

NuMicro[®] Family

Arm[®] Cortex[®]-M23-based Microcontroller

M253 Series

Technical Reference Manual

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro[®] microcontroller and microprocessor based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

TABLE OF CONTENTS

1	GENERAL DESCRIPTION.....	19
2	FEATURES	20
3	PARTS INFORMATION	31
3.1	Package Type	31
3.2	M253 Selection Guide	32
3.3	M253 Selection Code.....	33
4	PIN CONFIGURATION	34
4.1	Pin Configuration	34
4.1.1	M253 Pin Diagram.....	34
4.1.2	M253 Multi-function Pin Diagram	36
4.2	M253 Pin Mapping	41
4.3	M253 Pin Functional Description	43
5	BLOCK DIAGRAM.....	49
6	FUNCTIONAL DESCRIPTION.....	50
6.1	Arm® Cortex®-M23 Core.....	50
6.2	System Manager.....	52
6.2.1	Overview	52
6.2.2	System Reset.....	52
6.2.3	System Power Distribution	58
6.2.4	Power Modes and Wake-up Sources	59
6.2.5	Chip Bus Matrix	62
6.2.6	System Memory Map	63
6.2.7	SRAM Memory Organization	65
6.2.8	IRC Auto Trim.....	66
6.2.9	Register Map	67
6.2.10	Register Description.....	69
6.2.11	System Timer (SysTick).....	111
6.2.12	Nested Vectored Interrupt Controller (NVIC).....	116
6.2.13	System Control Register	137
6.3	Clock Controller	146
6.3.1	Overview	146
6.3.2	Clock Generator.....	147
6.3.3	System Clock and SysTick Clock.....	148
6.3.4	Peripherals Clock	149
6.3.5	Power-down Mode Clock	149
6.3.6	Clock Output.....	150
6.3.7	USB Clock Source.....	151

6.3.8	Register Map	152
6.3.9	Register Description	153
6.4	Flash Memory Controller (FMC)	177
6.4.1	Overview	177
6.4.2	Features	177
6.4.3	Block Diagram	177
6.4.4	Functional Description	179
6.4.5	Register Map	204
6.4.6	Register Description	205
6.5	General Purpose I/O (GPIO)	225
6.5.1	Overview	225
6.5.2	Features	225
6.5.3	Block Diagram	226
6.5.4	Basic Configuration	226
6.5.5	Functional Description	226
6.5.6	Register Map	231
6.5.7	Register Description	233
6.6	PDMA Controller (PDMA)	247
6.6.1	Overview	247
6.6.2	Features	247
6.6.3	Block Diagram	247
6.6.4	Basic Configuration	247
6.6.5	Functional Description	248
6.6.6	Register Map	254
6.6.7	Register Description	255
6.7	Timer Controller (TMR)	284
6.7.1	Overview	284
6.7.2	Features	284
6.7.4	Block Diagram	285
6.7.5	Basic Configuration	287
6.7.6	Timer Functional Description	287
6.7.7	Register Map	294
6.7.8	Register Description	296
6.8	Watchdog Timer (WDT)	309
6.8.1	Overview	309
6.8.2	Features	309
6.8.3	Block Diagram	309
6.8.4	Basic Configuration	309
6.8.5	Functional Description	310
6.8.6	Register Map	312
6.8.7	Register Description	313

6.9 Window Watchdog Timer (WWDT).....	317
6.9.1 Overview	317
6.9.2 Features	317
6.9.3 Block Diagram.....	317
6.9.4 Basic Configuration	317
6.9.5 Functional Description	318
6.9.6 Register Map.....	322
6.9.7 Register Description.....	323
6.10 Real Time Clock (RTC).....	328
6.10.1 Overview	328
6.10.2 Features	328
6.10.3 Block Diagram.....	329
6.10.4 Basic Configuration	329
6.10.5 Functional Description	329
6.10.6 Register Map.....	334
6.10.7 Register Description.....	335
6.11 Basic PWM Generator and Capture Timer (BPWM).....	352
6.11.1 Overview	352
6.11.2 Features	352
6.11.3 Block Diagram.....	353
6.11.4 Basic Configuration	355
6.11.5 Functional Description	356
6.11.6 Register Map.....	371
6.11.7 Register Description.....	374
6.12 UART Interface Controller (UART).....	405
6.12.1 Overview	405
6.12.2 Features	405
6.12.3 Block Diagram.....	407
6.12.4 Basic Configuration	409
6.12.5 Functional Description	412
6.12.6 Register Map.....	429
6.12.7 Register Description.....	431
6.13 Serial Peripheral Interface (SPI).....	459
6.13.1 Overview	459
6.13.2 Features	459
6.13.3 Block Diagram.....	460
6.13.4 Basic Configuration	460
6.13.5 Functional Description	461
6.13.6 Timing Diagram.....	477
6.13.7 Programming Examples	479
6.13.8 Register Map.....	482

6.13.9	Register Description.....	483
6.14	I ² C Serial Interface Controller (I ² C).....	504
6.14.1	Overview.....	504
6.14.2	Features.....	504
6.14.3	Block Diagram.....	505
6.14.4	Basic Configuration.....	505
6.14.5	Register Map.....	522
6.14.6	Register Description.....	523
6.15	USCI - Universal Serial Control Interface Controller (USCI).....	535
6.15.1	Overview.....	535
6.15.2	Features.....	535
6.15.3	Block Diagram.....	535
6.15.4	Functional Description.....	535
6.16	USCI – UART Mode.....	547
6.16.1	Overview.....	547
6.16.2	Features.....	547
6.16.3	Block Diagram.....	547
6.16.4	Basic Configuration.....	547
6.16.5	Functional Description.....	548
6.16.6	Register Map.....	557
6.16.7	Register Description.....	558
6.17	USCI - SPI Mode.....	580
6.17.1	Overview.....	580
6.17.2	Features.....	580
6.17.3	Block Diagram.....	581
6.17.4	Basic Configuration.....	581
6.17.5	Functional Description.....	582
6.17.6	Register Map.....	594
6.17.7	Register Description.....	595
6.18	USCI - I ² C Mode.....	617
6.18.1	Overview.....	617
6.18.2	Features.....	617
6.18.3	Block Diagram.....	618
6.18.4	Basic Configuration.....	618
6.18.5	Functional Description.....	619
6.18.6	Register Map.....	638
6.18.7	Register Description.....	639
6.19	USB 2.0 Full-Speed Device Controller (USBBD).....	658
6.19.1	Overview.....	658
6.19.2	Features.....	658
6.19.3	Block Diagram.....	659

6.19.4	Basic Configuration	659
6.19.5	Functional Description	659
6.19.6	Register Map	665
6.19.7	Register Description	668
6.20	Controller Area Network with Feasibility Data Rate (CAN FD)	695
6.20.1	Overview	695
6.20.2	Features	695
6.20.3	Block Diagram	695
6.20.4	Basic Configuration	697
6.20.5	Functional Description	698
6.20.6	Register Map	726
6.20.7	Register Description	728
6.21	Enhanced 12-bit Analog-to-Digital Converter (EADC)	781
6.21.1	Overview	781
6.21.2	Features	781
6.21.3	Block Diagram	782
6.21.4	Basic Configuration	782
6.21.5	Functional Description	783
6.21.6	Register Map	796
6.21.7	Register Description	798
6.22	CRC Controller (CRC)	824
6.22.1	Overview	824
6.22.2	Features	824
6.22.3	Block Diagram	824
6.22.4	Basic Configuration	824
6.22.5	Functional Description	825
6.22.6	Register Map	827
6.22.7	Register Description	828
6.23	Peripherals Interconnection	833
6.23.1	Overview	833
6.23.2	Peripherals Interconnect Matrix table	833
6.23.3	Functional Description	833
7	APPLICATION CIRCUIT	835
7.1	Power Supply Scheme	835
7.2	Peripheral Application Scheme	836
8	ELECTRICAL CHARACTERISTICS	837
9	PACKAGE DIMENSIONS	838
9.1	QFN 33L (5x5x0.8 mm)	838
9.2	LQFP 48L (7x7x1.4 mm Footprint 2.0 mm)	840
10	ABBREVIATIONS	841

10.1 Abbreviations..... 841

11 REVISION HISTORY 843

LIST OF FIGURES

Figure 4.1-1 M253 QFN 33-pin Diagram	34
Figure 4.1-2 M253 LQFP 48-pin Diagram.....	35
Figure 4.1-3 M253ZE3AE Multi-function Pin Diagram	36
Figure 4.1-4 M253LD3AE/M253LE3AE Multi-function Pin Diagram.....	38
Figure 4.3-1 M253 Block Diagram	49
Figure 6.1-1 Cortex®-M23 Block Diagram.....	50
Figure 6.2-1 System Reset Sources	53
Figure 6.2-2 nRESET Reset Waveform	55
Figure 6.2-3 Power-on Reset (POR) Waveform	55
Figure 6.2-4 Low Voltage Reset (LVR) Waveform.....	56
Figure 6.2-5 Brown-out Detector (BOD) Waveform	57
Figure 6.2-6 NuMicro® M253 Series Power Distribution Diagram	58
Figure 6.2-7 Power Mode State Machine	60
Figure 6.2-8 SRAM Memory Organization	65
Figure 6.3-1 Clock Generator Global View Diagram.....	146
Figure 6.3-2 Clock Generator Block Diagram	147
Figure 6.3-3 System Clock Block Diagram	148
Figure 6.3-4 HXT Stop Protect Procedure	149
Figure 6.3-5 SysTick Clock Control Block Diagram	149
Figure 6.3-6 Clock Output Block Diagram	150
Figure 6.3-7 USB Clock Source	151
Figure 6.4-1 Flash Memory Controller Block Diagram.....	178
Figure 6.4-2 Flash Memory Map	189
Figure 6.4-3 System Memory Map with IAP Mode	190
Figure 6.4-4 APROM/LDROM Boot with IAP Mode.....	191
Figure 6.4-5 VECMAP Setting Example in IAP Mode.....	191
Figure 6.4-6 System Memory Map without IAP mode	192
Figure 6.4-7 ISP Procedure Example	195
Figure 6.4-8 Example for Accelerating Interrupt by VECMAP	196
Figure 6.4-9 ISP 32-bit Programming Procedure.....	197
Figure 6.4-10 Firmware in SRAM for Multi-word Programming.....	198
Figure 6.4-11 Multi-word Programming Flow.....	199
Figure 6.4-12 Fast Flash Programming Verification Flow.....	200
Figure 6.4-13 Verification Flow	200
Figure 6.4-14 Flash CRC32 Checksum Calculation	201
Figure 6.4-15 CRC-32 Checksum Calculation Flow	202

Figure 6.4-16 All-One Verification Flow	203
Figure 6.5-1 GPIO Controller Block Diagram.....	226
Figure 6.5-2 Input Mode.....	227
Figure 6.5-3 Push-Pull Output.....	227
Figure 6.5-4 Open-Drain Output	228
Figure 6.5-5 Quasi-Bidirectional I/O Mode.....	228
Figure 6.5-6 GPIO Rising Edge Trigger Interrupt	229
Figure 6.5-7 GPIO Falling Edge Trigger Interrupt.....	230
Figure 6.6-1 PDMA Controller Block Diagram	247
Figure 6.6-2 Descriptor Table Entry Structure	248
Figure 6.6-3 Basic Mode Finite State Machine	249
Figure 6.6-4 Descriptor Table Link List Structure	250
Figure 6.6-5 Scatter-gather Mode Finite State Machine	250
Figure 6.6-6 Example of Single Transfer Type and Burst Transfer Type in Basic Mode	252
Figure 6.6-7 Example of PDMA Channel 0 Time-out Counter Operation.....	253
Figure 6.7-1 Timer Controller Block Diagram	285
Figure 6.7-2 Clock Source of Timer Controller	286
Figure 6.7-3 Continuous Counting Mode	289
Figure 6.7-4 External Capture Mode.....	290
Figure 6.7-5 Reset Counter Mode.....	290
Figure 6.7-6 Internal Timer Trigger	291
Figure 6.7-7 Capture interrupt trigger PDMA	292
Figure 6.7-8 Inter-Timer Trigger Capture Timing	293
Figure 6.8-1 Watchdog Timer Block Diagram.....	309
Figure 6.8-2 Watchdog Timer Clock Control.....	310
Figure 6.8-3 Watchdog Timer Time-out Interval and Reset Period Timing	311
Figure 6.9-1 WWDT Block Diagram.....	317
Figure 6.9-2 WWDT Clock Control.....	317
Figure 6.9-3 WWDT Reset and Reload Behavior	319
Figure 6.9-4 WWDT Reload Counter When CNTDAT > CMPDAT	319
Figure 6.9-5 WWDT Reload Counter When WWDT_CNT < WINCMP	320
Figure 6.9-6 WWDT Interrupt and Reset Signals	320
Figure 6.10-1 RTC Block Diagram	329
Figure 6.11-1 BPWM Generator Overview Block Diagram.....	353
Figure 6.11-2 BPWM Clock Source Control	353
Figure 6.11-3 BPWM Clock Source Control	354
Figure 6.11-4 BPWM Independent Mode Architecture Diagram	355

Figure 6.11-5 BPWM_CH0 CLKPSC Waveform	356
Figure 6.11-6 BPWM Counter Clear Waveform.....	357
Figure 6.11-7 BPWM Up Counter Type	357
Figure 6.11-8 BPWM Down Counter Type	358
Figure 6.11-9 BPWM Up-Down Counter Type.....	358
Figure 6.11-10 BPWM CMPDAT Events in Up-Down Counter Type	359
Figure 6.11-11 Period Loading Mode with Up-Counter Type	360
Figure 6.11-12 Immediately Loading Mode with Up-Counter Type	361
Figure 6.11-13 Center Loading Mode with Up-Down-Counter Type	362
Figure 6.11-14 BPWM Pulse Generation (Left: Asymmetric Pulse, Right: Variety Pulse)	363
Figure 6.11-15 BPWM 0% to 100% Pulse Generation (Left: Up Counter Type, Right: Up-down Counter Type)	363
Figure 6.11-16 BPWM_CH0 Output Control 3 Steps.....	364
Figure 6.11-17 Mask Control Waveform Illustration.....	365
Figure 6.11-18 Initial State and Polarity Control	365
Figure 6.11-19 BPWM_CH0 and BPWM_CH1 Pair Interrupt Architecture Diagram.....	366
Figure 6.11-20 BPWM_CH0 and BPWM_CH1 Pair Trigger EADC Source Block Diagram	367
Figure 6.11-21 BPWM CH0~ CH5 Trigger EADC Block Diagram	367
Figure 6.11-22 BPWM Trigger EADC in Up-Down Counter Type Timing Waveform	368
Figure 6.11-23 BPWM_CH0 Capture Block Diagram	369
Figure 6.11-24 Capture Operation Waveform.....	370
Figure 6.12-1 UART Clock Control Diagram.....	407
Figure 6.12-2 UART Block Diagram.....	408
Figure 6.12-3 Auto-Baud Rate Measurement.....	415
Figure 6.12-4 Transmit Delay Time Operation.....	416
Figure 6.12-5 UART nCTS Wake-up Case1.....	416
Figure 6.12-6 UART nCTS Wake-up Case2.....	417
Figure 6.12-7 UART Data Wake-up.....	417
Figure 6.12-8 UART Received Data FIFO reached threshold wake-up	418
Figure 6.12-9 UART RS-485 AAD Mode Address Match Wake-up.....	418
Figure 6.12-10 UART Received Data FIFO threshold time-out wake-up	419
Figure 6.12-11 Auto-Flow Control Block Diagram	422
Figure 6.12-12 UART nCTS Auto-Flow Control Enabled	422
Figure 6.12-13 UART nRTS Auto-Flow Control Enabled	423
Figure 6.12-14 UART nRTS Auto-Flow with Software Control	423
Figure 6.12-15 IrDA Control Block Diagram	424
Figure 6.12-16 IrDA TX/RX Timing Diagram	425
Figure 6.12-17 RS-485 nRTS Driving Level in Auto Direction Mode	426

Figure 6.12-18 RS-485 nRTS Driving Level with Software Control	427
Figure 6.12-19 Structure of RS-485 Frame	427
Figure 6.13-1 SPI Block Diagram.....	460
Figure 6.13-2 SPI Peripheral Clock.....	462
Figure 6.13-3 SPI Full-Duplex Master Mode Application Block Diagram	462
Figure 6.13-4 SPI Full-Duplex Slave Mode Application Block Diagram	463
Figure 6.13-5 32-bit in One Transaction	464
Figure 6.13-6 Automatic Slave Selection (SSACTPOL = 0, SUSPITV > 0x2)	464
Figure 6.13-7 Automatic Slave Selection (SSACTPOL = 0, SUSPITV < 0x3)	465
Figure 6.13-8 Byte Reorder Function.....	465
Figure 6.13-9 Timing Waveform for Byte Suspend	466
Figure 6.13-10 SPI Half-Duplex Master Mode Application Block Diagram.....	466
Figure 6.13-11 SPI Half-Duplex Slave Mode Application Block Diagram.....	466
Figure 6.13-12 FIFO Threshold Comparator	468
Figure 6.13-13 Transmit FIFO Buffer Example for 8~16 Bits of Data Length.....	469
Figure 6.13-14 Transmit FIFO Buffer Example for 17~32 Bits of Data Length.....	469
Figure 6.13-15 Receive FIFO Buffer Example for 16 Bits of Data Length.....	470
Figure 6.13-16 Receive FIFO Buffer Example for 32 Bits of Data Length.....	471
Figure 6.13-17 TX Underflow Event and Slave Under Run Event.....	471
Figure 6.13-18 Slave Mode Bit Count Error	472
Figure 6.13-19 I ² S Data Format Timing Diagram.....	474
Figure 6.13-20 MSB Justified Data Format Timing Diagram	474
Figure 6.13-21 PCM Mode A Timing Diagram	474
Figure 6.13-22 PCM Mode B Timing Diagram	474
Figure 6.13-23 FIFO Contents for Various I ² S Modes	475
Figure 6.13-24 SPI Timing in Master Mode	478
Figure 6.13-25 SPI Timing in Master Mode (Alternate Phase of SPIx_CLK)	478
Figure 6.13-26 SPI Timing in Slave Mode	479
Figure 6.13-27 SPI Timing in Slave Mode (Alternate Phase of SPIx_CLK)	479
Figure 6.14-1 I ² C Controller Block Diagram.....	505
Figure 6.14-2 I ² C Bus Timing	506
Figure 6.14-3 I ² C Protocol.....	506
Figure 6.14-4 START and STOP Conditions	507
Figure 6.14-5 Bit Transfer on the I ² C Bus	507
Figure 6.14-6 Acknowledge on the I ² C Bus	508
Figure 6.14-7 Master Transmits Data to Slave by 7-bit	508
Figure 6.14-8 Master Reads Data from Slave by 7-bit.....	508

Figure 6.14-9 Master Transmits Data to Slave by 10-bit	509
Figure 6.14-10 Master Reads Data from Slave by 10-bit	509
Figure 6.14-11 Control I ² C Bus according to the Current I ² C Status	510
Figure 6.14-12 Master Transmitter Mode Control Flow	510
Figure 6.14-13 Master Receiver Mode Control Flow	511
Figure 6.14-14 Slave Mode Control Flow	512
Figure 6.14-15 GC Mode	513
Figure 6.14-16 Arbitration Lost.....	514
Figure 6.14-17 Setup Time Wrong Adjustment.....	516
Figure 6.14-18 Hold Time Wrong Adjustment.....	516
Figure 6.14-19 I ² C Data Shifting Direction	517
Figure 6.14-20 I ² C Time-out Count Block Diagram	519
Figure 6.14-21 I ² C Wake-Up Related Signals Waveform	520
Figure 6.14-22 EEPROM Random Read	521
Figure 6.14-23 Protocol of EEPROM Random Read.....	521
Figure 6.15-1 USCI Block Diagram.....	535
Figure 6.15-2 Input Conditioning for USC1x_DAT[1:0] and USC1x_CTL[1:0]	536
Figure 6.15-3 Input Conditioning for USC1x_CLK	537
Figure 6.15-4 Block Diagram of Data Buffering	538
Figure 6.15-5 Data Access Structure	539
Figure 6.15-6 Transmit Data Path.....	539
Figure 6.15-7 Receive Data Path.....	541
Figure 6.15-8 Protocol-Relative Clock Generator	542
Figure 6.15-9 Basic Clock Divider Counter	543
Figure 6.15-10 Block of Timing Measurement Counter	543
Figure 6.15-11 Sample Time Counter	544
Figure 6.15-12 Event and Interrupt Structure	545
Figure 6.16-1 USCI-UART Mode Block Diagram.....	547
Figure 6.16-2 UART Signal Connection for Full-Duplex Communication	549
Figure 6.16-3 UART Standard Frame Format	550
Figure 6.16-4 UART Bit Timing (Data Sample Time).....	551
Figure 6.16-5 UART Auto Baud Rate Control.....	553
Figure 6.16-6 Incoming Data Wake-Up	554
Figure 6.16-7 nCTS Wake-Up Case 1	554
Figure 6.16-8 nCTS Wake-Up Case 2	555
Figure 6.17-1 SPI Master Mode Application Block Diagram.....	580
Figure 6.17-2 SPI Slave Mode Application Block Diagram.....	580

Figure 6.17-3 USCI SPI Mode Block Diagram	581
Figure 6.17-4 4-Wire Full-Duplex SPI Communication Signals (Master Mode)	582
Figure 6.17-5 4-Wire Full-Duplex SPI Communication Signals (Slave Mode)	583
Figure 6.17-6 SPI Communication with Different SPI Clock Configuration (SCLKMODE=0x0)..	584
Figure 6.17-7 SPI Communication with Different SPI Clock Configuration (SCLKMODE=0x1)..	584
Figure 6.17-8 SPI Communication with Different SPI Clock Configuration (SCLKMODE=0x2)..	584
Figure 6.17-9 SPI Communication with Different SPI Clock Configuration (SCLKMODE=0x3)..	585
Figure 6.17-10 16-bit Data Length in One Word Transaction with MSB First Format	586
Figure 6.17-11 Word Suspend Interval between Two Transaction Words	586
Figure 6.17-12 Auto Slave Select (SUSPITV $\geq$ 0x3)	587
Figure 6.17-13 Auto Slave Select (SUSPITV < 0x3)	587
Figure 6.17-14 One Output Data Channel Half-duplex (SPI Master Mode)	588
Figure 6.17-15 One Input Data Channel Half-duplex (SPI Master Mode)	588
Figure 6.17-16 SPI Timing in Master Mode	590
Figure 6.17-17 SPI Timing in Master Mode (Alternate Phase of Serial Bus Clock)	591
Figure 6.17-18 SPI Timing in Slave Mode	591
Figure 6.17-19 SPI Timing in Slave Mode (Alternate Phase of Serial Bus Clock)	592
Figure 6.18-1 I ² C Bus Timing	617
Figure 6.18-2 USCI I ² C Mode Block Diagram	618
Figure 6.18-3 I ² C Protocol	620
Figure 6.18-4 START and STOP Conditions	620
Figure 6.18-5 Bit Transfer on the I ² C Bus	621
Figure 6.18-6 Acknowledge on the I ² C Bus	622
Figure 6.18-7 Arbitration Lost	623
Figure 6.18-8 Control I ² C Bus according to Current I ² C Status	625
Figure 6.18-9 Master Transmits Data to Slave with a 7-bit Address	626
Figure 6.18-10 Master Reads Data from Slave with a 7-bit Address	626
Figure 6.18-11 Master Transmits Data to Slave by 10-bit Address	626
Figure 6.18-12 Master Reads Data from Slave by 10-bit Address	626
Figure 6.18-13 Master Transmitter Mode Control Flow with 7-bit Address	627
Figure 6.18-14 Master Receiver Mode Control Flow with 7-bit Address	628
Figure 6.18-15 Master Transmitter Mode Control Flow with 10-bit Address	629
Figure 6.18-16 Master Receiver Mode Control Flow with 10-bit Address	630
Figure 6.18-17 Slave Mode Control Flow with 7-bit Address	631
Figure 6.18-18 Slave Mode Control Flow with 10-bit Address	632
Figure 6.18-19 GC Mode with 7-bit Address	633
Figure 6.18-20 Setup Time Wrong Adjustment	635

Figure 6.18-21 Hold Time Wrong Adjustment.....	635
Figure 6.18-22 I ² C Time-out Count Block Diagram	636
Figure 6.18-23 EEPROM Random Read	637
Figure 6.18-24 Protocol of EEPROM Random Read.....	637
Figure 6.19.1 USB Block Diagram	659
Figure 6.19.2 NEVWK Interrupt Operation Flow.....	661
Figure 6.19.3 Endpoint SRAM Structure	662
Figure 6.19.4 Setup Transaction Followed by Data IN Transaction	662
Figure 6.19.5 Data Out Transfer	663
Figure 6.19.6 LPM State Transition Diagram.....	664
Figure 6.20-1 CAN FD Controller Block Diagram	697
Figure 6.20-2 Transceiver Delay Measurement.....	701
Figure 6.20-3 Pin Controller in Bus Monitor Mode	702
Figure 6.20-4 Pin Control in Loop Back Modes	704
Figure 6.20-5 Standard Message ID Filter Path	707
Figure 6.20-6 Extended Message ID Filter Path.....	708
Figure 6.20-7 Rx FIFO Status	709
Figure 6.20-8 Rx FIFO Overflow Handling.....	710
Figure 6.20-9 Example of Mixed Configuration Dedicated Tx Buffers / Tx FIFO	714
Figure 6.20-10 Example of mixed Configuration Dedicated Tx Buffers / Tx Queue	714
Figure 6.20-11 Message RAM Configuration.....	717
Figure 6.20-12 Bit Timing	724
Figure 6.21-1 EADC Converter Block Diagram	782
Figure 6.21-2 Sample Module 0~3 Block Diagram	784
Figure 6.21-3 Sample Module 16~17 Block Diagram	784
Figure 6.21-4 EADC Clock Control	785
Figure 6.21-5 Example EADC Conversion Timing Diagram, n=0~3.....	786
Figure 6.21-6 Sample Module Conversion Priority Arbitrator Diagram.....	787
Figure 6.21-7 Specific Sample Module ADC EOC Signal for ADINT0~3 Interrupt.....	788
Figure 6.21-8 EADC0_ST De-bounce Timing Diagram	789
Figure 6.21-9 BPWM-triggered EADC Start Conversion	789
Figure 6.21-10 External triggered EADC Start Conversion	790
Figure 6.21-11 Conversion Start Delay Timing Diagram	791
Figure 6.21-12 ADC Extend Sampling Timing Diagram	792
Figure 6.21-13 EADC_DATn (n= 0~15) Aligned Format	792
Figure 6.21-14 ADC Conversion Result Monitor Logics Diagram	794
Figure 6.21-15 ADC Controller Interrupts	795

Figure 6.22-1 CRC Generator Block Diagram 824

Figure 6.22-2 CHECKSUM Bit Order Reverse Functional Block..... 825

Figure 6.22-3 Write Data Bit Order Reverse Functional Block 826

List of Tables

Table 1-1 NuMicro® M253 Series Key Features Support Table	19
Table 4.1-1 M253ZE3AE Multi-function Pin Table	37
Table 4.1-2 M253LD3AE/M253LE3AE Multi-function Pin Table	40
Table 6.2-1 Reset Value of Registers	54
Table 6.2-2 Power Mode Table	59
Table 6.2-3 Power Mode Difference Table	59
Table 6.2-4 Power Mode Difference Table	59
Table 6.2-5 Clocks in Power Modes	61
Table 6.2-6 Condition of Entering Power-down Mode Again	62
Table 6.2-7 Address Space Assignments for On-Chip Controllers.....	64
Table 6.2-8 Exception Model	117
Table 6.2-9 Interrupt Number Table.....	119
Table 6.4-1 Boot Source Selection Table	192
Table 6.4-2 ISP Command List.....	194
Table 6.4-3 FMC Control Registers for Flash Programming	196
Table 6.6-1 Channel Priority Table	248
Table 6.8-1 Watchdog Timer Time-out Interval Period Selection	311
Table 6.9-1 WWDT Prescaler Value Selection	318
Table 6.9-2 CMPDAT Setting Limitation	321
Table 6.10-1 RTC Read/Write Enable	330
Table 6.10-212/24 Hour Time Scale Selection	331
Table 6.10-3 Registers Value after Powered On	333
Table 6.11-1 BPWM Clock Source Control Registers Setting Table	354
Table 6.11-2 BPWM Pulse Generation Event Priority for Up-Counter	363
Table 6.11-3 BPWM Pulse Generation Event Priority for Down-Counter	364
Table 6.11-4 BPWM Pulse Generation Event Priority for Up-Down-Counter	364
Table 6.12-1 NuMicro® M253 Series UART Features	406
Table 6.12-2 UART Interrupt.....	409
Table 6.12-3 UART Interface Controller Pin	412
Table 6.12-4 UART controller Baud Rate Equation Table	412
Table 6.12-5 UART controller Baud Rate Parameter Setting Example Table	413
Table 6.12-6 UART controller Baud Rate Register Setting Example Table	413
Table 6.12-7 Baud Rate Compensation Example Table 1	414
Table 6.12-8 Baud Rate Compensation Example Table 2	414
Table 6.12-9 UART controller Interrupt Source and Flag List.....	420
Table 6.12-10 UART Line Control of Word and Stop Length Setting	421

Table 6.12-11 UART Line Control of PARITY Bit Setting	421
Table 6.13-1 SPI/I ² S Interface Controller Pin Description (SPI0)	461
Table 6.13-2 Dummy Data Number for I ² S / PCM Master Mode and Monaural Mode	476
Table 6.13-3 Dummy Data Number for I ² S / PCM Master Mode and Stereo Mode	476
Table 6.13-4 Dummy Data Number for I ² S Slave Mode and Monaural Mode	476
Table 6.13-5 Dummy Data Number for PCM Slave Mode and Monaural Mode	477
Table 6.13-6 Dummy Data Number for I ² S Slave Mode and Stereo Mode	477
Table 6.13-7 Dummy Data Number for PCM Slave Mode and Stereo Mode	477
Table 6.14-1 Relationship between I ² C Baud Rate and PCLK	515
Table 6.14-2 I ² C Status Code Description	518
Table 6.15-1 Input Signals for Different Protocols	536
Table 6.15-2 Output Signals for Different Protocols	537
Table 6.15-3 Data Transfer Events and Interrupt Handling	545
Table 6.15-4 Protocol-specific Events and Interrupt Handling	546
Table 6.16-1 Input Signals for UART Protocol	549
Table 6.16-2 Output Signals for UART Protocol	549
Table 6.16-3 Baud Rate Relationship	552
Table 6.17-1 Signals for SPI communication	582
Table 6.17-2 Serial Bus Clock Configuration	583
Table 6.18-1 Relationship between I ² C Baud Rate and PCLK	635
Table 6.19-1 USB Link Power Manager (Lx) States	663
Table 6.20-1 DLC Coding in CAN FD Format	700
Table 6.20-2 Rx Buffer / FIFO Element Size	710
Table 6.20-3 Example Filter Configuration for Rx Buffers	711
Table 6.20-4 Possible Configuration for Frame Transmission	712
Table 6.20-5 Tx Buffer / FIFO / Queue Element Size	713
Table 6.20-6 Rx Buffer and FIFO Element	717
Table 6.20-7 Rx Buffer and FIFO Element Description	718
Table 6.20-8 Tx Buffer and FIFO Element	719
Table 6.20-9 Tx Buffer Element Description	720
Table 6.20-10 Tx Event FIFO Element	721
Table 6.20-11 Tx Event FIFO Element Description	721
Table 6.20-12 Standard Message ID Filter Element	722
Table 6.20-13 Standard Message ID Filter Element Description	722
Table 6.20-14 Extended Message ID Filter Element	723
Table 6.20-15 Extended Message ID Filter Element Description	723
Table 6.21-1 Setting of Accumulation and Conversion Result Precision	793

Table 6.21-2 The Setting of Averaging and Conversion Result Precision..... 793

Table 6.23-1 Peripherals Interconnect Matrix table 833

Table 10.1-1 List of Abbreviations..... 842

1 GENERAL DESCRIPTION

The NuMicro® M253 series 32-bit microcontroller is based on Arm® Cortex®-M23 core using Armv8-M architecture. It runs up to 48 MHz and features up to 128 Kbytes Flash, 16 Kbytes SRAM, 1.75V ~ 5.5V wide operating voltage, and -40°C to +105°C operating temperature.

Crystal-less USB 2.0 full speed device interface

The M253 series includes a crystal-less USB 2.0 full speed device supports maximum 1 Kbytes buffer size, provides up to 17 configurable endpoints, up to 5 VCOMs for configurable Isochronous, Bulk, Interrupt and Control transfer types.

UART interface

The M253 series supports up to 5 sets of UART with RS-485 and IrDA mode. The data payloads of 4 UARTs is equipped with 16/16 bytes entry FIFO for both receive and transmit side. The UART interfaced support flow control and PDMA mode.

CAN FD interface

One set of CAN FD interface, eligible for automotive related and industrial automation applications, is supported.

Rich Pheripherals for comprehensive product application scenarios

The M253 series provides plenty of peripherals such as Timers, Watchdog Timers, PDMA, up to 5 sets of UART, 2 sets of I²C, 1 set of SPI/I²S, Universal Serial Control Interface (USCI), and 6 channels of 16-bit PWM. It also includes high performance analog front-end circuit blocks, such as 12 channels of 12-bit 840 kSPS ADC, and a temperature sensor to enhance product performance, to reduce external components and form factor simultaneously.

Supported packages from QFN33 (5mm x 5mm) to LQFP48 (7mm x 7mm) with pin-compatible for different part numbers make the system design and parts change easily.

For the development, Nuvoton provides the NuMaker M253 evaluation boards and Nu-Link debugger. The 3rd Party IDEs such as Keil® MDK, IAR EWARM and Eclipse IDE with GNU GCC compilers, are also supported.

USCI*: supports UART, SPI or I²C

Product Line	UART	I ² C	SPI/I ² S	USCI*	USBD	CAN FD	Timer	PDMA	ADC	PWM
M253	5	2	1	1	1	1	4	5	12	6

Table 1-1 NuMicro® M253 Series Key Features Support Table

The NuMicro® M253 series is suitable for a wide range of applications such as:

- USB Bridge
- Data Collector
- Battery Management System
- Automotive Related Module

2 FEATURES

Core and System	
Arm® Cortex®-M23 without TrustZone®	• Arm® Cortex®-M23 processor, running up to 48 MHz when $V_{DD} = 1.75V \sim 5.5V$ • Built-in PMSAv8 Memory Protection Unit (MPU) • Built-in Nested Vectored Interrupt Controller (NVIC) • 32-bit Single-cycle hardware multiplier and 32-bit 17-cycle hardware divider • 24-bit system tick timer • Supports Programmable and maskable interrupt • Supports Low Power Sleep mode by WFI and WFE instructions • Supports single cycle I/O access • Supports XOM feature with 1 region
Low power mode and current	• Low Power mode: – Idle mode • Power-down mode (PD) – Fast Wake-up Power-down mode (FWPD)
Wake-up source and wakeup time	• USCI, RTC, WDT, I²C, Timer, UART, BOD, LVR, POR, GPIO, USB, Debug interface, NMI and Reset pin from Power-down mode or Fast Wake-up Power-down mode
Power supply and low voltage detect	• Built-in LDO for wide operating voltage from 1.75V to 5.5V • Core power voltage: 1.5V • Brown-out detector – With 7 levels: 4.4V/3.7V/3.0V/2.7V/2.4V/2.0V/1.8V – Supports Brown-out Interrupt and Reset option • Low Voltage Reset – Threshold voltage levels: 1.55V
Cyclic Redundancy Calculation Unit	• Supports four common polynomials CRC-CCITT, CRC-8, CRC-16, and CRC-32 • Programmable order reverse setting for input data and CRC checksum • Programmable 1's complement setting for input data and CRC checksum. • Supports 8-/16-/32-bit of data width • Programmable seed value

	• 8-bit write mode: 1-AHB clock cycle operation • 16-bit write mode: 2-AHB clock cycle operation • 32-bit write mode: 4-AHB clock cycle operation • Supports using PDMA to write data to perform CRC operation
Security	• 96-bit Unique ID (UID) • 128-bit Unique Customer ID (UCID)
Memories	
Flash	• Up to 128 KB application ROM (APROM) • 4 KB Flash for user program loader (LDROM) • Up to 48 MHz with zero wait state for consecutive address read access • 12 bytes User Configuration Block to control system initiation. • 512B page erase for all embedded Flash • 32-bit and multi-word Flash programming function. • Supports In-System-Programming (ISP), In-Application-Programming (IAP) update embedded Flash memory • Supports CRC-32 checksum calculation function • Supports Flash all one verification function (hardware can check page erase verify) • Hardware external read protection of whole Flash memory by Security Lock Bit • Supports XOM feature with 1 region
SRAM	• 16 KB embedded SRAM • Supports byte-, half-word- and word-access • Supports PDMA mode
Peripheral DMA (PDMA)	• Up to 5 independent configurable channels for automatic data transfer between memories and peripherals • Channel 0, 1 support time-out function • Basic and Scatter-Gather Transfer modes • Each channel supports circular buffer management using Scatter-Gather Transfer mode • Two types of priorities modes: Fixed-priority and Round-robin modes • Transfer data width of 8, 16, and 32 bits • Single and burst transfer type • Source and destination address can be increment or fixed

	• PDMA transfer count up to 65536 • Request source form software, PSIO, SPI/I²S, UART, USCI, EADC, DAC, PWM capture event or TIMER
Clocks	
Clock Source	• Built-in 4.032 MHz internal high speed RC oscillator (MIRC) for system operation • Built-in 48 MHz internal high speed RC oscillator (HIRC) for system operation • Built-in 38.4 kHz internal low speed RC oscillator (LIRC) for Watchdog Timer and wake-up operation. • Built-in 4~32 MHz external high speed crystal oscillator (HXT) for precise timing operation • Built-in 32.768 kHz external low speed crystal oscillator (LXT) for RTC function and low-power system operation • Supports clock on-the-fly switch • Supports clock failure detection for high/low speed external crystal oscillator • HXT clock frequency accuracy detector • Supports exception (NMI) generated once a clock failure detected • Supports divided clock output
Timers	
	TIMER mode
32-bit Timer	• 4 sets of 32-bit timers with 24-bit up counters and 8-bit prescale counters • Independent clock source for each timer • One-shot, Periodic, Toggle and Continuous Counting operation modes • Event counting function to count the event from external pin • Input capture function to capture or reset counter value • External capture pin event for interval measurement. • External capture pin event to reset 24-bit up counter. • Chip wake-up from Idle/Power-down mode if a timer interrupt signal is generated • Timer interrupt flag or external capture interrupt flag to trigger BPWM, EADC and PDMA. • Inter-Timer trigger capture mode
BPWM	• Supports maximum clock frequency up to 96 MHz • Each module provides 6 output channels

	• Supports independent mode for BPWM output/Capture input channel • Supports 12-bit prescaler from 1 to 4096 • Supports 16-bit resolution BPWM counter, each module provides 1 BPWM counter - Up, down or up/down counter operation type • Supports mask function and tri-state enable for each BPWM pin • Supports interrupt on the following events: - BPWM counter match 0, period value or compared value • Supports trigger ADC on the following events: - BPWM counter match 0, period value or compared value • Capture Function Features - Up to 6 capture input channels with 16-bit resolution - Supports rising or falling capture condition - Supports input rising/falling capture interrupt - Supports rising/falling capture with counter reload option
Watchdog	• 20-bit free running up counter for WDT time-out interval • Clock sources from LIRC (default), HCLK/2048 or LXT • 9 selectable time-out period from 488us ~ 32 sec • Able to wake up from Power-down or Idle mode • Interrupt or reset selectable on watchdog time-out • Selectable WDT reset delay period, including 1026, 130, 18 or 3 WDT_CLK reset delay period • Force WDT enabled after chip power on or reset. • WDT time-out wake-up function only if WDT clock source is selected as LIRC or LXT
Window Watchdog	• Clock sources from HCLK/2048 (default) or LIRC • Window set by 6-bit down counter with 11-bit prescaler • WWDT counter suspends in Idle/Power-down mode • Supports Interrupt
RTC	• Software compensation by setting frequency compensate register (FCR), compensated clock accuracy reaches $\pm 5\text{ppm}$ within 5 seconds • RTC counter (second, minute, hour) and calendar counter (day, month, year) • Alarm registers (second, minute, hour, day, month, year) • Selectable 12-hour or 24-hour mode

- Automatic leap year recognition
- Day of the Week counter
- Daylight Saving Time software control
- Periodic time tick interrupt with 8 period options 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 or 1 second
- 1 Hz clock output for RTC calibration
- Wake-up from idle mode and Power-down mode
- 32 kHz oscillator gain control
- RTC Time Tick and Alarm Match interrupt

Analog Interfaces

EADC

- Conversion results held in 6 data registers with valid and overrun indicators.
- Analog input voltage: 0~AV_{DD} (Max to AV_{DD}).
- Reference voltage from AV_{DD}
- 12-bit resolution and 10-bit accuracy guaranteed
- Up to 12 single-end analog external input channels
- Supports 2 internal channels:
 - Band-gap VBG output
 - Temperature sensor input
- Four ADC interrupts (ADINT0~3) with individual interrupt vector addresses.
- ADC clock frequency up to 16 MHz.
- Up to 730 KSPS conversion rate.
- Up to 4 sample modules
 - Each of sample module 0~3 which is configurable for ADC converter channel
 - EADC_CH0~7, 12~15 and trigger source.
 - Configurable PDMA
 - Configured resolution for 12-bit or 16-bit result
 - Supports Left-adjusted result
 - Averaging and oversampling (2ⁿ times, n=0~8) to support up to 16-bit result
 - Configurable sampling time for each sample module.
 - Conversion results held in 6 data registers with valid and overrun indicators.
- Supports digital comparator to monitor conversion result that can be under or over the compare register setting
- Generate an interrupt when conversion result matches the

compare register setting.

- An A/D conversion can be started by:
 - Write 1 to SWTRGn (EADC_SWTRG[n], n = 0~3)
 - Write 1 to INTSWTRGn (EADC_SWTRG[n], n = 16,17) for internal channels
 - External pin STADC
 - Timer0~3 overflow pulse triggers
 - ADINT0/1 interrupt EOC (End of conversion) pulse triggers
 - BPWM triggers
- Supports PDMA transfer
- Auto turn on/off ADC power at power down or operation mode with wait state

Communication Interfaces

UART

- Supports up to 5 UARTs: UART0, UART1, UART2, UART3, and UART4
- UART clock source can be from LIRC
- UART baud rate clock from LXT(32.768 kHz) with 9600bps in Power-down mode
- Baud rate up to 10 Mbps
- Full-duplex asynchronous communications
- Supports one-wire half-duplex communications
- Separates receive and transmit 16/16 bytes FIFO
- Programmable receiver buffer trigger level
- Hardware auto-flow control (CTS and RTS)
- IrDA (SIR) function
 - Supports 3/16 bit duration for normal mode
- RS-485 9-bit mode and direction control
- Programmable baud-rate generator up to 1/16 system clock
- 8-bit receiver FIFO time-out detection function
- Separates receive and transmit 16/16 bytes FIFO (Supported by UART0~UART3)
- Programmable transmitting data delay time between the last stop and the next start bit
- Auto-Baud Rate measurement and baud rate compensation function
- Break error, frame error, parity error and receive/transmit FIFO overflow detection function
- Supports RS-485 mode:

	- RS-485 9-bit mode - Hardware or software enables to program nRTS pin to control RS-485 transmission direction - nCTS, incoming data, Received Data FIFO reached threshold and RS-485 Address Match (AAD mode) wake-up function in Power-down mode. - Hardware or software enables to program nRTS pin to control RS-485 transmission direction • Fully programmable serial-interface: - Programmable number of data bit, 5-, 6-, 7-, 8- bit character - Programmable parity bit, even, odd, no parity or stick parity bit generation and detection - Programmable stop bit, 1, 1.5, or 2 stop bit generation • Supports PDMA mode
SPI	• Supports Master or Slave mode operation • Master and slave mode up to 25 MHz (when chip works at $V_{DD} = 3.0 \sim 5.5V$) • Supports 2-bit Transfer mode • Supports Dual and Quad I/O Transfer mode • Configurable bit length of a transaction word from 8 to 32-bit • Provides separate 8-level depth transmit and receive FIFO buffers • Supports MSB first or LSB first transfer sequence • Supports Byte Reorder function • Supports Byte or Word Suspend mode • Supports PDMA transfer • Supports 3-Wire, no slave selection signal, bi-direction interface • Supports one data channel half-duplex transfer • Supports receive-only mode
I ² C	• Up to 2 sets of I²C device • Master/Slave mode • Bidirectional data transfer between masters and slaves • Multi-master bus (no central master) • 7-bit and 10-bit addressing mode • Standard mode (100 kbps), Fast mode (400 kbps) and Fast mode plus (1 Mbps) • Arbitration between simultaneously transmitting masters without corruption of serial data on the bus • Serial clock synchronization allows devices with different bit rates

to communicate via one serial bus

- Serial clock synchronization can be used as a handshake mechanism to suspend and resume serial transfer
 - Supports 14-bit time-out counter requesting the I²C interrupt if the I²C bus hangs up and timer-out counter overflows
 - Programmable clocks allow versatile rate control
 - Multiple address recognition (four slave address with mask option)
 - Supports setup/hold time programmable
 - Multi-address Power-down wake-up function
 - Supports PDMA transfer
-

SPI Mode

- Up to 1 set of SPI controller
- Master or Slave mode operation
- Configurable bit length of a transfer word from 8 to 32-bit
- Provides separate 4-level of 32-bit (or 8-level of 16-bit) transmit and receive FIFO buffers which depended on SPI setting of data width
- MSB first or LSB first transfer sequence
- Supports byte reorder function
- Byte or Word Suspend mode
- Master and slave mode up to 25 MHz ($V_{DD} = 3.0V \sim 5.5V$)
- Supports one data channel half-duplex transfer
- Supports receive-only mode
- Supports PDMA transfer

SPI/I²S

I²S Mode

- Up to 1 sets of I²S by SPI controllers
 - Interface with external audio CODEC
 - Supports Master and Slave mode
 - Capable of handling 8-, 16-, 24- and 32-bit word sizes
 - Mono and stereo audio data
 - PCM mode A, PCM mode B, I²S and MSB justified data format
 - Each provides two 4-level FIFO data buffers, one for transmitting and the other for receiving
 - Generates interrupt requests when buffer levels cross a programmable boundary
 - Each supports two PDMA requests, one for transmitting and the other for receiving
-

**Universal Serial Control
Interface (USCI)**

- Up to 1 set of USCI
- Supports UART, SPI and I²C function
- Single byte TX and RX buffer mode

USCI_UART

- One transmit buffer and two receive buffer for data payload
- Hardware auto flow control function and programmable flow control trigger level
- Programmable baud-rate generator
- Supports 9-bit data transfer
- Baud rate detection by built-in capture event of baud rate generator
- Supports Wake-up function (Data and nCTS Wakeup Only)
- Supports PDMA transfer

USCI_SPI

- Master or Slave mode operation
- Configurable bit length of a transfer word from 4 to 16-bit
- One transmit buffer and two receive buffer for data payload
- MSB first or LSB first transfer sequence
- Word suspend function
- Supports PDMA transfer
- Supports 3-wire, no slave select signal, bi-direction interface
- Wake-up function: input slave select transition
- Supports one data channel half-duplex transfer

USCI_I2C

- Full master and slave device capability
- 7-bit/10-bit addressing mode
- Communication in Standard mode (100 kbps), Fast mode (up to 400 kbps) and Fast mode plus (1 Mbps)
- Multi-master bus
- One transmit buffer and two receive buffer for data payload
- 10-bit bus time out capability
- Supports Bus monitor mode
- Wake-up by data toggle or address match in Power-down mode
- Multiple address recognition
- Setup/hold time programmable

GPIO

- Four I/O modes:

- Quasi bi-direction
- Push-Pull output
- Open-Drain output
- Input only with high impendence
- TTL/Schmitt trigger input selectable
- I/O pin configured as interrupt source with edge/level trigger setting
- Independent pull-up/pull-down control
- High driver and high sink current I/O (up to 16 mA at 5V, 25°C)
- Minimum I/O Speed
 - 25 MHz when $V_{DD} = 2.7 \sim 5.5 \text{ V}$ ($-40^{\circ}\text{C} \sim +105^{\circ}\text{C}$, $CL=35p$, high skew rate enabled)
 - 10 MHz when $V_{DD} = 1.75 \sim 5.5 \text{ V}$ ($-40^{\circ}\text{C} \sim +105^{\circ}\text{C}$, $CL=35p$, high skew rate enabled)
- Software selectable slew rate control
- Supports wake-up function
- Supports I/O de-bounce with LIRC at power down
- I/O configurations of multi-function pin are controlled by module or MFOS register settings.

Advanced Connectivity

USB 2.0 Full Speed

- Compliant with USB 2.0 Full-Speed specification
- Provides 1 interrupt vector with 4 different interrupt events (NEVWK, VBUSDET, USB and BUS)
- Suspend function when no bus activity exists for 3 ms
- Supports 17 endpoints for configurable Control/Bulk/Interrupt/Isochronous transfer types and maximum 1024 bytes buffer size
- Provides remote wake-up capability
- Start of Frame (SOF) locked clock pulse generation
- Supports USB 2.0 Link Power Management (LPM)
- Supports Crystal-less function

CAN-FD

- Supports protocol version 2.0 part A, B and ISO 11898-1:2015
- Up to 64 data bytes supported
- Supports CAN Error Logging
- Supports AUTOSAR and J1939
- Two Receive FIFOs of three payloads each (up to 64 Bytes per payload)

-
- Separate signaling on reception of High Priority Messages
 - Configurable Transmit FIFO / queue of three payload (up to 64 Bytes per payload)
 - Support programmable loop-back test mode
 - Support maskable module interrupts
 - Support Power-down
-

3 PARTS INFORMATION

3.1 Package Type

Package is Halogen-free, RoHS-compliant and TSCA-compliant.

Part No.	TSSOP20	TSSOP28	QFN33	LQFP48	LQFP64	LQFP80	LQFP128
M253_64K				M253LD3AE			
M253_128K			M253ZE3AE	M253LE3AE			

3.2 M253 Selection Guide

PART NUMBER		M253ZE3AE	M253LD3AE	M253LE3AE
Flash (KB)		128	64	128
SRAM (KB)		16	16	16
LDROM (KB)		4		
I/O		22	37	
32-bit Timer		4		
WDT/WWDT		√		
LXT		√		
RTC		√		
PWM		6		
Connectivity	USCI*	1		
	UART	5		
	SPI/I ² S	1		
	I ² C	2		
	CAN FD	1		
	USB 2.0 FS	1		
PDMA		5		
EBI		-		
CRC		√		
PLL(MHz)		-		
ISO-60730		√		
ACMP		2		
12-bit 840 kSPS SAR ADC		10	12	
Package		QFN33	LQFP48	

3.3 M253 Selection Code

M2	53	L	E	3	A	E
Core	Line	Package	Flash	SRAM	Reserve	Temperature
Cortex-M2	53: USB, CAN FD	Z: QFN33 (5x5 mm) L: LQFP48 (7x7 mm)	D: 64 KB E: 128 KB	3: 16 KB		E: -40°C ~ 105°C

4 PIN CONFIGURATION

Users can find pin configuration information in the M253 Multi-function Pin diagram sections or by using [NuTool - PinConfigure](#). The NuTool - PinConfigure contains all NuMicro® Family chip series with all part number, and helps users configure GPIO multi-function correctly and handily.

4.1 Pin Configuration

4.1.1 M253 Pin Diagram

4.1.1.1 M253 QFN 33-Pin Diagram

Corresponding Part Number: M253ZE3AE

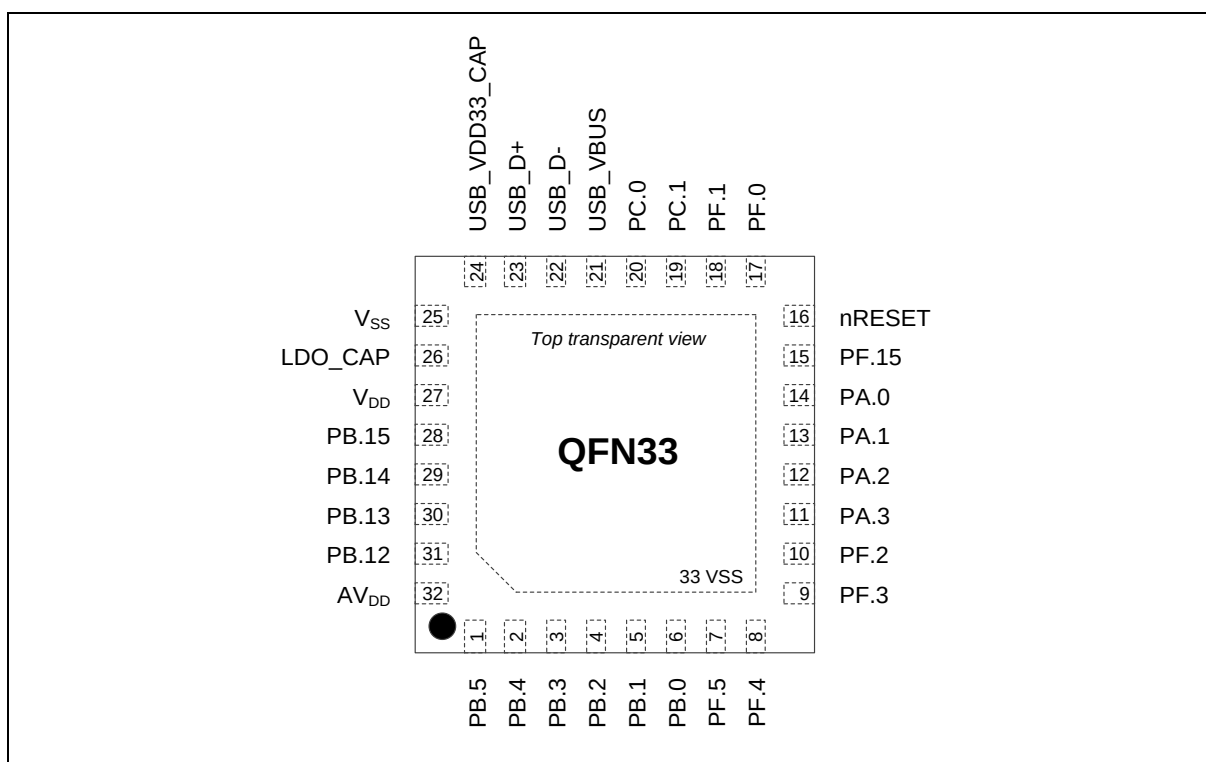


Figure 4.1-1 M253 QFN 33-pin Diagram

4.1.1.2 M253 LQFP 48-Pin Diagram

Corresponding Part Number: M253LD3AE, M253LE3AE

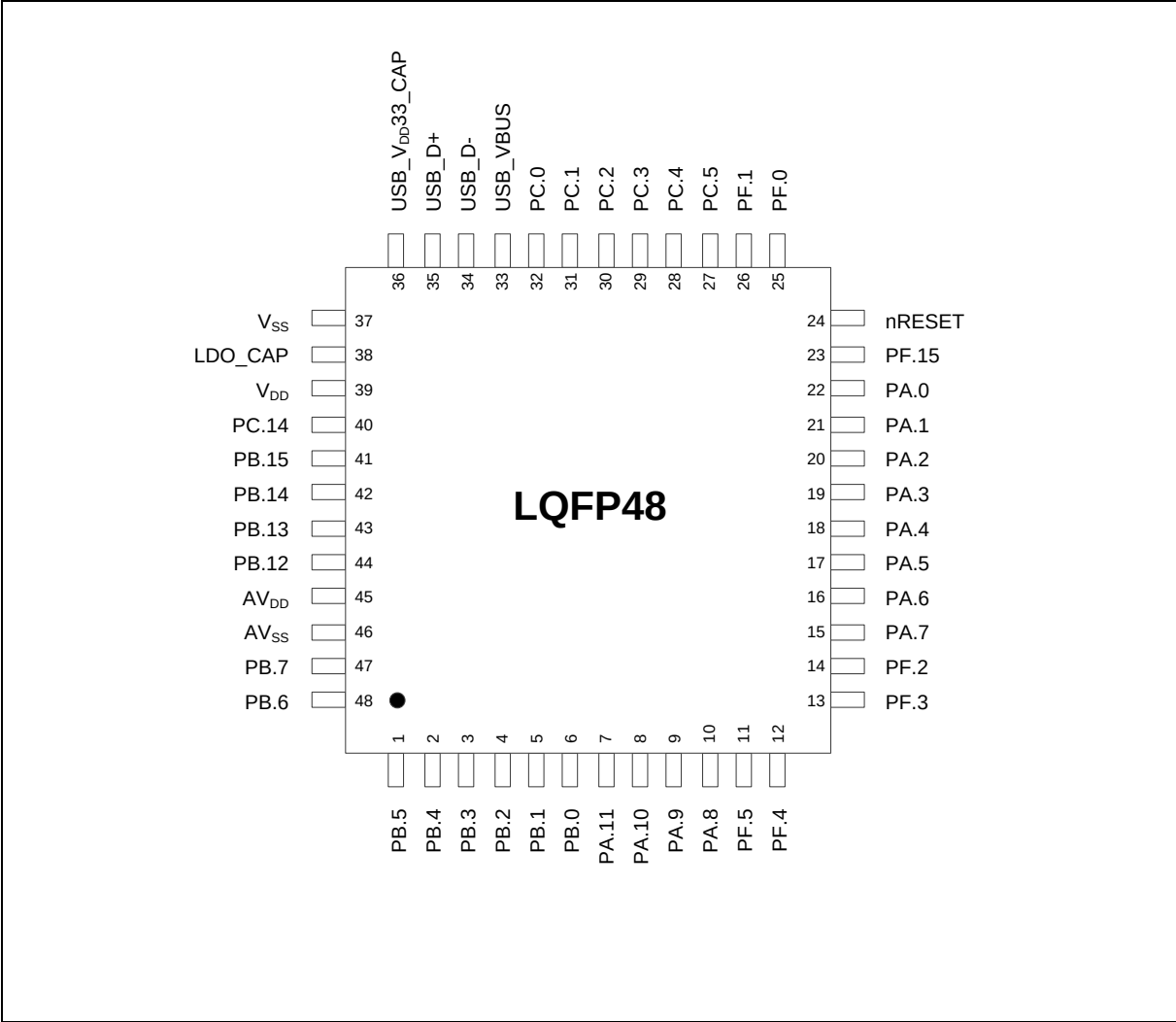


Figure 4.1-2 M253 LQFP 48-pin Diagram

4.1.2 M253 Multi-function Pin Diagram

4.1.2.1 M253 QFN 33-Pin Multi-function Pin Diagram

Corresponding Part Number: M253ZE3AE

M253ZE3AE

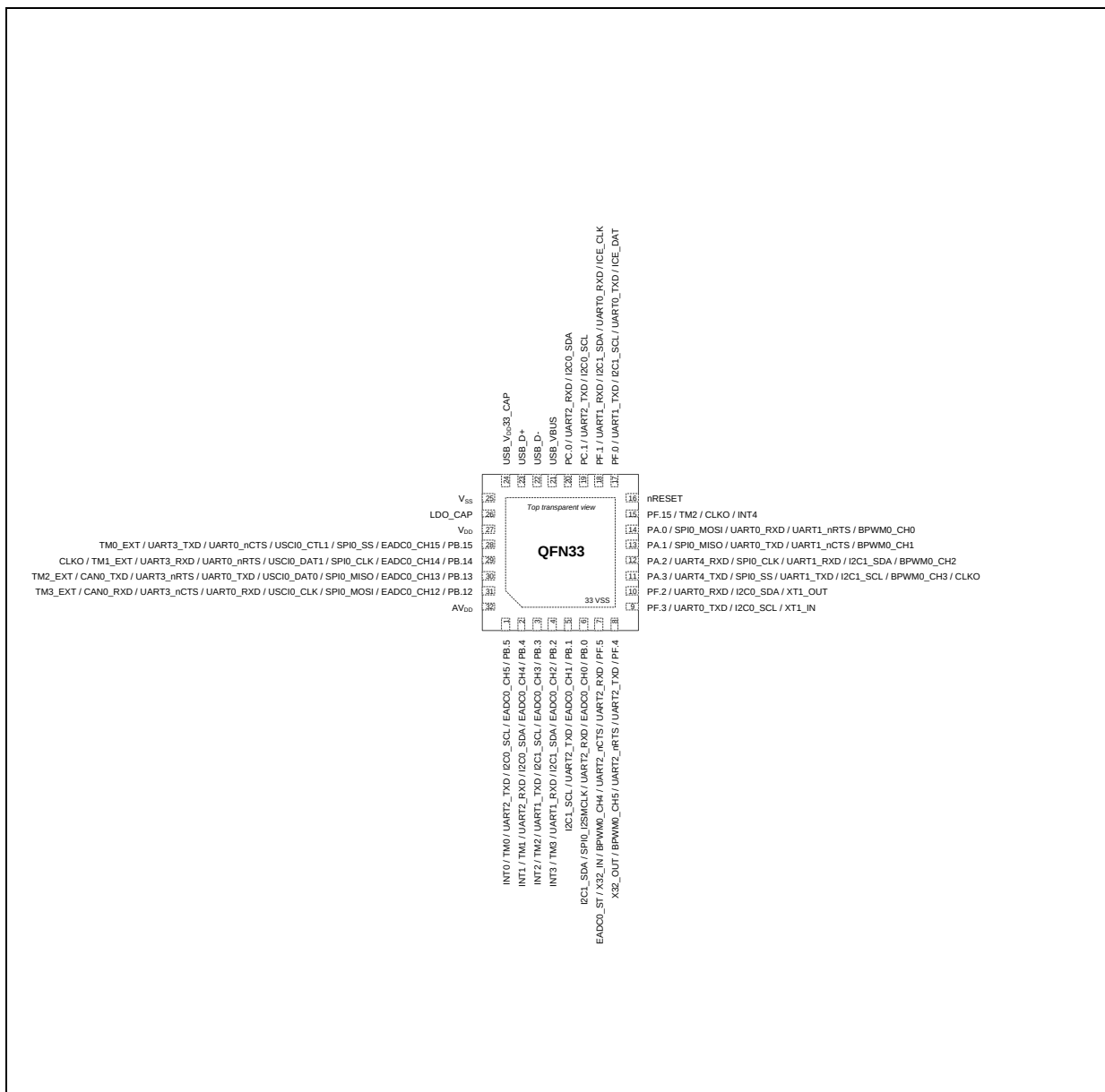


Figure 4.1-3 M253ZE3AE Multi-function Pin Diagram

Pin	M253ZE3AE Pin Function
1	PB.5 / EADC0_CH5 / I2C0_SCL / UART2_TXD / TM0 / INT0
2	PB.4 / EADC0_CH4 / I2C0_SDA / UART2_RXD / TM1 / INT1
3	PB.3 / EADC0_CH3 / I2C1_SCL / UART1_TXD / TM2 / INT2

Pin	M253ZE3AE Pin Function
4	PB.2 / EADC0_CH2 / I2C1_SDA / UART1_RXD / TM3 / INT3
5	PB.1 / EADC0_CH1 / UART2_TXD / I2C1_SCL
6	PB.0 / EADC0_CH0 / UART2_RXD / SPI0_I2SMCLK / I2C1_SDA
7	PF.5 / UART2_RXD / UART2_nCTS / BPWM0_CH4 / X32_IN / EADC0_ST
8	PF.4 / UART2_TXD / UART2_nRTS / BPWM0_CH5 / X32_OUT
9	PF.3 / UART0_TXD / I2C0_SCL / XT1_IN
10	PF.2 / UART0_RXD / I2C0_SDA / XT1_OUT
11	PA.3 / UART4_TXD / SPI0_SS / UART1_TXD / I2C1_SCL / BPWM0_CH3 / CLK0
12	PA.2 / UART4_RXD / SPI0_CLK / UART1_RXD / I2C1_SDA / BPWM0_CH2
13	PA.1 / SPI0_MISO / UART0_TXD / UART1_nCTS / BPWM0_CH1
14	PA.0 / SPI0_MOSI / UART0_RXD / UART1_nRTS / BPWM0_CH0
15	PF.15 / TM2 / CLK0 / INT4
16	nRESET
17	PF.0 / UART1_TXD / I2C1_SCL / UART0_TXD / ICE_DAT
18	PF.1 / UART1_RXD / I2C1_SDA / UART0_RXD / ICE_CLK
19	PC.1 / UART2_TXD / I2C0_SCL
20	PC.0 / UART2_RXD / I2C0_SDA
21	USB_VBUS
22	USB_D-
23	USB_D+
24	USB_VDD33_CAP
25	VSS
26	LDO_CAP
27	VDD
28	PB.15 / EADC0_CH15 / SPI0_SS / USCI0_CTL1 / UART0_nCTS / UART3_TXD / TM0_EXT
29	PB.14 / EADC0_CH14 / SPI0_CLK / USCI0_DAT1 / UART0_nRTS / UART3_RXD / TM1_EXT / CLK0
30	PB.13 / EADC0_CH13 / SPI0_MISO / USCI0_DAT0 / UART0_TXD / UART3_nRTS / CAN0_TXD / TM2_EXT
31	PB.12 / EADC0_CH12 / SPI0_MOSI / USCI0_CLK / UART0_RXD / UART3_nCTS / CAN0_RXD / TM3_EXT
32	AVDD

Table 4.1-1 M253ZE3AE Multi-function Pin Table

4.1.2.2 M253 LQFP 48-Pin Multi-function Pin Diagram

Corresponding Part Number: M253LD3AE, M253LE3AE

M253LD3AE/M253LE3AE

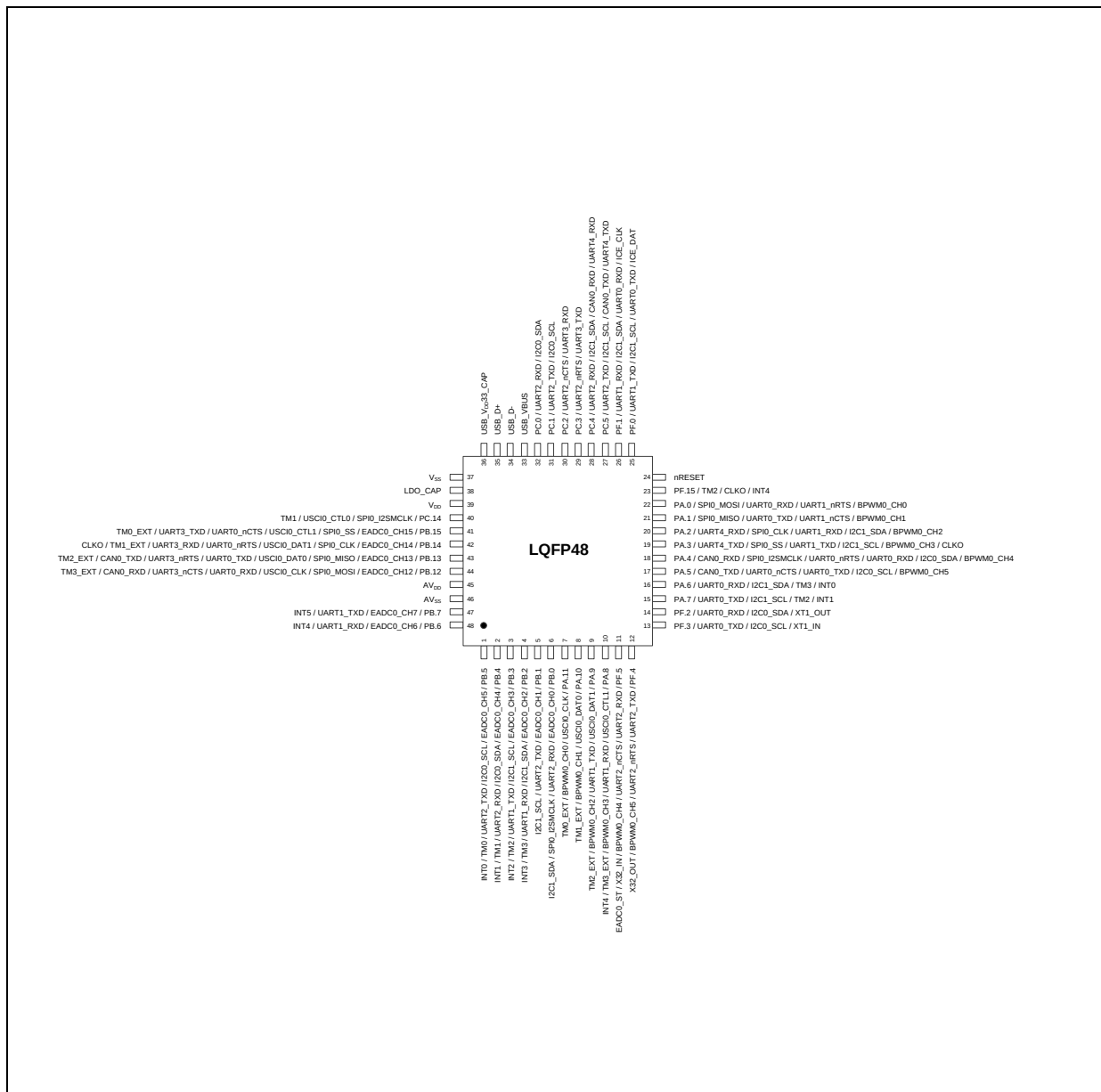


Figure 4.1-4 M253LD3AE/M253LE3AE Multi-function Pin Diagram

Pin	M253LD3AE / M253LE3AE Pin Function
1	PB.5 / EADC0_CH5 / I2C0_SCL / UART2_TXD / TM0 / INT0
2	PB.4 / EADC0_CH4 / I2C0_SDA / UART2_RXD / TM1 / INT1
3	PB.3 / EADC0_CH3 / I2C1_SCL / UART1_TXD / TM2 / INT2
4	PB.2 / EADC0_CH2 / I2C1_SDA / UART1_RXD / TM3 / INT3

Pin	M253LD3AE / M253LE3AE Pin Function
5	PB.1 / EADC0_CH1 / UART2_TXD / I2C1_SCL
6	PB.0 / EADC0_CH0 / UART2_RXD / SPI0_I2SMCLK / I2C1_SDA
7	PA.11 / USCIO_CLK / BPWM0_CH0 / TM0_EXT
8	PA.10 / USCIO_DAT0 / BPWM0_CH1 / TM1_EXT
9	PA.9 / USCIO_DAT1 / UART1_TXD / BPWM0_CH2 / TM2_EXT
10	PA.8 / USCIO_CTL1 / UART1_RXD / BPWM0_CH3 / TM3_EXT / INT4
11	PF.5 / UART2_RXD / UART2_nCTS / BPWM0_CH4 / X32_IN / EADC0_ST
12	PF.4 / UART2_TXD / UART2_nRTS / BPWM0_CH5 / X32_OUT
13	PF.3 / UART0_TXD / I2C0_SCL / XT1_IN
14	PF.2 / UART0_RXD / I2C0_SDA / XT1_OUT
15	PA.7 / UART0_TXD / I2C1_SCL / TM2 / INT1
16	PA.6 / UART0_RXD / I2C1_SDA / TM3 / INT0
17	PA.5 / CAN0_TXD / UART0_nCTS / UART0_TXD / I2C0_SCL / BPWM0_CH5
18	PA.4 / CAN0_RXD / SPI0_I2SMCLK / UART0_nRTS / UART0_RXD / I2C0_SDA / BPWM0_CH4
19	PA.3 / UART4_TXD / SPI0_SS / UART1_TXD / I2C1_SCL / BPWM0_CH3 / CLK0
20	PA.2 / UART4_RXD / SPI0_CLK / UART1_RXD / I2C1_SDA / BPWM0_CH2
21	PA.1 / SPI0_MISO / UART0_TXD / UART1_nCTS / BPWM0_CH1
22	PA.0 / SPI0_MOSI / UART0_RXD / UART1_nRTS / BPWM0_CH0
23	PF.15 / TM2 / CLK0 / INT4
24	nRESET
25	PF.0 / UART1_TXD / I2C1_SCL / UART0_TXD / ICE_DAT
26	PF.1 / UART1_RXD / I2C1_SDA / UART0_RXD / ICE_CLK
27	PC.5 / UART2_TXD / I2C1_SCL / CAN0_TXD / UART4_TXD
28	PC.4 / UART2_RXD / I2C1_SDA / CAN0_RXD / UART4_RXD
29	PC.3 / UART2_nRTS / UART3_TXD
30	PC.2 / UART2_nCTS / UART3_RXD
31	PC.1 / UART2_TXD / I2C0_SCL
32	PC.0 / UART2_RXD / I2C0_SDA
33	USB_VBUS
34	USB_D-
35	USB_D+
36	USB_VDD33_CAP

Pin	M253LD3AE / M253LE3AE Pin Function
37	VSS
38	LDO_CAP
39	V _{DD}
40	PC.14 / SPI0_I2SMCLK / USCIO_CTL0 / TM1
41	PB.15 / EADC0_CH15 / SPI0_SS / USCIO_CTL1 / UART0_nCTS / UART3_TXD / TM0_EXT
42	PB.14 / EADC0_CH14 / SPI0_CLK / USCIO_DAT1 / UART0_nRTS / UART3_RXD / TM1_EXT / CLKO
43	PB.13 / EADC0_CH13 / SPI0_MISO / USCIO_DAT0 / UART0_TXD / UART3_nRTS / CAN0_TXD / TM2_EXT
44	PB.12 / EADC0_CH12 / SPI0_MOSI / USCIO_CLK / UART0_RXD / UART3_nCTS / CAN0_RXD / TM3_EXT
45	AV _{DD}
46	AVSS
47	PB.7 / EADC0_CH7 / UART1_TXD / INT5
48	PB.6 / EADC0_CH6 / UART1_RXD / INT4

Table 4.1-2 M253LD3AE/M253LE3AE Multi-function Pin Table

4.2 M253 Pin Mapping

Different part number with same package might has different function. Please refer to the M253 Selection Guide section, Pin Configuration section or [NuTool - PinConfig](#).

Corresponding Part Number: M253

	M253	
Pin Name	33 Pin	48 Pin
PB.5	1	1
PB.4	2	2
PB.3	3	3
PB.2	4	4
PB.1	5	5
PB.0	6	6
PA.11		7
PA.10		8
PA.9		9
PA.8		10
PF.5	7	11
PF.4	8	12
PF.3	9	13
PF.2	10	14
PA.7		15
PA.6		16
PA.5		17
PA.4		18
PA.3	11	19
PA.2	12	20
PA.1	13	21
PA.0	14	22
PF.15	15	23
nRESET	16	24
PF.0	17	25
PF.1	18	26
PC.5		27
PC.4		28
PC.3		29

PC.2		30
PC.1	19	31
PC.0	20	32
USB_V _{BUS}	21	33
USB_D-	22	34
USB_D+	23	35
USB_V _{DD33_CAP}	24	36
V _{SS}	25	37
LDO_CAP	26	38
V _{DD}	27	39
PC.14		40
PB.15	28	41
PB.14	29	42
PB.13	30	43
PB.12	31	44
AV _{DD}	32	45
AV _{SS}		46
PB.7		47
PB.6		48

4.3 M253 Pin Functional Description

Group	Pin Name	GPIO	MFP	Type	Description
BPWM0	BPWM0_CH0	PA.11	MFP9	I/O	BPWM0 channel 0 output/capture input.
		PA.0	MFP12	I/O	
	BPWM0_CH1	PA.10	MFP9	I/O	BPWM0 channel 1 output/capture input.
		PA.1	MFP12	I/O	
	BPWM0_CH2	PA.9	MFP9	I/O	BPWM0 channel 2 output/capture input.
		PA.2	MFP12	I/O	
	BPWM0_CH3	PA.8	MFP9	I/O	BPWM0 channel 3 output/capture input.
		PA.3	MFP12	I/O	
	BPWM0_CH4	PF.5	MFP8	I/O	BPWM0 channel 4 output/capture input.
		PA.4	MFP12	I/O	
	BPWM0_CH5	PF.4	MFP8	I/O	BPWM0 channel 5 output/capture input.
		PA.5	MFP12	I/O	
CAN0	CAN0_RXD	PA.4	MFP2	I	CAN0 bus receiver input.
		PC.4	MFP10	I	
		PB.12	MFP9	I	
	CAN0_TXD	PA.5	MFP2	O	CAN0 bus transmitter output.
		PC.5	MFP10	O	
		PB.13	MFP9	O	
CLKO	CLKO	PA.3	MFP14	O	Clock Out
		PF.15	MFP14	O	
		PB.14	MFP14	O	
EADC0	EADC0_CH0	PB.0	MFP1	A	EADC0 channel 0 analog input.
	EADC0_CH1	PB.1	MFP1	A	EADC0 channel 1 analog input.
	EADC0_CH2	PB.2	MFP1	A	EADC0 channel 2 analog input.
	EADC0_CH3	PB.3	MFP1	A	EADC0 channel 3 analog input.
	EADC0_CH4	PB.4	MFP1	A	EADC0 channel 4 analog input.
	EADC0_CH5	PB.5	MFP1	A	EADC0 channel 5 analog input.
	EADC0_CH6	PB.6	MFP1	A	EADC0 channel 6 analog input.
	EADC0_CH7	PB.7	MFP1	A	EADC0 channel 7 analog input.
	EADC0_CH12	PB.12	MFP1	A	EADC0 channel 12 analog input.
	EADC0_CH13	PB.13	MFP1	A	EADC0 channel 13 analog input.

Group	Pin Name	GPIO	MFP	Type	Description
	EADC0_CH14	PB.14	MFP1	A	EADC0 channel 14 analog input.
	EADC0_CH15	PB.15	MFP1	A	EADC0 channel 15 analog input.
	EADC0_ST	PF.5	MFP11	I	EADC0 external trigger input.
I2C0	I2C0_SCL	PB.5	MFP6	I/O	I2C0 clock pin.
		PF.3	MFP4	I/O	
		PA.5	MFP9	I/O	
		PC.1	MFP9	I/O	
	I2C0_SDA	PB.4	MFP6	I/O	I2C0 data input/output pin.
		PF.2	MFP4	I/O	
		PA.4	MFP9	I/O	
		PC.0	MFP9	I/O	
I2C1	I2C1_SCL	PB.3	MFP4	I/O	I2C1 clock pin.
		PB.1	MFP9	I/O	
		PA.7	MFP8	I/O	
		PA.3	MFP9	I/O	
		PF.0	MFP3	I/O	
		PC.5	MFP9	I/O	
	I2C1_SDA	PB.2	MFP4	I/O	I2C1 data input/output pin.
		PB.0	MFP9	I/O	
		PA.6	MFP8	I/O	
		PA.2	MFP9	I/O	
		PF.1	MFP3	I/O	
		PC.4	MFP9	I/O	
ICE	ICE_CLK	PF.1	MFP14	I	Serial wired debugger clock pin. Note: It is recommended to use 100 kΩ pull-up resistor on ICE_CLK pin.
	ICE_DAT	PF.0	MFP14	I/O	Serial wired debugger data pin. Note: It is recommended to use 100 kΩ pull-up resistor on ICE_DAT pin.
INT0	INT0	PB.5	MFP15	I	External interrupt 0 input pin.
		PA.6	MFP15	I	
INT1	INT1	PB.4	MFP15	I	External interrupt 1 input pin.
		PA.7	MFP15	I	

Group	Pin Name	GPIO	MFP	Type	Description
INT2	INT2	PB.3	MFP15	I	External interrupt 2 input pin.
INT3	INT3	PB.2	MFP15	I	External interrupt 3 input pin.
INT4	INT4	PA.8	MFP15	I	External interrupt 4 input pin.
		PF.15	MFP15	I	
		PB.6	MFP13	I	
INT5	INT5	PB.7	MFP13	I	External interrupt 5 input pin.
SPI0	SPI0_CLK	PA.2	MFP4	I/O	SPI0 serial clock pin.
		PB.14	MFP4	I/O	
	SPI0_I2SMCLK	PB.0	MFP8	I/O	SPI0 I ² S master clock output pin
		PA.4	MFP4	I/O	
		PC.14	MFP4	I/O	
	SPI0_MISO	PA.1	MFP4	I/O	SPI0 MISO (Master In, Slave Out) pin.
		PB.13	MFP4	I/O	
	SPI0_MOSI	PA.0	MFP4	I/O	SPI0 MOSI (Master Out, Slave In) pin.
		PB.12	MFP4	I/O	
	SPI0_SS	PA.3	MFP4	I/O	SPI0 slave select pin.
		PB.15	MFP4	I/O	
TM0	TM0	PB.5	MFP14	I/O	Timer0 event counter input/toggle output pin.
	TM0_EXT	PA.11	MFP13	I/O	Timer0 external capture input/toggle output pin.
		PB.15	MFP13	I/O	
TM1	TM1	PB.4	MFP14	I/O	Timer1 event counter input/toggle output pin.
		PC.14	MFP13	I/O	
	TM1_EXT	PA.10	MFP13	I/O	Timer1 external capture input/toggle output pin.
		PB.14	MFP13	I/O	
TM2	TM2	PB.3	MFP14	I/O	Timer2 event counter input/toggle output pin.
		PA.7	MFP14	I/O	
		PF.15	MFP13	I/O	
	TM2_EXT	PA.9	MFP13	I/O	Timer2 external capture input/toggle output pin.
		PB.13	MFP13	I/O	
TM3	TM3	PB.2	MFP14	I/O	Timer3 event counter input/toggle output pin.
		PA.6	MFP14	I/O	

Group	Pin Name	GPIO	MFP	Type	Description
	TM3_EXT	PA.8	MFP13	I/O	Timer3 external capture input/toggle output pin.
		PB.12	MFP13	I/O	
UART0	UART0_RXD	PF.2	MFP3	I	UART0 data receiver input pin.
		PA.6	MFP7	I	
		PA.4	MFP8	I	
		PA.0	MFP7	I	
		PF.1	MFP4	I	
		PB.12	MFP6	I	
	UART0_TXD	PF.3	MFP3	O	UART0 data transmitter output pin.
		PA.7	MFP7	O	
		PA.5	MFP8	O	
		PA.1	MFP7	O	
		PF.0	MFP4	O	
		PB.13	MFP6	O	
	UART0_nCTS	PA.5	MFP7	I	UART0 clear to send input pin.
		PB.15	MFP6	I	
	UART0_nRTS	PA.4	MFP7	O	UART0 request to Send output pin.
		PB.14	MFP6	O	
UART1	UART1_RXD	PB.2	MFP6	I	UART1 data receiver input pin.
		PA.8	MFP7	I	
		PA.2	MFP8	I	
		PF.1	MFP2	I	
		PB.6	MFP6	I	
	UART1_TXD	PB.3	MFP6	O	UART1 data transmitter output pin.
		PA.9	MFP7	O	
		PA.3	MFP8	O	
		PF.0	MFP2	O	
		PB.7	MFP6	O	
	UART1_nCTS	PA.1	MFP8	I	UART1 clear to send input pin.
	UART1_nRTS	PA.0	MFP8	O	UART1 request to Send output pin.
UART2	UART2_RXD	PB.4	MFP13	I	UART2 data receiver input pin.
		PB.0	MFP7	I	

Group	Pin Name	GPIO	MFP	Type	Description
		PF.5	MFP2	I	
		PC.4	MFP8	I	
		PC.0	MFP8	I	
	UART2_TXD	PB.5	MFP13	O	UART2 data transmitter output pin.
		PB.1	MFP7	O	
		PF.4	MFP2	O	
		PC.5	MFP8	O	
		PC.1	MFP8	O	
	UART2_nCTS	PF.5	MFP4	I	UART2 clear to Send input pin.
		PC.2	MFP8	I	
	UART2_nRTS	PF.4	MFP4	O	UART2 request to Send output pin.
		PC.3	MFP8	O	
UART3	UART3_RXD	PC.2	MFP11	I	UART3 data receiver input pin.
		PB.14	MFP7	I	
	UART3_TXD	PC.3	MFP11	O	UART3 data transmitter output pin.
		PB.15	MFP7	O	
	UART3_nCTS	PB.12	MFP7	I	UART3 clear to Send input pin.
	UART3_nRTS	PB.13	MFP7	O	UART3 request to Send output pin.
UART4	UART4_RXD	PA.2	MFP2	I	UART4 data receiver input pin.
		PC.4	MFP11	I	
	UART4_TXD	PA.3	MFP2	O	UART4 data transmitter output pin.
		PC.5	MFP11	O	
USC10	USC10_CLK	PA.11	MFP6	I/O	USC10 clock pin.
		PB.12	MFP5	I/O	
	USC10_CTL0	PC.14	MFP5	I/O	USC10 control 0 pin.
		PA.8	MFP6	I/O	
	USC10_CTL1	PB.15	MFP5	I/O	USC10 control 1 pin.
		PA.10	MFP6	I/O	
	USC10_DAT0	PB.13	MFP5	I/O	USC10 data 0 pin.
		PA.9	MFP6	I/O	
	USC10_DAT1	PB.14	MFP5	I/O	USC10 data 1 pin.
X32	X32_IN	PF.5	MFP10	I	External 32.768 kHz crystal input pin.

Group	Pin Name	GPIO	MFP	Type	Description
	X32_OUT	PF.4	MFP10	O	External 32.768 kHz crystal output pin.
XT1	XT1_IN	PF.3	MFP10	I	External 4~32 MHz (high speed) crystal input pin.
	XT1_OUT	PF.2	MFP10	O	External 4~32 MHz (high speed) crystal output pin.

5 BLOCK DIAGRAM

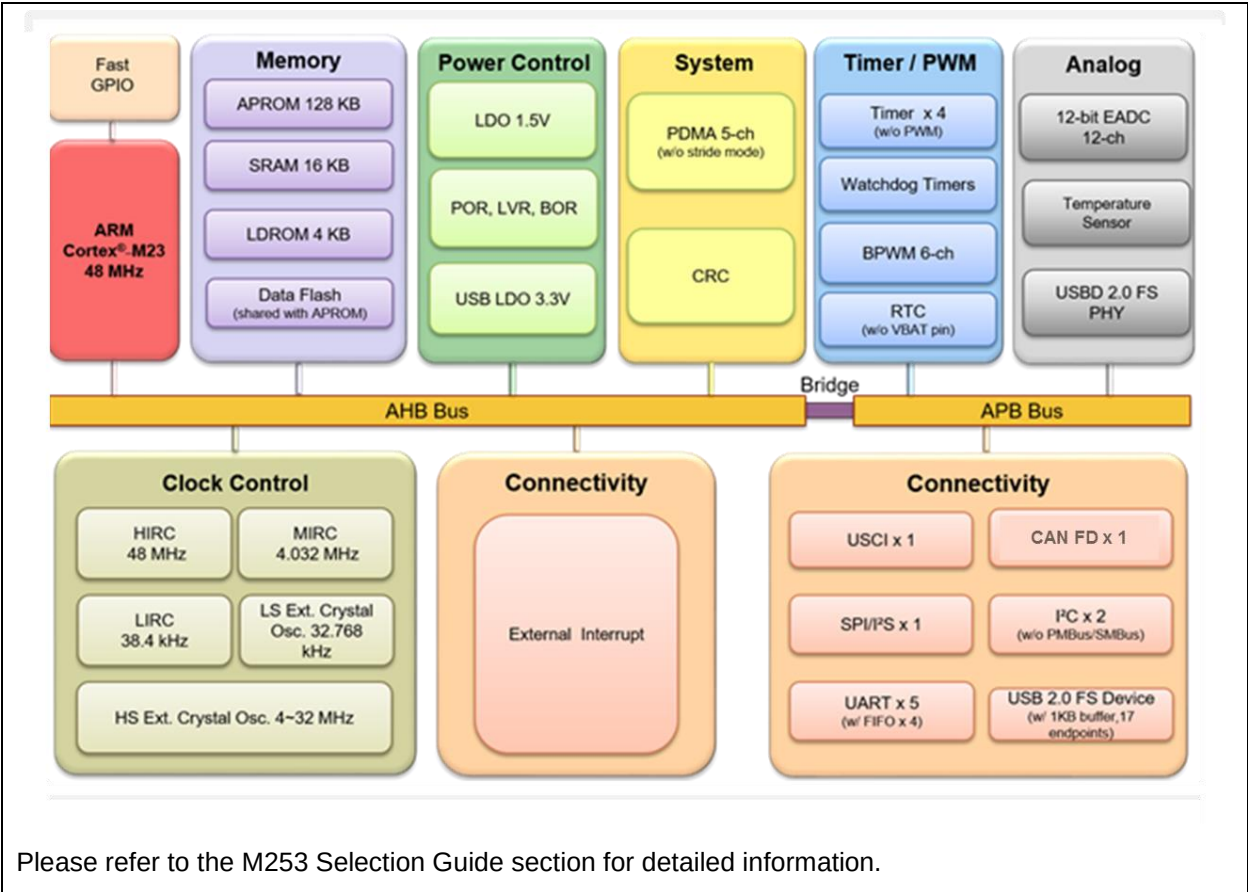


Figure 4.3-1 M253 Block Diagram

6 FUNCTIONAL DESCRIPTION

6.1 Arm® Cortex®-M23 Core

The Cortex®-M23 processor is a low gate count, two-stage, and highly energy efficient 32-bit RISC processor, which has an AMBA AHB5 interface supporting Arm® TrustZone® technology, a debug access port supporting serial wire debug and single-cycle I/O ports. It has an NVIC component and MPU for memory-protection functionality. The processor also supports Security Extension. The NuMicro® M253 is embedded with Cortex®-M23 processor. Figure 6.1-1 shows the functional controller of the processor.

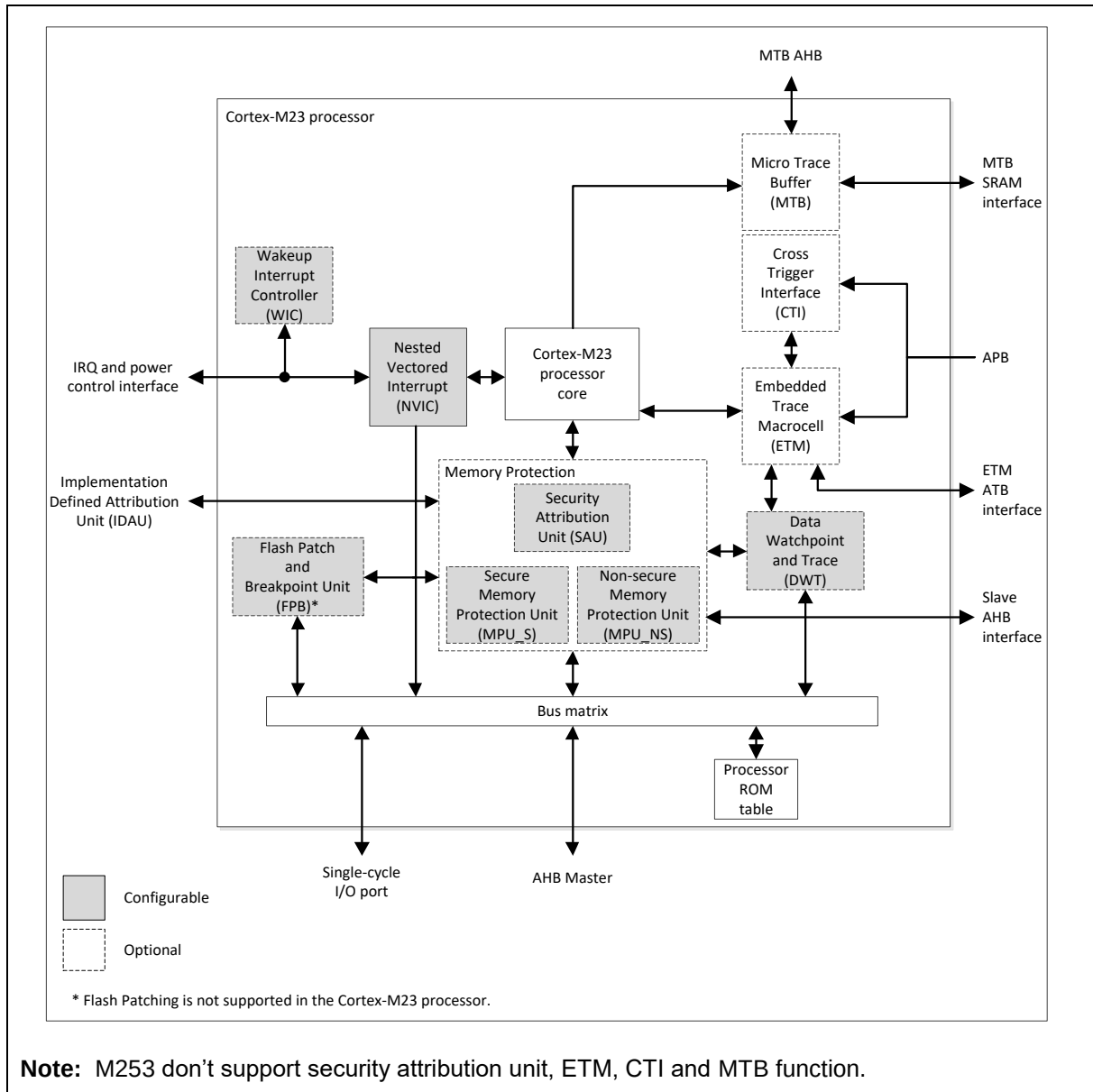


Figure 6.1-1 Cortex®-M23 Block Diagram

Cortex®-M23 processor features:

- Arm®v8-M Baseline architecture.
- Arm®v8-M Baseline Thumb®-2 instruction set that combines high code density with 32-bit performance.
- Support for single-cycle I/O access.
- Power control optimization of system components.
- Integrated sleep modes for low power consumption.
- Optimized code fetching for reduced Flash and ROM power consumption.
- A 32-bit Single cycle Hardware multiplier.
- A 32-bit Hardware divider.
- Deterministic, high-performance interrupt handling for time-critical applications.
- Deterministic instruction cycle timing.
- Support for system level debug authentication.
- Support for Arm® Debug Interface Architecture ADIV5.1 Serial Wire Debug (SWD).
- ETM for instruction trace.
- Separated privileged and unprivileged modes.
- Security Extension supporting a Secure and a Non-secure state.
- Security Attribution Unit (SAU).
- SysTick timers for both Secure and Non-secure states.
- A Nested Vectored Interrupt Controller (NVIC) closely integrated with the processor with up to 240 interrupts.

6.2 System Manager

6.2.1 Overview

System management includes the following sections:

- System Reset
- System Power Distribution
- SRAM Memory Organization
- System Timer (SysTick)
- Nested Vectored Interrupt Controller (NVIC)
- System Control register

6.2.2 System Reset

The system reset can be issued by one of the events listed below. These reset event flags can be read from SYS_RSTSTS register to determine the reset source. Hardware reset sources are from peripheral signals. Software reset can trigger reset through setting control registers.

- Hardware Reset Sources
 - Power-on Reset
 - Low level on the nRESET pin with glitch filter time 24us
 - Watchdog Time-out Reset and Window Watchdog Reset (WDT/MWDT Reset)
 - Low Voltage Reset (LVR)
 - Brown-out Detector Reset (BOD Reset)
 - CPU Lockup Reset
- Software Reset Sources
 - CHIP Reset will reset whole chip by writing 1 to CHIPRST (SYS_IPRST0[0])
 - MCU Reset to reboot but keeping the booting setting from APROM or LDROM by writing 1 to SYSRESETREQ (AIRC[2])
 - CPU Reset for Cortex®-M23 core Only by writing 1 to CPURST (SYS_IPRST0[1])

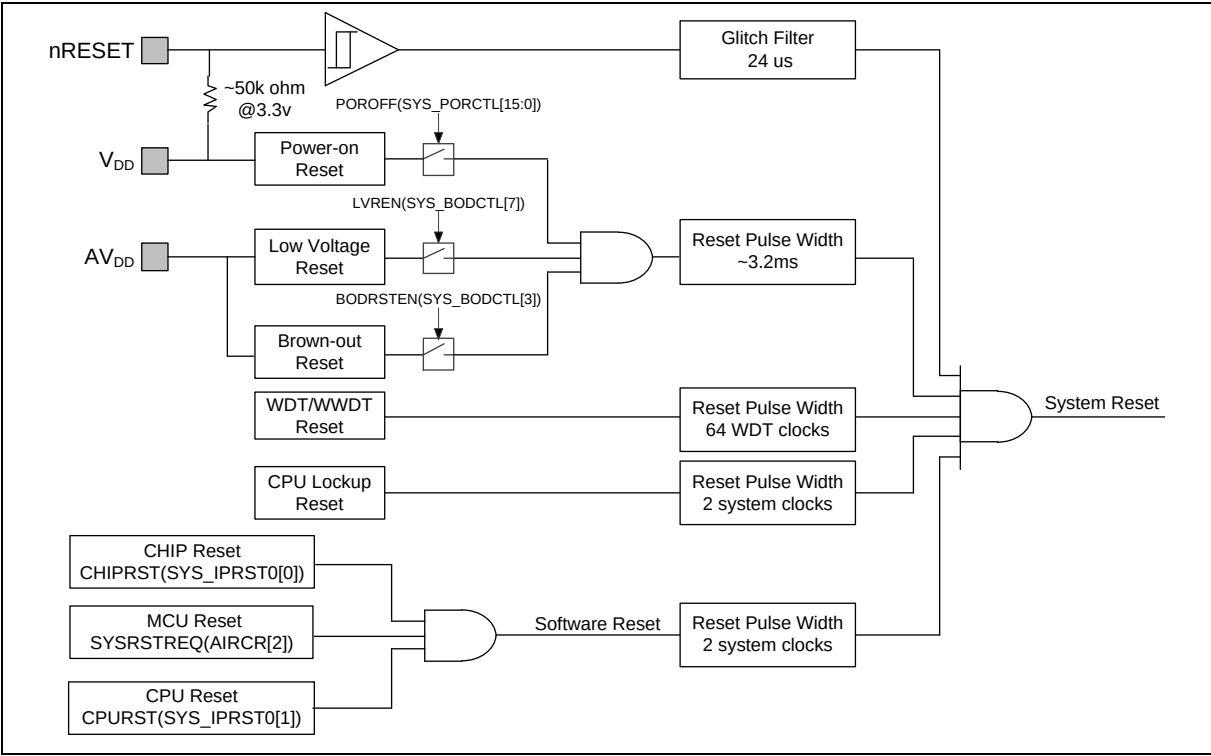


Figure 6.2-1 System Reset Sources

There are a total of 9 reset sources in the NuMicro® family. In general, CPU reset is used to reset Cortex®-M23 only; the other reset sources will reset Cortex®-M23 and all peripherals. However, there are small differences between each reset source and they are listed in Table 6.2-1.

Reset Sources Register	POR	NRESET	WDT	LVR	BOD	Lockup	CHIP	MCU	CPU
SYS_RSTSTS	Bit 0 = 1	Bit 1 = 1	Bit 2 = 1	Bit 3 = 1	Bit 4 = 1	Bit 8 = 1	Bit 0 = 1	Bit 5 = 1	Bit 7 = 1
CHIPRST (SYS_IPRST0[0])	0x0	-	-	-	-	-	-	-	-
BODEN (SYS_BODCTL[0])	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	-	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	-
BODVL (SYS_BODCTL[2:1])									
BODRSTEN (SYS_BODCTL[3])									
HXTEN (CLK_PWRCTL[0])	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	
LXTEN (CLK_PWRCTL[1])	0x0	-	-	-	-	-	-	-	-
WDTCKEN (CLK_APBCLK0[0])	0x1	-	0x1	-	-	-	0x1	-	-
HCLKSEL	Reload from	Reload from	Reload from	Reload from	Reload from	Reload from	Reload from	Reload from	-

(CLK_CLKSEL0[2:0])	CONFIG0	CONFIG0	CONFIG0	CONFIG0	CONFIG0	CONFIG0	CONFIG0	CONFIG0	
WDTSEL (CLK_CLKSEL1[1:0])	0x3	0x3	-	-	-	-	-	-	-
HXTSTB (CLK_STATUS[0])	0x0	-	-	-	-	-	-	-	-
LXTSTB (CLK_STATUS[1])	0x0	-	-	-	-	-	-	-	-
PLLSTB (CLK_STATUS[2])	0x0	-	-	-	-	-	-	-	-
HIRCSTB (CLK_STATUS[4])	0x0	-	-	-	-	-	-	-	-
CLKSFAIL (CLK_STATUS[7])	0x0	0x0	-	-	-	-	-	-	-
RSTEN (WDT_CTL[1])	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	-	Reload from CONFIG0	-	-
WDTEN (WDT_CTL[7])									
WDT_CTL except bit 1 and bit 7.	0x0800	0x0800	0x0800	0x0800	0x0800	-	0x0800	-	-
WDT_ALTCTL	0x0000	0x0000	0x0000	0x0000	0x0000	-	0x0000	-	-
WWDT_RLDCNT	0x0000	0x0000	0x0000	0x0000	0x0000	-	0x0000	-	-
WWDT_CTL	0x3F0800	0x3F0800	0x3F0800	0x3F0800	0x3F0800	-	0x3F0800	-	-
WWDT_STATUS	0x0000	0x0000	0x0000	0x0000	0x0000	-	0x0000	-	-
WWDT_CNT	0x3F	0x3F	0x3F	0x3F	0x3F	-	0x3F	-	-
BS (FMC_ISPCTL[1])	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	-	Reload from CONFIG0	-	-
BL (FMC_ISPCTL[16])									
FMC_DFBA	Reload from CONFIG1	Reload from CONFIG1	Reload from CONFIG1	Reload from CONFIG1	Reload from CONFIG1	-	Reload from CONFIG1	-	-
CBS (FMC_ISPSTS[2:1])	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	-	Reload from CONFIG0	-	-
VECMAP (FMC_ISPSTS[23:9])	Reload base on CONFIG0	Reload base on CONFIG0	Reload base on CONFIG0	Reload base on CONFIG0	Reload base on CONFIG0	-	Reload base on CONFIG0	-	-
Other Peripheral Registers	Reset Value								
FMC Registers	Reset Value								
Note: '-' means that the value of register keeps original setting.									

Table 6.2-1 Reset Value of Registers

6.2.2.1 nRESET Reset

The nRESET reset means to generate a reset signal by pulling low nRESET pin, which is an asynchronous reset input pin and can be used to reset system at any time. When the nRESET voltage is lower than $0.2 V_{DD}$ and the state keeps longer than 24 μs (glitch filter), chip will be reset. The nRESET reset will control the chip in reset state until the nRESET voltage rises above $0.7 V_{DD}$ and the state keeps longer than 24 μs (glitch filter). The PINRF(SYS_RSTSTS[1]) will be set to 1 if the previous reset source is nRESET reset. Table 6.2-2 shows the nRESET reset waveform.

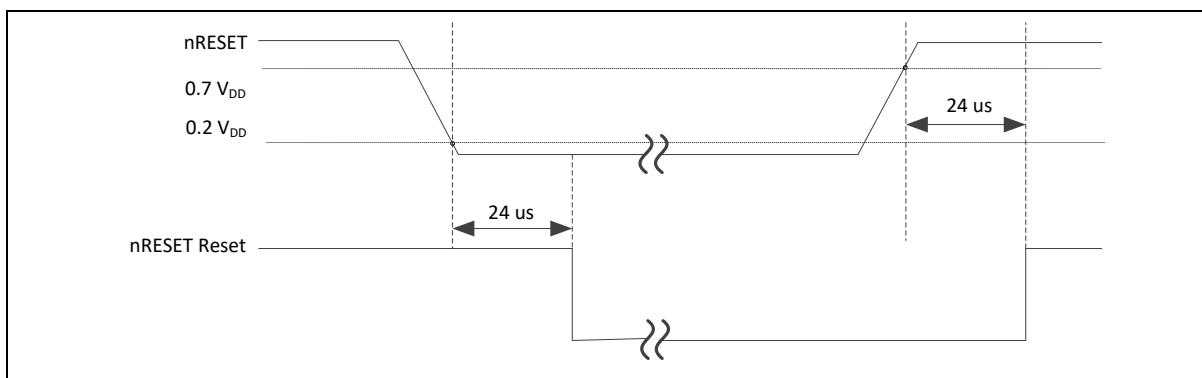


Figure 6.2-2 nRESET Reset Waveform

6.2.2.2 Power-on Reset (POR)

The Power-on reset (POR) is used to generate a stable system reset signal and forces the system to be reset when power-on to avoid unexpected behavior of MCU. When applying the power to MCU, the POR module will detect the rising voltage and generate reset signal to system until the voltage is ready for MCU operation. At POR reset, the PORF(SYS_RSTSTS[0]) will be set to 1 to indicate there is a POR reset event. The PORF(SYS_RSTSTS[0]) bit can be cleared by writing 1 to it. Figure 6.2-3 shows the power-on reset waveform.

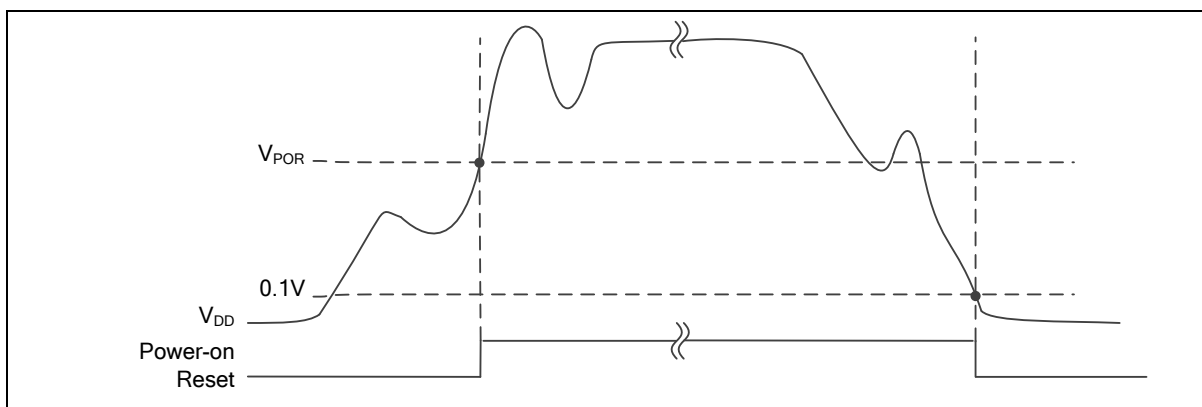


Figure 6.2-3 Power-on Reset (POR) Waveform

6.2.2.3 Low Voltage Reset (LVR)

If the Low Voltage Reset function is enabled by setting the Low Voltage Reset Enable Bit LVREN (SYS_BODCTL[7]) to 1, after 200 μs delay, LVR detection circuit will be stable and the LVR function will be active. Then LVR function will detect AV_{DD} during system operation. When the AV_{DD} voltage is lower than V_{LVR} and the state keeps longer than De-glitch time set by LVRDGSEL (SYS_BODCTL[14:12]), chip will be reset. The LVR reset will control the chip in reset state until the AV_{DD} voltage rises above V_{LVR} and the state keeps longer than De-glitch time set by LVRDGSEL (SYS_BODCTL[14:12]). The default setting of Low Voltage Reset is enabled without De-glitch function. Figure 6.2-4 shows the Low Voltage Reset waveform.

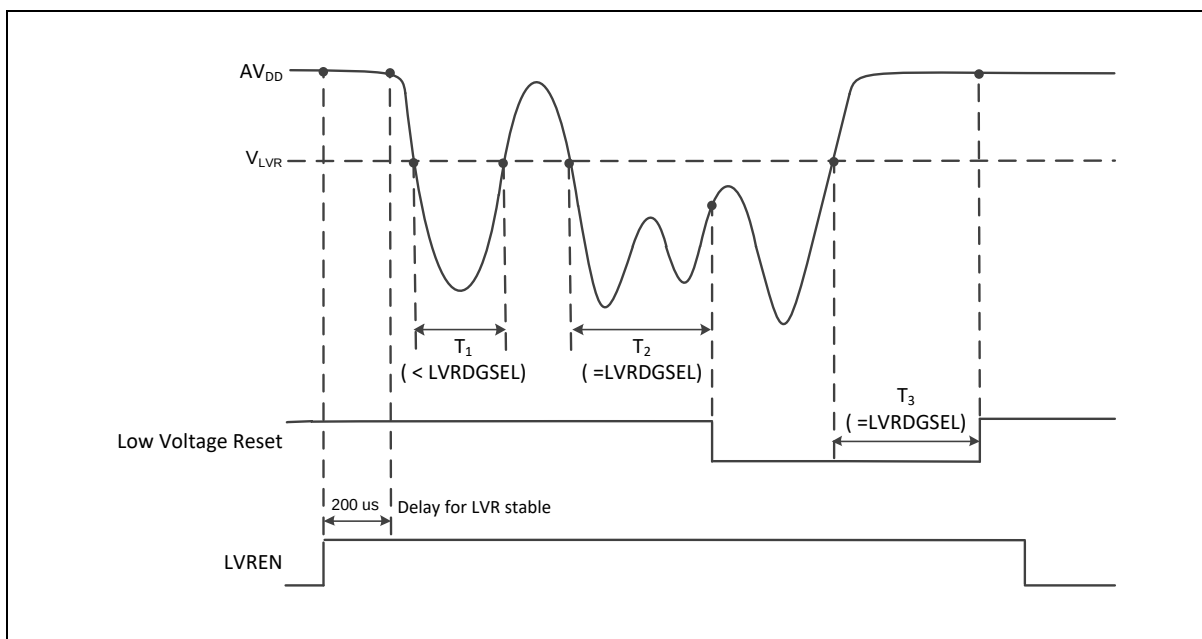


Figure 6.2-4 Low Voltage Reset (LVR) Waveform

6.2.2.4 Brown-out Detector Reset (BOD Reset)

If the Brown-out Detector (BOD) function is enabled by setting the Brown-out Detector Enable Bit BODEN (SYS_BODCTL[0]), Brown-out Detector function will detect AV_{DD} during system operation. When the AV_{DD} voltage is lower than V_{BOD} which is decided by BODEN and BODVL (SYS_BODCTL[18:16]) and the state keeps longer than De-glitch time set by BODDGSEL (SYS_BODCTL[10:8]), chip will be reset. The BOD reset will control the chip in reset state until the AV_{DD} voltage rises above V_{BOD} and the state keeps longer than De-glitch time set by BODDGSEL. The default value of BODEN, BODVL and BODRSTEN (SYS_BODCTL[3]) is set by Flash controller user configuration register CBODEN (CONFIG0 [19]), CBOV (CONFIG0 [23:21]) and CBORST(CONFIG0[20]) respectively. User can determine the initial BOD setting by setting the CONFIG0 register. Figure 6.2-5 shows the Brown-out Detector waveform.

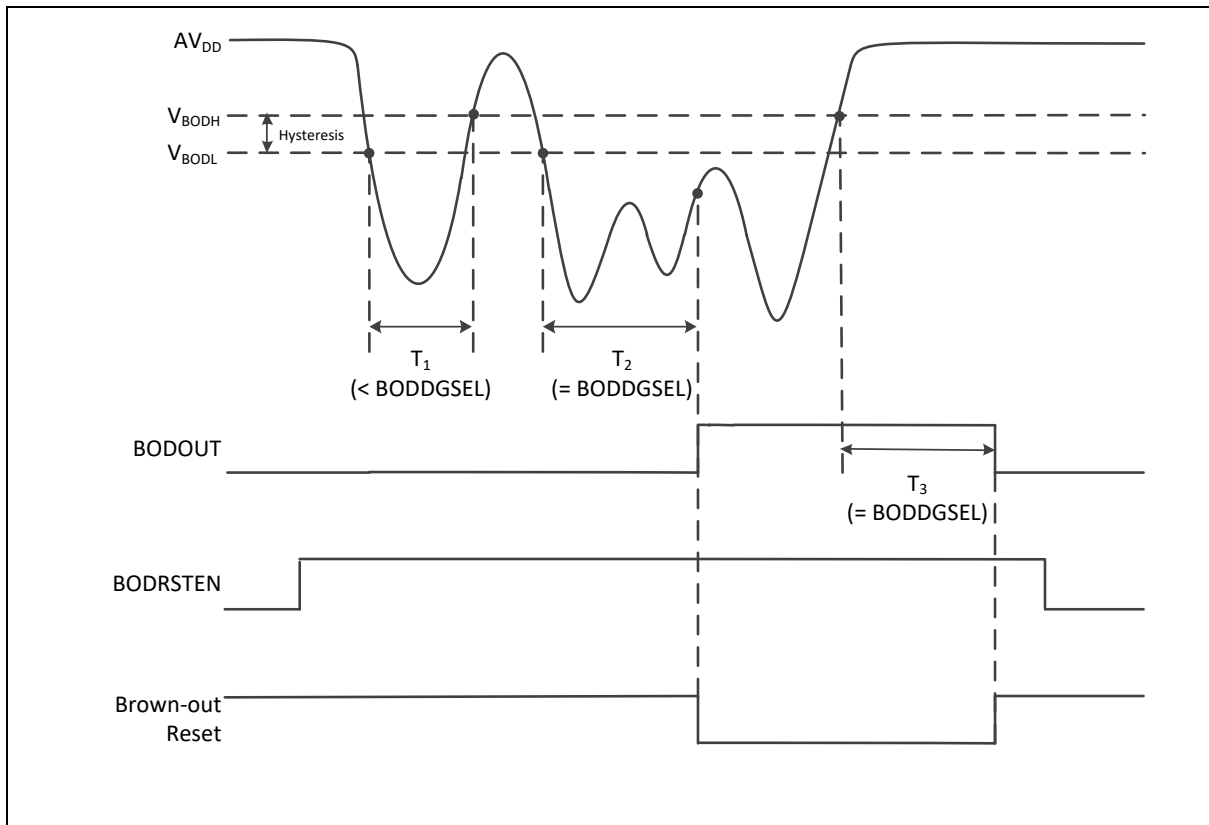


Figure 6.2-5 Brown-out Detector (BOD) Waveform

6.2.2.5 Watchdog Timer Reset (WDT)

In most industrial applications, system reliability is very important. To automatically recover the MCU from failure status is one way to improve system reliability. The watchdog timer (WDT) is widely used to check if the system works fine. If the MCU is crashed or out of control, it may cause the watchdog time-out. User may decide to enable system reset during watchdog time-out to recover the system and take action for the system crash/out-of-control after reset.

Software can check if the reset is caused by watchdog time-out to indicate the previous reset is a watchdog reset and handle the failure of MCU after watchdog time-out reset by checking `WDTRF(SYS_RSTSTS[2])`.

6.2.2.6 CPU Lockup Reset

CPU enters lockup status after CPU produces hardfault at hardfault handler and chip gives immediate indication of seriously errant kernel software. This is the result of the CPU being locked because of an unrecoverable exception following the activation of the processor's built in system state protection hardware. When chip enters debug mode, the CPU lockup reset will be ignored.

6.2.2.7 CPU Reset, CHIP Reset and MCU Reset

The CPU Reset means only Cortex®-M23 core is reset and all other peripherals remain the same status after CPU reset. User can set the `CPURST(SYS_IPRST0[1])` to 1 to assert the CPU Reset signal.

The CHIP Reset is same with Power-on Reset. The CPU and all peripherals are reset and `BS(FMC_ISPCTL[1])` bit is automatically reloaded from `CONFIG0` setting. User can set the `CHIPRST(SYS_IPRST0[1])` to 1 to assert the CHIP Reset signal.

The MCU Reset is similar with CHIP Reset. The difference is that `BS(FMC_ISPCTL[1])` will not be reloaded from `CONFIG0` setting and keep its original software setting for booting from `APROM` or `LDROM`. User can set the `SYSRESETREQ(AIRCR[2])` to 1 to assert the MCU Reset.

6.2.3 System Power Distribution

In this chip, power distribution is divided into four segments:

- Analog power from AV_{DD} and AV_{SS} provides the power for analog components operation.
- Digital power from V_{DD} and V_{SS} supplies the power to the internal regulator which provides a fixed 1.5V power for digital operation and I/O pins.
- USB transceiver power from V_{BUS} offers the power for operating the USB transceiver.
- RTC power from regulator uninterrupted power domain provides, the power for RTC and 20 bytes backup registers.

Analog power (AV_{DD}) should be the same voltage level of the digital power (V_{DD}). Figure 6.2-6 shows the power distribution of the M253 series.

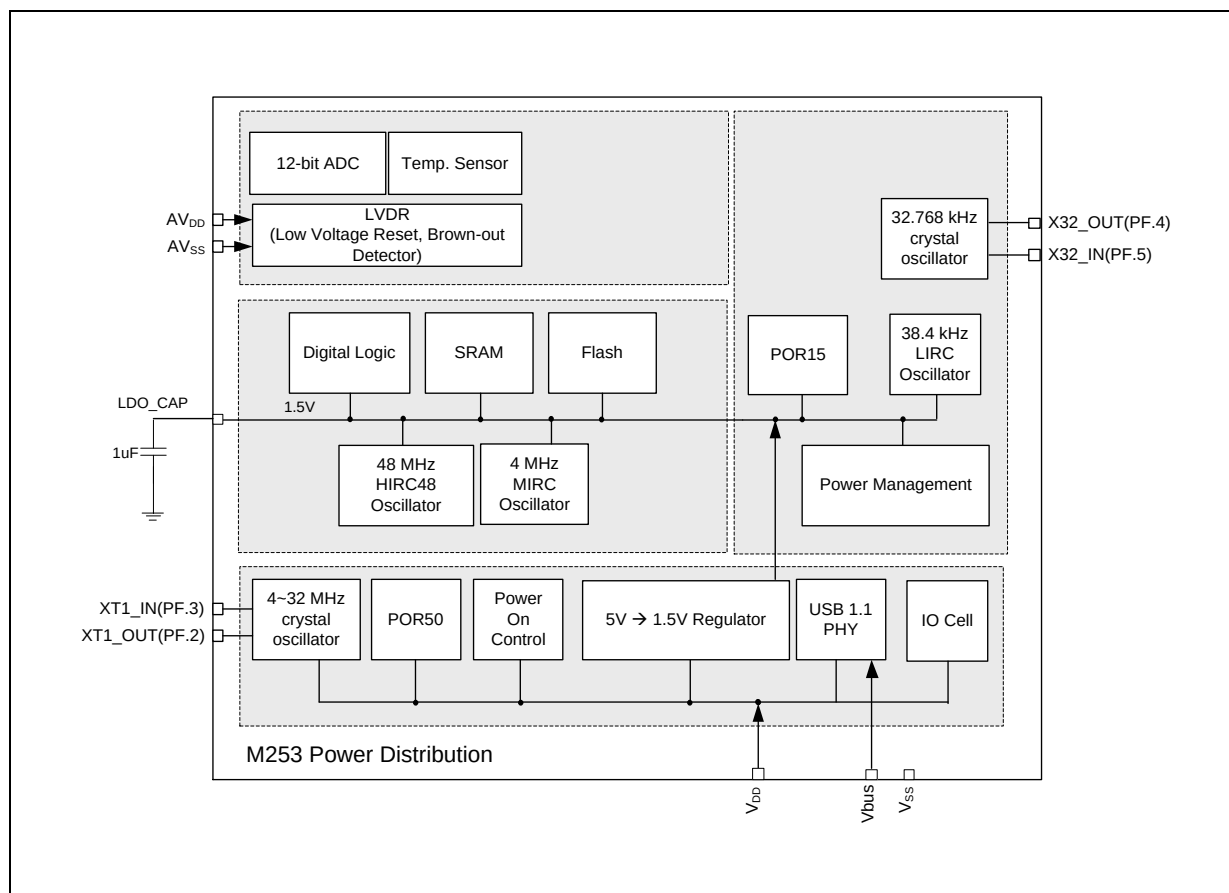


Figure 6.2-6 NuMicro® M253 Series Power Distribution Diagram

6.2.4 Power Modes and Wake-up Sources

The M253 series has a power manager unit to support several operating modes for saving power. Table 6.2-2 lists all power modes in the M253 series.

Mode	CPU Operating Maximum Speed (MHz)	LDO_CAP(V)	Clock Disable
Normal mode	48	1.5	All clocks are disabled by control register.
Idle mode	CPU enters Sleep mode	1.5	Only CPU clock is disabled.
Power-down mode	CPU enters Deep Sleep mode	1.5	Most clocks are disabled except LIRC/LXT, and only RTC/WDT/Timer/UART peripheral clocks still enable if their clock sources are selected as LIRC/LXT.
Fast wake up Power-down mode (FWPD)	CPU enters Sleep mode	1.5	Most clocks are disabled except LIRC/LXT, and only RTC/WDT/Timer/UART peripheral clocks still enable if their clock sources are selected as LIRC/LXT.

Table 6.2-2 Power Mode Table

There are different power mode entry settings. Each power mode has different entry setting and leaving condition. Table 6.2-3 shows the entry setting for each power mode. When chip power-on, chip is running ar normal mode. User can enter each mode by setting SLEEPDEEP (SCR[2]), PDEN (CLK_PWRCTL[7]) and PDMSEL (CLK_PMUCTL[2:0]) and execute WFI instruction.

Register/Instruction Mode	SLEEPDEEP (SCR[2])	PDEN (CLK_PWRCTL[7])	PDMSEL (CLK_PMUCTL[2:0])	CPU Run WFI Instruction
Normal mode	0	0	0	NO
Idle mode (CPU enters Sleep mode)	0	0	0	YES
Power-down mode (CPU enters Deep Sleep mode)	1	1	0	YES
Fast wake up Power-down mode (FWPD)	1	1	2	YES

Table 6.2-3 Power Mode Difference Table

There are several wake-up sources in Idle mode and Power-down mode. Table 6.2-4 lists the available clocks for each power mode.

Power Mode	Normal Mode	Idle Mode	Power-Down Mode
Definition	CPU is in active state	CPU is in sleep state	CPU is in sleep state and all clocks stop except LXT and LIRC. SRAM content retended.
Entry Condition	Chip is in normal mode after system reset released	CPU executes WFI instruction.	CPU sets sleep mode enable and power down enable and executes WFI instruction.
Wake-up Sources	N/A	All interrupts	RTC, WDT, I ² C, Timer, UART, BOD, GPIO, EINT, USCI and USB
Available Clocks	All	All except CPU clock	LXT and LIRC
After Wake-up	N/A	CPU back to normal mode	CPU back to normal mode

Table 6.2-4 Power Mode Difference Table

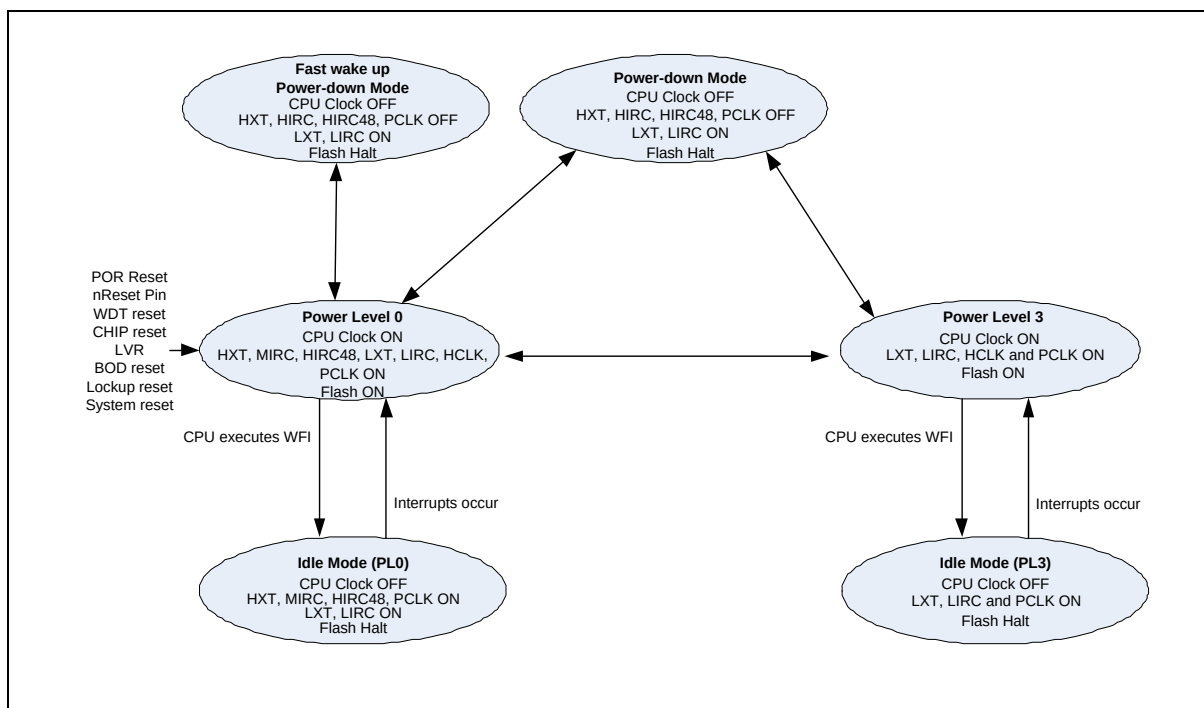


Figure 6.2-7 Power Mode State Machine

1. LXT (32768 Hz XTL) ON or OFF depends on SW setting in normal mode.
2. LIRC (38.4 kHz OSC) ON or OFF depends on SOFTWARE setting in normal mode.
3. If TIMER clock source is selected as LIRC/LXT and LIRC/LXT is on.
4. If WDT clock source is selected as LIRC and LIRC is on.
5. If RTC clock source is selected as LXT and LXT is on.
6. If UART clock source is selected as LXT and LXT is on.

	Normal Mode	Idle Mode	Power-Down Mode PD
HXT (4~32 MHz XTL)	ON	ON	Halt
MIRC (4 MHz OSC)	ON	ON	Halt
HIRC48 (48 MHz OSC)	ON	ON	Halt
LXT (32768 Hz XTL)	ON	ON	ON/OFF ¹
LIRC (38.4 kHz OSC)	ON	ON	ON/OFF ²
LDO	ON	ON	ON
CPU	ON	Halt	Halt
HCLK/PCLK	ON	ON	Halt
SRAM retention	ON	ON	ON
FLASH	ON	ON	Halt
GPIO	ON	ON	Halt
PDMA	ON	ON	Halt
TIMER	ON	ON	ON/OFF ³
WDT	ON	ON	ON/OFF ⁴
WWDT	ON	ON	Halt
RTC	ON	ON	ON/OFF ⁵
UART	ON	ON	ON/OFF ⁶
USCI	ON	ON	Halt
I ² C	ON	ON	Halt
SPI	ON	ON	Halt
USBD	ON	ON	Halt
ADC	ON	ON	Halt

Table 6.2-5 Clocks in Power Modes

Wake-up sources in Power-down mode:

RTC, WDT, I²C, Timer, UART, USCI, BOD, GPIO, and USB.

After chip enters power down, the following wake-up sources can wake chip up to normal mode. Table 6.2-5 lists the condition about how to enter Power-down mode again for each peripheral.

*User needs to wait this condition before setting PDEN(CLK_PWRCTL[7]) and execute WFI to enter Power-down mode.

Wake-Up Source	Wake-Up Condition	Power-Down Mode	System Can Enter Power-Down Mode Again Condition*
		PD FWKPD	
BOD	Brown-Out Detector Interrupt	Y	After software writes 1 to clear BODIF (SYS_BODCTL[4]).
LVR	LVR Reset	Y	After software writes 1 to clear LVRF (SYS_RSTSTS[3]).
INT	External Interrupt	Y	After software write 1 to clear the Px_INTSRC[n] bit.
GPIO	GPIO Interrupt	Y	After software write 1 to clear the Px_INTSRC[n] bit.
TIMER	Timer Interrupt	Y	After software writes 1 to clear TWKF (TIMERx_INTSTS[1]) and TIF (TIMERx_INTSTS[0]).
WDT	WDT Interrupt	Y	After software writes 1 to clear WKF (WDT_CTL[5]) (Write Protect).
RTC	Alarm Interrupt	Y	After software writes 1 to clear ALMIF (RTC_INTSTS[0]).
	Time Tick Interrupt	Y	After software writes 1 to clear TICKIF (RTC_INTSTS[1]).
UART	nCTS wake-up	Y	After software writes 1 to clear CTSWKF (UARTx_WKSTS[0]).
	Incoming Data wake-up	Y	After software writes 1 to clear DATWKF (UARTx_WKSTS[1]).
	Received FIFO Threshold Wake-up	Y	After software writes 1 to clear RFRTWKF (UARTx_WKSTS[2]).
	RS-485 AAD Mode Wake-up	Y	After software writes 1 to clear RS485WKF (UARTx_WKSTS[3]).
	Received FIFO Threshold Time-out Wake-up	Y	After software writes 1 to clear TOUTWKF (UARTx_WKSTS[4]).
USCI UART	CTS Toggle	Y	After software writes 1 to clear WKF (UUART_WKSTS[0]).
	Data Toggle	Y	After software writes 1 to clear WKF (UUART_WKSTS[0]).
USCI I ² C	Data toggle	Y	After software writes 1 to clear WKF (UI2C_WKSTS[0]).
	Address match	Y	After software writes 1 to clear WKAKDONE (UI2C_PROTSTS[16], and then writes 1 to clear WKF (UI2C_WKSTS[0]).
USCI SPI	SS Toggle	Y	After software writes 1 to clear WKF (USPI_WKSTS[0]).
I ² C	Address match wake-up	Y	After software writes 1 to clear WKAKDONE (I2C_WKSTS[1]). Then software writes 1 to clear WKIF(I2C_WKSTS[0]).
USBD	Remote Wake-up	Y	After software writes 1 to clear BUSIF (USBD_INTSTS[0]).

Table 6.2-6 Condition of Entering Power-down Mode Again

6.2.5 Chip Bus Matrix

The M253 series provides 4G-byte addressing space. The memory locations assigned to each on-chip controllers are shown in Table 6.2-7. The detailed register definition, memory space, and programming will be described in the following sections for each on-chip peripheral. The M253 series only supports little-endian data format.

6.2.6 System Memory Map

The M253 series provides 4G-byte addressing space. The memory locations assigned to each on-chip controllers are shown in Table 6.2-7. The detailed register definition, memory space, and programming will be described in the following sections for each on-chip peripheral. The M253 series only supports little-endian data format.

Address Space	Token	Controllers
Flash and SRAM Memory Space		
0x0000_0000 – 0x0001_FFFF	FLASH_BA	FLASH Memory Space (128 Kbytes)
0x2000_0000 – 0x2000_3FFF	SRAM0_BA	SRAM Memory Space (16 Kbytes)
0x6000_0000 – 0x6FFF_FFFF	EXTMEM_BA	External Memory Space (256 Mbytes)
Peripheral Controllers Space (0x4000_0000 – 0x400F_FFFF)		
0x4000_0000 – 0x4000_01FF	SYS_BA	System Control Registers
0x4000_0200 – 0x4000_02FF	CLK_BA	Clock Control Registers
0x4000_0300 – 0x4000_03FF	NMI_BA	NMI Control Registers
0x4000_4000 – 0x4000_4FFF	GPIO_BA	GPIO Control Registers
0x4000_8000 – 0x4000_8FFF	PDMA_BA	Peripheral DMA Control Registers
0x4000_C000 – 0x4000_CFFF	FMC_BA	Flash Memory Control Registers
0x4003_1000 – 0x4003_1FFF	CRC_BA	CRC Generator Registers
APB Controllers Space (0x4000_0000 ~ 0x400F_FFFF)		
0x4004_0000 – 0x4004_0FFF	WDT_BA	Watchdog Timer Control Registers
0x4004_1000 – 0x4004_1FFF	RTC_BA	Real Time Clock (RTC) Control Register
0x4004_3000 – 0x4004_3FFF	EADC_BA	Enhanced Analog-Digital-Converter (EADC) Control Registers
0x4005_0000 – 0x4005_0FFF	TMR01_BA	Timer0/Timer1 Control Registers
0x4005_1000 – 0x4005_1FFF	TMR23_BA	Timer2/Timer3 Control Registers
0x4005_A000 – 0x4005_AFFF	BPWM0_BA	BPWM0 Control Registers
0x4005_B000 – 0x4005_BFFF	BPWM1_BA	BPWM1 Control Registers
0x4006_0000 – 0x4006_1FFF	SPI0_BA	SPI0 Control Registers
0x4007_0000 – 0x4007_0FFF	UART0_BA	UART0 Control Registers
0x4007_1000 – 0x4007_1FFF	UART1_BA	UART1 Control Registers
0x4007_2000 – 0x4007_2FFF	UART2_BA	UART2 Control Registers
0x4007_3000 – 0x4007_3FFF	UART3_BA	UART3 Control Registers
0x4007_4000 – 0x4007_4FFF	UART4_BA	UART4 Control Registers
0x4008_0000 – 0x4008_0FFF	I2C0_BA	I2C0 Control Registers
0x4008_1000 – 0x4008_1FFF	I2C1_BA	I2C1 Control Registers
0x400A_0000 – 0x400A_0FFF	CANFD0_BA	CANFD0 Control Register
0x400C_0000 – 0x400C_0FFF	USBD_BA	USB Device Control Register
0x400D_0000 – 0x400D_0FFF	USCI0_BA	USCI0 Control Registers

System Controllers Space (0xE000_E000 ~ 0xE000_EFFF)		
0xE000_E010 – 0xE000_E0FF	SCS_BA	System Timer Control Registers
0xE000_E100 – 0xE000_ECFF	SCS_BA	External Interrupt Controller Control Registers
0xE000_ED00 – 0xE000_ED8F	SCS_BA	System Control Registers

Table 6.2-7 Address Space Assignments for On-Chip Controllers

6.2.7 SRAM Memory Organization

The M253 series supports embedded SRAM with up to 16 Kbytes size.

- Supports up to 16 Kbytes SRAM
- Supports byte /half word /word write
- Supports oversize response error

Table 6.2-9 shows the M253 series SRAM organization. The address between 0x2000_4000 to 0x3FFF_FFFF is illegal memory space and chip will enter hardfault if CPU accesses these illegal memory addresses.

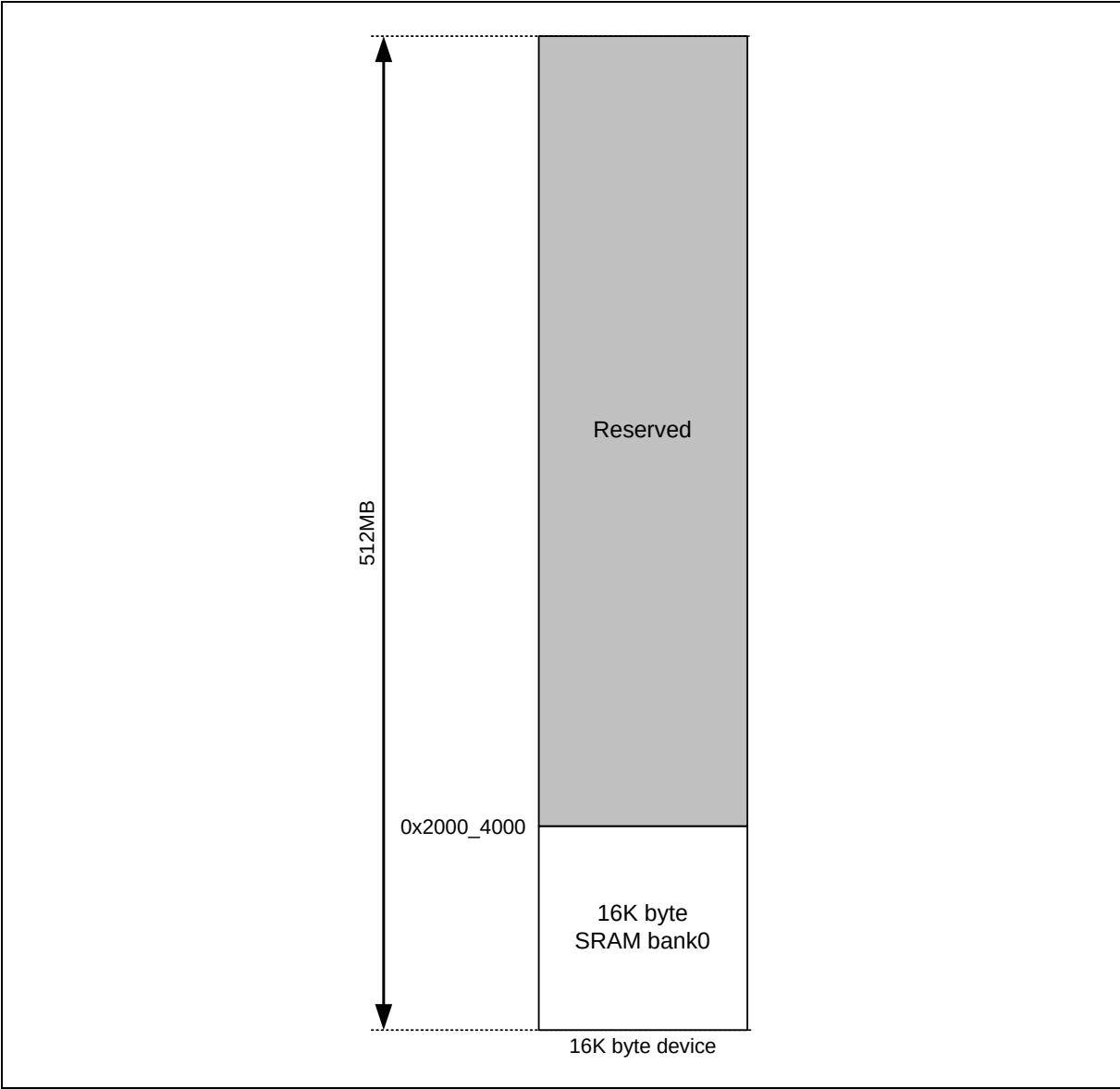


Figure 6.2-8 SRAM Memory Organization

6.2.8 IRC Auto Trim

This chip supports auto-trim function: the HIRC trim (48 MHz RC oscillator) and MIRC trim (4.032 RC oscillator), according to the accurate LXT (32.768 kHz crystal oscillator) or internal USB synchronous mode, automatically gets accurate output frequency, 0.25 % deviation within all temperature ranges.

For instance, the system needs an accurate 4.032 MHz clock. In such case, if neither uses PLL as the system clock source nor solders 32.768 kHz crystal in system, user has to set REFCKSEL (SYS_MIRCTIMCTL[10] reference clock selection) to "1", set FREQSEL (SYS_MIRCTIMCTL[1:0] trim frequency selection) to "10", and the auto-trim function will be enabled. Interrupt status bit FREQLOCK (SYS_MIRCTIMSTS[0] MIRC frequency lock status) "1" indicates the MIRC output frequency is accurate within 0.25% deviation.

In HIRC case, the system needs an accurate 48 MHz clock. In such case, if neither uses PLL as the system clock source nor solders 32.768 kHz crystal in system, user has to set REFCKSEL (SYS_HIRCTIMCTL[10] reference clock selection) to "1", set FREQSEL (SYS_HIRCTIMCTL[1:0] trim frequency selection) to "10", and the auto-trim function will be enabled. Interrupt status bit FREQLOCK (SYS_HIRCTIMSTS[8] HIRC frequency lock status) "1" indicates the HIRC output frequency is accurate within 0.25% deviation.

HIRC trim and MIRC trim can only work properly when the clock sources are stable. When the RC clock or the reference clock is not stable or the system goes into power down, HIRC trim and MIRC trim need to wait until the clock is stable or system wakes up, and then it can be enabled or will get a clock error flag.

6.2.9 Register Map

R: read only, W: write only, R/W: both read and write

Register	Offset	R/W	Description	Reset Value
SYS Base Address: SYS_BA = 0x4000_0000				
SYS_PDID	SYS_BA+0x00	R	Part Device Identification Number Register	0xFFFF_XXXX
SYS_RSTSTS	SYS_BA+0x04	R/W	System Reset Status Register	0x0000_0043
SYS_IPRST0	SYS_BA+0x08	R/W	Peripheral Reset Control Register 0	0x0000_0000
SYS_IPRST1	SYS_BA+0x0C	R/W	Peripheral Reset Control Register 1	0x0000_0000
SYS_IPRST2	SYS_BA+0x10	R/W	Peripheral Reset Control Register 2	0x0000_0000
SYS_ALTCTL	SYS_BA+0x14	R/W	Miscellaneous Control Register	0x0000_0000
SYS_BODCTL	SYS_BA+0x18	R/W	Brown-out Detector Control Register	0x000X_038X
SYS_IVSCTL	SYS_BA+0x1C	R/W	Internal Voltage Source Control Register	0x0000_0000
SYS_PORCTL0	SYS_BA+0x24	R/W	Power-On-reset Controller Register 0	0x0000_0000
SYS_GPA_MFPL	SYS_BA+0x30	R/W	GPIOA Low Byte Multiple Function Control Register	0x0000_0000
SYS_GPA_MFPH	SYS_BA+0x34	R/W	GPIOA High Byte Multiple Function Control Register	0x0000_0000
SYS_GPB_MFPL	SYS_BA+0x38	R/W	GPIOB Low Byte Multiple Function Control Register	0x0000_0000
SYS_GPB_MFPH	SYS_BA+0x3C	R/W	GPIOB High Byte Multiple Function Control Register	0x0000_0000
SYS_GPC_MFPL	SYS_BA+0x40	R/W	GPIOC Low Byte Multiple Function Control Register	0x0000_0000
SYS_GPC_MFPH	SYS_BA+0x44	R/W	GPIOC High Byte Multiple Function Control Register	0x0000_0000
SYS_GPF_MFPL	SYS_BA+0x58	R/W	GPIOF Low Byte Multiple Function Control Register	0x0000_00ee
SYS_GPF_MFPH	SYS_BA+0x5C	R/W	GPIOF High Byte Multiple Function Control Register	0x0000_0000
SYS_GPA_MFOS	SYS_BA+0x80	R/W	GPIOA Multiple Function Output Select Register	0x0000_0000
SYS_GPB_MFOS	SYS_BA+0x84	R/W	GPIOB Multiple Function Output Select Register	0x0000_0000
SYS_GPC_MFOS	SYS_BA+0x88	R/W	GPIOC Multiple Function Output Select Register	0x0000_0000
SYS_GPF_MFOS	SYS_BA+0x94	R/W	GPIOF Multiple Function Output Select Register	0x0000_0000
SYS_MODCTL	SYS_BA+0xC0	R/W	Modulation Control Register	0x0000_0000
SYS_SRAM_BISTCTL	SYS_BA+0xD0	R/W	System SRAM BIST Test Control Register	0x0000_0000
SYS_SRAM_BISTSTS	SYS_BA+0xD4	R	System SRAM BIST Test Status Register	0x00xx_00xx
SYS_HIRCTRIMCTL	SYS_BA+0xF0	R/W	HIRC Trim Control Register	0x0008_0000
SYS_HIRCTRIMIEN	SYS_BA+0xF4	R/W	HIRC Trim Interrupt Enable Register	0x0000_0000
SYS_HIRCTRIMSTS	SYS_BA+0xF8	R/W	HIRC Trim Interrupt Status Register	0x0000_0000

SYS_REGLCTL	SYS_BA+0x100	R/W	Register Lock Control Register	0x0000_0000
SYS_MIRCTIMCTL	SYS_BA+0x104	R/W	MIRC Trim Control Register	0x0008_0010
SYS_MIRCTIMEN	SYS_BA+0x108	R/W	MIRC Trim Interrupt Enable Register	0x0000_0000
SYS_MIRCTIMSTS	SYS_BA+0x10C	R/W	MIRC Trim Interrupt Status Register	0x0000_0000
SYS_PORCTL1	SYS_BA+0x1EC	R/W	Power-On-reset Controller Register 1	0x0000_0000
SYS_PLCTL	SYS_BA+0x1F8	R/W	Power Level Control Register	0x0000_0000
SYS_PLSTS	SYS_BA+0x1FC	R/W	Power Level Status Register	0x0000_000X

6.2.10 Register Description

Part Device Identification Number Register (SYS_PDID)

Register	Offset	R/W	Description	Reset Value
SYS_PDID	SYS_BA+0x00	R	Part Device Identification Number Register	0XXXXX_XXXX

[1] Every part number has a unique default reset value.

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

Bits	Description
[31:0]	PDID Part Device Identification Number (Read Only) This register reflects device part number code. Software can read this register to identify which device is used.

System Reset Status Register (SYS_RSTSTS)

This register provides specific information for software to identify this chip's reset source from last operation.

Register	Offset	R/W	Description	Reset Value
SYS_RSTSTS	SYS_BA+0x04	R/W	System Reset Status Register	0x0000_0043

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							CPULKRF
7	6	5	4	3	2	1	0
CPURF	PMURF	SYSRF	BODRF	LVRF	WDTRF	PINRF	PORF

Bits	Description
[31:9]	Reserved Reserved.
[8]	CPULKRF CPU Lockup Reset Flag 0 = No reset from CPU lockup happened. 1 = The Cortex®-M23 lockup happened and chip is reset. Note 1: Write 1 to clear this bit to 0. Note 2: When CPU lockup happened under ICE is connected, This flag will set to 1 but chip will not reset.
[7]	CPURF CPU Reset Flag The CPU reset flag is set by hardware if software writes CPURST (SYS_IPRST0[1]) 1 to reset Cortex®- M23 core and Flash Memory Controller (FMC). 0 = No reset from CPU. 1 = The Cortex®-M23 Core and FMC are reset by software setting CPURST to 1. Note: Write 1 to clear this bit to 0.
[6]	PMURF PMU Reset Flag, The PMU reset flag is set by any reset signal when MCU is in power down state. 0 = No reset in power down state. 1 = Any reset signal happens in power down state. Note: Write 1 to clear this bit to 0.
[5]	SYSRF System Reset Flag The system reset flag is set by the "Reset Signal" from the Cortex®M23Core to indicate the previous reset source. 0 = No reset from Cortex®-M23. 1 = The Cortex®- M23 had issued the reset signal to reset the system by writing 1 to the bit SYSRESETREQ(AIRCR[2], Application Interrupt and Reset Control Register, address = 0xE000ED0C) in system control registers of Cortex®-M23 core. Note: Write 1 to clear this bit to 0.

Bits	Description	
[4]	BODRF	BOD Reset Flag The BOD reset flag is set by the "Reset Signal" from the Brown-Out Detector to indicate the previous reset source. 0 = No reset from BOD. 1 = BOD had issued the reset signal to reset the system. Note: Write 1 to clear this bit to 0.
[3]	LVRF	LVR Reset Flag The LVR reset flag is set by the "Reset Signal" from the Low Voltage Reset Controller to indicate the previous reset source. 0 = No reset from LVR. 1 = LVR controller had issued the reset signal to reset the system. Note: Write 1 to clear this bit to 0.
[2]	WDTRF	WDT Reset Flag The WDT reset flag is set by the "Reset Signal" from the Watchdog Timer or Window Watchdog Timer to indicate the previous reset source. 0 = No reset from watchdog timer or window watchdog timer. 1 = The watchdog timer or window watchdog timer had issued the reset signal to reset the system. Note 1: Write 1 to clear this bit to 0. Note 2: Watchdog Timer register RSTF(WDT_CTL[2]) bit is set if the system has been reset by WDT time-out reset. Window Watchdog Timer register WWDRF(WWDT_STATUS[1]) bit is set if the system has been reset by WWDT time-out reset.
[1]	PINRF	nRESET Pin Reset Flag The nRESET pin reset flag is set by the "Reset Signal" from the nRESET Pin to indicate the previous reset source. 0 = No reset from nRESET pin. 1 = Pin nRESET had issued the reset signal to reset the system. Note: Write 1 to clear this bit to 0.
[0]	PORF	POR Reset Flag The POR reset flag is set by the "Reset Signal" from the Power-on Reset (POR) Controller or bit CHIPRST (SYS_IPRST0[0]) to indicate the previous reset source. 0 = No reset from POR or CHIPRST. 1 = Power-on Reset (POR) or CHIPRST had issued the reset signal to reset the system. Note: Write 1 to clear this bit to 0.

Peripheral Reset Control Register 0 (SYS_IPRST0)

Register	Offset	R/W	Description	Reset Value
SYS_IPRST0	SYS_BA+0x08	R/W	Peripheral Reset Control Register 0	0x0000_0000

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

Bits	Description
[31:8]	Reserved Reserved.
[7]	CRCRST CRC Calculation Controller Reset (Write Protect) Setting this bit to 1 will generate a reset signal to the CRC calculation controller. User needs to set this bit to 0 to release from the reset state. 0 = CRC calculation controller normal operation. 1 = CRC calculation controller reset. Note: This bit is write protected. Refer to the SYS_REGLCTL register.
[6:3]	Reserved Reserved.
[2]	PDMARST PDMA Controller Reset (Write Protect) Setting this bit to 1 will generate a reset signal to the PDMA. User needs to set this bit to 0 to release from reset state. 0 = PDMA controller normal operation. 1 = PDMA controller reset. Note: This bit is write protected. Refer to the SYS_REGLCTL register.
[1]	CPURST Processor Core One-shot Reset (Write Protect) Setting this bit will only reset the processor core and Flash Memory Controller(FMC), and this bit will automatically return to 0 after the 2 clock cycles. 0 = Processor core normal operation. 1 = Processor core one-shot reset. Note: This bit is write protected. Refer to the SYS_REGLCTL register.
[0]	CHIPRST Chip One-shot Reset (Write Protect) Setting this bit will reset the whole chip, including Processor core and all peripherals, and this bit will automatically return to 0 after the 2 clock cycles. The CHIPRST is same as the POR reset, all the chip controllers is reset and the chip setting from Flash are also reload. For the difference between CHIPRST and SYSRESETREQ(AIRCR[2]), please refer to section 6.2.2. 0 = Chip normal operation. 1 = Chip one-shot reset.

		Note 1: This bit is write protected. Refer to the SYS_REGLCTL register. Note 2: Reset by power on reset

Peripheral Reset Control Register 1 (SYS_IPRST1)

Setting these bits 1 will generate asynchronous reset signals to the corresponding module controller. Users need to set these bits to 0 to release corresponding module controller from reset state.

Register	Offset	R/W	Description	Reset Value
SYS_IPRST1	SYS_BA+0x0C	R/W	Peripheral Reset Control Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved			EADCRST	USBD RST	Reserved		CANFD0RST
23	22	21	20	19	18	17	16
Reserved			UART4RST	UART3RST	UART2RST	UART1RST	UART0RST
15	14	13	12	11	10	9	8
Reserved		SPI0RST	Reserved			I2C1RST	I2C0RST
7	6	5	4	3	2	1	0
Reserved		TMR3RST	TMR2RST	TMR1RST	TMR0RST	GPIORST	Reserved

Bits	Description
[31:29]	Reserved Reserved.
[28]	EADCRST EADC Controller Reset 0 = EADC controller normal operation. 1 = EADC controller reset.
[27]	USBD RST USBD Controller Reset 0 = USBD controller normal operation. 1 = USBD controller reset.
[26:25]	Reserved Reserved.
[24]	CANFD0RST CANFD0 Controller Reset 0 = CANFD0 controller normal operation. 1 = CANFD0 controller reset.
[23:21]	Reserved Reserved.
[20]	UART4RST UART4 Controller Reset 0 = UART4 controller normal operation. 1 = UART4 controller reset.
[19]	UART3RST UART3 Controller Reset 0 = UART3 controller normal operation. 1 = UART3 controller reset.
[18]	UART2RST UART2 Controller Reset 0 = UART2 controller normal operation. 1 = UART2 controller reset.
[17]	UART1RST UART1 Controller Reset 0 = UART1 controller normal operation.

		1 = UART1 controller reset.
[16]	UART0RST	UART0 Controller Reset 0 = UART0 controller normal operation. 1 = UART0 controller reset.
[15:14]	Reserved	Reserved.
[13]	SPI0RST	SPI0 Controller Reset 0 = SPI0 controller normal operation. 1 = SPI0 controller reset.
[12:10]	Reserved	Reserved.
[9]	I2C1RST	I2C1 Controller Reset 0 = I2C1 controller normal operation. 1 = I2C1 controller reset.
[8]	I2C0RST	I2C0 Controller Reset 0 = I2C0 controller normal operation. 1 = I2C0 controller reset.
[7:6]	Reserved	Reserved.
[5]	TMR3RST	Timer3 Controller Reset 0 = Timer3 controller normal operation. 1 = Timer3 controller reset.
[4]	TMR2RST	Timer2 Controller Reset 0 = Timer2 controller normal operation. 1 = Timer2 controller reset.
[3]	TMR1RST	Timer1 Controller Reset 0 = Timer1 controller normal operation. 1 = Timer1 controller reset.
[2]	TMR0RST	Timer0 Controller Reset 0 = Timer0 controller normal operation. 1 = Timer0 controller reset.
[1]	GPORST	GPIO Controller Reset 0 = GPIO controller normal operation. 1 = GPIO controller reset.
[0]	Reserved	Reserved.

Peripheral Reset Control Register 2 (SYS_IPRST2)

Setting these bits to 1 will generate asynchronous reset signals to the corresponding module controller. Users need to set these bits to 0 to release corresponding module controller from reset state.

Register	Offset	R/W	Description	Reset Value
SYS_IPRST2	SYS_BA+0x10	R/W	Peripheral Reset Control Register 2	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved					BPWM0RST	Reserved	
15	14	13	12	11	10	9	8
Reserved						USCI0RST	
7	6	5	4	3	2	1	0
Reserved							

Bits	Description	
[31:9]	Reserved	Reserved.
[18]	BPWM0RST	BPWM0 Controller Reset 0 = BPWM0 controller normal operation. 1 = BPWM0 controller reset.
[17:9]	Reserved	Reserved.
[8]	USCI0RST	USCI0 Controller Reset 0 = USCI0 controller normal operation. 1 = USCI0 controller reset.
[7:0]	Reserved	Reserved.

Miscellaneous Control Register (SYS_ALTCTL)

Register	Offset	R/W	Description	Reset Value
SYS_ALTCTL	SYS_BA+0x14	R/W	Miscellaneous Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
CANFD0CKSTP	CANFD0_PDEN	Reserved					

Bits	Description
[31:6]	Reserved
[7]	CANFD0CKSTP CANFD0 Clock Stop Acknowledgement (Read Only) 0 = CANFD0 clock didn't stop. 1 = CANFD0 clock stop.
[6]	CANFD0_PDEN CANFD0 Power Down Enable Bit 0 = CANFD0 Power-down mode Disabled. 1 = CANFD0 Power-down mode Enabled.
[5:0]	Reserved

Brown-out Detector Control Register (SYS_BODCTL)

Partial of the SYS_BODCTL control registers values are initiated by the Flash configuration and partial bits are write-protected bit.

Register	Offset	R/W	Description	Reset Value
SYS_BODCTL	SYS_BA+0x18	R/W	Brown-out Detector Control Register	0x000X_038X

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved					BODVL		
15	14	13	12	11	10	9	8
Reserved	LVRDGSEL			Reserved	BODDGSEL		
7	6	5	4	3	2	1	0
LVREN	BODOUT	BODLPM	BODIF	BODRSTEN	Reserved		BODEN

Bits	Description
[31:19]	Reserved Reserved.
[18:16]	BODVL Brown-out Detector Threshold Voltage Selection (Write Protect) The default value is set by Flash controller user configuration register CBOV (CONFIG0 [23:21]). 000 = Reserved. 001 = Brown-Out Detector threshold voltage is 1.8V. 010 = Brown-Out Detector threshold voltage is 2.0V. 011 = Brown-Out Detector threshold voltage is 2.4V. 100 = Brown-Out Detector threshold voltage is 2.7V. 101 = Brown-Out Detector threshold voltage is 3.0V. 110 = Brown-Out Detector threshold voltage is 3.7V. 111 = Brown-Out Detector threshold voltage is 4.4V. Note 1: This bit is write protected. Refer to the SYS_REGLCTL register. Note 2: Reset by power on reset.
[15]	Reserved Reserved.
[14:12]	LVRDGSEL LVR Output De-glitch Time Select (Write Protect) 000 = Without de-glitch function. 001 = 4 MIRC clock (4 MHz), 1 us. 010 = 8 MIRC clock (4 MHz), 2 us. 011 = 16 MIRC clock (4 MHz), 4 us. 100 = 32 MIRC clock (4 MHz), 8 us. 101 = 64 MIRC clock (4 MHz), 16 us. 110 = 128 MIRC clock (4 MHz), 32 us. 111 = 256 MIRC clock (4 MHz), 64 us. Note 1: These bits are write protected. Refer to the SYS_REGLCTL register. Note 2: The MIRC is enabled automatically when LVRDGSEL is not 000 and LVREN is 1.

Bits	Description	
[11]	Reserved	Reserved.
[10:8]	BODDGSEL	Brown-out Detector Output De-glitch Time Select (Write Protect) 000 = BOD output is sampled by LIRC. 001 = 4 system clock (HCLK). 010 = 8 system clock (HCLK). 011 = 16 system clock (HCLK). 100 = 32 system clock (HCLK). 101 = 64 system clock (HCLK). 110 = 128 system clock (HCLK). 111 = 256 system clock (HCLK). Note: These bits are write protected. Refer to the SYS_REGLCTL register.
[7]	LVREN	Low Voltage Reset Enable Bit (Write Protect) The LVR function resets the chip when the input power voltage is lower than LVR circuit setting. LVR function is enabled by default. 0 = Low Voltage Reset function Disabled. 1 = Low Voltage Reset function Enabled. Note 1: After enabling the bit, the LVR function will be active with 3ms delay for LVR output stable (default). Note 2: For BOD low power mode to be active, this bit must be set to 1. Note 3: This bit is write protected. Refer to the SYS_REGLCTL register.
[6]	BODOUT	Brown-out Detector Output Status 0 = Brown-out Detector output status is 0. It means the detected voltage is higher than BODVL setting or BODEN is 0. 1 = Brown-out Detector output status is 1. It means the detected voltage is lower than BODVL setting. If the BODEN is 0, BOD function is disabled. This bit always responds 0000.
[5]	BODLPM	Brown-out Detector Low Power Mode (Write Protect) 0 = BOD operate in normal mode (default). 1 = BOD Low Power mode Enabled. Note 1: The BOD consumes about 100uA in normal mode, the low power mode can reduce the current to about 1/10 but slow the BOD response. Note 2: For BOD low power mode to be active, LVREN must be set to 1 Note 3: This bit is write protected. Refer to the SYS_REGLCTL register.
[4]	BODIF	Brown-out Detector Interrupt Flag 0 = Brown-out Detector does not detect any voltage draft at V_{DD} down through or up through the voltage of BODVL setting. 1 = When Brown-out Detector detects the V_{DD} is dropped down through the voltage of BODVL setting or the V_{DD} is raised up through the voltage of BODVL setting, this bit is set to 1 and the brown-out interrupt is requested if brown-out interrupt is enabled. Note: Write 1 to clear this bit to 0.

Bits	Description	
[3]	BODRSTEN	Brown-out Reset Enable Bit (Write Protect) The default value is set by Flash controller user configuration register CBORST(CONFIG0[20]) bit. 0 = Brown-out "INTERRUPT" function Enabled. 1 = Brown-out "RESET" function Enabled. Note 1: While the Brown-out Detector function is enabled (BODEN high) and BOD reset function is enabled (BODRSTEN high), BOD will assert a signal to reset chip when the detected voltage is lower than the threshold (BODOUT high). While the BOD function is enabled (BODEN high) and BOD interrupt function is enabled (BODRSTEN low), BOD will assert an interrupt if BODOUT is high. BOD interrupt will keep till to the BODEN set to 0. BOD interrupt can be blocked by disabling the NVIC BOD interrupt or disabling BOD function (set BODEN low). Note 2: This bit is write protected. Refer to the SYS_REGLCTL register. Note 3: Reset by power on reset.
[2:1]	Reserved	Reserved.
[0]	BODEN	Brown-out Detector Enable Bit (Write Protect) The default value is set by Flash controller user configuration register CBODEN (CONFIG0 [19]). 0 = Brown-out Detector function Disabled. 1 = Brown-out Detector function Enabled. Note 1: This bit is write protected. Refer to the SYS_REGLCTL register. Note 2: Reset by power on reset.

Internal Voltage Source Control Register (SYS_IVSCTL)

Register	Offset	R/W	Description	Reset Value
SYS_IVSCTL	SYS_BA+0x1C	R/W	Internal Voltage Source Control Register	0x0000_0000

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

Bits	Description	
[31:1]	Reserved	Reserved.
[0]	VTEMPEN	Temperature Sensor Enable Bit This bit is used to enable/disable temperature sensor function. 0 = Temperature sensor function Disabled (default). 1 = Temperature sensor function Enabled.

Power-on Reset Controller Register 0 (SYS_PORCTL0)

Register	Offset	R/W	Description	Reset Value
SYS_PORCTL0	SYS_BA+0x24	R/W	Power-On-reset Controller Register 0	0x0000_0000

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

Bits	Description
[31:16]	Reserved Reserved.
[15:0]	PORMASK Power-on Reset Mask Enable Bit (Write Protect) When powered on, the POR circuit generates a reset signal to reset the whole chip function, but noise on the power may cause the POR active again. User can mask internal POR signal to avoid unpredictable noise to cause chip reset by writing 0x5AA5 to this field. The POR function will be active again when this field is set to another value or chip is reset by other reset source, including: nRESET, Watchdog, LVR reset, BOD reset, ICE reset command and the software-chip reset function. Note: These bits are write protected. Refer to the SYS_REGLCTL register.

GPIOA Low Byte Multiple Function Control Register (SYS_GPA_MFPL)

Register	Offset	R/W	Description	Reset Value
SYS_GPA_MFPL	SYS_BA+0x30	R/W	GPIOA Low Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PA7MFP				PA6MFP			
23	22	21	20	19	18	17	16
PA5MFP				PA4MFP			
15	14	13	12	11	10	9	8
PA3MFP				PA2MFP			
7	6	5	4	3	2	1	0
PA1MFP				PA0MFP			

Bits	Description	
[31:28]	PA7MFP	PA.7 Multi-function Pin Selection
[27:24]	PA6MFP	PA.6 Multi-function Pin Selection
[23:20]	PA5MFP	PA.5 Multi-function Pin Selection
[19:16]	PA4MFP	PA.4 Multi-function Pin Selection
[15:12]	PA3MFP	PA.3 Multi-function Pin Selection
[11:8]	PA2MFP	PA.2 Multi-function Pin Selection
[7:4]	PA1MFP	PA.1 Multi-function Pin Selection
[3:0]	PA0MFP	PA.0 Multi-function Pin Selection

GPIOA High Byte Multiple Function Control Register (SYS GPA MFPH)

Register	Offset	R/W	Description	Reset Value
SYS_GPA_MFPH	SYS_BA+0x34	R/W	GPIOA High Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PA15MFP				PA14MFP			
23	22	21	20	19	18	17	16
PA13MFP				PA12MFP			
15	14	13	12	11	10	9	8
PA11MFP				PA10MFP			
7	6	5	4	3	2	1	0
PA9MFP				PA8MFP			

Bits	Description	
[31:28]	PA15MFP	PA.15 Multi-function Pin Selection
[27:24]	PA14MFP	PA.14 Multi-function Pin Selection
[23:20]	PA13MFP	PA.13 Multi-function Pin Selection
[19:16]	PA12MFP	PA.12 Multi-function Pin Selection
[15:12]	PA11MFP	PA.11 Multi-function Pin Selection
[11:8]	PA10MFP	PA.10 Multi-function Pin Selection
[7:4]	PA9MFP	PA.9 Multi-function Pin Selection
[3:0]	PA8MFP	PA.8 Multi-function Pin Selection

GPIOB Low Byte Multiple Function Control Register (SYS_GPB_MFPL)

Register	Offset	R/W	Description	Reset Value
SYS_GPB_MFPL	SYS_BA+0x38	R/W	GPIOB Low Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PB7MFP				PB6MFP			
23	22	21	20	19	18	17	16
PB5MFP				PB4MFP			
15	14	13	12	11	10	9	8
PB3MFP				PB2MFP			
7	6	5	4	3	2	1	0
PB1MFP				PB0MFP			

Bits	Description	
[31:28]	PB7MFP	PB.7 Multi-function Pin Selection
[27:24]	PB6MFP	PB.6 Multi-function Pin Selection
[23:20]	PB5MFP	PB.5 Multi-function Pin Selection
[19:16]	PB4MFP	PB.4 Multi-function Pin Selection
[15:12]	PB3MFP	PB.3 Multi-function Pin Selection
[11:8]	PB2MFP	PB.2 Multi-function Pin Selection
[7:4]	PB1MFP	PB.1 Multi-function Pin Selection
[3:0]	PB0MFP	PB.0 Multi-function Pin Selection

GPIOB High Byte Multiple Function Control Register (SYS_GPB_MFPH)

Register	Offset	R/W	Description	Reset Value
SYS_GPB_MFPH	SYS_BA+0x3C	R/W	GPIOB High Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PB15MFP				PB14MFP			
23	22	21	20	19	18	17	16
PB13MFP				PB12MFP			
15	14	13	12	11	10	9	8
PB11MFP				PB10MFP			
7	6	5	4	3	2	1	0
PB9MFP				PB8MFP			

Bits	Description
[31:28]	PB15MFP PB.15 Multi-function Pin Selection
[27:24]	PB14MFP PB.14 Multi-function Pin Selection
[23:20]	PB13MFP PB.13 Multi-function Pin Selection
[19:16]	PB12MFP PB.12 Multi-function Pin Selection
[15:12]	PB11MFP PB.11 Multi-function Pin Selection
[11:8]	PB10MFP PB.10 Multi-function Pin Selection
[7:4]	PB9MFP PB.9 Multi-function Pin Selection
[3:0]	PB8MFP PB.8 Multi-function Pin Selection

GPIOC Low Byte Multiple Function Control Register (SYS_GPC_MFPL)

Register	Offset	R/W	Description	Reset Value
SYS_GPC_MFPL	SYS_BA+0x40	R/W	GPIOC Low Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PC7MFP				PC6MFP			
23	22	21	20	19	18	17	16
PC5MFP				PC4MFP			
15	14	13	12	11	10	9	8
PC3MFP				PC2MFP			
7	6	5	4	3	2	1	0
PC1MFP				PC0MFP			

Bits	Description	
[31:28]	PC7MFP	PC.7 Multi-function Pin Selection
[27:24]	PC6MFP	PC.6 Multi-function Pin Selection
[23:20]	PC5MFP	PC.5 Multi-function Pin Selection
[19:16]	PC4MFP	PC.4 Multi-function Pin Selection
[15:12]	PC3MFP	PC.3 Multi-function Pin Selection
[11:8]	PC2MFP	PC.2 Multi-function Pin Selection
[7:4]	PC1MFP	PC.1 Multi-function Pin Selection
[3:0]	PC0MFP	PC.0 Multi-function Pin Selection

GPIOC High Byte Multiple Function Control Register (SYS_GPC_MFPH)

Register	Offset	R/W	Description	Reset Value
SYS_GPC_MFPH	SYS_BA+0x44	R/W	GPIOC High Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PC15MFP				PC14MFP			
23	22	21	20	19	18	17	16
PC13MFP				PC12MFP			
15	14	13	12	11	10	9	8
PC11MFP				PC10MFP			
7	6	5	4	3	2	1	0
PC9MFP				PC8MFP			

Bits	Description	
[31:28]	PC15MFP	PC.15 Multi-function Pin Selection
[27:24]	PC14MFP	PC.14 Multi-function Pin Selection
[23:20]	PC13MFP	PC.13 Multi-function Pin Selection
[19:16]	PC12MFP	PC.12 Multi-function Pin Selection
[15:12]	PC11MFP	PC.11 Multi-function Pin Selection
[11:8]	PC10MFP	PC.10 Multi-function Pin Selection
[7:4]	PC9MFP	PC.9 Multi-function Pin Selection
[3:0]	PC8MFP	PC.8 Multi-function Pin Selection

GPIOF Low Byte Multiple Function Control Register (SYS_GPF_MFPL)

Register	Offset	R/W	Description	Reset Value
SYS_GPF_MFPL	SYS_BA+0x58	R/W	GPIOF Low Byte Multiple Function Control Register	0x0000_00ee

31	30	29	28	27	26	25	24
PF7MFP				PF6MFP			
23	22	21	20	19	18	17	16
PF5MFP				PF4MFP			
15	14	13	12	11	10	9	8
PF3MFP				PF2MFP			
7	6	5	4	3	2	1	0
PF1MFP				PF0MFP			

Bits	Description	
[31:28]	PF7MFP	PF.7 Multi-function Pin Selection
[27:24]	PF6MFP	PF.6 Multi-function Pin Selection
[23:20]	PF5MFP	PF.5 Multi-function Pin Selection
[19:16]	PF4MFP	PF.4 Multi-function Pin Selection
[15:12]	PF3MFP	PF.3 Multi-function Pin Selection
[11:8]	PF2MFP	PF.2 Multi-function Pin Selection
[7:4]	PF1MFP	PF.1 Multi-function Pin Selection
[3:0]	PF0MFP	PF.0 Multi-function Pin Selection

GPIOF High Byte Multiple Function Control Register (SYS_GPF_MFPH)

Register	Offset	R/W	Description	Reset Value
SYS_GPF_MFPH	SYS_BA+0x5C	R/W	GPIOF High Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PF15MFP				PF14MFP			
23	22	21	20	19	18	17	16
PF13MFP				PF12MFP			
15	14	13	12	11	10	9	8
PF11MFP				PF10MFP			
7	6	5	4	3	2	1	0
PF9MFP				PF8MFP			

Bits	Description	
[31:28]	PF15MFP	PF.15 Multi-function Pin Selection
[27:24]	PF14MFP	PF.14 Multi-function Pin Selection
[23:20]	PF13MFP	PF.13 Multi-function Pin Selection
[19:16]	PF12MFP	PF.12 Multi-function Pin Selection
[15:12]	PF11MFP	PF.11 Multi-function Pin Selection
[11:8]	PF10MFP	PF.10 Multi-function Pin Selection
[7:4]	PF9MFP	PF.9 Multi-function Pin Selection
[3:0]	PF8MFP	PF.8 Multi-function Pin Selection

GPIO A-F Multiple Function Output Select Register (SYS_GP_x MFOS)

Register	Offset	R/W	Description	Reset Value
SYS_GPA_MFOS	SYS_BA+0x80	R/W	GPIOA Multiple Function Output Select Register	0x0000_0000
SYS_GPB_MFOS	SYS_BA+0x84	R/W	GPIOB Multiple Function Output Select Register	0x0000_0000
SYS_GPC_MFOS	SYS_BA+0x88	R/W	GPIOC Multiple Function Output Select Register	0x0000_0000
SYS_GPF_MFOS	SYS_BA+0x94	R/W	GPIOF Multiple Function Output Select Register	0x0000_0000

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

Bits	Description
[31:16]	Reserved Reserved.
[n] n=0,1..15	GPIOA-f Pin[n] Multiple Function Pin Output Mode Select This bit used to select multiple function pin output mode type for Px.n pin 0 = Multiple function pin output mode type is Push-pull mode. 1 = Multiple function pin output mode type is Open-drain mode. Note: Max. n=15.

System SRAM BIST Test Control Register (SYS_SRAM_BISTCTL)

Register	Offset	R/W	Description	Reset Value
SYS_SRAM_BISTCTL	SYS_BA+0xD0	R/W	System SRAM BIST Test Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
PDMABIST	CANFD0BIST	Reserved	USBBI	Reserved	FMC BIST	Reserved	SRBIST

Bits	Description
[31:8]	Reserved Reserved.
[7]	PDMABIST PDMA BIST Enable Bit (Write Protect) This bit enables BIST test for PDMA RAM. 0 = System PDMA BIST Disabled. 1 = System PDMA BIST Enabled. Note: This bit is write protected. Refer to the SYS_REGLCTL register.
[6]	CANFD0BIST CANFD0 BIST Enable Bit (Write Protect) This bit enables BIST test for CANFD0 RAM. 0 = System CANFD0 BIST Disabled. 1 = System CANFD0 BIST Enabled. Note: This bit is write protected. Refer to the SYS_REGLCTL register.
[5]	Reserved Reserved.
[4]	USBBI USB BIST Enable Bit (Write Protect) This bit enables BIST test for USB RAM 0 = System USB BIST Disabled. 1 = System USB BIST Enabled. Note: This bit is write protected. Refer to the SYS_REGLCTL register.
[3]	Reserved Reserved.
[2]	FMC BIST FMC CACHE BIST Enable Bit (Write Protect) This bit enables BIST test for CACHE RAM. 0 = System CACHE BIST Disabled. 1 = System CACHE BIST Enabled. Note: This bit is write protected. Refer to the SYS_REGLCTL register.
[1]	Reserved Reserved.
[0]	SRBIST SRAM-BIST Enable Bit (Write Protect) This bit enables BIST test for SRAM.

		0 = System SRAM BIST Disabled. 1 = System SRAM BIST Enabled. Note: This bit is write protected. Refer to the SYS_REGLCTL register.

System SRAM BIST Test Status Register (SYS_SRAM_BISTSTS)

Register	Offset	R/W	Description	Reset Value
SYS_SRAM_BISTSTS	SYS_BA+0xD4	R	System SRAM BIST Test Status Register	0x00xx_00xx

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
PDMAEND	CANFD0END	Reserved	USBEND	Reserved	CR1BEND	CR0BEND	SRBEND
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
PDMABISTF	CANFD0BISTF	Reserved	USBDEF	Reserved	CR1BISTEF	CR0BISTEF	SRBISTEF

Bits	Description
[31:24]	Reserved Reserved.
[23]	PDMAEND PDMA SRAM BIST Test Finish 0 = PDMA SRAM BIST is active. 1 = PDMA SRAM BIST test finished.
[22]	CANFD0END CANFD0 SRAM BIST Test Finish 0 = CANFD0 SRAM BIST is active. 1 = CANFD0 SRAM BIST test finished.
[21]	Reserved Reserved.
[20]	USBEND USB SRAM BIST Test Finish 0 = USB SRAM BIST is active. 1 = USB SRAM BIST test finished.
[19]	Reserved Reserved.
[18]	CR1BEND CACHE 1 SRAM BIST Test Finish 0 = System CACHE RAM BIST is active. 1 = System CACHE RAM BIST test finished.
[17]	CR0BEND CACHE 0 SRAM BIST Test Finish 0 = System CACHE RAM BIST is active. 1 = System CACHE RAM BIST test finished.
[16]	SRBEND SRAM BIST Test Finish 0 = System SRAM BIST active. 1 = system SRAM BIST finished.
[15:8]	Reserved Reserved.
[7]	PDMABISTF PDMA SRAM BIST Failed Flag 0 = PDMA SRAM BIST passed. 1 = PDMA SRAM BIST failed.

[6]	CANFD0BISTF	CANFD0 SRAM BIST Failed Flag 0 = CANFD0 SRAM BIST passed. 1 = CANFD0 SRAM BIST failed.
[5]	Reserved	Reserved.
[4]	USBBEF	USB SRAM BIST Fail Flag 0 = USB SRAM BIST test passed. 1 = USB SRAM BIST test failed.
[3]	Reserved	Reserved.
[2]	CR1BISTEF	CACHE1 SRAM BIST Fail Flag 0 = System CACHE RAM BIST test passed. 1 = System CACHE RAM BIST test failed.
[1]	CR0BISTEF	CACHE0 SRAM BIST Fail Flag 0 = System CACHE RAM BIST test passed. 1 = System CACHE RAM BIST test failed.
[0]	SRBISTEF	System SRAM BIST Fail Flag 0 = System SRAM BIST test passed. 1 = System SRAM BIST test failed.

HIRC Trim Control Register (SYS_HIRCTRIMCTL)

Register	Offset	R/W	Description	Reset Value
SYS_HIRCTRIMCTL	SYS_BA+0xF0	R/W	HIRC Trim Control Register	0x0008_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved				BOUNDARY			
15	14	13	12	11	10	9	8
Reserved					REFCKSEL	BOUNDEN	CESTOPEN
7	6	5	4	3	2	1	0
RETRYCNT		LOOPSEL		Reserved		FREQSEL	

Bits	Description
[31:21]	Reserved Reserved.
[20:16]	BOUNDARY Boundary Selection Fill the boundary range from 0x1 to 0x1F. 0x0 is reserved. Note: This field is effective only when the BOUNDEN(SYS_HIRCTRIMCTL[9]) is enabled.
[15:11]	Reserved Reserved.
[10]	REFCKSEL Reference Clock Selection 0 = HIRC trim reference clock is from LXT (32.768 kHz). 1 = HIRC trim reference clock is from internal USB synchronous mode
[9]	BOUNDEN Boundary Enable Bit 0 = Boundary function Disabled. 1 = Boundary function Enabled.
[8]	CESTOPEN Clock Error Stop Enable Bit 0 = The trim operation keeps going if clock is inaccurate. 1 = The trim operation stops if clock is inaccurate.
[7:6]	RETRYCNT Trim Value Update Limitation Count This field defines that how many times the auto trim circuit will try to update the HIRC trim value before the frequency of HIRC locked. Once the HIRC locked, the internal trim value update counter will be reset. If the trim value update counter reached this limitation value and frequency of HIRC is still not locked, the auto trim operation will be disabled and FREQSEL will be cleared to 00. 00 = Trim retry count limitation is 64 loops. 01 = Trim retry count limitation is 128 loops. 10 = Trim retry count limitation is 256 loops. 11 = Trim retry count limitation is 512 loops.
[5:4]	LOOPSEL Trim Calculation Loop Selection This field defines that trim value calculation is based on how many reference clocks. 00 = Trim value calculation is based on average difference in 4 clocks of reference clock.

		01 = Trim value calculation is based on average difference in 8 clocks of reference clock. 10 = Trim value calculation is based on average difference in 16 clocks of reference clock. 11 = Trim value calculation is based on average difference in 32 clocks of reference clock. Note: For example, if LOOPSEL is set as 00, auto trim circuit will calculate trim value based on the average frequency difference in 4 clocks of reference clock.
[3:2]	Reserved	Reserved.
[1:0]	FREQSEL	Trim Frequency Selection This field indicates the target frequency of 48 MHz internal high speed RC oscillator (HIRC) auto trim. During auto trim operation, if clock error detected with CESTOPEN is set to 1 or trim retry limitation count reached, this field will be cleared to 00 automatically. 00 = Disable HIRC auto trim function. 01 = Enable HIRC auto trim function and trim HIRC to 48 MHz. 10 = Reserved. 11 = Reserved.

HIRC Trim Interrupt Enable Register (SYS_HIRCTRIMIEN)

Register	Offset	R/W	Description	Reset Value
SYS_HIRCTRIMIEN	SYS_BA+0xF4	R/W	HIRC Trim Interrupt Enable Register	0x0000_0000

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

Bits	Description
[31:3]	Reserved Reserved.
[2]	CLKEIEN Clock Error Interrupt Enable Bit This bit controls if CPU would get an interrupt while clock is inaccurate during auto trim operation. If this bit is set to 1, and CLKERRIF(SYS_HIRCTRIMSTS[2]) is set during auto trim operation, an interrupt will be triggered to notify the clock frequency is inaccurate. 0 = Disable CLKERRIF(SYS_HIRCTRIMSTS[2]) status to trigger an interrupt to CPU. 1 = Enable CLKERRIF(SYS_HIRCTRIMSTS[2]) status to trigger an interrupt to CPU.
[1]	TFALIEN Trim Failure Interrupt Enable Bit This bit controls if an interrupt will be triggered while HIRC trim value update limitation count reached and HIRC frequency is still not locked on target frequency set by FREQSEL(SYS_HIRCTRIMCTL[1:0]). If this bit is high and TFALIF(SYS_HIRCTRIMSTS[1]) is set during auto trim operation, an interrupt will be triggered to notify that HIRC trim value update limitation count reached. 0 = Disable TFALIF(SYS_HIRCTRIMSTS[1]) status to trigger an interrupt to CPU. 1 = Enable TFALIF(SYS_HIRCTRIMSTS[1]) status to trigger an interrupt to CPU.
[0]	Reserved Reserved.

HIRC Trim Interrupt Status Register (SYS_HIRCTRIMSTS)

Register	Offset	R/W	Description	Reset Value
SYS_HIRCTRIMSTS	SYS_BA+0xF8	R/W	HIRC Trim Interrupt Status Register	0x0000_0000

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

Bits	Description
[31:4]	Reserved Reserved.
[3]	OVBDF Over Boundary Status When the over boundary function is set, if there occurs the over boundary condition, this flag will be set. 0 = Over boundary condition did not occur. 1 = Over boundary condition occurred. Note: Write 1 to clear this flag.
[2]	CLKERIF Clock Error Interrupt Status When the frequency of 32.768 kHz external low speed crystal oscillator (LXT) or 48 MHz internal high speed RC oscillator (HIRC) is shift larger to unreasonable value, this bit will be set and to be an indicate that clock frequency is inaccurate. Once this bit is set to 1, the auto trim operation stopped and FREQSEL(SYS_HIRCTRIMCTL[1:0]) will be cleared to 00 by hardware automatically if CESTOPEN(SYS_HIRCTRIMCTL[8]) is set to 1. If this bit is set and CLKEIEN(SYS_HIRCTRIMIEN[2]) is high, an interrupt will be triggered to notify the clock frequency is inaccuracy. Write 1 to clear this to 0. 0 = Clock frequency is accurate. 1 = Clock frequency is inaccurate. Note: Reset by power on reset.
[1]	TFAILIF Trim Failure Interrupt Status This bit indicates that HIRC trim value update limitation count reached and the HIRC clock frequency is still not locked. Once this bit is set, the auto trim operation stopped and FREQSEL(SYS_HIRCTRIMCTL[1:0]) will be cleared to 00 by hardware automatically. If this bit is set and TFALIEN(SYS_HIRCTRIMIEN[1]) is high, an interrupt will be triggered to notify that HIRC trim value update limitation count reached. Write 1 to clear this to 0. 0 = Trim value update limitation count not reached. 1 = Trim value update limitation count reached and HIRC frequency still not locked. Note: Reset by power on reset.
[0]	FREQLOCK HIRC Frequency Lock Status This bit indicates the HIRC frequency is locked. This is a status bit and doesn't trigger any interrupt

		Write 1 to clear this to 0. This bit will be set automatically if the frequency is locked and the RC_TRIM is enabled. 0 = The internal high-speed oscillator frequency is not locked at 48 MHz. 1 = The internal high-speed oscillator frequency locked at 48 MHz. Note: Reset by power on reset.

Register Lock Control Register (SYS_REGLCTL)

Some of the system control registers need to be protected to avoid inadvertent write and disturb the chip operation. These system control registers are protected after the power-on reset till user to disable register protection. For user to program these protected registers, a register protection disable sequence needs to be followed by a special programming. The register protection disable sequence is writing the data "59h", "16h" "88h" to the register SYS_REGLCTL address at 0x4000_0100 continuously. Any different data value, different sequence or any other write to other address during these three data writing will abort the whole sequence.

After the protection is disabled, user can check the protection disable bit at address 0x4000_0100 bit0, 1 is protection disable, and 0 is protection enable. Then user can update the target protected register value and then write any data to the address "0x4000_0100" to enable register protection.

This register is written to disable/enable register protection and read for the REGLCTL status.

Register	Offset	R/W	Description	Reset Value
SYS_REGLCTL	SYS_BA+0x100	R/W	Register Lock Control Register	0x0000_0000

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

Bits	Description
[31:8]	Reserved Reserved.
[7:0]	REGLCTL Register Lock Control Code Some registers have write-protection function. Writing these registers have to disable the protected function by writing the sequence value "59h", "16h", "88h" to this field. After this sequence is completed, the REGLCTL bit will be set to 1 and write-protection registers can be normal write. Register Lock Control Disable Index 0 = Write-protection Enabled for writing protected registers. Any write to the protected register is ignored. 1 = Write-protection Disabled for writing protected registers. The Protected registers are: NMIEN address 0x4000_0300 FMC_ISPCTL address 0x4000_C000 (Flash ISP Control register) FMC_ISPTRG address 0x4000_C010 (ISP Trigger Control register) FMC_ISPSTS address 0x4000_C040 WDT_CTL address 0x4004_0000 FMC_FTCTL address 0x4000_5018 FMC_ICPCMD address 0x4000_501C EADC_TEST address 0x4004_3200 AHBMCTL address 0x4000_0400

	SYS_IPRST0	address 0x4000_0008
	SYS_BODCTL	address 0x4000_0018
	SYS_PORCTL0	address 0x4000_0024
	SYS_SRAM_BISTCTL	address 0x4000_00D0
	SYS_PORCTL1	address 0x4000_1EC
	CLK_PWRCTL	address 0x4000_0200
	CLK_APBCLK0[0]	address 0x4000_0208
	CLK_CLKSEL0	address 0x4000_0110
	CLK_CLKSEL1[3:0]	address 0x4000_0214
	CLK_PLLCTL	address 0x4000_0240
	CLK_PMUCTL	address 0x4000_0290
	CLK_HXTFSEL	address 0x4000_02B4

MIRC Trim Control Register (SYS_MIRCTRIMCTL)

Register	Offset	R/W	Description	Reset Value
SYS_MIRCTRIMCTL	SYS_BA+0x104	R/W	MIRC Trim Control Register	0x0008_0010

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved				BOUNDARY			
15	14	13	12	11	10	9	8
Reserved					REFCKSEL	BOUNDEN	CESTOPEN
7	6	5	4	3	2	1	0
RETRYCNT		LOOPSEL		Reserved		FREQSEL	

Bits	Description
[31:21]	Reserved Reserved.
[20:16]	BOUNDARY Boundary Selection Fill the boundary range from 0x1 to 0x1F. 0x0 is reserved. Note: This field is effective only when the BOUNDEN(SYS_MIRCTRIMCTL[9]) is enabled.
[15:11]	Reserved Reserved.
[10]	REFCKSEL Reference Clock Selection 0 = MIRC trim reference clock is from LXT (32.768 kHz). 1 = MIRC trim reference clock is from internal USB synchronous mode
[9]	BOUNDEN Boundary Enable Bit 0 = Boundary function Disabled. 1 = Boundary function Enabled.
[8]	CESTOPEN Clock Error Stop Enable Bit 0 = The trim operation keeps going if clock is inaccurate. 1 = The trim operation stops if clock is inaccurate.
[7:6]	RETRYCNT Trim Value Update Limitation Count This field defines that how many times the auto trim circuit will try to update the MIRC trim value before the frequency of MIRC locked. Once the MIRC locked, the internal trim value update counter will be reset. If the trim value update counter reached this limitation value and frequency of MIRC is still not locked, the auto trim operation will be disabled and FREQSEL will be cleared to 00. 00 = Trim retry count limitation is 64 loops. 01 = Trim retry count limitation is 128 loops. 10 = Trim retry count limitation is 256 loops. 11 = Trim retry count limitation is 512 loops.
[5:4]	LOOPSEL Trim Calculation Loop Selection This field defines that trim value calculation is based on how many reference clocks. 00 = Reserved.

		01 = Trim value calculation is based on average difference in 8 clocks of reference clock. 10 = Trim value calculation is based on average difference in 16 clocks of reference clock. 11 = Trim value calculation is based on average difference in 32 clocks of reference clock. Note: For example, if LOOPSEL is set as 00, auto trim circuit will calculate trim value based on the average frequency difference in 4 clocks of reference clock.
[3:2]	Reserved	Reserved.
[1:0]	FREQSEL	Trim Frequency Selection This field indicates the target frequency of medium speed RC oscillator (MIRC) auto trim. During auto trim operation, if clock error detected with CESTOPEN is set to 1 or trim retry limitation count reached, this field will be cleared to 00 automatically. 00 = Disable HIRC auto trim function. 01 = Reserved. 10 = Enable HIRC auto trim function and trim MIRC to 4.032 MHz.

MIRC Trim Interrupt Enable Register (SYS_MIRCTRIMIEN)

Register	Offset	R/W	Description	Reset Value
SYS_MIRCTRIMIEN	SYS_BA+0x108	R/W	MIRC Trim Interrupt Enable Register	0x0000_0000

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

Bits	Description
[31:3]	Reserved Reserved.
[2]	CLKEIEN Clock Error Interrupt Enable Bit This bit controls if CPU would get an interrupt while clock is inaccurate during auto trim operation. If this bit is set to 1, and CLKERRIF(SYS_MIRCTRIMSTS[2]) is set during auto trim operation, an interrupt will be triggered to notify the clock frequency is inaccurate. 0 = Disable CLKERRIF(SYS_MIRCTRIMSTS[2]) status to trigger an interrupt to CPU. 1 = Enable CLKERRIF(SYS_MIRCTRIMSTS[2]) status to trigger an interrupt to CPU.
[1]	TFALIEN Trim Failure Interrupt Enable Bit This bit controls if an interrupt will be triggered while MIRC trim value update limitation count reached and MIRC frequency still not locked on target frequency set by FREQSEL(SYS_MIRTRIMCTL[1:0]). If this bit is high and TFALIF(SYS_MIRCTRIMSTS[1]) is set during auto trim operation, an interrupt will be triggered to notify that MIRC trim value update limitation count reached. 0 = Disable TFALIF(SYS_MIRCTRIMSTS[1]) status to trigger an interrupt to CPU. 1 = Enable TFALIF(SYS_MIRCTRIMSTS[1]) status to trigger an interrupt to CPU.
[0]	Reserved Reserved.

MIRC Trim Interrupt Status Register (SYS_MIRCTRIMSTS)

Register	Offset	R/W	Description	Reset Value
SYS_MIRCTRIMSTS	SYS_BA+0x10C	R/W	MIRC Trim Interrupt Status Register	0x0000_0000

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

Bits	Description
[31:4]	Reserved Reserved.
[3]	OVBDF Over Boundary Status When the over boundary function is set, if there occurs the over boundary condition, this flag will be set. 0 = Over boundary condition did not occur. 1 = Over boundary condition occurred. Note: Write 1 to clear this flag.
[2]	CLKERIF Clock Error Interrupt Status When the frequency of 32.768 kHz external low speed crystal oscillator (LXT) or internal medium speed RC oscillator (MIRC) is shift larger to unreasonable value, this bit will be set and be an indicate that clock frequency is inaccurate. Once this bit is set to 1, the auto trim operation stopped and FREQSEL(SYS_MIRCTRIMCTL[1:0]) will be cleared to 00 by hardware automatically if CESTOPEN(SYS_MIRCTRIMCTL[8]) is set to 1. If this bit is set and CLKEIEN(SYS_MIRCTRIMIEN[2]) is high, an interrupt will be triggered to notify the clock frequency is inaccurate. Write 1 to clear this to 0. 0 = Clock frequency is accurate. 1 = Clock frequency is inaccurate. Note: Reset by power on reset.
[1]	TFAILIF Trim Failure Interrupt Status This bit indicates that MIRC trim value update limitation count reached and the MIRC clock frequency is still not locked. Once this bit is set, the auto trim operation stopped and FREQSEL(SYS_MIRCTRIMCTL[1:0]) will be cleared to 00 by hardware automatically. If this bit is set and TFALIEN(SYS_MIRCTRIMIEN[1]) is high, an interrupt will be triggered to notify that MIRC trim value update limitation count reached. Write 1 to clear this to 0. 0 = Trim value update limitation count not reached. 1 = Trim value update limitation count reached and MIRC frequency still not locked. Note: Reset by power on reset.
[0]	FREQLOCK MIRC Frequency Lock Status This bit indicates the MIRC frequency is locked. This is a status bit and doesn't trigger any interrupt.

		Write 1 to clear this to 0. This bit will be set automatically if the frequency is locked and the RC_TRIM is enabled. 0 = The internal medium-speed oscillator frequency not locked. 1 = The internal medium-speed oscillator frequency locked. Note: Reset by power on reset.

Power on Reset Controller Register 1 (SYS_PORCTL1)

Register	Offset	R/W	Description	Reset Value
SYS_PORCTL1	SYS_BA+0x1EC	R/W	Power-On-reset Controller Register 1	0x0000_0000

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

Bits	Description
[31:16]	Reserved Reserved.
[15:0]	Power-on Reset Enable Bit (Write Protect) After powered on, User can turn off internal analog POR circuit to save power by writing 0x5AA5 to this field. The analog POR circuit will be active again when this field is set to another value or chip is reset by other reset source, including: nRESET, Watchdog, LVR reset, BOD reset, ICE reset command and the software-chip reset function. Note: This bit is write protected. Refer to the SYS_REGLCTL register.

Power Level Control Register (SYS_PLCTL)

Register	Offset	R/W	Description	Reset Value
SYS_PLCTL	SYS_BA+0x1F8	R/W	Power Level Control Register	0x0000_0000

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

Bits	Description	
[31:2]	Reserved	Reserved.
[1:0]	PLSEL	Power Level Select(Write Protect) 00 = Set to power level 0 (PL0). 11 = Set to power level 3 (PL3). Others = Reserved. Note : When system is at PL3, HCLK clock has to come from LXT or LIRC.

Power Level Status Register (SYS_PLSTS)

Register	Offset	R/W	Description	Reset Value
SYS_PLSTS	SYS_BA+0x1FC	R/W	Power Level Status Register	0x0000_000X

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

Bits	Description
[31:10]	Reserved Reserved.
[9:8]	CURPL Current Power Level (Read Only) This bit field reflect the current power level. 00 = Current power level is PL0. 11 = Current power level is PL3. Others = Reserved. Note : When system is at PL3, HCLK clock has to come from LXT or LIRC.
[7:1]	Reserved Reserved.
[0]	PLCBUSY Power Level Change Busy Bit (Read Only) This bit is set by hardware when power level is changing. After power level change is completed, this bit will be cleared automatically by hardware. 0 = Power level change is completed. 1 = Power level change is ongoing.

6.2.11 System Timer (SysTick)

The Cortex®-M23 includes an integrated system timer, SysTick, which provides a simple, 24-bit clear-on-write, decrementing, wrap-on-zero counter with a flexible control mechanism. The counter can be used as a Real Time Operating System (RTOS) tick timer or as a simple counter.

When system timer is enabled, it will count down from the value in the SysTick Current Value Register (SYST_VAL) to zero, and reload (wrap) to the value in the SysTick Reload Value Register (SYST_LOAD) on the next clock cycle, and then decrement on subsequent clocks. When the counter transitions to zero, the COUNTFLAG status bit is set. The COUNTFLAG bit clears on reads.

The SYST_VAL value is UNKNOWN on reset. Software should write to the register to clear it to zero before enabling the feature. This ensures the timer will count from the SYST_LOAD value rather than an arbitrary value when it is enabled.

If the SYST_LOAD is zero, the timer will be maintained with a current value of zero after it is reloaded with this value. This mechanism can be used to disable the feature independently from the timer enable bit.

For more detailed information, please refer to the “Arm® Cortex®-M23 Technical Reference Manual” and “Arm® v8-M Architecture Reference Manual”.

6.2.11.1 System Timer Control Register Map

R: read only, W: write only, R/W: both read and write

Register	Offset	R/W	Description	Reset Value
SYST Base Address: SCS_BA = 0xE000_E000				
SYST_CTRL	SCS_BA+0x10	R/W	SysTick Control and Status Register	0x0000_0000
SYST_LOAD	SCS_BA+0x14	R/W	SysTick Reload Value Register	0xFFFF_FFFF
SYST_VAL	SCS_BA+0x18	R/W	SysTick Current Value Register	0xFFFF_FFFF

6.2.11.2 System Timer Control Register Description

SysTick Control and Status Register (SYST_CTRL)

Register	Offset	R/W	Description	Reset Value
SYST_CTRL	SCS_BA+0x10	R/W	SysTick Control and Status Register	0x0000_0000

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

Bits	Description
[31:17]	Reserved Reserved.
[16]	COUNTFLAG System Tick Counter Flag Returns 1 if timer counted to 0 since last time this register was read. COUNTFLAG is set by a count transition from 1 to 0. COUNTFLAG is cleared on read or by a write to the Current Value register.
[15:3]	Reserved Reserved.
[2]	CLKSRC System Tick Clock Source Selection 0 = Clock source is the (optional) external reference clock. 1 = Core clock used for SysTick.
[1]	TICKINT System Tick Interrupt Enabled 0 = Counting down to 0 does not cause the SysTick exception to be pended. Software can use COUNTFLAG to determine if a count to zero has occurred. 1 = Counting down to 0 will cause the SysTick exception to be pended. Clearing the SysTick current value register by a register write in software will not cause SysTick to be pended.
[0]	ENABLE System Tick Counter Enabled 0 = Counter Disabled. 1 = Counter will operate in a multi-shot manner.

SysTick Reload Value Register (SYST_LOAD)

Register	Offset	R/W	Description	Reset Value
SYST_LOAD	SCS_BA+0x14	R/W	SysTick Reload Value Register	0xFFFF_XXXX

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

Bits	Description	
[31:24]	Reserved	Reserved.
[23:0]	RELOAD	System Tick Reload Value The value to load into the Current Value register when the counter reaches 0.

SysTick Current Value Register (SYST_VAL)

Register	Offset	R/W	Description	Reset Value
SYST_VAL	SCS_BA+0x18	R/W	SysTick Current Value Register	0XXXXX_XXXX

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

Bits	Description
[31:24]	Reserved Reserved.
[23:0]	CURRENT System Tick Current Value Current counter value. This is the value of the counter at the time it is sampled. The counter does not provide read-modify-write protection. The register is write-clear. A software write of any value will clear the register to 0.

6.2.12 Nested Vectored Interrupt Controller (NVIC)

The NVIC and the processor core interface are closely coupled to enable low latency interrupt processing and efficient processing of late arriving interrupts. The NVIC maintains knowledge of the stacked, or nested, interrupts to enable tail-chaining of interrupts. You can only fully access the NVIC from privileged mode, but you can cause interrupts to enter a pending state in user mode if you enable the Configuration and Control Register. Any other user mode access causes a bus fault. You can access all NVIC registers using byte, halfword, and word accesses unless otherwise stated. NVIC registers are located within the SCS (System Control Space). All NVIC registers and system debug registers are little-endian regardless of the endianness state of the processor.

The NVIC supports:

- An implementation-defined number of interrupts, in the range 1-64 interrupts.
- A programmable priority level of 0-3 for each interrupt; a higher level corresponds to a lower priority, so level 0 is the highest interrupt priority.
- Level and pulse detection of interrupt signals.
- Dynamic reprioritization of interrupts.
- Grouping of priority values into group priority and sub-priority fields.
- Interrupt tail-chaining.
- An external Non Maskable Interrupt (NMI)
- WIC with Ultra-low Power Sleep mode support

The processor automatically stacks its state on exception entry and unstacks this state on exception exit, with no instruction overhead. This provides low latency exception handling.

6.2.12.1 Exception Model and System Interrupt Map

Table 6.2-8 lists the exception model supported by the M253 series. Software can set 4 levels of priority on some of these exceptions as well as on all interrupts. The highest user-configurable priority is denoted as "0x00" and the lowest priority is denoted as "0xC0" (The 6-LSB always 0). The default priority of all the user-configurable interrupts is "0x00". Note that priority "0" is treated as the fourth priority on the system, after three system exceptions "Reset", "NMI" and "Hard Fault".

When any interrupts is accepted, the processor will automatically fetch the starting address of the interrupt service routine (ISR) from a vector table in memory. On system reset, the vector table is fixed at address 0x00000000. Privileged software can write to the VTOR to relocate the vector table start address to a different memory location, in the range 0x00000080 to 0x3FFFFFF80.

The vector table contains the initialization value for the stack pointer on reset, and the entry point addresses for all exception handlers. The vector number on previous page defines the order of entries in the vector table associated with exception handler entry as illustrated in previous section.

Exception Type	Vector Number	Vector Address	Priority
Reset	1	0x00000004	-3
NMI	2	0x00000008	-2
Hard Fault	3	0x0000000C	-1
Reserved	4~ 10		Reserved
SVCall	11	0x0000002C	Configurable
Reserved	12~13		Reserved
PendSV	14	0x00000038	Configurable
SysTick	15	0x0000003C	Configurable
Interrupt (IRQ0 ~ IRQ63)	16 ~ 63	0x00000000 + (Vector Number)*4	Configurable

Table 6.2-8 Exception Model

Vector Number	Interrupt Number (Bit In Interrupt Registers)	Interrupt Name	Interrupt Description
0 ~ 15	-	-	System exceptions
16	0	BODOUT	Brown-Out low voltage detected interrupt
17	1	IRC_INT	IRC TRIM interrupt
18	2	PWRWU_INT	Clock controller interrupt for chip wake-up from power-down state
19	3	Reserved	Reserved
20	4	CLKFAIL	Clock fail detected interrupt
21	5	Reserved	Reserved
22	6	RTC_INT	Real time clock interrupt
23	7	Reserved	Reserved
24	8	WDT_INT	Watchdog Timer interrupt
25	9	WWDT_INT	Window Watchdog Timer interrupt
26	10	EINT0	External interrupt from PA.0, PD.2 or PE.4 pins
27	11	EINT1	External interrupt from PB.0, PD.3 or PE.5 pins
28	12	EINT2	External interrupt from PC.0 pin
29	13	EINT3	External interrupt from PD.0 pin
30	14	EINT4	External interrupt from PE.0 pin
31	15	EINT5	External interrupt from PF.0 pin
32	16	GPA_INT	External interrupt from PA[15:0] pin
33	17	GPB_INT	External interrupt from PB[15:0] pin
34	18	GPC_INT	External interrupt from PC[15:0] pin

35	19	Reserved	Reserved
36	20	Reserved	Reserved
37	21	GPF_INT	External interrupt from PF[15:0] pin
38	22	Reserved	Reserved
39	23	SPI0_INT	SPI0 interrupt
40	24	CANFD0_INT0	CANFD0 interrupt 0
41	25	CANFD0_INT1	CANFD0 interrupt 1
42	26	Reserved	Reserved
43	27	Reserved	Reserved
44	28	Reserved	Reserved
45	29	Reserved	Reserved
46	30	Reserved	Reserved
47	31	Reserved	Reserved
48	32	TMR0_INT	Timer 0 interrupt
49	33	TMR1_INT	Timer 1 interrupt
50	34	TMR2_INT	Timer 2 interrupt
51	35	TMR3_INT	Timer 3 interrupt
52	36	UART0_INT	UART0 interrupt
53	37	UART1_INT	UART1 interrupt
54	38	I2C0_INT	I2C0 interrupt
55	39	I2C1_INT	I2C1 interrupt
56	40	PDMA_INT	PDMA interrupt
57	41	Reserved	Reserved
58	42	EADC_INT	EADC interrupt source 0
59	43	EADC1_INT	EADC interrupt source 1
60	44	Reserved	Reserved
61	45	BPWM0	BPWM0 interrupt
62	46	EADC_INT2	EADC interrupt source 2
63	47	EADC_INT3	EADC interrupt source 3
64	48	UART2_INT	UART2 interrupt
65	49	UART3_INT	UART3 interrupt
66	50	USCI0	USCI0 interrupt
67	51	UART4_INT	UART4 interrupt
68	52	Reserved	Reserved
69	53	USBD_INT	USB device interrupt

70	54	Reserved	Reserved
71	55	Reserved	Reserved
72	56	Reserved	Reserved
73	57	Reserved	Reserved
74	58	Reserved	Reserved
75	59	Reserved	Reserved
76	60	Reserved	Reserved
77	61	Reserved	Reserved
78	62	Reserved	Reserved
79	63	Reserved	Reserved

Table 6.2-9 Interrupt Number Table

6.2.12.2 Operation Description

NVIC interrupts can be enabled and disabled by writing to their corresponding Interrupt Set-Enable or Interrupt Clear-Enable register bit-field. The registers use a write-1-to-enable and write-1-to-clear policy, both registers reading back the current enabled state of the corresponding interrupts. When an interrupt is disabled, interrupt assertion will cause the interrupt to become Pending, however, the interrupt will not activate. If an interrupt is Active when it is disabled, it remains in its Active state until cleared by reset or an exception return. Clearing the enable bit prevents new activations of the associated interrupt.

NVIC interrupts can be pended/un-pended using a complementary pair of registers to those used to enable/disable the interrupts, named the Set-Pending Register and Clear-Pending Register respectively. The registers use a write-1-to-enable and write-1-to-clear policy, both registers reading back the current pended state of the corresponding interrupts. The Clear-Pending Register has no effect on the execution status of an Active interrupt.

NVIC interrupts are prioritized by updating an 8-bit field within a 32-bit register (each register supporting four interrupts).

The general registers associated with the NVIC are all accessible from a block of memory in the System Control Space and will be described in the next section.

6.2.12.3 NVIC Control Registers

R: read only, W: write only, R/W: both read and write

Register	Offset	R/W	Description	Reset Value
NVIC Base Address: NVIC_BA = 0xE000_E100				
NVIC_ISER0	NVIC_BA+0x000	R/W	IRQ0 ~ IRQ31 Set-enable Control Register	0x0000_0000
NVIC_ISER1	NVIC_BA+0x004	R/W	IRQ32 ~ IRQ63 Set-enable Control Register	0x0000_0000
NVIC_ICER0	NVIC_BA+0x080	R/W	IRQ0 ~ IRQ31 Clear-enable Control Register	0x0000_0000
NVIC_ICER1	NVIC_BA+0x084	R/W	IRQ32 ~ IRQ63 Clear-enable Control Register	0x0000_0000
NVIC_ISPR0	NVIC_BA+0x100	R/W	IRQ0 ~ IRQ31 Set-pending Control Register	0x0000_0000
NVIC_ISPR1	NVIC_BA+0x104	R/W	IRQ32 ~ IRQ63 Set-pending Control Register	0x0000_0000
NVIC_ICPR0	NVIC_BA+0x180	R/W	IRQ0 ~ IRQ31 Clear-pending Control Register	0x0000_0000
NVIC_ICPR1	NVIC_BA+0x184	R/W	IRQ32 ~ IRQ63 Clear-pending Control Register	0x0000_0000
NVIC_IABR0	NVIC_BA+0x200	R/W	IRQ0 ~ IRQ31 Active Bit Register	0x0000_0000
NVIC_IABR1	NVIC_BA+0x204	R/W	IRQ32 ~ IRQ63 Active Bit Register	0x0000_0000
NVIC_IPRn n=0,1..15	0xE000E400 +0x4*n	R/W	IRQ0 ~ IRQ63 Priority Control Register	0x0000_0000

IRQ0 ~ IRQ31 Set-enable Control Register (NVIC_ISER0)

Register	Offset	R/W	Description	Reset Value
NVIC_ISER0	NVIC_BA+0x000	R/W	IRQ0 ~ IRQ31 Set-enable Control Register	0x0000_0000

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

Bits	Description
[31:0]	SETENA Interrupt Set Enable Bit The NVIC_ISER0-NVIC_ISER1 registers enable interrupts, and show which interrupts are enabled Write Operation: 0 = No effect. 1 = Interrupt Enabled. Read Operation: 0 = Interrupt Disabled. 1 = Interrupt Enabled.

IRQ31 ~ IRQ63 Set-enable Control Register (NVIC_ISER1)

Register	Offset	R/W	Description	Reset Value
NVIC_ISER1	NVIC_BA+0x004	R/W	IRQ32 ~ IRQ63 Set-enable Control Register	0x0000_0000

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

Bits	Description
[31:0]	SETENA Interrupt Set Enable Bit The NVIC_ISER0-NVIC_ISER2 registers enable interrupts, and show which interrupts are enabled Write Operation: 0 = No effect. 1 = Interrupt Enabled. Read Operation: 0 = Interrupt Disabled. 1 = Interrupt Enabled.

IRQ0 ~ IRQ31 Clear-enable Control Register (NVIC_ICER0)

Register	Offset	R/W	Description	Reset Value
NVIC_ICER0	NVIC_BA+0x080	R/W	IRQ0 ~ IRQ31 Clear-enable Control Register	0x0000_0000

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

Bits	Description
[31:0]	Interrupt Clear Enable Bit The NVIC_ICER0-NVIC_ICER1 registers disable interrupts, and show which interrupts are enabled. Write Operation: 0 = No effect. 1 = Interrupt Disabled. Read Operation: 0 = Interrupt Disabled. 1 = Interrupt Enabled.

IRQ32 ~ IRQ63 Clear-enable Control Register (NVIC_ICER1)

Register	Offset	R/W	Description	Reset Value
NVIC_ICER1	NVIC_BA+0x084	R/W	IRQ32 ~ IRQ63 Clear-enable Control Register	0x0000_0000

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

Bits	Description
[31:0]	Interrupt Clear Enable Bit The NVIC_ICER0-NVIC_ICER2 registers disable interrupts, and show which interrupts are enabled. Write Operation: 0 = No effect. 1 = Interrupt Disabled. Read Operation: 0 = Interrupt Disabled. 1 = Interrupt Enabled.

IRQ0 ~ IRQ31 Set-pending Control Register (NVIC_ISPR0)

Register	Offset	R/W	Description	Reset Value
NVIC_ISPR0	NVIC_BA+0x100	R/W	IRQ0 ~ IRQ31 Set-pending Control Register	0x0000_0000

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

Bits	Description
[31:0]	SETPEND Interrupt Set-pending The NVIC_ISPR0-NVIC_ISPR1 registers force interrupts into the pending state, and show which interrupts are pending Write Operation: 0 = No effect. 1 = Changes interrupt state to pending. Read Operation: 0 = Interrupt is not pending. 1 = Interrupt is pending.

IRQ32 ~ IRQ63 Set-pending Control Register (NVIC_ISPR1)

Register	Offset	R/W	Description	Reset Value
NVIC_ISPR1	NVIC_BA+0x104	R/W	IRQ32 ~ IRQ63 Set-pending Control Register	0x0000_0000

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

Bits	Description
[31:0]	SETPEND Interrupt Set-pending The NVIC_ISPR0-NVIC_ISPR1 registers force interrupts into the pending state, and show which interrupts are pending Write Operation: 0 = No effect. 1 = Changes interrupt state to pending. Read Operation: 0 = Interrupt is not pending. 1 = Interrupt is pending.

IRQ0 ~ IRQ31 Clear-pending Control Register (NVIC_ICPR0)

Register	Offset	R/W	Description	Reset Value
NVIC_ICPR0	NVIC_BA+0x180	R/W	IRQ0 ~ IRQ31 Clear-pending Control Register	0x0000_0000

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

Bits	Description
[31:0]	CALPEND Interrupt Clear-pending The NVIC_ICPR0-NVIC_ICPR1 registers remove the pending state from interrupts, and show which interrupts are pending Write Operation: 0 = No effect. 1 = Removes pending state an interrupt. Read Operation: 0 = Interrupt is not pending. 1 = Interrupt is pending.

IRQ32 ~ IRQ63 Clear-pending Control Register (NVIC_ICPR1)

Register	Offset	R/W	Description	Reset Value
NVIC_ICPR1	NVIC_BA+0x184	R/W	IRQ32 ~ IRQ63 Clear-pending Control Register	0x0000_0000

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

Bits	Description
[31:0]	CALPEND Interrupt Clear-pending The NVIC_ICPR0-NVIC_ICPR1 registers remove the pending state from interrupts, and show which interrupts are pending Write Operation: 0 = No effect. 1 = Removes pending state an interrupt. Read Operation: 0 = Interrupt is not pending. 1 = Interrupt is pending.

IRQ0 ~ IRQ31 Active Bit Register (NVIC_IABR0)

Register	Offset	R/W	Description	Reset Value
NVIC_IABR0	NVIC_BA+0x200	R/W	IRQ0 ~ IRQ31 Active Bit Register	0x0000_0000

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

Bits	Description	
[31:0]	ACTIVE	Interrupt Active Flags The NVIC_IABR0-NVIC_IABR1 registers indicate which interrupts are active. 0 = interrupt not active. 1 = interrupt active.

IRQ32 ~ IRQ63 Active Bit Register (NVIC_IABR1)

Register	Offset	R/W	Description	Reset Value
NVIC_IABR1	NVIC_BA+0x204	R/W	IRQ32 ~ IRQ63 Active Bit Register	0x0000_0000

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

Bits	Description	
[31:0]	ACTIVE	Interrupt Active Flags The NVIC_IABR0-NVIC_IABR1 registers indicate which interrupts are active. 0 = interrupt not active. 1 = interrupt active.

IRQ0 ~ IRQ71 Interrupt Priority Register (NVIC_IPRn)

Register	Offset	R/W	Description	Reset Value
NVIC_IPRn n=0,1..15	0xE000E400 +0x4*n	R/W	IRQ0 ~ IRQ63 Priority Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PRI_4n_3		Reserved					
23	22	21	20	19	18	17	16
PRI_4n_2		Reserved					
15	14	13	12	11	10	9	8
PRI_4n_1		Reserved					
7	6	5	4	3	2	1	0
PRI_4n_0		Reserved					

Bits	Description	
[31:30]	PRI_4n_3	Priority of IRQ_4n+3 "0" denotes the highest priority and "3" denotes the lowest priority
[29:24]	Reserved	Reserved.
[23:22]	PRI_4n_2	Priority of IRQ_4n+2 "0" denotes the highest priority and "3" denotes the lowest priority
[21:16]	Reserved	Reserved.
[15:14]	PRI_4n_1	Priority of IRQ_4n+1 "0" denotes the highest priority and "3" denotes the lowest priority
[13:8]	Reserved	Reserved.
[7:6]	PRI_4n_0	Priority of IRQ_4n+0 "0" denotes the highest priority and "3" denotes the lowest priority
[5:0]	Reserved	Reserved.

6.2.12.4 NMI Control Registers

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
NMI Base Address: NMI_BA = 0x4000_0300				
NMIEN	NMI_BA+0x00	R/W	NMI Source Interrupt Enable Register	0x0000_0000
NMISTS	NMI_BA+0x04	R	NMI Source Interrupt Status Register	0x0000_0000

NMI Source Interrupt Enable Register (NMIEN)

Register	Offset	R/W	Description	Reset Value
NMIEN	NMI_BA+0x00	R/W	NMI Source Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
UART1_INT	UART0_INT	EINT5	EINT4	EINT3	EINT2	EINT1	EINT0
7	6	5	4	3	2	1	0
Reserved	RTC_INT	Reserved	CLKFAIL	Reserved	PWRWU_INT	IRC_INT	BODOUT

Bits	Description
[31:16]	Reserved Reserved.
[15]	UART1_INT UART1 NMI Source Enable Bit (Write Protect) 0 = UART1 NMI source Disabled. 1 = UART1 NMI source Enabled. Note: This bit is write protected. Refer to the SYS_REGLCTL register.
[14]	UART0_INT UART0 NMI Source Enable Bit (Write Protect) 0 = UART0 NMI source Disabled. 1 = UART0 NMI source Enabled. Note: This bit is write protected. Refer to the SYS_REGLCTL register.
[13]	EINT5 External Interrupt From PF.0 Pin NMI Source Enable Bit (Write Protect) 0 = External interrupt from PF.0 pin NMI source Disabled. 1 = External interrupt from PF.0 pin NMI source Enabled. Note: This bit is write protected. Refer to the SYS_REGLCTL register.
[12]	EINT4 External Interrupt From PE.0 Pin NMI Source Enable Bit (Write Protect) 0 = External interrupt from PE.0 pin NMI source Disabled. 1 = External interrupt from PE.0 pin NMI source Enabled. Note: This bit is write protected. Refer to the SYS_REGLCTL register.
[11]	EINT3 External Interrupt From PD.0 Pin NMI Source Enable Bit (Write Protect) 0 = External interrupt from PD.0 pin NMI source Disabled. 1 = External interrupt from PD.0 pin NMI source Enabled. Note: This bit is write protected. Refer to the SYS_REGLCTL register.
[10]	EINT2 External Interrupt From PC.0 Pin NMI Source Enable Bit (Write Protect) 0 = External interrupt from PC.0 pin NMI source Disabled. 1 = External interrupt from PC.0 pin NMI source Enabled. Note: This bit is write protected. Refer to the SYS_REGLCTL register.
[9]	EINT1 External Interrupt From PB.0, PD.3 or PE.5 Pin NMI Source Enable Bit (Write Protect)

		0 = External interrupt from PB.0, PD.3 or PE.5 pin NMI source Disabled. 1 = External interrupt from PB.0, PD.3 or PE.5 pin NMI source Enabled. Note: This bit is write protected. Refer to the SYS_REGLCTL register.
[8]	EINT0	External Interrupt From PA.0, PD.2 or PE.4 Pin NMI Source Enable Bit (Write Protect) 0 = External interrupt from PA.0, PD.2 or PE.4 pin NMI source Disabled. 1 = External interrupt from PA.0, PD.2 or PE.4 pin NMI source Enabled. Note: This bit is write protected. Refer to the SYS_REGLCTL register.
[7]	Reserved	Reserved.
[6]	RTC_INT	RTC NMI Source Enable Bit (Write Protect) 0 = RTC NMI source Disabled. 1 = RTC NMI source Enabled. Note: This bit is write protected. Refer to the SYS_REGLCTL register.
[5]	Reserved	Reserved.
[4]	CLKFAIL	Clock Fail Detected NMI Source Enable Bit (Write Protect) 0 = Clock fail detected interrupt NMI source Disabled. 1 = Clock fail detected interrupt NMI source Enabled. Note: This bit is write protected. Refer to the SYS_REGLCTL register.
[3]	Reserved	Reserved.
[2]	PWRWU_INT	Power-down Mode Wake-up NMI Source Enable Bit (Write Protect) 0 = Power-down mode wake-up NMI source Disabled. 1 = Power-down mode wake-up NMI source Enabled. Note: This bit is write protected. Refer to the SYS_REGLCTL register.
[1]	IRC_INT	IRC TRIM NMI Source Enable Bit (Write Protect) 0 = IRC TRIM NMI source Disabled. 1 = IRC TRIM NMI source Enabled. Note: This bit is write protected. Refer to the SYS_REGLCTL register.
[0]	BODOUT	BOD NMI Source Enable Bit (Write Protect) 0 = BOD NMI source Disabled. 1 = BOD NMI source Enabled. Note: This bit is write protected. Refer to the SYS_REGLCTL register.

