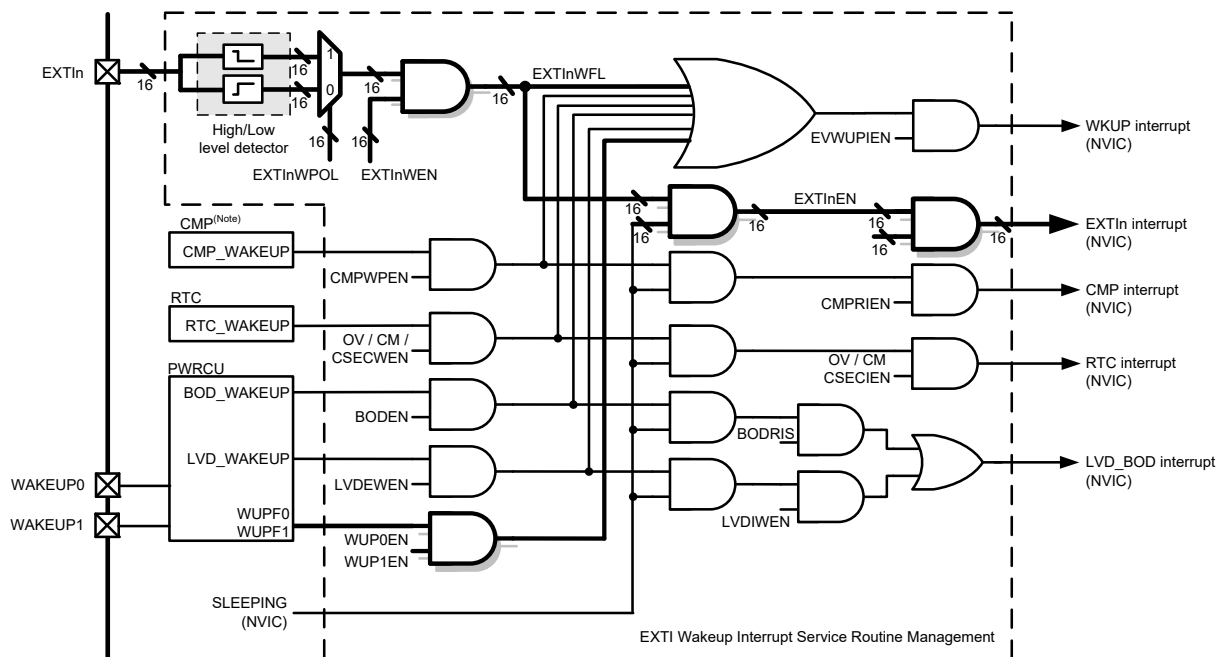


Note: CMP is only available for the HT32F54243/HT32F54253 devices.

**Figure 27. EXTI Wakeup Event Management**

### External Interrupt/Event Line Mapping

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

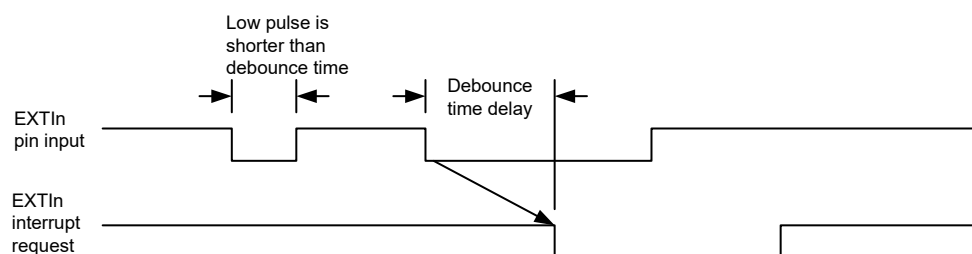


Note: CMP is only available for the HT32F54243/HT32F54253 devices.

**Figure 28. EXTI Wakeup Interrupt Service Routine Management**

## Interrupt and Debounce

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



**Figure 29. EXTI Interrupt Debounce Function**

## Register Map

The following table shows the EXTI registers and reset values.

**Table 27. EXTI Register Map**

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

## Register Descriptions

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

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

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

Reset value: 0x0000\_0000

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

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

## EXTI Interrupt Control Register – EXTICR

This register is used to control the EXTI interrupt.

Offset: 0x040

Reset value: 0x0000\_0000

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

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

## EXTI Interrupt Edge Flag Register – EXTIEDGEFLGR

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

Offset: 0x044

Reset value: 0x0000\_0000

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

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

## EXTI Interrupt Edge Status Register – EXTIEDGESR

This register indicates the polarity of a detected EXTI edge.

Offset: 0x048

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	EXTI15EDS	EXTI14EDS	EXTI13EDS	EXTI12EDS	EXTI11EDS	EXTI10EDS	EXTI9EDS	EXTI8EDS
	WC	0	WC	0	WC	0	WC	0
	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 Edge Detection Status (n = 0 ~ 15) 0: Negative edge is detected 1: Positive edge is detected Software should write 1 to clear it.



## EXTI Interrupt Software Set Command Register – EXTISSCR

This register is used to activate the EXTI interrupt.

Offset: 0x04C

Reset value: 0x0000\_0000

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

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

## EXTI Interrupt Wakeup Control Register – EXTIWAKUPCR

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

Offset: 0x050

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
	EVWUPIEN	Reserved						
Type/Reset	RW	0						
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	EXTI15WEN	EXTI14WEN	EXTI13WEN	EXTI12WEN	EXTI11WEN	EXTI10WEN	EXTI9WEN	EXTI8WEN
Type/Reset	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

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

## EXTI Interrupt Wakeup Polarity Register – EXTIWAKUPPOLR

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

Offset: 0x054

Reset value: 0x0000\_0000

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

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

## EXTI Interrupt Wakeup Flag Register – EXTIWAKUPFLG

This register is the EXTI interrupt wakeup flag register.

Offset: 0x058

Reset value: 0x0000\_0000

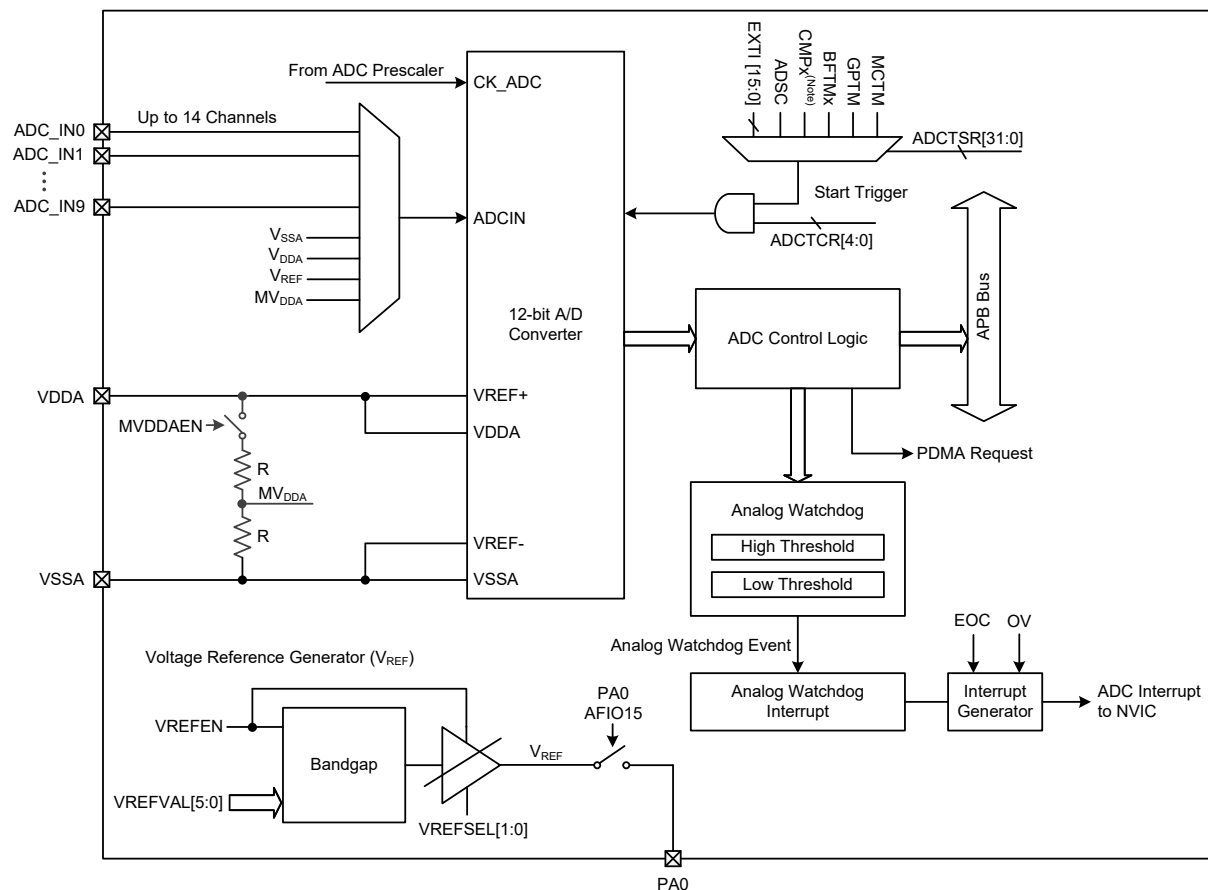
	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	EXTI15WFL	EXTI14WFL	EXTI13WFL	EXTI12WFL	EXTI11WFL	EXTI10WFL	EXTI9WFL	EXTI8WFL
	WC	0	WC	0	WC	0	WC	0
	7	6	5	4	3	2	1	0
Type/Reset	EXTI7WFL	EXTI6WFL	EXTI5WFL	EXTI4WFL	EXTI3WFL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

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

# 12 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 14 multiplexed channels including 10 external channels on which the external analog signal can be supplied and 4 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.



Note: The CMP0/CMP1 trigger is only available for the HT32F54243/HT32F54253 devices.

**Figure 30. ADC with  $V_{REF}$  Block Diagram**

## Features

- 12-bit SAR ADC engine
- Up to 1 Msps conversion rate
- Up to 10 external analog input channels
- 1 channel for internal voltage reference ( $V_{REF}$ )
- 1 channel for monitor external  $V_{DDA}$  power support pin
- 1 channel for monitor  $V_{SSA}$
- 1 channel for monitor  $MV_{DDA}$
- 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
  - MCTM trigger
  - GPTM trigger
  - BFTM0 / BFTM1 / BFTM2 / BFTM3 trigger, the BFTM2 and BFTM3 are only available for HT32F54243 and HT32F54253 devices
  - CMP0 / CMP1 trigger (HT32F54243/HT32F54253 only)
- 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 (HT32F54243/HT32F54253 only)

## 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 (ADCEN bit = '0').

### Channel Selection

The A/D converter supports 14 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

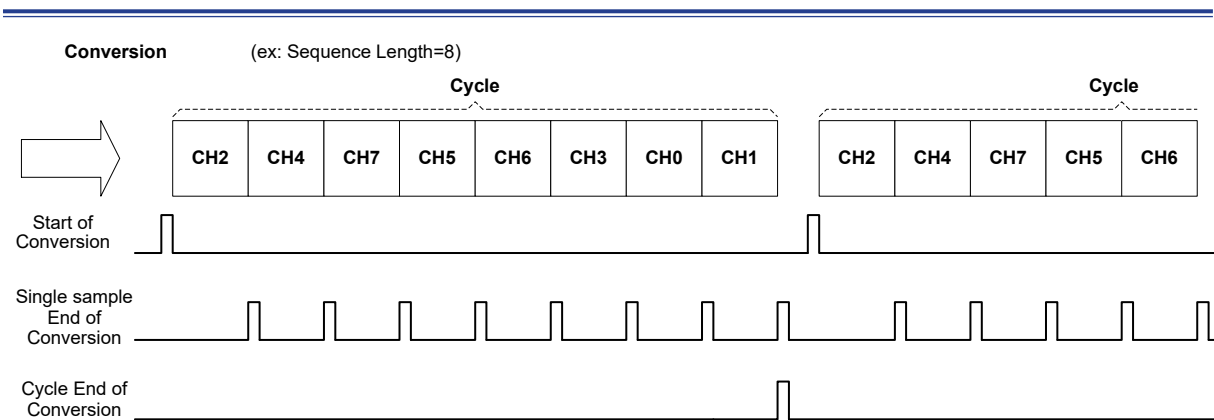
The A/D has three operating conversion modes. The conversion modes are One Shot Conversion Mode, Continuous Conversion Mode and Discontinuous Conversion mode. Details are provided later.

#### 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, a comparator output transition event, an external EXTI event or a TM event determined by the Trigger Control Register ADCTCR and the Trigger Source Register ADCTSR.

#### After Conversion:

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



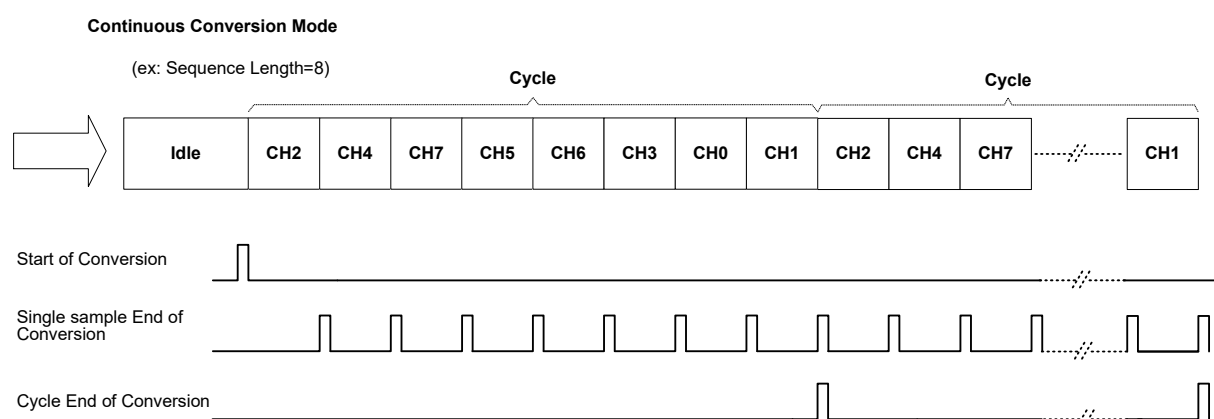
**Figure 31. One Shot Conversion Mode**

### 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, a comparator output transition event, an external `EXTI` event or a `TM` event determined by the Trigger Control Register `ADCTCR` and the Trigger Source Register `ADCTSR`.

#### After conversion:

- The converted data will be stored in the 16-bit `ADCDy` ( $y = 0 \sim 7$ ) 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 32. Continuous Conversion Mode**



### Discontinuous Conversion Mode

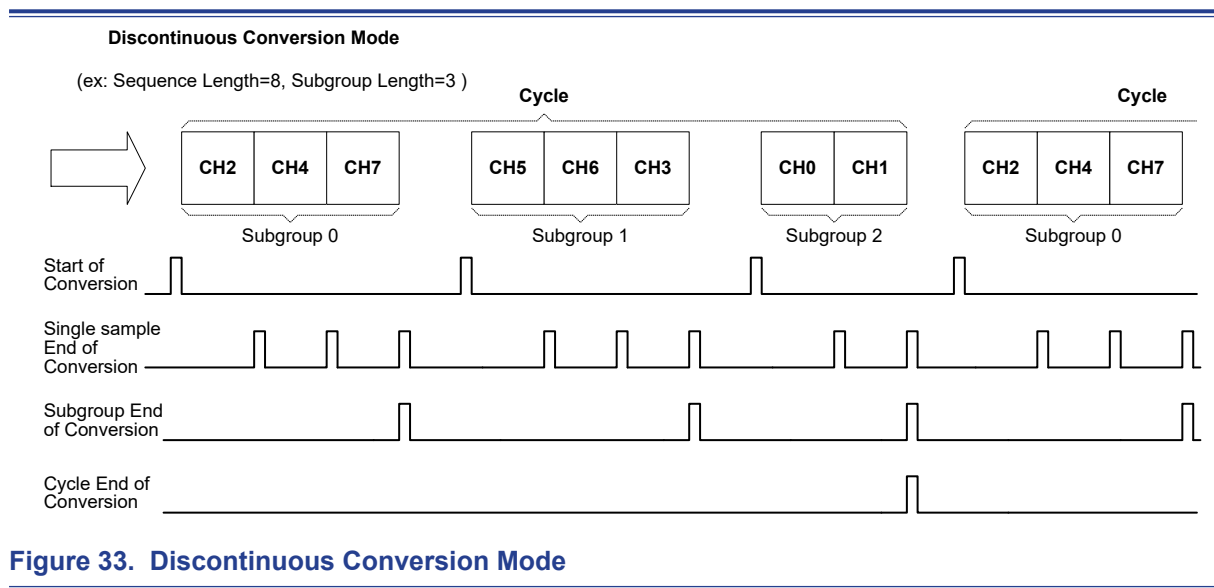
The A/D converter will operate in the Discontinuous Conversion Mode for channels group when the A/D conversion mode bit field ADMODE [1:0] in the ADCCR register is set to 0x3. The group to be converted can have up to 8 channels and can be arranged in a specific sequence by configuring the ADCLSTn registers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

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



## Start Conversion on External Event

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

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

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

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

## Sampling Time Setting

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

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

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

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

Example:

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

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

## Data Format

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

**Table 28. Data Format in ADCDR [15:0]**

Description	ADCDR register Data Format
Right aligned	"0_0_0_0_d11_d10_d9_d8_d7_d6_d5_d4_d3_d2_d1_d0"

## Analog Watchdog

The A/D converter includes a watchdog function to monitor the converted data. There are two kinds of thresholds for the watchdog monitor function, known as the watchdog 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.

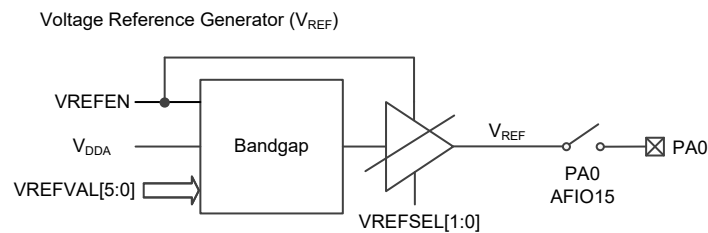
The A/D Converter interrupt clear bits are used to clear the associated A/D converter interrupt raw and interrupt status bits. Writing a 1 into the specific A/D converter interrupt clear bit in the A/D converter interrupt clear register ADCICLR will clear the corresponding A/D converter interrupt raw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

## PDMA Request (HT32F54243/HT32F54253 only)

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 and Comparators. 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 34. 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 29. A/D Converter Register Map**

Register	Offset	Description	Reset Value
ADCCR	0x000	ADC Conversion Control Register	0x0000_0000
ADCLST0	0x004	ADC Conversion List Register 0	0x0000_0000
ADCLST1	0x008	ADC Conversion List Register 1	0x0000_0000
ADCSTR	0x020	ADC Input Sampling Time Register	0x0000_0000
ADCDR0	0x030	ADC Conversion Data Register 0	0x0000_0000
ADCDR1	0x034	ADC Conversion Data Register 1	0x0000_0000
ADCDR2	0x038	ADC Conversion Data Register 2	0x0000_0000
ADCDR3	0x03C	ADC Conversion Data Register 3	0x0000_0000
ADCDR4	0x040	ADC Conversion Data Register 4	0x0000_0000
ADCDR5	0x044	ADC Conversion Data Register 5	0x0000_0000
ADCDR6	0x048	ADC Conversion Data Register 6	0x0000_0000
ADCDR7	0x04C	ADC Conversion Data Register 7	0x0000_0000
ADCTCR	0x070	ADC Trigger Control Register	0x0000_0000
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 PDMA Request Register	0x0000_0000
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					ADSUBL		
	15	14	13	12	11	10	9	8
Type/Reset	Reserved					RW	0	RW
	7	6	5	4	3	2	1	0
Type/Reset	ADCEN	ADCRST	Reserved				ADMODE	
	RW	0	RW	0			RW	0

Bits	Field	Descriptions
[18:16]	ADSUBL	ADC Conversion Subgroup Length The ADSUBL field specifies the conversion channel length of each subgroup in the Discontinuous Conversion Mode. Subgroup length = ADSUBL [2:0] + 1. If the sequence length (ADSEQL [2:0] + 1) is not a multiple of the subgroup length (ADSUBL [2:0] + 1), the last subgroup will be the rest of the group channels that have not been converted.
[10:8]	ADSEQL	ADC Conversion Length 0x0: The channel specified by the ADSEQ0 field in the ADCLST0 register will be converted Others: Sequence length = ADSEQL [2:0] + 1 The ADSEQL field specifies the whole conversion sequence length for the conversion group.
[7]	ADCEN	ADC Enable 0: ADC is disabled 1: ADC is enabled When this bit is cleared to 0, the A/D converter will be disabled and the CK_ADC clock will also be switched off.
[6]	ADCRST	ADC Reset 0: No effect 1: Reset A/D converter except for the A/D Converter controller

Bits	Field	Descriptions															
[1:0]	ADMODE	ADC Conversion Mode															
		<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: ADC_IN8 0x09: ADC_IN9 0x0A: Analog ground V <sub>SSA</sub> 0x0B: Analog power V <sub>DDA</sub> 0x0C: Internal Voltage Reference (V <sub>REF</sub> ) 0x0D: Analog power MV <sub>DDA</sub> (V <sub>DDA</sub> /2) 0x0E ~ 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: ADC_IN8 0x09: ADC_IN9 0x0A: Analog ground V <sub>SSA</sub> 0x0B: Analog power V <sub>DDA</sub> 0x0C: Internal Voltage Reference (V <sub>REF</sub> ) 0x0D: Analog power MV <sub>DDA</sub> (V <sub>DDA</sub> /2) 0x0E ~ 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

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

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

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

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

Offset: 0x030 ~ 0x04C

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
	ADVLDy		Reserved					
Type/Reset	RC	0						
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	ADDy							
Type/Reset	RO	0	RO	0	RO	0	RO	0
	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			CMP	TM1	TM0	ADEXTI	ADSW
				RW	0	RW	0	RW
								0

Bits	Field	Descriptions
[4]	CMP	ADC Conversion CMP Event Trigger enable control 0: Disable conversion trigger by CMP output transition 1: Enable conversion trigger by CMP output transition This bit is only available for the HT32F54243/HT32F54253 devices.
[3]	TM1	ADC Conversion BFTM Event Trigger enable control 0: Disable conversion trigger by BFTM events 1: Enable conversion trigger by BFTM events
[2]	TM0	ADC Conversion GPTM or MCTM Event Trigger enable control 0: Disable conversion trigger by GPTM or MCTM events 1: Enable conversion trigger by GPTM or MCTM events
[1]	ADEXTI	ADC Conversion EXTI Event Trigger enable control 0: Disable conversion trigger by EXTI lines 1: Enable conversion trigger by EXTI lines
[0]	ADSW	ADC Conversion Software Trigger enable control 0: Disable conversion trigger by software trigger bit 1: Enable conversion trigger by software trigger bit

## ADC Trigger Source Register – ADCTSR

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

Offset: 0x074

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	Reserved					TM0E			
Type/Reset						RW	0	RW	0
	23	22	21	20	19	18	17	16	
	TM1S[2:1]		Reserved	CMPS	TM1S[0]	TM0S			
Type/Reset	RW	0	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	0

Bits	Field	Descriptions
[26:24]	TM0E	GPTM or MCTM Trigger Event Selection of ADC Conversion 000: GPTM or MCTM MTO event 001: GPTM or MCTM CH00 event 010: GPTM or MCTM CH10 event 011: GPTM or MCTM CH20 event 100: GPTM or MCTM CH30 event Others: Reserved – Should not be used to avoid unpredictable results
[20]	CMPS	CMP Trigger Selection of ADC Conversion 0: CMP0 1: CMP1 This bit is only available for the HT32F54243/HT32F54253 devices.
[23:22], [19]	TM1S	BFTM Trigger Timer Selection of ADC Conversion 000: BFTM0 001: BFTM1 Others: Reserved – Should not be used to avoid unpredictable results
[18:16]	TM0S	GPTM or MCTM Trigger Timer Selection of ADC Conversion 000: MCTM 010: GPTM Others: Reserved - Should not be used to avoid unpredictable results
[11:8]	ADEXTIS	EXTI Trigger Source Selection of ADC Conversion 0x00: EXTI line 0 0x01: EXTI line 1 ... 0x0F: EXTI line 15 Note that the EXTI line active edge to start an A/D conversion is determined in the External Interrupt/Event Control Unit, EXTI.
[0]	ADSC	ADC Conversion Software Trigger Bit 0: No operation 1: Start conversion immediately This bit is set by software to start a conversion manually and 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 ... 1001: ADC_IN9 converted data is higher than the upper threshold Others: Reserved If one of these status bits is set to 1 by the watchdog monitor function, the status field value should first be stored in the user-defined memory location in the corresponding ISR. Otherwise, the ADUCH field will be changed if another input channel converted data is higher than the upper threshold.
[19:16]	ADLCH	Lower Threshold Channel Status 0000: ADC_IN0 converted data is lower then the lower threshold 0001: ADC_IN1 converted data is lower then the lower threshold ... 1001: ADC_IN9 converted data is lower then the lower threshold Others: Reserved If one of these status bits is set to 1 by the watchdog monitor function, the status field value should first be stored in the user-defined memory location in the corresponding ISR. Otherwise, the ADLCH field will be changed if another input channel converted data is lower than the lower threshold.
[11:8]	ADWCH	ADC Watchdog Specific Channel Selection 0000: ADC_IN0 is monitored 0001: ADC_IN1 is monitored ... 1001: ADC_IN9 is monitored Others: Reserved
[2]	ADWALL	ADC Watchdog Specific or All Channel Setting 0: Only the channel which specified by the ADWCH field is monitored 1: All channels are monitored

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

### ADC Watchdog Threshold Register – ADCTR

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

Offset: 0x07C

Reset value: 0x0000\_0000

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

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 PDMA Request Register – ADCDMAR

This register contains the ADC PDMA 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 PDMA Request Enable Bit 0: ADC cycle end of conversion PDMA request is disabled 1: ADC cycle end of conversion PDMA request is enabled
[1]	ADDMAG	ADC Subgroup EOC PDMA Request Enable Bit 0: ADC subgroup end of conversion PDMA request is disabled 1: ADC subgroup end of conversion PDMA request is enabled
[0]	ADDMAS	ADC Single EOC PDMA Request Enable Bit 0: ADC single end of conversion PDMA request is disabled 1: ADC single end of conversion PDMA 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: 2.5 V 01: 3.0 V 10: 4.0 V 11: 4.5 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

Bits	Field	Descriptions
[5:0]	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 Comparator (CMP)

## Introduction

Two general purpose comparators (CMP) are implemented within the devices. They can be configured either as standalone comparators or combined with the different kinds of peripheral IP. Each comparator is capable of asserting interrupts to the NVIC or waking up the CPU from the Sleep, Deep-Sleep1 or Deep-Sleep2 mode through the EXTI wakeup event management unit. This function is only available for the HT32F54243/HT32F54253 devices.

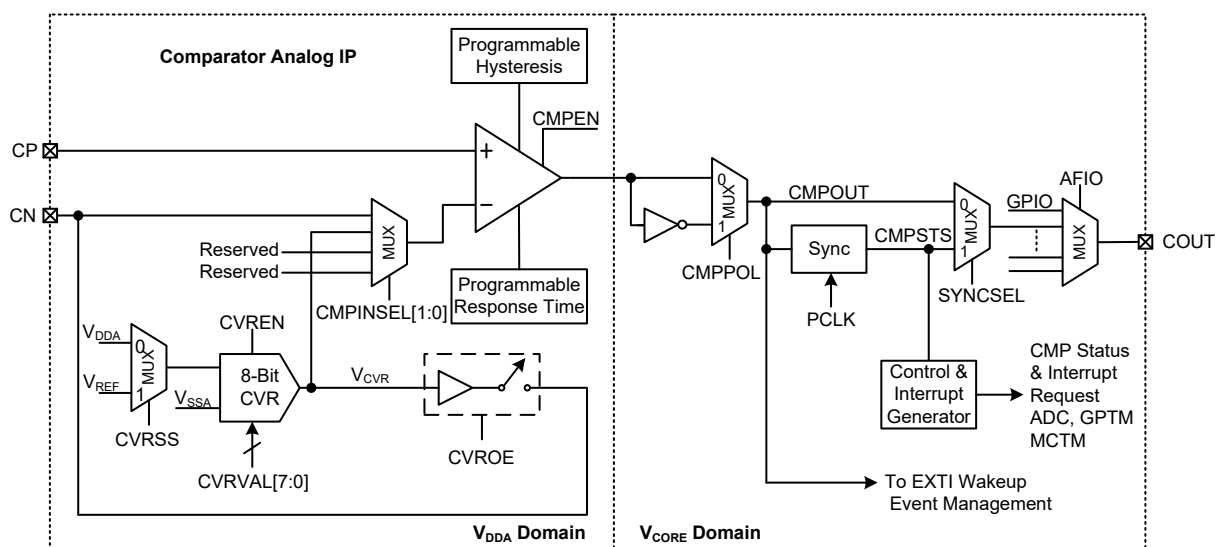


Figure 35. Comparator Block Diagram

## Features

- Rail-to-rail comparator
- Configurable negative inputs for flexible voltage selection
  - External CN pin
  - Internal 8-bit CVR output
- Programmable hysteresis
- Programmable response speed and power consumption
- Comparator output can be routed to I/O pin, to multiple timers or to ADC trigger inputs
- 8-bit CVR can be configured to dedicated I/O for voltage reference
- Comparator has interrupt generation capability with wakeup function from the Sleep, Deep-Sleep1 or Deep-Sleep2 mode through the EXTI controller

## Functional Descriptions

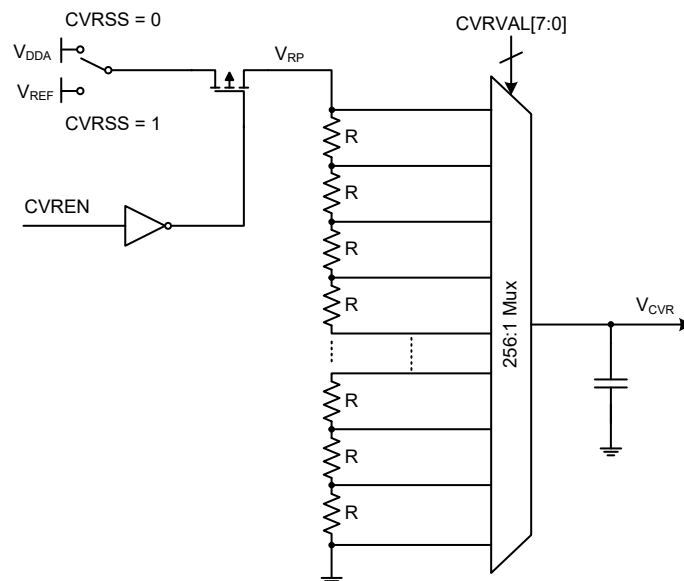
### Comparator Inputs and Output

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

### Comparator Voltage Reference

The comparator voltage reference is a 256-tap resistor ladder network that provides a selectable reference voltage. A block diagram of the comparator voltage reference is shown in the figure above. It also has a power-down function to conserve power when the reference is not used. The comparator voltage reference provides 256 distinct levels. The equation used to calculate the value of the reference voltage is as follows:

$$V_{CVR} = CVRVAL \times (V_{RP} - V_{SSA}) / 255$$



**Figure 36. Comparator Voltage Reference Block Diagram**

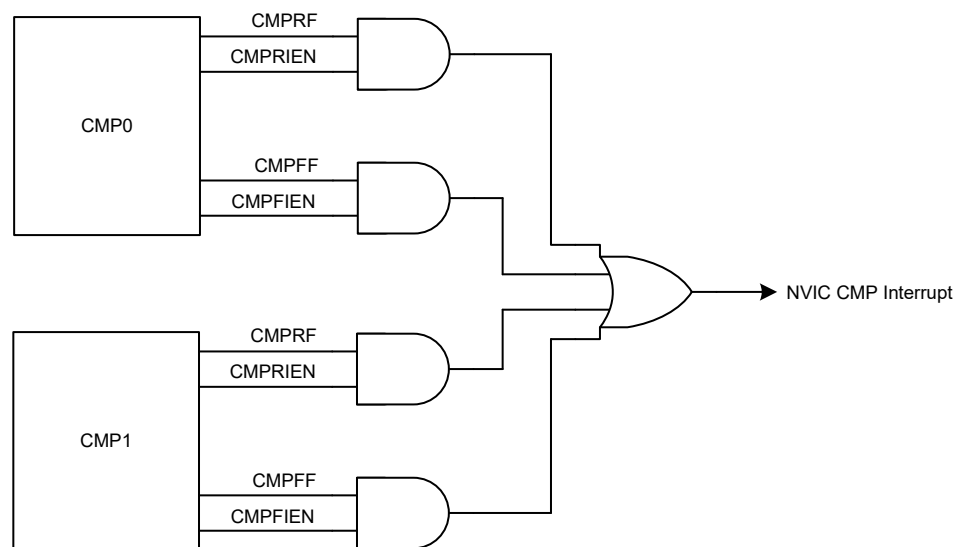
The supply voltage  $V_{RP}$  can come from either the  $V_{DDA}$  or the internal voltage reference  $V_{REF}$  by configuring the CVRSS bit in the Comparator Control Register CMPCRn. The CVR output  $V_{CVR}$  is used to provide a reference voltage for the analog comparator. It can be internally used or configured to connect to the CN pin by setting the CVROE bit in the Comparator Control Register CMPCRn. The settling time of the comparator voltage reference must be considered when the  $V_{CVR}$  output voltage is changed.



## Interrupts and Wakeup

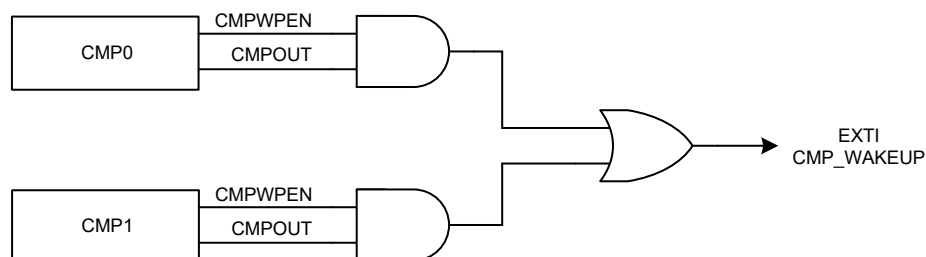
The comparator can generate an interrupt when its output waveform generates a rising or falling edge and its corresponding interrupt enable control bit is also set.

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



**Figure 37. Comparator Interrupt Signals**

The comparator outputs are also internally connected to the EXTI Wakeup Event Management unit. The comparator output rising transition is used to wake up the MCU from the Sleep, Deep-Sleep1 or Deep-Sleep2 mode when the comparator wakeup enable bit CMPWPEN is set in the Comparator Control Register CMPCRn. A block diagram of wakeup signals for comparators is shown in the following figure.



**Figure 38. Comparator Wakeup Signals**

## Power Mode and Hysteresis

The comparator response time can be programmed to meet the trade-off between the power consumption and application speed requirements. The bit CMPSM in the CMPCRn register can be programmed as “0” to make the comparator operate in the low speed mode with low power consumption.

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

## Comparator Write-Protected Mechanism

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

## Register Map

The following table shows the CMP registers and reset values.

**Table 30. CMP Register Map**

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

## Register Descriptions

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

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

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

Reset value: 0x0001\_0000

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

Bits	Field	Descriptions
[31:16]	PROTECT	Register Protection For write operation: 0x9C3A: Disable the CMPCRn register write protection Others values: Enable the CMPCRn register write protection For read operation: 0x0000: CMPCRn register write protection is disabled 0x0001: CMPCRn register write protection is enabled These bits are used to enable or disable the write protection of the field [14:0] in the CMPCRn register. Enabling the write protection will make the field [14:0] in the CMPCRn register become read-only to prevent any unexpected write operation. The value read from this field indicates if the write protection is enabled or not.
[15]	CMPSTS	Comparator Output Status 0: Output is low 1: Output is high This read-only bit is a copy of the comparator output state after the polarity selection and synchronization.
[14]	CMPWPEN	Comparator Wakeup Enable 0: Disable comparator wakeup function 1: Enable comparator wakeup function This bit is used to enable the MCU wakeup function from the Sleep, Deep-Sleep1 or Deep-Sleep2 mode when the comparator output after the polarity selection changes state from low to high.

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

### 13 Comparator (CMP)

Offset: 0x004 (n = 0), 0x104 (n = 1)  
Reset value: 0x0000 0000

Bits	Field	Descriptions
[7:0]	CVRVAL	<p>Comparator Voltage Reference Value</p> <p>There are 256 levels of the comparator voltage reference, which is set using the CVRVAL bits. The relationship between the CVRVAL register value and the CVR output, <math>V_{CVR}</math>, is given by the following equation:</p> $V_{CVR} = \text{CVRVAL} \times (V_{RP} - V_{SSA}) / 255, \text{ where } V_{RP} \text{ can come from the } V_{DDA} \text{ or } V_{REF}.$

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

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

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

Reset value: 0x0000\_0000

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

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

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

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

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

Reset value: 0x0000\_0000

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

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

# 14 General-Purpose Timer (GPTM)

## Introduction

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

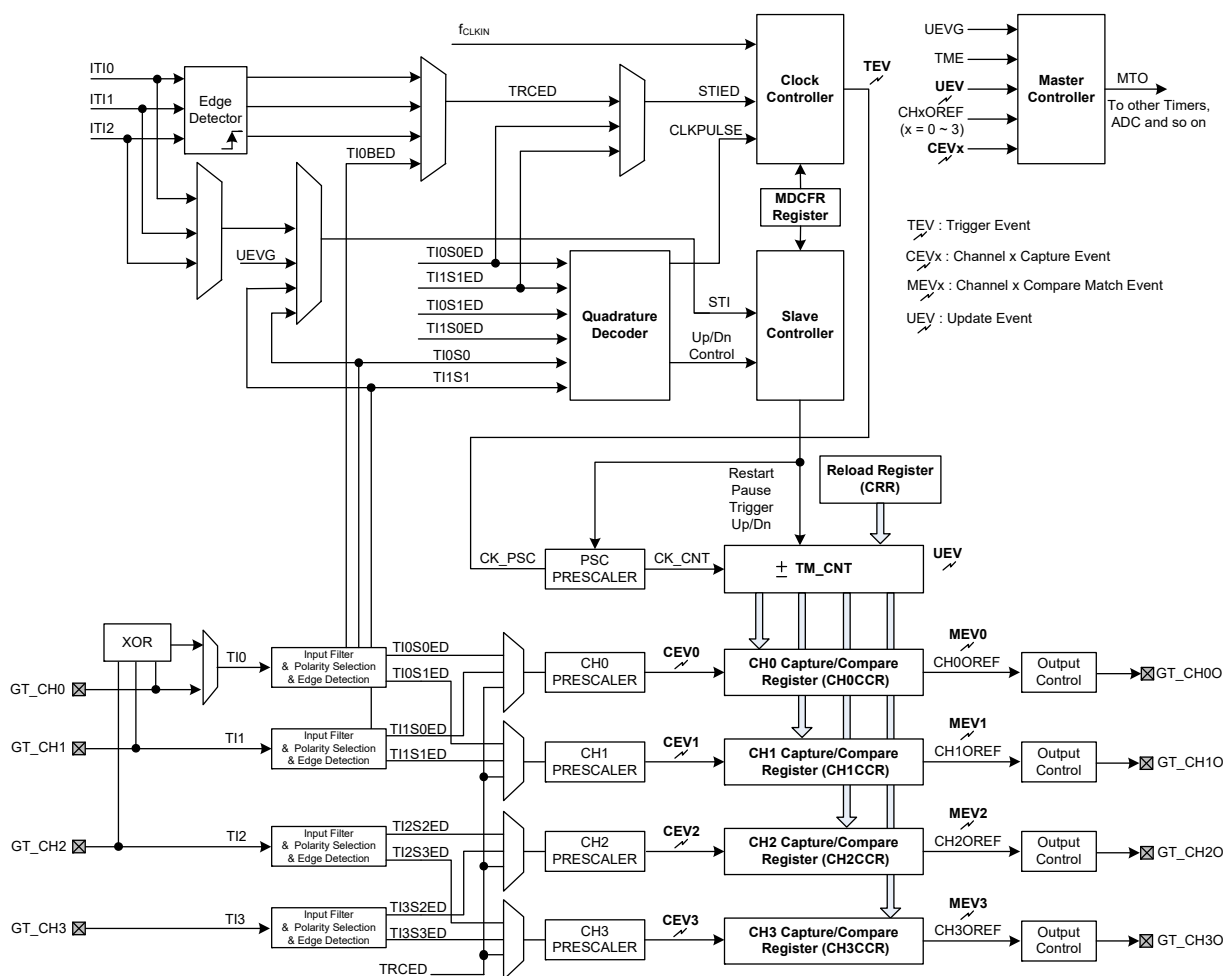


Figure 39. GPTM Block Diagram



## Features

- 16-bit up/down auto-reload counter
- 16-bit programmable prescaler that allows division of the prescaler clock source by any factor between 1 and 65536 to generate the counter clock frequency
- Up to 4 independent channels for:
  - Input Capture function
  - Compare Match Output
  - Generation of PWM waveform – Edge and Center-aligned Mode
  - Single Pulse Mode Output
- Encoder interface controller with two inputs using quadrature decoder
- Synchronization circuit to control the timer with external signals and to interconnect several timers together
- Interrupt/PDMA generation with the following events, PDMA is only available for the HT32F54243/HT32F54253 devices
  - Update event
  - Trigger event
  - Input capture event
  - Output compare match event
- GPTM Master/Slave mode controller

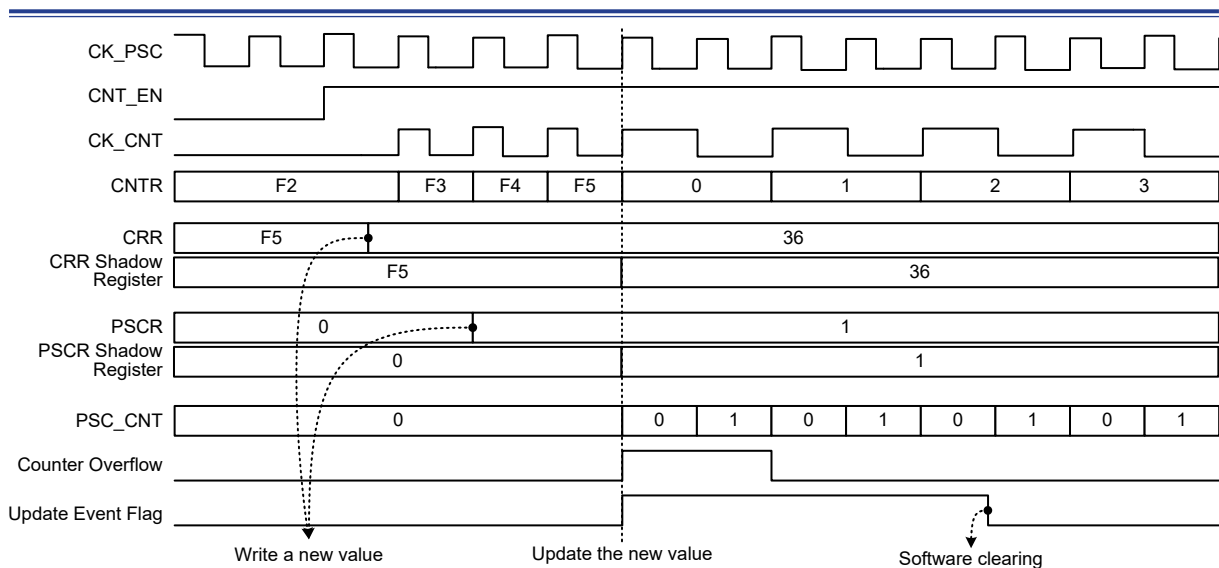
## Functional Descriptions

### 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, 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.

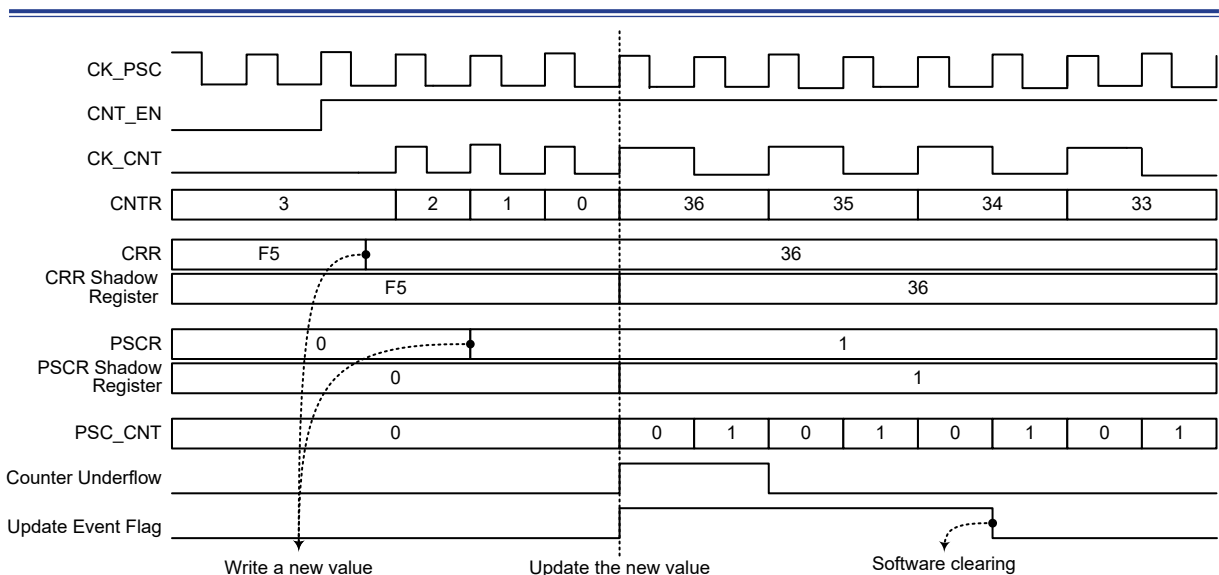


**Figure 40. Up-counting Example**

### 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, 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.



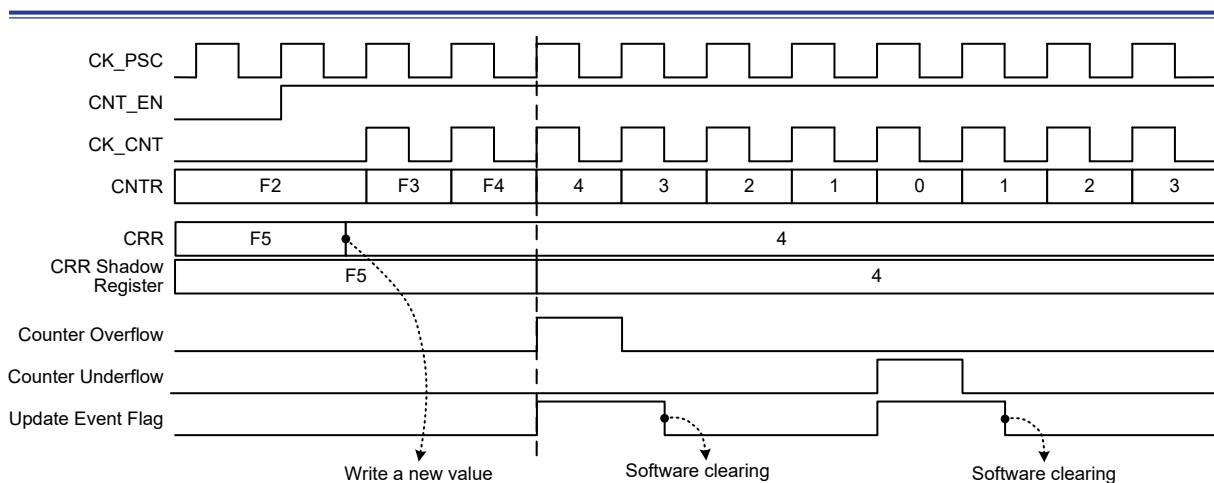
**Figure 41. Down-counting Example**

### 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 42. 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 $f_{CLKIN}$ :

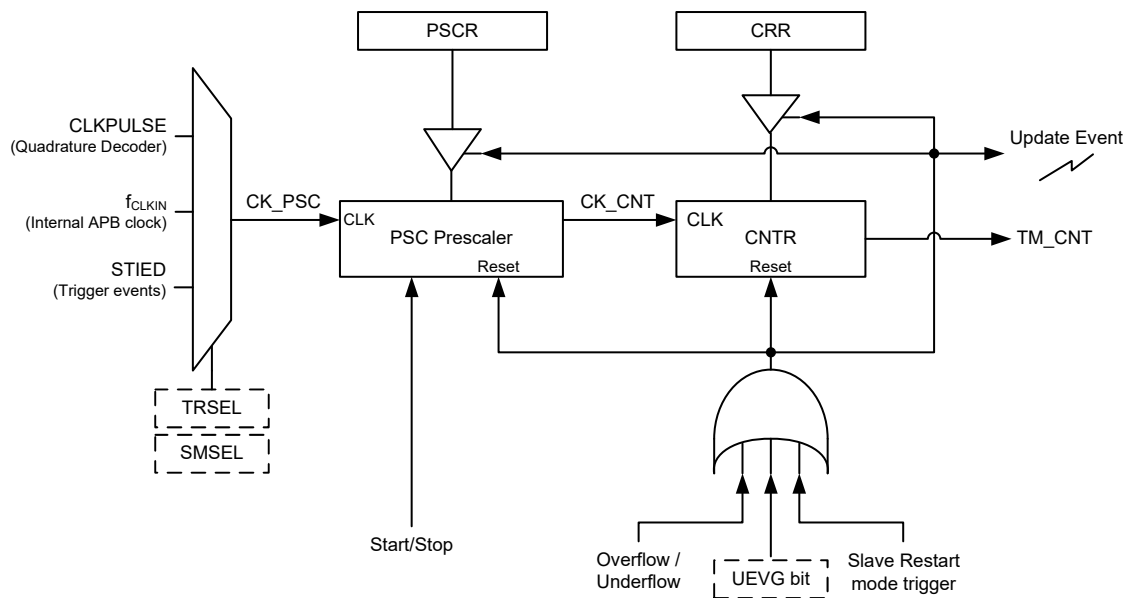
The default internal clock source is the APB clock  $f_{CLKIN}$  used to drive the counter prescaler when the slave mode is disabled. When the slave mode selection bits SMSEL in the MDCFR register are set to 0x4, 0x5 or 0x6, the internal APB clock  $f_{CLKIN}$  is the counter prescaler driving clock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



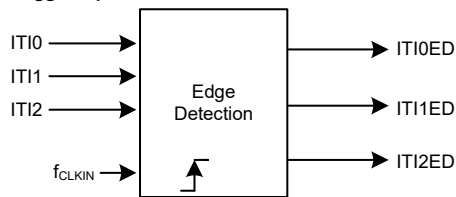
**Figure 43. GPTM Clock Source Selection**

## Trigger Controller

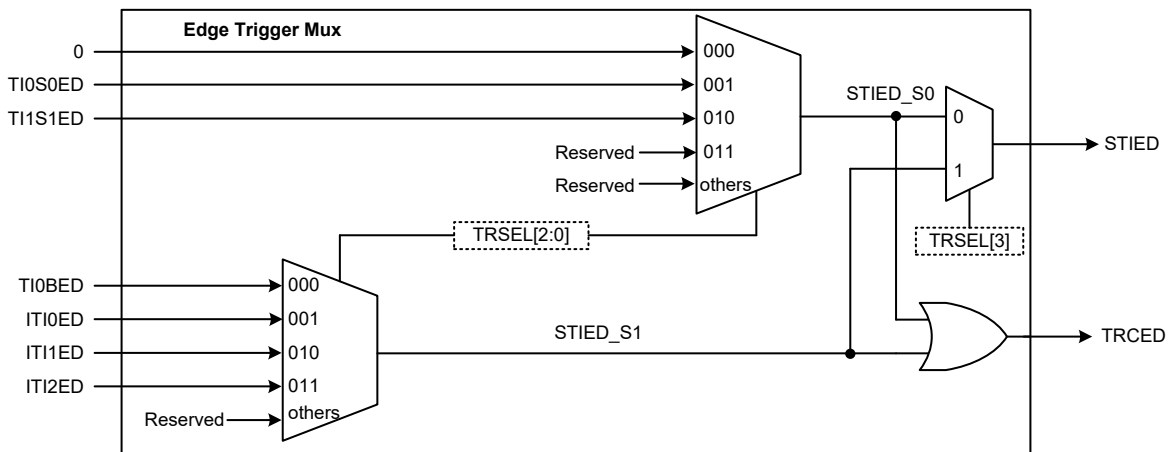
The trigger controller is used to select the trigger source and setup the trigger level or trigger edge condition. For the internal trigger input, it can be selected by the Trigger Selection bits TRSEL in the TRCFR register. For all the trigger sources except the UEVG bit software trigger, the internal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

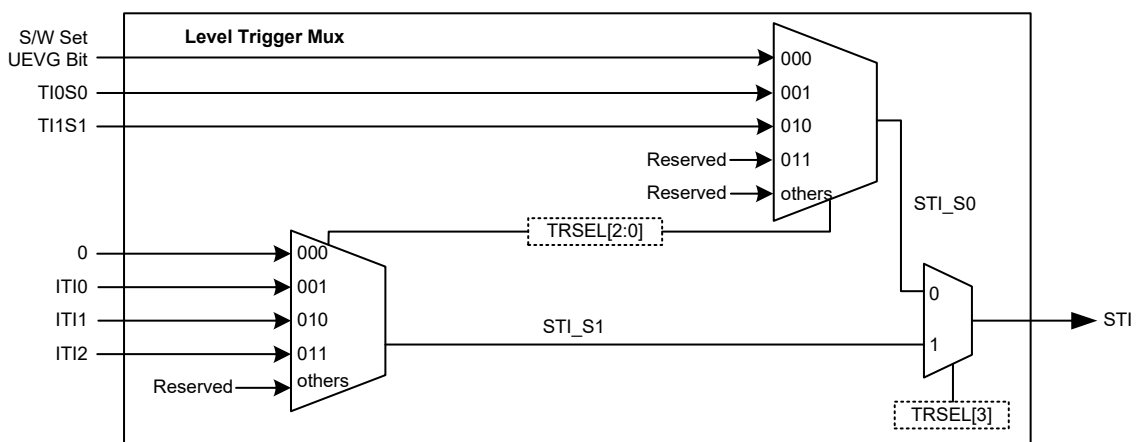
**Internal Trigger Input**



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



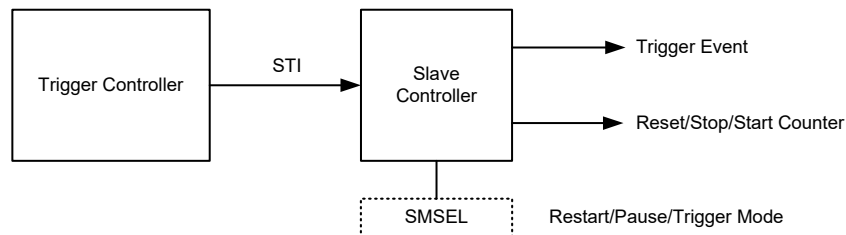
**Level Trigger Source = Internal (ITIx) + Channel input (TIn) + Software UEVG bit**



**Figure 44. Trigger Controller Block**

## Slave Controller

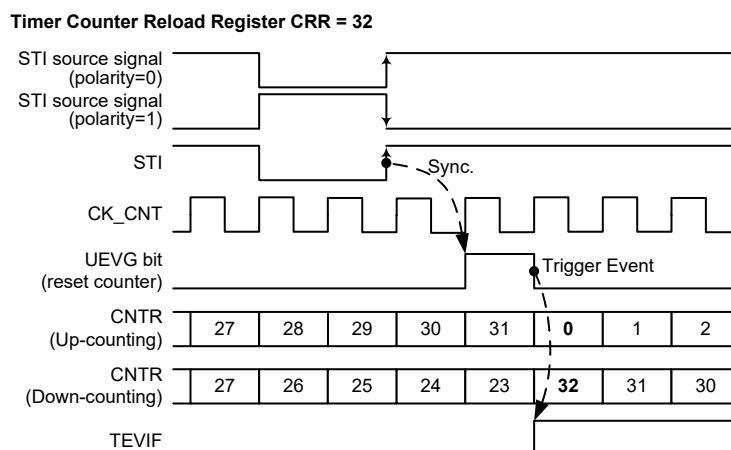
The GPTM can be synchronized with an external trigger in several modes including the Restart mode, the Pause mode and the Trigger mode which can be selected by the SMSEL field in the MDCFR register. The trigger input of these modes comes from the STI signal which is selected by the TRSEL field in the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



**Figure 45. Slave Controller Diagram**

### Restart Mode

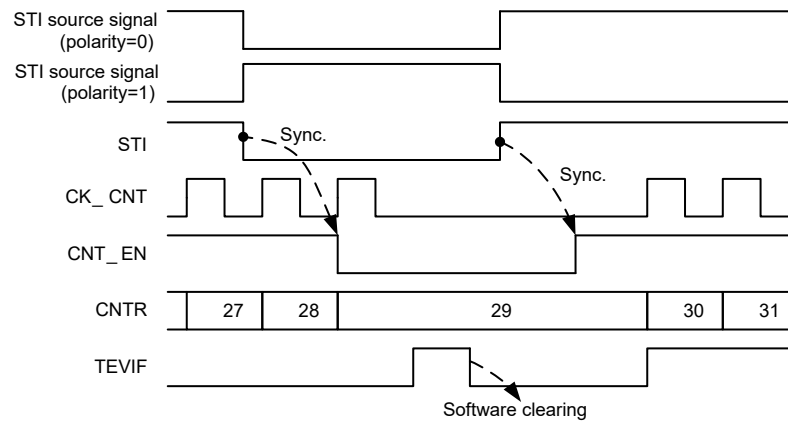
The counter and its prescaler can be reinitialized in response to a rising edge of the STI signal. When an STI rising edge occurs, the update event software generation bit named UEVG will automatically be asserted by hardware and the trigger event flag will also be set. Then the counter and prescaler 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 46. GPTM in Restart Mode**

### Pause Mode

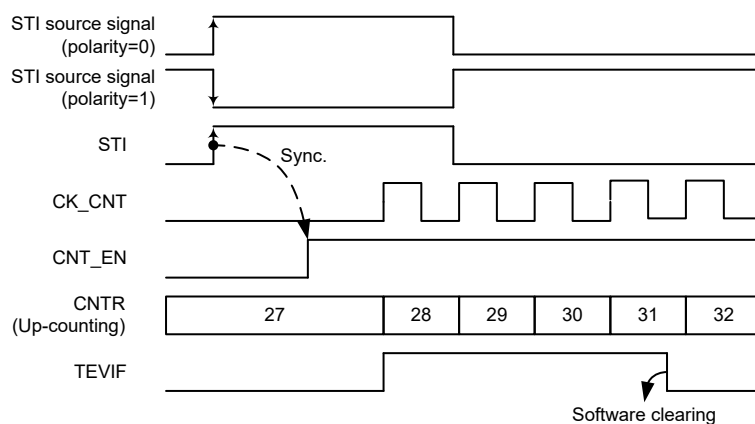
In the Pause Mode, the selected STI input signal level is used to control the counter start/stop operation. The counter starts to count when the selected STI signal is at a high level and stops counting when the STI signal is changed to a low level, here the counter will maintain its present value and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



**Figure 47. GPTM in Pause Mode**

### Trigger Mode

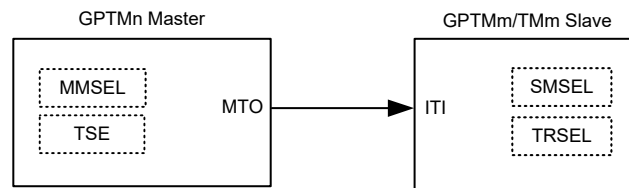
After the counter is disabled to count, the counter can resume counting when an STI rising edge signal occurs. When an STI rising edge occurs, the counter will start to count from the current value in the counter. Note that if the STI signal is selected to be derived from the UEVG bit software trigger, 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 48. GPTM in Trigger Mode**

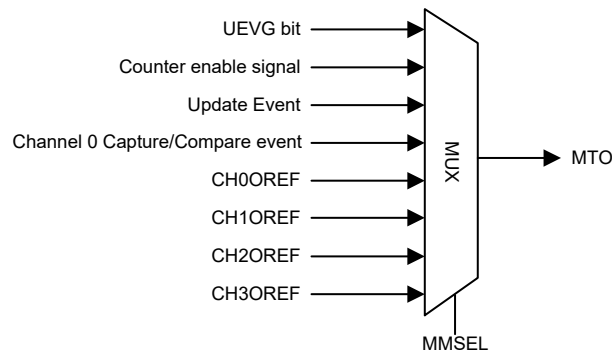
## Master Controller

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 49. 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 50. 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.

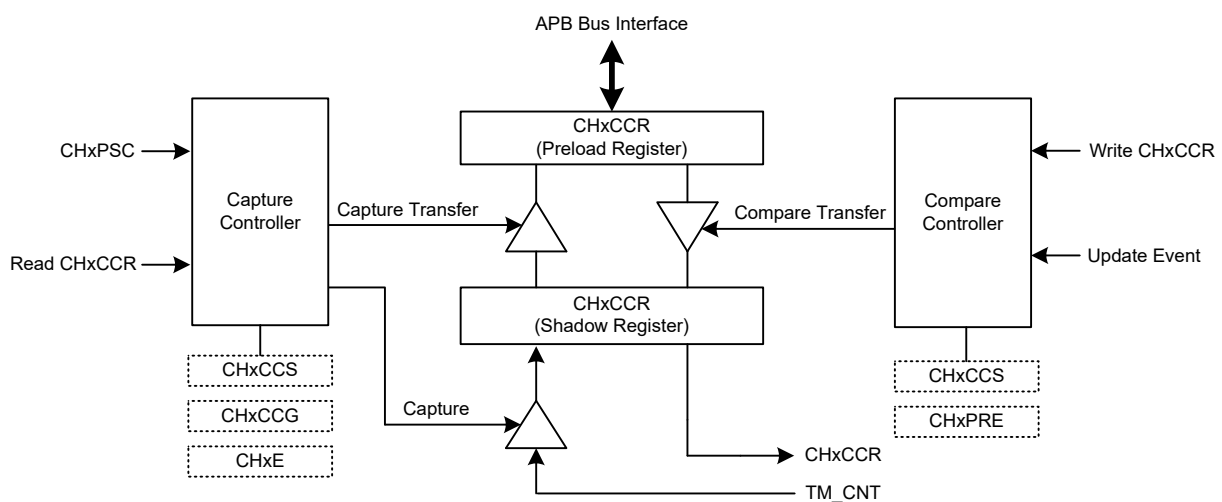


## Channel Controller

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.

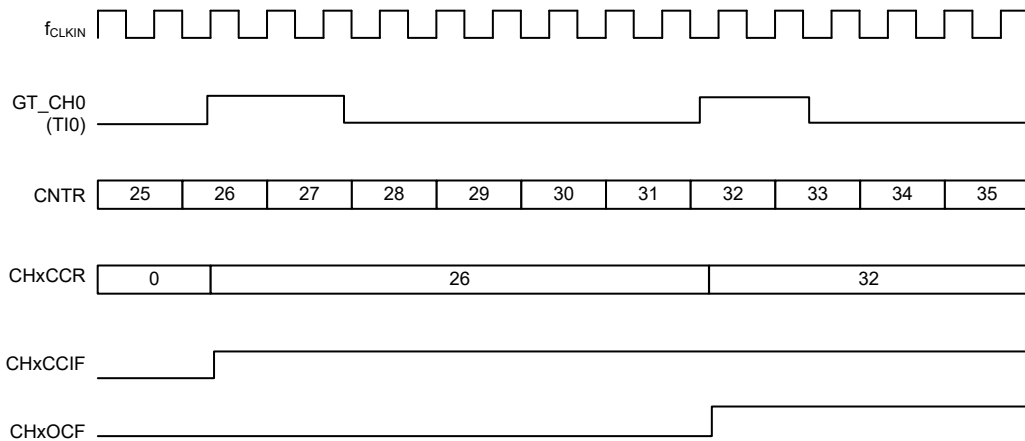
When used in the compare match output mode, the contents of the CHxCCR preload register is copied into the associated shadow register, the counter value is then compared with the register value.



**Figure 51. 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.



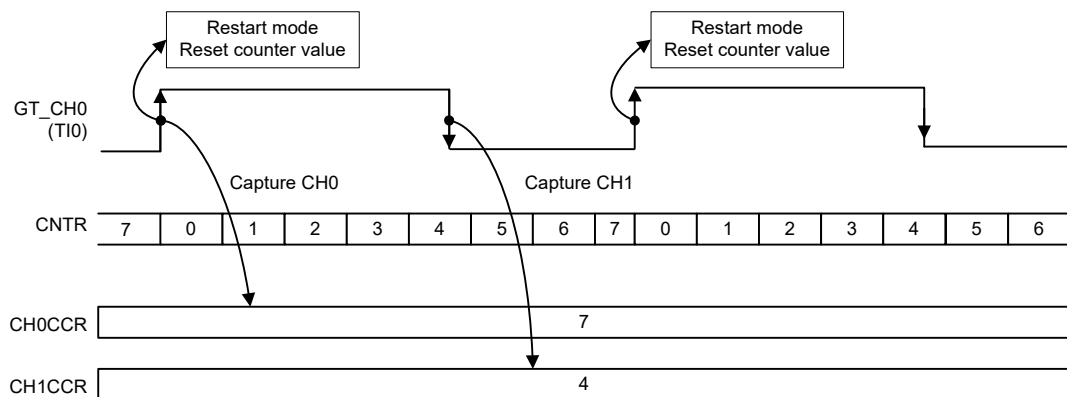
**Figure 52. Input Capture Mode**

### 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  $SMSEL$  field in the  $MDCFR$  register to  $0x4$ .
- Enable the input capture mode by setting the  $CH0E$  and  $CH1E$  bits in the  $CHCTR$  register to 1.

