

ARM CORTEX<sup>®</sup> -M  
32-BIT MICROCONTROLLER

**NuMicro<sup>®</sup> Family**  
**NUC126 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 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](http://www.nuvoton.com)

**TABLE OF CONTENTS**

**1 GENERAL DESCRIPTION .....19**

    1.1 Key Feature and Application ..... 19

**2 FEATURES.....20**

    2.1 NuMicro® NUC126 Features ..... 20

**3 ABBREVIATIONS .....28**

    3.1 Abbreviations ..... 28

**4 PARTS INFORMATION LIST AND PIN CONFIGURATION .....30**

    4.1 NuMicro® NUC126 Selection Guide ..... 30

        4.1.1 NuMicro® NUC126 Naming Rule ..... 30

        4.1.2 NuMicro® NUC126 USB Series (M452 Compatible) Selection Guide ..... 31

    4.2 Pin Configuration..... 32

        4.2.1 NuMicro® NUC126 USB Series QFN48 Pin Diagram..... 32

        4.2.2 NuMicro® NUC126 USB Series LQFP48 Pin Diagram..... 33

        4.2.3 NuMicro® NUC126 USB Series LQFP64 Pin Diagram..... 34

        4.2.4 NuMicro® NUC126 USB Series LQFP100 Pin Diagram ..... 35

    4.3 Pin Description ..... 36

        4.3.1 NUC126 USB Series Pin Description ..... 36

        4.3.2 GPIO Multi-function Pin Summary ..... 53

**5 BLOCK DIAGRAM .....69**

    5.1 NuMicro® NUC126 Block Diagram ..... 69

**6 FUNCTIONAL DESCRIPTION .....70**

    6.1 ARM® Cortex®-M0 Core ..... 70

    6.2 System Manager ..... 72

        6.2.1 Overview ..... 72

        6.2.2 System Reset ..... 72

        6.2.3 Power Modes and Wake-up Sources..... 79

        6.2.4 System Power Distribution ..... 82

        6.2.5 System Memory Map..... 84

        6.2.6 SRAM Memory Organization..... 86

        6.2.7 Register Lock..... 87

        6.2.8 Auto Trim ..... 87

        6.2.9 UART1\_TXD modulation with PWM ..... 88

        6.2.10 Voltage Detector (VDET) ..... 89

        6.2.11 Register Map ..... 90

        6.2.12 Register Description ..... 92

        6.2.13 System Timer (SysTick) ..... 133

        6.2.14 Nested Vectored Interrupt Controller (NVIC) ..... 138

- 6.2.15 System Control ..... 161
- 6.3 Clock Controller ..... 169
  - 6.3.1 Overview ..... 169
  - 6.3.2 System Clock and SysTick Clock ..... 172
  - 6.3.3 Peripherals Clock ..... 173
  - 6.3.4 Power-down Mode Clock ..... 174
  - 6.3.5 Clock Output ..... 174
  - 6.3.6 Register Map ..... 176
  - 6.3.7 Register Description ..... 177
- 6.4 Flash Memory Controller (FMC) ..... 205
  - 6.4.1 Overview ..... 205
  - 6.4.2 Features ..... 205
  - 6.4.3 Block Diagram ..... 206
  - 6.4.4 Functional Description ..... 207
  - 6.4.5 Register Map ..... 229
  - 6.4.6 Register Description ..... 230
- 6.5 Analog Comparator Controller (ACMP) ..... 247
  - 6.5.1 Overview ..... 247
  - 6.5.2 Features ..... 247
  - 6.5.3 Block Diagram ..... 248
  - 6.5.4 Basic Configuration ..... 249
  - 6.5.5 Functional Description ..... 249
  - 6.5.6 Register Map ..... 254
  - 6.5.7 Register Description ..... 255
- 6.6 Analog-to-Digital Converter (ADC) ..... 262
  - 6.6.1 Overview ..... 262
  - 6.6.2 Features ..... 262
  - 6.6.3 Block Diagram ..... 263
  - 6.6.4 Basic Configuration ..... 263
  - 6.6.5 Functional Description ..... 264
  - 6.6.6 Register Map ..... 272
  - 6.6.7 Register Description ..... 274
- 6.7 CRC Controller (CRC) ..... 289
  - 6.7.1 Overview ..... 289
  - 6.7.2 Features ..... 289
  - 6.7.3 Block Diagram ..... 289
  - 6.7.4 Basic Configuration ..... 290
  - 6.7.5 Functional Description ..... 290
  - 6.7.6 Register Map ..... 292
  - 6.7.7 Register Description ..... 293
- 6.8 External Bus Interface (EBI) ..... 298

- 6.8.1 Overview ..... 298
- 6.8.2 Features ..... 298
- 6.8.3 Block Diagram ..... 299
- 6.8.4 Basic Configuration..... 299
- 6.8.5 Functional Description..... 300
- 6.8.6 Register Map ..... 308
- 6.8.7 Register Description ..... 309
- 6.9 General Purpose I/O (GPIO)..... 313
  - 6.9.1 Overview ..... 313
  - 6.9.2 Features ..... 313
  - 6.9.3 Block Diagram ..... 314
  - 6.9.4 Basic Configuration..... 315
  - 6.9.5 Functional Description..... 315
  - 6.9.6 Register Map ..... 318
  - 6.9.7 Register Description ..... 321
- 6.10 Hardware Divider (HDIV) ..... 336
  - 6.10.1 Overview ..... 336
  - 6.10.2 Features ..... 336
  - 6.10.3 Block Diagram ..... 336
  - 6.10.4 Basic Configuration..... 336
  - 6.10.5 Functional Description..... 336
  - 6.10.6 Register Map ..... 338
  - 6.10.7 Register Description ..... 339
- 6.11 I<sup>2</sup>C Serial Interface Controller (I<sup>2</sup>C) ..... 344
  - 6.11.1 Overview ..... 344
  - 6.11.2 Features ..... 344
  - 6.11.3 Block Diagram ..... 345
  - 6.11.4 Basic Configuration..... 345
  - 6.11.5 Functional Description..... 346
  - 6.11.6 Register Map ..... 362
  - 6.11.7 Register Description ..... 363
- 6.12 PDMA Controller (PDMA) ..... 376
  - 6.12.1 Overview ..... 376
  - 6.12.2 Features ..... 376
  - 6.12.3 Block Diagram ..... 376
  - 6.12.4 Basic Configuration..... 376
  - 6.12.5 Functional Description..... 377
  - 6.12.6 Register Map ..... 384
  - 6.12.7 Register Description ..... 386
- 6.13 PWM Generator and Capture Timer (PWM)..... 413
  - 6.13.1 Overview ..... 413



- 6.13.2 Features ..... 413
- 6.13.3 Block Diagram ..... 415
- 6.13.4 Basic Configuration..... 418
- 6.13.5 Functional Description..... 419
- 6.13.6 Register Map ..... 450
- 6.13.7 Register Description ..... 454
- 6.14 Real Time Clock (RTC) ..... 518
  - 6.14.1 Overview ..... 518
  - 6.14.2 Features ..... 518
  - 6.14.3 Block Diagram ..... 518
  - 6.14.4 Basic Configuration..... 519
  - 6.14.5 Functional Description..... 519
  - 6.14.6 Register Map ..... 525
  - 6.14.7 Register Description ..... 526
- 6.15 Smart Card Host Interface (SC)..... 546
  - 6.15.1 Overview ..... 546
  - 6.15.2 Features ..... 546
  - 6.15.3 Block Diagram ..... 546
  - 6.15.4 Basic Configuration..... 548
  - 6.15.5 Functional Description..... 549
  - 6.15.6 Register Map ..... 559
  - 6.15.7 Register Description ..... 560
- 6.16 Serial Peripheral Interface (SPI)..... 586
  - 6.16.1 Overview ..... 586
  - 6.16.2 Features ..... 586
  - 6.16.3 Block Diagram ..... 587
  - 6.16.4 Basic Configuration..... 587
  - 6.16.5 Functional Description..... 589
  - 6.16.6 Timing Diagram ..... 602
  - 6.16.7 Programming Examples..... 604
  - 6.16.8 Register Map ..... 606
  - 6.16.9 Register Description ..... 607
- 6.17 Timer Controller (TMR) ..... 626
  - 6.17.1 Overview ..... 626
  - 6.17.2 Features ..... 626
  - 6.17.3 Block Diagram ..... 628
  - 6.17.4 Basic Configuration..... 632
  - 6.17.5 Timer Functional Description..... 633
  - 6.17.6 PWM Functional Description ..... 638
  - 6.17.7 Register Map ..... 654
  - 6.17.8 Register Description ..... 659
- 6.18 USB Device Controller (USB)..... 703

- 6.18.1 Overview ..... 703
- 6.18.2 Features ..... 703
- 6.18.3 Block Diagram ..... 704
- 6.18.4 Basic Configuration..... 704
- 6.18.5 Functional Description..... 704
- 6.18.6 Register Map ..... 710
- 6.18.7 Register Description ..... 712
- 6.19 USCI - Universal Serial Control Interface Controller ..... 731
  - 6.19.1 Overview ..... 731
  - 6.19.2 Features ..... 731
  - 6.19.3 Block Diagram ..... 731
  - 6.19.4 Functional Description..... 731
- 6.20 USCI – UART Mode..... 742
  - 6.20.1 Overview ..... 742
  - 6.20.2 Features ..... 742
  - 6.20.3 Block Diagram ..... 742
  - 6.20.4 Basic Configuration..... 743
  - 6.20.5 Functional Description..... 744
  - 6.20.6 Register Map ..... 753
  - 6.20.7 Register Description ..... 754
- 6.21 USCI - SPI Mode..... 775
  - 6.21.1 Overview ..... 775
  - 6.21.2 Features ..... 775
  - 6.21.3 Block Diagram ..... 776
  - 6.21.4 Basic Configuration..... 776
  - 6.21.5 Functional Description..... 778
  - 6.21.6 Register Map ..... 792
  - 6.21.7 Register Description ..... 793
- 6.22 USCI - I<sup>2</sup>C Mode..... 815
  - 6.22.1 Overview ..... 815
  - 6.22.2 Features ..... 815
  - 6.22.3 Block Diagram ..... 816
  - 6.22.4 Basic Configuration..... 816
  - 6.22.5 Functional Description..... 818
  - 6.22.6 Register Map ..... 835
  - 6.22.7 Register Description ..... 836
- 6.23 UART Interface Controller (UART)..... 855
  - 6.23.1 Overview ..... 855
  - 6.23.2 Features ..... 855
  - 6.23.3 Block Diagram ..... 856
  - 6.23.4 Basic Configuration..... 858

6.23.5	Functional Description.....	860
6.23.6	Register Map .....	884
6.23.7	Register Description .....	886
6.24	Watchdog Timer (WDT).....	919
6.24.1	Overview .....	919
6.24.2	Features .....	919
6.24.3	Block Diagram .....	919
6.24.4	Basic Configuration.....	919
6.24.5	Functional Description.....	920
6.24.6	Register Map .....	922
6.24.7	Register Description .....	923
6.25	Window Watchdog Timer (WWDT) .....	927
6.25.1	Overview .....	927
6.25.2	Features .....	927
6.25.3	Block Diagram .....	927
6.25.4	Basic Configuration.....	928
6.25.5	Functional Description.....	928
6.25.6	Register Map .....	931
6.25.7	Register Description .....	932
<b>7</b>	<b>APPLICATION CIRCUIT .....</b>	<b>937</b>
<b>8</b>	<b>PACKAGE DIMENSIONS .....</b>	<b>938</b>
8.1	LQFP 100L (14x14x1.4 mm footprint 2.0 mm).....	938
8.2	LQFP 64L (7x7x1.4 mm footprint 2.0 mm).....	939
8.3	LQFP 48L (7x7x1.4 mm Footprint 2.0 mm) .....	940
8.4	QFN 48L (7x7x0.8 mm).....	941
<b>9</b>	<b>REVISION HISTORY.....</b>	<b>942</b>

**LIST OF FIGURES**

Figure 4.2-1 NuMicro® NUC126 USB Series QFN 48-pin Diagram .....32

Figure 4.2-2 NuMicro® NUC126 USB Series LQFP 48-pin Diagram ..... 33

Figure 4.2-3 NuMicro® NUC126 USB Series LQFP 64-pin Diagram ..... 34

Figure 4.2-4 NuMicro® NUC126 USB Series LQFP 100-pin Diagram ..... 35

Figure 5.1-1 NuMicro® NUC126 Block Diagram.....69

Figure 6.1-1 Functional Block Diagram .....70

Figure 6.2-1 System Reset Sources.....73

Figure 6.2-2 nRESET Reset Waveform .....76

Figure 6.2-3 Power-on Reset (POR) Waveform.....76

Figure 6.2-4 Low Voltage Reset (LVR) Waveform .....77

Figure 6.2-5 Brown-out Detector (BOD) Waveform .....78

Figure 6.2-6 NuMicro® NUC126 Power Mode State Machine.....80

Figure 6.2-7 NuMicro® NUC126 Power Distribution Diagram ..... 83

Figure 6.2-8 SRAM Block Diagram .....86

Figure 6.2-9 SRAM Memory Organization .....87

Figure 6.2-10 UART1\_TXD Modulated with PWM Channel.....88

Figure 6.2-11 VDET Block Diagram .....89

Figure 6.3-1 Clock Generator Block Diagram..... 170

Figure 6.3-2 Clock Generator Global View Diagram ..... 171

Figure 6.3-3 System Clock Block Diagram ..... 172

Figure 6.3-4 HXT Stop Protect Procedure..... 173

Figure 6.3-5 SysTick Clock Control Block Diagram ..... 173

Figure 6.3-6 Clock Source of Clock Output ..... 174

Figure 6.3-7 Clock Output Block Diagram ..... 175

Figure 6.4-1 Flash Memory Controller Block Diagram .....206

Figure 6.4-2 Data Flash .....208

Figure 6.4-3 SPROM Security Mode .....213

Figure 6.4-4 Flash Memory Map .....214

Figure 6.4-5 System Memory Map with IAP Mode .....215

Figure 6.4-6 APROM/LDROM boot with IAP Mode.....216

Figure 6.4-7 VECMAP setting example in IAP Mode .....216

Figure 6.4-8 System Memory Map without IAP mode .....217

Figure 6.4-9 ISP Procedure Example.....220

Figure 6.4-10 Example for accelerating interrupt by VECMAP. ....221

Figure 6.4-11 ISP 32-bit Programming Procedure .....222

Figure 6.4-12 ISP 64-bit Programming Procedure .....223

Figure 6.4-13 Firmware in SRAM for Multi-word Programming .....224

Figure 6.4-14 Multi-word Programming Flow .....225

Figure 6.4-15 CRC-32 Checksum Calculation .....226

Figure 6.4-16 CRC-32 Checksum Calculation Flow.....227

Figure 6.4-17 All-One Verification Flow.....228

Figure 6.5-1 Analog Comparator Block Diagram .....248

Figure 6.5-2 Comparator Hysteresis Function of ACMP0 .....250

Figure 6.5-3 Window Latch Mode of ACMP0 .....250

Figure 6.5-4 Filter Function Example .....251

Figure 6.5-5 Comparator Controller Interrupt .....251

Figure 6.5-6 Comparator Reference Voltage Block Diagram .....252

Figure 6.5-7 Example Connection of Window Compare Mode .....253

Figure 6.5-8 Example of Window Compare Mode .....253

Figure 6.6-1 AD Controller Block Diagram .....263

Figure 6.6-2 ADC Peripheral Clock Control.....265

Figure 6.6-3 Single Mode Conversion Timing Diagram .....266

Figure 6.6-4 Burst Mode Conversion Timing Diagram .....267

Figure 6.6-5 Single-Cycle Scan Mode on Enabled Channels Timing Diagram.....268

Figure 6.6-6 Continuous Scan Mode on Enabled Channels Timing Diagram.....269

Figure 6.6-7 A/D Conversion Result Monitor Logic Diagram .....270

Figure 6.6-8 A/D Controller Interrupt .....271

Figure 6.6-9 Conversion Result Mapping Diagram of ADC Single-end Input .....275

Figure 6.6-10 Conversion Result Mapping Diagram of ADC Differential Input .....276

Figure 6.7-1 CRC Generator Block Diagram.....289

Figure 6.7-2 CHECKSUM Bit Order Reverse Functional Block .....290

Figure 6.7-3 Write Data Bit Order Reverse Functional Block .....291

Figure 6.8-1 EBI Block Diagram .....299

Figure 6.8-2 Connection of 16-bit Device with 16-bit EBI Data Width.....301

Figure 6.8-3 Connection of 8-bit Device with 8-bit EBI Data Width.....302

Figure 6.8-4 Timing Control Waveform for 16-bit Data Width .....304

Figure 6.8-5 Timing Control Waveform for 8-bit Data Width .....305

Figure 6.8-6 Timing Control Waveform for Insert Idle Cycle .....306

Figure 6.8-7 Timing Control Waveform for Continuous Data Access Mode .....307

Figure 6.9-1 GPIO Controller Block Diagram .....314

Figure 6.9-2 GPIO Clock Control Diagram .....315

Figure 6.10-1 Hardware Divider Block Diagram .....336

Figure 6.11-1 I<sup>2</sup>C Controller Block Diagram .....345

Figure 6.11-2 I<sup>2</sup>C Bus Timing ..... 346

Figure 6.11-3 I<sup>2</sup>C Protocol ..... 347

Figure 6.11-4 START and STOP Conditions..... 347

Figure 6.11-5 Bit Transfer on the I<sup>2</sup>C Bus ..... 348

Figure 6.11-6 Acknowledge on the I<sup>2</sup>C Bus..... 348

Figure 6.11-7 Master Transmits Data to Slave by 7-bit..... 349

Figure 6.11-8 Master Reads Data from Slave by 7-bit ..... 349

Figure 6.11-9 Control I<sup>2</sup>C Bus according to the current I<sup>2</sup>C Status ..... 350

Figure 6.11-10 Master Transmitter Mode Control Flow..... 350

Figure 6.11-11 Master Receiver Mode Control Flow..... 351

Figure 6.11-12 Slave Mode Control Flow ..... 352

Figure 6.11-13 GC Mode ..... 353

Figure 6.11-14 Arbitration Lost ..... 354

Figure 6.11-15 Timing of Two-level Buffer Transmit in Master Write ..... 355

Figure 6.11-16 Timing of Two-level Buffer Transmit in Slave Read..... 355

Figure 6.11-17 Setup Time Wrong Adjustment ..... 356

Figure 6.11-18 Hold Time Wrong Adjustment ..... 356

Figure 6.11-19 I<sup>2</sup>C Data Shifting Direction ..... 357

Figure 6.11-20 I<sup>2</sup>C Time-out Count Block Diagram ..... 359

Figure 6.11-21 I<sup>2</sup>C Wake-Up Related Signals Waveform ..... 360

Figure 6.11-22 EEPROM Random Read ..... 361

Figure 6.11-23 Protocol of EEPROM Random Read ..... 361

Figure 6.12-1 PDMA Controller Block Diagram..... 376

Figure 6.12-2 Descriptor Table Entry Structure..... 377

Figure 6.12-3 Descriptor Table Operation in Basic Mode ..... 378

Figure 6.12-4 Descriptor Table Link List Structure ..... 380

Figure 6.12-5 Scatter-Gather Mode Finite State Machine..... 380

Figure 6.12-6 Example of Single Transfer Type and Burst Transfer Type in Basic Mode ..... 382

Figure 6.12-7 Example of PDMA Channel 0 Time-out Counter Operation ..... 383

Figure 6.13-1 PWM Generator Overview Block Diagram..... 415

Figure 6.13-2 PWM System Clock Source Control ..... 415

Figure 6.13-3 PWM Clock Source Control ..... 416

Figure 6.13-4 PWM Independent Mode Architecture Diagram ..... 417

Figure 6.13-5 PWM Complementary Mode Architecture Diagram ..... 418

Figure 6.13-6 PWM0\_CH0 Prescaler Waveform in Up Counter Type ..... 420

Figure 6.13-7 PWM Up Counter Type ..... 420

Figure 6.13-8 PWM Down Counter Type ..... 421

Figure 6.13-9 PWM Up-Down Counter Type..... 421

Figure 6.13-10 PWM Compared point Events in Up-Down Counter Type..... 422

Figure 6.13-11 PWM Double Buffering Illustration ..... 423

Figure 6.13-12 Period Loading in Up-Count Mode..... 424

Figure 6.13-13 Immediately Loading in Up-Count Mode..... 425

Figure 6.13-14 Window Loading in Up-Count Mode ..... 426

Figure 6.13-15 Center Loading in Up-Down-Count Mode..... 427

Figure 6.13-16 PWM One-shot Mode Output Waveform ..... 428

Figure 6.13-17 PWM Pulse Generation..... 429

Figure 6.13-18 PWM 0% to 100% Pulse Generation ..... 429

Figure 6.13-19 PWM Independent Mode Waveform ..... 430

Figure 6.13-20 PWM Complementary Mode Waveform ..... 431

Figure 6.13-21 PWM Group Function Waveform ..... 432

Figure 6.13-22 PWM SYNC\_IN Noise Filter Block Diagram ..... 432

Figure 6.13-23 PWM Counter Synchronous Function Block Diagram ..... 433

Figure 6.13-24 PWM Synchronous Function with Synchronize source from SYNC\_IN Signal ... 434

Figure 6.13-25 PWMx\_CH0 Output Control in Independent Mode ..... 434

Figure 6.13-26 PWMx\_CH0 and PWMx\_CH1 Output Control in Complementary Mode ..... 435

Figure 6.13-27 Dead-Time Insertion..... 435

Figure 6.13-28 Illustration of Mask Control Waveform ..... 436

Figure 6.13-29 Brake Noise Filter Block Diagram ..... 437

Figure 6.13-30 Brake Block Diagram for PWMx\_CH0 and PWMx\_CH1 Pair ..... 438

Figure 6.13-31 Edge Detector Waveform for PWMx\_CH0 and PWMx\_CH1 Pair ..... 439

Figure 6.13-32 Level Detector Waveform for PWMx\_CH0 and PWMx\_CH1 Pair ..... 439

Figure 6.13-33 Brake Source Block Diagram ..... 440

Figure 6.13-34 Brake System Fail Block Diagram ..... 440

Figure 6.13-35 PWM LEB Function Waveform ..... 441

Figure 6.13-36 Initial State and Polarity Control with Rising Edge Dead-Time Insertion..... 442

Figure 6.13-37 PWMx\_CH0 and PWMx\_CH1 Pair Accumulate Interrupt Waveform ..... 443

Figure 6.13-38 PWMx\_CH0 and PWMx\_CH1 Pair Interrupt Architecture Diagram..... 444

Figure 6.13-39 PWMx\_CH0 and PWMx\_CH1 Pair Trigger ADC Block Diagram ..... 445

Figure 6.13-40 PWM Trigger ADC in Up-Down Counter Type Timing Waveform ..... 446

Figure 6.13-41 PWMx\_CH0 Capture Block Diagram ..... 447

Figure 6.13-42 Capture Operation Waveform ..... 448

Figure 6.13-43 Capture PDMA Operation Waveform of Channel 0 ..... 449

Figure 6.14-1 RTC Block Diagram ..... 519

Figure 6.14-2 Backup I/O Control Diagram ..... 524

Figure 6.15-1 SC Clock Control Diagram .....547

Figure 6.15-2 SC Controller Block Diagram .....547

Figure 6.15-3 SC Data Character .....549

Figure 6.15-4 SC Activation Sequence .....550

Figure 6.15-5 SC Warm Reset Sequence .....551

Figure 6.15-6 SC Deactivation Sequence .....552

Figure 6.15-7 Basic Operation Flow .....553

Figure 6.15-8 Initial Character TS .....554

Figure 6.15-9 SC Error Signal .....554

Figure 6.15-10 Transmit Direction Block Guard Time Operation .....557

Figure 6.15-11 Receive Direction Block Guard Time Operation .....557

Figure 6.15-12 Extra Guard Time Operation .....558

Figure 6.16-1 SPI Block Diagram (SPI0/1) .....587

Figure 6.16-2 SPI Peripheral Clock .....589

Figure 6.16-3 SPI Full-Duplex Master Mode Application Block Diagram .....590

Figure 6.16-4 SPI Full-Duplex Slave Mode Application Block Diagram .....590

Figure 6.16-5 32-Bit in One Transaction .....591

Figure 6.16-6 Automatic Slave Selection (SSACTPOL = 0, SUSPITV > 0x2) .....592

Figure 6.16-7 Automatic Slave Selection (SSACTPOL = 0, SUSPITV < 0x3) .....592

Figure 6.16-8 Byte Reorder Function .....593

Figure 6.16-9 Timing Waveform for Byte Suspend .....593

Figure 6.16-10 SPI Half-Duplex Master Mode Application Block Diagram .....594

Figure 6.16-11 SPI Half-Duplex Slave Mode Application Block Diagram .....594

Figure 6.16-12 FIFO Threshold Comparator .....595

Figure 6.16-13 Transmit FIFO Buffer Example .....596

Figure 6.16-14 Receive FIFO Buffer Example .....597

Figure 6.16-15 TX Underflow Event and Slave Under Run Event .....597

Figure 6.16-16 Slave Mode Bit Count Error .....598

Figure 6.16-17 I<sup>2</sup>S Data Format Timing Diagram .....599

Figure 6.16-18 MSB Justified Data Format Timing Diagram .....600

Figure 6.16-19 PCM Mode A Timing Diagram .....600

Figure 6.16-20 PCM Mode B Timing Diagram .....600

Figure 6.16-21 FIFO Contents for Various I<sup>2</sup>S Modes .....601

Figure 6.16-22 SPI Timing in Master Mode .....602

Figure 6.16-23 SPI Timing in Master Mode (Alternate Phase of SPIx\_CLK) .....602

Figure 6.16-24 SPI Timing in Slave Mode .....603

Figure 6.16-25 SPI Timing in Slave Mode (Alternate Phase of SPIx\_CLK) .....603



Figure 6.17-1 Timer Controller Block Diagram .....628

Figure 6.17-2 Clock Source of Timer Controller ..... 629

Figure 6.17-3 PWM Generator Overview Block Diagram.....630

Figure 6.17-4 PWM System Clock Source Control .....630

Figure 6.17-5 PWM Counter Clock Source Control .....631

Figure 6.17-6 PWM Independent Mode Architecture Diagram .....631

Figure 6.17-7 PWM Complementary Mode Architecture Diagram .....632

Figure 6.17-8 Continuous Counting Mode.....635

Figure 6.17-9 External Capture Mode .....636

Figure 6.17-10 External Reset Counter Mode.....636

Figure 6.17-11 Internal Timer Trigger.....637

Figure 6.17-12 Inter-Timer Trigger Capture Timing .....638

Figure 6.17-13 PWM Prescale Waveform in Up Count Type .....638

Figure 6.17-14 PWM Up Count Type .....639

Figure 6.17-15 PWM Down Count Type.....639

Figure 6.17-16 PWM Up-Down Count Type.....640

Figure 6.17-17 PWM Comparator Events in Up-Down Count Type.....641

Figure 6.17-18 Period Loading Mode with Up Count Type .....642

Figure 6.17-19 Immediately Loading Mode with Up Count Type .....643

Figure 6.17-20 PWM Pulse Generation in Up-Down Count Type .....643

Figure 6.17-21 PWM 0% to 100% Duty Cycle in Up Count Type and Up-Down Count Type .....644

Figure 6.17-22 PWM Independent Mode Output Waveform .....645

Figure 6.17-23 PWM Complementary Mode Output Waveform.....645

Figure 6.17-24 PWMx\_CH0 Output Control in Independent Mode .....645

Figure 6.17-25 PWMx\_CH0 and PWMx\_CH1 Output Control in Complementary Mode .....646

Figure 6.17-26 Dead-Time Insertion.....646

Figure 6.17-27 PWM Output Mask Control Waveform.....647

Figure 6.17-28 Brake Pin Noise Filter Block Diagram.....647

Figure 6.17-29 Brake Event Block Diagram for PWMx\_CH0 and PWMx\_CH1 .....648

Figure 6.17-30 Edge Detector Brake Waveform for PWMx\_CH0 and PWMx\_CH1 .....649

Figure 6.17-31 Level Detector Brake Waveform for PWMx\_CH0 and PWMx\_CH1 .....650

Figure 6.17-32 Brake Source Block Diagram .....651

Figure 6.17-33 System Fail Brake Block Diagram .....651

Figure 6.17-34 PWMx\_CH0 and PWMx\_CH1 Polarity Control with Dead-Time Insertion .....652

Figure 6.17-35 PWM Interrupt Architecture Diagram .....653

Figure 6.17-36 PWM Trigger ADC Block Diagram .....653

Figure 6.18-1 USB Block Diagram .....704

Figure 6.18-2 WKIDLE Interrupt Operation Flow .....706

Figure 6.18-3 Endpoint SRAM Structure .....707

Figure 6.18-4 Setup Transaction Followed by Data IN Transaction .....707

Figure 6.18-5 Data Out Transfer .....708

Figure 6.19-1 USCI Block Diagram .....731

Figure 6.19-2 Input Conditioning for USC1x\_DAT[1:0] and USC1x\_CTL[1:0].....732

Figure 6.19-3 Input Conditioning for USC1x\_CLK .....733

Figure 6.19-4 Block Diagram of Data Buffering.....734

Figure 6.19-5 Data Access Structure .....735

Figure 6.19-6 Transmit Data Path .....735

Figure 6.19-7 Receive Data Path .....736

Figure 6.19-8 Protocol-Relative Clock Generator .....737

Figure 6.19-9 Basic Clock Divider Counter .....738

Figure 6.19-10 Block of Timing Measurement Counter.....738

Figure 6.19-11 Sample Time Counter .....739

Figure 6.19-12 Event and Interrupt Structure .....740

Figure 6.20-1 USCI UART Mode Block Diagram .....742

Figure 6.20-2 UART Signal Connection for Full-Duplex Communication .....745

Figure 6.20-3 UART Standard Frame Format.....746

Figure 6.20-4 UART Bit Timing (data sample time) .....748

Figure 6.20-5 UART Auto Baud Rate Control .....749

Figure 6.20-6 Incoming Data Wake-Up.....750

Figure 6.20-7 nCTS Wake-Up Case 1.....750

Figure 6.20-8 nCTS Wake-Up Case 2.....751

Figure 6.21-1 SPI Master Mode Application Block Diagram .....775

Figure 6.21-2 SPI Slave Mode Application Block Diagram .....775

Figure 6.21-3 USCI SPI Mode Block Diagram .....776

Figure 6.21-4 Wire Full-Duplex SPI Communication Signals (Master Mode) .....779

Figure 6.21-5 Wire Full-Duplex SPI Communication Signals (Slave Mode) .....779

Figure 6.21-6 SPI Communication with Different SPI Clock Configuration (SCLKMODE=0x0) ..780

Figure 6.21-7 SPI Communication with Different SPI Clock Configuration (SCLKMODE=0x1) ..781

Figure 6.21-8 SPI Communication with Different SPI Clock Configuration (SCLKMODE=0x2) ..781

Figure 6.21-9 SPI Communication with Different SPI Clock Configuration (SCLKMODE=0x3) ..782

Figure 6.21-10 16-bit data Length in One Word Transaction with MSB First Format .....783

Figure 6.21-11 Word Suspend Interval between Two Transaction Words.....783

Figure 6.21-12 Auto Slave Select (SUSPITV ≥ 0x3) .....784

Figure 6.21-13 Auto Slave Select (SUSPITV < 0x3).....784

Figure 6.21-14 One Output Data Channel Half-duplex (SPI Master Mode) ..... 785

Figure 6.21-15 One Input Data Channel Half-duplex (SPI Master Mode)..... 786

Figure 6.21-16 SPI Timing in Master Mode ..... 788

Figure 6.21-17 SPI Timing in Master Mode (Alternate Phase of Serial Bus Clock) ..... 788

Figure 6.21-18 SPI Timing in Slave Mode ..... 789

Figure 6.21-19 SPI Timing in Slave Mode (Alternate Phase of Serial Bus Clock) ..... 789

Figure 6.22-1 I<sup>2</sup>C Bus Timing ..... 815

Figure 6.22-2 USCI I<sup>2</sup>C Mode Block Diagram ..... 816

Figure 6.22-3 I<sup>2</sup>C Protocol ..... 818

Figure 6.22-4 START and STOP Conditions..... 819

Figure 6.22-5 Bit Transfer on the I<sup>2</sup>C Bus ..... 820

Figure 6.22-6 Acknowledge on the I<sup>2</sup>C Bus..... 820

Figure 6.22-7 Arbitration Lost ..... 821

Figure 6.22-8 Control I<sup>2</sup>C Bus according to Current I<sup>2</sup>C Status..... 824

Figure 6.22-9 Master Transmits Data to Slave with a 7-bit address ..... 825

Figure 6.22-10 Master Reads Data from Slave with a 7-bit address ..... 825

Figure 6.22-11 Master Transmits Data to Slave by 10-bit address ..... 825

Figure 6.22-12 Master Reads Data from Slave by 10-bit address ..... 826

Figure 6.22-13 Master Transmitter Mode Control Flow with 7-bit Address ..... 826

Figure 6.22-14 Master Receiver Mode Control Flow with 7-bit Address ..... 827

Figure 6.22-15 Save Mode Control Flow with 7-bit address ..... 828

Figure 6.22-16 GC Mode with 7-bit address..... 830

Figure 6.22-17 Setup Time Wrong Adjustment ..... 832

Figure 6.22-18 Hold Time Wrong Adjustment ..... 832

Figure 6.22-19 I<sup>2</sup>C Time-out Count Block Diagram ..... 833

Figure 6.22-20 EEPROM Random Read ..... 834

Figure 6.22-21 Protocol of EEPROM Random Read ..... 834

Figure 6.23-1 UART Clock Control Diagram ..... 856

Figure 6.23-2 UART Block Diagram ..... 856

Figure 6.23-3 Auto-Baud Rate Measurement ..... 863

Figure 6.23-4 Transmit Delay Time Operation ..... 864

Figure 6.23-5 UART nCTS Wake-up Case1 ..... 864

Figure 6.23-6 UART nCTS Wake-up Case2 ..... 865

Figure 6.23-7 UART Data Wake-up ..... 865

Figure 6.23-8 UART Received Data FIFO Reached Threshold Wake-up ..... 866

Figure 6.23-9 UART RS-485 AAD Mode Address Match Wake-up ..... 866

Figure 6.23-10 UART Received Data FIFO Threshold Time-out Wake-up..... 867

Figure 6.23-11 Auto-Flow Control Block Diagram .....870

Figure 6.23-12 UART nCTS Auto-Flow Control Enabled .....871

Figure 6.23-13 UART nRTS Auto-Flow Control Enabled .....871

Figure 6.23-14 UART nRTS Auto-Flow with Software Control.....872

Figure 6.23-15 IrDA Control Block Diagram .....872

Figure 6.23-16 IrDA TX/RX Timing Diagram .....873

Figure 6.23-17 Structure of LIN Frame.....874

Figure 6.23-18 Structure of LIN Byte.....874

Figure 6.23-19 Break Detection in LIN Mode .....876

Figure 6.23-20 LIN Frame ID and Parity Format.....876

Figure 6.23-21 LIN Sync Field Measurement.....878

Figure 6.23-22 UART\_BAUD Update Sequence in AR Mode if SLVDUEN is 1 .....879

Figure 6.23-23 UART\_BAUD Update Sequence in AR mode if SLVDUEN is 0 .....879

Figure 6.23-24 RS-485 nRTS Driving Level in Auto Direction Mode .....881

Figure 6.23-25 RS-485 nRTS Driving Level with Software Control.....882

Figure 6.23-26 Structure of RS-485 Frame .....883

Figure 6.24-1 Watchdog Timer Block Diagram .....919

Figure 6.24-2 Watchdog Timer Clock Control .....920

Figure 6.24-3 Watchdog Timer Time-out Interval and Reset Period Timing.....921

Figure 6.25-1 WWDT Block Diagram .....927

Figure 6.25-2 WWDT Clock Control .....928

Figure 6.25-3 WWDT Compare Match Interrupt when CMPDAT is 0x3E.....929

Figure 6.25-4 WWDT Counter Reload and Reset Behavior.....930

**List of Tables**

Table 1.1-1 Key Features Support Table ..... 19

Table 3.1-1 List of Abbreviations ..... 29

Table 4.3-1 NUC126 GPIO Multi-function Table ..... 68

Table 6.2-1 Reset Value of Registers..... 75

Table 6.2-2 Power Mode Difference Table ..... 79

Table 6.2-3 Clocks in Power Modes ..... 81

Table 6.2-4 Condition of Entering Power-down Mode Again ..... 82

Table 6.2-5 Address Space Assignments for On-Chip Controllers ..... 85

Table 6.2-6 Exception Model..... 139

Table 6.2-7 Interrupt Number Table ..... 140

Table 6.3-1 Clock Stable Count Value Table ..... 169

Table 6.3-2 Power-down Mode Control Table..... 179

Table 6.4-1 Boot Source Selection Table..... 218

Table 6.4-2 ISP Command List ..... 219

Table 6.4-3 FMC control registers for Flash Programming ..... 222

Table 6.8-1 EBI Memory Mapping of Bank0 and Bank1 ..... 300

Table 6.8-2 EBI Timing Control Setting Table ..... 303

Table 6.11-1 Relationship between I<sup>2</sup>C Baud Rate and PCLK ..... 356

Table 6.11-2 I<sup>2</sup>C Status Code Description..... 358

Table 6.12-1 Channel Priority Table..... 377

Table 6.13-1 PWM System Clock Source Control Registers Setting Table ..... 416

Table 6.13-2 PWM Pulse Generation Event Priority for Up-Counter ..... 429

Table 6.13-3 PWM Pulse Generation Event Priority for Down-Counter..... 430

Table 6.13-4 PWM Pulse Generation Event Priority for Up-Down-Counter..... 430

Table 6.14-1 RTC Read/Write Access Attribute ..... 520

Table 6.14-2 12/24-Hour Time Scale Selection ..... 521

Table 6.14-3 Registers Default Value After Powered On..... 522

Table 6.14-4 Registers Power Domain..... 523

Table 6.15-1 SC Host Controller Pin Description ..... 548

Table 6.15-2 UART Pin Description ..... 548

Table 6.15-3 Timer0/Timer1/Timer2 Operation Mode ..... 557

Table 6.16-1 SPI/I<sup>2</sup>S Interface Controller Pin ..... 589

Table 6.17-1 PWM Pulse Generation Event Priority in Up Count Type ..... 644

Table 6.17-2 PWM Pulse Generation Event Priority in Down Count Type ..... 644

Table 6.17-3 PWM Pulse Generation Event Priority in Up-Down Count Type..... 644

Table 6.19-1 Input Signals for Different Protocols..... 732

Table 6.19-2 Output Signals for Different Protocols ..... 733

Table 6.19-3 Data Transfer Events and Interrupt Handling ..... 740

Table 6.19-4 Protocol-specific Events and Interrupt Handling ..... 741

Table 6.20-1 Input Signals for UART Protocols ..... 745

Table 6.20-2 Output Signals for Different Protocols ..... 746

Table 6.21-1 SPI Communication Signals ..... 778

Table 6.21-2 Serial Bus Clock Configuration ..... 780

Table 6.22-1 Relationship between I<sup>2</sup>C Baud Rate and PCLK ..... 831

Table 6.23-1 UART Interrupt ..... 858

Table 6.23-2 UART Interface Controller Pin ..... 858

Table 6.23-3 UART Controller Baud Rate Equation Table ..... 860

Table 6.23-4 UART Controller Baud Rate Parameter Setting Example Table ..... 861

Table 6.23-5 UART Controller Baud Rate Register Setting Example Table ..... 861

Table 6.23-6 Baud Rate Compensation Example Table 1 ..... 862

Table 6.23-7 Baud Rate Compensation Example Table 2 ..... 862

Table 6.23-8 UART Controller Interrupt Source and Flag List ..... 869

Table 6.23-9 UART Line Control of Word and Stop Length Setting ..... 869

Table 6.23-10 UART Line Control of Parity Bit Setting ..... 870

Table 6.23-11 LIN Header Selection in Master Mode ..... 875

Table 6.24-1 Watchdog Timer Time-out Interval Period Selection ..... 920

Table 6.25-1 WWDAT Prescale Value Selection and Time-out Period ..... 929

Table 6.25-2 CMPDAT Setting Limitation ..... 930

## 1 GENERAL DESCRIPTION

The NuMicro<sup>®</sup> NUC126 series microcontroller based on the ARM<sup>®</sup> Cortex<sup>®</sup>-M0 core operates at up to 72 MHz. With its crystal-less USB 2.0 FS interface, it is able to generate precise frequency required by USB protocol without the need of external crystal. It features adjustable V<sub>DDIO</sub> pins for specific I/O pins with a wide range of voltage from 1.8V to 5.5V for various operating voltages of external components, a unique high-speed PWM with clock frequency up to 144 MHz for precision control, and an integrated hardware divider to speed up the calculation for the control algorithms. Apart from that, the NUC126 also integrates SPROM (Security Protection ROM) which provides a secure code execution area to protect the intelligent property of developers. The NUC126 series is ideal for industrial control, motor control and metering applications.

The NUC126 series supports the wide voltage range from 2.5V to 5.5V and temperature ranging from -40°C to 105°C, up to 256 Kbytes of Flash memory, 20 Kbytes of SRAM, 4 Kbytes of ISP (In-System Programming) ROM as well as ICP (In-Circuit Programming) ROM and IAP (In-Application Programming) ROM in 48-, 64- or 100-pin packages. It also supports high immunity of 8KV ESD (HBM)/4KV EFT. It is also equipped with plenty of peripherals such as USB interface, Timers, Watchdog Timers, RTC, PDMA, EBI, UART, Smart Card Interface, SPI, I<sup>2</sup>S, I<sup>2</sup>C, GPIO, up to 12 channels of 16-bit PWM, up to 20 channels of 12-bit ADC, analog comparator, temperature sensor, low voltage reset, brown-out detector, 96-bit UID (Unique Identification), and 128-bit UCID (Unique Customer Identification).

### 1.1 Key Feature and Application

Product Line	USB	USCI	UART	I <sup>2</sup> C	SPI/I <sup>2</sup> S	ISO 7816	PWM	EBI	PDMA	ADC	ACMP	RTC V <sub>BAT</sub>	V <sub>DDIO</sub>
NUC126	2.0 FS Device	3	3	2	2	2	12	Y	5	20	2	Y	Y

Table 1.1-1 Key Features Support Table

The NuMicro<sup>®</sup> NUC126 series is suitable for a wide range of applications such as:

- Industrial Automation
- PLCs
- Inverters
- Home Automation
- Security Alarm System
- Power Metering
- Portable Data Collector
- Portable RFID Reader
- System Supervisors
- Smart Card Reader
- Printer
- Bar Code Scanner
- Motor Control
- Digital Power

## 2 FEATURES

### 2.1 NuMicro® NUC126 Features

- Core
  - ARM® Cortex®-M0 core running up to 72 MHz
  - One 24-bit system timer
  - Supports low power sleep mode
  - Single-cycle 32-bit hardware multiplier
  - NVIC for the 32 interrupt inputs, each with 4-levels of priority
  - Supports programmable mask-able interrupts
  - Serial Wire Debug supports with 2 watch-points/4 breakpoints
- Built-in LDO for wide operating voltage ranged from 2.5V to 5.5V
- Flash Memory
  - Supports 256/128 KB application ROM (APROM)
  - Supports 4 KB Flash for loader (LDROM)
  - Supports 2 KB Security Protection Rom (SPROM)
  - Supports 12 bytes User Configuration block to control system initiation
  - Supports Data Flash with configurable memory size
  - Supports 2 KB page erase for all embedded flash
  - 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 external read protection of whole flash memory by Security Lock Bit
  - Supports 2-wired ICP update through SWD/ICE interface
- SRAM Memory
  - 20 KB embedded SRAM
  - Supports byte-, half-word- and word-access
  - Supports PDMA mode
- Hardware Divider
  - Signed (two's complement) integer calculation
  - 32-bit dividend with 16-bit divisor calculation capacity
  - 32-bit quotient and 32-bit remainder outputs (16-bit remainder with sign extends to 32-bit)
  - Divided by zero warning flag
  - 6 HCLK clocks taken for one cycle calculation
  - Write divisor to trigger calculation
  - Waiting for calculation ready automatically when reading quotient and remainder
- PDMA (Peripheral DMA)
  - Supports 5 independent configurable channels for automatic data transfer between memories and peripherals
  - Supports single and burst transfer type
  - Supports Normal and Scatter-Gather Transfer modes
  - Supports two types of priorities modes: Fixed-priority and Round-robin modes
  - Supports byte-, half-word- and word-access
  - Supports incrementing mode for the source and destination address for each channel
  - Supports time-out function for channel 0 and channel 1
  - Supports software and SPI/I2S, UART, USCI, USB, ADC, PWM and TIMER request
- Clock Control



- Built-in 22.1184 MHz high speed RC oscillator for system operation (Frequency variation < 2% at -40°C ~ +105°C)
- Built-in 48 MHz internal high speed RC oscillator for USB device operation
- Built-in 10 kHz low speed RC oscillator for Watchdog Timer and Wake-up operation
- Built-in 4~24 MHz high speed crystal oscillator for precise timing operation
- Built-in 32.768 kHz low speed crystal oscillator for Real Time Clock
- Supports PLL up to 144 MHz for high resolution PWM operation
- Supports dynamically calibrating the HIRC48 to 48 MHz  $\pm 0.25\%$  by external 32.768K crystal oscillator (LXT)
- Supports dynamically calibrating the HIRC to 22.1184Mhz by external 32.768K crystal oscillator (LXT)
- Supports clock on-the-fly switch
- Supports clock failure detection for system clock
- Supports auto clock switch once clock failure detected
- Supports exception (NMI) generated once a clock failure detected
- Supports divided clock output

●GPIO

- Four I/O modes
- TTL/Schmitt trigger input selectable
- I/O pin configured as interrupt source with edge/level trigger setting
- Supports high driver and high sink current I/O (up to 20 mA at 5V)
- Supports software selectable slew rate control
- Supports up to 81/49/35 GPIOs for LQFP100/64/48 respectively

●Timer/PWM

- Supports 4 sets of Timers/PWM

Timer Mode	PWM Mode
TM_CNT_OUT	PWM_CH0
TM_EXT	PWM_CH1 (Complementary)

- Timer Mode
  - Supports 4 sets of 32-bit timers with 24-bit up-timer and one 8-bit pre-scale counter
  - Independent clock source for each timer
  - Provides one-shot, periodic, toggle and continuous counting operation modes
  - Supports event counting function to count the event from external pin
  - Supports input capture function to capture or reset counter value
  - 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 PWM, EADC and PDMA function
  - Supports Inter-Timer trigger mode
- PWM Mode
  - Supports maximum clock frequency up to 72MHz
  - Supports independent mode for 4 sets of independent PWM output channel
  - Supports complementary mode for 4 sets of complementary paired PWM output channel with 12-bit Dead-time generator
  - Supports 12-bit pre-scalar from 1 to 4096

- Supports 16-bit resolution PWM counter, each timer provides 1 PWM counter
  - Supports up, down and up/down counter operation type
  - Supports one-shot or Auto-reload counter operation mode
  - Supports mask function and tri-state enable for each PWM pin
  - Supports brake function
  - Supports interrupt when PWM counter match zero, period value or compared value, and brake condition happened
  - Supports trigger ADC when PWM counter match zero, period value or compared value
- Watchdog Timer
    - Supports multiple clock sources from LIRC(default selection), HCLK/2048 and LXT 8 selectable time-out period from 1.6ms ~ 26.0sec (depending on clock source)
    - Able to wake up from Power-down or Idle mode
    - Interrupt or reset selectable on watchdog time-out
  - Window Watchdog Timer
    - Supports multiple clock sources from HCLK/2048 (default selection) and LIRC Window set by 6-bit counter with 11-bit prescale
    - Interrupt or reset selectable on time-out
  - RTC
    - Supports separate battery power pin VBAT
    - Supports software compensation by setting frequency compensate register (FCR)
    - Supports RTC counter (second, minute, hour) and calendar counter (day, month, year)
    - Supports Alarm registers (second, minute, hour, day, month, year)
    - Supports Alarm mask registers
    - Selectable 12-hour or 24-hour mode
    - Automatic leap year recognition
    - Supports periodic time tick interrupt with 8 period options 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 and 1 second
    - Supports wake-up function
  - PWM
    - Supports maximum clock frequency up to 144MHz
    - Supports up to two PWM modules, each module provides 6 output channels.
    - Supports independent mode for PWM output/Capture input channel
    - Supports complementary mode for 2 complementary paired PWM output channel
      - Dead-time insertion with 12-bit resolution
      - Two compared values during one period
    - Supports 12-bit pre-scalar from 1 to 4096
    - Supports 16-bit resolution PWM counter
      - Up, down and up/down counter operation type
    - Supports mask function and tri-state enable for each PWM pin
    - Supports brake function
      - Brake source from pin and system safety events: clock failed, Brown-out detection and CPU lockup.
      - Noise filter for brake source from pin
      - Edge detect brake source to control brake state until brake interrupt cleared
      - Level detect brake source to auto recover function after brake condition removed
    - Supports interrupt on the following events:
      - PWM counter match zero, period value or compared value
      - Brake condition happened
    - Supports trigger ADC on the following events:



- Supports Auto-Baud Rate measurement and baud rate compensation function
- 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 LIN function mode
  - Supports LIN master/slave mode
  - Supports programmable break generation function for transmitter
  - Supports break detection function for receiver
- Supports RS-485 mode
  - Supports RS-485 9-bit mode
  - Supports hardware or software enables to program nRTS pin to control RS-485 transmission direction
- Supports nCTS, incoming data, Received Data FIFO reached threshold and RS-485 Address Match (AAD mode) wake-up function
- Supports PDMA transfer
- Smart Card Host (SC)

- Supports up to two Smart Card Hosts

SC Mode	UART Mode
SC_DATA	Rx
SC_CLK	Tx
SC_CD	-
SC_PWR	-
SC_RST	-

- SC Mode
  - Supports up to two ISO-7816-3 ports
  - Compliant to ISO-7816-3 T=0, T=1
  - Separate receive / transmit 4 bytes entry FIFO for data payloads
  - Programmable transmission clock frequency
  - Programmable receiver buffer trigger level
  - Programmable guard time selection (11 ETU ~ 266 ETU)
  - One 24-bit and two 8-bit time-out counters for Answer to Request (ATR) and waiting times processing
  - Supports auto inverse convention function
  - Supports transmitter and receiver error retry and error limit function
  - Supports hardware activation sequence process
  - Supports hardware warm reset sequence process
  - Supports hardware deactivation sequence process
  - Supports hardware auto deactivation sequence when detecting the card is removal
- UART Mode
  - Full duplex, asynchronous communications
  - Supports receiving / transmitting 4-bytes FIFO
  - Supports programmable baud rate generator for each channel
  - Programmable even, odd or no parity bit generation and detection
  - Programmable stop bit, 1 or 2 stop bit generation
- SPI/I<sup>2</sup>S

- Supports up to two SPI/I<sup>2</sup>S controllers

SPI Mode	I <sup>2</sup> S Mode
SPI_CLK	I2S_BCLK
SPI_SS	I2S_LRCLK
SPI_MOSI	I2S_DO
SPI_MISO	I2S_DI
-	I2S_MCLK

- SPI Mode
  - Supports Master or Slave mode operation
  - Configurable bit length of a transfer word from 8 to 32-bit
  - Provides separate 4-/8-level depth transmit and receive FIFO buffers
  - Supports MSB first or LSB first transfer sequence
  - Supports Byte Reorder function
  - Supports PDMA transfer
- I<sup>2</sup>S Mode
  - Supports Master or Slave mode operation
  - Capable of handling 8-, 16-, 24- and 32-bit word sizes in I2S mode
  - Provides separate 4-level depth transmit and receive FIFO buffers in I2S mode
  - Supports monaural and stereo audio data in I2S mode
  - Supports PCM mode A, PCM mode B, I2S and MSB justified data format in I2S mode
  - Supports PDMA transfer

● I<sup>2</sup>C

- Supports up to two sets of I2C device
- Supports Master/Slave mode
- Supports bidirectional data transfer between masters and slaves
- Supports multi-master bus (no central master)
- 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
- Supports multiple address recognition, four slave address with mask option
- Supports two-level buffer function
- Supports setup/hold time programmable
- Supports wake-up function

● USB 2.0 FS Device Controller

- Crystal-less USB 2.0 FS Device
- Compliant to USB specification version 2.0
- On-chip USB Transceiver
- Supports Control, Bulk In/Out, Interrupt and Isochronous transfers
- Auto suspend function when no bus signaling for 3 ms
- Supports USB 2.0 Link Power Management (LPM)
- Provides 8 programmable endpoints
- Supports 512 Bytes internal SRAM as USB buffer
- Provides remote wake-up capability
- On-chip 5V to 3.3V LDO for USB PHY

● ADC

- Supports 12-bit SAR ADC
- 12-bit resolution and 10-bit accuracy is guaranteed
- Analog input voltage range: 0~  $AV_{DD}$
- Supports external VREF pin
- Up to 20 single-end analog input channels
- Maximum ADC peripheral clock frequency is 16 MHz
- Conversion rate up to 800K SPS at 5V
- Configurable ADC internal sampling time
- Supports single-scan, single-cycle-scan, and continuous scan and scan on enabled channels
- Supports individual conversion result register with valid and overrun indicators for each channel
- Supports digital comparator to monitor conversion result and user can select whether to generate an interrupt when conversion result matches the compare register setting
- An A/D conversion can be triggered by:
  - Software enable
  - External pin (STADC)
  - Timer 0~3 overflow pulse trigger
  - PWM triggers with optional start delay period
- Supports 4 internal channels for
  - Operational amplifier output
  - Band-gap VBG input
  - Temperature sensor input
  - VBAT voltage measure
- Supports internal reference voltage: 2.048V, 2.560V, 3.072V and 4.096V
- Supports PDMA transfer

● Analog Comparator

- Supports up to 2 rail-to-rail analog comparators
- Supports 4 multiplexed I/O pins at positive node.
- Supports I/O pin and internal voltages at negative node
- Support selectable internal voltage reference from:
  - Band-gap  $V_{BG}$
  - Voltage divider source from  $AV_{DD}$  and internal reference voltage.
- Supports programmable hysteresis
- Supports programmable speed and power consumption
- Interrupts generated when compare results change, interrupt event condition is programmable.
- Supports power-down wake-up
- Supports triggers for break events and cycle-by-cycle control for PWM

● Cyclic Redundancy Calculation Unit

- Supports four common polynomials CRC-CCITT, CRC-8, CRC-16, and CRC-32
- Programmable initial 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
- Interrupt generated once checksum error occurs

● User Configurable VDD1=1.8~5.5V IO Interface

- Supports UART0, SPI0 and I2C0

● Supports 96-bit Unique ID (UID)

● Supports 128-bit Unique Customer ID (UCID)

● One built-in temperature sensor with 1°C resolution

- Brown-out detector
  - With 8 levels: 4.3 V/ 3.7V/ 2.7V/ 2.2V
  - Supports Brown-out Interrupt and Reset option
- Low Voltage Reset
  - Threshold voltage levels: 2.0 V
- Power consumption
  - Chip power down current < 10 uA with RAM data retention.
  - $V_{BAT}$  power domain operating current <1.5 uA
- Operating Temperature: -40°C ~105°C
- Packages
  - All Green package (RoHS)
  - LQFP 100-pin
  - LQFP 64-pin(7mmx7mm)
  - LQFP 48-pin
  - QFN 48-pin

### 3 ABBREVIATIONS

#### 3.1 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
DAP	Debug Access Port
DES	Data Encryption Standard
EBI	External Bus Interface
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	22.1184 MHz Internal High Speed RC Oscillator
HXT	4~24 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



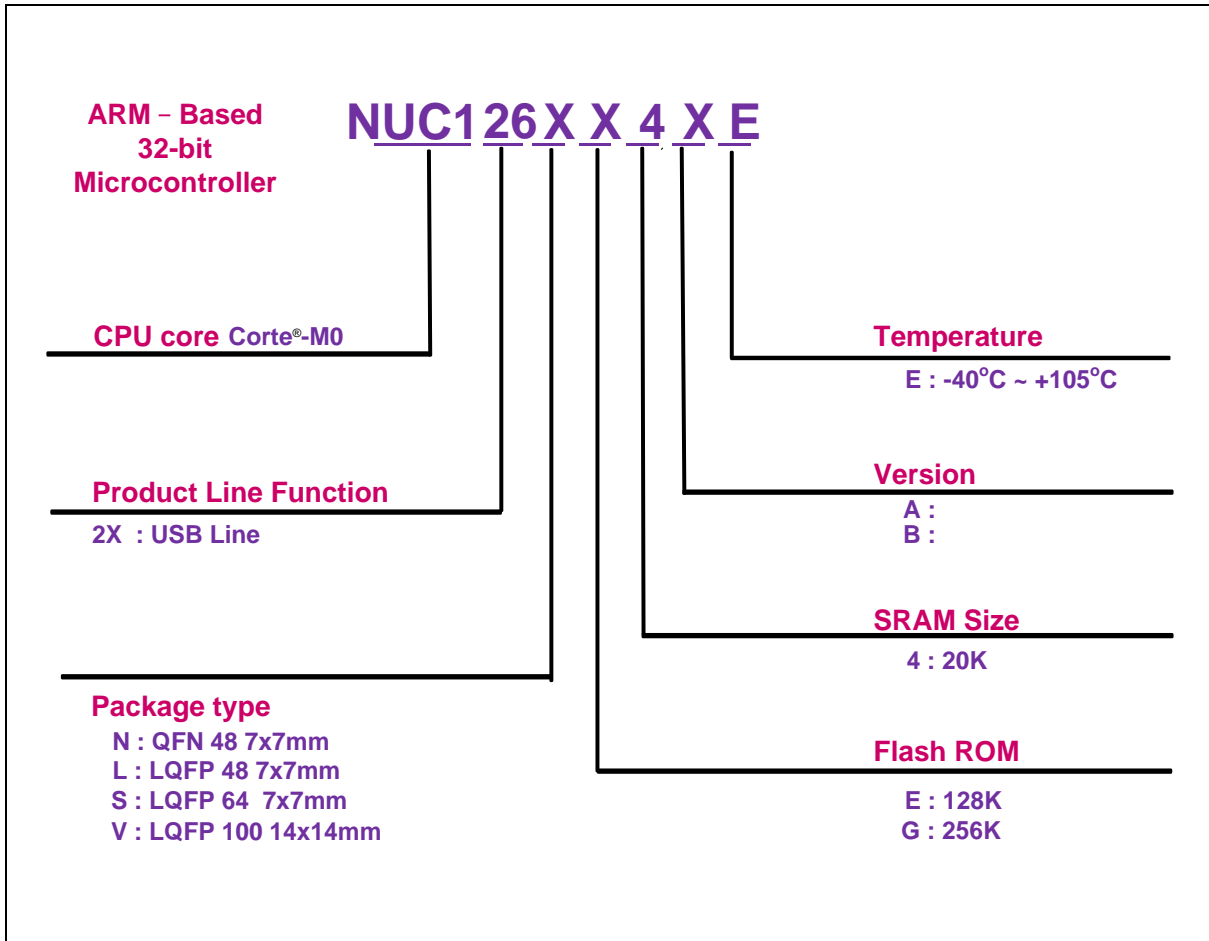
SPI	Serial Peripheral Interface
SPS	Samples per Second
TDES	Triple Data Encryption Standard
TMR	Timer Controller
UART	Universal Asynchronous Receiver/Transmitter
UCID	Unique Customer ID
USB	Universal Serial Bus
WDT	Watchdog Timer
WWDT	Window Watchdog Timer

Table 3.1-1 List of Abbreviations

4 PARTS INFORMATION LIST AND PIN CONFIGURATION

4.1 NuMicro® NUC126 Selection Guide

4.1.1 NuMicro® NUC126 Naming Rule



4.1.2 NuMicro® NUC126 USB Series (M452 Compatible) Selection Guide

Part Number	Flash (KB)	SRAM (KB)	Data Flash (KB)	SPROM (KB)	ISP ROM (KB)	I/O	Timer/PWM	PWM	Connectivity						ADC(12-Bit)	ACMP	PDMA	VBAT(RTC)	V <sub>DDIO</sub>	EBI	ICP/IAP/ISP	Package
									USB	USCI*	UART	SC/UART	SPI/I <sup>2</sup> S	I <sup>2</sup> C								
NUC126NE4AE	128	20	Conf*	2	4	35	4	10	1	3	3	2	2	2	9-ch	2	5	√	√	√	√	QFN 48
NUC126LE4AE	128	20	Conf*	2	4	35	4	10	1	3	3	2	2	2	9-ch	2	5	√	√	√	√	LQFP 48
NUC126LG4AE	256	20	Conf*	2	4	35	4	10	1	3	3	2	2	2	9-ch	2	5	√	√	√	√	LQFP 48
NUC126SE4AE	128	20	Conf*	2	4	49	4	12	1	3	3	2	2	2	15-ch	2	5	√	√	√	√	LQFP 64*
NUC126SG4AE	256	20	Conf*	2	4	49	4	12	1	3	3	2	2	2	15-ch	2	5	√	√	√	√	LQFP 64*
NUC126VG4AE	256	20	Conf*	2	4	81	4	12	1	3	3	2	2	2	20-ch	2	5	√	√	√	√	LQFP 100

Conf\*: Configurable  
 USCI\*: support UART, SPI or I<sup>2</sup>C  
 LQFP64\*: 7x7 mm

## 4.2 Pin Configuration

### 4.2.1 NuMicro® NUC126 USB Series QFN48 Pin Diagram

Corresponding Part Number: NUC126NE4AE

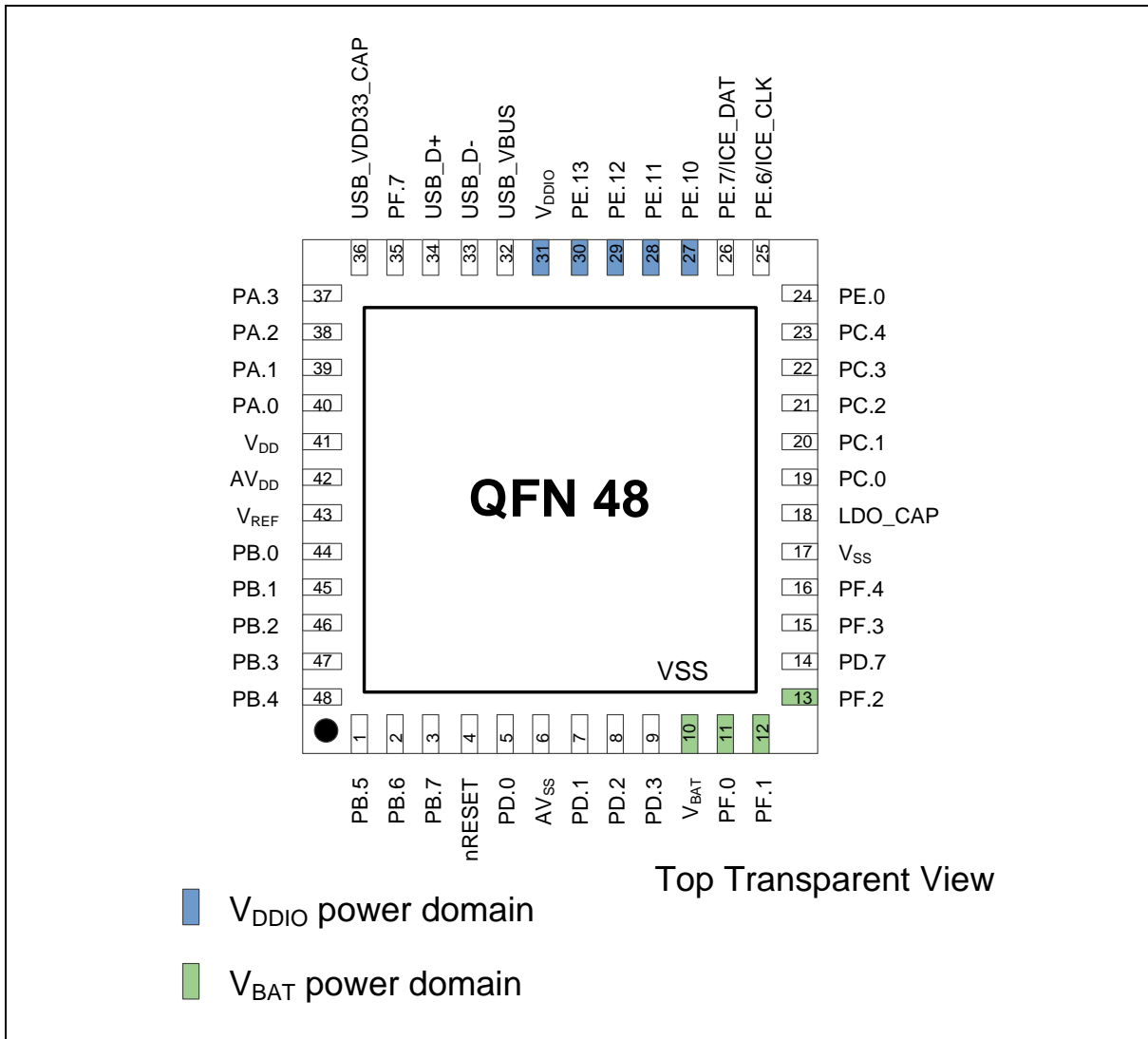


Figure 4.2-1 NuMicro® NUC126 USB Series QFN 48-pin Diagram

4.2.2 NuMicro® NUC126 USB Series LQFP48 Pin Diagram

Corresponding Part Number: NUC126LE4AE, NUC126LG4AE

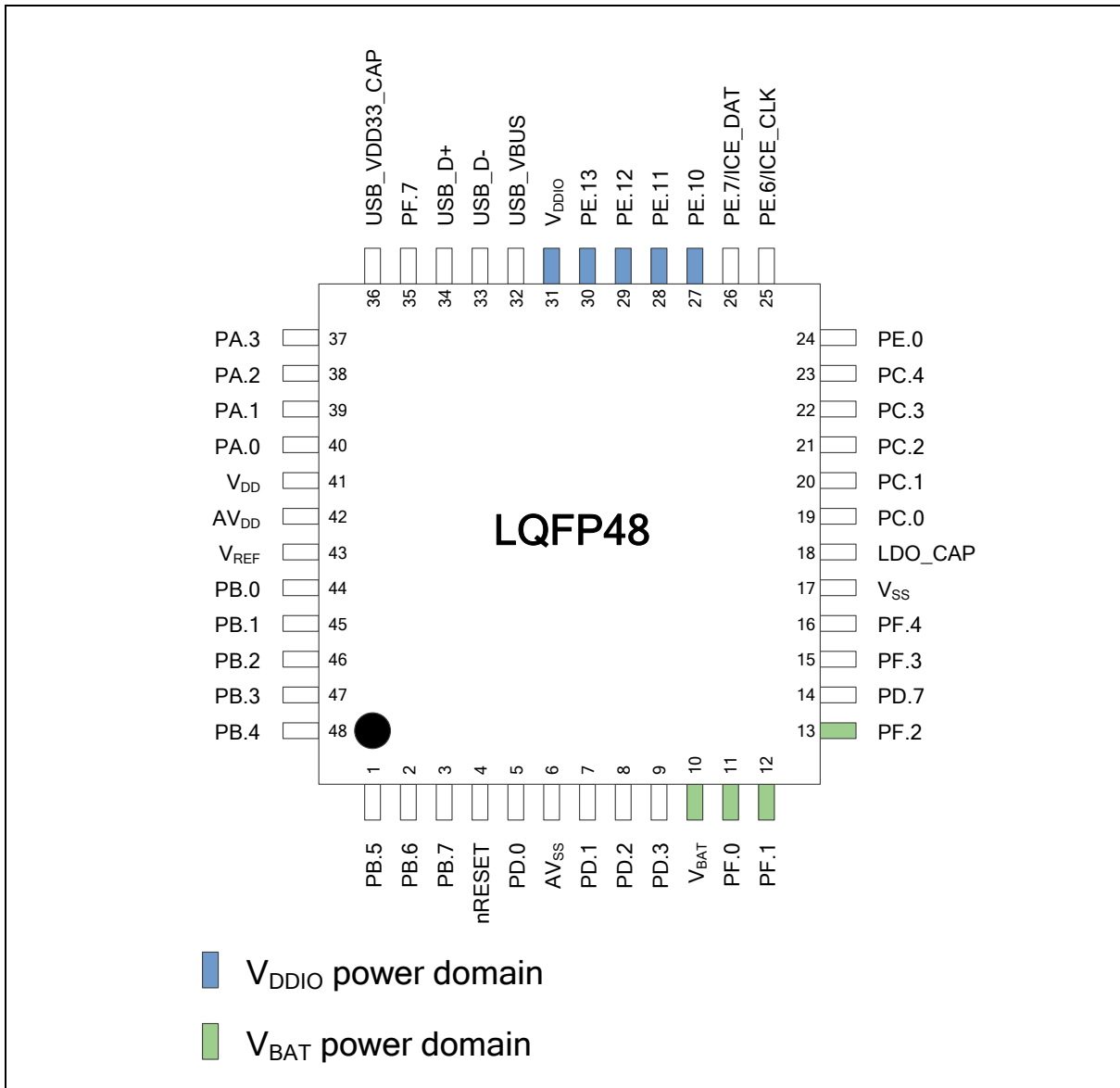


Figure 4.2-2 NuMicro® NUC126 USB Series LQFP 48-pin Diagram

4.2.3 NuMicro® NUC126 USB Series LQFP64 Pin Diagram

Corresponding Part Number: NUC126SE4AE, NUC126SG4AE

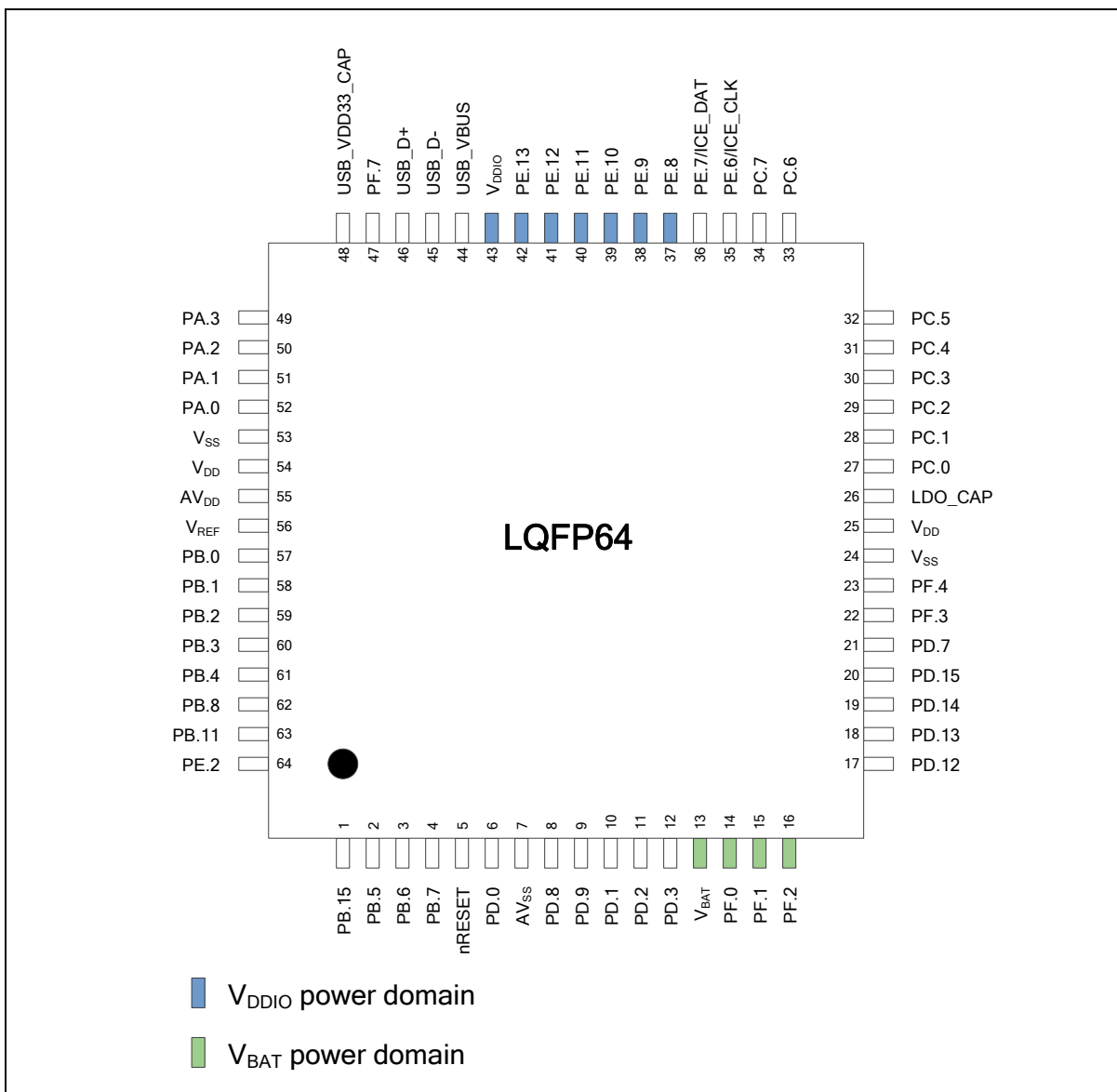


Figure 4.2-3 NuMicro® NUC126 USB Series LQFP 64-pin Diagram

4.2.4 NuMicro® NUC126 USB Series LQFP100 Pin Diagram  
 Corresponding Part Number: NUC126VG4AE

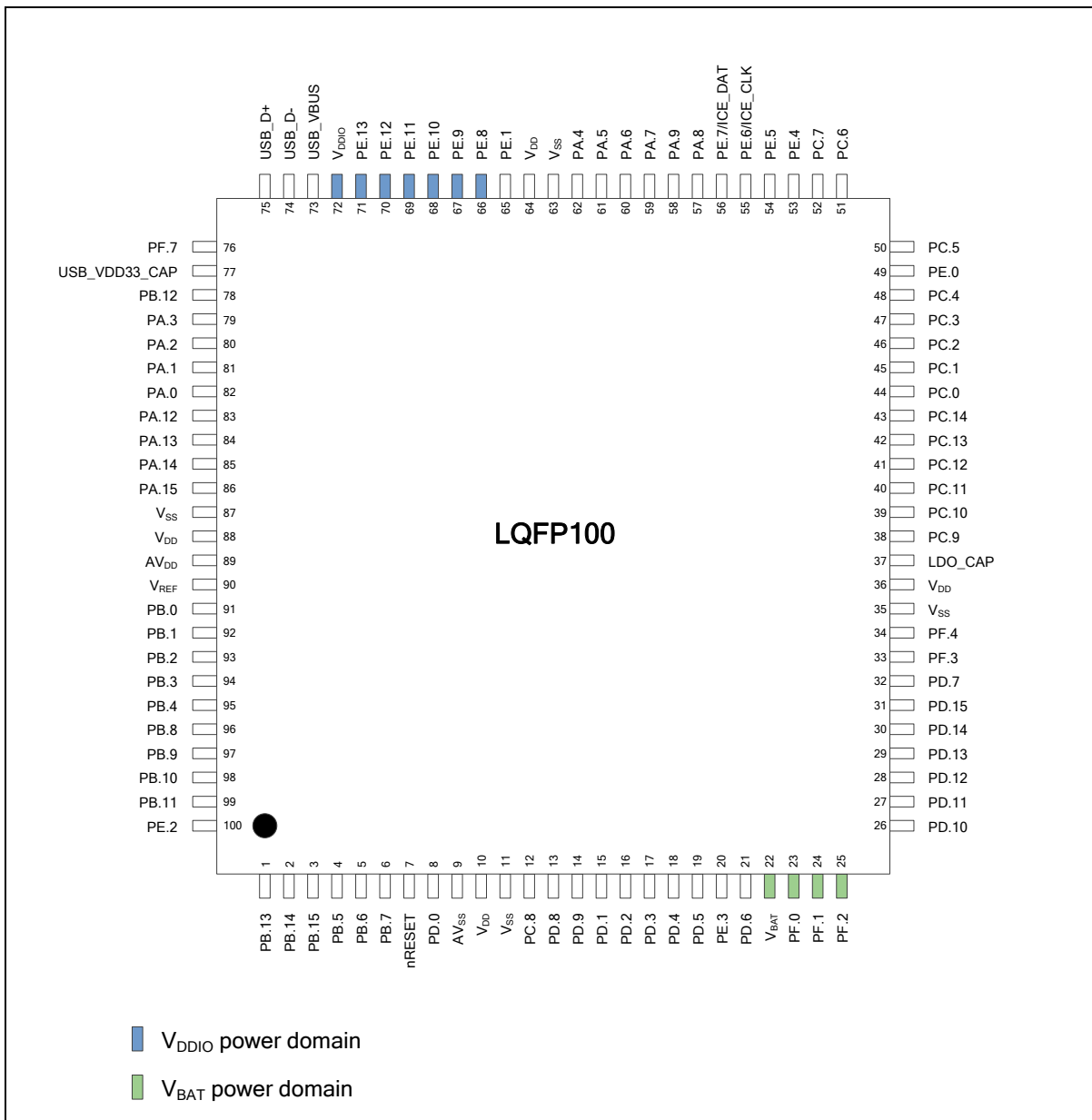


Figure 4.2-4 NuMicro® NUC126 USB Series LQFP 100-pin Diagram

### 4.3 Pin Description

#### 4.3.1 NUC126 USB Series Pin Description

MFP\* = Multi-function pin. (Refer to section SYS\_GP<sub>x</sub>\_MFPL and SYS\_GP<sub>x</sub>\_MFPH)

PA.0 MFP0 means SYS\_GP0\_MFPL[3:0]=0x0.

PA.9 MFP5 means SYS\_GP0\_MFPH[7:4]=0x5.

48 Pin	64 Pin	100 Pin	Pin Name	Type	MFP	Description
		1	PB.13	I/O	MFP0	General purpose digital I/O pin.
			ADC0_CH10	A	MFP1	ADC0 channel 10 analog input.
		2	PB.14	I/O	MFP0	General purpose digital I/O pin.
			ADC0_CH11	A	MFP1	ADC0 channel 11 analog input.
	1	3	PB.15	I/O	MFP0	General purpose digital I/O pin.
			ADC0_CH12	A	MFP1	ADC0 channel 12 analog input.
			ACMP0_P3	A	MFP5	Analog comparator 0 positive input 3 pin.
			EBI_nCS1	O	MFP7	EBI chip select 1 output pin.
1	2	4	PB.5	I/O	MFP0	General purpose digital I/O pin.
			ADC0_CH13	A	MFP1	ADC0 channel 13 analog input.
			SPI0_MOSI	I/O	MFP2	SPI0 MOSI (Master Out, Slave In) pin.
			SPI1_MOSI	I/O	MFP3	SPI1 MOSI (Master Out, Slave In) pin.
			ACMP0_P2	A	MFP5	Analog comparator 0 positive input 2 pin.
			SC1_RST	O	MFP6	Smart Card 1 reset pin.
			EBI_AD6	I/O	MFP7	EBI address/data bus bit 6.
			UART2_RXD	I	MFP9	UART2 data receiver input pin.
2	3	5	PB.6	I/O	MFP0	General purpose digital I/O pin.
			ADC0_CH14	A	MFP1	ADC0 channel 14 analog input.
			SPI0_MISO	I/O	MFP2	SPI0 MISO (Master In, Slave Out) pin.
			SPI1_MISO	I/O	MFP3	SPI1 MISO (Master In, Slave Out) pin.
			ACMP0_P1	A	MFP5	Analog comparator 0 positive input 1 pin.
			SC1_PWR	O	MFP6	Smart Card 1 power pin.
			EBI_AD5	I/O	MFP7	EBI address/data bus bit 5.
3	4	6	PB.7	I/O	MFP0	General purpose digital I/O pin.
			ADC0_CH15	A	MFP1	ADC0 channel 15 analog input.
			SPI0_CLK	I/O	MFP2	SPI0 serial clock pin.



48 Pin	64 Pin	100 Pin	Pin Name	Type	MFP	Description
			SPI1_CLK	I/O	MFP3	SPI1 serial clock pin.
			USCI2_CTL1	I/O	MFP4	USCI2 control 1 pin.
			ACMP0_P0	A	MFP5	Analog comparator 0 positive input 0 pin.
			SC1_DAT	I/O	MFP6	Smart Card 1 data pin.
			EBI_AD4	I/O	MFP7	EBI address/data bus bit 4.
4	5	7	nRESET	I	MFP0	External reset input: active LOW, with an internal pull-up. Set this pin low reset to initial state.
5	6	8	PD.0	I/O	MFP0	General purpose digital I/O pin.
			SPI0_I2SMCLK	I/O	MFP1	SPI0 I2S master clock output pin
			SPI1_I2SMCLK	I/O	MFP2	SPI1 I2S master clock output pin
			UART0_RXD	I	MFP3	UART0 data receiver input pin.
			USCI2_CTL0	I/O	MFP4	USCI2 control 0 pin.
			ACMP1_N	A	MFP5	Analog comparator 1 negative input pin.
			SC1_CLK	O	MFP6	Smart Card 1 clock pin.
			INT3	I	MFP8	External interrupt 3 input pin.
6	7	9	AV <sub>SS</sub>	P	MFP0	Ground pin for analog circuit.
		10	V <sub>DD</sub>	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
		11	V <sub>SS</sub>	P	MFP0	Ground pin for digital circuit.
		12	PC.8	I/O	MFP0	General purpose digital I/O pin.
			ADC0_CH16	A	MFP1	ADC0 channel 16 analog input.
			UART0_nRTS	O	MFP3	UART0 request to Send output pin.
	8	13	PD.8	I/O	MFP0	General purpose digital I/O pin.
			ADC0_CH17	A	MFP1	ADC0 channel 17 analog input.
			UART0_nCTS	I	MFP3	UART0 clear to Send input pin.
			USCI2_CTL1	I/O	MFP4	USCI2 control 1 pin.
			TM2	I/O	MFP6	Timer2 event counter input/toggle output pin.
			EBI_nCS0	O	MFP7	EBI chip select 0 output pin.
	9	14	PD.9	I/O	MFP0	General purpose digital I/O pin.
			ADC0_CH18	A	MFP1	ADC0 channel 18 analog input.
			UART0_RXD	I	MFP3	UART0 data receiver input pin.
			USCI2_CTL0	I/O	MFP4	USCI2 control 0 pin.

48 Pin	64 Pin	100 Pin	Pin Name	Type	MFP	Description
			ACMP1_P3	A	MFP5	Analog comparator 1 positive input 3 pin.
			TM3	I/O	MFP6	Timer3 event counter input/toggle output pin.
			EBI_ALE	O	MFP7	EBI address latch enable output pin.
7	10	15	PD.1	I/O	MFP0	General purpose digital I/O pin.
			ADC0_CH19	A	MFP1	ADC0 channel 19 analog input.
			PWM0_SYNC_IN	I	MFP2	PWM0 counter synchronous trigger input pin.
			UART0_TXD	O	MFP3	UART0 data transmitter output pin.
			USCI2_CLK	I/O	MFP4	USCI2 clock pin.
			ACMP1_P2	A	MFP5	Analog comparator 1 positive input 2 pin.
			TM0	I/O	MFP6	Timer0 event counter input/toggle output pin.
			EBI_nRD	O	MFP7	EBI read enable output pin.
8	11	16	PD.2	I/O	MFP0	General purpose digital I/O pin.
			ADC0_ST	I	MFP1	ADC0 external trigger input pin.
			TM0_EXT	I/O	MFP3	Timer0 external capture input/toggle output pin.
			USCI2_DAT0	I/O	MFP4	USCI2 data 0 pin.
			ACMP1_P1	A	MFP5	Analog comparator 1 positive input 1 pin.
			PWM0_BRAKE0	I	MFP6	PWM0 Brake 0 input pin.
			EBI_nWR	O	MFP7	EBI write enable output pin.
			INT0	I	MFP8	External interrupt 0 input pin.
9	12	17	PD.3	I/O	MFP0	General purpose digital I/O pin.
			TM2	I/O	MFP1	Timer2 event counter input/toggle output pin.
			SPI0_I2SMCLK	I/O	MFP2	SPI0 I2S master clock output pin
			TM1_EXT	I/O	MFP3	Timer1 external capture input/toggle output pin.
			USCI2_DAT1	I/O	MFP4	USCI2 data 1 pin.
			ACMP1_P0	A	MFP5	Analog comparator 1 positive input 0 pin.
			PWM0_BRAKE1	I	MFP6	PWM0 Brake 1 input pin.
			EBI_MCLK	O	MFP7	EBI external clock output pin.
			INT1	I	MFP8	External interrupt 1 input pin.
		18	PD.4	I/O	MFP0	General purpose digital I/O pin.
			SPI1_CLK	I/O	MFP2	SPI1 serial clock pin.
			I2C0_SDA	I/O	MFP3	I2C0 data input/output pin.

48 Pin	64 Pin	100 Pin	Pin Name	Type	MFP	Description
			UART2_nRTS	O	MFP4	UART2 request to Send output pin.
			PWM0_BRAKE0	I	MFP5	PWM0 Brake 0 input pin.
			TM0	I/O	MFP6	Timer0 event counter input/toggle output pin.
		19	PD.5	I/O	MFP0	General purpose digital I/O pin.
			CLKO	O	MFP1	Clock Out
			SPI1_MISO	I/O	MFP2	SPI1 MISO (Master In, Slave Out) pin.
			I2C0_SCL	I/O	MFP3	I2C0 clock pin.
			UART2_nCTS	I	MFP4	UART2 clear to Send input pin.
			PWM0_BRAKE1	I	MFP5	PWM0 Brake 1 input pin.
			TM1	I/O	MFP6	Timer1 event counter input/toggle output pin.
		20	PE.3	I/O	MFP0	General purpose digital I/O pin.
			SPI1_MOSI	I/O	MFP2	SPI1 MOSI (Master Out, Slave In) pin.
			UART2_RXD	I	MFP4	UART2 data receiver input pin.
			PWM0_CH3	I/O	MFP6	PWM0 channel 3 output/capture input.
		21	PD.6	I/O	MFP0	General purpose digital I/O pin.
			CLKO	O	MFP1	Clock Out
			SPI1_SS	I/O	MFP2	SPI1 slave select pin.
			UART0_RXD	I	MFP3	UART0 data receiver input pin.
			UART2_TXD	O	MFP4	UART2 data transmitter output pin.
			ACMP0_O	O	MFP5	Analog comparator 0 output pin.
			PWM0_CH5	I/O	MFP6	PWM0 channel 5 output/capture input.
			EBI_nWR	O	MFP7	EBI write enable output pin.
10	13	22	V <sub>BAT</sub>	P	MFP0	Power supply by batteries for RTC.
11	14	23	PF.0	I/O	MFP0	General purpose digital I/O pin.
			X32_OUT	O	MFP1	External 32.768 kHz crystal output pin.
			USCI2_CTL1	I/O	MFP5	USCI2 control 1 pin.
			INT5	I	MFP8	External interrupt 5 input pin.
12	15	24	PF.1	I/O	MFP0	General purpose digital I/O pin.
			X32_IN	I	MFP1	External 32.768 kHz crystal input pin.
			USCI2_CTL0	I/O	MFP5	USCI2 control 0 pin.
			PWM1_BRAKE0	I	MFP6	PWM1 Brake 0 input pin.

48 Pin	64 Pin	100 Pin	Pin Name	Type	MFP	Description
13	16	25	PF.2	I/O	MFP0	General purpose digital I/O pin.
			USCI2_CLK	I/O	MFP5	USCI2 clock pin.
			PWM1_BRAKE1	I	MFP6	PWM1 Brake 1 input pin.
		26	PD.10	I/O	MFP0	General purpose digital I/O pin.
			TM2	I/O	MFP4	Timer2 event counter input/toggle output pin.
			USCI2_DAT0	I/O	MFP5	USCI2 data 0 pin.
		27	PD.11	I/O	MFP0	General purpose digital I/O pin.
			TM3	I/O	MFP4	Timer3 event counter input/toggle output pin.
			USCI2_DAT1	I/O	MFP5	USCI2 data 1 pin.
17		28	PD.12	I/O	MFP0	General purpose digital I/O pin.
			USCI1_CTL0	I/O	MFP1	USCI1 control 0 pin.
			SPI1_SS	I/O	MFP2	SPI1 slave select pin.
			UART0_TXD	O	MFP3	UART0 data transmitter output pin.
			PWM1_CH0	I/O	MFP6	PWM1 channel 0 output/capture input.
			EBI_ADR16	O	MFP7	EBI address bus bit 16.
18		29	PD.13	I/O	MFP0	General purpose digital I/O pin.
			USCI1_DAT1	I/O	MFP1	USCI1 data 1 pin.
			SPI1_MOSI	I/O	MFP2	SPI1 MOSI (Master Out, Slave In) pin.
			UART0_RXD	I	MFP3	UART0 data receiver input pin.
			PWM1_CH1	I/O	MFP6	PWM1 channel 1 output/capture input.
			EBI_ADR17	O	MFP7	EBI address bus bit 17.
19		30	PD.14	I/O	MFP0	General purpose digital I/O pin.
			USCI1_DAT0	I/O	MFP1	USCI1 data 0 pin.
			SPI1_MISO	I/O	MFP2	SPI1 MISO (Master In, Slave Out) pin.
			UART0_nCTS	I	MFP3	UART0 clear to Send input pin.
			PWM1_CH2	I/O	MFP6	PWM1 channel 2 output/capture input.
			EBI_ADR18	O	MFP7	EBI address bus bit 18.
20		31	PD.15	I/O	MFP0	General purpose digital I/O pin.
			USCI1_CLK	I/O	MFP1	USCI1 clock pin.
			SPI1_CLK	I/O	MFP2	SPI1 serial clock pin.
			UART0_nRTS	O	MFP3	UART0 request to Send output pin.

48 Pin	64 Pin	100 Pin	Pin Name	Type	MFP	Description
			PWM1_CH3	I/O	MFP6	PWM1 channel 3 output/capture input.
			EBI_ADR19	O	MFP7	EBI address bus bit 19.
14	21	32	PD.7	I/O	MFP0	General purpose digital I/O pin.
			USCI1_CTL1	I/O	MFP1	USCI1 control 1 pin.
			SPI0_I2SMCLK	I/O	MFP2	SPI0 I2S master clock output pin
			PWM0_SYNC_IN	I	MFP3	PWM0 counter synchronous trigger input pin.
			TM1	I/O	MFP4	Timer1 event counter input/toggle output pin.
			ACMP0_O	O	MFP5	Analog comparator 0 output pin.
			PWM0_CH5	I/O	MFP6	PWM0 channel 5 output/capture input.
			EBI_nRD	O	MFP7	EBI read enable output pin.
15	22	33	PF.3	I/O	MFP0	General purpose digital I/O pin.
			XT1_OUT	O	MFP1	External 4~24 MHz (high speed) crystal output pin.
			I2C1_SCL	I/O	MFP3	I2C1 clock pin.
16	23	34	PF.4	I/O	MFP0	General purpose digital I/O pin.
			XT1_IN	I	MFP1	External 4~24 MHz (high speed) crystal input pin.
			I2C1_SDA	I/O	MFP3	I2C1 data input/output pin.
17	24	35	V <sub>SS</sub>	P	MFP0	Ground pin for digital circuit.
	25	36	V <sub>DD</sub>	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
18	26	37	LDO_CAP	A	MFP0	LDO output pin.
		38	PC.9	I/O	MFP0	General purpose digital I/O pin.
			SPI0_I2SMCLK	I/O	MFP2	SPI0 I2S master clock output pin
			I2C1_SCL	I/O	MFP3	I2C1 clock pin.
			USCI2_CTL1	I/O	MFP4	USCI2 control 1 pin.
			PWM1_CH0	I/O	MFP6	PWM1 channel 0 output/capture input.
		39	PC.10	I/O	MFP0	General purpose digital I/O pin.
			SPI0_MOSI	I/O	MFP2	SPI0 MOSI (Master Out, Slave In) pin.
			I2C1_SDA	I/O	MFP3	I2C1 data input/output pin.
			USCI2_DAT1	I/O	MFP4	USCI2 data 1 pin.
			PWM1_CH1	I/O	MFP6	PWM1 channel 1 output/capture input.
		40	PC.11	I/O	MFP0	General purpose digital I/O pin.
			SPI0_MISO	I/O	MFP2	SPI0 MISO (Master In, Slave Out) pin.

48 Pin	64 Pin	100 Pin	Pin Name	Type	MFP	Description
			USCI2_CLK	I/O	MFP4	USCI2 clock pin.
			PWM1_CH2	I/O	MFP6	PWM1 channel 2 output/capture input.
		41	PC.12	I/O	MFP0	General purpose digital I/O pin.
			SPI0_CLK	I/O	MFP2	SPI0 serial clock pin.
			USCI2_CTL0	I/O	MFP4	USCI2 control 0 pin.
			PWM1_CH3	I/O	MFP6	PWM1 channel 3 output/capture input.
		42	PC.13	I/O	MFP0	General purpose digital I/O pin.
			SPI0_SS	I/O	MFP2	SPI0 slave select pin.
			USCI2_DAT0	I/O	MFP4	USCI2 data 0 pin.
			PWM1_CH4	I/O	MFP6	PWM1 channel 4 output/capture input.
		43	PC.14	I/O	MFP0	General purpose digital I/O pin.
			PWM1_CH5	I/O	MFP6	PWM1 channel 5 output/capture input.
19	27	44	PC.0	I/O	MFP0	General purpose digital I/O pin.
			SC0_DAT	I/O	MFP1	Smart Card 0 data pin.
			SPI0_CLK	I/O	MFP2	SPI0 serial clock pin.
			UART2_nCTS	I	MFP3	UART2 clear to Send input pin.
			USCI0_DAT0	I/O	MFP4	USCI0 data 0 pin.
			ACMP0_WLAT	I	MFP5	Analog comparator 0 window latch input pin
			PWM0_CH0	I/O	MFP6	PWM0 channel 0 output/capture input.
			EBI_AD8	I/O	MFP7	EBI address/data bus bit 8.
			INT2	I	MFP8	External interrupt 2 input pin.
20	28	45	PC.1	I/O	MFP0	General purpose digital I/O pin.
			CLKO	O	MFP1	Clock Out
			SC0_CLK	O	MFP2	Smart Card 0 clock pin.
			UART2_nRTS	O	MFP3	UART2 request to Send output pin.
			USCI0_DAT1	I/O	MFP4	USCI0 data 1 pin.
			ACMP1_WLAT	I	MFP5	Analog comparator 1 window latch input pin
			PWM0_CH1	I/O	MFP6	PWM0 channel 1 output/capture input.
			EBI_AD9	I/O	MFP7	EBI address/data bus bit 9.
21	29	46	PC.2	I/O	MFP0	General purpose digital I/O pin.
			SC0_RST	O	MFP1	Smart Card 0 reset pin.

48 Pin	64 Pin	100 Pin	Pin Name	Type	MFP	Description
			SPI0_SS	I/O	MFP2	SPI0 slave select pin.
			UART2_TXD	O	MFP3	UART2 data transmitter output pin.
			USCI0_CTL1	I/O	MFP4	USCI0 control 1 pin.
			ACMP1_O	O	MFP5	Analog comparator 1 output pin.
			PWM0_CH2	I/O	MFP6	PWM0 channel 2 output/capture input.
			EBI_AD10	I/O	MFP7	EBI address/data bus bit 10.
22	30	47	PC.3	I/O	MFP0	General purpose digital I/O pin.
			SC0_PWR	O	MFP1	Smart Card 0 power pin.
			SPI0_MOSI	I/O	MFP2	SPI0 MOSI (Master Out, Slave In) pin.
			UART2_RXD	I	MFP3	UART2 data receiver input pin.
			USCI0_CTL0	I/O	MFP5	USCI0 control 0 pin.
			PWM0_CH3	I/O	MFP6	PWM0 channel 3 output/capture input.
			EBI_AD11	I/O	MFP7	EBI address/data bus bit 11.
23	31	48	PC.4	I/O	MFP0	General purpose digital I/O pin.
			SC0_nCD	I	MFP1	Smart Card 0 card detect pin.
			SPI0_MISO	I/O	MFP2	SPI0 MISO (Master In, Slave Out) pin.
			I2C1_SCL	I/O	MFP3	I2C1 clock pin.
			USCI0_CLK	I/O	MFP5	USCI0 clock pin.
			PWM0_CH4	I/O	MFP6	PWM0 channel 4 output/capture input.
			EBI_AD12	I/O	MFP7	EBI address/data bus bit 12.
24		49	PE.0	I/O	MFP0	General purpose digital I/O pin.
			SPI0_CLK	I/O	MFP2	SPI0 serial clock pin.
			I2C1_SDA	I/O	MFP3	I2C1 data input/output pin.
			TM2_EXT	I/O	MFP4	Timer2 external capture input/toggle output pin.
			SC0_nCD	I	MFP5	Smart Card 0 card detect pin.
			PWM0_CH0	I/O	MFP6	PWM0 channel 0 output/capture input.
			EBI_nCS1	O	MFP7	EBI chip select 1 output pin.
			INT4	I	MFP8	External interrupt 4 input pin.
	32	50	PC.5	I/O	MFP0	General purpose digital I/O pin.
			SPI0_I2SMCLK	I/O	MFP2	SPI0 I2S master clock output pin
			I2C1_SDA	I/O	MFP3	I2C1 data input/output pin.

48 Pin	64 Pin	100 Pin	Pin Name	Type	MFP	Description
			USCI0_DAT0	I/O	MFP4	USCI0 data 0 pin.
			PWM0_CH5	I/O	MFP6	PWM0 channel 5 output/capture input.
			EBI_AD13	I/O	MFP7	EBI address/data bus bit 13.
	33	51	PC.6	I/O	MFP0	General purpose digital I/O pin.
			USCI0_DAT1	I/O	MFP4	USCI0 data 1 pin.
			ACMP1_O	O	MFP5	Analog comparator 1 output pin.
			PWM1_CH0	I/O	MFP6	PWM1 channel 0 output/capture input.
			EBI_AD14	I/O	MFP7	EBI address/data bus bit 14.
	34	52	PC.7	I/O	MFP0	General purpose digital I/O pin.
			USCI0_CTL1	I/O	MFP4	USCI0 control 1 pin.
			PWM1_CH1	I/O	MFP6	PWM1 channel 1 output/capture input.
			EBI_AD15	I/O	MFP7	EBI address/data bus bit 15.
		53	PE.4	I/O	MFP0	General purpose digital I/O pin.
			I2C0_SCL	I/O	MFP2	I2C0 clock pin.
			I2C1_SCL	I/O	MFP3	I2C1 clock pin.
			USCI0_CTL0	I/O	MFP4	USCI0 control 0 pin.
			SC0_PWR	O	MFP5	Smart Card 0 power pin.
			PWM1_BRAKE0	I	MFP6	PWM1 Brake 0 input pin.
			EBI_nCS0	O	MFP7	EBI chip select 0 output pin.
			INT0	I	MFP8	External interrupt 0 input pin.
		54	PE.5	I/O	MFP0	General purpose digital I/O pin.
			I2C0_SDA	I/O	MFP2	I2C0 data input/output pin.
			I2C1_SDA	I/O	MFP3	I2C1 data input/output pin.
			USCI0_CLK	I/O	MFP4	USCI0 clock pin.
			SC0_RST	O	MFP5	Smart Card 0 reset pin.
			PWM1_BRAKE1	I	MFP6	PWM1 Brake 1 input pin.
			EBI_ALE	O	MFP7	EBI address latch enable output pin.
			INT1	I	MFP8	External interrupt 1 input pin.
25	35	55	PE.6	I/O	MFP0	General purpose digital I/O pin.
			ICE_CLK	I	MFP1	Serial wired debugger clock pin.
			I2C0_SCL	I/O	MFP2	I2C0 clock pin.