NMI Source Interrupt Status Register (NMISTS)

Register	Offset	R/W	Description	Reset Value
NMISTS	NMI_BA+0x04	R	NMI Source Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
UART1_INT	UART0_INT	EINT5	EINT4	EINT3	EINT2	EINT1	EINT0
7	6	5	4	3	2	1	0
Reserved	RTC_INT	Reserved	CLKFAIL	Reserved	PWRWU_INT	IRC_INT	BODOUT

Bits	Description
[31:16]	Reserved Reserved.
[15]	UART1_INT UART1 Interrupt Flag (Read Only) 0 = UART1 interrupt is deasserted. 1 = UART1 interrupt is asserted.
[14]	UART0_INT UART0 Interrupt Flag (Read Only) 0 = UART1 interrupt is deasserted. 1 = UART1 interrupt is asserted.
[13]	EINT5 External Interrupt From PF.0 Pin Interrupt Flag (Read Only) 0 = External Interrupt from PF.0 interrupt is deasserted. 1 = External Interrupt from PF.0 interrupt is asserted.
[12]	EINT4 External Interrupt From PE.0 Pin Interrupt Flag (Read Only) 0 = External Interrupt from PE.0 interrupt is deasserted. 1 = External Interrupt from PE.0 interrupt is asserted.
[11]	EINT3 External Interrupt From PD.0 Pin Interrupt Flag (Read Only) 0 = External Interrupt from PD.0 interrupt is deasserted. 1 = External Interrupt from PD.0 interrupt is asserted.
[10]	EINT2 External Interrupt From PC.0 Pin Interrupt Flag (Read Only) 0 = External Interrupt from PC.0 interrupt is deasserted. 1 = External Interrupt from PC.0 interrupt is asserted.
[9]	EINT1 External Interrupt From PB.0, PD.3 or PE.5 Pin Interrupt Flag (Read Only) 0 = External Interrupt from PB.0, PD.3 or PE.5 interrupt is deasserted. 1 = External Interrupt from PB.0, PD.3 or PE.5 interrupt is asserted.
[8]	EINT0 External Interrupt From PA.0, PD.2 or PE.4 Pin Interrupt Flag (Read Only) 0 = External Interrupt from PA.0, PD.2 or PE.4 interrupt is deasserted. 1 = External Interrupt from PA.0, PD.2 or PE.4 interrupt is asserted.
[7]	Reserved Reserved.

[6]	RTC_INT	RTC Interrupt Flag (Read Only) 0 = RTC interrupt is deasserted. 1 = RTC interrupt is asserted.
[5]	Reserved	Reserved.
[4]	CLKFAIL	Clock Fail Detected Interrupt Flag (Read Only) 0 = Clock fail detected interrupt is deasserted. 1 = Clock fail detected interrupt is asserted.
[3]	Reserved	Reserved.
[2]	PWRWU_INT	Power-down Mode Wake-up Interrupt Flag (Read Only) 0 = Power-down mode wake-up interrupt is deasserted. 1 = Power-down mode wake-up interrupt is asserted.
[1]	IRC_INT	IRC TRIM Interrupt Flag (Read Only) 0 = HIRC TRIM interrupt is deasserted. 1 = HIRC TRIM interrupt is asserted.
[0]	BODOUT	BOD Interrupt Flag (Read Only) 0 = BOD interrupt is deasserted. 1 = BOD interrupt is asserted.

6.2.13 System Control Register

The Cortex®-M23 status and operation mode control are managed by System Control Registers. Including CPUID, Cortex®-M23 interrupt priority and Cortex®-M23 power management can be controlled through these system control registers.

For more detailed information, please refer to the “Arm® Cortex®-M23 Technical Reference Manual” and “Arm® v6-M Architecture Reference Manual”.

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
SCR Base Address: SCS_BA = 0xE000_E000				
ICSR	SCS_BA+0xD04	R/W	Interrupt Control and State Register	0x0000_0000
VTOR	SCS_BA+0xD08	R/W	Vector Table Offset Register	0x0000_0000
AIRCR	SCS_BA+0xD0C	R/W	Application Interrupt and Reset Control Register	0xFA05_2000
SCR	SCS_BA+0xD10	R/W	System Control Register	0x0000_0000
CCR	SCS_BA+0xD14	R/W	Configuration and Control Register	0x0000_0209
SHPR2	SCS_BA+0xD1C	R/W	System Handler Priority Register 2	0x0000_0000
SHPR3	SCS_BA+0xD20	R/W	System Handler Priority Register 3	0x0000_0000
SHCSR	SCS_BA+0xD24	R/W	System Handler Control and State Register	0x0000_0000

Interrupt Control State Register (ICSR)

Register	Offset	R/W	Description	Reset Value
ICSR	SCS_BA+0xD04	R/W	Interrupt Control and State Register	0x0000_0000

31	30	29	28	27	26	25	24
NMIPENDSET	NMIPENDCLR	Reserved	PENDSVSET	PENDSVCLR	PENDSTSET	PENDSTCLR	Reserved
23	22	21	20	19	18	17	16
ISRPREEMPT	ISR_PENDING	Reserved		VECTPENDING			
15	14	13	12	11	10	9	8
VECTPENDING				Reserved			
7	6	5	4	3	2	1	0
VECTACTIVE							

Bits	Description	
[31]	NMIPENDSET	NMI Set-pending Bit Write Operation: 0 = No effect. 1 = Changes NMI exception state to pending. Read Operation: 0 = NMI exception is not pending. 1 = NMI exception is pending. Note: If AIRCR.BFHFNMINS is 0, this bit is RAZ/WI from Non-secure state.
[30]	NMIPENDCLR	NMI Bit-pending Bit 0 = No effect. 1 = Clear pending status. Note: If AIRCR.BFHFNMINS is 0, this bit is RAZ/WI from Non-secure state.
[29]	Reserved	Reserved.
[28]	PENDSVSET	PendSV Set-pending Bit Write Operation: 0 = No effect. 1 = Changes PendSV exception state to pending. Read Operation: 0 = PendSV exception is not pending. 1 = PendSV exception is pending. Note: Writing 1 to this bit is the only way to set the PendSV exception state to pending.
[27]	PENDSVCLR	PendSV Clear-pending Bit Write Operation: 0 = No effect. 1 = Removes the pending state from the PendSV exception. Note: This is a write only bit. To clear the PENDSV bit, you must "write 0 to PENDSVSET and write 1 to PENDSVCLR" at the same time.

[26]	PENDSTSET	SysTick Exception Set-pending Bit Write Operation: 0 = No effect. 1 = Changes SysTick exception state to pending. Read Operation: 0 = SysTick exception is not pending. 1 = SysTick exception is pending.
[25]	PENDSTCLR	SysTick Exception Clear-pending Bit Write Operation: 0 = No effect. 1 = Removes the pending state from the SysTick exception. Note: This is a write only bit. To clear the PENDST bit, you must "write 0 to PENDSTSET and write 1 to PENDSTCLR" at the same time.
[24]	Reserved	Reserved.
[23]	ISRPREEMPT	Interrupt Preempt Bit (Read Only) If set, a pending exception will be serviced on exit from the debug halt state.
[22]	ISRPENDING	Interrupt Pending Flag, Excluding NMI and Faults (Read Only) 0 = Interrupt not pending. 1 = Interrupt pending.
[21:20]	Reserved	Reserved.
[19:12]	VECTPENDING	Number of the Highest Pended Exception (Read Only) 0 = no pending exceptions. Nonzero = the exception number of the highest priority pending enabled exception.
[11:8]	Reserved	Reserved.
[7:0]	VECTACTIVE	Number of the Current Active Exception (Read Only) 0 = Thread mode. Non-zero = The exception number of the currently active exception.

Vector Table Offset Register (VTOR)

Register	Offset	R/W	Description	Reset Value
VTOR	SCS_BA+0xD08	R/W	Vector Table Offset Register	0x0000_0000

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

Bits	Description	
[31:9]	TBLOFF	Table Offset Bits The vector table address for the selected Security state.
[8:0]	Reserved	Reserved.

Application Interrupt and Reset Control Register (AIRCR)

Register	Offset	R/W	Description	Reset Value
AIRCR	SCS_BA+0xD0C	R/W	Application Interrupt and Reset Control Register	0xFA05_2000

31	30	29	28	27	26	25	24
VECTORKEY							
23	22	21	20	19	18	17	16
VECTORKEY							
15	14	13	12	11	10	9	8
ENDIANNESS	PRIS	Reserved					
7	6	5	4	3	2	1	0
Reserved				SYSRESETREQS	SYSRESETREQ	VECTCLRACTIVE	Reserved

Bits	Description
[31:16]	VECTORKEY Register Access Key When writing this register, this field should be 0x05FA, otherwise the write action will be ignored. The VECTORKEY field is used to prevent accidental write to this register from resetting the system or clearing of the exception status.
[15]	ENDIANNESS Data Endianness (Read Only) 0 = Little-endian. 1 = Big-endian.
[14]	PRIS Priority Secure Exceptions Bit 0 = Priority ranges of Secure and Non-secure exceptions are identical. 1 = Non-secure exceptions are de-prioritized.
[13:4]	Reserved Reserved.
[3]	SYSRESETREQS System Reset Request Secure Only Bit 0 = SYSRESETREQ functionality is available to both security states. 1 = SYSRESETREQ functionality is available to secure state only.
[2]	SYSRESETREQ System Reset Request Bit Writing This Bit to 1 Will Cause A Reset Signal To Be Asserted To The Chip And Indicate A Reset Is Requested This bit is write only and self-cleared as part of the reset sequence.
[1]	VECTCLRACTIVE Exception Active Status Clear Bit Setting This Bit To 1 Will Clears All Active State Information For Fixed And Configurable Exceptions This bit is write only and can only be written when the core is halted. Note: It is the debugger's responsibility to re-initialize the stack. Note: This bit reads as zero.
[0]	Reserved Reserved.

System Control Register (SCR)

Register	Offset	R/W	Description	Reset Value
SCR	SCS_BA+0xD10	R/W	System Control Register	0x0000_0000

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

Bits	Description
[31:5]	Reserved Reserved.
[4]	SEVONPEND Send Event on Pending 0 = Only enabled interrupts or events can wake up the processor, while disabled interrupts are excluded. 1 = Enabled events and all interrupts, including disabled interrupts, can wake up the processor. When an event or interrupt enters pending state, the event signal wakes up the processor from WFE. If the processor is not waiting for an event, the event is registered and affects the next WFE. The processor also wakes up on execution of an SEV instruction or an external event.
[3]	Reserved Reserved.
[2]	SLEEPDEEP Processor Deep Sleep and Sleep Mode Selection Control Whether the Processor Uses Sleep Or Deep Sleep as its Low Power Mode. 0 = Sleep. 1 = Deep sleep.
[1]	SLEEPONEXIT Sleep-on-exit Enable Control This bit indicate Sleep-On-Exit when Returning from Handler Mode to Thread Mode. 0 = Do not sleep when returning to Thread mode. 1 = Enter sleep, or deep sleep, on return from an ISR to Thread mode. Note: Setting this bit to 1 enables an interrupt driven application to avoid returning to an empty main application.
[0]	Reserved Reserved.

System Handler Priority Register 2 (SHPR2)

Register	Offset	R/W	Description	Reset Value
SHPR2	SCS_BA+0xD1C	R/W	System Handler Priority Register 2	0x0000_0000

31	30	29	28	27	26	25	24
PRI_11		Reserved					
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							

Bits	Description	
[31:30]	PRI_11	Priority of System Handler 11 – SVCall “0” denotes the highest priority and “3” denotes the lowest priority.
[29:0]	Reserved	Reserved.

System Handler Priority Register 3 (SHPR3)

Register	Offset	R/W	Description	Reset Value
SHPR3	SCS_BA+0xD20	R/W	System Handler Priority Register 3	0x0000_0000

31	30	29	28	27	26	25	24
PRI_15		Reserved					
23	22	21	20	19	18	17	16
PRI_14		Reserved					
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							

Bits	Description	
[31:30]	PRI_15	Priority of System Handler 15 – SysTick “0” denotes the highest priority and “3” denotes the lowest priority.
[29:24]	Reserved	Reserved.
[23:22]	PRI_14	Priority of System Handler 14 – PendSV “0” denotes the highest priority and “3” denotes the lowest priority.
[21:0]	Reserved	Reserved.

System Handler Control and State Register (SHCSR)

Register	Offset	R/W	Description	Reset Value
SHCSR	SCS_BA+0xD24	R/W	System Handler Control and State Register	0x0000_0000

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

Bits	Description	
[31:22]	Reserved	Reserved.
[21]	HARDFaultPENDED	HardFault Exception Pending State This bit indicates and allows modification of the pending state of the HardFault exception corresponding to the selected Security state. This bit is banked between Security states. The possible values of this bit are: 0 = HardFault exception not pending for the selected Security state. 1 = HardFault exception pending for the selected Security state.
[20:0]	Reserved	Reserved.

6.3 Clock Controller

6.3.1 Overview

The clock controller generates clocks for the whole chip, including system clocks and all peripheral clocks. The clock controller also implements the power control function with the individually clock ON/OFF control, clock source selection and a clock divider. The chip will not enter Power-down mode until CPU sets the Power-down enable bit PDEN(CLK_PWRCTL[7]) and Cortex®-M23 core executes the WFI instruction. After that, chip enters Power-down mode and waits for wake-up interrupt source triggered to leave Power-down mode. In Power-down mode, the clock controller turns off the 4~32 MHz external high speed crystal (HXT), 48 MHz internal high speed RC oscillator (HIRC) and 4 MHz internal median speed RC oscillator (MIRC) to reduce the overall system power consumption. Figure 6.3-1 shows the clock generator and the overview of the clock source control.

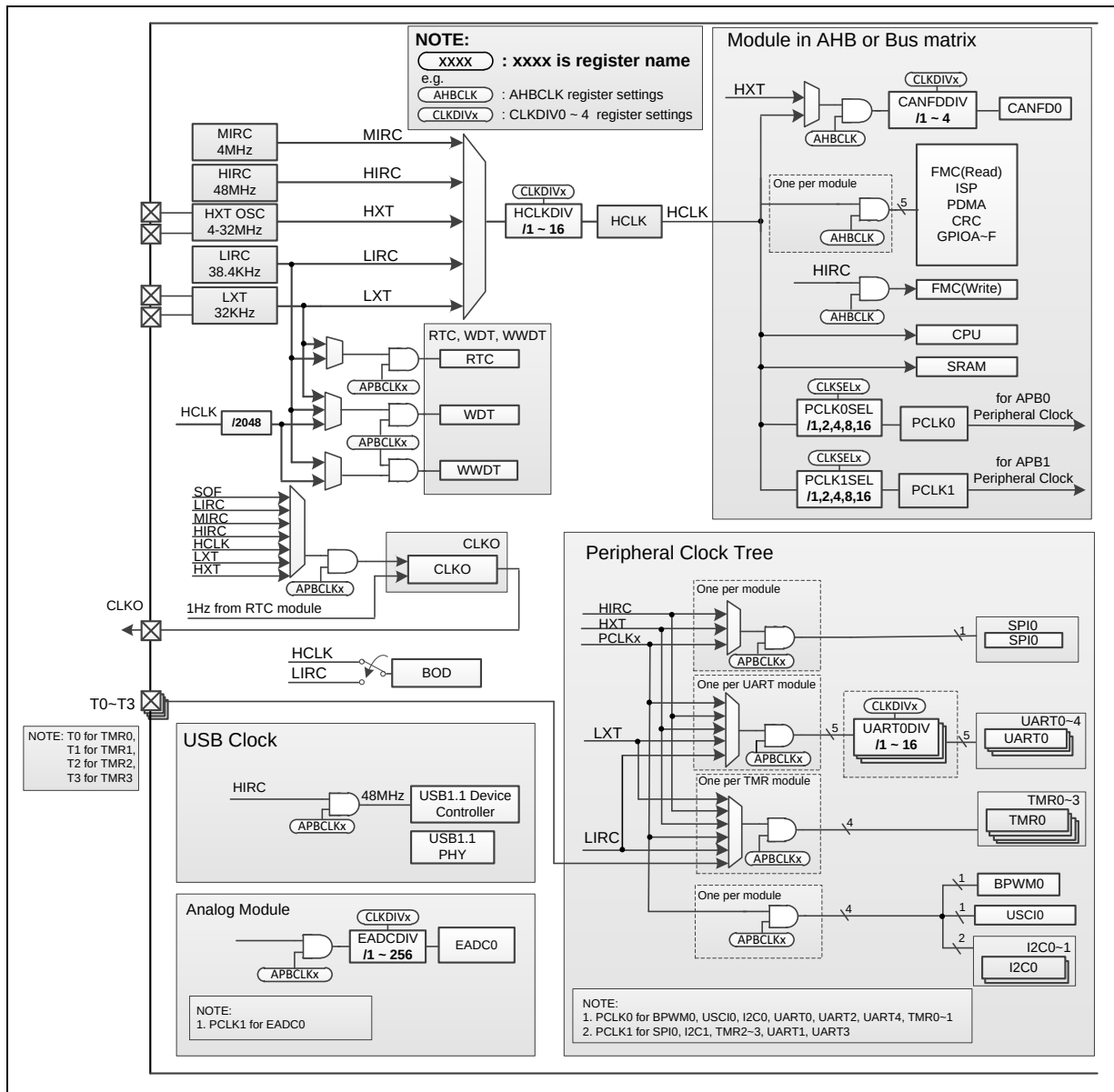


Figure 6.3-1 Clock Generator Global View Diagram

6.3.2 Clock Generator

The clock generator consists of 5 clock sources, which are listed below:

- 32.768 kHz external low speed crystal oscillator (LXT)
- 4~32 MHz external high speed crystal oscillator (HXT)
- 48 MHz internal high speed RC oscillator (HIRC)
- 38.4 kHz internal low speed RC oscillator (LIRC)
- 4 MHz internal medium speed oscillator (MIRC)

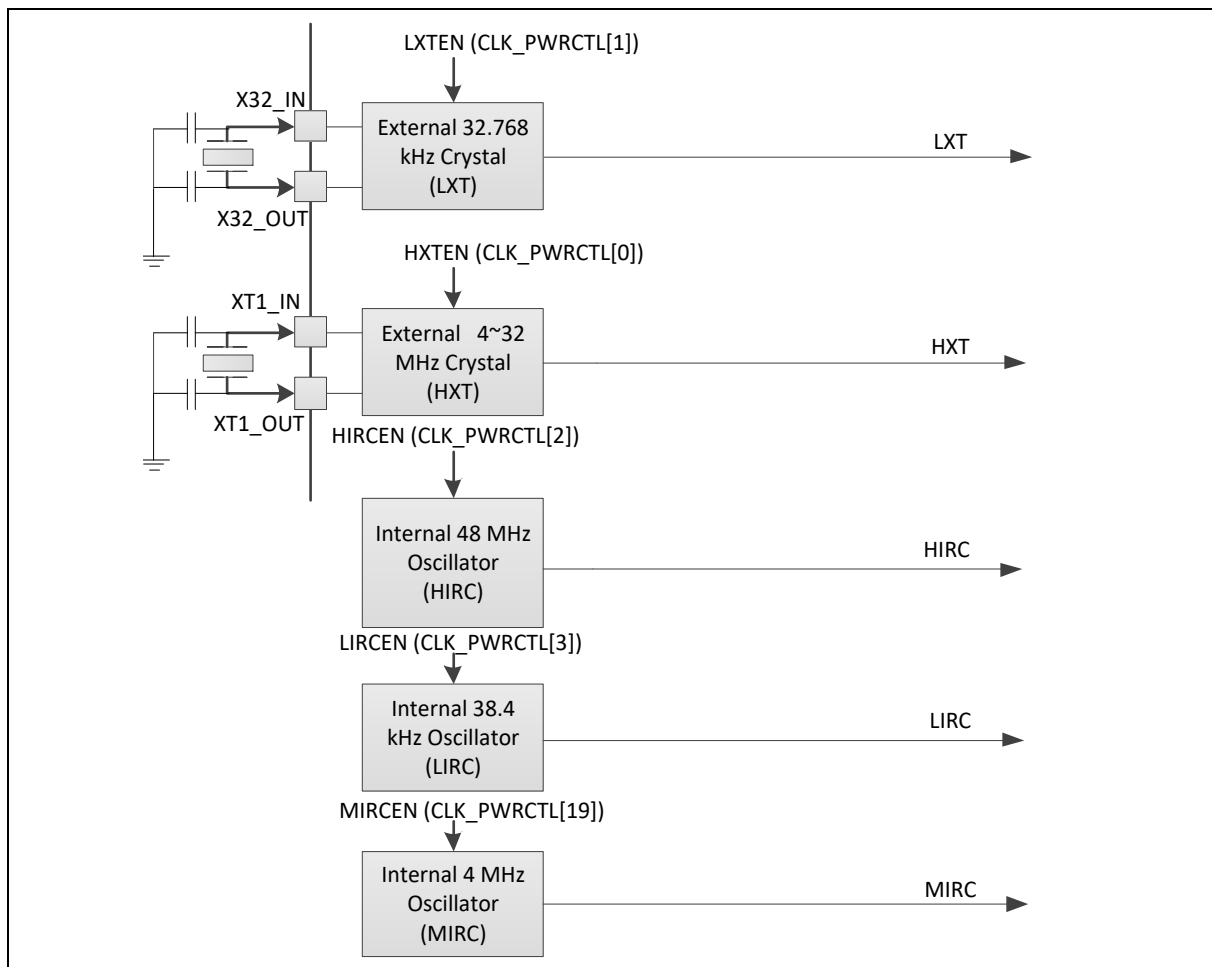


Figure 6.3-2 Clock Generator Block Diagram

6.3.3 System Clock and SysTick Clock

The system clock has 5 clock sources, which are generated from clock generator block. The clock source switch depends on the register HCLKSEL (CLK_CLKSEL0[2:0]). The block diagram is shown in Figure 6.3-3

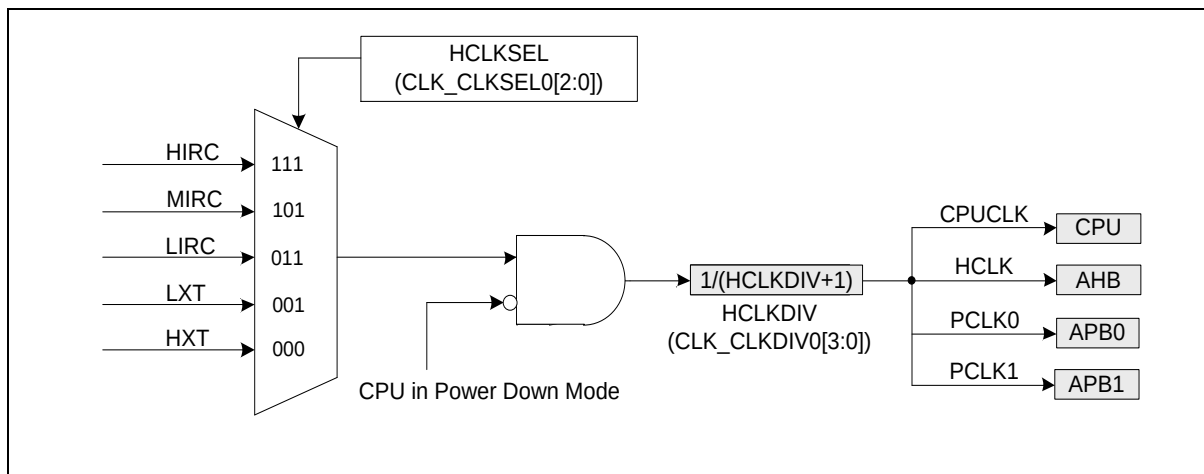


Figure 6.3-3 System Clock Block Diagram

There are two clock fail detectors to observe HXT and LXT clock source and they have individual enable and interrupt control. When HXT detector is enabled, the MIRC clock is enabled automatically. When LXT detector is enabled, the LIRC clock is enabled automatically.

When HXT clock detector is enabled, the system clock will auto switch to MIRC if HXT clock stop being detected on the following condition: system clock source comes from HXT. If HXT clock stop condition is detected, the HXTFIF (CLK_CLKDSTS[0]) is set to 1 and chip will enter interrupt if HXTFIE (CLK_CLKDCTL[5]) is set to 1. User can try to recover HXT by disable HXT and enable HXT again to check if the clock stable bit is set to 1 or not. If HXT clock stable bit is set to 1, it means HXT is recover to oscillate after re-enable action and user can switch system clock to HXT again.

The HXT clock stop detect and system clock switch to MIRC procedure is shown in Figure 6.3-4.

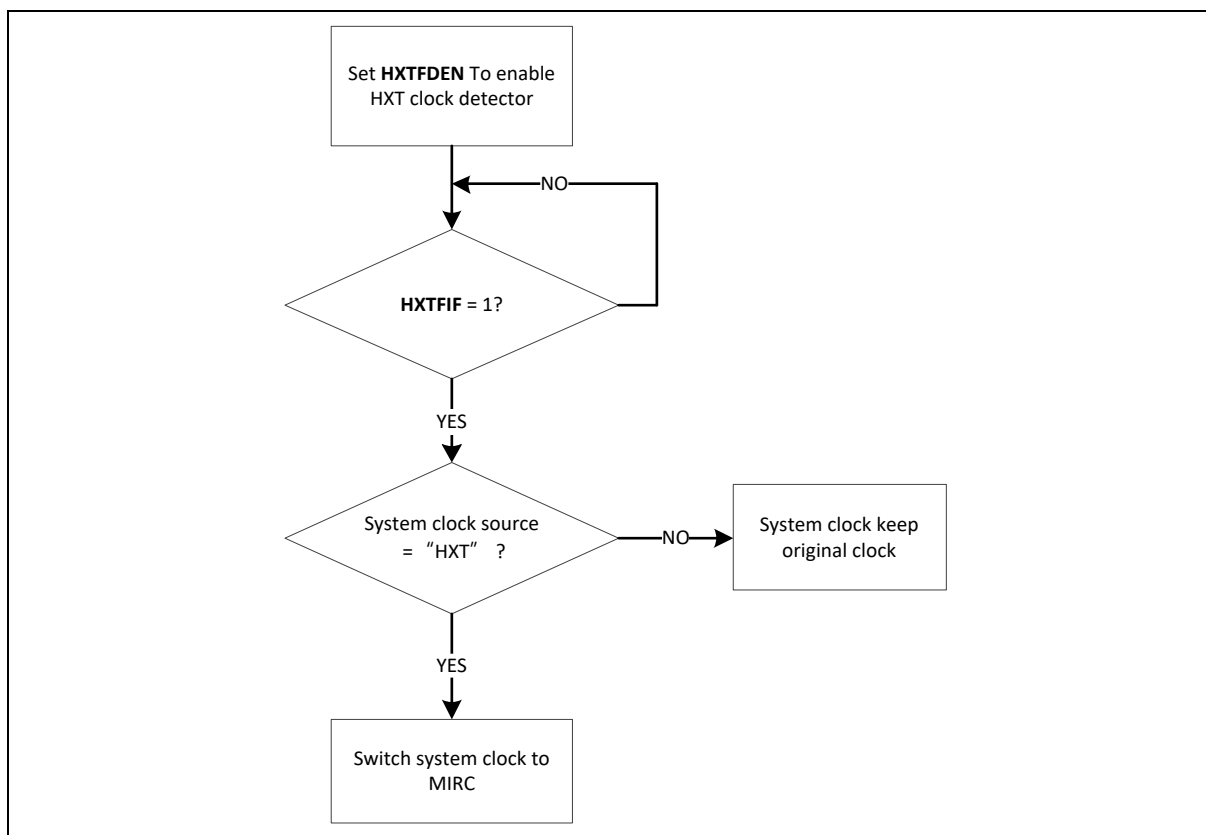


Figure 6.3-4 HXT Stop Protect Procedure

The clock source of SysTick in Cortex®-M23 core can use CPU clock or external clock (SYST_CTRL[2]). If using external clock, the SysTick clock (STCLK) has 5 clock sources. The clock source switch depends on the setting of the register STCLKSEL (CLK_CLKSEL0[5:3]). The block diagram is shown in Figure 6.3-5.

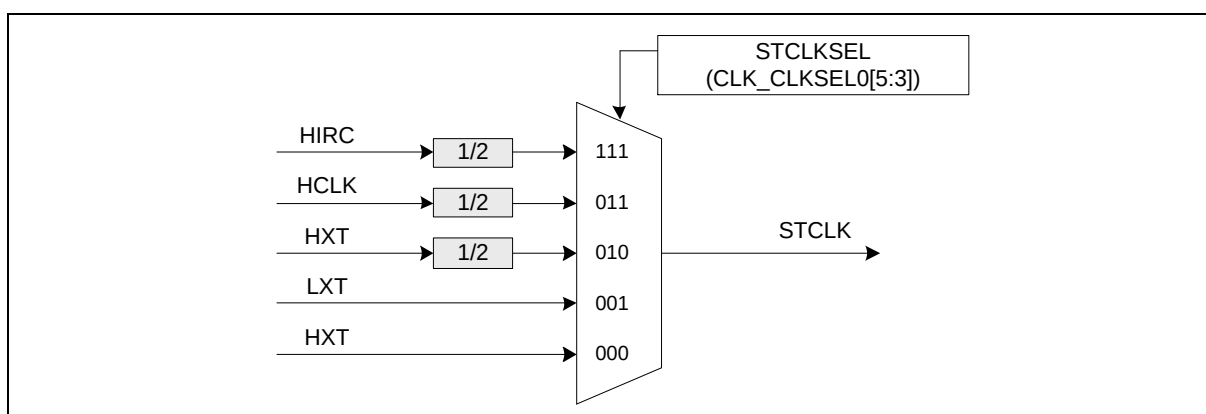


Figure 6.3-5 SysTick Clock Control Block Diagram

6.3.4 Peripherals Clock

The peripherals clock has different clock source switch setting, which depends on the different peripheral. Please refer to the CLK_CLKSEL1 and CLK_CLKSEL2 register description.

6.3.5 Power-down Mode Clock

For theses clocks, which still keep active, are listed below:

- Clock Generator
 - 38.4 kHz internal low speed RC oscillator (LIRC) clock
 - 32.768 kHz external low speed crystal oscillator (LXT) clock
 - 4 MHz internal medium speed oscillator (MIRC) clock if LCD and TK enabled.
- Peripherals Clock, except for HCLK, PCLK0 and PCLK1(When the modules adopt LXT or LIRC as clock source)

This device is equipped with a power-of-2 frequency divider which is composed by 16 chained divide-by-2 shift registers. One of the 16 shift register outputs selected by a sixteen to one multiplexer is reflected to CLK0 function pin. Therefore there are 16 options of power-of-2 divided clocks with the frequency from $F_{in}/2^1$ to $F_{in}/2^{16}$ where F_{in} is input clock frequency to the clock divider.

When writing 1 to DIV1EN (CLK_CLKOCTL[5]), the chained counter starts to count. When writing 0 to DIV1EN (CLK_CLKOCTL[5]), the chained counter continuously runs till divided clock reaches low state and stays in low state.

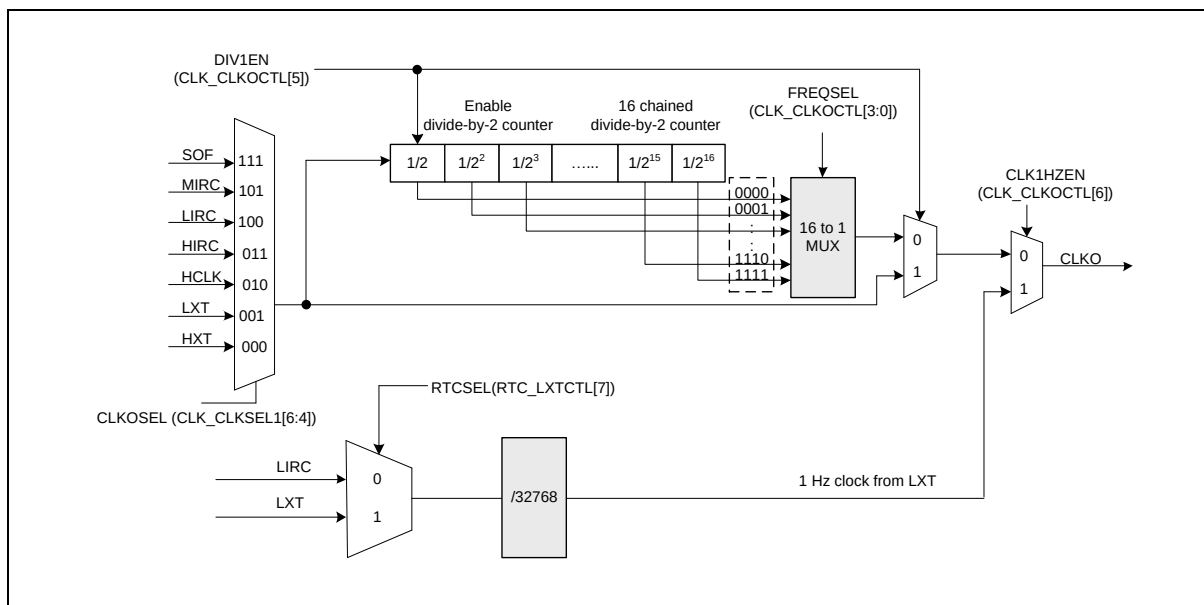


Figure 6.3-6 Clock Output Block Diagram

6.3.7 USB Clock Source

The clock source of USB 1.0 is generated from 48 MHz HIRC. The generated clocks are shown in Figure 6.3-7.

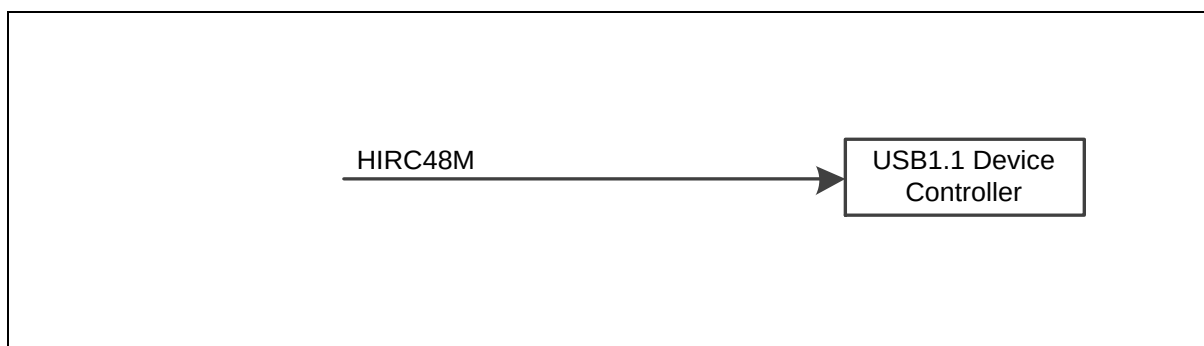


Figure 6.3-7 USB Clock Source

6.3.8 Register Map

R: read only, W: write only, R/W: both read and write

Register	Offset	R/W	Description	Reset Value
CLK Base Address: CLK_BA = 0x4000_0200				
CLK_PWRCTL	CLK_BA+0x00	R/W	System Power-down Control Register	0x0079_X01X
CLK_AHBCLK	CLK_BA+0x04	R/W	AHB Devices Clock Enable Control Register	0x0000_8004
CLK_APBCLK0	CLK_BA+0x08	R/W	APB Devices Clock Enable Control Register 0	0x0000_0001
CLK_APBCLK1	CLK_BA+0x0C	R/W	APB Devices Clock Enable Control Register 1	0x0000_0000
CLK_CLKSEL0	CLK_BA+0x10	R/W	Clock Source Select Control Register 0	0x0000_001d
CLK_CLKSEL1	CLK_BA+0x14	R/W	Clock Source Select Control Register 1	0x4422_222b
CLK_CLKSEL2	CLK_BA+0x18	R/W	Clock Source Select Control Register 2	0x0000_0020
CLK_CLKSEL3	CLK_BA+0x1C	R/W	Clock Source Select Control Register 3	0x4400_0040
CLK_CLKDIV0	CLK_BA+0x20	R/W	Clock Divider Number Register 0	0x0000_0000
CLK_CLKDIV4	CLK_BA+0x30	R/W	Clock Divider Number Register 4	0x0000_0000
CLK_PCLKDIV	CLK_BA+0x34	R/W	APB Clock Divider Register	0x0000_0000
CLK_STATUS	CLK_BA+0x50	R	Clock Status Monitor Register	0x0000_00XX
CLK_CLKOCTL	CLK_BA+0x60	R/W	Clock Output Control Register	0x0000_0000
CLK_CLKDCTL	CLK_BA+0x70	R/W	Clock Fail Detector Control Register	0x0000_0000
CLK_CLKDSTS	CLK_BA+0x74	R/W	Clock Fail Detector Status Register	0x0000_0000
CLK_CDUPB	CLK_BA+0x78	R/W	Clock Frequency Range Detector Upper Boundary Register	0x0000_0000
CLK_CDLOWB	CLK_BA+0x7c	R/W	Clock Frequency Range Detector Lower Boundary Register	0x0000_0000
CLK_PMUCTL	CLK_BA+0x90	R/W	Power Manager Control Register	0x0000_0000
CLK_HXTFSEL	CLK_BA+0xB4	R/W	HXT Filter Select Control Register	0x0000_000X

6.3.9 Register Description

System Power-down Control Register (CLK_PWRCTL)

Register	Offset	R/W	Description	Reset Value
CLK_PWRCTL	CLK_BA+0x00	R/W	System Power-down Control Register	0x0079_X01X

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved	HXTGAIN			MIRCEN	Reserved		
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
PDEN	PDWKIF	PDWKIEN	PDWKDLY	LIRCEN	HIRCEN	LXTEN	HXTEN

Bits	Description
[31:23]	Reserved Reserved.
[22:20]	HXTGAIN HXT Gain Control Bit (Write Protect) This is a protected register. Please refer to open lock sequence to program it. Gain control is used to enlarge the gain of crystal to make sure crystal work normally. If gain control is enabled, crystal will consume more power than gain control off. 000= HXT frequency 1~4 MHz. 001= HXT frequency 4~8 MHz. 010= HXT frequency 8~12 MHz. 011= HXT frequency 12~ 16 MHz. 100= HXT frequency 16~24 MHz. 101= HXT frequency 24~32 MHz. 110= HXT frequency 24~32 MHz. 111= HXT frequency 24~32 MHz. Note 1: This bit is write protected. Refer to the SYS_REGLCTL register. Note 2: Reset by power on reset.
[19]	MIRCEN MIRC Enable Bit (Write Protect) 0 = 4 MHz internal high speed RC oscillator (MIRC) Disabled. 1 = 4 MHz internal high speed RC oscillator (MIRC) Enabled. Note 1: This bit is write protected. Refer to the SYS_REGLCTL register. Note 2: This bit is forced on when CLKSEL0[2:0] = 101b.
[18:8]	Reserved Reserved.
[7]	PDEN System Power-down Enable (Write Protect) When this bit is set to 1, Power-down mode is enabled, chip enters Power-down mode immediately after the PDEN bit set. When chip wakes up from Power-down mode, this bit is auto cleared. Users need to set this bit again for next Power-down. In Power-down mode, HXT ,MIRC and HIRC will be disabled in this mode, but LXT and LIRC are not controlled by Power-down mode.

		In Power-down mode, the HCLK, PCLK0 and PCLK1 clocks are disabled, and ignored the clock source selection. The clocks of peripheral are not controlled by Power-down mode, if the peripheral clock source is from LXT or LIRC. 0 = Chip operating normally or chip in idle mode because of WFI command. 1 = Chip enters Power-down mode instantly or wait CPU sleep command WFI. Note: This bit is write protected. Refer to the SYS_REGLCTL register.
[6]	PDWKIF	Power-down Mode Wake-up Interrupt Status Set by "Power-down wake-up event" indicates that resume from Power-down mode" The flag is set if any wake-up source is occurred. Refer to Power Modes and Wake-up Sources section. Note 1: Write 1 to clear the bit to 0. Note 2: This bit works only if PDWKIEN (CLK_PWRCTL[5]) is set to 1.
[5]	PDWKIEN	Power-down Mode Wake-up Interrupt Enable Bit (Write Protect) 0 = Power-down mode wake-up interrupt Disabled. 1 = Power-down mode wake-up interrupt Enabled. Note 1: The interrupt will occur when both PDWKIF and PDWKIEN are high. Note 2: This bit is write protected. Refer to the SYS_REGLCTL register.
[4]	PDWKDLY	Enable the Wake-up Delay Counter (Write Protect) When the chip wakes up from Power-down mode, the clock control will delay certain clock cycles to wait system clock stable. The delayed clock cycle is 4096 clock cycles when chip works at 4~32 MHz external high speed crystal oscillator (HXT), The delayed clock cycle is 512 clock cycles when chip works at 48 MHz internal high speed RC oscillator (HIRC) The delayed clock cycle is 32 clock cycles when chip works at 4 MHz internal median speed RC oscillator (MIRC) 0 = Clock cycles delay Disabled. 1 = Clock cycles delay Enabled. Note: This bit is write protected. Refer to the SYS_REGLCTL register.
[3]	LIRCEN	LIRC Enable Bit (Write Protect) 0 = 38.4 kHz internal low speed RC oscillator (LIRC) Disabled. 1 = 38.4 kHz internal low speed RC oscillator (LIRC) Enabled. Note 1: This bit is write protected. Refer to the SYS_REGLCTL register. Note 2: LIRC will also be forced on when 1. Power down and ~(CONFIG0[3] & CONFIG0[4] & ~CONFIG0[31] & CONFIG0[30]) 2. Not power down and ~(CONFIG0[3] & CONFIG0[4] & CONFIG0[31])
[2]	HIRCEN	HIRC Enable Bit (Write Protect) 0 = 48 MHz internal high speed RC oscillator (HIRC) Disabled. 1 = 48 MHz internal high speed RC oscillator (HIRC) Enabled. Note: This bit is write protected. Refer to the SYS_REGLCTL register.
[1]	LXTEN	LXT Enable Bit (Write Protect) 0 = 32.768 kHz external low speed crystal (LXT) Disabled. 1 = 32.768 kHz external low speed crystal (LXT) Enabled. Note 1: This bit is write protected. Refer to the SYS_REGLCTL register. Note 2: Reset by RTC powr on reset.
[0]	HXTEN	HXT Enable Bit (Write Protect) 0 = 4~32 MHz external high speed crystal (HXT) Disabled. 1 = 4~32 MHz external high speed crystal (HXT) Enabled. Note 1: This bit is write protected. Refer to the SYS_REGLCTL register. Note 2: Reset by powr on reset.

AHB Devices Clock Enable Control Register (CLK_AHBCLK)

The bits in this register are used to enable/disable clock for system clock, AHB bus devices clock.

Register	Offset	R/W	Description	Reset Value
CLK_AHBCLK	CLK_BA+0x04	R/W	AHB Devices Clock Enable Control Register	0x0000_8004

31	30	29	28	27	26	25	24
Reserved		GPFCEN	Reserved		GPCCKEN	GPBCKEN	GPACKEN
23	22	21	20	19	18	17	16
CANFD0CKEN	Reserved						
15	14	13	12	11	10	9	8
FMCIDLE	Reserved						
7	6	5	4	3	2	1	0
CRCCKEN	Reserved		EXSTCKEN	Reserved	ISPCKEN	PDMACKEN	Reserved

Bits	Description
[31:30]	Reserved Reserved.
[29]	GPFCEN GPIOF Clock Enable Bit 0 = GPIOF port clock Disabled. 1 = GPIOF port clock Enabled.
[28:27]	Reserved Reserved.
[26]	GPCCKEN GPIOC Clock Enable Bit 0 = GPIOC port clock Disabled. 1 = GPIOC port clock Enabled.
[25]	GPBCKEN GPIOB Clock Enable Bit 0 = GPIOB port clock Disabled. 1 = GPIOB port clock Enabled.
[24]	GPACKEN GPIOA Clock Enable Bit 0 = GPIOA port clock Disabled. 1 = GPIOA port clock Enabled.
[23]	CANFD0CKEN CANFD0 Clock Enable Bit 0 = CANFD0 clock Disabled. 1 = CANFD0 clock Enabled.
[22:16]	Reserved Reserved.
[15]	FMCIDLE Flash Memory Controller Clock Enable Bit in IDLE Mode 0 = FMC clock Disabled when chip is under IDLE mode; in this case, PDMA cannot access FMC memory. 1 = FMC clock Enabled when chip is under IDLE mode; PDMA can access FMC memory.
[14:8]	Reserved Reserved.
[7]	CRCCKEN CRC Generator Controller Clock Enable Bit

		0 = CRC peripheral clock Disabled. 1 = CRC peripheral clock Enabled.
[6:5]	Reserved	Reserved.
[4]	EXSTCKEN	External System Tick Clock Enable Bit 0 = External System tick clock Disabled. 1 = External System tick clock Enabled.
[3]	Reserved	Reserved.
[2]	ISPCKEN	Flash ISP Controller Clock Enable Bit 0 = Flash ISP peripheral clock Disabled. 1 = Flash ISP peripheral clock Enabled.
[1]	PDMACKEN	PDMA Controller Clock Enable Bit 0 = PDMA peripheral clock Disabled. 1 = PDMA peripheral clock Enabled.
[0]	Reserved	Reserved.

APB Devices Clock Enable Control Register 0 (CLK_APBCLK0)

The bits in this register are used to enable/disable clock for peripheral controller clocks.

Register	Offset	R/W	Description	Reset Value
CLK_APBCLK0	CLK_BA+0x08	R/W	APB Devices Clock Enable Control Register 0	0x0000_0001

31	30	29	28	27	26	25	24
Reserved			EADCCKEN	USBDCKEN	Reserved		
23	22	21	20	19	18	17	16
Reserved			UART4CKEN	UART3CKEN	UART2CKEN	UART1CKEN	UART0CKEN
15	14	13	12	11	10	9	8
Reserved		SPI0CKEN	Reserved			I2C1CKEN	I2C0CKEN
7	6	5	4	3	2	1	0
Reserved	CLKOCKEN	TMR3CKEN	TMR2CKEN	TMR1CKEN	TMR0CKEN	RTCKEN	WDTCKEN

Bits	Description
[31:29]	Reserved
[28]	EADCCKEN Enhanced Analog-digital-converter (EADC) Clock Enable Bit 0 = EADC clock Disabled. 1 = EADC clock Enabled.
[27]	USBDCKEN USB Device Clock Enable Bit 0 = USB Device clock Disabled. 1 = USB Device clock Enabled.
[26:21]	Reserved
[20]	UART4CKEN UART4 Clock Enable Bit 0 = UART4 clock Disabled. 1 = UART4 clock Enabled.
[19]	UART3CKEN UART3 Clock Enable Bit 0 = UART3 clock Disabled. 1 = UART3 clock Enabled.
[18]	UART2CKEN UART2 Clock Enable Bit 0 = UART2 clock Disabled. 1 = UART2 clock Enabled.
[17]	UART1CKEN UART1 Clock Enable Bit 0 = UART1 clock Disabled. 1 = UART1 clock Enabled.
[16]	UART0CKEN UART0 Clock Enable Bit 0 = UART0 clock Disabled. 1 = UART0 clock Enabled.
[15:14]	Reserved

[13]	SPI0CKEN	SPI0 Clock Enable Bit 0 = SPI0 clock Disabled. 1 = SPI0 clock Enabled.
[12:10]	Reserved	Reserved.
[9]	I2C1CKEN	I2C1 Clock Enable Bit 0 = I2C1 clock Disabled. 1 = I2C1 clock Enabled.
[8]	I2C0CKEN	I2C0 Clock Enable Bit 0 = I2C0 clock Disabled. 1 = I2C0 clock Enabled.
[7]	Reserved	Reserved.
[6]	CLKOCKEN	CLKO Clock Enable Bit 0 = CLKO clock Disabled. 1 = CLKO clock Enabled.
[5]	TMR3CKEN	Timer3 Clock Enable Bit 0 = Timer3 clock Disabled. 1 = Timer3 clock Enabled.
[4]	TMR2CKEN	Timer2 Clock Enable Bit 0 = Timer2 clock Disabled. 1 = Timer2 clock Enabled.
[3]	TMR1CKEN	Timer1 Clock Enable Bit 0 = Timer1 clock Disabled. 1 = Timer1 clock Enabled.
[2]	TMR0CKEN	Timer0 Clock Enable Bit 0 = Timer0 clock Disabled. 1 = Timer0 clock Enabled.
[1]	RTCKEN	Real-time-clock APB Interface Clock Enable Bit This bit is used to control the RTC APB clock. 0 = RTC clock Disabled. 1 = RTC clock Enabled.
[0]	WDTCKEN	Watchdog Timer Clock Enable Bit (Write Protect) 0 = Watchdog timer clock Disabled. 1 = Watchdog timer clock Enabled. Note 1: This bit is write protected. Refer to the SYS_REGLCTL register. Note 2: This bit is forced to 1 when CONFIG0[3] or CONFIG0[4] or CONFIG0[31] is 0. Note 3: Reset by power on reset or watchdog reset or software chip reset.

APB Devices Clock Enable Control Register 1 (CLK_APBCLK1)

The bits in this register are used to enable/disable clock for peripheral controller clocks.

Register	Offset	R/W	Description	Reset Value
CLK_APBCLK1	CLK_BA+0x0C	R/W	APB Devices Clock Enable Control Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved					BPWM0CKEN	Reserved	
15	14	13	12	11	10	9	8
Reserved							USCI0CKEN
7	6	5	4	3	2	1	0
Reserved							

Bits	Description	
[31:19]	Reserved	Reserved.
[18]	BPWM0CKEN	BPWM0 Clock Enable Bit 0 = BPWM0 clock Disabled. 1 = BPWM0 clock Enabled.
[17:9]	Reserved	Reserved.
[8]	USCI0CKEN	USCI0 Clock Enable Bit 0 = USCI0 clock Disabled. 1 = USCI0 clock Enabled.
[7:0]	Reserved	Reserved.

Clock Source Select Control Register 0 (CLK_CLKSEL0)

Register	Offset	R/W	Description	Reset Value
CLK_CLKSEL0	CLK_BA+0x10	R/W	Clock Source Select Control Register 0	0x0000_001d

31	30	29	28	27	26	25	24
Reserved							CANFD0SEL
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		STCLKSEL			HCLKSEL		

Bits	Description
[31:25]	Reserved Reserved.
[24]	CANFD0SEL CANFD0 Clock Source Selection 0 = Clock source from HCLK. 1 = Clock source from HXT.
[23:6]	Reserved Reserved.
[5:3]	STCLKSEL Cortex®-M23 SysTick Clock Source Selection (Write Protect) If SYST_CTRL[2]=0, SysTick uses clock source listed below. 000 = Clock source from HXT. 001 = Clock source from LXT. 010 = Clock source from HXT/2. 011 = Clock source from HCLK/2. 111 = Clock source from HIRC/2. Note 1: if SysTick clock source is not from HCLK (i.e. SYST_CTRL[2] = 0), SysTick clock source must be less than or equal to HCLK/2. Note 2: This bit is write protected. Refer to the SYS_REGLCTL register.
[2:0]	HCLKSEL HCLK Clock Source Selection (Write Protect) Before clock switching, the related clock sources (both pre-select and new-select) must be turned on. 000 = Clock source from HXT. 001 = Clock source from LXT. 011 = Clock source from LIRC. 101 = Clock source from MIRC. 111 = Clock source from HIRC. Other = Reserved. Note 1: This bit is write protected. Refer to the SYS_REGLCTL register. Note 2: Reset by power on reset.

Clock Source Select Control Register 1 (CLK_CLKSEL1)

Before clock switching, the related clock sources (pre-selected and newly-selected) must be turned on.

Register	Offset	R/W	Description	Reset Value
CLK_CLKSEL1	CLK_BA+0x14	R/W	Clock Source Select Control Register 1	0x4422_222b

31	30	29	28	27	26	25	24
Reserved	UART1SEL			Reserved	UART0SEL		
23	22	21	20	19	18	17	16
Reserved	TMR3SEL			Reserved	TMR2SEL		
15	14	13	12	11	10	9	8
Reserved	TMR1SEL			Reserved	TMR0SEL		
7	6	5	4	3	2	1	0
Reserved	CLKOSEL			WWDTSEL		WDTSEL	

Bits	Description
[31]	Reserved Reserved.
[30:28]	UART1SEL UART1 Clock Source Selection 000 = Clock source from 4~32 MHz external high speed crystal oscillator (HXT). 010 = Clock source from 32.768 kHz external low speed crystal oscillator (LXT). 011 = Clock source from 48 MHz internal high speed RC oscillator (HIRC). 100 = Clock source from PCLK1. 101 = Clock source from 38.4 kHz internal low speed RC oscillator (LIRC). Others = Reserved.
[27]	Reserved Reserved.
[26:24]	UART0SEL UART0 Clock Source Selection 000 = Clock source from 4~32 MHz external high speed crystal oscillator (HXT). 010 = Clock source from 32.768 kHz external low speed crystal oscillator (LXT). 011 = Clock source from 48 MHz internal high speed RC oscillator (HIRC). 100 = Clock source from PCLK0. 101 = Clock source from 38.4 kHz internal low speed RC oscillator (LIRC). Others = Reserved.
[23]	Reserved Reserved.
[22:20]	TMR3SEL TIMER3 Clock Source Selection 000 = Clock source from 4~32 MHz external high speed crystal oscillator (HXT). 001 = Clock source from 32.768 kHz external low speed crystal oscillator (LXT). 010 = Clock source from PCLK1. 011 = Clock source from external clock T3 pin. 101 = Clock source from 38.4 kHz internal low speed RC oscillator (LIRC). 111 = Clock source from 48 MHz internal high speed RC oscillator (HIRC). Others = Reserved.

[19]	Reserved	Reserved.
[18:16]	TMR2SEL	TIMER2 Clock Source Selection 000 = Clock source from 4~32 MHz external high speed crystal oscillator (HXT). 001 = Clock source from 32.768 kHz external low speed crystal oscillator (LXT). 010 = Clock source from PCLK1. 011 = Clock source from external clock T2 pin. 101 = Clock source from 38.4 kHz internal low speed RC oscillator (LIRC). 111 = Clock source from 48 MHz internal high speed RC oscillator (HIRC). Others = Reserved.
[15]	Reserved	Reserved.
[14:12]	TMR1SEL	TIMER1 Clock Source Selection 000 = Clock source from 4~32 MHz external high speed crystal oscillator (HXT). 001 = Clock source from 32.768 kHz external low speed crystal oscillator (LXT). 010 = Clock source from PCLK0. 011 = Clock source from external clock T1 pin. 101 = Clock source from 38.4 kHz internal low speed RC oscillator (LIRC). 111 = Clock source from 48 MHz internal high speed RC oscillator (HIRC). Others = Reserved.
[11]	Reserved	Reserved.
[10:8]	TMR0SEL	TIMER0 Clock Source Selection 000 = Clock source from 4~32 MHz external high speed crystal oscillator (HXT). 001 = Clock source from 32.768 kHz external low speed crystal oscillator (LXT). 010 = Clock source from PCLK0. 011 = Clock source from external clock T0 pin. 101 = Clock source from 38.4 kHz internal low speed RC oscillator (LIRC). 111 = Clock source from 48 MHz internal high speed RC oscillator (HIRC). Others = Reserved.
[7]	Reserved	Reserved.
[6:4]	CLKOSEL	Clock Divider Clock Source Selection 000 = Clock source from 4~32 MHz external high speed crystal oscillator (HXT). 001 = Clock source from 32.768 kHz external low speed crystal oscillator (LXT). 010 = Clock source from HCLK. 011 = Clock source from 48 MHz internal high speed RC oscillator (HIRC). 100 = Clock source from 38.4 kHz internal low speed RC oscillator (LIRC). 101 = Clock source from 4 MHz internal medium speed RC oscillator (MIRC). 111 = Clock source from USB SOF. Others = Reserved.
[3:2]	WWDTSEL	Window Watchdog Timer Clock Source Selection (Write Protect) 10 = Clock source from HCLK/2048. 11 = Clock source from 38.4 kHz internal low speed RC oscillator (LIRC). Others = Reserved.
[1:0]	WDTSEL	Watchdog Timer Clock Source Selection (Write Protect) 00 = Reserved. 01 = Clock source from 32.768 kHz external low speed crystal oscillator (LXT). 10 = Clock source from HCLK/2048.

		11 = Clock source from 38.4 kHz internal low speed RC oscillator (LIRC). Note 1: This bit is write protected. Refer to the SYS_REGLCTL register. Note 2: This bit is forced to 11 when CONFIG0[31] or CONFIG0[4] or CONFIG0[3] is 0.

Clock Source Select Control Register 2 (CLK_CLKSEL2)

Before clock switching, the related clock sources (pre-select and new-select) must be turned on.

Register	Offset	R/W	Description	Reset Value
CLK_CLKSEL2	CLK_BA+0x18	R/W	Clock Source Select Control Register 2	0x0000_0020

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		SPI0SEL		Reserved			

Bits	Description
[31:6]	Reserved Reserved.
[5:4]	SPI0SEL SPI0 Clock Source Selection 00 = Clock source from 4~32 MHz external high speed crystal oscillator (HXT). 01 = Reserved. 10 = Clock source from PCLK1. 11 = Clock source from 48 MHz internal high speed RC oscillator (HIRC).
[3:0]	Reserved Reserved.

Clock Source Select Control Register 3 (CLK_CLKSEL3)

Before clock switching, the related clock sources (pre-select and new-select) must be turned on.

Register	Offset	R/W	Description	Reset Value
CLK_CLKSEL3	CLK_BA+0x1C	R/W	Clock Source Select Control Register 3	0x4400_0040

31	30	29	28	27	26	25	24
Reserved	UART3SEL			Reserved	UART2SEL		
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved	UART4SEL			Reserved			

Bits	Description
[31]	Reserved Reserved.
[30:28]	UART3SEL UART3 Clock Source Selection 000 = Clock source from 4~32 MHz external high speed crystal oscillator (HXT). 010 = Clock source from 32.768 kHz external low speed crystal oscillator (LXT). 011 = Clock source from 48 MHz internal high speed RC oscillator (HIRC). 100 = Clock source from PCLK1. 101 = Clock source from 38.4 kHz internal low speed RC oscillator (LIRC). Others = Reserved.
[27]	Reserved Reserved.
[26:24]	UART2SEL UART2 Clock Source Selection 000 = Clock source from 4~32 MHz external high speed crystal oscillator (HXT). 010 = Clock source from 32.768 kHz external low speed crystal oscillator (LXT). 011 = Clock source from 48 MHz internal high speed RC oscillator (HIRC). 100 = Clock source from PCLK0. 101 = Clock source from 38.4 kHz internal low speed RC oscillator (LIRC). Others = Reserved.
[23:7]	Reserved Reserved.
[6:4]	UART4SEL UART4 Clock Source Selection 000 = Clock source from 4~32 MHz external high speed crystal oscillator (HXT). 010 = Clock source from 32.768 kHz external low speed crystal oscillator (LXT). 011 = Clock source from 48 MHz internal high speed RC oscillator (HIRC). 100 = Clock source from PCLK0. 101 = Clock source from 38.4 kHz internal low speed RC oscillator (LIRC). Others = Reserved.
[3:0]	Reserved Reserved.

Clock Divider Number Register 0 (CLK_CLKDIV0)

Register	Offset	R/W	Description	Reset Value
CLK_CLKDIV0	CLK_BA+0x20	R/W	Clock Divider Number Register 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
EADCDIV							
15	14	13	12	11	10	9	8
UART1DIV				UART0DIV			
7	6	5	4	3	2	1	0
Reserved				HCLKDIV			

Bits	Description
[31:24]	Reserved Reserved.
[23:16]	EADCDIV EADC Clock Divide Number From EADC Clock Source EADC clock frequency = (EADC clock source frequency) / (EADCDIV + 1).
[15:12]	UART1DIV UART1 Clock Divide Number From UART1 Clock Source UART1 clock frequency = (UART1 clock source frequency) / (UART1DIV + 1).
[11:8]	UART0DIV UART0 Clock Divide Number From UART0 Clock Source UART0 clock frequency = (UART0 clock source frequency) / (UART0DIV + 1).
[7:4]	Reserved Reserved.
[3:0]	HCLKDIV HCLK Clock Divide Number From HCLK Clock Source HCLK clock frequency = (HCLK clock source frequency) / (HCLKDIV + 1). Note: This field is only reset by power on reset.

Clock Divider Number Register 4 (CLK_CLKDIV4)

Register	Offset	R/W	Description	Reset Value
CLK_CLKDIV4	CLK_BA+0x30	R/W	Clock Divider Number Register 4	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved						CANFD0DIV	
15	14	13	12	11	10	9	8
Reserved				UART4DIV			
7	6	5	4	3	2	1	0
UART3DIV				UART2DIV			

Bits	Description
[31:18]	Reserved Reserved.
[17:16]	CANFD0DIV CANFD0 Clock Divide Number From CANFD0 Clock Source CANFD0 clock frequency = (CANFD0 clock source frequency) / (CANFD0DIV + 1).
[15:12]	Reserved Reserved.
[11:8]	UART4DIV UART4 Clock Divide Number From UART4 Clock Source UART4 clock frequency = (UART4 clock source frequency) / (UART4DIV + 1).
[7:4]	UART3DIV UART3 Clock Divide Number From UART3 Clock Source UART3 clock frequency = (UART3 clock source frequency) / (UART3DIV + 1).
[3:0]	UART2DIV UART2 Clock Divide Number From UART2 Clock Source UART2 clock frequency = (UART2 clock source frequency) / (UART2DIV + 1).

APB Clock Divider Register (CLK_PCLKDIV)

Register	Offset	R/W	Description	Reset Value
CLK_PCLKDIV	CLK_BA+0x34	R/W	APB Clock Divider Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved	APB1DIV			Reserved	APB0DIV		

Bits	Description	
[31:7]	Reserved	Reserved.
[6:4]	APB1DIV	APB1 Clock Divider APB1 clock can be divided from HCLK 000: PCLK1 = HCLK. 001: PCLK1 = 1/2 HCLK. 010: PCLK1 = 1/4 HCLK. 011: PCLK1 = 1/8 HCLK. 100: PCLK1 = 1/16 HCLK. 101: PCLK1 = 1/32 HCLK. Others: Reserved.
[3]	Reserved	Reserved.
[2:0]	APB0DIV	APB0 Clock Divider APB0 clock can be divided from HCLK 000: PCLK0 = HCLK. 001: PCLK0 = 1/2 HCLK. 010: PCLK0 = 1/4 HCLK. 011: PCLK0 = 1/8 HCLK. 100: PCLK0 = 1/16 HCLK. 101: PCLK0 = 1/32 HCLK. Others: Reserved.

Clock Status Monitor Register (CLK_STATUS)

The bits in this register are used to monitor if the chip clock source is stable or not, and whether the clock switch is failed.

Register	Offset	R/W	Description	Reset Value
CLK_STATUS	CLK_BA+0x50	R	Clock Status Monitor Register	0x0000_00XX

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
CLKSFALL	MIRCSTB	Reserved	HIRCSTB	LIRCSTB	Reserved	LXTSTB	HXTSTB

Bits	Description
[31:8]	Reserved Reserved.
[7]	CLKSFALL Clock Switching Fail Flag This bit is updated when software switches system clock source. If switch target clock is stable, this bit will be set to 0. If switch target clock is not stable, this bit will be set to 1. 0 = Clock switching success. 1 = Clock switching failure. Note: Write 1 to clear the bit to 0.
[6]	MIRCSTB MIRC Clock Source Stable Flag (Read Only) 0 = 4 MHz internal mid speed RC oscillator (MIRC) clock is not stable or disabled. 1 = 4 MHz internal mid speed RC oscillator (MIRC) clock is stable and enabled.
[5]	Reserved Reserved.
[4]	HIRCSTB HIRC Clock Source Stable Flag (Read Only) 0 = 48 MHz internal high speed RC oscillator (HIRC) clock is not stable or disabled. 1 = 48 MHz internal high speed RC oscillator (HIRC) clock is stable and enabled.
[3]	LIRCSTB LIRC Clock Source Stable Flag (Read Only) 0 = 38.4 kHz internal low speed RC oscillator (LIRC) clock is not stable or disabled. 1 = 38.4 kHz internal low speed RC oscillator (LIRC) clock is stable and enabled.
[2]	Reserved Reserved.
[1]	LXTSTB LXT Clock Source Stable Flag (Read Only) 0 = 32.768 kHz external low speed crystal oscillator (LXT) clock is not stable or disabled. 1 = 32.768 kHz external low speed crystal oscillator (LXT) clock is stable and enabled.
[0]	HXTSTB HXT Clock Source Stable Flag (Read Only) 0 = 4~32 MHz external high speed crystal oscillator (HXT) clock is not stable or disabled. 1 = 4~32 MHz external high speed crystal oscillator (HXT) clock is stable and enabled.

Clock Output Control Register (CLK_CLKOCTL)

Register	Offset	R/W	Description	Reset Value
CLK_CLKOCTL	CLK_BA+0x60	R/W	Clock Output Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved	CLK1HZEN	DIV1EN	Reserved	FREQSEL			

Bits	Description
[31:7]	Reserved Reserved.
[6]	CLK1HZEN Clock Output 1Hz Enable Bit 0 = 1 Hz clock output for 32.768 kHz frequency compensation Disabled. 1 = 1 Hz clock output for 32.768 kHz frequency compensation Enabled.
[5]	DIV1EN Clock Output Divide One Enable Bit 0 = Clock Output will output clock with source frequency divided by FREQSEL. 1 = Clock Output will output clock with source frequency.
[4]	Reserved Reserved.
[3:0]	FREQSEL Clock Output Frequency Selection The formula of output frequency is $F_{out} = F_{in}/2^{(N+1)}$. F_{in} is the input clock frequency. F_{out} is the frequency of divider output clock. N is the 4-bit value of FREQSEL[3:0].

Clock Fail Detector Control Register (CLK_CLKDCTL)

Register	Offset	R/W	Description	Reset Value
CLK_CLKDCTL	CLK_BA+0x70	R/W	Clock Fail Detector Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved						HXTFQIEN	HXTFQDEN
15	14	13	12	11	10	9	8
Reserved		LXTFIEN	LXTFDEN	Reserved			
7	6	5	4	3	2	1	0
Reserved		HXTFIEN	HXTFDEN	Reserved			

Bits	Description
[31:18]	Reserved Reserved.
[17]	HXTFQIEN HXT Clock Frequency Range Detector Interrupt Enable Bit 0 = 4~32 MHz external high speed crystal oscillator (HXT) clock frequency range detector fail interrupt Disabled. 1 = 4~32 MHz external high speed crystal oscillator (HXT) clock frequency range detector fail interrupt Enabled.
[16]	HXTFQDEN HXT Clock Frequency Range Detector Enable Bit 0 = 4~32 MHz external high speed crystal oscillator (HXT) clock frequency range detector Disabled. 1 = 4~32 MHz external high speed crystal oscillator (HXT) clock frequency range detector Enabled.
[15:14]	Reserved Reserved.
[13]	LXTFIEN LXT Clock Fail Interrupt Enable Bit 0 = 32.768 kHz external low speed crystal oscillator (LXT) clock fail interrupt Disabled. 1 = 32.768 kHz external low speed crystal oscillator (LXT) clock fail interrupt Enabled.
[12]	LXTFDEN LXT Clock Fail Detector Enable Bit 0 = 32.768 kHz external low speed crystal oscillator (LXT) clock fail detector Disabled. 1 = 32.768 kHz external low speed crystal oscillator (LXT) clock fail detector Enabled.
[11:6]	Reserved Reserved.
[5]	HXTFIEN HXT Clock Fail Interrupt Enable Bit 0 = 4~32 MHz external high speed crystal oscillator (HXT) clock fail interrupt Disabled. 1 = 4~32 MHz external high speed crystal oscillator (HXT) clock fail interrupt Enabled.
[4]	HXTFDEN HXT Clock Fail Detector Enable Bit 0 = 4~32 MHz external high speed crystal oscillator (HXT) clock fail detector Disabled. 1 = 4~32 MHz external high speed crystal oscillator (HXT) clock fail detector Enabled.
[3:0]	Reserved Reserved.

Clock Fail Detector Status Register (CLK_CLKDSTS)

Register	Offset	R/W	Description	Reset Value
CLK_CLKDSTS	CLK_BA+0x74	R/W	Clock Fail Detector Status Register	0x0000_0000

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

Bits	Description
[31:9]	Reserved Reserved.
[8]	HXTFQIF HXT Clock Frequency Range Detector Interrupt Flag (Write Protect) 0 = 4~32 MHz external high speed crystal oscillator (HXT) clock frequency is normal. 1 = 4~32 MHz external high speed crystal oscillator (HXT) clock frequency is abnormal. Note: Write 1 to clear the bit to 0.
[7:2]	Reserved Reserved.
[1]	LXTFIF LXT Clock Fail Interrupt Flag (Write Protect) 0 = 32.768 kHz external low speed crystal oscillator (LXT) clock is normal. 1 = 32.768 kHz external low speed crystal oscillator (LXT) stops. Note: Write 1 to clear the bit to 0.
[0]	HXTFIF HXT Clock Fail Interrupt Flag (Write Protect) 0 = 4~32 MHz external high speed crystal oscillator (HXT) clock is normal. 1 = 4~32 MHz external high speed crystal oscillator (HXT) clock stops. Note: Write 1 to clear the bit to 0.

Clock Frequency Range Detector Upper Boundary Register (CLK_CDUPB)

Register	Offset	R/W	Description	Reset Value
CLK_CDUPB	CLK_BA+0x78	R/W	Clock Frequency Range Detector Upper Boundary Register	0x0000_0000

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

Bits	Description
[31:10]	Reserved Reserved.
[9:0]	HXT Clock Frequency Range Detector Upper Boundary Value The bits define the maximum value of frequency range detector window. When HXT frequency higher than this maximum frequency value, the HXT Clock Frequency Range Detector Interrupt Flag will set to 1. Note :Frequency out of range will be asserted when $HIRC\_period * 512 > HXT\_period * CLK\_DUPB$ or $HIRC\_period * 512 < HXT\_period * CLK\_CDLOWB$.

Clock Frequency Range Detector Lower Boundary Register (CLK_CDLOWB)

Register	Offset	R/W	Description	Reset Value
CLK_CDLOWB	CLK_BA+0x7c	R/W	Clock Frequency Range Detector Lower Boundary Register	0x0000_0000

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

Bits	Description
[31:10]	Reserved Reserved.
[9:0]	LOWERBD HXT Clock Frequency Range Detector Lower Boundary Value The bits define the minimum value of frequency range detector window. When HXT frequency lower than this minimum frequency value, the HXT Clock Frequency Range Detector Interrupt Flag will set to 1. Note :The frequency out of range will be asserted when $HIRC\_period * 512 > HXT\_period * CLK\_DUPB$ or $HIRC\_period * 512 < HXT\_period * CLK\_CDLOWB$.

Power Manager Control Register (CLK_PMUCTL)

Register	Offset	R/W	Description	Reset Value
CLK_PMUCTL	CLK_BA+0x90	R/W	Power Manager Control Register	0x0000_0000

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

Bits	Description
[31:3]	Reserved Reserved.
[2:0]	PDMSEL Power-down Mode Selection (Write Protect) This is a protected register. Please refer to open lock sequence to program it. These bits control chip power-down mode grade selection when CPU execute WFI/WFE instruction. 000 = Power-down mode is selected. (PD) 010 = fast wake up. others = Reserved. Note: This bit is write protected. Refer to the SYS_REGLCTL register.

HXT Filter Select Control Register (CLK_HXTFSEL)

Register	Offset	R/W	Description	Reset Value
CLK_HXTFSEL	CLK_BA+0xB4	R/W	HXT Filter Select Control Register	0x0000_000X

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

Bits	Description
[31:1]	Reserved Reserved.
[0]	HXTFSEL HXT Filter Select 0 = HXT frequency is greater than 12 MHz. 1 = HXT frequency is less than or equal to 12 MHz. Note 1: This bit is write protected. Refer to the SYS_REGLCTL register. Note 2: This bit should not be changed during HXT running.

6.4 Flash Memory Controller (FMC)

6.4.1 Overview

The FMC is equipped with 128 Kbytes on-chip embedded Flash for application. A User Configuration block provides for system initiation. A 4 Kbytes loader ROM (LDROM) is used for In-System-Programming (ISP) function. XOM (Execution Only Memory) setting block to conceal user program in XOM region. A 512 bytes cache with zero wait cycle is used to improve Flash access performance. This chip also supports In-Application-Programming (IAP) function. User switches the code executing without chip reset after the embedded Flash is updated.

6.4.2 Features

- Supports 128 Kbytes application ROM (APROM)
- Supports 4 Kbytes loader ROM (LDROM)
- Supports 1 XOM (Execution Only Memory) region to conceal user program in APROM
- Supports 12 bytes User Configuration block to control system initiation.
- Supports 512 bytes page erase for all embedded Flash
- Supports 32-bit and multi-word Flash programming function
- Supports CRC32 checksum calculation function
- Supports Flash all one verification function
- Supports embedded SRAM remap to system vector memory
- Supports In-System-Programming (ISP) / In-Application-Programming (IAP) to update embedded Flash memory
- Supports cache memory to improve Flash access performance and reduce power consumption

6.4.3 Block Diagram

The Flash memory controller (FMC) consists of AHB slave interface, cache memory controller, Flash control registers, Flash initialization controller, Flash operation control and embedded Flash memory. The block diagram of Flash memory controller is shown in Figure 6.4-1.

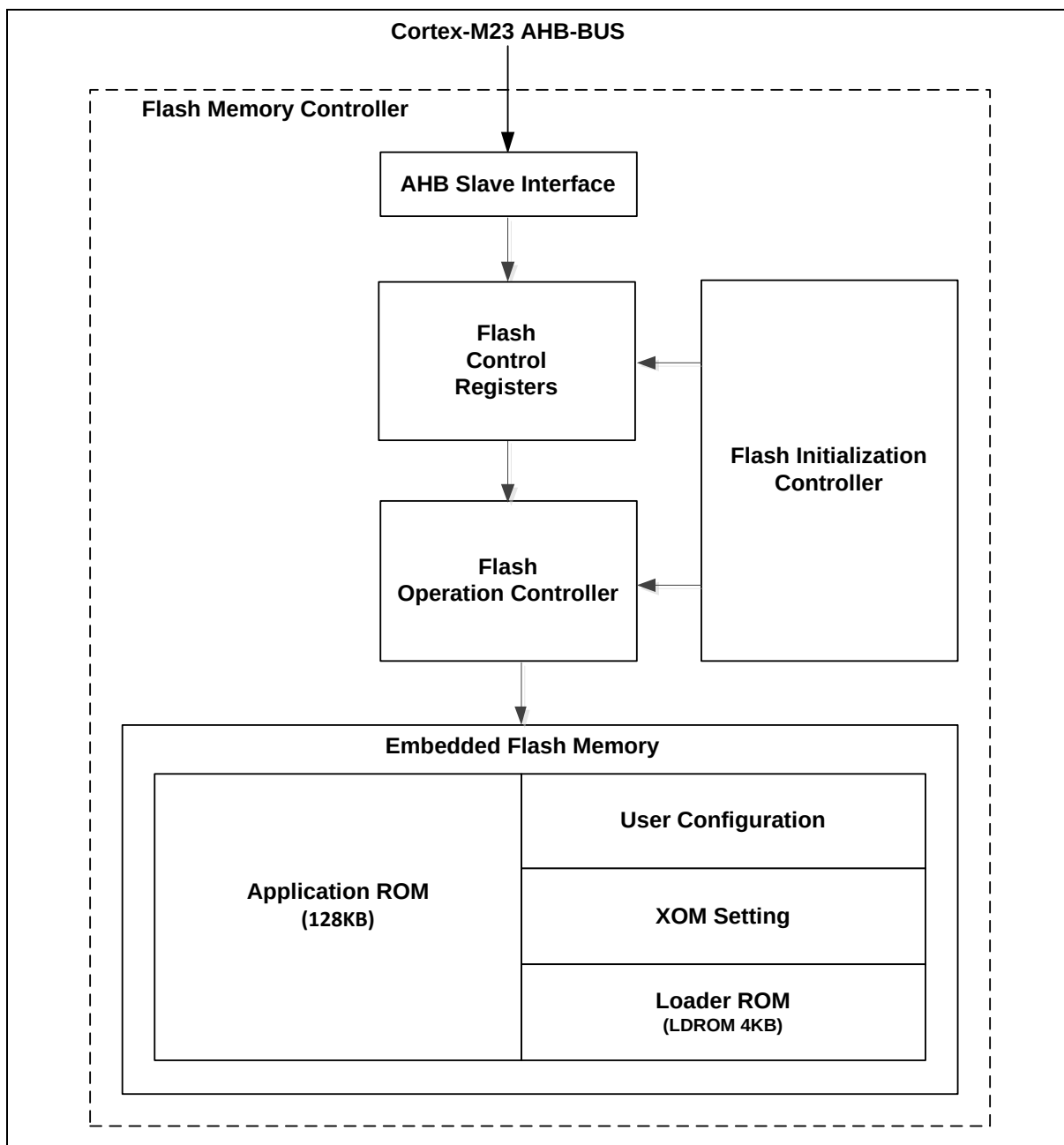


Figure 6.4-1 Flash Memory Controller Block Diagram

AHB Slave Interface

There is one AHB slave interface in Flash memory controller, the bus is from Cortex®-M23 AHB-Lite Bus for the instruction and data fetch and Flash control registers access including ISP registers.

Cache Memory Controller

A 512 bytes cache with zero wait cycle is implemented between Cortex®-M23 CPU and embedded Flash memory. This cache memory controller improves the Flash access performance and reduces power consumption of the embedded Flash memory.

Flash Control Registers

All of ISP control and status registers are in the Flash control registers. The detailed registers description is in the Register Description section.

Flash Initialization Controller

When chip is powered on or active from reset, the Flash initialization controller will start to access Flash automatically and check the Flash stability, and also reload User Configuration content to the Flash control registers for system initiation.

Flash Operation Controller

The Flash operations, such as Flash erase, Flash program, and Flash read operation, have specific control timing for embedded Flash memory. The Flash operation controller generates those control timing by requested from the cache memory controller, the Flash control registers and the Flash initialization controller.

Embedded Flash Memory

The embedded Flash memory is the main memory for user application code and parameters. It consists of the user configuration block, 4 Kbytes LDROM, 512 bytes XOM setting page and 128 Kbytes APROM. The page erase Flash size is 512 bytes, and minimum program bit size is 32 bits.

6.4.4 Functional Description

FMC functions include the memory organization, boot selection, IAP, ISP, the embedded Flash programming, and checksum calculation. The Flash memory map and system memory map are also introduced in the memory organization.

6.4.4.1 Memory Organization

The FMC memory consists of the embedded Flash memory which is programmable, and includes APROM, LDROM, XOM setting pages and the User Configuration block. The address map includes Flash memory map and four system address maps: LDROM with IAP, LDROM without IAP, APROM with IAP and APROM without IAP, functions.

6.4.4.2 LDROM and APROM

LDROM is designed for a loader to implement In-System-Programming (ISP) function by user. LDROM is a 4 Kbytes embedded Flash memory, the Flash address range is from 0x0010_0000 to 0x0010_0FFF. APROM is main memory for user applications. APROM size is 128 Kbytes. The page size of all of the embedded Flash memory is 512 bytes.

6.4.4.3 User Configuration Block

User Configuration block is internal programmable configuration area for boot options, such as Flash security lock, boot select and brown-out voltage level. It works like a fuse for power on setting. It is loaded from Flash memory to its corresponding control registers during chip power on. User can set these bits according to different application requests. User Configuration block can be updated by ISP function and located at 0x0030_0000 with three 32 bits words (CONFIG0, CONFIG1 and CONFIG2). Any change on User Configuration block will take effect after system reboot.

CONFIG0 (Address = 0x0030 0000)

31	30	29	28	27	26	25	24
CWDTEN[2]	CWDTPDEN	Reserved				CBCKSEL	Reserved
23	22	21	20	19	18	17	16
CBOV			CBORST	CBODEN	Reserved		
15	14	13	12	11	10	9	8
Reserved			ICELOCK	Reserved	CIOINI	Reserved	
7	6	5	4	3	2	1	0
CBS		Reserved	CWDTEN[1:0]		Reserved	LOCK	Reserved

Bits	Descriptions	
[31]	CWDTEN[2]	Watchdog Timer Hardware Enable Bit When watchdog timer hardware enable function is enabled, the watchdog enable bit WDTEN (WDT_CTL[7]) and watchdog reset enable bit RSTEN (WDT_CTL[1]) is set to 1 automatically after power on. The clock source of watchdog timer is force at LIRC and LIRC can't be disabled in normal operation mode. However, in Power-down mode, the LIRC can be disabled by setting CWDTEN[2:0]=011, CWDTPDEN=1 and LIRCEN=0 (CLK_PWRCTL[3]). CWDTEN[2:0] is CONFIG0[31][4][3], 011 = WDT hardware enable function is active. WDT clock is always on except chip enters Power-down mode. When chip enter Power-down mode, WDT clock is always on if CWDTPDEN is 0 or WDT clock is controlled by LIRCEN (CLK_PWRCTL[3]) if CWDTPDEN is 1. Please refer to bit field description of CWDTPDEN. 111 = WDT hardware enable function is inactive, WDT clock source only can be changed in this case. Others = WDT hardware enable function is active. WDT clock is always on.
[30]	CWDTPDEN	Watchdog Clock Power-down Enable Bit This bit should be used with CWDTEN. When WDT enabled by CWDTEN, Use this bit to control WDT wake-up when system is in Power-down mode. If it is necessary to wake up system by WDT, set CWDTPDEN=0 to make sure WDT keep working in Power-down mode. If user does not want to wake up system by WDT, just set CWDTPDEN=1 and LIRCEN=0 to let WDT suspend in power down. 0 = Watchdog Timer clock kept enabled when chip enters Power-down. 1 = Watchdog Timer clock is controlled by LIRCEN (CLK_PWRCTL[3]) when chip enters Power-down. Note: This bit only works if CWDTEN[2:0] is set to 011.
[29:26]	Reserved	Reserved.
[25]	CBCKSEL	Booting Clock Selection 0 = 1 MHz, MIRC 4 MHz divide 4. 1 = 4 MHz, MIRC 4 MHz divide 1 (default).
[24]	Reserved	Reserved.

[23:21]	CBOV	Brown-Out Voltage Selection 000 = Reserved. 001 = Brown-out voltage is 1.8V. 010 = Brown-out voltage is 2.0V. 011 = Brown-out voltage is 2.4V. 100 = Brown-out voltage is 2.7V. 101 = Brown-out voltage is 3.0V. 110 = Brown-out voltage is 3.7V. 111 = Brown-out voltage is 4.4V.
[20]	CBORST	Brown-Out Reset Enable Bit 0 = Brown-out reset Enabled after powered on. 1 = Brown-out reset Disabled after powered on.
[19]	CBODEN	Brown-Out Detector Enable Bit 0 = Brown-out detect Enabled after powered on. 1 = Brown-out detect Disabled after powered on.
[18:13]	Reserved	Reserved.
[12]	ICELOCK	ICE Lock Bit This bit only used to disable ICE function. User may use it with LOCK (CONFIG0[1]) bit to increase security level. 0 = ICE function Disabled. 1 = ICE function Enabled.
[11]	Reserved	Reserved.
[10]	CIOINI	I/O Initial State Selection 0 = All GPIO set as Quasi-bidirectional mode after chip powered on. 1 = All GPIO set as input tri-state mode after powered on.
[9:8]	Reserved	Reserved.
[7:6]	CBS	Chip Booting Selection When CBS[0] = 0, the LDROM base address is mapping to 0x100000 and APROM base address is mapping to 0x0. User could access both APROM and LDROM without boot switching. In other words, if IAP mode is supported, the code in LDROM and APROM can be called by each other. 00 = Boot from LDROM with IAP mode. 01 = Boot from LDROM without IAP mode. 10 = Boot from APROM with IAP mode. 11 = Boot from APROM without IAP mode. Note: BS (FMC_ISPCTL[1]) only works on CBS[0] = 1. VECMAP function only works on CBS[0] = 0.
[5]	Reserved	Reserved.
[4:3]	CWDTEN[1:0]	Watchdog Timer Hardware Enable Bit Please refer to CWDTEN[2] (CONFIG0[31]) for detailed description.
[2]	Reserved	Reserved.
[1]	LOCK	Security Lock Control This bit is used to enable security lock function. If security lock function enabled, only the software in APROM and LDROM could read correct data. Any other way, e.g. ICE, will get 0xFFFFFFFF when read APROM and LDROM. 0 = Flash memory content is locked. 1 = Flash memory content is not locked.

[0]	Reserved	Reserved.

CONFIG1 (Address = 0x0030 0004)

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

Bits	Descriptions	
[31:0]	Reserved	Reserved.

CONFIG2 (Address = 0x0030 0008)

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

Bits	Descriptions	
[31:8]	Reserved	Reserved.
[7:0]	ALOCK	Advanced Security Lock Control 0x5A = Flash memory content is unlocked if LOCK (CONFIG0[1]) is set to 1. Others = Flash memory content is locked. Note: ALOCK will be programmed as 0x5A after executing page erase or whole chip erase

6.4.4.4 Execution Only Memory (XOM)

The execution only memory (XOM) is used to store instructions for security application which are not allowed for data access via AHB-Bus. There is only one XOM region in APROM (XOM0). To define a new XOM region, user must complete XOM settings in XOM setting page (0x20_0000 ~ 0x20_0008) via In-System-Programming (ISP) function. XOMR0BASE is used to define XOM region start address and XOMR0SIZE is used to define total page size of XOM region. Both XOMR0BASE and XOMR0SIZE must be page alignment. XOMR0CTRL is used to define the XOM region attribute, which can be defined as active, inactive or debug mode. After setting and restarting Flash initialization by any reset except CPU reset and MCU reset, user can check XOM status from FMC_XOMSTS and check each XOM range (i.e. base and size) from FMC_XOMR0STS. User needs to do chip reset after setting or erasing XOM.

When using XOM setting page and XOM regions, users must pay attention to the following limitations:

- XOM region cannot be programmed after finishing its XOM setting
- XOM setting cannot be changed during runtime
- Only use Flash 32-bit program command to program XOM setting page
- Clear XOM setting and XOM region by mass erase command or XOM page erase function (a special erase command for XOM)
- User must restart Flash initialization to activate any new XOM setting
- ICE can not step in XOM region except debug mode, otherwise ICE will disconnect

Once the XOM region is active, user can only adopt mass erase command or XOM page erase function to clear XOM region and its corresponding XOM setting. To execute XOM page erase function, user must perform FLASH Page Erase (0x22) by writing the base address of the target XOM region in FMC_ISPADDR and the constant value 0x0055aa03 in FMC_ISPDAT. User should not set XOM at region where chip can boot from.

XOMR0BASE (Address = 0x0020 0000)

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved	XOMR0BASE						
15	14	13	12	11	10	9	8
XOMR0BASE							
7	6	5	4	3	2	1	0
XOMR0BASE							

Bits	Descriptions	
[31:23]	Reserved	Reserved.
[22:0]	XOMR0BASE	Base Address of XOM Region 0 (Write Only) XOMR0BASE must be page-aligned. Note: Page size is 512 bytes.

XOMR0SIZE (Address = 0x0020 0004)

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							XOMR0SIZE
7	6	5	4	3	2	1	0
XOMR0SIZE							

Bits	Descriptions	
[31:9]	Reserved	Reserved.
[8:0]	XOMR0SIZE	Page Number of XOM Region 0 (Write Only) XOMR0SIZE must be page-aligned and less than 256 pages. The XOMR0SIZE minimum value is 1 page. If XOMR0SIZE is written to 0, the register value will be set to 1. Note: Page size is 512 bytes.

XOMR0CTRL (Address = 0x0020 0008)

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
XOMR0CTRL							

Bits	Descriptions	
[31:8]	Reserved	Reserved.
[7:0]	XOMR0CTRL	Control of XOM Region 0 (Write Only) 0x5a = XOM region 0 is off. 0x50 = XOM is in debug mode. The others = XOM region 0 is active. Note 1: But for Flash behavior, user cannot write 0 to 1. User needs to check the lock value is effective to change 0x5A(0101_1010). Note 2: The active state has come into effect after power-on or reset cycle.

6.4.4.5 Flash Memory Map

The Flash memory map is different from system memory map. The system memory map is used by CPU fetch code or data from FMC memory. The Flash memory map is used for ISP function to read, program or erase FMC memory. The Flash memory map is as Figure 6.4-2.

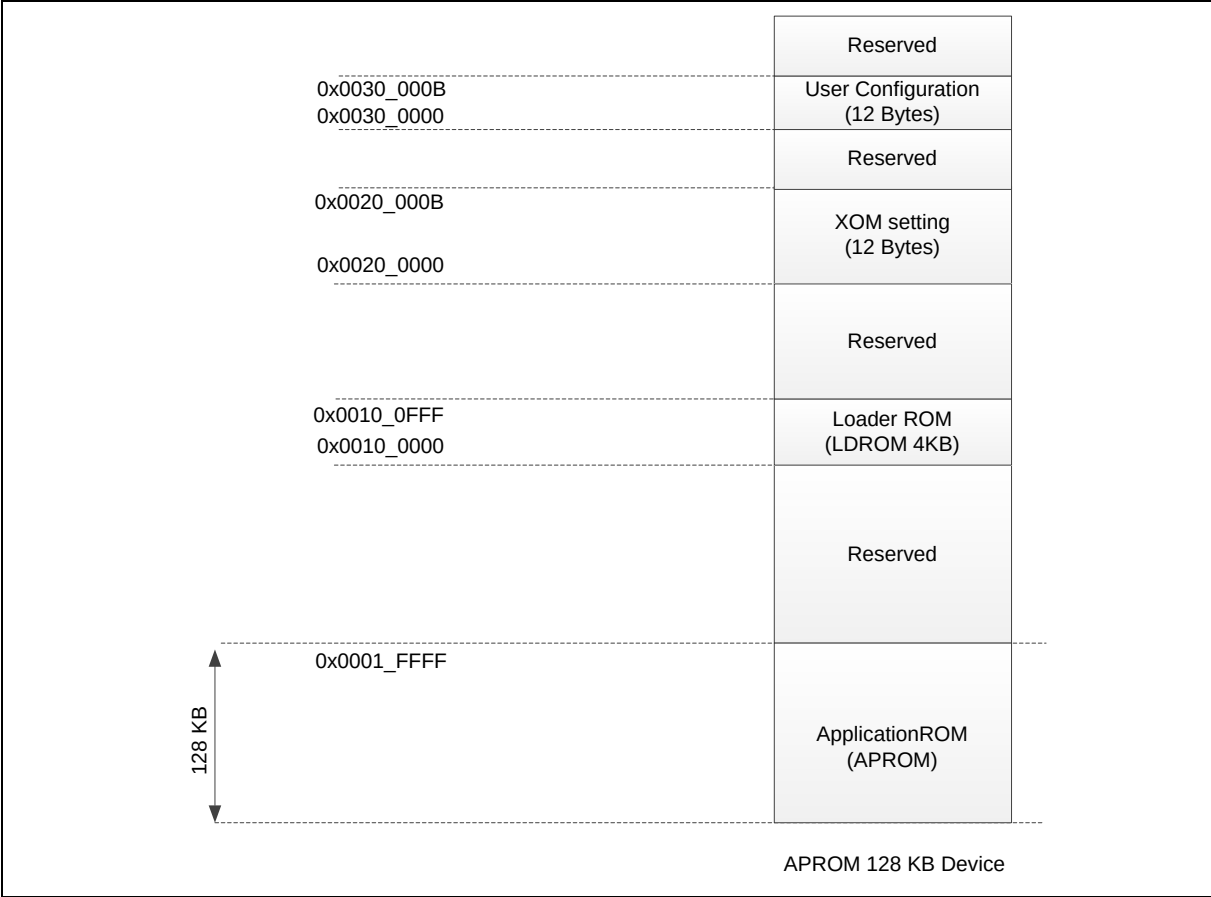


Figure 6.4-2 Flash Memory Map

6.4.4.6 System Memory Map with IAP Mode

The system memory map is used by CPU to fetch code or data from Flash memory. In IAP mode, CPU can read and execute the code from APROM and LDROM. It also supports to call the functions in LDROM from APROM or call the function in APROM from LDROM. That is why it called IAP mode.

The address from 0x0000_0000 to 0x0000_01FF is called vector map space. The vector map consist of vector table for stack and exceptions. By remapping the vector table in APROM or LDROM to the vector map space, it is possible to reboot to different applications.

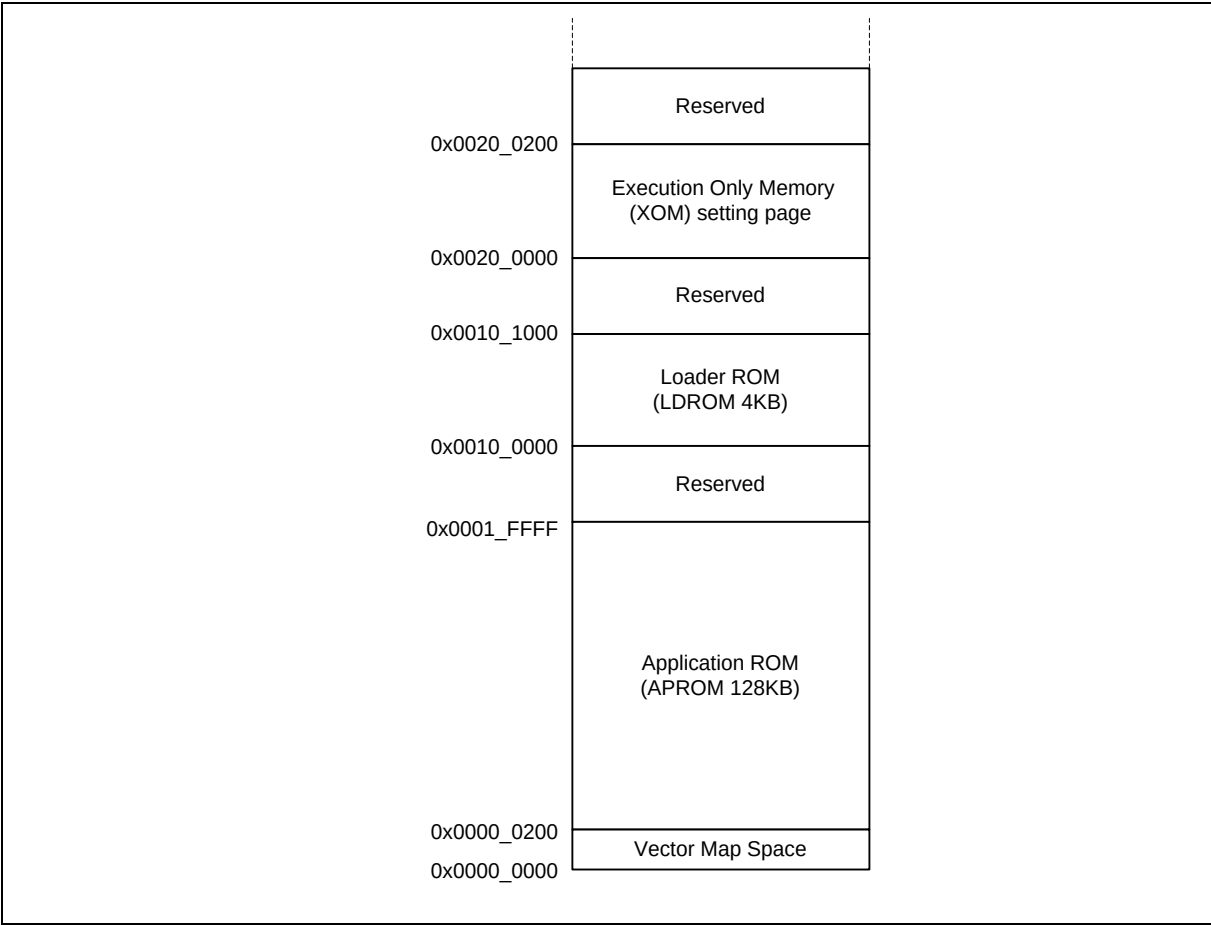


Figure 6.4-3 System Memory Map with IAP Mode

There are two kinds of system memory map with IAP mode when chip booting: (1) LDROM with IAP, (2) APROM with IAP.

In APROM with IAP mode, the vector map of APROM (0x0000_0000~0x0000_01FF) is mapping to the vector map space. Therefore, when CPU reset or System Reset Rquest reset, the stack pointer and reset hander in APROM vector table are used to reboot the system. This is so called boot from APROM.

In LDROM with IAP mode, the vector table of LDROM (0x0010_0000~0x0010_01FF) is mapping to the vector map space. Therefore, when CPU reset or System Reset Rquest reset, the stack pointer and reset hander in LDROM vector table are used to reboot the system. This is so called boot from LDROM.

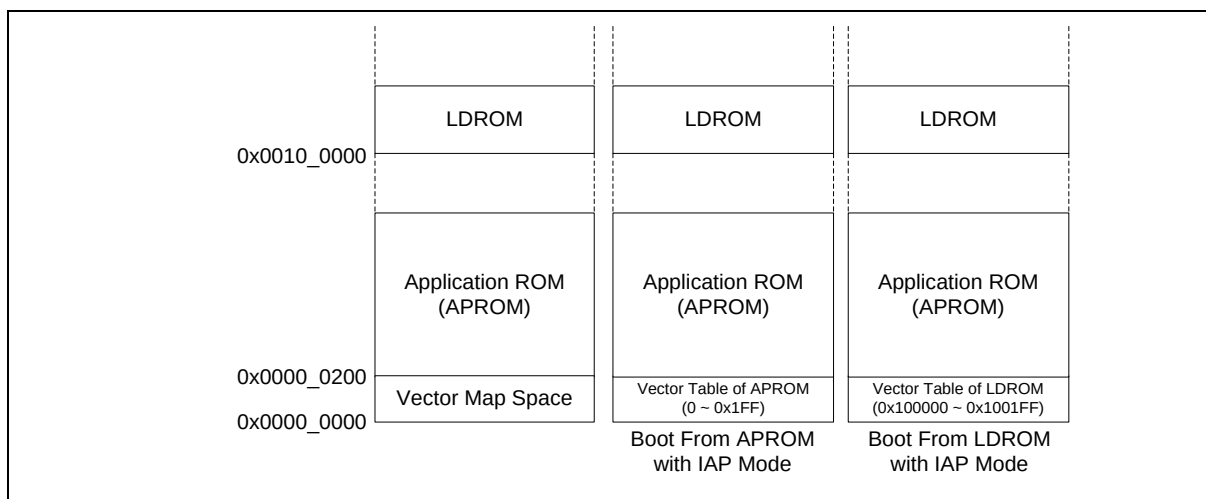


Figure 6.4-4 APROM/LDROM Boot with IAP Mode

The vector map space not only can map from vector table of APROM or LDROM. User can also map any 512 bytes alignment space of APROM, LDROM or SRAM to vector map space. To set vector map space mapping, user should write the target remap address to FMC_ISPADDR register and then trigger ISP procedure with the “Vector Page Remap” command (0x2E). The final system memory vector mapping address is shown on VECMAP (FMC_ISPSTS[29:9]). Please note that the vector mapping function only valid in IAP mode.

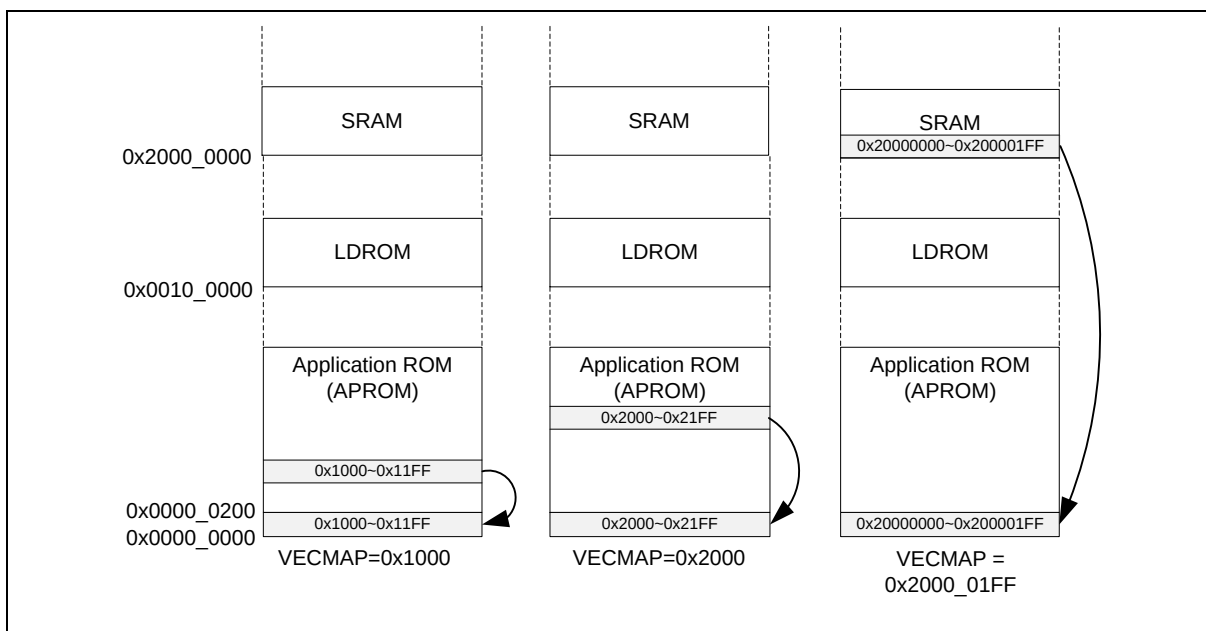


Figure 6.4-5 VECMAP Setting Example in IAP Mode

6.4.4.7 System Memory Map without IAP Mode

In system memory map without IAP mode, the system memory vector mapping is not supported. There are two kinds of system memory map without IAP mode when chip booting: (1)LDROM without IAP, (2) APROM without IAP.

In LDROM without IAP mode, LDROM base is mapping to 0x0000_0000. CPU cannot read or execute code from APROM. In other words, APROM is absent in the system memory map of LDROM without

IAP.

In APROM without IAP mode, APROM base is mapping to 0x0000_0000. CPU cannot read or execute code from LDROM. In this mode, LDROM is absent for current system memory map.

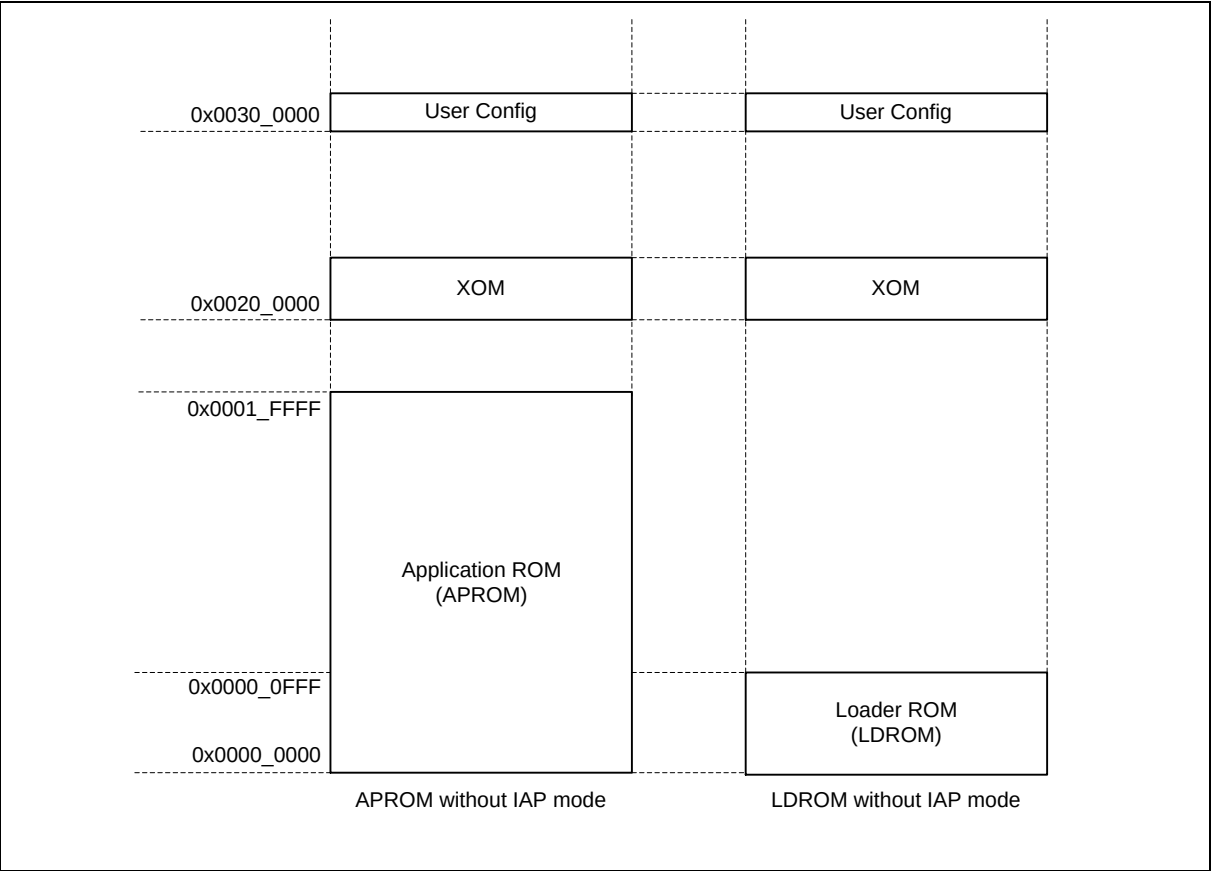


Figure 6.4-6 System Memory Map without IAP mode

6.4.4.8 Boot Selection

This chip provides four booting sources for user to select, including LDROM with IAP, LDROM without IAP, APROM with IAP and APROM without IAP. The booting source and system memory map are setting by CBS (CONFIG0[7:6]). The vector mapping function only supported in IAP mode, i.e., APROM with IAP and LDROM with IAP.

CBS[1:0]	Boot Selection/System Memory Map	Vector Mapping Supporting
00	LDROM with IAP	Yes
01	LDROM without IAP	No
10	APROM with IAP	Yes
11	APROM without IAP	No

Table 6.4-1 Boot Source Selection Table

6.4.4.9 In-Application-Programming (IAP)

In-Application-Programming (IAP) mode for user to switch the code executing between APROM, LDROM. User can enable the IAP mode by booting chip and setting the chip boot selection bits in CBS (CONFIG0[7:6]) as 2'b10 or 2'b00.

When chip boots with IAP function is enabled, any executable code (align to 512 bytes) is allowed to map to the system memory vector. User can change the remap address to FMC_ISPADDR and then trigger ISP procedure with the "Vector Page Remap" command (0x2E).

6.4.4.10 In-System-Programming (ISP)

In-System-Programming (ISP) function allowing the embedded Flash memory to be reprogrammed under software control. ISP is performed without addition devices and just utilize on-chip connectivity interface, such as UART, USB, I²C and SPI. Table 6.4-2 lists all supported ISP commands, including command code, address and data limitations.

ISP provides the following functions for embedded Flash memory.

- Supports Flash page erase function
- Supports Flash data program function
- Supports Flash data read function
- Supports company ID read function
- Supports device ID read function
- Supports unique ID read function
- Supports memory CRC32 checksum calculation function
- Supports Flash all one verification function
- Supports system memory vector remap function

ISP Commands

ISP Command	FMC_ISPCMD	FMC_ISPADDR	FMC_ISPDAT FMC_MPDAT0~ FMC_MPDAT3
Flash Page Erase	0x22	Valid address of Flash memory. It must be page alignment.	N/A or 0x0055AA03 if page erase XOM.
Flash 32-bit Program	0x21	Valid address of Flash memory. It must be word alignment.	FMC_ISPDAT :Programming Data FMC_MPDAT0~FMC_MPDAT3 : N/A
FLASH Multi-Word Program	0x27	Valid address of Flash memory organization. It must be 32-bit alignment.	FMC_ISPDAT :N/A FMC_MPDAT0: 1'st Programming Data FMC_MPDAT1: 2'nd Programming Data FMC_MPDAT2: 3'rd Programming Data FMC_MPDAT3: 4'th Programming Data
Flash 32-bit Read	0x00	Valid address of Flash memory organization It must be word alignment.	FMC_ISPDAT: Return Data FMC_MPDAT0~FMC_MPDAT3 : N/A

Read Company ID	0x0B	0x0000_0000	FMC_ISPDAT: 0x0000_00DA
Read Device ID	0x0C	0x0000_0000	FMC_ISPDAT: Return Device ID
Read Checksum	0x0D	Keep address of "Run Checksum Calculation"	FMC_ISPDAT: Return Checksum
Run Checksum Calculation	0x2D	Valid start address of memory organization It must be 512 bytes page alignment	FMC_ISPDAT: Size It must be 512 bytes alignment
Run Flash All One Verification	0x28	Valid start address of memory organization It must be 512 bytes page alignment	FMC_ISPDAT: Size It must be 512 bytes alignment
Read Unique ID	0x04	0x0000_0000	FMC_ISPDAT: Unique ID Word 0
		0x0000_0004	FMC_ISPDAT: Unique ID Word 1
		0x0000_0008	FMC_ISPDAT: Unique ID Word 2
Vector Remap	0x2E	Valid address in APROM, LDROM or SRAM with 512 bytes alignment	N/A

Table 6.4-2 ISP Command List

ISP Procedure

The FMC controller provides embedded Flash memory read, erase and program operation. Several control bits of FMC control register are write-protected, thus it is necessary to unlock before setting.

After unlocking the protected register bits, user needs to set the FMC_ISPCTL control register to decide to update LDROM, APROM or user configuration block, and then set ISPEN (FMC_ISPCTL[0]) to enable ISP function. .

Once the FMC_ISPCTL register is set properly, user can set FMC_ISPCMD (refer to Table 6.4-2) for specify operation. Set FMC_ISPADDR for target Flash memory based on Flash memory organization. FMC_ISPDAT can be used to set the data to program or used to return the read data according to FMC_ISPCMD. For Checksum calculation and All-One Verification commands, FMC_ISPDAT is used to define the size for calculation or verification.

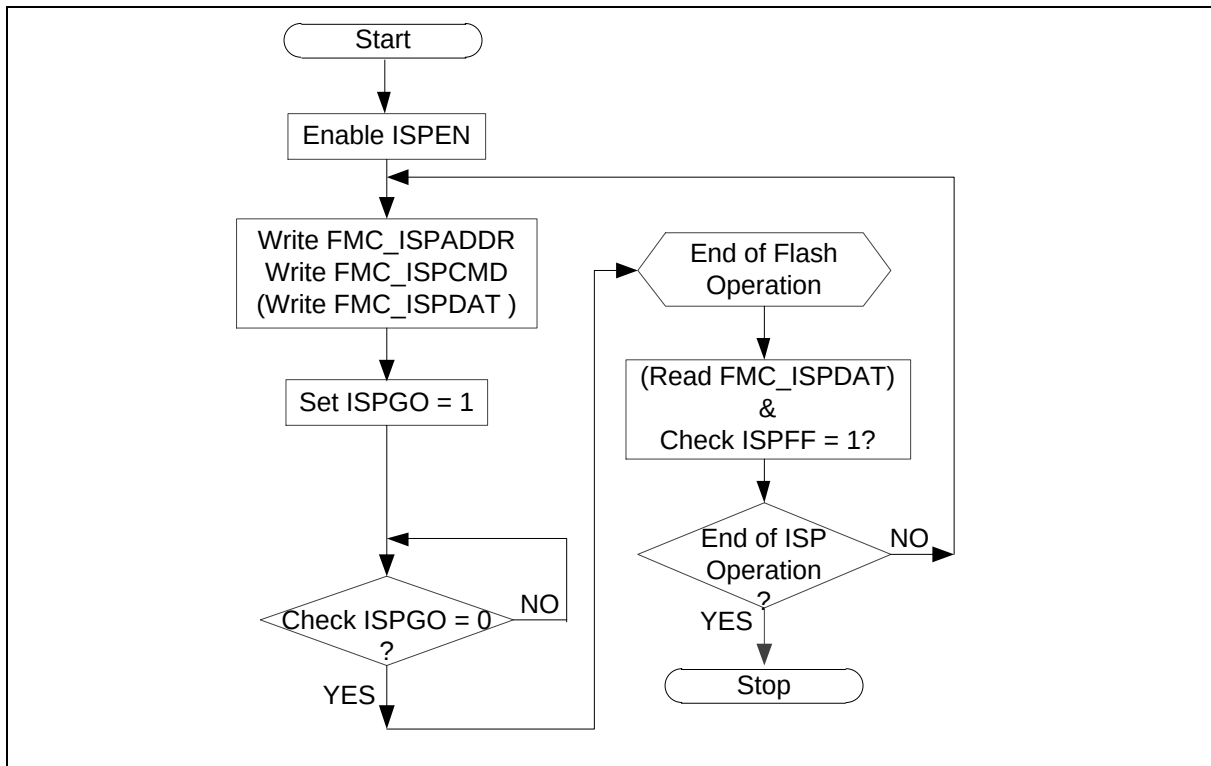


Figure 6.4-7 ISP Procedure Example

Finally, set the ISPGO (FMC_ISPTRG[0]) register to perform the relative ISP function. When ISP function is active, the ISPBUSY (FMC_ISPSTS[0]) and MPBUSY (FMC_MPSTS[0]) be set to 1. The ISPGO (FMC_ISPTRG[0]), ISPBUSY (FMC_ISPSTS[0]) and MPBUSY (FMC_MPSTS[0]) are self-cleared when ISP function has been done.

Several error conditions will be checked after ISP is completed. If an error condition occurs, ISP operation is not started and the ISP fail flag will be set instead. ISPFF (FMC_ISPSTS[6]) flag can only be cleared by software. The next ISP procedure can be started even ISPFF (FMC_ISPSTS[6]) bit is kept as 1. Therefore, it is recommended to check the ISPFF (FMC_ISPSTS[6]) bit and clear it after each ISP operation if it is set to 1.

While FMC is processing ISP command, CPU will be halt to wait ISP done if CPU trying to access Flash memory. For example, if any interrupt request occurs, CPU will not service it till ISP operation is finished. User could move their code and exception handlers to SRAM to avoid this situation. the peripheral still keeps working as usual when ISP processing.

When ISP operation is finished, the ISPGO bit will be cleared by hardware automatically. User can check whether ISP operation is finished or not by the ISPGO (FMC_ISPTRG[0]) bit.

6.4.4.11 VECMAP for Interrupt and Memory Programming

Accelerate Interrupt by VECMAP

In IAP mode, VECMAP function could be used to map 512 bytes SRAM to vector map space. It means it is possible to store all exception vectors to SRAM. Then, if any exceptions assert, CPU can read exception handler from SRAM with zero wait state to speed up exception latency.

Because the vector map space is fixed to be 512 bytes, user must copy all 512 bytes to SRAM before remapping SRAM to vector map space. Otherwise, CPU may get wrong data from vector map space after remapping. Figure 6.4-8 shows an example to accelerating interrupt by VECMAP.

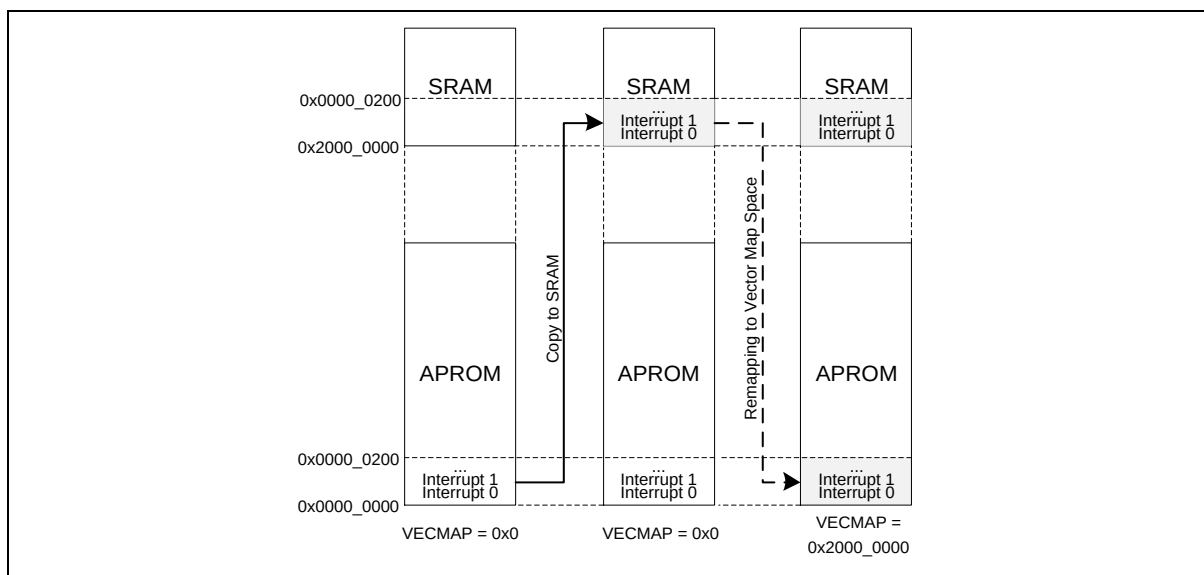


Figure 6.4-8 Example for Accelerating Interrupt by VECMAP

Avoid CPU Holt When Flash Programming

When Flash memory controller is in busy, any CPU access to Flash memory will cause CPU holt for waiting Flash controller ready. If Flash controller is busy in page erasing, it may cause CPU holt for a long time to erase pages. To avoid this situation, user needs to avoid CPU access Flash memory when page erasing. The easiest way is to execute code in SRAM and use VECMAP to map all exceptions to SRAM. By executing code in SRAM, CPU will not access Flash to get instructions. By mapping all exceptions to SRAM, all interrupts won't need to get exception handler from Flash memory.

6.4.4.12 Embedded Flash Memory Programming

This chip provides 32-bit and multi-word Flash memory programming function to speed up Flash updated procedure. Table 6.4-3 lists required FMC control registers in each embedded Flash programming function.

Register	Description	32-Bit Programming	Multi-Word Programming
FMC_ISPCTL	ISP Control Register	√	√
FMC_ISPADDR	ISP Address Register	√	√
FMC_ISPDAT	ISP Data Register	√	N/A
FMC_ISPCMD	ISP CMD Register	0x21	0x27
FMC_ISPTRG	ISP Trigger Register	√	√
FMC_ISPSTS	ISP Status Register	√	N/A
FMC_MPDAT0	ISP Data0 Register	N/A	√
FMC_MPDAT1	ISP Data1 Register	N/A	√
FMC_MPDAT2	ISP Data2 Register	N/A	√
FMC_MPDAT3	ISP Data3 Register	N/A	√
FMC_MPSTS	ISP Multi-Program status	N/A	√
FMC_MPADDR	ISP Multi-Program Address	N/A	√

Table 6.4-3 FMC Control Registers for Flash Programming

FMC_ISPDAT is used for 32-bit programming data register. Figure 6.4-9 shows ISP 32-bit programming procedure.

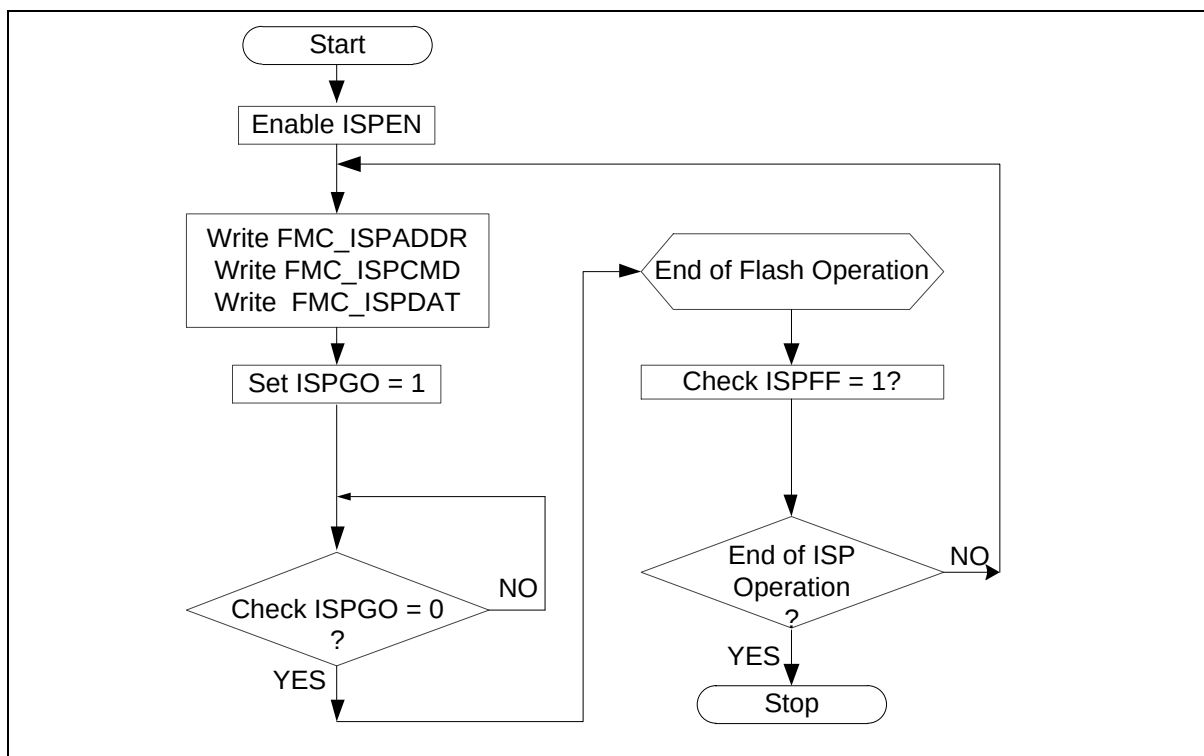


Figure 6.4-9 ISP 32-bit Programming Procedure

Multi-word Programming

This chip supports multi-word programming function to speed up Flash updated procedure. The maximum programming length is up to 128 bytes, and the minimum programming length is 4 bytes (1 words). The multi-word programming is the fastest programming function if the programming words more than 4 bytes.

In multi-word programming operation, Cortex®-M23 CPU has to monitor the empty status of the programming buffer. CPU has to prepare the next data for programming continuity. The multi-program firmware should not be located in APROM or LDROM, because CPU instruction fetch cannot be hold during ISP processing. The firmware has to be located in embedded SRAM of chip to avoid CPU hold. The multi-word programming code also needs to make sure all exceptions won't access Flash memory when ISP processing. Please note that, when doing multi-word programming, HCLK must select clock source from HIRC and frequency not lower than 4 MHz.

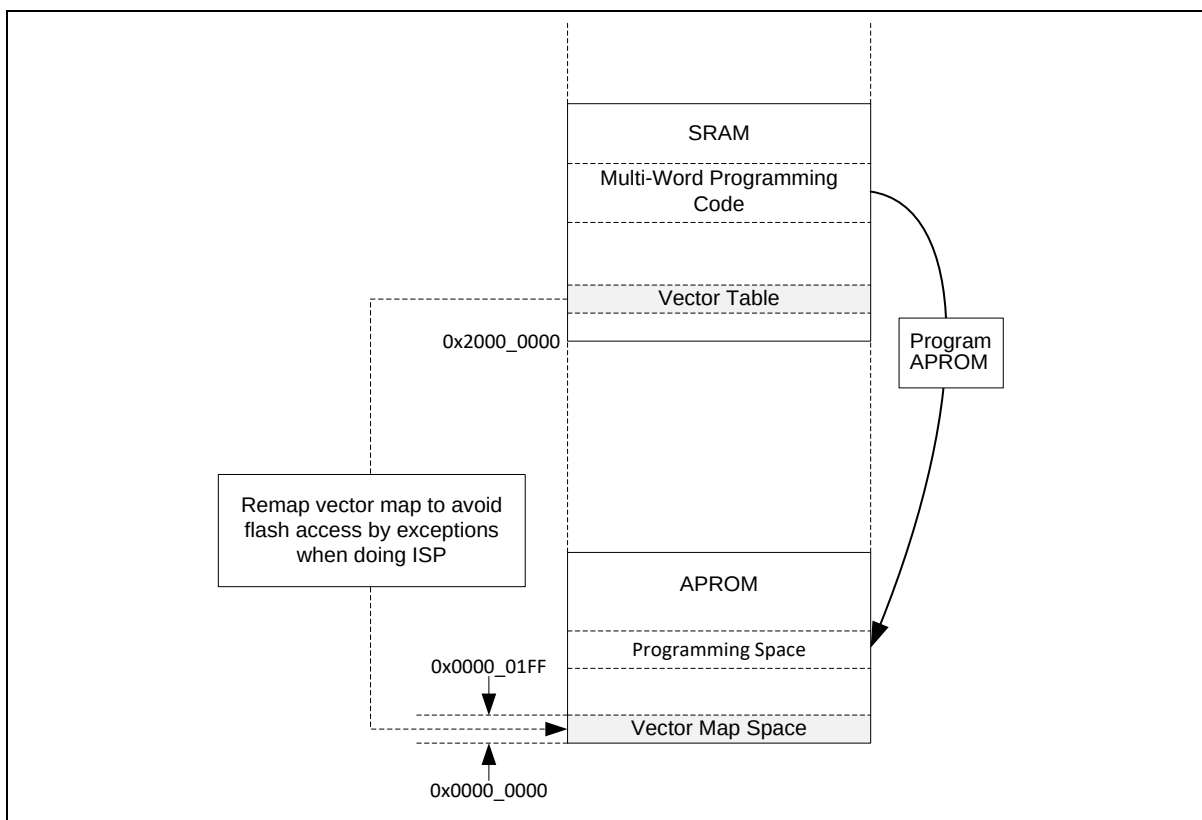


Figure 6.4-10 Firmware in SRAM for Multi-word Programming

The multi-word programming flow is shown in Figure 6.4-11. The starting ISP address (FMC_ISPADDR) has to be 4-byte align. FMC_MPDATA0 is the data word of the offset 0x0, FMC_MPDATA1 is the second word (offset 0x4), FMC_MPDATA2 is the third word (offset 0x8), and FMC_MPDATA3 is forth word (offset 0xC). If the starting ISP address FMC_ISPADDR [3] is 0, the 1st data word should put on FMC_MPDATA0, and 2nd word is FMC_MPDATA1, 3rd word is FMC_MPDATA2, and 4th word is FMC_MPDATA3. If the starting ISP address FMC_ISPADDR [3] is 1, the 1st data word should put on FMC_MPDATA2, and 2nd word is FMC_MPDATA3, 3rd word is FMC_MPDATA0, and 4th word is FMC_MPDATA1. The maximum programming size is 128 bytes. While FMC controller performs multi-word programming operation, CPU needs to monitor the buffer status D3~D0 (FMC_MPSTS[7:4]) and MPBUSY (FMC_MPSTS[0]) to wait any one buffer empty, and then CPU needs to update the next programming data (FMC_MPDATA0, FMC_MPDATA1, FMC_MPDATA2 and FMC_MPDATA3) before D3~D0 all equal to 0. Otherwise, FMC controller will exit multi-word programming operation (MPBUSY (FMC_MPSTS[0]) = 0). If CPU cannot update the data in time (MPBUSY (FMC_MPSTS[0]) = 0), CPU needs restart a new multi-word programming procedure to continue, FMC_MPADDR provides the last program address information. At the end of operation, CPU has to check ISPFF (FMC_MPSTS[2]) to confirm the multi-word operation successful complete.

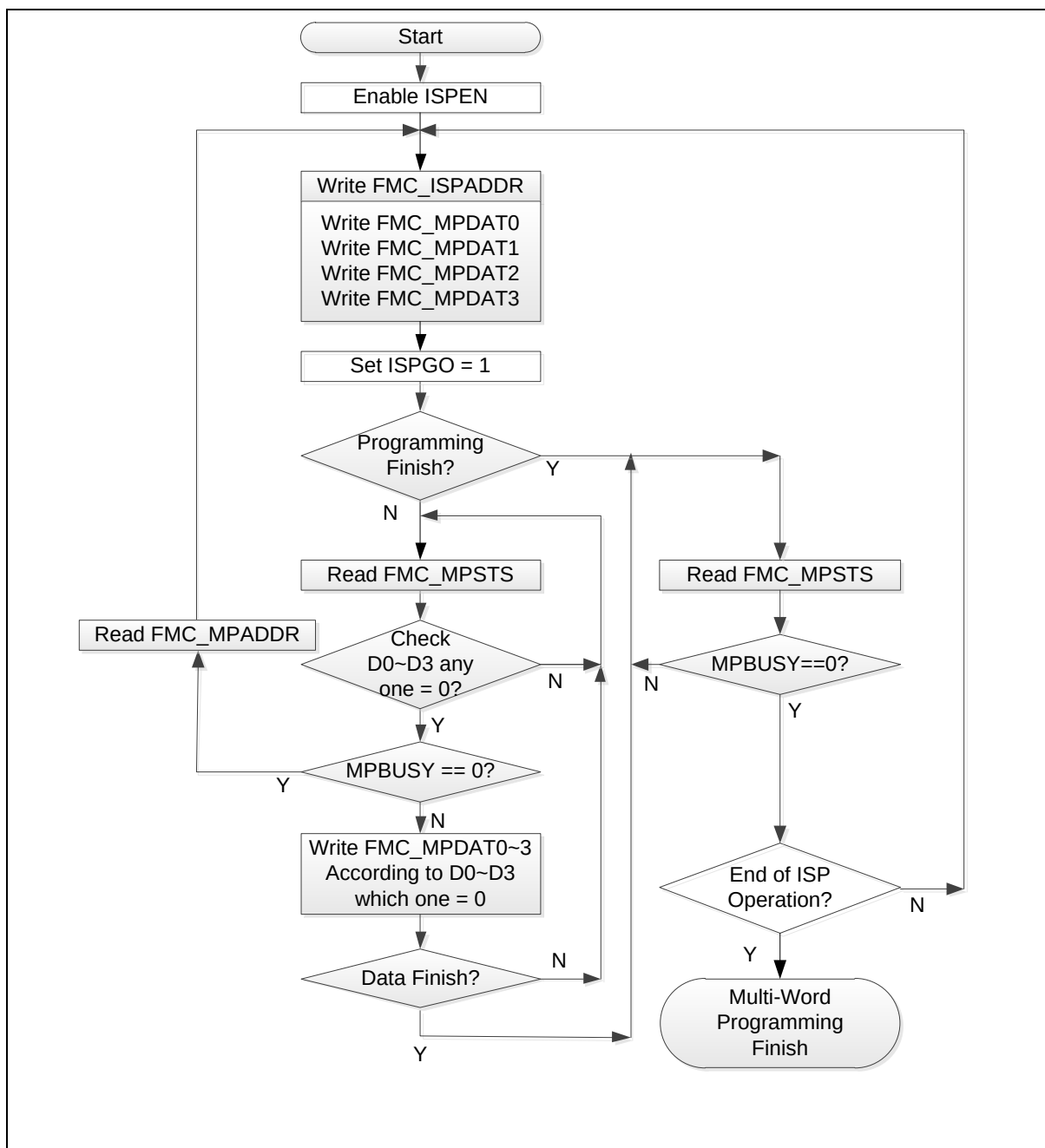


Figure 6.4-11 Multi-word Programming Flow

6.4.4.13 Fast Flash Programming Verification

In traditional Flash programming operation, the controller receives the programming trigger event then control the timing to perform the programming embedded Flash memory as show in Figure 6.4-12.

This chip supports the fast Flash programming verification function, which provides hardware verification for Flash programming to save time of the CPU read back and comparison. When data is programmed to the embedded Flash memory, the controller asserts the Flash read operation to read data out, and performs data comparison with data in. Finally, the comparison result is saved in PGFF (FMC_ISPSTS[5]). The PGFF is set to 1 if output data is not the same as the input programming data. The flag is kept until clear by software or a new erase operation. The fast Flash programming verification flow is shown in Figure 6.4-12.

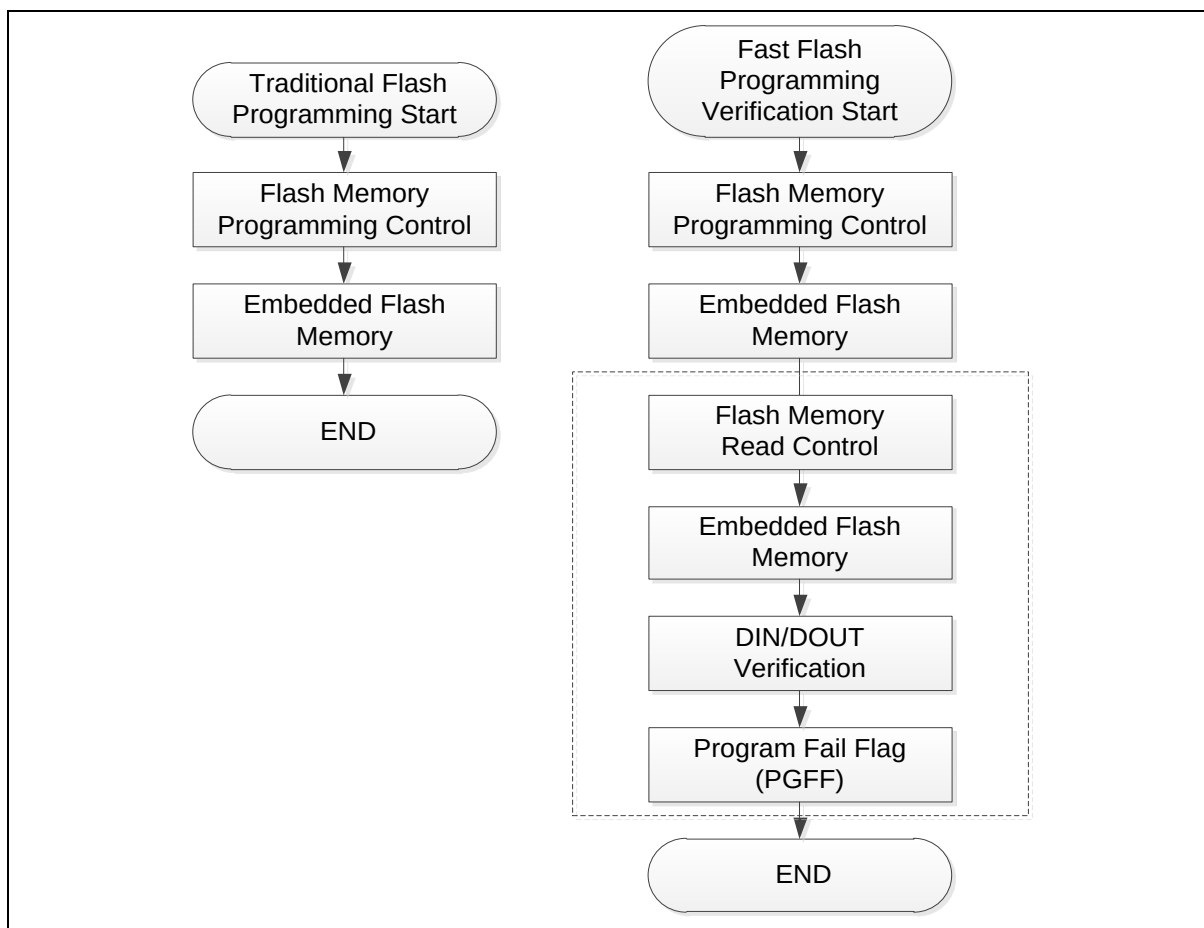


Figure 6.4-12 Fast Flash Programming Verification Flow

In traditional Flash updated operation, the Flash memory has to perform three steps to complete the Flash memory updated procedure, (1) Flash ERASE (2) Flash PROGRAM (3) Flash READ back all of data to check the correction. In this chip, it only reads FMC_ISPSTS to check PGFF flag in Step (3) without reading data back to confirm. Figure 6.4-12 compares traditional programming verification flow and fast programming verification flow.

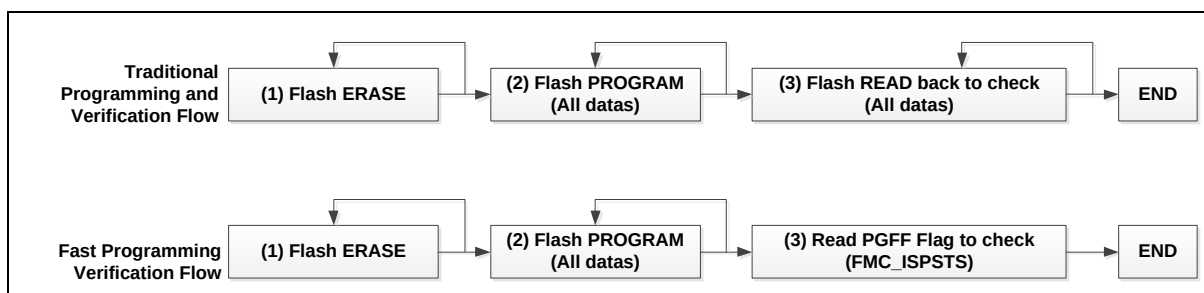


Figure 6.4-13 Verification Flow

The fast Flash programming verification function is released for 32-bit programming, but multi-word programming operation is not suitable due to the embedded Flash HV (High Voltage) of continue programming.

6.4.4.14 CRC32 Checksum Calculation

This chip supports the CRC32 checksum calculation, and help user quickly check the memory content

includes APROM and LDROM. The CRC32 polynomial is

$$\text{CRC-32} : X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$$

With seed = 0xFFFF_FFFF

The CRC32 checksum calculation flow is shown in .

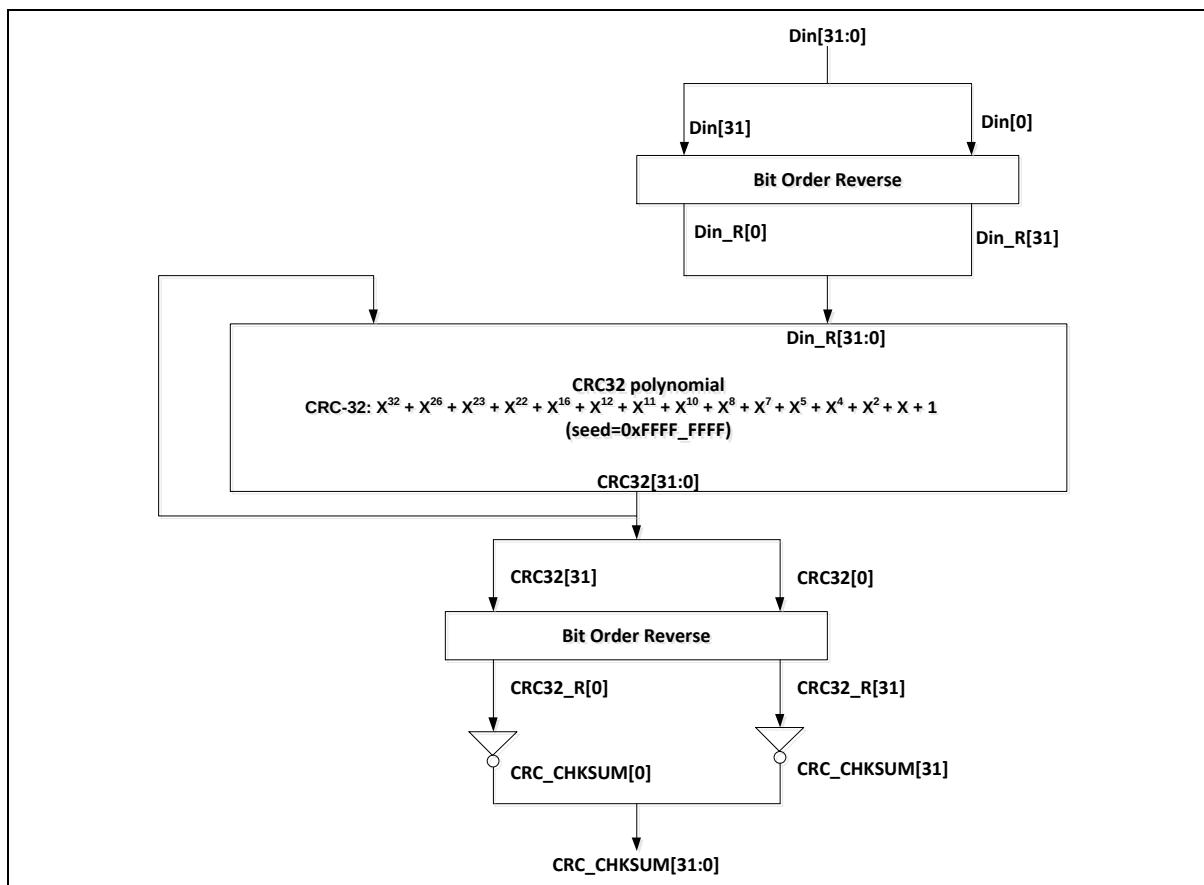


Figure 6.4-14 Flash CRC32 Checksum Calculation

Three steps complete this CRC32 checksum calculation.

1. Step 1. Perform ISP "Run Memory CRC32 Checksum" operation
2. Step 2. Perform ISP "Read Memory CRC32 Checksum" operation
3. Step 3. Read FMC_ISPDAT to get checksum.

In Step 1, user has to set the memory starting address (FMC_ISPADDR) and size (FMC_ISPDAT) to calculate. Both address and size have to be 4 Kbytes alignment, the size should be ≥ 4 Kbytes and the starting address includes APROM and LDROM.

In Step 2, the FMC_ISPADDR should be kept as the same as Step 1.

In Step 3, the checksum is read from FMC_ISPDAT. If the checksum is 0x0000_0000, there is one of two conditions (1) Checksum calculation is in-progress, (2) Address and size is over device limitation

When Flash is locked, user can still calculate the CRC of the data in Flash. It is useful for checking data corrupt or data retention issue under Flash security lock mode or XOM active mode

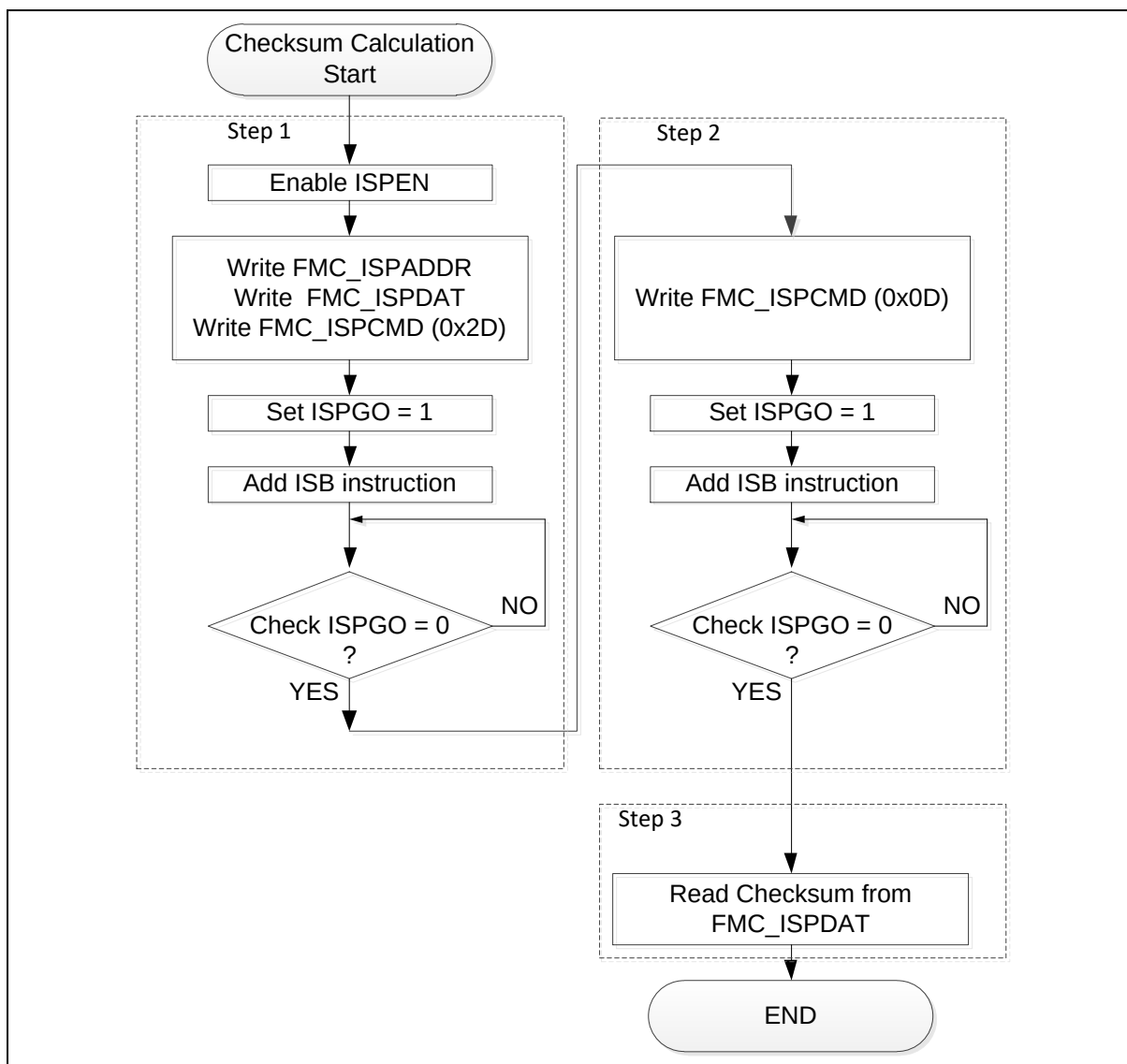


Figure 6.4-15 CRC-32 Checksum Calculation Flow

6.4.4.15 Flash All One Verification

This chip supports the Flash all one verification function to help user quickly check a memory block content blanking for APROM and LDROM after Flash erase operation.

Two steps complete this Flash all one verification.

Two-step flow:

1. Step 1.Perform ISP “Run Flash All One Verification” operation
2. Step 2.Read ALLONE(FMC_ISPSTA[7])bit to get the verification result
 - ALLONE: 1, all of Flash bits are 1 in verification block memory.
 - ALLONE: 0, Flash bits are not all 1 in verification block memory.

In Step 1, user has to set the memory starting address (FMC_ISPADDR) and size (FMC_ISPDAT) to

verify. Both address and size have to be 512 bytes alignment, the size should be ≥ 512 bytes and the starting address includes APROM and LDROM.

In Step 2, the FMC_ISPADDR should be kept as the same as Step 1.

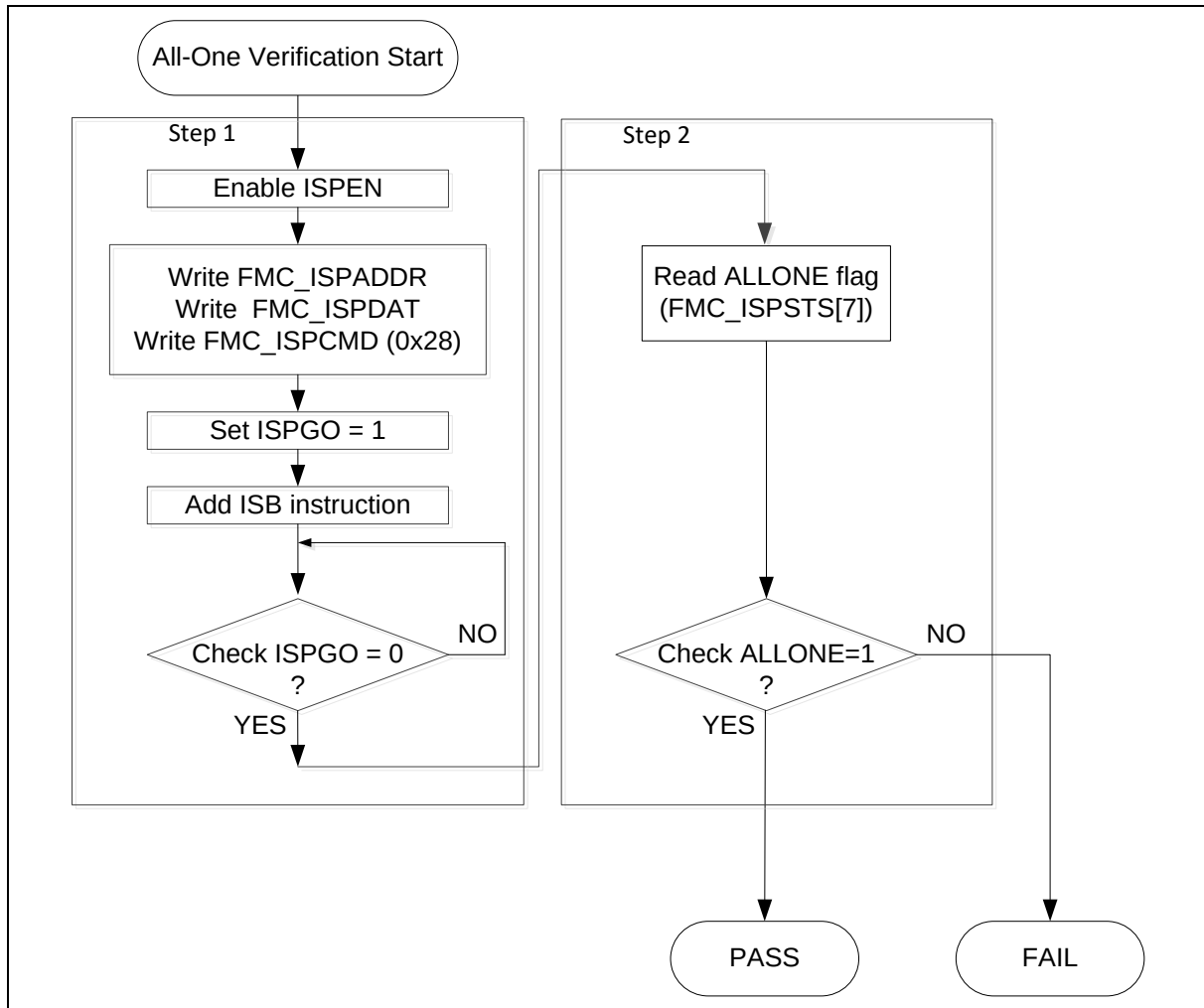


Figure 6.4-16 All-One Verification Flow

6.4.5 Register Map

R: read only, W: write only, R/W: both read and write

Register	Offset	R/W	Description	Reset Value
FMC Base Address FMC_BA = 0x4000_C000				
FMC_ISPCTL	FMC_BA+0x00	R/W	ISP Control Register	0x0000_0000
FMC_ISPADDR	FMC_BA+0x04	R/W	ISP Address Register	0x0000_0000
FMC_ISPDAT	FMC_BA+0x08	R/W	ISP Data Register	0x0000_0000
FMC_ISPCMD	FMC_BA+0x0C	R/W	ISP Command Register	0x0000_0000
FMC_ISPTRG	FMC_BA+0x10	R/W	ISP Trigger Control Register	0x0000_0000
FMC_FTCTL	FMC_BA+0x18	R/W	Flash Access Time Control Register	0x0000_0000
FMC_ISPSTS	FMC_BA+0x40	R/W	ISP Status Register	0x0000_000X
FMC_CYCCTL	FMC_BA+0x4C	R/W	Flash Access Cycle Control Register	0x0000_0001
FMC_MPDAT0	FMC_BA+0x80	R/W	ISP Data0 Register	0x0000_0000
FMC_MPDAT1	FMC_BA+0x84	R/W	ISP Data1 Register	0x0000_0000
FMC_MPDAT2	FMC_BA+0x88	R/W	ISP Data2 Register	0x0000_0000
FMC_MPDAT3	FMC_BA+0x8C	R/W	ISP Data3 Register	0x0000_0000
FMC_MPSTS	FMC_BA+0xC0	R	ISP Multi-program Status Register	0x0000_0000
FMC_MPADDR	FMC_BA+0xC4	R	ISP Multi-program Address Register	0x0000_0000
FMC_XOMR0STS0	FMC_BA+0xD0	R	XOM Region 0 Status Register 0	0x00FF_FFFF
FMC_XOMR0STS1	FMC_BA+0xD4	R	XOM Region 0 Status Register 1	0x0000_001F
FMC_XOMSTS	FMC_BA+0xE0	R	XOM Status Register	0x0000_0000

6.4.6 Register Description

ISP Control Register (FMC_ISPCTL)

Register	Offset	R/W	Description	Reset Value
FMC_ISPCTL	FMC_BA+0x00	R/W	ISP Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved	ISPFF	LDUEN	CFGUEN	APUEN	Reserved	BS	ISPEN

Bits	Description
[31:7]	Reserved Reserved.
[6]	ISPFF ISP Fail Flag (Write Protect) This bit is set by hardware when a triggered ISP meets any of the following conditions: This bit needs to be cleared by writing 1 to it. (1) APROM writes to itself if APUEN is set to 0. (2) LDROM writes to itself if LDUEN is set to 0. (3) CONFIG is erased/programmed if CFGUEN is set to 0. (4) Page Erase command at LOCK mode with ICE connection (5) Erase or Program command at brown-out detected (6) Destination address is illegal, such as over an available range. (7) Invalid ISP commands (8) ISP CMD in XOM region, except mass erase, page erase and chksum command (9) The wrong setting of page erase ISP CMD in XOM (10) Violate XOM setting one time protection Note: This bit is write protected. Refer to the SYS_REGLCTL register.
[5]	LDUEN LDROM Update Enable Bit (Write Protect) 0 = LDROM cannot be updated. 1 = LDROM can be updated. Note: This bit is write protected. Refer to the SYS_REGLCTL register.
[4]	CFGUEN CONFIG Update Enable Bit (Write Protect) 0 = CONFIG cannot be updated. 1 = CONFIG can be updated. Note: This bit is write protected. Refer to the SYS_REGLCTL register.

[3]	APUEN	APROM Update Enable Bit (Write Protect) 0 = APROM cannot be updated when the chip runs in APROM. 1 = APROM can be updated when the chip runs in APROM. Note: This bit is write protected. Refer to the SYS_REGLCTL register.
[2]	Reserved	Reserved.
[1]	BS	Boot Select (Write Protect) /clear this bit to select next booting from LDROM/APROM, respectively. This bit also functions as chip booting status flag, which can be used to check where chip booted from. This bit is initiated with the inversed value of CBS[1] (CONFIG0[7]) after any reset is happened except CPU reset or system reset is happened 0 = Booting from APROM. 1 = Booting from LDROM. Note: This bit is write protected. Refer to the SYS_REGLCTL register.
[0]	ISPEN	ISP Enable Bit (Write Protect) ISP function enable bit. Set this bit to enable ISP function. 0 = ISP function Disabled. 1 = ISP function Enabled. Note: This bit is write protected. Refer to the SYS_REGLCTL register.

ISP Address (FMC ISPADDR)

Register	Offset	R/W	Description	Reset Value
FMC_ISPADDR	FMC_BA+0x04	R/W	ISP Address Register	0x0000_0000

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

Bits	Description
[31:0]	ISP Address The NuMicro® M23 series is equipped with embedded Flash. ISPADDR[1:0] must be kept 00 for ISP 32-bit operation. ISPADDR[2:0] must be kept 000 for ISP 64-bit operation. For Checksum Calculation command, this field is the Flash starting address for checksum calculation, 512 bytes alignment is necessary for checksum calculation.

ISP Data Register (FMC_ISPDAT)

Register	Offset	R/W	Description	Reset Value
FMC_ISPDAT	FMC_BA+0x08	R/W	ISP Data Register	0x0000_0000

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

Bits	Description
[31:0]	ISP Data Write data to this register before ISP program operation. Read data from this register after ISP read operation. For Run Checksum Calculation command, ISPDAT is the memory size (byte) and 512 bytes alignment. For ISP Read Checksum command, ISPDAT is the checksum result. If ISPDAT = 0x0000_0000, it means that (1) the checksum calculation is in progress, (2) the memory range for checksum calculation is incorrect.

ISP Command Register (FMC_ISPCMD)

Register	Offset	R/W	Description	Reset Value
FMC_ISPCMD	FMC_BA+0x0C	R/W	ISP Command Register	0x0000_0000

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

Bits	Description	
[31:7]	Reserved	Reserved.
[6:0]	CMD	ISP Command ISP command table is shown below: 0x00= FLASH Read. 0x04= Read Unique ID. 0x0B= Read Company ID. 0x0C= Read Device ID. 0x0D= Read Checksum. 0x21= FLASH 32-bit Program. 0x22= FLASH Page Erase. 0x27= FLASH Multi-Word Program. 0x28= Run Flash All-One Verification. 0x2D= Run Checksum Calculation. 0x2E= Vector Remap. The other commands are invalid.

ISP Trigger Control Register (FMC_ISPTRG)

Register	Offset	R/W	Description	Reset Value
FMC_ISPTRG	FMC_BA+0x10	R/W	ISP Trigger Control Register	0x0000_0000

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

Bits	Description
[31:1]	Reserved Reserved.
[0]	ISPGO ISP Start Trigger (Write Protect) Write 1 to start ISP operation and this bit will be cleared to 0 by hardware automatically when ISP operation is finished. 0 = ISP operation is finished. 1 = ISP is progressed. Note: This bit is write-protected. Refer to the SYS_REGLCTL register.

Flash Access Time Control Register (FMC_FTCTL)

Register	Offset	R/W	Description	Reset Value
FMC_FTCTL	FMC_BA+0x18	R/W	Flash Access Time Control Register	0x0000_0000

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

Bits	Description
[31:10]	Reserved Reserved.
[9]	CACHEINV Flash Cache Invalidation (Write Protect) 0 = Flash Cache Invalidation finished (default). 1 = Flash Cache Invalidation. Note 1: Write 1 to start cache invalidation. The value will be changed to 0 once the process finishes. Note 2: This bit is write-protected. Refer to the SYS_REGLCTL register.
[8:0]	Reserved Reserved.

ISP Status Register (FMC_ISPSTS)

Register	Offset	R/W	Description	Reset Value
FMC_ISPSTS	FMC_BA+0x40	R/W	ISP Status Register	0x0000_000X

31	30	29	28	27	26	25	24
Reserved		VECMAP					
23	22	21	20	19	18	17	16
VECMAP							
15	14	13	12	11	10	9	8
VECMAP							Reserved
7	6	5	4	3	2	1	0
ALLONE	ISPFF	PGFF	Reserved		CBS		ISPBUSY

Bits	Description
[31:30]	Reserved Reserved.
[29:9]	VECMAP Vector Page Mapping Address (Read Only) All access to 0x0000_0000~0x0000_01FF is remapped to the Flash memory or SRAM address {VECMAP[20:0], 9'h000} ~ {VECMAP[20:0], 9'h1FF}. VECMAP [20:19] = 2'b00 system vector address is mapped to Flash memory. VECMAP [20:19] = 2'b10 system vector address is mapped to SRAM memory. VECMAP [18:12] should be 0.
[8]	Reserved Reserved.
[7]	ALLONE Flash All-one Verification Flag This bit is set by hardware if all of Flash bits are 1, and clear if Flash bits are not all 1 after "Run Flash All-One Verification" complete; this bit also can be clear by writing 1 0 = Flash bits are not all 1 after "Run Flash All-One Verification" complete. 1 = All of Flash bits are 1 after "Run Flash All-One Verification" complete.
[6]	ISPFF ISP Fail Flag (Write Protect) This bit is the mirror of ISPFF (FMC_ISPCTL[6]), it needs to be cleared by writing 1 to FMC_ISPCTL[6] or FMC_ISPSTS[6]. This bit is set by hardware when a triggered ISP meets any of the following conditions: (1) APROM writes to itself if APUEN is set to 0. (2) LDROM writes to itself if LDUEN is set to 0. (3) CONFIG is erased/programmed if CFGUEN is set to 0. (4) Page Erase command at LOCK mode with ICE connection (5) Erase or Program command at brown-out detected (6) Destination address is illegal, such as over an available range. (7) Invalid ISP commands (8) ISP CMD in XOM region, except mass erase, page erase and chksum command (9) The wrong setting of page erase ISP CMD in XOM (10) Violate XOM setting one time protection Note: This bit is write-protected. Refer to the SYS_REGLCTL register.

[5]	PGFF	Flash Program with Fast Verification Flag (Read Only) This bit is set if data is mismatched at ISP programming verification. This bit is clear by performing ISP Flash erase or ISP read CID operation 0 = Flash Program is success. 1 = Flash Program is fail. Program data is different with data in the Flash memory
[4:3]	Reserved	Reserved.
[2:1]	CBS	Boot Selection of CONFIG (Read Only) This bit is initiated with the CBS (CONFIG0[7:6]) after any reset is happened except CPU reset or system reset is happened. 00 = LDROM with IAP mode. 01 = LDROM without IAP mode. 10 = APROM with IAP mode. 11 = APROM without IAP mode.
[0]	ISPBUSY	ISP Busy Flag (Read Only) Write 1 to start ISP operation and this bit will be cleared to 0 by hardware automatically when ISP operation is finished. This bit is the mirror of ISPGO(FMC_ISPTRG[0]). 0 = ISP operation is finished. 1 = ISP is progressed.

Flash Access Cycle Control Register (FMC_CYCCTL)

Register	Offset	R/W	Description	Reset Value
FMC_CYCCTL	FMC_BA+0x4C	R/W	Flash Access Cycle Control Register	0x0000_0001

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

Bits	Description
[31:4]	Reserved Reserved.
[3:0]	CYCLE Flash Access Cycle Control (Write Protect) 0001 = CPU access with zero wait cycle if cache hit or cache is disabled; Flash access cycle is 1;. The HCLK working frequency range is <19 MHz; 0010 = CPU access with one wait cycles if cache miss; Flash access cycle is 2;. The optimized HCLK working frequency range is 18~33 MHz 0011 = CPU access with two wait cycles if cahce miss; Flash access cycle is 3;. The optimized HCLK working frequency range is 33~50 MHz Note: This bit is write protected. Refer to the SYS_REGLCTL register.

ISP Data 0 Register (FMC_MPDAT0)

Register	Offset	R/W	Description	Reset Value
FMC_MPDAT0	FMC_BA+0x80	R/W	ISP Data0 Register	0x0000_0000

31	30	29	28	27	26	25	24
ISPDAT0							
23	22	21	20	19	18	17	16
ISPDAT0							
15	14	13	12	11	10	9	8
ISPDAT0							
7	6	5	4	3	2	1	0
ISPDAT0							

Bits	Description
[31:0]	ISP Data 0 ISPDAT0 This register is the first 32-bit data for 32-bit/multi-word programming, and it is also the mirror of FMC_ISPDAT, both registers keep the same data.

ISP Data 1 Register (FMC MPDAT1)

Register	Offset	R/W	Description	Reset Value
FMC_MPDAT1	FMC_BA+0x84	R/W	ISP Data1 Register	0x0000_0000

31	30	29	28	27	26	25	24
ISP DAT1							
23	22	21	20	19	18	17	16
ISP DAT1							
15	14	13	12	11	10	9	8
ISP DAT1							
7	6	5	4	3	2	1	0
ISP DAT1							

Bits	Description	
[31:0]	ISP DAT1	ISP Data 1 This register is the second 32-bit data for multi-word programming.

ISP Data 2 Register (FMC MPDAT2)

Register	Offset	R/W	Description	Reset Value
FMC_MPDAT2	FMC_BA+0x88	R/W	ISP Data2 Register	0x0000_0000

31	30	29	28	27	26	25	24
ISPDAT2							
23	22	21	20	19	18	17	16
ISPDAT2							
15	14	13	12	11	10	9	8
ISPDAT2							
7	6	5	4	3	2	1	0
ISPDAT2							

Bits	Description
[31:0]	ISPDAT2 ISP Data 2 This register is the third 32-bit data for multi-word programming.

ISP Data 3 Register (FMC MPDAT3)

Register	Offset	R/W	Description	Reset Value
FMC_MPDAT3	FMC_BA+0x8C	R/W	ISP Data3 Register	0x0000_0000

31	30	29	28	27	26	25	24
ISPDAT3							
23	22	21	20	19	18	17	16
ISPDAT3							
15	14	13	12	11	10	9	8
ISPDAT3							
7	6	5	4	3	2	1	0
ISPDAT3							

Bits	Description	
[31:0]	ISPDAT3	ISP Data 3 This register is the fourth 32-bit data for multi-word programming.

ISP Multi-program Status Register (FMC_MPSTS)

Register	Offset	R/W	Description	Reset Value
FMC_MPSTS	FMC_BA+0xC0	R	ISP Multi-program Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
D3	D2	D1	D0	Reserved	ISPFF	PPGO	MPBUSY

Bits	Description
[31:8]	Reserved Reserved.
[7]	D3 ISP DATA 3 Flag (Read Only) This bit is set when FMC_MPDAT3 is written and auto-clear to 0 when the FMC_MPDAT3 data is programmed to Flash complete. 0 = FMC_MPDAT3 register is empty, or program to Flash complete. 1 = FMC_MPDAT3 register has been written, and not program to Flash complete.
[6]	D2 ISP DATA 2 Flag (Read Only) This bit is set when FMC_MPDAT2 is written and auto-clear to 0 when the FMC_MPDAT2 data is programmed to Flash complete. 0 = FMC_MPDAT2 register is empty, or program to Flash complete. 1 = FMC_MPDAT2 register has been written, and not program to Flash complete.
[5]	D1 ISP DATA 1 Flag (Read Only) This bit is set when FMC_MPDAT1 is written and auto-clear to 0 when the FMC_MPDAT1 data is programmed to Flash complete. 0 = FMC_MPDAT1 register is empty, or program to Flash complete. 1 = FMC_MPDAT1 register has been written, and not program to Flash complete.
[4]	D0 ISP DATA 0 Flag (Read Only) This bit is set when FMC_MPDAT0 is written and auto-clear to 0 when the FMC_MPDAT0 data is programmed to Flash complete. 0 = FMC_MPDAT0 register is empty, or program to Flash complete. 1 = FMC_MPDAT0 register has been written, and not program to Flash complete.
[3]	Reserved Reserved.

[2]	ISPFF	ISP Fail Flag (Read Only) This bit is the mirror of ISPFF (FMC_ISPCTL[6]), it needs to be cleared by writing 1 to FMC_ISPCTL[6] or FMC_ISPSTS[6]. This bit is set by hardware when a triggered ISP meets any of the following conditions: (1) APROM writes to itself if APUEN is set to 0. (2) LDROM writes to itself if LDUEN is set to 0. (3) CONFIG is erased/programmed if CFGUEN is set to 0. (4) Page Erase command at LOCK mode with ICE connection (5) Erase or Program command at brown-out detected (6) Destination address is illegal, such as over an available range. (7) Invalid ISP commands.
[1]	PPGO	ISP Multi-program Status (Read Only) 0 = ISP multi-word program operation is not active. 1 = ISP multi-word program operation is in progress.
[0]	MPBUSY	ISP Multi-word Program Busy Flag (Read Only) Write 1 to start ISP Multi-Word program operation and this bit will be cleared to 0 by hardware automatically when ISP Multi-Word program operation is finished. This bit is the mirror of ISPGO(FMC_ISPTRG[0]). 0 = ISP Multi-Word program operation is finished. 1 = ISP Multi-Word program operation is progressed.

ISP Multi-word Program Address Register (FMC MPADDR)

Register	Offset	R/W	Description	Reset Value
FMC_MPADDR	FMC_BA+0xC4	R	ISP Multi-program Address Register	0x0000_0000

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

Bits	Description
[31:0]	MPADDR ISP Multi-word Program Address MPADDR is the address of ISP multi-word program operation when ISPGO flag is 1. MPADDR will keep the final ISP address when ISP multi-word program is complete.

XOM Region0 Status Register 0 (FMC_XOMR0STS0)

Register	Offset	R/W	Description	Reset Value
FMC_XOMR0STS0	FMC_BA+0xD0	R	XOM Region 0 Status Register 0	0x00FF_FFFF

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

Bits	Description	
[31:24]	Reserved	Reserved.
[23:0]	BASE	XOM Region 0 Base Address (Page-aligned) BASE is the base address of XOM Region 0.

XOM Region0 Status Register 1 (FMC_XOMR0STS1)

Register	Offset	R/W	Description	Reset Value
FMC_XOMR0STS1	FMC_BA+0xD4	R	XOM Region 0 Status Register 1	0x0000_001F

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

Bits	Description	
[31:9]	Reserved	Reserved.
[8:0]	SIZE	XOM Region 0 Size (Page-aligned) SIZE is the page number of XOM Region 0.

XOM Status Register (FMC_XOMSTS)

Register	Offset	R/W	Description	Reset Value
FMC_XOMSTS	FMC_BA+0xE0	R	XOM Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved			XOMPEF	Reserved			XOMR0ON

Bits	Description	
[31:5]	Reserved	Reserved.
[4]	XOMPEF	XOM Page Erase Function Fail XOM page erase function status. If XOMPEF is set to 1, user needs to erase XOM region again. 0 = Success. 1 = Fail.
[3:1]	Reserved	Reserved.
[0]	XOMR0ON	XOM Region 0 On XOM Region 0 active status. 0 = No active. 1 = XOM region 0 is active.

6.5 General Purpose I/O (GPIO)

6.5.1 Overview

This chip has up to 38 General Purpose I/O pins to be shared with other function pins depending on the chip configuration. These 38 pins are arranged in 4 ports named as PA, PB, PC, and PF. PA and PB has 12 pins on port. PC has 7 pins on port. PF has 7 pins on port. Each of the 38 pins is independent and has the corresponding register bits to control the pin mode function and data.

The I/O type of each of I/O pins can be configured by software individually as Input, Push-pull output, Open-drain output or Quasi-bidirectional mode. After the chip is reset, the I/O mode of all pins are depending on CIOINI (CONFIG0[10]). Each I/O pin has a very weakly individual pull-up resistor which is about 50 k Ω . Please refer to the M253 Datasheet for detailed pin operation voltage information about V_{DD} electrical characteristics.

6.5.2 Features

- Four I/O modes:
 - Quasi-bidirectional mode
 - Push-Pull Output mode
 - Open-Drain Output mode
 - Input only with high impedance mode
- TTL/Schmitt trigger input selectable
- I/O pin can be configured as interrupt source with edge/level setting
- Supports High Drive and High Slew Rate I/O mode
- Configurable default I/O mode of all pins after reset by CIOINI (CONFIG0[10]) setting
 - CIOINI = 0, all GPIO pins in Quasi-bidirectional mode after chip reset
 - CIOINI = 1, all GPIO pins in input mode after chip reset
- Support independent pull-up and pull-down control
- Enabling the pin interrupt function will also enable the wake-up function
- Improve access efficiency by using single cycle I/O bus

6.5.3 Block Diagram

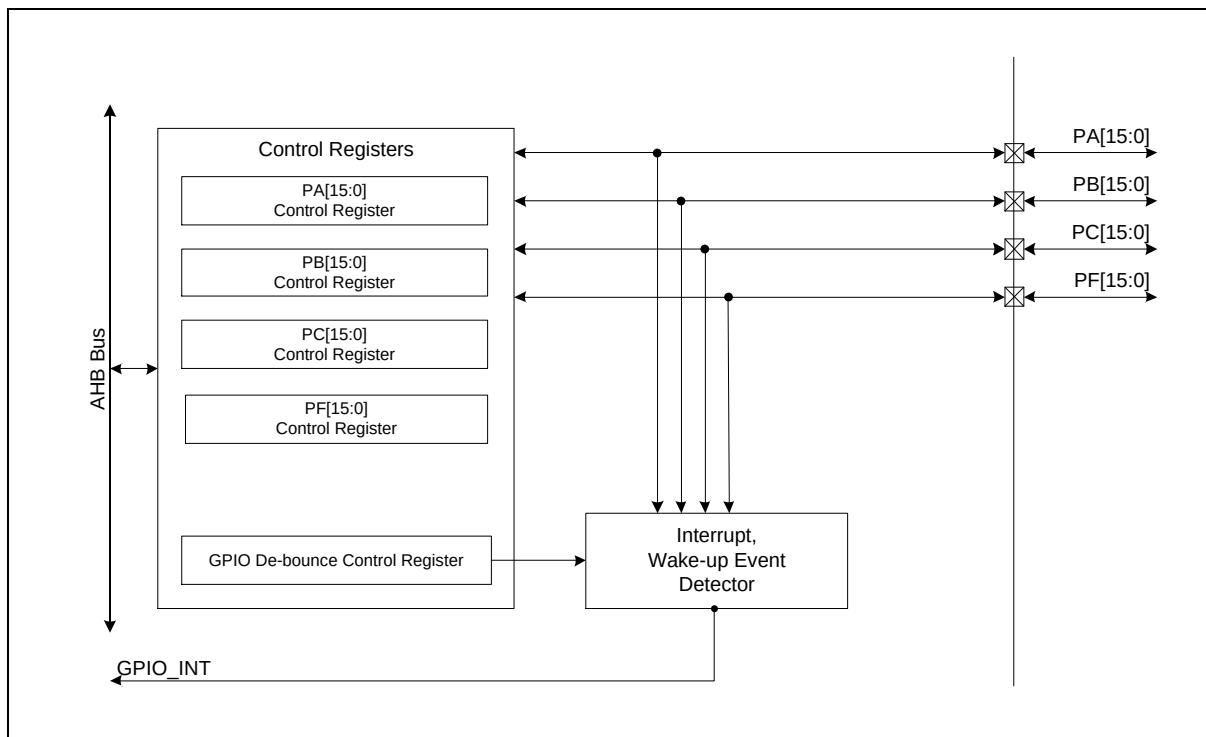


Figure 6.5-1 GPIO Controller Block Diagram

Note: The PA.12/ PA.13/ PA.14/ PA.15/ PB.8/ PB.9/ PB.10/ PB.11/ PC.6/ PC.7/ PC.8/ PC.9/ PC.10/ PC.11/ PC.12/ PC.13/ PC.15/ PF.6/ PF.7/ PF.8/ PF.9/ PF.10/ PF.11/ PF.12/ PF.13/ PF.14 pins are not available.

6.5.4 Basic Configuration

- Reset configuration
 - Reset GPIO in GPIORST (SYS_IPRST1[1])
- Pin configuration

Group	Pin Name	GPIO	MFP
INT0	INT0	PA.6, PB.5	MFP15
INT1	INT1	PA.7, PB.4	MFP15
INT2	INT2	PB.3	MFP15
INT3	INT3	PB.2	MFP15
INT4	INT4	PB.6	MFP13
		PA.8	MFP15
		PF.15	MFP15
INT5	INT5	PB.7	MFP13

6.5.5 Functional Description

6.5.5.1 Input Mode

Set $MODE_n$ ($Px\_MODE[2n+1:2n]$) to 00 as the $Px.n$ pin is in Input mode and the I/O pin is in tri-state (high impedance) without output drive capability. The PIN ($Px\_PIN[n]$) value reflects the status of the corresponding port pins.

Each I/O pin includes an internal resistor. Set ($Px\_PUSEL[2n+1:2n]$) to 01 to enable internal pull-up resistor and ($Px\_PUSEL[2n+1:2n]$) to 10 to enable internal pull-down resistor. Setting both pull-up/down is not available.

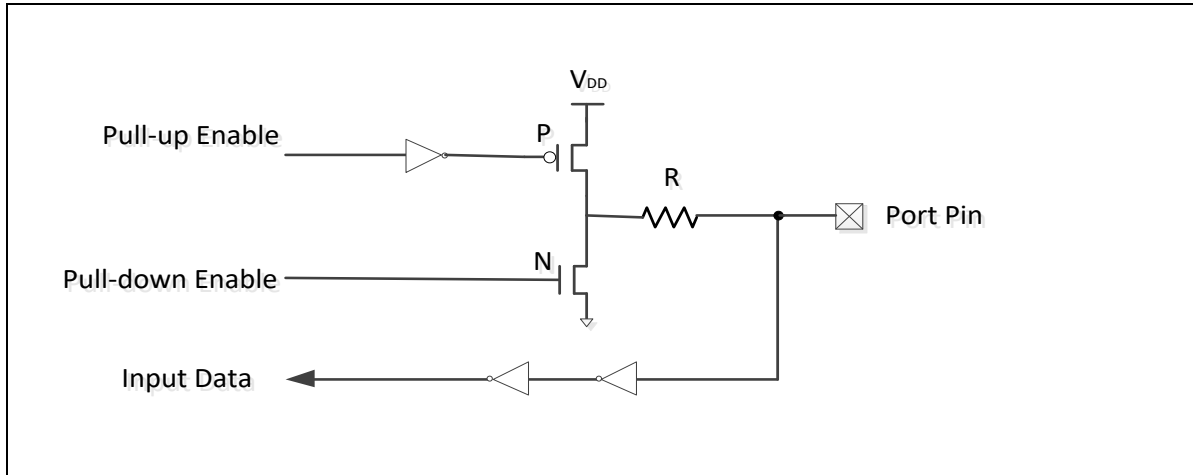


Figure 6.5-2 Input Mode

6.5.5.2 Push-pull Output Mode

Figure 6.5-3 shows the diagram of Push-pull Output Mode. Set $MODE_n$ ($Px\_MODE[2n+1:2n]$) to 01 as $Px.n$ pin is in Push-pull Output mode and the I/O pin supports digital output function with source/sink current capability. The bit value in the corresponding DOUT ($Px\_DOUT[n]$) is driven on the pin.

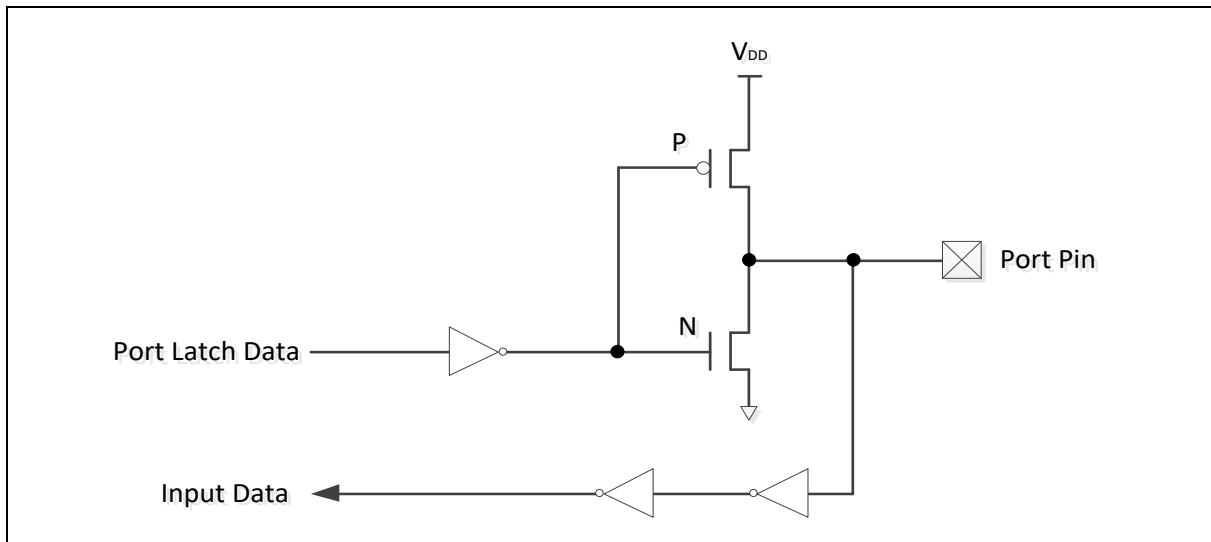


Figure 6.5-3 Push-Pull Output

6.5.5.3 Open-drain Mode

Figure 6.5-4 shows the diagram of Open-drain Mode. Set $MODE_n$ ($Px\_MODE[2n+1:2n]$) to 10 as the $Px.n$ pin is in Open-drain mode and the digital output function of I/O pin supports only sink current capability. An external pull-up register is needed for driving high state. If the bit value in the corresponding DOUT ($Px\_DOUT[n]$) bit is 0, the pin drives a low output on the pin. If the bit value in the

corresponding DOUT (Px_DOUT[n]) bit is 1, the pin output drives high that is controlled by external pull high resistor.

Each I/O pin includes an internal resistor. Set (Px_PUSEL[2n+1:2n]) to 01 to enable internal pull-up resistor. Only pull-up is available in Open-drain mode.

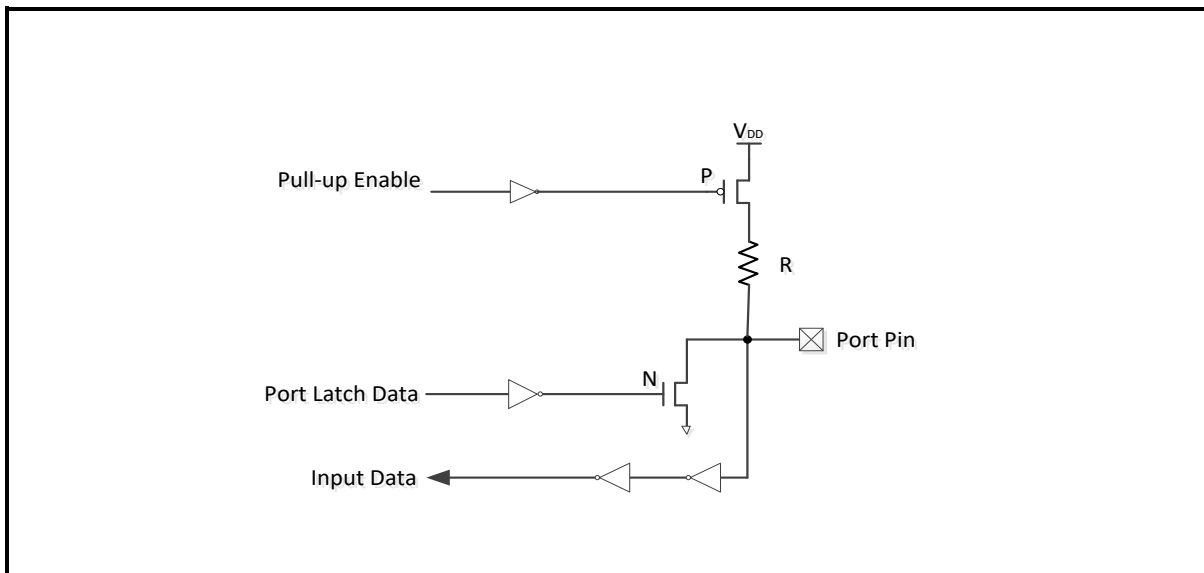


Figure 6.5-4 Open-Drain Output

6.5.5.4 Quasi-bidirectional Mode

Figure 6.5-5 shows the diagram of Quasi-bidirectional Mode. Set MODEn (Px_MODE[2n+1:2n]) to 11 as the Px.n pin is in Quasi-bidirectional mode and the I/O pin supports digital output and input function at the same time but the source current is only up to hundreds uA. Before the digital input function is performed the corresponding DOUT (Px_DOUT[n]) bit must be set to 1. The quasi-bidirectional output is common on the 80C51 and most of its derivatives. If the bit value in the corresponding DOUT (Px_DOUT[n]) bit is 0, the pin drives a low output on the pin. If the bit value in the corresponding DOUT (Px_DOUT[n]) bit is 1, the pin will check the pin value. If pin value is high, no action takes. If pin state is low, the pin will drive strong high with 2 clock cycles on the pin and then disable the strong output drive. Meanwhile, the pin status is controlled by internal pull-up resistor. Note that the source current capability in quasi-bidirectional mode, please refer to the M253 Datasheet for detailed information.

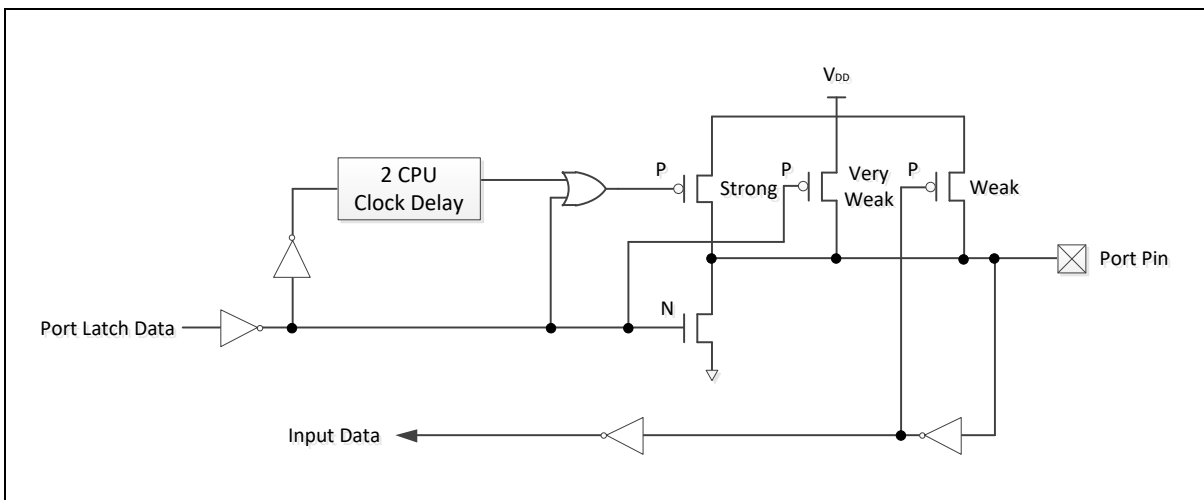


Figure 6.5-5 Quasi-Bidirectional I/O Mode

6.5.5.5 GPIO Interrupt and Wake-up Function

Each GPIO pin can be set as chip interrupt source by setting correlative RHIEN (Px_INTEN[n+16])/FLIEN (Px_INTEN[n]) bit and TYPE (Px_INTTYPE[n]). There are five types of interrupt conditions that can be selected: low level trigger, high level trigger, falling edge trigger, rising edge trigger and both rising and falling edge trigger. The GPIO can also be the chip wake-up source when chip enters Idle/Power-down mode. Level trigger source needs to keep its state until entering interrupt handler. The setting of wake-up trigger condition is the same as GPIO interrupt trigger.

6.5.5.6 GPIO De-bounce Function

GPIO de-bounce function can be used to sample interrupt input for each GPIO pin and prevent unexpected interrupt happened which caused by noise. GPIO de-bounce function only support edge detection trigger type. For edge trigger condition, there are three types of interrupt conditions that can be selected for de-bounce function: falling edge trigger, rising edge trigger and both rising and falling edge trigger. If user wants to use de-bounce function, de-bounce enable control register Px_DBEN must be set for corresponding GPIO pin. The de-bounce clock source can be HCLK or LIRC (38.4 kHz) by setting DBCLKSRC (Px_DBCTL[4]) register. And DBCLKSEL (Px_DBCTL[3:0]) register can control sampling cycle period.

Figure 6.5-56 shows GPIO rising edge trigger interrupt. The interval of time between the two valid sample signal is determined by DBCLKSRC (Px_DBCTL[4]) and DBCLKSEL (Px_DBCTL[3:0]). Each valid data from GPIO pin need to be sample twice. For rising edge setting, if pin status is low before setting DBEN (Px_DBEN), interrupt will happen when generating a pin high valid data. But, if pin status is high before setting DBEN (Px_DBEN), interrupt will happen when generating a pin low valid data first, and then generating a pin high valid data. For falling edge trigger, Figure 6.5-67 shows the situation is opposite to rising edge trigger.

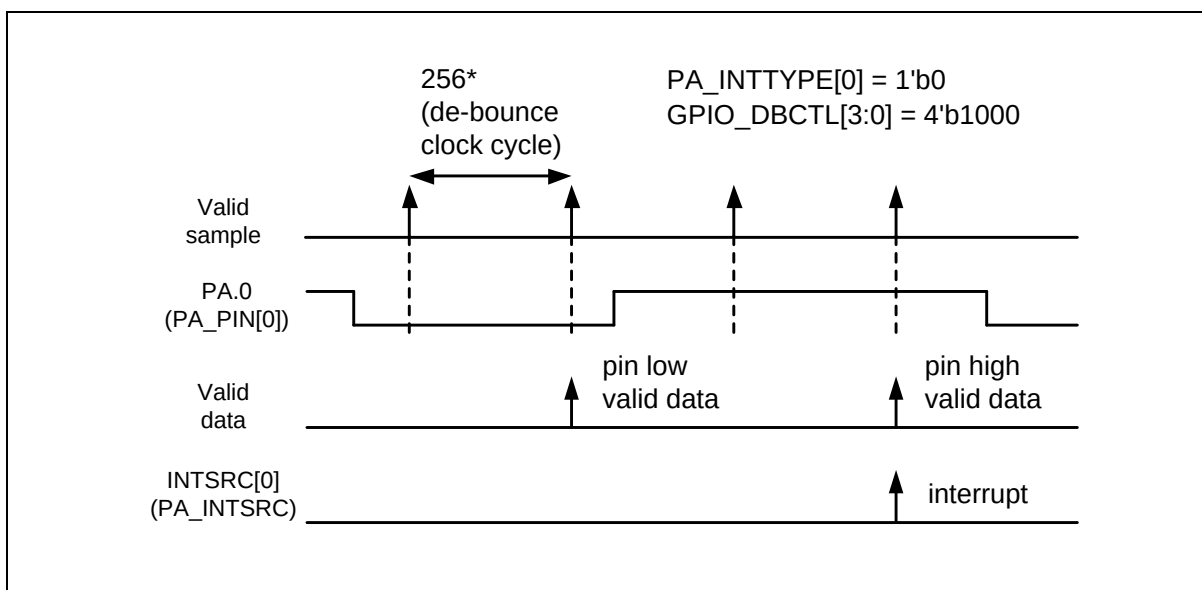


Figure 6.5-6 GPIO Rising Edge Trigger Interrupt

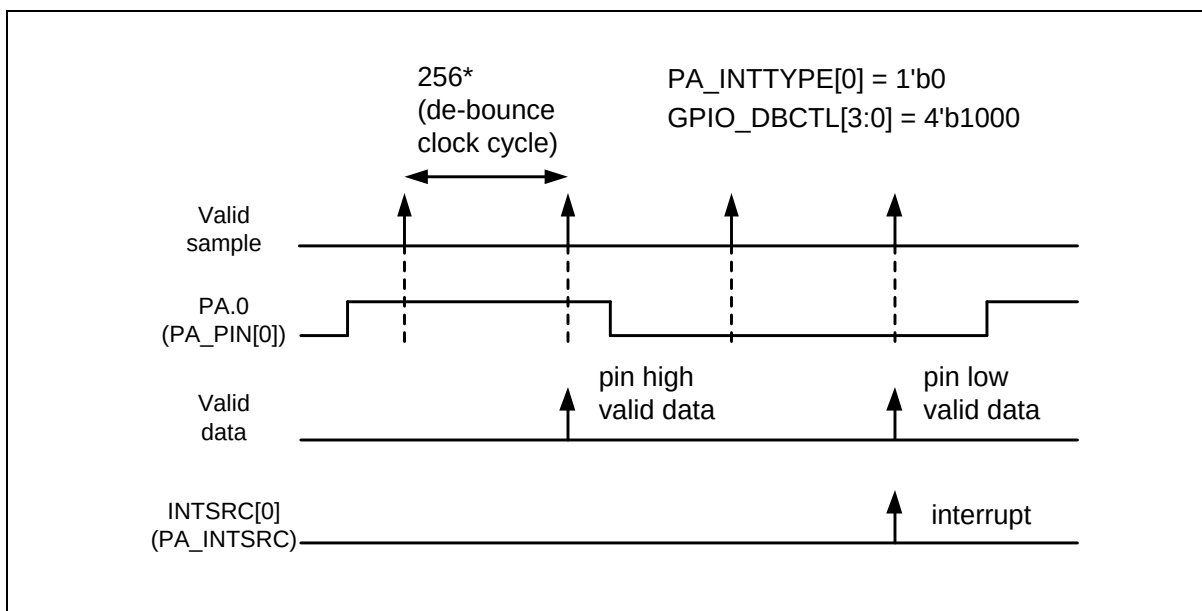


Figure 6.5-7 GPIO Falling Edge Trigger Interrupt

6.5.5.7 GPIO Digital Input Path Disable Control

User can disable GPIO digital input path by setting DINOFF (Px_DINOFF[n+16]). When GPIO digital input path is disabled, the digital input pin value PIN (Px_PIN[n]) is tied to low. By the way, the GPIO digital input path is forced disabled by hardware and DINOFF control is useless when I/O function is configured as ADC/ext_XTAL

6.5.6 Register Map

R: read only, W: write only, R/W: both read and write

Register	Offset	R/W	Description	Reset Value
GPIO Base Address: GPIO_BA = 0x4000_4000				
PA_MODE	GPIO_BA+0x000	R/W	PA I/O Mode Control	0XXXXX_XXXX
PA_DINOFF	GPIO_BA+0x004	R/W	PA Digital Input Path Disable Control	0x0000_0000
PA_DOUT	GPIO_BA+0x008	R/W	PA Data Output Value	0x0000_0FFF
PA_DATMSK	GPIO_BA+0x00C	R/W	PA Data Output Write Mask	0x0000_0000
PA_PIN	GPIO_BA+0x010	R	PA Pin Value	0x0000_XXXX
PA_DBEN	GPIO_BA+0x014	R/W	PA De-bounce Enable Control Register	0x0000_0000
PA_INTTYPE	GPIO_BA+0x018	R/W	PA Interrupt Trigger Type Control	0x0000_0000
PA_INTEN	GPIO_BA+0x01C	R/W	PA Interrupt Enable Control Register	0x0000_0000
PA_INTSRC	GPIO_BA+0x020	R/W	PA Interrupt Source Flag	0x0000_XXXX
PA_SMTEN	GPIO_BA+0x024	R/W	PA Input Schmitt Trigger Enable Register	0x0000_0000
PA_SLEWCTL	GPIO_BA+0x028	R/W	PA High Slew Rate Control Register	0x0000_0000
PA_PUSEL	GPIO_BA+0x030	R/W	PA Pull-up and Pull-down Selection Register	0x0000_0000
PB_MODE	GPIO_BA+0x040	R/W	PB I/O Mode Control	0XXXXX_XXXX
PB_DINOFF	GPIO_BA+0x044	R/W	PB Digital Input Path Disable Control	0x0000_0000
PB_DOUT	GPIO_BA+0x048	R/W	PB Data Output Value	0x0000_F0FF
PB_DATMSK	GPIO_BA+0x04C	R/W	PB Data Output Write Mask	0x0000_0000
PB_PIN	GPIO_BA+0x050	R	PB Pin Value	0x0000_XXXX
PB_DBEN	GPIO_BA+0x054	R/W	PB De-bounce Enable Control Register	0x0000_0000
PB_INTTYPE	GPIO_BA+0x058	R/W	PB Interrupt Trigger Type Control	0x0000_0000
PB_INTEN	GPIO_BA+0x05C	R/W	PB Interrupt Enable Control Register	0x0000_0000
PB_INTSRC	GPIO_BA+0x060	R/W	PB Interrupt Source Flag	0x0000_XXXX
PB_SMTEN	GPIO_BA+0x064	R/W	PB Input Schmitt Trigger Enable Register	0x0000_0000
PB_SLEWCTL	GPIO_BA+0x068	R/W	PB High Slew Rate Control Register	0x0000_0000
PB_PUSEL	GPIO_BA+0x070	R/W	PB Pull-up and Pull-down Selection Register	0x0000_0000
PC_MODE	GPIO_BA+0x080	R/W	PC I/O Mode Control	0XXXXX_XXXX
PC_DINOFF	GPIO_BA+0x084	R/W	PC Digital Input Path Disable Control	0x0000_0000
PC_DOUT	GPIO_BA+0x088	R/W	PC Data Output Value	0x0000_403F

PC_DATMSK	GPIO_BA+0x08C	R/W	PC Data Output Write Mask	0x0000_0000
PC_PIN	GPIO_BA+0x090	R	PC Pin Value	0x0000_XXXX
PC_DBEN	GPIO_BA+0x094	R/W	PC De-bounce Enable Control Register	0x0000_0000
PC_INTTYPE	GPIO_BA+0x098	R/W	PC Interrupt Trigger Type Control	0x0000_0000
PC_INTEN	GPIO_BA+0x09C	R/W	PC Interrupt Enable Control Register	0x0000_0000
PC_INTSRC	GPIO_BA+0x0A0	R/W	PC Interrupt Source Flag	0x0000_XXXX
PC_SMTEN	GPIO_BA+0x0A4	R/W	PC Input Schmitt Trigger Enable Register	0x0000_0000
PC_SLEWCTL	GPIO_BA+0x0A8	R/W	PC High Slew Rate Control Register	0x0000_0000
PC_PUSEL	GPIO_BA+0x0B0	R/W	PC Pull-up and Pull-down Selection Register	0x0000_0000
PF_MODE	GPIO_BA+0x140	R/W	PF I/O Mode Control	0xFFFF_XXXX
PF_DINOFF	GPIO_BA+0x144	R/W	PF Digital Input Path Disable Control	0x0000_0000
PF_DOUT	GPIO_BA+0x148	R/W	PF Data Output Value	0x0000_803F
PF_DATMSK	GPIO_BA+0x14C	R/W	PF Data Output Write Mask	0x0000_0000
PF_PIN	GPIO_BA+0x150	R	PF Pin Value	0x0000_XXXX
PF_DBEN	GPIO_BA+0x154	R/W	PF De-bounce Enable Control Register	0x0000_0000
PF_INTTYPE	GPIO_BA+0x158	R/W	PF Interrupt Trigger Type Control	0x0000_0000
PF_INTEN	GPIO_BA+0x15C	R/W	PF Interrupt Enable Control Register	0x0000_0000
PF_INTSRC	GPIO_BA+0x160	R/W	PF Interrupt Source Flag	0x0000_XXXX
PF_SMTEN	GPIO_BA+0x164	R/W	PF Input Schmitt Trigger Enable Register	0x0000_0000
PF_SLEWCTL	GPIO_BA+0x168	R/W	PF High Slew Rate Control Register	0x0000_0000
PF_PUSEL	GPIO_BA+0x170	R/W	PF Pull-up and Pull-down Selection Register	0x0000_0000
GPIO_DBCTL	GPIO_BA+0x440	R/W	Interrupt De-bounce Control Register	0x0027_0000
PAn_PDIO n=0,1,2,3,4,5,6,7,8,9,10,11	GPIO_BA+0x800+(0x04 * n)	R/W	GPIO PA.n Pin Data Input/Output Register	0x0000_000X
PBn_PDIO n=0,1,2,3,4,5,6,7,12,13,14,15	GPIO_BA+0x840+(0x04 * n)	R/W	GPIO PB.n Pin Data Input/Output Register	0x0000_000X
PCn_PDIO n=0,1,2,3,4,5,14	GPIO_BA+0x880+(0x04 * n)	R/W	GPIO PC.n Pin Data Input/Output Register	0x0000_000X
PFn_PDIO n=0,1,2,3,4,5,15	GPIO_BA+0x940+(0x04 * n)	R/W	GPIO PF.n Pin Data Input/Output Register	0x0000_000X

6.5.7 Register Description

Port A-F I/O Mode Control (Px_MODE)

Register	Offset	R/W	Description	Reset Value
PA_MODE	GPIO_BA+0x000	R/W	PA I/O Mode Control	0xFFFF_XXXX
PB_MODE	GPIO_BA+0x040	R/W	PB I/O Mode Control	0xFFFF_XXXX
PC_MODE	GPIO_BA+0x080	R/W	PC I/O Mode Control	0xFFFF_XXXX
PF_MODE	GPIO_BA+0x140	R/W	PF I/O Mode Control	0xFFFF_XXXX

31	30	29	28	27	26	25	24
MODE15		MODE14		MODE13		MODE12	
23	22	21	20	19	18	17	16
MODE11		MODE10		MODE9		MODE8	
15	14	13	12	11	10	9	8
MODE7		MODE6		MODE5		MODE4	
7	6	5	4	3	2	1	0
MODE3		MODE2		MODE1		MODE0	

Bits	Description
[2n+1:2n] n=0,1..15	Port A-F I/O Pin[n] Mode Control Determine each I/O mode of Px.n pins. 00 = Px.n is in Input mode. 01 = Px.n is in Push-pull Output mode. 10 = Px.n is in Open-drain Output mode. 11 = Px.n is in Quasi-bidirectional mode. Note 1: The initial value of this field is defined by CIOINI (CONFIG0 [10]). If CIOINI is set to 0, the default value is 0xFFFF_FFFF and all pins will be quasi-bidirectional mode after chip powered on. If CIOINI is set to 1, the default value is 0x0000_0000 and all pins will be input mode after chip powered on. Note 2: The PA.12~15/PB.8~11/PC.6~13/PC.15/PF.6~14 pin is ineffective. Note 3: If MFOS is enabled then GPIO mode setting is ignored.

Port A-F Digital Input Path Disable Control (Px_DINOFF)

Register	Offset	R/W	Description	Reset Value
PA_DINOFF	GPIO_BA+0x004	R/W	PA Digital Input Path Disable Control	0x0000_0000
PB_DINOFF	GPIO_BA+0x044	R/W	PB Digital Input Path Disable Control	0x0000_0000
PC_DINOFF	GPIO_BA+0x084	R/W	PC Digital Input Path Disable Control	0x0000_0000
PF_DINOFF	GPIO_BA+0x144	R/W	PF Digital Input Path Disable Control	0x0000_0000

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

Bits	Description
[n+16] n=0,1..15	DINOFF[n] Port A-F Pin[n] Digital Input Path Disable Bit Each of these bits is used to control if the digital input path of corresponding Px.n pin is disabled. If input is analog signal, users can disable Px.n digital input path to avoid input current leakage. 0 = Px.n digital input path Enabled. 1 = Px.n digital input path Disabled (digital input tied to low). Note: The PA.12~15/PB.8~11/PC.6~13/PC.15/PF.6~14 pin is ineffective.
[15:0]	Reserved Reserved.

Port A-F Data Output Value (Px DOUT)

Register	Offset	R/W	Description	Reset Value
PA_DOUT	GPIO_BA+0x008	R/W	PA Data Output Value	0x0000_0FFF
PB_DOUT	GPIO_BA+0x048	R/W	PB Data Output Value	0x0000_F0FF
PC_DOUT	GPIO_BA+0x088	R/W	PC Data Output Value	0x0000_403F
PF_DOUT	GPIO_BA+0x148	R/W	PF Data Output Value	0x0000_803F

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

Bits	Description
[31:16]	Reserved Reserved.
[n] n=0,1..15	DOUT[n] Port A-F Pin[n] Output Value Each of these bits controls the status of a Px.n pin when the Px.n is configured as Push-pull output, Open-drain output or Quasi-bidirectional mode. 0 = Px.n will drive Low if the Px.n pin is configured as Push-pull output, Open-drain output or Quasi-bidirectional mode. 1 = Px.n will drive High if the Px.n pin is configured as Push-pull output or Quasi-bidirectional mode. Note: The PA.12~15/PB.8~11/PC.6~13/PC.15/PF.6~14 pin is ineffective.

Port A-F Data Output Write Mask (Px DATMSK)

Register	Offset	R/W	Description	Reset Value
PA_DATMSK	GPIO_BA+0x00C	R/W	PA Data Output Write Mask	0x0000_0000
PB_DATMSK	GPIO_BA+0x04C	R/W	PB Data Output Write Mask	0x0000_0000
PC_DATMSK	GPIO_BA+0x08C	R/W	PC Data Output Write Mask	0x0000_0000
PF_DATMSK	GPIO_BA+0x14C	R/W	PF Data Output Write Mask	0x0000_0000

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

Bits	Description
[31:8]	Reserved Reserved.
[n] n=0,1..15	Port A-F Pin[n] Data Output Write Mask These bits are used to protect the corresponding DOUT (Px_DOUT[n]) bit. When the DATMSK (Px_DATMSK[n]) bit is set to 1, the corresponding DOUT (Px_DOUT[n]) bit is protected. If the write signal is masked, writing data to the protect bit is ineffective. 0 = Corresponding DOUT (Px_DOUT[n]) bit can be updated. 1 = Corresponding DOUT (Px_DOUT[n]) bit protected. Note 1: This function only protects the corresponding DOUT (Px_DOUT[n]) bit, and will not protect the corresponding PDIO (Pxn_PDIO[0]) bit. Note 2: The PA.12~15/PB.8~11/PC.6~13/PC.15/PF.6~14 pin is ineffective.

Port A-F Pin Value (Px PIN)

Register	Offset	R/W	Description	Reset Value
PA_PIN	GPIO_BA+0x010	R	PA Pin Value	0x0000_XXXX
PB_PIN	GPIO_BA+0x050	R	PB Pin Value	0x0000_XXXX
PC_PIN	GPIO_BA+0x090	R	PC Pin Value	0x0000_XXXX
PF_PIN	GPIO_BA+0x150	R	PF Pin Value	0x0000_XXXX

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

Bits	Description
[31:16]	Reserved Reserved.
[n] n=0,1..15	PIN[n] Port A-F Pin[n] Pin Value Each bit of the register reflects the actual status of the respective Px.n pin. 0 = The corresponding pin status is low. 1 = The corresponding pin status is high. Note: The PA.12~15/PB.8~11/PC.6~13/PC.15/PF.6~14 pin is ineffective.

Port A-F De-bounce Enable Control Register (Px_DBEN)

Register	Offset	R/W	Description	Reset Value
PA_DBEN	GPIO_BA+0x014	R/W	PA De-bounce Enable Control Register	0x0000_0000
PB_DBEN	GPIO_BA+0x054	R/W	PB De-bounce Enable Control Register	0x0000_0000
PC_DBEN	GPIO_BA+0x094	R/W	PC De-bounce Enable Control Register	0x0000_0000
PF_DBEN	GPIO_BA+0x154	R/W	PF De-bounce Enable Control Register	0x0000_0000

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

Bits	Description
[31:16]	Reserved Reserved.
[n] n=0,1..15	Port A-F Pin[n] Input Signal De-bounce Enable Bit The DBEN[n] bit is used to enable the de-bounce function for each corresponding bit. If the input signal pulse width cannot be sampled by continuous two de-bounce sample cycle, the input signal transition is seen as the signal bounce and will not trigger the interrupt. The de-bounce clock source is controlled by DBCLKSRC (GPIO_DBCTL [4]), one de-bounce sample cycle period is controlled by DBCLKSEL (GPIO_DBCTL [3:0]). 0 = Px.n de-bounce function Disabled. 1 = Px.n de-bounce function Enabled. The de-bounce function is valid only for edge triggered interrupt. If the interrupt mode is level triggered, the de-bounce enable bit is ineffective. Note: The PA.12~15/PB.8~11/PC.6~13/PC.15/PF.6~14 pin is ineffective.

Port A-F Interrupt Type Control (Px_INTTYPE)

Register	Offset	R/W	Description	Reset Value
PA_INTTYPE	GPIO_BA+0x018	R/W	PA Interrupt Trigger Type Control	0x0000_0000
PB_INTTYPE	GPIO_BA+0x058	R/W	PB Interrupt Trigger Type Control	0x0000_0000
PC_INTTYPE	GPIO_BA+0x098	R/W	PC Interrupt Trigger Type Control	0x0000_0000
PF_INTTYPE	GPIO_BA+0x158	R/W	PF Interrupt Trigger Type Control	0x0000_0000

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

Bits	Description
[31:16]	Reserved Reserved.
[n] n=0,1..15	Port A-F Pin[n] Edge or Level Detection Interrupt Trigger Type Control TYPE (Px_INTTYPE[n]) bit is used to control the triggered interrupt is by level trigger or by edge trigger. If the interrupt is by edge trigger, the trigger source can be controlled by de-bounce. If the interrupt is by level trigger, the input source is sampled by one HCLK clock and generates the interrupt. 0 = Edge trigger interrupt. 1 = Level trigger interrupt. If the pin is set as the level trigger interrupt, only one level can be set on the registers RHIEN (Px_INTEN[n+16])/FLIEN (Px_INTEN[n]). If both levels to trigger interrupt are set, the setting has no effect and no interrupt will occur. The de-bounce function is valid only for edge triggered interrupt. If the interrupt mode is level triggered, the de-bounce enable bit is ineffective. Note: The PA.12~15/PB.8~11/PC.6~13/PC.15/PF.6~14 pin is ineffective.

Port A-F Interrupt Enable Control Register (Px_INTEN)

Register	Offset	R/W	Description	Reset Value
PA_INTEN	GPIO_BA+0x01C	R/W	PA Interrupt Enable Control Register	0x0000_0000
PB_INTEN	GPIO_BA+0x05C	R/W	PB Interrupt Enable Control Register	0x0000_0000
PC_INTEN	GPIO_BA+0x09C	R/W	PC Interrupt Enable Control Register	0x0000_0000
PF_INTEN	GPIO_BA+0x15C	R/W	PF Interrupt Enable Control Register	0x0000_0000

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

Bits	Description
[n+16] n=0,1..15	Port A-F Pin[n] Rising Edge or High Level Interrupt Trigger Type Enable Bit The RHIEN (Px_INTEN[n+16]) bit is used to enable the interrupt for each of the corresponding input Px.n pin. Set bit to 1 also enable the pin wake-up function. When setting the RHIEN (Px_INTEN[n+16]) bit to 1 : If the interrupt is level trigger (TYPE (Px_INTTYPE[n]) bit is set to 1), the input Px.n pin will generate the interrupt while this pin state is at high level. If the interrupt is edge trigger (TYPE (Px_INTTYPE[n]) bit is set to 0), the input Px.n pin will generate the interrupt while this pin state changed from low to high. 0 = Px.n level high or low to high interrupt Disabled. 1 = Px.n level high or low to high interrupt Enabled. Note: The PA.12~15/PB.8~11/PC.6~13/PC.15/PF.6~14 pin is ineffective.
[n] n=0,1..15	Port A-F Pin[n] Falling Edge or Low Level Interrupt Trigger Type Enable Bit The FLIEN (Px_INTEN[n]) bit is used to enable the interrupt for each of the corresponding input Px.n pin. Set bit to 1 also enable the pin wake-up function. When setting the FLIEN (Px_INTEN[n]) bit to 1 : If the interrupt is level trigger (TYPE (Px_INTTYPE[n]) bit is set to 1), the input Px.n pin will generate the interrupt while this pin state is at low level. If the interrupt is edge trigger (TYPE (Px_INTTYPE[n]) bit is set to 0), the input Px.n pin will generate the interrupt while this pin state changed from high to low. 0 = Px.n level low or high to low interrupt Disabled. 1 = Px.n level low or high to low interrupt Enabled. Note: The PA.12~15/PB.8~11/PC.6~13/PC.15/PF.6~14 pin is ineffective.

Port A-F Interrupt Source Flag (Px_INTSRC)

Register	Offset	R/W	Description	Reset Value
PA_INTSRC	GPIO_BA+0x020	R/W	PA Interrupt Source Flag	0x0000_XXXX
PB_INTSRC	GPIO_BA+0x060	R/W	PB Interrupt Source Flag	0x0000_XXXX
PC_INTSRC	GPIO_BA+0x0A0	R/W	PC Interrupt Source Flag	0x0000_XXXX
PF_INTSRC	GPIO_BA+0x160	R/W	PF Interrupt Source Flag	0x0000_XXXX

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

Bits	Description
[31:16]	Reserved Reserved.
[n] n=0,1..15	INTSRC[n] Port A-F Pin[n] Interrupt Source Flag Write Operation: 0 = No action. 1 = Clear the corresponding pending interrupt. Read Operation: 0 = No interrupt at Px.n. 1 = Px.n generates an interrupt. Note: The PA.12~15/PB.8~11/PC.6~13/PC.15/PF.6~14 pin is ineffective.

Port A-F Input Schmitt Trigger Enable Register (Px_SMTEN)

Register	Offset	R/W	Description	Reset Value
PA_SMTEN	GPIO_BA+0x024	R/W	PA Input Schmitt Trigger Enable Register	0x0000_0000
PB_SMTEN	GPIO_BA+0x064	R/W	PB Input Schmitt Trigger Enable Register	0x0000_0000
PC_SMTEN	GPIO_BA+0x0A4	R/W	PC Input Schmitt Trigger Enable Register	0x0000_0000
PF_SMTEN	GPIO_BA+0x164	R/W	PF Input Schmitt Trigger Enable Register	0x0000_0000

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

Bits	Description
[31:16]	Reserved Reserved.
[n] n=0,1..15	SMTEN[n] Port A-F Pin[n] Input Schmitt Trigger Enable Bit 0 = Px.n input schmitt trigger function Disabled. 1 = Px.n input schmitt trigger function Enabled. Note: The PA.12~15/PB.8~11/PC.6~13/PC.15/PF.6~14 pin is ineffective.

Port A-F High Slew Rate Control Register (Px_SLEWCTL)

Register	Offset	R/W	Description	Reset Value
PA_SLEWCTL	GPIO_BA+0x028	R/W	PA High Slew Rate Control Register	0x0000_0000
PB_SLEWCTL	GPIO_BA+0x068	R/W	PB High Slew Rate Control Register	0x0000_0000
PC_SLEWCTL	GPIO_BA+0x0A8	R/W	PC High Slew Rate Control Register	0x0000_0000
PF_SLEWCTL	GPIO_BA+0x168	R/W	PF High Slew Rate Control Register	0x0000_0000

31	30	29	28	27	26	25	24
HSREN15		HSREN14		HSREN13		HSREN12	
23	22	21	20	19	18	17	16
HSREN11		HSREN10		HSREN9		HSREN8	
15	14	13	12	11	10	9	8
HSREN07		HSREN6		HSREN5		HSREN4	
7	6	5	4	3	2	1	0
HSREN03		HSREN2		HSREN1		HSREN0	

Bits	Description
[2n+1:2n] n=0,1..15	HSREnn Port A-F Pin[n] High Slew Rate Control 00 = Px.n output with normal slew rate mode. 01 = Px.n output with high slew rate mode. 10 = Reserved. 11 = Reserved. Note 1: The PA.12~15/PB.8~11/PC.6~13/PC.15/PF.6~14 pin is ineffective. Note 2: Please refer to the M253 Datasheet for detailed pin operation voltage information about V_{DD}, V_{DDIO} electrical characteristics.

Port A-F Pull-up and Pull-down Selection Register (Px_PUSEL)

Register	Offset	R/W	Description	Reset Value
PA_PUSEL	GPIO_BA+0x030	R/W	PA Pull-up and Pull-down Selection Register	0x0000_0000
PB_PUSEL	GPIO_BA+0x070	R/W	PB Pull-up and Pull-down Selection Register	0x0000_0000
PC_PUSEL	GPIO_BA+0x0B0	R/W	PC Pull-up and Pull-down Selection Register	0x0000_0000
PF_PUSEL	GPIO_BA+0x170	R/W	PF Pull-up and Pull-down Selection Register	0x0000_0000

31	30	29	28	27	26	25	24
PUSEL15		PUSEL14		PUSEL13		PUSEL12	
23	22	21	20	19	18	17	16
PUSEL11		PUSEL10		PUSEL9		PUSEL8	
15	14	13	12	11	10	9	8
PUSEL7		PUSEL6		PUSEL5		PUSEL4	
7	6	5	4	3	2	1	0
PUSEL3		PUSEL2		PUSEL1		PUSEL0	

Bits	Description
[2n+1:2n] n=0,1..15	Port A-F Pin[n] Pull-up and Pull-down Enable Register Determine each I/O Pull-up/pull-down of Px.n pins. 00 = Px.n pull-up and pull-down disable. 01 = Px.n pull-up enable. 10 = Px.n pull-down enable. 11 = Px.n pull-up and pull-down disable. Note 1: Basically, the pull-up and pull-down control has following behavior limitation. The independent pull-up control register is only valid when MODEn is set as input and open-drain mode even if I/O function is switched to multi-function pin, e.g. UARTx_RXD. The independent pull-down control register is only valid when MODEn is set as tri-state mode. When both pull-up and pull-down are set as 1 at "tri-state" mode, keep I/O in tri-state mode. Note 2: The PA.12~15/PB.8~11/PC.6~13/PC.15/PF.6~14 pin is ineffective.

Interrupt De-bounce Control Register (GPIO_DBCTL)

Register	Offset	R/W	Description	Reset Value
GPIO_DBCTL	GPIO_BA+0x440	R/W	Interrupt De-bounce Control Register	0x0027_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved		ICLKONF	Reserved	Reserved	ICLKONC	ICLKONB	ICLKONA
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved			DBCLKSRC	DBCLKSEL			

Bits	Description
[31:22]	Reserved Reserved.
[21:16] x=A,B..F	ICLKONx Interrupt Clock on Mode 0 = Edge detection circuit is active only if I/O pin corresponding RHIE (Px_INTEN[n+16])/FLIE (Px_INTEN[n]) bit is set to 1. 1 = All I/O pins edge detection circuit is always active after reset. Note: It is recommended to disable this bit to save system power if no special application concern. Each bit controls each GPIO group.
[5]	Reserved Reserved.
[4]	DBCLKSRC De-bounce Counter Clock Source Selection 0 = De-bounce counter clock source is the HCLK. 1 = De-bounce counter clock source is the 38.4 kHz internal low speed RC oscillator (LIRC).
[3:0]	DBCLKSEL De-bounce Sampling Cycle Selection 0000 = Sample interrupt input once per 1 clocks. 0001 = Sample interrupt input once per 2 clocks. 0010 = Sample interrupt input once per 4 clocks. 0011 = Sample interrupt input once per 8 clocks. 0100 = Sample interrupt input once per 16 clocks. 0101 = Sample interrupt input once per 32 clocks. 0110 = Sample interrupt input once per 64 clocks. 0111 = Sample interrupt input once per 128 clocks. 1000 = Sample interrupt input once per 256 clocks. 1001 = Sample interrupt input once per 2*256 clocks. 1010 = Sample interrupt input once per 4*256 clocks. 1011 = Sample interrupt input once per 8*256 clocks. 1100 = Sample interrupt input once per 16*256 clocks. 1101 = Sample interrupt input once per 32*256 clocks. 1110 = Sample interrupt input once per 64*256 clocks. 1111 = Sample interrupt input once per 128*256 clocks.