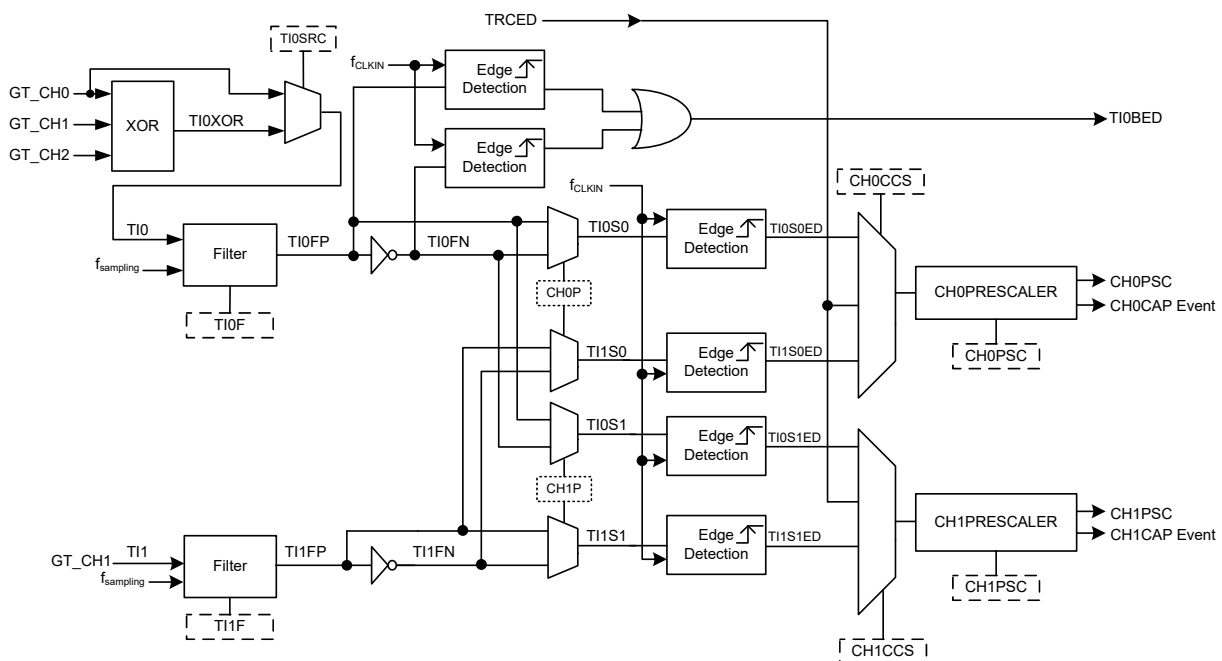
As the following diagram shows, the high pulse width on the  $GT\_CH0$  pin will be captured into the  $CH1CCR$  register while the input period will be captured into the  $CH0CCR$  register after input capture operation.



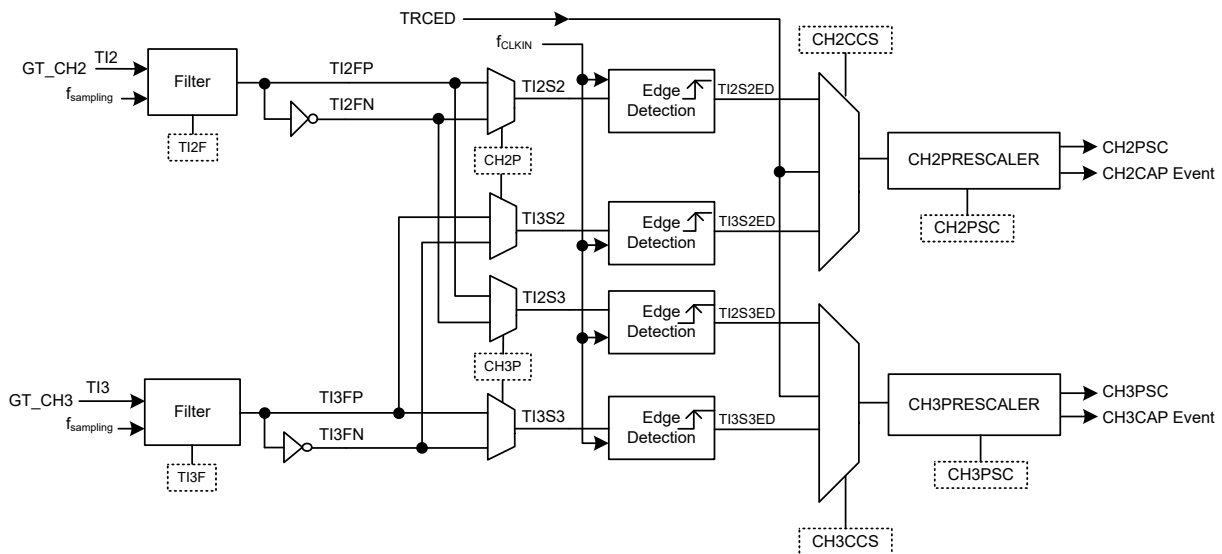
**Figure 53. PWM Pulse Width Measurement Example**

## Input Stage

The input stage consists of a digital filter, a channel polarity selection, edge detection and a channel input 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 54. Channel 0 and Channel 1 Input Stages**

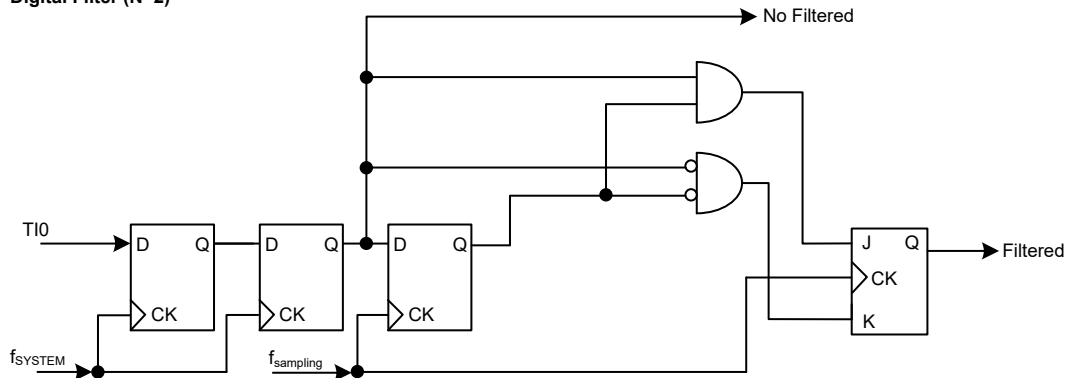


**Figure 55. Channel 2 and Channel 3 Input Stages**

### 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 TIxF field in the CHxICFR register.

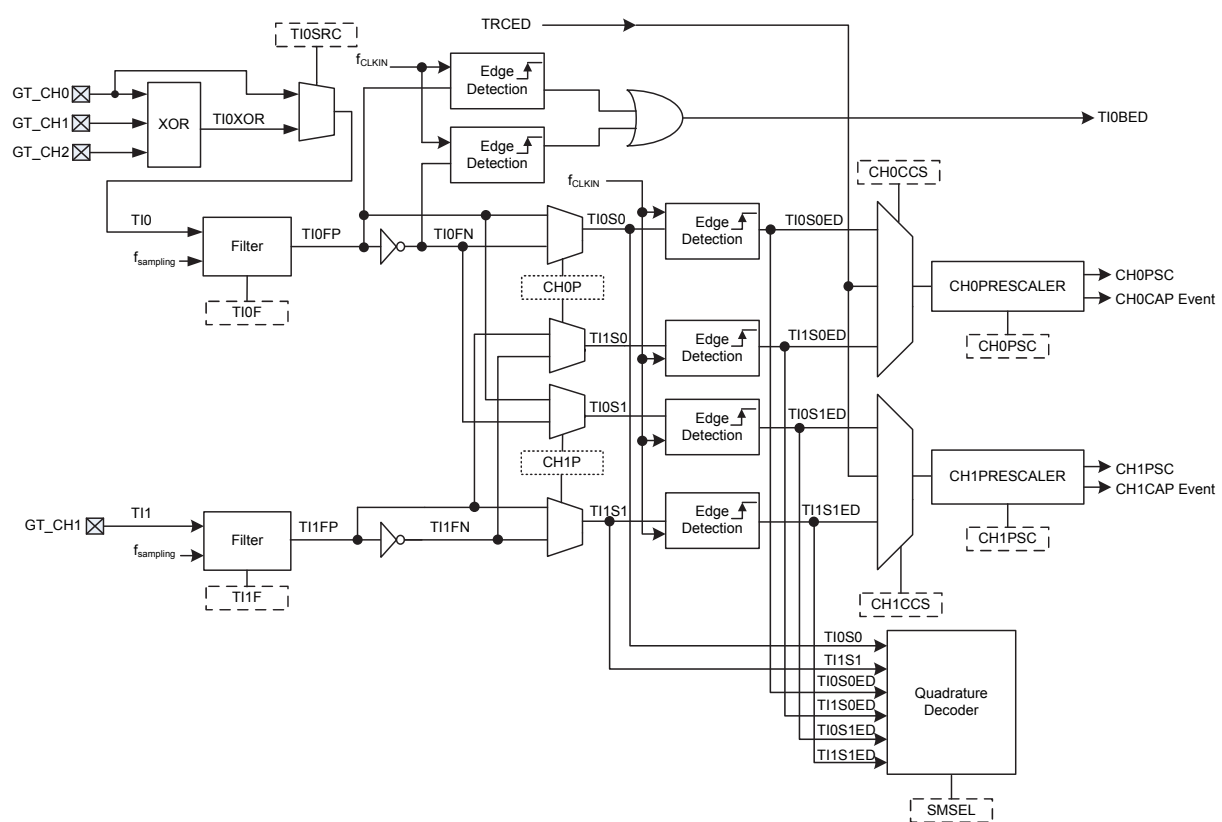
#### Digital Filter (N=2)



**Figure 56. 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.

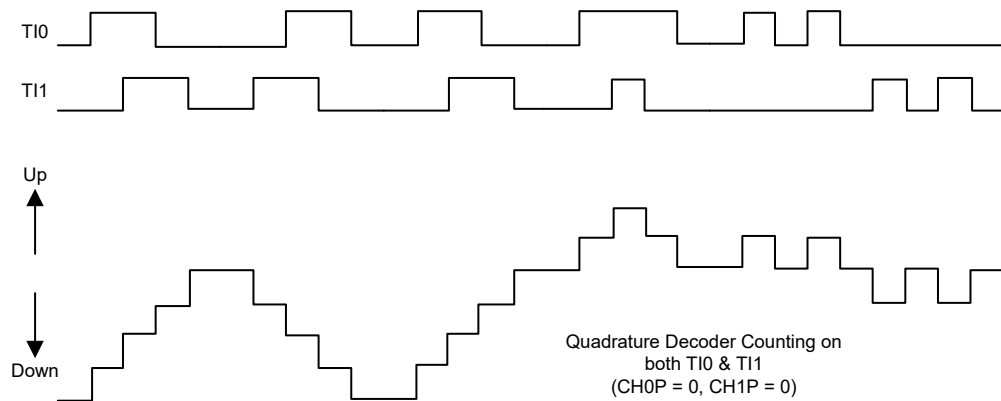


**Figure 57. Input Stage and Quadrature Decoder Block Diagram**

**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

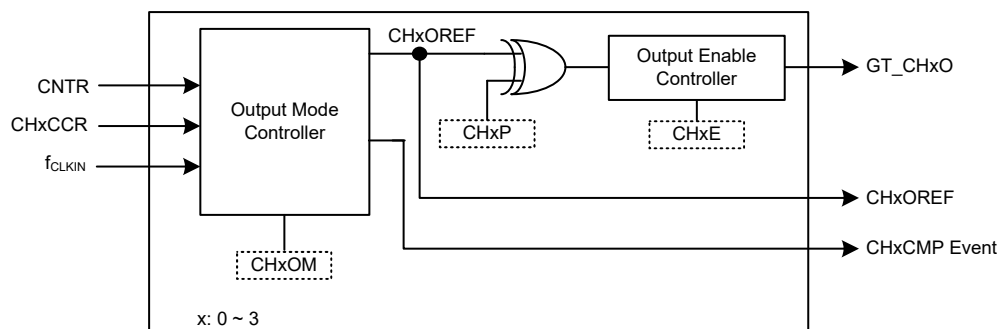
**Note:** “—” → means “no counting”; “X” → impossible



**Figure 58. 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.



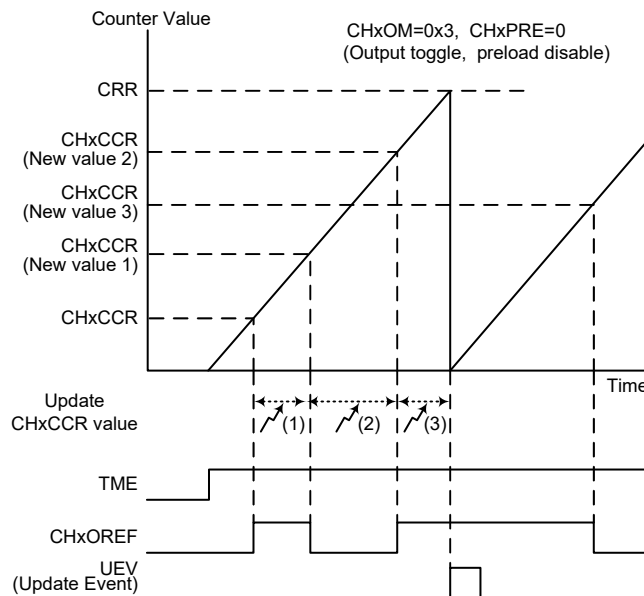
**Figure 59. Output Stage Block Diagram**

### Channel Output Reference Signal

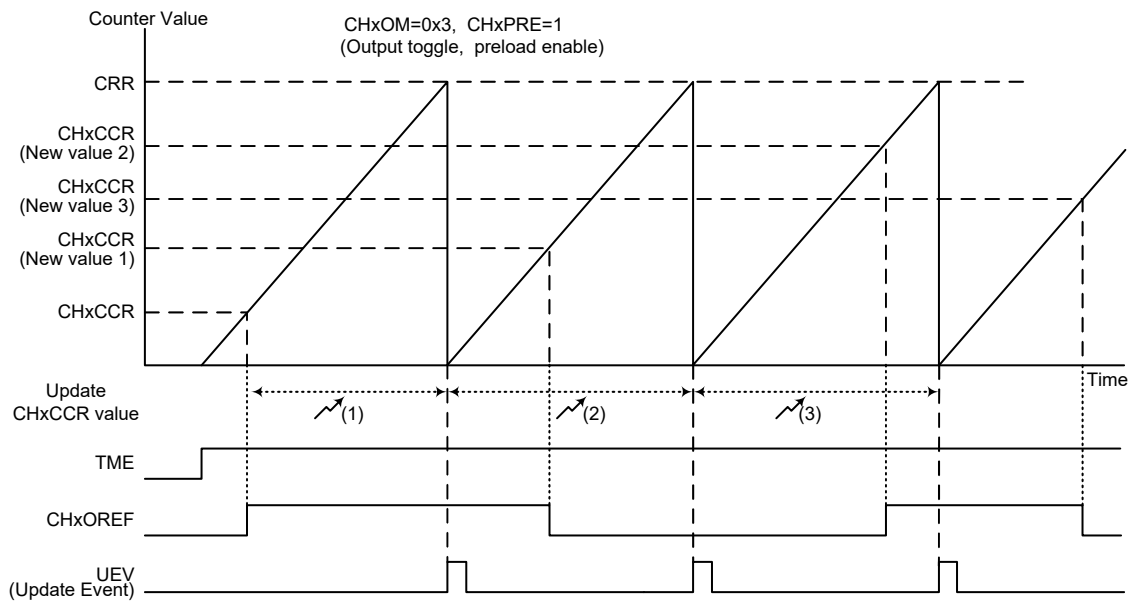
When the GPTM is used in the compare match output mode, the Channel x Output Reference signal, CHxOREF, is defined by the CHxOM field setup. The CHxOREF signal has several types of output function which defines what happens to the output when the counter value matches the contents of the CHxCCR register.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

**Table 32. Compare Match Output Setup**

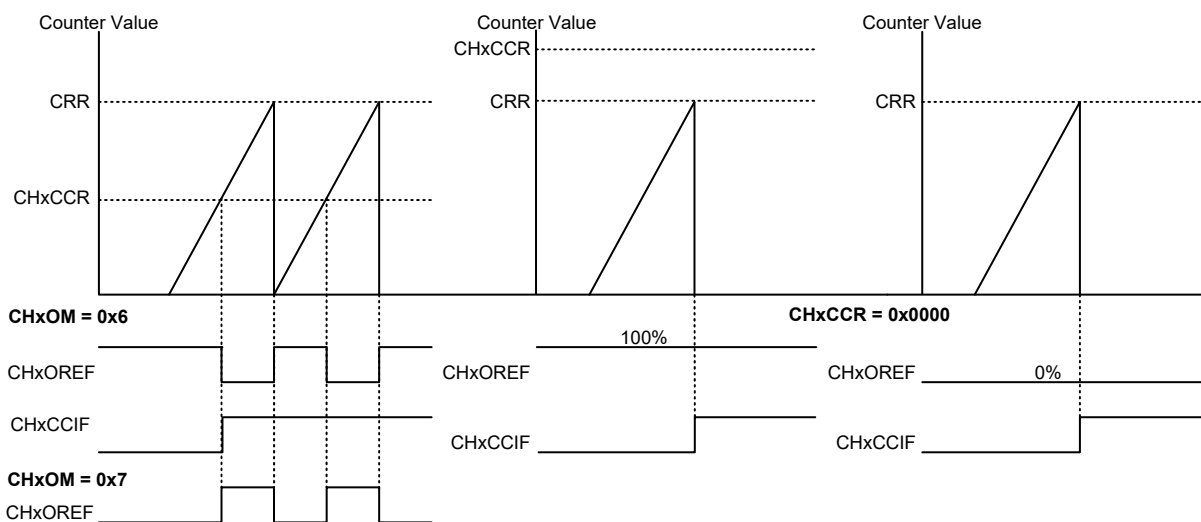
CHxOM Value	Compare Match Level
0x0	No change
0x1	Clear Output to 0
0x2	Set Output to 1
0x3	Toggle Output
0x4	Force Inactive Level
0x5	Force Active Level
0x6	PWM Mode 1
0x7	PWM Mode 2



**Figure 60. Toggle Mode Channel Output Reference Signal (CHxPRE = 0)**



**Figure 61. Toggle Mode Channel Output Reference Signal (CHxPRE = 1)**



**Figure 62. PWM Mode Channel Output Reference Signal and Counter in Up-counting Mode**



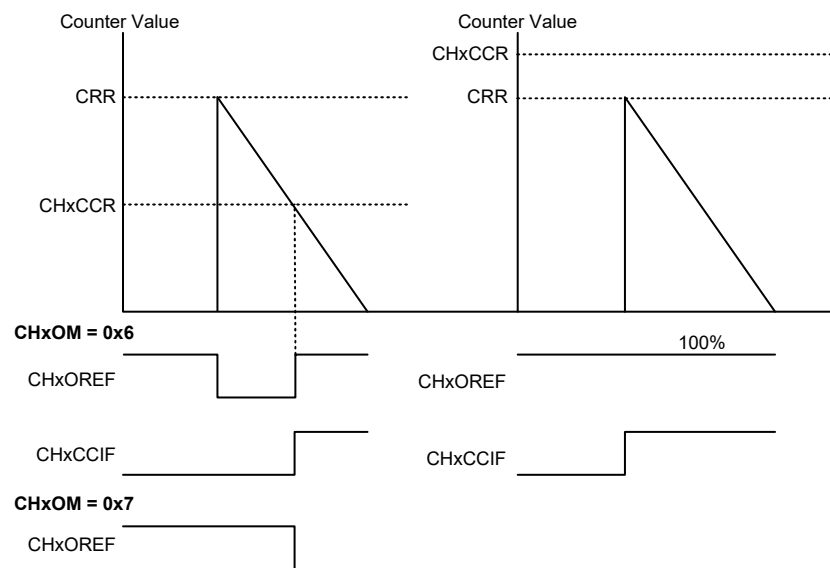


Figure 63. PWM Mode Channel Output Reference Signal and Counter in Down-counting Mode

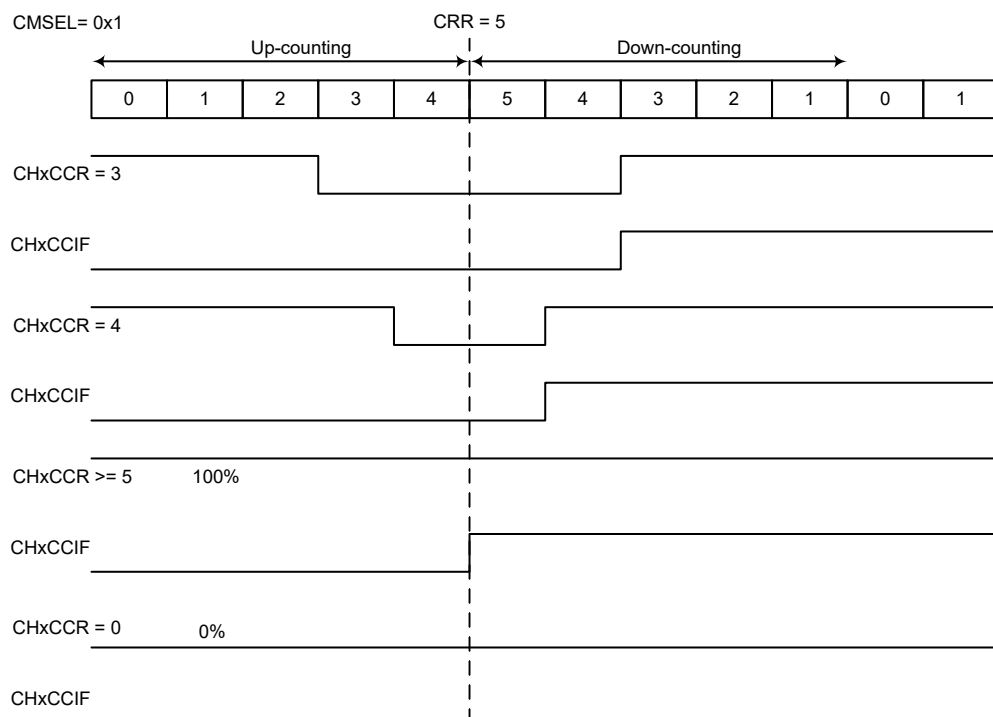


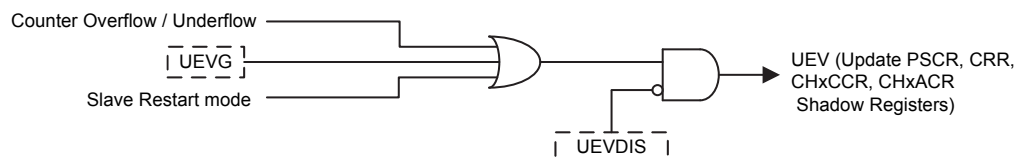
Figure 64. PWM Mode Channel Output Reference Signal and Counter in Centre-aligned Mode

## Update Management

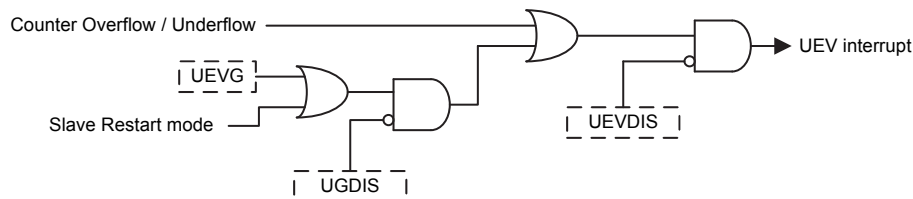
The Update event is used to update the CRR, the PSCR, the CHxACR and the CHxCCR values from the actual registers to the corresponding shadow registers. An update event occurs when the counter overflows or underflows, the software update control bit is triggered or an update event from the slave controller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

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

### Update Event Management



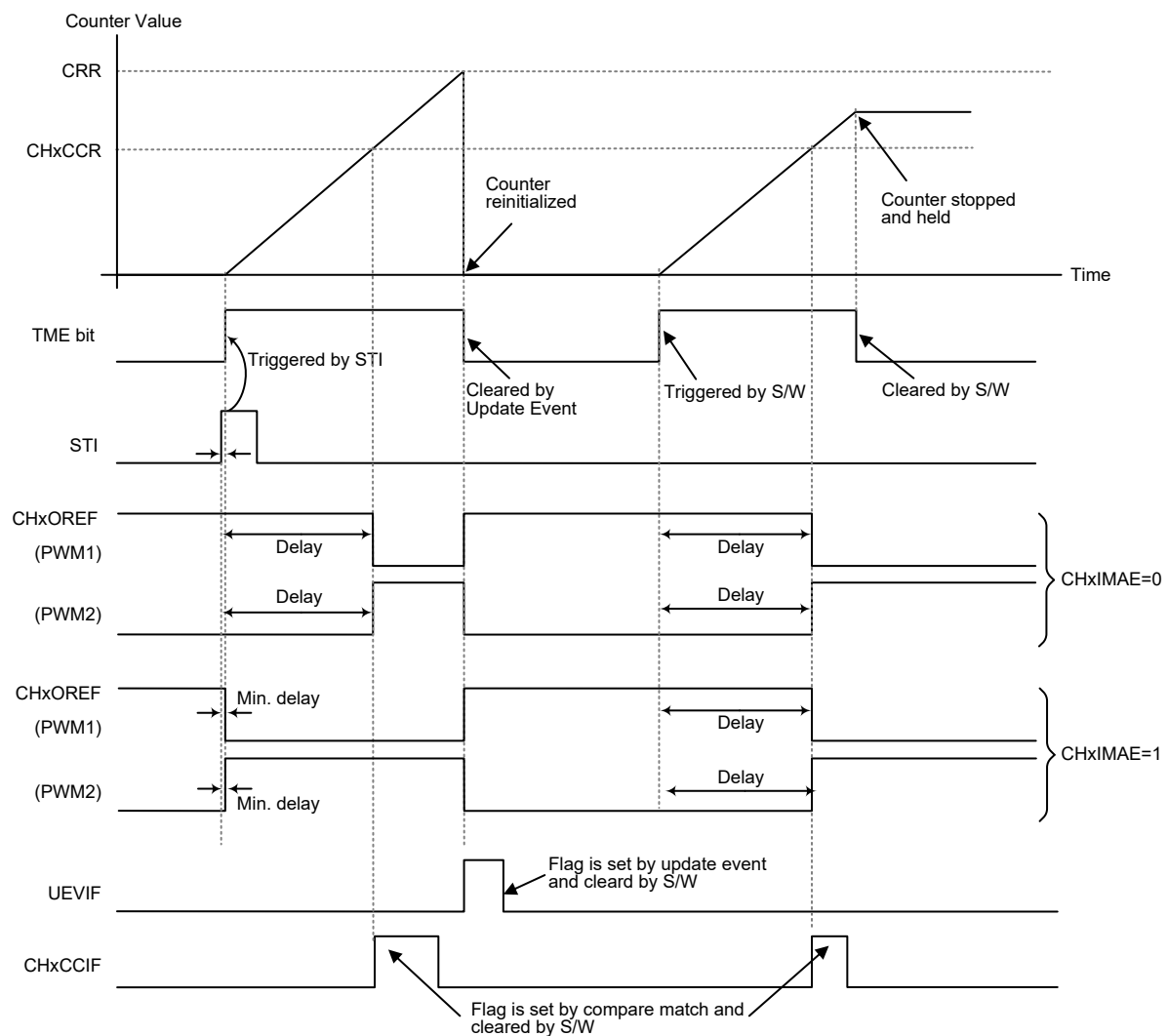
### Update Event Interrupt Management



**Figure 65. Update Event Setting Diagram**

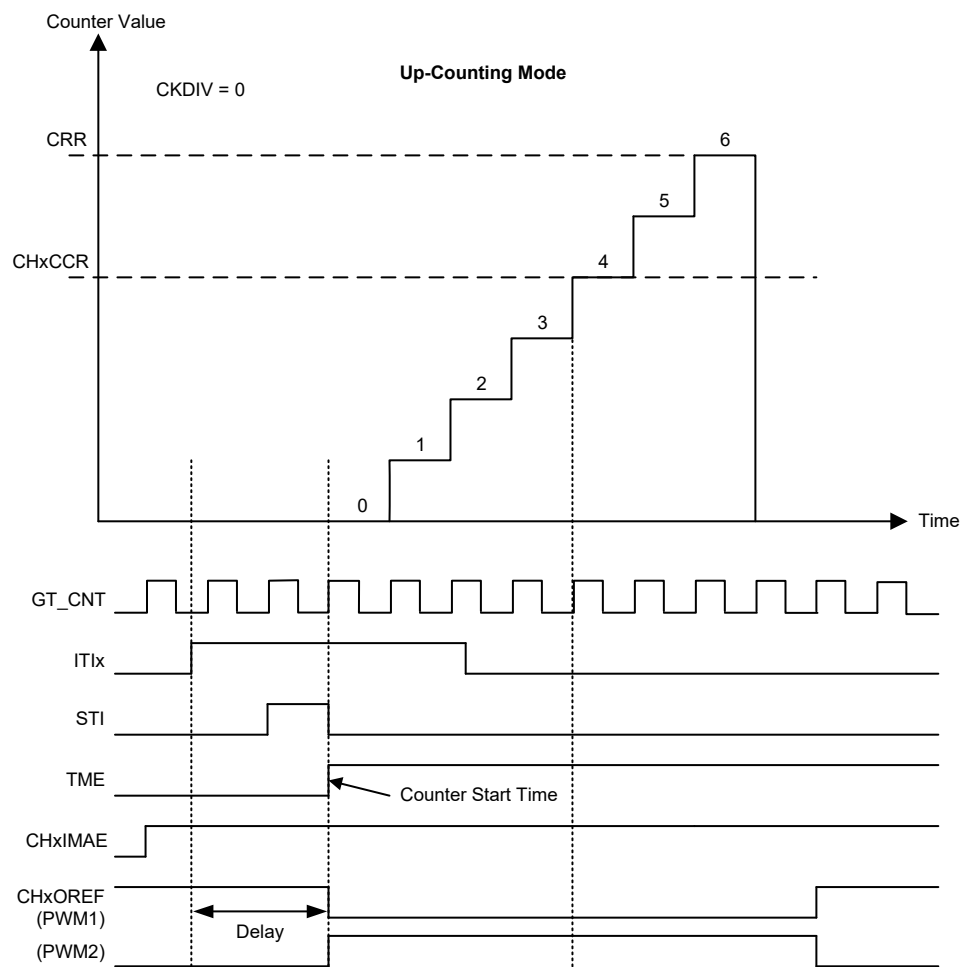
## Single Pulse Mode

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 66. 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 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 67. Immediate Active Mode 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 following figure 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.

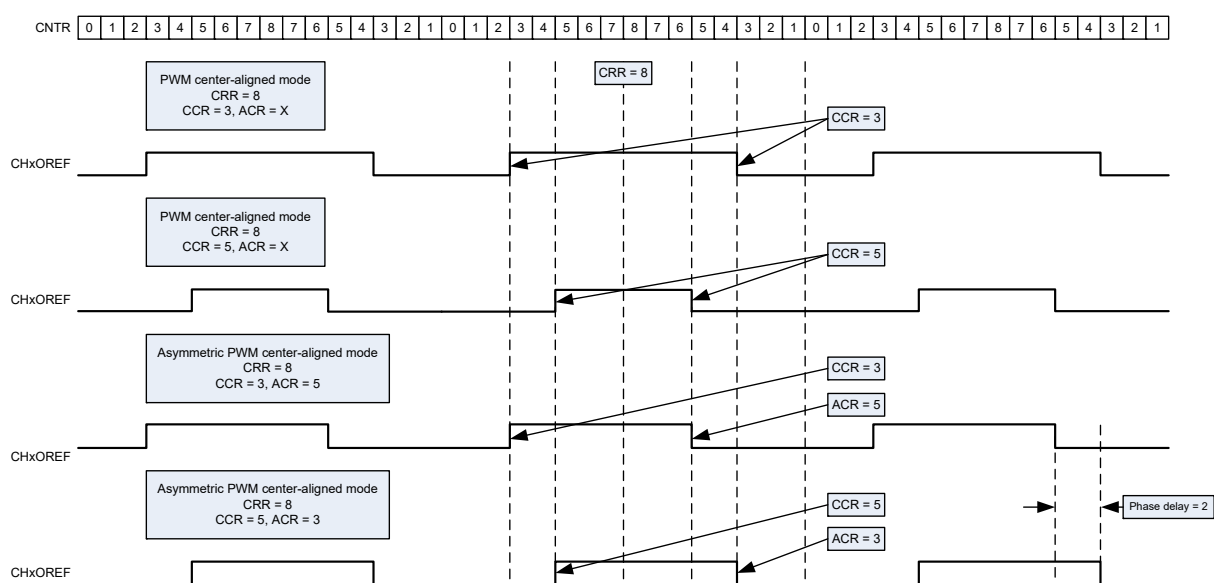


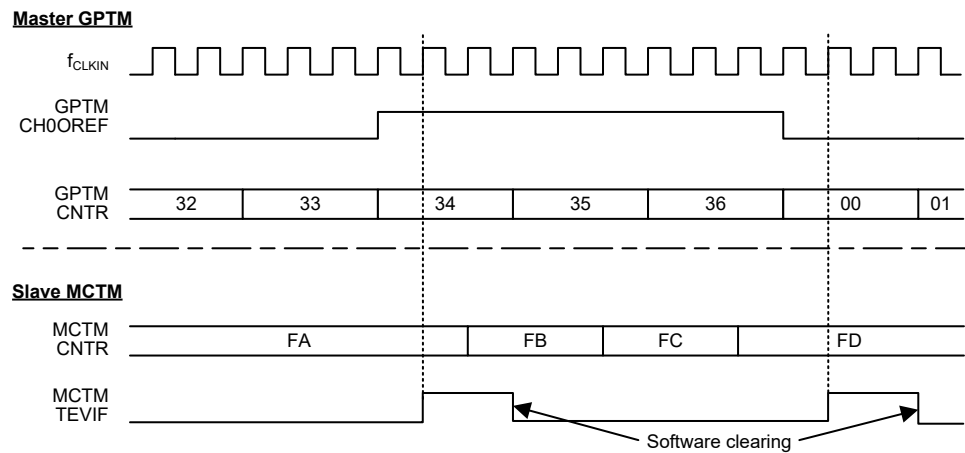
Figure 68. Asymmetric PWM Mode versus Center-Aligned Counting Mode

## 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

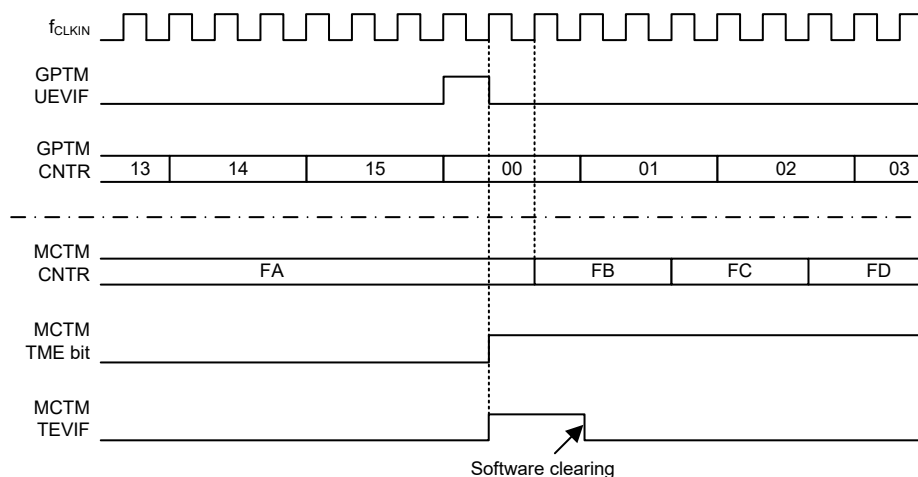
- Configure GPTM as the master mode to send its channel 0 Output Reference signal CH0OREF as a trigger output (MMSEL = 0x4).
- Configure GPTM CH0OREF waveform.
- Configure MCTM to receive its input trigger source from the GPTM trigger output (TRSEL = 0xA).
- Configure MCTM to operate in the pause mode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



**Figure 69. Pausing MCTM Using the GPTM CH0OREF Signal**

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

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

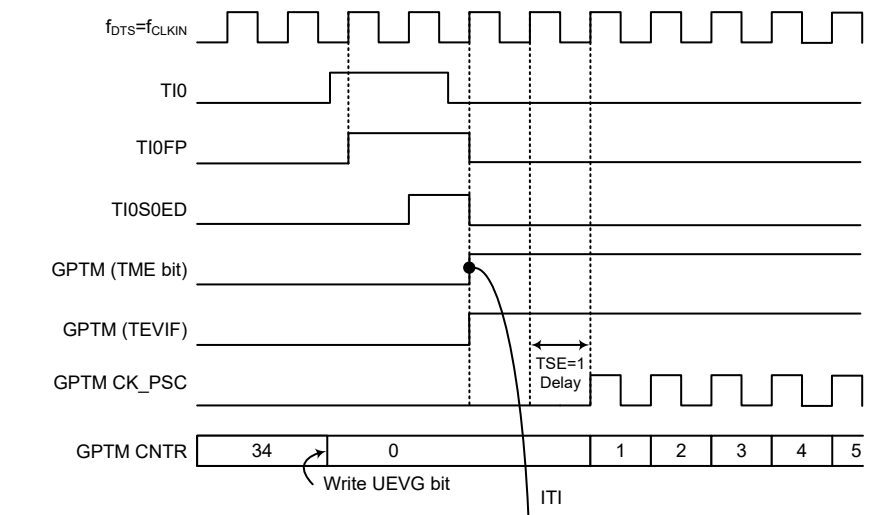


**Figure 70. Triggering MCTM with GPTM Update Event**

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

- Configure GPTM to operate in the master mode to send its enable signal as a trigger output (MMSEL = 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 MCTM to receive its input trigger source from the GPTM trigger output (TRSEL = 0xA).
- Configure MCTM to be in the slave trigger mode (SMSEL = 0x6).

#### Master GPTM



#### Slave MCTM

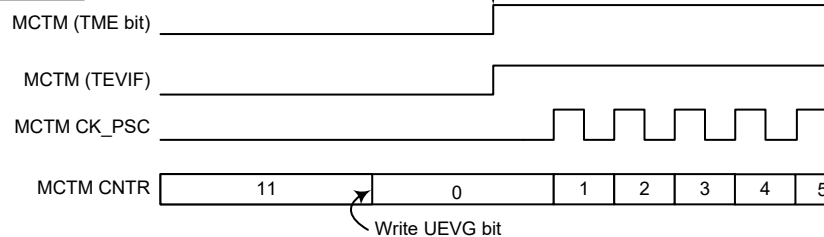


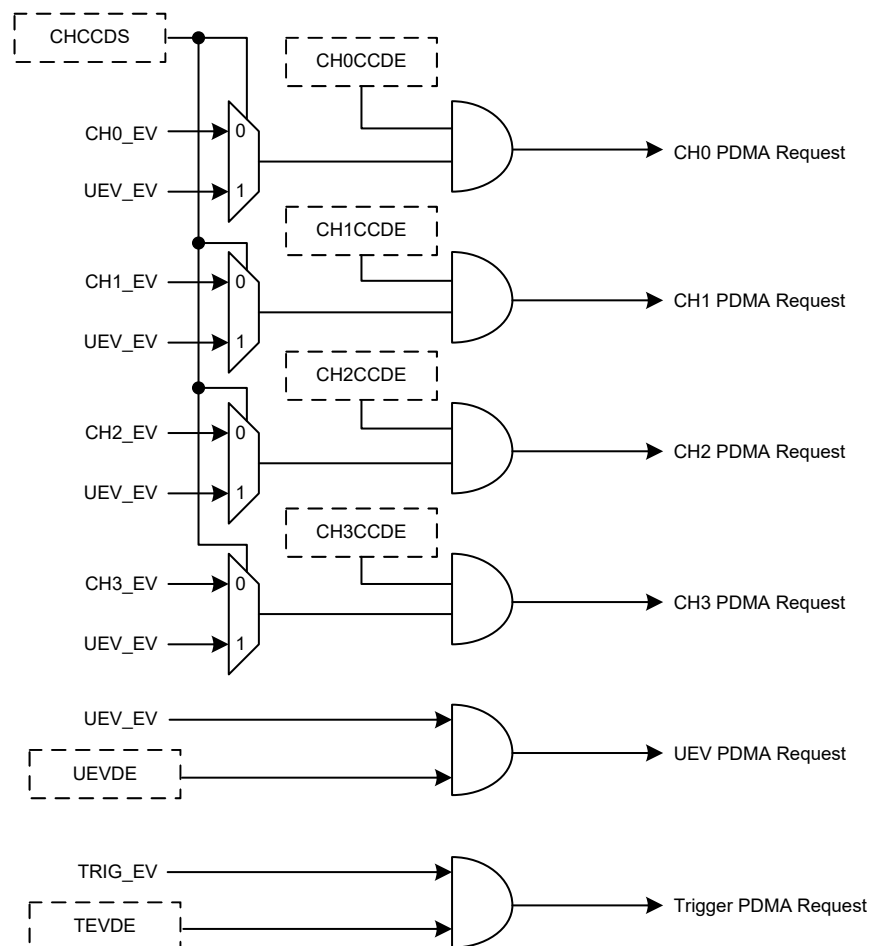
Figure 71. Trigger GPTM and MCTM with the GPTM CH0 Input

## 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 (HT32F54243/HT32F54253 only)

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 72. GPTM PDMA Mapping Diagram**



## Register Map

The following table shows the GPTM registers and reset values. The PDMA related describes are only available for the HT32F54243/HT32F54253 devices.

**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
Type/Reset	Reserved							DIR
								RW 0
	23	22	21	20	19	18	17	16
Type/Reset	Reserved						CMSEL	
							RW 0	RW 0
	15	14	13	12	11	10	9	8
Type/Reset	Reserved						CKDIV	
							RW 0	RW 0
	7	6	5	4	3	2	1	0
Type/Reset	Reserved						UGDIS	UEVDIS
							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	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>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
Type/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 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	—	MCTM	—

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

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

## Channel 0 Input Configuration Register – CH0ICFR

This register specifies the channel 0 input mode configuration.

Offset: 0x020

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
	TI0SRC	Reserved						
Type/Reset	RW	0						
	23	22	21	20	19	18	17	16
	Reserved				CH0PSC		CH0CCS	
Type/Reset					RW	0	RW	0
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved				TI0F			
Type/Reset					RW	0	RW	0

Bits	Field	Descriptions
[31]	TI0SRC	Channel 0 Input Source TI0 Selection 0: The GT_CH0 pin is connected to channel 0 input TI0 1: The XOR operation output of the GT_CH0, GT_CH1, and GT_CH2 pins are connected to the channel 0 input TI0
[19:18]	CH0PSC	Channel 0 Capture Input Source Prescaler Setting These bits define the effective events of the channel 0 capture input. Note that the prescaler is reset once the Channel 0 Capture/Compare Enable bit, CH0E, in the Channel Control register named CHCTR is cleared to 0. 00: No prescaler, channel 0 capture input signal is chosen for each active event 01: Channel 0 Capture input signal is chosen for every 2 events 10: Channel 0 Capture input signal is chosen for every 4 events 11: Channel 0 Capture input signal is chosen for every 8 events
[17:16]	CH0CCS	Channel 0 Capture/Compare Selection 00: Channel 0 is configured as an output 01: Channel 0 is configured as an input derived from the TI0 signal 10: Channel 0 is configured as an input derived from the TI1 signal 11: Channel 0 is configured as an input which comes from the TRCED signal derived from the Trigger Controller Note: The CH0CCS field can be accessed only when the CH0E bit is cleared to 0.



Bits	Field	Descriptions
[3:0]	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				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]	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</p> <p>01: Channel 1 is configured as an input derived from the TI1 signal</p> <p>10: Channel 1 is configured as an input derived from the TI0 signal</p> <p>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_{\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 2 Input Configuration Register – CH2ICFR

This register specifies the channel 2 input mode configuration.

Offset: 0x028

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved				CH2PSC		CH2CCS	
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved				TI2F			
					RW	0	RW	0
							RW	0
								RW
								0

Bits	Field	Descriptions
[19:18]	CH2PSC	<p>Channel 2 Capture Input Source Prescaler Setting</p> <p>These bits define the effective events of the channel 2 capture input. Note that the prescaler is reset once the Channel 2 Capture/Compare Enable bit, CH2E, in the Channel Control register named CHCTR is cleared to 0.</p> <p>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</p>
[17:16]	CH2CCS	<p>Channel 2 Capture/Compare Selection</p> <p>00: Channel 2 is configured as an output            01: Channel 2 is configured as an input derived from the TI2 signal            10: Channel 2 is configured as an input derived from the TI3 signal            11: Channel 2 is configured as an input which comes from the TRCED signal derived from the Trigger Controller</p> <p>Note: The CH2CCS field can be accessed only when the CH2E bit is cleared to 0.</p>

Bits	Field	Descriptions
[3:0]	TI2F	<p>Channel 2 Input Source TI2 Filter Setting</p> <p>These bits define the frequency divided ratio used to sample the TI2 signal. The Digital filter in the GPTM is an N-event counter where N is defined as how many valid transitions are necessary to output a filtered signal.</p> <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			
					RW	0	RW	0

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

Bits	Field	Descriptions
[17:16]	CH3CCS	<p>Channel 3 Capture/Compare Selection</p> <p>00: Channel 3 is configured as an output 01: Channel 3 is configured as an input derived from the TI3 signal 10: Channel 3 is configured as an input derived from the TI2 signal 11: Channel 3 is configured as an input which comes from the TRCED signal derived from the Trigger Controller</p> <p>Note: The CH3CCS field can be accessed only when the CH3E bit is cleared to 0.</p>
[3:0]	TI3F	<p>Channel 3 Input Source TI3 Filter Setting</p> <p>These bits define the frequency divided ratio used to sample the TI3 signal. The Digital filter in the GPTM is an N-event counter where N is defined as how many valid transitions are necessary to output a filtered signal.</p> <p>0000: No filter, the sampling clock is <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	RW
								0

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

Bits	Field	Descriptions
[8][2:0]	CH0OM[3:0]	<p>Channel 0 Output Mode Setting</p> <p>These bits define the functional types of the output reference signal CH0OREF.</p> <p>0000: No Change</p> <p>0001: Output 0 on compare match</p> <p>0010: Output 1 on compare match</p> <p>0011: Output toggles on compare match</p> <p>0100: Force inactive – CH0OREF is forced to 0</p> <p>0101: Force active – CH0OREF is forced to 1</p> <p>0110: PWM mode 1</p> <ul style="list-style-type: none"> <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

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

Bits	Field	Descriptions
[8][2:0]	CH2OM[3:0]	<p>Channel 2 Output Mode Setting</p> <p>These bits define the functional types of the output reference signal CH2OREF.</p> <p>0000: No Change</p> <p>0001: Output 0 on compare match</p> <p>0010: Output 1 on compare match</p> <p>0011: Output toggles on compare match</p> <p>0100: Force inactive – CH2OREF is forced to 0</p> <p>0101: Force active – CH2OREF is forced to 1</p> <p>0110: PWM mode 1</p> <ul style="list-style-type: none"> <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

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

Bits	Field	Descriptions
[8][2:0]	CH3OM[3:0]	<p>Channel 3 Output Mode Setting</p> <p>These bits define the functional types of the output reference signal CH3OREF.</p> <p>0000: No Change</p> <p>0001: Output 0 on compare match</p> <p>0010: Output 1 on compare match</p> <p>0011: Output toggles on compare match</p> <p>0100: Force inactive – CH3OREF is forced to 0</p> <p>0101: Force active – CH3OREF is forced to 1</p> <p>0110: PWM mode 1</p> <ul style="list-style-type: none"> <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>
[0]	CH0E	<p>Channel 0 Capture/Compare Enable</p> <ul style="list-style-type: none"> <li>- Channel 0 is configured as an input (CH0CCS = 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 0 is configured as an output (CH0CCS = 0x0) <ul style="list-style-type: none"> <li>0: Off – Channel 0 output signal CH0O is not active</li> <li>1: On – Channel 0 output signal CH0O is generated on the corresponding output pin</li> </ul> </li> </ul>

## Channel Polarity Configuration Register – CHPOLR

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

Offset: 0x054

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/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
[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
[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



Bits	Field	Descriptions
[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		
						WO	0	WO	0	
	7	6	5	4	3	2	1	0		
Type/Reset	Reserved				CH3CCG	CH2CCG	CH1CCG	CH0CCG		
					WO	0	WO	0	WO	0

Bits	Field	Descriptions
[10]	TEVG	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.

Bits	Field	Descriptions
[3]	CH3CCG	<p>Channel 3 Capture/Compare Generation</p> <p>A Channel 3 capture/compare event can be generated by setting this bit. It is cleared by hardware automatically.</p> <p>0: No action 1: Capture/compare event is generated on channel 3</p> <p>If Channel 3 is configured as an input, the counter value is captured into the CH3CCR register and then the CH3CCIF bit is set. If Channel 3 is configured as an output, the CH3CCIF bit is set.</p>
[2]	CH2CCG	<p>Channel 2 Capture/Compare Generation</p> <p>A Channel 2 capture/compare event can be generated by setting this bit. It is cleared by hardware automatically.</p> <p>0: No action 1: Capture/compare event is generated on channel 2</p> <p>If Channel 2 is configured as an input, the counter value is captured into the CH2CCR register and then the CH2CCIF bit is set. If Channel 2 is configured as an output, the CH2CCIF bit is set.</p>
[1]	CH1CCG	<p>Channel 1 Capture/Compare Generation</p> <p>A Channel 1 capture/compare event can be generated by setting this bit. It is cleared by hardware automatically.</p> <p>0: No action 1: Capture/compare event is generated on channel 1</p> <p>If Channel 1 is configured as an input, the counter value is captured into the CH1CCR register and then the CH1CCIF bit is set. If Channel 1 is configured as an output, the CH1CCIF bit is set.</p>
[0]	CH0CCG	<p>Channel 0 Capture/Compare Generation</p> <p>A Channel 0 capture/compare event can be generated by setting this bit. It is cleared by hardware automatically.</p> <p>0: No action 1: Capture/compare event is generated on channel 0</p> <p>If Channel 0 is configured as an input, the counter value is captured into the CH0CCR register and then the CH0CCIF bit is set. If Channel 0 is configured as an output, the CH0CCIF bit is set.</p>

## Timer Interrupt Status Register – INTSR

This register stores the timer interrupt status.

Offset: 0x07C

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[10]	TEVIF	Trigger Event Interrupt Flag This flag is set by hardware on a trigger event and is cleared by software. 0: No trigger event occurs 1: Trigger event occurs
[8]	UEVIF	Update Event Interrupt Flag This bit is set by hardware on an update event and is cleared by software. 0: No update event occurs 1: Update event occurs Note: The update event is derived from the following conditions: - The counter overflows or underflows - The UEVG bit is asserted - A restart trigger event occurs from the slave trigger input
[7]	CH3OCF	Channel 3 Over-Capture Flag This flag is set by hardware and cleared by software. 0: No over-capture event is detected 1: Capture event occurs again when the CH3CCIF bit is already set and it is not yet cleared by software
[6]	CH2OCF	Channel 2 Over-Capture Flag This flag is set by hardware and cleared by software. 0: No over-capture event is detected 1: Capture event occurs again when the CH2CCIF bit is already set and it is not cleared yet by software
[5]	CH1OCF	Channel 1 Over-Capture Flag This flag is set by hardware and cleared by software. 0: No over-capture event is detected 1: Capture event occurs again when the CH1CCIF bit is already set and it is not cleared yet by software.

Bits	Field	Descriptions
[4]	CH0OCF	<p>Channel 0 Over-Capture Flag</p> <p>This flag is set by hardware and cleared by software.</p> <p>0: No over-capture event is detected</p> <p>1: Capture event occurs again when the CH0CCIF bit is already set and it is not yet cleared by software.</p>
[3]	CH3CCIF	<p>Channel 3 Capture/Compare Interrupt Flag</p> <p>- Channel 3 is configured as an output:</p> <p>0: No match event occurs</p> <p>1: The content of the counter CNTR has matched the content of the CH3CCR register</p> <p>This flag is set by hardware when the counter value matches the CH3CCR value except in the center-aligned mode. It is cleared by software.</p> <p>- Channel 3 is configured as an input:</p> <p>0: No input capture occurs</p> <p>1: Input capture occurs</p> <p>This bit is set by hardware on a capture event. It is cleared by software or by reading the CH3CCR register.</p>
[2]	CH2CCIF	<p>Channel 2 Capture/Compare Interrupt Flag</p> <p>- Channel 2 is configured as an output:</p> <p>0: No match event occurs</p> <p>1: The content of the counter CNTR has matched the content of the CH2CCR register</p> <p>This flag is set by hardware when the counter value matches the CH2CCR value except in the center-aligned mode. It is cleared by software.</p> <p>- Channel 2 is configured as an input:</p> <p>0: No input capture occurs</p> <p>1: Input capture occurs.</p> <p>This bit is set by hardware on a capture event. It is cleared by software or by reading the CH2CCR register.</p>
[1]	CH1CCIF	<p>Channel 1 Capture/Compare Interrupt Flag</p> <p>- Channel 1 is configured as an output:</p> <p>0: No match event occurs</p> <p>1: The content of the counter CNTR has matched the content of the CH1CCR register</p> <p>This flag is set by hardware when the counter value matches the CH1CCR value except in the center-aligned mode. It is cleared by software.</p> <p>- Channel 1 is configured as an input:</p> <p>0: No input capture occurs</p> <p>1: Input capture occurs</p> <p>This bit is set by hardware on a capture event. It is cleared by software or by reading the CH1CCR register.</p>

Bits	Field	Descriptions
[0]	CH0CCIF	<p>Channel 0 Capture/Compare Interrupt Flag</p> <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> </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> </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> <p>This bit is set by hardware on a capture event. It is cleared by software or by reading the CH0CCR register.</p>

### Timer Counter Register – CNTR

This register stores the timer counter value.

Offset: 0x080

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[15:0]	CNTV	Counter Value

## Timer Prescaler Register – PSCR

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

Offset: 0x084

Reset value: 0x0000\_0000

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

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

# 15 Motor Control Timer (MCTM)

## Introduction

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

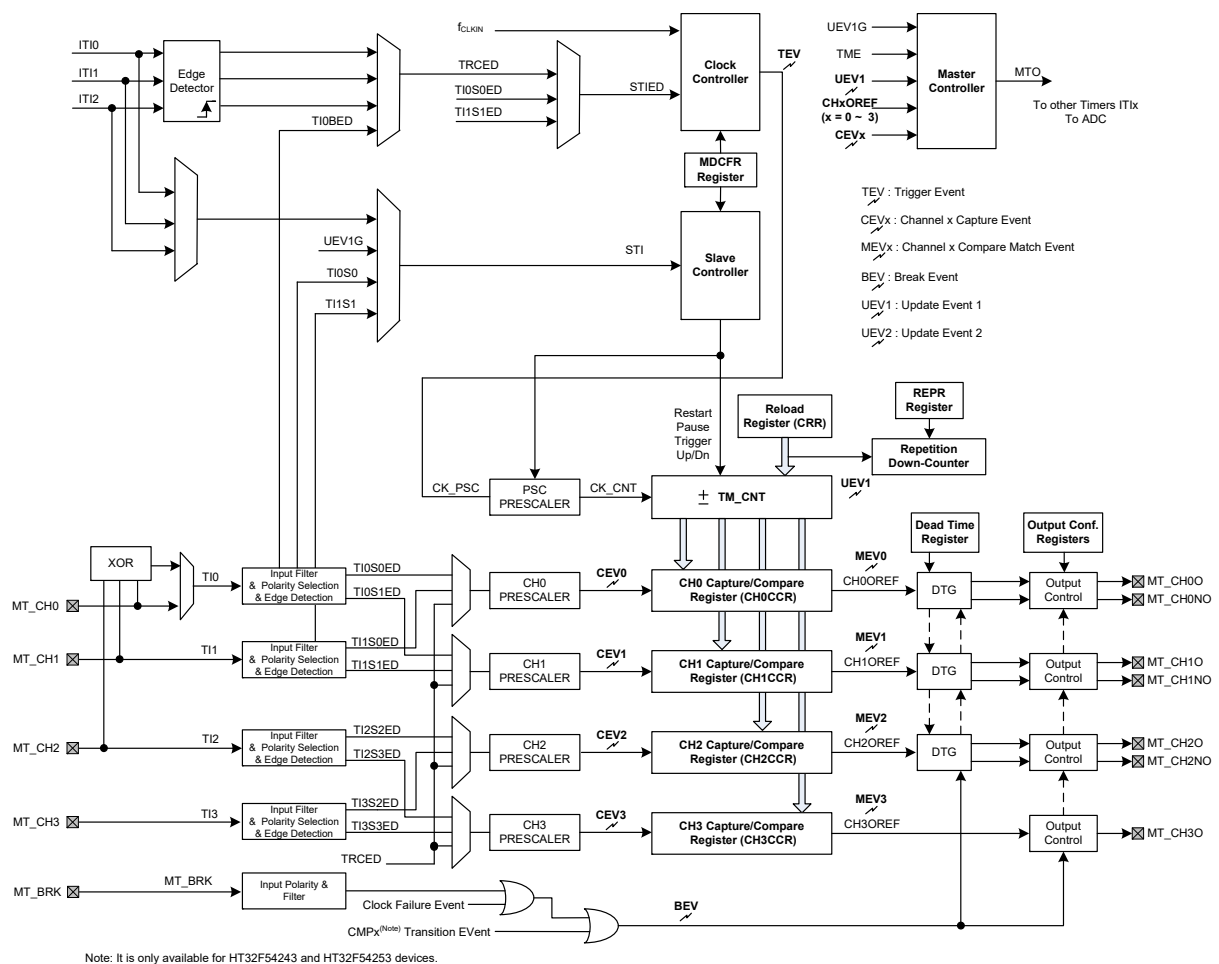


Figure 73. MCTM Block Diagram

## Features

- 16-bit up/down auto-reload counter
- 16-bit programmable prescaler that allows division 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
  - PWM waveform Generation – Edge and Center-aligned Counting Mode
  - Single Pulse Mode Output
- Complementary Outputs with programmable dead-time insertion
- Repetition counter updates timer registers only after a given number of counter cycles
- Synchronization circuit to control the timer with external signals and can interconnect several timers together
- Interrupt/PDMA generation on the following events, PDMA is only available for the HT32F54243/HT32F54253 devices
  - Update event 1
  - Update event 2
  - Trigger event
  - Input capture event
  - Output compare match
  - Break event – only interrupt
- MCTM Master/Slave mode controller
- Supports 3-phase motor control and hall sensor interface
- Break input signals to assert the timer output signals in reset state or in a known state

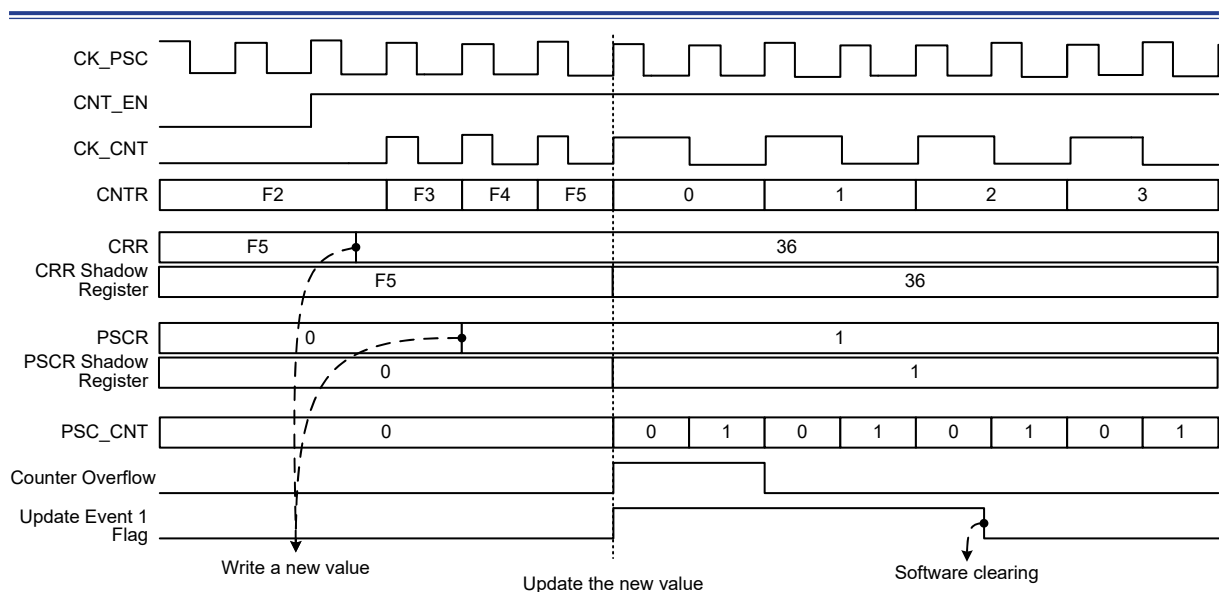
## Functional Descriptions

### Counter Mode

#### Up-Counting

In this mode the counter counts continuously from 0 to the counter-reload value, which is defined in the CRR register, in a count-up direction, 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 cleared to 0 for the up-counting mode.

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



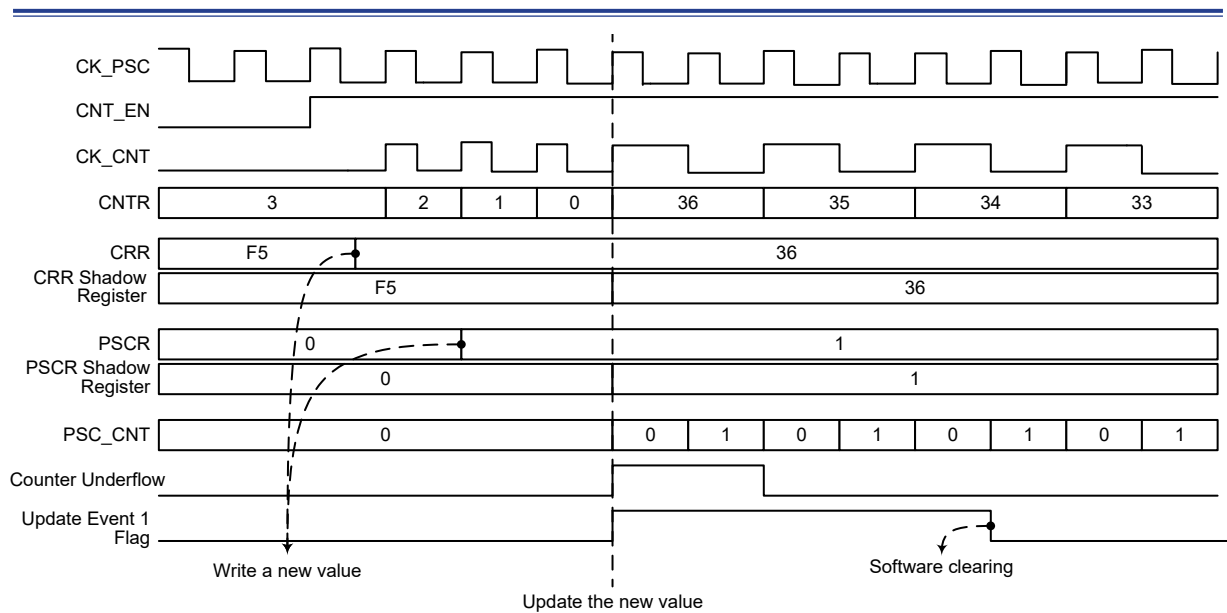
**Figure 74. Up-Counting Example**



### Down-Counting

In this mode the counter counts continuously from the counter-reload value, which is defined in the CRR register, to 0 in a count-down direction, 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 an update event 1 is generated by setting the UEV1G bit in the EVGR register to 1, the counter value will also be initialised to the counter-reload value.



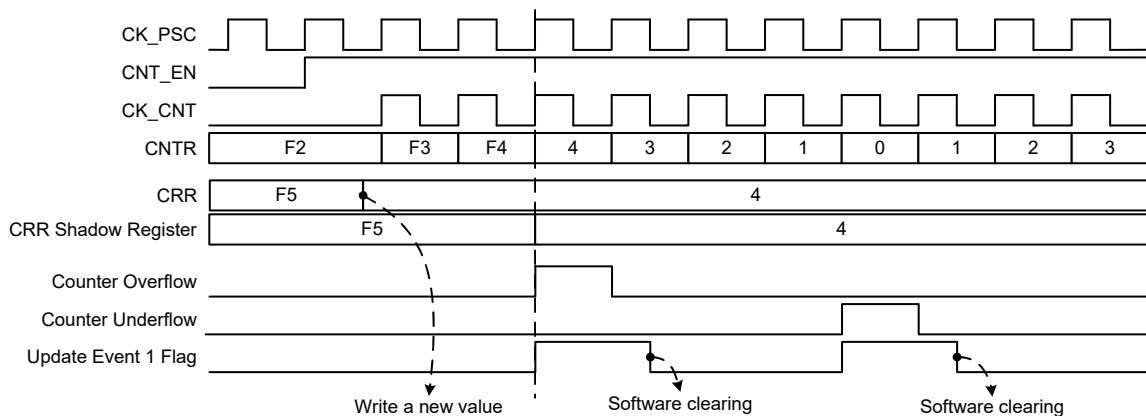
**Figure 75. Down-Counting Example**

### Center-Aligned Counting

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

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

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

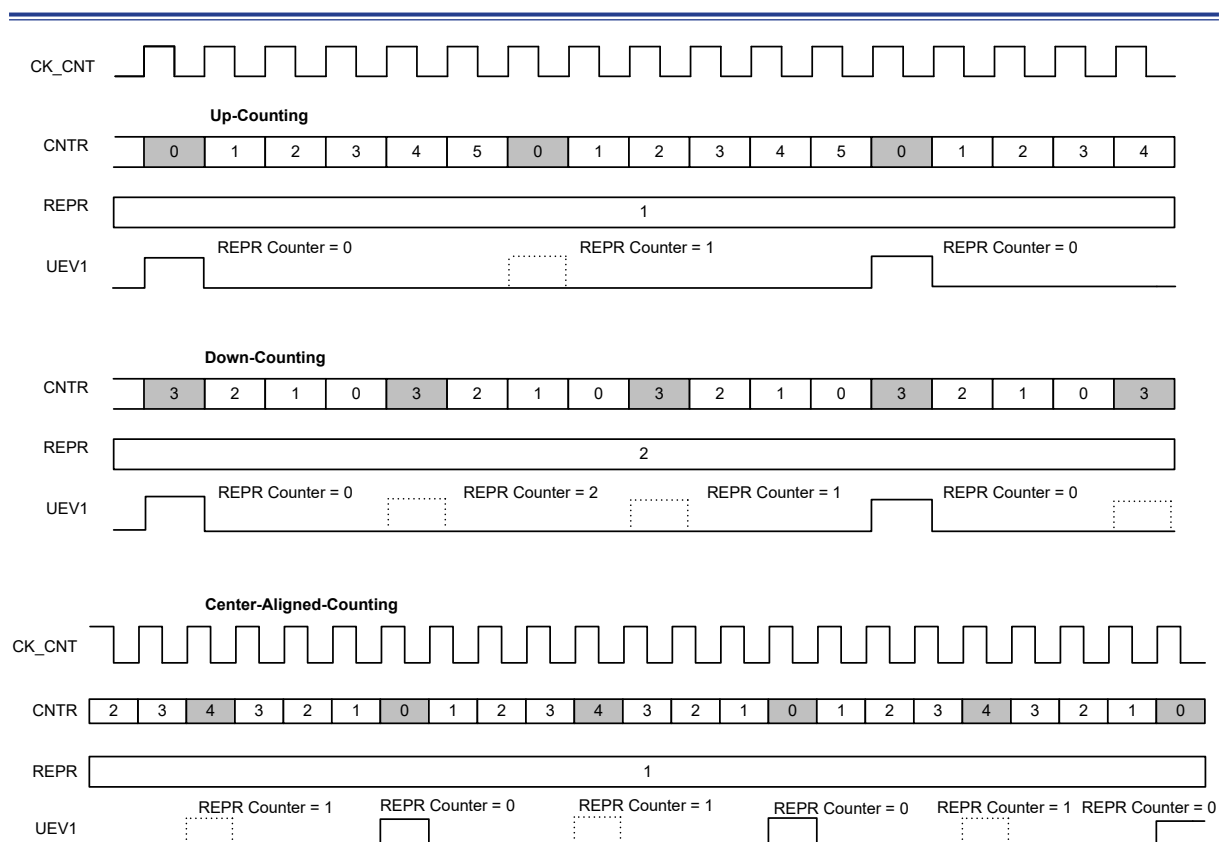


**Figure 76. Center-Aligned Counting Example**

### Repetition Down-Counter Operation

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

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



**Figure 77. Update Event 1 Dependent Repetition Mechanism Example**

## Clock Controller

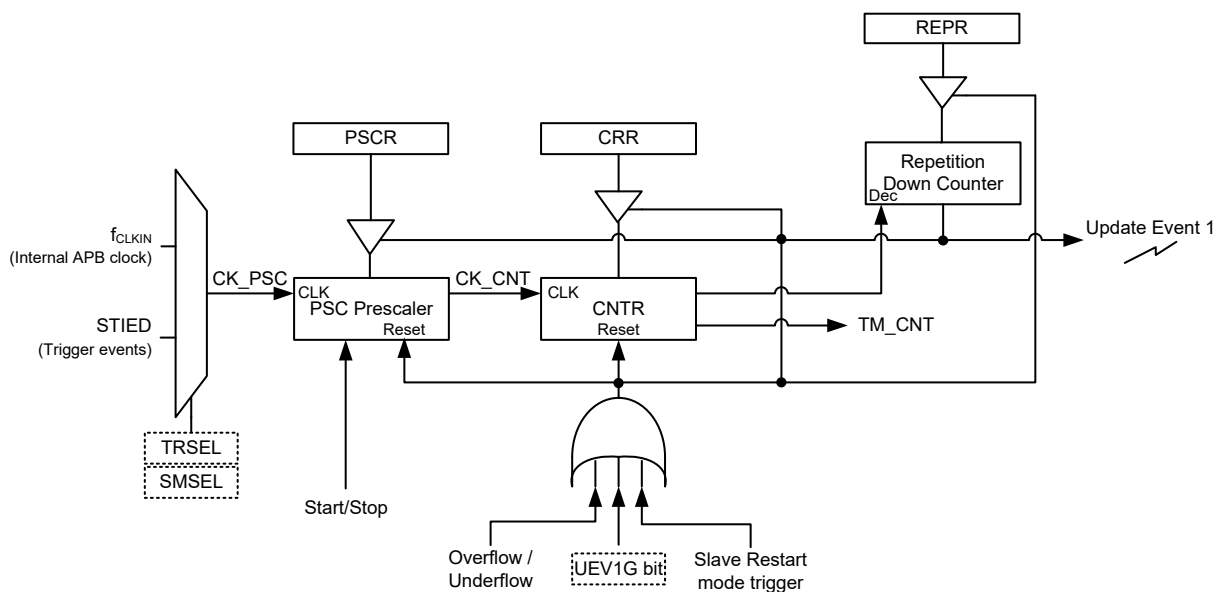
The following describes the Timer Module clock controller which determines the internal prescaler counter clock source.