48 Pin	64 Pin	100 Pin	Pin Name	Type	MFP	Description
			UART0_RXD	I	MFP3	UART0 data receiver input pin.
26	36	56	PE.7	I/O	MFP0	General purpose digital I/O pin.
			ICE_DAT	O	MFP1	Serial wired debugger data pin.
			I2C0_SDA	I/O	MFP2	I2C0 data input/output pin.
			UART0_TXD	O	MFP3	UART0 data transmitter output pin.
		57	PA.8	I/O	MFP0	General purpose digital I/O pin.
			CLKO	O	MFP1	Clock Out
			I2C1_SCL	I/O	MFP2	I2C1 clock pin.
			UART1_TXD	O	MFP3	UART1 data transmitter output pin.
			SC0_PWR	O	MFP4	Smart Card 0 power pin.
			SC1_RST	O	MFP5	Smart Card 1 reset pin.
			TM_BRAKE0	I	MFP6	TM_BRAKE0 I Timer Brake * input pin.
			PWM0_BRAKE0	I	MFP7	PWM0 Brake 0 input pin.
			TM1	I/O	MFP8	Timer1 event counter input/toggle output pin.
		58	PA.9	I/O	MFP0	General purpose digital I/O pin.
			SPI1_I2SMCLK	I/O	MFP1	SPI1 I2S master clock output pin
			I2C1_SDA	I/O	MFP2	I2C1 data input/output pin.
			UART1_RXD	I	MFP3	UART1 data receiver input pin.
			SC0_RST	O	MFP4	Smart Card 0 reset pin.
			SC1_PWR	O	MFP5	Smart Card 1 power pin.
			TM_BRAKE1	I	MFP6	TM_BRAKE1 I Timer Brake * input pin.
			PWM1_BRAKE1	I	MFP7	PWM1 Brake 1 input pin.
			TM2	I/O	MFP8	Timer2 event counter input/toggle output pin.
		59	PA.7	I/O	MFP0	General purpose digital I/O pin.
			SPI1_CLK	I/O	MFP2	SPI1 serial clock pin.
			TM0_EXT	I/O	MFP3	Timer0 external capture input/toggle output pin.
			TM_BRAKE1	I	MFP6	TM_BRAKE1 I Timer Brake * input pin.
			EBI_AD7	I/O	MFP7	EBI address/data bus bit 7.
		60	PA.6	I/O	MFP0	General purpose digital I/O pin.
			SPI1_MISO	I/O	MFP2	SPI1 MISO (Master In, Slave Out) pin.
			TM1_EXT	I/O	MFP3	Timer1 external capture input/toggle output pin.

48 Pin	64 Pin	100 Pin	Pin Name	Type	MFP	Description
			TM_BRAKE2	I	MFP6	TM_BRAKE2 I Timer Brake * input pin.
			EBI_AD6	I/O	MFP7	EBI address/data bus bit 6.
		61	PA.5	I/O	MFP0	General purpose digital I/O pin.
			SPI1_MOSI	I/O	MFP2	SPI1 MOSI (Master Out, Slave In) pin.
			TM2_EXT	I/O	MFP3	Timer2 external capture input/toggle output pin.
			TM_BRAKE3	I	MFP6	TM_BRAKE3 I Timer Brake * input pin.
			EBI_AD5	I/O	MFP7	EBI address/data bus bit 5.
		62	PA.4	I/O	MFP0	General purpose digital I/O pin.
			SPI1_SS	I/O	MFP2	SPI1 slave select pin.
			TM3_EXT	I/O	MFP3	Timer3 external capture input/toggle output pin.
			EBI_AD4	I/O	MFP7	EBI address/data bus bit 4.
		63	V <sub>SS</sub>	P	MFP0	Ground pin for digital circuit.
		64	V <sub>DD</sub>	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
		65	PE.1	I/O	MFP0	General purpose digital I/O pin.
			TM3_EXT	I/O	MFP3	Timer3 external capture input/toggle output pin.
			SC0_nCD	I	MFP5	Smart Card 0 card detect pin.
			PWM0_CH1	I/O	MFP6	PWM0 channel 1 output/capture input.
	37	66	PE.8	I/O	MFP0	General purpose digital I/O pin.
			UART1_TXD	O	MFP1	UART1 data transmitter output pin.
			TM0	I/O	MFP3	Timer0 event counter input/toggle output pin.
			I2C1_SCL	I/O	MFP4	I2C1 clock pin.
			SC0_PWR	O	MFP5	Smart Card 0 power pin.
	38	67	PE.9	I/O	MFP0	General purpose digital I/O pin.
			UART1_RXD	I	MFP1	UART1 data receiver input pin.
			TM1	I/O	MFP3	Timer1 event counter input/toggle output pin.
			I2C1_SDA	I/O	MFP4	I2C1 data input/output pin.
			SC0_RST	O	MFP5	Smart Card 0 reset pin.
27	39	68	PE.10	I/O	MFP0	General purpose digital I/O pin.
			SPI1_MISO	I/O	MFP1	SPI1 MISO (Master In, Slave Out) pin.
			SPI0_MISO	I/O	MFP2	SPI0 MISO (Master In, Slave Out) pin.
			UART1_nCTS	I	MFP3	UART1 clear to Send input pin.

48 Pin	64 Pin	100 Pin	Pin Name	Type	MFP	Description
			SC0_DAT	I/O	MFP5	Smart Card 0 data pin.
			SPI1_CLK	I/O	MFP6	SPI1 serial clock pin.
			EBI_AD7	I/O	MFP7	EBI address/data bus bit 7.
			TM0_EXT	I/O	MFP8	Timer0 external capture input/toggle output pin.
28	40	69	PE.11	I/O	MFP0	General purpose digital I/O pin.
			SPI1_MOSI	I/O	MFP1	SPI1 MOSI (Master Out, Slave In) pin.
			SPI0_MOSI	I/O	MFP2	SPI0 MOSI (Master Out, Slave In) pin.
			UART1_nRTS	O	MFP3	UART1 request to Send output pin.
			SC0_CLK	O	MFP5	Smart Card 0 clock pin.
			SPI1_MISO	I/O	MFP6	SPI1 MISO (Master In, Slave Out) pin.
			EBI_AD6	I/O	MFP7	EBI address/data bus bit 6.
			TM1_EXT	I/O	MFP8	Timer1 external capture input/toggle output pin.
29	41	70	PE.12	I/O	MFP0	General purpose digital I/O pin.
			SPI1_SS	I/O	MFP1	SPI1 slave select pin.
			SPI0_SS	I/O	MFP2	SPI0 slave select pin.
			UART1_TXD	O	MFP3	UART1 data transmitter output pin.
			I2C0_SCL	I/O	MFP4	I2C0 clock pin.
			SPI1_MOSI	I/O	MFP6	SPI1 MOSI (Master Out, Slave In) pin.
			EBI_AD5	I/O	MFP7	EBI address/data bus bit 5.
			TM2_EXT	I/O	MFP8	Timer2 external capture input/toggle output pin.
30	42	71	PE.13	I/O	MFP0	General purpose digital I/O pin.
			SPI1_CLK	I/O	MFP1	SPI1 serial clock pin.
			SPI0_CLK	I/O	MFP2	SPI0 serial clock pin.
			UART1_RXD	I	MFP3	UART1 data receiver input pin.
			I2C0_SDA	I/O	MFP4	I2C0 data input/output pin.
			SPI1_SS	I/O	MFP6	SPI1 slave select pin.
			EBI_AD4	I/O	MFP7	EBI address/data bus bit 4.
			TM3_EXT	I/O	MFP8	Timer3 external capture input/toggle output pin.
31	43	72	V <sub>DDIO</sub>	P	MFP0	Power supply for PE.1, PE.8~PE.13.
32	44	73	USB_VBUS	P	MFP0	Power supply from USB host or HUB.
33	45	74	USB_D-	A	MFP0	USB differential signal D-.

48 Pin	64 Pin	100 Pin	Pin Name	Type	MFP	Description
34	46	75	USB_D+	A	MFP0	USB differential signal D+.
35	47	76	PF.7	I/O	MFP0	General purpose digital I/O pin.
36	48	77	USB_VDD33_CAP	A	MFP0	Internal power regulator output 3.3V decoupling pin.
		78	PB.12	I/O	MFP0	General purpose digital I/O pin.
			PWM1_CH1	I/O	MFP6	PWM1 channel 1 output/capture input.
37	49	79	PA.3	I/O	MFP0	General purpose digital I/O pin.
			UART0_RXD	I	MFP2	UART0 data receiver input pin.
			UART0_nRTS	O	MFP3	UART0 request to Send output pin.
			I2C0_SCL	I/O	MFP4	I2C0 clock pin.
			SC0_PWR	O	MFP5	Smart Card 0 power pin.
			PWM1_CH2	I/O	MFP6	PWM1 channel 2 output/capture input.
			EBI_AD3	I/O	MFP7	EBI address/data bus bit 3.
			USCI1_CLK	I/O	MFP8	USCI1 clock pin.
38	50	80	PA.2	I/O	MFP0	General purpose digital I/O pin.
			UART0_TXD	O	MFP2	UART0 data transmitter output pin.
			UART0_nCTS	I	MFP3	UART0 clear to Send input pin.
			I2C0_SDA	I/O	MFP4	I2C0 data input/output pin.
			SC0_RST	O	MFP5	Smart Card 0 reset pin.
			PWM1_CH3	I/O	MFP6	PWM1 channel 3 output/capture input.
			EBI_AD2	I/O	MFP7	EBI address/data bus bit 2.
			USCI1_CTL0	I/O	MFP8	USCI1 control 0 pin.
39	51	81	PA.1	I/O	MFP0	General purpose digital I/O pin.
			UART1_nRTS	O	MFP1	UART1 request to Send output pin.
			UART1_RXD	I	MFP3	UART1 data receiver input pin.
			USCI1_CTL1	I/O	MFP4	USCI1 control 1 pin.
			SC0_DAT	I/O	MFP5	Smart Card 0 data pin.
			PWM1_CH4	I/O	MFP6	PWM1 channel 4 output/capture input.
			EBI_AD1	I/O	MFP7	EBI address/data bus bit 1.
40	52	82	PA.0	I/O	MFP0	General purpose digital I/O pin.
			UART1_nCTS	I	MFP1	UART1 clear to Send input pin.
			UART1_TXD	O	MFP3	UART1 data transmitter output pin.

48 Pin	64 Pin	100 Pin	Pin Name	Type	MFP	Description
			USCI1_CTL0	I/O	MFP4	USCI1 control 0 pin.
			SC0_CLK	O	MFP5	Smart Card 0 clock pin.
			PWM1_CH5	I/O	MFP6	PWM1 channel 5 output/capture input.
			EBI_AD0	I/O	MFP7	EBI address/data bus bit 0.
			INT0	I	MFP8	External interrupt 0 input pin.
		83	PA.12	I/O	MFP0	General purpose digital I/O pin.
			SPI1_I2SMCLK	I/O	MFP2	SPI1 I2S master clock output pin
			UART2_RXD	I	MFP3	UART2 data receiver input pin.
			UART1_RXD	I	MFP4	UART1 data receiver input pin.
			TM_BRAKE2	I	MFP6	TM_BRAKE2 I Timer Brake * input pin.
		84	PA.13	I/O	MFP0	General purpose digital I/O pin.
			UART2_TXD	O	MFP3	UART2 data transmitter output pin.
			UART1_TXD	O	MFP4	UART1 data transmitter output pin.
			TM_BRAKE3	I	MFP6	TM_BRAKE3 I Timer Brake * input pin.
		85	PA.14	I/O	MFP0	General purpose digital I/O pin.
			UART2_nCTS	I	MFP3	UART2 clear to Send input pin.
			USCI1_CTL1	I/O	MFP4	USCI1 control 1 pin.
			TM2	I/O	MFP6	Timer2 event counter input/toggle output pin.
		86	PA.15	I/O	MFP0	General purpose digital I/O pin.
			UART2_nRTS	O	MFP3	UART2 request to Send output pin.
			USCI1_CLK	I/O	MFP4	USCI1 clock pin.
			TM3	I/O	MFP6	Timer3 event counter input/toggle output pin.
	53	87	V <sub>SS</sub>	P	MFP0	Ground pin for digital circuit.
41	54	88	V <sub>DD</sub>	P	MFP0	Power supply for I/O ports and LDO source for internal PLL and digital circuit.
42	55	89	AV <sub>DD</sub>	P	MFP0	Power supply for internal analog circuit.
43	56	90	V <sub>REF</sub>	A	MFP0	ADC reference voltage input. Note: This pin needs to be connected with a 1uF capacitor.
44	57	91	PB.0	I/O	MFP0	General purpose digital I/O pin.
			ADC0_CH0	A	MFP1	ADC0 channel 0 analog input.
			VDET_P0	A	MFP2	Voltage detector positive input 0 pin.

48 Pin	64 Pin	100 Pin	Pin Name	Type	MFP	Description
			UART2_RXD	I	MFP3	UART2 data receiver input pin.
			TM2	I/O	MFP4	Timer2 event counter input/toggle output pin.
			USCI1_DAT0	I/O	MFP6	USCI1 data 0 pin.
			EBI_nWRL	O	MFP7	EBI low byte write enable output pin.
			INT1	I	MFP8	External interrupt 1 input pin.
			TM1_EXT	I/O	MFP10	Timer1 external capture input/toggle output pin.
45	58	92	PB.1	I/O	MFP0	General purpose digital I/O pin.
			ADC0_CH1	A	MFP1	ADC0 channel 1 analog input.
			VDET_P1	A	MFP2	Voltage detector positive input 1 pin.
			UART2_TXD	O	MFP3	UART2 data transmitter output pin.
			TM3	I/O	MFP4	Timer3 event counter input/toggle output pin.
			SC0_RST	O	MFP5	Smart Card 0 reset pin.
			PWM0_SYNC_OUT	O	MFP6	PWM0 counter synchronous trigger output pin.
			EBI_nWRH	O	MFP7	EBI high byte write enable output pin
			USCI1_DAT1	I/O	MFP8	USCI1 data 1 pin.
46	59	93	PB.2	I/O	MFP0	General purpose digital I/O pin.
			ADC0_CH2	A	MFP1	ADC0 channel 2 analog input.
			SPI0_CLK	I/O	MFP2	SPI0 serial clock pin.
			SPI1_CLK	I/O	MFP3	SPI1 serial clock pin.
			UART1_RXD	I	MFP4	UART1 data receiver input pin.
			SC0_nCD	I	MFP5	Smart Card 0 card detect pin.
			TM_BRAKE0	I	MFP6	TM_BRAKE0 I Timer Brake * input pin.
			EBI_nCS0	O	MFP7	EBI chip select 0 output pin.
			USCI0_DAT0	I/O	MFP8	USCI0 data 0 pin.
			TM2_EXT	I/O	MFP10	Timer2 external capture input/toggle output pin.
47	60	94	PB.3	I/O	MFP0	General purpose digital I/O pin.
			ADC0_CH3	A	MFP1	ADC0 channel 3 analog input.
			SPI0_MISO	I/O	MFP2	SPI0 MISO (Master In, Slave Out) pin.
			SPI1_MISO	I/O	MFP3	SPI1 MISO (Master In, Slave Out) pin.
			UART1_TXD	O	MFP4	UART1 data transmitter output pin.
			TM_BRAKE1	I	MFP6	TM_BRAKE1 I Timer Brake * input pin.

48 Pin	64 Pin	100 Pin	Pin Name	Type	MFP	Description
			EBI_ALE	O	MFP7	EBI address latch enable output pin.
			USCI0_DAT1	I/O	MFP8	USCI0 data 1 pin.
			TM0_EXT	I/O	MFP10	Timer0 external capture input/toggle output pin.
48	61	95	PB.4	I/O	MFP0	General purpose digital I/O pin.
			ADC0_CH4	A	MFP1	ADC0 channel 4 analog input.
			SPI0_SS	I/O	MFP2	SPI0 slave select pin.
			SPI1_SS	I/O	MFP3	SPI1 slave select pin.
			UART1_nCTS	I	MFP4	UART1 clear to Send input pin.
			ACMP0_N	A	MFP5	Analog comparator 0 negative input pin.
			SC1_nCD	I	MFP6	Smart Card 1 card detect pin.
			EBI_AD7	I/O	MFP7	EBI address/data bus bit 7.
			USCI0_CTL1	I/O	MFP8	USCI0 control 1 pin.
			UART2_RXD	I	MFP9	UART2 data receiver input pin.
			TM1_EXT	I/O	MFP10	Timer1 external capture input/toggle output pin.
	62	96	PB.8	I/O	MFP0	General purpose digital I/O pin.
			ADC0_CH5	A	MFP1	ADC0 channel 5 analog input.
			UART1_nRTS	O	MFP4	UART1 request to Send output pin.
			TM_BRAKE2	I	MFP5	TM_BRAKE2 I Timer Brake * input pin.
			PWM0_CH2	I/O	MFP6	PWM0 channel 2 output/capture input.
			USCI0_CTL0	I/O	MFP8	USCI0 control 0 pin.
		97	PB.9	I/O	MFP0	General purpose digital I/O pin.
			ADC0_CH6	A	MFP1	ADC0 channel 6 analog input.
			USCI0_CLK	I/O	MFP8	USCI0 clock pin.
		98	PB.10	I/O	MFP0	General purpose digital I/O pin.
			ADC0_CH7	A	MFP1	ADC0 channel 7 analog input.
	63	99	PB.11	I/O	MFP0	General purpose digital I/O pin.
			ADC0_CH8	A	MFP1	ADC0 channel 8 analog input.
	64	100	PE.2	I/O	MFP0	General purpose digital I/O pin.
			ADC0_CH9	A	MFP1	ADC0 channel 9 analog input.
			UART1_nRTS	O	MFP4	UART1 request to Send output pin.
			TM_BRAKE3	I	MFP5	TM_BRAKE3 I Timer Brake * input pin.

48 Pin	64 Pin	100 Pin	Pin Name	Type	MFP	Description
			PWM0_CH2	I/O	MFP6	PWM0 channel 2 output/capture input.
			USCI0_CTL0	I/O	MFP8	USCI0 control 0 pin.



### 4.3.2 GPIO Multi-function Pin Summary

MFP\* = Multi-function pin. (Refer to section SYS\_GP<sub>x</sub>\_MFPL and SYS\_GP<sub>x</sub>\_MFPH)

PA.0 MFP0 means SYS\_GP0\_MFPL[3:0]=0x0.

PA.9 MFP5 means SYS\_GP0\_MFPH[7:4]=0x5.

	Pin Name	Type	MFP	Description
PA.0	PA.0	I/O	MFP0	General purpose digital I/O pin.
	UART1_nCTS	I	MFP1	UART1 clear to Send input pin.
	UART1_TXD	O	MFP3	UART1 data transmitter output pin.
	USCI1_CTL0	I/O	MFP4	USCI1 control 0 pin.
	SC0_CLK	O	MFP5	Smart Card 0 clock pin.
	PWM1_CH5	I/O	MFP6	PWM1 channel 5 output/capture input.
	EBI_AD0	I/O	MFP7	EBI address/data bus bit 0.
	INT0	I	MFP8	External interrupt 0 input pin.
PA.1	PA.1	I/O	MFP0	General purpose digital I/O pin.
	UART1_nRTS	O	MFP1	UART1 request to Send output pin.
	UART1_RXD	I	MFP3	UART1 data receiver input pin.
	USCI1_CTL1	I/O	MFP4	USCI1 control 1 pin.
	SC0_DAT	I/O	MFP5	Smart Card 0 data pin.
	PWM1_CH4	I/O	MFP6	PWM1 channel 4 output/capture input.
	EBI_AD1	I/O	MFP7	EBI address/data bus bit 1.
PA.2	PA.2	I/O	MFP0	General purpose digital I/O pin.
	UART0_TXD	O	MFP2	UART0 data transmitter output pin.
	UART0_nCTS	I	MFP3	UART0 clear to Send input pin.
	I2C0_SDA	I/O	MFP4	I <sup>2</sup> C0 data input/output pin.
	SC0_RST	O	MFP5	Smart Card 0 reset pin.
	PWM1_CH3	I/O	MFP6	PWM1 channel 3 output/capture input.
	EBI_AD2	I/O	MFP7	EBI address/data bus bit 2.
	USCI1_CTL0	I/O	MFP8	USCI1 control 0 pin.
PA.3	PA.3	I/O	MFP0	General purpose digital I/O pin.
	UART0_RXD	I	MFP2	UART0 data receiver input pin.
	UART0_nRTS	O	MFP3	UART0 request to Send output pin.
	I2C0_SCL	I/O	MFP4	I <sup>2</sup> C0 clock pin.
	SC0_PWR	O	MFP5	Smart Card 0 power pin.

	Pin Name	Type	MFP	Description
	PWM1_CH2	I/O	MFP6	PWM1 channel 2 output/capture input.
	EBI_AD3	I/O	MFP7	EBI address/data bus bit 3.
	USCI1_CLK	I/O	MFP8	USCI1 clock pin.
PA.4	PA.4	I/O	MFP0	General purpose digital I/O pin.
	SPI1_SS	I/O	MFP2	SPI1 slave select pin.
	TM3_EXT	I/O	MFP3	Timer3 external capture input/toggle output pin.
	EBI_AD4	I/O	MFP7	EBI address/data bus bit 4.
PA.5	PA.5	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MOSI	I/O	MFP2	SPI1 MOSI (Master Out, Slave In) pin.
	TM2_EXT	I/O	MFP3	Timer2 external capture input/toggle output pin.
	TM_BRAKE3	I	MFP6	TM_BRAKE3 I Timer Brake * input pin.
	EBI_AD5	I/O	MFP7	EBI address/data bus bit 5.
PA.6	PA.6	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MISO	I/O	MFP2	SPI1 MISO (Master In, Slave Out) pin.
	TM1_EXT	I/O	MFP3	Timer1 external capture input/toggle output pin.
	TM_BRAKE2	I	MFP6	TM_BRAKE2 I Timer Brake * input pin.
	EBI_AD6	I/O	MFP7	EBI address/data bus bit 6.
PA.7	PA.7	I/O	MFP0	General purpose digital I/O pin.
	SPI1_CLK	I/O	MFP2	SPI1 serial clock pin.
	TM0_EXT	I/O	MFP3	Timer0 external capture input/toggle output pin.
	TM_BRAKE1	I	MFP6	TM_BRAKE1 I Timer Brake * input pin.
	EBI_AD7	I/O	MFP7	EBI address/data bus bit 7.
PA.8	PA.8	I/O	MFP0	General purpose digital I/O pin.
	CLKO	O	MFP1	Clock Out
	I2C1_SCL	I/O	MFP2	I <sup>2</sup> C1 clock pin.
	UART1_TXD	O	MFP3	UART1 data transmitter output pin.
	SC0_PWR	O	MFP4	Smart Card 0 power pin.
	SC1_RST	O	MFP5	Smart Card 1 reset pin.
	TM_BRAKE0	I	MFP6	TM_BRAKE0 I Timer Brake * input pin.
	PWM0_BRAKE0	I	MFP7	PWM0 Brake 0 input pin.
	TM1	I/O	MFP8	Timer1 event counter input/toggle output pin.
PA.9	PA.9	I/O	MFP0	General purpose digital I/O pin.

	Pin Name	Type	MFP	Description
	SPI1_I2SMCLK	I/O	MFP1	SPI1 I <sup>2</sup> S master clock output pin
	I2C1_SDA	I/O	MFP2	I <sup>2</sup> C1 data input/output pin.
	UART1_RXD	I	MFP3	UART1 data receiver input pin.
	SC0_RST	O	MFP4	Smart Card 0 reset pin.
	SC1_PWR	O	MFP5	Smart Card 1 power pin.
	TM_BRAKE1	I	MFP6	TM_BRAKE1 I Timer Brake * input pin.
	PWM1_BRAKE1	I	MFP7	PWM1 Brake 1 input pin.
	TM2	I/O	MFP8	Timer2 event counter input/toggle output pin.
PA.10	PA.10	I/O	MFP0	General purpose digital I/O pin.
	UART1_nCTS	I	MFP3	UART1 clear to Send input pin.
	SC1_DAT	I/O	MFP5	Smart Card 1 data pin.
PA.11	PA.11	I/O	MFP0	General purpose digital I/O pin.
	UART1_nRTS	O	MFP3	UART1 request to Send output pin.
	SC1_CLK	O	MFP5	Smart Card 1 clock pin.
PA.12	PA.12	I/O	MFP0	General purpose digital I/O pin.
	SPI1_I2SMCLK	I/O	MFP2	SPI1 I <sup>2</sup> S master clock output pin
	UART2_RXD	I	MFP3	UART2 data receiver input pin.
	UART1_RXD	I	MFP4	UART1 data receiver input pin.
	TM_BRAKE2	I	MFP6	TM_BRAKE2 I Timer Brake * input pin.
PA.13	PA.13	I/O	MFP0	General purpose digital I/O pin.
	UART2_TXD	O	MFP3	UART2 data transmitter output pin.
	UART1_TXD	O	MFP4	UART1 data transmitter output pin.
	TM_BRAKE3	I	MFP6	TM_BRAKE3 I Timer Brake * input pin.
PA.14	PA.14	I/O	MFP0	General purpose digital I/O pin.
	UART2_nCTS	I	MFP3	UART2 clear to Send input pin.
	USCI1_CTL1	I/O	MFP4	USCI1 control 1 pin.
	TM2	I/O	MFP6	Timer2 event counter input/toggle output pin.
PA.15	PA.15	I/O	MFP0	General purpose digital I/O pin.
	UART2_nRTS	O	MFP3	UART2 request to Send output pin.
	USCI1_CLK	I/O	MFP4	USCI1 clock pin.
	TM3	I/O	MFP6	Timer3 event counter input/toggle output pin.
PB.0	PB.0	I/O	MFP0	General purpose digital I/O pin.

	Pin Name	Type	MFP	Description
	ADC0_CH0	A	MFP1	ADC0 channel 0 analog input.
	VDET_P0	A	MFP2	Voltage detector positive input 0 pin.
	UART2_RXD	I	MFP3	UART2 data receiver input pin.
	TM2	I/O	MFP4	Timer2 event counter input/toggle output pin.
	USCI1_DAT0	I/O	MFP6	USCI1 data 0 pin.
	EBI_nWRL	O	MFP7	EBI low byte write enable output pin.
	INT1	I	MFP8	External interrupt 1 input pin.
	TM1_EXT	I/O	MFP10	Timer1 external capture input/toggle output pin.
PB.1	PB.1	I/O	MFP0	General purpose digital I/O pin.
	ADC0_CH1	A	MFP1	ADC0 channel 1 analog input.
	VDET_P1	A	MFP2	Voltage detector positive input 1 pin.
	UART2_TXD	O	MFP3	UART2 data transmitter output pin.
	TM3	I/O	MFP4	Timer3 event counter input/toggle output pin.
	SC0_RST	O	MFP5	Smart Card 0 reset pin.
	PWM0_SYNC_OUT	O	MFP6	PWM0 counter synchronous trigger output pin.
	EBI_nWRH	O	MFP7	EBI high byte write enable output pin
	USCI1_DAT1	I/O	MFP8	USCI1 data 1 pin.
PB.2	PB.2	I/O	MFP0	General purpose digital I/O pin.
	ADC0_CH2	A	MFP1	ADC0 channel 2 analog input.
	SPI0_CLK	I/O	MFP2	SPI0 serial clock pin.
	SPI1_CLK	I/O	MFP3	SPI1 serial clock pin.
	UART1_RXD	I	MFP4	UART1 data receiver input pin.
	SC0_nCD	I	MFP5	Smart Card 0 card detect pin.
	TM_BRAKE0	I	MFP6	TM_BRAKE0 I Timer Brake * input pin.
	EBI_nCS0	O	MFP7	EBI chip select 0 output pin.
	USCI0_DAT0	I/O	MFP8	USCI0 data 0 pin.
	TM2_EXT	I/O	MFP10	Timer2 external capture input/toggle output pin.
PB.3	PB.3	I/O	MFP0	General purpose digital I/O pin.
	ADC0_CH3	A	MFP1	ADC0 channel 3 analog input.
	SPI0_MISO	I/O	MFP2	SPI0 MISO (Master In, Slave Out) pin.
	SPI1_MISO	I/O	MFP3	SPI1 MISO (Master In, Slave Out) pin.
	UART1_TXD	O	MFP4	UART1 data transmitter output pin.

	Pin Name	Type	MFP	Description
	TM_BRAKE1	I	MFP6	TM_BRAKE1   Timer Brake * input pin.
	EBI_ALE	O	MFP7	EBI address latch enable output pin.
	USCIO_DAT1	I/O	MFP8	USCIO data 1 pin.
	TM0_EXT	I/O	MFP10	Timer0 external capture input/toggle output pin.
PB.4	PB.4	I/O	MFP0	General purpose digital I/O pin.
	ADC0_CH4	A	MFP1	ADC0 channel 4 analog input.
	SPI0_SS	I/O	MFP2	SPI0 slave select pin.
	SPI1_SS	I/O	MFP3	SPI1 slave select pin.
	UART1_nCTS	I	MFP4	UART1 clear to Send input pin.
	ACMP0_N	A	MFP5	Analog comparator 0 negative input pin.
	SC1_nCD	I	MFP6	Smart Card 1 card detect pin.
	EBI_AD7	I/O	MFP7	EBI address/data bus bit 7.
	USCIO_CTL1	I/O	MFP8	USCIO control 1 pin.
	UART2_RXD	I	MFP9	UART2 data receiver input pin.
TM1_EXT	I/O	MFP10	Timer1 external capture input/toggle output pin.	
PB.5	PB.5	I/O	MFP0	General purpose digital I/O pin.
	ADC0_CH13	A	MFP1	ADC0 channel 13 analog input.
	SPI0_MOSI	I/O	MFP2	SPI0 MOSI (Master Out, Slave In) pin.
	SPI1_MOSI	I/O	MFP3	SPI1 MOSI (Master Out, Slave In) pin.
	ACMP0_P2	A	MFP5	Analog comparator 0 positive input 2 pin.
	SC1_RST	O	MFP6	Smart Card 1 reset pin.
	EBI_AD6	I/O	MFP7	EBI address/data bus bit 6.
	UART2_RXD	I	MFP9	UART2 data receiver input pin.
PB.6	PB.6	I/O	MFP0	General purpose digital I/O pin.
	ADC0_CH14	A	MFP1	ADC0 channel 14 analog input.
	SPI0_MISO	I/O	MFP2	SPI0 MISO (Master In, Slave Out) pin.
	SPI1_MISO	I/O	MFP3	SPI1 MISO (Master In, Slave Out) pin.
	ACMP0_P1	A	MFP5	Analog comparator 0 positive input 1 pin.
	SC1_PWR	O	MFP6	Smart Card 1 power pin.
	EBI_AD5	I/O	MFP7	EBI address/data bus bit 5.
PB.7	PB.7	I/O	MFP0	General purpose digital I/O pin.
	ADC0_CH15	A	MFP1	ADC0 channel 15 analog input.

	Pin Name	Type	MFP	Description
	SPI0_CLK	I/O	MFP2	SPI0 serial clock pin.
	SPI1_CLK	I/O	MFP3	SPI1 serial clock pin.
	USCI2_CTL1	I/O	MFP4	USCI2 control 1 pin.
	ACMP0_P0	A	MFP5	Analog comparator 0 positive input 0 pin.
	SC1_DAT	I/O	MFP6	Smart Card 1 data pin.
	EBI_AD4	I/O	MFP7	EBI address/data bus bit 4.
PB.8	PB.8	I/O	MFP0	General purpose digital I/O pin.
	ADC0_CH5	A	MFP1	ADC0 channel 5 analog input.
	UART1_nRTS	O	MFP4	UART1 request to Send output pin.
	TM_BRAKE2	I	MFP5	TM_BRAKE2 I Timer Brake * input pin.
	PWM0_CH2	I/O	MFP6	PWM0 channel 2 output/capture input.
	USCI0_CTL0	I/O	MFP8	USCI0 control 0 pin.
PB.9	PB.9	I/O	MFP0	General purpose digital I/O pin.
	ADC0_CH6	A	MFP1	ADC0 channel 6 analog input.
	USCI0_CLK	I/O	MFP8	USCI0 clock pin.
PB.10	PB.10	I/O	MFP0	General purpose digital I/O pin.
	ADC0_CH7	A	MFP1	ADC0 channel 7 analog input.
PB.11	PB.11	I/O	MFP0	General purpose digital I/O pin.
	ADC0_CH8	A	MFP1	ADC0 channel 8 analog input.
PB.12	PB.12	I/O	MFP0	General purpose digital I/O pin.
	PWM1_CH1	I/O	MFP6	PWM1 channel 1 output/capture input.
PB.13	PB.13	I/O	MFP0	General purpose digital I/O pin.
	ADC0_CH10	A	MFP1	ADC0 channel 10 analog input.
PB.14	PB.14	I/O	MFP0	General purpose digital I/O pin.
	ADC0_CH11	A	MFP1	ADC0 channel 11 analog input.
PB.15	PB.15	I/O	MFP0	General purpose digital I/O pin.
	ADC0_CH12	A	MFP1	ADC0 channel 12 analog input.
	ACMP0_P3	A	MFP5	Analog comparator 0 positive input 3 pin.
	EBI_nCS1	O	MFP7	EBI chip select 1 output pin.
PC.0	PC.0	I/O	MFP0	General purpose digital I/O pin.
	SC0_DAT	I/O	MFP1	Smart Card 0 data pin.
	SPI0_CLK	I/O	MFP2	SPI0 serial clock pin.

	Pin Name	Type	MFP	Description
	UART2_nCTS	I	MFP3	UART2 clear to Send input pin.
	USCI0_DAT0	I/O	MFP4	USCI0 data 0 pin.
	ACMP0_WLAT	I	MFP5	Analog comparator 0 window latch input pin
	PWM0_CH0	I/O	MFP6	PWM0 channel 0 output/capture input.
	EBI_AD8	I/O	MFP7	EBI address/data bus bit 8.
	INT2	I	MFP8	External interrupt 2 input pin.
PC.1	PC.1	I/O	MFP0	General purpose digital I/O pin.
	CLKO	O	MFP1	Clock Out
	SC0_CLK	O	MFP2	Smart Card 0 clock pin.
	UART2_nRTS	O	MFP3	UART2 request to Send output pin.
	USCI0_DAT1	I/O	MFP4	USCI0 data 1 pin.
	ACMP1_WLAT	I	MFP5	Analog comparator 1 window latch input pin
	PWM0_CH1	I/O	MFP6	PWM0 channel 1 output/capture input.
	EBI_AD9	I/O	MFP7	EBI address/data bus bit 9.
PC.2	PC.2	I/O	MFP0	General purpose digital I/O pin.
	SC0_RST	O	MFP1	Smart Card 0 reset pin.
	SPI0_SS	I/O	MFP2	SPI0 slave select pin.
	UART2_TXD	O	MFP3	UART2 data transmitter output pin.
	USCI0_CTL1	I/O	MFP4	USCI0 control 1 pin.
	ACMP1_O	O	MFP5	Analog comparator 1 output pin.
	PWM0_CH2	I/O	MFP6	PWM0 channel 2 output/capture input.
	EBI_AD10	I/O	MFP7	EBI address/data bus bit 10.
PC.3	PC.3	I/O	MFP0	General purpose digital I/O pin.
	SC0_PWR	O	MFP1	Smart Card 0 power pin.
	SPI0_MOSI	I/O	MFP2	SPI0 MOSI (Master Out, Slave In) pin.
	UART2_RXD	I	MFP3	UART2 data receiver input pin.
	USCI0_CTL0	I/O	MFP5	USCI0 control 0 pin.
	PWM0_CH3	I/O	MFP6	PWM0 channel 3 output/capture input.
	EBI_AD11	I/O	MFP7	EBI address/data bus bit 11.
PC.4	PC.4	I/O	MFP0	General purpose digital I/O pin.
	SC0_nCD	I	MFP1	Smart Card 0 card detect pin.
	SPI0_MISO	I/O	MFP2	SPI0 MISO (Master In, Slave Out) pin.

	Pin Name	Type	MFP	Description
	I2C1_SCL	I/O	MFP3	I <sup>2</sup> C1 clock pin.
	USCI0_CLK	I/O	MFP5	USCI0 clock pin.
	PWM0_CH4	I/O	MFP6	PWM0 channel 4 output/capture input.
	EBI_AD12	I/O	MFP7	EBI address/data bus bit 12.
PC.5	PC.5	I/O	MFP0	General purpose digital I/O pin.
	SPI0_I2SMCLK	I/O	MFP2	SPI0 I <sup>2</sup> S master clock output pin
	I2C1_SDA	I/O	MFP3	I <sup>2</sup> C1 data input/output pin.
	USCI0_DAT0	I/O	MFP4	USCI0 data 0 pin.
	PWM0_CH5	I/O	MFP6	PWM0 channel 5 output/capture input.
	EBI_AD13	I/O	MFP7	EBI address/data bus bit 13.
PC.6	PC.6	I/O	MFP0	General purpose digital I/O pin.
	USCI0_DAT1	I/O	MFP4	USCI0 data 1 pin.
	ACMP1_O	O	MFP5	Analog comparator 1 output pin.
	PWM1_CH0	I/O	MFP6	PWM1 channel 0 output/capture input.
	EBI_AD14	I/O	MFP7	EBI address/data bus bit 14.
PC.7	PC.7	I/O	MFP0	General purpose digital I/O pin.
	USCI0_CTL1	I/O	MFP4	USCI0 control 1 pin.
	PWM1_CH1	I/O	MFP6	PWM1 channel 1 output/capture input.
	EBI_AD15	I/O	MFP7	EBI address/data bus bit 15.
PC.8	PC.8	I/O	MFP0	General purpose digital I/O pin.
	ADC0_CH16	A	MFP1	ADC0 channel 16 analog input.
	UART0_nRTS	O	MFP3	UART0 request to Send output pin.
PC.9	PC.9	I/O	MFP0	General purpose digital I/O pin.
	SPI0_I2SMCLK	I/O	MFP2	SPI0 I <sup>2</sup> S master clock output pin
	I2C1_SCL	I/O	MFP3	I <sup>2</sup> C1 clock pin.
	USCI2_CTL1	I/O	MFP4	USCI2 control 1 pin.
	PWM1_CH0	I/O	MFP6	PWM1 channel 0 output/capture input.
PC.10	PC.10	I/O	MFP0	General purpose digital I/O pin.
	SPI0_MOSI	I/O	MFP2	SPI0 MOSI (Master Out, Slave In) pin.
	I2C1_SDA	I/O	MFP3	I <sup>2</sup> C1 data input/output pin.
	USCI2_DAT1	I/O	MFP4	USCI2 data 1 pin.
	PWM1_CH1	I/O	MFP6	PWM1 channel 1 output/capture input.



	Pin Name	Type	MFP	Description
PC.11	PC.11	I/O	MFP0	General purpose digital I/O pin.
	SPI0_MISO	I/O	MFP2	SPI0 MISO (Master In, Slave Out) pin.
	USCI2_CLK	I/O	MFP4	USCI2 clock pin.
	PWM1_CH2	I/O	MFP6	PWM1 channel 2 output/capture input.
PC.12	PC.12	I/O	MFP0	General purpose digital I/O pin.
	SPI0_CLK	I/O	MFP2	SPI0 serial clock pin.
	USCI2_CTL0	I/O	MFP4	USCI2 control 0 pin.
	PWM1_CH3	I/O	MFP6	PWM1 channel 3 output/capture input.
PC.13	PC.13	I/O	MFP0	General purpose digital I/O pin.
	SPI0_SS	I/O	MFP2	SPI0 slave select pin.
	USCI2_DAT0	I/O	MFP4	USCI2 data 0 pin.
	PWM1_CH4	I/O	MFP6	PWM1 channel 4 output/capture input.
PC.14	PC.14	I/O	MFP0	General purpose digital I/O pin.
	PWM1_CH5	I/O	MFP6	PWM1 channel 5 output/capture input.
PC.15	PC.15	I/O	MFP0	General purpose digital I/O pin.
	PWM1_CH0	I/O	MFP6	PWM1 channel 0 output/capture input.
PD.0	PD.0	I/O	MFP0	General purpose digital I/O pin.
	SPI0_I2SMCLK	I/O	MFP1	SPI0 I <sup>2</sup> S master clock output pin
	SPI1_I2SMCLK	I/O	MFP2	SPI1 I <sup>2</sup> S master clock output pin
	UART0_RXD	I	MFP3	UART0 data receiver input pin.
	USCI2_CTL0	I/O	MFP4	USCI2 control 0 pin.
	ACMP1_N	A	MFP5	Analog comparator 1 negative input pin.
	SC1_CLK	O	MFP6	Smart Card 1 clock pin.
	INT3	I	MFP8	External interrupt 3 input pin.
PD.1	PD.1	I/O	MFP0	General purpose digital I/O pin.
	ADC0_CH19	A	MFP1	ADC0 channel 19 analog input.
	PWM0_SYNC_IN	I	MFP2	PWM0 counter synchronous trigger input pin.
	UART0_TXD	O	MFP3	UART0 data transmitter output pin.
	USCI2_CLK	I/O	MFP4	USCI2 clock pin.
	ACMP1_P2	A	MFP5	Analog comparator 1 positive input 2 pin.
	TM0	I/O	MFP6	Timer0 event counter input/toggle output pin.
	EBI_nRD	O	MFP7	EBI read enable output pin.

	Pin Name	Type	MFP	Description
PD.2	PD.2	I/O	MFP0	General purpose digital I/O pin.
	ADC0_ST	I	MFP1	ADC0 external trigger input pin.
	TM0_EXT	I/O	MFP3	Timer0 external capture input/toggle output pin.
	USCI2_DAT0	I/O	MFP4	USCI2 data 0 pin.
	ACMP1_P1	A	MFP5	Analog comparator 1 positive input 1 pin.
	PWM0_BRAKE0	I	MFP6	PWM0 Brake 0 input pin.
	EBI_nWR	O	MFP7	EBI write enable output pin.
	INT0	I	MFP8	External interrupt 0 input pin.
PD.3	PD.3	I/O	MFP0	General purpose digital I/O pin.
	TM2	I/O	MFP1	Timer2 event counter input/toggle output pin.
	SPI0_I2SMCLK	I/O	MFP2	SPI0 I <sup>2</sup> S master clock output pin
	TM1_EXT	I/O	MFP3	Timer1 external capture input/toggle output pin.
	USCI2_DAT1	I/O	MFP4	USCI2 data 1 pin.
	ACMP1_P0	A	MFP5	Analog comparator 1 positive input 0 pin.
	PWM0_BRAKE1	I	MFP6	PWM0 Brake 1 input pin.
	EBI_MCLK	O	MFP7	EBI external clock output pin.
	INT1	I	MFP8	External interrupt 1 input pin.
PD.4	PD.4	I/O	MFP0	General purpose digital I/O pin.
	SPI1_CLK	I/O	MFP2	SPI1 serial clock pin.
	I2C0_SDA	I/O	MFP3	I <sup>2</sup> C0 data input/output pin.
	UART2_nRTS	O	MFP4	UART2 request to Send output pin.
	PWM0_BRAKE0	I	MFP5	PWM0 Brake 0 input pin.
	TM0	I/O	MFP6	Timer0 event counter input/toggle output pin.
PD.5	PD.5	I/O	MFP0	General purpose digital I/O pin.
	CLKO	O	MFP1	Clock Out
	SPI1_MISO	I/O	MFP2	SPI1 MISO (Master In, Slave Out) pin.
	I2C0_SCL	I/O	MFP3	I <sup>2</sup> C0 clock pin.
	UART2_nCTS	I	MFP4	UART2 clear to Send input pin.
	PWM0_BRAKE1	I	MFP5	PWM0 Brake 1 input pin.
	TM1	I/O	MFP6	Timer1 event counter input/toggle output pin.
PD.6	PD.6	I/O	MFP0	General purpose digital I/O pin.
	CLKO	O	MFP1	Clock Out

	Pin Name	Type	MFP	Description
	SPI1_SS	I/O	MFP2	SPI1 slave select pin.
	UART0_RXD	I	MFP3	UART0 data receiver input pin.
	UART2_TXD	O	MFP4	UART2 data transmitter output pin.
	ACMP0_O	O	MFP5	Analog comparator 0 output pin.
	PWM0_CH5	I/O	MFP6	PWM0 channel 5 output/capture input.
	EBI_nWR	O	MFP7	EBI write enable output pin.
PD.7	PD.7	I/O	MFP0	General purpose digital I/O pin.
	USCI1_CTL1	I/O	MFP1	USCI1 control 1 pin.
	SPI0_I2SMCLK	I/O	MFP2	SPI0 I <sup>2</sup> S master clock output pin
	PWM0_SYNC_IN	I	MFP3	PWM0 counter synchronous trigger input pin.
	TM1	I/O	MFP4	Timer1 event counter input/toggle output pin.
	ACMP0_O	O	MFP5	Analog comparator 0 output pin.
	PWM0_CH5	I/O	MFP6	PWM0 channel 5 output/capture input.
	EBI_nRD	O	MFP7	EBI read enable output pin.
PD.8	PD.8	I/O	MFP0	General purpose digital I/O pin.
	ADC0_CH17	A	MFP1	ADC0 channel 17 analog input.
	UART0_nCTS	I	MFP3	UART0 clear to Send input pin.
	USCI2_CTL1	I/O	MFP4	USCI2 control 1 pin.
	TM2	I/O	MFP6	Timer2 event counter input/toggle output pin.
	EBI_nCS0	O	MFP7	EBI chip select 0 output pin.
PD.9	PD.9	I/O	MFP0	General purpose digital I/O pin.
	ADC0_CH18	A	MFP1	ADC0 channel 18 analog input.
	UART0_RXD	I	MFP3	UART0 data receiver input pin.
	USCI2_CTL0	I/O	MFP4	USCI2 control 0 pin.
	ACMP1_P3	A	MFP5	Analog comparator 1 positive input 3 pin.
	TM3	I/O	MFP6	Timer3 event counter input/toggle output pin.
	EBI_ALE	O	MFP7	EBI address latch enable output pin.
PD.10	PD.10	I/O	MFP0	General purpose digital I/O pin.
	TM2	I/O	MFP4	Timer2 event counter input/toggle output pin.
	USCI2_DAT0	I/O	MFP5	USCI2 data 0 pin.
PD.11	PD.11	I/O	MFP0	General purpose digital I/O pin.
	TM3	I/O	MFP4	Timer3 event counter input/toggle output pin.

	Pin Name	Type	MFP	Description
	USCI2_DAT1	I/O	MFP5	USCI2 data 1 pin.
PD.12	PD.12	I/O	MFP0	General purpose digital I/O pin.
	USCI1_CTL0	I/O	MFP1	USCI1 control 0 pin.
	SPI1_SS	I/O	MFP2	SPI1 slave select pin.
	UART0_TXD	O	MFP3	UART0 data transmitter output pin.
	PWM1_CH0	I/O	MFP6	PWM1 channel 0 output/capture input.
	EBI_ADR16	O	MFP7	EBI address bus bit 16.
PD.13	PD.13	I/O	MFP0	General purpose digital I/O pin.
	USCI1_DAT1	I/O	MFP1	USCI1 data 1 pin.
	SPI1_MOSI	I/O	MFP2	SPI1 MOSI (Master Out, Slave In) pin.
	UART0_RXD	I	MFP3	UART0 data receiver input pin.
	PWM1_CH1	I/O	MFP6	PWM1 channel 1 output/capture input.
	EBI_ADR17	O	MFP7	EBI address bus bit 17.
PD.14	PD.14	I/O	MFP0	General purpose digital I/O pin.
	USCI1_DAT0	I/O	MFP1	USCI1 data 0 pin.
	SPI1_MISO	I/O	MFP2	SPI1 MISO (Master In, Slave Out) pin.
	UART0_nCTS	I	MFP3	UART0 clear to Send input pin.
	PWM1_CH2	I/O	MFP6	PWM1 channel 2 output/capture input.
	EBI_ADR18	O	MFP7	EBI address bus bit 18.
PD.15	PD.15	I/O	MFP0	General purpose digital I/O pin.
	USCI1_CLK	I/O	MFP1	USCI1 clock pin.
	SPI1_CLK	I/O	MFP2	SPI1 serial clock pin.
	UART0_nRTS	O	MFP3	UART0 request to Send output pin.
	PWM1_CH3	I/O	MFP6	PWM1 channel 3 output/capture input.
	EBI_ADR19	O	MFP7	EBI address bus bit 19.
PE.0	PE.0	I/O	MFP0	General purpose digital I/O pin.
	SPI0_CLK	I/O	MFP2	SPI0 serial clock pin.
	I2C1_SDA	I/O	MFP3	I <sup>2</sup> C1 data input/output pin.
	TM2_EXT	I/O	MFP4	Timer2 external capture input/toggle output pin.
	SC0_nCD	I	MFP5	Smart Card 0 card detect pin.
	PWM0_CH0	I/O	MFP6	PWM0 channel 0 output/capture input.
	EBI_nCS1	O	MFP7	EBI chip select 1 output pin.

	Pin Name	Type	MFP	Description
	INT4	I	MFP8	External interrupt 4 input pin.
PE.1	PE.1	I/O	MFP0	General purpose digital I/O pin.
	TM3_EXT	I/O	MFP3	Timer3 external capture input/toggle output pin.
	SC0_nCD	I	MFP5	Smart Card 0 card detect pin.
	PWM0_CH1	I/O	MFP6	PWM0 channel 1 output/capture input.
PE.2	PE.2	I/O	MFP0	General purpose digital I/O pin.
	ADC0_CH9	A	MFP1	ADC0 channel 9 analog input.
	UART1_nRTS	O	MFP4	UART1 request to Send output pin.
	TM_BRAKE3	I	MFP5	TM_BRAKE3 I Timer Brake * input pin.
	PWM0_CH2	I/O	MFP6	PWM0 channel 2 output/capture input.
	USCI0_CTL0	I/O	MFP8	USCI0 control 0 pin.
PE.3	PE.3	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MOSI	I/O	MFP2	SPI1 MOSI (Master Out, Slave In) pin.
	UART2_RXD	I	MFP4	UART2 data receiver input pin.
	PWM0_CH3	I/O	MFP6	PWM0 channel 3 output/capture input.
PE.4	PE.4	I/O	MFP0	General purpose digital I/O pin.
	I2C0_SCL	I/O	MFP2	I <sup>2</sup> C0 clock pin.
	I2C1_SCL	I/O	MFP3	I <sup>2</sup> C1 clock pin.
	USCI0_CTL0	I/O	MFP4	USCI0 control 0 pin.
	SC0_PWR	O	MFP5	Smart Card 0 power pin.
	PWM1_BRAKE0	I	MFP6	PWM1 Brake 0 input pin.
	EBI_nCS0	O	MFP7	EBI chip select 0 output pin.
	INT0	I	MFP8	External interrupt 0 input pin.
PE.5	PE.5	I/O	MFP0	General purpose digital I/O pin.
	I2C0_SDA	I/O	MFP2	I <sup>2</sup> C0 data input/output pin.
	I2C1_SDA	I/O	MFP3	I <sup>2</sup> C1 data input/output pin.
	USCI0_CLK	I/O	MFP4	USCI0 clock pin.
	SC0_RST	O	MFP5	Smart Card 0 reset pin.
	PWM1_BRAKE1	I	MFP6	PWM1 Brake 1 input pin.
	EBI_ALE	O	MFP7	EBI address latch enable output pin.
	INT1	I	MFP8	External interrupt 1 input pin.
PE.6	PE.6	I/O	MFP0	General purpose digital I/O pin.

	Pin Name	Type	MFP	Description
	ICE_CLK	I	MFP1	Serial wired debugger clock pin.
	I2C0_SCL	I/O	MFP2	I <sup>2</sup> C0 clock pin.
	UART0_RXD	I	MFP3	UART0 data receiver input pin.
PE.7	PE.7	I/O	MFP0	General purpose digital I/O pin.
	ICE_DAT	O	MFP1	Serial wired debugger data pin.
	I2C0_SDA	I/O	MFP2	I <sup>2</sup> C0 data input/output pin.
	UART0_TXD	O	MFP3	UART0 data transmitter output pin.
PE.8	PE.8	I/O	MFP0	General purpose digital I/O pin.
	UART1_TXD	O	MFP1	UART1 data transmitter output pin.
	TM0	I/O	MFP3	Timer0 event counter input/toggle output pin.
	I2C1_SCL	I/O	MFP4	I <sup>2</sup> C1 clock pin.
	SC0_PWR	O	MFP5	Smart Card 0 power pin.
PE.9	PE.9	I/O	MFP0	General purpose digital I/O pin.
	UART1_RXD	I	MFP1	UART1 data receiver input pin.
	TM1	I/O	MFP3	Timer1 event counter input/toggle output pin.
	I2C1_SDA	I/O	MFP4	I <sup>2</sup> C1 data input/output pin.
	SC0_RST	O	MFP5	Smart Card 0 reset pin.
PE.10	PE.10	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MISO	I/O	MFP1	SPI1 MISO (Master In, Slave Out) pin.
	SPI0_MISO	I/O	MFP2	SPI0 MISO (Master In, Slave Out) pin.
	UART1_nCTS	I	MFP3	UART1 clear to Send input pin.
	SC0_DAT	I/O	MFP5	Smart Card 0 data pin.
	SPI1_CLK	I/O	MFP6	SPI1 serial clock pin.
	EBI_AD7	I/O	MFP7	EBI address/data bus bit 7.
	TM0_EXT	I/O	MFP8	Timer0 external capture input/toggle output pin.
PE.11	PE.11	I/O	MFP0	General purpose digital I/O pin.
	SPI1_MOSI	I/O	MFP1	SPI1 MOSI (Master Out, Slave In) pin.
	SPI0_MOSI	I/O	MFP2	SPI0 MOSI (Master Out, Slave In) pin.
	UART1_nRTS	O	MFP3	UART1 request to Send output pin.
	SC0_CLK	O	MFP5	Smart Card 0 clock pin.
	SPI1_MISO	I/O	MFP6	SPI1 MISO (Master In, Slave Out) pin.
	EBI_AD6	I/O	MFP7	EBI address/data bus bit 6.

	Pin Name	Type	MFP	Description
	TM1_EXT	I/O	MFP8	Timer1 external capture input/toggle output pin.
PE.12	PE.12	I/O	MFP0	General purpose digital I/O pin.
	SPI1_SS	I/O	MFP1	SPI1 slave select pin.
	SPI0_SS	I/O	MFP2	SPI0 slave select pin.
	UART1_TXD	O	MFP3	UART1 data transmitter output pin.
	I2C0_SCL	I/O	MFP4	I <sup>2</sup> C0 clock pin.
	SPI1_MOSI	I/O	MFP6	SPI1 MOSI (Master Out, Slave In) pin.
	EBI_AD5	I/O	MFP7	EBI address/data bus bit 5.
	TM2_EXT	I/O	MFP8	Timer2 external capture input/toggle output pin.
PE.13	PE.13	I/O	MFP0	General purpose digital I/O pin.
	SPI1_CLK	I/O	MFP1	SPI1 serial clock pin.
	SPI0_CLK	I/O	MFP2	SPI0 serial clock pin.
	UART1_RXD	I	MFP3	UART1 data receiver input pin.
	I2C0_SDA	I/O	MFP4	I <sup>2</sup> C0 data input/output pin.
	SPI1_SS	I/O	MFP6	SPI1 slave select pin.
	EBI_AD4	I/O	MFP7	EBI address/data bus bit 4.
	TM3_EXT	I/O	MFP8	Timer3 external capture input/toggle output pin.
PF.0	PF.0	I/O	MFP0	General purpose digital I/O pin.
	X32_OUT	O	MFP1	External 32.768 kHz crystal output pin.
	USCI2_CTL1	I/O	MFP5	USCI2 control 1 pin.
	INT5	I	MFP8	External interrupt 5 input pin.
PF.1	PF.1	I/O	MFP0	General purpose digital I/O pin.
	X32_IN	I	MFP1	External 32.768 kHz crystal input pin.
	USCI2_CTL0	I/O	MFP5	USCI2 control 0 pin.
	PWM1_BRAKE0	I	MFP6	PWM1 Brake 0 input pin.
PF.2	PF.2	I/O	MFP0	General purpose digital I/O pin.
	USCI2_CLK	I/O	MFP5	USCI2 clock pin.
	PWM1_BRAKE1	I	MFP6	PWM1 Brake 1 input pin.
PF.3	PF.3	I/O	MFP0	General purpose digital I/O pin.
	XT1_OUT	O	MFP1	External 4~24 MHz (high speed) crystal output pin.
	I2C1_SCL	I/O	MFP3	I <sup>2</sup> C1 clock pin.
PF.4	PF.4	I/O	MFP0	General purpose digital I/O pin.

	Pin Name	Type	MFP	Description
	XT1_IN	I	MFP1	External 4~24 MHz (high speed) crystal input pin.
	I2C1_SDA	I/O	MFP3	I <sup>2</sup> C1 data input/output pin.
PF.5	PF.5	I/O	MFP0	General purpose digital I/O pin.
	TM3_EXT	I/O	MFP3	Timer3 external capture input/toggle output pin.
	SC1_nCD	I	MFP5	Smart Card 1 card detect pin.
	TM_BRAKE0	I	MFP6	TM_BRAKE0 I Timer Brake * input pin.
PF.6	PF.6	I/O	MFP0	General purpose digital I/O pin.
PF.7	PF.7	I/O	MFP0	General purpose digital I/O pin.

Table 4.3-1 NUC126 GPIO Multi-function Table



## 5 BLOCK DIAGRAM

### 5.1 NuMicro® NUC126 Block Diagram

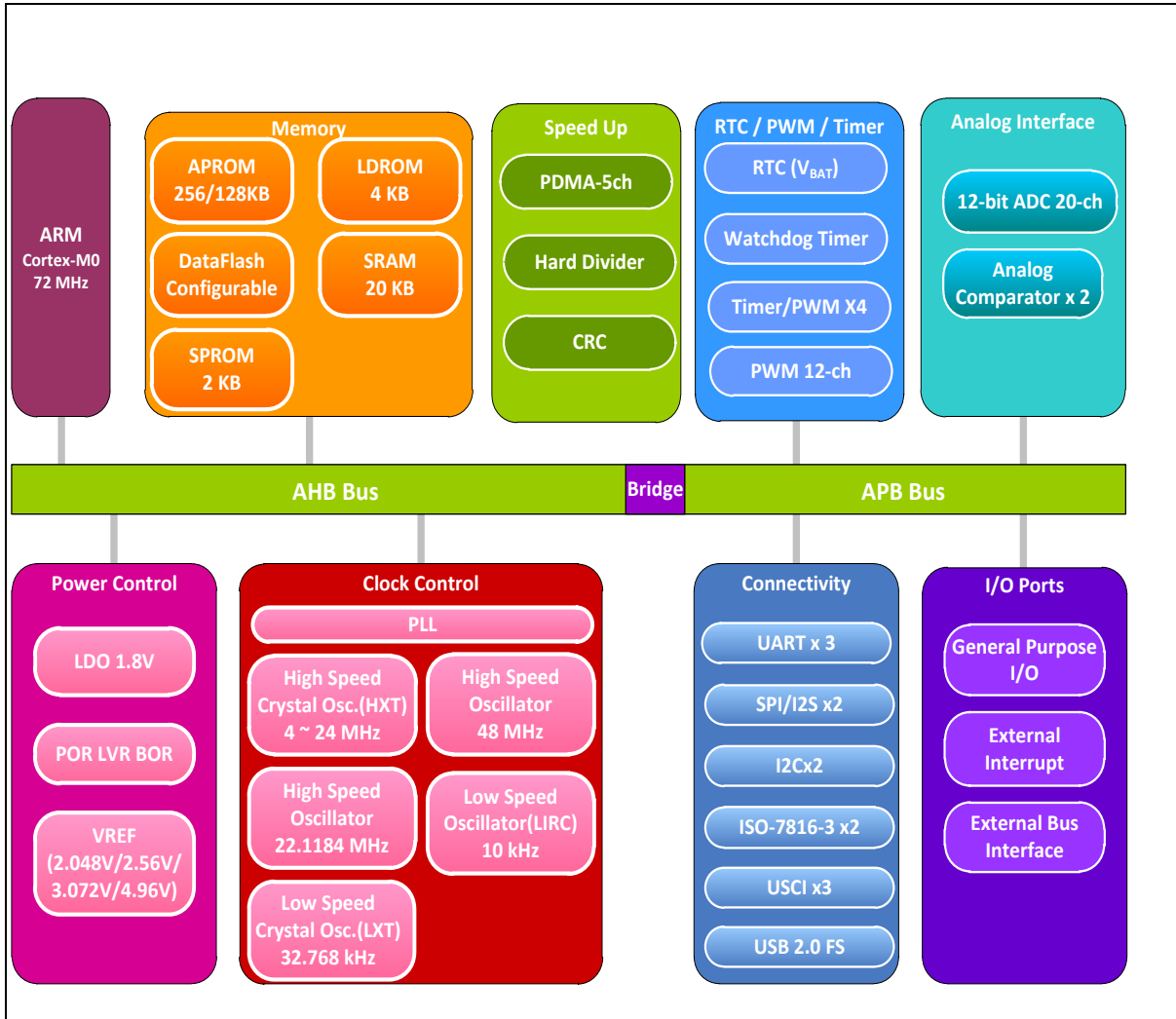


Figure 5.1-1 NuMicro® NUC126 Block Diagram

## 6 FUNCTIONAL DESCRIPTION

### 6.1 ARM® Cortex®-M0 Core

The Cortex®-M0 processor is a configurable, multistage, 32-bit RISC processor, which has an AMBA AHB-Lite interface and includes an NVIC component. It also has optional hardware debug functionality. The processor can execute Thumb code and is compatible with other Cortex®-M profile processor. The profile supports two modes -Thread mode and Handler mode. Handler mode is entered as a result of an exception. An exception return can only be issued in Handler mode. Thread mode is entered on Reset, and can be entered as a result of an exception return. Figure 6.1-1 shows the functional controller of processor.

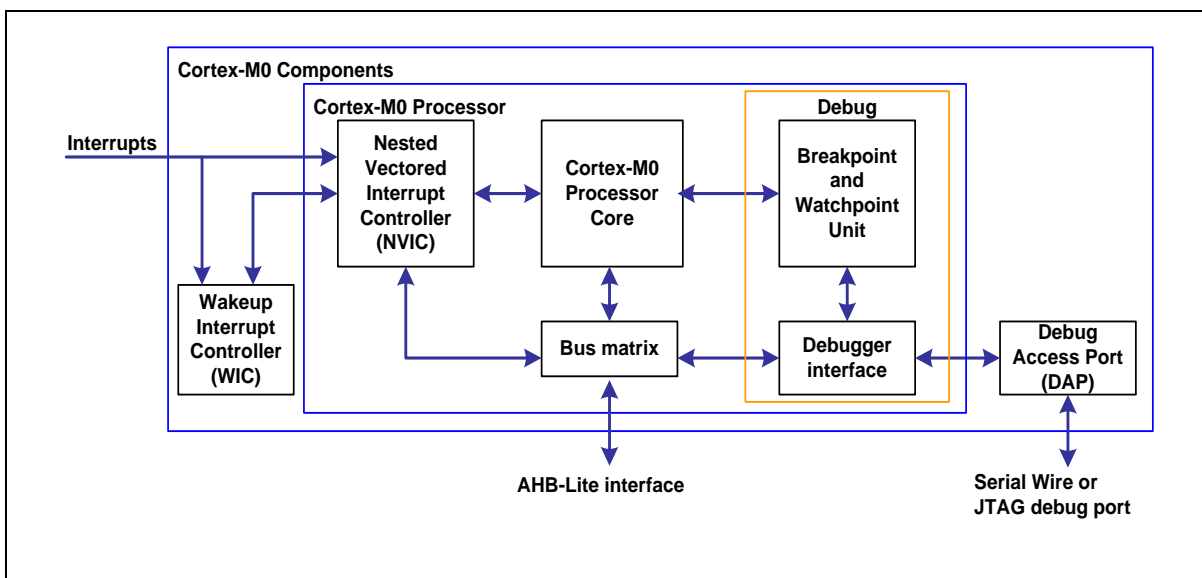


Figure 6.1-1 Functional Block Diagram

The implemented device provides:

- A low gate count processor:
  - ARMv6-M Thumb® instruction set
  - Thumb-2 technology
  - ARMv6-M compliant 24-bit SysTick timer
  - A 32-bit hardware multiplier
  - System interface supported with little-endian data accesses
  - Ability to have deterministic, fixed-latency, interrupt handling
  - Load/store-multiples and multicycle-multiplies that can be abandoned and restarted to facilitate rapid interrupt handling
  - C Application Binary Interface compliant exception model. This is the ARMv6-M, C Application Binary Interface (C-ABI) compliant exception model that enables the use of pure C functions as interrupt handlers
  - Low Power Sleep mode entry using the Wait For Interrupt (WFI), Wait For Event (WFE) instructions, or return from interrupt sleep-on-exit feature
- NVIC:
  - 32 external interrupt inputs, each with four levels of priority

- Dedicated Non-maskable Interrupt (NMI) input
- Supports for both level-sensitive and pulse-sensitive interrupt lines
- Supports Wake-up Interrupt Controller (WIC) and, providing Ultra-low Power Sleep mode
- Debug support:
  - Four hardware breakpoints
  - Two watchpoints
  - Program Counter Sampling Register (PCSR) for non-intrusive code profiling
  - Single step and vector catch capabilities
- Bus interfaces:
  - Single 32-bit AMBA-3 AHB-Lite system interface that provides simple integration to all system peripherals and memory
  - Single 32-bit slave port that supports the DAP (Debug Access Port)

## 6.2 System Manager

### 6.2.1 Overview

The system manager provides the functions of system control, power modes, wake-up sources, reset sources, system memory map, product ID and multi-function pin control. The following sections describe the functions for

- System Reset
- Power Modes and Wake-up Sources
- System Power Distribution
- SRAM Memory organization
- System Control Register for Part Number ID, Chip Reset and Multi-function Pin Control
- 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 (POR)
  - Low level on the nRESET pin
  - Watchdog Time-out Reset and Window Watchdog Reset (WDT/WWDT 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 (AIRCR[2])
  - CPU Reset for Cortex<sup>®</sup>-M0 core Only by writing 1 to CPURST (SYS\_IPRST0[1])

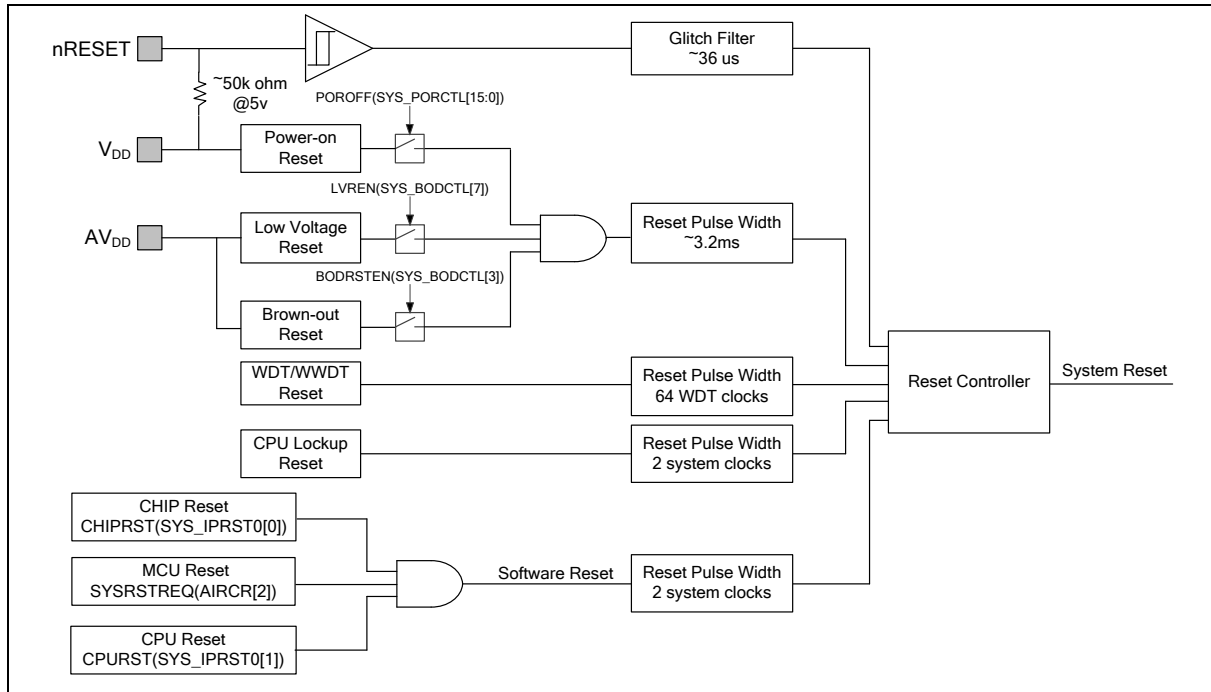


Figure 6.2-1 System Reset Sources

There are a total of 9 reset sources in the NuMicro<sup>®</sup> family. In general, CPU reset is used to reset Cortex<sup>®</sup>-M0 only; the other reset sources will reset Cortex<sup>®</sup>-M0 and all peripherals. However, there are small differences between each reset source and they are listed in Table 6.2-1.

Reset Sources Register									
<b>SYS_RSTSTS</b>	0x01	Bit 1 = 1	Bit 2 = 1	Bit 3 = 1	Bit 4 = 1	Bit 8 = 1	Bit 0 = 1	Bit 5 = 1	Bit 7 = 1
<b>CHIPRST (SYS_IPRST0[0])</b>	0x0	-	-	-	-	-	-	-	-
<b>BODEN (SYS_BODCTL[0])</b>	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	-	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	-
<b>BODVL (SYS_BODCTL[2:1])</b>									
<b>BODRSTEN (SYS_BODCTL[3])</b>									
<b>HXTEN (CLK_PWRCTL[0])</b>	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	
<b>LXTEN (CLK_PWRCTL[1])</b>	0x0	-	-	-	-	-	-	-	-
<b>WDTCKEN (CLK_APBCLK0[0])</b>	0x1	-	0x1	-	-	-	0x1	-	-
<b>HCLKSEL (CLK_CLKSEL0[2:0])</b>	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	-
<b>WDTSEL (CLK_CLKSEL1[1:0])</b>	0x3	0x3	-	-	-	-	-	-	-
<b>HXTSTB (CLK_STATUS[0])</b>	0x0	-	-	-	-	-	-	-	-
<b>LXTSTB (CLK_STATUS[1])</b>	0x0	-	-	-	-	-	-	-	-
<b>PLLSTB (CLK_STATUS[2])</b>	0x0	-	-	-	-	-	-	-	-
<b>HIRCSTB (CLK_STATUS[4])</b>	0x0	-	-	-	-	-	-	-	-
<b>CLKSFAIL (CLK_STATUS[7])</b>	0x0	0x0	-	-	-	-	-	-	-
<b>RSTEN (WDT_CTL[1])</b>	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	-	Reload from CONFIG0	-	-
<b>WDTEN (WDT_CTL[7])</b>									

<b>WDT_CTL</b> except bit 1 and bit 7.	0x0700	0x0700	0x0700	0x0700	0x0700	-	0x0700	-	-
<b>WDT_ALTCTL</b>	0x0000	0x0000	0x0000	0x0000	0x0000	-	0x0000	-	-
<b>WWDT_RLDCNT</b>	0x0000	0x0000	0x0000	0x0000	0x0000	-	0x0000	-	-
<b>WWDT_CTL</b>	0x3F0800	0x3F0800	0x3F0800	0x3F0800	0x3F0800	-	0x3F0800	-	-
<b>WWDT_STATUS</b>	0x0000	0x0000	0x0000	0x0000	0x0000	-	0x0000	-	-
<b>WWDT_CNT</b>	0x3F	0x3F	0x3F	0x3F	0x3F	-	0x3F	-	-
<b>BS</b> (FMC_ISPCTL[1])	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	-	Reload from CONFIG0	-	-
<b>BL</b> (FMC_ISPCTL[16])									
<b>FMC_DFBA</b>	Reload from CONFIG1	Reload from CONFIG1	Reload from CONFIG1	Reload from CONFIG1	Reload from CONFIG1	-	Reload from CONFIG1	-	-
<b>CBS</b> (FMC_ISPSTS[2:1])	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	-	Reload from CONFIG0	-	-
<b>VECMAP</b> (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	-	-
<b>Other Peripheral Registers</b>	Reset Value								-
<b>FMC Registers</b>	Reset Value								
<b>Note:</b> '-' 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<sub>DD</sub> and the state keeps longer than 36 us (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<sub>DD</sub> and the state keeps longer than 36 us (glitch filter). The PINRF(SYS\_RSTSTS[1]) will be set to 1 if the previous reset source is nRESET reset. Figure 6.2-2 shows the nRESET reset waveform.

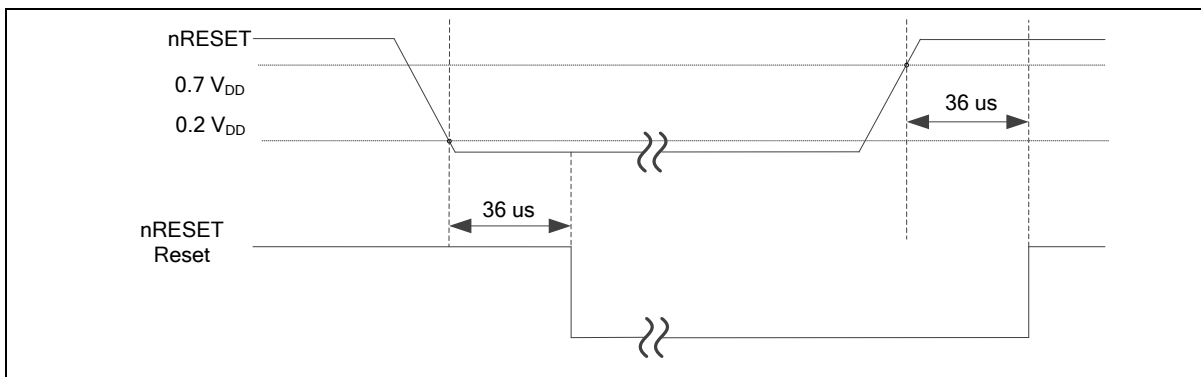


Figure 6.2-2 nRESET Reset Waveform

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.

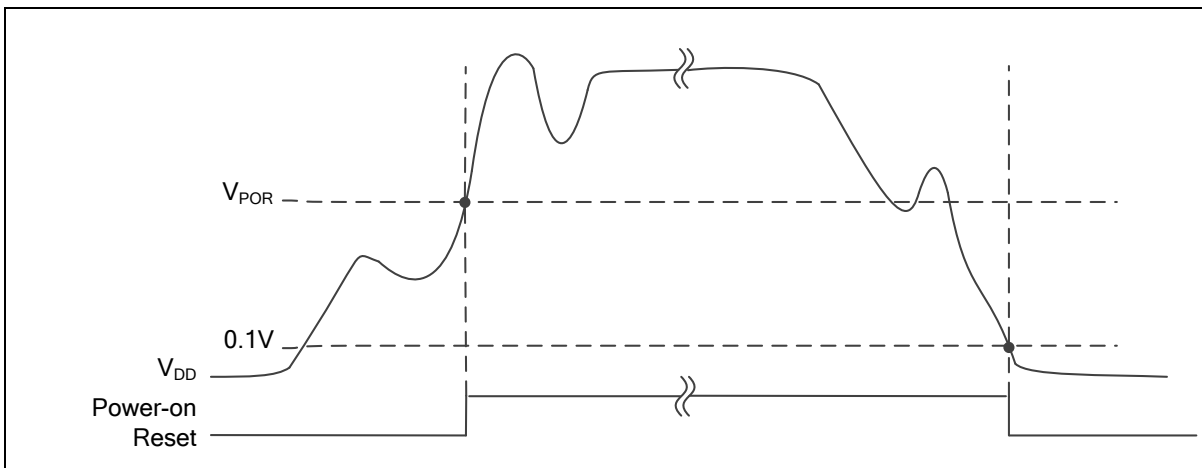


Figure 6.2-3 Power-on Reset (POR) Waveform

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 200us delay, LVR detection circuit will be stable and the LVR function will be active. Then LVR function will detect AV<sub>DD</sub> during system operation. When the AV<sub>DD</sub> voltage is lower than V<sub>LVR</sub> 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<sub>DD</sub> voltage rises above V<sub>LVR</sub> 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.



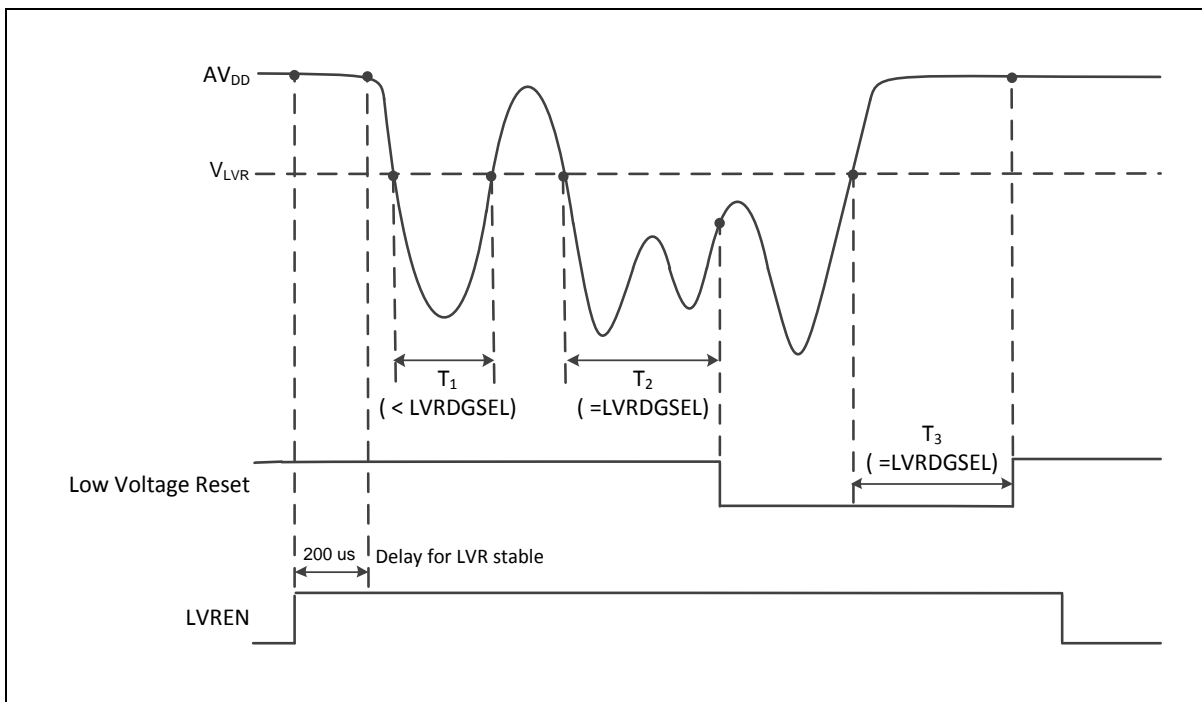


Figure 6.2-4 Low Voltage Reset (LVR) Waveform

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<sub>DD</sub> during system operation. When the AV<sub>DD</sub> voltage is lower than V<sub>BOD</sub> which is decided by BODEN (SYS\_BODCTL[0]) and BODVL (SYS\_BODCTL[2:1]) 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<sub>DD</sub> voltage rises above V<sub>BOD</sub> and the state keeps longer than De-glitch time set by BODDGSEL (SYS\_BODCTL[10:8]). The default value of BODEN, BODVL and BODRSTEN (SYS\_BODCTL[3]) is set by flash controller user configuration register CBODEN (CONFIG0 [23]), CBOV (CONFIG0 [22:21]) 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.

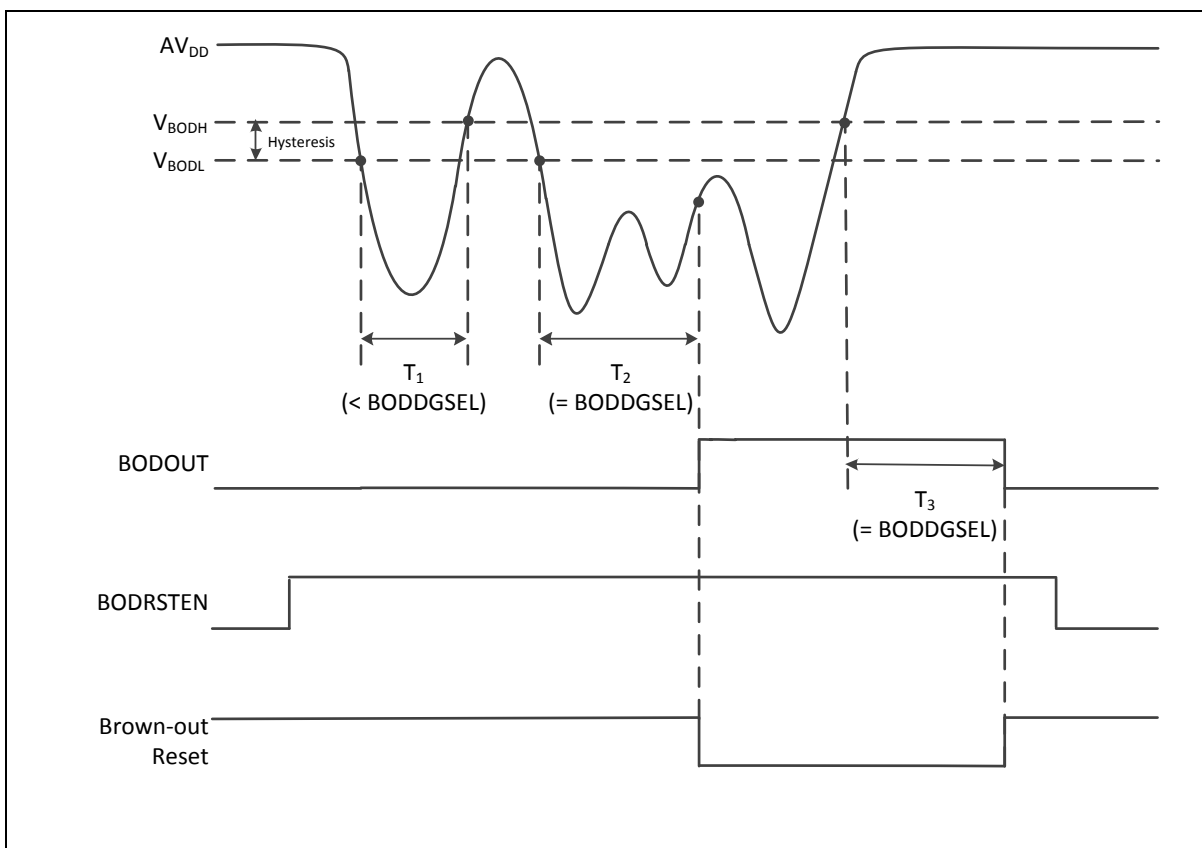


Figure 6.2-5 Brown-out Detector (BOD) Waveform

### 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<sup>®</sup>-M0 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 Power Modes and Wake-up Sources

There are several wake-up sources in Idle mode and Power-down mode. Table 6.2-2 lists the available clocks for each power mode.

Power Mode	Normal Mode	Idle Mode	Power-Down Mode
<b>Definition</b>	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 retained.
<b>Entry Condition</b>	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.
<b>Wake-up Sources</b>	N/A	All interrupts	RTC, WDT, I <sup>2</sup> C, Timer, UART, BOD, GPIO, EINT, USCI, USB, ACMP and EBOD.
<b>Available Clocks</b>	All	All except CPU clock	LXT and LIRC
<b>After Wake-up</b>	N/A	CPU back to normal mode	CPU back to normal mode

Table 6.2-2 Power Mode Difference Table

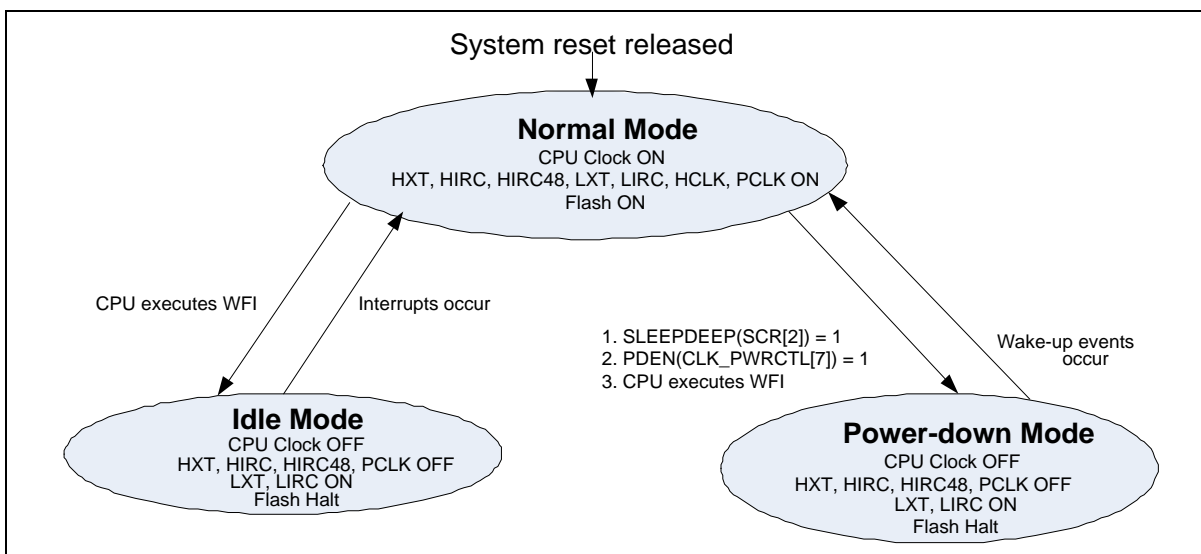


Figure 6.2-6 NuMicro® NUC126 Power Mode State Machine

1. LXT (32768 Hz XTL) ON or OFF depends on SW setting in normal mode.
2. LIRC (10 kHz OSC) ON or OFF depends on S/W 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
HXT (4~20 MHz XTL)	ON	ON	Halt
HIRC (22.1184 MHz OSC)	ON	ON	Halt
HIRC48 (48 MHz OSC)	ON	ON	Halt
LXT (32768 Hz XTL)	ON	ON	ON/OFF <sup>1</sup>
LIRC (10 kHz OSC)	ON	ON	ON/OFF <sup>2</sup>
PLL	ON	ON	Halt
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 <sup>3</sup>
PWM	ON	ON	Halt
WDT	ON	ON	ON/OFF <sup>4</sup>

WWDT	ON	ON	Halt
RTC	ON	ON	ON/OFF <sup>5</sup>
UART	ON	ON	ON/OFF <sup>6</sup>
SC	ON	ON	Halt
USCI	ON	ON	Halt
I <sup>2</sup> C	ON	ON	Halt
SPI	ON	ON	Halt
USB	ON	ON	Halt
ADC	ON	ON	Halt
ACMP	ON	ON	Halt

Table 6.2-3 Clocks in Power Modes

**Wake-up sources in Power-down mode:**

RTC, WDT, I<sup>2</sup>C, Timer, UART, USCI, BOD, EBOD, GPIO, USB, and ACMP.

After chip enters power down, the following wake-up sources can wake chip up to normal mode. Table 6.2-4 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	System Can Enter Power-Down Mode Again Condition*
BOD	Brown-Out Detector Interrupt	After software writes 1 to clear BODIF (SYS_BODCTL[4]).
EBOD	External Brown-Out Detector Interrupt	After software writes 1 to clear EBODIF (SYS_BODCTL[19]).
GPIO	GPIO Interrupt	After software write 1 to clear the Px_INTSRC[n] bit.
TIMER	Timer Interrupt	After software writes 1 to clear TWKF (TIMERx_INTSTS[1]) and TIF (TIMERx_INTSTS[0]).
WDT	WDT Interrupt	After software writes 1 to clear WKF (WDT_CTL[5]) (Write Protect).
RTC	Alarm Interrupt	After software writes 1 to clear ALMIF (RTC_INTSTS[0]).
	Time Tick Interrupt	After software writes 1 to clear TICKIF (RTC_INTSTS[1]).
UART	nCTS wake-up	After software writes 1 to clear CTSWKF (UARTx_WKSTS[0]).
	RX Data wake-up	After software writes 1 to clear DATWKF (UARTx_WKSTS[1]).
	Received FIFO Threshold Wake-up	After software writes 1 to clear RFRTWKF (UARTx_WKSTS[2]).
	RS-485 AAD Mode Wake-up	After software writes 1 to clear RS485WKF (UARTx_WKSTS[3]).
	Received FIFO Threshold Time-out Wake-up	After software writes 1 to clear TOUTWKF (UARTx_WKSTS[4]).
USCI UART	CTS Toggle	After software writes 1 to clear WKF (UUART_WKSTS[0]).
	Data Toggle	After software writes 1 to clear WKF (UUART_WKSTS[0]).
USCI I2C	Data toggle	After software writes 1 to clear WKF (UI2C_WKSTS[0]).

	Address match	After software writes 1 to clear WKAKDONE (UI2C_PROTSTS[16]), then writes 1 to clear WKF (UI2C_WKSTS[0]).
USCI SPI	SS Toggle	After software writes 1 to clear WKF (USPI_WKSTS[0]).
I <sup>2</sup> C	Address match wake-up	After software writes 1 to clear WKAKDONE (I2C_WKSTS[1]). Then software writes 1 to clear WKIF (I2C_WKSTS[0]).
USBD	Remote Wake-up	After software writes 1 to clear BUSIF (USBD_INTSTS[0]).
ACMP	Comparator Power-Down Wake-Up Interrupt	After software writes 1 to clear WKIF0 (ACMP_STATUS[8]) and WKIF1 (ACMP_STATUS[9]).

Table 6.2-4 Condition of Entering Power-down Mode Again

### 6.2.4 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. The  $V_{REF}$  should be connected with an external 1uF capacitor that should be located close to the  $V_{REF}$  pin to avoid power noise for analog applications.
- Digital power from  $V_{DD}$  and  $V_{SS}$  supplies the power to the internal regulator which provides a fixed 1.8 V 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  $V_{BAT}$  provides the power for RTC.
- A dedicated power from  $V_{DDIO}$  supplies the power for PE.8 ~ PE.13.

The outputs of internal voltage regulators, LDO and VDD33, require an external capacitor which should be located close to the corresponding pin. Analog power ( $AV_{DD}$ ) should be the same voltage level of the digital power ( $V_{DD}$ ). Figure 6.2-7 shows the power distribution of the NUC126 series.

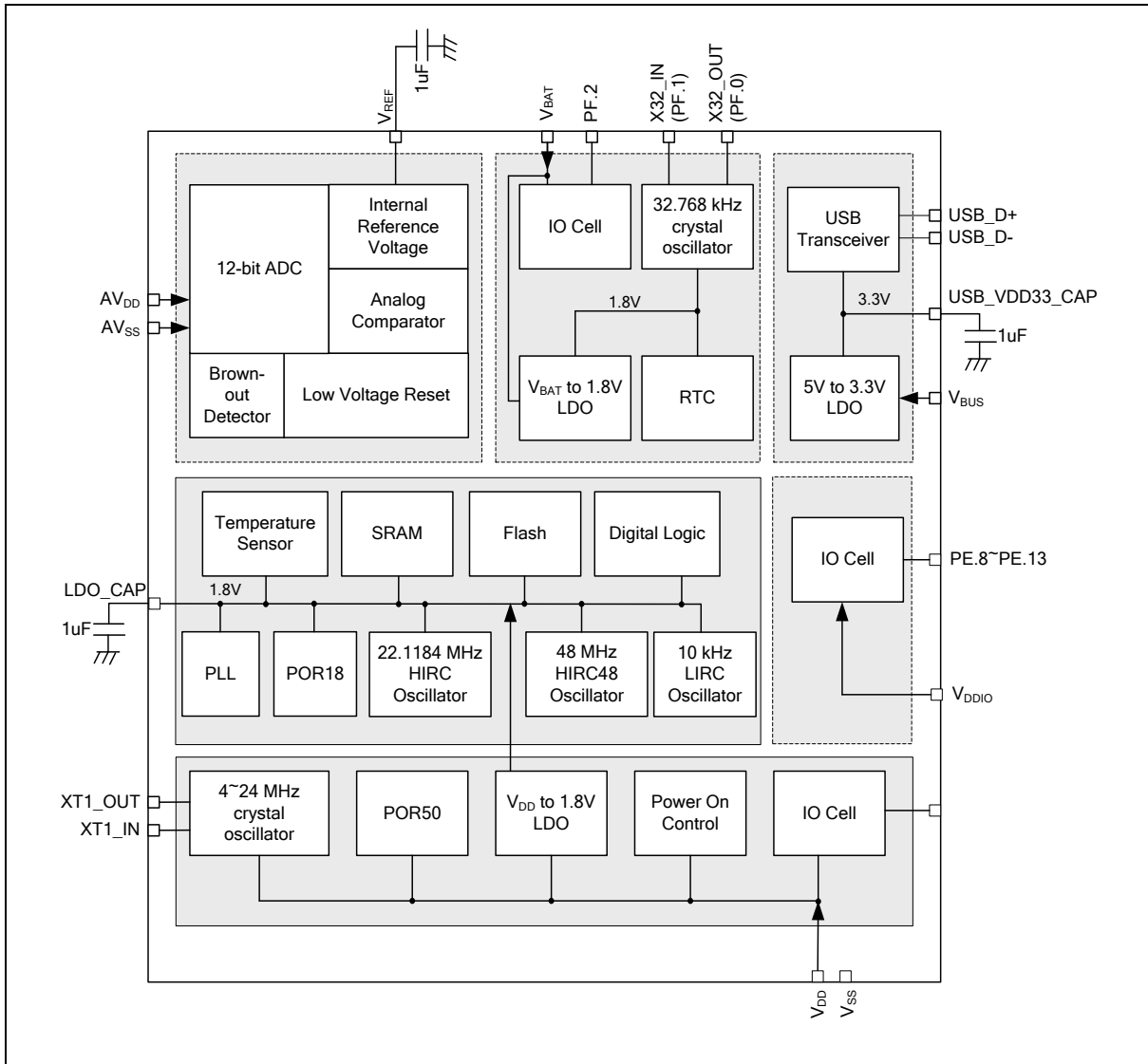


Figure 6.2-7 NuMicro® NUC126 Power Distribution Diagram

### 6.2.5 System Memory Map

The NUC126 series provides 4G-byte addressing space. The memory locations assigned to each on-chip controllers are shown in Table 6.2-5. The detailed register definition, memory space, and programming will be described in the following sections for each on-chip peripheral. The NUC126 series only supports little-endian data format.

Address Space	Token	Controllers
<b>Flash and SRAM Memory Space</b>		
0x0000_0000 – 0x0001_FFFF	FLASH_BA	FLASH Memory Space (128 KB)
0x0000_0000 – 0x0003_FFFF	FLASH_BA	FLASH Memory Space (256 KB)
0x0004_0000 – 0x0005_FFFF	Reserved	Reserved
0x0006_0000 – 0x0007_FFFF	Reserved	Reserved
0x2000_0000 – 0x2000_4FFF	SRAM_BA	SRAM Memory Space (20 KB)
0x2000_4000 – 0x2000_BFFF	Reserved	Reserved
0x2000_C000 – 0x2000_FFFF	Reserved	Reserved
0x6000_0000 – 0x601F_FFFF	EXTMEM_BA	External Memory Space for EBI Interface (2 MB)
<b>AHB Controllers Space (0x5000_0000 – 0x501F_FFFF)</b>		
0x5000_0000 – 0x5000_01FF	SYS_BA	System Control Registers
0x5000_0200 – 0x5000_02FF	CLK_BA	Clock Control Registers
0x5000_0300 – 0x5000_03FF	INT_BA	Interrupt Multiplexer Control Registers
0x5000_4000 – 0x5000_7FFF	GPIO_BA	GPIO Control Registers
0x5000_8000 – 0x5000_BFFF	PDMA_BA	Peripheral DMA Control Registers
0x5000_C000 – 0x5000_FFFF	FMC_BA	Flash Memory Control Registers
0x5001_0000 – 0x5001_03FF	EBI_BA	EBI Control Registers
0x5001_4000 – 0x5001_7FFF	HDIV_BA	Hardware Divider Registers
0x5001_8000 – 0x5001_FFFF	CRC_BA	CRC Generator Registers
<b>Peripheral Controllers Space (0x4000_0000 – 0x401F_FFFF)</b>		
0x4000_4000 – 0x4000_7FFF	WDT_BA	Watchdog Timer Control Registers
0x4000_8000 – 0x4000_BFFF	RTC_BA	Real Time Clock (RTC) Control Register
0x4001_0000 – 0x4001_3FFF	TMR01_BA	Timer0/Timer1 Control Registers
0x4002_0000 – 0x4002_3FFF	I2C0_BA	I <sup>2</sup> C0 Interface Control Registers
0x4003_0000 – 0x4003_3FFF	SPI0_BA	SPI0 with master/slave function Control Registers
0x4003_4000 – 0x4003_7FFF	SPI1_BA	SPI1 with master/slave function Control Registers
0x4004_0000 – 0x4004_3FFF	PWM0_BA	PWM0 Control Registers
0x4004_4000 – 0x4004_7FFF	Reserved	Reserved
0x4005_0000 – 0x4005_3FFF	UART0_BA	UART0 Control Registers
0x4006_0000 – 0x4006_3FFF	USB_D_BA	USB 2.0 FS device Controller Registers
0x4007_0000 – 0x4007_3FFF	USCI0_BA	USCI0 Control Registers



0x4007_4000 – 0x4007_7FFF	USCI2_BA	USCI2 Control Registers
0x400D_0000 – 0x400D_3FFF	ACMP01_BA	Analog Comparator Control Registers
0x400D_4000 – 0x400D_7FFF	Reserved	Reserved
0x400E_0000 – 0x400E_FFFF	ADC_BA	Analog-Digital-Converter (ADC) Control Registers
0x4010_0000 – 0x4010_3FFF	Reserved	Reserved
0x4011_0000 – 0x4011_3FFF	TMR23_BA	Timer2/Timer3 Control Registers
0x4012_0000 – 0x4012_3FFF	I2C1_BA	I <sup>2</sup> C1 Interface Control Registers
0x4014_0000 – 0x4014_3FFF	PWM1_BA	PWM1 Control Registers
0x4014_4000 – 0x4014_7FFF	Reserved	Reserved
0x4015_0000 – 0x4015_3FFF	UART1_BA	UART1 Control Registers
0x4015_4000 – 0x4015_7FFF	UART2_BA	UART2 Control Registers
0x4017_0000 – 0x4017_3FFF	USCI1_BA	USCI1 Control Registers
0x4017_4000 – 0x4017_7FFF	Reserved	Reserved
0x4019_0000 – 0x4019_3FFF	SC0_BA	SC0 Control Registers
0x4019_4000 – 0x4019_7FFF	SC1_BA	SC1 Control Registers
0x401A_0000 – 0x401A_3FFF	Reserved	Reserved
<b>System Controllers Space (0xE000_E000 ~ 0xE000_EFFF)</b>		
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 6.2-5 Address Space Assignments for On-Chip Controllers

### 6.2.6 SRAM Memory Organization

The NUC126 supports embedded SRAM with total 20 Kbytes size in one bank.