GPIO Px.n Pin Data Input/Output Register (Pxn PDIO)

Register	Offset	R/W	Description	Reset Value
PAn_PDIO n=0,1,2,3,4,5,6,7,8,9,10,11	GPIO_BA+0x800+(0x04 * n)	R/W	GPIO PA.n Pin Data Input/Output Register	0x0000_000X
PBn_PDIO n=0,1,2,3,4,5,6,7,12,13,14,15	GPIO_BA+0x840+(0x04 * n)	R/W	GPIO PB.n Pin Data Input/Output Register	0x0000_000X
PCn_PDIO n=0,1,2,3,4,5,14	GPIO_BA+0x880+(0x04 * n)	R/W	GPIO PC.n Pin Data Input/Output Register	0x0000_000X
PFn_PDIO n=0,1,2,3,4,5,15	GPIO_BA+0x940+(0x04 * n)	R/W	GPIO PF.n Pin Data Input/Output Register	0x0000_000X

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

Bits	Description
[31:1]	Reserved Reserved.
[0]	PDIO GPIO Px.n Pin Data Input/Output Writing this bit can control one GPIO pin output value. 0 = Corresponding GPIO pin set to low. 1 = Corresponding GPIO pin set to high. Read this register to get GPIO pin status. For example, writing PA0_PDIO will reflect the written value to bit DOUT (Px_DOUT[0]); reading PA0_PDIO will return the value of PIN (PA_PIN[0]). Note 1: The writing operation will not be affected by register DATMSK (Px_DATMSK[n]). Note 2: The PA.12~15/PB.8~11/PC.6~13/PC.15/PF.6~14 pin is ineffective.

6.6 PDMA Controller (PDMA)

6.6.1 Overview

The peripheral direct memory access (PDMA) controller is used to provide high-speed data transfer. The PDMA controller can transfer data from one address to another without CPU intervention. This has the benefit of reducing the workload of CPU and keeps CPU resources free for other applications. The PDMA controller has a total of 5 channels and each channel can perform transfer between memory and peripherals or between memory and memory.

6.6.2 Features

- Supports 5 independently configurable channels
- Supports selectable 2 level of priority (fixed priority or round-robin priority)
- Supports transfer data width of 8, 16, and 32 bits
- Supports source and destination address increment size can be byte, half-word, word or no increment
- Request source can be from software, PSIO, SPI/I²S, UART, USCI, EADC, DAC, PWM capture event and TIMER
- Supports Scatter-gather mode to perform sophisticated transfer through the use of the descriptor link list table
- Supports single and burst transfer type
- Supports time-out function on channel 0 and channel1

6.6.3 Block Diagram

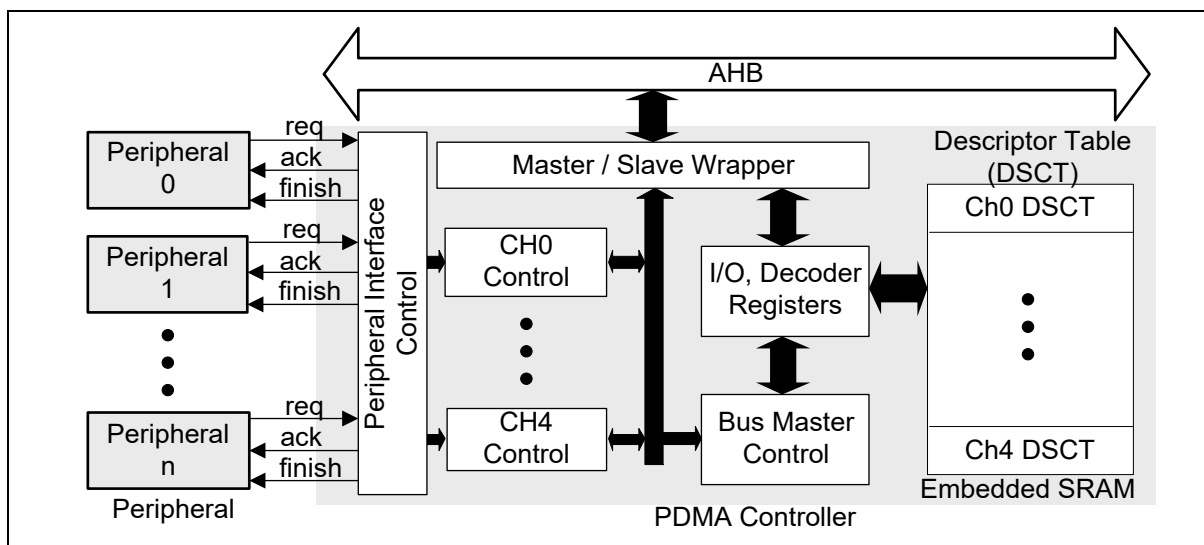


Figure 6.6-1 PDMA Controller Block Diagram

6.6.4 Basic Configuration

- Clock Source Configuration
 - Enable PDMA controller clock in PDMACKEN(CLK_AHBCLK [1]).
- Reset Configuration
 - Reset PDMA controller in PDMARST (SYS_IPRST0[2]).

6.6.5 Functional Description

The PDMA controller transfers data from one address to another without CPU intervention. The PDMA controller supports 5 independent channels and serves only one channel at one time, as the result, PDMA controller supports two level channel priorities: fixed and round-robin priority, PDMA controller serves channel in order from highest to lowest priority channel. The PDMA controller supports two operation modes: Basic mode and Scatter-gather mode. Basic mode is used to perform one descriptor table transfer. Scatter-gather mode has more entries for each PDMA channel, and thus the PDMA controller supports sophisticated transfer through the entries. The descriptor table entry data structure contains many transfer information including the transfer source address, transfer destination address, transfer count, burst size, transfer type and operation mode. Figure 6.6-2 shows the diagram of descriptor table (DSCT) data structure.

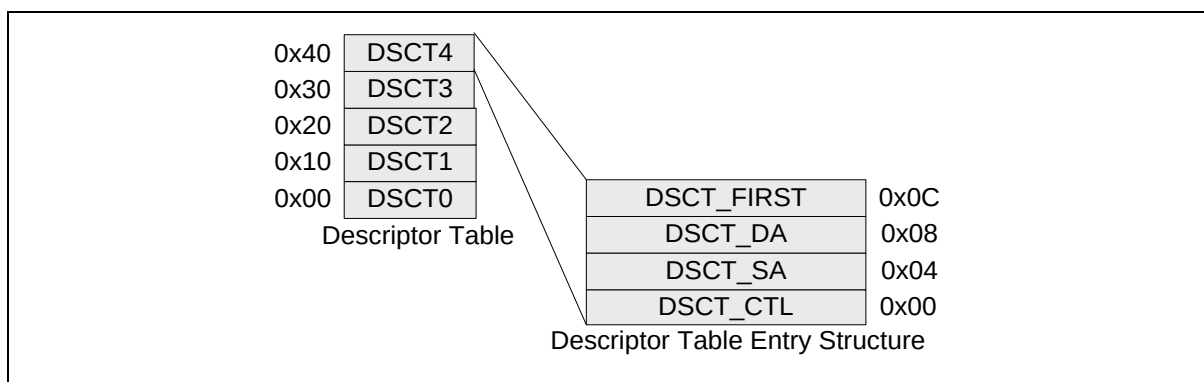


Figure 6.6-2 Descriptor Table Entry Structure

The PDMA controller also supports single and burst transfer type and the request source can be from software or peripheral request, transfer between memory to memory using software request. A single transfer means that software or peripheral is ready to transfer one data (every data needs one request), and the burst transfer means that software or peripherals will transfer multiple data (multiple data only need one request).

6.6.5.1 Channel Priority

The PDMA controller supports two level channel priorities including fixed and round-robin priority. The fixed priority channel has higher priority than round-robin priority channel. If multiple channels are set as fixed or round-robin priority, the higher channel will have higher priority. The priority order is listed in Table 6.6-1.

PDMA_PRISET	Channel Number	Priority Setting	Arbitration Priority In Descending Order
1	4	Channel4, Fixed Priority	Highest
1	3	Channel3, Fixed Priority	---
---	---	---	---
1	0	Channel0, Fixed Priority	---
0	4	Channel4, Round-Robin Priority	---
0	3	Channel3, Round-Robin Priority	---
---	---	---	---
0	0	Channel0, Round-Robin Priority	Lowest

Table 6.6-1 Channel Priority Table

6.6.5.2 PDMA Operation Mode

The PDMA controller supports two operation modes including Basic mode and Scatter-gather mode.

Basic Mode

Basic mode is used to perform one descriptor table transfer mode. This mode can be used to transfer data between memory and memory, peripherals and memory or peripherals and peripherals, but if user want to transfer data between peripherals and peripherals, one thing must be sured is that the request from peripherals knows that the data is ready for transfer or not. PDMA controller operation mode can be set from OPMODE (PDMA_DSCTn_CTL[1:0], n denotes PDMA channel), the default setting is in idle state (OPMODE (PDMA_DSCTn_CTL[1:0]) = 0x0) and recommend user configure the descriptor table in idle state. If operation mode is not in idle state, user re-configure channel setting may make some operation error.

User must fill the transfer count TXCNT (PDMA_DSCTn_CTL[31:16]) register and select transfer width TXWIDTH (PDMA_DSCTn_CTL[13:12]), destination address increment size DAINC (PDMA_DSCTn_CTL[11:10]), source address increment size SAINC (PDMA_DSCTn_CTL[9:8]), burst size BURSIZE (PDMA_DSCTn_CTL[6:4]) and transfer type TXTYPE (PDMA_DSCTn_CTL[2]), then the PDMA controller will perform transfer operation in transfer state after receiving request signal. Finishing this task will generate an interrupt to CPU if corresponding PDMA interrupt bit INTENn (PDMA_INTEN[4:0]) is enabled and the operation mode will be updated to idle state as shown in Figure 6.6-3. If software configures the operation mode to idle state, the PDMA controller will not perform any transfer and then clear this operation request. Finishing this task will also generate an interrupt to CPU if corresponding PDMA interrupt bit is enabled.

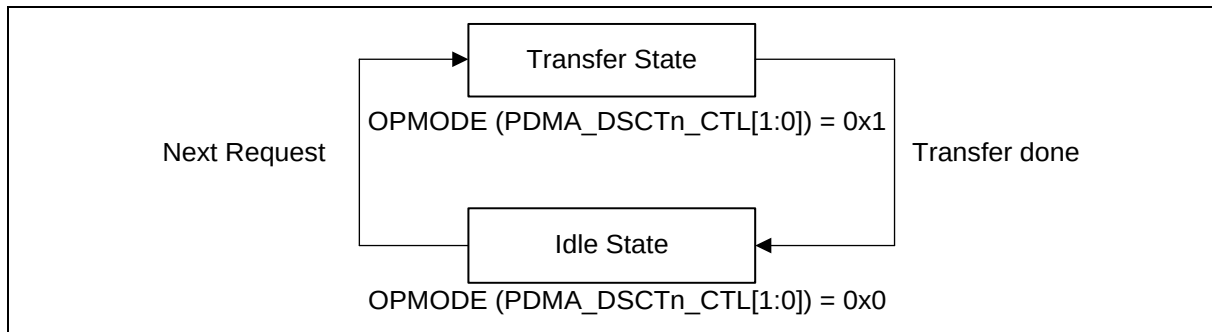


Figure 6.6-3 Basic Mode Finite State Machine

Scatter-gather Mode

Scatter-gather mode is a complex mode and can perform sophisticated transfer through the use of the description link list table as shown in Figure 6.6-4. Through operation mode user can perform peripheral wrapper-around, and multiple PDMA task can be used for data transfer between varied locations in system memory instead of a set of continuous locations. Scatter-gather mode only needs a request to finish all table entries task till the last task with OPMODE (PDMA_DSCTn_CTL[1:0]) is idle state without ack. It also means Scatter-gather mode can only be used to transfer data between memory to memory without handshaking.

In Scatter-Gather mode, user has to set OPMODE (PDMA_DSCTn_CTL[1:0]) to 0x2 to enable Scatter-Gather mode and set PDMA_DSCTn_NEXT to the next descriptor table entry start address. In Scatter-Gather mode, the PDMA controller first loads the next descriptor table entry start address from PDMA_DSCTn_NEXT when PDMA is triggered. Then it processes the tasks in the next descriptor table. After finishing each task, the PDMA controller will generate an interrupt to CPU if the corresponding PDMA interrupt bit is enabled and TBINTDIS (PDMA_DSCTn_CTL[7]) bit is "0" (when finishing task and TBINTDIS bit is "0", the corresponding TDIFn (PDMA_TDSTS[6:0]) flag will be asserted and if this bit is "1" TDIFn will not be active).

If channel 4 has been triggered, and the operation mode is in Scatter-gather mode (OPMODE

(PDMA_DSCTn_CTL[1:0]) = 0x2), the hardware will load the real PDMA information task from the address generated by adding PDMA_DSCTn_NEXT (link address) and PDMA_SCATBA (base address) registers. For example, base address is 0x2000_0000 (only MSB 16 bits valid in PDMA_SCATBA), the current link address is 0x0000_0100 (only LSB 16-bits without last two bits [1:0] valid in PDMA_DSCTn_NEXT), and then the next DSCT entry start address is 0x2000_0100.

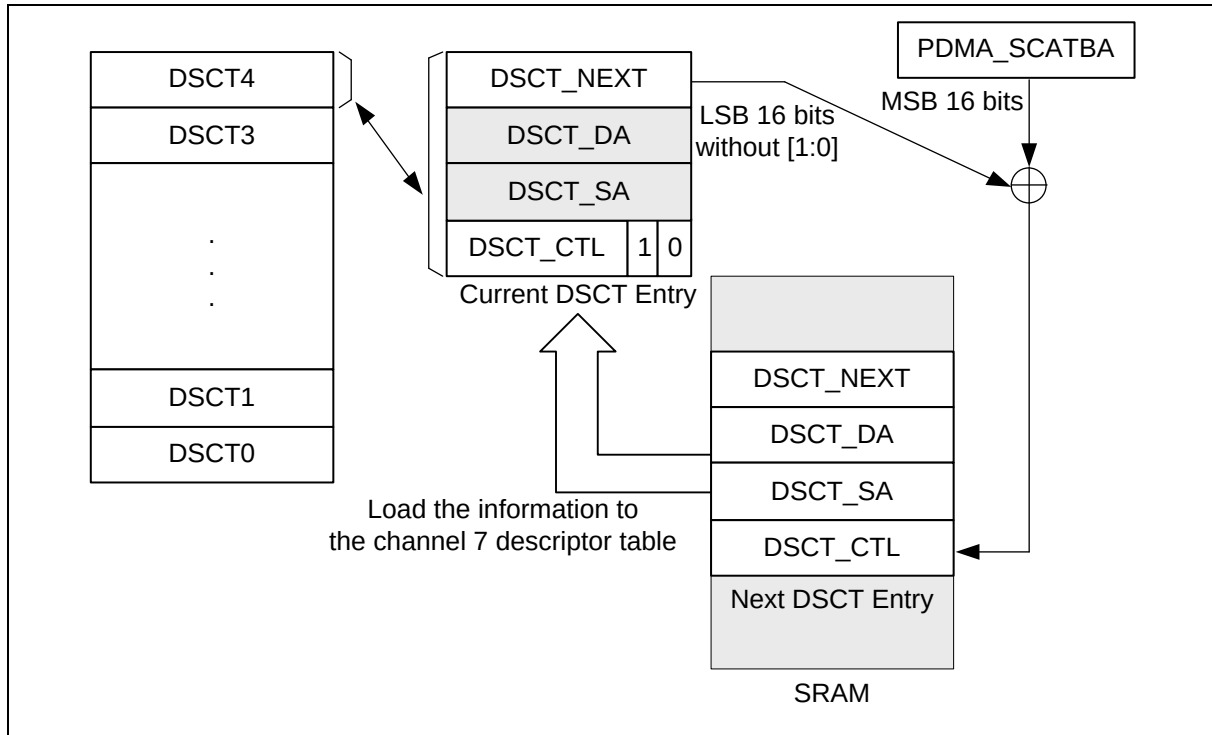


Figure 6.6-4 Descriptor Table Link List Structure

The above link list table operation is DSCT state in Scatter-gather Mode as shown in Figure 6.6-5. When loading the information is finished, it will go to transfer state and start transfer by this information automatically. However, if the next PDMA information is also set to Scatter-gather mode, the hardware will catch the next PDMA information block when the current task is finished. The Scatter-gather mode switches to basic mode when doing the next task. Then, the basic mode switches to Idle state when the last task is finished.

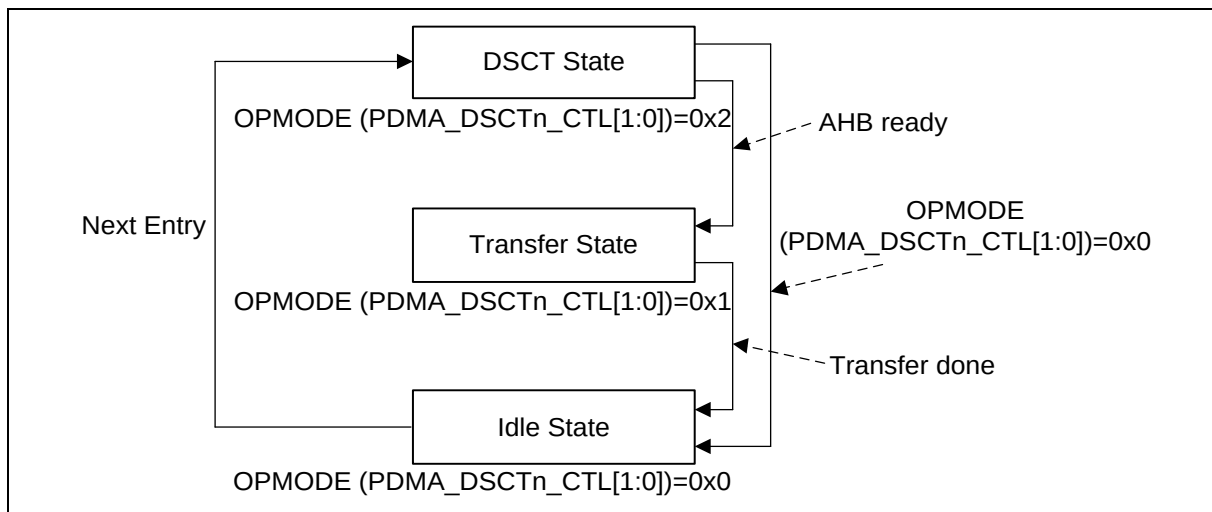


Figure 6.6-5 Scatter-gather Mode Finite State Machine

6.6.5.3 Transfer Type

The PDMA controller supports two transfer types: single transfer type and burst transfer type, configure by setting TXTYPE (PDMA_DSCTn_CTL[2]).

When the PDMA controller is operated in single transfer type, each transfer data needs one request signal for one transfer, after transferred data, TXCNT (PDMA_DSCTn_CTL[31:16]) will decrease 1. Transfer will be finished after the TXCNT (PDMA_DSCTn_CTL[31:16]) decreases to 0. In this mode, the BURSIZ (PDMA_DSCTn_CTL[6:4]) is not useful to control the transfer size. The BURSIZ (PDMA_DSCTn_CTL[6:4]) will be fixed as one.

For the burst transfer type, the PDMA controller transfers TXCNT (PDMA_DSCTn_CTL[31:16]) of data and need only one request signal. After transferred BURSIZ (PDMA_DSCTn_CTL[6:4]) of data, TXCNT (PDMA_DSCTn_CTL[31:16]) will decrease BURSIZ number. Transfer will be done after the transfer count TXCNT (PDMA_DSCTn_CTL[31:16]) decreases to 0. Note that burst transfer type can only be used for PDMA controller to do burst transfer between memory and memory. User must use single request type for memory-to-peripheral and peripheral-to-memory transfers.

Figure 6.6-6 shows an example about single and burst transfer type in basic mode. In this example, channel 1 uses single transfer type and TXCNT (PDMA_DSCTn_CTL[31:16]) = 127. Channel 0 uses burst transfer type, BURSIZ (PDMA_DSCTn_CTL[6:4]) = 128 and TXCNT (PDMA_DSCTn_CTL[31:16]) = 255. The operation sequence is described below:

1. Channel 0 and channel 1 get the trigger signal at the same time.
2. Channel 1 has higher priority than channel 0 by default; the PDMA controller will load the channel 1 descriptor table first and executing. But channel 1 is single transfer type, and thus the PDMA controller will only transfer one transfer data.
3. Then, the PDMA controller turns to the channel 0 and loads channel 0's descriptor table. The channel 0 is burst transfer type and the burst size selected to 128. Therefore, the PDMA controller will transfer 128 transfer data.
4. When channel 0 transfers 128 data, channel 1 gets another request signal, then after channel 0 finishes 128 transfer data, the PDMA controller will turn to channel 1 and transfer next one data.
5. After channel 1 transfers data, the PDMA controller switches to low priority channel 0 to continuous next 128 data transfer. If no channel 1 request receives, PDMA will start next channel 0, 128 data transfer.
6. The PDMA controller will complete transfer when channel 0 finishes data transfer 256 times, and channel 1 finishes transferring 128 times.

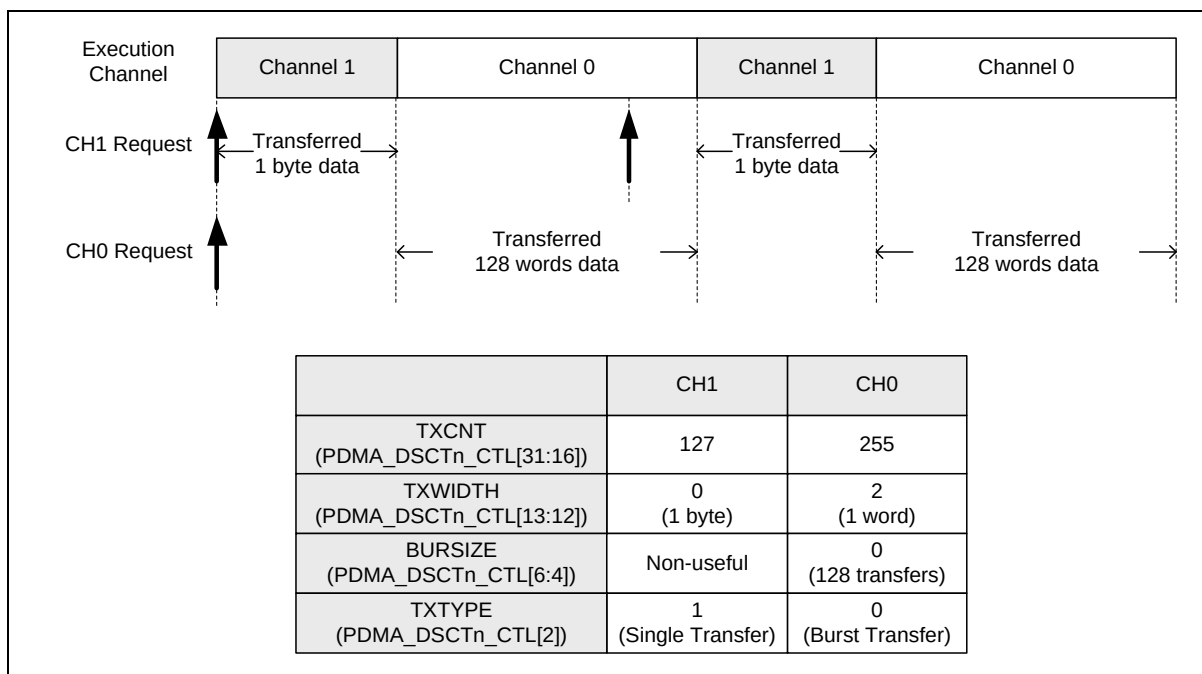


Figure 6.6-6 Example of Single Transfer Type and Burst Transfer Type in Basic Mode

6.6.5.4 Channel Time-out

Only PDMA channel 0 and channel 1 support time-out function. When the transfer channel is enabled and selected to the peripheral, corresponding channel time-out TOUTENn (PDMA_TOUTEN [n], n=0,1) is enabled, then channel's corresponding time-out counter will start count up from 0 while the channel has received trigger signal from the peripheral.

The time-out counter is based on output of HCLK prescaler, which is setting by corresponding channel's TOUTPSCn (PDMA_TOUTPSC [2+4n:4n], n=0,1). If time-out counter counts up from 0 to corresponding channel's TOCn (PDMA_TOC0_1 [16(n+1)-1:16n], n=0,1), the PDMA controller will generate interrupt signal when corresponding TOUTIENn (PDMA_TOUTIEN [n], n=0,1) is enabled. When time-out occurred, corresponding channel's REQTOFn (PDMA_INTSTS [n+8], n=0,1) will be set to indicate channel time-out is happened.

Time-out counter will restart from 0 while counter count to TOCn (PDMA_TOC0_1 [16(n+1)-1:16n], n=0,1), received trigger signal, time-out function is disabled or chip enters Power-down mode. The time-out counter will keep counting until time-out function is disabled.

Figure 6.6-7 shows an example about time-out counter operation. The operation sequence is described below:

1. The channel 0 time-out counter is not counting when time-out function is enabled by setting TOUTEN0(PDMA_TOUTEN[0]) bit to 1.
2. Time-out counter starts counting from 0 to the value of TOC0(PDMA_TOC0_1[15:0]) bits when receiving the first peripheral request.
3. Time-out counter is reset to 0 by received second peripheral request.
4. Channel 0 request time-out flag(REQTOF0(PDMA_INTSTS[8])) is set to high when time-out counter counts to 5. The counter will keep counting.
5. Time-out counter is reset to 0 when time-out function is disabled.

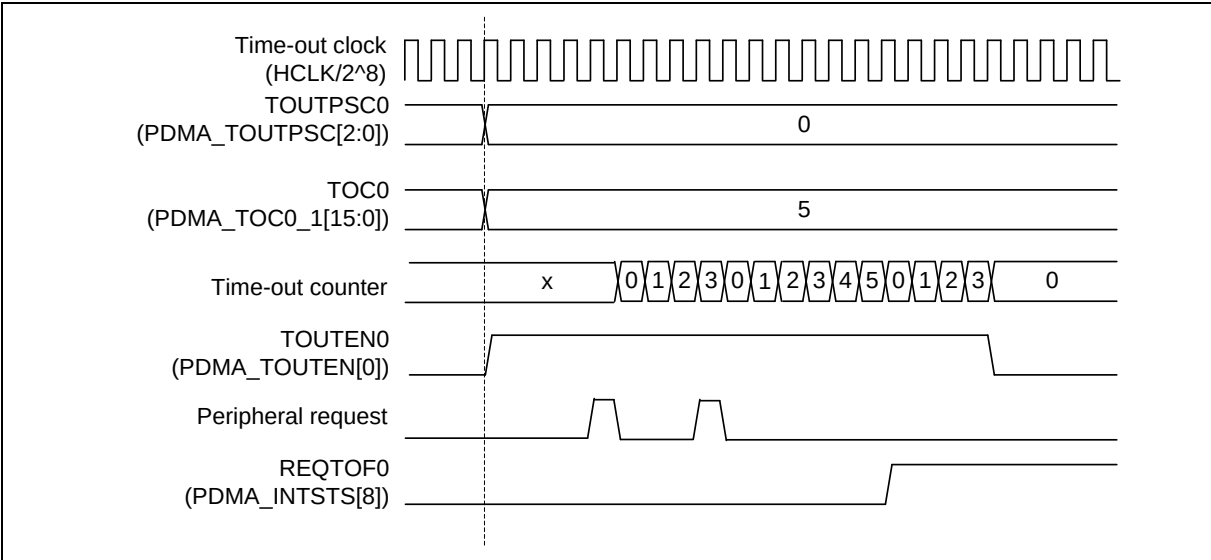


Figure 6.6-7 Example of PDMA Channel 0 Time-out Counter Operation

6.6.6 Register Map

R: read only, W: write only, R/W: both read and write

Register	Offset	R/W	Description	Reset Value
PDMA Base Address: PDMA_BA = 0x4000_8000				
PDMA_DSCTn_CTL n = 0,1..4	PDMA_BA+0x10*n	R/W	Descriptor Table Control Register of PDMA Channel n	0xFFFF_FFFF
PDMA_DSCTn_SA n = 0,1..4	PDMA_BA+0x0004+0x10*n	R/W	Source Address Register of PDMA Channel n	0xFFFF_FFFF
PDMA_DSCTn_DA n = 0,1..4	PDMA_BA+0x0008+0x10*n	R/W	Destination Address Register of PDMA Channel n	0xFFFF_FFFF
PDMA_DSCTn_NEXT n = 0,1..4	PDMA_BA+0x000c+0x10*n	R/W	Next Scatter-gather Descriptor Table Offset Address of PDMA Channel n	0xFFFF_FFFF
PDMA_CURSCATn n = 0,1..4	PDMA_BA+0x0100+0x004*n	R	Current Scatter-gather Descriptor Table Address of PDMA Channel n	0xFFFF_FFFF
PDMA_CHCTL	PDMA_BA+0x400	R/W	PDMA Channel Control Register	0x0000_0000
PDMA_PAUSE	PDMA_BA+0x404	W	PDMA Transfer Pause Control Register	0x0000_0000
PDMA_SWREQ	PDMA_BA+0x408	W	PDMA Software Request Register	0x0000_0000
PDMA_TRGSTS	PDMA_BA+0x40C	R	PDMA Channel Request Status Register	0x0000_0000
PDMA_PRISET	PDMA_BA+0x410	R/W	PDMA Fixed Priority Setting Register	0x0000_0000
PDMA_PRICLR	PDMA_BA+0x414	W	PDMA Fixed Priority Clear Register	0x0000_0000
PDMA_INTEN	PDMA_BA+0x418	R/W	PDMA Interrupt Enable Register	0x0000_0000
PDMA_INTSTS	PDMA_BA+0x41C	R/W	PDMA Interrupt Status Register	0x0000_0000
PDMA_ABTSTS	PDMA_BA+0x420	R/W	PDMA Channel Read/Write Target Abort Flag Register	0x0000_0000
PDMA_TDSTS	PDMA_BA+0x424	R/W	PDMA Channel Transfer Done Flag Register	0x0000_0000
PDMA_ALIGN	PDMA_BA+0x428	R/W	PDMA Transfer Alignment Status Register	0x0000_0000
PDMA_TACTSTS	PDMA_BA+0x42C	R	PDMA Transfer Active Flag Register	0x0000_0000
PDMA_TOUTPSC	PDMA_BA+0x430	R/W	PDMA Time-out Prescaler Register	0x0000_0000
PDMA_TOUTEN	PDMA_BA+0x434	R/W	PDMA Time-out Enable Register	0x0000_0000
PDMA_TOUTIEN	PDMA_BA+0x438	R/W	PDMA Time-out Interrupt Enable Register	0x0000_0000
PDMA_SCATBA	PDMA_BA+0x43C	R/W	PDMA Scatter-gather Descriptor Table Base Address Register	0x2000_0000
PDMA_TOC0_1	PDMA_BA+0x440	R/W	PDMA Time-out Counter Ch1 and Ch0 Register	0xFFFF_FFFF
PDMA_CHRST	PDMA_BA+0x460	R/W	PDMA Channel Reset Register	0x0000_0000
PDMA_REQSEL0_3	PDMA_BA+0x480	R/W	PDMA Request Source Select Register 0	0x0000_0000
PDMA_REQSEL4	PDMA_BA+0x484	R/W	PDMA Request Source Select Register 1	0x0000_0000

6.6.7 Register Description

Descriptor Table Control Register (PDMA_DSCTn_CTL)

Register	Offset	R/W	Description	Reset Value
PDMA_DSCTn_CTL	PDMA_BA+0x10*n	R/W	Descriptor Table Control Register of PDMA Channel n	0xFFFF_XXXX

31	30	29	28	27	26	25	24
TXCNT							
23	22	21	20	19	18	17	16
TXCNT							
15	14	13	12	11	10	9	8
Reserved		TXWIDTH		DAINC		SAINC	
7	6	5	4	3	2	1	0
TBINTDIS	BURSIZE			Reserved	TXTYPE	OPMODE	

Bits	Description	
[31:16]	TXCNT	Transfer Count The TXCNT represents the required number of PDMA transfer, the real transfer count is (TXCNT + 1); The maximum transfer count is 65536, every transfer may be byte, half-word or word that is dependent on TXWIDTH field. Note: When PDMA finishes each transfer data, this field will be decreased immediately.
[15:14]	Reserved	Reserved
[13:12]	TXWIDTH	Transfer Width Selection This field is used for transfer width. 00 = One byte (8 bit) is transferred for every operation. 01 = One half-word (16 bit) is transferred for every operation. 10 = One word (32-bit) is transferred for every operation. 11 = Reserved. Note: The PDMA transfer source address (PDMA_DSCT_SA) and PDMA transfer destination address (PDMA_DSCT_DA) should be alignment under the TXWIDTH selection
[11:10]	DAINC	Destination Address Increment This field is used to set the destination address increment size. 11 = No increment (fixed address). Others = Increment and size is depended on TXWIDTH selection. Note: The fixed address function do not support in memory to memory transfer type.
[9:8]	SAINC	Source Address Increment This field is used to set the source address increment size. 11 = No increment (fixed address). Others = Increment and size is depended on TXWIDTH selection. Note: The fixed address function do not support in memory to memory transfer type.
[7]	TBINTDIS	Table Interrupt Disable Bit This field can be used to decide whether to enable table interrupt or not. If the TBINTDIS

Bits	Description	
		bit is enabled it will not generates TDIFn(PDMA_TDSTS[4:0]) when PDMA controller finishes transfer task. 0 = Table interrupt Enabled. 1 = Table interrupt Disabled. Note: This function is only for Scatter-gather mode.
[6:4]	BURSIZE	Burst Size This field is used for peripheral to determine the burst size or used for determine the re-arbitration size. 000 = 128 Transfers. 001 = 64 Transfers. 010 = 32 Transfers. 011 = 16 Transfers. 100 = 8 Transfers. 101 = 4 Transfers. 110 = 2 Transfers. 111 = 1 Transfers. Note: This field is only useful in burst transfer type.
[3]	Reserved	Reserved.
[2]	TXTYPE	Transfer Type 0 = Burst transfer type. 1 = Single transfer type.
[1:0]	OPMODE	PDMA Operation Mode Selection 00 = Idle state: Channel is stopped or this table is complete. When PDMA finishes channel table task, OPMODE will be cleared to idle state automatically. 01 = Basic mode: The descriptor table only has one task. When this task is finished, the PDMA_INTSTS[1] will be asserted. 10 = Scatter-gather mode: When operating in this mode, user must give the next descriptor table address in PDMA_DSCT_NEXT register; PDMA controller will ignore this task, then load the next task to execute. 11 = Reserved. Note: Before filling new transfer task in the Descriptor Table, user must check the PDMA_INTSTS[1] to make sure the current task is complete.

Start Source Address Register (PDMA_DSCTn_SA)

Register	Offset	R/W	Description	Reset Value
PDMA_DSCTn_SA	PDMA_BA+0x0004+0x10*n	R/W	Source Address Register of PDMA Channel n	0xFFFF_XXXX

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

Bits	Description	
[31:0]	SA	PDMA Transfer Source Address This field indicates a 32-bit source address of PDMA controller.

Destination Address Register (PDMA_DSCTn_DA)

Register	Offset	R/W	Description	Reset Value
PDMA_DSCTn_DA	PDMA_BA+0x0008+0x10*n	R/W	Destination Address Register of PDMA Channel n	0xFFFF_XXXX

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

Bits	Description
[31:0]	DA PDMA Transfer Destination Address This field indicates a 32-bit destination address of PDMA controller.

Next Scatter-gather Descriptor Table Offset Address (PDMA_DSCTn_NEXT)

Register	Offset	R/W	Description	Reset Value
PDMA_DSCTn_NEXT	PDMA_BA+0x000c+0x10*n	R/W	Next Scatter-gather Descriptor Table Offset Address of PDMA Channel n	0xFFFF_XXXX

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

Bits	Description	
[31:16]	EXENEXT	PDMA Execution Next Descriptor Table Offset This field indicates the offset of next descriptor table address of current execution descriptor table in system memory. Note: Write operation is useless in this field.
[15:0]	NEXT	PDMA Next Descriptor Table Offset This field indicates the offset of the next descriptor table address in system memory. Write Operation: If the system memory based address is 0x2000_0000 (PDMA_SCATBA), and the next descriptor table is start from 0x2000_0100, then this field must fill in 0x0100. Read Operation: When operating in Scatter-gather mode, the last two bits NEXT[1:0] will become reserved, and indicate the first next address of system memory. Note 1: The descriptor table address must be word boundary. Note 2: Before filled transfer task in the descriptor table, user must check if the descriptor table is complete.

Current Scatter-gather Descriptor Table Address (PDMA_CURSCATn)

Register	Offset	R/W	Description	Reset Value
PDMA_CURSCATn	PDMA_BA+0x0100+0x004*n	R	Current Scatter-gather Descriptor Table Address of PDMA Channel n	0XXXXX_XXXX

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

Bits	Description
[31:0]	PDMA Current Description Address (Read Only) This field indicates a 32-bit current external description address of PDMA controller. Note: This field is read only and used for Scatter-gather mode only to indicate the current external description address.

Channel Control Register (PDMA_CHCTL)

Register	Offset	R/W	Description	Reset Value
PDMA_CHCTL	PDMA_BA+0x400	R/W	PDMA Channel Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved			CHEN4	CHEN3	CHEN2	CHEN1	CHEN0

Bits	Description	
[31:5]	Reserved	Reserved.
[n] n=0,1..4	CHENn	PDMA Channel Enable Bits Set this bit to 1 to enable PDMA _n operation. Channel cannot be active if it is not set as enabled. 0 = PDMA channel [n] Disabled. 1 = PDMA channel [n] Enabled. Note: Setting the corresponding bit of PDMA_PAUSE or PDMA_CHRST register will also clear this bit.

PDMA Transfer Pause Control Register (PDMA_PAUSE)

Register	Offset	R/W	Description	Reset Value
PDMA_PAUSE	PDMA_BA+0x404	W	PDMA Transfer Pause Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved			PAUSE4	PAUSE3	PAUSE2	PAUSE1	PAUSE0

Bits	Description	
[31:5]	Reserved	Reserved.
[n] n=0,1..4	PAUSEn	PDMA Channel n Transfer Pause Control (Write Only) User can set PAUSEn bit field to pause the PDMA transfer. When user sets PAUSEn bit, the PDMA controller will pause the on-going transfer, then clear the channel enable bit CHEN(PDMA_CHCTL [n], n=0,1.. 4) and clear request active flag (PDMA_TRGSTS[n:0], n=0,1.. 4). If the paused channel is enabled again, the remaining transfers will be processed. 0 = No effect. 1 = Pause PDMA channel n transfer.

PDMA Software Request Register (PDMA_SWREQ)

Register	Offset	R/W	Description	Reset Value
PDMA_SWREQ	PDMA_BA+0x408	W	PDMA Software Request Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved			SWREQ4	SWREQ3	SWREQ2	SWREQ1	SWREQ0

Bits	Description	
[31:5]	Reserved	Reserved.
[n] n=0,1..4	SWREQn	PDMA Software Request (Write Only) Set this bit to 1 to generate a software request to PDMA [n]. 0 = No effect. 1 = Generate a software request. Note 1: User can read PDMA_TRGSTS register to know which channel is on active. Active flag may be triggered by software request or peripheral request. Note 2: If user does not enable corresponding PDMA channel, the software request will be ignored.

PDMA Channel Request Status Register (PDMA_TRGSTS)

Register	Offset	R/W	Description	Reset Value
PDMA_TRGSTS	PDMA_BA+0x40C	R	PDMA Channel Request Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved			REQSTS4	REQSTS3	REQSTS2	REQSTS1	REQSTS0

Bits	Description	
[31:5]	Reserved	Reserved.
[n] n=0,1..4	REQSTS _n	PDMA Channel Request Status (Read Only) This flag indicates whether channel[n] have a request or not, no matter request from software or peripheral. When PDMA controller finishes channel transfer, this bit will be cleared automatically. 0 = PDMA Channel n has no request. 1 = PDMA Channel n has a request. Note: If user pauses or resets each PDMA transfer by setting PDMA_PAUSE or PDMA_CHRST register respectively, this bit will be cleared automatically after finishing the current transfer.

PDMA Fixed Priority Setting Register (PDMA_PRISET)

Register	Offset	R/W	Description	Reset Value
PDMA_PRISET	PDMA_BA+0x410	R/W	PDMA Fixed Priority Setting Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved			FPRISET4	FPRISET3	FPRISET2	FPRISET1	FPRISET0

Bits	Description	
[31:5]	Reserved	Reserved.
[n] n=0,1..4	FPRISETn	PDMA Fixed Priority Setting Set this bit to 1 to enable fixed priority level. Write Operation: 0 = No effect. 1 = Set PDMA channel [n] to fixed priority channel. Read Operation: 0 = Corresponding PDMA channel is round-robin priority. 1 = Corresponding PDMA channel is fixed priority. Note: This field only set to fixed priority, clear fixed priority use PDMA_PRICLR register.

PDMA Fix Priority Clear Register (PDMA_PRICLR)

Register	Offset	R/W	Description	Reset Value
PDMA_PRICLR	PDMA_BA+0x414	W	PDMA Fixed Priority Clear Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved			FPRICLR4	FPRICLR3	FPRICLR2	FPRICLR1	FPRICLR0

Bits	Description	
[31:5]	Reserved	Reserved.
[n] n=0,1..4	FPRICLRn	PDMA Fixed Priority Clear Bits (Write Only) Set this bit to 1 to clear fixed priority level. 0 = No effect. 1 = Clear PDMA channel [n] fixed priority setting. Note: User can read PDMA_PRISET register to know the channel priority.

PDMA Interrupt Enable Register (PDMA_INTEN)

Register	Offset	R/W	Description	Reset Value
PDMA_INTEN	PDMA_BA+0x418	R/W	PDMA Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved			INTEN4	INTEN3	INTEN2	INTEN1	INTEN0

Bits	Description	
[31:5]	Reserved	Reserved.
[n] n=0,1..4	INTENn	PDMA Interrupt Enable Bits This field is used to enable PDMA channel[n] interrupt. 0 = PDMA channel n interrupt Disabled. 1 = PDMA channel n interrupt Enabled. Note: The interrupt flag is time-out, abort, transfer done and align.

PDMA Interrupt Status Register (PDMA_INTSTS)

Register	Offset	R/W	Description	Reset Value
PDMA_INTSTS	PDMA_BA+0x41C	R/W	PDMA Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved						REQTOF1	REQTOF0
7	6	5	4	3	2	1	0
Reserved					ALIGNF	TDIF	ABTIF

Bits	Description
[31:10]	Reserved Reserved.
[9]	REQTOF1 Request Time-out Flag for Channel 1 This flag indicates that PDMA controller has waited peripheral request for a period defined by PDMA_TOC1, user can write 1 to clear this bit. 0 = No request time-out. 1 = Peripheral request time-out. Note: Please disable time-out function before clear this bit.
[8]	REQTOF0 Request Time-out Flag for Channel 0 This flag indicates that PDMA controller has waited peripheral request for a period defined by PDMA_TOC0, user can write 1 to clear this bit. 0 = No request time-out. 1 = Peripheral request time-out. Note: Please disable time-out function before clear this bit.
[7:3]	Reserved Reserved.
[2]	ALIGNF Transfer Alignment Interrupt Flag (Read Only) 0 = PDMA channel source address and destination address both follow transfer width setting. 1 = PDMA channel source address or destination address is not follow transfer width setting.
[1]	TDIF Transfer Done Interrupt Flag (Read Only) This bit indicates that PDMA controller has finished transmission; User can read PDMA_TDSTS register to indicate which channel finished transfer. 0 = Not finished yet. 1 = PDMA channel has finished transmission.
[0]	ABTIF PDMA Read/Write Target Abort Interrupt Flag (Read Only) This bit indicates that PDMA has target abort error; Software can read PDMA_ABTSTS register to find which channel has target abort error. 0 = No AHB bus ERROR response received.

Bits	Description	
		1 = AHB bus ERROR response received.

PDMA Channel Read/Write Target Abort Flag Register (PDMA_ABTSST)

Register	Offset	R/W	Description	Reset Value
PDMA_ABTSST	PDMA_BA+0x420	R/W	PDMA Channel Read/Write Target Abort Flag Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved			ABTIF4	ABTIF3	ABTIF2	ABTIF1	ABTIF0

Bits	Description	
[31:5]	Reserved	Reserved.
[n] n=0,1..4	ABTIFn	PDMA Read/Write Target Abort Interrupt Status Flag This bit indicates which PDMA controller has target abort error; User can write 1 to clear these bits. 0 = No AHB bus ERROR response received when channel n transfer. 1 = AHB bus ERROR response received when channel n transfer.

PDMA Channel Transfer Done Flag Register (PDMA_TDSTS)

Register	Offset	R/W	Description	Reset Value
PDMA_TDSTS	PDMA_BA+0x424	R/W	PDMA Channel Transfer Done Flag Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved			TDIF4	TDIF3	TDIF2	TDIF1	TDIF0

Bits	Description	
[31:5]	Reserved	Reserved.
[n] n=0,1..4	TDIFn	Transfer Done Flag This bit indicates whether PDMA controller channel transfer has been finished or not, user can write 1 to clear these bits. 0 = PDMA channel transfer has not finished. 1 = PDMA channel has finished transmission.

PDMA Transfer Alignment Status Register (PDMA_ALIGN)

Register	Offset	R/W	Description	Reset Value
PDMA_ALIGN	PDMA_BA+0x428	R/W	PDMA Transfer Alignment Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved			ALIGN4	ALIGN3	ALIGN2	ALIGN1	ALIGN0

Bits	Description	
[31:5]	Reserved	Reserved.
[n] n=0,1..4	ALIGNn	Transfer Alignment Flag 0 = PDMA channel source address and destination address both follow transfer width setting. 1 = PDMA channel source address or destination address is not follow transfer width setting.

PDMA Transfer Active Flag Register (PDMA_TACTSTS)

Register	Offset	R/W	Description	Reset Value
PDMA_TACTSTS	PDMA_BA+0x42C	R	PDMA Transfer Active Flag Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved			TXACTF4	TXACTF3	TXACTF2	TXACTF1	TXACTF0

Bits	Description	
[31:5]	Reserved	Reserved.
[n] n=0,1..4	TXACTFn	Transfer on Active Flag (Read Only) This bit indicates which PDMA channel is in active. 0 = PDMA channel is not finished. 1 = PDMA channel is active.

PDMA Time-out Prescaler Register (PDMA TOUTPSC)

Register	Offset	R/W	Description	Reset Value
PDMA_TOUTPSC	PDMA_BA+0x430	R/W	PDMA Time-out Prescaler Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved	TOUTPSC1			Reserved	TOUTPSC0		

Bits	Description	
[31:7]	Reserved	Reserved.
[6:4]	TOUTPSC1	PDMA Channel 1 Time-out Clock Source Prescaler Bits 000 = PDMA channel 1 time-out clock source is HCLK/2 ⁸ . 001 = PDMA channel 1 time-out clock source is HCLK/2 ⁹ . 010 = PDMA channel 1 time-out clock source is HCLK/2 ¹⁰ . 011 = PDMA channel 1 time-out clock source is HCLK/2 ¹¹ . 100 = PDMA channel 1 time-out clock source is HCLK/2 ¹² . 101 = PDMA channel 1 time-out clock source is HCLK/2 ¹³ . 110 = PDMA channel 1 time-out clock source is HCLK/2 ¹⁴ . 111 = PDMA channel 1 time-out clock source is HCLK/2 ¹⁵ .
[3]	Reserved	Reserved.
[2:0]	TOUTPSC0	PDMA Channel 0 Time-out Clock Source Prescaler Bits 000 = PDMA channel 0 time-out clock source is HCLK/2 ⁸ . 001 = PDMA channel 0 time-out clock source is HCLK/2 ⁹ . 010 = PDMA channel 0 time-out clock source is HCLK/2 ¹⁰ . 011 = PDMA channel 0 time-out clock source is HCLK/2 ¹¹ . 100 = PDMA channel 0 time-out clock source is HCLK/2 ¹² . 101 = PDMA channel 0 time-out clock source is HCLK/2 ¹³ . 110 = PDMA channel 0 time-out clock source is HCLK/2 ¹⁴ . 111 = PDMA channel 0 time-out clock source is HCLK/2 ¹⁵ .

PDMA Time-out Enable Register (PDMA TOUTEN)

Register	Offset	R/W	Description	Reset Value
PDMA_TOUTEN	PDMA_BA+0x434	R/W	PDMA Time-out Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved						TOUTEN1	TOUTEN0

Bits	Description	
[31:2]	Reserved	Reserved.
[n] n=0,1	TOUTENn	PDMA Time-out Enable Bits 0 = PDMA Channel n time-out function Disabled. 1 = PDMA Channel n time-out function Enabled.

PDMA Time-out Interrupt Enable Register (PDMA_TOUTIEN)

Register	Offset	R/W	Description	Reset Value
PDMA_TOUTIEN	PDMA_BA+0x438	R/W	PDMA Time-out Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved						TOUTIEN1	TOUTIEN0

Bits	Description	
[31:2]	Reserved	Reserved.
[n] n=0,1	TOUTIENn	PDMA Time-out Interrupt Enable Bits 0 = PDMA Channel n time-out interrupt Disabled. 1 = PDMA Channel n time-out interrupt Enabled.

PDMA Scatter-gather Descriptor Table Base Address Register (PDMA_SCATBA)

Register	Offset	R/W	Description	Reset Value
PDMA_SCATBA	PDMA_BA+0x43C	R/W	PDMA Scatter-gather Descriptor Table Base Address Register	0x2000_0000

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

Bits	Description
[31:16]	PDMA Scatter-gather Descriptor Table Address In Scatter-gather mode, this is the base address for calculating the next link - list address. The next link address equation is Next Link Address = PDMA_SCATBA + PDMA_DSCT_NEXT. Note: Only useful in Scatter-gather mode.
[15:0]	Reserved.

PDMA Time-out Period Counter Register 0 (PDMA_TOC0_1)

Register	Offset	R/W	Description	Reset Value
PDMA_TOC0_1	PDMA_BA+0x440	R/W	PDMA Time-out Counter Ch1 and Ch0 Register	0xFFFF_FFFF

31	30	29	28	27	26	25	24
TOC1							
23	22	21	20	19	18	17	16
TOC1							
15	14	13	12	11	10	9	8
TOC0							
7	6	5	4	3	2	1	0
TOC0							

Bits	Description	
[31:16]	TOC1	Time-out Counter for Channel 1 This controls the period of time-out function for channel 1. The calculation unit is based on TOUTPSC1 (PDMA_TOUTPSC0[6:4]) clock. For the example of time-out period, refer to TOC0 bit description.
[15:0]	TOC0	Time-out Counter for Channel 0 This controls the period of time-out function for channel 0. The calculation unit is based on TOUTPSC0 (PDMA_TOUTPSC0[2:0]) clock. Time-out period = (Period of time-out clock) * (16-bit TOCn), n = 0,1.

PDMA Channel Reset Register (PDMA_CHRST)

Register	Offset	R/W	Description	Reset Value
PDMA_CHRST	PDMA_BA+0x460	R/W	PDMA Channel Reset Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved			CH4RST	CH3RST	CH2RST	CH1RST	CH0RST

Bits	Description	
[31:5]	Reserved	Reserved.
[4:0]	CHnRST	Channel n Reset 0 = corresponding channel n is not reset. 1 = corresponding channel n is reset.

PDMA Request Source Select Register 0 (PDMA_REQSEL0_3)

Register	Offset	R/W	Description	Reset Value
PDMA_REQSEL0_3	PDMA_BA+0x480	R/W	PDMA Request Source Select Register 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	REQSRC3						
23	22	21	20	19	18	17	16
Reserved	REQSRC2						
15	14	13	12	11	10	9	8
Reserved	REQSRC1						
7	6	5	4	3	2	1	0
Reserved	REQSRC0						

Bits	Description	
[31]	Reserved	Reserved.
[30:24]	REQSRC3	Channel 3 Request Source Selection This field defines which peripheral is connected to PDMA channel 3. User can configure the peripheral setting by REQSRC3. Note: The channel configuration is the same as REQSRC0 field. Please refer to the explanation of REQSRC0.
[23]	Reserved	Reserved.
[22:16]	REQSRC2	Channel 2 Request Source Selection This field defines which peripheral is connected to PDMA channel 2. User can configure the peripheral setting by REQSRC2. Note: The channel configuration is the same as REQSRC0 field. Please refer to the explanation of REQSRC0.
[15]	Reserved	Reserved.
[14:8]	REQSRC1	Channel 1 Request Source Selection This field defines which peripheral is connected to PDMA channel 1. User can configure the peripheral setting by REQSRC1. Note: The channel configuration is the same as REQSRC0 field. Please refer to the explanation of REQSRC0.
[7]	Reserved	Reserved.
[6:0]	REQSRC0	Channel 0 Request Source Selection This field defines which peripheral is connected to PDMA channel 0. User can configure the peripheral by setting REQSRC0. 0 = Disable PDMA peripheral request. 1 = Reserved. 2 = Reserved. 3 = Reserved. 4 = Channel connects to UART0_TX. 5 = Channel connects to UART0_RX.

Bits	Description
	6 = Channel connects to UART1_TX. 7 = Channel connects to UART1_RX. 8 = Channel connects to UART2_TX. 9 = Channel connects to UART2_RX. 10 = Channel connects to UART3_TX. 11 = Channel connects to UART3_RX. 12 = Channel connects to UART4_TX. 13 = Channel connects to UART4_RX. 14 = Reserved. 15 = Reserved. 16 = Channel connects to USCIO_TX. 17 = Channel connects to USCIO_RX. 18 = Reserved. 19 = Reserved. 20 = Reserved. 21 = Reserved. 22 = Channel connects to SPI0_TX. 23 = Channel connects to SPI0_RX. 24 = Reserved. 25 = Reserved. 26 = Reserved. 27 = Reserved. 28 = Reserved. 29 = Reserved. 30 = Reserved. 31 = Reserved. 32 = Reserved. 33 = Reserved. 34 = Reserved. 35 = Reserved. 36 = Reserved. 37 = Reserved. 38 = Channel connects to I2C0_TX. 39 = Channel connects to I2C0_RX. 40 = Channel connects to I2C1_TX. 41 = Channel connects to I2C1_RX. 42 = Reserved. 43 = Reserved. 44 = Reserved. 45 = Reserved. 46 = Channel connects to TMR0. 47 = Channel connects to TMR1. 48 = Channel connects to TMR2. 49 = Channel connects to TMR3. 50 = Channel connects to ADC_RX. 52 = Reserved. 53 = Reserved. 54 = Reserved.

Bits	Description
	55 = Reserved. 56 = Reserved. 57 = Reserved. 58 = Reserved. 59 = Reserved. 60 = Reserved. 61 = Reserved. 62 = Reserved. 63 = Reserved. 64 = Reserved. 65 = Reserved. Others = Reserved. Note 1: A peripheral cannot be assigned to two channels at the same time. Note 2: This field is useless when transfer between memory and memory.

PDMA Request Source Select Register 1 (PDMA_REQSEL4)

Register	Offset	R/W	Description	Reset Value
PDMA_REQSEL4	PDMA_BA+0x484	R/W	PDMA Request Source Select Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved	REQSRC4						

Bits	Description	
[31:7]	Reserved	Reserved.
[6:0]	REQSRC4	Channel 4 Request Source Selection This field defines which peripheral is connected to PDMA channel 4. User can configure the peripheral setting by REQSRC4. Note: The channel configuration is the same as REQSRC0 field. Please refer to the explanation of REQSRC0.

6.7 Timer Controller (TMR)

6.7.1 Overview

The timer controller includes four 32-bit timers, Timer0 ~ Timer3, allowing user to easily implement a timer control for applications. The timer can perform functions, such as frequency measurement, delay timing, clock generation, and event counting by external input pins, and interval measurement by external capture pins.

6.7.2 Features

6.7.2.1 Timer Function Features

- Four sets of 32-bit timers, each timer having one 24-bit up counter and one 8-bit prescale counter
- Independent clock source for each timer
- Provides one-shot, periodic, toggle-output and continuous counting operation modes
- 24-bit up counter value is readable through CNT (TIMERx_CNT[23:0])
- Supports event counting function
- 24-bit capture value is readable through CAPDAT (TIMERx_CAP[23:0])
- Supports external capture pin (TMx_EXT) event for interval measurement
- Supports external capture pin (TMx_EXT) event to reset 24-bit up counter
- Supports internal clock (HIRC, LIRC) and external clock (HXT, LXT) for capture event
- Supports chip wake-up from Idle/Power-down mode if a timer interrupt signal is generated
- Support Timer0 ~ Timer3 time-out interrupt signal or capture interrupt signal to trigger BPWM, EADC, DAC and PDMA function
- Supports Inter-Timer trigger mode

6.7.4 Block Diagram

The timer controller block diagram and clock control are shown as follows.

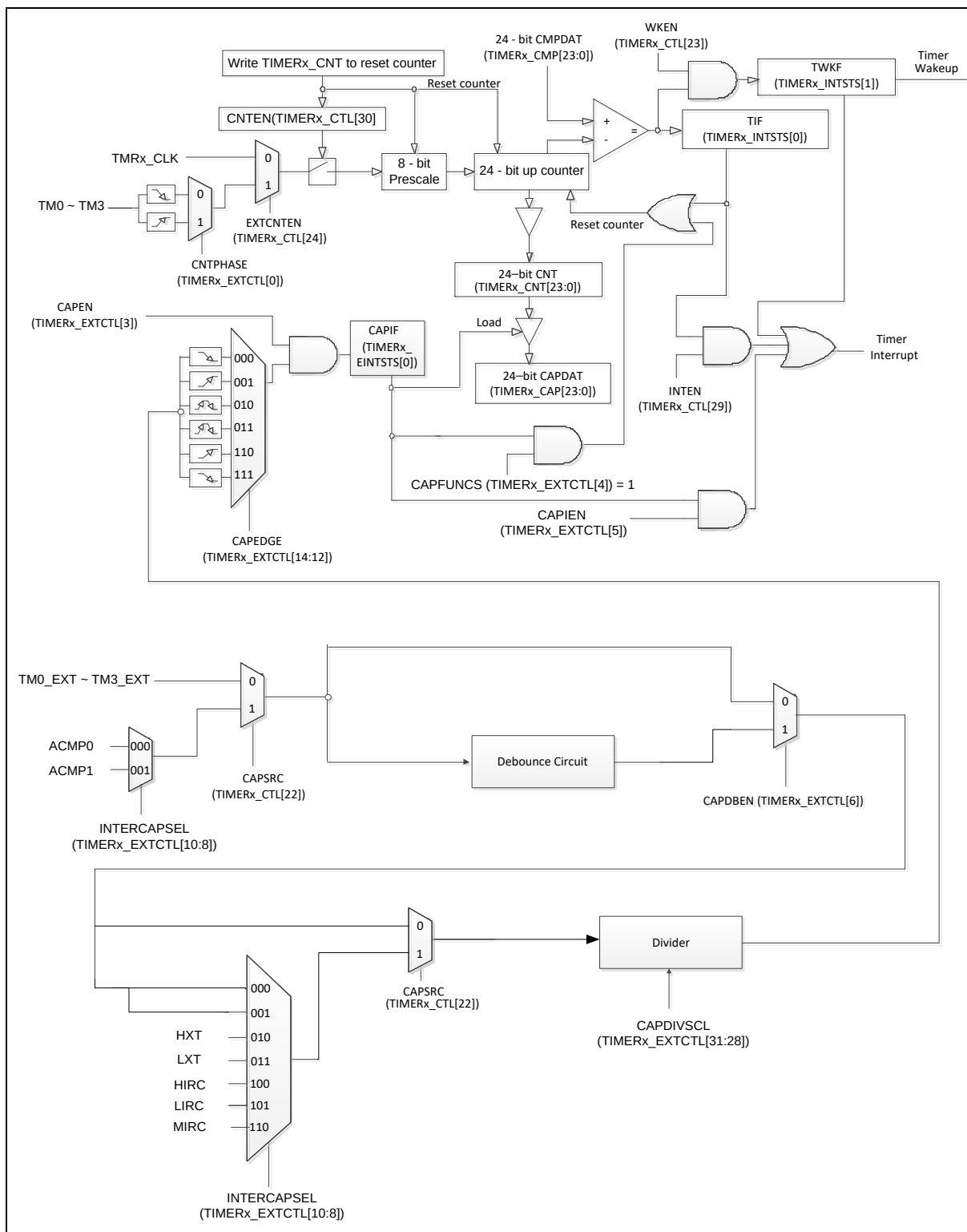


Figure 6.7-1 Timer Controller Block Diagram

The clock source of Timer0 ~ Timer3 in timer mode can be enabled in TMRxCKEN

(CLK_APBCLK0[5:2]) and selected as different frequency in TMR0SEL (CLK_CLKSEL1[10:8]) for Timer0, TMR1SEL (CLK_CLKSEL1[14:12]) for Timer1, TMR2SEL (CLK_CLKSEL1[18:16]) for Timer2 and TMR3SEL (CLK_CLKSEL1[22:20]) for Timer3 as Figure 6.7-2.

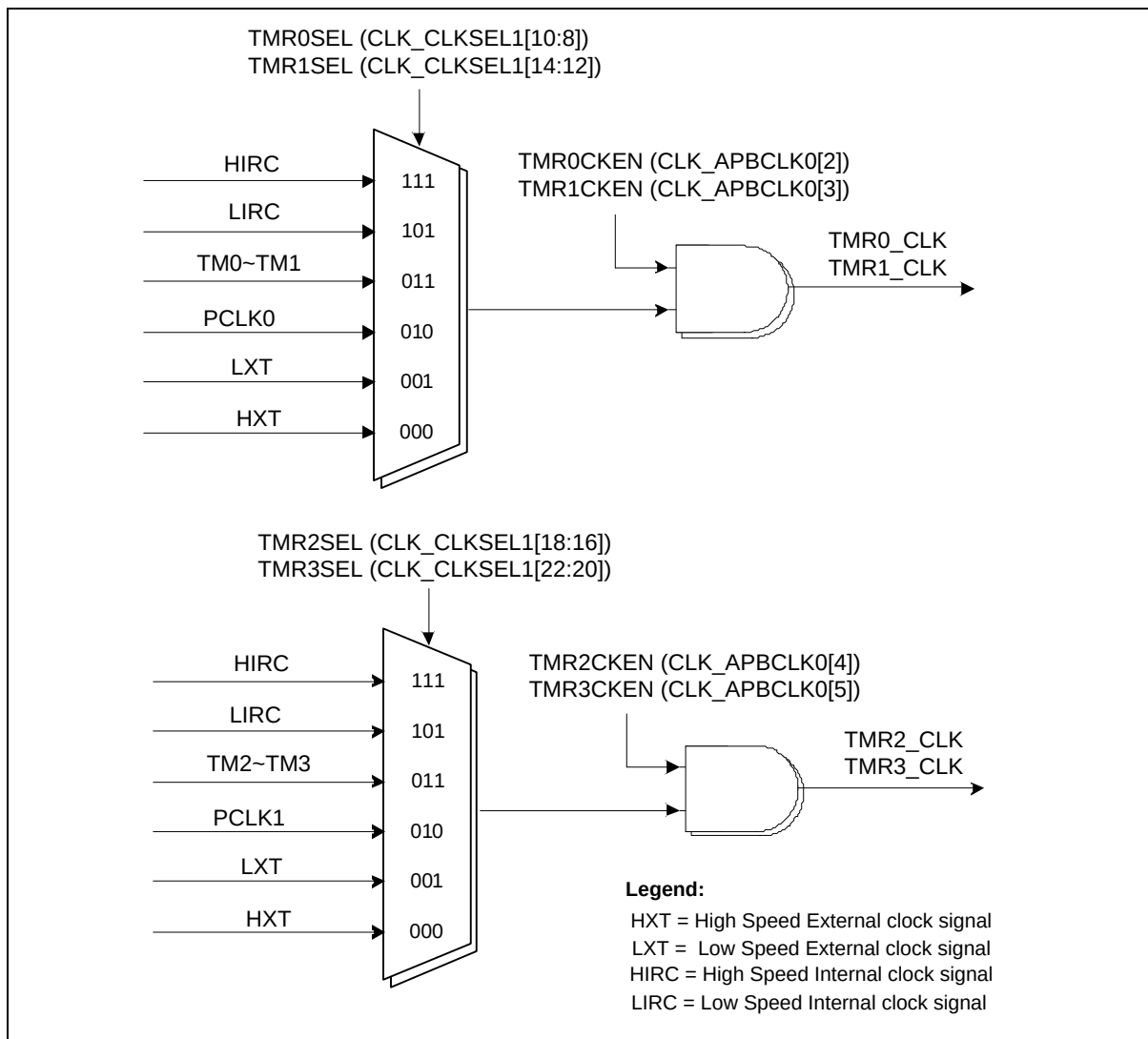


Figure 6.7-2 Clock Source of Timer Controller

6.7.5 Basic Configuration

The clock source of Timer0 ~ Timer3 in timer mode can be enabled in TMRxCKEN (CLK_APBCLK0[5:2]) and selected as different frequency in TMR0SEL (CLK_CLKSEL1[10:8]) for Timer0, TMR1SEL (CLK_CLKSEL1[14:12]) for Timer1, TMR2SEL (CLK_CLKSEL1[18:16]) for Timer2 and TMR3SEL (CLK_CLKSEL1[22:20]) for Timer3.

6.7.5.1 TIMER01 Basic Configurations

- Clock source configuration
 - Enable TIMER0 peripheral clock in TMR0CKEN (CLK_APBCLK0[2]).
 - Enable TIMER1 peripheral clock in TMR1CKEN (CLK_APBCLK0[3]).
- Reset configuration
 - Reset TIMER0 controller in TMR0RST (SYS_IPRST1[2]).
 - Reset TIMER1 controller in TMR1RST (SYS_IPRST1[3]).
-

6.7.5.2 TIMER23 Basic Configurations

- Clock source configuration
 - Enable TIMER2 peripheral clock in TMR2CKEN (CLK_APBCLK0[4]).
 - Enable TIMER3 peripheral clock in TMR3CKEN (CLK_APBCLK0[5]).
- Reset configuration
 - Reset TIMER2 controller in TMR2RST (SYS_IPRST1[4]).
 - Reset TIMER3 controller in TMR3RST (SYS_IPRST1[5]).
-

6.7.6 Timer Functional Description

6.7.6.1 Timer Interrupt Flag

The timer controller supports the following interrupt flags: TIF (TIMERx_INTSTS[0]) which is set while the timer counter value CNT (TIMERx_CNT[23:0]) matches the timer compared value CMPDAT (TIMERx_CMP[23:0]), and CAPIF (TIMERx_EINTSTS[0]) which is set when the transition on the TMx_EXT pin, ACMP, internal clock (HIRC, LIRC) or external clock (HXT, LXT) is associated with CAPEDGE (TIMERx_EXTCTL[14:12]) setting. The TWKF (TIMERx_INTSTS[1]) bit indicates the interrupt wake-up flag status of timer. User can set CAPSRC (TIMERx_CTL[22]) and INTERCAPSEL (TIMERx_EXTCTL[10:8]) to select capture source. Setting WKEN (TIMERx_CTL[23]) to 1 can use wake-up function.

6.7.6.2 Timer Counting Mode

The timer controller provides four timer counting modes: one-shot, periodic, toggle-output and continuous counting operation modes:

One-shot Mode

If the timer controller is configured at one-shot mode (TIMERx_CTL[28:27] is 00) and CNTEN (TIMERx_CTL[30]) is set, the timer counter starts up counting. Once the CNT (TIMERx_CNT[23:0]) value reaches CMPDAT (TIMERx_CMP[23:0]) value, the TIF (TIMERx_INTSTS[0]) will be set to 1, CNT value and CNTEN bit is cleared automatically by timer controller then timer counting operation stops. In the meantime, if the INTEN (TIMERx_CTL[29]) is enabled, the timer interrupt signal is generated and sent to NVIC to inform CPU also.

User can monitor the RSTACT (TIMERx_CNT[31]) to ensure counter reset operation active and disable ICE debug mode acknowledgement effects TIMER counting by setting ICEDEBUG (TIMERx_CTL[31]) to 1.

Periodic Mode

If the timer controller is configured at periodic mode (TIMERx_CTL[28:27] is 01) and CNTEN (TIMERx_CTL[30]) is set, the timer counter starts up counting. Once the CNT (TIMERx_CNT[23:0]) value reaches CMPDAT (TIMERx_CMP[23:0]) value, the TIF (TIMERx_INTSTS[0]) will be set to 1, CNT value will be cleared automatically by timer controller and timer counter operates counting again. In the meantime, if the INTEN (TIMERx_CTL[29]) bit is enabled, the timer interrupt signal is generated and sent to NVIC to inform CPU also. In this mode, the timer controller operates counting and compares with CMPDAT value periodically until the CNTEN bit is cleared by user.

User can set PERIOSEL (TIMERx_CTL[20]) to select Timer behavior.

Toggle-Output Mode

If the timer controller is configured at toggle-output mode (TIMERx_CTL[28:27] is 10) and CNTEN (TIMERx_CTL[30]) is set, the timer counter starts up counting. The counting operation of toggle-output mode is almost the same as periodic mode, except toggle-output mode has associated TM0 ~ TM3 or TM0_EXT ~ TM3_EXT pin to output signal while specify TIF (TIMERx_INTSTS[0]) is set. User can set TGLPINSEL (TIMERx_CTL[21]) to choose TMx or TMx_EXT as toggle-output pin. Thus, the toggle-output signal on TM0 ~ TM3 pin is high and changing back and forth with 50% duty cycle.

Continuous Counting Mode

If the timer controller is configured at continuous counting mode (TIMERx_CTL[28:27] is 11) and CNTEN (TIMERx_CTL[30]) is set, the timer counter starts up counting. Once the CNT (TIMERx_CNT[23:0]) value reaches CMPDAT (TIMERx_CMP[23:0]) value, the TIF (TIMERx_INTSTS[0]) will be set to 1 and CNT value keeps up counting. In the meantime, if the INTEN (TIMERx_CTL[29]) is enabled, the timer interrupt signal is generated and sent to NVIC to inform CPU also. User can change different CMPDAT value immediately without disabling timer counting and restarting timer counting in this mode.

For example, CMPDAT value is set as 80, first. The TIF will set to 1 when CNT value is equal to 80, timer counter is kept counting and CNT value will not goes back to 0, it continues to count 81, 82, 83, ... to ($2^{24} - 1$), 0, 1, 2, 3, ... to ($2^{24} - 1$) again and again. Next, if user programs CMPDAT value as 200 and clears TIF, the TIF will set to 1 again when CNT value reaches to 200. At last, user programs CMPDAT as 500 and clears TIF, the TIF will set to 1 again when CNT value reaches to 500.

In this mode, the timer counting is continuous. So this operation mode is called as continuous counting mode.

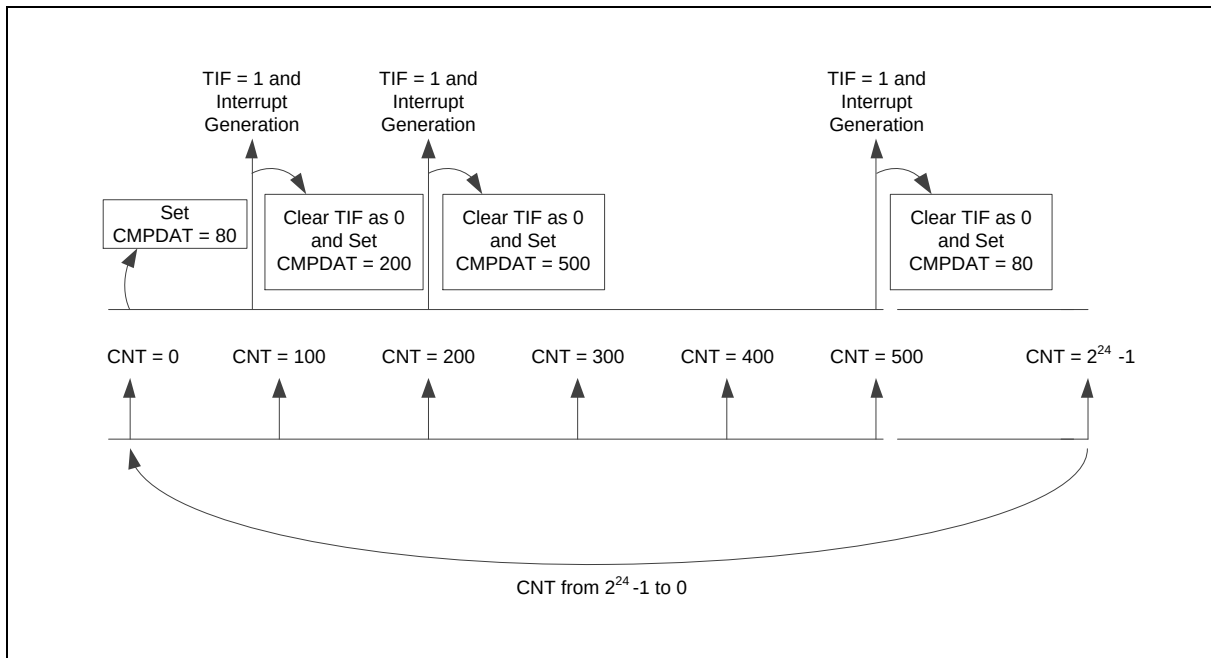


Figure 6.7-3 Continuous Counting Mode

6.7.6.3 Event Counting Mode

The timer controller also provides an application which can count the input event from TMx (x= 0~3) pin and the number of event will reflect to CNT (TIMERx_CNT[23:0]) value. It is also called as event counting function. In this function, EXTCNTEN (TIMERx_CTL[24]) should be set and the timer peripheral clock source should be set as PCLK.

If ECNTSSEL (TIMERx_EXTCTL[16]) is 0, the event counter source is from external TMx pin. User can enable or disable TMx pin de-bounce circuit by setting CNTDBEN (TIMERx_EXTCTL[7]). The input event frequency should be less than 1/3 PCLK if TMx pin de-bounce disabled or less than 1/8 PCLK if TMx pin de-bounce enabled to assure the returned CNT value is correct, and user can also select edge detection phase of TMx pin by setting CNTPHASE (TIMERx_EXTCTL[0]) bit.

In event counting mode, the timer counting operation mode can be selected as one-shot, periodic and continuous counting mode to counts the counter value CNT (TIMERx_CNT[23:0]) for TMx.

6.7.6.4 Capture Mode

The event capture function is used to load CNT (TIMERx_CNT[23:0]) value to CAPDAT (TIMERx_CAP[23:0]) value while edge transition detected on TMx_EXT (x= 0~3) pin, ACMP, internal clock and external clock. In this mode, CAPFUNCS (TIMERx_EXTCTL[4]) should be as 0 to trigger event capture function and the timer peripheral clock source should be set as PCLK.

If CAPSRC (TIMERx_CTL[22]) is 0, the capture event is triggered by TMx_EXT pin transition. User can enable or disable TMx_EXT pin de-bounce circuit by setting CAPDBEN (TIMERx_EXTCTL[6]). The transition frequency of TMx_EXT pin should be less than 1/3 PCLK if TMx_EXT pin de-bounce disabled or less than 1/8 PCLK if TMx_EXT pin de-bounce enabled to assure the capture function can be work normally, and user can also select edge transition detection of TMx_EXT pin by setting CAPEDGE (TIMERx_EXTCTL[14:12]).

In event capture mode, user does not consider what timer counting operation mode is selected, the capture event occurred only if edge transition on TMx_EXT pin is detected.

User can enable CAPIEN (TIMERx_EXTCTL[5]) to use capture interrupt function. When the TMx_EXT edge transition meets setting, CAPIF is high.

User must consider the Timer will keep register `TIMERx_CAP` unchanged and drop the new capture value if the CPU does not clear the `CAPIF` status.

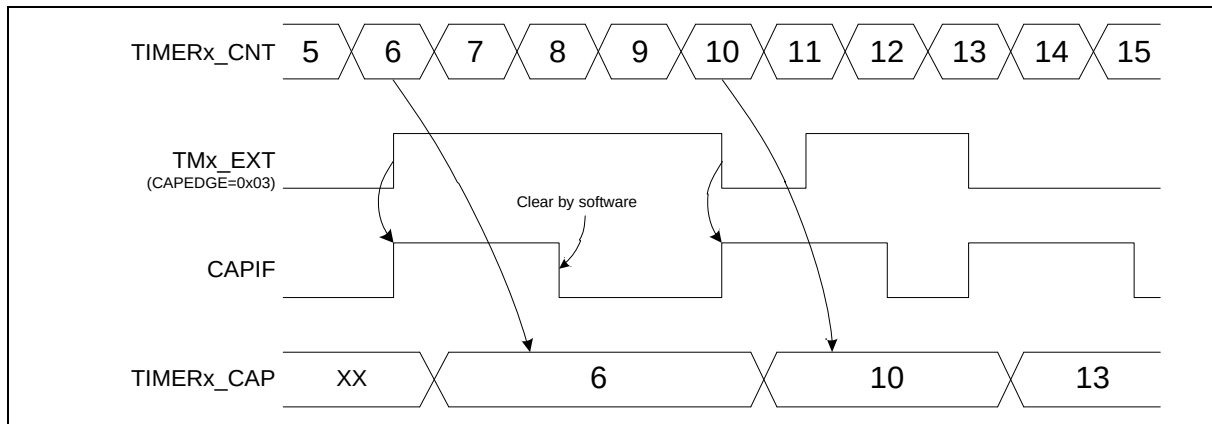


Figure 6.7-4 External Capture Mode

If `CAPSRC` (`TIMERx_CTL[22]`) is 1, set `INTERCAPSEL` (`TIMERx_EXTCTL[10:8]`) to choose different capture source. The capture event can be triggered by internal output signal transition on `ACMP0` if `INTERCAPSEL` (`TIMERx_EXTCTL[10:8]`) is 000, or `ACMP1` if `INTERCAPSEL` is 001 ; Other capture sources are `HXT`, `LXT`, `HIRC`, `LIRC` if `INTERCAPSEL` is 010, 011, 100, 101 respectively. For example, if user wants to switch `LXT` or `LIRC`, both of capture source should be set off first.

The capture source can be divided by capture divider. User can set `CAPDIVSCL` (`TIMERx_EXTCTL[31:28]`) to select different divider number. Be aware that timer clock frequency must be four times than capture source at least.

6.7.6.5 Reset Counter Mode

The timer controller also provides reset counter function to reset `CNT` (`TIMERx_CNT[23:0]`) value while capture event is generated. In this mode, `CAPFUNCS` (`TIMERx_EXTCTL[4]`) should be 1. User must set `CAPSRC` and `INTERCAPSEL` to select `TMx_EXT` transition, internal `ACMPx` output signal and internal clock or external clock to trigger reset counter value.

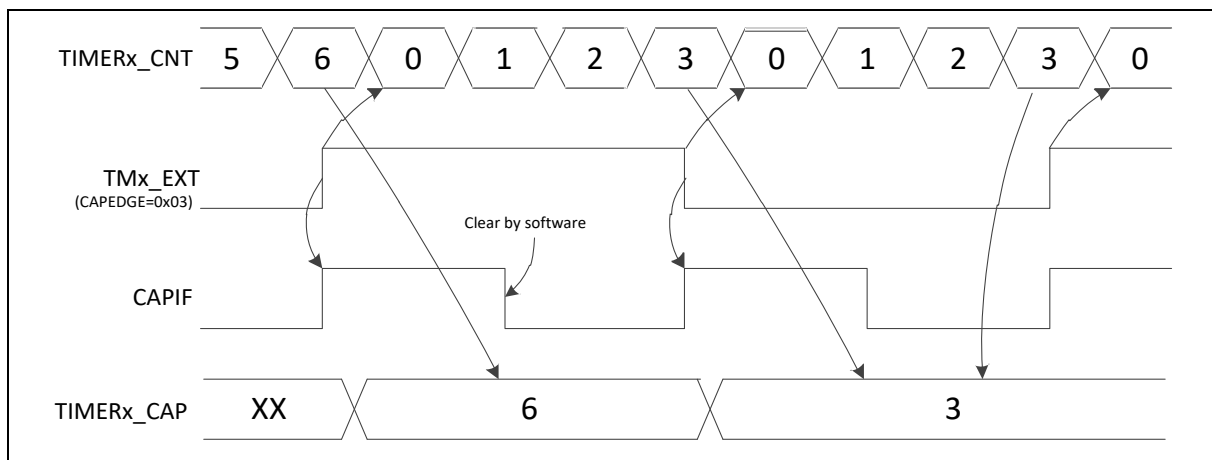


Figure 6.7-5 Reset Counter Mode

6.7.6.6 Timer Trigger Function

The timer controller provides timer time-out interrupt or capture interrupt to trigger `BPWM`, `EADC`, `DAC` and `PDMA`. If `TRGSSEL` (`TIMERx_TRGCTL[0]`) is 0, time-out interrupt signal is used to trigger `BPWM`, `EADC`, `DAC` and `PDMA`. If `TRGSSEL` (`TIMERx_TRGCTL[0]`) is 1, capture interrupt signal is used to trigger `BPWM`, `EADC`, `DAC` and `PDMA`.

When the TRGPWM (TIMERx_TRGCTL[1]) is set, if the timer interrupt signal is generated, the timer controller will generate a trigger pulse as BPWM external clock source.

When the TRGEADC (TIMERx_TRGCTL[2]) is set, if the timer interrupt signal is generated, the timer controller will trigger EADC to start converter.

When the TRGDAC (TIMERx_TRGCTL[3]) is set, if the timer interrupt signal is generated, the timer controller will trigger DAC to start converter.

When the TRGPDMA (TIMERx_TRGCTL[4]) is set, if the timer interrupt signal is generated, the timer controller will trigger PDMA.

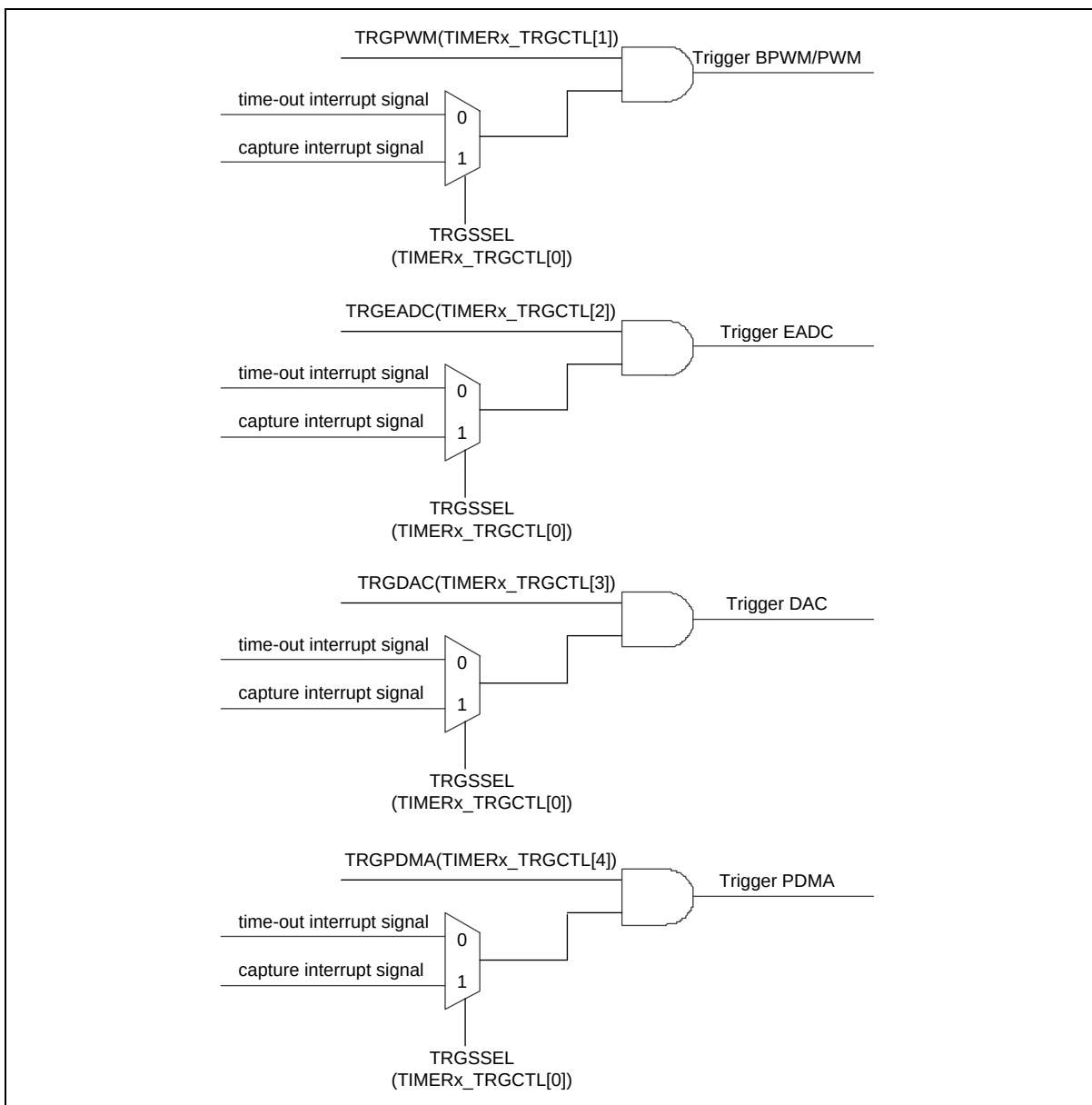


Figure 6.7-6 Internal Timer Trigger

When both the TRGPDMA (TIMERx_TRGCTL[4]) and TRGSSEL (TIMERx_TRGCTL[0]) are 1, TIMERx_CAP will change when encountering timer capture edge no matter whether CPU clear CAPIF status or not. In Trigger EADC DAC or BPWM function, user must consider that the Timer will keep register TIMERx_CAP unchanged if the CPU does not clear the CAPIF status.

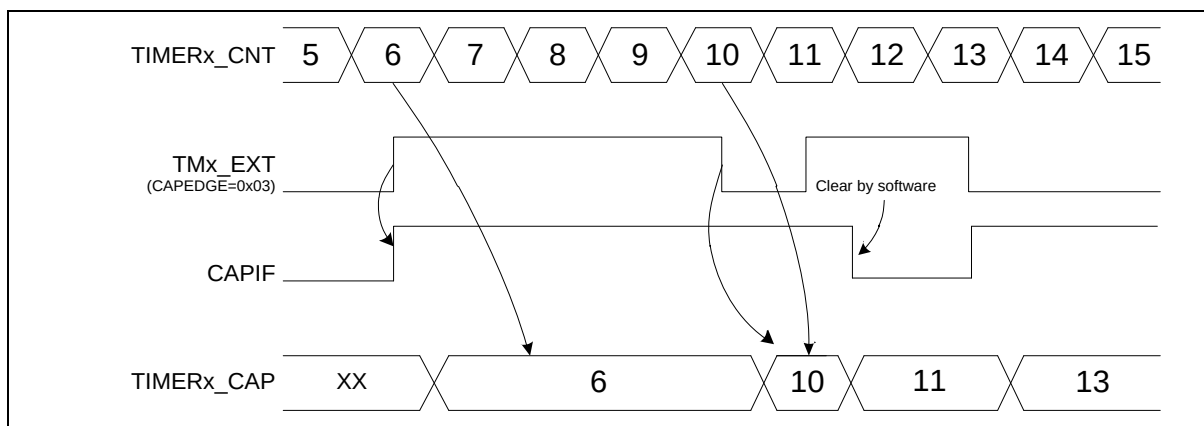


Figure 6.7-7 Capture interrupt trigger PDMA

6.7.6.7 Inter-Timer Trigger Capture Mode

In this mode, the Timer0/2 will be forced in event counting mode, counting with external event, and will generate an internal signal (INTR_TMR_TRG) to trigger Timer1/3 start or stop counting. Also, the Timer1/3 will be forced in capture mode and start/stop trigger-counting by Timer0/2 counter status.

Setting Timer0 Inter-timer Trigger Capture enabled, trigger-counting capture function is forced on Timer1. Setting Timer2 Inter-Timer Trigger Capture enabled, trigger-counting capture function is forced on Timer3.

Start Trigger

While INTRGEN (TIMERx_CTL[19]) in Timer0/2 is set, the Timer0/2 will make a rising-edge transition of INTR_TMR_TRG while Timer0/2 24-bit counter value (CNT) is counting from 0x0 to 0x1 and Timer1/3 counter will start counting immediately and automatically.

Stop Trigger

When Timer0/2 CNT reaches the Timer0/2 CMPDAT (TIMERx_CMP[23:0]) value, the Timer0/2 will make a falling-edge transition of INTR_TMR_TRG. Then Timer0/2 counter mode function will be disabled and INTRGEN (TIMERx_CTL[19]) will be cleared by hardware then Timer1/3 will stop counting also. At the same time, the Timer1/3 CNT value will be saved into Timer1/3 CAPDAT (TIMERx_CAP[23:0]).

User can use inter-timer trigger mode to measure the period of external event (TMx) more precisely. Figure 6.7-8 shows the sample flow of Inter-Timer Trigger Capture Mode for Timer0 as event counting mode and Timer1 as trigger-counting capture mode.

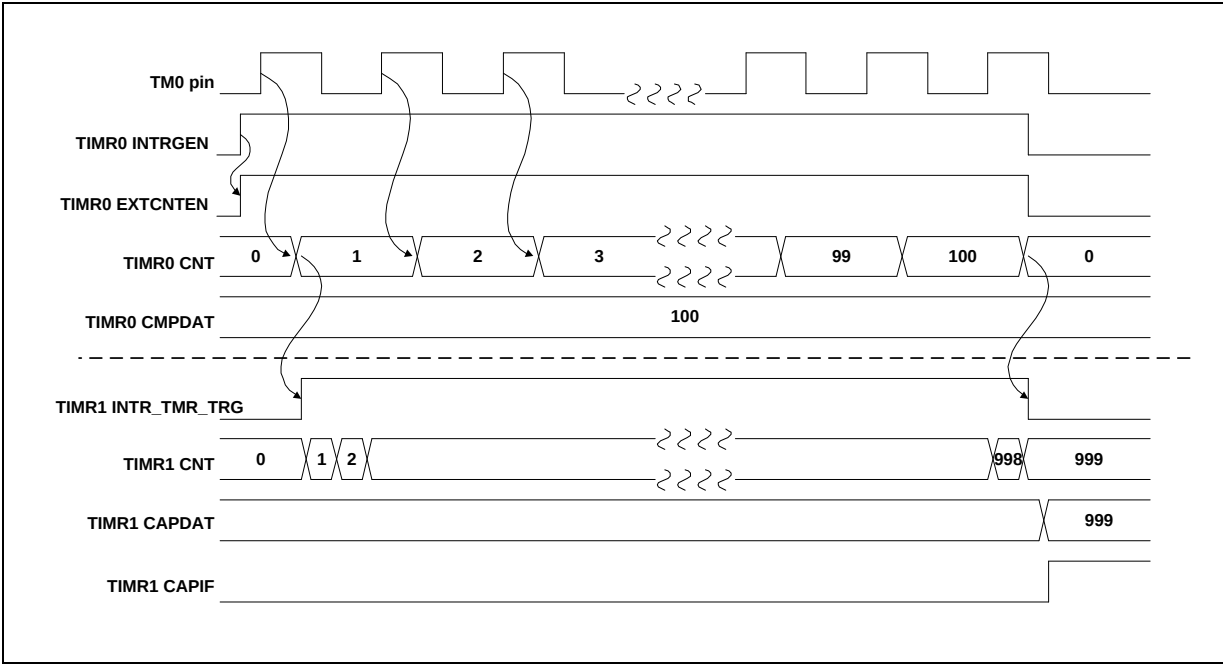


Figure 6.7-8 Inter-Timer Trigger Capture Timing

User must clear Timer1/3 CAPIF if user wants to use inter-timer trigger function again.

6.7.7 Register Map

R: read only, W: write only, R/W: both read and write

Register	Offset	R/W	Description	Reset Value
TIMER Base Address: TMR01_BA = 0x4005_0000 TMR23_BA = 0x4005_1000				
TIMER0_CTL	TMR01_BA+0x00	R/W	Timer0 Control Register	0x0000_0005
TIMER0_CMP	TMR01_BA+0x04	R/W	Timer0 Comparator Register	0x0000_0000
TIMER0_INTSTS	TMR01_BA+0x08	R/W	Timer0 Interrupt Status Register	0x0000_0000
TIMER0_CNT	TMR01_BA+0x0C	R/W	Timer0 Data Register	0x0000_0000
TIMER0_CAP	TMR01_BA+0x10	R	Timer0 Capture Data Register	0x0000_0000
TIMER0_EXTCTL	TMR01_BA+0x14	R/W	Timer0 External Control Register	0x0000_0000
TIMER0_EINTSTS	TMR01_BA+0x18	R/W	Timer0 External Interrupt Status Register	0x0000_0000
TIMER0_TRGCTL	TMR01_BA+0x1C	R/W	Timer0 Trigger Control Register	0x0000_0000
TIMER1_CTL	TMR01_BA+0x100	R/W	Timer1 Control Register	0x0000_0005
TIMER1_CMP	TMR01_BA+0x104	R/W	Timer1 Comparator Register	0x0000_0000
TIMER1_INTSTS	TMR01_BA+0x108	R/W	Timer1 Interrupt Status Register	0x0000_0000
TIMER1_CNT	TMR01_BA+0x10C	R/W	Timer1 Data Register	0x0000_0000
TIMER1_CAP	TMR01_BA+0x110	R	Timer1 Capture Data Register	0x0000_0000
TIMER1_EXTCTL	TMR01_BA+0x114	R/W	Timer1 External Control Register	0x0000_0000
TIMER1_EINTSTS	TMR01_BA+0x118	R/W	Timer1 External Interrupt Status Register	0x0000_0000
TIMER1_TRGCTL	TMR01_BA+0x11C	R/W	Timer1 Trigger Control Register	0x0000_0000
TIMER2_CTL	TMR23_BA+0x00	R/W	Timer2 Control Register	0x0000_0005
TIMER2_CMP	TMR23_BA+0x04	R/W	Timer2 Comparator Register	0x0000_0000
TIMER2_INTSTS	TMR23_BA+0x08	R/W	Timer2 Interrupt Status Register	0x0000_0000
TIMER2_CNT	TMR23_BA+0x0C	R/W	Timer2 Data Register	0x0000_0000
TIMER2_CAP	TMR23_BA+0x10	R	Timer2 Capture Data Register	0x0000_0000
TIMER2_EXTCTL	TMR23_BA+0x14	R/W	Timer2 External Control Register	0x0000_0000
TIMER2_EINTSTS	TMR23_BA+0x18	R/W	Timer2 External Interrupt Status Register	0x0000_0000
TIMER2_TRGCTL	TMR23_BA+0x1C	R/W	Timer2 Trigger Control Register	0x0000_0000
TIMER3_CTL	TMR23_BA+0x100	R/W	Timer3 Control Register	0x0000_0005
TIMER3_CMP	TMR23_BA+0x104	R/W	Timer3 Comparator Register	0x0000_0000

TIMER3_INTSTS	TMR23_BA+0x108	R/W	Timer3 Interrupt Status Register	0x0000_0000
TIMER3_CNT	TMR23_BA+0x10C	R/W	Timer3 Data Register	0x0000_0000
TIMER3_CAP	TMR23_BA+0x110	R	Timer3 Capture Data Register	0x0000_0000
TIMER3_EXTCTL	TMR23_BA+0x114	R/W	Timer3 External Control Register	0x0000_0000
TIMER3_EINTSTS	TMR23_BA+0x118	R/W	Timer3 External Interrupt Status Register	0x0000_0000
TIMER3_TRGCTL	TMR23_BA+0x11C	R/W	Timer3 Trigger Control Register	0x0000_0000

6.7.8 Register Description

Timer Control Register (TIMERx_CTL)

Register	Offset	R/W	Description	Reset Value
TIMER0_CTL	TMR01_BA+0x00	R/W	Timer0 Control Register	0x0000_0005
TIMER1_CTL	TMR01_BA+0x100	R/W	Timer1 Control Register	0x0000_0005
TIMER2_CTL	TMR23_BA+0x00	R/W	Timer2 Control Register	0x0000_0005
TIMER3_CTL	TMR23_BA+0x100	R/W	Timer3 Control Register	0x0000_0005

31	30	29	28	27	26	25	24
ICEDEBUG	CNTEN	INTEN	OPMODE		Reserved	ACTSTS	EXTCNTEN
23	22	21	20	19	18	17	16
WKEN	CAPSRC	TGLPINSEL	PERIOSEL	INTRGEN	Reserved		
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
PSC							

Bits	Description
[31]	ICEDEBUG ICE Debug Mode Acknowledge Disable Bit (Write Protect) 0 = ICE debug mode acknowledgement effects TIMER counting. TIMER counter will be held while CPU is held by ICE. 1 = ICE debug mode acknowledgement Disabled. TIMER counter will keep going no matter CPU is held by ICE or not. Note: This bit is write protected. Refer to the SYS_REGLCTL register.
[30]	CNTEN Timer Counting Enable Bit 0 = Stops/Suspends counting. 1 = Starts counting. Note 1: In stop status, and then set CNTEN to 1 will enable the 24-bit up counter to keep counting from the last stop counting value. Note 2: This bit is auto-cleared by hardware in one-shot mode (TIMER_CTL[28:27] = 00) when the timer time-out interrupt flag TIF (TIMERx_INTSTS[0]) is generated. Note 3: Set enable/disable this bit needs 2 * TMR_CLK period to become active, user can read ACTSTS (TIMERx_CTL[25]) to check enable/disable command is completed or not.
[29]	INTEN Timer Interrupt Enable Bit 0 = Timer time-out interrupt Disabled. 1 = Timer time-out interrupt Enabled. Note: If this bit is enabled, when the timer time-out interrupt flag TIF is set to 1, the timer interrupt signal is generated and inform to CPU.
[28:27]	OPMODE Timer Counting Mode Select 00 = The timer controller is operated in One-shot mode.

		01 = The timer controller is operated in Periodic mode. 10 = The timer controller is operated in Toggle-output mode. 11 = The timer controller is operated in Continuous Counting mode.
[26]	Reserved	Reserved.
[25]	ACTSTS	Timer Active Status Bit (Read Only) This bit indicates the 24-bit up counter status. 0 = 24-bit up counter is not active. 1 = 24-bit up counter is active. Note: This bit may active when CNT 0 transition to CNT 1.
[24]	EXTCNTEN	Event Counter Mode Enable Bit This bit is for external counting pin function enabled. 0 = Event counter mode Disabled. 1 = Event counter mode Enabled. Note: When timer is used as an event counter, this bit should be set to 1 and select PCLK as timer clock source.
[23]	WKEN	Wake-up Function Enable Bit If this bit is set to 1, while timer interrupt flag TIF (TIMERx_INTSTS[0]) is 1 and INTEN (TIMERx_CTL[29]) is enabled, the timer interrupt signal will generate a wake-up trigger event to CPU. 0 = Wake-up function Disabled if timer interrupt signal generated. 1 = Wake-up function Enabled if timer interrupt signal generated.
[22]	CAPSRC	Capture Pin Source Selection 0 = Capture Function source is from TMx_EXT (x= 0~3) pin. 1 = Capture Function source is from internal ACMP output signal, internal clock (LIRC, HIRC), or external clock (HXT, LXT). Note: When CAPSRC = 1, User can set INTERCAPSEL (TIMERx_EXTCTL[10:8]) to decide which internal ACMP output signal or which clock is as timer capture source.
[21]	TGLPINSEL	Toggle-output Pin Select 0 = Toggle mode output to TMx (Timer Event Counter Pin). 1 = Toggle mode output to TMx_EXT (Timer External Capture Pin).
[20]	PERIOSEL	Periodic Mode Behavior Selection Enable Bit 0 = The behavior selection in periodic mode is Disabled. When user updates CMPDAT while timer is running in periodic mode, CNT will be reset to default value. 1 = The behavior selection in periodic mode is Enabled. When user updates CMPDAT while timer is running in periodic mode, the limitations as bellows list, If updated CMPDAT value > CNT, CMPDAT will be updated and CNT keep running continually. If updated CMPDAT value = CNT, timer time-out interrupt will be asserted immediately. If updated CMPDAT value < CNT, CNT will be reset to default value.
[19]	INTRGEN	Inter-timer Trigger Mode Enable Bit Setting this bit will enable the inter-timer trigger capture function. The Timer0/2 will be in event counter mode and counting with external clock source or event. Also, Timer1/3 will be in trigger-counting mode of capture function. 0 = Inter-Timer Trigger Capture mode Disabled. 1 = Inter-Timer Trigger Capture mode Enabled. Note: For Timer1/3, this bit is ineffective and the read back value is always 0.
[18:8]	Reserved	Reserved.
[7:0]	PSC	Prescale Counter

		Timer input clock or event source is divided by (PSC+1) before it is fed to the timer up counter. If this field is 0 (PSC = 0), then there is no scaling. Note: Update prescale counter value will reset internal 8-bit prescale counter and 24-bit up counter value.

Timer Comparator Register (TIMERx_CMP)

Register	Offset	R/W	Description	Reset Value
TIMER0_CMP	TMR01_BA+0x04	R/W	Timer0 Comparator Register	0x0000_0000
TIMER1_CMP	TMR01_BA+0x104	R/W	Timer1 Comparator Register	0x0000_0000
TIMER2_CMP	TMR23_BA+0x04	R/W	Timer2 Comparator Register	0x0000_0000
TIMER3_CMP	TMR23_BA+0x104	R/W	Timer3 Comparator Register	0x0000_0000

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

Bits	Description
[31:24]	Reserved Reserved.
[23:0]	Timer Comparator Value CMPDAT is a 24-bit compared value register. When the internal 24-bit up counter value is equal to CMPDAT value, the TIF (TIMERx_INTSTS[0] Timer Interrupt Flag) will set to 1. Time-out period = (Period of timer clock input) * (8-bit PSC + 1) * (24-bit CMPDAT). Note 1: Never write 0x0 or 0x1 in CMPDAT field, or the core will run into unknown state. Note 2: When timer is operating at continuous counting mode, the 24-bit up counter will keep counting continuously even if user writes a new value into CMPDAT field. But if timer is operating at other modes, the 24-bit up counter will restart counting from 0 and using newest CMPDAT value to be the timer compared value while user writes a new value into CMPDAT field.

Timer Interrupt Status Register (TIMERx INTSTS)

Register	Offset	R/W	Description	Reset Value
TIMER0_INTSTS	TMR01_BA+0x08	R/W	Timer0 Interrupt Status Register	0x0000_0000
TIMER1_INTSTS	TMR01_BA+0x108	R/W	Timer1 Interrupt Status Register	0x0000_0000
TIMER2_INTSTS	TMR23_BA+0x08	R/W	Timer2 Interrupt Status Register	0x0000_0000
TIMER3_INTSTS	TMR23_BA+0x108	R/W	Timer3 Interrupt Status Register	0x0000_0000

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

Bits	Description
[31:2]	Reserved Reserved.
[1]	Timer Wake-up Flag This bit indicates the interrupt wake-up flag status of timer. 0 = Timer does not cause CPU wake-up. 1 = CPU wake-up from Idle or Power-down mode if timer time-out interrupt signal generated. Note: This bit is cleared by writing 1 to it.
[0]	Timer Interrupt Flag This bit indicates the interrupt flag status of Timer while 24-bit timer up counter CNT (TIMERx_CNT[23:0]) value reaches to CMPDAT (TIMERx_CMP[23:0]) value. 0 = No effect. 1 = CNT value matches the CMPDAT value. Note: This bit is cleared by writing 1 to it.

Timer Data Register (TIMERx_CNT)

Register	Offset	R/W	Description	Reset Value
TIMER0_CNT	TMR01_BA+0x0C	R/W	Timer0 Data Register	0x0000_0000
TIMER1_CNT	TMR01_BA+0x10C	R/W	Timer1 Data Register	0x0000_0000
TIMER2_CNT	TMR23_BA+0x0C	R/W	Timer2 Data Register	0x0000_0000
TIMER3_CNT	TMR23_BA+0x10C	R/W	Timer3 Data Register	0x0000_0000

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

Bits	Description
[31]	RSTACT Timer Data Register Reset Active (Read Only) This bit indicates if the counter reset operation active. When user writes this CNT register, timer starts to reset its internal 24-bit timer up-counter to 0 and reload 8-bit pre-scale counter. At the same time, timer set this flag to 1 to indicate the counter reset operation is in progress. Once the counter reset operation done, timer clear this bit to 0 automatically. 0 = Reset operation is done. 1 = Reset operation triggered by writing TIMERx_CNT is in progress.
[30:24]	Reserved Reserved.
[23:0]	CNT Timer Data Register Read operation. Read this register to get CNT value. For example: If EXTCNTEN (TIMERx_CTL[24]) is 0, user can read CNT value for getting current 24-bit counter value. If EXTCNTEN (TIMERx_CTL[24]) is 1, user can read CNT value for getting current 24-bit event input counter value. Write operation. Writing any value to this register will reset current CNT value to 0 and reload internal 8-bit prescale counter.

Timer Capture Data Register (TIMERx_CAP)

Register	Offset	R/W	Description	Reset Value
TIMER0_CAP	TMR01_BA+0x10	R	Timer0 Capture Data Register	0x0000_0000
TIMER1_CAP	TMR01_BA+0x110	R	Timer1 Capture Data Register	0x0000_0000
TIMER2_CAP	TMR23_BA+0x10	R	Timer2 Capture Data Register	0x0000_0000
TIMER3_CAP	TMR23_BA+0x110	R	Timer3 Capture Data Register	0x0000_0000

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

Bits	Description
[31:24]	Reserved Reserved.
[23:0]	CAPDAT Timer Capture Data Register When CAPEN (TIMERx_EXTCTL[3]) bit is set and a transition on TMx_EXT pin matched the CAPEDGE (TIMERx_EXTCTL[14:12]) setting, CAPIF (TIMERx_EINTSTS[0]) will set to 1 and the current timer counter value CNT (TIMERx_CNT[23:0]) will be auto-loaded into this CAPDAT field. Note: User must consider the Timer will keep register TIMERx_CAP unchanged and drop the new capture value if the CPU does not clear the CAPIF status.

Timer External Control Register (TIMERx_EXTCTL)

Register	Offset	R/W	Description	Reset Value
TIMER0_EXTCTL	TMR01_BA+0x14	R/W	Timer0 External Control Register	0x0000_0000
TIMER1_EXTCTL	TMR01_BA+0x114	R/W	Timer1 External Control Register	0x0000_0000
TIMER2_EXTCTL	TMR23_BA+0x14	R/W	Timer2 External Control Register	0x0000_0000
TIMER3_EXTCTL	TMR23_BA+0x114	R/W	Timer3 External Control Register	0x0000_0000

31	30	29	28	27	26	25	24
CAPDIVSCL				Reserved			
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved	CAPEDGE			Reserved	INTERCAPSEL		
7	6	5	4	3	2	1	0
CNTDBEN	CAPDBEN	CAPIEN	CAPFUNCS	CAPEN	Reserved		CNTPHASE

Bits	Description
[31:28]	CAPDIVSCL Timer Capture Source Divider Scale This bits indicate the divide scale for capture source divider 0000 = Capture source/1. 0001 = Capture source/2. 0010 = Capture source/4. 0011 = Capture source/8. 0100 = Capture source/16. 0101 = Capture source/32. 0110 = Capture source/64. 0111 = Capture source/128. 1000 = Capture source/256. 1001~1111 =Reserved. Note: Sets INTERCAPSEL (TIMERx_EXTCTL[10:8]) and CAPSRC (TIMERx_CTL[22]) to select capture source.
[27:15]	Reserved Reserved.
[14:12]	CAPEDGE Timer External Capture Pin Edge Detect When first capture event is generated, the CNT (TIMERx_CNT[23:0]) will be reset to 0 and first CAPDAT (TIMERx_CAP[23:0]) should be to 0. 000 = Capture event occurred when detect falling edge transfer on TMx_EXT (x= 0~3) pin. 001 = Capture event occurred when detect rising edge transfer on TMx_EXT (x= 0~3) pin. 010 = Capture event occurred when detect both falling and rising edge transfer on TMx_EXT (x= 0~3) pin, and first capture event occurred at falling edge transfer. 011 = Capture event occurred when detect both rising and falling edge transfer on TMx_EXT (x= 0~3) pin, and first capture event occurred at rising edge transfer.

		110 = First capture event occurred at falling edge, follows capture events are at rising edge transfer on TMx_EXT (x= 0~3) pin. 111 = First capture event occurred at rising edge, follows capture events are at falling edge transfer on TMx_EXT (x= 0~3) pin. 100, 101 = Reserved. Note: Set CAPSRC (TIMERx_CTL[22]) and INTERCAPSEL (TIMERx_EXTCTL[10:8]) to select capture source.
[11]	Reserved	Reserved.
[10:8]	INTERCAPSEL	Internal Capture Source Select 000 = Capture Function source is from internal ACMP0 output signal. 001 = Capture Function source is from internal ACMP1 output signal. 010 = Capture Function source is from HXT. 011 = Capture Function source is from LXT. 100 = Capture Function source is from HIRC. 101 = Capture Function source is from LIRC. 110 = Capture Function source is from MIRC. 111 = Reserved. Note: these bits only available when CAPSRC (TIMERx_CTL[22]) is 1.
[7]	CNTDBEN	Timer Counter Pin De-bounce Enable Bit 0 = TMx (x= 0~3) pin de-bounce Disabled. 1 = TMx (x= 0~3) pin de-bounce Enabled. Note: If this bit is enabled, the edge detection of TMx pin is detected with de-bounce circuit.
[6]	CAPDBEN	Timer External Capture Pin De-bounce Enable Bit 0 = TMx_EXT (x= 0~3) pin de-bounce or ACMP output de-bounce Disabled. 1 = TMx_EXT (x= 0~3) pin de-bounce or ACMP output de-bounce Enabled. Note: If this bit is enabled, the edge detection of TMx_EXT pin or ACMP output is detected with de-bounce circuit.
[5]	CAPIEN	Timer External Capture Interrupt Enable Bit 0 = TMx_EXT (x= 0~3) pin, ACMP, internal clock, or external clock detection Interrupt Disabled. 1 = TMx_EXT (x= 0~3) pin, ACMP, internal clock, or external clock detection Interrupt Enabled. Note: CAPIEN is used to enable timer external interrupt. If CAPIEN enabled, timer will rise an interrupt when CAPIF (TIMERx_EINTSTS[0]) is 1. For example, while CAPIEN = 1, CAPEN = 1, and CAPEDGE = 00, a 1 to 0 transition on the TMx_EXT (x= 0~3) pin, ACMP, internal clock, or external clock will cause the CAPIF to be set then the interrupt signal is generated and sent to NVIC to inform CPU.
[4]	CAPFUNCS	Capture Function Selection 0 = External Capture Mode Enabled. 1 = External Reset Mode Enabled. Note 1: When CAPFUNCS is 0, transition on TMx_EXT (x= 0~3) pin is using to save current 24-bit timer counter value (CNT value) to CAPDAT field. Note 2: When CAPFUNCS is 1, transition on TMx_EXT (x= 0~3) pin is using to save current 24-bit timer counter value (CNT value) to CAPDAT field then CNT value will be reset immediately.
[3]	CAPEN	Timer Capture Enable Bit This bit enables the capture input function. 0 = Capture source Disabled. 1 = Capture source Enabled. Note: When CAPEN is 1, user can set INTERCAPSEL (TIMERx_EXTCTL [10:8]) and CAPSRC (TIMERx_CTL[22]) to select capture source.
[2:1]	Reserved	Reserved.

[0]	CNTPHASE	Timer External Count Phase This bit indicates the detection phase of external counting pin TMx (x= 0~3). 0 = A falling edge of external counting pin will be counted. 1 = A rising edge of external counting pin will be counted.

Timer External Interrupt Status Register (TIMERx EINTSTS)

Register	Offset	R/W	Description	Reset Value
TIMER0_EINTSTS	TMR01_BA+0x18	R/W	Timer0 External Interrupt Status Register	0x0000_0000
TIMER1_EINTSTS	TMR01_BA+0x118	R/W	Timer1 External Interrupt Status Register	0x0000_0000
TIMER2_EINTSTS	TMR23_BA+0x18	R/W	Timer2 External Interrupt Status Register	0x0000_0000
TIMER3_EINTSTS	TMR23_BA+0x118	R/W	Timer3 External Interrupt Status Register	0x0000_0000

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

Bits	Description
[31:1]	Reserved Reserved.
[0]	CAPIF Timer External Capture Interrupt Flag This bit indicates the timer external capture interrupt flag status. 0 = TMx_EXT (x= 0~3) pin, ACMP, internal clock, or external clock interrupt did not occur. 1 = TMx_EXT (x= 0~3) pin, ACMP, internal clock, or external clock interrupt occurred. Note 1: This bit is cleared by writing 1 to it. Note 2: When CAPEN (TIMERx_EXTCTL[3]) bit is set, CAPFUNCS (TIMERx_EXTCTL[4]) bit is 0, and a transition on TMx_EXT (x= 0~3) pin, ACMP, internal clock, or external clock matched the CAPEDGE (TIMERx_EXTCTL[14:12]) setting, this bit will set to 1 by hardware. Note 3: There is a new incoming capture event detected before CPU clearing the CAPIF status. If the above condition occurred, the Timer will keep register TIMERx_CAP unchanged and drop the new capture value.

Timer Trigger Control Register (TIMERx_TRGCTL)

Register	Offset	R/W	Description	Reset Value
TIMER0_TRGCTL	TMR01_BA+0x1C	R/W	Timer0 Trigger Control Register	0x0000_0000
TIMER1_TRGCTL	TMR01_BA+0x11C	R/W	Timer1 Trigger Control Register	0x0000_0000
TIMER2_TRGCTL	TMR23_BA+0x1C	R/W	Timer2 Trigger Control Register	0x0000_0000
TIMER3_TRGCTL	TMR23_BA+0x11C	R/W	Timer3 Trigger Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved			TRGPDMA	TRGDAC	TRGEADC	TRGPWM	TRGSSEL

Bits	Description
[31:8]	Reserved Reserved.
[7:5]	Reserved Reserved.
[4]	TRGPDMA Trigger PDMA Enable Bit If this bit is set to 1, each timer time-out event or capture event can be triggered PDMA transfer. 0 = Timer interrupt trigger PDMA Disabled. 1 = Timer interrupt trigger PDMA Enabled. Note: If TRGSSEL (TIMERx_TRGCTL[0]) = 0, time-out interrupt signal will trigger PDMA transfer. If TRGSSEL (TIMERx_TRGCTL[0]) = 1, capture interrupt signal will trigger PDMA transfer.
[3]	TRGDAC Trigger DAC Enable Bit If this bit is set to 1, timer time-out interrupt or capture interrupt can be triggered DAC. 0 = Timer interrupt trigger DAC Disabled. 1 = Timer interrupt trigger DAC Enabled. Note: If TRGSSEL (TIMERx_TRGCTL[0]) = 0, time-out interrupt signal will trigger DAC. If TRGSSEL (TIMERx_TRGCTL[0]) = 1, capture interrupt signal will trigger DAC.
[2]	TRGEADC Trigger EADC Enable Bit If this bit is set to 1, each timer time-out event or capture event can be triggered EADC conversion. 0 = Timer interrupt trigger EADC Disabled. 1 = Timer interrupt trigger EADC Enabled. Note: If TRGSSEL (TIMERx_TRGCTL[0]) = 0, time-out interrupt signal will trigger EADC conversion. If TRGSSEL (TIMERx_TRGCTL[0]) = 1, capture interrupt signal will trigger EADC conversion.
[1]	TRGPWM Trigger BPWM Enable Bit

		If this bit is set to 1, each timer time-out event or capture event can be as BPWM counter clock source. 0 = Timer interrupt trigger BPWM Disabled. 1 = Timer interrupt trigger BPWM Enabled. Note: If TRGSSEL (TIMERx_TRGCTL[0]) = 0, time-out interrupt signal as BPWM counter clock source. If TRGSSEL (TIMERx_TRGCTL[0]) = 1, capture interrupt signal as BPWM counter clock source.
[0]	TRGSSEL	Trigger Source Select Bit This bit is used to select internal trigger source is form timer time-out interrupt signal or capture interrupt signal. 0 = Time-out interrupt signal is used to internal trigger BPWM, PDMA, DAC, and EADC. 1 = Capture interrupt signal is used to internal trigger BPWM, PDMA, DAC, and EADC.

6.8 Watchdog Timer (WDT)

6.8.1 Overview

The Watchdog Timer (WDT) is used to perform a system reset when system runs into an unknown state. This prevents system from hanging for an infinite period of time. Besides, this Watchdog Timer supports the function to wake up system from Idle/Power-down mode.

6.8.2 Features

- 20-bit free running up counter for WDT time-out interval
- Selectable time-out interval ($2^4 \sim 2^{20}$) and the time-out interval is $417\mu\text{s} \sim 27.3\text{ s}$ if $\text{WDT\_CLK} = 38.4\text{ kHz (LIRC)}$.
- System kept in reset state for a period of $(1 / \text{WDT\_CLK}) * 63$
- Supports selectable WDT reset delay period, including 1026, 130, 18 or 3 WDT_CLK reset delay period
- Supports to force WDT enabled after chip powered on or reset by setting $\text{CWDTEN}[2:0]$ in Config0 register
- Supports WDT time-out wake-up function only if WDT clock source is selected as 38.4 kHz LIRC or LXT.

6.8.3 Block Diagram

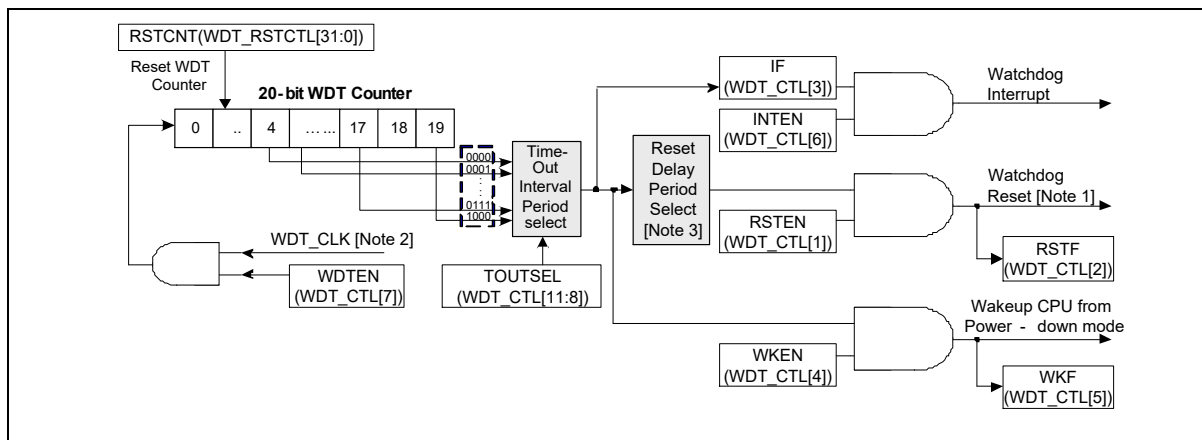


Figure 6.8-1 Watchdog Timer Block Diagram

Note1: WDT resets CPU and lasts 63 WDT CLK.

Note2: The WDT reset delay period can be selected as 3/18/130/1026 WDT_CLK.

Note3: Chip can be woken up by WDT time-out interrupt signal generated only if WDT clock source is selected to LIRC or LXT.

6.8.4 Basic Configuration

- Clock Source Configuration
 - Select the source of WDT peripheral clock on WDTSEL (CLK_CLKSEL1[1:0])
 - Enable WDT peripheral clock in WDTCKEN (CLK_APBCLK0[0]).
 - Force enable WDT controller after chip powered on or reset in CWDTEN[2:0] (CWDTEN[2] is Config0[31], CWDTEN[1:0] is Config0[4:3])

- WDT clock source can be changed only if CWDTEN[2:0] is 111

The WDT clock control is shown in Figure 6.8-2.

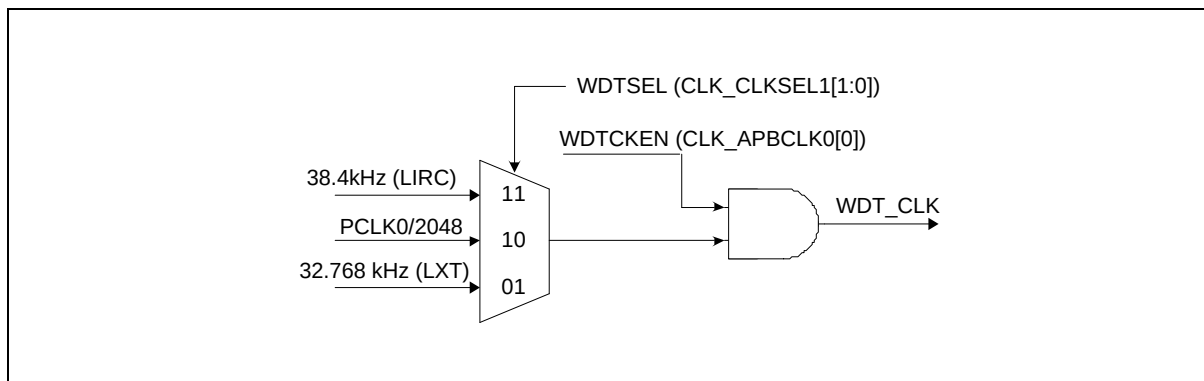


Figure 6.8-2 Watchdog Timer Clock Control

6.8.5 Functional Description

The WDT includes an 20-bit free running up counter with programmable time-out intervals. Table 6.8-1 shows the WDT time-out interval period selection and Figure 6.8-3 shows the WDT time-out interval and reset period timing.

6.8.5.1 WDT Time-out Interrupt

Setting WDTEEN (WDT_CTL [7]) to 1 will enable the WDT function and the WDT counter to start counting up. The SYNC (WDT_CTL[30]) can be indicated whether enable/disable WDTEEN function is completed or not. There are eight time-out interval period can be selected by setting TOUTSEL (WDT_CTL[10:8]). When the WDT up counter reaches the TOUTSEL (WDT_CTL[11:8]) settings, WDT time-out interrupt will occur then WDT time-out interrupt flag IF (WDT_CTL[3]) will be set to 1 immediately. If INTEN (WDT_CTL[6]) is enabled, WDT time-out interrupt will inform CPU.

6.8.5.2 WDT Reset Delay Period and Reset System

There is a specified T_{RSTD} reset delay period follows the IF (WDT_CTL[3]) is setting to 1. User should set WDT_RSTCNT to reset the 20-bit WDT up counter value to avoid generate WDT time-out reset signal before the T_{RSTD} reset delay period expires. Moreover, user should set RSTDSEL (WDT_ALTCTL [1:0]) to select reset delay period to clear WDT counter. If the WDT up counter value has not been cleared after the specific T_{RSTD} delay period expires, the WDT control will set RSTF (WDT_CTL[2]) to 1 if RSTEN (WDT_CTL[1]) bit is enabled, then chip enters to reset state immediately. T_{RST} reset period will keep last 63 WDT clocks then chip restart executing program from reset vector (0x0000_0000). The RSTF (WDT_CTL[2]) will keep 1 after WDT time-out resets the chip, user can check RSTF (WDT_CTL[2]) by software to recognize the system has been reset by WDT time-out reset or not.

TOUTSEL	Time-Out Interval Period T_{TIS}	Reset Delay Period T_{RSTD}
0000	$2^4 * T_{WDT}$	$(3/18/130/1026) * T_{WDT}$
0001	$2^6 * T_{WDT}$	$(3/18/130/1026) * T_{WDT}$
0010	$2^8 * T_{WDT}$	$(3/18/130/1026) * T_{WDT}$
0011	$2^{10} * T_{WDT}$	$(3/18/130/1026) * T_{WDT}$
0100	$2^{12} * T_{WDT}$	$(3/18/130/1026) * T_{WDT}$
0101	$2^{14} * T_{WDT}$	$(3/18/130/1026) * T_{WDT}$
0110	$2^{16} * T_{WDT}$	$(3/18/130/1026) * T_{WDT}$
0111	$2^{18} * T_{WDT}$	$(3/18/130/1026) * T_{WDT}$
1000	$2^{20} * T_{WDT}$	$(3/18/130/1026) * T_{WDT}$

Table 6.8-1 Watchdog Timer Time-out Interval Period Selection

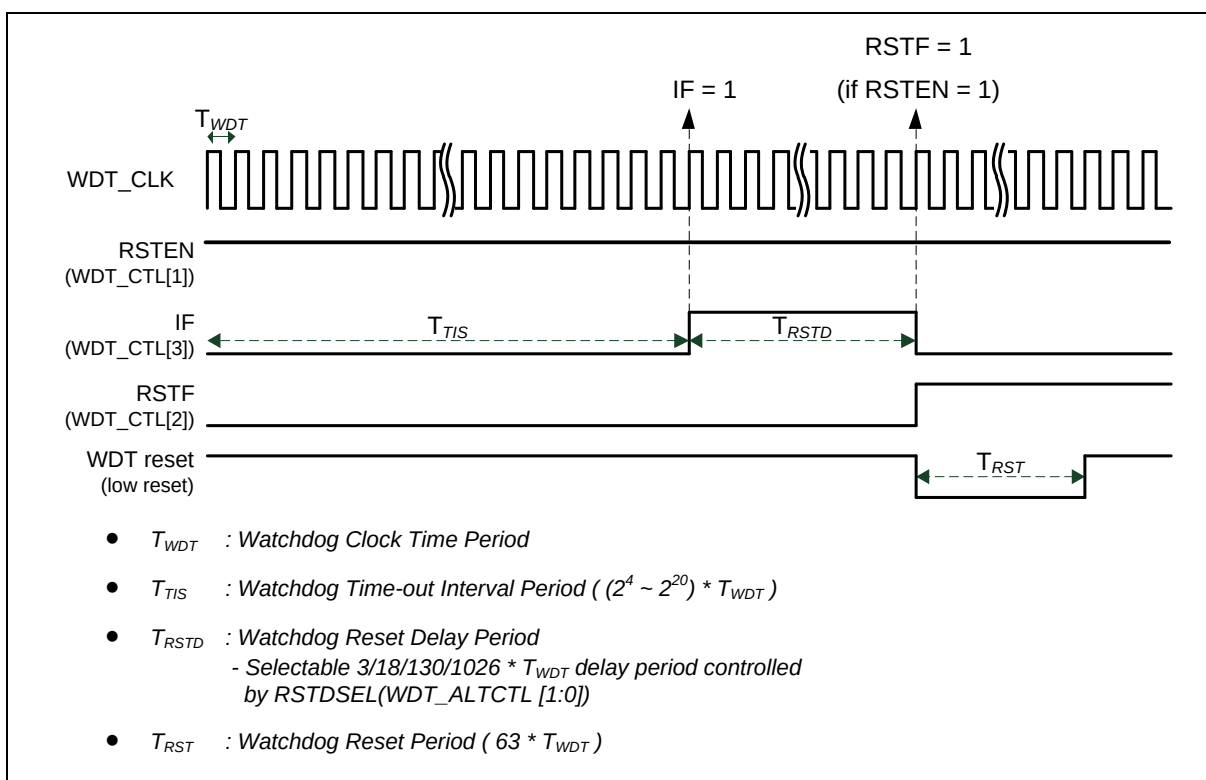


Figure 6.8-3 Watchdog Timer Time-out Interval and Reset Period Timing

6.8.5.3 WDT Wake-up

If WDT clock source is selected to LIRC or LXT, system can be woken up from Power-down mode while WDT time-out interrupt signal is generated and WKEN (WDT_CTL[4]) enabled. Note that user should set LXTEN (CLK_PWRCTL [1]) or LIRCEN (CLK_PWRCTL [3]) to select clock source before system enters Power-down mode because the system peripheral clock are disabled when system is in Power-down mode. In the meanwhile, the WKF (WDT_CTL[5]) will be set to 1 automatically, and user can check WKF (WDT_CTL[5]) status by software to recognize the system has been woken up by WDT time-out interrupt or not.

6.8.5.4 WDT ICE Debug

When ICE is connected to MCU, WDT counter is counting or not by ICEDEBUG (WDT_CTL[31]). The default value of ICEDEBUG is 0, WDT counter will stop counting when CPU is held by ICE. If ICEDEBUG is set to 1, WDT counter will keep counting no matter CPU is held by ICE or not.

6.8.6 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
WDT Base Address: WDT_BA = 0x4004_0000				
WDT_CTL	WDT_BA+0x00	R/W	WDT Control Register	0x0000_0800
WDT_ALTCTL	WDT_BA+0x04	R/W	WDT Alternative Control Register	0x0000_0000
WDT_RSTCNT	WDT_BA+0x08	W	WDT Reset Counter Register	0x0000_0000

6.8.7 Register Description

WDT Control Register (WDT_CTL)

Register	Offset	R/W	Description	Reset Value
WDT_CTL	WDT_BA+0x00	R/W	WDT Control Register	0x0000_0800

31	30	29	28	27	26	25	24
ICEDEBUG	SYNC	Reserved					
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved				TOUTSEL			
7	6	5	4	3	2	1	0
WDTEN	INTEN	WKF	WKEN	IF	RSTF	RSTEN	Reserved

Bits	Description
[31]	ICEDEBUG ICE Debug Mode Acknowledge Disable Bit (Write Protect) 0 = ICE debug mode acknowledgement affects WDT counting. WDT up counter will be held while CPU is held by ICE. 1 = ICE debug mode acknowledgement Disabled. WDT up counter will keep going no matter CPU is held by ICE or not. Note: This bit is write protected. Refer to the SYS_REGLCTL register.
[30]	SYNC WDT Enable Control SYNC Flag Indicator (Read Only) If user executes enable/disable WDTEN (WDT_CTL[7]), this flag can be indicated enable/disable WDTEN function is completed or not. 0 = Set WDTEN bit is completed. 1 = Set WDTEN bit is synchronizing and not become active yet. Note: Performing enable or disable WDTEN bit needs 2 * WDT_CLK period to become active.
[29:11]	Reserved Reserved.
[11:8]	TOUTSEL WDT Time-out Interval Selection (Write Protect) These four bits select the time-out interval period for the WDT. 0000 = $2^4 * \text{WDT\_CLK}$. 0001 = $2^6 * \text{WDT\_CLK}$. 0010 = $2^8 * \text{WDT\_CLK}$. 0011 = $2^{10} * \text{WDT\_CLK}$. 0100 = $2^{12} * \text{WDT\_CLK}$. 0101 = $2^{14} * \text{WDT\_CLK}$. 0110 = $2^{16} * \text{WDT\_CLK}$. 0111 = $2^{18} * \text{WDT\_CLK}$. 1000 = $2^{20} * \text{WDT\_CLK}$. Note: This bit is write protected. Refer to the SYS_REGLCTL register.

[7]	WDTEN	WDT Enable Bit (Write Protect) 0 = WDT Disabled (This action will reset the internal up counter value). 1 = WDT Enabled. Note1: This bit is write protected. Refer to the SYS_REGLCTL register. Note2: If CWDTEN[2:0] (combined by Config0[31] and Config0[4:3]) bits is not configured to 111, this bit is forced as 1 and user cannot change this bit to 0.
[6]	INTEN	WDT Time-out Interrupt Enable Bit (Write Protect) If this bit is enabled, the WDT time-out interrupt signal is generated and inform to CPU. 0 = WDT time-out interrupt Disabled. 1 = WDT time-out interrupt Enabled. Note: This bit is write protected. Refer to the SYS_REGLCTL register.
[5]	WKF	WDT Time-out Wake-up Flag (Write Protect) This bit indicates the interrupt wake-up flag status of WDT 0 = WDT does not cause chip wake-up. 1 = Chip wake-up from Idle or Power-down mode if WDT time-out interrupt signal generated. Note1: This bit is write protected. Refer to the SYS_REGLCTL register. Note2: This bit is cleared by writing 1 to it.
[4]	WKEN	WDT Time-out Wake-up Function Control (Write Protect) If this bit is set to 1, while WDT time-out interrupt flag IF (WDT_CTL[3]) is generated to 1 and interrupt enable bit INTEN (WDT_CTL[6]) is enabled, the WDT time-out interrupt signal will generate a wake-up trigger event to chip. 0 = Wake-up trigger event Disabled if WDT time-out interrupt signal generated. 1 = Wake-up trigger event Enabled if WDT time-out interrupt signal generated. Note1: This bit is write protected. Refer to the SYS_REGLCTL register. Note2: Chip can be woken up by WDT time-out interrupt signal generated only if WDT clock source is selected to 38.4 kHz internal low speed RC oscillator (LIRC) or LXT.
[3]	IF	WDT Time-out Interrupt Flag This bit will set to 1 while WDT up counter value reaches the selected WDT time-out interval 0 = WDT time-out interrupt did not occur. 1 = WDT time-out interrupt occurred. Note: This bit is cleared by writing 1 to it.
[2]	RSTF	WDT Time-out Reset Flag This bit indicates the system has been reset by WDT time-out reset or not. 0 = WDT time-out reset did not occur. 1 = WDT time-out reset occurred. Note: This bit is cleared by writing 1 to it.
[1]	RSTEN	WDT Time-out Reset Enable Bit (Write Protect) Setting this bit will enable the WDT time-out reset function if the WDT up counter value has not been cleared after the specific WDT reset delay period expires. 0 = WDT time-out reset function Disabled. 1 = WDT time-out reset function Enabled. Note: This bit is write protected. Refer to the SYS_REGLCTL register.
[0]	Reserved	Reserved.

WDT Alternative Control Register (WDT_ALTCTL)

Register	Offset	R/W	Description	Reset Value
WDT_ALTCTL	WDT_BA+0x04	R/W	WDT Alternative Control Register	0x0000_0000

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

Bits	Description	
[31:2]	Reserved	Reserved.
[1:0]	RSTDSEL	WDT Reset Delay Selection (Write Protect) When WDT time-out happened, user has a time named WDT Reset Delay Period to clear WDT counter by writing 0x00005aa5 to RSTCNT (WDT_RSTCNT[31:0]) to prevent WDT time-out reset happened. User can select a suitable setting of RSTDSEL for different WDT Reset Delay Period. 00 = WDT Reset Delay Period is 1026 * WDT_CLK. 01 = WDT Reset Delay Period is 130 * WDT_CLK. 10 = WDT Reset Delay Period is 18 * WDT_CLK. 11 = WDT Reset Delay Period is 3 * WDT_CLK. Note1: This bit is write protected. Refer to the SYS_REGLCTL register. Note2: This register will be reset to 0 if WDT time-out reset happened.

WDT Reset Counter Register (WDT_RSTCNT)

Register	Offset	R/W	Description	Reset Value
WDT_RSTCNT	WDT_BA+0x08	W	WDT Reset Counter Register	0x0000_0000

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

Bits	Description
[31:0]	WDT Reset Counter Register Writing 0x00005AA5 to this field will reset the internal 18-bit WDT up counter value to 0. Note1: Performing RSTCNT to reset counter needs 2 * WDT_CLK period to become active.

6.9 Window Watchdog Timer (WWDT)

6.9.1 Overview

The Window Watchdog Timer (WWDT) is used to perform a system reset within a specified window period to prevent software running to uncontrollable status by any unpredictable condition.

6.9.2 Features

- 6-bit down counter value (CNTDAT, WWDT_CNT[5:0]) and 6-bit compare value (CMPDAT, WWDT_CTL[21:16]) to make the WWDT time-out window period flexible
- Supports 4-bit value (PSCSEL, WWDT_CTL[11:8]) to programmable maximum 11-bit prescale counter period of WWDT counter
- WWDT counter suspends in Idle/Power-down mode

6.9.3 Block Diagram

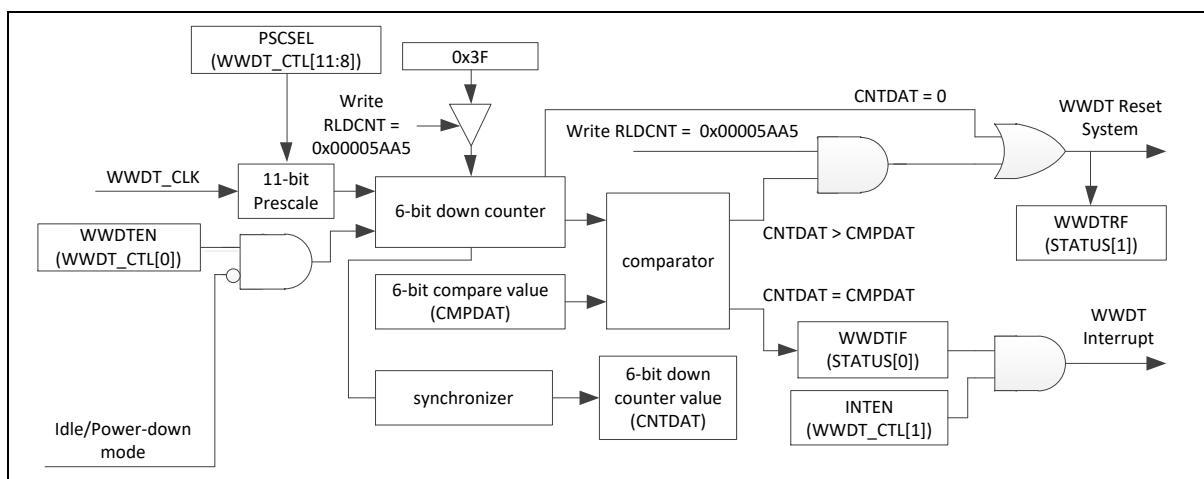


Figure 6.9-1 WWDT Block Diagram

6.9.4 Basic Configuration

- Clock Source Configuration
 - Select the source of WWDT peripheral clock on WWDTSEL (CLK_CLKSEL1[31:30])
 - Enable WWDT peripheral clock in WDTCKEN (CLK_APBCLK0[0]).

The WWDT clock control is shown in Figure 6.9-2.

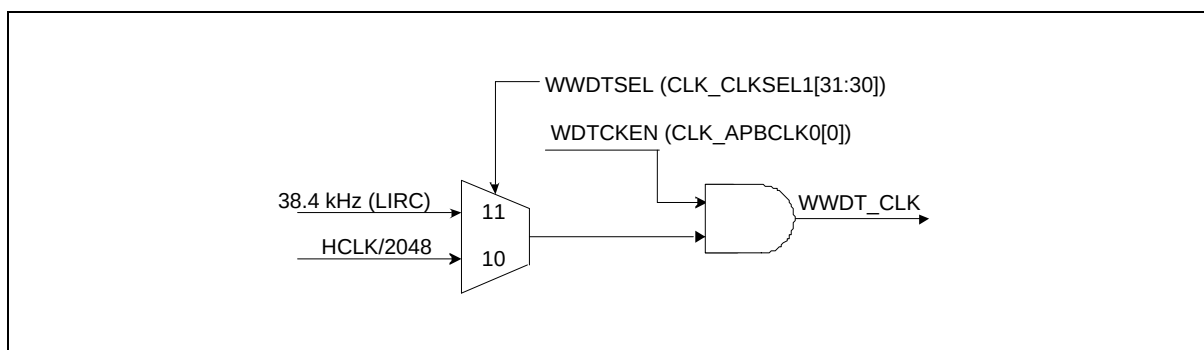


Figure 6.9-2 WWDT Clock Control

6.9.5 Functional Description

The WWDT includes a 6-bit down counter with programmable prescale value to define different WWDT time-out intervals. The clock source of 6-bit WWDT is based on system clock divide 2048 (HCLK/2048) or 38.4 kHz internal low speed RC oscillator (LIRC) with a programmable 11-bit prescale counter value which controlled by PSCSEL (WWDT_CTL[11:8]). Also, the correlate of PSCSEL (WWDT_CTL[11:8]) and prescale value are listed in Table 6.9-1.

PSCSEL	Prescaler Value	Max. Time-Out Period	Max. Time-Out Interval (WWDT_CLK=38.4 kHz)
0000	1	$1 * 64 * T_{WWDT}$	1.667 ms
0001	2	$2 * 64 * T_{WWDT}$	3.333 ms
0010	4	$4 * 64 * T_{WWDT}$	6.667 ms
0011	8	$8 * 64 * T_{WWDT}$	13.333 ms
0100	16	$16 * 64 * T_{WWDT}$	26.667 ms
0101	32	$32 * 64 * T_{WWDT}$	53.333 ms
0110	64	$64 * 64 * T_{WWDT}$	106.667 ms
0111	128	$128 * 64 * T_{WWDT}$	213.333 ms
1000	192	$192 * 64 * T_{WWDT}$	320 ms
1001	256	$256 * 64 * T_{WWDT}$	426.667 ms
1010	384	$384 * 64 * T_{WWDT}$	640 ms
1011	512	$512 * 64 * T_{WWDT}$	853.333 ms
1100	768	$768 * 64 * T_{WWDT}$	1.28 s
1101	1024	$1024 * 64 * T_{WWDT}$	1.701 s
1110	1536	$1536 * 64 * T_{WWDT}$	2.56 s
1111	2048	$2048 * 64 * T_{WWDT}$	3.413 s

Table 6.9-1 WWDT Prescaler Value Selection

6.9.5.1 WWDT Counting

When the WWDTEN (WWDT_CTL[0]) is set, WWDT down counter will start counting from 0x3F to 0. To prevent program runs to disable WWDT counter counting unexpected, the WWDT_CTL register can only be written once after chip is powered on or reset. User cannot disable WWDT counter counting (WWDTEN), change counter prescale period (PSCSEL) or change window compare value (CMPDAT) while WWDTEN (WWDT_CTL[0]) has been enabled by user unless chip is reset.

To avoid the system is reset while CPU clock is disabled, the WWDT counter will stop counting when CPU enters Idle/Power-down mode. After CPU enters normal mode, the WWDT counter will start down counting.

6.9.5.2 WWDT Compare Match Interrupt

During down counting by the WWDT counter, the WWDTIF (WWDT_STATUS[0]) is set to 1 while the WWDT counter value (CNTDAT) is equal to window compare value (CMPDAT) and WWDTIF can be cleared by user; if INTEN (WWDT_CTL[1]) is also set to 1 by user, the WWDT compare match interrupt signal is generated also while WWDTIF is set to 1 by hardware.

6.9.5.3 WWDT Reset System

Figure 6.9-3 shows three cases of WWDT reset and reload behavior.

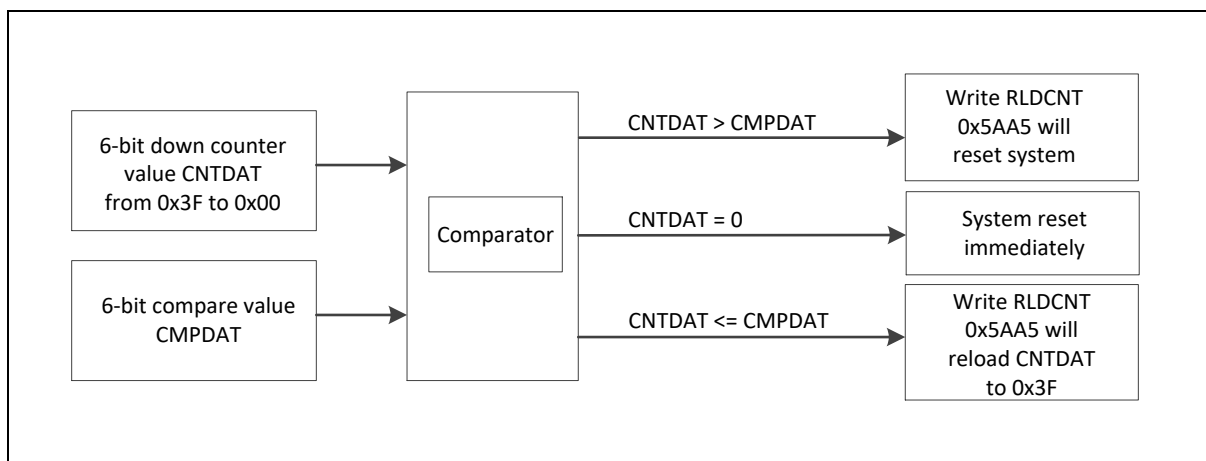


Figure 6.9-3 WWDT Reset and Reload Behavior

If the current CNTDAT (WWDT_CNT[5:0]) is larger than CMPDAT (WWDT_CTL[21:16]) and user writes 0x00005AA5 to the WWDT_RLDCNT register, the WWDT reset system signal will be generated immediately to cause chip reset also. The waveform of WWDT reload counter when CNTDAT > CMPDAT is shown in Figure 6.9-4.

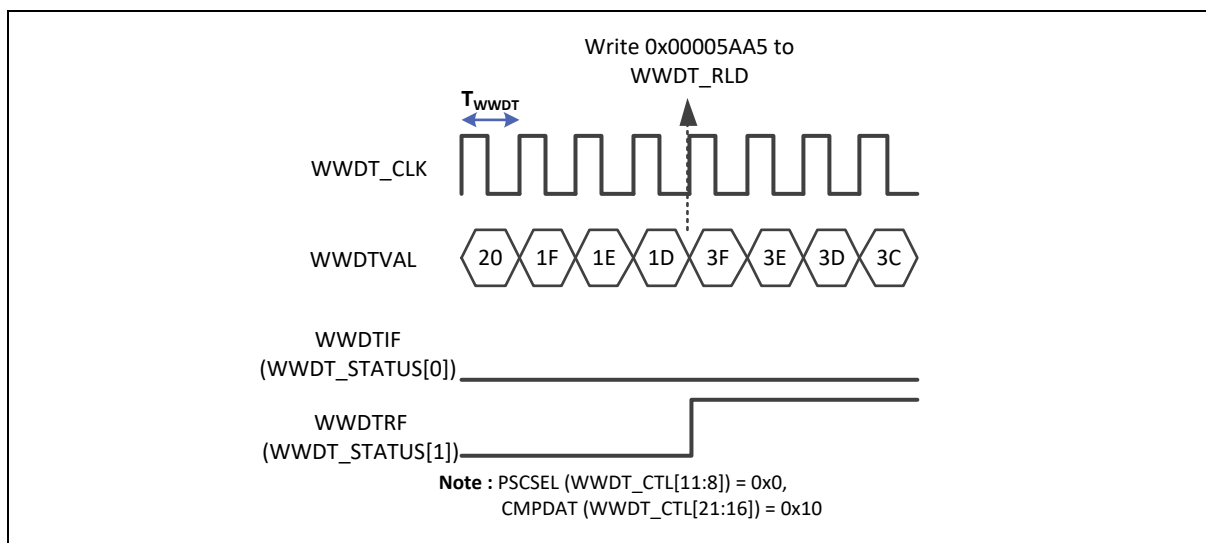


Figure 6.9-4 WWDT Reload Counter When CNTDAT > CMPDAT

When WWDTIF (WWDT_STATUS[0]) is generated, user must reload WWDT counter value to 0x3F by writing 0x00005AA5 to WWDT_RLDCNT register, and also to prevent WWDT counter value reached to 0 and generate WWDT reset system signal to inform system reset. Figure 6.9-5 shows the waveform of WWDT reload counter when CNTDAT < CMPDAT and Figure 6.9-6 shows WWDT generates reset system signal (WWDTTRF) if user doesn't write 0x00005AA5 to WWDT_RLD before WWDT counter value reaches 0.

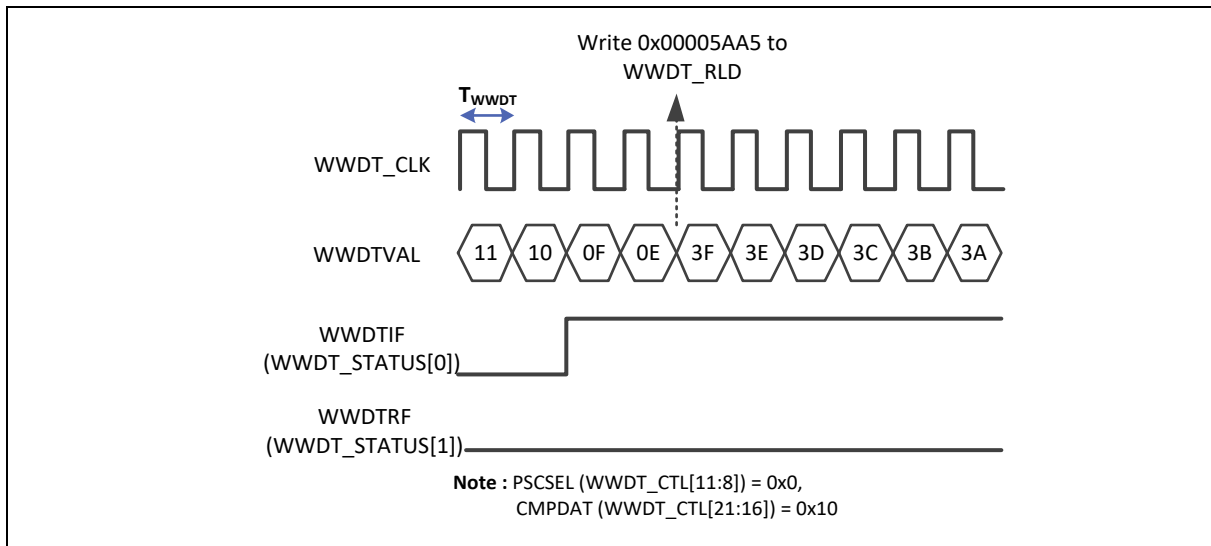


Figure 6.9-5 WWDT Reload Counter When WWDT_CNT < WINCMP

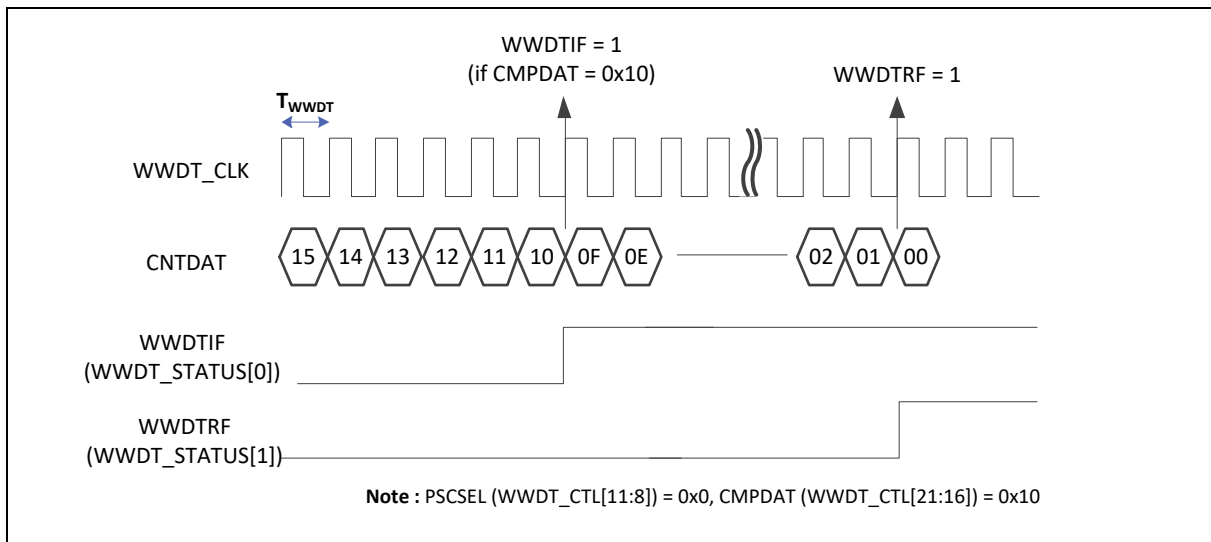


Figure 6.9-6 WWDT Interrupt and Reset Signals

6.9.5.4 WWDT Window Setting Limitation

When user writes 0x00005AA5 to WWDT_RLDCNT register to reload WWDT counter value to 0x3F, it needs 3 WWDT clocks to sync the reload command to actually perform reload action. Note that if user sets PSCSEL (WWDT_CTL[11:8]) to 0000, the counter prescale value should be as 1, and the CMPDAT (WWDT_CTL[21:16]) must be larger than 2. Otherwise, writing WWDT_RLDCNT register to reload WWDT counter value to 0x3F is unavailable, WWDTIF (WWDT_STATUS[0]) is generated, and WWDT reset system event always happened. The WWDT CMPDAT setting limitation is shown in Table 6.9-2.

If user sets CMPDATA as 0x3F and 0x0, the interrupt doesn't occur. The reset occurs when WWDT counts to 0x0, so the interrupt doesn't occur when CMPDATA is 0x0.

PSCSEL	Prescale Value	Valid CMPDAT Value
0000	1	0x3 ~ 0x3E
0001	2	0x2 ~ 0x3E
Others	Others	0x1 ~ 0x3E

Table 6.9-2 CMPDAT Setting Limitation

6.9.5.5 WWDT ICE Debug

When ICE is connected to MCU, the WWDT counter is counting or not by ICEDEBUG (WWDT_CTL[31]). The default value of ICEDEBUG is 0. The WWDT counter will stop counting when CPU is held by ICE. If ICEDEBUG is set to 1, WWDT counter will keep counting no matter CPU is held by ICE or not.

6.9.6 Register Map

R: read only, W: write only, R/W: both read and write

Register	Offset	R/W	Description	Reset Value
WWDT Base Address: WWDT_BA = 0x4004_0100				
WWDT_RLDCNT	WWDT_BA+0x00	W	WWDT Reload Counter Register	0x0000_0000
WWDT_CTL	WWDT_BA+0x04	R/W	WWDT Control Register	0x003F_0800
WWDT_STATUS	WWDT_BA+0x08	R/W	WWDT Status Register	0x0000_0000
WWDT_CNT	WWDT_BA+0x0C	R	WWDT Counter Value Register	0x0000_003F

6.9.7 Register Description

WWDT Reload Counter Register (WWDT_RLDCNT)

Register	Offset	R/W	Description	Reset Value
WWDT_RLDCNT	WWDT_BA+0x00	W	WWDT Reload Counter Register	0x0000_0000

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

Bits	Description
[31:0]	RLDCNT WWDT Reload Counter Register Writing 0x00005AA5 to this register will reload the WWDT counter value to 0x3F. Note: User can only write WWDT_RLDCNT register to reload WWDT counter value when current WWDT counter value between 0 and CMPDAT (WWDT_CTL[21:16]). If user writes WWDT_RLDCNT when current WWDT counter value is larger than CMPDAT, WWDT reset signal will be generated immediately.

WWDT Control Register (WWDT_CTL)

Register	Offset	R/W	Description	Reset Value
WWDT_CTL	WWDT_BA+0x04	R/W	WWDT Control Register	0x003F_0800

Note: This register can be written only one time after chip is powered on or reset.

31	30	29	28	27	26	25	24
ICEDEBUG	Reserved						
23	22	21	20	19	18	17	16
Reserved		CMPDAT					
15	14	13	12	11	10	9	8
Reserved				PSCSEL			
7	6	5	4	3	2	1	0
Reserved						INTEN	WWDTEN

Bits	Description
[31]	ICEDEBUG ICE Debug Mode Acknowledge Disable Bit 0 = ICE debug mode acknowledgement effects WWDT counting. WWDT down counter will be held while CPU is held by ICE. 1 = ICE debug mode acknowledgement Disabled. Note: WWDT down counter will keep going no matter CPU is held by ICE or not.
[30:22]	Reserved Reserved.
[21:16]	CMPDAT WWDT Window Compare Register Set this register to adjust the valid reload window. Note: User can only write WWDT_RLDCNT register to reload WWDT counter value when current WWDT counter value between 0 and CMPDAT. If user writes WWDT_RLDCNT register when current WWDT counter value larger than CMPDAT, WWDT reset signal will generate immediately.
[15:12]	Reserved Reserved.
[11:8]	PSCSEL WWDT Counter Prescale Period Selection 0000 = Pre-scale is 1; Max time-out period is $1 * 64 * WWDT\_CLK$. 0001 = Pre-scale is 2; Max time-out period is $2 * 64 * WWDT\_CLK$. 0010 = Pre-scale is 4; Max time-out period is $4 * 64 * WWDT\_CLK$. 0011 = Pre-scale is 8; Max time-out period is $8 * 64 * WWDT\_CLK$. 0100 = Pre-scale is 16; Max time-out period is $16 * 64 * WWDT\_CLK$. 0101 = Pre-scale is 32; Max time-out period is $32 * 64 * WWDT\_CLK$. 0110 = Pre-scale is 64; Max time-out period is $64 * 64 * WWDT\_CLK$. 0111 = Pre-scale is 128; Max time-out period is $128 * 64 * WWDT\_CLK$. 1000 = Pre-scale is 192; Max time-out period is $192 * 64 * WWDT\_CLK$. 1001 = Pre-scale is 256; Max time-out period is $256 * 64 * WWDT\_CLK$. 1010 = Pre-scale is 384; Max time-out period is $384 * 64 * WWDT\_CLK$. 1011 = Pre-scale is 512; Max time-out period is $512 * 64 * WWDT\_CLK$. 1100 = Pre-scale is 768; Max time-out period is $768 * 64 * WWDT\_CLK$. 1101 = Pre-scale is 1024; Max time-out period is $1024 * 64 * WWDT\_CLK$.

		1110 = Pre-scale is 1536; Max time-out period is $1536 * 64 * \text{WWDT\_CLK}$. 1111 = Pre-scale is 2048; Max time-out period is $2048 * 64 * \text{WWDT\_CLK}$.
[7:2]	Reserved	Reserved.
[1]	INTEN	WWDT Interrupt Enable Bit If this bit is enabled, the WWDT counter compare match interrupt signal is generated and inform to CPU. 0 = WWDT counter compare match interrupt Disabled. 1 = WWDT counter compare match interrupt Enabled.
[0]	WWDTEN	WWDT Enable Bit 0 = WWDT counter is stopped. 1 = WWDT counter starts counting.

WWDT Status Register (WWDT_STATUS)

Register	Offset	R/W	Description	Reset Value
WWDT_STATUS	WWDT_BA+0x08	R/W	WWDT Status Register	0x0000_0000

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

Bits	Description
[31:2]	Reserved Reserved.
[1]	WWDTRF WWDT Timer-out Reset Flag This bit indicates the system has been reset by WWDT time-out reset or not. 0 = WWDT time-out reset did not occur. 1 = WWDT time-out reset occurred. Note: This bit is cleared by writing 1 to it.
[0]	WWDTIF WWDT Compare Match Interrupt Flag This bit indicates the interrupt flag status of WWDT while WWDT counter value matches CMPDAT (WWDT_CTL[21:16]). 0 = No effect. 1 = WWDT counter value matches CMPDAT. Note: This bit is cleared by writing 1 to it.

WWDT Counter Value Register (WWDT_CNT)

Register	Offset	R/W	Description	Reset Value
WWDT_CNT	WWDT_BA+0x0C	R	WWDT Counter Value Register	0x0000_003F

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

Bits	Description	
[31:6]	Reserved	Reserved.
[5:0]	CNTDAT	WWDT Counter Value CNTDAT will be updated continuously to monitor 6-bit WWDT down counter value.

6.10 Real Time Clock (RTC)

6.10.1 Overview

The Real Time Clock (RTC) controller provides the real time and calendar message. The RTC offers programmable time tick and alarm match interrupts. The data format of time and calendar messages are expressed in BCD format. A digital frequency compensation feature is available to compensate external crystal oscillator frequency accuracy.

6.10.2 Features

- Supports real time counter in RTC_TIME (hour, minute, second) and calendar counter in RTC_CAL (year, month, day) for RTC time and calendar check.
- Supports alarm time (hour, minute, second) and calendar (year, month, day) settings in RTC_TALM and RTC_CALM.
- Supports alarm time (hour, minute, second) and calendar (year, month, day) mask enable in RTC_TAMSK and RTC_CAMSK.
- Selectable 12-hour or 24-hour time scale in RTC_CLKFMT register.
-
- Supports Leap Year indication in RTC_LEAPYEAR register.
- Supports Day of the Week counter in RTC_WEEKDAY register.
- Frequency of RTC clock source compensate by RTC_FREQADJ register.
- All time and calendar message expressed in BCD format.
- Supports periodic RTC Time Tick interrupt with 8 period interval options 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 and 1 second.
- Supports RTC Time Tick and Alarm Match interrupt.
- Supports 1 Hz clock output.
- Supports chip wake-up from Idle or Power-down mode while a RTC interrupt signal is generated.
- Supports Daylight Saving Time software control in RTC_DSTCTL.

6.10.3 Block Diagram

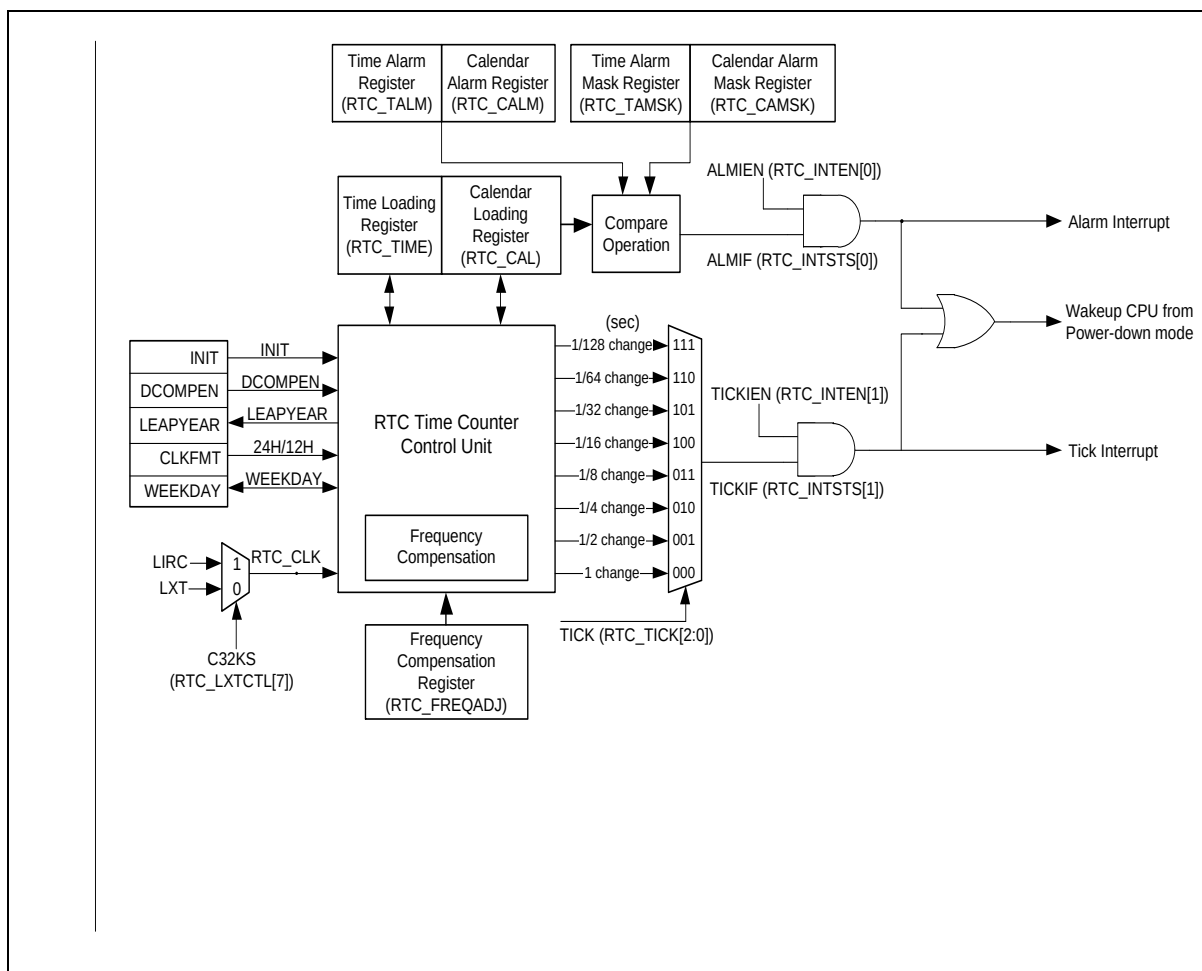


Figure 6.10-1 RTC Block Diagram

6.10.4 Basic Configuration

- Clock Source Configuration
 - The RTC controller clock source is enabled by RTCCKEN (APBCLK0[1]) and RTC Time Counter source is selected by RTC_LXTCTL[7] LXT or LIRC.
- Pin Configuration

Group	Pin Name	GPIO	MFP
X32	X32_OUT	PF.4	MFP10
	X32_IN	PF.5	MFP10

6.10.5 Functional Description

6.10.5.1 RTC Initiation

When a RTC block is powered on, RTC is at reset state. User has to write a number 0xa5eb1357 to RTC initial register INIT(RTC_INIT[31:0]) to make RTC leaving reset state. Once the INIT register is written as 0xa5eb1357, the RTC will be in normal active state permanently. User can read

INIT[0](RTC_INIT[0]) to check the RTC is at normal active state or reset state.

The RTC control registers access attributes are shown in Table 6.10-1.

Register	INIR = 0
RTC_INIT	available
RTC_FREQADJ	available
RTC_TIME	Not available
RTC_CAL	Not available
RTC_CLKFMT	Not available
RTC_WEEKDAY	Not available
RTC_TALM	Not available
RTC_CALM	Not available
RTC_LEAPYEAR	Not available
RTC_INTEN	available
RTC_INTSTS	available
RTC_TICK	Not available
RTC_TAMSK	Not available
RTC_CAMSK	Not available
RTC_LXTCTL	available
RTC_DSTCTL	available

Table 6.10-1 RTC Read/Write Enable

6.10.5.2 Frequency Compensation

Frequency compensation circuit supports dynamic compensation to adjust compensation value without reset the compensation circuit. The enable bit for dynamic compensation is DCOMPEN(RTC_CLKFMT[16]).

If enable dynamic compensation, the minimal interval to continuous writing RTC_FREQADJ is 0.24 seconds, and FCRBUSY(RTC_FREQADJ[31]) flag is used to indicate that new register write operation is prohibited. And the compensate value will load into compensation circuit after 0.24~1.16 seconds when write to RTC_FREQADJ.

The RTC_FREQADJ register allows user to make digital compensation to a clock input. Please follow the example and formula below to write the actual frequency of 32k crystal to RTC_FREQADJ register. Following are the compensation examples for higher or lower than 32768 Hz.

Example 1:

Frequency counter measurement : 32773.65 Hz

Integer Part: 32773 => RTC_FREQADJ[12:8] = 0x15, Refer the INTEGER(RTC_FREQADJ[13:8]) to get detailed setting value.

Fraction Part: 0.65 X 64 = 41.6(0x2A) => RTC_FREQADJ[5:0]=0x2A

Example 2:

Frequency counter measurement : 32763.25 Hz

Integer part: 32763=> RTC_FREQADJ[12:8] = 0x0B, Refer the INTEGER(RTC_FREQADJ[13:8]) to get detailed setting value.

Fraction part: $0.25 \times 64 = 16(0x10) \Rightarrow \text{RTC\_FREQADJ}[5:0] = 0x10$

Note: The value of RTC_FREQADJ register will be the default value (0x0000_1000) while the compensation is not executed. User can utilize a frequency counter to measure RTC clock source via clock output function in manufacturing. In the meanwhile, user can use clock output function to check the result of RTC frequency compensation.

6.10.5.3 Time and Calendar Counter

RTC_TIME and RTC_CAL are used to load the real time and calendar. RTC_TALM and RTC_CALM are used for setup alarm time and calendar.

6.10.5.4 12/24 hour Time Scale Selection

The 12/24 hour time scale selection depends on 24HEN (RTC_CLKFMT[0]). When RTC runs as 12-hour time scale mode, RTC_TIME[21] (the high bit of TENHR[1:0]) means AM/PM indication, if RTC_TIME[21] is 1, it indicates PM time message and RTC_TIME[21] is 0 indicates AM time message.) Table 6.10-2 shows RTC_TIME mapping table of 12/24 hour time scale selection.

Note: The Hour Value Write Into RTC_TIME[21:16], Messages Are Expressed In BCD Format.			
24-Hour Time Scale (24HEN = 1)		12-Hour Time Scale (PM Time + 0x20) (24HEN = 0) (PM Time + 0x20)	
0x00 (AM12)	0x12 (PM12)	0x12 (AM12)	0x32 (PM12)
0x01 (AM01)	0x13 (PM01)	0x01 (AM01)	0x21 (PM01)
0x02 (AM02)	0x14 (PM02)	0x02 (AM02)	0x22 (PM02)
0x03 (AM03)	0x15 (PM03)	0x03 (AM03)	0x23 (PM03)
0x04 (AM04)	0x16 (PM04)	0x04 (AM04)	0x24 (PM04)
0x05 (AM05)	0x17 (PM05)	0x05 (AM05)	0x25 (PM05)
0x06 (AM06)	0x18 (PM06)	0x06 (AM06)	0x26 (PM06)
0x07 (AM07)	0x19 (PM07)	0x07 (AM07)	0x27 (PM07)
0x08 (AM08)	0x20 (PM08)	0x08 (AM08)	0x28 (PM08)
0x09 (AM09)	0x21 (PM09)	0x09 (AM09)	0x29 (PM09)
0x10 (AM10)	0x22 (PM10)	0x10 (AM10)	0x30 (PM10)
0x11 (AM11)	0x23 (PM11)	0x11 (AM11)	0x31 (PM11)

Table 6.10-212/24 Hour Time Scale Selection

6.10.5.5 Day of the Week Counter

The RTC controller provides day of week in WEEKDAY bits (RTC_WEEKDAY[2:0]). The value is defined from 0 to 6 to represent Sunday to Saturday respectively.

6.10.5.6 Periodic Time Tick Interrupt

The periodic time tick interrupt has 8 period interval options 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 and 1 second that are selected by TICK bits (RTC_TICK[2:0]). When Periodic Time Tick interrupt is enabled

by setting TICKIEN (RTC_INTEN[1]) to 1, the Periodic Time Tick interrupt is requested periodically in the period selected by RTC_TICK[2:0] settings.

6.10.5.7 Alarm Interrupt

When the real time and calendar message in RTC_TIME and RTC_CAL registers are equal to alarm time and calendar values in RTC_TALM and RTC_CALM registers, the RTC alarm interrupt flag ALMIF (RTC_INTSTS[0]) is set to 1 and the RTC alarm interrupt signal assert if the alarm interrupt enable ALMIEN (RTC_INTEN[0]) is enabled.

The RTC controller provides time alarm mask register (RTC_TAMSK register) and Calendar Alarm Mask Register (RTC_CAMSK register) to mask the specified digit and generate periodic interrupt without changing the alarm match condition in RTC_TALM and RTC_CALM registers in each alarm interrupt service routine.

6.10.5.8 Daylight Saving Time

The RTC controller also provides RTC_DSTCTL register to store the control settings of daylight saving time application. User can read DSBK(RTC_DSTCTL[2]) value to check current RTC date/time counter runs in daylight saving time mode or normal mode.

6.10.5.9 1 Hz Clock Output

The RTC controller provides 1Hz clock output to CLKO function pin. User can set CLK1HZEN (CLK_CLKOCTL[6]) to 1 and enable RTC, 1Hz clock will output to CLKO function pin.

6.10.5.10 Application Note

1. All data in RTC_TALM, RTC_CALM, RTC_TIME and RTC_CAL registers are all expressed in BCD format.
2. User has to make sure that the loaded values are reasonable. For example, Load RTC_CAL as 201a (year), 13 (month), 00 (day), or RTC_CAL does not match with RTC_WEEKDAY, etc.
3. In RTC_CAL and RTC_CALM, only 2 BCD digits are used to express "year". The 2 BCD digits of xy means 20xy, rather than 19xy or 21xy.
4. Example of 12-Hour Time Setting
If current RTC time is PM12:59:30 in 12-Hour Time Scale mode, the RTC_TIME setting as:
5. HOUR:
RTC_TIME[21:16]: 0x32 (0x12+0x20) combined by TENHR (RTC_TIME[21:20]) is 0x3, HR (RTC_TIME[19:16]) is 0x2.
6. MIN:
RTC_TIME[14:8]: 0x59 combined by TENMIN (RTC_TIME[14:12]) is 0x5, MIN (RTC_TIME[11:8]) is 0x9.
7. SEC:
RTC_TIME[6:0]: 0x30 combined by TENSEC (RTC_TIME[6:4]) is 0x3, SEC (RTC_TIME[3:0]) is 0x0.
8. Table 6.10-3 shows registers value after first powered on.

Register	Reset State
RTC_INIT	0
RTC_CAL	15/8/8 (year/month/day)
RTC_TIME	00:00:00 (hour : minute : second)
RTC_CALM	00/00/00 (year/month/day)
RTC_TALM	00:00:00 (hour : minute : second)
RTC_CLKFMT	1 (24-hour mode)
RTC_WEEKDAY	6 (Saturday)
RTC_INTEN	0
RTC_INTSTS	0
RTC_LEAPYEAR	0
RTC_TICK	0
RTC_DSTCTL	0

Table 6.10-3 Registers Value after Powered On

6.10.6 Register Map

R: read only, W: write only, R/W: both read and write

Register	Offset	R/W	Description	Reset Value
RTC Base Address: RTC_BA = 0x4004_1000				
RTC_INIT	RTC_BA+0x00	R/W	RTC Initiation Register	0x0000_0000
RTC_FREQADJ	RTC_BA+0x08	R/W	RTC Frequency Compensation Register	0x0000_1000
RTC_TIME	RTC_BA+0x0C	R/W	RTC Time Loading Register	0x0000_0000
RTC_CAL	RTC_BA+0x10	R/W	RTC Calendar Loading Register	0x0015_0808
RTC_CLKFMT	RTC_BA+0x14	R/W	RTC Time Scale Selection Register	0x0000_0001
RTC_WEEKDAY	RTC_BA+0x18	R/W	RTC Day of the Week Register	0x0000_0006
RTC_TALM	RTC_BA+0x1C	R/W	RTC Time Alarm Register	0x0000_0000
RTC_CALM	RTC_BA+0x20	R/W	RTC Calendar Alarm Register	0x0000_0000
RTC_LEAPYEAR	RTC_BA+0x24	R	RTC Leap Year Indicator Register	0x0000_0000
RTC_INTEN	RTC_BA+0x28	R/W	RTC Interrupt Enable Register	0x0000_0000
RTC_INTSTS	RTC_BA+0x2C	R/W	RTC Interrupt Status Register	0x0000_0000
RTC_TICK	RTC_BA+0x30	R/W	RTC Time Tick Register	0x0000_0000
RTC_TAMSK	RTC_BA+0x34	R/W	RTC Time Alarm Mask Register	0x0000_0000
RTC_CAMSK	RTC_BA+0x38	R/W	RTC Calendar Alarm Mask Register	0x0000_0000
RTC_LXTCTL	RTC_BA+0x100	R/W	RTC 32.768 kHz Oscillator Control Register	0x0000_0006
RTC_DSTCTL	RTC_BA+0x110	R/W	RTC Daylight Saving Time Control Register	0x0000_0000

6.10.7 Register Description

RTC Initiation Register (RTC_INIT)

Register	Offset	R/W	Description	Reset Value
RTC_INIT	RTC_BA+0x00	R/W	RTC Initiation Register	0x0000_0000

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

Bits	Description
[31:1]	INIT RTC Initiation (Write Only) When RTC block is powered on, RTC is at reset state. User has to write a number (0x a5eb1357) to INIT to make RTC leave reset state. Once the INIT is written as 0xa5eb1357, the RTC will be in un-reset state permanently. The INIT is a write-only field and read value will be always 0.
[0]	ACTIVE RTC Active Status (Read Only) 0 = RTC is at reset state. 1 = RTC is at normal active state.

RTC Frequency Compensation Register (RTC_FREQADJ)

Register	Offset	R/W	Description	Reset Value
RTC_FREQADJ	RTC_BA+0x08	R/W	RTC Frequency Compensation Register	0x0000_1000

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

Bits	Description
[31]	FCRBUSY Frequency Compensation Register Write Operation Busy (Read Only) 0 = The new register write operation is acceptable. 1 = The last write operation is in progress and new register write operation prohibited. Note: This bit is only used when DCOMPEN(RTC_CLKFMT[16]) enabled.
[30:13]	Reserved Reserved.
[12:8]	INTEGER Integer Part 00000 = Integer part of detected value is 32752. 00001 = Integer part of detected value is 32753. 00010 = Integer part of detected value is 32754. 00011 = Integer part of detected value is 32755. 00100 = Integer part of detected value is 32756. 00101 = Integer part of detected value is 32757. 00110 = Integer part of detected value is 32758. 00111 = Integer part of detected value is 32759. 01000 = Integer part of detected value is 32760. 01001 = Integer part of detected value is 32761. 01010 = Integer part of detected value is 32762. 01011 = Integer part of detected value is 32763. 01100 = Integer part of detected value is 32764. 01101 = Integer part of detected value is 32765. 01110 = Integer part of detected value is 32766. 01111 = Integer part of detected value is 32767. 10000 = Integer part of detected value is 32768. 10001 = Integer part of detected value is 32769. 10010 = Integer part of detected value is 32770. 10011 = Integer part of detected value is 32771. 10100 = Integer part of detected value is 32772. 10101 = Integer part of detected value is 32773.

		10110 = Integer part of detected value is 32774. 10111 = Integer part of detected value is 32775. 11000 = Integer part of detected value is 32776. 11001 = Integer part of detected value is 32777. 11010 = Integer part of detected value is 32778. 11011 = Integer part of detected value is 32779. 11100 = Integer part of detected value is 32780. 11101 = Integer part of detected value is 32781. 11110 = Integer part of detected value is 32782. 11111 = Integer part of detected value is 32783.
[7:6]	Reserved	Reserved.
[5:0]	FRACTION	Fraction Part Formula: FRACTION = (fraction part of detected value) X 64. Note: Digit in FCR must be expressed as hexadecimal number.

Note: If dynamic compensation is not enable, i.e. DCOMPEN(RTC_CLKFMT[16]) = 0, FREQADJ's counter will be reset for start to compensate when writing RTC_FREQADJ. Imply RTC Time will be restarted.

RTC Time Loading Register (RTC_TIME)

Register	Offset	R/W	Description	Reset Value
RTC_TIME	RTC_BA+0x0C	R/W	RTC Time Loading Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved		TENHR		HR			
15	14	13	12	11	10	9	8
Reserved	TENMIN			MIN			
7	6	5	4	3	2	1	0
Reserved	TENSEC			SEC			

Bits	Description
[31:22]	Reserved Reserved.
[21:20]	TENHR 10-Hour Time Digit (0~2) When RTC runs as 12-hour time scale mode, RTC_TIME[21] (the high bit of TENHR[1:0]) means AM/PM indication (If RTC_TIME[21] is 1, it indicates PM time message.)
[19:16]	HR 1-Hour Time Digit (0~9)
[15]	Reserved Reserved.
[14:12]	TENMIN 10-Min Time Digit (0~5)
[11:8]	MIN 1-Min Time Digit (0~9)
[7]	Reserved Reserved.
[6:4]	TENSEC 10-Sec Time Digit (0~5)
[3:0]	SEC 1-Sec Time Digit (0~9)

Note:

1. RTC_TIME is a BCD digit counter and RTC will not check loaded data.
2. The reasonable value range is listed in the parenthesis.
3. FREQADJ's counter will be reset for start to Compensate when writing RTC_FREQADJ, RTC_TIME, RTC_CAL, RTC_WEEKDAY. Imply RTC Time will be restarted.

RTC Calendar Loading Register (RTC_CAL)

Register	Offset	R/W	Description	Reset Value
RTC_CAL	RTC_BA+0x10	R/W	RTC Calendar Loading Register	0x0015_0808

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
TENYEAR				YEAR			
15	14	13	12	11	10	9	8
Reserved			TENMON	MON			
7	6	5	4	3	2	1	0
Reserved		TENDAY		DAY			

Bits	Description	
[31:24]	Reserved	Reserved.
[23:20]	TENYEAR	10-Year Calendar Digit (0~9)
[19:16]	YEAR	1-Year Calendar Digit (0~9)
[15:13]	Reserved	Reserved.
[12]	TENMON	10-Month Calendar Digit (0~1)
[11:8]	MON	1-Month Calendar Digit (0~9)
[7:6]	Reserved	Reserved.
[5:4]	TENDAY	10-Day Calendar Digit (0~3)
[3:0]	DAY	1-Day Calendar Digit (0~9)

Note:

1. RTC_CAL is a BCD digit counter and RTC will not check loaded data.
2. The reasonable value range is listed in the parenthesis.
3. FREQADJ's counter will be reset for start to Compensate when writing RTC_FREQADJ , RTC_TIME, RTC_CAL, RTC_WEEKDAY. Imply RTC Time will be restarted.

RTC Time Scale Selection Register (RTC_CLKFMT)

Register	Offset	R/W	Description	Reset Value
RTC_CLKFMT	RTC_BA+0x14	R/W	RTC Time Scale Selection Register	0x0000_0001

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

Bits	Description
[31:17]	Reserved Reserved.
[16]	DCOMPEN Dynamic Compensation Enable Bit 0 = Dynamic Compensation Disabled. 1 = Dynamic Compensation Enabled.
[15:1]	Reserved Reserved.
[0]	24HEN 24-hour / 12-hour Time Scale Selection Indicates that RTC_TIME and RTC_TALM are in 24-hour time scale or 12-hour time scale 0 = 12-hour time scale with AM and PM indication selected. 1 = 24-hour time scale selected.

RTC Day of the Week Register (RTC_WEEKDAY)

Register	Offset	R/W	Description	Reset Value
RTC_WEEKDAY	RTC_BA+0x18	R/W	RTC Day of the Week Register	0x0000_0006

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

Bits	Description	
[31:3]	Reserved	Reserved.
[2:0]	WEEKDAY	Day of the Week Register 000 = Sunday. 001 = Monday. 010 = Tuesday. 011 = Wednesday. 100 = Thursday. 101 = Friday. 110 = Saturday. 111 = Reserved.

Note: FREQADJ's counter will be reset for start to Compensate when writing RTC_FREQADJ, RTC_TIME, RTC_CAL, RTC_WEEKDAY. Imply RTC Time will be restarted.

RTC Time Alarm Register (RTC_TALM)

Register	Offset	R/W	Description	Reset Value
RTC_TALM	RTC_BA+0x1C	R/W	RTC Time Alarm Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved		TENHR		HR			
15	14	13	12	11	10	9	8
Reserved	TENMIN			MIN			
7	6	5	4	3	2	1	0
Reserved	TENSEC			SEC			

Bits	Description
[31:22]	Reserved Reserved.
[21:20]	TENHR 10-Hour Time Digit of Alarm Setting (0~2) When RTC runs as 12-hour time scale mode, RTC_TIME[21] (the high bit of TENHR[1:0]) means AM/PM indication (If RTC_TIME[21] is 1, it indicates PM time message.)
[19:16]	HR 1-Hour Time Digit of Alarm Setting (0~9)
[15]	Reserved Reserved.
[14:12]	TENMIN 10-Min Time Digit of Alarm Setting (0~5)
[11:8]	MIN 1-Min Time Digit of Alarm Setting (0~9)
[7]	Reserved Reserved.
[6:4]	TENSEC 10-Sec Time Digit of Alarm Setting (0~5)
[3:0]	SEC 1-Sec Time Digit of Alarm Setting (0~9)

Note:

1. RTC_TALM is a BCD digit counter.
2. The reasonable value range is listed in the parenthesis.

RTC Calendar Alarm Register (RTC_CALM)

Register	Offset	R/W	Description	Reset Value
RTC_CALM	RTC_BA+0x20	R/W	RTC Calendar Alarm Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
TENYEAR				YEAR			
15	14	13	12	11	10	9	8
Reserved			TENMON	MON			
7	6	5	4	3	2	1	0
Reserved		TENDAY		DAY			

Bits	Description	
[31:24]	Reserved	Reserved.
[23:20]	TENYEAR	10-Year Calendar Digit of Alarm Setting (0~9)
[19:16]	YEAR	1-Year Calendar Digit of Alarm Setting (0~9)
[15:13]	Reserved	Reserved.
[12]	TENMON	10-Month Calendar Digit of Alarm Setting (0~1)
[11:8]	MON	1-Month Calendar Digit of Alarm Setting (0~9)
[7:6]	Reserved	Reserved.
[5:4]	TENDAY	10-Day Calendar Digit of Alarm Setting (0~3)
[3:0]	DAY	1-Day Calendar Digit of Alarm Setting (0~9)

Note:

1. RTC_CALM is a BCD digit counter.
2. The reasonable value range is listed in the parenthesis.

RTC Leap Year Indication Register (RTC_LEAPYEAR)

Register	Offset	R/W	Description	Reset Value
RTC_LEAPYEAR	RTC_BA+0x24	R	RTC Leap Year Indicator Register	0x0000_0000

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

Bits	Description	
[31:1]	Reserved	Reserved.
[0]	LEAPYEAR	Leap Year Indication (Read Only) 0 = This year is not a leap year. 1 = This year is leap year.

RTC Interrupt Enable Register (RTC_INTEN)

Register	Offset	R/W	Description	Reset Value
RTC_INTEN	RTC_BA+0x28	R/W	RTC Interrupt Enable Register	0x0000_0000

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

Bits	Description
[31:2]	Reserved Reserved.
[1]	TICKIEN Time Tick Interrupt Enable Bit Set TICKIEN to 1 can also enable chip wake-up function when RTC tick interrupt event is generated. 0 = RTC Time Tick interrupt Disabled. 1 = RTC Time Tick interrupt Enabled.
[0]	ALMIEN Alarm Interrupt Enable Bit Set ALMIEN to 1 can also enable chip wake-up function when RTC alarm interrupt event is generated. 0 = RTC Alarm interrupt Disabled. 1 = RTC Alarm interrupt Enabled.

RTC Interrupt Status Register (RTC_INTSTS)

Register	Offset	R/W	Description	Reset Value
RTC_INTSTS	RTC_BA+0x2C	R/W	RTC Interrupt Status Register	0x0000_0000

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

Bits	Description	
[31:2]	Reserved	Reserved.
[1]	TICKIF	RTC Time Tick Interrupt Flag 0 = Tick condition did not occur. 1 = Tick condition occurred. Note: Write 1 to clear this bit.
[0]	ALMIF	RTC Alarm Interrupt Flag 0 = Alarm condition is not matched. 1 = Alarm condition is matched. Note: Write 1 to clear this bit.

RTC Time Tick Register (RTC_TICK)

Register	Offset	R/W	Description	Reset Value
RTC_TICK	RTC_BA+0x30	R/W	RTC Time Tick Register	0x0000_0000

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

Bits	Description
[31:3]	Reserved Reserved.
[2:0]	TICK Time Tick Register These bits are used to select RTC time tick period for Periodic Time Tick Interrupt request. 000 = Time tick is 1 second. 001 = Time tick is 1/2 second. 010 = Time tick is 1/4 second. 011 = Time tick is 1/8 second. 100 = Time tick is 1/16 second. 101 = Time tick is 1/32 second. 110 = Time tick is 1/64 second. 111 = Time tick is 1/128 second. Note: If C32KS(RTC_LXTCTL[7]) is 1, the tick period will become 1, 1/3, 1/5, 1/10, 1/19, 1/38, 1/75, 1/150 second. But the 1/3, 1/5, 1/10, 1/19, 1/38 are not uniform output.

RTC Time Alarm MASK Register (RTC_TAMSK)

Register	Offset	R/W	Description	Reset Value
RTC_TAMSK	RTC_BA+0x34	R/W	RTC Time Alarm Mask Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		MTENHR	MHR	MTENMIN	MMIN	MTENSEC	MSEC

Bits	Description
[31:6]	Reserved Reserved.
[5]	MTENHR Mask 10-Hour Time Digit of Alarm Setting (0~2)
[4]	MHR Mask 1-Hour Time Digit of Alarm Setting (0~9)
[3]	MTENMIN Mask 10-Min Time Digit of Alarm Setting (0~5)
[2]	MMIN Mask 1-Min Time Digit of Alarm Setting (0~9)
[1]	MTENSEC Mask 10-Sec Time Digit of Alarm Setting (0~5)
[0]	MSEC Mask 1-Sec Time Digit of Alarm Setting (0~9)

Note:

1. RTC_TALM is a BCD digit counter and RTC will not check loaded data.
2. The reasonable value range is listed in the parenthesis.
3. MTENHR/MHR base on 24 hour Time Scale.

RTC Calendar Alarm MASK Register (RTC_CAMSK)

Register	Offset	R/W	Description	Reset Value
RTC_CAMSK	RTC_BA+0x38	R/W	RTC Calendar Alarm Mask Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		MTENYEAR	MYEAR	MTENMON	MMON	MTENDAY	MDAY

Bits	Description
[31:6]	Reserved
[5]	MTENYEAR
[4]	MYEAR
[3]	MTENMON
[2]	MMON
[1]	MTENDAY
[0]	MDAY

Note:

1. RTC_CALM is a BCD digit counter and RTC will not check loaded data.
2. The reasonable value range is listed in the parenthesis.

RTC 32K Oscillator Control Register (RTC_LXTCTL)

Register	Offset	R/W	Description	Reset Value
RTC_LXTCTL	RTC_BA+0x100	R/W	RTC 32.768 kHz Oscillator Control Register	0x0000_0006

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
C32KS	Reserved				GAIN		Reserved

Bits	Description
[31:8]	Reserved Reserved.
[7]	C32KS Clock 32K Source Selection: 0 = Internal 32K clock is from 32K crystal . 1 = Internal 32K clock is from LIRC32K.
[6:3]	Reserved Reserved.
[2:1]	GAIN Oscillator Gain Option User can select oscillator gain according to crystal external loading and operating temperature range. The larger gain value corresponding to stronger driving capability and higher power consumption. 000 = L0 mode. 001 = L1 mode. 010 = L2 mode. 011 = L3 mode Note: Please refer to the M253 Datasheet for detailed information about LXT electrical characteristics.
[0]	Reserved Reserved.

RTC Daylight Saving Time Control Register (RTC_DSTCTL)

Register	Offset	R/W	Description	Reset Value
RTC_DSTCTL	RTC_BA+0x110	R/W	RTC Daylight Saving Time Control Register	0x0000_0000

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

Bits	Description
[31:3]	Reserved Reserved.
[2]	DSBAK Daylight Saving Back 0= Daylight Saving Change is not performed. 1= Daylight Saving Change is performed.
[1]	SUBHR Subtract 1 Hour 0 = No effect. 1 = Indicates RTC hour digit has been subtracted one hour for winter time change.
[0]	ADDHR Add 1 Hour 0 = No effect. 1 = Indicates RTC hour digit has been added one hour for summer time change.

6.11 Basic PWM Generator and Capture Timer (BPWM)

6.11.1 Overview

The chip provides one BPWM generators — BPWM0 as shown in Figure 6.11-1. Each BPWM supports 6 channels of BPWM output or input capture. There is a 12-bit prescaler to support flexible clock to the 16-bit BPWM counter with 16-bit comparator. The BPWM counter supports up, down and up-down counter types, all 6 channels share one counter. BPWM uses the comparator compared with counter to generate events. These events are used to generate BPWM pulse, interrupt and trigger signal for EADC to start conversion. For BPWM output control unit, it supports polarity output, independent pin mask and tri-state output enable.

The BPWM generator also supports input capture function to latch BPWM counter value to corresponding register when input channel has a rising transition, falling transition or both transition is happened.

6.11.2 Features

6.11.2.1 BPWM Function Features

- Supports maximum clock frequency up to maximum PLL frequency.
- Supports up to one BPWM modules; each module provides 6 output channels
- Supports independent mode for BPWM output/Capture input channel
- Supports 12-bit prescaler from 1 to 4096
- Supports 16-bit resolution BPWM counter; each module provides 1 BPWM counter
 - Up, down and up/down counter operation type
- Supports mask function and tri-state enable for each BPWM pin
- Supports interrupt in the following events:
 - BPWM counter matches 0, period value or compared value
- Supports trigger EADC in the following events:
 - BPWM counter matches 0, period value or compared value

6.11.2.2 Capture Function Features

- Supports up to 6 capture input channels with 16-bit resolution
- Supports rising or falling capture condition
- Supports input rising/falling capture interrupt
- Supports rising/falling capture with counter reload option

6.11.3 Block Diagram

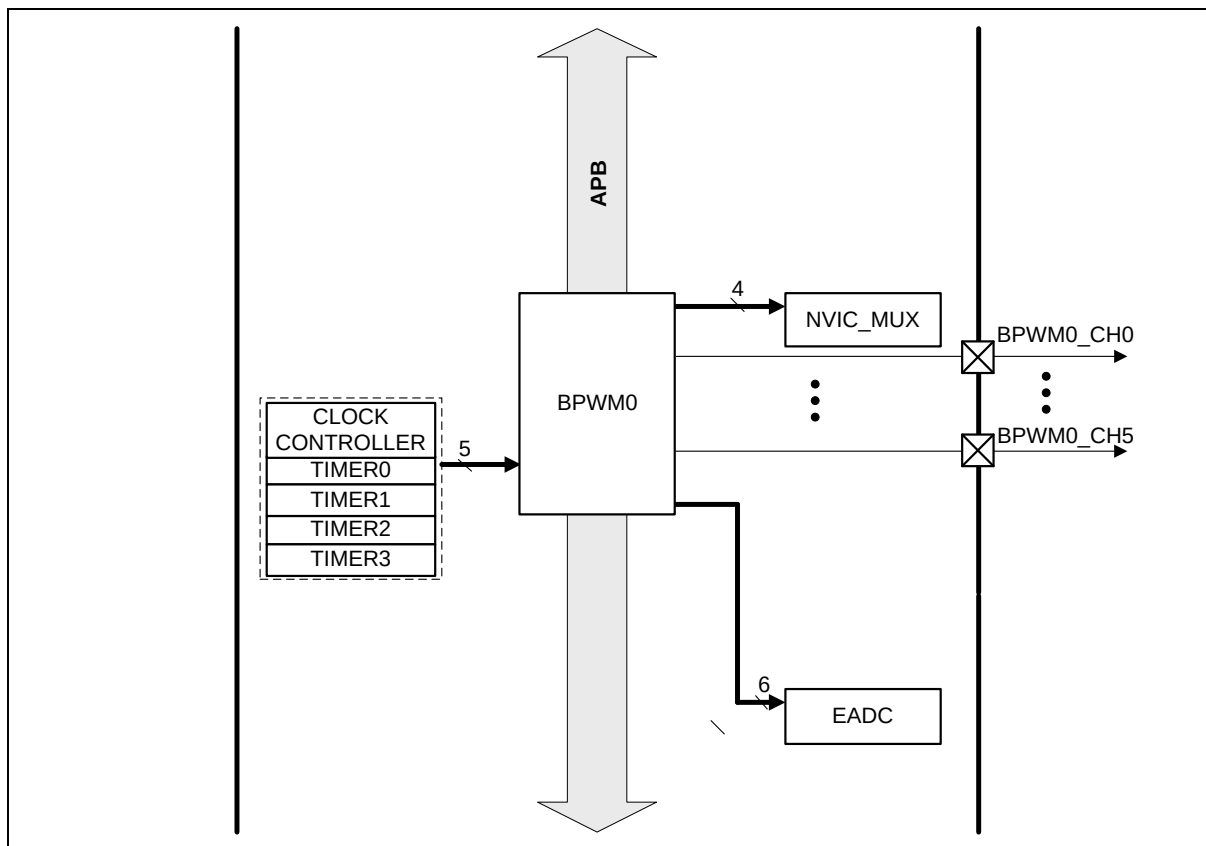


Figure 6.11-1 BPWM Generator Overview Block Diagram

Each BPWM generator has only one clock source inputs and can be selected from BPWM Clock or four TIMER trigger BPWM outputs as shown in Figure 6.11-2 by ECLKSRC0 (BPWM_CLKSRC[2:0]) for BPWM_CLK0. In general case, BPWM0 Clock must be selected from PCLK0 by setting BPWM0SEL (CLK_CLKSEL2[8]) to 1 and BPWM1 Clock from PCLK1 by setting BPWM1SEL (CLK_CLKSEL2[9]) to 1.

When operating in maximum PLL clock frequency as shown in Figure 6.11-3, and Table 6.11-1, BPWM0 and BPWM1 Clock must be selected to PLL clock by setting BPWM0SEL (CLK_CLKSEL2[8]) and BPWM1SEL (CLK_CLKSEL2[9]) to 0.

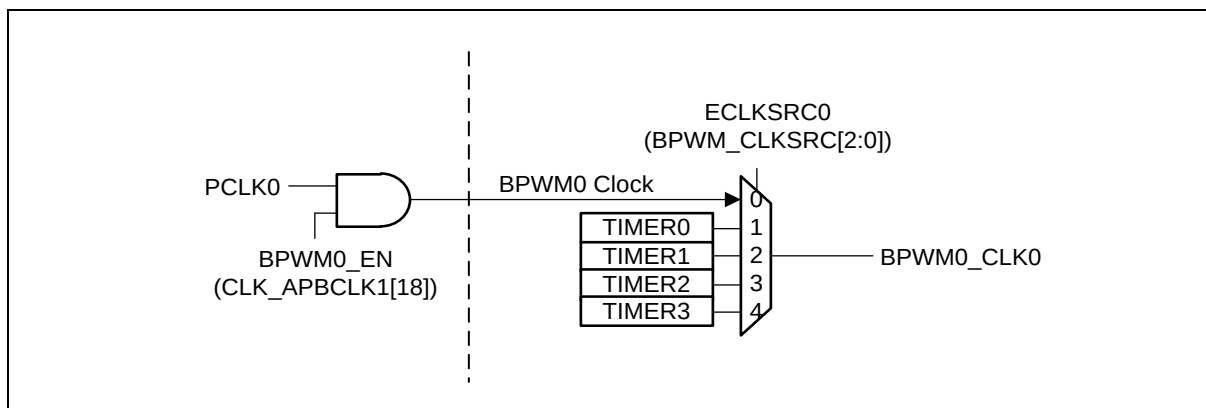


Figure 6.11-2 BPWM Clock Source Control

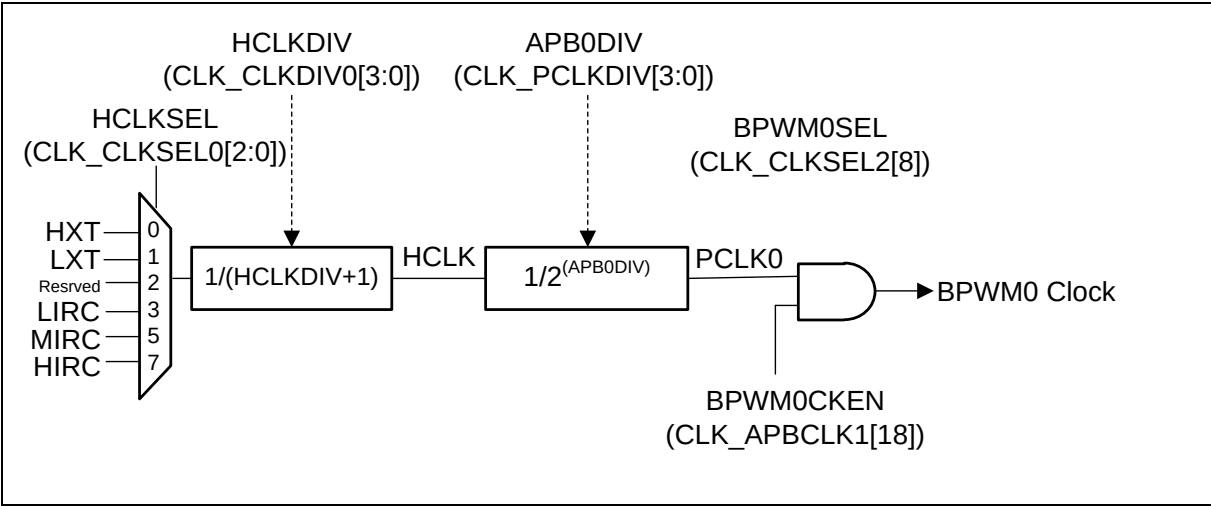


Figure 6.11-3 BPWM Clock Source Control

Frequency Ratio PCLK:BPWM Clock	HCLK	PCLK	BPWM Clock	HCLKSEL CLK_CLKSEL0[2:0]	HCLKDIV CLK_CLKDIV0[3:0]	APBnDIV (CLK_PCLKDIVn [2+4n:4n]), N Denotes 0 Or 1	BPWMnSEL (CLK_CLKSEL2[N+8]), N Denotes 0 Or 1
1:1	HCLK	PCLK	PCLK	Don't care	Don't care	Don't care	1

Table 6.11-1 BPWM Clock Source Control Registers Setting Table

Figure 6.11-4 illustrates the architecture of BPWM Independent mode. All six channels share the same counter. When the counter counts to 0, PERIOD (BPWM_PERIOD[15:0]), or equal to the comparator, events will be generated. These events are passed to the corresponding generators to generate BPWM pulse, interrupt signal and trigger signal for EADC to start conversion. Output control is used to change the BPWM pulse output state.

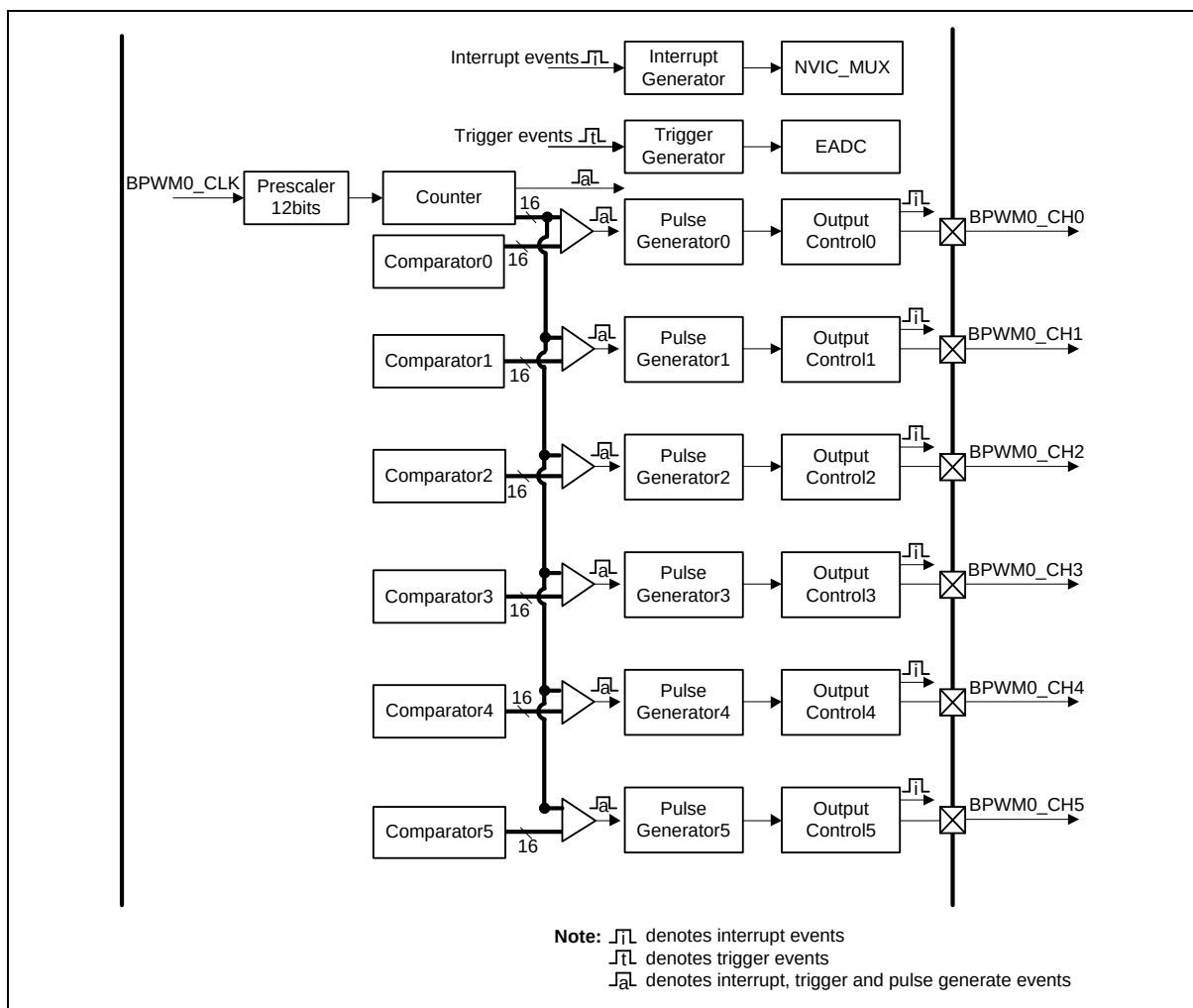


Figure 6.11-4 BPWM Independent Mode Architecture Diagram

6.11.4 Basic Configuration

6.11.4.1 BPWM0 Basic Configuration

- Clock Source Configuration
 - Select the source of BPWM0 peripheral clock on BPWM0SEL (CLK_CLKSEL2[8]).
 - Enable BPWM0 peripheral clock in BPWM0CKEN (CLK_APBCLK1[18]).
- Reset Configuration
 - Reset BPWM0 controller in BPWM0RST (SYS_IPRST2[18]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
BPWM0	BPWM0_CH0	PA.11	MFP9
		PA.0	MFP12
		PE.2	MFP13
	BPWM0_CH1	PA.10	MFP9

		PA.1	MFP12
		PE.3	MFP13
	BPWM0_CH2	PA.9	MFP9
		PA.2	MFP12
		PE.4	MFP13
	BPWM0_CH3	PA.8	MFP9
		PA.3	MFP12
		PE.5	MFP13
	BPWM0_CH4	PF.5	MFP8
		PA.4	MFP12
		PE.6	MFP13
	BPWM0_CH5	PF.4	MFP8
		PD.12	MFP9
		PA.5	MFP12
		PE.7	MFP13

6.11.5 Functional Description

6.11.5.1 BPWM Prescaler

The BPWM prescaler is used to divide clock source, prescaler counting CLKPSC +1 times, and the BPWM counter only counts once. The prescale is set by CLKPSC (BPWM_CLKPSC[11:0]). Figure 6.11-5 shows an example of BPWM channel 0 CLKPSC waveform. The prescale counter will reload CLKPSC at the beginning of the next prescale counter down-count.

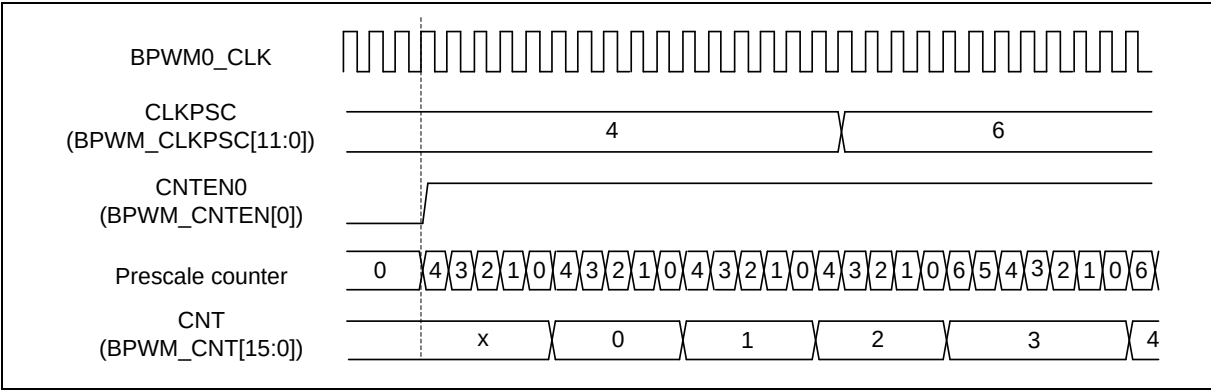


Figure 6.11-5 BPWM_CH0 CLKPSC Waveform

6.11.5.2 BPWM Counter

BPWM has one counter, and supports 3 counter types operation: Up Counter, Down Counter and Up-Down Counter types.

For BPWM channel0, CNT(BPWM_CNT[15:0]) can clear to 0x00 by CNTCLR0 (BPWM_CNTCLR[0]) when the prescale counter counts down to 0, and CNTCLR0(BPWM_CNTCLR[0]) will be set as 0 by hardware automatically.

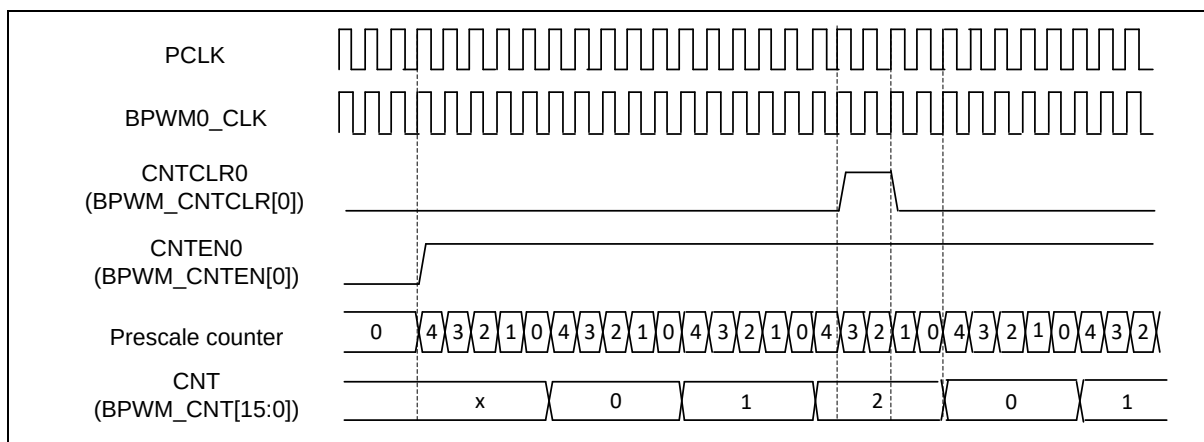


Figure 6.11-6 BPWM Counter Clear Waveform

6.11.5.3 Up Counter Type

In the up counter operation, the 16 bits BPWM counter is an up counter and starts up-counting from 0 to PERIOD (BPWM_PERIOD) to finish a BPWM period. The current counter value can be found by reading the CNT (BPWM_CNT[15:0]). BPWM generates zero point event when the counter counts to 0 and generates period point event when counting to PERIOD. An example of the period time in up counter type, the BPWM period time = (PERIOD+1) * (CLKPSC+1) * BPWMx_CLK clock time, is shown in Figure 6.11-7.

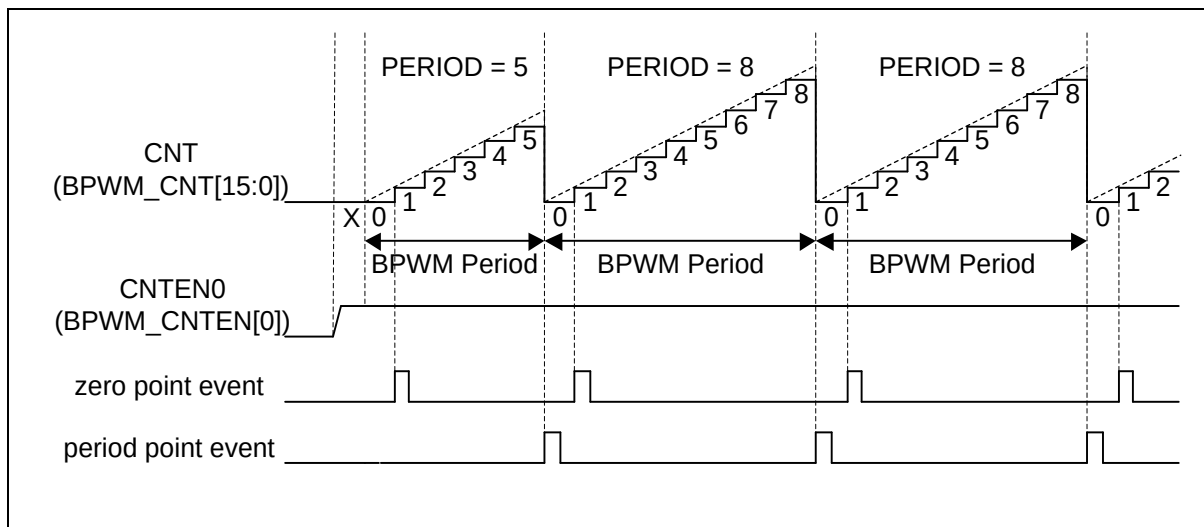


Figure 6.11-7 BPWM Up Counter Type

6.11.5.4 Down Counter Type

In the down counter operation, the 16 bits BPWM counter is a down counter and starts down-counting from PERIOD to 0 to finish a BPWM period. The current counter value can be found by reading the CNT. BPWM generates zero point event when the counter counts to 0 and generates period point event when counting to PERIOD. An example of the period time in down counter type, the BPWM period time = (PERIOD+1) * (CLKPSC+1) * BPWMx CLK clock time, is shown in Figure 6.11-8.

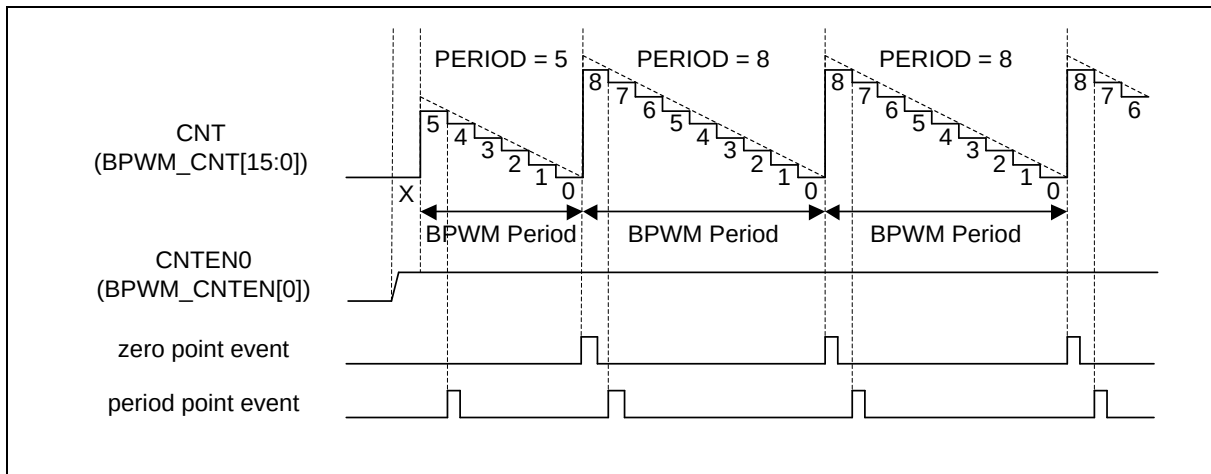


Figure 6.11-8 BPWM Down Counter Type

6.11.5.5 Up-Down Counter Type

In the up-down counter operation, the 16 bits BPWM counter is an up-down counter and starts counting up from 0 to PERIOD and then starts counting down to 0 to finish a BPWM period. The current counter value can be found by reading the CNT. BPWM generates zero point event when counter counts to 0 and generates center point event when counting to PERIOD. An example of the period time in up-down counter type, the BPWM period time = $(2 \times \text{PERIOD}) \times (\text{CLKPSC} + 1) \times \text{BPWMx\_CLK}$ clock time, is shown in Figure 6.11-9. The DIRF (BPWM_CNT[16]) is a counter direction indicator flag, where high is up counting, and low is down counting.