### ■ Internal APB clock $f_{CLKIN}$

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

### ■ STIED

The counter prescaler can count during each rising edge of the STI signal. This mode can be selected by setting the SMSEL field to 0x7 in the MDCFR register. Here the counter will act as an event counter. The input event, known as STI here, can be selected by setting the TRSEL field to an available value except the value of 0x0. When the STI signal is selected as the clock source, the internal edge detection circuitry will generate a clock pulse during each STI signal rising edge to drive the counter prescaler. It is important to note that if the TRSEL field is set to 0x0 to select the software UEV1G bit as the trigger source, then when the SMSEL field is set to 0x7, the counter will be updated instead of counting.

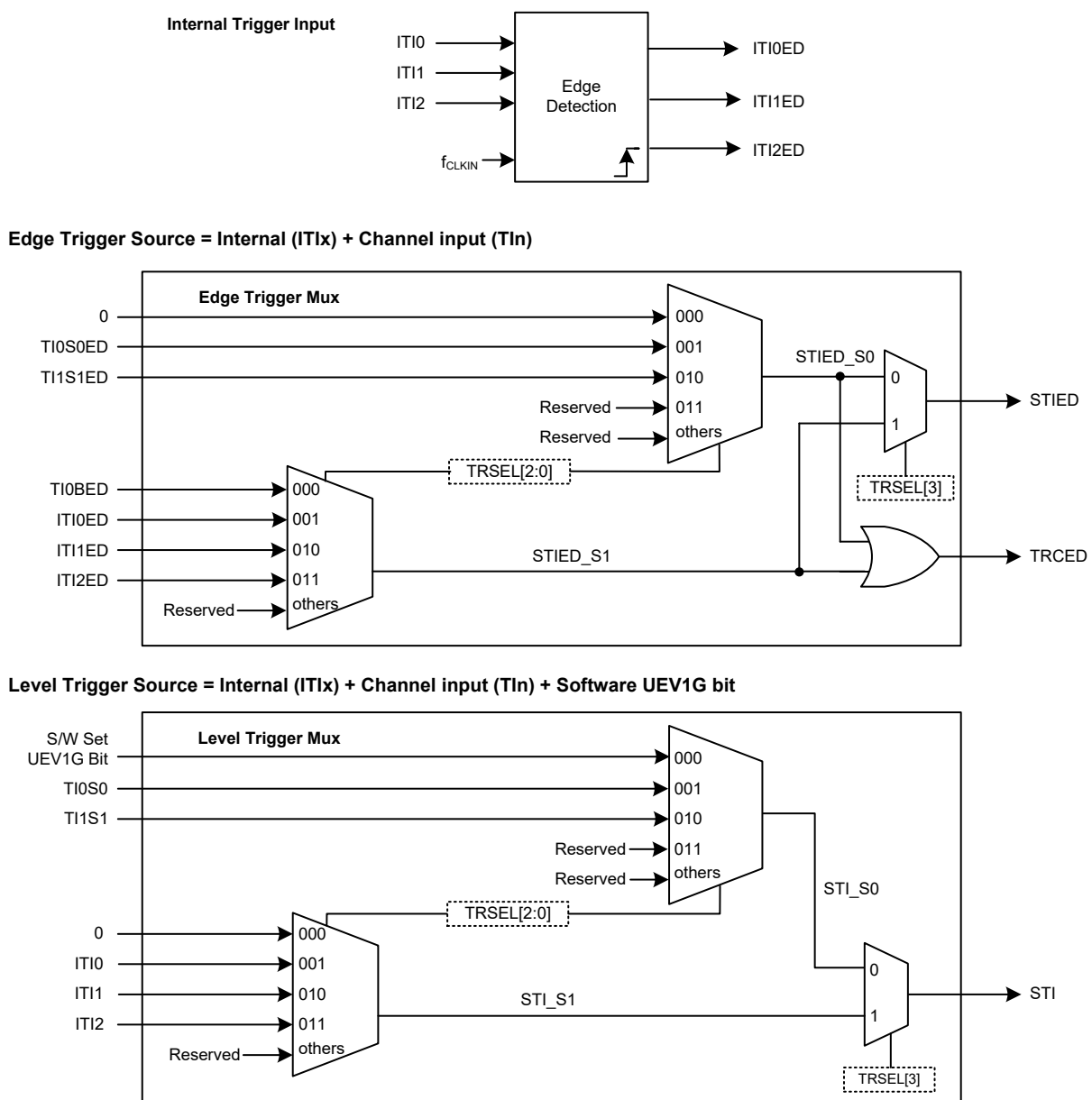


**Figure 78. MCTM Clock Source Selection**

## Trigger Controller

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

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



**Figure 79. Trigger Controller Block**

## Slave Controller

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

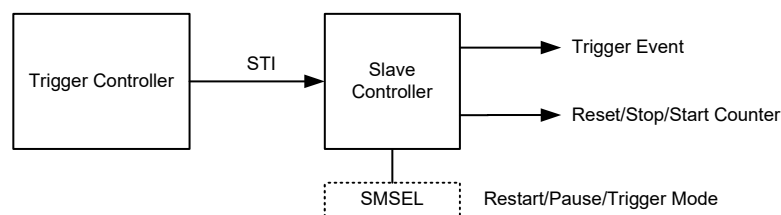


Figure 80. Slave Controller Diagram

### Restart Mode

The counter and its prescaler can be reinitialised in response to an STI signal rising edge. When an STI rising edge occurs, the update event 1 software generation bit named UEV1G will automatically be asserted by hardware and the trigger event flag, TEVIF, will also be set. Then the counter and prescaler will be reinitialized. Although the UEV1G bit is set to 1 by hardware, whether the update event really occurs or not depends upon the update event 1 disable control bit UEV1DIS is set to 1 or not. If the UEV1DIS bit is set to 1 to disable the update event, then no update event will be generated, however the counter and prescaler are still reinitialized when an STI rising edge occurs. If the UEV1DIS bit in the CNTCFR register is cleared to enable the update event, then an update event will be generated together with the STI rising edge and all the preloaded registers will be updated.

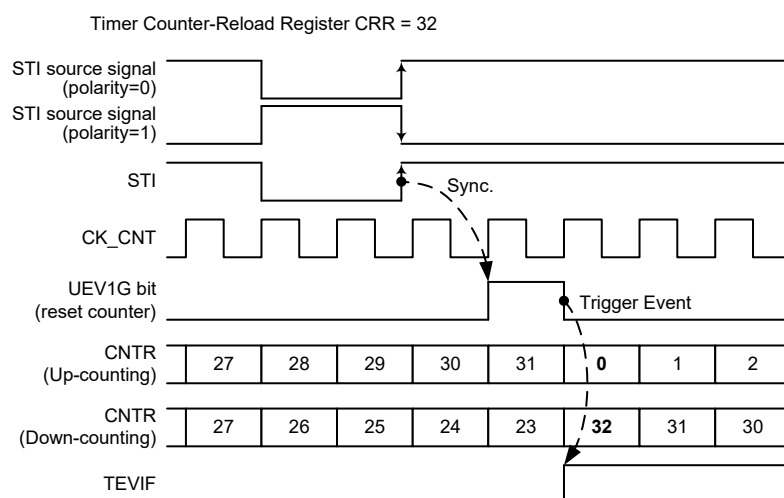


Figure 81. MCTM in Restart Mode

## Pause Mode

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

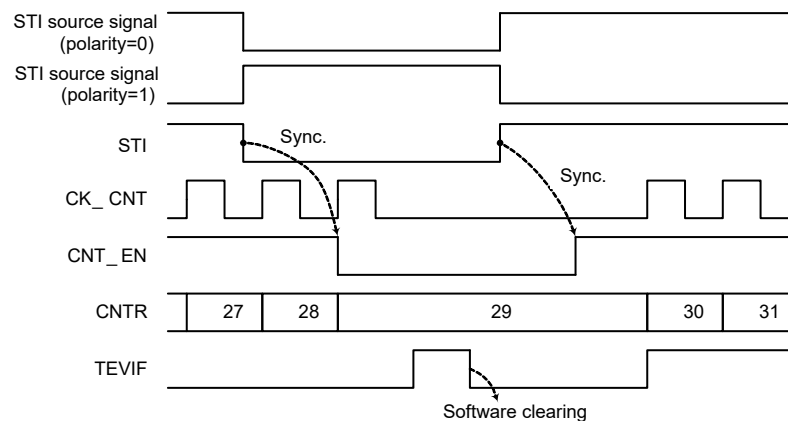


Figure 82. MCTM in Pause Mode

## Trigger Mode

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

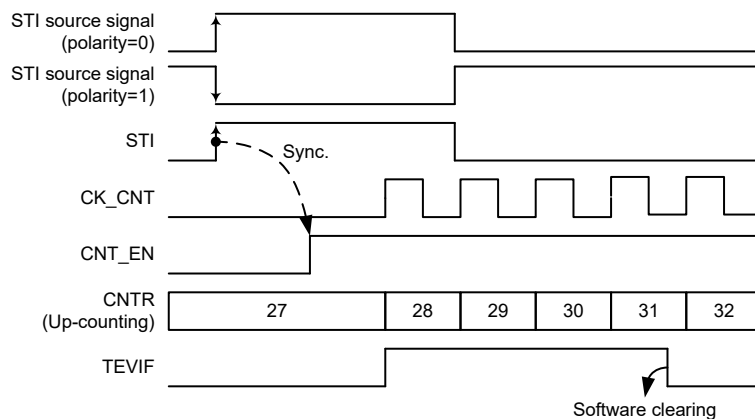
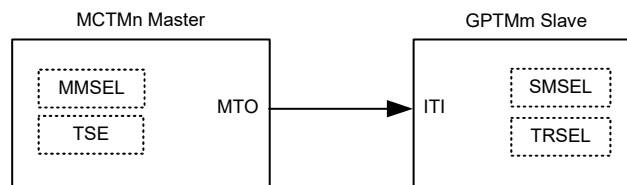


Figure 83. MCTM in Trigger Mode

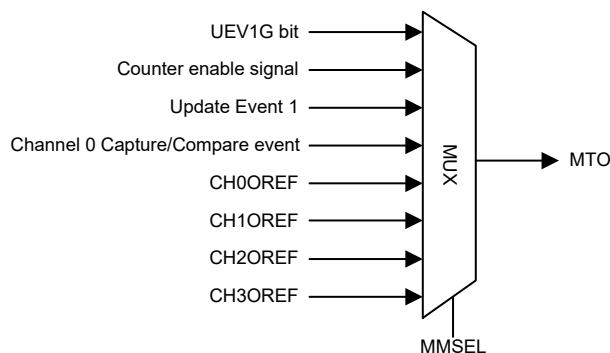
## Master Controller

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



**Figure 84. Master MCTMn and Slave GPTMm Connection**

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



**Figure 85. MTO Selection**

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

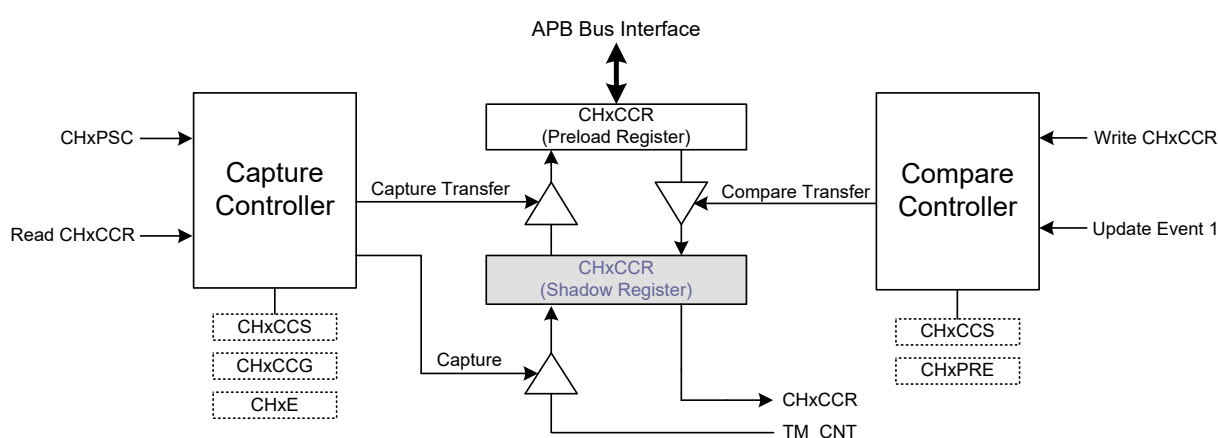


## Channel Controller

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

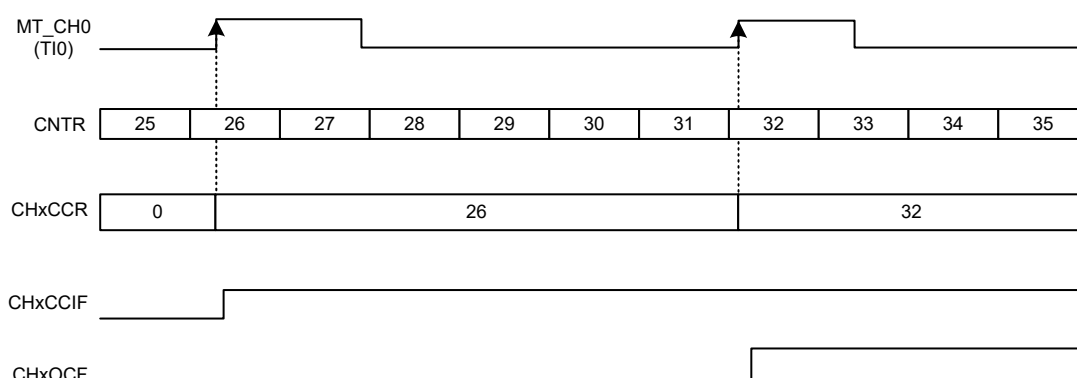
When used in the compare match output mode, the contents of the CHxCCR preload register is copied into the associated shadow register, the counter value is then compared with the register value.



### Figure 86. Capture/Compare Block Diagram

### Capture Counter Value Transferred to CHxCCR

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



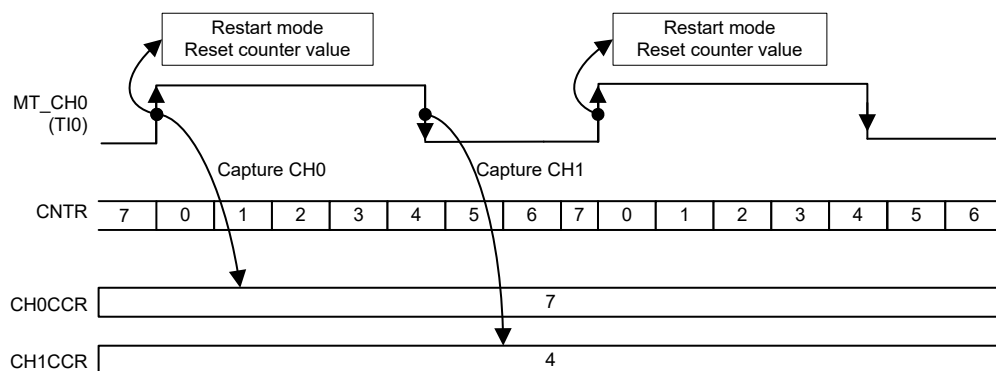
### Figure 87. Input Capture Mode

## Pulse Width Measurement

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

- Configure the capture channel 0 (CH0CCS = 0x1) to select the TI0 signal as the capture input.
- Configure the CH0P bit to 0 to choose the rising edge of the TI0 input as the active polarity.
- Configure the capture channel 1 (CH1CCS = 0x2) to select the TI0 signal as the capture input.
- Set the CH1P bit to 1 to choose the falling edge of the TI0 input as the active polarity.
- Setup 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 SMSEL field in the MDCFR register to 0x4
- Enable the input capture mode by setting the CH0E and CH1E bits in the CHCTR register to 1.

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



**Figure 88. PWM Pulse Width Measurement Example**

## Input Stage

The input stage consists of a digital filter, a channel polarity selection, edge detection and a channel prescaler. The channel 0 input signal TI0 can be chosen to come from the MT\_CH0 signal or the Exclusive-OR function of the MT\_CH0, MT\_CH1 and MT\_CH2 signals. The channel input signal TIx is sampled by a digital filter to generate a filtered input signal TIxFP. Then the channel polarity and the edge detection block can generate a TIxSyED signal for the input capture function. The effective input event number can be set by the channel capture input source prescaler setting field CHxPSC.

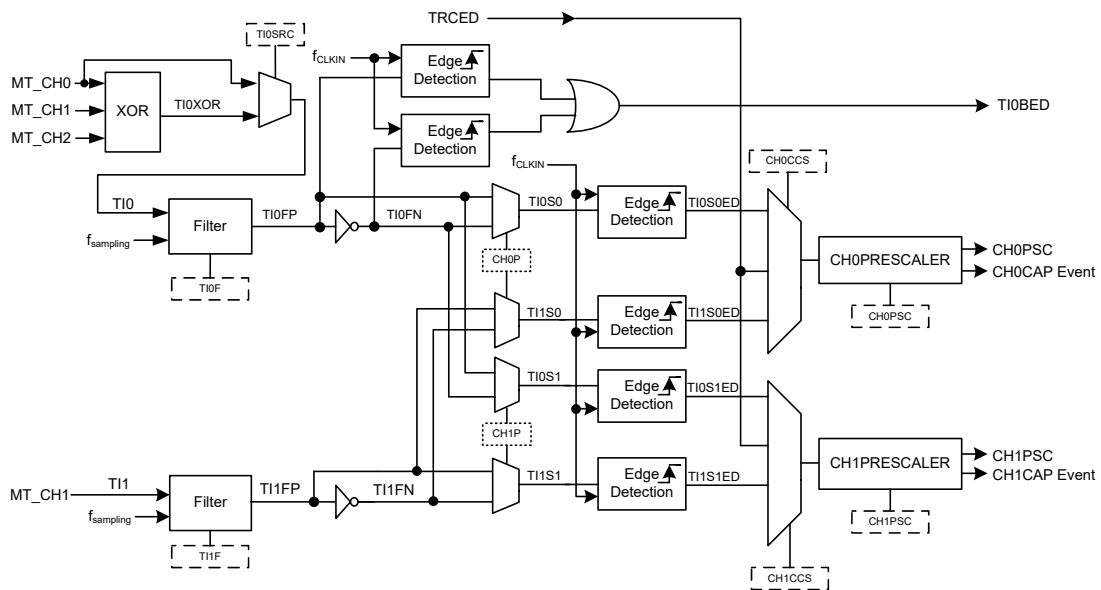


Figure 89. Channel 0 and Channel 1 Input Stages

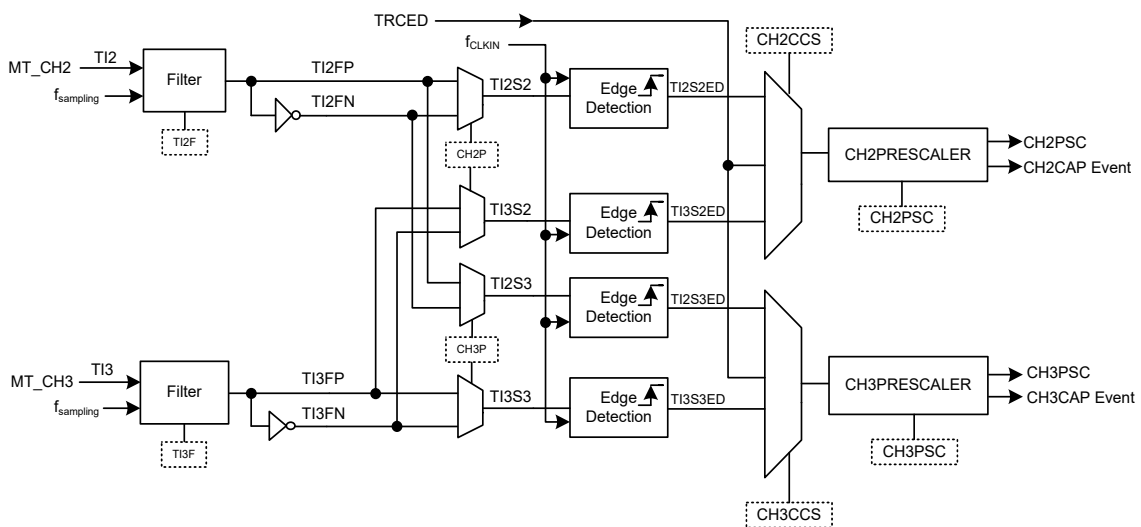


Figure 90. Channel 2 and Channel 3 Input Stages

## Digital Filter

The digital filters are embedded in the input stage and clock controller block for the MT\_CH0 ~ MT\_CH3 pins. The digital filter in the MCTM is an N-event counter where N refers to how many valid transitions are necessary to output a filtered signal. The N value can be 0, 2, 4, 5, 6 or 8 according to the selection for each filter.

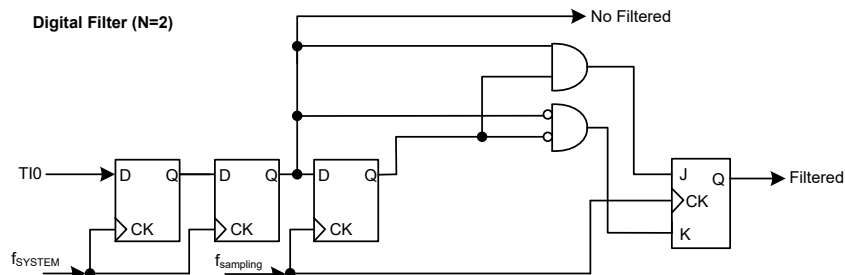


Figure 91. TIO Digital Filter Diagram with N = 2

## Output Stage

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

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

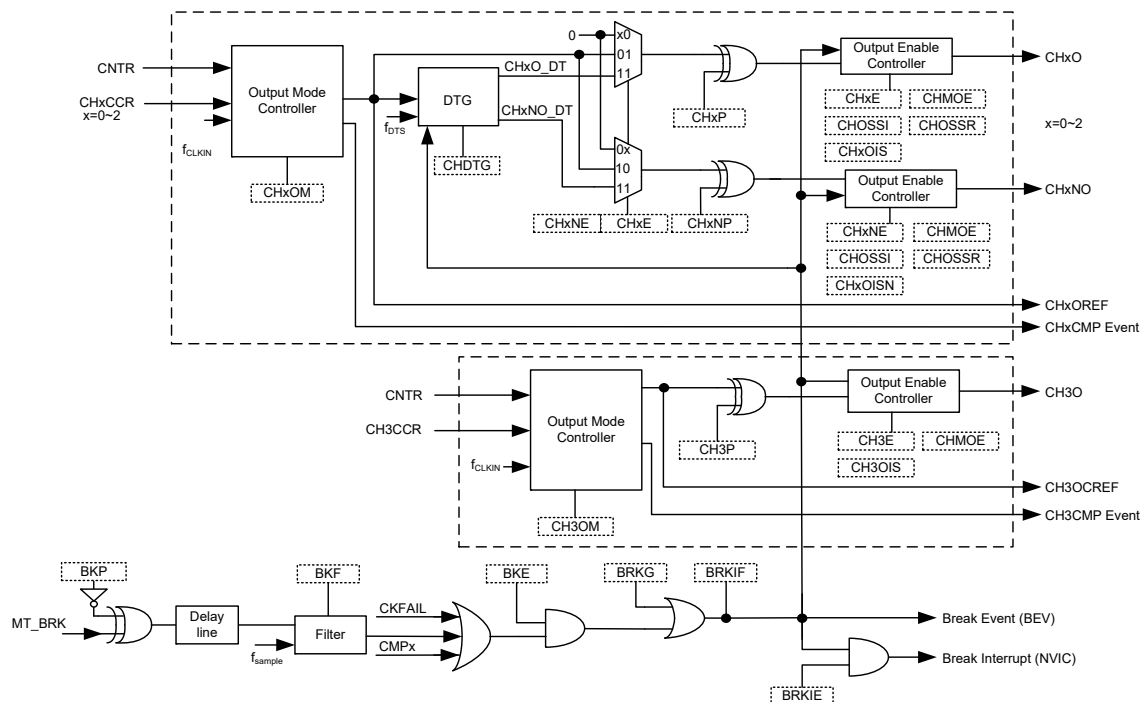


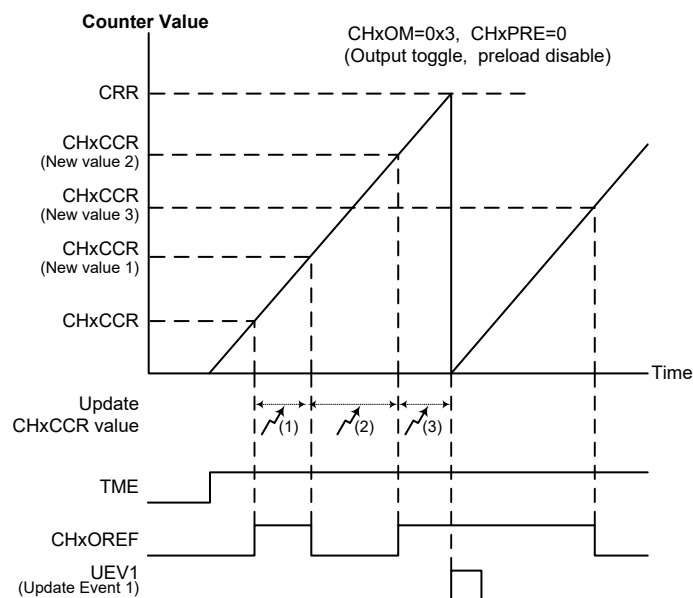
Figure 92. Output Stage Block Diagram

### Channel Output Reference Signal

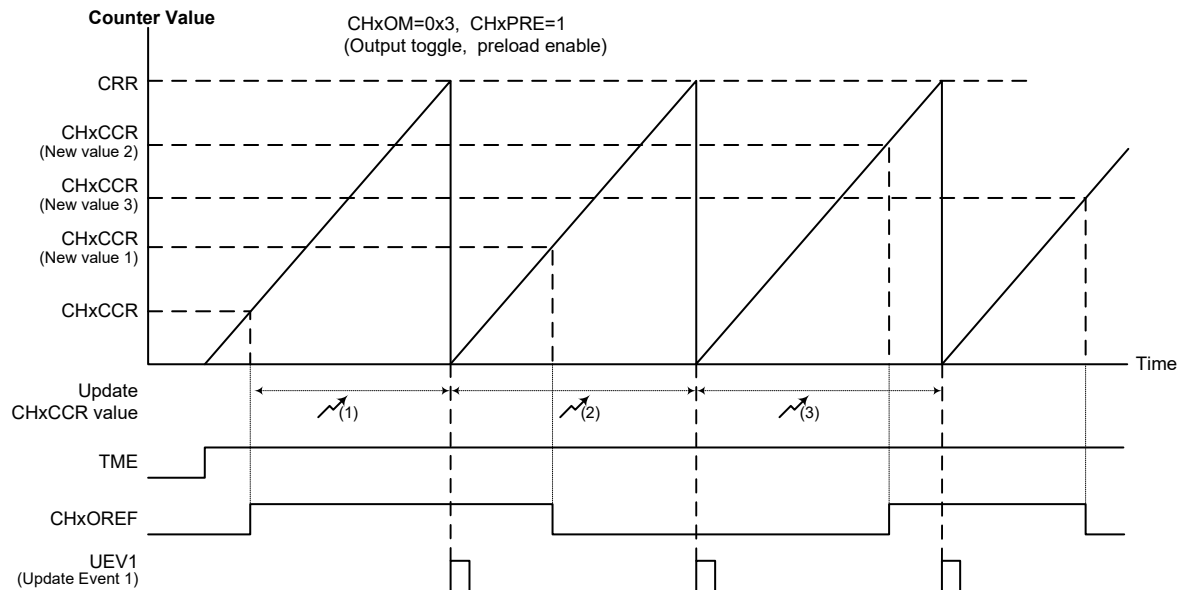
When the MCTM is used in the compare match output mode, the CHxOREF signal (Channel x Output Reference signal) is defined by the CHxOM field setup. The CHxOREF signal has several types of output function which defines what happens to the output when the counter value matches the contents of the CHxCCR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

**Table 35. Compare Match Output Setup**

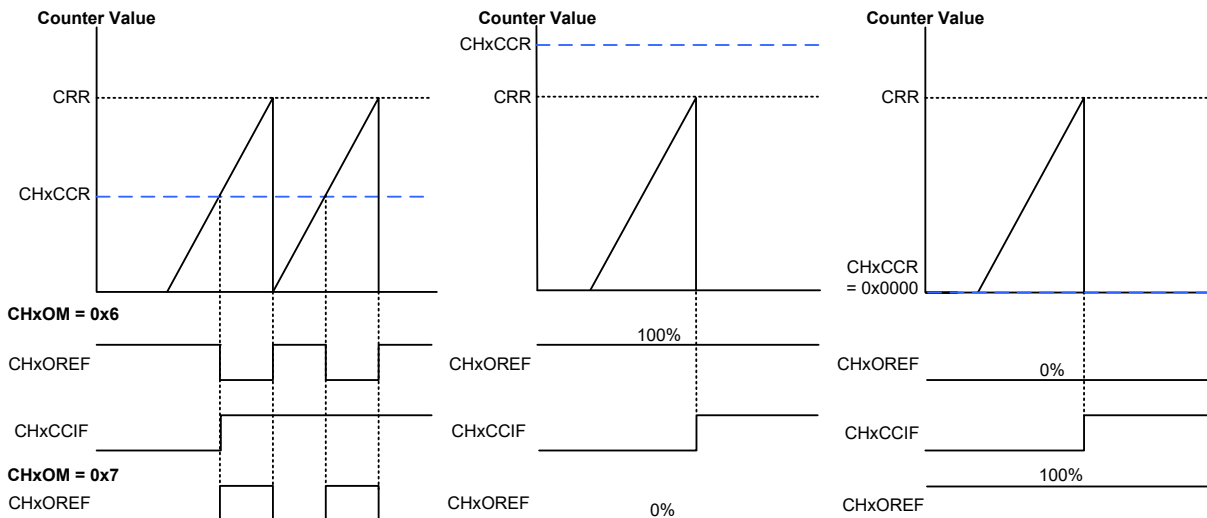
CHxOM Value	Compare Match Level
0x0	No change
0x1	Clear Output to 0
0x2	Set Output to 1
0x3	Toggle Output
0x4	Force Inactive Level
0x5	Force Active Level
0x6	PWM Mode 1
0x7	PWM Mode 2



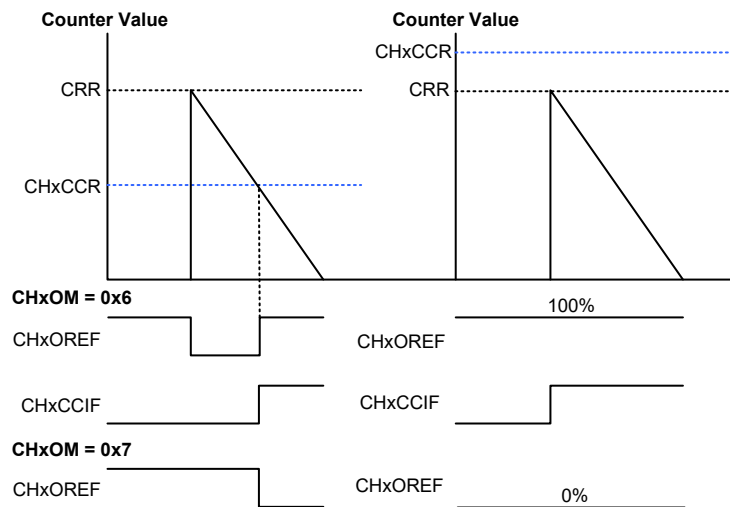
**Figure 93. Toggle Mode Channel Output Reference Signal – CHxPRE = 0**



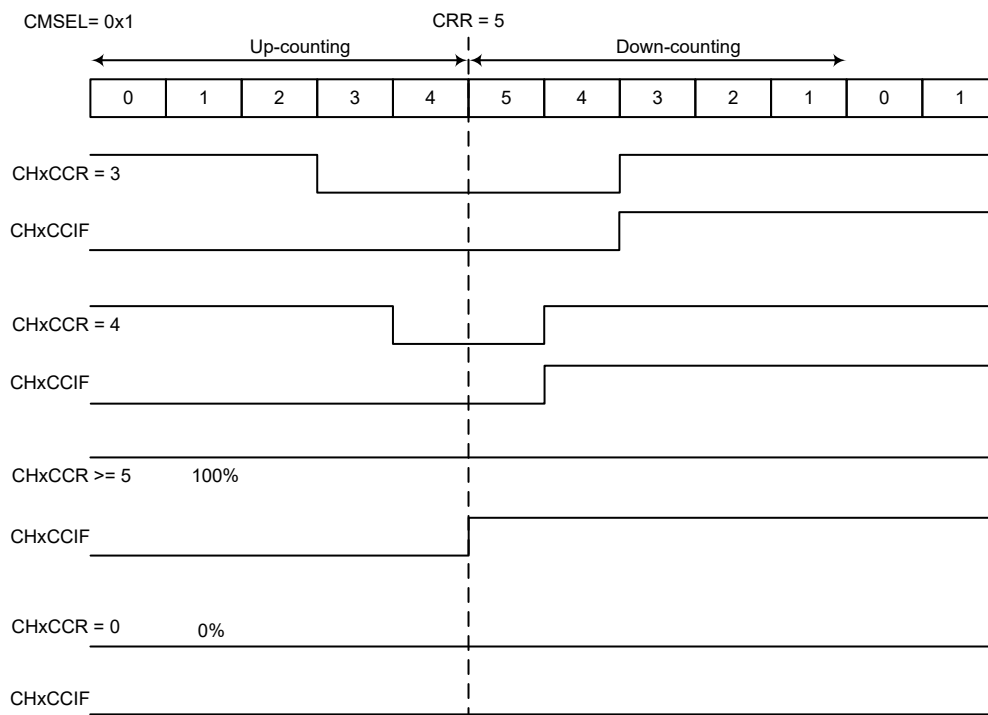
**Figure 94. Toggle Mode Channel Output Reference Signal – CHxPRE = 1**



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



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

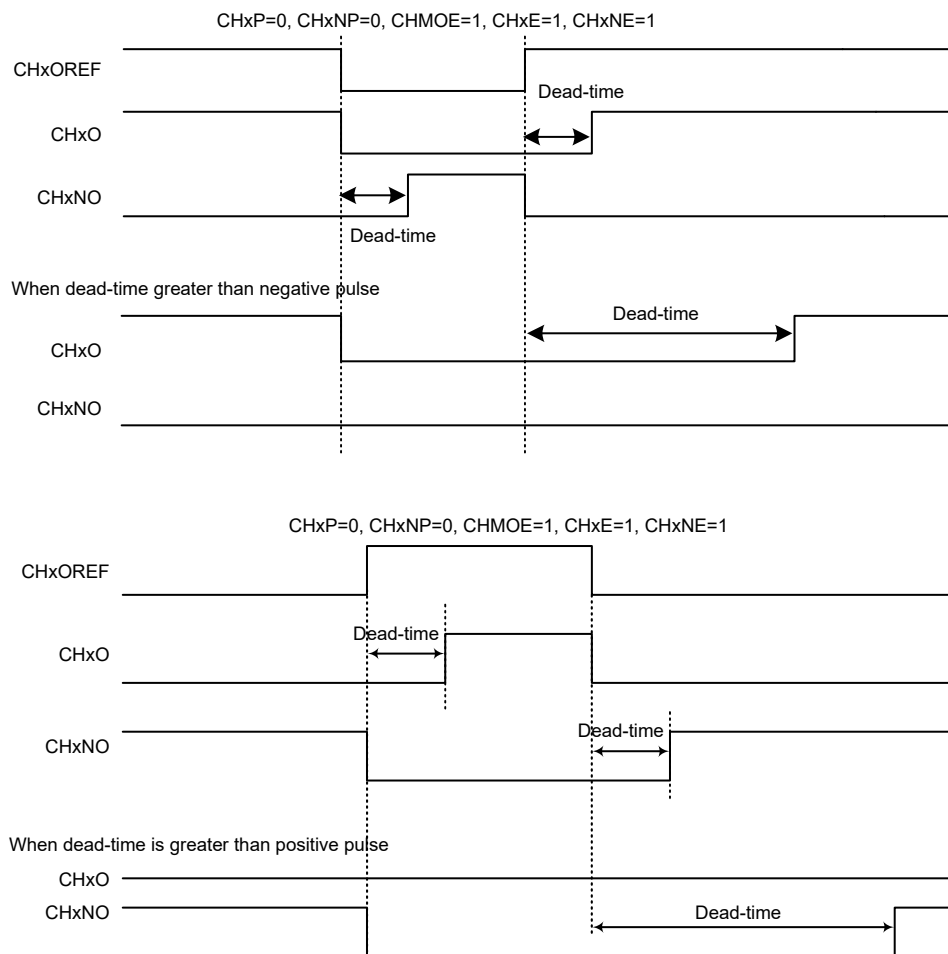


**Figure 97. PWM Mode 1 Channel Output Reference Signal and Counter in Centre-Aligned Counting Mode**

### Dead Time Generator

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

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



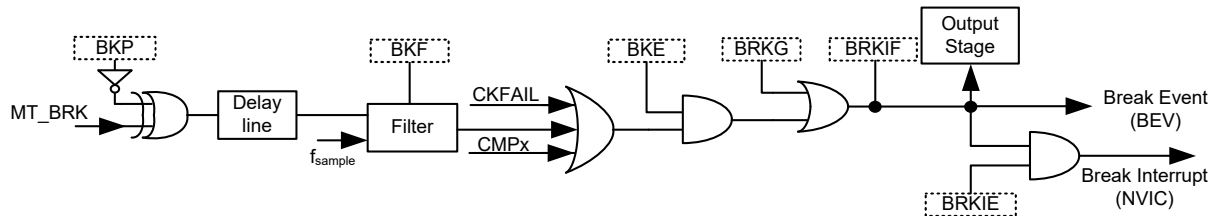
**Figure 98. Dead-Time Insertion Performed for Complementary Outputs**

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



## Break Function

The MCTM includes break function and one input signal for MCTM break. The MT\_BRK is default function and from the external MT\_BRK pin. The detailed block diagram is shown as below figure.

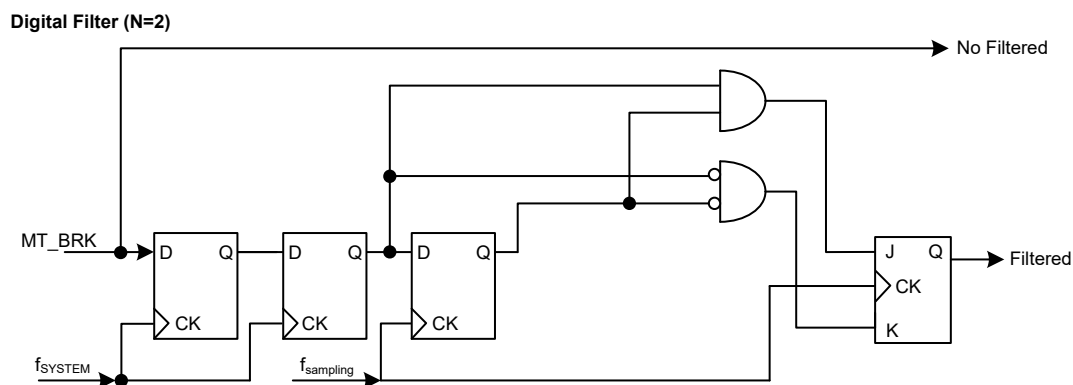


### Figure 99. MCTM Break Signal Bolck Diagram

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

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

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



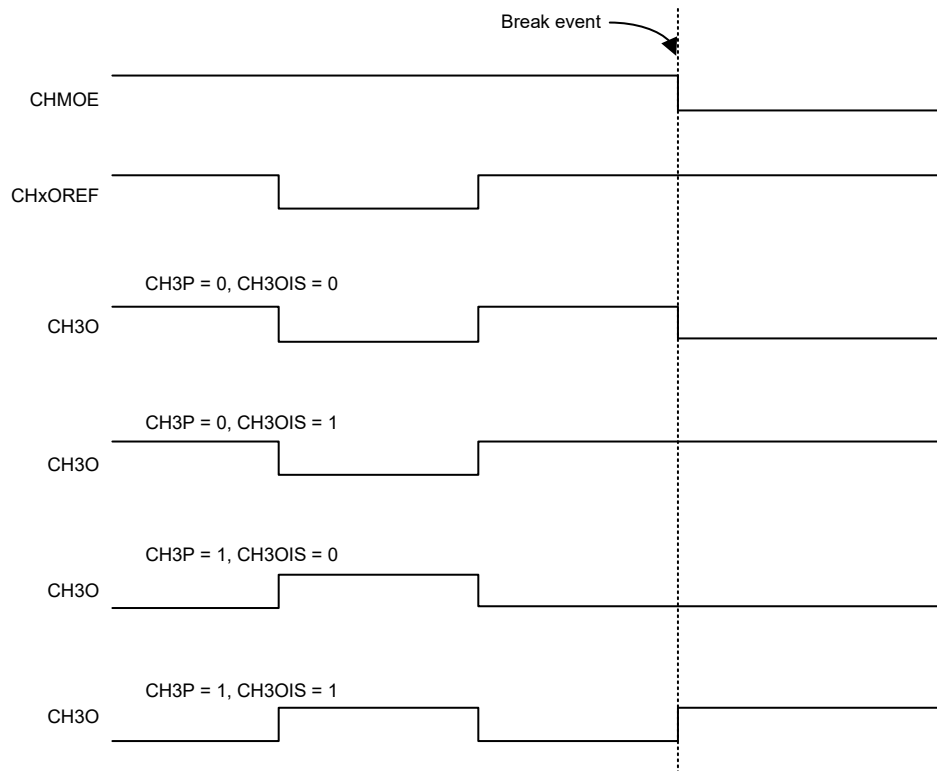
**Figure 100. MT\_BRK Pin Digital Filter Diagram with N = 2**

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

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

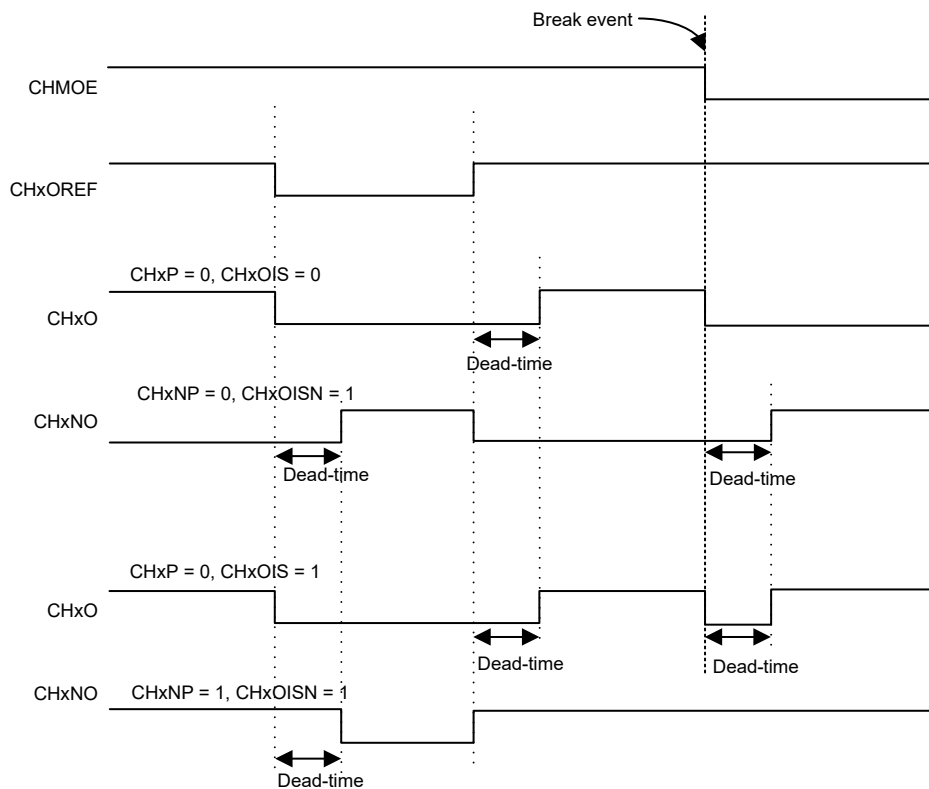
- If complementary outputs are not used (Channel 3), the channel will output an idle state.

The main output enable control bit CHMOE cannot be set until the break event is cleared.



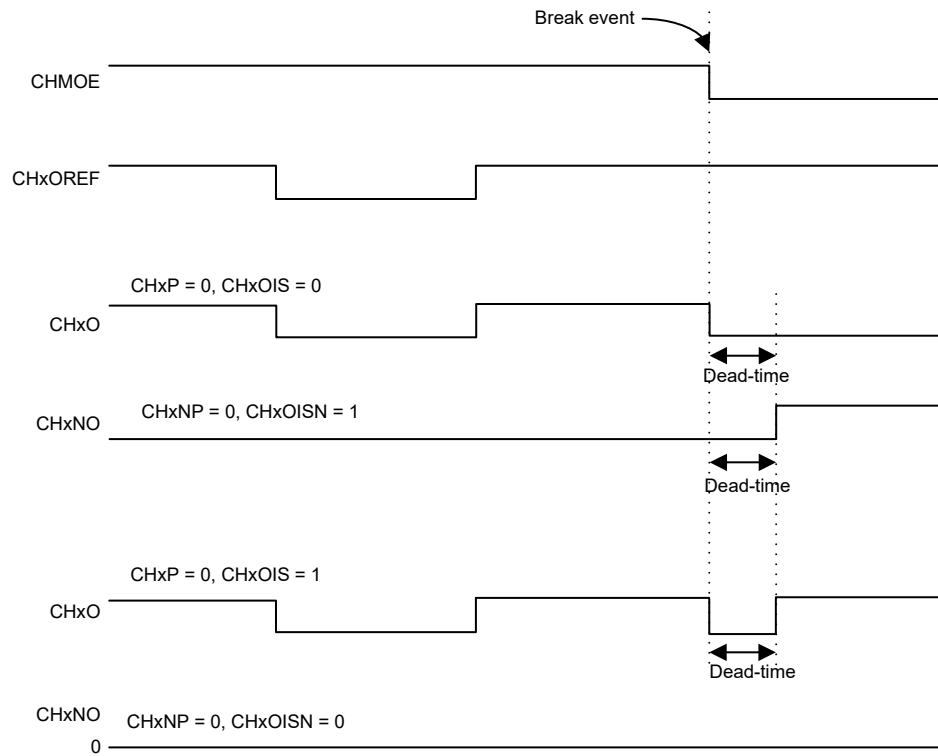
**Figure 101. Channel 3 Output with a Break Event Occurrence**

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



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

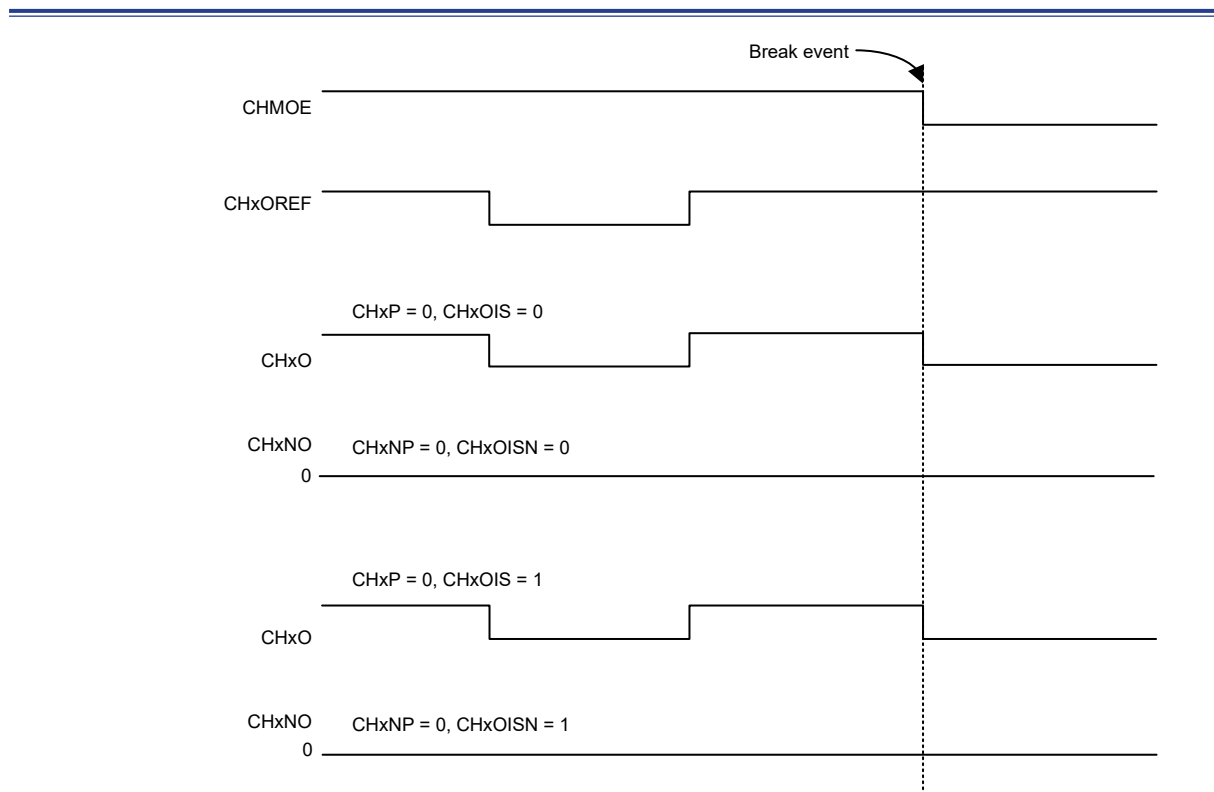
The accompanying diagram shows the output states in the case of the output being enabled by setting the CHxE bit to 1 and the complementary output being disabled by clearing the CHxNE to 0 when a break event occurs.



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

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

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



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

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

#### Channel Complementary Output with Break Function

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

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

Control Bits					Output Status	
CHMOE	CHOSI	CHOSSR	CHxE	CHxNE	MT_CHx Pin Output State	MT_CHxN Pin Output State
1 (Run)	x	0	0	0	Output disabled - floating - not driven by the timer MT_CHx <sup>(1)</sup> = floating MT_CHx_OEN <sup>(2)</sup> = 1	Output disabled - floating - not driven by the timer MT_CHxN = floating MT_CHxN_OEN = 1
		0	0	1	Output disabled - floating - not driven by the timer MT_CHx_OEN = 1	Output enabled MT_CHxN = CHx_OREF xor CHxNP MT_CHxN_OEN = 0
		0	1	0	Output enabled MT_CHx = CHx_OREF xor CHxP MT_CHx_OEN = 0	Output disabled - floating - not driven by the timer MT_CHxN = floating MT_CHxN_OEN = 1
		0	1	1	Output enabled MT_CHx = CHx_OREF xor CHxP + dead-time MT_CHx_OEN = 0	Output enabled MT_CHxN = not CHx_OREF xor CHxNP + dead-time MT_CHxN_OEN = 0
		1	0	0	Output disabled - floating - not driven by the timer MT_CHx = floating MT_CHx_OEN = 1	Output disabled - floating - not driven by the timer MT_CHxN = floating MT_CHxN_OEN = 1
		1	0	1	Off-State MT_CHx = CHxP MT_CHx_OEN = 0	Output enabled MT_CHxN = CHx_OREF xor CHxNP MT_CHxN_OEN = 0
		1	1	0	Output enabled MT_CHx = CHx_OCREF xor CHxP MT_CHx_OEN = 0	Off-State MT_CHxN = CHxNP MT_CHxN_OEN = 0
		1	1	1	Output enabled MT_CHx = CHx_OREF xor CHxP + dead-time MT_CHx_OEN = 0	Output enabled MT_CHxN = not CHx_OREF xor CHxNP + dead-time MT_CHxN_OEN = 0
0 (Idle)	0	x	0	0	Before dead-time: Output disabled - floating MT_CHx = floating, MT_CHxN = floating MT_CHx_OEN = 1, MT_CHxN_OEN = 1	
	0		0	1		
	0		1	0		
	0		1	1		
	1		0	0	After dead-time: Output enabled MT_CHx = CHxOIS, MT_CHxN = CHxOISN MT_CHx_OEN = 0, MT_CHxN_OEN = 0	
	1		0	1		
	1		1	0		
	1		1	1		

Note: 1. The MT\_CHx pin is the MCTM I/O Pin.

2. The MT\_CHx\_OEN and MT\_CHxN\_OEN signals are the MCTM I/O pin output enable combinational logic control signals which are active low.

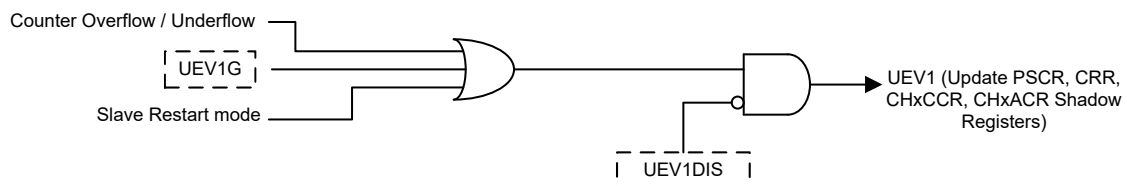
## Update Management

The update events are categorised into two different types which are the update event 1, UEV1, and update event 2, UEV2. The update event 1 is used to update the CRR, the PSCR, the CHxACR and the CHxCCR values from the actual registers to the corresponding shadow registers. An update event 1 occurs when the counter overflows or underflows, the UEV1G bit is set or the slave restart mode is triggered. The update event 2 is used to update the CHxE, CHxNE and CHxOM control bits. An update event 2 is generated when a rising edge on the STI occurs or the corresponding software update event 2 generation bit, UEV2G, is set.

### Update Event 1

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

#### Update Event 1 Management



#### Update Event 1 Interrupt Management

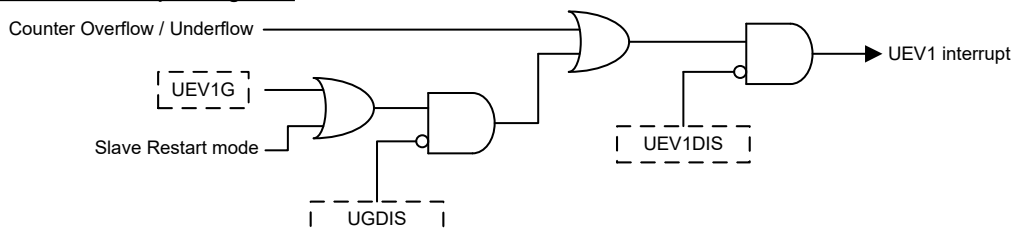
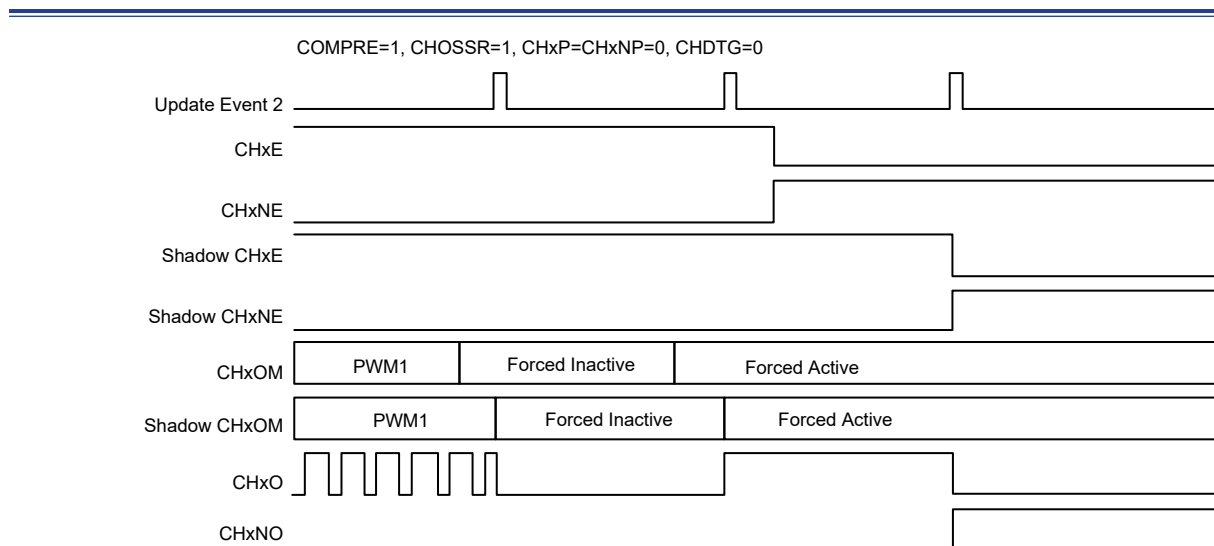


Figure 105. Update Event 1 Setup Diagram

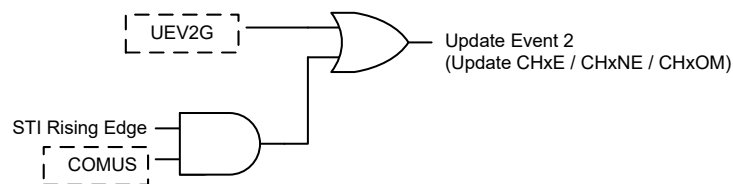
## Update Event 2

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



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

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



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



## Single Pulse Mode

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

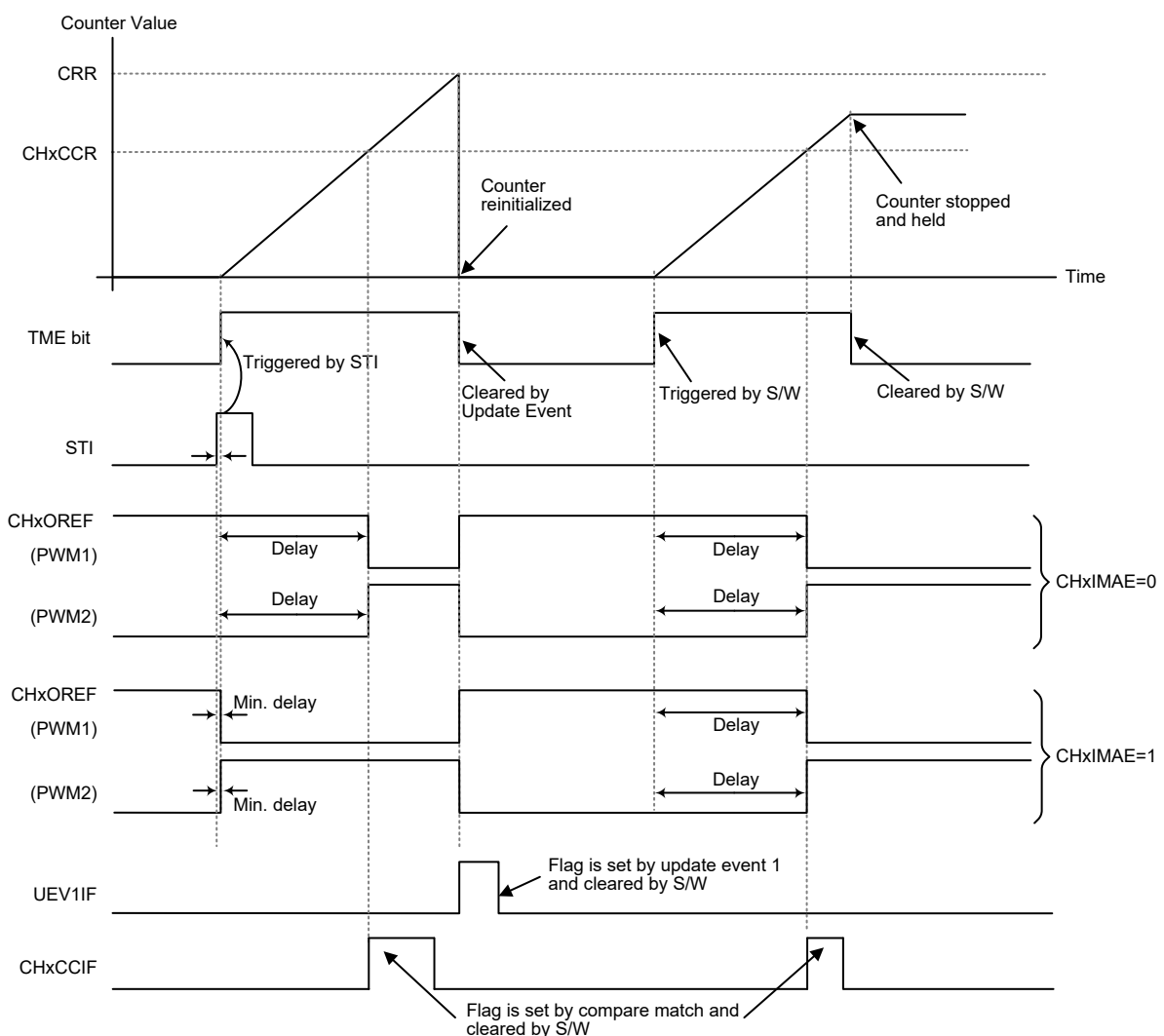
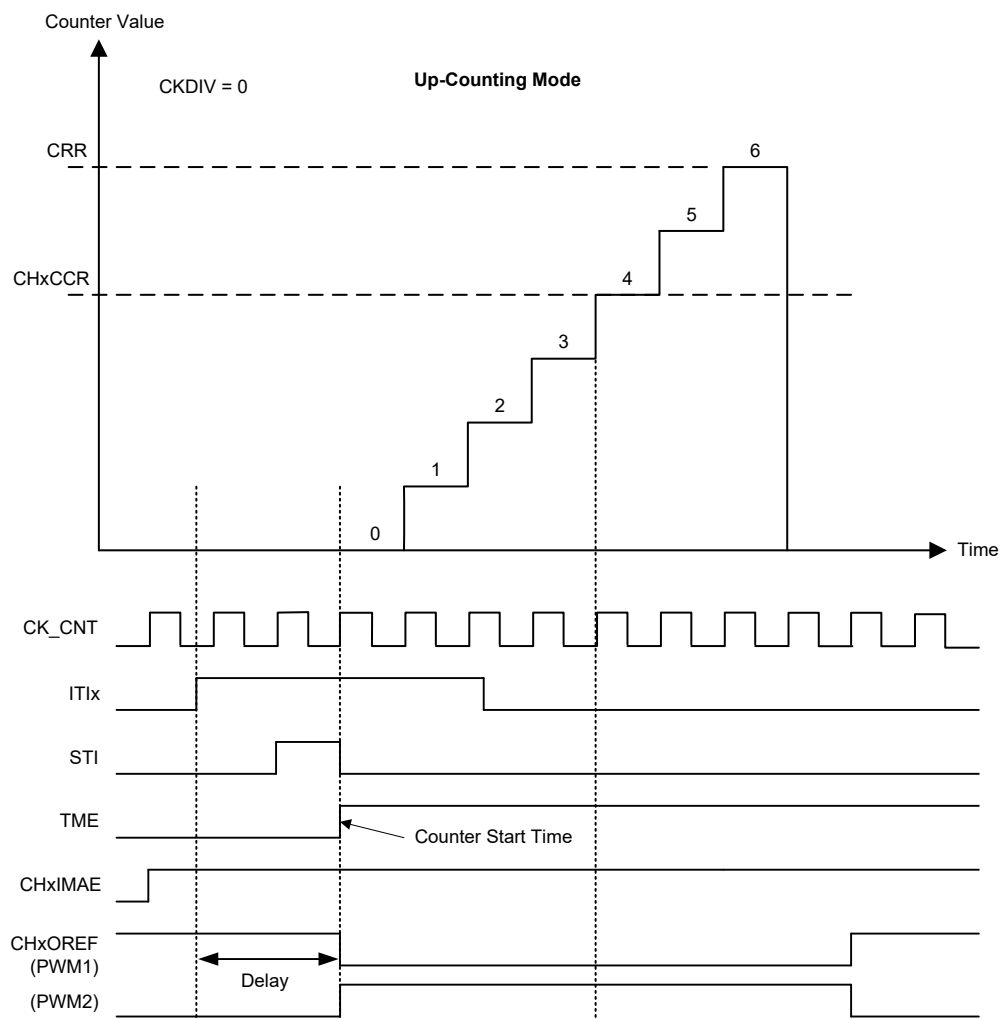


Figure 108. Single Pulse Mode

In the Single Pulse mode, the STI active edge which sets the TME bit to 1 will enable the counter. However, there exist several clock delays to perform the comparison result between the counter value and the CHxCCR value. In order to reduce the delay to a value, users 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 to which the CHxOREF signal will change to as the compare match event occurs without taking the comparison result into account. The CHxIMAE bit is available only when the output channel is configured to operate in the PWM mode 1 or PWM mode 2 and the trigger source is derived from the STI signal.