- Supports total 20 Kbytes SRAM
- Supports byte / half word / word write
- Supports oversize response error

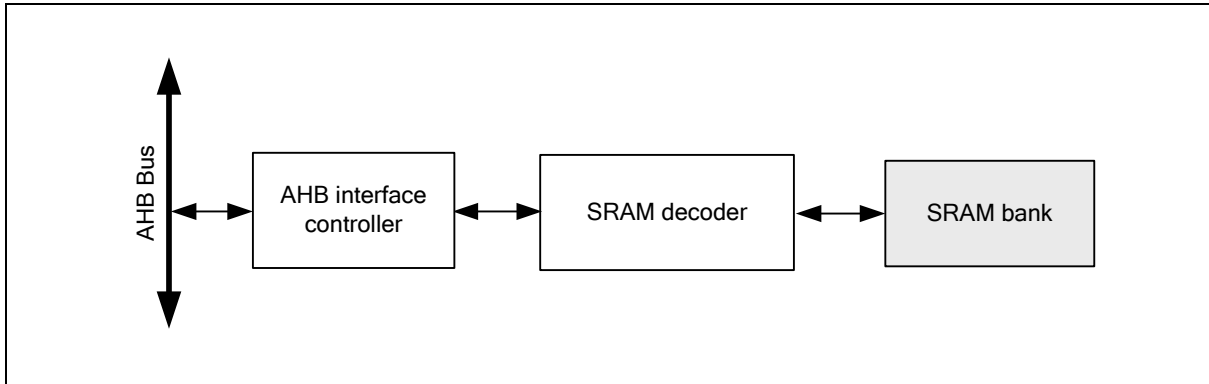


Figure 6.2-8 SRAM Block Diagram

Figure 6.2-9 shows the SRAM organization of NUC126. There is one SRAM bank in the NUC126 and addressed to 20 Kbytes. The address space is from 0x2000\_0000 to 0x2000\_4FFF. The address between 0x2000\_5000 to 0x3FFF\_FFFF is illegal memory space and chip will enter hardfault if CPU accesses these illegal memory addresses.

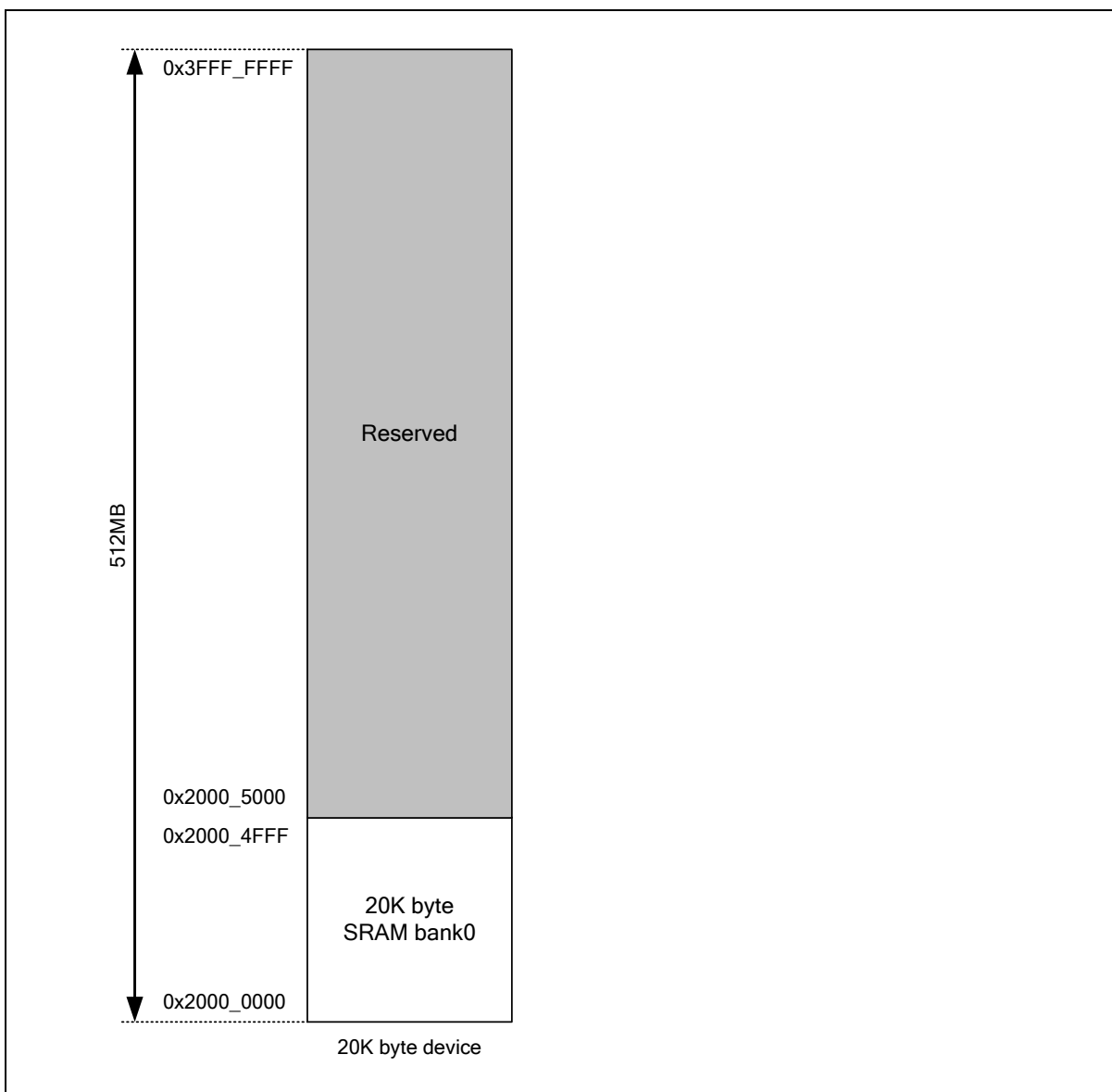


Figure 6.2-9 SRAM Memory Organization

### 6.2.7 Register Lock

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 0x5000\_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 0x5000\_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 “0x5000\_0100” to enable register protection.

### 6.2.8 Auto Trim

This chip supports auto-trim function: the HIRC trim (48 MHz and 22.1184 MHz RC oscillator),

according to the accurate external 32.768 kHz crystal oscillator or internal USB synchronous mode, automatically gets accurate HIRC output frequency, 0.25 % deviation within all temperature ranges.

For instance, the system needs an accurate 22.1184 MHz clock. In such case, if users do not want to use PLL as the system clock source, they need to solder 32.768 kHz crystal in system, and set FREQSEL (SYS\_IRCTCTL0[1:0] trim frequency selection) to “01”, set REFCKSEL (SYS\_IRCTCTL0[9] reference clock selection) to “0”, and the auto-trim function will be enabled. Interrupt status bit FREQLOCK (SYS\_IRCTISTS[0] HIRC frequency lock status) “1” indicates the HIRC0 output frequency is accurate within 0.25% deviation. To get better results, it is recommended to set both LOOPSEL (SYS\_IRCTCTL[5:4] trim calculation loop) and RETRYCNT (SYS\_IRCTCTL[7:6] trim value update limitation count) to “11”.

Another example is that the system needs an accurate 48 MHz clock for USB application. In such case, if neither using use PLL as the system clock source nor soldering 32.768 kHz crystal in system, user has to set REFCKSEL (SYS\_IRCTCTL1[10] reference clock selection) to “1”, set FREQSEL (SYS\_IRCTCTL1[1:0] trim frequency selection) to “10”, and the auto-trim function will be enabled. Interrupt status bit FREQLOCK1 (SYS\_IRCTISTS[8] HIRC frequency lock status) “1” indicates the HIRC1 output frequency is accurate within 0.25% deviation.

### 6.2.9 UART1\_TXD modulation with PWM

This chip supports UART1\_TXD to modulate with PWM channel. User can set MODPWMSEL(SYS\_MODCTL[6:4]) to choice which PWM0 channel to modulate with UART1\_TXD and set MODEN(SYS\_MODCTL[0]) to enable modulation function. User can set TXDINV(UART\_LINE[8]) to inverse UART1\_TXD before moulating with PWM.

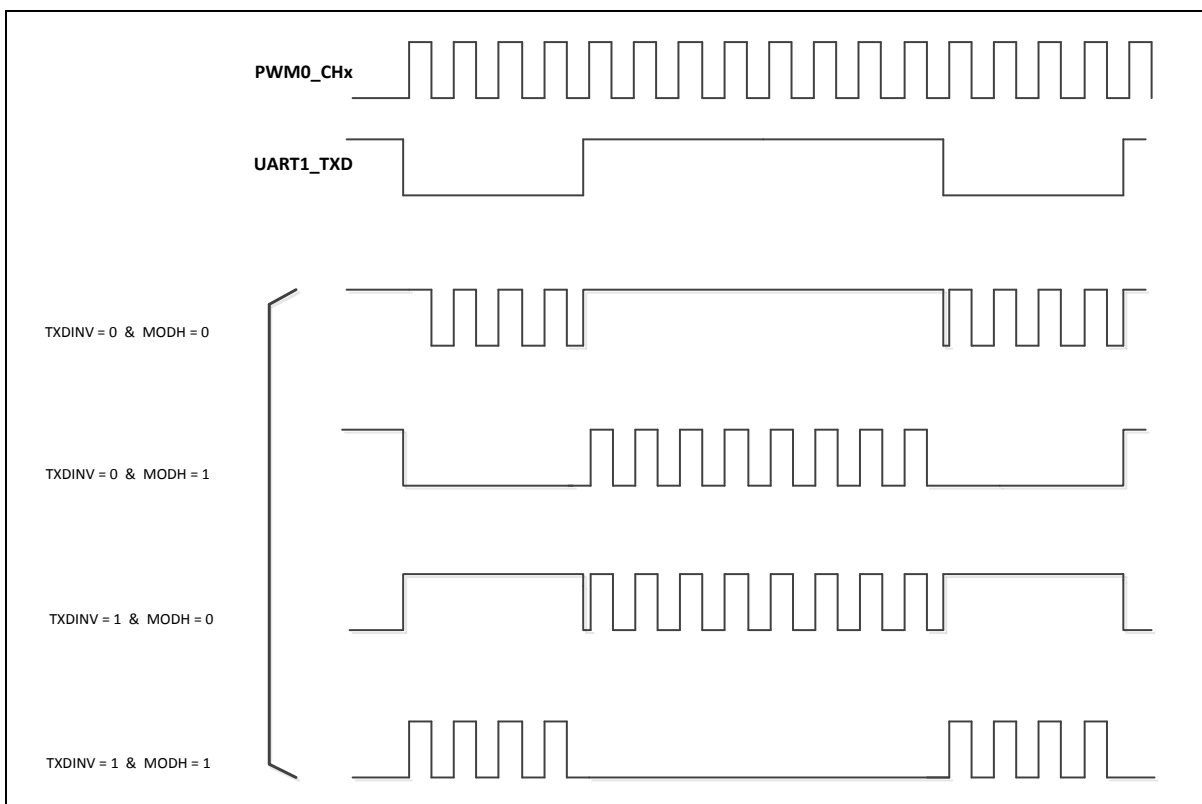


Figure 6.2-10 UART1\_TXD Modulated with PWM Channel

**6.2.10 Voltage Detector (VDET)**

This chip supports low power comparator to detect external voltage. User can control Bandgap active interval and comparator active interval to achieve low power detection purpose. There is no debounce function in Power-down mode since no HCLK available in Power-down mode.

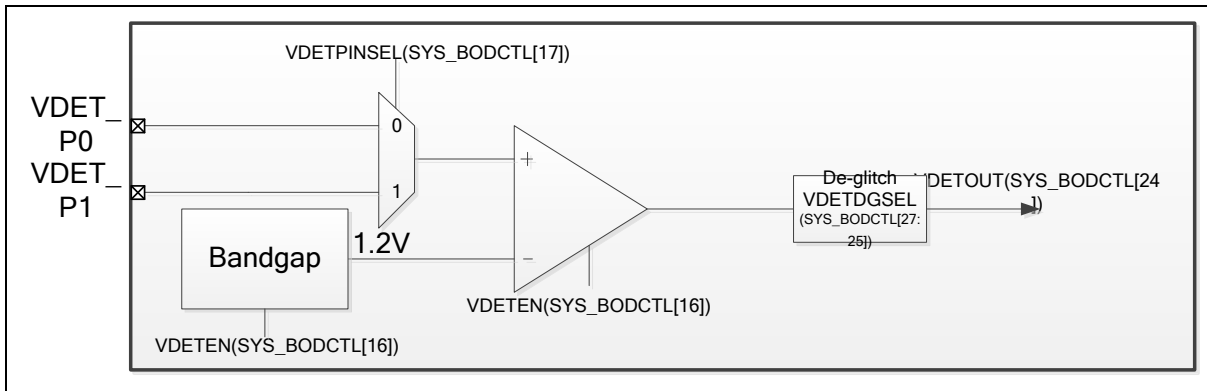


Figure 6.2-11 VDET Block Diagram

### 6.2.11 Register Map

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

Register	Offset	R/W	Description	Reset Value
<b>SYS Base Address:</b> SYS_BA = 0x5000_0000				
SYS_PDID	SYS_BA+0x00	R	Part Device Identification Number Register	0XXXXX_XXXX <sup>[1]</sup>
SYS_RSTSTS	SYS_BA+0x04	R/W	System Reset Status Register	0x0000_004B
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_BODCTL	SYS_BA+0x18	R/W	Brown-out Detector Control Register	0x0000_038X
SYS_IVSCTL	SYS_BA+0x1C	R/W	Internal Voltage Source Control Register	0x0000_0000
SYS_PORCTL	SYS_BA+0x24	R/W	Power-on Reset Controller Register	0x0000_0000
SYS_VREFCTL	SYS_BA+0x28	R/W	V <sub>REF</sub> Control Register	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_GPD_MFPL	SYS_BA+0x48	R/W	GPIOD Low Byte Multiple Function Control Register	0x0000_0000
SYS_GPD_MFPH	SYS_BA+0x4C	R/W	GPIOD High Byte Multiple Function Control Register	0x0000_0000
SYS_GPE_MFPL	SYS_BA+0x50	R/W	GPIOE Low Byte Multiple Function Control Register	0x1100_0000
SYS_GPE_MFPH	SYS_BA+0x54	R/W	GPIOE High Byte Multiple Function Control Register	0x0000_0000
SYS_GPF_MFPL	SYS_BA+0x58	R/W	GPIOF Low Byte Multiple Function Control Register	0x000X_X000
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_IRCTCTL0	SYS_BA+0x80	R/W	HIRC0 Trim Control Register	0x0000_0000
SYS_IRCTIEN	SYS_BA+0x84	R/W	HIRC Trim Interrupt Enable Register	0x0000_0000
SYS_IRCTISTS	SYS_BA+0x88	R/W	HIRC Trim Interrupt Status Register	0x0000_0000

<b>SYS_IRCTL1</b>	SYS_BA+0x90	R/W	HIRC1 Trim Control Register	0x0000_0400
<b>SYS_REGLCTL</b>	SYS_BA+0x100	R/W	Register Lock Control Register	0x0000_0000
<b>SYS_TSOFFSET</b>	SYS_BA+0x114	R	Temperature Sensor Offset Register	0x0000_0XXX

6.2.12 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]

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

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

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

NUC126 USB Series	
Part Number	PDID
NUC126VG4AE	0x00C0_5231
NUC126SG4AE	0x00C0_5212
NUC126SE4AE	0x00C0_5213
NUC126LG4AE	0x00C0_5204
NUC126LE4AE	0x00C0_5205



**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_004B

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	Reserved	MCURF	BODRF	LVRF	WDTRF	PINRF	PORF

Bits	Description	
[31:9]	Reserved	Reserved.
[8]	CPULKRF	<p><b>CPU Lockup Reset Flag</b></p> <p>The CPU lockup reset flag is set by hardware if Cortex®-M0 lockup happened.</p> <p>0 = No reset from CPU lockup happened.</p> <p>1 = The Cortex®-M0 lockup happened and chip is reset.</p> <p><b>Note:</b> This bit can be cleared by software writing '1'.</p>
[7]	CPURF	<p><b>CPU Reset Flag</b></p> <p>The CPU reset flag is set by hardware if software writes CPURST (SYS_IPRST0[1]) 1 to reset Cortex®-M0 Core and Flash Memory Controller (FMC).</p> <p>0 = No reset from CPU.</p> <p>1 = The Cortex®-M0 Core and FMC are reset by software setting CPURST to 1.</p> <p><b>Note:</b> This bit can be cleared by software writing '1'.</p>
[6]	Reserved	Reserved.
[5]	MCURF	<p><b>MCU Reset Flag</b></p> <p>The MCU reset flag is set by the "Reset Signal" from the Cortex®-M0 Core to indicate the previous reset source.</p> <p>0 = No reset from Cortex®-M0.</p> <p>1 = The Cortex®-M0 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®-M0 core.</p> <p><b>Note:</b> This bit can be cleared by software writing '1'.</p>

Bits	Description	
[4]	<b>BODRF</b>	<p><b>BOD Reset Flag</b></p> <p>The BOD reset flag is set by the “Reset Signal” from the Brown-out Detector to indicate the previous reset source.</p> <p>0 = No reset from BOD.</p> <p>1 = The BOD had issued the reset signal to reset the system.</p> <p><b>Note:</b> This bit can be cleared by software writing ‘1’.</p>
[3]	<b>LVRF</b>	<p><b>LVR Reset Flag</b></p> <p>The LVR reset flag is set by the “Reset Signal” from the Low Voltage Reset Controller to indicate the previous reset source.</p> <p>0 = No reset from LVR.</p> <p>1 = LVR controller had issued the reset signal to reset the system.</p> <p><b>Note:</b> This bit can be cleared by software writing ‘1’.</p>
[2]	<b>WDTRF</b>	<p><b>WDT Reset Flag</b></p> <p>The WDT reset flag is set by the “Reset Signal” from the Watchdog Timer or Window Watchdog Timer to indicate the previous reset source.</p> <p>0 = No reset from watchdog timer or window watchdog timer.</p> <p>1 = The watchdog timer or window watchdog timer had issued the reset signal to reset the system.</p> <p><b>Note1:</b> This bit can be cleared by software writing ‘1’.</p> <p><b>Note2:</b> 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.</p>
[1]	<b>PINRF</b>	<p><b>nRESET Pin Reset Flag</b></p> <p>The nRESET pin reset flag is set by the “Reset Signal” from the nRESET Pin to indicate the previous reset source.</p> <p>0 = No reset from nRESET pin.</p> <p>1 = Pin nRESET had issued the reset signal to reset the system.</p> <p><b>Note:</b> This bit can be cleared by software writing ‘1’.</p>
[0]	<b>PORF</b>	<p><b>POR Reset Flag</b></p> <p>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.</p> <p>0 = No reset from POR or CHIPRST.</p> <p>1 = Power-on Reset (POR) or CHIPRST had issued the reset signal to reset the system.</p> <p><b>Note:</b> This bit can be cleared by software writing ‘1’.</p>

**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		HDIVRST	EBIRST	PDMARST	CPURST	CHIPRST

Bits	Description	
[31:8]	Reserved	Reserved.
[7]	CRCRST	<p><b>CRC Calculation Controller Reset (Write Protect)</b> Set 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. <b>Note:</b> This bit is write protected. Refer to the SYS_REGLCTL register.</p>
[6:5]	Reserved	Reserved.
[4]	HDIVRST	<p><b>HDIV Controller Reset (Write Protect)</b> Set this bit to 1 will generate a reset signal to the HDIV controller. User needs to set this bit to 0 to release from the reset state. 0 = HDIV controller normal operation. 1 = HDIV controller reset. <b>Note:</b> This bit is write protected. Refer to the SYS_REGLCTL register.</p>
[3]	EBIRST	<p><b>EBI Controller Reset (Write Protect)</b> Set this bit to 1 will generate a reset signal to the EBI. User needs to set this bit to 0 to release from the reset state. 0 = EBI controller normal operation. 1 = EBI controller reset. <b>Note:</b> This bit is write protected. Refer to the SYS_REGLCTL register.</p>
[2]	PDMARST	<p><b>PDMA Controller Reset (Write Protect)</b> 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. <b>Note:</b> This bit is write protected. Refer to the SYS_REGLCTL register.</p>
[1]	CPURST	<p><b>Processor Core One-shot Reset (Write Protect)</b> Setting this bit will only reset the processor core and Flash Memory Controller(FMC), and</p>

		<p>this bit will automatically return to 0 after the 2 clock cycles.</p> <p>0 = Processor core normal operation.</p> <p>1 = Processor core one-shot reset.</p> <p><b>Note:</b> This bit is write protected. Refer to the SYS_REGLCTL register.</p>
[0]	<b>CHIPRST</b>	<p><b>Chip One-shot Reset (Write Protect)</b></p> <p>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.</p> <p>The CHIPRST is same as the POR reset, all the chip controllers is reset and the chip setting from flash are also reload.</p> <p>About the difference between CHIPRST and SYSRESETREQ(AIRCR[2]), please refer to section 6.2.2</p> <p>0 = Chip normal operation.</p> <p>1 = Chip one-shot reset.</p> <p><b>Note:</b> This bit is write protected. Refer to the SYS_REGLCTL register.</p>

**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			ADCRST	USBDNST	Reserved		
23	22	21	20	19	18	17	16
Reserved	ACMP01RST	PWM1RST	PWM0RST	Reserved	UART2RST	UART1RST	UART0RST
15	14	13	12	11	10	9	8
Reserved		SPI1RST	SPI0RST	Reserved		I2C1RST	I2C0RST
7	6	5	4	3	2	1	0
Reserved		TMR3RST	TMR2RST	TMR1RST	TMR0RST	GPIORST	Reserved

Bits	Description
[31:29]	<b>Reserved</b> Reserved.
[28]	<b>ADCRST</b> <b>ADC Controller Reset</b> 0 = ADC controller normal operation. 1 = ADC controller reset.
[27]	<b>USBDNST</b> <b>USB Device Controller Reset</b> 0 = USB device controller normal operation. 1 = USB device controller reset.
[26:23]	<b>Reserved</b> Reserved.
[22]	<b>ACMP01RST</b> <b>ACMP01 Controller Reset</b> 0 = ACMP01 controller normal operation. 1 = ACMP01 controller reset.
[21]	<b>PWM1RST</b> <b>PWM1 Controller Reset</b> 0 = PWM1 controller normal operation. 1 = PWM1 controller reset.
[20]	<b>PWM0RST</b> <b>PWM0 Controller Reset</b> 0 = PWM0 controller normal operation. 1 = PWM0 controller reset.
[19]	<b>Reserved</b> Reserved.
[18]	<b>UART2RST</b> <b>UART2 Controller Reset</b> 0 = UART2 controller normal operation. 1 = UART2 controller reset.
[17]	<b>UART1RST</b> <b>UART1 Controller Reset</b> 0 = UART1 controller normal operation.

		1 = UART1 controller reset.
[16]	<b>UART0RST</b>	<b>UART0 Controller Reset</b> 0 = UART0 controller normal operation. 1 = UART0 controller reset.
[15:14]	<b>Reserved</b>	Reserved.
[13]	<b>SPI1RST</b>	<b>SPI1 Controller Reset</b> 0 = SPI1 controller normal operation. 1 = SPI1 controller reset.
[12]	<b>SPI0RST</b>	<b>SPI0 Controller Reset</b> 0 = SPI0 controller normal operation. 1 = SPI0 controller reset.
[11:10]	<b>Reserved</b>	Reserved.
[9]	<b>I2C1RST</b>	<b>I2C1 Controller Reset</b> 0 = I2C1 controller normal operation. 1 = I2C1 controller reset.
[8]	<b>I2C0RST</b>	<b>I2C0 Controller Reset</b> 0 = I2C0 controller normal operation. 1 = I2C0 controller reset.
[7:6]	<b>Reserved</b>	Reserved.
[5]	<b>TMR3RST</b>	<b>Timer3 Controller Reset</b> 0 = Timer3 controller normal operation. 1 = Timer3 controller reset.
[4]	<b>TMR2RST</b>	<b>Timer2 Controller Reset</b> 0 = Timer2 controller normal operation. 1 = Timer2 controller reset.
[3]	<b>TMR1RST</b>	<b>Timer1 Controller Reset</b> 0 = Timer1 controller normal operation. 1 = Timer1 controller reset.
[2]	<b>TMR0RST</b>	<b>Timer0 Controller Reset</b> 0 = Timer0 controller normal operation. 1 = Timer0 controller reset.
[1]	<b>GPIORST</b>	<b>GPIO Controller Reset</b> 0 = GPIO controller normal operation. 1 = GPIO controller reset.
[0]	<b>Reserved</b>	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							
15	14	13	12	11	10	9	8
Reserved					USCI2RST	USCI1RST	USCI0RST
7	6	5	4	3	2	1	0
Reserved						SC1RST	SC0RST

Bits	Description	
[31:11]	Reserved	Reserved.
[10]	USCI2RST	<b>USCI2 Controller Reset</b> 0 = USCI2 controller normal operation. 1 = USCI2 controller reset.
[9]	USCI1RST	<b>USCI1 Controller Reset</b> 0 = USCI1 controller normal operation. 1 = USCI1 controller reset.
[8]	USCI0RST	<b>USCI0 Controller Reset</b> 0 = USCI0 controller normal operation. 1 = USCI0 controller reset.
[7:2]	Reserved	Reserved.
[1]	SC1RST	<b>SC1 Controller Reset</b> 0 = SC1 controller normal operation. 1 = SC1 controller reset.
[0]	SC0RST	<b>SC0 Controller Reset</b> 0 = SC0 controller normal operation. 1 = SC0 controller reset.

**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	0x0000_038X

31	30	29	28	27	26	25	24
Reserved				VDETDGSEL			VDETDOUT
23	22	21	20	19	18	17	16
VDETBGSEL		VDETDTSSEL		VDETIF	VDETIEN	VDETPINSEL	VDETEN
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	BODVL		BODEN

Bits	Description	
[31:28]	Reserved	Reserved.
[27:25]	VDETDGSEL	<p><b>Voltage Detector Output De-glitch Time Select (Write Protect)</b></p> <p>000 = VDET output is sampled by VDET clock.                      001 = 16 system clock (HCLK).                      010 = 32 system clock (HCLK).                      011 = 64 system clock (HCLK).                      100 = 128 system clock (HCLK).                      101 = 256 system clock (HCLK).                      110 = 512 system clock (HCLK).                      111 = 1024 system clock (HCLK).</p> <p><b>Note:</b> These bits are write protected. Refer to the SYS_REGLCTL register.</p>
[24]	VDETDOUT	<p><b>Voltage Detector Output Status</b></p> <p>0 = VDET output status is 0.                      It means the detected voltage is higher than Bandgap or VDETEN is 0.                      1 = VDET output status is 1.                      It means the detected voltage is lower than Bandgap. If the VDETEN is 0, VDET function disabled, this bit always responds 0.</p>
[23:20]	Reserved	Reserved.
[19]	VDETIF	<p><b>Voltage Detector Interrupt Flag</b></p> <p>0 = VDET does not detect any voltage draft at external pin down through or up through the voltage of Bandgap.                      1 = When VDET detects the external pin is dropped down through the voltage of Bandgap or the external pin is raised up through the voltage of Bandgap, this bit is set to 1 and the brown-out interrupt is requested if brown-out interrupt is enabled.</p> <p><b>Note:</b> This bit can be cleared by software writing '1'.</p>



Bits	Description	
[18]	VDETIEN	<b>Voltage Detector Interrupt Enable Bit</b> 0 = VDET interrupt Disabled. 1 = VDET interrupt Enabled.
[17]	VDETPINSEL	<b>Voltage Detector External Input Voltage Pin Selection</b> 0 = The input voltage is from VDET_P0 (PB.0). 1 = The input voltage is from VDET_P1 (PB.1). <b>Note1:</b> If VDET_P0 is selected, multi-function pin must be selected correctly in PB0MFP (SYS_GPB_MFPL[3:0]). <b>Note2:</b> If VDET_P1 is selected, multi-function pin must be selected correctly in PB1MFP (SYS_GPB_MFPL[7:4]).
[16]	VDETEN	<b>Voltage Detector Enable Bit</b> 0 = VDET detect external input voltage function Disabled. 1 = VDET detect external input voltage function Enabled. <b>Note1:</b> This function is still active in whole chip power-down mode. <b>Note2:</b> This function need use LIRC or LXT as VDET clock source, which is selected in VDETCKSEL (CLK_BODCLK[0]). <b>Note2:</b> The input pin for VDET detect voltage is selectabe by VDETPINSEL (SYS_BODCTL[17]).
[15]	Reserved	Reserved.
[14:12]	LVRDGSEL	<b>LVR Output De-glitch Time Select (Write Protect)</b> 000 = Without de-glitch function. 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). <b>Note:</b> These bits are write protected. Refer to the SYS_REGLCTL register.
[11]	Reserved	Reserved.
[10:8]	BODDGSEL	<b>Brown-out Detector Output De-glitch Time Select (Write Protect)</b> 000 = BOD output is sampled by RC10K clock. 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). <b>Note:</b> These bits are write protected. Refer to the SYS_REGLCTL register.

Bits	Description	
[7]	LVREN	<p><b>Low Voltage Reset Enable Bit (Write Protect)</b>                      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.  <b>Note1:</b> After enabling the bit, the LVR function will be active with 200us delay for LVR output stable (default).  <b>Note2:</b> This bit is write protected. Refer to the SYS_REGLCTL register.</p>
[6]	BODOUT	<p><b>Brown-out Detector Output Status</b>                      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 disabled, this bit always responds 0.</p>
[5]	BODLPM	<p><b>Brown-out Detector Low Power Mode (Write Protect)</b>                      0 = BOD operate in normal mode (default).                      1 = BOD Low Power mode Enabled.  <b>Note1:</b> 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.  <b>Note2:</b> This bit is write protected. Refer to the SYS_REGLCTL register.</p>
[4]	BODIF	<p><b>Brown-out Detector Interrupt Flag</b>                      0 = Brown-out Detector does not detect any voltage draft at V<sub>DD</sub> down through or up through the voltage of BODVL setting.                      1 = When Brown-out Detector detects the V<sub>DD</sub> is dropped down through the voltage of BODVL setting or the V<sub>DD</sub> 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.  <b>Note:</b> This bit can be cleared by software writing '1'.</p>
[3]	BODRSTEN	<p><b>Brown-out Reset Enable Bit (Write Protect)</b>                      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.  <b>Note1:</b>                      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).  <b>Note2:</b> This bit is write protected. Refer to the SYS_REGLCTL register.</p>
[2:1]	BODVL	<p><b>Brown-out Detector Threshold Voltage Selection (Write Protect)</b>                      The default value is set by flash controller user configuration register CBOV (CONFIG0 [22:21]).                      00 = Brown-Out Detector threshold voltage is 2.2V.                      01 = Brown-Out Detector threshold voltage is 2.7V.                      10 = Brown-Out Detector threshold voltage is 3.7V.                      11 = Brown-Out Detector threshold voltage is 4.5V.  <b>Note:</b> This bit is write protected. Refer to the SYS_REGLCTL register.</p>

Bits	Description	
[0]	<b>BODEN</b>	<p><b>Brown-out Detector Enable Bit (Write Protect)</b></p> <p>The default value is set by flash controller user configuration register CBODEN (CONFIG0 [23]).</p> <p>0 = Brown-out Detector function Disabled.</p> <p>1 = Brown-out Detector function Enabled.</p> <p><b>Note:</b> This bit is write protected. Refer to the SYS_REGLCTL register.</p>

**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						VBATUGEN	VTEMPEN

Bits	Description	
[31:2]	Reserved	Reserved.
[1]	VBATUGEN	<p><b>VBAT Unity Gain Buffer Enable Bit</b></p> <p>This bit is used to enable/disable VBAT unity gain buffer function.</p> <p>0 = VBAT unity gain buffer function Disabled (default).</p> <p>1 = VBAT unity gain buffer function Enabled.</p> <p><b>Note:</b> After this bit is set to 1, the value of VBAT unity gain buffer output voltage can be obtained from ADC conversion result. Please refer to ADC function chapter for details.</p>
[0]	VTEMPEN	<p><b>Temperature Sensor Enable Bit</b></p> <p>This bit is used to enable/disable temperature sensor function.</p> <p>0 = Temperature sensor function Disabled (default).</p> <p>1 = Temperature sensor function Enabled.</p> <p><b>Note:</b> After this bit is set to 1, the value of temperature sensor output can be obtained from ADC conversion result. Please refer to ADC function chapter for details.</p>

**Power-on Reset Controller Register (SYS\_PORCTL)**

Register	Offset	R/W	Description	Reset Value
SYS_PORCTL	SYS_BA+0x24	R/W	Power-on Reset Controller 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
POROFF[15:8]							
7	6	5	4	3	2	1	0
POROFF[7:0]							

Bits	Description	
[31:16]	Reserved	Reserved.
[15:0]	POROFF	<p><b>Power-on Reset Enable Bit (Write Protect)</b></p> <p>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 disable internal POR circuit to avoid unpredictable noise to cause chip reset by writing 0x5AA5 to this field.</p> <p>The POR function will be active again when this field is set to another value or chip is reset by other reset source, including:</p> <p>nRESET, Watchdog, LVR reset, BOD reset, ICE reset command and the software-chip reset function.</p> <p><b>Note:</b> This bit is write protected. Refer to the SYS_REGLCTL register.</p>

**V<sub>REF</sub> Control Register (SYS\_VREFCTL)**

Register	Offset	R/W	Description	Reset Value
SYS_VREFCTL	SYS_BA+0x28	R/W	V <sub>REF</sub> 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				VREFCTL			

Bits	Description	
[31:5]	Reserved	Reserved.
[4:0]	VREFCTL	<p><b>Int_V<sub>REF</sub> Control Bits (Write Protect)</b></p> <p>00000 = From V<sub>REF</sub> pin.</p> <p>00011 = V<sub>REF</sub> is internal 2.56V.</p> <p>00111 = V<sub>REF</sub> is internal 2.048V.</p> <p>01011 = V<sub>REF</sub> is internal 3.072V.</p> <p>01111 = V<sub>REF</sub> is internal 4.096V.</p> <p>10000 = V<sub>REF</sub> is from AV<sub>DD</sub>.</p> <p>Others = Reserved.</p> <p><b>Note:</b> These bit are write protected. Refer to the SYS_REGLCTL register.</p>

**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	PC15 Multi-function Pin Selection
[27:24]	PC14MFP	PC14 Multi-function Pin Selection
[23:20]	PC13MFP	PC13 Multi-function Pin Selection
[19:16]	PC12MFP	PC12 Multi-function Pin Selection
[15:12]	PC11MFP	PC11 Multi-function Pin Selection
[11:8]	PC10MFP	PC10 Multi-function Pin Selection
[7:4]	PC9MFP	PC9 Multi-function Pin Selection
[3:0]	PC8MFP	PC8 Multi-function Pin Selection

**GPIOD Low Byte Multiple Function Control Register (SYS\_GPD\_MFPL)**

Register	Offset	R/W	Description	Reset Value
SYS_GPD_MFPL	SYS_BA+0x48	R/W	GPIOD Low Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PD7MFP				PD6MFP			
23	22	21	20	19	18	17	16
PD5MFP				PD4MFP			
15	14	13	12	11	10	9	8
PD3MFP				PD2MFP			
7	6	5	4	3	2	1	0
PD1MFP				PD0MFP			

Bits	Description	
[31:28]	PD7MFP	PD.7 Multi-function Pin Selection
[27:24]	PD6MFP	PD.6 Multi-function Pin Selection
[23:20]	PD5MFP	PD.5 Multi-function Pin Selection
[19:16]	PD4MFP	PD.4 Multi-function Pin Selection
[15:12]	PD3MFP	PD.3 Multi-function Pin Selection
[11:8]	PD2MFP	PD.2 Multi-function Pin Selection
[7:4]	PD1MFP	PD.1 Multi-function Pin Selection
[3:0]	PD0MFP	PD.0 Multi-function Pin Selection

**GPIOD High Byte Multiple Function Control Register (SYS\_GPD\_MFPH)**

Register	Offset	R/W	Description	Reset Value
SYS_GPD_MFPH	SYS_BA+0x4C	R/W	GPIOD High Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PD15MFP				PD14MFP			
23	22	21	20	19	18	17	16
PD13MFP				PD12MFP			
15	14	13	12	11	10	9	8
PD11MFP				PD10MFP			
7	6	5	4	3	2	1	0
PD9MFP				PD8MFP			

Bits	Description	
[31:28]	PD15MFP	PD.15 Multi-function Pin Selection
[27:24]	PD14MFP	PD.14 Multi-function Pin Selection
[23:20]	PD13MFP	PD.13 Multi-function Pin Selection
[19:16]	PD12MFP	PD.12 Multi-function Pin Selection
[15:12]	PD11MFP	PD.11 Multi-function Pin Selection
[11:8]	PD10MFP	PD.10 Multi-function Pin Selection
[7:4]	PD9MFP	PD.9 Multi-function Pin Selection
[3:0]	PD8MFP	PD.8 Multi-function Pin Selection

**GPIOE Low Byte Multiple Function Control Register (SYS\_GPE\_MFPL)**

Register	Offset	R/W	Description	Reset Value
SYS_GPE_MFPL	SYS_BA+0x50	R/W	GPIOE Low Byte Multiple Function Control Register	0x1100_0000

31	30	29	28	27	26	25	24
PE7MFP				PE6MFP			
23	22	21	20	19	18	17	16
PE5MFP				PE4MFP			
15	14	13	12	11	10	9	8
PE3MFP				PE2MFP			
7	6	5	4	3	2	1	0
PE1MFP				PE0MFP			

Bits	Description	
[31:28]	PE7MFP	PE.7 Multi-function Pin Selection
[27:24]	PE6MFP	PE.6 Multi-function Pin Selection
[23:20]	PE5MFP	PE.5 Multi-function Pin Selection
[19:16]	PE4MFP	PE.4 Multi-function Pin Selection
[15:12]	PE3MFP	PE.3 Multi-function Pin Selection
[11:8]	PE2MFP	PE.2 Multi-function Pin Selection
[7:4]	PE1MFP	PE.1 Multi-function Pin Selection
[3:0]	PE0MFP	PE.0 Multi-function Pin Selection

**GPIOE High Byte Multiple Function Control Register (SYS\_GPE\_MFPH)**

Register	Offset	R/W	Description	Reset Value
SYS_GPE_MFPH	SYS_BA+0x54	R/W	GPIOE High Byte Multiple Function Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
PE13MFP				PE12MFP			
15	14	13	12	11	10	9	8
PE11MFP				PE10MFP			
7	6	5	4	3	2	1	0
PE9MFP				PE8MFP			

Bits	Description	
[31:24]	Reserved	Reserved.
[23:20]	PE13MFP	PE.13 Multi-function Pin Selection
[19:16]	PE12MFP	PE.12 Multi-function Pin Selection
[15:12]	PE11MFP	PE.11 Multi-function Pin Selection
[11:8]	PE10MFP	PE.10 Multi-function Pin Selection
[7:4]	PE9MFP	PE.9 Multi-function Pin Selection
[3:0]	PE8MFP	PE.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	0x000X_X000

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	<p><b>PF.4 Multi-function Pin Selection</b></p> <p>The default value is set by flash controller user configuration register CFGXT1(CONFIG0[27]) bit.</p> <p>0 = PF.4 pin is configured as GPIO pins.</p> <p>1 = PF.4 pin is configured as external 4~24 MHz external high speed crystal oscillator (HXT) pins.</p>
[15:12]	PF3MFP	<p><b>PF.3 Multi-function Pin Selection</b></p> <p>The default value is set by flash controller user configuration register CFGXT1(CONFIG0[27]) bit.</p> <p>0 = PF.3 pin is configured as GPIO pins.</p> <p>1 = PF.3 pin is configured as external 4~24 MHz external high speed crystal oscillator (HXT) pins.</p>
[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

**Modulation Control Register (SYS\_MODCTL)**

Register	Offset	R/W	Description	Reset Value
SYS_MODCTL	SYS_BA+0xC0	R/W	Modulation 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	MODPWMSEL			Reserved		MODH	MODEN

Bits	Description	Description
[31:7]	Reserved	Reserved.
[6:4]	MODPWMSEL	<p><b>PWM0 Channel Select for Modulation</b>                      Select the PWM0 channel to modulate with the UART1_TXD.                      000 = PWM0 channel 0 modulate with UART1_TXD.                      001 = PWM0 channel 1 modulate with UART1_TXD.                      010 = PWM0 channel 2 modulate with UART1_TXD.                      011 = PWM0 channel 3 modulate with UART1_TXD.                      Others = Reserved.  <b>Note:</b> This bit is valid while MODEN (SYS_MODCTL[0]) is set to 1.</p>
[3:2]	Reserved	Reserved.
[1]	MODH	<p><b>Modulation at Data High</b>                      Select modulation pulse(PWM) at UART1_TXD high or low                      0 = Modulation pulse at UART1_TXD low.                      1 = Modulation pulse at UART1_TXD high.</p>
[0]	MODEN	<p><b>Modulation Function Enable Bit</b>                      This bit enables modulation function by modulating with PWM channel output and UART1_TXD.                      0 = Modulation Function Disabled.                      1 = Modulation Function Enabled.</p>

**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
Reserved			USBBI <b>S</b> T	Reserved	CRBI <b>S</b> T	Reserved	SRBI <b>S</b> T

Bits	Description	
[31:5]	Reserved	Reserved.
[4]	USBBI <b>S</b> T	<p><b>USB BIST Enable Bit (Write Protect)</b>                      This bit enables BIST test for USB RAM                      0 = System USB BIST Disabled.                      1 = System USB BIST Enabled.  <b>Note:</b> This bit is write protected. Refer to the SYS_REGLCTL register.</p>
[3]	Reserved	Reserved.
[2]	CRBI <b>S</b> T	<p><b>CACHE BIST Enable Bit (Write Protect)</b>                      This bit enables BIST test for CACHE RAM                      0 = System CACHE BIST Disabled.                      1 = System CACHE BIST Enabled.  <b>Note:</b> This bit is write protected. Refer to the SYS_REGLCTL register.</p>
[1]	Reserved	Reserved.
[0]	SRBI <b>S</b> T	<p><b>SRAM BIST Enable Bit (Write Protect)</b>                      This bit enables BIST test for SRAM located in address 0x2000_0000 ~0x2000_4FFF                      0 = System SRAM BIST Disabled.                      1 = System SRAM BIST Enabled.  <b>Note:</b> This bit is write protected. Refer to the SYS_REGLCTL register.</p>

**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
Reserved			USBEND	Reserved	CRBEND	Reserved	SRBEND
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved			USBBEF	Reserved	CRBISTEF	Reserved	SRBISTEF

Bits	Description
[31:21]	Reserved
[20]	<b>USBEND</b> USB SRAM BIST Test Finish 0 = USB SRAM BIST is active. 1 = USB SRAM BIST test finish.
[19]	Reserved
[18]	<b>CRBEND</b> CACHE SRAM BIST Test Finish 0 = System CACHE RAM BIST is active. 1 = System CACHE RAM BIST test finish.
[17]	Reserved
[16]	<b>SRBEND</b> SRAM BIST Test Finish 0 = System SRAM BIST active. 1 = System SRAM BIST finish.
[15:5]	Reserved
[4]	<b>USBBEF</b> USB SRAM BIST Fail Flag 0 = USB SRAM BIST test pass. 1 = USB SRAM BIST test fail.
[3]	Reserved
[2]	<b>CRBISTEF</b> CACHE SRAM BIST Fail Flag 0 = System CACHE RAM BIST test pass. 1 = System CACHE RAM BIST test fail.
[1]	Reserved
[0]	<b>SRBISTEF</b> System SRAM BIST Fail Flag 0 = System SRAM BIST test pass. 1 = System SRAM BIST test fail.

**HIRC0 Trim Control Register (SYS\_IRCTCTL0)**

Register	Offset	R/W	Description	Reset Value
SYS_IRCTCTL0	SYS_BA+0x80	R/W	HIRC0 Trim 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					REFCKSEL	Reserved	CESTOPEN
7	6	5	4	3	2	1	0
RETRYCNT		LOOPSEL			Reserved		FREQSEL

Bits	Description	
[31:10]	Reserved	Reserved.
[10]	REFCKSEL	<b>Reference Clock Selection</b> 0 = HIRC trim reference clock is from external 32.768 kHz crystal oscillator . 1 = HIRC trim reference clock is from internal USB synchronous mode.
[9]	Reserved	Reserved.
[8]	CESTOPEN	<b>Clock Error Stop Enable Bit</b> 0 = The trim operation is keep going if clock is inaccuracy. 1 = The trim operation is stopped if clock is inaccuracy.
[7:6]	RETRYCNT	<b>Trim Value Update Limitation Count</b> This field defines that how many times the auto trim circuit will try to update the HIRC0 trim value before the frequency of HIRC0 locked. Once the HIRC0 locked, the internal trim value update counter will be reset. If the trim value update counter reached this limitation value and frequency of HIRC0 still doesn't lock, the auto trim operation will be disabled and FREQSEL(SYS_IRCTCTL0[1:0]) 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	<b>Trim Calculation Loop Selection</b> This field defines that trim value calculation is based on how many clocks of reference clock (32.768 kHz, LXT). 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. <b>Note:</b> 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	<p><b>Trim Frequency Selection</b></p> <p>This field indicates the target frequency of internal high speed RC oscillator 0 (HIRC0) auto trim.</p> <p>During auto trim operation, if clock error detected with CESTOPEN(SYS_IRCTCTL0[8]) is set to 1 or trim retry limitation count reached, this field will be cleared to 00 automatically.</p> <p>00 = Disable HIRC0 auto trim function.                      01 = Enable HIRC0 auto trim function and trim HIRC to 22.1184 MHz.                      10 = Reserved.                      11 = Reserved.</p>

**HIRC Trim Interrupt Enable Register (SYS\_IRCTIEN)**

Register	Offset	R/W	Description	Reset Value
SYS_IRCTIEN	SYS_BA+0x84	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					CLKEIEN1	TFALIEN1	Reserved
7	6	5	4	3	2	1	0
Reserved					CLKEIEN	TFALIEN	Reserved

Bits	Description	
[31:11]	Reserved	Reserved.
[10]	CLKEIEN1	<p><b>HIRC1 Clock Error Interrupt Enable Bit</b></p> <p>This bit controls if CPU would get an interrupt while HIRC1 clock is inaccuracy during auto trim operation.</p> <p>If this bit is set to 1, and CLKERRIF(SYS_IRCTSTS[2]) is set during auto trim operation, an interrupt will be triggered to notify the clock frequency is inaccuracy.</p> <p>0 = Disable CLKERRIF(SYS_IRCTSTS[2]) status to trigger an interrupt to CPU. 1 = Enable CLKERRIF(SYS_IRCTSTS[2]) status to trigger an interrupt to CPU.</p>
[9]	TFALIEN1	<p><b>HIRC1 Trim Failure Interrupt Enable Bit</b></p> <p>This bit controls if an interrupt will be triggered while HIRC1 trim value update limitation count reached and HIRC1 frequency still not locked on target frequency set by FREQSEL(SYS_IRCTCTL1[1:0]).</p> <p>If this bit is high and TFALIF(SYS_IRCTSTS[1]) is set during auto trim operation, an interrupt will be triggered to notify that HIRC1 trim value update limitation count was reached.</p> <p>0 = Disable TFALIF(SYS_IRCTSTS[1]) status to trigger an interrupt to CPU. 1 = Enable TFALIF(SYS_IRCTSTS[1]) status to trigger an interrupt to CPU.</p>
[8:3]	Reserved	Reserved.
[2]	CLKEIEN	<p><b>HIRC0 Clock Error Interrupt Enable Bit</b></p> <p>This bit controls if CPU would get an interrupt while HIRC0 clock is inaccuracy during auto trim operation.</p> <p>If this bit is set to 1, and CLKERRIF(SYS_IRCTSTS0[2]) is set during auto trim operation, an interrupt will be triggered to notify the clock frequency is inaccuracy.</p> <p>0 = Disable CLKERRIF(SYS_IRCTSTS0[2]) status to trigger an interrupt to CPU. 1 = Enable CLKERRIF(SYS_IRCTSTS0[2]) status to trigger an interrupt to CPU.</p>
[1]	TFALIEN	<p><b>HIRC0 Trim Failure Interrupt Enable Bit</b></p> <p>This bit controls if an interrupt will be triggered while HIRC0 trim value update limitation count reached and HIRC frequency still not locked on target frequency set by FREQSEL(SYS_IRCTCTL0[1:0]).</p> <p>If this bit is high and TFALIF(SYS_IRCTSTS0[1]) is set during auto trim operation, an</p>

		<p>interrupt will be triggered to notify that HIRC0 trim value update limitation count was reached.</p> <p>0 = Disable TFAILIF(SYS_IRCTSTS0[1]) status to trigger an interrupt to CPU.</p> <p>1 = Enable TFAILIF(SYS_IRCTSTS0[1]) status to trigger an interrupt to CPU.</p>
[0]	<b>Reserved</b>	Reserved.



**HIRC Trim Interrupt Status Register (SYS\_IRCTISTS)**

Register	Offset	R/W	Description	Reset Value
SYS_IRCTISTS	SYS_BA+0x88	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					CLKERRIF1	TFAILIF1	FREQLOCK1
7	6	5	4	3	2	1	0
Reserved					CLKERRIF	TFAILIF	FREQLOCK

Bits	Description	
[31:11]	Reserved	Reserved.
[10]	CLKERRIF1	<p><b>HIRC1 Clock Error Interrupt Status</b></p> <p>When the frequency of internal USB synchronous or 48 MHz internal high speed RC oscillator 1 (HIRC1) is shift larger to unreasonable value, this bit will be set and to be an indicate that clock frequency is inaccuracy</p> <p>Once this bit is set to 1, the auto trim operation stopped and FREQSEL(SYS_IRCTCL1[1:0]) will be cleared to 00 by hardware automatically if CESTOPEN(SYS_IRCTCTL1[8]) is set to 1.</p> <p>If this bit is set and CLKEIEN(SYS_IRCTIEN1[2]) is high, an interrupt will be triggered to notify the clock frequency is inaccuracy. Write 1 to clear this to 0.</p> <p>0 = HIRC1 Clock frequency is accuracy. 1 = HIRC1 Clock frequency is inaccuracy.</p>
[9]	TFAILIF1	<p><b>HIRC1 Trim Failure Interrupt Status</b></p> <p>This bit indicates that HIRC1 trim value update limitation count reached and the HIRC1 clock frequency still doesn't be locked. Once this bit is set, the auto trim operation stopped and FREQSEL(SYS_IRCTCTL1[1:0]) will be cleared to 00 by hardware automatically.</p> <p>If this bit is set and TFALIEN(SYS_IRCTIEN1[1]) is high, an interrupt will be triggered to notify that HIRC1 trim value update limitation count was reached. Write 1 to clear this to 0.</p> <p>0 = HIRC1 trim value update limitation count does not reach. 1 = HIRC1 trim value update limitation count reached and frequency still not locked.</p>
[8]	FREQLOCK1	<p><b>HIRC1 Frequency Lock Status</b></p> <p>This bit indicates the HIRC1 frequency is locked.</p> <p>This is a status bit and doesn't trigger any interrupt.</p> <p>0 = The internal high-speed RC oscillator 1 frequency doesn't lock at 48 MHz yet. 1 = The internal high-speed RC oscillator 1 frequency locked at 48 MHz.</p>
[7:3]	Reserved	Reserved.
[2]	CLKERRIF	<p><b>Clock Error Interrupt Status</b></p> <p>When the frequency of 32.768 kHz external low speed crystal oscillator (LXT) or 22.1184 MHz internal high speed RC oscillator 0 (HIRC0) is shift larger to unreasonable value, this bit will be set and to be an indicate that clock frequency is inaccuracy</p>

		<p>Once this bit is set to 1, the auto trim operation stopped and FREQSEL(SYS_IRCTCL0[1:0]) will be cleared to 00 by hardware automatically if CESTOPEN(SYS_IRCTCTL0[8]) is set to 1.</p> <p>If this bit is set and CLKEIEN(SYS_IRCTIEN0[2]) is high, an interrupt will be triggered to notify the clock frequency is inaccuracy. Write 1 to clear this to 0.</p> <p>0 = Clock frequency is accuracy. 1 = Clock frequency is inaccuracy.</p>
[1]	<b>TFAILIF</b>	<p><b>Trim Failure Interrupt Status</b></p> <p>This bit indicates that HIRC0 trim value update limitation count reached and the HIRC0 clock frequency still doesn't be locked. Once this bit is set, the auto trim operation stopped and FREQSEL(SYS_IRCTCTL0[1:0]) will be cleared to 00 by hardware automatically.</p> <p>If this bit is set and TFALIEN(SYS_IRCTIEN0[1]) is high, an interrupt will be triggered to notify that HIRC0 trim value update limitation count was reached. Write 1 to clear this to 0.</p> <p>0 = Trim value update limitation count does not reach. 1 = Trim value update limitation count reached and HIRC frequency still not locked.</p>
[0]	<b>FREQLOCK</b>	<p><b>HIRC Frequency Lock Status</b></p> <p>This bit indicates the HIRC0 frequency is locked.</p> <p>This is a status bit and doesn't trigger any interrupt.</p> <p>0 = The internal high-speed RC oscillator 0 frequency doesn't lock at 22.1184 MHz yet. 1 = The internal high-speed RC oscillator 0 frequency locked at 22.1184 MHz.</p>

**HIRC1 Trim Control Register (SYS\_IRCTCTL1)**

Register	Offset	R/W	Description	Reset Value
SYS_IRCTCTL1	SYS_BA+0x90	R/W	HIRC1 Trim Control Register	0x0000_0400

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					REFCKSEL	Reserved	CESTOPEN
7	6	5	4	3	2	1	0
RETRYCNT		LOOPSEL			Reserved		FREQSEL

Bits	Description	
[31:11]	Reserved	Reserved.
[10]	REFCKSEL	<b>Reference Clock Selection</b> 0 = HIRC trim reference clock is from external 32.768 kHz crystal oscillator. 1 = HIRC trim reference clock is from internal USB synchronous mode.
[9]	Reserved	Reserved.
[8]	CESTOPEN	<b>Clock Error Stop Enable Bit</b> 0 = The trim operation is keep going if clock is inaccuracy. 1 = The trim operation is stopped if clock is inaccuracy.
[7:6]	RETRYCNT	<b>Trim Value Update Limitation Count</b> This field defines that how many times the auto trim circuit will try to update the HIRC1 trim value before the frequency of HIRC1 locked. Once the HIRC1 locked, the internal trim value update counter will be reset. If the trim value update counter reached this limitation value and frequency of HIRC1 still doesn't lock, the auto trim operation will be disabled and FREQSEL(SYS_IRCTCTL1[1:0]) 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	<b>Trim Calculation Loop Selection</b> This field defines that trim value calculation is based on how many clocks of reference clock. 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. <b>Note:</b> 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	<p><b>Trim Frequency Selection</b></p> <p>This field indicates the target frequency of internal high speed RC oscillator 1 (HIRC 1) auto trim.</p> <p>During auto trim operation, if clock error detected with CESTOPEN(SYS_IRCTL1[8]) is set to 1 or trim retry limitation count reached, this field will be cleared to 00 automatically.</p> <p>00 = Disable HIRC1 auto trim function.            01 = Reserved.            10 = Enable HIRC1 auto trim function and trim HIRC to 48 MHz.            11 = Reserved.</p>

**Register Lock Control Register (SYS\_REGLCTL)**

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							REGLCTL[0]

Bits	Description	
[31:8]	Reserved	Reserved.
[7:1]	REGLCTL	<p><b>Register Lock Control Code (Write Only)</b></p> <p>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.</p>
[0]	REGLCTL[0]	<p><b>Register Lock Control Disable Index (Read Only)</b></p> <p>0 = Write-protection Enabled for writing protected registers. Any write to the protected register is ignored.                      1 = Write-protection Disabled for writing protected registers.</p> <p>The Protected registers are:</p> <p><b>SYS_IPRST0:</b> address 0x5000_0008  <b>SYS_BODCTL:</b> address 0x5000_0018  <b>SYS_PORCTL:</b> address 0x5000_0024  <b>SYS_VREFCTL:</b> address 0x5000_0028  <b>SYS_SRAM_BISTCTL:</b> address 0x5000_00D0  <b>CLK_PWRCTL[13]:</b> address 0x5000_0200 (HIRC48 Enable Bit)  <b>CLK_PWRCTL[12]:</b> address 0x5000_0200 (HXT Crystal Type Select Bit)  <b>CLK_PWRCTL[11:10]:</b> address 0x5000_0200 (HXT Gain Control Bit)  <b>CLK_PWRCTL[7]:</b> address 0x5000_0200 (System Power-down Enable)  <b>CLK_PWRCTL[5]:</b> address 0x5000_0200 (Power-down Mode Wake-up Interrupt Enable Bit)  <b>CLK_PWRCTL[4]:</b> address 0x5000_0200 (Enable the Wake-up Delay Counter)  <b>CLK_PWRCTL[3]:</b> address 0x5000_0200 (LIRC Enable Bit)  <b>CLK_PWRCTL[2]:</b> address 0x5000_0200 (HIRC Enable Bit)  <b>CLK_PWRCTL[1]:</b> address 0x5000_0200 (LXT Enable Bit)  <b>CLK_PWRCTL[0]:</b> address 0x5000_0200 (HXT Enable Bit)  <b>CLK_APBCLK0 [0]:</b> address 0x5000_0208 (bit[0] is watchdog clock enable)  <b>CLK_CLKSEL0:</b> address 0x5000_0210 (for HCLK and CPU STCLK clock source select)</p>

		<p>CLK_CLKSEL1 [1:0]: address 0x5000_0214 (for watchdog clock source select)</p> <p>CLK_CLKSEL3[8]: address 0x5000_0234 (<b>USB Clock Source Selection</b>)</p> <p>CLK_CLKDSTS: address 0x5000_0274</p> <p>FMC_ISPCTL: address 0x5000_C000 (Flash ISP Control register)</p> <p>FMC_ISPTRG: address 0x5000_C010 (ISP Trigger Control register)</p> <p>FMC_ISPSTS: address 0x5000_C040</p> <p>WDT_CTL: address 0x4000_4000</p> <p>FMC_FTCTL: address 0x5000_C018</p> <p>PWM0_CTL: address 0x4004_0000</p> <p>PWM1_CTL: address 0x4014_0000</p> <p>PWM0_DTCTL0_1: address 0x4004_0070</p> <p>PWM1_DTCTL0_1: address 0x4014_0070</p> <p>PWM0_DTCTL2_3: address 0x4004_0074</p> <p>PWM1_DTCTL2_3: address 0x4014_0074</p> <p>PWM0_DTCTL4_5: address 0x4004_0078</p> <p>PWM1_DTCTL4_5: address 0x4014_0078</p> <p>PWM0_BRKCTL0_1: address 0x4004_00C8</p> <p>PWM1_BRKCTL0_1: address 0x4014_00C8</p> <p>PWM0_BRKCTL2_3: address 0x4004_00CC</p> <p>PWM1_BRKCTL2_3: address 0x4014_00CC</p> <p>PWM0_BRKCTL4_5: address 0x4004_00D0</p> <p>PWM1_BRKCTL4_5: address 0x4014_00D0</p> <p>PWM0_INTEN1: address 0x4004_00E4</p> <p>PWM1_INTEN1: address 0x4014_00E4</p> <p>PWM0_INTSTS1: address 0x4004_00EC</p> <p>PWM1_INTSTS1: address 0x4014_00EC</p> <p>TIMER0_PWMCTL: address 0x4001_0040</p> <p>TIMER1_PWMCTL: address 0x4001_0140</p> <p>TIMER2_PWMCTL: address 0x4011_0040</p> <p>TIMER3_PWMCTL: address 0x4011_0140</p> <p>TIMER0_PWMDTCTL: address 0x4001_0058</p> <p>TIMER1_PWMDTCTL: address 0x4001_0158</p> <p>TIMER2_PWMDTCTL: address 0x4011_0058</p> <p>TIMER3_PWMDTCTL: address 0x4011_0158</p> <p>TIMER0_PWMBRKCTL: address 0x4001_0070</p> <p>TIMER1_PWMBRKCTL: address 0x4001_0170</p> <p>TIMER2_PWMBRKCTL: address 0x4011_0070</p> <p>TIMER3_PWMBRKCTL: address 0x4011_0170</p> <p>TIMER0_PWMSWBRK: address 0x4001_007C</p> <p>TIMER1_PWMSWBRK: address 0x4001_017C</p> <p>TIMER2_PWMSWBRK: address 0x4011_007C</p> <p>TIMER3_PWMSWBRK: address 0x4011_017C</p> <p>TIMER0_PWMINTEN1: address 0x4001_0084</p> <p>TIMER1_PWMINTEN1: address 0x4001_0184</p> <p>TIMER2_PWMINTEN1: address 0x4011_0084</p> <p>TIMER3_PWMINTEN1: address 0x4011_0184</p> <p>TIMER0_PWMINTSTS1: address 0x4001_008C</p> <p>TIMER1_PWMINTSTS1: address 0x4001_018C</p> <p>TIMER2_PWMINTSTS1: address 0x4011_008C</p> <p>TIMER3_PWMINTSTS1: address 0x4011_018C</p>
--	--	--

		TIMER3_PWMINTSTS1: address 0x4011_018C
--	--	--

**Temperature Sensor Offset Register (SYS\_TSOFFSET)**

Register	Offset	R/W	Description	Reset Value
SYS_TSOFFSET	SYS_BA+0x114	R	Temperature Sensor Offset 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				VTEMP[11:0]			
7	6	5	4	3	2	1	0
VTEMP[7:0]							

Bits	Description	
[31:12]	Reserved	Reserved.
[11:0]	VTEMP	<b>Temperature Sensor Offset Value</b> This field reflects temperature sensor output voltage offset at 25°C from flash.



### 6.2.13 System Timer (SysTick)

The Cortex<sup>®</sup>-M0 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\_CVR) to 0, and reload (wrap) to the value in the SysTick Reload Value Register (SYST\_RVR) on the next clock cycle, then decrement on subsequent clocks. When the counter transitions to 0, the COUNTFLAG status bit is set. The COUNTFLAG bit clears on reads.

The SYST\_CVR value is UNKNOWN on reset. Software should write to the register to clear it to 0 before enabling the feature. This ensures the timer will count from the SYST\_RVR value rather than an arbitrary value when it is enabled.

If the SYST\_RVR is 0, the timer will be maintained with a current value of 0 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<sup>®</sup> Cortex<sup>®</sup>-M0 Technical Reference Manual” and “ARM<sup>®</sup> v6-M Architecture Reference Manual”.

6.2.13.1 System Timer Control Register Map

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

Register	Offset	R/W	Description	Reset Value
<b>SYST Base Address:</b>				
<b>SCS_BA = 0xE000_E000</b>				
<b>SYST_CSR</b>	SCS_BA+0x10	R/W	SysTick Control and Status Register	0x0000_0000
<b>SYST_RVR</b>	SCS_BA+0x14	R/W	SysTick Reload Value Register	0xFFFF_XXXX
<b>SYST_CVR</b>	SCS_BA+0x18	R/W	SysTick Current Value Register	0xFFFF_XXXX

6.2.13.2 System Timer Control Register Description

**SysTick Control and Status Register (SYST\_CSR)**

Register	Offset	R/W	Description	Reset Value
SYST_CSR	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	<p><b>System Tick Counter Flag</b></p> <p>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.</p>
[15:3]	Reserved	Reserved.
[2]	CLKSRC	<p><b>System Tick Clock Source Selection</b></p> <p>0 = Clock source is the (optional) external reference clock. 1 = Core clock used for SysTick.</p>
[1]	TICKINT	<p><b>System Tick Interrupt Enabled</b></p> <p>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.</p>
[0]	ENABLE	<p><b>System Tick Counter Enabled</b></p> <p>0 = Counter Disabled. 1 = Counter will operate in a multi-shot manner.</p>

**SysTick Reload Value Register (SYST\_RVR)**

Register	Offset	R/W	Description	Reset Value
SYST_RVR	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[23:16]							
15	14	13	12	11	10	9	8
RELOAD[15:8]							
7	6	5	4	3	2	1	0
RELOAD[7:0]							

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

**SysTick Current Value Register (SYST\_CVR)**

Register	Offset	R/W	Description	Reset Value
SYST_CVR	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 [23:16]							
15	14	13	12	11	10	9	8
CURRENT [15:8]							
7	6	5	4	3	2	1	0
CURRENT[7:0]							

Bits	Description	
[31:24]	Reserved	Reserved.
[23:0]	CURRENT	<p><b>System Tick Current Value</b></p> <p>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.</p>

### 6.2.14 Nested Vectored Interrupt Controller (NVIC)

The Cortex<sup>®</sup>-M0 provides an interrupt controller as an integral part of the exception mode, named as “Nested Vectored Interrupt Controller (NVIC)”, which is closely coupled to the processor kernel and provides following features:

- Nested and Vectored interrupt support
- Automatic processor state saving and restoration
- Reduced and deterministic interrupt latency

The NVIC prioritizes and handles all supported exceptions. All exceptions are handled in “Handler Mode”. This NVIC architecture supports 32 (IRQ[31:0]) discrete interrupts with 4 levels of priority. All of the interrupts and most of the system exceptions can be configured to different priority levels. When an interrupt occurs, the NVIC will compare the priority of the new interrupt to the current running one’s priority. If the priority of the new interrupt is higher than the current one, the new interrupt handler will override the current handler.

When an interrupt is accepted, the starting address of the interrupt service routine (ISR) is fetched from a vector table in memory. There is no need to determine which interrupt is accepted and branch to the starting address of the correlated ISR by software. While the starting address is fetched, NVIC will also automatically save processor state including the registers “PC, PSR, LR, R0~R3, R12” to the stack. At the end of the ISR, the NVIC will restore the mentioned registers from stack and resume the normal execution. Thus it will take less and deterministic time to process the interrupt request.

The NVIC supports “Tail Chaining” which handles back-to-back interrupts efficiently without the overhead of states saving and restoration and therefore reduces delay time in switching to pending ISR at the end of current ISR. The NVIC also supports “Late Arrival” which improves the efficiency of concurrent ISRs. When a higher priority interrupt request occurs before the current ISR starts to execute (at the stage of state saving and starting address fetching), the NVIC will give priority to the higher one without delay penalty. Thus it advances the real-time capability.

For more detailed information, please refer to the “ARM<sup>®</sup> Cortex<sup>®</sup>-M0 Technical Reference Manual” and “ARM<sup>®</sup> v6-M Architecture Reference Manual”.

#### 6.2.14.1 Exception Model and System Interrupt Map

The following table lists the exception model supported by the NUC126 series. Software can set four levels of priority on some of these exceptions as well as on all interrupts. The highest user-configurable priority is denoted as “0” and the lowest priority is denoted as “3”. The default priority of all the user-configurable interrupts is “0”. Note that priority “0” is treated as the fourth priority on the system, after three system exceptions “Reset”, “NMI” and “Hard Fault”.

Exception Type	Vector Number	Vector Address	Priority
Reset	1	0x00000004	-3
NMI	2	0x00000008	-2
Hard Fault	3	0x0000000C	-1
Reserved	4 ~ 10		Reserved
SVCcall	11	0x0000002C	Configurable
Reserved	12 ~ 13		Reserved
PendSV	14	0x00000038	Configurable
SysTick	15	0x0000003C	Configurable
Interrupt (IRQ0 ~ IRQ)	16 ~ 47	0x00000000 +	Configurable

		(Vector Number)*4	
--	--	-------------------	--

Table 6.2-6 Exception Model

Vector Number	Interrupt Number (Bit In Interrupt Registers)	Interrupt Name	Interrupt Description
0 ~ 15	-	-	System exceptions
16	0	BOD_INT	Brown-out low voltage detected interrupt
17	1	WDT_INT	Window Watchdog Timer interrupt
18	2	EINT024	External interrupt from PA.0/PC.0/PD.2/PE.0/PE.4 pin
19	3	EINT135	External interrupt from PB.0/PC.0/ PD.0/PD.3/PE.5/PF.0 pin
20	4	GPAB_INT	External signal interrupt from PA[15:0]/PB[13:0]
21	5	GPCDEF_INT	External interrupt from PC[15:0]/PD[15:0]/PE[13:0]/PF[7:0]
22	6	PWM0_INT	PWM0 interrupt
23	7	PWM1_INT	PWM1 interrupt
24	8	TMR0_INT	Timer 0 interrupt
25	9	TMR1_INT	Timer 1 interrupt
26	10	TMR2_INT	Timer 2 interrupt
27	11	TMR3_INT	Timer 3 interrupt
28	12	UART02_INT	UART0 and UART2 interrupt
29	13	UART1_INT	UART1 interrupt
30	14	SPI0_INT	SPI0 interrupt
31	15	SPI1_INT	SPI1 interrupt
32	16		Reserved
33	17		Reserved
34	18	I2C0_INT	I <sup>2</sup> C0 interrupt
35	19	I2C1_INT	I <sup>2</sup> C1 interrupt
36	20		Reserved
37	21		Reserved
38	22	USCI_INT	USCI0, USCI1 and USCI2 interrupt
39	23	USB_D_INT	USB Device interrupt
40	24	SC_INT	SC0 and SC1 interrupt
41	25	ACMP01_INT	Analog Comparator interrupt
42	26	PDMA_INT	PDMA interrupt
43	27		Reserved
44	28	PWRWU_INT	Clock controller interrupt for chip wake-up from Power-down state

45	29	ADC_INT	ADC interrupt
46	30	CLKDIRC_INT	Clock fail detect and IRC TRIM interrupt
47	31	RTC_INT	Real Time Clock interrupt

Table 6.2-7 Interrupt Number Table

6.2.14.2 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 next section.



6.2.14.3 NVIC Control Registers

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

Register	Offset	R/W	Description	Reset Value
<b>NVIC Base Address:</b>				
<b>SCS_BA = 0xE000_E000</b>				
<b>NVIC_ISER</b>	SCS_BA+0x100	R/W	IRQ0 ~ IRQ31 Set-Enable Control Register	0x0000_0000
<b>NVIC_ICER</b>	SCS_BA+0x180	R/W	IRQ0 ~ IRQ31 Clear-Enable Control Register	0x0000_0000
<b>NVIC_ISPR</b>	SCS_BA+0x200	R/W	IRQ0 ~ IRQ31 Set-Pending Control Register	0x0000_0000
<b>NVIC_ICPR</b>	SCS_BA+0x280	R/W	IRQ0 ~ IRQ31 Clear-Pending Control Register	0x0000_0000
<b>NVIC_IPR0</b>	SCS_BA+0x400	R/W	IRQ0 ~ IRQ3 Priority Control Register	0x0000_0000
<b>NVIC_IPR1</b>	SCS_BA+0x404	R/W	IRQ4 ~ IRQ7 Priority Control Register	0x0000_0000
<b>NVIC_IPR2</b>	SCS_BA+0x408	R/W	IRQ8 ~ IRQ11 Priority Control Register	0x0000_0000
<b>NVIC_IPR3</b>	SCS_BA+0x40C	R/W	IRQ12 ~ IRQ15 Priority Control Register	0x0000_0000
<b>NVIC_IPR4</b>	SCS_BA+0x410	R/W	IRQ16 ~ IRQ19 Priority Control Register	0x0000_0000
<b>NVIC_IPR5</b>	SCS_BA+0x414	R/W	IRQ20 ~ IRQ23 Priority Control Register	0x0000_0000
<b>NVIC_IPR6</b>	SCS_BA+0x418	R/W	IRQ24 ~ IRQ27 Priority Control Register	0x0000_0000
<b>NVIC_IPR7</b>	SCS_BA+0x41C	R/W	IRQ28 ~ IRQ31 Priority Control Register	0x0000_0000

**IRQ0 ~ IRQ31 Set-Enable Control Register (NVIC\_ISER)**

Register	Offset	R/W	Description	Reset Value
NVIC_ISER	SCS_BA+0x100	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]	<p><b>SETENA</b></p> <p><b>Interrupt Enable Register</b>                      Enable one or more interrupts. Each bit represents an interrupt number from IRQ0 ~ IRQ31 (Vector number from 16 ~ 47).                      Write Operation:                      0 = No effect.                      1 = Write 1 to enable associated interrupt.                      Read Operation:                      0 = Associated interrupt status Disabled.                      1 = Associated interrupt status Enabled.  <b>Note:</b> Read value indicates the current enable status.</p>

**IRQ0 ~ IRQ31 Clear-Enable Control Register (NVIC\_ICER)**

Register	Offset	R/W	Description	Reset Value
NVIC_ICER	SCS_BA+0x180	R/W	IRQ0 ~ IRQ31 Clear-Enable Control Register	0x0000_0000

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

Bits	Description
[31:0]	<p><b>CLRENA</b></p> <p><b>Interrupt Disable Bits</b>                      Disable one or more interrupts. Each bit represents an interrupt number from IRQ0 ~ IRQ31 (Vector number from 16 ~ 47).                      Write Operation:                      0 = No effect.                      1 = Write 1 to disable associated interrupt.                      Read Operation:                      0 = Associated interrupt status Disabled.                      1 = Associated interrupt status Enabled.  <b>Note:</b> Read value indicates the current enable status.</p>

**IRQ0 ~ IRQ31 Set-Pending Control Register (NVIC\_ISPR)**

Register	Offset	R/W	Description	Reset Value
NVIC_ISPR	SCS_BA+0x200	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
SETPEN							
15	14	13	12	11	10	9	8
SETPEND							
7	6	5	4	3	2	1	0
SETPEND							

Bits	Description
[31:0]	<p><b>SETPEND</b></p> <p><b>Set Interrupt Pending Bits</b>                      Write Operation:                      0 = No effect.                      1 = Write 1 to set pending state. Each bit represents an interrupt number from IRQ0 ~ IRQ31 (Vector number from 16 ~ 47).                      Read Operation:                      0 = Associated interrupt in not in pending status.                      1 = Associated interrupt is in pending status.  <b>Note:</b> Read value indicates the current pending status.</p>

**IRQ0 ~ IRQ31 Clear-Pending Control Register (NVIC\_ICPR)**

Register	Offset	R/W	Description	Reset Value
NVIC_ICPR	SCS_BA+0x280	R/W	IRQ0 ~ IRQ31 Clear-Pending Control Register	0x0000_0000

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

Bits	Description
[31:0]	<p><b>CLRPEND</b></p> <p><b>Clear Interrupt Pending Bits</b>                      Write Operation:                      0 = No effect.                      1 = Write 1 to clear pending state. Each bit represents an interrupt number from IRQ0 ~ IRQ31 (Vector number from 16 ~ 47).                      Read Operation:                      0 = Associated interrupt in not in pending status.                      1 = Associated interrupt is in pending status.  <b>Note:</b> Read value indicates the current pending status.</p>

**IRQ0 ~ IRQ3 Priority Register (NVIC\_IPR0)**

Register	Offset	R/W	Description	Reset Value
NVIC_IPR0	SCS_BA+0x400	R/W	IRQ0 ~ IRQ3 Priority Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PRI_3		Reserved					
23	22	21	20	19	18	17	16
PRI_2		Reserved					
15	14	13	12	11	10	9	8
PRI_1		Reserved					
7	6	5	4	3	2	1	0
PRI_0		Reserved					

Bits	Description	
[31:30]	PRI_3	<b>Priority of IRQ3</b> "0" denotes the highest priority and "3" denotes the lowest priority.
[29:24]	Reserved	Reserved.
[23:22]	PRI_2	<b>Priority of IRQ2</b> "0" denotes the highest priority and "3" denotes the lowest priority.
[21:16]	Reserved	Reserved.
[15:14]	PRI_1	<b>Priority of IRQ1</b> "0" denotes the highest priority and "3" denotes the lowest priority.
[13:8]	Reserved	Reserved.
[7:6]	PRI_0	<b>Priority of IRQ0</b> "0" denotes the highest priority and "3" denotes the lowest priority.
[5:0]	Reserved	Reserved.

**IRQ4 ~ IRQ7 Priority Register (NVIC\_IPR1)**

Register	Offset	R/W	Description	Reset Value
NVIC_IPR1	SCS_BA+0x404	R/W	IRQ4 ~ IRQ7 Priority Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PRI_7		Reserved					
23	22	21	20	19	18	17	16
PRI_6		Reserved					
15	14	13	12	11	10	9	8
PRI_5		Reserved					
7	6	5	4	3	2	1	0
PRI_4		Reserved					

Bits	Description	
[31:30]	PRI_7	<b>Priority of IRQ7</b> "0" denotes the highest priority and "3" denotes the lowest priority.
[29:24]	Reserved	Reserved.
[23:22]	PRI_6	<b>Priority of IRQ6</b> "0" denotes the highest priority and "3" denotes the lowest priority.
[21:16]	Reserved	Reserved.
[15:14]	PRI_5	<b>Priority of IRQ5</b> "0" denotes the highest priority and "3" denotes the lowest priority.
[13:8]	Reserved	Reserved.
[7:6]	PRI_4	<b>Priority of IRQ4</b> "0" denotes the highest priority and "3" denotes the lowest priority.
[5:0]	Reserved	Reserved.

**IRQ8 ~ IRQ11 Priority Register (NVIC\_IPR2)**

Register	Offset	R/W	Description	Reset Value
NVIC_IPR2	SCS_BA+0x408	R/W	IRQ8 ~ IRQ11 Priority Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PRI_11		Reserved					
23	22	21	20	19	18	17	16
PRI_10		Reserved					
15	14	13	12	11	10	9	8
PRI_9		Reserved					
7	6	5	4	3	2	1	0
PRI_8		Reserved					

Bits	Description	
[31:30]	PRI_11	<b>Priority of IRQ11</b> "0" denotes the highest priority and "3" denotes the lowest priority.
[29:24]	Reserved	Reserved.
[23:22]	PRI_10	<b>Priority of IRQ10</b> "0" denotes the highest priority and "3" denotes the lowest priority.
[21:16]	Reserved	Reserved.
[15:14]	PRI_9	<b>Priority of IRQ9</b> "0" denotes the highest priority and "3" denotes the lowest priority.
[13:8]	Reserved	Reserved.
[7:6]	PRI_8	<b>Priority of IRQ8</b> "0" denotes the highest priority and "3" denotes the lowest priority.
[5:0]	Reserved	Reserved.



**IRQ12 ~ IRQ15 Priority Register (NVIC\_IPR3)**

Register	Offset	R/W	Description	Reset Value
NVIC_IPR3	SCS_BA+0x40C	R/W	IRQ12 ~ IRQ15 Priority Control Register	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
PRI_13		Reserved					
7	6	5	4	3	2	1	0
PRI_12		Reserved					

Bits	Description
[31:30]	<b>PRI_15</b> <b>Priority of IRQ15</b> "0" denotes the highest priority and "3" denotes the lowest priority.
[29:24]	<b>Reserved</b> Reserved.
[23:22]	<b>PRI_14</b> <b>Priority of IRQ14</b> "0" denotes the highest priority and "3" denotes the lowest priority.
[21:16]	<b>Reserved</b> Reserved.
[15:14]	<b>PRI_13</b> <b>Priority of IRQ13</b> "0" denotes the highest priority and "3" denotes the lowest priority.
[13:8]	<b>Reserved</b> Reserved.
[7:6]	<b>PRI_12</b> <b>Priority of IRQ12</b> "0" denotes the highest priority and "3" denotes the lowest priority.
[5:0]	<b>Reserved</b> Reserved.

**IRQ16 ~ IRQ19 Priority Register (NVIC\_IPR4)**

Register	Offset	R/W	Description	Reset Value
NVIC_IPR4	SCS_BA+0x410	R/W	IRQ16 ~ IRQ19 Priority Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PRI_19		Reserved					
23	22	21	20	19	18	17	16
PRI_18		Reserved					
15	14	13	12	11	10	9	8
PRI_17		Reserved					
7	6	5	4	3	2	1	0
PRI_16		Reserved					

Bits	Description
[31:30]	<b>PRI_19</b> <b>Priority of IRQ19</b> "0" denotes the highest priority and "3" denotes the lowest priority.
[29:24]	<b>Reserved</b> Reserved.
[23:22]	<b>PRI_18</b> <b>Priority of IRQ18</b> "0" denotes the highest priority and "3" denotes the lowest priority.
[21:16]	<b>Reserved</b> Reserved.
[15:14]	<b>PRI_17</b> <b>Priority of IRQ17</b> "0" denotes the highest priority and "3" denotes the lowest priority.
[13:8]	<b>Reserved</b> Reserved.
[7:6]	<b>PRI_16</b> <b>Priority of IRQ16</b> "0" denotes the highest priority and "3" denotes the lowest priority.
[5:0]	<b>Reserved</b> Reserved.

**IRQ20 ~ IRQ23 Priority Register (NVIC\_IPR5)**

Register	Offset	R/W	Description	Reset Value
NVIC_IPR5	SCS_BA+0x414	R/W	IRQ20 ~ IRQ23 Priority Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PRI_23		Reserved					
23	22	21	20	19	18	17	16
PRI_22		Reserved					
15	14	13	12	11	10	9	8
PRI_21		Reserved					
7	6	5	4	3	2	1	0
PRI_20		Reserved					

Bits	Description
[31:30]	<b>PRI_23</b> <b>Priority of IRQ23</b> "0" denotes the highest priority and "3" denotes the lowest priority.
[29:24]	<b>Reserved</b> Reserved.
[23:22]	<b>PRI_22</b> <b>Priority of IRQ22</b> "0" denotes the highest priority and "3" denotes the lowest priority.
[21:16]	<b>Reserved</b> Reserved.
[15:14]	<b>PRI_21</b> <b>Priority of IRQ21</b> "0" denotes the highest priority and "3" denotes the lowest priority
[13:8]	<b>Reserved</b> Reserved.
[7:6]	<b>PRI_20</b> <b>Priority of IRQ20</b> "0" denotes the highest priority and "3" denotes the lowest priority.
[5:0]	<b>Reserved</b> Reserved.

**IRQ24 ~ IRQ27 Priority Register (NVIC\_IPR6)**

Register	Offset	R/W	Description	Reset Value
NVIC_IPR6	SCS_BA+0x418	R/W	IRQ24 ~ IRQ27 Priority Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PRI_27		Reserved					
23	22	21	20	19	18	17	16
PRI_26		Reserved					
15	14	13	12	11	10	9	8
PRI_25		Reserved					
7	6	5	4	3	2	1	0
PRI_24		Reserved					

Bits	Description	
[31:30]	PRI_27	<b>Priority of IRQ27</b> "0" denotes the highest priority and "3" denotes the lowest priority.
[29:24]	Reserved	Reserved.
[23:22]	PRI_26	<b>Priority of IRQ26</b> "0" denotes the highest priority and "3" denotes the lowest priority.
[21:16]	Reserved	Reserved.
[15:14]	PRI_25	<b>Priority of IRQ25</b> "0" denotes the highest priority and "3" denotes the lowest priority.
[13:8]	Reserved	Reserved.
[7:6]	PRI_24	<b>Priority of IRQ24</b> "0" denotes the highest priority and "3" denotes the lowest priority.
[5:0]	Reserved	Reserved.

**IRQ28 ~ IRQ31 Priority Register (NVIC\_IPR7)**

Register	Offset	R/W	Description	Reset Value
NVIC_IPR7	SCS_BA+0x41C	R/W	IRQ28 ~ IRQ31 Priority Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PRI_31		Reserved					
23	22	21	20	19	18	17	16
PRI_30		Reserved					
15	14	13	12	11	10	9	8
PRI_29		Reserved					
7	6	5	4	3	2	1	0
PRI_28		Reserved					

Bits	Description
[31:30]	<b>PRI_31</b> <b>Priority of IRQ31</b> "0" denotes the highest priority and "3" denotes the lowest priority.
[29:24]	<b>Reserved</b> Reserved.
[23:22]	<b>PRI_30</b> <b>Priority of IRQ30</b> "0" denotes the highest priority and "3" denotes the lowest priority.
[21:16]	<b>Reserved</b> Reserved.
[15:14]	<b>PRI_29</b> <b>Priority of IRQ29</b> "0" denotes the highest priority and "3" denotes the lowest priority.
[13:8]	<b>Reserved</b> Reserved.
[7:6]	<b>PRI_28</b> <b>Priority of IRQ28</b> "0" denotes the highest priority and "3" denotes the lowest priority.
[5:0]	<b>Reserved</b> Reserved.

6.2.14.4 Interrupt Source Register Map

Besides the interrupt control registers associated with the NVIC, the NUC126 series also implement some specific control registers to facilitate the interrupt functions, including “interrupt source identification”, “NMI source selection” and “interrupt test mode”, which are described below.

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

Register	Offset	R/W	Description	Reset Value
<b>INT Base Address:</b>				
<b>INT_BA = 0x5000_0300</b>				
IRQ0_SRC	INT_BA+0x00	R	IRQ0 (BOD) Interrupt Source Identity	0XXXXX_XXXX
IRQ1_SRC	INT_BA+0x04	R	IRQ1 (WDT) Interrupt Source Identity	0XXXXX_XXXX
IRQ2_SRC	INT_BA+0x08	R	IRQ2 (EINT0/2/4) Interrupt Source Identity	0XXXXX_XXXX
IRQ3_SRC	INT_BA+0x0C	R	IRQ3 (EINT1/3/5) Interrupt Source Identity	0XXXXX_XXXX
IRQ4_SRC	INT_BA+0x10	R	IRQ4 (GPA/B) Interrupt Source Identity	0XXXXX_XXXX
IRQ5_SRC	INT_BA+0x14	R	IRQ5 (GPC/D/E/F) Interrupt Source Identity	0XXXXX_XXXX
IRQ6_SRC	INT_BA+0x18	R	IRQ6 (PWM0) Interrupt Source Identity	0XXXXX_XXXX
IRQ7_SRC	INT_BA+0x1C	R	IRQ7 (PWM1) Interrupt Source Identity	0XXXXX_XXXX
IRQ8_SRC	INT_BA+0x20	R	IRQ8 (TMR0) Interrupt Source Identity	0XXXXX_XXXX
IRQ9_SRC	INT_BA+0x24	R	IRQ9 (TMR1) Interrupt Source Identity	0XXXXX_XXXX
IRQ10_SRC	INT_BA+0x28	R	IRQ10 (TMR2) Interrupt Source Identity	0XXXXX_XXXX
IRQ11_SRC	INT_BA+0x2C	R	IRQ11 (TMR3) Interrupt Source Identity	0XXXXX_XXXX
IRQ12_SRC	INT_BA+0x30	R	IRQ12 (UART0/2) Interrupt Source Identity	0XXXXX_XXXX
IRQ13_SRC	INT_BA+0x34	R	IRQ13 (UART1) Interrupt Source Identity	0XXXXX_XXXX
IRQ14_SRC	INT_BA+0x38	R	IRQ14 (SPI0) Interrupt Source Identity	0XXXXX_XXXX
IRQ15_SRC	INT_BA+0x3C	R	IRQ15 (SPI1) Interrupt Source Identity	0XXXXX_XXXX
IRQ16_SRC	INT_BA+0x40	R	Reserved	0XXXXX_XXXX
IRQ17_SRC	INT_BA+0x44	R	Reserved	0XXXXX_XXXX
IRQ18_SRC	INT_BA+0x48	R	IRQ18 (I <sup>2</sup> C0) Interrupt Source Identity	0XXXXX_XXXX
IRQ19_SRC	INT_BA+0x4C	R	IRQ19 (I <sup>2</sup> C1) Interrupt Source Identity	0XXXXX_XXXX
IRQ20_SRC	INT_BA+0x50	R	Reserved	0XXXXX_XXXX
IRQ21_SRC	INT_BA+0x54	R	Reserved	0XXXXX_XXXX
IRQ22_SRC	INT_BA+0x58	R	IRQ22 (USCI0/1/2) Interrupt Source Identity	0XXXXX_XXXX
IRQ23_SRC	INT_BA+0x5C	R	IRQ23 (USB0) Interrupt Source Identity	0XXXXX_XXXX
IRQ24_SRC	INT_BA+0x60	R	IRQ24 (SC0/1) Interrupt Source Identity	0XXXXX_XXXX

<b>IRQ25_SRC</b>	INT_BA+0x64	R	IRQ25 (ACMP) Interrupt Source Identity	0xFFFF_XXXX
<b>IRQ26_SRC</b>	INT_BA+0x68	R	IRQ26 (PDMA) Interrupt Source Identity	0xFFFF_XXXX
<b>IRQ27_SRC</b>	INT_BA+0x6C	R	Reserved	0xFFFF_XXXX
<b>IRQ28_SRC</b>	INT_BA+0x70	R	IRQ28 (PWRWU) Interrupt Source Identity	0xFFFF_XXXX
<b>IRQ29_SRC</b>	INT_BA+0x74	R	IRQ29 (ADC) Interrupt Source Identity	0xFFFF_XXXX
<b>IRQ30_SRC</b>	INT_BA+0x78	R	IRQ30 (IRC/CLKD) Interrupt Source Identity	0xFFFF_XXXX
<b>IRQ31_SRC</b>	INT_BA+0x7C	R	IRQ31 (RTC) Interrupt Source Identity	0xFFFF_XXXX
<b>NMI_SEL</b>	INT_BA+0x80	R/W	NMI Source Interrupt Select Control Register	0x0000_0000

6.2.14.5 Interrupt Source Register Description

**Interrupt Source Identity Register (IRQn\_SRC)**

Register	Offset	R/W	Description	Reset Value
IRQ0_SRC	INT_BA+0x00	R	IRQ0 (BOD) Interrupt Source Identity	0xFFFF_FFFF
IRQ1_SRC	INT_BA+0x04	R	IRQ1 (WDT) Interrupt Source Identity	0xFFFF_FFFF
IRQ2_SRC	INT_BA+0x08	R	IRQ2 (EINT0/2/4) Interrupt Source Identity	0xFFFF_FFFF
IRQ3_SRC	INT_BA+0x0C	R	IRQ3 (EINT1/3/5) Interrupt Source Identity	0xFFFF_FFFF
IRQ4_SRC	INT_BA+0x10	R	IRQ4 (GPA/B) Interrupt Source Identity	0xFFFF_FFFF
IRQ5_SRC	INT_BA+0x14	R	IRQ5 (GPC/D/E/F) Interrupt Source Identity	0xFFFF_FFFF
IRQ6_SRC	INT_BA+0x18	R	IRQ6 (PWM0) Interrupt Source Identity	0xFFFF_FFFF
IRQ7_SRC	INT_BA+0x1C	R	IRQ7 (PWM1) Interrupt Source Identity	0xFFFF_FFFF
IRQ8_SRC	INT_BA+0x20	R	IRQ8 (TMR0) Interrupt Source Identity	0xFFFF_FFFF
IRQ9_SRC	INT_BA+0x24	R	IRQ9 (TMR1) Interrupt Source Identity	0xFFFF_FFFF
IRQ10_SRC	INT_BA+0x28	R	IRQ10 (TMR2) Interrupt Source Identity	0xFFFF_FFFF
IRQ11_SRC	INT_BA+0x2C	R	IRQ11 (TMR3) Interrupt Source Identity	0xFFFF_FFFF
IRQ12_SRC	INT_BA+0x30	R	IRQ12 (UART0/2) Interrupt Source Identity	0xFFFF_FFFF
IRQ13_SRC	INT_BA+0x34	R	IRQ13 (UART1) Interrupt Source Identity	0xFFFF_FFFF
IRQ14_SRC	INT_BA+0x38	R	IRQ14 (SPI0) Interrupt Source Identity	0xFFFF_FFFF
IRQ15_SRC	INT_BA+0x3C	R	IRQ15 (SPI1) Interrupt Source Identity	0xFFFF_FFFF
IRQ16_SRC	INT_BA+0x40	R	Reserved	0xFFFF_FFFF
IRQ17_SRC	INT_BA+0x44	R	Reserved	0xFFFF_FFFF
IRQ18_SRC	INT_BA+0x48	R	IRQ18 (I <sup>2</sup> C0) Interrupt Source Identity	0xFFFF_FFFF
IRQ19_SRC	INT_BA+0x4C	R	IRQ19 (I <sup>2</sup> C1) Interrupt Source Identity	0xFFFF_FFFF
IRQ20_SRC	INT_BA+0x50	R	Reserved	0xFFFF_FFFF
IRQ21_SRC	INT_BA+0x54	R	Reserved	0xFFFF_FFFF
IRQ22_SRC	INT_BA+0x58	R	IRQ22 (USCI0/1/2) Interrupt Source Identity	0xFFFF_FFFF
IRQ23_SRC	INT_BA+0x5C	R	IRQ23 (USB0) Interrupt Source Identity	0xFFFF_FFFF
IRQ24_SRC	INT_BA+0x60	R	IRQ24 (SC0/1) Interrupt Source Identity	0xFFFF_FFFF
IRQ25_SRC	INT_BA+0x64	R	IRQ25 (ACMP) Interrupt Source Identity	0xFFFF_FFFF
IRQ26_SRC	INT_BA+0x68	R	IRQ26 (PDMA) Interrupt Source Identity	0xFFFF_FFFF
IRQ27_SRC	INT_BA+0x6C	R	Reserved	0xFFFF_FFFF



<b>IRQ28_SRC</b>	INT_BA+0x70	R	IRQ28 (PWRWU) Interrupt Source Identity	0xFFFF_XXXX
<b>IRQ29_SRC</b>	INT_BA+0x74	R	IRQ29 (ADC) Interrupt Source Identity	0xFFFF_XXXX
<b>IRQ30_SRC</b>	INT_BA+0x78	R	IRQ30 (IRC/CLKD) Interrupt Source Identity	0xFFFF_XXXX
<b>IRQ31_SRC</b>	INT_BA+0x7C	R	IRQ31 (RTC) Interrupt Source Identity	0xFFFF_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
Reserved							
7	6	5	4	3	2	1	0
Reserved				INT_SRC			

Bits	Description	
[31:4]	<b>Reserved</b>	Reserved.
[3:0]	<b>INT_SRC</b>	<b>Interrupt Source</b> Define the interrupt sources for interrupt event.

Bits	Address	INT-Num	Description
[2:0]	INT_BA+0x00	0	Bit2: 0 Bit1: 0 Bit0: BOD_INT
[2:0]	INT_BA+0x04	1	Bit2: 0 Bit1: WWDT_INT Bit0: WDT_INT
[2:0]	INT_BA+0x08	2	Bit2: EINT4 – external interrupt 4 from PE.0 Bit1: EINT2 – external interrupt 2 from PC.0 Bit0: EINT0 – external interrupt 0 from PA.0/PD.2/PE.4
[2:0]	INT_BA+0x0C	3	Bit2: EINT5 – external interrupt 5 from PF.0 Bit1: EINT3 – external interrupt 3 from PD.0 Bit0: EINT1 – external interrupt 1 from PB.0/PD.3/PE.5
[2:0]	INT_BA+0x10	4	Bit2: 0 Bit1: GPB_INT Bit0: GPA_INT
[3:0]	INT_BA+0x14	5	Bit3: GPF_INT Bit2: GPE_INT Bit1: GPD_INT

			Bit0: GPC_INT
[3:0]	INT_BA+0x18	6	Bit2: 0 Bit1: BRAKE0_INT Bit0: PWM0_INT
[2:0]	INT_BA+0x1C	7	Bit2: 0 Bit1: BRAKE1_INT Bit0: PWM1_INT
[2:0]	INT_BA+0x20	8	Bit2: 0 Bit1: 0 Bit0: TMR0_INT
[2:0]	INT_BA+0x24	9	Bit2: 0 Bit1: 0 Bit0: TMR1_INT
[2:0]	INT_BA+0x28	10	Bit2: 0 Bit1: 0 Bit0: TMR2_INT
[2:0]	INT_BA+0x2C	11	Bit2: 0 Bit1: 0 Bit0: TMR3_INT
[2:0]	INT_BA+0x30	12	Bit2: 0 Bit1: UART2_INT Bit0: UART0_INT
[2:0]	INT_BA+0x34	13	Bit2: 0 Bit1: 0 Bit0: UART1_INT
[2:0]	INT_BA+0x38	14	Bit2: 0 Bit1: 0 Bit0: SPI0_INT
[2:0]	INT_BA+0x3C	15	Bit2: 0 Bit1: 0 Bit0: SPI1_INT
[2:0]	INT_BA+0x40	16	Reserved
[2:0]	INT_BA+0x44	17	Reserved
[2:0]	INT_BA+0x48	18	Bit2: 0 Bit1: 0 Bit0: I2C0_INT
[2:0]	INT_BA+0x4C	19	Bit2: 0 Bit1: 0 Bit0: I2C1_INT
[3:0]	INT_BA+0x58	22	Bit2: USCI2_INT Bit1: USCI1_INT Bit0: USCI0_INT
[2:0]	INT_BA+0x5C	23	Bit2: 0

			Bit1: 0 Bit0: USBD_INT
[2:0]	INT_BA+0x60	24	Bit2: 0 Bit1: SMC1_INT Bit0: SMC0_INT
[2:0]	INT_BA+0x64	25	Bit2: 0 Bit1: 0 Bit0: ACMP_INT
[2:0]	INT_BA+0x68	26	Bit2: 0 Bit1: 0 Bit0: PDMA_INT
[2:0]	INT_BA+0x6C	27	Reserved
[2:0]	INT_BA+0x70	28	Bit2: 0 Bit1: 0 Bit0: PWRWU_INT
[2:0]	INT_BA+0x74	29	Bit2: 0 Bit1: 0 Bit0: ADC_INT0
[2:0]	INT_BA+0x78	30	Bit2: SRAMF_INT Bit1: CLKD_INT Bit0: IRC_INT
[2:0]	INT_BA+0x7C	31	Bit2: 0 Bit1: 0 Bit0: RTC_INT

**NMI Source Interrupt Select Control Register (NMI\_SEL)**

Register	Offset	R/W	Description	Reset Value
NMI_SEL	INT_BA+0x80	R/W	NMI Source Interrupt Select 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							NMI_EN
7	6	5	4	3	2	1	0
Reserved			NMI_SEL				

Bits	Description	
[31:8]	Reserved	Reserved.
[8]	NMI_EN	<p><b>NMI Interrupt Enable Bit (Write Protect)</b></p> <p>0 = NMI interrupt Disabled. 1 = NMI interrupt Enabled.</p> <p><b>Note:</b> This bit is write protected. Refer to the SYS_REGLCTL register.</p>
[7:5]	Reserved	Reserved.
[4:0]	NMI_SEL	<p><b>NMI Interrupt Source Selection</b></p> <p>The NMI interrupt to Cortex<sup>®</sup>-M0 can be selected from one of the peripheral interrupt by setting NMI_SEL.</p>

### 6.2.15 System Control

The Cortex<sup>®</sup>-M0 status and operating mode control are managed by System Control Registers. Including CPUID, Cortex<sup>®</sup>-M0 interrupt priority and Cortex<sup>®</sup>-M0 power management can be controlled through these system control registers.

For more detailed information, please refer to the “ARM<sup>®</sup> Cortex<sup>®</sup>-M0 Technical Reference Manual” and “ARM<sup>®</sup> v6-M Architecture Reference Manual”.