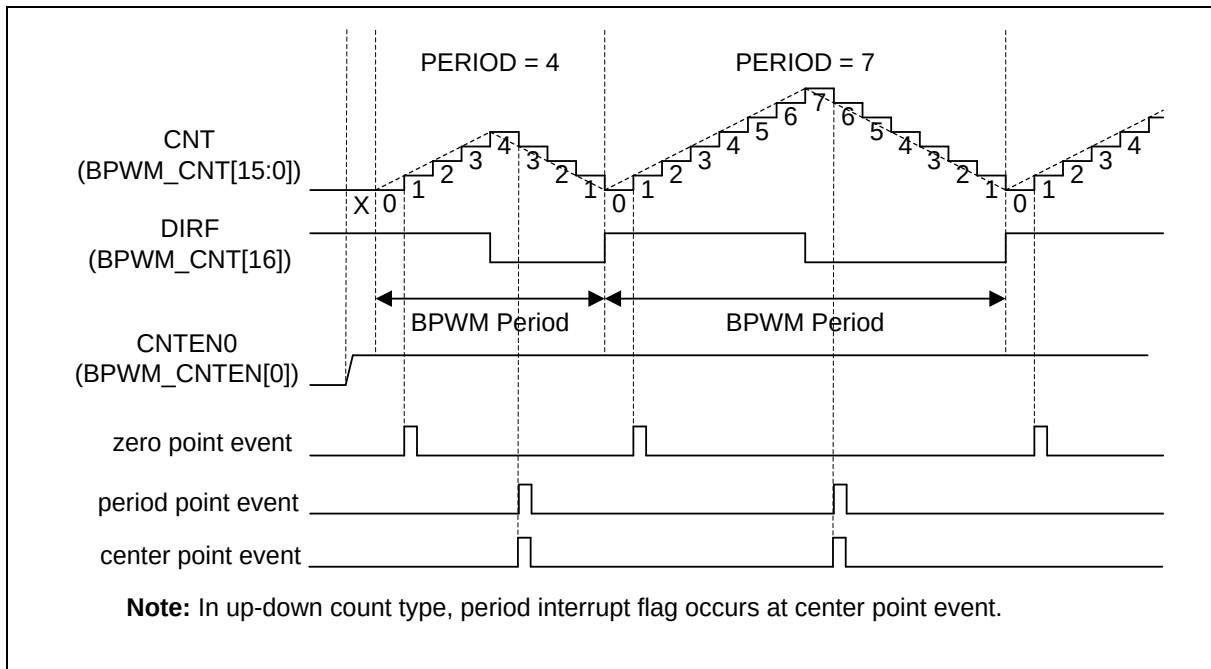


Figure 6.11-9 BPWM Up-Down Counter Type

6.11.5.6 BPWM Comparator

The CMPDAT (BPWM_CMPDATn[15:0]) is a basic comparator register of BPWM channel n; each channel only has one CMPDAT. The CMPDAT's value is continuously compared to the counter value. When the counter is equal to the compared register, BPWM generates an event and uses the event to generate BPWM pulse, interrupt or use to trigger EADC. In up-down counter type, two events will be

generated in a BPWM period as shown in Figure 6.11-10.

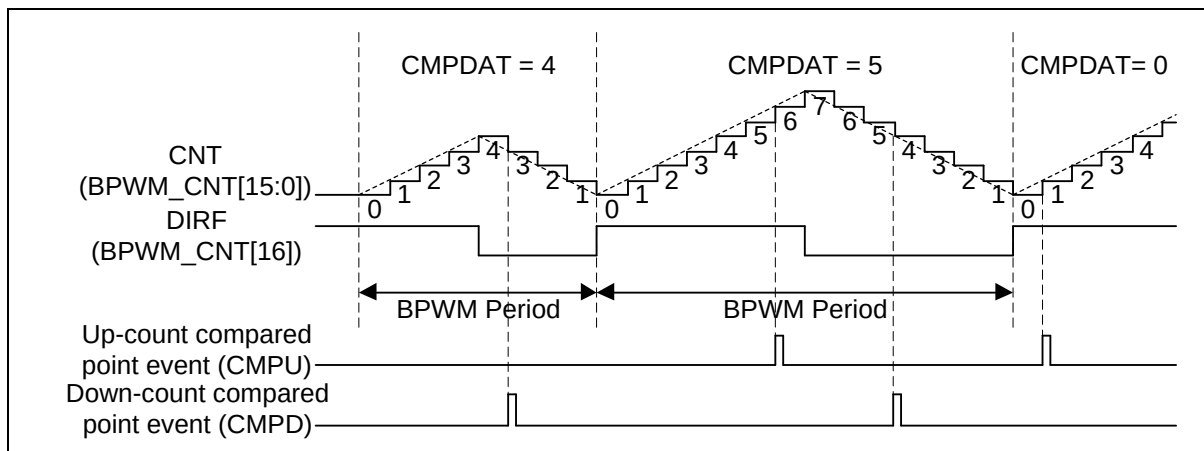


Figure 6.11-10 BPWM CMPDAT Events in Up-Down Counter Type

6.11.5.7 Period Loading Mode

Period Loading mode is the default loading mode. It has lowest priority in loading modes. PERIOD and CMPDAT will both load to their buffer while a period is completed. For example, after BPWM counter up counts from 0 to PERIOD in up-counter operation or down counts from PERIOD to 0 in the down-counter operation or up counts from 0 to PERIOD and then down counts to 0 in up-down counter operation.

Figure 6.11-11 shows period loading timing of up-count operation, where PERIOD DATA0 denotes the initial data of PERIOD, PERIOD DATA1 denotes the first updated PERIOD data by software and so on, CMPDAT also follows this rule. The following describes steps sequence of Figure 6.11-11. User can know the PERIOD and CMPDAT update condition, by watching BPWM period and CMPU event.

1. Software writes CMPDAT DATA1 to CMPDAT at point 1.
2. Hardware loads CMPDAT DATA1 to CMPBUF at the end of PWM period at point 2.
3. Software writes PERIOD DATA1 to PERIOD at point 3.
4. Hardware loads PERIOD DATA1 to PBUF at the end of PWM period at point 4.
5. Software writes PERIOD DATA2 to PERIOD at point 5.
6. Hardware loads PERIOD DATA2 to PBUF at the end of PWM period at point 6.

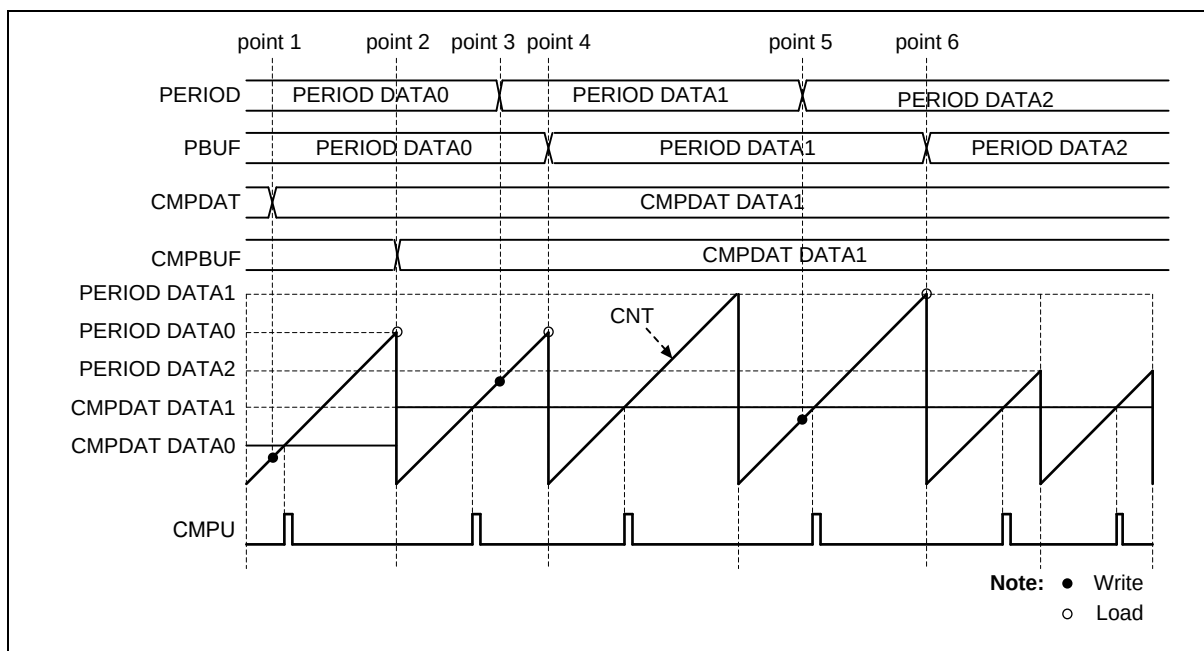


Figure 6.11-11 Period Loading Mode with Up-Counter Type

6.11.5.8 Immediately Loading Mode

If the IMMLDENn (BPWM_CTL0[21:16]) bit which corresponds to BPWM channel n is set to 1, software will load a value to buffer from PERIOD and CMPDAT immediately while software updates PERIOD or CMPDAT. If the update PERIOD value is less than current counter value, counter will count to 0xFFFF, when counter counts to 0xFFFF and prescale count to 0, the flag CNTMAX0 (BPWM_STATUS[0]) will raise, and then counter will count wraparound. Immediately loading mode has the highest priority. If IMMLDENn has been set, other loading mode for channel n will become invalid. Figure 6.11-12 shows an example and its steps sequence is described below.

1. Software writes CMPDAT DATA1 and hardware immediately loads CMPDAT DATA1 to CMPBUF at point 1.
2. Software writes PERIOD DATA1 which is greater than the current counter value at point 2; counter will continue counting until it is equal to PERIOD DATA1 to finish a period loading.
3. Software writes PERIOD DATA2 which is less than the current counter value at point 3; counter will continue counting to its maximum value 0xFFFF and count wraparound from 0 to PERIOD DATA2 to finish this period loading.

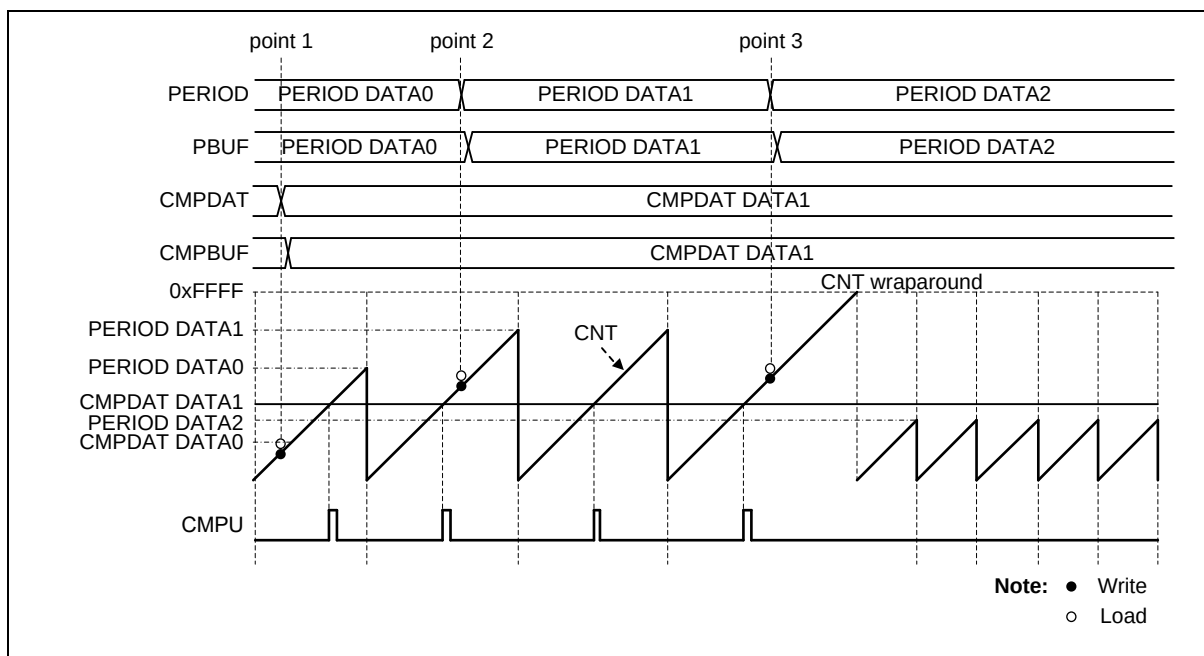


Figure 6.11-12 Immediately Loading Mode with Up-Counter Type

6.11.5.9 Center Loading Mode

If the CTRLDN (BPWM_CTL0[5:0]) bit that corresponds to BPWM channel n is set to 1 and in up-down counter type, CMPDAT will load to CMPBUFn in center of a period, that is, counter counts to PERIOD. PERIOD loading timing is the same as period loading mode. Figure 6.11-13 shows an example and its steps sequence is described below.

1. Software writes CMPDAT DATA1 at point 1.
2. Hardware loads CMPDAT DATA1 to CMPBUF at center of PWM period at point 2.
3. Software writes PERIOD DATA1 at point 3.
4. Hardware loads PERIOD DATA1 to PBUF at the end of PWM period at point 4.
5. Software writes CMPDAT DATA2 at point 5.
6. Hardware loads CMPDAT DATA2 to CMPBUF at center of PWM period at point 6.
7. Software writes PERIOD DATA2 at point 7.
8. Hardware loads PERIOD DATA2 to PBUF at the end of PWM period at point 8.

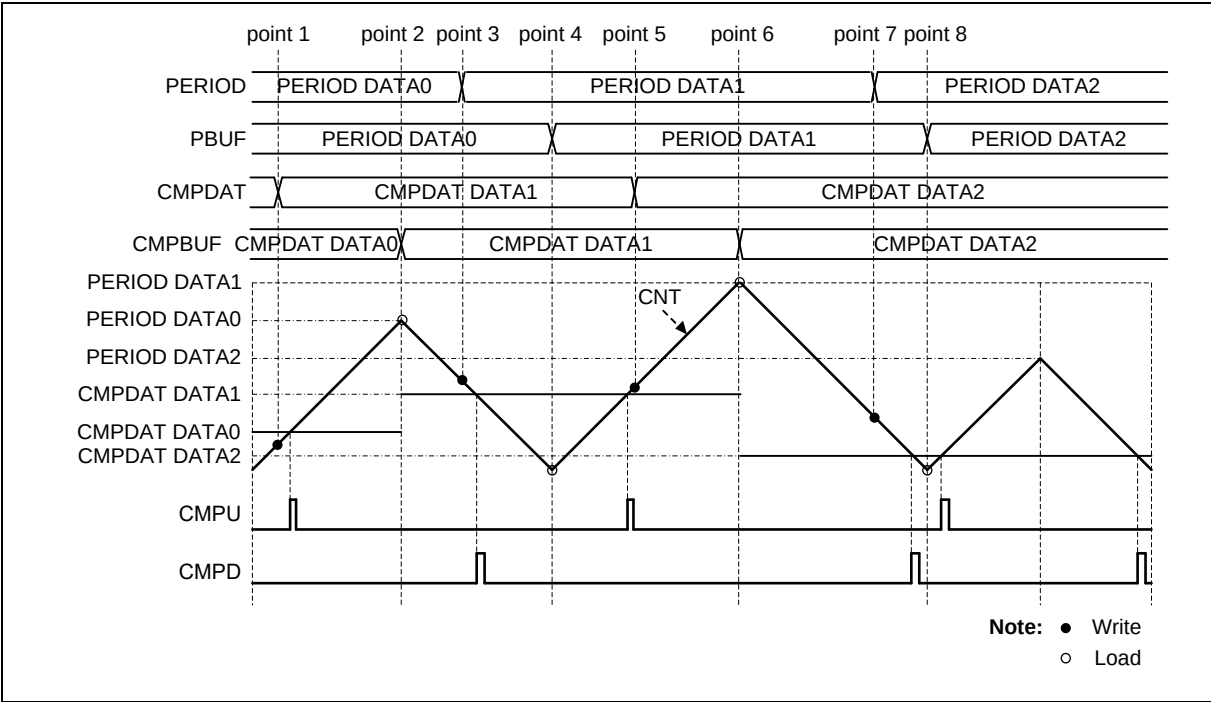


Figure 6.11-13 Center Loading Mode with Up-Down-Counter Type

6.11.5.10 BPWM Pulse Generator

The BPWM pulse generator uses counter and comparator events to generate BPWM pulse. The events are: zero point, period point in up counter type and down counter type, center point in up-down counter type and counter equal to comparator point in three types. As to up-down counter type, there are two counter equal comparator points, one at up count another at down count.

Each event point can decide BPWM waveform to do nothing (X), set Low (L), set High (H) or toggle (T) by setting BPWM_WGCTL0 and BPWM_WGCTL1 registers. Using these points can easily generate asymmetric BPWM pulse or variant waveform as shown in Figure 6.11-14. In the figure, there is a comparator n to generate BPWM pulse, where n denotes channel number 0 to 5. CMPU denotes CNT is equal to CMPDAT when counting up, and CMPD denotes CNT is equal to CMPDAT when counting down.

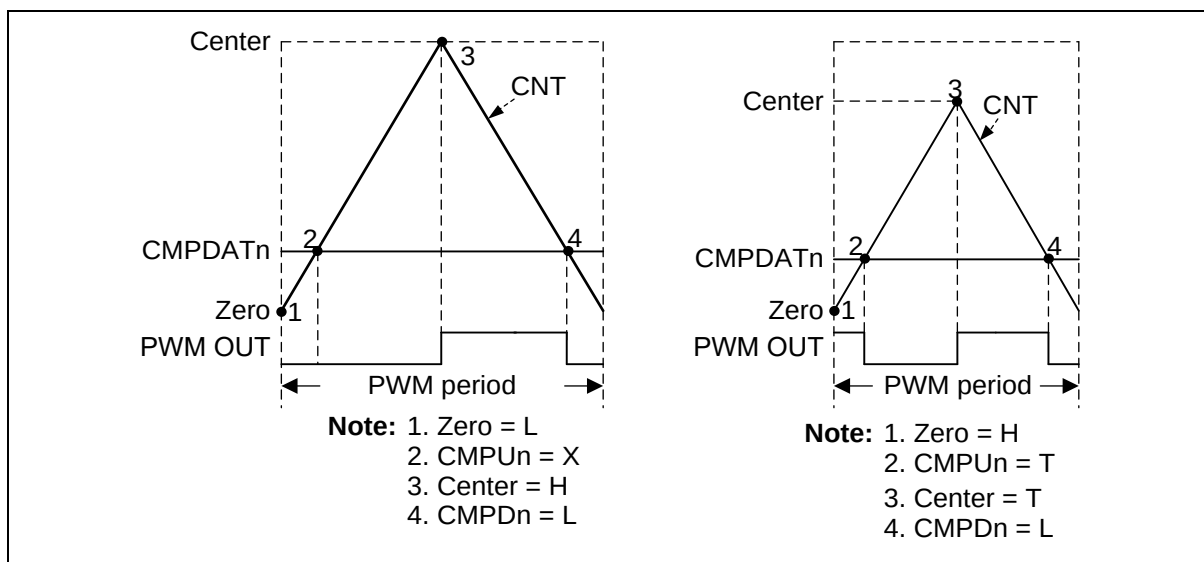


Figure 6.11-14 BPWM Pulse Generation (Left: Asymmetric Pulse, Right: Variety Pulse)

The generation events may be sometimes set to the same value, as the reason, events priority between different counter types are list below, up counter type (Table 6.11-2) down counter type (Table 6.11-3) and up-down counter type (Table 6.11-4). By using event priority, user can easily generate 0% to 100% duty pulse as shown in Figure 6.11-15.

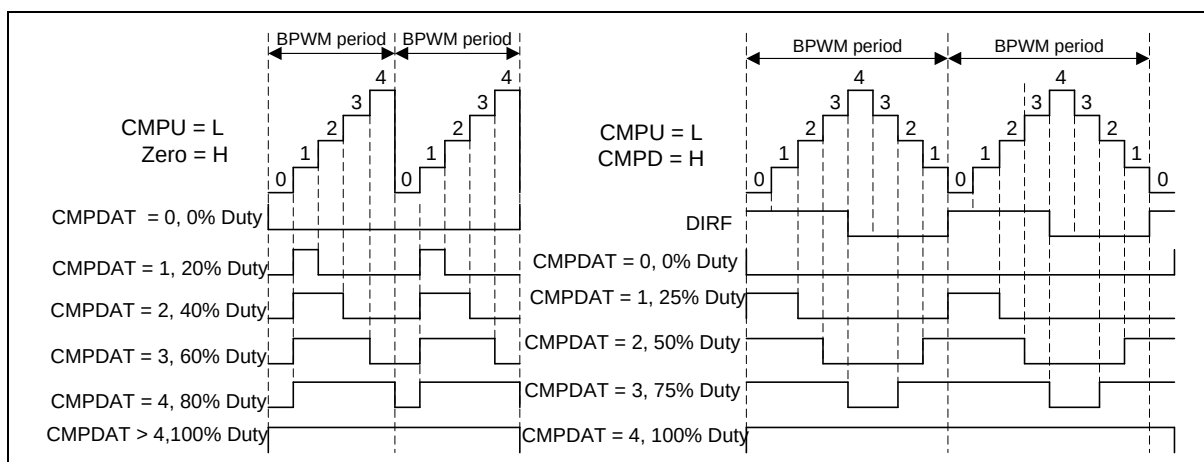


Figure 6.11-15 BPWM 0% to 100% Pulse Generation (Left: Up Counter Type, Right: Up-down Counter Type)

Priority	Up Event
1 (Highest)	Period event (CNT = PERIOD)
2	Compare up event(CNT = CMPUn)
3 (Lowest)	Zero event (CNT = 0)

Table 6.11-2 BPWM Pulse Generation Event Priority for Up-Counter

Priority	Down Event
1 (Highest)	Zero event (CNT = 0)
2	Compare down event (CNT = CMPDn)
3 (Lowest)	Period event (CNT = PERIOD)

Table 6.11-3 BPWM Pulse Generation Event Priority for Down-Counter

Priority	Up Event	Down Event
1 (Highest)	Compare up event (CNT = CMPUn)	Compare down event (CNT = CMPDn)
2 (Lowest)	Zero event (CNT = 0)	Period (center) event (CNT = PERIOD)

Table 6.11-4 BPWM Pulse Generation Event Priority for Up-Down-Counter

6.11.5.11 BPWM Output Control

After BPWM pulse generation, there are three steps to control the output of BPWM channels. There are Mask, Pin Polarity and Output Enable three steps as shown in Figure 6.11-16.

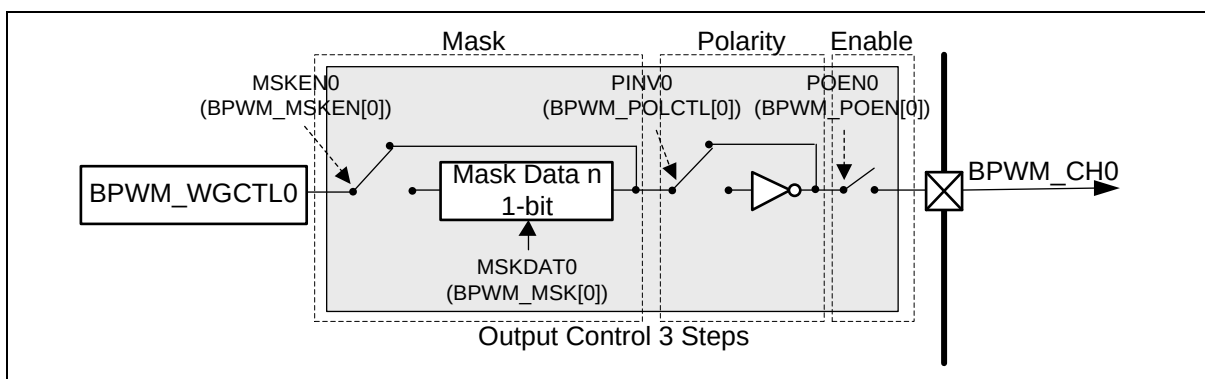


Figure 6.11-16 BPWM_CH0 Output Control 3 Steps

6.11.5.12 BPWM Mask Output Function

Each of the BPWM output channels can be manually overridden by using the appropriate bits in the BPWM Mask Enable Control Register (BPWM_MSKEN) and BPWM Masked Data Register (BPWM_MSK) to drive the BPWM channel outputs to specified logic states independent of the duty cycle comparison units. The BPWM mask bits are useful when controlling various types of Electrically Commutated Motor (ECM) like a BLDC motor. The BPWM_MSKEN register contains six bits, MSKENn(BPWM_MSKEN[5:0]) determine which BPWM channel output will be overridden, MSKENn(BPWM_MSKEN[5:0]) bits are active-high. The BPWM_MSK register contains six bits, MSKDATn(BPWM_MSK[5:0]), which determine the state of the BPWM channel output when the channel is masked via the MSKDAT bits. Figure 6.11-17 shows an example of how BPWM mask control can be used for the override feature.

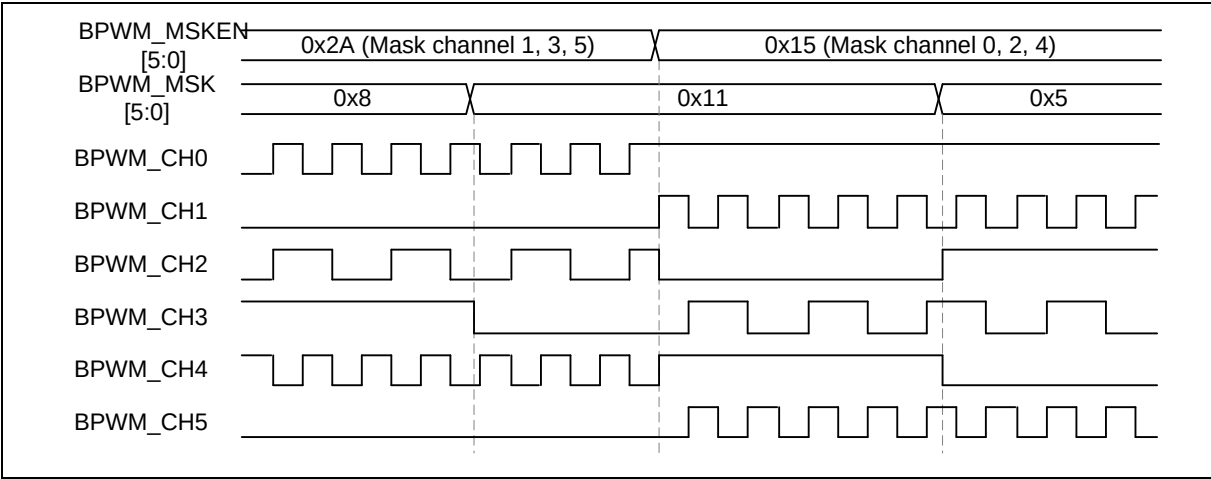


Figure 6.11-17 Mask Control Waveform Illustration

6.11.5.13 Polarity Control

Each BPWM port from BPWM_CH0 to BPWM_CH5 has an independent polarity control module to configure the polarity of the active state of BPWM output. By default, the BPWM output is active high. This implies the BPWM OFF state is low and ON state is high. This definition is variable through setting BPWM Negative Polarity Control Register (BPWM_POLCTL), for each individual BPWM channel. Figure 6.11-18 shows the initial state before BPWM starts with different polarity settings.

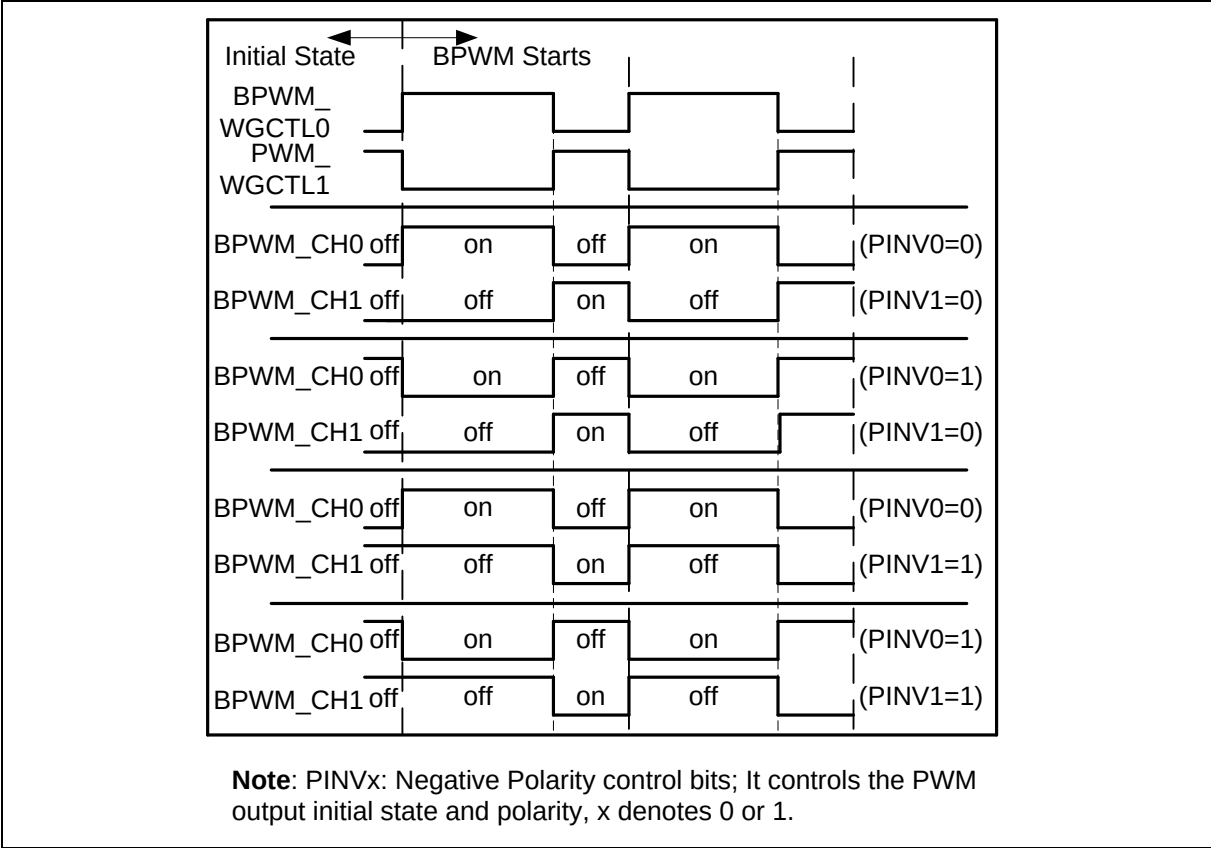


Figure 6.11-18 Initial State and Polarity Control

6.11.5.14 BPWM Interrupt Generator

There are two independent interrupts for each BPWM as shown in Figure 6.11-19.

BPWM interrupt (BPWM_INT) comes from BPWM complementary pair events. The counter can generate the Zero point Interrupt Flag ZIF0 (BPWM_INTSTS0[0]) and the Period point Interrupt Flag PIF0 (BPWM_INTSTS0[8]). When BPWM channel n's counter is equal to the comparator value stored in BPWM_CMPDATn, the different interrupt flags will be triggered depending on the counting direction. If the matching occurs at up-count direction, the Up Interrupt Flag CMPUIFn (BPWM_INTSTS0[21:16]) is set and if matching at the opposite direction, the Down Interrupt Flag CMPDIFn (BPWM_INTSTS0[29:24]) is set. If the correspond interrupt enable bits are set, the trigger events will generates interrupt signals.

Another interrupt is the capture interrupt (CAP_INT). It shares the BPWM_INT vector in NVIC, CAP_INT can be generated when the CAPRIFn (BPWM_CAPIF[5:0]) is triggered and the Capture Rising Interrupt Enable bit CAPRIENn (BPWM_CAPIEN[5:0]) is set to 1. Or in the falling edge condition, the CAPFIFn (BPWM_CAPIF[13:8]) can be triggered when the Capture Falling Interrupt Enable bit CAPFIENn (BPWM_CAPIEN[13:8]) is set to 1.

Figure 6.11-19 demonstrates the architecture of the BPWM interrupts.

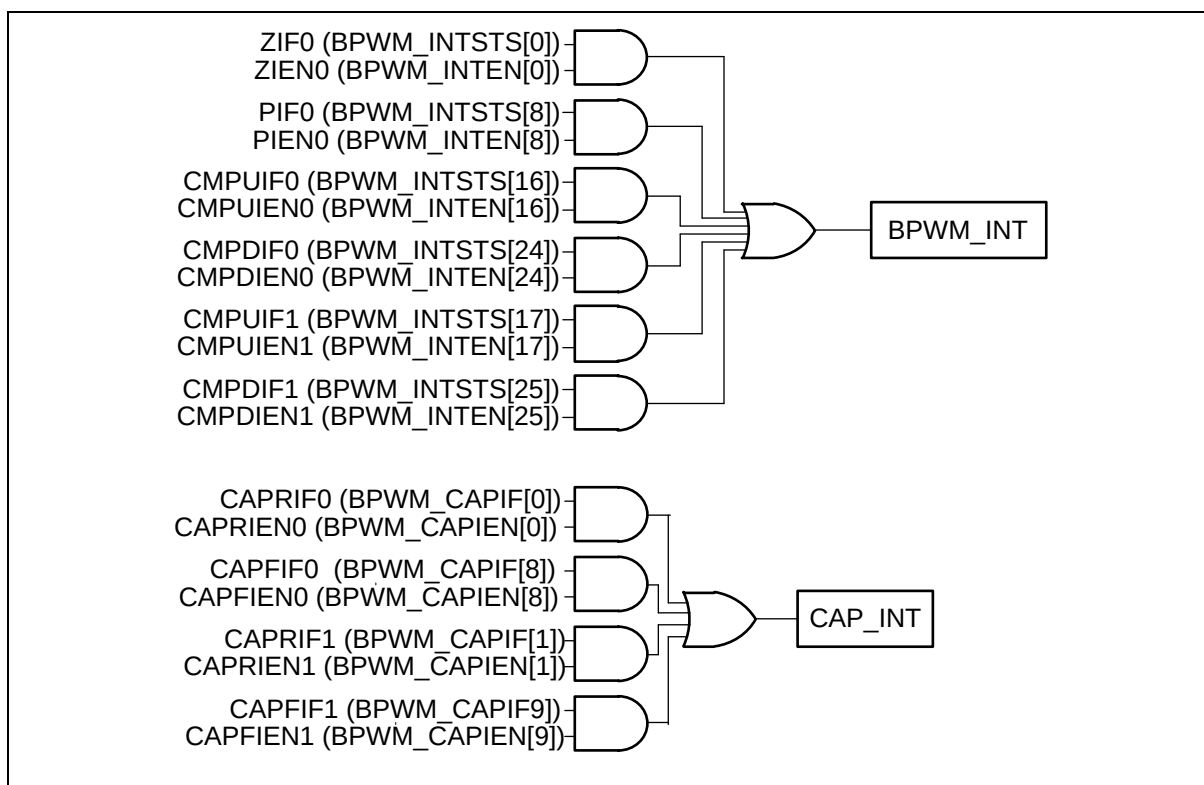


Figure 6.11-19 BPWM_CH0 and BPWM_CH1 Pair Interrupt Architecture Diagram

6.11.5.15 BPWM Trigger EADC Generator

BPWM can be one of the EADC conversion trigger source. Each BPWM pair channels share the same trigger source. Setting TRGSELn is to select the trigger sources, where TRGSELn is TRGSEL0, TRGSEL1, ..., and TRGSEL5, which are located in BPWM_EADCTS0 and BPWM_EADCTS1, respectively. Setting TRGENn is to enable the trigger output to EADC, where TRGENn is TRGEN0, TRGEN1, ..., TRGEN5, which are located in BPWM_EADCTS0[7], BPWM_EADCTS0[15], BPWM_EADCTS0[23], BPWM_EADCTS0[31], BPWM_EADCTS1[7] and BPWM_EADCTS1[15], respectively. The number n (n = 0,1, ...,5) denotes BPWM channel number.

There are 7 BPWM events can be selected as the trigger source for one pair of channels. Figure 6.11-20

is an example of BPWM_CH0 and BPWM_CH1. BPWM can trigger EADC to start conversion in different timings by setting PERIOD and CMPDAT. Figure 6.11-22 is the trigger EADC timing waveform in the up-down counter type.

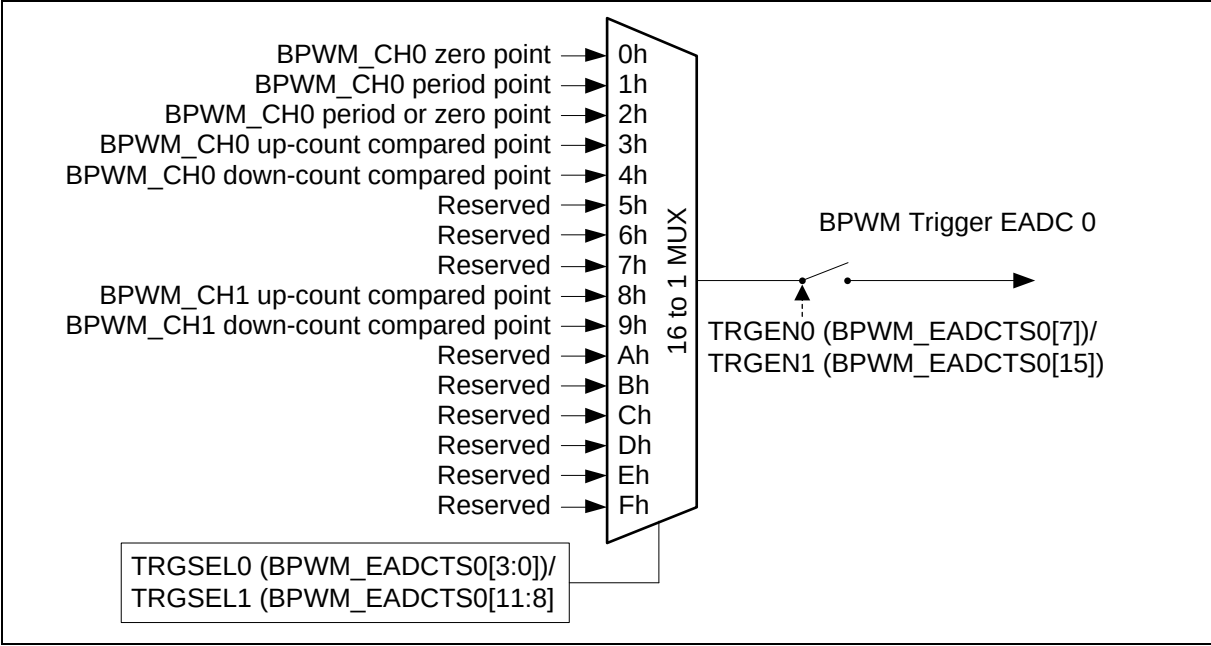


Figure 6.11-20 BPWM_CH0 and BPWM_CH1 Pair Trigger EADC Source Block Diagram

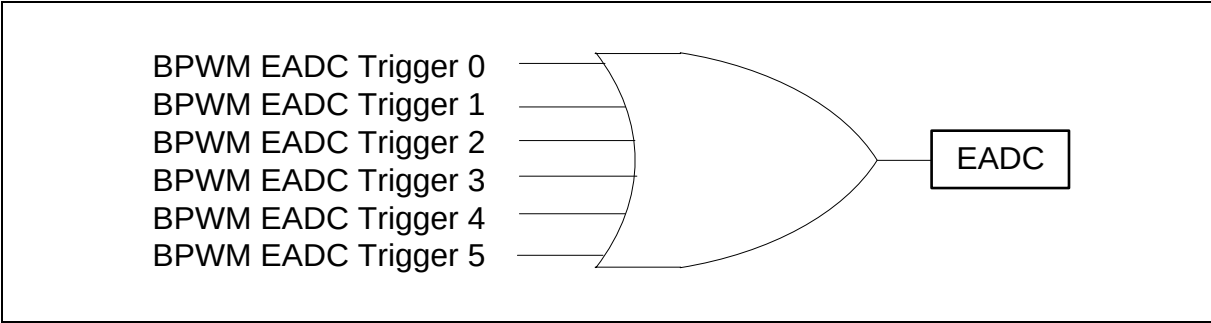


Figure 6.11-21 BPWM CH0~ CH5 Trigger EADC Block Diagram

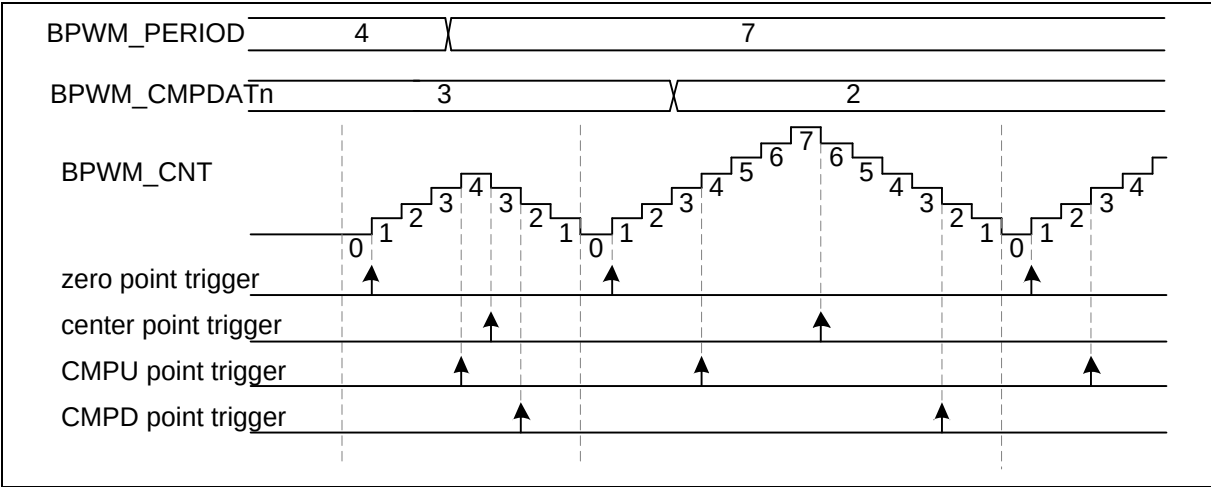


Figure 6.11-22 BPWM Trigger EADC in Up-Down Counter Type Timing Waveform

6.11.5.16 Capture Operation

The channels of the capture input and the BPWM output share the same pin and counter. The counter can operate in up or down counter type. The capture function will always latch the BPWM counter to the register RCAPDATn (BPWM_RCAPDATn[15:0]) or the register FCAPDATn (BPWM_FCAPDATn[15:0]) if the input channel has a rising transition or a falling transition, respectively. The capture function will also generate an interrupt CAP_INT (using BPWM_INT vector) if the rising or falling latch occurs and the corresponding channel n's rising or falling interrupt enable bits are set, where the CAPRIENn (BPWM_CAPIEN[5:0]) is for the rising edge and the CAPFIENn (BPWM_CAPIEN[13:8]) is for the falling edge. When rising or falling latch occurs, the corresponding BPWM counter may be reloaded with the value BPWM_PERIOD, depending on the setting of RCRLDENn or FCRLDENn (where RCRLDENn and FCRLDENn are located at BPWM_CAPCTL[21:16] and BPWM_CAPCTL[29:24], respectively). Note that the corresponding GPIO pins must be configured as the capture function by enable the CAPINENn (BPWM_CAPINEN[5:0]) for the corresponding capture channel n. Figure 6.11-23 is the capture block diagram of channel 0.

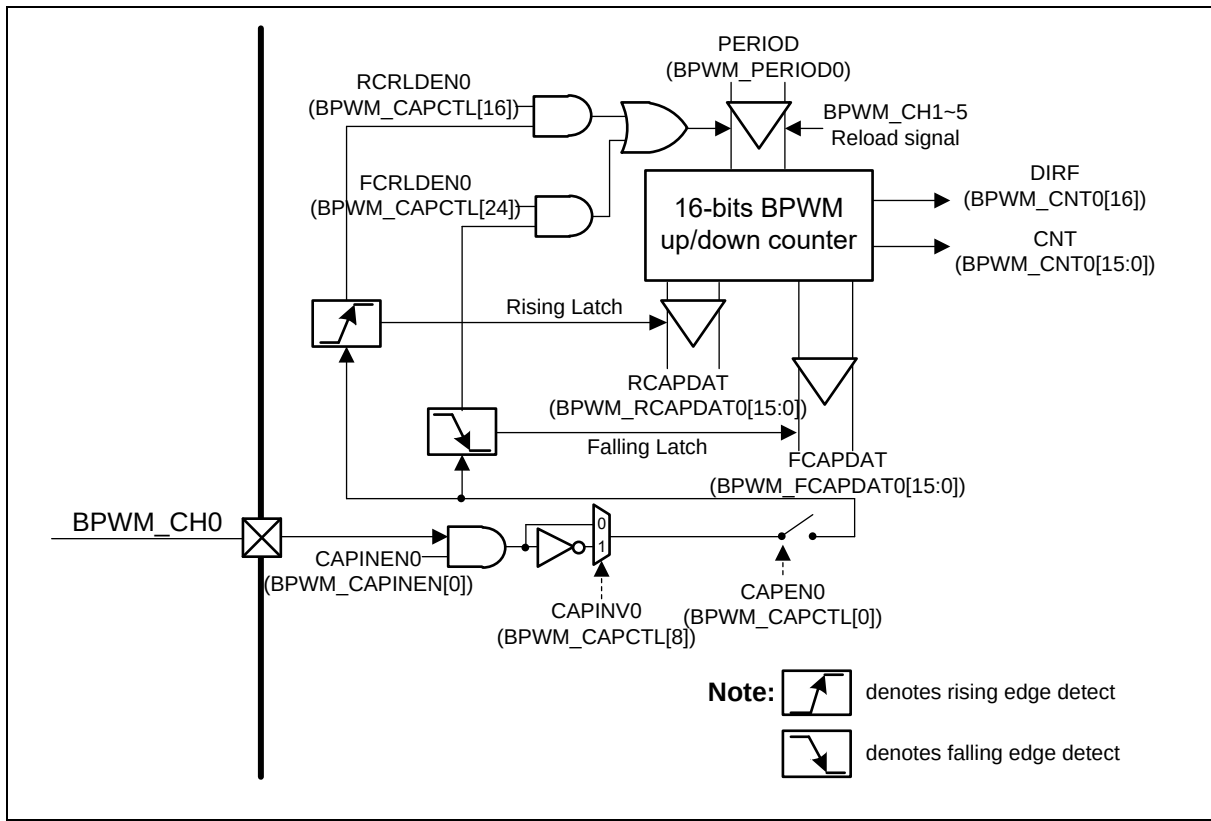


Figure 6.11-23 BPWM_CH0 Capture Block Diagram

Figure 6.11-24 illustrates the capture function timing. In this case, the capture counter is set as BPWM down counter type and the PERIOD is set to 8 so that the counter counts in the down direction, from 8 to 0. When detecting a falling edge at the capture input pin, the capture function latches counter value to the BPWM_FCAPDATn. When detecting the rising edge, it latches the counter value to the BPWM_RCAPDATn. In this timing diagram, when the falling edge is detected at the first time, the capture function will reload the counter value from the PERIOD setting because the FCRLDENn is enabled. But at the second time, the falling edge does not result in a reload because of the disabled FCRLDENn. In this example, the counter also reloads at the rising edge of the capture input because the RCRLDENn is enabled, too.

Moreover, if the case is setup as the up counter type, the counter will reload the value zero and count up to the value PERIOD. It is important that the counter is shared by all channels, so the counter reloads time also controlled by all channels' reload signals.

Figure 6.11-24 also illustrates the timing example for the interrupt and interrupt flag generation. When the rising edge at channel n is detected, the corresponding bit CAPRIFn (BPWM_CAPIF[5:0]) is set by hardware. Similarly, a falling edge detection at channel n causes the corresponding bit CAPFIFn (BPWM_CAPIF[13:8]) set by hardware. CAPRIFn (BPWM_CAPIF[5:0]) and CAPFIFn (BPWM_CAPIF[13:8]) can be cleared by software by writing '1'. If the CAPRIFn (BPWM_CAPIF[5:0]) is set and the CAPRIENn is enabled, the capture function generates an interrupt. If the CAPFIFn (BPWM_CAPIF[13:8]) is set and the CAPFIENn is enabled, the interrupt also happens.

A condition which is not shown in this figure is: if the rising latch happens again when the CAPRIFn (BPWM_CAPIF[5:0]) is already set, the Over run status CRIFOVn (BPWM_CAPSTS[5:0]) will be set to 1 by hardware to indicate the CAPRIFn (BPWM_CAPIF[5:0]) overrunning. Also, if the falling latch happens again, the same hardware operation occurs for the interrupt flag CAPFIF n (BPWM_CAPIF[13:8]) and the Over run status CFIFOVn (BPWM_CAPSTS[13:8]).

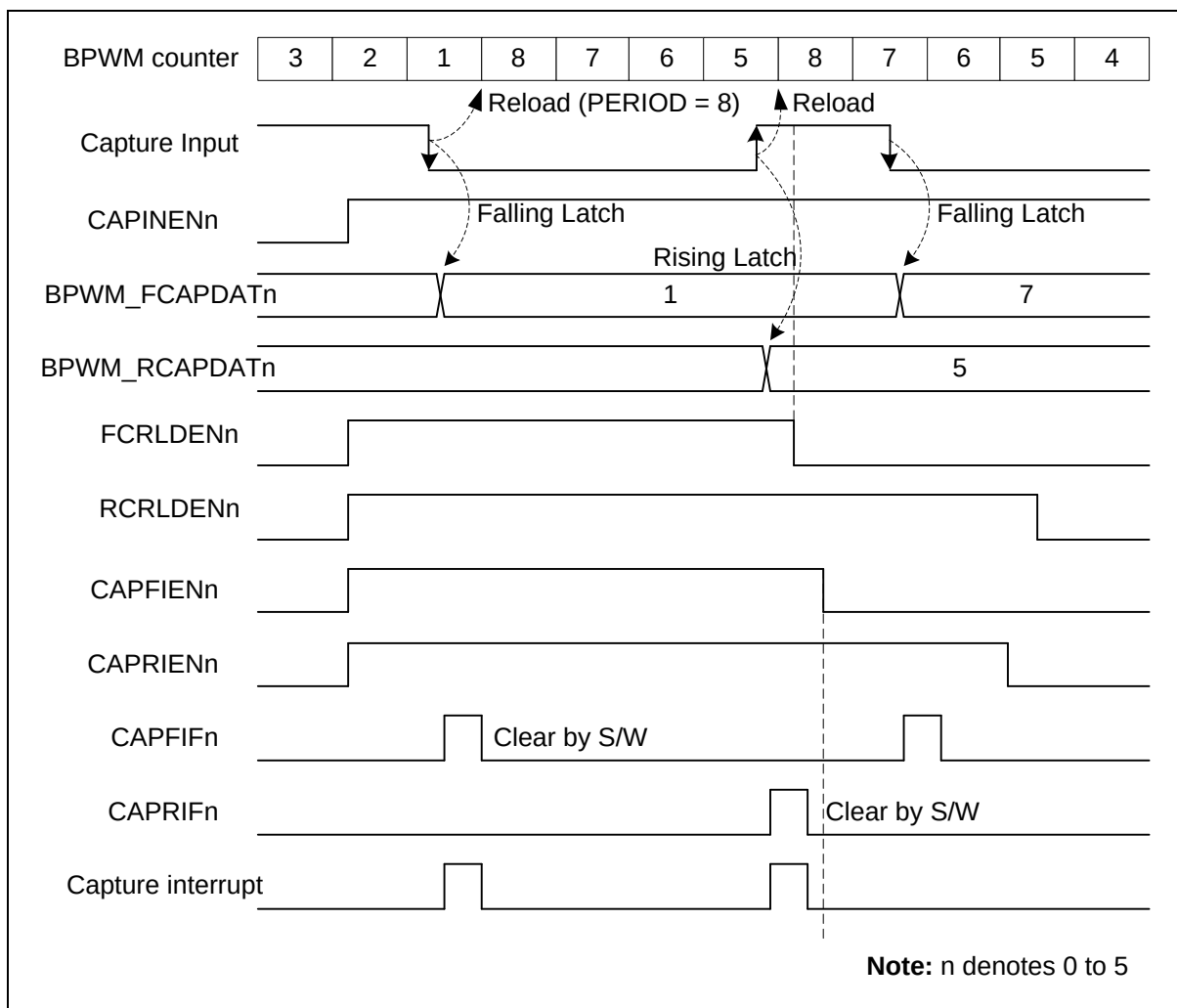


Figure 6.11-24 Capture Operation Waveform

The capture pulse width meeting the following conditions can be calculated according to the formula.

1. The capture positive or negative pulse width is shorter than a counter period.
2. The counter operates in down counter type.
3. The counter can be reloaded by both falling and rising capture events through setting FCRLDENn and RCRLDENn bits of PWM_CAPCTL register to 1.

For the negative pulse case, the channel low pulse width is calculated as $(BPWM\_PERIOD + 1 - BPWM\_RCAPDATn)$ BPWM counter time, where one BPWM counter time is $(CLKPSC+1) * BPWMx\_CLK$ clock time. In the case shown in Figure 6.11-24, low pulse width is $8+1-5 = 4$ BPWM counter time.

For the positive pulse case, the channel high pulse width is calculated as $(BPWM\_PERIOD + 1 - BPWM\_FCAPDATn)$ BPWM counter time, where one BPWM counter time is $(CLKPSC+1) * BPWMx\_CLK$ clock time. In the case shown in Figure 6.11-24, high pulse width is $8+1-7 = 2$ BPWM counter time.

6.11.6 Register Map

R: read only, W: write only, R/W: both read and write

Register	Offset	R/W	Description	Reset Value
BPWM Base Address: BPWM_BA = 0x4005_A000				
BPWM_CTL0	BPWM_BA+0x00	R/W	BPWM Control Register 0	0x0000_0000
BPWM_CTL1	BPWM_BA+0x04	R/W	BPWM Control Register 1	0x0000_0000
BPWM_CLKSRC	BPWM_BA+0x10	R/W	BPWM Clock Source Register	0x0000_0000
BPWM_CLKPSC	BPWM_BA+0x14	R/W	BPWM Clock Prescale Register	0x0000_0000
BPWM_CNTEN	BPWM_BA+0x20	R/W	BPWM Counter Enable Register	0x0000_0000
BPWM_CNTCLR	BPWM_BA+0x24	R/W	BPWM Clear Counter Register	0x0000_0000
BPWM_PERIOD	BPWM_BA+0x30	R/W	BPWM Period Register	0x0000_0000
BPWM_CMPDAT0	BPWM_BA+0x50	R/W	BPWM Comparator Register 0	0x0000_0000
BPWM_CMPDAT1	BPWM_BA+0x54	R/W	BPWM Comparator Register 1	0x0000_0000
BPWM_CMPDAT2	BPWM_BA+0x58	R/W	BPWM Comparator Register 2	0x0000_0000
BPWM_CMPDAT3	BPWM_BA+0x5C	R/W	BPWM Comparator Register 3	0x0000_0000
BPWM_CMPDAT4	BPWM_BA+0x60	R/W	BPWM Comparator Register 4	0x0000_0000
BPWM_CMPDAT5	BPWM_BA+0x64	R/W	BPWM Comparator Register 5	0x0000_0000
BPWM_CNT	BPWM_BA+0x90	R	BPWM Counter Register	0x0000_0000
BPWM_WGCTL0	BPWM_BA+0xB0	R/W	BPWM Generation Register 0	0x0000_0000
BPWM_WGCTL1	BPWM_BA+0xB4	R/W	BPWM Generation Register 1	0x0000_0000
BPWM_MSKEN	BPWM_BA+0xB8	R/W	BPWM Mask Enable Register	0x0000_0000
BPWM_MSK	BPWM_BA+0xBC	R/W	BPWM Mask Data Register	0x0000_0000
BPWM_POLCTL	BPWM_BA+0xD4	R/W	BPWM Pin Polar Inverse Register	0x0000_0000

BPWM_POEN	BPWM_BA+0xD8	R/W	BPWM Output Enable Register	0x0000_0000
BPWM_INTEN	BPWM_BA+0xE0	R/W	BPWM Interrupt Enable Register	0x0000_0000
BPWM_INTSTS	BPWM_BA+0xE8	R/W	BPWM Interrupt Flag Register	0x0000_0000
BPWM_EADCTS0	BPWM_BA+0xF8	R/W	BPWM Trigger EADC Source Select Register 0	0x0000_0000
BPWM_EADCTS1	BPWM_BA+0xFC	R/W	BPWM Trigger EADC Source Select Register 1	0x0000_0000
BPWM_STATUS	BPWM_BA+0x120	R/W	BPWM Status Register	0x0000_0000
BPWM_CAPINEN	BPWM_BA+0x200	R/W	BPWM Capture Input Enable Register	0x0000_0000
BPWM_CAPCTL	BPWM_BA+0x204	R/W	BPWM Capture Control Register	0x0000_0000
BPWM_CAPSTS	BPWM_BA+0x208	R	BPWM Capture Status Register	0x0000_0000
BPWM_RCAPDAT0	BPWM_BA+0x20C	R	BPWM Rising Capture Data Register 0	0x0000_0000
BPWM_FCAPDAT0	BPWM_BA+0x210	R	BPWM Falling Capture Data Register 0	0x0000_0000
BPWM_RCAPDAT1	BPWM_BA+0x214	R	BPWM Rising Capture Data Register 1	0x0000_0000
BPWM_FCAPDAT1	BPWM_BA+0x218	R	BPWM Falling Capture Data Register 1	0x0000_0000
BPWM_RCAPDAT2	BPWM_BA+0x21C	R	BPWM Rising Capture Data Register 2	0x0000_0000
BPWM_FCAPDAT2	BPWM_BA+0x220	R	BPWM Falling Capture Data Register 2	0x0000_0000
BPWM_RCAPDAT3	BPWM_BA+0x224	R	BPWM Rising Capture Data Register 3	0x0000_0000
BPWM_FCAPDAT3	BPWM_BA+0x228	R	BPWM Falling Capture Data Register 3	0x0000_0000
BPWM_RCAPDAT4	BPWM_BA+0x22C	R	BPWM Rising Capture Data Register 4	0x0000_0000
BPWM_FCAPDAT4	BPWM_BA+0x230	R	BPWM Falling Capture Data Register 4	0x0000_0000
BPWM_RCAPDAT5	BPWM_BA+0x234	R	BPWM Rising Capture Data Register 5	0x0000_0000

BPWM_FCAPDAT5	BPWM_BA+0x238	R	BPWM Falling Capture Data Register 5	0x0000_0000
BPWM_CAPIEN	BPWM_BA+0x250	R/W	BPWM Capture Interrupt Enable Register	0x0000_0000
BPWM_CAPIF	BPWM_BA+0x254	R/W	BPWM Capture Interrupt Flag Register	0x0000_0000
BPWM_PBUF	BPWM_BA+0x304	R	BPWM PERIOD Buffer	0x0000_0000
BPWM_CMPBUF0	BPWM_BA+0x31C	R	BPWM CMPDAT 0 Buffer	0x0000_0000
BPWM_CMPBUF1	BPWM_BA+0x320	R	BPWM CMPDAT 1 Buffer	0x0000_0000
BPWM_CMPBUF2	BPWM_BA+0x324	R	BPWM CMPDAT 2 Buffer	0x0000_0000
BPWM_CMPBUF3	BPWM_BA+0x328	R	BPWM CMPDAT 3 Buffer	0x0000_0000
BPWM_CMPBUF4	BPWM_BA+0x32C	R	BPWM CMPDAT 4 Buffer	0x0000_0000
BPWM_CMPBUF5	BPWM_BA+0x330	R	BPWM CMPDAT 5 Buffer	0x0000_0000

6.11.7 Register Description

BPWM Control Register 0 (BPWM_CTL0)

Register	Offset	R/W	Description	Reset Value
BPWM_CTL0	BPWM_BA+0x00	R/W	BPWM Control Register 0	0x0000_0000

31	30	29	28	27	26	25	24
DBGTRIOFF	DBGHALT	Reserved					
23	22	21	20	19	18	17	16
Reserved		IMMLDEN5	IMMLDEN4	IMMLDEN3	IMMLDEN2	IMMLDEN1	IMMLDEN0
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		CTRLD5	CTRLD4	CTRLD3	CTRLD2	CTRLD1	CTRLD0

Bits	Description
[31]	DBGTRIOFF ICE Debug Mode Acknowledge Disable (Write Protect) 0 = ICE debug mode acknowledgement effects BPWM output. BPWM pin will be forced as tri-state while ICE debug mode acknowledged. 1 = ICE debug mode acknowledgement Disabled. BPWM pin will keep output no matter ICE debug mode acknowledged or not. Note: This bit is write protected. Refer to SYS_REGLCTL register.
[30]	DBGHALT ICE Debug Mode Counter Halt (Write Protect) If counter halt is enabled, BPWM all counters will keep current value until exit ICE debug mode. 0 = ICE debug mode counter halt Disabled. 1 = ICE debug mode counter halt Enabled. Note: This bit is write protected. Refer to SYS_REGLCTL register.
[29:22]	Reserved Reserved.
[16+n] n=0,1..5	IMMLDENn Immediately Load Enable Bits Each bit n controls the corresponding BPWM channel n. 0 = PERIOD will load to PBUF at the end point of each period. CMPDAT will load to CMPBUF at the end point or center point of each period by setting CTRLDN bit. 1 = PERIOD/CMPDAT will load to PBUF and CMPBUF immediately when software updates PERIOD/CMPDAT. Note: If IMMLDENn is enabled, CTRLDN will be invalid.
[15:6]	Reserved Reserved.
[n] n=0,1..5	CTRLDN Center Re-load Each bit n controls the corresponding BPWM channel n. In up-down counter type, PERIOD will load to PBUF at the end point of each period. CMPDAT will load to CMPBUF at the center point of a period.

BPWM Control Register 1 (BPWM_CTL1)

Register	Offset	R/W	Description	Reset Value
BPWM_CTL1	BPWM_BA+0x04	R/W	BPWM Control Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved						CNTTYPE0	

Bits	Description	
[31:2]	Reserved	Reserved.
[1:0]	CNTTYPE0	BPWM Counter Behavior Type 0 Each bit n controls corresponding BPWM channel n. 00 = Up counter type (supports in capture mode). 01 = Down count type (supports in capture mode). 10 = Up-down counter type. 11 = Reserved.

BPWM Clock Source Register (BPWM_CLKSRC)

Register	Offset	R/W	Description	Reset Value
BPWM_CLKSRC	BPWM_BA+0x10	R/W	BPWM Clock Source Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved					ECLKSRC0		

Bits	Description	
[31:3]	Reserved	Reserved.
[2:0]	ECLKSRC0	BPWM_CH01 External Clock Source Select 000 = BPWMx_CLK, x denotes 0 or 1. 001 = TIMER0 overflow. 010 = TIMER1 overflow. 011 = TIMER2 overflow. 100 = TIMER3 overflow. Others = Reserved.

BPWM Clock Prescale Register (BPWM_CLKPSC)

Register	Offset	R/W	Description	Reset Value
BPWM_CLKPSC	BPWM_BA+0x14	R/W	BPWM Clock Prescale Register	0x0000_0000

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

Bits	Description
[31:12]	Reserved Reserved.
[11:0]	CLKPSC BPWM Counter Clock Prescale The clock of BPWM counter is decided by clock prescaler. Each BPWM pair share one BPWM counter clock prescaler. The clock of BPWM counter is divided by (CLKPSC+ 1).

BPWM Counter Enable Register (BPWM_CNTEN)

Register	Offset	R/W	Description	Reset Value
BPWM_CNTEN	BPWM_BA+0x20	R/W	BPWM Counter Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							CNTEN0

Bits	Description	
[31:1]	Reserved	Reserved.
[0]	CNTEN0	BPWM Counter 0 Enable Bit 0 = BPWM Counter and clock prescaler stop running. 1 = BPWM Counter and clock prescaler start running.

BPWM Clear Counter Register (BPWM_CNTCLR)

Register	Offset	R/W	Description	Reset Value
BPWM_CNTCLR	BPWM_BA+0x24	R/W	BPWM Clear Counter Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							CNTCLR0

Bits	Description	
[31:1]	Reserved	Reserved.
[0]	CNTCLR0	Clear BPWM Counter Control Bit 0 0 = No effect. 1 = Clear 16-bit BPWM counter to 0000H. Note: It is automatically cleared by hardware.

BPWM Period Register (BPWM_PERIOD)

Register	Offset	R/W	Description	Reset Value
BPWM_PERIOD	BPWM_BA+0x30	R/W	BPWM Period Register	0x0000_0000

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

Bits	Description
[31:16]	Reserved Reserved.
[15:0]	PERIOD BPWM Period Register Up-Count mode: In this mode, BPWM counter counts from 0 to PERIOD, and restarts from 0. Down-Count mode: In this mode, BPWM counter counts from PERIOD to 0, and restarts from PERIOD. BPWM period time = (PERIOD+1) * BPWM_CLK period. Up-Down-Count mode: In this mode, BPWM counter counts from 0 to PERIOD, then decrements to 0 and repeats again. BPWM period time = 2 * PERIOD * BPWM_CLK period.

BPWM Comparator Register 0~5 (BPWM_CMPDAT0~5)

Register	Offset	R/W	Description	Reset Value
BPWM_CMPDAT0	BPWM_BA+0x50	R/W	BPWM Comparator Register 0	0x0000_0000
BPWM_CMPDAT1	BPWM_BA+0x54	R/W	BPWM Comparator Register 1	0x0000_0000
BPWM_CMPDAT2	BPWM_BA+0x58	R/W	BPWM Comparator Register 2	0x0000_0000
BPWM_CMPDAT3	BPWM_BA+0x5C	R/W	BPWM Comparator Register 3	0x0000_0000
BPWM_CMPDAT4	BPWM_BA+0x60	R/W	BPWM Comparator Register 4	0x0000_0000
BPWM_CMPDAT5	BPWM_BA+0x64	R/W	BPWM Comparator Register 5	0x0000_0000

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

Bits	Description
[31:16]	Reserved Reserved.
[15:0]	CMPDAT BPWM Comparator Register CMPDAT use to compare with CNT to generate BPWM waveform, interrupt and trigger EADC. In independent mode, CMPDAT0~5 denote as 6 independent BPWM_CH0~5 compared point.

BPWM Counter Register (BPWM_CNT)

Register	Offset	R/W	Description	Reset Value
BPWM_CNT	BPWM_BA+0x90	R	BPWM Counter Register	0x0000_0000

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

Bits	Description	
[31:17]	Reserved	Reserved.
[16]	DIRF	BPWM Direction Indicator Flag (Read Only) 0 = Counter is down counting. 1 = Counter is up counting.
[15:0]	CNT	BPWM Data Register (Read Only) Monitor CNT to know the current value in 16-bit period counter.

BPWM Generation Register 0 (BPWM_WGCTL0)

Register	Offset	R/W	Description	Reset Value
BPWM_WGCTL0	BPWM_BA+0xB0	R/W	BPWM Generation Register 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				PRDPCTL5		PRDPCTL4	
23	22	21	20	19	18	17	16
PRDPCTL3		PRDPCTL2		PRDPCTL1		PRDPCTL0	
15	14	13	12	11	10	9	8
Reserved				ZPCTL5		ZPCTL4	
7	6	5	4	3	2	1	0
ZPCTL3		ZPCTL2		ZPCTL1		ZPCTL0	

Bits	Description	
[31:28]	Reserved	Reserved.
[16+2n+1:16+2n] n=0,1..5	PRDPCTLn	BPWM Period or Center Point Control Each bit n controls the corresponding BPWM channel n. 00 = Do nothing. 01 = BPWM period (center) point output Low. 10 = BPWM period (center) point output High. 11 = BPWM period (center) point output Toggle. BPWM can control output level when BPWM counter count to (PERIOD+1). Note: This bit is center point control when BPWM counter operating in up-down counter type.
[15:12]	Reserved	Reserved.
[2n+1:2n] n=0,1..5	ZPCTLn	BPWM Zero Point Control Each bit n controls the corresponding BPWM channel n. 00 = Do nothing. 01 = BPWM zero point output Low. 10 = BPWM zero point output High. 11 = BPWM zero point output Toggle. Note: BPWM can control output level when BPWM counter counts to 0.

BPWM Generation Register 1 (BPWM_WGCTL1)

Register	Offset	R/W	Description	Reset Value
BPWM_WGCTL1	BPWM_BA+0xB4	R/W	BPWM Generation Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				CMPDCTL5		CMPDCTL4	
23	22	21	20	19	18	17	16
CMPDCTL3		CMPDCTL2		CMPDCTL1		CMPDCTL0	
15	14	13	12	11	10	9	8
Reserved				CMPUCTL5		CMPUCTL4	
7	6	5	4	3	2	1	0
CMPUCTL3		CMPUCTL2		CMPUCTL1		CMPUCTL0	

Bits	Description	
[31:28]	Reserved	Reserved.
[16+2n+1:16+2n] n=0,1..5	CMPDCTLn	BPWM Compare Down Point Control Each bit n controls the corresponding BPWM channel n. 00 = Do nothing. 01 = BPWM compare down point output Low. 10 = BPWM compare down point output High. 11 = BPWM compare down point output Toggle. Note: BPWM can control output level when BPWM counter down counts to CMPDAT.
[15:12]	Reserved	Reserved.
[2n+1:2n] n=0,1..5	CMPUCTLn	BPWM Compare Up Point Control Each bit n controls the corresponding BPWM channel n. 00 = Do nothing. 01 = BPWM compare up point output Low. 10 = BPWM compare up point output High. 11 = BPWM compare up point output Toggle. Note: BPWM can control output level when BPWM counter up counts to CMPDAT.

BPWM Mask Enable Register (BPWM_MSKEN)

Register	Offset	R/W	Description	Reset Value
BPWM_MSKEN	BPWM_BA+0xB8	R/W	BPWM Mask Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		MSKEN5	MSKEN4	MSKEN3	MSKEN2	MSKEN1	MSKEN0

Bits	Description
[31:6]	Reserved Reserved.
[n] n=0,1..5	MSKENn BPWM Mask Enable Bits Each bit n controls the corresponding BPWM channel n. The BPWM output signal will be masked when this bit is enabled. The corresponding BPWM channel n will output MSKDATn (BPWM_MSK[5:0]) data. 0 = BPWM output signal is non-masked. 1 = BPWM output signal is masked and output MSKDATn data.

BPWM Mask DATA Register (BPWM_MSK)

Register	Offset	R/W	Description	Reset Value
BPWM_MSK	BPWM_BA+0xBC	R/W	BPWM Mask Data Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		MSKDAT5	MSKDAT4	MSKDAT3	MSKDAT2	MSKDAT1	MSKDAT0

Bits	Description	
[31:6]	Reserved	Reserved.
[n] n=0,1..5	MSKDATn	BPWM Mask Data Bit This data bit control the state of BPWMn output pin if the corresponding mask function is enabled. Each bit n controls the corresponding BPWM channel n. 0 = Output logic low to BPWMn. 1 = Output logic high to BPWMn.

BPWM Pin Polar Inverse Control (BPWM_POLCTL)

Register	Offset	R/W	Description	Reset Value
BPWM_POLCTL	BPWM_BA+0xD4	R/W	BPWM Pin Polar Inverse Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		PINV5	PINV4	PINV3	PINV2	PINV1	PINV0

Bits	Description	
[31:6]	Reserved	Reserved.
[n] n=0,1..5	PINVn	BPWM PIN Polar Inverse Control The register controls polarity state of BPWM output pin. Each bit n controls the corresponding BPWM channel n. 0 = BPWM output pin polar inverse Disabled. 1 = BPWM output pin polar inverse Enabled.

BPWM Output Enable Register (BPWM_POEN)

Register	Offset	R/W	Description	Reset Value
BPWM_POEN	BPWM_BA+0xD8	R/W	BPWM Output Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		POEN5	POEN4	POEN3	POEN2	POEN1	POEN0

Bits	Description	
[31:6]	Reserved	Reserved.
[n] n=0,1..5	POENn	BPWM Pin Output Enable Bits Each bit n controls the corresponding BPWM channel n. 0 = BPWM pin at tri-state. 1 = BPWM pin in output mode.

BPWM Interrupt Enable Register (BPWM_INTEN)

Register	Offset	R/W	Description	Reset Value
BPWM_INTEN	BPWM_BA+0xE0	R/W	BPWM Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		CMPDIEN5	CMPDIEN4	CMPDIEN3	CMPDIEN2	CMPDIEN1	CMPDIEN0
23	22	21	20	19	18	17	16
Reserved		CMPUIEN5	CMPUIEN4	CMPUIEN3	CMPUIEN2	CMPUIEN1	CMPUIEN0
15	14	13	12	11	10	9	8
Reserved							PIEN0
7	6	5	4	3	2	1	0
Reserved							ZIEN0

Bits	Description	
[31:30]	Reserved	Reserved.
[24+n] n=0,1..5	CMPDIENn	BPWM Compare Down Count Interrupt Enable Bits Each bit n controls the corresponding BPWM channel n. 0 = Compare down count interrupt Disabled. 1 = Compare down count interrupt Enabled.
[23:22]	Reserved	Reserved.
[16+n] n=0,1..5	CMPUIENn	BPWM Compare Up Count Interrupt Enable Bits Each bit n controls the corresponding BPWM channel n. 0 = Compare up count interrupt Disabled. 1 = Compare up count interrupt Enabled.
[15:9]	Reserved	Reserved.
[8]	PIEN0	BPWM Period Point Interrupt 0 Enable Bit 0 = Period point interrupt Disabled. 1 = Period point interrupt Enabled. Note: Up-down counter type period point means center point.
[7:1]	Reserved	Reserved.
[0]	ZIEN0	BPWM Zero Point Interrupt 0 Enable Bit 0 = Zero point interrupt Disabled. 1 = Zero point interrupt Enabled.

BPWM Interrupt Flag Register (BPWM_INTSTS)

Register	Offset	R/W	Description	Reset Value
BPWM_INTSTS	BPWM_BA+0xE8	R/W	BPWM Interrupt Flag Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		CMPDIF5	CMPDIF4	CMPDIF3	CMPDIF2	CMPDIF1	CMPDIF0
23	22	21	20	19	18	17	16
Reserved		CMPUIF5	CMPUIF4	CMPUIF3	CMPUIF2	CMPUIF1	CMPUIF0
15	14	13	12	11	10	9	8
Reserved							PIF0
7	6	5	4	3	2	1	0
Reserved							ZIF0

Bits	Description
[31:30]	Reserved Reserved.
[24+n] n=0,1..5	CMPDIFn BPWM Compare Down Count Interrupt Flag Each bit n controls the corresponding BPWM channel n. Flag is set by hardware when BPWM counter down counts and reaches BPWM_CMPDATn; software can clear this bit by writing 1 to it. Note: If CMPDAT is equal to PERIOD, this flag does not work in down counter type selection.
[23:22]	Reserved Reserved.
[16+n] n=0,1..5	CMPUIFn BPWM Compare Up Count Interrupt Flag Flag is set by hardware when BPWM counter up counts and reaches BPWM_CMPDATn; software can clear this bit by writing 1 to it. Each bit n controls the corresponding BPWM channel n. Note: If CMPDAT is equal to PERIOD, this flag does not work in up counter type selection.
[15:9]	Reserved Reserved.
[8]	PIF0 BPWM Period Point Interrupt Flag 0 This bit is set by hardware when BPWM_CH0 counter reaches BPWM_PERIOD0; software can write 1 to clear this bit to 0.
[7:1]	Reserved Reserved.
[0]	ZIF0 BPWM Zero Point Interrupt Flag 0 This bit is set by hardware when BPWM_CH0 counter reaches 0; software can write 1 to clear this bit to 0.

BPWM Trigger EADC Source Select Register 0 (BPWM_EADCTS0)

Register	Offset	R/W	Description	Reset Value
BPWM_EADCTS0	BPWM_BA+0xF8	R/W	BPWM Trigger EADC Source Select Register 0	0x0000_0000

31	30	29	28	27	26	25	24
TRGEN3	Reserved			TRGSEL3			
23	22	21	20	19	18	17	16
TRGEN2	Reserved			TRGSEL2			
15	14	13	12	11	10	9	8
TRGEN1	Reserved			TRGSEL1			
7	6	5	4	3	2	1	0
TRGEN0	Reserved			TRGSEL0			

Bits	Description	
[31]	TRGEN3	BPWM_CH3 Trigger EADC Enable Bit 0 = BPWM Channel 3 Trigger EADC function Disabled. 1 = BPWM Channel 3 Trigger EADC function Enabled.
[30:28]	Reserved	Reserved.
[27:24]	TRGSEL3	BPWM_CH3 Trigger EADC Source Select 0000 = BPWM_CH2 zero point. 0001 = BPWM_CH2 period point. 0010 = BPWM_CH2 zero or period point. 0011 = BPWM_CH2 up-count CMPDAT point. 0100 = BPWM_CH2 down-count CMPDAT point. 0101 = Reserved. 0110 = Reserved. 0111 = Reserved. 1000 = BPWM_CH3 up-count CMPDAT point. 1001 = BPWM_CH3 down-count CMPDAT point. Others reserved.
[23]	TRGEN2	BPWM_CH2 Trigger EADC Enable Bit 0 = BPWM Channel 2 Trigger EADC function Disabled. 1 = BPWM Channel 2 Trigger EADC function Enabled.
[22:20]	Reserved	Reserved.
[19:16]	TRGSEL2	BPWM_CH2 Trigger EADC Source Select 0000 = BPWM_CH2 zero point. 0001 = BPWM_CH2 period point. 0010 = BPWM_CH2 zero or period point. 0011 = BPWM_CH2 up-count CMPDAT point. 0100 = BPWM_CH2 down-count CMPDAT point. 0101 = Reserved.

		0110 = Reserved. 0111 = Reserved. 1000 = BPWM_CH3 up-count CMPDAT point. 1001 = BPWM_CH3 down-count CMPDAT point. Others reserved
[15]	TRGEN1	BPWM_CH1 Trigger EADC Enable Bit 0 = BPWM Channel 1 Trigger EADC function Disabled. 1 = BPWM Channel 1 Trigger EADC function Enabled.
[14:12]	Reserved	Reserved.
[11:8]	TRGSEL1	BPWM_CH1 Trigger EADC Source Select 0000 = BPWM_CH0 zero point. 0001 = BPWM_CH0 period point. 0010 = BPWM_CH0 zero or period point. 0011 = BPWM_CH0 up-count CMPDAT point. 0100 = BPWM_CH0 down-count CMPDAT point. 0101 = Reserved. 0110 = Reserved. 0111 = Reserved. 1000 = BPWM_CH1 up-count CMPDAT point. 1001 = BPWM_CH1 down-count CMPDAT point. Others reserved
[7]	TRGEN0	BPWM_CH0 Trigger EADC Enable Bit 0 = BPWM Channel 0 Trigger EADC function Disabled. 1 = BPWM Channel 0 Trigger EADC function Enabled.
[6:4]	Reserved	Reserved.
[3:0]	TRGSEL0	BPWM_CH0 Trigger EADC Source Select 0000 = BPWM_CH0 zero point. 0001 = BPWM_CH0 period point. 0010 = BPWM_CH0 zero or period point. 0011 = BPWM_CH0 up-count CMPDAT point. 0100 = BPWM_CH0 down-count CMPDAT point. 0101 = Reserved. 0110 = Reserved. 0111 = Reserved. 1000 = BPWM_CH1 up-count CMPDAT point. 1001 = BPWM_CH1 down-count CMPDAT point. Others reserved

BPWM Trigger EADC Source Select Register 1 (BPWM_EADCTS1)

Register	Offset	R/W	Description	Reset Value
BPWM_EADCTS1	BPWM_BA+0xFC	R/W	BPWM Trigger EADC Source Select Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
TRGEN5	Reserved			TRGSEL5			
7	6	5	4	3	2	1	0
TRGEN4	Reserved			TRGSEL4			

Bits	Description	
[31:16]	Reserved	Reserved.
[15]	TRGEN5	BPWM_CH5 Trigger EADC Enable Bit 0 = BPWM Channel 5 Trigger EADC function Disabled. 1 = BPWM Channel 5 Trigger EADC function Enabled.
[14:12]	Reserved	Reserved.
[11:8]	TRGSEL5	BPWM_CH5 Trigger EADC Source Select 0000 = BPWM_CH4 zero point. 0001 = BPWM_CH4 period point. 0010 = BPWM_CH4 zero or period point. 0011 = BPWM_CH4 up-count CMPDAT point. 0100 = BPWM_CH4 down-count CMPDAT point. 0101 = Reserved. 0110 = Reserved. 0111 = Reserved. 1000 = BPWM_CH5 up-count CMPDAT point. 1001 = BPWM_CH5 down-count CMPDAT point. Others reserved.
[7]	TRGEN4	BPWM_CH4 Trigger EADC Enable Bit 0 = BPWM Channel 4 Trigger EADC function Disabled. 1 = BPWM Channel 4 Trigger EADC function Enabled.
[6:4]	Reserved	Reserved.
[3:0]	TRGSEL4	BPWM_CH4 Trigger EADC Source Select 0000 = BPWM_CH4 zero point. 0001 = BPWM_CH4 period point. 0010 = BPWM_CH4 zero or period point. 0011 = BPWM_CH4 up-count CMPDAT point.

		0100 = BPWM_CH4 down-count CMPDAT point. 0101 = Reserved. 0110 = Reserved. 0111 = Reserved. 1000 = BPWM_CH5 up-count CMPDAT point. 1001 = BPWM_CH5 down-count CMPDAT point. Others reserved.

BPWM Status Register (BPWM_STATUS)

Register	Offset	R/W	Description	Reset Value
BPWM_STATUS	BPWM_BA+0x120	R/W	BPWM Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved		EADCTRG5	EADCTRG4	EADCTRG3	EADCTRG2	EADCTRG1	EADCTRG0
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							CNTMAX0

Bits	Description	
[31:22]	Reserved	Reserved.
[16+n] n=0,1..5	EADCTRGn	EADC Start of Conversion Status Each bit n controls the corresponding BPWM channel n. 0 = No EADC start of conversion trigger event occurred. 1 = An EADC start of conversion trigger event occurred. Note: This bit can be cleared by software write 1.
[15:1]	Reserved	Reserved.
[0]	CNTMAX0	Time-base Counter 0 Equal to 0xFFFF Latched Status 0 = The time-base counter never reached its maximum value 0xFFFF. 1 = The time-base counter reached its maximum value. Note: This bit can be cleared by software write 1.

BPWM Capture Input Enable Register (BPWM_CAPINEN)

Register	Offset	R/W	Description	Reset Value
BPWM_CAPINEN	BPWM_BA+0x200	R/W	BPWM Capture Input Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		CAPINEN5	CAPINEN4	CAPINEN3	CAPINEN2	CAPINEN1	CAPINEN0

Bits	Description	
[31:6]	Reserved	Reserved.
[n] n=0,1..5	CAPINENn	Capture Input Enable Bits Each bit n controls the corresponding BPWM channel n. 0 = BPWM Channel capture input path Disabled. The input of BPWM channel capture function is always regarded as 0. 1 = BPWM Channel capture input path Enabled. The input of BPWM channel capture function comes from correlative multifunction pin.

BPWM Capture Control Register (BPWM_CAPCTL)

Register	Offset	R/W	Description	Reset Value
BPWM_CAPCTL	BPWM_BA+0x204	R/W	BPWM Capture Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		FCRLDEN5	FCRLDEN4	FCRLDEN3	FCRLDEN2	FCRLDEN1	FCRLDEN0
23	22	21	20	19	18	17	16
Reserved		RCRLDEN5	RCRLDEN4	RCRLDEN3	RCRLDEN2	RCRLDEN1	RCRLDEN0
15	14	13	12	11	10	9	8
Reserved		CAPINV5	CAPINV4	CAPINV3	CAPINV2	CAPINV1	CAPINV0
7	6	5	4	3	2	1	0
Reserved		CAPEN5	CAPEN4	CAPEN3	CAPEN2	CAPEN1	CAPEN0

Bits	Description
[31:30]	Reserved Reserved.
[24+n] n=0,1..5	FCRLDENn Falling Capture Reload Enable Bits Each bit n controls the corresponding BPWM channel n. 0 = Falling capture reload counter Disabled. 1 = Falling capture reload counter Enabled.
[23:22]	Reserved Reserved.
[16+n] n=0,1..5	RCRLDENn Rising Capture Reload Enable Bits Each bit n controls the corresponding BPWM channel n. 0 = Rising capture reload counter Disabled. 1 = Rising capture reload counter Enabled.
[15:14]	Reserved Reserved.
[8+n] n=0,1..5	CAPINVn Capture Inverter Enable Bits Each bit n controls the corresponding BPWM channel n. 0 = Capture source inverter Disabled. 1 = Capture source inverter Enabled. Reverse the input signal from GPIO.
[7:6]	Reserved Reserved.
[n] n=0,1..5	CAPENn Capture Function Enable Bits Each bit n controls the corresponding BPWM channel n. 0 = Capture function Disabled. RCAPDAT/FCAPDAT register will not be updated. 1 = Capture function Enabled. Capture latched the BPWM counter value when detected rising or falling edge of input signal and saved to RCAPDAT (Rising latch) and FCAPDAT (Falling latch).

BPWM Capture Status Register (BPWM CAPSTS)

Register	Offset	R/W	Description	Reset Value
BPWM_CAPSTS	BPWM_BA+0x208	R	BPWM Capture Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		CFIFOV5	CFIFOV4	CFIFOV3	CFIFOV2	CFIFOV1	CFIFOV0
7	6	5	4	3	2	1	0
Reserved		CRIFOV5	CRIFOV4	CRIFOV3	CRIFOV2	CRIFOV1	CRIFOV0

Bits	Description
[31:14]	Reserved Reserved.
[8+n] n=0,1..5	CFIFOVn Capture Falling Interrupt Flag Overrun Status (Read Only) This flag indicates if falling latch happened when the corresponding CAPFIF is 1. Each bit n controls the corresponding BPWM channel n. Note: This bit will be cleared automatically when the corresponding CAPFIF is cleared.
[7:6]	Reserved Reserved.
[n] n=0,1..5	CRIFOVn Capture Rising Interrupt Flag Overrun Status (Read Only) This flag indicates if rising latch happened when the corresponding CAPRIF is 1. Each bit n controls the corresponding BPWM channel n. Note: This bit will be cleared automatically when the corresponding CAPRIF is cleared.

BPWM Rising Capture Data Register 0~5 (BPWM_RCAPDAT 0~5)

Register	Offset	R/W	Description	Reset Value
BPWM_RCAPDAT0	BPWM_BA+0x20C	R	BPWM Rising Capture Data Register 0	0x0000_0000
BPWM_RCAPDAT1	BPWM_BA+0x214	R	BPWM Rising Capture Data Register 1	0x0000_0000
BPWM_RCAPDAT2	BPWM_BA+0x21C	R	BPWM Rising Capture Data Register 2	0x0000_0000
BPWM_RCAPDAT3	BPWM_BA+0x224	R	BPWM Rising Capture Data Register 3	0x0000_0000
BPWM_RCAPDAT4	BPWM_BA+0x22C	R	BPWM Rising Capture Data Register 4	0x0000_0000
BPWM_RCAPDAT5	BPWM_BA+0x234	R	BPWM Rising Capture Data Register 5	0x0000_0000

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

Bits	Description
[31:16]	Reserved Reserved.
[15:0]	RCAPDAT BPWM Rising Capture Data (Read Only) When rising capture condition happened, the BPWM counter value will be saved in this register.

BPWM Falling Capture Data Register 0~5 (BPWM FCAPDAT 0~5)

Register	Offset	R/W	Description	Reset Value
BPWM_FCAPDAT0	BPWM_BA+0x210	R	BPWM Falling Capture Data Register 0	0x0000_0000
BPWM_FCAPDAT1	BPWM_BA+0x218	R	BPWM Falling Capture Data Register 1	0x0000_0000
BPWM_FCAPDAT2	BPWM_BA+0x220	R	BPWM Falling Capture Data Register 2	0x0000_0000
BPWM_FCAPDAT3	BPWM_BA+0x228	R	BPWM Falling Capture Data Register 3	0x0000_0000
BPWM_FCAPDAT4	BPWM_BA+0x230	R	BPWM Falling Capture Data Register 4	0x0000_0000
BPWM_FCAPDAT5	BPWM_BA+0x238	R	BPWM Falling Capture Data Register 5	0x0000_0000

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

Bits	Description
[31:16]	Reserved Reserved.
[15:0]	FCAPDAT BPWM Falling Capture Data (Read Only) When falling capture condition happened, the BPWM counter value will be saved in this register.

BPWM Capture Interrupt Enable Register (BPWM_CAPIEN)

Register	Offset	R/W	Description	Reset Value
BPWM_CAPIEN	BPWM_BA+0x250	R/W	BPWM Capture Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		CAPFIEN5	CAPFIEN4	CAPFIEN3	CAPFIEN2	CAPFIEN1	CAPFIEN0
7	6	5	4	3	2	1	0
Reserved		CAPRIEN5	CAPRIEN4	CAPRIEN3	CAPRIEN2	CAPRIEN1	CAPRIEN0

Bits	Description	
[31:14]	Reserved	Reserved.
[13:8]	CAPFIENn	BPWM Capture Falling Latch Interrupt Enable Bits Each bit n controls the corresponding BPWM channel n. 0 = Capture falling edge latch interrupt Disabled. 1 = Capture falling edge latch interrupt Enabled.
[7:6]	Reserved	Reserved.
[5:0]	CAPRIENn	BPWM Capture Rising Latch Interrupt Enable Bits Each bit n controls the corresponding BPWM channel n. 0 = Capture rising edge latch interrupt Disabled. 1 = Capture rising edge latch interrupt Enabled.

BPWM Capture Interrupt Flag Register (BPWM_CAPIF)

Register	Offset	R/W	Description	Reset Value
BPWM_CAPIF	BPWM_BA+0x254	R/W	BPWM Capture Interrupt Flag Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		CAPFIF5	CAPFIF4	CAPFIF3	CAPFIF2	CAPFIF1	CAPFIF0
7	6	5	4	3	2	1	0
Reserved		CAPRIF5	CAPRIF4	CAPRIF3	CAPRIF2	CAPRIF1	CAPRIF0

Bits	Description	
[31:14]	Reserved	Reserved.
[8+n] n=0,1..5	CAPFIFn	BPWM Capture Falling Latch Interrupt Flag Each bit n controls the corresponding BPWM channel n. 0 = No capture falling latch condition happened. 1 = Capture falling latch condition happened, this flag will be set to high. Note: This bit is cleared by writing 1 to it.
[7:6]	Reserved	Reserved.
[n] n=0,1..5	CAPRIFn	BPWM Capture Rising Latch Interrupt Flag Each bit n controls the corresponding BPWM channel n. 0 = No capture rising latch condition happened. 1 = Capture rising latch condition happened, this flag will be set to high. Note: This bit is cleared by writing 1 to it.

BPWM Period Register Buffer (BPWM_PBUF)

Register	Offset	R/W	Description	Reset Value
BPWM_PBUF	BPWM_BA+0x304	R	BPWM PERIOD Buffer	0x0000_0000

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

Bits	Description	
[31:16]	Reserved	Reserved.
[15:0]	PBUF	BPWM Period Buffer (Read Only) Used as PERIOD active register.

BPWM Comparator Register Buffer 0~5 (BPWM_CMPBUF0~5)

Register	Offset	R/W	Description	Reset Value
BPWM_CMPBUF0	BPWM_BA+0x31C	R	BPWM CMPDAT 0 Buffer	0x0000_0000
BPWM_CMPBUF1	BPWM_BA+0x320	R	BPWM CMPDAT 1 Buffer	0x0000_0000
BPWM_CMPBUF2	BPWM_BA+0x324	R	BPWM CMPDAT 2 Buffer	0x0000_0000
BPWM_CMPBUF3	BPWM_BA+0x328	R	BPWM CMPDAT 3 Buffer	0x0000_0000
BPWM_CMPBUF4	BPWM_BA+0x32C	R	BPWM CMPDAT 4 Buffer	0x0000_0000
BPWM_CMPBUF5	BPWM_BA+0x330	R	BPWM CMPDAT 5 Buffer	0x0000_0000

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

Bits	Description	
[31:16]	Reserved	Reserved.
[15:0]	CMPBUF	BPWM Comparator Buffer (Read Only) Used as CMP active register.

6.12 UART Interface Controller (UART)

6.12.1 Overview

The chip provides five channels of Universal Asynchronous Receiver/Transmitters (UART). The UART controller performs serial-to-parallel conversion on data received from the peripheral and parallel-to-serial conversion on data transmitted from the CPU. Each UART controller channel supports ten types of interrupts. The UART controller supports flow control function. The UART controller also supports IrDA SIR, RS-485, and Single-wire function modes and auto-baud rate measuring function.

6.12.2 Features

- Full-duplex asynchronous communications
- Separates receive and transmit 16/16 bytes or 1/1 byte entry FIFO for data payloads
- Supports hardware auto-flow control
- Programmable receiver buffer trigger level
- Supports programmable baud rate generator for each channel individually
- Supports nCTS, incoming data, Received Data FIFO reached threshold and RS-485 Address Match (AAD mode) wake-up function
- Supports 8-bit receiver buffer time-out detection function
- Programmable transmitting data delay time between the last stop and the next START bit by setting DLY (UART_TOUT [15:8])
- Supports Auto-Baud Rate measurement and baud rate compensation function
 - Support 9600 bps for UART_CLK is selected LXT.
- Supports break error, frame error, parity error and receive/transmit buffer overflow detection function
- Fully programmable serial-interface characteristics
 - Programmable number of data bit, 5-, 6-, 7-, 8- bit character
 - Programmable PARITY bit, even, odd, no parity or stick PARITY bit generation and detection
 - Programmable STOP bit, 1, 1.5, or 2 STOP bit generation
- Supports IrDA SIR function mode
 - Supports for 3/16 bit duration for normal mode
- Supports RS-485 function mode
 - Supports RS-485 9-bit mode
 - Supports hardware or software enables to program nRTS pin to control RS-485 transmission direction
- Supports PDMA transfer function
- Support Single-wire function mode

UART Feature	UART0 ~ UART3	UART4	USCI-UART
FIFO	16 Bytes	1 Bytes	TX: 1byte RX: 2byte
Auto Flow Control (CTS/RTS)	√	√	√
IrDA	√	√	-
LIN	-	-	-
RS-485 Function Mode	√	√	√
nCTS Wake-up	√	-	√
Incoming Data Wake-up	√	√	√
Received Data FIFO reached threshold Wake-up	√	-	-
RS-485 Address Match (AAD mode) Wake-up	√	-	-
Received Data FIFO reached threshold Time-out Wake-up	√	-	
Baud Rate Compensation	√	-	
Auto-Baud Rate Measurement	√	√	√
STOP bit Length	1, 1.5, 2 bit	1, 1.5, 2 bit	1, 2 bit
Word Length	5, 6, 7, 8 bits	5, 6, 7, 8 bits	6~13 bits
Even / Odd Parity	√	√	√
Stick Bit	√	√	-

Table 6.12-1 NuMicro® M253 Series UART Features

6.12.3 Block Diagram

The UART clock control and block diagram are shown in Figure 6.12-1 and Figure 6.12-2 respectively.

Note: The frequency of UARTx_CLK should not be greater than 30 times HCLK.

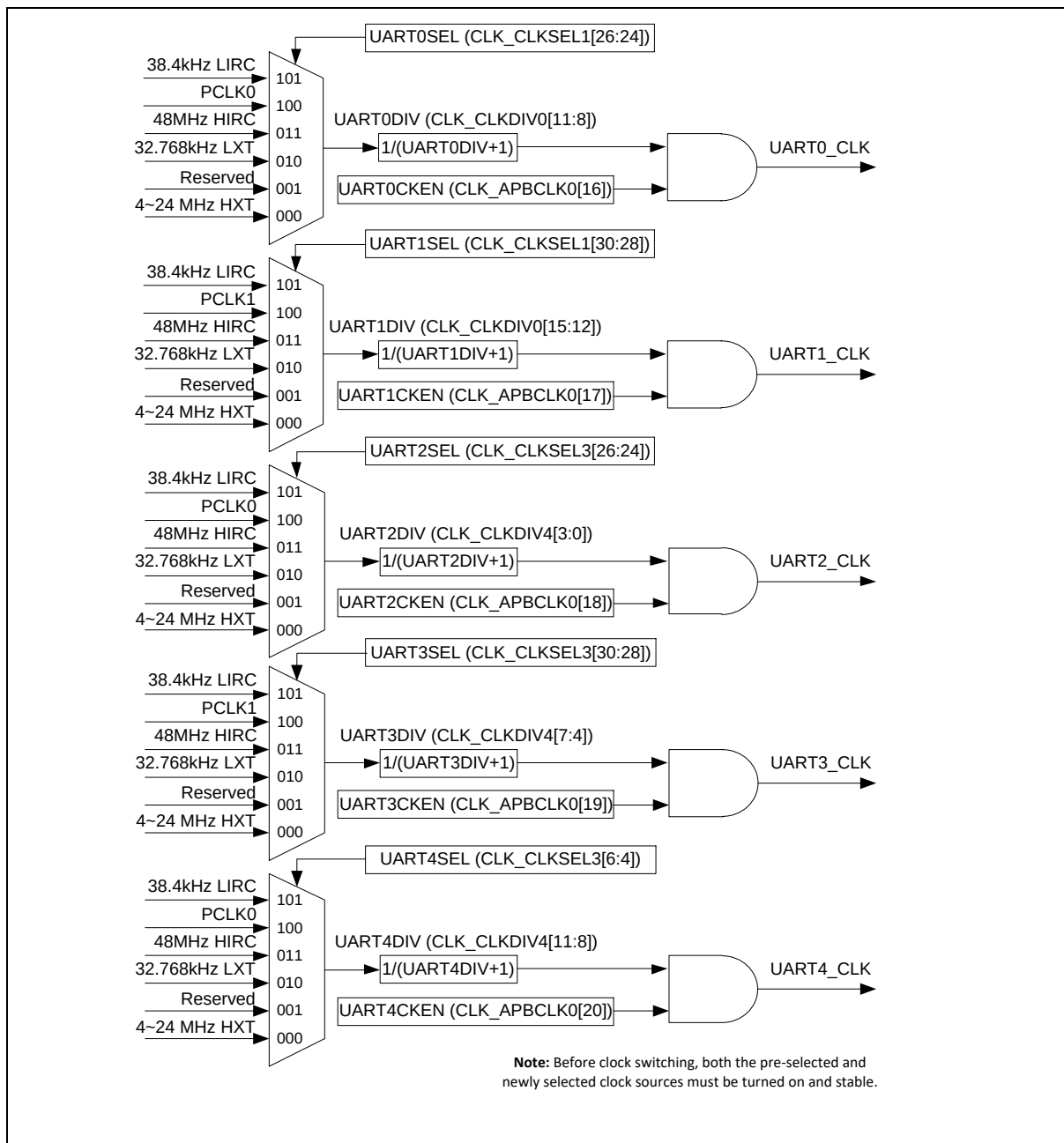


Figure 6.12-1 UART Clock Control Diagram

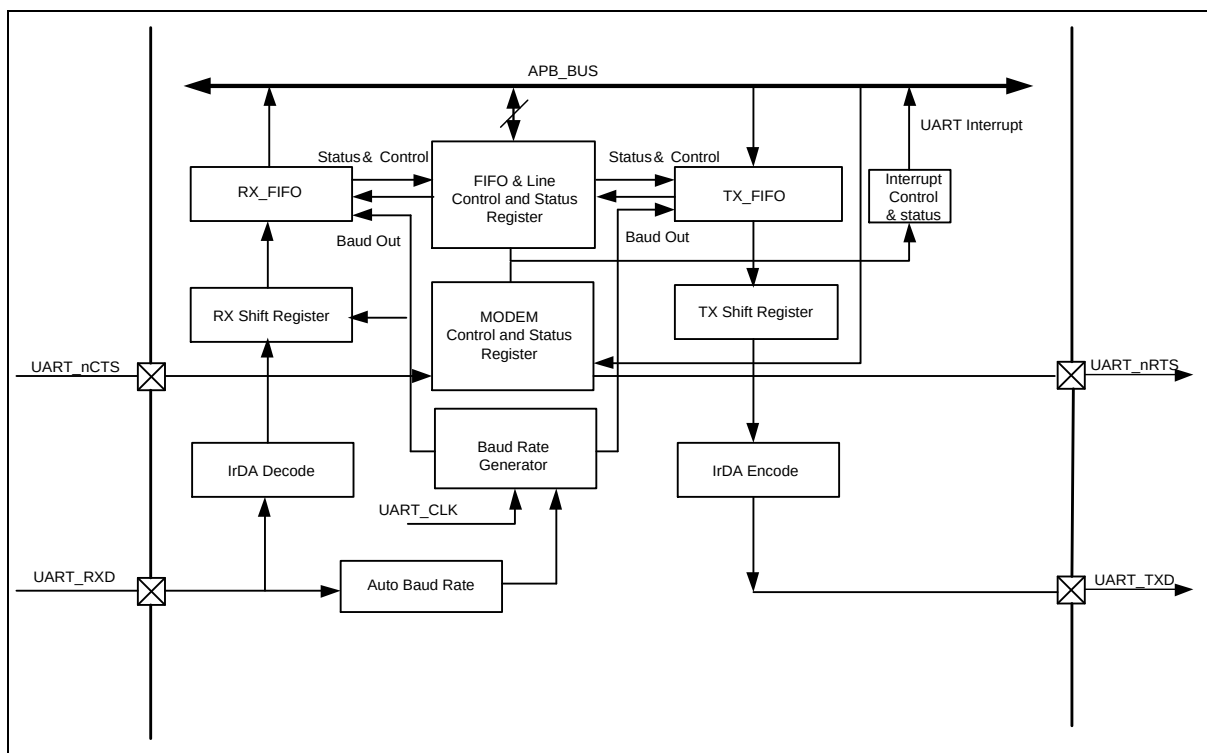


Figure 6.12-2 UART Block Diagram

Each block is described in detail as follows:

TX_FIFO

The transmitter is buffered with a 16 bytes FIFO to reduce the number of interrupts presented to the CPU.

RX_FIFO

The receiver is buffered with a 16 bytes FIFO (plus three error bits, BIF (UART_FIFOSTS[6]), FEF (UART_FIFOSTS[5]), PEF (UART_FIFOSTS[4])) to reduce the number of interrupts presented to the CPU.

TX Shift Register

This block is responsible for shifting out the transmitting data serially.

RX Shift Register

This block is responsible for shifting in the receiving data serially.

Modem Control and Status Register

This register controls the interface to the MODEM or data set (or a peripheral device emulating a MODEM).

Baud Rate Generator

Divide the external clock by the divisor to get the desired baud rate clock. Refer to baud rate equation.

IrDA Encode

This block is IrDA encoding control block.

IrDA Decode

This block is IrDA decoding control block.

FIFO & Line Control and Status Register

This field is register set that including the FIFO control register (UART_FIFO), FIFO status register

(UART_FIFOSTS), and line control register (UART_LINE) for transmitter and receiver. The time-out register (UART_TOUT) identifies the condition of time-out interrupt.

Auto-Baud Rate Measurement

This block is responsible for auto-baud rate measurement.

Interrupt Control and Status Register

There are ten types of interrupts. Interrupt enable register (UART_INTEN) enable or disable the responding interrupt and interrupt status register (UART_INTSTS) identifying the occurrence of the responding interrupt.

Interrupt	Description
RDAINT	Receive Data Available Interrupt.
THREINT	Transmit Holding Register Empty Interrupt.
TXENDINT	Transmitter Empty Interrupt.
RLSINT	Receive Line Status Interrupt (parity error or frame error or break error).
MODEMINT	MODEM Status Interrupt.
RXTOINT	Receiver Buffer Time-out Interrupt.
BUFERRINT	Buffer Error Interrupt.
WKINT	Wake-up Interrupt.
ABRINT	Auto-Baud Rate Interrupt.
SWBEINT	Single-wire Bit Error Detect Interrupt.

Table 6.12-2 UART Interrupt

6.12.4 Basic Configuration

The basic configurations of UART0 are as follows:

- Clock Source Configuration
 - Select the source of UART0 peripheral clock on UART0SEL (CLK_CLKSEL1[26:24]).
 - Select the clock divider number of UART0 peripheral clock on UART0DIV (CLK_CLKDIV0[11:8]).
 - Enable UART0 peripheral clock in UART0CKEN (CLK_APBCLK0[16]).
- Reset UART0 controller in UART0RST (SYS_IPRST1[16]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
UART0	UART0_RXD	PF.2	MFP3
		PF.1	MFP4
		PB.12	MFP6
		PA.0, PA.6	MFP7
		PA.4	MFP8
	UART0_TXD	PF.3	MFP3
		PF.0	MFP4

		PB.13	MFP6
		PA.1, PA.7	MFP7
		PA.5	MFP8
	UART0_nCTS	PB.15	MFP6
		PA.5	MFP7
	UART0_nRTS	PB.14	MFP6
		PA.4	MFP7

The basic configurations of UART1 are as follows:

- Clock Source Configuration
 - Select the source of UART1 peripheral clock on UART1SEL (CLK_CLKSEL1[30:28]).
 - Select the clock divider number of UART1 peripheral clock on UART1DIV (CLK_CLKDIV0[15:12]).
 - Enable UART1 peripheral clock in UART1CKEN (CLK_APBCLK0[17]).
- Reset UART1 controller in UART1RST (SYS_IPRST1[17]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
UART1	UART1_RXD	PF.1	MFP2
		PB.2, PB.6	MFP6
		PA.8	MFP7
		PA.2	MFP8
	UART1_TXD	PF.0	MFP2
		PB.3, PB.7	MFP6
		PA.9	MFP7
		PA.3	MFP8
	UART1_nCTS	PA.1	MFP8
	UART1_nRTS	PA.0	MFP8

The basic configurations of UART2 are as follows:

- Clock Source Configuration
 - Select the source of UART2 peripheral clock on UART2SEL (CLK_CLKSEL3[26:24]).
 - Select clock divider number of UART2 peripheral clock on UART2DIV (CLK_CLKDIV4[3:0]).
 - Enable UART2 peripheral clock in UART2CKEN (CLK_APBCLK0[18]).
- Reset UART2 controller in UART2RST (SYS_IPRST1[18]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
UART2	UART2_RXD	PF.5	MFP2

		PB.0	MFP7
		PC.0, PC.4	MFP8
		PB.4	MFP13
	UART2_TXD	PF.4	MFP2
		PB.1	MFP7
		PC.1, PC.5	MFP8
		PB.5	MFP13
	UART2_nCTS	PF.5	MFP4
		PC.2	MFP8
	UART2_nRTS	PF.4	MFP4
		PC.3	MFP8

The basic configurations of UART3 are as follows:

- Clock Source Configuration
 - Select the source of UART3 peripheral clock on UART3SEL (CLK_CLKSEL3[30:28]).
 - Select the clock divider number of UART3 peripheral clock on UART3DIV (CLK_CLKDIV4[7:4]).
 - Enable UART3 peripheral clock in UART3CKEN (CLK_APBCLK0[19]).
- Reset UART3 controller in UART3RST (SYS_IPRST1[19]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
UART3	UART3_RXD	PB.14	MFP7
		PC.2	MFP11
	UART3_TXD	PB.15	MFP7
		PC.3	MFP11
	UART3_nCTS	PB.12	MFP7
	UART3_nRTS	PB.13	MFP7

The basic configurations of UART4 are as follows:

- Clock Source Configuration
 - Select the source of UART4 peripheral clock on UART4SEL (CLK_CLKSEL3[6:4]).
 - Select the clock divider number of UART4 peripheral clock on UART4DIV (CLK_CLKDIV4[11:8]).
 - Enable UART4 peripheral clock in UART4CKEN (CLK_APBCLK0[20]).
- Reset UART4 controller in UART4RST (SYS_IPRST1[20]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
UART4	UART4_RXD	PA.2	MFP2

		PC.4	MFP11
	UART4_TXD	PA.3	MFP2
		PC.5	MFP11

UART Interface Controller Pin description is shown in Table 6.12-3:

Pin	Type	Description
UARTx_TXD	Output	UARTx transmit
UARTx_RXD	Input	UARTx receive
UARTx_nCTS	Input	UARTx modem clear to send
UARTx_nRTS	Output	UARTx modem request to send

Table 6.12-3 UART Interface Controller Pin

6.12.5 Functional Description

The UART controller supports four function modes including UART, IrDA, RS-485, and Single-wire mode. User can select a function by setting the UART_FUNCSEL register. The four function modes will be described in following section.

6.12.5.1 UART Controller Baud Rate Generator

The UART controller includes a programmable baud rate generator capable of dividing clock input by divisors to produce the serial clock that transmitter and receiver need. Table 6.12-4 list the UART baud rate equations in the various conditions. Table 6.12-5 and Table 6.12-6 list the UART baud rate parameter and register setting example. In IrDA function mode, the baud rate generator must be set in mode 0. More detailed register description is shown in UART_BAUD register. There are three setting modes. Mode 0 is set by UART_BAUD[29:28] with 00. Mode 1 is set by UART_BAUD[29:28] with 10. Mode 2 is set by UART_BAUD[29:28] with 11.

Mode	BAUDM1	BAUDM0	Baud Rate Equation
Mode 0	0	0	$UART\_CLK / [16 * (BRD+2)]$.
Mode 1	1	0	$UART\_CLK / [(EDIVM1+1) * (BRD+2)]$, EDIVM1 must ≥ 8 .
Mode 2	1	1	$UART\_CLK / (BRD+2)$. If $UART\_CLK \leq 3 * HCLK$, BRD must ≥ 8 . If $UART\_CLK > 3 * HCLK$, BRD must $\geq 3 * N - 1$. N is the smallest integer larger than or equal to the ratio of $UART\_CLK / HCLK$. For example, if $3 * HCLK < UART\_CLK \leq 4 * HCLK$, BRD must ≥ 11 . if $4 * HCLK < UART\_CLK \leq 5 * HCLK$, BRD must ≥ 14 . (If the UART_CLK is selected from LXT, BRD can be greater than or equal to 1)

Table 6.12-4 UART controller Baud Rate Equation Table

UART Peripheral Clock = 12 MHz			
Baud Rate	Mode 0	Mode 1	Mode 2
921600	Not support	Not recommended	BRD=11
460800	Not recommended	BRD=0, EDIVM1 =12	BRD=24
230400	Not recommended	BRD =2, EDIVM1 =12	BRD =50
115200	Not recommended	BRD =6, EDIVM1 =12	BRD =102
57600	BRD =11	BRD =14, EDIVM1 =12	BRD =206
38400	BRD =18	BRD =22, EDIVM1 =12	BRD =311
19200	BRD =37	BRD =123, EDIVM1 =4	BRD =623
9600	BRD =76	BRD =123, EDIVM1 =9	BRD =1248
4800	BRD =154	BRD =248, EDIVM1 =9	BRD =2498

Table 6.12-5 UART controller Baud Rate Parameter Setting Example Table

UART Peripheral Clock = 12 MHz			
Baud Rate	UART_BAUD Value		
	Mode 0	Mode 1	Mode 2
921600	Not support	Not recommended	0x3000_000B
460800	Not recommended	0x2C00_0000	0x3000_0018
230400	Not recommended	0x2C00_0002	0x3000_0032
115200	Not recommended	0x2C00_0006	0x3000_0066
57600	0x0000_000B	0x2C00_000E	0x3000_00CE
38400	0x0000_0012	0x2C00_0016	0x3000_0137
19200	0x0000_0025	0x2400_007B	0x3000_026F
9600	0x0000_004C	0x2900_007B	0x3000_04E0
4800	0x0000_009A	0x2900_00F8	0x3000_09C2

Table 6.12-6 UART controller Baud Rate Register Setting Example Table

6.12.5.2 UART Controller Baud Rate Compensation

The UART controller supports baud rate compensation function. It is used to optimize the precision in each bit. The precision of the compensation is half of UART module clock because there is BRCOMPDEC (UART_BRCOMP[31]) to define the positive or negative compensation in each bit. If the BRCOMPDEC (UART_BRCOMP[31]) = 0, it is positive compensation for each bit, one more module clock will be append in the compensated bit. If the BRCOMPDEC (UART_BRCOMP[31]) = 1, it is negative compensation for each bit, decrease one module clock in the compensated bit.

There is 9-bits location, BRCOMP[8:0] (UART_BRCOMP[8:0]), can be configured by user to define the relative bit is compensated or not. BRCOMP[7:0] is used to define the compensation of DAT (UART_DAT[7:0]) and BRCOMP[8] is used to define PARITY (UART_DAT[8]).

Example:

1. UART's peripheral clock = 32.768 kHz and baud rate is 9600 bps

Baud rate is 9600 bps, UART peripheral clock is 32.768 kHz → 3.413 peripheral clock/bit

If the baud divider is set 1 (3 peripheral clock/bit), the inaccuracy of each bit is -0.413 peripheral clock and BRCOMPDEC (UART_BRCOMP[31]) = 0, so that the BRCOMP (UART_BRCOMP[8:0]) can be set as 9'b010100101 = 0xA5.

Bit	Name	Total INACCURACY	BRCOMP Compensated	Final Inaccuracy
0	START	-0.413	x	-0.413
1	UART_DAT[0]	-0.826(-0.413-0.413)	1	0.174
2	UART_DAT[1]	-0.239(0.174-0.413)	0	-0.239
3	UART_DAT[2]	-0.652(-0.239-0.413)	1	0.348
4	UART_DAT[3]	-0.065(0.348-0.413)	0	-0.065
5	UART_DAT[4]	-0.478(-0.065-0.413)	0	-0.478
6	UART_DAT[5]	-0.891(-0.478-0.413)	1	0.109
7	UART_DAT[6]	-0.304(0.109-0.413)	0	-0.304
8	UART_DAT[7]	-0.717(-0.304-0.413)	1	0.283
9	PARITY	-0.130(0.283-0.413)	0	-0.13

Table 6.12-7 Baud Rate Compensation Example Table 1

2. UART's peripheral clock = 32.768 kHz and baud rate is 4800 bps

Baud rate is 4800 bps, UART peripheral clock is 32.768 kHz → 6.827 peripheral clock/bit

If the baud divider is set 5 (7 peripheral clock/bit), the inaccuracy of each bit is 0.173 peripheral clock and BRCOMPDEC (UART_BRCOMP[31]) = 1, so that the BRCOMP (UART_BRCOMP[8:0]) can be set as 9'b010000010 = 0x82.

Bit	Name	Total INACCURACY	BRCOMP Compensated	Final Inaccuracy
0	START	0.173	x	0.173
1	UART_DAT[0]	0.346(0.173+0.173)	0	0.346
2	UART_DAT[1]	0.519(0.346+0.173)	1	-0.481
3	UART_DAT[2]	-0.308(-0.481+0.173)	0	-0.308
4	UART_DAT[3]	-0.135(-0.308+0.173)	0	-0.135
5	UART_DAT[4]	-0.038(-0.135+0.173)	0	0.038
6	UART_DAT[5]	0.211(0.038+0.173)	0	0.211
7	UART_DAT[6]	0.384(0.211+0.173)	0	0.384
8	UART_DAT[7]	0.557(0.384+0.173)	1	-0.443
9	PARITY	-0.270(-0.443+0.173)	0	-0.270

Table 6.12-8 Baud Rate Compensation Example Table 2

UART Controller Auto-Baud Rate Function Mode:

Auto-Baud Rate function can measure baud rate of receiving data from UART RX pin automatically.

When the Auto-Baud Rate measurement is finished, the measuring baud rate is loaded to BRD (UART_BAUD[15:0]). Both of the BAUDM1 (UART_BAUD[29]) and BAUDM0 (UART_BAUD[28]) are set to 1 automatically. UART RX data from START bit to 1st rising edge time is set by $2^{ABRDBITS}$ bit time in Auto-Baud Rate function detection frame.

$2^{ABRDBITS}$ bit time from START bit to the 1st rising edge is calculated by setting ABRDBITS (UART_ALTCTL[20:19]). Setting ABRDEN (UART_ALTCTL[18]) to enable auto-baud rate function. In beginning stage, the UART RX is kept at 1. Once falling edge is detected, START bit is received. The auto-baud rate counter is reset and starts counting. The auto-baud rate counter will be stop when the 1st rising edge is detected. Then, auto-baud rate counter value divided by ABRDBITS (UART_ALTCTL[20:19]) is loaded to BRD (UART_BAUD[15:0]) automatically. ABRDEN (UART_ALTCTL[18]) is cleared. The Auto-Baud is shown in Figure 6.12-3. Once the auto-baud rate measurement is finished, the ABRDIF (UART_FIFOSTS[1]) is set. When auto-baud rate counter is overflow, ABRDTOIF (UART_FIFOSTS[2]) is set. ABRDIF (UART_FIFOSTS[1]) or ABRDTOIF (UART_FIFOSTS[2]) cause the auto-baud rate flag ABRIF (UART_ALTCTL[17]) generated. If the ABRIEN (UART_INTEN[18]) is enabled, ABRIF (UART_ALTCTL[17]) cause the auto-baud rate interrupt ABRINT (UART_INTSTS[31]) generated.

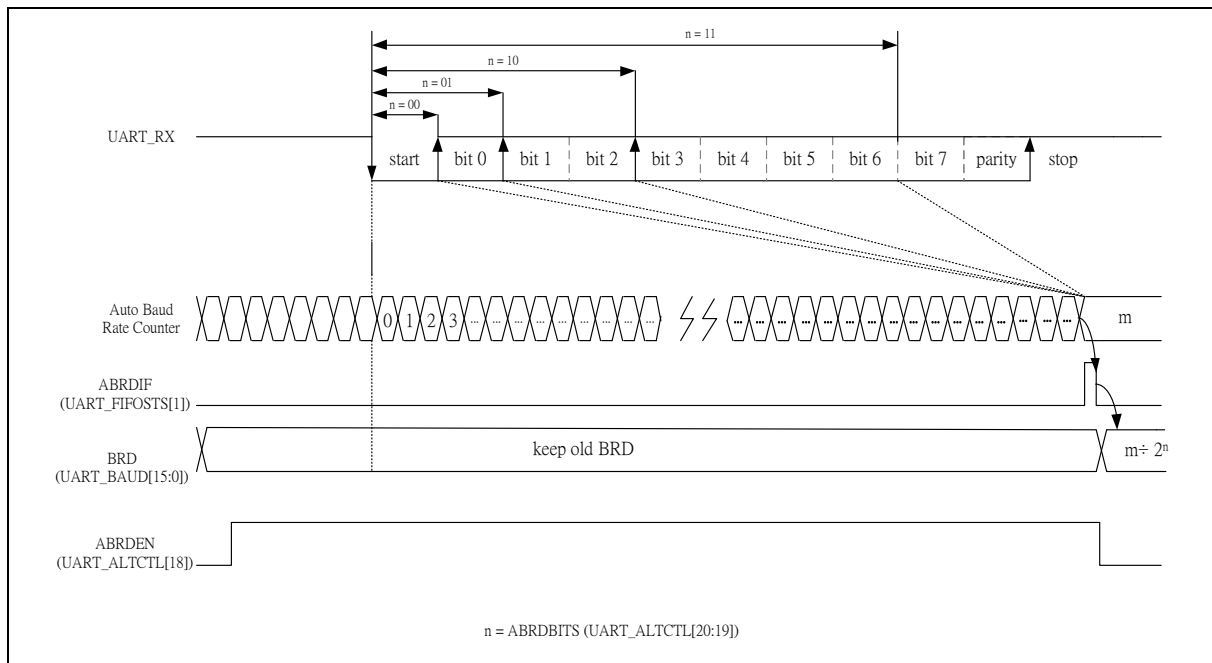


Figure 6.12-3 Auto-Baud Rate Measurement

6.12.5.3 Programming Sequence Example:

1. Program ABRDBITS (UART_ALTCTL[20:19]) to determine UART RX data 1st rising edge time from Start by $2^{ABRDBITS}$ bit time.
2. Set ABRIEN (UART_INTEN[18]) to enable auto-baud rate function interrupt.
3. Set ABRDEN (UART_ALTCTL[18]) to enable auto-baud rate function.
4. ABRDIF (UART_FIFOSTS[1]) is set, the auto-baud rate measurement is finished.
5. Operate UART transmit and receive action.
6. ABRDTOIF (UART_FIFOSTS[2]) is set, if auto-baud rate counter is overflow.
7. Go to Step 3.

6.12.5.4 UART Controller Transmit Delay Time Value

The UART controller programs DLY (UART_TOUT [15:8]) to control the transfer delay time between the last STOP bit and next START bit in transmission. The unit is baud. The operation is shown in Figure 6.12-4.

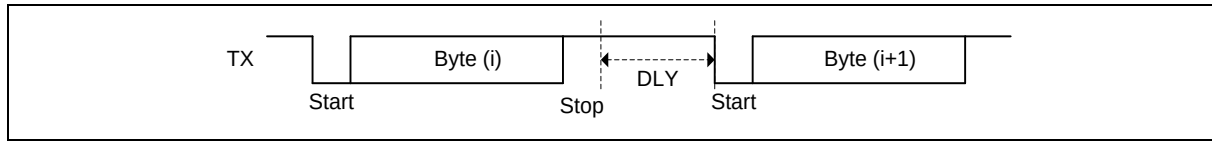


Figure 6.12-4 Transmit Delay Time Operation

6.12.5.5 UART Controller FIFO Control and Status

The UART controller is built-in with a 16 bytes transmitter FIFO (TX_FIFO) and a 16 bytes receiver FIFO (RX_FIFO) that reduces the number of interrupts presented to the CPU. The CPU can read the status of the UART at any time during operation. The reported status information includes condition of the transfer operations being performed by the UART, as well as 3 error conditions (parity error, framing error, break interrupt) occur if receiving data has parity, frame or break error. UART, IrDA and RS-485 mode support FIFO control and status function.

6.12.5.6 UART Controller Wake-up Function

The UART controller supports wake-up system function. The wake-up function includes nCTS pin, incoming data wake-up, Received Data FIFO reached threshold wake-up, RS-485 Address Match (AAD mode) wake-up and Received Data FIFO threshold time-out wake-up function. CTSWKF (UART_WKSTS[0]), DATWKF (UART_WKSTS[1]), RFRTWKF (UART_WKSTS[2]), RS485WKF (UART_WKSTS[3]) or TOUTWKF (UART_WKSTS[4]) cause the wake-up interrupt flag WKIF(UART_INTSTS[6]) generated. If the WKIEN (UART_INTEN[6]) is enabled, the wake-up interrupt flag WKIF(UART_INTSTS[6]) cause the wake-up interrupt WKINT (UART_INTSTS[14]) generated.

nCTS pin wake-up :

When the system is in Power-down mode and WKCTSEN (UART_WKCTL[0]) is set, the toggle of nCTS pin can wake-up system. If the WKCTSEN (UART_WKCTL[0]) is enabled, the toggle of nCTS pin cause the nCTS wake-up flag CTSWKF (UART_WKSTS[0]) generated. The nCTS wake-up is shown in Figure 6.12-5 and Figure 6.12-6.

nCTS Wake-up Case 1 (nCTS transition from low to high)

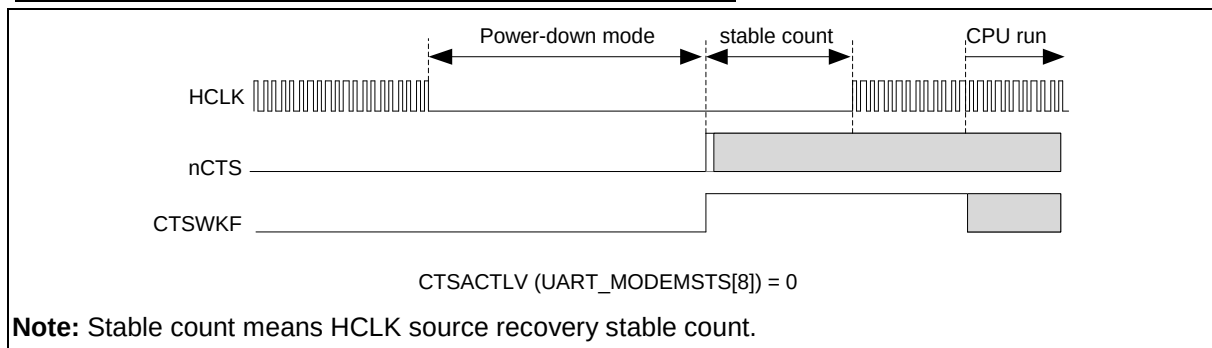


Figure 6.12-5 UART nCTS Wake-up Case1

nCTS Wake-up Case 2 (nCTS transition from high to low)

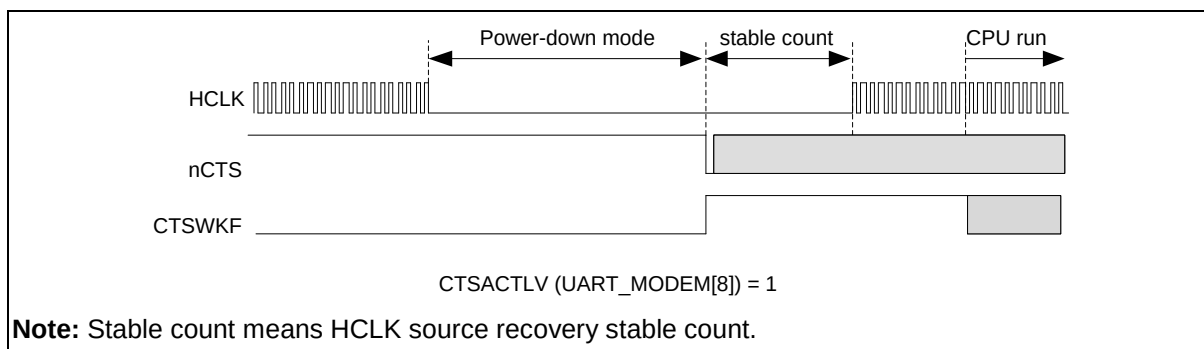


Figure 6.12-6 UART nCTS Wake-up Case2

Incoming Data Wake-up

When system is in Power-down mode and the WKDATEN (UART_WKCTL [1]) is set, the toggle of incoming data (UART_RXD) pin can wake-up the system. In order to receive the incoming data after the system wake-up, the STCOMP (UART_DWKCOMP[15:0]) shall be set. These bits field of STCOMP indicate how many clock cycle selected by UART_CLK do the UART controller can get the 1st bit (START bit) when the system is wake-up from Power-down mode.

When incoming data wakes system up, the incoming data will be received and stored in FIFO. If the WKDATEN (UART_WKCTL[1]) is enabled, the toggle of incoming data (UART_RXD) pin cause the incoming data wake-up flag DATWKF (UART_WKSTS[1]) generated. The incoming data wake-up is shown in Figure 6.12-7.

Note 1: The UART controller clock source should be selected as HIRC, and the compensation time for START bit. Refer to the M253 Datasheet for detailed information about wakeup time electrical characteristics. The STCOMP (UART_DWKCOMP[15:0]) = (wake-up stable time) * (HIRC frequency).

Note 2: The value of BRD (UART_BAUD[15:0]) should be greater than STCOMP (UART_DWKCOMP[15:0]).

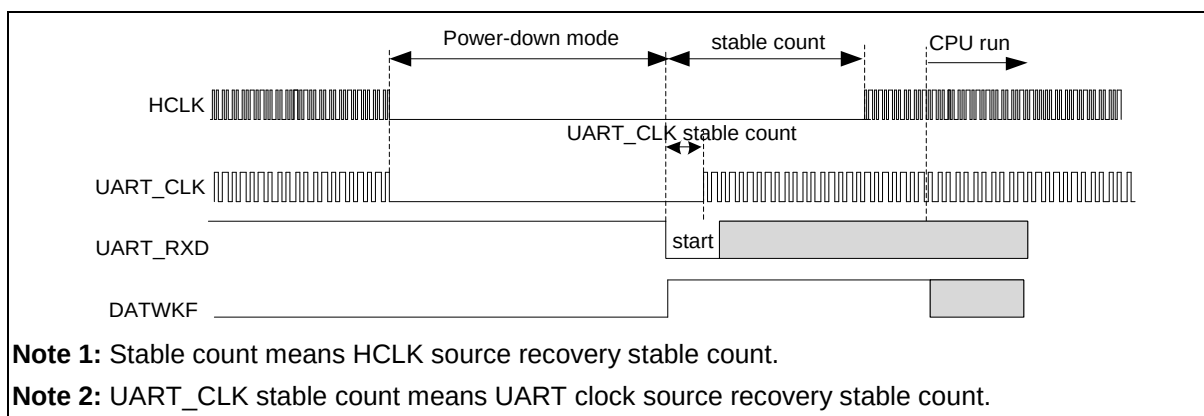


Figure 6.12-7 UART Data Wake-up

Received Data FIFO Reached Threshold Wake-up

The received data FIFO threshold reached wake-up function is enabled by setting WKFRFTEN (UART_WKCTL[2]). In Power-down mode, when the number of received data in RX FIFO reaches the threshold value RFITL (UART_FIFO[7:4]), it can wake-up the system. If the WKFRFTEN (UART_WKCTL[2]) is enabled, the number of received data in RX FIFO reaches the threshold value RFITL (UART_FIFO[7:4]) cause the received data FIFO reached threshold wake-up flag RFRTWK (UART_WKSTS[2]) generated. The Received Data FIFO reached threshold wake-up is shown in Figure 6.12-8.

Note: The UART controller clock source should be selected as LXT in Power-down mode to receive

data.

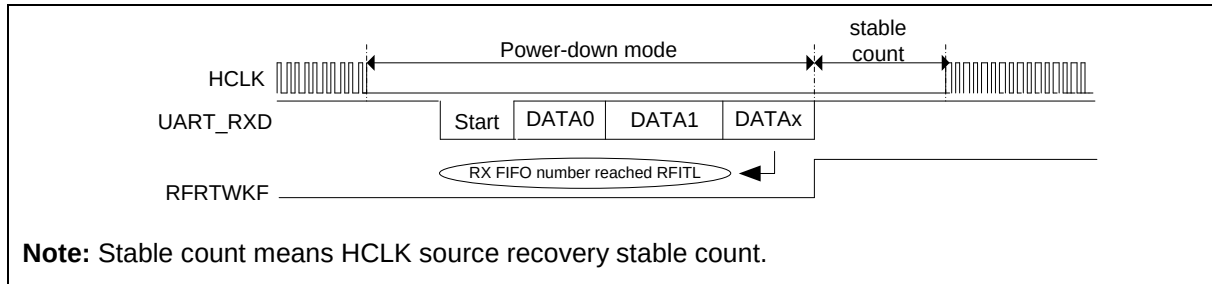


Figure 6.12-8 UART Received Data FIFO reached threshold wake-up

RS-485 Address Match (AAD Mode) Wake-up

The RS-485 address match wake-up function is enabled by setting WKRFRTEN (UART_WKCTL[2]) and WKRS485EN (UART_WKCTL[3]). This function is used for RS-485 Auto Address Detection (AAD) mode in RS-485 function mode and ADDRDN (UART_ALTCTL[15]) is set to 1. In Power-down mode, when an address byte is detected and matches the ADDRDMV (UART_ALTCTL[31:24]) or the number of received data in RX FIFO reaches the threshold value RFITL (UART_FIFO[7:4]), it can wake-up the system. If the WKRS485EN (UART_WKCTL[3]) is enabled, when an address byte is detected and matches the ADDRDMV (UART_ALTCTL[31:24]) that cause the RS485 address match (AAD mode) wake-up flag RS485WKF (UART_WKSTS[3]) generated. The RS-485 Address Match (AAD mode) wake-up is shown in Figure 6.12-9.

Note: The UART controller clock source should be selected as LXT in Power-down mode to receive data.

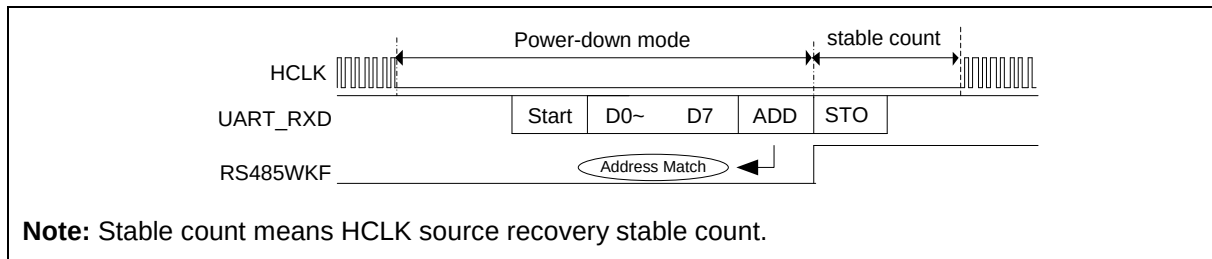


Figure 6.12-9 UART RS-485 AAD Mode Address Match Wake-up

Received Data FIFO Threshold Time-out Wake-up

The received data FIFO threshold time-out wake-up function is enabled by setting WKRFRTEN (UART_WKCTL[2]) and WKTOOUTEN (UART_WKCTL[4]). Setting TOCNTEN (UART_INTEN[11]) to enable receiver buffer time-out counter. In Power-down mode, when the number of received data in RX FIFO does not reach the threshold value RFITL (UART_FIFO[7:4]) and the time-out counter equals the time-out value TOIC (UART_TOUT[7:0]), it can wake-up the system. If the WKTOOUTEN (UART_WKCTL[4]) is enabled, when the time-out counter equals the time-out value TOIC (UART_TOUT[7:0]) that cause the Received Data FIFO threshold time-out wake-up flag TOUTWKF (UART_WKSTS[4]) generated. The Received Data FIFO threshold time-out wake-up is shown in Figure 6.12-10.

Note: The UART controller clock source should be selected as LXT in Power-down mode to receive data.

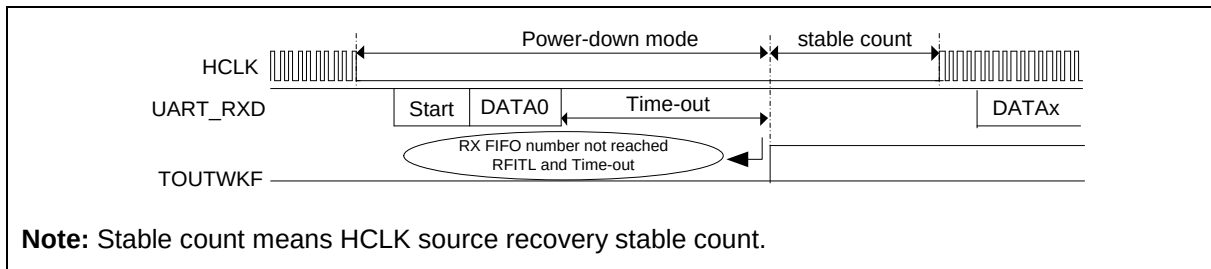


Figure 6.12-10 UART Received Data FIFO threshold time-out wake-up

6.12.5.7 UART Controller Interrupt and Status

Each UART controller supports ten types of interrupts including:

- Receive Data Available Interrupt (RDAINT)
- Transmit Holding Register Empty Interrupt (THREINT)
- Transmitter Empty Interrupt (TXENDIF)
- Receive Line Status Interrupt (RLSINT)
 - Break Interrupt Flag (BIF)
 - Framing Error Flag (FEF)
 - Parity Error Flag (PEF)
 - RS-485 Address Byte Detect Flag (ADDRDETf)
- MODEM Status Interrupt (MODEMINT)
 - Detect nCTS State Change Flag (CTSDETf)
- Receiver Buffer Time-out Interrupt (RXTOINT)
- Buffer Error Interrupt (BUFERRINT)
 - TX Overflow Error Interrupt Flag (TXOVIF)
 - RX Overflow Error Interrupt Flag (RXOVIF)
- Wake-up Interrupt (WKINT)
 - nCTS Wake-up Flag (CTSWKF)
 - Incoming Data Wake-up Flag (DATWKF)
 - Received Data FIFO Reached Threshold Wake-up Flag (RFRTWKF)
 - RS-485 Address Match (AAD mode) Wake-up Flag (RS485WKF)
 - Received Data FIFO Threshold Time-out Wake-up Flag (TOUTWKF)
- Auto-Baud Rate Interrupt (ABRINT)
 - Auto-baud Rate Detect Interrupt Flag (ABRDIF)
 - Auto-baud Rate Detect Time-out Interrupt Flag (ABRDTOIF)
- Single-wire Bit Error Detect Interrupt (SWBEINT)

Table 6.12-9 describes the interrupt sources and flags. The interrupt is generated when the interrupt flag is generated and the interrupt enable bit is set. User must clear the interrupt flag after the interrupt is generated.

Interrupt Source	Interrupt Indicator	Interrupt Enable Bit	Interrupt Flag	Flag Caused By	Flag Cleared By
Receive Data Available Interrupt	RDAINT	RDAIEN	RDAIF	N/A	Read UART_DAT
Transmit Holding Register Empty Interrupt	THREINT	THREIEN	THREIF	N/A	Write UART_DAT
Transmitter Empty Interrupt	TXENDINT	TXENDIEN	TXENDIF	N/A	Write UART_DAT
Receive Line Status Interrupt	RLSINT	RLSIEN	RLSIF	RLSIF = BIF	Write '1' to BIF
				RLSIF = FEF	Write '1' to FEF
				RLSIF = PEF	Write '1' to PEF
				RLSIF = ADDRDET	Write '1' to ADDRDET
Modem Status Interrupt	MODEMINT	MODEMIEN	MODEMIF	MODEMIF = CTSDET	Write '1' to CTSDET
Receiver Buffer Time-out Interrupt	RXTOINT	RXTOIEN	RXTOIF	N/A	Read UART_DAT
Buffer Error Interrupt	BUFERRINT	BUFERRIEN	BUFERRIF	BUFERRIF = TXOVIF	Write '1' to TXOVIF
				BUFERRIF = RXOVIF	Write '1' to RXOVIF
Wake-up Interrupt	WKINT	WKIEN	WKIF	WKIF = CTSWKF	Write '1' to CTSWKF
				WKIF = DATWKF	Write '1' to DATWKF
				WKIF = RFRTWKF	Write '1' to RFRTWKF
				WKIF = RS485WKF	Write '1' to RS485WKF
				WKIF = TOUTWKF	Write '1' to TOUTWKF
Auto-Baud Rate Interrupt	ABRINT	ABRIEN	ABRIF	ABRIF = ABRDIF	Write '1' to ABRDIF
				ABRIF = ABRDIOF	Write '1' to ABRDIOF
Single-wire Bit Error Detect Interrupt	SWBEINT	SWBEIEN	SWBEIF	N/A	Writing '1' to SWBEIF

Table 6.12-9 UART controller Interrupt Source and Flag List

6.12.5.8 UART Function Mode

The UART controller provides UART function (Setting FUNCSEL (UART_FUNCSEL [2:0]) to '000' to enable UART function mode).

The UART provides full-duplex and asynchronous communications. The transmitter and receiver contain 16 bytes FIFO for payloads. User can program receiver buffer trigger level and receiver buffer time-out detection for receiver. The transmitting data delay time between the last stop and the next START bit can be programmed by setting DLY (UART_TOUT [15:8]) register. The UART supports hardware auto-flow control that provides programmable nRTS flow control trigger level. The number of data bytes in RX FIFO is equal to or greater than RTSTRGLV (UART_FIFO[19:16]), the nRTS is de-asserted.

UART Line Control Function

The UART controller supports fully programmable serial-interface characteristics by setting the UART_LINE register. User can program UART_LINE register for the word length, STOP bit and PARITY bit setting. Table 6.12-10 and Table 6.12-11 list the UART word, STOP bit length and the PARITY bit settings.

NSB (UART_LINE[2])	WLS (UART_LINE[1:0])	Word Length (Bit)	Stop Length (Bit)
0	00	5	1
0	01	6	1
0	10	7	1
0	11	8	1
1	00	5	1.5
1	01	6	2
1	10	7	2
1	11	8	2

Table 6.12-10 UART Line Control of Word and Stop Length Setting

Parity Type	SPE (UART_LINE[5])	EPE (UART_LINE[4])	PSS (UART_LINE[7])	PBE (UART_LINE[3])	Description
No Parity	x	x	x	0	No PARITY bit output.
Parity source from UART_DATA	x	x	1	1	PARITY bit is generated and checked by software.
Odd Parity	0	0	0	1	Odd Parity is calculated by adding all the "1's" in a data stream and adding a PARITY bit to the total bits, to make the total count an odd number.
Even Parity	0	1	0	1	Even Parity is calculated by adding all the "1's" in a data stream and adding a PARITY bit to the total bits, to make the count an even number.
Forced Mask Parity	1	0	0	1	PARITY bit always logic 1. PARITY bit on the serial byte is set to "1" regardless of total number of "1's" (even or odd counts).
Forced Space Parity	1	1	0	1	PARITY bit always logic 0. PARITY bit on the serial byte is set to "0" regardless of total number of "1's" (even or odd counts).

Table 6.12-11 UART Line Control of PARITY Bit Setting

UART Auto-Flow Control Function

The UART supports auto-flow control function that uses two signals, nCTS (clear-to-send) and nRTS (request-to-send), to control the flow of data transfer between the UART and external devices (e.g. Modem). When auto-flow is enabled, the UART is not allowed to receive data until the UART asserts nRTS to external device. When the number of bytes stored in the RX FIFO equals the value of RTSTRGLV (UART_FIFO [19:16]), the nRTS is de-asserted. The UART sends data out when UART detects nCTS is asserted from external device. If the valid asserted nCTS is not detected, the UART will not send data out. The auto flow control block diagram is shown in Figure 6.12-11.

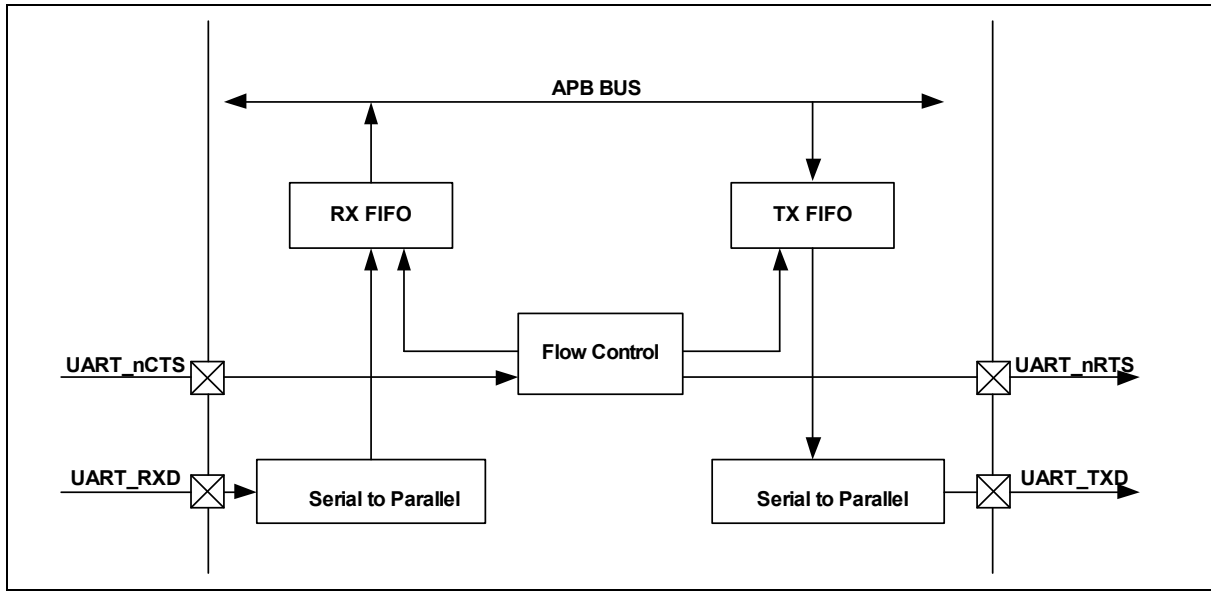


Figure 6.12-11 Auto-Flow Control Block Diagram

Figure 6.12-12 demonstrates the nCTS auto-flow control of UART function mode. User must set ATOCTSEN (UART_INTEN [13]) to enable nCTS auto-flow control function. The CTSACTLV (UART_MODEMSTS [8]) can set nCTS pin input active state. The CTSDETF (UART_MODEMSTS[0]) is set when any state change of nCTS pin input has occurred, and then TX data will be automatically transmitted from TX FIFO.

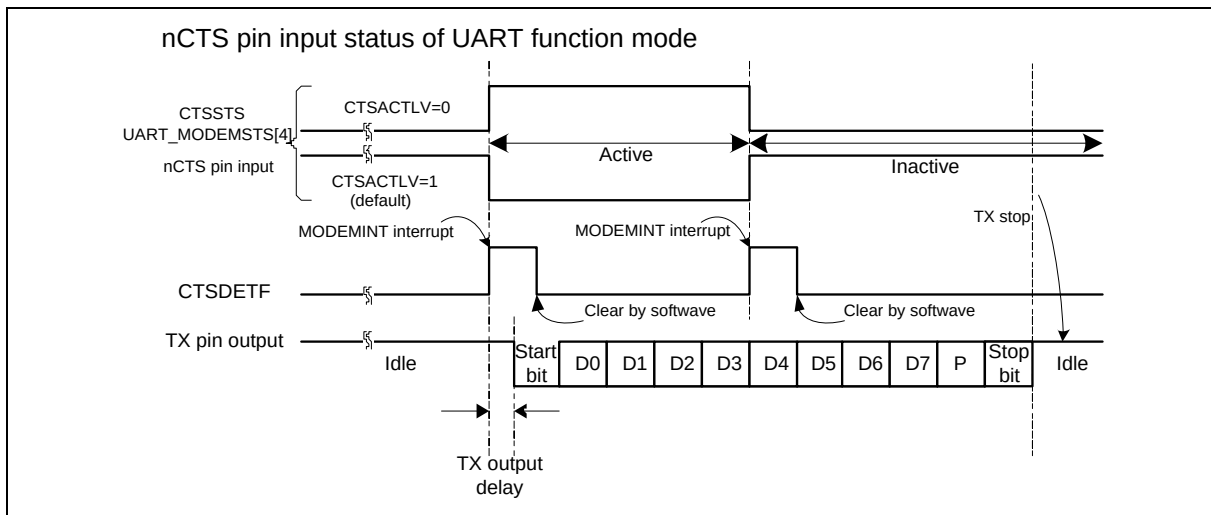


Figure 6.12-12 UART nCTS Auto-Flow Control Enabled

As shown in Figure 6.12-13, in UART nRTS auto-flow control mode (ATORTSEN(UART_INTEN[12])=1), the nRTS internal signal is controlled by UART FIFO controller with RTSTRGLV(UART_FIFO[19:16]) trigger level.

Setting RTSACTLV(UART_MODEM[9]) can control the nRTS pin output is inverse or non-inverse from nRTS signal. User can read the RTSSTS (UART_MODEM[13]) bit to get real nRTS pin output voltage logic status.

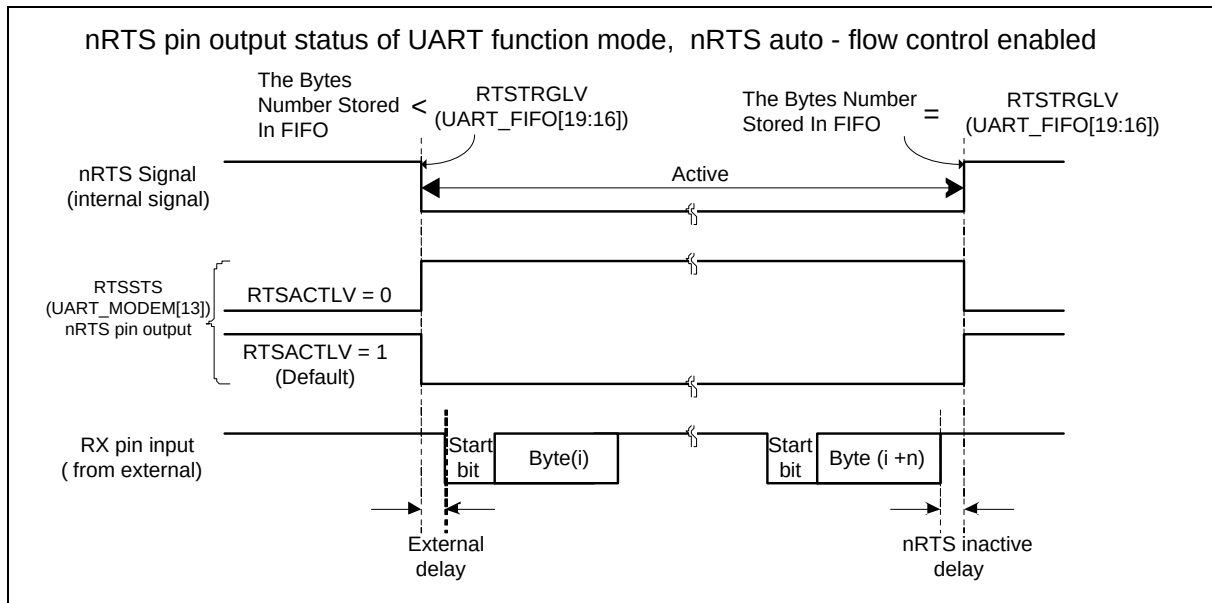


Figure 6.12-13 UART nRTS Auto-Flow Control Enabled

As shown in Figure 6.12-14, in software mode (ATORTSEN(UART_INTEN[12])=0), the nRTS flow is directly controlled by software programming of RTS(UART_MODEM[1]) control bit.

Setting RTSACTLV(UART_MODEM[9]) can control the nRTS pin output is inverse or non-inverse from RTS(UART_MODEM[1]) control bit. User can read the RTSSTS(UART_MODEM[13]) bit to get real nRTS pin output voltage logic status.

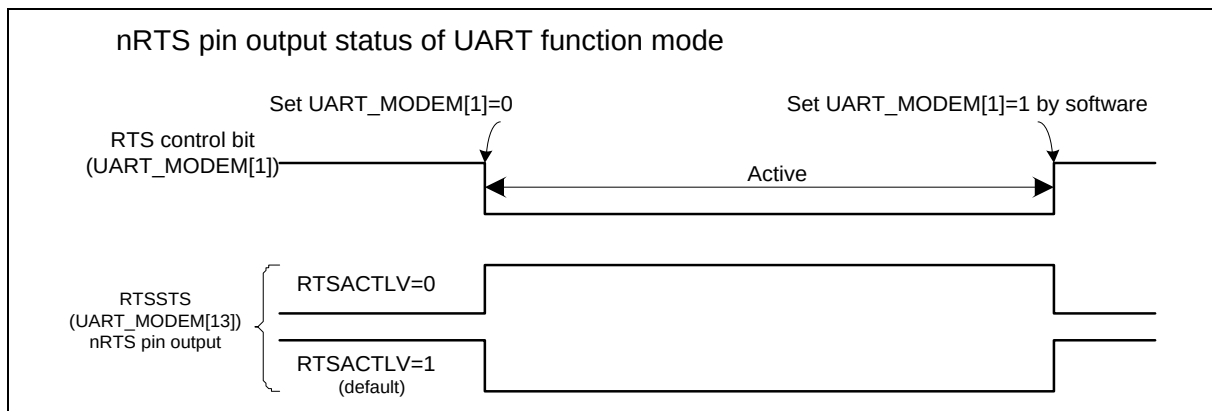


Figure 6.12-14 UART nRTS Auto-Flow with Software Control

6.12.5.9 IrDA Function Mode

The UART controller also provides Serial IrDA (SIR, Serial Infrared) function (Setting UART_FUNCSEL [2:0] to '010' to enable the IrDA function). The SIR specification defines a short-range infrared asynchronous serial transmission mode with one START bit, 8 data bits, and 1 STOP bit. The maximum data rate is 115.2 kbps. The IrDA SIR block contains an IrDA SIR protocol encoder/decoder. The IrDA SIR protocol is half-duplex only. So, it cannot transmit and receive data at the same time. The IrDA SIR physical layer specifies a minimum 10 ms transfer delay between transmission and reception, and this delay feature must be implemented by software.

In IrDA mode, the BAUDM1 (UART_BAUD [29]) must be cleared.

Baud Rate = Clock / (16 * (BRD + 2)), where BRD (UART_BAUD[15:0]) is Baud Rate Divider in UART_BAUD register.

Note: The tolerance of baud-rate is $\pm 5\%$ between IrDA master and IrDA slave.

The IrDA control block diagram is shown in Figure 6.12-15.

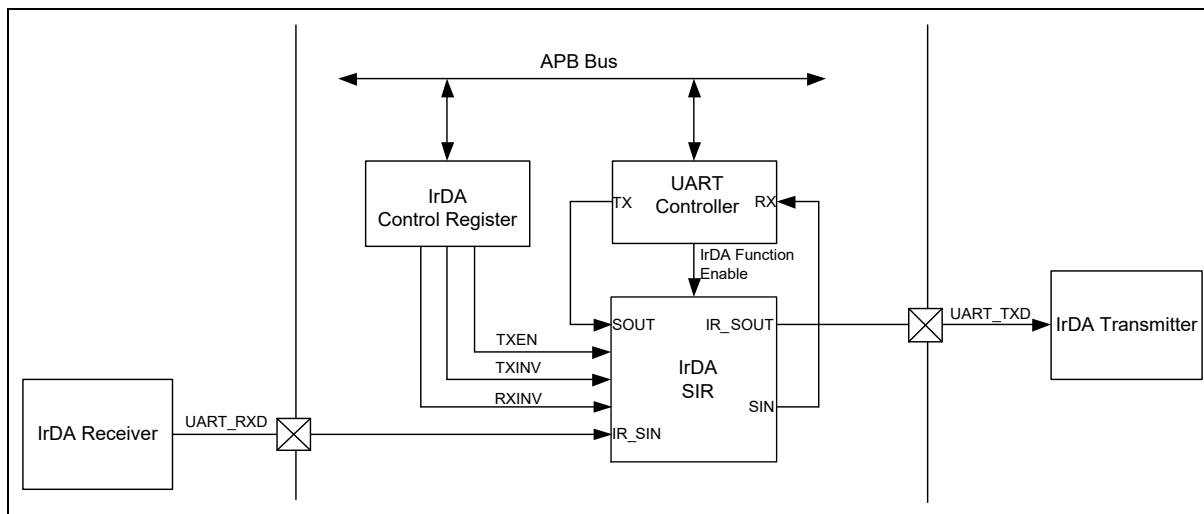


Figure 6.12-15 IrDA Control Block Diagram

IrDA SIR Transmit Encoder

The IrDA SIR Transmit Encoder modulates Non-Return-to-Zero (NRZ) transmit bit stream output from UART. The IrDA SIR physical layer specifies the use of Return-to-Zero, Inverted (RZI) modulation scheme which represents logic 0 as an infra light pulse. The modulated output pulse stream is transmitted to an external output driver and infrared light emitting diode.

The transmitted pulse width is specified as 3/16 period of baud rate.

IrDA SIR Receive Decoder

The IrDA SIR Receive Decoder demodulates the Return-to-Zero bit stream from the input detector and outputs the NRZ serial bits stream to the UART received data input.

In idle state, the decoder input is high. A START bit is detected when the decoder input is LOW. In normal operation, the RXINV (UART_IRDA[6]) is set to '1' and TXINV (UART_IRDA[5]) is set to '0'.

IrDA SIR Operation

The IrDA SIR encoder/decoder provides functionality which converts between UART data stream and half-duplex serial SIR interface. Figure 6.12-16 is IrDA encoder/decoder waveform.

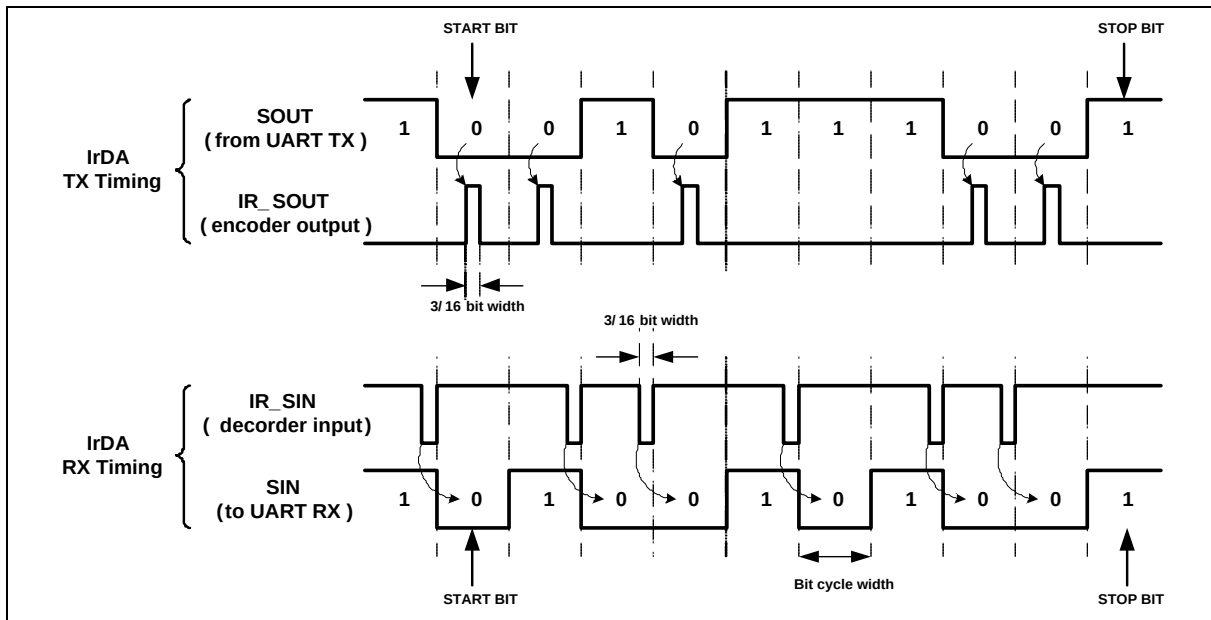


Figure 6.12-16 IrDA TX/RX Timing Diagram

6.12.5.10 RS-485 Function Mode

Another alternate function of UART controller is RS-485 function (user must set UART_FUNCSEL [2:0] to '011' to enable RS-485 function), and direction control provided by nRTS pin from an asynchronous serial port. The RS-485 transceiver control is implemented by using the nRTS control signal to enable the RS-485 driver. Many characteristics of the RX and TX are same as UART in RS-485 mode.

The UART controller can be configured as a RS-485 addressable slave or a RS-485 master. RS-485 master transmitter will identify an address character by setting the parity (9th bit) to 1. For data characters, the parity is set to 0. Software can use UART_LINE register to control the 9th bit (When the PBE, EPE and SPE are set, the 9th bit is transmitted 0 and when PBE and SPE are set and EPE is cleared, the 9th bit is transmitted 1).

The controller supports three operation modes: RS-485 Normal Multidrop Operation Mode (NMM), RS-485 Auto Address Detection Operation Mode (AAD) and RS-485 Auto Direction Control Operation Mode (AUD). Software can choose any operation mode by programming the UART_ALTCTL register, and drive the transfer delay time between the last STOP bit leaving the TX FIFO and the de-assertion of by setting DLY (UART_TOUT [15:8]) register.

RS-485 Normal Multidrop Operation Mode (NMM)

In RS-485 Normal Multidrop Operation Mode (RS485NMM (UART_ALTCTL[8]) = 1), in first, software must decide the data which before the address byte be detected will be stored in RX FIFO or not. If software wants to ignore any data before address byte detected, the flow is set RXOFF (UART_FIFO [8]) then enable RS485NMM (UART_ALTCTL [8]) and the receiver will ignore any data until an address byte is detected (bit 9 = 1) and the address byte data will be stored in the RX FIFO. If software wants to receive any data before address byte detected, the flow is disables RXOFF (UART_FIFO [8]) then enable RS485NMM (UART_ALTCTL [8]) and the receiver will received any data.

If an address byte is detected (bit 9 = 1), it will generate an interrupt to CPU and RXOFF (UART_FIFO [8]) can decide whether accepting the following data bytes are stored in the RX FIFO. If software disables receiver by setting RXOFF (UART_FIFO [8]) register, when a next address byte is detected, the controller will clear the RXOFF (UART_FIFO [8]) bit and the address byte data will be stored in the RX FIFO.

RS-485 Auto Address Detection Operation Mode (AAD)

In RS-485 Auto Address Detection Operation Mode (RS485AAD (UART_ALTCTL[9]) = 1), the receiver will ignore any data until an address byte is detected (bit 9 = 1) and the address byte data matches the ADDR MV (UART_ALTCTL[31:24]) value. The address byte data will be stored in the RX FIFO. The all received byte data will be accepted and stored in the RX FIFO until an address byte data not match the ADDR MV (UART_ALTCTL[31:24]) value.

RS-485 Auto Direction Function (AUD)

Another option function of RS-485 controllers is RS-485 auto direction control function (RS485AUD (UART_ALTCTL[10]) = 1). The RS-485 transceiver control is implemented by using the nRTS control signal from an asynchronous serial port. The nRTS line is connected to the RS-485 transceiver enable pin such that setting the nRTS line to high (logic 1) enables the RS-485 transceiver. Setting the nRTS line to low (logic 0) puts the transceiver into the tri-state condition. User can set RTSACTLV in UART_MODEM register to change the nRTS driving level.

Figure 6.12-17 demonstrates the RS-485 nRTS driving level in AUD mode. The nRTS pin will be automatically driven during TX data transmission.

Setting RTSACTLV(UART_MODEM[9]) can control nRTS pin output driving level. User can read the RTSSTS(UART_MODEM[13]) bit to get real nRTS pin output voltage logic status.

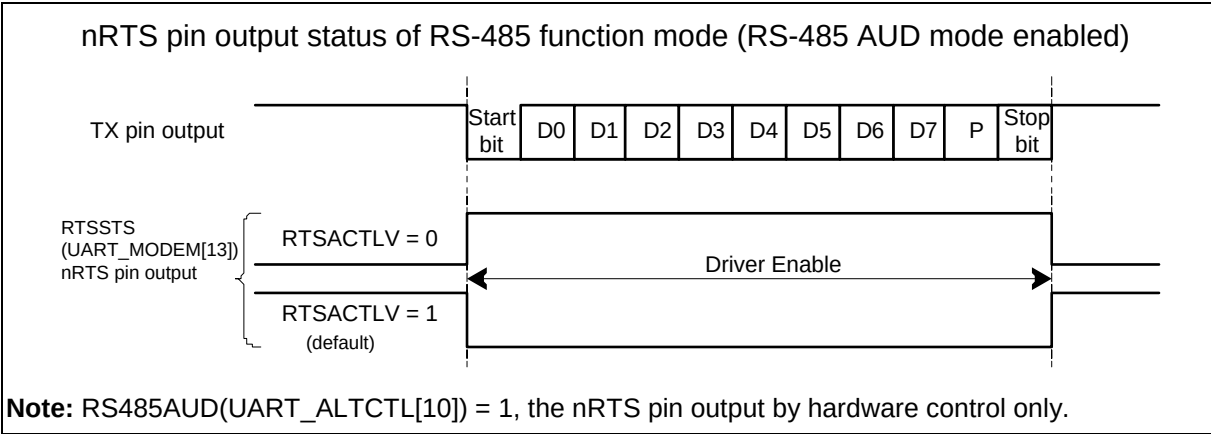


Figure 6.12-17 RS-485 nRTS Driving Level in Auto Direction Mode

Figure 6.12-18 demonstrates the RS-485 nRTS driving level in software control (RS485AUD (UART_ALTCTL[10])=0). The nRTS driving level is controlled by programing the RTS(UART_MODEM[1]) control bit.

Setting RTSACTLV (UART_MODEM[9]) can control the nRTS pin output is inverse or non-inverse from RTS(UART_MODEM[1]) control bit. User can read the RTSSTS (UART_MODEM[13]) bit to get real nRTS pin output voltage logic status. The structure of RS-485 frame is shown in Figure 6.12-19.

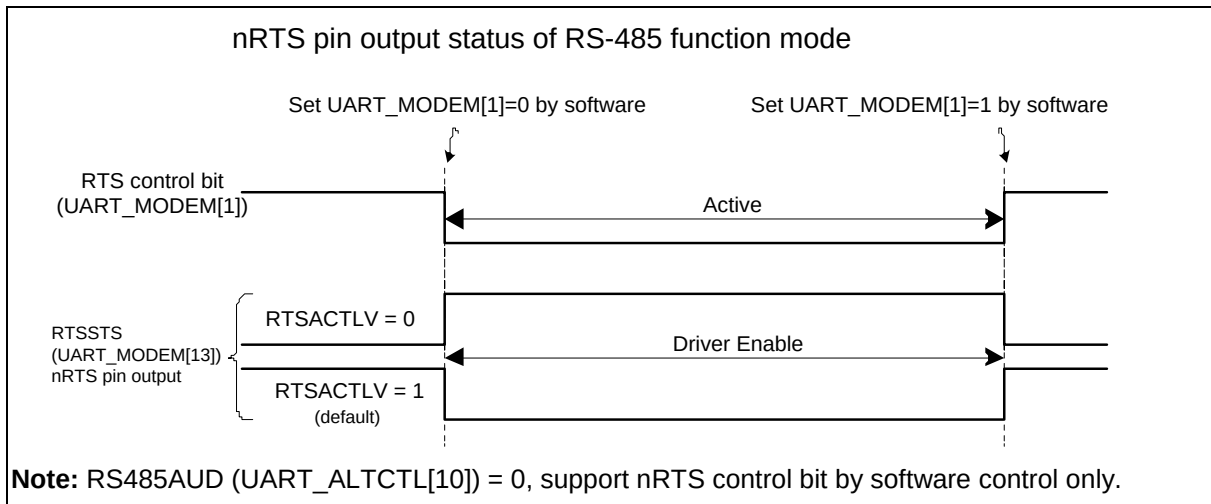


Figure 6.12-18 RS-485 nRTS Driving Level with Software Control

Programming Sequence Example:

1. Program FUNCSEL in UART_FUNCSEL to select RS-485 function.
2. Program the RXOFF (UART_FIFO[8]) to determine enable or disable the receiver RS-485 receiver.
3. Program the RS485NMM (UART_ALTCTL[8]) or RS485AAD (UART_ALTCTL[9]) mode.
4. If the RS485AAD (UART_ALTCTL[9]) mode is selected, the ADDRNV (UART_ALTCTL[31:24]) is programmed for auto address match value.
5. Determine auto direction control by programming RS485AUD (UART_ALTCTL[10]).

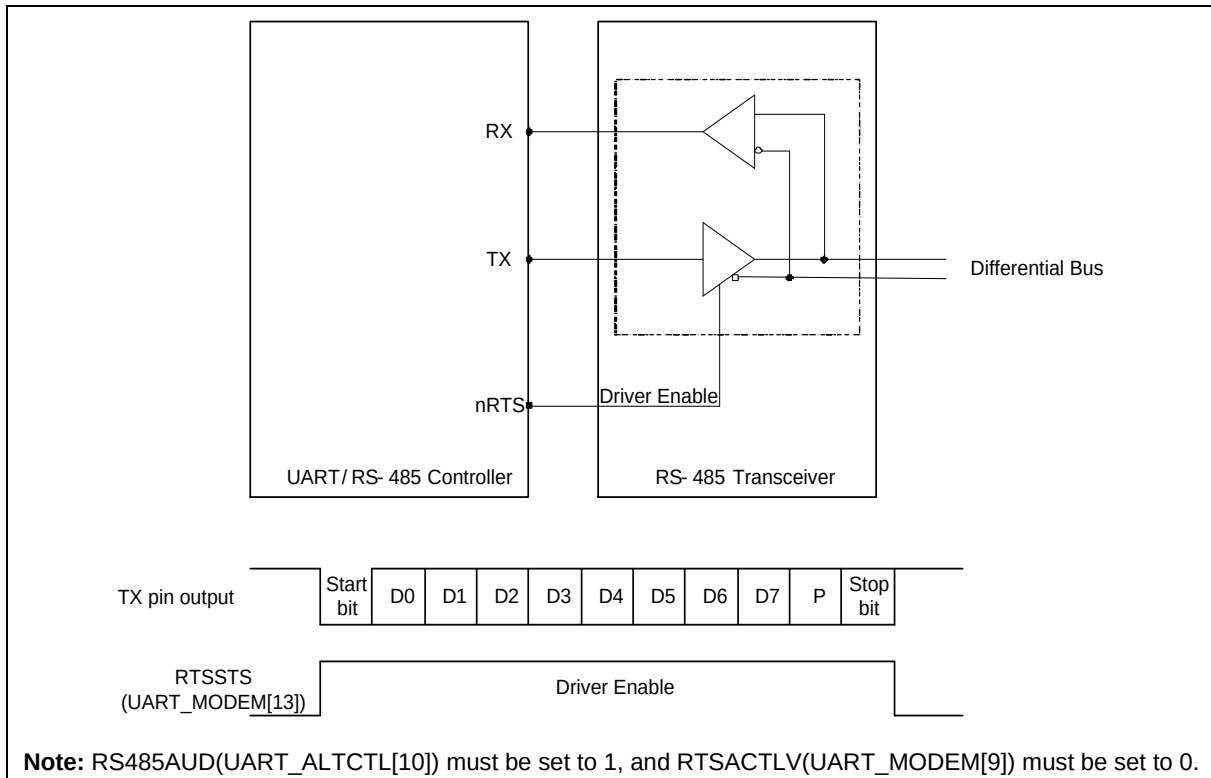


Figure 6.12-19 Structure of RS-485 Frame

6.12.5.11 UART Single-wire Half Duplex

The UART controller provides single-wire half duplex function in UART function mode. Setting UART_FUNCSEL [2:0] to '100' to enable the UART Single-wire function. The single-wire bus is idle (RXIDLE(UART_FIFOSTS[29]) = 1) in RX state. Before writing data to transmit buffer (UART_DAT[7:0]), the bus state should be checked in idle (RXIDLE(UART_FIFOSTS[29])). By writing data to transmit buffer, the bus state transfers to TX state immediately. After the transmission, the bus state transfers from TX state to RX state.

The UART will not allowed to receive data in single-wire half duplex function TX mode. If nRTS is asserted in TX mode, nRTS will make the docking UART device send out data and cause bus confliction. To reduce the bus confliction, the UART controller supports flow control function and bit error detection but does not support auto-flow control function. The nRTS is automatically inactivated in single-wire TX state. In TX state, the UART controller will monitor bus state. If the bus state is not equal to TX state, the SWBEIF(UART_INTSTS[16]) is set.

6.12.5.12 PDMA Transfer Function

The UART controller supports PDMA transfer function.

By configuring PDMA parameter and set UART_DAT as the PDMA destination address. When TXPDMAEN (UART_INTEN[14]) is set to 1, the controller will issue request to PDMA controller to start the PDMA transmission process automatically.

By configuring PDMA parameter and set UART_DAT as the PDMA source address. When RXPDMAEN (UART_INTEN[15]) is set to 1, the controller will start the PDMA reception process. UART controller will issue request to PDMA controller automatically when there is data in the RX FIFO buffer.

6.12.6 Register Map

R: read only, W: write only, R/W: both read and write

Register	Offset	R/W	Description	Reset Value
UART Base Address: $\text{UARTx\_BA} = 0x4007\_0000 + (0x1000 * x)$ $x=0,1,2,3,4$				
UART_DAT $x=0,1,2,3,4$	UARTx_BA+0x00	R/W	UART Receive/Transmit Buffer Register	Undefined
UART_INTEN $x=0,1,2,3,4$	UARTx_BA+0x04	R/W	UART Interrupt Enable Register	0x0000_0000
UART_FIFO $x=0,1,2,3,4$	UARTx_BA+0x08	R/W	UART FIFO Control Register	0x0000_0101
UART_LINE $x=0,1,2,3,4$	UARTx_BA+0x0C	R/W	UART Line Control Register	0x0000_0000
UART_MODEM $x=0,1,2,3,4$	UARTx_BA+0x10	R/W	UART Modem Control Register	0x0000_0200
UART_MODEMSTS $x=0,1,2,3,4$	UARTx_BA+0x14	R/W	UART Modem Status Register	0x0000_0110
UART_FIFOSTS $x=0,1,2,3,4$	UARTx_BA+0x18	R/W	UART FIFO Status Register	0xB040_4000
UART_INTSTS $x=0,1,2,3,4$	UARTx_BA+0x1C	R/W	UART Interrupt Status Register	0x0040_0002
UART_TOUT $x=0,1,2,3,4$	UARTx_BA+0x20	R/W	UART Time-out Register	0x0000_0000
UART_BAUD $x=0,1,2,3,4$	UARTx_BA+0x24	R/W	UART Baud Rate Divider Register	0x0F00_0000
UART_IRDA $x=0,1,2,3,4$	UARTx_BA+0x28	R/W	UART IrDA Control Register	0x0000_0040
UART_ALTCTL $x=0,1,2,3,4$	UARTx_BA+0x2C	R/W	UART Alternate Control/Status Register	0x0000_000C
UART_FUNCSEL $x=0,1,2,3,4$	UARTx_BA+0x30	R/W	UART Function Select Register	0x0000_0000
UART_BRCOMP $x=0,1,2,3$	UARTx_BA+0x3C	R/W	UART Baud Rate Compensation Register	0x0000_0000
UART_WKCTL $x=0,1,2,3,4$	UARTx_BA+0x40	R/W	UART Wake-up Control Register	0x0000_0000

UART_WKSTS x=0,1,2,3,4	UARTx_BA+0x44	R/W	UART Wake-up Status Register	0x0000_0000
UART_DWKCOMP x=0,1,2,3,4	UARTx_BA+0x48	R/W	UART Incoming Data Wake-up Compensation Register	0x0000_0000

6.12.7 Register Description

UART Receive/Transmit Buffer Register (UART_DAT)

Register	Offset	R/W	Description	Reset Value
UART_DAT x=0,1,2,3,4	UARTx_BA+0x00	R/W	UART Receive/Transmit Buffer Register	Undefined

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

Bits	Description
[31:9]	Reserved Reserved.
[8]	PARITY PARITY Bit Receive/Transmit Buffer Write Operation: By writing to this bit, the PARITY bit will be stored in transmitter FIFO. If PBE (UART_LINE[3]) and PSS (UART_LINE[7]) are set, the UART controller will send out this bit follow the DAT (UART_DAT[7:0]) through the UART_TXD. Read Operation: If PBE (UART_LINE[3]) and PSS (UART_LINE[7]) are enabled, the PARITY bit can be read by this bit. Note: This bit has effect only when PBE (UART_LINE[3]) and PSS (UART_LINE[7]) are set.
[7:0]	DAT Data Receive/Transmit Buffer Write Operation: By writing one byte to this register, the data byte will be stored in transmitter FIFO. The UART controller will send out the data stored in transmitter FIFO top location through the UART_TXD. Read Operation: By reading this register, the UART controller will return an 8-bit data received from receiver FIFO.

UART Interrupt Enable Register (UART_INTEN)

Register	Offset	R/W	Description	Reset Value
UART_INTEN x=0,1,2,3,4	UARTx_BA+0x04	R/W	UART Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved	TXENDIEN	Reserved			ABRIEN	Reserved	SWBEIEN
15	14	13	12	11	10	9	8
RXPDMAEN	TXPDMAEN	ATOCTSEN	ATORTSEN	TOCNTEN	Reserved		
7	6	5	4	3	2	1	0
Reserved	WKIEN	BUFERRIEN	RXTOIEN	MODEMIEN	RLSIEN	THREIEN	RDAIEN

Bits	Description
[31:23]	Reserved Reserved.
[22]	TXENDIEN Transmitter Empty Interrupt Enable Bit If TXENDIEN (UART_INTEN[22]) is enabled, the Transmitter Empty interrupt TXENDINT (UART_INTSTS[30]) will be generated when TXENDIF (UART_INTSTS[22]) is set (TX FIFO (UART_DAT) is empty and the STOP bit of the last byte has been transmitted). 0 = Transmitter empty interrupt Disabled. 1 = Transmitter empty interrupt Enabled.
[21:19]	Reserved Reserved.
[18]	ABRIEN Auto-baud Rate Interrupt Enable Bit 0 = Auto-baud rate interrupt Disabled. 1 = Auto-baud rate interrupt Enabled.
[17]	Reserved Reserved.
[16]	SWBEIEN Single-wire Bit Error Detection Interrupt Enable Bit Set this bit, the Single-wire Half Duplex Bit Error Detection Interrupt SWBEINT (UART_INTSTS[24]) is generated when Single-wire Bit Error Detection SWBEIF (UART_INTSTS[16]) is set. 0 = Single-wire Bit Error Detect Interrupt Disabled. 1 = Single-wire Bit Error Detect Interrupt Enabled. Note: This bit is valid when FUNCSEL (UART_FUNCSEL[2:0]) is select UART Single-wire mode.
[15]	RXPDMAEN RX PDMA Enable Bit This bit can enable or disable RX PDMA service. 0 = RX PDMA Disabled. 1 = RX PDMA Enabled. Note: If RLSIEN (UART_INTEN[2]) is enabled and HWRLSINT (UART_INTSTS[26]) is set to 1, the RLS (Receive Line Status) Interrupt is caused. If RLS interrupt is caused by Break Error Flag BIF (UART_FIFOSTS[6]), Frame Error Flag FEF (UART_FIFO[5]) or Parity Error Flag PEF (UART_FIFOSTS[4]), UART PDMA receive request operation is stopped. Clear Break Error Flag BIF or Frame Error Flag FEF or Parity Error Flag PEF by writing "1" to corresponding BIF, FEF and PEF to make UART PDMA receive request operation continue.

[14]	TXPDMAEN	TX PDMA Enable Bit 0 = TX PDMA Disabled. 1 = TX PDMA Enabled. Note: If RLSIEN (UART_INTEN[2]) is enabled and HWRLSINT (UART_INTSTS[26]) is set to 1, the RLS (Receive Line Status) Interrupt is caused. If RLS interrupt is caused by Break Error Flag BIF(UART_FIFOSTS[6]), Frame Error Flag FEF(UART_FIFO[5]) or Parity Error Flag PEF(UART_FIFOSTS[4]), UART PDMA transmit request operation is stopped. Clear Break Error Flag BIF or Frame Error Flag FEF or Parity Error Flag PEF by writing "1" to corresponding BIF, FEF and PEF to make UART PDMA transmit request operation continue.
[13]	ATOCTSEN	nCTS Auto-flow Control Enable Bit 0 = nCTS auto-flow control Disabled. 1 = nCTS auto-flow control Enabled. Note: When nCTS auto-flow is enabled, the UART will send data to external device if nCTS input assert (UART will not send data to device until nCTS is asserted).
[12]	ATORTSEN	nRTS Auto-flow Control Enable Bit 0 = nRTS auto-flow control Disabled. 1 = nRTS auto-flow control Enabled. Note: When nRTS auto-flow is enabled, if the number of bytes in the RX FIFO equals the RTSTRGLV (UART_FIFO[19:16]), the UART will de-assert nRTS signal.
[11]	TOCNTEN	Receive Buffer Time-out Counter Enable Bit 0 = Receive Buffer Time-out counter Disabled. 1 = Receive Buffer Time-out counter Enabled.
[10:7]	Reserved	Reserved.
[6]	WKIEN	Wake-up Interrupt Enable Bit 0 = Wake-up Interrupt Disabled. 1 = Wake-up Interrupt Enabled.
[5]	BUFERRIEN	Buffer Error Interrupt Enable Bit 0 = Buffer error interrupt Disabled. 1 = Buffer error interrupt Enabled.
[4]	RXTOIEN	RX Time-out Interrupt Enable Bit 0 = RX time-out interrupt Disabled. 1 = RX time-out interrupt Enabled.
[3]	MODEMIEN	Modem Status Interrupt Enable Bit 0 = Modem status interrupt Disabled. 1 = Modem status interrupt Enabled.
[2]	RLSIEN	Receive Line Status Interrupt Enable Bit 0 = Receive Line Status interrupt Disabled. 1 = Receive Line Status interrupt Enabled.
[1]	THREIEN	Transmit Holding Register Empty Interrupt Enable Bit 0 = Transmit holding register empty interrupt Disabled. 1 = Transmit holding register empty interrupt Enabled.
[0]	RDAIEN	Receive Data Available Interrupt Enable Bit 0 = Receive data available interrupt Disabled. 1 = Receive data available interrupt Enabled.

UART FIFO Control Register (UART_FIFO)

Register	Offset	R/W	Description	Reset Value
UART_FIFO x=0,1,2,3,4	UARTx_BA+0x08	R/W	UART FIFO Control Register	0x0000_0101

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved				RTSTRGLV			
15	14	13	12	11	10	9	8
Reserved							RXOFF
7	6	5	4	3	2	1	0
RFITL				Reserved	TXRST	RXRST	Reserved

Bits	Description
[31:20]	Reserved Reserved.
[19:16]	RTSTRGLV nRTS Trigger Level for Auto-flow Control 0000 = nRTS Trigger Level is 1 byte. 0001 = nRTS Trigger Level is 4 bytes. 0010 = nRTS Trigger Level is 8 bytes. 0011 = nRTS Trigger Level is 14 bytes. Others = Reserved. Note: This field is used for auto nRTS flow control.
[15:9]	Reserved Reserved.
[8]	RXOFF Receiver Disable Bit The receiver is disabled or not (set 1 to disable receiver). 0 = Receiver Enabled. 1 = Receiver Disabled. Note: This bit is used for RS-485 Normal Multi-drop mode. It should be programmed before RS485NMM (UART_ALTCTL [8]) is programmed.
[7:4]	RFITL RX FIFO Interrupt Trigger Level When the number of bytes in the receive FIFO equals the RFITL, the RDAIF (UART_INTSTS[0]) will be set (if RDAIEN (UART_INTEN [0]) enabled, and an interrupt will be generated). 0000 = RX FIFO Interrupt Trigger Level is 1 byte. 0001 = RX FIFO Interrupt Trigger Level is 4 bytes. 0010 = RX FIFO Interrupt Trigger Level is 8 bytes. 0011 = RX FIFO Interrupt Trigger Level is 14 bytes. Others = Reserved.
[3]	Reserved Reserved.
[2]	TXRST TX Field Software Reset When TXRST (UART_FIFO[2]) is set, all the byte in the transmit FIFO and TX internal state machine are cleared.

		0 = No effect. 1 = Reset the TX internal state machine and pointers. Note 1: This bit will automatically clear at least 3 UART peripheral clock cycles. Note 2: Before setting this bit, it should wait for the TXEMPTYF (UART_FIFOSTS[28]) be set.
[1]	RXRST	RX Field Software Reset When RXRST (UART_FIFO[1]) is set, all the byte in the receiver FIFO and RX internal state machine are cleared. 0 = No effect. 1 = Reset the RX internal state machine and pointers. Note 1: This bit will automatically clear at least 3 UART peripheral clock cycles. Note 2: Before setting this bit, it should wait for the RXIDLE (UART_FIFOSTS[29]) be set.
[0]	Reserved	Reserved.

UART Line Control Register (UART_LINE)

Register	Offset	R/W	Description	Reset Value
UART_LINE x=0,1,2,3,4	UARTx_BA+0x0C	R/W	UART Line Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved						RXDINV	TXDINV
7	6	5	4	3	2	1	0
PSS	BCB	SPE	EPE	PBE	NSB	WLS	

Bits	Description
[31:10]	Reserved Reserved.
[9]	RXDINV RX Data Inverted 0 = Received data signal inverted Disabled. 1 = Received data signal inverted Enabled. Note 1: Before setting this bit, TXRXDIS (UART_FUNCSEL[3]) should be set then waited for TXRXACT (UART_FIFOSTS[31]) is cleared. When the configuration is done, cleared TXRXDIS (UART_FUNCSEL[3]) to activate UART controller. Note 2: This bit is valid when FUNCSEL (UART_FUNCSEL[2:0]) is select UART or RS485 function.
[8]	TXDINV TX Data Inverted 0 = Transmitted data signal inverted Disabled. 1 = Transmitted data signal inverted Enabled. Note 1: Before setting this bit, TXRXDIS (UART_FUNCSEL[3]) should be set then waited for TXRXACT (UART_FIFOSTS[31]) is cleared. When the configuration is done, cleared TXRXDIS (UART_FUNCSEL[3]) to activate UART controller. Note 2: This bit is valid when FUNCSEL (UART_FUNCSEL[2:0]) is select UART or RS485 function.
[7]	PSS PARITY Bit Source Selection The PARITY bit can be selected to be generated and checked automatically or by software. 0 = PARITY bit is generated by EPE (UART_LINE[4]) and SPE (UART_LINE[5]) setting and checked automatically. 1 = PARITY bit generated and checked by software. Note 1: This bit has effect only when PBE (UART_LINE[3]) is set. Note 2: If PSS is 0, the PARITY bit is transmitted and checked automatically. If PSS is 1, the transmitted PARITY bit value can be determined by writing PARITY (UART_DAT[8]) and the PARITY bit can be read by reading PARITY (UART_DAT[8]).
[6]	BCB Break Control Bit 0 = Break Control Disabled. 1 = Break Control Enabled. Note: When this bit is set to logic 1, the transmitted serial data output (TX) is forced to the Spacing State (logic 0). This bit acts only on TX line and has no effect on the transmitter logic.

[5]	SPE	Stick Parity Enable Bit 0 = Stick parity Disabled. 1 = Stick parity Enabled. Note: If PBE (UART_LINE[3]) and EPE (UART_LINE[4]) are logic 1, the PARITY bit is transmitted and checked as logic 0. If PBE (UART_LINE[3]) is 1 and EPE (UART_LINE[4]) is 0 then the PARITY bit is transmitted and checked as 1.
[4]	EPE	Even Parity Enable Bit 0 = Odd number of logic 1's is transmitted and checked in each word. 1 = Even number of logic 1's is transmitted and checked in each word. Note: This bit has effect only when PBE (UART_LINE[3]) is set.
[3]	PBE	PARITY Bit Enable Bit 0 = PARITY bit generated Disabled. 1 = PARITY bit generated Enabled. Note: PARITY bit is generated on each outgoing character and is checked on each incoming data.
[2]	NSB	Number of "STOP Bit" 0 = One "STOP bit" is generated in the transmitted data. 1 = When select 5-bit word length, 1.5 "STOP bit" is generated in the transmitted data. When select 6-, 7- and 8-bit word length, 2 "STOP bit" is generated in the transmitted data.
[1:0]	WLS	Word Length Selection This field sets UART word length. 00 = 5 bits. 01 = 6 bits. 10 = 7 bits. 11 = 8 bits.

UART Modem Control Register (UART_MODEM)

Register	Offset	R/W	Description	Reset Value
UART_MODEM x=0,1,2,3,4	UARTx_BA+0x10	R/W	UART Modem Control Register	0x0000_0200

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

Bits	Description
[31:14]	Reserved Reserved.
[13]	RTSSTS nRTS Pin Status (Read Only) This bit mirror from nRTS pin output of voltage logic status. 0 = nRTS pin output is low level voltage logic state. 1 = nRTS pin output is high level voltage logic state.
[12:10]	Reserved Reserved.
[9]	RTSACTLV nRTS Pin Active Level This bit defines the active level state of nRTS pin output. 0 = nRTS pin output is high level active. 1 = nRTS pin output is low level active. (Default) Note 1: Refer to Figure 6.12-13 and Figure 6.12-14 for UART function mode. Note 2: Refer to Figure 6.12-17 and Figure 6.12-18 for RS-485 function mode. Note 3: Before setting this bit, TXRXDIS (UART_FUNCSEL[3]) should be set then waited for TXRXACT (UART_FIFOSTS[31]) is cleared. When the configuration is done, cleared TXRXDIS (UART_FUNCSEL[3]) to activate UART controller.
[8:2]	Reserved Reserved.
[1]	RTS nRTS Signal Control This bit is direct control internal nRTS (Request-to-send) signal active or not, and then drive the nRTS pin output with RTSACTLV bit configuration. 0 = nRTS signal is active. 1 = nRTS signal is inactive. Note 1: The nRTS signal control bit is not effective when nRTS auto-flow control is enabled in UART function mode. Note 2: The nRTS signal control bit is not effective when RS-485 auto direction mode (AUD) is enabled in RS-485 function mode. Note 3: Single-wire mode is support this feature.
[0]	Reserved Reserved.

UART Modem Status Register (UART_MODEMSTS)

Register	Offset	R/W	Description	Reset Value
UART_MODEMSTS x=0,1,2,3,4	UARTx_BA+0x14	R/W	UART Modem Status Register	0x0000_0110

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

Bits	Description
[31:9]	Reserved Reserved.
[8]	CTSACTLV nCTS Pin Active Level This bit defines the active level state of nCTS pin input. 0 = nCTS pin input is high level active. 1 = nCTS pin input is low level active. (Default) Note: Before setting this bit, TXRXDIS (UART_FUNCSEL[3]) should be set then waited for TXRXACT (UART_FIFOSTS[31]) is cleared. When the configuration is done, cleared TXRXDIS (UART_FUNCSEL[3]) to activate UART controller.
[7:5]	Reserved Reserved.
[4]	CTSSTS nCTS Pin Status (Read Only) This bit mirror from nCTS pin input of voltage logic status. 0 = nCTS pin input is low level voltage logic state. 1 = nCTS pin input is high level voltage logic state. Note: This bit echoes when UART controller peripheral clock is enabled, and nCTS multi-function port is selected.
[3:1]	Reserved Reserved.
[0]	CTSDETF Detect nCTS State Change Flag This bit is set whenever nCTS input has change state, and it will generate Modem interrupt to CPU when MODEMIEN (UART_INTEN [3]) is set to 1. 0 = nCTS input has not change state. 1 = nCTS input has change state. Note: This bit can be cleared by writing "1" to it.

UART FIFO Status Register (UART_FIFOSTS)

Register	Offset	R/W	Description	Reset Value
UART_FIFOSTS x=0,1,2,3,4	UARTx_BA+0x18	R/W	UART FIFO Status Register	0xB040_4000

31	30	29	28	27	26	25	24
TXRXACT	Reserved	RXIDLE	TXEMPTYF	Reserved			TXOVIF
23	22	21	20	19	18	17	16
TXFULL	TXEMPTY	TXPTR					
15	14	13	12	11	10	9	8
RXFULL	RXEMPTY	RXPTR					
7	6	5	4	3	2	1	0
Reserved	BIF	FEF	PEF	ADDRDETf	ABRDTOIF	ABRDIF	RXOVIF

Bits	Description
[31]	TX and RX Active Status (Read Only) This bit indicates TX and RX are active or inactive. 0 = TX and RX are inactive. 1 = TX and RX are active. (Default) Note: When TXRXDIS (UART_FUNCSEL[3]) is set and both TX and RX are in idle state, this bit is cleared. The UART controller cannot transmit or receive data at this moment. Otherwise this bit is set.
[30]	Reserved Reserved.
[29]	RX Idle Status (Read Only) This bit is set by hardware when RX is idle. 0 = RX is busy. 1 = RX is idle. (Default)
[28]	Transmitter Empty Flag (Read Only) This bit is set by hardware when TX FIFO (UART_DAT) is empty and the STOP bit of the last byte has been transmitted. 0 = TX FIFO is not empty or the STOP bit of the last byte has been not transmitted. 1 = TX FIFO is empty and the STOP bit of the last byte has been transmitted. Note: This bit is cleared automatically when TX FIFO is not empty or the last byte transmission has not completed.
[27:25]	Reserved Reserved.
[24]	TX Overflow Error Interrupt Flag If TX FIFO (UART_DAT) is full, an additional write to UART_DAT will cause this bit to logic 1. 0 = TX FIFO is not overflow. 1 = TX FIFO is overflow. Note: This bit can be cleared by writing "1" to it.
[23]	Transmitter FIFO Full (Read Only) This bit indicates TX FIFO full or not. 0 = TX FIFO is not full.

		1 = TX FIFO is full. Note: This bit is set when the number of usage in TX FIFO Buffer is equal to 16, otherwise it is cleared by hardware.
[22]	TXEMPTY	Transmitter FIFO Empty (Read Only) This bit indicates TX FIFO empty or not. 0 = TX FIFO is not empty. 1 = TX FIFO is empty. Note: When the last byte of TX FIFO has been transferred to Transmitter Shift Register, hardware sets this bit high. It will be cleared when writing data into UART_DAT (TX FIFO not empty).
[21:16]	TXPTR	TX FIFO Pointer (Read Only) This field indicates the TX FIFO Buffer Pointer. When CPU writes one byte into UART_DAT, TXPTR increases one. When one byte of TX FIFO is transferred to Transmitter Shift Register, TXPTR decreases one. The Maximum value shown in TXPTR is 15. When the using level of TX FIFO Buffer is equal to 16, the TXFULL bit is set to 1 and TXPTR will show 0. As one byte of TX FIFO is transferred to Transmitter Shift Register, the TXFULL bit is cleared to 0 and TXPTR will show 15.
[15]	RXFULL	Receiver FIFO Full (Read Only) This bit initiates RX FIFO full or not. 0 = RX FIFO is not full. 1 = RX FIFO is full. Note: This bit is set when the number of usage in RX FIFO Buffer is equal to 16, otherwise it is cleared by hardware.
[14]	RXEMPTY	Receiver FIFO Empty (Read Only) This bit initiate RX FIFO empty or not. 0 = RX FIFO is not empty. 1 = RX FIFO is empty. Note: When the last byte of RX FIFO has been read by CPU, hardware sets this bit high. It will be cleared when UART receives any new data.
[13:8]	RXPTR	RX FIFO Pointer (Read Only) This field indicates the RX FIFO Buffer Pointer. When UART receives one byte from external device, RXPTR increases one. When one byte of RX FIFO is read by CPU, RXPTR decreases one. The Maximum value shown in RXPTR is 15. When the using level of RX FIFO Buffer is equal to 16, the RXFULL bit is set to 1 and RXPTR will show 0. As one byte of RX FIFO is read by CPU, the RXFULL bit is cleared to 0 and RXPTR will show 15.
[7]	Reserved	Reserved.
[6]	BIF	Break Interrupt Flag This bit is set to logic 1 whenever the received data input (RX) is held in the "spacing state" (logic 0) for longer than a full word transmission time (that is, the total time of "START bit" + data bits + parity + STOP bits). 0 = No Break interrupt is generated. 1 = Break interrupt is generated. Note: This bit can be cleared by writing "1" to it.
[5]	FEF	Framing Error Flag This bit is set to logic 1 whenever the received character does not have a valid "STOP bit" (that is, the STOP bit following the last data bit or PARITY bit is detected as logic 0). 0 = No framing error is generated. 1 = Framing error is generated. Note: This bit can be cleared by writing "1" to it.
[4]	PEF	Parity Error Flag This bit is set to logic 1 whenever the received character does not have a valid "PARITY bit".

		0 = No parity error is generated. 1 = Parity error is generated. Note: This bit can be cleared by writing "1" to it.
[3]	ADDRDET	RS-485 Address Byte Detect Flag 0 = Receiver detects a data that is not an address bit (bit 9 = '0'). 1 = Receiver detects a data that is an address bit (bit 9 = '1'). Note 1: This field is used for RS-485 function mode and ADDRDN (UART_ALTCTL[15]) is set to 1 to enable Address detection mode. Note 2: This bit can be cleared by writing "1" to it.
[2]	ABRDTOIF	Auto-baud Rate Detect Time-out Interrupt Flag This bit is set to logic "1" in Auto-baud Rate Detect mode when the baud rate counter is overflow. 0 = Auto-baud rate counter is underflow. 1 = Auto-baud rate counter is overflow. Note: This bit can be cleared by writing "1" to it.
[1]	ABRDIF	Auto-baud Rate Detect Interrupt Flag This bit is set to logic "1" when auto-baud rate detect function is finished. 0 = Auto-baud rate detect function is not finished. 1 = Auto-baud rate detect function is finished. Note: This bit can be cleared by writing "1" to it.
[0]	RXOVIF	RX Overflow Error Interrupt Flag This bit is set when RX FIFO overflow. If the number of bytes of received data is greater than RX_FIFO (UART_DAT) size 16 bytes, this bit will be set. 0 = RX FIFO is not overflow. 1 = RX FIFO is overflow. Note: This bit can be cleared by writing "1" to it.

UART Interrupt Status Register (UART_INTSTS)

Register	Offset	R/W	Description	Reset Value
UART_INTSTS x=0,1,2,3,4	UARTx_BA+0x1C	R/W	UART Interrupt Status Register	0x0040_0002

31	30	29	28	27	26	25	24
ABRINT	TXENDINT	HWBUFEINT	HWTOINT	HWMODINT	HWRLSINT	Reserved	SWBEINT
23	22	21	20	19	18	17	16
Reserved	TXENDIF	HWBUFEIF	HWTOIF	HWMODIF	HWRLSIF	Reserved	SWBEIF
15	14	13	12	11	10	9	8
Reserved	WKINT	BUFERRINT	RXTOINT	MODEMINT	RLSINT	THREINT	RDAINT
7	6	5	4	3	2	1	0
Reserved	WKIF	BUFERRIF	RXTOIF	MODEMIF	RLSIF	THREIF	RDAIF

Bits	Description
[31] ABRINT	Auto-baud Rate Interrupt Indicator (Read Only) This bit is set if ABRIEN (UART_INTEN[18]) and ABRIF (UART_ALTCTL[17]) are both set to 1. 0 = No Auto-baud Rate interrupt is generated. 1 = The Auto-baud Rate interrupt is generated.
[30] TXENDINT	Transmitter Empty Interrupt Indicator (Read Only) This bit is set if TXENDIEN (UART_INTEN[22]) and TXENDIF(UART_INTSTS[22]) are both set to 1. 0 = No Transmitter Empty interrupt is generated. 1 = Transmitter Empty interrupt is generated.
[29] HWBUFEINT	PDMA Mode Buffer Error Interrupt Indicator (Read Only) This bit is set if BUFERRIEN (UART_INTEN[5]) and HWBUFEIF (UART_INTSTS[21]) are both set to 1. 0 = No buffer error interrupt is generated in PDMA mode. 1 = Buffer error interrupt is generated in PDMA mode.
[28] HWTOINT	PDMA Mode RX Time-out Interrupt Indicator (Read Only) This bit is set if RXTOIEN (UART_INTEN[4]) and HWTOIF(UART_INTSTS[20]) are both set to 1. 0 = No RX time-out interrupt is generated in PDMA mode. 1 = RX time-out interrupt is generated in PDMA mode.
[27] HWMODINT	PDMA Mode MODEM Status Interrupt Indicator (Read Only) This bit is set if MODEMIEN (UART_INTEN[3]) and HWMODIF(UART_INTSTS[19]) are both set to 1. 0 = No Modem interrupt is generated in PDMA mode. 1 = Modem interrupt is generated in PDMA mode.
[26] HWRLSINT	PDMA Mode Receive Line Status Interrupt Indicator (Read Only) This bit is set if RLSIEN (UART_INTEN[2]) and HWRLSIF(UART_INTSTS[18]) are both set to 1. 0 = No RLS interrupt is generated in PDMA mode. 1 = RLS interrupt is generated in PDMA mode.
[25] Reserved	Reserved.

[24]	SWBEINT	Single-wire Bit Error Detect Interrupt Indicator (Read Only) This bit is set if SWBEIEN (UART_INTEN[16]) and SWBEIF (UART_INTSTS[16]) are both set to 1. 0 = No Single-wire Bit Error Detection Interrupt generated. 1 = Single-wire Bit Error Detection Interrupt generated.
[23]	Reserved	Reserved.
[22]	TXENDIF	Transmitter Empty Interrupt Flag This bit is set when TX FIFO (UART_DAT) is empty and the STOP bit of the last byte has been transmitted (TXEMPTYF (UART_FIFOSTS[28]) is set). If TXENDIEN (UART_INTEN[22]) is enabled, the Transmitter Empty interrupt will be generated. 0 = No transmitter empty interrupt flag is generated. 1 = Transmitter empty interrupt flag is generated. Note: This bit is cleared automatically when TX FIFO is not empty or the last byte transmission has not completed.
[21]	HWBUFEIF	PDMA Mode Buffer Error Interrupt Flag (Read Only) This bit is set when the TX or RX FIFO overflows (TXOVIF (UART_FIFOSTS [24]) or RXOVIF (UART_FIFOSTS[0]) is set). When BUFERRIF (UART_INTSTS[5]) is set, the transfer maybe is not correct. If BUFERRIEN (UART_INTEN [5]) is enabled, the buffer error interrupt will be generated. 0 = No buffer error interrupt flag is generated in PDMA mode. 1 = Buffer error interrupt flag is generated in PDMA mode. Note: This bit is cleared when both TXOVIF (UART_FIFOSTS[24]) and RXOVIF (UART_FIFOSTS[0]) are cleared.
[20]	HWTOIF	PDMA Mode RX Time-out Interrupt Flag (Read Only) This bit is set when the RX FIFO is not empty and no activities occurred in the RX FIFO and the time-out counter equal to TOIC (UART_TOUT[7:0]). If RXTOIEN (UART_INTEN [4]) is enabled, the RX time-out interrupt will be generated. 0 = No RX time-out interrupt flag is generated in PDMA mode. 1 = RX time-out interrupt flag is generated in PDMA mode. Note: This bit is read only and user can read UART_DAT (RX is in active) to clear it.
[19]	HWMODIF	PDMA Mode MODEM Interrupt Flag (Read Only) This bit is set when the nCTS pin has state change (CTSDETF (UART_MODEMSTS [0] =1)). If MODEMIEN (UART_INTEN [3]) is enabled, the Modem interrupt will be generated. 0 = No Modem interrupt flag is generated in PDMA mode. 1 = Modem interrupt flag is generated in PDMA mode. Note: This bit is read only and reset to 0 when the bit CTSDETF (UART_MODEMSTS[0]) is cleared by writing 1 on CTSDETF (UART_MODEMSTS [0]).
[18]	HWRLSIF	PDMA Mode Receive Line Status Flag (Read Only) This bit is set when the RX receive data have parity error, frame error or break error (at least one of 3 bits, BIF (UART_FIFOSTS[6]), FEF (UART_FIFOSTS[5]) and PEF (UART_FIFOSTS[4]) is set). If RLSIEN (UART_INTEN [2]) is enabled, the RLS interrupt will be generated. 0 = No RLS interrupt flag is generated in PDMA mode. 1 = RLS interrupt flag is generated in PDMA mode. Note 1: In RS-485 function mode, this field includes "receiver detect any address byte received address byte character (bit9 = '1') bit". Note 2: In UART function mode, this bit is read only and reset to 0 when all bits of BIF(UART_FIFOSTS[6]), FEF(UART_FIFOSTS[5]) and PEF(UART_FIFOSTS[4]) are cleared. Note 3: In RS-485 function mode, this bit is read only and reset to 0 when all bits of BIF(UART_FIFOSTS[6]), FEF(UART_FIFOSTS[5]), PEF(UART_FIFOSTS[4]) and ADDRDETF (UART_FIFOSTS[3]) are cleared.
[17]	Reserved	Reserved.
[16]	SWBEIF	Single-wire Bit Error Detection Interrupt Flag

		This bit is set when the single wire bus state is not equal to UART controller TX state in Single-wire mode. 0 = No single-wire bit error detection interrupt flag is generated. 1 = Single-wire bit error detection interrupt flag is generated. Note 1: This bit is active when FUNCSEL (UART_FUNCSEL[2:0]) is select UART Single-wire mode. Note 2: This bit can be cleared by writing "1" to it.
[15]	Reserved	Reserved.
[14]	WKINT	UART Wake-up Interrupt Indicator (Read Only) This bit is set if WKIEN (UART_INTEN[6]) and WKIF (UART_INTSTS[6]) are both set to 1. 0 = No UART wake-up interrupt is generated. 1 = UART wake-up interrupt is generated.
[13]	BUFERRINT	Buffer Error Interrupt Indicator (Read Only) This bit is set if BUFERRIEN(UART_INTEN[5]) and BUFERRIF(UART_INTSTS[5]) are both set to 1. 0 = No buffer error interrupt is generated. 1 = Buffer error interrupt is generated.
[12]	RXTOINT	RX Time-out Interrupt Indicator (Read Only) This bit is set if RXTOIEN (UART_INTEN[4]) and RXTOIF(UART_INTSTS[4]) are both set to 1. 0 = No RX time-out interrupt is generated. 1 = RX time-out interrupt is generated.
[11]	MODEMINT	MODEM Status Interrupt Indicator (Read Only) This bit is set if MODEMIEN(UART_INTEN[3]) and MODEMIF(UART_INTSTS[3]) are both set to 1 0 = No Modem interrupt is generated. 1 = Modem interrupt is generated..
[10]	RLSINT	Receive Line Status Interrupt Indicator (Read Only) This bit is set if RLSIEN (UART_INTEN[2]) and RLSIF(UART_INTSTS[2]) are both set to 1. 0 = No RLS interrupt is generated. 1 = RLS interrupt is generated.
[9]	THREINT	Transmit Holding Register Empty Interrupt Indicator (Read Only) This bit is set if THREIEN (UART_INTEN[1]) and THREIF(UART_INTSTS[1]) are both set to 1. 0 = No THRE interrupt is generated. 1 = THRE interrupt is generated.
[8]	RDAINT	Receive Data Available Interrupt Indicator (Read Only) This bit is set if RDAIEN (UART_INTEN[0]) and RDAIF (UART_INTSTS[0]) are both set to 1. 0 = No RDA interrupt is generated. 1 = RDA interrupt is generated.
[7]	Reserved	Reserved.
[6]	WKIF	UART Wake-up Interrupt Flag (Read Only) This bit is set when TOUTWKF (UART_WKSTS[4]), RS485WKF (UART_WKSTS[3]), RFRTWKF (UART_WKSTS[2]), DATWKF (UART_WKSTS[1]) or CTSWKF(UART_WKSTS[0]) is set to 1. 0 = No UART wake-up interrupt flag is generated. 1 = UART wake-up interrupt flag is generated. Note: This bit is cleared if all of TOUTWKF, RS485WKF, RFRTWKF, DATWKF and CTSWKF are cleared to 0 by writing 1 to the corresponding interrupt flag.
[5]	BUFERRIF	Buffer Error Interrupt Flag (Read Only) This bit is set when the TX FIFO or RX FIFO overflows (TXOVIF (UART_FIFOSTS[24]) or RXOVIF (UART_FIFOSTS[0]) is set). When BUFERRIF (UART_INTSTS[5]) is set, the transfer is not correct. If BUFERRIEN (UART_INTEN [5]) is enabled, the buffer error interrupt will be generated.

		0 = No buffer error interrupt flag is generated. 1 = Buffer error interrupt flag is generated. Note: This bit is cleared if both of RXOVIF(UART_FIFOSTS[0]) and TXOVIF(UART_FIFOSTS[24]) are cleared to 0 by writing 1 to RXOVIF(UART_FIFOSTS[0]) and TXOVIF(UART_FIFOSTS[24]).
[4]	RXTOIF	RX Time-out Interrupt Flag (Read Only) This bit is set when the RX FIFO is not empty and no activities occurred in the RX FIFO and the time-out counter equal to TOIC (UART_TOUT[7:0]). If RXTOIEN (UART_INTEN [4]) is enabled, the RX time-out interrupt will be generated. 0 = No RX time-out interrupt flag is generated. 1 = RX time-out interrupt flag is generated. Note: This bit is read only and user can read UART_DAT (RX is in active) to clear it.
[3]	MODEMIF	MODEM Interrupt Flag (Read Only) This bit is set when the nCTS pin has state change (CTSDETF (UART_MODEMSTS[0]) = 1). If MODEMIEN (UART_INTEN [3]) is enabled, the Modem interrupt will be generated. 0 = No Modem interrupt flag is generated. 1 = Modem interrupt flag is generated. Note: This bit is read only and reset to 0 when bit CTSDETF is cleared by a write 1 on CTSDETF(UART_MODEMSTS[0]).
[2]	RLSIF	Receive Line Interrupt Flag (Read Only) This bit is set when the RX receive data have parity error, frame error or break error (at least one of 3 bits, BIF(UART_FIFOSTS[6]), FEF(UART_FIFOSTS[5]) and PEF(UART_FIFOSTS[4]), is set). If RLSIEN (UART_INTEN [2]) is enabled, the RLS interrupt will be generated. 0 = No RLS interrupt flag is generated. 1 = RLS interrupt flag is generated. Note 1: In RS-485 function mode, this field is set include "receiver detect and received address byte character (bit9 = '1') bit". At the same time, the bit of ADDRDETF (UART_FIFOSTS[3]) is also set. Note 2: This bit is read only and reset to 0 when all bits of BIF (UART_FIFOSTS[6]), FEF(UART_FIFOSTS[5]) and PEF(UART_FIFOSTS[4]) are cleared. Note 3: In RS-485 function mode, this bit is read only and reset to 0 when all bits of BIF (UART_FIFOSTS[6]), FEF(UART_FIFOSTS[5]), PEF(UART_FIFOSTS[4]) and ADDRDETF (UART_FIFOSTS[3]) are cleared.
[1]	THREIF	Transmit Holding Register Empty Interrupt Flag This bit is set when the last data of TX FIFO is transferred to Transmitter Shift Register. If THREIEN (UART_INTEN[1]) is enabled, the THRE interrupt will be generated. 0 = No THRE interrupt flag is generated. 1 = THRE interrupt flag is generated. Note: This bit is read only and it will be cleared when writing data into UART_DAT (TX FIFO not empty).
[0]	RDAIF	Receive Data Available Interrupt Flag When the number of bytes in the RX FIFO equals the RFITL then the RDAIF(UART_INTSTS[0]) will be set. If RDAIEN (UART_INTEN [0]) is enabled, the RDA interrupt will be generated. 0 = No RDA interrupt flag is generated. 1 = RDA interrupt flag is generated. Note: This bit is read only and it will be cleared when the number of unread bytes of RX FIFO drops below the threshold level (RFITL(UART_FIFO[7:4])).

UART Time-out Register (UART TOUT)

Register	Offset	R/W	Description	Reset Value
UART_TOUT x=0,1,2,3,4	UARTx_BA+0x20	R/W	UART Time-out Register	0x0000_0000

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

Bits	Description
[31:16]	Reserved Reserved.
[15:8]	DLY TX Delay Time Value This field is used to program the transfer delay time between the last STOP bit and next START bit. The unit is bit time.
[7:0]	TOIC Time-out Interrupt Comparator The time-out counter resets and starts counting (the counting clock = baud rate) whenever the RX FIFO receives a new data word if time out counter is enabled by setting TOCNTEN (UART_INTEN[11]). Once the content of time-out counter is equal to that of time-out interrupt comparator (TOIC (UART_TOUT[7:0])), a receiver time-out interrupt (RXTOINT(UART_INTSTS[12])) is generated if RXTOIEN (UART_INTEN [4]) enabled. A new incoming data word or RX FIFO empty will clear RXTOIF (UART_INTSTS[4]). In order to avoid receiver time-out interrupt generation immediately during one character is being received, TOIC value should be set between 40 and 255. So, for example, if TOIC is set with 40, the time-out interrupt is generated after four characters are not received when 1 STOP bit and no parity check is set for UART transfer.

UART Baud Rate Divider Register (UART_BAUD)

Register	Offset	R/W	Description	Reset Value
UART_BAUD x=0,1,2,3,4	UARTx_BA+0x24	R/W	UART Baud Rate Divider Register	0x0F00_0000

31	30	29	28	27	26	25	24
Reserved		BAUDM1	BAUDM0	EDIVM1			
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
BRD							
7	6	5	4	3	2	1	0
BRD							

Bits	Description
[31:30]	Reserved Reserved.
[29]	BAUDM1 BAUD Rate Mode Selection Bit 1 This bit is baud rate mode selection bit 1. UART provides three baud rate calculation modes. This bit combines with BAUDM0 (UART_BAUD[28]) to select baud rate calculation mode. The detail description is shown in Table 6.12-4. Note: In IrDA mode must be operated in mode 0.
[28]	BAUDM0 BAUD Rate Mode Selection Bit 0 This bit is baud rate mode selection bit 0. UART provides three baud rate calculation modes. This bit combines with BAUDM1 (UART_BAUD[29]) to select baud rate calculation mode. The detail description is shown in Table 6.12-4.
[27:24]	EDIVM1 Extra Divider for BAUD Rate Mode 1 This field is used for baud rate calculation in mode 1 and has no effect for baud rate calculation in mode 0 and mode 2. The detail description is shown in Table 6.12-4.
[23:16]	Reserved Reserved.
[15:0]	BRD Baud Rate Divider The field indicates the baud rate divider. This field is used in baud rate calculation. The detail description is shown in Table 6.12-4.

UART IrDA Control Register (UART_IRDA)

Register	Offset	R/W	Description	Reset Value
UART_IRDA x=0,1,2,3,4	UARTx_BA+0x28	R/W	UART IrDA Control Register	0x0000_0040

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

Bits	Description
[31:7]	Reserved Reserved.
[6]	RXINV IrDA Inverse Receive Input Signal 0 = None inverse receiving input signal. 1 = Inverse receiving input signal. (Default) Note 1: Before setting this bit, TXRXDIS (UART_FUNCSEL[3]) should be set then waited for TXRXACT (UART_FIFOSTS[31]) is cleared. When the configuration is done, cleared TXRXDIS (UART_FUNCSEL[3]) to activate UART controller. Note 2: This bit is valid when FUNCSEL (UART_FUNCSEL[2:0]) is select IrDA function.
[5]	TXINV IrDA Inverse Transmitting Output Signal 0 = None inverse transmitting signal. (Default). 1 = Inverse transmitting output signal. Note 1: Before setting this bit, TXRXDIS (UART_FUNCSEL[3]) should be set then waited for TXRXACT (UART_FIFOSTS[31]) is cleared. When the configuration is done, cleared TXRXDIS (UART_FUNCSEL[3]) to activate UART controller. Note 2: This bit is valid when FUNCSEL (UART_FUNCSEL[2:0]) is select IrDA function.
[4:2]	Reserved Reserved.
[1]	TXEN IrDA Receiver/Transmitter Selection Enable Bit 0 = IrDA Transmitter Disabled and Receiver Enabled. (Default) 1 = IrDA Transmitter Enabled and Receiver Disabled.
[0]	Reserved Reserved.

UART Alternate Control/Status Register (UART_ALTCTL)

Register	Offset	R/W	Description	Reset Value
UART_ALTCTL x=0,1,2,3,4	UARTx_BA+0x2C	R/W	UART Alternate Control/Status Register	0x0000_000C

31	30	29	28	27	26	25	24
ADDRMV							
23	22	21	20	19	18	17	16
Reserved			ABRDBITS		ABRDEN	ABRIF	Reserved
15	14	13	12	11	10	9	8
ADDRDEN	Reserved				RS485AUD	RS485AAD	RS485NMM
7	6	5	4	3	2	1	0
Reserved							

Bits	Description
[31:24] ADDR MV	Address Match Value This field contains the RS-485 address match values. Note: This field is used for RS-485 auto address detection mode.
[23:21] Reserved	Reserved.
[20:19] ABR DBITS	Auto-baud Rate Detect Bit Length 00 = 1-bit time from START bit to the 1 st rising edge. The input pattern shall be 0x01. 01 = 2-bit time from START bit to the 1 st rising edge. The input pattern shall be 0x02. 10 = 4-bit time from START bit to the 1 st rising edge. The input pattern shall be 0x08. 11 = 8-bit time from START bit to the 1 st rising edge. The input pattern shall be 0x80. Note : The calculation of bit number includes the START bit.
[18] ABR DEN	Auto-baud Rate Detect Enable Bit 0 = Auto-baud rate detect function Disabled. 1 = Auto-baud rate detect function Enabled. Note : This bit is cleared automatically after auto-baud detection is finished.
[17] ABR IF	Auto-baud Rate Interrupt Flag (Read Only) This bit is set when auto-baud rate detection function finished or the auto-baud rate counter was overflow and if ABRIEN(UART_INTEN [18]) is set then the auto-baud rate interrupt will be generated. 0 = No auto-baud rate interrupt flag is generated. 1 = Auto-baud rate interrupt flag is generated. Note: This bit is read only, but it can be cleared by writing "1" to ABRDIOIF (UART_FIFOSTS[2]) and ABRDIF(UART_FIFOSTS[1]).
[16] Reserved	Reserved.
[15] ADDR DEN	RS-485 Address Detection Enable Bit This bit is used to enable RS-485 Address Detection mode. 0 = Address detection mode Disabled. 1 = Address detection mode Enabled.

		Note: This bit is used for RS-485 any operation mode.
[14:11]	Reserved	Reserved.
[10]	RS485AUD	RS-485 Auto Direction Function 0 = RS-485 Auto Direction Operation function (AUD) Disabled. 1 = RS-485 Auto Direction Operation function (AUD) Enabled. Note: It can be active with RS-485_AAD or RS-485_NMM operation mode.
[9]	RS485AAD	RS-485 Auto Address Detection Operation Mode 0 = RS-485 Auto Address Detection Operation mode (AAD) Disabled. 1 = RS-485 Auto Address Detection Operation mode (AAD) Enabled. Note: It cannot be active with RS-485_NMM operation mode.
[8]	RS485NMM	RS-485 Normal Multi-drop Operation Mode 0 = RS-485 Normal Multi-drop Operation mode (NMM) Disabled. 1 = RS-485 Normal Multi-drop Operation mode (NMM) Enabled. Note: It cannot be active with RS-485_AAD operation mode.
[7:0]	Reserved	Reserved.

UART Function Select Register (UART_FUNCSEL)

Register	Offset	R/W	Description	Reset Value
UART_FUNCSEL x=0,1,2,3,4	UARTx_BA+0x30	R/W	UART Function Select Register	0x0000_0000

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

Bits	Description
[31:7]	Reserved Reserved.
[6]	DGE Deglitch Enable Bit 0 = Deglitch Disabled. 1 = Deglitch Enabled. Note 1: When this bit is set to logic 1, any pulse width less than about 150 ns will be considered a glitch and will be removed in the serial data input (RX). This bit acts only on RX line and has no effect on the transmitter logic. Note 2: It is recommended to set this bit only when operating at baud rate under 2.5 Mbps.
[5:4]	Reserved Reserved.
[3]	TXRXDIS TX and RX Disable Bit Setting this bit can disable TX and RX. 0 = TX and RX Enabled. 1 = TX and RX Disabled. Note: The TX and RX will not be disabled immediately when this bit is set. The TX and RX complete current task before TX and RX are disabled. When TX and RX are disabled, the TXRXACT (UART_FIFOSTS[31]) is cleared.
[2:0]	FUNCSEL Function Select 000 = UART function. 010 = IrDA function. 011 = RS-485 function. 100 = UART Single-wire function. Others = Reserved.