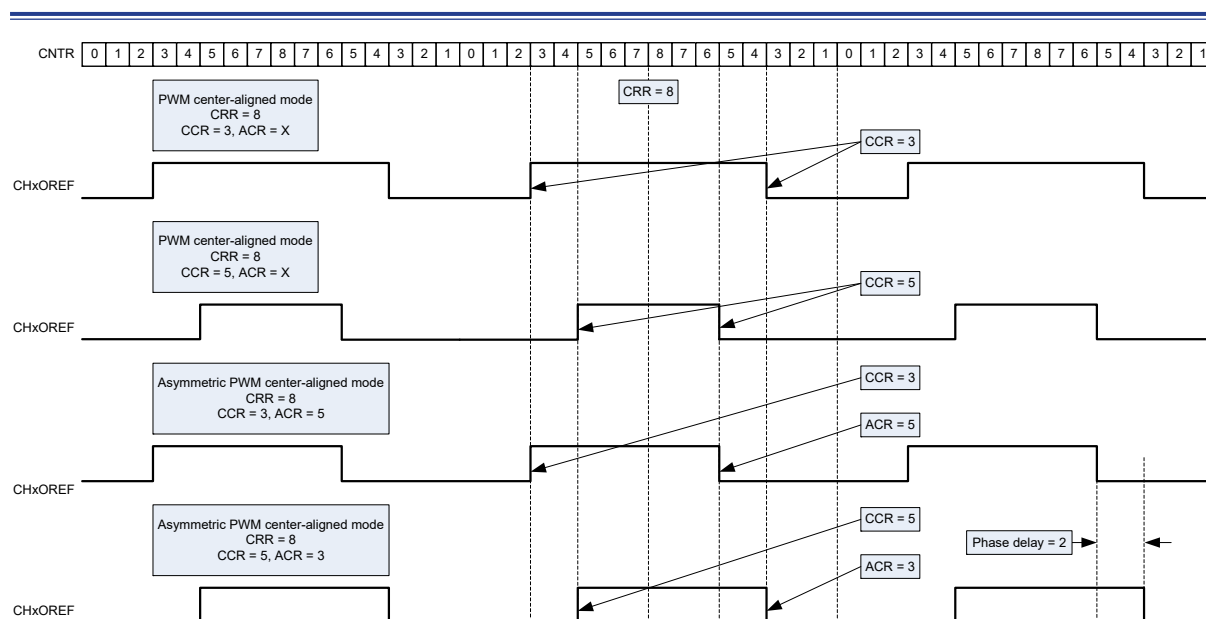


**Figure 109. Immediate Active Mode Delay**

## 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 in counting down stage, then the value in the CHxACR register is used as the down-count compare value. The following figure 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.



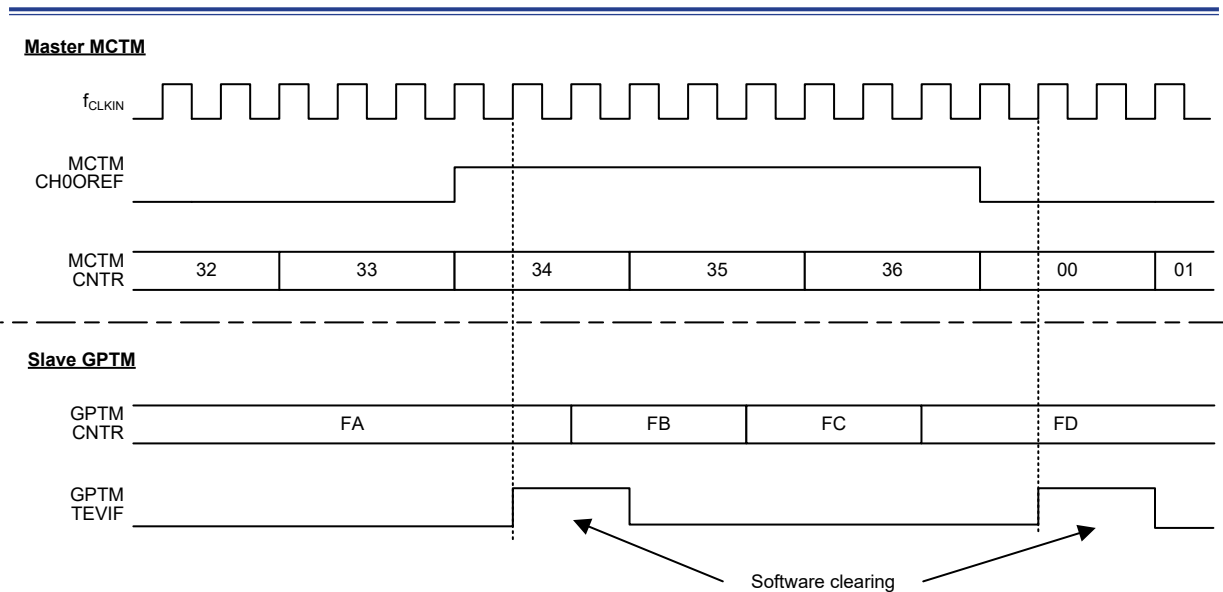
**Figure 110. Asymmetric PWM Mode versus Center-Aligned Counting Mode**

## Timer Interconnection

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

### Using One Timer to Trigger another Timer to Start or Stop Counting

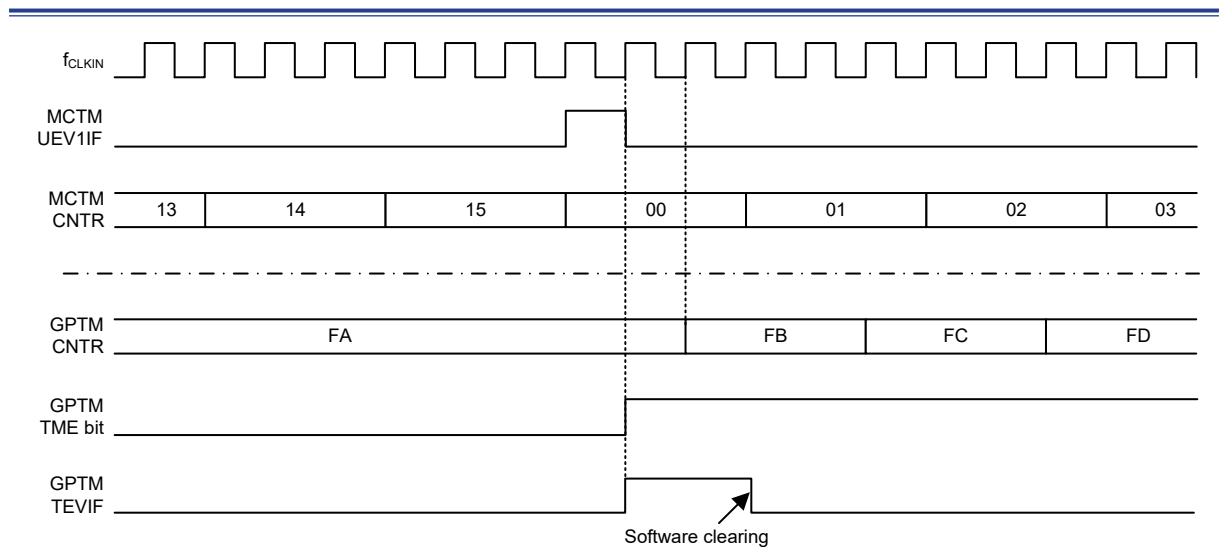
- Configure MCTM to be in the master mode and to send its channel 0 Output Reference signal CH0OREF as a trigger output (MMSEL = 0x4).
- Configure the MCTM CH0OREF waveform.
- Configure the GPTM to receive its input trigger source from the MCTM trigger output (TRSEL = 0xA).
- Configure GPTM to operate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



**Figure 111. Pausing GPTM Using the MCTM CH0OREF Signal**

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

- Configure MCTM to operate in the master mode and to send its Update Event 1 UEV1 as the trigger output (MMSEL = 0x2).
- Configure the MCTM period by setting the CRR register.
- Configure GPTM to get the input trigger source from the MCTM trigger output (TRSEL = 0xA).
- Configure GPTM to be in the slave trigger mode (SMSEL = 0x6).
- Start MCTM by writing '1' to the TME bit.

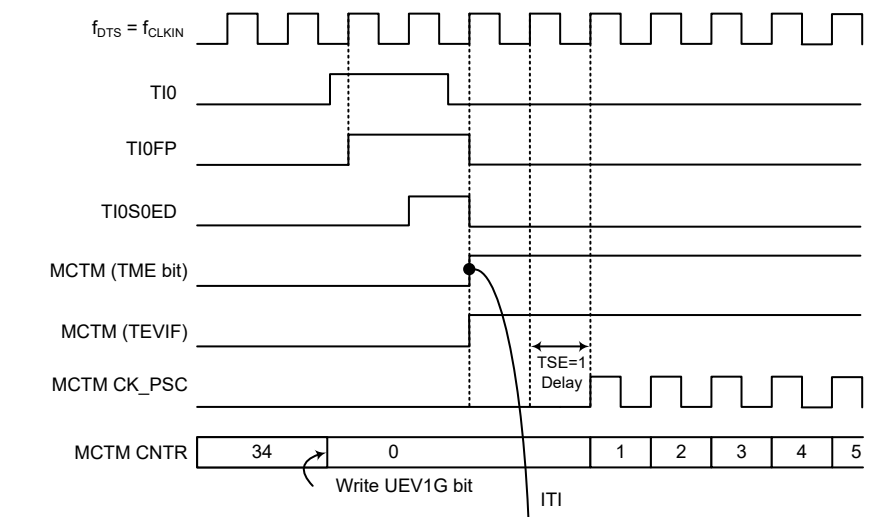


**Figure 112. Triggering GPTM with MCTM Update Event 1**

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

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

#### Master MCTM



#### Slave GPTM

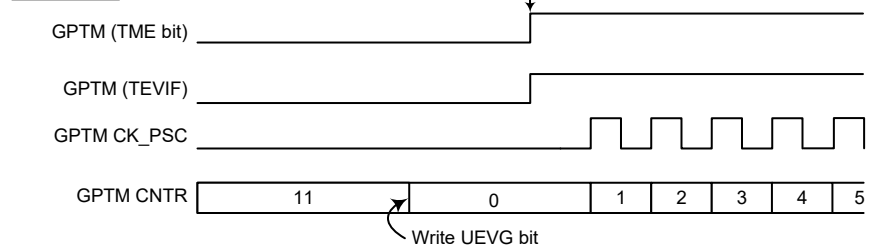


Figure 113. Trigger MCTM and GPTM with the MCTM CH0 Input

## Using One Timer as a Hall Sensor Interface to Trigger another Timer with Update Event 2

### GPTM

- Configure channel 0 to choose an input XOR function (TI0SRC = 1)
- Configure channel 0 to be in the input capture mode and TRCED as capture source (CH0CCS = 0x3) and Enable channel 0 (CH0E = 1)
- Configure the UEVG bit as the source of MTO (MMSEL = 0x0)
- Configure TI0BED to be connected to STI (TRSEL = 0x8)
- Configure the counter to be in the slave restart mode (SMSEL = 0x4)
- Enable GPTM (TME = 1)

### MCTM

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

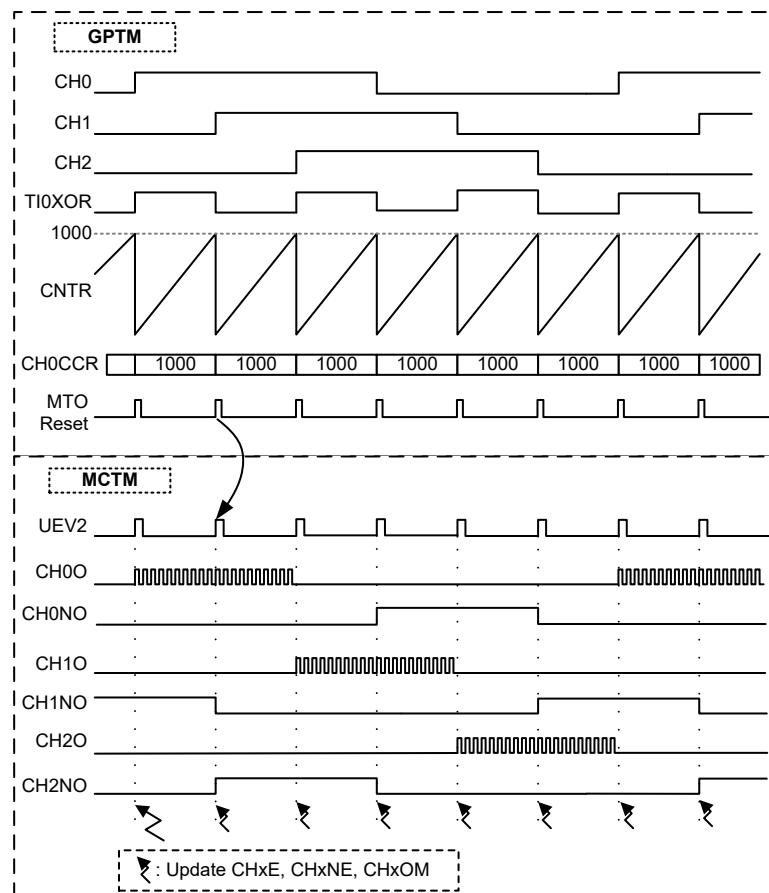


Figure 114. CH0XOR Input as Hall Sensor Interface

## Trigger Peripheral Start

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

## Lock Level Table

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

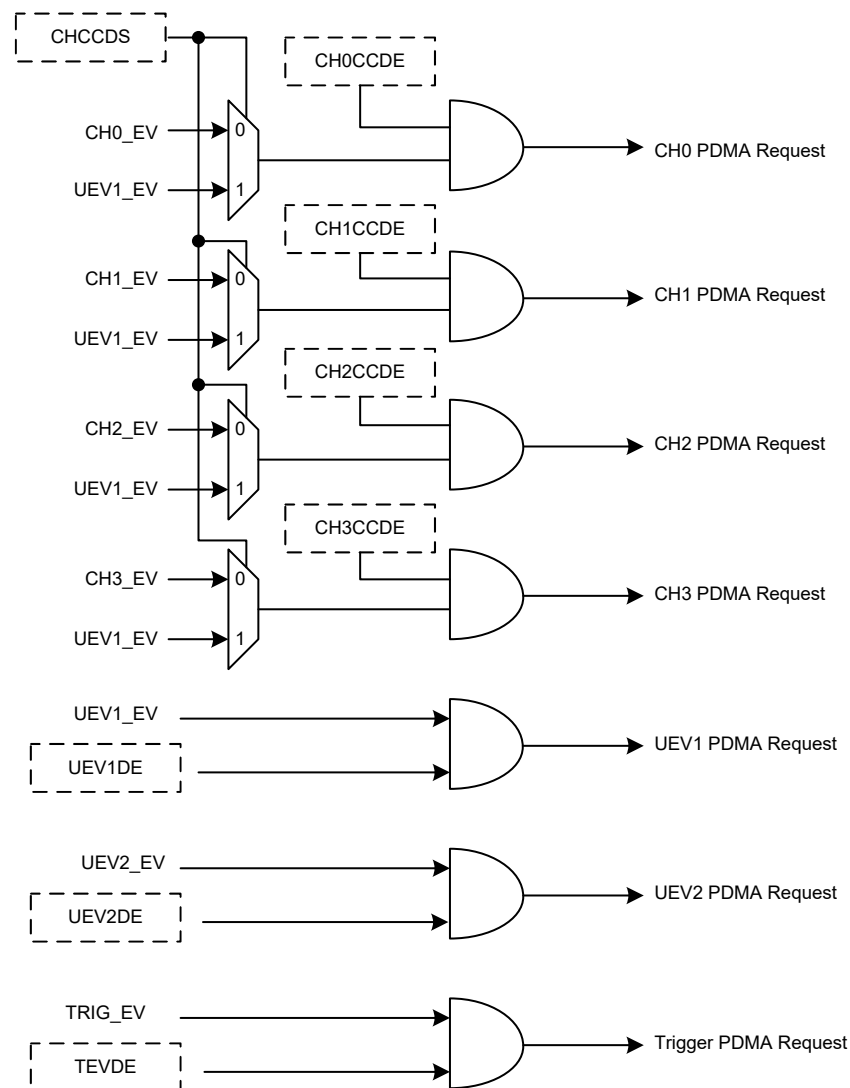
**Table 37. Lock Level Table**

Lock Configuration	Protected Bits					
Lock Level 1 (LOCKLV = "01")	CHDTG	CHxOIS	CHxOISN	BKE	BKP	CHAOE
Lock Level 2 (LOCKLV = "10")	CHDTG	CHxOIS	CHxOISN	BKE	BKP	CHAOE
	CHxP	CHxNP	CHxSSI	CHxSSR	MCTMEN <sup>(1)</sup>	CKMEN <sup>(2)</sup>
Lock Level 3 (LOCKLV = "11")	CHDTG	CHxOIS	CHxOISN	BKE	BKP	CHAOE
	CHxP	CHxNP	CHxSSI	CHxSSR	MCTMEN <sup>(1)</sup>	CKMEN <sup>(2)</sup>
	CHxPRE	CHxOM				

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

## PDMA Request (HT32F54243/HT32F54253 only)

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



**Figure 115. MCTM PDMA Mapping Diagram**



## Register Map

The following table shows the MCTM registers and reset values. The PDMA related describes are only available for the HT32F54243/HT32F54253 devices.

**Table 38. MCTM Register Map**

Register	Offset	Description	Reset Value
CNTCFR	0x000	Timer Counter Configuration Register	0x0000_0000
MDCFR	0x004	Timer Mode Configuration Register	0x0000_0000
TRCFR	0x008	Timer Trigger Configuration Register	0x0000_0000
CTR	0x010	Timer Control Register	0x0000_0000
CH0ICFR	0x020	Channel 0 Input Configuration Register	0x0000_0000
CH1ICFR	0x024	Channel 1 Input Configuration Register	0x0000_0000
CH2ICFR	0x028	Channel 2 Input Configuration Register	0x0000_0000
CH3ICFR	0x02C	Channel 3 Input Configuration Register	0x0000_0000
CH0OCFR	0x040	Channel 0 Output Configuration Register	0x0000_0000
CH1OCFR	0x044	Channel 1 Output Configuration Register	0x0000_0000
CH2OCFR	0x048	Channel 2 Output Configuration Register	0x0000_0000
CH3OCFR	0x04C	Channel 3 Output Configuration Register	0x0000_0000
CHCTR	0x050	Channel Control Register	0x0000_0000
CHPOLR	0x054	Channel Polarity Configuration Register	0x0000_0000
CHBRKCFR	0x06C	Channel Break Configuration Register	0x0000_0000
CHBRKCTR	0x070	Channel Break Control Register	0x0000_0002
DICTR	0x074	Timer PDMA/Interrupt Control Register	0x0000_0000
EVGR	0x078	Timer Event Generator Register	0x0000_0000
INTSR	0x07C	Timer Interrupt Status Register	0x0000_0000
CNTR	0x080	Timer Counter Register	0x0000_0000
PSCR	0x084	Timer Prescaler Register	0x0000_0000
CRR	0x088	Timer Counter-Reload Register	0x0000_FFFF
REPR	0x08C	Timer Repetition Register	0x0000_0000
CH0CCR	0x090	Channel 0 Capture/Compare Register	0x0000_0000
CH1CCR	0x094	Channel 1 Capture/Compare Register	0x0000_0000
CH2CCR	0x098	Channel 2 Capture/Compare Register	0x0000_0000
CH3CCR	0x09C	Channel 3 Capture/Compare Register	0x0000_0000
CH0ACR	0x0A0	Channel 0 Asymmetric Compare Register	0x0000_0000
CH1ACR	0x0A4	Channel 1 Asymmetric Compare Register	0x0000_0000
CH2ACR	0x0A8	Channel 2 Asymmetric Compare Register	0x0000_0000
CH3ACR	0x0AC	Channel 3 Asymmetric Compare Register	0x0000_0000

## Register Descriptions

### Timer Counter Configuration Register – CNTCFR

This register specifies the MCTM counter configuration.

Offset: 0x000

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
	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	UEV1DIS
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 counting mode.
[17:16]	CMSEL	Counter Mode Selection 00: Edge-aligned counting mode. Normal up-counting and down-counting available for this mode. Counting direction is defined by the DIR bit. 01: Center-aligned counting mode 1. The counter counts up and down alternatively. The compare match interrupt flag is set during the count-down period. 10: Center-aligned counting mode 2. The counter counts up and down alternatively. The compare match interrupt flag is set during the count-up period. 11: Center-aligned counting mode 3. The counter counts up and down alternatively. The compare match interrupt flag is set during the count-up and count-down period.
[9:8]	CKDIV	Clock Division These two bits define the frequency ratio between the timer clock ( $f_{CLKIN}$ ) and the dead-time clock ( $f_{DTS}$ ). The dead-time clock is also used as the digital filter sampling clock. 00: $f_{DTS} = f_{CLKIN}$ 01: $f_{DTS} = f_{CLKIN}/2$ 10: $f_{DTS} = f_{CLKIN}/4$ 11: Reserved

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

### Timer Mode Configuration Register – MDCFR

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

Offset: 0x004

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
	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 an update event occurred or not 1: Counter stops counting at the next update event and then the TME bit is cleared by hardware

Bits	Field	Descriptions																											
[18:16]	MMSEL	<p>Master Mode Selection</p> <p>Master mode selection is used to select the MTO signal source which is used to synchronise the other slave timer.</p> <table> <tr> <th>MMSEL [2:0]</th><th>Mode</th><th>Descriptions</th></tr> <tr> <td>000</td><td>Reset Mode</td><td>The MTO in the Reset mode is an output derived from one of the following cases: 1. Software setting UEV1G bit 2. Slave has trigger input when used in slave restart mode</td></tr> <tr> <td>001</td><td>Enable Mode</td><td>The Counter Enable signal is used as the trigger output.</td></tr> <tr> <td>010</td><td>Update Mode</td><td>The update event 1 is used as the trigger output according to one of the following cases when the UEV1DIS bit is cleared to 0: 1. Counter overflow/underflow 2. Software setting UEV1G 3. Slave has trigger input when used in slave restart mode</td></tr> <tr> <td>011</td><td>Capture/Compare Mode</td><td>When a Channel 0 capture or compare match event occurs, it will generate a positive pulse which is used as the master trigger output.</td></tr> <tr> <td>100</td><td>Compare output 0</td><td>The Channel 0 Output reference signal named CH0OREF is used as the trigger output.</td></tr> <tr> <td>101</td><td>Compare output 1</td><td>The Channel 1 Output reference signal named CH1OREF is used as the trigger output.</td></tr> <tr> <td>110</td><td>Compare output 2</td><td>The Channel 2 Output reference signal named CH2OREF is used as the trigger output.</td></tr> <tr> <td>111</td><td>Compare output 3</td><td>The Channel 3 Output reference signal named CH3OREF is used as the trigger output.</td></tr> </table>	MMSEL [2:0]	Mode	Descriptions	000	Reset Mode	The MTO in the Reset mode is an output derived from one of the following cases: 1. Software setting UEV1G bit 2. Slave has trigger input when used in slave restart mode	001	Enable Mode	The Counter Enable signal is used as the trigger output.	010	Update Mode	The update event 1 is used as the trigger output according to one of the following cases when the UEV1DIS bit is cleared to 0: 1. Counter overflow/underflow 2. Software setting UEV1G 3. Slave has trigger input when used in slave restart mode	011	Capture/Compare Mode	When a Channel 0 capture or compare match event occurs, it will generate a positive pulse which is used as the master trigger output.	100	Compare output 0	The Channel 0 Output reference signal named CH0OREF is used as the trigger output.	101	Compare output 1	The Channel 1 Output reference signal named CH1OREF is used as the trigger output.	110	Compare output 2	The Channel 2 Output reference signal named CH2OREF is used as the trigger output.	111	Compare output 3	The Channel 3 Output reference signal named CH3OREF is used as the trigger output.
MMSEL [2:0]	Mode	Descriptions																											
000	Reset Mode	The MTO in the Reset mode is an output derived from one of the following cases: 1. Software setting UEV1G bit 2. Slave has trigger input when used in slave restart mode																											
001	Enable Mode	The Counter Enable signal is used as the trigger output.																											
010	Update Mode	The update event 1 is used as the trigger output according to one of the following cases when the UEV1DIS bit is cleared to 0: 1. Counter overflow/underflow 2. Software setting UEV1G 3. Slave has trigger input when used in slave restart mode																											
011	Capture/Compare Mode	When a Channel 0 capture or compare match event occurs, it will generate a positive pulse which is used as the master trigger output.																											
100	Compare output 0	The Channel 0 Output reference signal named CH0OREF is used as the trigger output.																											
101	Compare output 1	The Channel 1 Output reference signal named CH1OREF is used as the trigger output.																											
110	Compare output 2	The Channel 2 Output reference signal named CH2OREF is used as the trigger output.																											
111	Compare output 3	The Channel 3 Output reference signal named CH3OREF is used as the trigger output.																											

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

## Timer Trigger Configuration Register – TRCFR

This register specifies the trigger source selection of MCTM.

Offset: 0x008

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/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 UEV1G bit</p> <p>0001: Channel 0 filtered input – TI0S0</p> <p>0010: Channel 1 filtered input – TI1S1</p> <p>1000: Channel 0 Edge Detector – TI0BED</p> <p>1001: Internal Timer Trigger 0 – ITI0</p> <p>1010: Internal Timer Trigger 1 – ITI1</p> <p>1011: Internal Timer 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 39. MCTM Internal Trigger Connection**

Slave Timing Module	ITI0	ITI1	ITI2
MCTM	—	GPTM	—

## Timer Control Register – CTR

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

Offset: 0x010

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[16]	CHCCDS	Channel Capture/Compare PDMA Selection 0: Channel PDMA request derived from the channel capture/compare event 1: Channel PDMA request derived from the update event 1
[9]	COMUS	Capture/Compare Control Update Selection 0: Updated by setting the UEV2G bit only 1: Updated by setting the UEV2G bit or when an STI signal rising edge occurs This bit is only available when the capture/compare preload function is enabled by setting the COMPARE bit to 1.
[8]	COMPARE	Capture/Compare Preloaded Enable Control 0: CHxE, CHxNE and CHxOM bits are not preloaded 1: CHxE, CHxNE and CHxOM bits are preloaded If this bit is set to 1, the corresponding capture/compare control bits including the CHxE, CHxNE and CHxOM bits will be updated when the update event 2 occurs.
[1]	CRBE	Counter-Reload register Buffer Enable 0: Counter-Reload register can be updated immediately 1: Counter-Reload register cannot be updated until the update event occurs
[0]	TME	Timer Enable bit 0: MCTM off 1: MCTM on – MCTM functions normally When the TME bit is cleared to 0, the counter is stopped and the MCTM consumes no power in any operational mode except for the single pulse mode and the slave trigger mode. In these two modes the TME bit can automatically be set to 1 by hardware which permits all the MCTM registers to function normally.

## Channel 0 Input Configuration Register – CH0ICFR

This register specifies the channel 0 input mode configuration.

Offset: 0x020

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
	TI0SRC	Reserved						
Type/Reset	RW	0						
	23	22	21	20	19	18	17	16
	Reserved				CH0PSC		CH0CCS	
Type/Reset					RW	0	RW	0
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved				TI0F			
Type/Reset					RW	0	RW	0

Bits	Field	Descriptions
[31]	TI0SRC	Channel 0 Input Source TI0 Selection 0: The MT_CH0 pin is connected to the channel 0 input TI0 1: The XOR operation output of the MT_CH0, MT_CH1, and MT_CH2 pins are connected to the channel 0 input TI0
[19:18]	CH0PSC	Channel 0 Capture Input Source Prescaler Setting These bits define the effective events of the channel 0 capture input. Note that the prescaler is reset once the Channel 0 Capture/Compare Enable bit, CH0E, in the Channel Control register named CHCTR is cleared to 0. 00: No prescaler, channel 0 capture input signal is chosen for each active event 01: Channel 0 Capture input signal is chosen for every 2 events 10: Channel 0 Capture input signal is chosen for every 4 events 11: Channel 0 Capture input signal is chosen for every 8 events
[17:16]	CH0CCS	Channel 0 Capture/Compare Selection 00: Channel 0 is configured as an output 01: Channel 0 is configured as an input derived from the TI0 signal 10: Channel 0 is configured as an input derived from the TI1 signal 11: Channel 0 is configured as an input which comes from the TRCED signal derived from the Trigger Controller Note: The CH0CCS field can be accessed only when the CH0E bit is cleared to 0.



Bits	Field	Descriptions
[3:0]	TIOF	<p>Channel 0 Input Source TIO Filter Setting</p> <p>These bits define the frequency divided ratio used to sample the TIO signal. The Digital filter in the MCTM is an N-event counter where N is defined as how many valid transitions are necessary to output a filtered signal.</p> <p>0000: No filter, the sampling clock is <math>f_{\text{SYSTEM}}</math>.</p> <p>0001: <math>f_{\text{sampling}} = f_{\text{CLKIN}}</math>, <math>N = 2</math></p> <p>0010: <math>f_{\text{sampling}} = f_{\text{CLKIN}}</math>, <math>N = 4</math></p> <p>0011: <math>f_{\text{sampling}} = f_{\text{CLKIN}}</math>, <math>N = 8</math></p> <p>0100: <math>f_{\text{sampling}} = f_{\text{DTS}}/2</math>, <math>N = 6</math></p> <p>0101: <math>f_{\text{sampling}} = f_{\text{DTS}}/2</math>, <math>N = 8</math></p> <p>0110: <math>f_{\text{sampling}} = f_{\text{DTS}}/4</math>, <math>N = 6</math></p> <p>0111: <math>f_{\text{sampling}} = f_{\text{DTS}}/4</math>, <math>N = 8</math></p> <p>1000: <math>f_{\text{sampling}} = f_{\text{DTS}}/8</math>, <math>N = 6</math></p> <p>1001: <math>f_{\text{sampling}} = f_{\text{DTS}}/8</math>, <math>N = 8</math></p> <p>1010: <math>f_{\text{sampling}} = f_{\text{DTS}}/16</math>, <math>N = 5</math></p> <p>1011: <math>f_{\text{sampling}} = f_{\text{DTS}}/16</math>, <math>N = 6</math></p> <p>1100: <math>f_{\text{sampling}} = f_{\text{DTS}}/16</math>, <math>N = 8</math></p> <p>1101: <math>f_{\text{sampling}} = f_{\text{DTS}}/32</math>, <math>N = 5</math></p> <p>1110: <math>f_{\text{sampling}} = f_{\text{DTS}}/32</math>, <math>N = 6</math></p> <p>1111: <math>f_{\text{sampling}} = f_{\text{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	Channel 1 Capture Input Source Prescaler Setting These bits define the effective events of the channel 1 capture input. Note that the prescaler is reset once the Channel 1 Capture/Compare Enable bit, CH1E, in the Channel Control register named CHCTR is cleared to 0. 00: No prescaler, channel 1 capture input signal is chosen for each active event 01: Channel 1 Capture input signal is chosen for every 2 events 10: Channel 1 Capture input signal is chosen for every 4 events 11: Channel 1 Capture input signal is chosen for every 8 events
[17:16]	CH1CCS	Channel 1 Capture/Compare Selection 00: Channel 1 is configured as an output 01: Channel 1 is configured as an input derived from the 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 Note: The CH1CCS field can be accessed only when the CH1E bit is cleared to 0.

Bits	Field	Descriptions
[3:0]	TI1F	<p>Channel 1 Input Source TI1 Filter Setting</p> <p>These bits define the frequency divide ratio used to sample the TI1 signal. The Digital filter in the MCTM is an N-event counter where N is defined as how many valid transitions are necessary to output a filtered signal</p> <p>0000: No filter, the sampling clock is <math>f_{\text{SYSTEM}}</math>.</p> <p>0001: <math>f_{\text{sampling}} = f_{\text{CLKIN}}</math>, <math>N = 2</math></p> <p>0010: <math>f_{\text{sampling}} = f_{\text{CLKIN}}</math>, <math>N = 4</math></p> <p>0011: <math>f_{\text{sampling}} = f_{\text{CLKIN}}</math>, <math>N = 8</math></p> <p>0100: <math>f_{\text{sampling}} = f_{\text{DTS}}/2</math>, <math>N = 6</math></p> <p>0101: <math>f_{\text{sampling}} = f_{\text{DTS}}/2</math>, <math>N = 8</math></p> <p>0110: <math>f_{\text{sampling}} = f_{\text{DTS}}/4</math>, <math>N = 6</math></p> <p>0111: <math>f_{\text{sampling}} = f_{\text{DTS}}/4</math>, <math>N = 8</math></p> <p>1000: <math>f_{\text{sampling}} = f_{\text{DTS}}/8</math>, <math>N = 6</math></p> <p>1001: <math>f_{\text{sampling}} = f_{\text{DTS}}/8</math>, <math>N = 8</math></p> <p>1010: <math>f_{\text{sampling}} = f_{\text{DTS}}/16</math>, <math>N = 5</math></p> <p>1011: <math>f_{\text{sampling}} = f_{\text{DTS}}/16</math>, <math>N = 6</math></p> <p>1100: <math>f_{\text{sampling}} = f_{\text{DTS}}/16</math>, <math>N = 8</math></p> <p>1101: <math>f_{\text{sampling}} = f_{\text{DTS}}/32</math>, <math>N = 5</math></p> <p>1110: <math>f_{\text{sampling}} = f_{\text{DTS}}/32</math>, <math>N = 6</math></p> <p>1111: <math>f_{\text{sampling}} = f_{\text{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				CH2PSC		CH2CCS	
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved				TI2F			
					RW	0	RW	0
							RW	0
								RW
								0

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 divide ratio used to sample the TI2 signal. The Digital filter in the MCTM is an N-event counter where N is defined as how many valid transitions are necessary to output a filtered signal.</p> <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			
					RW	0	RW	0
							RW	0
								RW
								0

Bits	Field	Descriptions
[19:18]	CH3PSC	Channel 3 Capture Input Source Prescaler Setting 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. 00: No prescaler, channel 3 capture input signal is chosen for each active event 01: Channel 3 Capture input signal is chosen for every 2 events 10: Channel 3 Capture input signal is chosen for every 4 events 11: Channel 3 Capture input signal is chosen for every 8 events
[17:16]	CH3CCS	Channel 3 Capture/Compare Selection 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 Note: The CH3CCS field can be accessed only when the CH3E bit is cleared to 0.

Bits	Field	Descriptions
[3:0]	TI3F	<p>Channel 3 Input Source TI3 Filter Setting</p> <p>These bits define the frequency divide ratio used to sample the TI3 signal. The digital filter in the GPTM is an N-event counter where N is defined as how many valid transitions are necessary to output a filtered signal</p> <p>0000: No filter, the sampling clock is <math>f_{\text{SYSTEM}}</math>.</p> <p>0001: <math>f_{\text{sampling}} = f_{\text{CLKIN}}</math>, <math>N = 2</math></p> <p>0010: <math>f_{\text{sampling}} = f_{\text{CLKIN}}</math>, <math>N = 4</math></p> <p>0011: <math>f_{\text{sampling}} = f_{\text{CLKIN}}</math>, <math>N = 8</math></p> <p>0100: <math>f_{\text{sampling}} = f_{\text{DTS}} / 2</math>, <math>N = 6</math></p> <p>0101: <math>f_{\text{sampling}} = f_{\text{DTS}} / 2</math>, <math>N = 8</math></p> <p>0110: <math>f_{\text{sampling}} = f_{\text{DTS}} / 4</math>, <math>N = 6</math></p> <p>0111: <math>f_{\text{sampling}} = f_{\text{DTS}} / 4</math>, <math>N = 8</math></p> <p>1000: <math>f_{\text{sampling}} = f_{\text{DTS}} / 8</math>, <math>N = 6</math></p> <p>1001: <math>f_{\text{sampling}} = f_{\text{DTS}} / 8</math>, <math>N = 8</math></p> <p>1010: <math>f_{\text{sampling}} = f_{\text{DTS}} / 16</math>, <math>N = 5</math></p> <p>1011: <math>f_{\text{sampling}} = f_{\text{DTS}} / 16</math>, <math>N = 6</math></p> <p>1100: <math>f_{\text{sampling}} = f_{\text{DTS}} / 16</math>, <math>N = 8</math></p> <p>1101: <math>f_{\text{sampling}} = f_{\text{DTS}} / 32</math>, <math>N = 5</math></p> <p>1110: <math>f_{\text{sampling}} = f_{\text{DTS}} / 32</math>, <math>N = 6</math></p> <p>1111: <math>f_{\text{sampling}} = f_{\text{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	Channel 0 Immediate Active Enable 0: No action 1: Single pulse Immediate Active Mode is enabled The CH0OREF will be forced to the compare matched level immediately after an available trigger event occurs irrespective of the result of the comparison between the CNTR and the CH0CCR values. The effective duration ends automatically at the next overflow or underflow event. Note: The CH0IMAE bit is available only if channel 0 is configured operate in PWM mode 1 or PWM mode 2.
[4]	CH0PRE	Channel 0 Capture/Compare Register (CH0CCR) Preload Enable 0: CH0CCR preload function is disabled The CH0CCR register can be immediately assigned a new value when the CH0PRE bit is cleared to 0 and the updated CH0CCR value is used immediately. 1: CH0CCR preload function is enabled The new CH0CCR value will not be transferred to its shadow register until an update event 1 occurs.



Bits	Field	Descriptions
[8][2:0]	CH0OM[3:0]	<p>Channel 0 Output Mode Setting</p> <p>These bits define the functional types of the output reference signal CH0OREF.</p> <p>0000: No Change</p> <p>0001: Output 0 on compare match</p> <p>0010: Output 1 on compare match</p> <p>0011: Output toggles on compare match</p> <p>0100: Force inactive – CH0OREF is forced to 0</p> <p>0101: Force active – CH0OREF is forced to 1</p> <p>0110: PWM mode 1</p> <ul style="list-style-type: none"> <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 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]	
								RW	0
	7	6	5	4	3	2	1	0	
Type/Reset	Reserved		CH1IMAE	CH1PRE	Reserved		CH1OM[2:0]		
			RW	0	RW	0		RW	0

Bits	Field	Descriptions
[5]	CH1IMAE	Channel 1 Immediate Active Enable 0: No action 1: Single pulse Immediate Active Mode is enabled The CH1OREF will be forced to the compare matched level immediately after an available trigger event occurs irrespective of the result of the comparison between the CNTR and the CH1CCR values. The effective duration ends automatically at the next overflow or underflow event. Note: The CH1IMAE bit is available only if channel 1 is configured to be operated in 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 an update event 1 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	Channel 2 Immediate Active Enable 0: No action 1: Single pulse Immediate Active Mode is enabled The CH2OREF will be forced to the compare matched level immediately after an available trigger event occurs irrespective of the result of the comparison between the CNTR and the CH2CCR values. The effective duration ends automatically at the next overflow or underflow event. Note: The CH2IMAE bit is available only if the channel 2 is configured to be operated in PWM mode 1 or PWM mode 2.
[4]	CH2PRE	Channel 2 Capture/Compare Register (CH2CCR) Preload Enable 0: CH2CCR preload function is disabled The CH2CCR register can be immediately assigned a new value when the CH2PRE bit is cleared to 0 and the updated CH2CCR value is used immediately. 1: CH2CCR preload function is enabled The new CH2CCR value will not be transferred to its shadow register until an update event 1 occurs.

Bits	Field	Descriptions
[8][2:0]	CH2OM[3:0]	<p>Channel 2 Output Mode Setting</p> <p>These bits define the functional types of the output reference signal CH2OREF.</p> <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]
	7	6	5	4	3	2	1	0
Type/Reset	Reserved		CH3IMAE	CH3PRE	Reserved	CH3OM[2:0]		
			RW	0	RW	0	RW	0

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

Bits	Field	Descriptions
[8][2:0]	CH3OM[3:0]	<p>Channel 3 Output Mode Setting</p> <p>These bits define the functional types of the output reference signal CH3OREF</p> <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	CH2NE	CH2E	CH1NE	CH1E	CH0NE	CH0E
	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[6]	CH3E	Channel 3 Capture/Compare Enable - Channel 3 is configured as an input (CH3CCS = 0x1/0x2/0x3) 0: Input Capture Mode is disabled 1: Input Capture Mode is enabled - Channel 3 is configured as an output (CH3CCS = 0x0) 0: Off – Channel 3 output signal CH3O is not active 1: On – Channel 3 output signal CH3O is generated on the corresponding output pin depending on the condition of the CHMOE, CHOSS1, CHOSSR and CH3OIS bits.
[5]	CH2NE	Channel 2 Capture/Compare Complementary Enable 0: Off – Channel 2 complementary output CH2NO is not active. The CH2NO level is then determined by the CHMOE, CHOSS1, CHOSSR, CH2OIS, CH2OISN and CH2E bits. 1: On – Channel 2 complementary output CH2NO is generated on the corresponding output pin depending on the condition of the CHMOE, CHOSS1, CHOSSR, CH2OIS, CH2OISN and CH2E bits.
[4]	CH2E	Channel 2 Capture/Compare Enable - Channel 2 is configured as an input (CH2CCS = 0x1/0x2/0x3) 0: Input Capture Mode is disabled 1: Input Capture Mode is enabled - Channel 2 is configured as an output (CH2CCS = 0x0) 0: Off – Channel 2 output signal CH2O is not active. The CH2O level is then determined by the condition of the CHMOE, CHOSS1, CHOSSR, CH2OIS, CH2OISN and CH2NE bits. 1: On – Channel 2 output signal CH2O is generated on the corresponding output pin determined by the condition of the CHMOE, CHOSS1, CHOSSR, CH2OIS, CH2OISN and CH2NE bits.



Bits	Field	Descriptions
[3]	CH1NE	Channel 1 Capture/Compare Complementary Enable 0: Off – Channel 1 complementary output CH1NO is not active. The CH1NO level is then determined by the condition of the CHMOE, CHOSSI, CHOSSR, CH1OIS, CH1OISN and CH1E bits. 1: On – Channel 1 complementary output CH1NO is generated on the corresponding output pin determined by the condition of the CHMOE, CHOSSI, CHOSSR, CH1OIS, CH1OISN and CH1E bits.
[2]	CH1E	Channel 1 Capture/Compare Enable - Channel 1 is configured as an input (CH1CCS = 0x1/0x2/0x3) 0: Input Capture Mode is disabled 1: Input Capture Mode is enabled - Channel 1 is configured as an output (CH1CCS = 0x0) 0: Off – Channel 1 output signal CH1O is not active. The CH1O level is then determined by the condition of the CHMOE, CHOSSI, CHOSSR, CH1OIS, CH1OISN and CH1NE bits. 1: On – Channel 1 output signal CH1O is generated on the corresponding output pin depending on the condition of the CHMOE, CHOSSI, CHOSSR, CH1OIS, CH1OISN and CH1NE bits.
[1]	CH0NE	Channel 0 Capture/Compare Complementary Enable 0: Off – Channel 0 complementary output CH0NO is not active. The CH0NO level is then determined by the condition of the CHMOE, CHOSSI, CHOSSR, CH0OIS, CH0OISN and CH0E bits. 1: On – Channel 0 complementary output CH0NO is generated on the corresponding output pin depending on the condition of the CHMOE, CHOSSI, CHOSSR, CH0OIS, CH0OISN and CH0E bits.
[0]	CH0E	Channel 0 Capture/Compare Enable - Channel 0 is configured as an input (CH0CCS = 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. The CH0O level is then determined by the condition of the CHMOE, CHOSSI, CHOSSR, CH0OIS, CH0OISN and CH0NE bits. 1: On – Channel 0 output signal CH0O is generated on the corresponding output pin determined by the condition of the CHMOE, CHOSSI, CHOSSR, CH0OIS, CH0OISN and CH0NE bits.

## Channel Polarity Configuration Register – CHPOLR

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

Offset: 0x054

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/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	CH2NP	CH2P	CH1NP	CH1P	CH0NP	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
[5]	CH2NP	Channel 2 Capture/Compare Complementary Polarity 0: Channel 2 Output is active high 1: Channel 2 Output is active low
[4]	CH2P	Channel 2 Capture/Compare Polarity - When Channel 2 is configured as an input (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
[3]	CH1NP	Channel 1 Capture/Compare Complementary Polarity 0: Channel 1 Output is active high 1: Channel 1 Output is active low
[2]	CH1P	Channel 1 Capture/Compare Polarity - When Channel 1 is configured as an input (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
[1]	CH0NP	Channel 0 Capture/Compare Complementary Polarity 0: Channel 0 Output is active high 1: Channel 0 Output is active low

Bits	Field	Descriptions
[0]	CH0P	Channel 0 Capture/Compare Polarity - When Channel 0 is configured as an input (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

### Channel Break Configuration Register – CHBRKCFR

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

Offset: 0x06C

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved	CH3OIS	CH2OISN	CH2OIS	CH1OISN	CH1OIS	CH0OISN	CH0OIS
		RW	0	RW	0	RW	0	RW
			0		0		0	

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

## Channel Break Control Register – CHBRKCTR

This register specifies the channel break control bits.

Offset: 0x070

Reset value: 0x0000\_0002

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

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

Bits	Field	Descriptions
[11:8]	BKF	<p>Break Input Filter Setting</p> <p>These bits define the frequency ratio used to sample the MT_BRK signal. The digital filter in the MCTM is an N-event counter where N is defined as how many valid transitions are necessary to output a filtered signal.</p> <p>0000: No filter – do not need sample clock</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>
[5]	CHAOE	<p>Channel Automatic Output Enable</p> <p>0: CHMOE can be set only by software</p> <p>1: CHMOE can be set by software or automatically by an update event 1</p>
[4]	CHMOE	<p>Channel Main Output Enable</p> <p>Cleared asynchronously by hardware on a break event occurrence.</p> <p>0: MT_CHxO and MT_CHxNO are disabled or forced to idle states</p> <p>1: MT_CHxO and MT_CHxNO are enabled if the enable bits (CHxE, CHxNE) are set</p>
[1]	BKP	<p>Break Input Polarity</p> <p>0: Break input is active low</p> <p>1: Break input is active high</p>
[0]	BKE	<p>Break Enable</p> <p>0: Break inputs is disabled</p> <p>1: Break inputs is enabled</p>

## Timer PDMA/Interrupt Control Register – DICTR

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

Offset: 0x074

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
	Reserved					TEVDE	UEV2DE	UEV1DE
Type/Reset						RW 0	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				BRKIE	TEVIE	UEV2IE	UEV1IE
Type/Reset					RW 0	RW 0	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
[25]	UEV2DE	Update event 2 PDMA Request Enable 0: Update event 2 PDMA request is disabled 1: Update event 2 PDMA request is enabled
[24]	UEV1DE	Update event 1 PDMA Request Enable 0: Update event 1 PDMA request is disabled 1: Update event 1 PDMA request is enabled
[19]	CH3CCDE	Channel 3 Capture/Compare PDMA Request Enable 0: Channel 3 PDMA request is disabled 1: Channel 3 PDMA request is enabled
[18]	CH2CCDE	Channel 2 Capture/Compare PDMA Request Enable 0: Channel 2 PDMA request is disabled 1: Channel 2 PDMA request is enabled
[17]	CH1CCDE	Channel 1 Capture/Compare PDMA Request Enable 0: Channel 1 PDMA request is disabled 1: Channel 1 PDMA request is enabled
[16]	CH0CCDE	Channel 0 Capture/Compare PDMA Request Enable 0: Channel 0 PDMA request is disabled 1: Channel 0 PDMA request is enabled
[11]	BRKIE	Break event Interrupt Enable 0: Break event interrupt is disabled 1: Break event interrupt is enabled
[10]	TEVIE	Trigger event Interrupt Enable 0: Trigger event interrupt is disabled 1: Trigger event interrupt is enabled
[9]	UEV2IE	Update event 2 Interrupt Enable 0: Update event 2 interrupt is disabled 1: Update event 2 interrupt is enabled

Bits	Field	Descriptions
[8]	UEV1IE	Update event 1 Interrupt Enable 0: Update event 1 interrupt is disabled 1: Update event 1 interrupt is enabled
[3]	CH3CCIE	Channel 3 Capture/Compare Interrupt Enable 0: Channel 3 interrupt is disabled 1: Channel 3 interrupt is enabled
[2]	CH2CCIE	Channel 2 Capture/Compare Interrupt Enable 0: Channel 2 interrupt is disabled 1: Channel 2 interrupt is enabled
[1]	CH1CCIE	Channel 1 Capture/Compare Interrupt Enable 0: Channel 1 interrupt is disabled 1: Channel 1 interrupt is enabled
[0]	CH0CCIE	Channel 0 Capture/Compare Interrupt Enable 0: Channel 0 interrupt is disabled 1: Channel 0 interrupt is enabled

## Timer Event Generator Register – EVGR

This register contains the software event generation bits.

Offset: 0x078

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved				BRKG	TEVG	UEV2G	UEV1G
	7	6	5	4	3	2	1	0
Type/Reset	Reserved				CH3CCG	CH2CCG	CH1CCG	CH0CCG
					WO	0	WO	0
					WO	0	WO	0
					WO	0	WO	0
					WO	0	WO	0

Bits	Field	Descriptions
[11]	BRKG	Software Break Event Generation The break event BEV can be generated by setting this bit. It is automatically cleared by hardware. 0: No action 1: The BRKIF flag is set and then the CHMOE bit will be cleared
[10]	TEVG	Trigger Event Generation The trigger event TEV can be generated by setting this bit. It is cleared by hardware automatically. 0: No action 1: The TEVIF flag is set
[9]	UEV2G	Update Event 2 Generation The update event 2 UEV2 can be generated by setting this bit. It is cleared by hardware automatically. 0: No action 1: Update the CHxE, CHxNE and CHxOM bits when COMPRES bit in the CTR register is set to 1
[8]	UEV1G	Update Event 1 Generation The update event 1 UEV1 can be generated by setting this bit. It is cleared by hardware automatically. 0: No action 1: Reinitialise the counter The counter value returns to 0 or the CRR preload value, depending on the counter mode in which the current timer is being used. An update operation on any related registers will also be executed. For a more detailed description, refer to the corresponding section.



Bits	Field	Descriptions
[3]	CH3CCG	<p>Channel 3 Capture/Compare Generation</p> <p>A Channel 3 capture/compare event can be generated by setting this bit. It is cleared by hardware automatically.</p> <p>0: No action 1: Capture/compare event is generated on channel 3</p> <p>If Channel 3 is configured as an input, the counter value is captured into the CH3CCR register and then the CH3CCIF bit is set. If Channel 3 is configured as an output, the CH3CCIF bit is set.</p>
[2]	CH2CCG	<p>Channel 2 Capture/Compare Generation</p> <p>A Channel 2 capture/compare event can be generated by setting this bit. It is cleared by hardware automatically.</p> <p>0: No action 1: Capture/compare event is generated on channel 2</p> <p>If Channel 2 is configured as an input, the counter value is captured into the CH2CCR register and then the CH2CCIF bit is set. If Channel 2 is configured as an output, the CH2CCIF bit is set.</p>
[1]	CH1CCG	<p>Channel 1 Capture/Compare Generation</p> <p>A Channel 1 capture/compare event can be generated by setting this bit. It is cleared by hardware automatically.</p> <p>0: No action 1: Capture/compare event is generated on channel 1</p> <p>If Channel 1 is configured as an input, the counter value is captured into the CH1CCR register and then the CH1CCIF bit is set. If Channel 1 is configured as an output, the CH1CCIF bit is set.</p>
[0]	CH0CCG	<p>Channel 0 Capture/Compare Generation</p> <p>A Channel 0 capture/compare event can be generated by setting this bit. It is cleared by hardware automatically.</p> <p>0: No action 1: Capture/compare event is generated on channel 0</p> <p>If Channel 0 is configured as an input, the counter value is captured into the CH0CCR register and then the CH0CCIF bit is set. If Channel 0 is configured as an output, the CH0CCIF bit is set.</p>

## Timer Interrupt Status Register – INTSR

This register stores the timer interrupt status.

Offset: 0x07C

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved				BRKIF	TEVIF	UEV2IF	UEV1IF
	7	6	5	4	3	2	1	0
Type/Reset	CH3OCF	CH2OCF	CH1OCF	CH0OCF	CH3CCIF	CH2CCIF	CH1CCIF	CH0CCIF
	W0C	0	W0C	0	W0C	0	W0C	0

Bits	Field	Descriptions
[11]	BRKIF	Break Event Interrupt Flag This flag is set by hardware when a break event occurs and is cleared by software. 0: No break event occurs 1: Break event occurs
[10]	TEVIF	Trigger Event Interrupt Flag This flag is set by hardware when a trigger event occurs and is cleared by software. 0: No trigger event occurs 1: Trigger event occurs
[9]	UEV2IF	Update Event 2 Interrupt Flag This bit is set by hardware when an update event 2 occurs and is cleared by software. 0: No update event 2 occurs 1: Update event 2 occurs
[8]	UEV1IF	Update Event 1 Interrupt Flag This bit is set by hardware when an update event 1 occurs and is cleared by software. 0: No update event 1 occurs 1: Update event 1 occurs Note: The update event 1 is sourced from the following conditions: - A counter overflow or underflow - The UEV1G bit is set with UEV1DIS = 0 - An STI rising edge is received in slave restart mode with UEV1DIS = 0
[7]	CH3OCF	Channel 3 Over-capture Flag This flag is set by hardware and cleared by software. 0: No over-capture event is detected 1: Capture event occurs again when the CH3CCIF bit is already set and it is not yet cleared by software

Bits	Field	Descriptions
[6]	CH2OCF	<p>Channel 2 Over-capture Flag</p> <p>This flag is set by hardware and cleared by software.</p> <p>0: No over-capture event is detected</p> <p>1: Capture event occurs again when the CH2CCIF bit is already set and it is not cleared yet by software</p>
[5]	CH1OCF	<p>Channel 1 Over-capture Flag</p> <p>This flag is set by hardware and cleared by software.</p> <p>0: No over-capture event is detected</p> <p>1: Capture event occurs again when the CH1CCIF bit is already set and it is not cleared yet by software</p>
[4]	CH0OCF	<p>Channel 0 Over-capture Flag</p> <p>This flag is set by hardware and cleared by software.</p> <p>0: No over-capture event is detected</p> <p>1: Capture event occurs again when the CH0CCIF bit is already set and it is not yet cleared by software</p>
[3]	CH3CCIF	<p>Channel 3 Capture/Compare Interrupt Flag</p> <p>- Channel 3 is configured as an output</p> <p>0: No match event occurs</p> <p>1: The contents of the counter CNTR have matched the contents of the CH3CCR register</p> <p>This flag is set by hardware when the counter value matches the CH3CCR value with exception in the center-aligned counting mode. It is cleared by software.</p> <p>- Channel 3 is configured as an input</p> <p>0: No input capture occurs</p> <p>1: Input capture occurs</p> <p>This bit is set by hardware when a capture event occurs. It is cleared by software or by reading the CH3CCR register</p>
[2]	CH2CCIF	<p>Channel 2 Capture/Compare Interrupt Flag</p> <p>- Channel 2 is configured as an output</p> <p>0: No match event occurs</p> <p>1: The contents of the counter CNTR have matched the contents of the CH2CCR register</p> <p>This flag is set by hardware when the counter value matches the CH2CCR value with exception in the center-aligned counting mode. It is cleared by software.</p> <p>- Channel 2 is configured as an input</p> <p>0: No input capture occurs</p> <p>1: Input capture occurs</p> <p>This bit is set by hardware on a capture event. It is cleared by software or by reading the CH2CCR register.</p>
[1]	CH1CCIF	<p>Channel 1 Capture/Compare Interrupt Flag</p> <p>- Channel 1 is configured as an output</p> <p>0: No match event occurs</p> <p>1: The contents of the counter CNTR have matched the contents of the CH1CCR register</p> <p>This flag is set by hardware when the counter value matches the CH1CCR value with exception in the center-aligned counting mode. It is cleared by software.</p> <p>- Channel 1 is configured as an input</p> <p>0: No input capture occurs</p> <p>1: Input capture occurs</p> <p>This bit is set by hardware on a capture event. It is cleared by software or by reading the CH1CCR register.</p>

Bits	Field	Descriptions
[0]	CH0CCIF	<p>Channel 0 Capture/Compare Interrupt Flag</p> <ul style="list-style-type: none"> <li>- Channel 0 is configured as an output <ul style="list-style-type: none"> <li>0: No match event occurs</li> <li>1: The contents of the counter CNTR have matched the content of the CH0CCR register</li> </ul> </li> </ul> <p>This flag is set by hardware when the counter value matches the CH0CCR value with exception in the center-aligned counting mode. It is cleared by software.</p> <ul style="list-style-type: none"> <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> </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>

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

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> $f_{CK\_CNT} = \frac{f_{CK\_PSC}}{PSCV[15:0] + 1}, \text{ where the } f_{CK\_PSC} \text{ is the prescaler input clock source.}$

## 15 Motor Control Timer (MCTM)

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

## Timer Repetition Register – REPR

Offset:	0x08C
---------	-------

Reset value: 0x0000 0000

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

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

## 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
	7	6	5	4	3	2	1	0
Type/Reset	CH1CCV							
	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[15:0]	CH1CCV	<p>Channel 1 Capture/Compare Value</p> <ul style="list-style-type: none"> <li>- When Channel 1 is configured as an output 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
	7	6	5	4	3	2	1	0
Type/Reset	CH2CCV							
	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[15:0]	CH2CCV	<p>Channel 2 Capture/Compare Value</p> <ul style="list-style-type: none"> <li>- When Channel 2 is configured as an output 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
<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>							
[15:0]	CH0ACV	Channel 0 Asymmetric Compare Value When channel 0 is configured as asymmetric PWM mode and the counter is counting down, the value written into this register will be compared to the counter.							

## Channel 1 Asymmetric Compare Register – CH1ACR

This register specifies the timer channel 1 asymmetric compare value.

Offset: 0x0A4

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
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
<b>Bits</b>	<b>Field</b>	<b>Descriptions</b>							
[15:0]	CH1ACV	Channel 1 Asymmetric Compare Value When channel 1 is configured as asymmetric PWM mode and the counter is counting down, the value written into this register will be compared to the counter.							

## Channel 2 Asymmetric Compare Register – CH2ACR

This register specifies the timer channel 2 asymmetric compare value.

Offset: 0x0A8

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
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
	7	6	5	4	3	2	1	0
Type/Reset	CH2ACV							
	RW	0	RW	0	RW	0	RW	0

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

## Channel 3 Asymmetric Compare Register – CH3ACR

This register specifies the timer channel 3 asymmetric compare value.

Offset: 0x0AC

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
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
	7	6	5	4	3	2	1	0
Type/Reset	CH3ACV							
	RW	0	RW	0	RW	0	RW	0

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

# 16 Single-Channel Timer (SCTM)

## Introduction

The Single-Channel Timer consists of one 16-bit up-counter, one 16-bit Capture/Compare 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 PWM output.

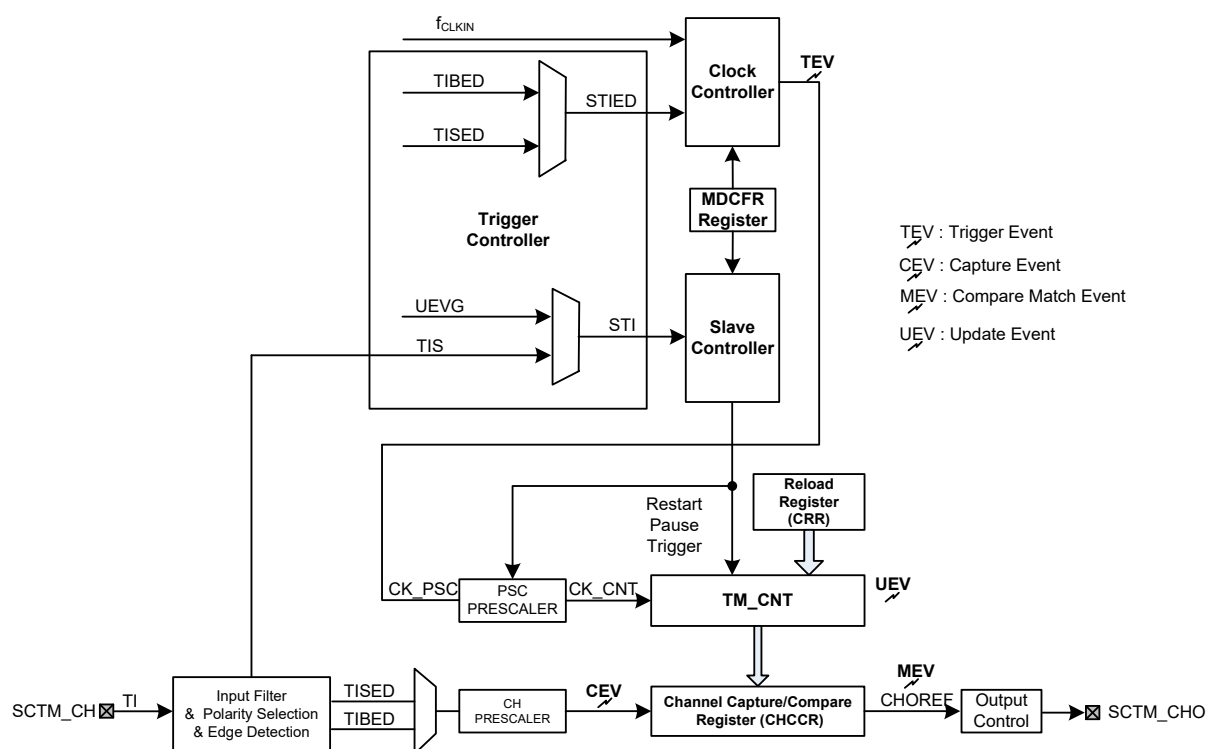


Figure 116. SCTM Block Diagram

## Features

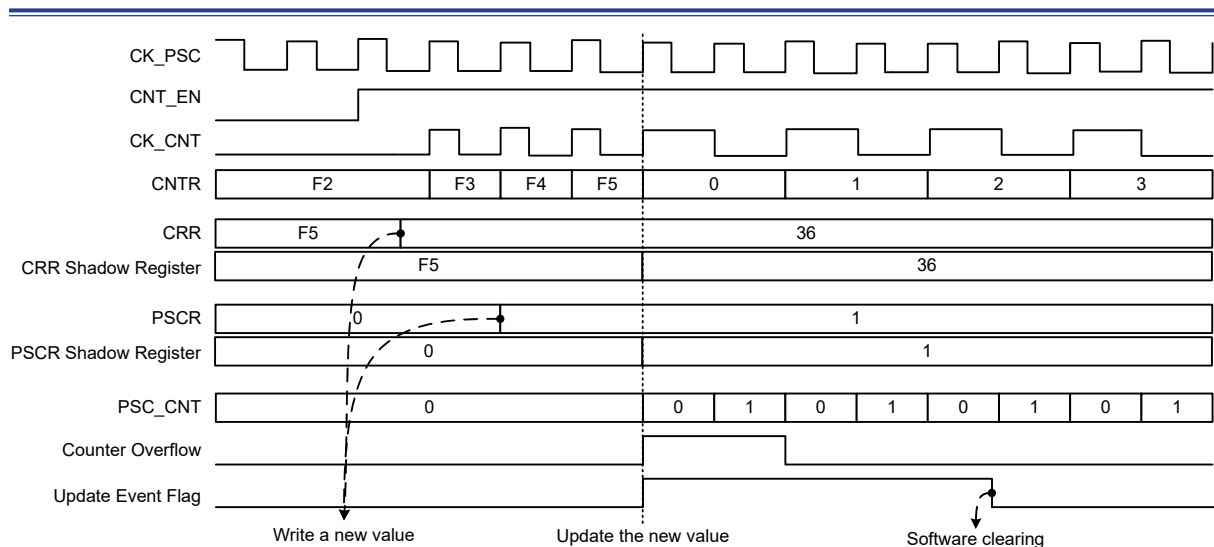
- 16-bit auto-reload up-counter
- 16-bit programmable prescaler that allows division of the prescaler clock source by any factor between 1 and 65536 to generate the counter clock frequency
- Single channel for:
  - Input Capture function
  - Compare Match Output
  - PWM waveform output
- Interrupt generation 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, 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 117. Up-Counting Example**

## Clock Controller

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

### ■ Internal APB clock $f_{CLKIN}$

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

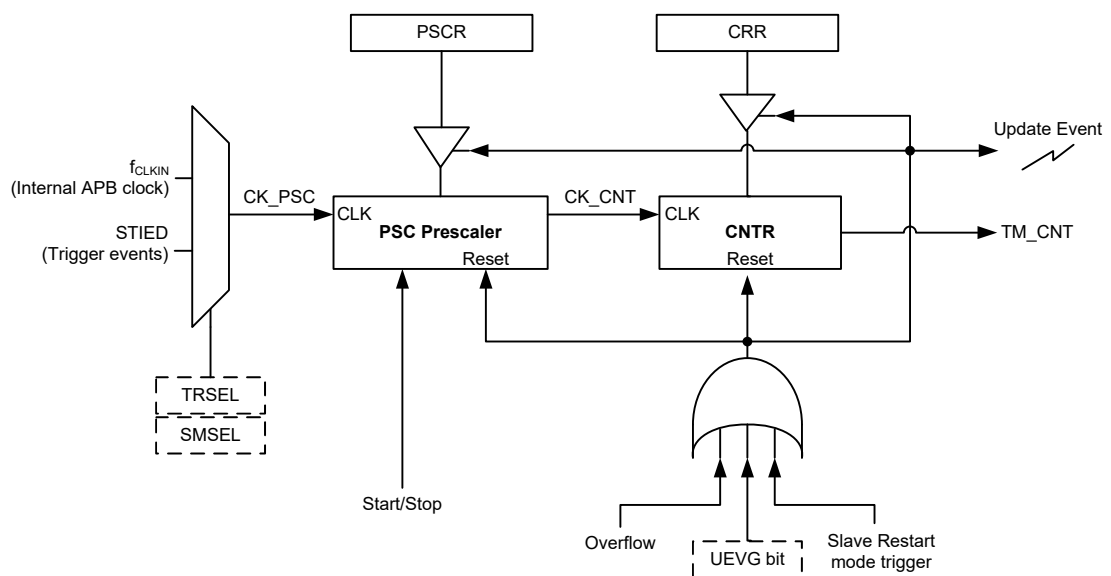
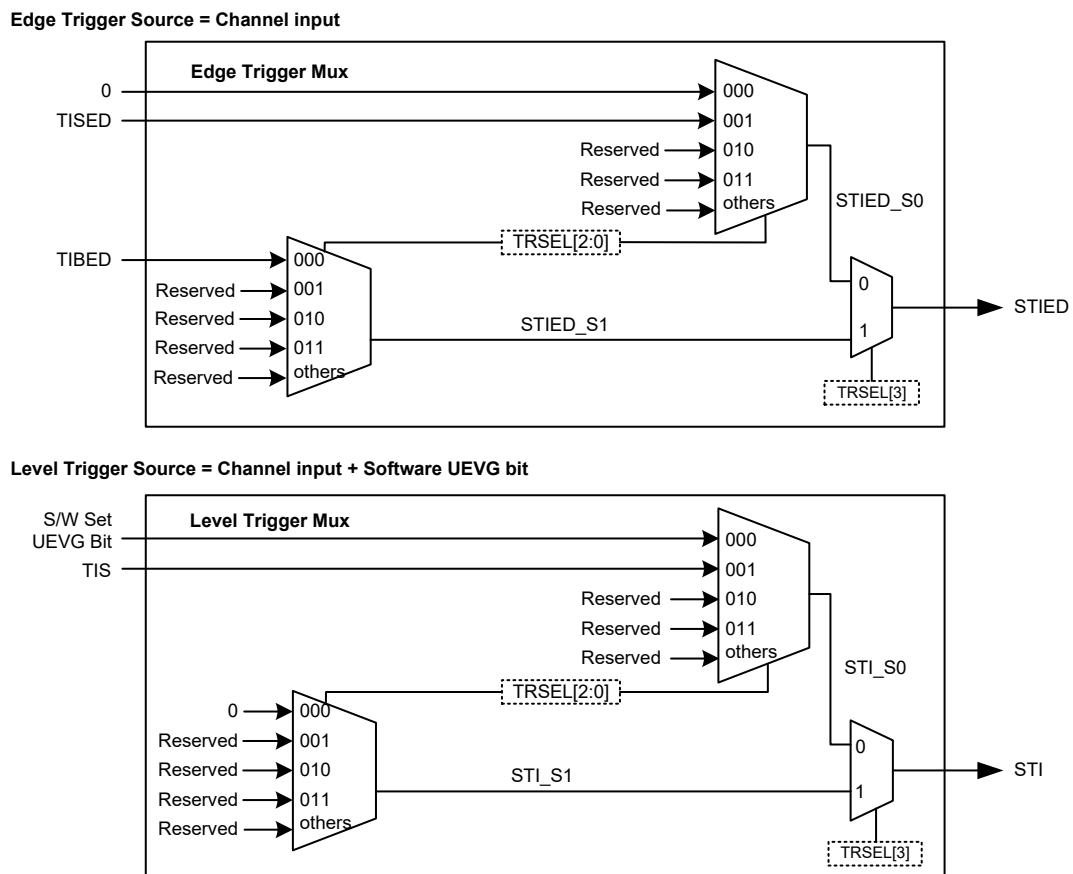


Figure 118. SCTM Clock Source Selection

## 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 119. Trigger Controller Block**



## Slave Controller

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

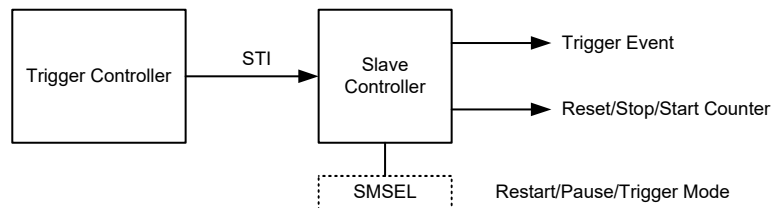


Figure 120. Slave Controller Diagram

### Restart Mode

The counter and its prescaler can be reinitialized in response to a rising edge of the STI signal. When 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.