#### 6.2.15.1 System Control Register Map

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

Register	Offset	R/W	Description	Reset Value
<b>SCS Base Address:</b>				
<b>SCS_BA = 0xE000_E000</b>				
<b>CPUID</b>	SCS_BA+0xD00	R	CPUID Register	0x410C_C200
<b>ICSR</b>	SCS_BA+0xD04	R/W	Interrupt Control and State Register	0x0000_0000
<b>AIRCR</b>	SCS_BA+0xD0C	R/W	Application Interrupt and Reset Control Register	0xFA05_0000
<b>SCR</b>	SCS_BA+0xD10	R/W	System Control Register	0x0000_0000
<b>SHPR2</b>	SCS_BA+0xD1C	R/W	System Handler Priority Register 2	0x0000_0000
<b>SHPR3</b>	SCS_BA+0xD20	R/W	System Handler Priority Register 3	0x0000_0000

6.2.15.2 System Control Register Description

**CPUID Register (CPUID)**

Register	Offset	R/W	Description	Reset Value
CPUID	SCS_BA+0xD00	R	CPUID Register	0x410C_C200

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

Bits	Description	
[31:24]	<b>IMPLEMENTER</b>	<b>Implementer Code Assigned by ARM</b> Implementer code assigned by ARM. (ARM = 0x41).
[23:20]	<b>Reserved</b>	Reserved.
[19:16]	<b>PART</b>	<b>Architecture of the Processor</b> Read as 0xC for ARMv6-M parts
[15:4]	<b>PARTNO</b>	<b>Part Number of the Processor</b> Read as 0xC20.
[3:0]	<b>REVISION</b>	<b>Revision Number</b> Read as 0x0

**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	Reserved		PENDSVSET	PENDSVCLR	PENDSTSET	PENDSTCLR	Reserved
23	22	21	20	19	18	17	16
ISRPREEMPT	ISRPENDING	Reserved				VECTPENDING[5:4]	
15	14	13	12	11	10	9	8
VECTPENDING[3:0]				Reserved			
7	6	5	4	3	2	1	0
Reserved		VECTACTIVE[5:0]					

Bits	Description
[31]	<p><b>NMIPENDSET</b></p> <p><b>NMI Set-pending Bit</b>                      Write Operation:                      0 = No effect.                      1 = Changes NMI exception state to pending.                      Read Operation:                      0 = NMI exception not pending.                      1 = NMI exception pending.                      Because NMI is the highest-priority exception, normally the processor enters the NMI exception handler as soon as it detects a write of 1 to this bit. Entering the handler then clears this bit to 0. This means a read of this bit by the NMI exception handler returns 1 only if the NMI signal is reasserted while the processor is executing that handler.</p>
[30:29]	Reserved
[28]	<p><b>PENDSVSET</b></p> <p><b>PendSV Set-pending Bit</b>                      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.  <b>Note:</b> Writing 1 to this bit is the only way to set the PendSV exception state to pending.</p>
[27]	<p><b>PENDSVCLR</b></p> <p><b>PendSV Clear-pending Bit</b>                      Write Operation:                      0 = No effect.                      1 = Removes the pending state from the PendSV exception.                      This is a write only bit. When you want to clear PENDSV bit, you must “write 0 to PENDSVSET and write 1 to PENDSVCLR” at the same time.</p>
[26]	<p><b>PENDSTSET</b></p> <p><b>SysTick Exception Set-pending Bit</b>                      Write Operation:</p>

		<p>0 = No effect.                  1 = Changes SysTick exception state to pending.                  Read Operation:                  0 = SysTick exception is not pending.                  1 = SysTick exception is pending.</p>
[25]	<b>PENDSTCLR</b>	<p><b>SysTick Exception Clear-pending Bit</b>                  Write Operation:                  0 = No effect.                  1 = Removes the pending state from the SysTick exception.                  This is a write only bit. When you want to clear PENDST bit, you must “write 0 to PENDSTSET and write 1 to PENDSTCLR” at the same time.</p>
[24]	<b>Reserved</b>	Reserved.
[23]	<b>ISRPREEMPT</b>	<p><b>If Set, a Pending Exception Will Be Serviced on Exit From the Debug Halt State</b>                  This bit is read only.</p>
[22]	<b>ISRPENDING</b>	<p><b>Interrupt Pending Flag, Excluding NMI and Faults:</b>                  0 = Interrupt not pending.                  1 = Interrupt pending.                  This bit is read only.</p>
[21:18]	<b>Reserved</b>	Reserved.
[17:12]	<b>VECTPENDING</b>	<p><b>Indicates the Exception Number of the Highest Priority Pending Enabled Exception:</b>                  0 = No pending exceptions.                  Non-zero = Exception number of the highest priority pending enabled exception.</p>
[11:6]	<b>Reserved</b>	Reserved.
[5:0]	<b>VECTACTIVE</b>	<p><b>Contains the Active Exception Number</b>                  0 = Thread mode.                  Non-zero = Exception number of the currently active exception.</p>



**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_0000

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

Bits	Description	
[31:16]	VECTORKEY	<p><b>Register Access Key</b></p> <p>Write Operation: When writing to this register, the VECTORKEY field need to be set to 0x05FA, otherwise the write operation would be ignored. The VECTORKEY field is used to prevent accidental write to this register from resetting the system or clearing of the exception status.</p> <p>Read Operation: Read as 0xFA05.</p>
[15:3]	Reserved	Reserved.
[2]	SYSRESETREQ	<p><b>System Reset Request</b></p> <p>Writing this bit 1 will cause a reset signal to be asserted to the chip to indicate a reset is requested.</p> <p>The bit is a write only bit and self-clears as part of the reset sequence.</p>
[1]	VECTCLRACTIVE	<p><b>Exception Active Status Clear Bit</b></p> <p>Reserved for debug use. When writing to the register, user must write 0 to this bit, otherwise behavior is unpredictable.</p>
[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	<p><b>Send Event on Pending Bit</b></p> <p>0 = Only enabled interrupts or events can wake-up the processor, disabled interrupts are excluded.</p> <p>1 = Enabled events and all interrupts, including disabled interrupts, can wake-up the processor.</p> <p>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.</p> <p>The processor also wakes up on execution of an SEV instruction or an external event.</p>
[3]	Reserved	Reserved.
[2]	SLEEPDEEP	<p><b>Processor Deep Sleep and Sleep Mode Selection</b></p> <p>Controls whether the processor uses sleep or deep sleep as its low power mode:</p> <p>0 = Sleep mode.</p> <p>1 = Deep Sleep mode.</p>
[1]	SLEEPONEXIT	<p><b>Sleep-on-exit Enable Bit</b></p> <p>This bit indicates sleep-on-exit when returning from Handler mode to Thread mode.</p> <p>0 = Do not sleep when returning to Thread mode.</p> <p>1 = Enter Sleep or Deep Sleep when returning from ISR to Thread mode.</p> <p>Setting this bit to 1 enables an interrupt driven application to avoid returning to an empty main application..</p>
[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	<b>Priority of System Handler 15 – SysTick</b> “0” denotes the highest priority and “3” denotes the lowest priority
[29:24]	Reserved	Reserved.
[23:22]	PRI_14	<b>Priority of System Handler 14 – PendSV</b> “0” denotes the highest priority and “3” denotes the lowest priority
[21:0]	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<sup>®</sup>-M0 core executes the WFI instruction. After that, chip enters Power-down mode and wait for wake-up interrupt source triggered to leave Power-down mode. In Power-down mode, the clock controller turns off the 4~24 MHz external high speed crystal (HXT), internal 22.1184 MHz internal high speed RC oscillator (HIRC) and 48 MHz internal high speed RC oscillator (HIRC48) to reduce the overall system power consumption. Figure 6.3-1 shows the clock generator and the overview of the clock source control.

The clock generator consists of 6 clock sources, which are listed below:

- 32.768 kHz external low-speed crystal oscillator (LXT)
- 4~24 MHz external high speed crystal oscillator (HXT)
- Programmable PLL output clock frequency (PLLFOUT), PLL source can be selected from external 4~24 MHz external high speed crystal (HXT) or 22.1184 MHz internal high speed oscillator (HIRC)
- 22.1184 MHz internal high speed RC oscillator (HIRC)
- 48 MHz internal high speed RC oscillator (HIRC48)
- 10 kHz internal low speed RC oscillator (LIRC)

Each of these clock sources has certain stable time to wait for clock operating at stable frequency. When clock source is enabled, a stable counter start counting and correlated clock stable index (HIRCSTB(CLK\_STATUS[4]), LIRCSTB(CLK\_STATUS[3]), PLLSTB(CLK\_STATUS[2]), HXTSTB(CLK\_STATUS[0]), LXTSTB(CLK\_STATUS[1]) and HIRC48STB(CLK\_STATUS[5])) are set to 1 after stable counter value reach a define value as shown in Table 6.3-1. System and peripheral can use the clock as its operating clock only when correlate clock stable index is set to 1. The clock stable index will auto clear when user disables the clock source (LIRCEN(CLK\_PWRCTL[3]), HIRCEN(CLK\_PWRCTL[2]), HXTEN(CLK\_PWRCTL[0]), PD(CLK\_PLLCTL[16]), LXTEN(CLK\_PWRCTL[1]) and HIRC48EN(CLK\_PWRCTL[13])). Besides, the clock stable index of HXT, HIRC and PLL will auto clear when chip enter power-down and clock stable counter will re-counting after chip wake-up if correlate clock is enabled.

Clock Source	Clock Stable Count Value	Clock Stable time
HXT	4096 HXT clock	341.33 uS for 12 Mhz
PLL	It's based on the value of STBSEL (CLK_PLLCTL[23]) STBSEL = 0, stable count is 6144 clocks of PLL clock source. STBSEL = 1, stable count is 12288 clocks of PLL clock source. (Default)	STBSEL = 0, 512 uS for 512 Mhz STBSEL = 1, 1024 uS for 12 Mhz
HIRC48	512 HIRC48 clock	10.67 uS for 48 Mhz
HIRC	256 HIRC clock	11.574 uS for 22.1184 Mhz
LIRC	1 LIRC clock	100 uS for 10 kHz
LXT	1 LXT clock	30.51 uS for 32.768 khz

Table 6.3-1 Clock Stable Count Value Table

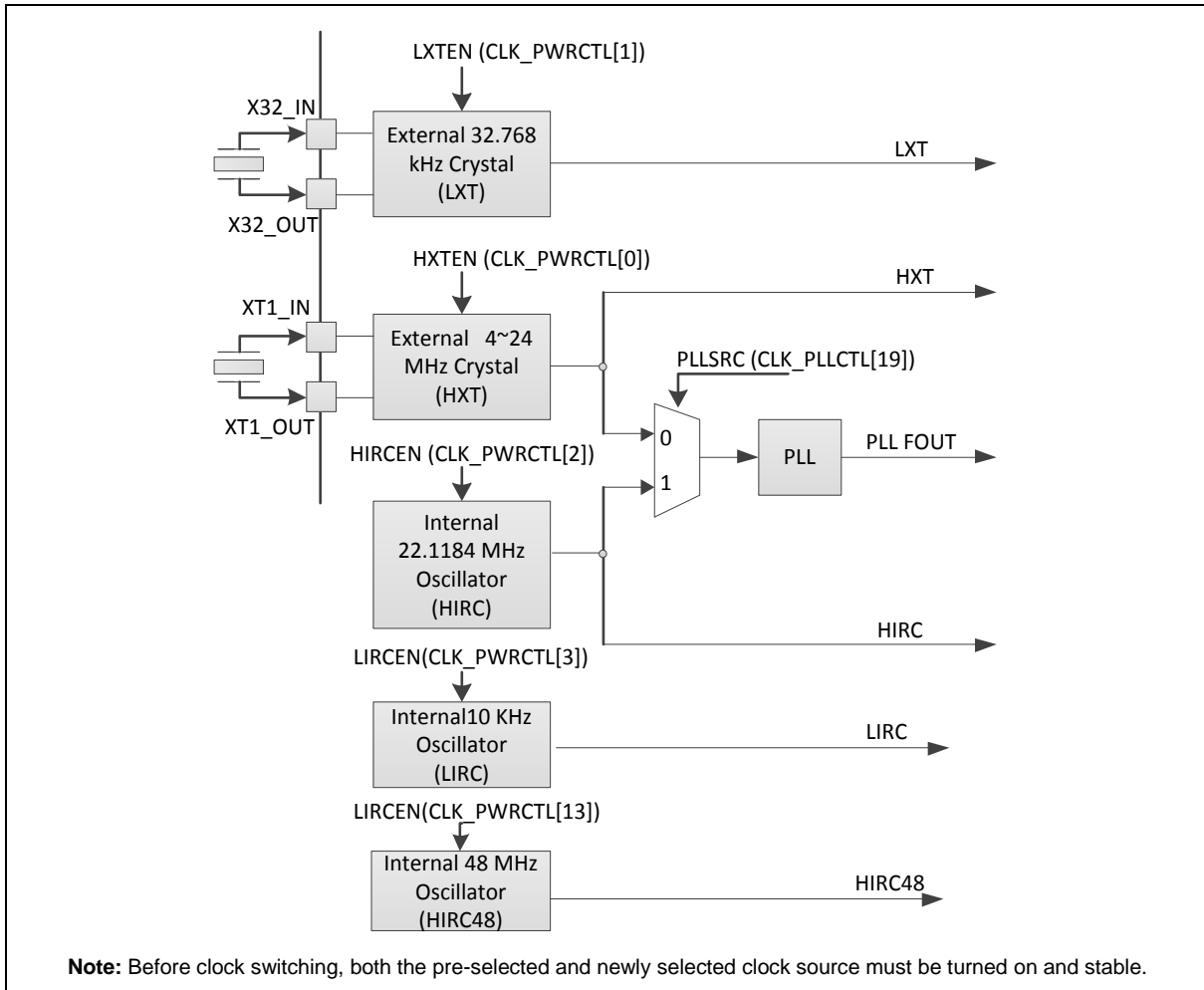


Figure 6.3-1 Clock Generator Block Diagram

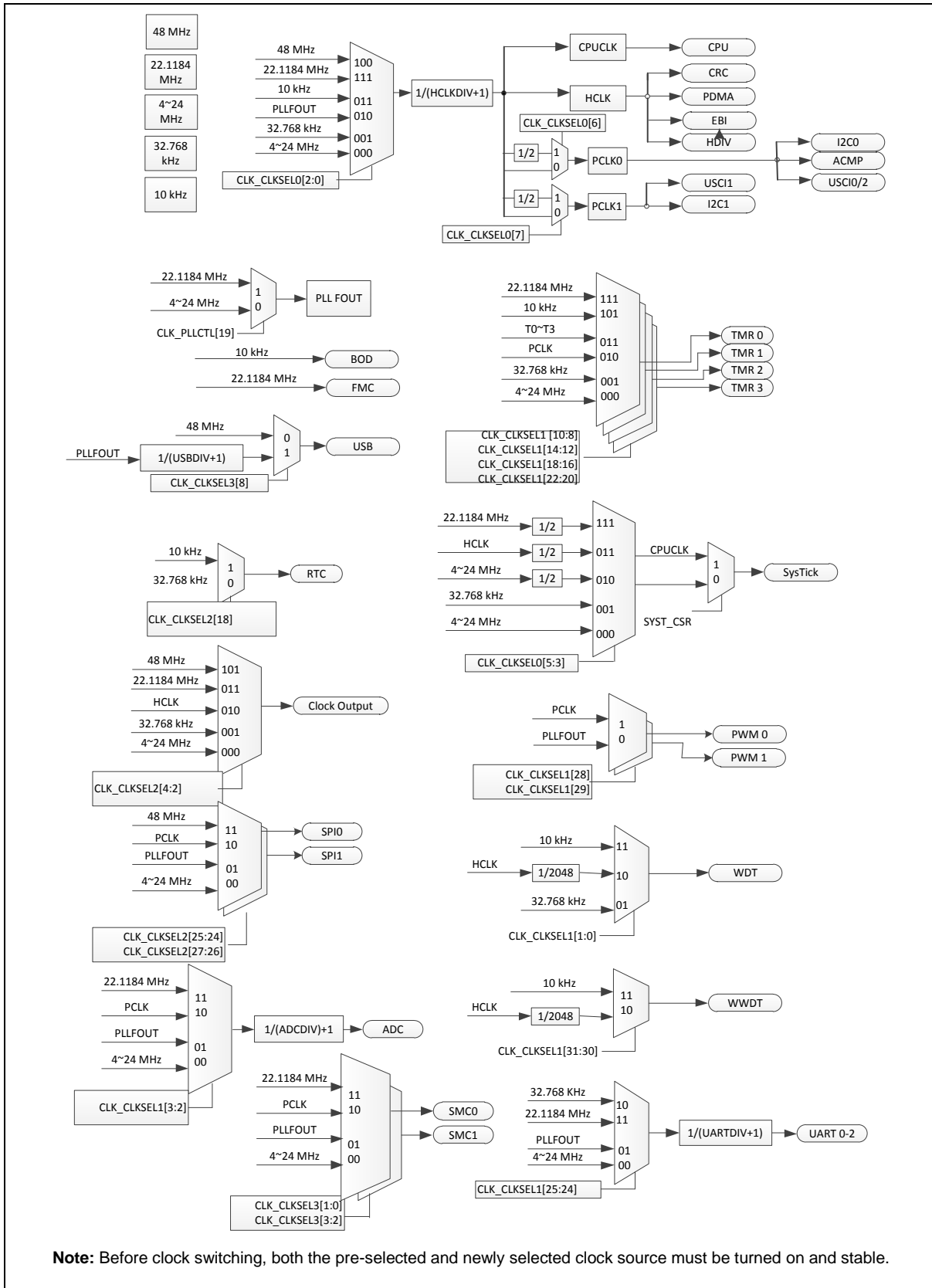


Figure 6.3-2 Clock Generator Global View Diagram

### 6.3.2 System Clock and SysTick Clock

The system clock has 6 clock sources, which were 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.

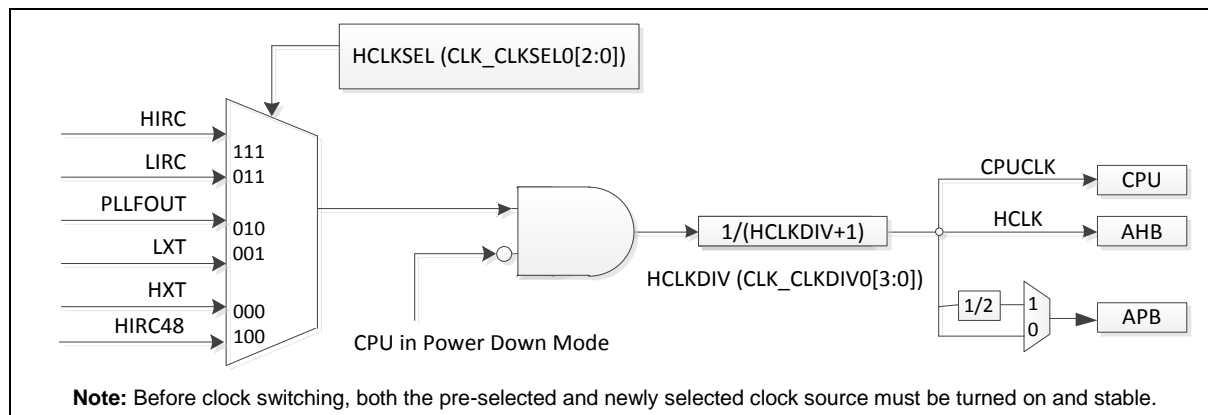


Figure 6.3-3 System Clock Block Diagram

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 HIRC 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 HIRC if HXT clock stop being detected on the following condition: system clock source comes from HXT or system clock source comes from PLL with HXT as the input of PLL. If HXT clock stop condition is detected, the HXTFIF (CLK\_CLKDSTS[0]) is set to 1 and chip will enter interrupt if HXTFIEN (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 HIRC procedure is shown in Figure 6.3-4.



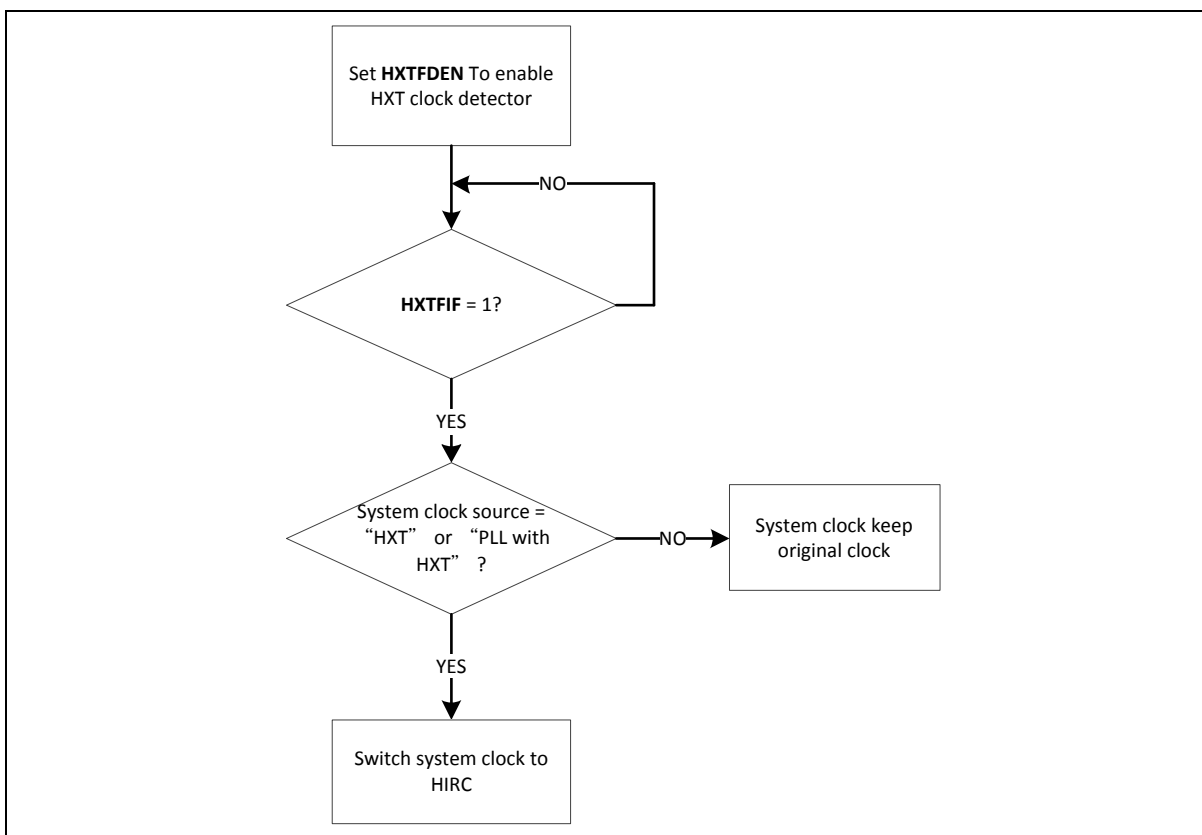


Figure 6.3-4 HXT Stop Protect Procedure

The clock source of SysTick in Cortex<sup>®</sup>-M0 core can use CPU clock or external clock (SYST\_CSR[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.

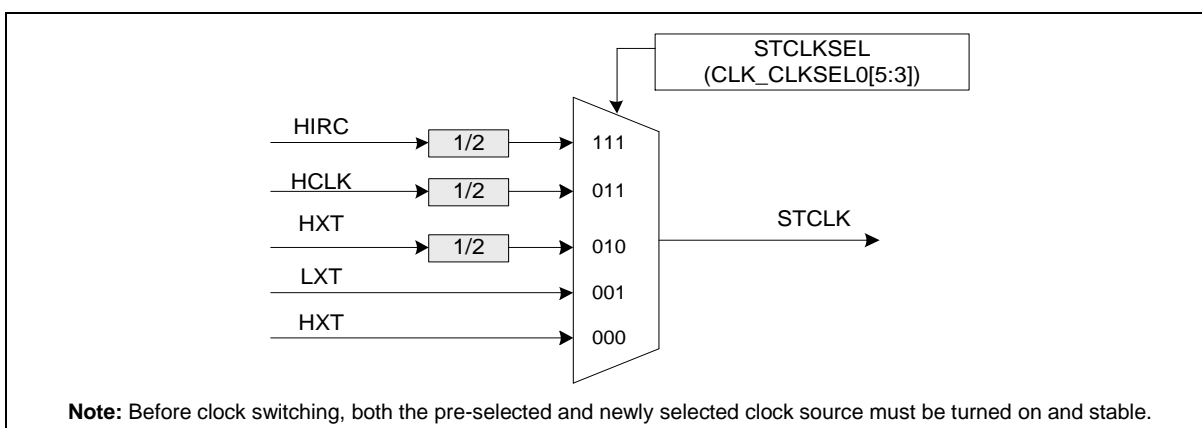


Figure 6.3-5 SysTick Clock Control Block Diagram

### 6.3.3 Peripherals Clock

The peripherals clock had different clock source switch setting, which depends on the different peripheral. Please refer to the CLK\_CLKSEL1, CLK\_CLKSEL2 and CLK\_CLKSEL3 register description in section 6.3.7.

### 6.3.4 Power-down Mode Clock

When entering Power-down mode, system clocks, some clock sources, and some peripheral clocks are disabled. Some clock sources and peripherals clock are still active in Power-down mode.

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

- Clock Generator
  - 10 kHz internal low-speed RC oscillator (LIRC) clock
  - 32.768 kHz external low-speed crystal oscillator (LXT) clock
- Peripherals Clock (When the modules adopt LXT or LIRC as clock source)

### 6.3.5 Clock Output

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 CLKO 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.

The output formula is  $F_{out} = F_{in}/2^{(N+1)}$ , where  $F_{in}$  is the input clock frequency,  $F_{out}$  is the clock divider output frequency and N is the 4-bit value in FREQSEL (CLK\_CLKOCTL[3:0]).

When writing 1 to CLKOEN (CLK\_CLKOCTL[4]), the chained counter starts to count. When writing 0 to CLKOEN (CLK\_CLKOCTL[4]), the chained counter continuously runs till divided clock reaches low state and stay in low state.

if DIV11EN(CLK\_CLKOCTL[5]) set to 1, the clock output clock (CLKO\_CLK) will bypass power-of-2 frequency divider. The clock output clock will be output to CLKO pin directly.

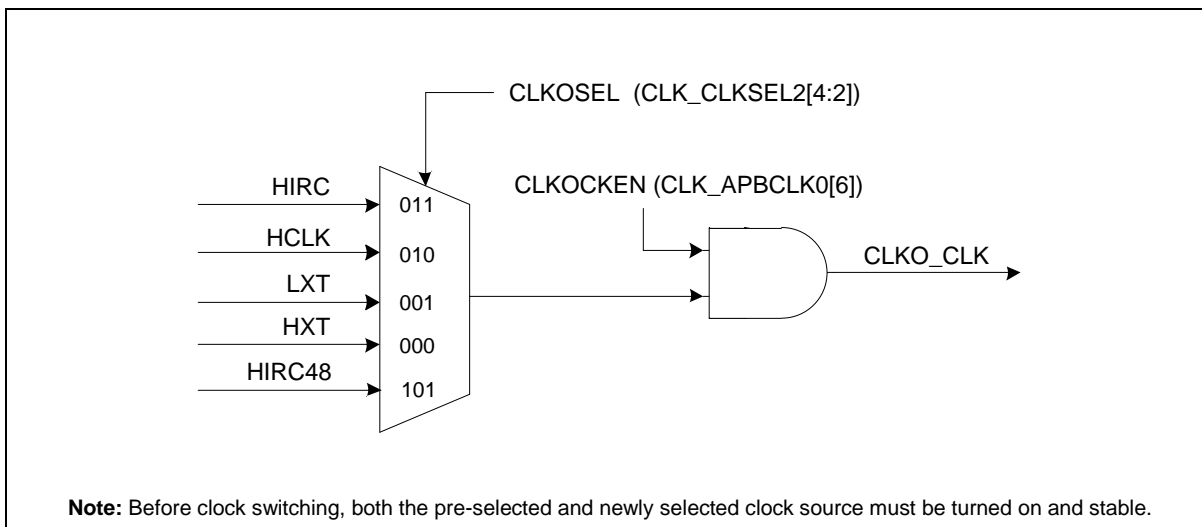


Figure 6.3-6 Clock Source of Clock Output

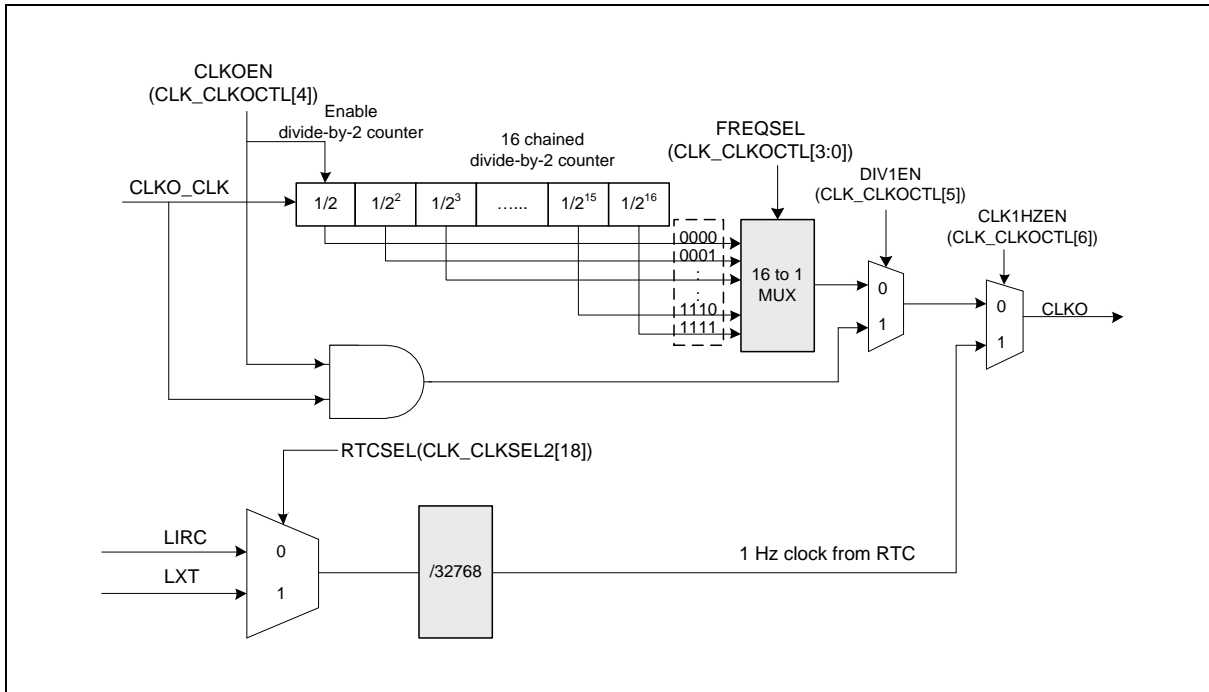


Figure 6.3-7 Clock Output Block Diagram

### 6.3.6 Register Map

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

Register	Offset	R/W	Description	Reset Value
<b>CLK Base Address:</b> CLK_BA = 0x5000_0200				
CLK_PWRCTL	CLK_BA+0x00	R/W	System Power-down Control Register	0x0000_1C1X
CLK_AHBCLK	CLK_BA+0x04	R/W	AHB Devices Clock Enable Control Register	0x003F_8004
CLK_APBCLK0	CLK_BA+0x08	R/W	APB Devices Clock Enable Control Register 0	0x0000_0001
CLK_APBCLK1	CLK_BA+0x30	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_003X
CLK_CLKSEL1	CLK_BA+0x14	R/W	Clock Source Select Control Register 1	0x3377_770F
CLK_CLKSEL2	CLK_BA+0x1C	R/W	Clock Source Select Control Register 2	0x0002_0008
CLK_CLKSEL3	CLK_BA+0x34	R/W	Clock Source Select Control Register 3	0x0000_0000
CLK_CLKDIV0	CLK_BA+0x18	R/W	Clock Divider Number Register 0	0x0000_0000
CLK_CLKDIV1	CLK_BA+0x38	R/W	Clock Divider Number Register 1	0x0000_0000
CLK_PLLCTL	CLK_BA+0x20	R/W	PLL Control Register	0x008D_8418
CLK_STATUS	CLK_BA+0x0C	R	Clock Status Monitor Register	0x0000_00XX
CLK_CLKOCTL	CLK_BA+0x24	R/W	Clock Output Control Register	0x0000_0000
CLK_BODCLK	CLK_BA+0x40	R/W	Clock Source Select for BOD 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 Detector Upper Boundary Register	0x0000_0000
CLK_CDLOWB	CLK_BA+0x7c	R/W	Clock Frequency Detector Low Boundary Register	0x0000_0000

6.3.7 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	0x0000_1C1X

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		HIRC48EN	HXTSELTYP	HXTGAIN		Reserved	
7	6	5	4	3	2	1	0
PDEN	PDWKIF	PDWKIEN	PDWKDLY	LIRCEN	HIRCEN	LXTEN	HXTEN

Bits	Description	
[31:14]	Reserved	Reserved.
[13]	HIRC48EN	<p><b>HIRC48 Enable Bit (Write Protect)</b></p> <p>0 = 48 MHz internal high speed RC oscillator (HIRC48) Disabled.                      1 = 48 MHz internal high speed RC oscillator (HIRC48) Enabled.</p> <p><b>Note:</b> This bit is write protected. Refer to the SYS_REGLCTL register.</p>
[12]	HXTSELTYP	<p><b>HXT Crystal Type Select Bit (Write Protect)</b></p> <p>This is a protected register. Please refer to open lock sequence to program it.</p> <p>0 = Select INV type.                      1 = Select GM type.</p> <p><b>Note:</b> This bit is write protected. Refer to the SYS_REGLCTL register.</p>
[11:10]	HXTGAIN	<p><b>HXT Gain Control Bit (Write Protect)</b></p> <p>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.</p> <p>00 = HXT frequency is lower than from 8 MHz.                      01 = HXT frequency is from 8 MHz to 12 MHz.                      10 = HXT frequency is from 12 MHz to 16 MHz.                      11 = HXT frequency is higher than 16 MHz.</p> <p><b>Note:</b> This bit is write protected. Refer to the SYS_REGLCTL register.</p>
[9:8]	Reserved	Reserved.
[7]	PDEN	<p><b>System Power-down Enable (Write Protect)</b></p> <p>When this bit is set to 1, Power-down mode is enabled and chip keeps active till the CPU sleep mode is also active and then the chip enters Power-down mode.</p> <p>When chip wakes up from Power-down mode, this bit is auto cleared. Users need to set this bit again for next Power-down.</p> <p>In Power-down mode, HXT, HIRC and the HIRC48 will be disabled in this mode, but LXT and LIRC are not controlled by Power-down mode.</p>

		In Power-down mode, the PLL and system clock 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 waits CPU sleep command WFI and then enters Power-down mode. <b>Note:</b> This bit is write protected. Refer to the SYS_REGLCTL register.
[6]	PDWKIF	<b>Power-down Mode Wake-up Interrupt Status</b> Set by "Power-down wake-up event", it indicates that resume from Power-down mode" The flag is set if the EINT0~5, GPIO, USB, UART0~2, WDT, ACMP01, BOD, EBOD, RTC, TMR0~3, I <sup>2</sup> C0~1 or USCIO~2 wake-up occurred. <b>Note1:</b> This bit can be cleared by software writing '1'. <b>Note2:</b> This bit works only if PDWKIEN (CLK_PWRCTL[5]) set to 1.
[5]	PDWKIEN	<b>Power-down Mode Wake-up Interrupt Enable Bit (Write Protect)</b> 0 = Power-down mode wake-up interrupt Disabled. 1 = Power-down mode wake-up interrupt Enabled. <b>Note1:</b> The interrupt will occur when both PDWKIF and PDWKIEN are high. <b>Note2:</b> This bit is write protected. Refer to the SYS_REGLCTL register.
[4]	PDWKDLY	<b>Enable the Wake-up Delay Counter (Write Protect)</b> 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 work at 4~24 MHz external high speed crystal oscillator (HXT), 256 clock cycles when chip work at 22.1184 MHz internal high speed RC oscillator (HIRC) and 512 clock cycles when chip work at 48 MHz internal high speed RC oscillator (HIRC48). 0 = Clock cycles delay Disabled. 1 = Clock cycles delay Enabled. <b>Note:</b> This bit is write protected. Refer to the SYS_REGLCTL register.
[3]	LIRCEN	<b>LIRC Enable Bit (Write Protect)</b> 0 = 10 kHz internal low speed RC oscillator (LIRC) Disabled. 1 = 10 kHz internal low speed RC oscillator (LIRC) Enabled. <b>Note:</b> This bit is write protected. Refer to the SYS_REGLCTL register.
[2]	HIRCEN	<b>HIRC Enable Bit (Write Protect)</b> 0 = 22.1184 MHz internal high speed RC oscillator (HIRC) Disabled. 1 = 22.1184 MHz internal high speed RC oscillator (HIRC) Enabled. <b>Note:</b> This bit is write protected. Refer to the SYS_REGLCTL register.
[1]	LXTEN	<b>LXT Enable Bit (Write Protect)</b> 0 = 32.768 KHz External Low Speed Crystal (LXT) Disabled. 1 = 32.768 KHz External Low Speed Crystal (LXT) Enabled. <b>Note:</b> This bit is write protected. Refer to the SYS_REGLCTL register.
[0]	HXTEN	<b>HXT Enable Bit (Write Protect)</b> The bit default value is set by flash controller user configuration register CONFIG0 [26:24]. When the default clock source is from HXT, this bit is set to 1 automatically. 0 = 4~24 MHz External High Speed Crystal (HXT) Disabled. 1 = 4~24 MHz External High Speed Crystal (HXT) Enabled. <b>Note:</b> This bit is write protected. Refer to the SYS_REGLCTL register.

Register/Instruction Mode	SLEEPDEEP (SCR[2])	PDEN (CLK_PWRCTL)	CPU Run WFI Instruction	Clock Disable
---------------------------	--------------------	-------------------	-------------------------	---------------

		[7]		
<b>Normal operation</b>	0	0	NO	All clocks are disabled by control register.
<b>Idle mode</b> (CPU enter Sleep mode)	0	0	YES	Only CPU clock is disabled.
<b>Power-down mode</b> (CPU enters Deep Sleep mode)	1	1	YES	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 6.3-2 Power-down Mode Control Table

When the chip enters Power-down mode, user can wake up chip by some interrupt sources. User should enable the related interrupt sources and NVIC IRQ enable bits (NVIC\_ISER) before set PDEN bit in CLK\_PWRCTL[7] to ensure chip can enter Power-down and wake up successfully.

**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	0x003F_8004

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved		GPIOFCKEN	GPIOECKEN	GPIODCKEN	GPIOCCKEN	GPIOBCKEN	GPIOACKEN
15	14	13	12	11	10	9	8
FMCIDLE	Reserved						
7	6	5	4	3	2	1	0
CRCCCKEN	Reserved		HDIVCKEN	EBICKEN	ISPCKEN	PDMACKEN	Reserved

Bits	Description	
[31:22]	Reserved	Reserved.
[21]	GPIOFCKEN	<b>General Purpose I/O PF Group Clock Enable Bit</b> 0 = GPIO PF group clock Disabled. 1 = GPIO PF group clock Enabled.
[20]	GPIOECKEN	<b>General Purpose I/O PE Group Clock Enable Bit</b> 0 = GPIO PE group clock Disabled. 1 = GPIO PE group clock Enabled.
[19]	GPIODCKEN	<b>General Purpose I/O PD Group Clock Enable Bit</b> 0 = GPIO PD group clock Disabled. 1 = GPIO PD group clock Enabled.
[18]	GPIOCCKEN	<b>General Purpose I/O PC Group Clock Enable Bit</b> 0 = GPIO PC group clock Disabled. 1 = GPIO PC group clock Enabled.
[17]	GPIOBCKEN	<b>General Purpose I/O PB Group Clock Enable Bit</b> 0 = GPIO PB group clock Disabled. 1 = GPIO PB group clock Enabled.
[16]	GPIOACKEN	<b>General Purpose I/O PA Group Clock Enable Bit</b> 0 = GPIO PA group clock Disabled. 1 = GPIO PA group clock Enabled.
[15]	FMCIDLE	<b>Flash Memory Controller Clock Enable Bit in IDLE Mode</b> 0 = FMC peripheral clock Disabled when chip operating at IDLE mode. 1 = FMC peripheral clock Enabled when chip operating at IDLE mode.
[14:8]	Reserved	Reserved.



[7]	<b>CRCCKEN</b>	<b>CRC Generator Controller Clock Enable Bit</b> 0 = CRC peripheral clock Disabled. 1 = CRC peripheral clock Enabled.
[6:5]	<b>Reserved</b>	Reserved.
[4]	<b>HDIVCKEN</b>	<b>Hardware Divider Controller Clock Enable Bit</b> 0 = Hardware divider peripheral clock Disabled. 1 = Hardware divider peripheral clock Enabled.
[3]	<b>EBICKEN</b>	<b>EBI Controller Clock Enable Bit</b> 0 = EBI peripheral clock Disabled. 1 = EBI peripheral clock Enabled.
[2]	<b>ISPCKEN</b>	<b>Flash ISP Controller Clock Enable Bit</b> 0 = Flash ISP peripheral clock Disabled. 1 = Flash ISP peripheral clock Enabled.
[1]	<b>PDMACKEN</b>	<b>PDMA Controller Clock Enable Bit</b> 0 = PDMA peripheral clock Disabled. 1 = PDMA peripheral clock Enabled.
[0]	<b>Reserved</b>	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	ACMP01CKEN	Reserved	ADCCKEN	USBCKEN	Reserved		
23	22	21	20	19	18	17	16
Reserved		PWM1CKEN	PWM0CKEN	Reserved	UART2CKEN	UART1CKEN	UART0CKEN
15	14	13	12	11	10	9	8
Reserved		SPI1CKEN	SPI0CKEN	Reserved		I2C1CKEN	I2C0CKEN
7	6	5	4	3	2	1	0
Reserved	CLKCKEN	TMR3CKEN	TMR2CKEN	TMR1CKEN	TMR0CKEN	RTCKEN	WDTCKEN

Bits	Description	
[31]	Reserved	Reserved.
[30]	ACMP01CKEN	<b>Analog Comparator 0/1 Clock Enable Bit</b> 0 = Analog Comparator 0/1 clock Disabled. 1 = Analog Comparator 0/1 clock Enabled.
[29]	Reserved	Reserved.
[28]	ADCCKEN	<b>Analog-digital-converter (ADC) Clock Enable Bit</b> 0 = ADC clock Disabled. 1 = ADC clock Enabled.
[27]	USBCKEN	<b>USB Device Clock Enable Bit</b> 0 = USB Device clock Disabled. 1 = USB Device clock Enabled.
[26:22]	Reserved	Reserved.
[21]	PWM1CKEN	<b>PWM1 Clock Enable Bit</b> 0 = PWM1 clock Disabled. 1 = PWM1 clock Enabled.
[20]	PWM0CKEN	<b>PWM0 Clock Enable Bit</b> 0 = PWM0 clock Disabled. 1 = PWM0 clock Enabled.
[19]	Reserved	Reserved.
[18]	UART2CKEN	<b>UART2 Clock Enable Bit</b> 0 = UART2 clock Disabled. 1 = UART2 clock Enabled.
[17]	UART1CKEN	<b>UART1 Clock Enable Bit</b>

		0 = UART1 clock Disabled. 1 = UART1 clock Enabled.
[16]	<b>UART0CKEN</b>	<b>UART0 Clock Enable Bit</b> 0 = UART0 clock Disabled. 1 = UART0 clock Enabled.
[15:14]	<b>Reserved</b>	Reserved.
[13]	<b>SPI1CKEN</b>	<b>SPI1 Clock Enable Bit</b> 0 = SPI1 Clock Disabled. 1 = SPI1 Clock Enabled.
[12]	<b>SPI0CKEN</b>	<b>SPI0 Clock Enable Bit</b> 0 = SPI0 Clock Disabled. 1 = SPI0 Clock Enabled.
[11:10]	<b>Reserved</b>	Reserved.
[9]	<b>I2C1CKEN</b>	<b>I2C1 Clock Enable Bit</b> 0 = I2C1 Clock Disabled. 1 = I2C1 Clock Enabled.
[8]	<b>I2C0CKEN</b>	<b>I2C0 Clock Enable Bit</b> 0 = I2C0 Clock Disabled. 1 = I2C0 Clock Enabled.
[7]	<b>Reserved</b>	Reserved.
[6]	<b>CLKOCKEN</b>	<b>CLKO Clock Enable Bit</b> 0 = CLKO Clock Disabled. 1 = CLKO Clock Enabled.
[5]	<b>TMR3CKEN</b>	<b>Timer3 Clock Enable Bit</b> 0 = Timer3 Clock Disabled. 1 = Timer3 Clock Enabled.
[4]	<b>TMR2CKEN</b>	<b>Timer2 Clock Enable Bit</b> 0 = Timer2 Clock Disabled. 1 = Timer2 Clock Enabled.
[3]	<b>TMR1CKEN</b>	<b>Timer1 Clock Enable Bit</b> 0 = Timer1 Clock Disabled. 1 = Timer1 Clock Enabled.
[2]	<b>TMR0CKEN</b>	<b>Timer0 Clock Enable Bit</b> 0 = Timer0 Clock Disabled. 1 = Timer0 Clock Enabled.
[1]	<b>RTCCKEN</b>	<b>Real-time-clock APB Interface Clock Enable Bit</b> This bit is used to control the RTC APB clock only. The RTC peripheral clock source is selected from RTCSEL(CLK_CLKSEL2[18]). It can be selected to external 32.768 kHz low speed crystal (LXT) or 10 kHz internal low speed RC oscillator (LIRC). 0 = RTC Clock Disabled. 1 = RTC Clock Enabled.
[0]	<b>WDTCKEN</b>	<b>Watchdog Timer Clock Enable Bit (Write Protect)</b> 0 = Watchdog Timer Clock Disabled. 1 = Watchdog Timer Clock Enabled.

		<b>Note:</b> This bit is write protected. Refer to the SYS_REGLCTL register.
--	--	--

**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+0x30	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							
15	14	13	12	11	10	9	8
Reserved					USCI2CKEN	USCI1CKEN	USCI0CKEN
7	6	5	4	3	2	1	0
Reserved						SC1CKEN	SC0CKEN

Bits	Description	
[31:11]	Reserved	Reserved.
[10]	USCI2CKEN	<b>USCI2 Clock Enable Bit</b> 0 = USCI2 clock Disabled. 1 = USCI2 clock Enabled.
[9]	USCI1CKEN	<b>USCI1 Clock Enable Bit</b> 0 = USCI1 clock Disabled. 1 = USCI1 clock Enabled.
[8]	USCI0CKEN	<b>USCI0 Clock Enable Bit</b> 0 = USCI0 clock Disabled. 1 = USCI0 clock Enabled.
[7:2]	Reserved	Reserved.
[1]	SC1CKEN	<b>SC1 Clock Enable Bit</b> 0 = SC1 clock Disabled. 1 = SC1 clock Enabled.
[0]	SC0CKEN	<b>SC0 Clock Enable Bit</b> 0 = SC0 Clock Disabled. 1 = SC0 Clock Enabled.

**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_003X

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
PCLK1SEL	PCLK0SEL	STCLKSEL			HCLKSEL		

Bits	Description	
[31:8]	Reserved	Reserved.
[7]	PCLK1SEL	<p><b>PCLK1 Clock Source Selection (Write Protect)</b></p> <p>0 = APB1 BUS clock source from HCLK.                      1 = APB1 BUS clock source from HCLK/2.</p> <p><b>Note:</b> This bit is write protected. Refer to the SYS_REGLCTL register.</p>
[6]	PCLK0SEL	<p><b>PCLK0 Clock Source Selection (Write Protect)</b></p> <p>0 = APB0 BUS clock source from HCLK.                      1 = APB0 BUS clock source from HCLK/2.</p> <p><b>Note:</b> This bit is write protected. Refer to the SYS_REGLCTL register.</p>
[5:3]	STCLKSEL	<p><b>Cortex<sup>®</sup>-M0 SysTick Clock Source Selection (Write Protect)</b></p> <p>If SYST_CTRL[2]=0, SysTick uses listed clock source below.</p> <p>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.</p> <p><b>Note1:</b> if SysTick clock source is not from HCLK (i.e. SYST_CTRL[2] = 0), SysTick clock source must less than or equal to HCLK/2.</p> <p><b>Note2:</b> These bits are write protected. Refer to the SYS_REGLCTL register.</p>
[2:0]	HCLKSEL	<p><b>HCLK Clock Source Selection (Write Protect)</b></p> <p>Before clock switching, the related clock sources (both pre-select and new-select) must be turned on.</p> <p>The default value is reloaded from the value of CFOSC (CONFIG0[26:24]) in user configuration register of Flash controller by any reset. Therefore the default value is either 000b or 111b.</p> <p>000 = Clock source from HXT.                      001 = Clock source from LXT.                      010 = Clock source from PLL clock.</p>

		<p>011 = Clock source from LIRC.          100 = Clock source from HIRC48.          111= Clock source from HIRC clock.          Other = Reserved.  <b>Note:</b> These bits are write protected. Refer to the SYS_REGLCTL register.</p>
--	--	---

**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	0x3377_770F

31	30	29	28	27	26	25	24
Reserved		PWM1SEL	PWM0SEL	Reserved		UARTSEL	
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				ADCSEL		WDTSEL	

Bits	Description	
[31:30]	Reserved	Reserved.
[29]	PWM1SEL	<b>PWM1 Clock Source Selection</b> The peripheral clock source of PWM1 is defined by PWM1SEL. 0 = Clock source from PLL clock. 1 = Clock source from PCLK1.
[28]	PWM0SEL	<b>PWM0 Clock Source Selection</b> The peripheral clock source of PWM0 is defined by PWM0SEL. 0 = Clock source from PLL clock. 1 = Clock source from PCLK0.
[27:26]	Reserved	Reserved.
[25:24]	UARTSEL	<b>UART Clock Source Selection</b> 00 = Clock source from 4~24 MHz external high speed crystal oscillator (HXT) clock. 01 = Clock source from PLL clock. 10 = Clock source from 32.768 kHz external low speed crystal oscillator (LXT) clock. 11 = Clock source from 22.1184 MHz internal high speed RC oscillator (HIRC) clock.
[23]	Reserved	Reserved.
[22:20]	TMR3SEL	<b>TIMER3 Clock Source Selection</b> 000 = Clock source from 4~24 MHz external high speed crystal oscillator (HXT) clock. 001 = Clock source from 32.768 kHz external low speed crystal oscillator (LXT) clock. 010 = Clock source from PCLK1. 011 = Clock source from external clock T3 pin. 101 = Clock source from 10 kHz internal low speed RC oscillator (LIRC) clock. 111 = Clock source from 22.1184 MHz internal high speed RC oscillator (HIRC) clock. Others = Reserved.



[19]	Reserved	Reserved.
[18:16]	TMR2SEL	<p><b>TIMER2 Clock Source Selection</b></p> <p>000 = Clock source from 4~24 MHz external high speed crystal oscillator (HXT) clock.          001 = Clock source from 32.768 kHz external low speed crystal oscillator (LXT) clock.          010 = Clock source from PCLK1.          011 = Clock source from external clock T2 pin.          101 = Clock source from 10 kHz internal low speed RC oscillator (LIRC) clock.          111 = Clock source from 22.1184 MHz internal high speed RC oscillator (HIRC) clock.          Others = Reserved.</p>
[15]	Reserved	Reserved.
[14:12]	TMR1SEL	<p><b>TIMER1 Clock Source Selection</b></p> <p>000 = Clock source from 4~24 MHz external high speed crystal oscillator (HXT) clock.          001 = Clock source from 32.768 kHz external low speed crystal oscillator (LXT) clock.          010 = Clock source from PCLK0.          011 = Clock source from external clock T1 pin.          101 = Clock source from 10 kHz internal low speed RC oscillator (LIRC) clock.          111 = Clock source from 22.1184 MHz internal high speed RC oscillator (HIRC) clock.          Others = Reserved.</p>
[11]	Reserved	Reserved.
[10:8]	TMR0SEL	<p><b>TIMER0 Clock Source Selection</b></p> <p>000 = Clock source from 4~24 MHz external high speed crystal oscillator (HXT) clock.          001 = Clock source from 32.768 kHz external low speed crystal oscillator (LXT) clock.          010 = Clock source from PCLK0.          011 = Clock source from external clock T0 pin.          101 = Clock source from 10 kHz internal low speed RC oscillator (LIRC) clock.          111 = Clock source from 22.1184 MHz internal high speed RC oscillator (HIRC) clock.          Others = Reserved.</p>
[7:4]	Reserved	Reserved.
[3:2]	ADCSEL	<p><b>ADC Clock Source Selection</b></p> <p>00 = Clock source from 4~24 MHz external high speed crystal oscillator (HXT) clock.          01 = Clock source from PLL.          10 = Clock source from PCLK0.          11 = Clock source from 22.1184 MHz internal high speed RC oscillator (HIRC) clock.</p>
[1:0]	WDTSEL	<p><b>Watchdog Timer Clock Source Selection (Write Protect)</b></p> <p>00 = Reserved.          01 = Clock source from 32.768 kHz external low speed crystal oscillator (LXT) clock.          10 = Clock source from HCLK/2048 clock.          11 = Clock source from 10 kHz internal low speed RC oscillator (LIRC) clock.  <b>Note:</b> These bits are write protected. Refer to the SYS_REGLCTL register.</p>

**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+0x1C	R/W	Clock Source Select Control Register 2	0x0002_0008

31	30	29	28	27	26	25	24
Reserved				SPI1SEL		SPI0SEL	
23	22	21	20	19	18	17	16
Reserved					RTCSEL	WWDTSEL	
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved			CLKOSEL			Reserved	

Bits	Description	
[31:28]	Reserved	Reserved.
[27:26]	SPI1SEL	<b>SPI1 Clock Source Selection</b> 00 = Clock source from 4~24 MHz external high speed crystal oscillator (HXT) clock. 01 = Clock source from PLL clock. 10 = Clock source from PCLK0. 11 = Clock source from 48 MHz internal high speed RC oscillator (HIRC48) clock.
[25:24]	SPI0SEL	<b>SPI0 Clock Source Selection</b> 00 = Clock source from 4~24 MHz external high speed crystal oscillator (HXT) clock. 01 = Clock source from PLL clock. 10 = Clock source from PCLK0. 11 = Clock source from 48 MHz internal high speed RC oscillator (HIRC48) clock.
[23:19]	Reserved	Reserved.
[18]	RTCSEL	<b>RTC Clock Source Selection</b> 0 = Clock source from 32.768 kHz external low speed crystal oscillator (LXT). 1 = Clock source from 10 kHz internal low speed RC oscillator (LIRC).
[17:16]	WWDTSEL	<b>Window Watchdog Timer Clock Source Selection</b> 10 = Clock source from HCLK/2048 clock. 11 = Clock source from 10 kHz internal low speed RC oscillator (LIRC). Others = Reserved.
[15:5]	Reserved	Reserved.
[4:2]	CLKOSEL	<b>Clock Divider Clock Source Selection</b> 000 = Clock source from 4~24 MHz external high speed crystal oscillator (HXT) clock. 001 = Clock source from 32.768 kHz external low speed crystal oscillator (LXT) clock. 010 = Clock source from HCLK.

		011 = Clock source from 22.1184 MHz internal high speed RC oscillator (HIRC) clock. 101 = Clock source from 48 MHz internal high speed RC oscillator (HIRC48) clock. Others = Reserved.
[1:0]	<b>Reserved</b>	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+0x34	R/W	Clock Source Select Control Register 3	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							USBSEL
7	6	5	4	3	2	1	0
Reserved				SC1SEL		SC0SEL	

Bits	Description	
[31:9]	Reserved	Reserved.
[8]	USBSEL	<b>USB Clock Source Selection(Write Protect)</b> 0 = Clock source from 48MHz internal high speed RC oscillator (HIRC48) clock. 1 = Clock source from PLL clock. <b>Note:</b> This bit is write protected. Refer to the SYS_REGLCTL register.
[7:4]	Reserved	Reserved.
[3:2]	SC1SEL	<b>SC1 Clock Source Selection</b> 00 = Clock source from 4~24 MHz external high speed crystal oscillator (HXT) clock. 01 = Clock source from PLL clock. 10 = Clock source from PCLK1. 11 = Clock source from 22.1184 MHz internal high speed RC oscillator (HIRC) clock.
[1:0]	SC0SEL	<b>SC0 Clock Source Selection</b> 00 = Clock source from 4~24 MHz external high speed crystal oscillator (HXT) clock. 01 = Clock source from PLL clock. 10 = Clock source from PCLK1. 11 = Clock source from 22.1184 MHz internal high speed RC oscillator (HIRC) clock.

**Clock Divider Number Register 0 (CLK\_CLKDIV0)**

Register	Offset	R/W	Description	Reset Value
CLK_CLKDIV0	CLK_BA+0x18	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
ADCDIV							
15	14	13	12	11	10	9	8
Reserved				UARTDIV			
7	6	5	4	3	2	1	0
USBDIV				HCLKDIV			

Bits	Description	
[31:24]	Reserved	Reserved.
[23:16]	ADCDIV	<b>ADC Clock Divide Number From ADC Clock Source</b> ADC clock frequency = (ADC clock source frequency) / (ADCDIV + 1).
[15:12]	Reserved	Reserved.
[11:8]	UARTDIV	<b>UART Clock Divide Number From UART Clock Source</b> UART clock frequency = (UART clock source frequency) / (UARTDIV + 1).
[7:4]	USBDIV	<b>USB Clock Divide Number From PLL Source</b> USB clock frequency = (PLL frequency) / (USBDIV + 1). <b>Note:</b> If the HIRC48 is selected, it is delivery to USB clock directly.
[3:0]	HCLKDIV	<b>HCLK Clock Divide Number From HCLK Clock Source</b> HCLK clock frequency = (HCLK clock source frequency) / (HCLKDIV + 1).

**Clock Divider Number Register 1 (CLK\_CLKDIV1)**

Register	Offset	R/W	Description	Reset Value
CLK_CLKDIV1	CLK_BA+0x38	R/W	Clock Divider Number 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
SC1DIV							
7	6	5	4	3	2	1	0
SC0DIV							

Bits	Description	
[31:16]	Reserved	Reserved.
[15:8]	SC1DIV	<b>SC1 Clock Divide Number From SC1 Clock Source</b> SC1 clock frequency = (SC1 clock source frequency) / (SC1DIV + 1).
[7:0]	SC0DIV	<b>SC0 Clock Divide Number From SC0 Clock Source</b> SC0 clock frequency = (SC0 clock source frequency) / (SC0DIV + 1).

**PLL Control Register (CLK\_PLLCTL)**

The PLL reference clock input is from the 4~24 MHz external high speed crystal oscillator (HXT) clock input or from the 22.1184 MHz internal high speed RC oscillator (HIRC). This register is used to control the PLL output frequency and PLL operation mode.

Register	Offset	R/W	Description	Reset Value
CLK_PLLCTL	CLK_BA+0x20	R/W	PLL Control Register	0x008D_8418

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

Bits	Description	
[31:24]	Reserved	Reserved.
[23]	STBSEL	<b>PLL Stable Counter Selection</b> 0 = PLL stable time is 6144 PLL source clock (suitable for source clock is equal to or less than 12MHz). 1 = PLL stable time is 12288 PLL source clock (suitable for source clock is larger than 12MHz).
[22:20]	Reserved	Reserved.
[19]	PLLSRC	<b>PLL Source Clock Selection</b> 0 = PLL source clock from external 4~24 MHz high-speed crystal (HXT). 1 = PLL source clock from internal 22.1184 MHz high-speed oscillator (HIRC).
[18]	OE	<b>PLL OE (FOUT Enable) Control</b> 0 = PLL FOUT Enabled. 1 = PLL FOUT is fixed low.
[17]	BP	<b>PLL Bypass Control</b> 0 = PLL is in normal mode (default). 1 = PLL clock output is same as PLL input clock FIN.
[16]	PD	<b>Power-down Mode</b> If set PDEN(CLK_PWRCTL[7]) bit to 1, the PLL will enter Power-down mode, too. 0 = PLL is in normal mode. 1 = PLL is in Power-down mode (default).
[15:14]	OUTDIV	<b>PLL Output Divider Control</b> Refer to the formulas below the table.
[13:9]	INDIV	<b>PLL Input Divider Control</b>

		Refer to the formulas below the table.
[8:0]	FBDIV	<b>PLL Feedback Divider Control</b> Refer to the formulas below the table.

**Output Clock Frequency Setting**

$$F_{OUT} = F_{IN} \times \frac{NF}{NR} \times \frac{1}{NO}$$

Constraint:

1.  $4MHz < F_{IN} < 24MHz$
2.  $800KHz < \frac{F_{IN}}{2 * NR} < 8MHz$
3.  $100MHz < F_{CO} = F_{IN} * \frac{NF}{NR} < 200MHz,$   
 $F_{CO} > 120MHz$  is preferred

Symbol	Description
FOUT	Output Clock Frequency
FIN	Input (Reference) Clock Frequency
NR	Input Divider (INDIV + 2)
NF	Feedback Divider (FBDIV + 2)
NO	OUTDIV = "00" : NO = 1 OUTDIV = "01" : NO = 2 OUTDIV = "10" : NO = 2 OUTDIV = "11" : NO = 4



**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+0x0C	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
CLKSFAIL	Reserved	HIRC48STB	HIRCSTB	LIRCSTB	PLLSTB	LXTSTB	HXTSTB

Bits	Description
[31:8]	<b>Reserved</b> Reserved.
[7]	<b>CLKSFAIL</b> <b>Clock Switching Fail Flag (Read Only)</b> 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. <b>Note:</b> After selected clock source is stable, hardware will switch system clock to selected clock automatically, and CLKSFAIL will be cleared automatically by hardware.
[6]	<b>Reserved</b> Reserved.
[5]	<b>HIRC48STB</b> <b>HIRC48 Clock Source Stable Flag (Read Only)</b> 0 = 48 MHz internal high speed RC oscillator (HIRC48) clock is not stable or disabled. 1 = 48 MHz internal high speed RC oscillator (HIRC48) clock is stable and enabled.
[4]	<b>HIRCSTB</b> <b>HIRC Clock Source Stable Flag (Read Only)</b> 0 = 22.1184 MHz internal high speed RC oscillator (HIRC) clock is not stable or disabled. 1 = 22.1184 MHz internal high speed RC oscillator (HIRC) clock is stable and enabled.
[3]	<b>LIRCSTB</b> <b>LIRC Clock Source Stable Flag (Read Only)</b> 0 = 10 kHz internal low speed RC oscillator (LIRC) clock is not stable or disabled. 1 = 10 kHz internal low speed RC oscillator (LIRC) clock is stable and enabled.
[2]	<b>PLLSTB</b> <b>Internal PLL Clock Source Stable Flag (Read Only)</b> 0 = Internal PLL clock is not stable or disabled. 1 = Internal PLL clock is stable and enabled.
[1]	<b>LXTSTB</b> <b>LXT Clock Source Stable Flag (Read Only)</b> 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]	<b>HXTSTB</b>	<b>HXT Clock Source Stable Flag (Read Only)</b> 0 = 4~24 MHz external high speed crystal oscillator (HXT) clock is not stable or disabled. 1 = 4~24 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+0x24	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	CLKOEN	FREQSEL				

Bits	Description
[31:7]	<b>Reserved</b> Reserved.
[6]	<b>CLK1HZEN</b> <b>Clock Output 1Hz Enable Bit</b> 0 = 1 Hz clock output for 32.768 kHz external low speed crystal oscillator (LXT) frequency compensation Disabled. 1 = 1 Hz clock output for 32.768 kHz external low speed crystal oscillator (LXT) frequency compensation Enabled.
[5]	<b>DIV1EN</b> <b>Clock Output Divide One Enable Bit</b> 0 = Clock Output will output clock with source frequency divided by FREQSEL. 1 = Clock Output will output clock with source frequency.
[4]	<b>CLKOEN</b> <b>Clock Output Enable Bit</b> 0 = Clock Output function Disabled. 1 = Clock Output function Enabled.
[3:0]	<b>FREQSEL</b> <b>Clock Output Frequency Selection</b> 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 Source Select for BOD Control Register (CLK\_BODCLK)**

Register	Offset	R/W	Description	Reset Value
CLK_BODCLK	CLK_BA+0x40	R/W	Clock Source Select for BOD 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							VDETCKSEL

Bits	Description
[31:1]	Reserved Reserved.
[0]	<p><b>VDETCKSEL</b></p> <p><b>Clock Source Selection for Voltage Detector</b> The Voltage Detector clock source for detecting external input voltage is defined by VDETCKSEL.</p> <p>0 = Clock source is from 10 kHz internal low speed RC oscillator (LIRC) clock. 1 = Clock source is from 32.768 kHz external low speed crystal oscillator (LXT) clock.</p> <p><b>Note1:</b> If LIRC is selected, LIRCEN (CLK_PWRCTL[3]) must be enabled. <b>Note2:</b> If LXT is selected, LXTEN (CLK_PWRCTL[1]) must be enabled.</p>

**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	<b>HXT Clock Frequency Monitor Interrupt Enable Bit</b> 0 = 4~24 MHz external high speed crystal oscillator (HXT) clock frequency monitor fail interrupt Disabled. 1 = 4~24 MHz external high speed crystal oscillator (HXT) clock frequency monitor fail interrupt Enabled.
[16]	HXTFQDEN	<b>HXT Clock Frequency Monitor Enable Bit</b> 0 = 4~24 MHz external high speed crystal oscillator (HXT) clock frequency monitor Disabled. 1 = 4~24 MHz external high speed crystal oscillator (HXT) clock frequency monitor Enabled.
[15:14]	Reserved	Reserved.
[13]	LXTFIEN	<b>LXT Clock Fail Interrupt Enable Bit</b> 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	<b>LXT Clock Fail Detector Enable Bit</b> 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	<b>HXT Clock Fail Interrupt Enable Bit</b> 0 = 4~24 MHz external high speed crystal oscillator (HXT)clock Fail interrupt Disabled. 1 = 4~24 MHz external high speed crystal oscillator (HXT)clock Fail interrupt Enabled.
[4]	HXTFDEN	<b>HXT Clock Fail Detector Enable Bit</b> 0 = 4~24 MHz external high speed crystal oscillator (HXT) clock Fail detector Disabled. 1 = 4~24 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]	<b>Reserved</b> Reserved.
[8]	<b>HXTFQIF</b> <b>HXT Clock Frequency Monitor Interrupt Flag (Write Protect)</b> 0 = 4–24 MHz external high speed crystal oscillator (HXT) clock normal. 1 = 4–24 MHz external high speed crystal oscillator (HXT) clock frequency abnormal. <b>Note1:</b> This bit can be cleared to 0 by software writing '1'. <b>Note2:</b> This bit is write protected. Refer to the SYS_REGLCTL register.
[7:2]	<b>Reserved</b> Reserved.
[1]	<b>LXTFIF</b> <b>LXT Clock Fail Interrupt Flag (Write Protect)</b> 0 = 32.768 kHz external low speed crystal oscillator (LXT) clock normal. 1 = 32.768 kHz external low speed crystal oscillator (LXT) stop. <b>Note1:</b> This bit can be cleared to 0 by software writing '1'. <b>Note2:</b> This bit is write protected. Refer to the SYS_REGLCTL register.
[0]	<b>HXTFIF</b> <b>HXT Clock Fail Interrupt Flag (Write Protect)</b> 0 = 4–24 MHz external high speed crystal oscillator (HXT) clock normal. 1 = 4–24 MHz external high speed crystal oscillator (HXT) clock stop. <b>Note1:</b> This bit can be cleared to 0 by software writing '1'. <b>Note2:</b> This bit is write protected. Refer to the SYS_REGLCTL register.

**Clock Frequency Detector Upper Boundary Register (CLK\_CDUPB)**

Register	Offset	R/W	Description	Reset Value
CLK_CDUPB	CLK_BA+0x78	R/W	Clock Frequency 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]	UPERBD	<p><b>HXT Clock Frequency Detector Upper Boundary</b></p> <p>The bits define the high value of frequency monitor window.</p> <p>When HXT frequency monitor value higher than this register, the HXT frequency detect fail interrupt flag will set to 1.</p>

**Clock Frequency Detector Low Boundary Register (CLK\_CDLOWB)**

Register	Offset	R/W	Description	Reset Value
CLK_CDLOWB	CLK_BA+0x7c	R/W	Clock Frequency Detector Low 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[9:8]	
7	6	5	4	3	2	1	0
LOWERBD							

Bits	Description	
[31:10]	Reserved	Reserved.
[9:0]	LOWERBD	<p><b>HXT Clock Frequency Detector Low Boundary</b></p> <p>The bits define the low value of frequency monitor window.</p> <p>When HXT frequency monitor value lower than this register, the HXT frequency detect fail interrupt flag will set to 1.</p>



## 6.4 Flash Memory Controller (FMC)

### 6.4.1 Overview

The NUC126 series is equipped with 128/256 Kbytes on-chip embedded flash for application and configurable Data Flash to store some application dependent data. A User Configuration block provides for system initiation. A 4 Kbytes loader ROM (LDROM) is used for In-System-Programming (ISP) function. A 2 Kbytes security protection ROM (SPROM) can conceal user program. A 4KB 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 the chip reset after the embedded flash updated.

### 6.4.2 Features

- Supports 128/256 Kbytes application ROM (APROM).
- Supports 4 Kbytes loader ROM (LDROM).
- Supports 2 Kbytes security protection ROM (SPROM) to conceal user program.
- Supports Data Flash with configurable memory size.
- Supports 12 bytes User Configuration block to control system initiation.
- Supports 2 Kbytes page erase for all embedded flash.
- Supports 32-bit/64-bit and multi-word flash programming function.
- Supports CRC-32 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 as follows.

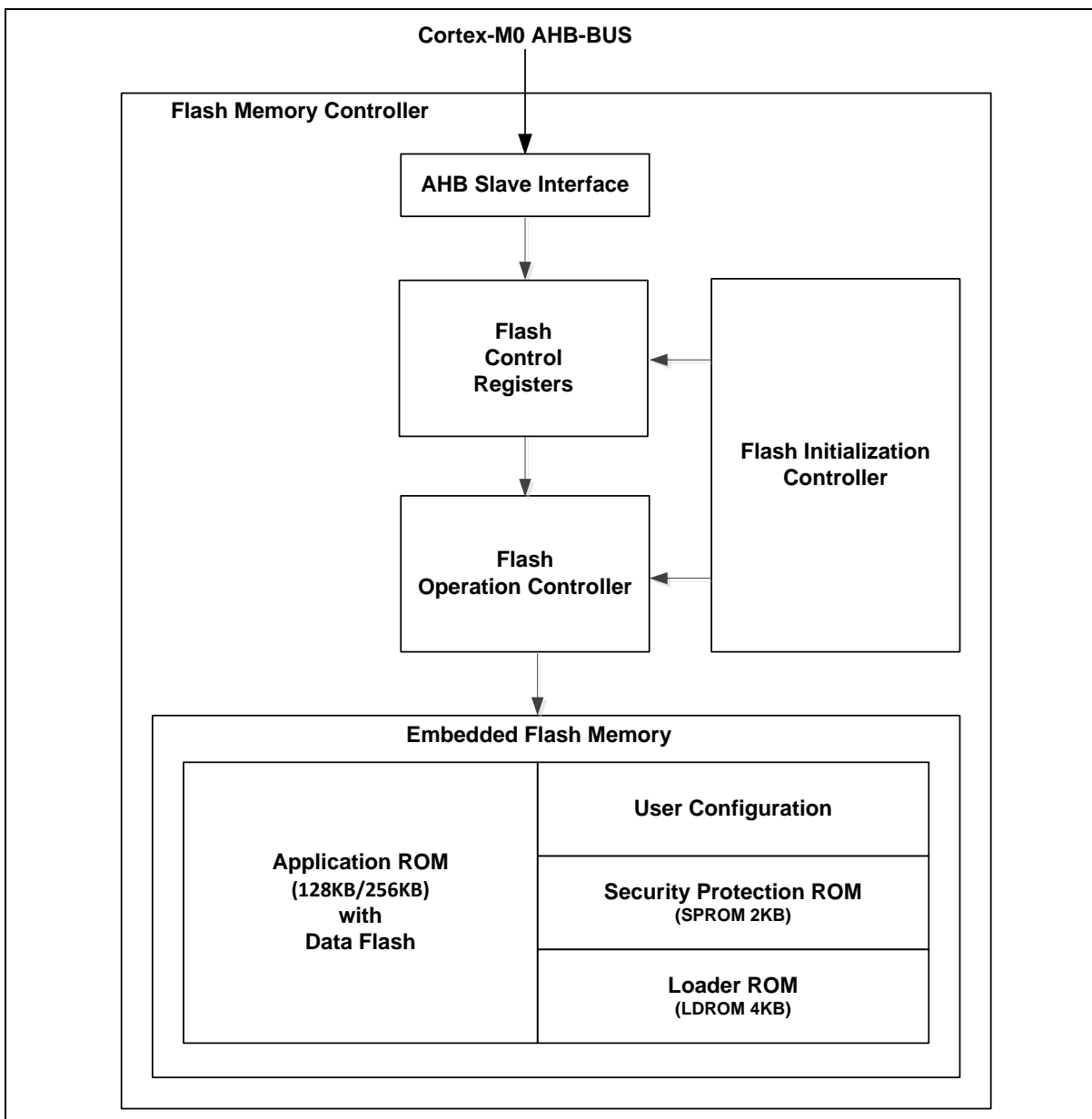


Figure 6.4-1 Flash Memory Controller Block Diagram

#### AHB Slave Interface

There is one AHB slave interfaces in flash memory controller, the bus is from Cortex<sup>®</sup>-M0 AHB-Lite Bus for the instruction and data fetch and flash control registers access including ISP registers.

#### Flash Control Registers

All of ISP control and status registers are in the flash control registers. The detail 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 for flash memory according to access requests.

#### **Embedded Flash Memory**

The embedded flash memory is the main memory for user application code and parameters. It consists of the user configuration block, 4KB LDROM, 2KB SPROM and 128KB/256KB APROM with Data Flash. The page erase flash size is 2KB, and 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, SPROM, Data Flash 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.

#### *LDROM, APROM and Data Flash*

LDROM is designed for a loader to implement In-System-Programming (ISP) function by user. LDROM is a 4KB embedded flash memory, the flash address range is from 0x0010\_0000 to 0x0010\_0FFF. APROM is main memory for user applications. APROM size is 128KB or 256KB. Data Flash is used to store application parameters (not instruction). Data Flash is shared with APROM for 128KB device and size is configurable. The base address of Data Flash is determined by DFBA (CONFIG1[19:0]). Figure 6.4-2 shows the memory map of Data Flash which is share the memory space of APROM. The page size of all embedded flash memory is 2KB.

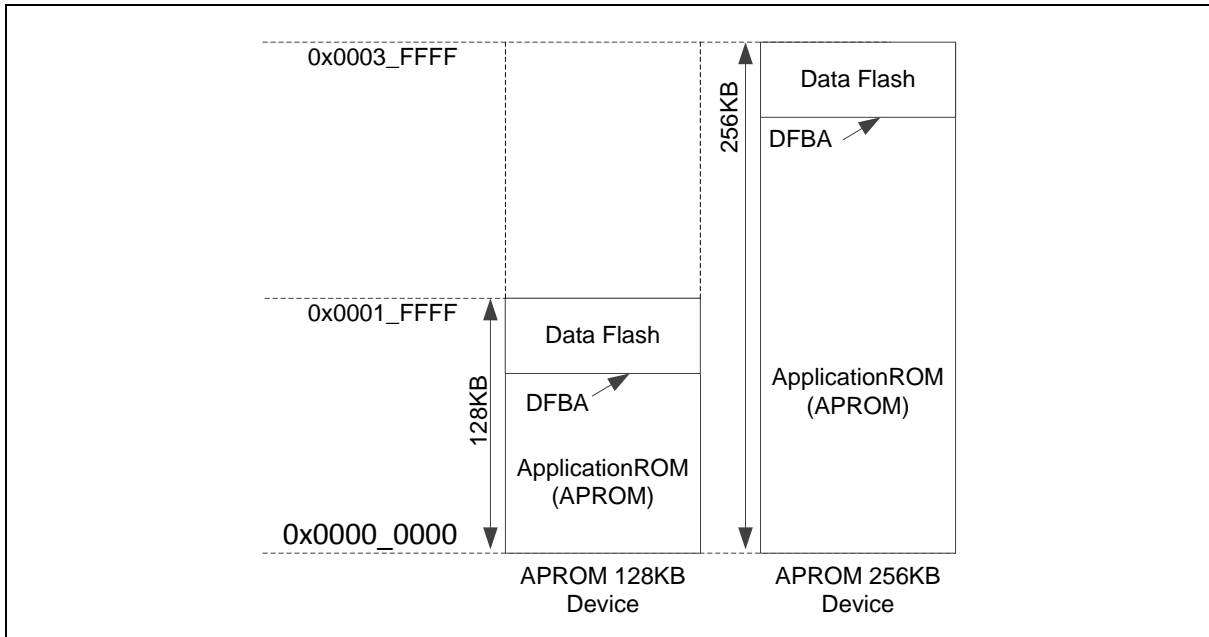


Figure 6.4-2 Data Flash

*User Configuration Block*

User Configuration block is internal programmable configuration area for boot options, such as flash security lock, boot select, brown-out voltage level, and Data Flash base address. 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 requirements. User Configuration block can be updated by ISP function and located at 0x0030\_0000 with three 32 bits words (CONFIG0 and CONFIG1). Any change on User Configuration block only take effect after system reboot

**CONFIG0 (Address = 0x0030\_0000)**

31	30	29	28	27	26	25	24
CWDTEN[2]	CWDTPDEN	Reserved		CFGXT1	CFOSC	Reserved	
23	22	21	20	19	18	17	16
CBODEN	CBOV		CBORST	Reserved			
15	14	13	12	11	10	9	8
Reserved			ICELOCK	Reserved	CIOINI	Reserved	
7	6	5	4	3	2	1	0
CBS		MERASE	CWDTEN[1:0]		Reserved	LOCK	DFEN

Bits	Descriptions	
[31]	CWDTEN[2]	<p><b>Watchdog Timer Hardware Enable Bit</b></p> <p>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, if in Power-down mode, the LIRC may be able to be disabled by setting CWDTPDEN=1 and LIRCEN=0 (CLK_PWRCTL[3]).</p> <p><b>CWDTEN[2:0]</b> is CONFIG0[31][4][3].</p> <p>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.</p> <p>111 = WDT hardware enable function is inactive.</p> <p>Others = WDT hardware enable function is active. WDT clock is always on.</p>
[30]	CWDTPDEN	<p><b>Watchdog Clock Power-down Enable Bit</b></p> <p>This bit should be used with CWDTEN. When WDT enabled by CWDTEN, User can use this bit to control WDT wakeup when system is in Power-down mode. If it is necessary to wakeup system by WDT, then we can set CWDTPDEN=0 to make sure WDT keep working in Power-down mode. If we don't want to wakeup system by WDT, we may just set CWDTPDEN=1 and LIRCEN=0 to let WDT suspend in power down.</p> <p>0 = Watchdog Timer clock kept enabled when chip enters Power-down.</p> <p>1 = Watchdog Timer clock is controlled by LIRCEN (CLK_PWRCTL[3]) when chip enters Power-down.</p> <p><b>Note:</b> This bit only works if CWDTEN[2:0] is set to 011</p>
[29]	Reserved	Reserved.

[27]	CFGXT1	<p><b>PF[4:3] Multi-Function Select</b></p> <p>If user don't need HXT in his application, he can use CFGXT1 to change PF[4:3] power on default to GPIO to avoid the effect of crystal oscillator circuite.</p> <p>0 = PF[4:3] pins are configured as GPIO pins.</p> <p>1 = PF[4:3] pins are configured as external 4~24 MHz external high speed crystal oscillator (HXT) pins.</p>
[26]	CFOSC	<p><b>CPU Clock Source Selection After Reset</b></p> <p>The value of CFOSC will be loaded to HCLK (CLK_CLKSEL0[2:0]) in system clock controller after any reset occurs. HCLK[2:0] = 111 (HIRC) if CFOSC = 1, HCLK[2:0] = 000 (HXT) if CFGSC=0.</p> <p>0 = 4~24 MHz external high speed crystal oscillator (HXT)</p> <p>1 = 22.1184 MHz internal high speed RC oscillator (HIRC)</p>
[25:24]	Reserved	Reserved.
[23]	CBODEN	<p><b>Brown-Out Detector Enable Bit</b></p> <p>0= Brown-out detect Enabled after powered on.</p> <p>1= Brown-out detect Disabled after powered on.</p>
[22:21]	CBOV	<p><b>Brown-Out Voltage Selection</b></p> <p>00 = Brown-out voltage is 2.2V.</p> <p>01 = Brown-out voltage is 2.7V.</p> <p>10 = Brown-out voltage is 3.7V.</p> <p>11 = Brown-out voltage is 4.5V.</p>
[20]	CBORST	<p><b>Brown-Out Reset Enable Bit</b></p> <p>0 = Brown-out reset Enabled after powered on.</p> <p>1 = Brown-out reset Disabled after powered on.</p>
[19]	Reserved	Reserved.
[14:13]	Reserved	Reserved.
[12]	ICELOCK	<p><b>ICE Lock Bit</b></p> <p>This bit only used to disable ICE function. User may use it with LOCK (CONFIG0[1]) bit to increase security level.</p> <p>0 = Disable ICE function.</p> <p>1 = Enable ICE function.</p>
[11]	Reserved	Reserved.
[10]	CIOINI	<p><b>I/O Initial State Selection</b></p> <p>0 = All GPIO set as input tri-state mode after powered on.</p> <p>1 = All GPIO set as Quasi-bidirectional mode after chip powered on.</p>
[9:8]	Reserved	Reserved.

[7:6]	<b>CBS</b>	<p><b>Chip Booting Selection</b></p> <p>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.</p> <p>CBS value is valid when MBS =1.</p> <p>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.</p> <p><b>Note:</b>          BS (FMC_ISPCTL[ 1]) only works on CBS[0] = 1.VECMAP function only works on CBS[0] = 0.</p>
[5]	<b>MERASE</b>	<p><b>ISP Flash Mass Erase Enable Bit</b></p> <p>This bit is used to control mess erase function.</p> <p>0 = ISP Flash Mass Erase command Enabled.          1 = ISP Flash Mass Erase command Disabled.</p>
[4:3]	<b>CWDTEN[1:0]</b>	<p><b>Watchdog Timer Hardware Enable Bit</b></p> <p>Please refer to CWDTEN[2] (CONFIG0[31]) for detail descriptions.</p>
[2]	<b>Reserved</b>	Reserved.
[1]	<b>LOCK</b>	<p><b>Security Lock Control</b></p> <p>This bit is used to enable security lock function. If security lock function enabled, only the software in APROM, LDROM or SPROM could read correct data. Any other way, e.g. ICE, will get 0xffffffff when read APROM, LDROM or SPROM.</p> <p>0 = Flash memory content is locked.          1 = Flash memory content is not locked.</p>
[0]	<b>DFEN</b>	<p><b>Data Flash Enable Bit</b></p> <p>The Data Flash is shared with APROM, and the base address of Data Flash is decided by DFBA (CONFIG1[19:0]) when DFEN is 0.</p> <p>0 = Data Flash Enabled.          1 = Data Flash Disabled.</p>

**CONFIG1 (Address = 0x0030\_0004)**

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

Bits	Descriptions	
[31:20]	<b>Reserved</b>	Reserved.
[19:0]	<b>DFBA</b>	<b>Data Flash Base Address</b> This register works only when DFEN (CONFIG0[0]) set to 0. If DFEN (CONFIG0[0]) is set to 0, the Data Flash base address is defined by user. Since on-chip flash erase unit is 2 Kbytes, it is mandatory to keep bit 10-0 as 0.



*Security Protection Memory (SPROM)*

The security protection memory (SPROM) is a special flash memory area for security applications. It supports independent lock mechanism which is different to security function of LOCK (CONFIG0[1]). In other words, user can lock SPROM without lock whole flash memory. This make it is possible for user to only protect their security key or key function in SPROM without lock all flash memory. Therefore, end customer still can develop or modify the application in APROM or LDROM when SPROM is locked.

The SPROM size is 2 Kbytes and its base address is 0x20\_0000. SPROM is only erased by “page erase command” with a key data, 0x0055AA03, in ISPDAT. The mass erase command cannot be used to erase SPROM. The last byte of SPROM (address: 0x0020\_07FF) is used to control the SPROM code is secured or non-secured as show in Figure 6.4-3. SPROM supports three mode, they are non-security mode, debug mode and security mode. All of them are controlled by last byte of SPROM.

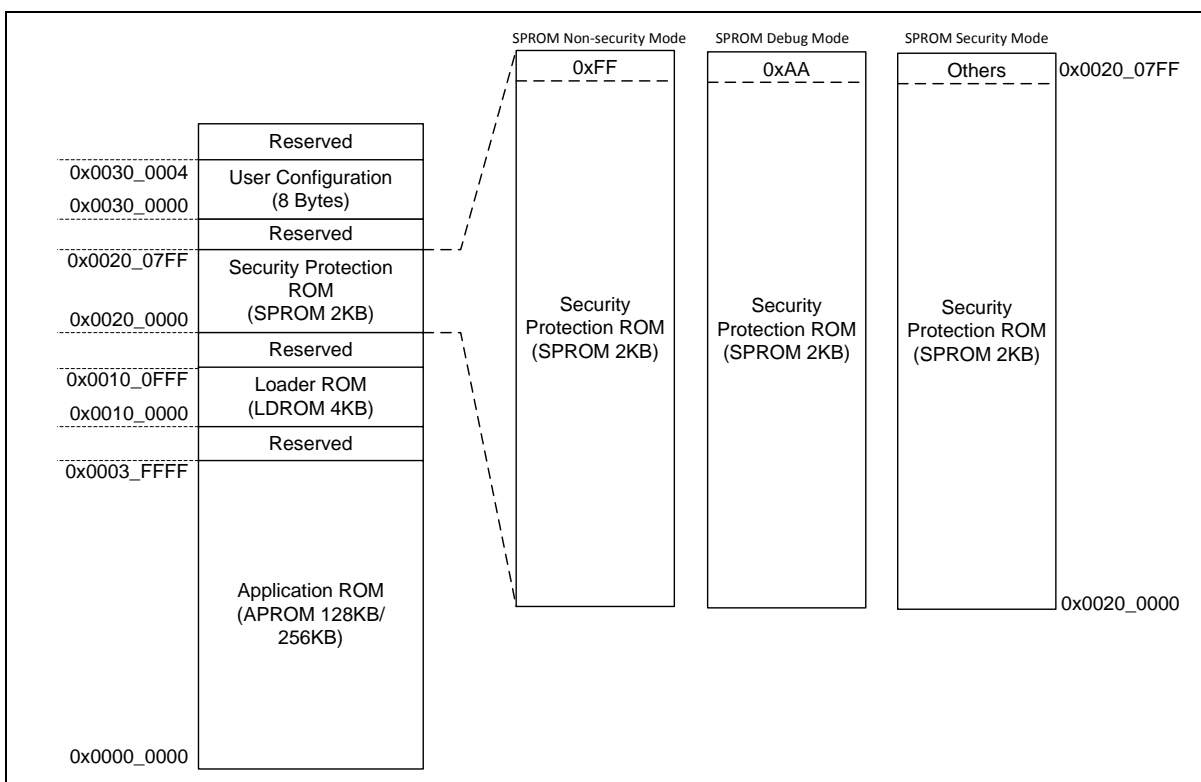


Figure 6.4-3 SPROM Security Mode

- Non-secured Mode**  
 SPROM is in non-secured mode when last byte of SPROM is 0xFF. In other words, SPROM will enter non-secured mode whenever SPROM been erased. In this mode, user can read, program and erase SPROM without any limitation.
- Debug Mode**  
 If user write 0xAA to last byte of SPROM and reboot, SPROM will be in debug mode. All SPROM behavior in debug mode is the same as secured mode except user can step into SPROM or set breakpoint in SPROM by ICE. This is helpful for user to debug the code in SPROM.

**Secured Mode**  
 To lock SPROM for protecting the code in SRPOM, user must write a bytes which is not 0xFF or 0xAA to last bytes of SPROM, e.g. 0x0, 0x1, 0x3a.... , then reboot system to let secured mode take effect. Once in secured mode, all data read by CPU or ICE will be

0x0. User cannot debug code in secured mode because it is forbidden to set breakpoint or step into SPROM. However, the code in SPROM still could work as usual by free running.

The SCODE (FMC\_ISPSTS[31]) is SPROM secured flag. If the flag is 1, it means SPROM is in secured mode or debug mode. In other words, this flag is set to 1 at booting if the last byte of SPROM isn't 0xFF, and can be cleared after the SPROM page erase operation complete.

In SPROM code development stage, user can also set SCODE=1 by software to force SPROM to enter secured mode without program last byte of SPROM.

*Flash Memory Map*

In the NUC126 series, the flash memory map is different from system memory map. The system memory map is used by CPU fetch code or data from flash memory. The flash memory map is used for ISP function to read, program or erase flash memory. The flash memory map is as Figure 6.4-4.

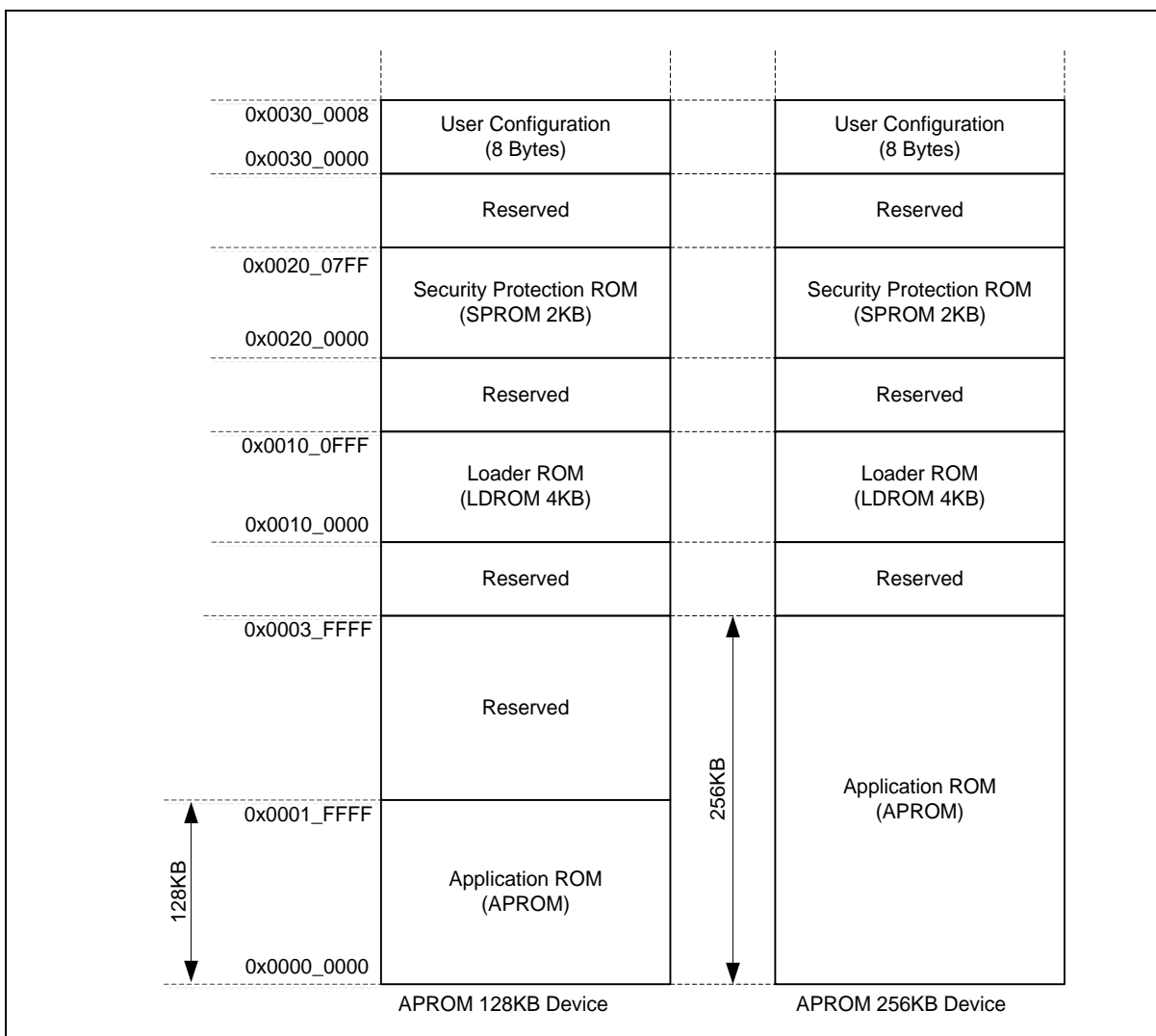


Figure 6.4-4 Flash Memory Map

*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 Data Flash is shared with APROM and the Data Flash base address is defined by CONFIG1. The content of CONFIG1 is loaded into DFBA (Data Flash Base Address Register) at the flash initiation. The memory space from DFBA to 0x0001\_FFFF/0x0003\_FFFF is the Data Flash region for CPU data access, and the space from 0x0000\_0200 to (DFBA-1) is APROM region for CPU instruction access.

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.

There are two kinds of system memory map with IAP mode when chip booting, LDROM with IAP and APROM with IAP.

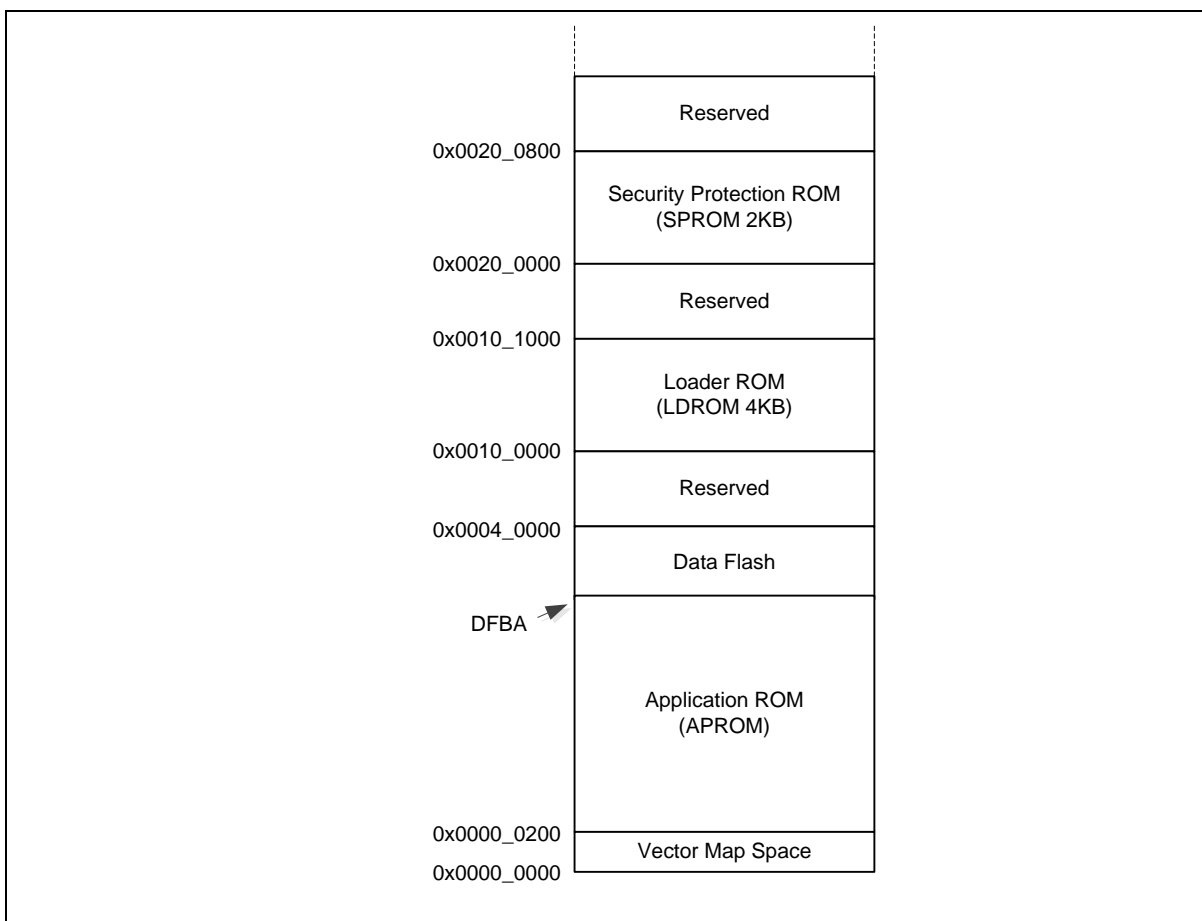


Figure 6.4-5 System Memory Map with IAP Mode

- **APROM with IAP mode**  
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 Request reset, the stack pointer and reset handler in APROM vector table are used to reboot the system. This is so called boot from APROM.
- **LDROM with IAP mode**  
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 Request

reset, the stack pointer and reset handler in LDROM vector table are used to reboot the system. This is so called boot from LDROM.

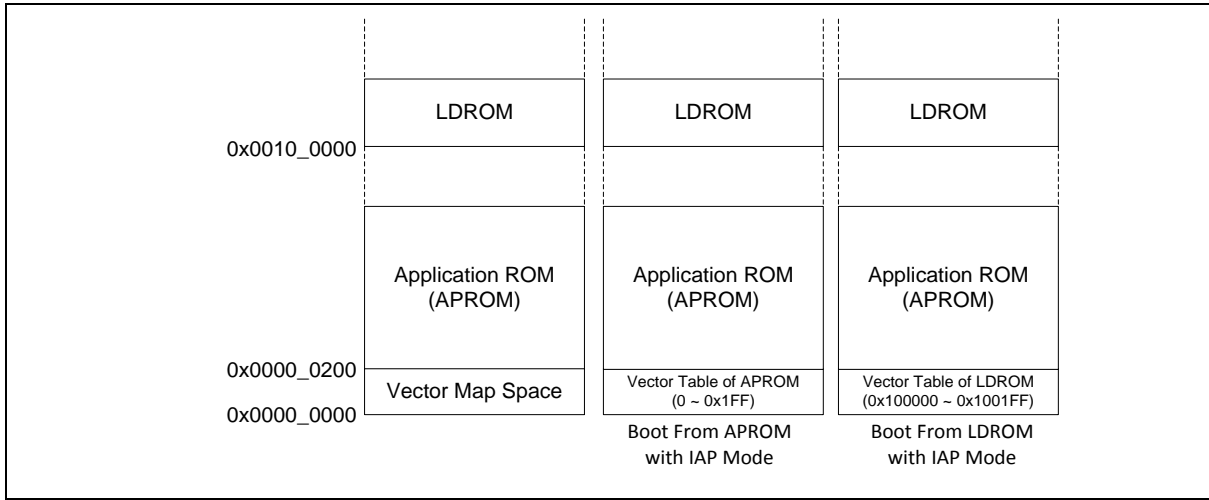


Figure 6.4-6 APROM/LDROM boot with IAP Mode

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 and it cannot support SPROM.