UART Baud Rate Compensation Register (UART_BRCOMP)

Register	Offset	R/W	Description	Reset Value
UART_BRCOMP x=0,1,2,3	UARTx_BA+0x3C	R/W	UART Baud Rate Compensation Register	0x0000_0000

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

Bits	Description
[31]	BRCOMPDEC Baud Rate Compensation Decrease 0 = Positive (increase one module clock) compensation for each compensated bit. 1 = Negative (decrease one module clock) compensation for each compensated bit.
[30:9]	Reserved Reserved.
[8:0]	BRCOMP Baud Rate Compensation Patten These 9-bits are used to define the relative bit is compensated or not. BRCOMP[7:0] is used to define the compensation of DAT (UART_DAT[7:0]) and BRCOMP[8] is used to define PARITY (UART_DAT[8]).

UART Wake-up Control Register (UART_WKCTL)

Register	Offset	R/W	Description	Reset Value
UART_WKCTL x=0,1,2,3,4	UARTx_BA+0x40	R/W	UART Wake-up Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved			WKTOUTEN	WKRS485EN	WKRFR TEN	WKDATEN	WKCTSEN

Bits	Description
[31:5]	Reserved Reserved.
[4]	WKTOUTEN Received Data FIFO Reached Threshold Time-out Wake-up Enable Bit 0 = Received Data FIFO reached threshold time-out wake-up system function Disabled. 1 = Received Data FIFO reached threshold time-out wake-up system function Enabled. Note 1: When the system is in Power-down mode, Received Data FIFO reached threshold time-out will wake up system from Power-down mode. Note 2: It is suggested the function is enabled when the WKRFR TEN (UART_WKCTL[2]) is set to 1.
[3]	WKRS485EN RS-485 Address Match Wake-up Enable Bit 0 = RS-485 Address Match (AAD mode) wake-up system function Disabled. 1 = RS-485 Address Match (AAD mode) wake-up system function Enabled. Note 1: When the system is in Power-down mode, RS-485 Address Match will wake up system from Power-down mode. Note 2: This bit is used for RS-485 Auto Address Detection (AAD) mode in RS-485 function mode and ADDR DEN (UART_ALTCTL[15]) is set to 1.
[2]	WKRFR TEN Received Data FIFO Reached Threshold Wake-up Enable Bit 0 = Received Data FIFO reached threshold wake-up system function Disabled. 1 = Received Data FIFO reached threshold wake-up system function Enabled. Note: When the system is in Power-down mode, Received Data FIFO reached threshold will wake-up system from Power-down mode.
[1]	WKDATEN Incoming Data Wake-up Enable Bit 0 = Incoming data wake-up system function Disabled. 1 = Incoming data wake-up system function Enabled. Note: When the system is in Power-down mode, incoming data will wake-up system from Power-down mode.
[0]	WKCTSEN nCTS Wake-up Enable Bit 0 = nCTS Wake-up system function Disabled. 1 = nCTS Wake-up system function Enabled. Note: When the system is in Power-down mode, an external nCTS change will wake up system from

		Power-down mode.

UART Wake-up Status Register (UART_WKSTS)

Register	Offset	R/W	Description	Reset Value
UART_WKSTS x=0,1,2,3,4	UARTx_BA+0x44	R/W	UART Wake-up Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved			TOUTWKF	RS485WKF	RFRTWKF	DATWKF	CTSWKF

Bits	Description
[31:5]	Reserved Reserved.
[4]	TOUTWKF Received Data FIFO Threshold Time-out Wake-up Flag This bit is set if chip wake-up from power-down state by Received Data FIFO Threshold Time-out wake-up. 0 = Chip stays in power-down state. 1 = Chip wake-up from power-down state by Received Data FIFO reached threshold time-out. Note 1: If WKOUTEN (UART_WKCTL[4]) is enabled, the Received Data FIFO reached threshold time-out wake-up cause this bit is set to '1'. Note 2: This bit can be cleared by writing '1' to it.
[3]	RS485WKF RS-485 Address Match Wake-up Flag This bit is set if chip wake-up from power-down state by RS-485 Address Match (AAD mode). 0 = Chip stays in power-down state. 1 = Chip wake-up from power-down state by RS-485 Address Match (AAD mode) wake-up. Note 1: If WKRS485EN (UART_WKCTL[3]) is enabled, the RS-485 Address Match (AAD mode) wake-up cause this bit is set to '1'. Note 2: This bit can be cleared by writing '1' to it.
[2]	RFRTWKF Received Data FIFO Reached Threshold Wake-up Flag This bit is set if chip wake-up from power-down state by Received Data FIFO reached threshold wake-up. 0 = Chip stays in power-down state. 1 = Chip wake-up from power-down state by Received Data FIFO Reached Threshold wake-up. Note 1: If WKRFRTEN (UART_WKCTL[2]) is enabled, the Received Data FIFO Reached Threshold wake-up cause this bit is set to '1'. Note 2: This bit can be cleared by writing '1' to it.
[1]	DATWKF Incoming Data Wake-up Flag This bit is set if chip wake-up from power-down state by data wake-up. 0 = Chip stays in power-down state. 1 = Chip wake-up from power-down state by Incoming Data wake-up.

		Note 1: If WKDATEN (UART_WKCTL[1]) is enabled, the Incoming Data wake-up cause this bit is set to '1'. Note 2: This bit can be cleared by writing '1' to it.
[0]	CTSWKF	nCTS Wake-up Flag This bit is set if chip wake-up from power-down state by nCTS wake-up. 0 = Chip stays in power-down state. 1 = Chip wake-up from power-down state by nCTS wake-up. Note 1: If WKCTSEN (UART_WKCTL[0]) is enabled, the nCTS wake-up cause this bit is set to '1'. Note 2: This bit can be cleared by writing '1' to it.

UART Incoming Data Wake-up Compensation Register (UART_DWKCOMP)

Register	Offset	R/W	Description	Reset Value
UART_DWKCOMP x=0,1,2,3,4	UARTx_BA+0x48	R/W	UART Incoming Data Wake-up Compensation Register	0x0000_0000

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

Bits	Description
[31:16]	Reserved Reserved.
[15:0]	STCOMP START Bit Compensation Value These bits field indicate how many clock cycle selected by UART_CLK do the UART controller can get the 1 st bit (START bit) when the device is woken up from Power-down mode. Note: It is valid only when WKDATEN (UART_WKCTL[1]) is set.

6.13 Serial Peripheral Interface (SPI)

6.13.1 Overview

The Serial Peripheral Interface (SPI) applies to synchronous serial data communication and allows full duplex transfer. Devices communicate in Master/Slave mode with the 4-wire bi-direction interface. The chip contains one SPI controller performing a serial-to-parallel conversion on data received from a peripheral device, and a parallel-to-serial conversion on data transmitted to a peripheral device. The SPI controller can be configured as a master or a slave device and supports the PDMA function to access the data buffer. The SPI controller also supports I²S mode to connect external audio CODEC. Please refer to the M253 Datasheet for detailed information about maximum SPI clock frequency of SPI master mode and SPI slave mode and range of SPI operation voltage.

6.13.2 Features

- SPI Mode
 - Supports one SPI controller
 - Supports Master or Slave mode operation
 - Configurable bit length of a transaction word from 8 to 32-bit
 - Provides separate 4-level depth transmit and receive FIFO buffers
 - Supports MSB first or LSB first transfer sequence
 - Supports Byte Reorder function
 - Supports Byte or Word Suspend mode
 - Supports PDMA transfer
 - Supports one data channel half-duplex transfer
 - Supports receive-only mode
- I²S Mode
 - Supports Master or Slave
 - Capable of handling 8-, 16-, 24- and 32-bit word sizes
 - Provides two 4-level FIFO data buffers, one for transmitting and the other for receiving
 - Supports monaural and stereo audio data
 - Supports PCM mode A, PCM mode B, I²S and MSB justified data format
 - Supports two PDMA requests, one for transmitting and the other for receiving

6.13.3 Block Diagram

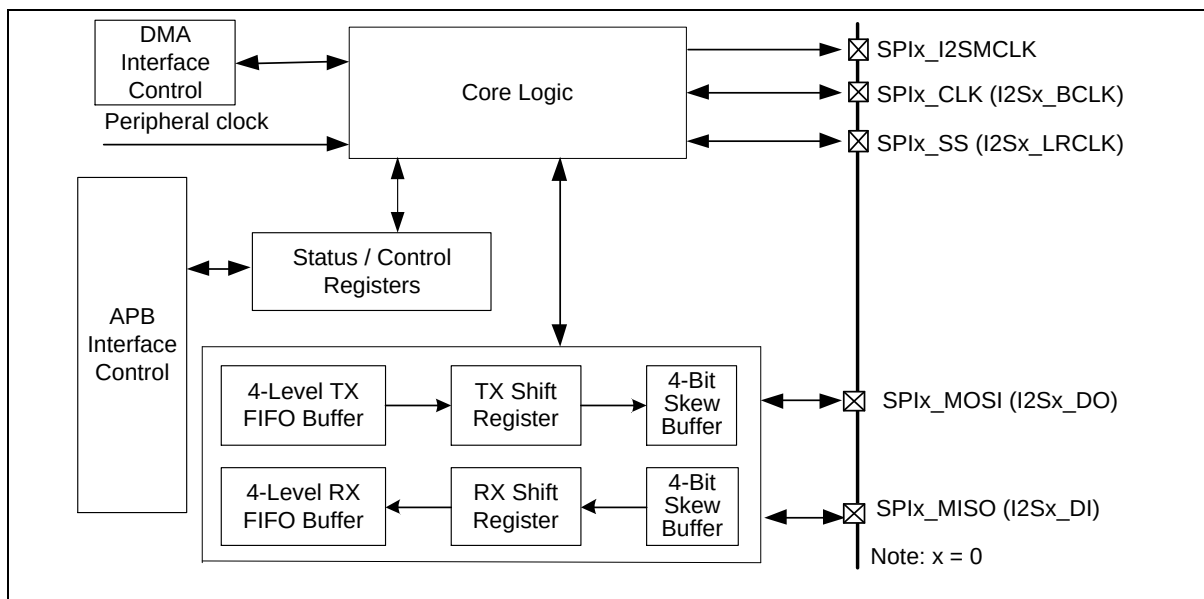


Figure 6.13-1 SPI Block Diagram

TX FIFO Buffer:

The transmit FIFO buffer is a 4-level depth, 32-bit wide, first-in, first-out register buffer. The data can be written to the transmit FIFO buffer in advance through software by writing the SPIx_TX register. In SPI mode, the transmit FIFO will be configured as 8-level while data length is set as 8~16 bits.

RX FIFO Buffer:

The receive FIFO buffer is also a 4-level depth, 32-bit wide, first-in, first-out register buffer. The receive control logic will store the receive data to this buffer. The FIFO buffer data can be read from SPIx_RX register by software. In SPI mode, the receive FIFO will be configured as 8-level while data length is set as 8~16 bits.

TX Shift Register:

The transmit shift register is a 32-bit wide register buffer. The transmit data is loaded from the TX FIFO buffer and shifted out bit-by-bit to the skew buffer.

RX Shift Register:

The receive shift register is also a 32-bit wide register buffer. The receive data is shift in bit-by-bit from the skew buffer and is loaded into RX FIFO buffer when a transaction done.

Skew Buffer:

The skew buffer is a 4-level 1-bit buffer. There are two skew buffers in transmitting and received side. In received side, it is used to shift bits into RX shift register from SPI bus. In transmitting side, it is used to shift bits into SPI bus from TX shift register.

6.13.4 Basic Configuration

6.13.4.1 SPI0 Basic Configuration

- Clock source Configuration
 - Select the source of SPI0 peripheral clock on SPI0SEL (CLK_CLKSEL2[5:4]).

- Enable SPI0 peripheral clock in SPI0CKEN (CLK_APBCLK0[13]).
- Reset Configuration
 - Reset SPI0 controller in SPI0RST (SYS_IPRST1[13]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
SPI0	SPI0_CLK	PA.2, PB.14	MFP4
	SPI0_I2SMCLK	PA.4, PC.14	MFP4
		PB.0	MFP8
	SPI0_MISO	PA.1, PB.13	MFP4
	SPI0_MOSI	PA.0, PB.12	MFP4
	SPI0_SS	PA.3, PB.15	MFP4

SPI/I²S (SPI0) Interface Controller Pin description is shown as follows:

Pin	SPI Mode	I ² S Mode
SPIx_SS	SPI slave selection pin	I ² S left/right channel synchronization clock pin (I2Sx_LRCLK)
SPIx_CLK	SPI clock pin	I ² S bit clock pin (I2Sx_BCLK)
SPIx_MISO	SPI master input or slave output pin	I ² S data input pin (I2Sx_DI)
SPIx_MOSI	SPI master output or slave input pin	I ² S data output pin (I2Sx_DO)
SPIx_I2SMCLK	Not available	I ² S Master clock output pin

Table 6.13-1 SPI/I²S Interface Controller Pin Description (SPI0)

6.13.5 Functional Description

6.13.5.1 Terminology

SPI Peripheral Clock and SPI Bus Clock

The SPI controller needs the peripheral clock to drive the SPI logic unit to perform the data transfer. The peripheral clock rate is determined by the settings of clock divisor (SPIx_CLKDIV) and the clock source which can be HXT, PCLK or HIRC. SPIxSEL of CLK_CLKSEL2 register determines the clock source of the peripheral clock. The DIVIDER (SPIx_CLKDIV[8:0]) setting determines the divisor of the clock rate calculation.

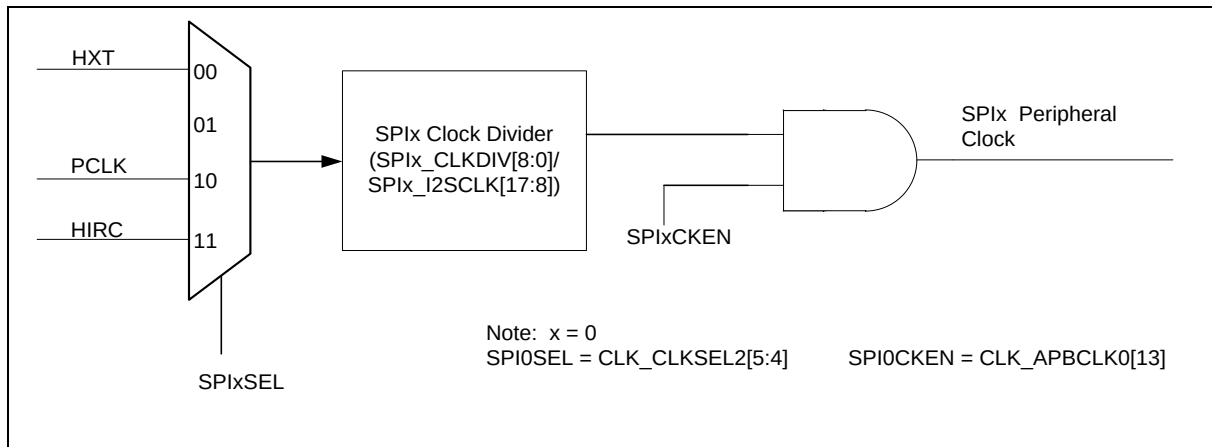


Figure 6.13-2 SPI Peripheral Clock

In Master mode, the frequency of the SPI bus clock is equal to the peripheral clock rate. In general, the SPI bus clock is denoted as SPI clock. In Slave mode, the SPI bus clock is provided by a master device. The frequency of SPI peripheral clock cannot be faster than the system clock rate regardless of Master or Slave mode.

In I²S mode, the peripheral clock rate is equal to I²S bit clock rate determined by SPIx_I2SCLK register.

Master/Slave mode

The SPI controllers can be set as Master or Slave mode by setting the SLAVE (SPIx_CTL[18]) to communicate with the off-chip SPI slave or master device. The HALFDPX (SPIx_CTL[14]) can be used to select the full-duplex or half-duplex in SPI transmission. The application block diagrams in Master and Slave mode are shown below.

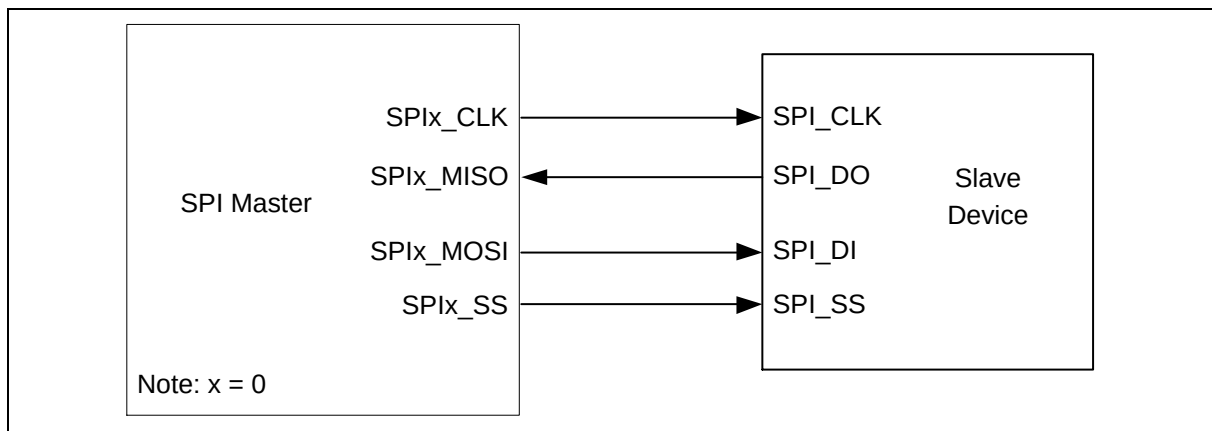


Figure 6.13-3 SPI Full-Duplex Master Mode Application Block Diagram

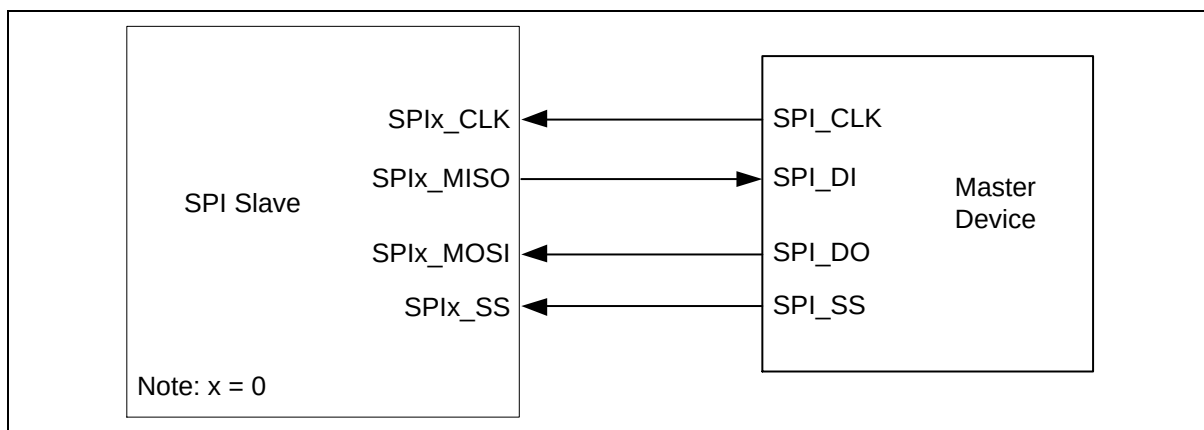


Figure 6.13-4 SPI Full-Duplex Slave Mode Application Block Diagram

Slave Selection

In Master mode, the SPI controller can drive off-chip slave device through the slave select output pin SPIx_SS. In Slave mode, the off-chip master device drives the slave selection signal from the SPIx_SS input port to this SPI controller. The duration between the slave select active edge and the first SPI clock input shall over 3 SPI peripheral clock cycles of slave.

In Master/Slave mode, the active state of slave selection signal can be programmed to low or high active in SSACTPOL (SPIx_SSCTL[2]). The selection of slave select conditions depends on what type of device is connected. In Slave mode, to recognize the inactive state of the slave selection signal, the inactive period of the slave selection signal must be larger than or equal to 3 peripheral clock cycles between two successive transactions.

In Slave mode, the SPIx_MISO signal type is controlled by the SPIx_SS pin. If the SPIx_SS pin is inactive state, the SPIx_MISO is set to tri-state to avoid interference with other Slave devices. If the SPIx_SS pin is active state, the SPIx_MISO is set to output mode for data transfer.

Timing Condition

The CLKPOL (SPIx_CTL[3]) defines the SPI clock idle state. If CLKPOL = 1, the output SPI clock is idle at high state; if CLKPOL = 0, it is idle at low state.

TXNEG (SPIx_CTL[2]) defines the data transmitted out either on negative edge or on positive edge of SPI clock. RXNEG (SPIx_CTL[1]) defines the data received either on negative edge or on positive edge of SPI clock.

Note: The settings of TXNEG and RXNEG are mutual exclusive. In other words, do not transmit and receive data at the same clock edge.

Transmit/Receive Bit Length

The bit length of a transaction word is defined in DWIDTH (SPIx_CTL[12:8]) and can be configured up to 32-bit length in a transaction word for transmitting and receiving.

When SPI controller finishes a transaction, i.e. receives or transmits a specific count of bits defined in DWIDTH (SPIx_CTL[12:8]), the unit transfer interrupt flag UNITIF (SPIx_STATUS[1]) will be set to 1.

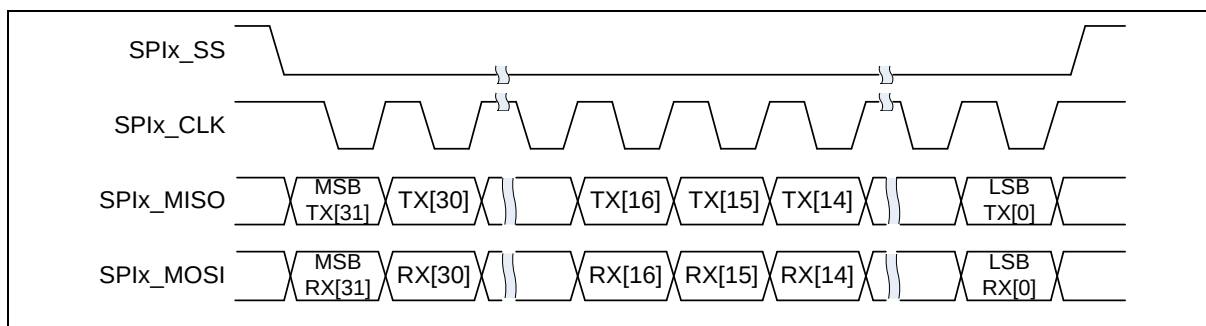


Figure 6.13-5 32-bit in One Transaction

LSB/MSB First

LSB (SPIx_CTL[13]) defines the bit transfer sequence in a transaction. If the LSB (SPIx_CTL[13]) is set to 1, the transfer sequence is LSB first. The bit 0 will be transferred firstly. If the LSB (SPIx_CTL[13]) is cleared to 0, the transfer sequence is MSB first.

Suspend Interval

SUSPITV (SPIx_CTL[7:4]) provides a configurable suspend interval, 0.5 ~ 15.5 SPI clock periods, between two successive transaction words in Master mode. The definition of the suspend interval is the interval between the last clock edge of the preceding transaction word and the first clock edge of the following transaction word. The default value of SUSPITV is 0x3 (3.5 SPI clock cycles).

6.13.5.2 Automatic Slave Selection

In Master mode, if AUTOSS (SPIx_SSCTL[3]) is set, the slave selection signal will be generated automatically and output to the SPIx_SS pin according to whether SS (SPIx_SSCTL[0]) is enabled or not. The slave selection signal will be set to active state by the SPI controller when the SPI data transfer is started by writing to FIFO. It will be set to inactive state when SPI bus is idle. If SPI bus is not idle, i.e. TX FIFO, TX shift register or TX skew buffer is not empty, the slave selection signal will be set to inactive state between transactions if the value of SUSPITV (SPIx_CTL[7:4]) is greater than or equal to 3.

In Master mode, if the value of SUSPITV is less than 3 and the AUTOSS is set as 1, the slave selection signal will be kept at active state between two successive transactions.

If the AUTOSS bit is cleared, the slave selection output signal will be determined by the SS setting. The active state of the slave selection output signal is specified in SSACTPOL (SPIx_SSCTL[2]).

The duration between the slave selection signal active edge and the first SPI bus clock edge is 1 SPI bus clock cycle and the duration between the last SPI bus clock and the slave selection signal inactive edge is 1.5 SPI bus clock cycle.

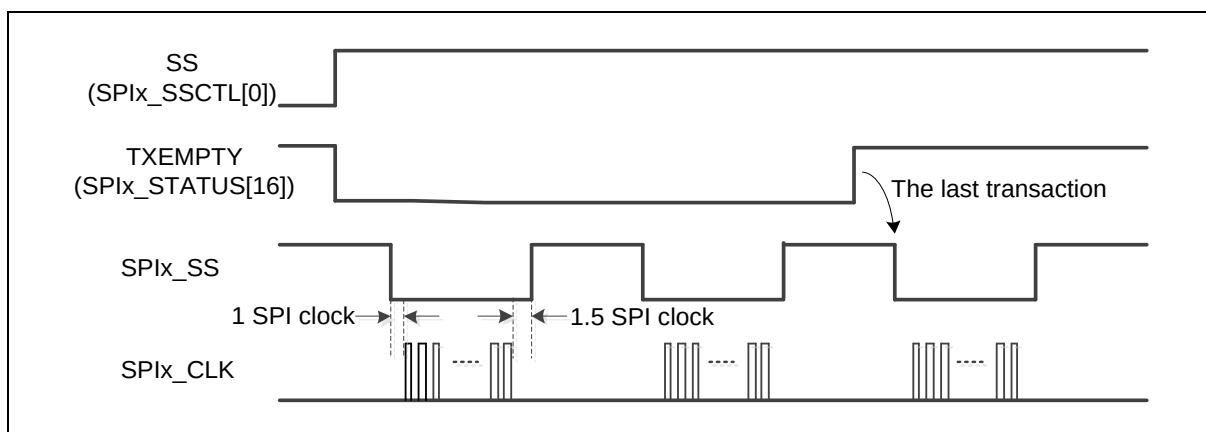


Figure 6.13-6 Automatic Slave Selection (SSACTPOL = 0, SUSPITV > 0x2)

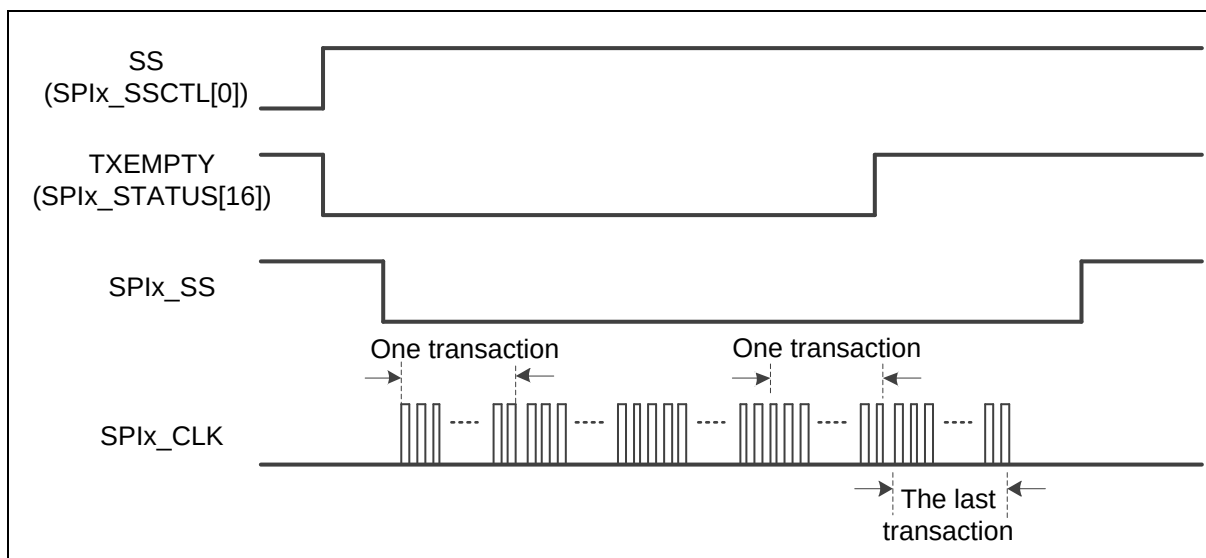


Figure 6.13-7 Automatic Slave Selection (SSACTPOL = 0, SUSPITV < 0x3)

6.13.5.3 Byte Reorder and Suspend Function

When the transfer is set as MSB first (LSB = 0) and the REORDER (SPIx_CTL[19]) is set to 1, the data stored in the TX buffer and RX buffer will be rearranged in the order as [Byte0, Byte1, Byte2, Byte3] in 32-bit transfer (DWIDTH = 0). The sequence of transmitted/received data will be Byte0, Byte1, Byte2, and then Byte3. If the DWIDTH is set as 24-bit transfer mode, the data in TX buffer and RX buffer will be rearranged as [unknown byte, Byte0, Byte1, Byte2]. The SPI controller will transmit/receive data with the sequence of Byte0, Byte1 and then Byte2. Each byte will be transmitted/received with MSB first. The rule of 16-bit mode is the same as above. Byte Reorder function is only available when DWIDTH is configured as 16, 24, and 32 bits.

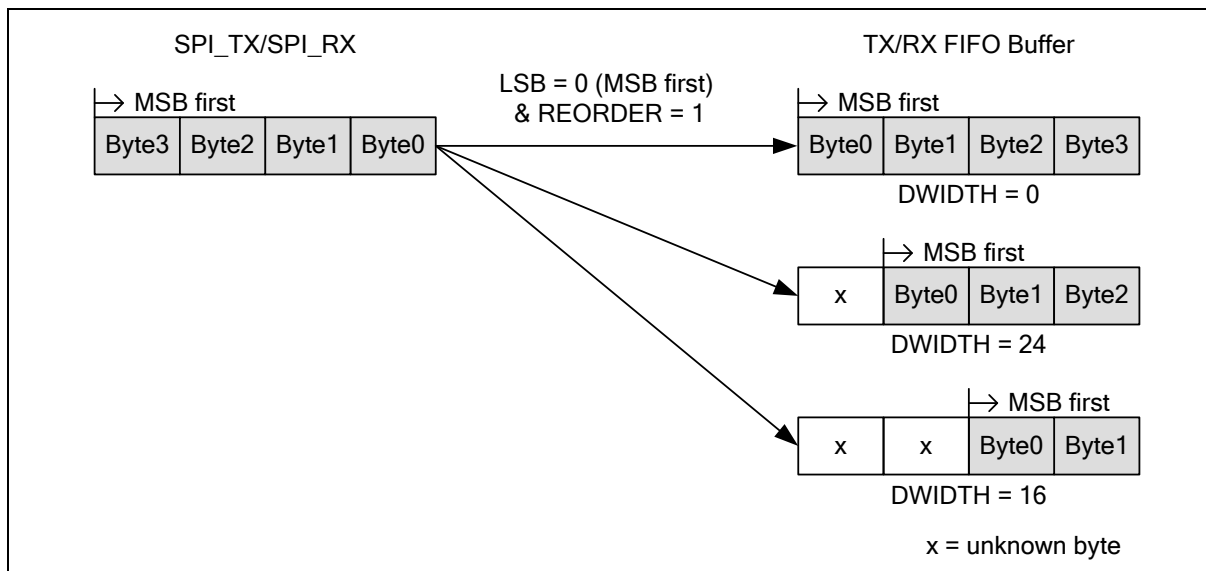


Figure 6.13-8 Byte Reorder Function

In Master mode, if REORDER (SPIx_CTL[19]) is set to 1, a suspend interval of 0.5 ~ 15.5 SPI clock periods will be inserted by hardware between two successive bytes in a transaction word. The suspend interval is configured in SUSPITV (SPIx_CTL[7:4]).

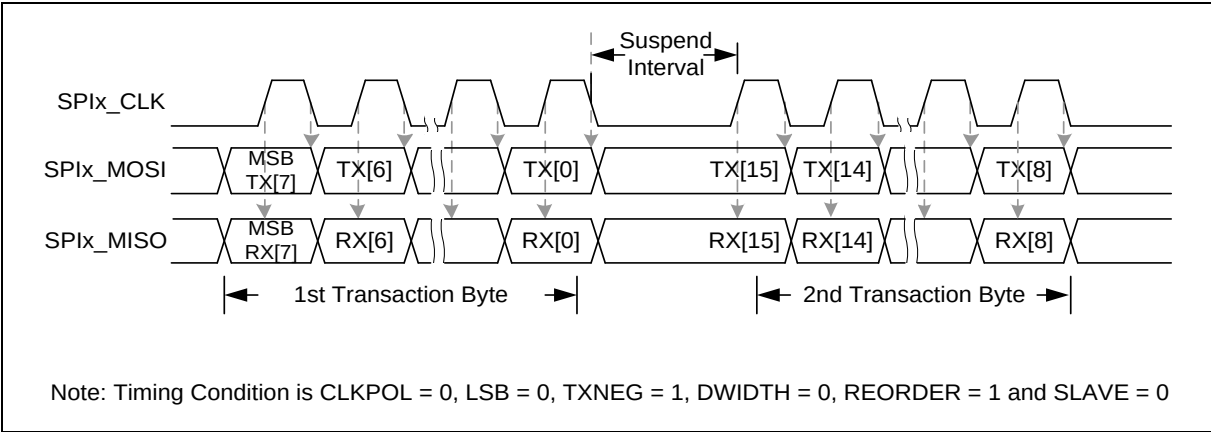


Figure 6.13-9 Timing Waveform for Byte Suspend

6.13.5.4 Half-Duplex Communication

The SPI controller can communicate in half-duplex mode by setting HALFDPX (SPIx_CTL[14]) bit. In half-duplex mode, there is only one data line for receiving or transmitting data direction which is defined by DATDIR (SPIx_CTL[20]). In half-duplex configuration, the SPIx_MISO pin is free for other applications and it can be configured as GPIO. Enabling or disabling the control bit HALFDPX (SPIx_CTL[14]) will produce TXFBCLR (SPIx_FIFOCCTL[9]) and RXFBCLR (SPIx_FIFOCCTL[8]) at the same time automatically.

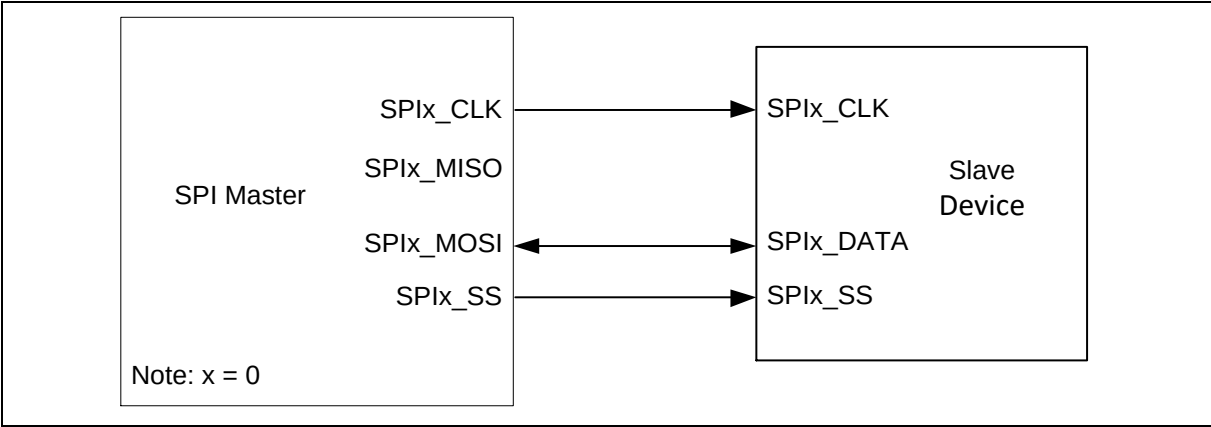


Figure 6.13-10 SPI Half-Duplex Master Mode Application Block Diagram

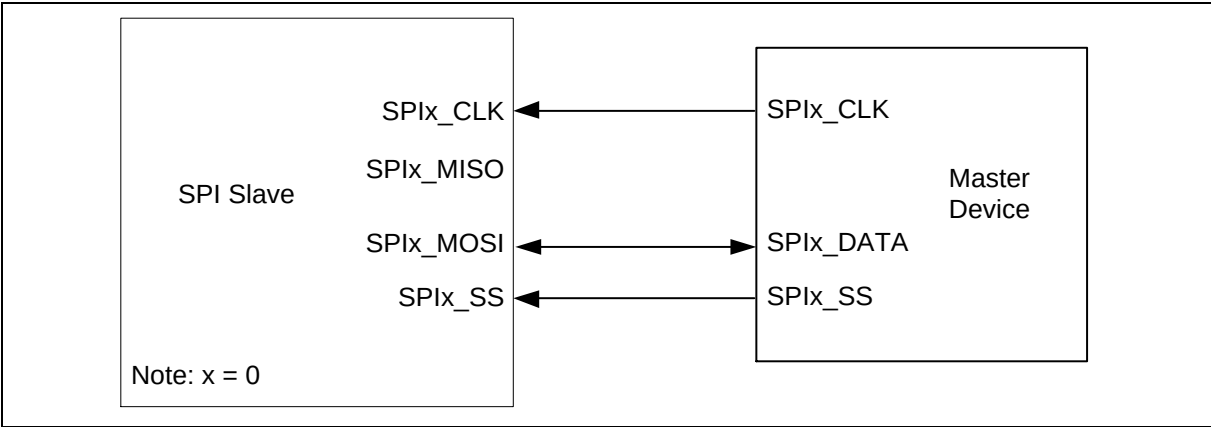


Figure 6.13-11 SPI Half-Duplex Slave Mode Application Block Diagram

6.13.5.5 Receive-Only Mode

In SPI Master device, it can communicate in receive-only mode by setting RXONLY (SPIx_CTL[15]). In this configuration, the SPI Master device will generate SPI bus clock continuously as long as the receive-only mode is enabled for receiving data bit from SPI slave device. If AUTOSS (SPIx_SSCTL[3]) is enabled in receive-only mode, SPI Master will keep activating the slave select signal.

The remaining SPIx_MOSI pin of SPI Master device is not used for communication and can be configured as GPIO. The status BUSY (SPIx_STATUS[0]) will be asserted in receive-only mode due to the generation of SPI bus clock. Entering this mode will produce the TXFBCLR (SPIx_FIFCTL[9]) and RXFBCLR (SPIx_FIFCTL[8]) at the same time automatically. When user enables this mode, the output SPI bus clock will be sent out after 6 peripheral clock cycles. In this mode, the data which has been written into transmit FIFO will be loaded into transmit shift register and sent out.

When user sets RXONLY (SPIx_CTL[15]) enable, SPI RX data with data bit width of DWIDTH (SPIx_CTL[12:8]) will be received into RX FIFO and SPI clock will be sent to SPI slave device until RX FIFO is full.

For data bit width of 8~16 bits, the SPI master will send SPI output clock to SPI slave and receive RX data when RX FIFO counter RXCNT (SPIx_STATUS[27:24]) is less than or equal to 6.

For data bit width of 17~32 bits, the SPI master will send SPI output clock to SPI slave and receive RX data when RX FIFO counter RXCNT (SPIx_STATUS[27:24]) is less than or equal to 2.

6.13.5.6 PDMA Transfer Function

The SPI controller supports PDMA transfer function.

When TXPDMAEN (SPIx_PDMACTL[0]) is set to 1, the controller will issue request to PDMA controller to start the PDMA transmission process automatically.

When RXPDMAEN (SPIx_PDMACTL[1]) is set to 1, the controller will start the PDMA reception process. SPI controller will issue request to PDMA controller automatically when there is data in the RX FIFO buffer.

When PDMA transfer is done, user needs to disable TXPDMAEN/RXPDMAEN. After re-setting control registers of PDMA, user enables TXPDMAEN/RXPDMAEN again.

Note: The SPI supports single request PDMA (Read/Write) only, burst request PDMA is not supported.

6.13.5.7 FIFO Buffer Operation

The SPI controllers equip with four 32-bit wide transmit and receive FIFO buffers. The data stored in the transmit FIFO buffer will be read and sent out by the transmission control logic. If the transmit FIFO buffer is full, the TXFULL (SPIx_STATUS[17]) will be set to 1. When the SPI transmission logic unit draws out the last datum of the transmit FIFO buffer, so that the transmit FIFO buffer is empty, the TXEMPTY (SPIx_STATUS[16]) will be set to 1. Note that the TXEMPTY (SPIx_STATUS[16]) flag is set to 1 while the last transaction is still in progress. In Master mode, the BUSY (SPIx_STATUS[0]) is set to 1 when the FIFO buffer is written any data or there is any transaction on the SPI bus. (e.g. the slave selection signal is active and the SPI controller is receiving data in Slave mode). It will set to 0 when the transmit FIFO is empty and the current transaction has done. Thus, the status of BUSY (SPIx_STATUS[0]) should be checked by software to make sure whether the SPI is in idle or not.

The receive control logic will store the SPI input data into the receive FIFO buffer. There are FIFO related status bits, like RXEMPTY (SPIx_STATUS[8]) and RXFULL (SPIx_STATUS[9]), to indicate the current status of RX FIFO buffer.

The transmitting and receiving threshold can be configured by setting TXTH (SPIx_FIFCTL[30:28]) and RXTH (SPIx_FIFCTL[26:24]). When the count of valid data stored in transmit FIFO buffer is less than or equal to TXTH (SPIx_FIFCTL[30:28]) setting, TXTHIF (SPIx_STATUS[18]) will be set to 1. When the count of valid data stored in receive FIFO buffer is larger than RXTH (SPIx_FIFCTL[26:24]) setting, RXTHIF (SPIx_STATUS[10]) will be set to 1.

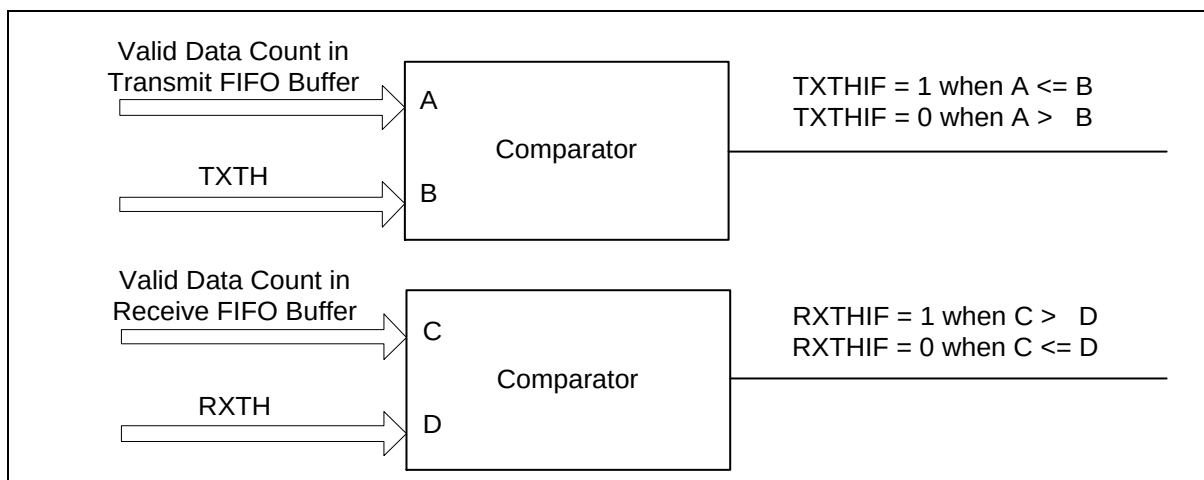


Figure 6.13-12 FIFO Threshold Comparator

In Master mode, when the first datum is written to the SPIx_TX register, the TXEMPTY flag (SPIx_STATUS[16]) will be cleared to 0. The transmission will start after 1 PCLK clock cycles and 6 peripheral clock cycles. User can write the next data into SPIx_TX register immediately. The SPI controller will insert a suspend interval between two successive transactions. The period of suspend interval is decided by the setting of SUSPITV (SPIx_CTL[7:4]). If the SUSPITV (SPIx_CTL[7:4]) equals 0, SPI controller can perform continuous transfer. User can write data into SPIx_TX register as long as the TXFULL (SPIx_STATUS[17]) is 0.

When user sets DWIDTH (SPIx_CTL[12:8]) to 8 bits ~ 16 bits, the FIFO structure will be configured to eight 16-bit wide transmit and receive FIFO buffers automatically.

The Example 1 of Figure 6.13-13 indicates the updated condition of TXEMPTY (SPIx_STATUS[16]) and the relationship among the FIFO buffer, shift register and the skew buffer for 8~16 bits of data length. The TXEMPTY (SPIx_STATUS[16]) is set to 0 when the Data0 is written into the FIFO buffer. The Data0 will be loaded into the shift register by the core logic and the TXEMPTY (SPIx_STATUS[16]) will be to 1. The Data0 in shift register will be shift into skew buffer by bit for transmission until the transfer is done.

The Example 2 of Figure 6.13-13 indicates the updated condition of TXFULL (SPIx_STATUS[17]) when there are 8 data in the FIFO buffer and the next data of Data9 does not be written into the FIFO buffer when the TXFULL = 1.

The example 1 of Figure 6.13-13 indicates the updated condition of TXEMPTY (SPIx_STATUS[16]) and the relationship among the FIFO buffer, shift register and the skew buffer for 17~32 bits of data length.

The Example 2 of Figure 6.13-14 indicates the updated condition of TXFULL (SPIx_STATUS[17]) when there are 4 data in the FIFO buffer and the next data of Data5 is not written into the FIFO buffer when TXFULL = 1.

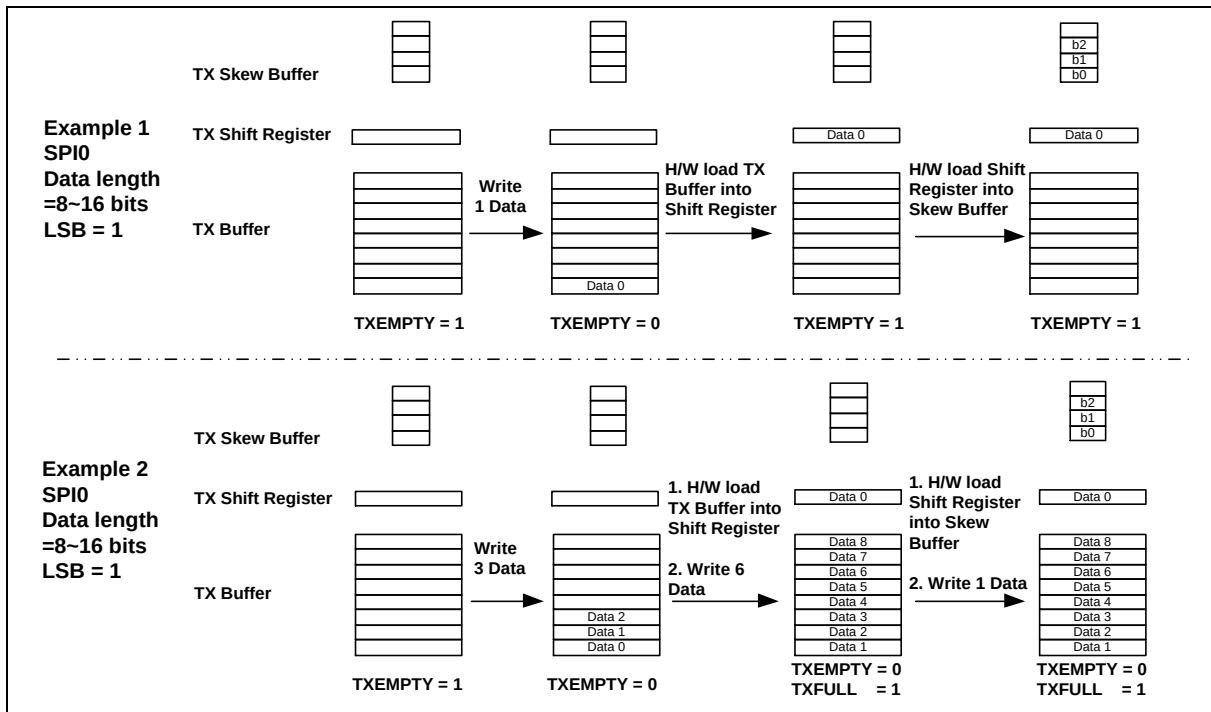


Figure 6.13-13 Transmit FIFO Buffer Example for 8~16 Bits of Data Length

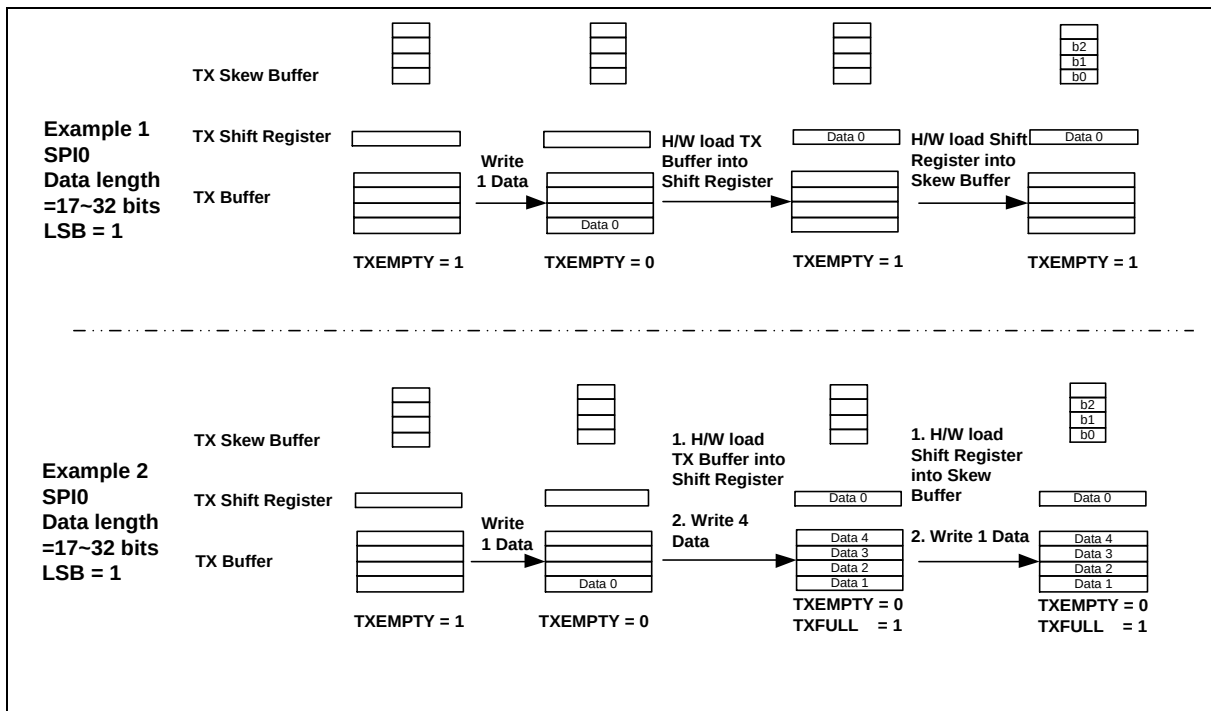


Figure 6.13-14 Transmit FIFO Buffer Example for 17~32 Bits of Data Length

The subsequent transactions will be triggered automatically if the transmitted data are updated in time. If the SPIx_TX register does not be updated after all data transfers are done, the transfer will stop.

In Master mode, during receiving operation, the serial data are received from SPIx_MISO pin and stored to receive FIFO buffer.

The received data (Data0's b0, b1, ...b31) is stored into skew buffer first according the serial clock (SPIx_CLK) and then it is shift into the shift register bit by bit. The core logic will load the data in shift register into FIFO buffer when the received data bit count reaches the value of DWIDTH (SPIx_CTL[12:8]).

The RXEMPTY (SPIx_STATUS[8]) will be cleared to 0 while the receive FIFO buffer contains unread data (see the Example 1 of Receive FIFO Buffer Example in Figure 6.13-15). The received data can be read by software from SPIx_RX register as long as the RXEMPTY (SPIx_STATUS[8]) is 0. If the receive FIFO buffer contains 8 unread data, the RXFULL (SPIx_STATUS[9]) will be set to 1 (see the Example 2 of Receive FIFO Buffer Example in Figure 6.13-15).

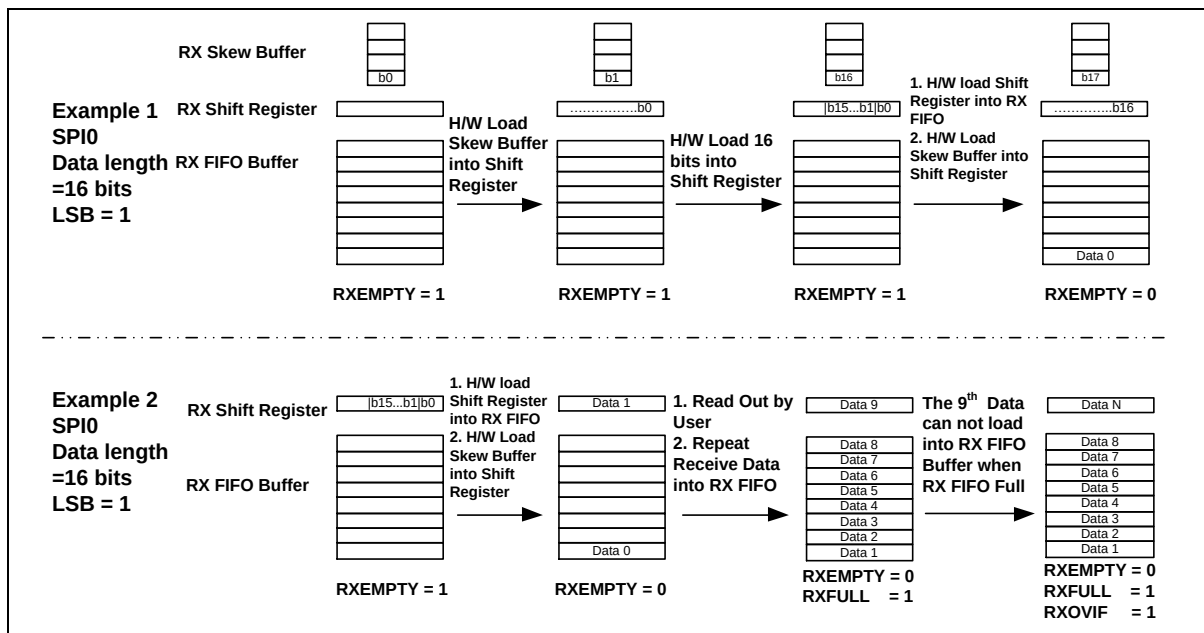


Figure 6.13-15 Receive FIFO Buffer Example for 16 Bits of Data Length

The RXEMPTY (SPIx_STATUS[8]) will be cleared to 0 while the receive FIFO buffer contains unread data (see the Example 1 of Receive FIFO Buffer Example in Figure 6.13-15). The received data can be read by software from SPIx_RX register as long as the RXEMPTY (SPIx_STATUS[8]) is 0. If the receive FIFO buffer contains 4 unread data, the RXFULL (SPIx_STATUS[9]) will be set to 1 (see the Example 2 of Receive FIFO Buffer Example in Figure 6.13-16).

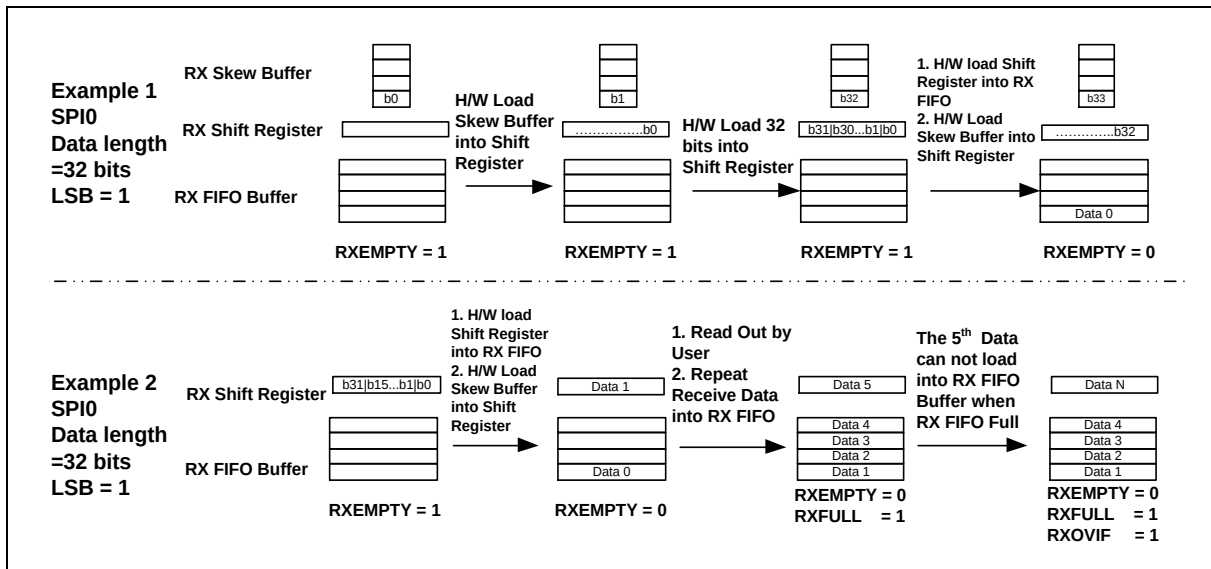


Figure 6.13-16 Receive FIFO Buffer Example for 32 Bits of Data Length

In Slave mode, during transmission operation, when data is written to the SPIx_TX register by software, the data will be loaded into transmit FIFO buffer and the TXEMPTY (SPIx_STATUS[16]) will be set to 0. The transmission will start when the slave device receives clock signal from master. Data can be written to SPIx_TX register as long as the TXFULL (SPIx_STATUS[17]) is 0. After all data have been drawn out by the SPI transmission logic unit and the SPIx_TX register is not updated by software, the TXEMPTY (SPIx_STATUS[16]) will be set to 1.

If there is no any data written to the SPIx_TX register, the transmit underflow interrupt flag, TXUFIF (SPIx_STATUS[19]) will be set to 1 when the slave selection signal is active. The output data will be held by TXUFPOL (SPIx_FIFCTL[6]) setting during this transfer until the slave selection signal goes to inactive state. When the transmit underflow event occurs, the slave under run interrupt flag, SLVURIF (SPIx_STATUS[7]), will be set to 1 as SPIx_SS goes to inactive state.

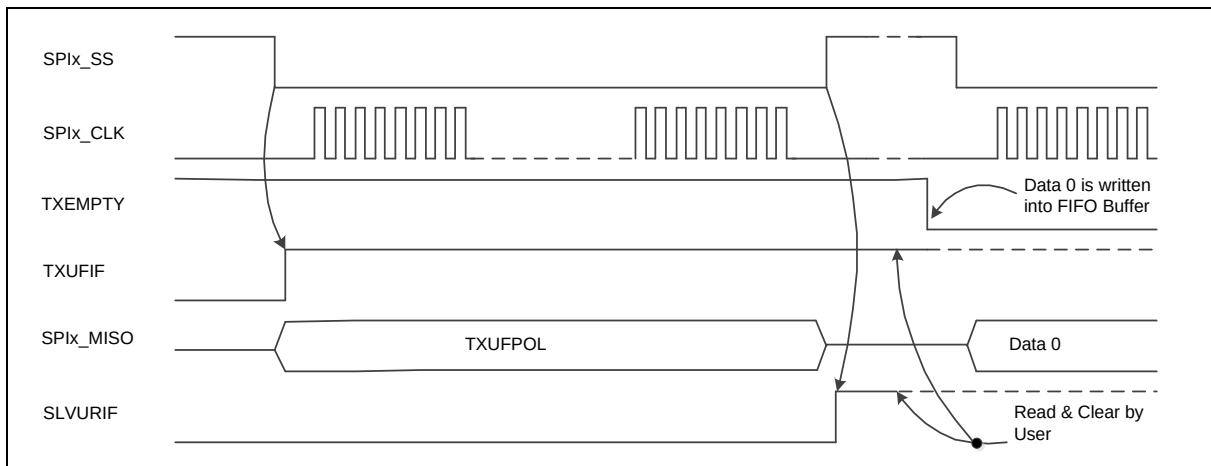


Figure 6.13-17 TX Underflow Event and Slave Under Run Event

In Slave mode, during receiving operation, the serial data is received from SPIx_MOSI pin and stored to SPIx_RX register. The reception mechanism is similar to Master mode reception operation. If the receive FIFO buffer contains 4 unread data, the RXFULL (SPIx_STATUS[9]) will be set to 1 and the RXOVIF (SPIx_STATUS[11]) will be set to 1 if there is more serial data received from SPIx_MOSI and follow-up data will be dropped (refer to the Receive FIFO Buffer Examples in Figure 6.13-15 and Figure 6.13-16). When the receive bit count is mismatched with the DWIDTH (SPIx_CTL[12:8]) and the slave

selection line goes to inactive state, the SLVBEIF (SPIx_STATUS[6]) will be set to 1.

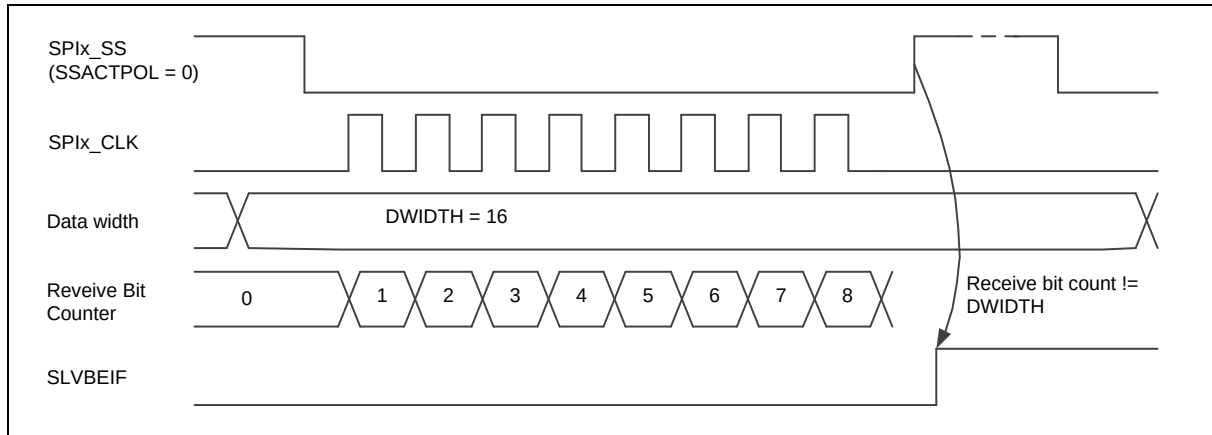


Figure 6.13-18 Slave Mode Bit Count Error

The SPI controller has the receive time-out function. When the receive FIFO is not empty and no read operation in receive FIFO over 64 SPI peripheral clock periods in Master mode or over 576 SPI peripheral clock periods in Slave mode, the receive time-out occurs and the RXTOIF (SPIx_STATUS[12]) will be set to 1. When the receive FIFO is read by user, the time-out status will be cleared automatically.

6.13.5.8 Interrupt

- SPI unit transfer interrupt

As the SPI controller finishes a unit transfer, the unit transfer interrupt flag UNITIF (SPIx_STATUS[1]) will be set to 1. The unit transfer interrupt event will generate an interrupt to CPU if the unit transfer interrupt enable bit UNITIEN (SPIx_CTL[17]) is set. The unit transfer interrupt flag can be cleared only by writing 1 to it.

- SPI slave selection active/inactive interrupt

In Slave mode, the slave selection active/inactive interrupt flag, SSACTIF (SPIx_STATUS[2]) and SSINAIF (SPIx_STATUS[3]), will be set to 1 when the SPIEN (SPIx_CTL[0]) and SLAVE (SPIx_CTL[18]) are set to 1 and the slave selection signal goes to active/inactive state. The SPI controller will issue an interrupt if the SSINAIF (SPIx_SSCTL[13]) or SSACTIEN (SPIx_SSCTL[12]), are set to 1.

- Slave bit count error interrupt

In Slave mode, if the transmit/receive bit count is mismatched with the DWIDTH (SPIx_CTL[12:8]) when the slave selection line goes to inactive state, the SLVBEIF (SPIx_STATUS[6]) will be set to 1. The uncompleted transaction will be dropped from TX and RX shift registers. The SPI controller will issue an interrupt if the SLVBEIEN (SPIx_SSCTL[8]) is set to 1.

Note: If the slave selection signal is active but there is no any serial clock input, the SLVBEIF (SPIx_STATUS[6]) will be set to 1 when the slave selection signal goes to inactive state.

- TX underflow interrupt

In SPI Slave mode, if there is no any data is written to the SPIx_TX register, the TXUFIF (SPIx_STATUS[19]) will be set to 1 when the slave selection signal is active. The SPI controller will issue a TX underflow interrupt if the TXUFIEN (SPIx_FIFCTL[7]) is set to 1.

Note: If underflow event occurs in SPI Slave mode, there are two conditions which make SPI Slave mode return to idle state and then goes for next transfer: (1) set TXRST to 1 (2) slave select signal is changed to inactive state.

- Slave TX under run interrupt

If the TX underflow event occurs, the SLVURIF (SPIx_STATUS[7]) will be set to 1 when SPIx_SS goes to inactive state. The SPI controller will issue a TX under run interrupt if the SLVURIEN (SPIx_SSCTL[9]) is set to 1.

- Receive Overrun interrupt

In Slave mode, if the receive FIFO buffer contains 4 unread data for 17~32 bits of data length, the RXFULL (SPIx_STATUS[9]) will be set to 1 and the RXOVIF (SPIx_STATUS[11]) will be set to 1 if there is more serial data is received from SPI bus and the follow-up data will be dropped. The SPI controller will issue an interrupt if the RXOVIEN (SPIx_FIFCTL[5]) is set to 1.

If the receive FIFO buffer contains 8 unread data for 8~16 bits of data length, the RXFULL (SPIx_STATUS[9]) will be set to 1 and the RXOVIF (SPIx_STATUS[11]) will be set to 1 if there is more serial data received from SPI bus and the follow-up data will be dropped. The SPI controller will issue an interrupt if the RXOVIEN (SPIx_FIFCTL[5]) is set to 1.

- Receive FIFO time-out interrupt

If there is a received data in the FIFO buffer and it is not read by software over 64 SPI peripheral clock periods in Master mode or over 576 SPI peripheral clock periods in Slave mode, it will send a RX time-out interrupt to the system if the RX time-out interrupt enable bit, RXTOIEN (SPIx_FIFCTL[4]), is set to 1.

- Transmit FIFO interrupt

In FIFO mode, if the valid data count of the transmit FIFO buffer is less than or equal to the setting value of TXTH (SPIx_FIFCTL[30:28]), the transmit FIFO interrupt flag TXTHIF (SPIx_STATUS[18]) will be set to 1. The SPI controller will generate a transmit FIFO interrupt to the system if the transmit FIFO interrupt enable bit, TXTHIEN (SPIx_FIFCTL[3]), is set to 1.

- Receive FIFO interrupt

In FIFO mode, if the valid data count of the receive FIFO buffer is larger than the setting value of RXTH (SPIx_FIFCTL[26:24]), the receive FIFO interrupt flag RXTHIF (SPIx_STATUS[10]) will be set to 1. The SPI controller will generate a receive FIFO interrupt to the system if the receive FIFO interrupt enable bit, RXTHIEN (SPIx_FIFCTL[2]), is set to 1.

6.13.5.9 I²S Mode

The SPI0 controllers support I²S mode with PCM mode A, PCM mode B, MSB justified and I²S data format. The bit count of an audio channel is determined by WDWIDTH (SPIx_I2SCTL[5:4]). The transfer sequence is always first from the most significant bit, MSB. Data are read on rising clock edge and are driven on falling clock edge.

In I²S data format, the MSB is sent and latched on the second clock of an audio channel. The I2Sx_LRCLK signal indicates which audio channel is in transferring.

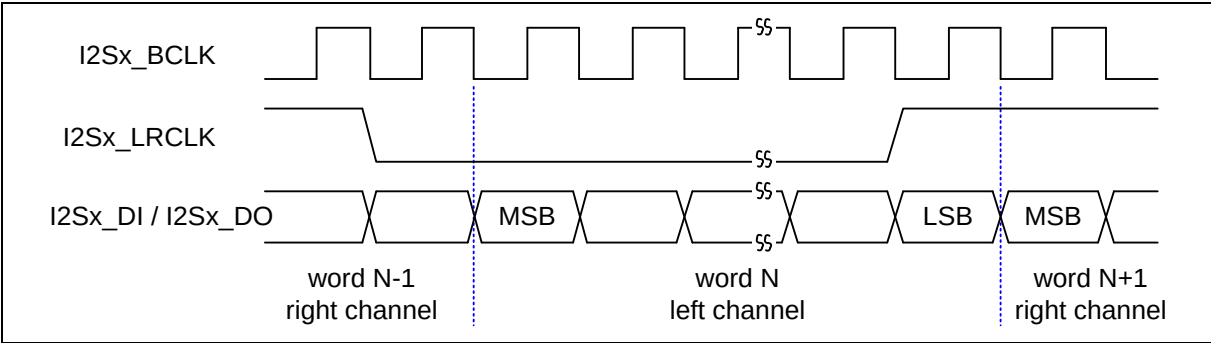


Figure 6.13-19 I²S Data Format Timing Diagram

In MSB justified data format, the MSB is sent and latched on the first clock of an audio channel.

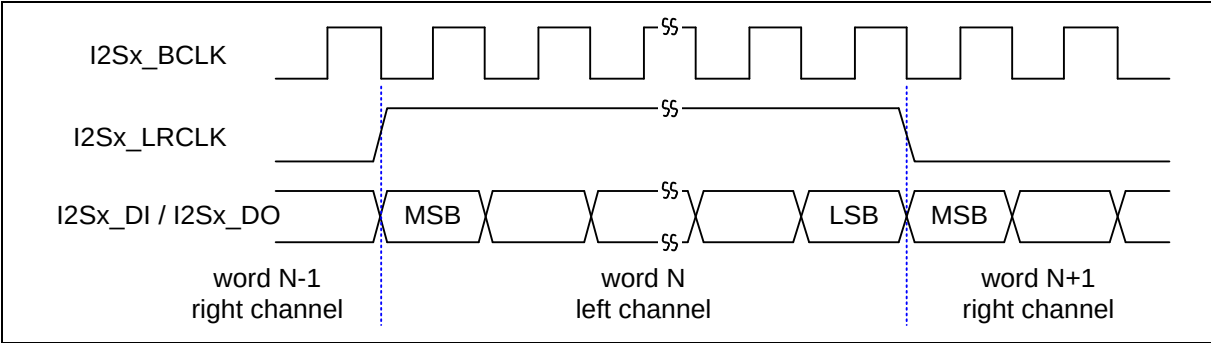


Figure 6.13-20 MSB Justified Data Format Timing Diagram

The I2Sx_LRCLK signal also supports PCM mode A and PCM mode B. The I2Sx_LRCLK signal in PCM mode indicates the beginning of an audio frame.

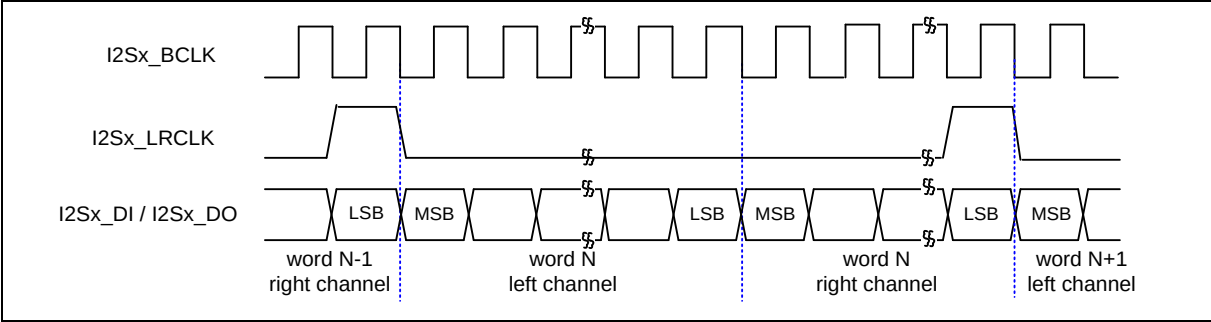


Figure 6.13-21 PCM Mode A Timing Diagram

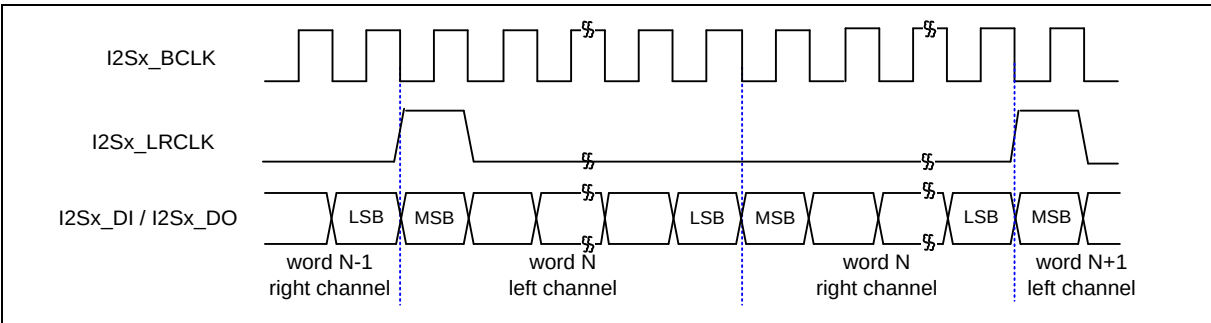


Figure 6.13-22 PCM Mode B Timing Diagram

6.13.5.10 I²S Mode FIFO Operation

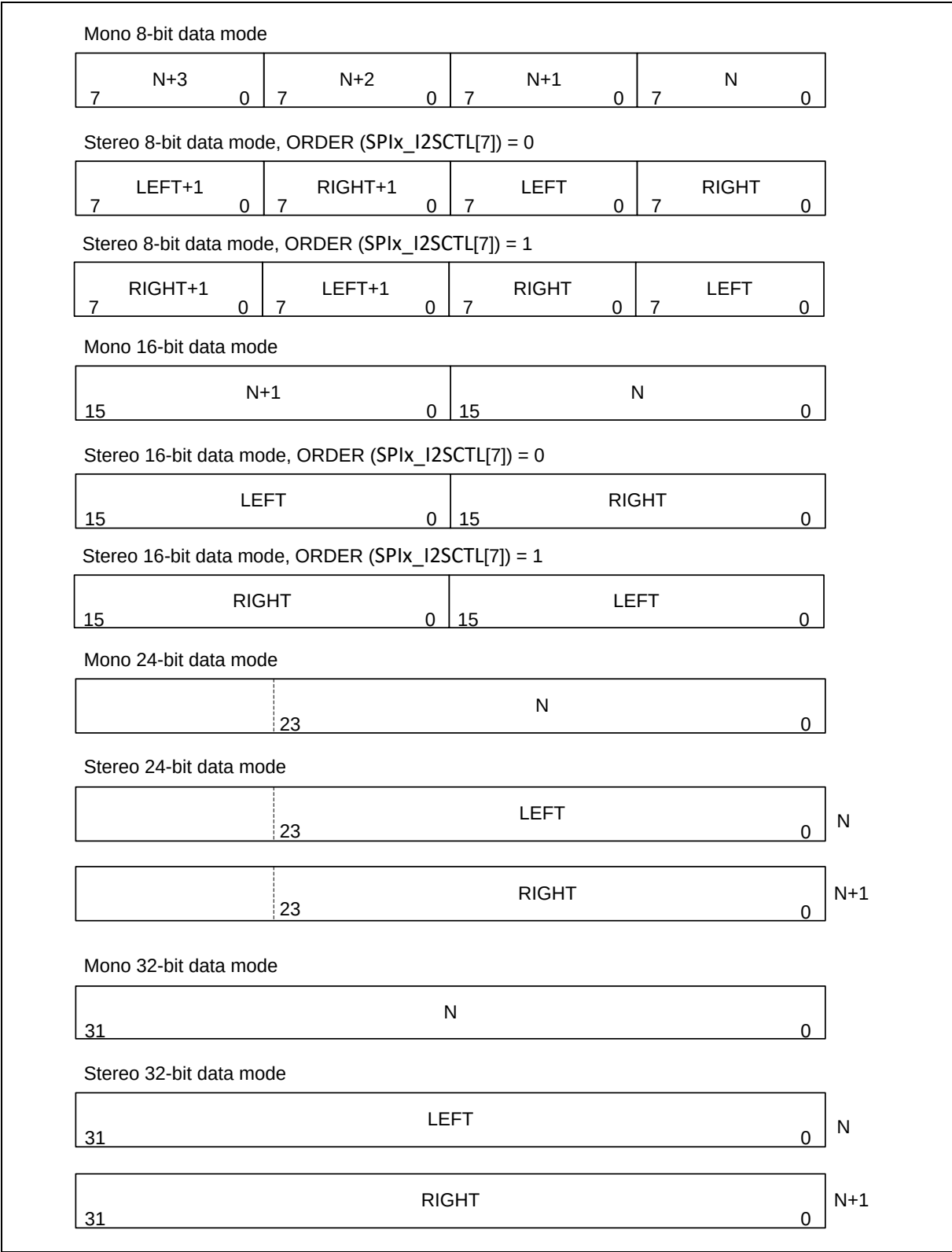


Figure 6.13-23 FIFO Contents for Various I²S Modes

6.13.5.11 Dummy Data Number for I²S / PCM Master mode and Monaural Mode

Before I²S / PCM master starts to send TX data to external slave device, we set control registers I2SEN (SPIx_I2SCTL[0]) enable, TXEN (SPIx_I2SCTL[1]) enable, and write TX data to TX FIFO. After master sends dummy data (data with zero value) to external slave device, master will send TX FIFO data to external slave device. Table 6.13-2 shows number of dummy data for monaural mode, and the unit of dummy data number is L channel + R channel.

Data Width (SPIx_I2SCTL[5:4])	Dummy Data Number (Unit = L Channel + R Channel)
8 bits	0
16 bits	0
24 bits	1
32 bits	1

Table 6.13-2 Dummy Data Number for I²S / PCM Master Mode and Monaural Mode

6.13.5.12 Dummy Data Number for I²S / PCM Master mode and Stereo Mode

Before I²S / PCM master starts to send TX data to external slave device, we set control registers I2SEN (SPIx_I2SCTL[0]) enable, TXEN (SPIx_I2SCTL[1]) enable, and write TX data to TX FIFO. After master sends dummy data (data with zero value) to external slave device, master will send TX FIFO data to external slave device. Table 6.13-3 shows number of dummy data for stereo mode, and the unit of dummy data number is L channel + R channel.

Data Width (SPIx_I2SCTL[5:4])	Dummy Data Number (Unit = L Channel + R Channel)
8 bits	0
16 bits	0
24 bits	1
32 bits	1

Table 6.13-3 Dummy Data Number for I²S / PCM Master Mode and Stereo Mode

6.13.5.13 Dummy Data Number for I²S Slave mode and Monaural Mode

Before I²S / PCM slave starts to send TX data to external master device, we set control registers I2SEN (SPIx_I2SCTL[0]) enable, TXEN (SPIx_I2SCTL[1]) enable, SLAVE mode (SPIx_I2SCTL[8]), and write TX data to TX FIFO. After slave sends dummy data (data with zero value) to external master device, slave will send TX FIFO data to external master device. Table 6.13-4 shows number of dummy data for I²S slave monaural mode, and the unit of dummy data number is L channel + R channel.

Data Width (SPIx_I2SCTL[5:4])	Dummy Data Number (Unit = L Channel + R Channel)
8 bits	3
16 bits	2
24 bits	2
32 bits	2

Table 6.13-4 Dummy Data Number for I²S Slave Mode and Monaural Mode

6.13.5.14 Dummy Data Number for PCM Slave mode and Monaural Mode

Before I²S / PCM slave starts to send TX data to external master device, we set control registers I2SEN (SPIx_I2SCTL[0]) enable, TXEN (SPIx_I2SCTL[1]) enable, SLAVE mode (SPIx_I2SCTL[8]), and write

TX data to TX FIFO. After slave sends dummy data (data with zero value) to external master device, slave will send TX FIFO data to external master device. Table 6.13-5 shows number of dummy data for PCM slave monaural mode, and the unit of dummy data number is L channel + R channel.

Data Width (SPIx_I2SCTL[5:4])	Dummy Data Number (Unit = L Channel + R Channel)
8 bits	2
16 bits	1
24 bits	1
32 bits	1

Table 6.13-5 Dummy Data Number for PCM Slave Mode and Monaural Mode

6.13.5.15 Dummy Data Number for I²S Slave mode and Stereo Mode

Before I²S / PCM slave starts to send TX data to external master device, we set control registers I2SEN (SPIx_I2SCTL[0]) enable, TXEN (SPIx_I2SCTL[1]) enable, SLAVE mode (SPIx_I2SCTL[8]), and write TX data to TX FIFO. After slave sends dummy data (data with zero value) to external master device, slave will send TX FIFO data to external master device. Table 6.13-6 shows number of dummy data for I²S slave stereo mode, and the unit of dummy data number is L channel + R channel.

Data Width (SPIx_I2SCTL[5:4])	Dummy Data Number (Unit = L Channel + R Channel)
8 bits	3
16 bits	2
24 bits	2
32 bits	2

Table 6.13-6 Dummy Data Number for I²S Slave Mode and Stereo Mode

6.13.5.16 Dummy Data Number for PCM Slave mode and Stereo Mode

Before I²S / PCM slave starts to send TX data to external master device, we set control registers I2SEN (SPIx_I2SCTL[0]) enable, TXEN (SPIx_I2SCTL[1]) enable, SLAVE mode (SPIx_I2SCTL[8]), and write TX data to TX FIFO. After slave sends dummy data (data with zero value) to external master device, slave will send TX FIFO data to external master device. Table 6.13-7 shows number of dummy data for PCM slave stereo mode, and the unit of dummy data number is L channel + R channel.

Data Width (SPIx_I2SCTL[5:4])	Dummy Data Number (Unit = L Channel + R Channel)
8 bits	2
16 bits	1
24 bits	1
32 bits	1

Table 6.13-7 Dummy Data Number for PCM Slave Mode and Stereo Mode

6.13.6 Timing Diagram

The active state of slave selection signal can be defined by setting the SSACTPOL (SPIx_SSCTL[2]). The SPI clock which is in idle state can be configured as high or low state by setting the CLKPOL (SPIx_CTL[3]). It also provides the bit length of a transaction word in DWIDTH (SPIx_CTL[12:8]), and transmitting/receiving data from MSB or LSB first in LSB (SPIx_CTL[13]). User can also select which edge of SPI clock to transmit/receive data in TXNEG/RXNEG (SPIx_CTL[2:1]). Four SPI timing

diagrams for master/slave operations and the related settings are shown below.

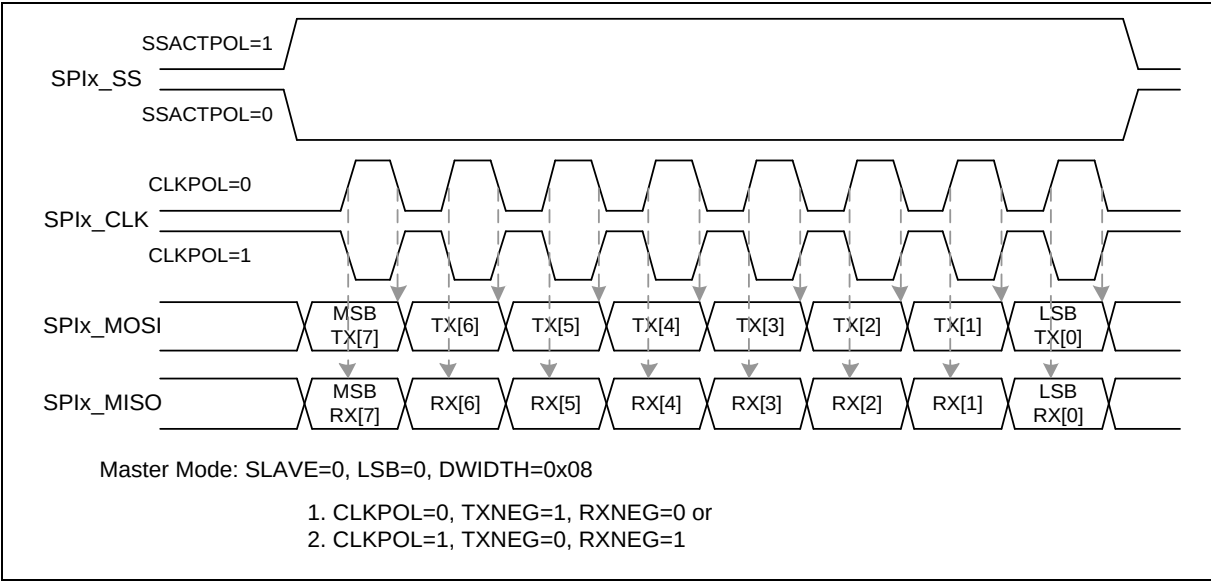


Figure 6.13-24 SPI Timing in Master Mode

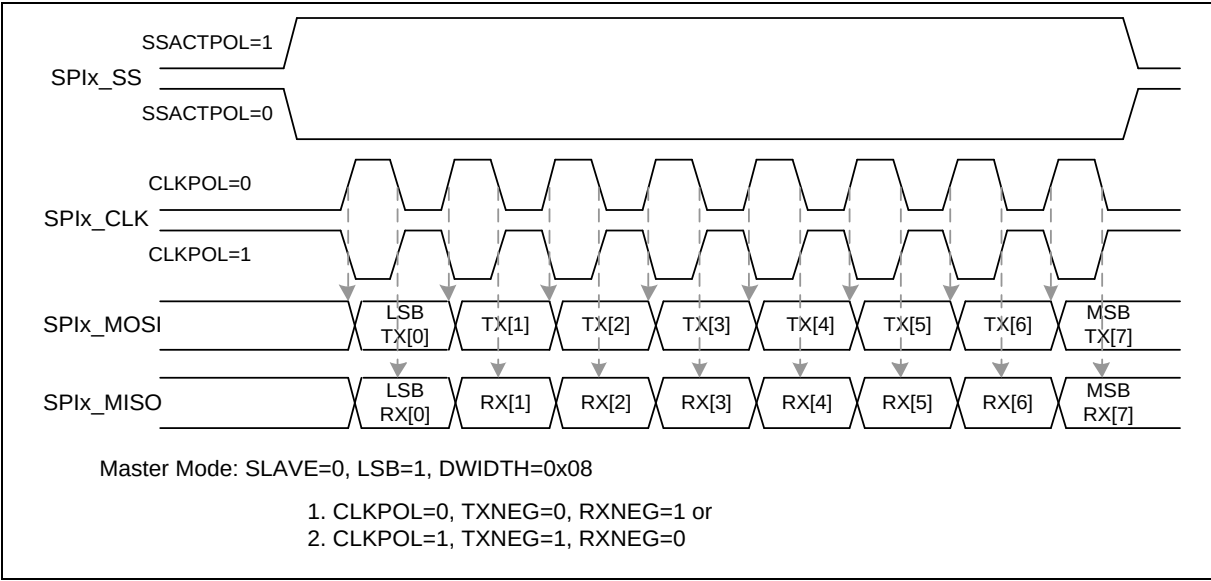


Figure 6.13-25 SPI Timing in Master Mode (Alternate Phase of SPIx_CLK)

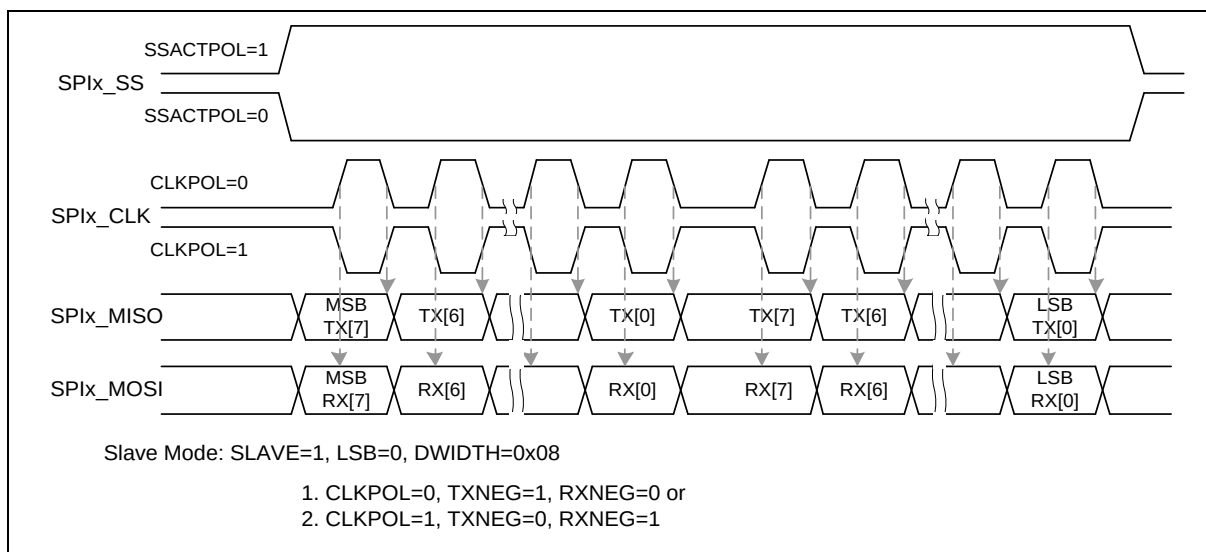


Figure 6.13-26 SPI Timing in Slave Mode

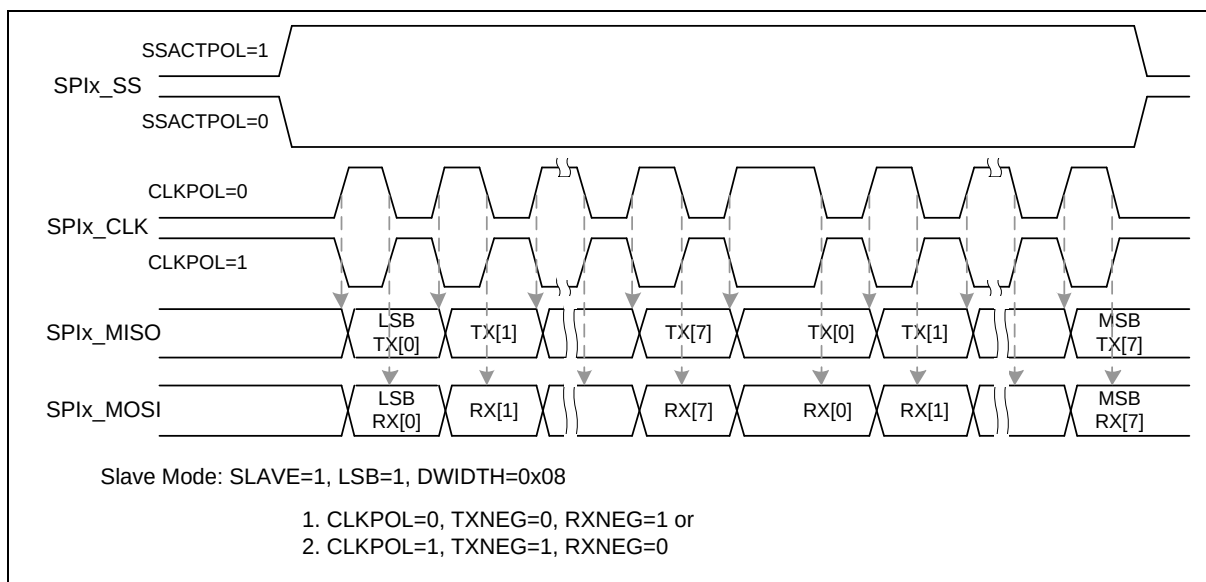


Figure 6.13-27 SPI Timing in Slave Mode (Alternate Phase of SPIx_CLK)

6.13.7 Programming Examples

Example 1:

The SPI controller is set as a full-duplex master to access an off-chip slave device with the following specifications:

- Data bit is latched on positive edge of SPI bus clock.
- Data bit is driven on negative edge of SPI bus clock.
- Data is transferred from MSB first.
- SPI bus clock is idle at low state.
- Only one byte of data to be transmitted/received in a transaction.
- Uses the first SPI slave select pin to connect with an off-chip slave device. The slave

selection signal is active low.

The operation flow is as follows:

1. Set DIVIDER (SPIx_CLKDIV [8:0]) to determine the output frequency of SPI clock.
2. Write the SPIx_SSCTL register a proper value for the related settings of Master mode:
 - 1) Clear AUTOSS (SPIx_SSCTL[3]) to 0 to disable the Automatic Slave Selection function.
 - 2) Configure slave selection signal as active low by clearing SSACTPOL (SPIx_SSCTL[2]) to 0.
 - 3) Enable slave selection signal by setting SS (SPIx_SSCTL[0]) to 1 to activate the off-chip slave device.
3. Write the related settings into the SPIx_CTL register to control the SPI master actions.
 - 1) Configure this SPI controller as master device by setting SLAVE (SPIx_CTL[18]) to 0.
 - 2) Force the SPI clock idle state at low by clearing CLKPOL (SPIx_CTL[3]) to 0.
 - 3) Select data transmitted on negative edge of SPI bus clock by setting TXNEG (SPIx_CTL[2]) to 1.
 - 4) Select data latched on positive edge of SPI bus clock by clearing RXNEG (SPIx_CTL[1]) to 0.
 - 5) Set the bit length of a transaction as 8-bit in DWIDTH bit field (SPIx_CTL[12:8] = 0x08).
 - 6) Set MSB transfer first by clearing LSB (SPIx_CTL[13]) to 0.
4. Set SPIEN (SPIx_CTL[0]) to 1 to enable the data transfer with the SPI interface.
5. If this SPI master attempts to transmit (write) one byte data to the off-chip slave device, write the byte data that will be transmitted into the SPIx_TX register.
6. Waiting for SPI interrupt if the UNITIEN (SPIx_CTL[17]) is set to 1, or just polling the unit transfer interrupt flag UNITIF (SPIx_STATUS[1]).
7. Read out the received one byte data from SPIx_RX register.
8. Go to 5) to continue another data transfer or set SS (SPIx_SSCTL[0]) to 0 to inactivate the off-chip slave device.

Example 2:

The SPI controller is set as a full-duplex slave device and connects with an off-chip master device. The off-chip master device communicates with the on-chip SPI slave controller through the SPI interface with the following specifications:

- Data bit is latched on positive edge of SPI bus clock.
- Data bit is driven on negative edge of SPI bus clock.
- Data is transferred from LSB first.
- SPI bus clock is idle at high state.
- Only one byte of data to be transmitted/received in a transaction.
- Slave selection signal is active high.

The operation flow is as follows:

1. Write the SPIx_SSCTL register a proper value for the related settings of Slave mode.
2. Select high level for the input of slave selection signal by setting SSACTPOL (SPIx_SSCTL[2]) to 1.
3. Write the related settings into the SPIx_CTL register to control this SPI slave actions

- 1) Set the SPI controller as slave device by setting SLAVE (SPIx_CTL[18]) to 1.
- 2) Select the SPI clock idle state at high by setting CLKPOL (SPIx_CTL[3]) to 1.
- 3) Select data transmitted on negative edge of SPI bus clock by setting TXNEG (SPIx_CTL[2]) to 1.
- 4) Select data latched on positive edge of SPI bus clock by clearing RXNEG (SPIx_CTL[1]) to 0.
- 5) Set the bit length of a transaction as 8-bit in DWIDTH bit field (SPIx_CTL[12:8] = 0x08).
4. Set LSB transfer first by setting LSB (SPIx_CTL[13]) to 1.
5. Set the SPIEN (SPIx_CTL[0]) to 1. Wait for the slave select trigger input and SPI clock input from the off-chip master device to start the data transfer.
6. If this SPI slave attempts to transmit (be read) one byte data to the off-chip master device, write the byte data that will be transmitted into the SPIx_TX register.
7. If this SPI slave just only attempts to receive (be written) one byte data from the off-chip master device and does not care what data will be transmitted, the SPIx_TX register does not need to be updated by software.
8. Waiting for SPI interrupt if the UNITIEN (SPIx_CTL[17]) is set to 1, or just polling the unit transfer interrupt flag UNITIF (SPIx_STATUS[1]).
9. Read out the received one byte data from SPIx_RX register.
10. Go to 7 to continue another data transfer or stop data transfer.

6.13.8 Register Map

R: read only, W: write only, R/W: both read and write

Register	Offset	R/W	Description	Reset Value
SPI Base Address: SPIx_BA = 0x4006_1000 x=0				
SPIx_CTL	SPIx_BA+0x00	R/W	SPI Control Register	0x0000_0034
SPIx_CLKDIV	SPIx_BA+0x04	R/W	SPI Clock Divider Register	0x0000_0000
SPIx_SSCTL	SPIx_BA+0x08	R/W	SPI Slave Select Control Register	0x0000_0000
SPIx_PDMACTL	SPIx_BA+0x0C	R/W	SPI PDMA Control Register	0x0000_0000
SPIx_FIFOCTL	SPIx_BA+0x10	R/W	SPI FIFO Control Register	0x2200_0000
SPIx_STATUS	SPIx_BA+0x14	R/W	SPI Status Register	0x0005_0110
SPIx_TX	SPIx_BA+0x20	W	SPI Data Transmit Register	0x0000_0000
SPIx_RX	SPIx_BA+0x30	R	SPI Data Receive Register	0x0000_0000
SPIx_I2SCTL	SPIx_BA+0x60	R/W	I ² S Control Register	0x0000_0000
SPIx_I2SCLK	SPIx_BA+0x64	R/W	I ² S Clock Divider Control Register	0x0000_0000
SPIx_I2SSTS	SPIx_BA+0x68	R/W	I ² S Status Register	0x0005_0100

6.13.9 Register Description

SPI Control Register (SPIx_CTL)

Register	Offset	R/W	Description	Reset Value
SPIx_CTL	SPIx_BA+0x00	R/W	SPI Control Register	0x0000_0034

Note: Not supported in I²S mode.

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved			DATDIR	REORDER	SLAVE	UNITIEN	Reserved
15	14	13	12	11	10	9	8
RXONLY	HALFDPX	LSB	DWIDTH				
7	6	5	4	3	2	1	0
SUSPITV				CLKPOL	TXNEG	RXNEG	SPIEN

Bits	Description
[31:21]	Reserved Reserved.
[20]	DATDIR Data Port Direction Control This bit is used to select the data input/output direction in half-duplex transfer and Dual/Quad transfer 0 = SPI data is input direction. 1 = SPI data is output direction.
[19]	REORDER Byte Reorder Function Enable Bit 0 = Byte Reorder function Disabled. 1 = Byte Reorder function Enabled. A byte suspend interval will be inserted among each byte. The period of the byte suspend interval depends on the setting of SUSPITV. Note: Byte Reorder function is only available if DWIDTH is defined as 16, 24, and 32 bits.
[18]	SLAVE Slave Mode Control 0 = Master mode. 1 = Slave mode.
[17]	UNITIEN Unit Transfer Interrupt Enable Bit 0 = SPI unit transfer interrupt Disabled. 1 = SPI unit transfer interrupt Enabled.
[16]	Reserved Reserved.
[15]	RXONLY Receive-only Mode Enable Bit (Master Only) This bit field is only available in Master mode. In receive-only mode, SPI Master will generate SPI bus clock continuously for receiving data bit from SPI slave device and assert the BUSY status. 0 = Receive-only mode Disabled. 1 = Receive-only mode Enabled.
[14]	HALFDPX SPI Half-duplex Transfer Enable Bit This bit is used to select full-duplex or half-duplex for SPI transfer. The bit field DATDIR

		(SPIx_CTL[20]) can be used to set the data direction in half-duplex transfer. 0 = SPI operates in full-duplex transfer. 1 = SPI operates in half-duplex transfer.
[13]	LSB	Send LSB First 0 = The MSB, which bit of transmit/receive register depends on the setting of DWIDTH, is transmitted/received first. 1 = The LSB, bit 0 of the SPI TX register, is sent first to the SPI data output pin, and the first bit received from the SPI data input pin will be put in the LSB position of the RX register (bit 0 of SPI_RX).
[12:8]	DWIDTH	Data Width This field specifies how many bits can be transmitted / received in one transaction. The minimum bit length is 8 bits and can up to 32 bits. DWIDTH = 0x08 8 bits. DWIDTH = 0x09 9 bits. DWIDTH = 0x1F 31 bits. DWIDTH = 0x00 32 bits. Note: This bit field will decide the depth of TX/RX FIFO configuration in SPI mode. Therefore, changing this bit field will clear TX/RX FIFO by hardware automatically.
[7:4]	SUSPITV	Suspend Interval (Master Only) The four bits provide configurable suspend interval between two successive transmit/receive transaction in a transfer. The definition of the suspend interval is the interval between the last clock edge of the preceding transaction word and the first clock edge of the following transaction word. The default value is 0x3. The period of the suspend interval is obtained according to the following equation. $(SUSPITV + 0.5) * \text{period of SPICLK clock cycle}$ Example: SUSPITV = 0x0 0.5 SPICLK clock cycle. SUSPITV = 0x1 1.5 SPICLK clock cycle. SUSPITV = 0xE 14.5 SPICLK clock cycle. SUSPITV = 0xF 15.5 SPICLK clock cycle.
[3]	CLKPOL	Clock Polarity 0 = SPI bus clock is idle low. 1 = SPI bus clock is idle high.
[2]	TXNEG	Transmit on Negative Edge 0 = Transmitted data output signal is changed on the rising edge of SPI bus clock. 1 = Transmitted data output signal is changed on the falling edge of SPI bus clock.
[1]	RXNEG	Receive on Negative Edge 0 = Received data input signal is latched on the rising edge of SPI bus clock. 1 = Received data input signal is latched on the falling edge of SPI bus clock.
[0]	SPIEN	SPI Transfer Control Enable Bit In Master mode, the transfer will start when there is data in the FIFO buffer after this bit is set to 1. In Slave mode, this device is ready to receive data when this bit is set to 1. 0 = Transfer control Disabled. 1 = Transfer control Enabled. Note: Before changing the configurations of SPIx_CTL, SPIx_CLKDIV, SPIx_SSCTL and SPIx_FIFCTL registers, user shall clear the SPIEN (SPIx_CTL[0]) and confirm the SPIENSTS (SPIx_STATUS[15]) is 0.

SPI Clock Divider Register (SPIx_CLKDIV)

Register	Offset	R/W	Description	Reset Value
SPIx_CLKDIV	SPIx_BA+0x04	R/W	SPI Clock Divider Register	0x0000_0000

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

Bits	Description	
[31:9]	Reserved	Reserved.
[8:0]	DIVIDER	Clock Divider The value in this field is the frequency divider for generating the peripheral clock, f_{spi_clk}, and the SPI bus clock of SPI Master. The frequency is obtained according to the following equation. $f_{spi\_clk} = \frac{f_{spi\_clock\_src}}{(DIVIDER + 1)}$ where $f_{spi\_clock\_src}$ is the peripheral clock source, which is defined in the clock control register, CLK_CLKSEL2. Note1: Not supported in I²S mode. Note2: The time interval must be larger than or equal 8 peripheral clock cycles between releasing SPI IP software reset and setting this clock divider register.

Note: DIVIDER should be set carefully because the peripheral clock frequency must be slower than or equal to system frequency.

SPI Slave Select Control Register (SPIx_SSCTL)

Register	Offset	R/W	Description	Reset Value
SPIx_SSCTL	SPIx_BA+0x08	R/W	SPI Slave Select Control Register	0x0000_0000

Note: Not supported in I²S mode.

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		SSINAIEN	SSACTIEN	Reserved		SLVURIEN	SLVBEIEN
7	6	5	4	3	2	1	0
Reserved				AUTOSS	SSACTPOL	Reserved	SS

Bits	Description
[31:14]	Reserved Reserved.
[13]	SSINAIEN Slave Select Inactive Interrupt Enable Bit 0 = Slave select inactive interrupt Disabled. 1 = Slave select inactive interrupt Enabled.
[12]	SSACTIEN Slave Select Active Interrupt Enable Bit 0 = Slave select active interrupt Disabled. 1 = Slave select active interrupt Enabled.
[11:10]	Reserved Reserved.
[9]	SLVURIEN Slave Mode TX Under Run Interrupt Enable Bit 0 = Slave mode TX under run interrupt Disabled. 1 = Slave mode TX under run interrupt Enabled.
[8]	SLVBEIEN Slave Mode Bit Count Error Interrupt Enable Bit 0 = Slave mode bit count error interrupt Disabled. 1 = Slave mode bit count error interrupt Enabled.
[7:4]	Reserved Reserved.
[3]	AUTOSS Automatic Slave Selection Function Enable Bit (Master Only) 0 = Automatic slave selection function Disabled. Slave selection signal will be asserted/de-asserted according to SS (SPIx_SSCTL[0]). 1 = Automatic slave selection function Enabled.
[2]	SSACTPOL Slave Selection Active Polarity This bit defines the active polarity of slave selection signal (SPIx_SS). 0 = The slave selection signal SPIx_SS is active low. 1 = The slave selection signal SPIx_SS is active high.
[1]	Reserved Reserved.

[0]	SS	Slave Selection Control (Master Only) If AUTOSS bit is cleared to 0, 0 = set the SPIx_SS line to inactive state. 1 = set the SPIx_SS line to active state. If the AUTOSS bit is set to 1, 0 = Keep the SPIx_SS line at inactive state. 1 = SPIx_SS line will be automatically driven to active state for the duration of data transfer, and will be driven to inactive state for the rest of the time. The active state of SPIx_SS is specified in SSACTPOL (SPIx_SSCTL[2]).

SPI PDMA Control Register (SPIx_PDMACTL)

Register	Offset	R/W	Description	Reset Value
SPIx_PDMACTL	SPIx_BA+0x0C	R/W	SPI PDMA Control Register	0x0000_0000

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

Bits	Description	
[31:3]	Reserved	Reserved.
[2]	PDMARST	PDMA Reset 0 = No effect. 1 = Reset the PDMA control logic of the SPI controller. This bit will be automatically cleared to 0.
[1]	RXPDMAEN	Receive PDMA Enable Bit 0 = Receive PDMA function Disabled. 1 = Receive PDMA function Enabled.
[0]	TXPDMAEN	Transmit PDMA Enable Bit 0 = Transmit PDMA function Disabled. 1 = Transmit PDMA function Enabled. Note: In SPI Master mode with full duplex transfer, if both TX and RX PDMA functions are enabled, RX PDMA function cannot be enabled prior to TX PDMA function. User can enable TX PDMA function firstly or enable both functions simultaneously.

SPI FIFO Control Register (SPIx_FIFCTL)

Register	Offset	R/W	Description	Reset Value
SPIx_FIFCTL	SPIx_BA+0x10	R/W	SPI FIFO Control Register	0x2200_0000

31	30	29	28	27	26	25	24
Reserved	TXTH			Reserved	RXTH		
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved						TXFBCLR	RXFBCLR
7	6	5	4	3	2	1	0
TXUFIEN	TXUFPOL	RXOVIEN	RXTOIEN	TXTHIEN	RXTHIEN	TXRST	RXRST

Bits	Description
[31]	Reserved
[30:28]	Transmit FIFO Threshold If the valid data count of the transmit FIFO buffer is less than or equal to the TXTH setting, the TXTHIF bit will be set to 1, else the TXTHIF bit will be cleared to 0. The MSB of this bit field is only meaningful while SPI mode 8~16 bits of data length.
[27]	Reserved
[26:24]	Receive FIFO Threshold If the valid data count of the receive FIFO buffer is larger than the RXTH setting, the RXTHIF bit will be set to 1, else the RXTHIF bit will be cleared to 0. The MSB of this bit field is only meaningful while SPI mode 8~16 bits of data length.
[23:10]	Reserved
[9]	Transmit FIFO Buffer Clear 0 = No effect. 1 = Clear transmit FIFO pointer. The TXFULL bit will be cleared to 0 and the TXEMPTY bit will be set to 1. This bit will be cleared to 0 by hardware about 1 system clock after it is set to 1. Note: The TX shift register will not be cleared.
[8]	Receive FIFO Buffer Clear 0 = No effect. 1 = Clear receive FIFO pointer. The RXFULL bit will be cleared to 0 and the RXEMPTY bit will be set to 1. This bit will be cleared to 0 by hardware about 1 system clock after it is set to 1. Note: The RX shift register will not be cleared.
[7]	TX Underflow Interrupt Enable Bit When TX underflow event occurs in Slave mode, TXUFIF (SPIx_STATUS[19]) will be set to 1. This bit is used to enable the TX underflow interrupt. 0 = Slave TX underflow interrupt Disabled. 1 = Slave TX underflow interrupt Enabled.

[6]	TXUFPOL	TX Underflow Data Polarity 0 = The SPI data out is keep 0 if there is TX underflow event in Slave mode. 1 = The SPI data out is keep 1 if there is TX underflow event in Slave mode. Note 1: The TX underflow event occurs if there is no any data in TX FIFO when the slave selection signal is active. Note 2: This bit should be set as 0 in I ² S mode. Note 3: When TX underflow event occurs, SPIx_MISO pin state will be determined by this setting even though TX FIFO is not empty afterward. Data stored in TX FIFO will be sent through SPIx_MISO pin in the next transfer frame.
[5]	RXOVIE	Receive FIFO Overrun Interrupt Enable Bit 0 = Receive FIFO overrun interrupt Disabled. 1 = Receive FIFO overrun interrupt Enabled.
[4]	RXTOIE	Slave Receive Time-out Interrupt Enable Bit 0 = Receive time-out interrupt Disabled. 1 = Receive time-out interrupt Enabled.
[3]	TXTHIE	Transmit FIFO Threshold Interrupt Enable Bit 0 = TX FIFO threshold interrupt Disabled. 1 = TX FIFO threshold interrupt Enabled.
[2]	RXTHIE	Receive FIFO Threshold Interrupt Enable Bit 0 = RX FIFO threshold interrupt Disabled. 1 = RX FIFO threshold interrupt Enabled.
[1]	TXRST	Transmit Reset 0 = No effect. 1 = Reset transmit FIFO pointer and transmit circuit. The TXFULL bit will be cleared to 0 and the TXEMPTY bit will be set to 1. This bit will be cleared to 0 by hardware about 3 system clock cycles + 2 peripheral clock cycles after it is set to 1. User can read TXRXRST (SPIx_STATUS[23]) to check if reset is accomplished or not. Note: If TX underflow event occurs in SPI Slave mode, this bit can be used to make SPI return to idle state.
[0]	RXRST	Receive Reset 0 = No effect. 1 = Reset receive FIFO pointer and receive circuit. The RXFULL bit will be cleared to 0 and the RXEMPTY bit will be set to 1. This bit will be cleared to 0 by hardware about 3 system clock cycles + 2 peripheral clock cycles after it is set to 1. User can read TXRXRST (SPIx_STATUS[23]) to check if reset is accomplished or not.

SPI Status Register (SPIx STATUS)

Register	Offset	R/W	Description	Reset Value
SPIx_STATUS	SPIx_BA+0x14	R/W	SPI Status Register	0x0005_0110

Note: Not supported in I²S mode.

31	30	29	28	27	26	25	24
TXCNT				RCXNT			
23	22	21	20	19	18	17	16
TXRXRST	Reserved			TXUFIF	TXTHIF	TXFULL	TXEMPTY
15	14	13	12	11	10	9	8
SPIENSTS	Reserved			RXTOIF	RXOVIF	RXTHIF	RXFULL
7	6	5	4	3	2	1	0
SLVURIF	SLVBEIF	Reserved	SSLINE	SSINAIF	SSACTIF	UNITIF	BUSY

Bits	Description
[31:28]	TXCNT Transmit FIFO Data Count (Read Only) This bit field indicates the valid data count of transmit FIFO buffer.
[27:24]	RCXNT Receive FIFO Data Count (Read Only) This bit field indicates the valid data count of receive FIFO buffer.
[23]	TXRXRST TX or RX Reset Status (Read Only) 0 = The reset function of TXRST or RXRST is done. 1 = Doing the reset function of TXRST or RXRST. Note: Both the reset operations of TXRST and RXRST need 3 system clock cycles + 2 peripheral clock cycles. User can check the status of this bit to monitor the reset function is doing or done.
[22:20]	Reserved Reserved.
[19]	TXUFIF TX Underflow Interrupt Flag When the TX underflow event occurs, this bit will be set to 1, the state of data output pin depends on the setting of TXUFPOL. 0 = No effect. 1 = No data in Transmit FIFO and TX shift register when the slave selection signal is active. Note 1: This bit will be cleared by writing 1 to it. Note 2: If reset slave's transmission circuit when slave selection signal is active, this flag will be set to 1 after 2 peripheral clock cycles + 3 system clock cycles since the reset operation is done.
[18]	TXTHIF Transmit FIFO Threshold Interrupt Flag (Read Only) 0 = The valid data count within the transmit FIFO buffer is larger than the setting value of TXTH. 1 = The valid data count within the transmit FIFO buffer is less than or equal to the setting value of TXTH.
[17]	TXFULL Transmit FIFO Buffer Full Indicator (Read Only) 0 = Transmit FIFO buffer is not full. 1 = Transmit FIFO buffer is full.

[16]	TXEMPTY	Transmit FIFO Buffer Empty Indicator (Read Only) 0 = Transmit FIFO buffer is not empty. 1 = Transmit FIFO buffer is empty.
[15]	SPIENSTS	SPI Enable Status (Read Only) 0 = SPI controller Disabled. 1 = SPI controller Enabled. Note: The SPI peripheral clock is asynchronous with the system clock. In order to make sure the SPI control logic is disabled, this bit indicates the real status of SPI controller.
[14:13]	Reserved	Reserved.
[12]	RXTOIF	Receive Time-out Interrupt Flag 0 = No receive FIFO time-out event. 1 = Receive FIFO buffer is not empty and no read operation on receive FIFO buffer over 64 SPI peripheral clock periods in Master mode or over 576 SPI peripheral clock periods in Slave mode. When the received FIFO buffer is read by software, the time-out status will be cleared automatically. Note: This bit will be cleared by writing 1 to it.
[11]	RXOVIF	Receive FIFO Overrun Interrupt Flag When the receive FIFO buffer is full, the follow-up data will be dropped and this bit will be set to 1. 0 = No FIFO is overrun. 1 = Receive FIFO is overrun. Note: This bit will be cleared by writing 1 to it.
[10]	RXTHIF	Receive FIFO Threshold Interrupt Flag (Read Only) 0 = The valid data count within the receive FIFO buffer is smaller than or equal to the setting value of RXTH. 1 = The valid data count within the receive FIFO buffer is larger than the setting value of RXTH.
[9]	RXFULL	Receive FIFO Buffer Full Indicator (Read Only) 0 = Receive FIFO buffer is not full. 1 = Receive FIFO buffer is full.
[8]	RXEMPTY	Receive FIFO Buffer Empty Indicator (Read Only) 0 = Receive FIFO buffer is not empty. 1 = Receive FIFO buffer is empty.
[7]	SLVURIF	Slave Mode TX Under Run Interrupt Flag In Slave mode, if TX underflow event occurs and the slave select line goes to inactive state, this interrupt flag will be set to 1. 0 = No Slave TX under run event. 1 = Slave TX under run event occurred. Note: This bit will be cleared by writing 1 to it.
[6]	SLVBEIF	Slave Mode Bit Count Error Interrupt Flag In Slave mode, when the slave select line goes to inactive state, if bit counter is mismatched with DWIDTH, this interrupt flag will be set to 1. 0 = No Slave mode bit count error event. 1 = Slave mode bit count error event occurred. Note: If the slave select active but there is no any bus clock input, the SLVBEIF also active when the slave select goes to inactive state. This bit will be cleared by writing 1 to it.
[5]	Reserved	Reserved.
[4]	SSLINE	Slave Select Line Bus Status (Read Only)

		0 = The slave select line status is 0. 1 = The slave select line status is 1. Note: This bit is only available in Slave mode. If SSACTPOL (SPIx_SSCTL[2]) is set 0, and the SSLINE is 1, the SPI slave select is in inactive status.
[3]	SSINAIF	Slave Select Inactive Interrupt Flag 0 = Slave select inactive interrupt was cleared or not occurred. 1 = Slave select inactive interrupt event occurred. Note: Only available in Slave mode. This bit will be cleared by writing 1 to it.
[2]	SSACTIF	Slave Select Active Interrupt Flag 0 = Slave select active interrupt was cleared or not occurred. 1 = Slave select active interrupt event occurred. Note: Only available in Slave mode. This bit will be cleared by writing 1 to it.
[1]	UNITIF	Unit Transfer Interrupt Flag 0 = No transaction has been finished since this bit was cleared to 0. 1 = SPI controller has finished one unit transfer. Note: This bit will be cleared by writing 1 to it.
[0]	BUSY	Busy Status (Read Only) 0 = SPI controller is in idle state. 1 = SPI controller is in busy state. The following lists the bus busy conditions: - SPIEN (SPIx_CTL[0]) = 1 and TXEMPTY = 0. - For SPI Master mode, SPIEN (SPIx_CTL[0]) = 1 and TXEMPTY = 1 but the current transaction is not finished yet. - For SPI Master mode, SPIEN (SPIx_CTL[0]) = 1 and RXONLY = 1. - For SPI Slave mode, SPIEN (SPIx_CTL[0]) = 1 and there is serial clock input into the SPI core logic when slave select is active. - For SPI Slave mode, SPIEN (SPIx_CTL[0]) = 1 and the transmit buffer or transmit shift register is not empty even if the slave select is inactive.

SPI Data Transmit Register (SPIx_TX)

Register	Offset	R/W	Description	Reset Value
SPIx_TX	SPIx_BA+0x20	W	SPI Data Transmit Register	0x0000_0000

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

Bits	Description
[31:0]	Data Transmit Register The data transmit registers pass through the transmitted data into the 4-level transmit FIFO buffers. The number of valid bits depends on the setting of DWIDTH (SPIx_CTL[12:8]) in SPI mode or WDWIDTH (SPIx_I2SCTL[5:4]) in I²S mode. In SPI mode, if DWIDTH is set to 0x08, the bits TX[7:0] will be transmitted. If DWIDTH is set to 0x00, the SPI controller will perform a 32-bit transfer. In I²S mode, if WDWIDTH (SPIx_I2SCTL[5:4]) is set to 0x2, the data width of audio channel is 24-bit and corresponding to TX[23:0]. If WDWIDTH is set as 0x0, 0x1, or 0x3, all bits of this field are valid and referred to the data arrangement in I²S mode FIFO operation section. Note: In Master mode, SPI controller will start to transfer the SPI bus clock after 1 APB clock and 6 peripheral clock cycles after user writes to this register.

SPI Data Receive Register (SPIx_RX)

Register	Offset	R/W	Description	Reset Value
SPIx_RX	SPIx_BA+0x30	R	SPI Data Receive Register	0x0000_0000

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

Bits	Description
[31:0]	Data Receive Register (Read Only) There are 4-level FIFO buffers in this controller. The data receive register holds the data received from SPI data input pin. If the RXEMPTY (SPIx_STATUS[8] or SPIx_I2SSTS[8]) is not set to 1, the receive FIFO buffers can be accessed through software by reading this register.

I²S Control Register (SPIx_I2SCTL)

Register	Offset	R/W	Description	Reset Value
SPIx_I2SCTL	SPIx_BA+0x60	R/W	I ² S Control Register	0x0000_0000

Note: Not supported in SPI mode.

31	30	29	28	27	26	25	24
SLVERRIEN	Reserved	FORMAT		Reserved		LZCIEN	RZCIEN
23	22	21	20	19	18	17	16
RXLCH	Reserved					LZCEN	RZCEN
15	14	13	12	11	10	9	8
MCLKEN	Reserved						SLAVE
7	6	5	4	3	2	1	0
ORDER	MONO	WDWIDTH		MUTE	RXEN	TXEN	I2SEN

Bits	Description	
[31]	SLVERRIEN	Bit Number Error Interrupt Enable Bit for Slave Mode Interrupt occurs if this bit is set to 1 and bit number error event occurs. 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[30]	Reserved	Reserved.
[29:28]	FORMAT	Data Format Selection 00 = I ² S data format. 01 = MSB justified data format. 10 = PCM mode A. 11 = PCM mode B.
[27:26]	Reserved	Reserved.
[25]	LZCIEN	Left Channel Zero Cross Interrupt Enable Bit Interrupt occurs if this bit is set to 1 and left channel zero cross event occurs. 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[24]	RZCIEN	Right Channel Zero Cross Interrupt Enable Bit Interrupt occurs if this bit is set to 1 and right channel zero cross event occurs. 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[23]	RXLCH	Receive Left Channel Enable Bit When monaural format is selected (MONO = 1), I ² S controller will receive right channel data if RXLCH is set to 0, and receive left channel data if RXLCH is set to 1. 0 = Receive right channel data in Mono mode. 1 = Receive left channel data in Mono mode.
[22:18]	Reserved	Reserved.

[17]	LZCEN	Left Channel Zero Cross Detection Enable Bit If this bit is set to 1, when left channel data sign bit changes or next shift data bits are all 0 then LZCIF flag in SPIx_I2SSTS register is set to 1. This function is only available in transmit operation. 0 = Left channel zero cross detection Disabled. 1 = Left channel zero cross detection Enabled.
[16]	RZCEN	Right Channel Zero Cross Detection Enable Bit If this bit is set to 1, when right channel data sign bit change or next shift data bits are all 0 then RZCIF flag in SPIx_I2SSTS register is set to 1. This function is only available in transmit operation. 0 = Right channel zero cross detection Disabled. 1 = Right channel zero cross detection Enabled.
[15]	MCLKEN	Master Clock Enable Bit If MCLKEN is set to 1, I ² S controller will generate master clock on SPIx_I2SMCLK pin for external audio devices. 0 = Master clock Disabled. 1 = Master clock Enabled.
[14:9]	Reserved	Reserved.
[8]	SLAVE	Slave Mode I ² S can operate as master or slave. For Master mode, I2Sx_BCLK and I2Sx_LRCLK pins are output mode and send bit clock from this chip to audio CODEC chip. In Slave mode, I2Sx_BCLK and I2Sx_LRCLK pins are input mode and I2Sx_BCLK and I2Sx_LRCLK signals are received from outer audio CODEC chip. 0 = Master mode. 1 = Slave mode.
[7]	ORDER	Stereo Data Order in FIFO 0 = Left channel data at high byte. 1 = Left channel data at low byte.
[6]	MONO	Monaural Data 0 = Data is stereo format. 1 = Data is monaural format.
[5:4]	WDWIDTH	Word Width 00 = data size is 8-bit. 01 = data size is 16-bit. 10 = data size is 24-bit. 11 = data size is 32-bit.
[3]	MUTE	Transmit Mute Enable Bit 0 = Transmit data is shifted from buffer. 1 = Transmit channel zero.
[2]	RXEN	Receive Enable Bit 0 = Data receive Disabled. 1 = Data receive Enabled.
[1]	TXEN	Transmit Enable Bit 0 = Data transmit Disabled. 1 = Data transmit Enabled.

[0]	I2SEN	I²S Controller Enable Bit 0 = I²S mode Disabled. 1 = I²S mode Enabled. Note 1: If enabling this bit, I2Sx_BCLK will start to output in Master mode. Note 2: Before changing the configurations of SPIx_I2SCTL, SPIx_I2SCLK, and SPIx_FIFOCTL registers, user shall clear the I2SEN (SPIx_I2SCTL[0]) and confirm the I2SENSTS (SPIx_I2SSTS[15]) is 0.

I²S Clock Divider Control Register (SPIx_I2SCLK)

Register	Offset	R/W	Description	Reset Value
SPIx_I2SCLK	SPIx_BA+0x64	R/W	I ² S Clock Divider Control Register	0x0000_0000

Note: Not supported in SPI mode.

31	30	29	28	27	26	25	24
Reserved						I2SSLAVE	I2SMODE
23	22	21	20	19	18	17	16
Reserved						BCLKDIV	
15	14	13	12	11	10	9	8
BCLKDIV							
7	6	5	4	3	2	1	0
Reserved	MCLKDIV						

Bits	Description
[31:26]	Reserved Reserved.
[25]	I2SSLAVE I²S Clock Divider Number Selection for I²S Slave Mode and I²S Master Mode User sets I2SSLAVE to set frequency of peripheral clock of I ² S master mode and I ² S slave mode when BCLKDIV (SPIx_I2SCLK[17:8]) is set. I2SSLAVE needs to set before I2SEN (SPIx_I2SCTL[0]) is enabled. 0 = The frequency of peripheral clock is set to I ² S master mode. 1 = The frequency of peripheral clock is set to I ² S slave mode.
[24]	I2SMODE I²S Clock Divider Number Selection for I²S Mode and SPI Mode User sets I2SMODE to set frequency of peripheral clock of I ² S mode or SPI mode when BCLKDIV (SPIx_I2SCLK[17:8]) or DIVIDER (SPIx_CLKDIV[8:0]) is set. User needs to set I2SMODE before I2SEN (SPIx_I2SCTL[0]) or SPIEN (SPIx_CTL[0]) is enabled. 0 = The frequency of peripheral clock is set to SPI mode. 1 = The frequency of peripheral clock is set to I ² S mode.
[23:18]	Reserved Reserved.

[17:8]	BCLKDIV	Bit Clock Divider The I²S controller will generate bit clock in Master mode. The clock frequency of bit clock, f_{BCLK}, is determined by the following expression: $f_{BCLK} = \frac{f_{i2s\_clock\_src}}{2 \times (BCLKDIV + 1)}$ where $f_{i2s\_clock\_src}$ is the frequency of I²S peripheral clock source, which is defined in the clock control register CLK_CLKSEL2. In I²S Slave mode, this field is used to define the frequency of peripheral clock and it's determined by $f_{i2s\_clock\_src} \div \left(\frac{BCLKDIV}{2} + 1 \right)$. The peripheral clock frequency in I²S Slave mode must be equal to or faster than 6 times of input bit clock. Note: The time interval must be larger than or equal 8 peripheral clock cycles between releasing SPI IP software reset and setting this clock divider register.
[7]	Reserved	Reserved.
[6:0]	MCLKDIV	Master Clock Divider If MCLKEN is set to 1, I²S controller will generate master clock for external audio devices. The frequency of master clock, f_{MCLK}, is determined by the following expressions: If MCLKDIV >= 1, $f_{MCLK} = \frac{f_{i2s\_clock\_src}}{2 \times MCLKDIV}$ If MCLKDIV = 0, $f_{MCLK} = f_{i2s\_clock\_src}$ where $f_{i2s\_clock\_src}$ is the frequency of I²S peripheral clock source, which is defined in the clock control register CLK_CLKSEL2. In general, the master clock rate is 256 times sampling clock rate.

Note: BCLKDIV should be set carefully because the peripheral clock frequency must be slower than or equal to system frequency.

I²S Status Register (SPIx_I2SSTS)

Register	Offset	R/W	Description	Reset Value
SPIx_I2SSTS	SPIx_BA+0x68	R/W	I ² S Status Register	0x0005_0100

Note: Not supported in SPI mode.

31	30	29	28	27	26	25	24
Reserved	TXCNT			Reserved	RXCNT		
23	22	21	20	19	18	17	16
TXRXRST	SLVERRIF	LZCIF	RZCIF	TXUFIF	TXTHIF	TXFULL	TXEMPTY
15	14	13	12	11	10	9	8
I2SENSTS	Reserved			RXTOIF	RXOVIF	RXTHIF	RXFULL
7	6	5	4	3	2	1	0
Reserved			RIGHT	Reserved			

Bits	Description
[31]	Reserved
[30:28]	TXCNT Transmit FIFO Data Count (Read Only) This bit field indicates the valid data count of transmit FIFO buffer.
[27]	Reserved
[26:24]	RXCNT Receive FIFO Data Count (Read Only) This bit field indicates the valid data count of receive FIFO buffer.
[23]	TXRXRST TX or RX Reset Status (Read Only) 0 = The reset function of TXRST or RXRST is done. 1 = Doing the reset function of TXRST or RXRST. Note: Both the reset operations of TXRST and RXRST need 3 system clock cycles + 2 peripheral clock cycles. User can check the status of this bit to monitor the reset function is doing or done.
[22]	SLVERRIF Bit Number Error Interrupt Flag for Slave Mode 0 = No bit number error event occurred. 1 = Bit number error event occurred. Note: This bit will be cleared by writing 1 to it.
[21]	LZCIF Left Channel Zero Cross Interrupt Flag 0 = No zero cross event occurred on left channel. 1 = Zero cross event occurred on left channel.
[20]	RZCIF Right Channel Zero Cross Interrupt Flag 0 = No zero cross event occurred on right channel. 1 = Zero cross event occurred on right channel.
[19]	TXUFIF Transmit FIFO Underflow Interrupt Flag When the transmit FIFO buffer is empty and there is no datum written into the FIFO buffer, if there is more bus clock input, this bit will be set to 1. Note: This bit will be cleared by writing 1 to it.

[18]	TXTHIF	Transmit FIFO Threshold Interrupt Flag (Read Only) 0 = The valid data count within the transmit FIFO buffer is larger than the setting value of TXTH. 1 = The valid data count within the transmit FIFO buffer is less than or equal to the setting value of TXTH. Note: If TXTHIEN = 1 and TXTHIF = 1, the SPI/I ² S controller will generate a SPI interrupt request.
[17]	TXFULL	Transmit FIFO Buffer Full Indicator (Read Only) 0 = Transmit FIFO buffer is not full. 1 = Transmit FIFO buffer is full.
[16]	TXEMPTY	Transmit FIFO Buffer Empty Indicator (Read Only) 0 = Transmit FIFO buffer is not empty. 1 = Transmit FIFO buffer is empty.
[15]	I2SENSTS	I²S Enable Status (Read Only) 0 = The SPI/I ² S control logic is disabled. 1 = The SPI/I ² S control logic is enabled. Note: The SPI peripheral clock is asynchronous with the system clock. In order to make sure the SPI/I ² S control logic is disabled, this bit indicates the real status of SPI/I ² S control logic for user.
[14:13]	Reserved	Reserved.
[12]	RXTOIF	Receive Time-out Interrupt Flag 0 = No receive FIFO time-out event. 1 = Receive FIFO buffer is not empty and no read operation on receive FIFO buffer over 64 SPI peripheral clock period in Master mode or over 576 SPI peripheral clock period in Slave mode. When the received FIFO buffer is read by software, the time-out status will be cleared automatically. Note: This bit will be cleared by writing 1 to it.
[11]	RXOVIF	Receive FIFO Overrun Interrupt Flag When the receive FIFO buffer is full, the follow-up data will be dropped and this bit will be set to 1. Note: This bit will be cleared by writing 1 to it.
[10]	RXTHIF	Receive FIFO Threshold Interrupt Flag (Read Only) 0 = The valid data count within the receive FIFO buffer is smaller than or equal to the setting value of RXTH. 1 = The valid data count within the receive FIFO buffer is larger than the setting value of RXTH. Note: If RXTHIEN = 1 and RXTHIF = 1, the SPI/I ² S controller will generate a SPI interrupt request.
[9]	RXFULL	Receive FIFO Buffer Full Indicator (Read Only) 0 = Receive FIFO buffer is not full. 1 = Receive FIFO buffer is full.
[8]	RXEMPTY	Receive FIFO Buffer Empty Indicator (Read Only) 0 = Receive FIFO buffer is not empty. 1 = Receive FIFO buffer is empty.
[7:5]	Reserved	Reserved.
[4]	RIGHT	Right Channel (Read Only) This bit indicates the current transmit data is belong to which channel. 0 = Left channel.

		1 = Right channel.
[3:0]	Reserved	Reserved.

6.14 I²C Serial Interface Controller (I²C)

6.14.1 Overview

I²C is a two-wire, bi-directional serial bus that provides a simple and efficient method of data exchange between devices. The I²C standard is a true multi-master bus including collision detection and arbitration that prevents data corruption if two or more masters attempt to control the bus simultaneously.

There are two sets of I²C controllers that support Power-down wake-up function.

6.14.2 Features

The I²C bus uses two wires (SDA and SCL) to transfer information between devices connected to the bus. The main features of the I²C bus include:

- Supports up to two I²C ports
- Master/Slave mode
- Bidirectional data transfer between masters and slaves
- Multi-master bus (no central master)
- Supports Standard mode (100 kbps), Fast mode (400 kbps) and Fast mode plus (1 Mbps)
- Arbitration between simultaneously transmitting masters without corruption of serial data on the bus
- Serial clock synchronization allows devices with different bit rates to communicate via one serial bus
- Serial clock synchronization used as a handshake mechanism to suspend and resume serial transfer
- Built-in 14-bit time-out counter requesting the I²C interrupt if the I²C bus hangs up and timer-out counter overflow
- Programmable clocks allow for versatile rate control
- Supports 7-bit addressing and 10-bit addressing mode
- Supports multiple address recognition (four slave address with mask option)
- Supports Power-down wake-up function
- Supports PDMA with one buffer capability
- Supports setup/hold time programmable

6.14.3 Block Diagram

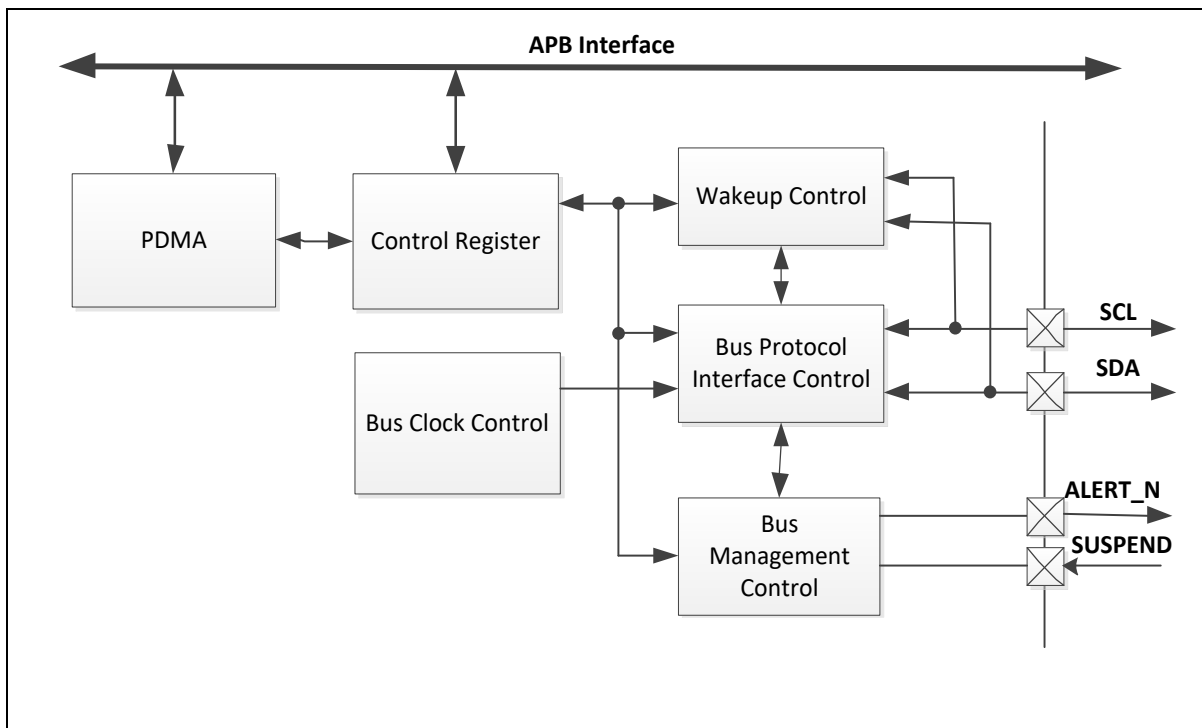


Figure 6.14-1 I²C Controller Block Diagram

6.14.4 Basic Configuration

6.14.4.1 I²C0 Basic Configurations

- Clock Source Configuration
 - Enable I²C0 peripheral clock in I2C0CKEN (CLK_APBCLK0[8]).
- Reset Configuration
 - Reset I²C0 controller in I2C0RST (SYS_IPRST1[8]).
- Pin Configuration I²C1 Basic Configurations
- Clock Source Configuration
 - Enable I²C1 peripheral clock in I2C1CKEN (CLK_APBCLK0[9]).
- Reset Configuration
 - Reset I²C1 controller in I2C1RST (SYS_IPRST1[9]).
- Pin Configuration Functional Description

On I²C bus, data is transferred between a Master and a Slave. Data bits transfer on the SCL and SDA lines are synchronously on a byte-by-byte basis. Each data byte is 8-bit long. There is one SCL clock pulse for each data bit with the MSB being transmitted first, and an acknowledge bit follows each transferred byte. Each bit is sampled during the high period of SCL; therefore, the SDA line may be changed only during the low period of SCL and must be held stable during the high period of SCL. A transition on the SDA line while SCL is high is interpreted as a command (START or STOP). Please refer to Figure 6.14-2 for more detailed I²C Bus Timing.

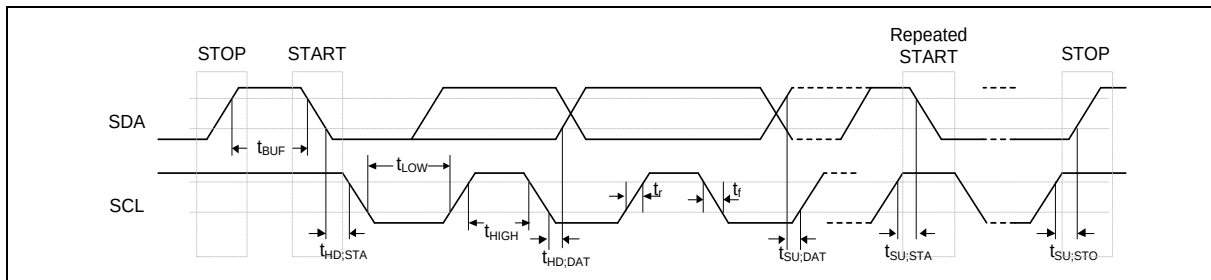


Figure 6.14-2 I²C Bus Timing

The on-chip I²C provides the serial interface that meets the I²C bus standard mode specification. The I²C port handles byte transfers autonomously. To enable this port, the bit I2CEN in I2C_CTL0 should be set to '1'. The I²C hardware interfaces to the I²C bus via two pins: SDA and SCL. When I/O pins are used as I²C ports, user must set the pins function to I²C in advance.

Note: -up resistor is needed for I²C operation as the SDA and SCL are open-drain pins.

6.14.4.2 I²C Protocol

Figure 6.14-3 shows the typical I²C protocol. Normally, a standard communication consists of four parts:

- START or Repeated START signal generation
- Slave address and R/W bit transfer
- Data transfer
- STOP signal generation

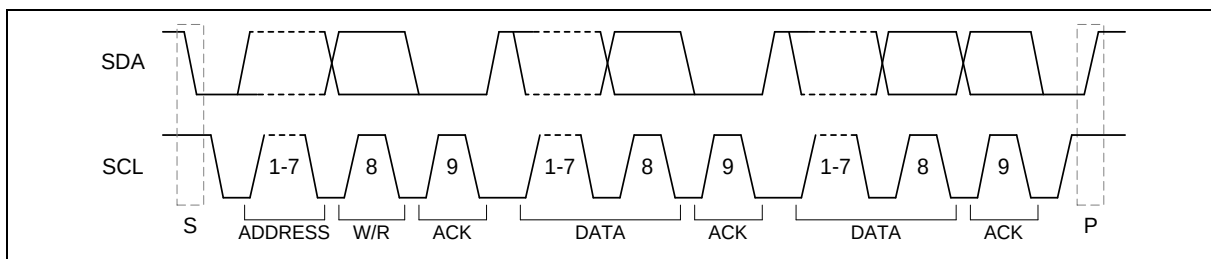


Figure 6.14-3 I²C Protocol

- START or Repeated START signal

When the bus is free/idle, which means no master device is engaging the bus (both SCL and SDA lines are high), a master can initiate a transfer by sending a START signal. A START signal, usually referred to as the "S" bit, is defined as a HIGH to LOW transition on the SDA line while SCL is HIGH. The START signal denotes the beginning of a new data transmission.

After having sent the address byte (address and read/write bit), the master may send any number of bytes followed by a stop condition. Instead of sending the stop condition it is also allowed to send another start condition again followed by an address (and of course including a read/write bit) and more data. The start condition is called as Repeat START (Sr). This is defined recursively allowing any number of start conditions to be sent. The purpose of this is to allow combined write/read operations to one or more devices without releasing the bus and thus with the guarantee that the operation is not interrupted. The controller uses this method to communicate with another slave or the same slave in a different transfer direction (e.g. from writing to a device to reading from a device) without releasing the bus.

- STOP signal

The master can terminate the communication by generating a STOP signal. A STOP signal, usually referred to as the “P” bit, is defined as a LOW to HIGH transition on the SDA line while SCL is HIGH.

Figure 6.14-4 shows the waveform of START, Repeat START and STOP.

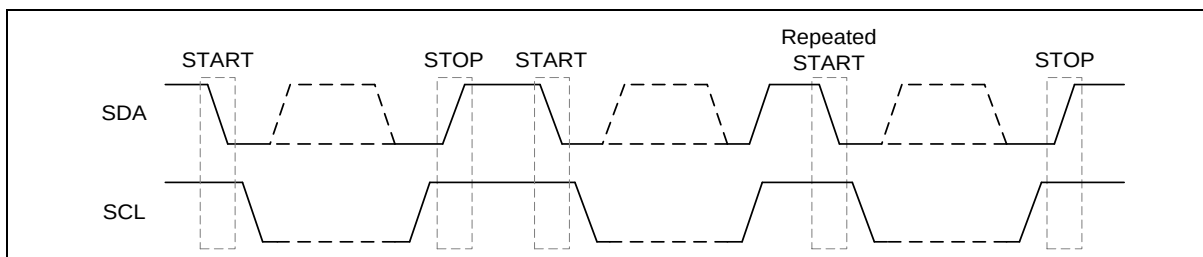


Figure 6.14-4 START and STOP Conditions

- Slave Address Transfer

After a (Repeated) START condition, the master sends a slave address to identify the target device of the communication. The start address can comprise one or two address bytes (for 7-bit or for 10-bit addressing schemes). After an address byte, a slave sensitive to the transmitted address has to acknowledge the reception.

Therefore, the slave’s address can be programmed in the device, where it is compared to the received address. In case of a match, the slave answers with an acknowledge (SDA = 0). Slaves that are not targeted answer with a non-acknowledge (SDA = 1). In addition to the match of the programmed address, another address byte value has to be answered with an acknowledge if the slave is capable to handle the corresponding requests.

- Data Transfer

When a slave receives a correct address with an R/W bit, the data will follow R/W bit specified to transfer. Each transferred byte is followed by an acknowledge bit on the 9th SCL clock cycle. If the slave signals a Not Acknowledge (NACK), the master can generate a STOP signal to abort the data transfer or generate a Repeated START signal and start a new transfer cycle.

If the master, as a receiving device, does Not Acknowledge (NACK) the slave, the slave releases the SDA line for the master to generate a STOP or Repeated START signal. Figure 6.14-5 and Figure 6.14-6 shows the waveform of bit transfer and acknowledge.

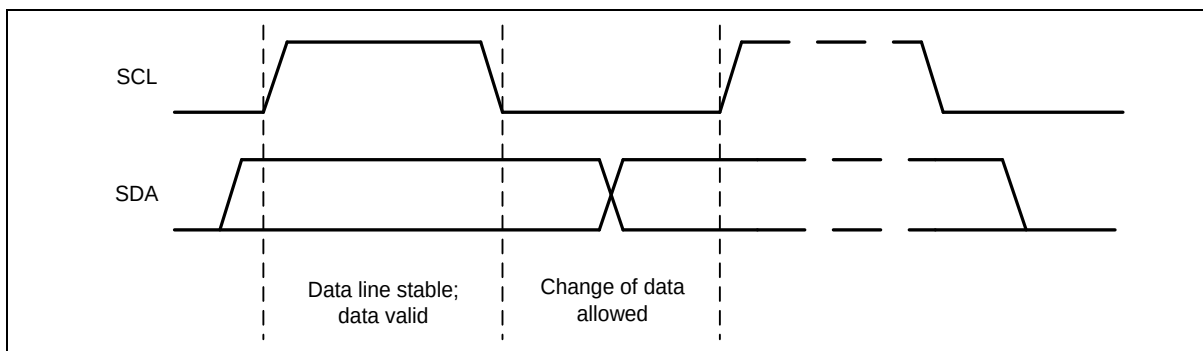


Figure 6.14-5 Bit Transfer on the I²C Bus

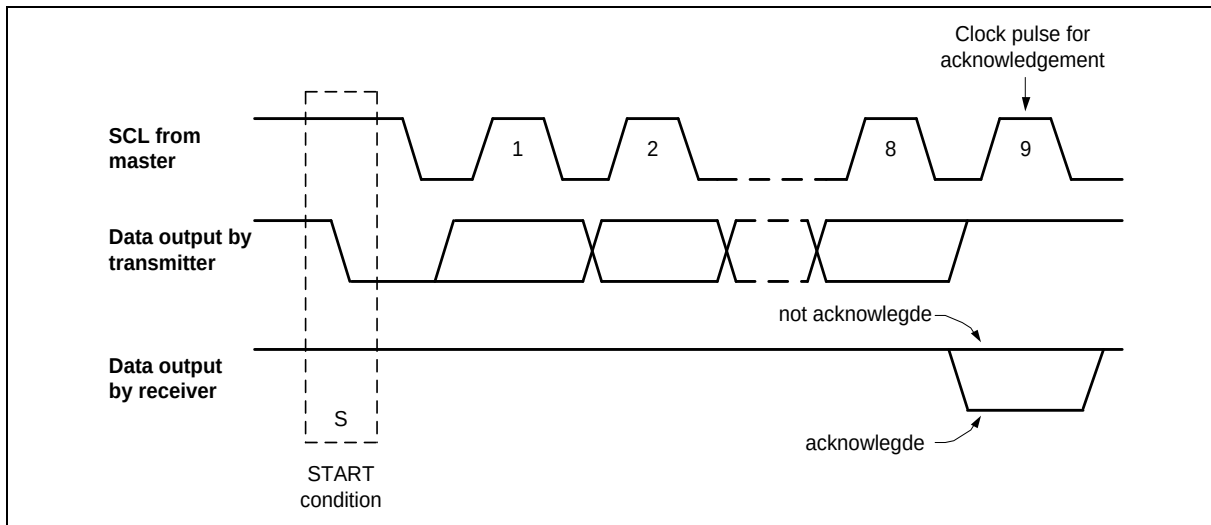


Figure 6.14-6 Acknowledge on the I²C Bus

- Data transfer on I²C bus

Figure 6.14-7 shows a master transmits data to slave by 7-bit. A master addresses a slave with a 7-bit address and 1-bit write index to denote that the master wants to transmit data to the slave. The master keeps transmitting data after the slave returns acknowledge to the master.

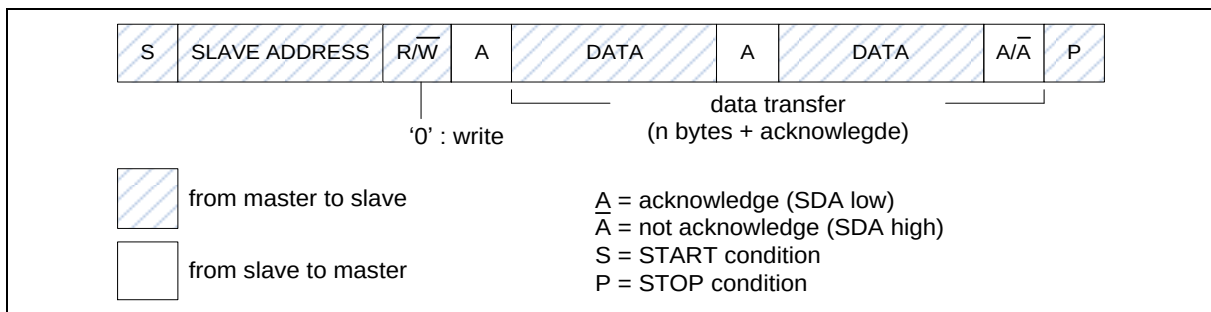


Figure 6.14-7 Master Transmits Data to Slave by 7-bit

Figure 6.14-8 shows a master read data from slave by 7-bit. A master addresses a slave with a 7-bit address and 1-bit read index to denote that the master wants to read data from the slave. The slave will start transmitting data after the slave returns acknowledge to the master.

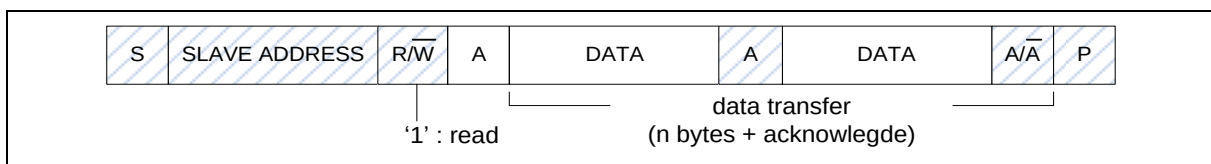


Figure 6.14-8 Master Reads Data from Slave by 7-bit

Figure 6.14-9 shows a master transmits data to slave by 10-bit. A master addresses a slave with a 10-bit address. First byte contains 10-bit address indicator (5'b11110) and 2-bit address with write index, second byte contains 8-bit address. The master keeps transmitting data after the second byte end. Note that 7-bit and 10-bit address device can work on the same bus.

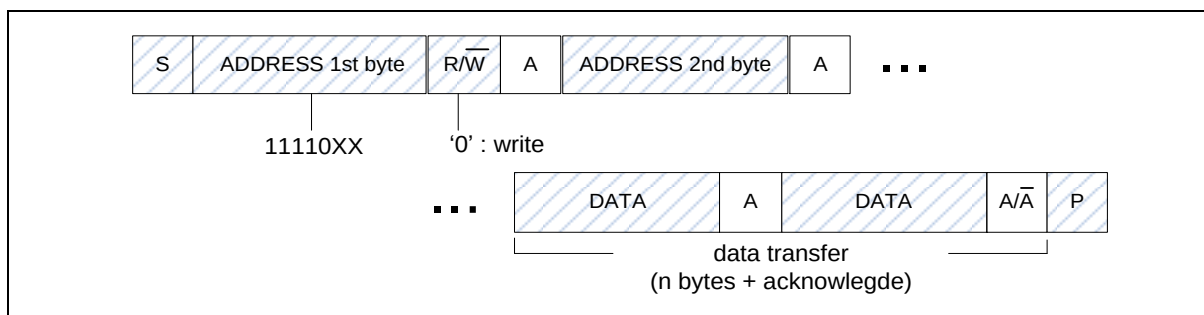


Figure 6.14-9 Master Transmits Data to Slave by 10-bit

Figure 6.14-10 shows a master read data from slave by 10-bit. A master addresses a slave with a 10-bit address. First master transmits 10-bit address to slave, after that master transmits first byte with read index. The slave will start transmitting data after the first byte with read index.

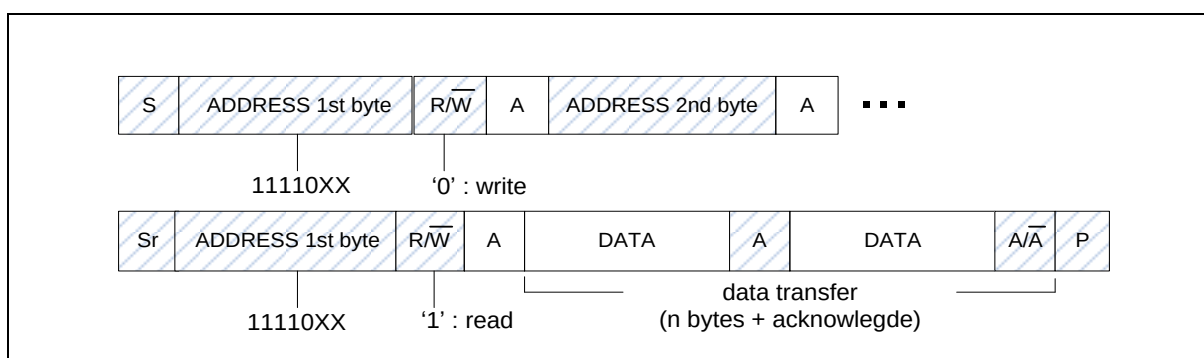


Figure 6.14-10 Master Reads Data from Slave by 10-bit

6.14.4.3 Operation Modes

The on-chip I²C ports support three operation modes, Master, Slave, and General Call Mode.

In a given application, I²C port may operate as a master or as a slave. In Slave mode, the I²C port hardware looks for its own slave address and the general call address. If one of these addresses is detected, and if the slave is willing to receive or transmit data from/to master (by setting the AA bit), acknowledge pulse will be transmitted out on the 9th clock, hence an interrupt is requested on both master and slave devices if interrupt is enabled. When the microcontroller wishes to become the bus master, hardware waits until the bus is free before entering Master mode so that a possible slave action is not be interrupted. If bus arbitration is lost in Master mode, I²C port switches to Slave mode immediately and can detect its own slave address in the same serial transfer.

To control the I²C bus transfer in each mode, user needs to set I2C_CTL0, I2C_DAT registers according to current status code of I2C_STATUS0 register. In other words, for each I²C bus action, user needs to check current status by I2C_STATUS0 register, and then set I2C_CTL0, I2C_DAT registers to take bus action. Finally, check the response status by I2C_STATUS0.

The bits, STA, STO and AA in I2C_CTL0 register are used to control the next state of the I²C hardware after SI flag of I2C_CTL0 [3] register is cleared. Upon completion of the new action, a new status code will be updated in I2C_STATUS0 register and the SI flag of I2C_CTL0 register will be set. But the SI flag will not be set when I²C STOP. If the I²C interrupt control bit INTEN (I2C_CTL0 [7]) is set, appropriate action or software branch of the new status code can be performed in the Interrupt service routine.

Figure 6.14-11 shows the current I²C status code is 0x08, and then set I2C_DATA=SLA+W and (STA,STO,SI,AA) = (0,0,1,x) to send the address to I²C bus. If a slave on the bus matches the address and response ACK, the I2C_STATUS0 will be updated by status code 0x18.

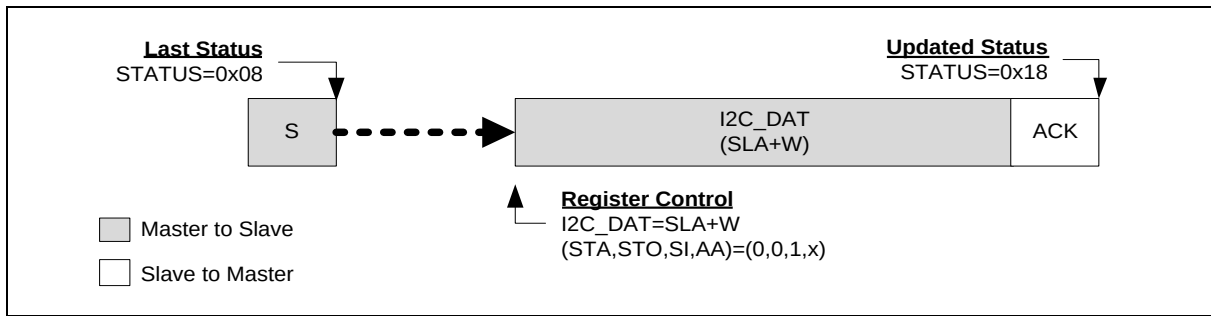


Figure 6.14-11 Control I2C Bus according to the Current I2C Status

Master Mode

In Figure 6.14-12 and Figure 6.14-13, all possible protocols for I2C master are shown. User needs to follow proper path of the flow to implement required I2C protocol.

In other words, user can send a START signal to bus and I2C will be in Master Transmitter (MT) mode (Figure 6.14-12) or Master receiver (MR) mode (Figure 6.14-13) after START signal has been sent successfully and new status code would be 0x08. Followed by START signal, user can send slave address, read/write bit, data and Repeat START, STOP to perform I2C protocol.

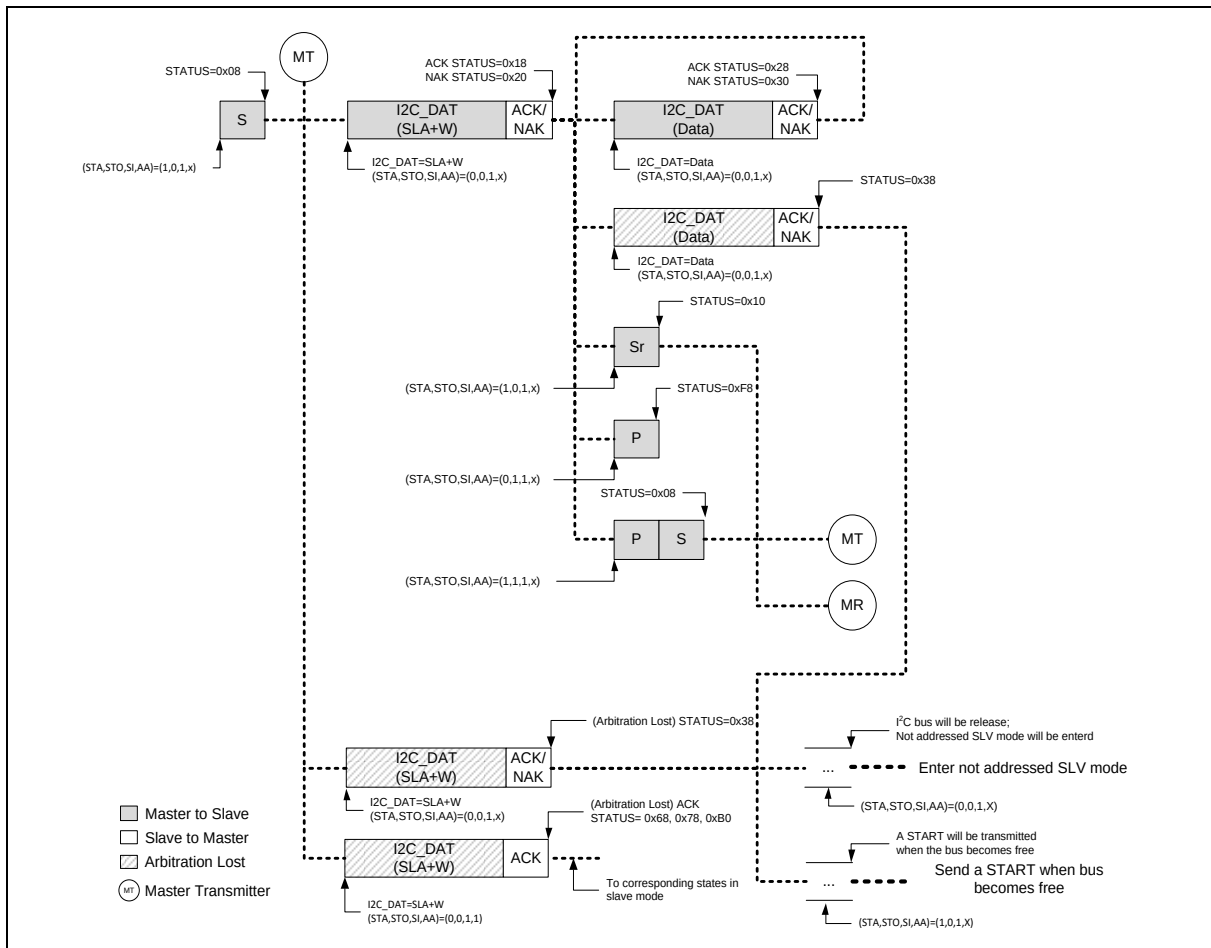


Figure 6.14-12 Master Transmitter Mode Control Flow

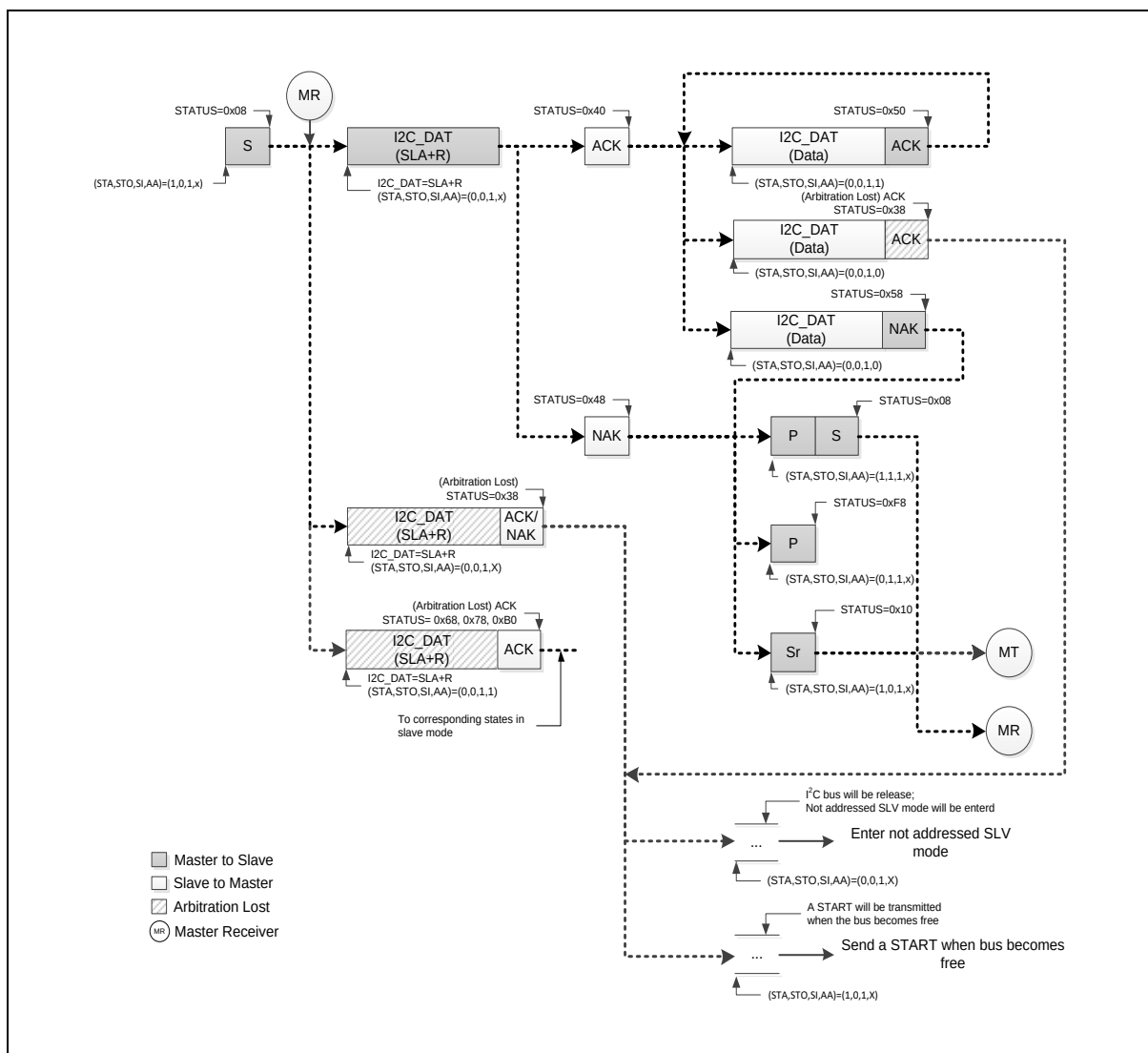


Figure 6.14-13 Master Receiver Mode Control Flow

If the I²C is in Master mode and gets arbitration lost, the status code will be 0x38. In status 0x38, user may set (STA, STO, SI, AA) = (1, 0, 1, X) to send START to re-start Master operation when bus become free. Otherwise, user may set (STA, STO, SI, AA) = (0, 0, 1, X) to release I²C bus and enter not addressed Slave mode.

Slave Mode

When reset default, I²C is not addressed and will not recognize the address on I²C bus. User can set slave address by I2C_ADDRn (n=0~3) and set (STA, STO, SI, AA) = (0, 0, 1, 1) to let I²C recognize the address sent by master. Figure 6.14-14 shows all the possible flow for I²C in Slave mode. Users need to follow a proper flow (as shown in Figure 6.14-14 to implement their own I²C protocol.

If bus arbitration is lost in Master mode, I²C port switches to Slave mode immediately and can detect its own slave address in the same serial transfer. If the detected address is SLA+W (Master want to write data to Slave) after arbitration lost, the status code is 0x68. If the detected address is SLA+R (Master want to read data from Slave) after arbitration lost, the status code is 0xB0.

Note: During I²C communication, the SCL clock will be released when writing '1' to clear SI flag in Slave mode.

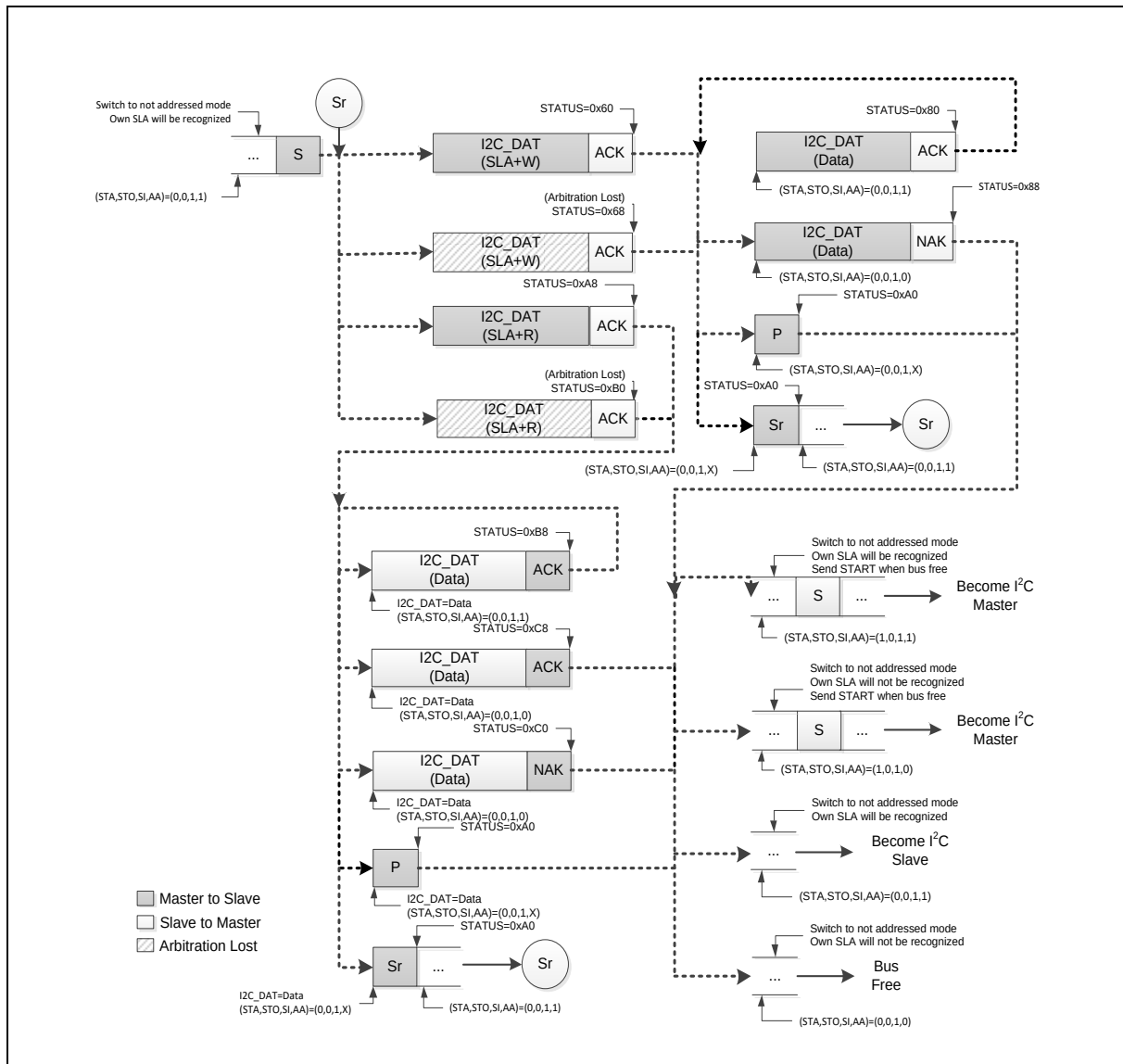


Figure 6.14-14 Slave Mode Control Flow

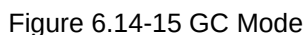
If I²C is still receiving data in addressed Slave mode but got a STOP or Repeat START, the status code will be 0xA0. User could follow the action for status code 0x88 as shown in the above figure when getting 0xA0 status.

If I²C is still transmitting data in addressed Slave mode but got a STOP or Repeat START, the status code will be 0xA0. User could follow the action for status code 0xC8 as shown in the above figure when getting 0xA0 status.

Note: After slave gets status of 0x88, 0xC8, 0xC0 and 0xA0, slave can switch to not address mode and own SLA will not be recognized. If entering this status, slave will not receive any I²C signal or address from master. At this status, I²C should enter idle mode.

General Call (GC) Mode

If the GC bit (I2C_ADDRn [0]) is set, the I²C port hardware will respond to General Call address (00H). User can clear GC bit to disable general call function. When the GC bit is set and the I²C in Slave mode, it can receive the general call address by 0x00 after master send general call address to I²C bus, then it will follow status of GC mode.



Note: After slave gets status of 0x98 and 0xA0, slave can switch to not address mode and own SLA will not be recognized. If entering this status, slave will not receive any I²C signal or address from master. At this time, the I²C controller should enter idle mode.

In some applications, there are two or more masters on the same I²C bus to access slaves, and the masters may transmit data simultaneously. The I²C supports multi-master by including collision detection and arbitration to prevent data corruption.

If for some reason two masters initiate command at the same time, the arbitration procedure determines which master wins and can continue with the command. Arbitration is performed on the SDA signal while the SCL signal is high. Each master checks if the SDA signal on the bus corresponds to the generated SDA signal. If the SDA signal on the bus is low but it should be high, then this master has lost arbitration. The device that has lost arbitration can generate SCL pulses until the byte ends and must then release the bus and go into slave mode. The arbitration procedure can continue until all the data is transferred. This means that in multi-master system each master must monitor the bus for collisions and act accordingly.

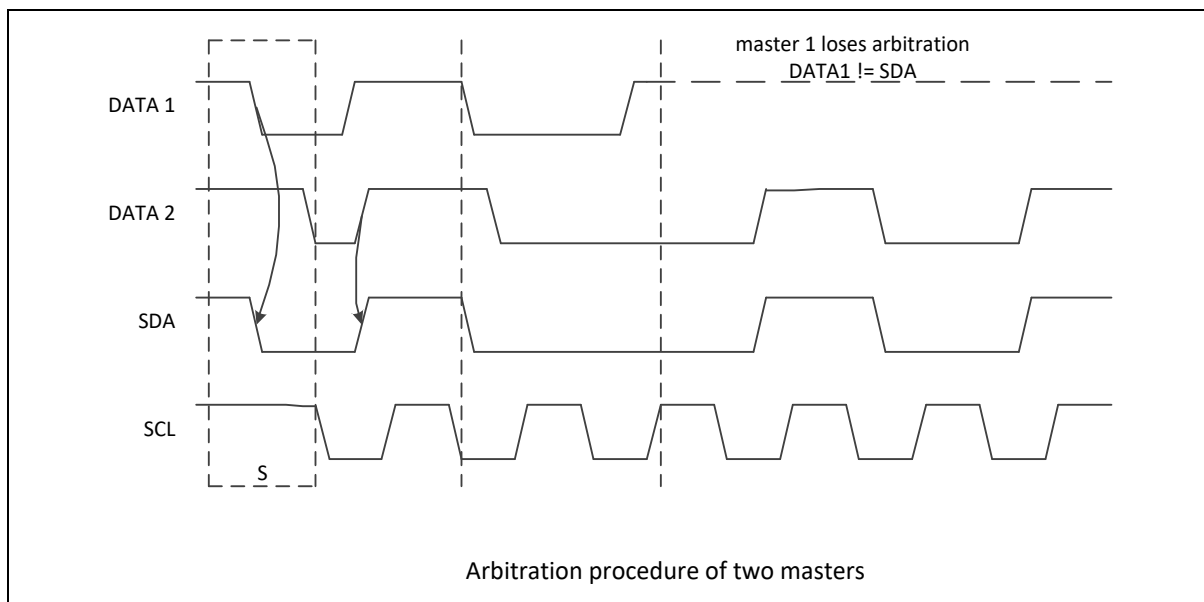


Figure 6.14-16 Arbitration Lost

- When I2C_STATUS0 = 0x38, an "Arbitration Lost" is received. Arbitration lost event maybe occur during the send START bit, data bits or STOP bit. User could set (STA, STO, SI, AA) = (1, 0, 1, X) to send START again when bus free, or set (STA, STO, SI, AA) = (0, 0, 1, X) to not addressed Slave mode. User can detect bus free by ONBUSY (I2C_STATUS1 [8]).
- When I2C_STATUS0 = 0x00, a "Bus Error" is received. To recover I²C bus from a bus error, STO should be set and SI should be cleared, and then STO is cleared to release bus.
 - Set (STA, STO, SI, AA) = (0, 1, 1, X) to stop current transfer
 - Set (STA, STO, SI, AA) = (0, 0, 1, X) to release bus

6.14.4.4 PDMA Transfer Function

The I²C controller supports PDMA transfer function. When TXPDMAEN (I2C_CTL1 [0]) is set to 1, the I²C controller will issue request to PDMA controller to start the DMA transmission process automatically.

When RXPDMAEN (I2C_CTL1 [1]) is set to 1, the I²C controller will start the receive PDMA process. The I²C controller will issue the request to PDMA controller automatically when there is data written into the received BUFFER.

When I²C enters PDMA mode, the mostly status interrupt will be masked. Let the interrupt not occur besides the bus error or NACK or STOP interrupt (0x20, 0x30, 0x38, 0x48, 0x58, 0x00, 0xA0, 0xC0, 0x88 and 0x98).

Set the PDMASTR (I2C_CTL1 [8]) only the I²C controller in master TX mode. If PDMASTR is cleared to 0, I²C will send STOP automatically after PDMA transfer done and buffer empty. If PDMASTR is set to 1, SI will be set to 1 and I²C bus will be stretched by hardware after PDMA transfer done and buffer empty.

6.14.4.5 Programmable Setup and Hold Times

To guarantee a correct data setup and hold time, the timing must be configured. By programming HTCTL (I2C_TMCTL[24:16]) to configure hold time and STCTL (I2C_TMCTL[8:0]) to configure setup time.

For the delay timing, refer to peripheral clock (PCLK). When device stretches master clock, the setup and hold time configuration value will not affected by stretched.

User should focus the limitation of setup and hold time configuration, the timing setting must follow I²C protocol. Once setup time configuration greater than design limitation, that means if setup time setting make SCL output less than three PCLKs, the I²C controller cannot work normally due to SCL must sample three times. And once hold time configuration greater than I²C clock limitation, I²C will occur bus error. It is suggested that user calculate suitable timing with baud rate and protocol before setting timing. Table 6.14-1 shows the relationship between I²C baud rate and PCLK, the number of table represent one clock duty contain how many PCLKs. Setup and hold time configuration even can program some extreme values in the design, but user should follow I²C protocol standard.

I ² C Baud Rate PCLK	100k	200k	400k	800k	1200k
12 MHz	120	60	30	15	10
24 MHz	240	120	60	30	20
48 MHz	480	240	120	60	40
72 MHz	720	360	180	90	60

Table 6.14-1 Relationship between I²C Baud Rate and PCLK

For setup time wrong adjustment example, assuming one SCL cycle contains 5 PCLKs and set STCTL (I2C_TMCTL[8:0]) to 3 that stretch three PCLKs for setup time setting. The setup time maximum setting value: $ST_{limit} = (I2C\_CLKDIV[7:0] + 1) \times 2 - 6$.

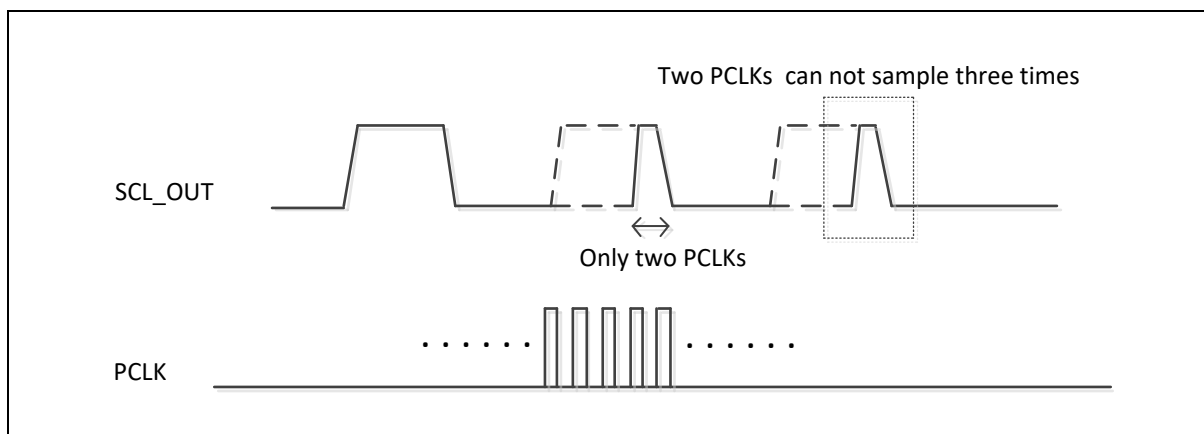


Figure 6.14-17 Setup Time Wrong Adjustment

For hold time wrong adjustment example, use I²C Baud Rate = 1200k and PCLK = 72 MHz, the SCL high/low duty = 60 PCLK. When HTCTL (I2C_TMCTL[24:16]) is set to 61 and STCTL (I2C_TMCTL[8:0]) is set to 0, then SDA output delay will over SCL high duty and cause bus error. The hold time maximum setting value: $HT_{limit} = (I2C\_CLKDIV[7:0] + 1) \times 2 - 9$.

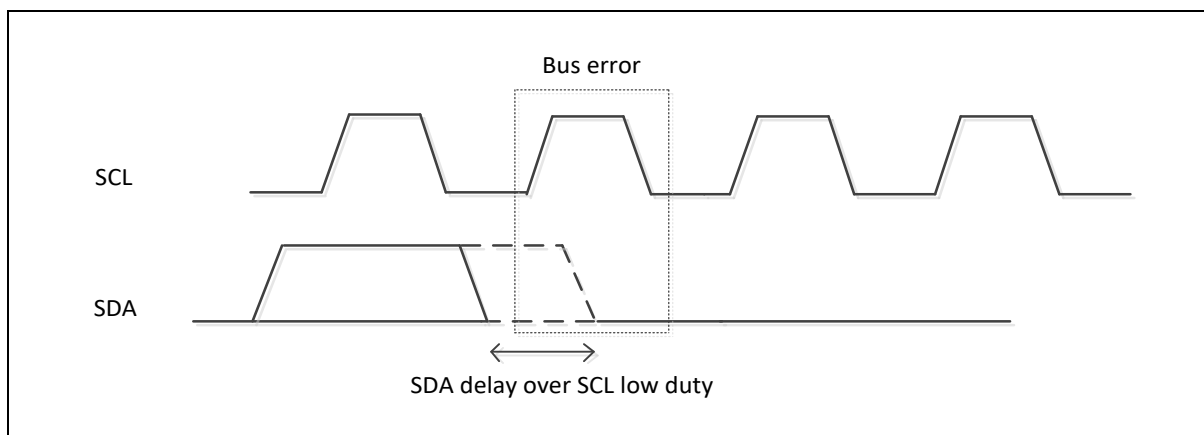


Figure 6.14-18 Hold Time Wrong Adjustment

6.14.4.6 I²C Protocol Registers

To control I²C port through the following fifteen special function registers: I2C_CTL0 (control register), I2C_STATUS0 (status register), I2C_DAT (data register), I2C_ADDRn (address registers, n=0~3), I2C_ADDRMSKn (address mask registers, n=0~3), I2C_CLKDIV (clock rate register), I2C_TOCTL (Time-out control register), I2C_WKCTL(wake up control register) and I2C_WKSTS(wake up status register).

Address Registers (I2C_ADDR)

The I²C port is equipped with four slave address registers, I2C_ADDRn (n=0~3). The contents of the register are irrelevant when I²C is in Master mode. In Slave mode, the bit field ADDR(I2C_ADDRn[7:1]) must be loaded with the chip's own slave address. The I²C hardware will react if the contents of I2C_ADDRn are matched with the received slave address.

The I²C ports support the "General Call" function. If the GC bit (I2C_ADDRn [0]) is set the I²C port hardware will respond to General Call address (00H). Clear GC bit to disable general call function.

When the GC bit is set and the I²C is in Slave mode, it can receive the general call address by 00H after Master send general call address to I²C bus, then it will follow status of GC mode.

Slave Address Mask Registers (I2C_ADDRMSK)

The I²C bus controller supports multiple address recognition with four address mask registers I2C_ADDRMSKn (n=0~3). When the bit in the address mask register is set to 1, it means the received corresponding address bit is "Don't care". If the bit is set to 0, it means the received corresponding register bit should be exactly the same as address register.

Data Register (I2C_DAT)

This register contains a byte of serial data to be transmitted or a byte which just has been received. The CPU can be read from or written to the 8-bit (I2C_DAT [7:0]) directly while it is not in the process of shifting a byte. When I²C is in a defined state and the serial interrupt flag (SI) is set, data in I2C_DAT [7:0] remains stable. While data is being shifted out, data on the bus is simultaneously being shifted in; I2C_DAT [7:0] always contains the last data byte presented on the bus.

The acknowledge bit is controlled by the I²C hardware and cannot be accessed by the CPU. Serial data is shifted into I2C_DAT [7:0] on the rising edges of serial clock pulses on the SCL line. When a byte has been shifted into I2C_DAT [7:0], the serial data is available in I2C_DAT [7:0], and the acknowledge bit (ACK or NACK) is returned by the control logic during the ninth clock pulse. In order to monitor bus status while sending data, the bus data will be shifted to I2C_DAT[7:0] when sending I2C_DAT[7:0] to bus. In the case of sending data, serial data bits are shifted out from I2C_DAT [7:0] on the falling edge of SCL clocks, and is shifted to I2C_DAT [7:0] on the rising edge of SCL clocks. Figure 6.14-19 shows I²C Data Shifting Direction.

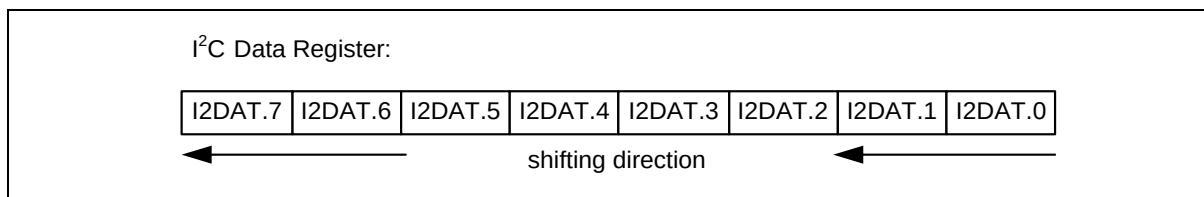


Figure 6.14-19 I²C Data Shifting Direction

Control Register (I2C_CTL0)

The CPU can be read from and written to I2C_CTL0 [7:0] directly. When the I²C port is enabled by setting I2CEN (I2C_CTL0 [6]) to high, the internal states will be controlled by I2C_CTL0 and I²C logic hardware.

There are two bits are affected by hardware: the SI bit is set when the I²C hardware requests a serial interrupt, and the STO bit is cleared when a STOP condition is present on the bus. The STO bit is also cleared when I2CEN = 0.

Once a new status code is generated and stored in I2C_STATUS0, the I²C Interrupt Flag bit SI (I2C_CTL0 [3]) will be set automatically. If the Enable Interrupt bit INTEN (I2C_CTL0 [7]) is set at this time, the I²C interrupt will be generated. The bit field I2C_STATUS0[7:0] stores the internal state code, the content keeps stable until SI is cleared by software.

Status Register (I2C_STATUS0)

I2C_STATUS0 [7:0] is an 8-bit read-only register. The bit field I2C_STATUS0 [7:0] contains the status code and there are 26 possible status codes. All states are listed in Table 6.14-2. When I2C_STATUS0 [7:0] is F8H, no serial interrupt is requested. All other I2C_STATUS0 [7:0] values correspond to the defined I²C states. When each of these states is entered, a status interrupt is requested (SI = 1). A valid status code is present in I2C_STATUS0[7:0] one cycle PCLK after SI set by hardware and is still present one cycle PCLK after SI reset by software.

In addition, the state 00H stands for a Bus Error, which occurs when a START or STOP condition is present at an incorrect position in the I²C format frame. A Bus Error may occur during the serial transfer of an address byte, a data byte or an acknowledge bit. To recover I²C from bus error, STO should be set and SI should be cleared to enter Not Addressed Slave mode. Then STO is cleared to release bus and to wait for a new communication. The I²C bus cannot recognize stop condition during this action

when a bus error occurs.

Master Mode		Slave Mode	
STATUS	Description	STATUS	Description
0x08 ^[1]	Start	0xA0	Slave Transmit Repeat Start or Stop
0x10 ^[1]	Master Repeat Start	0xA8 ^[1]	Slave Transmit Address ACK
0x18 ^[1]	Master Transmit Address ACK	0xB8 ^[1]	Slave Transmit Data ACK
0x20	Master Transmit Address NACK	0xC0	Slave Transmit Data NACK
0x28 ^[1]	Master Transmit Data ACK	0xC8 ^[1]	Slave Transmit Last Data ACK
0x30	Master Transmit Data NACK	0x60 ^[1]	Slave Receive Address ACK
0x38	Master Arbitration Lost	0x68 ^[1]	Slave Receive Arbitration Lost
0x40 ^[1]	Master Receive Address ACK	0x80 ^[1]	Slave Receive Data ACK
0x48	Master Receive Address NACK	0x88	Slave Receive Data NACK
0x50 ^[1]	Master Receive Data ACK	0x70 ^[1]	GC mode Address ACK
0x58	Master Receive Data NACK	0x78 ^[1]	GC mode Arbitration Lost
0x00	Bus error	0x90 ^[1]	GC mode Data ACK
		0x98	GC mode Data NACK
		0xB0 ^[1]	Address Transmit Arbitration Lost
0xF0	If the BMDEN =1 and the ACKMEN bit is enabled, the information of I2C_STATUS0 will be fixed as 0xF0 in slave receive condition.		
0xF8	Bus Released Note: Status “0xF8” exists in both master/slave modes, and it won’t raise interrupt.		
Note [1]: No interrupt in PDMA mode			

Table 6.14-2 I²C Status Code Description

Clock Baud Rate Bits (I2C_CLKDIV)

The data baud rate of I²C is determined by DIVIDER(I2C_CLKDIV [7:0]) register when I²C is in Master mode, and it is not necessary in a Slave mode. In the Slave mode, I²C will automatically synchronize it with any clock frequency from master I²C device. In the slave mode, system clock frequency should be greater than I²C bus maximum clock 20 times.

The data baud rate of I²C setting is Data Baud Rate of I²C = (system clock) / (4x (I2C_CLKDIV [7:0] +1)). If system clock = 16 MHz, the I2C_CLKDIV [7:0] = 40 (28H), the data baud rate of I²C = 16 MHz / (4x (40 +1)) = 97.5 Kbits/sec.

Time-out Control Register (I2C_TOCTL)

There is a 14-bit time-out counter which can be used to deal with the I²C bus hang-up. If the time-out counter is enabled, the counter starts up counting until it overflows (TOIF=1) and generates I²C interrupt to CPU or stops counting by clearing TOCEN to 0. When time-out counter is enabled, writing 1 to the SI flag will reset counter and re-start up counting after SI is cleared. If I²C bus hangs up, it causes the I2C_STATUS0 and flag SI are not updated for a period, the 14-bit time-out counter may overflow and acknowledge CPU the I²C interrupt. Refer to Figure 6.14-20 for the 14-bit time-out counter. User may write 1 to clear TOIF to 0.

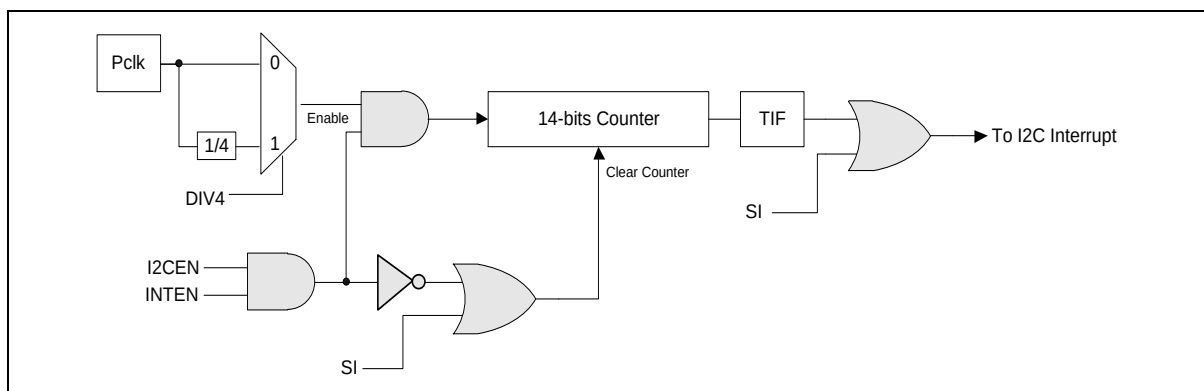


Figure 6.14-20 I²C Time-out Count Block Diagram

Wake-up Control Register (I2C_WKCTL)

When chip enters Power-down mode and sets WKEN (I2C_WKCTL [0]) to 1, other I²C master can wake up the chip by addressing the I²C device, user must configure the related setting before entering sleep mode. The ACK bit cycle of address match frame is done in power-down. The controller will stretch the SCL to low when the address is matched the device's address and the ACK cycle done, then the I²C controller will go ahead. If NHDBUSEN (I2C_WKCTL [7]) is set, the controller will not stretch the SCL to low. Note that when the controller doesn't stretch the SCL to low, transmit or receive data will perform immediately. If data transmitted or received when SI event is not clear, user must reset the I²C controller and execute the original operation again.

Wake-up Status Register (I2C_WKSTS)

When system is woken up by other I²C master device, WKIF is set to indicate this event. User needs write "1" to clear this bit.

When the chip is woken-up by address match with one of the device address register (I2C_ADDRn), the user shall check the WKAKDONE (I2C_WKSTS [1]) bit is set to 1 to confirm the address byte has done. The WKAKDONE bit indicates that the ACK bit cycle of address byte is done in power-down. The controller will stretch the SCL to low when the address is matched the device's slave address and the ACK cycle done. The SCL is stretched until WKAKDONE is clear by user. If the frequency of SCL is low speed and the system has wakeup from address match frame, the user shall check WKAKDONE to confirm this frame has transaction done and then to do the wakeup procedure. Note that user cannot release WKIF through clearing the WKAKDONE bit to 0.

The WRSTSWK (I2C_WKSTS [2]) bit records the Read/Write command before the I²C controller sends address. The user can read this bit's status to prepare the next transmitted data (WRSTSWK = 0) or to wait the incoming data (WRSTSWK = 1) can be stored in time after the system is woken up by the address match frame. Note that the WRSTSWK (I2C_WKSTS [2]) bit is cleared when writing one to the WKAKDONE (I2C_WKSTS [1]) bit.

When system is woken up by other I²C master device, WKIF is set to indicate this event. User needs to write "1" to clear this bit.

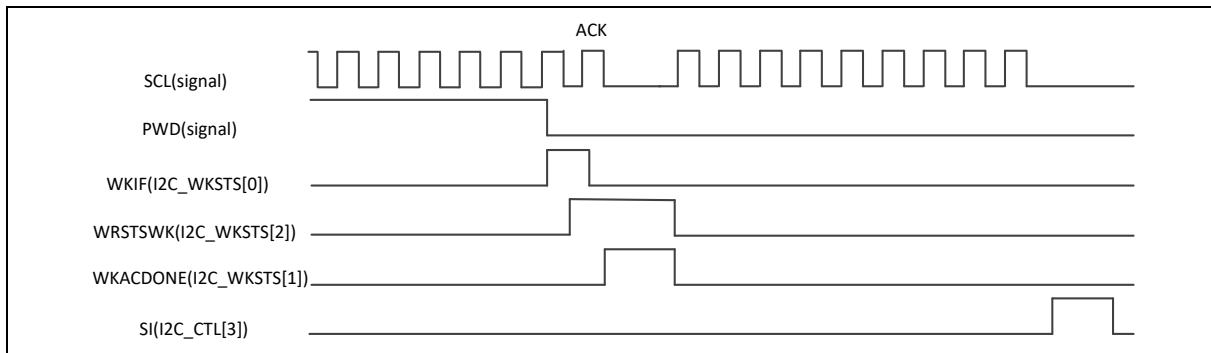


Figure 6.14-21 I²C Wake-Up Related Signals Waveform

I²C Control Register 1 (I2C_CTL1)

If enable 10-bit addressing mode ADDR10EN (I2C_CTL1 [9]) is set, the I²C will run in 10-bit mode.

For PDMA function, set TXPDMAEN (I2C_CTL1 [0]) and RXPDMAEN (I2C_CTL1 [1]) can be set to operate. And set PDMARST (I2C_CTL1 [2]) to reset the PDMA control logic.

I²C Status Register 1 (I2C_STATUS1)

The I²C controller supports four slave address flag registers, ADMAT0, ADMAT1, ADMAT2 and ADMAT3 (I2C_STATUS1[3:0]). Every control register represent which address is used and set 1 to inform software.

I²C Timing Configure Control Register (I2C_TMCTL)

In order to configure setup/hold time, the HTCTL (I2C_TMCTL[24:16]) and STCTL (I2C_TMCTL[8:0]) are set based on actual demand.

6.14.4.7 Example for Random Read on EEPROM

The following steps are used to configure the I2C0 related registers when using I²C to read data from EEPROM.

1. Set I2C0 the multi-function pin as SCL and SDA pins. The multi-function configuration reference Basic Configuration.
2. Enable I2C0 APB clock. The clock configuration reference Basic Configuration.
3. Set I2C0RST=1 to reset I2C0 controller then set I2C0 controller to normal operation. The reset controller configuration reference Basic Configuration.
4. Set I2CEN=1 to enable I2C0 controller in the "I2C_CTL0" register.
5. Give I2C0 clock a divided register value for I²C clock rate in the "I2C_CLKDIV".
6. Enable system I2C0 IRQ in system "NVIC" control register.
7. Set INTEN=1 to enable I2C0 Interrupt in the "I2C_CTL0" register.
8. Set I2C0 address registers "I2C_ADDR0 ~ I2C_ADDR3".

Random read operation is one of the methods of access EEPROM. The method allows the master to access any address of EEPROM space. Figure 6.14-22 shows the EEPROM random read operation.

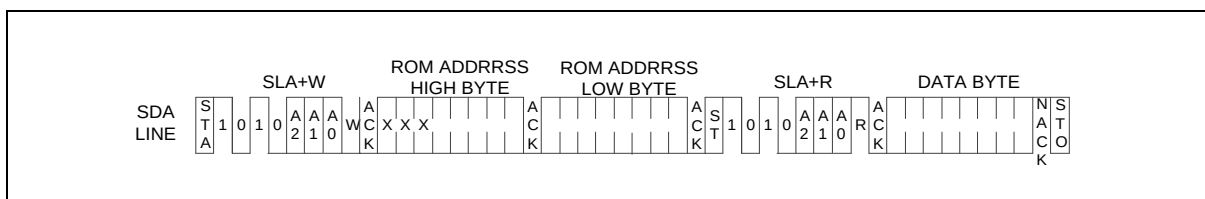


Figure 6.14-22 EEPROM Random Read

Figure 6.14-23 shows how to use the I²C controller to implement the protocol of EEPROM random read.

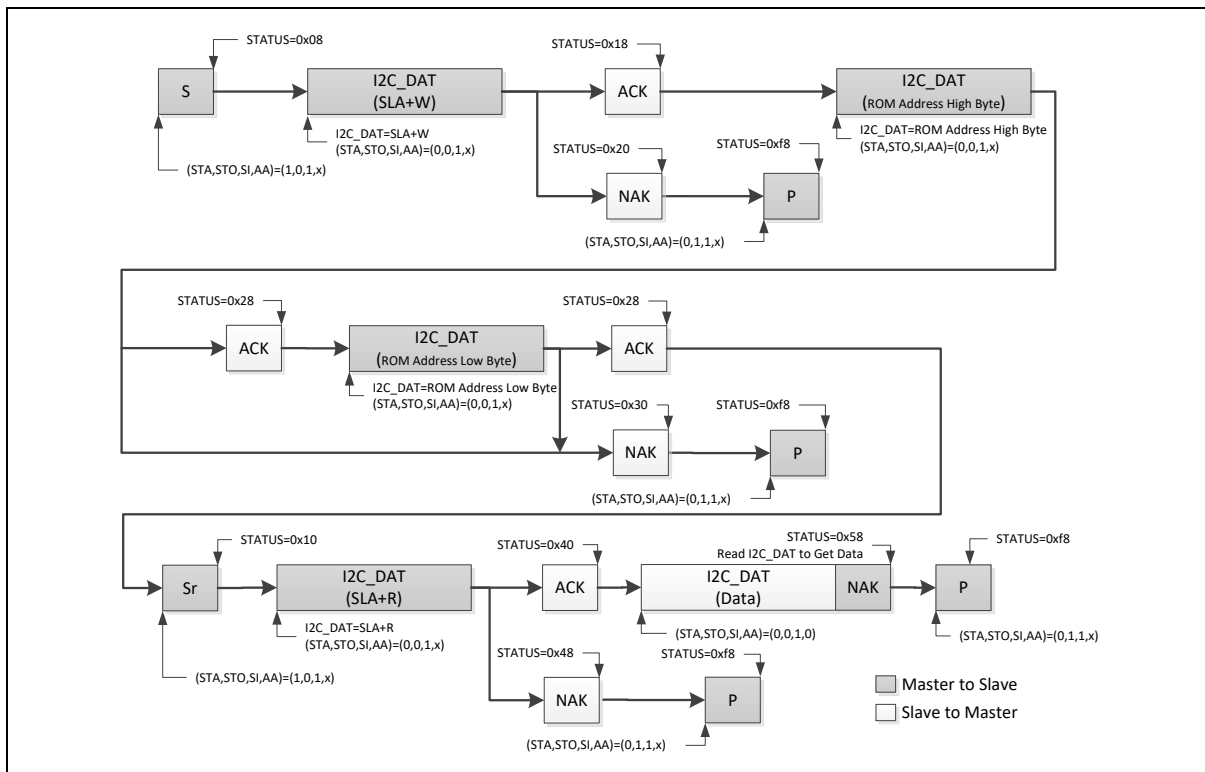


Figure 6.14-23 Protocol of EEPROM Random Read

The I²C controller, which is a master, sends START to bus. Then, it sends a SLA+W (Slave address + Write bit) to EEPROM followed by two bytes data address to set the EEPROM address to read. Finally, a Repeat START followed by SLA+R is sent to read the data from EEPROM.

6.14.5 Register Map

R: read only, W: write only, R/W: both read and write

Register	Offset	R/W	Description	Reset Value
I²C Base Address: $I2Cn\_BA = 0x4008\_0000 + (0x1000 * n)$ $n = 0,1$				
I2C_CTL0	I2Cn_BA+0x00	R/W	I ² C Control Register 0	0x0000_0000
I2C_ADDR0	I2Cn_BA+0x04	R/W	I ² C Slave Address Register0	0x0000_0000
I2C_DAT	I2Cn_BA+0x08	R/W	I ² C Data Register	0x0000_0000
I2C_STATUS0	I2Cn_BA+0x0C	R	I ² C Status Register 0	0x0000_00F8
I2C_CLKDIV	I2Cn_BA+0x10	R/W	I ² C Clock Divided Register	0x0000_0000
I2C_TOCTL	I2Cn_BA+0x14	R/W	I ² C Time-out Control Register	0x0000_0000
I2C_ADDR1	I2Cn_BA+0x18	R/W	I ² C Slave Address Register1	0x0000_0000
I2C_ADDR2	I2Cn_BA+0x1C	R/W	I ² C Slave Address Register2	0x0000_0000
I2C_ADDR3	I2Cn_BA+0x20	R/W	I ² C Slave Address Register3	0x0000_0000
I2C_ADDRMSK0	I2Cn_BA+0x24	R/W	I ² C Slave Address Mask Register0	0x0000_0000
I2C_ADDRMSK1	I2Cn_BA+0x28	R/W	I ² C Slave Address Mask Register1	0x0000_0000
I2C_ADDRMSK2	I2Cn_BA+0x2C	R/W	I ² C Slave Address Mask Register2	0x0000_0000
I2C_ADDRMSK3	I2Cn_BA+0x30	R/W	I ² C Slave Address Mask Register3	0x0000_0000
I2C_WKCTL	I2Cn_BA+0x3C	R/W	I ² C Wake-up Control Register	0x0000_0000
I2C_WKSTS	I2Cn_BA+0x40	R/W	I ² C Wake-up Status Register	0x0000_0000
I2C_CTL1	I2Cn_BA+0x44	R/W	I ² C Control Register 1	0x0000_0000
I2C_STATUS1	I2Cn_BA+0x48	R/W	I ² C Status Register 1	0x0000_0000
I2C_TMCTL	I2Cn_BA+0x4C	R/W	I ² C Timing Configure Control Register	0x0002_0000

6.14.6 Register Description

I²C Control Register (I2C_CTL0)

Register	Offset	R/W	Description	Reset Value
I2C_CTL0	I2Cn_BA+0x00	R/W	I ² C Control Register 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
INTEN	I2CEN	STA	STO	SI	AA	Reserved	

Bits	Description
[31:8]	Reserved Reserved.
[7]	INTEN Enable Interrupt 0 = I ² C interrupt Disabled. 1 = I ² C interrupt Enabled.
[6]	I2CEN I²C Controller Enable Bit Set to enable I ² C serial function controller. When I2CEN=1 the I ² C serial function enable. The multi-function pin function must set to SDA, and SCL of I ² C function first. 0 = I ² C controller Disabled. 1 = I ² C controller Enabled.
[5]	STA I²C START Control Setting STA to logic 1 to enter Master mode, the I ² C hardware sends a START or Repeat START condition to bus when the bus is free.
[4]	STO I²C STOP Control In Master mode, setting STO to transmit a STOP condition to bus then I ² C controller will check the bus condition if a STOP condition is detected. This bit will be cleared by hardware automatically.
[3]	SI I²C Interrupt Flag When a new I ² C state is present in the I2C_STATUS0 register, the SI flag is set by hardware. If bit INTEN (I2C_CTL0 [7]) is set, the I ² C interrupt is requested. SI must be cleared by software. Clear SI by writing 1 to this bit. For ACKMEN is set in slave read mode, the SI flag is set in 8th clock period for user to confirm the acknowledge bit and 9th clock period for user to read the data in the data buffer.
[2]	AA Assert Acknowledge Control When AA =1 prior to address or data is received, an acknowledged (low level to SDA) will be returned during the acknowledge clock pulse on the SCL line when 1.) A slave is acknowledging the address sent from master, 2.) The receiver devices are acknowledging the data sent by transmitter. When AA=0 prior to address or data received, a Not acknowledged (high level to SDA) will be returned during the acknowledge clock pulse on the SCL line.
[1:0]	Reserved Reserved.

I²C Data Register (I2C_DAT)

Register	Offset	R/W	Description	Reset Value
I2C_DAT	I2Cn_BA+0x08	R/W	I ² C Data Register	0x0000_0000

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

Bits	Description	
[31:8]	Reserved	Reserved.
[7:0]	DAT	I²C Data Bit [7:0] is located with the 8-bit transferred/received data of I ² C serial port.

I²C Status Register (I2C_STATUS0)

Register	Offset	R/W	Description	Reset Value
I2C_STATUS0	I2Cn_BA+0x0C	R	I ² C Status Register 0	0x0000_00F8

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

Bits	Description
[31:8]	Reserved Reserved.
[7:0]	I²C Status STATUS The three least significant bits are always 0. The five most significant bits contain the status code. There are 28 possible status codes. When the content of I2C_STATUS0 is F8H, no serial interrupt is requested. Others I2C_STATUS0 values correspond to defined I ² C states. When each of these states is entered, a status interrupt is requested (SI = 1). A valid status code is present in I2C_STATUS0 one cycle after SI is set by hardware and is still present one cycle after SI has been reset by software. In addition, states 00H stands for a Bus Error. A Bus Error occurs when a START or STOP condition is present at an illegal position in the formation frame. Example of illegal position are during the serial transfer of an address byte, a data byte or an acknowledge bit.

I²C Clock Divided Register (I2C_CLKDIV)

Register	Offset	R/W	Description	Reset Value
I2C_CLKDIV	I2Cn_BA+0x10	R/W	I ² C Clock Divided Register	0x0000_0000

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

Bits	Description
[31:12]	Reserved Reserved.
[9:0]	DIVIDER I ² C Clock Divided Indicates the I ² C clock rate: Data Baud Rate of I ² C = (system clock) / (4x (I2C_CLKDIV+1)). Note: The minimum value of I2C_CLKDIV is 4.

I²C Time-out Control Register (I2C_TOCTL)

Register	Offset	R/W	Description	Reset Value
I2C_TOCTL	I2Cn_BA+0x14	R/W	I ² C Time-out Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved					TOCEN	TOCDIV4	TOIF

Bits	Description
[31:3]	Reserved Reserved.
[2]	TOCEN Time-out Counter Enable Bit When enabled, the 14-bit time-out counter will start counting when SI is cleared. Setting flag SI to '1' will reset counter and re-start up counting after SI is cleared. 0 = Time-out counter Disabled. 1 = Time-out counter Enabled.
[1]	TOCDIV4 Time-out Counter Input Clock Divided by 4 When enabled, the time-out period is extended 4 times. 0 = Time-out period is extend 4 times Disabled. 1 = Time-out period is extend 4 times Enabled.
[0]	TOIF Time-out Flag This bit is set by hardware when I ² C time-out happened and it can interrupt CPU if I ² C interrupt enable bit (INTEN) is set to 1. Note: Software can write 1 to clear this bit.

I²C Slave Address Register (ADDRx)

Register	Offset	R/W	Description	Reset Value
I2C_ADDR0	I2Cn_BA+0x04	R/W	I ² C Slave Address Register0	0x0000_0000
I2C_ADDR1	I2Cn_BA+0x18	R/W	I ² C Slave Address Register1	0x0000_0000
I2C_ADDR2	I2Cn_BA+0x1C	R/W	I ² C Slave Address Register2	0x0000_0000
I2C_ADDR3	I2Cn_BA+0x20	R/W	I ² C Slave Address Register3	0x0000_0000

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

Bits	Description
[31:11]	Reserved Reserved.
[10:1]	ADDR I²C Address The content of this register is irrelevant when I ² C is in Master mode. In the slave mode, the seven most significant bits must be loaded with the chip's own address. The I ² C hardware will react if either of the address is matched. Note: When software sets 10'h000, the address can not be used.
[0]	GC General Call Function 0 = General Call Function Disabled. 1 = General Call Function Enabled.

I²C Slave Address Mask Register (ADDRMSKx)

Register	Offset	R/W	Description	Reset Value
I2C_ADDRMSK0	I2Cn_BA+0x24	R/W	I ² C Slave Address Mask Register0	0x0000_0000
I2C_ADDRMSK1	I2Cn_BA+0x28	R/W	I ² C Slave Address Mask Register1	0x0000_0000
I2C_ADDRMSK2	I2Cn_BA+0x2C	R/W	I ² C Slave Address Mask Register2	0x0000_0000
I2C_ADDRMSK3	I2Cn_BA+0x30	R/W	I ² C Slave Address Mask Register3	0x0000_0000

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

Bits	Description
[31:11]	Reserved Reserved.
[10:1]	I²C Address Mask 0 = Mask Disabled (the received corresponding register bit should be exact the same as address register.). 1 = Mask Enabled (the received corresponding address bit is don't care.). ADDRMSK I ² C bus controllers support multiple address recognition with four address mask register. When the bit in the address mask register is set to one, it means the received corresponding address bit is don't-care. If the bit is set to zero, that means the received corresponding register bit should be exact the same as address register. Note: The wake-up function cannot use address mask.
[0]	Reserved Reserved.

I²C Wake-up Control Register (I2C_WKCTL)

Register	Offset	R/W	Description	Reset Value
I2C_WKCTL	I2Cn_BA+0x3C	R/W	I ² C Wake-up Control Register	0x0000_0000

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

Bits	Description
[31:8]	Reserved Reserved.
[7]	NHDBUSEN I²C No Hold BUS Enable Bit 0 = I ² C holds bus after wake-up. 1 = I ² C does not hold bus after wake-up. Note: The I ² C controller could respond when WKIF event is not cleared. It may cause error data transmitted or received. If data transmitted or received when WKIF event is not cleared, user must reset I ² C controller and execute the original operation again.
[6:1]	Reserved Reserved.
[0]	WKEN I²C Wake-up Enable Bit 0 = I ² C wake-up function Disabled. 1 = I ² C wake-up function Enabled.

I²C Wake-up Status Register (I2C_WKSTS)

Register	Offset	R/W	Description	Reset Value
I2C_WKSTS	I2Cn_BA+0x40	R/W	I ² C Wake-up Status Register	0x0000_0000

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

Bits	Description
[31:3]	Reserved Reserved.
[2]	WRSTSWK Read/Write Status Bit in Address Wakeup Frame (Read Only) 0 = Write command be record on the address match wakeup frame. 1 = Read command be record on the address match wakeup frame. Note: This bit will be cleared when software can write 1 to WKAKDONE (I2C_WKSTS[1]) bit.
[1]	WKAKDONE Wakeup Address Frame Acknowledge Bit Done 0 = The ACK bit cycle of address match frame isn't done. 1 = The ACK bit cycle of address match frame is done in power-down. Note: This bit cannot release WKIF. Software can write 1 to clear this bit.
[0]	WKIF I²C Wake-up Flag When chip is woken up from Power-down mode by I ² C, this bit is set to 1. Software can write 1 to clear this bit.

I²C Control Register 1 (I2C_CTL1)

Register	Offset	R/W	Description	Reset Value
I2C_CTL1	I2Cn_BA+0x44	R/W	I ² C Control Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved						ADDR10EN	PDMASTR
7	6	5	4	3	2	1	0
Reserved					PDMARST	RXPDMAEN	TXPDMAEN

Bits	Description
[31:8]	Reserved Reserved.
[9]	ADDR10EN Address 10-bit Function Enable Bit 0 = Address match 10-bit function Disabled. 1 = Address match 10-bit function Enabled.
[8]	PDMASTR PDMA Stretch Bit 0 = I ² C send STOP automatically after PDMA transfer done. (only master TX) 1 = I ² C SCL bus is stretched by hardware after PDMA transfer done if the SI is not cleared. (only master TX)
[7:3]	Reserved Reserved.
[2]	PDMARST PDMA Reset 0 = No effect. 1 = Reset the I ² C request to PDMA.
[1]	RXPDMAEN PDMA Receive Channel Available 0 = Receive PDMA function Disabled. 1 = Receive PDMA function Enabled.
[0]	TXPDMAEN PDMA Transmit Channel Available 0 = Transmit PDMA function Disabled. 1 = Transmit PDMA function Enabled.

I²C Status Register 1 (I2C_STATUS1)

Register	Offset	R/W	Description	Reset Value
I2C_STATUS1	I2Cn_BA+0x48	R/W	I ² C Status Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							ONBUSY
7	6	5	4	3	2	1	0
Reserved				ADMAT3	ADMAT2	ADMAT1	ADMAT0

Bits	Description
[31:8]	Reserved Reserved.
[8]	ONBUSY On Bus Busy (Read Only) Indicates that a communication is in progress on the bus. It is set by hardware when a START condition is detected. It is cleared by hardware when a STOP condition is detected. 0 = The bus is IDLE (both SCLK and SDA High). 1 = The bus is busy.
[7:4]	Reserved Reserved.
[3]	ADMAT3 I²C Address 3 Match Status When address 3 is matched, hardware will inform which address used. This bit will set to 1, and software can write 1 to clear this bit.
[2]	ADMAT2 I²C Address 2 Match Status When address 2 is matched, hardware will inform which address used. This bit will set to 1, and software can write 1 to clear this bit.
[1]	ADMAT1 I²C Address 1 Match Status When address 1 is matched, hardware will inform which address used. This bit will set to 1, and software can write 1 to clear this bit.
[0]	ADMAT0 I²C Address 0 Match Status When address 0 is matched, hardware will inform which address used. This bit will set to 1, and software can write 1 to clear this bit.

I²C Timing Configure Control Register (I2C_TMCTL)

Register	Offset	R/W	Description	Reset Value
I2C_TMCTL	I2Cn_BA+0x4C	R/W	I ² C Timing Configure Control Register	0x0002_0000

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

Bits	Description
[31:25]	Reserved Reserved.
[24:16]	HTCTL Hold Time Configure Control This field is used to generate the delay timing between SCL falling edge and SDA rising edge in transmission mode. The delay hold time is numbers of peripheral clock = HTCTL x PCLK.
[15:9]	Reserved Reserved.
[8:0]	STCTL Setup Time Configure Control This field is used to generate a delay timing between SDA falling edge and SCL rising edge in transmission mode. The delay setup time is numbers of peripheral clock = STCTL x PCLK. Note: Setup time setting should not make SCL output less than three PCLKs.

6.15 USCI - Universal Serial Control Interface Controller (USCI)

6.15.1 Overview

The Universal Serial Control Interface (USCI) is a flexible interface module covering several serial communication protocols. The user can configure this controller as UART, SPI, or I²C functional protocol.

6.15.2 Features

The controller can be individually configured to match the application needs. The following protocols are supported:

- UART
- SPI
- I²C

6.15.3 Block Diagram

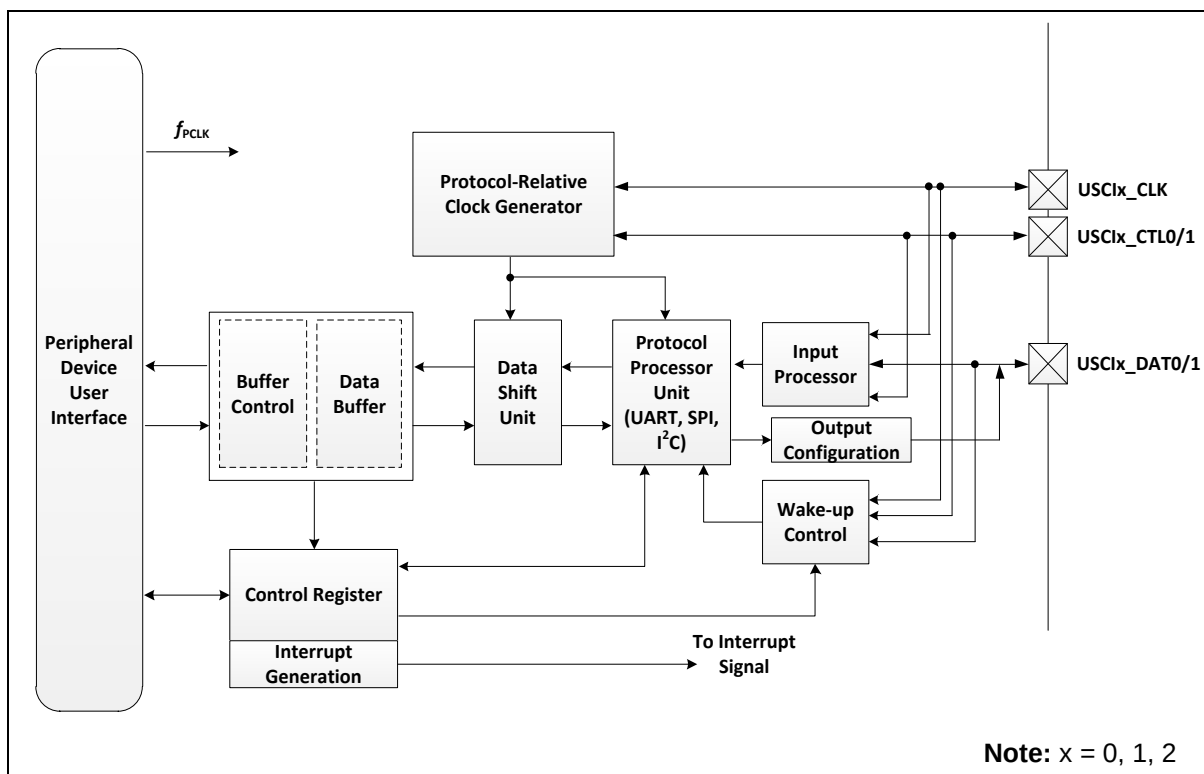


Figure 6.15-1 USCI Block Diagram

6.15.4 Functional Description

The structure of the Universal Serial Control Interface (USCI) controller is shown in Figure 6.15-1 USCI Block Diagram. The input signal is implemented in input processor. The data buffers and the data shift unit support the data transfers. Each protocol-specific function is handled by the protocol processor unit. The timing and time event control signals of the specific protocol are handled by the protocol-relative clock generator. All the protocol-specific events are processed in the interrupt generation unit. The wake-up function of the specific protocol is implemented in the wake-up control unit.