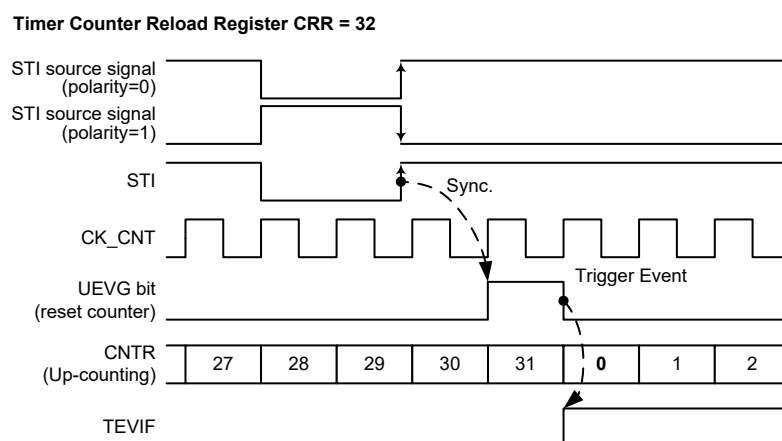


Figure 121. SCTM in Restart Mode

### Pause Mode

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

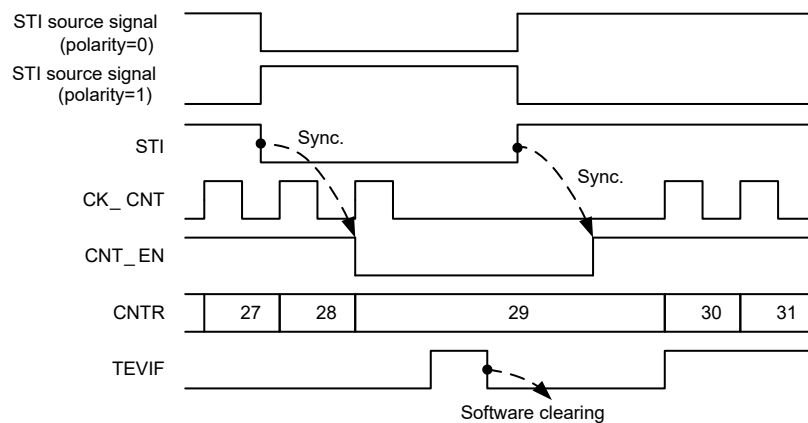


Figure 122. SCTM in Pause Mode

### Trigger Mode

After the counter is disabled to count, the counter can resume counting when 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.

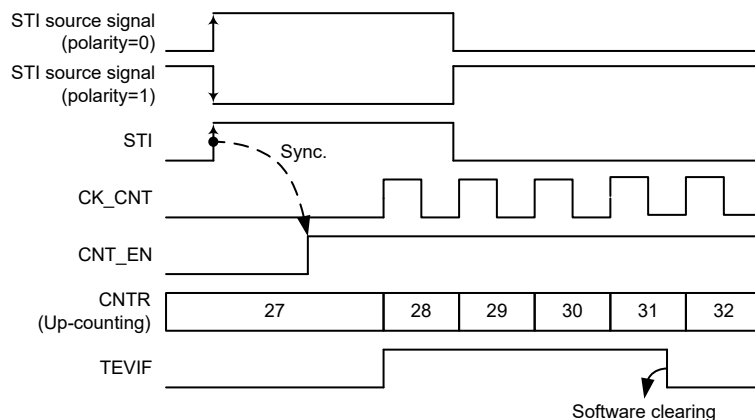


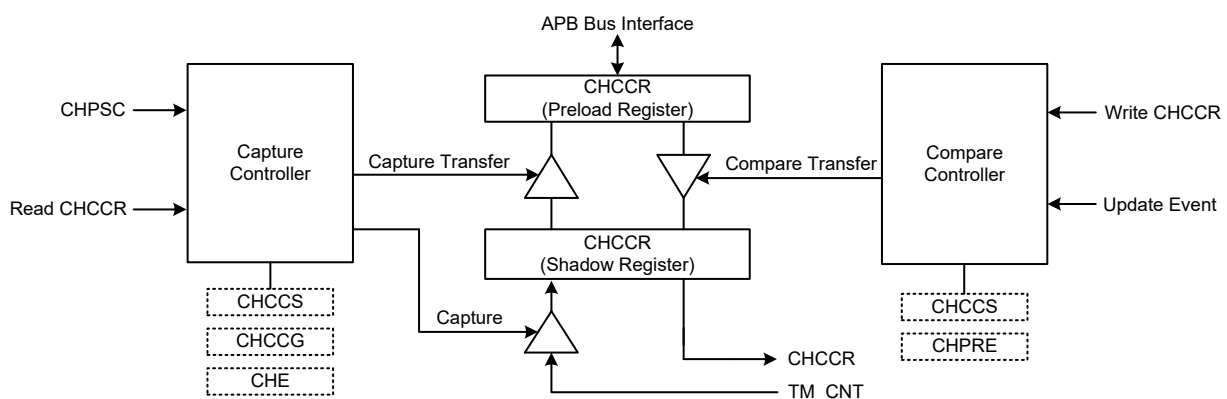
Figure 123. SCTM in Trigger Mode

## Channel Controller

The SCTM channel can be used as the capture input or compare match output. The capture input or compare match output channel is composed of a preload register and a shadow register. Data access of the APB bus is always 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.

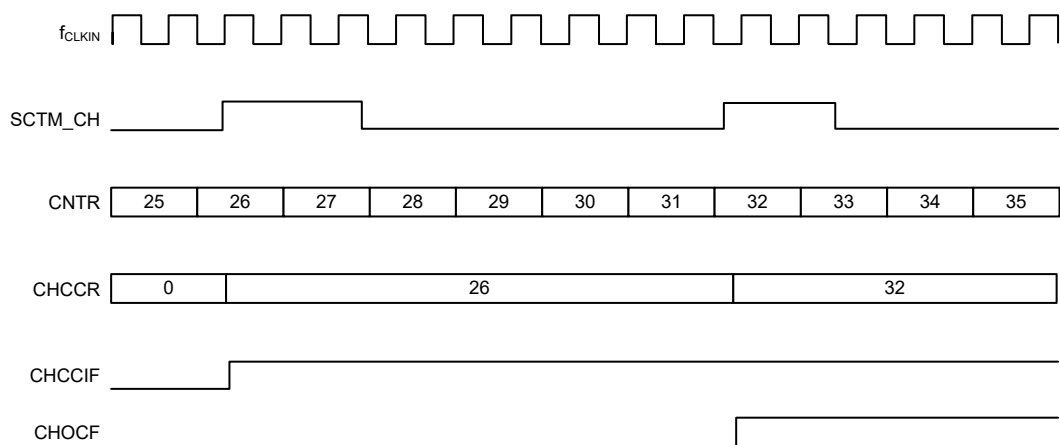
When used in the compare match output mode, the contents of the CHCCR preload register is copied into the associated shadow register, the counter value is then compared with the register value.



**Figure 124. Capture/Compare Block Diagram**

### Capture Counter Value Transferred to CHCCR

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



**Figure 125. Input Capture Mode**

## Input Stage

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

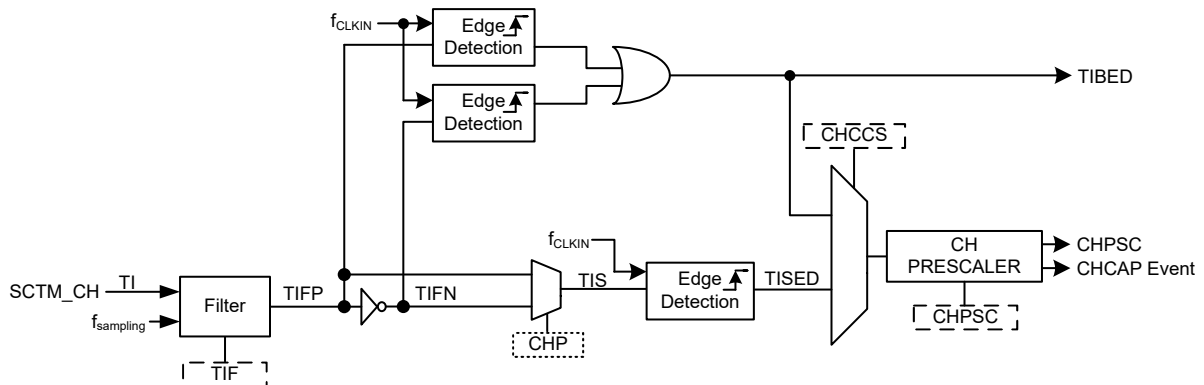


Figure 126. Channel Input Stages

## 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.

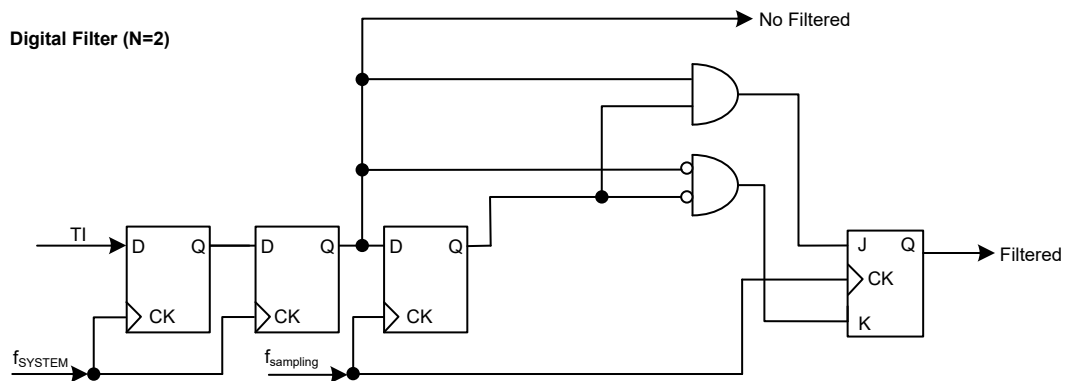


Figure 127. TI Digital Filter Diagram with N = 2

## 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.

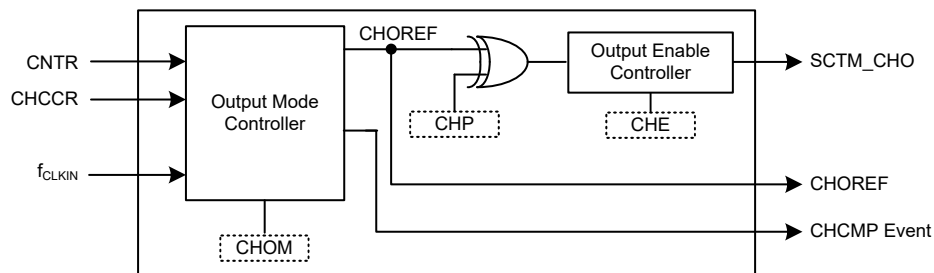


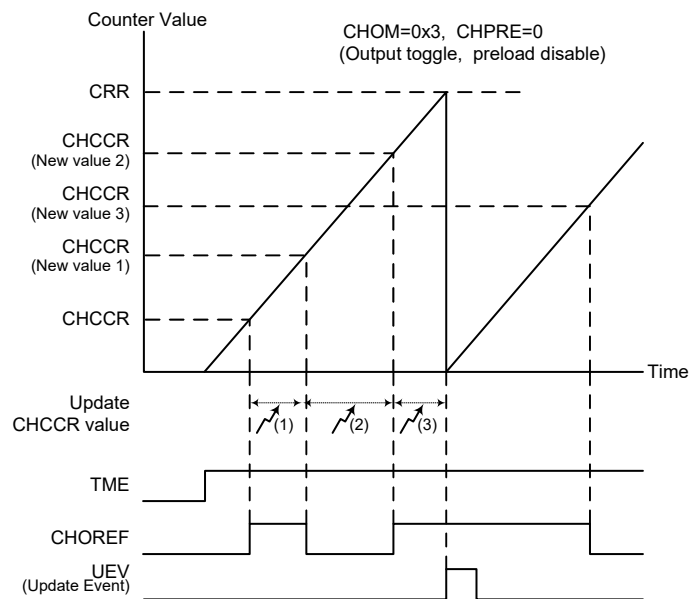
Figure 128. Output Stage Block Diagram

## Channel Output Reference Signal

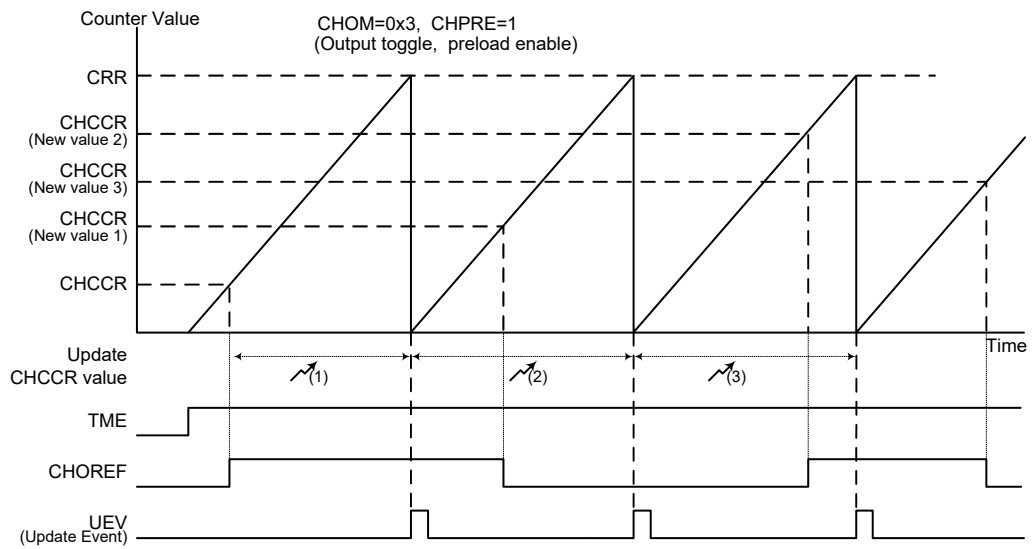
When the SCTM is used in the compare match output mode, the CHOREF signal (Channel Output Reference signal) is defined by the CHOM bit setup. The CHOREF signal has several types of output function which defines what happens to the output when the counter value matches the contents of the CHCCR register.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

Table 40. Compare Match Output Setup

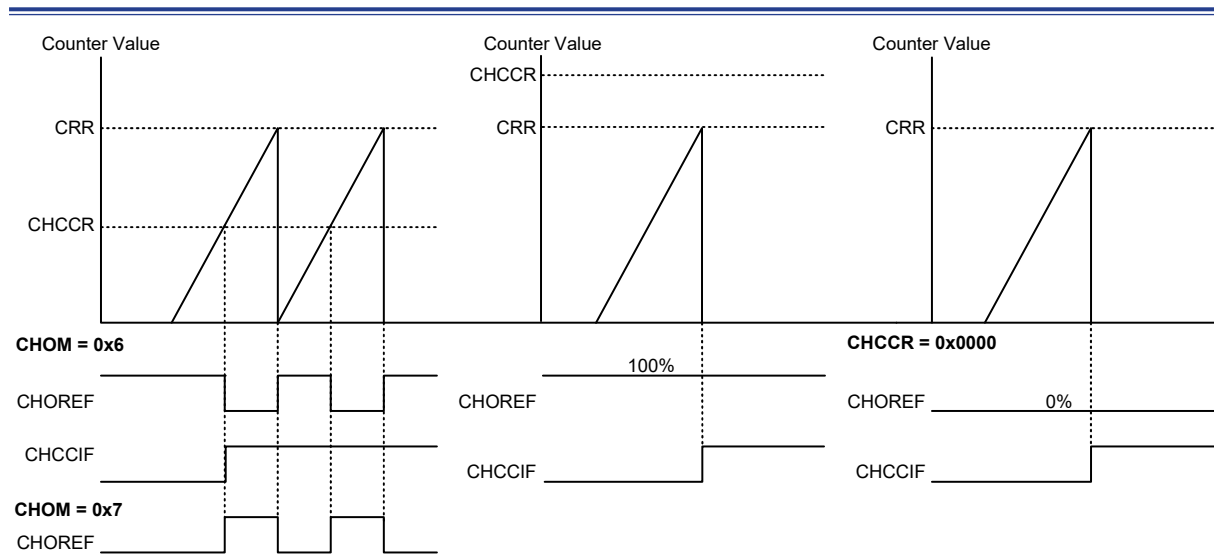
CHOM Value	Compare Match Level
0x0	No change
0x1	Clear Output to 0
0x2	Set Output to 1
0x3	Toggle Output
0x4	Force Inactive Level
0x5	Force Active Level
0x6	PWM Mode 1
0x7	PWM Mode 2



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



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



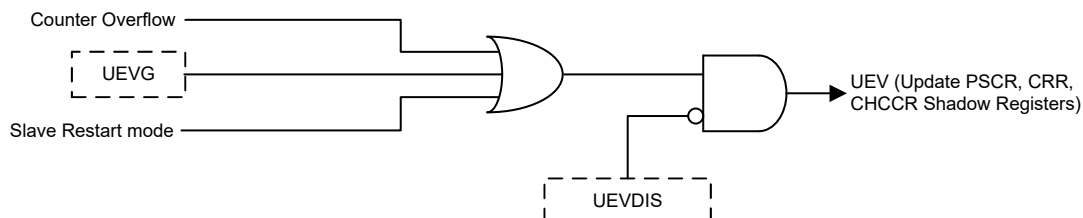
**Figure 131. PWM Mode Channel Output Reference Signal**

## Update Management

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

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

#### Update Event Management



#### Update Event Interrupt Management

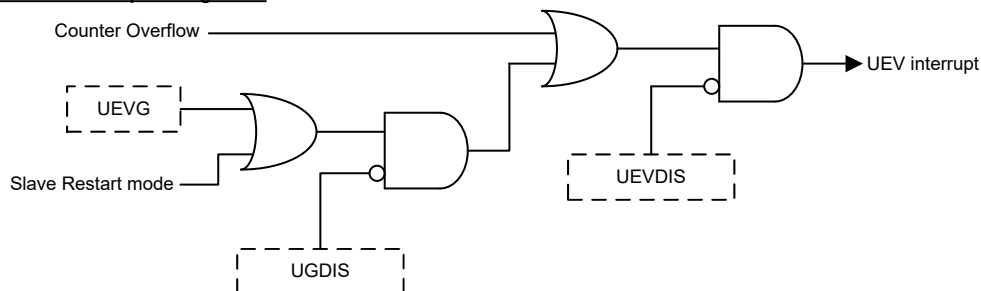


Figure 132. Update Event Setting Diagram

## Register Map

The following table shows the SCTM registers and reset values.

Table 41. SCTM Register Map

Register	Offset	Description	Reset Value
CNTCFR	0x000	Timer Counter Configuration Register	0x0000_0000
MDCFR	0x004	Timer Mode Configuration Register	0x0000_0000
TRCFR	0x008	Timer Trigger Configuration Register	0x0000_0000
CTR	0x010	Timer Control Register	0x0000_0000
CHICFR	0x020	Channel Input Configuration Register	0x0000_0000
CHOCFR	0x040	Channel Output Configuration Register	0x0000_0000
CHCTR	0x050	Channel Control Register	0x0000_0000
CHPOLR	0x054	Channel Polarity Configuration Register	0x0000_0000
DICTR	0x074	Timer Interrupt Control Register	0x0000_0000
EVGR	0x078	Timer Event Generator Register	0x0000_0000
INTSR	0x07C	Timer Interrupt Status Register	0x0000_0000
CNTR	0x080	Timer Counter Register	0x0000_0000
PSCR	0x084	Timer Prescaler Register	0x0000_0000
CRR	0x088	Timer Counter-Reload Register	0x0000_FFFF
CHCCR	0x090	Channel Capture/Compare Register	0x0000_0000



## Timer Counter Configuration Register – CNTCFR

Reset value: 0x0000 0000

Bits	Field	Descriptions
[9:8]	CKDIV	<p>Clock Division</p> <p>These two bits define the frequency ratio between the timer clock (<math>f_{CLKIN}</math>) and the dead-time clock (<math>f_{DTS}</math>). The dead-time clock is also used for digital filter sampling clock.</p> <p>00: <math>f_{DTS} = f_{CLKIN}</math>            01: <math>f_{DTS} = f_{CLKIN} / 2</math>            10: <math>f_{DTS} = f_{CLKIN} / 4</math>            11: Reserved</p>
[1]	UGDIS	<p>Update event interrupt generation disable control</p> <p>0: Any of the following events will generate an update interrupt</p> <ul style="list-style-type: none"> <li>- Counter overflow</li> <li>- Setting the UEVG bit</li> <li>- Update generation through the slave mode</li> </ul> <p>1: Only counter overflow generates an update interrupt</p>
[0]	UEVDIS	<p>Update event Disable control</p> <p>0: Enable the update event request by one of following events:</p> <ul style="list-style-type: none"> <li>- Counter overflow</li> <li>- Setting the UEVG bit</li> <li>- Update generation through the slave mode</li> </ul> <p>1: Disable the update event (However the counter and the prescaler are reinitialized if the UEVG bit is set or if a hardware restart is received from the slave mode)</p>

## Timer Mode Configuration Register – MDCFR

This register specifies the SCTM slave mode selection.

Offset: 0x004

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/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 slave trigger mode. In this mode 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
								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	<p>Channel Capture/Compare Enable</p> <ul style="list-style-type: none"> <li>- Channel is configured as an input (CHCCS = 0x1/0x3) <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 is configured as an output (CHCCS = 0x0) <ul style="list-style-type: none"> <li>0: Off – Channel output signal CHO is not active</li> <li>1: On – Channel output signal CHO generated on the corresponding output pin</li> </ul> </li> </ul>



## Channel Polarity Configuration Register – CHPOLR

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

Offset: 0x054

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/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

This register contains the timer interrupt enable control bits.

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
Type/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	
						RW	0	RW    0	
	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. 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
						W0C 0		W0C 0
	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>

# 17 Basic Function Timer (BFTM)

## Introduction

The Basic Function Timer 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.

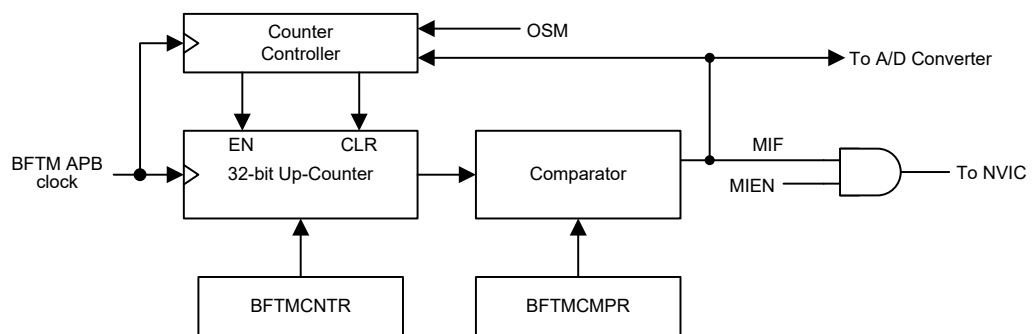


Figure 133. BFTM Block Diagram

## 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

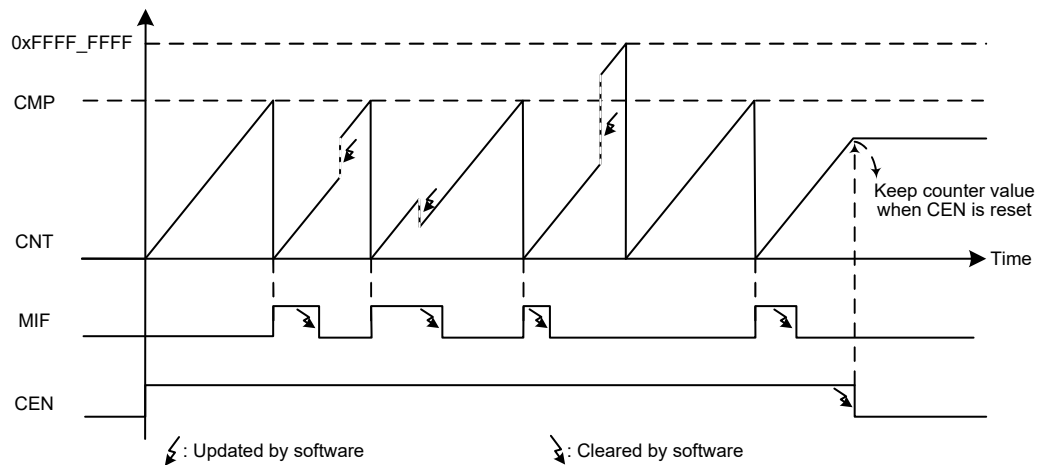
The BFTM is a 32-bit up-counting counter which is driven by the BFTM APB clock, PCLK. The counter value can be changed or read at any time even when the timer is counting. The BFTM supports two operating modes known as the repetitive mode and one shot mode allowing the measurement of time intervals or 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 BFTMCMR register. When the BFTM operates in the repetitive mode and the counter reaches a value equal to the specific compare value in the BFTMCMR register, the timer will generate a



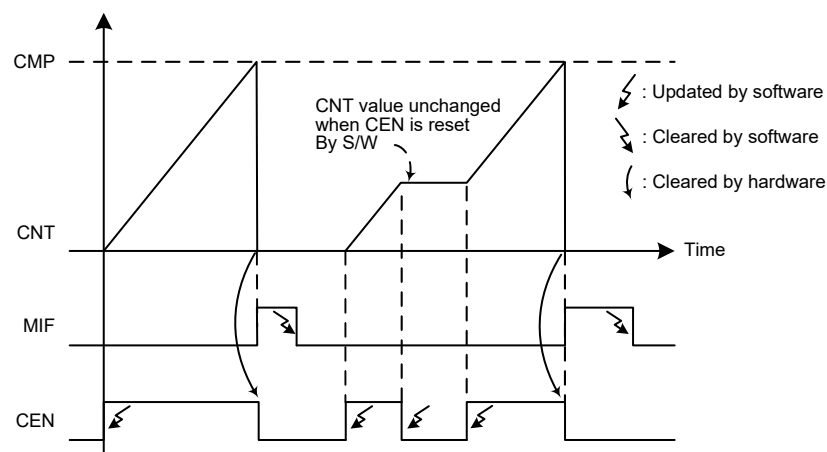
compare match event signal, MIF. When this occurs, the counter will be reset to 0 and resume its counting operation. When the MIF signal is generated, a BFTM compare match interrupt will also be generated periodically if the compare match interrupt is enabled by setting the corresponding interrupt control 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.



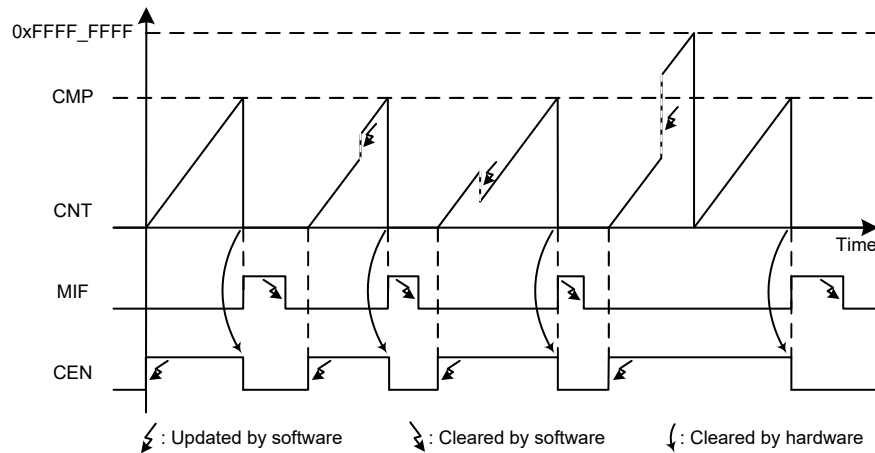
**Figure 134. BFTM – Repetitive Mode**

### 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 135. BFTM – One Shot Mode**



**Figure 136. 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 reset values.

**Table 42. BFTM Register Map**

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

## Register Descriptions

### BFTM Control Register – BFTMCR

This register specifies the overall BFTM control bits.

Offset: 0x000

Reset value: 0x0000\_0000

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

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

## BFTM Status Register – BFTMSR

This register specifies the BFTM status.

Offset: 0x004

Reset value: 0x0000\_0000

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

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

## BFTM Counter Value Register – BFTMCNTR

This register specifies the BFTM counter value.

Offset: 0x008

Reset value: 0x0000\_0000

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

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

## BFTM Compare Value Register – BFTMCMPR

The register specifies the BFTM compare value.

Offset: 0x00C

Reset value: 0xFFFF\_FFFF

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

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

# 18 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. 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.

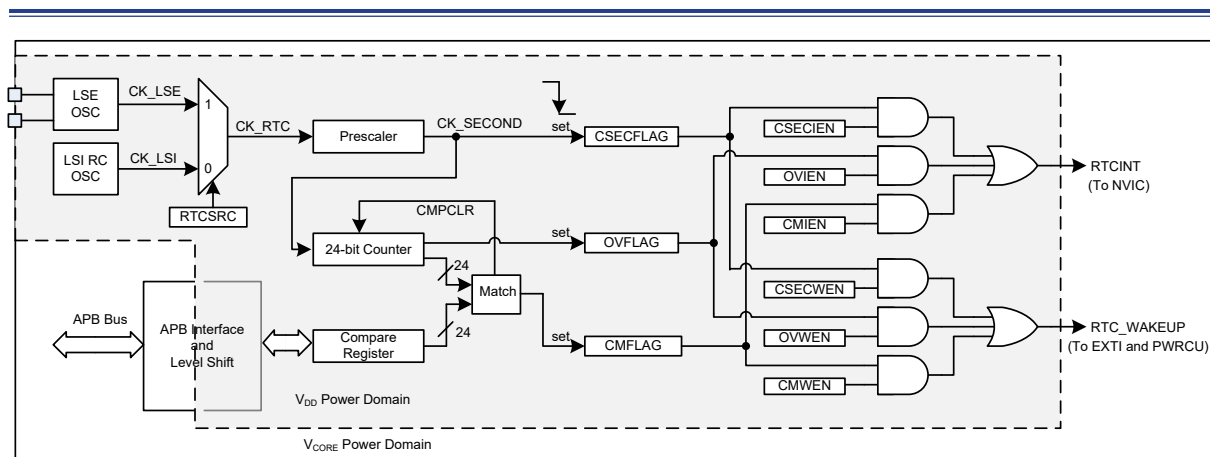


Figure 137. RTC Block Diagram

## Features

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

## Functional Descriptions

### RTC Related Register Reset

Most of 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 PWCURST 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 saving modes 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.

### 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 43. 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 MCU 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 RTCOUNT output format according to the mode, polarity, and event selection setting.

**Table 44. RTCOUNT Output Mode and Active Level Setting**

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

$T_R$ : RTCOUNT 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 45. 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_0F00
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

	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 PWCURST bit in the PWRCR register</li> <li>- V<sub>DD</sub> Domain power on reset – POR</li> <li>- Compare match (RTCCNT = RTCCMP) when CMPCLR = 1 (in the RTCCR register)</li> <li>- RTCEN bit changed from 0 to 1</li> </ul>

## RTC Compare Register – RTCCMP

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

Address: 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	RTCCMPV								
	15	14	13	12	11	10	9	8	
Type/Reset	RTCCMPV								
	7	6	5	4	3	2	1	0	
Type/Reset	RTCCMPV								

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

## RTC Control Register – RTCCR

This register specifies a range of RTC circuitry control bits.

Address: 0x008

Reset value: 0x0000\_0F00

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

Bits	Field	Descriptions
[20]	ROLF	<p>RTCOUNT Level Mode Flag</p> <p>0: RTCOUT Output is inactive</p> <p>1: RTCOUT Output is holding as active level</p> <p>Set by hardware when in the level mode (ROWM = 1) and a RTCOUT output event occurs. Cleared by software reading this flag. The RTCOUT signal will return to the inactive level after software has read this bit.</p>
[19]	ROAP	<p>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 RTCOUT 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 RTCOUT signal is output on the RTCOUT pin when a RTC compare match event or the RTC second clock (CK_SECOND) event occurs.</p>
[16]	ROEN	<p>RTCOUNT Output Pin Enable</p> <p>0: Disable RTCOUT output pin</p> <p>1: Enable RTCOUT output pin</p> <p>When the ROEN bit is set to 1, the RTCOUT signal will be at an active level once a RTC compare match 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 RTCOUT 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 (It can also be reset by changing the RTCEN bit 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 0xFF_FFFF to 0x00_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

	31	30	29	28	27	26	25	24
Type/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
						OVWEN	CMWEN	CSECWEN
						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]	OVWEN	Counter Overflow Interrupt Enable 0: Counter Overflow Interrupt is disabled 1: Counter Overflow Interrupt is enabled
[1]	CMWEN	Compare Match Interrupt Enable 0: Compare Match Interrupt is disabled 1: Compare Match Interrupt is enabled
[0]	CSECWEN	Counter Clock CK_SECOND Interrupt Enable 0: Counter Clock CK_SECOND Interrupt is disabled 1: Counter Clock CK_SECOND Interrupt is enabled

# 19 Watchdog Timer (WDT)

## Introduction

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

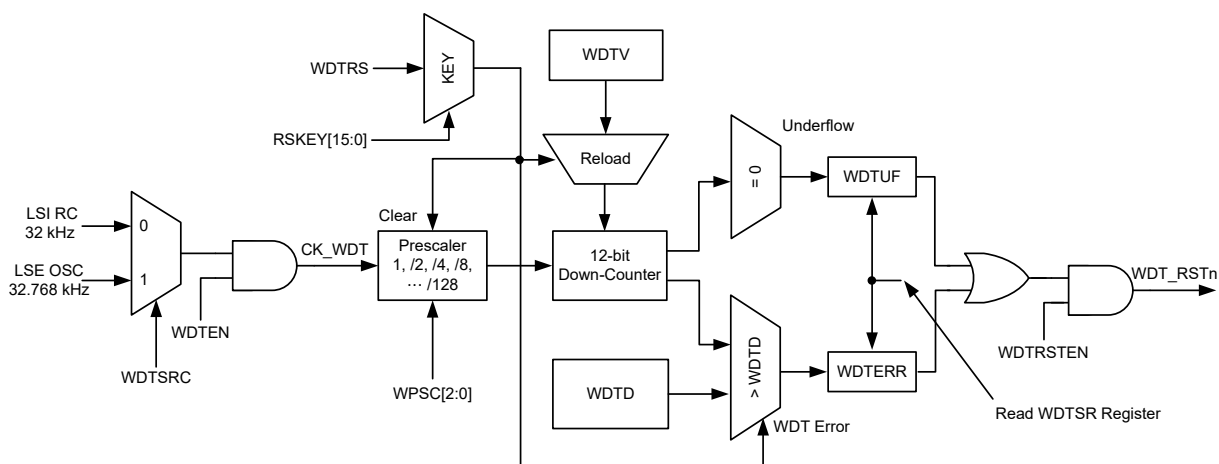


Figure 138. Watchdog Timer Block Diagram

## 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.

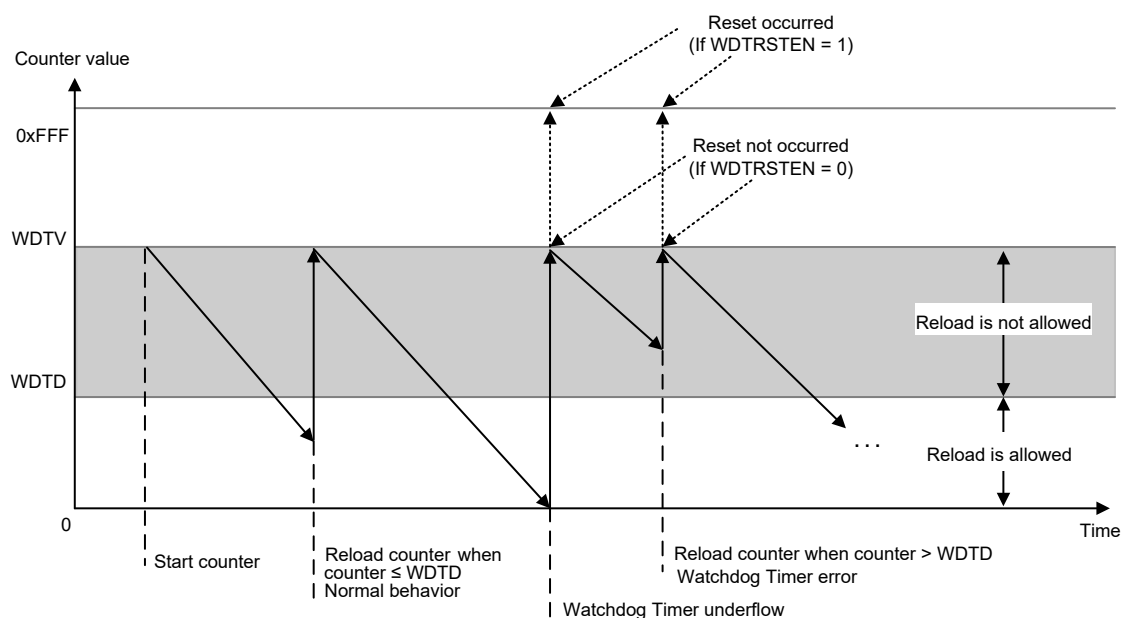
When the system enters the Sleep mode or Deep-Sleep1 mode, the Watchdog Timer counter will either continue to count or stop depending on the WDTSHLT field setup in the WDTMR0 register. However, the Watchdog Timer will always stop when the system is in the Deep-Sleep2 mode. When the Watchdog stops counting, 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 139. Watchdog Timer Behavior**

## Register Map

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

**Table 46. 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 stops counting, the count value is retained so that it continues counting after the system wakes up from these three sleep modes. If a Watchdog reset occurs in the Sleep or Deep-Sleep1 mode, it will wake up the device.
[13]	WDRSTEN	Watchdog Timer Reset Enable 0: A Watchdog Timer underflow or error has no effect on the system reset 1: A Watchdog Timer underflow or error triggers a Watchdog Timer system reset
[11:0]	WDTV	Watchdog Timer Counter Value WDTV defines the value loaded into the 12-bit Watchdog down-counter.

## Watchdog Timer Mode Register 1 – WDTMR1

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

Offset: 0x008

Reset value: 0x0000\_7FFF

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

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

## Watchdog Timer Status Register – WDTSR

This register specifies the Watchdog timer status.

Offset: 0x00C

Reset value: 0x0000\_0000

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

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

## Watchdog Timer Protection Register – WDTPR

This register specifies the Watchdog timer protect key configuration.

Offset: 0x010

Reset value: 0x0000\_0000

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

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

## Watchdog Timer Clock Selection Register – WDTCSR

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

Offset: 0x018

Reset value: 0x0000\_0000

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

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

## 20 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: 100 kHz in the Standard mode, 400 kHz in the Fast mode and 1 MHz in the Fast mode plus. The SCL period generation registers are used to set different kinds of duty cycle implementation for the SCL pulse.

The SDA line which is connected to the whole I<sup>2</sup>C bus is a bidirectional 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.

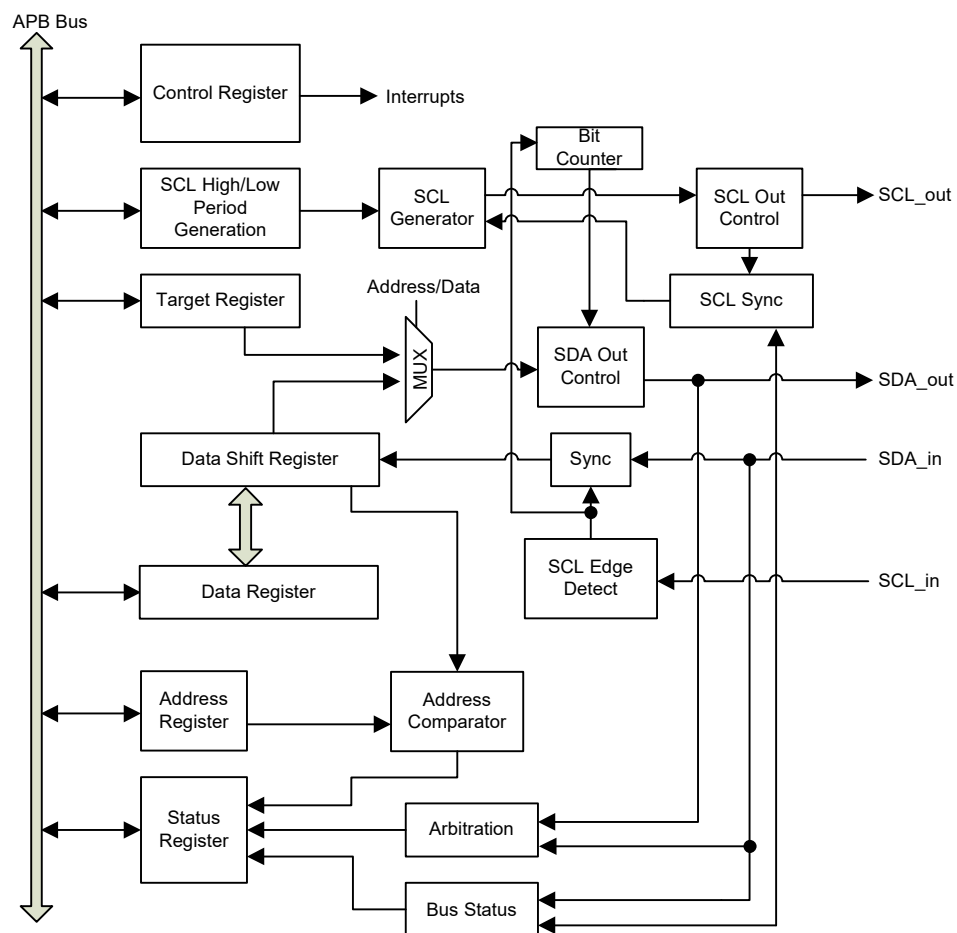


Figure 140. I<sup>2</sup>C Module Block Diagram



## 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
- Bidirectional data transfer between master and slave
- Multi-master bus – no central master
  - The same interface can act as Master or Slave
- Arbitration among simultaneous transmitting masters without corrupting serial data on the bus
- Clock synchronization
  - Allow devices with different bit rates to communicate via one serial bus
- Supports 7-bit and 10-bit addressing modes and general call addressing
- Multiple slave addresses using address mask function
- Timeout function
- Supports PDMA Interface, the PDMA related describes are only available for the HT32F54243/HT32F54253 devices

## Functional Descriptions

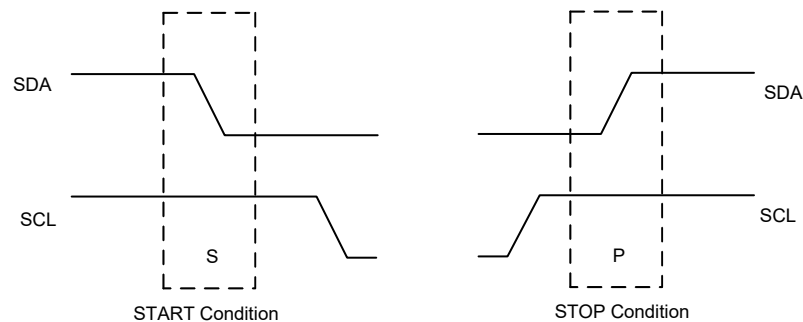
### Two-Wire Serial Interface

The I<sup>2</sup>C module has two external lines, the serial data SDA and serial clock SCL lines, to carry information between the interconnected devices connected to the bus. The SCL and SDA lines are both bidirectional and must be connected to a pull-high resistor. When the I<sup>2</sup>C bus is in the free or idle state, both pins are at a high level to perform the required wired-AND function for multiple connected devices.

### START and STOP Conditions

A master device can initialize a transfer by sending a START signal and terminate the transfer with a STOP signal. A START signal is usually referred to as the “S” bit, which is defined as a High to Low transition on the SDA line while the SCL line is high. A STOP signal is usually referred to as the “P” bit, which is defined as a Low to High transition on the SDA line while SCL is high.

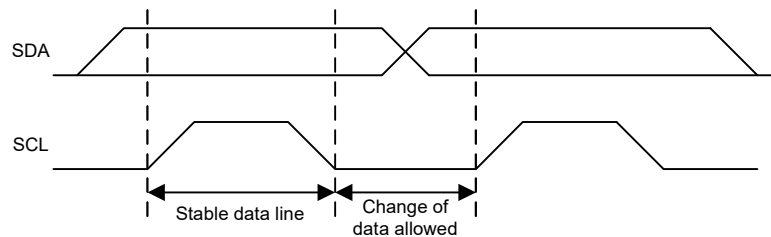
A repeated START signal, which is denoted as the “Sr” bit, is functionally identical to the normal START condition. A repeated START signal allows the I<sup>2</sup>C interface to communicate with another slave device or with the same device but in a different transfer direction without releasing the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



**Figure 141. 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 142. 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

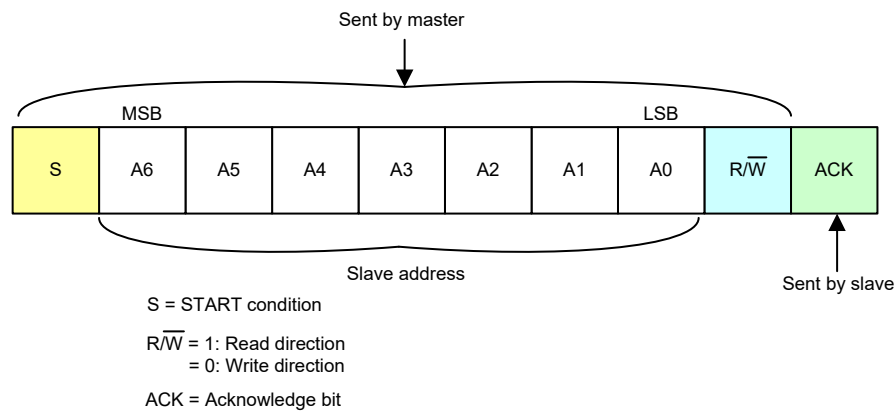
The 7-bit address format is composed of the 7-bit length slave address, which the master device wants to communicate with, a  $R/\overline{W}$  bit and an ACK bit. The  $R/\overline{W}$  bit defines the direction of the data transfer.

$R/\overline{W} = 0$  (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 the master.

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



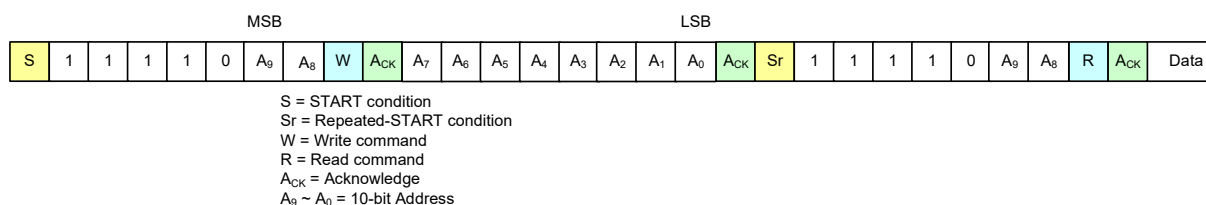
**Figure 143. 7-bit Addressing Mode**

### 10-bit Address Format

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



**Figure 144. 10-bit Addressing Write Transmit Mode**



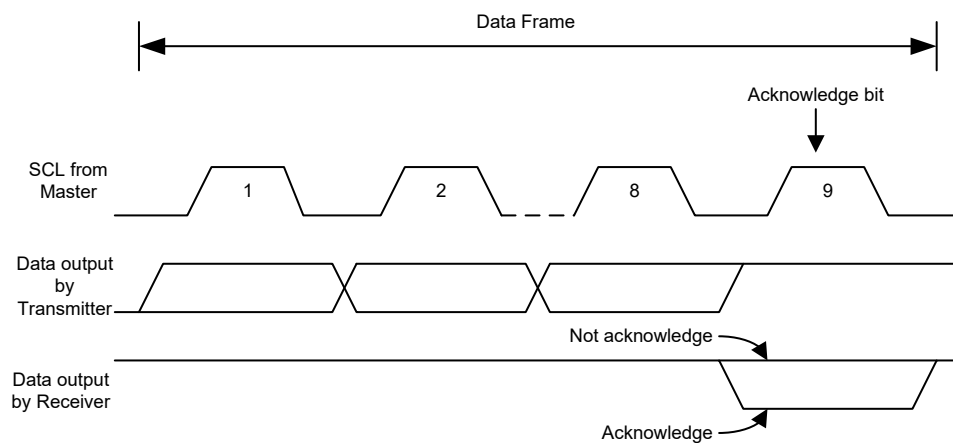
**Figure 145. 10-bits Addressing Read Receive Mode**

## Data Transfer and Acknowledge

Once the slave device address has been matched, the data can be transmitted to or received from the slave device according to the transfer direction specified by the R/ $\overline{W}$  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.

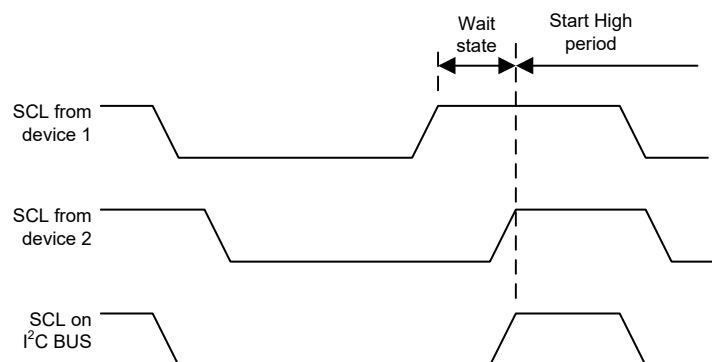
If the master device sends a Not-Acknowledge (NACK) signal to the slave device, the slave device should release the SDA line for the master device to generate a STOP signal to terminate the transfer.



**Figure 146. 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.

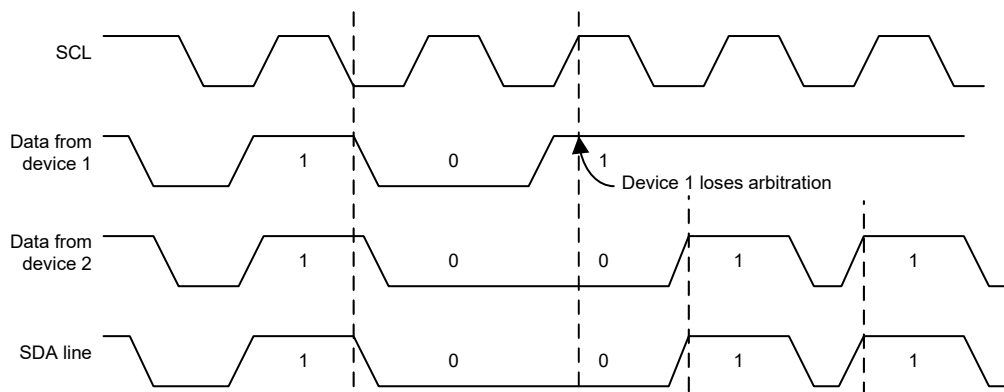


**Figure 147. Clock Synchronization during Arbitration**

## 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 148. Two Masters Arbitration Procedure**

## General Call Addressing

The general call addressing function can be used to address all the devices connected to the I<sup>2</sup>C bus. The master device can activate the general call function by writing a value “0x00” into the TAR field 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 an address mask function for users to decide which address bit can be ignored during the comparison with the address frame sent from the master. The ADRS flag will be asserted when the unmasked address bits and the address frame sent from the master are matched. Note that this function is only available in the slave mode.

For instance, the user sets a data transfer with the 7-bit addressing mode together with the I2CADDR register value as 0x05 and the I2CADDR register value as 0x55, this means if an address which is sent by an 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 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.

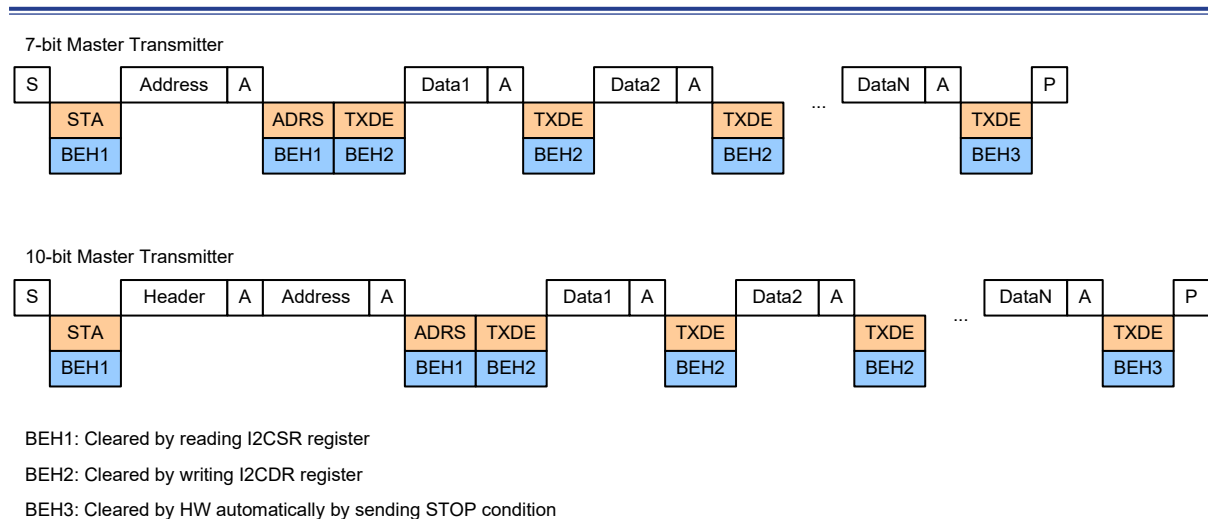
#### Data Frame

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 149. Master Transmitter Timing Diagram**

## Master Receiver Mode

### Start Condition

The target slave device address and communication direction must be written into the I2CTAR register. The STA flag in the I2CSR register is set by hardware after a start condition occurs. In order to send the following address frame, the STA flag must be cleared to 0 if it has been set to 1. The STA flag is cleared by reading the I2CSR register.

### Address Frame

In the 7-bit addressing mode: The ADRS flag is set after the address frame is sent by the master device and the acknowledge signal from the address matched slave device is received. In order to receive the following data frame, the ADRS bit must be cleared to 0 if it has been set to 1. The ADRS bit is cleared after reading the I2CSR register.

In the 10-bit addressing mode: The ADRS bit in the I2CSR register will be set twice in the 10-bit addressing mode. The first time the ADRS bit is set is when the first header byte and the second address byte are sent and the acknowledge signals from the slave device are received. The second time the ADRS bit is set is when the second header byte is sent and the slave acknowledge signal is received. In order to receive the following data frame, the ADRS bit must be cleared to 0 if it has been set to 1. The ADRS bit is cleared after reading the I2CSR register. The detailed master receiver mode timing diagram is shown in the following figure.

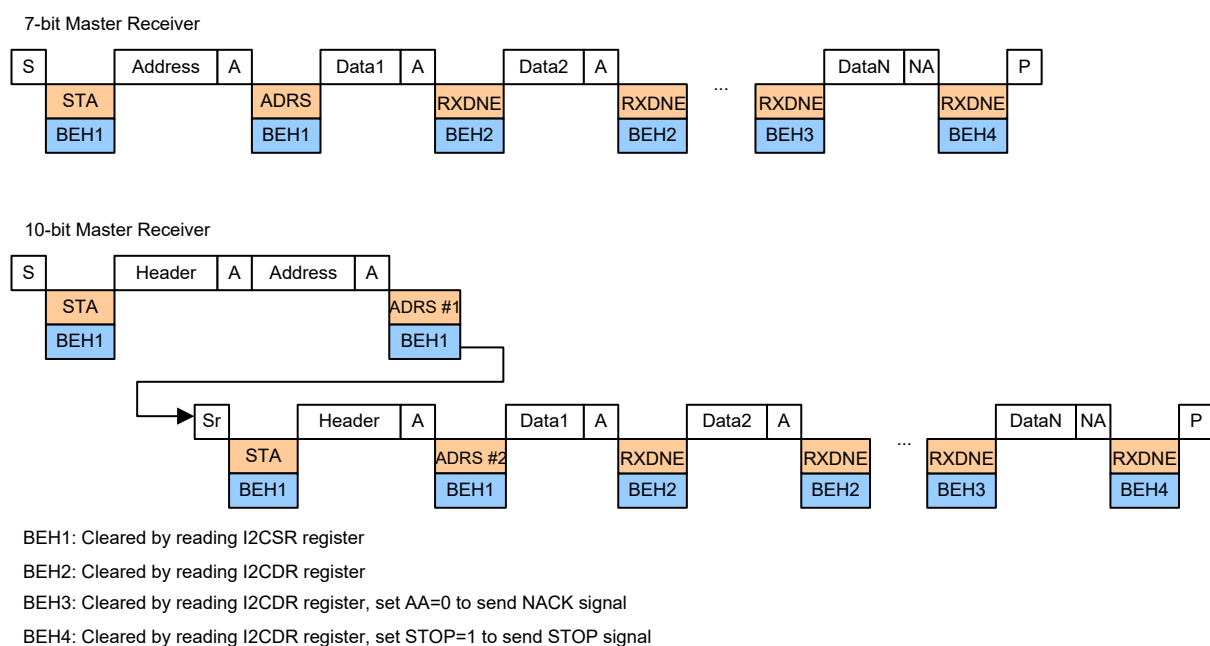
### Data Frame

In the master receiver mode, data is transmitted from the slave device. Once a data is received by the master device, the RXDNE flag in the I2CSR register is set but it will not hold the SCL line. However, if the device receives a complete new data byte and the RXDNE flag has already been set

to 1, the RXBF bit in the I2CSR register will be set to 1 and the SCL line will be held at a logic low state. When this situation occurs, data from the I2CDR register should be read to continue the data transfer process. The RXDNE flag can be cleared after reading the I2CDR register.

### Close / Continue Transmission

The master device needs to reset the AA bit in the I2CCR register to send an 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 an 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 150. Master Receiver Timing Diagram**

## Slave Transmitter Mode

### Address Frame

In the 7-bit addressing mode, the ADRS bit in the I2CSR register is set after the slave device receives the calling address which matches with the slave device address. In the 10-bit addressing mode, the ADRS bit is set for the first time when the first header byte and the second address byte are both matched. Not that when the second header byte is also matched, the ADRS bit will be set again. After the ADRS bit has been set to 1, it must be cleared to 0 to continue the data transfer process. The ADRS bit is cleared after reading the I2CSR register.

### Data Frame

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

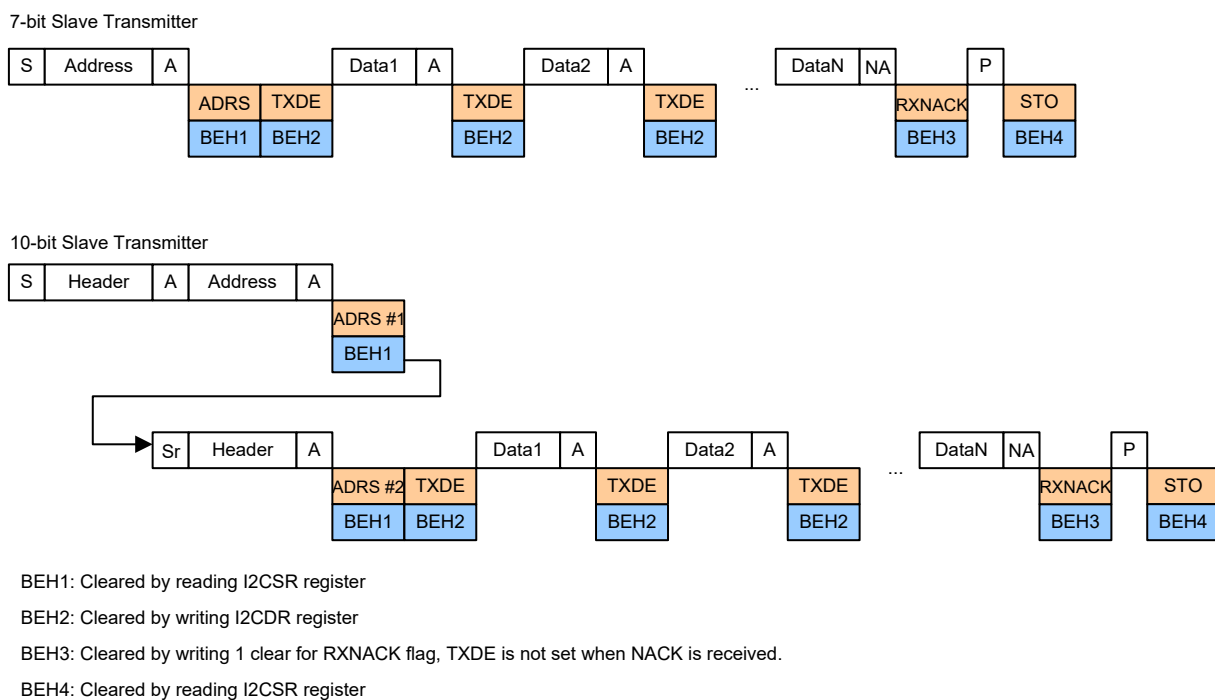


**Receive Not-Acknowledge**

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

**STOP Condition**

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



**Figure 151. Slave Transmitter Timing Diagram**

## Slave Receiver Mode

### Address Frame

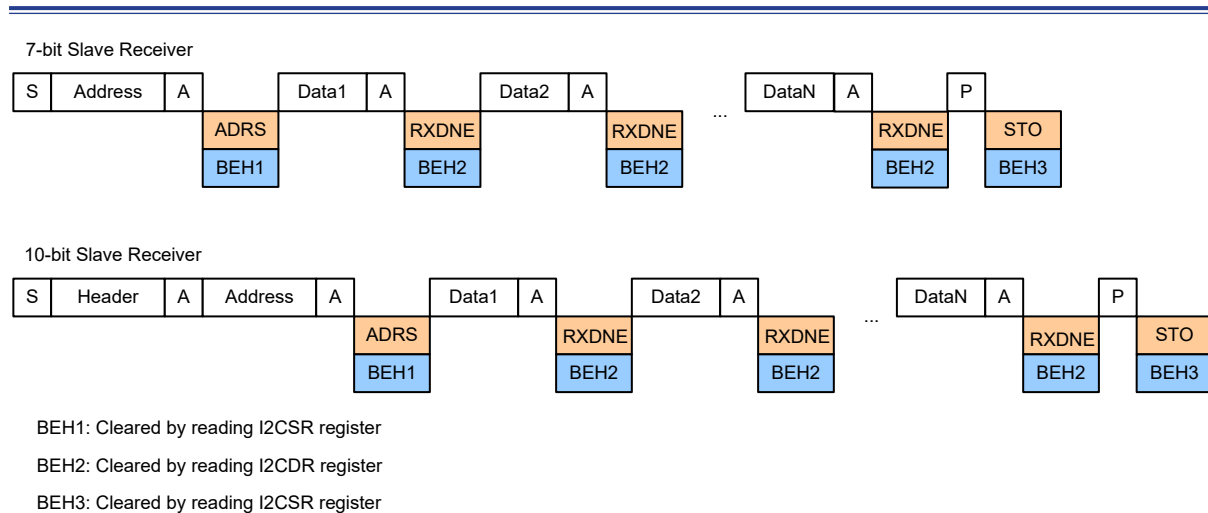
The ADRS bit in the I2CSR register is set after the slave device receives the calling address which matches with the slave device address. After the ADRS bit has been set to 1, it must be cleared to 0 to continue the data transfer process. The ADRS flag is cleared after reading the I2CSR register.

### 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 152. 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 47. Conditions of Holding SCL line**

Type	Condition	Description	Eliminating Condition
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 asserted after receiving an 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 address frame is sent and an ACK from slave is returned (Note: Reference Figure 149 and Figure 150) Slave: I <sup>2</sup> C is addressed as slave device (Note: Reference Figure 151 and Figure 152)	Reading I2CSR register
	STA	Master sends a START signal	Reading I2CSR register
	RXBF	Received a complete new data and meanwhile the RXDNE flag has been set already before.	Reading I2CDR register

Type	Condition	Description	Eliminating Condition
Event	Master receives NACK	No matter in address or data frame, once received an NACK signal will hold SCL line in master mode.	Set TAR Set STOP
	Master sends NACK used in receiver mode	Occurred when receiving the last data byte in Master receiver mode (Note: Reference Figure 150, and RXNACK flag won't be asserted in 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 flag is asserted.

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

- The 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 (HT32F54243/HT32F54253 only)

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, an NACK signal will automatically be sent out to properly terminate the data transfer.

For a more detailed description about the PDMA configurations, refer to the PDMA chapter.

## Register Map

The following table shows the I<sup>2</sup>C registers and reset values. The PDMA related describes are only available for the HT32F54243/HT32F54253 devices.