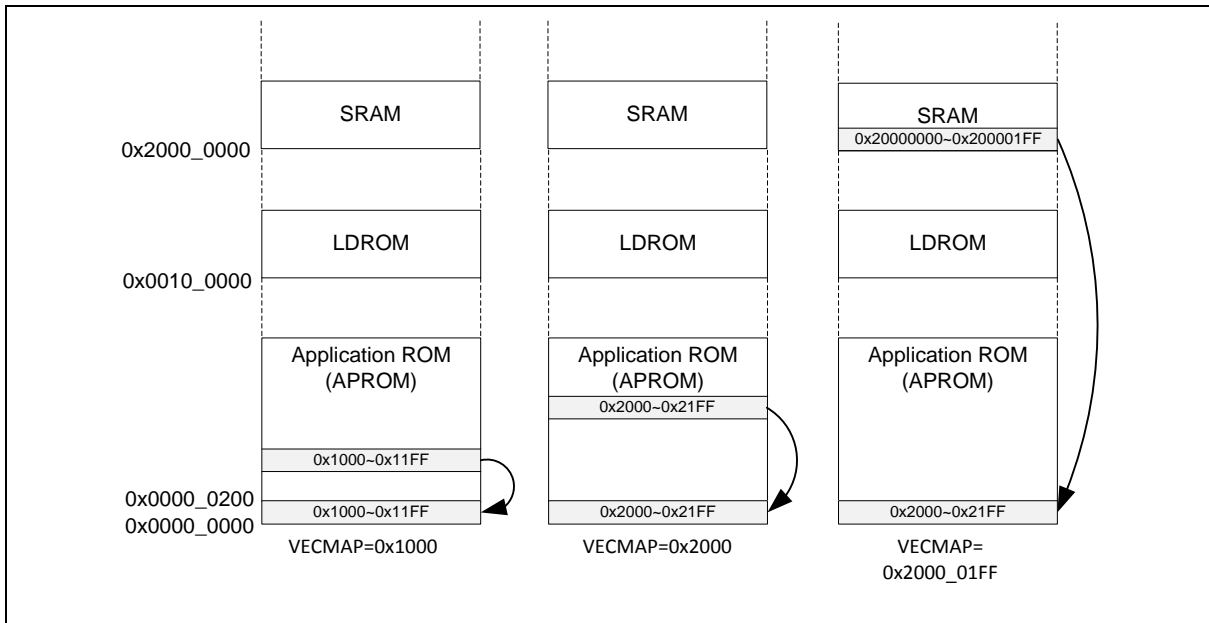


Figure 6.4-7 VECMAP setting example in IAP Mode

*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. One is LDROM without IAP, the other one is 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. Just as IAP mode, the Data Flash is shared with APROM and the Data Flash base address is defined by CONFIG1. Therefore, in APROM without IAP mode, we still can use Data Flash function.

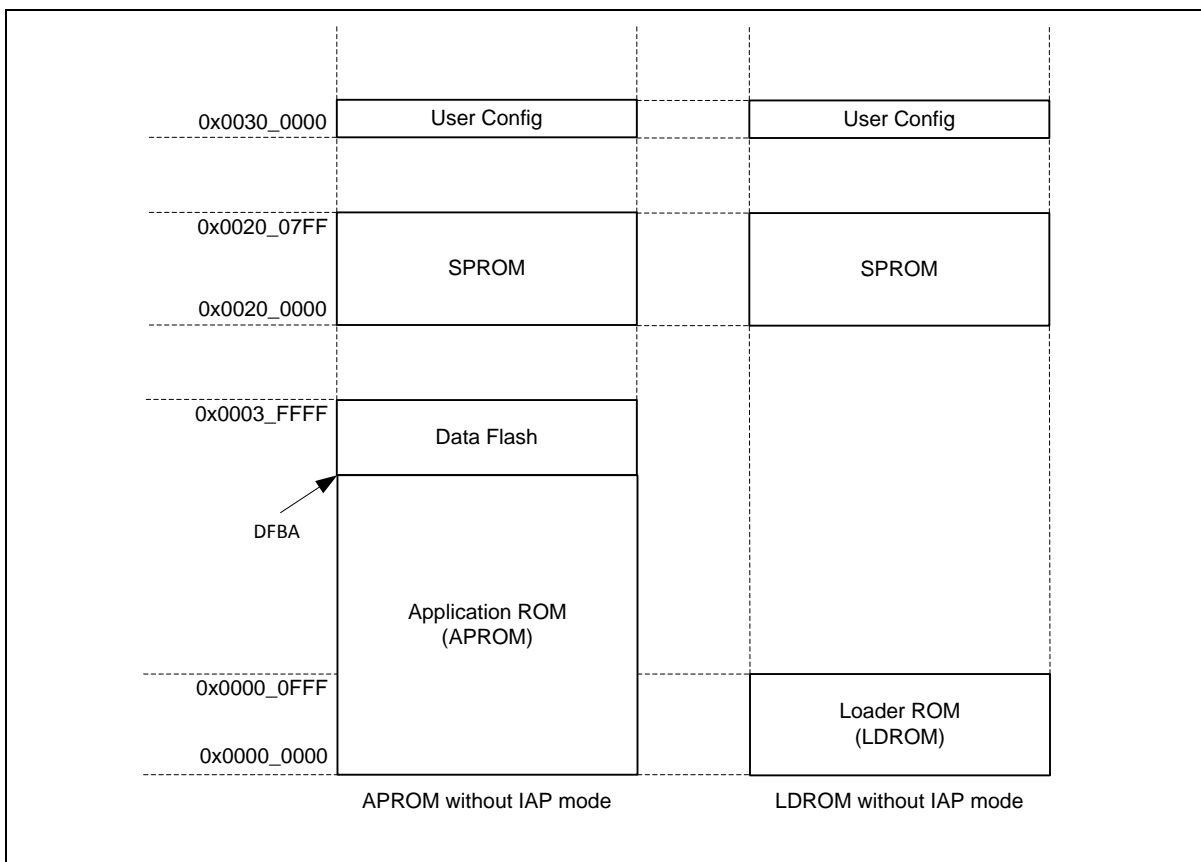


Figure 6.4-8 System Memory Map without IAP mode

**6.4.4.2 Boot Selection**

The NUC126 series 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

6.4.4.3 In-Application-Programming (IAP)

The NUC126 Series provides In-Application-Programming (IAP) mode for user to switch the code executing between APROM, LDROM and SPROM. User can enable the IAP mode by booting chip and setting the chip boot selection bits in CBS (CONFIG0[7:6]) as 10 or 00.

When chip boots with IAP mode enabled, any executable code (align to 512 bytes) is allowed to map to the system memory vector. User can set the remap address by FMC\_ISPADDR and then trigger ISP procedure with the “Vector Page Remap” command (0x2E).

6.4.4.4 In-System-Programming (ISP)

The NUC126 series supports 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 interfaces, such as UART, USB, I<sup>2</sup>C and SPI.

The NUC126 ISP provides the following functions for embedded flash memory.

- Supports flash page erase function
- Supports flash mass 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

Table 6.4-2 shows all supported ISP commands. Including command code, address and data limitations.

ISP CMD	Code	ISPADDR	ISPDAT MPDAT0~ MPDAT3
FLASH Page Erase	0x22	Valid address of flash memory. It must be page alignment.	N/A or 0x0055AA03 if erase SPROM.
FLASH Mass Erase (Valid only when CFG0[5]=0)	0x26	0x0000_0000	N/A
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 64-bit Program	0x61	Valid address of flash memory organization. It must be 64-bit alignment.	FMC_ISPDAT:N/A FMC_MPDAT0: LSB Programming Data FMC_MPDAT1: MSB Programming Data FMC_MPDAT2~FMC_MPDAT3:

			N/A
FLASH Multi-Word Program	0x27	Valid address of flash memory organization. It must be 64-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
FLASH 64-bit Read	0x40	Valid address of flash memory organization. It must be 64-bit alignment.	FMC_ISPDAT: LSB Return Data FMC_MPDAT0: LSB Return Data FMC_MPDAT1: MSB Return Data FMC_MPDAT2~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, SPROM 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 above ISP command list) for specify operation, e.g. set FMC\_ISPCMD=0x0 for 32-bit read. Set FMC\_ISPADDR for target flash memory to perform the specified ISP command. 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. For page erase SPROM, FMC\_ISPDAT is used to store the SPROM erase key 0x0055AA03.

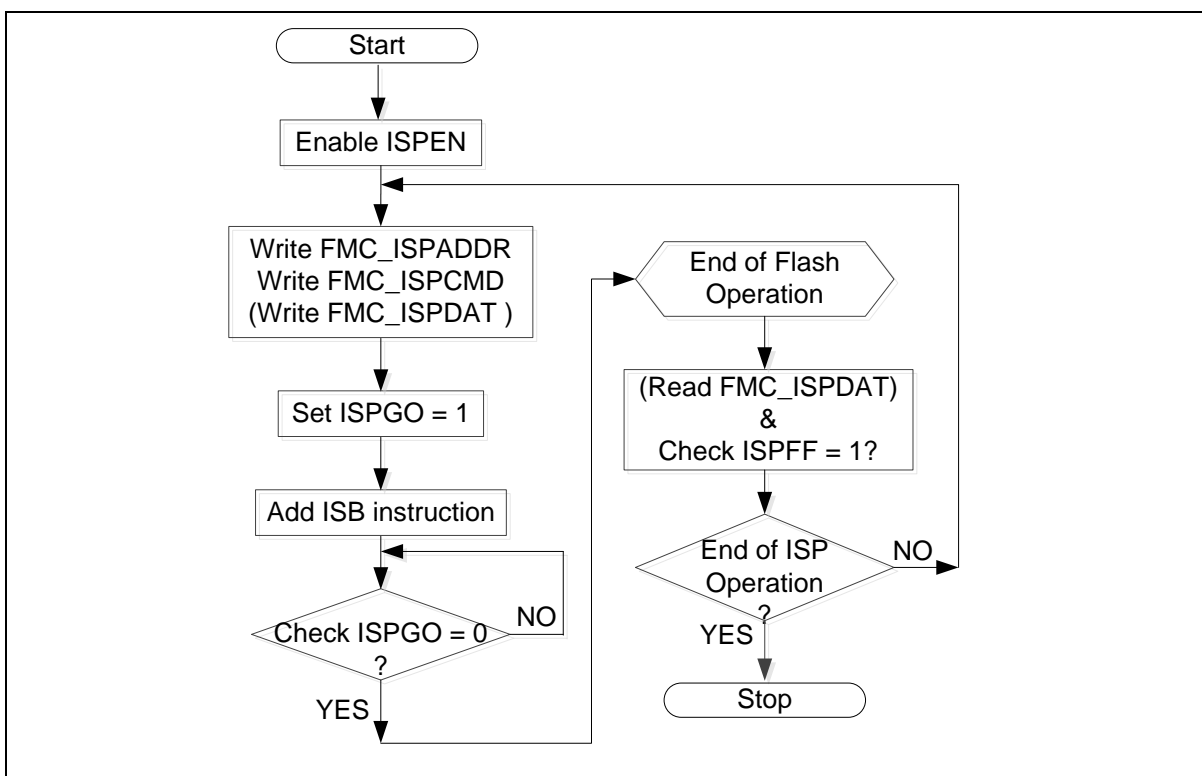


Figure 6.4-9 ISP Procedure Example

Finally, set the ISPGO (FMC\_ISPTRG[0]) register to trigger the relative ISP function. The ISPGO(FMC\_ISPTRG[0]) bit is self-cleared when ISP function has been done. To make sure ISP function has been finished before CPU goes ahead, ISB (Instruction Synchronization Barrier) instruction is used right after ISPGO(FMC\_ISPTRG[0]) setting.

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. User should add



ISB (Instruction Synchronization Barrier) instruction next to the instruction in which ISPGO (FMC\_ISPTRG[0]) bit is set 1 to ensure correct execution of the instructions following ISP operation.

6.4.4.5 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 6.4-10 shows an example to accelerating interrupt by VECMAP.

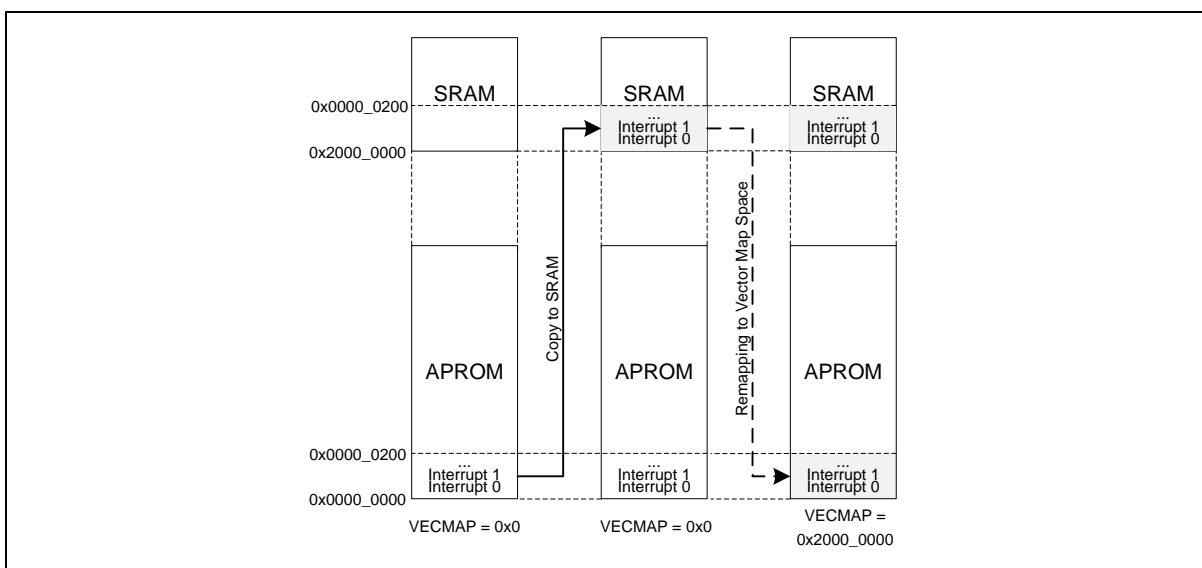


Figure 6.4-10 Example for accelerating interrupt by VECMAP.

**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.

6.4.4.6 Embedded Flash Memory Programming

The NUC126 series provides 32-bit, 64-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	64-Bit Programming	Multi-Word Programming
FMC_ISPCTL	ISP Control Register	●	●	●
FMC_ISPADDR	ISP Address Register	●	●	●
FMC_ISPDAT	ISP Data Register	●	N/A	N/A

FMC_ISPCMD	ISP CMD Register	0x21	0x61	0x27
FMC_ISPTRG	ISP Trigger Register	•	•	•
FMC_ISPSTS	ISP Status Register	•	•	N/A
FMC_MPDATA0	ISP Data0 Register	N/A	•	•
FMC_MPDATA1	ISP Data1 Register	N/A	•	•
FMC_MPDATA2	ISP Data2 Register	N/A	N/A	•
FMC_MPDATA3	ISP Data3 Register	N/A	N/A	•
FMC_MPSTS	ISP Multi-Program status	N/A	N/A	•
FMC_MPADDR	ISP Multi-Program Address	N/A	N/A	•

Table 6.4-3 FMC control registers for Flash Programming

### 64-bit Programming

The NUC126 series 64-bit programming function is faster than 32-bit programming. FMC\_ISPDAT is used for 32-bit programming data register. In 64-bit programming, there are two programming data registers, one is FMC\_MPDATA0 for LSB word, and the other is FMC\_MPDATA1 for MSB word, and ISP command is 0x61, the other registers are the same as 32-bit programming. Figure 6.4-11 / Figure 6.4-12 shows ISP 32-bit / 64-bit programming procedure.

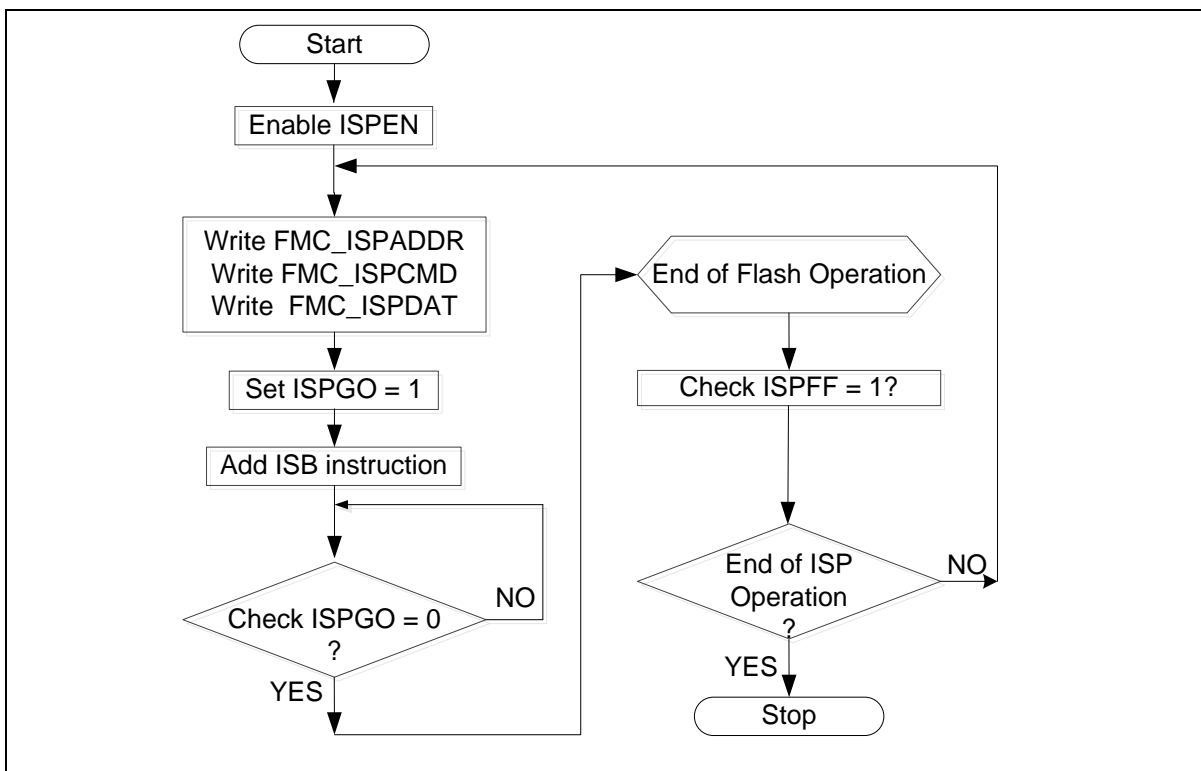


Figure 6.4-11 ISP 32-bit Programming Procedure

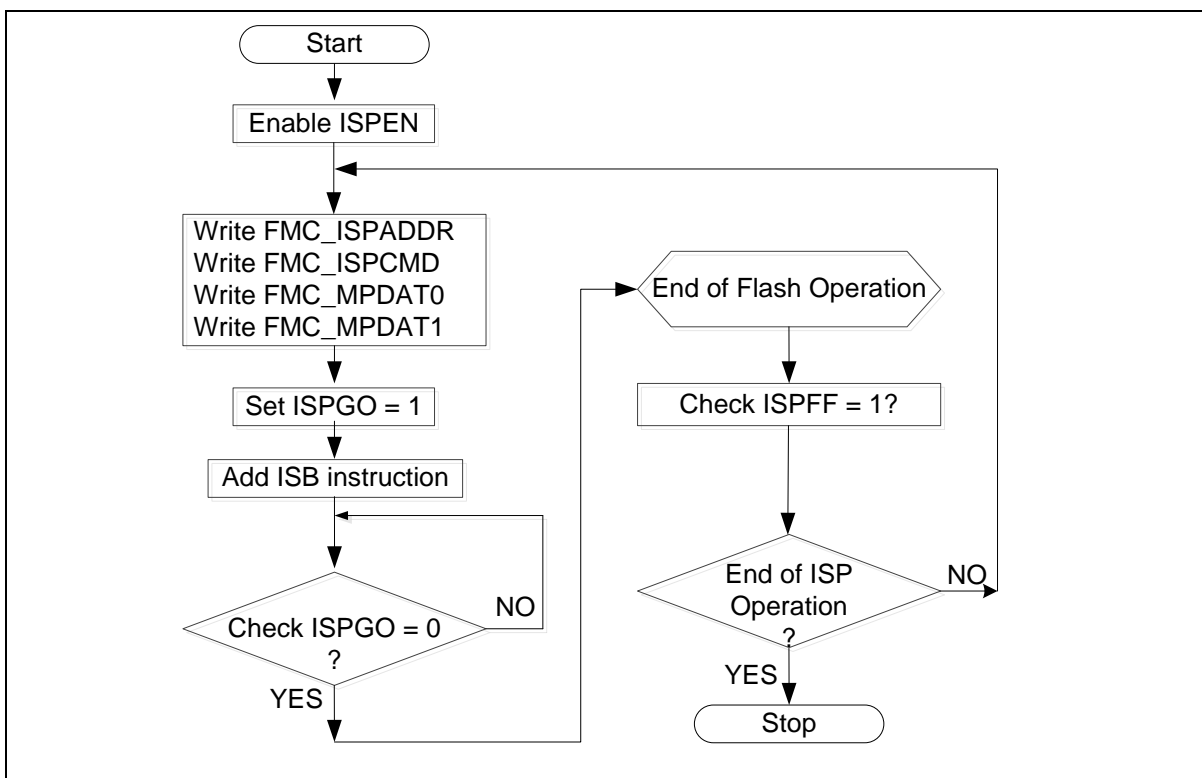


Figure 6.4-12 ISP 64-bit Programming Procedure

**Multi-word Programming**

The NUC126 series supports multi-word programming function to speed up flash updated procedure. The maximum programming length is up to 256 bytes, and the minimum programming length is 8 bytes (2 words). The multi-word programming is the fastest programming function if the programming words more than 8 bytes.

In multi-word programming operation, CPU has to monitor the empty status of the programming buffer and prepare the next data for programming continuity. The multi-program firmware should not be located in APROM, LDROM or SPROM, because CPU instruction fetch cannot be hold during ISP processing. In other words, the multi-word programming code should executed in SRAM. The multi-word programming code also needs to make sure all exceptions won't access flash memory when ISP processing.

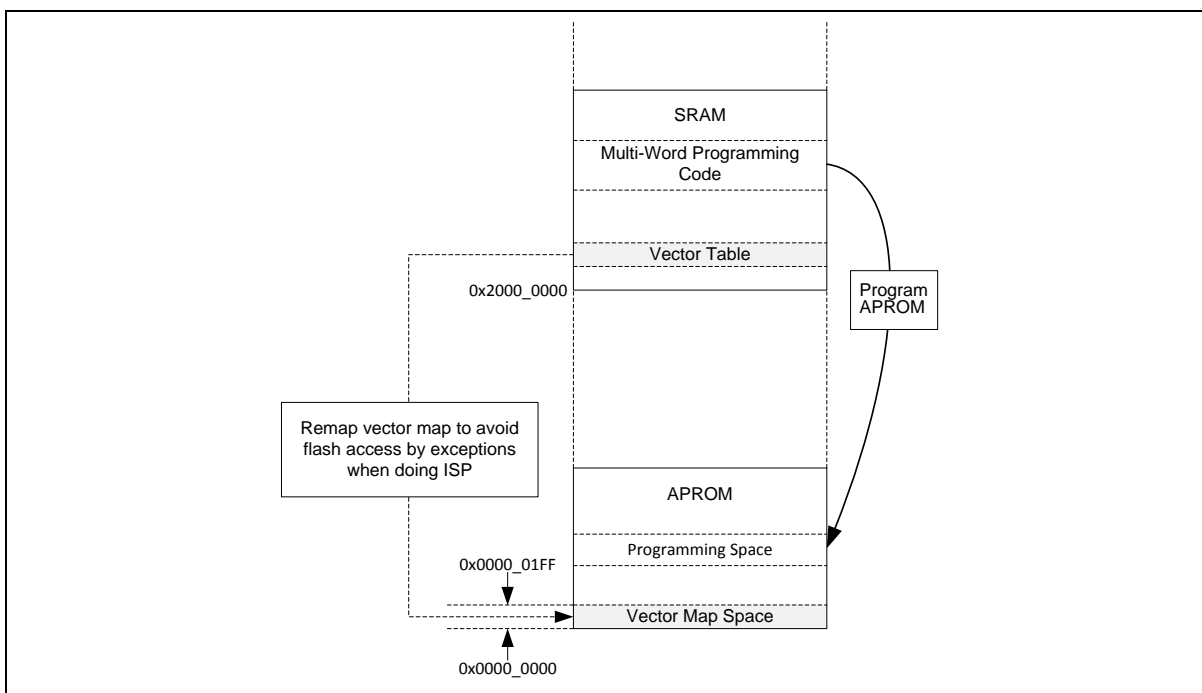


Figure 6.4-13 Firmware in SRAM for Multi-word Programming

The multi-word programming flow is shown as bellow. The starting ISP address (FMC\_ISPADDR) has to be 8-byte align. FMC\_MPDAT0 is the data word of the offset 0x0, FMC\_MPDAT1 is the second word (offset 0x4), FMC\_MPDAT2 is the third word (offset 0x8), and FMC\_MPDAT3 is forth word (offset 0xC). If the starting ISP address FMC\_ISPADDR [3] is 0, the 1<sup>st</sup> data word should put on FMC\_MPDAT0, and 2<sup>nd</sup> word is FMC\_MPDAT1, 3<sup>rd</sup> word is FMC\_MPDAT2, and 4<sup>th</sup> word is FMC\_MPDAT3. If the starting ISP address FMC\_ISPADDR [3] is 1, the 1<sup>st</sup> data word should put on FMC\_MPDAT2, and 2<sup>nd</sup> word is FMC\_MPDAT3, 3<sup>rd</sup> word is FMC\_MPDAT0, and 4<sup>th</sup> word is FMC\_MPDAT1. The maximum programming size is 256 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 the buffer empty ((D1,D0)=00, or (D3,D2)=00), and then CPU needs to update the next programming data (FMC\_MPDAT0, FMC\_MPDAT1, FMC\_MPDAT2 and FMC\_MPDAT3) in time. 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.

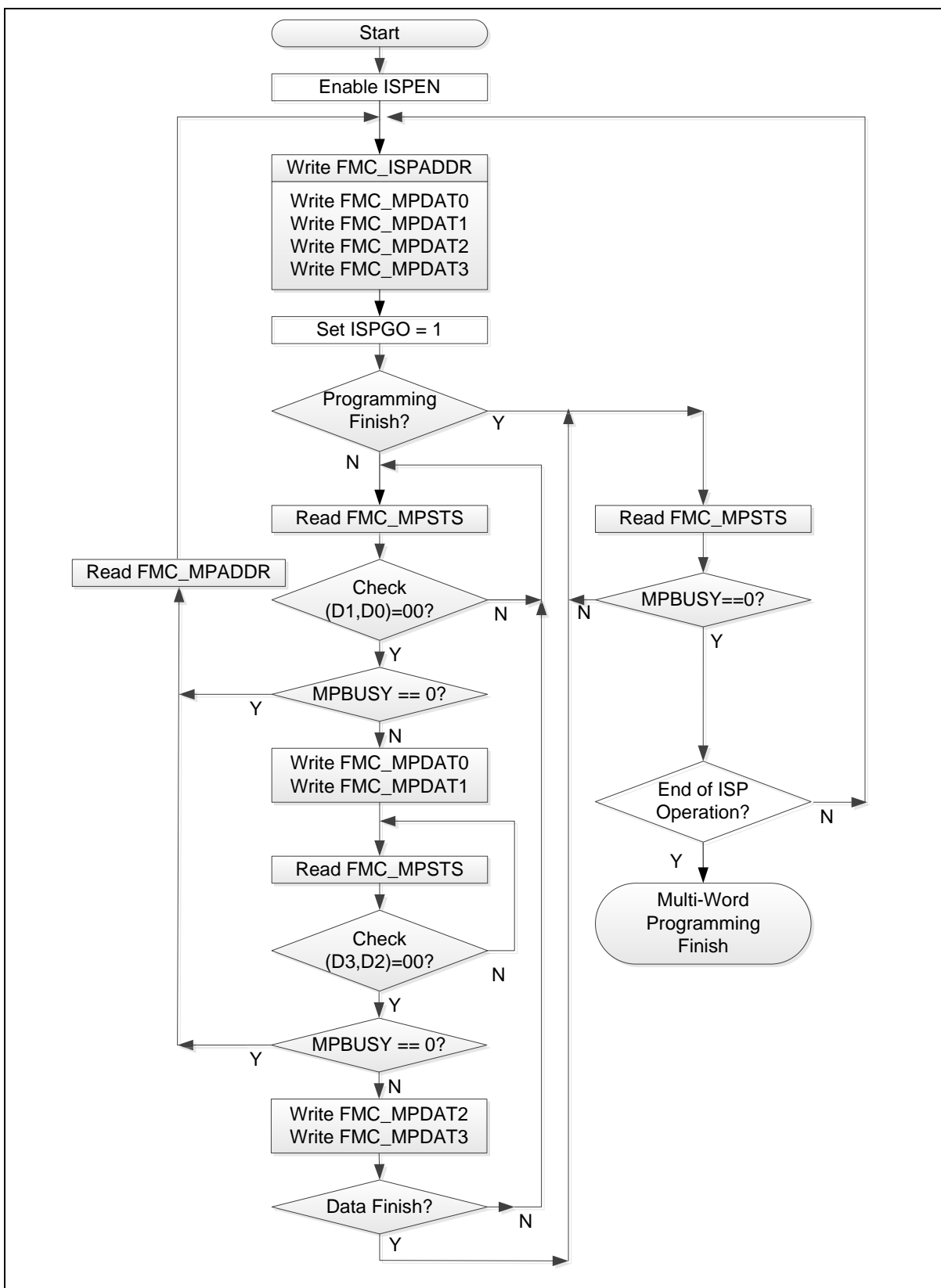


Figure 6.4-14 Multi-word Programming Flow

6.4.4.7 CRC-32 Checksum Calculation

The NUC126 series supports the Cyclic Redundancy Check (CRC) generator to perform CRC-32 checksum calculation, and help user quickly check the memory content includes APROM, LDROM and SPROM. The CRC-32 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

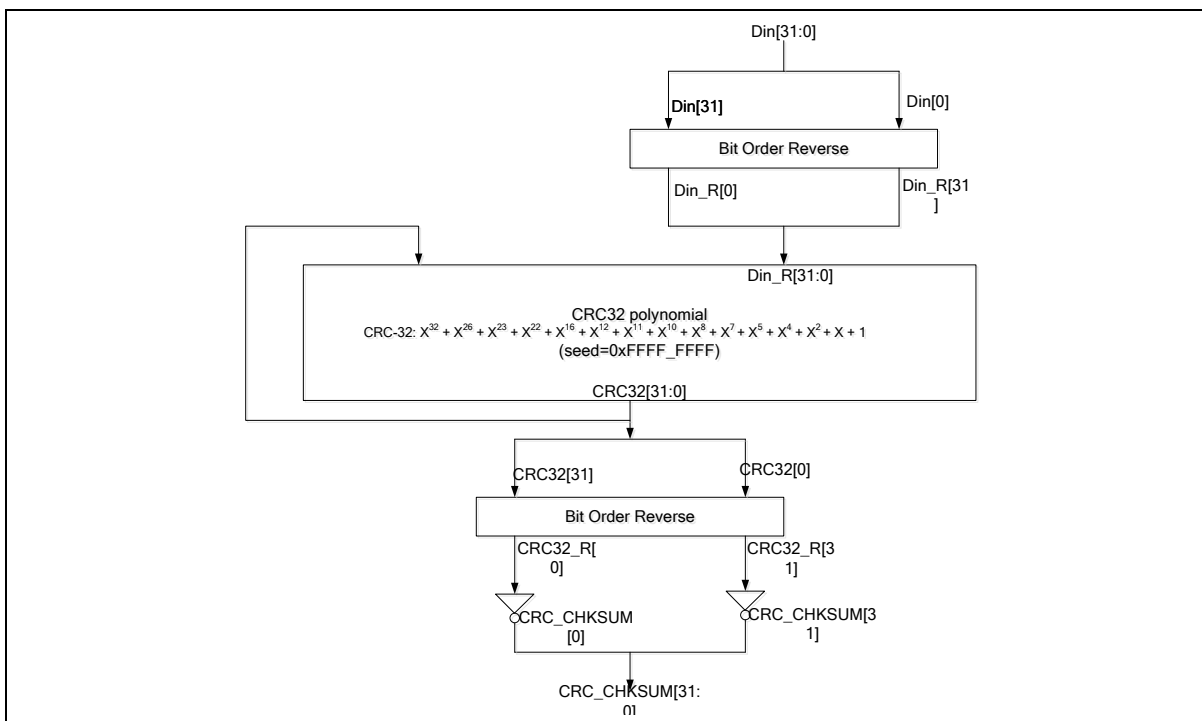


Figure 6.4-15 CRC-32 Checksum Calculation

Three steps complete this checksum calculation.

- Step 1: perform ISP “Run Memory Checksum” operation
- Step 2: perform ISP “Read Memory Checksum” operation
- 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 512 bytes alignment, the size should be  $\geq 512$  bytes and the starting address includes APROM, LDROM and SPROM.

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, it means that checksum calculation is in-progress or address and size are over device limitation.

When flash is locked, user still can 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 SPROM locked mode.

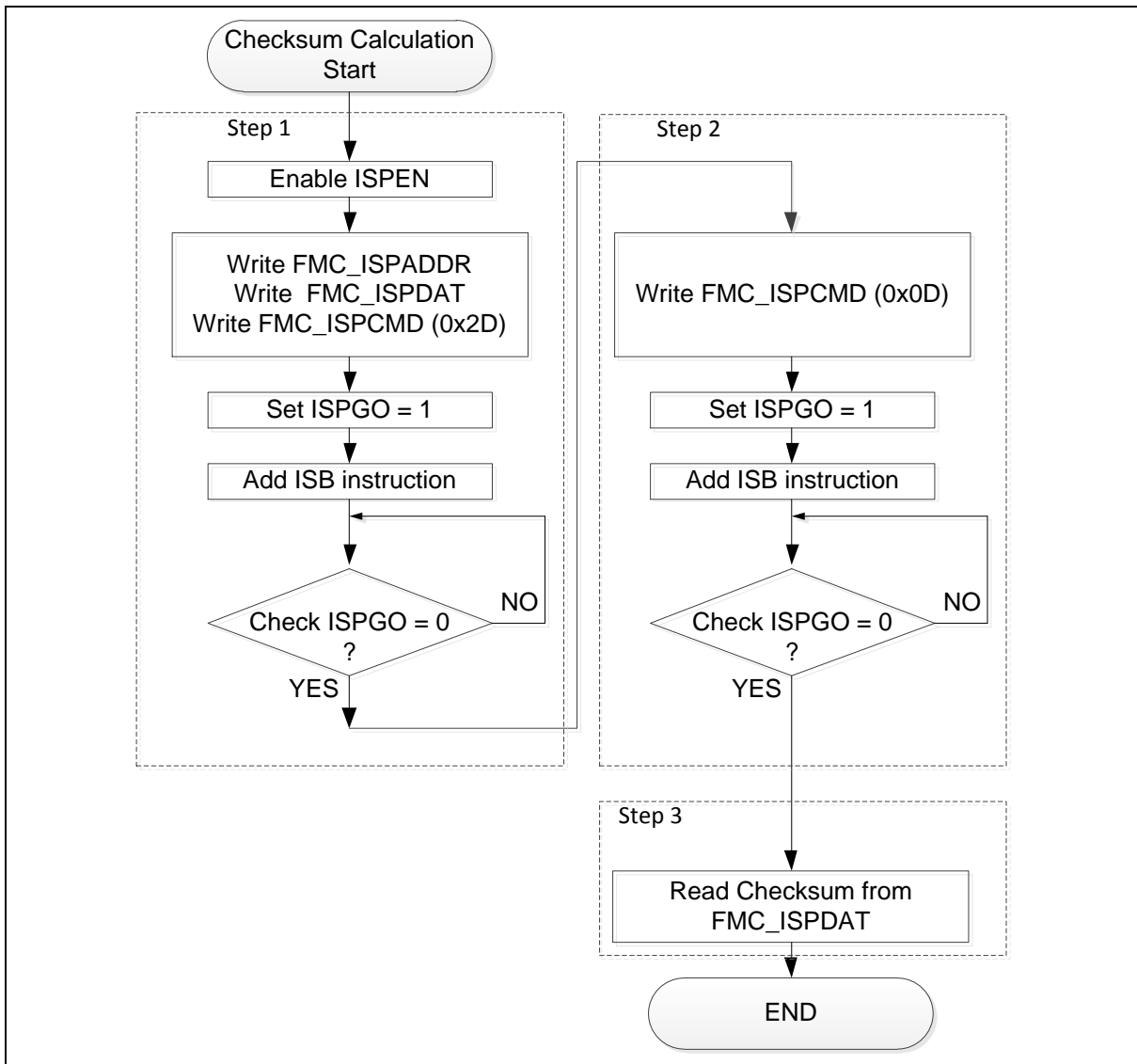


Figure 6.4-16 CRC-32 Checksum Calculation Flow

6.4.4.8 Flash All-One Verification

The NUC126 series supports the flash all one verification function to help user quickly check a memory block content blanking for APROM, LDR0M, and SPROM after flash erase operation.

Two steps complete this flash all-one verification.

Step 1: perform ISP “Run Flash All One Verification” operation

Step 2: read ALLONE(FMC\_ISPSTS[7])bit to get the verification result. The ALLONE must be 1 if all one verification pass. Otherwise, the all one verification fail.

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, LDR0M and SPROM.

In step 2, the FMC\_ISPADDR should be kept as the same as step 1.

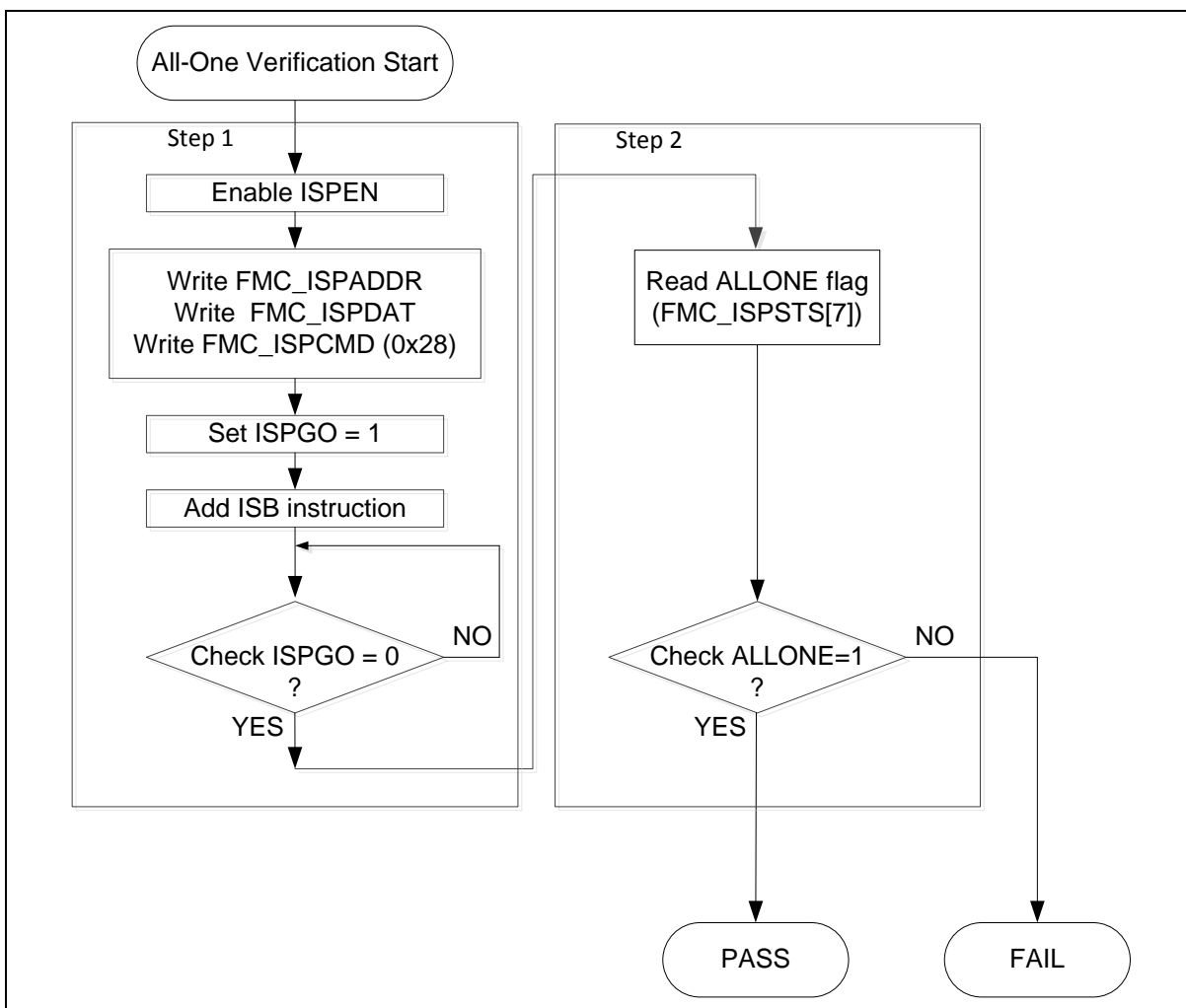


Figure 6.4-17 All-One Verification Flow



### 6.4.5 Register Map

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

Register	Offset	R/W	Description	Reset Value
<b>FMC Base Address</b>				
<b>FMC_BA = 0x5000_C000</b>				
<b>FMC_ISPCTL</b>	FMC_BA+0x00	R/W	ISP Control Register	0x0000_0000
<b>FMC_ISPADDR</b>	FMC_BA+0x04	R/W	ISP Address Register	0x0000_0000
<b>FMC_ISPDAT</b>	FMC_BA+0x08	R/W	ISP Data Register	0x0000_0000
<b>FMC_ISPCMD</b>	FMC_BA+0x0C	R/W	ISP CMD Register	0x0000_0000
<b>FMC_ISPTRG</b>	FMC_BA+0x10	R/W	ISP Trigger Control Register	0x0000_0000
<b>FMC_DFBA</b>	FMC_BA+0x14	R	Data Flash Base Address	0XXXXX_XXXX
<b>FMC_FTCTL</b>	FMC_BA+0x18	R/W	Flash Access Time Control Register	0x0000_0000
<b>FMC_ISPSTS</b>	FMC_BA+0x40	R/W	ISP Status Register	0xX000_000X
<b>FMC_MPDAT0</b>	FMC_BA+0x80	R/W	ISP Data0 Register	0x0000_0000
<b>FMC_MPDAT1</b>	FMC_BA+0x84	R/W	ISP Data1 Register	0x0000_0000
<b>FMC_MPDAT2</b>	FMC_BA+0x88	R/W	ISP Data2 Register	0x0000_0000
<b>FMC_MPDAT3</b>	FMC_BA+0x8C	R/W	ISP Data3 Register	0x0000_0000
<b>FMC_MPSTS</b>	FMC_BA+0xC0	R	ISP Multi-Program Status Register	0x0000_0000
<b>FMC_MPADDR</b>	FMC_BA+0xC4	R	ISP Multi-Program Address 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	ISPPF	LDUEN	CFGUEN	APUEN	SPUEN	BS	ISPEN

Bits	Description
[31:15]	<b>Reserved</b> Reserved.
[14:12]	<b>Reserved</b> To be 000, Reserved
[11]	<b>Reserved</b> Reserved.
[10:8]	<b>Reserved</b> To be 000, Reserved
[7]	<b>Reserved</b> Reserved.
[6]	<p><b>ISPPF</b></p> <p><b>ISP Fail Flag (Write Protect)</b> 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.</p> <ol style="list-style-type: none"> <li>(1) APROM writes to itself if APUEN is set to 0.</li> <li>(2) LDROM writes to itself if LDUEN is set to 0.</li> <li>(3) CONFIG is erased/programmed if CFGUEN is set to 0.</li> <li>(4) SPROM is erased/programmed if SPUEN is set to 0</li> <li>(5) SPROM is programmed at SPROM secured mode.</li> <li>(6) Page Erase command at LOCK mode with ICE connection</li> <li>(7) Erase or Program command at brown-out detected</li> <li>(8) Destination address is illegal, such as over an available range.</li> <li>(9) Invalid ISP commands</li> </ol> <p><b>Note:</b> This bit is write-protected. Refer to the SYS_REGLCTL register.</p>
[5]	<p><b>LDUEN</b></p> <p><b>LDROM Update Enable Bit (Write Protect)</b> LDROM update enable bit. 0 = LDROM cannot be updated. 1 = LDROM can be updated.</p> <p><b>Note:</b> This bit is write-protected. Refer to the SYS_REGLCTL register.</p>

[4]	CFGUEN	<p><b>CONFIG Update Enable Bit (Write Protect)</b></p> <p>0 = CONFIG cannot be updated. 1 = CONFIG can be updated.</p> <p><b>Note:</b> This bit is write-protected. Refer to the SYS_REGLCTL register.</p>
[3]	APUEN	<p><b>APROM Update Enable Bit (Write Protect)</b></p> <p>0 = APROM cannot be updated when the chip runs in APROM. 1 = APROM can be updated when the chip runs in APROM.</p> <p><b>Note:</b> This bit is write-protected. Refer to the SYS_REGLCTL register.</p>
[2]	SPUEN	<p><b>SPROM Update Enable Bit (Write Protect)</b></p> <p>0 = SPROM cannot be updated. 1 = SPROM can be updated.</p> <p><b>Note:</b> This bit is write-protected. Refer to the SYS_REGLCTL register.</p>
[1]	BS	<p><b>Boot Select (Write Protect)</b></p> <p>Set/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 (CPU is 1) or system reset (SYS) is happened</p> <p>0 = Booting from APROM. 1 = Booting from LDROM.</p> <p><b>Note:</b> This bit is write-protected. Refer to the SYS_REGLCTL register.</p>
[0]	ISPEN	<p><b>ISP Enable Bit (Write Protect)</b></p>

**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]	ISPADDR	ISP Address

**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]	ISPDAT	ISP Data

**ISP CMD (FMC\_ISPCMD)**

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

**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)

**Data Flash Base Address Register (FMC\_DFBA)**

Register	Offset	R/W	Description	Reset Value
FMC_DFBA	FMC_BA+0x14	R	Data Flash Base Address	0xXXXX_XXXX

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

Bits	Description	
[31:0]	DFBA	Data Flash Base Address



**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							MFPSOFF
7	6	5	4	3	2	1	0
CACHEOFF	FOM			FATS			FPSEN

Bits	Description	
[31:8]	Reserved	Reserved.
[7]	CACHEOFF	<p><b>Flash Cache Disable Bit (Write Protect)</b>                      0 = Flash Cache function Enabled (default).                      1 = Flash Cache function Disabled.  <b>Note:</b> This bit is write-protected. Refer to the SYS_REGLCTL register.</p>
[6:4]	FOM	<p><b>Frequency Optimization Mode (Write Protect)</b>                      The NUC126 series supports adjustable flash access timing to optimize the flash access cycles in different working frequency.                      0x1 = Frequency ≤ 24MHz.                      1x1 = Frequency ≤ 72MHz.                      Others = Frequency ≤ 48MHz.  <b>Note:</b> This bit is write-protected. Refer to the SYS_REGLCTL register.</p>
[3:0]	Reserved	Reserved.

**ISP Status Register (FMC ISPSTS)**

Register	Offset	R/W	Description	Reset Value
FMC_ISPSTS	FMC_BA+0x40	R/W	ISP Status Register	0xX000_000X

31	30	29	28	27	26	25	24
SCODE	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	ISPPF	Reserved			CBS		ISPBUSY

Bits	Description	
[31]	SCODE	<p><b>Security Code Active Flag</b></p> <p>This bit is set by hardware when detecting SPROM secured code is active at flash initiation, or software writes 1 to this bit to make secured code active; this bit is clear by SPROM page erase operation.</p> <p>0 = Secured code is inactive. 1 = Secured code is active.</p>
[30]	Reserved	Reserved.
[29:9]	VECMAP	<p><b>Vector Page Mapping Address (Read Only)</b></p> <p>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}, except SPROM.</p> <p>VECMAP [20:19] = 00 system vector address is mapped to flash memory. VECMAP [20:19] = 10 system vector address is mapped to SRAM memory. VECMAP [18:12] should be 0.</p>
[8]	Reserved	Reserved.
[7]	ALLONE	<p><b>Flash All-one Verification Flag</b></p> <p>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 can also be cleared by writing 1</p> <p>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.</p>

[6]	ISPFF	<p><b>ISP Fail Flag (Write Protect)</b></p> <p>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:</p> <ul style="list-style-type: none"> <li>(1) APROM writes to itself if APUEN is set to 0.</li> <li>(2) LDROM writes to itself if LDUEN is set to 0.</li> <li>(3) CONFIG is erased/programmed if CFGUEN is set to 0.</li> <li>(4) SPROM is erased/programmed if SPUEN is set to 0</li> <li>(5) SPROM is programmed at SPROM secured mode.</li> <li>(6) Page Erase command at LOCK mode with ICE connection</li> <li>(7) Erase or Program command at brown-out detected</li> <li>(8) Destination address is illegal, such as over an available range.</li> <li>(9) Invalid ISP commands.</li> <li>(10) system vector address is remapped to SPROM.</li> </ul> <p><b>Note:</b> This bit is write-protected. Refer to the SYS_REGLCTL register.</p>
[5:3]	Reserved	Reserved.
[2:1]	CBS	<p><b>Boot Selection of CONFIG (Read Only)</b></p> <p>This bit is initiated with the CBS (CONFIG0[7:6]) after any reset is happened except CPU reset (CPU is 1) or system reset (SYS) is happened.</p> <ul style="list-style-type: none"> <li>00 = LDROM with IAP mode.</li> <li>01 = LDROM without IAP mode.</li> <li>10 = APROM with IAP mode.</li> <li>11 = APROM without IAP mode.</li> </ul>
[0]	ISPBUSY	<b>ISP Busy Flag (Read Only)</b>

**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]	ISPDAT0	ISP Data 0

**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

**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
ISP DAT2							
23	22	21	20	19	18	17	16
ISP DAT2							
15	14	13	12	11	10	9	8
ISP DAT2							
7	6	5	4	3	2	1	0
ISP DAT2							

Bits	Description	
[31:0]	ISP DAT2	ISP Data 2

**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

**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	ISPPF	PPGO	MPBUSY

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



[2]	<b>ISPFF</b>	<p><b>ISP Fail Flag (Read Only)</b></p> <p>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:</p> <ul style="list-style-type: none"> <li>(1) APROM writes to itself if APUEN is set to 0.</li> <li>(2) LDROM writes to itself if LDUEN is set to 0.</li> <li>(3) CONFIG is erased/programmed if CFGUEN is set to 0.</li> <li>(4) Page Erase command at LOCK mode with ICE connection</li> <li>(5) Erase or Program command at brown-out detected</li> <li>(6) Destination address is illegal, such as over an available range.</li> <li>(7) Invalid ISP commands</li> </ul>
[1]	<b>PPGO</b>	<p><b>ISP Multi-program Status (Read Only)</b></p> <p>0 = ISP multi-word program operation is not active.            1 = ISP multi-word program operation is in progress.</p>
[0]	<b>MPBUSY</b>	<p><b>ISP Multi-word Program Busy Flag (Read Only)</b></p>

**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

## 6.5 Analog Comparator Controller (ACMP)

### 6.5.1 Overview

NUC126 contains two analog comparators. The comparator output is logic 1 when positive input is greater than negative input; otherwise, the output is 0. Each comparator can be configured to generate an interrupt when the comparator output state changes.

### 6.5.2 Features

- Analog input voltage range: 0 ~  $V_{DDA}$  (voltage of  $AV_{DD}$  pin)
- Supports hysteresis function
- Supports wake-up function
- Selectable input sources of positive input and negative input
- ACMP0 supports
  - 4 positive sources:
    - ◆ ACMP0\_P0, ACMP0\_P1, ACMP0\_P2, or ACMP0\_P3
  - 3 negative sources:
    - ◆ ACMP0\_N
    - ◆ Comparator Reference Voltage (CRV)
    - ◆ Internal band-gap voltage ( $V_{BG}$ )
- ACMP1 supports
  - 4 positive sources:
    - ◆ ACMP1\_P0, ACMP1\_P1, ACMP1\_P2, or ACMP1\_P3
  - 3 negative sources
    - ◆ ACMP1\_N
    - ◆ Comparator Reference Voltage (CRV)
    - ◆ Internal band-gap voltage ( $V_{BG}$ )
- Shares one ACMP interrupt vector for all comparators
- Supports window Latch mode
- Supports window compare mode



### 6.5.4 Basic Configuration

- Clock Source Configuration
  - Enable ACMP peripheral clock in ACMP01CKEN (CLK\_APBCLK0[30]).
- Reset Configuration
  - Reset ACMP controller in ACMP01RST (SYS\_IPRST1[22]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
ACMP0	ACMP0_N	PB.4	MFP5
	ACMP0_O	PD.6, PD.7	MFP5
	ACMP0_P0	PB.7	MFP5
	ACMP0_P1	PB.6	MFP5
	ACMP0_P2	PB.5	MFP5
	ACMP0_P3	PB.15	MFP5
	ACMP0_WLAT	PC.0	MFP5
ACMP1	ACMP1_N	PD.0	MFP5
	ACMP1_O	PC.2, PC.6	MFP5
	ACMP1_P0	PD.3	MFP5
	ACMP1_P1	PD.2	MFP5
	ACMP1_P2	PD.1	MFP5
	ACMP1_P3	PD.9	MFP5
	ACMP1_WLAT	PC.1	MFP5

### 6.5.5 Functional Description

#### 6.5.5.1 Hysteresis Function

The analog comparator provides the hysteresis function to make the comparator to have a stable output transition. If comparator output is 0, it will not be changed to 1 until the positive input voltage exceeds the negative input voltage by a high threshold voltage. Similarly, if comparator output is 1, it will not be changed to 0 until the positive input voltage drops below the negative input voltage by a low threshold voltage.

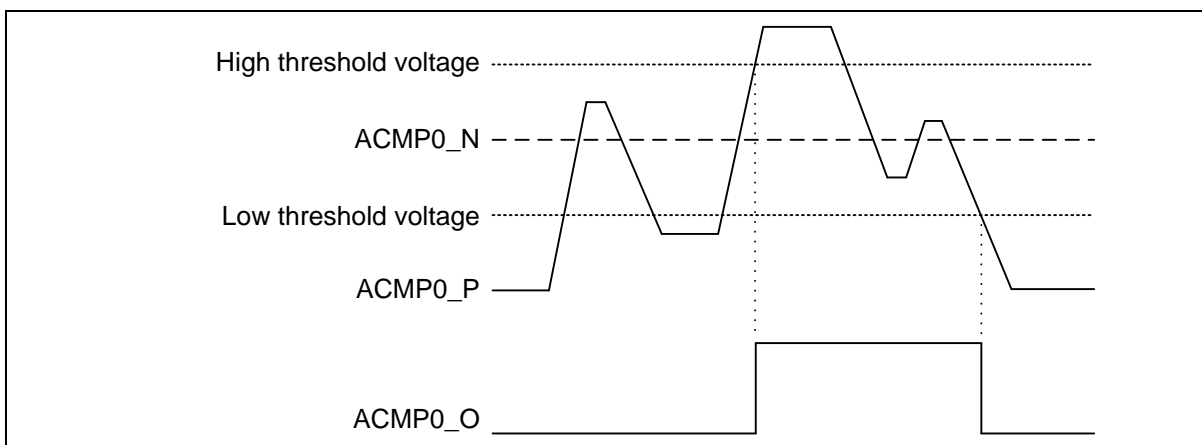


Figure 6.5-2 Comparator Hysteresis Function of ACMP0

6.5.5.2 Window Latch Function

Figure 6.5-3 shows the comparator operation of window latch function. Window latch function can be enabled by setting WLATEN (ACMP\_CTL0/1[17]) to 1. When window latch function enabled, ACMP0/1\_WLAT pin is used to control the output WLATOUT0/1. When ACMP0/1\_WLAT pin is high, ACMP0/1 passes through to WLATOUT0/1. When ACMP0/1\_WLAT pin is low, WLATOUT0/1 will keep last state of WLATOUT0/1.

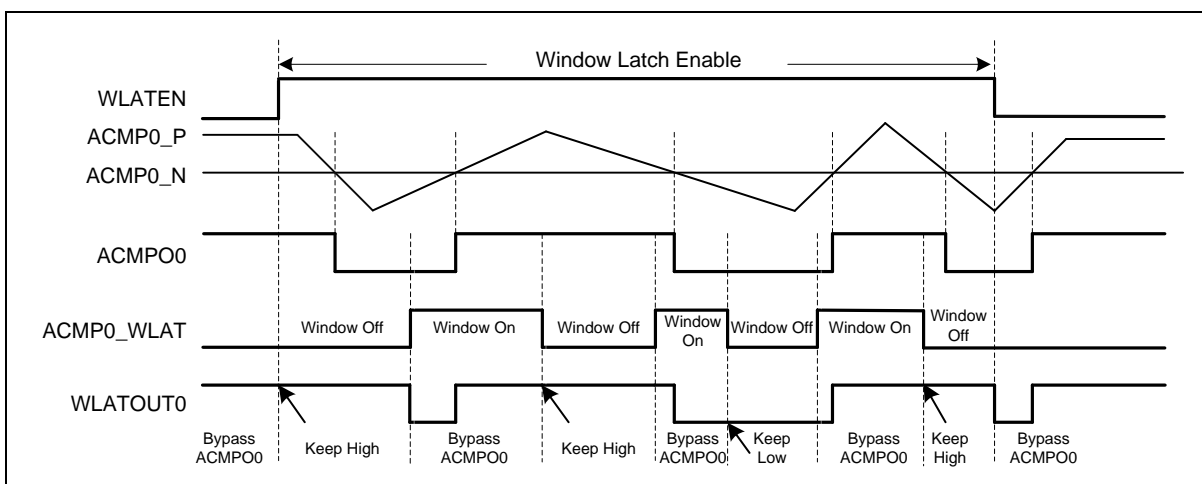


Figure 6.5-3 Window Latch Mode of ACMP0

6.5.5.3 Filter Function

The analog comparator provides filter function to avoid the un-stable state of comparator output.

By setting FILTSEL (ACMP\_CTL0[15:13], ACMP\_CTL1[15:13]), the comparator output would be sampled by consecutive PCLKs. With longer sample clocks, the comparator output would be more stable. But the sensitivity of comparator output would be reduced.

Figure 6.5-4 shows an example of filter function of ACMP0 with FILTSEL = 3 (4 PCLK). In this example, the comparing result is sampled by PCLK. All result must keep for 4 PCLK clocks before it can be output to ACMP00. If the comparing result is shorter than 4 PCLK, it will be filtered.

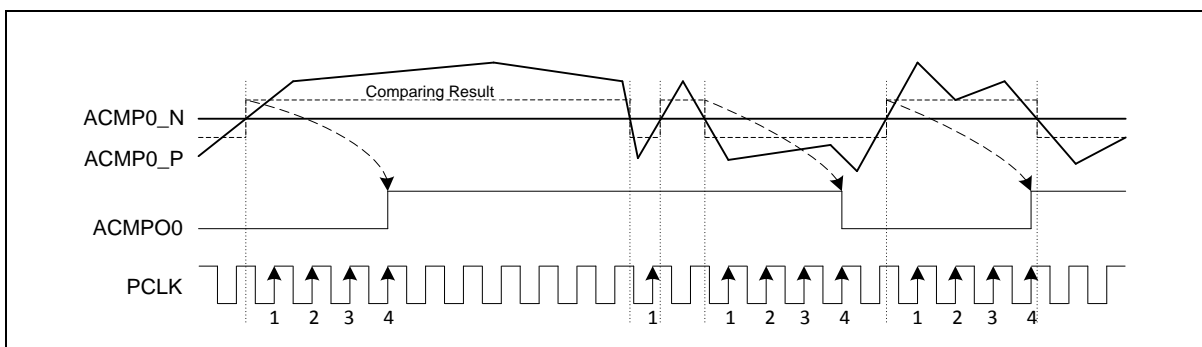


Figure 6.5-4 Filter Function Example

#### 6.5.5.4 Interrupt

The outputs of ACMP0 and ACMP1 are reflected at ACMPO0 (ACMP\_STATUS[4]) and ACMPO1 (ACMP\_STATUS[5]) respectively. Then they are processed by window latch and filter functions. Finally, the output signal could be utilized to assert interrupts. If ACMPIE of ACMP\_CTL0/1 register is set to 1, the interrupt will be enabled. If the output state ACMPO0/1 is changed as the setting of INTPOL (ACMP\_CTL0/1[9:8]), the comparator interrupt will be asserted and the corresponding flag, ACMPIF0 (ACMP\_STATUS[0]) and ACMPIF1 (ACMP\_STATUS[1]), will be set to 1. The interrupt flag can be cleared to 0 by writing 1.

If ACMP wakeup function enabled and system wakeup from power down by ACMP with interrupt enabled (ACMPIE), the WKIF (ACMP\_STATUS[8], ACMP\_STATUS[9]) will be set and cause interrupt rising.

Figure 6.5-5 shows the interrupts of ACMP is coming from ACMPIF or WKIF and enabled or disabled by ACMPIE.

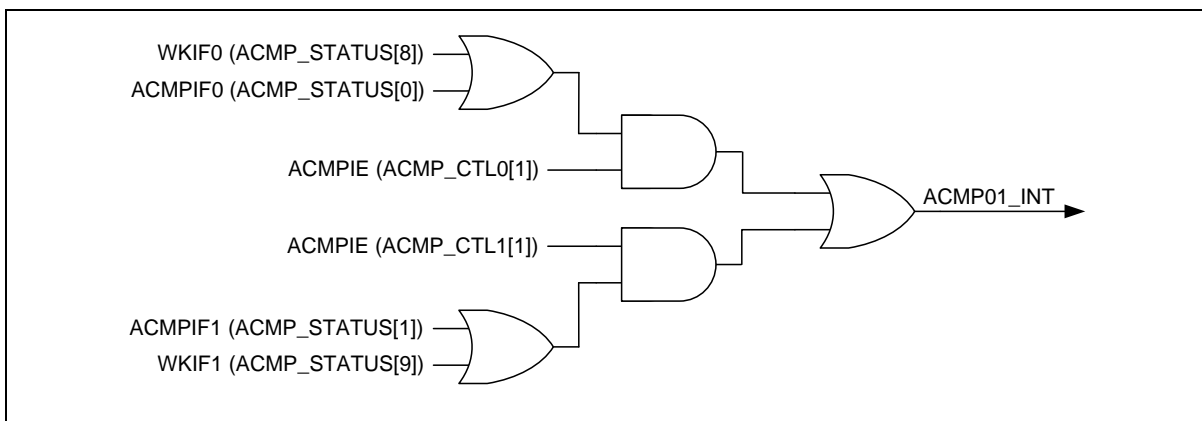


Figure 6.5-5 Comparator Controller Interrupt

#### 6.5.5.5 Comparator Reference Voltage (CRV)

The comparator reference voltage (CRV) module is responsible for generating reference voltage for comparators. The CRV module consists of resistor ladder and analog switch. User can set the CRV output voltage that is equal to CRV source voltage \* (1/6+CRVCTL/24) by setting CRVCTL (ACMP\_VREF[3:0]). The CRV output voltage can be selected as the negative input of comparator by setting NEGSEL (ACMP\_CTL0[5:4], ACMP\_CTL1[5:4]).

The resistor ladder will be disabled by hardware to reduce power consumption when NEGSEL (ACMP\_CTL0[5:4], ACMP\_CTL1[5:4]) is not selected to CRV module. The reference voltage of resistor ladder can be the voltage of AV<sub>DD</sub> pin or the INT\_VREF voltage which is controlled by SYS\_VREFCTL register.

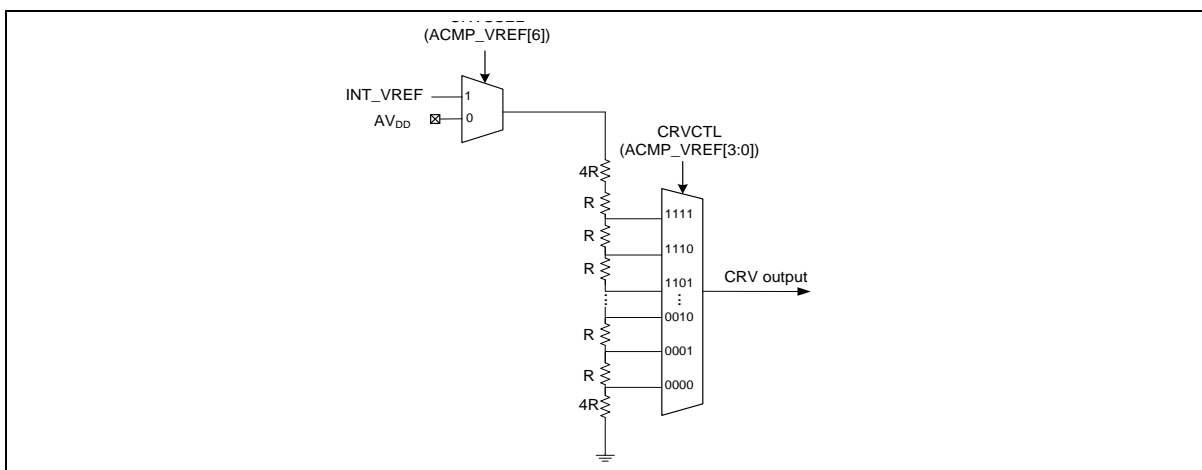


Figure 6.5-6 Comparator Reference Voltage Block Diagram

### 6.5.5.6 Window Compare Mode

The NUC126 comparator provides window compare mode. When window compare mode is enabled by setting WCMPSEL (ACMP\_CTL0/1[18]) to 1, user can monitor a specific analog voltage source with a designated range. User can connect the specific analog voltage source to either the positive inputs of both comparators or the negative inputs of both comparators. The upper bound and lower bound of the designated range are determined by the voltages applied to the other inputs of both comparators. If the output of a comparator is low and the other comparator outputs high, which means two comparators implies the upper and lower bound. User can directly monitor a specific analog voltage source via ACMPWO (ACMP\_STATUS[16]). If ACMPWO is high, it implies a specific analog voltage source is in the range of upper and lower bound, which are called as the analog voltage is in the window.

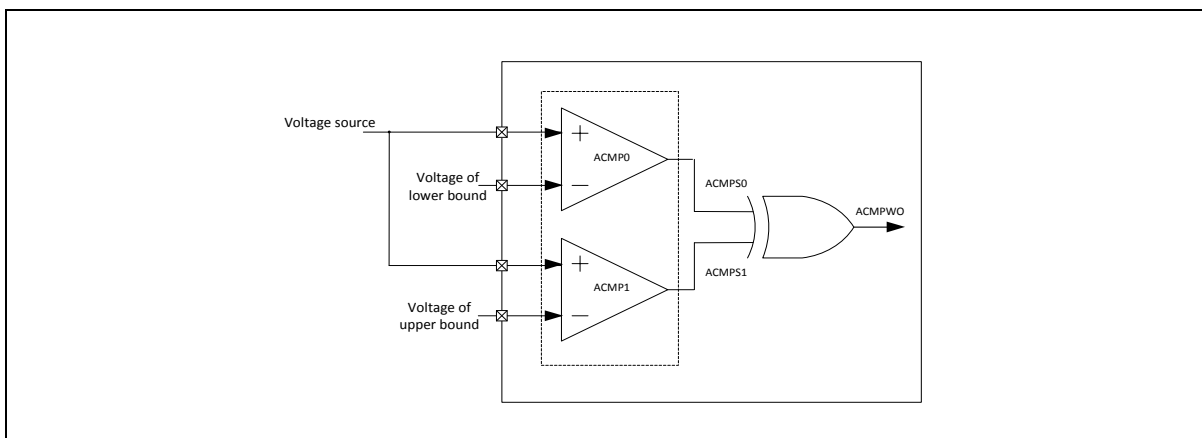


Figure 6.5-7

Figure 6.5-7 illustrates a example connection of window compare mode. In this example, once window compare mode is selected, user can choose one of four positive input sources of each comparator and connect these two inputs together outside the chip.



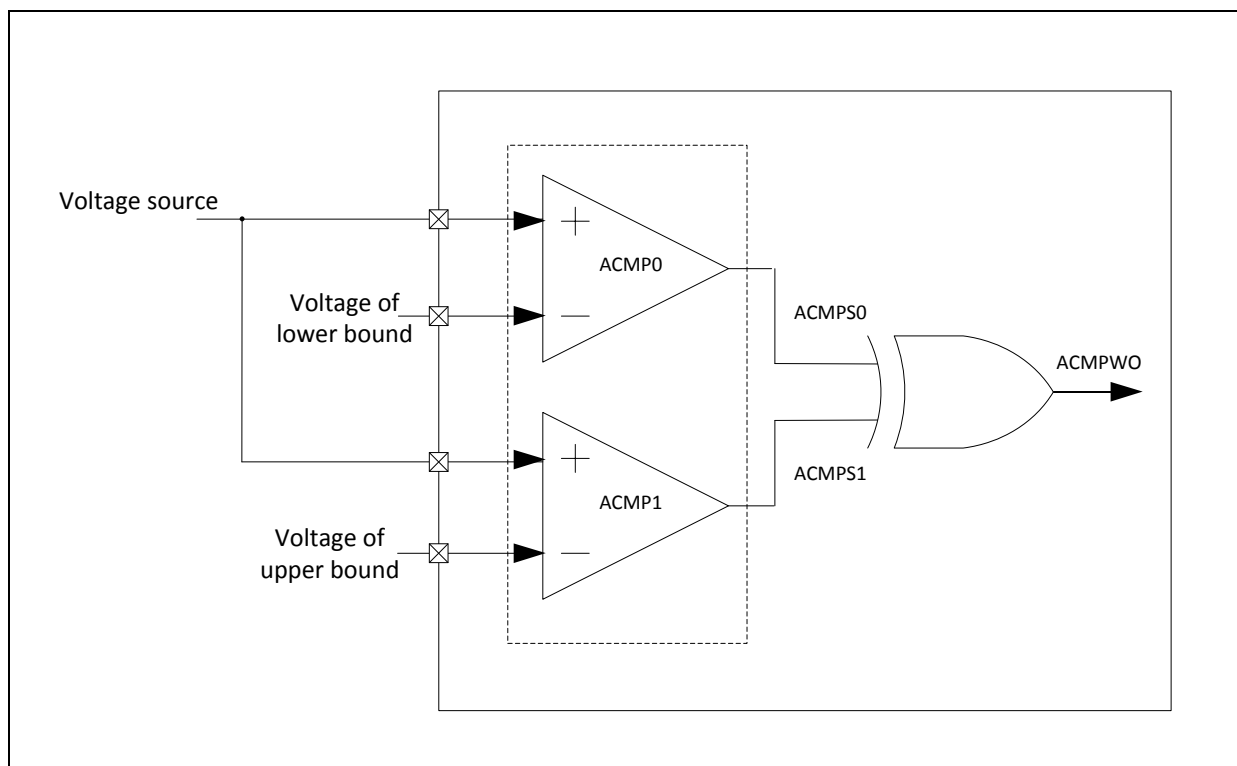


Figure 6.5-7 Example Connection of Window Compare Mode

The comparator window output, ACMPW0 (ACMP\_STATUS[16]), is come from ACMPS0 XOR ACMPS1.

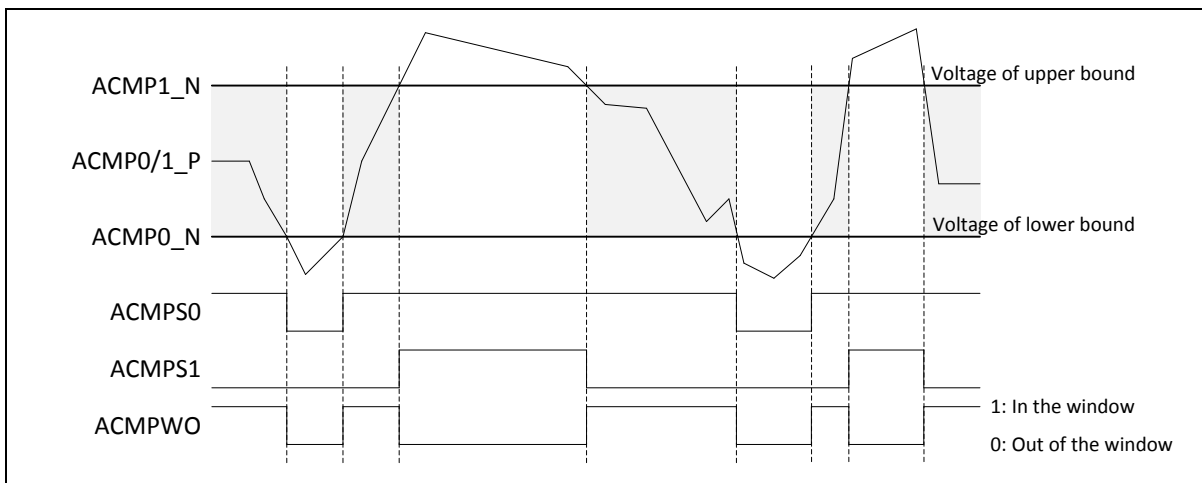


Figure 6.5-8 Example of Window Compare Mode

As shown in Figure 6.5-8, if ACMPW0 equals 1, it means positive input voltage inside the window. Otherwise, the positive input voltage outside the window. Therefore, user can use ACMPW0 to monitor voltage transition of external analog pin. Furthermore, ACMPW0 still can apply to window latch, filter functions and interrupt of ACMP.

**6.5.6 Register Map**

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

Register	Offset	R/W	Description	Reset Value
<b>ACMP Base Address:</b> <b>ACMP01_BA = 0x400D_0000</b>				
<b>ACMP_CTL0</b>	ACMP01_BA+0x00	R/W	Analog Comparator 0 Control Register	0x0000_0000
<b>ACMP_CTL1</b>	ACMP01_BA+0x04	R/W	Analog Comparator 1 Control Register	0x0000_0000
<b>ACMP_STATUS</b>	ACMP01_BA+0x08	R/W	Analog Comparator Status Register	0x0000_0000
<b>ACMP_VREF</b>	ACMP01_BA+0x0C	R/W	Analog Comparator Reference Voltage Control Register	0x0000_0000

6.5.7 Register Description

Analog Comparator 0 Control Register (ACMP\_CTL0)

Register	Offset	R/W	Description	Reset Value
ACMP_CTL0	ACMP01_BA+0x00	R/W	Analog Comparator 0 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved					WCMPSEL	WLATEN	WKEN
15	14	13	12	11	10	9	8
FILTSEL			OUTSEL	Reserved		INTPOL	
7	6	5	4	3	2	1	0
POSSEL		NEGSEL		ACMPOINV	HYSEN	ACMPIE	ACMPEN

Bits	Description	
[31:19]	Reserved	Reserved.
[18]	WCMPSEL	<b>Window Compare Mode Selection</b> 0 = Window Compare Mode Disabled. 1 = Window Compare Mode Selected.
[17]	WLATEN	<b>Window Latch Function Enable Bit</b> 0 = Window Latch Function Disabled. 1 = Window Latch Function Enabled.
[16]	WKEN	<b>Power-down Wake-up Enable Bit</b> 0 = Wake-up function Disabled. 1 = Wake-up function Enabled.
[15:13]	FILTSEL	<b>Comparator Output Filter Count Selection</b> 000 = Filter function Disabled. 001 = ACMP0 output is sampled 1 consecutive PCLK. 010 = ACMP0 output is sampled 2 consecutive PCLKs. 011 = ACMP0 output is sampled 4 consecutive PCLKs. 100 = ACMP0 output is sampled 8 consecutive PCLKs. 101 = ACMP0 output is sampled 16 consecutive PCLKs. 110 = ACMP0 output is sampled 32 consecutive PCLKs. 111 = ACMP0 output is sampled 64 consecutive PCLKs.
[12]	OUTSEL	<b>Comparator Output Selection</b> 0 = Comparator 0 output to ACMP0_O pin is unfiltered comparator output. 1 = Comparator 0 output to ACMP0_O pin is from filter output.
[11:10]	Reserved	Reserved.
[9:8]	INTPOL	<b>Interrupt Condition Polarity Selection</b> ACMPIF0 will be set to 1 when comparator output edge condition is detected.

		00 = Rising edge or falling edge. 01 = Rising edge. 10 = Falling edge. 11 = Reserved.
[7:6]	<b>POSSEL</b>	<b>Comparator Positive Input Selection</b> 00 = Input from ACMP0_P0. 01 = Input from ACMP0_P1. 10 = Input from ACMP0_P2. 11 = Input from ACMP0_P3.
[5:4]	<b>NEGSEL</b>	<b>Comparator Negative Input Selection</b> 00 = ACMP0_N pin. 01 = Internal comparator reference voltage (CRV). 10 = Band-gap voltage. 11 = Reserved.
[3]	<b>ACMPOINV</b>	<b>Comparator Output Inverse</b> 0 = Comparator 0 output inverse Disabled. 1 = Comparator 0 output inverse Enabled.
[2]	<b>HYSEN</b>	<b>Comparator Hysteresis Enable Bit</b> 0 = Comparator 0 hysteresis Disabled. 1 = Comparator 0 hysteresis Enabled.
[1]	<b>ACMPIE</b>	<b>Comparator Interrupt Enable Bit</b> 0 = Comparator 0 interrupt Disabled. 1 = Comparator 0 interrupt Enabled. If WKEN (ACMP_CTL0[16]) is set to 1, the wake-up interrupt function will be enabled as well.
[0]	<b>ACMPEN</b>	<b>Comparator Enable Bit</b> 0 = Comparator 0 Disabled. 1 = Comparator 0 Enabled.

**Analog Comparator 1 Control Register (ACMP\_CTL1)**

Register	Offset	R/W	Description	Reset Value
ACMP_CTL1	ACMP01_BA+0x04	R/W	Analog Comparator 1 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved					WCMPSEL	WLATEN	WKEN
15	14	13	12	11	10	9	8
FILTSEL			OUTSEL	Reserved		INTPOL	
7	6	5	4	3	2	1	0
POSSEL		NEGSEL		ACMPOINV	HYSEN	ACMPIE	ACMPEN

Bits	Description	
[31:19]	Reserved	Reserved.
[18]	WCMPSEL	<b>Window Compare Mode Selection</b> 0 = Window compare mode Disabled. 1 = Window compare mode Selected.
[17]	WLATEN	<b>Window Latch Function Enable Bit</b> 0 = Window Latch function Disabled. 1 = Window Latch function Enabled.
[16]	WKEN	<b>Power-down Wakeup Enable Bit</b> 0 = Wake-up function Disabled. 1 = Wake-up function Enabled.
[15:13]	FILTSEL	<b>Comparator Output Filter Count Selection</b> 000 = Filter function Disabled. 001 = ACMP1 output is sampled 1 consecutive PCLK. 010 = ACMP1 output is sampled 2 consecutive PCLKs. 011 = ACMP1 output is sampled 4 consecutive PCLKs. 100 = ACMP1 output is sampled 8 consecutive PCLKs. 101 = ACMP1 output is sampled 16 consecutive PCLKs. 110 = ACMP1 output is sampled 32 consecutive PCLKs. 111 = ACMP1 output is sampled 64 consecutive PCLKs.
[12]	OUTSEL	<b>Comparator Output Select</b> 0 = Comparator 1 output to ACMP1_O pin is unfiltered comparator output. 1 = Comparator 1 output to ACMP1_O pin is from filter output.
[11:10]	Reserved	Reserved.
[9:8]	INTPOL	<b>Interrupt Condition Polarity Selection</b> ACMPIF1 will be set to 1 when comparator output edge condition is detected.

Bits	Description	
		00 = Rising edge or falling edge. 01 = Rising edge. 10 = Falling edge. 11 = Reserved.
[7:6]	<b>POSSEL</b>	<b>Comparator Positive Input Selection</b> 00 = Input from ACMP1_P0. 01 = Input from ACMP1_P1. 10 = Input from ACMP1_P2. 11 = Input from ACMP1_P3.
[5:4]	<b>NEGSEL</b>	<b>Comparator Negative Input Selection</b> 00 = ACMP1_N pin. 01 = Internal comparator reference voltage (CRV). 10 = Band-gap voltage. 11 = Ground.
[3]	<b>ACMPOINV</b>	<b>Comparator Output Inverse Control</b> 0 = Comparator 1 output inverse Disabled. 1 = Comparator 1 output inverse Enabled.
[2]	<b>HYSEN</b>	<b>Comparator Hysteresis Enable Bit</b> 0 = Comparator 1 hysteresis Disabled. 1 = Comparator 1 hysteresis Enabled.
[1]	<b>ACMPIE</b>	<b>Comparator Interrupt Enable Bit</b> 0 = Comparator 1 interrupt Disabled. 1 = Comparator 1 interrupt Enabled. If WKEN (ACMP_CTL1[16]) is set to 1, the wake-up interrupt function will be enabled as well.
[0]	<b>ACMPEN</b>	<b>Comparator Enable Bit</b> 0 = Comparator 1 Disabled. 1 = Comparator 1 Enabled.

**Analog Comparator Status Register (ACMP\_STATUS)**

Register	Offset	R/W	Description	Reset Value
ACMP_STATUS	ACMP01_BA+0x08	R/W	Analog Comparator Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							ACMPWO
15	14	13	12	11	10	9	8
Reserved		ACMPS1	ACMPS0	Reserved		WKIF1	WKIF0
7	6	5	4	3	2	1	0
Reserved		ACMPO1	ACMPO0	Reserved		ACMPIF1	ACMPIF0

Bits	Description	
[31:17]	Reserved	Reserved.
[16]	ACMPWO	<p><b>Comparator Window Output</b></p> <p>This bit shows the output status of window compare mode</p> <p>0 = The positive input voltage is outside the window.</p> <p>1 = The positive input voltage is in the window.</p>
[15:14]	Reserved	Reserved.
[13]	ACMPS1	<p><b>Comparator 1 Status</b></p> <p>Synchronized to the PCLK to allow reading by software. Cleared when the comparator 1 is disabled, i.e. ACMPEN (ACMP_CTL1[0]) is cleared to 0.</p>
[12]	ACMPS0	<p><b>Comparator 0 Status</b></p> <p>Synchronized to the PCLK to allow reading by software. Cleared when the comparator 0 is disabled, i.e. ACMPEN (ACMP_CTL0[0]) is cleared to 0.</p>
[11:10]	Reserved	Reserved.
[9]	WKIF1	<p><b>Comparator 1 Power-down Wake-up Interrupt Flag</b></p> <p>This bit will be set to 1 when ACMP1 wake-up interrupt event occurs.</p> <p>0 = No power-down wake-up occurred.</p> <p>1 = Power-down wake-up occurred.</p> <p><b>Note:</b> Write 1 to clear this bit to 0.</p>
[8]	WKIF0	<p><b>Comparator 0 Power-down Wake-up Interrupt Flag</b></p> <p>This bit will be set to 1 when ACMP0 wake-up interrupt event occurs.</p> <p>0 = No power-down wake-up occurred.</p> <p>1 = Power-down wake-up occurred.</p> <p><b>Note:</b> Write 1 to clear this bit to 0.</p>
[7:6]	Reserved	Reserved.
[5]	ACMPO1	<p><b>Comparator 1 Output</b></p> <p>Synchronized to the PCLK to allow reading by software. Cleared when the comparator 1</p>

Bits	Description	
		is disabled, i.e. ACPEN (ACMP_CTL1[0]) is cleared to 0.
[4]	<b>ACMPO0</b>	<b>Comparator 0 Output</b> Synchronized to the PCLK to allow reading by software. Cleared when the comparator 0 is disabled, i.e. ACPEN (ACMP_CTL0[0]) is cleared to 0.
[3:2]	<b>Reserved</b>	Reserved.
[1]	<b>ACMPIF1</b>	<b>Comparator 1 Interrupt Flag</b> This bit is set by hardware when the edge condition defined by INTPOL (ACMP_CTL1[9:8]) is detected on comparator 1 output. This will cause an interrupt if ACMPIE (ACMP_CTL1[1]) is set to 1. <b>Note:</b> Write 1 to clear this bit to 0.
[0]	<b>ACMPIF0</b>	<b>Comparator 0 Interrupt Flag</b> This bit is set by hardware when the edge condition defined by INTPOL (ACMP_CTL0[9:8]) is detected on comparator 0 output. This will generate an interrupt if ACMPIE (ACMP_CTL0[1]) is set to 1. <b>Note:</b> Write 1 to clear this bit to 0.



**ACMP Reference Voltage Control Register (ACMP\_VREF)**

Register	Offset	R/W	Description	Reset Value
ACMP_VREF	ACMP01_BA+0x0C	R/W	Analog Comparator Reference Voltage 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	CRVSSEL	Reserved		CRVCTL			

Bits	Description	
[31:7]	Reserved	Reserved.
[6]	CRVSSEL	<b>CRV Source Voltage Selection</b> 0 = AV <sub>DD</sub> is selected as CRV voltage source. 1 = The reference voltage defined by SYS_VREFCTL register is selected as CRV source voltage.
[5:4]	Reserved	Reserved.
[3:0]	CRVCTL	<b>Comparator Reference Voltage Setting</b> CRV = CRV source voltage * (1/6+CRVCTL/24).

## 6.6 Analog-to-Digital Converter (ADC)

### 6.6.1 Overview

The NUC126 series contains one 12-bit successive approximation analog-to-digital converter (SAR A/D converter) with twenty input channels. The A/D converter supports four operation modes: Single, Burst, Single-cycle Scan and Continuous Scan mode. The A/D converter can be started by software, external pin (STADC/PD.2), timer0~3 overflow pulse trigger and PWM trigger.

### 6.6.2 Features

- Analog input voltage range: 0 ~  $AV_{DD}$ .
- 12-bit resolution and 10-bit accuracy is guaranteed
- Up to 20 single-end analog input channels or 10 differential analog input channels
- Maximum ADC peripheral clock frequency is 16 MHz
- Up to 800k SPS sampling rate
- Configurable ADC internal sampling time
- Four operation modes:
  - Single mode: A/D conversion is performed one time on a specified channel.
  - Burst mode: A/D converter samples and converts the specified single channel and sequentially stores the result in FIFO.
  - Single-cycle Scan mode: A/D conversion is performed only one cycle on all specified channels with the sequence from the smallest numbered channel to the largest numbered channel.
  - Continuous Scan mode: A/D converter continuously performs Single-cycle Scan mode until software stops A/D conversion.
- An A/D conversion can be started by:
  - Software Write 1 to ADST bit
  - External pin (STADC)
  - Timer 0~3 overflow pulse trigger
  - PWM trigger with optional start delay period
- Each conversion result is held in data register of each channel with valid and overrun indicators.
- Conversion result can be compared with specified value and user can select whether to generate an interrupt when conversion result matches the compare register setting.
- 3 internal channels, they are band-gap voltage ( $V_{BG}$ ), temperature sensor ( $V_{TEMP}$ ), and Battery power ( $V_{BAT}$ )
- Support PDMA transfer mode.

**Note1:** ADC sampling rate = (ADC peripheral clock frequency) / (total ADC conversion cycle)

**Note2:** If the internal channel ( $V_{TEMP}$ ) is selected to convert, the sampling rate needs to be less than 300k SPS for accurate result.

**Note3:** If the internal channel for band-gap voltage is active, the maximum sampling rate will be 300k SPS.

### 6.6.3 Block Diagram

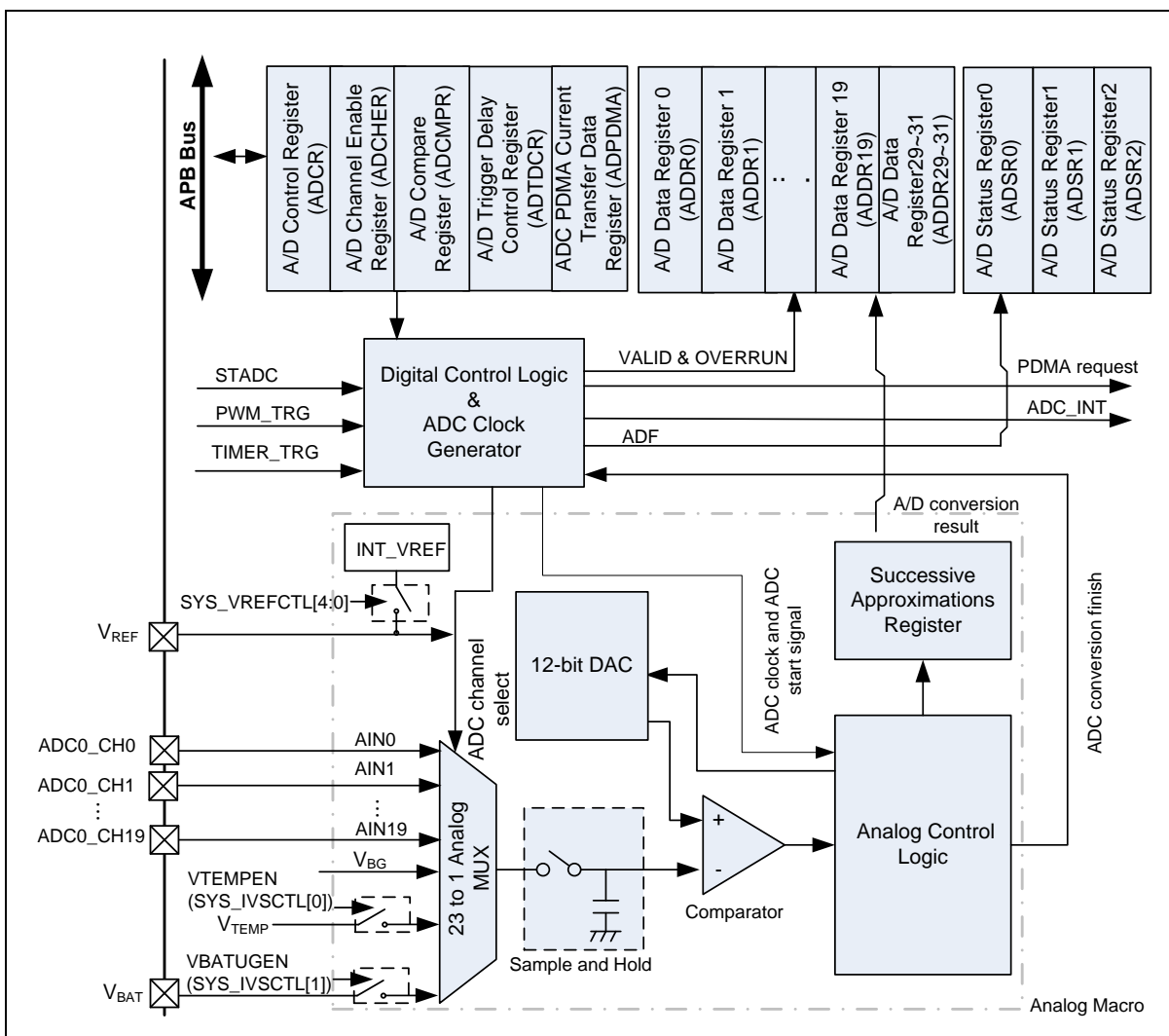


Figure 6.6-1 AD Controller Block Diagram

### 6.6.4 Basic Configuration

- Clock Source Configuration
  - Select the source of ADC peripheral clock on ADCSEL(CLKSEL1[3:2]).
  - Select the clock divider number of ADC peripheral clock on ADCDIV (CLKDIV0[23:16]).
  - Enable ADC peripheral clock in ADCKEN (CLK\_APBCLK0[28]).
- Reset Configuration
  - Reset ADC controller in ADCRST (SYS\_IPRST1[28]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
ADC0	ADC0_CH0	PB.0	MFP1

ADC0_CH1	PB.1	MFP1
ADC0_CH2	PB.2	MFP1
ADC0_CH3	PB.3	MFP1
ADC0_CH4	PB.4	MFP1
ADC0_CH5	PB.8	MFP1
ADC0_CH6	PB.9	MFP1
ADC0_CH7	PB.10	MFP1
ADC0_CH8	PB.11	MFP1
ADC0_CH9	PE.2	MFP1
ADC0_CH10	PB.13	MFP1
ADC0_CH11	PB.14	MFP1
ADC0_CH12	PB.15	MFP1
ADC0_CH13	PB.5	MFP1
ADC0_CH14	PB.6	MFP1
ADC0_CH15	PB.7	MFP1
ADC0_CH16	PC.8	MFP1
ADC0_CH17	PD.8	MFP1
ADC0_CH18	PD.9	MFP1
ADC0_CH19	PD.1	MFP1
ADC0_ST	PD.2	MFP1

### 6.6.5 Functional Description

The A/D converter operates by successive approximation with 12-bit resolution. The ADC has four operation modes: Single, Burst, Single-cycle Scan mode and Continuous Scan mode. When user wants to change the operation mode or analog input channel, in order to prevent incorrect operation, software must clear ADST(ADCR[11]) bit to 0 in advance.

#### 6.6.5.1 ADC peripheral Clock Generator

The maximum sampling rate is up to 800K SPS. The ADC has four clock sources selected by ADCSEL (CLKSEL1[3:2]), the ADC peripheral clock frequency is divided by an 8-bit pre-scalar with the following formula:

$ADC\ peripheral\ clock\ frequency = (ADC\ peripheral\ clock\ source\ frequency) / (ADCDIV+1);$   
 where the 8-bit ADCDIV is located in register CLKDIV0[23:16].

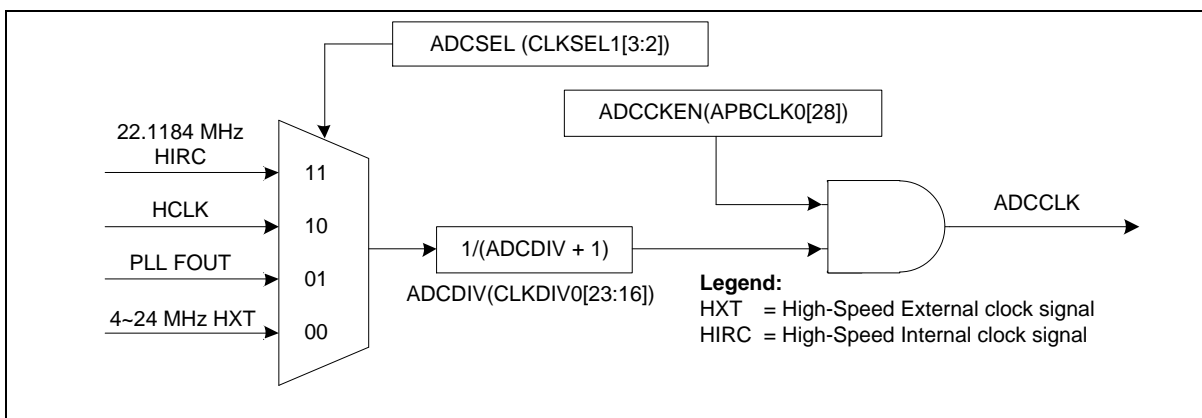


Figure 6.6-2 ADC Peripheral Clock Control

### 6.6.5.2 Single Mode

In Single mode, A/D conversion is performed only once on the specified single channel. The operations are as follows:

1. A/D conversion will be started when the ADST bit of ADCR register is set to 1 by software or external trigger input.
2. When A/D conversion is finished, the result is stored in the ADC data register corresponding to the channel.
3. The ADF bit of ADSR0 register will be set to 1. If the ADIE bit of ADCR register is set to 1, the ADC interrupt will be asserted.
4. The ADST bit remains 1 during A/D conversion. When A/D conversion ends, the ADST bit is automatically cleared to 0 and the A/D converter enters idle state.

**Note1:** If software enables more than one channel in Single mode, only the channel with the smallest number will be selected and the other enabled channels will be ignored.

**Note2:** If ADST bit is cleared to 0 before ADC conversion done, ADC cannot finish the current conversion, the BUSY bit will be cleared to 0 immediately and A/D converter enters idle state directly.

An example timing diagram for Single mode is shown in Figure 6.6-3.

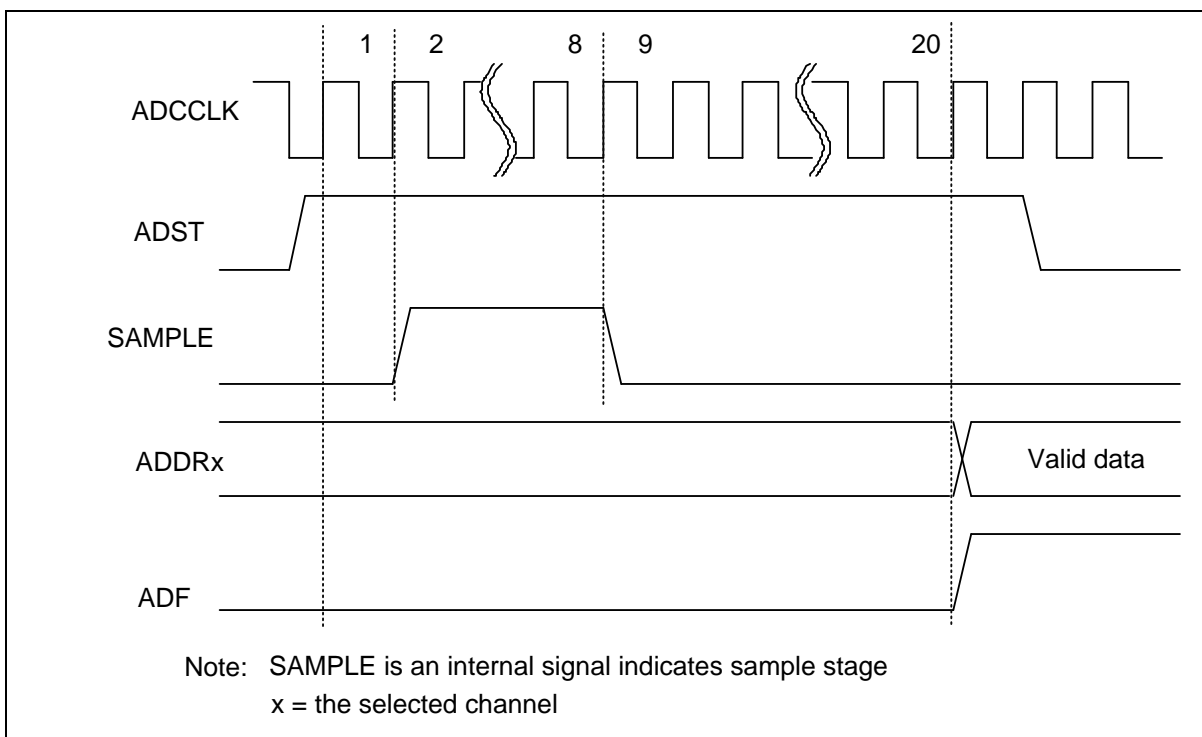


Figure 6.6-3 Single Mode Conversion Timing Diagram

### 6.6.5.3 Burst Mode

In Burst mode, A/D converter samples and converts the specified single channel and sequentially stores the result into FIFO (up to 16 samples). The operations are as follows:

1. When the ADST bit in ADCR register is set to 1 by software or external trigger input, A/D conversion is started on the enabled channel with the smallest number.
2. When A/D conversion for the specified channel is completed, the result is sequentially transferred to FIFO and can be accessed only from the ADC data register 0.
3. When more than or equal to 8 samples in FIFO, the ADF bit in ADSR0 register is set to 1. If the ADIE bit of ADCR register is set to 1 at this time, an ADC interrupt is requested after finishing the A/D conversion.
4. Steps 2 to 3 are repeated as long as the ADST bit remains 1. When the ADST bit is cleared to 0, ADC cannot finish the current conversion and A/D converter enters idle state directly

An example timing diagram for Burst mode is shown in Figure 6.6-4.