The USCI is equipped with three protocols including UART, SPI, and I²C. They can be selected by FUNMODE (USCI_CTL [2:0]). Note that the FUNMODE must be set to 0 before changing protocol.

6.15.4.1 I/O Processor

Input Signal

All input stages offer the similar feature set. They are used for all protocols.

Table 6.15-1 lists the relative input signals for each selected protocol. Each input signal is handled by an input processor for signal conditioning, such as signal inverse selection control, or a digital input filter.

Selected Protocol		UART	SPI	I ² C
Serial Bus Clock Input	USCIx_CLK	-	SPI_CLK	SCL
Control Input	USCIx_CTL0	nCTS	SPI_SS	-
	USCIx_CTL1	-	-	-
Data Input	USCIx_DAT0	RX	SPI_MOSI_0	SDA
	USCIx_DAT1	-	SPI_MISO_0	-

Table 6.15-1 Input Signals for Different Protocols

The description of protocol-specific items are given in the related protocol chapters.

General Input Structure

The input structures of data and control signals include inverter, digital filter and edge detection (data signal only).

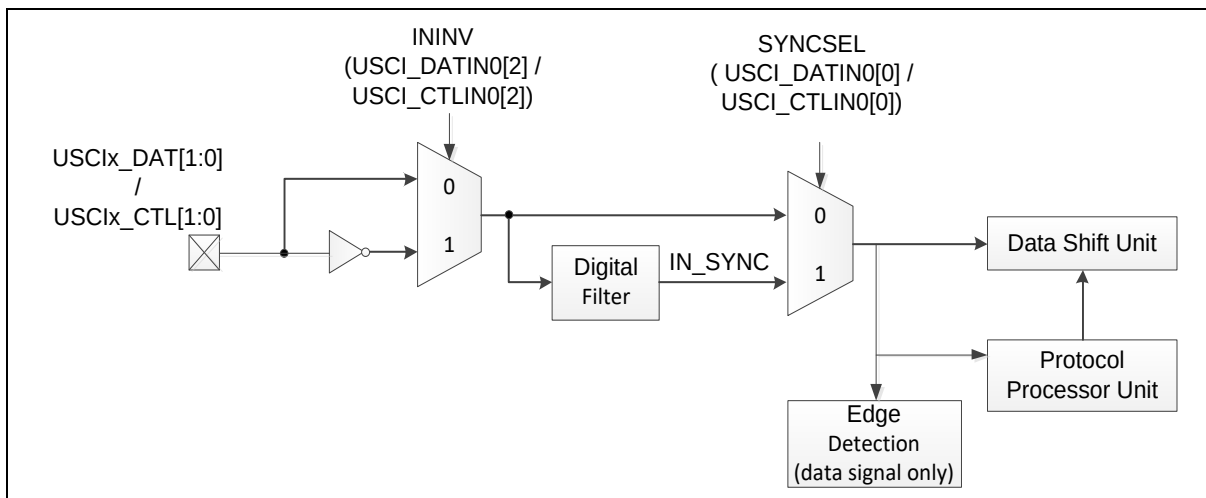


Figure 6.15-2 Input Conditioning for USCIx_DAT[1:0] and USCIx_CTL[1:0]

The input structure of USCIx_CLK is similar to USCIx_CTL[1:0] input structure, except it does not support inverse function.

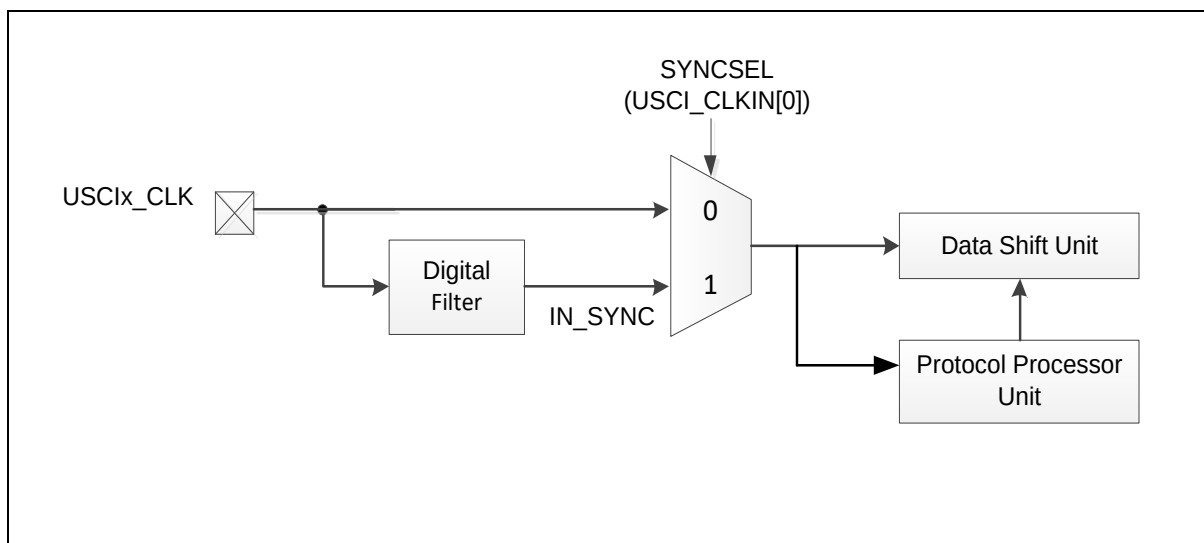


Figure 6.15-3 Input Conditioning for USCIX_CLK

All configurations of control, clock and data input structures are in USCCTLIN0, USCCLKIN and USCDATIN0 registers respectively. EDGEDET (USCDATIN0[4:3]) is used to select the edge detection condition. Note that the EDGEDET for USCDATIN0 must be set 2'b10 in UART mode. The programmable edge detection indicates that the desired event has occurred by activating the trigger signal.

ININV (USCDATIN0[2] / USCCTLIN0[2]) allows a polarity inversion of the selected input signal to adapt the input signal polarity to the internal polarity of the data shift unit and the protocol state machine.

If the SYNCSEL (USCDATIN0[0] / USCCTLIN0[0] / USCCLKIN[0]) is set to 0, the paths of input signals do not contain any delay due to synchronization or filtering. If there is noise on the input signals, the noise can be filtered from the input signal (signal IN_SYNC is synchronized to f_{PCLK}) when SYNCSEL is set to 1. The synchronization leads to a delay in the signal path of 2-3 times the period of f_{PCLK} .

Output Signals

Table 6.15-2 shows the relative output signals for each protocol. The number of actually used outputs depends on the selected protocol and they can be classified according to their meaning for the protocols.

Selected Protocol		UART	SPI	I ² C
Serial Bus Clock Output	USCIX_CLK	-	SPI_CLK	SCL
	USCIX_CTL0	-	SPI_SS	-
Control Output	USCIX_CTL1	nRTS	-	-
	USCIX_DAT0	-	SPI_MOSI_0	SDA
Data Output	USCIX_DAT1	TX	SPI_MISO_0	-

Table 6.15-2 Output Signals for Different Protocols

The description of protocol-specific items are given in the related protocol chapters.

6.15.4.2 Data Buffering

The data handling of the USCI controller is based on a Data Shift Unit (DSU) and a buffer structure. Both of the data shift and buffer registers are 16-bit wide. The inputs of Data Shift Unit include the shift data, the serial bus clock, and the shift control. The output pin of transmission can be USCIX_DAT0 pin

or USCIx_DAT1 pin depends on what protocol is selected.

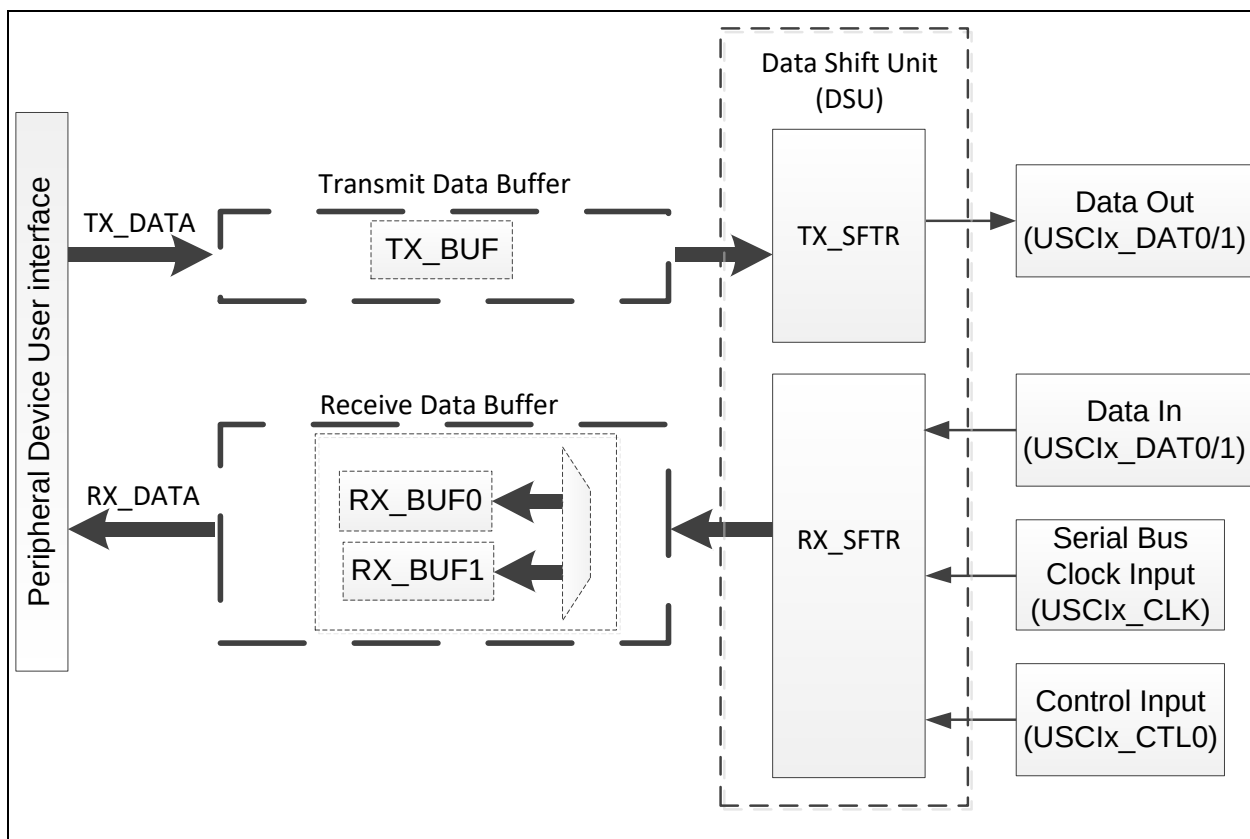


Figure 6.15-4 Block Diagram of Data Buffering

The operation of data handling includes:

- The peripheral device clock (PCLK) is used to handle data, status and control information.
- A transmitter includes transmit shift register (TX_SFTR) and a transmit data buffer (TX_BUF). The TXFULL / TXEMPTY (USCI_BUFSTS[9:8]) and TXENDIF (USCI_PROTSTS[2]) can indicate the status of transmitter.
- A receiver includes receive shift register (RX_SFTR) and a double receive buffer structure (RX_BUF0, RX_BUF1). In double buffer structure, user need not care about the reception sequence and two received data can be hold if user does not read the data of USCI_RXDAT register in time.

Data Access Structure

The Data Access Structure includes read access to received data and write access of data to be transmitted. The received data is stored in the receiver buffers including RX_BUF0 and RX_BUF1. User need not care about the reception sequence. The receive buffer can be accessed by reading USCI_RXDAT register. The first received data is read out first and the next received data becomes visible in USCI_RXDAT and can be read out next.

Transmit data can be loaded to TX_BUF by writing to the transmit register USCI_TXDAT.

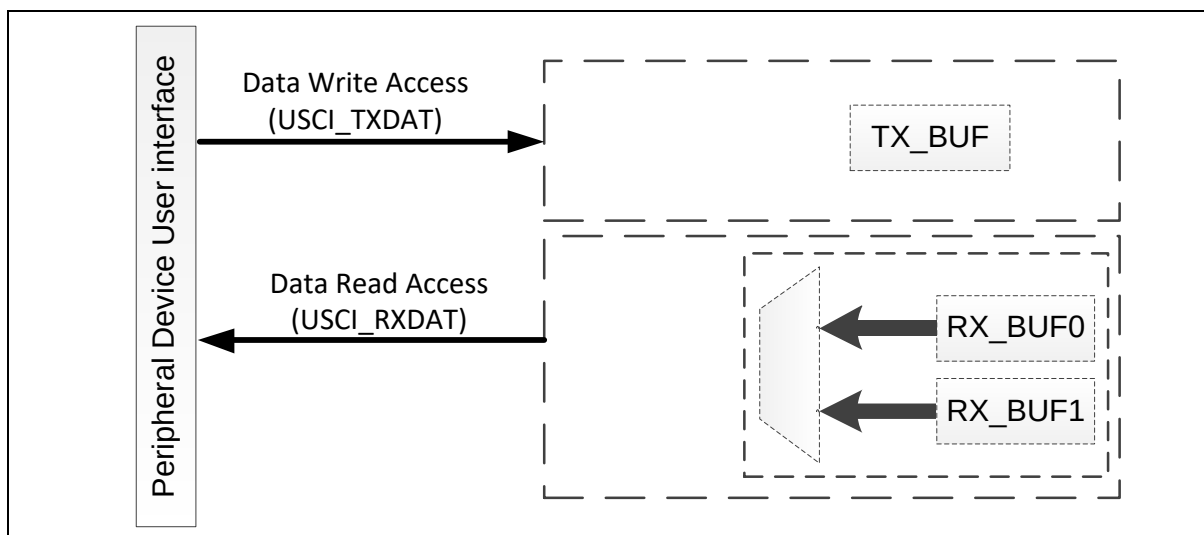


Figure 6.15-5 Data Access Structure

Transmit Data Path

The transmit data path is based on 16-bit wide transmit shift register (TX_SFTR) and transmit buffer TX_BUF. The data transfer parameters like data word length is controlled commonly for transmission and reception by the line control register USCI_LINECTL.

Transmit Buffering

The transmit shift register cannot be directly accessed by user. It is updated automatically with the value stored in the transmit buffer (TX_BUF) if a currently transmitted data is finished and new data is valid for transmission.

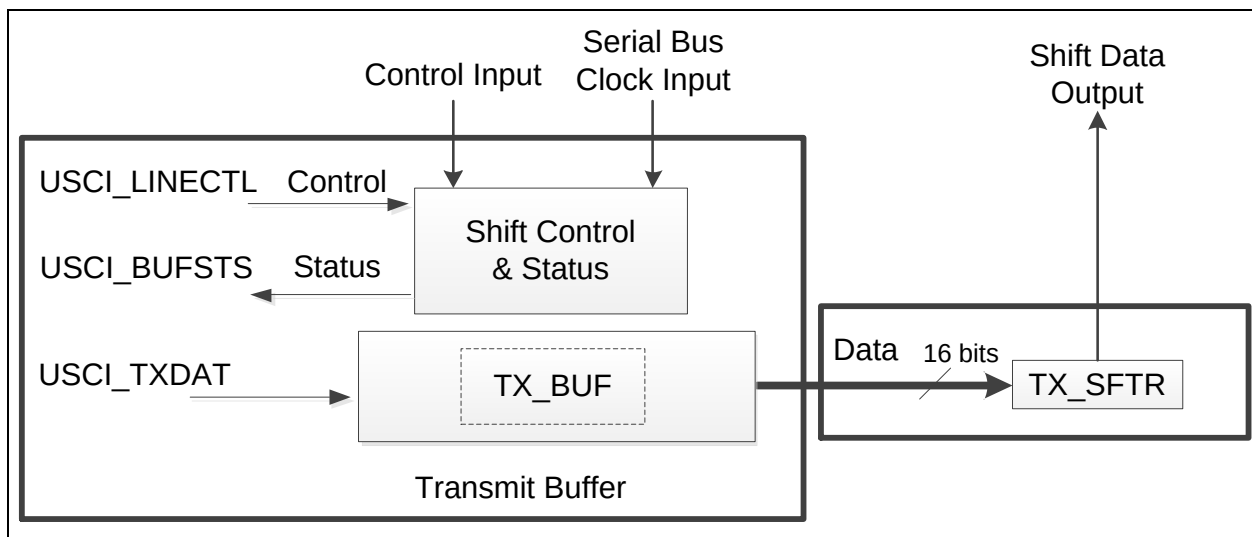


Figure 6.15-6 Transmit Data Path

Transmit Data Validation

The status of TXEMPTY (USCI_BUFSTS[8]) indicates the transmission data is valid or not in the transmit buffer (TX_BUF) and the TXSTIF (USCI_PROTSTS[1]) labels the start conditions for each data.

- If the USCI controller is a Master, the data transfer can only be started with valid data in the transmit buffer (TX_BUF). In this case, the transmit shift register is loaded with the content of transmit buffer.

Note: Master defines the start of data transfer.

- If the USCI controller is a Slave, a data transfer requested by Master and it has to be started independently of the status in transmit buffer (TX_BUF). If a data transfer is requested and started by the Master, the transmit shift register is loaded from specific protocol control signal if it is valid for transmission.

Note: Slave can not define the start itself, but has to react.

- The timing of loading data from transmit buffer to data shift unit depends on protocol configurations.
- **UART:** A transmission of the data word in transmit buffer can be started if TXEMPTY = 0 in normal operation. In auto flow control, A transmission of the data word in transmit buffer can be started while TXEMPTY = 0 and USCIX_CTL0 in active stage.
- **SPI:** In Master mode, data transmission will be started when TXEMPTY is 0. In Slave mode, the data transmission can be started only when slave selection signal is at active state and clock is presented on USCIX_CLK pin.
- **I²C:** A transmission of the data byte in transmit buffer can be started if TXEMPTY = 0.
- A transmission data which is located in transmit buffer can be started if the TXEMPTY (USCI_BUFSTS [8]) = 0. The content of the transmit buffer (in TX_BUF condition) should not be overwritten with new data while it is valid for transmission and a new transmission can start. If the content of TX_BUF has to be changed, user can set TXRST (USCI_BUFCTL [16]) to 1 to clear the content of TX_BUF before updating the data. Moreover, TXEMPTY (USCI_BUFSTS [8]) will be cleared automatically when transmit buffer (TX_BUF) is updated with new data. While a transmission is in progress, TX_BUF can be loaded with new data. User has to update the TX_BUF before a new transmission.

Receive Data Path

The receive data path is based on 16-bit wide receive shift register RX_SFTR and receive buffers RX_BUF0 and RX_BUF1. The data transfer parameters like data word length, or the shift direction are controlled commonly for transmission and reception by the line control register USCI_LINECTL. Register USCI_BUFSTS monitors the data validation of USCI_RXDAT.

Receive Buffering

The receive shift register cannot be directly accessed by user, but its content is automatically loaded into the receive buffer if a complete data word has been received or the frame is finished. The received data words in Receive Buffer can be read out automatically from register USCI_RXDAT.

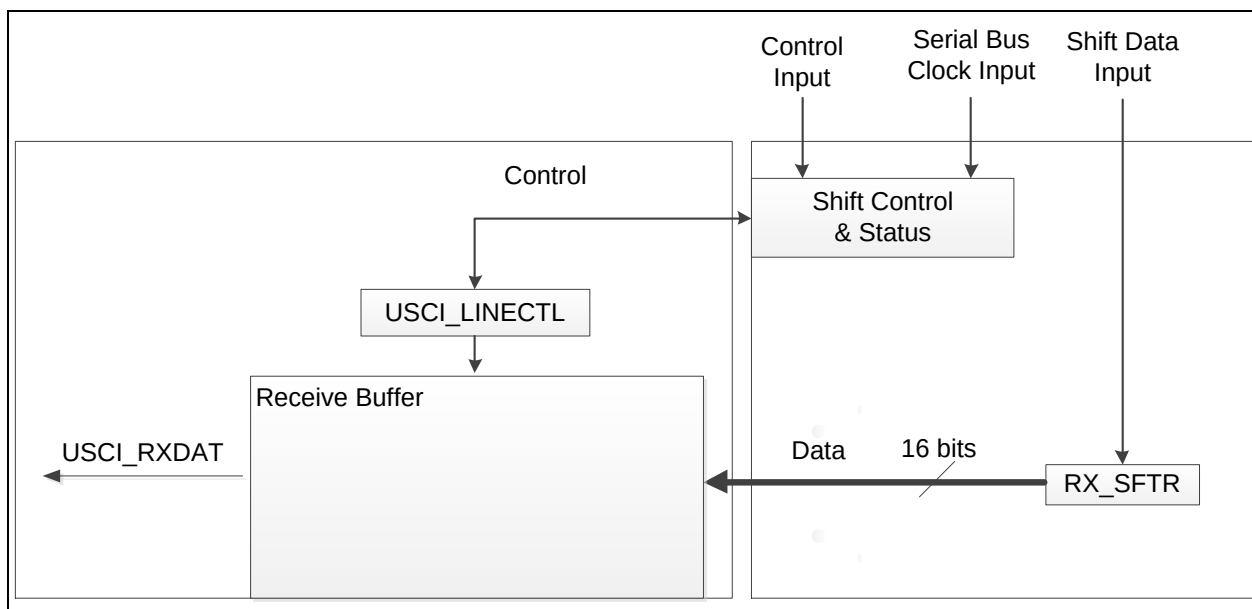


Figure 6.15-7 Receive Data Path

6.15.4.3 Port Direction Control

In SPI protocol with half-duplex configurations, the data port is bidirectional. Port direction control is intended to control the pin direction through a dedicated hardware interface.

The direction of selected pin is controlled by PORTDIR (USCI_TXDAT[16]). When user writes USCI_TXDAT register, the transmit data and its port direction are settled simultaneously.

6.15.4.4 Protocol Control and Status

The protocol-related control and status information are located in the protocol control register USCI_PROTCTL and in the protocol status register USCI_PROTSTS. These registers are shared between the available protocols. As a consequence, the meaning of the bit positions in these registers is different within the protocols. Refer to each protocol's relative register for detail information.

6.15.4.5 Protocol-Relative Clock Generator

The USCIB controller contains a protocol-relative clock generator and it is controlled by register USCI_BRGEN. It is reset when the USCI_BRGEN register is written. The structured of protocol-relative clock generator is shown in Figure 6.15-8.

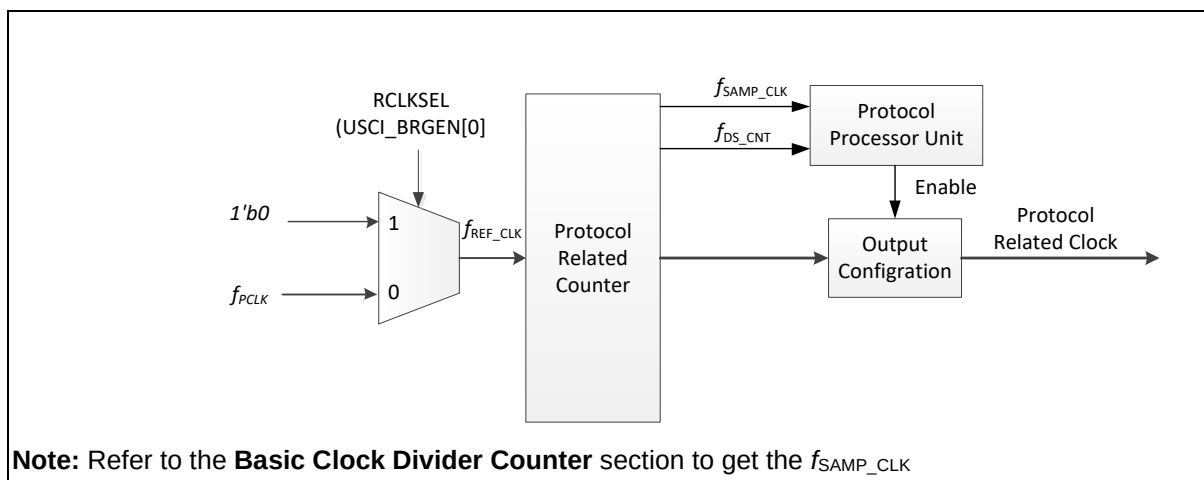


Figure 6.15-8 Protocol-Relative Clock Generator

The protocol related counter contains basic clock divider counter and timing measurement counter. It is based on a divider stages, providing the frequencies needed for the different protocols. It contains:

- The basic clock divider counter provides the protocol-relative clock signal and other protocol-related signals (f_{SAMP_CLK} and f_{DS_CLK}).
- The timing measurement counter for time interval measurement, e.g. baud rate detection on UART protocol.
- The output signals of protocol-relative clock generator can be made available on pins (e.g. USCIx_CLK for SPI).

Basic Clock Divider Counter

The basic clock divider counter is used for an integer division delivering f_{REF_CLK2} , f_{REF_CLK} , f_{DIV_CLK} , f_{SCLK} , and f_{SAMP_CLK} . The frequencies of this divider are controlled by PTCLKSEL (USCI_BRGEN [1]), CLKDIV (USCI_BRGEN [25:16]), SPCLKSEL (USCI_BRGEN [3:2]).

The basic clock divider counter is used to generate the relative protocol timing signals.

$$f_{DIV\_CLK} = f_{REF\_CLK} \times \frac{1}{CLKDIV + 1} \text{ if PTCLKSEL} = 0$$

$$f_{DIV\_CLK} = f_{REF\_CLK} \times \frac{1}{(CLKDIV + 1) \times 2} \text{ if PTCLKSEL} = 1$$

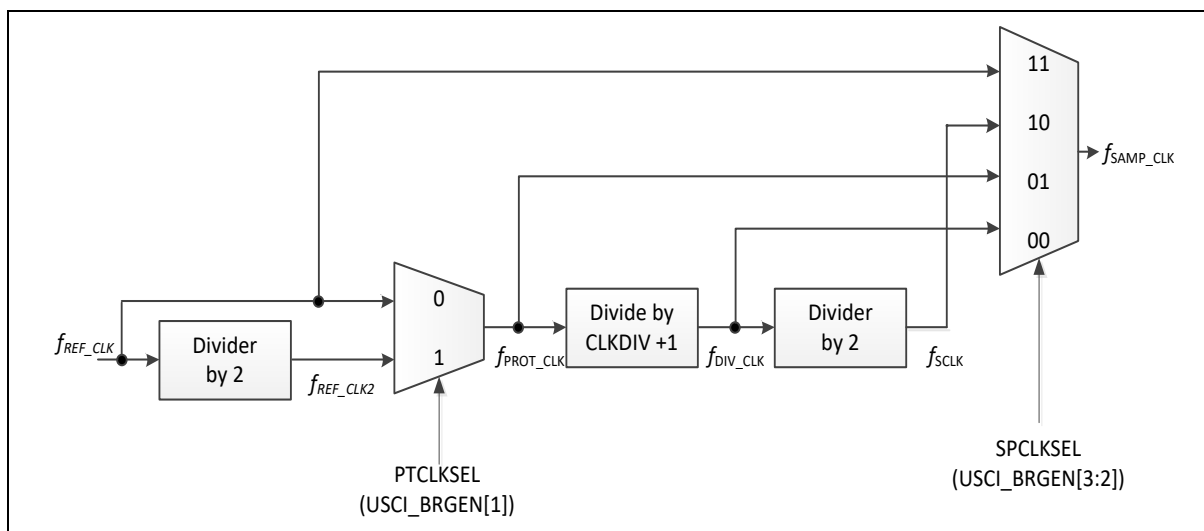


Figure 6.15-9 Basic Clock Divider Counter

Timing Measurement Counter

The timing measurement counter is used for time interval measurement and is enabled by TMCNTEN (USCI_BRGEN [4]) = 1. When TMCNTSRC (USCI_BRGEN [5]) is set to 1, the timer works on f_{DIV_CLK} , otherwise, the timer works independently from f_{PROT_CLK} . Therefore, any serial data reception or transmission can continue while the timer is performing timing measurements. The timer counts the length of protocol-related signals with f_{PROT_CLK} or f_{DIV_CLK} . It stops counting when it reaches the user-specified value.

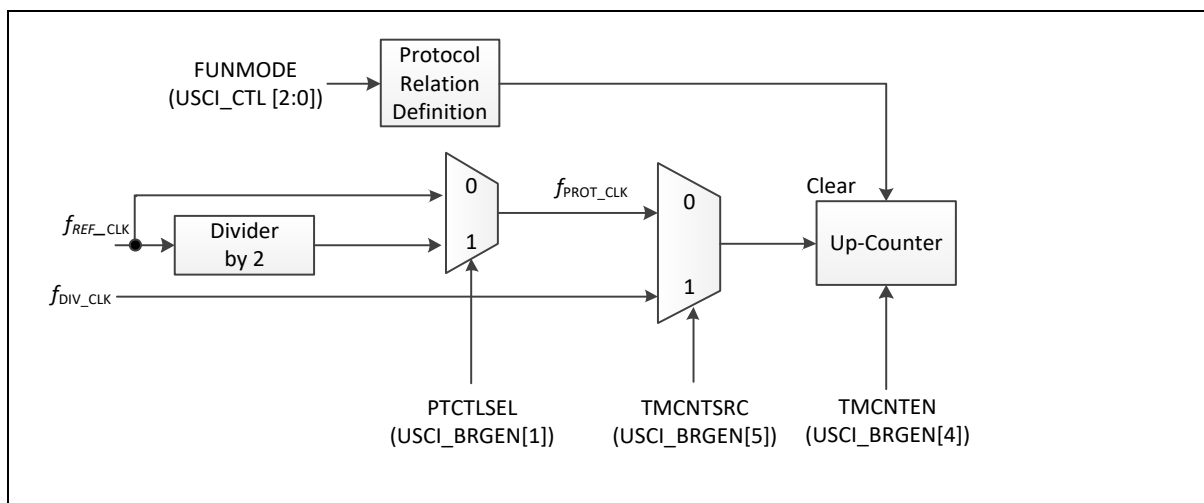


Figure 6.15-10 Block of Timing Measurement Counter

The timing measurement counter is used to perform time-out function or auto-baud rate mechanism. Its functionality depends on the selected protocol as shown below.

- UART: The timing measurement counter is used in auto baud rate detection.
- SPI: The timing measurement counter is used for counting the slave time-out period.
- I²C: The timing measurement counter indicates time-out clock cycle.

Sample Time Counter

A sample time counter associated to the protocol related counter defining protocol-specific timings, such

shift control signals or bit timings, based on the input frequency $f_{\text{SAMP\_CLK}}$. The sample time counter allows generating time intervals for protocol-specific purposes. The period of a sample frequency $f_{\text{PDS\_CNT}}$ is given by the selected input frequency $f_{\text{SAMP\_CLK}}$ and the programmed pre-divider value (PDSCNT (USCI_BRGEN [9:8])). The meaning of the sample time depends on the selected protocol. Please refer to the corresponding chapters for more protocol-specific information.

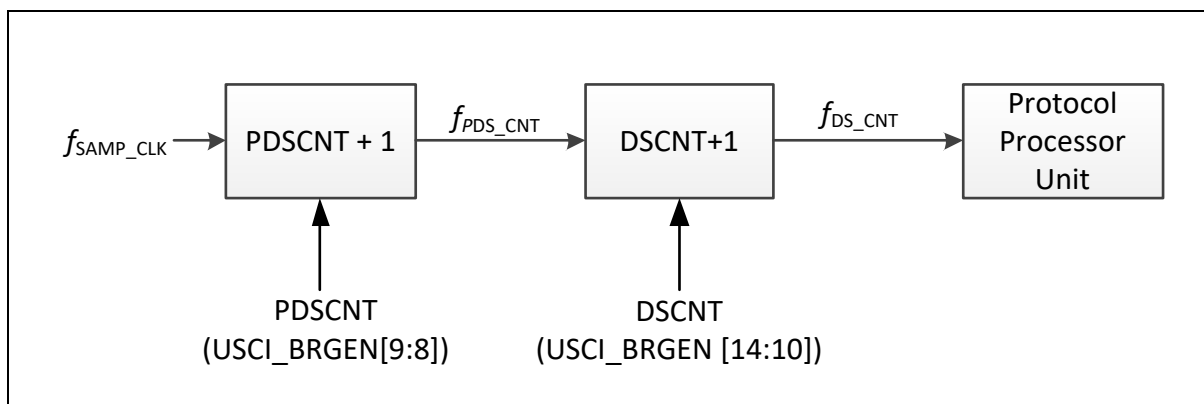


Figure 6.15-11 Sample Time Counter

6.15.4.6 Data Transfer Events and Interrupts

The data transfer events are based on the transmission or reception of a data word. The related indication flags are located in register USCI_PROTSTS. All events can be individually enabled for interrupt generation. If the FUNMODE (USCI_CTL [2:0]) is set to 0, the USCI is disabled. When FUNMODE (USCI_CTL [2:0]) is setting for a protocol port, the internal states will be controlled by logic hardware of the selected protocol.

- Transmit start interrupt event to indicate that a data word has been started:
A transmit start interrupt event occurs when the data is loaded into transmitted shift register. It is indicated by flag TXSTIF (USCI_PROTSTS [1]) and, if TXSTIEN (USCI_INTEN [1]) is enabled, it leads to transmit start interrupt.
- Transmit end interrupt event to indicate that a data word transmission has been done:
A transmit end interrupt event occurs when the current transmit data in shift register had been finished. It is indicated by flag TXENDIF (USCI_PROTSTS [2]) and, if TXENDIEN (USCI_INTEN [2]) is enabled, it leads to transmit end interrupt. This event also indicates when the shift control settings (word length, shift direction, etc.) are internally “frozen” for the current data word transmission. In UART and I²C mode, the transmit data is valid according to TXEMPTY (USCI_BUFSTS [8]) and protocol-relative internal signal with the transmit end interrupt event.
- Receiver start event to indicate that a data word reception has started:
When the receive clock edge that shifts in the first bit of a new data word is detected and reception is enabled, a receiver start event occurs. It is indicated by flag RXSTIF (USCI_PROTSTS [3]) and, if RXSTIEN (USCI_INTEN [3]) is enabled, it leads to receiver start interrupt.
- Receive event to indicate that a data word has been received:
If a new received word becomes available in the receive buffer, a receive event occurs. It is indicated by flag RXENDIF (USCI_PROTSTS [4]) and, if RXENDIEN (USCI_INTEN [4]) is enabled, it leads to receive interrupt.
- Data lost event to indicate a loss of the newest received data word:
If the data word available in register USCI_RXDAT (oldest data word from RX_BUF0 or RX_BUF1) has not been read out and the receive buffer is FULL, the new incoming data

will lose and this event occurs. It is indicated by flag RXOVIF (USCI_BUFSTS[3]) and, if RXOVIF (USCI_BUFCTL[14]) is enabled, it leads to a protocol interrupt.

The general event and interrupt structure is shown in Figure 6.15-12.

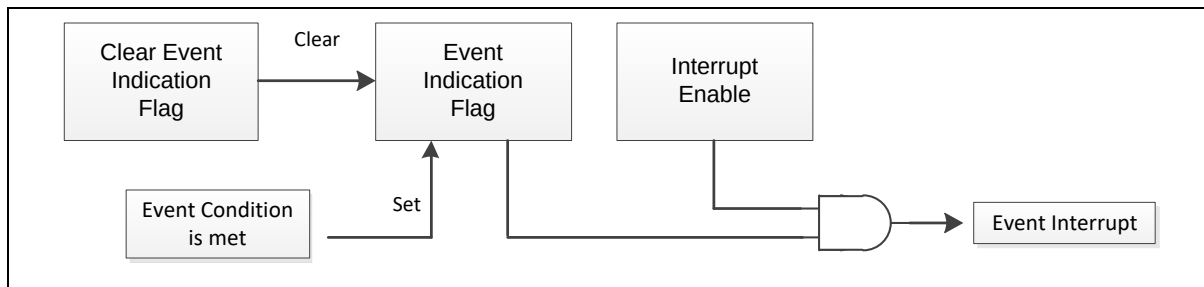


Figure 6.15-12 Event and Interrupt Structure

Each general interrupt enable can be set by RXENDIEN, RXSTIEN, TXENDIEN, and TXSTIEN of USCI_INTEN [4:1]. The events include receive end interrupt event, receive start interrupt event, transmit end interrupt event, and transmit start interrupt event. For protocol-specific interrupt, it is specified in each protocol interrupt enable register.

If a defined condition is met, an event is detected and an event indication flag becomes automatically set. The flag stays set until it is cleared by software. If enabled, an interrupt can be generated if an event is detected.

The registers, bits and bit fields indicate the data transfer events and control the general interrupts of a USCI are shown in Table 6.15-3.

Event	Indication Flag	Indication Cleared By	Interrupt Enabled By
Transmit start interrupt event	TXSTIF (USCI_PROTSTS [1])	It is cleared by software writes 1 to corresponding interrupt bit of USCI_PROTSTS.	TXSTIEN (USCI_INTEN [1])
Transmit end interrupt event	TXENDIF (USCI_PROTSTS [2])		TXENDIEN (USCI_INTEN [2])
Receive start interrupt event	RXSTIF (USCI_PROTSTS [3])		RXSTIEN (USCI_INTEN [3])
Receive end interrupt event	RXENDIF (USCI_PROTSTS [4])		RXENDIEN (USCI_INTEN [4])

Table 6.15-3 Data Transfer Events and Interrupt Handling

6.15.4.7 Protocol-specific Events and Interrupts

These events are related to protocol-specific actions that are described in the corresponding protocol chapters. The related indication flags are located in register USCI_PROTSTS. All events can be individually enabled for the generation of the common protocol interrupt.

Event	Indication Flag	Indication Cleared By	Interrupt Enabled By
Protocol-specific events in UART mode	USCI_PROTSTS [17:16] and USCI_PROTSTS [11:5]	It is cleared by software writes 1 to corresponding interrupt bit of USCI_PROTSTS.	USCI_PROTIEN[2:1]
Protocol-specific events in SPI mode	USCI_PROTSTS [9:8], USCI_PROTSTS [6:5]		USCI_PROTIEN [3:0]
Protocol-specific events in	USCI_PROTSTS [13:8],		USCI_PROTIEN [6:0]

I ² C mode	USCI_PROTSTS [5]		

Table 6.15-4 Protocol-specific Events and Interrupt Handling

6.15.4.8 Wake-up

The protocol-related wake-up functional information is located in the Wake-up Control Register (USCI_WKCTL) and in the Wake-up Status Register (USCI_WKSTS). These registers are shared between the available protocols. As a consequence, the meaning of the bit positions in these registers is different within the protocols.

6.15.4.9 PDMA

The USCI supports PDMA transfer function. When PDMAEN (USCI_PDMACTL [3]) is set to 1, the PDMA function is enabled.

When TXPDMAEN (USCI_PDMACTL [1]) is set to 1, the controller will issue request to PDMA controller to start the PDMA transmission process automatically.

When RXPDMAEN (USCI_PDMACTL [2]) is set to 1, the controller will start the PDMA reception process. USCI will issue request to PDMA controller automatically when there is data in the receive FIFO buffer.

In UART function, the requirement of RXPDMAEN will be cleared and hold if there is any error condition events including frame error, parity error or break detection. The user shall read out the current data and then the requirement of RXPDMAEN will send to the PDMA module in the next data.

6.16 USCI – UART Mode

6.16.1 Overview

The asynchronous serial channel UART covers the reception and the transmission of asynchronous data frames. It performs a serial-to-parallel conversion on data received from the peripheral, and a parallel-to-serial conversion on data transmitted from the controller. The receiver and transmitter being independent, frames can start at different points in time for transmission and reception.

The UART controller also provides auto flow control. There are two conditions to wake up the system.

6.16.2 Features

- Supports one transmit buffer and two receive buffers for data payload
- Supports hardware auto flow control function
- Supports programmable baud-rate generator
- Supports 9-bit Data Transfer (9-bit RS-485)
- Baud rate detection by built-in capture event of baud rate generator
- Supports PDMA capability
- Supports Wake-up function (Data and nCTS Wake-up Only)

6.16.3 Block Diagram

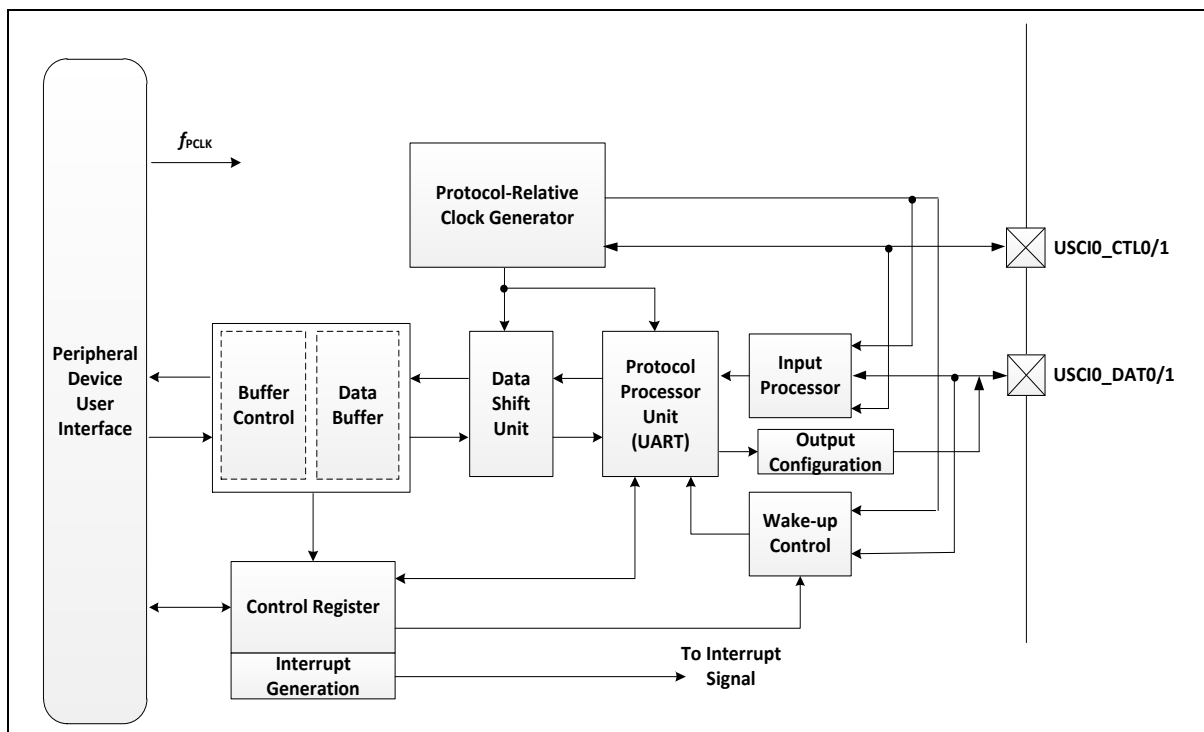


Figure 6.16-1 USCI-UART Mode Block Diagram

6.16.4 Basic Configuration

The basic configurations of USCI0_UART are as follows:

- Clock Source Configuration

- Enable USCI0 peripheral clock in USCI0CKEN (CLK_APBCLK1[8]).
- Enable USCI0_UART function in FUNMODE (UUART_CTL[2:0]=0x2).
- Reset Configuration
 - Reset USCI0 controller in USCI0RST (SYS_IPRST2[8]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
USCI0			
	USCI0_CTL0		
		PC.14	MFP5
	USCI0_CTL1		
		PB.15	MFP5
		PA.8	MFP6
	USCI0_DAT0		
		PB.13	MFP5
		PA.10	MFP6
	USCI0_DAT1		
		PB.14	MFP5
		PA.9	MFP6

6.16.5 Functional Description

6.16.5.1 USCI Common Functional Description

Please refer to USCI section for detailed information.

6.16.5.2 Signal Description

An UART connection is characterized by the use of a single connection line between a transmitter and a receiver. The receiver input signal (RXD) is handled by the input stage USCI0_DAT0 and the transmit output (TXD) signal is handled by the output stage of USCI0_DAT1.

For full-duplex communication, an independent communication line is needed for each transfer direction. Figure 6.16-2 shows an example with a point-to-point full-duplex connection between two communication partners UART module A and UART module B.

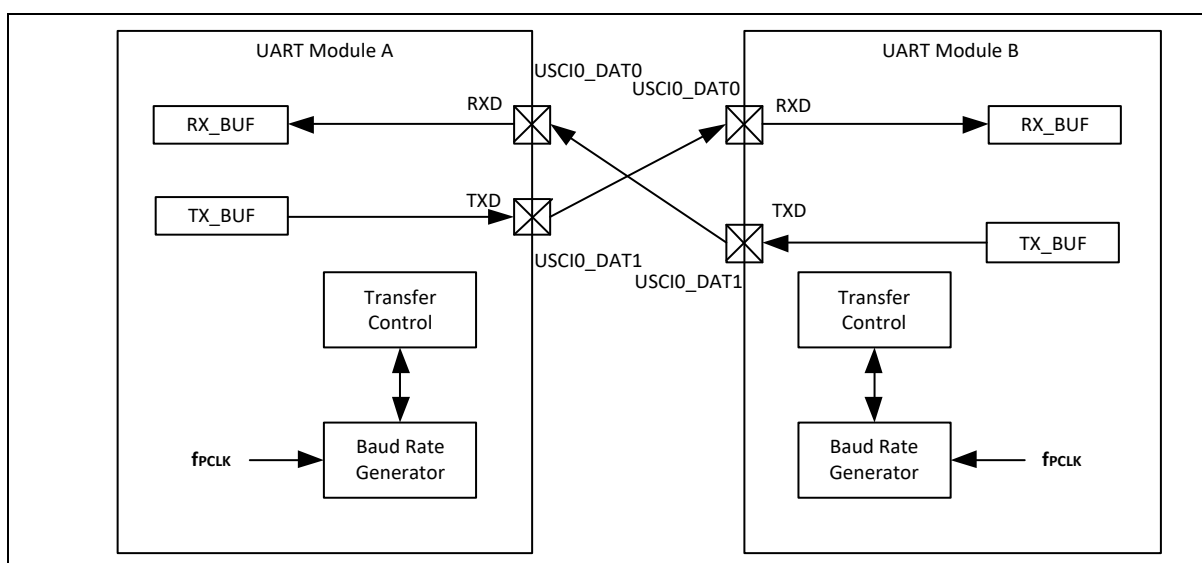


Figure 6.16-2 UART Signal Connection for Full-Duplex Communication

Input Signal

For UART protocol, the number of input signals is shown in Table 6.16-1. Each input signal is handled by an input processor for signal conditioning, such as signal inverse selection control, or a digital input filter. They can be classified according to their meaning for the protocols (see Table 6.16-1).

Selected Protocol		UART
Control Input	USC10_CTL0	nCTS
	USC10_CTL1	X
Data Input	USC10_DAT0	RX
	USC10_DAT1	X

Table 6.16-1 Input Signals for UART Protocol

Output Signals

For UART protocol, up to each protocol-related output signals are available. The number of actually used outputs depends on the selected protocol. They can be classified according to their meaning for the protocols.

Selected Protocol		UART
Control Output	USC10_CTL0	X
	USC10_CTL1	nRTS
Data Output	USC10_DAT0	X
	USC10_DAT1	TX

Table 6.16-2 Output Signals for UART Protocol

6.16.5.3 Frame Format

A standard UART frame is shown in Figure 6.16-3. It consists of:

- An idle time with the signal level 1.

- One start of frame bit (SOF) with the signal level 0.
- 6~13 bit data
- A parity bit (P), programmable for either even or odd parity. It is optionally possible to handle frames without parity bit.
- One or two stop bits with the signal level 1.

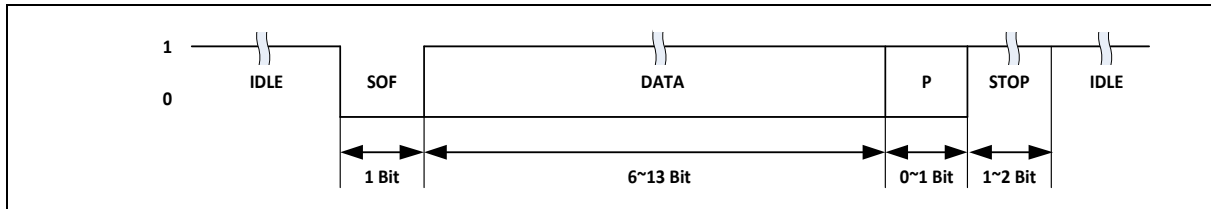


Figure 6.16-3 UART Standard Frame Format

The protocol specific bits (SOF, P, STOP) are automatically handled by the UART protocol state machine and do not appear in the data flow via the receive and transmit buffers.

Start Bit

The receiver input signal USCIO_DAT0 is checked for a falling edge. An SOF bit is detected when a falling edge occurs while the receiver is idle or after the sampling point of the last stop bit. To increase noise immunity, the SOF bit timing starts with the first falling edge that is detected. If the sampled bit value of the SOF is 1, the previous falling edge is considered to be due to noise and the receiver is considered to be idle again.

Data Field

The length of the data field (number of data bits) can be programmed by the bit field of DWIDTH (UART_LINECTL[11:8]). It can vary between 6 to 13 data bits.

Note: In UART protocol, the data transmission order is LSB first by setting LSB (UART_LINECTL[0]) to 1.

Parity Bit

The UART allows parity generation for transmission and parity check for reception on frame base. The type of parity can be selected by bit field PARITYEN (UART_PROTCTL[1]) and EVENPARITY (UART_PROTCTL[2]), common for transmission and reception (no parity, even or odd parity). If the parity handling is disabled, the UART frame does not contain any parity bit. For consistency reasons, all communication partners have to be programmed to the same parity mode.

After the last data bit of the data field, the transmitter automatically sends out its calculated parity bit if parity generation has been enabled. The receiver interprets this bit as received parity and compares it to its internally calculated one. The result of the parity check and frame check (STOP bit) are monitored in the protocol status registers (UART_PROTSTS). The register contains bits to monitor a protocol-related status and protocol-related error indication (FRMERR, PARITYERR).

Stop Bit

Each UART frame is completed by 1 or 2 of stop bits with the signal level 1 (same level as the idle level). The number of stop bits is programmable by bit STOPB (UART_PROTCTL[0]). A new start bit can be transferred directly after the last stop bit.

Transfer Status Indication

RXBUSY (UART_PROTSTS[10]) indicates the receiver status.

The receiver status can be monitored by RXBUSY bit. In this case, bit RXBUSY is set during a complete frame reception from the beginning of the start of frame bit to the end of the last stop bit.

6.16.5.4 Operating Mode

To operate the UART protocol, the following issues have to be considered:

Select UART Mode

The UART protocol can be selected by setting FUNMODEOE (UART_CTL[2:0]) to 0x2 and the UART protocol can be enabled by setting PROTEN (UART_PROTCTL [31]) to 1. Note that the FUNMODE must be set 0 before protocol changing and it is recommended to configure all parameters of the UART before UART protocol is enabled.

Pin Connections

The USCIO_DAT0 pin is used for UART receive data input signal (RX) in UART protocol. The property of input data signal can be configured in UART_DATIN0. It is suggested to set EDGEDET (UART_DATIN0[4:3]) as 10B for start bit detection.

The USCIO_DAT1 pin is used for UART transmit data output signal (TX) in UART protocol. The property of output data signal can be configured in UART_LINECTL.

The USCIO_CTL0 pin is used for UART clear to send signal (nCTS) in UART protocol. The property of input control signal can be configured in UART_CTLIN0.

The USCIO_CTL1 pin is used for UART request to send signal (nRTS) in UART protocol. The property of output control signal can be configured in UART_LINECTL.

Bit Timing Configuration

The desired baud rate setting has to be selected, comprising the baud rate generator and the bit timing.

Frame Format Configuration

The word length, the stop bit number, and the parity mode has to be set up according to the application requirements by programming UART_LINECTL and the UART_PROTCTL register. If required by the application, the data input and output signals can be inverted. The data transmission order is LSB first by setting LSB (UART_LINECTL[0]) to 1.

6.16.5.5 Bit Timing

In UART mode, each frame bit is divided into data sample time in order to provide granularity in the sub-bit range to adjust the sample point to the application requirements. The number of data sample time per bit is defined by bit fields DSCNT (UART_BRGEN[14:10]) and the length of a data sample time is given by PDSCNT (UART_BRGEN[9:8]).

In the example given in Figure 6.16-4, one bit time is composed of 16 data sample time DSCNT(UART_BRGEN[14:10]) = 15. It is not recommended to program less and equal than 4 data sample time per bit time.

The position of the sampling point for the bit value is fixed in 1/2 samples time. It is possible to sample the bit value to take the average of samples.

The bit timing setup (number of data sample time) is common for the transmitter and the receiver because they use the same hardware circuit.

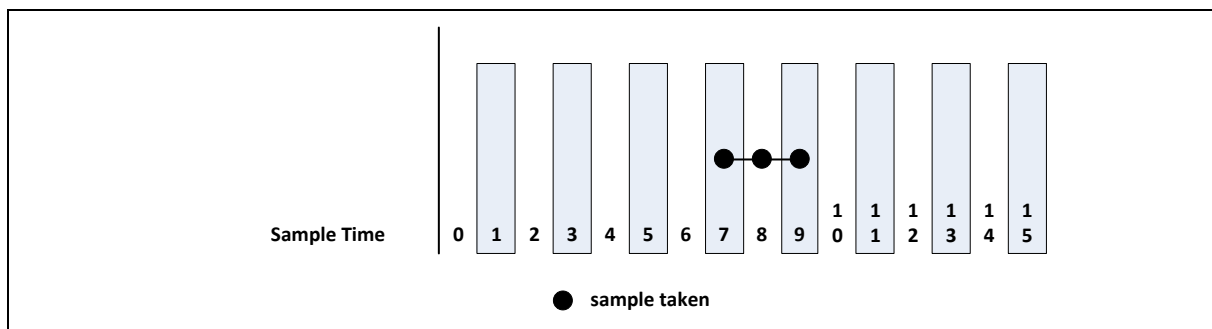


Figure 6.16-4 UART Bit Timing (Data Sample Time)

6.16.5.6 Baud Rate Generation

The baud rate f_{UART} in UART mode depends on the number of data sample time per bit time and their

timing. The baud rate setting should only be changed while the transmitter and the receiver are idle. The bits RCLKSEL, SPCLKSEL, PDSCNT, and DSCNT define the baud rate setting:

RCLKSEL (UART_BRGEN [0])

to define the input frequency f_{REF_CLK}

SPCLKSEL (UART_BRGEN[3:2])

to define the multiple source of the sample clock f_{SAMP_CLK}

PDSCNT (UART_BRGEN [9:8])

to define the length of a data sample time (division of f_{REF_CLK} by 1, 2, 3, or 4)

DSCNT (UART_BRGEN [14:10])

to define the number of data sample time per bit time

The standard setting is given by RCLKSEL = 0 ($f_{REF\_CLK} = f_{PCLK}$), PTCLKSEL = 0 ($f_{PROT\_CLK} = f_{REF\_CLK}$) and SPCLKSEL = 0 ($f_{SAMP\_CLK} = f_{DIV\_CLK}$). Under these conditions, the baud rate is given by:

$$f_{UART} = f_{REF\_CLK} \times \frac{1}{CLKDIV + 1} \times \frac{1}{PDSCNT + 1} \times \frac{1}{DSCNT + 1}$$

In order to generate slower frequencies, additional divide-by-2 stages can be selected by PTCLKSEL = 1 ($f_{PROT\_CLK} = f_{REF\_CLK2}$), leading to:

$$f_{UART} = \frac{f_{REF\_CLK}}{2} \times \frac{1}{CLKDIV + 1} \times \frac{1}{PDSCNT + 1} \times \frac{1}{DSCNT + 1}$$

If SPCLKSEL = 2 ($f_{SAMP\_CLK} = f_{SCLK}$), and RCLKSEL = 0 ($f_{REF\_CLK} = f_{PCLK}$), PTCLKSEL = 0 ($f_{PROT\_CLK} = f_{REF\_CLK}$). The baud rate is given by:

$$f_{UART} = f_{REF\_CLK} \times \frac{1}{CLKDIV + 1} \times \frac{1}{2} \times \frac{1}{PDSCNT + 1} \times \frac{1}{DSCNT + 1}$$

There is error tolerance for the UART baud rate after setting the baud rate parameter. Table 6.16-3 lists the baud rate setting examples and the relative error percentage for user to calculate his relative baud rate setting. The clock source is standard setting (SPCLKSEL=0, PTCLKSEL=0 and RCLKSEL=0).

HCLK Source	PCLK Source	Expect Baud Rate	CLKDIV (UART_BRGEN[25:16])	DSCNT (UART_BRGEN[14:10])	PDSCNT	Active Baud Rate	Error Percentage
12 MHz	HCLK	115200	0xC	0x7	0x0	115384	0.16%
12 MHz	HCLK	9600	0x7C	0x9	0x0	9600	0%
12 MHz	HCLK	2400	0x1F3	0x9	0x0	2400	0%

Table 6.16-3 Baud Rate Relationship

6.16.5.7 Auto Baud Rate Detection

The UART controller supports auto baud rate detection function. It is used to identify the input baud rate from the receiver signal (USC10_DAT0) and then revised the baud rate clock divider CLKDIV (UART_BRGEN[25:16]) after the baud rate function done to meet the detected baud rate information. According the section of Timing Measurement Counter, the timing measurement counter is used for time interval measurement of the input signal (USC10_DAT0) and the actual timer value is captured into bit field BRDETITV (UART_PROTCTL [24:16]) in each falling edge of the detected signal.

When the ABREN (UART_PROTCTL[6]) bit is enabled, the 0x55 data patterns is necessary for auto baud rate detection. The falling edge of input signal starts the baud rate counter and it loads the timing measurement counter value into the BRDETITV (UART_PROTCTL [24:16]) in the next falling edge. It is suggested to use the fDIV_CLK (TMCNTSRC (UART_BRGENC[5]) =1) as the counter source.

The CLKDIV (UART_BRGEN[25:16]) will be revised by BRDETITV (UART_PROTCTL [25:16]) after

the auto baud rate function done (the time of 4th falling edge of input signal). If the user want to receive the next successive frame correctly, it is better to set the value of CLKDIV (UUART_BRGEN[25:16]) and DSCNT (UUART_BRGEN[14:10]) as the same value (the value shall be among the rang of 0xF and 0x5 because the DSCNT is used to define the sample counter of each bit and the PDSCNT (UUART_BRGEN[9:8]) is 0x0).

During the auto baud rate detection, the ABRDETIF (UUART_PROTSTS[9]) and the BRDETITV (UUART_PROTCTL [24:16]) will be updated after each falling edge of input signal and the auto baud rate pattern, 0x55, won't be received into the receiver buffer after the frame done. The bit of ABREN will be cleared by hardware after the 4th falling edge of input signal is detected thus the user can read the status of ABREN to know the auto baud rate function is done or not.

If the CLKDIV and DSCNT are not set as the same value in calculation the auto baud rate function, the user shall calculate the proper average baud rate by the value of BRDETITV and CLKDIV after the auto baud rate function done.

If the baud rate of input signal is very slower and the bit time of timing measurement counter can't calculate the correct period of the input bit time, there is a ABERRSTS bit (UUART_PROTSTS[11]) to indicate the error information of the auto baud rate detection. At this time, the user shall revise the value of CLKDIV and require the Host device to send the 0x55 pattern again.

According the limitation of timing measurement counter, the maximum auto baud rate detection is 0x1FE for BRDETITV. The UART Auto Baud Rate Control is shown in Figure 6.16-5.

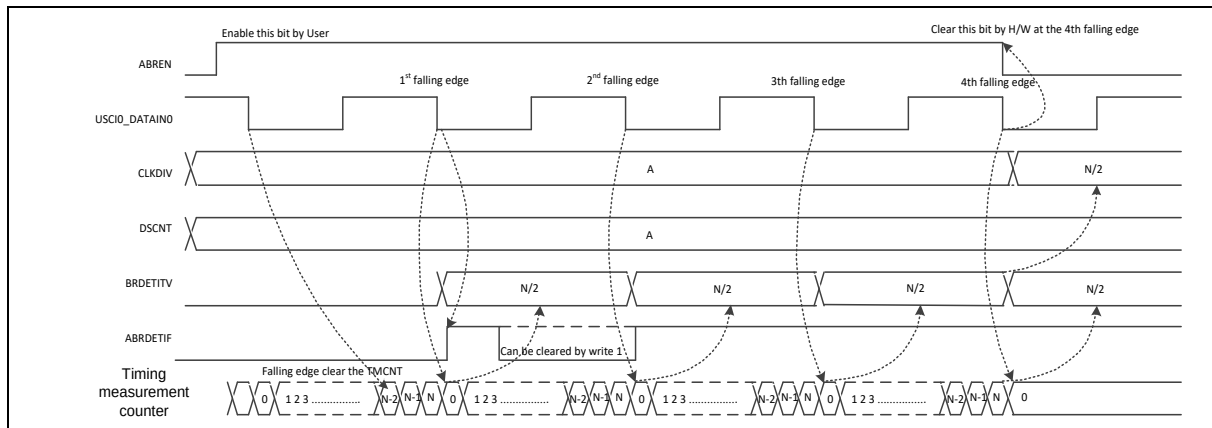


Figure 6.16-5 UART Auto Baud Rate Control

6.16.5.8 Auto Flow Control

The UART supports hardware auto-flow control that provides nRTS flow control by indicator RXFULL (UUART_BUFSTS[1]) on receiver buffer. When the buffer is full (RXFULL = 1), the nRTS is de-asserted.

The UART also provides nCTS flow control on transmitter. The nCTS is used to control the transmitted data is sent out when the nCTS is asserted.

6.16.5.9 RS-485 Support

The UART controller can play the role of the RS-485 master transmitter will identify an address character by setting the parity (9-th bit) to 1. For data characters, the parity is set to 0. Software can use the bit15 of each data to control the parity bit (PARITYEN (UUART_PROTCTL[1]) be set) when the STICKEN (UUART_PROTCTL[26]) is set. For example, if the STICKEN is set to 1 and data sequence are 0x8015, 0x8033, 0x0055, 0x0033 and 0x80AA the transmitted parity of data 0x15, 0x33, 0x55, 0x33 and 0xAA will be 1, 1, 0, 0 and 1.

The UART controller can also play as an RS-485 addressable slave, the protocol-related error of PARITYERR (UUART_PROTSTS[5]) can be acted as the address bit detection when the PARITYEN (UUART_PROTCTL[1]), EVENPARITY (UUART_PROTCTL[2]) and STICKEN (UUART_PROTCTL[26]) were set. If the PARITYERR was set, it means that the address bit in the

received bus is detected otherwise, the data is received into Buffer.

6.16.5.10 Wake-up Function

The USCI Controller in UART mode supports wake-up system function. The wake-up source includes incoming data and nCTS pin. Each wake-up source description is as follows:

- (a) Incoming data wake-up

When system is in power-down and both of the WKEN (UART_WKCTL [0]) and DATWKEN (UART_PROTCTL[9]) are set, the toggle of incoming data pin can wake-up the system. In order to receive the incoming data after the system wake-up, the WAKECNT (UART_PROTCTL[14:11]) shall be set. These bits field of WAKECNT (UART_PROTCTL[14:11]) indicate how many clock cycle selected by fPDS_CNT do the controller can get the 1st bit (start bit) when the device is woken up from Power-down mode. The incoming data wake-up is shown in Figure 6.16-6.

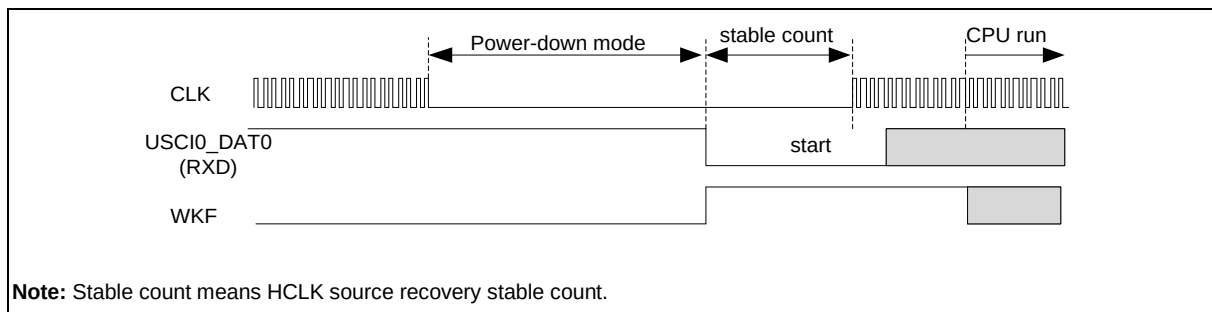


Figure 6.16-6 Incoming Data Wake-Up

- (b) nCTS pin wake-up

When system is in power-down and both of the WKEN (UART_WKCTL [0]) and CTSWKEN (UART_PROTCTL[10]) are set, the toggle of nCTS pin can wake-up the system. The nCTS wake-up is shown in Figure 6.16-7 and Figure 6.16-8.

Case 1(nCTS transition from low to high):

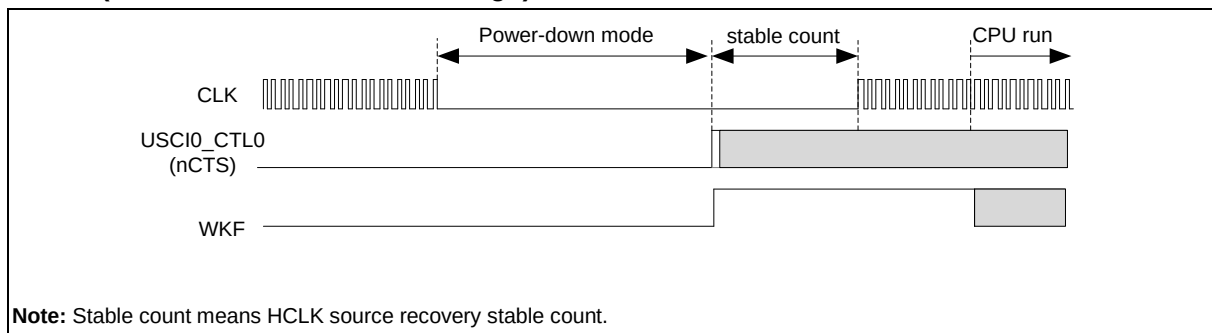


Figure 6.16-7 nCTS Wake-Up Case 1

Case 2 (nCTS transition from high to low):

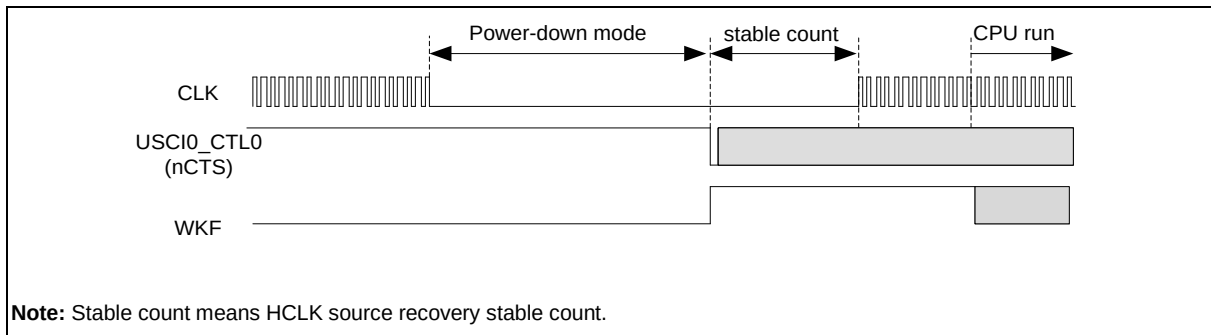


Figure 6.16-8 nCTS Wake-Up Case 2

6.16.5.11 Interrupt Events

The UART provided interrupt for protocol event and data transfer event. The description show below:

Protocol Interrupt Events

The following protocol-related events are generated in UART mode and can lead to a protocol interrupt.

Please note that the bits in register UUART_PROTSTS are not automatically cleared by hardware and have to be cleared by software in order to monitor new incoming events.

Receiver Line Status

The protocol-related error FRMERR (UUART_PROTSTS[6]) or PARITYERR (UUART_PROTSTS[5]) are two flags that are assigned to each received data word in the corresponding receiver buffer status registers.

In UART mode, the result of the parity check by the protocol-related error indication PARITYERR (0 = received parity bit equal to calculated parity value), and the result of frame check by the protocol-related error indication FRMERR (0 = received stop bit equal to the format value '1'). This information is elaborated for each data frame.

The break error flag BREAK (UUART_PROTSTS[7]) is assigned when the receive data is 0, the received parity and the stop bit are also 0.

The interrupt indicates that there are parity error, frame error or the break data detection in the BREAK, FRMERR, PARITYERR (UUART_PROTSTS[7:5]) bits.

Auto Baud Rate Detection

The auto baud rate interrupt, ABRDETIF (UUART_PROTSTS [9]), indicates that the timing measurement counter has getting 2-bit duration for auto baud rate capture function.

The auto baud rate detection function will be enabled in the first falling edge of receiver signal. The auto baud rate detection function is measurement after the next following falling is detected and it is finished when the frame transfer done. After the transfer done, the timing measurement counter value divided by twice is equal to the number of sample time per bit. The user can read the value of BRDETIV (UUART_PROTCTL[24:16]) and write into the baud rate generator register CLKDIV (UUART_BRGEN[25:16]).

Data Transfer Interrupt Handling

The data transfer interrupts indicate events related to UART frame handling.

Transmit Start Interrupt

Bit TXSTIF (UUART_PROTSTS [1]) is set after the start bit of a data word. In buffer mode, this is the earliest point in time when a new data word can be written to UUART_TXDAT.

Transmitter Finished

This interrupt indicates that the transmitter has completely finished all data in the buffer. Bit TXENDIF (UUART_PROTSTS [2]) becomes set at the end of the last stop bit.

Receiver Starts Interrupt

Bit RXSTIF (UART_PROTSTS [3]) is set after the sample point of the start bit.

Receiver Frame Finished

This interrupt indicates that the receiver has completely finished a frame. Bit RXENDIF (UART_PROTSTS [4]) becomes set at the end of the last receive bit.

6.16.5.12 Programming Example

The following steps are used to configure the UART protocol setting and the data transmission.

1. Set FUNMODE (UART_CTL[2:0]) to 0x2 to select UART protocol.
2. Write baud rate generator register UART_BRGEN to select desired baud rate.
 - Set SPCLKSEL (UART_BRGEN[3:2]), PTCLKSEL (UART_BRGEN[1]) and RCLKSEL (UART_BRGEN[0]) to select the clock source.
 - Configure CLKDIV (UART_BRGEN[25:16]), DSCNT (UART_BRGEN[14:10]) and PSSCNT (UART_BRGEN[9:8]) to determine the baud rate divider.
3. Write line control register UART_LINECTL and protocol control register UART_PROTCTL to configure the transmission data format and UART protocol setting.
 - Program data field length in DWIDTH (UART_LINECTL[11:8]).
 - Enable parity bit and determine the parity bit type by setting EVENPARITY (UART_PROTCTL[2]) and PARITYEN (UART_PROTCTL[1]).
 - Configure stop bit length by setting STOPB (UART_PROTCTL[0]).
 - Enable LSB (UART_LINECTL[0]) to select LSB first transmission for UART protocol.
 - Set EDGEDET (UART_DATIN0[4:3]) to 0x2 to select the detected edge as falling edge for receiver start bit detection.
4. Set PROTEN (UART_PROTCTL[31]) to 1 to enable UART protocol.
5. Transmit and receive data.
 - Write transmit data register UART_TXDAT to transmit data.
 - Wait until TXSTIF(UART_PROTSTS[1]) is set and then user can write the next data in UART_TXDAT.
 - When TXENDIF(UART_PROTSTS[2]) is set, the transmit buffer is empty and the stop bit of the last data has been transmitted.
 - If RXENDIF(UART_PROTSTS[4]) is set, the receiver has finished a data frame completely. User can get the data by reading receive data register UART_RXDAT.

6.16.6 Register Map

R: read only, W: write only, R/W: both read and write

Register	Offset	R/W	Description	Reset Value
USCI_UART Base Address: $UUARTn\_BA = 0x400D\_0000 + (0x1000 * n)$ $n = 0$				
UUART_CTL	UUARTn_BA+0x00	R/W	USCI Control Register	0x0000_0000
UUART_INTEN	UUARTn_BA+0x04	R/W	USCI Interrupt Enable Register	0x0000_0000
UUART_BRGEN	UUARTn_BA+0x08	R/W	USCI Baud Rate Generator Register	0x0000_3C00
UUART_DATIN0	UUARTn_BA+0x10	R/W	USCI Input Data Signal Configuration Register 0	0x0000_0000
UUART_CTLIN0	UUARTn_BA+0x20	R/W	USCI Input Control Signal Configuration Register 0	0x0000_0000
UUART_CLKIN	UUARTn_BA+0x28	R/W	USCI Input Clock Signal Configuration Register	0x0000_0000
UUART_LINECTL	UUARTn_BA+0x2C	R/W	USCI Line Control Register	0x0000_0000
UUART_TXDAT	UUARTn_BA+0x30	W	USCI Transmit Data Register	0x0000_0000
UUART_RXDAT	UUARTn_BA+0x34	R	USCI Receive Data Register	0x0000_0000
UUART_BUFCTL	UUARTn_BA+0x38	R/W	USCI Transmit/Receive Buffer Control Register	0x0000_0000
UUART_BUFSTS	UUARTn_BA+0x3C	R/W	USCI Transmit/Receive Buffer Status Register	0x0000_0101
UUART_PDMACTL	UUARTn_BA+0x40	R/W	USCI PDMA Control Register	0x0000_0000
UUART_WKCTL	UUARTn_BA+0x54	R/W	USCI Wake-up Control Register	0x0000_0000
UUART_WKSTS	UUARTn_BA+0x58	R/W	USCI Wake-up Status Register	0x0000_0000
UUART_PROTCTL	UUARTn_BA+0x5C	R/W	USCI Protocol Control Register	0x0000_0000
UUART_PROTIEN	UUARTn_BA+0x60	R/W	USCI Protocol Interrupt Enable Register	0x0000_0000
UUART_PROTSTS	UUARTn_BA+0x64	R/W	USCI Protocol Status Register	0x0000_0000

6.16.7 Register Description

USCI Control Register (UART_CTL)

Register	Offset	R/W	Description	Reset Value
UART_CTL	UARTn_BA+0x00	R/W	USCI Control Register	0x0000_0000

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

Bits	Description
[31:3]	Reserved Reserved.
[2:0]	Function Mode This bit field selects the protocol for this USCI controller. Selecting a protocol that is not available or a reserved combination disables the USCI. When switching between two protocols, the USCI has to be disabled before selecting a new protocol. Simultaneously, the USCI will be reset when user write 000 to FUNMODE. FUNMODE 000 = The USCI is disabled. All protocol related state machines are set to idle state. 001 = The SPI protocol is selected. 010 = The UART protocol is selected. 100 = The I ² C protocol is selected. Others = Reserved.

USCI Interrupt Enable Register (UART_INTEN)

Register	Offset	R/W	Description	Reset Value
UART_INTEN	UARTn_BA+0x04	R/W	USCI Interrupt Enable Register	0x0000_0000

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

Bits	Description	
[31:5]	Reserved	Reserved.
[4]	RXENDIEN	Receive End Interrupt Enable Bit This bit enables the interrupt generation in case of a receive finish event. 0 = The receive end interrupt Disabled. 1 = The receive end interrupt Enabled.
[3]	RXSTIEN	Receive Start Interrupt Enable Bit This bit enables the interrupt generation in case of a receive start event. 0 = The receive start interrupt Disabled. 1 = The receive start interrupt Enabled.
[2]	TXENDIEN	Transmit End Interrupt Enable Bit This bit enables the interrupt generation in case of a transmit finish event. 0 = The transmit finish interrupt Disabled. 1 = The transmit finish interrupt Enabled.
[1]	TXSTIEN	Transmit Start Interrupt Enable Bit This bit enables the interrupt generation in case of a transmit start event. 0 = The transmit start interrupt Disabled. 1 = The transmit start interrupt Enabled.
[0]	Reserved	Reserved.

USCI Baud Rate Generator Register (UART_BRGEN)

Register	Offset	R/W	Description	Reset Value
UART_BRGEN	UARTn_BA+0x08	R/W	USCI Baud Rate Generator Register	0x0000_3C00

31	30	29	28	27	26	25	24
Reserved						CLKDIV	
23	22	21	20	19	18	17	16
CLKDIV							
15	14	13	12	11	10	9	8
Reserved	DSCNT					PDSCNT	
7	6	5	4	3	2	1	0
Reserved		TMCNTSRC	TMCNTEN	SPCLKSEL		PTCLKSEL	RCLKSEL

Bits	Description
[31:26]	Reserved Reserved.
[25:16]	CLKDIV Clock Divider This bit field defines the ratio between the protocol clock frequency $f_{\text{PROT\_CLK}}$ and the clock divider frequency $f_{\text{DIV\_CLK}}$ ($f_{\text{DIV\_CLK}} = f_{\text{PROT\_CLK}} / (\text{CLKDIV} + 1)$). Note: In UART function, it can be updated by hardware in the 4 th falling edge of the input data 0x55 when the auto baud rate function (ABREN(UART_PROTCTL[6])) is enabled. The revised value is the average bit time between bit 5 and bit 6. The user can use revised CLKDIV and new BRDETIV (UART_PROTCTL[24:16]) to calculate the precise baud rate.
[15]	Reserved Reserved.
[14:10]	DSCNT Denominator for Sample Counter This bit field defines the divide ratio of the sample clock $f_{\text{SAMP\_CLK}}$. The divided frequency $f_{\text{DS\_CNT}} = f_{\text{PDS\_CNT}} / (\text{DSCNT} + 1)$. Note: The maximum value of DSCNT is 0xF on UART mode and suggest to set over 4 to confirm the receiver data is sampled in right value.
[9:8]	PDSCNT Pre-divider for Sample Counter This bit field defines the divide ratio of the clock division from sample clock $f_{\text{SAMP\_CLK}}$. The divided frequency $f_{\text{PDS\_CNT}} = f_{\text{SAMP\_CLK}} / (\text{PDSCNT} + 1)$.
[7:6]	Reserved Reserved.
[5]	TMCNTSRC Timing Measurement Counter Clock Source Selection 0 = Timing measurement counter with $f_{\text{PROT\_CLK}}$. 1 = Timing measurement counter with $f_{\text{DIV\_CLK}}$.
[4]	TMCNTEN Timing Measurement Counter Enable Bit This bit enables the 10-bit timing measurement counter. 0 = Timing measurement counter is Disabled. 1 = Timing measurement counter is Enabled.
[3:2]	SPCLKSEL Sample Clock Source Selection This bit field used for the clock source selection of a sample clock ($f_{\text{SAMP\_CLK}}$) for the protocol processor. 00 = $f_{\text{SAMP\_CLK}}$ is selected to $f_{\text{DIV\_CLK}}$.

		01 = $f_{\text{SAMP\_CLK}}$ is selected to $f_{\text{PROT\_CLK}}$. 10 = $f_{\text{SAMP\_CLK}}$ is selected to f_{SCLK}. 11 = $f_{\text{SAMP\_CLK}}$ is selected to $f_{\text{REF\_CLK}}$.
[1]	PTCLKSEL	Protocol Clock Source Selection This bit selects the source signal of protocol clock ($f_{\text{PROT\_CLK}}$). 0 = Reference clock $f_{\text{REF\_CLK}}$. 1 = $f_{\text{REF\_CLK}2}$ (its frequency is half of $f_{\text{REF\_CLK}}$).
[0]	RCLKSEL	Reference Clock Source Selection This bit selects the source signal of reference clock ($f_{\text{REF\_CLK}}$). 0 = Peripheral device clock f_{PCLK}. 1 = Reserved.

USCI Input Data Signal Configuration (UART_DATIN0)

Register	Offset	R/W	Description	Reset Value
UART_DATIN0	UARTn_BA+0x10	R/W	USCI Input Data Signal Configuration Register 0	0x0000_0000

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

Bits	Description
[31:5]	Reserved Reserved.
[4:3]	EDGEDET Input Signal Edge Detection Mode This bit field selects which edge activates the trigger event of input data signal. 00 = The trigger event activation is disabled. 01 = A rising edge activates the trigger event of input data signal. 10 = A falling edge activates the trigger event of input data signal. 11 = Both edges activate the trigger event of input data signal. Note: In UART function mode, it is suggested to set this bit field as 0x2.
[2]	ININV Input Signal Inverse Selection This bit defines the inverter enable of the input asynchronous signal. 0 = The un-synchronized input signal will not be inverted. 1 = The un-synchronized input signal will be inverted.
[1]	Reserved Reserved.
[0]	SYNCSEL Input Signal Synchronization Selection This bit selects if the un-synchronized input signal (with optionally inverted) or the synchronized (and optionally filtered) signal can be used as input for the data shift unit. 0 = The un-synchronized signal can be taken as input for the data shift unit. 1 = The synchronized signal can be taken as input for the data shift unit.

USCI Input Control Signal Configuration (UART_CTLIN0)

Register	Offset	R/W	Description	Reset Value
UART_CTLIN0	UARTn_BA+0x20	R/W	USCI Input Control Signal Configuration Register 0	0x0000_0000

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

Bits	Description	
[31:3]	Reserved	Reserved.
[2]	ININV	Input Signal Inverse Selection This bit defines the inverter enable of the input asynchronous signal. 0 = The un-synchronized input signal will not be inverted. 1 = The un-synchronized input signal will be inverted.
[1]	Reserved	Reserved.
[0]	SYNCSEL	Input Synchronization Signal Selection This bit selects if the un-synchronized input signal (with optionally inverted) or the synchronized (and optionally filtered) signal can be used as input for the data shift unit. 0 = The un-synchronized signal can be taken as input for the data shift unit. 1 = The synchronized signal can be taken as input for the data shift unit.

USCI Input Clock Signal Configuration Register (UART_CLKIN)

Register	Offset	R/W	Description	Reset Value
UART_CLKIN	UARTn_BA+0x28	R/W	USCI Input Clock Signal Configuration Register	0x0000_0000

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

Bits	Description
[31:1]	Reserved Reserved.
[0]	SYNCSEL Input Synchronization Signal Selection This bit selects if the un-synchronized input signal or the synchronized (and optionally filtered) signal can be used as input for the data shift unit. 0 = The un-synchronized signal can be taken as input for the data shift unit. 1 = The synchronized signal can be taken as input for the data shift unit.

USCI Line Control Register (UART_LINECTL)

Register	Offset	R/W	Description	Reset Value
UART_LINECTL	UARTn_BA+0x2C	R/W	USCI Line Control Register	0x0000_0000

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

Bits	Description
[31:12]	Reserved Reserved.
[11:8]	DWIDTH Word Length of Transmission This bit defines the data word length (amount of bits) for reception and transmission. The data word is always right-aligned in the data buffer. USCI support word length from 4 to 16 bits. 0000 = The data word contains 16 bits located at bit positions [15:0]. 0001 = Reserved. 0010 = Reserved. 0011 = Reserved. 0100 = The data word contains 4 bits located at bit positions [3:0]. 0101 = The data word contains 5 bits located at bit positions [4:0]. 0110 = The data word contains 6 bits located at bit positions [5:0]. 0111 = The data word contains 7 bits located at bit positions [6:0]. 1000 = The data word contains 8 bits located at bit positions [7:0]. 1001 = The data word contains 9 bits located at bit positions [8:0]. 1010 = The data word contains 10 bits located at bit positions [9:0]. 1011 = The data word contains 11 bits located at bit positions [10:0]. 1100 = The data word contains 12 bits located at bit positions [11:0]. 1101 = The data word contains 13 bits located at bit positions [12:0]. 1110 = The data word contains 14 bits located at bit positions [13:0]. 1111 = The data word contains 15 bits located at bit positions [14:0]. Note: In UART protocol, the length can be configured as 6~13 bits.
[7]	CTLOINV Control Signal Output Inverse Selection This bit defines the relation between the internal control signal and the output control signal. 0 = No effect. 1 = The control signal will be inverted before its output. Note: In UART protocol, the control signal means nRTS signal.
[6]	Reserved Reserved.
[5]	DATOINV Data Output Inverse Selection

		This bit defines the relation between the internal shift data value and the output data signal of USCIO_DAT1 pin. 0 = The value of USCIO_DAT1 is equal to the data shift register. 1 = The value of USCIO_DAT1 is the inversion of data shift register.
[4:1]	Reserved	Reserved.
[0]	LSB	LSB First Transmission Selection 0 = The MSB, which bit of transmit/receive data buffer depends on the setting of DWIDTH, is transmitted/received first. 1 = The LSB, the bit 0 of data buffer, will be transmitted/received first.

USCI Transmit Data Register (UART_TXDAT)

Register	Offset	R/W	Description	Reset Value
UART_TXDAT	UARTn_BA+0x30	W	USCI Transmit Data Register	0x0000_0000

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

Bits	Description	
[31:16]	Reserved	Reserved.
[15:0]	TXDAT	Transmit Data Software can use this bit field to write 16-bit transmit data for transmission.

USCI Receive Data Register (UART_RXDAT)

Register	Offset	R/W	Description	Reset Value
UART_RXDAT	UARTn_BA+0x34	R	USCI Receive Data Register	0x0000_0000

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

Bits	Description
[31:16]	Reserved Reserved.
[15:0]	RXDAT Received Data This bit field monitors the received data which stored in receive data buffer. Note: RXDAT[15:13] indicate the same frame status of BREAK, FRMERR and PARITYERR (UART_PROTSTS[7:5]).

USCI Transmitter/Receive Buffer Control Register (UART_BUFCTL)

Register	Offset	R/W	Description	Reset Value
UART_BUFCTL	UARTn_BA+0x38	R/W	USCI Transmit/Receive Buffer Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved						RXRST	TXRST
15	14	13	12	11	10	9	8
RXCLR	RXOVLEN	Reserved					
7	6	5	4	3	2	1	0
TXCLR	Reserved						

Bits	Description
[31:18]	Reserved Reserved.
[17]	RXRST Receive Reset 0 = No effect. 1 = Reset the receive-related counters, state machine, and the content of receive shift register and data buffer. Note 1: It is cleared automatically after one PCLK cycle. Note 2: It is suggested to check the RXBUSY (UART_PROTSTS[10]) before this bit will be set to 1.
[16]	TXRST Transmit Reset 0 = No effect. 1 = Reset the transmit-related counters, state machine, and the content of transmit shift register and data buffer. Note: It is cleared automatically after one PCLK cycle.
[15]	RXCLR Clear Receive Buffer 0 = No effect. 1 = The receive buffer is cleared (filling level is cleared and output pointer is set to input pointer value). Should only be used while the buffer is not taking part in data traffic. Note: It is cleared automatically after one PCLK cycle.
[14]	RXOVLEN Receive Buffer Overrun Error Interrupt Enable Bit 0 = Receive overrun interrupt Disabled. 1 = Receive overrun interrupt Enabled.
[13:8]	Reserved Reserved.
[7]	TXCLR Clear Transmit Buffer 0 = No effect. 1 = The transmit buffer is cleared (filling level is cleared and output pointer is set to input pointer value). Should only be used while the buffer is not taking part in data traffic. Note: It is cleared automatically after one PCLK cycle.
[6:0]	Reserved Reserved.

USCI Transmit/Receive Buffer Status Register (UART_BUFSTS)

Register	Offset	R/W	Description	Reset Value
UART_BUFSTS	UARTn_BA+0x3C	R/W	USCI Transmit/Receive Buffer Status Register	0x0000_0101

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved						TXFULL	TXEMPTY
7	6	5	4	3	2	1	0
Reserved				RXOVIF	Reserved	RXFULL	RXEMPTY

Bits	Description
[31:10]	Reserved Reserved.
[9]	TXFULL Transmit Buffer Full Indicator (Read Only) 0 = Transmit buffer is not full. 1 = Transmit buffer is full.
[8]	TXEMPTY Transmit Buffer Empty Indicator (Read Only) 0 = Transmit buffer is not empty. 1 = Transmit buffer is empty.
[7:4]	Reserved Reserved.
[3]	RXOVIF Receive Buffer Over-run Error Interrupt Status This bit indicates that a receive buffer overrun error event has been detected. If RXOVIEN (UART_BUFCTL[14]) is enabled, the corresponding interrupt request is activated. 0 = A receive buffer overrun error event has not been detected. 1 = A receive buffer overrun error event has been detected. Note: It is cleared by software writing 1 into this bit.
[2]	Reserved Reserved.
[1]	RXFULL Receive Buffer Full Indicator (Read Only) 0 = Receive buffer is not full. 1 = Receive buffer is full.
[0]	RXEMPTY Receive Buffer Empty Indicator (Read Only) 0 = Receive buffer is not empty. 1 = Receive buffer is empty.

USCI PDMA Control Register (UART_PDMACTL)

Register	Offset	R/W	Description	Reset Value
UART_PDMACTL	UARTn_BA+0x40	R/W	USCI PDMA Control Register	0x0000_0000

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

Bits	Description
[31:4]	Reserved Reserved.
[3]	PDMAEN PDMA Mode Enable Bit 0 = PDMA function Disabled. 1 = PDMA function Enabled.
[2]	RXPDMAEN PDMA Receive Channel Available 0 = Receive PDMA function Disabled. 1 = Receive PDMA function Enabled.
[1]	TXPDMAEN PDMA Transmit Channel Available 0 = Transmit PDMA function Disabled. 1 = Transmit PDMA function Enabled.
[0]	PDMARST PDMA Reset 0 = No effect. 1 = Reset the USCI's PDMA control logic. This bit will be cleared to 0 automatically.

USCI Wake-up Control Register (UART_WKCTL)

Register	Offset	R/W	Description	Reset Value
UART_WKCTL	UARTn_BA+0x54	R/W	USCI Wake-up Control Register	0x0000_0000

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

Bits	Description
[31:3]	Reserved Reserved.
[2]	PDBOPT Power Down Blocking Option 0 = If user attempts to enter Power-down mode by executing WFI while the protocol is in transferring, MCU will stop the transfer and enter Power-down mode immediately. 1 = If user attempts to enter Power-down mode by executing WFI while the protocol is in transferring, the on-going transfer will not be stopped and MCU will enter idle mode immediately.
[1]	Reserved Reserved.
[0]	WKEN Wake-up Enable Bit 0 = Wake-up function Disabled. 1 = Wake-up function Enabled.

USCI Wake-up Status Register (UART WKSTS)

Register	Offset	R/W	Description	Reset Value
UART_WKSTS	UARTn_BA+0x58	R/W	USCI Wake-up Status Register	0x0000_0000

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

Bits	Description
[31:1]	Reserved Reserved.
[0]	WKF Wake-up Flag When chip is woken up from Power-down mode, this bit is set to 1. Software can write 1 to clear this bit.

USCI Protocol Control Register – UART (UUART_PROTCTL)

Register	Offset	R/W	Description	Reset Value
UUART_PROTCTL	UUARTn_BA+0x5C	R/W	USCI Protocol Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PROTEN	DGE	BCEN	Reserved		STICKEN	Reserved	BRDETITV
23	22	21	20	19	18	17	16
BRDETITV							
15	14	13	12	11	10	9	8
Reserved	WAKECNT				CTSWKEN	DATWKEN	Reserved
7	6	5	4	3	2	1	0
Reserved	ABREN	RTSAUDIREN	CTSAUTOEN	RTSAUTOEN	EVENPARITY	PARITYEN	STOPB

Bits	Description
[31]	PROTEN UART Protocol Enable Bit 0 = UART Protocol Disabled. 1 = UART Protocol Enabled.
[30]	DGE Deglitch Enable Bit 0 = Deglitch Disabled. 1 = Deglitch Enabled. Note 1: When this bit is set to logic 1, any pulse width less than about 150 ns will be considered a glitch and will be removed in the serial data input (RX). This bit acts only on RX line and has no effect on the transmitter logic. Note 2: It is recommended to set this bit only when operating at baud rate under 2.5 Mbps.
[29]	BCEN Transmit Break Control Enable Bit 0 = Transmit Break Control Disabled. 1 = Transmit Break Control Enabled. Note: When this bit is set to logic 1, the serial data output (TX) is forced to the Spacing State (logic 0). This bit acts only on TX line and has no effect on the receiver logic.
[28:27]	Reserved.
[26]	STICKEN Stick Parity Enable Bit 0 = Stick parity Disabled. 1 = Stick parity Enabled. Note: Refer to RS-485 Support section for detailed information.
[25]	Reserved.
[24:16]	BRDETITV Baud Rate Detection Interval This bit fields indicate how many clock cycle selected by TMCNTSRC (UUART_BRGEN [5]) does the slave calculates the baud rate in one bits. The order of the bus shall be 1 and 0 step by step (e.g. the input data pattern shall be 0x55). The user can read the value to know the current input baud rate of the bus whenever the ABRDETIF (UUART_PROTCTL[9]) is set. Note: This bit can be cleared to 0 by software writing '0' to the BRDETITV.

[15]	Reserved	Reserved.
[14:11]	WAKECNT	Wake-up Counter These bits field indicate how many clock cycle selected by f_{PDS_CNT} do the slave can get the 1 st bit (start bit) when the device is woken up from Power-down mode.
[10]	CTSWKEN	nCTS Wake-up Mode Enable Bit 0 = nCTS wake-up mode Disabled. 1 = nCTS wake-up mode Enabled.
[9]	DATWKEN	Data Wake-up Mode Enable Bit 0 = Data wake-up mode Disabled. 1 = Data wake-up mode Enabled.
[8:7]	Reserved	Reserved.
[6]	ABREN	Auto-baud Rate Detect Enable Bit 0 = Auto-baud rate detect function Disabled. 1 = Auto-baud rate detect function Enabled. Note: When the auto - baud rate detect operation finishes, hardware will clear this bit. The associated interrupt ABRDETIF (UART_PROTSTS[9]) will be generated (If ARBIEN (UART_PROTIEN [1]) is enabled).
[5]	RTSAUDIREN	nRTS Auto Direction Enable Bit When nRTS auto direction is enabled, if the transmitted bytes in the TX buffer is empty, the nRTS signal is inactive automatically. 0 = nRTS auto direction control Disabled. 1 = nRTS auto direction control Enabled. Note 1: This bit is used for nRTS auto direction control for RS485. Note 2: This bit has effect only when the RTSAUTOEN is not set.
[4]	CTSAUTOEN	nCTS Auto-flow Control Enable Bit When nCTS auto-flow is enabled, the UART will send data to external device when nCTS input assert (UART will not send data to device if nCTS input is dis-asserted). 0 = nCTS auto-flow control Disabled. 1 = nCTS auto-flow control Enabled.
[3]	RTSAUTOEN	nRTS Auto-flow Control Enable Bit When nRTS auto-flow is enabled, if the receiver buffer is full (RXFULL (UART_BUFSTS[1] =1), the UART will de-assert nRTS signal. 0 = nRTS auto-flow control Disabled. 1 = nRTS auto-flow control Enabled. Note: This bit has effect only when the RTSAUDIREN is not set.
[2]	EVENPARITY	Even Parity Enable Bit 0 = Odd number of logic 1's is transmitted and checked in each word. 1 = Even number of logic 1's is transmitted and checked in each word. Note: This bit has effect only when PARITYEN is set.
[1]	PARITYEN	Parity Enable Bit This bit defines the parity bit is enabled in an UART frame. 0 = The parity bit Disabled. 1 = The parity bit Enabled.
[0]	STOPB	Stop Bits This bit defines the number of stop bits in an UART frame. 0 = The number of stop bits is 1.

		1 = The number of stop bits is 2.

USCI Protocol Interrupt Enable Register – UART (UART_PROTIEN)

Register	Offset	R/W	Description	Reset Value
UART_PROTIEN	UARTn_BA+0x60	R/W	USCI Protocol Interrupt Enable Register	0x0000_0000

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

Bits	Description
[31:3]	Reserved Reserved.
[2]	RLSIEN Receive Line Status Interrupt Enable Bit 0 = Receive line status interrupt Disabled. 1 = Receive line status interrupt Enabled. Note: UART_PROTSTS[7:5] indicates the current interrupt event for receive line status interrupt.
[1]	ABRIEN Auto-baud Rate Interrupt Enable Bit 0 = Auto-baud rate interrupt Disabled. 1 = Auto-baud rate interrupt Enabled.
[0]	Reserved Reserved.

USCI Protocol Status Register – UART (UUART_PROTSTS)

Register	Offset	R/W	Description	Reset Value
UUART_PROTSTS	UUARTn_BA+0x64	R/W	USCI Protocol Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved						CTSLV	CTSSYNCLV
15	14	13	12	11	10	9	8
Reserved				ABERRSTS	RXBUSY	ABRDETIF	Reserved
7	6	5	4	3	2	1	0
BREAK	FRMERR	PARITYERR	RXENDIF	RXSTIF	TXENDIF	TXSTIF	Reserved

Bits	Description
[31:18]	Reserved Reserved.
[17]	CTSLV nCTS Pin Status (Read Only) This bit used to monitor the current status of nCTS pin input. 0 = nCTS pin input is low level voltage logic state. 1 = nCTS pin input is high level voltage logic state.
[16]	CTSSYNCLV nCTS Synchronized Level Status (Read Only) This bit used to indicate the current status of the internal synchronized nCTS signal. 0 = The internal synchronized nCTS is low. 1 = The internal synchronized nCTS is high.
[15:12]	Reserved Reserved.
[11]	ABERRSTS Auto-baud Rate Error Status This bit is set when auto-baud rate detection counter overrun. When the auto-baud rate counter overrun, the user shall revise the CLKDIV (UUART_BRGEN[25:16]) value and enable ABREN (UUART_PROTCTL[6]) to detect the correct baud rate again. 0 = Auto-baud rate detect counter is not overrun. 1 = Auto-baud rate detect counter is overrun. Note 1: This bit is set at the same time of ABRDETIF. Note 2: This bit can be cleared by writing “1” to ABRDETIF or ABERRSTS.
[10]	RXBUSY RX Bus Status Flag (Read Only) This bit indicates the busy status of the receiver. 0 = The receiver is Idle. 1 = The receiver is BUSY.
[9]	ABRDETIF Auto-baud Rate Interrupt Flag This bit is set when auto-baud rate detection is done among the falling edge of the input data. If the ABRIEN (UUART_PROTCTL[6]) is set, the auto-baud rate interrupt will be generated. This bit can be set 4 times when the input data pattern is 0x55 and it is cleared before the next falling edge of the input bus.

		0 = Auto-baud rate detect function is not done. 1 = One Bit auto-baud rate detect function is done. Note: This bit can be cleared by writing “1” to it.
[8]	Reserved	Reserved.
[7]	BREAK	Break Flag This bit is set to logic 1 whenever the received data input (RX) is held in the “spacing state” (logic 0) for longer than a full word transmission time (that is, the total time of “start bit” + data bits + parity + stop bits). 0 = No Break is generated. 1 = Break is generated in the receiver bus. Note: This bit can be cleared by writing “1” among the BREAK, FRMERR and PARITYERR bits.
[6]	FRMERR	Framing Error Flag This bit is set to logic 1 whenever the received character does not have a valid “stop bit” (that is, the stop bit following the last data bit or parity bit is detected as logic 0). 0 = No framing error is generated. 1 = Framing error is generated. Note: This bit can be cleared by writing “1” among the BREAK, FRMERR and PARITYERR bits.
[5]	PARITYERR	Parity Error Flag This bit is set to logic 1 whenever the received character does not have a valid “parity bit”. 0 = No parity error is generated. 1 = Parity error is generated. Note: This bit can be cleared by writing “1” among the BREAK, FRMERR and PARITYERR bits.
[4]	RXENDIF	Receive End Interrupt Flag 0 = A receive finish interrupt status has not occurred. 1 = A receive finish interrupt status has occurred. Note: It is cleared by software writing 1 into this bit.
[3]	RXSTIF	Receive Start Interrupt Flag 0 = A receive start interrupt status has not occurred. 1 = A receive start interrupt status has occurred. Note: It is cleared by software writing 1 into this bit.
[2]	TXENDIF	Transmit End Interrupt Flag 0 = A transmit end interrupt status has not occurred. 1 = A transmit end interrupt status has occurred. Note: It is cleared by software writing 1 into this bit.
[1]	TXSTIF	Transmit Start Interrupt Flag 0 = A transmit start interrupt status has not occurred. 1 = A transmit start interrupt status has occurred. Note 1: It is cleared by software writing one into this bit. Note 2: Used for user to load next transmit data when there is no data in transmit buffer.
[0]	Reserved	Reserved.

6.17 USCI - SPI Mode

6.17.1 Overview

The SPI protocol of USCI controller applies to synchronous serial data communication and allows full duplex transfer. It supports both master and Slave operation mode with the 4-wire bi-direction interface. SPI mode of USCI controller performs a serial-to-parallel conversion on data received from a peripheral device, and a parallel-to-serial conversion on data transmitted to a peripheral device. The SPI mode is selected by FUNMODE (USPI_CTL[2:0]) = 0x1

This SPI protocol can operate as master or Slave mode by setting the SLAVE (USPI_PROTCTL[0]) to communicate with the off-chip SPI Slave or master device. The application block diagrams in master and Slave mode are shown below.

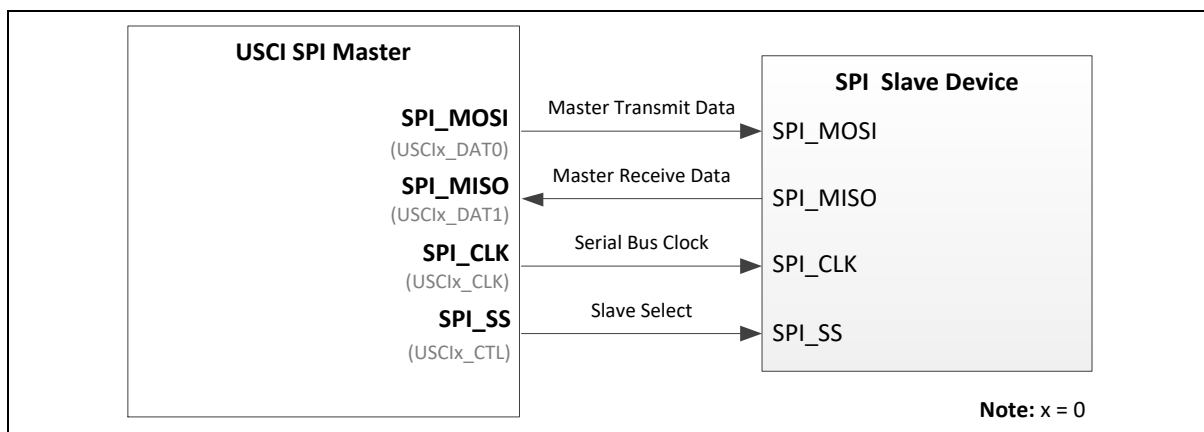


Figure 6.17-1 SPI Master Mode Application Block Diagram

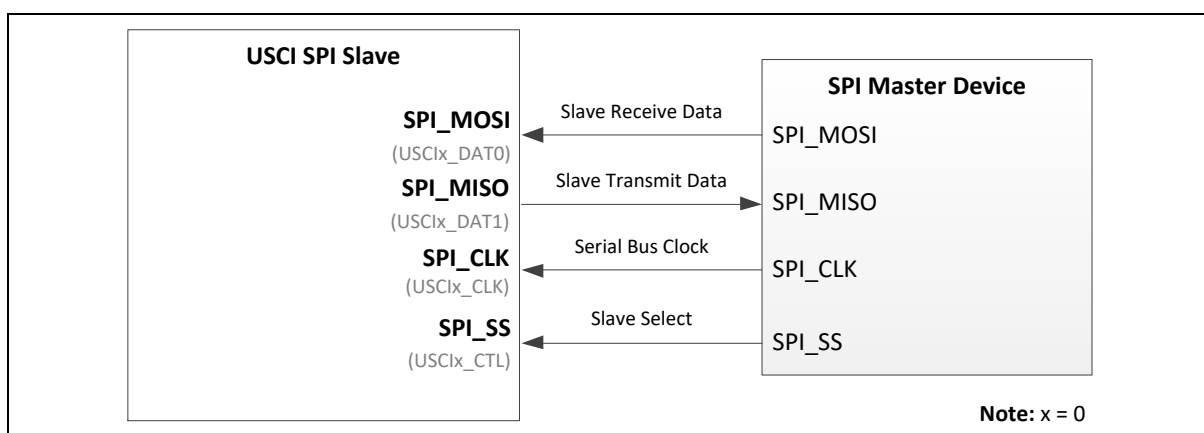


Figure 6.17-2 SPI Slave Mode Application Block Diagram

6.17.2 Features

- Configurable bit length of a transfer word from 4 to 16-bit
- Supports one transmit buffer and two receive buffers for data payload
- Supports MSB first or LSB first transfer sequence
- Supports Word Suspend function
- Supports PDMA transfer

- Supports 3-wire, no slave select signal, bi-direction interface
- Supports wake-up function by slave select signal in Slave mode
- Supports one data channel half-duplex transfer

6.17.3 Block Diagram

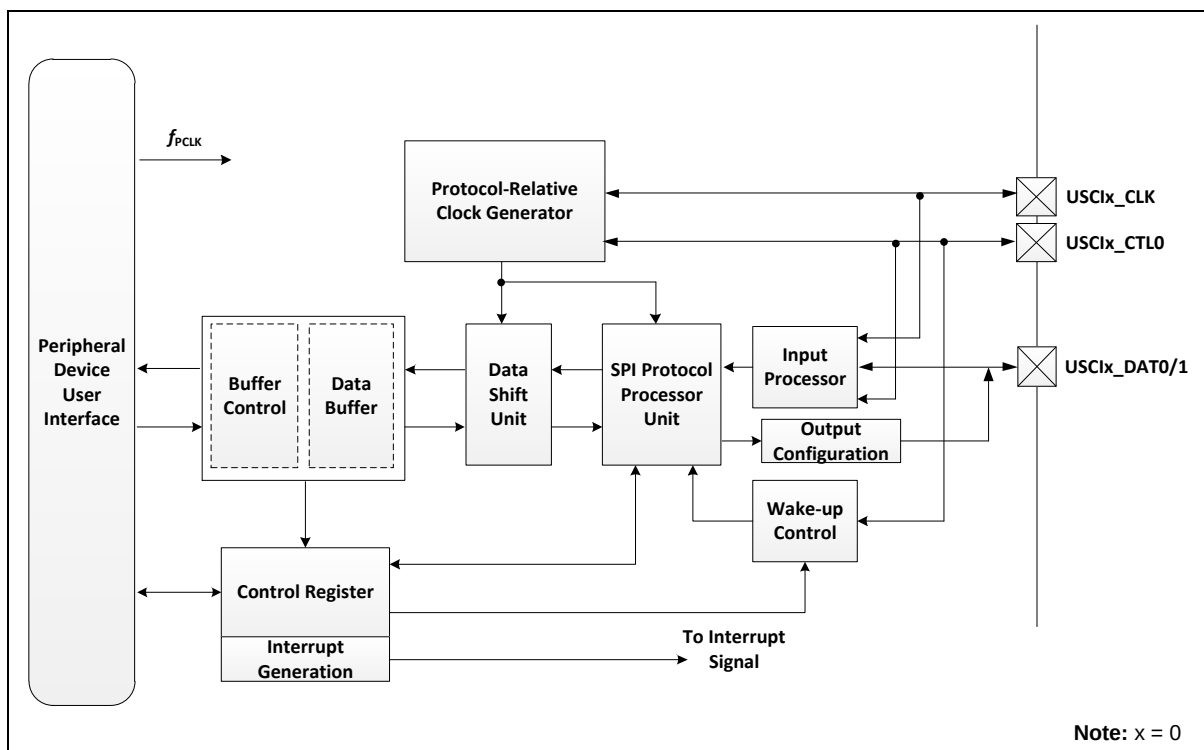


Figure 6.17-3 USCI SPI Mode Block Diagram

6.17.4 Basic Configuration

6.17.4.1 USCI0 SPI Basic Configurations

- Clock Source Configuration
 - Enable USCI0 peripheral clock in USCI0CKEN (CLK_APBCLK1[8]).
 - Enable USCI0_SPI function USPI_CTL[2:0] register, USPI_CTL[2:0]=0x1.
- Reset Configuration
 - Reset USCI0 controller in USCI0RST (SYS_IPRST2[8]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
USCI0	USCI0_CLK	PB.12	MFP5
		PA.11	MFP6
	USCI0_CTL0	PC.14	MFP5
	USCI0_DAT0	PB.13	MFP5
		PA.10	MFP6

	USCI0_DAT1	PB.14	MFP5
		PA.9	MFP6

6.17.5 Functional Description

6.17.5.1 USPI Common Functional Description

Please refer to USCI section for detailed information.

6.17.5.2 Signal Description

A device operating in Master mode controls the start and end of a data transfer, as well as the generation of the SPI bus clock and slave select signal. The slave select signal indicates the start and the end of a data transfer, and the master device can use it to enable the transmitting or receiving operations of Slave device. Slave device receives the SPI bus clock and optionally a slave select signal for data transaction. The signals for SPI communication are shown in Table 6.17-1.

SPI Mode	Receive Data	Transmit Data	Serial Bus Clock	Slave Select
Full-duplex SPI Master	SPI_MISO (USC1x_DAT1)	SPI_MOSI (USC1x_DAT0)	SPI_CLK (USC1x_CLK)	SPI_SS (USC1x_CTL0)
Full-duplex SPI Slave	SPI_MOSI (USC1x_DAT0)	SPI_MISO (USC1x_DAT1)	SPI_CLK (USC1x_CLK)	SPI_SS (USC1x_CTL0)
Half-duplex SPI Master/Slave	SPI_MOSI (USC1x_DAT0)	SPI_MOSI (USC1x_DAT0)	SPI_CLK (USC1x_CLK)	SPI_SS (USC1x_CTL0)

Table 6.17-1 Signals for SPI communication

● SPI Communication Signals

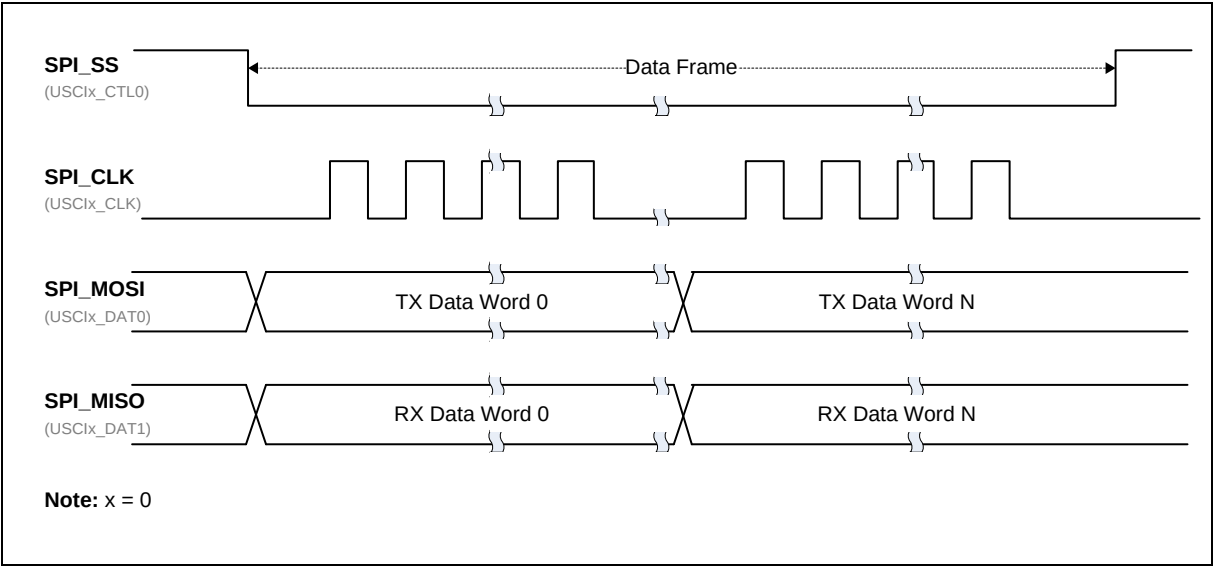


Figure 6.17-4 4-Wire Full-Duplex SPI Communication Signals (Master Mode)

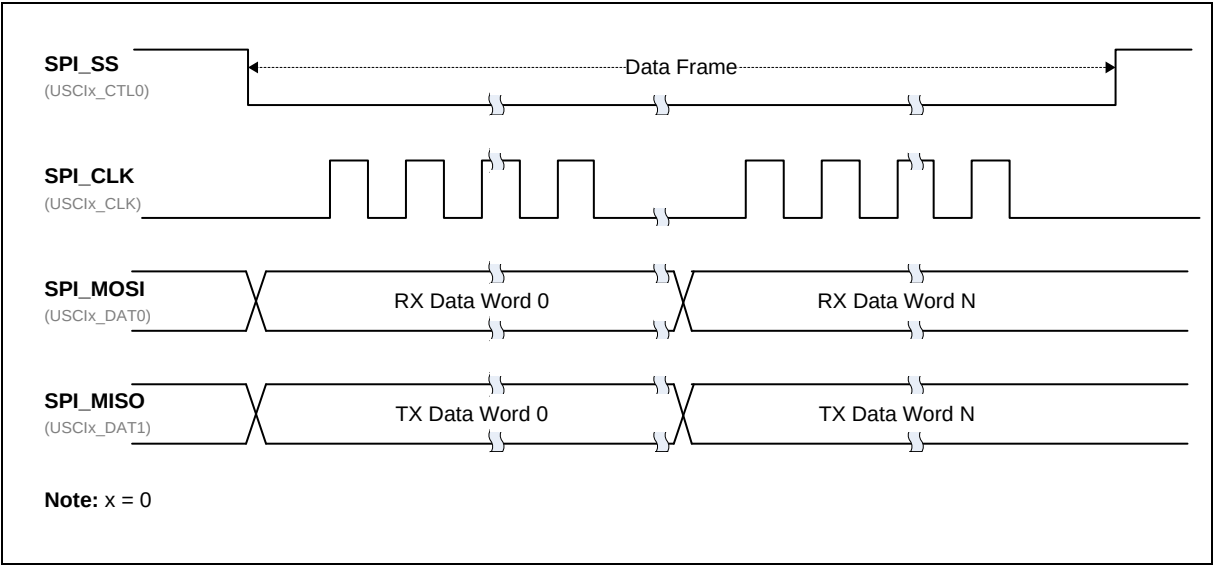


Figure 6.17-5 4-Wire Full-Duplex SPI Communication Signals (Slave Mode)

6.17.5.3 Serial Bus Clock Configuration

The USPI controller needs the peripheral clock to drive the USPI logic unit to perform the data transfer. The peripheral clock of USPI controller is from PCLK .

In Master mode, the frequency of the SPI bus clock is determined by protocol-relative clock generator. In general, the SPI bus clock is denoted as SPI clock. The frequency of SPI clock is half of f_{SAMP_CLK} , which can be selected by SPCLKSEL (USPI_BRGEN[3:2]).

In Slave mode, the SPI bus clock is provided by an off-chip Master device. The peripheral clock frequency, f_{PCLK} , of SPI Slave device must be 5-times faster than the serial bus clock rate of the SPI Master device connected together (i.e. the clock rate of serial bus clock < 1/5 peripheral clock f_{PCLK} in Slave mode).

In SPI protocol, SCLKMODE (USPI_PROTCTL[7:6]) defines not only the idle state of serial bus clock but also the serial clock edge used for transmit and receive data. Both Master and Slave devices on the same communication bus should have the same SCLKMODE configuration. The four kinds of serial bus clock configuration are shown below.

SCLKMODE [1:0]	SPI Clock Idle State	Transmit Timing	Receive Timing
0x0	Low	Falling edge	Rising edge
0x1	Low	Rising edge	Falling edge
0x2	High	Rising edge	Falling edge
0x3	High	Falling edge	Rising edge

Table 6.17-2 Serial Bus Clock Configuration

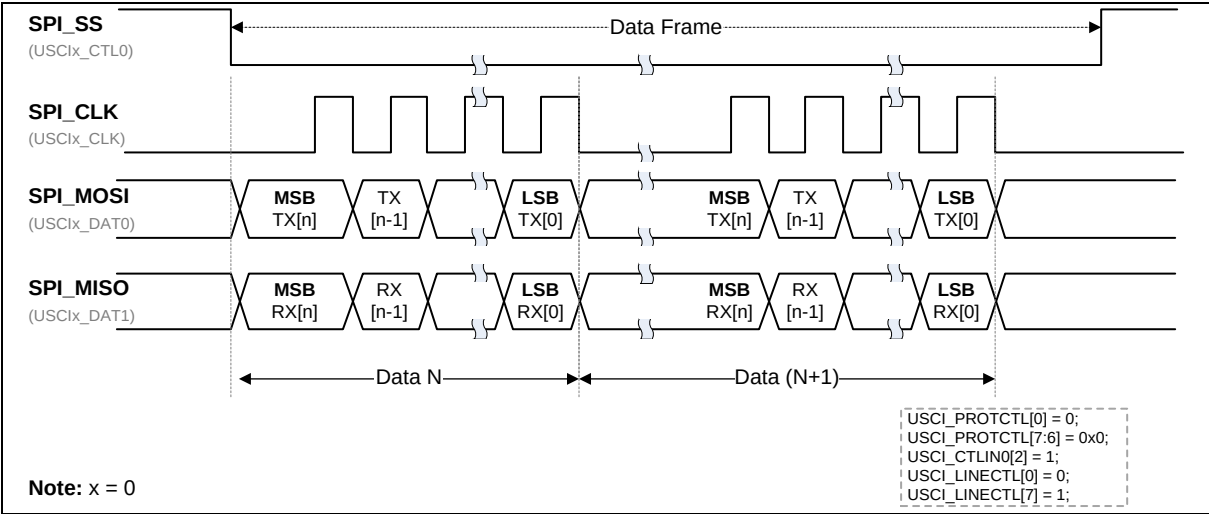


Figure 6.17-6 SPI Communication with Different SPI Clock Configuration (SCLKMODE=0x0)

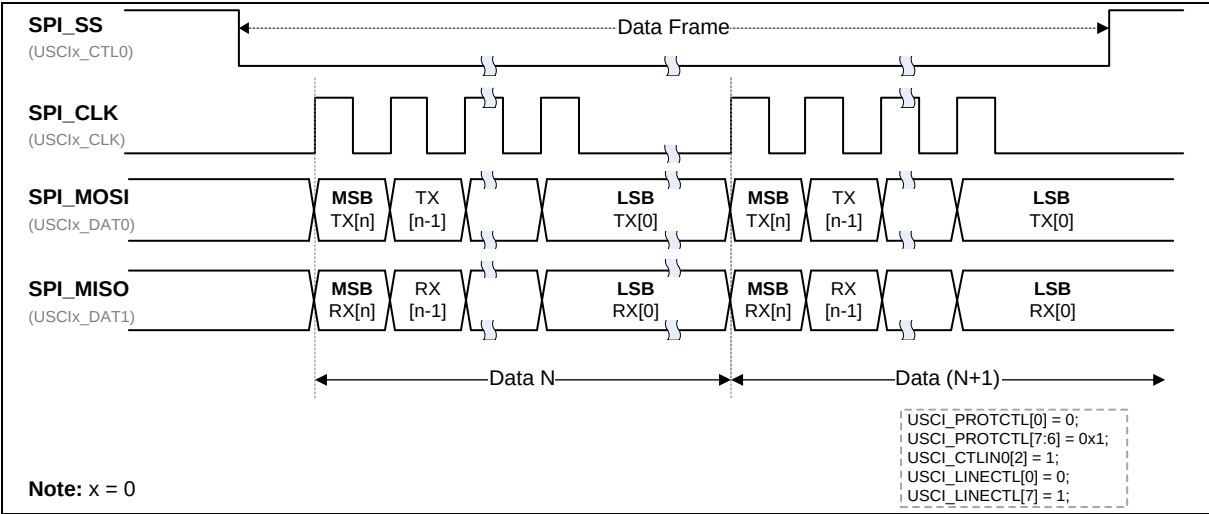


Figure 6.17-7 SPI Communication with Different SPI Clock Configuration (SCLKMODE=0x1)

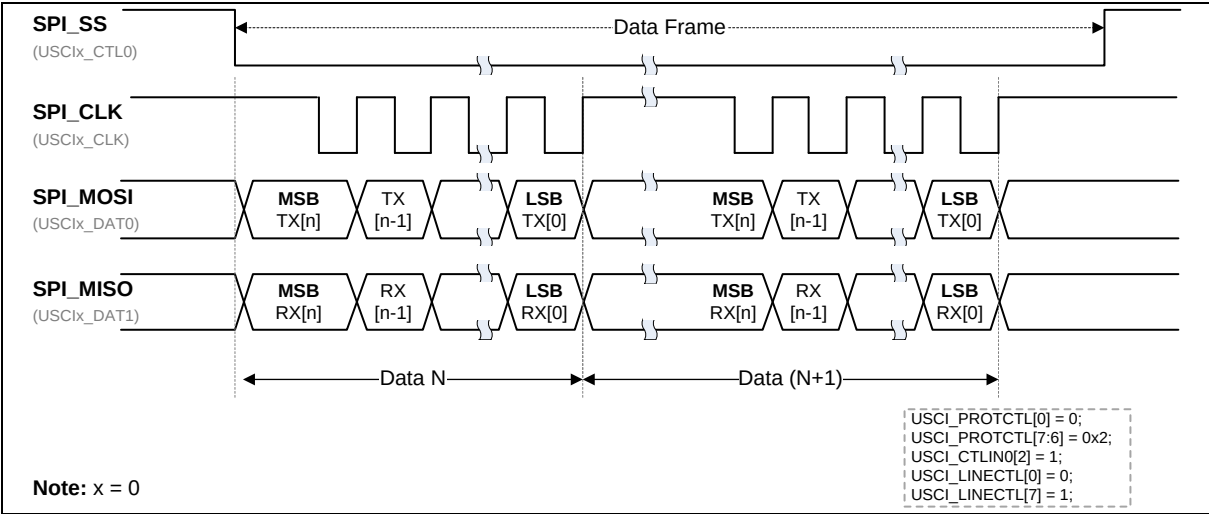


Figure 6.17-8 SPI Communication with Different SPI Clock Configuration (SCLKMODE=0x2)

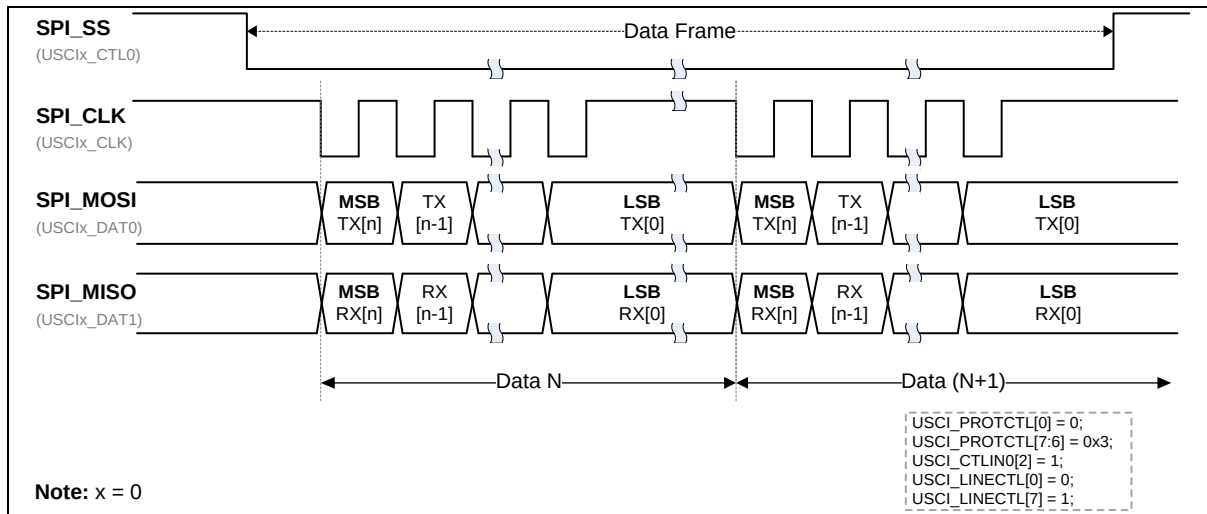


Figure 6.17-9 SPI Communication with Different SPI Clock Configuration (SCLKMODE=0x3)

6.17.5.4 Slave Select Signal

The slave selection signal of SPI protocol is active high by default. In SPI Master mode, the USPI controller can drive the control signal to off-chip SPI Slave device through slave select pin SPI_SS (USCIX_CTL0). In SPI Slave mode, the received slave select signal can be inverted by ININV (USPI_CTLIN0[2]).

If the slave select signal of external SPI Master device is low active, the ININV (USPI_CTLIN0[2]) setting of slave device should be set to 1 for the inversion of input control signal. If USPI operates as SPI Master mode, the output slave select inversion CTLOINV (USPI_LINECTL[7]) is also needed to set as 1 for the external SPI Slave device whose slave select signal is active low.

The duration between the slave select active edge and the first SPI clock input edge shall over 2 SPI bus clock cycles.

The input slave select signal of SPI Slave has to be kept inactive for at least 2 SPI bus clock cycles between two consecutive frames in order to correctly detect the end of a frame. In Slave mode, the SPI_MISO signal type is controlled by the SPI_SS pin. If the SPI_SS pin is inactive state, the SPI_MISO is set to tri-state to avoid interference with other Slave devices. If the SPI_SS pin is active state, the SPI_MISO is set to output mode for data transfer.

6.17.5.5 Transmit and Receive Data

The bit length of a transmit/receive data word in SPI protocol of USPI controller is defined in DWIDTH (USPI_LINECTL[11:8]), and it can be configured up to 16-bit length for transmitting and receiving data in SPI communication.

The LSB bit (USPI_LINECTL[0]) defines the order of transfer data bit. If the LSB bit is set to 1, the transmission data sequence is LSB first. If the LSB bit is cleared to 0, the transmission data sequence is MSB first.

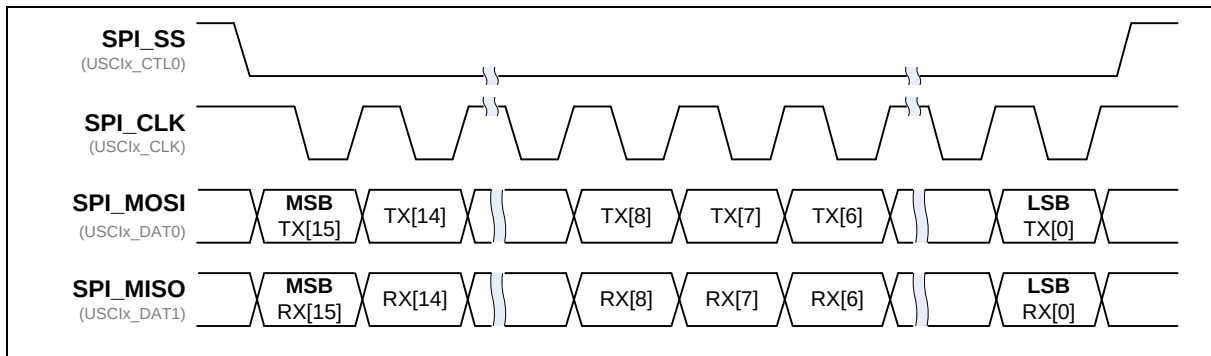


Figure 6.17-10 16-bit Data Length in One Word Transaction with MSB First Format

6.17.5.6 Word Suspend

SUSPITV (USPI_PROTCTL[11:8]) provides a configurable suspend interval, 0.5 ~ 15.5 SPI clock periods, between two successive transaction words in Master mode. The definition of the suspend interval is the interval between the last clock edge of the preceding transaction word and the first clock edge of the following transaction word. The default value of SUSPITV (USPI_PROTCTL[11:8]) is 0x3 (3.5 SPI clock cycles).

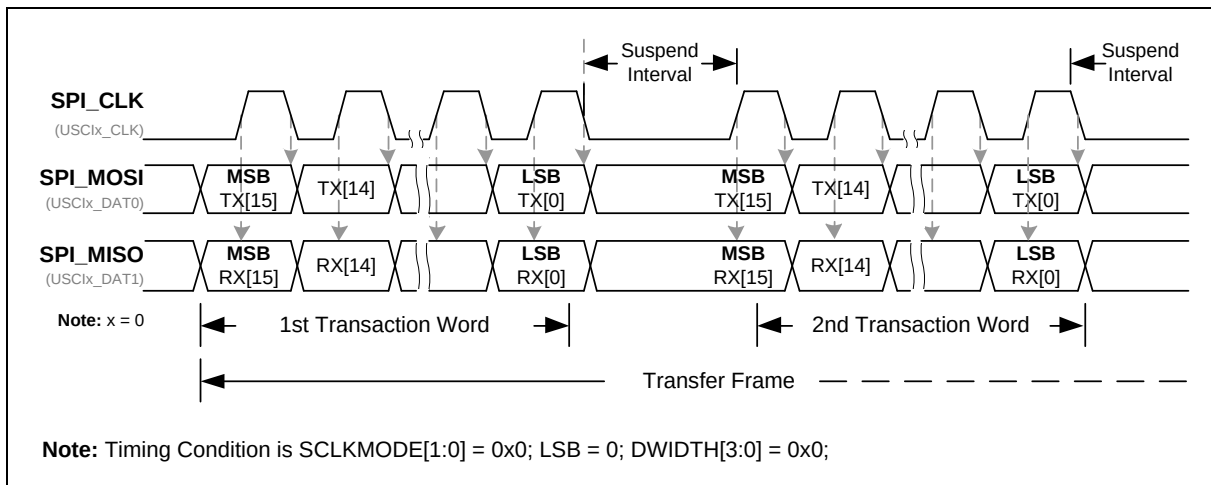


Figure 6.17-11 Word Suspend Interval between Two Transaction Words

6.17.5.7 Automatic Slave Select Function

AUTOSS (USPI_PROTCTL[3]) is used for SPI Master mode to enable the automatic slave select function. If the bit AUTOSS (USPI_PROTCTL[3]) is set, the slave select signal will be generated automatically and the setting value of SS (USPI_PROTCTL[2]) will not affect the output slave select (through USCIX_CTL0 line). This means that the slave select signal will be asserted by the USPI controller when the SPI data transfer is started by writing to the transmit buffer. And, it will be de-asserted after either all transaction is finished or one word transaction done if the value of SUSPITV (USPI_PROTCTL[11:8]) is equal to or greater than 3.

If the AUTOSS bit (USPI_PROTCTL[3]) is cleared, the slave select on USCIX_CTL0 pin will be asserted/de-asserted by setting/clearing the SS (USPI_PROTCTL[2]). The internal slave select signal is active high and the CTLOINV (USPI_LINECTL[7]) can be used for the inversion of the slave select signal.

In SPI Master mode, if the value of SUSPITV (USPI_PROTCTL[11:8]) is less than 3 and the AUTOSS (USPI_PROTCTL[3]) is set as 1, the slave select signal will be kept at active state between two successive word transactions.

In SPI Slave mode, to recognize the inactive state of the slave select signal, the inactive period of the received slave select signal must be larger than 2 peripheral clock cycles between two successive transactions.

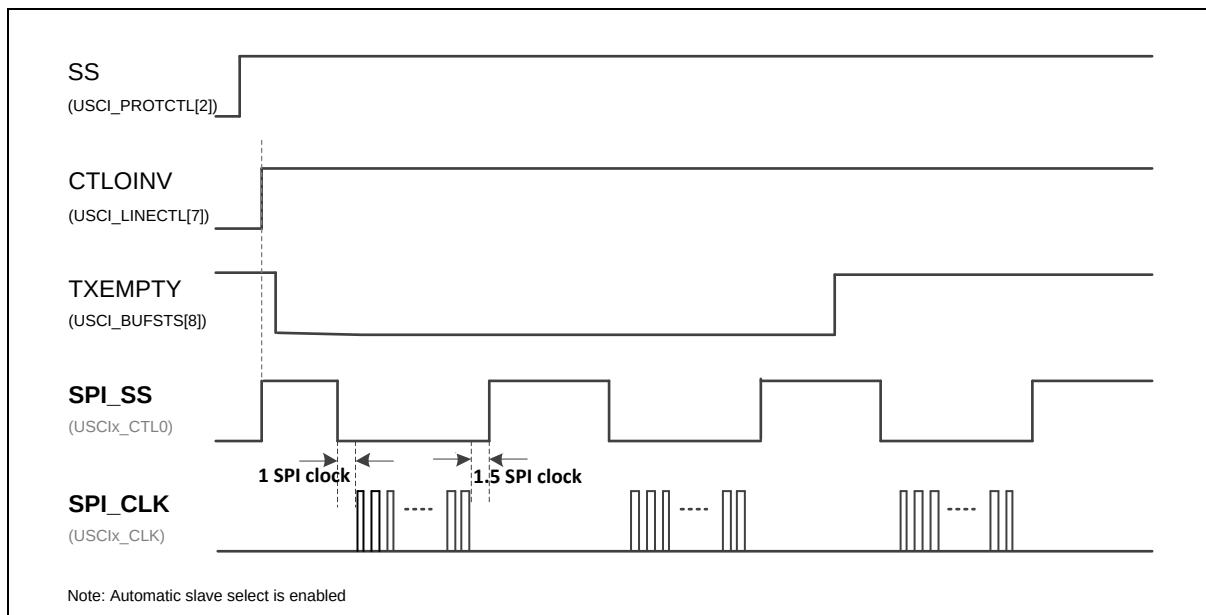


Figure 6.17-12 Auto Slave Select ($SUSPITV \geq 0x3$)

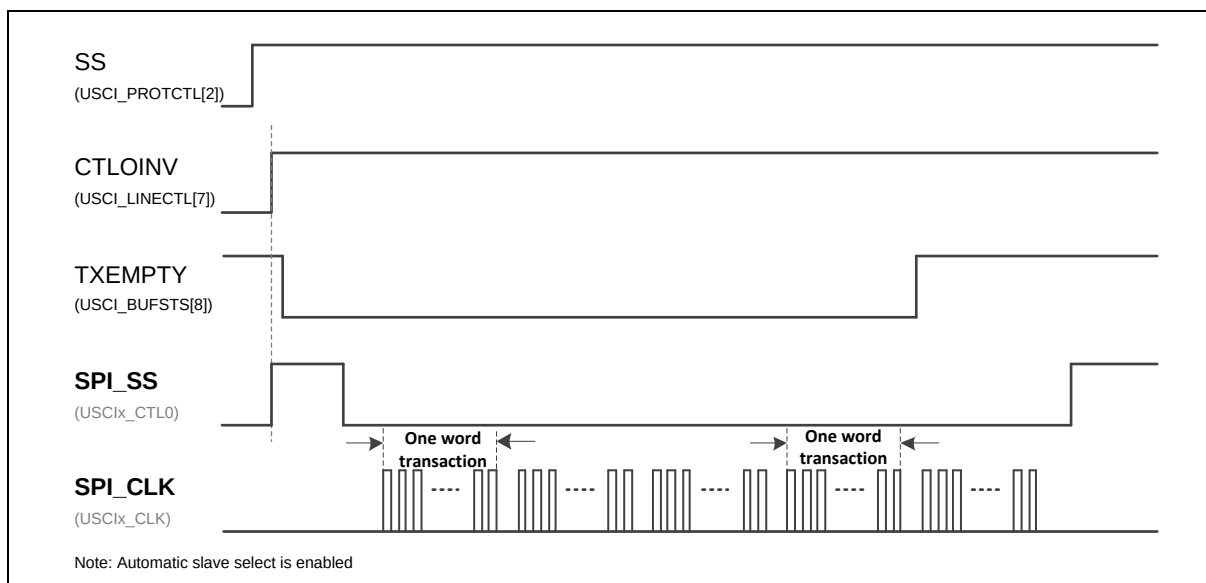


Figure 6.17-13 Auto Slave Select ($SUSPITV < 0x3$)

6.17.5.8 Slave 3-wire Mode

When the SLV3WIRE (USPI_PROTCTL[1]) is set by software to enable the Slave 3-wire mode, the USCI SPI communication can work with no slave select signal in Slave mode. The SLV3WIRE (USPI_PROTCTL[1]) only takes effect in SPI Slave mode. Only three pins, SPI_CLK (through USCIx_CLK line), SPI_MOSI (through USCIx_DAT0 line), and SPI_MISO (through USCIx_DAT1 line), are required to communicate with a SPI Master. When the SLV3WIRE (USPI_PROTCTL[1]) is set to 1, the SPI Slave will be ready to transmit/receive data after the SPI protocol is enabled by setting FUNMODE(USPI_CTL [2:0]) to 0x1.

6.17.5.9 Data Transfer Mode

The USPI controller supports full-duplex SPI transfer and one data channel half-duplex SPI transfer.

- Full-duplex SPI transfer

In full-duplex SPI transfer, there are two data pins. One is used for transmitting data and the other is used for receiving data. Thus, data transmission and data reception can be performed simultaneously.

SCLKMODE (USPI_PROTCTL[7:6]) defines the transition timing of the data shift output signal on USCIX_DAT0 pin. The transition may happen at the corresponding edge of SPI bus clock or active edge of slave select signal. The level of the last data bit of a data word is held on USCIX_DAT0 pin until the next data word begins with the next corresponding edge of the serial bus clock.

- One data channel half-duplex SPI transfer

In one data channel half-duplex SPI transfer, there is only one data pin for data transfer. Thus, the data transmission and data reception are at different time interval. The data shift direction is determined by PORTDIR (USPI_TXDAT[16]). Refer to the register description for more detailed information.

The function of one data channel half-duplex SPI transfer is similar to the full-duplex SPI protocol. All the transfer data timing is the same as the full-duplex SPI transfer.

Figure 6.17-14 shows the one output data channel and one input data channel half-duplex transfer diagrams with the external device.

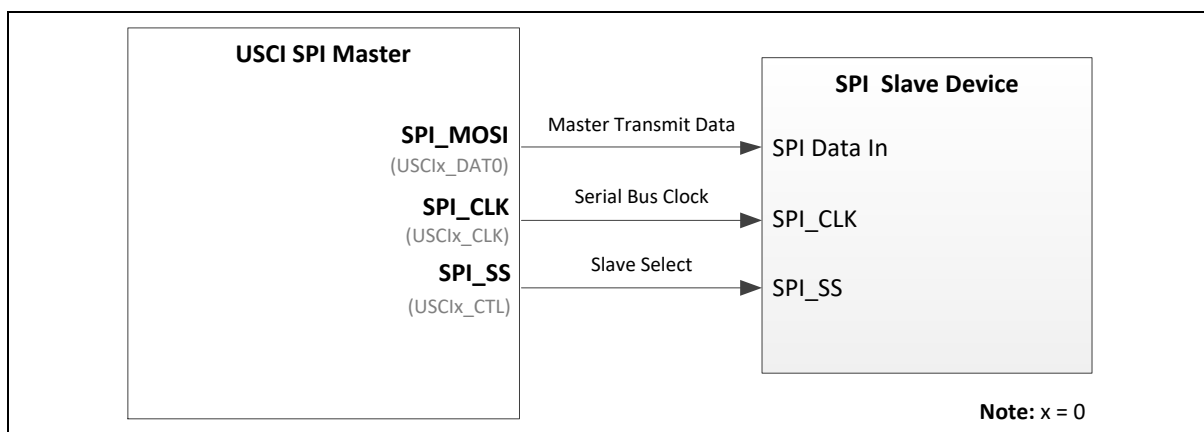


Figure 6.17-14 One Output Data Channel Half-duplex (SPI Master Mode)

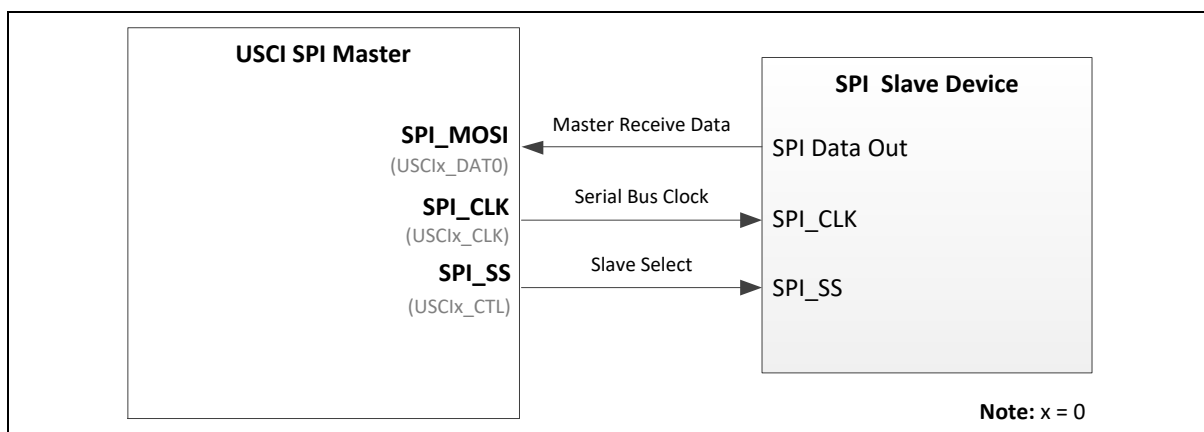


Figure 6.17-15 One Input Data Channel Half-duplex (SPI Master Mode)

The one data channel half-duplex transfer mode can be configured by TSMSEL[2:0] (USPI_PROTCTL[14:12]) and PORTDIR (USPI_TXDAT[16]) settings. When TSMSEL (USPI_PROTCTL[14:12]) is set to 0x4, one data channel half-duplex transfer mode is selected. The PORTDIR (USPI_TXDAT[16]) is used to define the direction of the corresponding transmit data. When the PORTDIR bit is set to 0, the USPI controller will send the corresponding data to external SPI device. When the PORTDIR bit is set to 1, the controller will read the corresponding data from the external SPI device.

For example, in one data channel half-duplex transfer mode with PORTDIR=0, USCI SPI transmits data through USCIx_DAT0 pin; if PORTDIR=1, USCI SPI receives data through USCIx_DAT0 pin.

6.17.5.10 Interrupt

Data Transfer Interrupts

- Transmit start interrupt
The interrupt event TXSTIF (USPI_PROTSTS[1]) is set after the start of the first data bit of a transmit data word. It can be cleared only by writing 1 to it.
- Transmit end interrupt
The interrupt event TXENDIF (USPI_PROTSTS[2]) is set after the start of the last data bit of the last transmit data which has been stored in transmit buffer. It can be cleared only by writing 1 to it.
- Receive start interrupt
The interrupt event RXSTIF (USPI_PROTSTS[3]) is set after the start of the first data bit of a receive data word. It can be cleared only by writing 1 to it.
- Receive end interrupt
The interrupt event RXENDIF (USPI_PROTSTS[4]) is set after the start of the last data bit of a receive data word. It can be cleared only by writing 1 to it.

Protocol-Related Interrupts

- SPI slave select interrupt
In SPI Slave mode, there are slave select active and in-active interrupt flags, SSACTIF (USPI_PROTSTS[9]) and SSINAIF (USPI_PROTSTS[8]), will be set to 1 when SLAVE (USPI_PROTCTL[0]) is set to 1 and Slave senses the slave select signal active or inactive. The SPI controller will issue an interrupt if SSINAIF (USPI_PROTSTS[8]) or SSACTIF (USPI_PROTSTS[9]), are set to 1. Because the internal slave select signal in SPI function is active high, the ININV (USPI_CTLIN0[2]) can be used for inverting the slave select signal comes from an active low device.
- Slave time-out interrupt
In SPI Slave mode, there is Slave time-out function for user to know that there is no serial clock input during the period of one word transaction. The Slave time-out function uses the timing measurement counter for the calculation of Slave time-out period which is defined by SLVTOCNT (USPI_PROTCTL[25:16]). TMCNTSRC (USPI_BRGEN[5]) can be used for clock frequency selection of timing measurement counter to calculate the Slave time-out period.

When the timing measurement counter is enabled by TMCNTEN (USPI_BRGEN[4]) and the setting value of SLVTOCNT (USPI_PROTCTL[25:16]) is not 0 in SPI Slave mode, the timing measurement counter will start counting after the first input serial clock of each received word data. This counter will be reset while receiving the following input serial clock and then keep counting. Finally, the timing measurement counter will be cleared and stopped after the finish of the current word transaction. If the value of the time-out counter is equal to or greater than the value of SLVTOCNT (USPI_PROTCTL[25:16]) before one word transaction is done, the Slave time-out interrupt event occurs and the SLVTOIF

(USPI_PROTSTS[5]) will be set to 1.

Buffer-Related Interrupts

The buffer-related interrupts are available if there is transmit/receive buffer in USPI controller.

- Receive buffer overrun interrupt

If there is receive buffer overrun event, RXOVIF (USPI_BUFSTS[3]) will be set as 1. It can be cleared by write 1 into it.

- Transmit buffer under-run interrupt

If there is transmit buffer under-run event, TXUDRIF (USPI_BUFSTS[11]) will be set as 1. It can be cleared by write 1 into it.

6.17.5.11 Timing Diagram

The slave select signal of USCI SPI protocol is active high by default, and it can be inverted by CTLOINV (USPI_LINECTL[7]) setting.

The idle state of serial bus clock and the serial bus clock edge used for transmit/receive data can be configured by setting SCLKMODE (USPI_PROTCTL[7:6]). The bit length of a transaction word data is determined by DWIDTH (USPI_LINECTL[11:8]), and data bit transfer sequence is determined by LSB (USPI_LINECTL[0]). Four SPI timing diagrams for Master/Slave operations and the related settings are shown below.

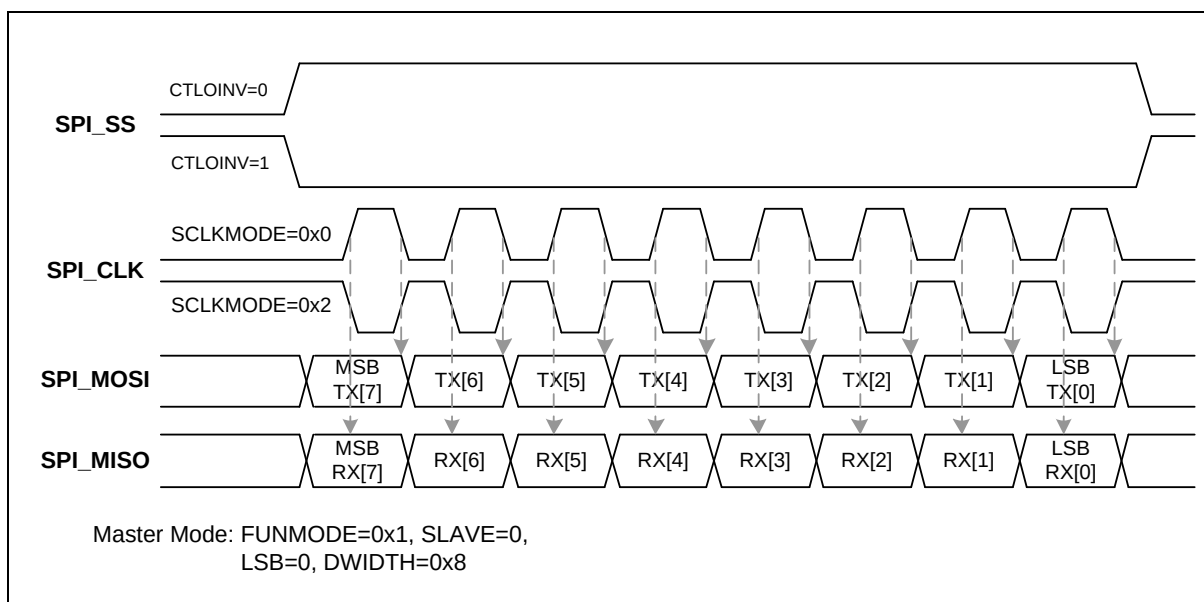


Figure 6.17-16 SPI Timing in Master Mode

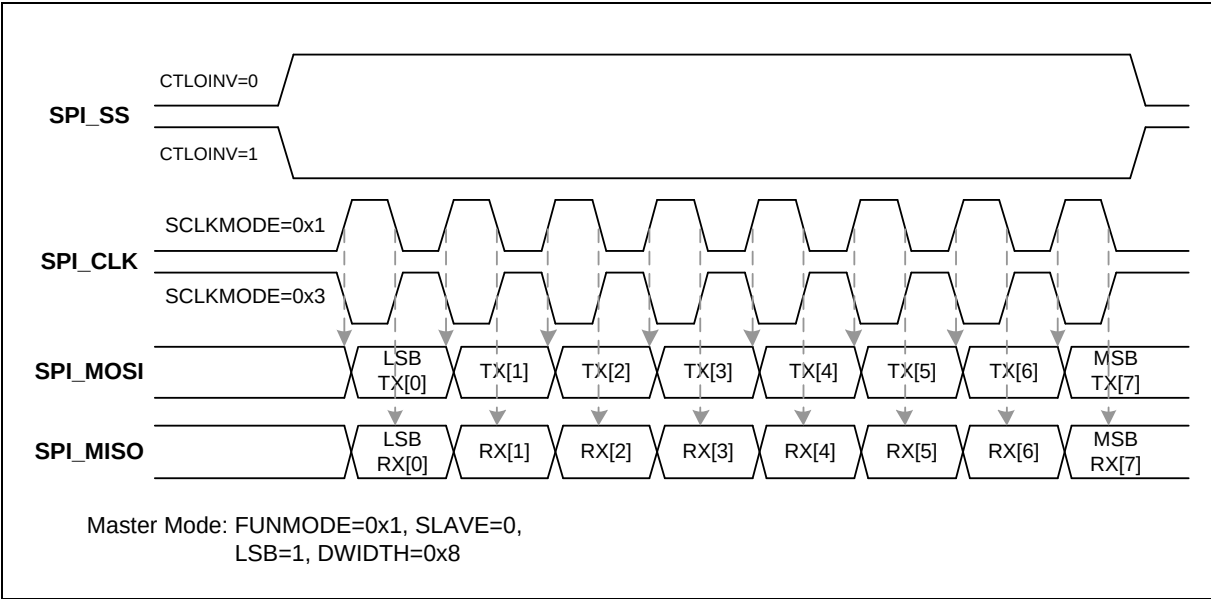


Figure 6.17-17 SPI Timing in Master Mode (Alternate Phase of Serial Bus Clock)

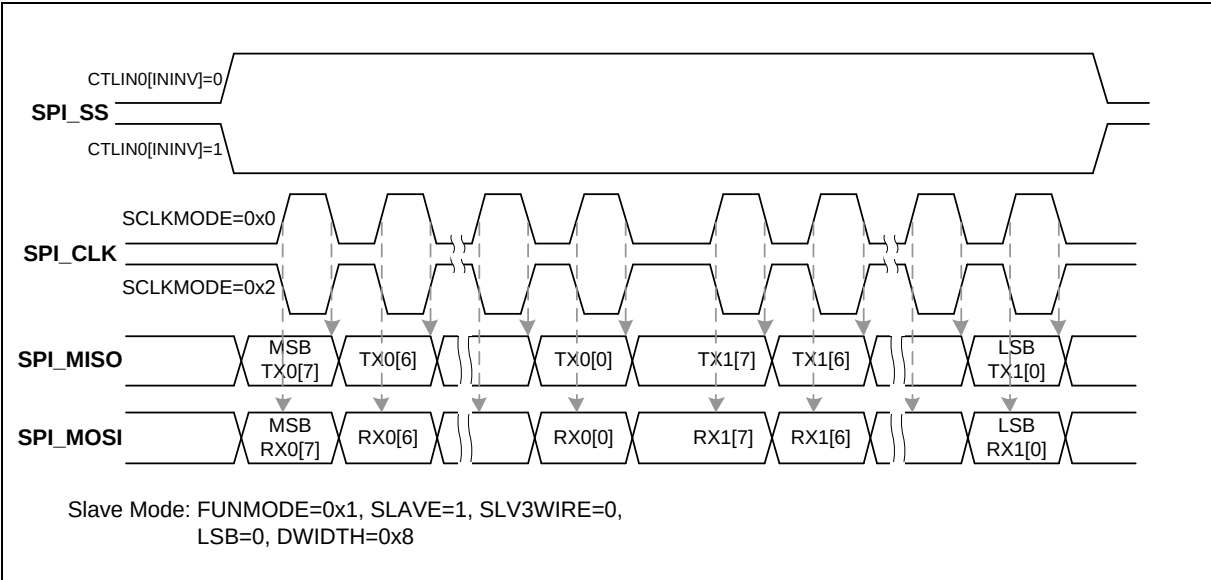


Figure 6.17-18 SPI Timing in Slave Mode

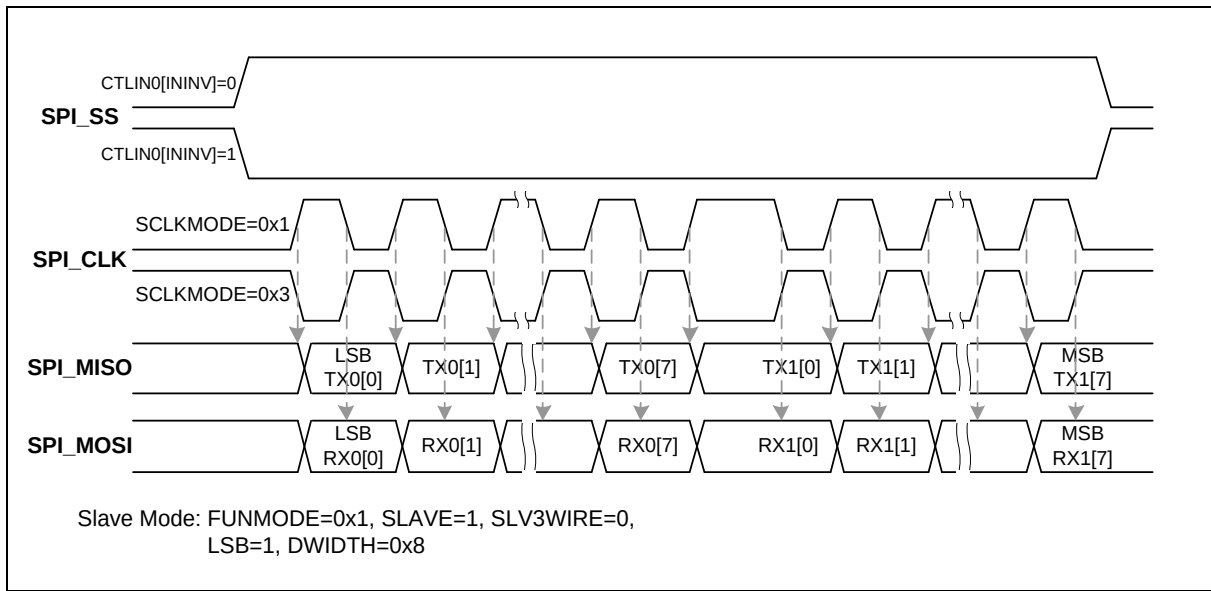


Figure 6.17-19 SPI Timing in Slave Mode (Alternate Phase of Serial Bus Clock)

6.17.5.12 Programming Flow

This section describes the programming flow for USCI SPI data transfer.

For Master mode:

1. Enable USPI peripheral clock by setting CLK_APBCLK1 register.
2. Configure user-specified pins as USPI function pins by setting corresponding multiple function control registers.
3. Set FUNMODE (USPI_CTL[2:0]) to 1 to select SPI mode.
4. Set USPI_BRGEN register to determine the SPI bus clock frequency.
5. According to the requirements of user's application, configure the settings as follows.
 - CTLOINV (USPI_LINECTL[7]): If the slave selection signal is active low, set this bit to 1; otherwise, set it to 0.
 - DWIDTH (USPI_LINECTL[11:8]): Data width setting.
 - LSB (USPI_LINECTL[0]): LSB first or MSB first.
 - TSMSEL (USPI_PROTCTL[14:12]): Full-duplex SPI transfer or one channel half-duplex SPI transfer.
 - SCLKMODE (USPI_PROTCTL[7:6]): Determine the clock timing.
 - AUTOSS (USPI_PROTCTL[3]): Enable automatic slave select function or not.
 - SLAVE (USPI_PROTCTL[0]): Set to 0 for Master mode.
 - Set PROTEN (USPI_PROTCTL[31]) to 1 to enable SPI protocol.
6. If automatic slave select function is disabled (AUTOSS=0), set SS (USPI_PROTCTL[2]) to 1 before data transfer; set SS to 0 to inactivate the slave selection signal by user's application.
7. Write USPI_TXDAT register to trigger SPI transfer. In half-duplex SPI transfer, the data pin direction is determined by PORTDIR (USPI_TXDAT[16]) setting.
8. User can get the received data by reading USPI_RXDAT register as long as RXEMPTY (USPI_BUFSTS[0]) is 0. The SPI data transfer can be triggered by writing USPI_TXDAT register as long as TXFULL (USPI_BUFSTS[9]) is 0.

For Slave mode:

1. Enable USPI peripheral clock by setting CLK_APBCLK1 register.
2. Configure user-specified pins as USPI function pins by setting corresponding multiple function control registers.
3. Set FUNMODE (USPI_CTL[2:0]) to 1 to select SPI mode.
4. According to the requirements of user's application, configure the settings as follows.
 - ININV (USPI_CTLIN0[2]): If the slave selection signal is active low, set this bit to 1; otherwise, set it to 0.
 - DWIDTH (USPI_LINECTL[11:8]): Data width setting.
 - LSB (USPI_LINECTL[0]): LSB first or MSB first.
 - TSMSEL (USPI_PROTCTL[14:12]): Full-duplex SPI transfer or one channel half-duplex SPI transfer.
 - SCLKMODE (USPI_PROTCTL[7:6]): Determine the clock timing.
 - SLAVE (USPI_PROTCTL[0]): Set to 1 for Slave mode.
5. Set PROTEN (USPI_PROTCTL[31]) to 1 to enable SPI protocol.
6. Write USPI_TXDAT register for transmission. In half-duplex SPI transfer, the data pin direction is determined by PORTDIR (USPI_TXDAT[16]) setting.
7. User can get the received data by reading USPI_RXDAT register as long as RXEMPTY (USPI_BUFSTS[0]) is 0. The next datum for transmission can be written to USPI_TXDAT register as long as TXFULL (USPI_BUFSTS[9]) is 0.

6.17.5.13 Wake-up Function

The USCI Controller in SPI mode supports wake-up system function. The wake-up source in SPI protocol is the transition of input slave select signal.

6.17.6 Register Map

R: read only, W: write only, R/W: both read and write

Register	Offset	R/W	Description	Reset Value
USCI_SPI Base Address: USPIn_BA = 0x400D_0000 + (0x1000 * n) n= 0				
USPI_CTL	USPIn_BA+0x00	R/W	USCI Control Register	0x0000_0000
USPI_INTEN	USPIn_BA+0x04	R/W	USCI Interrupt Enable Register	0x0000_0000
USPI_BRGEN	USPIn_BA+0x08	R/W	USCI Baud Rate Generator Register	0x0000_3C00
USPI_DATIN0	USPIn_BA+0x10	R/W	USCI Input Data Signal Configuration Register 0	0x0000_0000
USPI_CTLIN0	USPIn_BA+0x20	R/W	USCI Input Control Signal Configuration Register 0	0x0000_0000
USPI_CLKIN	USPIn_BA+0x28	R/W	USCI Input Clock Signal Configuration Register	0x0000_0000
USPI_LINECTL	USPIn_BA+0x2C	R/W	USCI Line Control Register	0x0000_0000
USPI_TXDAT	USPIn_BA+0x30	W	USCI Transmit Data Register	0x0000_0000
USPI_RXDAT	USPIn_BA+0x34	R	USCI Receive Data Register	0x0000_0000
USPI_BUFCTL	USPIn_BA+0x38	R/W	USCI Transmit/Receive Buffer Control Register	0x0000_0000
USPI_BUFSTS	USPIn_BA+0x3C	R	USCI Transmit/Receive Buffer Status Register	0x0000_0101
USPI_PDMACTL	USPIn_BA+0x40	R/W	USCI PDMA Control Register	0x0000_0000
USPI_WKCTL	USPIn_BA+0x54	R/W	USCI Wake-up Control Register	0x0000_0000
USPI_WKSTS	USPIn_BA+0x58	R/W	USCI Wake-up Status Register	0x0000_0000
USPI_PROTCTL	USPIn_BA+0x5C	R/W	USCI Protocol Control Register	0x0000_0300
USPI_PROTIEN	USPIn_BA+0x60	R/W	USCI Protocol Interrupt Enable Register	0x0000_0000
USPI_PROTSTS	USPIn_BA+0x64	R/W	USCI Protocol Status Register	0x0000_0000

6.17.7 Register Description

USCI Control Register (USPI_CTL)

Register	Offset	R/W	Description	Reset Value
USPI_CTL	USPIn_BA+0x00	R/W	USCI Control Register	0x0000_0000

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

Bits	Description	
[31:3]	Reserved	Reserved.
[2:0]	FUNMODE	Function Mode This bit field selects the protocol for this USCI controller. Selecting a protocol that is not available or a reserved combination disables the USCI. When switching between two protocols, the USCI has to be disabled before selecting a new protocol. Simultaneously, the USCI will be reset when user write 000 to FUNMODE. 000 = The USCI is disabled. All protocol related state machines are set to idle state. 001 = The SPI protocol is selected. 010 = The UART protocol is selected. 100 = The I ² C protocol is selected. Note: Other bit combinations are reserved.

USCI Interrupt Enable Register (USPI_INTEN)

Register	Offset	R/W	Description	Reset Value
USPI_INTEN	USPIn_BA+0x04	R/W	USCI Interrupt Enable Register	0x0000_0000

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

Bits	Description	
[31:5]	Reserved	Reserved.
[4]	RXENDIEN	Receive End Interrupt Enable Bit This bit enables the interrupt generation in case of a receive finish event. 0 = The receive end interrupt Disabled. 1 = The receive end interrupt Enabled. Note: The receive finish event happens when hardware receives the last bit of RX data into shift data unit.
[3]	RXSTIEN	Receive Start Interrupt Enable Bit This bit enables the interrupt generation in case of a receive start event. 0 = The receive start interrupt Disabled. 1 = The receive start interrupt Enabled. Note: For SPI master mode, the receive start event happens when SPI master sends slave select active and spi clock to the external SPI slave. For SPI slave mode, the receive start event happens when slave select of SPI slave is active and spi clock of SPI slave is inputted from the external SPI master.
[2]	TXENDIEN	Transmit End Interrupt Enable Bit This bit enables the interrupt generation in case of a transmit finish event. 0 = The transmit finish interrupt Disabled. 1 = The transmit finish interrupt Enabled. Note: The transmit finish event happens when hardware sends the last bit of TX data from shift data unit.
[1]	TXSTIEN	Transmit Start Interrupt Enable Bit This bit enables the interrupt generation in case of a transmit start event. 0 = The transmit start interrupt Disabled. 1 = The transmit start interrupt Enabled. Note: The transmit start event happens when hardware starts to move TX data from data buffer to shift data unit.
[0]	Reserved	Reserved.

USCI Baud Rate Generator Register (USPI_BRGEN)

Register	Offset	R/W	Description	Reset Value
USPI_BRGEN	USPIn_BA+0x08	R/W	USCI Baud Rate Generator Register	0x0000_3C00

31	30	29	28	27	26	25	24
Reserved						CLKDIV	
23	22	21	20	19	18	17	16
CLKDIV							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		TMCNTSRC	TMCNTEN	SPCLKSEL		PTCLKSEL	RCLKSEL

Bits	Description	
[31:26]	Reserved	Reserved.
[25:16]	CLKDIV	Clock Divider This bit field defines the ratio between the protocol clock frequency $f_{\text{PROT\_CLK}}$ and the clock divider frequency $f_{\text{DIV\_CLK}}$ ($f_{\text{DIV\_CLK}} = f_{\text{PROT\_CLK}} / (\text{CLKDIV} + 1)$).
[15:6]	Reserved	Reserved.
[5]	TMCNTSRC	Time Measurement Counter Clock Source Selection 0 = Time measurement counter with $f_{\text{PROT\_CLK}}$. 1 = Time measurement counter with $f_{\text{DIV\_CLK}}$.
[4]	TMCNTEN	Time Measurement Counter Enable Bit This bit enables the 10-bit timing measurement counter. 0 = Time measurement counter Disabled. 1 = Time measurement counter Enabled.
[3:2]	SPCLKSEL	Sample Clock Source Selection This bit field used for the clock source selection of sample clock ($f_{\text{SAMP\_CLK}}$) for the protocol processor. 00 = $f_{\text{DIV\_CLK}}$. 01 = $f_{\text{PROT\_CLK}}$. 10 = f_{SCLK} . 11 = $f_{\text{REF\_CLK}}$.
[1]	PTCLKSEL	Protocol Clock Source Selection This bit selects the source of protocol clock ($f_{\text{PROT\_CLK}}$). 0 = Reference clock $f_{\text{REF\_CLK}}$. 1 = $f_{\text{REF\_CLK}2}$ (its frequency is half of $f_{\text{REF\_CLK}}$).
[0]	RCLKSEL	Reference Clock Source Selection This bit selects the source of reference clock ($f_{\text{REF\_CLK}}$). 0 = Peripheral device clock f_{PCLK} . 1 = Reserved.

USCI Input Data Signal Configuration (USPI_DATIN0)

Register	Offset	R/W	Description	Reset Value
USPI_DATIN0	USPIn_BA+0x10	R/W	USCI Input Data Signal Configuration Register 0	0x0000_0000

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

Bits	Description	
[31:3]	Reserved	Reserved.
[2]	ININV	Input Signal Inverse Selection This bit defines the inverter enable of the input asynchronous signal. 0 = The un-synchronized input signal will not be inverted. 1 = The un-synchronized input signal will be inverted. Note: In SPI protocol, it is suggested this bit should be set as 0.
[1]	Reserved	Reserved.
[0]	SYNCSEL	Input Signal Synchronization Selection This bit selects if the un-synchronized input signal (with optionally inverted) or the synchronized (and optionally filtered) signal, which is synchronized with PCLK, can be used as input for the data shift unit. 0 = The un-synchronized signal can be taken as input for the data shift unit. 1 = The synchronized signal can be taken as input for the data shift unit. Note: In SPI protocol, it is suggested this bit should be set as 0.

USCI Input Control Signal Configuration (USPI_CTLIN0)

Register	Offset	R/W	Description	Reset Value
USPI_CTLIN0	USPIn_BA+0x20	R/W	USCI Input Control Signal Configuration Register 0	0x0000_0000

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

Bits	Description	
[31:3]	Reserved	Reserved.
[2]	ININV	Input Signal Inverse Selection This bit defines the inverter enable of the input asynchronous signal. 0 = The un-synchronized input signal will not be inverted. 1 = The un-synchronized input signal will be inverted.
[1]	Reserved	Reserved.
[0]	SYNCSEL	Input Synchronization Signal Selection This bit selects if the un-synchronized input signal (with optionally inverted) or the synchronized (and optionally filtered) signal, which is synchronized with PCLK, can be used as input for the data shift unit. 0 = The un-synchronized signal can be taken as input for the data shift unit. 1 = The synchronized signal can be taken as input for the data shift unit. Note: In SPI protocol, it is suggested this bit should be set as 0.

USCI Input Clock Signal Configuration (USPI_CLKIN)

Register	Offset	R/W	Description	Reset Value
USPI_CLKIN	USPIn_BA+0x28	R/W	USCI Input Clock Signal Configuration Register	0x0000_0000

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

Bits	Description	
[31:1]	Reserved	Reserved.
[0]	SYNCSEL	Input Synchronization Signal Selection This bit selects if the un-synchronized input signal or the synchronized (and optionally filtered) signal, which is synchronized with PCLK, can be used as input for the data shift unit. 0 = The un-synchronized signal can be taken as input for the data shift unit. 1 = The synchronized signal can be taken as input for the data shift unit. Note: In SPI protocol, it is suggested this bit should be set as 0.

USCI Line Control Register (USPI_LINECTL)

Register	Offset	R/W	Description	Reset Value
USPI_LINECTL	USPIn_BA+0x2C	R/W	USCI Line Control Register	0x0000_0000

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

Bits	Description	
[31:12]	Reserved	Reserved.
[11:8]	DWIDTH	Word Length of Transmission This bit field defines the data word length (amount of bits) for reception and transmission. The data word is always right-aligned in the data buffer. USCI support word length from 4 to 16 bits. 0x0: The data word contains 16 bits located at bit positions [15:0]. 0x1: Reserved. 0x2: Reserved. 0x3: Reserved. 0x4: The data word contains 4 bits located at bit positions [3:0]. 0x5: The data word contains 5 bits located at bit positions [4:0]. ... 0xF: The data word contains 15 bits located at bit positions [14:0].
[7]	CTLOINV	Control Signal Output Inverse Selection This bit defines the relation between the internal control signal and the output control signal. 0 = No effect. 1 = The control signal will be inverted before its output. Note: The control signal has different definitions in different protocol. In SPI protocol, the control signal means slave select signal.
[6]	Reserved	Reserved.
[5]	DATOINV	Data Output Inverse Selection This bit defines the relation between the internal shift data value and the output data signal of USCIX_DAT0/1 pin. 0 = Data output values of USCIX_DAT0/1 pins are not inverted. 1 = Data output values of USCIX_DAT0/1 pins are inverted.
[4:1]	Reserved	Reserved.
[0]	LSB	LSB First Transmission Selection 0 = The MSB, which bit of transmit/receive data buffer depends on the setting of DWIDTH,

		is transmitted/received first. 1 = The LSB, the bit 0 of data buffer, will be transmitted/received first.

USCI Transmit Data Register (USPI_TXDAT)

Register	Offset	R/W	Description	Reset Value
USPI_TXDAT	USPIn_BA+0x30	W	USCI Transmit Data Register	0x0000_0000

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

Bits	Description	
[31:17]	Reserved	Reserved.
[16]	PORTDIR	Port Direction Control This bit field is only available while USCI operates in SPI protocol (FUNMODE = 0x1) with half-duplex transfer. It is used to define the direction of the data port pin. When software writes USPI_TXDAT register, the transmit data and its port direction are settled simultaneously. 0 = The data pin is configured as output mode. 1 = The data pin is configured as input mode.
[15:0]	TXDAT	Transmit Data Software can use this bit field to write 16-bit transmit data for transmission. In order to avoid overwriting the transmit data, user have to check TXEMPTY (USPI_BUFSTS[8]) status before writing transmit data into this bit field.

USCI Receive Data Register (USPI_RXDAT)

Register	Offset	R/W	Description	Reset Value
USPI_RXDAT	USPIn_BA+0x34	R	USCI Receive Data Register	0x0000_0000

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

Bits	Description	
[31:16]	Reserved	Reserved.
[15:0]	RXDAT	Received Data This bit field monitors the received data which stored in receive data buffer.

USCI Transmit/Receive Buffer Control Register (USPI_BUFCTL)

Register	Offset	R/W	Description	Reset Value
USPI_BUFCTL	USPIn_BA+0x38	R/W	USCI Transmit/Receive Buffer Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved						RXRST	TXRST
15	14	13	12	11	10	9	8
RXCLR	RXOVLEN	Reserved					
7	6	5	4	3	2	1	0
TXCLR	TXUDRIEN	Reserved					

Bits	Description	
[31:18]	Reserved	Reserved.
[17]	RXRST	Receive Reset 0 = No effect. 1 = Reset the receive-related counters, state machine, and the content of receive shift register and data buffer. Note: It is cleared automatically after one PCLK cycle.
[16]	TXRST	Transmit Reset 0 = No effect. 1 = Reset the transmit-related counters, state machine, and the content of transmit shift register and data buffer. Note1: It is cleared automatically after one PCLK cycle. Note2: Write 1 to this bit will set the output data pin to zero if USPI_PROTCTL[28]=0.
[15]	RXCLR	Clear Receive Buffer 0 = No effect. 1 = The receive buffer is cleared. Should only be used while the buffer is not taking part in data traffic. Note: It is cleared automatically after one PCLK cycle.
[14]	RXOVLEN	Receive Buffer Overrun Interrupt Enable Bit 0 = Receive overrun interrupt Disabled. 1 = Receive overrun interrupt Enabled.
[13:8]	Reserved	Reserved.
[7]	TXCLR	Clear Transmit Buffer 0 = No effect. 1 = The transmit buffer is cleared. Should only be used while the buffer is not taking part in data traffic. Note: It is cleared automatically after one PCLK cycle.
[6]	TXUDRIEN	Slave Transmit Under-run Interrupt Enable Bit

		0 = Transmit under-run interrupt Disabled. 1 = Transmit under-run interrupt Enabled.
[5:0]	Reserved	Reserved.

USCI Transmit/Receive Buffer Status Register (USPI_BUFSTS)

Register	Offset	R/W	Description	Reset Value
USPI_BUFSTS	USPIn_BA+0x3C	R	USCI Transmit/Receive Buffer Status Register	0x0000_0101

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved				TXUDRIF	Reserved	TXFULL	TXEMPTY
7	6	5	4	3	2	1	0
Reserved				RXOVIF	Reserved	RXFULL	RXEMPTY

Bits	Description	
[31:12]	Reserved	Reserved.
[11]	TXUDRIF	Transmit Buffer Under-run Interrupt Status This bit indicates that a transmit buffer under-run event has been detected. If enabled by TXUDRIEN (USPI_BUFCTL[6]), the corresponding interrupt request is activated. It is cleared by software writes 1 to this bit. 0 = A transmit buffer under-run event has not been detected. 1 = A transmit buffer under-run event has been detected.
[10]	Reserved	Reserved.
[9]	TXFULL	Transmit Buffer Full Indicator 0 = Transmit buffer is not full. 1 = Transmit buffer is full.
[8]	TXEMPTY	Transmit Buffer Empty Indicator 0 = Transmit buffer is not empty. 1 = Transmit buffer is empty and available for the next transmission datum.
[7:4]	Reserved	Reserved.
[3]	RXOVIF	Receive Buffer Over-run Interrupt Status This bit indicates that a receive buffer overrun event has been detected. If RXOVIEN (USPI_BUFCTL[14]) is enabled, the corresponding interrupt request is activated. It is cleared by software writes 1 to this bit. 0 = A receive buffer overrun event has not been detected. 1 = A receive buffer overrun event has been detected.
[2]	Reserved	Reserved.
[1]	RXFULL	Receive Buffer Full Indicator 0 = Receive buffer is not full. 1 = Receive buffer is full.
[0]	RXEMPTY	Receive Buffer Empty Indicator 0 = Receive buffer is not empty.

		1 = Receive buffer is empty.

USCI PDMA Control Register (USPI_PDMACTL)

Register	Offset	R/W	Description	Reset Value
USPI_PDMACTL	USPIn_BA+0x40	R/W	USCI PDMA Control Register	0x0000_0000

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

Bits	Description	
[31:4]	Reserved	Reserved.
[3]	PDMAEN	PDMA Mode Enable Bit 0 = PDMA function Disabled. 1 = PDMA function Enabled.
[2]	RXPDMAEN	PDMA Receive Channel Available 0 = Receive PDMA function Disabled. 1 = Receive PDMA function Enabled.
[1]	TXPDMAEN	PDMA Transmit Channel Available 0 = Transmit PDMA function Disabled. 1 = Transmit PDMA function Enabled.
[0]	PDMARST	PDMA Reset 0 = No effect. 1 = Reset the USCI's PDMA control logic. This bit will be cleared to 0 automatically.

USCI Wake-up Control Register (USPI_WKCTL)

Register	Offset	R/W	Description	Reset Value
USPI_WKCTL	USPIn_BA+0x54	R/W	USCI Wake-up Control Register	0x0000_0000

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

Bits	Description	
[31:3]	Reserved	Reserved.
[2]	PDBOPT	Power Down Blocking Option 0 = If user attempts to enter Power-down mode by executing WFI while the protocol is in transferring, MCU will stop the transfer and enter Power-down mode immediately. 1 = If user attempts to enter Power-down mode by executing WFI while the protocol is in transferring, the on-going transfer will not be stopped and MCU will enter idle mode immediately.
[1]	Reserved	Reserved.
[0]	WKEN	Wake-up Enable Bit 0 = Wake-up function Disabled. 1 = Wake-up function Enabled.

USCI Wake-up Status Register (USPI_WKSTS)

Register	Offset	R/W	Description	Reset Value
USPI_WKSTS	USPIn_BA+0x58	R/W	USCI Wake-up Status Register	0x0000_0000

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

Bits	Description	
[31:1]	Reserved	Reserved.
[0]	WKF	Wake-up Flag When chip is woken up from Power-down mode, this bit is set to 1. Software can write 1 to clear this bit. Note: The wake-up flag also is set to 1 when PDBOPT (USCI_WKCTL[2]) is enabled.

USCI Protocol Control Register – USPI_PROTCTL (SPI)

Register	Offset	R/W	Description	Reset Value
USPI_PROTCTL	USPIn_BA+0x5C	R/W	USCI Protocol Control Register	0x0000_0300

31	30	29	28	27	26	25	24
PROTEN	Reserved		TXUDRPOL	Reserved		SLVTOCNT	
23	22	21	20	19	18	17	16
SLVTOCNT							
15	14	13	12	11	10	9	8
Reserved	TSMSEL			SUSPITV			
7	6	5	4	3	2	1	0
SCLKMODE		Reserved		AUTOSS	SS	SLV3WIRE	SLAVE

Bits	Description	
[31]	PROTEN	SPI Protocol Enable Bit 0 = SPI Protocol Disabled. 1 = SPI Protocol Enabled.
[30:29]	Reserved	Reserved.
[28]	TXUDRPOL	Transmit Under-run Data Polarity (for Slave) This bit defines the transmitting data level of USC1x_DAT1 when no data is available for transferring. 0 = The output data level is 0 if TX under run event occurs. 1 = The output data level is 1 if TX under run event occurs.
[27:26]	Reserved	Reserved.
[25:16]	SLVTOCNT	Slave Mode Time-out Period (Slave Only) In Slave mode, this bit field is used for Slave time-out period. This bit field indicates how many clock periods (selected by TMCNTSRC, USPI_BRGEN[5]) between the two edges of input SCLK will assert the Slave time-out event. Writing 0x0 into this bit field will disable the Slave time-out function. Example: Assume SLVTOCNT is 0x0A and TMCNTSRC (USPI_BRGEN[5]) is 1, it means the time-out event will occur if the state of SPI bus clock pin is not changed more than (10+1) periods of f_{DIV_CLK} .
[15]	Reserved	Reserved.
[14:12]	TSMSEL	Transmit Data Mode Selection This bit field describes how receive and transmit data is shifted in and out. 000 = TSMSEL: Full-duplex SPI. 100 = TSMSEL: Half-duplex SPI. Others = Reserved. Note: Changing the value of this bit field will produce the TXRST and RXRST to clear the TX/RX data buffer automatically.
[11:8]	SUSPITV	Suspend Interval (Master Only) This bit field provides the configurable suspend interval between two successive transmit/receive transaction in a transfer. The definition of the suspend interval is the interval

		between the last clock edge of the preceding transaction word and the first clock edge of the following transaction word. The default value is 0x3. The period of the suspend interval is obtained according to the following equation. $(SUSPITV + 0.5) * \text{period of SPI\_CLK clock cycle}$ Example: SUSPITV = 0x0 ... 0.5 SPI_CLK clock cycle. SUSPITV = 0x1 ... 1.5 SPI_CLK clock cycle. SUSPITV = 0xE ... 14.5 SPI_CLK clock cycle. SUSPITV = 0xF ... 15.5 SPI_CLK clock cycle.
[7:6]	SCLKMODE	Serial Bus Clock Mode This bit field defines the SCLK idle status, data transmit, and data receive edge. 00 = MODE0. The idle state of SPI clock is low level. Data is transmitted with falling edge and received with rising edge. 01 = MODE1. The idle state of SPI clock is low level. Data is transmitted with rising edge and received with falling edge. 10 = MODE2. The idle state of SPI clock is high level. Data is transmitted with rising edge and received with falling edge. 11 = MODE3. The idle state of SPI clock is high level. Data is transmitted with falling edge and received with rising edge.
[5:4]	Reserved	Reserved.
[3]	AUTOSS	Automatic Slave Select Function Enable (Master Only) 0 = Slave select signal will be controlled by the setting value of SS (USPI_PROTCTL[2]) bit. 1 = Slave select signal will be generated automatically. The slave select signal will be asserted by the SPI controller when transmit/receive is started, and will be de-asserted after each transmit/receive is finished.
[2]	SS	Slave Select Control (Master Only) If AUTOSS bit is cleared, setting this bit to 1 will set the slave select signal to active state, and setting this bit to 0 will set the slave select signal back to inactive state. If the AUTOSS function is enabled (AUTOSS = 1), the setting value of this bit will not affect the current state of slave select signal. Note: In SPI protocol, the internal slave select signal is active high.
[1]	SLV3WIRE	Slave 3-wire Mode Selection (Slave Only) The SPI protocol can work with 3-wire interface (without slave select signal) in Slave mode. 0 = 4-wire bi-direction interface. 1 = 3-wire bi-direction interface.
[0]	SLAVE	Slave Mode Selection 0 = Master mode. 1 = Slave mode.

USCI Protocol Interrupt Enable Register – USPI_PROTIEN (SPI)

Register	Offset	R/W	Description	Reset Value
USPI_PROTIEN	USPIn_BA+0x60	R/W	USCI Protocol Interrupt Enable Register	0x0000_0000

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

Bits	Description
[31:4]	Reserved Reserved.
[3]	SLVBEIEN Slave Mode Bit Count Error Interrupt Enable Bit If data transfer is terminated by slave time-out or slave select inactive event in Slave mode, so that the transmit/receive data bit count does not match the setting of DWIDTH (USPI_LINECTL[11:8]). Bit count error event occurs. 0 = The Slave mode bit count error interrupt Disabled. 1 = The Slave mode bit count error interrupt Enabled.
[2]	SLVTOIEN Slave Time-out Interrupt Enable Bit In SPI protocol, this bit enables the interrupt generation in case of a Slave time-out event. 0 = The Slave time-out interrupt Disabled. 1 = The Slave time-out interrupt Enabled.
[1]	SSACTIEN Slave Select Active Interrupt Enable Bit This bit enables/disables the generation of a slave select interrupt if the slave select changes to active. 0 = Slave select active interrupt generation Disabled. 1 = Slave select active interrupt generation Enabled.
[0]	SSINAIEN Slave Select Inactive Interrupt Enable Bit This bit enables/disables the generation of a slave select interrupt if the slave select changes to inactive. 0 = Slave select inactive interrupt generation Disabled. 1 = Slave select inactive interrupt generation Enabled.

USCI Protocol Status Register – USPI PROTSTS (SPI)

Register	Offset	R/W	Description	Reset Value
USPI_PROTSTS	USPIn_BA+0x64	R/W	USCI Protocol Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved					SLVUDR	BUSY	SSLINE
15	14	13	12	11	10	9	8
Reserved						SSACTIF	SSINAIF
7	6	5	4	3	2	1	0
Reserved	SLVBEIF	SLVTOIF	RXENDIF	RXSTIF	TXENDIF	TXSTIF	Reserved

Bits	Description	
[31:19]	Reserved	Reserved.
[18]	SLVUDR	Slave Mode Transmit Under-run Status (Read Only) In Slave mode, if there is no available transmit data in buffer while transmit data shift out caused by input serial bus clock, this status flag will be set to 1. This bit indicates whether the current shift-out data of word transmission is switched to TXUDRPOL (USPI_PROTCTL[28]) or not. 0 = Slave transmit under-run event does not occur. 1 = Slave transmit under-run event occurs.
[17]	BUSY	Busy Status (Read Only) 0 = SPI is in idle state. 1 = SPI is in busy state. The following listing are the bus busy conditions: f. USPI_PROTCTL[31] = 1 and the TXEMPTY = 0. g. For SPI Master mode, the TXEMPTY = 1 but the current transaction is not finished yet. h. For SPI Slave mode, the USPI_PROTCTL[31] = 1 and there is serial clock input into the SPI core logic when slave select is active. i. For SPI Slave mode, the USPI_PROTCTL[31] = 1 and the transmit buffer or transmit shift register is not empty even if the slave select is inactive.
[16]	SSLINE	Slave Select Line Bus Status (Read Only) This bit is only available in Slave mode. It used to monitor the current status of the input slave select signal on the bus. 0 = The slave select line status is 0. 1 = The slave select line status is 1.
[15:10]	Reserved	Reserved.
[9]	SSACTIF	Slave Select Active Interrupt Flag (for Slave Only) This bit indicates that the internal slave select signal has changed to active. It is cleared by software writes one to this bit. 0 = The slave select signal has not changed to active.

		1 = The slave select signal has changed to active. Note1: The internal slave select signal is active high. Note2: SSACTIF fix to 0 when slave 3-wire mode enable.
[8]	SSINAIF	Slave Select Inactive Interrupt Flag (for Slave Only) This bit indicates that the internal slave select signal has changed to inactive. It is cleared by software writes 1 to this bit 0 = The slave select signal has not changed to inactive. 1 = The slave select signal has changed to inactive. Note1: The internal slave select signal is active high. Note2: SSINAIF fix to 0 when slave 3-wire mode enable.
[7]	Reserved	Reserved.
[6]	SLVBEIF	Slave Bit Count Error Interrupt Flag (for Slave Only) 0 = Slave bit count error event did not occur. 1 = Slave bit count error event occurred. Note1: It is cleared by software write 1 to this bit. If the transmit/receive data bit count does not match the setting of DWIDTH (USPI_LINECTL[11:8]), bit count error event occurs. Note2: SLVBEIF fix to 0 when slave 3-wire mode enable.
[5]	SLVTOIF	Slave Time-out Interrupt Flag (for Slave Only) 0 = Slave time-out event did not occur. 1 = Slave time-out event occurred. Note: It is cleared by software write 1 to this bit
[4]	RXENDIF	Receive End Interrupt Flag 0 = Receive end event did not occur. 1 = Receive end event occurred. Note: It is cleared by software write 1 to this bit. The receive end event happens when hardware receives the last bit of RX data into shift data unit.
[3]	RXSTIF	Receive Start Interrupt Flag 0 = Receive start event did not occur. 1 = Receive start event occurred. Note: It is cleared by software write 1 to this bit. For SPI master mode, the receive start event happens when SPI master sends slave select active and spi clock to the external SPI slave. For SPI slave mode, the receive start event happens when slave select of SPI slave is active and spi clock of SPI slave is inputted from the external SPI master.
[2]	TXENDIF	Transmit End Interrupt Flag 0 = Transmit end event did not occur. 1 = Transmit end event occurred. Note: It is cleared by software write 1 to this bit. The transmit end event happens when hardware sends the last bit of TX data from shift data unit.
[1]	TXSTIF	Transmit Start Interrupt Flag 0 = Transmit start event did not occur. 1 = Transmit start event occurred. Note: It is cleared by software write 1 to this bit. The transmit start event happens when hardware starts to move TX data from data buffer to shift data unit.
[0]	Reserved	Reserved.

6.18 USCI - I²C Mode

6.18.1 Overview

On I²C bus, data is transferred between a Master and a Slave. Data bits transfer on the SCL and SDA lines are synchronously on a byte-by-byte basis. Each data byte is 8-bit. There is one SCL clock pulse for each data bit with the MSB being transmitted first, and an acknowledge bit follows each transferred byte. Each bit is sampled during the high period of SCL; therefore, the SDA line may be changed only during the low period of SCL and must be held stable during the high period of SCL. A transition on the SDA line while SCL is high is interpreted as a command (START or STOP). Please refer to Figure 6.18-1 for more detailed I²C BUS Timing.

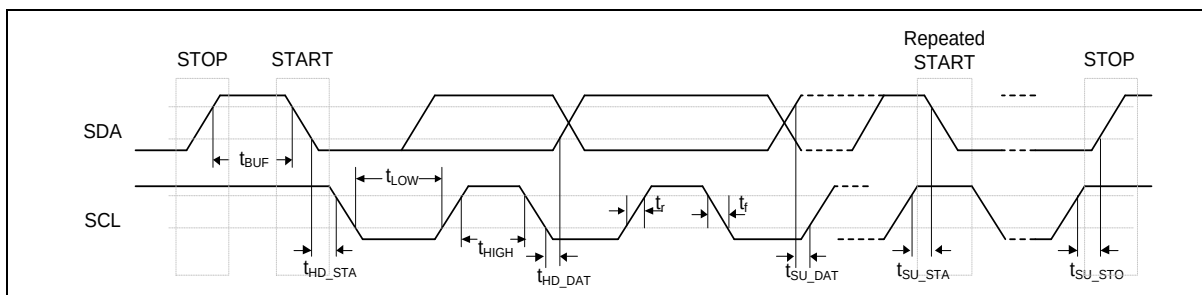


Figure 6.18-1 I²C Bus Timing

The device's on-chip I²C provides the serial interface that meets the I²C bus standard mode specification. The I²C port handles byte transfers autonomously. The I²C mode is selected by FUNMODE (UI2C_CTL [2:0]) = 100B. When enabling this port, the USCI interfaces to the I²C bus via two pins: SDA and SCL. When I/O pins are used as I²C ports, user must set the pins function to I²C in advance.

Note: Pull-up resistor is needed for I²C operation because the SDA and SCL are set to open-drain pins when USCI is selected to I²C operation mode.

6.18.2 Features

- Full master and slave device capability
- Supports 7-bit addressing, as well as 10-bit addressing
- Communication in standard mode (100 kBit/s) or in fast mode (up to 400 kBit/s)
- Supports multi-master bus
- Supports one transmit buffer and two receive buffers for data payload
- Supports 10-bit bus time-out capability
- Supports bus monitor mode.
- Supports Power down wake-up by data toggle or address match
- Supports setup/hold time programmable
- Supports multiple address recognition (two slave address with mask option)

6.18.3 Block Diagram

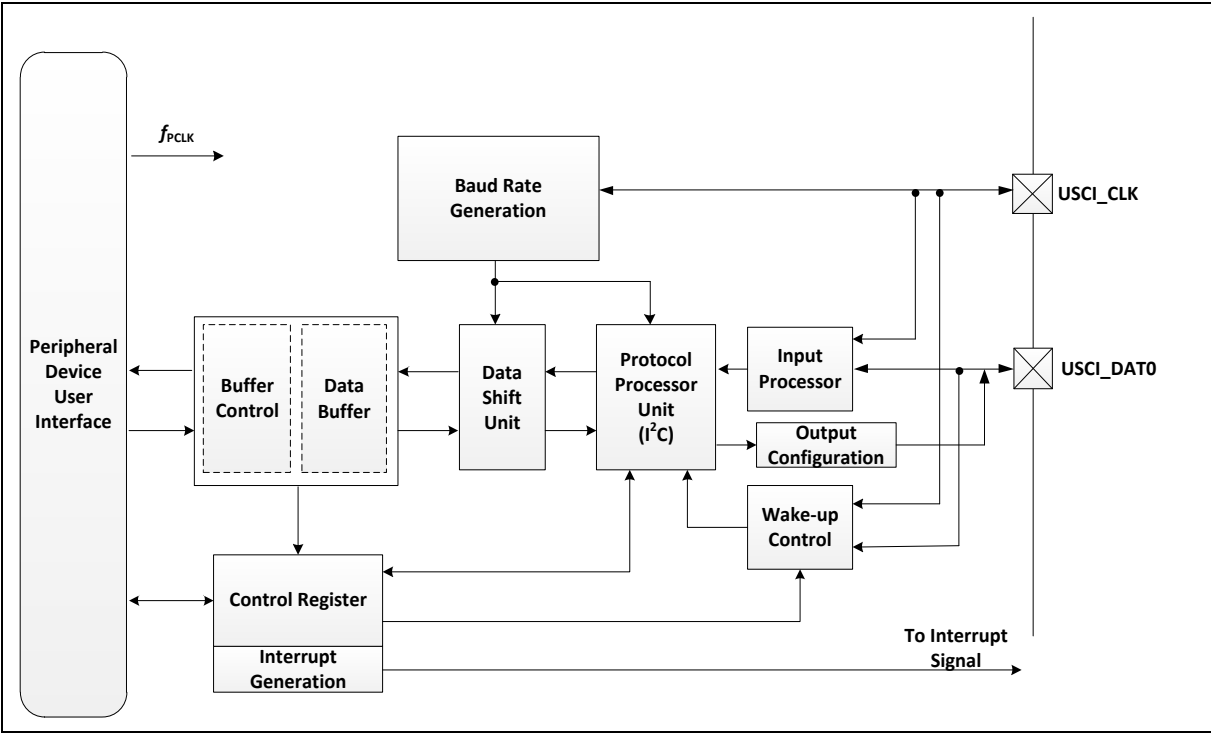


Figure 6.18-2 USCI I²C Mode Block Diagram

6.18.4 Basic Configuration

6.18.4.1 USCIO I²C Basic Configurations

The basic configurations of USCIO_I2C are as follows:

- Clock Source Configuration
 - Enable USCIO peripheral clock in USCIOCKEN (CLK_APBCLK1[8]).
 - Enable USCIO_I2C functi on UI2C_CTL[2:0]) register, UI2C_CTL[2:0]=3'b100
- Reset Configuration
 - Reset USCIO controller in USCIO_RST (SYS_IPRST2[8]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
USCIO	USCIO_CLK	PD.0	MFP3
		PB.12	MFP5
		PA.11	MFP6
		PE.2	MFP7
	USCIO_DAT0	PD.1	MFP3
		PB.13	MFP5
		PA.10	MFP6
		PE.3	MFP7

The basic configurations of USCI1_I2C are as follows:

Clock Source Configuration

- Enable USCI1 peripheral clock in USCI1CKEN (CLK_APBCLK1[9]).
- Enable USCI1_I2C functi on UI2C_CTL[2:0]) register, UI2C_CTL[2:0]=3'b100
- Reset Configuration
 - Reset USCI1 controller in USCI1RST (SYS_IPRST2[9]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
USCI1	USCI1_CLK	PB.8	MFP4
		PD.7, PE.12	MFP6
		PB.1	MFP8
	USCI1_DAT0	PB.7	MFP4
		PD.5, PE.10	MFP6
		PB.2	MFP8

The basic configurations of USCI2_I2C are as follows:

- Clock Source Configuration
 - Enable USCI2 peripheral clock in USCI2CKEN (CLK_APBCLK1[10]).
 - Enable USCI2_I2C functi on UI2C_CTL[2:0]) register, UI2C_CTL[2:0]=3'b100
- Reset Configuration
 - Reset USCI2 controller in USCI2RST (SYS_IPRST2[10]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
USCI2	USCI2_CLK	GPC.14	MFP7
		GPA.14	MFP8
		GPA.2	MFP11
		GPC.2	MFP13
	USCI2_DAT0	GPA.13	MFP8
		GPE.0	MFP10
		GPA.1	MFP11
		GPC.1	MFP13

6.18.5 Functional Description

6.18.5.1 START or Repeated START Signal

Figure 6.18-3 shows the typical I²C protocol. Normally, a standard communication consists of four parts:

- START or Repeated START signal generation
- Slave address and R/W bit transfer
- Data transfer

● STOP signal generation

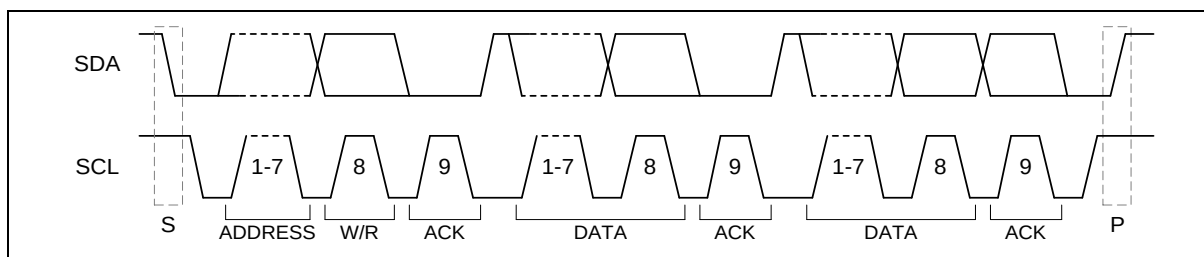


Figure 6.18-3 I²C Protocol

When the bus is free/idle, meaning no master device is engaging the bus (both SCL and SDA lines are high), a master can initiate a transfer by sending a START signal. A START signal, usually referred to as the “S” bit, is defined as a HIGH to LOW transition on the SDA line while SCL is HIGH. The START signal denotes the beginning of a new data transmission.

A Repeated START is not a STOP signal between two START signals and usually referred to as the “Sr” bit. The master uses this method to communicate with another slave or the same slave in a different transfer direction (e.g. from writing to a device to reading from a device) without releasing the bus idle flag.

6.18.5.2 STOP Signal

The master can terminate the communication by generating a STOP signal. A STOP signal, usually referred to as the “P” bit, is defined as a LOW to HIGH transition on the SDA line while SCL is HIGH. The section between STOP and START is called bus free.

Figure 6.18-4 shows the waveform of START, Repeat START and STOP.

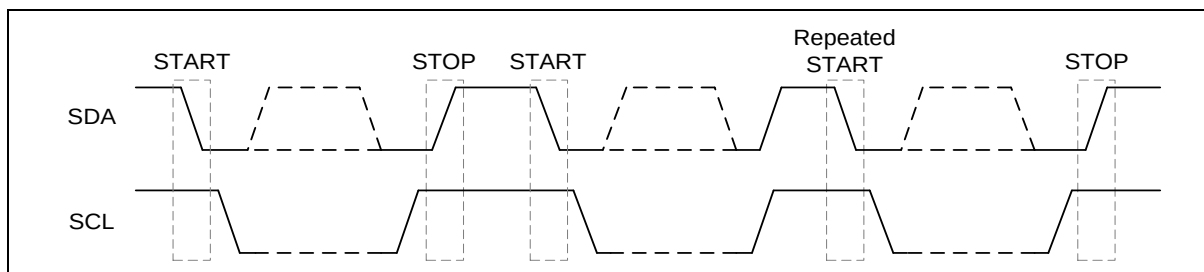


Figure 6.18-4 START and STOP Conditions

6.18.5.3 Slave Address Transfer

After a (Repeated) START condition, the master sends a slave address to identify the target device of the communication. The start address can comprise one or two address bytes (for 7-bit or for 10-bit addressing schemes). After an address byte, a slave sensitive to the transmitted address has to acknowledge the reception.

Therefore, the slave’s address can be programmed in the device, where it is compared to the received address. In case of a match, the slave answers with an acknowledge (SDA = 0). Slaves that are not targeted answer with a non-acknowledge (SDA = 1). In addition to the match of the programmed address, another address byte value has to be answered with an acknowledge if the slave is capable to handle the corresponding requests. The address byte 00H indicates a general call address that can be acknowledged.

In order to allow selective acknowledges for the different values of the address byte(s), the following control mechanism is implemented:

- If the GCFUNC bit (UI2C_PROTCTL [0]) is set the I²C port hardware will respond to General Call address (00H). Clear GC bit to disable general call function.

- The I²C port is equipped with one device address registers, UI2C_DEVADDRn (n = 0~1). In 7-bit address mode, the first 7 bits of a received first address byte are compared to the programmed slave address (UI2C_DEVADDRn [6:0]). If these bits match, the slave sends an acknowledge.
- For 10 bit addressing mode, if the slave address is programmed to 1111 0XXB, the XX bits are compared to the bits UI2C_DEVADDR [9:8] to check for address match and also sends an acknowledge when ADDR10EN (UI2C_PROTCTL [4]) is set. The slave waits for a second address byte compares it with UI2C_DEVADDR [7:0] and sends an acknowledge accordingly to cover the 10 bit addressing mode. The user has to take care about reserved addresses (refer to I²C specification for more detailed description). Only the address 1111 0XXB is supported. Under each of these conditions, bit SLASEL (UI2C_PROTSTS [14]) will be set when the addressing delivered a match. This SLASEL (UI2C_PROTSTS [14]) bit is cleared automatically by a (Repeated) START or STOP condition.
- The I²C port is equipped with multiple address recognition with one address mask registers I2C_ADDRMSKn (n = 0~1). When the bit in the address mask register is set to 1, it means the received corresponding address bit is "Don't care". If the bit is set to 0, it means the received corresponding register bit should be exactly the same as address register.

6.18.5.4 Data Transfer

When a slave receives a correct address with an R/W bit, the data will follow R/W bit specified to transfer. Each transferred byte is followed by an acknowledge bit on the 9th SCL clock cycle. If the slave signals a Not Acknowledge (NACK), the master can generate a STOP signal to abort the data transfer or generate a Repeated START signal and start a new transfer cycle.

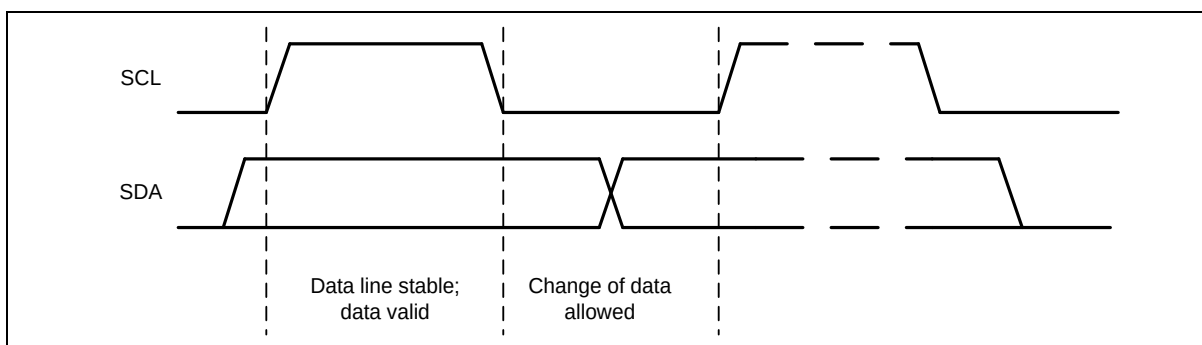


Figure 6.18-5 Bit Transfer on the I²C Bus

If the master receives data does Not Acknowledge (NACK) the slave, the slave releases the SDA line for the master to generate a STOP or Repeated START signal.

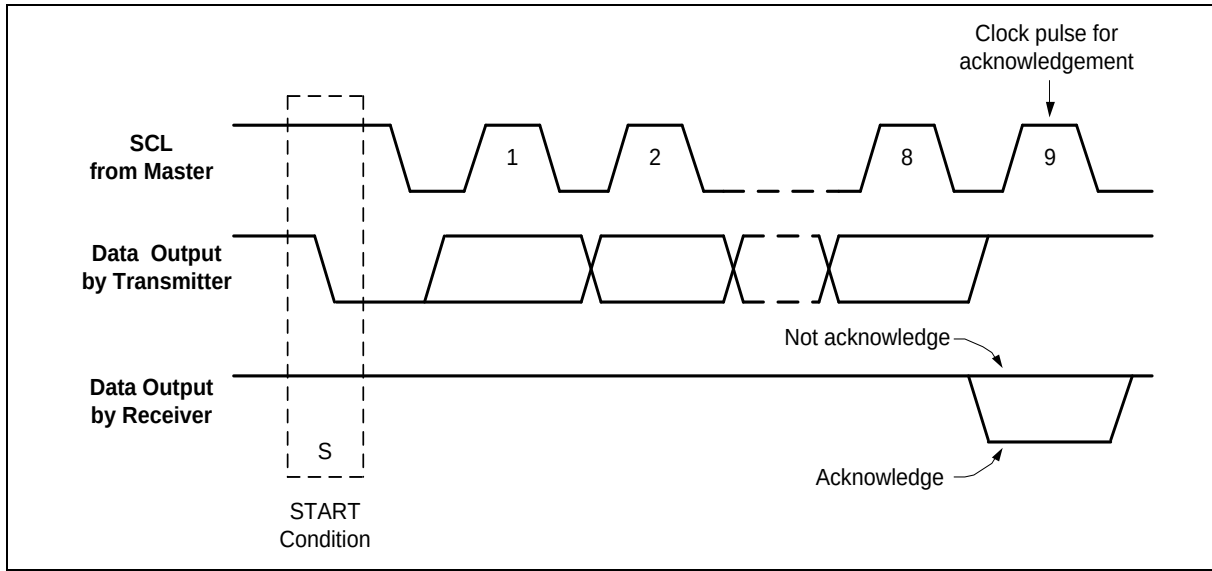


Figure 6.18-6 Acknowledge on the I²C Bus

6.18.5.5 Clock Baud Rate Bits

The data baud rate of I²C is determined by UI2C_BRGEN register when I²C is in Master Mode, and it is not necessary in a Slave mode. In the Slave mode, I²C will automatically synchronize it with any clock frequency from master I²C device. The bits RCLKSEL, SPCLKSEL, PDSCNT, and DSCNT define the baud rate setting:

- RCLKSEL (UI2C_BRGEN [0])
to define the input frequency f_{REF_CLK}
- SPCLKSEL (UI2C_BRGEN[3:2])
to define the multiple source of the sample clock f_{SAMP_CLK}
- PDSCNT (UI2C_BRGEN [9:8])
to define the length of a data sample time (division of f_{REF_CLK} by 1, 2, 3, or 4)
- DSCNT (UI2C_BRGEN [14:10])
to define the number of data sample time per bit time

The standard setting is given by RCLKSEL = 0 ($f_{REF\_CLK} = f_{PCLK}$), PTCLKSEL = 0 ($f_{PROT\_CLK} = f_{REF\_CLK}$) and SPCLKSEL = 2'b00 ($f_{SAMP\_CLK} = f_{DIV\_CLK}$). Under these conditions, the baud rate is given by:

$$f_{I2C} = f_{REF\_CLK} \times \frac{1}{CLKDIV + 1} \times \frac{1}{PDSCNT + 1} \times \frac{1}{DSCNT + 1}$$

In order to generate slower frequencies, additional divide-by-2 stages can be selected by PTCLKSEL = 1 ($f_{PROT\_CLK} = f_{REF\_CLK}/2$), leading to:

$$f_{I2C} = \frac{f_{REF\_CLK}}{2} \times \frac{1}{CLKDIV + 1} \times \frac{1}{PDSCNT + 1} \times \frac{1}{DSCNT + 1}$$

If SPCLKSEL = 2'b10 ($f_{SAMP\_CLK} = f_{SCLK}$), and RCLKSEL = 0 ($f_{REF\_CLK} = f_{PCLK}$), PTCLKSEL = 0 ($f_{PROT\_CLK} = f_{REF\_CLK}$). The baud rate is given by:

$$f_{I2C} = f_{REF\_CLK} \times \frac{1}{CLKDIV + 1} \times \frac{1}{2} \times \frac{1}{PDSCNT + 1} \times \frac{1}{DSCNT + 1}$$

6.18.5.6 Byte Stretching

If a device is selected as master/slave transmit mode and should transmit a data byte but the transmit buffer TXDAT does not contain valid data to be transmitted, the device ties down SCL = 0 at the end of the previous acknowledge bit. The waiting period is finished if software writes 1 to PTRG (UI2C_PROTCTL [5]).

6.18.5.7 Multi-master Arbitration

In some applications, there are two or more masters on the same I²C bus to access slaves, and the masters may transmit data simultaneously. The I²C supports multi-master by including collision detection and arbitration to prevent data corruption.

If two masters sometimes initiate I²C command at the same time, the arbitration procedure determines which master wins and can continue with the command. Arbitration is performed on the SDA signal while the SCL signal is high. Each master checks if the SDA signal on the bus corresponds to the generated SDA signal. If the SDA signal on the bus is low but it should be high, then this master has lost arbitration. Master I²C device that has lost arbitration can generate SCL pulses until the byte ends and must then release the bus and go into slave mode. The arbitration procedure can continue until all the data is transferred. This means that in multi-master system each I²C master must monitor the I²C bus for collisions and act accordingly. Figure 6.18-7 describes master1 data and master2 data are compete arbitration.

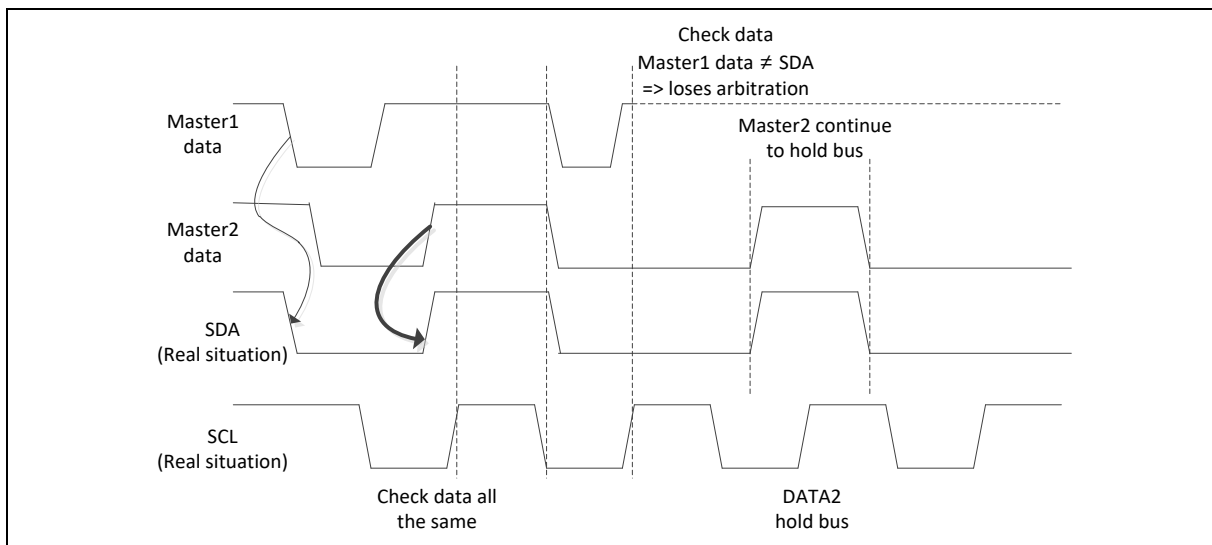


Figure 6.18-7 Arbitration Lost

In this case, during the address and data transmission, the master transmitter checks at the rising edge of SCL for each data bit if the value it is sending is equal to the value read on the SDA line. If yes, master can hold bus continuously. If this is not the case (transmitted value = 1, value read = 0), the master has lost the transmit arbitration. This is indicated by interrupt flag ARBLOIF (UI2C_PROTSTS [11]) and can generate a protocol interrupt if enabled by ARBLOIEN (UI2C_PROTIEN [4]).

When the transmit arbitration has been lost, the software has to initialize the complete frame again, starting with the first address byte together with the START condition for a new master transmit attempt. Arbitration also takes place for the ACK bit. If master arbitration lost and match the device address, then master will turn to slave.

6.18.5.8 Transmission Chain

The I²C bus protocol requiring a kind of in-bit-response during the arbitration phase and while a slave is transmitting, the resulting loop delay of the transmission chain can limit the reachable maximal baud rate, strongly depending on the bus characteristics (bus load, module frequency, etc.).

The shift clock SCL is generated by the master device, output on the wire, then it passes through the

input stage and the input filter. Now, the edges can be detected and the SDA data signal can be generated accordingly. The SDA signal passes through the output stage and the wire to the master receiver part. There, it passes through the input stage and the input filter before it is sampled.

This complete loop has to be finished (including all settling times to obtain stable signal levels) before the SCL signal changes again. The delays in this path have to be taken into account for the calculation of the baud rate as a function of f_{PCLK} and f_{PROT_CLK} . It is suggested that user adopt f_{PCLK} .

6.18.5.9 Non-Acknowledge and Error Conditions

In case of a non-acknowledge (NACKIF (UI2C_PROTSTS [10])) or an error (ERRIF (UI2C_PROTSTS [12])), no further transmission will take place. User software doesn't invalidate the transmit buffer and disable transmissions, before configuring the transmission (by writing TXDAT) again with appropriate values to react on the previous event.

6.18.5.10 I²C Protocol Interrupt Events

The following protocol-related events are generated in I²C mode and can lead to a protocol interrupt.

Please note that the bits in register UI2C_PROTSTS are not all automatically cleared by hardware and have to be cleared by software in order to monitor new incoming events.

- START condition received at a correct position in a frame (STARIF (UI2C_PROTSTS [8]))
- STOP condition transferred at a correct position in a frame (STORIF (UI2C_PROTSTS [9]))
- Master arbitration lost (ARBLOIF (UI2C_PROTSTS [11]))
- Slave read requested (SLAREAD (UI2C_PROTSTS [15]))
- Acknowledge received (ACKIF (UI2C_PROTSTS [13]))
- Non-acknowledge received (NACKIF (UI2C_PROTSTS [10]))
- START condition not at the expected position in a frame (ERRIF (UI2C_PROTSTS [12]))
- STOP condition not at the expected position in a frame (ERRIF (UI2C_PROTSTS [12]))

6.18.5.11 Operating the I²C

To operate the I²C protocol, the following issues have to be considered:

Select I²C Mode

It is recommended to configure all parameters of the I²C that do not change during run time while FUNMODE (UI2C_CTL [2:0]) = 000B. The I²C control flow has to be done while FUNMODE (UI2C_CTL [2:0]) = 000B to avoid unintended edges of the input signals and the I²C mode can be enabled by FUNMODE (UI2C_CTL [2:0]) = 100B afterwards.

Step 1. Set FUNMODE (UI2C_CTL [2:0]) = 000B

Step 2. Set FUNMODE (UI2C_CTL [2:0]) = 100B

Pin Connections

The pins used for SDA and SCL have to be set to open-drain mode by USCI controller to support the wired-AND structure of the I²C bus lines.

Note: The step to enable the alternate output port functions should only be done after the I²C mode is enabled, to avoid unintended spikes on the output.

Bit Timing Configuration

In standard mode (100 kBit/s) a minimum module frequency of 2 MHz is necessary, whereas in fast mode (400 kBit/s) a minimum of 10 MHz is required. Additionally, if the digital filter stage should be used to eliminate spikes up to 50 ns, a filter frequency of 20 MHz is necessary. There could be an uncertainty

in the SCL high phase timing of maximum $1/f_{\text{PROT\_CLK}}$ if another I²C participant lengthens the SCL low phase on the bus. Note that the SCL maximum frequency is $f_{\text{SAMP\_CLK}}/2$ and the SPCLKSEL (UI2C_BRGEN [3:2]) must be set to 0 for selecting $f_{\text{SAMP\_CLK}} = f_{\text{DIV\_CLK}}$.

Data Format Configuration

The data format has to be configured for 8 data bits (DWIDTH (UI2C_LINECTL [11:8]) = 8), and MSB shifted first (LSB (UI2C_LINECTL [0]) = 0). As a result, UI2C_LINECTL has to be set to 0x800.

Control Flow

The on-chip I²C ports support three operation modes, Master, Slave, and General Call Mode.

In a given application, I²C port may operate as a master or as a slave. In Slave mode, the I²C port hardware looks for its own slave address and the general call address. If one of these addresses is detected, and if the slave is willing to receive or transmit data from/to master (by setting the AA bit), acknowledge pulse will be transmitted out on the 9th clock, hence an interrupt is requested on both master and slave devices if interrupt is enabled. When the microcontroller wishes to become the bus master, hardware waits until the bus is free before entering Master mode so that a possible slave action is not be interrupted. If address arbitration is lost in Master mode, I²C port switches to Slave mode immediately and can detect its own slave address in the same serial transfer.

To control the I²C bus transfer in each mode, user needs to set UI2C_PROTCTL, UI2C_PROTIEN, TXDAT registers according to current status of UI2C_PROTSTS register. In other words, for each I²C bus action, user needs to check current status by UI2C_PROTSTS register, and then set UI2C_PROTCTL, UI2C_PROTIEN, TXDAT registers to take bus action. Finally, check the response status by UI2C_PROTSTS.

The bits, STA, STO and AA in UI2C_PROTCTL register are used to control the next state of the I²C hardware after interrupt signal is cleared. Upon completion of the new action, a new status will be updated in UI2C_PROTSTS register will be set. If the I²C interrupt control bit of UI2C_PROTIEN is set, appropriate action or software branch of the new status can be performed in the Interrupt service routine.

Figure 6.18-8 shows the current I²C STARIF (UI2C_PROTSTS [8]) is set to 1 by hardware, and then set TXDAT = SLA+W (Slave address + Write bit), (PTRG, STA, STO, AA) = (1, 0, 0, x) to send the address to I²C bus, and write 1 to STARIF (UI2C_PROTSTS [8]) to clear flag. If a slave on the bus matches the address and response ACK, the UI2C_PROTSTS will be updated by ACKIF (UI2C_PROTSTS [13]) setting.