**Table 48. 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	RW	0	RW	0	RW	1	RW	0
	7	6	5	4	3	2	1	0
Type/Reset	RW	0	Reserved		RW	0	RW	0

Bits	Field	Descriptions
[15:14]	SEQFILTER	SDA or SCL Input Sequential Filter Configuration Bits 00: Sequential filter is disabled 01: 1 PCLK glitch filter 1x: 2 PCLK glitch filter Note: This setting would affect the frequency of SCL. Details are described in I2CSLPGR register.

Bits	Field	Descriptions
[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 ENTOUT bit to 1.
[10]	DMANACK	PDMA Mode NACK Control 0: No operation 1: The I <sup>2</sup> C master receiver module sends an NACK signal automatically after receiving the last byte from the slave transmitter in the PDMA mode
[9]	RXDMAE	PDMA Mode RX Request Enable Control 0: RX PDMA request is disabled 1: RX PDMA 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.
[8]	TXDMAE	PDMA Mode TX Request Enable Control 0: TX PDMA request is disabled 1: TX PDMA 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
					0	RW	0	RW
	7	6	5	4	3	2	1	0
Type/Reset	Reserved				GCSIE	ADRSIE	STOIE	STAIE
					RW	0	RW	0
					0	RW	0	RW
					0	RW	0	RW

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

Bits	Field	Descriptions
[9:0]	ADDR	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
	7	6	5	4	3	2	1	0
Type/Reset	Reserved				GCS	ADRS	STO	STA
					RC	0	RC	0

Bits	Field	Descriptions
[21]	TXNRX	Transmitter / Receiver Mode 0: Receiver mode 1: Transmitter mode Read only bit.
[20]	MASTER	Master Mode 0: I <sup>2</sup> C is in the slave mode or idle 1: I <sup>2</sup> C is in the master mode The I <sup>2</sup> C interface is switched as a master device on the I <sup>2</sup> C bus when the I2CTAR register is assigned and the I <sup>2</sup> C bus is idle. The MASTER bit is cleared by hardware when software disables the I <sup>2</sup> C bus by clearing the I2CEN bit to 0 or sends a STOP condition to the I <sup>2</sup> C bus or the bus error is detected. This bit is set and cleared by hardware and is a read only bit.
[19]	BUSBUSY	Bus Busy 0: I <sup>2</sup> C bus is idle 1: I <sup>2</sup> C bus is busy The I <sup>2</sup> C interface hardware starts to detect the I <sup>2</sup> C bus status if the interface is enabled by setting the I2CEN bit to 1. It is set to 1 when the SDA or SCL signal is detected to have a logic low state and cleared when a STOP condition is detected.
[18]	RXBF	Buffer Full Flag in Receiver Mode 0: Data buffer is not full 1: Data buffer is full This bit is set when the data register I2CDR has already stored a data byte and meanwhile the data shift register also has been received a complete new data byte. The RXBF bit is cleared by software reading the I2CDR register.



Bits	Field	Descriptions
[17]	TXDE	<p>Data Register Empty in Transmitter Mode</p> <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, if both the GCEN and the AA bit are set to 1, then it is switched as a general call slave. This flag is cleared automatically after being read.</p>

Bits	Field	Descriptions
[2]	ADRS	<p>Address Transmit (master mode) / Address Receive (slave mode) Flag</p> <p>Address Sent in Master Mode:</p> <p>0: Address frame has not been transmitted</p> <p>1: Address frame has been transmitted</p> <p>For the 7-bit addressing mode, this bit is set after the master device receives the address frame acknowledge bit sent from the slave device. For the 10-bit addressing mode, this bit is set after receiving the acknowledge bits of the first header byte and the second address. Note that when the second header byte, if exists, is acknowledged, this bit will also be set.</p> <p>Address Matched in Slave Mode:</p> <p>0: 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
	7	6	5	4	3	2	1	0
Type/Reset	SHPG							
	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[15:0]	SHPG	<p>SCL Clock High Period Generation</p> <p>High period duration setting <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	RW
	7	6	5	4	3	2	1	0	
Type/Reset	SLPG								
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:0]	SLPG	<p>SCL Clock Low Period Generation</p> <p>Low period duration setting <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>

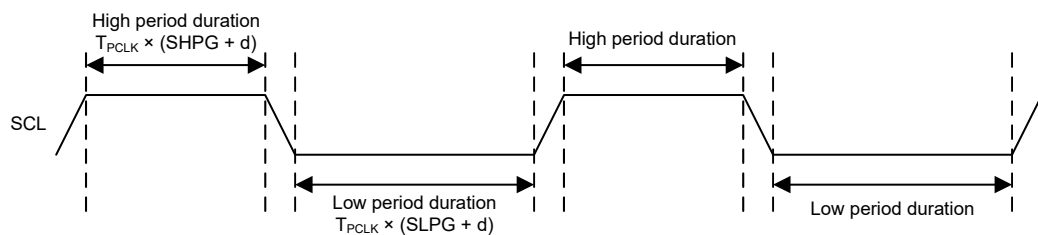


Figure 153. SCL Timing Diagram

Table 49. 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					
	8 MHz	20 MHz	24 MHz	40 MHz	48 MHz	60 MHz
100 kHz (Standard Mode)	68	188	228	388	468	588
400 kHz (Fast Mode)	8	38	48	88	108	138
1 MHz (Fast Mode Plus)	N/A	8	12	28	36	48

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>

This register specifies the target device address to be communicated.

Reset value: 0x0000 0000

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

Bits	Field	Descriptions
[10]	RWD	<p>Read or Write Direction</p> <p>0: Write direction to target slave address</p> <p>1: Read direction from target slave address</p> <p>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.</p>
[9:0]	TAR	<p>Target Slave Address</p> <p>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, it is suggested to set the I2CTAR register after a byte transfer is completed. It is not allowed to set TAR in the address frame. I2CTAR[9:7] is not available under the 7-bit addressing mode.</p>

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.

Reset value: 0x0000 0000

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

Bits	Field	Descriptions
[9:0]	ADDMR	<p>Address Mask Control Bit</p> <p>The ADDMR[i] is used to specify whether the i<sup>th</sup> bit of the ADDR field 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
Type/Reset	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 Timeout 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 any one of the following conditions occurs:</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 any one of the following conditions occurs:</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>

# 21 Serial Peripheral Interface (SPI)

## Introduction

The Serial Peripheral Interface, SPI, provides an SPI protocol data transmit and receive functions in both master or slave mode. The SPI interface uses 4 pins, among which are 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.

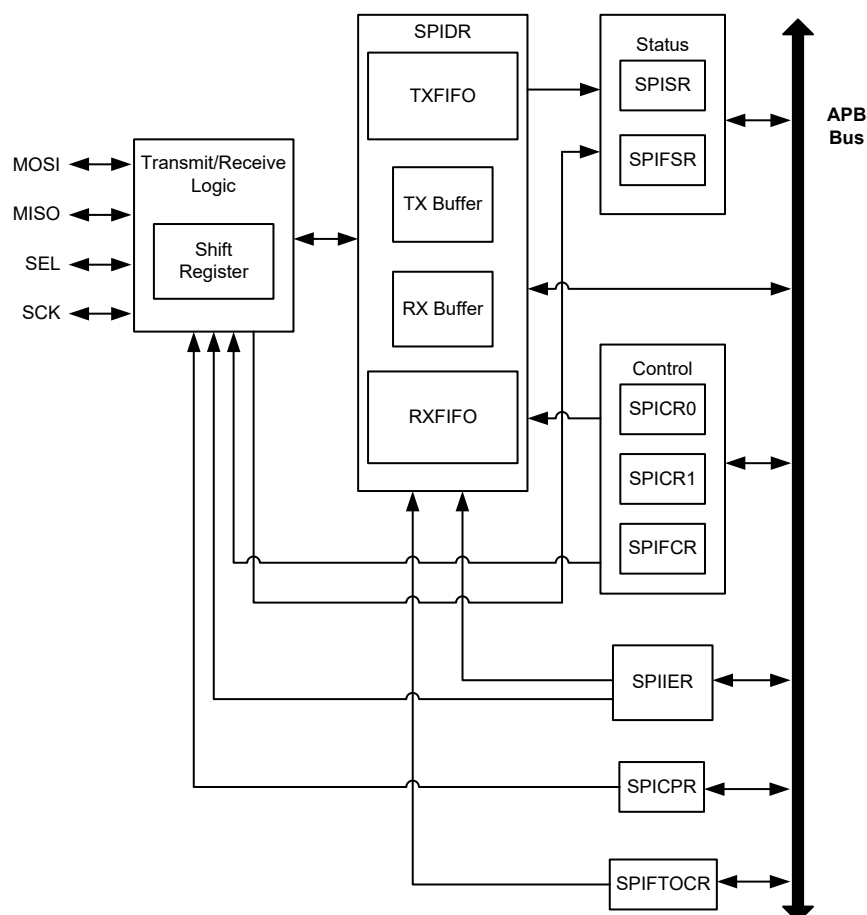


Figure 154. SPI Block Diagram

## 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, the PDMA related describes are only available for the HT32F54243/HT32F54253 devices

## 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.

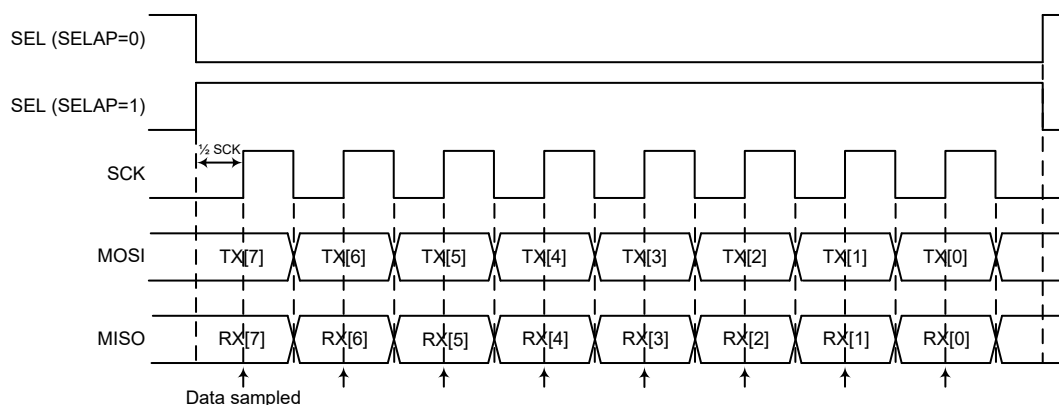
There are four formats contained in the SPI interface. The accompanying table shows how to configure these formats by setting the FORMAT field in the SPICR1 register.

**Table 50. SPI Interface Format Setup**

FORMAT [2:0]	CPOL	CPHA
001	0	0
010	0	1
110	1	0
101	1	1
Others	Reserved	

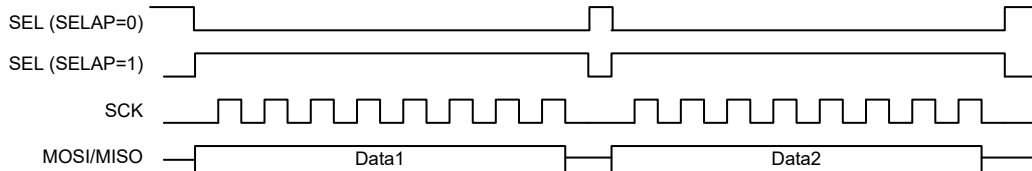
### CPOL = 0, CPHA = 0

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. The accompanying figure shows the single byte data transfer timing of this format.



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

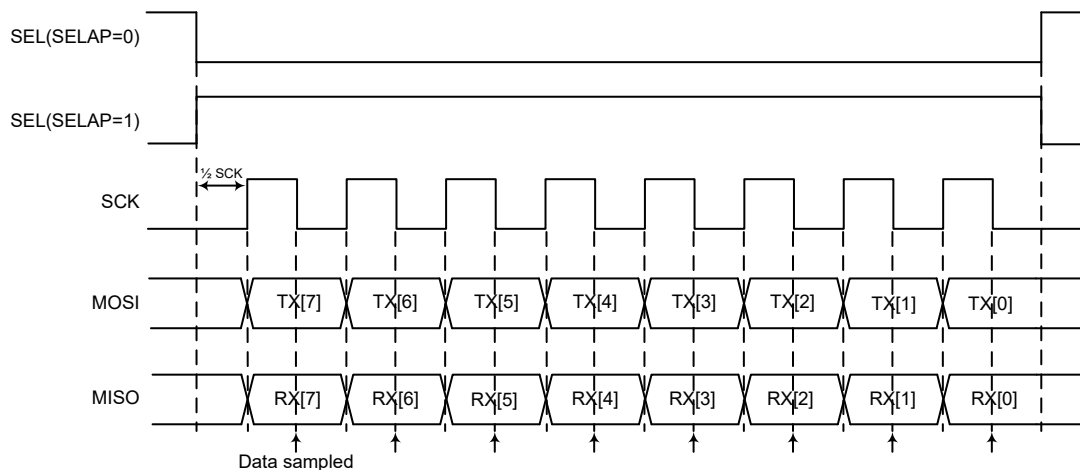
The accompanying figure shows the continuous data transfer timing diagram of this format. Note that the SEL signal must change to an inactive level between each data frame.



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

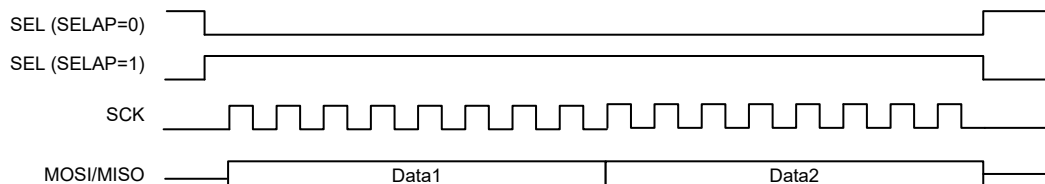
#### CPOL = 0, CPHA = 1

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. The accompanying figure shows the single data byte transfer timing.



**Figure 157. SPI Single Byte Transfer Timing Diagram – CPOL = 0, CPHA = 1**

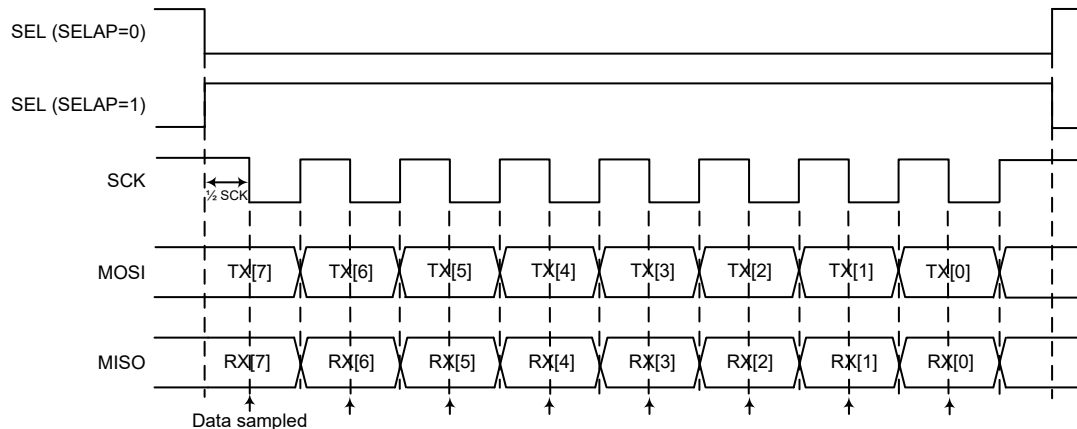
The accompanying figure shows the continuous data transfer diagram timing. Note that the SEL signal must remain active until the last data transfer has completed.



**Figure 158. SPI Continuous Transfer Timing Diagram – CPOL = 0, CPHA = 1**

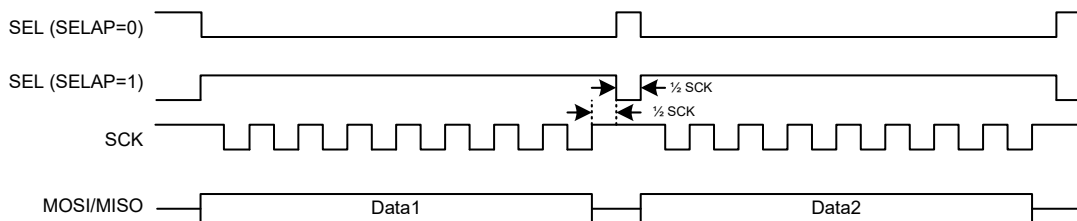
### CPOL = 1, CPHA = 0

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. The accompanying figure shows the single byte transfer timing of this format.



**Figure 159. SPI Single Byte Transfer Timing Diagram – CPOL = 1, CPHA = 0**

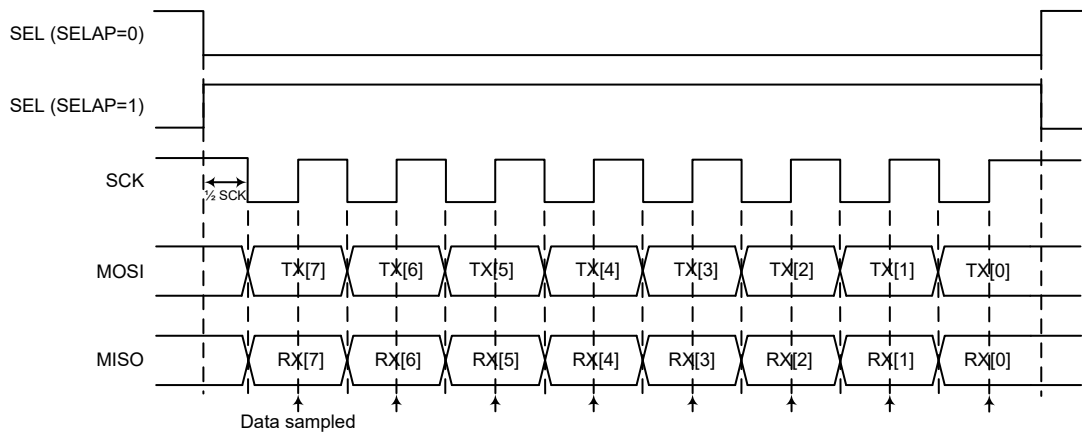
The accompanying figure shows the continuous data transfer timing of this format. Note that the SEL signal must change to an inactive level between each data frame.



**Figure 160. SPI Continuous Transfer Timing Diagram – CPOL = 1, CPHA = 0**

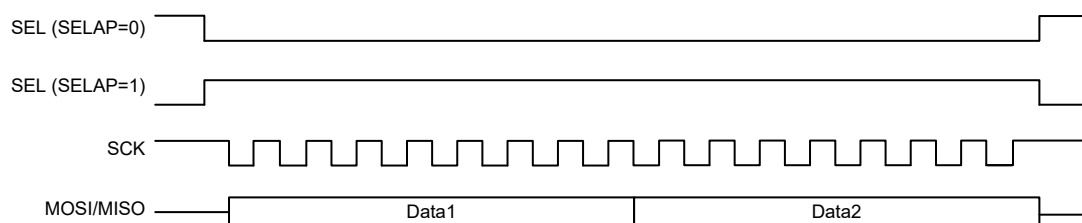
### CPOL = 1, CPHA = 1

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. The accompanying figure shows the single byte transfer timing of this format.



**Figure 161. SPI Single Byte Transfer Timing Diagram – CPOL = 1, CPHA = 1**

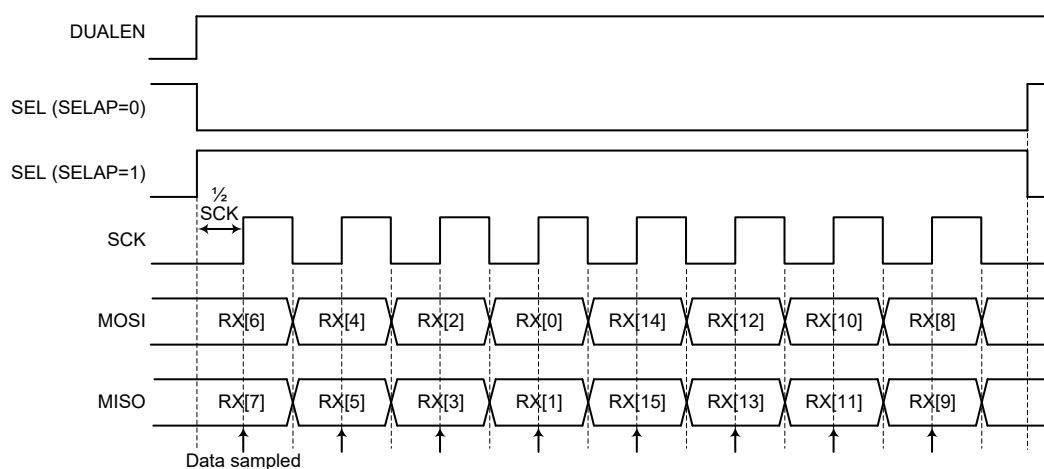
The accompanying figure shows the continuous data transfer timing of this format. Note that the SEL signal must remain active until the last data transfer has completed.



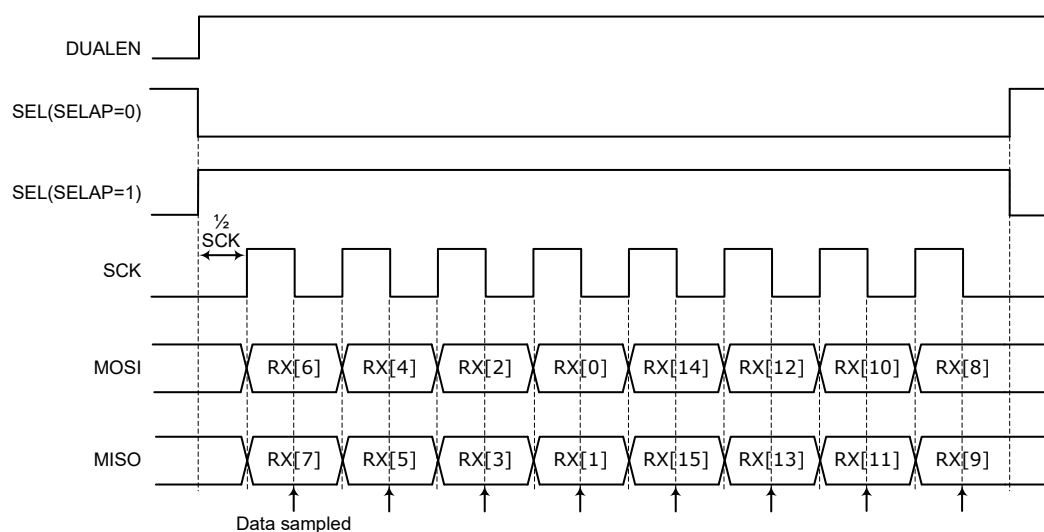
**Figure 162. SPI Continuous Transfer Timing Diagram – CPOL = 1, CPHA = 1**

## 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.

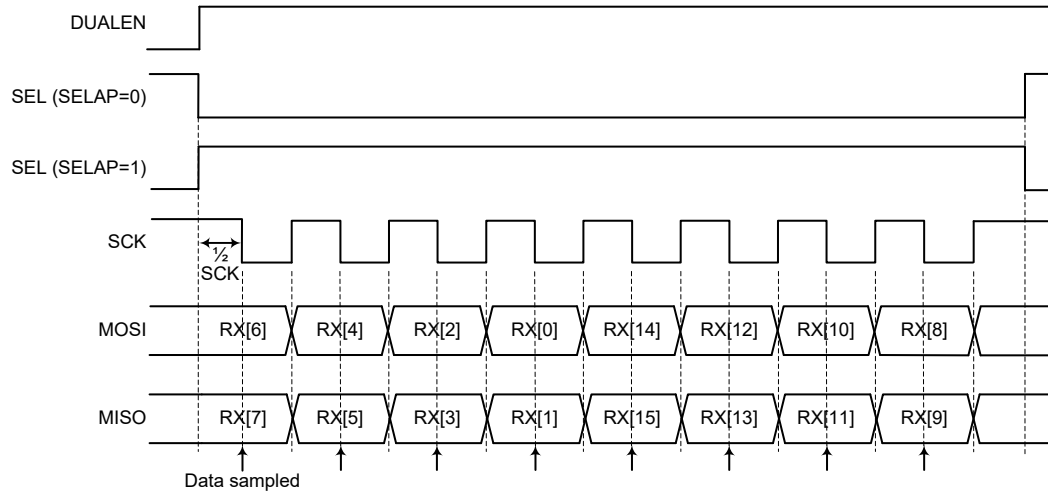


**Figure 163. SPI Dual Mode Bit Sequence – CPOL = 0, CPHA = 0, DFL = 0x8 (16-bit), MSB Transmitted First**

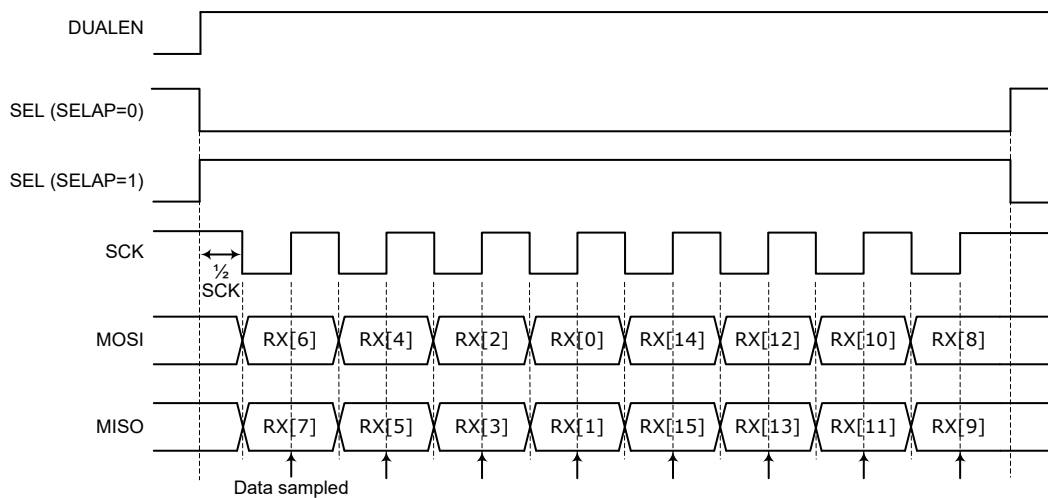


**Figure 164. SPI Dual Mode Bit Sequence – CPOL = 0, CPHA = 1, DFL = 0x8 (16-bit), MSB Transmitted First**



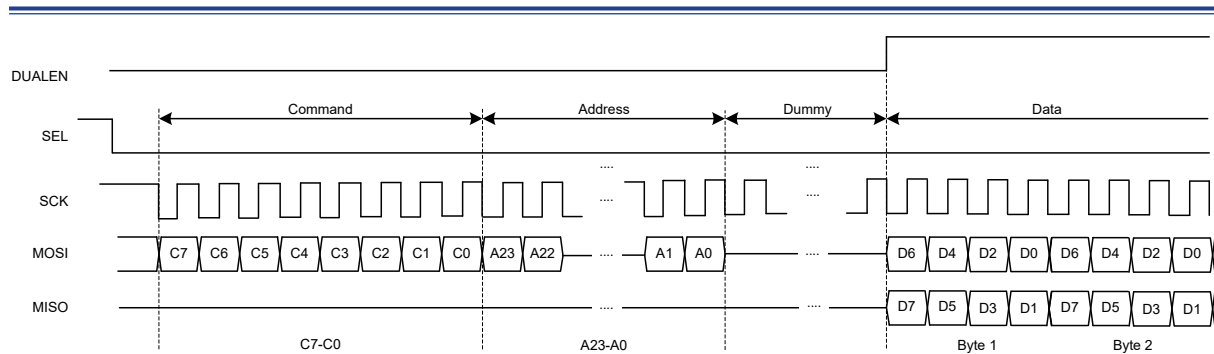


**Figure 165. SPI Dual Mode Bit Sequence – CPOL = 1, CPHA = 0, DFL = 0x8 (16-bit), MSB Transmitted First**



**Figure 166. SPI Dual Mode Bit Sequence – CPOL = 1, CPHA = 1, DFL = 0x8 (16-bit), MSB Transmitted First**

The accompanying figure shows the bit sequence of the SPI Dual mode reading data from an external serial SPI Flash.



**Figure 167. 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.

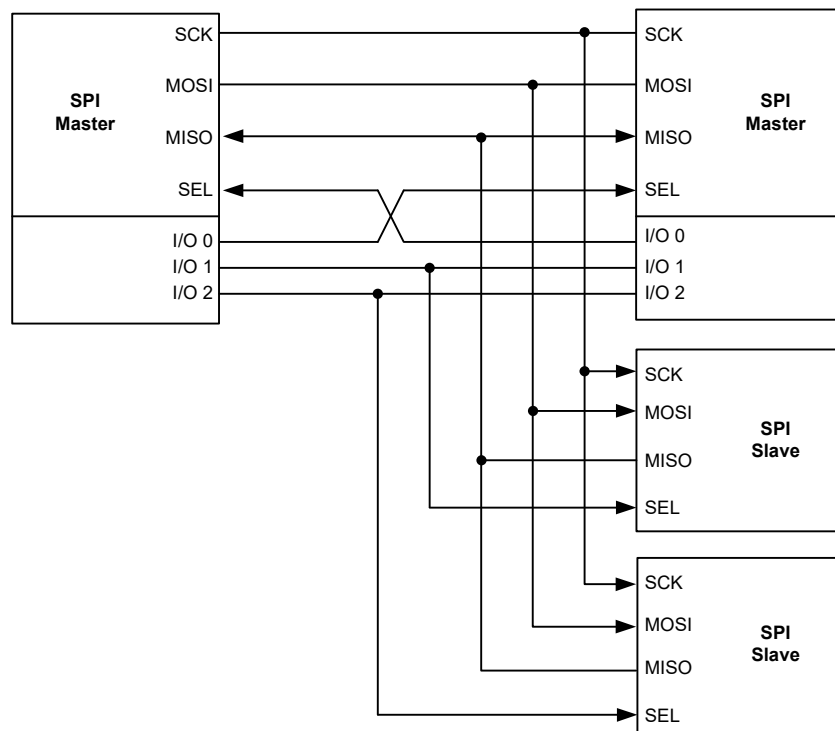


Figure 168. SPI Multi-Master Slave Environment

**Table 51. 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 52. 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 cannot 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 (HT32F54243/HT32F54253 only)

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. The PDMA related describes are only available for the HT32F54243/HT32F54253 devices.

**Table 53. 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	Reserved	MODE	SELM	FIRSTBIT	SELAP	FORMAT		
		RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0
Type/Reset	Reserved				DFL			
					RW	0	RW	0

Bits	Field	Descriptions
[14]	MODE	Master or Slave Mode 0: Slave mode 1: Master mode
[13]	SELM	Slave Select Mode 0: SEL signal is controlled by software – asserted or de-asserted by the SSEL bit 1: SEL signal is controlled by hardware – generated automatically by the SPI hardware Note that the SELM bit is available for master mode only, i.e., MODE = 1.
[12]	FIRSTBIT	LSB or MSB Transmitted First 0: MSB is transmitted first 1: LSB is transmitted first
[11]	SELAP	Slave Select Active Polarity 0: SEL signal is active low 1: SEL signal is active high



Bits	Field	Descriptions																																							
[10:8]	FORMAT	<p>SPI Data Transfer Format</p> <p>These three bits are used to determine the data transfer format of the SPI interface.</p> <table border="1"> <thead> <tr> <th>FORMAT [2:0]</th><th>CPOL</th><th>CPHA</th></tr> </thead> <tbody> <tr> <td>001</td><td>0</td><td>0</td></tr> <tr> <td>010</td><td>0</td><td>1</td></tr> <tr> <td>110</td><td>1</td><td>0</td></tr> <tr> <td>101</td><td>1</td><td>1</td></tr> <tr> <td>Others</td><td colspan="2">Reserved</td></tr> </tbody> </table> <p>CPOL: Clock Polarity            0: SCK Idle state is low            1: SCK Idle state is high</p> <p>CPHA: Clock Phase            0: Data is captured on the first SCK clock edge            1: Data is captured on the second SCK clock edge</p>	FORMAT [2:0]	CPOL	CPHA	001	0	0	010	0	1	110	1	0	101	1	1	Others	Reserved																						
FORMAT [2:0]	CPOL	CPHA																																							
001	0	0																																							
010	0	1																																							
110	1	0																																							
101	1	1																																							
Others	Reserved																																								
[3:0]	DFL	<p>Data Frame Length</p> <p>Selects the data transfer frame from 1 bit to 16 bits.</p> <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	MFIEN	ROIEN	WCIEN	RXBNEIEN	TXEIEN	TXBEIEN
	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[7]	TOIEN	Time Out Interrupt Enable 0: Disable 1: Enable
[6]	SAIEN	Slave Abort Interrupt Enable 0: Disable 1: Enable
[5]	MFIEN	Mode Fault Interrupt Enable 0: Disable 1: Enable
[4]	ROIEN	Read Overrun Interrupt Enable 0: Disable 1: Enable
[3]	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]	TXEIEN	Transmission Register Empty Interrupt Enable 0: Disable 1: Enable The transmission register empty interrupt request will be generated when the TXE flag and the TXEIEN bit are set.

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

## SPI Clock Prescaler Register – SPICPR

This register specifies the SPI clock prescaler ratio.

Offset: 0x00C

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
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	SPI Clock Prescaler The SPI clock (SCK) is determined by the following equation: $f_{SCK} = f_{PCLK} / (2 \times (CP + 1))$ , where the CP ranges is from 0 to 65535 Note: For the SPI master mode, the APB clock ( $f_{PCLK}$ ) must be at least 2 times faster than the SPI SCK output.

## SPI Data Register – SPIDR

This register stores the SPI received or transmitted Data.

Offset: 0x010

Reset value: 0x0000\_0000

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

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

## 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	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	Reserved							BUSY	
									RO 0
	7	6	5	4	3	2	1	0	
Type/Reset	TO	SA	MF	RO	WC	RXBNE	TXE	TXBE	
	WC 0	WC 0	WC 0	WC 0	WC 0	RO 0	RO 0	1 RO 1	

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

Bits	Field	Descriptions
[2]	RXBNE	RX Buffer Not Empty flag 0: RX buffer is empty 1: RX buffer is not empty This bit indicates the RX buffer status in the non-FIFO mode. It is also used to indicate if the RX FIFO trigger level has been reached in the FIFO mode. This bit will be cleared when the SPI RX buffer is empty in the non-FIFO mode or if the number of data contained in RX FIFO is less than the trigger level which is specified by the RXFTLS field in the SPIFCR register in the SPI FIFO mode.
[1]	TXE	Transmission Register Empty flag 0: TX buffer or TX shift register is not empty 1: TX buffer and TX shift register both are empty
[0]	TXBE	TX Buffer Empty flag 0: TX buffer is not empty 1: TX buffer is empty In the FIFO mode, this bit if set indicates that the number of data contained in TX FIFO is equal to or less than the trigger level specified by the TXFTLS field in the SPIFCR register.

### SPI FIFO Control Register – SPIFCR

This register contains the related SPI FIFO control including the FIFO enable control and the FIFO trigger level selections.

Offset: 0x018

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[10]	FIFOEN	FIFO Enable 0: FIFO is disabled 1: FIFO is enabled This bit cannot be set or reset when the SPI interface is in transmitting.

Bits	Field	Descriptions
[7:4]	RXFTLS	<p>RX FIFO Trigger Level Select</p> <p>0000: Trigger level is 0</p> <p>0001: Trigger level is 1</p> <p>...</p> <p>1000: Trigger level is 8</p> <p>Others: Reserved</p> <p>The RXFTLS field is used to specify the RX FIFO trigger level. When the number of data contained in the RX FIFO is equal to or greater than the trigger level defined by the RXFTLS field, the RXBNE flag will be set</p>
[3:0]	TXFTLS	<p>TX FIFO Trigger Level Select</p> <p>0000: Trigger level is 0</p> <p>0001: Trigger level is 1</p> <p>...</p> <p>1000: Trigger level is 8</p> <p>Others: Reserved</p> <p>The TXFTLS field is used to specify the TX FIFO trigger level. When the number of data contained in the TX FIFO is equal to or less than the trigger level defined by the TXFTLS field, the TXBE flag will be set.</p>

### SPI FIFO Status Register – SPIFSR

This register contains the relevant SPI FIFO status.

Offset: 0x01C

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/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	<p>RX FIFO Status</p> <p>0000: RX FIFO empty</p> <p>0001: RX FIFO contains 1 data</p> <p>...</p> <p>1000: RX FIFO contains 8 data</p> <p>Others: Reserved</p>

Bits	Field	Descriptions
[3:0]	TXFS	TX FIFO Status 0000: TX FIFO empty 0001: TX FIFO contains 1 data ... 1000: TX FIFO contains 8 data Others: Reserved

### SPI FIFO Time Out Counter Register – SPIFCR

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

Offset: 0x020

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	TOC							
	RW	0	RW	0	RW	0	RW	0
	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 $f_{PCLK}$ .



# 22 Universal Synchronous Asynchronous Receiver Transmitter (USART)

## Introduction

The Universal Synchronous Asynchronous Receiver Transceiver, USART, provides a flexible full duplex data exchange using synchronous or asynchronous transfer. The USART is used to translate data between parallel and serial interfaces, and is also commonly used for RS232 standard communication. The USART peripheral function supports a variety of interrupts.

The USART module includes an 8-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.

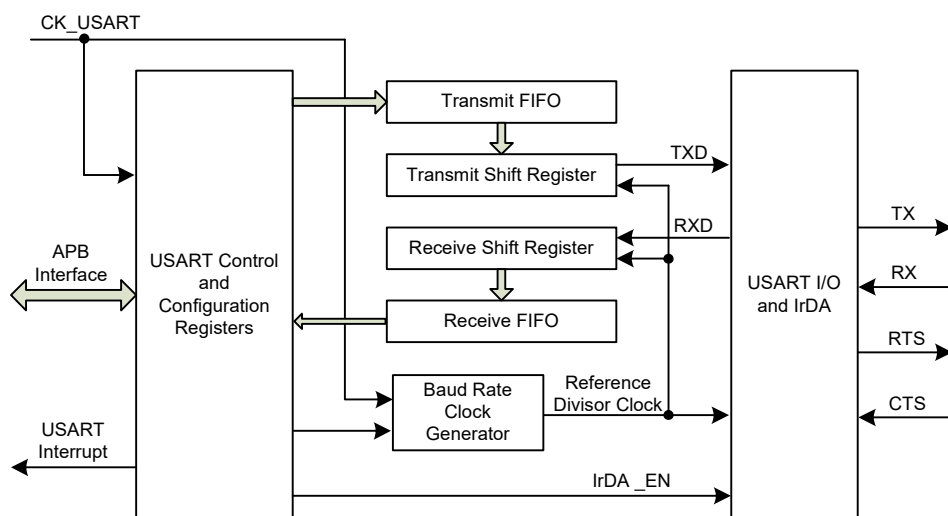


Figure 169. USART Block Diagram

## Features

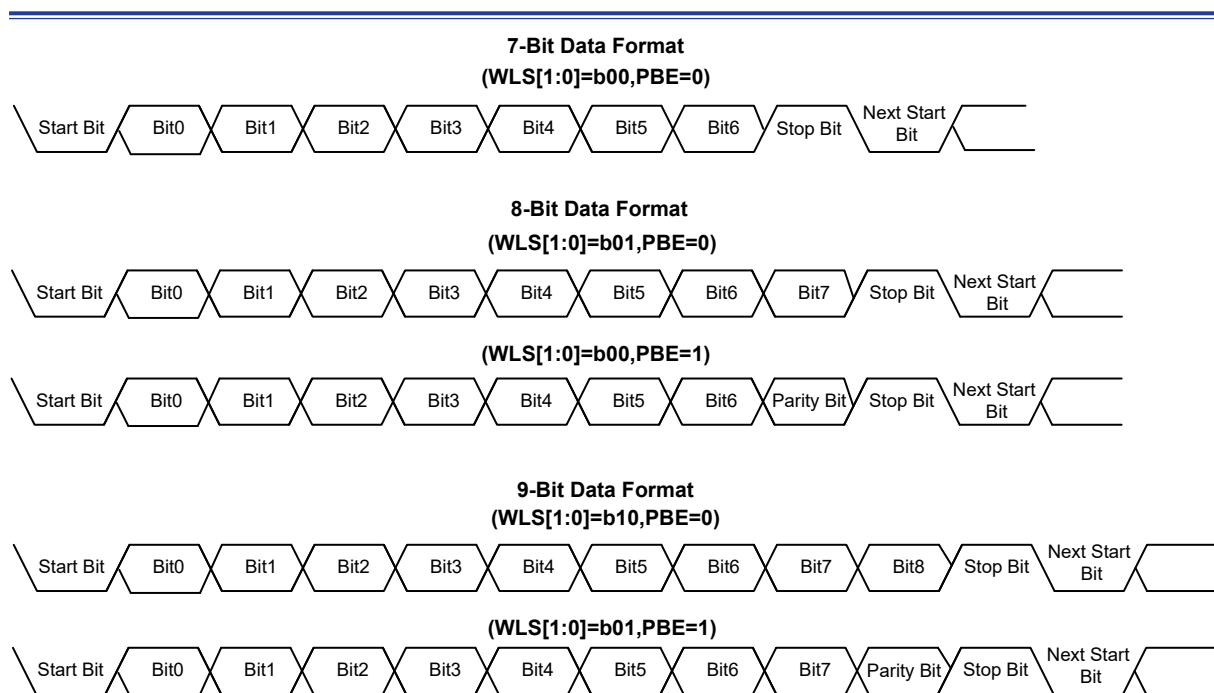
- Supports both asynchronous and clocked synchronous serial communication modes
- Full Duplex Communication Capability
- Programmable baud rate clock frequency up to ( $f_{\text{PLCK}}/16$ ) MHz for asynchronous mode and ( $f_{\text{PLCK}}/8$ ) MHz for synchronous mode
- IrDA SIR encoder and decoder
  - 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
  - Transmit FIFO: 8-level
- Supports PDMA Interface, the PDMA related describes are only available for the HT32F54243/HT32F54253 devices

## 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.



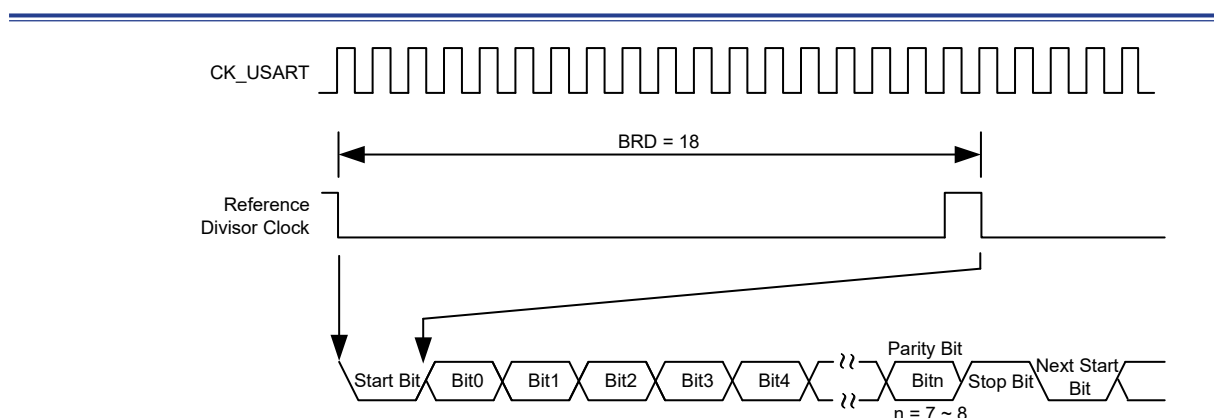
**Figure 170. USART Serial Data Format**

## 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 171. USART Clock CK\_USART and Data Frame Timing**

**Table 54. 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 55. Baud Rate Deviation Error Calculation – CK\_USART = 48 MHz**

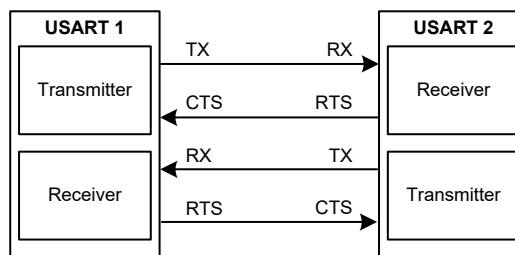
Baud Rate		CK_USART = 48 MHz		
No.	Kbps	Actual	BRD	Deviation Error Rate
1	2.4	2.4	20000	0.00%
2	9.6	9.6	5000	0.00%
3	19.2	19.2	2500	0.00%
4	57.6	57.6	833	0.04%
5	115.2	115.1	417	-0.08%
6	230.4	230.8	208	0.16%
7	460.8	461.5	104	0.16%
8	921.6	923.1	52	0.16%
9	2250	2285.7	21	1.59%
10	3000	3000.0	16	0.00%

**Table 56. 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

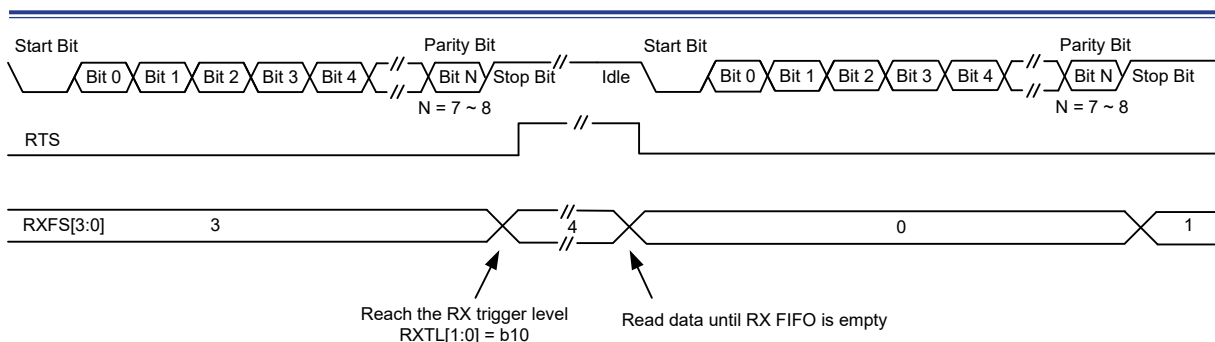
The USART supports the hardware flow control function which is enabled by setting the HFCEN bit in the USRCR register to 1. It is possible to control the serial data flow between two USART devices by using the CTS input and the RTS output. The following figure shows the connection diagram in this mode. 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 172. 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 215 shows the example of RTS flow control.

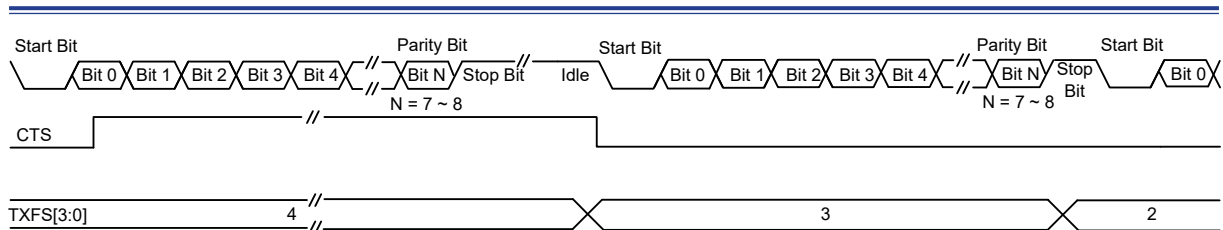


**Figure 173. USART RTS Flow Control**

### 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 following figure shows an example of communication with CTS flow control.

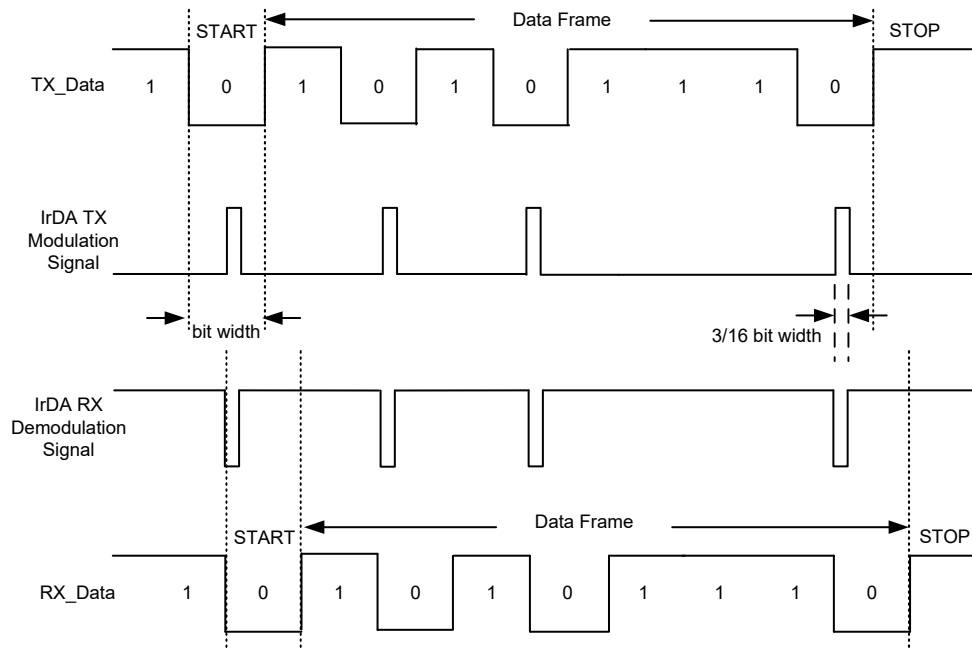


**Figure 174. USART CTS Flow Control**

## IrDA

The USART IrDA mode is provided for half-duplex point-to-point wireless communication.

The USART module includes an integrated modulator and demodulator which allow a wireless communication using infrared transceivers. The transmitter specifies a logic data '0' as a 'high' pulse and a logic data '1' as a 'low' level while the Receiver specifies a logic data '0' as a 'low' pulse and a logic data '1' as 'high' level in the IrDA mode.



**Figure 175. 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.

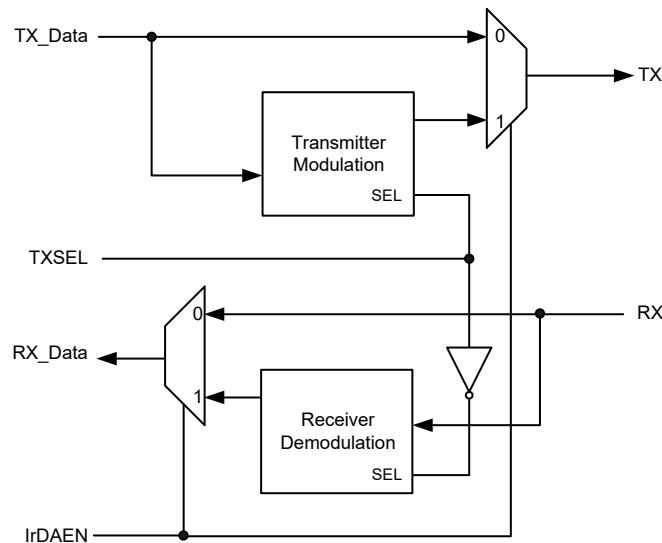
$$T_{\text{IrDA\_L}} = 3 \times \text{IrDAPSC} / \text{CK\_USART}$$

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 176. USART I/O and IrDA Block Diagram**

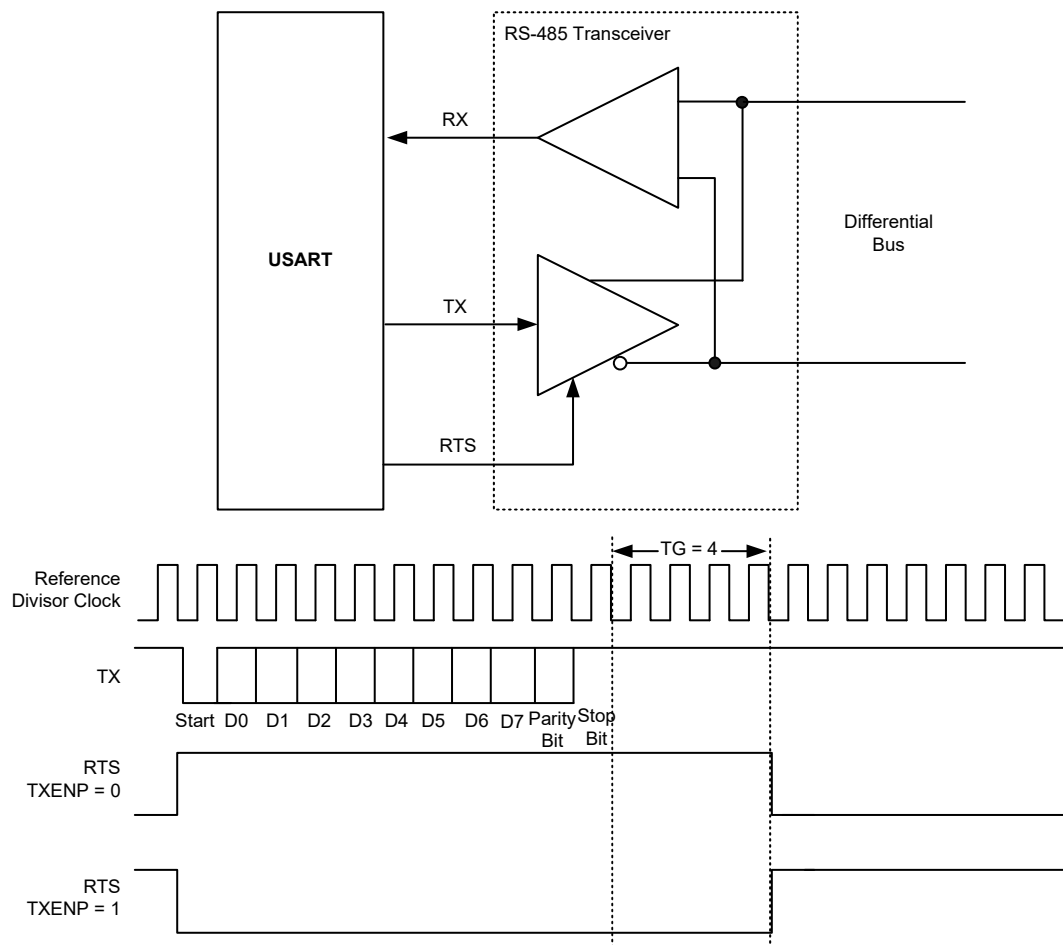
## RS485 Mode

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 177. 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.

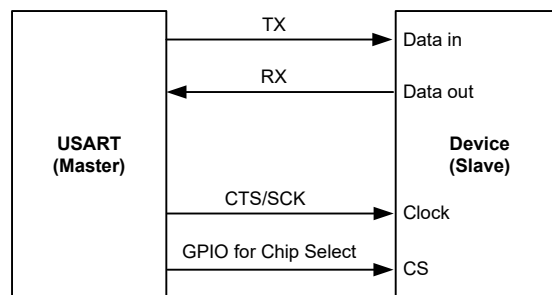
### RS485 Auto Address Detection Operation Mode – AAD

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

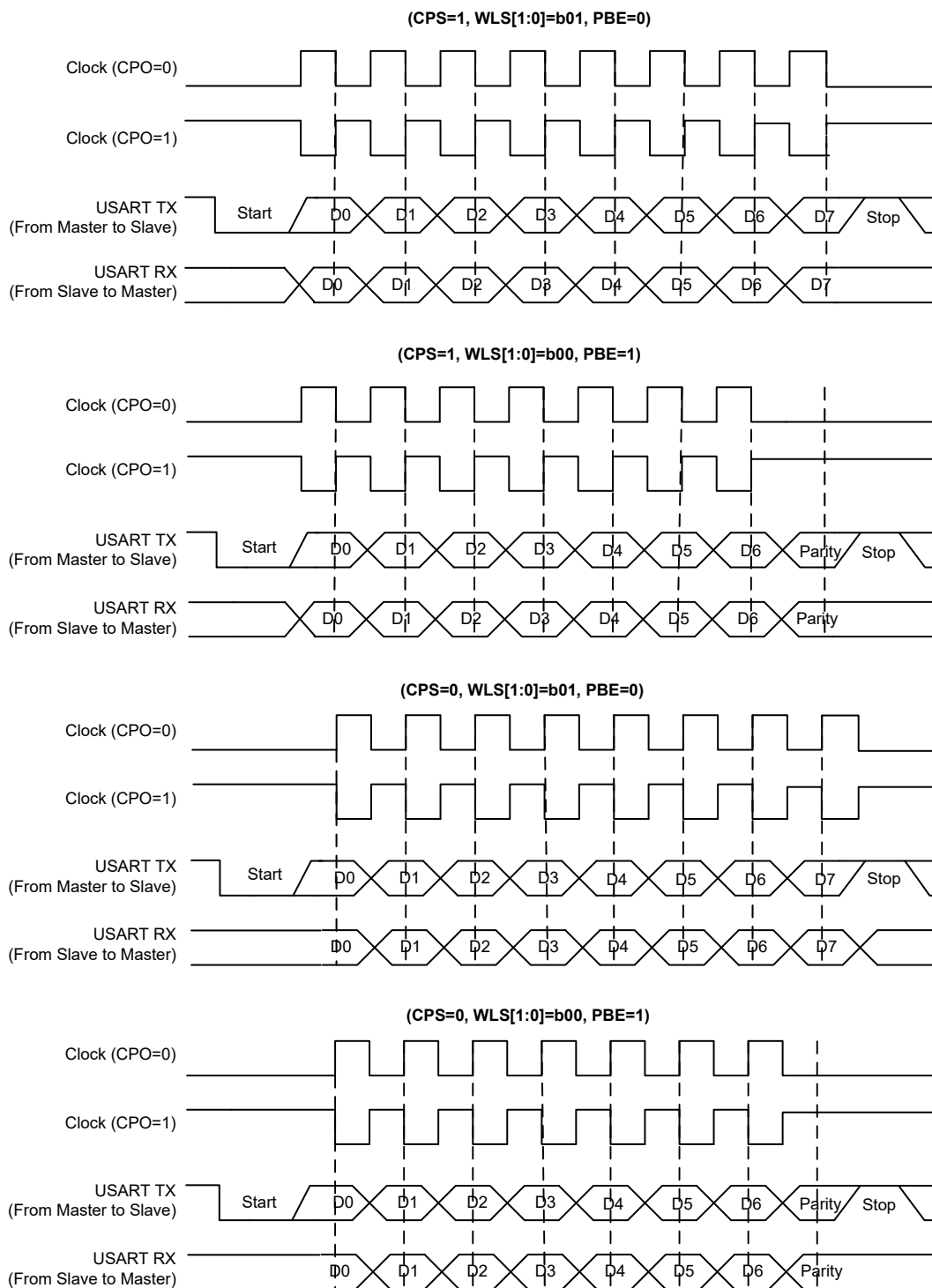
The data is transmitted in a full-duplex style in the USART Synchronous Master Mode, i.e., data transmission and reception both occur at the same time and only support master mode. The USART CTS pin is the synchronous USART transmitter clock output. In this mode, no clock pulses will be sent to the CTS 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 following figure.

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 178. USART Synchronous Transmission Example**

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



**Figure 179. 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 (HT32F54243/HT32F54253 only)

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. The PDMA related describes are only available for the HT32F54243/HT32F54253 devices.

**Table 57. 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
USDLR	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 8 × 9 bits. By reading this register, the USART will return a 7, 8 and 9-bit received data. The DB field bit 8 is valid for the 9-bit mode only and is fixed at 0 for the 8-bit mode. For the 7-bit mode, the DB[6:0] field contains the available bits.</p> <p>Writing data to this buffer register will load data into the Transmit FIFO. The Transmit FIFO has a capacity of up to 8 × 9 bits. By writing to this register, the USART will send out 7, 8 or 9-bit transmitted data. The DB field bit 8 is valid for the 9-bit mode only and will be ignored for the 8-bit mode. For the 7-bit mode, the DB[6:0] field contains the available bits.</p>

## USART Control Register – USRCR

The register specifies the serial parameters such as data length, parity and stop bit for the USART. It also contains the USART enable control bits together with the USART mode and data transfer mode selection.

Offset: 0x004

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	RTS	BCB	SPE	EPE	PBE	NSB	WLS	
	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0
Type/Reset	RXDMAEN	TXDMAEN	URRXEN	URTXEN	HFCEN	TRSM	MODE	
	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[15]	RTS	Request-To-Send Signal 0: Drive USART RTS pin to logic 1 1: Drive USART RTS pin to logic 0 Note that the RTS bit is used to control the USART RTS pin status when the HFCEN bit is reset. When the HFCEN bit is set, this RTS bit is read only and indicates the pin status that is controlled by hardware flow control function.
[14]	BCB	Break Control Bit When this bit is set 1, the serial data output on the USART TX pin will be forced to the Spacing State (logic 0). This bit acts only on the USART TX output pin and has no effect on the transmitter logic.
[13]	SPE	Stick Parity Enable 0: Disable stick parity 1: Stick Parity bit is transmitted This bit is only available when the PBE bit is set to 1. If both the PBE and SPE bits are set to 1 and the EPE bit is cleared to 0, the transmitted parity bit will be stuck to 1. However, when the PBE and SPE bits are set to 1 and also the EPE bit is set to 1, the transmitted parity bit will be stuck to 0.
[12]	EPE	Even Parity Enable 0: Odd number of logic 1's are transmitted or checked in the data word and parity bits 1: Even number of logic 1's are transmitted or checked in the data word and parity bits This bit is only available when the PBE bit is set to 1.

Bits	Field	Descriptions
[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.
[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 PDMA Enable 0: Disable 1: Enable
[6]	TXDMAEN	USART TX PDMA 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		
Type/Reset	Reserved									
	23	22	21	20	19	18	17	16		
Type/Reset	Reserved									
	15	14	13	12	11	10	9	8		
Type/Reset	Reserved						CTSIE	RXT0IE		
	7	6	5	4	3	2	1	0		
Type/Reset	RSADDIE	BIE	FEIE	PEIE	OEIE	TXCIE	TXDEIE	RXDRIE		
	RW	0	RW	0	RW	0	RW	0	RW	0

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]	RXT0IE	Receive FIFO Time-Out Interrupt Enable 0: Disable 1: Enable If this bit is set, an interrupt will be generated when the RXT0F 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.



Bits	Field	Descriptions
[6]	RXTOF	<p>Receive FIFO Time-Out Flag</p> <p>0: RX FIFO Time-Out does not occur 1: RX FIFO Time-Out occurs</p> <p>The RXTOF bit will be set if the RX FIFO is not empty and no activities have occurred in the RX FIFO during the time-out duration specified by the RXTOC field. If an RX FIFO time-out condition has occurred, this flag must be cleared before reading the RX FIFO. Writing 1 to this bit clears the flag.</p>
[5]	RXDR	<p>Receive FIFO Ready Flag</p> <p>0: RX FIFO level is less than threshold 1: RX FIFO level is equal to or higher than threshold</p> <p>The RXDR bit will be set when the FIFO received data amount has reached the threshold level specified by the RXTL field in the USRFCR register. This bit will be cleared when the data is read from the USRDR register and the RX FIFO level is less than threshold setting.</p>
[4]	BII	<p>Break Interrupt Indicator</p> <p>This bit will be set to 1 whenever the received data input is held in the “spacing state” (logic 0) for longer than a full word transmission time, which is the total time of “start bit” + data bits + “parity” + “stop bits” duration. Writing 1 to this bit clears the flag.</p>
[3]	FEI	<p>Framing Error Indicator</p> <p>This bit will be set 1 whenever the next character to be read in the RX FIFO does not have a valid “stop bit”, which means the stop bit following the last data bit or parity bit is detected as logic 0. Writing 1 to this bit clears the flag.</p>
[2]	PEI	<p>Parity Error Indicator</p> <p>This bit will be set to 1 whenever the received character does not have a valid “parity bit”. Writing 1 to this bit clears the flag.</p>
[1]	OEI	<p>Overrun Error Indicator</p> <p>An overrun error will occur only after the RX FIFO is full and when the next character has been completely received in the RX shift register. The character in the shift register will be overwritten if a new character is received in the RX shift register after an overrun event occurs, but the data in the RX FIFO will not be overwritten. The OEI bit is used to indicate the overrun event as soon as it happens. Writing 1 to this bit clears the flag.</p>
[0]	RXDNE	<p>RX FIFO Data Not Empty</p> <p>0: RX FIFO is empty 1: RX FIFO contains at least 1 received data word</p>

## USART Timing Parameter Register – USRTPR

This register contains the USART timing parameters including the transmitter time guard parameters and the receive FIFO time-out value together with the RX FIFO time-out interrupt enable control.

Offset: 0x014

Reset value: 0x0000\_0000

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

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 – cannot 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 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 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 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						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 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.