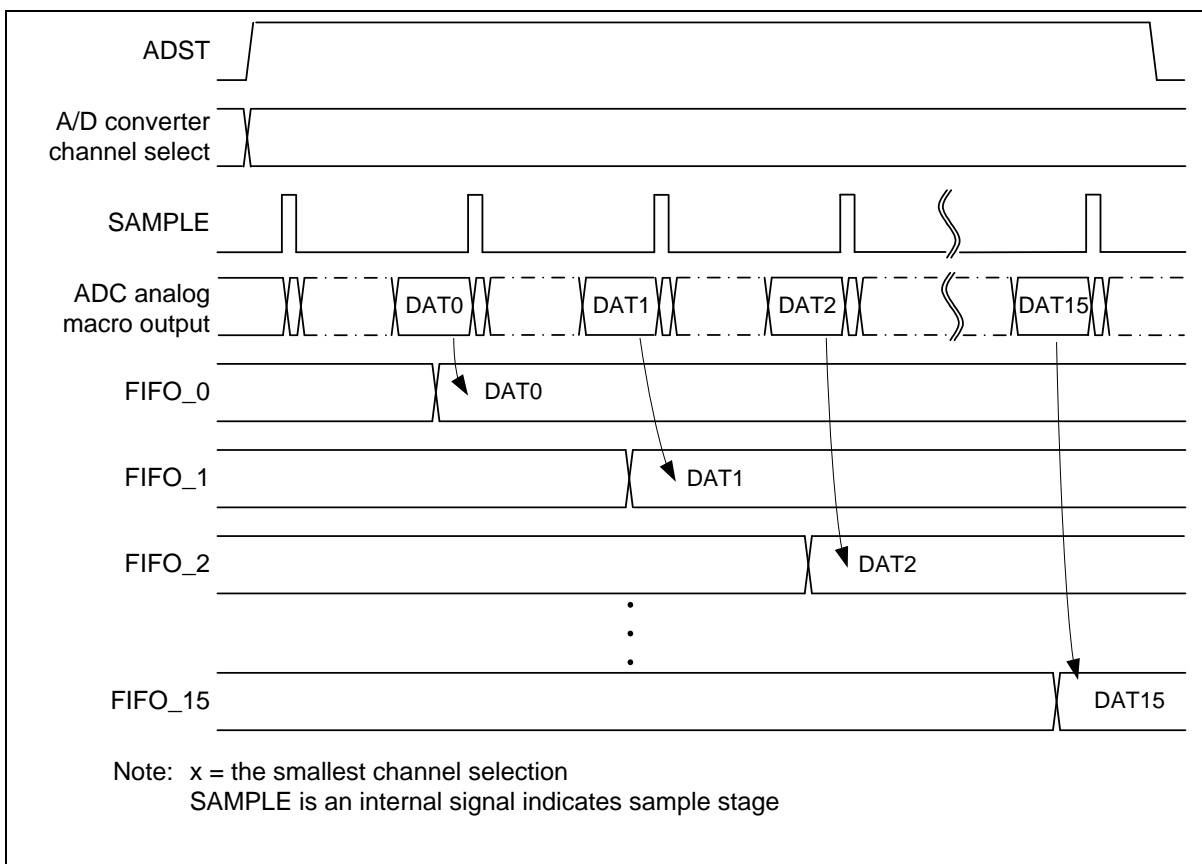


Figure 6.6-4 Burst Mode Conversion Timing Diagram

**Note1:** If software enables more than one channel in Burst mode, only the channel with the smallest number is converted and other enabled channels will be ignored.

**Note2:** User can obtain the conversion results by reading the ADC data register 0 (ADDR0) repeatedly until VALIDF (ADSR0[8]) turns to 0. For example, if there are 8 conversion results at FIFO, it needs to read ADC data register 0 (ADDR0) 8 times to get all of result.

#### 6.6.5.4 Single-Cycle Scan Mode

In Single-cycle Scan mode, A/D converter samples and converts all of the specified channels once in the sequence from the smallest number enabled channel to the largest number enabled channel. Operations are as follows:

1. When the ADST bit in ADCR register is set to 1 by software or external trigger input, A/D conversion is started on the enabled channel with the smallest number.
2. When A/D conversion for each enabled channel is completed, the result is sequentially transferred to the ADC data register corresponding to each channel.
3. When the conversions of all the enabled channels are completed, the ADF bit in ADSR0 register is set to 1. If the ADC interrupt function is enabled, the ADC interrupt occurs.
4. After ADC finishes one cycle conversion, the ADST bit is automatically cleared to 0 and the A/D converter enters idle state. If ADST bit is cleared to 0 before all enabled ADC channels conversion done, ADC cannot finish the current conversion and A/D converter enters idle state directly.

An example timing diagram for Single-cycle Scan mode on enabled channels (0, 2, 3 and 7) is shown in Figure 6.6-5.

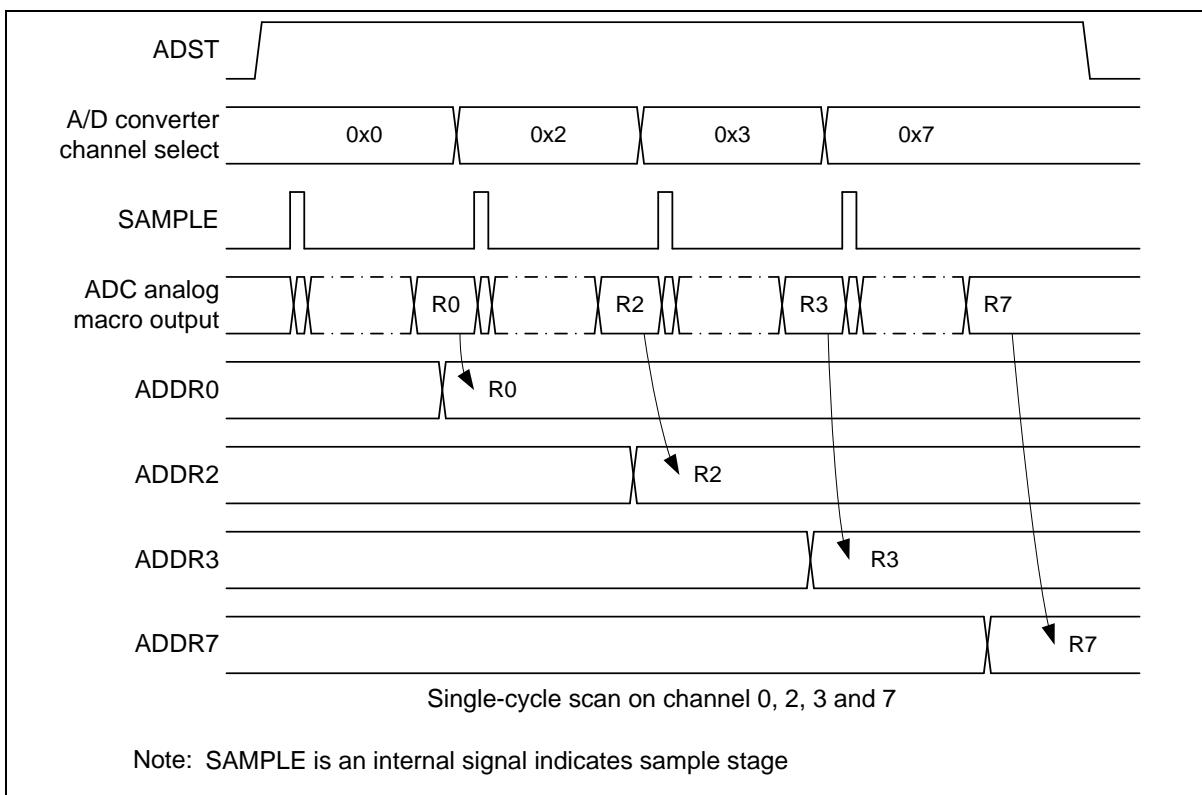


Figure 6.6-5 Single-Cycle Scan Mode on Enabled Channels Timing Diagram

### 6.6.5.5 Continuous Scan Mode

In Continuous Scan mode, A/D conversion is performed sequentially on the specified channels that enabled by CHEN bits in ADCHER register (maximum 20 channels for ADC). The operations are as follows:

1. When the ADST bit in ADCR register is set to 1 by software or external trigger input, A/D conversion is started on the enabled channel with the smallest number.
2. When A/D conversion for each enabled channel is completed, the result of each enabled channel is stored in the ADC data register corresponding to each enabled channel.
3. When A/D converter completes the conversions of all enabled channels sequentially, the ADF bit in ADSR0 register will be set to 1. If the ADC interrupt function is enabled, the ADC interrupt occurs. The conversion of the enabled channel with the smallest number will start again if software does not clear the ADST bit.
4. As long as the ADST bit remains at 1, the step 2 ~ 3 will be repeated. When ADST is cleared to 0, ADC cannot finish the current conversion and A/D converter enters idle state directly.

An example timing diagram for Continuous Scan mode on enabled channels (0, 2, 3 and 7) is shown in Figure 6.6-6.



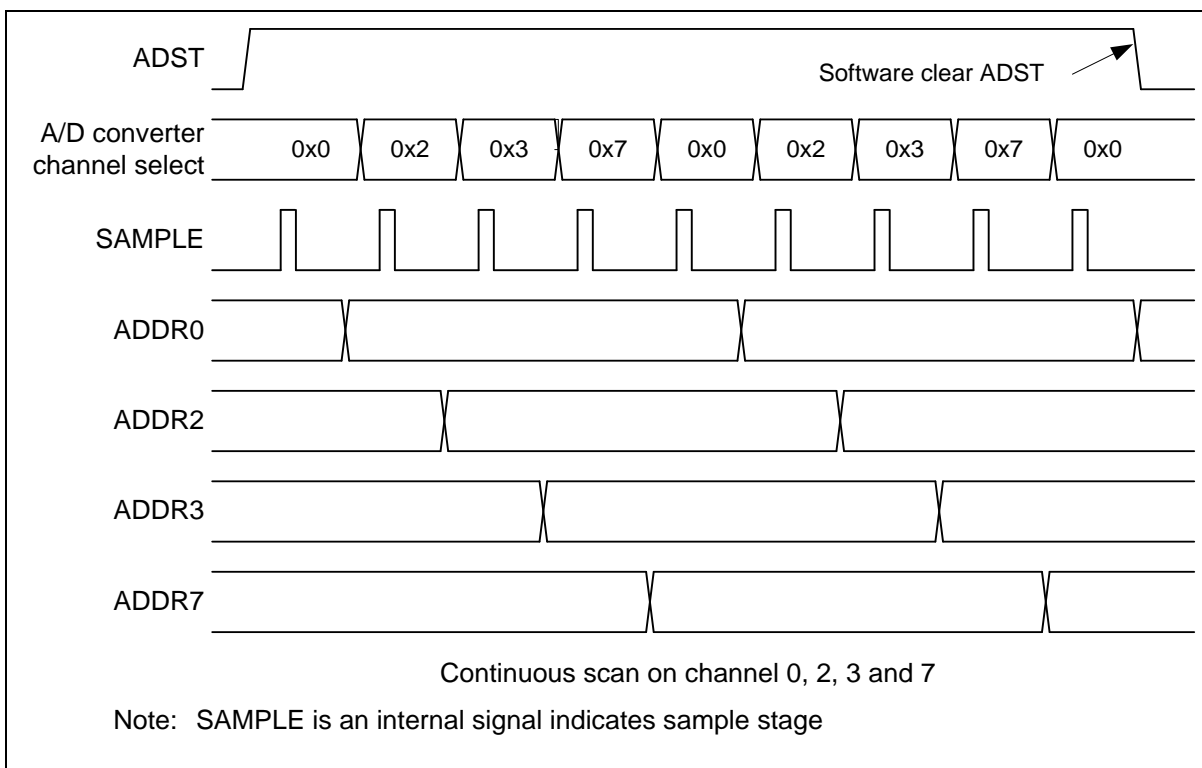


Figure 6.6-6 Continuous Scan Mode on Enabled Channels Timing Diagram

#### 6.6.5.6 External Trigger Input

In Single-cycle Scan mode, A/D conversion can be triggered by external pin request. When the TRGEN bit of ADCR register is set to 1 to enable ADC external trigger function, setting the TRGS(ADCR[5:4]) bits to 2'b00 is to select external trigger input from the STADC pin. Software can set TRGCOND(ADCR[7:6]) to select trigger condition is falling/rising edge or low/high level. If level trigger condition is selected, the STADC pin must be kept at specified state at least 8 PCLKs. The ADST bit will be set to 1 at the 9<sup>th</sup> PCLK and start to convert. Conversion will keep going if external trigger input is kept at active state in level trigger mode. It is stopped only when external condition trigger condition disappears. If edge trigger condition is selected, the high and low state must be kept at least 4 PCLKs. Pulse that is shorter than this specification will be ignored.

**Note:** User enables the external trigger function or enables ADC must be at least 4 PCLKs after enabling ADC peripheral clock.

#### 6.6.5.7 Timer trigger

There are 4 Timer trigger sources for ADC. When the TRGEN bit of ADCR register is set to high to enable ADC external hardware trigger function, setting the TRGS (ADCR[5:4]) bits to 2'b01 is to select external hardware trigger input source from Timer trigger. The detail trigger conditions of Timer are described described at TIMER0\_CTL ~ TIMER3\_CTL register.

#### 6.6.5.8 PWM trigger

In all operation mode, A/D conversion can be triggered by PWM request. When the TRGEN bit of ADCR register is set to high to enable ADC external hardware trigger function, setting the TRGS(ADCR[5:4]) bits to 2'b11 is to select external hardware trigger input source from PWM trigger. When PWM trigger is enabled, setting PTDT (ADTDCR[7:0]) bits can insert a delay time between PWM trigger condition and ADC start conversion.

6.6.5.9 Conversion Result Monitor by Compare Mode Function

The NUC126 series ADC controller provides two compare registers, ADCMPRx(x=0,1), to monitor maximum two specified channels. Software can select which channel to be monitored by setting CMPCH (ADCMPRx[7:3]). CMPCOND (ADCMPRx[2]) bit is used to determine the compare condition. If CMPCOND bit is cleared to 0, the internal match counter will increase one when the conversion result is less than the value specified in CMPD (ADCMPRx[27:16]); if CMPCOND bit is set to 1, the internal match counter will increase one when the conversion result is greater than or equal to the value specified in CMPD (ADCMPRx[27:16]). When the conversion of the channel specified by CMPCH(ADCMPRx[7:3]) is completed, the comparing action will be triggered one time automatically. When the compare result meets the setting, compare match counter will increase 1, otherwise, the compare match counter will be cleared to 0. When the match counter reaches the setting of (CMPMATCNT+1) then CMPF0/1 (ADSR0[1]/[2]) bit will be set to 1, if CMPIE (ADCMPRx[1]) bit is set then an ADC interrupt request is generated. Software can use it to monitor the external analog input pin voltage transition in scan mode without imposing a load on software. The detailed logic diagram is shown in Figure 6.6-7.

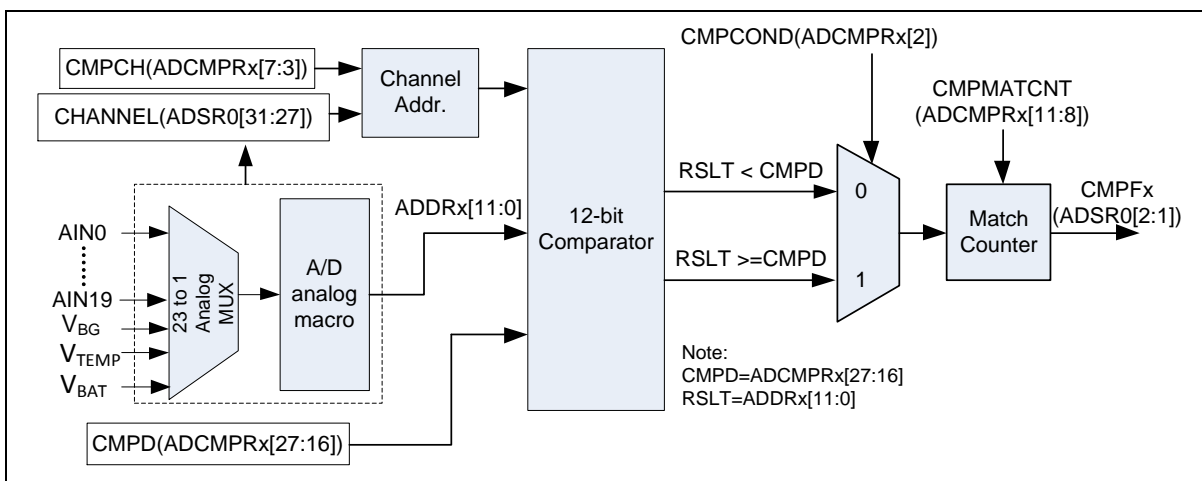


Figure 6.6-7 A/D Conversion Result Monitor Logic Diagram

6.6.5.10 Compare Window Mode

The ADC controller supports a compare window mode. User can set CMPWEN (ADCMPR0[15]) to enable this function. If user enables this function, CMPF0 (ADSR0[1]) will be set only when compared conditions of two conversion result monitor logic are matched and CMPF1 (ADSR0[2]) will always be zero. The range of compare window is between CMPD (ADCMPR0[27:16]) and CMPD (ADCMPR1[27:16]).

6.6.5.11 PDMA Transfer Mode

When A/D conversion is finished, the conversion result will be loaded into ADDRx(x=0~19, 29~31) register and VALID(ADDRx[17]) bit will be set to 1. If the PTEN(ADCR[9]) bit is set, ADC controller will generate a request to PDMA. User can use PDMA to transfer the conversion results to a user-specified memory space without CPU's intervention. The source address of PDMA operation is fixed at ADPDMA, no matter what channels was selected. When PDMA is transferring the conversion result, ADC will continue converting the next selected channel if the operation mode of ADC is burst mode, single scan mode or continuous scan mode. User can monitor current PDMA transfer data through reading ADPDMA register. If ADC completes the conversion of a selected channel and the last conversion result of the same channel has not been transferred by PDMA, OVERRUN(ADSR2[31:0]) bit of the corresponding channel will be set and the last ADC conversion result will be overwritten by the new ADC conversion result. PDMA will transfer the latest data of selected channels to the user-specified destination address.

6.6.5.12 Interrupt Sources

There are three interrupt sources of ADC interrupt. When an ADC operation mode finishes its conversion, the A/D conversion end flag, ADF(ADSR0[0]), will be set to 1. The CMPF0(ADSR0[1]) and CMPF1(ADSR0[2]) are the compare flags of compare function. When the conversion result meets the settings of ADCMPR0/1 registers, the corresponding flag will be set to 1. When one of the flags, ADF, CMPF0 and CMPF1, is set to 1 and the corresponding interrupt enable bit, ADIE of ADCR register and CMPIE of ADCMPR0/1 registers, is set to 1, the ADC interrupt will be asserted. Software can clear these flags to revoke the interrupt request.

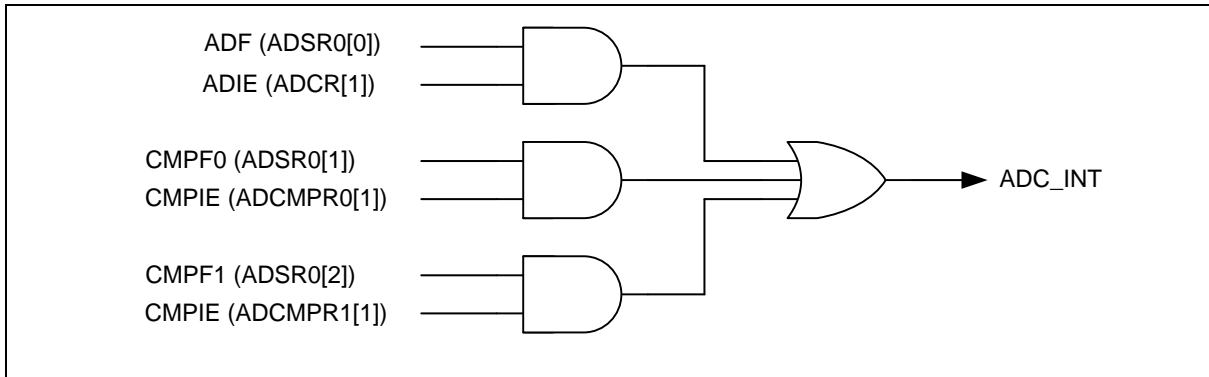


Figure 6.6-8 A/D Controller Interrupt

### 6.6.6 Register Map

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

Register	Offset	R/W	Description	Reset Value
<b>ADC Base Address:</b>				
<b>ADC_BA = 0x400E_0000</b>				
ADC_ADDR0	ADC_BA+0x00	R	ADC Data Register 0	0x0000_0000
ADC_ADDR1	ADC_BA+0x04	R	ADC Data Register 1	0x0000_0000
ADC_ADDR2	ADC_BA+0x08	R	ADC Data Register 2	0x0000_0000
ADC_ADDR3	ADC_BA+0x0C	R	ADC Data Register 3	0x0000_0000
ADC_ADDR4	ADC_BA+0x10	R	ADC Data Register 4	0x0000_0000
ADC_ADDR5	ADC_BA+0x14	R	ADC Data Register 5	0x0000_0000
ADC_ADDR6	ADC_BA+0x18	R	ADC Data Register 6	0x0000_0000
ADC_ADDR7	ADC_BA+0x1C	R	ADC Data Register 7	0x0000_0000
ADC_ADDR8	ADC_BA+0x20	R	ADC Data Register 8	0x0000_0000
ADC_ADDR9	ADC_BA+0x24	R	ADC Data Register 9	0x0000_0000
ADC_ADDR10	ADC_BA+0x28	R	ADC Data Register 10	0x0000_0000
ADC_ADDR11	ADC_BA+0x2C	R	ADC Data Register 11	0x0000_0000
ADC_ADDR12	ADC_BA+0x30	R	ADC Data Register 12	0x0000_0000
ADC_ADDR13	ADC_BA+0x34	R	ADC Data Register 13	0x0000_0000
ADC_ADDR14	ADC_BA+0x38	R	ADC Data Register 14	0x0000_0000
ADC_ADDR15	ADC_BA+0x3C	R	ADC Data Register 15	0x0000_0000
ADC_ADDR16	ADC_BA+0x40	R	ADC Data Register 16	0x0000_0000
ADC_ADDR17	ADC_BA+0x44	R	ADC Data Register 17	0x0000_0000
ADC_ADDR18	ADC_BA+0x48	R	ADC Data Register 18	0x0000_0000
ADC_ADDR19	ADC_BA+0x4C	R	ADC Data Register 19	0x0000_0000
ADC_ADDR29	ADC_BA+0x74	R	ADC Data Register 29	0x0000_0000
ADC_ADDR30	ADC_BA+0x78	R	ADC Data Register 30	0x0000_0000
ADC_ADDR31	ADC_BA+0x7C	R	ADC Data Register 31	0x0000_0000
ADC_ADCR	ADC_BA+0x80	R/W	ADC Control Register	0x0005_0000
ADC_ADCHER	ADC_BA+0x84	R/W	ADC Channel Enable Register	0x0000_0000
ADC_ADCMPRO	ADC_BA+0x88	R/W	ADC Compare Register 0	0x0000_0000
ADC_ADCMPR1	ADC_BA+0x8C	R/W	ADC Compare Register 1	0x0000_0000

<b>ADC_ADSR0</b>	ADC_BA+0x90	R/W	ADC Status Register0	0x0000_0000
<b>ADC_ADSR1</b>	ADC_BA+0x94	R	ADC Status Register1	0x0000_0000
<b>ADC_ADSR2</b>	ADC_BA+0x98	R	ADC Status Register2	0x0000_0000
<b>ADC_ADTDCR</b>	ADC_BA+0x9C	R/W	ADC Trigger Delay Control Register	0x0000_0000
<b>ADC_ADPDMA</b>	ADC_BA+0x100	R	ADC PDMA Current Transfer Data Register	0x0000_0000

6.6.7 Register Description

ADC Data Registers (ADC\_ADDRx x = 0~19, 29~31)

Register	Offset	R/W	Description	Reset Value
ADC_ADDR0	ADC_BA+0x00	R	ADC Data Register 0	0x0000_0000
ADC_ADDR1	ADC_BA+0x04	R	ADC Data Register 1	0x0000_0000
ADC_ADDR2	ADC_BA+0x08	R	ADC Data Register 2	0x0000_0000
ADC_ADDR3	ADC_BA+0x0C	R	ADC Data Register 3	0x0000_0000
ADC_ADDR4	ADC_BA+0x10	R	ADC Data Register 4	0x0000_0000
ADC_ADDR5	ADC_BA+0x14	R	ADC Data Register 5	0x0000_0000
ADC_ADDR6	ADC_BA+0x18	R	ADC Data Register 6	0x0000_0000
ADC_ADDR7	ADC_BA+0x1C	R	ADC Data Register 7	0x0000_0000
ADC_ADDR8	ADC_BA+0x20	R	ADC Data Register 8	0x0000_0000
ADC_ADDR9	ADC_BA+0x24	R	ADC Data Register 9	0x0000_0000
ADC_ADDR10	ADC_BA+0x28	R	ADC Data Register 10	0x0000_0000
ADC_ADDR11	ADC_BA+0x2C	R	ADC Data Register 11	0x0000_0000
ADC_ADDR12	ADC_BA+0x30	R	ADC Data Register 12	0x0000_0000
ADC_ADDR13	ADC_BA+0x34	R	ADC Data Register 13	0x0000_0000
ADC_ADDR14	ADC_BA+0x38	R	ADC Data Register 14	0x0000_0000
ADC_ADDR15	ADC_BA+0x3C	R	ADC Data Register 15	0x0000_0000
ADC_ADDR16	ADC_BA+0x40	R	ADC Data Register 16	0x0000_0000
ADC_ADDR17	ADC_BA+0x44	R	ADC Data Register 17	0x0000_0000
ADC_ADDR18	ADC_BA+0x48	R	ADC Data Register 18	0x0000_0000
ADC_ADDR19	ADC_BA+0x4C	R	ADC Data Register 19	0x0000_0000
ADC_ADDR29	ADC_BA+0x74	R	ADC Data Register 29	0x0000_0000
ADC_ADDR30	ADC_BA+0x78	R	ADC Data Register 30	0x0000_0000
ADC_ADDR31	ADC_BA+0x7C	R	ADC Data Register 31	0x0000_0000

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

Bits	Description	
[31:18]	Reserved	Reserved.
[17]	VALID	<b>Valid Flag (Read Only)</b> This bit will be set to 1 when the conversion of the corresponding channel is completed. This bit will be cleared to 0 by hardware after ADDR register is read. 0 = Data in RSLT bits is not valid. 1 = Data in RSLT bits is valid.
[16]	OVERRUN	<b>Overrun Flag (Read Only)</b> If converted data in RSLT bits has not been read before new conversion result is loaded to this register, OVERRUN bit is set to 1. It is cleared by hardware after ADDR register is read. 0 = Data in RSLT bits is not overwrote. 1 = Data in RSLT bits is overwrote.
[15:0]	RSLT	<b>A/D Conversion Result (Read Only)</b> This field contains conversion result of ADC.

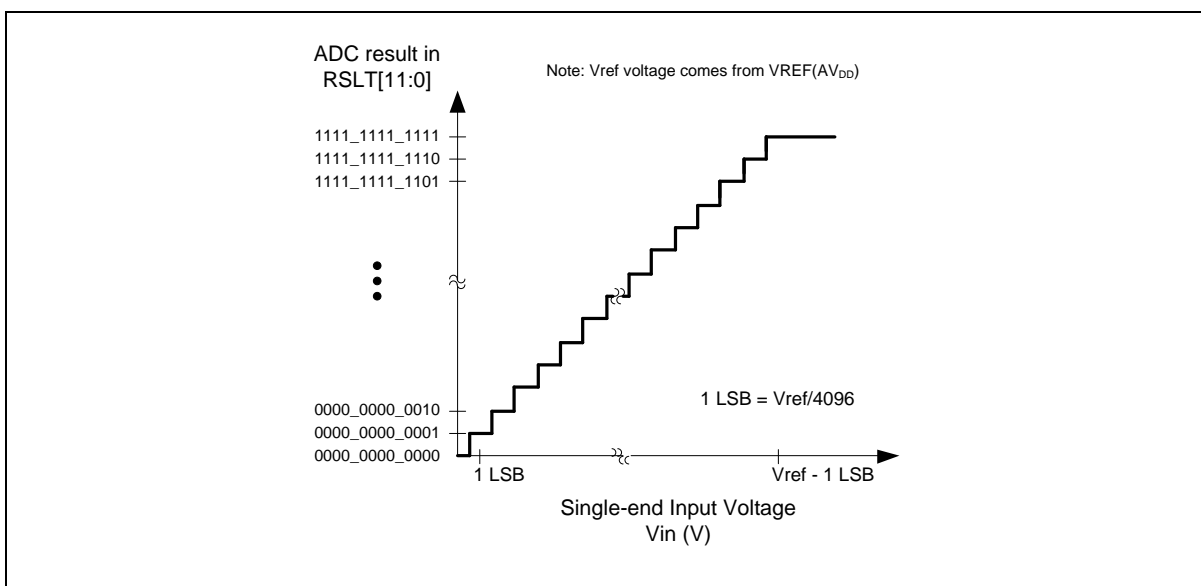


Figure 6.6-9 Conversion Result Mapping Diagram of ADC Single-end Input

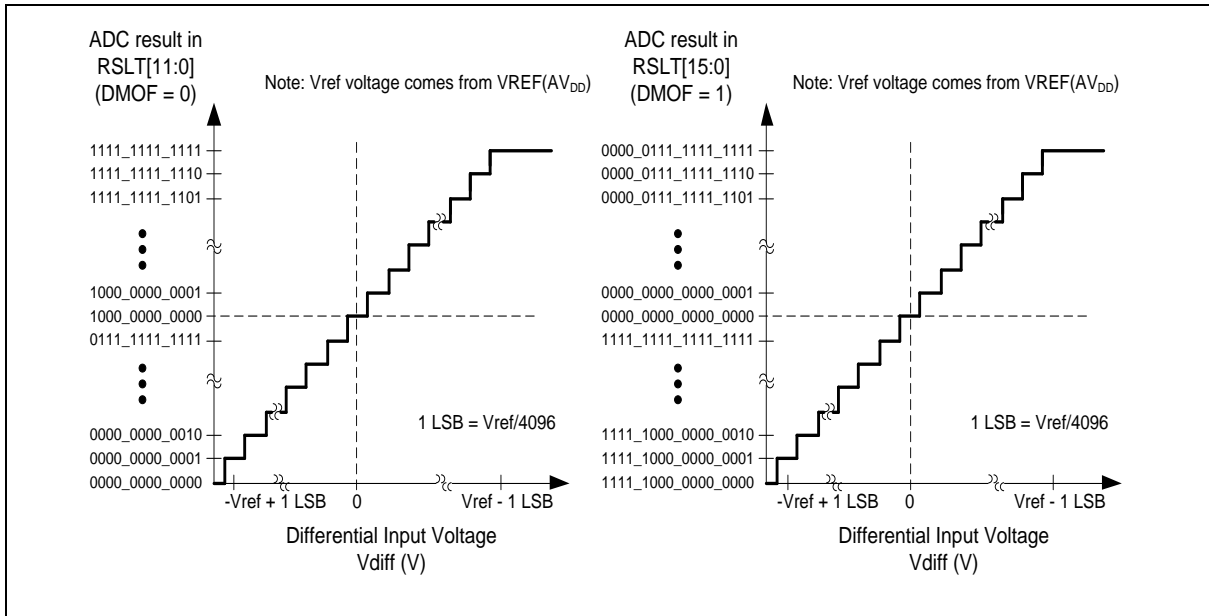


Figure 6.6-10 Conversion Result Mapping Diagram of ADC Differential Input



**ADC Control Register (ADC\_ADCR)**

Register	Offset	R/W	Description	Reset Value
ADC_ADCR	ADC_BA+0x80	R/W	ADC Control Register	0x0005_0000

31	30	29	28	27	26	25	24
DMOF		Reserved					
23	22	21	20	19	18	17	16
Reserved				SMPTSEL			
15	14	13	12	11	10	9	8
Reserved				ADST	DIFFEN	PTEN	TRGEN
7	6	5	4	3	2	1	0
TRGCOND		TRGS		ADMD		ADIE	ADEN

Bits	Description	
[31]	DMOF	<p><b>Differential Input Mode Output Format</b></p> <p>If user enables differential input mode, the conversion result can be expressed with binary straight format (unsigned format) or 2's complement format (signed format).</p> <p>0 = A/D Conversion result will be filled in RSLT at ADDR<sub>x</sub> registers with unsigned format (straight binary format).</p> <p>1 = A/D Conversion result will be filled in RSLT at ADDR<sub>x</sub> registers with 2's complement format.</p>
[30:19]	Reserved	Reserved.
[18:16]	SMPTSEL	<p><b>ADC Internal Sampling Time Selection</b></p> <p>Total ADC conversion cycle = sampling cycle + 12.</p> <p>000 = 4 ADC clock for sampling; 16 ADC clock for complete conversion.</p> <p>001 = 5 ADC clock for sampling; 17 ADC clock for complete conversion.</p> <p>010 = 6 ADC clock for sampling; 18 ADC clock for complete conversion.</p> <p>011 = 7 ADC clock for sampling; 19 ADC clock for complete conversion.</p> <p>100 = 8 ADC clock for sampling; 20 ADC clock for complete conversion.</p> <p>101 = 9 ADC clock for sampling; 21 ADC clock for complete conversion.</p> <p>110 = 10 ADC clock for sampling; 22 ADC clock for complete conversion.</p> <p>111 = 11 ADC clock for sampling; 23 ADC clock for complete conversion.</p>
[15:12]	Reserved	Reserved.
[11]	ADST	<p><b>A/D Conversion Start</b></p> <p>ADST bit can be set to 1 from four sources: software, external pin STADC, PWM trigger and Timer trigger. ADST bit will be cleared to 0 by hardware automatically at the ends of Single mode and Single-cycle Scan mode. In Continuous Scan mode and Burst mode, A/D conversion is continuously performed until software writes 0 to this bit or chip is reset.</p> <p>0 = Conversion stops and A/D converter enters idle state.</p> <p>1 = Conversion starts.</p>
[10]	DIFFEN	<p><b>Differential Input Mode Control</b></p> <p>Differential input voltage (<math>V_{diff}</math>) = <math>V_{plus} - V_{minus}</math>.</p>

		<p>where <math>V_{plus}</math> is the analog input; <math>V_{minus}</math> is the inverted analog input.</p> <p>The <math>V_{plus}</math> of differential input paired channel <math>x</math> is from ADC0_CH<math>y</math> pin; <math>V_{minus}</math> is from ADC0_CH<math>z</math> pin, <math>x=0,1..9</math>, <math>y=2*x</math>, <math>z=y+1</math>.</p> <p>0 = Single-end analog input mode. 1 = Differential analog input mode.</p> <p><b>Note:</b> In Differential Input mode, only the even number of the two corresponding channels needs to be enabled in ADCHER register. The conversion result will be placed to the corresponding data register of the enabled channel.</p>
[9]	PTEN	<p><b>PDMA Transfer Enable Bit</b></p> <p>When A/D conversion is completed, the converted data is loaded into ADDR0~19, ADDR29~ADDR31. Software can enable this bit to generate a PDMA data transfer request.</p> <p>0 = PDMA data transfer Disabled. 1 = PDMA data transfer in ADDR0~19, ADDR29~ADDR31 Enabled.</p> <p><b>Note:</b> When PTEN=1, software must set ADIE=0 to disable interrupt.</p>
[8]	TRGEN	<p><b>External Trigger Enable Bit</b></p> <p>Enable or disable triggering of A/D conversion by external STADC pin, PWM trigger and Timer trigger. If external trigger is enabled, the ADST bit can be set to 1 by the selected hardware trigger source.</p> <p>0 = External trigger Disabled. 1 = External trigger Enabled.</p> <p><b>Note:</b> The ADC external trigger function is only supported in Single-cycle Scan mode.</p>
[7:6]	TRGCOND	<p><b>External Trigger Condition</b></p> <p>These two bits decide external pin STADC trigger event is level or edge. The signal must be kept at stable state at least 8 PCLKs for level trigger and at least 4 PCLKs for edge trigger.</p> <p>00 = Low level. 01 = High level. 10 = Falling edge. 11 = Rising edge.</p>
[5:4]	TRGS	<p><b>Hardware Trigger Source</b></p> <p>00 = A/D conversion is started by external STADC pin. 01 = Timer0 ~ Timer3 overflow pulse trigger. 10 = Reserved. 11 = A/D conversion is started by PWM trigger.</p> <p><b>Note:</b> Software should clear TRGEN bit and ADST bit to 0 before changing TRGS bits.</p>
[3:2]	ADMD	<p><b>A/D Converter Operation Mode Control</b></p> <p>00 = Single conversion. 01 = Burst conversion. 10 = Single-cycle Scan. 11 = Continuous Scan.</p> <p><b>Note1:</b> When changing the operation mode, software should clear ADST bit first. <b>Note2:</b> In Burst mode, the A/D result data is always at ADC Data Register 0.</p>
[1]	ADIE	<p><b>A/D Interrupt Enable Bit</b></p> <p>A/D conversion end interrupt request is generated if ADIE bit is set to 1.</p> <p>0 = A/D interrupt function Disabled. 1 = A/D interrupt function Enabled.</p>
[0]	ADEN	<p><b>A/D Converter Enable Bit</b></p>

		<p>0 = A/D converter Disabled.                  1 = A/D converter Enabled.</p> <p><b>Note:</b> Before starting A/D conversion function, this bit should be set to 1. Clear it to 0 to disable A/D converter analog circuit to save power consumption.</p>
--	--	---

**ADC Channel Enable Register (ADC\_ADCHER)**

Register	Offset	R/W	Description	Reset Value
ADC_ADCHER	ADC_BA+0x84	R/W	ADC Channel Enable Register	0x0000_0000

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

Bits	Description
[31:0]	<p><b>CHEN</b></p> <p><b>Analog Input Channel Enable Control</b></p> <p>Set ADCHER[19:0] bits to enable the corresponding analog input channel 19 ~ 0. If DIFFEN bit is set to 1, only the even number channel needs to be enabled.</p> <p>Besides, set ADCHER[29] to ADCHER[31] bits will enable internal channel for band-gap voltage, temperature sensor and battery power respectively. Other bits are reserved.</p> <p>0 = Channel Disabled. 1 = Channel Enabled.</p> <p><b>Note1:</b> If the internal channel for band-gap voltage (CHEN[29]) is active, the maximum sampling rate will be 300k SPS.</p> <p><b>Note2:</b> If the internal channel for temperature sensor (CHEN[30]) is active, the maximum sampling rate will be 300k SPS.</p>

**ADC Compare Register 0/1 (ADC\_ADCMPR0/1)**

Register	Offset	R/W	Description	Reset Value
ADC_ADCMPR0	ADC_BA+0x88	R/W	ADC Compare Register 0	0x0000_0000
ADC_ADCMPR1	ADC_BA+0x8C	R/W	ADC Compare Register 1	0x0000_0000

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

Bits	Description	
[31:28]	Reserved	Reserved.
[27:16]	CMPD	<p><b>Comparison Data</b> The 12-bit data is used to compare with conversion result of specified channel. <b>Note:</b> CMPD bits should be filled in unsigned format (straight binary format).</p>
[15]	CMPWEN	<p><b>Compare Window Mode Enable Bit</b> 0 = Compare Window Mode Disabled. 1 = Compare Window Mode Enabled. <b>Note:</b> This bit is only presented in ADCMPR0 register.</p>
[14:12]	Reserved	Reserved.
[11:8]	CMPMATCNT	<p><b>Compare Match Count</b> When the specified A/D channel analog conversion result matches the compare condition defined by CMPCOND bit, the internal match counter will increase 1. When the internal counter reaches the value to (CMPMATCNT +1), the CMPFx bit will be set.</p>
[7:3]	CMPCH	<p><b>Compare Channel Selection</b> 00000 = Channel 0 conversion result is selected to be compared. 00001 = Channel 1 conversion result is selected to be compared. 00010 = Channel 2 conversion result is selected to be compared. 00011 = Channel 3 conversion result is selected to be compared. 00100 = Channel 4 conversion result is selected to be compared. 00101 = Channel 5 conversion result is selected to be compared. 00110 = Channel 6 conversion result is selected to be compared. 00111 = Channel 7 conversion result is selected to be compared. 01000 = Channel 8 conversion result is selected to be compared. 01001 = Channel 9 conversion result is selected to be compared.</p>

		<p>01010 = Channel 10 conversion result is selected to be compared.                  01011 = Channel 11 conversion result is selected to be compared.                  01100 = Channel 12 conversion result is selected to be compared.                  01101 = Channel 13 conversion result is selected to be compared.                  01110 = Channel 14 conversion result is selected to be compared.                  01111 = Channel 15 conversion result is selected to be compared.                  10000 = Channel 16 conversion result is selected to be compared.                  10001 = Channel 17 conversion result is selected to be compared.                  10010 = Channel 18 conversion result is selected to be compared.                  10011 = Channel 19 conversion result is selected to be compared.                  11101 = Band-gap voltage conversion result is selected to be compared.                  11110 = Temperature sensor conversion result is selected to be compared.                  11111 = Battery power conversion result is selected to be compared.                  Others = Reserved.</p>
[2]	<b>CMPCOND</b>	<p><b>Compare Condition</b>                  0 = Set the compare condition as that when a 12-bit A/D conversion result is less than the 12-bit CMPD bits, the internal match counter will increase one.                  1 = Set the compare condition as that when a 12-bit A/D conversion result is greater than or equal to the 12-bit CMPD bits, the internal match counter will increase one.  <b>Note:</b> When the internal counter reaches to (CMPMATCNT +1), the CMPFx bit will be set.</p>
[1]	<b>CMPIE</b>	<p><b>Compare Interrupt Enable Bit</b>                  If the compare function is enabled and the compare condition matches the setting of CMPCOND and CMPMATCNT, CMPFx bit will be asserted, in the meanwhile, if CMPIE bit is set to 1, a compare interrupt request is generated.                  0 = Compare function interrupt Disabled.                  1 = Compare function interrupt Enabled.</p>
[0]	<b>CMPEN</b>	<p><b>Compare Enable Bit</b>                  Set this bit to 1 to enable ADC controller to compare CMPD (ADCMPRx[27:16]) with specified channel conversion result when converted data is loaded into ADDR register.                  0 = Compare function Disabled.                  1 = Compare function Enabled.</p>

**ADC Status Register0 (ADC\_ADSR0)**

Register	Offset	R/W	Description	Reset Value
ADC_ADSR0	ADC_BA+0x90	R/W	ADC Status Register0	0x0000_0000

31	30	29	28	27	26	25	24
CHANNEL					Reserved		
23	22	21	20	19	18	17	16
Reserved							OVERRUNF
15	14	13	12	11	10	9	8
Reserved							VALIDF
7	6	5	4	3	2	1	0
BUSY	Reserved				CMPF1	CMPF0	ADF

Bits	Description	
[31:27]	CHANNEL	<b>Current Conversion Channel (Read Only)</b> When BUSY=1, this field reflects current conversion channel. When BUSY=0, it shows the number of the next converted channel.
[26:17]	Reserved	Reserved.
[16]	OVERRUNF	<b>Overrun Flag (Read Only)</b> If any one of OVERRUN (ADDRx[16]) is set, this flag will be set to 1. <b>Note:</b> When ADC is in burst mode and the FIFO is overrun, this flag will be set to 1.
[15:9]	Reserved	Reserved.
[8]	VALIDF	<b>Data Valid Flag (Read Only)</b> If any one of VALID (ADDRx[17]) is set, this flag will be set to 1. <b>Note:</b> When ADC is in burst mode and any conversion result is valid, this flag will be set to 1.
[7]	BUSY	<b>BUSY/IDLE (Read Only)</b> This bit is a mirror of ADST bit in ADCR register. 0 = A/D converter is in idle state. 1 = A/D converter is busy at conversion.
[6:3]	Reserved	Reserved.
[2]	CMPF1	<b>Compare Flag 1</b> When the A/D conversion result of the selected channel meets setting condition in ADCMPR1 register then this bit is set to 1; it is cleared by writing 1 to it 0 = Conversion result in ADDR does not meet ADCMPR1 setting. 1 = Conversion result in ADDR meets ADCMPR1 setting.
[1]	CMPF0	<b>Compare Flag 0</b> When the A/D conversion result of the selected channel meets setting condition in ADCMPR0 register then this bit is set to 1. This bit is cleared by writing 1 to it. 0 = Conversion result in ADDR does not meet ADCMPR0 setting. 1 = Conversion result in ADDR meets ADCMPR0 setting.

[0]	ADF	<p><b>A/D Conversion End Flag</b></p> <p>A status flag that indicates the end of A/D conversion. Software can write 1 to clear this bit.</p> <p>ADF bit is set to 1 at the following three conditions:</p> <ol style="list-style-type: none"> <li>1. When A/D conversion ends in Single mode.</li> <li>2. When A/D conversion ends on all specified channels in Single-cycle Scan mode and Continuous Scan mode.</li> <li>3. When more than or equal to 8 samples in FIFO in Burst mode.</li> </ol>
-----	-----	---



**ADC Status Register1 (ADC\_ADSR1)**

Register	Offset	R/W	Description	Reset Value
ADC_ADSR1	ADC_BA+0x94	R	ADC Status Register1	0x0000_0000

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

Bits	Description
[31:0]	<p><b>VALID</b></p> <p><b>Data Valid Flag (Read Only)</b>                      VALID[31:29, 19:0] are the mirror of the VALID bits in ADDR31[17] ~ ADDR29[17], ADDR19[17]~ ADDR0[17]. The other bits are reserved.  <b>Note:</b> When ADC is in burst mode and any conversion result is valid, VALID[31:29, 19:0] will be set to 1.</p>

**ADC Status Register2 (ADC\_ADSR2)**

Register	Offset	R/W	Description	Reset Value
ADC_ADSR2	ADC_BA+0x98	R	ADC Status Register2	0x0000_0000

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

Bits	Description
[31:0]	<p><b>OVERRUN</b></p> <p><b>Overflow Flag (Read Only)</b>                      OVERRUN[31:29, 19:0] are the mirror of the OVERRUN bit in ADDR31[16] ~ ADDR29[16], ADDR19[16] ~ ADDR0[16]. The other bits are reserved.</p> <p><b>Note:</b> When ADC is in burst mode and the FIFO is overrun, OVERRUN[31:29, 19:0] will be set to 1.</p>

**ADC Trigger Delay Control Register (ADC\_ADTDCR)**

Register	Offset	R/W	Description	Reset Value
ADC_ADTDCR	ADC_BA+0x9C	R/W	ADC Trigger Delay 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
PTDT							

Bits	Description	
[31:8]	Reserved	Reserved.
[7:0]	PTDT	<b>PWM Trigger Delay Time</b> Set this field will delay ADC start conversion time after PWM trigger. PWM trigger delay time is (4 * PTDT) * system clock

**ADC PDMA Current Transfer Data Register (ADC\_ADPDMA)**

Register	Offset	R/W	Description	Reset Value
ADC_ADPDMA	ADC_BA+0x100	R	ADC 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	
[31:18]	Reserved	Reserved.
[17:0]	CURDAT	<p><b>ADC PDMA Current Transfer Data Register (Read Only)</b></p> <p>When PDMA transferring, read this register can monitor current PDMA transfer data. Current PDMA transfer data could be the content of ADDR0 ~ ADDR19 and ADDR29 ~ ADDR31 registers.</p>

## 6.7 CRC Controller (CRC)

### 6.7.1 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.

### 6.7.2 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 program DATA (CRC\_DAT[31:0]) to perform CRC operation

### 6.7.3 Block Diagram

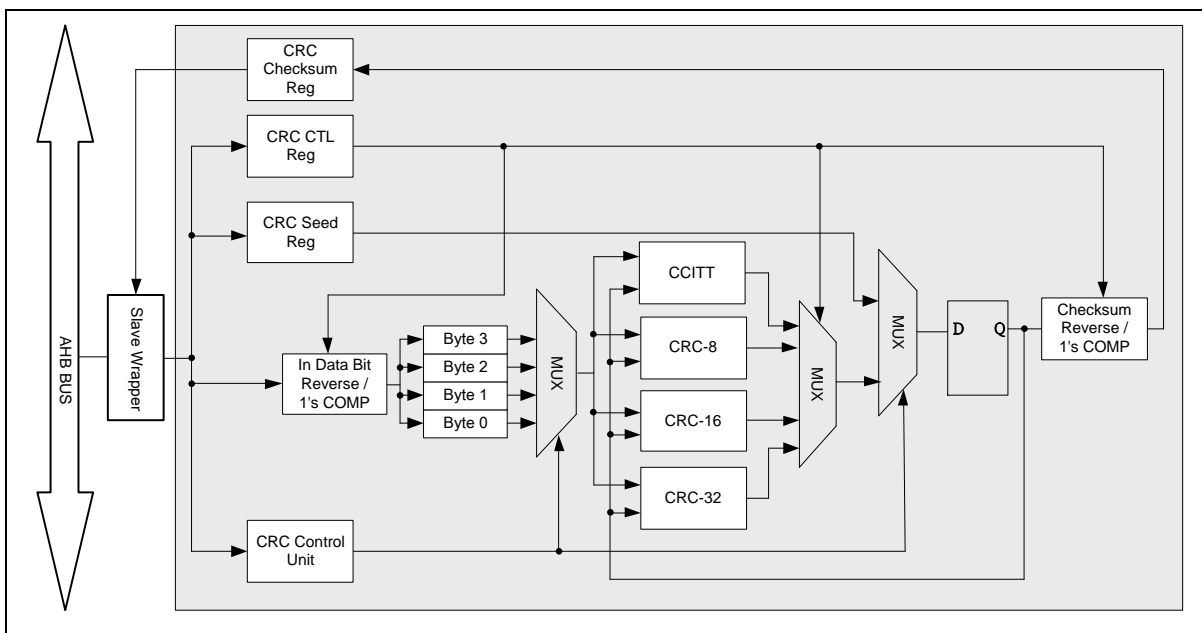


Figure 6.7-1 CRC Generator Block Diagram

### 6.7.4 Basic Configuration

- Clock Source Configuration
  - Enable CRC peripheral clock in CRCKEN (CLK\_AHBCLK[7]).
- Reset Configuration
  - Reset CRC in CRCRST (SYS\_IPRST0[7]).

User can start to program DATA (CRC\_DAT[31:0]) to calculate CRC checksum result in CHECKSUM (CRC\_CHECKSUM[31:0]).

### 6.7.5 Functional Description

The CRC generator can perform CRC calculation with four common polynomial settings which includes CRC-CCITT, CRC-8, CRC-16 and CRC-32. User can choose the CRC operation polynomial mode by setting CRCMODE (CRC\_CTL[31:30]).

The following is an example for program sequence of CRC calculation.

1. Enable CRC generator by setting CRCEN (CRC\_CTL[0]).
2. Initial setting for CRC calculation.
  - Configure 1's complement for CRC checksum by setting CHKSFMT (CRC\_CTL[27]).
  - Configure bit order reverse for CRC checksum by setting CHKSREV (CRC\_CTL[25]). The functional block is also shown in Figure 6.7-2.
  - Configure 1's complement for CRC write data by setting DATFMT (CRC\_CTL[26]).
  - Configure bit order reverse for CRC write data by setting DATREV (CRC\_CTL[24]). The functional block is also shown in Figure 6.7-3.
3. Perform CHKSINIT (CRC\_CTL[1]) to load the initial checksum value CHECKSUM (CRC\_CHECKSUM[31:0]) from SEED (CRC\_SEED[31:0]).
4. Write data to DATA (CRC\_DAT[31:0]) to calculate CRC checksum.
5. Get the CRC checksum result by reading CHECKSUM (CRC\_CHECKSUM[31:0]).

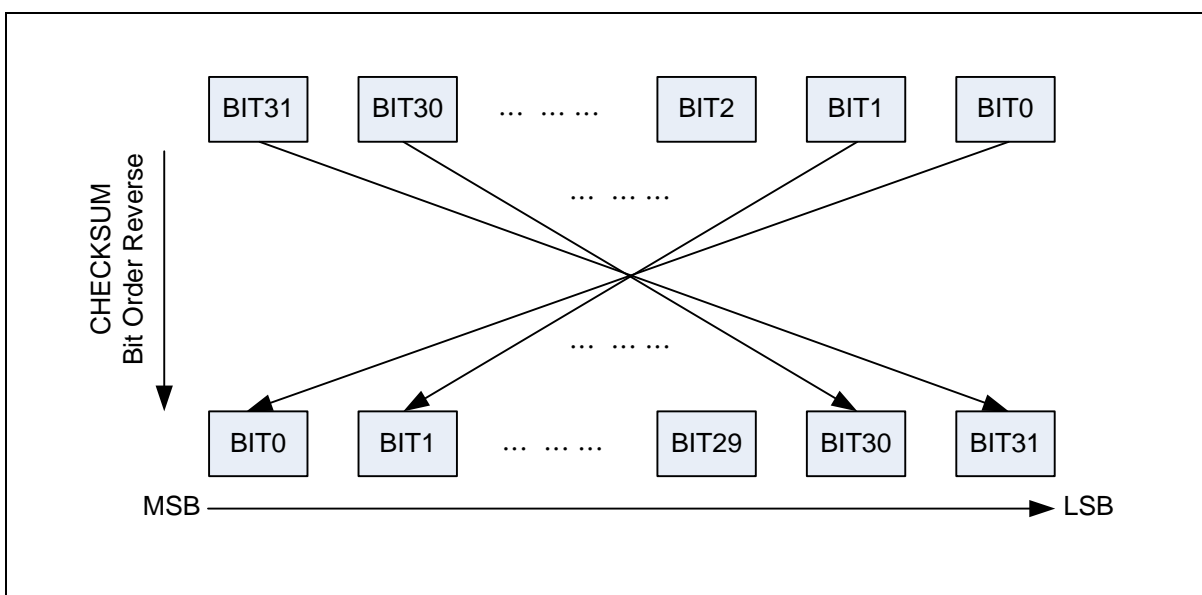


Figure 6.7-2 CHECKSUM Bit Order Reverse Functional Block

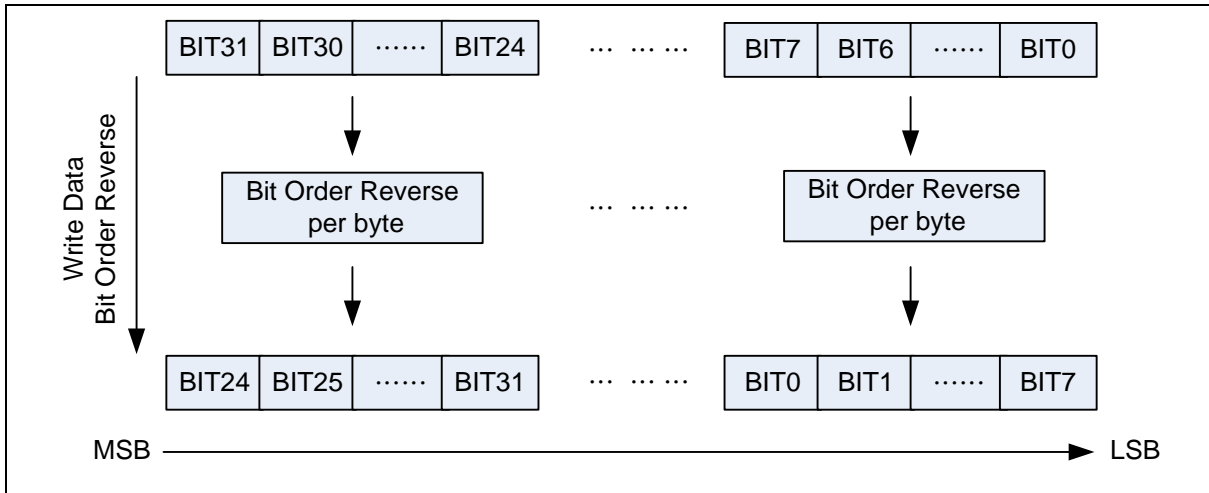


Figure 6.7-3 Write Data Bit Order Reverse Functional Block

**6.7.6 Register Map**

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

Register	Offset	R/W	Description	Reset Value
<b>CRC Base Address:</b> CRC_BA = 0x5001_8000				
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



### 6.7.7 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	CRCEN

Bits	Description
[31:30]	<p><b>CRCMODE</b></p> <p><b>CRC Polynomial Mode</b>                      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.</p>
[29:28]	<p><b>DATLEN</b></p> <p><b>CPU Write Data Length</b>                      This field indicates the valid write data length of DATA (CRC_DAT[31:0]).                      00 = Data length is 8-bit mode.                      01 = Data length is 16-bit mode.                      1x = Data length is 32-bit mode.  <b>Note:</b> 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].</p>
[27]	<p><b>CHKSFMT</b></p> <p><b>Checksum 1's Complement Enable Bit</b>                      This bit is used to enable the 1's complement function for checksum result CHECKSUM (CRC_CHECKSUM[31:0]).                      0 = 1's complement for CRC CHECKSUM Disabled.                      1 = 1's complement for CRC CHECKSUM Enabled.</p>
[26]	<p><b>DATFMT</b></p> <p><b>Write Data 1's Complement Enable Bit</b>                      This bit is used to enable the 1's complement function for write data value DATA (CRC_DATA[31:0]).                      0 = 1's complement for CRC DATA Disabled.                      1 = 1's complement for CRC DATA Enabled.</p>

[25]	<b>CHKSREV</b>	<p><b>Checksum Bit Order Reverse Enable Bit</b></p> <p>This bit is used to enable the bit order reverse function for checksum result CHECKSUM (CRC_CHECKSUM[31:0]).</p> <p>0 = Bit order reverse for CRC CHECKSUM Disabled. 1 = Bit order reverse for CRC CHECKSUM Enabled.</p> <p><b>Note:</b> If the checksum result is 0xDD7B0F2E, the bit order reverse result for CRC checksum is 0x74F0DEBB.</p>
[24]	<b>DATREV</b>	<p><b>Write Data Bit Order Reverse Enable Bit</b></p> <p>This bit is used to enable the bit order reverse function per byte for write data value DATA (CRC_DATA[31:0]).</p> <p>0 = Bit order reversed for CRC DATA Disabled. 1 = Bit order reversed for CRC DATA Enabled (per byte).</p> <p><b>Note:</b> If the write data is 0xAABBCCDD, the bit order reverse for CRC write data in is 0x55DD33BB.</p>
[23:2]	<b>Reserved</b>	Reserved.
[1]	<b>CHKSINIT</b>	<p><b>Checksum Initialization</b></p> <p>Set this bit will auto reolad SEED (CRC_SEED [31:0]) to CHECKSUM (CRC_CHECKSUM[31:0]) as CRC operation initial value.</p> <p>0 = No effect. 1 = Reolad SEED value to CHECKSUM as CRC operation initial value.</p> <p><b>Note:</b> This bit will be cleared automatically.</p>
[0]	<b>CRGEN</b>	<p><b>CRC Generator Enable Bit</b></p> <p>Set this bit 1 to enable CRC generator for CRC operation.</p> <p>0 = No effect. 1 = CRC generator is active.</p>

**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]	DATA	<p><b>CRC Write Data Bits</b></p> <p>User can write data directly by CPU mode or use PDMA function to write data to this field to perform CRC operation.</p> <p><b>Note:</b> 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].</p>

**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]	<p><b>SEED</b></p> <p><b>CRC Seed Value</b> This field indicates the CRC seed value.</p> <p><b>Note1:</b> This SEED value will be loaded to checksum initial value CHECKSUM (CRC_CHECKSUM[31:0]) after set CHKSINIT (CRC_CTL[1]) to 1.</p> <p><b>Note2:</b> The valid bits of CRC_SEED[31:0] is correlated to CRCMODE (CRC_CTL[31:30]).</p>

**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	<p><b>CRC Checksum Results</b>                      This field indicates the CRC checksum result.</p> <p><b>Note:</b> The valid bits of CRC_CHECKSUM[31:0] is correlated to CRCMODE (CRC_CTL[31:30]).</p>

## 6.8 External Bus Interface (EBI)

### 6.8.1 Overview

The NUC126 series is equipped with an external bus interface (EBI) for external device used. To save the connections between external device and the NUC126, EBI operating at address bus and data bus multiplex mode. The EBI supports two chip selects that can connect two external devices with different timing setting requirement.

### 6.8.2 Features

- Supports address bus and data bus multiplex mode to save the address pins
- Supports two chip selects with polarity control
- Supports external devices with maximum 1 MB size for each chip select
- Supports variable external bus base clock (MCLK) which based on HCLK
- Supports 8-bit or 16-bit data width for each chip select
- Supports variable address latch enable time (tALE)
- Supports variable data access time (tACC) and data access hold time (tAHD) for each chip select
- Supports configurable idle cycle for different access condition: Idle of Write command finish (W2X) and Idle of Read-to-Read (R2R)
- Supports continuous data access mode to bypass tASU, tALE and tLHD cycles for improving EBI access

### 6.8.3 Block Diagram

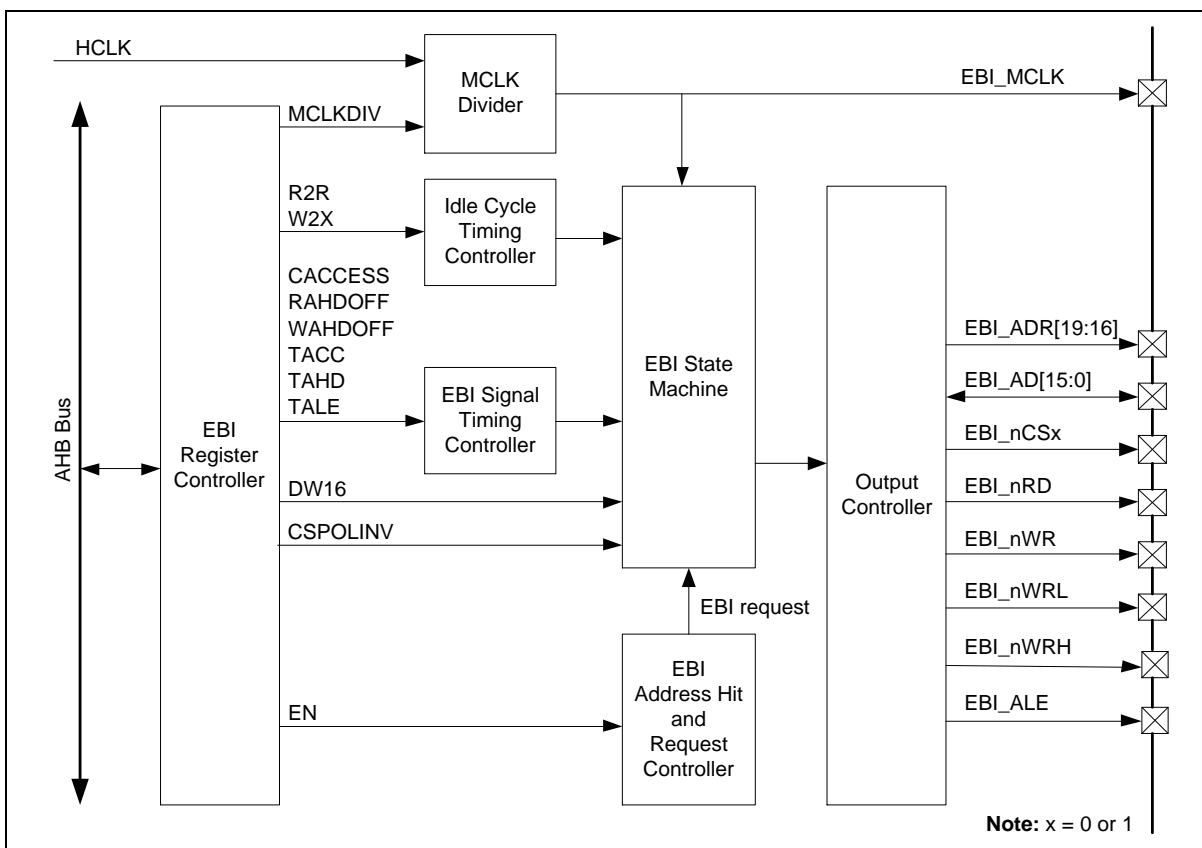


Figure 6.8-1 EBI Block Diagram

### 6.8.4 Basic Configuration

- Clock Source Configuration
  - Enable EBI peripheral clock in CLK\_AHBCLK[3].
- Reset Configuration
  - Reset EBI controller in EBIRST (SYS\_IPRST0[3]).
- Pin Configuration
  - The EBI Controller function pins are configured in SYS\_GPA\_MFPL, SYS\_GPB\_MFPL, SYS\_GPB\_MFPH, SYS\_GPC\_MFPL, SYS\_GPD\_MFPL, SYS\_GPD\_MFPH, SYS\_GPE\_MFPL and SYS\_GPE\_MFPH Multiple Function Control Registers.

Group	Pin Name	GPIO	MFP
EBI	EBI_AD0	PA.0	MFP7
	EBI_AD1	PA.1	MFP7
	EBI_AD2	PA.2	MFP7
	EBI_AD3	PA.3	MFP7
	EBI_AD4	PA.4, PB.7, PE.13	MFP7

EBI_AD5	PA.5, PB.6, PE.12	MFP7
EBI_AD6	PA.6, PB.5, PE.11	MFP7
EBI_AD7	PA.7, PB.4, PE.10	MFP7
EBI_AD8	PC.0	MFP7
EBI_AD9	PC.1	MFP7
EBI_AD10	PC.2	MFP7
EBI_AD11	PC.3	MFP7
EBI_AD12	PC.4	MFP7
EBI_AD13	PC.5	MFP7
EBI_AD14	PC.6	MFP7
EBI_AD15	PC.7	MFP7
EBI_ADR16	PD.12	MFP7
EBI_ADR17	PD.13	MFP7
EBI_ADR18	PD.14	MFP7
EBI_ADR19	PD.15	MFP7
EBI_ALE	PB.3, PD.9, PE.5	MFP7
EBI_MCLK	PD.3	MFP7
EBI_nCS0	PB.2, PD.8, PE.4	MFP7
EBI_nCS1	PB.15, PE.0	MFP7
EBI_nRD	PD.1, PD.7	MFP7
EBI_nWR	PD.2, PD.6	MFP7
EBI_nWRH	PB.1	MFP7
EBI_nWRL	PB.0	MFP7

### 6.8.5 Functional Description

#### 6.8.5.1 EBI Area and Address Hit

The EBI memory mapping address is located at 0x6000\_0000 ~ 0x601F\_FFFF and the total memory space is 2 MB which split into EBI bank0 and bank1. Table 6.8-1 shows the EBI memory mapping of bank0 and bank1. When system request address hit EBI's memory space, the corresponding EBI chip select signal is assert and EBI state machine operates.

Chip Select	Address Mapping
EBI_nCS0	0x6000_0000 ~ 0x600F_FFFF
EBI_nCS1	0x6010_0000 ~ 0x601F_FFFF

Table 6.8-1 EBI Memory Mapping of Bank0 and Bank1



To map the whole EBI memory space, it requires 20-bit address for 8-bit data width device and 19-bit address for 16-bit data width device. For device package that output less than 20-bit address, EBI will map device to mirror space. For example, the external device with 18-bit EBI address on bank0, EBI control will mapped external device to memory address 0x6000\_0000 ~ 0x6003\_FFFF, 0x6004\_0000 ~ 0x6007\_FFFF, 0x6008\_0000 ~ 0x600B\_FFFF and 0x600C\_0000 ~ 0x600F\_FFFF simultaneously.

6.8.5.2 EBI Data Width Connection - Address Bus and Data Bus Multiplex Mode

The EBI controller supports device whose address bus and data bus are multiplexed. For the external device, the connection to device needs additional latch device to latch the address. In this case, EBI\_ALE pin is connected to the latch device to latch the address value, EBI\_AD pins are the input of the latch device, and the output of the latch device is connected to the address of external device.

Figure 6.8-2 shows an example for the connection of 16-bit device with 16-bit data width. For 16-bit device, the EBI\_AD[15:0] shared by Addr[15:0] and 16-bit Data[15:0], EBI\_ADR[18:16] is dedicated for Addr[18:16] and could be connected to 16-bit device directly. The EBI\_ADR[19] will be ignored in this case. Figure 6.8-3 shows an example for the connection of 8-bit device with 8-bit data width. For 8-bit device, only EBI\_AD[7:0] shared by Addr[7:0] and 8-bit Data[7:0], EBI\_AD[15:8] is dedicated for Addr[15:8], EBI\_ADR[19:16] is dedicated for Addr[19:16] and could be connected to 8-bit device directly.

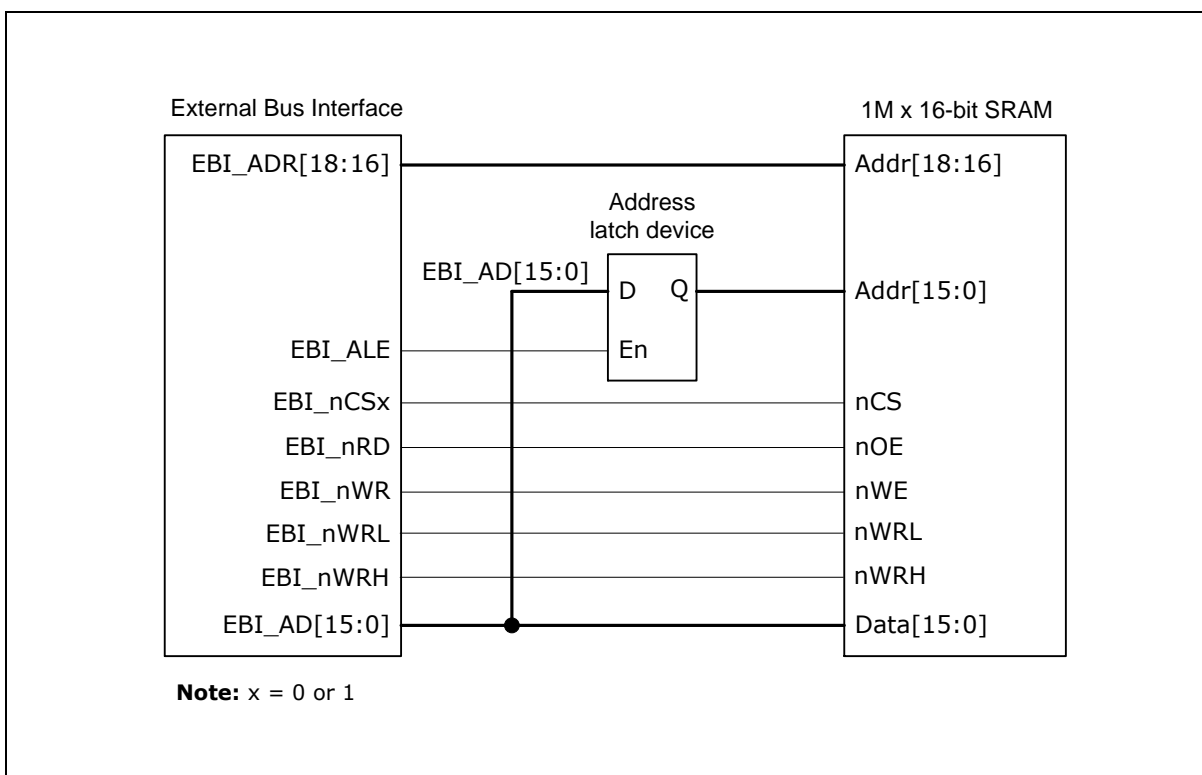


Figure 6.8-2 Connection of 16-bit Device with 16-bit EBI Data Width

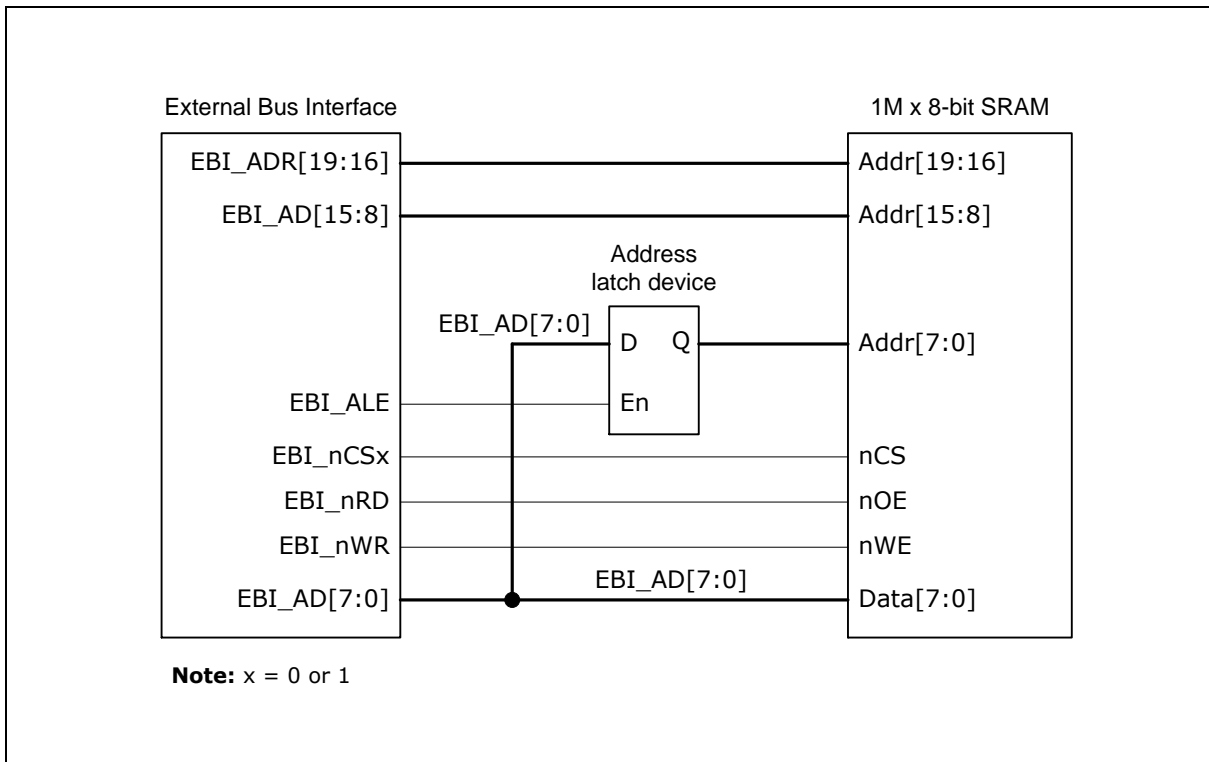


Figure 6.8-3 Connection of 8-bit Device with 8-bit EBI Data Width

When system access data width is larger than EBI data width setting, EBI controller will finish a system access command by operating EBI access with data width setting more than once. For example, if system requests a 32-bit data access through EBI device and data width in EBI controller is 8-bit, EBI controller will operate accessing 8-bit data four times to complete 32-bit data access.

6.8.5.3 EBI Operating Control

**MCLK Control**

In the chip, all EBI signals will be synchronized by EBI\_MCLK when EBI is operating. When chip connects to the external device with slower operating frequency, the EBI\_MCLK can divide most to HCLK/128 by setting MCLKDIV (EBI\_CTLx[10:8]). Therefore, chip can suitable for a wide frequency range of EBI device. If EBI\_MCLK is set to HCLK/1, EBI signals are synchronized by positive edge of EBI\_MCLK, else by negative edge of EBI\_MCLK.

**Operation and Access Timing Control**

At the start of EBI access, chip select (EBI\_nCS0 and EBI\_nCS1) asserts to low if CSPOLINV (EBI\_CTLx[2]) is 0 and wait one EBI\_MCLK for address setup time (tASU) for address stable. Then EBI\_ALE asserts to high after address is stable and keeps for a period of time (tALE) for address latch. After latch address, EBI\_ALE asserts to low and wait one EBI\_MCLK for latch hold time (tLHD) and another one EBI\_MCLK cycle (tA2D) that is inserted behind address hold time to be the bus turn-around time for address change to data. Then EBI\_nRD asserts to low when read access or EBI\_nWR asserts to low when write access. Then EBI\_nRD or EBI\_nWR asserts to high after keeps access time (tACC) for reading output stable or writing finish. After that, EBI signals keep for data access hold time (tAHD) and chip select asserts to high, address is released by current access control. Figure 6.8-4 and Figure 6.8-5 shows the timing control waveform for 16-bit data width and 8-bit data width.

The EBI controller provides a flexible timing control for different external device. In EBI timing control, tASU, tLHD and tA2D are fixed to one EBI\_MCLK cycle, tAHD can modulate to 1~8 EBI\_MCLK cycles by setting TAHD (EBI\_TCTLx[10:8]), tACC can modulate to 1~32 EBI\_MCLK cycles by setting TACC (EBI\_TCTLx[7:3]), and tALE can modulate to 1~8 EBI\_MCLK cycles by setting TALE (EBI\_CTL0[18:16]) only. Some external device can support zero data access hold time accessing, the EBI controller can skipped tAHD cycles to increase access speed by setting WAHDOFF (EBI\_TCTLx[23]) and RAHDOFF (EBI\_TCTLx[22]).

For each chip select, the EBI provide individual register for timing control except tALE can only be controlled by EBI\_CTL0. Table 6.8-2 shows the timing control setting of EBI controller.

Parameter	Value	Unit	Description
tASU	1	MCLK	Address Latch Setup Time.
tALE	1 ~ 8	MCLK	ALE High Period. Controlled by TALE (EBI_CTL0[18:16]).
tLHD	1	MCLK	Address Latch Hold Time.
tA2D	1	MCLK	Address To Data Delay (Bus Turn-Around Time).
tACC	1 ~ 32	MCLK	Data Access Time. Controlled by TACC (EBI_TCTLx[7:3]).
tAHD	1 ~ 8	MCLK	Data Access Hold Time. Controlled by TAHD (EBI_TCTLx[10:8]).
IDLE	0 ~ 15	MCLK	Idle Cycle. Controlled by R2R (EBI_TCTLx[27:24]) and W2X (EBI_TCTLx[15:12]).

Table 6.8-2 EBI Timing Control Setting Table

Figure 6.8-4 shows an example of setting 16-bit data width by 16-bit read/write access commands. In this example, EBI\_AD[15:0] bus is used for being Address output[15:0] and Data[15:0]. When EBI\_ALE asserts to high, EBI\_AD is address output. After address is latched, EBI\_ALE asserts to low and the EBI\_AD bus change to high impedance to wait device output data in read access operation, or it is used for being write data output.

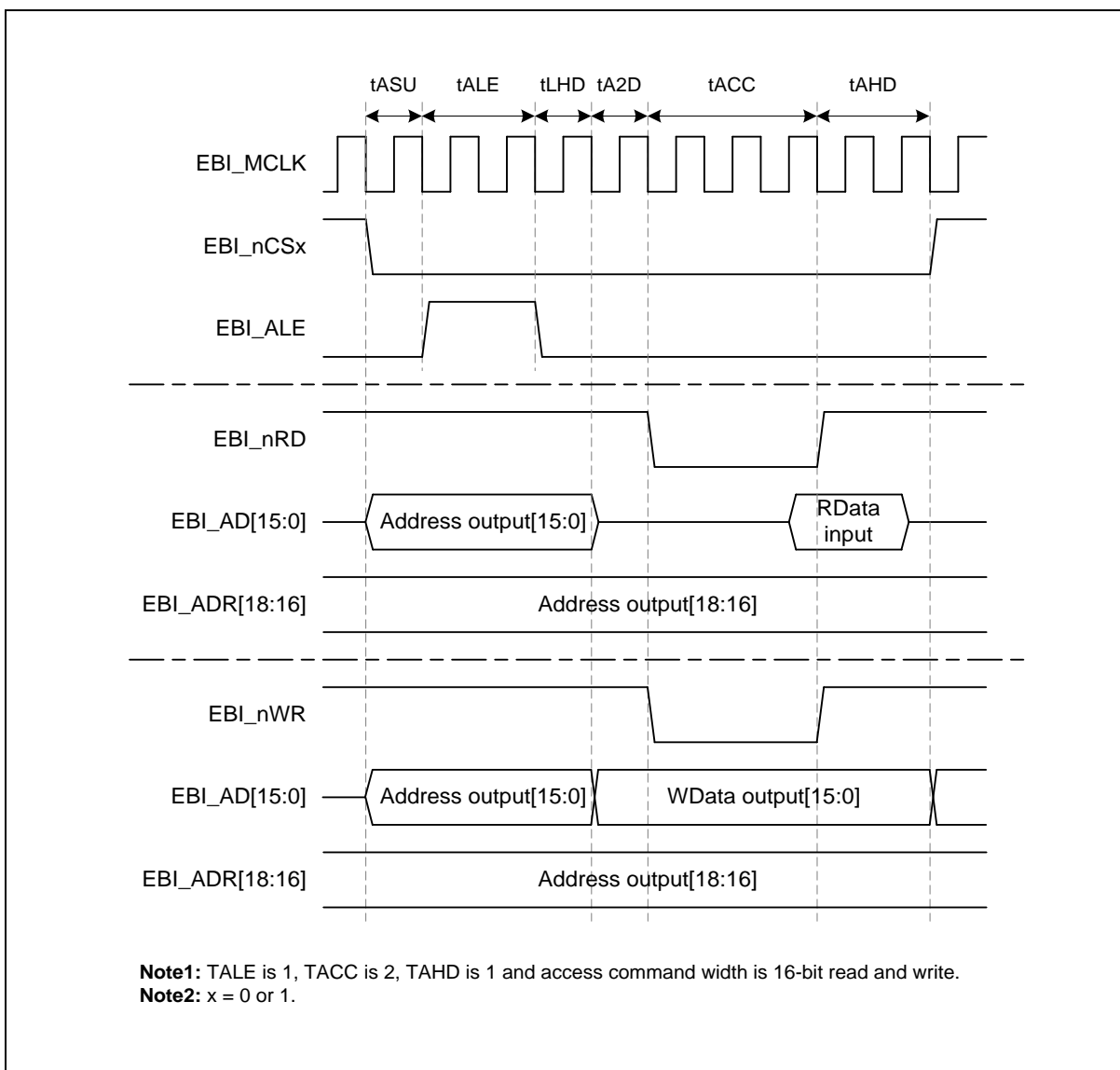


Figure 6.8-4 Timing Control Waveform for 16-bit Data Width

Figure 6.8-5 shows an example of setting 8-bit data width by 8-bit read/write access commands. The difference between 8-bit and 16-bit data width is EBI\_AD[15:8]. In 8-bit data width setting, EBI\_AD[15:8] always be Address output[15:8] so that external latch need only 8-bit width.

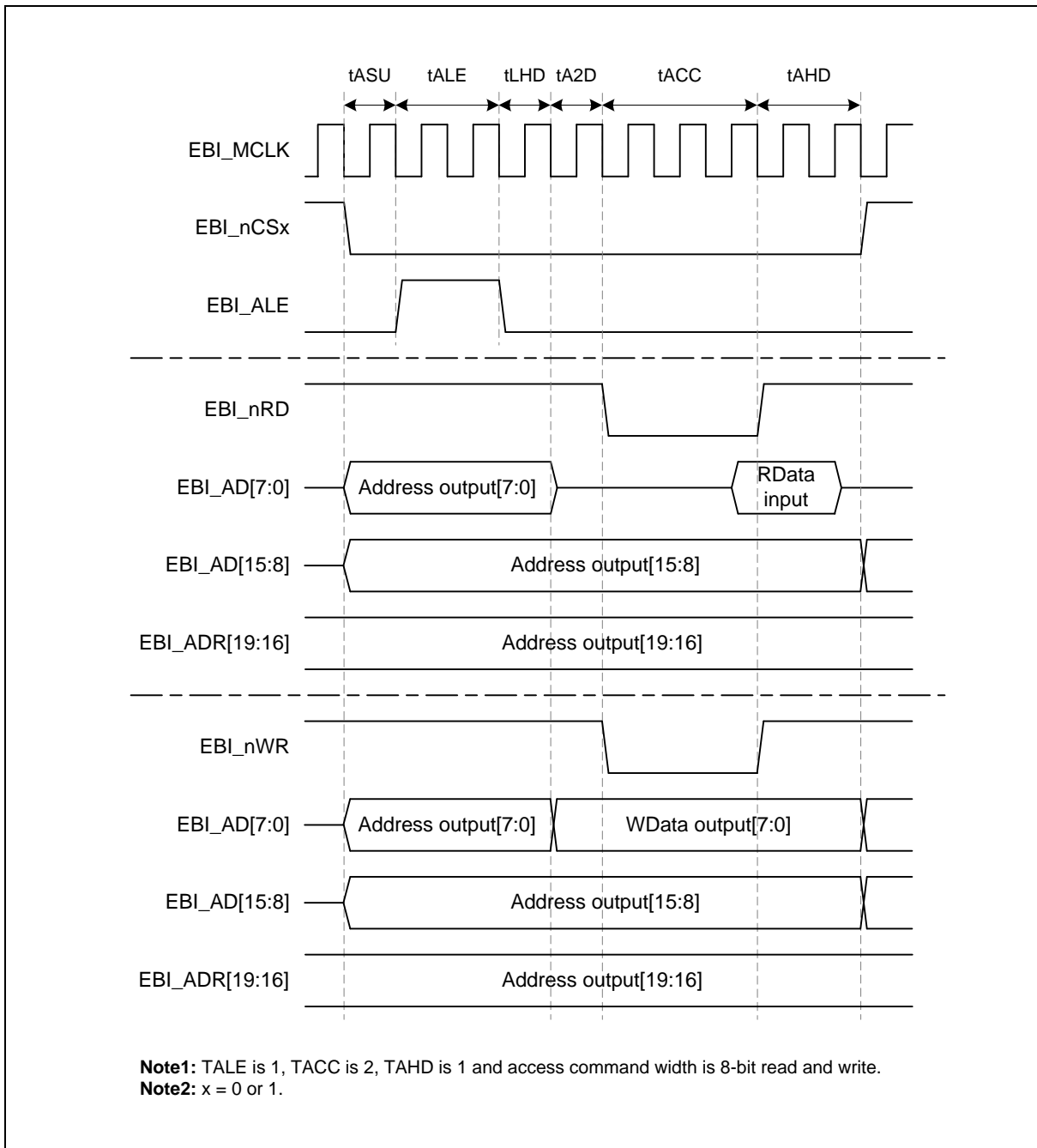


Figure 6.8-5 Timing Control Waveform for 8-bit Data Width

### Insert Idle Cycle

When EBI accessing continuously, there may occur bus conflict if the device access time is much slow with system operating. EBI controller supply additional idle cycle as shown in Figure 6.8-6 to solve this problem. During idle cycle, all control signals of EBI are inactive. There are two conditions that EBI controller can insert idle cycle by timing control after write access, W2X (EBI\_TCTLx[15:12]) and after read access and before next read access, R2R (EBI\_TCTLx[27:24]).

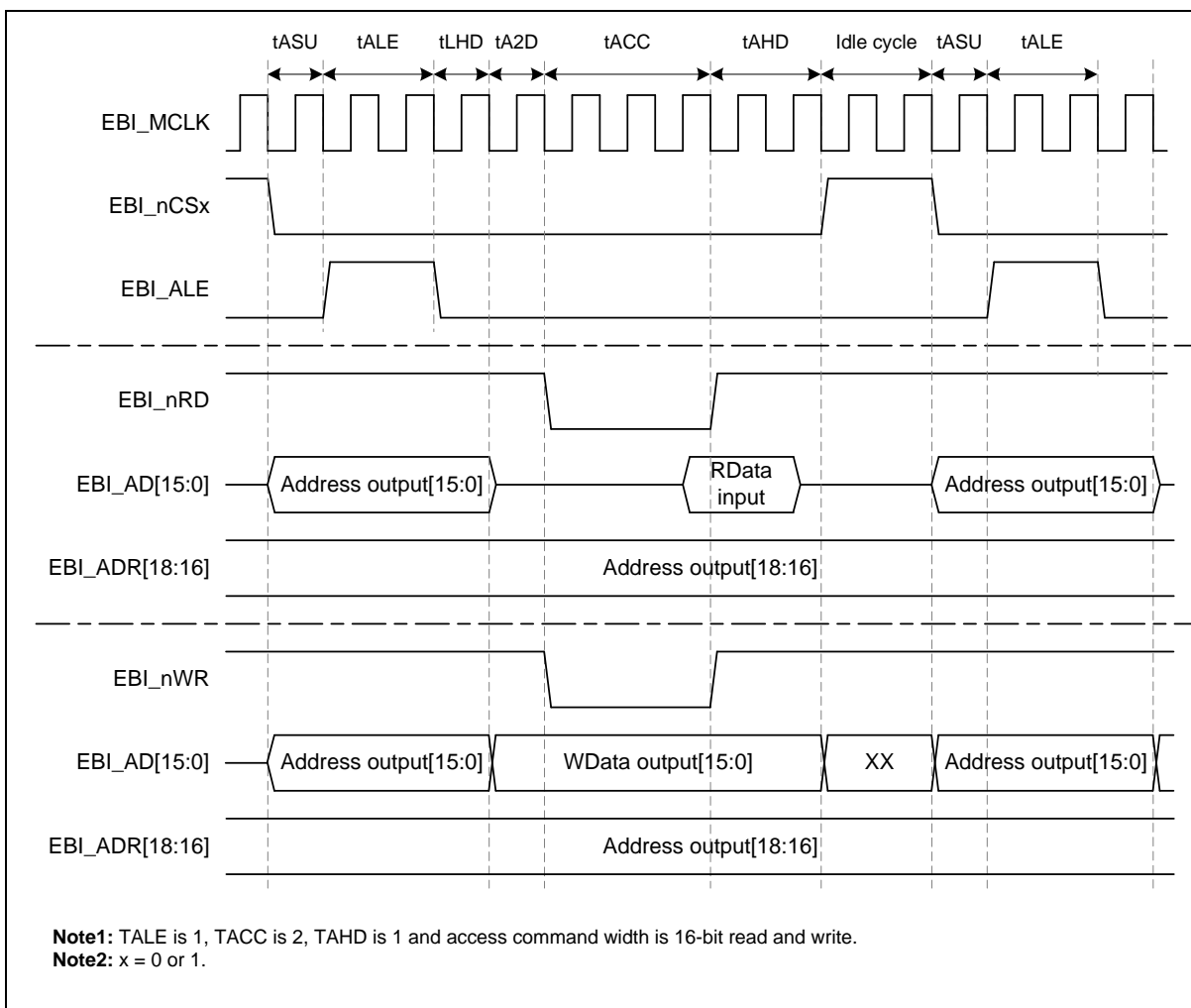


Figure 6.8-6 Timing Control Waveform for Insert Idle Cycle

### Chip Select Polarity Control

EBI controller supports chip select polarity control for connecting to variable external device. When CSPOLINV (EBI\_CTLx[2]) is set to 0, the chip select pins (EBI\_nCSx) works as low active behavior. It means the external device can be access only under EBI\_nCSx at low state. When CSPOLINV (EBI\_CTLx[2]) is set to 1, the chip select pin (EBI\_nCSx) works as high active behavior. It means the external device can be access only under EBI\_nCSx at high state.

### Continuous Data Access Mode

The EBI controller supports continuous data access mode for the device which needs faster data access and do not need address control interface. User can enable this mode by setting CACCESS (EBI\_CTLx[4]) to 1 for each bank. When EBI continuous data access mode is enabled, the tASU, tALE, tLHD cycles are ignored and EBI can access data continuously within one read or write command. The timing waveform of continuous data access mode is shown as Figure 6.8-7.

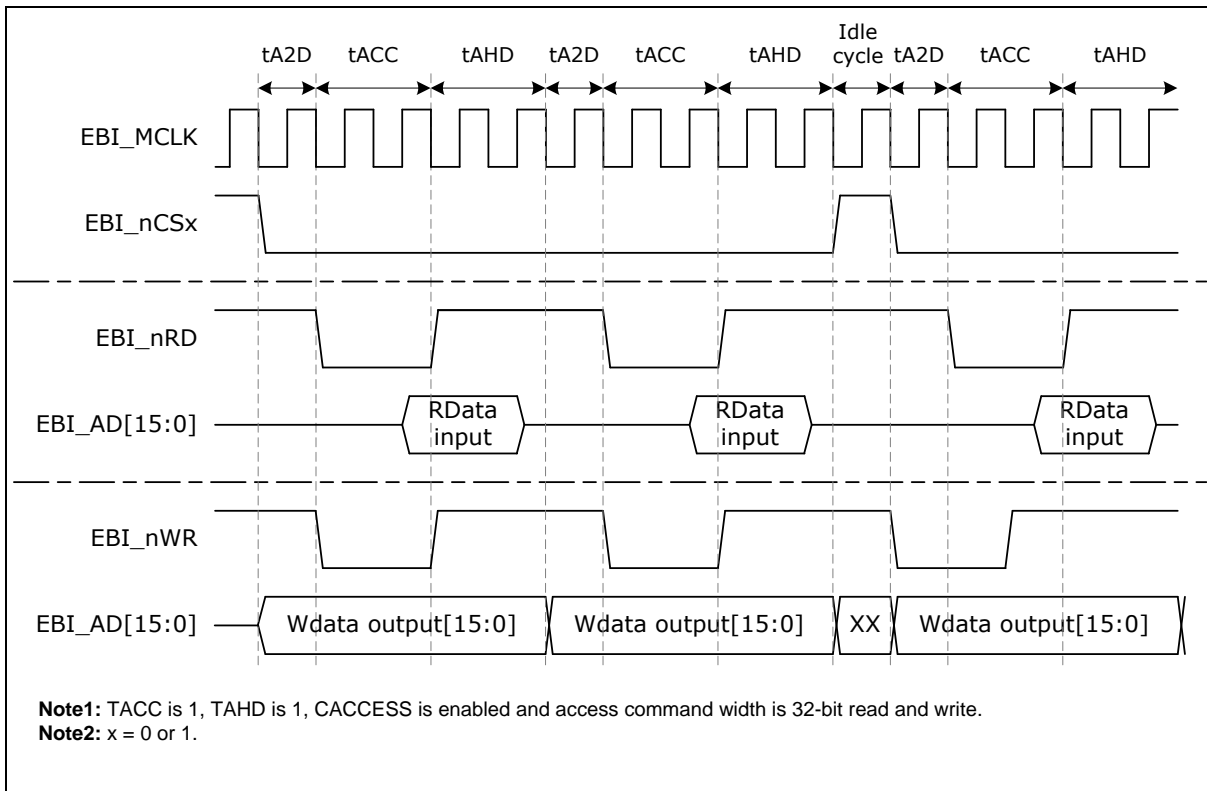


Figure 6.8-7 Timing Control Waveform for Continuous Data Access Mode

**6.8.6 Register Map**

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

Register	Offset	R/W	Description	Reset Value
<b>EBI Base Address:</b>				
<b>EBI_BA = 0x5001_0000</b>				
<b>EBI_CTL0</b>	EBI_BA+0x00	R/W	External Bus Interface Bank0 Control Register	0x0000_0000
<b>EBI_TCTL0</b>	EBI_BA+0x04	R/W	External Bus Interface Bank0 Timing Control Register	0x0000_0000
<b>EBI_CTL1</b>	EBI_BA+0x10	R/W	External Bus Interface Bank1 Control Register	0x0000_0000
<b>EBI_TCTL1</b>	EBI_BA+0x14	R/W	External Bus Interface Bank1 Timing Control Register	0x0000_0000



6.8.7 Register Description

External Bus Interface Control Register (EBI\_CTLx)

Register	Offset	R/W	Description	Reset Value
EBI_CTL0	EBI_BA+0x00	R/W	External Bus Interface Bank0 Control Register	0x0000_0000
EBI_CTL1	EBI_BA+0x10	R/W	External Bus Interface Bank1 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reversed							Reserved
23	22	21	20	19	18	17	16
Reversed					TALE		
15	14	13	12	11	10	9	8
Reversed					MCLKDIV		
7	6	5	4	3	2	1	0
Reversed			CACCESS	Reversed	CSPOLINV	DW16	EN

Bits	Description	
[31:19]	Reserved	Reserved.
[18:16]	TALE	<p><b>Extend Time of ALE</b></p> <p>The EBI_ALE high pulse period (tALE) to latch the address can be controlled by TALE.</p> $tALE = (TALE + 1) * EBI\_MCLK.$ <p><b>Note:</b> This field only available in EBI_CTL0 register</p>
[15:11]	Reserved	Reserved.
[10:8]	MCLKDIV	<p><b>External Output Clock Divider</b></p> <p>The frequency of EBI output clock (MCLK) is controlled by MCLKDIV as follow:</p> <p>000 = HCLK/1.                      001 = HCLK/2.                      010 = HCLK/4.                      011 = HCLK/8.                      100 = HCLK/16.                      101 = HCLK/32.                      110 = HCLK/64.                      111 = HCLK/128.</p>
[7:5]	Reserved	Reserved.
[4]	CACCESS	<p><b>Continuous Data Access Mode</b></p> <p>When continuous access mode enabled, the tASU, tALE and tLHD cycles are bypass for continuous data transfer request.</p> <p>0 = Continuous data access mode Disabled.                      1 = Continuous data access mode Enabled.</p>
[3]	Reserved	Reserved.

[2]	<b>CSPOLINV</b>	<p><b>Chip Select Pin Polar Inverse</b></p> <p>This bit defines the active level of EBI chip select pin (EBI_nCSx), x = 0 or 1.</p> <p>0 = Chip select pin (EBI_nCSx) is active low.</p> <p>1 = Chip select pin (EBI_nCSx) is active high.</p>
[1]	<b>DW16</b>	<p><b>EBI Data Width 16-bit Select</b></p> <p>This bit defines if the EBI data width is 8-bit or 16-bit.</p> <p>0 = EBI data width is 8-bit.</p> <p>1 = EBI data width is 16-bit.</p>
[0]	<b>EN</b>	<p><b>EBI Enable Bit</b></p> <p>This bit is the functional enable bit for EBI.</p> <p>0 = EBI function Disabled.</p> <p>1 = EBI function Enabled.</p>

**External Bus Interface Timing Control Register (EBI\_TCTLx)**

Register	Offset	R/W	Description	Reset Value
EBI_TCTL0	EBI_BA+0x04	R/W	External Bus Interface Bank0 Timing Control Register	0x0000_0000
EBI_TCTL1	EBI_BA+0x14	R/W	External Bus Interface Bank1 Timing Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				R2R			
23	22	21	20	19	18	17	16
WAHDOFF	RAHDOFF	Reserved					
15	14	13	12	11	10	9	8
W2X				Reversed	TAHD		
7	6	5	4	3	2	1	0
TACC					Reserved		

Bits	Description	
[31:30]	Reserved	Reserved.
[27:24]	R2R	<b>Idle Cycle Between Read-to-read</b> This field defines the number of R2R idle cycle. When read action is finish and next action is going to read, R2R idle cycle is inserted and EBI_nCSx return to idle state, x = 0 or 1. $R2R \text{ idle cycle} = (R2R * EBI\_MCLK)$ .
[23]	WAHDOFF	<b>Access Hold Time Disable Control When Write</b> 0 = Data Access Hold Time (tAHD) during EBI writing Enabled. 1 = Data Access Hold Time (tAHD) during EBI writing Disabled.
[22]	RAHDOFF	<b>Access Hold Time Disable Control When Read</b> 0 = Data Access Hold Time (tAHD) during EBI reading Enabled. 1 = Data Access Hold Time (tAHD) during EBI reading Disabled.
[21:16]	Reserved	Reserved.
[15:12]	W2X	<b>Idle Cycle After Write</b> This field defines the number of W2X idle cycle. When write action is finish, W2X idle cycle is inserted and EBI_nCSx return to idle state, x = 0 or 1. $W2X \text{ idle cycle} = (W2X * EBI\_MCLK)$ .
[11]	Reserved	Reserved.
[10:8]	TAHD	<b>EBI Data Access Hold Time</b> TAHD define data access hold time (tAHD). $tAHD = (TAHD + 1) * EBI\_MCLK$ .
[7:3]	TACC	<b>EBI Data Access Time</b> TACC define data access time (tACC).

		$tACC = (TACC + 1) * EBI\_MCLK.$
[2:0]	<b>Reserved</b>	Reserved.