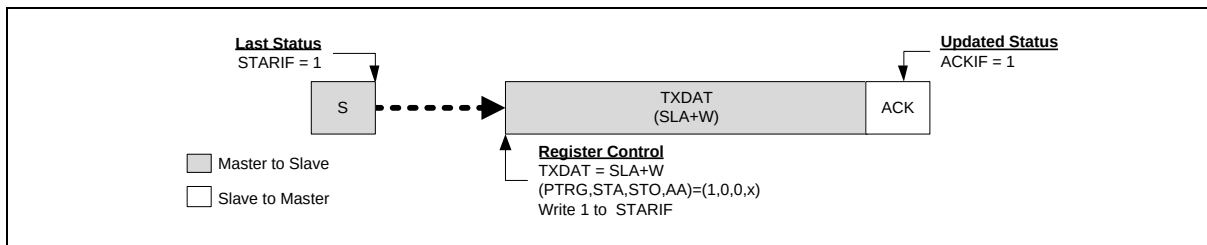


Figure 6.18-8 Control I²C Bus according to Current I²C Status

Data Transfer on the I²C Bus

Figure 6.18-9 shows a master transmits data to slave. A master addresses a slave with a 7-bit address and 1-bit write index to denote that the master wants to transmit data to the slave. The master keeps transmitting data after the slave returns acknowledge to the master.

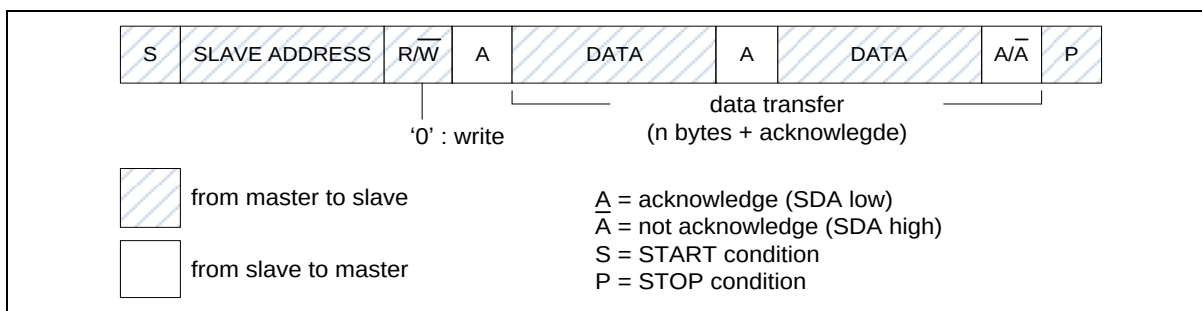


Figure 6.18-10 shows a master read data from slave. A master addresses a slave with a 7-bit address and 1-bit read index to denote that the master wants to read data from the slave. The slave will start transmitting data after the slave returns acknowledge to the master.

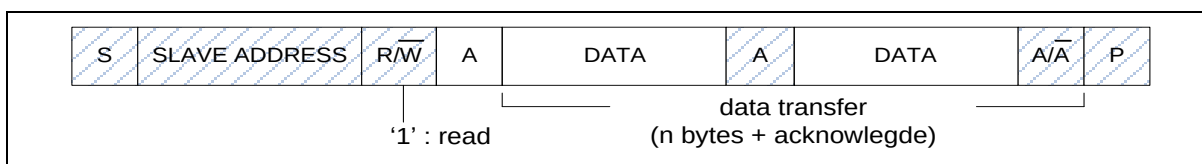


Figure 6.18-11 shows a master transmits data to slave by 10-bit address. A master addresses a slave with a 10-bit address. First byte contains 10-bit address indicator (5'b11110) and 2-bit address with write index, second byte contains 8-bit address. The master keeps transmitting data after the second byte end. Note that 7-bit and 10-bit address device can work on the same bus.

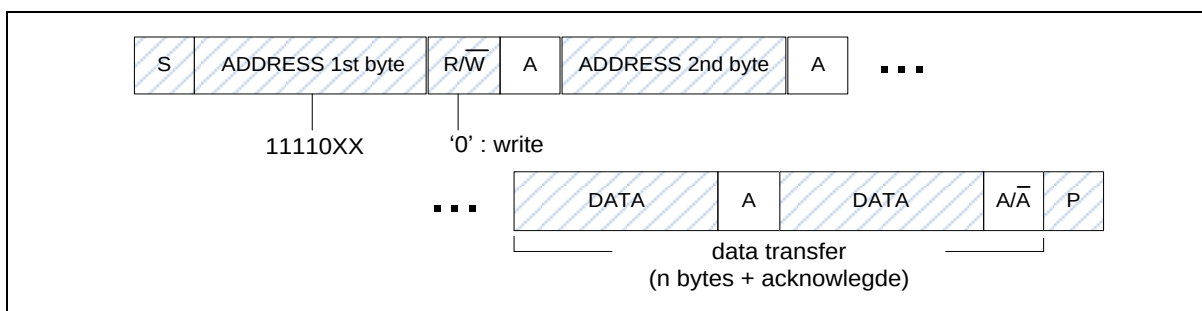
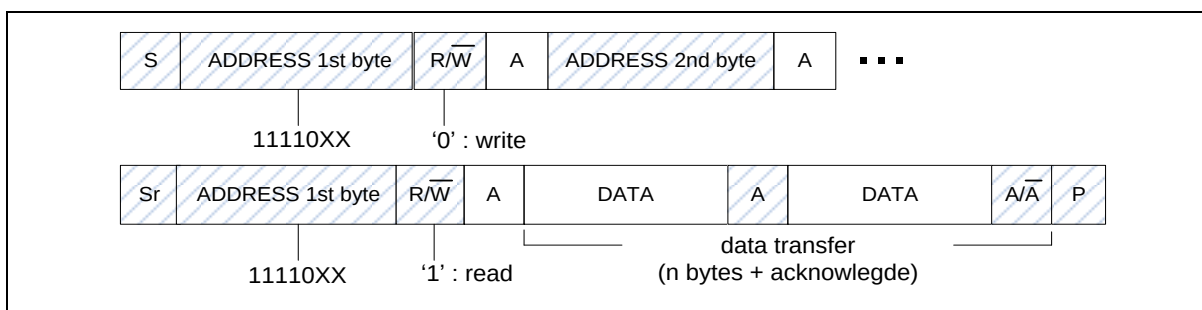


Figure 6.18-12 shows a master read data from slave by 10-bit address. A master addresses a slave with a 10-bit address. First master transmits 10-bit address to slave, after that master transmits first byte with read index. The slave will start transmitting data after the first byte with read index.



In Figure 6.18-13 and Figure 6.18-14, all possible protocols for I²C master are shown. User needs to follow proper path of the flow to implement required I²C protocol.

In other words, user can send a START signal to bus and I²C will be in Master Transmitter mode (Figure 6.18-13) or Master receiver mode (Figure 6.18-14) after START signal has been sent successfully and new status register would be set STARIF (UI2C_PROTSTS [8]). Followed by START signal, user can send slave address, read/write bit, data and Repeat START, STOP to perform I²C protocol.

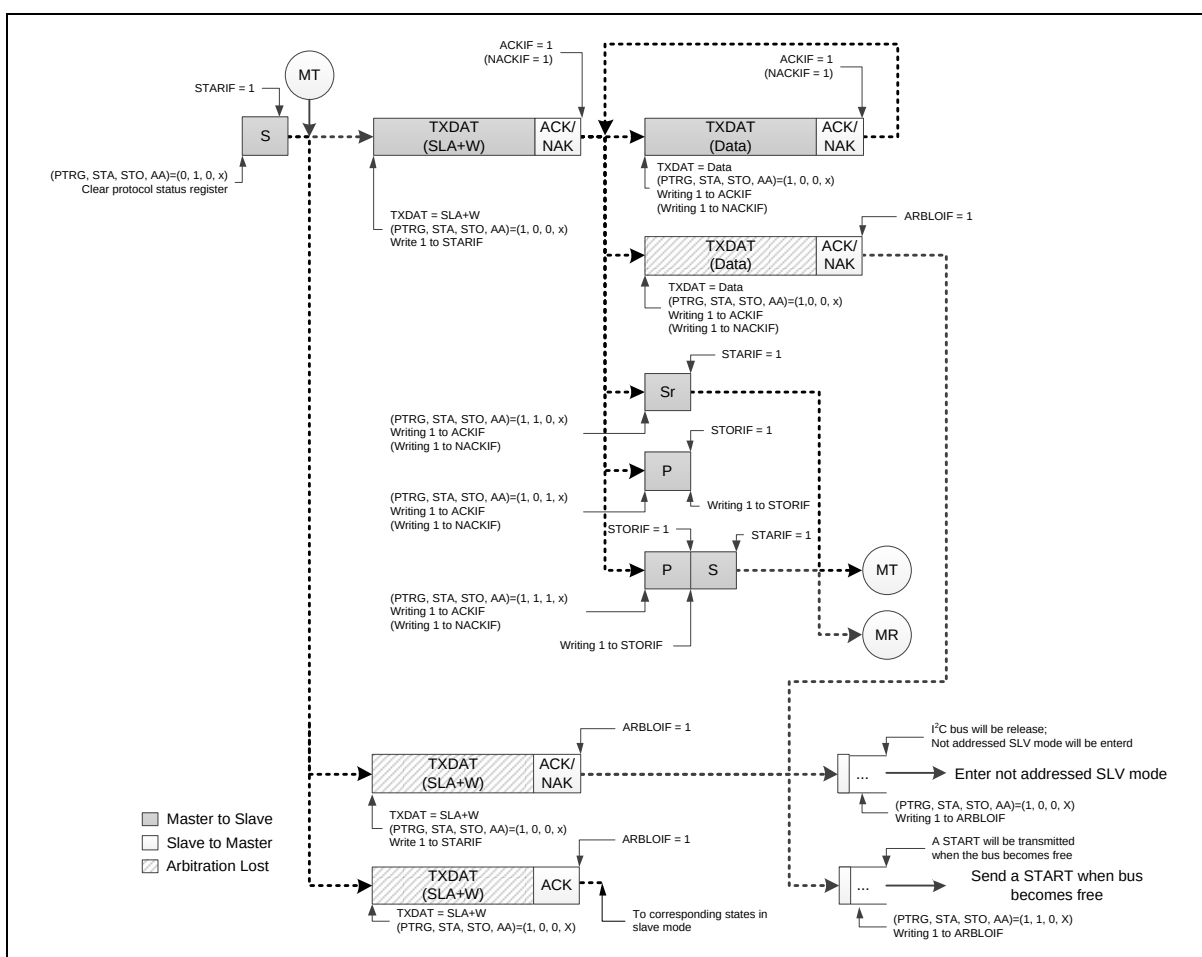


Figure 6.18-13 Master Transmitter Mode Control Flow with 7-bit Address

Figure 6.18-14 Master Receiver Mode Control Flow with 7-bit Address

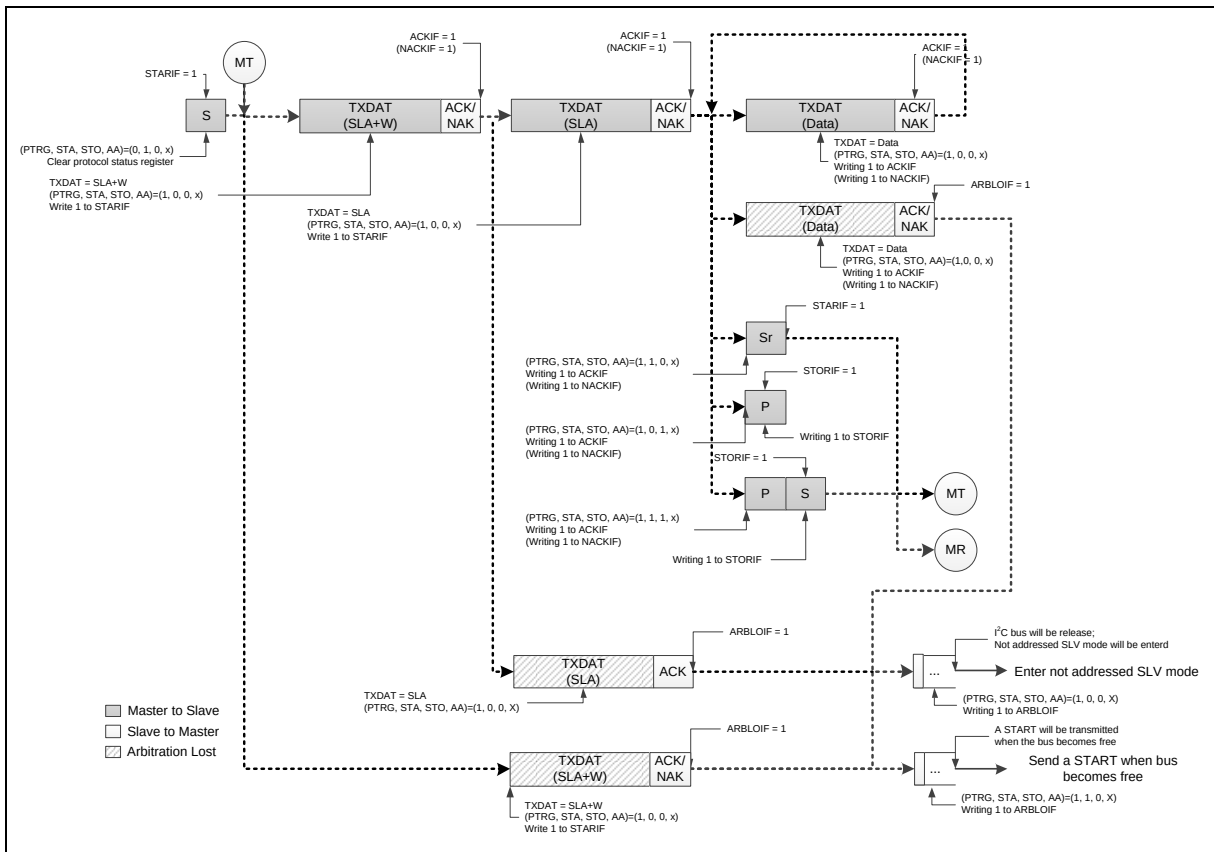


Figure 6.18-15 Master Transmitter Mode Control Flow with 10-bit Address

If the I²C is in Master mode and gets arbitration lost, the bit of ARBLOIF (UI2C_PROTSTS [11]) will be set. User may writing 1 to ARBLOIF (UI2C_PROTSTS [11]) and set (PTRG, STA, STO, AA) = (1, 1, 0, X) to send START to re-start Master operation when bus becomes free. Otherwise, user may write 1 to ARBLOIF (UI2C_PROTSTS [11]) and set (PTRG, STA, STO, AA) = (1, 0, 0, X) to release I²C bus and enter not addressed Slave mode.

When reset, I²C is not addressed and will not recognize the address on I²C bus. User can set device address by UI2C_DEVADDRn and set (PTRG, STA, STO, AA) = (1, 0, 0, 1) to let I²C recognize the address sent by master. Figure 6.18-17 shows all the possible flow for I²C in Slave mode. Users need to follow a proper flow as shown in Figure 6.18-17 to implement their own I²C protocol.

The I²C controller supports two slave address match flags, ADMAT0 and ADMAT1 on UI2C_ADMAT[1:0] register. Every control register represents which address is used and set 1 to inform software.

Note: During I²C communication, the SCL clock will be released when writing '1' to PTRG (UI2C PROTCTL [5]) in Slave mode.

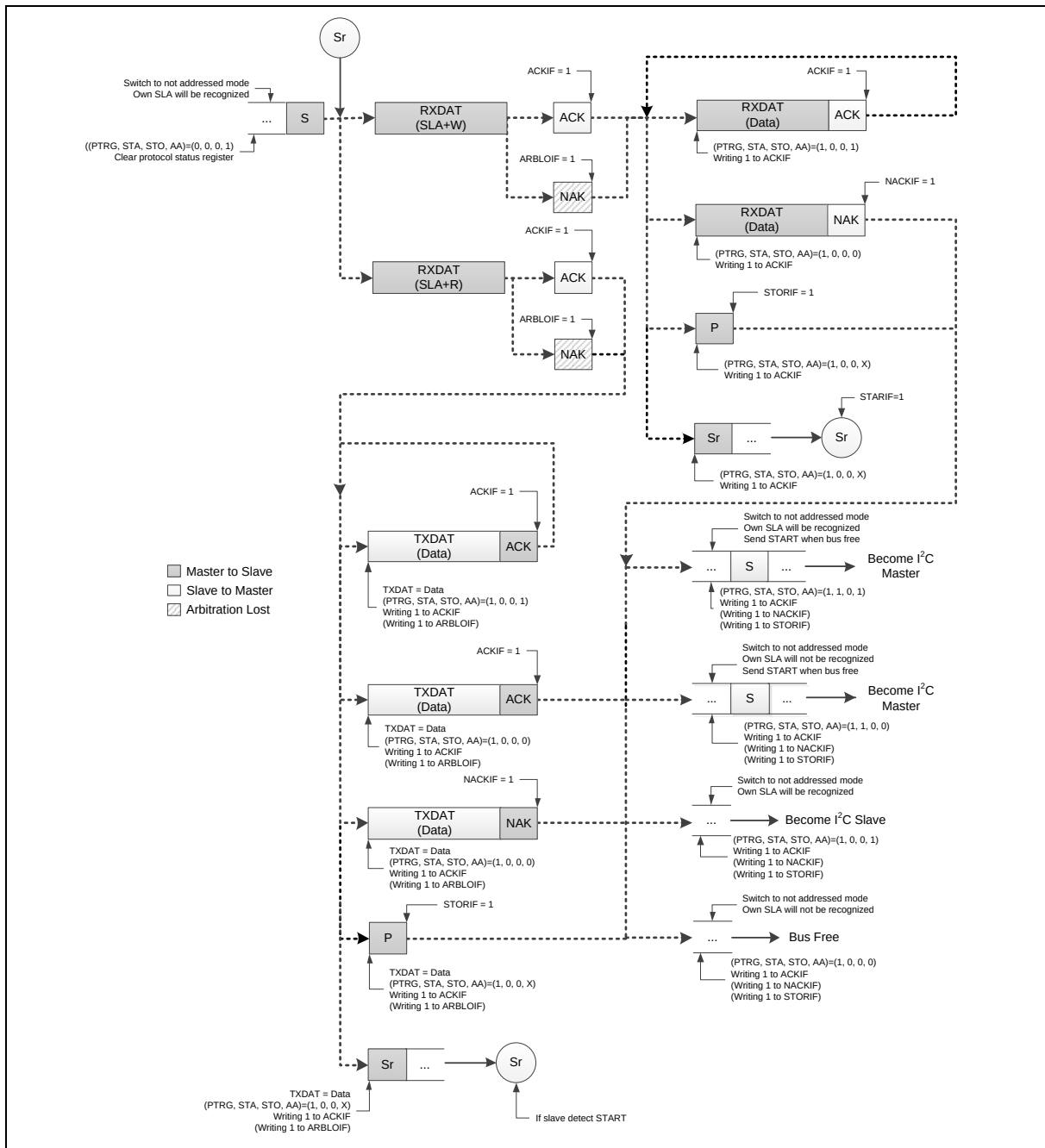


Figure 6.18-17 Slave Mode Control Flow with 7-bit Address

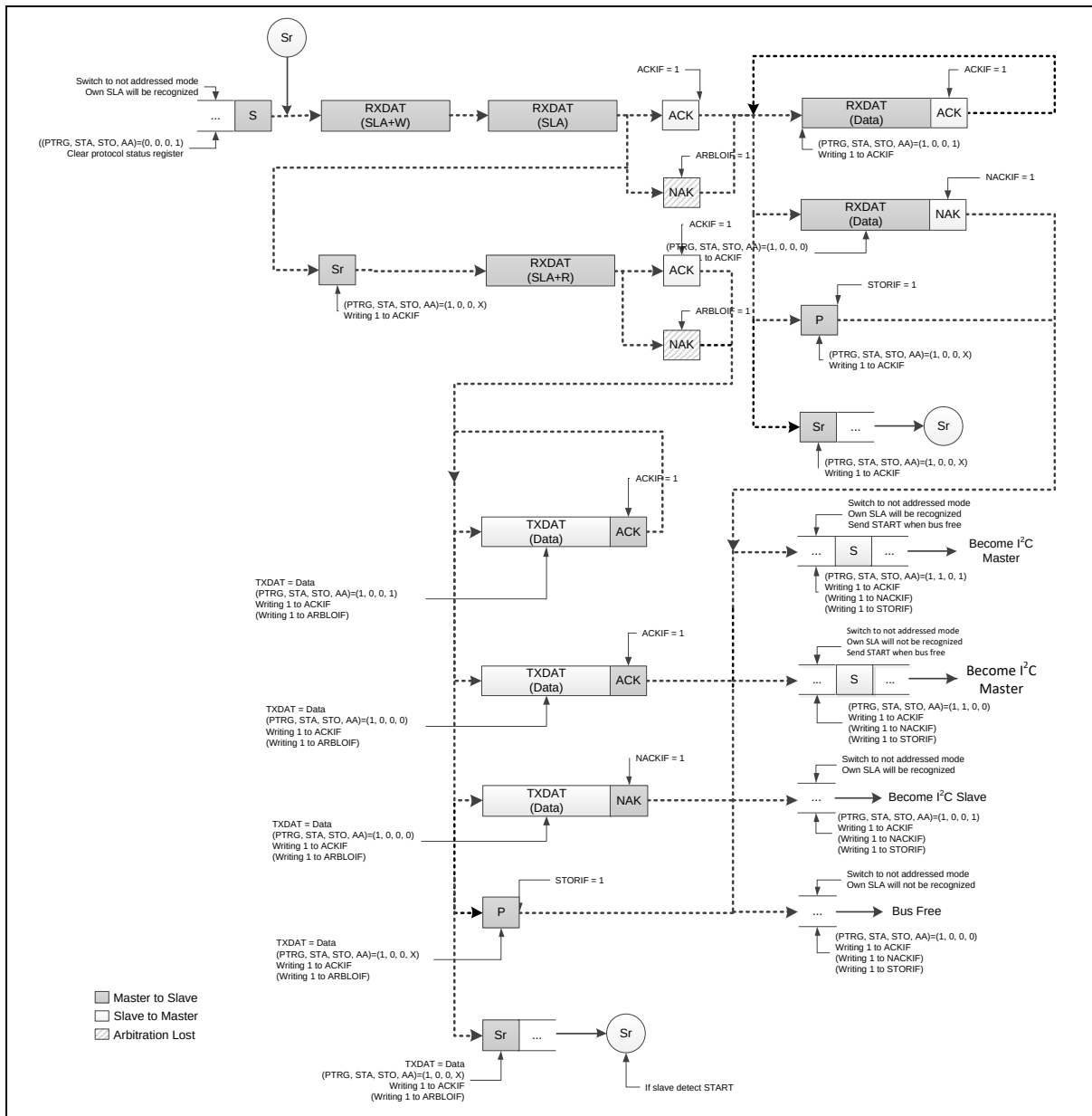


Figure 6.18-18 Slave Mode Control Flow with 10-bit Address

If I²C is still transmitting and receiving data in addressed Slave mode but got a STOP or Repeat START, the register STORIF (UI2C_PROTSTS [9]) or STARIF (UI2C_PROTSTS [8]) will be set. User could follow the action for NACKIF (UI2C_PROTSTS [10]) as shown in Figure 6.18-8 when STARIF (UI2C_PROTSTS [8]) is set.

Note: After slave gets interrupt flag of NACKIF (UI2C_PROTSTS [10]) and start/stop symbol including STARIF (UI2C_PROTSTS [8]) and STORIF (UI2C_PROTSTS [9]), slave can switch to not address mode and own SLA will not be recognized. If setting this interrupt flag, slave will not receive any I²C signal or address from master. At this status, I²C should be reset by setting FUNMODE (UI2C_CTL [2:0]) = 000B to leave this status.

General Call (GC) Mode

If the GCFUNC bit (UI2C_PROTCTL [0]) is set, the I²C port hardware will respond to General Call address (00H). User can clear GC bit to disable general call function. When the GC bit is set and the

I²C in slave mode, it can receive the general call address by 0x00 after master send general call address to I²C bus, and then it also will follow protocol status register.

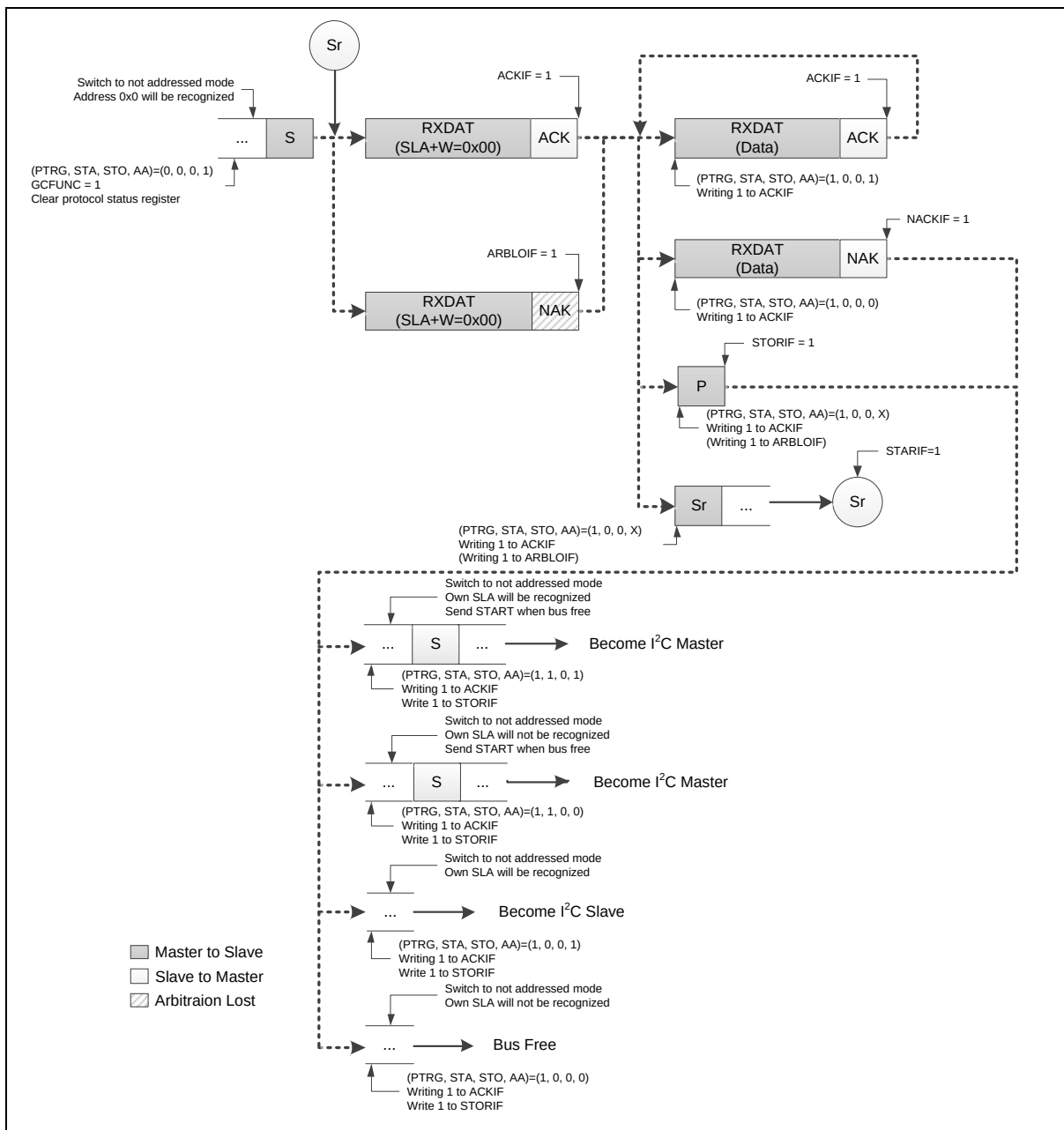


Figure 6.18-19 GC Mode with 7-bit Address

If I²C is still receiving data in GC mode but got a STOP or Repeat START, the STORIF (UI2C_PROTSTS [9]) or STARIF (UI2C_PROTSTS [8]) will be set. User could follow the action for NACKIF (UI2C_PROTSTS [10]) in Figure 6.18-19 when got STORIF (UI2C_PROTSTS [9]) or STARIF (UI2C_PROTSTS [8]) is set.

Note: After slave gets interrupt flag of NACKIF (UI2C_PROTSTS [10]) and start/stop symbol including STARIF (UI2C_PROTSTS [8]) and STORIF (UI2C_PROTSTS [9]), slave can switch to not address mode and own SLA will not be recognized. If setting this interrupt flag, slave will not receive any I²C signal or address from master. At this time, I²C controller should be reset by setting FUNMODE (UI2C_CTL [2:0]) = 000B to leave this status.

Protocol Functional Description

Monitor Mode

When I²C enters monitor mode, this device always returns NACK to master after each frame reception even address matching. Moreover, this device will store any receive data including address, command code, and data.

Interrupt in Monitor Mode

All interrupts will occur as normal process when the MONEN (UI2C_PROTCTL [9]) is set. Note that the first interrupt will occur when initial START, it not the same as I²C slave, but the other interrupts are the same.

Subsequent to the address-match detection, interrupts will be generated after each data byte is received as slave mode control flow, or after each byte that the module believes it has transmitted for a slave-read transfer. In this second case, the data register will actually contain data transmitted by some other slave on the bus which was actually addressed by the master. If user wants to watch other device, user can set address mask and monitor.

If the monitor has not had time to respond to interrupt, the SCL signal will be pulled to low when SCLOUTEN (UI2C_PROTCTL [8]) is set to 1. User must set PTRG (UI2C_PROTCTL [5]) to release bus when SCLOUTEN (UI2C_PROTCTL [8]) is set to 1. If SCLOUTEN (UI2C_PROTCTL [8]) is not set to 1, user doesn't need to set PTRG (UI2C_PROTCTL [5]) to 1.

When device address match, but the device response NACK, this address will be received into buffer and NACK interrupt will be generated.

Following all of these interrupts, the processor may read the data register to see what was actually transmitted on the bus.

Loss of Arbitration in Monitor Mode

In monitor mode, the I²C module will not be able to respond to a request for information by the bus master or issue an ACK. Some other slave on the bus will respond instead. Software should be aware of the fact that the module is in monitor mode and should not respond to any loss of arbitration state that is detected.

Programmable Setup and Hold Time

In order to guarantee a correct data setup and hold time, the timing must be configured. By programming HTCTL (UI2C_TMCTL[24:16]) to configure hold time and STCTL (UI2C_TMCTL[8:0]) to configure setup time.

The delay timing refer peripheral clock (PCLK). When device stretch master clock, the setup and hold time configuration value will not affected by stretched.

User should focus the limitation of setup and hold time configuration, the timing setting must follow I²C protocol. Once setup time configuration greater than design limitation, that means if setup time setting make SCL output less than three PCLKs, I²C controller can't work normally due to SCL must sample three times. And once hold time configuration greater than I²C clock limitation, I²C will occur bus error. Suggest that user calculate suitable timing with baud rate and protocol before setting timing. Table 6.18-1 shows the relationship between I²C baud rate and PCLK, the number of table represent one clock duty contain how many PCLKs. Setup and hold time configuration even can program some extreme values in the design, but user should follow I²C protocol standard.

I ² C Baud Rate PCLK	100k	200k	400k	800k	1200k
12 MHz	120	60	30	15	10
24 MHz	240	120	60	30	20
48 MHz	480	240	120	60	40
72 MHz	720	360	180	90	60

Table 6.18-1 Relationship between I²C Baud Rate and PCLK

For setup time wrong adjustment example, assuming one SCL cycle contains ten PCLKs and set STCTL (UI2C_TMCTL[8:0]) to 3 that stretch three PCLKs for setup time setting. The setup time setting limitation: $ST_{limit} = (UI2C_BRGEN[25:16]+1) - 6$.

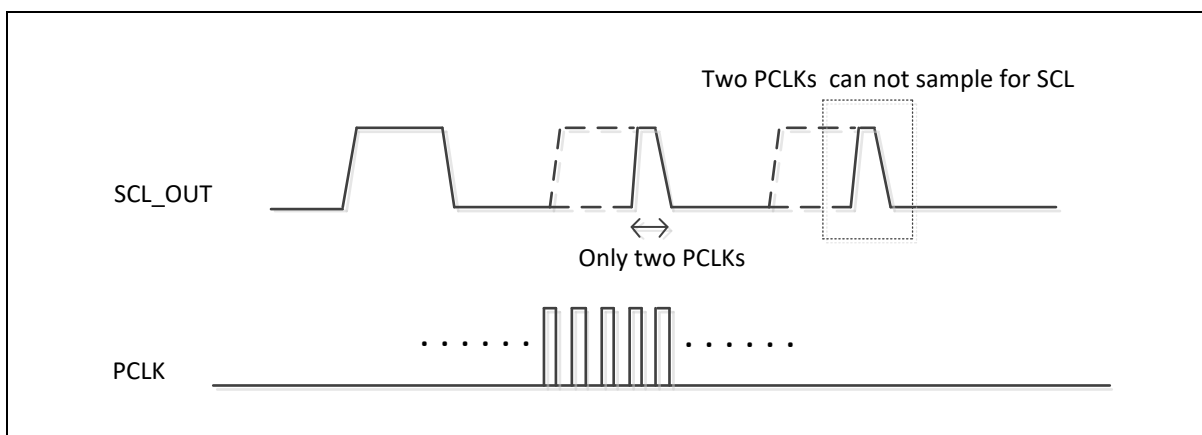


Figure 6.18-20 Setup Time Wrong Adjustment

For hold time wrong adjustment example, use I²C Baud Rate = 1200k and PCLK = 72 MHz, the SCL high/low duty = 60 PCLK. When HTCTL (UI2C_TMCTL[24:16]) is set to 63 and STCTL (UI2C_TMCTL[8:0]) is set to 0, then SDA output delay will over SCL high duty and cause bus error. The hold time setting limitation: $HT_{limit} = (UI2C_BRGEN[25:16]+1) - 9$.

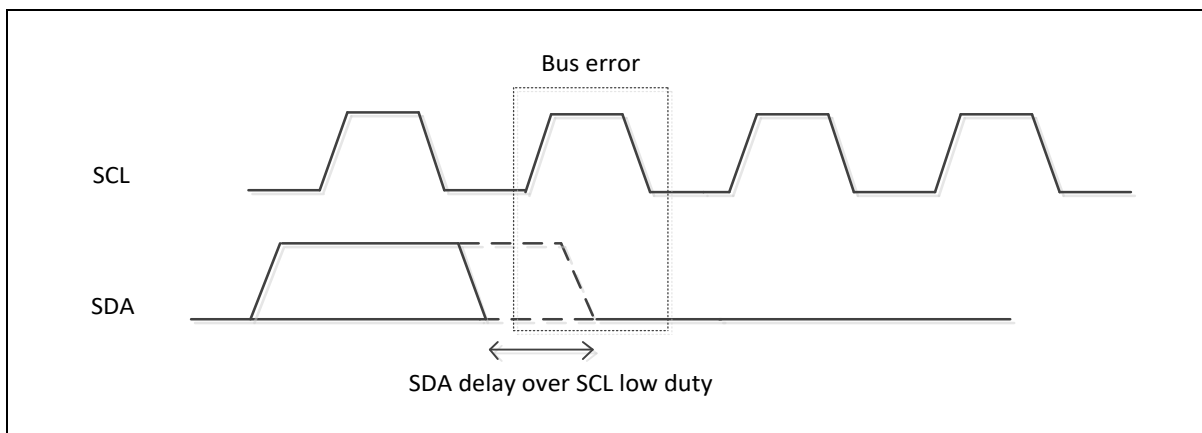


Figure 6.18-21 Hold Time Wrong Adjustment

I²C Time-out Function

There is a 10 bits time-out counter TOCNT (UI2C_PROTCTL [25:16]) which can be used to deal with the I²C bus hang-up. If the time-out counter is enabled, the counter starts up counting until it equals

TOCNT (UI2C_PROTCTL [25:16]) and generates I²C interrupt to CPU or stops counting by clearing TOIEN (UI2C_PROTIEN [0]) to 0 or setting all I²C interrupt signal (ACKIF, ERRIF, ARBLOIF, NACKIF, STORIF, STARIF). User may write 1 to clear TOIF (UI2C_PROTSTS[5]) to 0. When time-out counter is enabled, writing 1 to the TOIF will reset counter and re-start up counting after TOIF is cleared. Refer to Figure 6.18-22 for the time-out counter TOCNT (UI2C_PROTCTL [25:16]). $T_{TOCNT} = (TOCNT (UI2C\_PROTCTL [25:16]) + 1) \times 32 (5\text{-bit}) \times T_{PCLK}$. Note that the time counter clock source TMCNTSRC (UI2C_BRGEN [5]) must be set as 0.

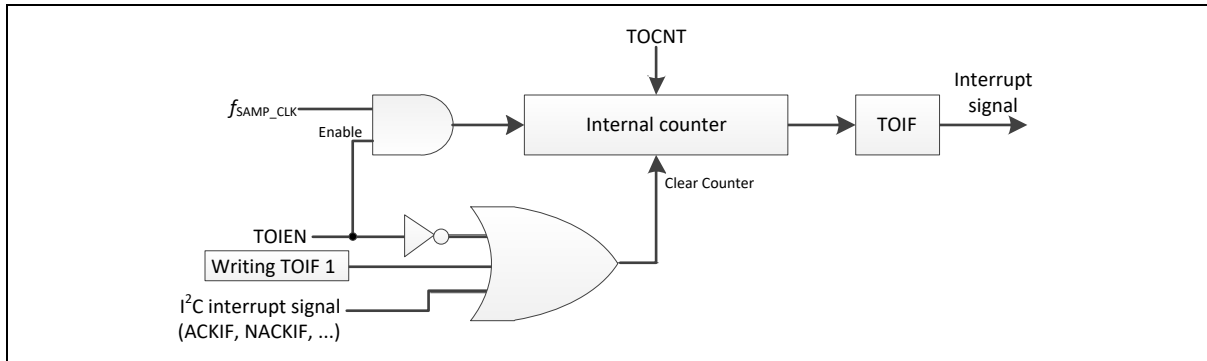


Figure 6.18-22 I²C Time-out Count Block Diagram

Wake-up Function

When chip enters Power-down mode and sets WKEN (WKCTL[0]) to 1, other I²C master can wake up the chip by addressing the I²C device. User must configure the related setting before entering sleep mode. The ACK bit cycle of address match frame is done in power-down. The controller will stretch the SCL to low when the address is matched the device's address and the ACK cycle done. The SCL is stretched until the bit is clear by user. If the frequency of SCL is low speed and the system has wakeup from address match frame, the user shall check this bit to confirm this frame has transaction done and then to do the wake-up procedure. Therefore, when the chip is woken up by address match with one of the device address register (UI2C_DEVADDRn), the user shall check the WKAKDONE (UI2C_PROTSTS [16]) bit is set to 1 to confirm if the address wakeup frame has been done. The WKAKDONE bit indicates that the ACK bit cycle of address match frame is done in power-down. The controller will stretch the SCL to low when the address is matched the device's slave address and the ACK cycle done. The SCL is stretched until the WKAKDONE bit is clear by user. If the frequency of SCL is low speed and the system has wakeup from address match frame, the user shall check this bit to confirm this frame has transaction done and then to do the wake-up procedure. Note that user must clear WKUPIF after clearing the WKAKDONE bit to 0.

The WRSTSWK (UI2C_PROTSTS [17]) bit records the Read/Write command on the address match wake-up frame. The user can use read this bit's status to prepare the next transmitted data (WRSTSWK = 0) or to wait the incoming data (WRSTSWK = 1) can be stored in time after the system is woken up by the address match frame.

When system is woken up by other I²C master device, WKF (UI2C_WKSTS [0]) is set to indicate this event. User needs write "1" to clear this bit.

Example for Random Read on EEPROM

The following steps are used to configure the USCI0_I2C related registers when using I²C protocol to read data from EEPROM.

1. Set USCI0_I2C the multi-function pin as SCL and SDA pins. The multi-function configuration reference Basic Configuration.
2. Enable USCI0 APB clock. The multi-function configuration reference Basic Configuration.
3. Set USCI0RST=1 to reset USCI controller then set USCI0RST=0 let USCI controller to normal operation. The reset controller configuration reference Basic Configuration.

4. Set FUNMODE =100 to enable USCI0_I2C controller in the “UI2C_CTL” register.
5. Give USCI0_I2C clock a divided register value for USCI0_I2C clock rate in the “UI2C_BRGEN”.
6. Enable system I2C0 IRQ in system “NVIC” control register.
7. Set ACKIEN, ERRIEN, ARBLOIEN, NACKIEN, STORIEN, STARIEN, and TOIEN to enable I²C Interrupt in the “UI2C_PROTIEN” register.
8. Set USCI address registers “UI2C_ADDR0 ~ UI2C_ADDR1”.

Random read operation is one of the methods of access EEPROM. The method allows the master to access any address of EEPROM space. Figure 6.18-23 shows the EEPROM random read operation.

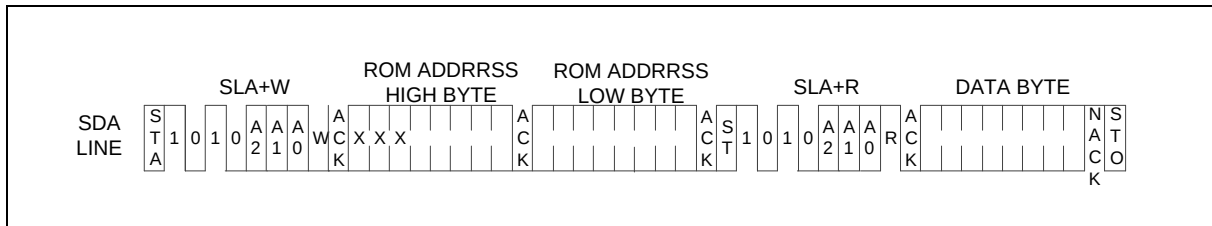


Figure 6.18-23 EEPROM Random Read

Figure 6.18-24 shows how to use I²C controller to implement the protocol of EEPROM random read.

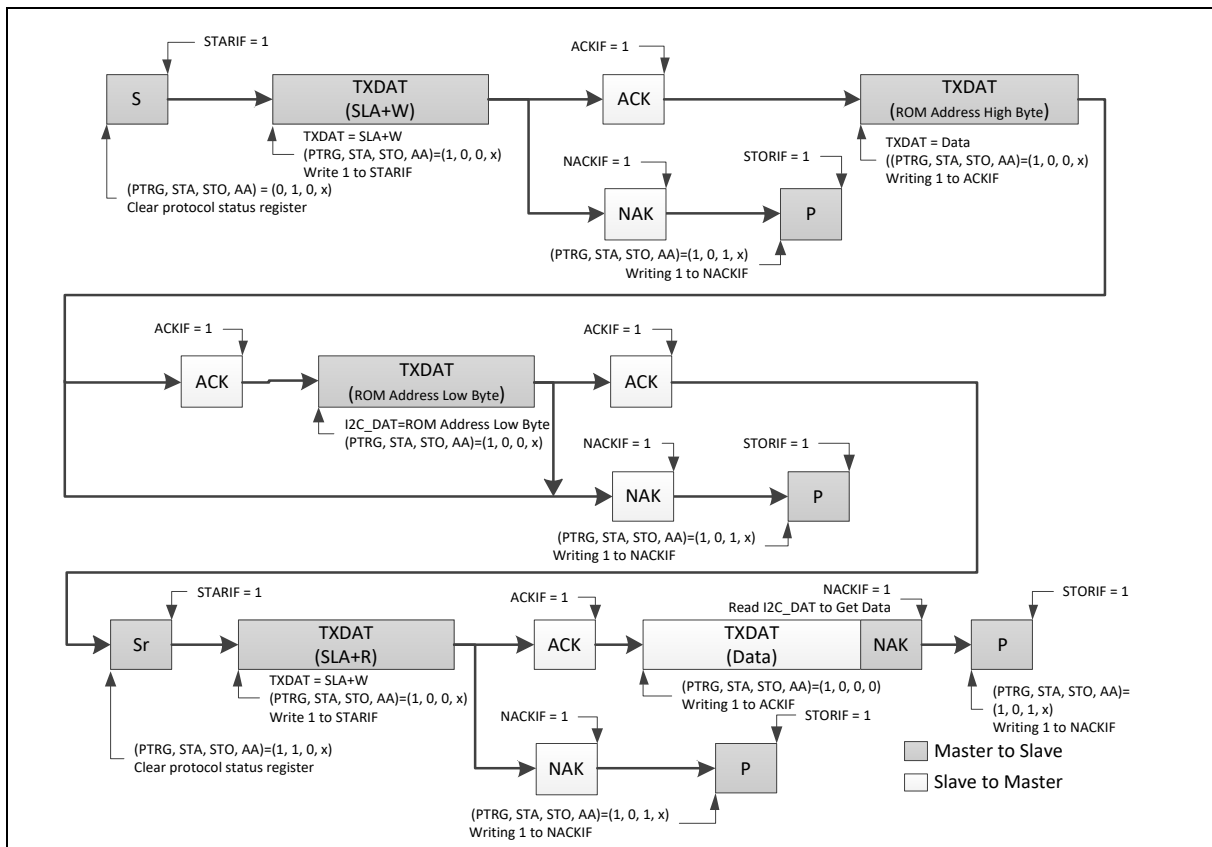


Figure 6.18-24 Protocol of EEPROM Random Read

The I²C controller, which is a master, sends START to bus. Then, it sends a SLA+W (Slave address + Write bit) to EEPROM followed by two bytes data address to set the EEPROM address to read. Finally, a Repeat START followed by SLA+R is sent to read the data from EEPROM.

6.18.6 Register Map

R: read only, W: write only, R/W: both read and write

Register	Offset	R/W	Description	Reset Value
USCI_I2C Base Address: $UI2Cn\_BA = 0x400D\_0000 + (0x1000 * n)$ $n = 0, 1, 2$				
UI2C_CTL	UI2Cn_BA+0x00	R/W	USCI Control Register	0x0000_0000
UI2C_BRGEN	UI2Cn_BA+0x08	R/W	USCI Baud Rate Generator Register	0x0000_3C00
UI2C_LINECTL	UI2Cn_BA+0x2C	R/W	USCI Line Control Register	0x0000_0000
UI2C_TXDAT	UI2Cn_BA+0x30	W	USCI Transmit Data Register	0x0000_0000
UI2C_RXDAT	UI2Cn_BA+0x34	R	USCI Receive Data Register	0x0000_0000
UI2C_DEVADDR0	UI2Cn_BA+0x44	R/W	USCI Device Address Register 0	0x0000_0000
UI2C_DEVADDR1	UI2Cn_BA+0x48	R/W	USCI Device Address Register 1	0x0000_0000
UI2C_ADDRMSK0	UI2Cn_BA+0x4C	R/W	USCI Device Address Mask Register 0	0x0000_0000
UI2C_ADDRMSK1	UI2Cn_BA+0x50	R/W	USCI Device Address Mask Register 1	0x0000_0000
UI2C_WKCTL	UI2Cn_BA+0x54	R/W	USCI Wake-up Control Register	0x0000_0000
UI2C_WKSTS	UI2Cn_BA+0x58	R/W	USCI Wake-up Status Register	0x0000_0000
UI2C_PROTCTL	UI2Cn_BA+0x5C	R/W	USCI Protocol Control Register	0x0000_0000
UI2C_PROTIEN	UI2Cn_BA+0x60	R/W	USCI Protocol Interrupt Enable Register	0x0000_0000
UI2C_PROTSTS	UI2Cn_BA+0x64	R/W	USCI Protocol Status Register	0x0000_0000
UI2C_ADMAT	UI2Cn_BA+0x88	R/W	I ² C Slave Match Address Register	0x0000_0000
UI2C_TMCTL	UI2Cn_BA+0x8C	R/W	I ² C Timing Configure Control Register	0x0002_0000

6.18.7 Register Description

USCI Control Register (UI2C_CTL)

Register	Offset	R/W	Description	Reset Value
UI2C_CTL	UI2Cn_BA+0x00	R/W	USCI Control Register	0x0000_0000

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

Bits	Description	
[31:3]	Reserved	Reserved.
[2:0]	FUNMODE	Function Mode This bit field selects the protocol for this USCI controller. Selecting a protocol that is not available or a reserved combination disables the USCI. When switching between two protocols, the USCI has to be disabled before selecting a new protocol. Simultaneously, the USCI will be reset when user write 000 to FUNMODE. 000 = The USCI is disabled. All protocol related state machines are set to idle state. 001 = The SPI protocol is selected. 010 = The UART protocol is selected. 100 = The I²C protocol is selected. Note: Other bit combinations are reserved.

USCI Baud Rate Generator Register (UI2C_BRGEN)

Register	Offset	R/W	Description	Reset Value
UI2C_BRGEN	UI2Cn_BA+0x08	R/W	USCI Baud Rate Generator Register	0x0000_3C00

31	30	29	28	27	26	25	24
Reserved						CLKDIV	
23	22	21	20	19	18	17	16
CLKDIV							
15	14	13	12	11	10	9	8
Reserved	DSCNT					PDSCNT	
7	6	5	4	3	2	1	0
Reserved		TMCNTSRC	TMCNTEN	SPCLKSEL		PTCLKSEL	RCLKSEL

Bits	Description	
[31:26]	Reserved	Reserved.
[25:16]	CLKDIV	Clock Divider This bit field defines the ratio between the protocol clock frequency $f_{\text{PROT\_CLK}}$ and the clock divider frequency $f_{\text{DIV\_CLK}}$ ($f_{\text{DIV\_CLK}} = f_{\text{PROT\_CLK}} / (\text{CLKDIV} + 1)$).
[15]	Reserved	Reserved.
[14:10]	DSCNT	Denominator for Sample Counter This bit field defines the divide ratio of the sample clock $f_{\text{SAMP\_CLK}}$. The divided frequency $f_{\text{DS\_CNT}} = f_{\text{PDS\_CNT}} / (\text{DSCNT} + 1)$. Note: The maximum value of DSCNT is 0xF on UART mode and suggest to set over 4 to confirm the receiver data is sampled in right value.
[9:8]	PDSCNT	Pre-divider for Sample Counter This bit field defines the divide ratio of the clock division from sample clock $f_{\text{SAMP\_CLK}}$. The divided frequency $f_{\text{PDS\_CNT}} = f_{\text{SAMP\_CLK}} / (\text{PDSCNT} + 1)$.
[7:6]	Reserved	Reserved.
[5]	TMCNTSRC	Time Measurement Counter Clock Source Selection 0 = Time measurement counter with $f_{\text{PROT\_CLK}}$. 1 = Time measurement counter with $f_{\text{DIV\_CLK}}$.
[4]	TMCNTEN	Time Measurement Counter Enable Bit This bit enables the 10-bit timing measurement counter. 0 = Time measurement counter Disabled. 1 = Time measurement counter Enabled.
[3:2]	SPCLKSEL	Sample Clock Source Selection This bit field used for the clock source selection of a sample clock ($f_{\text{SAMP\_CLK}}$) for the protocol processor. 00 = $f_{\text{SAMP\_CLK}} = f_{\text{DIV\_CLK}}$. 01 = $f_{\text{SAMP\_CLK}} = f_{\text{PROT\_CLK}}$. 10 = $f_{\text{SAMP\_CLK}} = f_{\text{SCLK}}$.

		11 = $f_{\text{SAMP\_CLK}} = f_{\text{REF\_CLK}}$.
[1]	PTCLKSEL	Protocol Clock Source Selection This bit selects the source signal of protocol clock ($f_{\text{PROT\_CLK}}$). 0 = Reference clock $f_{\text{REF\_CLK}}$. 1 = $f_{\text{REF\_CLK2}}$ (its frequency is half of $f_{\text{REF\_CLK}}$).
[0]	RCLKSEL	Reference Clock Source Selection This bit selects the source signal of reference clock ($f_{\text{REF\_CLK}}$). 0 = Peripheral device clock f_{PCLK} . 1 = Reserved.

USCI Line Control Register (UI2C_LINECTL)

Register	Offset	R/W	Description	Reset Value
UI2C_LINECTL	UI2Cn_BA+0x2C	R/W	USCI Line Control Register	0x0000_0000

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

Bits	Description	
[31:12]	Reserved	Reserved.
[11:8]	DWIDTH	Word Length of Transmission This bit field defines the data word length (amount of bits) for reception and transmission. The data word is always right-aligned in the data buffer. USCI support word length from 4 to 16 bits. 0x0: The data word contains 16 bits located at bit positions [15:0]. 0x1: Reserved. 0x2: Reserved. 0x3: Reserved. 0x4: The data word contains 4 bits located at bit positions [3:0]. 0x5: The data word contains 5 bits located at bit positions [4:0]. ... 0xF: The data word contains 15 bits located at bit positions [14:0].
[7:1]	Reserved	Reserved.
[0]	LSB	LSB First Transmission Selection 0 = The MSB, which bit of transmit/receive data buffer depends on the setting of DWIDTH, is transmitted/received first. 1 = The LSB, the bit 0 of data buffer, will be transmitted/received first.

USCI Transmit Data Register (UI2C_TXDAT)

Register	Offset	R/W	Description	Reset Value
UI2C_TXDAT	UI2Cn_BA+0x30	W	USCI Transmit Data Register	0x0000_0000

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

Bits	Description	
[31:16]	Reserved	Reserved.
[15:0]	TXDAT	Transmit Data Software can use this bit field to write 16-bit transmit data for transmission.

USCI Receive Data Register (UI2C_RXDAT)

Register	Offset	R/W	Description	Reset Value
UI2C_RXDAT	UI2Cn_BA+0x34	R	USCI Receive Data Register	0x0000_0000

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

Bits	Description	
[31:16]	Reserved	Reserved.
[15:0]	RXDAT	Received Data This bit field monitors the received data which stored in receive data buffer. Note: In I ² C protocol, RXDAT[12:8] indicate the different transmission conditions defined in I ² C.

USCI Device Address Register (UI2C DEVADDR)

Register	Offset	R/W	Description	Reset Value
UI2C_DEVADDR0	UI2Cn_BA+0x44	R/W	USCI Device Address Register 0	0x0000_0000
UI2C_DEVADDR1	UI2Cn_BA+0x48	R/W	USCI Device Address Register 1	0x0000_0000

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

Bits	Description	
[31:10]	Reserved	Reserved.
[9:0]	DEVADDR	Device Address In I²C protocol, this bit field contains the programmed slave address. If the first received address byte are 1111 0AAX_B, the AA bits are compared to the bits DEVADDR[9:8] to check for address match, where the X is R/W bit. Then the second address byte is also compared to DEVADDR[7:0]. Note 1: The DEVADDR [9:7] must be set 3'b000 when I²C operating in 7-bit address mode. Note 2: When software sets 10'h000, the address cannot be used.

USCI Device Address Mask Register (UI2C_ADDRMSK) – for I²C Only

Register	Offset	R/W	Description	Reset Value
UI2C_ADDRMSK0	UI2Cn_BA+0x4C	R/W	USCI Device Address Mask Register 0	0x0000_0000
UI2C_ADDRMSK1	UI2Cn_BA+0x50	R/W	USCI Device Address Mask Register 1	0x0000_0000

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

Bits	Description
[31:10]	Reserved Reserved.
[9:0]	ADDRMSK USCI Device Address Mask 0 = Mask Disabled (the received corresponding register bit should be exact the same as address register.). 1 = Mask Enabled (the received corresponding address bit is don't care.). USCI supports multiple address recognition with two address mask register. When the bit in the address mask register is set to 1, it means the received corresponding address bit is don't-care. If the bit is set to 0, that means the received corresponding register bit should be exact the same as address register. Note: The wake-up function cannot use address mask.

USCI Wake-up Control Register (UI2C_WKCTL)

Register	Offset	R/W	Description	Reset Value
UI2C_WKCTL	UI2Cn_BA+0x54	R/W	USCI Wake-up Control Register	0x0000_0000

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

Bits	Description	
[31:2]	Reserved	Reserved.
[1]	WKADDREN	Wake-up Address Match Enable Bit 0 = The chip is woken up according to data toggle. 1 = The chip is woken up according to address match.
[0]	WKEN	Wake-up Enable Bit 0 = Wake-up function Disabled. 1 = Wake-up function Enabled.

USCI Wake-up Status Register (UI2C WKSTS)

Register	Offset	R/W	Description	Reset Value
UI2C_WKSTS	UI2Cn_BA+0x58	R/W	USCI Wake-up Status Register	0x0000_0000

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

Bits	Description	
[31:1]	Reserved	Reserved.
[0]	WKF	Wake-up Flag When chip is woken up from Power-down mode, this bit is set to 1. Software can write 1 to clear this bit.

USCI Protocol Control Register – I²C (UI2C_PROTCTL)

Register	Offset	R/W	Description	Reset Value
UI2C_PROTCTL	UI2Cn_BA+0x5C	R/W	USCI Protocol Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PROTEN	Reserved					TOCNT	
23	22	21	20	19	18	17	16
TOCNT							
15	14	13	12	11	10	9	8
Reserved						MONEN	SCLOUTEN
7	6	5	4	3	2	1	0
Reserved		PTRG	ADDR10EN	STA	STO	AA	GCFUNC

Bits	Description	
[31]	PROTEN	I²C Protocol Enable Bit 0 = I ² C Protocol Disabled. 1 = I ² C Protocol Enabled.
[30:26]	Reserved	Reserved.
[25:16]	TOCNT	Time-out Clock Cycle This bit field indicates how many clock cycle selected by TMCNTSRC (UI2C_BRGEN [5]) when each interrupt flags are clear. The time-out is enable when TOCNT bigger than 0. Note: The TMCNTSRC (UI2C_BRGEN [5]) must be set as 0 on I ² C mode.
[15:10]	Reserved	Reserved.
[9]	MONEN	Monitor Mode Enable Bit This bit enables monitor mode. In monitor mode the SDA output will be put in high impedance mode. This prevents the I ² C module from outputting data of any kind (including ACK) onto the I ² C data bus. 0 = The monitor mode Disabled. 1 = The monitor mode Enabled. Note: Depending on the state of the SCLOUTEN bit, the SCL output may be also forced high, preventing the module from having control over the I ² C clock line.
[8]	SCLOUTEN	SCL Output Enable Bit This bit enables monitor pulling SCL to low. This monitor will pull SCL to low until it has had time to respond to an I ² C interrupt. 0 = SCL output will be forced high due to open drain mechanism. 1 = I ² C module may act as a slave peripheral just like in normal operation, the I ² C holds the clock line low until it has had time to clear I ² C interrupt.
[7:6]	Reserved	Reserved.

[5]	PTRG	I²C Protocol Trigger (Write Only) When a new state is present in the UI2C_PROTSTS register, if the related interrupt enable bits are set, the I ² C interrupt is requested. It must write one by software to this bit after the related interrupt flags are set to 1 and the I ² C protocol function will go ahead until the STOP is active or the PROTEN is disabled. 0 = I2C's stretch disabled and the I ² C protocol function will go ahead. 1 = I2C's stretch active.
[4]	ADDR10EN	Address 10-bit Function Enable Bit 0 = Address match 10 bit function Disabled. 1 = Address match 10 bit function Enabled.
[3]	STA	I²C START Control Setting STA to logic 1 to enter Master mode, the I ² C hardware sends a START or repeat START condition to bus when the bus is free.
[2]	STO	I²C STOP Control In Master mode, setting STO to transmit a STOP condition to bus then I ² C hardware will check the bus condition if a STOP condition is detected this bit will be cleared by hardware automatically. In a slave mode, setting STO resets I ² C hardware to the defined "not addressed" slave mode when bus error (UI2C_PROTSTS.ERRIF = 1).
[1]	AA	Assert Acknowledge Control When AA =1 prior to address or data received, an acknowledged (low level to SDA) will be returned during the acknowledge clock pulse on the SCL line when (1) A slave is acknowledging the address sent from master, (2) The receiver devices are acknowledging the data sent by transmitter. When AA=0 prior to address or data received, a Not acknowledged (high level to SDA) will be returned during the acknowledge clock pulse on the SCL line.
[0]	GCFUNC	General Call Function 0 = General Call Function Disabled. 1 = General Call Function Enabled.

USCI Protocol Interrupt Enable Register – I²C (UI2C_PROTIEN)

Register	Offset	R/W	Description	Reset Value
UI2C_PROTIEN	UI2Cn_BA+0x60	R/W	USCI Protocol Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved	ACKIEN	ERRIEN	ARBLOIEN	NACKIEN	STORIEN	STARIEN	TOIEN

Bits	Description	
[31:7]	Reserved	Reserved.
[6]	ACKIEN	Acknowledge Interrupt Enable Bit This bit enables the generation of a protocol interrupt if an acknowledge is detected by a master. 0 = The acknowledge interrupt Disabled. 1 = The acknowledge interrupt Enabled.
[5]	ERRIEN	Error Interrupt Enable Bit This bit enables the generation of a protocol interrupt if an I ² C error condition is detected (indicated by ERR (UI2C_PROTSTS [16])). 0 = The error interrupt Disabled. 1 = The error interrupt Enabled.
[4]	ARBLOIEN	Arbitration Lost Interrupt Enable Bit This bit enables the generation of a protocol interrupt if an arbitration lost event is detected. 0 = The arbitration lost interrupt Disabled. 1 = The arbitration lost interrupt Enabled.
[3]	NACKIEN	Non - Acknowledge Interrupt Enable Bit This bit enables the generation of a protocol interrupt if a Non - acknowledge is detected by a master. 0 = The non - acknowledge interrupt Disabled. 1 = The non - acknowledge interrupt Enabled.
[2]	STORIEN	STOP Condition Received Interrupt Enable Bit This bit enables the generation of a protocol interrupt if a STOP condition is detected. 0 = The stop condition interrupt Disabled. 1 = The stop condition interrupt Enabled.
[1]	STARIEN	START Condition Received Interrupt Enable Bit This bit enables the generation of a protocol interrupt if a START condition is detected. 0 = The start condition interrupt Disabled.

		1 = The start condition interrupt Enabled.
[0]	TOIEN	Time-out Interrupt Enable Bit In I ² C protocol, this bit enables the interrupt generation in case of a time-out event. 0 = The time-out interrupt Disabled. 1 = The time-out interrupt Enabled.

USCI Protocol Status Register – I²C (UI2C PROTSTS)

Register	Offset	R/W	Description	Reset Value
UI2C_PROTSTS	UI2Cn_BA+0x64	R/W	USCI Protocol Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved				ERRARBLO	BUSHANG	WRSTSWK	WKAKDONE
15	14	13	12	11	10	9	8
SLAREAD	SLASEL	ACKIF	ERRIF	ARBLOIF	NACKIF	STORIF	STARIF
7	6	5	4	3	2	1	0
Reserved	ONBUSY	TOIF	Reserved				

Bits	Description
[31:20]	Reserved Reserved.
[19]	ERRARBLO Error Arbitration Lost This bit indicates bus arbitration lost due to bigger noise which is can't be filtered by input processor. The I ² C can send start condition when ERRARBLO is set. Thus this bit doesn't be cared on slave mode. 0 = The bus is normal status for transmission. 1 = The bus is error arbitration lost status for transmission. Note: This bit has no interrupt signal, and it will be cleared automatically by hardware when a START condition is present.
[18]	BUSHANG Bus Hang-up This bit indicates bus hang-up status. There is 4-bit counter count when SCL hold high and refer f _{SAMP_CLK} . The hang-up counter will count to overflow and set this bit when SDA is low. The counter will be reset by falling edge of SCL signal. 0 = The bus is normal status for transmission. 1 = The bus is hang-up status for transmission. Note: This bit has no interrupt signal, and it will be cleared automatically by hardware when a START condition is present.
[17]	WRSTSWK Read/Write Status Bit in Address Wake-up Frame 0 = Write command is recorded on the address match wake-up frame. 1 = Read command is recorded on the address match wake-up frame.
[16]	WKAKDONE Wake-up Address Frame Acknowledge Bit Done 0 = The ACK bit cycle of address match frame isn't done. 1 = The ACK bit cycle of address match frame is done in power-down. Note: This bit can't release when WKUPIF is set.
[15]	SLAREAD Slave Read Request Status This bit indicates that a slave read request has been detected. 0 = A slave R/W bit is 1 has not been detected. 1 = A slave R/W bit is 1 has been detected. Note: This bit has no interrupt signal, and it will be cleared automatically by hardware.

[14]	SLASEL	Slave Select Status This bit indicates that this device has been selected as slave. 0 = The device is not selected as slave. 1 = The device is selected as slave. Note: This bit has no interrupt signal, and it will be cleared automatically by hardware.
[13]	ACKIF	Acknowledge Received Interrupt Flag This bit indicates that an acknowledge has been received in master mode. A protocol interrupt can be generated if UI2C_PROTIEN.ACKIEN = 1. 0 = An acknowledge has not been received. 1 = An acknowledge has been received. Note: It is cleared by software writing 1 into this bit.
[12]	ERRIF	Error Interrupt Flag This bit indicates that a Bus Error occurs when a START or STOP condition is present at an illegal position in the formation frame. Example of illegal position are during the serial transfer of an address byte, a data byte or an acknowledge bit. A protocol interrupt can be generated if UI2C_PROTIEN.ERRIEN = 1. 0 = An I ² C error has not been detected. 1 = An I ² C error has been detected. Note 1: It is cleared by software writing 1 into this bit Note 2: This bit is set for slave mode, and user must write 1 into STO register to the defined "not addressed" slave mode.
[11]	ARBLOIF	Arbitration Lost Interrupt Flag This bit indicates that an arbitration has been lost. A protocol interrupt can be generated if UI2C_PROTIEN.ARBLOIEN = 1. 0 = An arbitration has not been lost. 1 = An arbitration has been lost. Note: It is cleared by software writing 1 into this bit.
[10]	NACKIF	Non - Acknowledge Received Interrupt Flag This bit indicates that a non - acknowledge has been received in master mode. A protocol interrupt can be generated if UI2C_PROTIEN.NACKIEN = 1. 0 = A non - acknowledge has not been received. 1 = A non - acknowledge has been received. Note: It is cleared by software writing 1 into this bit.
[9]	STORIF	Stop Condition Received Interrupt Flag This bit indicates that a stop condition has been detected on the I ² C bus lines. A protocol interrupt can be generated if UI2C_PROTIEN.STORIEN = 1. 0 = A stop condition has not yet been detected. 1 = A stop condition has been detected. Note 1: It is cleared by software writing 1 into this bit.
[8]	STARIF	Start Condition Received Interrupt Flag This bit indicates that a start condition or repeated start condition has been detected on master mode. However, this bit also indicates that a repeated start condition has been detected on slave mode. A protocol interrupt can be generated if UI2C_PROTIEN.STARIEN = 1. 0 = A start condition has not yet been detected. 1 = A start condition has been detected. Note: It is cleared by software writing 1 into this bit.
[7]	Reserved	Reserved.
[6]	ONBUSY	On Bus Busy

		Indicates that a communication is in progress on the bus. It is set by hardware when a START condition is detected. It is cleared by hardware when a STOP condition is detected 0 = The bus is IDLE (both SCLK and SDA High). 1 = The bus is busy.
[5]	TOIF	Time-out Interrupt Flag 0 = A time-out interrupt status has not occurred. 1 = A time-out interrupt status has occurred. Note: It is cleared by software writing 1 into this bit
[4:0]	Reserved	Reserved.

USCI Slave Match Address Register (UI2C_ADMAT)

Register	Offset	R/W	Description	Reset Value
UI2C_ADMAT	UI2Cn_BA+0x88	R/W	I ² C Slave Match Address Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved						ADMAT1	ADMAT0

Bits	Description	
[31:2]	Reserved	Reserved.
[1]	ADMAT1	USCI Address 1 Match Status Register When address 1 is matched, hardware will inform which address used. This bit will set to 1, and software can write 1 to clear this bit.
[0]	ADMAT0	USCI Address 0 Match Status Register When address 0 is matched, hardware will inform which address used. This bit will set to 1, and software can write 1 to clear this bit.

USCI Timing Configure Control Register (UI2C_TMCTL)

Register	Offset	R/W	Description	Reset Value
UI2C_TMCTL	UI2Cn_BA+0x8C	R/W	I ² C Timing Configure Control Register	0x0002_0000

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

Bits	Description	
[31:25]	Reserved	Reserved.
[24:16]	HTCTL	Hold Time Configure Control This field is used to generate the delay timing between SCL falling edge SDA edge in transmission mode. The delay hold time is numbers of peripheral clock = HTCTL x f _{PLK} .
[15:9]	Reserved	Reserved.
[8:0]	STCTL	Setup Time Configure Control This field is used to generate a delay timing between SDA edge and SCL rising edge in transmission mode. The delay setup time is numbers of peripheral clock = STCTL x f _{PLK} .

6.19 USB 2.0 Full-Speed Device Controller (USBD)

6.19.1 Overview

There is one set of USB 2.0 full-speed device controller and transceiver in this device. It is compliant with USB 2.0 full-speed device specification and supports control/bulk/interrupt/isochronous transfer types.

In this device controller, there are two main interfaces: APB bus and USB bus which comes from the USB PHY transceiver. For the APB bus, the CPU can program control registers through it. There are 1 Kbytes internal SRAM as data buffer in this controller. For IN or OUT transfer, it is necessary to write data to SRAM or read data from SRAM through the APB interface or SIE. User needs to set the effective starting address of SRAM for each endpoint buffer through buffer segmentation register (USBD_BUFSEG0~16).

There are 17 endpoints in this controller. Each of the endpoint can be configured as IN or OUT endpoint. All the operations including Control, Bulk, Interrupt and Isochronous transfer are implemented in this block. The block of "Endpoint Control" is also used to manage the data sequential synchronization, endpoint states, current start address, transaction status, and data buffer status for each endpoint.

There are five different interrupt events in this controller. They are no-event-wake-up, device plug-in or plug-out event, USB events, such as IN ACK, OUT ACK, etc., and BUS events, such as suspend and resume, etc. Any event will cause an interrupt, and users just need to check the related event flags in interrupt event status register (USBD_INTSTS) to acknowledge what kind of interrupt occurred, and then check the related USB Endpoint Status Register (USBD_EPSTS0 and USBD_EPSTS1) to acknowledge what kind of event occurred in this endpoint.

A software-disconnect function is supported for this USB controller. It is used to simulate the disconnection of this device from the host. If user enables SE0 bit (USBD_SE0), the USB controller will force the output of USB_D+ and USB_D- to level low and its function is disabled. After disabling the SE0 bit, the host will enumerate the USB device again.

For more information on the Universal Serial Bus, please refer to *Universal Serial Bus Specification Revision 1.1*.

6.19.2 Features

- Compliant with USB 2.0 Full-Speed specification
- Provides 1 interrupt vector with 5 different interrupt events (SOF, NEVWK, VBUSDET, USB and BUS)
- Supports Control/Bulk/Interrupt/Isochronous transfer type
- Supports suspend function when no bus activity existing for 3ms
- Supports 17 endpoints for configurable Control/Bulk/Interrupt/Isochronous transfer types and maximum 1 Kbytes buffer size
- Provides remote wake-up capability

6.19.3 Block Diagram

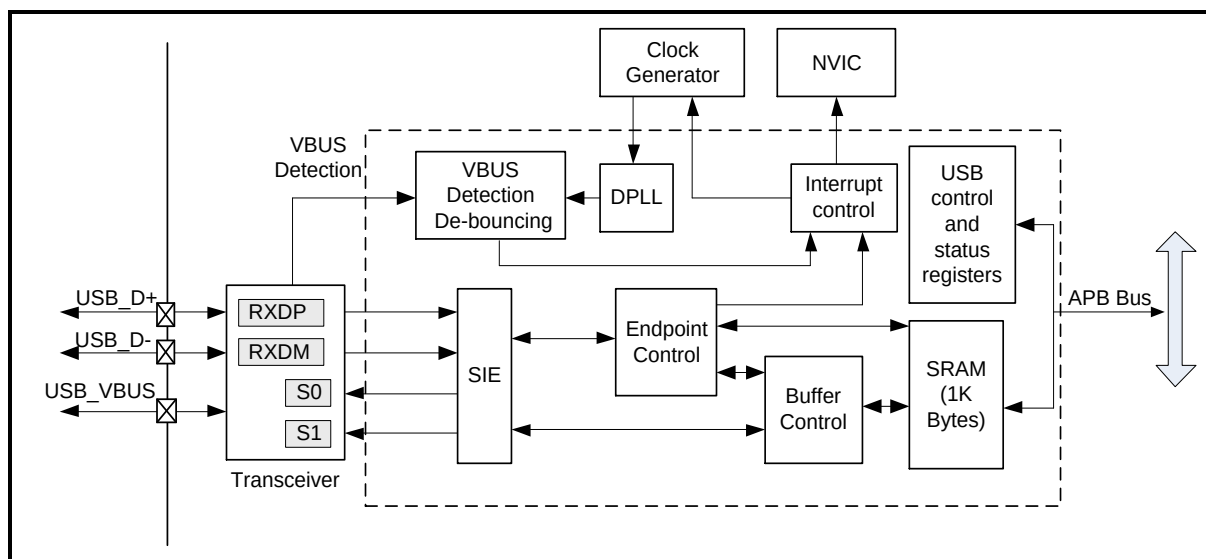


Figure 6.19.1 USB Block Diagram

6.19.4 Basic Configuration

The USB clock source is derived from HIRC or PLL. If user would like to achieve crystal-less, HIRC must be selected as USB clock source, or user has to use the PLL related configurations with crystal before USB device controller is enabled. Set the USBCKEN (CLK_APBCLK0[27]) bit to enable USB clock and 4-bit pre-scaler USBDIV (CLK_CLKDIV0[7:4]) to generate the proper USB clock rate.

Note: USB peripheral clock frequency needs to be greater than 12 MHz when USB receives/transmits data from/to the host.

6.19.4.1 USB 2.0 Full-Speed Basic Configuration

- Clock source Configuration
 - Enable HIRC (CLK_PWRCTL[2]) or set PLL controller (CLK_PLLCTL)
 - Set USBSEL (CLK_CLKSEL0[8]) to select USB clock source from HIRC or PLL
 - Select the clock divider number of USB peripheral clock on USBDIV (CLK_CLKDIV0[7:4])
 - Enable USB peripheral clock in USBCKEN (CLK_APBCLK0[27])
- Reset Configuration
 - Reset USB controller in USBDRST (SYS_IPRST1[27])

6.19.5 Functional Description

6.19.5.1 Serial Interface Engine (SIE)

The SIE is the front-end of the device controller and handles most of the USB packet protocol. The SIE typically comprehends signaling up to the transaction level. The functions that it handles could include:

- Packet recognition and transaction sequencing
- SOP, EOP, RESET, RESUME signal detection/generation
- Clock/Data separation
- NRZI Data encoding/decoding and bit-stuffing

- CRC generation and checking (for Token and Data)
- Packet ID (PID) generation and checking/decoding
- Serial-Parallel/Parallel-Serial conversion

6.19.5.2 Endpoint Control

This controller supports 17 endpoints. Each of the endpoint can be configured as Control, Bulk, Interrupt, or Isochronous transfer type. All the operations including Control, Bulk, Interrupt and Isochronous transfer are implemented in this block. It is also used to manage the data sequential synchronization, endpoint state control, current endpoint start address, current transaction status, and data buffer status in each endpoint.

6.19.5.3 Digital Phase Lock Loop (DPLL)

The bit rate of USB data is 12 MHz. The DPLL uses the 48 MHz which comes from the clock controller to lock the input data RXDP and RXDM. The 12 MHz bit rate clock is also converted from DPLL.

6.19.5.4 V_{BUS} Detection De-bouncing

A USB device may be plugged-in or plugged-out from the USB host. To monitor the state of a USB device when it is detached from the USB host, the device controller provides hardware de-bouncing for USB V_{BUS} detection interrupt to avoid bounce problems on USB plug-in or unplug. V_{BUS} detection interrupt appears about 10ms later than USB plug-in or plug-out. User can acknowledge USB plug-in/plug-out by reading USBD_VBUSDET register. The VBUSDET flag represents the current state on the bus without de-bouncing. If VBUSDET is 1, it means the USB cable is plugged-in. If user polls the flag to check USB state, software de-bouncing must be added if needed.

6.19.5.5 Interrupt control

This USB provides 1 interrupt vector with 5 interrupt events (SOF, NEVWK, VBUSDET, USB and BUS). The SOF interrupt occurs after receiving SOF packet. The NEVWK event occurs after waking up the system from Power-down mode (The power mode function is defined in system power-down control register, CLK_PWRCTL). The VBUSDET event is used for USB plug-in or unplug. The USB event notifies users of some USB requests, such as IN ACK, OUT ACK. And the BUS event notifies users of some bus events, such as suspend and resume. The related bits must be set in the interrupt enable register (USBD_INTEN) of USB Device Controller to enable USB interrupts.

NEVWK interrupt is only presented when no other USB interrupt events happened more than 20ms after the chip is woken up from Power-down mode. After the chip enters Power-down mode, any power change on USB_VBUS or receiving resume event in USB suspend state can wake up this chip if USB wake-up function is enabled. If this change is not intentionally, no interrupt but NEVWK interrupt will occur. After waking up by USB, this interrupt will occur when no other USB interrupt events are presented for more than 20ms. Figure 6.19.2 shows the control flow of wake-up interrupt.

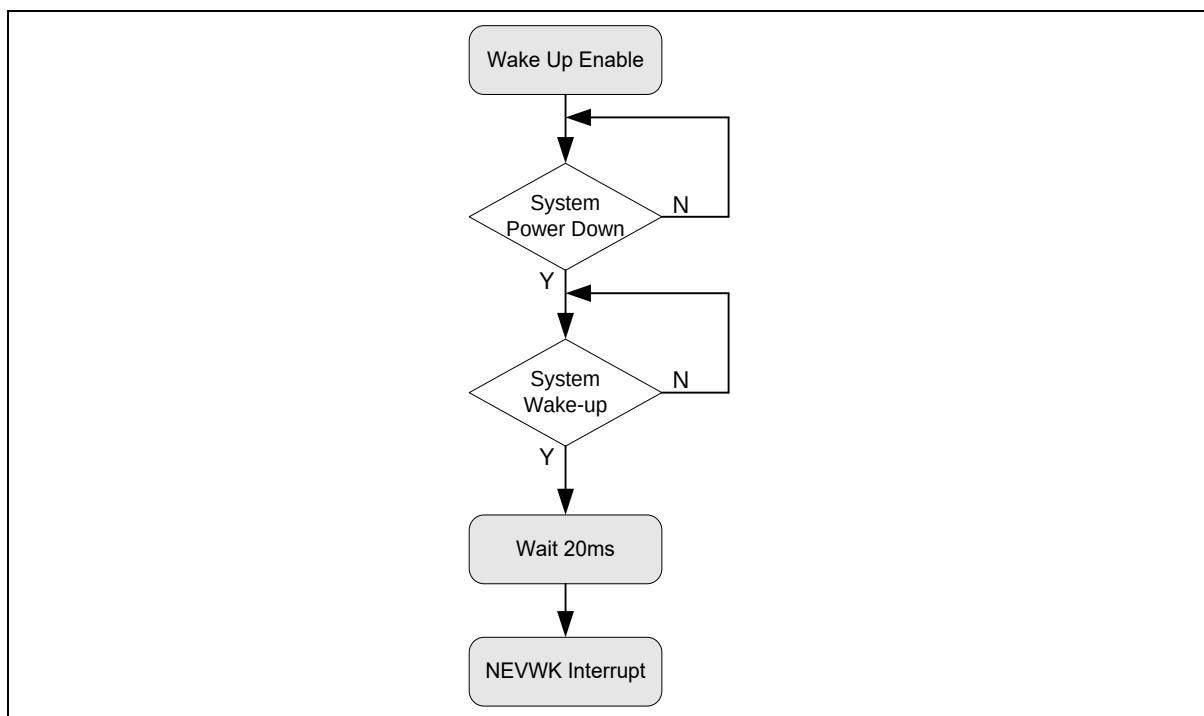


Figure 6.19.2 NEVWK Interrupt Operation Flow

The USB interrupt is used to notify users of any USB event on the bus, and user can read EPSTS (USBD_EPSTS0 and USBD_EPSTS1) and EPEVT11~0 (USBD_INTSTS[27:16]) to take necessary responses.

Same as USB interrupt, BUS interrupt notifies users of some bus events, like USB reset, suspend, time-out, and resume. A user can read USBD_ATTR to acknowledge bus events.

6.19.5.6 Power Saving

User can write 0 to USBD_ATTR[4] to disable PHY under special circumstances, like suspend, to conserve power.

6.19.5.7 Buffer Control

There is 1 Kbytes SRAM in the controller and the 17 endpoints share this buffer. User shall configure each endpoint's effective starting address in the buffer segmentation register before the USB function active. The "Buffer Control" block is used to control each endpoint's effective starting address and its SRAM size is defined in the USBD_MXPLD0~16 register.

Figure 6.19.3 depicts the starting address for each endpoint according to the content of USBD_BUFSEG0~16 and USBD_MXPLD0~16 registers. If the USBD_BUFSEG0 is programmed as 0x08h and USBD_MXPLD0 is set as 0x40h, the SRAM size of endpoint 0 starts from USBD_BA+0x108h and ends in USBD_BA+0x148h.

Note: The USBD SRAM base is USBD_BA+0x100h.

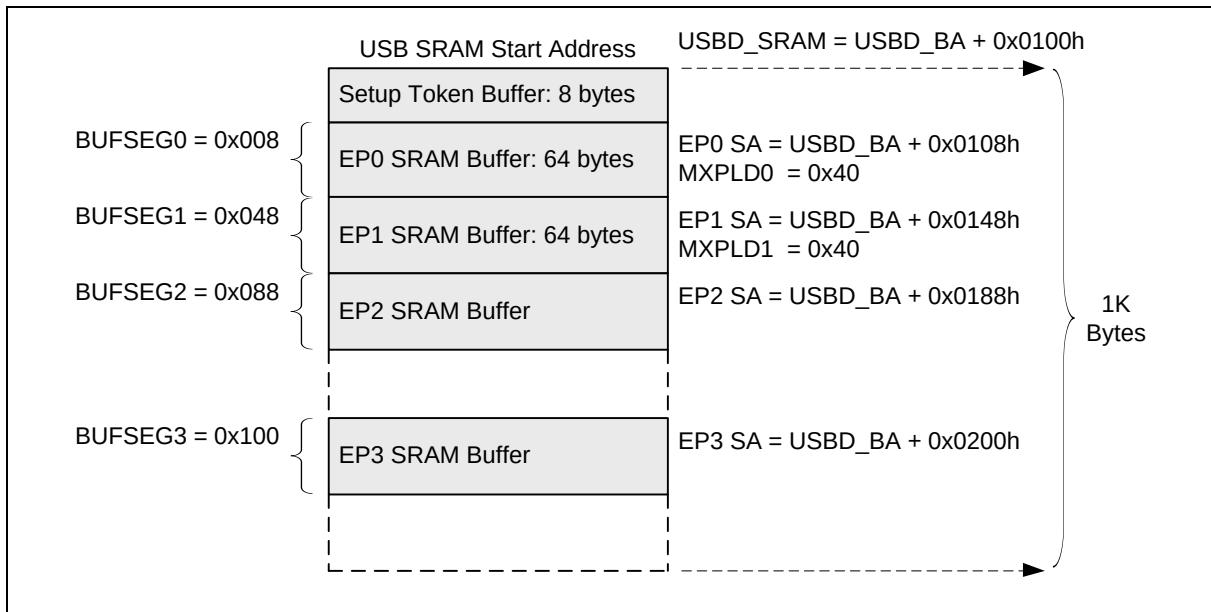


Figure 6.19.3 Endpoint SRAM Structure

6.19.5.8 Handling Transactions with USB Device Peripheral

User can use interrupt or polling `USBD_INTSTS` to monitor the USB transactions. When transactions occur, `USBD_INTSTS` will be set by hardware and send an interrupt request to CPU (if related interrupt enabled), or user can polling `USBD_INTSTS` to get these events without interrupt. The following is the control flow with interrupt enabled.

When USB host has requested data from a device controller, user needs to prepare related data in the specified endpoint buffer in advance. After buffering the required data, user needs to write the actual data length in the specified `USBD_MXPLD0~16` register. Once this register is written, the internal signal “`In_Rdy`” will be asserted and the buffering data will be transmitted immediately after receiving associated IN token from host. Note that after transferring the specified data, the signal “`In_Rdy`” will de-assert automatically by hardware.

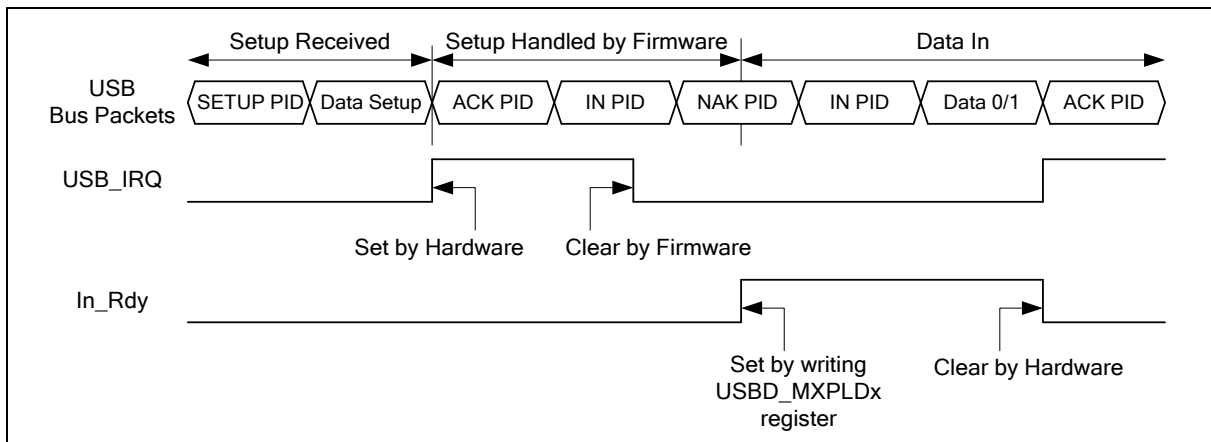


Figure 6.19.4 Setup Transaction Followed by Data IN Transaction

Alternatively, when USB host wants to transmit data to the OUT endpoint in the device controller, hardware will buffer these data to the specified endpoint buffer. After this transaction is completed, hardware will record the data length in specified `USBD_MXPLD0~16` register and de-assert the internal signal “`Out_Rdy`”. This will avoid hardware accepting next transaction until user moves out the current data in the related endpoint buffer. Once users have processed this transaction, the specified `USBD_`

MXPLD0~16 register needs to be written by firmware to assert the signal “Out_Rdy” again to accept the next transaction.

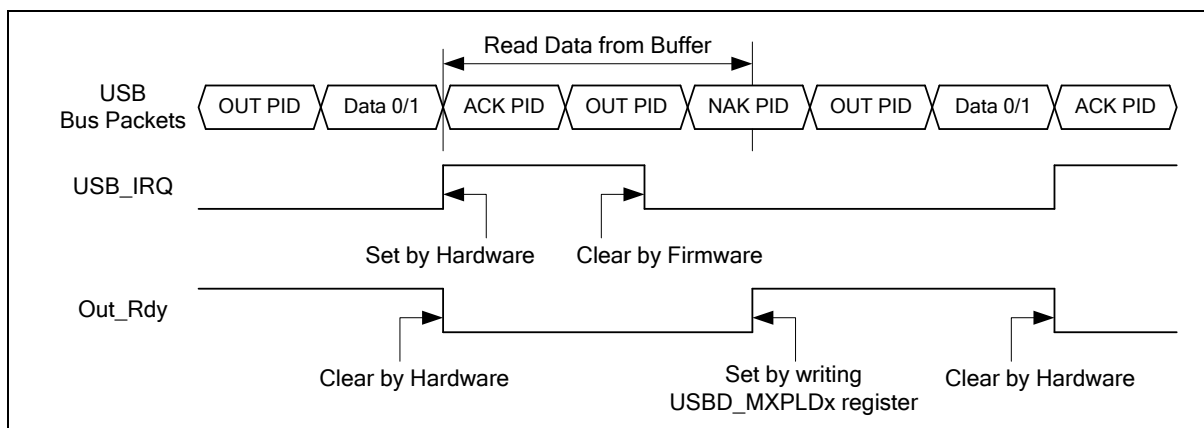


Figure 6.19.5 Data Out Transfer

6.19.5.9 Link Power Management (LPM)

Power Management (LPM) which is similar to the suspend/resume function, but has transitional latencies of tens of microseconds between power states (instead of three to greater than 20 millisecond latencies of the USB2.0 suspend/resume)

Through the LPM mechanism host lets device state fast from an enable state (called L0), to a new sleep state (called L1). For detailed definition of L0 and L1 state, see Table 6.19-1. The register USB_D_ATTR & USB_D_LPMATTR can let user know current power state for LPM mechanism.

LPM State	Description
L0 (On)	In this state, the port is enabled for propagation of transaction signaling traffic. A port in L0 is either actively transmitting or receiving data (L0-Active) or able to do so but not currently transmitting or receiving information (L0-Idle). While in this state Start-of-Frame (SOF) packets are issued by the host at a rate corresponding to the speed of the client device.
L1 (Sleep)	L1 is similar to L2 (below) but supports finer granularity in use. When in L1, the line state is identical to L2. Entry to L1 is started by a request to a hub or host port to transition to L1. A LPM transaction is sent to the downstream device. The requested transition can only occur if the device response with an ACK handshake. Exit from L1 is via remote wake, resume signaling, reset signaling or disconnect. L1 does not impose any specific power draw requirements (from V_{BUS}) on the attached device as L2 does. Either the host or device can initiate resume signaling when in L1. Although the signaling levels of resume are the same as L2, the duration of the signaling and transitional latencies associated with the L1 to L0 transition are much shorter.
L2 (Suspend)	This is the formalized name for USB 2.0 Suspend, Entry to L2 is nominally triggered by a command to a hub or host port to transition to suspend. The device discovers the suspend condition via observing 3ms of inactivity. The resultant line state is either Low or Full-speed idle. L2 also imposes power draw requirements (from V_{BUS}) on the attached device. Exit from this state is via remote wake, resume signaling, reset signaling or disconnect.
L3 (Off)	In this state, the port is not capable of performing any data signaling. It corresponds to the powered-off, disconnected, and disabled states.

Table 6.19-1 USB Link Power Manager (Lx) States

For the state transaction process, please refer to Figure 6.19.6, and for more information on the USB Link Power Manager (LPM), please refer to USB2.0 Link Power Management ECN(Engineering Change Notice).

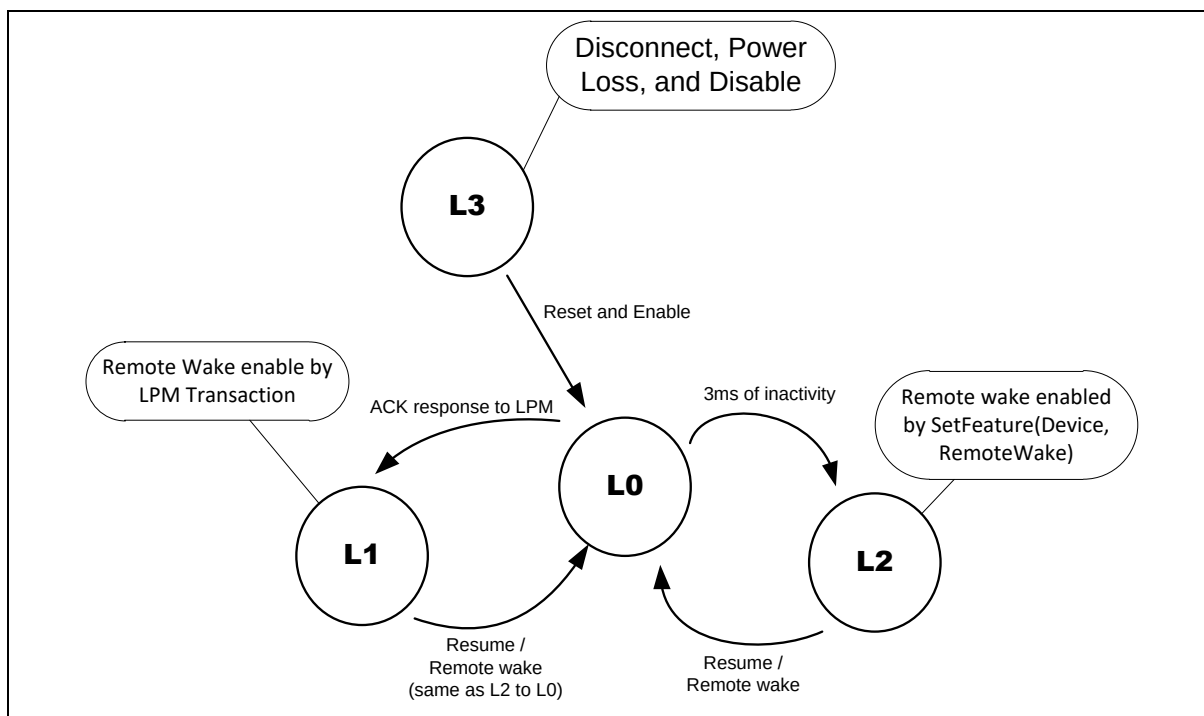


Figure 6.19.6 LPM State Transition Diagram

6.19.6 Register Map

R: read only, W: write only, R/W: both read and write

Register	Offset	R/W	Description	Reset Value
USB_D Base Address: USB_D_BA = 0x400C_0000				
USB_D_INTEN	USB_D_BA+0x000	R/W	USB Device Interrupt Enable Register	0x0000_0000
USB_D_INTSTS	USB_D_BA+0x004	R/W	USB Device Interrupt Event Status Register	0x0000_0000
USB_D_FADDR	USB_D_BA+0x008	R/W	USB Device Function Address Register	0x0000_0000
USB_D_EPSTS	USB_D_BA+0x00C	R	USB Device Endpoint Status Register	0x0000_0000
USB_D_ATTR	USB_D_BA+0x010	R/W	USB Device Bus Status and Attribution Register	0x0000_0040
USB_D_VBUSDET	USB_D_BA+0x014	R	USB Device V _{BUS} Detection Register	0x0000_0000
USB_D_STBUFSEG	USB_D_BA+0x018	R/W	SETUP Token Buffer Segmentation Register	0x0000_0000
USB_D_EPSTS0	USB_D_BA+0x020	R	USB Device Endpoint Status Register 0	0x0000_0000
USB_D_EPSTS1	USB_D_BA+0x024	R	USB Device Endpoint Status Register 1	0x0000_0000
USB_D_EPSTS2	USB_D_BA+0x028	R	USB Device Endpoint Status Register 2	0x0000_0000
USB_D_LPMATTR	USB_D_BA+0x088	R	USB LPM Attribution Register	0x0000_0000
USB_D_FN	USB_D_BA+0x08C	R	USB Frame Number Register	0x0000_0XXX
USB_D_SE0	USB_D_BA+0x090	R/W	USB Device Drive SE0 Control Register	0x0000_0001
USB_D_BUFSEG0	USB_D_BA+0x500	R/W	Endpoint 0 Buffer Segmentation Register	0x0000_0000
USB_D_MXPLD0	USB_D_BA+0x504	R/W	Endpoint 0 Maximal Payload Register	0x0000_0000
USB_D_CFG0	USB_D_BA+0x508	R/W	Endpoint 0 Configuration Register	0x0000_0000
USB_D_CFGP0	USB_D_BA+0x50C	R/W	Endpoint 0 Set Stall and Clear In/Out Ready Control Register	0x0000_0000
USB_D_BUFSEG1	USB_D_BA+0x510	R/W	Endpoint 1 Buffer Segmentation Register	0x0000_0000
USB_D_MXPLD1	USB_D_BA+0x514	R/W	Endpoint 1 Maximal Payload Register	0x0000_0000
USB_D_CFG1	USB_D_BA+0x518	R/W	Endpoint 1 Configuration Register	0x0000_0000
USB_D_CFGP1	USB_D_BA+0x51C	R/W	Endpoint 1 Set Stall and Clear In/Out Ready Control Register	0x0000_0000
USB_D_BUFSEG2	USB_D_BA+0x520	R/W	Endpoint 2 Buffer Segmentation Register	0x0000_0000
USB_D_MXPLD2	USB_D_BA+0x524	R/W	Endpoint 2 Maximal Payload Register	0x0000_0000
USB_D_CFG2	USB_D_BA+0x528	R/W	Endpoint 2 Configuration Register	0x0000_0000
USB_D_CFGP2	USB_D_BA+0x52C	R/W	Endpoint 2 Set Stall and Clear In/Out Ready Control Register	0x0000_0000
USB_D_BUFSEG3	USB_D_BA+0x530	R/W	Endpoint 3 Buffer Segmentation Register	0x0000_0000
USB_D_MXPLD3	USB_D_BA+0x534	R/W	Endpoint 3 Maximal Payload Register	0x0000_0000

USBD_CFG3	USBD_BA+0x538	R/W	Endpoint 3 Configuration Register	0x0000_0000
USBD_CFGP3	USBD_BA+0x53C	R/W	Endpoint 3 Set Stall and Clear In/Out Ready Control Register	0x0000_0000
USBD_BUFSEG4	USBD_BA+0x540	R/W	Endpoint 4 Buffer Segmentation Register	0x0000_0000
USBD_MXPLD4	USBD_BA+0x544	R/W	Endpoint 4 Maximal Payload Register	0x0000_0000
USBD_CFG4	USBD_BA+0x548	R/W	Endpoint 4 Configuration Register	0x0000_0000
USBD_CFGP4	USBD_BA+0x54C	R/W	Endpoint 4 Set Stall and Clear In/Out Ready Control Register	0x0000_0000
USBD_BUFSEG5	USBD_BA+0x550	R/W	Endpoint 5 Buffer Segmentation Register	0x0000_0000
USBD_MXPLD5	USBD_BA+0x554	R/W	Endpoint 5 Maximal Payload Register	0x0000_0000
USBD_CFG5	USBD_BA+0x558	R/W	Endpoint 5 Configuration Register	0x0000_0000
USBD_CFGP5	USBD_BA+0x55C	R/W	Endpoint 5 Set Stall and Clear In/Out Ready Control Register	0x0000_0000
USBD_BUFSEG6	USBD_BA+0x560	R/W	Endpoint 6 Buffer Segmentation Register	0x0000_0000
USBD_MXPLD6	USBD_BA+0x564	R/W	Endpoint 6 Maximal Payload Register	0x0000_0000
USBD_CFG6	USBD_BA+0x568	R/W	Endpoint 6 Configuration Register	0x0000_0000
USBD_CFGP6	USBD_BA+0x56C	R/W	Endpoint 6 Set Stall and Clear In/Out Ready Control Register	0x0000_0000
USBD_BUFSEG7	USBD_BA+0x570	R/W	Endpoint 7 Buffer Segmentation Register	0x0000_0000
USBD_MXPLD7	USBD_BA+0x574	R/W	Endpoint 7 Maximal Payload Register	0x0000_0000
USBD_CFG7	USBD_BA+0x578	R/W	Endpoint 7 Configuration Register	0x0000_0000
USBD_CFGP7	USBD_BA+0x57C	R/W	Endpoint 7 Set Stall and Clear In/Out Ready Control Register	0x0000_0000
USBD_BUFSEG8	USBD_BA+0x580	R/W	Endpoint 8 Buffer Segmentation Register	0x0000_0000
USBD_MXPLD8	USBD_BA+0x584	R/W	Endpoint 8 Maximal Payload Register	0x0000_0000
USBD_CFG8	USBD_BA+0x588	R/W	Endpoint 8 Configuration Register	0x0000_0000
USBD_CFGP8	USBD_BA+0x58C	R/W	Endpoint 8 Set Stall and Clear In/Out Ready Control Register	0x0000_0000
USBD_BUFSEG9	USBD_BA+0x590	R/W	Endpoint 9 Buffer Segmentation Register	0x0000_0000
USBD_MXPLD9	USBD_BA+0x594	R/W	Endpoint 9 Maximal Payload Register	0x0000_0000
USBD_CFG9	USBD_BA+0x598	R/W	Endpoint 9 Configuration Register	0x0000_0000
USBD_CFGP9	USBD_BA+0x59C	R/W	Endpoint 9 Set Stall and Clear In/Out Ready Control Register	0x0000_0000
USBD_BUFSEG10	USBD_BA+0x5A0	R/W	Endpoint 10 Buffer Segmentation Register	0x0000_0000
USBD_MXPLD10	USBD_BA+0x5A4	R/W	Endpoint 10 Maximal Payload Register	0x0000_0000
USBD_CFG10	USBD_BA+0x5A8	R/W	Endpoint 10 Configuration Register	0x0000_0000
USBD_CFGP10	USBD_BA+0x5AC	R/W	Endpoint 10 Set Stall and Clear In/Out Ready Control Register	0x0000_0000
USBD_BUFSEG11	USBD_BA+0x5B0	R/W	Endpoint 11 Buffer Segmentation Register	0x0000_0000

USBD_MXPLD11	USBD_BA+0x5B4	R/W	Endpoint 11 Maximal Payload Register	0x0000_0000
USBD_CFG11	USBD_BA+0x5B8	R/W	Endpoint 11 Configuration Register	0x0000_0000
USBD_CFGP11	USBD_BA+0x5BC	R/W	Endpoint 11 Set Stall and Clear In/Out Ready Control Register	0x0000_0000
USBD_BUFSEG12	USBD_BA+0x5C0	R/W	Endpoint 12 Buffer Segmentation Register	0x0000_0000
USBD_MXPLD12	USBD_BA+0x5C4	R/W	Endpoint 12 Maximal Payload Register	0x0000_0000
USBD_CFG12	USBD_BA+0x5C8	R/W	Endpoint 12 Configuration Register	0x0000_0000
USBD_CFGP12	USBD_BA+0x5CC	R/W	Endpoint 12 Set Stall and Clear In/Out Ready Control Register	0x0000_0000
USBD_BUFSEG13	USBD_BA+0x5D0	R/W	Endpoint 13 Buffer Segmentation Register	0x0000_0000
USBD_MXPLD13	USBD_BA+0x5D4	R/W	Endpoint 13 Maximal Payload Register	0x0000_0000
USBD_CFG13	USBD_BA+0x5D8	R/W	Endpoint 13 Configuration Register	0x0000_0000
USBD_CFGP13	USBD_BA+0x5DC	R/W	Endpoint 13 Set Stall and Clear In/Out Ready Control Register	0x0000_0000
USBD_BUFSEG14	USBD_BA+0x5E0	R/W	Endpoint 14 Buffer Segmentation Register	0x0000_0000
USBD_MXPLD14	USBD_BA+0x5E4	R/W	Endpoint 14 Maximal Payload Register	0x0000_0000
USBD_CFG14	USBD_BA+0x5E8	R/W	Endpoint 14 Configuration Register	0x0000_0000
USBD_CFGP14	USBD_BA+0x5EC	R/W	Endpoint 14 Set Stall and Clear In/Out Ready Control Register	0x0000_0000
USBD_BUFSEG15	USBD_BA+0x5F0	R/W	Endpoint 15 Buffer Segmentation Register	0x0000_0000
USBD_MXPLD15	USBD_BA+0x5F4	R/W	Endpoint 15 Maximal Payload Register	0x0000_0000
USBD_CFG15	USBD_BA+0x5F8	R/W	Endpoint 15 Configuration Register	0x0000_0000
USBD_CFGP15	USBD_BA+0x5FC	R/W	Endpoint 15 Set Stall and Clear In/Out Ready Control Register	0x0000_0000
USBD_BUFSEG16	USBD_BA+0x600	R/W	Endpoint 16 Buffer Segmentation Register	0x0000_0000
USBD_MXPLD16	USBD_BA+0x604	R/W	Endpoint 16 Maximal Payload Register	0x0000_0000
USBD_CFG16	USBD_BA+0x608	R/W	Endpoint 16 Configuration Register	0x0000_0000
USBD_CFGP16	USBD_BA+0x60C	R/W	Endpoint 16 Set Stall and Clear In/Out Ready Control Register	0x0000_0000

Memory Type	Address	Size	Description
USBD_BA = 0x400C_0000			
USBD_SRAM	USBD_BA+0x100 ~ USBD_BA+0x4FF	1024 Bytes	The SRAM is used for the entire endpoints buffer. Refer to the endpoint SRAM structure and its description.

6.19.7 Register Description

USB Interrupt Enable Register (USBD_INTEN)

Register	Offset	R/W	Description	Reset Value
USBD_INTEN	USBD_BA+0x000	R/W	USB Device Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
INNAKEN	Reserved						WKEN
7	6	5	4	3	2	1	0
Reserved			SOFIEN	NEVWKIEN	VBDETIEN	USBIEN	BUSIEN

Bits	Description
[31:16]	Reserved Reserved.
[15]	INNAKEN Active NAK Function and Its Status in IN Token 0 = When the device responds NAK after receiving IN token, IN NAK status will not be updated to USBD_EPSTS0 and USBD_EPSTS1 register, so that the USB interrupt event will not be asserted. 1 = IN NAK status will be updated to USBD_EPSTS0 and USBD_EPSTS1 register and the USB interrupt event will be asserted when the device responds NAK after receiving IN token.
[14:9]	Reserved Reserved.
[8]	WKEN Wake-up Function Enable Bit 0 = USB wake-up function Disabled. 1 = USB wake-up function Enabled. Note: If woken up by any change by V_{BUS} state, VBDETIEN must be enabled. If woken up by receiving resume signal, BUSIEN must be enabled.
[7:5]	Reserved Reserved.
[4]	SOFIEN Start of Frame Interrupt Enable Bit 0 = SOF Interrupt Disabled. 1 = SOF Interrupt Enabled.
[3]	NEVWKIEN USB No-event-wake-up Interrupt Enable Bit 0 = No-event-wake-up Interrupt Disabled. 1 = No-event-wake-up Interrupt Enabled.
[2]	VBDETIEN V_{BUS} Detection Interrupt Enable Bit 0 = V_{BUS} detection Interrupt Disabled. 1 = V_{BUS} detection Interrupt Enabled.
[1]	USBIEN USB Event Interrupt Enable Bit 0 = USB event interrupt Disabled. 1 = USB event interrupt Enabled.

[0]	BUSIEN	Bus Event Interrupt Enable Bit 0 = BUS event interrupt Disabled. 1 = BUS event interrupt Enabled.

USB Interrupt Event Status Register (USB_D_INTSTS)

Register	Offset	R/W	Description	Reset Value
USB_D_INTSTS	USB_D_BA+0x004	R/W	USB Device Interrupt Event Status Register	0x0000_0000

31	30	29	28	27	26	25	24
SETUP	EPEVT14	EPEVT13	EPEVT12	EPEVT11	EPEVT10	EPEVT9	EPEVT8
23	22	21	20	19	18	17	16
EPEVT7	EPEVT6	EPEVT5	EPEVT4	EPEVT3	EPEVT2	EPEVT1	EPEVT0
15	14	13	12	11	10	9	8
Reserved						EPEVT16	EPEVT15
7	6	5	4	3	2	1	0
Reserved			SOFIF	NEVWKIF	VBDDETIF	USBIF	BUSIF

Bits	Description
[31]	SETUP Setup Event Status 0 = No Setup event. 1 = Setup event occurred, and it is cleared by writing 1 to USB_D_INTSTS[31].
[30]	EPEVT14 Endpoint 14's USB Event Status 0 = No event occurred in endpoint 14. 1 = USB event occurred on Endpoint 14; check USB_D_EPSTS1[27:24] to know which kind of USB event occurred, and it is cleared by writing 1 to USB_D_INTSTS[30] or USB_D_INTSTS[1].
[29]	EPEVT13 Endpoint 13's USB Event Status 0 = No event occurred in endpoint 13. 1 = USB event occurred on Endpoint 13; check USB_D_EPSTS1[23:20] to know which kind of USB event occurred, and it is cleared by writing 1 to USB_D_INTSTS[29] or USB_D_INTSTS[1].
[28]	EPEVT12 Endpoint 12's USB Event Status 0 = No event occurred in endpoint 12. 1 = USB event occurred on Endpoint 12; check USB_D_EPSTS1[19:16] to know which kind of USB event occurred, and it is cleared by writing 1 to USB_D_INTSTS[28] or USB_D_INTSTS[1].
[27]	EPEVT11 Endpoint 11's USB Event Status 0 = No event occurred in endpoint 11. 1 = USB event occurred on Endpoint 11; check USB_D_EPSTS1[15:12] to know which kind of USB event occurred, and it is cleared by writing 1 to USB_D_INTSTS[27] or USB_D_INTSTS[1].
[26]	EPEVT10 Endpoint 10's USB Event Status 0 = No event occurred in endpoint 10. 1 = USB event occurred on Endpoint 10, check USB_D_EPSTS1[11 :8] to know which kind of USB event occurred, and it is cleared by writing 1 to USB_D_INTSTS[26] or USB_D_INTSTS[1].
[25]	EPEVT9 Endpoint 9's USB Event Status 0 = No event occurred in endpoint 9. 1 = USB event occurred on Endpoint 9, check USB_D_EPSTS1[7 :4] to know which kind of USB event occurred, and it is cleared by writing 1 to USB_D_INTSTS[25] or USB_D_INTSTS[1].
[24]	EPEVT8 Endpoint 8's USB Event Status

		0 = No event occurred in endpoint 8. 1 = USB event occurred on Endpoint 8; check USBD_EPSTS1[3:0] to know which kind of USB event occurred, and it is cleared by writing 1 to USBD_INTSTS[24] or USBD_INTSTS[1].
[23]	EPEVT7	Endpoint 7's USB Event Status 0 = No event occurred in endpoint 7. 1 = USB event occurred on Endpoint 7, check USBD_EPSTS0[31:28] to know which kind of USB event occurred, and it is cleared by writing 1 to USBD_INTSTS[23] or USBD_INTSTS[1].
[22]	EPEVT6	Endpoint 6's USB Event Status 0 = No event occurred in endpoint 6. 1 = USB event occurred on Endpoint 6, check USBD_EPSTS0[27:24] to know which kind of USB event occurred, and it is cleared by writing 1 to USBD_INTSTS[22] or USBD_INTSTS[1].
[21]	EPEVT5	Endpoint 5's USB Event Status 0 = No event occurred in endpoint 5. 1 = USB event occurred on Endpoint 5, check USBD_EPSTS0[23:20] to know which kind of USB event occurred, and it is cleared by writing 1 to USBD_INTSTS[21] or USBD_INTSTS[1].
[20]	EPEVT4	Endpoint 4's USB Event Status 0 = No event occurred in endpoint 4. 1 = USB event occurred on Endpoint 4, check USBD_EPSTS0[19:16] to know which kind of USB event occurred, and it is cleared by writing 1 to USBD_INTSTS[20] or USBD_INTSTS[1].
[19]	EPEVT3	Endpoint 3's USB Event Status 0 = No event occurred in endpoint 3. 1 = USB event occurred on Endpoint 3, check USBD_EPSTS0[15:12] to know which kind of USB event occurred, and it is cleared by writing 1 to USBD_INTSTS[19] or USBD_INTSTS[1].
[18]	EPEVT2	Endpoint 2's USB Event Status 0 = No event occurred in endpoint 2. 1 = USB event occurred on Endpoint 2, check USBD_EPSTS0[11:8] to know which kind of USB event occurred, and it is cleared by writing 1 to USBD_INTSTS[18] or USBD_INTSTS[1].
[17]	EPEVT1	Endpoint 1's USB Event Status 0 = No event occurred in endpoint 1. 1 = USB event occurred on Endpoint 1, check USBD_EPSTS0[7:4] to know which kind of USB event occurred, and it is cleared by writing 1 to USBD_INTSTS[17] or USBD_INTSTS[1].
[16]	EPEVT0	Endpoint 0's USB Event Status 0 = No event occurred in endpoint 0. 1 = USB event occurred on Endpoint 0, check USBD_EPSTS0[3:0] to know which kind of USB event occurred, and it is cleared by writing 1 to USBD_INTSTS[16] or USBD_INTSTS[1].
[15:10]	Reserved	Reserved.
[9]	EPEVT16	Endpoint 16's USB Event Status 0 = No event occurred in endpoint 16. 1 = USB event occurred on Endpoint 16; check USBD_EPSTS2[3:0] to know which kind of USB event occurred, and it is cleared by writing 1 to USBD_INTSTS[30] or USBD_INTSTS[1].
[8]	EPEVT15	Endpoint 15's USB Event Status 0 = No event occurred in endpoint 15. 1 = USB event occurred on Endpoint 15; check USBD_EPSTS1[31:28] to know which kind of USB event occurred, and it is cleared by writing 1 to USBD_INTSTS[8] or USBD_INTSTS[1].
[7:5]	Reserved	Reserved.
[4]	SOFIF	Start of Frame Interrupt Status 0 = SOF event did not occur.

		1 = SOF event occurred, and it is cleared by writing 1 to USBD_INTSTS[4].
[3]	NEVWKIF	No-event-wake-up Interrupt Status 0 = NEVWK event did not occur. 1 = No-event-wake-up event occurred, and it is cleared by writing 1 to USBD_INTSTS[3].
[2]	VBDETIF	V_{BUS} Detection Interrupt Status 0 = There is not attached/detached event in the USB. 1 = There is attached/detached event in the USB bus and it is cleared by writing 1 to USBD_INTSTS[2].
[1]	USBIF	USB Event Interrupt Status The USB event includes the SETUP Token, IN Token, OUT ACK, ISO IN, or ISO OUT events in the bus. 0 = No USB event occurred. 1 = USB event occurred; check EPSTS0~11[3:0] to know which kind of USB event occurred, and it is cleared by writing 1 to USBD_INTSTS[1] or EPSTS0~11 and SETUP (USBD_INTSTS[31]).
[0]	BUSIF	BUS Interrupt Status The BUS event means that there is one of the suspension or the resume function in the bus. 0 = No BUS event occurred. 1 = Bus event occurred; check USBD_ATTR[3:0] to know which kind of bus event occurred, and it is cleared by writing 1 to USBD_INTSTS[0].

USB Device Function Address Register (USBD_FADDR)

A 7-bit value is used as the address of a device on the USB BUS.

Register	Offset	R/W	Description	Reset Value
USBD_FADDR	USBD_BA+0x008	R/W	USB Device Function Address Register	0x0000_0000

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

Bits	Description	
[31:7]	Reserved	Reserved.
[6:0]	FADDR	USB Device Function Address

USB Endpoint Status Register (USBD_EPSTS)

Register	Offset	R/W	Description	Reset Value
USBD_EPSTS	USBD_BA+0x00C	R	USB Device Endpoint Status Register	0x0000_0000

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

Bits	Description
[31:8]	Reserved Reserved.
[7]	OV Overrun It indicates that the received data is more than the maximum payload number or not. 0 = No overrun. 1 = Out Data is more than the Max Payload in MXPLD register or the Setup Data is more than 8 Bytes.
[6:0]	Reserved Reserved.

USB Bus Status and Attribution Register (USBD_ATTR)

Register	Offset	R/W	Description	Reset Value
USBD_ATTR	USBD_BA+0x010	R/W	USB Device Bus Status and Attribution Register	0x0000_0040

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		L1RESUME	L1SUSPEND	LPMACK	BYTEM	PWRDN	DPPUEN
7	6	5	4	3	2	1	0
USBEN	Reserved	RWAKEUP	PHYEN	TOUT	RESUME	SUSPEND	USBST

Bits	Description
[31:13]	Reserved Reserved.
[13]	L1RESUME LPM L1 Resume (Read Only) 0 = Bus has no LPM L1 state resume. 1 = LPM L1 state resume from LPM L1 state suspend.
[12]	L1SUSPEND LPM L1 Suspend (Read Only) 0 = Bus has no L1 state suspend. 1 = This bit is set by the hardware when LPM command to enter the L1 state is successfully received and acknowledged.
[11]	LPMACK LPM Token Acknowledge Enable Bit The NYET/ACK will be returned only on a successful LPM transaction if no errors in both the EXT token and the LPM token and a valid bLinkState = 0001 (L1) is received, else ERROR and STALL will be returned automatically, respectively. 0= The valid LPM Token will be NYET. 1= The valid LPM Token will be ACK.
[10]	BYTEM CPU Access USB SRAM Size Mode Selection 0 = Word mode: The size of the transfer from CPU to USB SRAM can be Word only. 1 = Byte mode: The size of the transfer from CPU to USB SRAM can be Byte only.
[9]	PWRDN Power-down PHY Transceiver, Low Active 0 = Power-down related circuit of PHY transceiver. 1 = Turn-on related circuit of PHY transceiver.
[8]	DPPUEN Pull-up Resistor on USB_DP Enable Bit 0 = Pull-up resistor in USB_D+ bus Disabled. 1 = Pull-up resistor in USB_D+ bus Active.
[7]	USBEN USB Controller Enable Bit 0 = USB Controller Disabled. 1 = USB Controller Enabled.
[6]	Reserved Reserved.

[5]	RWAKEUP	Remote Wake-up 0 = Release the USB bus from K state. 1 = Force USB bus to K (USB_D+ low, USB_D-: high) state, used for remote wake-up.
[4]	PHYEN	PHY Transceiver Function Enable Bit 0 = PHY transceiver function Disabled. 1 = PHY transceiver function Enabled.
[3]	TOUT	Time-out Status (Read Only) 0 = No time-out. 1 = No Bus response more than 18 bits time ($\frac{1}{12MHz} \times 18 = 1.5\mu S$).
[2]	RESUME	Resume Status (Read Only) 0 = No bus resume. 1 = Resume from suspend.
[1]	SUSPEND	Suspend Status (Read Only) 0 = Bus no suspend. 1 = Bus idle more than 3ms, either cable is plugged out or host is sleeping.
[0]	USBRST	USB Reset Status (Read Only) 0 = Bus no reset. 1 = Bus reset when SE0 (single-ended 0) more than 2.5us.

USB Device V_{BUS} Detection Register (USBD_VBUSDET)

Register	Offset	R/W	Description	Reset Value
USBD_VBUSDET	USBD_BA+0x014	R	USB Device V _{BUS} Detection Register	0x0000_0000

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

Bits	Description	
[31:1]	Reserved	Reserved.
[0]	VBUSDET	Device V_{BUS} Detection 0 = Controller is not attached to the USB host. 1 = Controller is attached to the USB host.

USB SETUP Token Buffer Segmentation Register (USBD_STBUFSEG)

Register	Offset	R/W	Description	Reset Value
USBD_STBUFSEG	USBD_BA+0x018	R/W	SETUP Token Buffer Segmentation Register	0x0000_0000

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

Bits	Description
[31:9]	Reserved Reserved.
[8:3]	STBUFSEG SETUP Token Buffer Segmentation It is used to indicate the offset address for the SETUP token with the USB Device SRAM starting address. The effective starting address is USBD_SRAM address + {STBUFSE.G. 3'b000} Where the USBD_SRAM address = USBD_BA+0x100h. Note: It is used for SETUP token only.
[2:0]	Reserved Reserved.

USB Endpoint Status Register 0 (USBD_EPSTS0)

Register	Offset	R/W	Description	Reset Value
USBD_EPSTS0	USBD_BA+0x020	R	USB Device Endpoint Status Register 0	0x0000_0000

31	30	29	28	27	26	25	24
EPSTS7				EPSTS6			
23	22	21	20	19	18	17	16
EPSTS5				EPSTS4			
15	14	13	12	11	10	9	8
EPSTS3				EPSTS2			
7	6	5	4	3	2	1	0
EPSTS1				EPSTS0			

Bits	Description
[31:28]	EPSTS7 Endpoint 7 Status These bits are used to indicate the current status of this endpoint 0000 = In ACK. 0001 = In NAK. 0010 = Out Packet Data0 ACK. 0110 = Out Packet Data1 ACK. 0111 = Isochronous transfer end.
[27:24]	EPSTS6 Endpoint 6 Status These bits are used to indicate the current status of this endpoint 0000 = In ACK. 0001 = In NAK. 0010 = Out Packet Data0 ACK. 0110 = Out Packet Data1 ACK. 0111 = Isochronous transfer end.
[23:20]	EPSTS5 Endpoint 5 Status These bits are used to indicate the current status of this endpoint 0000 = In ACK. 0001 = In NAK. 0010 = Out Packet Data0 ACK. 0110 = Out Packet Data1 ACK. 0111 = Isochronous transfer end.

[19:16]	EPSTS4	Endpoint 4 Status These Bits Are Used To Indicate The Current Status Of This Endpoint 0000 = In ACK. 0001 = In NAK. 0010 = Out Packet Data0 ACK. 0110 = Out Packet Data1 ACK. 0111 = Isochronous Transfer End.
[15:12]	EPSTS3	Endpoint 3 Status These bits are used to indicate the current status of this endpoint 0000 = In ACK. 0001 = In NAK. 0010 = Out Packet Data0 ACK. 0110 = Out Packet Data1 ACK. 0111 = Isochronous transfer end.
[11:8]	EPSTS2	Endpoint 2 Status These bits are used to indicate the current status of this endpoint 0000 = In ACK. 0001 = In NAK. 0010 = Out Packet Data0 ACK. 0110 = Out Packet Data1 ACK. 0111 = Isochronous transfer end.
[7:4]	EPSTS1	Endpoint 1 Status These bits are used to indicate the current status of this endpoint 0000 = In ACK. 0001 = In NAK. 0010 = Out Packet Data0 ACK. 0110 = Out Packet Data1 ACK. 0111 = Isochronous transfer end.
[3:0]	EPSTS0	Endpoint 0 Status These bits are used to indicate the current status of this endpoint 0000 = In ACK. 0001 = In NAK. 0010 = Out Packet Data0 ACK. 0110 = Out Packet Data1 ACK. 0111 = Isochronous transfer end.

USB Endpoint Status Register 1 (USBD_EPSTS1)

Register	Offset	R/W	Description	Reset Value
USBD_EPSTS1	USBD_BA+0x024	R	USB Device Endpoint Status Register 1	0x0000_0000

31	30	29	28	27	26	25	24
EPSTS15				EPSTS14			
23	22	21	20	19	18	17	16
EPSTS13				EPSTS12			
15	14	13	12	11	10	9	8
EPSTS11				EPSTS10			
7	6	5	4	3	2	1	0
EPSTS9				EPSTS8			

Bits	Description
[31:28]	EPSTS15 Endpoint 15 Status These bits are used to indicate the current status of this endpoint 0000 = In ACK. 0001 = In NAK. 0010 = Out Packet Data0 ACK. 0110 = Out Packet Data1 ACK. 0111 = Isochronous transfer end.
[27:24]	EPSTS14 Endpoint 14 Status These bits are used to indicate the current status of this endpoint 0000 = In ACK. 0001 = In NAK. 0010 = Out Packet Data0 ACK. 0110 = Out Packet Data1 ACK. 0111 = Isochronous transfer end.
[23:20]	EPSTS13 Endpoint 13 Status These bits are used to indicate the current status of this endpoint 0000 = In ACK. 0001 = In NAK. 0010 = Out Packet Data0 ACK. 0110 = Out Packet Data1 ACK. 0111 = Isochronous transfer end.
[19:16]	EPSTS12 Endpoint 12 Status These bits are used to indicate the current status of this endpoint 0000 = In ACK. 0001 = In NAK. 0010 = Out Packet Data0 ACK. 0110 = Out Packet Data1 ACK. 0111 = Isochronous transfer end.

[15:12]	EPSTS11	Endpoint 11 Status These bits are used to indicate the current status of this endpoint 0000 = In ACK. 0001 = In NAK. 0010 = Out Packet Data0 ACK. 0110 = Out Packet Data1 ACK. 0111 = Isochronous transfer end.
[11:8]	EPSTS10	Endpoint 10 Status These bits are used to indicate the current status of this endpoint 0000 = In ACK. 0001 = In NAK. 0010 = Out Packet Data0 ACK. 0110 = Out Packet Data1 ACK. 0111 = Isochronous transfer end.
[7:4]	EPSTS9	Endpoint 9 Status These bits are used to indicate the current status of this endpoint 0000 = In ACK. 0001 = In NAK. 0010 = Out Packet Data0 ACK. 0110 = Out Packet Data1 ACK. 0111 = Isochronous transfer end.
[3:0]	EPSTS8	Endpoint 8 Status These bits are used to indicate the current status of this endpoint 0000 = In ACK. 0001 = In NAK. 0010 = Out Packet Data0 ACK. 0110 = Out Packet Data1 ACK. 0111 = Isochronous transfer end.

USB Endpoint Status Register 2 (USBD_EPSTS2)

Register	Offset	R/W	Description	Reset Value
USBD_EPSTS2	USBD_BA+0x028	R	USB Device Endpoint Status Register 2	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				EPSTS16			

Bits	Description	
[31:4]	Reserved	Reserved.
[3:0]	EPSTS16	Endpoint 16 Status These bits are used to indicate the current status of this endpoint 0000 = In ACK. 0001 = In NAK. 0010 = Out Packet Data0 ACK. 0110 = Out Packet Data1 ACK. 0111 = Isochronous transfer end.

USB LPM Attribution Register (USBD_LPMATTR)

Register	Offset	R/W	Description	Reset Value
USBD_LPMATTR	USBD_BA+0x088	R	USB LPM Attribution Register	0x0000_0000

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

Bits	Description
[31:9]	Reserved Reserved.
[8]	LPMRWAKUP LPM Remote Wake-up This bit contains the bRemoteWake value received with last ACK LPM Token
[7:4]	LPMBESL LPM Best Effort Service Latency These bits contain the BESL value received with last ACK LPM Token
[3:0]	LPMLINKSTS LPM Link State These bits contain the bLinkState received with last ACK LPM Token

USB Frame Number Register (USBD_FN)

Register	Offset	R/W	Description	Reset Value
USBD_FN	USBD_BA+0x08C	R	USB Frame Number Register	0x0000_0XXX

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

Bits	Description	
[31:11]	Reserved	Reserved.
[10:0]	FN	Frame Number These bits contain the 11-bits frame number in the last received SOF packet. Note: Suggest to read USBD_FN after USBD_INTSTS[4] SOFIF interrupt is triggered and cleared.

USB Drive SE0 Register (USBD_SE0)

Register	Offset	R/W	Description	Reset Value
USBD_SE0	USBD_BA+0x090	R/W	USB Device Drive SE0 Control Register	0x0000_0001

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							SE0

Bits	Description	
[31:1]	Reserved	Reserved.
[0]	SE0	Drive Single Ended Zero in USB Bus The Single Ended Zero (SE0) is when both lines (USB_D+ and USB_D-) are being pulled low. 0 = Normal operation. 1 = Force USB PHY transceiver to drive SE0.

USB Buffer Segmentation Register (USB_BUFSEG0~16)

Register	Offset	R/W	Description	Reset Value
USBD_BUFSEG0	USBD_BA+0x500	R/W	Endpoint 0 Buffer Segmentation Register	0x0000_0000
USBD_BUFSEG1	USBD_BA+0x510	R/W	Endpoint 1 Buffer Segmentation Register	0x0000_0000
USBD_BUFSEG2	USBD_BA+0x520	R/W	Endpoint 2 Buffer Segmentation Register	0x0000_0000
USBD_BUFSEG3	USBD_BA+0x530	R/W	Endpoint 3 Buffer Segmentation Register	0x0000_0000
USBD_BUFSEG4	USBD_BA+0x540	R/W	Endpoint 4 Buffer Segmentation Register	0x0000_0000
USBD_BUFSEG5	USBD_BA+0x550	R/W	Endpoint 5 Buffer Segmentation Register	0x0000_0000
USBD_BUFSEG6	USBD_BA+0x560	R/W	Endpoint 6 Buffer Segmentation Register	0x0000_0000
USBD_BUFSEG7	USBD_BA+0x570	R/W	Endpoint 7 Buffer Segmentation Register	0x0000_0000
USBD_BUFSEG8	USBD_BA+0x580	R/W	Endpoint 8 Buffer Segmentation Register	0x0000_0000
USBD_BUFSEG9	USBD_BA+0x590	R/W	Endpoint 9 Buffer Segmentation Register	0x0000_0000
USBD_BUFSEG10	USBD_BA+0x5A0	R/W	Endpoint 10 Buffer Segmentation Register	0x0000_0000
USBD_BUFSEG11	USBD_BA+0x5B0	R/W	Endpoint 11 Buffer Segmentation Register	0x0000_0000
USBD_BUFSEG12	USBD_BA+0x5C0	R/W	Endpoint 12 Buffer Segmentation Register	0x0000_0000
USBD_BUFSEG13	USBD_BA+0x5D0	R/W	Endpoint 13 Buffer Segmentation Register	0x0000_0000
USBD_BUFSEG14	USBD_BA+0x5E0	R/W	Endpoint 14 Buffer Segmentation Register	0x0000_0000
USBD_BUFSEG15	USBD_BA+0x5F0	R/W	Endpoint 15 Buffer Segmentation Register	0x0000_0000
USBD_BUFSEG16	USBD_BA+0x600	R/W	Endpoint 16 Buffer Segmentation Register	0x0000_0000

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

Bits	Description
[31:9]	Reserved Reserved.
[8:3]	BUFSEG Endpoint Buffer Segmentation It is used to indicate the offset address for each endpoint with the USB SRAM starting address. The effective

		starting address of the endpoint is: USBD_SRAM address + { BUFSE.G. 3'b000} Where the USBD_SRAM address = USBD_BA+0x100h. Refer to the endpoint SRAM structure and its description.
[2:0]	Reserved	Reserved.

USB Maximal Payload Register (USB_MXPLD0~16)

Register	Offset	R/W	Description	Reset Value
USBD_MXPLD0	USBD_BA+0x504	R/W	Endpoint 0 Maximal Payload Register	0x0000_0000
USBD_MXPLD1	USBD_BA+0x514	R/W	Endpoint 1 Maximal Payload Register	0x0000_0000
USBD_MXPLD2	USBD_BA+0x524	R/W	Endpoint 2 Maximal Payload Register	0x0000_0000
USBD_MXPLD3	USBD_BA+0x534	R/W	Endpoint 3 Maximal Payload Register	0x0000_0000
USBD_MXPLD4	USBD_BA+0x544	R/W	Endpoint 4 Maximal Payload Register	0x0000_0000
USBD_MXPLD5	USBD_BA+0x554	R/W	Endpoint 5 Maximal Payload Register	0x0000_0000
USBD_MXPLD6	USBD_BA+0x564	R/W	Endpoint 6 Maximal Payload Register	0x0000_0000
USBD_MXPLD7	USBD_BA+0x574	R/W	Endpoint 7 Maximal Payload Register	0x0000_0000
USBD_MXPLD8	USBD_BA+0x584	R/W	Endpoint 8 Maximal Payload Register	0x0000_0000
USBD_MXPLD9	USBD_BA+0x594	R/W	Endpoint 9 Maximal Payload Register	0x0000_0000
USBD_MXPLD10	USBD_BA+0x5A4	R/W	Endpoint 10 Maximal Payload Register	0x0000_0000
USBD_MXPLD11	USBD_BA+0x5B4	R/W	Endpoint 11 Maximal Payload Register	0x0000_0000
USBD_MXPLD12	USBD_BA+0x5C4	R/W	Endpoint 12 Maximal Payload Register	0x0000_0000
USBD_MXPLD13	USBD_BA+0x5D4	R/W	Endpoint 13 Maximal Payload Register	0x0000_0000
USBD_MXPLD14	USBD_BA+0x5E4	R/W	Endpoint 14 Maximal Payload Register	0x0000_0000
USBD_MXPLD15	USBD_BA+0x5F4	R/W	Endpoint 15 Maximal Payload Register	0x0000_0000
USBD_MXPLD16	USBD_BA+0x604	R/W	Endpoint 16 Maximal Payload Register	0x0000_0000

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

Bits	Description
[31:9]	Reserved Reserved.
[8:0]	MXPLD Maximal Payload Define the data length which is transmitted to host (IN token) or the actual data length which is received from the host (OUT token). It also used to indicate that the endpoint is ready to be transmitted in IN token or

		received in OUT token. (1) When the register is written by CPU, For IN token, the value of MXPLD is used to define the data length to be transmitted and indicate the data buffer is ready. For OUT token, it means that the controller is ready to receive data from the host and the value of MXPLD is the maximal data length comes from host. (2) When the register is read by CPU, For IN token, the value of MXPLD is indicated by the data length be transmitted to host For OUT token, the value of MXPLD is indicated the actual data length receiving from host. Note: Once MXPLD is written, the data packets will be transmitted/received immediately after IN/OUT token arrived.

USB Configuration Register (USB_CFG0~16)

Register	Offset	R/W	Description	Reset Value
USBD_CFG0	USBD_BA+0x508	R/W	Endpoint 0 Configuration Register	0x0000_0000
USBD_CFG1	USBD_BA+0x518	R/W	Endpoint 1 Configuration Register	0x0000_0000
USBD_CFG2	USBD_BA+0x528	R/W	Endpoint 2 Configuration Register	0x0000_0000
USBD_CFG3	USBD_BA+0x538	R/W	Endpoint 3 Configuration Register	0x0000_0000
USBD_CFG4	USBD_BA+0x548	R/W	Endpoint 4 Configuration Register	0x0000_0000
USBD_CFG5	USBD_BA+0x558	R/W	Endpoint 5 Configuration Register	0x0000_0000
USBD_CFG6	USBD_BA+0x568	R/W	Endpoint 6 Configuration Register	0x0000_0000
USBD_CFG7	USBD_BA+0x578	R/W	Endpoint 7 Configuration Register	0x0000_0000
USBD_CFG8	USBD_BA+0x588	R/W	Endpoint 8 Configuration Register	0x0000_0000
USBD_CFG9	USBD_BA+0x598	R/W	Endpoint 9 Configuration Register	0x0000_0000
USBD_CFG10	USBD_BA+0x5A8	R/W	Endpoint 10 Configuration Register	0x0000_0000
USBD_CFG11	USBD_BA+0x5B8	R/W	Endpoint 11 Configuration Register	0x0000_0000
USBD_CFG12	USBD_BA+0x5C8	R/W	Endpoint 12 Configuration Register	0x0000_0000
USBD_CFG13	USBD_BA+0x5D8	R/W	Endpoint 13 Configuration Register	0x0000_0000
USBD_CFG14	USBD_BA+0x5E8	R/W	Endpoint 14 Configuration Register	0x0000_0000
USBD_CFG15	USBD_BA+0x5F8	R/W	Endpoint 15 Configuration Register	0x0000_0000
USBD_CFG16	USBD_BA+0x608	R/W	Endpoint 16 Configuration Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved						CSTALL	Reserved
7	6	5	4	3	2	1	0
DSQSYNC	STATE		ISOCH	EPNUM			

Bits	Description
[31:10]	Reserved Reserved.
[9]	CSTALL Clear STALL Response 0 = Disable the device to clear the STALL handshake in setup stage.

		1 = Clear the device to response STALL handshake in setup stage.
[8]	Reserved	Reserved.
[7]	DSQSYNC	Data Sequence Synchronization 0 = DATA0 PID. 1 = DATA1 PID. Note: It is used to specify the DATA0 or DATA1 PID in the following IN token transaction. Hardware will toggle automatically in IN token base on this bit.
[6:5]	STATE	Endpoint State 00 = Endpoint is Disabled. 01 = Out endpoint. 10 = IN endpoint. 11 = Undefined.
[4]	ISOCH	Isochronous Endpoint This bit is used to set the endpoint as Isochronous endpoint, no handshake. 0 = No Isochronous endpoint. 1 = Isochronous endpoint.
[3:0]	EPNUM	Endpoint Number These bits are used to define the endpoint number of the current endpoint.

USB Extra Configuration Register (USB_CFGP0~16)

Register	Offset	R/W	Description	Reset Value
USBD_CFGP0	USBD_BA+0x50C	R/W	Endpoint 0 Set Stall and Clear In/Out Ready Control Register	0x0000_0000
USBD_CFGP1	USBD_BA+0x51C	R/W	Endpoint 1 Set Stall and Clear In/Out Ready Control Register	0x0000_0000
USBD_CFGP2	USBD_BA+0x52C	R/W	Endpoint 2 Set Stall and Clear In/Out Ready Control Register	0x0000_0000
USBD_CFGP3	USBD_BA+0x53C	R/W	Endpoint 3 Set Stall and Clear In/Out Ready Control Register	0x0000_0000
USBD_CFGP4	USBD_BA+0x54C	R/W	Endpoint 4 Set Stall and Clear In/Out Ready Control Register	0x0000_0000
USBD_CFGP5	USBD_BA+0x55C	R/W	Endpoint 5 Set Stall and Clear In/Out Ready Control Register	0x0000_0000
USBD_CFGP6	USBD_BA+0x56C	R/W	Endpoint 6 Set Stall and Clear In/Out Ready Control Register	0x0000_0000
USBD_CFGP7	USBD_BA+0x57C	R/W	Endpoint 7 Set Stall and Clear In/Out Ready Control Register	0x0000_0000
USBD_CFGP8	USBD_BA+0x58C	R/W	Endpoint 8 Set Stall and Clear In/Out Ready Control Register	0x0000_0000
USBD_CFGP9	USBD_BA+0x59C	R/W	Endpoint 9 Set Stall and Clear In/Out Ready Control Register	0x0000_0000
USBD_CFGP10	USBD_BA+0x5AC	R/W	Endpoint 10 Set Stall and Clear In/Out Ready Control Register	0x0000_0000
USBD_CFGP11	USBD_BA+0x5BC	R/W	Endpoint 11 Set Stall and Clear In/Out Ready Control Register	0x0000_0000
USBD_CFGP12	USBD_BA+0x5CC	R/W	Endpoint 12 Set Stall and Clear In/Out Ready Control Register	0x0000_0000
USBD_CFGP13	USBD_BA+0x5DC	R/W	Endpoint 13 Set Stall and Clear In/Out Ready Control Register	0x0000_0000
USBD_CFGP14	USBD_BA+0x5EC	R/W	Endpoint 14 Set Stall and Clear In/Out Ready Control Register	0x0000_0000
USBD_CFGP15	USBD_BA+0x5FC	R/W	Endpoint 15 Set Stall and Clear In/Out Ready Control Register	0x0000_0000
USBD_CFGP16	USBD_BA+0x60C	R/W	Endpoint 16 Set Stall and Clear In/Out Ready Control Register	0x0000_0000

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

Bits	Description
[31:2]	Reserved Reserved.
[1]	SSTALL Set STALL 0 = Disable the device to response STALL.

		1 = Set the device to respond STALL automatically.
[0]	CLRRDY	Clear Ready When the USBD_MXPLD0~16 register is set by user, it means that the endpoint is ready to transmit or receive data. If the user wants to disable this transaction before the transaction start, users can set this bit to 1 to disable it and it is auto clear to 0. For IN token, write '1' to clear the IN token had ready to transmit the data to USB. For OUT token, write '1' to clear the OUT token had ready to receive the data from USB. This bit is write 1 only and is always 0 when it is read back.

6.20 Controller Area Network with Feasibility Data Rate (CAN FD)

6.20.1 Overview

The CAN FD controller performs communication according to ISO 11898-1:2015 and needs to be connected to additional transceiver hardware for the CAN bus physical layer.

The CAN FD controller consists of one CAN Core, Memory access control and arbiter, Tx Handler, Rx Handler, a shared Message RAM memory and a 32-bit AHB interface for control and configuration registers.

The message storage is intended to be a single-ported Message RAM outside of the CAN Core module. It is connected to the CAN Core via the memory control interface. The Message RAM implements filters, receive FIFOs, transmit event FIFOs and transmit FIFOs.

All functions concerning the handling of messages are implemented by the Rx Handler and the Tx Handler. The Rx Handler manages message acceptance filtering, the transfer of received messages from the CAN Core to the Message RAM as well as providing received message status information. The Tx Handler is responsible for the transfer of transmit messages from the Message RAM to the CAN Core as well as providing transmitted status information.

The controller's clock domain concept allows the separation among CAN Core clock and the AHB clock.

6.20.2 Features

- Conform with CAN protocol version 2.0 part A, B and ISO 11898-1: 2015
- CAN FD with up to 64 data bytes supported
- CAN Error Logging
- AUTOSAR support
- SAE J1939 support
- Improved acceptance filtering
- Two configurable Receive FIFOs
- Separate signaling on reception of High Priority Messages
- Configurable Transmit FIFO, Transmit Queue, Transmit Event FIFO
- Direct Message RAM access for CPU
- Programmable loop-back test mode
- Maskable module interrupts
- Two clock domains (CAN Core clock and AHB clock)
- Power-down support

6.20.3 Block Diagram

The controller interfaces with the AMBA AHB bus. Figure 6.20-1 shows the block diagram of the controller.

CAN Core

CAN Protocol Controller and Rx/Tx shift register that handle all 2.0 part A, B and ISO 11898-1:2015 protocol functions. Supports 11-bit and 29-bit identifiers.

Sync

Synchronizes signals from the AHB clock domain to the CAN Core clock domain and vice versa

Control and Configuration Registers

All registers and related configuration used to control and to configure the CAN Core.

Interrupt & power down control

The CAN FD controller provides two interrupt lines CANFDx_INT0 and CANFDx_INT1. By programming EINT0 and EINT1 bits in CANFD_ILE register, the interrupt lines can be enabled or disabled separately.

The CAN FD controller can be set into power down controlled by both CANFD_PDEN (SYS_ALTCTL[6]) and CANFDCKSTP (SYS_ALTCTL[7]).

Memory access control and Arbiter

The arbiter is responsible for the Message RAM access authority between the CPU and the interface of Tx/Rx Handler to an external 32-bit Message RAM.

Message RAM (SRAM)

Stores message objects and Identifier masks. The maximum Message RAM size is 256 words.

Tx/Rx Message Handler

State Machine that controls the data transfer between the Rx/Tx Shift Register of the CAN Core and the Message RAM as well as the generation of interrupts as programmed in the Control and Configuration registers.

AHB Interface

Connect the CAN FD controller to the AHB bus.

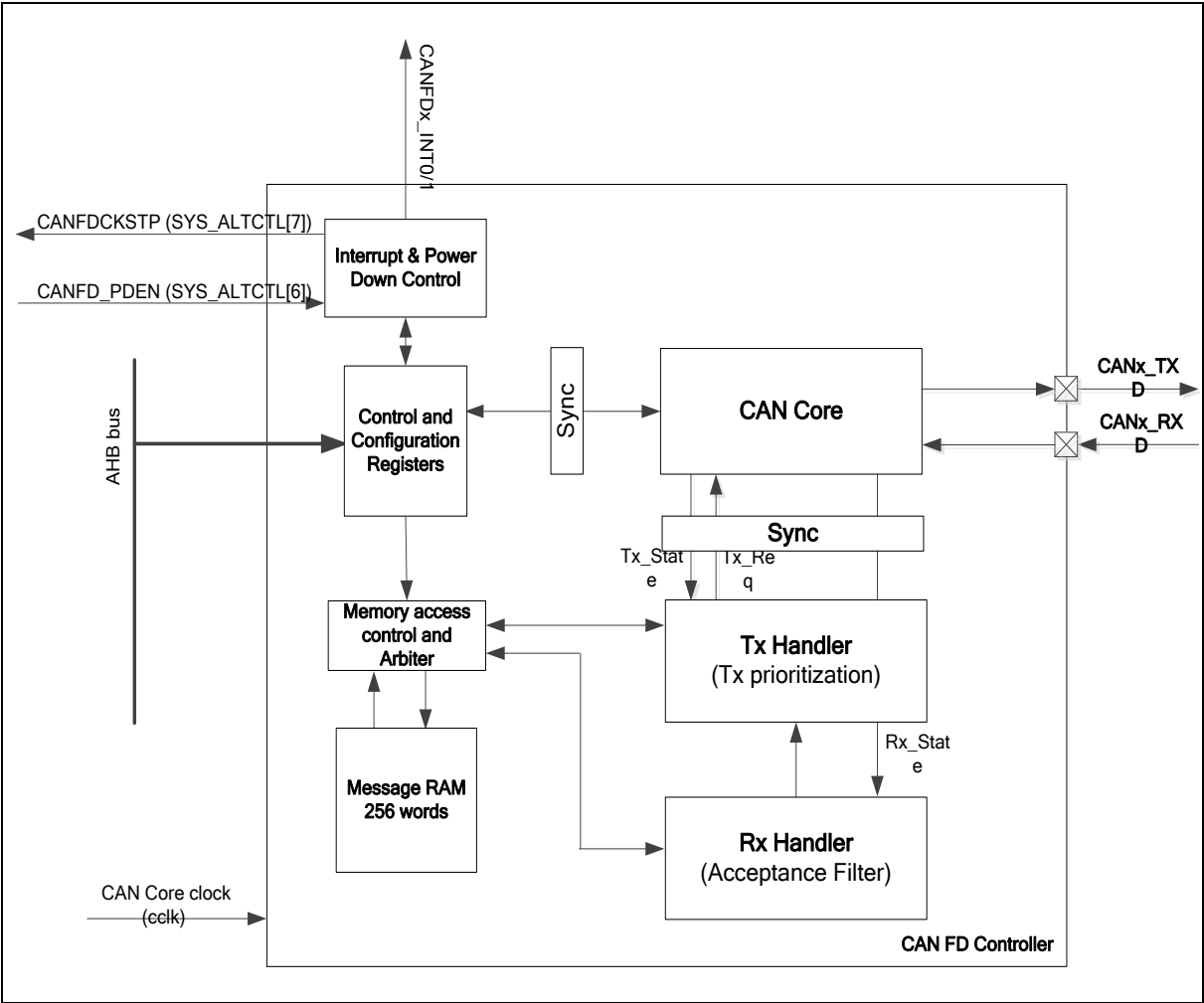


Figure 6.20-1 CAN FD Controller Block Diagram

6.20.4 Basic Configuration

6.20.4.1 CAN Basic Configuration

- Clock source Configuration
 - Enable CAN FD controller clock (CANFD0CKEN (CLKAHBC[23])).
- Reset Configuration
 - Reset CAN FD controller (CAN0RST (SYS_IPRST1[24])).
- Pin Configuration

Group	Pin Name	GPIO	MFP
CANFD0	CAN0_RXD	GPA4	MFP2
		GPB12	MFP9
		GPC5	MFP10
	CAN0_TXD	GPA5	MFP2
		GPB13	MFP9

		GPC6	MFP10

6.20.5 Functional Description

6.20.5.1 Operation Mode

Software Initialization

Software initialization is started by setting bit INIT (CANFD_CCCR[0]) bit, either by software or by a hardware reset, when an uncorrected bit error was detected in the Message RAM, or by going Bus_Off. While INIT (CANFD_CCCR[0]) bit is set, message transfer from and to the CAN bus is stopped, the status of the CAN bus output CANx_TXD pin is recessive (HIGH). The counters of the Error Management Logic EML are unchanged. Setting INIT (CANFD_CCCR[0]) bit does not change any configuration register. Resetting INIT (CANFD_CCCR[0]) bit finishes the software initialization. Afterwards the Bit Stream Processor (BSP) synchronizes itself to the data transfer on the CAN bus by waiting for the occurrence of a sequence of 11 consecutive recessive bits (=Bus_Idle) before it can take part in bus activities and start the message transfer.

Access to the CAN FD configuration registers is only enabled when both bits INIT (CANFD_CCCR[0]) and CCE (CANFD_CCCR[1]) are set (protected write). CCE (CANFD_CCCR[1]) can only be set/reset while INIT (CANFD_CCCR[0]) = 1. CCE (CANFD_CCCR[1]) is automatically reset when INIT (CANFD_CCCR[0]) is reset.

The following registers are reset when CCE (CANFD_CCCR[1]) is set:

- HPMS - High Priority Message Status
- RXF0S - Rx FIFO 0 Status
- RXF1S - Rx FIFO 1 Status
- TXFQS - Tx FIFO/Queue Status
- TXBRP - Tx Buffer Request Pending
- TXBTO - Tx Buffer Transmission Occurred
- TXBCF - Tx Buffer Cancellation Finished
- TXEFS - Tx Event FIFO Status

The Timeout Counter value TOC (CANFD_TOCV[15:0]) is preset to the value configured by TOP (CANFD_TOCC[31:16]) when CCE (CANFD_CCCR[1]) is set.

In addition the state machines of the Tx Handler and Rx Handler are held in idle state while CCE (CANFD_CCCR[1]) bit is set.

The following registers are only writeable while CCE (CANFD_CCCR[1]) bit is reset.

- TXBAR - Tx Buffer Add Request
- TXBCR - Tx Buffer Cancellation Request

TEST (CANFD_CCCR[7]) bit and MON (CANFD_CCCR[5]) bit can only be set by the software while both of INIT (CANFD_CCCR[0]) bit and CCE (CANFD_CCCR[1]) bit are set. Both bits may be reset at any time. DAR (CANFD_CCCR[6]) can only be set/reset while both INIT (CANFD_CCCR[0]) and CCE (CANFD_CCCR[1]) are set.

Normal Operation

Once the CAN FD controller is initialized and INIT (CANFD_CCCR[0]) is reset to 0, the CAN FD controller synchronizes itself to the CAN bus and is ready for communication.

After passing the acceptance filtering, received messages including Message ID and DLC are stored

into a dedicated Rx Buffer or into Rx FIFO 0 or Rx FIFO 1.

For messages to be transmitted dedicated Tx Buffers and/or a Tx FIFO or a Tx queue can be initialized or updated. Automated transmission on reception of remote frames is not implemented.

CAN FD Operation

There are two variants in the CAN FD frame transmission:

- (1). The CAN FD frame without bit rate switching.
- (2). The CAN FD frame where control field, data field, and CRC field are transmitted with a higher bit rate than the beginning and the end of the frame.

The previously reserved bit in CAN frames with 11-bit identifiers and the first previously reserved bit in CAN frames with 29-bit identifiers will now be decoded as FDF bit.

If FDF is recessive, it signifies a CAN FD frame.

If FDF is dominant, it signifies a Classic CAN frame.

In a CAN FD frame, the two bits following FDF, res and BRS, decide whether the bit rate inside of this CAN FD frame is switched. A CAN FD bit rate switch is signified by res = dominant and BRS = recessive. The coding of res = recessive is reserved for future expansion of the protocol.

In case the CAN FD controller receives a frame with FDF = recessive and res = recessive, it will signal a Protocol Exception Event by setting bit PXE (CANFD_PSR[14]) bit. When Protocol Exception Handling is enabled (PXHD (CANFD_CCCR[12]) is reset), this causes the operation state to change from Receiver (ACT (CANFD_PSR[4:3]) = 2'b10) to Integrating (ACT (CANFD_PSR[4:3]) = 00) at the next sample point.

In case Protocol Exception Handling is disabled (PXHD (CANFD_CCCR[12]) is set), the CAN FD controller will treat a recessive res bit as a form error and will respond with an error frame.

CAN FD operation is enabled by programming FDOE (CANFD_CCCR[8]). In case FDOE (CANFD_CCCR[8]) = 1, transmission and reception of CAN FD frames is enabled. Transmission and reception of Classic CAN frames is always possible. Whether a CAN FD frame or a Classic CAN frame is transmitted can be configured via bit FDF in the respective Tx Buffer element. With FDOE (CANFD_CCCR[8]) = 0, received frames are interpreted as Classic CAN frames, which leads to the transmission of an error frame when receiving a CAN FD frame. When CAN FD operation is disabled, no CAN FD frames are transmitted even if bit FDF of a Tx Buffer element is set. FDOE (CANFD_CCCR[8]) and BRSE (CANFD_CCCR[9]) can only be changed while INIT (CANFD_CCCR[0]) and CCE (CANFD_CCCR[1]) are both set.

With FDOE (CANFD_CCCR[8]) = 0, the setting of bits FDF and BRS is ignored and frames are transmitted in Classic CAN format. If FDOE (CANFD_CCCR[8]) is set and BRSE (CANFD_CCCR[9]) is clear, only bit FDF of a Tx Buffer element is evaluated. If both FDOE (CANFD_CCCR[8]) and BRSE (CANFD_CCCR[9]) are set, transmission of CAN FD frames with bit rate switching is enabled. All Tx Buffer elements with bits FDF and BRS set are transmitted in CAN FD format with bit rate switching.

A mode change during CAN operation is only recommended under the following conditions:

- The failure rate in the CAN FD data phase is significant higher than in the CAN FD arbitration phase. In this case, disable the CAN FD bit rate switching option for transmissions.
- During system startup all nodes are transmitting Classic CAN messages until it is verified that they are able to communicate in CAN FD format. If this is true, all nodes switch to CAN FD operation.
- Wake-up messages in CAN Partial Networking have to be transmitted in Classic CAN format.
- End-of-line programming in case not all nodes are CAN FD capable. Non CAN FD nodes are held in Silent mode until programming has completed. Then all nodes switch back to

Classic CAN communication.

In the CAN FD format, the coding of the DLC differs from the standard CAN format. The DLC codes 0 to 8 have the same coding as in standard CAN, the codes 9 to 15, in standard CAN all code a data field of 8 bytes, which are coded according to Table 6.20-1 below.

DLC	9	10	11	12	13	14	15
Number of Data Bytes	12	16	20	24	32	48	64

Table 6.20-1 DLC Coding in CAN FD Format

In CAN FD frames, the bit timing will be switched inside the frame, after the BRS (Bit Rate Switch) bit, if this bit is recessive. Before the BRS bit, in the CAN FD arbitration phase, the nominal CAN bit timing is used as defined by the Nominal Bit Timing & Prescaler Register NBTP. In the following CAN FD data phase, the data phase bit timing is used as defined by the Data Bit Timing & Prescaler Register DBTP. The bit timing is switched back from the data phase timing at the CRC delimiter or when an error is detected, whichever occurs first.

The maximum configurable bit rate in the CAN FD data phase depends on the CAN Core clock frequency (cclk). Example: with a CAN FD controller clock frequency of 20 MHz and the shortest configurable bit time of 4 time quanta (tq), the bit rate in the data phase is 5 Mbit/s.

In both data frame formats, CAN FD and CAN FD with bit rate switching, the value of the bit ESI (Error Status Indicator) is determined by the transmitter's error state at the start of the transmission. If the transmitter is error passive, ESI is transmitted recessive, else it is transmitted dominant.

Transmitter Delay Compensation

During the data phase of a CAN FD transmission only one node is transmitting, all others are receivers. The length of the bus line has no impact. When transmitting via pin CANx_TXD the CAN FD Controller receives the transmitted data from its local CAN transceiver via pin CANx_RXD. The received data is delayed by the transceiver loop delay. In case this delay is greater than TSEG1 (time segment before sample point), a bit error is detected. In order to enable a data phase bit time that is even shorter than the transceiver loop delay, the delay compensation is introduced. Without transceiver delay compensation, the bit rate in the data phase of a CAN FD frame is limited by the transceiver loop delay.

The CAN FD Controller implements a delay compensation mechanism to compensate the CAN transceiver loop delay, thereby enabling transmission with higher bit rates during the CAN FD data phase independent of the delay of a specific CAN transceiver.

To check for bit errors during the data phase of transmitting nodes, the delayed transmit data is compared against the received data at the Secondary Sample Point (SSP). If a bit error is detected, the transmitter will react on this bit error at the next following regular sample point. During arbitration phase the delay compensation is always disabled.

The transmitter delay compensation enables configurations where the data bit time is shorter than the transmitter delay, it is described in detail in ISO 11898-1:2015. It is enabled by setting bit TDC (DBTP[23]).

The received bit is compared against the transmitted bit at the SSP. The SSP position is defined as the sum of the measured delay from the CAN FD controllers transmit output CANx_TXD pin through the transceiver to the receive input CANx_RXD pin plus the transmitter delay compensation offset as configured by TDCO (TDCR[[14:8]). The transmitter delay compensation offset is used to adjust the position of the SSP inside the received bit (e.g. half of the bit time in the data phase). The position of the secondary sample point is rounded down to the next integer number of mtq (minimum time quantum).

TDCV (PSR[22:16]) shows the actual transmitter delay compensation value. TDCV (PSR[22:16]) is cleared when INIT (CANFD_CCCR[0]) is set and is updated at each transmission of an FD frame while TDC (DBTP[23]) is set.

The following boundary conditions have to be considered for the transmitter delay compensation implemented in the CAN FD controller:

- The sum of the measured delay from CANx_TXD to CANx_RXD and the configured transmitter delay compensation offset TDCO (TDCR[[14:8]]) has to be less than 6 bit times in the data phase.
- The sum of the measured delay from CANx_TXD to CANx_RXD and the configured transmitter delay compensation offset TDCO (TDCR[[14:8]]) has to be less than or equal to 127 mtq (minimum time quantum). In case this sum exceeds 127 mtq, the maximum value of 127 mtq is used for transmitter delay compensation.
- The data phase ends at the sample point of the CRC delimiter, that stops checking of receive bits at the SSPs.

If transmitter delay compensation is enabled by programming TDC (DBTP[23]) is set, the measurement is started within each transmitted CAN FD frame at the falling edge of bit FDF to bit res. The measurement is stopped when this edge is seen at the receive input CANx_RXD pin of the transmitter. The resolution of this measurement is one mtq (minimum time quantum).

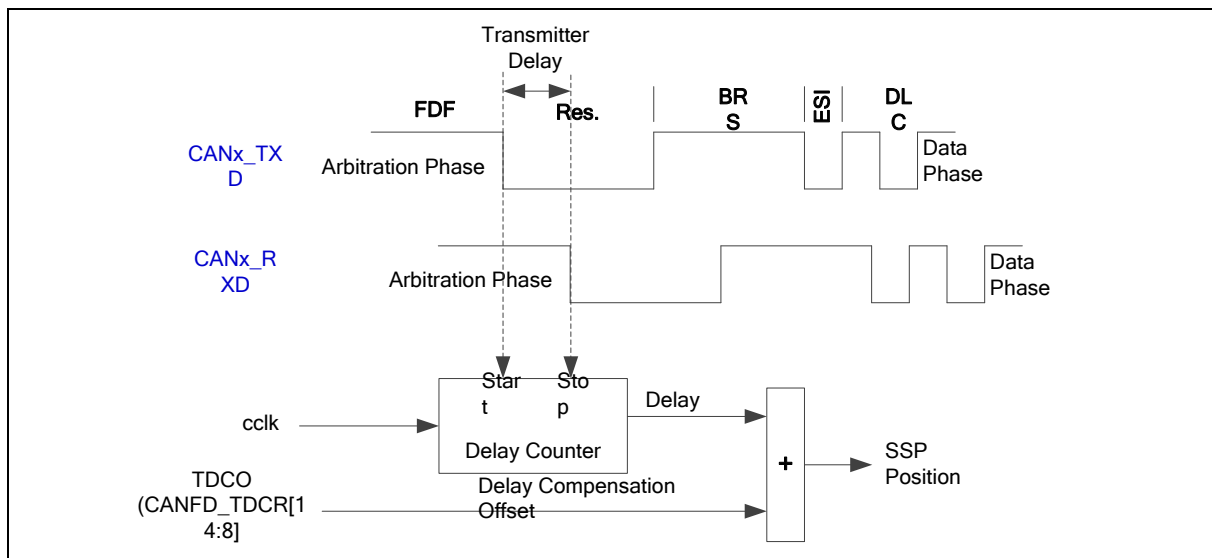


Figure 6.20-2 Transceiver Delay Measurement

To avoid that a dominant glitch inside the received FDF bit ends the delay compensation measurement before the falling edge of the received res bit, resulting in a too early SSP position, the use of a transmitter delay compensation filter window can be enabled by programming TDCF (CANFD_TDCR[6:0]). This defines a minimum value for the SSP position. Dominant edges on CANx_RXD, that would result in an earlier SSP position are ignored for transmitter delay measurement. The measurement is stopped when the SSP position is at least TDCF (CANFD_TDCR[6:0]) AND CANx_RXD is low.

Restricted Operation Mode

In Restricted Operation Mode the node is able to receive data and remote frames and to give acknowledge to valid frames, but it does not send data frames, remote frames, active error frames, or overload frames. In case of an error condition or overload condition, it does not send dominant bits, instead it waits for the occurrence of bus idle condition to resynchronize itself to the CAN communication. The error counters (REC (CANFD_ECR[14:8]), TEC (CANFD_ECR[7:0])) are frozen while Error Logging (CEL (CANFD_ECR[23:16])) is active. The software can set the CAN FD controller into Restricted Operation mode by setting bit ASM (CANFD_CCCR[2]). The bit can only be set by the software when

both CCE (CANFD_CCCR[1]) and INIT (CANFD_CCCR[0]) are set to 1. The bit can be reset by the software at any time.

Restricted Operation Mode is automatically entered when the Tx Handler was not able to read data from the Message RAM in time. To leave Restricted Operation Mode, the software has to reset ASM (CANFD_CCCR[2]).

The Restricted Operation Mode can be used in applications that adapt themselves to different CAN bit rates. In this case the application tests different bit rates and leaves the Restricted Operation Mode after it has received a valid frame.

Note: The Restricted Operation Mode must not be combined with the Loop Back Mode (internal or external).

Bus Monitor Mode

The CAN FD controller is set in Bus Monitoring Mode by programming MON (CANFD_CCCR[6]) to one. In Bus Monitoring Mode (refer to ISO 11898-1:2015, 10.14 Bus monitoring), the CAN FD controller is able to receive valid data frames and valid remote frames, but cannot start a transmission. In this mode, it sends only recessive bits on the CAN bus. If the controller is required to send a dominant bit (ACK bit, overload flag, active error flag), the bit is rerouted internally so that the controller monitors this dominant bit, although the CAN bus may remain in recessive state. In Bus Monitoring Mode register CANFD_TXBRP is held in reset state.

The Bus Monitoring Mode can be used to analyze the traffic on a CAN bus without affecting it by the transmission of dominant bits. Figure 6.20-3 shows the connection of signals CANx_TXD pin and CANx_RXD pin to the CAN FD controller in Bus Monitoring Mode.

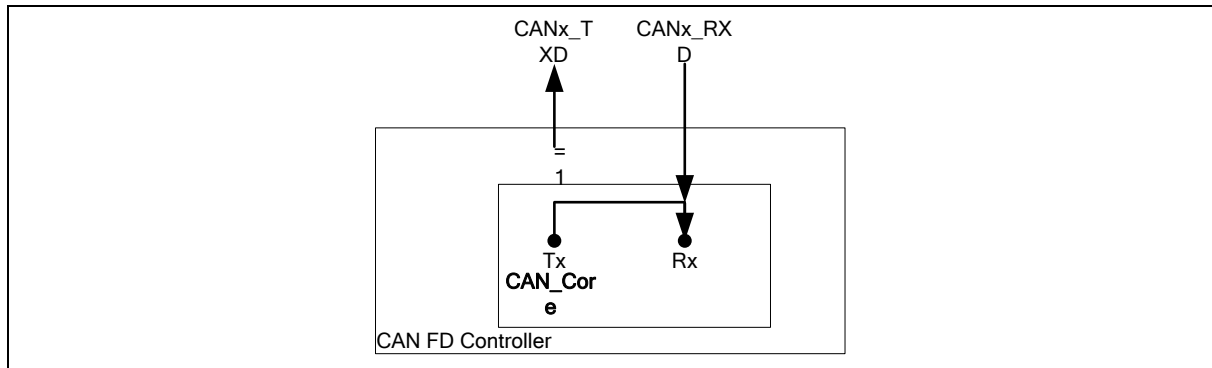


Figure 6.20-3 Pin Controller in Bus Monitor Mode

Disable Automatic Retransmission (DAR) Mode

According to the CAN Specification (refer to ISO 11898-1:2015, 8.3.4 Recovery Management), the CAN FD controller provides means for automatic retransmission of frames that have lost arbitration or that have been disturbed by errors during transmission. By default, automatic retransmission is enabled. To support time-triggered communication as described in ISO 11898-1:2015, chapter 9.2, the automatic retransmission may be disabled via DAR (CANFD_CCCR[6]).

Frame Transmission in DAR Mode

In DAR mode all transmissions are automatically cancelled after they started on the CAN bus. A Tx Buffer's Tx Request Pending bit TRPx (CANFD_TXBRP[31:0]) is reset after successful transmission, when a transmission has not yet been started at the point of cancellation, has been aborted due to lost arbitration, or when an error occurred during frame transmission.

- Successful transmission:
 - Corresponding Tx Buffer Transmission Occurred bit TOx (CANFD_TXBTO[31:0]) set
 - Corresponding Tx Buffer Cancellation Finished bit CFx (CANFD_TXBCF[31:0]) not set
- Successful transmission in spite of cancellation:

- Corresponding Tx Buffer Transmission Occurred bit TOx (CANFD_TXBTO[31:0]) set
- Corresponding Tx Buffer Cancellation Finished bit CFx (CANFD_TXBCF[31:0]) set
- Arbitration lost or frame transmission disturbed:
 - Corresponding Tx Buffer Transmission Occurred bit TOx (CANFD_TXBTO[31:0]) not set
 - Corresponding Tx Buffer Cancellation Finished bit CFx (CANFD_TXBCF[31:0]) set

In case of a successful frame transmission, and if storage of Tx events is enabled, a Tx Event FIFO element is written with Event Type ET = 10 (transmission in spite of cancellation).

Power Mode

The CAN FD controller can be set into Power-down mode controlled by input signal power_down_req or via CSR (CANFD_CCCR[4]). As long as the clock stop request signal power_down_req is active, bit CSR (CANFD_CCCR[4]) is read as one.

When all pending transmission requests have completed, the CAN FD controller waits until bus idle state is detected. Then the CAN FD controller sets then INIT (CANFD_CCCR[0]) to one to prevent any further CAN transfers. Now the CAN FD controller acknowledges that it is ready for power down by setting output signal power_down_ack to one and CCCR.CSA to one. Before the clocks are switched off, further register accesses can be made except for INIT (CANFD_CCCR[0]) which is held at one.

Note: In case of a heavily disturbed CAN bus, it may happen that idle state is never reached and INIT (CANFD_CCCR[0]) is therefore not set by the controller. This situation can be detected by polling ACT (CANFD_PSR[4:3]). In case the controller does not enter idle state, the software can write INIT (CANFD_CCCR[0]) to 1 which stops CAN communication of the CAN FD controller immediately, regardless whether there is a transmission/reception ongoing or not.

To leave Power-down mode, the application has to turn on the module clocks before resetting control register flag CSR (CANFD_CCCR[4]). The controller will acknowledge this by resetting CAA (CANFD_CCCR[3]). Afterwards, the application can restart CAN communication by resetting bit INIT (CANFD_CCCR[0]).

Test Mode

To enable write access to register TEST, bit TEST (CANFD_CCCR[7]) has to be set to 1. This allows the configuration of the test modes and test functions.

Four output functions are available for the CAN transmit pin CANx_TXD by programming TX (CANFD_TEST[6:5]). Additionally to its default function – the serial data output – it can drive the CAN Sample Point signal to monitor the CAN FD controller bit timing and it can drive constant dominant or recessive values. The actual value at pin CANx_RXD can be read from RX (CANFD_TEST[7]). Both functions can be used to check the CAN bus physical layer.

Due to the synchronization mechanism between CAN Core clock and AHB clock domain, there may be a delay of several AHB clock periods between writing to TX (CANFD_TEST[6:5]) until the new configuration is visible at output pin CANx_TXD. This applies also when reading input pin CANx_RXD via RX (CANFD_TEST[7]).

Note: Test modes should be used for production tests or self-test only. The software control for pin CANx_TXD interferes with all CAN protocol functions. It is not recommended to use test modes for application.

- External Loop Back Mode

The CAN FD controller can be set in External Loop Back Mode by programming LBCK (CANFD_TEST[4]) to one. In Loop Back Mode, the controller treats its own transmitted messages as received messages and stores them (if they pass acceptance filtering) into an Rx Buffer or an Rx FIFO. Figure 6.20-4 shows the connection of signals CANx_TXD pin and CANx_RXD pin to the CAN FD controller in External Loop Back Mode.

This mode is provided for hardware self-test. To be independent from external stimulation, the controller ignores acknowledge errors (recessive bit sampled in the acknowledge slot of a data/remote frame) in Loop Back Mode. In this mode the controller performs an internal feedback from its Tx output to its Rx input. The actual value of the CANx_RXD input pin is disregarded by the controller. The transmitted messages can be monitored at the CANx_TXD pin.

- Internal Loop Back Mode

Internal Loop Back Mode is entered by programming bits LBCK (CANFD_TEST[4]) and MON (CANFD_CCCR[6]) to one. This mode can be used for a “Hot Selftest”, meaning the controller can be tested without affecting a running CAN system connected to the pins CANx_TXD and CANx_RX. In this mode pin CANx_RXD is disconnected from the controller and pin CANx_TXD is held recessive. Figure 6.20-4 shows the connection of CANx_TXD and CANx_RXD to the controller in case of Internal Loop Back Mode.

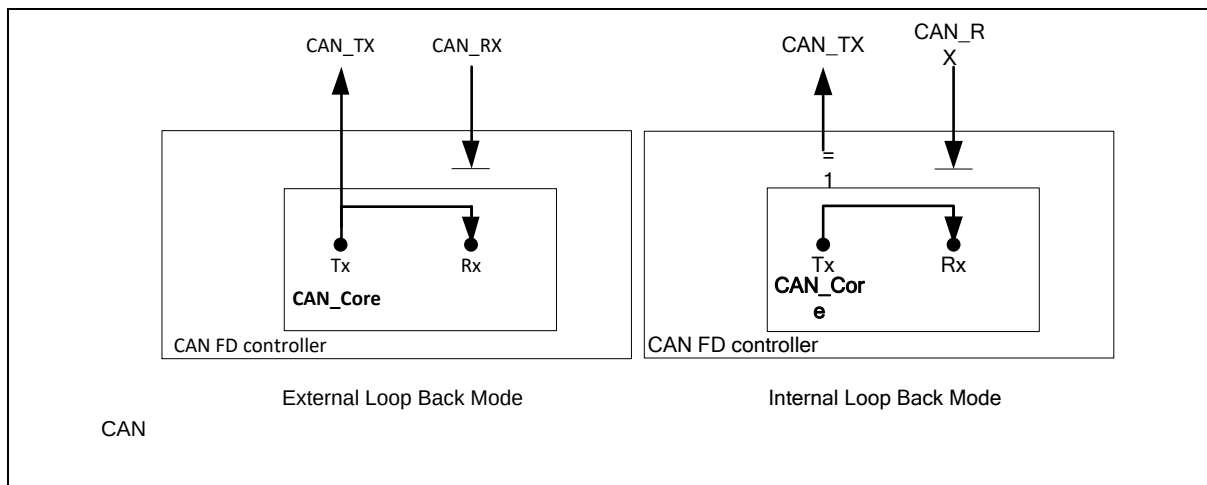


Figure 6.20-4 Pin Control in Loop Back Modes

6.20.5.2 Timestamp Generation

For timestamp generation the CAN FD controller supplies a 16-bit wrap-around counter. A pre-scaler TCP (CANFD_TSCC[19:16]) can be configured to clock the counter in multiples of CAN bit times (1...16). The counter is readable via TSC (CANFD_TSCV[15:0]). A write access to register CANFD_TSCV resets the counter to 0. When the timestamp counter wraps around interrupt flag TSW (CANFD_IR[16]) is set.

On start of frame reception / transmission (SOF) the counter value is captured and stored into the **Timestamp Generation** section of an Rx Buffer / Rx FIFO (CANFD_RXTS[15:0]) or Tx Event FIFO (CANFD_TXTS[15:0]) element.

By programming bit TSS (CANFD_TSCC[[1:0]]), a 16-bit timestamp can be used.

6.20.5.3 Timeout Counter

To signal timeout conditions for Rx FIFO 0, Rx FIFO 1, and the Tx Event FIFO the CANFD controller supplies a 16-bit timeout counter. It operates as down-counter and uses the same pre-scaler controlled by TCP (CANFD_TSCC[19:16]) as the timestamp counter. The timeout counter is configured via register TOCC. The actual counter value can be read from TOC (CANFD_TOCV[15:0]). The timeout counter can only be started while INIT (CANFD_CCCR[0]) = 0. It is stopped when INIT (CANFD_CCCR[0]) = 1, e.g. when the CAN FD Controller enters Bus_Off state.

The operation mode is selected by TOS (CANFD_TOCC[1:0]). When operating in Continuous Mode, the counter starts when INIT (CANFD_CCCR[0]) is reset. A write to TOCV presets the counter to the

value configured by TOP (CANFD_TOCC[31:16]) and continues down-counting.

When the timeout counter is controlled by one of the FIFOs, an empty FIFO presets the counter to the value configured by TOP (CANFD_TOCC[31:16]). Down-counting is started when the first FIFO element is stored. Writing to TOCV has no effect.

When the counter reaches 0, interrupt flag TOO (CANFD_IR[18]) is set. In Continuous Mode, the counter is immediately restarted at TOP (CANFD_TOCC[31:16]).

Note: The clock signal for the timeout counter is derived from the CAN Core's sample point signal. Therefore the point in time where the timeout counter is decremented may vary due to the synchronization / re-synchronization mechanism of the CAN Core. If the bit rate switch feature in CAN FD controller is used, the timeout counter is clocked differently in arbitration and data field.

6.20.5.4 Rx Handling

The Rx Handler controls the acceptance filtering, the transfer of received messages to the Rx Buffers or to one of the two Rx FIFOs, as well as the Rx FIFO's put and get Indices.

Acceptance Filtering

The CAN FD controller offers the possibility to configure two sets of acceptance filters, one for standard identifiers and one for extended identifiers. These filters can be assigned to an Rx Buffer or to Rx FIFO 0,1. For acceptance filtering each list of filters is executed from element #0 until the first matching element. Acceptance filtering stops at the first matching element. The following filter elements are not evaluated for this message.

The main features are:

- Each filter element can be configured as
 - range filter (from 0 to 128 elements for the 11-bit filter and from 0 to 64 for the 29-bit filter)
 - filter for one or two dedicated IDs
 - classic bit mask filter
- Each filter element is configurable for acceptance or rejection filtering
- Each filter element can be enabled / disabled individually
- Filters are checked sequentially, execution stops with the first matching filter element

Related configuration registers are:

- Global Filter Configuration CANFD_GFC
- Standard ID Filter Configuration CANFD_SIDFC
- Extended ID Filter Configuration CANFD_XIDF
- Extended ID AND Mask CANFD_XIDAM

Depending on the configuration of the filter element (SFEC/EFEC) a match triggers one of the following actions:

- Store received frame in FIFO 0 or FIFO 1
- Store received frame in Rx Buffer
- Store received frame in Rx Buffer and generate pulse at filter event pin
- Reject received frame
- Set High Priority Message interrupt flag HPM (CANFD_IR[8])
- Set High Priority Message interrupt flag HPM (CANFD_IR[8]) and store received frame in FIFO 0 or FIFO 1

Acceptance filtering is started after the complete identifier has been received. After acceptance filtering

has completed, and if a matching Rx Buffer or Rx FIFO has been found, the Message Handler starts writing the received message data in portions of 32 bit to the matching Rx Buffer or Rx FIFO. If the CAN protocol controller has detected an error condition (e.g. CRC error), this message is discarded with the following impact on the affected Rx Buffer or Rx FIFO:

Rx Buffer

New Data flag of matching Rx Buffer is not set, but Rx Buffer (partly) overwritten with received data. For error type refer to LEC (CANFD_PSR[2:0]) respectively DLEC (CANFD_PSR[10:8]).

Rx FIFO

Put index of matching Rx FIFO is not updated, but related Rx FIFO element (partly) overwritten with received data. For error type refer to LEC (CANFD_PSR[2:0]) respectively DLEC (CANFD_PSR[10:8]). In case the matching Rx FIFO is operated in overwrite mode, the boundary conditions described in **Rx FIFO Overwrite Mode** section have to be considered.

Note: When an accepted message is written to one of the two Rx FIFOs, or into an Rx Buffer, the unmodified received identifier is stored independent of the filter(s) used. The result of the acceptance filter process is strongly depending on the sequence of configured filter elements.

- Range Filter

The filter matches for all received frames with Message IDs in the range defined by SF1ID/SF2ID resp. EF1ID/EF2ID.

There are two possibilities when range filtering is used together with extended frames:

EFT = 2'b00: The Message ID of received frames is ANDed with the Extended ID AND Mask (CANFD_XIDAM) before the range filter is applied.

EFT = 2'b11: The Extended ID AND Mask (CANFD_XIDAM) is not used for range filtering.

- Filter for specific IDs

A filter element can be configured to filter for one or two specific Message IDs. To filter for one specific Message ID, the filter element has to be configured with SF1ID = SF2ID resp. EF1ID = EF2ID.

- Classic Bit Mask Filter

Classic bit mask filtering is intended to filter groups of Message IDs by masking single bits of a received Message ID. With classic bit mask filtering SF1ID/EF1ID is used as Message ID filter, while SF2ID/EF2ID is used as filter mask.

A zero bit at the filter mask will mask out the corresponding bit position of the configured ID filter, e.g. the value of the received Message ID at that bit position is not relevant for acceptance filtering.

Only those bits of the received Message ID where the corresponding mask bits are one are relevant for acceptance filtering.

In case all mask bits are one, a match occurs only when the received Message ID and the Message

ID filter are identical. If all mask bits are 0, all Message IDs match.

- Standard Message ID Filtering

Figure 6.20-6 below shows the flow for standard Message ID (11-bit Identifier) filtering. The Standard Message ID Filter element is described in **Message RAM** section.

Controlled by the global filter configuration GFC and the Standard ID filter configuration (CANFD_SIDFC) Message ID, Remote Transmission Request bit (RTR), and the Identifier Extension bit (IDE) of received frames are compared against the list of configured filter elements.

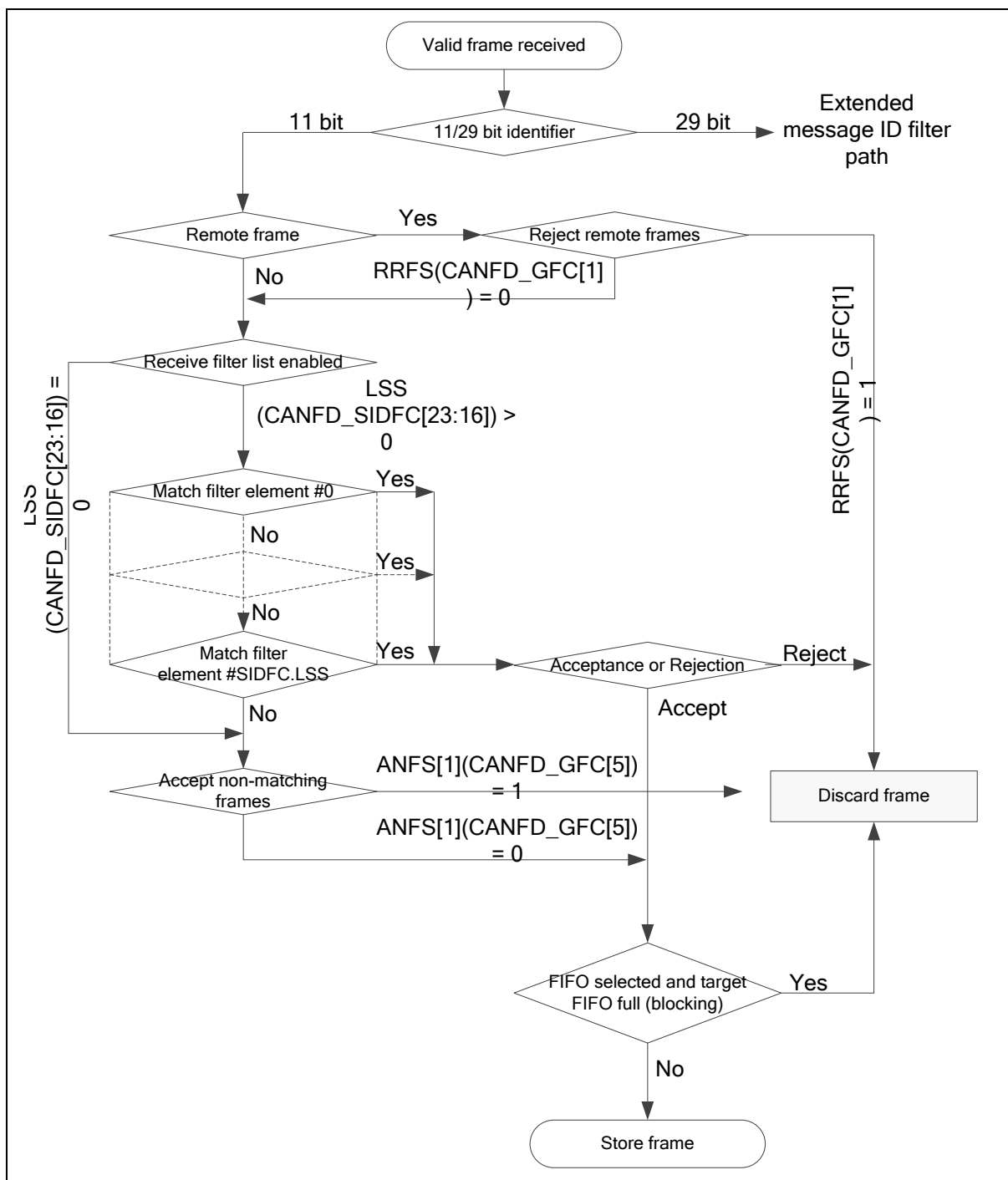


Figure 6.20-5 Standard Message ID Filter Path

- Extended Message ID Filtering

Figure 6.20-6 shows the flow for extended Message ID (29-bit Identifier) filtering. The Extended Message ID Filter element is described in **Message RAM** Section.

Controlled by the global filter configuration (CANFD_GFC) and the Extended ID filter configuration (CANFD_XIDFC) Message ID, Remote Transmission Request bit (RTR), and the Identifier Extension bit (IDE) of received frames are compared against the list of configured filter elements.

The Extended ID AND Mask (CANFD_XIDAM) is AND-ed with the received identifier before the filter list is executed.

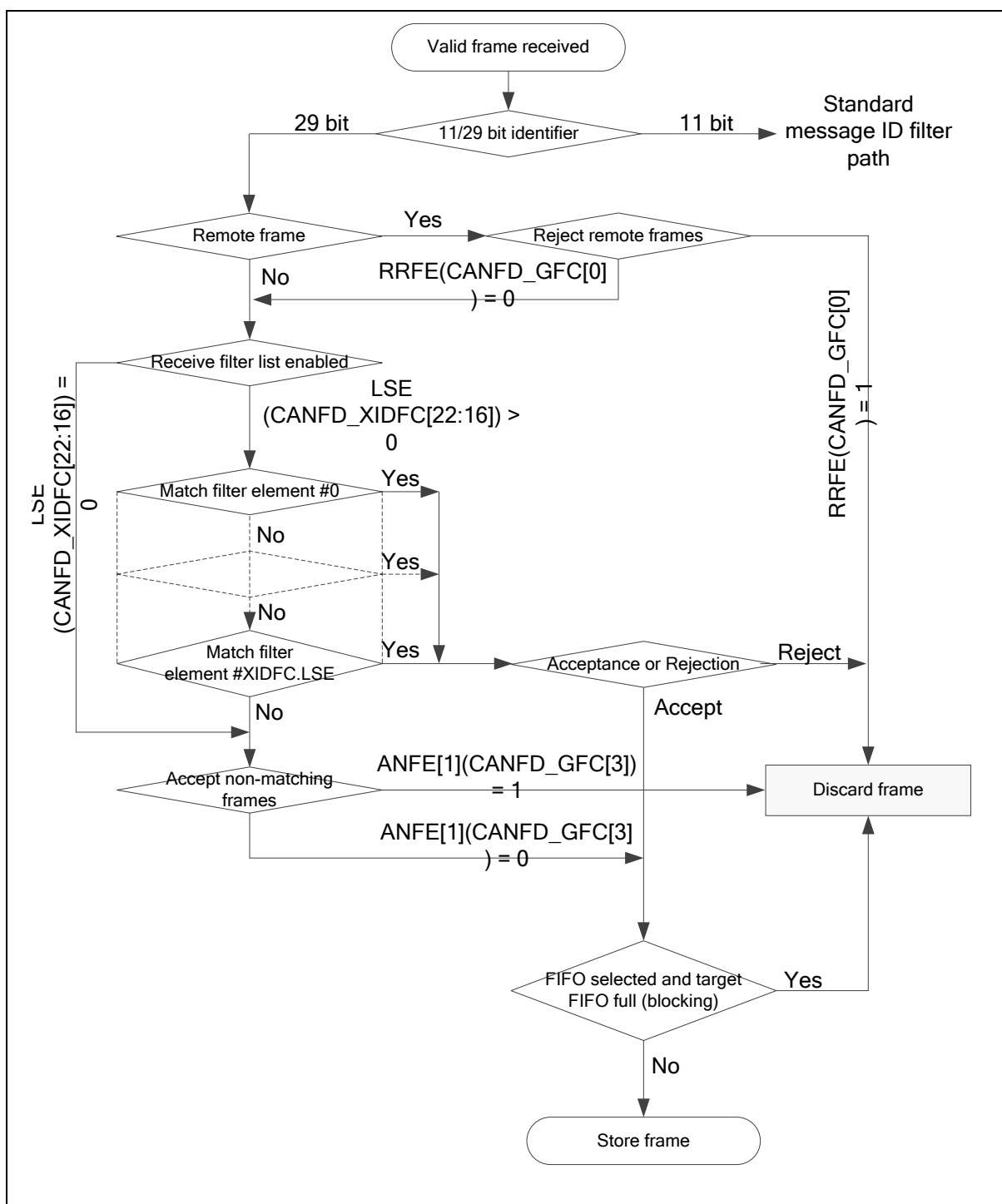


Figure 6.20-6 Extended Message ID Filter Path

Rx FIFOs

Rx FIFO 0 and Rx FIFO 1 can be configured to hold up to 64 elements each. Configuration of the two Rx FIFOs is done via registers CANFD_RXF0C and CANFD_RXF1C.

Received messages that passed acceptance filtering are transferred to the Rx FIFO as configured by

the matching filter element. For a description of the filter mechanisms available for Rx FIFO 0 and Rx FIFO 1 refer to **Acceptance Filtering** section. The Rx FIFO element is described in Rx Buffer and **FIFO Element** section.

To avoid an Rx FIFO overflow, the Rx FIFO watermark can be used. When the Rx FIFO fill level reaches the Rx FIFO watermark configured by FnWM (CANFD_RxFnC[30:24]), interrupt flag RFnW (CANFD_IR) is set. When the Rx FIFO put Index reaches the Rx FIFO get Index an Rx FIFO Full condition is signalled by FnF (CANFD_RXFnS). In addition interrupt flag RFnF (CANFD_IR) is set.

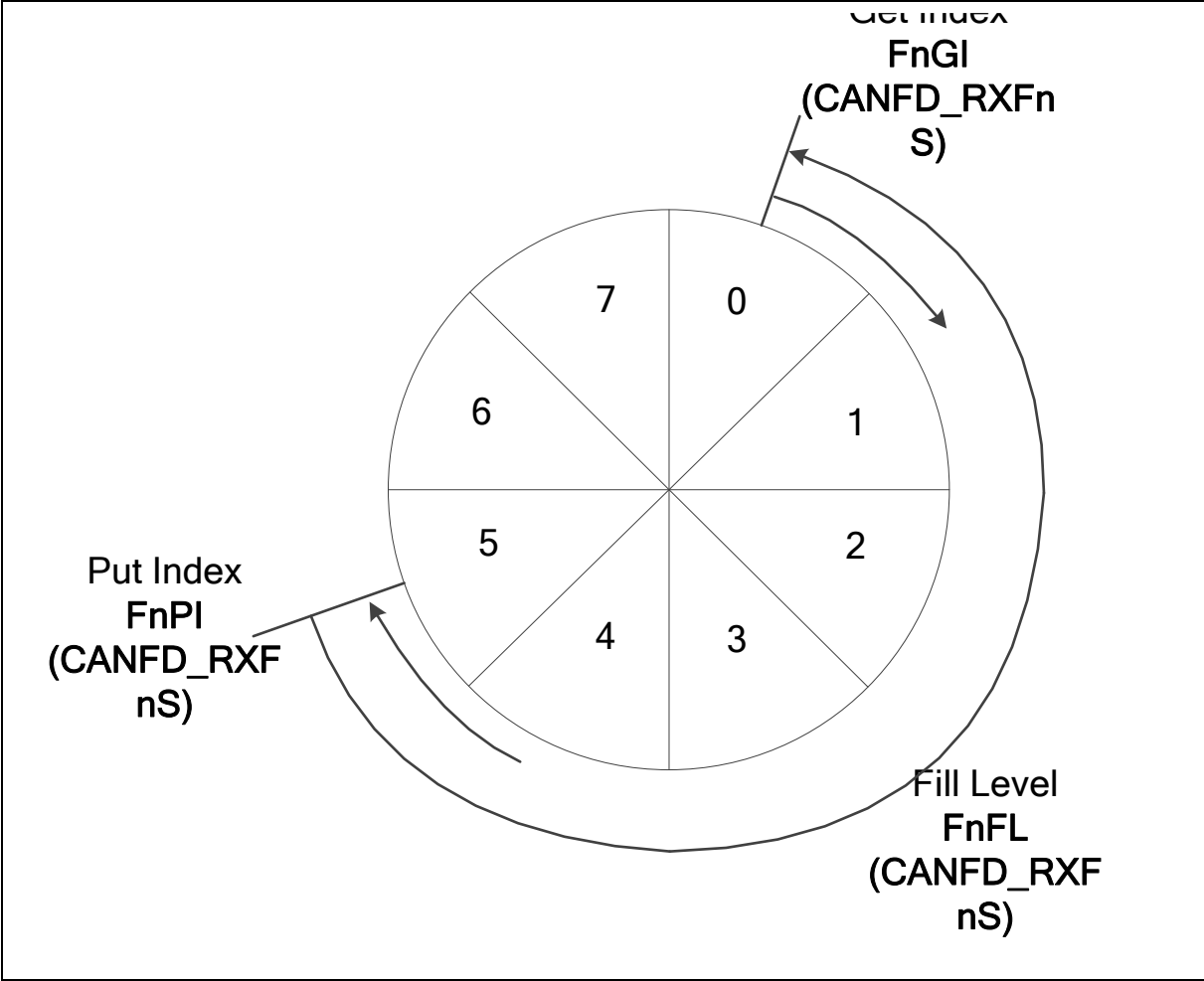


Figure 6.20-7 Rx FIFO Status

When reading from an Rx FIFO, Rx FIFO Get Index FnGI (CANFD_RXFnS) + FIFO element size has to be added to the corresponding Rx FIFO start address FnSA (CANFD_RXFnC).

RBDS(CANFD_RXESC[10:8]) FnDS(CANFD_RXESC[2:0])	Data Field [Bytes]	FIFO Element Size [RAM Words]
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8

101	32	10
110	48	14
111	64	18

Table 6.20-2 Rx Buffer / FIFO Element Size

- Rx FIFO Blocking Mode

The Rx FIFO blocking mode is configured by FnOM (CANFD_RXFnC) = 0. This is the default operation mode for the Rx FIFOs.

When an Rx FIFO full condition is reached (FnPI (CANFD_RXFnS) = FnGI (CANFD_RXFnS)), no further messages are written to the corresponding Rx FIFO until at least one message has been read out and the Rx FIFO Get Index has been incremented. An Rx FIFO full condition is signalled by FnF (CANFD_RXFnS) = 1. In addition interrupt flag RFnF (CANFD_IR) is set.

In case a message is received while the corresponding Rx FIFO is full, this message is discarded and the message lost condition is signalled by RFnL (CANFD_RXFnS) = 1. In addition interrupt flag RFnL (CANFD_IR) is set.

- Rx FIFO Overwrite Mode

The Rx FIFO overwrite mode is configured by FnOM (CANFD_RXFnC) = 1.

When an Rx FIFO full condition (FnPI (CANFD_RXFnS) = FnGI (CANFD_RXFnS)) is signalled by FnF (CANFD_RXFnS) = 1, the next message accepted for the FIFO will overwrite the oldest FIFO message. Put and get index are both incremented by one.

When an Rx FIFO is operated in overwrite mode and an Rx FIFO full condition is signalled, reading of the Rx FIFO elements should start at least at get index + 1. The reason for that is, that it might happen, that a received message is written to the Message RAM (put index) while the CPU is reading from the Message RAM (get index). In this case inconsistent data may be read from the respective Rx FIFO element. Adding an offset to the get index when reading from the Rx FIFO avoids this problem. The offset depends on how fast the CPU accesses the Rx FIFO. Figure 6.20-8 shows an offset of two with respect to the get index when reading the Rx FIFO. In this case the two messages stored in element 1 and 2 are lost.

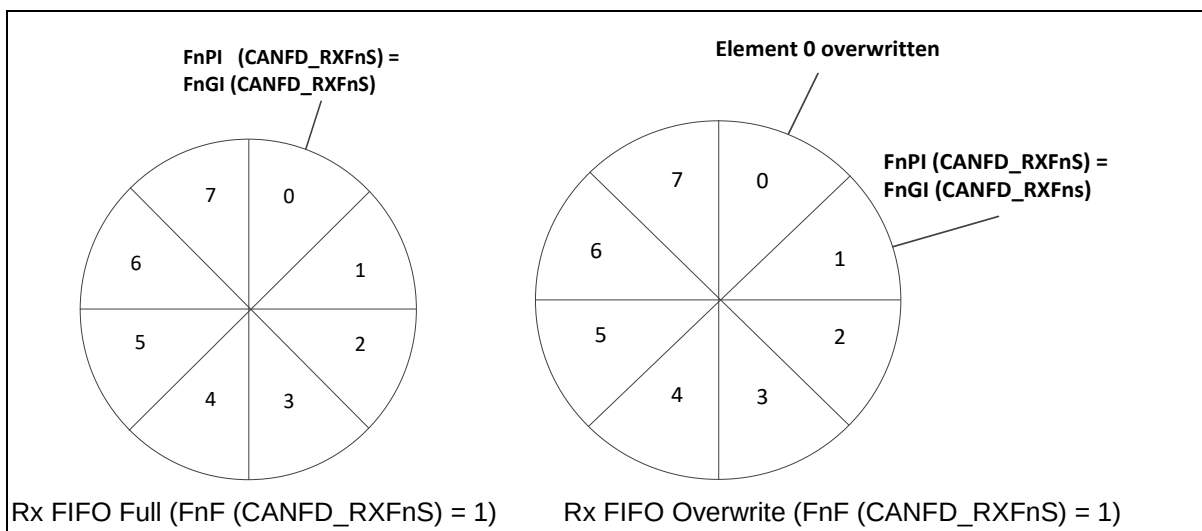


Figure 6.20-8 Rx FIFO Overflow Handling

After reading from the Rx FIFO, the number of the last element read has to be written to the Rx FIFO Acknowledge Index FnA (CANFD_RXFnA). This increments the get index to that element number. In case the put index has not been incremented to this Rx FIFO element, the Rx FIFO full condition is reset (FnF (CANFD_RXFnS) = '0').

Dedicated Rx Buffers

The CAN FD controller supports up to 64 dedicated Rx Buffers. The start address of the dedicated Rx Buffer section is configured via RBSA (CANFD_RXBC[15:2]).

For each Rx Buffer a Standard or Extended Message ID Filter Element with SFEC / EFEC = 111 and SFID2 / EFID2[10:9] = 00 has to be configured (refer to **Standard Message ID Filter Element** and **Extended Message ID Filter Element**).

After a received message has been accepted by a filter element, the message is stored into the Rx Buffer in the Message RAM referenced by the filter element. The format is the same as for an Rx FIFO element. In addition the flag DRX(CANFD_IR[19]) (Message stored in Dedicated Rx Buffer) in the interrupt register is set.

Filter Element	SFID1[10:0] EFID1[28:0]	SFID2[10:9] EFID2[10:9]	SFID2[5:0] EFID2[5:0]
0	ID message 1	00	00 0000
1	ID message 2	00	00 0001
2	ID message 3	00	00 0010

Table 6.20-3 Example Filter Configuration for Rx Buffers

After the last word of a matching received message has been written to the Message RAM, the respective New Data flag in register NDAT1,2 is set. As long as the New Data flag is set, the respective Rx Buffer is locked against updates from received matching frames. The New Data flags have to be reset by the software by writing a '1' to the respective bit position.

While an Rx buffer new data flag is set, a message ID filter element referencing this specific Rx Buffer will not match, causing the acceptance filtering to continue. The following message ID filter elements may cause the received message to be stored into another Rx Buffer, or into an Rx FIFO, or the message may be rejected, depending on filter configuration.

- Rx Buffer Handling
 - Reset interrupt flag DRX(CANFD_IR[19])
 - Read New Data registers
 - Read messages from Message RAM
 - Reset New Data flags of processed messages

6.20.5.5 Tx Handling

The Tx Handler handles transmission requests for the dedicated Tx Buffers, the Tx FIFO, and the Tx Queue. It controls the transfer of transmit messages to the CAN Core, the Put and Get Indices, and the Tx Event FIFO. Up to 32 Tx Buffers can be set up for message transmission. The CAN mode for transmission (Classic CAN or CAN FD) can be configured separately for each Tx Buffer element. The Tx Buffer element is described in **Tx Buffer Element** section. Table 6.20-4 describes the possible configurations for frame transmission.

CANFD_CCCR	Tx Buffer Element	Frame Transmission

BRSE	FDOE	FDF	BRS	
ignored	0	ignored	ignored	Classic CAN
0	1	0	ignored	Classic CAN
0	1	1	ignored	FD without bit rate switching
1	1	0	ignored	Classic CAN
1	1	1	0	FD without bit rate switching
1	1	1	1	FD with bit rate switching

Table 6.20-4 Possible Configuration for Frame Transmission

Note: AUTOSAR requires at least three Tx Queue Buffers and support of transmit cancellation

The Tx Handler starts a Tx scan of the Message RAM to check for the highest priority pending Tx request (Tx buffer with lowest message ID) when the Tx buffer request pending register CANFD_TXBRP is updated, or when a transmission has been started. CANFD_TXBRP is updated when a transmission finishes, or after the software has requested a transmission by writing to CANFD_TXBAR, or when the Host has cancelled a pending transmission by writing to CANFD_TXBCR.

Transmit Pause

The transmit pause feature is intended for use in CAN systems where the CAN message identifiers are (permanently) specified to specific values and cannot easily be changed. These message identifiers may have a higher CAN arbitration priority than other defined messages, while in a specific application their relative arbitration priority should be inverse. This may lead to a case where one electronic control unit (ECU) sends a burst of CAN messages that cause another ECU CAN messages to be delayed because that other messages have a lower CAN arbitration priority.

If e.g. CAN ECU-1 has the transmit pause feature enabled and is requested by its application software to transmit four messages, it will, after the first successful message transmission, wait for two CAN bit times of bus idle before it is allowed to start the next requested message. If there are other ECUs with pending messages, those messages are started in the idle time, they would not need to arbitrate with the next message of ECU-1. After having received a message, ECU-1 is allowed to start its next transmission as soon as the received message releases the CAN bus.

The transmit pause feature is controlled by bit TXP (CANFD_CCCR[14]). If the bit is set, the controller will, each time it has successfully transmitted a message, pause for two CAN bit times before starting the next transmission. This enables other CAN nodes in the network to transmit messages even if their messages have lower prior identifiers. Default is transmit pause disabled (TXP (CANFD_CCCR[14]) = 0).

This feature looses up burst transmissions coming from a single node and it protects against "babbling idiot" scenarios where the application program erroneously requests too many transmissions.

Dedicated Tx Buffers

Dedicated Tx Buffers are intended for message transmission under complete control of the CPU. Each Dedicated Tx Buffer is configured with a specific Message ID. In case that multiple Tx Buffers are configured with the same Message ID, the Tx Buffer with the lowest buffer number is transmitted first.

If the data section has been updated, a transmission is requested by an Add Request via ARn (CANFD_TXBAR[31:0]). The requested messages arbitrate internally with messages from an optional Tx FIFO or Tx queue and externally with messages on the CAN bus, and are sent out according to their Message ID.

A dedicated Tx buffer allocates four 32-bit words in the Message RAM (refer to Table 6.20-5). Therefore the start address of a dedicated Tx Buffer in the Message RAM is calculated by adding transmit buffer index (0...31) to the Tx buffer start address in TBSA (CANFD_TXBC[15:2]).

TBDS (CANFD_TXESC[2:0])	Data Field [Bytes]	Element Size [RAM Words]
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8
101	32	10
110	48	14
111	64	18

Table 6.20-5 Tx Buffer / FIFO / Queue Element Size

Tx FIFO

Tx FIFO operation is configured by programming TFQM (CANFD_TXBC[30]) to 0. Messages stored in the Tx FIFO are transmitted starting with the message referenced by the get index TFGI (CANFD_TXFQS[12:8]). After each transmission the get index is incremented cyclically until the Tx FIFO is empty. The Tx FIFO enables transmission of messages with the same Message ID from different Tx Buffers in the order these messages have been written to the Tx FIFO. The controller calculates the Tx FIFO Free Level TFGI (CANFD_TXFQS[5:0]) as difference between get and put index. It indicates the number of available (free) Tx FIFO elements.

New transmit messages have to be written to the Tx FIFO starting with the Tx Buffer referenced by the put index TFGI (CANFD_TXFQS[20:16]). An add request increments the Put Index to the next free Tx FIFO element. When the put index reaches the get index, Tx FIFO Full (TFGI (CANFD_TXFQS[21]) = 1) is signaled. In this case no further messages should be written to the Tx FIFO until the next message has been transmitted and the Get Index has been incremented.

When a single message is added to the Tx FIFO, the transmission is requested by writing a 1 to the TXBAR bit related to the Tx Buffer referenced by the Tx FIFO put index.

When multiple (n) messages are added to the Tx FIFO, they are written to n consecutive Tx Buffers starting with the Put Index. The transmissions are then requested via CANFD_TXBAR. The put index is then cyclically incremented by n. The number of requested Tx buffers should not exceed the number of free Tx Buffers as indicated by the Tx FIFO Free Level.

When a transmission request for the Tx Buffer referenced by the get Index is cancelled, the get index is incremented to the next Tx Buffer with pending transmission request and the Tx FIFO free level is recalculated. When transmission cancellation is applied to any other Tx Buffer, the get index and the FIFO free level remain unchanged.

A Tx FIFO element allocates element size 32-bit words in the Message RAM (refer to Table 6.20-5). Therefore the start address of the next available (free) Tx FIFO Buffer is calculated by adding the put index TFQPI (CANFD_TXFQS[20:16]) * element size to the Tx buffer start address TBSA (CANFD_TXBC[15:2]).

Tx Queue

Tx Queue operation is configured by programming TFQM (CANFD_TXBC[30]) to 1. Messages stored in the Tx Queue are transmitted starting with the message with the lowest Message ID (highest priority). In case that multiple Queue Buffers are configured with the same Message ID, the Queue Buffer with the lowest buffer number is transmitted first.

New messages have to be written to the Tx Buffer referenced by the Put Index TFGI (CANFD_TXFQS[20:16]). An Add Request cyclically increments the Put Index to the next free Tx Buffer. In case that the Tx Queue is full (TFGI (CANFD_TXFQS[21]) = 1), the Put Index is not valid and no

further message should be written to the Tx Queue until at least one of the requested messages has been sent out or a pending transmission request has been cancelled.

The application may use register CANFD_TXBRP instead of the put index and may place messages to any Tx Buffer without pending transmission request.

A Tx Queue Buffer allocates element size 32-bit words in the Message RAM (refer to Table 6.20-5). Therefore the start address of the next available (free) Tx Queue Buffer is calculated by adding put index TFQPI (CANFD_TXFQS[20:16]) put index TFGI (CANFD_TXFQS[20:16]) * element size to the Tx buffer start address TBSA (CANFD_TXBC[15:2]).

Mixed Dedicated Tx Buffers / Tx FIFO

In this case the Tx Buffers section in the Message RAM is subdivided into a set of dedicated Tx Buffers and a Tx FIFO. The number of dedicated Tx Buffers is configured by NDTB (CANFD_TXBC[21:16]). The number of Tx Buffers assigned to the Tx FIFO is configured by TFQS (CANFD_TXBC[29:24]). In case TFQS (CANFD_TXBC[29:24]) is programmed to 0, only Dedicated Tx Buffers are used.

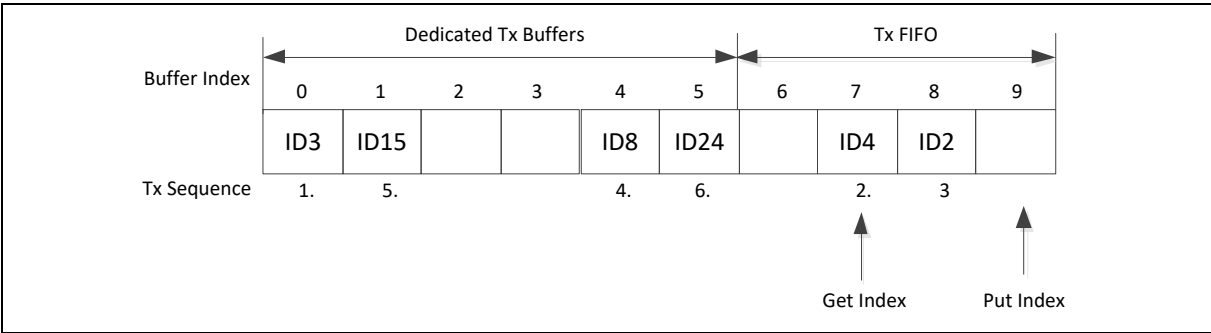


Figure 6.20-9 Example of Mixed Configuration Dedicated Tx Buffers / Tx FIFO

Tx prioritization:

- Scan Dedicated Tx Buffers and oldest pending Tx FIFO Buffer (referenced by TFGI (CANFD_TXFQS[12:8]))
- Buffer with lowest Message ID gets highest priority and is transmitted next.

Mixed Dedicated Tx Buffers / Tx Queue

In this case the Tx Buffers section in the Message RAM is subdivided into a set of dedicated Tx Buffers and a Tx Queue. The number of dedicated Tx Buffers is configured by NDTB (CANFD_TXBC[21:16]). The number of Tx Queue Buffers is configured by TFQS (CANFD_TXBC[29:24]). In case TFQS (CANFD_TXBC[29:24]) is programmed to 0, only dedicated Tx Buffers are used.

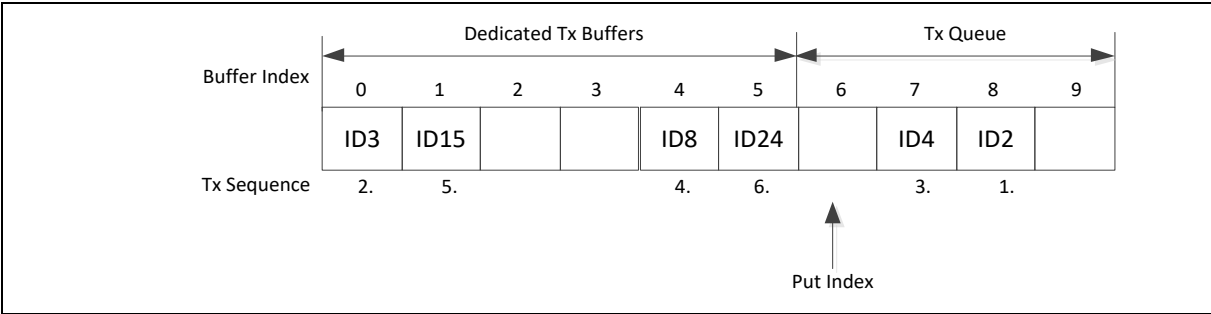


Figure 6.20-10 Example of mixed Configuration Dedicated Tx Buffers / Tx Queue

Tx prioritization:

- Scan all Tx Buffers with activated transmission request

- Tx Buffer with lowest Message ID gets highest priority and is transmitted next

Transmit Cancellation

The CAN FD controller supports transmit cancellation. This feature is especially intended for gateway applications and AUTOSAR based applications. To cancel a requested transmission from a dedicated Tx Buffer or a Tx Queue Buffer the software has to write a 1 to the corresponding bit position (=number of Tx Buffer) of register CANFD_TXBCR. Transmit cancellation is not intended for Tx FIFO operation.

Successful cancellation is signaled by setting the corresponding bit of register CANFD_TXBCF to 1. In case a transmit cancellation is requested while a transmission from a Tx Buffer is already ongoing, the corresponding CANFD_TXBRP bit remains set as long as the transmission is in progress. If the transmission was successful, the corresponding CANFD_TXBTO and CANFD_TXBCF bits are set. If the transmission was not successful, it is not repeated and only the corresponding CANFD_TXBCF bit is set.

Note: In case a pending transmission is cancelled immediately before this transmission could have been started, there follows a short time window where no transmission is started even if another message is also pending in this node. This may enable another node to transmit a message which may have a lower priority than the second message in this node.

Tx Event Handling

To support Tx event handling the controller has implemented a Tx Event FIFO. After the controller has transmitted a message on the CAN bus, message ID and timestamp are stored in a Tx Event FIFO element. To link a Tx event to a Tx Event FIFO element, the message marker from the transmitted Tx Buffer is copied into the Tx Event FIFO element.

The Tx Event FIFO can be configured to a maximum of 32 elements. The Tx Event FIFO element is described in **Tx Event FIFO Element** section.

The purpose of the Tx Event FIFO is to decouple handling transmit status information from transmit message handling i.e. a Tx Buffer holds only the message to be transmitted, while the transmit status is stored separately in the Tx Event FIFO. This has the advantage, especially when operating a dynamically managed transmit queue, that a Tx Buffer can be used for a new message immediately after successful transmission. There is no need to save transmit status information from a Tx Buffer before overwriting that Tx Buffer.

When a Tx Event FIFO full condition is signaled by TEFF (CANFD_IR[14]), no further elements are written to the Tx Event FIFO until at least one element has been read out and the Tx Event FIFO get Index has been incremented. In case a Tx event occurs while the Tx Event FIFO is full, this event is discarded and interrupt flag TEFL (CANFD_IR[15]) is set.

To avoid a Tx Event FIFO overflow, the Tx Event FIFO watermark can be used. When the Tx Event FIFO fill level reaches the Tx Event FIFO watermark configured by EFWM (CANFD_TXEFC[29:24]), interrupt flag TEFW (CANFD_IR[13]) is set.

When reading from the Tx Event FIFO, two times the Tx Event FIFO get Index EFGI (CANFD_TXEFS[12:8]) has to be added to the Tx Event FIFO start address EFSA (CANFD_TXEFC[15:2]).

6.20.5.6 FIFO Acknowledge Handling

The get indices of Rx FIFO 0, Rx FIFO 1, and the Tx Event FIFO are controlled by writing to the corresponding FIFO Acknowledge Index (refer to **Rx FIFO 0 Acknowledge**, **Rx FIFO 1 Acknowledge**, and **Tx Event FIFO Acknowledge**). Writing to the FIFO Acknowledge Index will set the FIFO Get Index to the FIFO Acknowledge Index plus one and thereby updates the FIFO Fill Level. There are two use cases:

- When only a single element has been read from the FIFO (the one being pointed to by the get index), this get index value is written to the FIFO Acknowledge Index.
- When a sequence of elements has been read from the FIFO, it is sufficient to write the

FIFO Acknowledge Index only once at the end of that read sequence (value: Index of the last element read), to update the FIFO's get index.

Due to the fact that the CPU has free access to the controller Message RAM, special care has to be taken when reading FIFO elements in an arbitrary order (get index not considered). This might be useful when reading a High Priority Message from one of the two Rx FIFOs. In this case the FIFO's Acknowledge Index should not be written because this would set the get index to a wrong position and also alters the FIFO's Fill Level. In this case some of the older FIFO elements would be lost.

Note: The application has to ensure that a valid value is written to the FIFO Acknowledge Index. The controller does not check for erroneous values.

6.20.5.7 Message RAM

For storage of Rx/Tx messages and for storage of the filter configuration a single-ported Message RAM has to be connected to the CAN Core module.

Message RAM Configuration

The address of the Message RAM and the address of the register are accessed through the same AHB interface. The maximum address of register used by the CAN FD controller is within 0x200. Therefore, the message SRAM used by the CAN controller must be used after 0x200. When reading or writing the SRAM content, the CAN controller will automatically subtract the offset of 0x200.

For example:

When the CPU wants to access the location of SRAM address is 0x100, its AHB address should be set as CANFD Base address + 0x200+0x100.

The Message RAM has a width of 32 bits. The CAN FD controller can be configured to allocate up to 256 words in the Message RAM. It is not necessary to configure each of the sections listed in Figure 6.20-11, nor is there any restriction with respect to the sequence of the sections.

When operated in CAN FD mode the required Message RAM size strongly depends on the element size configured for Rx FIFO0, Rx FIFO1, Rx Buffers, and Tx Buffers via RXESC.F0DS, RXESC.F1DS, RXESC.RBDS, and TXESC.TBDS.

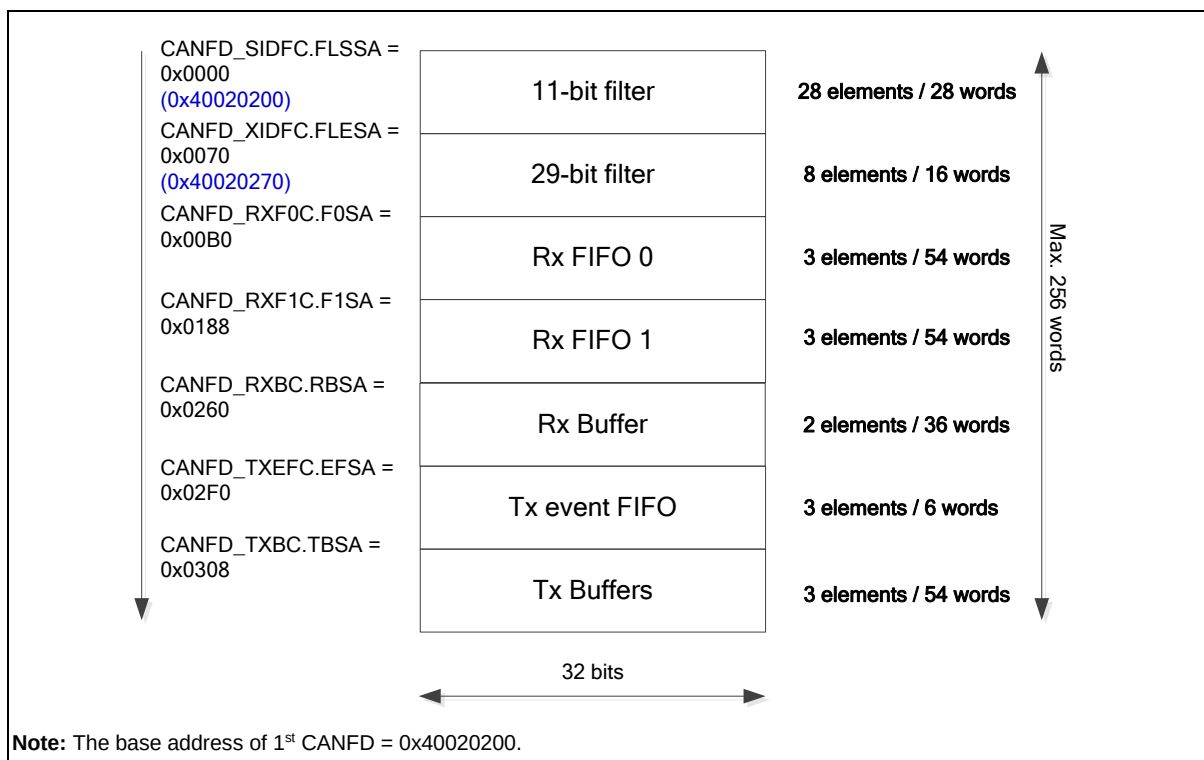


Figure 6.20-11 Message RAM Configuration

When the controller addresses the Message RAM it addresses 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses i.e. only bits 15 to 2 are evaluated, the two least significant bits are ignored.

Rx Buffer and FIFO Element

Two Rx FIFOs and up to 64 Rx Buffers can be configured in the Message RAM. Each Rx FIFO section can be configured to store up to 64 received messages. The structure of a Rx Buffer / FIFO element is shown in below. The element size can be configured for storage of CAN FD messages with up to 64 bytes data field via register CANFD_RXESC.