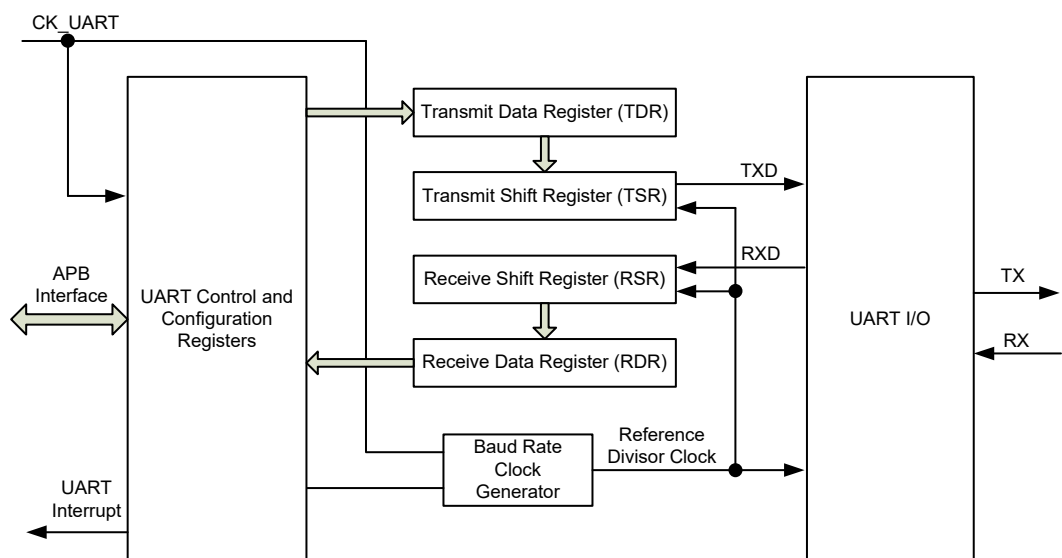


Figure 180. UART Block Diagram

## Features

- Supports asynchronous serial communication modes
- Full Duplex Communication Capability
- Programming baud rate clock frequency up to ( $f_{CLK}/16$ ) 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, the PDMA related describes are only available for the HT32F54243/HT32F54253 devices

## 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.

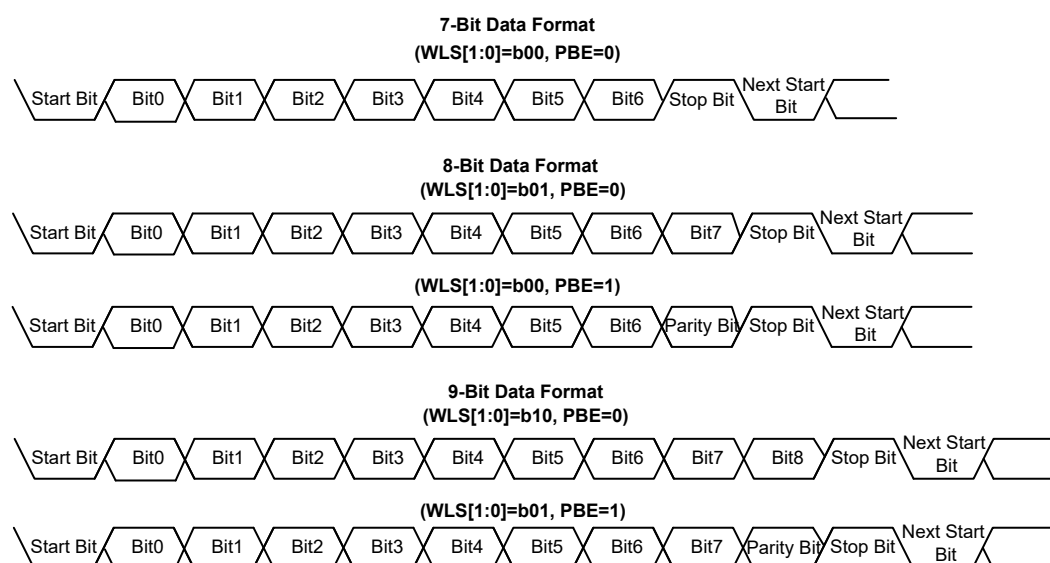


Figure 181. UART Serial Data Format

## 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.

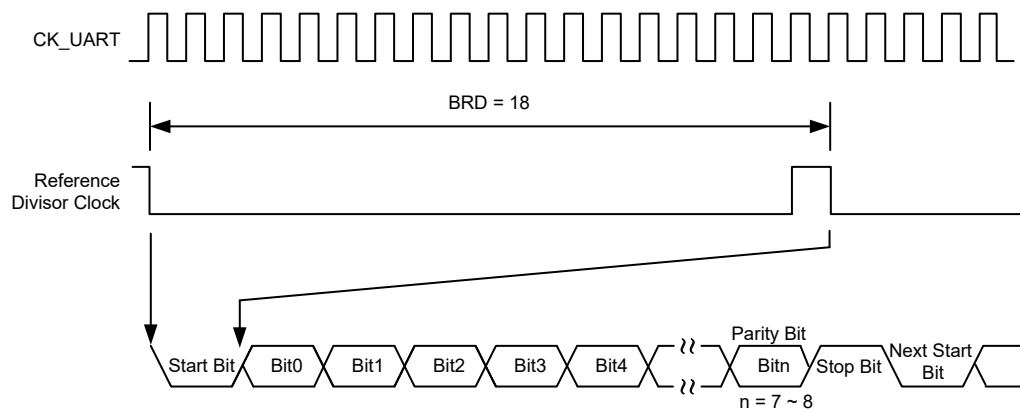


Figure 182. UART Clock CK\_UART and Data Frame Timing

Table 58. 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 59. 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 60. 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 (HT32F54243/HT32F54253 only)

The PDMA interface is integrated in the UART. The PDMA function can be enabled by setting the TXDMAEN or RXDMAEN bit in the URCCR 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).

## 23 Universal Asynchronous Receiver Transmitter (UART)

The following table shows the UART registers and reset values. The PDMA related describes are only available for the HT32F54243/HT32F54253 devices.

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

## UART Data Register – URDR

Offset:	0x000
Reset value:	0x0000 0000

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

## UART Control Register – URCR

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

Offset: 0x004

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24							
Type/Reset	Reserved														
	23	22	21	20	19	18	17	16							
Type/Reset	Reserved														
	15	14	13	12	11	10	9	8							
Type/Reset	Reserved	BCB	SPE	EPE	PBE	NSB	WLS								
		RW	0	RW	0	RW	0	RW	0	RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0							
Type/Reset	RXDMAEN	TXDMAEN	URRXEN	URTXEN	Reserved	TRSM	Reserved								
	RW	0	RW	0	RW	0	RW	0							

Bits	Field	Descriptions
[14]	BCB	Break Control Bit When this bit is set 1, the serial data output on the UART TX pin will be forced to the Spacing State (logic 0). This bit acts only on the UART TX output pin and has no effect on the transmitter logic.
[13]	SPE	Stick Parity Enable 0: Disable stick parity 1: Stick Parity bit is transmitted This bit is only available when the PBE bit is set to 1. If both the PBE and SPE bits are set to 1 and the EPE bit is cleared to 0, the transmitted parity bit will be stuck to 1. However, when the PBE and SPE bits are set to 1 and also the EPE bit is set to 1, the transmitted parity bit will be stuck to 0.
[12]	EPE	Even Parity Enable 0: Odd number of logic 1's are transmitted or checked in the data word and parity bits 1: Even number of logic 1's are transmitted or checked in the data word and parity bits This bit is only available when the PBE bit is set to 1.
[11]	PBE	Parity Bit Enable 0: Parity bit is not generated (transmitted data) and checked (receive data) during transfer 1: Parity bit is generated and checked during transfer Note: When the WLS field is set to "10" to select the 9-bit data format, writing to the PBE bit has no effect.
[10]	NSB	Number of STOP bit 0: One STOP bit is generated in the transmitted data 1: Two STOP bits are generated when 8-bit or 9-bit word length is selected



Bits	Field	Descriptions
[9:8]	WLS	Word Length Select 00: 7 bits 01: 8 bits 10: 9 bits 11: Reserved
[7]	RXDMAEN	UART RX PDMA Enable 0: Disable 1: Enable
[6]	TXDMAEN	UART TX PDMA 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.

Bits	Field	Descriptions
[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	
									RO 1
	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 0	WC 0	0	

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 TXDEIE 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 TXDEIE 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						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

# 24 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. This function is only available for the HT32F54243/HT32F54253 devices.

## 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

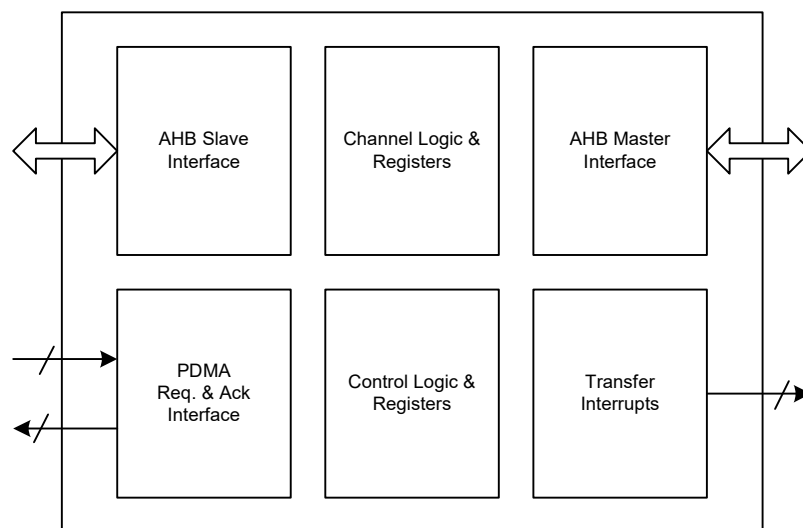


Figure 183. PDMA Block Diagram

## Functional Description

### AHB Master

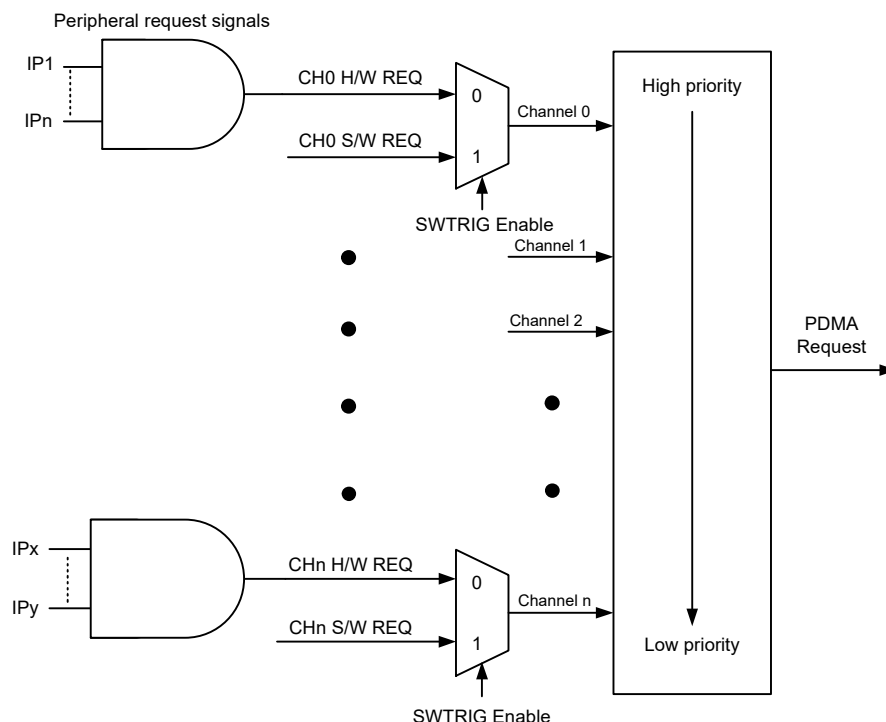
The PDMA is an AHB master connected to other AHB peripherals such as the Flash Memory, the SRAM memory and the AHB-to-APB bridge through the bus-matrix. The CPU and PDMA can access different AHB slaves at the same time via the bus-matrix.

### 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, which means that only one request must be enabled at a time in each PDMA channel. Refer to Figure 184 – PDMA request mapping architecture and detailed peripheral IP requests mapping table is shown as the Table 62. The peripheral DMA requests can be independently activated/de-activated by programming the DMA control bit in the registers of the corresponding peripheral.



**Figure 184. PDMA Request Mapping Architecture**

**Table 62. PDMA Channel Assignments**

IP (x = 0, 1) (y = 0, 1, 2, 3) (z = 0, 1, 2)	PDMA Channel Number					
	CH0	CH1	CH2	CH3	CH4	CH5
ADC	ADC					
SPIx	SPI0_RX	SPI0_TX			SPI1_RX	SPI1_TX
USARTx	USR0_RX	USR0_TX	USR1_RX	USR1_TX		
UARTy	UR2_RX	UR2_TX	UR0_RX	UR0_TX	UR1_RX UR3_RX	UR1_TX UR3_TX
I <sup>2</sup> Cz	I2C2_RX	I2C2_TX	I2C0_RX	I2C1_RX	I2C0_TX	I2C1_TX
MCTM	MT_CH0	MT_TRIG	MT_CH1	MT_CH2	MT_CH3 MT_UEV2	MT_UEV1
GPTM	GT_CH1 GT_CH3	GT_CH2 GT_UEV	GT_CH0 GT_TRIG			

## Channel Transfer

A PDMA channel transfer is split into multiple block transactions with PDMA arbitration occurring at the end of each block transaction. Although these channel transfers can all be activated, there is only one block transaction being transferred through the bus at a time. The channel transfer sequence depends upon the channel priority setting of each PDMA channel. The total transfer size is calculated from the block transaction count and block size. The block size is equal to the product of the block length and data bit width. For an efficient transfer, it is recommended that the block length is set as a multiple of 4.

The total transfer data size calculation is shown as the following equation:

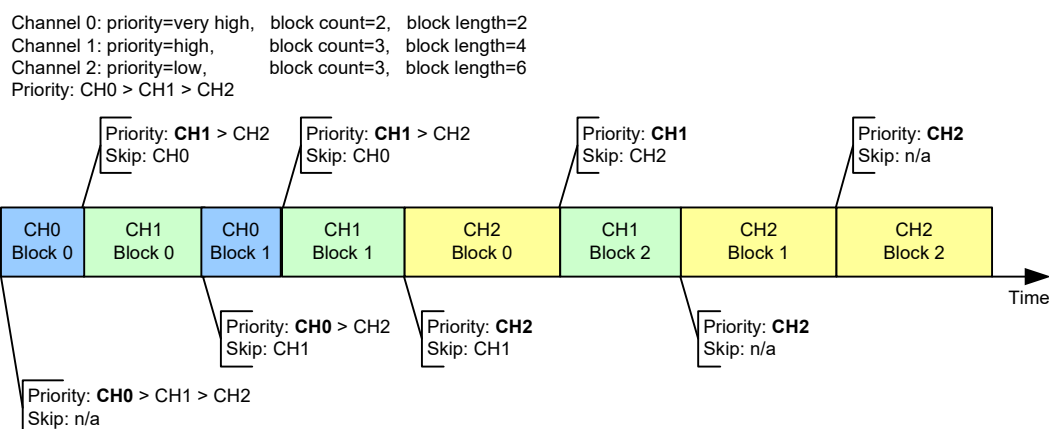
A PDMA channel total transfer data size = Block transaction count × (Block length × Data width)

## Channel Priority

The PDMA provides four priority levels, known as very high, high, medium and low, which can be configured by the application software. The PDMA also provides two methods to determine the channel priority. One is determined by application software configuration and the other is determined by the fixed hardware channel number. The PDMA arbitration processor will first check the software configuring channel priority level used to request the PDMA to provide the data transfer services. If more than one channel has the same priority, the channel with a smaller channel number will have priority over one with a larger channel number after arbitration.

Note that the highest priority channel will not occupy the PDMA service all the time when other lower priority channel requests are pending. The highest priority channel will be skipped for one block transaction time duration after one block transaction is complete. Then a block transaction requested by the second priority channel will be performed. After a block transaction of the second priority channel is complete, the PDMA arbitration processor will re-check all of the requested channel priority with the exception of the second priority channel since the second priority channel will be excluded after the end of a block transaction. Therefore, a block data transaction of the higher priority channel will be serviced and this channel will be excluded from the priority arbitration at the end of the block transaction. The PDMA will keep transferring the data using the method described above until all of the requested channel data transfer is complete. Refer to the accompanying figure for an example which shows the PDMA channel arbitration and scheduling.





**Figure 185. PDMA Channel Arbitration and Scheduling Example**

## Transfer Request

For a peripheral-to-memory or memory-to-peripheral transfer, one peripheral hardware request will trigger one block transaction of the dedicated PDMA channel. However, a complete data transfer of the relevant dedicated PDMA channel will be triggered when a software request occurs. It is recommended that the PDMA channel is configured to have a lower priority level and a smaller block length which is requested by the software for memory-to-memory data copy applications.

## Address Mode

The PDMA provides three kinds of address modes which are the linear address, circular address and fixed address modes. These different address modes are used to support different kinds of source and destination address arrangements. The following table shows the detailed address mode combinations.

**Table 63. PDMA Address Modes**

Source Address Mode	Destination Address Mode
Linear Increment / Decrement Address	Linear Increment / Decrement Address
Linear Increment / Decrement Address	Circular Increment / Decrement Address
Linear Increment / Decrement Address	Fixed Address
Circular Increment / Decrement Address	Linear Increment / Decrement Address
Circular Increment / Decrement Address	Circular Increment / Decrement Address
Fixed Address	Linear Increment / Decrement Address
Fixed Address	Fixed Address

### Linear Address Mode

After data is transferred, the current address will be increased or decreased by 1, 2 or 4 depending upon the data bit width setting.

### Circular Address Mode

After data is transferred, the current address will be increased or decreased by 1, 2 or 4 depending upon the data bit width setting. When a block transaction is complete, the current address is loaded with the configured start address.

### Fixed Address Mode

After data is transferred, the current address remains unchanged.

### Auto-Reload

When the auto-reload control bit, AUTORLn, in the PDMA channel n control register PDMACHnCR is set, both the channel n current address and the channel n current transfer size will be automatically reloaded with the corresponding start value after the current PDMA channel data transfer has totally completed. The channel n will still be activated and the next relative PDMA request can be serviced without any re-configuration using the application software.

### Transfer Interrupt

There are five transfer events during which the interrupts can be asserted for each PDMA channel. These are the block transaction end (BE), half transfer (HT), transfer complete (TC), transfer error (TE) and global transfer event (GE). Setting the corresponding control bits in the PDMA interrupt enable register PDMAIER will enable the relevant interrupt events. The global interrupt event, GE, will be generated if any of the four interrupt events including the BE, HT, TC and TE occurs. Clearing the BE, HT, TC or TE event flag will also clear the GE flag. Clearing the GE flag will automatically clear all other event flags. The TE interrupt event will occur when the PDMA accesses a system reserved address space or when the PDMA receives a request but the corresponding transfer size setting is equal to zero.

## Register Map

The following table shows the PDMA registers and reset values.

**Table 64. 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	
					RW	0	RW	0
	7	6	5	4	3	2	1	0
Type/Reset	SRCAMODn	SRCAINCn	DSTAMODn	DSTAINCn	DWIDTHn		SWTRIGN	CHnEN
	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[11]	AUTORLn	Channel n Auto-Reload Enable Control 0: Disable Auto-Reload function 1: Enable Auto-Reload function  If this bit is set to 1 to enable the auto-reload function, both the channel n current address and the channel n current transfer size will be reloaded with the relevant start value and the PDMA channel n will still be activated when a transfer is complete. If this bit is cleared to 0, the channel n current address and the channel n current transfer size will remain unchanged and the PDMA channel n will be disabled after a transfer completion.
[10]	FIXAENn	Channel n Fixed Address Enable control 0: Disable fixed address function in the circular address mode 1: Enable fixed address function in the circular address mode  Note that this bit is only available when the source or destination address mode is set to be in the circular address mode. For example, the source address mode is set as in the linear address mode and the destination address mode is set as in the circular mode. If this bit is set to enable the fixed address function, then the source address mode will still be in the linear address but the destination address mode will be in the fixed address mode instead of the circular address mode.
[9:8]	CHnPRI	Channel n Priority 00: Low 01: Medium 10: High 11: Very high  The CHnPRI field is used to configure the channel priority using the application program. If there are more than one channel which have the same software configured priority level, the channel with the smaller channel number will have priority to transfer one block of data after the arbitration.

Bits	Field	Descriptions
[7]	SRCAMODn	Channel n Source Address Mode selection 0: Linear address mode 1: Circular address mode In the linear address mode, the current source address value can be increased or decreased, determined by the SRCAINCn bit value during a complete transfer. In the circular address mode, the current source address value can be increased or decreased which is also determined by the SRCAINCn bit value during a block transfer and will be loaded with the lower 16-bit value of the PDMACHnSADR register, which will be regarded as the current source address when a block transaction has completed.
[6]	SRCAINCn	Channel n Source Address Increment control 0: Increment 1: Decrement This bit is used to determine whether the current source address is increased or decreased during a complete transfer in the linear address mode or a block transfer in the circular address mode.
[5]	DSTAMODn	Channel n Destination Address Mode selection 0: Linear address mode 1: Circular address mode In linear address mode, the current destination address value can be increased or decreased, determined by the DSTAINCn bit value during a complete transfer. In the circular address mode, the current destination address value can be increased or decreased which is also determined by the DSTAINCn bit value during a block transfer and will be loaded with the lower 16-bit value of the PDMACHnDADR register, which will be regarded as the current destination address when a block transfer has completed.
[4]	DSTAINCn	Channel n Destination Address Increment Control 0: Increment 1: Decrement This bit is used to determine if the current destination address is increased or decreased during a complete transfer in the linear address mode or a block transfer in the circular address mode.
[3:2]	DWIDTHn	Data Bit Width selection 00: 8-bit 01: 16-bit 10: 32-bit 11: Reserved The field is used to select the data bit width of the corresponding PDMA channel n.
[1]	SWTRIGn	Software Trigger control 0: No operation 1: Software triggered transfer request Setting this bit will generate a memory-to-memory software transfer request on the corresponding PDMA channel n. It is automatically cleared when a transfer has completely finished.

Bits	Field	Descriptions
[0]	CHnEN	Channel n Enable control 0: Disable the PDMA channel n 1: Enable the PDMA channel n Setting this bit will enable a software or hardware transfer request on the PDMA channel n. It is automatically cleared by hardware when a transfer has completed with the auto-reload function being disabled. However, if the AUTORLn bit is set to 1 to enable the auto-reload function, this bit will be remain high to enable the PDMA channel n function for the next transfer request instead of automatically being cleared by hardware after a transfer has finished.

### PDMA Channel n Source Address Register – PDMACHnSADR (n = 0 ~ 5)

This register specifies the source address of the PDMA channel n.

Offset: 0x004 (0), 0x01C (1), 0x034 (2), 0x04C (3), 0x064 (4), 0x07C (5)

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[31:0]	SADRn	Channel n Source Address The register is used to specify the 32-bit source address of the PDMA channel n.

## PDMA Channel n Destination Address Register – PDMACHnDADR (n = 0 ~ 5)

This register specifies the destination address of the PDMA channel n.

Offset: 0x008 (0), 0x020 (1), 0x038 (2), 0x050 (3), 0x068 (4), 0x080 (5)

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[31:0]	DADRn	Channel n Destination Address The register is used to specify the 32-bit destination address of the PDMA channel n.

## PDMA Channel n Transfer Size Register – PDMACHnTSR (n = 0 ~ 5)

This register is used to specify the block transaction count and block transaction length.

Offset: 0x010 (0), 0x028 (1), 0x040 (2), 0x058 (3), 0x070 (4), 0x088 (5)

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[31:16]	BLKCNTn	Channel n Block Transaction Count BLKCNTn represents the number of block transactions for a channel n complete transfer. The capacity of a complete transfer is the product of the BLKCNTn and BLKLENn values. The maximum BLKCNTn value is 65,535.
[7:0]	BLKLENn	Channel n Block Length The BLKLENn represents the length of a data block. The data width is defined by the DWIDTHn field in the PDMACHnCR register. The maximum BLKLENn value is 255.



## PDMA Channel n Current Transfer Size Register – PDMACHnCTSR (n = 0 ~ 5)

This register is used to indicate the current block transaction count.

Offset: 0x014 (0), 0x02C (1), 0x044 (2), 0x05C (3), 0x074 (4), 0x08C (5)

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[31:16]	CBLKCNTn	Channel n Current Block Count The CBLKCNTn field is a 16-bit read-only value indicating the number of data blocks that remain to be transferred. After a data block has transferred completely, the CBLKCNTn value will be decreased by 1. Writing a new value to the BLKCNTn field in the PDMACHnTSR register will update the CBLKCNTn field value.

## PDMA Interrupt Status Register – PDMAISR

This register is used to indicate the corresponding interrupt status of the PDMA channel 0 ~ 5.

Offset: 0x120

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
	Reserved		TEISTA5	TCISTA5	HTISTA5	BEISTA5	GEISTA5	TEISTA4
Type/Reset			RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	23	22	21	20	19	18	17	16
	TCISTA4	HTISTA4	BEISTA4	GEISTA4	TEISTA3	TCISTA3	HTISTA3	BEISTA3
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	15	14	13	12	11	10	9	8
	GEISTA3	TEISTA2	TCISTA2	HTISTA2	BEISTA2	GEISTA2	TEISTA1	TCISTA1
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	7	6	5	4	3	2	1	0
	HTISTA1	BEISTA1	GEISTA1	TEISTA0	TCISTA0	HTISTA0	BEISTA0	GEISTA0
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0

Bits	Field	Descriptions
[29], [24], [19], [14], [9], [4]	TEISTAn	Channel n Transfer Error Interrupt Status (n = 0 ~ 5) 0: No Transfer Error occurs 1: Transfer Error occurs This bit is set by hardware and is cleared by writing a "1" into the corresponding interrupt status clear bit in the PDMAISCR register. A Transfer error will occur when the PDMA accesses a system reserved address space or when the PDMA receives a request but the corresponding transfer capacity is equal to zero.
[28], [23], [18], [13], [8], [3]	TCISTAn	Channel n Transfer Complete Interrupt Status (n = 0 ~ 5) 0: No Transfer Completion Occurs 1: Transfer Completion Occurs This bit is set by hardware and is cleared by writing a "1" into the corresponding interrupt status clear bit in the PDMAISCR register. The Transfer Completion event will occur when the PDMA has completed a data transfer task.
[27], [22], [17], [12], [7], [2]	HTISTAn	Channel n Half Transfer Interrupt Status (n = 0 ~ 5) 0: No Half Transfer Event Occurs 1: Half Transfer Event Occurs This bit is set by hardware and is cleared by writing a "1" into the corresponding interrupt status clear bit in the PDMAISCR register. A Half Transfer event will occur when the PDMA has completed half of the data transfer task.
[26], [21], [16], [11], [6], [1]	BEISTAn	Channel n Block Transaction End Interrupt Status (n = 0 ~ 5) 0: No Block Transaction End Event Occurs 1: Block Transaction End Event Occurs This bit is set by hardware and is cleared by writing a "1" into the corresponding interrupt status clear bit in the PDMAISCR register. A Block Transaction End event will occur when the PDMA completes a data block transaction task.

Bits	Field	Descriptions
[25], [20], [15], [10], [5], [0]	GEISTAn	Channel n Global Transfer Interrupt Status (n = 0 ~ 5) 0: No TE, TC, HT or BE event occurs 1: TE, TC, HT, or BE event occurs This bit is set by hardware and is cleared by writing a "1" into the corresponding interrupt status clear bit, GEICLRn, in the PDMAISR register. A Global Transfer Event will occur if any of the BE, HT, TC and TE events occurs. Also clearing any of the BE, HT, TC and TE event interrupt flags will clear the GE interrupt flag. Note that if a "1" is written into the GEICLRn bit in the PDMAISR register to clear the GE interrupt flag, the BE, HT, TC and TE event interrupt flags will also be cleared to 0 together with the GE interrupt status flag.

### PDMA Interrupt Status Clear Register – PDMAISCR

This register is used to clear the corresponding interrupt status bits in the PDMAISR Register.

Offset: 0x128

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
	Reserved		TEICLR5	TCICLR5	HTICLR5	BEICLR5	GEICLR5	TEICLR4	
Type/Reset			WC	0	WC	0	WC	0	WC
	23	22	21	20	19	18	17	16	
	TCICLR4	HTICLR4	BEICLR4	GEICLR4	TEICLR3	TCICLR3	HTICLR3	BEICLR3	
Type/Reset	WC	0	WC	0	WC	0	WC	0	WC
	15	14	13	12	11	10	9	8	
	GEICLR3	TEICLR2	TCICLR2	HTICLR2	BEICLR2	GEICLR2	TEICLR1	TCICLR1	
Type/Reset	WC	0	WC	0	WC	0	WC	0	WC
	7	6	5	4	3	2	1	0	
	HTICLR1	BEICLR1	GEICLR1	TEICLR0	TCICLR0	HTICLR0	BEICLR0	GEICLR0	
Type/Reset	WC	0	WC	0	WC	0	WC	0	WC

Bits	Field	Descriptions
[29], [24], [19], [14], [9], [4]	TEICLRn	Channel n Transfer Error Interrupt Status Clear (n = 0 ~ 5) 0: No Operation 1: Clear the corresponding TEISTAn bit in the PDMAISR register Writing a "1" into the TEICLRn bit will clear the TEISTAn status bit in the PDMAISR register. This bit will be automatically cleared to 0 after a "1" is written.
[28], [23], [18], [13], [8], [3]	TCICLRn	Channel n Transfer Complete Interrupt Status Clear (n = 0 ~ 5) 0: No Operation 1: Clear the corresponding TCISTAn bit in the PDMAISR register Writing a "1" into the TCICLRn bit will clear the TCISTAn status bit in the PDMAISR register. This bit will be automatically cleared to 0 after a "1" is written.
[27], [22], [17], [12], [7], [2]	HTRICLRn	Channel n Half Transfer Interrupt Status Clear (n = 0 ~ 5) 0: No Operation 1: Clear the corresponding HTISTAn bit in the PDMAISR register Writing a "1" into the HTRICLRn bit will clear the HTISTAn status bit in the PDMAISR register. This bit will be automatically cleared to 0 after a "1" is written.

Bits	Field	Descriptions
[26], [21], [16], [11], [6], [1]	BEICLRn	Channel n Block Transaction End Interrupt Status Clear (n = 0 ~ 5) 0: No Operation 1: Clear the corresponding BEISTAn bit in the PDMAISR register Writing a "1" into the BEICLRn bit will clear the BEISTAn status bit in the PDMAISR register. This bit will automatically cleared to 0 after a data "1" is written.
[25], [20], [15], [10], [5], [0]	GEICLRn	Channel n Global Transfer Event Interrupt Status Clear (n = 0 ~ 5) 0: No Operation 1: Clear the TEISTAn, TCISTAn, HTISTAn, BEISTAn and GEISTAn bits in the PDMAISR register Writing a "1" into the GEICLRn bit will clear the GEISTAn status bit together with the TEISTAn, TCISTAn, HTISTAn and BEISTAn bits in the PDMAISR register. This bit will be automatically cleared to 0 after a "1" is written.

### PDMA Interrupt Enable Register – PDMAIER

This register is used to enable or disable the related interrupts of the PDMA channel 0 ~ 5.

Offset: 0x130

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24
	Reserved		TEIE5	TCIE5	HTIE5	BEIE5	GEIE5	TEIE4
Type/Reset			RW	0	RW	0	RW	0
	23	22	21	20	19	18	17	16
	TCIE4	HTIE4	BEIE4	GEIE4	TEIE3	TCIE3	HTIE3	BEIE3
Type/Reset	RW	0	RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8
	GEIE3	TEIE2	TCIE2	HTIE2	BEIE2	GEIE2	TEIE1	TCIE1
Type/Reset	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

Bits	Field	Descriptions
[29], [24], [19], [14], [9], [4]	TEIE n	Channel n Transfer Error Interrupt Enable control (n = 0 ~ 5) 0: Transfer Error interrupt is disabled 1: Transfer Error interrupt is enabled This bit is set and cleared by software.
[28], [23], [18], [13], [8], [3]	TCIE n	Channel n Transfer Complete Interrupt Enable control (n = 0 ~ 5) 0: Transfer Completion interrupt is disabled 1: Transfer Completion interrupt is enabled This bit is set and cleared by software.
[27], [22], [17], [12], [7], [2]	HTIE n	Channel n Half Transfer Interrupt Enable control (n = 0 ~ 5) 0: Half Transfer interrupt is disabled 1: Half Transfer interrupt is enabled This bit is set and cleared by software.

Bits	Field	Descriptions
[26], [21], [16], [11], [6], [1]	BEIEn	Channel n Block Transaction End Interrupt Enable control (n = 0 ~ 5) 0: Block Transaction End interrupt is disabled 1: Block Transaction End interrupt is enabled This bit is set and cleared by software.
[25], [20], [15], [10], [5], [0]	GEIEn	Channel n Global Transfer Event Interrupt Enable control (n = 0 ~ 5) 0: Global Transfer Event interrupt is disabled 1: Global Transfer Event interrupt is enabled This bit is set and cleared by software.

# 25 Divider (DIV)

## Introduction

In order to enhance MCU performance, a divider is implemented within the device.

## Features

- Signed/unsigned 32-bit divider
- Operation in 8 clock cycles, Load in 1 clock cycle
- Division by zero error flag

## Functional Descriptions

The division and modulus functions of the truncated division are related in the following way:

$$A / B = Q \dots R$$

Where “A” is Dividend, “B” is Divisor, “Q” is Quotient and “R” is Remainder. Divider requires a software trigger start signal by controlling the START bit in the CR register. The divider calculation complete flag will be set to 1 after 8 clock cycles, however, if the divisor register data is zero during the calculation, the division by zero error flag will be set to 1.

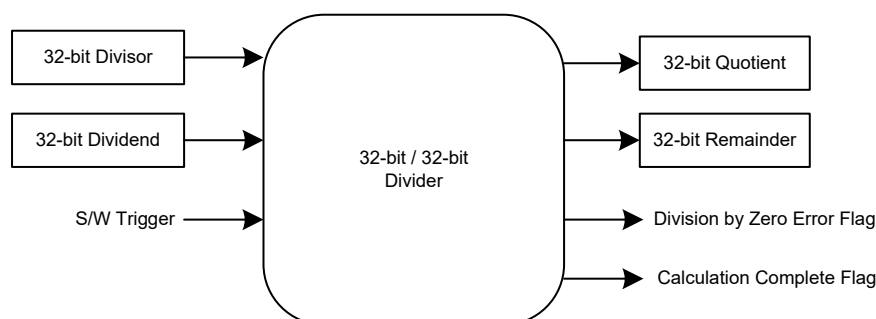


Figure 186. Divider Functional Diagram

## Register Map

The following table shows the DIV registers and reset values.

Table 65. DIV Register Map

Register	Offset	Description	Reset Value
CR	0x000	Divider Control Register	0x0000_0008
DDR	0x004	Dividend Data Register	0x0000_0000
DSR	0x008	Divisor Data Register	0x0000_0000
QTR	0x00C	Quotient Data Register	0x0000_0000
RMR	0x010	Remainder Data Register	0x0000_0000

## Register Descriptions

### Divider Control Register – CR

This register contains the divider calculation complete flag, division by zero error flag and the calculation start control bit.

Offset: 0x000

Reset value: 0x0000\_0008

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

Bits	Field	Descriptions
[3]	COM	Calculation Complete Flag 0: Data are invalid 1: New data are valid When this bit is set to 1 by hardware, it indicates that the divider calculation is completed and data are valid. This bit is cleared to 0 by hardware after the calculation start.
[2]	ZEF	Division By Zero Error Flag 0: Divisor is not zero 1: Divisor is zero This bit is cleared to 0 by hardware after the calculation start.
[0]	START	Calculation Start Control Bit 0: No operation 1: Start the divider calculation Writing 1 to this bit will start the divider calculation.

## Dividend Data Register – DDR

The register contains the dividend of the divider.

Offset: 0x004

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[31:0]	DDR	This bit field is used to specify the dividend of the divider calculation.

## Divisor Data Register – DSR

The register contains the divisor of the divider.

Offset: 0x008

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[31:0]	DSR	This bit field is used to specify the divisor of the divider calculation.



## Quotient Data Register – QTR

The register contains the quotient of the divider calculation result.

Offset: 0x00C

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[31:0]	QTR	This bit field is used to store the quotient of the divider calculation result.

## Remainder Data Register – RMR

The register contains the remainder of the divider calculation result.

Offset: 0x010

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[31:0]	RMR	This bit field is used to store the remainder of the divider calculation result.

# 26 Cyclic Redundancy Check (CRC)

## Introduction

The CRC (Cyclic Redundancy Check) calculation unit is an error detection technique test algorithm and is used to verify data transmission or storage data correctness. A CRC calculation takes a data stream or a block of data as input and generates a 16-bit or 32-bit output remainder. Ordinarily, a data stream is suffixed by a CRC code and used as a checksum when being sent or stored. Therefore, the received or restored data stream is calculated by the same generator polynomial as described above. If the new CRC code result does not match the one calculated earlier, that means data stream contains a data error.

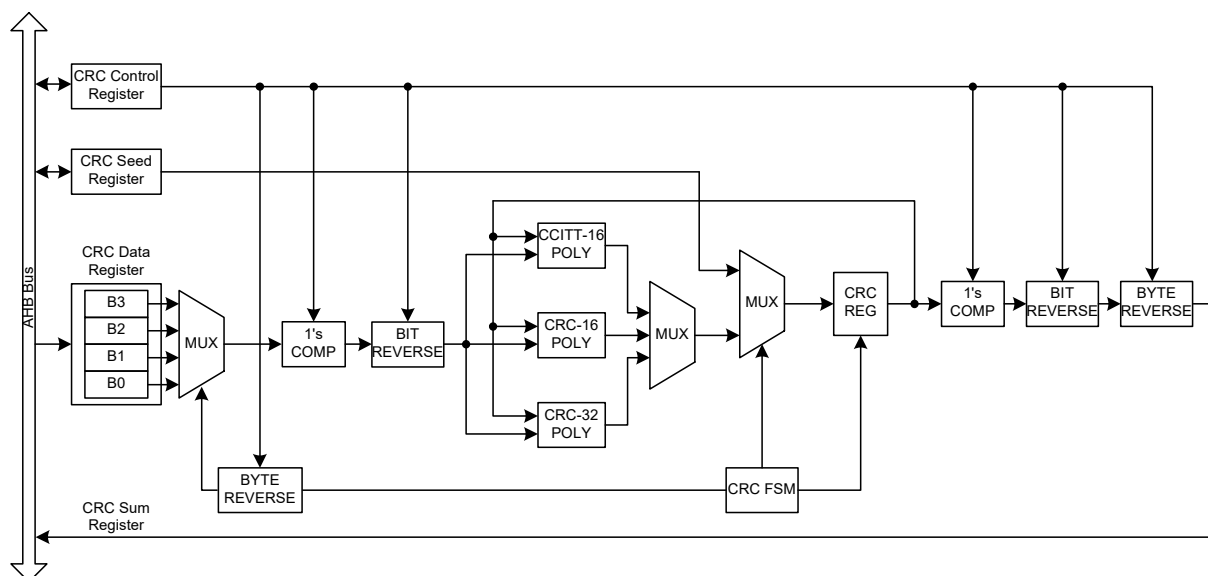


Figure 187. CRC Block Diagram

## Features

- Supports CRC16 polynomial:  $0x8005$ ,  $X^{16} + X^{15} + X^2 + 1$
- Supports CCITT CRC16 polynomial:  $0x1021$ ,  $X^{16} + X^{12} + X^5 + 1$
- 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, this function is only available for the HT32F54243/HT32F54253 devices

## Functional Descriptions

This unit only enables the calculation in the CRC16, CCITT CRC16 and IEEE-802.3 CRC32 polynomials. In this unit, the generator polynomial is fixed to the numeric values for those modes; therefore, the CRC value based on other generator polynomials cannot be calculated.

### CRC Computation

The CRC calculation unit has a 32-bit write CRC data register (CRCDR) and a read CRC checksum register (CRCCSR). The CRCDR register is used to input new data (write access) and the CRCCSR register is used to hold the result of the previous CRC calculation (read access). Each write operation to the CRCDR register creates a combination of the previous CRC value (stored in CRCCSR) and the new one. The CRC block diagram is shown as the figure above. The CRC unit calculates the CRC data register (CRCDR) value byte by byte and the default byte and bit order is big-endian. The CRCDR register can be written by word, right-aligned half-word and right-aligned byte. For the other registers only 32-bit access is allowed. The duration of the computation depends on data width:

- 4 AHB clock cycles for 32-bit data input
- 2 AHB clock cycles for 16-bit data input
- 1 AHB clock cycle for 8-bit data input

### Byte and Bit Reversal for CRC Computation

The byte reordering and byte-level bit reversal operation can be occurred before the data is used in the CRC calculation or after the CRC checksum output. They are configurable using the corresponding setting field of the CRCCR register. These operations occur on word or half-word writes. The hardware ignores the DATBYRV bit of the CRCCR register during any byte writes but the bit reversal setting DATBIRV are still applied to the byte. The accompanying figure shows the byte and bit reversal operation example.

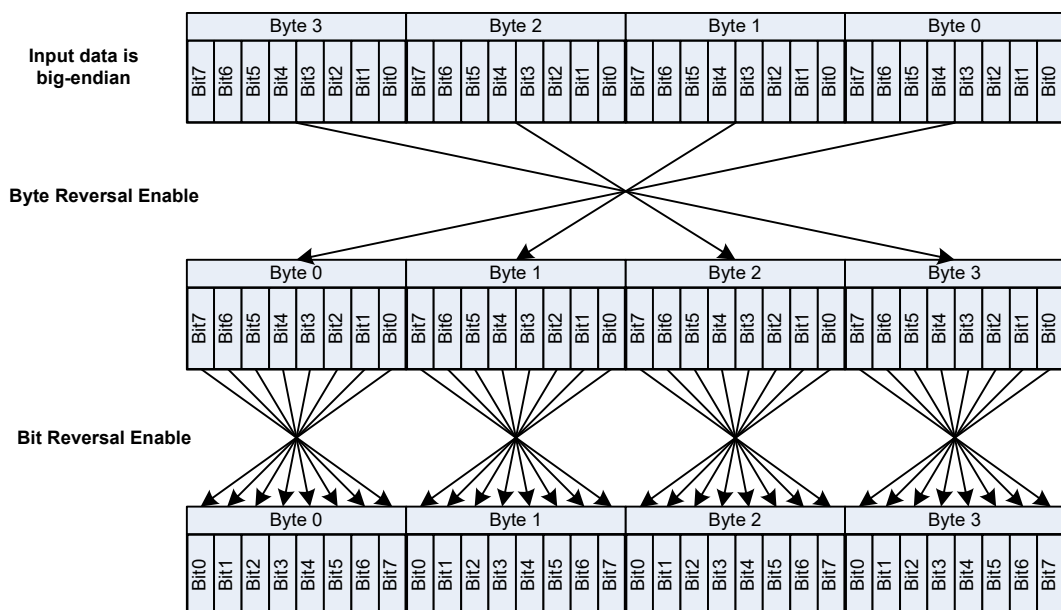


Figure 188. CRC Data Bit and Byte Reversal Example

## CRC with PDMA

A PDMA channel with software trigger may be used to transfer data into the CRC unit. If a huge block data needs to be calculated, the recommended PDMA model is to use the PDMA to transfer all available words of data and use software writes to transfer the other remaining bytes. To write data into the CRC unit, the PDMA should use word access method to transfer data from the source location of memory to the CRC data register (CRCDR) in 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 66. CRC Register Map**

Register	Offset	Description	Reset Value
CRCCR	0x000	CRC Control Register	0x0000_0000
CRCSDR	0x004	CRC Seed Register	0x0000_0000
CRCCSR	0x008	CRC Checksum Register	0x0000_0000
CRCDR	0x00C	CRC Data Register	0x0000_0000

## Register Descriptions

### CRC Control Register – CRCCR

This register specifies the corresponding CRC function enable control.

Offset: 0x000

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[7]	SUMCMPL	1's Complement operation on Checksum Output 0: Disable 1: Enable
[6]	SUMBYRV	Byte Reverse operation on Checksum Output 0: Disable 1: Enable

Bits	Field	Descriptions
[5]	SUMBIRV	Bit Reverse operation on Checksum Output 0: Disable 1: Enable
[4]	DATCMPL	1's Complement operation on Data 0: Disable 1: Enable
[3]	DATBYRV	Byte Reverse operation on Data 0: Disable 1: Enable
[2]	DATBIRV	Bit Reverse operation on Data 0: Disable 1: Enable
[1:0]	POLY	CRC polynomial 00: CRC-CCITT (0x1021) 01: CRC-16 (0x8005) 1x: CRC-32 (0x04C11DB7)

### CRC Seed Register – CRCSDR

This register is used to specify the CRC seed.

Offset: 0x004

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[31:0]	SEED	CRC Seed Data Put the 16/32-bit seed value in this register according to the polynomial setting in the CRCCR register.

## CRC Checksum Register – CRCCSR

This register contains the CRC checksum output.

Offset: 0x008

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[31:0]	CHKSUM	CRC Checksum Data Get the CRC 16/32-bit checksum result from this register according to the polynomial setting in the CRCCR register after all data are written to the CRCDR register.

## CRC Data Register – CRCDR

This register is used to specify the CRC input data.

Offset: 0x00C

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[31:0]	CRCDATA	CRC Input Data Byte, half-word and word writes are allowed. 1's complement, byte reverse and bit reverse operation can be applied.

# 27 LED Controller (LEDC)

## Introduction

The LED controller is used to drive 8-segment digital displays. For the HT32F54231/HT32F54241 devices, the LED controller can drive up to eight 8-segment digital displays. For the HT32F54243/HT32F54253 devices, the LED controller can drive up to twelve 8-segment digital displays. Users have the flexibility to configure the pin position and number of the COMs according to the digital displays in their application. In a complete frame period, the enabled COMs will be scanned from the lower to the higher. Taking an example of where four 8-segment LEDs are used and where COM0, COM5, COM6 and COM7 are enabled. Here COM0, COM5, COM6 and the COM7 will be scanned successively in this sequence within a complete frame period. The scanning time of each COM port is equal to 1/4 frame, which is subdivided into the dead time duty and the COM duty. Users can adjust the dead time duty to change the LED brightness.

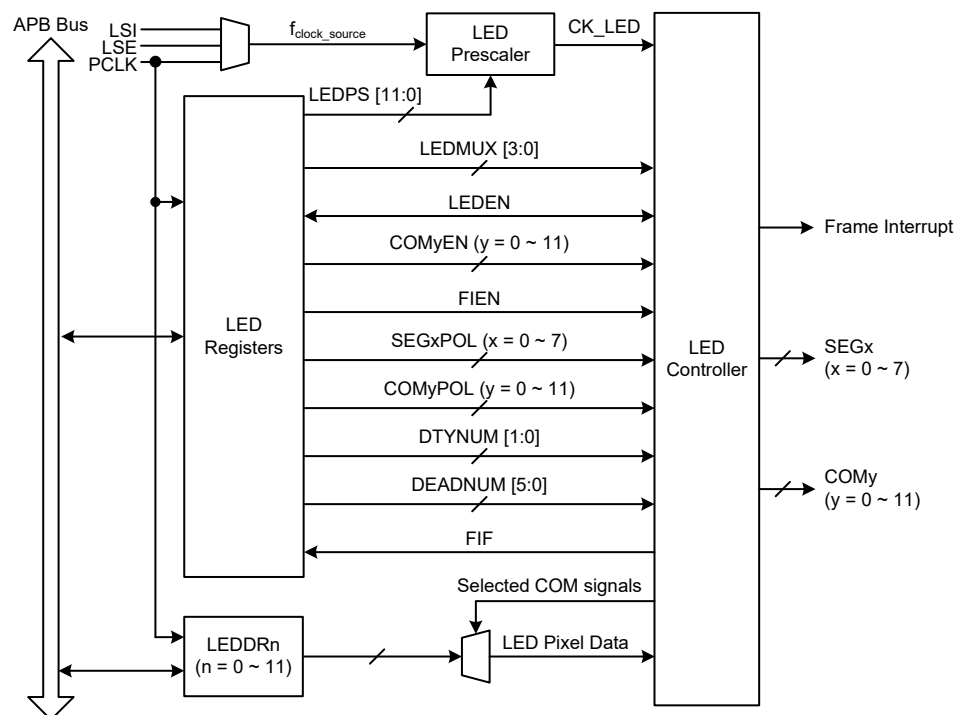


Figure 189. LEDC Block Diagram

## Features

- Supports 8-segment digital displays up to a maximum of N
  - For the HT32F54231/HT32F54241, N = 8
  - For the HT32F54243/HT32F54253, N = 12
- Supports 8-segment digital displays with common anode or common cathode
- Supports frame interrupt
- Three frequency sources: LSI, LSE and PCLK
- The LED light on/off times can be controlled using the dead time setting

## Functional Description

The position of each pixel point is represented by SEGx and COMy. The HT32F54231/HT32F54241 devices can drive up to eight 8-segment digital displays, x = 0 ~ 7, y = 0 ~ 7. The HT32F54243/HT32F54253 devices can drive up to twelve 8-segment digital displays, x = 0 ~ 7, y = 0 ~ 11. The number of COMs to be enabled is N. The following will take N = 4, here COM0, COM5, COM6 and COM7 are enabled as an example to introduce the LEDC functions.

### LEDC Basic Setting

The following show the steps to configure the LEDC drive module.

- Set the LEDSRC, LEDPS and DTYNUM bit fields in the LEDCR register.
- The required COMy can be enabled through the COMyEN bit in the LEDCER register.
- Configure the required COMy and SEGx pins for the LEDC functions using the AFIO function.
- Set the current drive capability of the COMy and SEGx.
- Select the COMy and SEGx output polarity by configuring the LEDPCR register.
- Select the dead time clock number by the DEADNUM bit field in the LEDDTCR register.
- Initialise the LEDDR register.
- The LEDEN bit in the LEDCR is set high to enable the LED driver module.

Except for the LEDEN bit which needs to be enabled during the last step, there is no sequence requirement for the setup steps provided above. When the LEDEN bit is set high, the LEDSRC, LEDPS and DTYNUM bit field contents should remain unchanged. However, the LEDDR and DEADNUM bit fields can be modified.

### LEDC Clock Source Selection

The LEDC clock can be sourced from LSI, LSE or PCLK, which is selected by the LEDSRC bit field in the LEDCR register. The selected clock source passes through a divider, the division ratio of which is selected by the LEDPS bit field in the LEDCR register. The LED clock, CK\_LED, is determined by the following equation:

$$f_{CK\_LED} = (LSI, LSE \text{ or } PCLK) / (LEDPS[11:0] + 1)$$



## LEDC Operational Description

Assuming that N digital displays are used, then each digital display has an operating time of 1/N frame. The required CK\_LED clock number for each digital display scan can be configured as 8, 16, 32 or 64 by the DTYNUM[1:0] bit field in the LEDDTCR register.

$$1/N \text{ frame} = (8, 16, 32 \text{ or } 64) \times \text{CK\_LED clock}$$

Each digital display scanning cycle can be subdivided into a dead time duty and a COM duty. The LED will not be illuminated during the dead time. The dead time clock number is selected by the DEADNUM[5:0] bit field. The duty clock number is selected by the DTYNUM[1:0] bit field. The dead time clock number should be less than the duty clock number. The maximum clock number of the dead time varies and is dependent on the duty clock number.

$$\text{COM duty} = 1/N \text{ frame} - \text{Dead time}$$

Note that 8-segment digital displays have two connection methods, namely common cathode and common anode. For this reason it may be necessary to invert the SEGx or COMy outputs in applications, which can be implemented by configuring the SEGx or COMy polarity control bits.

- The SEGxPOL bit is used to control the SEGx polarity. If SEGxPOL is set to 1, then SEGx will be an inverted output.
- The COMyPOL bit is used to control the COMy polarity. If COMyPOL is set to 1, then COMy will be an inverted output.

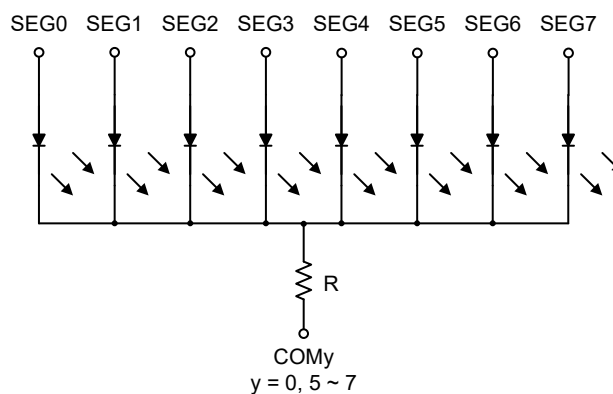
The following will introduce several connection methods in the application.

### Common Cathode 8-segment Digital Display Connection

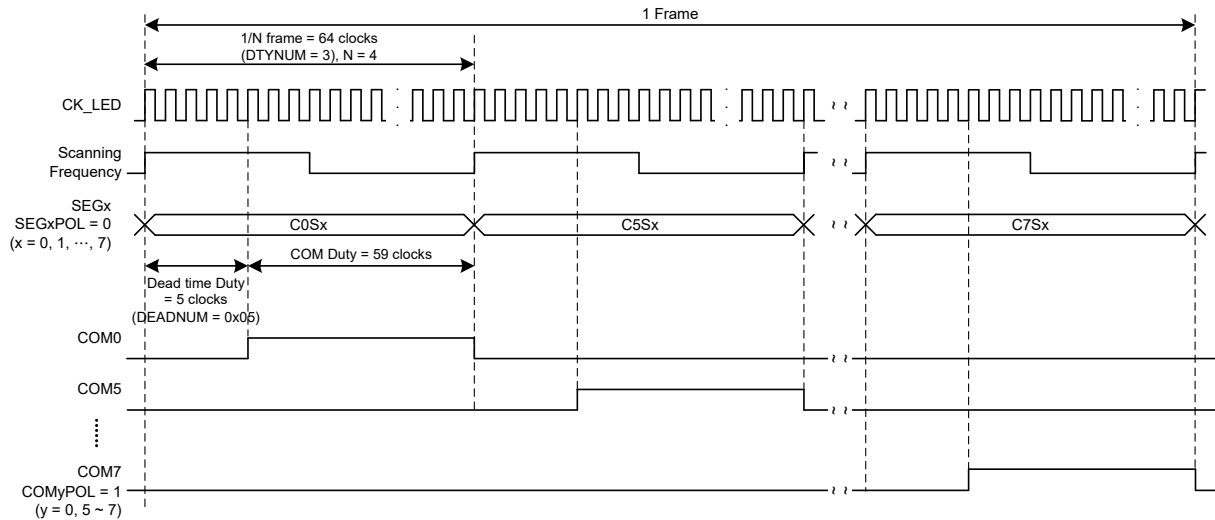
- SEGx output high and COMy output low → LED on
- COMy output high → LED off

The register configurations are as follows:

- Set SEGxPOL = 0 and COMyPOL = 0, the output is non-inverted
- Set DTYNUM[1:0] = b11, the period of each digital display scan is 64 CK\_LED clocks.
- The selected COMs are enabled.



**Figure 190. Common Cathode 8-segment Digital Display Connection**



**Figure 191. Common Cathode 8-segment Digital Display Timing**

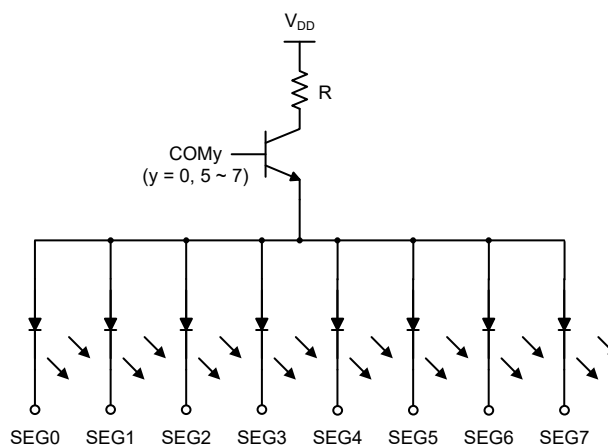
#### Common Anode 8-segment Digital Display + NPN BJT Connection

- SEGx output low and COMy output high → LED on
- COMy output low → LED off

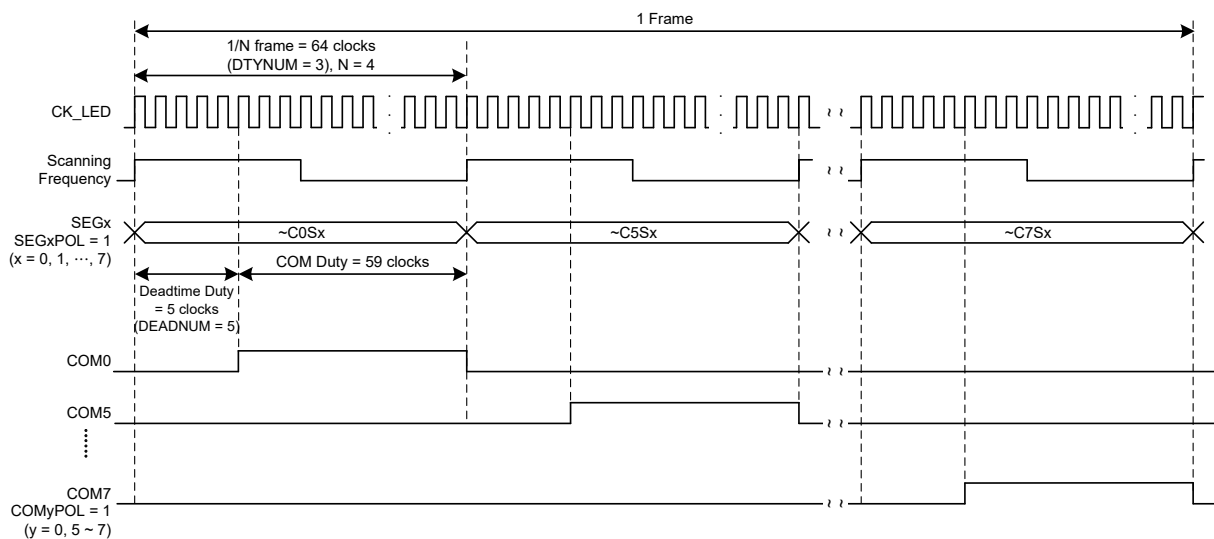
Due to the I/O drive capability limitation, an external transistor is needed to increase the current drive capability if using this connection method.

The register configurations are as follows:

- Set SEGxPOL = 1 and COMyPOL = 1, the output is inverted.
- Set DTYNUM[1:0] = b11, the period of each digital display scan is 64 CK\_LED clocks.
- The selected COMs are enabled.



**Figure 192. Common Anode 8-segment Digital Display + NPN BJT Connection**



**Figure 193. Common Anode 8-segment Digital Display+ NPN BJT Timing**

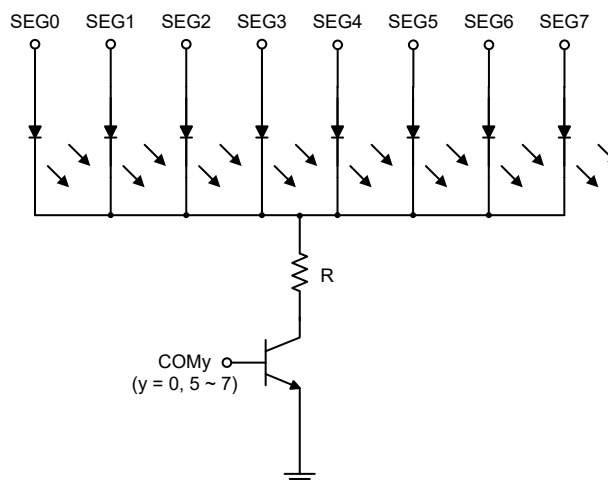
#### Common Cathode 8-segment Digital Display Connection + NPN Transistor

- SEG output high and COM output high → LED on
- COM output low → LED off

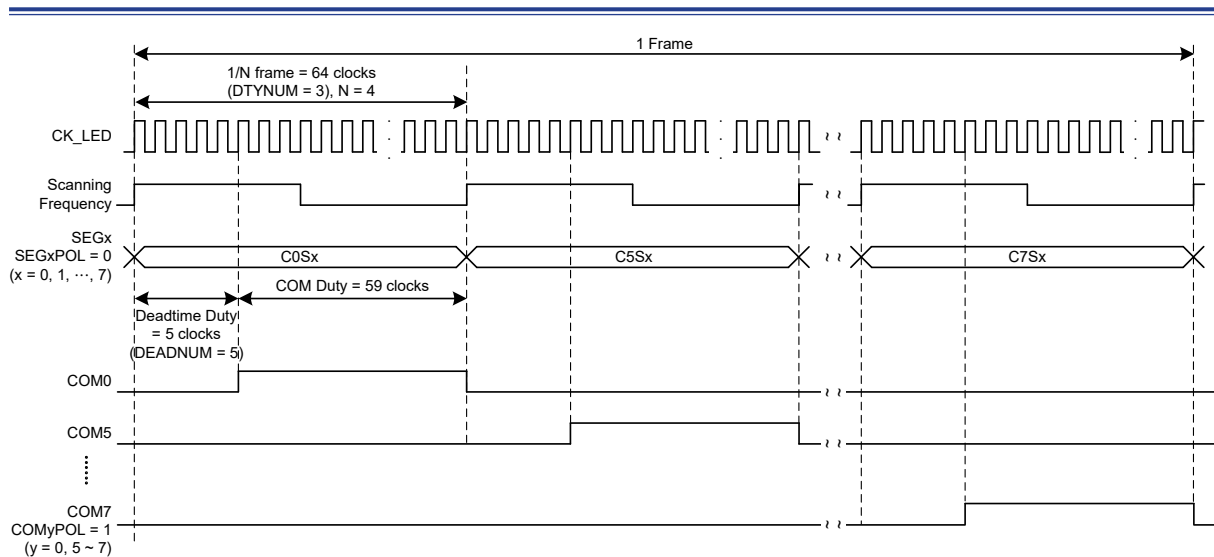
Connect an external transistor to prevent the LED current from affecting the device.

The register configurations are as follows.

- Set SEGxPOL = 1 and COMyPOL = 1, the output is inverted.
- Set DTYNUM[1:0] = b11, the period of each digital display scan is 64 CK\_LED clocks.
- The selected COMs are enabled.



**Figure 194. Common Cathode 8-segment Digital Display + NPN Transistor Connection**



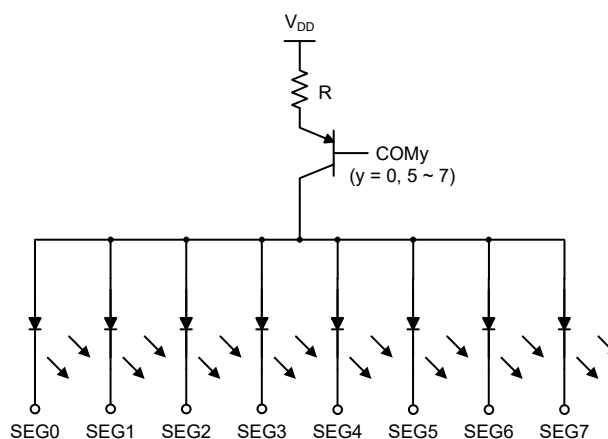
**Figure 195. Common Cathode 8-segment Digital Display + NPN Transistor Timing**

#### Common Anode 8-segment Digital Display + PNP BJT Connection

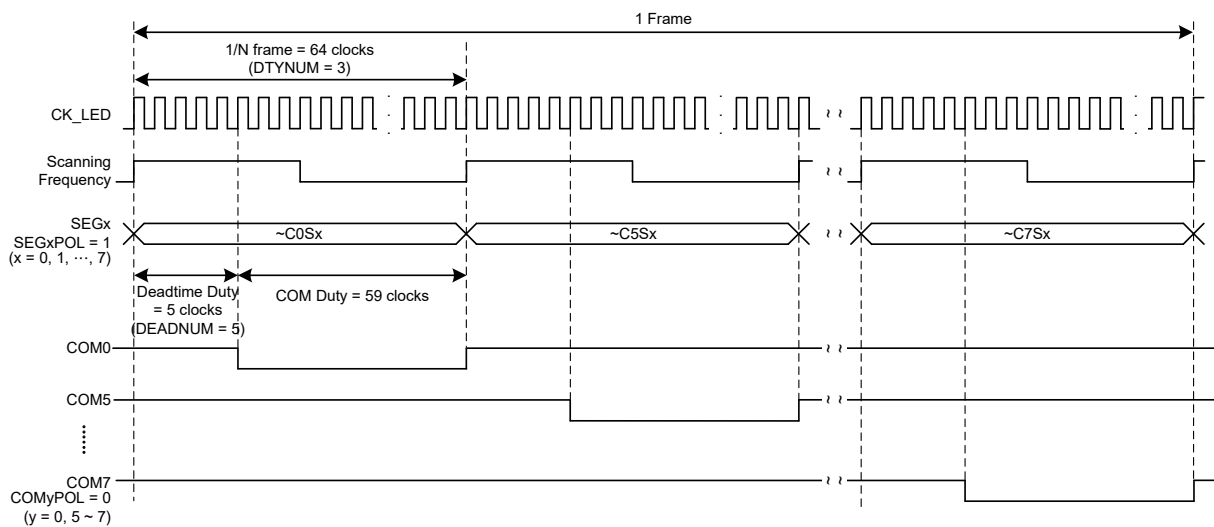
- SEG output low and COM output low → LED on
- COM output high → LED off

The register configurations are as follows:

- Set SEGxPOL = 1, the output is inverted. Set COMyPOL = 0, the output is non-inverted.
- Set DTYNUM[1:0] = b11, the period of each digital display scan is 64 CK\_LED clocks.
- The selected COMs are enabled.



**Figure 196. Common Anode 8-segment Digital Display + PNP BJT Connection**

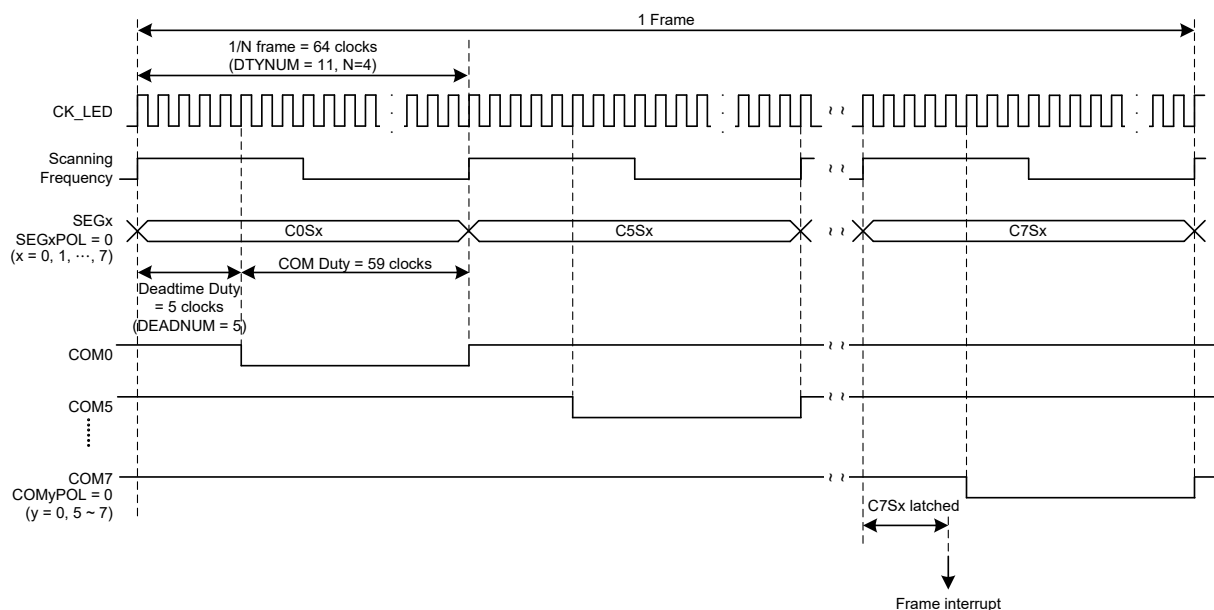


**Figure 197. Common Anode 8-segment Digital Display + PNP BJT Timing**

## LEDC Frame Interrupt

The frame interrupt can be enabled after the pixel data of the last COM duty is latched. Users can update the pixel data or adjust the dead time duty in the interrupt service routine.

The Frame interrupt flag, FIF, in the LEDSR register is set by hardware and reset by software by writing a “1” to it.



**Figure 198. Frame Interrupt Diagram**

## LEDC Data Update Method

After the LEDC function is enabled, if the displayed characters need to be updated, it is recommended to use a frame interrupt to update the pixel data. Using this method will more effectively maintain the display integrity of each frame. The corresponding relationship between the LED pixel data and (SEGx, COMy) is shown in the table below.

**Table 67. LED Pixel Data and (SEGx, COMy) Relationship**

LEDDRn	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
COMy	CnS7	CnS6	CnS5	CnS4	CnS3	CnS2	CnS1	CnS0

Note: SEGx corresponds to the CnSx bit in the LEDDRn register and COMy corresponds to the LEDDRn register. The CnSx value represents the pixel data of the 8-segment digital display.