## 6.9 General Purpose I/O (GPIO)

### 6.9.1 Overview

The NUC126 series has up to 86 General Purpose I/O pins to be shared with other function pins depending on the chip configuration. These 86 pins are arranged in 6 ports named as PA, PB, PC, PD, PE and PF. PA, PB, PC, PD has 16 pins on port. PE has 14 pins on port. PF has 8 pins on port. Each of the 86 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 CIOIN (CONFIG0[10]). Each I/O pin has a very weakly individual pull-up resistor which is about 110 k $\Omega$  ~ 300 k $\Omega$  for  $V_{DD}$  is from 5.0 V to 2.5 V.

### 6.9.2 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 Slew Rate I/O mode
- Configurable default I/O mode of all pins after reset by CIOINI (CONFIG0[10]) setting
  - CIOIN = 0, all GPIO pins in input tri-state mode after chip reset
  - CIOIN = 1, all GPIO pins in Quasi-bidirectional mode after chip reset
- I/O pin internal pull-up resistor enabled only in Quasi-bidirectional I/O mode
- Enabling the pin interrupt function will also enable the wake-up function

6.9.3 Block Diagram

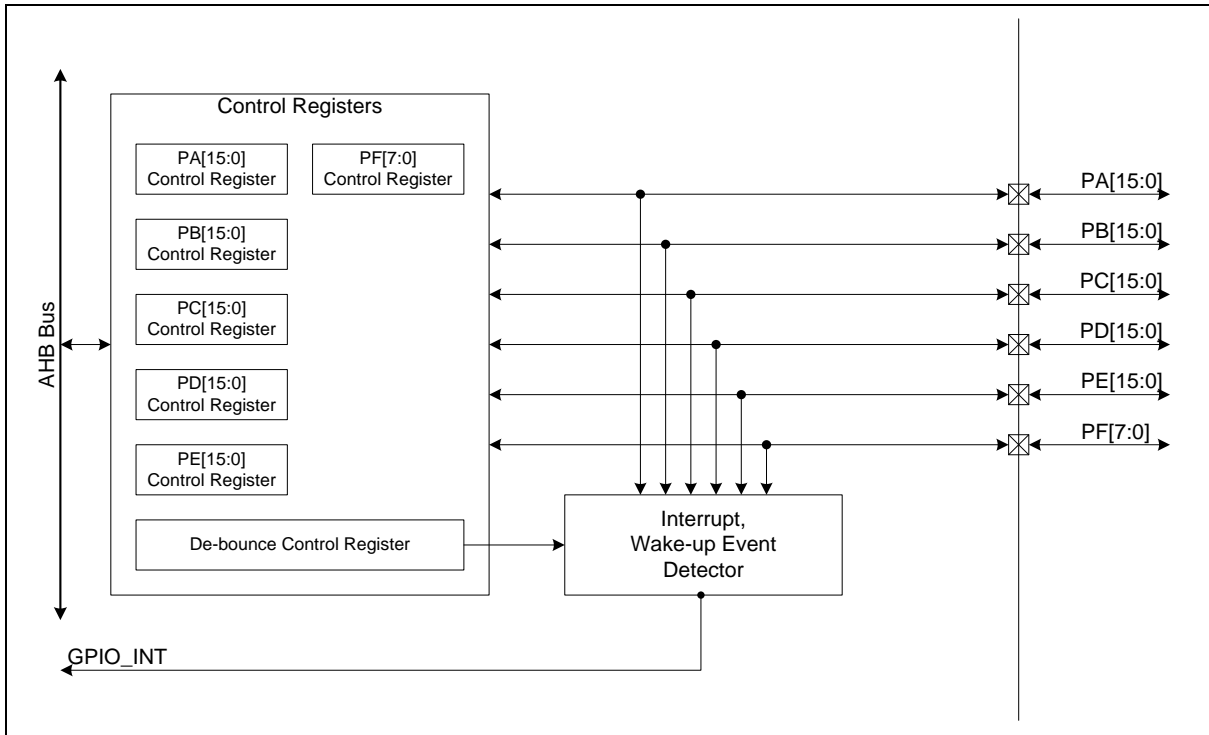


Figure 6.9-1 GPIO Controller Block Diagram

**Note:** The PE.14/PE.15 pin is ignored.

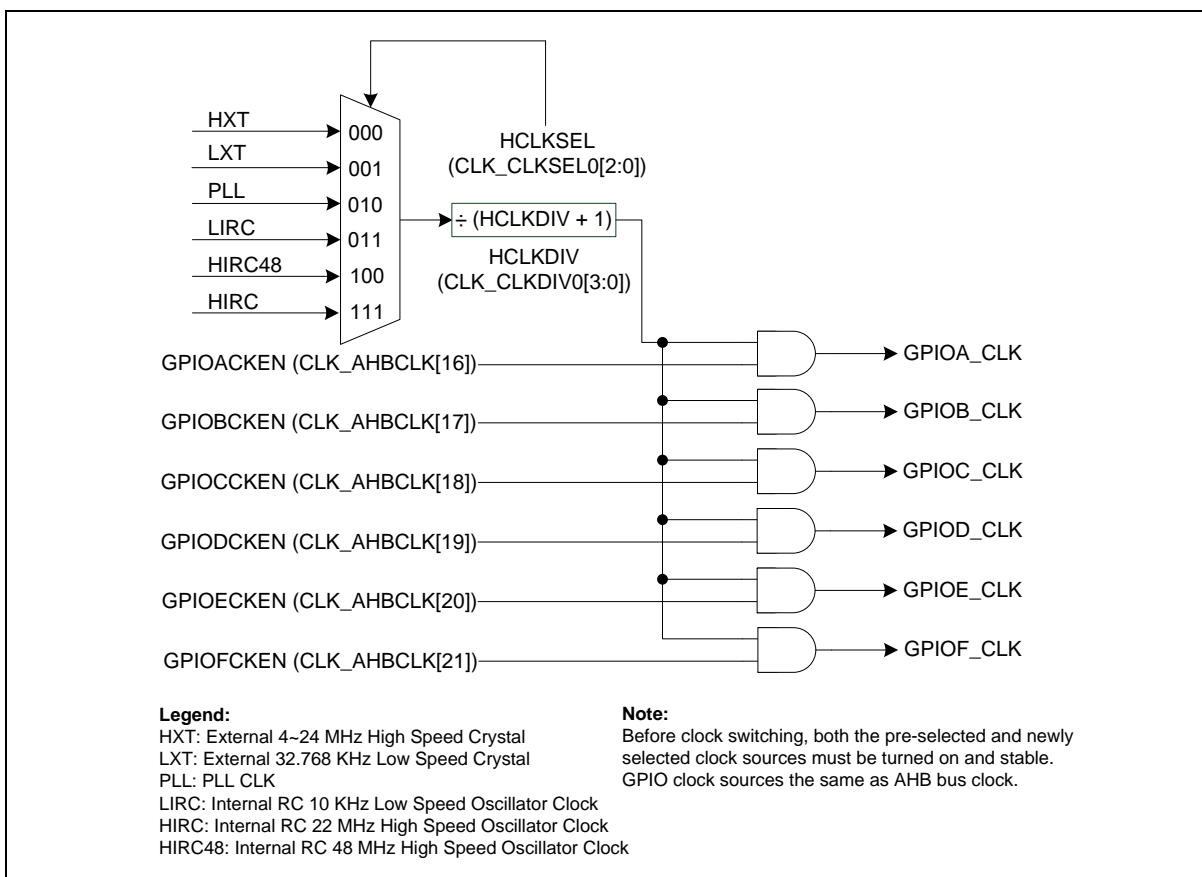


Figure 6.9-2 GPIO Clock Control Diagram

### 6.9.4 Basic Configuration

- Clock Source Configuration
  - Enable GPIO peripheral clock in CLK\_AHBCLK[21:16].
- Reset Configuration
  - Reset GPIO controller in GPIORST (SYS\_IPRST1[1]).
- Pin Configuration
  - The GPIO pin functions are configured in SYS\_PA\_MFPL, SYS\_PA\_MFPH, SYS\_PB\_MFPL, SYS\_PB\_MFPH, SYS\_PC\_MFPL, SYS\_PC\_MFPH, SYS\_PD\_MFPL, SYS\_PD\_MFPH, SYS\_PE\_MFPL, SYS\_PE\_MFPH and SYS\_PF\_MFPL registers.

### 6.9.5 Functional Description

#### 6.9.5.1 Input Mode

Set MODE<sub>n</sub> (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.

#### 6.9.5.2 Push-pull Output Mode

Set MODE<sub>n</sub> (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.

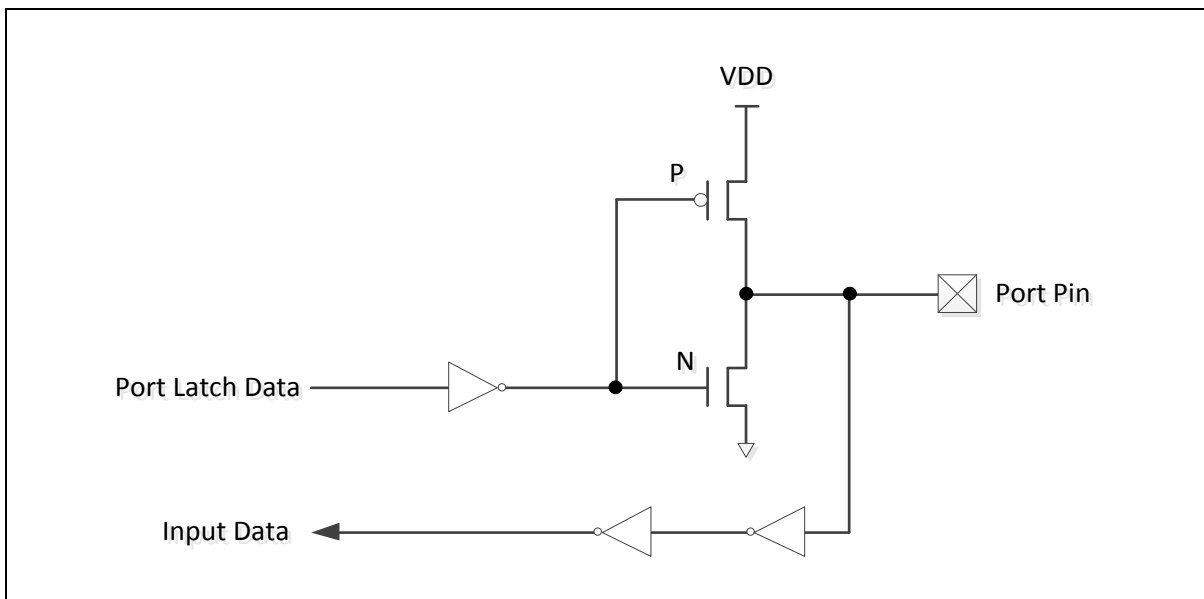


Figure 6.9-2 Push-Pull Output

### 6.9.5.3 Open-drain Mode

Set MODEn (Px\_MODE[2n+1:2n]) to 10 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 resistor is needed for driving high state. If the bit value in the corresponding DOUT (Px\_DOUT[n]) bit is 0, the pin drive 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.

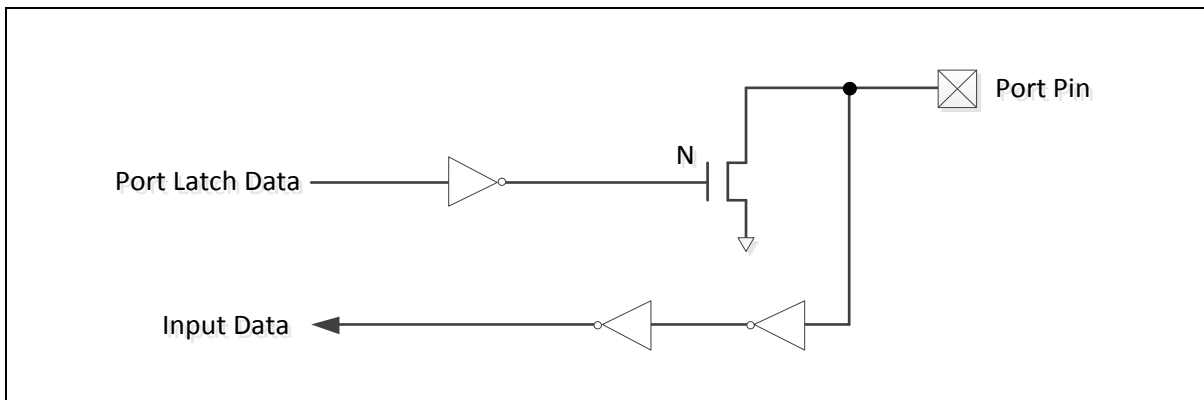


Figure 6.9-3 Open-Drain Output

### 6.9.5.4 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 drive 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 is only about 200 uA to 30



uA for  $V_{DD}$  is form 5.0 V to 2.5 V.

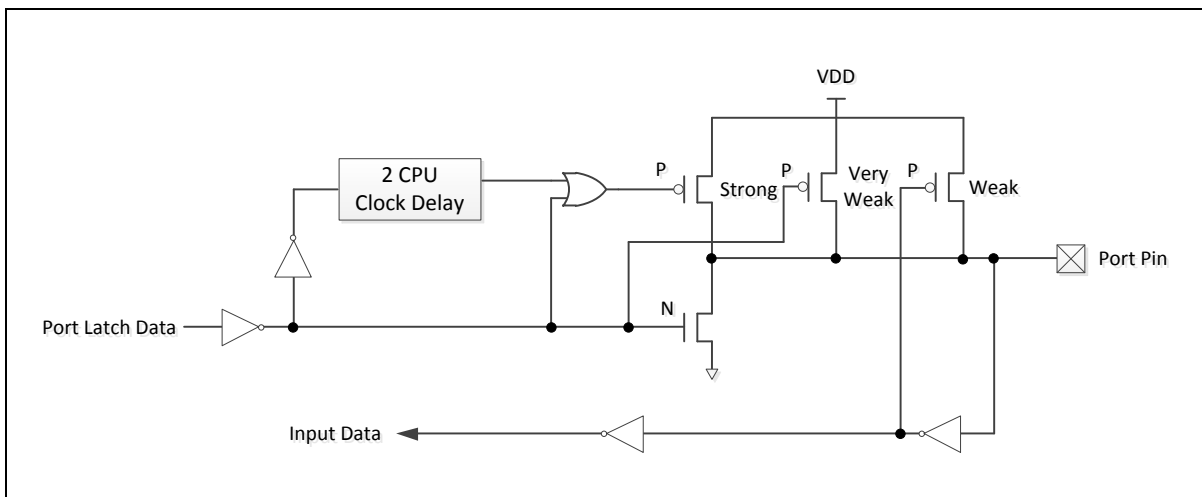


Figure 6.9-4 Quasi-Bidirectional I/O Mode

#### 6.9.5.5 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 condition can be selected: low level trigger, high level trigger, falling edge trigger, rising edge trigger and both rising and falling edge trigger. For edge trigger condition, user can enable input signal de-bounce function to prevent unexpected interrupt happened which caused by noise. The de-bounce clock source and sampling cycle period can be set through DBCLKSRC ( $GPIO\_DBCTL[4]$ ) and DBCLKSEL ( $GPIO\_DBCTL[3:0]$ ) register.

The GPIO can also be the chip wake-up source when chip enters Idle/Power-down mode. The setting of wake-up trigger condition is the same as GPIO interrupt trigger.

#### 6.9.5.6 GPIO Digital Input Path Disable Control

User can disable GPIO digital input path by setting DINOFF ( $Px\_DINOFF[n]$ ). 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 force disabled by hardware and DINOFF control is useless when I/O function configure as ADC/ACMP/ext. XTL..

### 6.9.6 Register Map

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

Register	Offset	R/W	Description	Reset Value
<b>GPIO Base Address:</b>				
<b>GPIO_BA = 0x5000_4000</b>				
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_FFFF
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	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	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	0x0000_0000
PA_SLEWCTL	GPIO_BA+0x028	R/W	PA High Slew Rate Control	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_FFFF
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	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	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	0x0000_0000
PB_SLEWCTL	GPIO_BA+0x068	R/W	PB High Slew Rate Control	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_FFFF
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	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	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	0x0000_0000
PC_SLEWCTL	GPIO_BA+0x0A8	R/W	PC High Slew Rate Control	0x0000_0000
PD_MODE	GPIO_BA+0x0C0	R/W	PD I/O Mode Control	0xFFFF_XXXX
PD_DINOFF	GPIO_BA+0x0C4	R/W	PD Digital Input Path Disable Control	0x0000_0000
PD_DOUT	GPIO_BA+0x0C8	R/W	PD Data Output Value	0x0000_FFFF
PD_DATMSK	GPIO_BA+0x0CC	R/W	PD Data Output Write Mask	0x0000_0000
PD_PIN	GPIO_BA+0x0D0	R	PD Pin Value	0x0000_XXXX
PD_DBEN	GPIO_BA+0x0D4	R/W	PD De-Bounce Enable Control	0x0000_0000
PD_INTTYPE	GPIO_BA+0x0D8	R/W	PD Interrupt Trigger Type Control	0x0000_0000
PD_INTEN	GPIO_BA+0x0DC	R/W	PD Interrupt Enable Control	0x0000_0000
PD_INTSRC	GPIO_BA+0x0E0	R/W	PD Interrupt Source Flag	0x0000_XXXX
PD_SMTEN	GPIO_BA+0x0E4	R/W	PD Input Schmitt Trigger Enable	0x0000_0000
PD_SLEWCTL	GPIO_BA+0x0E8	R/W	PD High Slew Rate Control	0x0000_0000
PE_MODE	GPIO_BA+0x100	R/W	PE I/O Mode Control	0xFFFF_XXXX
PE_DINOFF	GPIO_BA+0x104	R/W	PE Digital Input Path Disable Control	0x0000_0000
PE_DOUT	GPIO_BA+0x108	R/W	PE Data Output Value	0x0000_3FFF
PE_DATMSK	GPIO_BA+0x10C	R/W	PE Data Output Write Mask	0x0000_0000
PE_PIN	GPIO_BA+0x110	R	PE Pin Value	0x0000_XXXX
PE_DBEN	GPIO_BA+0x114	R/W	PE De-Bounce Enable Control	0x0000_0000
PE_INTTYPE	GPIO_BA+0x118	R/W	PE Interrupt Trigger Type Control	0x0000_0000
PE_INTEN	GPIO_BA+0x11C	R/W	PE Interrupt Enable Control	0x0000_0000
PE_INTSRC	GPIO_BA+0x120	R/W	PE Interrupt Source Flag	0x0000_XXXX
PE_SMTEN	GPIO_BA+0x124	R/W	PE Input Schmitt Trigger Enable	0x0000_0000
PE_SLEWCTL	GPIO_BA+0x128	R/W	PE High Slew Rate Control	0x0000_0000
PE_DRVCTL	GPIO_BA+0x12C	R/W	PE High Drive Strength Control	0x0000_0000
PF_MODE	GPIO_BA+0x140	R/W	PF I/O Mode Control	0x0000_XXXX
PF_DINOFF	GPIO_BA+0x144	R/W	PF Digital Input Path Disable Control	0x0000_0000

<b>PF_DOUT</b>	GPIO_BA+0x148	R/W	PF Data Output Value	0x0000_00FF
<b>PF_DATMSK</b>	GPIO_BA+0x14C	R/W	PF Data Output Write Mask	0x0000_0000
<b>PF_PIN</b>	GPIO_BA+0x150	R	PF Pin Value	0x0000_00XX
<b>PF_DBEN</b>	GPIO_BA+0x154	R/W	PF De-Bounce Enable Control	0x0000_0000
<b>PF_INTTYPE</b>	GPIO_BA+0x158	R/W	PF Interrupt Trigger Type Control	0x0000_0000
<b>PF_INTEN</b>	GPIO_BA+0x15C	R/W	PF Interrupt Enable Control	0x0000_0000
<b>PF_INTSRC</b>	GPIO_BA+0x160	R/W	PF Interrupt Source Flag	0x0000_00XX
<b>PF_SMTEN</b>	GPIO_BA+0x164	R/W	PF Input Schmitt Trigger Enable	0x0000_0000
<b>PF_SLEWCTL</b>	GPIO_BA+0x168	R/W	PF High Slew Rate Control	0x0000_0000
<b>GPIO_DBCTL</b>	GPIO_BA+0x180	R/W	Interrupt De-bounce Control	0x0000_0020
<b>PAn_PDIO</b> n=0,1..15	GPIO_BA+0x200+(0x04 * n)	R/W	GPIO PA.n Pin Data Input/Output	0x0000_000X
<b>PBn_PDIO</b> n=0,1..15	GPIO_BA+0x240+(0x04 * n)	R/W	GPIO PB.n Pin Data Input/Output	0x0000_000X
<b>PCn_PDIO</b> n=0,1..15	GPIO_BA+0x280+(0x04 * n)	R/W	GPIO PC.n Pin Data Input/Output	0x0000_000X
<b>PDn_PDIO</b> n=0,1..15	GPIO_BA+0x2C0+(0x04 * n)	R/W	GPIO PD.n Pin Data Input/Output	0x0000_000X
<b>PEn_PDIO</b> n=0,1..13	GPIO_BA+0x300+(0x04 * n)	R/W	GPIO PE.n Pin Data Input/Output	0x0000_000X
<b>PFn_PDIO</b> n=0,1..7	GPIO_BA+0x340+(0x04 * n)	R/W	GPIO PF.n Pin Data Input/Output	0x0000_000X

### 6.9.7 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	0xXXXX_XXXX
PB_MODE	GPIO_BA+0x040	R/W	PB I/O Mode Control	0xXXXX_XXXX
PC_MODE	GPIO_BA+0x080	R/W	PC I/O Mode Control	0xXXXX_XXXX
PD_MODE	GPIO_BA+0x0C0	R/W	PD I/O Mode Control	0xXXXX_XXXX
PE_MODE	GPIO_BA+0x100	R/W	PE I/O Mode Control	0xXXXX_XXXX
PF_MODE	GPIO_BA+0x140	R/W	PF I/O Mode Control	0x0000_XXXX

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

Bits	Description
[2n+1:2n] n=0,1..15	<p><b>MODE[n]</b></p> <p><b>Port A-f I/O Pin[n] Mode Control</b> 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.</p> <p><b>Note1:</b> The initial value of this field is defined by CIOINI (CONFIG0 [10]). If CIOINI is set to 1, the default value is 0xFFFF_FFFF and all pins will be quasi-bidirectional mode after chip powered on. If CIOINI is set to 0, the default value is 0x0000_0000 and all pins will be input mode after chip powered on.</p> <p><b>Note2:</b> Max. n=15 for port A/B/C/D/E. Max. n=7 for port F.</p> <p><b>Note3:</b> The PE.14/PE.15 pin is ignored.</p>

**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
PD_DINOFF	GPIO_BA+0x0C4	R/W	PD Digital Input Path Disable Control	0x0000_0000
PE_DINOFF	GPIO_BA+0x104	R/W	PE 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[n]							
23	22	21	20	19	18	17	16
DINOFF[n]							
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	<p><b>Port A-f Pin[n] Digital Input Path Disable Control</b></p> <p>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.</p> <p>0 = Px.n digital input path Enabled. 1 = Px.n digital input path Disabled (digital input tied to low).</p> <p><b>Note1:</b> Max. n=15 for port A/B/C/D/E. Max. n=7 for port F.</p> <p><b>Note2:</b> The PE.14/PE.15 pin is ignored.</p>
[15:0]	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_FFFF
PB_DOUT	GPIO_BA+0x048	R/W	PB Data Output Value	0x0000_FFFF
PC_DOUT	GPIO_BA+0x088	R/W	PC Data Output Value	0x0000_FFFF
PD_DOUT	GPIO_BA+0x0C8	R/W	PD Data Output Value	0x0000_FFFF
PE_DOUT	GPIO_BA+0x108	R/W	PE Data Output Value	0x0000_3FFF
PF_DOUT	GPIO_BA+0x148	R/W	PF Data Output Value	0x0000_00FF

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[n]							
7	6	5	4	3	2	1	0
DOUT[n]							

Bits	Description	
[31:16]	Reserved	Reserved.
[n] n=0,1..15	DOUT[n]	<p><b>Port A-f Pin[n] Output Value</b></p> <p>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.</p> <p>0 = Px.n will drive Low if the Px.n pin is configured as Push-pull output, Open-drain output or Quasi-bidirectional mode.</p> <p>1 = Px.n will drive High if the Px.n pin is configured as Push-pull output or Quasi-bidirectional mode.</p> <p><b>Note1:</b> Max. n=15 for port A/B/C/D/E. Max. n=7 for port F.</p> <p><b>Note2:</b> The PE.14/PE.15 pin is ignored.</p>

**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
PD_DATMSK	GPIO_BA+0x0CC	R/W	PD Data Output Write Mask	0x0000_0000
PE_DATMSK	GPIO_BA+0x10C	R/W	PE 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[n]							
7	6	5	4	3	2	1	0
DATMSK[n]							

Bits	Description
[31:16]	<b>Reserved</b> Reserved.
[n] n=0,1..15	<p><b>Port A-f Pin[n] Data Output Write Mask</b></p> <p>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 ignored.</p> <p>0 = Corresponding DOUT (Px_DOUT[n]) bit can be updated. 1 = Corresponding DOUT (Px_DOUT[n]) bit protected.</p> <p><b>Note1:</b> This function only protects the corresponding DOUT (Px_DOUT[n]) bit, and will not protect the corresponding PDIO (Pxn_PDIO[0]) bit.</p> <p><b>Note2:</b> Max. n=15 for port A/B/C/D/E. Max. n=7 for port F.</p> <p><b>Note3:</b> The PE.14/PE.15 pin is ignored.</p>



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
PD_PIN	GPIO_BA+0x0D0	R	PD Pin Value	0x0000_XXXX
PE_PIN	GPIO_BA+0x110	R	PE Pin Value	0x0000_XXXX
PF_PIN	GPIO_BA+0x150	R	PF Pin Value	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
PIN[n]							
7	6	5	4	3	2	1	0
PIN[n]							

Bits	Description
[31:16]	<b>Reserved</b> Reserved.
[n] n=0,1..15	<b>PIN[n]</b> <b>Port A-f Pin[n] Pin Value</b> Each bit of the register reflects the actual status of the respective Px.n pin. If the bit is 1, it indicates the corresponding pin status is high; else the pin status is low. <b>Note1:</b> Max. n=15 for port A/B/C/D/E. Max. n=7 for port F. <b>Note2:</b> The PE.14/PE.15 pin is ignored.

Port A-F De-Bounce Enable Control (Px\_DBEN)

Register	Offset	R/W	Description	Reset Value
PA_DBEN	GPIO_BA+0x014	R/W	PA De-Bounce Enable Control	0x0000_0000
PB_DBEN	GPIO_BA+0x054	R/W	PB De-Bounce Enable Control	0x0000_0000
PC_DBEN	GPIO_BA+0x094	R/W	PC De-Bounce Enable Control	0x0000_0000
PD_DBEN	GPIO_BA+0x0D4	R/W	PD De-Bounce Enable Control	0x0000_0000
PE_DBEN	GPIO_BA+0x114	R/W	PE De-Bounce Enable Control	0x0000_0000
PF_DBEN	GPIO_BA+0x154	R/W	PF De-Bounce Enable 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
DBEN[n]							
7	6	5	4	3	2	1	0
DBEN[n]							

Bits	Description	
[31:16]	Reserved	Reserved.
[n] n=0,1..15	DBEN[n]	<p><b>Port A-f Pin[n] Input Signal De-bounce Enable Bit</b></p> <p>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]).</p> <p>0 = Px.n de-bounce function Disabled. 1 = Px.n de-bounce function Enabled.</p> <p>The de-bounce function is valid only for edge triggered interrupt. If the interrupt mode is level triggered, the de-bounce enable bit is ignored.</p> <p><b>Note1:</b> Max. n=15 for port A/B/C/D/E. Max. n=7 for port F.</p> <p><b>Note2:</b> The PE.14/PE.15 pin is ignored.</p>

**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
PD_INTTYPE	GPIO_BA+0x0D8	R/W	PD Interrupt Trigger Type Control	0x0000_0000
PE_INTTYPE	GPIO_BA+0x118	R/W	PE 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[n]							
7	6	5	4	3	2	1	0
TYPE[n]							

Bits	Description	
[31:16]	Reserved	Reserved.
[n] n=0,1..15	TYPE[n]	<p><b>Port A-f Pin[n] Edge or Level Detection Interrupt Trigger Type Control</b></p> <p>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.</p> <p>0 = Edge trigger interrupt. 1 = Level trigger interrupt.</p> <p>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 is ignored and no interrupt will occur.</p> <p>The de-bounce function is valid only for edge triggered interrupt. If the interrupt mode is level triggered, the de-bounce enable bit is ignored.</p> <p><b>Note1:</b> Max. n=15 for port A/B/C/D/E. Max. n=7 for port F.</p> <p><b>Note2:</b> The PE.14/PE.15 pin is ignored.</p>

**Port A-F Interrupt Enable Control (Px\_INTEN)**

Register	Offset	R/W	Description	Reset Value
PA_INTEN	GPIO_BA+0x01C	R/W	PA Interrupt Enable Control	0x0000_0000
PB_INTEN	GPIO_BA+0x05C	R/W	PB Interrupt Enable Control	0x0000_0000
PC_INTEN	GPIO_BA+0x09C	R/W	PC Interrupt Enable Control	0x0000_0000
PD_INTEN	GPIO_BA+0x0DC	R/W	PD Interrupt Enable Control	0x0000_0000
PE_INTEN	GPIO_BA+0x11C	R/W	PE Interrupt Enable Control	0x0000_0000
PF_INTEN	GPIO_BA+0x15C	R/W	PF Interrupt Enable Control	0x0000_0000

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

Bits	Description
[n+16] n=0,1..15	<p><b>Port 0-5 Pin[n] Rising Edge or High Level Interrupt Trigger Type Enable Bit</b></p> <p>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.</p> <p>When setting the RHIEN (Px_INTEN[n+16]) bit to 1:</p> <p>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.</p> <p>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.</p> <p>0 = Px.n level high or low to high interrupt Disabled.</p> <p>1 = Px.n level high or low to high interrupt Enabled.</p> <p><b>Note1:</b> Max. n=15 for port A/B/C/D/E. Max. n=7 for port F.</p> <p><b>Note2:</b> The PE.14/PE.15 pin is ignored.</p>
[n] n=0,1..15	<p><b>Port A-f Pin[n] Falling Edge or Low Level Interrupt Trigger Type Enable Bit</b></p> <p>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.</p> <p>When setting the FLIEN (Px_INTEN[n]) bit to 1:</p> <p>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.</p> <p>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.</p>

	<p>0 = Px.n level low or high to low interrupt Disabled.          1 = Px.n level low or high to low interrupt Enabled.</p> <p><b>Note1:</b>          Max. n=15 for port A/B/C/D/E.          Max. n=7 for port F.</p> <p><b>Note2:</b> The PE.14/PE.15 pin is ignored.</p>
--	---

**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
PD_INTSRC	GPIO_BA+0x0E0	R/W	PD Interrupt Source Flag	0x0000_XXXX
PE_INTSRC	GPIO_BA+0x120	R/W	PE Interrupt Source Flag	0x0000_XXXX
PF_INTSRC	GPIO_BA+0x160	R/W	PF Interrupt Source Flag	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
INTSRC[n]							
7	6	5	4	3	2	1	0
INTSRC[n]							

Bits	Description
[31:16]	<b>Reserved</b> Reserved.
[n] n=0,1..15	<p><b>Port A-f Pin[n] Interrupt Source Flag</b></p> <p>Write Operation: 0 = No action. 1 = Clear the corresponding pending interrupt.</p> <p>Read Operation: 0 = No interrupt at Px.n. 1 = Px.n generates an interrupt.</p> <p><b>Note1:</b> Max. n=15 for port A/B/C/D/E. Max. n=7 for port F.</p> <p><b>Note2:</b> The PE.14/PE.15 pin is ignored.</p>

**Port A-F Input Schmitt Trigger Enable (Px\_SMTEN)**

Register	Offset	R/W	Description	Reset Value
PA_SMTEN	GPIO_BA+0x024	R/W	PA Input Schmitt Trigger Enable	0x0000_0000
PB_SMTEN	GPIO_BA+0x064	R/W	PB Input Schmitt Trigger Enable	0x0000_0000
PC_SMTEN	GPIO_BA+0x0A4	R/W	PC Input Schmitt Trigger Enable	0x0000_0000
PD_SMTEN	GPIO_BA+0x0E4	R/W	PD Input Schmitt Trigger Enable	0x0000_0000
PE_SMTEN	GPIO_BA+0x124	R/W	PE Input Schmitt Trigger Enable	0x0000_0000
PF_SMTEN	GPIO_BA+0x164	R/W	PF Input Schmitt Trigger 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
SMTEN[n]							
7	6	5	4	3	2	1	0
SMTEN[n]							

Bits	Description	
[31:16]	Reserved	Reserved.
[n] n=0,1..15	SMTEN[n]	<p><b>Port A-f Pin[n] Input Schmitt Trigger Enable Bit</b>                      0 = Px.n input schmitt trigger function Disabled.                      1 = Px.n input schmitt trigger function Enabled.</p> <p><b>Note1:</b>                      Max. n=15 for port A/B/C/D/E.                      Max. n=7 for port F.</p> <p><b>Note2:</b> The PE.14/PE.15 pin is ignored.</p>

Port A-F High Slew Rate Control (Px\_SLEWCTL)

Register	Offset	R/W	Description	Reset Value
PA_SLEWCTL	GPIO_BA+0x028	R/W	PA High Slew Rate Control	0x0000_0000
PB_SLEWCTL	GPIO_BA+0x068	R/W	PB High Slew Rate Control	0x0000_0000
PC_SLEWCTL	GPIO_BA+0x0A8	R/W	PC High Slew Rate Control	0x0000_0000
PD_SLEWCTL	GPIO_BA+0x0E8	R/W	PD High Slew Rate Control	0x0000_0000
PE_SLEWCTL	GPIO_BA+0x128	R/W	PE High Slew Rate Control	0x0000_0000
PF_SLEWCTL	GPIO_BA+0x168	R/W	PF High Slew Rate 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
HSREN[n]							
7	6	5	4	3	2	1	0
HSREN[n]							

Bits	Description	
[31:16]	Reserved	Reserved.
[n] n=0,1..15	HSREN[n]	<p><b>Port A-f Pin[n] High Slew Rate Control</b>                      0 = Px.n output with basic slew rate.                      1 = Px.n output with higher slew rate.</p> <p><b>Note1:</b>                      Max. n=15 for port A/B/C/D/E.                      Max. n=7 for port F.</p> <p><b>Note2:</b> The PE.14/PE.15 pin is ignored..</p>



**Port E High Drive Strength Control (Px\_DRVCTL)**

Register	Offset	R/W	Description	Reset Value
PE_DRVCTL	GPIO_BA+0x12C	R/W	PE High Drive Strength 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
Reserved		HDRVEN[n]					
7	6	5	4	3	2	1	0
Reserved							

Bits	Description	
[31:14]	Reserved	Reserved.
[n] n=8,9..13	HDRVEN[n]	<p><b>Port E Pin[n] Driving Strength Control</b></p> <p>0 = Px.n output with basic driving strength.</p> <p>1 = Px.n output with high driving strength.</p> <p><b>Note:</b></p> <p>n=8,9..13 for port E.</p>
[7:0]	Reserved	Reserved.

**Interrupt De-bounce Control (GPIO\_DBCTL)**

Register	Offset	R/W	Description	Reset Value
GPIO_DBCTL	GPIO_BA+0x180	R/W	Interrupt De-bounce Control	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		ICLKON	DBCLKSRC	DBCLKSEL			

Bits	Description	
[31:6]	Reserved	Reserved.
[5]	ICLKON	<p><b>Interrupt Clock on Mode</b></p> <p>0 = Edge detection circuit is active only if I/O pin corresponding RHIEN (Px_INTEN[n+16])/FLIEN (Px_INTEN[n]) bit is set to 1.</p> <p>1 = All I/O pins edge detection circuit is always active after reset.</p> <p><b>Note:</b> It is recommended to disable this bit to save system power if no special application concern.</p>
[4]	DBCLKSRC	<p><b>De-bounce Counter Clock Source Selection</b></p> <p>0 = De-bounce counter clock source is the HCLK.</p> <p>1 = De-bounce counter clock source is the internal 10 kHz internal low speed oscillator.</p>
[3:0]	DBCLKSEL	<p><b>De-bounce Sampling Cycle Selection</b></p> <p>0000 = Sample interrupt input once per 1 clocks.</p> <p>0001 = Sample interrupt input once per 2 clocks.</p> <p>0010 = Sample interrupt input once per 4 clocks.</p> <p>0011 = Sample interrupt input once per 8 clocks.</p> <p>0100 = Sample interrupt input once per 16 clocks.</p> <p>0101 = Sample interrupt input once per 32 clocks.</p> <p>0110 = Sample interrupt input once per 64 clocks.</p> <p>0111 = Sample interrupt input once per 128 clocks.</p> <p>1000 = Sample interrupt input once per 256 clocks.</p> <p>1001 = Sample interrupt input once per 2*256 clocks.</p> <p>1010 = Sample interrupt input once per 4*256 clocks.</p> <p>1011 = Sample interrupt input once per 8*256 clocks.</p> <p>1100 = Sample interrupt input once per 16*256 clocks.</p> <p>1101 = Sample interrupt input once per 32*256 clocks.</p> <p>1110 = Sample interrupt input once per 64*256 clocks.</p> <p>1111 = Sample interrupt input once per 128*256 clocks.</p>

**GPIO Px.n Pin Data Input/Output (Pxn\_PDIO)**

Register	Offset	R/W	Description	Reset Value
PAn_PDIO n=0,1..15	GPIO_BA+0x200+(0x04 * n)	R/W	GPIO PA.n Pin Data Input/Output	0x0000_000X
PBn_PDIO n=0,1..15	GPIO_BA+0x240+(0x04 * n)	R/W	GPIO PB.n Pin Data Input/Output	0x0000_000X
PCn_PDIO n=0,1..15	GPIO_BA+0x280+(0x04 * n)	R/W	GPIO PC.n Pin Data Input/Output	0x0000_000X
PDn_PDIO n=0,1..15	GPIO_BA+0x2C0+(0x04 * n)	R/W	GPIO PD.n Pin Data Input/Output	0x0000_000X
PEn_PDIO n=0,1..13	GPIO_BA+0x300+(0x04 * n)	R/W	GPIO PE.n Pin Data Input/Output	0x0000_000X
PFn_PDIO n=0,1..7	GPIO_BA+0x340+(0x04 * n)	R/W	GPIO PF.n Pin Data Input/Output	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	<p><b>GPIO Px.N Pin Data Input/Output</b>                      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 (PA_DOUT[0]), reading PA0_PDIO will return the value of PIN (PA_PIN[0]).</p> <p><b>Note1:</b> The writing operation will not be affected by register DATMSK (Px_DATMSK[n]).</p> <p><b>Note2:</b>                      Max. n=15 for port A/B/C/D/E.                      Max. n=7 for port F.</p> <p><b>Note3:</b> The PE.14/PE.15 pin is ignored.</p>

## 6.10 Hardware Divider (HDIV)

### 6.10.1 Overview

The hardware divider (HDIV) is useful to the high performance application. The hardware divider is a signed, integer divider with both quotient and remainder outputs.

### 6.10.2 Features

- Signed (two's complement) integer calculation
- 32-bit dividend with 16-bit divisor calculation capacity
- 32-bit quotient and 32-bit remainder outputs (16-bit remainder with sign extends to 32-bit)
- Divided by zero warning flag
- 6 HCLK clocks taken for one cycle calculation
- Write divisor to trigger calculation
- Waiting for calculation ready automatically when reading quotient and remainder

### 6.10.3 Block Diagram

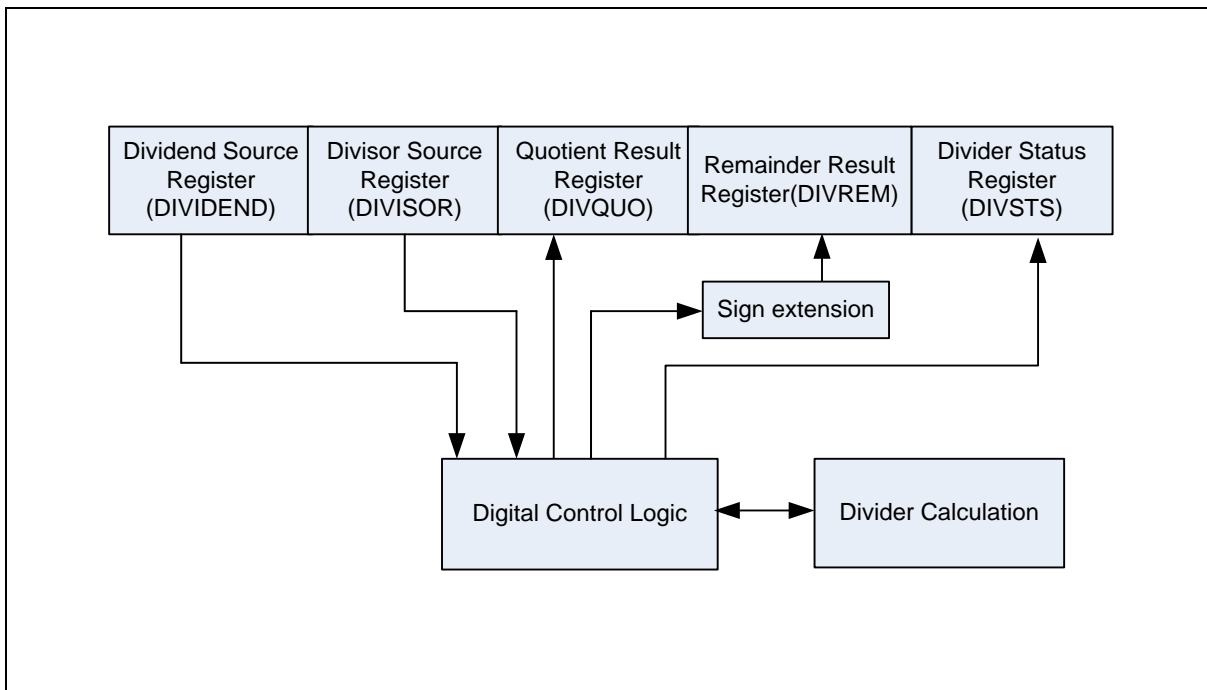


Figure 6.10-1 Hardware Divider Block Diagram

### 6.10.4 Basic Configuration

- Clock Source Configuration
  - Enable HDIV peripheral clock in HDIVCKEN (CLK\_AHBCLK[4]).
- Reset Configuration
  - Reset HDIV in HDIVRST (SYS\_IPRST0[4]).

### 6.10.5 Functional Description

To use hardware divider, it needs to set dividend first. Then set divisor and the hardware divider will trigger calculation automatically after divisor written. The calculation results including the quotient and remainder could be got by reading DIVQUO and DIVREM register. If CPU reads DIVQUO or DIVREM before hardware divider calculation finishing, CPU will be held until hardware divider finishing the calculation. Therefore, CPU can always get valid results after trigger one hardware divider calculation without software delay.

DIV0 flag of DIVSTS will be set if divisor is 0.

The dividend is 32-bit signed integer and divisor is 16-bit signed integer. The quotient is 32-bit signed integer and the remainder is 16-bit signed integer. It is noted that the case of dividing the minimum dividend by -1, the quotient is set to be the minimum negative value since overflow and the remainder is set to 0. This is the only case the quotient is not represented in a positive number when a negative number by a negative number.

The following figure shows the operation flow of hardware divider. To calculation  $X / Y$ , CPU needs to write  $X$  to DIVIDEND register, and then write  $Y$  to DIVISOR. CPU can read DIVQUO and DIVREM registers to get calculation results after DIVISOR been written.

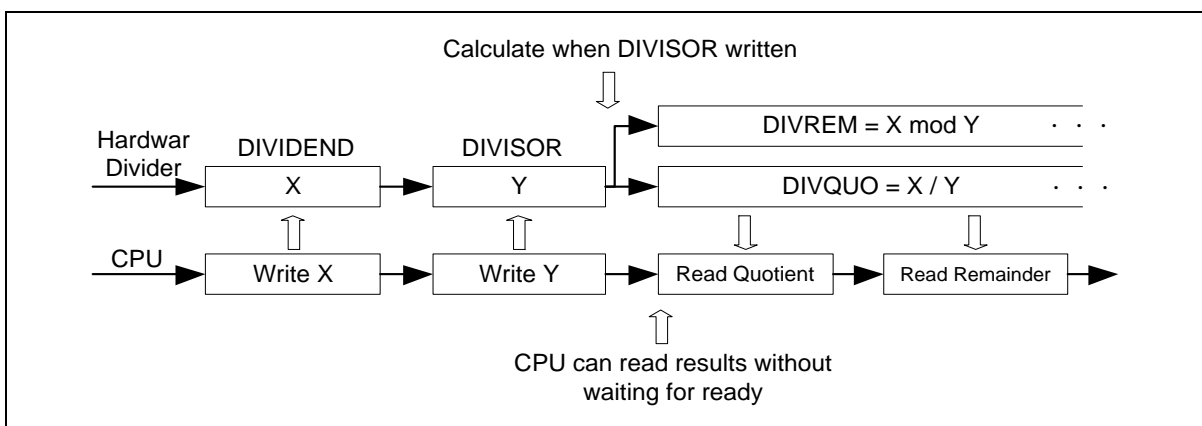


Figure 6.10-2 Hardware Divider Operation Flow

### 6.10.6 Register Map

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

Register	Offset	R/W	Description	Reset Value
<b>HDIV Base Address:</b>				
<b>HDIV_BA = 0x5001_4000</b>				
<b>HDIV_DIVIDEND</b>	HDIV_BA+0x00	R/W	Dividend Source Register	0x0000_0000
<b>HDIV_DIVISOR</b>	HDIV_BA+0x04	R/W	Divisor Source Register	0x0000_FFFF
<b>HDIV_DIVQUO</b>	HDIV_BA+0x08	R/W	Quotient Result Register	0x0000_0000
<b>HDIV_DIVREM</b>	HDIV_BA+0x0C	R/W	Remainder Result Register	0x0000_0000
<b>HDIV_DIVSTS</b>	HDIV_BA+0x10	R	Divider Status Register	0x0000_0001

6.10.7 Register Description

**Dividend Source Register (HDIV\_DIVIDEND)**

Register	Offset	R/W	Description	Reset Value
HDIV_DIVIDEND	HDIV_BA+0x00	R/W	Dividend Source Register	0x0000_0000

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

Bits	Description	
[31:0]	<b>DIVIDEND</b>	<b>Dividend Source</b> This register is given the dividend of divider before calculation starting.

**Divisor Source Register (HDIV\_DIVISOR)**

Register	Offset	R/W	Description	Reset Value
HDIV_DIVISOR	HDIV_BA+0x04	R/W	Divisor Source Register	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
DIVISOR							
7	6	5	4	3	2	1	0
DIVISOR							

Bits	Description	
[31:16]	<b>Reserved</b>	Reserved.
[15:0]	<b>DIVISOR</b>	<b>Divisor Source</b> This register is given the divisor of divider before calculation starts. <b>Note:</b> When this register is written, hardware divider will start calculate.



**Quotient Result Register (HDIV\_DIVQUO)**

Register	Offset	R/W	Description	Reset Value
HDIV_DIVQUO	HDIV_BA+0x08	R/W	Quotient Result Register	0x0000_0000

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

Bits	Description	
[31:0]	QUOTIENT	<p><b>Quotient Result</b> This register holds the quotient result of divider after calculation complete.</p>

**Remainder Result Register (HDIV\_DIVREM)**

Register	Offset	R/W	Description	Reset Value
HDIV_DIVREM	HDIV_BA+0x0C	R/W	Remainder Result Register	0x0000_0000

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

Bits	Description
[31:0]	<p><b>REMAINDER</b></p> <p><b>Remainder Result</b>                      The remainder of hardware divider is 16-bit sign integer (REMAINDER[15:0]), which holds the remainder result of divider after calculation complete. The remainder of hardware divider with sign extension (REMAINDER[31:16]) to 32-bit integer.                      This register holds the remainder result of divider after calculation complete.</p>

**Divider Status Register (HDIV\_DIVSTS)**

Register	Offset	R/W	Description	Reset Value
HDIV_DIVSTS	HDIV_BA+0x10	R	Divider Status 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						DIV0	FINISH

Bits	Description	
[31:2]	Reserved	Reserved.
[1]	DIV0	<p><b>Divisor Zero Warning</b>                      0 = The divisor is not 0.                      1 = The divisor is 0.</p> <p><b>Note:</b> The DIV0 flag is used to indicate divide-by-zero situation and updated whenever DIVISOR is written. This register is read only.</p>
[0]	FINISH	<p><b>Division Finish Flag</b>                      0 = Under Calculation.                      1 = Calculation finished.</p> <p>The flag will become low when the divider is in calculation. The flag will go back to high once the calculation finished.</p>

## 6.11 I<sup>2</sup>C Serial Interface Controller (I<sup>2</sup>C)

### 6.11.1 Overview

I<sup>2</sup>C is a two-wire, bi-directional serial bus that provides a simple and efficient method of data exchange between devices. The I<sup>2</sup>C 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 I<sup>2</sup>C controllers which support Power-down wake-up function.

### 6.11.2 Features

The I<sup>2</sup>C bus uses two wires (SDA and SCL) to transfer information between devices connected to the bus. The main features of the I<sup>2</sup>C bus include:

- Supports up to two I<sup>2</sup>C ports
- Master/Slave mode
- Bidirectional data transfer between masters and slaves
- Multi-master bus (no central master)
- Arbitration between simultaneously transmitting masters without corruption of serial data on the bus
- Serial clock synchronization allow 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 I<sup>2</sup>C interrupt if the I<sup>2</sup>C bus hangs up and timer-out counter overflows
- Programmable clocks allow for versatile rate control
- Supports 7-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 two-level buffer function
- Supports setup/hold time programmable

### 6.11.3 Block Diagram

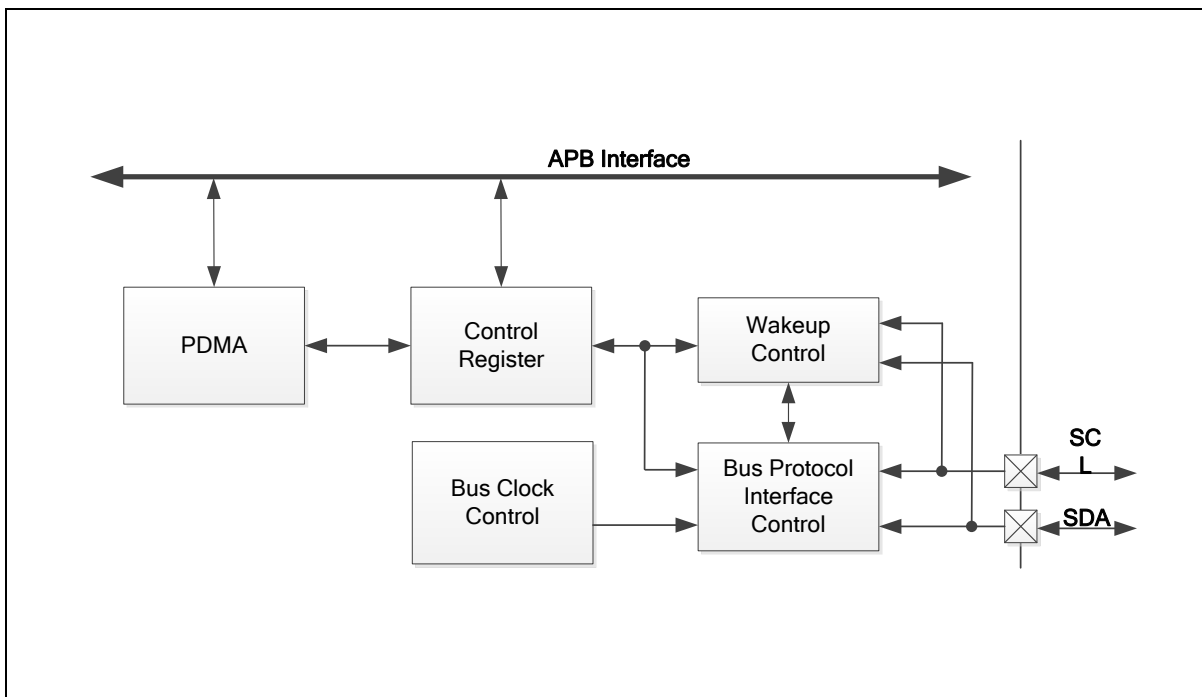


Figure 6.11-1 I<sup>2</sup>C Controller Block Diagram

### 6.11.4 Basic Configuration

#### 6.11.4.1 Basic Configuration of I2C0

- Clock Source Configuration
  - Enable I2C0 peripheral clock in I2C0CKEN (CLK\_APBCLK0[8]).
- Reset Configuration
  - Reset I2C0 controller in I2C0RST (SYS\_IPRST1[8]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
I2C0	I2C0_SCL	PE.4, PE.6	MFP2
		PD.5	MFP3
		PA.3, PE.12	MFP4
	I2C0_SDA	PE.5, PE.7	MFP2
		PD.4	MFP3
		PA.2, PE.13	MFP4

#### 6.11.4.2 Basic Configuration of I2C1

- Clock Source Configuration
  - Enable I2C1 peripheral clock in I2C1CKEN (CLK\_APBCLK0[9]).
- Reset Configuration

- Reset I2C1 controller in I2C1RST (SYS\_IPRST1[9]).

- Pin Configuration

Group	Pin Name	GPIO	MFP
I2C1	I2C1_SCL	PA.8	MFP2
		PC.4, PC.9, PE.4, PF.3	MFP3
		PE.8	MFP4
	I2C1_SDA	PA.9	MFP2
		PC.5, PC.10, PE.0, PE.5, PF.4	MFP3
		PE.9	MFP4

### 6.11.5 Functional Description

On I<sup>2</sup>C 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 6.11-2 for more detailed I<sup>2</sup>C BUS Timing.

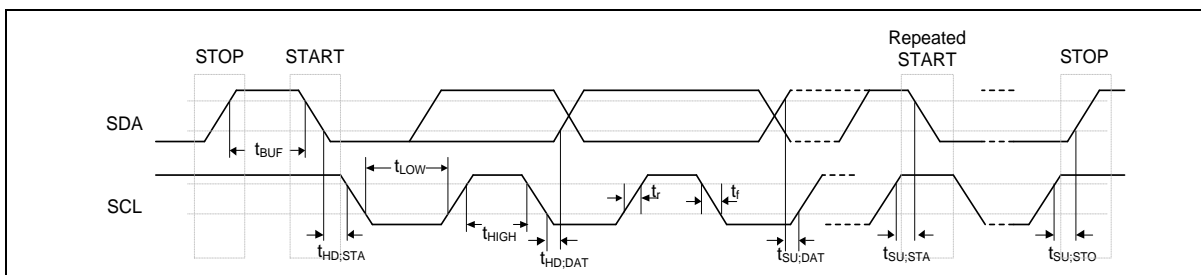


Figure 6.11-2 I<sup>2</sup>C Bus Timing

The device's on-chip I<sup>2</sup>C provides the serial interface that meets the I<sup>2</sup>C bus standard mode specification. The I<sup>2</sup>C port handles byte transfers autonomously. To enable this port, the bit I2CEN in (I2C\_CTL [6]) should be set to '1'. The I<sup>2</sup>C hardware interfaces to the I<sup>2</sup>C bus via two pins: SDA and SCL. When I/O pins are used as I<sup>2</sup>C ports, user must set the pins function to I<sup>2</sup>C in advance.

There is two-level buffer to improve the performance of I<sup>2</sup>C bus. In two-level buffer mode, the next transmitted or the last received data can be active even if the current data is transmitted or the last received isn't read back yet.

There are under run or over run interrupt when the two-level buffer mode is enabled and the interrupt event enable is set. When two-level buffer under run, user will capture the last data be read. When two-level buffer over run, user will capture the last data be written.

**Note:** Pull-up resistor is needed for I<sup>2</sup>C operation as the SDA and SCL are open-drain pins.

#### 6.11.5.1 I<sup>2</sup>C Protocol

Figure 6.11-3 shows the typical I<sup>2</sup>C 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

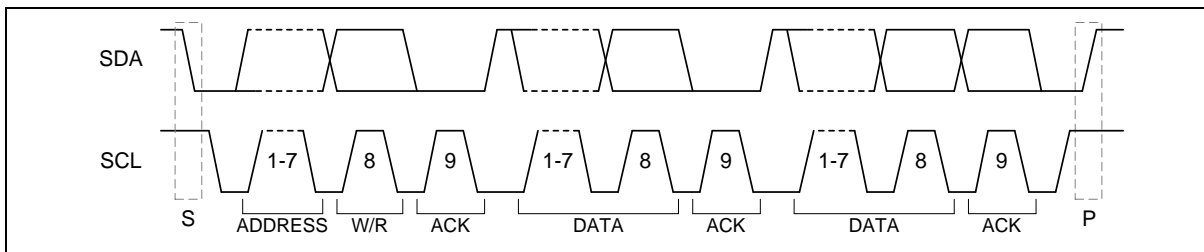


Figure 6.11-3 I<sup>2</sup>C Protocol

**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 6.11-4 shows the waveform of START, Repeat START and STOP.

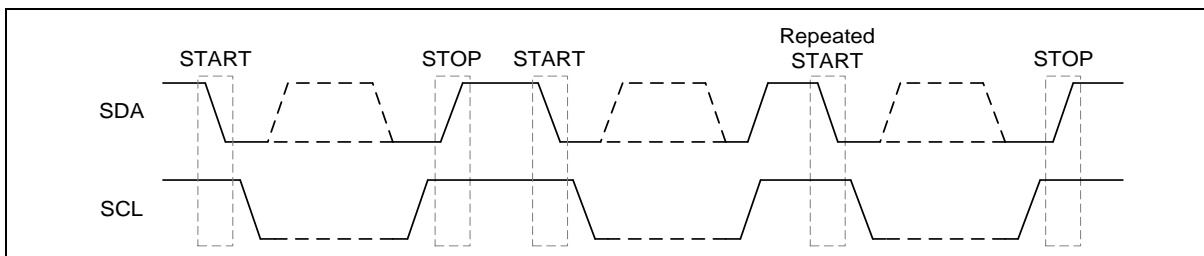


Figure 6.11-4 START and STOP Conditions

**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 byte. 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.

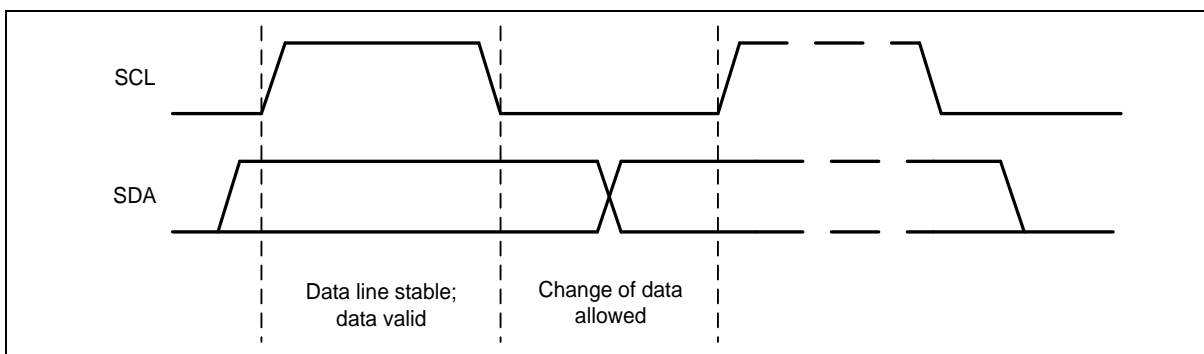


Figure 6.11-5 Bit Transfer on the I<sup>2</sup>C Bus

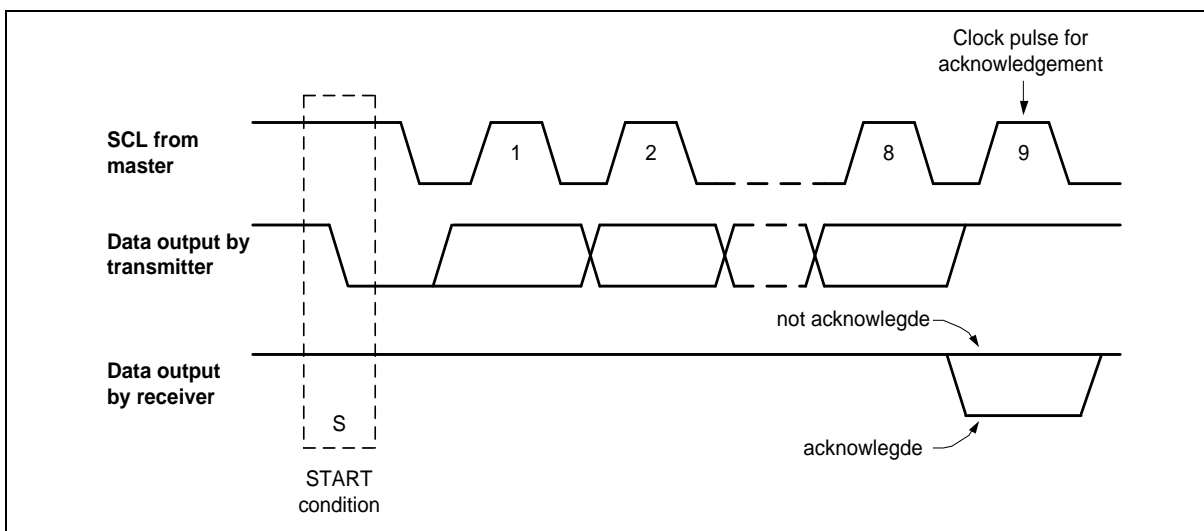


Figure 6.11-6 Acknowledge on the I<sup>2</sup>C Bus

**Data Transfer on I<sup>2</sup>C bus**

Figure 6.11-7 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.



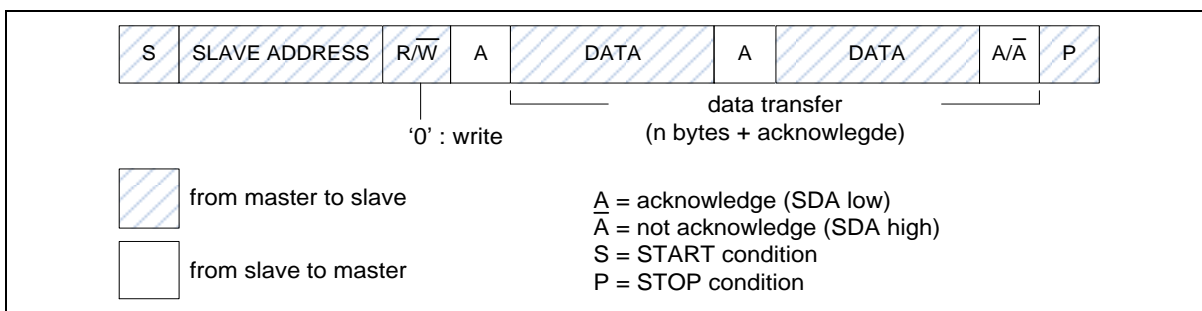


Figure 6.11-7 Master Transmits Data to Slave by 7-bit

Figure 6.11-8 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.

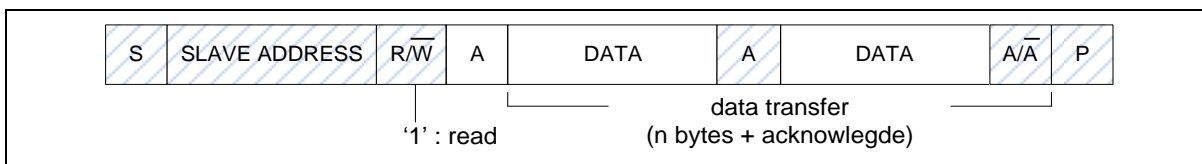


Figure 6.11-8 Master Reads Data from Slave by 7-bit

### 6.11.5.2 Operation Modes

The on-chip I<sup>2</sup>C ports support three operation modes, Master, Slave, and General Call Mode.

In a given application, I<sup>2</sup>C port may operate as a master or as a slave. In Slave mode, the I<sup>2</sup>C 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, I<sup>2</sup>C port switches to Slave mode immediately and can detect its own slave address in the same serial transfer.

To control the I<sup>2</sup>C bus transfer in each mode, user needs to set I2C\_CTL, I2C\_DAT registers according to current status code of I2C\_STATUS register. In other words, for each I<sup>2</sup>C bus action, user needs to check current status by I2C\_STATUS register, and then set I2C\_CTL, I2C\_DAT registers to take bus action. Finally, check the response status by I2C\_STATUS.

The bits, STA, STO and AA in I2C\_CTL register are used to control the next state of the I<sup>2</sup>C hardware after SI (I2C\_CTL [3]) register is cleared. Upon completion of the new action, a new status code will be updated in I2C\_STATUS register and the SI flag of I2C\_CTL register will be set. But the SI flag will not be set when I<sup>2</sup>C STOP. If the I<sup>2</sup>C interrupt control bit INTEN (I2C\_CTL [7]) is set, appropriate action or software branch of the new status code can be performed in the Interrupt service routine.

Figure 6.11-9 shows the current I<sup>2</sup>C 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 I<sup>2</sup>C bus. If a slave on the bus matches the address and response ACK, the I2C\_STATUS will be updated by status code 0x18.

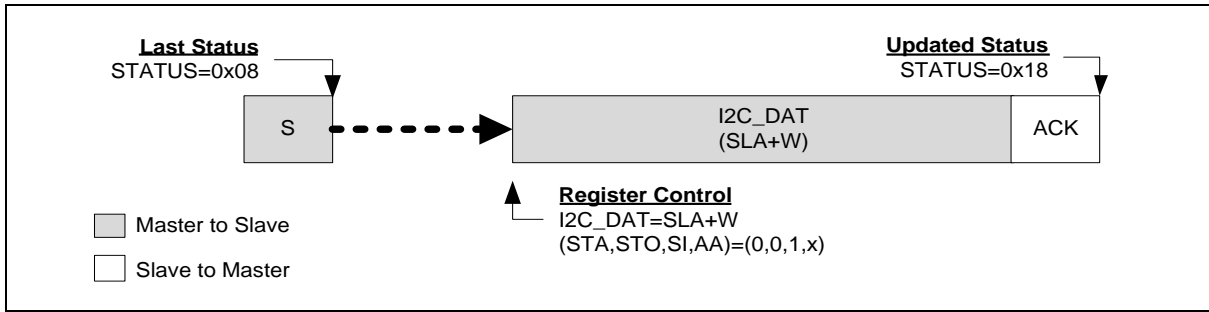


Figure 6.11-9 Control I<sup>2</sup>C Bus according to the current I<sup>2</sup>C Status

**Master Mode**

In Figure 6.11-10, all possible protocols for I<sup>2</sup>C master are shown. User needs to follow proper path of the flow to implement required I<sup>2</sup>C protocol.

In other words, user can send a START signal to bus and I<sup>2</sup>C will be in Master Transmitter (MT) mode (Figure 6.11-10) or Master receiver (MR) mode (Figure 6.11-11) 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 I<sup>2</sup>C protocol.

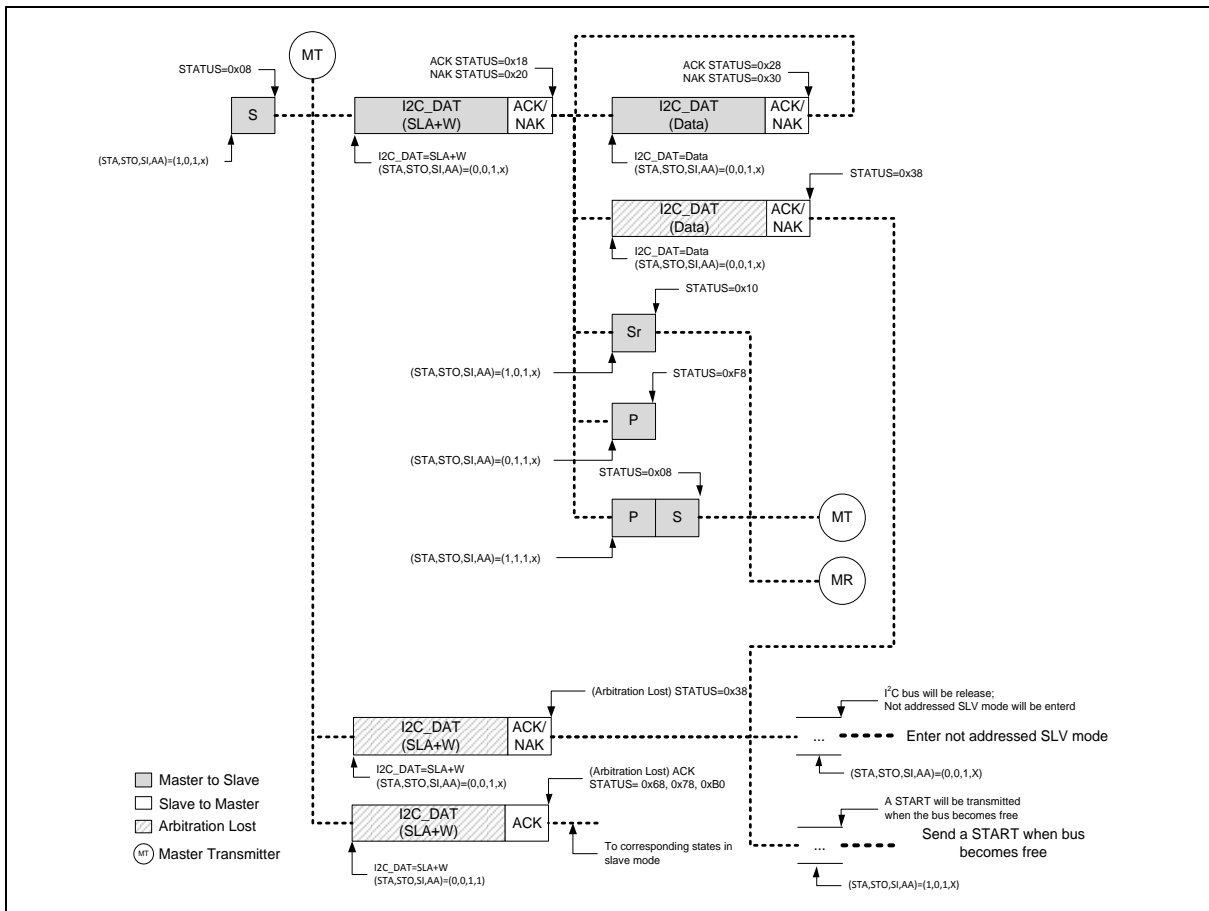


Figure 6.11-10 Master Transmitter Mode Control Flow

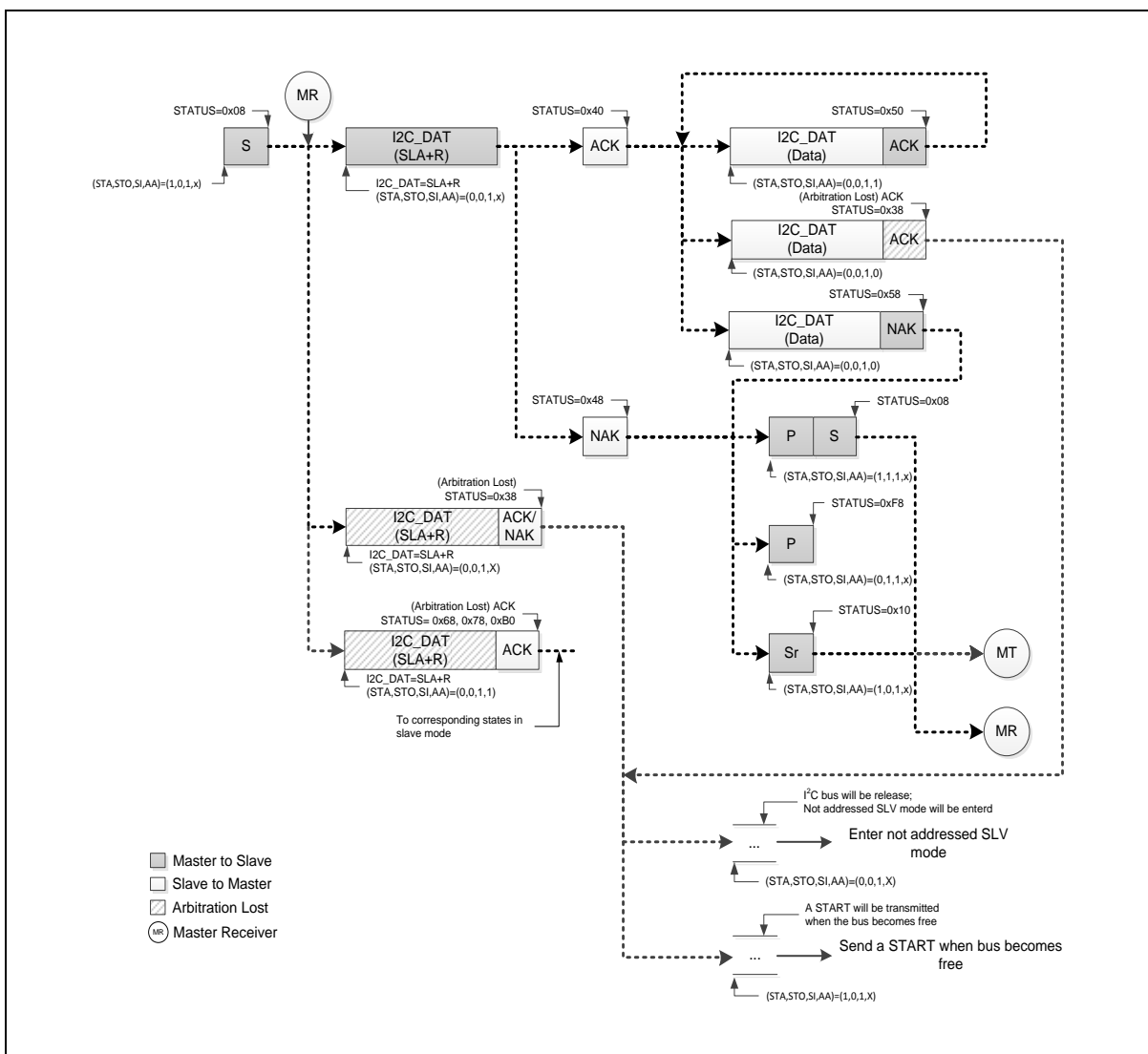


Figure 6.11-11 Master Receiver Mode Control Flow

If the I<sup>2</sup>C 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 I<sup>2</sup>C bus and enter not addressed Slave mode.

**Slave Mode**

When reset default, I<sup>2</sup>C is not addressed and will not recognize the address on I<sup>2</sup>C bus. User can set slave address by I2C\_ADDRn (n=0~3) and set (STA, STO, SI, AA) = (0, 0, 1, 1) to let I<sup>2</sup>C recognize the address sent by master. Figure 6.11-12 shows all the possible flow for I<sup>2</sup>C in Slave mode. Users need to follow a proper flow (as shown in Figure 6.11-12) to implement their own I<sup>2</sup>C protocol.

If bus arbitration is lost in Master mode, I<sup>2</sup>C 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 I<sup>2</sup>C communication, the SCL clock will be released when writing '1' to clear SI flag in Slave mode.

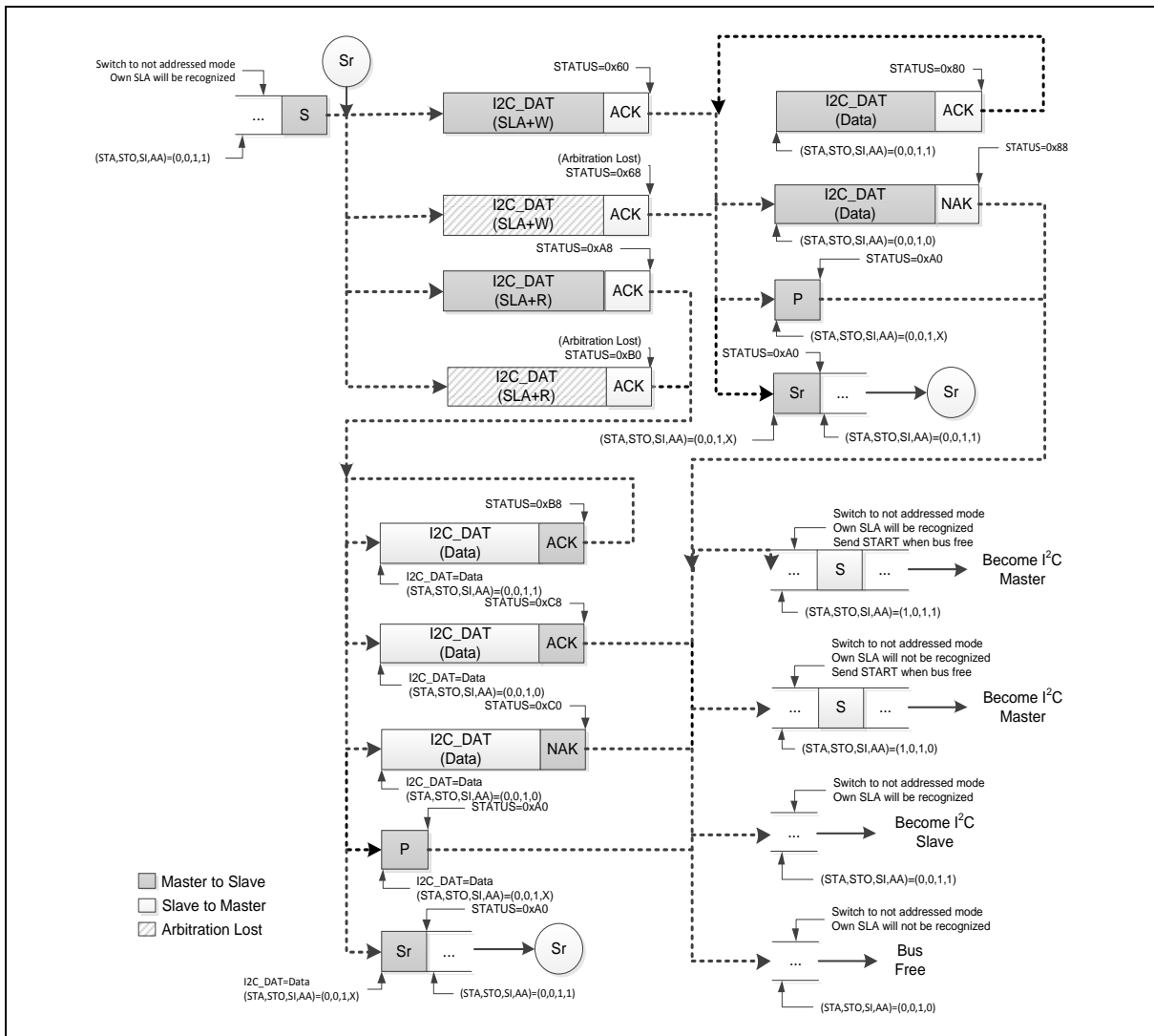


Figure 6.11-12 Slave Mode Control Flow

If I<sup>2</sup>C 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 Figure 6.11-12 when getting 0xA0 status.

If I<sup>2</sup>C 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 Figure 6.11-12 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 I<sup>2</sup>C signal or address from master. At this status, I<sup>2</sup>C should enter idle mode.

**General Call (GC) Mode**

If the GC bit (I2C\_ADDRn [0]) is set, the I<sup>2</sup>C 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 I<sup>2</sup>C in Slave mode, it can receive the general call address by 0x00 after master send general call address to I<sup>2</sup>C bus, then it will follow status of GC mode.

The GC mode can wake up when address matched.

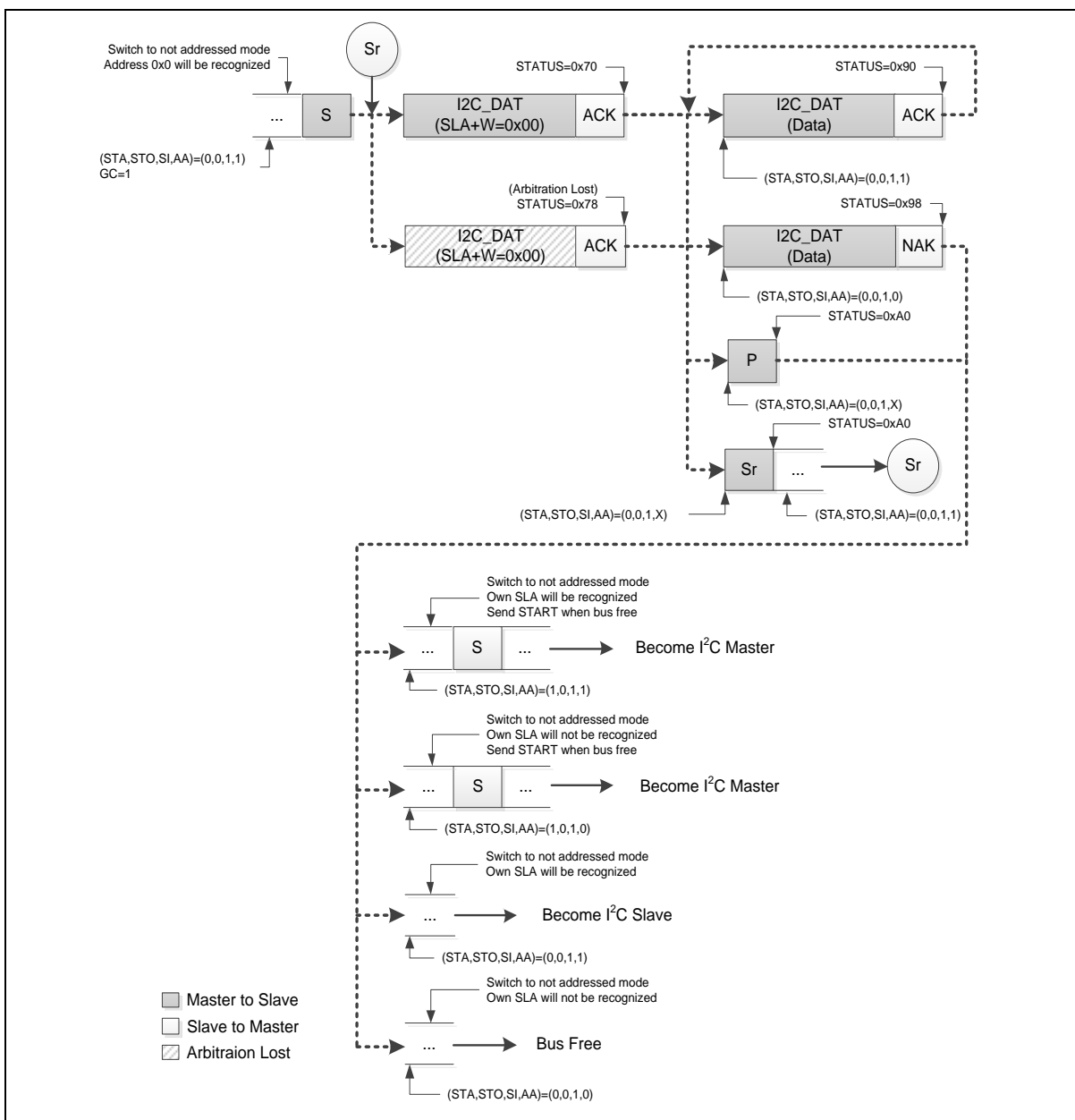


Figure 6.11-13 GC Mode

If I<sup>2</sup>C is still receiving data in GC mode but got a STOP or Repeat START, the status code will be 0xA0. User could follow the action for status code 0x98 in Figure 6.11-13 when getting 0xA0 status.

**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 I<sup>2</sup>C signal or address from master. At this time, I<sup>2</sup>C controller should enter idle mode.

**Multi-Master**

In some applications, there are two or more masters on the same I<sup>2</sup>C bus to access slaves, and the masters may transmit data simultaneously. The I<sup>2</sup>C 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.

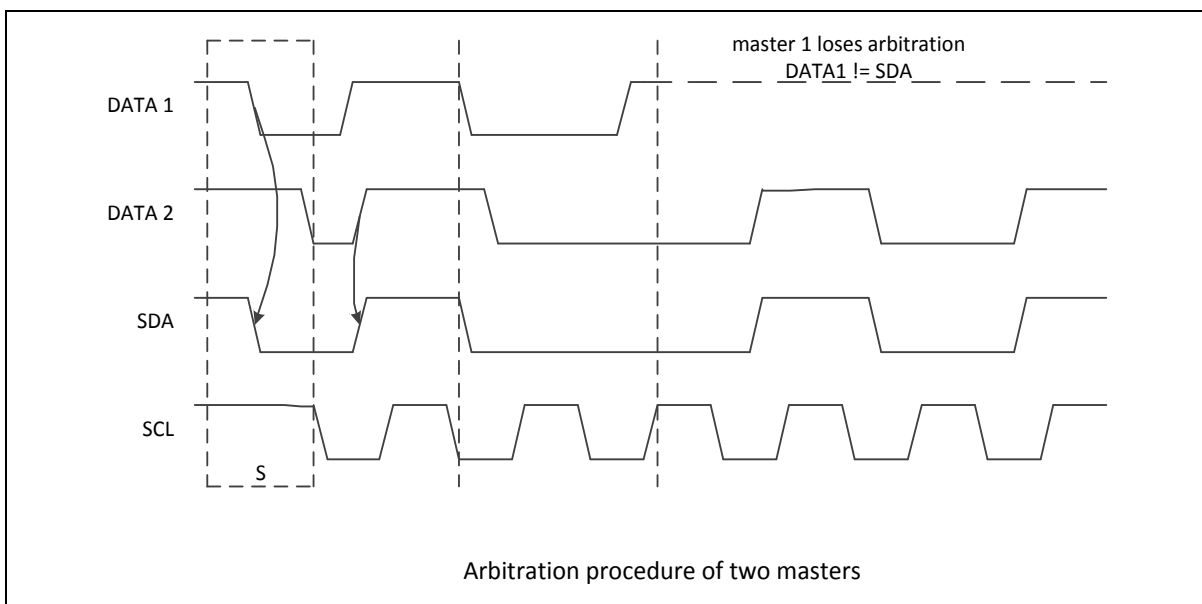


Figure 6.11-14 Arbitration Lost

- When I2C\_STATUS = 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\_STATUS = 0x00, a “Bus Error” is received. To recover I<sup>2</sup>C 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

### 6.11.5.3 PDMA Transfer Function

I<sup>2</sup>C controller supports PDMA transfer function.

When TXPDMAEN (I2C\_CTL1 [0]) is set to 1, the controller will issue request to PDMA controller to start the PDMA transmission process automatically. When RXPDMAEN (I2C\_CTL1 [1]) is set to 1, the I<sup>2</sup>C controller will start the receive PDMA process. I<sup>2</sup>C controller will issue the request to PDMA controller automatically when there is data written into the received BUFFER or the status of RXEMPTY (I2C\_STATUS1 [5]) is set to 0.

When I<sup>2</sup>C enter 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 I<sup>2</sup>C controller in master TX mode. If PDMASTR is cleared to 0, I<sup>2</sup>C will send STOP automatically after PDMA transfer done and buffer empty. If PDMASTR is set to 1, SI will be set to 1 and I<sup>2</sup>C bus will be stretched by hardware after PDMA transfer done and buffer

empty.

6.11.5.4 Two-level Buffer Mode on I2C bus

Set to enable the two-level buffer for I<sup>2</sup>C transmitted or received buffer. It is used to improve the performance of the I<sup>2</sup>C bus. If this TWOBUFEN bit is set = 1, the control bit of STA for repeat start or STO bit should be set as normal operation.

For example: if there are 4 data shall be transmitted and then stop it. The STO bit shall be set after the 4<sup>th</sup> data's SI event being clear. In this time, the 4<sup>th</sup> data can be transmitted and the I<sup>2</sup>C stop after the 4<sup>th</sup> data transmission done. The STOP signal will occur after the 4<sup>th</sup> data's SI event being clear.

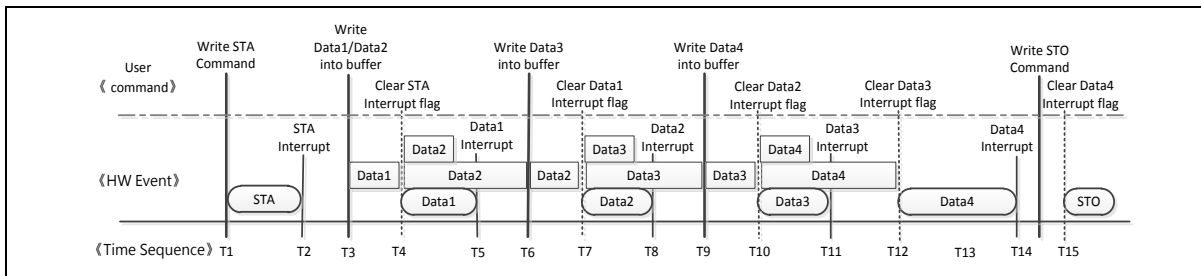


Figure 6.11-15 Timing of Two-level Buffer Transmit in Master Write

For example: if there are 4 data shall be received in Slave mode. The controller can receives the 2<sup>nd</sup> data in the I<sup>2</sup>C bus after the 1<sup>st</sup> data had been loaded into the received buffer and the user can read the 1<sup>st</sup> data after the 1<sup>st</sup> interrupt status be cleared.

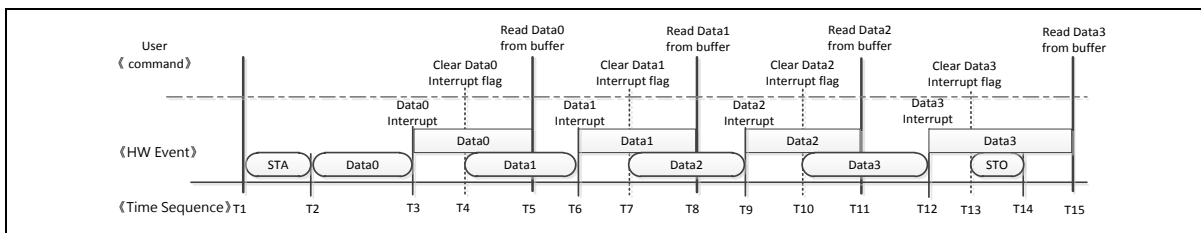


Figure 6.11-16 Timing of Two-level Buffer Transmit in Slave Read

If the buffer is not empty after the bus stop (STO), the user can set the TWOBUFEN bit or TWOBUFRST to '0' to make the buffer control state machine enter idle state and the buffer status will be set to the default value.

6.11.5.5 Programmable Setup and Hold Times

In order to guarantee a correct data setup and hold time, the timing must be configured. By programming HTCTL [5:0] (I2C\_TMCTL[11:6]) to configure hold time and STCTL [5:0] (I2C\_TMCTL[5:0]) 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 I<sup>2</sup>C protocol. Once setup time configuration greater than design limitation, that means if setup time setting make SCL output less than three PCLKs, I<sup>2</sup>C controller can't work normally due to SCL must sample three times. And once hold time configuration greater than I<sup>2</sup>C clock limitation, I<sup>2</sup>C will occur bus error. Suggest that user calculate suitable timing with baud rate and protocol before setting timing. Table 6.11-1 shows the relationship between I<sup>2</sup>C 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 our design, but user should follow I<sup>2</sup>C protocol standard.

I <sup>2</sup> C Baud Rate PCLK	100k	200k	400k	800k	1200k
12MHz	120	60	30	15	10
24MHz	240	120	60	30	20
48MHz	480	240	120	60	40
72MHz	720	360	180	90	60

Table 6.11-1 Relationship between I<sup>2</sup>C Baud Rate and PCLK

For setup time wrong adjustment example, we assume one SCL cycle contains 5 PCLKs and set STCTL [5:0] (I2C\_TMCTL[5:0]) to 3 that stretch three PCLKs for setup time setting. The setup time setting limitation:  $ST_{limit} = (I2C\_CLKDIV[7:0]+1) \times 2 - 6$ .

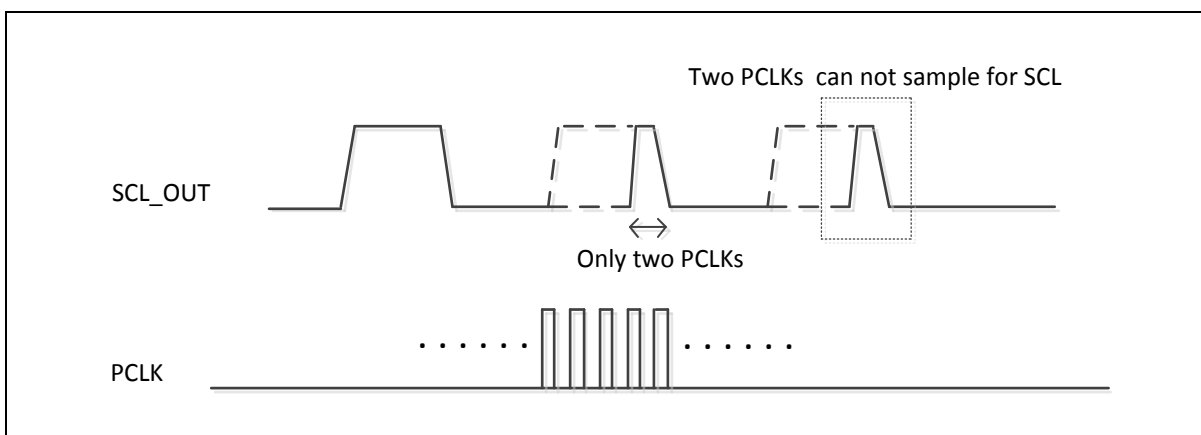


Figure 6.11-17 Setup Time Wrong Adjustment

For hold time wrong adjustment example, we use I<sup>2</sup>C Baud Rate = 1200k and PCLK = 72MHz, the SCL high/low duty = 60 PCLK. When we set HTCTL [5:0] (I2C\_TMCTL[11:6]) to 61 and STCTL [5:0] (I2C\_TMCTL[5:0]) to 0, then SDA output delay will over SCL high duty and cause bus error. The hold time setting limitation:  $HT_{limit} = (I2C\_CLKDIV[7:0]+1) \times 2 - 9$ .

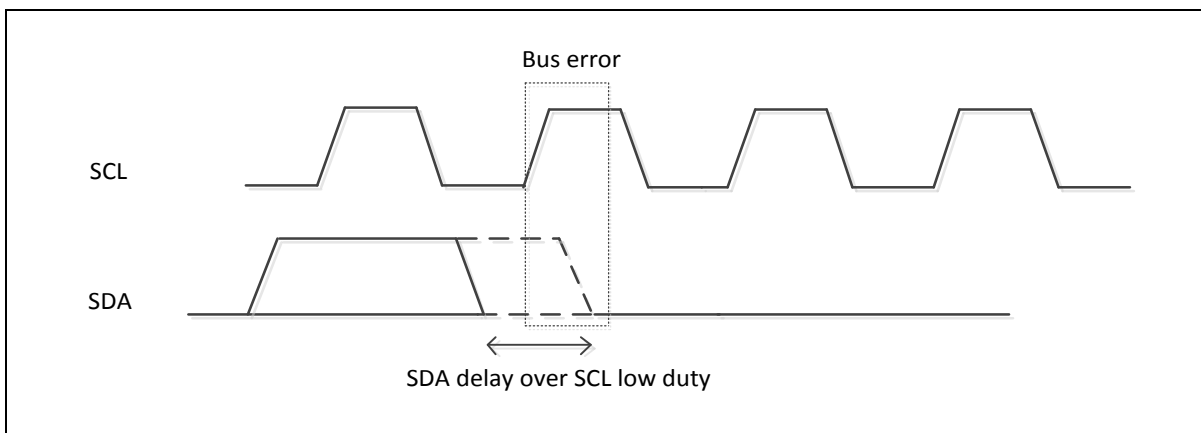


Figure 6.11-18 Hold Time Wrong Adjustment

### 6.11.5.6 I<sup>2</sup>C Protocol Registers

To control I<sup>2</sup>C port through the following fifteen special function registers: I2C\_CTL (control register), I2C\_STATUS (status register), I2C\_DAT (data register), I2C\_ADDRn (address registers, n=0~3),



I2C\_ADDRMSKn (address mask registers, n=0~3), I2C\_CLKDIV (clock divided 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 I<sup>2</sup>C port is equipped with four slave address registers, I2C\_ADDRn (n=0~3). The contents of the register are irrelevant when I<sup>2</sup>C 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 I<sup>2</sup>C hardware will react if the contents of I2C\_ADDRn are matched with the received slave address.

The I<sup>2</sup>C ports support the "General Call" function. If the GC bit (I2C\_ADDRn [0]) is set the I<sup>2</sup>C 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 I<sup>2</sup>C is in Slave mode, it can receive the general call address by 00H after Master send general call address to I<sup>2</sup>C bus, then it will follow status of GC mode.

**Slave Address Mask Registers (I2C\_ADDRMSK)**

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

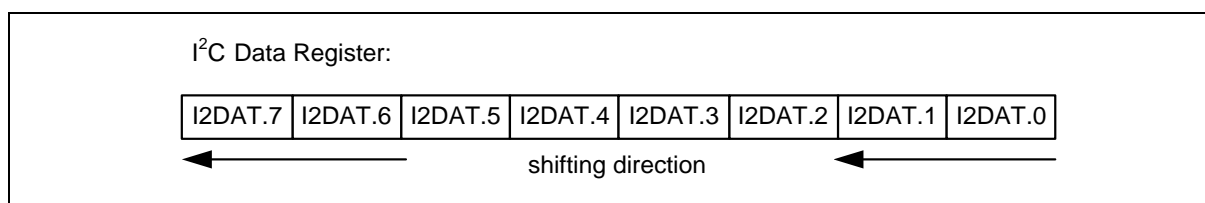


Figure 6.11-19 I<sup>2</sup>C Data Shifting Direction

**Control Register (I2C\_CTL)**

The CPU can be read from and written to I2C\_CTL [7:0] directly. When the I<sup>2</sup>C port is enabled by setting I2CEN (I2C\_CTL [6]) to high, the internal states will be controlled by I2C\_CTL and I<sup>2</sup>C logic hardware.

There are two bits are affected by hardware: the SI bit is set when the I<sup>2</sup>C 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\_STATUS, the I<sup>2</sup>C Interrupt Flag bit SI (I2C\_CTL [3]) will be set automatically. If the Enable Interrupt bit INTEN (I2C\_CTL [7]) is set at this time, the I<sup>2</sup>C interrupt will be generated. The bit field I2C\_STATUS[7:0] stores the internal state code, the content keeps stable until SI is cleared by software.

**Status Register (I2C\_STATUS)**

I2C\_STATUS [7:0] is an 8-bit read-only register. The bit field I2C\_STATUS [7:0] contains the status code and there are 26 possible status codes. All states are listed in Table 6.11-2. When I2C\_STATUS [7:0] is F8H, no serial interrupt is requested. All other I2C\_STATUS [7:0] values correspond to the defined I<sup>2</sup>C states. When each of these states is entered, a status interrupt is requested (SI = 1). A valid status code is present in I2C\_STATUS[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 I<sup>2</sup>C format frame. A Bus Error may occur during the serial transfer of an address byte, a data byte or an acknowledge bit. To recover I<sup>2</sup>C 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 I<sup>2</sup>C bus cannot recognize stop condition during this action when a bus error occurs.