Bit	31	24	23	16	15	8	7	0
R0	E S I	X T D	R T R	ID[28:0]				
R1	A N M F	FIDX[6:0]		-	-	F D F	B R S	DLC[3:0]
R2	DB3[7:0]		DB2[7:0]		DB1[7:0]		DB0[7:0]	
R3	DB7[7:0]		DB6[7:0]		DB5[7:0]		DB4[7:0]	

Rn	DBm[7:0]		DBm-1[7:0]		DBm-2[7:0]		DBm-3[7:0]	

Table 6.20-6 Rx Buffer and FIFO Element

Bit Field	Description
R0 Bit 31 ESI	Error State Indicator 0 = Transmitting node is error active 1 = Transmitting node is error passive
R0 Bit 30 XTD	Extended Identifier Signals to the MCU whether the received frame has a standard or extended identifier. 0 = 11-bit standard identifier 1 = 29-bit extended identifier
R0 Bit 29 RTR	Remote Transmission Request Signals to the Host whether the received frame is a data frame or a remote frame. 0 = Received frame is a data frame 1 = Received frame is a remote frame Note: There are no remote frames in CAN FD format. In CAN FD frames (FDF = 1'), the dominant RRS (Remote Request Substitution) bit replaces bit RTR (Remote Transmission Request)
R0 Bits 28:0 ID[28:0]	Identifier Standard or extended identifier depending on bit XTD. A standard identifier is stored into ID[28:18].
R1 Bit 31 ANMF	Accepted Non-matching Frame Acceptance of non-matching frames may be enabled via GFC.ANFS and GFC.ANFE. 0 = Received frame matching filter index FIDX 1 = Received frame did not match any Rx filter element

R1 Bits 30:24 FIDX[6:0]	Filter Index 0-127 = Index of matching Rx acceptance filter element (invalid if ANMF = '1'). Note: Range is 0 to SIDFC.LSS – 1 for standard and 0 to XIDFC.LSE – 1 for extended.
R1 Bit 21 FDF	FD Format 0 = Classical CAN frame format 1 = CAN FD frame format (new DLC-coding and CRC)
R1 Bit 20 BRS	Bit Rate Switch 0 = Frame received without bit rate switching 1 = Frame received with bit rate switching
R1 Bits 19:16 DLC[3:0]	Data Length Code 0-8 = CAN + CAN FD: received frame has 0-8 data bytes 9-15 = CAN: received frame has 8 data bytes 9-15 = CAN FD: received frame has 12/16/20/24/32/48/64 data bytes
R1 Bits 15:0 RXTS[15:0]	Rx Timestamp Timestamp Counter value captured on start of frame reception. Resolution depending on configuration of the Timestamp Counter Prescaler TCP (CANFD_TSCC[19:16]).
R2 Bits 31:24 DB3[7:0]	Data Byte 3
R2 Bits 23:16 DB2[7:0]	Data Byte 2
R2 Bits 15:8 DB1[7:0]	Data Byte 1
R2 Bits 7:0 DB0[7:0]	Data Byte 0
R3 Bits 31:24 DB7[7:0]	Data Byte 7
R3 Bits 23:16 DB6[7:0]	Data Byte 6
R3 Bits 15:8 DB5[7:0]	Data Byte 5
R3 Bits 7:0 DB4[7:0]	Data Byte 4
...	...
Rn Bits 31:24 DBm[7:0]	Data Byte mm
Rn Bits 23:16 DBm-1[7:0]	Data Byte m-1
Rn Bits 15:8 DBm-2[7:0]	Data Byte m-2
Rn Bits 7:0 DBm-3[7:0]	Data Byte m-3 Note: Depending on the configuration of the element size (RXESC), between two and sixteen 32-bit words (Rn = 2 ...17) are used for storage of a CAN message's data field

Table 6.20-7 Rx Buffer and FIFO Element Description

Tx Buffer Element

The Tx Buffers section can be configured to hold dedicated Tx Buffers as well as a Tx FIFO / Tx Queue. In case that the Tx Buffers section is shared by dedicated Tx buffers and a Tx FIFO / Tx Queue, the dedicated Tx Buffers start at the beginning of the Tx Buffers section followed by the buffers assigned to the Tx FIFO or Tx Queue. The Tx Handler distinguishes between dedicated Tx Buffers and Tx FIFO / Tx Queue by evaluating the Tx Buffer configuration TFQS (CANFD_TXBC[29:24]) and NDTB (CANFD_TXBC[21:16]). The element size can be configured for storage of CAN FD messages with up to 64 bytes data field via register TXESC.

Bit	31	24	23	16	15	8	7	0
T0	E S I	X T D	R T R	ID[28:0]				
T1	MM[7:0]		EFC	-	FDF	BRS	DLC[3:0]	MM[15:8]
T2	DB3[7:0]		DB2[7:0]			DB1[7:0]		DB0[7:0]
T3	DB7[7:0]		DB6[7:0]			DB5[7:0]		DB4[7:0]

Tn	DBm[7:0]		DBm-1[7:0]			DBm-2[7:0]		DBm-3[7:0]

Table 6.20-8 Tx Buffer and FIFO Element

Bit Field	Description
T0 Bit 31 ESI	Error State Indicator 0 = ESI bit in CAN FD format depends only on error passive flag 1 = ESI bit in CAN FD format transmitted recessive Note: The ESI bit of the transmit buffer is OR-ed with the error passive flag to decide the value of the ESI bit in the transmitted FD frame. As required by the CAN FD protocol specification, an error active node may optionally transmit the ESI bit recessive, but an error passive node will always transmit the ESI bit recessive.
T0 Bit 30 XTD	Extended Identifier 0 = 11-bit standard identifier 1 = 29-bit extended identifier
T0 Bit 29 RTR	Remote Transmission Request 0 = Transmit data frame 1 = Transmit remote frame Note: When RTR = 1, the controller transmits a remote frame according to ISO 11898-1:2015, even if FDOE (CANFD_CCCR[8]) enables the transmission in CAN FD format.
T0 Bits 28:0 ID[28:0]	Identifier Standard or extended identifier depending on bit XTD. A standard identifier has to be written to ID[28:18].
T1 Bits 31:24 MM[7:0]	Message Marker Written by CPU during Tx Buffer configuration. Copied into Tx Event FIFO element for identification of Tx message status.
T1 Bit 23 EFC	Event FIFO Control 0 = Don't store Tx events 1 = Store Tx events
T1 Bit 21 FDF	FD Format 0 = Frame transmitted in Classic CAN format 1 = Frame transmitted in CAN FD format

T1 Bit 20 BRS	Bit Rate Switch 0 = CAN FD frames transmitted without bit rate switching 1 = CAN FD frames transmitted with bit rate switching Note: Bits ESI, FDF, and BRS are only evaluated when CAN FD operation is enabled FDOE (CANFD_CCCR[8]) = 1. Bit BRS is only evaluated when in addition BRSE (CANFD_CCCR[9]) = 1.
T1 Bits 19:16 DLC[3:0]	Data Length Code 0-8 = CAN + CAN FD: transmit frame has 0-8 data bytes 9-15 = CAN: transmit frame has 8 data bytes 9-15 = CAN FD: transmit frame has 12/16/20/24/32/48/64 data bytes
T2 Bits 31:24 DB3[7:0]	Data Byte 3
T2 Bits 23:16 DB2[7:0]	Data Byte 2
T2 Bits 15:8 DB1[7:0]	Data Byte 1
T2 Bits 7:0 DB0[7:0]	Data Byte 0
T3 Bits 31:24 DB7[7:0]	Data Byte 7
T3 Bits 23:16 DB6[7:0]	Data Byte 6
T3 Bits 15:8 DB5[7:0]	Data Byte 5
T3 Bits 7:0 DB4[7:0]	Data Byte 4
... ..	
Tn Bits 31:24 DBm[7:0]	Data Byte m
Tn Bits 23:16 DBm-1[7:0]	Data Byte m-1
Tn Bits 15:8 DBm-2[7:0]	Data Byte m-2
Tn Bits 7:0 DBm-3[7:0]	Data Byte m-3 Note: Depending on the configuration of the element size (TXESC), between two and sixteen 32-bit words (Tn = 2 ..17) are used for storage of a CAN message's data field.

Table 6.20-9 Tx Buffer Element Description

Tx Event FIFO Element

Each element stores information about transmitted messages. By reading the Tx Event FIFO the MCU gets this information in the order the messages were transmitted. Status information about the Tx Event FIFO can be obtained from register CANFD_TXEFS.

Bit	31	24	23	16	15	8	7	0
E0	ESI	XTD	RTR	ID[28:0]				
E1	MM[7:0]			ET [1:0]	FDF	BRS	DLC[3:0]	TXTS[15:0]

Table 6.20-10 Tx Event FIFO Element

Bit Field	Description
E0 Bit 31 ESI	Error State Indicator 0 = Transmitting node is error active 1 = Transmitting node is error passive
E0 Bit 30 XTD	Extended Identifier 0 = 11-bit standard identifier 1 = 29-bit extended identifier
E0 Bit 29 RTR	Remote Transmission Request 0 = Data frame transmitted 1 = Remote frame transmitted
E0 Bits 28:0 ID[28:0]	Identifier Standard or extended identifier depending on bit XTD. A standard identifier is stored into ID[28:18].
E1 Bits 31:24 MM[7:0]	Message Marker Copied from Tx Buffer into Tx Event FIFO element for identification of Tx message status.
E1 Bit 23:22 ET[1:0]	Event Type 00 = Reserved 01 = Tx event 10 = Transmission in spite of cancellation (always set for transmissions in DAR mode) 11 = Reserved
E1 Bit 21 FDF	FD Format 0 = Classical CAN frame format 1 = CAN FD frame format (new DLC-coding and CRC)
E1 Bit 20 BRS	Bit Rate Switch 0 = Frame transmitted without bit rate switching 1 = Frame transmitted with bit rate switching
E1 Bits 19:16 DLC[3:0]	Data Length Code 0-8 = CAN + CAN FD: frame with 0-8 data bytes transmitted 9-15 = CAN: frame with 8 data bytes transmitted 9-15 = CAN FD: frame with 12/16/20/24/32/48/64 data bytes transmitted
E1 Bits 15:0 TXTS[15:0]	Tx Timestamp Timestamp Counter value captured on start of frame transmission. Resolution depending on configuration of the Timestamp Counter Prescaler TCP (CANFD_TSCC[19:16]).

Table 6.20-11 Tx Event FIFO Element Description

Standard Message ID Filter Element

Up to 128 filter elements can be configured for 11-bit IDs. When accessing a Standard Message ID Filter element, its address is the Filter List Standard Start Address FLSSA (CANFD_SIDFC[15:2]) plus the index of the filter element (0...127).

Bit	31	24	23	16	15	8	7	0
S0	SFT [1:0]	SFEC [2:0]	SFID1[10:0]			-	SFID2[10:0]	

Table 6.20-12 Standard Message ID Filter Element

Bit Field		Description
Bits 31:30 SFT[1:0]		Standard Filter Type 00 = Range filter from SFID1 to SFID2 ($SFID2 \geq SFID1$) 01 = Dual ID filter for SFID1 or SFID2 10 = Classic filter: SFID1 = filter, SFID2 = mask 11 = Filter element disabled Note: With SFT = 11 the filter element is disabled and the acceptance filtering continues (same behavior as with SFEC = 000)
Bit 29:27 SFEC[2:0]		Standard Filter Element Configuration All enabled filter elements are used for acceptance filtering of 11-bit ID frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If SFEC = 100, 101, or 110 a match sets interrupt flag HPM (CANFD_IR[8]) and, if enabled, an interrupt is generated. In this case register HPMS is updated with the status of the priority match. 000 = Disable filter element 001 = Store in Rx FIFO 0 if filter matches 010 = Store in Rx FIFO 1 if filter matches 011 = Reject ID if filter matches, not intended to be used with Sync messages 100 = Set priority if filter matches, not intended to be used with Sync messages, no storage 101 = Set priority and store in FIFO 0 if filter matches 110 = Set priority and store in FIFO 1 if filter matches 111 = Store into Rx Buffer, configuration of SFT[1:0] ignored
Bits 26:16 SFID1[10:0]		Standard Filter ID 1 First ID of standard ID filter element. When filtering for Rx Buffers, Sync messages, this field defines the ID of the message to be stored. The received identifiers must match exactly, no masking mechanism is used.
Bits 15:0	SFID2[10:0]	Standard filter ID 2 This bit field has a different meaning depending on the configuration of SFEC: SFEC = 001 ... 110 Second ID of standard ID filter element SFEC = 111 Filter for Rx buffers
	SFID2[10:9]	Decides whether the received message is stored into an Rx buffer . 00: Store message into an Rx buffer
	SFID2[8:6]	The bits are used to control the filter event pins at the Extension Interface. A '1' at the respective bit position enables generation of a pulse at the related filter event pin with the duration of one system clock period in case the filter matches. SFID2[8] is used by the calibration unit.
	SFID2[5:0]	Defines the offset to the Rx buffer start address RBSA (CANFD_RXBC[15:2]) for storage of a matching message.

Table 6.20-13 Standard Message ID Filter Element Description

Extended Message ID Filter Element

Up to 64 filter elements can be configured for 29-bit IDs. When accessing an Extended Message ID Filter element, its address is the Filter List Extended Start Address XIDFC.FLESA plus two times the index of the filter element (0...63).

Bit	31	24	23	16	15	8	7	0
F0	EFECV [2:0]		EFID1[28:0]					
F0	EFT [1:0]	-	EFID2[28:0]					

Table 6.20-14 Extended Message ID Filter Element

Field		Description
F0 Bit 31:29 EFEC[2:0]		Extended Filter Element Configuration All enabled filter elements are used for acceptance filtering of 29-bit ID frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If EFEC = 100, 101, or 110 a match sets interrupt flag HPM (CANFD_IR[8]) and, if enabled, an interrupt is generated. In this case register HPMS is updated with the status of the priority match. 000 = Disable filter element 001 = Store in Rx FIFO 0 if filter matches 010 = Store in Rx FIFO 1 if filter matches 011 = Reject ID if filter matches 100 = Set priority if filter matches 101 = Set priority and store in FIFO 0 if filter matches 110 = Set priority and store in FIFO 1 if filter matches 111 = Store into Rx Buffer, configuration of EFT[1:0] ignored
F0 Bits 28:0 EFID1[28:0]		Extended Filter ID 1 First ID of extended ID filter element. When filtering for Rx Buffers, Sync messages, this field defines the ID of an extended message to be stored. The received identifiers must match exactly, only CANFD_XIDAM masking mechanism (refer to Extended Message ID Filtering) is used.
F1 Bits 31:30 EFT[1:0]		Extended Filter Type 00 = Range filter from EFID1 to EFID2 (EFID2 > EFID1) 01 = Dual ID filter for EFID1 or EFID2 10 = Classic filter: EFID1 = filter, EFID2 = mask 11 = Range filter from EFID1 to EFID2 (EFID2 >= EFID1), CANFD_XIDAM mask not applied.
F1 Bits 28:0	EFID2[10:0]	Extended filter ID 2 This bit field has a different meaning depending on the configuration of EFEC: – SFEC = 001 ... 110 Second ID of extended ID filter element – SFEC = 111 Filter for Rx buffers
	EFID2[10:9]	Decides whether the received message is stored into an Rx Buffer. 00 = Store message into an Rx Buffer
	EFID2[8:6]	Is used to control the filter event pins at the Extension Interface. A one at the respective bit position enables generation of a pulse at the related filter event pin with the duration of one system clock period in case the filter matches.
	EFID2[5:0]	Defines the offset to the Rx Buffer Start Address RBSA (CANFD_RXBC[15:2]) for storage of a matching message.

Table 6.20-15 Extended Message ID Filter Element Description

6.20.5.8 Bit Timing

The bit timing logic monitors the serial bus-line and performs sampling and adjustment of the sample point by synchronizing on the start-bit edge and resynchronizing on the following edges.

As shown in Figure 6.20-12, its operation may be explained simply by splitting the bit time in three segments, as follows:

- Synchronization segment (SYNC_SEG): a bit change is expected to occur within this time segment, having a fixed length of one time quantum ($1 \times tq$).
- Bit segment 1 (BS1): defines the location of the sample point. It includes the PROP_SEG and PHASE_SEG1 of the CAN standard. Its duration is programmable between 1 and 16 time quanta, but may be automatically lengthened to compensate for positive phase drifts due to differences in the frequency of various nodes of the network.
- Bit segment 2 (BS2): defines the location of the transmit point. It represents the PHASE_SEG2 of the CAN standard, its duration is programmable between one and eight time quanta, but may also be automatically shortened to compensate for negative phase drifts.

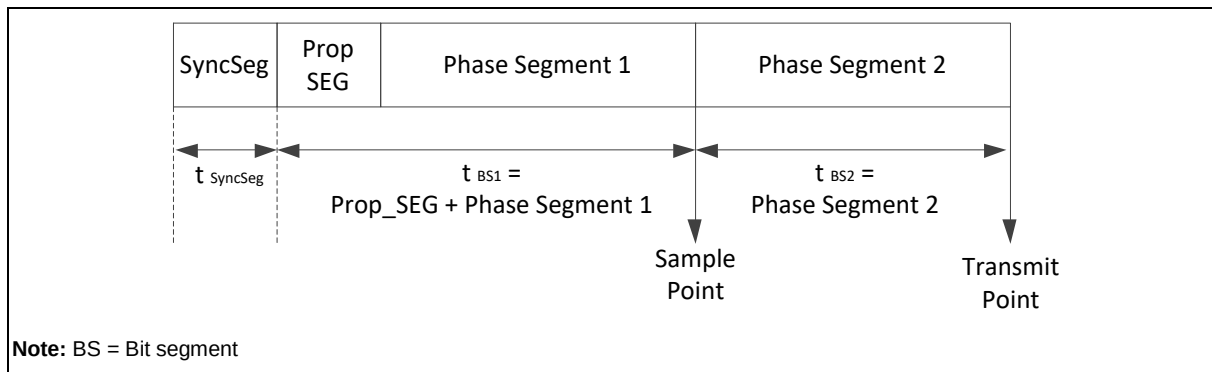


Figure 6.20-12 Bit Timing

The baud rate is the inverse of bit time (baud rate = $1 / \text{bit time}$), which, in turn, is the sum of three components. Figure 6.20-12 indicates that bit time = $t_{SyncSeg} + t_{BS1} + t_{BS2}$, where:

for the nominal bit time

- $tq = (NBTP.NBRP[8:0] + 1) * t_{canfd\_tq\_clk}$
- $t_{SyncSeg} = 1 tq$
- $t_{BS1} = tq * (NBTP.NTSEG1[7:0] + 1)$
- $t_{BS2} = tq * (NBTP.NTSEG2[6:0] + 1)$

for the data bit time

- $tq = (DBTP.DBRP[4:0] + 1) * t_{canfd\_tq\_clk}$
- $t_{SyncSeg} = 1 tq$
- $t_{BS1} = tq * (DBTP.DTSEG1[4:0] + 1)$
- $t_{BS2} = tq * (DBTP.DTSEG2[3:0] + 1)$

The (Re)Synchronization Jump Width (SJW) defines an upper bound for the amount of lengthening or shortening of the bit segments. It is programmable between one and four time quanta.

A valid edge is defined as the first transition in a bit time from dominant to recessive bus level, provided the controller itself does not send a recessive bit.

If a valid edge is detected in BS1 instead of SYNC_SEG. BS1 is extended by up to SJW so that the

sample point is delayed.

Conversely, if a valid edge is detected in BS2 instead of SYNC_SE.G. BS2 is shortened by up to SJW so that the transmit point is moved earlier.

As a safeguard against programming errors, the configuration of the Bit Timing register is only possible while the device is in Standby mode. Registers CANFD_DBTP and CANFD_NBTP (dedicated, respectively, to data and nominal bit timing) are only accessible when CCE (CANFD_CCCR[1]) and INIT (CANFD_CCCR[0]) are set.

Note: For a detailed description of the CAN bit timing and resynchronization mechanism, refer to the ISO 11898-1 standard. <http://www.bittiming.can-wiki.info/>

6.20.6 Register Map

R: read only, W: write only, R/W: both read and write

Register	Offset	R/W	Description	Reset Value
CANFD Base Address: CANFDx_BA = 0x4002_0000				
CANFD_DBTP	CANFDx_BA+0x00C	R	Data Bit Timing & Prescaler Register (P*)	0x0000_0A33
CANFD_TEST	CANFDx_BA+0x010	R	Test Register (P*)	0x0000_0000
CANFD_RWD	CANFDx_BA+0x014	R	RAM Watchdog (P*)	0x0000_0000
CANFD_CCCR	CANFDx_BA+0x018	R/W	CC Control Register (Pp*)	0x0000_0001
CANFD_NBTP	CANFDx_BA+0x01C	R	Nominal Bit Timing & Prescaler Register (P*)	0x0600_0A03
CANFD_TSCC	CANFDx_BA+0x020	R	Timestamp Counter Configuration (P*)	0x0000_0000
CANFD_TSCV	CANFDx_BA+0x024	R	Timestamp Counter Value (C*)	0x0000_0000
CANFD_TOCC	CANFDx_BA+0x028	R	Timeout Counter Configuration (P*)	0xFFFF_0000
CANFD_TOCV	CANFDx_BA+0x02C	R	Timeout Counter Value (C*)	0x0000_FFFF
CANFD_ECR	CANFDx_BA+0x040	R	Error Counter Register (X*)	0x0000_0000
CANFD_PSR	CANFDx_BA+0x044	R	Protocol Status Register (XS*)	0x0000_0707
CANFD_TDCR	CANFDx_BA+0x048	R	Transmitter Delay Compensation Register (P*)	0x0000_0000
CANFD_IR	CANFDx_BA+0x050	R/W	Interrupt Register	0x0000_0000
CANFD_IE	CANFDx_BA+0x054	R/W	Interrupt Enable	0x0000_0000
CANFD_ILS	CANFDx_BA+0x058	R/W	Interrupt Line Select	0x0000_0000
CANFD_ILE	CANFDx_BA+0x05C	R/W	Interrupt Line Enable	0x0000_0000
CANFD_GFC	CANFDx_BA+0x080	R	Global Filter Configuration (P*)	0x0000_0000
CANFD_SIDFC	CANFDx_BA+0x084	R	Standard ID Filter Configuration (P*)	0x0000_0000
CANFD_XIDFC	CANFDx_BA+0x088	R	Extended ID Filter Configuration (P*)	0x0000_0000
CANFD_XIDAM	CANFDx_BA+0x090	R	Extended ID AND Mask (P*)	0x1FFF_FFFF
CANFD_HPMS	CANFDx_BA+0x094	R	High Priority Message Status	0x0000_0000
CANFD_NDAT1	CANFDx_BA+0x098	R/W	New Data 1	0x0000_0000
CANFD_NDAT2	CANFDx_BA+0x09C	R/W	New Data 2	0x0000_0000
CANFD_RXF0C	CANFDx_BA+0x0A0	R	Rx FIFO 0 Configuration (P*)	0x0000_0000
CANFD_RXF0S	CANFDx_BA+0x0A4	R	Rx FIFO 0 Status	0x0000_0000
CANFD_RXF0A	CANFDx_BA+0x0A8	R/W	Rx FIFO 0 Acknowledge	0x0000_0000
CANFD_RXBC	CANFDx_BA+0x0AC	R	Rx Buffer Configuration (P*)	0x0000_0000

CANFD_RXF1C	CANFDx_BA+0x0B0	R	Rx FIFO 1 Configuration (P*)	0x0000_0000
CANFD_RXF1S	CANFDx_BA+0x0B4	R	Rx FIFO 1 Status	0x0000_0000
CANFD_RXF1A	CANFDx_BA+0x0B8	R/W	Rx FIFO 1 Acknowledge	0x0000_0000
CANFD_RXESC	CANFDx_BA+0x0BC	R	Rx Buffer / FIFO Element Size Configuration (P*)	0x0000_0000
CANFD_TXBC	CANFDx_BA+0x0C0	R	Tx Buffer Configuration (P*)	0x0000_0000
CANFD_TXFQS	CANFDx_BA+0x0C4	R	Tx FIFO/Queue Status	0x0000_0000
CANFD_TXESC	CANFDx_BA+0x0C8	R	Tx Buffer Element Size Configuration (P*)	0x0000_0000
CANFD_TXBRP	CANFDx_BA+0x0CC	R	Tx Buffer Request Pending	0x0000_0000
CANFD_TXBAR	CANFDx_BA+0x0D0	R/W	Tx Buffer Add Request	0x0000_0000
CANFD_TXBCR	CANFDx_BA+0x0D4	R/W	Tx Buffer Cancellation Request	0x0000_0000
CANFD_TXBTO	CANFDx_BA+0x0D8	R	Tx Buffer Transmission Occurred	0x0000_0000
CANFD_TXBCF	CANFDx_BA+0x0DC	R	Tx Buffer Cancellation Finished	0x0000_0000
CANFD_TXBTIE	CANFDx_BA+0x0E0	R/W	Tx Buffer Transmission Interrupt Enable	0x0000_0000
CANFD_TXBCIE	CANFDx_BA+0x0E4	R/W	Tx Buffer Cancellation Finished Interrupt Enable	0x0000_0000
CANFD_TXEFC	CANFDx_BA+0x0F0	R	Tx Event FIFO Configuration (P*)	0x0000_0000
CANFD_TXEFS	CANFDx_BA+0x0F4	R	Tx Event FIFO Status	0x0000_0000
CANFD_TXEFA	CANFDx_BA+0x0F8	R/W	Tx Event FIFO Acknowledge	0x0000_0000
Note: R = Read, S = Set on read, X = Reset on read, W = Write, P = Protected write, p = Protected set, C = Clear/preset on write				

6.20.7 Register Description

CANFD Data Bit Timing & Prescaler Register (CANFD_DBTP)

Register	Offset	R/W	Description	Reset Value
CANFD_DBTP	CANFDx_BA+0x00C	R	Data Bit Timing & Prescaler Register (P*)	0x0000_0A33

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
TDC	Reserved		DBRP				
15	14	13	12	11	10	9	8
Reserved			DTSEG1				
7	6	5	4	3	2	1	0
DTSEG2				DSJW			

Bits	Description
[31:24]	Reserved Reserved.
[23]	TDC Transmitter Delay Compensation 0 = Transmitter Delay Compensation Disabled. 1 = Transmitter Delay Compensation Enabled.
[22:21]	Reserved Reserved.
[20:16]	DBRP Data Bit Rate Prescaler The value by which the oscillator frequency is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid values for the Bit Rate Prescaler are 0 to 31. When TDC = '1', the range is limited to 0,1. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.
[15:13]	Reserved Reserved.
[12:8]	DTSEG1 Data time segment before sample point Valid values are 0 to 31. The actual interpretation by the hardware of this value is such that one more than the programmed value is used. $t_{BS1} = (DTSEG1 + 1) \times tq$.
[7:4]	DTSEG2 Data time segment after sample point Valid values are 0 to 15. The actual interpretation by the hardware of this value is such that one more than the programmed value is used. $t_{BS2} = (DTSEG2 + 1) \times tq$.
[3:0]	DSJW Data Re-Synchronization Jump Width Valid values are 0 to 15. The actual interpretation by the hardware of this value is such that one more than the programmed value is used. $t_{SJW} = (DSJW + 1) \times tq$.

CANFD TEST Register (CANFD_TEST)

Register	Offset	R/W	Description	Reset Value
CANFD_TEST	CANFDx_BA+0x010	R	Test Register (P*)	0x0000_0000

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

Bits	Description	
[31:8]	Reserved	Reserved.
[7]	RX	Receive Pin Monitors the actual value of pin CANx_RXD 0 = The CAN bus is dominant (CANx_RXD = 0). 1 = The CAN bus is recessive (CANx_RXD = 1).
[6:5]	TX	Control of Transmit Pin 00 = Reset value, CANx_TXD controlled by the CAN Core, updated at the end of the CAN bit time. 01 = Sample Point can be monitored at pin CANx_TXD. 10 = Dominant ('0') level at pin CANx_TXD. 11 = Recessive ('1') level at pin CANx_TXD.
[4]	LBCK	Loop Back Mode 0 = Reset value, Loop Back Mode Disabled. 1 = Loop Back Mode Enabled (refer to TEST Mode in the Operation Mode section).
[3:0]	Reserved	Reserved.

CANFD RAM Watchdog Register (CANFD_RWD)

The RAM watchdog monitors the READY output of the message RAM. A message RAM access starts the message RAM watchdog counter with the value configured by the WDC (CANFD_RWD[7:0]) bits. The counter is reloaded with WDC bits when the message RAM signals successful completion by activating its READY output. In case there is no response from the message RAM until the counter has counted down to 0, the counter stops and interrupt flag WDI (CANFD_IR[26]) bit is set.

Register	Offset	R/W	Description	Reset Value
CANFD_RWD	CANFDx_BA+0x014	R	RAM Watchdog (P*)	0x0000_0000

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

Bits	Description
[31:16]	Reserved Reserved.
[15:8]	WDV Watchdog Value Actual Message RAM Watchdog Counter Value.
[7:0]	WDC Watchdog Configuration Start value of the Message RAM Watchdog Counter. With the reset value of 00 the counter is disabled.

CANFD CC Control Register (CANFD_CCCR)

Register	Offset	R/W	Description	Reset Value
CANFD_CCCR	CANFDx_BA+0x018	R/W	CC Control Register (Pp*)	0x0000_0001

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
NISO	TXP	EFBI	PXHD	Reserved		BRSE	FDOE
7	6	5	4	3	2	1	0
TEST	DAR	MON	CSR	CSA	ASM	CCE	INIT

Bits	Description
[31:16]	Reserved Reserved.
[15]	NISO Non ISO Operation If this bit is set, the CAN FD controller uses the CAN FD frame format as specified by the Bosch CAN FD Specification V1.0. 0 = CAN FD frame format according to ISO 11898-1:2015. 1 = CAN FD frame format according to Bosch CAN FD Specification V1.0.
[14]	TXP Transmit Pause If this bit is set, the CAN FD controller pauses for two CAN bit times before starting the next transmission after itself has successfully transmitted a frame (refer to Tx Handling section). 0 = Transmit pause Disabled. 1 = Transmit pause Enabled.
[13]	EFBI Edge Filtering during Bus Integration 0 = Edge filtering Disabled. 1 = Two consecutive dominant tq required to detect an edge f or hard synchronization.
[12]	PXHD Protocol Exception Handling Disable 0 = Protocol exception handling Enabled. 1 = Protocol exception handling Disabled. Note: When protocol exception handling is disabled, the controller will transmit an error frame when it detects a protocol exception condition.
[11:10]	Reserved Reserved.
[9]	BRSE Bit Rate Switch Enable 0 = Bit rate switching for transmissions Disabled. 1 = Bit rate switching for transmissions Enabled. Note: When CAN FD operation is disabled FDOE = 0, BRSE is not evaluated.
[8]	FDOE FD Operation Enable 0 = FD operation Disabled. 1 = FD operation Enabled.

[7]	TEST	Test Mode Enable 0 = Normal operation, register TEST holds reset values. 1 = Test Mode, write access to register TEST enabled.
[6]	DAR	Disable Automatic Retransmission 0 = Automatic retransmission of messages not transmitted successfully Enabled. 1 = Automatic retransmission Disabled.
[5]	MON	Bus Monitoring Mode Bit MON can only be set by the Host when both CCE and INIT are set to 1. The bit can be reset by the Host at any time. 0 = Bus Monitoring Mode Disabled. 1 = Bus Monitoring Mode Enabled.
[4]	CSR	Clock Stop Request 0 = No clock stop is requested. 1 = Clock stop requested. When clock stop is requested, first INIT and then CSA will be set after all pending transfer requests have been completed and the CAN bus reached idle.
[3]	CSA	Clock Stop Acknowledge 0 = No clock stop acknowledged. 1 = The Controller may be set in power down by stopping AHB clock and CAN Core clock.
[2]	ASM	Restricted Operation Mode Bit ASM can only be set by the Host when both CCE and INIT are set to 1. The bit can be reset by the software at any time. This bit will be set automatically set to 1 when the Tx handler was not able to read data from the message RAM in time. For a description of the Restricted Operation Mode refer to Restricted Operation Mode . 0 = Normal CAN operation. 1 = Restricted Operation Mode active.
[1]	CCE	Configuration Change Enable 0 = The CPU has no write access to the protected configuration registers. 1 = The CPU has write access to the protected configuration registers (while CANFD_INIT (CANFD_CCCR[0]) = 1).
[0]	INIT	Initialization 0 = Normal Operation. 1 = Initialization is started. Note: Due to the synchronization mechanism between the two clock domains, there may be a delay until the value written to INIT can be read back. Therefore the programmer has to assure that the previous value written to INIT has been accepted by reading INIT before setting INIT to a new value.

CANFD Nominal Bit Timing & Prescaler Register (CANFD_NBTP)

Register	Offset	R/W	Description	Reset Value
CANFD_NBTP	CANFDx_BA+0x01C	R	Nominal Bit Timing & Prescaler Register (P*)	0x0600_0A03

31	30	29	28	27	26	25	24
NSJW							NBRP
23	22	21	20	19	18	17	16
NBRP							
15	14	13	12	11	10	9	8
NTSEG1							
7	6	5	4	3	2	1	0
Reserved	NTSEG2						

Bits	Description
[31:25] NSJW	Nominal Re-Synchronization Jump Width Valid values are 0 to 127, which should be smaller than NTSEG2. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used. $t_{SJW} = (NSJW + 1) \times t_q$.
[24:16] NBRP	Nominal Bit Rate Prescaler The value by which the oscillator frequency is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid values for the Bit Rate Prescaler are 0 to 511. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used
[15:8] NTSEG1	Nominal Time Segment before Sample Point Valid values are 1 to 255. The actual interpretation by the hardware of this value is such that one more than the programmed value is used. $t_{BS1} = (NTSEG1 + 1) \times t_q$.
[7]	Reserved.
[6:0] NTSEG2	Nominal Time Segment after Sample Point 0x01-0x7F Valid values are 1 to 127. The actual interpretation by the hardware of this value is such that one more than the programmed value is used. $t_{BS2} = (NTSEG2 + 1) \times t_q$. Note: With a CAN Core clock (cclk) of 8 MHz, the reset value of 0x06000A03 configures the controller for a bit rate of 500 kBit/s.

CANFD Timestamp Counter Configuration Register (CANFD_TSCC)

Register	Offset	R/W	Description	Reset Value
CANFD_TSCC	CANFDx_BA+0x020	R	Timestamp Counter Configuration (P*)	0x0000_0000

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

Bits	Description
[31:20]	Reserved Reserved.
[19:16]	TCP Timestamp Counter Prescaler Configures the timestamp and timeout counters time unit in multiples of CAN bit times [1...16]. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.
[15:2]	Reserved Reserved.
[1:0]	TSS Timestamp Select 00 = Timestamp counter value always 0x0000. 01 = Timestamp counter value incremented according to TCP. 10 = Reserved 11 = Same as "00".

CANFD Timestamp Counter Register (CANFD_TSCV)

Register	Offset	R/W	Description	Reset Value
CANFD_TSCV	CANFDx_BA+0x024	R	Timestamp Counter Value (C*)	0x0000_0000

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

Bits	Description
[31:16]	Reserved Reserved.
[15:0]	TSC Timestamp Counter The internal Timestamp Counter value is captured on start of frame (both Rx and Tx). When CANFD_TSS (TSCC[[1:0]] = 2'b01, the Timestamp Counter is incremented in multiples of CAN bit times [1...16] depending on the configuration of CANFD_TCP (CANFD_TSCC[19:16]). A wrap around sets interrupt flag CANFD_IR (CANFD_IR[16]). Write access resets the counter to 0. Note: A "wrap around" is a change of the Timestamp Counter value from non-zero to 0 not caused by write access to CANFD_TSCV.

CANFD Timeout Counter Configuration Register (CANFD TOCC)

Register	Offset	R/W	Description	Reset Value
CANFD_TOCC	CANFDx_BA+0x028	R	Timeout Counter Configuration (P*)	0xFFFF_0000

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

Bits	Description
[31:16]	TOP Timeout Period Start value of the Timeout Counter (down-counter). Configures the Timeout Period.
[15:3]	Reserved Reserved.
[2:1]	TOS Timeout Select When operating in Continuous mode, a write to CANFD_TOCV presets the counter to the value configured by CANFD_TOP (TOCC[31:16]) and continues down-counting. When the Timeout Counter is controlled by one of the FIFOs, an empty FIFO presets the counter to the value configured by CANFD_TOP (TOCC[31:16]). Down-counting is started when the first FIFO element is stored. 00 = Continuous operation. 01 = Timeout controlled by Tx Event FIFO. 10 = Timeout controlled by Rx FIFO 0. 11 = Timeout controlled by Rx FIFO 1.
[0]	ETOC Enable Timeout Counter 0 = Timeout Counter Disabled. 1 = Timeout Counter Enabled. Note: For use of timeout function with CAN FD, refer to Timeout Counter section.

CANFD Timeout Counter Value Register (CANFD_TOCV)

Register	Offset	R/W	Description	Reset Value
CANFD_TOCV	CANFDx_BA+0x02C	R	Timeout Counter Value (C*)	0x0000_FFFF

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

Bits	Description
[31:16]	Reserved Reserved.
[15:0]	TOC Timeout Counter The field is decremented in multiples of CAN bit times [1...16] depending on the configuration of TCP (CANFD_TSCC[19:16]). When decremented to 0, interrupt flag TOO (CANFD_IR[18]) is set and the timeout counter is stopped. Start and reset/restart conditions are configured via TOS (CANFD_TOCC[1:0]).

CANFD Error Counter Register (CANFD_ECR)

Register	Offset	R/W	Description	Reset Value
CANFD_ECR	CANFDx_BA+0x040	R	Error Counter Register (X*)	0x0000_0000

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

Bits	Description
[31:24]	Reserved Reserved.
[23:16]	CEL CAN Error Logging The counter is incremented each time when a CAN protocol error causes the 8-bit Transmit Error Counter TEC or the 7-bit Receive Error Counter REC to be incremented. The counter is also incremented when the Bus_Off limit is reached. It is not incremented when only RP is set without changing REC. The increment of CEL follows after the increment of REC or TEC. The counter is reset by read access to CEL. The counter stops at 0xFF; the next increment of TEC or REC sets interrupt flag ELO (CANFD_IR[22]).
[15]	RP Receive Error Passive 0 = The Receive Error Counter is below the error passive level of 128. 1 = The Receive Error Counter has reached the error passive level of 128.
[14:8]	REC Receive Error Counter Actual state of the Receive Error Counter, values between 0 and 127.
[7:0]	TEC Transmit Error Counter Actual state of the Transmit Error Counter, values between 0 and 255. Note: When ASM (CANFD_CCCR[2]) is set, the CAN protocol controller does not increment TEC and REC when a CAN protocol error is detected, but CEL is still incremented.

CANFD Protocol Status Register (CANFD_PSR)

Register	Offset	R/W	Description	Reset Value
CANFD_PSR	CANFDx_BA+0x044	R	Protocol Status Register (XS*)	0x0000_0707

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved	TDCV						
15	14	13	12	11	10	9	8
Reserved	PXE	RFDF	RBRS	RESI	DLEC		
7	6	5	4	3	2	1	0
BO	EW	EP	ACT		LEC		

Bits	Description
[31:23]	Reserved Reserved.
[22:16]	TDCV Transmitter Delay Compensation Value Position of the secondary sample point, defined by the sum of the measured delay from CANx_TXD to CANx_RXD and TDCO (TDCR[[14:8]]). The SSP position is, in the data phase, the number of minimum time quata (mtq) between the start of the transmitted bit and the secondary sample point. Valid values are 0 to 127 mtq.
[15]	Reserved Reserved.
[14]	PXE Protocol Exception Event 0 = No protocol exception event occurred since last read access. 1 = Protocol exception event occurred.
[13]	RFDF Received a CAN FD Message This bit is set independent of acceptance filtering. 0 = Since this bit was reset by the CPU, no CAN FD message has been received. 1 = Message in CAN FD format with FDF flag set has been received. Note: Byte access: Reading byte 0 will reset RFDF, reading bytes 3/2/1 has no impact.
[12]	RBRS BRS flag of last received CAN FD Message This bit is set together with RFDF, independent of acceptance filtering. 0 = Last received CAN FD message did not have its BRS flag set. 1 = Last received CAN FD message had its BRS flag set. Note: Byte access: Reading byte 0 will reset RBRS, reading bytes 3/2/1 has no impact.
[11]	RESI ESI flag of last received CAN FD Message This bit is set together with RFDF, independent of acceptance filtering. 0 = Last received CAN FD message did not have its ESI flag set. 1 = Last received CAN FD message had its ESI flag set.
[10:8]	DLEC Data Phase Last Error Code Type of last error that occurred in the data phase of a CAN FD format frame with its BRS flag set. Coding is the same as for LEC. This field will be cleared to 0 when a CAN FD format frame with its BRS flag set has been transferred (reception or transmission) without error.

[7]	BO	Bus_Off Status 0 = The CAN FD controller is not Bus_Off. 1 = The CAN FD controller is in Bus_Off state.
[6]	EW	Warning Status 0 = Both error counters are below the Error_Warning limit of 96. 1 = At least one of error counter has reached the Error_Warning limit of 96.
[5]	EP	Error Passive 0 = The CAN FD controller is in the Error_Active state. It normally takes part in bus communication and sends an active error flag when an error has been detected. 1 = The CAN FD controller is in the Error_Passive state.
[4:3]	ACT	Activity Monitors the module's CAN communication state. 00 = Synchronizing - node is synchronizing on CAN communication. 01 = Idle - node is neither receiver nor transmitter. 10 = Receiver - node is operating as receiver. 11 = Transmitter - node is operating as transmitter.
[2:0]	LEC	Last Error Code The LEC indicates the type of the last error to occur on the CAN bus. This field will be cleared to 0 when a message has been transferred (reception or transmission) without error. 000 = No Error: No error occurred since LEC has been reset by successful reception or transmission. 001 = Stuff Error: More than 5 equal bits in a sequence have occurred in a part of a received message where this is not allowed. 010 = Form Error: A fixed format part of a received frame has the wrong format. 011 = AckError: The message transmitted by the CANFD CONTROLLER was not acknowledged by another node. 100 = Bit1Error: During the transmission of a message (with the exception of the arbitration field), the device wanted to send a recessive level (bit of logical value 1), but the monitored bus value was dominant. 101 = Bit0Error : During the transmission of a message (or acknowledge bit, or active error flag, or overload flag), the device wanted to send a dominant level (data or identifier bit logical value 0), but the monitored bus value was recessive. During Bus_Off recovery this status is set each time a sequence of 11 recessive bits has been monitored. This enables the CPU to monitor the proceeding of the Bus_Off recovery sequence (indicating the bus is not stuck at dominant or continuously disturbed). 110 = CRCErrror: The CRC check sum of a received message was incorrect. The CRC of an incoming message does not match with the CRC calculated from the received data. 111 = NoChange: Any read access to the Protocol Status Register re-initializes the LEC to 7. When the LEC shows the value 7, no CAN bus event was detected since the last CPU read access to the Protocol Status Register.

Note: When a frame in CAN FD format has reached the data phase with BRS flag set, the next CAN event (error or valid frame) will be shown in DLEC instead of LEC. An error in a fixed stuff bit of a CAN FD CRC sequence will be shown as a Form Error, not Stuff Error.

Note: The Bus_Off recovery sequence (refer to ISO 11898-1:2015) cannot be shortened by setting or resetting CANFD_INIT (CANFD_CCCR[0]). If the device goes Bus_Off, it will set CANFD_INIT (CANFD_CCCR[0]) of its own accord, stopping all bus activities. Once CANFD_INIT (CANFD_CCCR[0]) has been cleared by the CPU, the device will then wait for 129 occurrences of Bus Idle (129 * 11 consecutive recessive bits) before resuming normal operation. At the end of the Bus_Off recovery sequence, the Error Management Counters will be reset. During the waiting time after the resetting of CANFD_INIT (CANFD_CCCR[0]), each time a sequence of 11 recessive bits has been monitored, a Bit0Error code is written to CANFD_LEC (CANFD_PSR[2:0]), enabling the CPU to readily check up whether the CAN bus is stuck at dominant or continuously disturbed and to monitor the Bus_Off recovery sequence. CANFD_ECR.REC is used to count these sequences.

CANFD Transmitter Delay Compensation Register (CANFD_TDCR)

Register	Offset	R/W	Description	Reset Value
CANFD_TDCR	CANFDx_BA+0x048	R	Transmitter Delay Compensation Register (P*)	0x0000_0000

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

Bits	Description
[31:15]	Reserved Reserved.
[14:8]	TDCO Transmitter Delay Compensation SSP Offset Offset value defining the distance between the measured delay from CANx_TXD to CANx_RXD and the secondary sample point. Valid values are 0 to 127 mtq.
[7]	Reserved Reserved.
[6:0]	TDCF Transmitter Delay Compensation Filter Window Length Defines the minimum value for the SSP position, dominant edges on CANx_RXD that would result in an earlier SSP position are ignored for transmitter delay measurement. The feature is enabled when TDCF is configured to a value greater than TDCO. Valid values are 0 to 127 mtq.

CANFD Interrupt Register (CANFD_IR)

The flags are set when one of the listed conditions is detected (edge-sensitive). A flag is cleared by writing a 1 to the corresponding bit position. Writing a 0 has no effect. A hard reset will clear the register.

Register	Offset	R/W	Description	Reset Value
CANFD_IR	CANFDx_BA+0x050	R/W	Interrupt Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		ARA	PED	PEA	WDI	BO	EW
23	22	21	20	19	18	17	16
EP	ELO	Reserved		DRX	TOO	MRAF	TSW
15	14	13	12	11	10	9	8
TEFL	TEFF	TEFW	TEFN	TFE	TCF	TC	HPM
7	6	5	4	3	2	1	0
RF1L	RF1F	RF1W	RF1N	RF0L	RF0F	RF0W	RF0N

Bits	Description
[31:30]	Reserved Reserved.
[29]	ARA Access to Reserved Address 0 = No access to reserved address occurred. 1 = Access to reserved address occurred.
[28]	PED Protocol Error in Data Phase 0 = No protocol error in data phase. 1 = Protocol error in data phase detected (DLEC (CANFD_PSR[10:8]) no equal 0 or 7). Note: Data bit time is used.
[27]	PEA Protocol Error in Arbitration Phase 0 = No protocol error in arbitration phase. 1 = Protocol error in arbitration phase detected (CANFD_LEC (CANFD_PSR[2:0]) no equal 0 or 7). Note: Nominal bit time is used.
[26]	WDI Watchdog Interrupt 0 = No Message RAM Watchdog event occurred. 1 = Message RAM Watchdog event due to missing READY.
[25]	BO Bus_Off Status 0 = Bus_Off status unchanged. 1 = Bus_Off status changed.
[24]	EW Warning Status 0 = Error_Warning status unchanged. 1 = Error_Warning status changed.
[23]	EP Error Passive 0 = Error_Passive status unchanged. 1 = Error_Passive status changed.

[22]	ELO	Error Logging Overflow 0 = CAN Error Logging Counter did not overflow. 1 = Overflow of CAN Error Logging Counter occurred.
[21:20]	Reserved	Reserved.
[19]	DRX	Message stored to Dedicated Rx Buffer The flag is set whenever a received message has been stored into a dedicated Rx Buffer. 0 = No Rx Buffer updated. 1 = At least one received message stored into an Rx Buffer.
[18]	TOO	Timeout Occurred 0 = No timeout. 1 = Timeout reached.
[17]	MRAF	Message RAM Access Failure The flag is set, when the Rx Handler - Has not completed acceptance filtering or storage of an accepted message until the arbitration field of the following message has been received. In this case acceptance filtering or message storage is aborted and the Rx Handler starts processing of the following message. - Was not able to write a message to the Message RAM. In this case message storage is aborted. In both cases the FIFO put index is not updated resp. the New Data flag for a dedicated Rx Buffer is not set, a partly stored message is overwritten when the next message is stored to this location. The flag is also set when the Tx Handler was not able to read a message from the Message RAM in time. In this case message transmission is aborted. In case of a Tx Handler access failure the CAN FD controller is switched into Restricted Operation Mode (refer to Restricted Operation Mode). To leave Restricted Operation Mode, the Host CPU has to reset CANFD_ASM (CANFD_CCCR[2]). 0 = No Message RAM access failure occurred. 1 = Message RAM access failure occurred.
[16]	TSW	Timestamp Wraparound 0 = No timestamp counter wrap-around. 1 = Timestamp counter wrapped around.
[15]	TEFL	Tx Event FIFO Element Lost 0 = No Tx Event FIFO element lost. 1 = Tx Event FIFO element lost, also set after write attempt to Tx Event FIFO of size zero.
[14]	TEFF	Tx Event FIFO Full 0 = Tx Event FIFO not full. 1 = Tx Event FIFO full.
[13]	TEFW	Tx Event FIFO Watermark Reached 0 = Tx Event FIFO fill level below watermark. 1 = Tx Event FIFO fill level reached watermark.
[12]	TEFN	Tx Event FIFO New Entry 0 = Tx Event FIFO unchanged. 1 = Tx Handler wrote Tx Event FIFO element.
[11]	TFE	Tx FIFO Empty 0 = Tx FIFO non-empty. 1 = Tx FIFO empty.
[10]	TCF	Transmission Cancellation Finished 0 = No transmission cancellation finished. 1 = Transmission cancellation finished.

[9]	TC	Transmission Completed 0 = No transmission completed. 1 = Transmission completed.
[8]	HPM	High Priority Message 0 = No high priority message received. 1 = High priority message received.
[7]	RF1L	Rx FIFO 1 Message Lost 0 = No Rx FIFO 1 message lost. 1 = Rx FIFO 1 message lost, also set after write attempt to Rx FIFO 1 of size zero.
[6]	RF1F	Rx FIFO 1 Full 0 = Rx FIFO 1 not full. 1 = Rx FIFO 1 full.
[5]	RF1W	Rx FIFO 1 Watermark Reached 0 = Rx FIFO 1 fill level below watermark. 1 = Rx FIFO 1 fill level reached watermark.
[4]	RF1N	Rx FIFO 1 New Message 0 = No new message written to Rx FIFO 1. 1 = New message written to Rx FIFO 1.
[3]	RF0L	Rx FIFO 0 Message Lost 0 = No Rx FIFO 0 message lost. 1 = Rx FIFO 0 message lost, also set after write attempt to Rx FIFO 0 of size zero.
[2]	RF0F	Rx FIFO 0 Full 0 = Rx FIFO 0 not full. 1 = Rx FIFO 0 full.
[1]	RF0W	Rx FIFO 0 Watermark Reached 0 = Rx FIFO 0 fill level below watermark. 1 = Rx FIFO 0 fill level reached watermark.
[0]	RF0N	Rx FIFO 0 New Message 0 = No new message written to Rx FIFO 0. 1 = New message written to Rx FIFO 0.

CANFD Interrupt Enable Register (CANFD IE)

Register	Offset	R/W	Description	Reset Value
CANFD_IE	CANFDx_BA+0x054	R/W	Interrupt Enable	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		ARAE	PEDE	PEAE	WDIE	BOE	EWE
23	22	21	20	19	18	17	16
EPE	ELOE	BEUE	BECE	DRXE	TOOE	MRAFE	TSWE
15	14	13	12	11	10	9	8
TEFLE	TEFFE	TEFWE	TEFNE	TFEE	TCFE	TCE	HPME
7	6	5	4	3	2	1	0
RF1LE	RF1FE	RF1WE	RF1NE	RF0LE	RF0FE	RF0WE	RF0NE

Bits	Description
[31:30]	Reserved Reserved.
[29]	ARAE Access to Reserved Address Enable 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[28]	PEDE Protocol Error in Data Phase Enable 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[27]	PEAE Protocol Error in Arbitration Phase Enable 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[26]	WDIE Watchdog Interrupt Enable 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[25]	BOE Bus_Off Status Interrupt Enable 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[24]	EWE Warning Status Interrupt Enable 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[23]	EPE Error Passive Interrupt Enable 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[22]	ELOE Error Logging Overflow Interrupt Enable 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[21]	BEUE Bit Error Uncorrected Interrupt Enable

		0 = Interrupt Disabled. 1 = Interrupt Enabled.
[20]	BECE	Bit Error Corrected Interrupt Enable 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[19]	DRXE	Message stored to Dedicated Rx Buffer Interrupt Enable 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[18]	TOOE	Timeout Occurred Interrupt Enable 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[17]	MRAFE	Message RAM Access Failure Interrupt Enable 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[16]	TSWE	Timestamp Wraparound Interrupt Enable 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[15]	TEFLE	Tx Event FIFO Event Lost Interrupt Enable 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[14]	TEFFE	Tx Event FIFO Full Interrupt Enable 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[13]	TEFWE	Tx Event FIFO Watermark Reached Interrupt Enable 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[12]	TEFNE	Tx Event FIFO New Entry Interrupt Enable 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[11]	TFEE	Tx FIFO Empty Interrupt Enable 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[10]	TCFE	Transmission Cancellation Finished Interrupt Enable 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[9]	TCE	Transmission Completed Interrupt Enable 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[8]	HPME	High Priority Message Interrupt Enable 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[7]	RF1LE	Rx FIFO 1 Message Lost Interrupt Enable 0 = Interrupt Disabled. 1 = Interrupt Enabled.

[6]	RF1FE	Rx FIFO 1 Full Interrupt Enable 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[5]	RF1WE	Rx FIFO 1 Watermark Reached Interrupt Enable 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[4]	RF1NE	Rx FIFO 1 New Message Interrupt Enable 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[3]	RF0LE	Rx FIFO 0 Message Lost Interrupt Enable 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[2]	RF0FE	Rx FIFO 0 Full Interrupt Enable 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[1]	RF0WE	Rx FIFO 0 Watermark Reached Interrupt Enable 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[0]	RF0NE	Rx FIFO 0 New Message Interrupt Enable 0 = Interrupt Disabled. 1 = Interrupt Enabled.

CANFD Interrupt Line Select Register (CANFD_ILS)

This register assigns an interrupt generated by a specific interrupt flag from the CANFD_IR register to one of the two module interrupt lines.

Register	Offset	R/W	Description	Reset Value
CANFD_ILS	CANFDx_BA+0x058	R/W	Interrupt Line Select	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		ARAL	PEDL	PEAL	WDIL	BOL	EWL
23	22	21	20	19	18	17	16
EPL	ELOL	Reserved		DRXL	TOOL	MRAFL	TSWL
15	14	13	12	11	10	9	8
TEFLL	TEFFL	TEFWL	TEFNL	TFEL	TCFL	TCL	HPML
7	6	5	4	3	2	1	0
RF1LL	RF1FL	RF1WL	RF1NL	RF0LL	RF0FL	RF0WL	RF0NL

Bits	Description	
[31:30]	Reserved	Reserved.
[29]	ARAL	Access to Reserved Address Line 0 = Interrupt assigned to CAN interrupt line 0. 1 = Interrupt assigned to CAN interrupt line 1.
[28]	PEDL	Protocol Error in Data Phase Line 0 = Interrupt assigned to CAN interrupt line 0. 1 = Interrupt assigned to CAN interrupt line 1.
[27]	PEAL	Protocol Error in Arbitration Phase Line 0 = Interrupt assigned to CAN interrupt line 0. 1 = Interrupt assigned to CAN interrupt line 1.
[26]	WDIL	Watchdog Interrupt Line 0 = Interrupt assigned to CAN interrupt line 0. 1 = Interrupt assigned to CAN interrupt line 1.
[25]	BOL	Bus_Off Status Interrupt Line 0 = Interrupt assigned to CAN interrupt line 0. 1 = Interrupt assigned to CAN interrupt line 1.
[24]	EWL	Warning Status Interrupt Line 0 = Interrupt assigned to CAN interrupt line 0. 1 = Interrupt assigned to CAN interrupt line 1.
[23]	EPL	Error Passive Interrupt Line 0 = Interrupt assigned to CAN interrupt line 0. 1 = Interrupt assigned to CAN interrupt line 1.
[22]	ELOL	Error Logging Overflow Interrupt Line 0 = Interrupt assigned to CAN interrupt line 0.

		1 = Interrupt assigned to CAN interrupt line 1.
[21:20]	Reserved	Reserved.
[19]	DRXL	Message stored to Dedicated Rx Buffer Interrupt Line 0 = Interrupt assigned to CAN interrupt line 0. 1 = Interrupt assigned to CAN interrupt line 1.
[18]	TOOL	Timeout Occurred Interrupt Line 0 = Interrupt assigned to CAN interrupt line 0. 1 = Interrupt assigned to CAN interrupt line 1.
[17]	MRAFL	Message RAM Access Failure Interrupt Line 0 = Interrupt assigned to CAN interrupt line 0. 1 = Interrupt assigned to CAN interrupt line 1.
[16]	TSWL	Timestamp Wraparound Interrupt Line 0 = Interrupt assigned to CAN interrupt line 0. 1 = Interrupt assigned to CAN interrupt line 1.
[15]	TEFLL	Tx Event FIFO Event Lost Interrupt Line 0 = Interrupt assigned to CAN interrupt line 0. 1 = Interrupt assigned to CAN interrupt line 1.
[14]	TEFFL	Tx Event FIFO Full Interrupt Line 0 = Interrupt assigned to CAN interrupt line 0. 1 = Interrupt assigned to CAN interrupt line 1.
[13]	TEFWL	Tx Event FIFO Watermark Reached Interrupt Line 0 = Interrupt assigned to CAN interrupt line 0. 1 = Interrupt assigned to CAN interrupt line 1.
[12]	TEFNL	Tx Event FIFO New Entry Interrupt Line 0 = Interrupt assigned to CAN interrupt line 0. 1 = Interrupt assigned to CAN interrupt line 1.
[11]	TFEL	Tx FIFO Empty Interrupt Line 0 = Interrupt assigned to CAN interrupt line 0. 1 = Interrupt assigned to CAN interrupt line 1.
[10]	TCFL	Transmission Cancellation Finished Interrupt Line 0 = Interrupt assigned to CAN interrupt line 0. 1 = Interrupt assigned to CAN interrupt line 1.
[9]	TCL	Transmission Completed Interrupt Line 0 = Interrupt assigned to CAN interrupt line 0. 1 = Interrupt assigned to CAN interrupt line 1.
[8]	HPML	High Priority Message Interrupt Line 0 = Interrupt assigned to CAN interrupt line 0. 1 = Interrupt assigned to CAN interrupt line 1.
[7]	RF1LL	Rx FIFO 1 Message Lost Interrupt Line 0 = Interrupt assigned to CAN interrupt line 0. 1 = Interrupt assigned to CAN interrupt line 1.
[6]	RF1FL	Rx FIFO 1 Full Interrupt Line 0 = Interrupt assigned to CAN interrupt line 0.

		1 = Interrupt assigned to CAN interrupt line 1.
[5]	RF1WL	Rx FIFO 1 Watermark Reached Interrupt Line 0 = Interrupt assigned to CAN interrupt line 0. 1 = Interrupt assigned to CAN interrupt line 1.
[4]	RF1NL	Rx FIFO 1 New Message Interrupt Line 0 = Interrupt assigned to CAN interrupt line 0. 1 = Interrupt assigned to CAN interrupt line 1.
[3]	RF0LL	Rx FIFO 0 Message Lost Interrupt Line 0 = Interrupt assigned to CAN interrupt line 0. 1 = Interrupt assigned to CAN interrupt line 1.
[2]	RF0FL	Rx FIFO 0 Full Interrupt Line 0 = Interrupt assigned to CAN interrupt line 0. 1 = Interrupt assigned to CAN interrupt line 1.
[1]	RF0WL	Rx FIFO 0 Watermark Reached Interrupt Line 0 = Interrupt assigned to CAN interrupt line 0. 1 = Interrupt assigned to CAN interrupt line 1.
[0]	RF0NL	Rx FIFO 0 New Message Interrupt Line 0 = Interrupt assigned to CAN interrupt line 0. 1 = Interrupt assigned to CAN interrupt line 1.

CANFD Interrupt Line Register (CANFD_ILE)

Register	Offset	R/W	Description	Reset Value
CANFD_ILE	CANFDx_BA+0x05C	R/W	Interrupt Line Enable	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved						ENT1	ENT0

Bits	Description	
[31:2]	Reserved	Reserved.
[1]	ENT1	Enable Interrupt Line 1 0 = Interrupt line canfd_int1 Disabled. 1 = Interrupt line canfd_int1 Enabled.
[0]	ENT0	Enable Interrupt Line 0 0 = Interrupt line canfd_int0 Disabled. 1 = Interrupt line canfd_int0 Enabled.

CANFD Global Filter Configuration Register (CANFD_GFC)

Register	Offset	R/W	Description	Reset Value
CANFD_GFC	CANFDx_BA+0x080	R	Global Filter Configuration (P*)	0x0000_0000

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

Bits	Description
[31:6]	Reserved Reserved.
[5:4]	ANFS Accept Non-matching Frames Standard Defines how received messages with 11-bit IDs that do not match any element of the filter list are treated. 00 = Accept in Rx FIFO 0. 01 = Accept in Rx FIFO 1. 10 = Reject. 11 = Reject.
[3:2]	ANFE Accept Non-matching Frames Extended Defines how received messages with 29-bit IDs that do not match any element of the filter list are treated. 00 = Accept in Rx FIFO 0. 01 = Accept in Rx FIFO 1. 10 = Reject. 11 = Reject.
[1]	RRFS Reject Remote Frames Standard 0 = Filter remote frames with 11-bit standard IDs. 1 = Reject all remote frames with 11-bit standard IDs.
[0]	RRFE Reject Remote Frames Extended 0 = Filter remote frames with 29-bit extended IDs. 1 = Reject all remote frames with 29-bit extended IDs.

CANFD Standard ID Filter Configuration Register (CANFD_SIDFC)

Register	Offset	R/W	Description	Reset Value
CANFD_SIDFC	CANFDx_BA+0x084	R	Standard ID Filter Configuration (P*)	0x0000_0000

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

Bits	Description	
[31:24]	Reserved	Reserved.
[23:16]	LSS	List Size Standard 0 = No standard Message ID filter. 1-128 = Number of standard Message ID filter elements. >128 = Values greater than 128 are interpreted as 128.
[15:2]	FLSSA	Filter List Standard Start Address Start address of standard Message ID filter list (32-bit word address, refer to Figure 6.20-11).
[1:0]	Reserved	Reserved.

CANFD Extended ID Filter Configuration Register (CANFD_XIDFC)

Register	Offset	R/W	Description	Reset Value
CANFD_XIDFC	CANFDx_BA+0x088	R	Extended ID Filter Configuration (P*)	0x0000_0000

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

Bits	Description
[31:23]	Reserved Reserved.
[22:16]	LSE List Size Extended 0 = No extended Message ID filter. 1-64 = Number of extended Message ID filter elements. >64 = Values greater than 64 are interpreted as 64.
[15:2]	FLESA Filter List Extended Start Address Start address of extended Message ID filter list (32-bit word address, refer to Figure 6.20-11).
[1:0]	Reserved Reserved.

CANFD Extended ID and Mask Register (CANFD_XIDAM)

Register	Offset	R/W	Description	Reset Value
CANFD_XIDAM	CANFDx_BA+0x090	R	Extended ID AND Mask (P*)	0x1FFF_FFFF

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

Bits	Description
[31:29]	Reserved Reserved.
[28:0]	EIDM Extended ID Mask For acceptance filtering of extended frames the Extended ID AND Mask is ANDed with the Message ID of a received frame. Intended for masking of 29-bit IDs in SAE J1939. With the reset value of all bits set to one the mask is not active.

CANFD High Priority Message Status Register (CANFD_HPMS)

Register	Offset	R/W	Description	Reset Value
CANFD_HPMS	CANFDx_BA+0x094	R	High Priority Message Status	0x0000_0000

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

Bits	Description
[31:16]	Reserved Reserved.
[15]	FLST Filter List Indicates the filter list of the matching filter element. 0 = Standard Filter List. 1 = Extended Filter List.
[14:8]	FIDX Filter Index Index of matching filter element. Range is 0 to CANFD_SIDFC.LSS - 1 or CANFD_XIDFC.LSE - 1
[7:6]	MSI Message Storage Indicator 00 = No FIFO selected. 01 = FIFO message lost. 10 = Message stored in FIFO 0. 11 = Message stored in FIFO 1.
[5:0]	BIDX Buffer Index Index of Rx FIFO element to which the message was stored. Only valid when MSI[1] = 1.

CANFD New Data 1 Register (CANFD_NDAT1)

Register	Offset	R/W	Description	Reset Value
CANFD_NDAT1	CANFDx_BA+0x098	R/W	New Data 1	0x0000_0000

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

Bits	Description
[31:0] n=0,1..31	New Data The register holds the New Data flags of Rx Buffers 0 to 31. The flags are set when the respective Rx Buffer has been updated from a received frame. The flags remain set until the Host clears them. A flag is cleared by writing a 1 to the corresponding bit position. Writing a 0 has no effect. A hard reset will clear the register. 0 = Rx Buffer not updated. 1 = Rx Buffer updated from new message.

CANFD New Data 2 Register (CANFD_NDAT2)

Register	Offset	R/W	Description	Reset Value
CANFD_NDAT2	CANFDx_BA+0x09C	R/W	New Data 2	0x0000_0000

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

Bits	Description
[31:0] n=32,33..63	New Data The register holds the New Data flags of Rx Buffers 32 to 63. The flags are set when the respective Rx Buffer has been updated from a received frame. The flags remain set until the Host clears them. A flag is cleared by writing a 1 to the corresponding bit position. Writing a 0 has no effect. A hard reset will clear the register. 0 = Rx Buffer not updated. 1 = Rx Buffer updated from new message.

CANFD Rx FIFO 0 Configuration Register (CANFD_RXF0C)

Register	Offset	R/W	Description	Reset Value
CANFD_RXF0C	CANFDx_BA+0x0A0	R	Rx FIFO 0 Configuration (P*)	0x0000_0000

31	30	29	28	27	26	25	24
F0OM	F0WM						
23	22	21	20	19	18	17	16
Reserved	F0S						
15	14	13	12	11	10	9	8
F0SA							
7	6	5	4	3	2	1	0
F0SA						Reserved	

Bits	Description	
[31]	F0OM	FIFO 0 Operation Mode FIFO 0 can be operated in blocking or in overwrite mode (refer to Rx FIFOs). 0 = FIFO 0 blocking mode. 1 = FIFO 0 overwrite mode.
[30:24]	F0WM	Rx FIFO 0 Watermark 0 = Watermark interrupt Disabled. 1-64 = Level for Rx FIFO 0 watermark interrupt (CANFD_IR.RF0W). >64 = Watermark interrupt Disabled.
[23]	Reserved	Reserved.
[22:16]	F0S	Rx FIFO 0 Size 0 = No Rx FIFO 0. 1-64 = Number of Rx FIFO 0 elements. >64 = Values greater than 64 are interpreted as 64. The Rx FIFO 0 elements are indexed from 0 to F0S-1
[15:2]	F0SA	Rx FIFO 0 Start Address Start address of Rx FIFO 0 in Message RAM (32-bit word address).
[1:0]	Reserved	Reserved.

CANFD Rx FIFO 0 Status Register (CANFD_RXF0S)

Register	Offset	R/W	Description	Reset Value
CANFD_RXF0S	CANFDx_BA+0x0A4	R	Rx FIFO 0 Status	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						RF0L	F0F
23	22	21	20	19	18	17	16
Reserved		F0PI					
15	14	13	12	11	10	9	8
Reserved		F0GI					
7	6	5	4	3	2	1	0
Reserved	F0FL						

Bits	Description
[31:26]	Reserved Reserved.
[25]	RF0L Rx FIFO 0 Message Lost This bit is a copy of interrupt flag CANFD_IR.RF0L. When CANFD_IR.RF0L is reset, this bit is also reset. 0 = No Rx FIFO 0 message lost. 1 = Rx FIFO 0 message lost, also set after write attempt to Rx FIFO 0 of size zero. Note: Overwriting the oldest message when F0OM (CANFD_RXF0C[31]) = 1 will not set this flag.
[24]	F0F Rx FIFO 0 Full 0 = Rx FIFO 0 not full. 1 = Rx FIFO 0 full.
[23:22]	Reserved Reserved.
[21:16]	F0PI Rx FIFO 0 Put Index Rx FIFO 0 write index pointer, range 0 to 63.
[15:14]	Reserved Reserved.
[13:8]	F0GI Rx FIFO 0 Get Index Rx FIFO 0 read index pointer, range 0 to 63.
[7]	Reserved Reserved.
[6:0]	F0FL Rx FIFO 0 Fill Level Number of elements stored in Rx FIFO 0, range 0 to 64

CANFD Rx FIFO 0 Acknowledge Register (CANFD_RXF0A)

Register	Offset	R/W	Description	Reset Value
CANFD_RXF0A	CANFDx_BA+0x0A8	R/W	Rx FIFO 0 Acknowledge	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		F0A					

Bits	Description
[31:6]	Reserved Reserved.
[5:0]	F0A Rx FIFO 0 Acknowledge Index After the Host has read a message or a sequence of messages from Rx FIFO 0 it has to write the buffer index of the last element read from Rx FIFO 0 to F0AI. This will set the Rx FIFO 0 Get Index F0GI (CANFD_RXF0S[13:8]) to F0AI (CANFD_RXF0A[5:0]) + 1 and update the FIFO 0 Fill Level CANFD_RXF0S.F0FL.

CANFD Rx Buffer Configuration Register (CANFD_RXBC)

Register	Offset	R/W	Description	Reset Value
CANFD_RXBC	CANFDx_BA+0x0AC	R	Rx Buffer Configuration (P*)	0x0000_0000

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

Bits	Description
[31:16]	Reserved Reserved.
[15:2]	RBSA Rx Buffer Start Address Configures the start address of the Rx Buffers section in the Message RAM (32-bit word address).
[1:0]	Reserved Reserved.

CANFD Rx FIFO 1 Configuration Register (CANFD_RXF1C)

Register	Offset	R/W	Description	Reset Value
CANFD_RXF1C	CANFDx_BA+0x0B0	R	Rx FIFO 1 Configuration (P*)	0x0000_0000

31	30	29	28	27	26	25	24
F1OM	F1WM						
23	22	21	20	19	18	17	16
Reserved	F1S						
15	14	13	12	11	10	9	8
F1SA							
7	6	5	4	3	2	1	0
F1SA						Reserved	

Bits	Description
[31]	F1OM FIFO 1 Operation Mode FIFO 1 can be operated in blocking or in overwrite mode (refer to Rx FIFOs). 0 = FIFO 1 blocking mode. 1 = FIFO 1 overwrite mode.
[30:24]	F1WM Rx FIFO 1 Watermark 0 = Watermark interrupt Disabled. 1-64 = Level for Rx FIFO 1 watermark interrupt (CANFD_IR.RF1W). >64 = Watermark interrupt Disabled.
[23]	Reserved Reserved.
[22:16]	F1S Rx FIFO 1 Size 0 = No Rx FIFO 1. 1-64 = Number of Rx FIFO 1 elements. >64 = Values greater than 64 are interpreted as 64. The Rx FIFO 1 elements are indexed from 0 to F1S - 1
[15:2]	F1SA Rx FIFO 1 Start Address Start address of Rx FIFO 1 in Message RAM (32-bit word address, refer to Figure 6.20-11).
[1:0]	Reserved Reserved.

CANFD Rx FIFO 1 Status Register (CANFD_RXF1S)

Register	Offset	R/W	Description	Reset Value
CANFD_RXF1S	CANFDx_BA+0x0B4	R	Rx FIFO 1 Status	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						RF1L	F1F
23	22	21	20	19	18	17	16
Reserved		F1P					
15	14	13	12	11	10	9	8
Reserved		F1G					
7	6	5	4	3	2	1	0
Reserved	F1FL						

Bits	Description
[31:26]	Reserved Reserved.
[25]	RF1L Rx FIFO 1 Message Lost This bit is a copy of interrupt flag CANFD_IR.RF1L. When CANFD_IR.RF1L is reset, this bit is also reset. 0 = No Rx FIFO 1 message lost. 1 = Rx FIFO 1 message lost, also set after write attempt to Rx FIFO 1 of size zero. Note: Overwriting the oldest message when F1OM (CANFD_RXF1C[31]) = 1 will not set this flag.
[24]	F1F Rx FIFO 1 Full 0 = Rx FIFO 1 not full. 1 = Rx FIFO 1 full.
[23:22]	Reserved Rx FIFO 1 Get Index Rx FIFO 1 read index pointer, range 0 to 63.
[21:16]	F1P Rx FIFO 1 Fill Level Number of elements stored in Rx FIFO 1, range 0 to 64.
[15:14]	Reserved Reserved.
[13:8]	F1G Rx FIFO 1 Get Index Rx FIFO 1 read index pointer, range 0 to 63.
[7]	Reserved Reserved.
[6:0]	F1FL Rx FIFO 1 Fill Level Number of elements stored in Rx FIFO 1, range 0 to 64

CANFD Rx FIFO 1 Acknowledge Register (CANFD_RXF1A)

Register	Offset	R/W	Description	Reset Value
CANFD_RXF1A	CANFDx_BA+0x0B8	R/W	Rx FIFO 1 Acknowledge	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		F1A					

Bits	Description
[31:6]	Reserved Reserved.
[5:0]	F1A Rx FIFO 1 Acknowledge Index After the Host has read a message or a sequence of messages from Rx FIFO 1 it has to write the buffer index of the last element read from Rx FIFO 1 to F1AI. This will set the Rx FIFO 1 Get Index F1GI (CANFD_RXF1S[13:8]) to F1AI (CANFD_RXF1A[5:0]) + 1 and update the FIFO 1 Fill Level F1FL (CANFD_RXF1S[6:0]).

CANFD Rx Buffer / FIFO Element Size Configuration Register (CANFD_RXESC)

Register	Offset	R/W	Description	Reset Value
CANFD_RXESC	CANFDx_BA+0x0BC	R	Rx Buffer / FIFO Element Size Configuration (P*)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved					RBDS		
7	6	5	4	3	2	1	0
Reserved	F1DS			Reserved	F0DS		

Bits	Description
[31:11]	Reserved Reserved.
[10:8]	RBDS Rx Buffer Data Field Size 000 = 8 byte data field. 001 = 12 byte data field. 010 = 16 byte data field. 011 = 20 byte data field. 100 = 24 byte data field. 101 = 32 byte data field. 110 = 48 byte data field. 111 = 64 byte data field.
[7]	Reserved Reserved.
[6:4]	F1DS Rx FIFO 1 Data Field Size 000 = 8 byte data field. 001 = 12 byte data field. 010 = 16 byte data field. 011 = 20 byte data field. 100 = 24 byte data field. 101 = 32 byte data field. 110 = 48 byte data field. 111 = 64 byte data field.
[3]	Reserved Reserved.
[2:0]	F0DS Rx FIFO 0 Data Field Size 000 = 8 byte data field. 001 = 12 byte data field. 010 = 16 byte data field. 011 = 20 byte data field. 100 = 24 byte data field.

		101 = 32 byte data field. 110 = 48 byte data field. 111 = 64 byte data field. Note: In case the data field size of an accepted CAN frame exceeds the data field size configured for the matching Rx Buffer or Rx FIFO, only the number of bytes as configured by CANFD_RXESC are stored to the Rx Buffer resp. Rx FIFO element. The rest of the frame data field is ignored.

CANFD Tx Buffer Configuration Register (CANFD_TXBC)

Register	Offset	R/W	Description	Reset Value
CANFD_TXBC	CANFDx_BA+0x0C0	R	Tx Buffer Configuration (P*)	0x0000_0000

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

Bits	Description
[31]	Reserved Reserved.
[30]	TFQM Tx FIFO/Queue Mode 0 = Tx FIFO operation. 1 = Tx Queue operation.
[29:24]	TFQS Transmit FIFO/Queue Size 0 = No Tx FIFO/Queue. 1-32 = Number of Tx Buffers used for Tx FIFO/Queue. >32 = Values greater than 32 are interpreted as 32.
[23:22]	Reserved Reserved.
[21:16]	NDTB Number of Dedicated Transmit Buffers 0 = No Dedicated Tx Buffers. 1-32 = Number of Dedicated Tx Buffers. >32 = Values greater than 32 are interpreted as 32.
[15:2]	TBSA Tx Buffers Start Address Start address of Tx Buffers section in Message RAM (32-bit word address, refer to Figure 6.20-11). Note: The sum of TFQS and NDTB may be not greater than 32. There is no check for erroneous configurations. The Tx Buffers section in the Message RAM starts with the dedicated Tx Buffers.
[1:0]	Reserved Reserved.

CANFD Tx FIFO / Queue Status Register (CANFD_TXFQS)

Register	Offset	R/W	Description	Reset Value
CANFD_TXFQS	CANFDx_BA+0x0C4	R	Tx FIFO/Queue Status	0x0000_0000

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

Bits	Description
[31:22]	Reserved Reserved.
[21]	TFQF Tx FIFO/Queue Full 0 = Tx FIFO/Queue not full. 1 = Tx FIFO/Queue full.
[20:16]	TFQP Tx FIFO/Queue Put Index Tx FIFO/Queue write index pointer, range 0 to 31.
[15:13]	Reserved Reserved.
[12:8]	TFG Tx FIFO Get Index Tx FIFO read index pointer, range 0 to 31. Read as 0 when Tx Queue operation is configured (TFQM (CANFD_TXBC[30]) = 1).
[7:6]	Reserved Reserved.
[5:0]	TFFL Tx FIFO Free Level Number of consecutive free Tx FIFO elements starting from TFGI, range 0 to 32. Read as 0 when Tx Queue operation is configured (TFQM (CANFD_TXBC[3]) = 1). Note: In case of mixed configurations where dedicated Tx Buffers are combined with a Tx FIFO or a Tx Queue, the Put and Get Indices indicate the number of the Tx Buffer starting with the first dedicated Tx Buffers. Example: For a configuration of 12 dedicated Tx Buffers and a Tx FIFO of 20 Buffers a Put Index of 15 points to the fourth buffer of the Tx FIFO.

CANFD Tx Buffer Element Size Configuration Register (CANFD_TXESC)

Register	Offset	R/W	Description	Reset Value
CANFD_TXESC	CANFDx_BA+0x0C8	R	Tx Buffer Element Size Configuration (P*)	0x0000_0000

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

Bits	Description
[31:3]	Reserved Reserved.
[2:0]	TBDS Tx Buffer Data Field Size 000 = 8 byte data field. 001 = 12 byte data field. 010 = 16 byte data field. 011 = 20 byte data field. 100 = 24 byte data field. 101 = 32 byte data field. 110 = 48 byte data field. 111 = 64 byte data field. Note: In case the data length code DLC of a Tx Buffer element is configured to a value higher than the Tx Buffer data field size CANFD_TXESC.TBDS, the bytes not defined by the Tx Buffer are transmitted as 0xCC (padding bytes).

CANFD Tx Buffer Request Pending Register (CANFD_TXBRP)

Register	Offset	R/W	Description	Reset Value
CANFD_TXBRP	CANFDx_BA+0x0CC	R	Tx Buffer Request Pending	0x0000_0000

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

Bits	Description
[31:0]	Transmission Request Pending Each Tx Buffer has its own Transmission Request Pending bit. The bits are set via register CANFD_TXBAR. The bits are reset after a requested transmission has completed or has been cancelled via register CANFD_TXBCR. CANFD_TXBRP bits are set only for those Tx Buffers configured via CANFD_TXBC. After a CANFD_TXBRP bit has been set, a Tx scan (refer to Tx Handling section) is started to check for the pending Tx request with the highest priority (Tx Buffer with lowest Message ID). A cancellation request resets the corresponding transmission request pending bit of register CANFD_TXBRP. In case a transmission has already been started when a cancellation is requested, this is done at the end of the transmission, regardless whether the transmission was successful or not. The cancellation request bits are reset directly after the corresponding CANFD_TXBRP bit has been reset. After a cancellation has been requested, a finished cancellation is signaled via CANFD_TXBCF. • after successful transmission together with the corresponding CANFD_TXBTO bit • when the transmission has not yet been started at the point of cancellation • when the transmission has been aborted due to lost arbitration • when an error occurred during frame transmission In DAR mode all transmissions are automatically cancelled if they are not successful. The corresponding CANFD_TXBCF bit is set for all unsuccessful transmissions. 0 = No transmission request pending. 1 = Transmission request pending. Note: CANFD_TXBRP bits which are set while a Tx scan is in progress are not considered during this particular Tx scan. In case a cancellation is requested for such a Tx Buffer, this Add Request is cancelled immediately, the corresponding CANFD_TXBRP bit is reset.

CANFD Tx Buffer Add Request Register (CANFD_TXBAR)

Register	Offset	R/W	Description	Reset Value
CANFD_TXBAR	CANFDx_BA+0x0D0	R/W	Tx Buffer Add Request	0x0000_0000

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

Bits	Description
[31:0]	ARn Add Request Each Tx Buffer has its own Add Request bit Writing a 1 will set the corresponding Add Request bit; writing a 0 has no impact This enables the Host to set transmission requests for multiple Tx Buffers with one write to CANFD_TXBAR CANFD_TXBAR bits are set only for those Tx Buffers configured via CANFD_TXBC When no Tx scan is running, the bits are reset immediately, else the bits remain set until the Tx scan process has completed. 0 = No transmission request added. 1 = Transmission requested added. Note: If an add request is applied for a Tx Buffer with pending transmission request (corresponding CANFD_TXBRP bit already set), this add request is ignored.

CANFD Tx Buffer Cancellation Request Register (CANFD_TXBCR)

Register	Offset	R/W	Description	Reset Value
CANFD_TXBCR	CANFDx_BA+0x0D4	R/W	Tx Buffer Cancellation Request	0x0000_0000

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

Bits	Description
[31:0]	Cancellation Request Each Tx Buffer has its own Cancellation Request bit. Writing a 1 will set the corresponding Cancellation Request bit; writing a 0 has no impact. This enables the Host to set cancellation requests for multiple Tx Buffers with one write to CANFD_TXBCR. CANFD_TXBCR bits are set only for those Tx Buffers configured via CANFD_TXBC. The bits remain set until the corresponding bit of CANFD_TXBRP is reset. 0 = No cancellation pending. 1 = Cancellation pending.

CANFD Tx Buffer Transmission Occurred Register (CANFD_TXBTO)

Register	Offset	R/W	Description	Reset Value
CANFD_TXBTO	CANFDx_BA+0x0D8	R	Tx Buffer Transmission Occurred	0x0000_0000

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

Bits	Description
[31:0]	Transmission Occurred Each Tx Buffer has its own Transmission Occurred bit. The bits are set when the corresponding CANFD_TXBRP bit is cleared after a successful transmission. The bits are reset when a new transmission is requested by writing a 1 to the corresponding bit of register CANFD_TXBAR. 0 = No transmission occurred. 1 = Transmission occurred.

CANFD Tx Buffer Cancellation Finished Register (CANFD_TXBCF)

Register	Offset	R/W	Description	Reset Value
CANFD_TXBCF	CANFDx_BA+0x0DC	R	Tx Buffer Cancellation Finished	0x0000_0000

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

Bits	Description
[31:0]	Cancellation Finished Each Tx Buffer has its own Cancellation Finished bit. The bits are set when the corresponding CANFD_TXBRP bit is cleared after a cancellation was requested via CANFD_TXBCR. In case the corresponding CANFD_TXBRP bit was not set at the point of cancellation, CF is set immediately. The bits are reset when a new transmission is requested by writing a 1 to the corresponding bit of register CANFD_TXBAR. 0 = No transmit buffer cancellation. 1 = Transmit buffer cancellation finished.

CANFD Tx Buffer Transmission Interrupt Enable Register (CANFD_TXBTIE)

Register	Offset	R/W	Description	Reset Value
CANFD_TXBTIE	CANFDx_BA+0x0E0	R/W	Tx Buffer Transmission Interrupt Enable	0x0000_0000

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

Bits	Description	
[31:0]	TIE_n	Transmission Interrupt Enable Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 = Transmission interrupt Disabled. 1 = Transmission interrupt Enabled.

CANFD Tx Buffer Cancellation Finished Interrupt Enable Register (CANFD_TXBCIE)

Register	Offset	R/W	Description	Reset Value
CANFD_TXBCIE	CANFDx_BA+0x0E4	R/W	Tx Buffer Cancellation Finished Interrupt Enable	0x0000_0000

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

Bits	Description	
[31:0]	CFIE _n	Cancellation Finished Interrupt Enable Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 = Cancellation finished interrupt Disabled. 1 = Cancellation finished interrupt Enabled.

CANFD Tx Event FIFO Configuration Register (CANFD_TXEFC)

Register	Offset	R/W	Description	Reset Value
CANFD_TXEFC	CANFDx_BA+0x0F0	R	Tx Event FIFO Configuration (P*)	0x0000_0000

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

Bits	Description
[31:30]	Reserved Reserved.
[29:24]	EFWN Event FIFO Watermark 0 = Watermark interrupt Disabled. 1-32 = Level for Tx Event FIFO watermark interrupt (TEFW (CANFD_IR[13])). >32 = Watermark interrupt Disabled.
[23:22]	Reserved Reserved.
[21:16]	EFS Event FIFO Size 0 = Tx Event FIFO Disabled. 1-32 = Number of Tx Event FIFO elements. >32 = Values greater than 32 are interpreted as 32. The Tx Event FIFO elements are indexed from 0 to EFS - 1
[15:2]	EFSA Event FIFO Start Address Start address of Tx Event FIFO in Message RAM (32-bit word address, refer to Figure 6.20-11).
[1:0]	Reserved Reserved.

CANFD Tx Event FIFO Status Register (CANFD_TXEFS)

Register	Offset	R/W	Description	Reset Value
CANFD_TXEFS	CANFDx_BA+0x0F4	R	Tx Event FIFO Status	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						TEFL	EFF
23	22	21	20	19	18	17	16
Reserved				EFP			
15	14	13	12	11	10	9	8
Reserved				EFG			
7	6	5	4	3	2	1	0
Reserved		EFFL					

Bits	Description
[31:26]	Reserved Reserved.
[25]	TEFL Tx Event FIFO Element Lost This bit is a copy of interrupt flag TEFL (CANFD_IR[15]). When TEFL is reset, this bit is also reset. 0 = No Tx Event FIFO element lost. 1 = Tx Event FIFO element lost, also set after write attempt to Tx Event FIFO of size zero.
[24]	EFF Event FIFO Full 0 = Tx Event FIFO not full. 1 = Tx Event FIFO full.
[23:21]	Reserved Reserved.
[20:16]	EFP Event FIFO Put Index Tx Event FIFO write index pointer, range 0 to 31
[15:13]	Reserved Reserved.
[12:8]	EFG Event FIFO Get Index Tx Event FIFO read index pointer, range 0 to 31
[7:6]	Reserved Reserved.
[5:0]	EFFL Event FIFO Fill Level Number of elements stored in Tx Event FIFO, range 0 to 32

CANFD Tx Event FIFO Acknowledge Register (CANFD_TXEFA)

Register	Offset	R/W	Description	Reset Value
CANFD_TXEFA	CANFDx_BA+0x0F8	R/W	Tx Event FIFO Acknowledge	0x0000_0000

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

Bits	Description
[31:5]	Reserved Reserved.
[4:0]	EFA Event FIFO Acknowledge Index After the Host has read an element or a sequence of elements from the Tx Event FIFO it has to write the index of the last element read from Tx Event FIFO to EFAI. This will set the Tx Event FIFO Get Index EFGI (CANFD_TXEFS[12:8]) to EFAI + 1 and update the Event FIFO Fill Level EFFL (CANFD_TXEFS[5:0])

6.21 Enhanced 12-bit Analog-to-Digital Converter (EADC)

6.21.1 Overview

The chip contains one 12-bit successive approximation analog-to-digital converter (SAR ADC converter) with 12 external input channels and 2 internal channels. The ADC converter can be started by software trigger, PWM0/1 triggers, BPWM0/1 triggers, Timer0~3 overflow pulse triggers, ADINT0, ADINT1 interrupt EOC (End of conversion) pulse trigger and external pin (EADC0_ST) input signal.

6.21.2 Features

- Analog input voltage range: 0~AV_{DD} (Max to AV_{DD})
- Reference voltage from AV_{DD}
- 12-bit resolution and 10-bit accuracy is guaranteed
- Up to 12 single-end analog external input channels
- 2 internal channels, they are band-gap voltage (V_{BG}), temperature sensor (V_{TEMP})
- Four EADC interrupts (ADINT0~3) with individual interrupt vector addresses
- Maximum EADC clock frequency is 16 MHz
- Up to 730 KSPS conversion rate
- Up to 4 sample modules:
 - Each of sample is configurable for EADC converter channel EADC_CH0~7,12~15 and trigger source
 - Sample module 16, 17 is fixed for EADC channel 16, 17 input sources as band-gap voltage, temperature sensor.
 - Configurable sampling time for each sample module
 - Supports left-adjusted result
 - 12-bit resolution for conversion result and 16-bit resolution for accumulated conversion result
 - Conversion results are held in 6 data registers with valid and overrun indicators
 - Averaging (2ⁿ times, n=0~8) to support up to 12-bit result and over-sampling, or called Accumulation, (2ⁿ times, n=0~8) to support up to 16-bit result
- An ADC conversion can be started by:
 - Write 1 to SWTRGn (EADC_SWTRG[n], n = 0~3)
 - Write 1 to INTSWTRGn (EADC_SWTRG[n], n = 16,17) for internal channel
 - External pin EADC0_ST
 - Timer0~3 overflow pulse triggers
 - ADINT0 and ADINT1 interrupt EOC (End of conversion) pulse triggers
 - BPWM0 triggers
- Supports configurable PDMA transfer
- Auto turn on/off EADC power at power off or operation mode with wait state(10us stable time)
- Supports digital comparator to monitor conversion result and user can select whether to generate an interrupt when conversion result matches the compare register setting

- Supports offset cancellation

6.21.3 Block Diagram

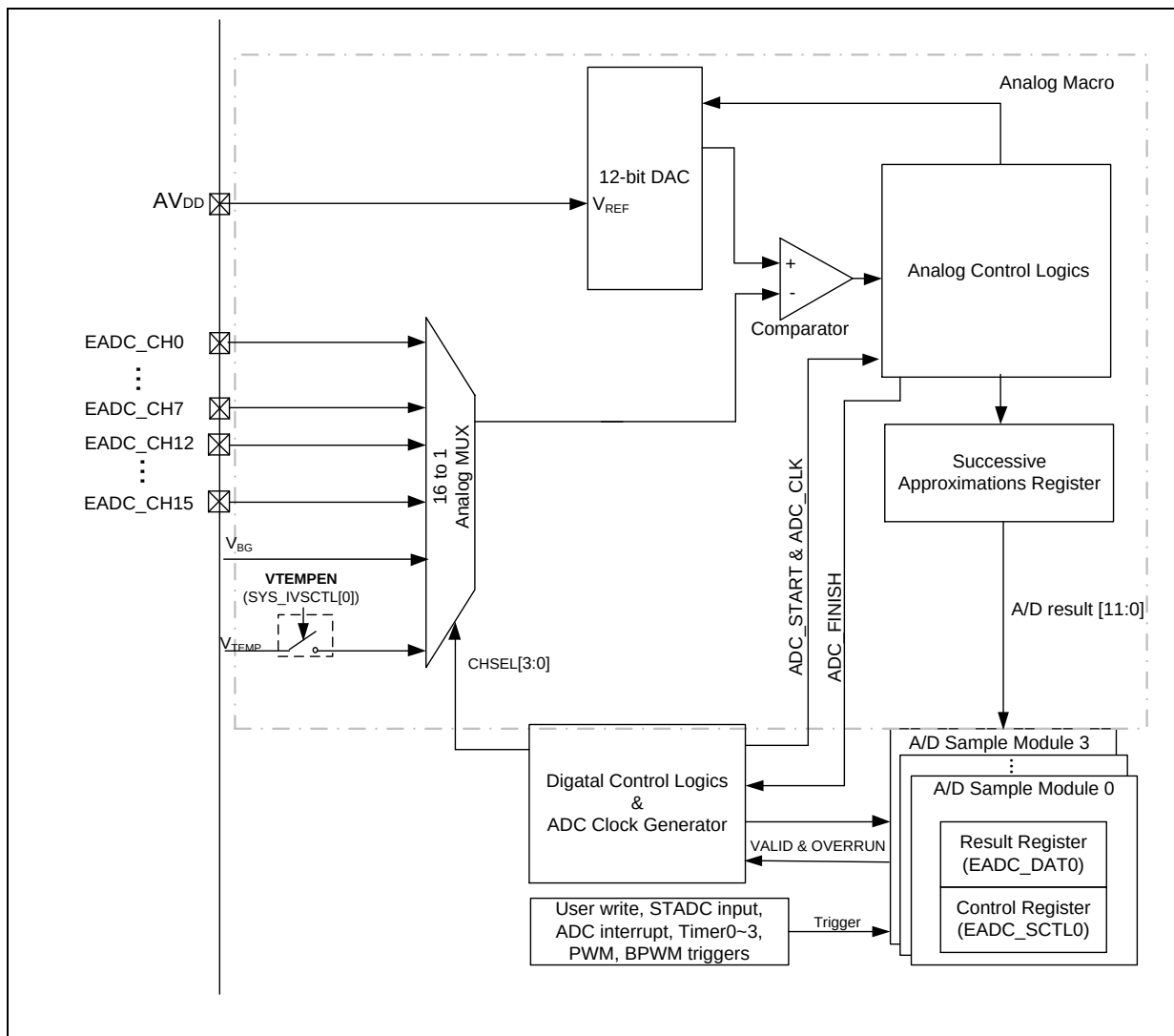


Figure 6.21-1 EADC Converter Block Diagram

6.21.4 Basic Configuration

- Clock Source Configuration
 - Select the clock divider number on EADC DIV (CKL_CLKDIV0[23:16])
 - Enable EADC peripheral clock in EADCCKEN (CLK_APBCLK0[28]).
- Reset Configuration
 - Reset EADC controller in ADCRST (EADC_CTL [1]).
- Pin Configuration

Group	Pin Name	GPIO	MFP

EADC0	EADC0_CH0	PB.0	MFP1
	EADC0_CH1	PB.1	MFP1
	EADC0_CH2	PB.2	MFP1
	EADC0_CH3	PB.3	MFP1
	EADC0_CH4	PB.4	MFP1
	EADC0_CH5	PB.5	MFP1
	EADC0_CH6	PB.6	MFP1
	EADC0_CH7	PB.7	MFP1
	EADC0_CH12	PB.12	MFP1
	EADC0_CH13	PB.13	MFP1
	EADC0_CH14	PB.14	MFP1
	EADC0_CH15	PB.15	MFP1
	EADC0_ST	PF.5	MFP11
		PD.12	MFP14

6.21.5 Functional Description

The EADC controller consists of a 16 channel analog switch, 4 sample modules and a 12-bit successive approximation analog-to-digital converter. The EADC operation is based on sample module 0~3, and each of them has its configuration to decide which trigger source to start the conversion, which channel to convert. Sample module 0~3 can be configured to EADC_CH0~7, EADC_CH12~15 channel and different trigger source. It provides user a flexible means to get the over-sampling results. The sample module 0~3 are shows as follows. Noted that EADC_CH8~11 is reserved.

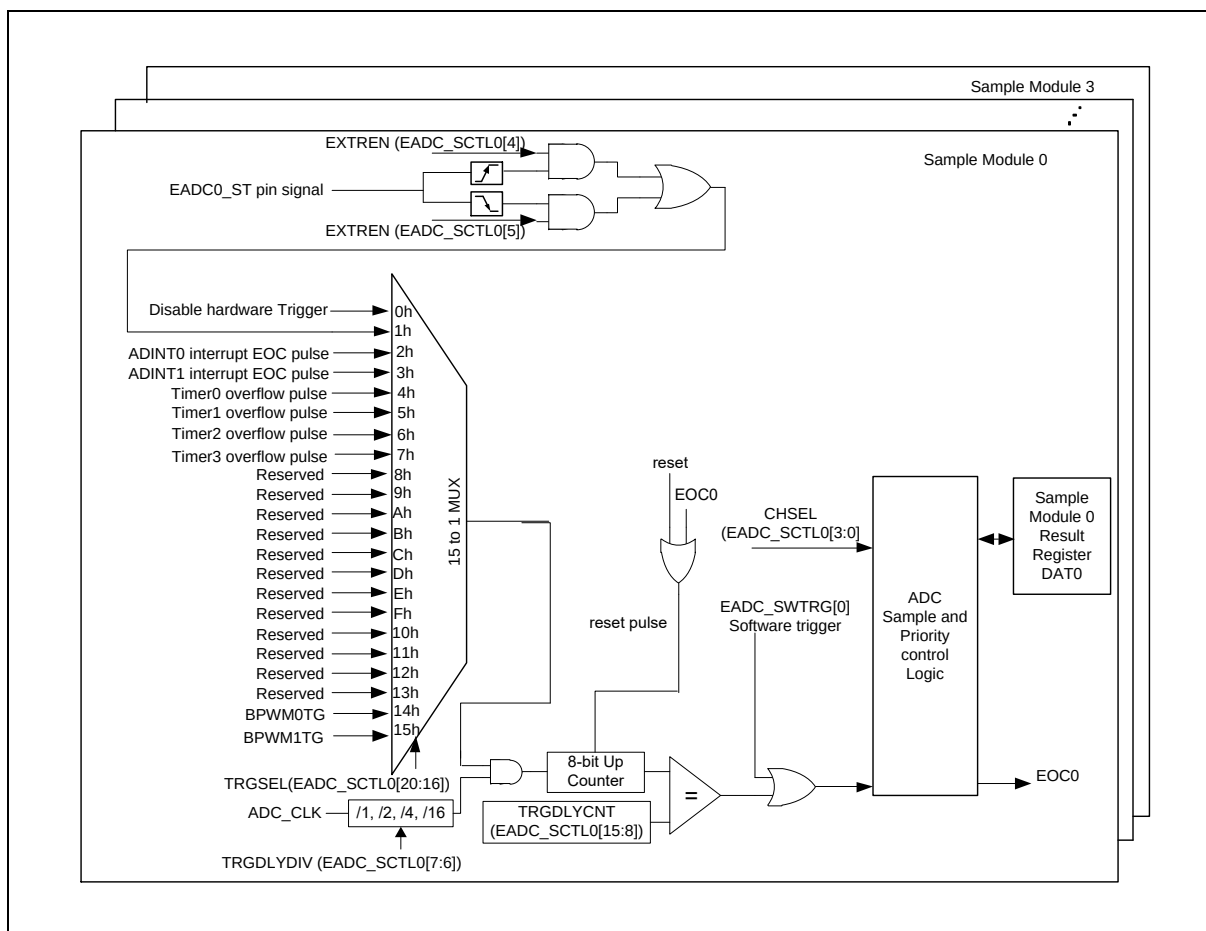


Figure 6.21-2 Sample Module 0~3 Block Diagram

Sample module 16~17 can convert internal channel (V_{BG} , V_{TEMP}) and can be triggered by user write SWTRGn (EADC_SWTRG[n], $n = 16 \sim 17$). Figure 6.21-3 shows the sample module 16~17.

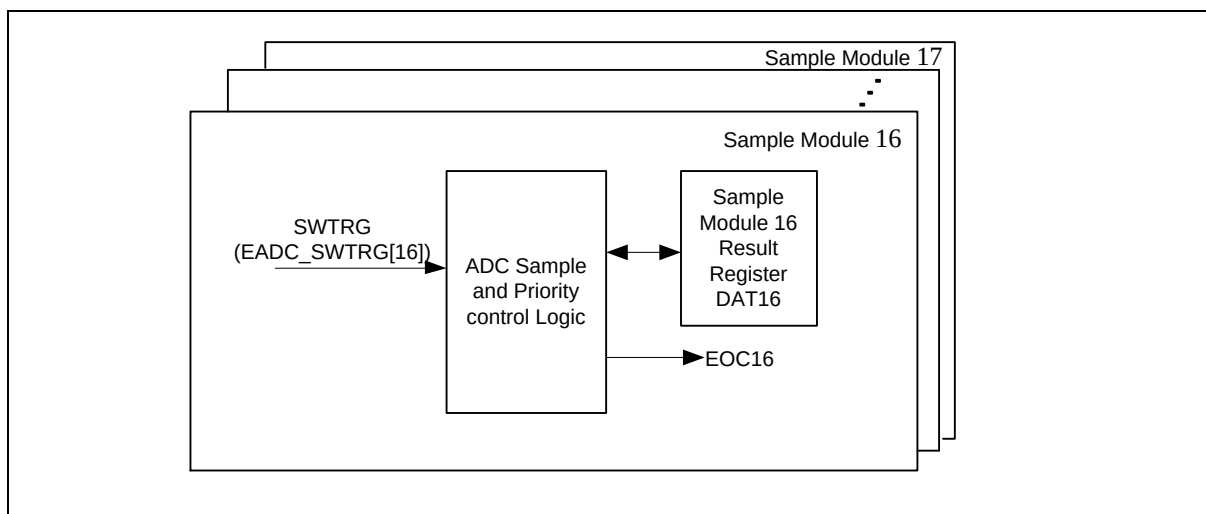


Figure 6.21-3 Sample Module 16~17 Block Diagram

The EADC conversion trigger sources in sample module 0~3 are listed below:

- Write 1 to SWTRGn (EADC_SWTRG[n], n = 0~3)
- External pin EADC0_ST
- Timer0~3 overflow pulse triggers
- ADINT0/1 EADC interrupt EOC (End of conversion) pulse triggers
- PWM0/1 triggers
- BPWM0/1 triggers

The ADINT0 or ADINT1 interrupt pulses are generated whenever the specific sample module ADC EOC (End of conversion) pulse is generated. ADINT0 or ADINT1 interrupt pulse triggers can be fed back to trigger another ADC conversion, and is useful if a continuous scan conversion is needed.

6.21.5.1 EADC Clock Generator

The maximum EADC clock frequency is up to 16 MHz and the maximum sampling rate is up to 730 KSPS. It needs 18 EADC clocks to complete an ADC conversion by default setting.

The clock control of EADC is shown as Figure 6.21-4. The EADC peripheral clock source is from PCLK1 clock, the EADC clock frequency is divided by an 8-bit pre-scalar with the following formula:

$$\text{EADC clock frequency} = (\text{PCLK1}) / (\text{EADC DIV} (\text{CLK\_CLKDIV0}[23:16]) + 1)$$

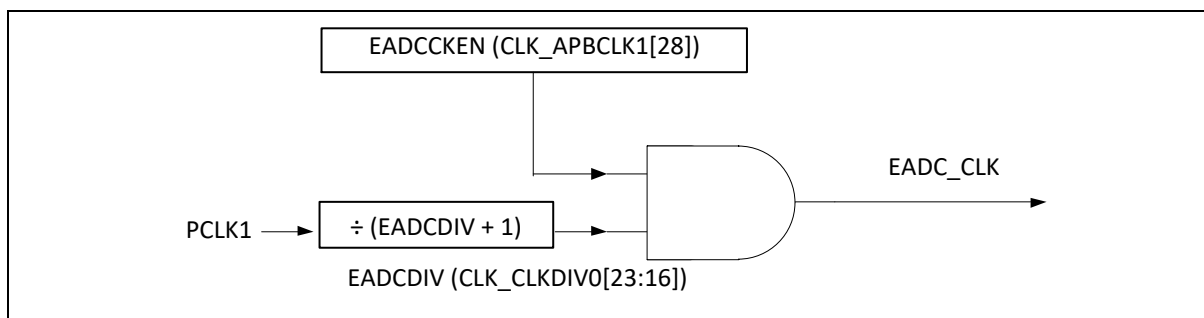


Figure 6.21-4 EADC Clock Control

6.21.5.2 EADC Software Trigger Mode

When an EADC conversion is performed on the sample module specified single channel, the operations are as follows:

1. ADC conversion is started when the SWTRGn (EADC_SWTRG[n], n=0~3) (internal channel is triggered by INTSWTRGn, n=16,17) is set to 1 by user or other trigger inputs.
2. When ADC conversion is finished, the 12-bit result is stored in the EADC data register EADC_DATn (n=0~3), (n=16,17 for internal channel) corresponding to the sample module.
3. Set SPLIEn (EADC_INTSRCm[n], n=0~3 and n=16,17, m=0~3) to define which sample module affects ADIFn (EADC_STATUS2[n], n=0~3) flag.
4. On completion of conversion, the ADIFn (EADC_STATUS2[3:0], n=0~3) is set to 1 and EADC interrupt (ADINTn, n=0~3) is requested if the EADCIENn (EADC_CTL[5:2], n=0~3) bit is set to 1.
5. The SWTRGn (n=0~3) bit remains 1 during ADC conversion. When ADC conversion ends, the SWTRGn (n=0~3) bit is automatically cleared to 0 and the ADC converter will do another pending conversion.

The timing diagram of a conversion cycle is shown in Figure 6.21-5.

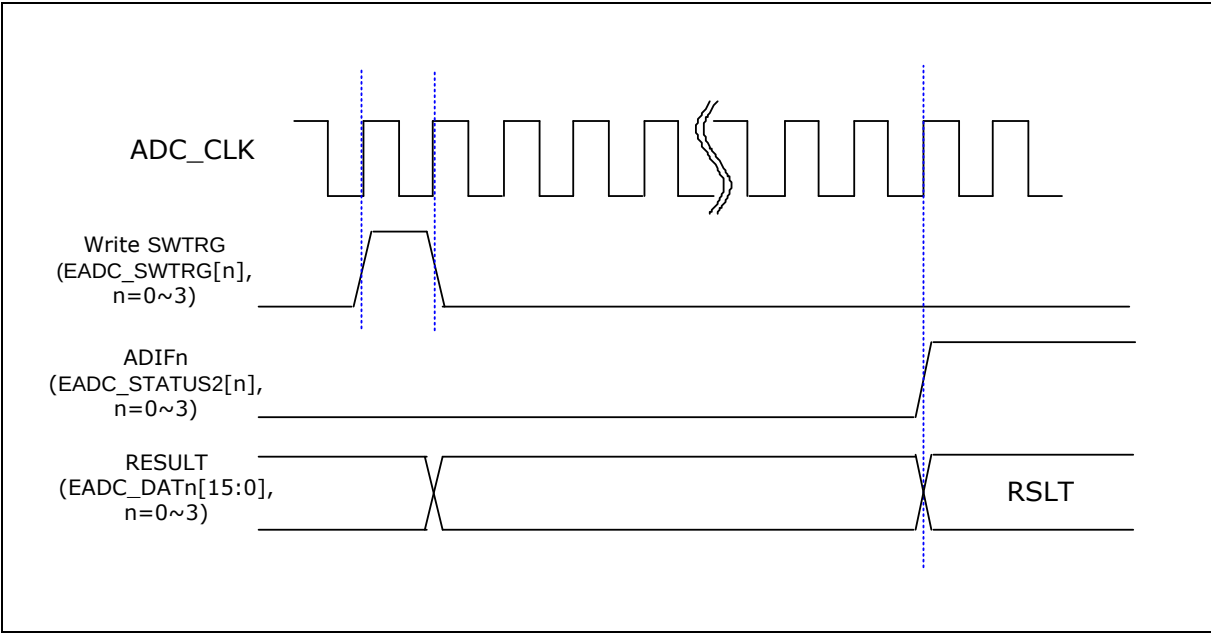


Figure 6.21-5 Example EADC Conversion Timing Diagram, n=0~3

If more than one sample module is enabled to convert analog signal, the sample module specified channel with highest priority is firstly converted and other enabled sample module will be pended. The lower number sample module has higher priority. The sample module 0 is highest priority and the sample module 3 is lowest priority.

6.21.5.3 EADC Conversion Priority

There is a priority group converter for determining the conversion order when multiple sample module trigger flags are set at the same time, if two sample module are triggered at the same time, the sample module with lower number will start to convert EADC first.

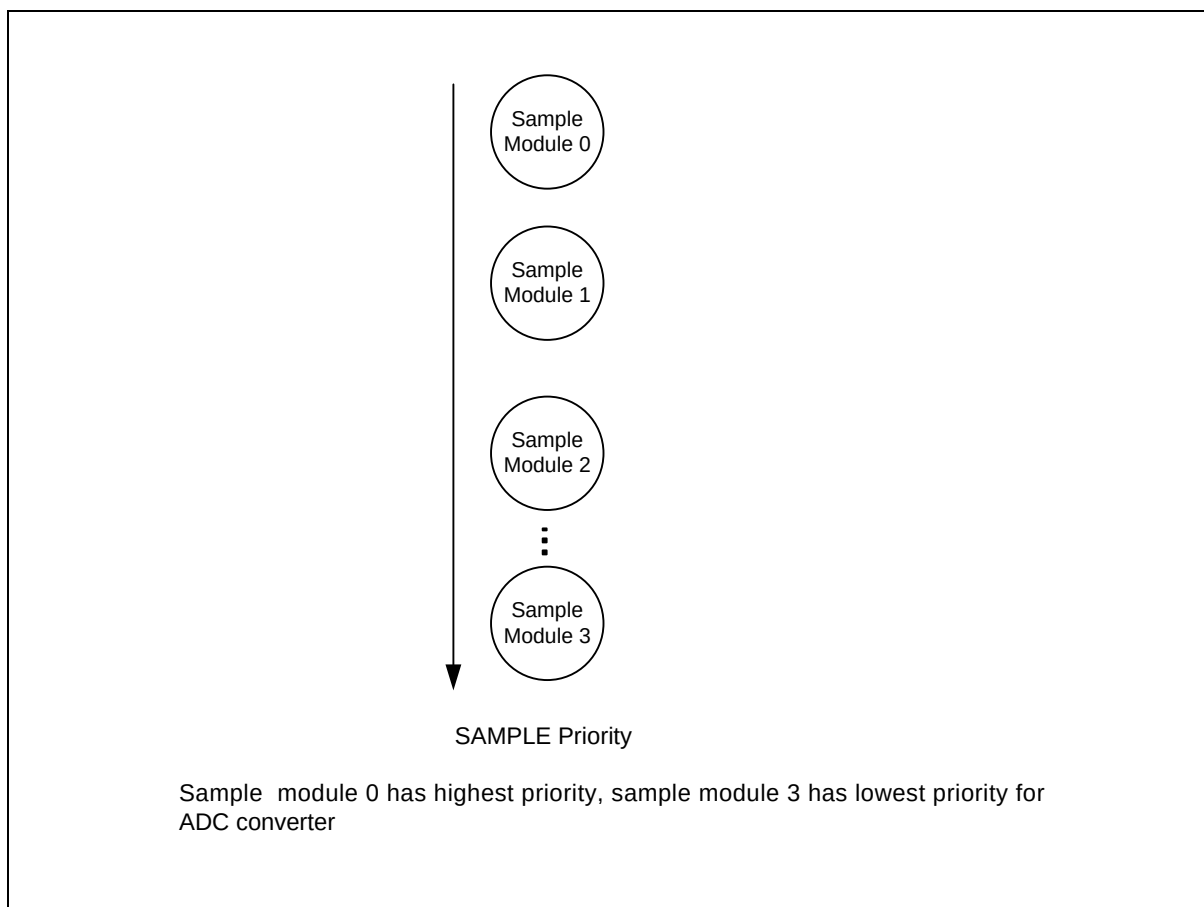


Figure 6.21-6 Sample Module Conversion Priority Arbitrator Diagram

6.21.5.4 EADC Sample Module End of Conversion (EOC) Interrupt Operation

There are 4 EADC interrupts ADINT0~3, and each of these interrupts has its own interrupt vector address and ADINT0/ADINT1 can be configured to set multiple sample module EOC pulse (sample module 0~3 End of conversion pulses) as its interrupt trigger source.

When EADCIEN0 (EADC_CTL[2]) = 1 and SPLIE_n (EADC_INTSRC0[n], n=0~3 and n=16,17) = 1, all sample module EOC (End of conversion) pulses can cause an ADINT0 interrupt.

The ADINT0, ADINT1 interrupt pulses are generated whenever the specific sample module ADC EOC pulse is generated. The ADINT0, ADINT1 interrupt pulses can also be the sample module conversion trigger sources. The ADINT0, ADINT1 interrupt pulses can be used to do the EADC continuous scan conversion.

The example of continuous scan triggered by interrupt is as follows:

1. If EADC sample module 2 EOC2 pulse is selected as ADINT0 interrupt trigger SPLIE2 (EADC_INTSRC0[2]) = 1 and ADINT0 is selected as sample module 0, 1, 2 hardware conversion trigger.
2. Set software trigger SWTRG2 (EADC_SWTRG[2]) to 1 to start a sample module 2 conversion an EOC2 pulse signal. The EOC2 pulse signal is served as ADINT0 interrupt pulse at end of sample module 2 EADC conversion. ADINT0 interrupt pulse will trigger the sample module 0, 1, 2 to start the EADC conversions.
3. ADINT0 interrupt pulse repeats to trigger sample module 0, 1, 2 EADC conversions automatically.

4. Clear TRGSEL (EADC_SCTL2[20:16]) to 0 to disable sample module 2 ADINT0 interrupt pulse hardware trigger, if needs to stop the continuous scan.

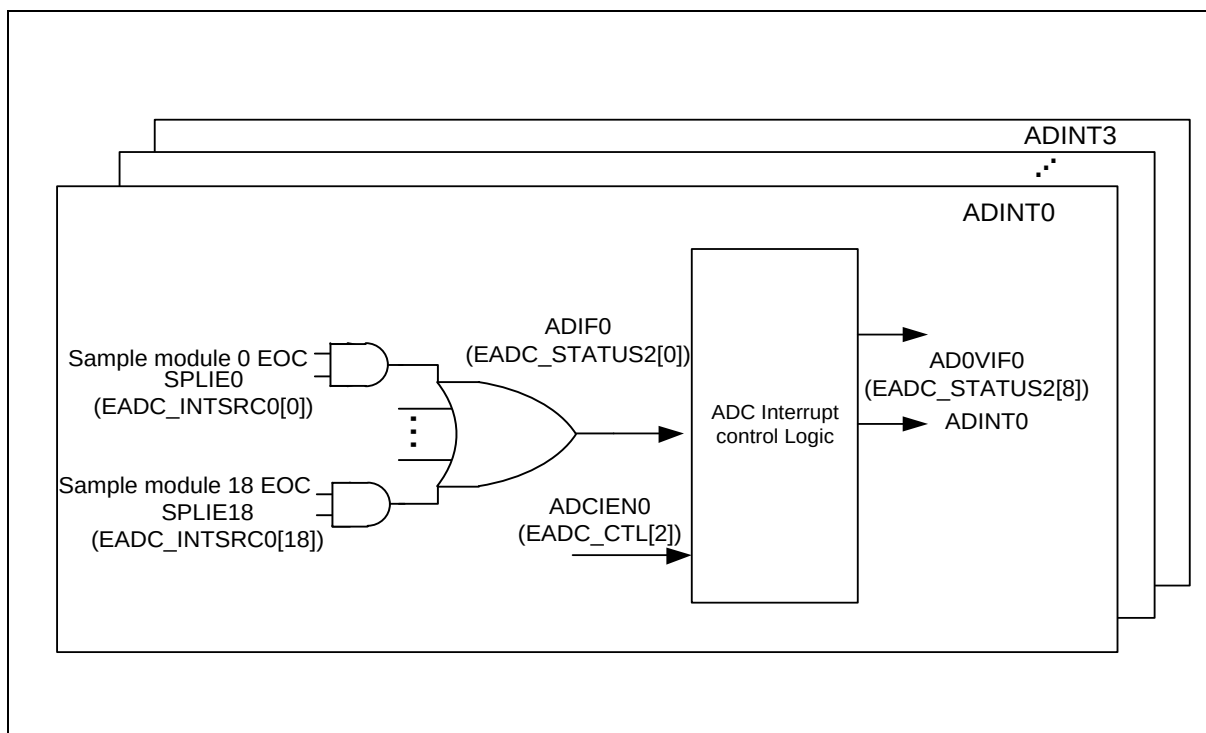


Figure 6.21-7 Specific Sample Module ADC EOC Signal for ADINT0~3 Interrupt

6.21.5.5 EADC Trigger by External Pin EADC0_ST, Timer Trigger, and BPWM Trigger

ADC conversion can be triggered by external pin EADC0_ST request. Setting the TRGSEL (EADC_SCTLn[20:16], n=0~3) to 0x01 is to select external trigger input from the EADC0_ST pin. User can set EXTREN (EADC_SCTLn[4], n=0~3) to enable pin EADC0_ST trigger condition is falling or rising edge. There is a de-bounce circuit to detect falling or rising edge. If rising edge trigger condition is selected, the low state must be kept at least 2 PCLK cycles and the following high state must be kept at least 3 PCLK cycles. If falling edge trigger condition is selected, the high state must be kept at least 2 PCLK cycles and the following low state must be kept at least 3 PCLK cycles. Pulse that is shorter than this specification will be ignored. The external trigger timing is shown in Figure 6.21-8.

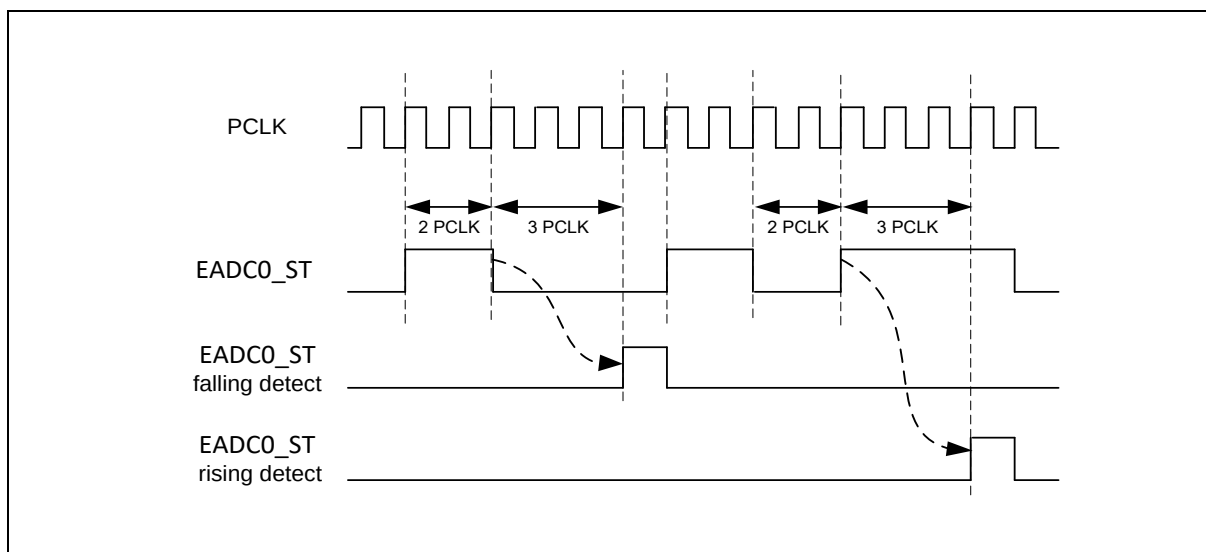


Figure 6.21-8 EADC0_ST De-bounce Timing Diagram

There are 4 Timer trigger sources, and 1 BPWM trigger sources which can be selected to configure sample module 0~3 for EADC start trigger. The detail trigger conditions are described at BPWM_EADCTS0, and TIMER0_CTL ~ TIMER3_CTL register.

6.21.5.6 EADC Trigger Delay

The device also allows user to configure the amount of delay period to EADC start after hardware detected the external trigger. User can configure the trigger delay time by setting TRGDLYCNT (EADC_SCTLn[15:8], n=0~3) and TRGDLYDIV (EADC_SCTLn[7:6], n=0~3). Figure 6.21-9 shows the programmable delay time for PWM-triggered EADC start conversion.

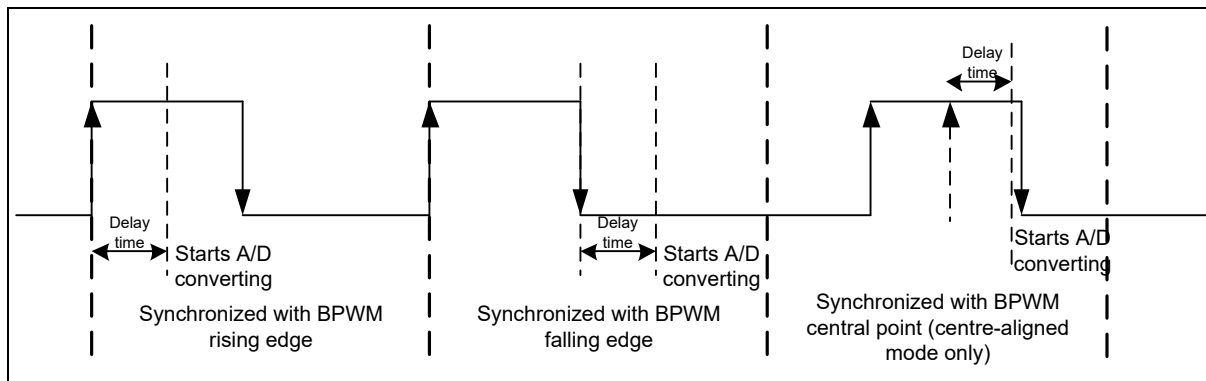


Figure 6.21-9 BPWM-triggered EADC Start Conversion

Figure 6.21-10 shows the programmable delay time for other trigger source.

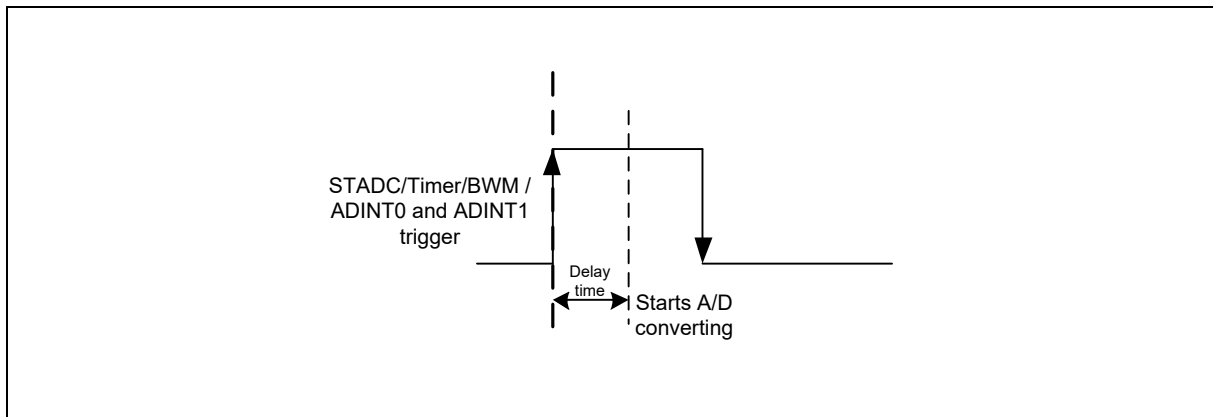


Figure 6.21-10 External triggered EADC Start Conversion

6.21.5.7 Input Sampling and ADC Conversion Time

The ADC converter sample the analog input when ADC conversion start delay time (T_d) has passed after $SWTRG_n$ ($EADC\_SWTRG[n]$, $n=0\sim3$ and $n=16,17$) is set to 1, then start conversion. Due to EADC clock is generated by PCLK divided by ($EADC_{DIV}(\text{CLK\_CLKDIV0}[23:16])+1$), the maximum delay time from user write $SWTRG_n$ to ADC start sampling analog input time is two EADC clock cycles. The start delay time is shown in Figure 6.21-11.

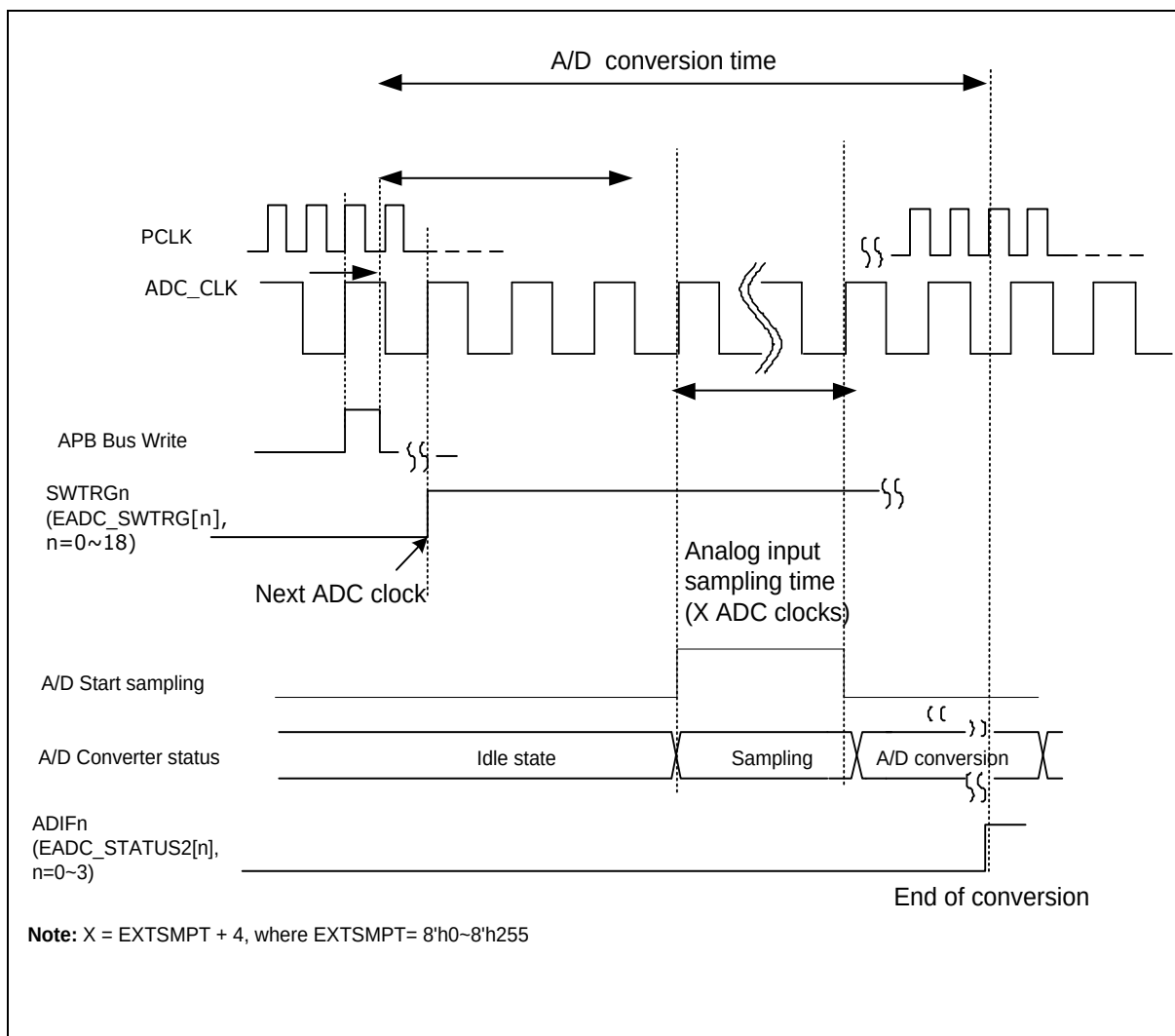


Figure 6.21-11 Conversion Start Delay Timing Diagram

6.21.5.8 ADC Extend Sampling Time

When ADC operates at high EADC clock rate, the sampling time of analog input voltage may not be enough if the analog channel has heavy loading to cause fully charge time is longer. User can set extend sampling time by writing EXTSMPT (EADC_SCTLn[31:24], n=0~15) for each sample module. The ADC extend sampling time is present between ADC controller judging which channel to be converted and ADC starting conversion. The range of extend sampling time is from 0 ~255 EADC clock. The extended sampling time is shown in Figure 6.21-12.

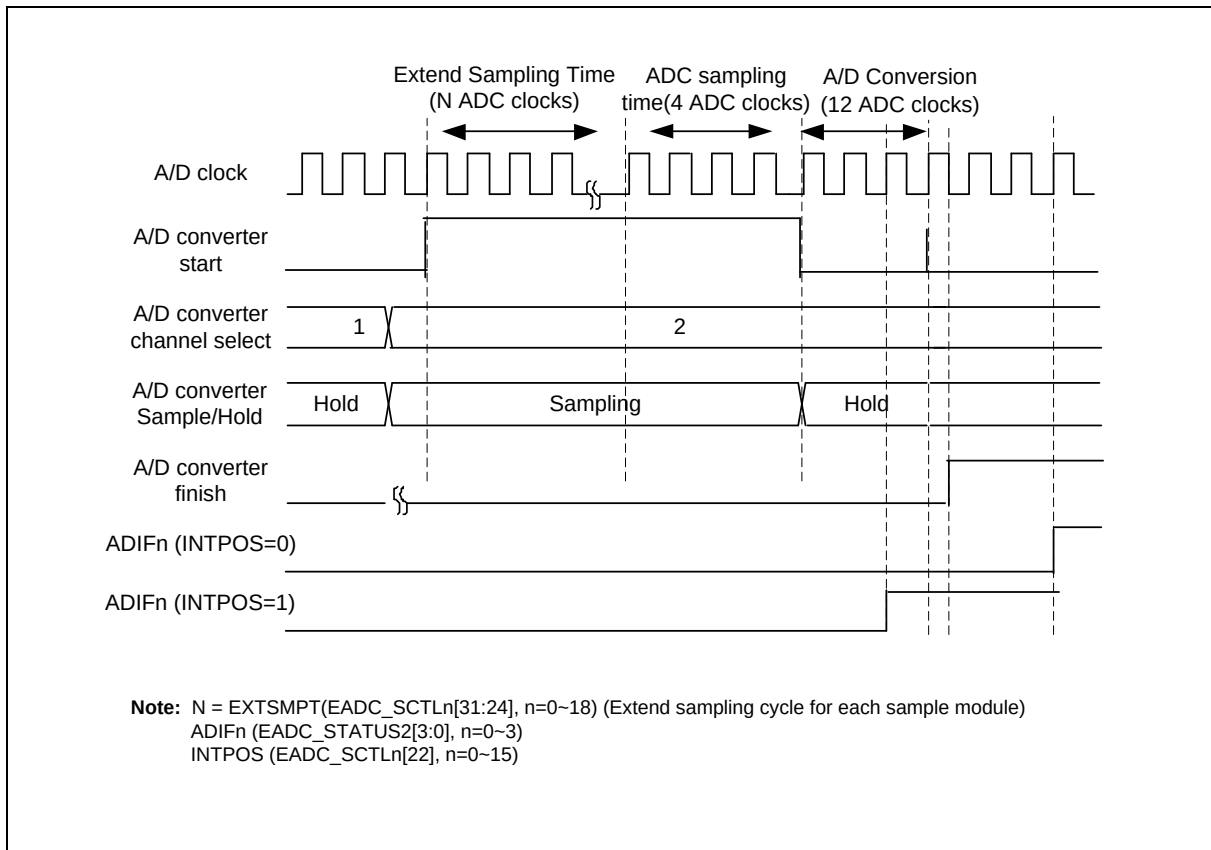


Figure 6.21-12 ADC Extend Sampling Timing Diagram

6.21.5.9 EADC conversion Result Format

At the end of each conversion (when an EOC event occurs), the result of the converted data is stored in the EADC_DATn data register which is 16-bit wide. The format of the EADC_DATn depends on the configured data alignment and resolution. The ALIGN (EADC_MnCTL1[0]) bit selects the alignment of the conversion result. Data can be right-aligned (ALIGN=0) or left-aligned (ALIGN=1) as shown in Figure 6.21-13. It is noted that n = 0~15.

- 12-bit left alignment: EADC conversion result is stored in EADC_DATn[15:4] bits. EADC_DATn[31:16] and EADC_DATn[3:0] are ignored in EADC conversion.
- 12-bit right alignment: EADC conversion result is stored in EADC_DATn[11:0] bits, EADC_DATn[31:12] are ignored in AC conversion. It is noted that n = 0~15.

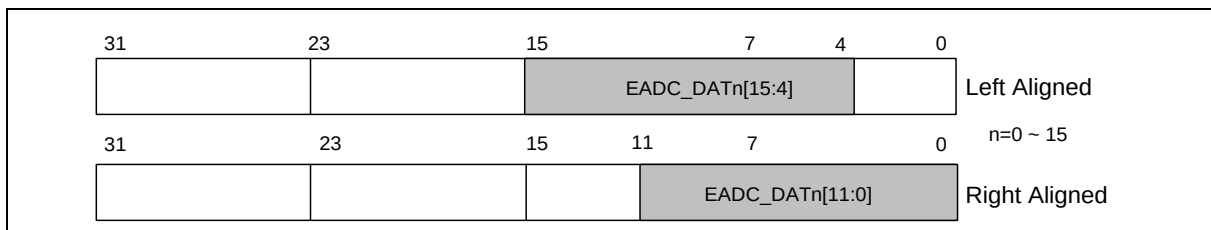


Figure 6.21-13 EADC_DATn (n= 0~15) Aligned Format

6.21.5.10 Accumulation

The function provides multiple consecutive conversion results to be accumulated to a final conversion result. Setting ACU (EADC_MnCTL1[7:4], n=0~15) determine which sample module and the number of samples to be accumulated. It is noted that when more than 16 samples to be accumulated, the final result after accumulating will exceed the 16-bit RESULT(EADC_DATn [0:15], n=0~18) register size. In this case, the result is right shifted automatically to fit within the appropriate register size. It is noted that accumulation function is only effective without data left alignment.

ACU (EADC_MnCTL1 [7:4]), N=0~3	Number Of Samples To Be Accumulated	Right Shift Division Factor	Final Result Precision
0x0	1	1	12 bits
0x1	2	1	13 bits
0x2	4	1	14 bits
0x3	8	1	15 bits
0x4	16	1	16 bits
0x5	32	2	16 bits
0x6	64	4	16 bits
0x7	128	8	16 bits
0x8	256	16	16 bits

Table 6.21-1 Setting of Accumulation and Conversion Result Precision

6.21.5.11 Averaging

The function increasing the sample accuracy with reduced sampling rate and the feature is suitable in noisy operating environment for getting more stable average conversion results. According to the accumulated final result in 6.21.5.10, setting AVG (EADC_MnCTL1[3]) can enable averaging function as shown in Table 6.21-2. The automatic division factor is decided by the number of accumulated samples and the final result will get 12 bits precision. It is noted that averaging is only effective when accumulating two more samples without data left alignment.

AVG EADC_MnCTL1 [1]	ACU EADC_MnCTL1 [7:4]	Number Of Samples To Be Accumulated	Automatic Division Factor	Final Result Precision
0x1	0x0	1	1	12 bits
0x1	0x1	2	2	12 bits
0x1	0x2	4	4	12 bits
0x1	0x3	8	8	12 bits
0x1	0x4	16	16	12 bits
0x1	0x5	32	32	12 bits
0x1	0x6	64	64	12 bits
0x1	0x7	128	128	12 bits
0x1	0x8	256	256	12 bits

Table 6.21-2 The Setting of Averaging and Conversion Result Precision

6.21.5.12 Conversion Result Monitor by Compare Mode

The EADC controller provides four sets of compare registers EADC_CMP0 ~ EADC_CMP3 to monitor a maximum of four specified sample module 0~3 conversion results from ADC conversion module, as shown in Figure 6.21-14. User can select which sample module result to be monitored by setting CMPSPn (EADC_CMPn[7:3], n =0~3) and CMPCOND (EADC_CMPn[2], where n =0~3) is used to check conversion result is less than specify value or greater than (equal to) value specified in CMPDAT (EADC_CMPn[27:16], where n =0~3). When the conversion of the sample module specified by CMPSPn is completed, the comparing action will be triggered one time automatically. When the compare result meets the compare condition, the internal compare match counter will increase 1. If the compare result does not meet the condition, the compare match counter will reset to 0. When counter value reach the setting of (CMPMCNT (EADC_CMPn[11:8])+1, where n =0~3) then EADCMPPn (EADC_STATUS2[7:4], where n =0~3) bit will be set to 1, if EADCMPIE (EADC_CMPn[1], n =0~3) is set then an ADINT3 interrupt request is generated. User can use it to monitor the external analog input pin voltage transition. Detailed logics diagram is shown in Figure 6.21-14.

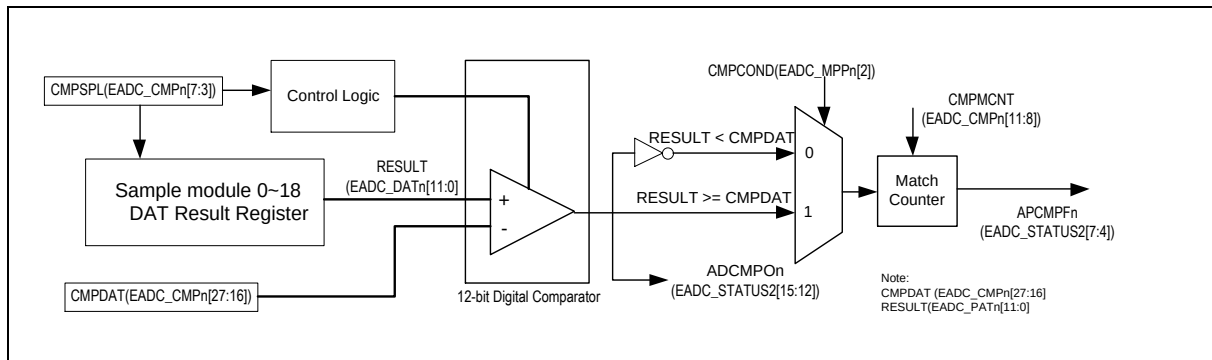


Figure 6.21-14 ADC Conversion Result Monitor Logics Diagram

The EADC controller supports a window compare mode. User can set CMPWEN (EADC_CMP0[15]/EADC_CMP2[15]) to enable this function. If user enables this function, EADCMPP0 (EADC_STATUS2[4]) will be set when both EADC_CMP0 and EADC_CMP1 compared condition matched. EADCMPP2 (EADC_STATUS2[6]) will be set when both EADC_CMP2 and EADC_CMP3 compared condition matched.

6.21.5.13 Auto Power On/Off Mode

The EADC has an automatic power management feature called auto power on-off mode. Setting AUTOFF (EADC_PWRCTL[5]) to 1 can enable this function. AUTOFF needs to be combined with AUTOPDTH (EADC_PWRCTL[23:20]) to work. When a Sample Module is finished, EADC will enter idle state. If AUTOFF (EADC_PWRCTL[5]) is set to 1 and the interval of time in idle state is longer than AUTOPDTH, EADC will power-down ADC converter analog circuit to reduce power consumption. The EADCEN (EADC_CTL[0]) will be set to 0 when EADC power-down and set to 1 when EADC is wake-up.

The EADC will automatically wake up when a conversion is started (by software or hardware trigger). A start-up time is automatically inserted between the trigger event which starts the conversion and the sampling time of the EADC. The start-up time must be longer than 10 us that can be set by STUPT (EADC_PWRCTL[19:8]).

6.21.5.14 PDMA Request

The EADC controller supports configurable PDMA. User can set source address of PDMA channels as EADC_CURDAT and enable PDMATEN (EADC_PDMACTL[3:0]), to decide which external channel should be performed PDMA transfer. For internal channel performed PDMA transfer, user can enable INTPDMATEN (EADC_PDMACTL[17:16]) to set source address of PDMA channels. After enabling PDMATEN and INTPDMATEN, if any VALID (EADC_DATn[17], n=0~3) is high, the EADC controller will send request to PDMA and PDMA will read EADC_CURDAT to get results. The EADC_CURDAT register is a shadow register of highest priority EADC_DAT register. The lower number sample module

is higher priority. After the PDMA reads EADC_CURDAT register, the VAILD of the shadow EADC_DAT register will be automatically cleared.

6.21.5.15 Interrupt Sources

The ADC converter generates ADIFn (EADC_STATUS2[3:0], n=0~3) at the start of conversion or the end of conversion decide by INTPOS (EADC_SCTLn[22], n=0~3). If EADCIENn (EADC_CTL[5:2], n=0~3) is set then conversion end interrupt request ADINTn (n=0~3) is generated. The controller of interrupts is shown as Figure 6.21-15.

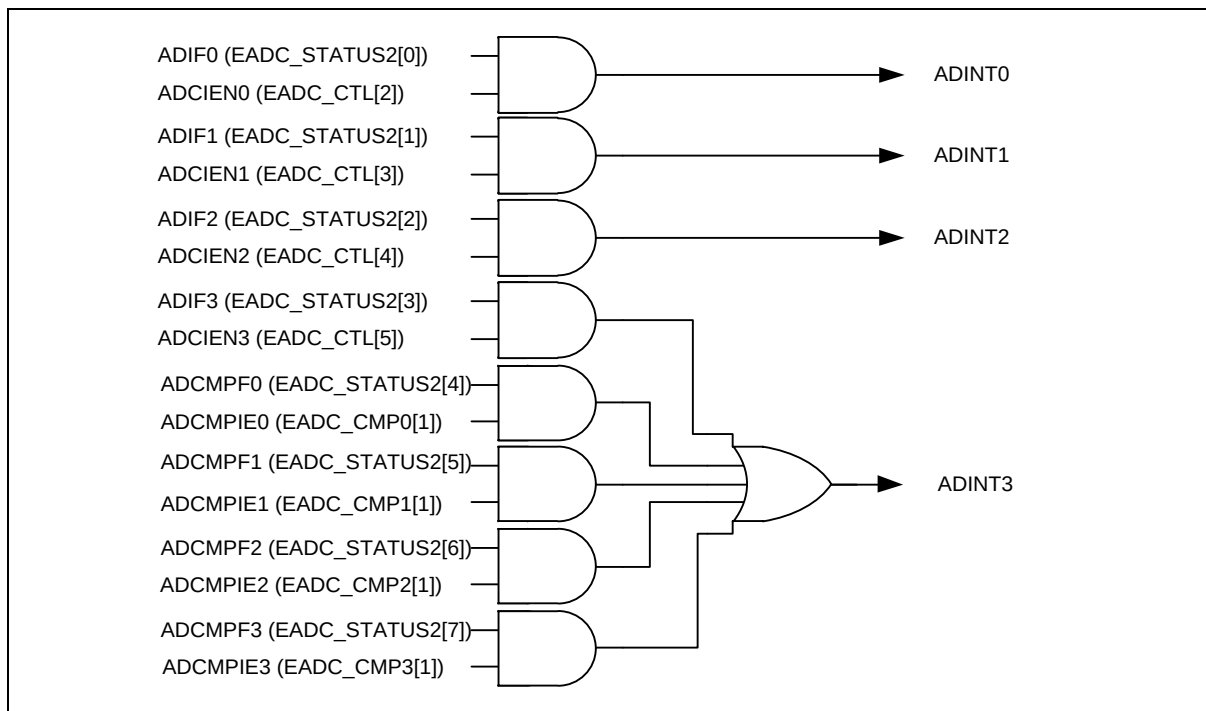


Figure 6.21-15 ADC Controller Interrupts

6.21.6 Register Map

R: read only, W: write only, R/W: both read and write

Register	Offset	R/W	Description	Reset Value
EADC Base Address: EADC_BA = 0x4004_3000				
EADC_DAT0	EADC_BA+0x00	R	ADC Data Register 0 for Sample Module 0	0x0000_0000
EADC_DAT1	EADC_BA+0x04	R	ADC Data Register 1 for Sample Module 1	0x0000_0000
EADC_DAT2	EADC_BA+0x08	R	ADC Data Register 2 for Sample Module 2	0x0000_0000
EADC_DAT3	EADC_BA+0x0C	R	ADC Data Register 3 for Sample Module 3	0x0000_0000
EADC_DAT16	EADC_BA+0x40	R	ADC Data Register 16 for Sample Module 16	0x0000_0000
EADC_DAT17	EADC_BA+0x44	R	ADC Data Register 17 for Sample Module 17	0x0000_0000
EADC_CURDAT	EADC_BA+0x4C	R	EADC PDMA Current Transfer Data Register	0x0000_0000
EADC_CTL	EADC_BA+0x50	R/W	ADC Control Register	0x0004_0000
EADC_SWTRG	EADC_BA+0x54	W	ADC Sample Module Software Start Register	0x0000_0000
EADC_PENDSTS	EADC_BA+0x58	R/W	ADC Sample Module Start of Conversion Pending Flag Register	0x0000_0000
EADC_OVSTS	EADC_BA+0x5C	R/W	ADC Sample Module Start of Conversion Overrun Flag Register	0x0000_0000
EADC_SCTL0	EADC_BA+0x80	R/W	ADC Sample Module 0 Control Register	0x0000_0000
EADC_SCTL1	EADC_BA+0x84	R/W	ADC Sample Module 1 Control Register	0x0000_0000
EADC_SCTL2	EADC_BA+0x88	R/W	ADC Sample Module 2 Control Register	0x0000_0000
EADC_SCTL3	EADC_BA+0x8C	R/W	ADC Sample Module 3 Control Register	0x0000_0000
EADC_SCTL16	EADC_BA+0xC0	R/W	ADC Sample Module 16 Control Register	0x0000_0000
EADC_SCTL17	EADC_BA+0xC4	R/W	ADC Sample Module 17 Control Register	0x0000_0000
EADC_INTSRC0	EADC_BA+0xD0	R/W	EADC Interrupt 0 Source Enable Control Register.	0x0000_0000
EADC_INTSRC1	EADC_BA+0xD4	R/W	EADC Interrupt 1 Source Enable Control Register.	0x0000_0000
EADC_INTSRC2	EADC_BA+0xD8	R/W	EADC Interrupt 2 Source Enable Control Register.	0x0000_0000
EADC_INTSRC3	EADC_BA+0xDC	R/W	EADC Interrupt 3 Source Enable Control Register.	0x0000_0000
EADC_CMP0	EADC_BA+0xE0	R/W	ADC Result Compare Register 0	0x0000_0000
EADC_CMP1	EADC_BA+0xE4	R/W	ADC Result Compare Register 1	0x0000_0000
EADC_CMP2	EADC_BA+0xE8	R/W	ADC Result Compare Register 2	0x0000_0000
EADC_CMP3	EADC_BA+0xEC	R/W	ADC Result Compare Register 3	0x0000_0000
EADC_STATUS0	EADC_BA+0xF0	R	ADC Status Register 0	0x0000_0000
EADC_STATUS1	EADC_BA+0xF4	R	ADC Status Register 1	0x0000_0000

Register	Offset	R/W	Description	Reset Value
EADC Base Address: EADC_BA = 0x4004_3000				
EADC_STATUS2	EADC_BA+0xF8	R/W	ADC Status Register 2	0x0011_0000
EADC_STATUS3	EADC_BA+0xFC	R	ADC Status Register 3	0x0000_001F
EADC_PWRCTL	EADC_BA+0x110	R/W	EADC Power Management Control Register	0x00f0_c000
EADC_PDMACTL	EADC_BA+0x130	R/W	EADC PDMA Control Register	0x0000_0000
EADC_M0CTL1	EADC_BA+0x140	R/W	EADC Sample Module0 Control Register 1	0x0000_0000
EADC_M1CTL1	EADC_BA+0x144	R/W	EADC Sample Module1 Control Register 1	0x0000_0000
EADC_M2CTL1	EADC_BA+0x148	R/W	EADC Sample Module2 Control Register 1	0x0000_0000
EADC_M3CTL1	EADC_BA+0x14C	R/W	EADC Sample Module3 Control Register 1	0x0000_0000
EADC_OFFSETCAL	EADC_BA+0x208	R/W	ADC Result Offset Cancellation Register	0x0000_001f

6.21.7 Register Description

ADC Data Registers (EADC_DAT0~ EADC_DAT17)

Register	Offset	R/W	Description	Reset Value
EADC_DAT0	EADC_BA+0x00	R	ADC Data Register 0 for Sample Module 0	0x0000_0000
EADC_DAT1	EADC_BA+0x04	R	ADC Data Register 1 for Sample Module 1	0x0000_0000
EADC_DAT2	EADC_BA+0x08	R	ADC Data Register 2 for Sample Module 2	0x0000_0000
EADC_DAT3	EADC_BA+0x0C	R	ADC Data Register 3 for Sample Module 3	0x0000_0000
EADC_DAT16	EADC_BA+0x40	R	ADC Data Register 16 for Sample Module 16	0x0000_0000
EADC_DAT17	EADC_BA+0x44	R	ADC Data Register 17 for Sample Module 17	0x0000_0000

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

Bits	Description	
[31:18]	Reserved	Reserved.
[17]	VALID	Valid Flag This bit is set to 1 when corresponding sample module channel analog input conversion is completed and cleared by hardware after EADC_DAT register is read. 0 = Data in RESULT[11:0] bits is not valid. 1 = Data in RESULT[11:0] bits is valid.
[16]	OV	Overflow Flag If converted data in RESULT[11:0] has not been read before new conversion result is loaded to this register, OV is set to 1. 0 = Data in RESULT[11:0] is recent conversion result. 1 = Data in RESULT[11:0] is overwrite. Note: It is cleared by hardware after EADC_DAT register is read.
[15:0]	RESULT	ADC Conversion Result This field contains 12 bits conversion result.

EADC PDMA Current Transfer Data Register (EADC_CURDAT)

Register	Offset	R/W	Description	Reset Value
EADC_CURDAT	EADC_BA+0x4C	R	EADC PDMA Current Transfer Data Register	0x0000_0000

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

Bits	Description
[31:19]	Reserved Reserved.
[18:0]	EADC PDMA Current Transfer Data (Read Only) This register is a shadow register of EADC_DATn (n=0~18) for PDMA support. Note: After PDMA reads this register, the VAILD of the shadow EADC_DAT register will be automatically cleared.

ADC Control Register (EADC_CTL)

Register	Offset	R/W	Description	Reset Value
EADC_CTL	EADC_BA+0x50	R/W	ADC Control Register	0x0004_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		EADCIEN3	EADCIEN2	EADCIEN1	EADCIEN0	EADCRST	EADCEN

Bits	Description
[31:6]	Reserved Reserved.
[5]	EADCIEN3 Specific Sample Module ADC ADINT3 Interrupt Enable Bit The ADC converter generates a conversion end ADIF3 (EADC_STATUS2[3]) upon the end of specific sample module ADC conversion. If EADCIEN3 bit is set then conversion end interrupt request ADINT3 is generated. 0 = Specific sample module ADC ADINT3 interrupt function Disabled. 1 = Specific sample module ADC ADINT3 interrupt function Enabled.
[4]	EADCIEN2 Specific Sample Module ADC ADINT2 Interrupt Enable Bit The ADC converter generates a conversion end ADIF2 (EADC_STATUS2[2]) upon the end of specific sample module ADC conversion. If EADCIEN2 bit is set then conversion end interrupt request ADINT2 is generated. 0 = Specific sample module ADC ADINT2 interrupt function Disabled. 1 = Specific sample module ADC ADINT2 interrupt function Enabled.
[3]	EADCIEN1 Specific Sample Module ADC ADINT1 Interrupt Enable Bit The ADC converter generates a conversion end ADIF1 (EADC_STATUS2[1]) upon the end of specific sample module ADC conversion. If EADCIEN1 bit is set then conversion end interrupt request ADINT1 is generated. 0 = Specific sample module ADC ADINT1 interrupt function Disabled. 1 = Specific sample module ADC ADINT1 interrupt function Enabled.
[2]	EADCIEN0 Specific Sample Module ADC ADINT0 Interrupt Enable Bit The ADC converter generates a conversion end ADIF0 (EADC_STATUS2[0]) upon the end of specific sample module ADC conversion. If EADCIEN0 bit is set then conversion end interrupt request ADINT0 is generated. 0 = Specific sample module ADC ADINT0 interrupt function Disabled. 1 = Specific sample module ADC ADINT0 interrupt function Enabled.
[1]	EADCRST EADC ADC Converter Control Circuits Reset 0 = No effect. 1 = Cause EADC control circuits reset to initial state, but not change the EADC registers value. Note: EADCRST bit remains 1 during EADC reset. When EADC reset end, the EADCRST bit is automatically cleared to 0.

Bits	Description	
[0]	EADCEN	ADC Converter Enable Bit 0 = Disabled EADC. 1 = Enabled EADC. Note: Before starting ADC conversion function, this bit should be set to 1. Clear it to 0 to disable ADC converter analog circuit power consumption.

ADC Sample Module Software Start Register (EADC_SWTRG)

Register	Offset	R/W	Description	Reset Value
EADC_SWTRG	EADC_BA+0x54	W	ADC Sample Module Software Start Register	0x0000_0000

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

Bits	Description
[31:18]	Reserved Reserved.
[17:16]	INTSWTRG ADC Sample Module 16,17 Software Force to Start EADC Conversion 0 = No effect. 1 = Cause an EADC conversion when the priority is given to sample module. Note: INTSWTRG is only for internal channels. After writing this register to start EADC conversion, the EADC_PENDSTS register will show which sample module will conversion. If user want to disable the conversion of the sample module, user can write EADC_PENDSTS register to clear it.
[15:4]	Reserved Reserved.
[3:0]	SWTRG ADC Sample Module 0~3, Sample Module 16,17 Software Force to Start EADC Conversion 0 = No effect. 1 = Cause an EADC conversion when the priority is given to sample module. Note: After writing this register to start EADC conversion, the EADC_PENDSTS register will show which sample module will conversion. If user want to disable the conversion of the sample module, user can write EADC_PENDSTS register to clear it.

ADC Sample Module Start of Conversion Pending Flag Register (EADC_PENDSTS)

Register	Offset	R/W	Description	Reset Value
EADC_PENDSTS	EADC_BA+0x58	R/W	ADC Sample Module Start of Conversion Pending Flag Register	0x0000_0000

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

Bits	Description
[31:18]	Reserved Reserved.
[17:16]	INTSTPF ADC Sample Module 17, 16 Start of Conversion Pending Flag Read Operation: 0 = There is no pending conversion for sample module. 1 = Sample module EADC start of conversion is pending. Write Operation: 1 = Clear pending flag and stop conversion for corresponding sample module. Note 1: This bit remains 1 during pending state. When the respective EADC conversion is end, the INTSTPF bit is automatically cleared to 0. Note 2: After stopping current conversion, the corresponding EADC_DATn (n=16,17) keeps its original value.
[15:4]	Reserved Reserved.
[3:0]	STPF ADC Sample Module 0~3 Start of Conversion Pending Flag Read Operation: 0 = There is no pending conversion for sample module. 1 = Sample module EADC start of conversion is pending. Write Operation: 1 = Clear pending flag and stop conversion for corresponding sample module. Note 1: This bit remains 1 during pending state. When the respective EADC conversion is end, the STPFn (n=0~3 bit is automatically cleared to 0. Note 2: After stopping current conversion, the corresponding EADC_DATn (n=0~3) keeps its original value.

ADC Sample Module Start of Conversion Overrun Flag Register (EADC_OVSTS)

Register	Offset	R/W	Description	Reset Value
EADC_OVSTS	EADC_BA+0x5C	R/W	ADC Sample Module Start of Conversion Overrun Flag Register	0x0000_0000

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

Bits	Description
[31:18]	Reserved Reserved.
[17:16]	INTSPOVF ADC SAMPLE16,17 Overrun Flag 0 = No sample module event overrun. 1 = Indicates a new sample module event is generated while an old one event is pending. Note: This bit is cleared by writing 1 to it.
[15:4]	Reserved Reserved.
[3:0]	SPOVF ADC SAMPLE0~3 Overrun Flag 0 = No sample module event overrun. 1 = Indicates a new sample module event is generated while an old one event is pending. Note: This bit is cleared by writing 1 to it.

ADC Sample Module 0~3 Control Registers (EADC_SCTL0~EADC_SCTL3)

Register	Offset	R/W	Description	Reset Value
EADC_SCTL0	EADC_BA+0x80	R/W	ADC Sample Module 0 Control Register	0x0000_0000
EADC_SCTL1	EADC_BA+0x84	R/W	ADC Sample Module 1 Control Register	0x0000_0000
EADC_SCTL2	EADC_BA+0x88	R/W	ADC Sample Module 2 Control Register	0x0000_0000
EADC_SCTL3	EADC_BA+0x8C	R/W	ADC Sample Module 3 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
EXTSMPT							
23	22	21	20	19	18	17	16
Reserved	INTPOS	Reserved	TRGSEL				
15	14	13	12	11	10	9	8
TRGDLYCNT							
7	6	5	4	3	2	1	0
TRGDLYDIV		EXTFEN	EXTREN	CHSEL			

Bits	Description
[31:24]	EXTSMPT EADC Sampling Time Extend When ADC converting at high conversion rate, the sampling time of analog input voltage may not enough if input channel loading is heavy, user can extend ADC sampling time after trigger source is coming to get enough sampling time. EXTSMPT can be set from 0~8'd251.
[23]	Reserved Reserved.
[22]	INTPOS Interrupt Flag Position Select 0 = Set ADIFn (EADC_STATUS2[n], n=0~3) at ADC end of conversion. 1 = Set ADIFn (EADC_STATUS2[n], n=0~3) at ADC start of conversion.
[21]	Reserved Reserved.

Bits	Description
[20:16]	TRGSEL ADC Sample Module Start of Conversion Trigger Source Selection 0H = Disable trigger. 1H = External trigger from EADC0_ST pin input. 2H = EADC ADINT0 interrupt EOC (End of conversion) pulse trigger. 3H = EADC ADINT1 interrupt EOC (End of conversion) pulse trigger. 4H = Timer0 overflow pulse trigger. 5H = Timer1 overflow pulse trigger. 6H = Timer2 overflow pulse trigger. 7H = Timer3 overflow pulse trigger. 8H = Reserved. 9H = Reserved. AH = Reserved. BH = Reserved. CH = Reserved. DH = Reserved. EH = Reserved. FH = Reserved. 10H = Reserved. 11H = Reserved. 12H = Reserved. 13H = Reserved. 14H =BPWM0TG. 15H =BPWM1TG. other = Reserved. Note: Refer to PWM_EADCTS0, PWM_EADCTS1, BPWM_EADCTS0, BPWM_EADCTS1 and TIMERN_CTL (n=0~3) to get more information for PWM, BPWM trigger and timer trigger.
[15:8]	TRGDLYCNT ADC Sample Module Start of Conversion Trigger Delay Time Trigger delay time = TRGDLYCNT x EADC_CLK x n (n=1,2,4,16 from TRGDLYDIV setting). Note: If TRGDLYCNT is set to 1, trigger delay time is actually the same as TRGDLYCNT is set to 2 for hardware operation.
[7:6]	TRGDLYDIV ADC Sample Module Start of Conversion Trigger Delay Clock Divider Selection Trigger delay clock frequency: 00 = EADC_CLK/1. 01 = EADC_CLK/2. 10 = EADC_CLK/4. 11 = EADC_CLK/16.
[5]	EXTFEN ADC External Trigger Falling Edge Enable Bit 0 = Falling edge Disabled when ADC selects EADC0_ST as trigger source. 1 = Falling edge Enabled when ADC selects EADC0_ST as trigger source.
[4]	EXTREN ADC External Trigger Rising Edge Enable Bit 0 = Rising edge Disabled when ADC selects EADC0_ST as trigger source. 1 = Rising edge Enabled when ADC selects EADC0_ST as trigger source.

Bits	Description	
[3:0]	CHSEL	ADC Sample Module Channel Selection 00H = EADC_CH0. 01H = EADC_CH1. 02H = EADC_CH2. 03H = EADC_CH3. 04H = EADC_CH4. 05H = EADC_CH5. 06H = EADC_CH6. 07H = EADC_CH7. 08H = Reserved. 09H = Reserved.. 0AH = Reserved. 0BH = Reserved. 0CH = EADC_CH12. 0DH = EADC_CH13. 0EH = EADC_CH14. 0FH = EADC_CH15. .

ADC Sample Module 16~17 Control Registers (EADC_SCTL16~EADC_SCTL17)

Register	Offset	R/W	Description	Reset Value
EADC_SCTL16	EADC_BA+0xC0	R/W	ADC Sample Module 16 Control Register	0x0000_0000
EADC_SCTL17	EADC_BA+0xC4	R/W	ADC Sample Module 17 Control Register	0x0000_0000

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

Bits	Description
[31:24]	EXTSMPT EADC Sampling Time Extend When ADC converting at high conversion rate, the sampling time of analog input voltage may not enough if input channel loading is heavy, SW can extend ADC sampling time after trigger source is coming to get enough sampling time. EXTSMPT can be set from 0~8'd255. The range of start delay time is from 0~255 EADC clock.
[23:0]	Reserved Reserved.

ADC Interrupt Source Enable Control Registers (EADC_INTSRC0~EADC_INTSRC3)

Register	Offset	R/W	Description	Reset Value
EADC_INTSRC0	EADC_BA+0xD0	R/W	EADC Interrupt 0 Source Enable Control Register.	0x0000_0000
EADC_INTSRC1	EADC_BA+0xD4	R/W	EADC Interrupt 1 Source Enable Control Register.	0x0000_0000
EADC_INTSRC2	EADC_BA+0xD8	R/W	EADC Interrupt 2 Source Enable Control Register.	0x0000_0000
EADC_INTSRC3	EADC_BA+0xDC	R/W	EADC Interrupt 3 Source Enable Control Register.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				SPLIE3	SPLIE2	SPLIE1	SPLIE0

Bits	Description
[31:4]	Reserved
[3]	SPLIE3 Sample Module 3 Interrupt Enable Bit 0 = Sample Module 3 interrupt Disabled. 1 = Sample Module 3 interrupt Enabled.
[2]	SPLIE2 Sample Module 2 Interrupt Enable Bit 0 = Sample Module 2 interrupt Disabled. 1 = Sample Module 2 interrupt Enabled.
[1]	SPLIE1 Sample Module 1 Interrupt Enable Bit 0 = Sample Module 1 interrupt Disabled. 1 = Sample Module 1 interrupt Enabled.
[0]	SPLIE0 Sample Module 0 Interrupt Enable Bit 0 = Sample Module 0 interrupt Disabled. 1 = Sample Module 0 interrupt Enabled.

ADC Result Compare Register 0/1/2/3 (EADC_CMP0/1/2/3)

Register	Offset	R/W	Description	Reset Value
EADC_CMP0	EADC_BA+0xE0	R/W	ADC Result Compare Register 0	0x0000_0000
EADC_CMP1	EADC_BA+0xE4	R/W	ADC Result Compare Register 1	0x0000_0000
EADC_CMP2	EADC_BA+0xE8	R/W	ADC Result Compare Register 2	0x0000_0000
EADC_CMP3	EADC_BA+0xEC	R/W	ADC Result Compare Register 3	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				CMPDAT			
23	22	21	20	19	18	17	16
CMPDAT							
15	14	13	12	11	10	9	8
CMPWEN	Reserved			CMPMCNT			
7	6	5	4	3	2	1	0
CMPSPL					CMPCOND	EADCMPIE	EADCMPIEN

Bits	Description
[31:28]	Reserved Reserved.
[27:16]	CMPDAT Comparison Data The 12 bits data is used to compare with conversion result of specified sample module. User can use it to monitor the external analog input pin voltage transition without imposing a load on software.
[15]	CMPWEN Compare Window Mode Enable Bit 0 = EADCMFP0 (EADC_STATUS2[4]) will be set when EADC_CMP0 compared condition matched. EADCMFP2 (EADC_STATUS2[6]) will be set when EADC_CMP2 compared condition matched 1 = EADCMFP0 (EADC_STATUS2[4]) will be set when both EADC_CMP0 and EADC_CMP1 compared condition matched. EADCMFP2 (EADC_STATUS2[6]) will be set when both EADC_CMP2 and EADC_CMP3 compared condition matched. Note: This bit is only present in EADC_CMP0 and EADC_CMP2 register.
[14:12]	Reserved Reserved.
[11:8]	CMPMCNT Compare Match Count When the specified ADC sample module analog conversion result matches the compare condition defined by CMPCOND (EADC_CMPn[2], n=0~3), the internal match counter will increase 1. If the compare result does not meet the compare condition, the internal compare match counter will reset to 0. When the internal counter reaches the value to (CMPMCNT +1), the EADCMFPn (EADC_STATUS2[7:4], n=0~3) will be set.

Bits	Description	
[7:3]	CMPSPL	Compare Sample Module Selection 00000 = Sample Module 0 conversion result EADC_DAT0 is selected to be compared. 00001 = Sample Module 1 conversion result EADC_DAT1 is selected to be compared. 00010 = Sample Module 2 conversion result EADC_DAT2 is selected to be compared. 00011 = Sample Module 3 conversion result EADC_DAT3 is selected to be compared. 00100 = Sample Module 4 conversion result EADC_DAT4 is selected to be compared. 00101 = Sample Module 5 conversion result EADC_DAT5 is selected to be compared. 00110 = Sample Module 6 conversion result EADC_DAT6 is selected to be compared. 00111 = Sample Module 7 conversion result EADC_DAT7 is selected to be compared. 01000 = Sample Module 8 conversion result EADC_DAT8 is selected to be compared. 01001 = Sample Module 9 conversion result EADC_DAT9 is selected to be compared. 01010 = Sample Module 10 conversion result EADC_DAT10 is selected to be compared. 01011 = Sample Module 11 conversion result EADC_DAT11 is selected to be compared. 01100 = Sample Module 12 conversion result EADC_DAT12 is selected to be compared. 01101 = Sample Module 13 conversion result EADC_DAT13 is selected to be compared. 01110 = Sample Module 14 conversion result EADC_DAT14 is selected to be compared. 01111 = Sample Module 15 conversion result EADC_DAT15 is selected to be compared. 10000 = Sample Module 16 conversion result EADC_DAT16 is selected to be compared. 10001 = Sample Module 17 conversion result EADC_DAT17 is selected to be compared. 10010 = Sample Module 18 conversion result EADC_DAT18 is selected to be compared.
[2]	CMPCOND	Compare Condition 0= Set the compare condition as that when a 12-bit ADC conversion result is less than the 12-bit CMPDAT (EADC_CMPn[27:16]), the internal match counter will increase one. 1= Set the compare condition as that when a 12-bit ADC conversion result is greater or equal to the 12-bit CMPDAT (EADC_CMPn[27:16]), the internal match counter will increase one. Note: When the internal counter reaches the value to (CMPMCNT (EADC_CMPn[11:8], n=0~3) +1), the EADCMPFn bit will be set.
[1]	EADCMPIE	ADC Result Compare Interrupt Enable Bit 0 = Compare function interrupt Disabled. 1 = Compare function interrupt Enabled. If the compare function is enabled and the compare condition matches the setting of CMPCOND (EADC_CMPn[2], n=0~3) and CMPMCNT (EADC_CMPn[11:8], n=0~3), EADCMPFn (EADC_STATUS2[7:4], n=0~3) will be asserted, in the meanwhile, if EADCMPIE is set to 1, a compare interrupt request is generated.
[0]	EADCMPE	ADC Result Compare Enable Bit 0 = Compare Disabled. 1 = Compare Enabled. Set this bit to 1 to enable compare CMPDAT (EADC_CMPn[27:16], n=0~3) with specified sample module conversion result when converted data is loaded into EADC_DAT register.

ADC Status Register 0 (EADC_STATUS0)

Register	Offset	R/W	Description	Reset Value
EADC_STATUS0	EADC_BA+0xF0	R	ADC Status Register 0	0x0000_0000

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

Bits	Description
[31:20]	Reserved Reserved.
[19:16]	OV EADC_DAT0~3 Overrun Flag It is a mirror to OV bit in sample module ADC result data register EADC_DATn. (n=0~3).
[15:4]	Reserved Reserved.
[3:0]	VALID EADC_DAT0~3 Data Valid Flag It is a mirror of VALID bit in sample module ADC result data register EADC_DATn. (n=0~3).

ADC Status Register 1 (EADC_STATUS1)

Register	Offset	R/W	Description	Reset Value
EADC_STATUS1	EADC_BA+0xF4	R	ADC Status Register 1	0x0000_0000

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

Bits	Description
[31:18]	Reserved Reserved.
[17:16]	OV EADC_DAT16~18 Overrun Flag It is a mirror to OV bit in sample module ADC result data register EADC_DATn. (n=16~17).
[15:2]	Reserved Reserved.
[2:0]	VALID EADC_DAT16~17 Data Valid Flag It is a mirror of VALID bit in sample module ADC result data register EADC_DATn. (n=16~17).

ADC Status Register 2 (EADC_STATUS2)

Register	Offset	R/W	Description	Reset Value
EADC_STATUS2	EADC_BA+0xF8	R/W	ADC Status Register 2	0x0011_0000

31	30	29	28	27	26	25	24
Reserved				AOV	AVALID	STOVF	ADOVIF
23	22	21	20	19	18	17	16
BUSY	Reserved			CHANNEL			
15	14	13	12	11	10	9	8
EADCMPO3	EADCMPO2	EADCMPO1	EADCMPO0	ADOVIF3	ADOVIF2	ADOVIF1	ADOVIF0
7	6	5	4	3	2	1	0
EADCMFP3	EADCMFP2	EADCMFP1	EADCMFP0	ADIF3	ADIF2	ADIF1	ADIF0

Bits	Description
[31:28]	Reserved Reserved.
[27]	AOV All Sample Module ADC Result Data Register Overrun Flags Check n=0~3,16,17. 0 = None of sample module data register overrun flag OV (EADC_DATn[16]) is set to 1. 1 = Any one of sample module data register overrun flag OV (EADC_DATn[16]) is set to 1. Note: This bit will keep 1 when any OVN Flag is equal to 1.
[26]	AVALID All Sample Module ADC Result Data Register EADC_DAT Data Valid Flag Check n=0~3,16,17. 0 = None of sample module data register valid flag VALID (EADC_DATn[17]) is set to 1. 1 = Any one of sample module data register valid flag VALID (EADC_DATn[17]) is set to 1. Note: This bit will keep 1 when any VALIDn Flag is equal to 1.
[25]	STOVF All ADC Sample Module Start of Conversion Overrun Flags Check n=0~3,16,17. 0 = None of sample module event overrun flag SPOVF (EADC_OVSTS[n], n=0~3), INTSPOVF (EADC_OVSTS[n], n=16,17) is set to 1. 1 = Any one of sample module event overrun flag SPOVF (EADC_OVSTS[n]) is set to 1. Note: This bit will keep 1 when any SPOVFn or INTSPOVFn Flag is equal to 1.
[24]	ADOVIF All ADC Interrupt Flag Overrun Bits Check n=0~3. 0 = None of ADINT interrupt flag ADOVIFn (EADC_STATUS2[11:8]) is overwritten to 1. 1 = Any one of ADINT interrupt flag ADOVIFn (EADC_STATUS2[11:8]) is overwritten to 1. Note: This bit will keep 1 when any ADOVIFn Flag is equal to 1.
[23]	BUSY ADC Converter Busy/Idle Status (Read Only) 0 = EADC is in idle state. 1 = EADC is busy for sample or conversion. Note: Once a trigger source is coming, this bit must wait 2 EADC_CLK synchronization then the BUSY status will be high. The status will be high to low when the current conversion is finished.

Bits	Description	
[22:21]	Reserved	Reserved.
[20:16]	CHANNEL	Current Conversion Channel (Read Only) This field reflects EADC current conversion channel when BUSY=1. 00H = EADC_CH0. 01H = EADC_CH1. 02H = EADC_CH2. 03H = EADC_CH3. 04H = EADC_CH4. 05H = EADC_CH5. 06H = EADC_CH6. 07H = EADC_CH7. 08H = Reserved. 09H = Reserved. 0AH = Reserved. 0BH = Reserved. 0CH = EADC_CH12. 0DH = EADC_CH13. 0EH = EADC_CH14. 0FH = EADC_CH15. 10H = VBG. 11H = VTEMP.
[15]	EADCMPO3	EADC Compare 3 Output Status The 12 bits compare3 data CMPDAT3 (EADC_CMP3[27:16]) is used to compare with conversion result of specified sample module. User can use it to monitor the external analog input pin voltage status. 0 = Conversion result in EADC_DAT is less than CMPDAT3 setting. 1 = Conversion result in EADC_DAT is greater than or equal to CMPDAT3 setting.
[14]	EADCMPO2	EADC Compare 2 Output Status The 12 bits compare2 data CMPDAT2 (EADC_CMP2[27:16]) is used to compare with conversion result of specified sample module. User can use it to monitor the external analog input pin voltage status. 0 = Conversion result in EADC_DAT is less than CMPDAT2 setting. 1 = Conversion result in EADC_DAT is greater than or equal to CMPDAT2 setting.
[13]	EADCMPO1	EADC Compare 1 Output Status The 12 bits compare1 data CMPDAT1 (EADC_CMP1[27:16]) is used to compare with conversion result of specified sample module. User can use it to monitor the external analog input pin voltage status. 0 = Conversion result in EADC_DAT is less than CMPDAT1 setting. 1 = Conversion result in EADC_DAT is greater than or equal to CMPDAT1 setting.
[12]	EADCMPO0	EADC Compare 0 Output Status The 12 bits compare0 data CMPDAT0 (EADC_CMP0[27:16]) is used to compare with conversion result of specified sample module. User can use it to monitor the external analog input pin voltage status. 0 = Conversion result in EADC_DAT is less than CMPDAT0 setting. 1 = Conversion result in EADC_DAT is greater than or equal to CMPDAT0 setting.
[11]	ADOVIF3	ADC ADINT3 Interrupt Flag Overrun 0 = ADINT3 interrupt flag is not overwritten to 1. 1 = ADINT3 interrupt flag is overwritten to 1. Note: This bit is cleared by writing 1 to it.

Bits	Description	
[10]	ADOVIF2	ADC ADINT2 Interrupt Flag Overrun 0 = ADINT2 interrupt flag is not overwritten to 1. 1 = ADINT2 interrupt flag is overwritten to 1. Note: This bit is cleared by writing 1 to it.
[9]	ADOVIF1	ADC ADINT1 Interrupt Flag Overrun 0 = ADINT1 interrupt flag is not overwritten to 1. 1 = ADINT1 interrupt flag is overwritten to 1. Note: This bit is cleared by writing 1 to it.
[8]	ADOVIF0	ADC ADINT0 Interrupt Flag Overrun 0 = ADINT0 interrupt flag is not overwritten to 1. 1 = ADINT0 interrupt flag is overwritten to 1. Note: This bit is cleared by writing 1 to it.
[7]	EADCMFPF3	EADC Compare 3 Flag When the specific sample module ADC conversion result meets setting condition in EADC_CMP3 then this bit is set to 1. 0 = Conversion result in EADC_DAT does not meet EADC_CMP3 register setting. 1 = Conversion result in EADC_DAT meets EADC_CMP3 register setting. Note: This bit is cleared by writing 1 to it.
[6]	EADCMFPF2	EADC Compare 2 Flag When the specific sample module ADC conversion result meets setting condition in EADC_CMP2 then this bit is set to 1. 0 = Conversion result in EADC_DAT does not meet EADC_CMP2 register setting. 1 = Conversion result in EADC_DAT meets EADC_CMP2 register setting. Note: This bit is cleared by writing 1 to it.
[5]	EADCMFPF1	EADC Compare 1 Flag When the specific sample module ADC conversion result meets setting condition in EADC_CMP1 then this bit is set to 1. 0 = Conversion result in EADC_DAT does not meet EADC_CMP1 register setting. 1 = Conversion result in EADC_DAT meets EADC_CMP1 register setting. Note: This bit is cleared by writing 1 to it.
[4]	EADCMFPF0	EADC Compare 0 Flag When the specific sample module ADC conversion result meets setting condition in EADC_CMP0 then this bit is set to 1. 0 = Conversion result in EADC_DAT does not meet EADC_CMP0 register setting. 1 = Conversion result in EADC_DAT meets EADC_CMP0 register setting. Note: This bit is cleared by writing 1 to it.
[3]	ADIF3	ADC ADINT3 Interrupt Flag 0 = No ADINT3 interrupt pulse received. 1 = ADINT3 interrupt pulse has been received. Note 1: This bit is cleared by writing 1 to it. Note 2: This bit indicates whether an ADC conversion of specific sample module has been completed

Bits	Description	
[2]	ADIF2	ADC ADINT2 Interrupt Flag 0 = No ADINT2 interrupt pulse received. 1 = ADINT2 interrupt pulse has been received. Note 1: This bit is cleared by writing 1 to it. Note 2: This bit indicates whether an ADC conversion of specific sample module has been completed
[1]	ADIF1	ADC ADINT1 Interrupt Flag 0 = No ADINT1 interrupt pulse received. 1 = ADINT1 interrupt pulse has been received. Note 1: This bit is cleared by writing 1 to it. Note 2: This bit indicates whether an ADC conversion of specific sample module has been completed
[0]	ADIF0	ADC ADINT0 Interrupt Flag 0 = No ADINT0 interrupt pulse received. 1 = ADINT0 interrupt pulse has been received. Note 1: This bit is cleared by writing 1 to it. Note 2: This bit indicates whether an ADC conversion of specific sample module has been completed

ADC Status Register 3 (EADC_STATUS3)

Register	Offset	R/W	Description	Reset Value
EADC_STATUS3	EADC_BA+0xFC	R	ADC Status Register 3	0x0000_001F

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

Bits	Description
[31:5]	Reserved Reserved.
[4:0]	CURSPL EADC Current Sample Module (Read Only) This register shows the current EADC is controlled by which sample module control logic modules. If the EADC is Idle, the bit filed will be set to 0x1F.

EADC Power Management Control Register (EADC_PWRCTL)

Register	Offset	R/W	Description	Reset Value
EADC_PWRCTL	EADC_BA+0x110	R/W	EADC Power Management Control Register	0x00f0_c000

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

Bits	Description
[31:24]	Reserved Reserved.
[23:20]	AUTOPDTHT Auto Power Down Threshold Time Auto Power Down Threshold Time = (1/EADC_CLK) x AUTOPDTHT. 0000 to 0110 = Reserved. 0111 = 8 EADC clock for power down threshold time. 1000 = 16 EADC clock for power down threshold time. 1001 = 32 EADC clock for power down threshold time. 1010 = 64 EADC clock for power down threshold time. 1011 = 128 EADC clock for power down threshold time. 1100 = 256 EADC clock for power down threshold time. Others = 256 EADC clock for power down threshold time.
[19:8]	STUPT EADC Start-up Time Set this bit fields to adjust start-up time. The minimum start-up time of EADC is 10us. EADC start-up time = (1/EADC_CLK) x STUPT.
[7:6]	Reserved Reserved.
[5]	AUTOFF Auto Off Mode 0 = Auto off function Disabled. 1 = Auto off function Enabled. When AUTOFF is set to 1, EADC will be powered off automatically to save power.
[4:1]	Reserved Reserved.
[0]	READY EADC Start-up Completely and Ready for Conversion (Read Only) 0 = Power-on sequence is still in progress. 1 = EADC is ready for conversion.

EADC PDMA Control Register (EADC_PDMACTL)

Register	Offset	R/W	Description	Reset Value
EADC_PDMACTL	EADC_BA+0x130	R/W	EADC PDMA Control Register	0x0000_0000

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

Bits	Description
[31:18]	Reserved Reserved.
[17:16]	INTPDMATEN PDMA Transfer Enable Bit for Internal Channel When EADC conversion is completed, the converted data is loaded into EADC_DATn (n:17, 16) register, user can enable this bit to generate a PDMA data transfer request. 0 = PDMA data transfer Disabled. 1 = PDMA data transfer Enabled.
[15:4]	Reserved Reserved.
[3:0]	PDMATEN PDMA Transfer Enable Bit for External Channel When EADC conversion is completed, the converted data is loaded into EADC_DATn (n:0 ~ 3) register, user can enable this bit to generate a PDMA data transfer request. 0 = PDMA data transfer Disabled. 1 = PDMA data transfer Enabled.

EADC Sample Module 0~3 Control Registers1 (EADC_M0CTL1~EADC_M3CTL1)

Register	Offset	R/W	Description	Reset Value
EADC_M0CTL1	EADC_BA+0x140	R/W	EADC Sample Module0 Control Register 1	0x0000_0000
EADC_M1CTL1	EADC_BA+0x144	R/W	EADC Sample Module1 Control Register 1	0x0000_0000
EADC_M2CTL1	EADC_BA+0x148	R/W	EADC Sample Module2 Control Register 1	0x0000_0000
EADC_M3CTL1	EADC_BA+0x14C	R/W	EADC Sample Module3 Control Register 1	0x0000_0000

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

Bits	Description
[31:8]	Reserved Reserved.
[7:4]	ACU Number of Accumulated Conversion Results Selection 0000 = 1 conversion result will be accumulated. 0001 = 2 conversion result will be accumulated. 0010 = 4 conversion result will be accumulated. 0011 = 8 conversion result will be accumulated. 0100 = 16 conversion result will be accumulated. 0101 = 32 conversion result will be accumulated. 0110 = 64 conversion result will be accumulated. 0111 = 128 conversion result will be accumulated. 1000 = 256 conversion result will be accumulated. Others = Reserved.
[3:2]	Reserved Reserved.
[1]	AVG Average Mode Selection 0 = Conversion results will be stored in data register without averaging. 1 = Conversion results in data register will be averaged. Note: This bit needs to work with ACU (EADC_MnCTL1[7:4], n=0~15).
[0]	ALIGN Alignment Selection 0 = The conversion result will be right aligned in data register. 1 = The conversion result will be left aligned in data register.

ADC Result Offset Cancellation Register (EADC_OFFSETCAL)

Register	Offset	R/W	Description	Reset Value
EADC_OFFSETCAL	EADC_BA+0x208	R/W	ADC Result Offset Cancellation Register	0x0000_001f

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

Bits	Description	
[31:5]	Reserved	Reserved.
[4:0]	OFFSETCANCEL	ADC Offset Cancellation Trim Bits When CALEN(EADC_CTL[8]) is set to 1, the offset cancellation trim bits will compensate ADC result offset. When this bit is set to 0, the offset cancellation trim bits have no effect to ADC result. Note 1: These 5 bits trim value wouldn't latched into EADC_OFFSETCAL automatically when Flash initialization. User must read DCR2 by ISP command first, then write the value to OFFSETCANCEL. Note 2: OFFSETCANCEL is signed format. OFFSETCANCEL will sign extension to 12 bit by hardware to perform signed addition with EADC conversion result if CALEN is enabled.

6.22 CRC Controller (CRC)

6.22.1 Overview

The Cyclic Redundancy Check (CRC) generator can perform CRC calculation with four common polynomials CRC-CCITT, CRC-8, CRC-16, and CRC-32 settings.

6.22.2 Features

- Supports four common polynomials CRC-CCITT, CRC-8, CRC-16, and CRC-32
 - CRC-CCITT: $X^{16} + X^{12} + X^5 + 1$
 - CRC-8: $X^8 + X^2 + X + 1$
 - CRC-16: $X^{16} + X^{15} + X^2 + 1$
 - CRC-32: $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
- Programmable seed value
- Supports programmable order reverse setting for input data and CRC checksum
- Supports programmable 1's complement setting for input data and CRC checksum
- Supports 8/16/32-bit of data width
 - 8-bit write mode: 1-AHB clock cycle operation
 - 16-bit write mode: 2-AHB clock cycle operation
 - 32-bit write mode: 4-AHB clock cycle operation
- Supports using PDMA to write data to perform CRC operation

6.22.3 Block Diagram

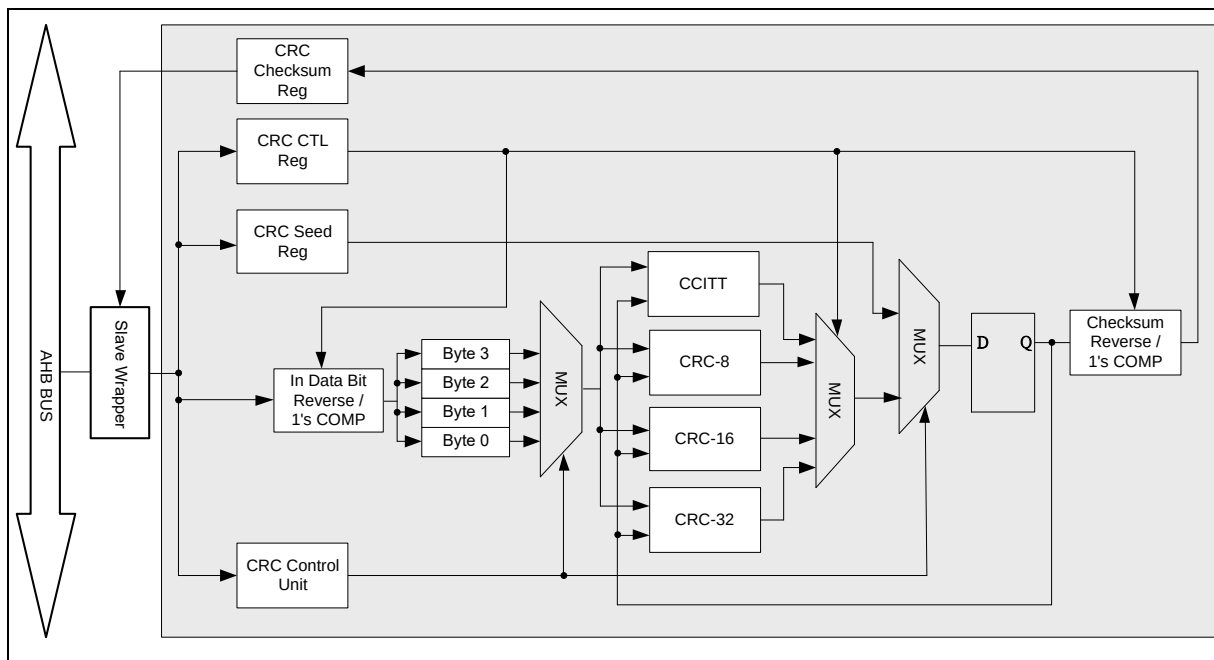


Figure 6.22-1 CRC Generator Block Diagram

6.22.4 Basic Configuration

- Clock Source Configuration

- Enable CRC peripheral clock in CRCKEN (CLK_AHBCLK[7]).
- Reset Configuration
 - Reset CRC controller in CRCRST (SYS_IPRST0[7]).

6.22.5 Functional Description

CRC generator can perform CRC calculation with four common polynomial settings. The operation polynomial includes CRC-CCITT, CRC-8, CRC-16 and CRC-32; User can choose the CRC operation polynomial mode by setting CRCMODE[1:0] (CRC_CTL[31:30] CRC Polynomial Mode).

The following is a program sequence example.

1. Enable CRC generator by setting CRCEN (CRC_CTL[0] CRC Channel Enable Bit).
 - 1) Initial setting for CRC calculation.
2. Configure 1's complement for CRC checksum by setting CHKSFMT (CRC_CTL[27] Checksum 1's Complement).
 - 1) Configure bit order reverse for CRC checksum by setting CHKSREV (CRC_CTL[25] Checksum Bit Order Reverse). The functional block is also shown in Figure 6.22-2 CHECKSUM Bit Order Reverse Functional Block
 - 2) .
- 3) Configure 1's complement for CRC write data by setting DATFMT (CRC_CTL[26] Write Data 1's Complement).
- 4) Configure bit order reverse for CRC write data per byte by setting DATREV (CRC_CTL[24] Write Data Bit Order Reverse). The functional block is also shown in Figure 6.22-3.
3. Perform CHKSINIT (CRC_CTL[1] Checksum Initialization) to load the initial checksum value from CRC_SEED register value.
4. Write data to CRC_DAT register to calculate CRC checksum.
5. Get the CRC checksum result by reading CRC_CHECKSUM register.

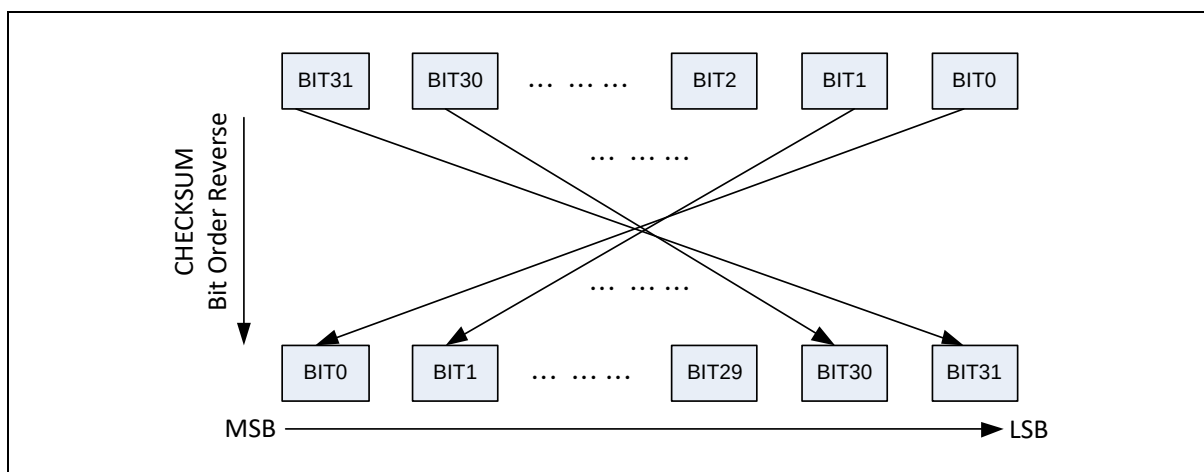


Figure 6.22-2 CHECKSUM Bit Order Reverse Functional Block

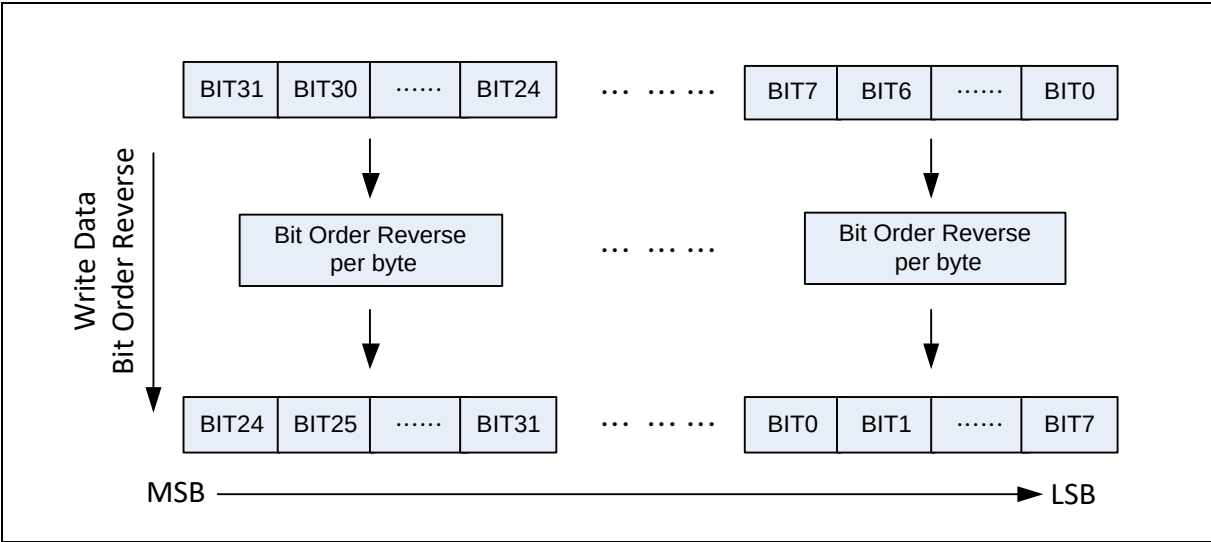


Figure 6.22-3 Write Data Bit Order Reverse Functional Block

6.22.6 Register Map

R: read only, W: write only, R/W: both read and write

Register	Offset	R/W	Description	Reset Value
CRC Base Address: CRC_BA = 0x4003_1000				
CRC_CTL	CRC_BA+0x00	R/W	CRC Control Register	0x2000_0000
CRC_DAT	CRC_BA+0x04	R/W	CRC Write Data Register	0x0000_0000
CRC_SEED	CRC_BA+0x08	R/W	CRC Seed Register	0xFFFF_FFFF
CRC_CHECKSUM	CRC_BA+0x0C	R	CRC Checksum Register	0xFFFF_FFFF

6.22.7 Register Description

CRC Control Register (CRC_CTL)

Register	Offset	R/W	Description	Reset Value
CRC_CTL	CRC_BA+0x00	R/W	CRC Control Register	0x2000_0000

31	30	29	28	27	26	25	24
CRCMODE		DATLEN		CHKSFMT	DATFMT	CHKSREV	DATREV
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved						CHKSINIT	CRCEEN

Bits	Description
[31:30] CRCMODE	CRC Polynomial Mode This field indicates the CRC operation polynomial mode. 00 = CRC-CCITT Polynomial mode. 01 = CRC-8 Polynomial mode. 10 = CRC-16 Polynomial mode. 11 = CRC-32 Polynomial mode.
[29:28] DATLEN	CPU Write Data Length This field indicates the write data length. 00 = Data length is 8-bit mode. 01 = Data length is 16-bit mode. 1x = Data length is 32-bit mode. Note: When the write data length is 8-bit mode, the valid data in CRC_DAT register is only DATA[7:0] bits; if the write data length is 16-bit mode, the valid data in CRC_DAT register is only DATA[15:0].
[27] CHKSFMT	Checksum 1's Complement This bit is used to enable the 1's complement function for checksum result in CRC_CHECKSUM register. 0 = 1's complement for CRC checksum Disabled. 1 = 1's complement for CRC checksum Enabled.
[26] DATFMT	Write Data 1's Complement This bit is used to enable the 1's complement function for write data value in CRC_DAT register. 0 = 1's complement for CRC writes data in Disabled. 1 = 1's complement for CRC writes data in Enabled.
[25] CHKSREV	Checksum Bit Order Reverse This bit is used to enable the bit order reverse function for checksum result in CRC_CHECKSUM register. 0 = Bit order reverse for CRC checksum Disabled. 1 = Bit order reverse for CRC checksum Enabled. Note: If the checksum result is 0xDD7B0F2E, the bit order reverse for CRC checksum is 0x74F0DEBB.

[24]	DATREV	Write Data Bit Order Reverse This bit is used to enable the bit order reverse function per byte for write data value in CRC_DAT register. 0 = Bit order reversed for CRC write data in Disabled. 1 = Bit order reversed for CRC write data in Enabled (per byte). Note: If the write data is 0xAABBCCDD, the bit order reverse for CRC write data in is 0x55DD33BB.
[23:2]	Reserved	Reserved.
[1]	CHKSINIT	Checksum Initialization 0 = No effect. 1 = Initial checksum value by auto reload CRC_SEED register value to CRC_CHECKSUM register value. Note: This bit will be cleared automatically.
[0]	CRCEN	CRC Channel Enable Bit 0 = No effect. 1 = CRC operation Enabled.

CRC Write Data Register (CRC_DAT)

Register	Offset	R/W	Description	Reset Value
CRC_DAT	CRC_BA+0x04	R/W	CRC Write Data Register	0x0000_0000

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

Bits	Description
[31:0]	CRC Write Data Bits User can write data directly by CPU mode or use PDMA function to write data to this field to perform CRC operation. Note: When the write data length is 8-bit mode, the valid data in CRC_DAT register is only DATA[7:0] bits; if the write data length is 16-bit mode, the valid data in CRC_DAT register is only DATA[15:0].

CRC Seed Register (CRC_SEED)

Register	Offset	R/W	Description	Reset Value
CRC_SEED	CRC_BA+0x08	R/W	CRC Seed Register	0xFFFF_FFFF

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

Bits	Description
[31:0]	CRC Seed Value This field indicates the CRC seed value. Note: This field will be reloaded as checksum initial value (CRC_CHECKSUM register) after perform CHKSINIT (CRC_CTL[1]).

CRC Checksum Register (CRC_CHECKSUM)

Register	Offset	R/W	Description	Reset Value
CRC_CHECKSUM	CRC_BA+0x0C	R	CRC Checksum Register	0xFFFF_FFFF

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

Bits	Description	
[31:0]	CHECKSUM	CRC Checksum Results This field indicates the CRC checksum result.

6.23 Peripherals Interconnection

6.23.1 Overview

Some peripherals have interconnections which allow autonomous communication or synchronous action between peripherals without needing to involve the CPU. Peripherals interact without CPU saves CPU resources, reduces power consumption, operates with no software latency and fast responds.

6.23.2 Peripherals Interconnect Matrix table

Source	Destination			
	BPWM	EADC	HIRC TRIM	Timer
BPWM	-	5	-	-
Internal Module	-	2, 3	-	-
External Pin	-	4	-	-
LIRC	-	-	-	8
LXT	-	-	7	-
Timer	1	6	-	-
USB 1.1 Device	-		7	9

Table 6.23-1 Peripherals Interconnect Matrix table

6.23.3 Functional Description

6.23.3.1 From Timer to BPWM

Timer Generates Trigger Pulses as BPWM External Clock Source

Timer0 ~ Timer3 time-out interrupt signal or capture interrupt signal to trigger BPWM.

When timer counter value matches the timer compared value or when the TMx_EXT pin edge transition meets setting, timer can generate a trigger pulse by setting described in section 6.7.6.6.

The setting of BPWM clock source is described in section 6.11.3.

6.23.3.2 From Internal Analog Module/External Pin to EADC Input Source

EADC Input Source

EADC input sources comprises external pins EADC_CH0~15, band-gap voltage (VBG), and temperature sensor (VTEMP).

The detailed EADC input sources are described in section 6.21.2 and 6.21.5.

6.23.3.3 From Internal Analog Module/External Pin to EADC Voltage Reference Source

EADC Voltage Reference Source

EADC votltage is referenced from AV_{DD}. The detailed EADC are described in section 6.21.2.

6.23.3.4 *From External Pin to EADC*

External Pin Trigger EADC Conversion

The EADC0_ST pin (edge detected – rising or falling. Falling edge: 2 PCLK clock high + 3 PCLK clock low. Rising edge: 2 PCLK clock low + 3 PCLK clock high.) is used to trigger EADC to start the conversion.

The detailed external pin trigger conditions are described in section 6.21.5.5.

6.23.3.5 *From BPWM to EADC*

BPWM Trigger EADC Conversion

The chip provides two BPWM generators – BPWM0. Each BPWM supports 6 channels of BPWM output or input capture. There is a 12-bit prescaler to support flexible clock to the 16-bit BPWM counter with 16-bit comparator. The BPWM counter supports up, down and up-down counter types. BPWM uses comparator compared with counter to generate events. These events use to generate BPWM pulse, interrupt and trigger signal for EADC to start conversion.

The detailed BPWM trigger conditions are described in section 6.11.5.15.

6.23.3.6 *From Timer to EADC*

Timer Trigger EADC Conversion

The timer controller provides timer time-out interrupt or capture interrupt to trigger EADC. If TRGSSEL (TIMERx_TRGCTL[0]) is 0, time-out interrupt signal is used to trigger EADC. If TRGSSEL (TIMERx_TRGCTL[0]) is 1, capture interrupt signal is used to trigger EADC.

In Timer mode, the signals CAPIF, TIF are used to trigger the EADC to start the conversion.

The detailed Timer trigger conditions are described in section 6.7.6.6.

6.23.3.7 *From LXT and USB 1.1 Device to HIRC TRIM*

Use LXT or USB Synchronous Mode to system auto-trim HIRC circuit

This chip supports auto-trim function: the HIRC trim (48 MHz RC oscillator) and MIRC trim (4.032 MHz RC oscillator), according to the accurate LXT (32.768 kHz crystal oscillator) or internal USB synchronous mode, automatically gets accurate output frequency, 0.25 % deviation within all temperature ranges.

The details of HIRC trim setting are described in section 6.2.8.

6.23.3.8 *From LIRC to Timer Capture Function*

Measure the Time Interval of LIRC clock Speed

Sets the timer capture source from LIRC clock and measures the time interval of the signal by using timer capture function. Users can use the results of time interval to trim LIRC through software.

The details of time capture function setting are described in section 6.7.6.4 and 6.7.6.5.

6.23.3.9 *From USB to Timer Event Counting Function*

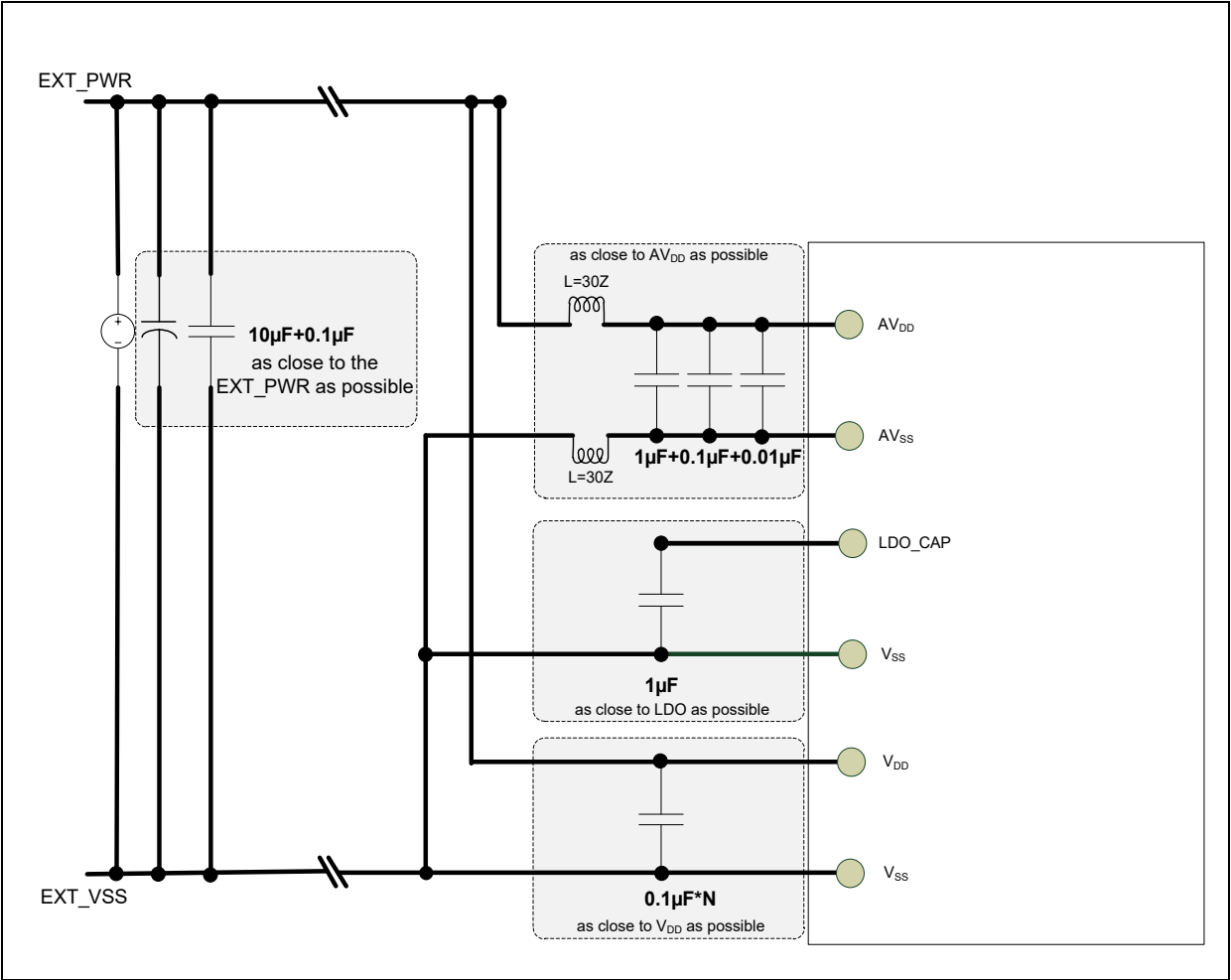
Event Counting Mode

If ECNTSSEL (TIMERx_EXTCTL[16]) is 1, the event counter source will generate by USB device detect the start-of-frame (SOF) packet.

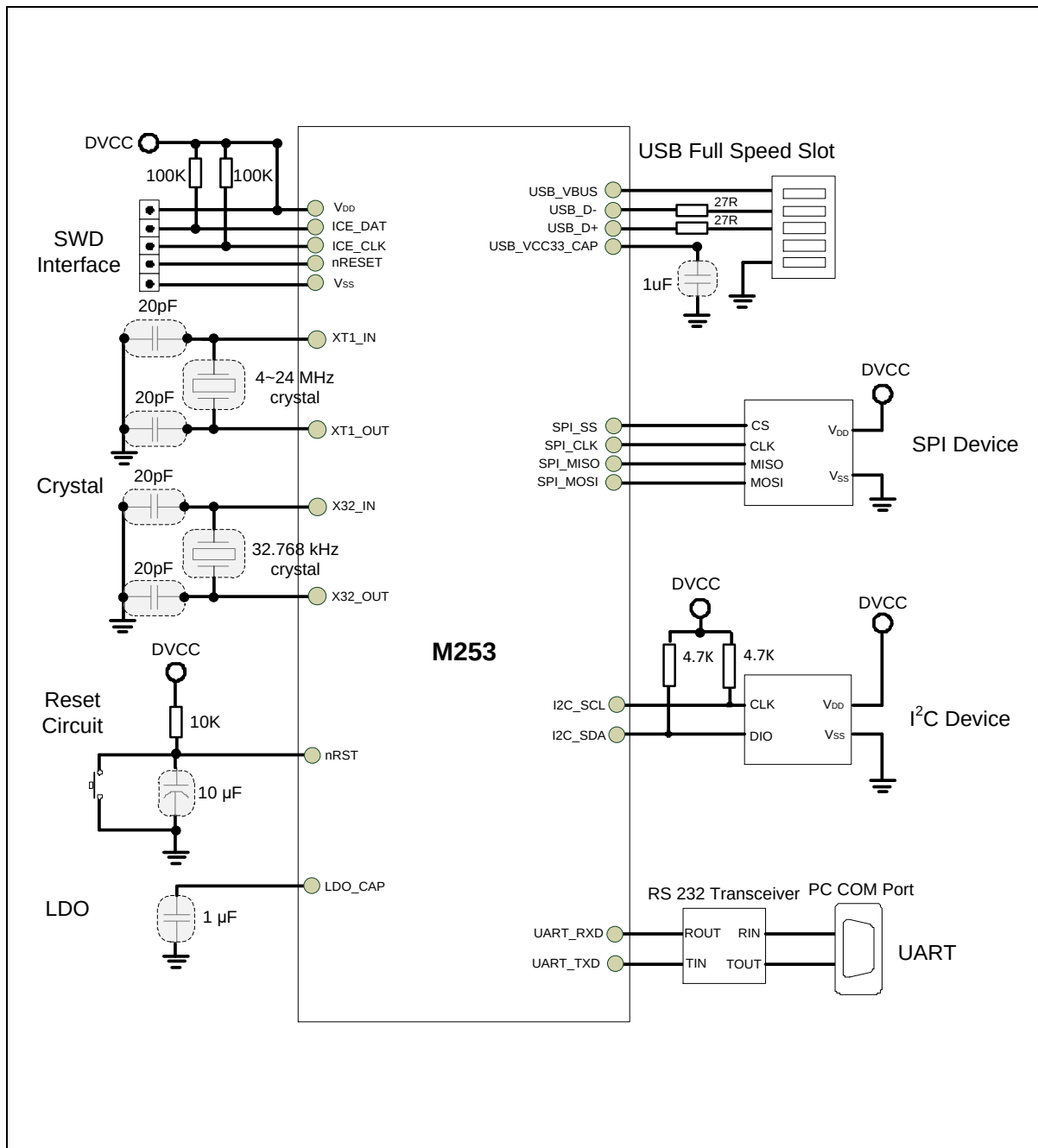
The details of event counting mode are described in section 6.7.4 and 6.7.6.3.

7 APPLICATION CIRCUIT

7.1 Power Supply Scheme



7.2 Peripheral Application Scheme



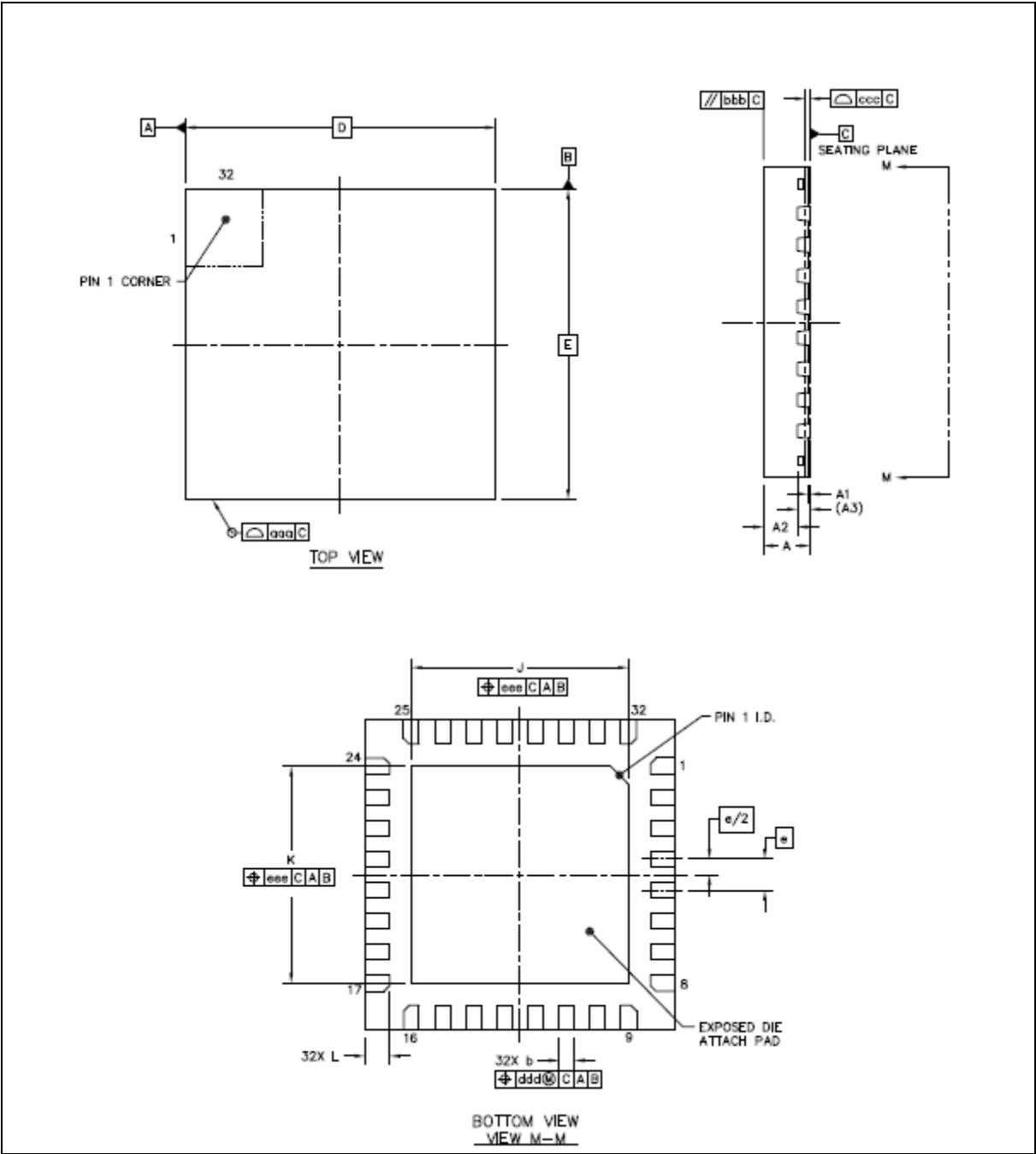
8 ELECTRICAL CHARACTERISTICS

Please refer to the relative Datasheet for detailed information about the M253 electrical characteristics.

9 PACKAGE DIMENSIONS

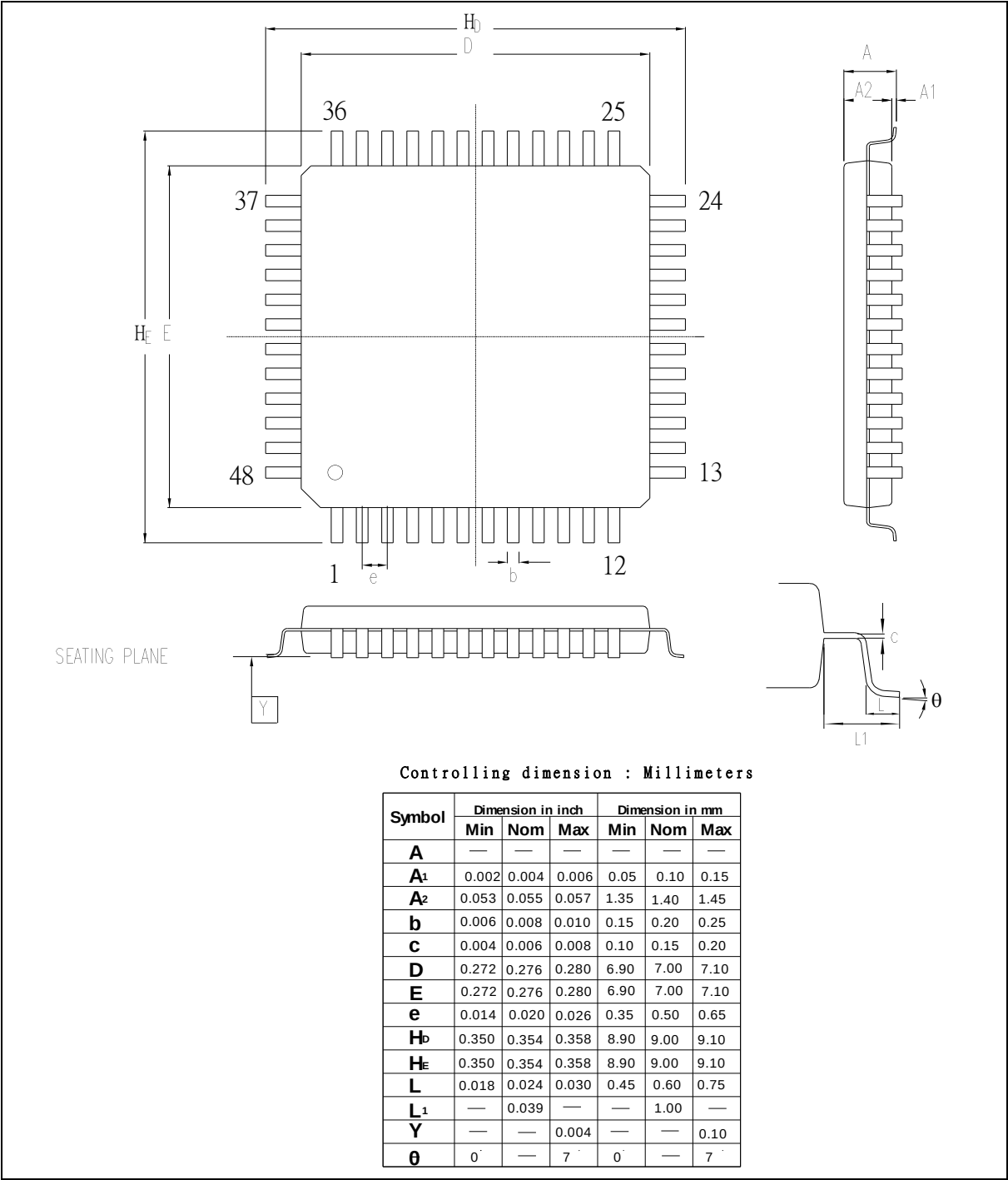
Package is Halogen-free, RoHS-compliant and TSCA-compliant.

9.1 QFN 33L (5x5x0.8 mm)



		SYMBOL	MIN	NOM	MAX
TOTAL THICKNESS		A	0.7	0.75	0.8
STAND OFF		A1	0	0.035	0.05
MOLD THICKNESS		A2	---	0.55	0.57
L/F THICKNESS		A3	0.203 REF		
LEAD WIDTH		b	0.2	0.25	0.3
BODY SIZE	X	D	5 BSC		
	Y	E	5 BSC		
LEAD PITCH		e	0.5 BSC		
EP SIZE	X	J	3.4	3.5	3.6
	Y	K	3.4	3.5	3.6
LEAD LENGTH		L	0.35	0.4	0.45
PACKAGE EDGE TOLERANCE		aaa	0.1		
MOLD FLATNESS		bbb	0.1		
COPLANARITY		ccc	0.08		
LEAD OFFSET		ddd	0.1		
EXPOSED PAD OFFSET		eee	0.1		

9.2 LQFP 48L (7x7x1.4 mm Footprint 2.0 mm)



10 ABBREVIATIONS

10.1 Abbreviations

Acronym	Description
ACMP	Analog Comparator Controller
ADC	Analog-to-Digital Converter
AES	Advanced Encryption Standard
APB	Advanced Peripheral Bus
AHB	Advanced High-Performance Bus
BOD	Brown-out Detection
CAN	Controller Area Network
DAP	Debug Access Port
DES	Data Encryption Standard
EADC	Enhanced Analog-to-Digital Converter
EBI	External Bus Interface
EMAC	Ethernet MAC Controller
EPWM	Enhanced Pulse Width Modulation
FIFO	First In, First Out
FMC	Flash Memory Controller
FPU	Floating-point Unit
GPIO	General-Purpose Input/Output
HCLK	The Clock of Advanced High-Performance Bus
HIRC	12 MHz Internal High Speed RC Oscillator
HXT	4~32 MHz External High Speed Crystal Oscillator
IAP	In Application Programming
ICP	In Circuit Programming
ISP	In System Programming
LDO	Low Dropout Regulator
LIN	Local Interconnect Network
LIRC	10 kHz internal low speed RC oscillator (LIRC)
MPU	Memory Protection Unit
NVIC	Nested Vectored Interrupt Controller
PCLK	The Clock of Advanced Peripheral Bus
PDMA	Peripheral Direct Memory Access
PLL	Phase-Locked Loop
PWM	Pulse Width Modulation

QEI	Quadrature Encoder Interface
SD	Secure Digital
SPI	Serial Peripheral Interface
SPS	Samples per Second
TDES	Triple Data Encryption Standard
TK	Touch Key
TMR	Timer Controller
UART	Universal Asynchronous Receiver/Transmitter
UCID	Unique Customer ID
USB	Universal Serial Bus
WDT	Watchdog Timer
WWDT	Window Watchdog Timer

Table 10.1-1 List of Abbreviations

11 REVISION HISTORY

Date	Revision	Description
2021.06.08	1.00	Initial version.
2023.01.31	1.01	1. Modified the condition of LVR Reset in Table 6.2-6. 2. Modified the description of Power Manager Control Register (CLK_PMUCTL). 3. Added "Package is Halogen-free, RoHS-compliant and TSCA-compliant." in chapter 3 and 9.

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.

*Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.*