## Frame Rate Calculation

The calculation formula of the frame rate frequency is as follows:

$$f_{\text{frame}} = f_{\text{CK\_LED}} / (\text{Duty clocks} \times N)$$

Where N is the number of COMs selected. For the HT32F54231/HT32F54241, the maximum value of N is 8. For the HT32F54243/HT32F54253, the maximum value of N is 12. The duty clock number can be configured as 8, 16, 32 or 64 by the DTYNUM[1:0] bit field in the LEDDTCR register. The  $f_{\text{CK\_LED}}$  is calculated by the following formula. The  $f_{\text{clock\_source}}$  can be selected to be sourced from LSI, LSE or PCLK.

$$f_{\text{CK\_LED}} = f_{\text{clock\_source}} / (\text{LEDPS} + 1)$$

### Examples

- (1) If the CK\_LED clock source  $f_{\text{clock\_source}}$  is 32.768 kHz, DTYNUM[1:0] = b11, LEDPS[11:0] = 0x000 and 8 COMs are enabled, then the frame rate can be calculated as follows.

$$\text{Frame rate} = 32768 \text{ Hz} / (64 \times 8) = 64 \text{ Hz}$$

If the operating frequency needs to be greater than 120 Hz, this can be implemented by setting the DTYNUM[1:0] bits to b10. The frame rate can be calculated as follows.

$$\text{Frame rate} = 32768 \text{ Hz} / (32 \times 8) = 128 \text{ Hz}$$

- (2) If the CK\_LED clock source  $f_{\text{clock\_source}}$  is 60 MHz, DTYNUM[1:0] = b11, LEDPS[11:0] = 0x7A0 and 8 COMs are enabled, then the frame rate can be calculated as follows.

$$\text{Frame rate} = [60 \text{ MHz} / (1952 + 1)] / (64 \times 8) = 60 \text{ Hz}$$

If the operating frequency needs to be greater than 120 Hz, this can be implemented by setting LEDPS[11:0] to 0x3CF. The frame rate can be calculated as follows.

$$\text{Frame rate} = [60 \text{ MHz} / (975 + 1)] / (64 \times 8) = 120 \text{ Hz}$$

If the operating frequency needs to be greater than 120 Hz, this can also be implemented by setting the DTYNUM[1:0] bits to b10. The frame rate can be calculated as follows.

$$\text{Frame rate} = [60 \text{ MHz} / (1952 + 1)] / (32 \times 8) = 120 \text{ Hz}$$

## Register Map

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

**Table 68. LEDC Register Map**

Register	Offset	Description	Reset Value
LEDCR	0x000	LED Control Register	0x0000_0000
LEDCER	0x004	LED COM Enable Register	0x0000_0000
LEDPCR	0x008	LED Polarity Control Register	0x0000_0000
LEDIER	0x00C	LED Interrupt Enable Register	0x0000_0000
LEDSR	0x010	LED Status Register	0x0000_0000
LEDDTCR	0x014	LED Dead Time Control Register	0x0000_0000
LEDDR0	0x018	LED Data Register 0	0x0000_0000
LEDDR1	0x01C	LED Data Register 1	0x0000_0000
LEDDR2	0x020	LED Data Register 2	0x0000_0000
LEDDR3	0x024	LED Data Register 3	0x0000_0000
LEDDR4	0x028	LED Data Register 4	0x0000_0000
LEDDR5	0x02C	LED Data Register 5	0x0000_0000
LEDDR6	0x030	LED Data Register 6	0x0000_0000
LEDDR7	0x034	LED Data Register 7	0x0000_0000
LEDDR8	0x038	LED Data Register 8	0x0000_0000
LEDDR9	0x03C	LED Data Register 9	0x0000_0000
LEDDR10	0x040	LED Data Register 10	0x0000_0000
LEDDR11	0x044	LED Data Register 11	0x0000_0000

## Register Descriptions

### LED Control Register – LEDCR

This register is used to control the LEDC clock source, prescaler, duty clock number and LEDC function enable.

Offset: 0x000

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[27:16]	LEDPS	LEDC Clock Prescaler 0x000: CK_LED = $f_{\text{clock\_source}}/1$ 0x001: CK_LED = $f_{\text{clock\_source}}/2$ 0x002: CK_LED = $f_{\text{clock\_source}}/3$ ... 0xFFFF: CK_LED = $f_{\text{clock\_source}}/4096$ The $f_{\text{clock\_source}}$ clock can be sourced from LSI, LSE or PCLK, which is selected by the LEDSRC bit field in the LEDCR register.
[13:12]	DTYNUM	Duty Clock Numbers 00: 8 CK_LED clocks 01: 16 CK_LED clocks 10: 32 CK_LED clocks 11: 64 CK_LED clocks Assuming that N digital displays are used, the number of COMs is N. Each digital display operating time is 1/N frame. The required CK_LED clock number for each digital display scanning cycle can be configured as 8, 16, 32 or 64 by the DTYNUM[1:0] bit field. Each digital display scanning cycle can be subdivided into a dead time duty and a COM duty.
[9:8]	LEDSRC	LEDC Clock Source Selection 00: PCLK 01: LSI 10: LSE 11: Reserved



Bits	Field	Descriptions
[0]	LEDEN	<p>LEDC Enable Bit</p> <p>0: Disable 1: Enable</p> <p>The LEDC state machine will be enabled when this bit is set to 1. If this bit is set to 0, the state machine will continue to output until the current frame operation is completed, then the state of the state machine will be cleared and finally the LEDEN bit will be cleared to zero by the hardware.</p> <p>Therefore, the LEDEN bit must be polled continuously after clearing the LEDEN bit has been executed, the LED controller will not actually cease operation until the LEDEN becomes 0.</p> <p>When the LEDEN bit is set high, the LEDSRC, LEDPS and DTYNUM bit field contents should remain unchanged. Even if new data is written into these bit fields, they are invalid, however the LEDDR and DEADNUM bit fields can be modified. It is recommended to update in the frame interrupt subroutine.</p>

### LED COM Enable Register – LEDCER

This register is used to control the COMy enable, (y = 0 ~ 11).

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				COM11EN	COM10EN	COM9EN	COM8EN	
	7	6	5	4	3	2	1	0	
Type/Reset	COM7EN	COM6EN	COM5EN	COM4EN	COM3EN	COM2EN	COM1EN	COM0EN	
	RW	0	RW	0	RW	0	RW	0	

Bits	Field	Descriptions
[11:0]	COMyEN	<p>COMy Enable Bit (y = 0 ~ 11)</p> <p>0: COMy disable 1: COMy enable</p> <p>In a complete frame period, the enabled COMs will be scanned from low number to high number.</p> <p>Assuming that four 8-segment LEDs are used, the COM7EN, COM6EN, COM5EN and COM0EN bits are set to 1 and the remaining bits are set to zero. Here COM0, COM5, COM6 and COM7 will be scanned successively within a complete frame period. The scanning time of for each COM port is equal to 1/4 frame, which is subdivided into dead time duty and COM duty. Users can adjust the dead time duty to change the LED brightness.</p> <p>Therefore, a complete frame scans COM0 → COM5 → COM6 → COM7 in sequence. If no COMyEN bit is set, the LEDC will not operate after LEDEN is enabled.</p>

## LED Polarity Control Register – LEDPCR

This register controls the polarity of the COMy and SEGx. (x = 0 ~ 7, y = 0 ~ 11)

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		
	SEG7POL	SEG6POL	SEG5POL	SEG4POL	SEG3POL	SEG2POL	SEG1POL	SEG0POL		
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0
	15	14	13	12	11	10	9	8		
	Reserved				COM11POL	COM10POL	COM9POL	COM8POL		
Type/Reset					RW	0	RW	0	RW	0
	7	6	5	4	3	2	1	0		
	COM7POL	COM6POL	COM5POL	COM4POL	COM3POL	COM2POL	COM1POL	COM0POL		
Type/Reset	RW	0	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[23:16]	SEGxPOL	SEGx Output Polarity (x = 0 ~ 7) 0: Output non-inverted 1: Output inverted
[11:0]	COMyPOL	COMy Output Polarity (y = 0 ~ 11) 0: Output non-inverted 1: Output inverted

## LED Interrupt Enable Register – LEDIER

This register is used to control the frame interrupt enable.

Offset: 0x00C

Reset value: 0x0000\_0000

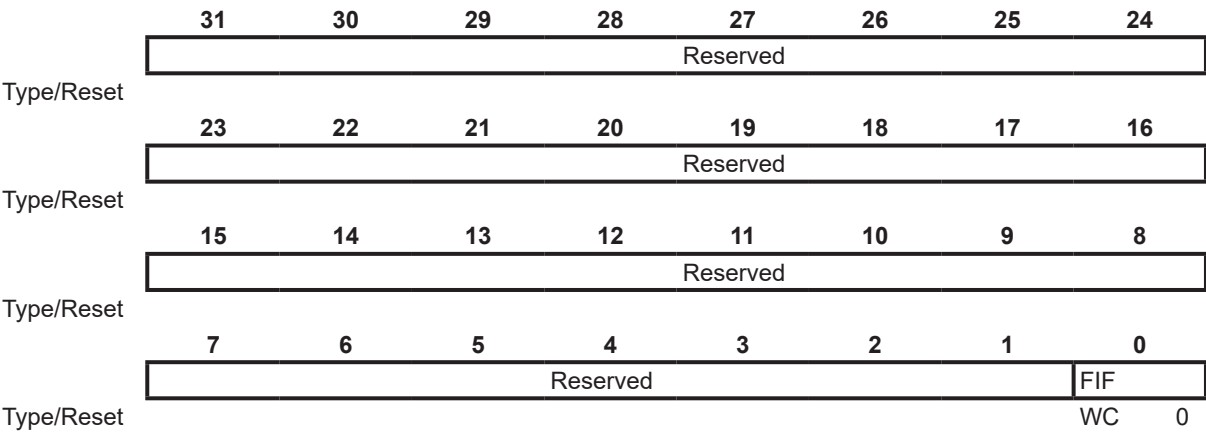
	31	30	29	28	27	26	25	24
Type/Reset	Reserved							
	23	22	21	20	19	18	17	16
Type/Reset	Reserved							
	15	14	13	12	11	10	9	8
Type/Reset	Reserved							
	7	6	5	4	3	2	1	0
Type/Reset	Reserved							FIEN
								RW 0

Bits	Field	Descriptions
[0]	FIEN	Frame Interrupt Enable 0: Disable 1: Enable

LED Status Register – LEDSR

This register specifies the frame interrupt flag.

Offset: 0x010  
Reset value: 0x0000\_0000



Bits	Field	Descriptions
[0]	FIF	Frame interrupt flag 0: No frame interrupt occurs 1: Frame interrupt occurs Set by hardware and reset by software writing 1.

## LED Dead Time Control Register – LEDDTCR

This register specifies the dead time duty.

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		DEADNUM					
			RW	0	RW	0	RW	0
					RW	0	RW	0
						RW	0	RW
							RW	0

Bits	Field	Descriptions
[5:0]	DEADNUM	<p>Dead time Clock Numbers</p> <p>000000: 0 CK_LED clock</p> <p>000001: 1 CK_LED clock</p> <p>000010: 2 CK_LED clocks</p> <p>000011: 3 CK_LED clocks</p> <p>...</p> <p>111111: 63 CK_LED clocks</p> <p>COM duty clock number = Duty clock number – Dead time clock number</p> <p>The dead time clock number should be less than duty clock number.</p> <p>The maximum clock number of the dead time varies and is dependent upon the duty clock number.</p> <p>If DTYNUM bits are set to 00, that is, the duty cycle clock is 8 CK_LED clocks, only bit 2 ~ bit 0 of the DEADNUM bit field is valid.</p> <p>If DTYNUM bits are set to 01, that is, the duty cycle clock is 16 CK_LED clocks, only bit 3 ~ bit 0 of the DEADNUM bit field is valid.</p> <p>If DTYNUM bits are set to 10, that is, the duty cycle clock is 32 CK_LED clocks, only bit 4 ~ bit 0 of the DEADNUM bit field is valid.</p> <p>If DTYNUM bits are set to 11, that is, the duty cycle clock is 64 CK_LED clocks, the bits of the DEADNUM bit field are all valid.</p>

## LED Data Register n – LEDDRn (n = 0 ~ 11)

This register specifies that the LCD pixel points that are represented by SEGx and COMy are to be illuminated.  
(x = 0 ~ 7, y = n = 0 ~ 11)

offset: 0x018 ~ 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							
	7	6	5	4	3	2	1	0
Type/Reset	CnS7	CnS6	CnS5	CnS4	CnS3	CnS2	CnS1	CnS0
	RW	0	RW	0	RW	0	RW	0
	0		0		0		0	

Bits	Field	Descriptions
[7:0]	CnSx	LED Pixel Data (x = 0 ~ 7, n = 0 ~ 11) 0: Output low 1: Output high

# 28 Touch Key – TKEY

## Introduction

The touch key function is fully integrated and requires no external components, allowing touch key functions to be implemented by the simple manipulation of internal registers. Keys are organised into several groups, with each group known as a module Mn. For the HT32F54231/HT32F54241 devices contain touch key up to 24, having a module number M0 to M5. For the HT32F54243/HT32F54253 contain touch key up to 28, having a module number M0 to M6. Each module is a fully independent set of four Touch Keys and each Touch Key has its own oscillator. Each module contains its own control logic circuits and register set. Examination of the register names will reveal the module number it is referring to.

## Features

- Four key oscillator frequencies: 1 MHz / 3 MHz / 7 MHz / 11 MHz
- 1024 level reference oscillator internal capacitor for frequency matching
- Single 16-bit C/F Counter
- Three scan modes: Manual mode, Auto scan mode and Periodic auto scan mode
- Hardware upper or lower threshold comparators
- Keys are organised into several groups, with each group known as a module
  - For the HT32F54231/HT32F54241, having a module number, M0 to M5
  - For the HT32F54243/HT32F54253, having a module number, M0 to M6
- Each module is a fully independent set of four Touch Keys and each Touch Key has its own oscillator

## Functional Description

When a finger touches or is in proximity to a touch pad, the capacitance of the pad will increase. By using this capacitance variation to change slightly the frequency of the internal sense oscillator, touch actions can be sensed by measuring these frequency changes. Using an internal programmable divider the reference clock is used to generate a fixed time period. By counting the number of generated clock cycles from the sense oscillator during this fixed time period touch key actions can be determined.

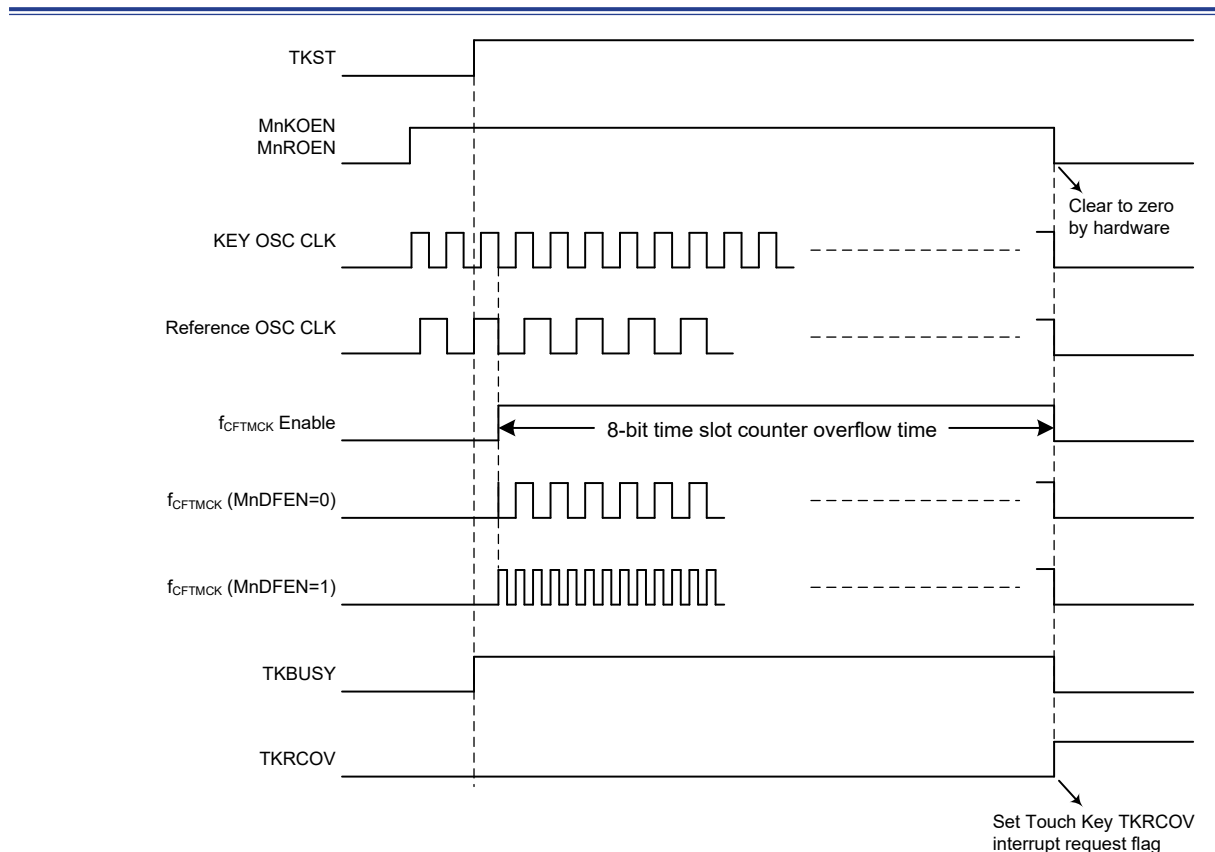
Each touch key module contains four touch key inputs which are either dedicated touch key pins or are shared logical I/O pins. If the desired function is selected by the AFIO configuration. Each touch key has its own independent sense oscillator. There are therefore four sense oscillators within each touch key module.

During this reference clock fixed interval, the number of clock cycles generated by the sense oscillator is measured, and it is this value that is used to determine if a touch action has been made or not.

There are three scan modes contained for the touch key function. The auto scan mode, the periodic auto scan mode and the manual scan mode are selected using the TKMOD bit filed in the TKCR register.

## Manual Mode

In the manual scan mode, the reference oscillator capacitor value should be properly configured before the scan operation begins and the touch key module 16-bit C/F counter value should be read by application program after the scan operation finishes. In the Manual mode, at the end of the fixed reference clock time interval a Touch key interrupt signal will be generated. The timing of touch sensor oscillation and reference oscillation is as follow:



**Figure 199. Touch Key Manual Scan Mode Timing Diagram**

All modules use the same started signal, TKST, in the TKCR register. The touch key module 16-bit C/F counter, touch key function 16-bit counter, 5-bit time slot unit period counter in all modules will be automatically cleared when the TKST bit is cleared to zero, but the 8-bit programmable time slot counter will not be cleared. The overflow time is set by users. When the TKST bit changes from low to high, the 16-bit C/F counter, touch key function 16-bit counter, 5-bit time slot unit period counter and 8-bit time slot timer counter will be automatically switched on.

The key oscillator and reference oscillator in all modules will be automatically stopped and the 16-bit C/F counter, touch key function 16-bit counter, 5-bit time slot unit period counter and 8-bit time slot timer counter will be automatically switched off when the time slot counter overflows. The clock source for the time slot counter is sourced from the module n reference oscillator or the touch key function clock frequency,  $f_{TKCLK}$ , divided by 32, 64 or 128, which is selected using the MnTSS bit in the TKMnCR register. The reference oscillator and key oscillator will be enabled by setting the MnROEN bit and MnKOEN bits in the TKMnCR register.



When the time slot counter in all the touch key modules or in the touch key module 0 overflows, an actual touch key TKRCOV interrupt will take place. The touch keys mentioned here are the keys which are enabled.

## Auto Scan Mode

The auto scan mode can minimise the load of the application program and improve the touch key scan operation performance. When the TKMOD[1:0] bits are set to b00, the auto scan mode is selected to scan the module keys in a specific sequence determined by the MnSKm[1:0] bits in the TKMnKCFGR register. Each Touch key has its own corresponding reference oscillator capacitance, its capacitance is set by TKMnKmCPR register.

In the auto scan mode the module n key oscillator and reference oscillator which are required to be used will be enabled by hardware automatically when the TKST bit is set from low to high and disabled automatically when the TKBUSY bit changes from high to low. When the TKST bit is set from low to high in the auto scan mode, the internal capacitor value of the reference oscillator for the selected key to be scanned in the time slot 0 will first be set by TKMnKmCPR register. Then the 16-bit C/F counter value will be written into the corresponding register TKMnKmCNTR. After this, the selected key will start to be scanned in time slot 0. At the end of the time slot 0 key scan operation, the reference oscillator internal capacitor value for the next selected key will be set by the next TKMnKmCPR register. Then the 16-bit C/F counter value of the current scanned key will be written into the TKMnKmCNTR register. The whole auto scan operation will sequentially be carried out in the above specific way from time slot 0 to time slot 3. At the end of the time slot 3 key scan operation, the reference oscillator internal capacitor value for the time slot 0 selected key will again be read from the MnKmCPR register. Then the 16-bit C/F counter value will be written into the relevant location of the time slot 3 scanned key in the TKMnKmCNTR register. After all the selected keys are scanned, the TKRCOV bit will be set high and the TKBUSY bit will be cleared to zero as well as an auto scan mode operation is completed.

## Periodic Auto Scan Mode

In addition to those actions mentioned in the auto scan mode, the periodic auto scan mode provides periodic auto scan and C/F counter upper/lower threshold comparison functions. When the TKMOD[1:0] bit field are set to 10 or 11, the periodic auto scan mode is selected to scan the module keys automatically and periodically. Note that this mode can be used in the Deep-Sleep modes, in order to monitor the touch key state and minimise power consumption.

In the periodic auto scan mode the touch key scan operation will be implemented automatically on a periodic basis, which can be determined by the ASMP[1:0] bit field in the TKCR register. The number of touch key scan times depends upon the Periodic auto scan mode time-out period,  $T_{ASMT0}$ , and the periodic auto scan mode period,  $T_{KEY}$ . Each auto scan operation will sequentially be carried out in a specific way from time slot 0 to time slot 3 like the auto scan mode. The reference oscillator internal capacitor value for each time slot selected key will be set by the TKMnKmCPR register. However, only at the end of the last scan operation in the Periodic auto scan mode time-out cycle, the 16-bit C/F counter value for all scanned keys will be written into the corresponding TKMnKmCNTR register.

In addition, the 16 bit upper/lower threshold value for the selected key to be scanned in the time slot 0 will be set by the TKMnKmTHR register before the selected key will start to be scanned in time slot 0. Each touch key has its own independent upper/lower threshold comparator. The upper/lower threshold comparison function will automatically be enabled in the periodic auto scan

mode. When any key C/F counter value is less than the lower threshold if MnKmTHS = 0, or larger than the upper threshold if MnKmTHS = 1, this indicates that the touch key state changes, then the MnKmTHF flag will be set high by the hardware, and an interrupt signal will be generated.

It should be noted that the settings of MnKOEN and MnROEN bits are different in different modes. The relationship among TKST, TKBUSY, MnKOEN and MnROEN is as follows.

Module n & State	MnTSS[1:0] = 0		MnTSS[1:0] = 1 ~ 3	
	TKST: 0 → 1	TKBUSY: 1 → 0	TKST: 0 → 1	TKBUSY: 1 → 0
	MnKOEN / MnROEN	MnKOEN / MnROEN	MnKOEN	MnKOEN
n = 0 M0K3EN ~ M0K0EN ≠ b0000	Set by H/W	Clear by H/W	Set by H/W <sup>(1)</sup>	Clear by H/W <sup>(1)</sup>
n = 1 ~ 6 MnK3EN ~ MnK0EN ≠ b0000	Set by H/W	Clear by H/W	Set by H/W <sup>(1)</sup>	Clear by H/W <sup>(1)</sup>
n = 0 ~ 6 M0K3EN ~ M0K0EN = b0000 & MnK3EN ~ MnK0EN ≠ b0000 (n ≠ 0)	Set by H/W except M0ROEN <sup>(2)</sup>	Clear by H/W	Set by H/W <sup>(1)</sup>	Clear by H/W <sup>(1)</sup>

H/W: Automatically set high or cleared to zero by hardware

Notes: 1. When MnTSS[1:0] ≠ 0, the time slot clock is selected from TKCLK/32, TKCLK/64 or TKCLK/128, the reference oscillator clock does not need to be enabled, the MnROEN bit will not be automatically set high.

2. When a specific module is enabled, its corresponding oscillator is automatically enabled, the MnROEN is automatically set high by the hardware. Since the module 0 reference clock is also used as a reference oscillator stable delay clock source of other module, the M0ROEN bit needs to be turned on manually.

## Touch Key Interrupts

The touch key has two independent interrupts, known as touch key TKRCOV interrupt and touch key module TKTH interrupt. In the manual scan mode, when the time slot counter in all the touch key modules overflows, an actual touch key TKRCOV interrupt will take place. In the auto scan mode, when the touch key auto scan operation is completed, the Touch Key TKRCOV Interrupt request flag, TKRCOVF, will be set. In the periodic auto scan mode, only after the last scan operation in the PST time-out cycle completes the 16-bit C/F counter content is written into the corresponding TKMnKmCNTR register, then the Touch Key TKRCOV Interrupt request flag, TKRCOVF, will be set. The touch keys mentioned here are the keys which are enabled. The 16-bit C/F counter, 16-bit counter, 5-bit time slot unit period counter and 8-bit time slot counter in the module will be automatically cleared. When any key C/F counter value is less than the lower threshold if MnKmTHS = 0, or larger than the upper threshold if MnKmTHS = 1, a touch key module TKTH interrupt will take place.

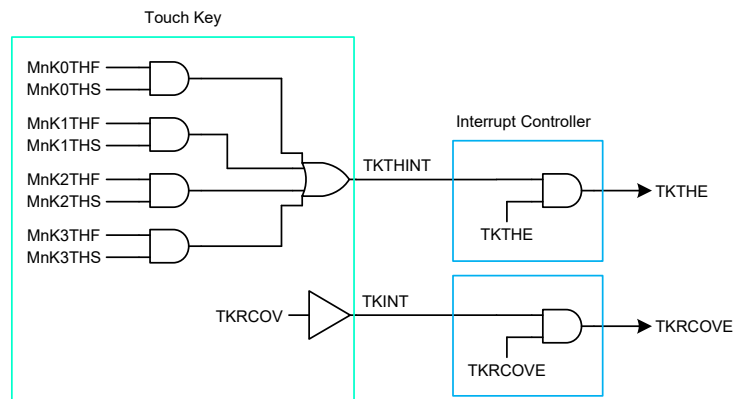


Figure 200. Touch Key Interrupts

## Touch Key Scan Operation Flowchart

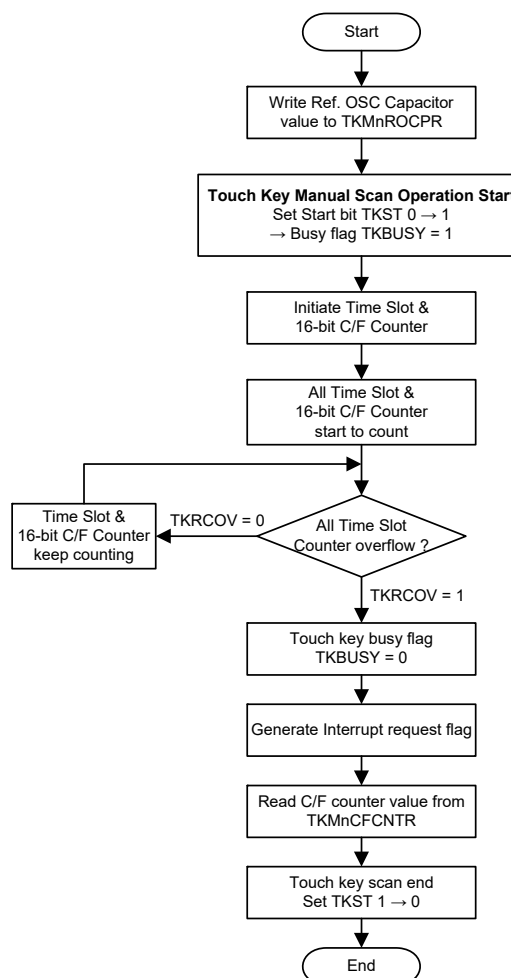


Figure 201. Touch Key Manual Scan Mode Flowchart – TKMOD[1:0] = 01

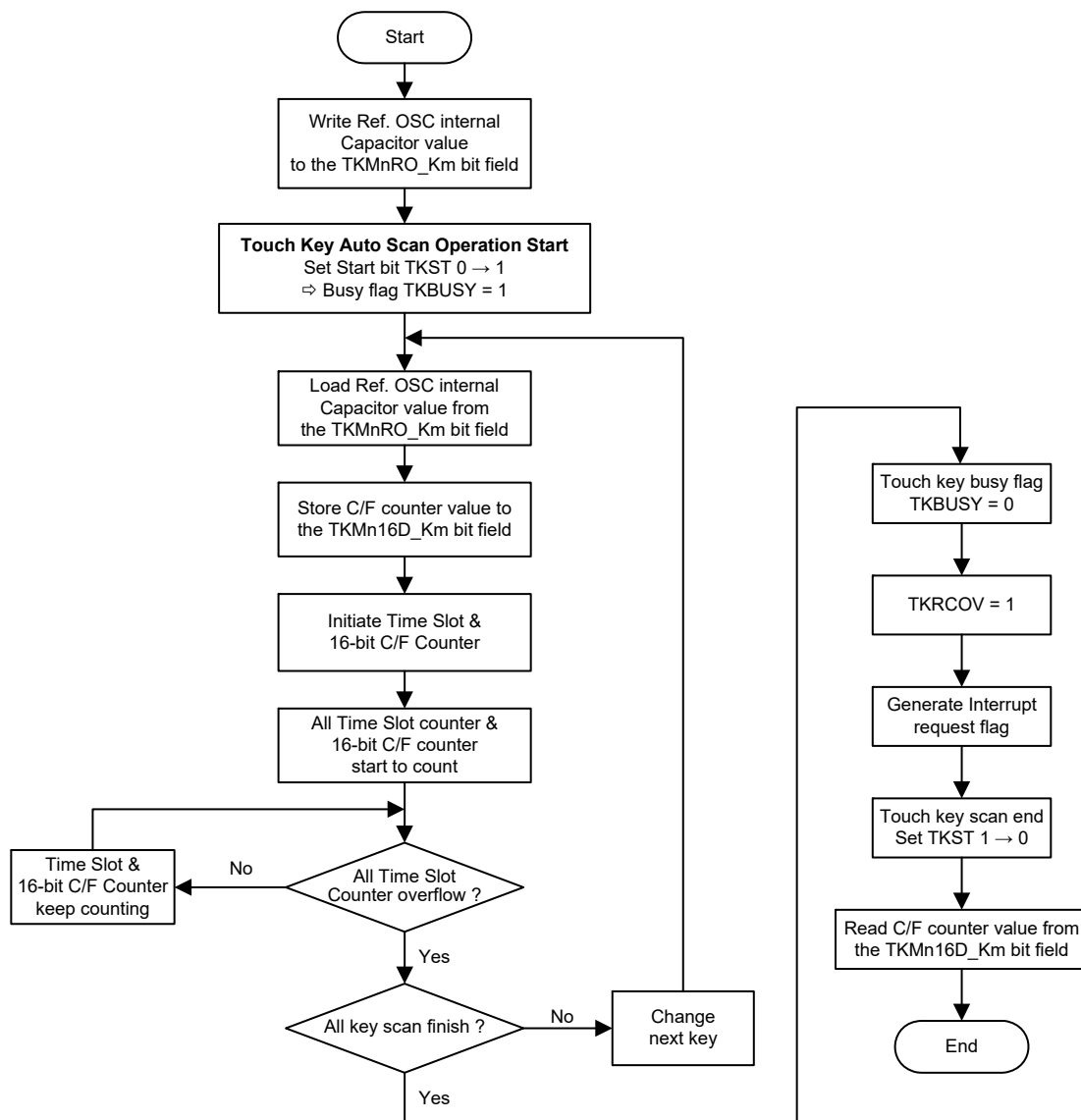


Figure 202. Touch Key Auto Scan Mode Flowchart – TKMOD[1:0] = 00

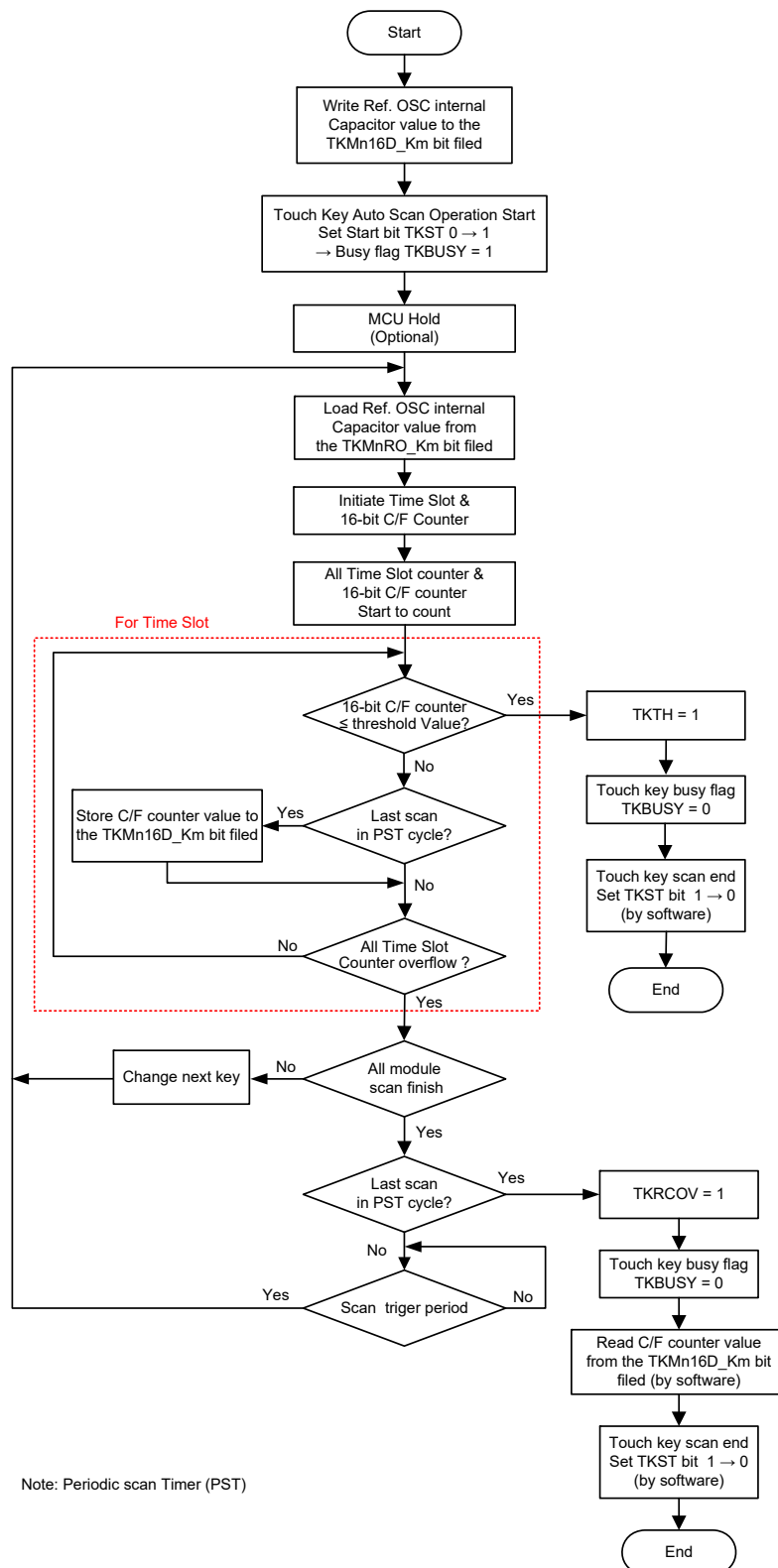


Figure 203. Touch Key Periodic Auto Scan Mode Flowchart – TKMOD[1:0] = 10 ~ 11

## Register Map

The following table shows the touch key registers and reset values.

**Table 69. Touch Key Register Map**

Register	Offset	Description	Reset Value
<b>Touch Key Global Registers</b>			
TKCR	0x000	Touch Key Control Register	0x0000_091A
TKCNTR	0x004	Touch Key Counter Register	0x0000_0000
TKTSCRR	0x008	Touch Key Time Slot Counter Reload Register	0x0000_0000
TKIER	0x00C	Touch Key Interrupt Enable Register	0x0000_0000
TKSR	0x010	Touch Key Status Register	0x0000_0020
<b>Touch Key Module n Registers, n = Module serial number</b>			
TKMnCR	0x100 × (n+1)+0x000	Touch Key Module n Control Register	0x0000_0000
TKMnKCFGR	0x100 × (n+1)+0x004	Touch Key Module n Key Configuration Register	0x00E4_0000
TKMnSR	0x100 × (n+1)+0x008	Touch Key Module n Status Register	0x0000_0000
TKMnROCPR	0x100 × (n+1)+0x00C	Touch Key Module n Reference Oscillator Capacitor Register	0x0000_0000
TKMnK3CPR	0x100 × (n+1)+0x010	Touch Key Module n Key 3 Capacitor Register	0x0000_0000
TKMnK2CPR	0x100 × (n+1)+0x014	Touch Key Module n Key 2 Capacitor Register	0x0000_0000
TKMnK1CPR	0x100 × (n+1)+0x018	Touch Key Module n Key 1 Capacitor Register	0x0000_0000
TKMnK0CPR	0x100 × (n+1)+0x01C	Touch Key Module n Key 0 Capacitor Register	0x0000_0000
TKMnCFCNTR	0x100 × (n+1)+0x020	Touch Key Module n C/F Counter Register	0x0000_0000
TKMnK3CNTR	0x100 × (n+1)+0x024	Touch Key Module n Key 3 Counter Register	0x0000_0000
TKMnK2CNTR	0x100 × (n+1)+0x028	Touch Key Module n Key 2 Counter Register	0x0000_0000
TKMnK1CNTR	0x100 × (n+1)+0x02C	Touch Key Module n Key 1 Counter Register	0x0000_0000
TKMnK0CNTR	0x100 × (n+1)+0x030	Touch Key Module n Key 0 Counter Register	0x0000_0000
TKMnK3THR	0x100 × (n+1)+0x034	Touch Key Module n Key 3 Threshold Register	0x0000_0000
TKMnK2THR	0x100 × (n+1)+0x038	Touch Key Module n Key 2 Threshold Register	0x0000_0000
TKMnK1THR	0x100 × (n+1)+0x03C	Touch Key Module n Key 1 Threshold Register	0x0000_0000
TKMnK0THR	0x100 × (n+1)+0x040	Touch Key Module n Key 0 Threshold Register	0x0000_0000

## Register Descriptions

### Touch Key Control Register – TKCR

This register specifies the touch key clock source, reference oscillator stable delay time, periodic auto scan mode time-out period,  $T_{ASMT0}$ , periodic auto scan mode period,  $T_{KEY}$ , touch key function 16-bit counter clock source, touch key oscillator and reference oscillator frequency, touch key scan mode and touch key detection enable control.

Offset: 0x000

Reset value: 0x0000\_091A

	31	30	29	28	27	26	25	24							
	TKCLKSEL		Reserved												
Type/Reset	RW		0												
	23	22	21	20	19	18	17	16							
	Reserved														
Type/Reset															
	15	14	13	12	11	10	9	8							
	RODLY				ASMT0			ASMP							
Type/Reset	RW	0	RW	0	RW	0	RW	1	RW	0	RW	0	RW	1	
	7	6	5	4	3	2	1	0							
	Reserved		TK16S		TKFS		TKMOD		TKST						
Type/Reset	RW		0	RW	0	RW	1	RW	1	RW	0	RW	1	RW	0

Bits	Field	Descriptions
[31]	TKCLKSEL	Touch Key Clock Source Selection 0: PCLK 1: 32 kHz from LSI
[15:13]	RODLY	Reference Oscillator Stable Delay Time Adjust for Touch Key 000: 4 Reference oscillator clock (default) 001: 2 Reference oscillator clock 010: 4 Reference oscillator clock 011: 8 Reference oscillator clock 100: 16 Reference oscillator clock 101: 32 Reference oscillator clock 110: 64 Reference oscillator clock 111: 4 Reference oscillator clock
[12:10]	ASMT0	Periodic Auto Scan Mode Time-out, $T_{ASMT0}$ , Selection 000: $T_{ASMT0} = 2^{13} / f_{LSI}$ 001: $T_{ASMT0} = 2^{14} / f_{LSI}$ 010: $T_{ASMT0} = 2^{15} / f_{LSI}$ (default) 011: $T_{ASMT0} = 2^{16} / f_{LSI}$ 100: $T_{ASMT0} = 2^{17} / f_{LSI}$ 101~111: $T_{ASMT0} = 2^{18} / f_{LSI}$ These three bits determine the division ratio of the Periodic scan timer (PST) source clock, which in turn determines the timeout period. Note: Periodic scan timer, PST, is cleared to zero when the TKST bit is from low to high.

Bits	Field	Descriptions
[9:8]	ASMP	<p>Periodic Auto Scan Mode Period, <math>T_{KEY}</math>, Selection</p> <p>00: <math>T_{ASMT0} / 2</math>  01: <math>T_{ASMT0} / 4</math> (default)  10: <math>T_{ASMT0} / 8</math>  11: <math>T_{ASMT0} / 16</math></p> <p>These bits are used to determine the touch key scan period and only available when the touch key function is configured to operate in the periodic auto scan mode. The number of touch key scan times is obtained by the Periodic auto scan mode time-out period, <math>T_{ASMT0}</math>, and the periodic auto scan mode period, <math>T_{KEY}</math>, using the equation, <math>N = T_{ASMT0} / T_{KEY} + 1</math></p> <p>Note: A Periodic scan timer time-out cycle scans N times, <math>N = (T_{ASMT0} / T_{KEY}) + 1</math> (Default Scan), as follows</p> <p>00: <math>T_{ASMT0} / 2</math>, <math>N = (2 + 1)</math>  01: <math>T_{ASMT0} / 4</math>, <math>N = (4 + 1)</math>  10: <math>T_{ASMT0} / 8</math>, <math>N = (8 + 1)</math>  11: <math>T_{ASMT0} / 16</math>, <math>N = (16 + 1)</math></p> <p>If ASMT0 [2:0] = b010, Periodic scan timer time-out period, <math>T_{ASMT0} = 2^{15} / f_{LSI}</math>, <math>f_{LSI} = 32</math> kHz, <math>T_{ASMT0} = 1.024</math> s.</p>
[6:5]	TK16S	<p>Touch Key Function 16-bit Counter Clock Source Selection</p> <p>00: <math>f_{TKCLK} / 16</math>  01: <math>f_{TKCLK} / 32</math>  10: <math>f_{TKCLK} / 64</math>  11: <math>f_{TKCLK} / 128</math></p> <p>The touch key function clock frequency is <math>f_{TKCLK}</math>, the touch key function clock is set by the TKCLKSEL bit.</p>
[4:3]	TKFS	<p>Touch Key Oscillator and Reference Oscillator Frequency Selection</p> <p>00: 1 MHz  01: 3 MHz  10: 7 MHz  11: 11 MHz</p>



Bits	Field	Descriptions
[2:1]	TKMOD	<p>Touch Key Scan Mode Selection</p> <p>00: Auto scan mode 01: Manual mode (default) 10: Periodic auto scan mode 11: Periodic auto scan mode</p> <p>In the manual scan mode, the reference oscillator capacitor value should be properly configured before the scan operation begins and the touch key module 16-bit C/F counter value should be read by application program after the scan operation finishes. The individual reference oscillator capacitor value and 16-bit C/F counter content for all the scanned keys will be read and written from the TKMnKmCPR and TKMnKmCNTR registers respectively. In the auto scan mode, the keys to be scanned can be arranged in a specific sequence which is determined by the MnSK3[1:0] ~ MnSK0[1:0] bits in the TKMnKCFGR register. The scan operation will not be stopped until all arranged keys are scanned.</p> <p>In the periodic auto scan mode, the touch key scan operation will be implemented automatically on a periodic basis, which can be determined by the ASMP1 ~ ASMP0 bits in the TKCR register. Only at the end of the last scan operation in the Periodic auto scan mode time-out cycle, <math>T_{ASMT0}</math>, the 16-bit C/F counter content for all scanned keys will be written into the corresponding TKMnKCFGR. In addition, when any key C/F counter value is less than the lower threshold if MnKmTHS = 0, or larger than the upper threshold if MnKmTHS = 1, the TKTH signal will be set high. The other actions in this mode are the same as those in the auto scan mode except the above mentioned.</p>
[0]	TKST	<p>Touch Key Detection Start Control</p> <p>0: Stopped or no operation 0→1: Start detection</p> <p>In all modules the 16-bit C/F counter, touch key function 16-bit counter and 5-bit time slot counter will be automatically cleared when this bit is cleared to zero. However, the 8-bit programmable time slot counter will not be cleared. When this bit is changed from low to high, the touch key module 16-bit C/F counter, touch key function 16-bit counter, 5-bit time slot unit period counter and 8-bit time slot counter will be switched on together with the key and reference oscillators to drive the corresponding counters.</p>

## Touch Key Counter Register – TKCNTR

This register is used to store the touch key function 16-bit counter value.

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	TK16D								
	7	6	5	4	3	2	1	0	
Type/Reset	TK16D								
	RO	0	RO	0	RO	0	RO	0	RO
	0	RO	0	RO	0	RO	0	RO	0

Bits	Field	Descriptions
[15:0]	TK16D	<p>Touch Key 16-bit Counter Value</p> <p>This 16-bit counter can be used to calibrate the reference or key oscillator frequency. When the touch key time slot counter overflows in the manual scan mode, this 16-bit counter will be stopped and the counter content will be unchanged. However, this 16-bit counter content will be cleared to zero at the end of the time slot 0, slot 1 and slot 2 but kept unchanged at the end of the time slot 3 in the auto scan mode or the periodic auto scan mode. This register will be cleared to zero when the TKST bit is cleared to zero.</p>

## 28 Touch Key – TKEY

Offset:	0x008
---------	-------

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

April 13, 2023

## Touch Key Interrupt Enable Register – TKIER

This register contains the corresponding Touch Key interrupt enable control bits and the interrupt wake-up function enable control bits.

Offset: 0x00C

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[3]	TKRCOVWUEN	Time Slot Counter Overflow Interrupt Wake-up Function Control 0: Disable 1: Enable If the TKRCOVWUEN is high, the device can be woken up from deep sleep modes when the time slot counter overflow interrupt occurs.
[2]	TKTHWUEN	Touch Key Threshold Match Interrupt Wake-up Function Control 0: Disable 1: Enable If the TKTHWUEN is high, the device can be woken up from deep sleep modes when the Touch Key threshold match interrupt occurs.
[1]	TKRCOVE	8-bit Time Slot Counter Overflow Interrupt Control 0: Disable 1: Enable
[0]	TKTHE	Touch Key Threshold Match Interrupt Control 0: Disable 1: Enable

## Touch Key Status Register – TKSR

This register contains the relevant Touch Key status.

Offset: 0x010

Reset value: 0x0000\_0020

	31	30	29	28	27	26	25	24
Type/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		PCKRDY	TKBUSY	TKCFOV	TK16OV	TKRCOVF	TKTHF
			RO	1 RO	0 WC	0 WC	0 WC	0 WC

Bits	Field	Descriptions
[5]	PCKRDY	Touch Key Function Clock Ready Flag after the Clock Source Is switched 1: Touch Key Function Clock = PCLK 0: Touch Key Function Clock = LSI
[4]	TKBUSY	Touch Key Scan Operation Busy Flag 0: Not busy – No scan operation is executed or scan operation is completed 1: Busy – Scan operation is executing This bit indicates whether the touch key scan operation is executing or not. It is set to 1 when the TKST bit is set from low to high to start the scan operation. In the manual scan mode this bit is cleared to 0 automatically when all module time slot counters is overflow. In the auto scan mode this bit is cleared to 0 automatically when the touch key scan operation is completed. In the periodic auto scan mode this bit is cleared to 0 automatically when the last scan operation in the Periodic auto scan mode time-out cycle is completed, or when any key C/F counter value is less than the lower threshold if MnKmTHS=0, or when the value is larger than the upper threshold if MnKmTHS=1.
[3]	TKCFOV	Touch Key 8-bit and 5-bit Time Slot Counters Overflow Flag 0: Not overflow occurs 1: Overflow occurs Set by hardware when touch key 8-bit and 5-bit time slot counters overflow. Reset by software writing 1. It should be noted that TKCFOV = 1 usually does not occur. If the TKCFOV bit needs to be tested, it can be achieved by disabling the reference oscillator.
[2]	TK16OV	Touch Key 16-bit Counter Overflow Flag 0: Not overflow occurs 1: Overflow occurs Set by hardware when touch key 16-bit counter overflows. Reset by software writing 1.

Bits	Field	Descriptions
[1]	TKRCOVF	<p>8-bit Time Slot Counter Overflow Flag</p> <p>0: No overflow occurs 1: Overflow occurs</p> <p>Set by hardware when 8-bit Time slot overflows. Reset by software writing 1.</p> <p>In the manual scan mode, if all module time slot counters are overflow, the TKRCOV bit and the Touch Key TKRCOV Interrupt request flag, TKRCOVF, will be set and all modules' key and reference oscillators will automatically stop. All touch key modules' 16-bit C/F counter, touch key function 16-bit counter, 5-bit time slot unit period counter and 8-bit time slot counter will be automatically switched off.</p> <p>In the auto scan mode, if all module time slot counters are overflow but the touch key auto scan operation is not completed, the TKRCOV bit will not be set. At this time, all touch key modules 16-bit C/F counter, touch key function 16-bit counter and 5-bit time slot unit period counter will be automatically cleared but the 8-bit time slot counter will be reloaded from the 8-bit time slot counter preload register. When the touch key auto scan operation is completed, the TKRCOV bit and the Touch Key TKRCOV Interrupt request flag, TKRCOVF, will be set and all modules' key and reference oscillators will automatically stop. All touch key modules' 16-bit C/F counter, touch key function 16-bit counter, 5-bit time slot unit period counter and 8-bit time slot counter will be automatically switched off.</p> <p>In the periodic auto scan mode, the TKRCOV bit is cleared to zero during the auto scan operation period. Only at the end of the last scan operation in the Periodic scan timer, PST, time-out cycle, the 16-bit C/F counter content will be written into the corresponding register, and then the TKRCOV bit will be set high by the hardware circuit. The other actions in this mode are the same as those in the auto scan mode except the above mentioned.</p>
[0]	TKTHF	<p>Touch Key Threshold Match Interrupt Flag</p> <p>This bit is set by the hardware when the Touch Key Module 16-bit C/F counter is less than the lower threshold if MnKmTHS = 0, or larger than the upper threshold if MnKmTHS = 1. This bit is cleared to 0 by writing 1.</p>

## Touch Key Module n Control Register – TKMnCR

This register specifies the touch key module n time slot counter clock source, touch key module n reference oscillator enable bit, touch key module n multi-frequency enable bit, touch key module n C/F oscillator frequency hopping function control selection and touch key module n reference and key oscillators hopping frequency selection.

Offset:  $0x100 \times (n + 1) + 0x000$

Reset value: 0x0000\_0000

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

Bits	Field	Descriptions
[9:8]	MnTSS	Touch Key Module n Time Slot Counter Clock Source Selection 00: Touch key module n reference oscillator 01: $f_{TKCLK} / 32$ 10: $f_{TKCLK} / 64$ 11: $f_{TKCLK} / 128$ The touch key function clock frequency is $f_{TKCLK}$ , the touch key function clock is set by the TKCLKSEL bit.
[7]	MnROEN	Touch Key Module n Reference Oscillator Control 0: Disable 1: Enable In the manual scan mode, this bit is used to enable/disable the touch key module n reference oscillator. The MnROEN bit is set by software and is unaffected by the TKST bit setting. If the module n reference oscillator is enabled, it will be disabled automatically when the TKBUSY bit is changed from high to low. In the auto scan mode or the periodic auto scan mode, this bit is controlled by hardware automatically. For the module 0, when the TKST bit is from low to high, the M0ROEN bit will be set high automatically. For the other module n ( $n \neq 0$ ), if the condition that $MnK3EN \sim MnK0EN \neq b0000$ and $MnTSS = 0$ is satisfied, the MnROEN bit will be set high automatically when the TKST bit is from low to high. In other conditions, the MnROEN bit will be unaffected by the TKST bit settings. When the TKBUSY bit is changed from high to low, the MnROEN bit will automatically be cleared to zero to disable the reference oscillator. For all modules, if the condition that $M0K3EN \sim M0K0EN = b0000$ and $MnK3EN \sim MnK0EN \neq b0000$ ( $n \neq 0$ ), the M0ROEN bit needs to be manually set by software.

Bits	Field	Descriptions
[6]	MnKOEN	<p>Touch Key Module n Key Oscillator Control</p> <p>0: Disable 1: Enable</p> <p>In the manual scan mode, this bit is used to enable/disable the module n key oscillator. When the module n key oscillator is enabled, if the relevant key is enabled to be scanned and will be disabled automatically when the TKBUSY bit is changed from high to low.</p> <p>In the auto scan mode or the periodic auto scan mode, this bit is controlled by hardware automatically. The MnKOEN bit will be set high automatically to enable the key oscillator when the TKST bit is from low to high. When the TKBUSY bit is changed from high to low, the MnKOEN bit will automatically be cleared to zero to disable the key oscillator.</p>
[5]	MnDFEN	<p>Touch Key Module n Multi-frequency Control</p> <p>0: Disable 1: Enable</p>
[3]	MnSOFC	<p>Touch Key Module n C/F Oscillator Frequency Hopping Function Control Selection</p> <p>0: The frequency hopping function is controlled by MnSOF bit field 1: The frequency hopping function is controlled by hardware regardless of the MnSOF bit field value</p>
[2:0]	MnSOF	<p>Touch Key Module n Reference and Key Oscillators Hopping Frequency Selection</p> <p>000: 1.020 MHz 001: 1.040 MHz 010: 1.059 MHz 011: 1.074 MHz 100: 1.085 MHz 101: 1.099 MHz 110: 1.111 MHz 111: 1.125 MHz</p> <p>These bits are used to select the touch key oscillator frequency for the hopping function. Note that these bits are only available when the MnSOFC bit is cleared to 0. The frequencies mentioned here are only for the condition where the key and reference oscillator frequency is selected to be 1 MHz, these values will be changed when the external or internal capacitor has different values. Users can adjust the key and reference oscillator frequency in scale when any other frequency is selected.</p>



## Touch Key Module n Key Configuration Register – TKMnKCFGR

This register specifies the touch key module n time slot 0 ~ time slot 3 key scan selection, touch key module n Key 0 ~ Key 3 upper/lower threshold comparison selection and touch key module n Key 0 ~ Key 3 enable control.

Offset:  $0x100 \times (n + 1) + 0x004$

Reset value: 0x00E4\_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	MnSK3		MnSK2		MnSK1		MnSK0		
	RW	1	RW	1	RW	1	RW	0	
	15	14	13	12	11	10	9	8	
Type/Reset	Reserved				MnK3THS	MnK2THS	MnK1THS	MnK0THS	
					RW	0	RW	0	
	7	6	5	4	3	2	1	0	
Type/Reset	Reserved				MnK3EN	MnK2EN	MnK1EN	MnK0EN	
					RW	0	RW	0	

Bits	Field	Descriptions
[23:22]	MnSK3	Touch Key Module n Time Slot 3 key Scan Selection 00: Key 0 01: Key 1 10: Key 2 11: Key 3 These bits are used to select the desired scan key in time slot 3 in the auto scan mode or the periodic auto scan mode. The settings for these bits are invalid when the TKMOD bit field is b01.
[21:20]	MnSK2	Touch Key Module n Time Slot 2 Key Scan Selection 00: Key 0 01: Key 1 10: Key 2 11: Key 3 These bits are used to select the desired scan key in time slot 2 in the auto scan mode or the periodic auto scan mode. The settings for these bits are invalid when the TKMOD bit field is b01.
[19:18]	MnSK1	Touch Key Module n Time Slot 1 Key Scan Selection 00: Key 0 01: Key 1 10: Key 2 11: Key 3 These bits are used to select the desired scan key in time slot 1 in the auto scan mode or the periodic auto scan mode. The settings for these bits are invalid when the TKMOD bit field is b01.

Bits	Field	Descriptions
[17:16]	MnSK0	Touch Key Module n Time Slot 0 Key Scan Selection (TKMOD = 01) 00: Key 0 01: Key 1 10: Key 2 11: Key 3 These bits are used to select the desired scan key in time slot 0 in the auto scan mode or the periodic auto scan mode or used as the multiplexer for scan key selection in the manual mode.
[11]	MnK3THS	Touch Key Module n Key 3 Upper or Lower Threshold Comparison Selection 0: Lower threshold comparison 1: Upper threshold comparison
[10]	MnK2THS	Touch Key Module n Key 2 Upper or Lower Threshold Comparison Selection 0: Lower threshold comparison 1: Upper threshold comparison
[9]	MnK1THS	Touch Key Module n Key 1 Upper or Lower Threshold Comparison Selection 0: Lower threshold comparison 1: Upper threshold comparison
[8]	MnK0THS	Touch Key Module n Key 0 Upper or Lower Threshold Comparison Selection 0: Lower threshold comparison 1: Upper threshold comparison
[3]	MnK3EN	Touch Key Module n Key 3 Enable Control 0: Disable 1: Enable
[2]	MnK2EN	Touch Key Module n Key 2 Enable Control 0: Disable 1: Enable
[1]	MnK1EN	Touch Key Module n Key 1 Enable Control 0: Disable 1: Enable
[0]	MnK0EN	Touch Key Module n Key 0 Enable Control 0: Disable 1: Enable

## Touch Key Module n Status Register – TKMnSR

This register contains the relevant Key 0 to Key 3 status.

Offset:  $0x100 \times (n + 1) + 0x008$

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				MnK3THF	MnK2THF	MnK1THF	MnK0THF
					WC	0	WC	0
						0	WC	0
							0	WC
								0

Bits	Field	Descriptions
[3]	MnK3THF	Touch Key Module n Key 3 Upper or Lower Threshold Comparison Flag 0: Not less than lower threshold or not larger than upper threshold 1: Less than lower threshold or larger than upper threshold This bit is cleared to 0 by writing 1.
[2]	MnK2THF	Touch Key Module n Key 2 Upper or Lower Threshold Comparison Flag 0: Not less than lower threshold or not larger than upper threshold 1: Less than lower threshold or larger than upper threshold This bit is cleared to 0 by writing 1.
[1]	MnK1THF	Touch Key Module n Key 1 Upper or Lower Threshold Comparison Flag 0: Not less than lower threshold or not larger than upper threshold 1: Less than lower threshold or larger than upper threshold This bit is cleared to 0 by writing 1.
[0]	MnK0THF	Touch Key Module n Key 0 Upper or Lower Threshold Comparison Flag 0: Not less than lower threshold or not larger than upper threshold 1: Less than lower threshold or larger than upper threshold This bit is cleared to 0 by writing 1.

## Touch Key Module n Reference Oscillator Capacitor Register – TKMnROCPR

This register is used to store the touch key module n reference oscillator capacitor value.

Offset:  $0x100 \times (n + 1) + 0x00C$

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

Bits	Field	Descriptions
[9:0]	TKMnRO	Touch Key Module n Reference Oscillator Capacitor Value The reference oscillator internal capacitor value = (TKMnRO[9:0] × 50 pF) / 1024.

## Touch Key Module n Key 3 Capacitor Register – TKMnK3CPR

This register specifies the module n Key 3 reference oscillator capacitor value.

Offset:  $0x100 \times (n + 1) + 0x010$

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						TKMnRO_K3	
	7	6	5	4	3	2	1	0
Type/Reset	TKMnRO_K3							
	RW	0	RW	0	RW	0	RW	0
	RW	0	RW	0	RW	0	RW	0

Bits	Field	Descriptions
[9:0]	TKMnRO_K3	Touch Key Module n Key 3 Reference Oscillator Capacitor Value

## Touch Key Module n Key 2 Capacitor Register – TKMnK2CPR

This register specifies the module n Key 2 reference oscillator capacitor value.

Offset:  $0x100 \times (n + 1) + 0x014$

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						TKMnRO_K2		
	7	6	5	4	3	2	1	0	
Type/Reset	TKMnRO_K2								
	RW	0	RW	0	RW	0	RW	0	
	RW	0	RW	0	RW	0	RW	0	

Bits	Field	Descriptions
[9:0]	TKMnRO_K2	Touch Key Module n Key 2 Reference Oscillator Capacitor Value

## Touch Key Module n Key 1 Capacitor Register – TKMnK1CPR

This register specifies the module n Key 1 reference oscillator capacitor value.

Offset:  $0x100 \times (n + 1) + 0x018$

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						TKMnRO_K1		
	7	6	5	4	3	2	1	0	
Type/Reset	TKMnRO_K1								
	RW	0	RW	0	RW	0	RW	0	
	RW	0	RW	0	RW	0	RW	0	

Bits	Field	Descriptions
[9:0]	TKMnRO_K1	Touch Key Module n Key 1 Reference Oscillator Capacitor Value

## Touch Key Module n Key 0 Capacitor Register – TKMnK0CPR

This register specifies the module n Key 0 reference oscillator capacitor value.

Offset:  $0x100 \times (n + 1) + 0x01C$

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						TKMnRO_K0		
Type/Reset							RW	0	RW
	7	6	5	4	3	2	1	0	
	TKMnRO_K0								
Type/Reset	RW	RW	RW	RW	0	RW	0	RW	0
	0	0	0						

Bits	Field	Descriptions
[9:0]	TKMnRO_K0	Touch Key Module n Key 0 Reference Oscillator Capacitor Value

## Touch Key Module n C/F Counter Register – TKMnCFCNTR

This register contains the touch key module n 16-bit C/F counter value.

Offset:  $0x100 \times (n + 1) + 0x020$

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	TKMn16D								
	RO	0	RO	0	RO	0	RO	0	RO
	7	6	5	4	3	2	1	0	
Type/Reset	TKMn16D								
	RO	0	RO	0	RO	0	RO	0	RO

Bits	Field	Descriptions
[15:0]	TKMn16D	<p>Touch Key Module n 16-bit C/F Counter Value</p> <p>This 16-bit C/F counter will be stopped and the counter content will be kept unchanged when the touch key time slot counter overflows in the manual scan mode. However, this 16-bit C/F counter content will be cleared to zero at the end of the time slot 0, slot 1 and slot 2 after it is written to the corresponding register but kept unchanged at the end of the time slot 3 when the auto scan mode or the periodic auto scan mode is selected.</p>

## Touch Key Module n Key 3 Counter Register – TKMnK3CNTR

This register contains the touch key module n Key 3 16-bit C/F counter value.

Offset:  $0x100 \times (n + 1) + 0x024$

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	TKMn16D_K3								
	RO	0	RO	0	RO	0	RO	0	RO
	7	6	5	4	3	2	1	0	
Type/Reset	TKMn16D_K3								
	RO	0	RO	0	RO	0	RO	0	RO

Bits	Field	Descriptions
[15:0]	TKMn16D_K3	Touch Key Module n Key 3 16-bit C/F Counter Value The TKMn16D_K3 bit field will be cleared when the corresponding module n Key 3 is disabled.

## Touch Key Module n Key 2 Counter Register – TKMnK2CNTR

This register contains the touch key module n Key 2 16-bit C/F counter value.

Offset:  $0x100 \times (n + 1) + 0x028$

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	TKMn16D_K2								
	RO	0	RO	0	RO	0	RO	0	RO
	7	6	5	4	3	2	1	0	
Type/Reset	TKMn16D_K2								
	RO	0	RO	0	RO	0	RO	0	RO

Bits	Field	Descriptions
[15:0]	TKMn16D_K2	Touch Key Module n Key 2 16-bit C/F Counter Value The TKMn16D_K2 bit field will be cleared when the corresponding module n Key 2 is disabled.



## Touch Key Module n Key 1 Counter Register – TKMnK1CNTR

This register contains the touch key module n Key 1 16-bit C/F counter value.

Offset:  $0x100 \times (n + 1) + 0x02C$

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	TKMn16D_K1								
	RO	0	RO	0	RO	0	RO	0	RO
	7	6	5	4	3	2	1	0	
Type/Reset	TKMn16D_K1								
	RO	0	RO	0	RO	0	RO	0	RO

Bits	Field	Descriptions
[15:0]	TKMn16D_K1	Touch Key Module n Key 1 16-bit C/F Counter Value The TKMn16D_K1 bit field will be cleared when the corresponding module n Key 1 is disabled.

## Touch Key Module n Key 0 Counter Register – TKMnK0CNTR

This register contains the touch key module n Key 0 16-bit C/F counter value.

Offset:  $0x100 \times (n + 1) + 0x030$

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	TKMn16D_K0								
	RO	0	RO	0	RO	0	RO	0	RO
	7	6	5	4	3	2	1	0	
Type/Reset	TKMn16D_K0								
	RO	0	RO	0	RO	0	RO	0	RO

Bits	Field	Descriptions
[15:0]	TKMn16D_K0	Touch Key Module n Key 0 16-bit C/F Counter Value The TKMn16D_K0 bit field will be cleared when the corresponding module n Key 0 is disabled.

## Touch Key Module n Key 3 Threshold Register – TKMnK3THR

This register specifies the touch key module n Key 3 threshold value.

Offset:  $0x100 \times (n + 1) + 0x034$

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	TK3MnTH16								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	TK3MnTH16								
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:0]	TK3MnTH16	Touch Key Module n Key 3 Threshold Value

## Touch Key Module n Key 2 Threshold Register – TKMnK2THR

This register specifies the touch key module n Key 2 threshold value.

Offset:  $0x100 \times (n + 1) + 0x038$

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	TK2MnTH16								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	TK2MnTH16								
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:0]	TK2MnTH16	Touch Key Module n Key 2 Threshold Value

## Touch Key Module n Key 1 Threshold Register – TKMnK1THR

This register specifies the touch key module n Key 1 threshold value.

Offset:  $0x100 \times (n + 1) + 0x03C$

Reset value: 0x0000\_0000

	31	30	29	28	27	26	25	24	
Type/Reset	Reserved								
	23	22	21	20	19	18	17	16	
Type/Reset	Reserved								
	15	14	13	12	11	10	9	8	
Type/Reset	TK1MnTH16								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	TK1MnTH16								
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:0]	TK1MnTH16	Touch Key Module n Key 1 Threshold Value

## Touch Key Module n Key 0 Threshold Register – TKMnK0THR

This register specifies the touch key module n Key 0 threshold value.

Offset:  $0x100 \times (n + 1) + 0x040$

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	TK0MnTH16								
	RW	0	RW	0	RW	0	RW	0	RW
	7	6	5	4	3	2	1	0	
Type/Reset	TK0MnTH16								
	RW	0	RW	0	RW	0	RW	0	RW

Bits	Field	Descriptions
[15:0]	TK0MnTH16	Touch Key Module n Key 0 Threshold Value

Copyright© 2023 by HOLTEK SEMICONDUCTOR INC. All Rights Reserved.

The information provided in this document has been produced with reasonable care and attention before publication, however, HOLTEK does not guarantee that the information is completely accurate. The information contained in this publication is provided for reference only and may be superseded by updates. HOLTEK disclaims any expressed, implied or statutory warranties, including but not limited to suitability for commercialization, satisfactory quality, specifications, characteristics, functions, fitness for a particular purpose, and non-infringement of any third-party's rights. HOLTEK disclaims all liability arising from the information and its application. In addition, HOLTEK does not recommend the use of HOLTEK's products where there is a risk of personal hazard due to malfunction or other reasons. HOLTEK hereby declares that it does not authorise the use of these products in life-saving, life-sustaining or safety critical components. Any use of HOLTEK's products in life-saving/sustaining or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold HOLTEK harmless from any damages, claims, suits, or expenses resulting from such use. The information provided in this document, including but not limited to the content, data, examples, materials, graphs, and trademarks, is the intellectual property of HOLTEK (and its licensors, where applicable) and is protected by copyright law and other intellectual property laws. No license, express or implied, to any intellectual property right, is granted by HOLTEK herein. HOLTEK reserves the right to revise the information described in the document at any time without prior notice. For the latest information, please contact us.