Master Mode		Slave Mode	
STATUS	Description	STATUS	Description
0x08 <sup>[1]</sup>	Start	0xA0	Slave Transmit Repeat Start or Stop
0x10 <sup>[1]</sup>	Master Repeat Start	0xA8 <sup>[1]</sup>	Slave Transmit Address ACK
0x18 <sup>[1]</sup>	Master Transmit Address ACK	0xB8 <sup>[1]</sup>	Slave Transmit Data ACK
0x20	Master Transmit Address NACK	0xC0	Slave Transmit Data NACK
0x28 <sup>[1]</sup>	Master Transmit Data ACK	0xC8 <sup>[1]</sup>	Slave Transmit Last Data ACK
0x30	Master Transmit Data NACK	0x60 <sup>[1]</sup>	Slave Receive Address ACK
0x38	Master Arbitration Lost	0x68 <sup>[1]</sup>	Slave Receive Arbitration Lost
0x40 <sup>[1]</sup>	Master Receive Address ACK	0x80 <sup>[1]</sup>	Slave Receive Data ACK
0x48	Master Receive Address NACK	0x88	Slave Receive Data NACK
0x50 <sup>[1]</sup>	Master Receive Data ACK	0x70 <sup>[1]</sup>	GC mode Address ACK
0x58	Master Receive Data NACK	0x78 <sup>[1]</sup>	GC mode Arbitration Lost
0x00	Bus error	0x90 <sup>[1]</sup>	GC mode Data ACK
		0x98	GC mode Data NACK
		0xB0 <sup>[1]</sup>	Address Transmit Arbitration Lost
0xF8	Bus Released <b>Note:</b> Status "0xF8" exists in both master/slave modes, and it won't raise interrupt. <b>Tag [1]:</b> No interrupt in PDMA mode.		

Table 6.11-2 I<sup>2</sup>C Status Code Description

**Clock Divided Register (I2C\_CLKDIV)**

The data baud rate of I<sup>2</sup>C is determined by CLK(I2C\_CLKDIV [7:0]) register when I<sup>2</sup>C is in Master Mode, and it is not necessary in a Slave mode. In the Slave mode, I<sup>2</sup>C will automatically synchronize it with any clock frequency from master I<sup>2</sup>C device. In the slave mode, system clock frequency should be greater than I<sup>2</sup>C bus maximum clock 20 times.

The data baud rate of I<sup>2</sup>C setting is Data Baud Rate of I<sup>2</sup>C = (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 I<sup>2</sup>C = 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 I<sup>2</sup>C bus hang-up. If the time-out

counter is enabled, the counter starts up counting until it overflows (TOIF=1) and generates I<sup>2</sup>C 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 I<sup>2</sup>C bus hangs up, it causes the I2C\_STATUS and flag SI are not updated for a period, the 14-bit time-out counter may overflow and acknowledge CPU the I<sup>2</sup>C interrupt. Refer to Figure 6.11-20 for the 14-bit time-out counter. User may write 1 to clear TOIF to 0.

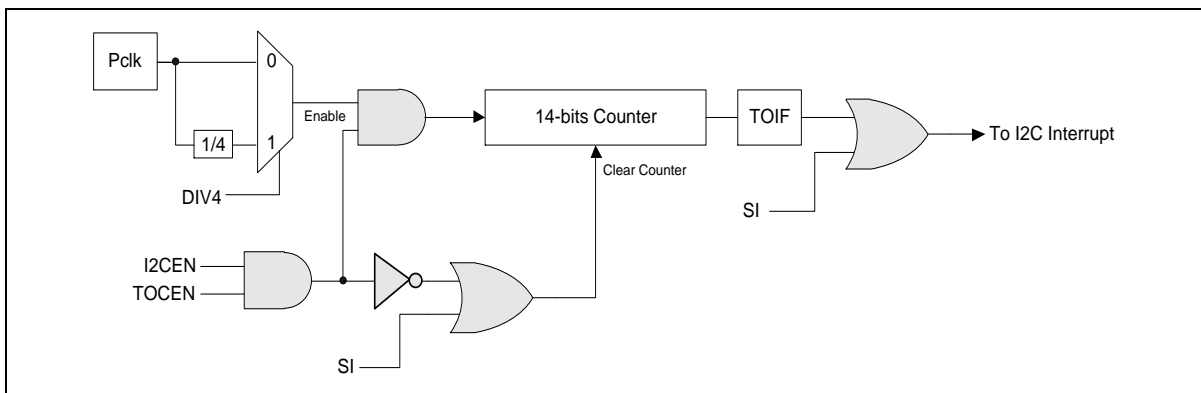


Figure 6.11-20 I<sup>2</sup>C Time-out Count Block Diagram

### Wake-up Control Register (I2C\_WKCTL)

When chip enters Power-down mode and set WKEN (WKCON [0]) to 1, other I<sup>2</sup>C master can wake up our chip by addressing our I<sup>2</sup>C 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 I<sup>2</sup>C controller will go ahead. If NHDBUSEN (WKCON [7]) is set, the controller will don't stretch the SCL to low. Note that when the controller don'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 I<sup>2</sup>C controller and execute the original operation again.

### Wake-up Status Register (I2C\_WKSTS)

When system is woken up by other I<sup>2</sup>C master device, WKIF (I2C\_WKSTS [0]) 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 can't release WKIF through clearing the WKAKDONE bit to 0.

The WRSTSWK (I2C\_WKSTS [2]) bit records the Read/Write command before the I<sup>2</sup>C controller send 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 wake-up by the address match frame. Note that the WRSTSWK (I2C\_WKSTS [2]) bit is cleared when write one to the WKAKDONE (I2C\_WKSTS [1]) bit.

When system is woken up by other I<sup>2</sup>C master device, WKIF is set to indicate this event. User needs write "1" to clear this bit.

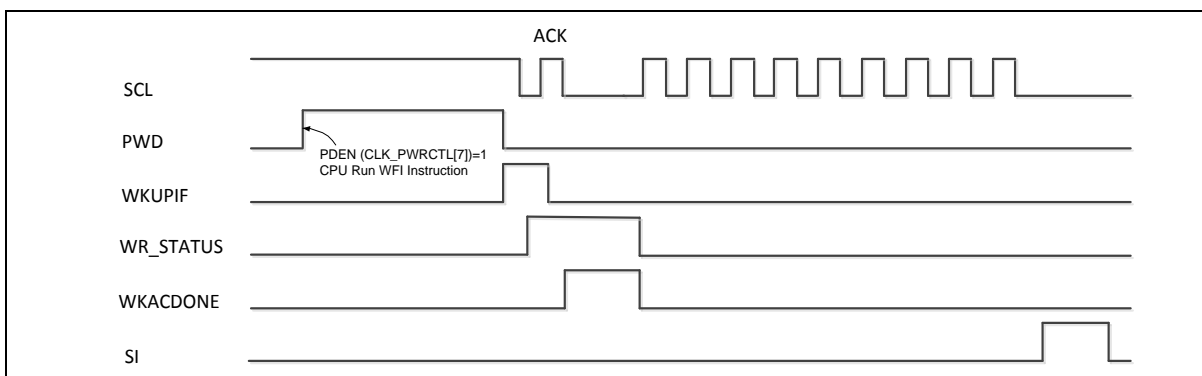


Figure 6.11-21 I<sup>2</sup>C Wake-Up Related Signals Waveform

### I<sup>2</sup>C Control Register 1 (I2C\_CTL1)

For the TWOBUFEN (I2C\_CTL1[5]) bit, it is used to enable the two-level buffer for I<sup>2</sup>C transmitted or received buffer. It is used to improve the performance of the I<sup>2</sup>C bus.

The two-level buffer status interrupt enable can be enabled by I2C\_CTL1[4:3] to generate the under run or over run event. It is worth noting when buffer empty, it will pull the bus. When I<sup>2</sup>C transmit or receive STOP, the data in buffer will be cleared. If user want to reset two-level buffer, then set the TWOBUFRST (I2C\_CTL1 [6]) to clear buffer.

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.

### I<sup>2</sup>C Status Register 1 (I2C\_STATUS1)

The two-level buffer status, busy free information, buffer empty, buffer full and overrun or under run are also list in the this register.

### I<sup>2</sup>C Timing Configure Control Register (I2C\_TMCTL)

To configure setup/hold time, the HTCTL [5:0] (I2C\_TMCTL[11:6]) and STCTL [5:0] (I2C\_TMCTL[5:0]) are set based on actual demand.

#### 6.11.5.7 Example for Random Read on EEPROM

The following steps are used to configure the I<sup>2</sup>C0 related registers when using I<sup>2</sup>C to read data from EEPROM.

1. Set I<sup>2</sup>C0 the multi-function pin in the SYS\_GPA\_MFPL or SYS\_GPD\_MFPL or SYS\_GPE\_MFPH or SYS\_GPA\_MFPH registers as SCL and SDA pins.
2. Enable I<sup>2</sup>C0 APB clock, I2C0CKEN=1 in the "CLK\_APBCLK0" register.
3. Set I2C0RST=1 to reset I<sup>2</sup>C0 controller then set I<sup>2</sup>C0 controller to normal operation, I2C0RST=0 in the "SYS\_IPRST1" register.
4. Set I2CEN=1 to enable I<sup>2</sup>C0 controller in the "I2C\_CTL" register.
5. Give I<sup>2</sup>C0 clock a divided register value for I<sup>2</sup>C clock rate in the "I2C\_CLKDIV".
6. Set SETENA=0x00000040 in the "NVIC\_IUSER" register to set I<sup>2</sup>C0 IRQ.
7. Set INTEN=1 to enable I<sup>2</sup>C0 Interrupt in the "I2C\_CTL" register.
8. Set I<sup>2</sup>C0 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 6.11-22 shows the EEPROM random read operation.

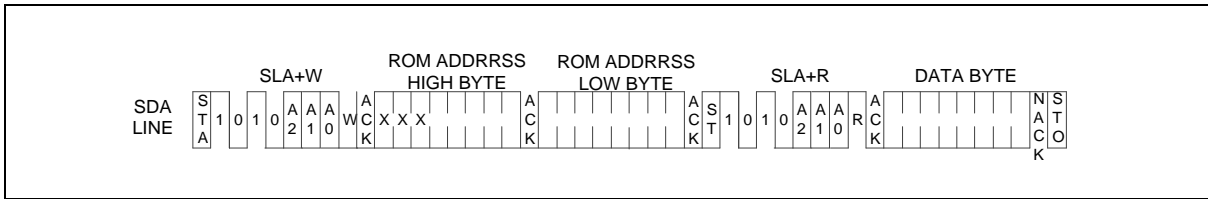


Figure 6.11-22 EEPROM Random Read

Figure 6.11-23 shows how to use I<sup>2</sup>C controller to implement the protocol of EEPROM random read.

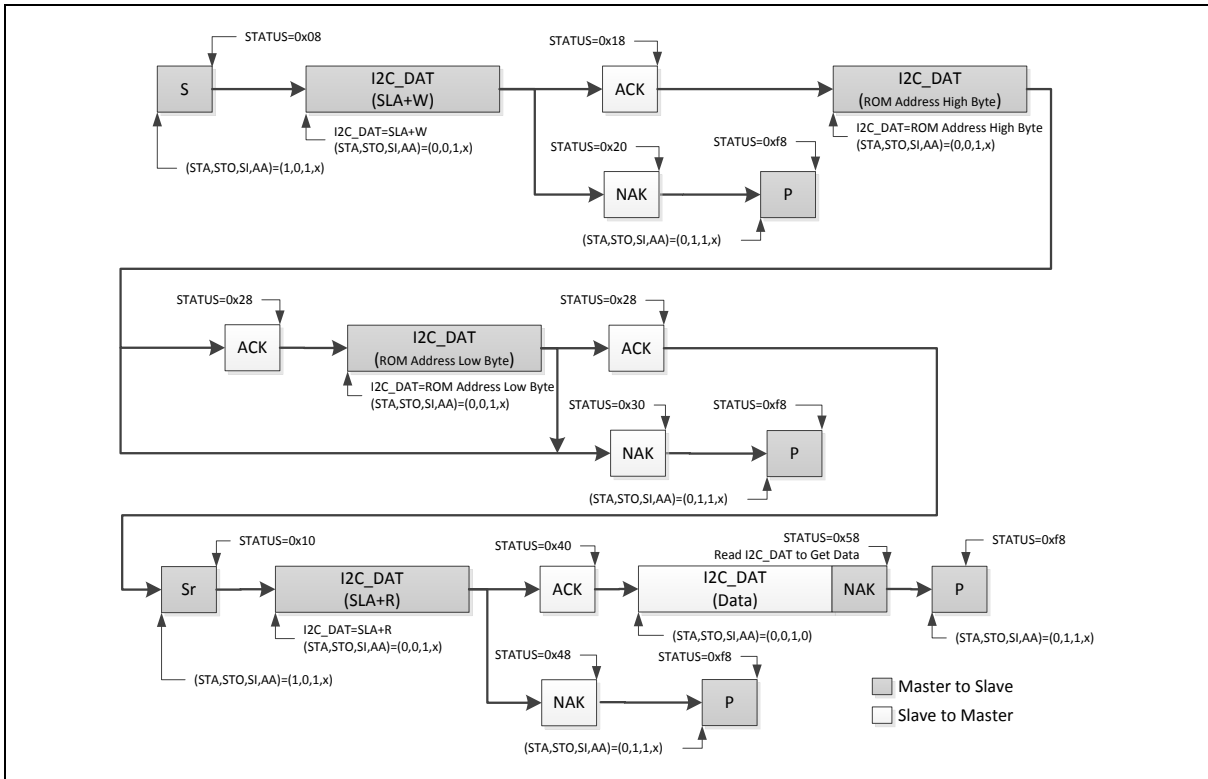


Figure 6.11-23 Protocol of EEPROM Random Read

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

### 6.11.6 Register Map

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

Register	Offset	R/W	Description	Reset Value
I <sup>2</sup> C Base Address: I2Cx_BA = 0x4002_0000 + (0x10_0000 *x) x= 0,1				
I2C_CTL	I2Cx_BA+0x00	R/W	I <sup>2</sup> C Control Register 0	0x0000_0000
I2C_ADDR0	I2Cx_BA+0x04	R/W	I <sup>2</sup> C Slave Address Register0	0x0000_0000
I2C_DAT	I2Cx_BA+0x08	R/W	I <sup>2</sup> C Data Register	0x0000_0000
I2C_STATUS	I2Cx_BA+0x0C	R	I <sup>2</sup> C Status Register 0	0x0000_00F8
I2C_CLKDIV	I2Cx_BA+0x10	R/W	I <sup>2</sup> C Clock Divided Register	0x0000_0000
I2C_TOCTL	I2Cx_BA+0x14	R/W	I <sup>2</sup> C Time-out Control Register	0x0000_0000
I2C_ADDR1	I2Cx_BA+0x18	R/W	I <sup>2</sup> C Slave Address Register1	0x0000_0000
I2C_ADDR2	I2Cx_BA+0x1C	R/W	I <sup>2</sup> C Slave Address Register2	0x0000_0000
I2C_ADDR3	I2Cx_BA+0x20	R/W	I <sup>2</sup> C Slave Address Register3	0x0000_0000
I2C_ADDRMSK0	I2Cx_BA+0x24	R/W	I <sup>2</sup> C Slave Address Mask Register0	0x0000_0000
I2C_ADDRMSK1	I2Cx_BA+0x28	R/W	I <sup>2</sup> C Slave Address Mask Register1	0x0000_0000
I2C_ADDRMSK2	I2Cx_BA+0x2C	R/W	I <sup>2</sup> C Slave Address Mask Register2	0x0000_0000
I2C_ADDRMSK3	I2Cx_BA+0x30	R/W	I <sup>2</sup> C Slave Address Mask Register3	0x0000_0000
I2C_WKCTL	I2Cx_BA+0x3C	R/W	I <sup>2</sup> C Wake-up Control Register	0x0000_0000
I2C_WKSTS	I2Cx_BA+0x40	R/W	I <sup>2</sup> C Wake-up Status Register	0x0000_0000
I2C_CTL1	I2Cx_BA+0x44	R/W	I <sup>2</sup> C Control Register 1	0x0000_0000
I2C_STATUS1	I2Cx_BA+0x48	R	I <sup>2</sup> C Status Register 1	0x0000_0000
I2C_TMCTL	I2Cx_BA+0x4C	R/W	I2C Timing Configure Control Register	0x0000_0000

6.11.7 Register Description

I<sup>2</sup>C Control Register (I2C\_CTL)

Register	Offset	R/W	Description	Reset Value
I2C_CTL	I2Cx_BA+0x00	R/W	I <sup>2</sup> 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	<b>Enable Interrupt</b> 0 = I <sup>2</sup> C interrupt Disabled. 1 = I <sup>2</sup> C interrupt Enabled.
[6]	I2CEN	<b>I<sup>2</sup>C Controller Enable Bit</b> Set to enable I <sup>2</sup> C serial function controller. When I2CEN=1 the I <sup>2</sup> C serial function enable. The multi-function pin function must set to SDA, and SCL of I <sup>2</sup> C function first. 0 = I <sup>2</sup> C serial function Disabled. 1 = I <sup>2</sup> C serial function Enabled.
[5]	STA	<b>I<sup>2</sup>C START Control</b> Setting STA to logic 1 to enter Master mode, the I <sup>2</sup> C hardware sends a START or repeat START condition to bus when the bus is free.
[4]	STO	<b>I<sup>2</sup>C STOP Control</b> In Master mode, setting STO to transmit a STOP condition to bus then I <sup>2</sup> C controller will check the bus condition if a STOP condition is detected. This bit will be cleared by hardware automatically.
[3]	SI	<b>I<sup>2</sup>C Interrupt Flag</b> When a new I <sup>2</sup> C state is present in the I2C_STATUS register, the SI flag is set by hardware. If bit INTEN (I2C_CTL [7]) is set, the I <sup>2</sup> C interrupt is requested. SI must be cleared by software. Clear SI by writing 1 to this bit.
[2]	AA	<b>Assert Acknowledge Control</b> 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.

**I<sup>2</sup>C Data Register (I2C\_DAT)**

Register	Offset	R/W	Description	Reset Value
I2C_DAT	I2Cx_BA+0x08	R/W	I <sup>2</sup> 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	I <sup>2</sup> C Data Bit [7:0] is located with the 8-bit transferred/received data of I <sup>2</sup> C serial port.



**I<sup>2</sup>C Status Register (I2C\_STATUS)**

Register	Offset	R/W	Description	Reset Value
I2C_STATUS	I2Cx_BA+0x0C	R	I <sup>2</sup> 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]	STATUS	<p><b>I<sup>2</sup>C Status</b></p> <p>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_STATUS is F8H, no serial interrupt is requested. Others I2C_STATUS values correspond to defined I<sup>2</sup>C states. When each of these states is entered, a status interrupt is requested (SI = 1). A valid status code is present in I2C_STATUS 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.</p>

**I<sup>2</sup>C Clock Divided Register (I2C\_CLKDIV)**

Register	Offset	R/W	Description	Reset Value
I2C_CLKDIV	I2Cx_BA+0x10	R/W	I <sup>2</sup> 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							
7	6	5	4	3	2	1	0
DIVIDER							

Bits	Description	
[31:8]	Reserved	Reserved.
[7:0]	DIVIDER	<p><b>I<sup>2</sup>C Clock Divided</b></p> <p>Indicates the I<sup>2</sup>C clock rate: Data Baud Rate of I<sup>2</sup>C = (system clock) / (4x (I2C_CLKDIV+1)).</p> <p><b>Note:</b> The minimum value of I2C_CLKDIV is 4.</p>

**I<sup>2</sup>C Time-out Control Register (I2C\_TOCTL)**

Register	Offset	R/W	Description	Reset Value
I2C_TOCTL	I2Cx_BA+0x14	R/W	I <sup>2</sup> 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	<p><b>Time-out Counter Enable Bit</b></p> <p>When Enabled, the 14-bit time-out counter will start counting when SI is clear. Setting flag SI to '1' will reset counter and re-start up counting after SI is cleared.</p> <p>0 = Time-out counter Disabled.</p> <p>1 = Time-out counter Enabled.</p>
[1]	TOCDIV4	<p><b>Time-out Counter Input Clock Divided by 4</b></p> <p>When Enabled, The time-out period is extend 4 times.</p> <p>0 = Time-out period is extend 4 times Disabled.</p> <p>1 = Time-out period is extend 4 times Enabled.</p>
[0]	TOIF	<p><b>Time-out Flag</b></p> <p>This bit is set by hardware when I<sup>2</sup>C time-out happened and it can interrupt CPU if I<sup>2</sup>C interrupt enable bit (INTEN) is set to 1.</p> <p><b>Note:</b> Software can write 1 to clear this bit.</p>

**I<sup>2</sup>C Slave Address Register (ADDRx)**

Register	Offset	R/W	Description	Reset Value
I2C_ADDR0	I2Cx_BA+0x04	R/W	I <sup>2</sup> C Slave Address Register0	0x0000_0000
I2C_ADDR1	I2Cx_BA+0x18	R/W	I <sup>2</sup> C Slave Address Register1	0x0000_0000
I2C_ADDR2	I2Cx_BA+0x1C	R/W	I <sup>2</sup> C Slave Address Register2	0x0000_0000
I2C_ADDR3	I2Cx_BA+0x20	R/W	I <sup>2</sup> 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							
7	6	5	4	3	2	1	0
ADDR							GC

Bits	Description	
[31:8]	Reserved	Reserved.
[7:1]	ADDR	<b>I<sup>2</sup>C Address</b> The content of this register is irrelevant when I <sup>2</sup> 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 <sup>2</sup> C hardware will react if either of the address is matched.
[0]	GC	<b>General Call Function</b> 0 = General Call Function Disabled. 1 = General Call Function Enabled.

**I<sup>2</sup>C Slave Address Mask Register (ADDRMSKx)**

Register	Offset	R/W	Description	Reset Value
I2C_ADDRMSK0	I2Cx_BA+0x24	R/W	I <sup>2</sup> C Slave Address Mask Register0	0x0000_0000
I2C_ADDRMSK1	I2Cx_BA+0x28	R/W	I <sup>2</sup> C Slave Address Mask Register1	0x0000_0000
I2C_ADDRMSK2	I2Cx_BA+0x2C	R/W	I <sup>2</sup> C Slave Address Mask Register2	0x0000_0000
I2C_ADDRMSK3	I2Cx_BA+0x30	R/W	I <sup>2</sup> 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							
7	6	5	4	3	2	1	0
ADDRMSK							Reserved

Bits	Description	
[31:8]	Reserved	Reserved.
[7:1]	ADDRMSK	<p><b>I<sup>2</sup>C Address Mask</b></p> <p>0 = Mask Disabled (the received corresponding register bit should be exact the same as address register.).</p> <p>1 = Mask Enabled (the received corresponding address bit is don't care.).</p> <p>I<sup>2</sup>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.</p>
[0]	Reserved	Reserved.

**I<sup>2</sup>C Wake-up Control Register (I2C\_WKCTL)**

Register	Offset	R/W	Description	Reset Value
I2C_WKCTL	I2Cx_BA+0x3C	R/W	I <sup>2</sup> 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	<p><b>I<sup>2</sup>C No Hold BUS Enable Bit</b></p> <p>0 = I<sup>2</sup>C don't hold bus after wake-up disable. 1 = I<sup>2</sup>C don't hold bus after wake-up enable.</p> <p><b>Note:</b> I<sup>2</sup>C controller could response when WKIF event is not clear, it may cause error data transmitted or received. If data transmitted or received when WKIF event is not clear, user must reset I<sup>2</sup>C controller and execute the original operation again.</p>
[6:1]	Reserved	Reserved.
[0]	WKEN	<p><b>I<sup>2</sup>C Wake-up Enable Bit</b></p> <p>0 = I<sup>2</sup>C wake-up function Disabled. 1 = I<sup>2</sup>C wake-up function Enabled.</p>

**I<sup>2</sup>C Wake-up Status Register (I2C\_WKSTS)**

Register	Offset	R/W	Description	Reset Value
I2C_WKSTS	I2Cx_BA+0x40	R/W	I <sup>2</sup> 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	<p><b>Read/Write Status Bit in Address Wakeup Frame</b></p> <p>0 = Write command be record on the address match wakeup frame.                      1 = Read command be record on the address match wakeup frame.</p> <p><b>Note:</b> This bit will be cleared when software can write 1 to WKAKDONE bit.</p>
[1]	WKAKDONE	<p><b>Wakeup Address Frame Acknowledge Bit Done</b></p> <p>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.</p> <p><b>Note:</b> This bit can't release WKIF. Software can write 1 to clear this bit.</p>
[0]	WKIF	<p><b>I<sup>2</sup>C Wake-up Flag</b></p> <p>When chip is woken up from Power-down mode by I<sup>2</sup>C, this bit is set to 1. Software can write 1 to clear this bit.</p>

I<sup>2</sup>C Control Register 1 (I2C\_CTL1)

Register	Offset	R/W	Description	Reset Value
I2C_CTL1	I2Cx_BA+0x44	R/W	I <sup>2</sup> 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							PDMASTR
7	6	5	4	3	2	1	0
NSTRETCH	TWOBUFRST	TWOBUFEN	UDRIEN	OVRIEN	PDMARST	RXPDMAEN	TXPDMAEN

Bits	Description	
[31:9]	Reserved	Reserved.
[8]	PDMASTR	<b>PDMA Stretch Bit</b> 0 = I <sup>2</sup> C sends STOP automatically after PDMA transfer done. (only master TX) 1 = I <sup>2</sup> C SCL bus is stretched by hardware after PDMA transfer done if the SI is not cleared. (only master TX)
[7]	NSTRETCH	<b>No Stretch on the I<sup>2</sup>C Bus</b> 0 = I <sup>2</sup> C SCL bus is stretched by hardware if the SI is not cleared in master mode. 1 = I <sup>2</sup> C SCL bus is not stretched by hardware if the SI is not cleared in master mode.
[6]	TWOBUFRST	<b>Two-level Buffer Reset</b> 0 = No effect. 1 = Reset the related counters, two-level buffer state machine, and the content of data buffer.
[5]	TWOBUFEN	<b>Two-level Buffer Enable Bit</b> 0 = Two-level buffer Disabled. 1 = Two-level buffer Enabled. Set to enable the two-level buffer for I <sup>2</sup> C transmitted or received buffer. It is used to improve the performance of the I <sup>2</sup> C bus. If this bit is set = 1, the control bit of STA for repeat start or STO bit should be set after the current SI is clear. For example: if there are 4 data shall be transmitted and then stop it. The STO bit shall be set after the 3 <sup>rd</sup> data's SI event being clear. In this time, the 4 <sup>th</sup> data can be transmitted and the I <sup>2</sup> C stop after the 4 <sup>th</sup> data transmission done.
[4]	UDRIEN	<b>I<sup>2</sup>C Under Run Interrupt Control Bit</b> Setting UDRIEN to logic 1 will send a interrupt to system when the TWOBUFEN bit is enabled and there is under run event happened in transmitted buffer.
[3]	OVRIEN	<b>I<sup>2</sup>C over Run Interrupt Control Bit</b> Setting OVRIEN to logic 1 will send a interrupt to system when the TWOBUFEN bit is enabled and there is over run event in received buffer.



[2]	<b>PDMARST</b>	<b>PDMA Reset</b> 0 = No effect. 1 = Reset the PDMA control logic. This bit will be cleared to 0 automatically.
[1]	<b>RXPDMAEN</b>	<b>PDMA Receive Channel Available</b> 0 = Receive PDMA function Disabled. 1 = Receive PDMA function Enabled.
[0]	<b>TXPDMAEN</b>	<b>PDMA Transmit Channel Available</b> 0 = Transmit PDMA function Disabled. 1 = Transmit PDMA function Enabled.

**I<sup>2</sup>C Status Register 1 (I2C\_STATUS1)**

Register	Offset	R/W	Description	Reset Value
I2C_STATUS1	I2Cx_BA+0x48	R	I <sup>2</sup> 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
UDR	OVR	EMPTY	FULL	Reserved			

Bits	Description	
[31:9]	Reserved	Reserved.
[8]	ONBUSY	<p><b>on Bus Busy</b></p> <p>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.</p> <p>0 = The bus is IDLE (both SCLK and SDA High).</p> <p>1 = The bus is busy.</p>
[7]	UDR	<p><b>I<sup>2</sup>C Under Run Status Bit</b></p> <p>This bit indicates the transmitted two-level buffer TX or RX is under run when the TWOBUFEN = 1.</p>
[6]	OVR	<p><b>I<sup>2</sup>C over Run Status Bit</b></p> <p>This bit indicates the received two-level buffer TX or RX is over run when the TWOBUFEN = 1.</p>
[5]	EMPTY	<p><b>Two-level Buffer Empty</b></p> <p>This bit indicates two-level buffer TX or RX empty or not when the TWOBUFEN = 1.</p> <p>This bit is set when POINTER is equal to 0.</p>
[4]	FULL	<p><b>Two-level Buffer Full</b></p> <p>This bit indicates two-level buffer TX or RX full or not when the TWOBUFEN = 1.</p> <p>This bit is set when POINTER is equal to 2</p>
[3:0]	Reserved	Reserved.

**I<sup>2</sup>C Timing Configure Control Register (I2C\_TMCTL)**

Register	Offset	R/W	Description	Reset Value
I2C_TMCTL	I2Cx_BA+0x4C	R/W	I <sup>2</sup> C Timing Configure 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				HTCTL			
7	6	5	4	3	2	1	0
HTCTL		STCTL					

Bits	Description	
[31:12]	Reserved	Reserved.
[11:6]	HTCTL	<p><b>Hold Time Configure Control Register</b></p> <p>This field is used to generate the delay timing between SCL falling edge and SDA rising edge in transmission mode.</p> <p>The delay hold time is numbers of peripheral clock = HTCTL x PCLK.</p>
[5:0]	STCTL	<p><b>Setup Time Configure Control Register</b></p> <p>This field is used to generate a delay timing between SDA falling edge and SCL rising edge in transmission mode.</p> <p>The delay setup time is numbers of peripheral clock = STCTL x PCLK.</p> <p><b>Note:</b> Setup time setting should not make SCL output less than three PCLKs.</p>

## 6.12 PDMA Controller (PDMA)

### 6.12.1 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. The PDMA supports time-out function for channel 0 and channel 1.

### 6.12.2 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
- Supports software and SPI, UART, I<sup>2</sup>S, I<sup>2</sup>C, USB, ADC, PWM and TIMER request
- 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 for channel0 and channel 1

### 6.12.3 Block Diagram

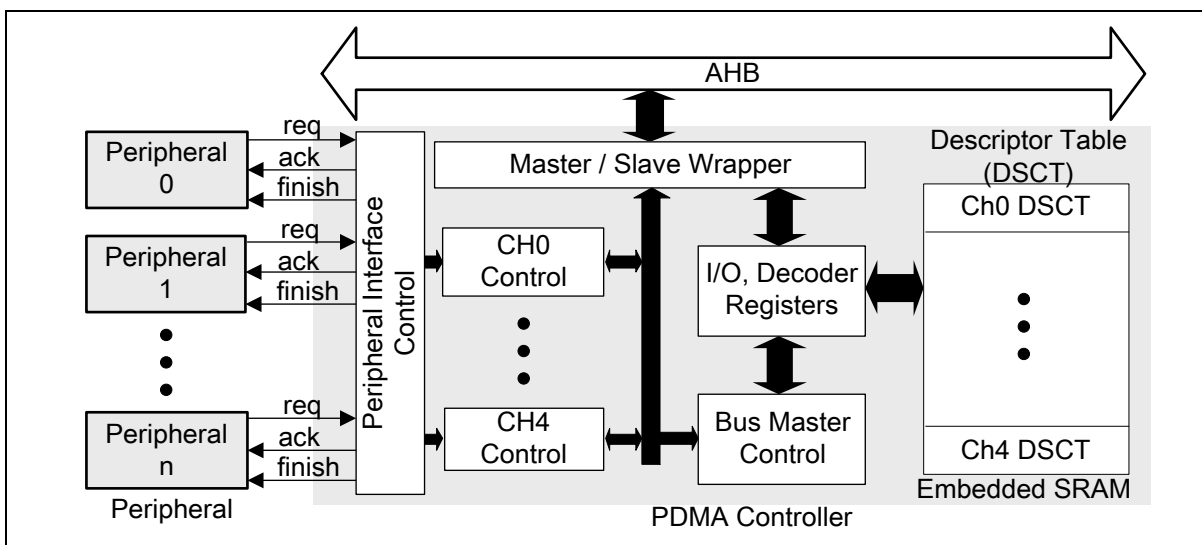


Figure 6.12-1 PDMA Controller Block Diagram

### 6.12.4 Basic Configuration

- Clock Source Configuration
  - Enable PDMA controller clock in PDMACKEN(CLK\_AHBCLK[1]).
- Reset Configuration
  - Reset PDMA controller in PDMARST (SYS\_IPRST0[2]).

### 6.12.5 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 6.12-2 shows the diagram of descriptor table (DSCT) data structure.

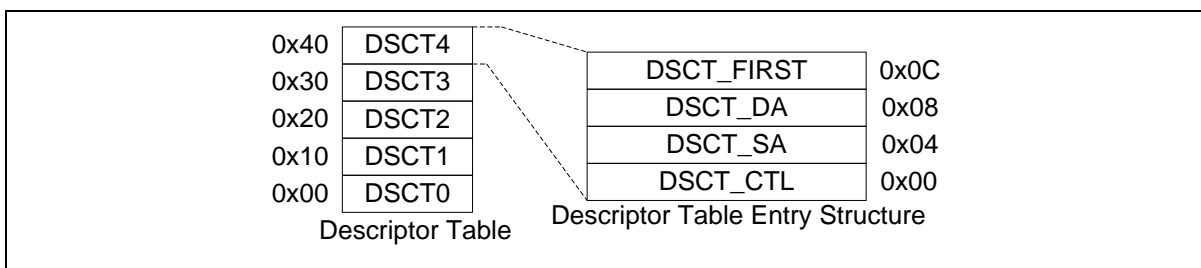


Figure 6.12-2 Descriptor Table Entry Structure

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).

PDMA time-out function is used to generate interrupt, while corresponding channel has been time-out.

#### 6.12.5.1 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.

A summary priority order is listed in Table 6.12-1. This table list all possible PDMA\_PRISET and channel number combination and shows the their priority from highest to lowest.

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 6.12-1 Channel Priority Table

6.12.5.2 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 that shown as Figure 6.12-3. This mode can be used to transfer data between memory and memory or peripherals and memory. PDMA controller operation mode can be set from OPMODE (PDMA\_DSCTn\_CTL[1:0], n denotes PDMA channel). PDMA will operation in idle mode by configured OPMODE bits to 1, the default setting is in idle state (OPMODE (PDMA\_DSCTn\_CTL[1:0]) = 0) 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 enable the transfer channel CHENn (PDMA\_CHCTL[4:0]) and fill the transfer count TXCNT (PDMA\_DSCTn\_CTL[29:16]) 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 TXYTYPE (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 each PDMA interrupt bit INTENn (PDMA\_INTEN[4:0]) is enabled and the operation mode will be updated to idle state as shown in Figure 6.12-4.

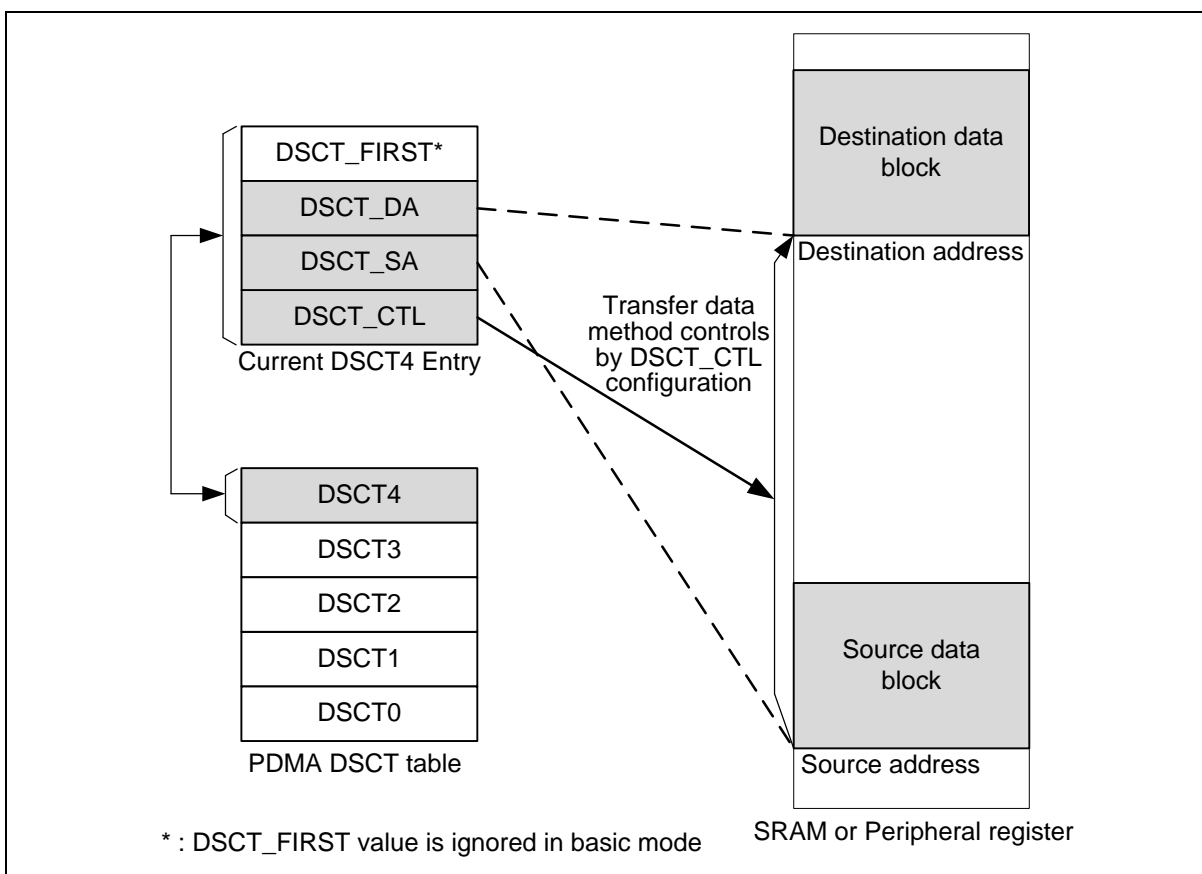


Figure 6.12-3 Descriptor Table Operation in Basic Mode

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 each PDMA interrupt bit is enabled.

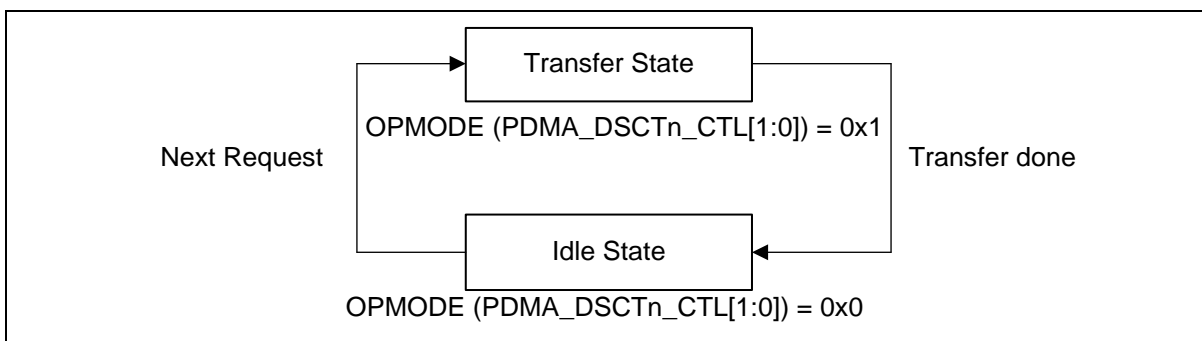


Figure 6.12-4 Basic Mode Finite State Machine

**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 6.12-4 Descriptor Table Link List Structure.

In Scatter-Gather mode, the first table entry is just used for jumping to the next table entry, the first task will not perform any operation transfer. Finishing each task will generate an interrupt to CPU if corresponding PDMA interrupt bit is enabled and TBINTDIS (PDMA\_DSCTn\_CTL[7]) bit is “0” (when finishing task and TBINTDIS bit is “0”, corresponding TDIFn (PDMA\_TDSTS[4:0]) flag will be asserted and if this bit is “1” TDIFn will not be active).

If channel 4 has been triggered as Figure 6.12-4 Descriptor Table Link List Structure, 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\_FIRST (link offset) and PDMA\_SCATBA (base address) registers. The base address is 0x2000\_0000 (only MSB 16bits valid in PDMA\_SCATBA), the first link offset is 0x0000\_0100 (only LSB 16bits in PDMA\_DSCTn\_FIRST), then current DSCT entry PDMA\_CURSCAT4 will update to address 0x2000\_0100 and copy DSCT\_CTL, DSCT\_SA, DSCT\_DA and DSCT\_NEXT from SRAM to PDMA\_DSCT4\_CTL, PDMA\_DSCT4\_SA, PDMA\_DSCT4\_DA and PDMA\_DSCT4\_FIRST register for execution the task of this table.

Furthermore, the low half-word of next DSCT\_NEXT in SRAM will load to the high half-word of PDMA\_DSCT4\_FIRST register. For example, if current table is the first DSCT4 entry, PDMA\_DSCT4\_FIRST will become 0x0200\_0100, high half-word is next table offset.

After finished the task, PDMA will load the information from next table address 0x2000\_0200, which is calculated by adding base address 0x2000\_0000 and next offset 0x0200 from SRAM.

Note that after each task of description link list table has been finished, the content of transfer count and operation mode in table will be cleared to 0 by hardware. To use the same link list table for transfer, user must reconfigure transfer count and operation mode.

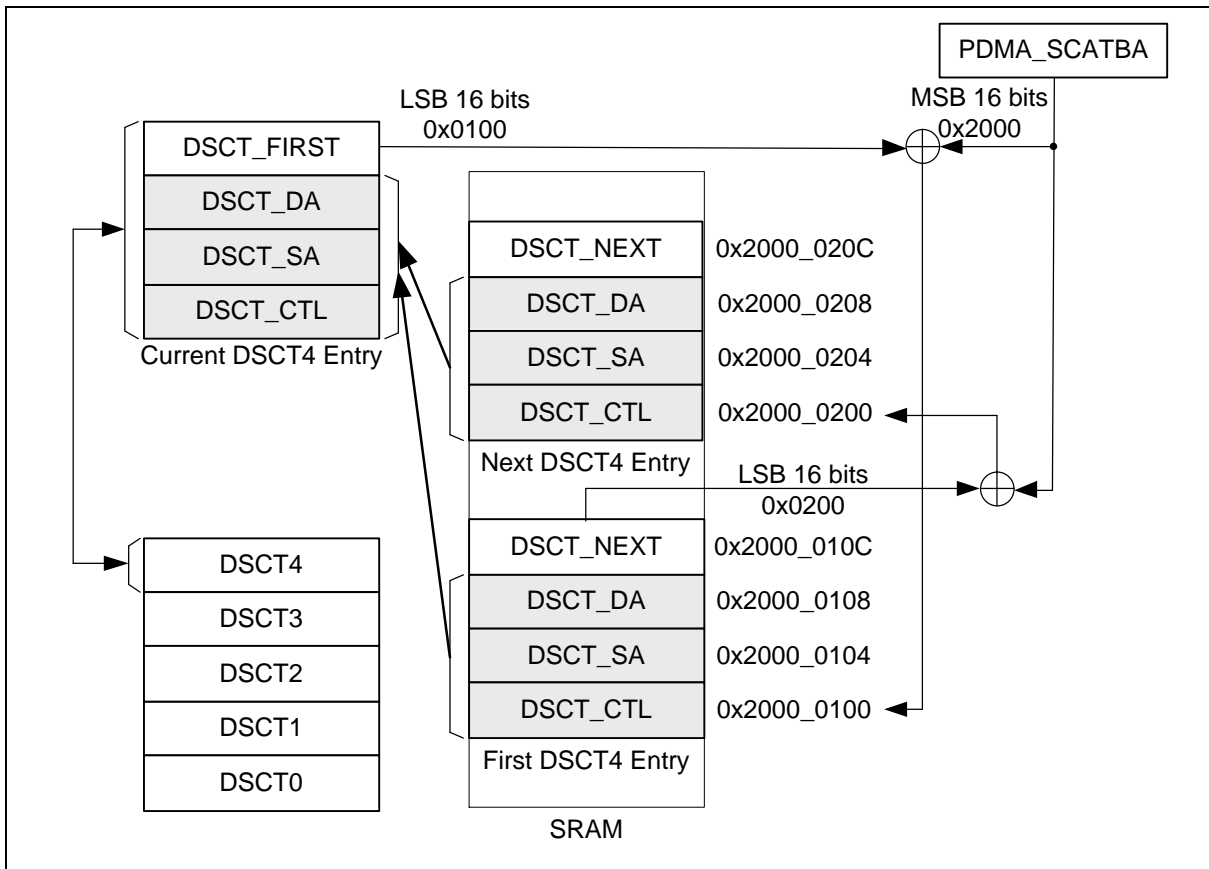


Figure 6.12-4 Descriptor Table Link List Structure

The above link list table operation is DSCT state in Scatter-Gather Mode as shown in Figure 6.12-5. 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 in the Scatter-Gather mode, the hardware will catch the next PDMA information block when the current task is finished. The Scatter-Gather mode stops until the PDMA controller operation mode switch to basic mode and transfer once or directly switch to idle state.

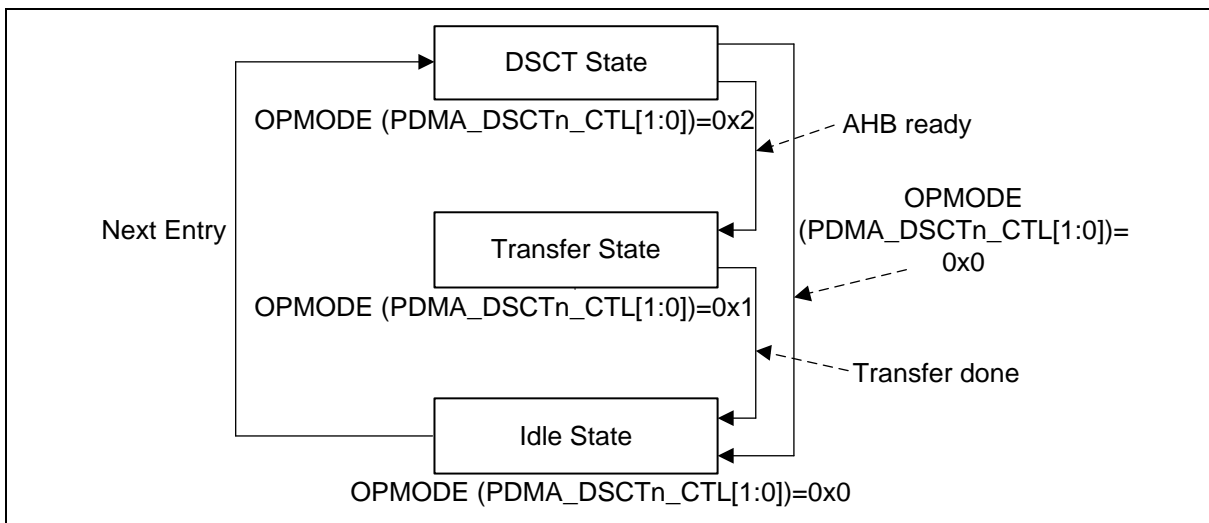


Figure 6.12-5 Scatter-Gather Mode Finite State Machine



Through Scatter-Gather mode, user can perform peripheral wrapper-around, multiple PDMA tasks or can be used for data transfer between varied locations in system memory instead of a set of contiguous locations.

The Ping-Pong audio buffer for I<sup>2</sup>S can be implemented by the wrapper-around two link list tables. The audio buffer can be divide to two parts that assign to two link list tables for PDMA transfer, then PDMA controller can loop around two buffer for transferring audio data to I<sup>2</sup>S TX buffer.

One PDMA task is controlled by one description link list table, so multiple description link list tables can make PDMA controller to perform multiple transfer tasks which the source/destination address can be different for each task, and it just uses one PDMA channel. Besides, the transfer block of multiple PDMA tasks can be scatter blocks in system memory.

### 6.12.5.3 Transfer Type

The PDMA controller supports two transfer types: single transfer type and burst transfer type. They are configured by setting TXTYPE (PDMA\_DSCTn\_CTL[2]) bit.

When PDMA controller operated in single transfer type, each transfer data needs one request signal which is comes from peripheral for one transfer, after transferred data, and TXCNT (PDMA\_DSCTn\_CTL[29:16]) bits will decrease 1. Transfer task will be finished after received TXCNT request signals, then TXCNT (PDMA\_DSCTn\_CTL[29:16]) bits are decrease to 0. In this mode, the BURSIZE (PDMA\_DSCTn\_CTL[6:4]) bits are not useful to control the transfer size, because of the settings of BURSIZE bits will be ignored by hardware.

For the burst transfer type, PDMA controller transfers TXCNT (PDMA\_DSCTn\_CTL[29:16]) of data and only need one request signal from software corresponding channel request SWREQn (PDMA\_SWREQ[4:0]). After transferred BURSIZE (PDMA\_DSCTn\_CTL[6:4]) of data, TXCNT (PDMA\_DSCTn\_CTL[29:16]) will decrease BURSIZE number. Transfer task will done until the transfer count TXCNT (PDMA\_DSCTn\_CTL[29:16]) decrease 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 6.12-6 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[29:16]) = 127, TXWIDTH(PDMA\_DSCTn\_CTL[13:12]) = 0(1 byte). Channel 0 uses burst transfer type, BURSIZE (PDMA\_DSCTn\_CTL[6:4]) = 0(128 transfers), TXCNT (PDMA\_DSCTn\_CTL[29:16]) = 255 and TXWIDTH(PDMA\_DSCTn\_CTL[13:12]) = 2(1 word). 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 transfer width is one byte, so PDMA controller will only transfer 1 byte data.
3. Then, PDMA controller turns to the channel 0 and loads channel 0's descriptor table. The channel 0 is burst transfer type, the burst size selected to 128 and transfer width is one word. Therefore, PDMA controller will transfer 128 words data.
4. When channel 0 transfers 128 words data, channel 1 gets another request signal, then after channel 0 finishes 128 words transferred, the PDMA controller will turn to channel 1 and transfer next 1 byte data.
5. After channel 1 transferred second 1 byte data, PDMA controller switches to low priority channel 0 to continuous next 128 words data transfer. If no channel 1 request receives, PDMA will start to transfer next 128 words data for channel 0.
6. PDMA controller will complete transfer when channel 0 finishes data transfer 256 words that received one request, and channel 1 finishes transferring 128 bytes after received 128 requests.

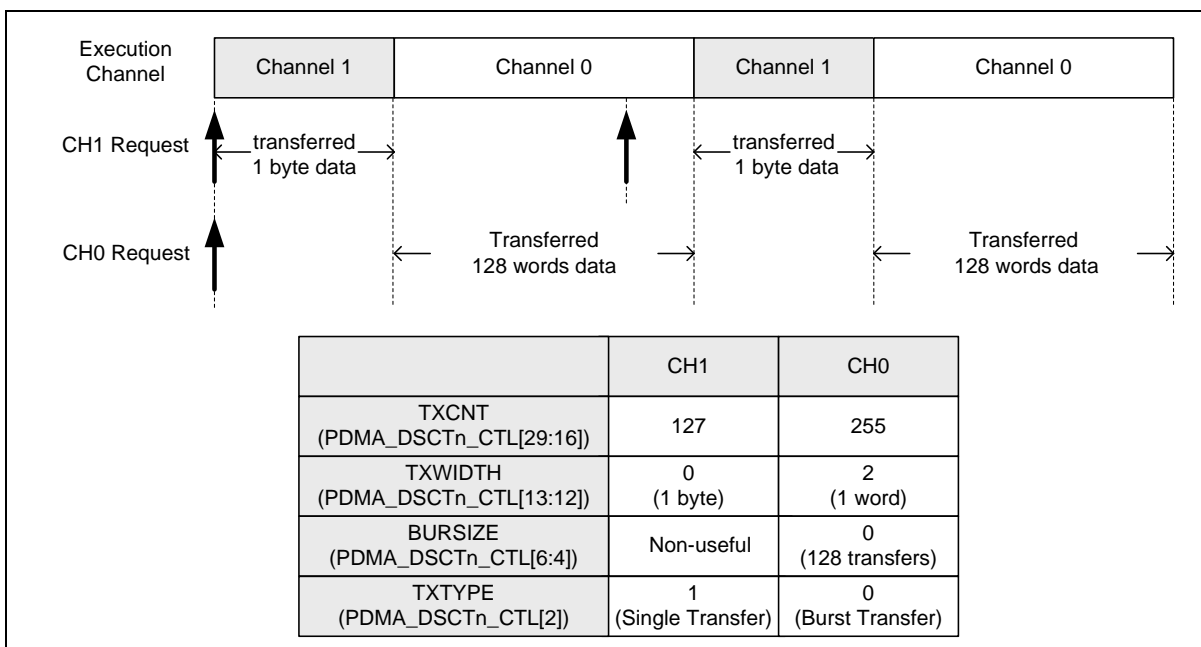


Figure 6.12-6 Example of Single Transfer Type and Burst Transfer Type in Basic Mode

#### 6.12.5.4 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 reset to 0 while counter count to TOCn (PDMA\_TOC0\_1 [16(n+1)-1:16n], n=0,1), received trigger signal, time-out function disabled or chip enter power-down mode.

Figure 6.12-7 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 set TOUTEN0(PDMA\_TOUTEN[0]) bit to 1.
2. Time-out counter is start counting from 0 to the value of TOC0(PDMA\_TOC0\_1[15:0]) bits when received 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 from 0 to 5, and user can clear REQTOF0 flag then polling REQTOF0 flag to check next time-out occurred.
5. Time-out counter is reset to 0 when time-out function is disabled.

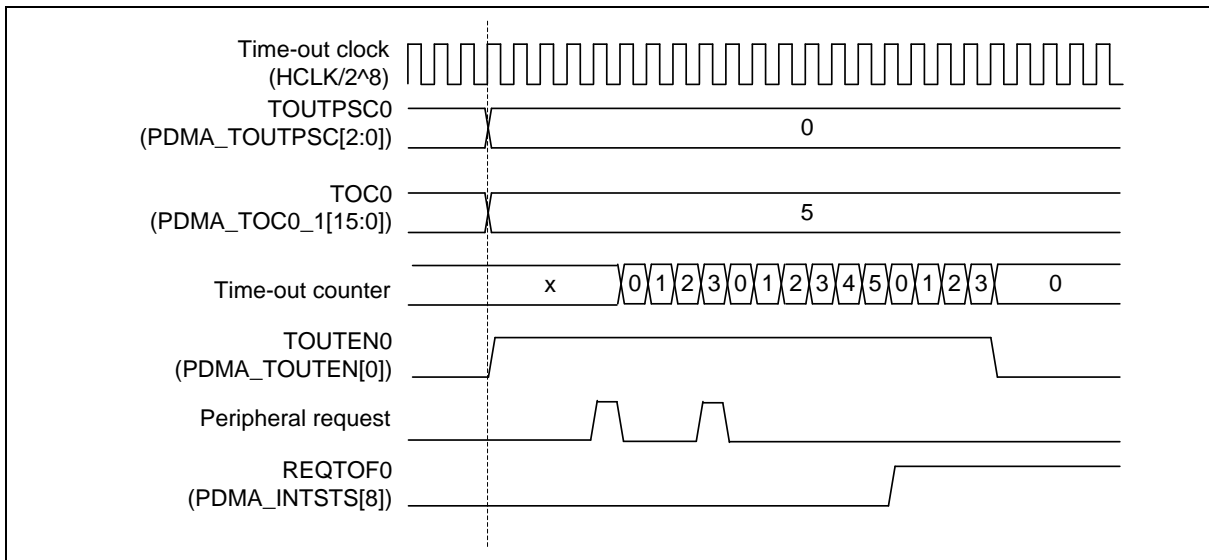


Figure 6.12-7 Example of PDMA Channel 0 Time-out Counter Operation

6.12.6 Register Map

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

Register	Offset	R/W	Description	Reset Value
<b>PDMA Base Address:</b>				
<b>PDMA_BA = 0x5000_8000</b>				
PDMA_DSCT0_CTL	PDMA_BA + 0x000	R/W	Descriptor Table Control Register of PDMA Channel 0	0xXXXX_XXXX
PDMA_DSCT0_SA	PDMA_BA + 0x004	R/W	Source Address Register of PDMA Channel 0	0xXXXX_XXXX
PDMA_DSCT0_DA	PDMA_BA + 0x008	R/W	Destination Address Register of PDMA Channel 0	0xXXXX_XXXX
PDMA_DSCT0_FIR ST	PDMA_BA + 0x00C	R/W	First Scatter-Gather Descriptor Table Offset of PDMA Channel 0	0xXXXX_XXXX
PDMA_DSCT1_CTL	PDMA_BA + 0x010	R/W	Descriptor Table Control Register of PDMA Channel 1	0xXXXX_XXXX
PDMA_DSCT1_SA	PDMA_BA + 0x014	R/W	Source Address Register of PDMA Channel 1	0xXXXX_XXXX
PDMA_DSCT1_DA	PDMA_BA + 0x018	R/W	Destination Address Register of PDMA Channel 1	0xXXXX_XXXX
PDMA_DSCT1_FIR ST	PDMA_BA + 0x01C	R/W	First Scatter-Gather Descriptor Table Offset of PDMA Channel 1	0xXXXX_XXXX
PDMA_DSCT2_CTL	PDMA_BA + 0x020	R/W	Descriptor Table Control Register of PDMA Channel 2	0xXXXX_XXXX
PDMA_DSCT2_SA	PDMA_BA + 0x024	R/W	Source Address Register of PDMA Channel 2	0xXXXX_XXXX
PDMA_DSCT2_DA	PDMA_BA + 0x028	R/W	Destination Address Register of PDMA Channel 2	0xXXXX_XXXX
PDMA_DSCT2_FIR ST	PDMA_BA + 0x02C	R/W	First Scatter-Gather Descriptor Table Offset of PDMA Channel 2	0xXXXX_XXXX
PDMA_DSCT3_CTL	PDMA_BA + 0x030	R/W	Descriptor Table Control Register of PDMA Channel 3	0xXXXX_XXXX
PDMA_DSCT3_SA	PDMA_BA + 0x034	R/W	Source Address Register of PDMA Channel 3	0xXXXX_XXXX
PDMA_DSCT3_DA	PDMA_BA + 0x038	R/W	Destination Address Register of PDMA Channel 3	0xXXXX_XXXX
PDMA_DSCT3_FIR ST	PDMA_BA + 0x03C	R/W	First Scatter-Gather Descriptor Table Offset of PDMA Channel 3	0xXXXX_XXXX
PDMA_DSCT4_CTL	PDMA_BA + 0x040	R/W	Descriptor Table Control Register of PDMA Channel 4	0xXXXX_XXXX
PDMA_DSCT4_SA	PDMA_BA + 0x044	R/W	Source Address Register of PDMA Channel 4	0xXXXX_XXXX
PDMA_DSCT4_DA	PDMA_BA + 0x048	R/W	Destination Address Register of PDMA Channel 4	0xXXXX_XXXX
PDMA_DSCT4_FIR ST	PDMA_BA + 0x04C	R/W	First Scatter-Gather Descriptor Table Offset of PDMA Channel 4	0xXXXX_XXXX
PDMA_CURSCAT0	PDMA_BA + 0x050	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 0	0xXXXX_XXXX
PDMA_CURSCAT1	PDMA_BA + 0x054	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 1	0xXXXX_XXXX
PDMA_CURSCAT2	PDMA_BA + 0x058	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 2	0xXXXX_XXXX
PDMA_CURSCAT3	PDMA_BA + 0x05C	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 3	0xXXXX_XXXX

PDMA_CURSCAT4	PDMA_BA + 0x060	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 4	0xXXXX_XXXX
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_SCATSTS	PDMA_BA + 0x428	R/W	PDMA Scatter-Gather Table Empty 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 Channel 0 and Channel 1 Time-out Counter Register	0x0000_0000
PDMA_RESET	PDMA_BA + 0x460	R/W	PDMA Channel Reset Control Register	0x0000_0000
PDMA_REQSEL0_3	PDMA_BA + 0x480	R/W	PDMA Channel 0 to Channel 3 Request Source Select Register	0x0000_0000
PDMA_REQSEL4	PDMA_BA + 0x484	R/W	PDMA Channel 4 Request Source Select Register	0x0000_0000

6.12.7 Register Description

PDMA Descriptor Table Control Register (PDMA\_DSCTn\_CTL)

Register	Offset	R/W	Description	Reset Value
PDMA_DSCT0_CTL	PDMA_BA + 0x000	R/W	Descriptor Table Control Register of PDMA Channel 0	0xXXXX_XXXX
PDMA_DSCT1_CTL	PDMA_BA + 0x010	R/W	Descriptor Table Control Register of PDMA Channel 1	0xXXXX_XXXX
PDMA_DSCT2_CTL	PDMA_BA + 0x020	R/W	Descriptor Table Control Register of PDMA Channel 2	0xXXXX_XXXX
PDMA_DSCT3_CTL	PDMA_BA + 0x030	R/W	Descriptor Table Control Register of PDMA Channel 3	0xXXXX_XXXX
PDMA_DSCT4_CTL	PDMA_BA + 0x040	R/W	Descriptor Table Control Register of PDMA Channel 4	0xXXXX_XXXX

31	30	29	28	27	26	25	24
Reserved		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
[31:30]	<b>Reserved</b> Reserved.
[29:16]	<b>TXCNT</b> <b>Transfer Count</b> The TXCNT represents the required number of PDMA transfer, the real transfer count is (TXCNT + 1); The maximum transfer count is 16384 , every transfer may be byte, half-word or word that is dependent on TXWIDTH field. <b>Note:</b> When PDMA finish each transfer data, this field will be decrease immediately.
[15:14]	<b>Reserved</b> Reserved.
[13:12]	<b>TXWIDTH</b> <b>Transfer Width Selection</b> 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. <b>Note:</b> The PDMA transfer source address (PDMA_DSCT_SA) and PDMA transfer destination address (PDMA_DSCT_DA) should be alignment under the TXWIDTH selection. For example, if source address is 0x2000_0202, but TXWIDTH is word transfer, the source address is not word alignment. The source address is aligned when TXWIDTH is byte or half-word transfer.

[11:10]	<b>DAINC</b>	<p><b>Destination Address Increment</b></p> <p>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.</p>
[9:8]	<b>SAINC</b>	<p><b>Source Address Increment</b></p> <p>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.</p>
[7]	<b>TBINTDIS</b>	<p><b>Table Interrupt Disable Bit</b></p> <p>This field can be used to decide whether to enable table interrupt or not. If the TBINTDIS bit is enabled when PDMA controller finishes transfer task, it will not generates transfer done interrupt.            0 = Table interrupt Enabled.            1 = Table interrupt Disabled.  <b>Note:</b> If this bit set to '1', the TEMPTYF will not be set.</p>
[6:4]	<b>BURSIZE</b>	<p><b>Burst Size</b></p> <p>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.  <b>Note:</b> This field is only useful in burst transfer type.</p>
[3]	<b>Reserved</b>	Reserved.
[2]	<b>TXTYPE</b>	<p><b>Transfer Type</b></p> <p>0 = Burst transfer type.            1 = Single transfer type.</p>
[1:0]	<b>OPMODE</b>	<p><b>PDMA Operation Mode Selection</b></p> <p>00 = Idle state: Channel is stopped or this table is complete, when PDMA finish 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 TDIF(PDMA_INTSTS[1]) will be asserted.            10 = Scatter-Gather mode: When operating in this mode, user must give the first descriptor table address in PDMA_DSCT_FIRST register; PDMA controller will ignore this task, then load the next task to execute.            11 = Reserved.  <b>Note:</b> Before filling transfer task in the Descriptor Table, user must check if the descriptor table is complete.</p>

**Source Address Register (PDMA\_DSCTn\_SA)**

Register	Offset	R/W	Description	Reset Value
PDMA_DSCT0_SA	PDMA_BA + 0x004	R/W	Source Address Register of PDMA Channel 0	0xXXXX_XXXX
PDMA_DSCT1_SA	PDMA_BA + 0x014	R/W	Source Address Register of PDMA Channel 1	0xXXXX_XXXX
PDMA_DSCT2_SA	PDMA_BA + 0x024	R/W	Source Address Register of PDMA Channel 2	0xXXXX_XXXX
PDMA_DSCT3_SA	PDMA_BA + 0x034	R/W	Source Address Register of PDMA Channel 3	0xXXXX_XXXX
PDMA_DSCT4_SA	PDMA_BA + 0x044	R/W	Source Address Register of PDMA Channel 4	0xXXXX_XXXX

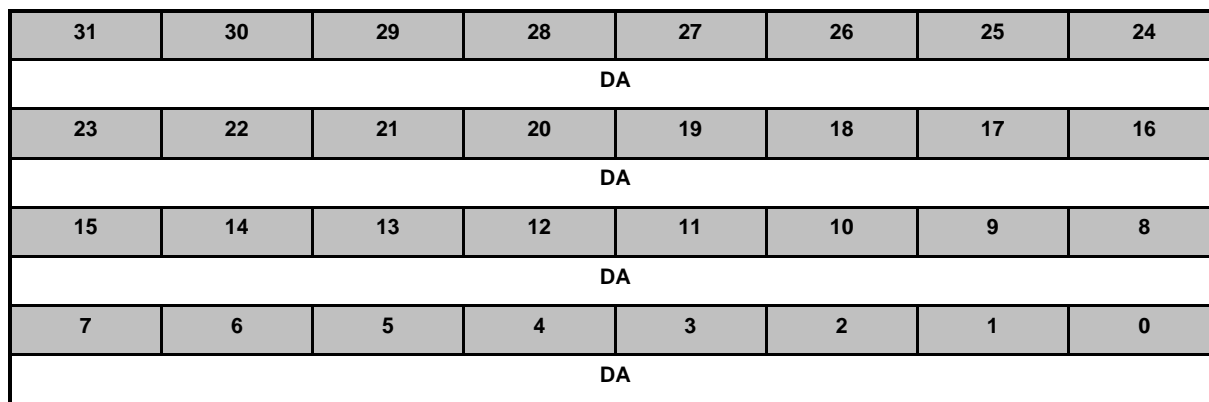
31	30	29	28	27	26	25	24
SA							
23	22	21	20	19	18	17	16
SA							
15	14	13	12	11	10	9	8
SA							
7	6	5	4	3	2	1	0
SA							

Bits	Description	
[31:0]	SA	<p><b>PDMA Transfer Source Address Register</b></p> <p>This field indicates a 32-bit source address of PDMA controller.</p> <p><b>Note:</b> The PDMA transfer source address should be aligned with the TXWIDTH(PDMA_DSCTn_CTL[13:12], n=0,1..4) selection.</p>



**Destination Address Register (PDMA\_DSCTn\_DA)**

Register	Offset	R/W	Description	Reset Value
PDMA_DSCT0_DA	PDMA_BA + 0x008	R/W	Destination Address Register of PDMA Channel 0	0xFFFF_FFFF
PDMA_DSCT1_DA	PDMA_BA + 0x018	R/W	Destination Address Register of PDMA Channel 1	0xFFFF_FFFF
PDMA_DSCT2_DA	PDMA_BA + 0x028	R/W	Destination Address Register of PDMA Channel 2	0xFFFF_FFFF
PDMA_DSCT3_DA	PDMA_BA + 0x038	R/W	Destination Address Register of PDMA Channel 3	0xFFFF_FFFF
PDMA_DSCT4_DA	PDMA_BA + 0x048	R/W	Destination Address Register of PDMA Channel 4	0xFFFF_FFFF



Bits	Description	
[31:0]	DA	<p><b>PDMA Transfer Destination Address Register</b></p> <p>This field indicates a 32-bit destination address of PDMA controller.</p> <p><b>Note:</b> The PDMA transfer destination address should be aligned with the TXWIDTH(PDMA_DSCTn_CTL[13:12], n=0,1..4) selection.</p>

**First Scatter-Gather Descriptor Table Offset (PDMA\_DSCTn\_FIRST)**

Register	Offset	R/W	Description	Reset Value
PDMA_DSCT0_FIRST	PDMA_BA + 0x00C	R/W	First Scatter-Gather Descriptor Table Offset of PDMA Channel 0	0xXXXX_XXXX
PDMA_DSCT1_FIRST	PDMA_BA + 0x01C	R/W	First Scatter-Gather Descriptor Table Offset of PDMA Channel 1	0xXXXX_XXXX
PDMA_DSCT2_FIRST	PDMA_BA + 0x02C	R/W	First Scatter-Gather Descriptor Table Offset of PDMA Channel 2	0xXXXX_XXXX
PDMA_DSCT3_FIRST	PDMA_BA + 0x03C	R/W	First Scatter-Gather Descriptor Table Offset of PDMA Channel 3	0xXXXX_XXXX
PDMA_DSCT4_FIRST	PDMA_BA + 0x04C	R/W	First Scatter-Gather Descriptor Table Offset of PDMA Channel 4	0xXXXX_XXXX

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

Bits	Description	
[31:16]	NEXT	<p><b>PDMA Next Descriptor Table Offset</b></p> <p>This field indicates the offset of next descriptor table address in system memory.</p> <p><b>Note:</b> write operation is useless in this field.</p>
[15:0]	FIRST	<p><b>PDMA First Descriptor Table Offset</b></p> <p>This field indicates the offset of the first descriptor table address in system memory.</p> <p><b>Write Operation:</b></p> <p>If the system memory based address is 0x2000_0000 (PDMA_SCATBA), and the first descriptor table is start from 0x2000_0100, then this field must fill in 0x0100.</p> <p><b>Read Operation:</b></p> <p>When operating in scatter-gather mode, the last two bits FIRST[1:0] will become reserved.</p> <p><b>Note1:</b> The first descriptor table address must be word boundary.</p> <p><b>Note2:</b> Before filled transfer task in the descriptor table, user must check if the descriptor table is complete.</p>

**Current Scatter-Gather Descriptor Table Address (PDMA\_CURSCATn)**

Register	Offset	R/W	Description	Reset Value
PDMA_CURSCAT0	PDMA_BA + 0x050	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 0	0XXXXX_XXXX
PDMA_CURSCAT1	PDMA_BA + 0x054	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 1	0XXXXX_XXXX
PDMA_CURSCAT2	PDMA_BA + 0x058	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 2	0XXXXX_XXXX
PDMA_CURSCAT3	PDMA_BA + 0x05C	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 3	0XXXXX_XXXX
PDMA_CURSCAT4	PDMA_BA + 0x060	R	Current Scatter-Gather Descriptor Table Address of PDMA Channel 4	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]	<p><b>CURADDR</b></p> <p><b>PDMA Current Description Address Register (Read Only)</b></p> <p>This field indicates a 32-bit current external description address of PDMA controller.</p> <p><b>Note:</b> This field is read only and only used for Scatter-Gather mode to indicate the current external description address.</p>

**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]	<b>Reserved</b> Reserved.
[n] n=0,1..4	<p><b>PDMA Channel N Enable Bit</b></p> <p>Set this bit to 1 to enable PDMA<sub>n</sub> operation. Channel cannot be active if it is not set as enabled.</p> <p>0 = PDMA channel [n] Disabled. 1 = PDMA channel [n] Enabled.</p> <p><b>Note:</b> Set PDMA_PAUSE or PDMA_RESET register will also clear this bit.</p>

**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	<p><b>PDMA Channel N Transfer Pause Control Register (Write Only)</b></p> <p>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. If re-enable the paused channel again, the remaining transfers will be processed.</p> <p>0 = No effect. 1 = Pause PDMA channel n transfer.</p>

**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	<p><b>PDMA Channel N Software Request Register (Write Only)</b> Set this bit to 1 to generate a software request to PDMA [n]. 0 = No effect. 1 = Generate a software request.</p> <p><b>Note1:</b> User can read PDMA_TRGSTS register to know which channel is on active. Active flag may be triggered by software request or peripheral request.</p> <p><b>Note2:</b> If user does not enable corresponding PDMA channel, the software request will be ignored.</p>

**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 <sub>n</sub>	<p><b>PDMA Channel N Request Status (Read Only)</b></p> <p>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.</p> <p>0 = PDMA Channel n has no request. 1 = PDMA Channel n has a request.</p> <p><b>Note:</b> If user pauses or resets each PDMA transfer by setting PDMA_PAUSE or PDMA_RESET register respectively, this bit will be cleared automatically after finishing current transfer.</p>

**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	<p><b>PDMA Channel N Fixed Priority Setting Register</b></p> <p>Set this bit to 1 to enable fixed priority level. The fixed priority channel has higher priority than round-robin priority channel. If multiple channels are set as the same priority, the higher number of channels have higher priority.</p> <p><b>Write Operation:</b> 0 = No effect. 1 = Set PDMA channel [n] to fixed priority channel.</p> <p><b>Read Operation:</b> 0 = Corresponding PDMA channel is round-robin priority. 1 = Corresponding PDMA channel is fixed priority.</p> <p><b>Note:</b> This field only set to fixed priority, clear fixed priority use PDMA_PRICLR register.</p>



**PDMA Fixed 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	<p><b>PDMA Channel N Fixed Priority Clear Register (Write Only)</b>                      Set this bit to 1 to clear fixed priority level.                      0 = No effect.                      1 = Clear PDMA channel [n] fixed priority setting.  <b>Note:</b> User can read PDMA_PRISET register to know the channel priority.</p>

**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	
[31:12]	Reserved	Reserved.
[n] n=0,1..4	INTENn	<b>PDMA Channel N Interrupt Enable Register</b> This field is used for enabling PDMA channel[n] interrupt. 0 = PDMA channel n interrupt Disabled. 1 = PDMA channel n interrupt Enabled.

**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					TEIF	TDIF	ABTIF

Bits	Description
[31:13]	<b>Reserved</b> Reserved.
[n+8] n=0,1	<b>REQTOFn</b> <b>PDMA Channel N Request Time-out Flag</b> This flag indicates that PDMA controller has waited peripheral request for a period defined by PDMA_TOCN, user can write 1 to clear these bits. 0 = No request time-out. 1 = Peripheral request time-out.
[7:3]	<b>Reserved</b> Reserved.
[2]	<b>TEIF</b> <b>Table Empty Interrupt Flag (Read Only)</b> This bit indicates PDMA channel scatter-gather table is empty. User can read PDMA_SCATSTS register to indicate which channel scatter-gather table is empty. 0 = PDMA channel scatter-gather table is not empty. 1 = PDMA channel scatter-gather table is empty.
[1]	<b>TDIF</b> <b>Transfer Done Interrupt Flag (Read Only)</b> 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]	<b>ABTIF</b> <b>PDMA Read/Write Target Abort Interrupt Flag (Read Only)</b> 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. 1 = AHB bus ERROR response received.

**PDMA Channel Read/Write Target Abort Flag Register (PDMA\_ABTSTS)**

Register	Offset	R/W	Description	Reset Value
PDMA_ABTSTS	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	<p><b>PDMA Channel N Read/Write Target Abort Interrupt Status Flag</b></p> <p>This bit indicates which PDMA controller has target abort error; User can write 1 to clear these bits.</p> <p>0 = No AHB bus ERROR response received when channel n transfer.</p> <p>1 = AHB bus ERROR response received when channel n transfer.</p>

**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	<p><b>PDMA Channel N Transfer Done Flag Register</b></p> <p>This bit indicates whether PDMA controller channel transfer has been finished or not, user can write 1 to clear these bits.</p> <p>0 = PDMA channel transfer has not finished.</p> <p>1 = PDMA channel has finished transmission.</p>

**PDMA Scatter-Gather Table Empty Status Register (PDMA\_SCATSTS)**

Register	Offset	R/W	Description	Reset Value
PDMA_SCATSTS	PDMA_BA + 0x428	R/W	PDMA Scatter-Gather Table Empty 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			EMPTYF4	EMPTYF3	EMPTYF2	EMPTYF1	EMPTYF0

Bits	Description	
[31:5]	Reserved	Reserved.
[n] n=0,1..4	EMPTYFn	<p><b>Table Empty Flag Register</b></p> <p>T This bit indicates which PDMA channel table is empty when channel have a request , no matter request from software or peripheral, but operation mode of channel descriptor table is idle state, or channel has finished current transfer and next table operation mode is idle state for PDMA Scatter-Gather mode. User can write 1 to clear these bits.</p> <p>0 = PDMA channel scatter-gather table is not empty.</p> <p>1 = PDMA channel scatter-gather table is empty and PDMA SWREQ has be set.</p>

**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	<b>PDMA Channel N Transfer on Active Flag Register (Read Only)</b> 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]	<b>Reserved</b> Reserved.
[6:4]	<b>TOUTPSC1</b> <b>PDMA Channel 1 Time-out Clock Source Prescaler Bits</b> 000 = PDMA channel 1 time-out clock source is HCLK/2 <sup>8</sup> . 001 = PDMA channel 1 time-out clock source is HCLK/2 <sup>9</sup> . 010 = PDMA channel 1 time-out clock source is HCLK/2 <sup>10</sup> . 011 = PDMA channel 1 time-out clock source is HCLK/2 <sup>11</sup> . 100 = PDMA channel 1 time-out clock source is HCLK/2 <sup>12</sup> . 101 = PDMA channel 1 time-out clock source is HCLK/2 <sup>13</sup> . 110 = PDMA channel 1 time-out clock source is HCLK/2 <sup>14</sup> . 111 = PDMA channel 1 time-out clock source is HCLK/2 <sup>15</sup> .
[3]	<b>Reserved</b> Reserved.
[2:0]	<b>TOUTPSC0</b> <b>PDMA Channel 0 Time-out Clock Source Prescaler Bits</b> 000 = PDMA channel 0 time-out clock source is HCLK/2 <sup>8</sup> . 001 = PDMA channel 0 time-out clock source is HCLK/2 <sup>9</sup> . 010 = PDMA channel 0 time-out clock source is HCLK/2 <sup>10</sup> . 011 = PDMA channel 0 time-out clock source is HCLK/2 <sup>11</sup> . 100 = PDMA channel 0 time-out clock source is HCLK/2 <sup>12</sup> . 101 = PDMA channel 0 time-out clock source is HCLK/2 <sup>13</sup> . 110 = PDMA channel 0 time-out clock source is HCLK/2 <sup>14</sup> . 111 = PDMA channel 0 time-out clock source is HCLK/2 <sup>15</sup> .



**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	<b>PDMA Channel N Time-out Enable Bit</b> 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	<b>PDMA Channel N Time-out Interrupt Enable Bit</b> 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]	<b>SCATBA</b>	<p><b>PDMA Scatter-gather Descriptor Table Address Register</b></p> <p>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_FIRST.</p> <p><b>Note:</b> Only useful in Scatter-Gather mode.</p>
[15:0]	<b>Reserved</b>	Reserved.

**PDMA Channel 0 and Channel 1 Time-out Counter Register (PDMA\_TOC0\_1)**

Register	Offset	R/W	Description	Reset Value
PDMA_TOC0_1	PDMA_BA + 0x440	R/W	PDMA Channel 0 and Channel 1 Time-out Counter Register	0x0000_0000

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]	<b>TOC1</b>	<p><b>Time-out Counter for Channel 1</b></p> <p>This controls the period of time-out function for channel 1. The calculation unit is based on TOUTPSC1 (PDMA_TOUTPSC[5:3]) clock. The example of time-out period can refer TOC0 bit description.</p>
[15:0]	<b>TOC0</b>	<p><b>Time-out Counter for Channel 0</b></p> <p>This controls the period of time-out function for channel 0. The calculation unit is based on TOUTPSC0 (PDMA_TOUTPSC[2:0]) clock.</p> <p>Time-out period = (Period of time-out clock) * (16-bit TOCn), n = 0,1.</p>

**PDMA Channel Reset Control Register (PDMA\_RESET)**

Register	Offset	R/W	Description	Reset Value
PDMA_RESET	PDMA_BA + 0x460	R/W	PDMA Channel Reset 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			RESET4	RESET3	RESET2	RESET1	RESET0

Bits	Description	
[31:5]	Reserved	Reserved.
[n] n=0,1..4	RESETn	<p><b>PDMA Channel N Reset Control Register</b></p> <p>User can set this bit field to reset the PDMA channel. When user sets RESETn bit, the PDMA controller will finish the on-going transfer then clear the channel enable bit CHEN(PDMA_CHCTL [n], n=0,1..4) and clear request active flag. If re-enable channel after channel reset, PDMA will re-load the channel description table to execute PDMA task.</p> <p>0 = No effect. 1 = Reset PDMA channel n.</p> <p><b>Note:</b> This bit will be cleared automatically after finishing reset process.</p>

**PDMA Channel 0 to Channel 3 Request Source Select Register (PDMA\_REQSEL0\_3)**

Register	Offset	R/W	Description	Reset Value
PDMA_REQSEL0_3	PDMA_BA + 0x480	R/W	PDMA Channel 0 to Channel 3 Request Source Select Register	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	
[31:30]	Reserved	Reserved.
[29:24]	REQSRC3	<p><b>Channel 3 Request Source Selection</b></p> <p>This field defines which peripheral is connected to PDMA channel 3. User can configure the peripheral setting by REQSRC3.</p> <p><b>Note:</b> The channel configuration is the same as REQSRC0 field. Please refer to the explanation of REQSRC0.</p>
[23:22]	Reserved	Reserved.
[21:16]	REQSRC2	<p><b>Channel 2 Request Source Selection</b></p> <p>This field defines which peripheral is connected to PDMA channel 2. User can configure the peripheral setting by REQSRC2.</p> <p><b>Note:</b> The channel configuration is the same as REQSRC0 field. Please refer to the explanation of REQSRC0.</p>
[15:14]	Reserved	Reserved.
[13:8]	REQSRC1	<p><b>Channel 1 Request Source Selection</b></p> <p>This field defines which peripheral is connected to PDMA channel 1. User can configure the peripheral setting by REQSRC1.</p> <p><b>Note:</b> The channel configuration is the same as REQSRC0 field. Please refer to the explanation of REQSRC0.</p>
[7:6]	Reserved	Reserved.
[5:0]	REQSRC0	<p><b>Channel 0 Request Source Selection</b></p> <p>This field defines which peripheral is connected to PDMA channel 0. User can configure the peripheral by setting REQSRC0.</p> <p>0 = Disable PDMA.                      1 = Reserved.                      2 = Channel connects to USB_TX.                      3 = Channel connects to USB_RX.                      4 = Channel connects to UART0_TX.</p>

Bits	Description
	<p>5 = Channel connects to UART0_RX.                      6 = Channel connects to UART1_TX.                      7 = Channel connects to UART1_RX.                      8 = Channel connects to UART2_TX.                      9 = Channel connects to UART2_RX.                      16 = Channel connects to SPI0_TX.                      17 = Channel connects to SPI0_RX.                      18 = Channel connects to SPI1_TX.                      19 = Channel connects to SPI1_RX.                      20 = Channel connects to ADC_RX.                      21 = Channel connects to PWM0_P1_RX.                      22 = Channel connects to PWM0_P2_RX.                      23 = Channel connects to PWM0_P3_RX.                      24 = Channel connects to PWM1_P1_RX.                      25 = Channel connects to PWM1_P2_RX.                      26 = Channel connects to PWM1_P3_RX.                      27 = Reserved.                      28 = Channel connects to I2C0_TX.                      29 = Channel connects to I2C0_RX.                      30 = Channel connects to I2C1_TX.                      31 = Channel connects to I2C1_RX.                      32 = Channel connects to TMR0.                      33 = Channel connects to TMR1.                      34 = Channel connects to TMR2.                      35 = Channel connects to TMR3.                      Others = Reserved.</p> <p><b>Note1:</b> A request source can't assign to two channels at the same time.  <b>Note2:</b> This field is useless when transfer between memory and memory.</p>

**PDMA Channel 4 Request Source Select Register (PDMA\_REQSEL4)**

Register	Offset	R/W	Description	Reset Value
PDMA_REQSEL4	PDMA_BA + 0x484	R/W	PDMA Channel 4 Request Source 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		REQSRC4					

Bits	Description	
[31:6]	Reserved	Reserved.
[5:0]	REQSRC4	<p><b>Channel 4 Request Source Selection</b></p> <p>This field defines which peripheral is connected to PDMA channel 4. User can configure the peripheral setting by REQSRC4.</p> <p><b>Note:</b> The channel configuration is the same as REQSRC0 field. Please refer to the explanation of REQSRC0.</p>



## 6.13 PWM Generator and Capture Timer (PWM)

### 6.13.1 Overview

The NUC126 provides two PWM generator: PWM0 and PWM1 as Figure 6.13-1. Each PWM supports 6 channels of PWM output or input capture. There is a 12-bit prescaler to support flexible clock to the 16-bit PWM counter with 16-bit comparator. The PWM counter supports up, down and up-down counter types. PWM uses comparator compared with counter to generate events. These events use to generate PWM pulse, interrupt and trigger signal for ADC to start conversion.

The PWM generator supports two standard PWM output modes: Independent mode and Complementary mode, they have difference architecture. There are two output functions based on standard output modes: Group function and Synchronous function. Group function can be enabled under Independent mode or complementary mode. Synchronous function only enabled under complementary mode. Complementary mode has two comparators to generate various PWM pulse with 12-bit dead-time generator and another free trigger comparator to generate trigger signal for ADC. For PWM output control unit, it supports polarity output, independent pin mask and brake functions.

The PWM generator also supports input capture function. It supports latch PWM counter value to corresponding register when input channel has a rising transition, falling transition or both transition is happened. Capture function also support PDMA to transfer captured data to memory.

### 6.13.2 Features

#### 6.13.2.1 PWM function features

- Supports maximum clock frequency up to 144MHz
- Supports up to two PWM modules, each module provides 6 output channels.
- Supports independent mode for PWM output/Capture input channel
- Supports complementary mode for 3 complementary paired PWM output channels:
  - Dead-time insertion with 12-bit resolution
  - Synchronous function for phase control
  - Two compared values during one period
- Supports 12-bit pre-scalar from 1 to 4096
- Supports 16-bit resolution PWM counter
  - Up, down and up-down counter operation type
- Supports one-shot or auto-reload counter operation mode
- Supports group function
- Supports synchronous function
- Supports mask function and tri-state enable for each PWM output pin
- Supports brake function
  - Brake source from pin, analog comparator, ADC result monitor and system safety events (clock failed, Brown-out detection and CPU lockup).
  - Noise filter for brake source from pin
  - Leading edge blanking (LEB) function for brake source from analog comparator
  - Edge detect brake source to control brake state until brake interrupt cleared
  - Level detect brake source to auto recover function after brake condition removed

- Supports interrupt on the following events:
  - PWM zero point, period point, up-count compared or down-count compared point events
  - Brake condition happened
- Supports trigger ADC on the following events:
  - PWM zero point, period point, zero or period point, up-count compared point, down-count compared point events
  - PWM up-count free trigger compared point, down-count free trigger compared point events

#### 6.13.2.2 Capture Function Features

- Supports up to 6 capture input channels with 16-bit resolution for each PWM module
- Supports rising or falling capture condition
- Supports input rising/falling capture interrupt
- Supports rising/falling capture with counter reload option
- Supports PDMA transfer function for PWM all channels

6.13.3 Block Diagram

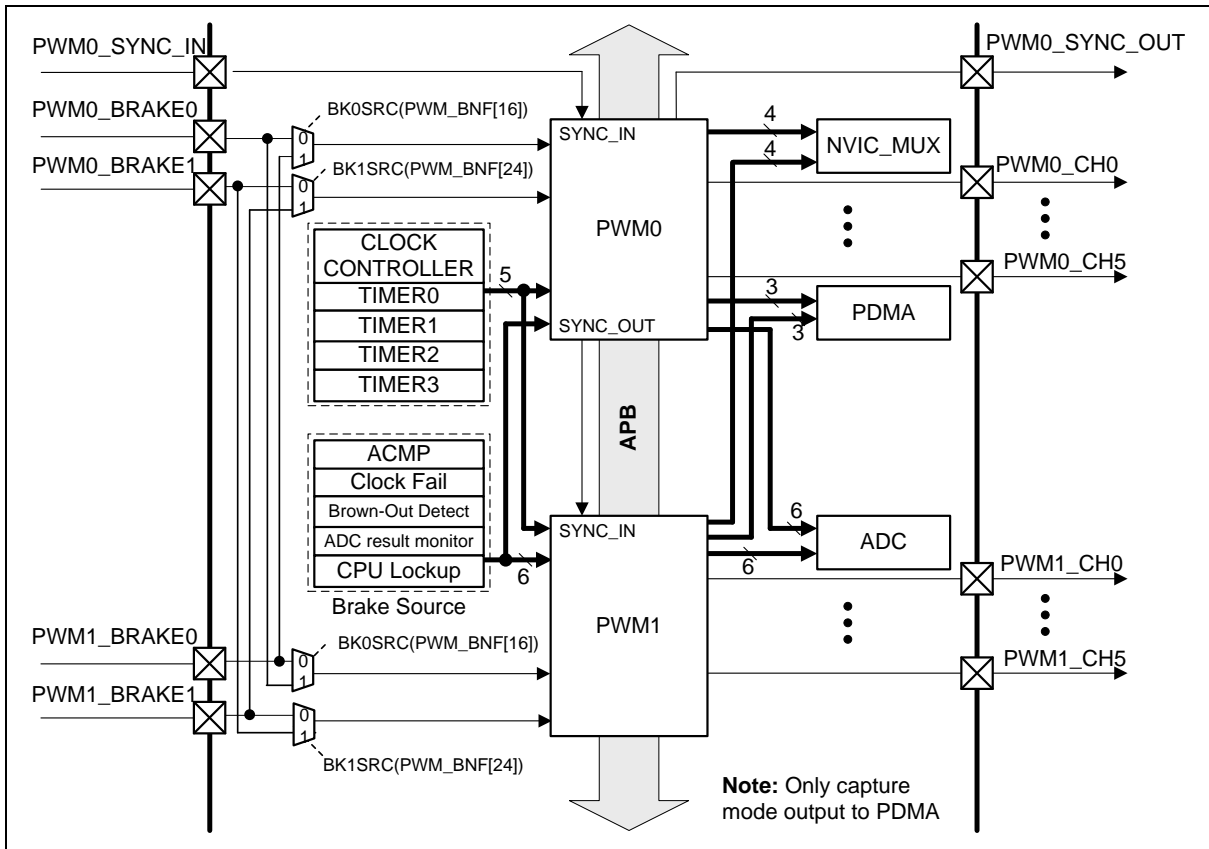


Figure 6.13-1 PWM Generator Overview Block Diagram

The PWM system clock frequency can be set equal or double to HCLK frequency as Figure 6.13-2, the detail register setting, please refer to Table 6.13-1.

The clock source of PWM counter(PWMx\_CLK0, x=0,1) can be selected from PWM0/PWM1 system clock or TIMERm interrupt events(TMRn\_INT, n=0,1..3) as Figure 6.13-3 by setting ECLKSRC0 (PWM\_CLKSRC[2:0]) for PWMx\_CLK0, ECLKSRC2 (PWM\_CLKSRC[10:8]) for PWMx\_CLK2 and ECLKSRC4 (PWM\_CLKSRC[18:16]) for PWMx\_CLK4. If the clock source of PWM counter is selected from TIMERm interrupt events, the TRGPWM(TIMERn\_TRGCTL[1], n=0,1..3) bit must be set as 1.

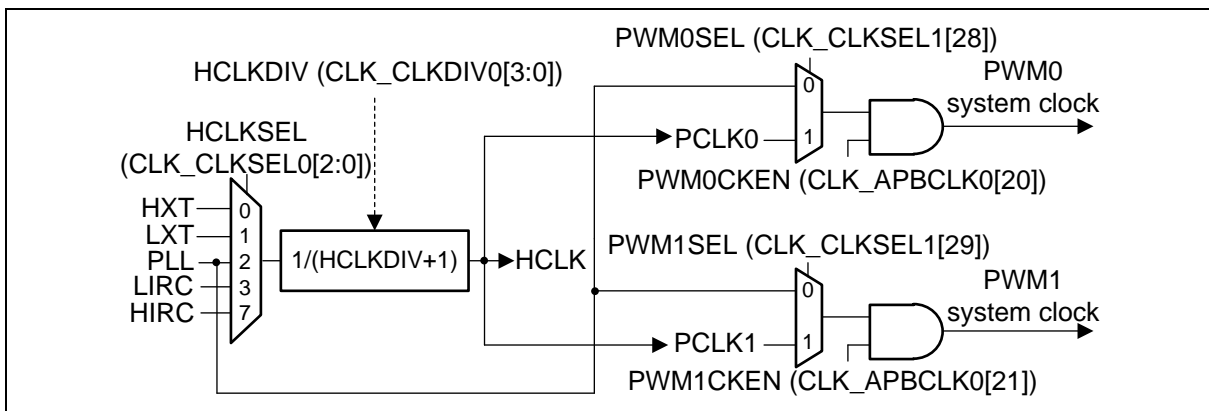


Figure 6.13-2 PWM System Clock Source Control

PWM System Clock/HCLK Frequency Ratio	HCLKSEL (CLK_CLKSEL0[2:0])	HCLKDIV (CLK_CLKDIV0[3:0])	PWM0SEL (CLK_CLKSEL2[0])
1/1	Don't care	Don't care	1
2/1	2	1	0

Table 6.13-1 PWM System Clock Source Control Registers Setting Table

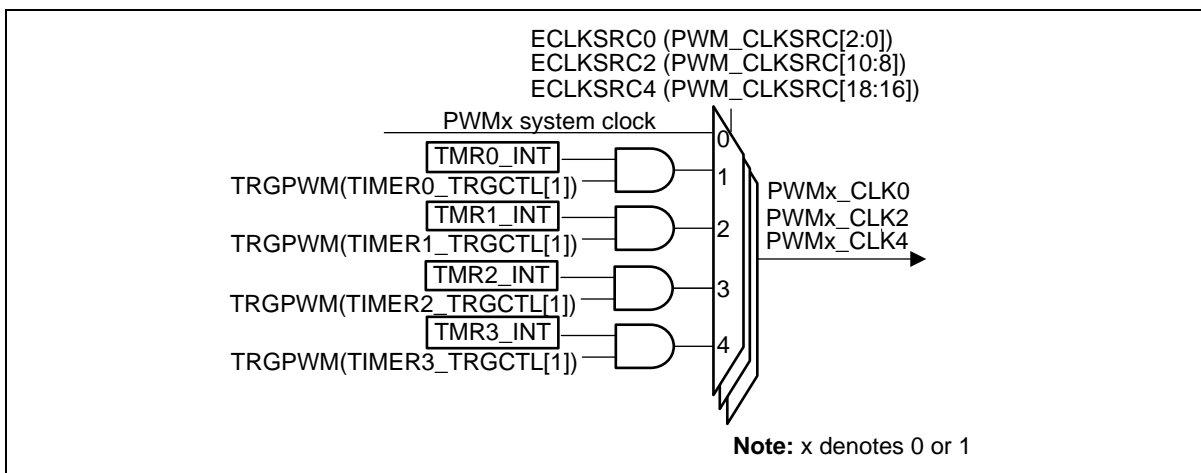


Figure 6.13-3 PWM Clock Source Control

Figure 6.13-4 and Figure 6.13-5 illustrate the architecture of PWM independent mode and complementary mode. No matter independent mode or complementary mode, paired channels' (PWMx\_CH0 and PWMx\_CH1, PWMx\_CH2 and PWMx\_CH3, PWMx\_CH4 and PWMx\_CH5) counters both come from the same clock source and prescaler. When counter count to 0, PERIOD (PWM\_PERIODn[15:0],n=0,1..5) or equal to comparator, events will be generated. These events are passed to corresponding generators to generate PWM pulse, interrupt signal and trigger signal for ADC to start conversion. Output control is used to change PWM pulse output state; brake function in output control also generates interrupt events. In complementary mode, synchronize function is available and even channel use odd channel comparator to generate events, free trigger comparator events only use to generate trigger ADC signals.

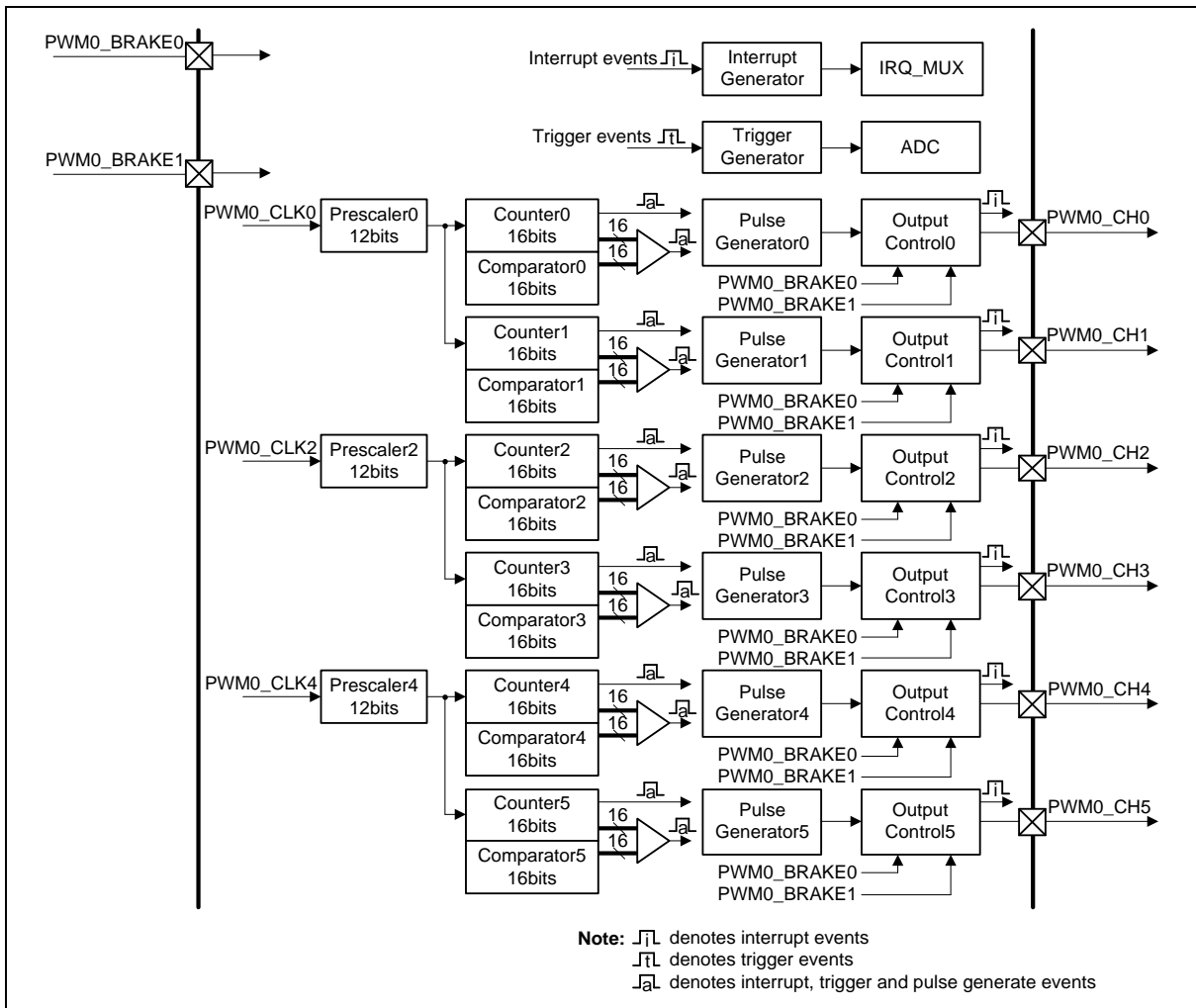


Figure 6.13-4 PWM Independent Mode Architecture Diagram

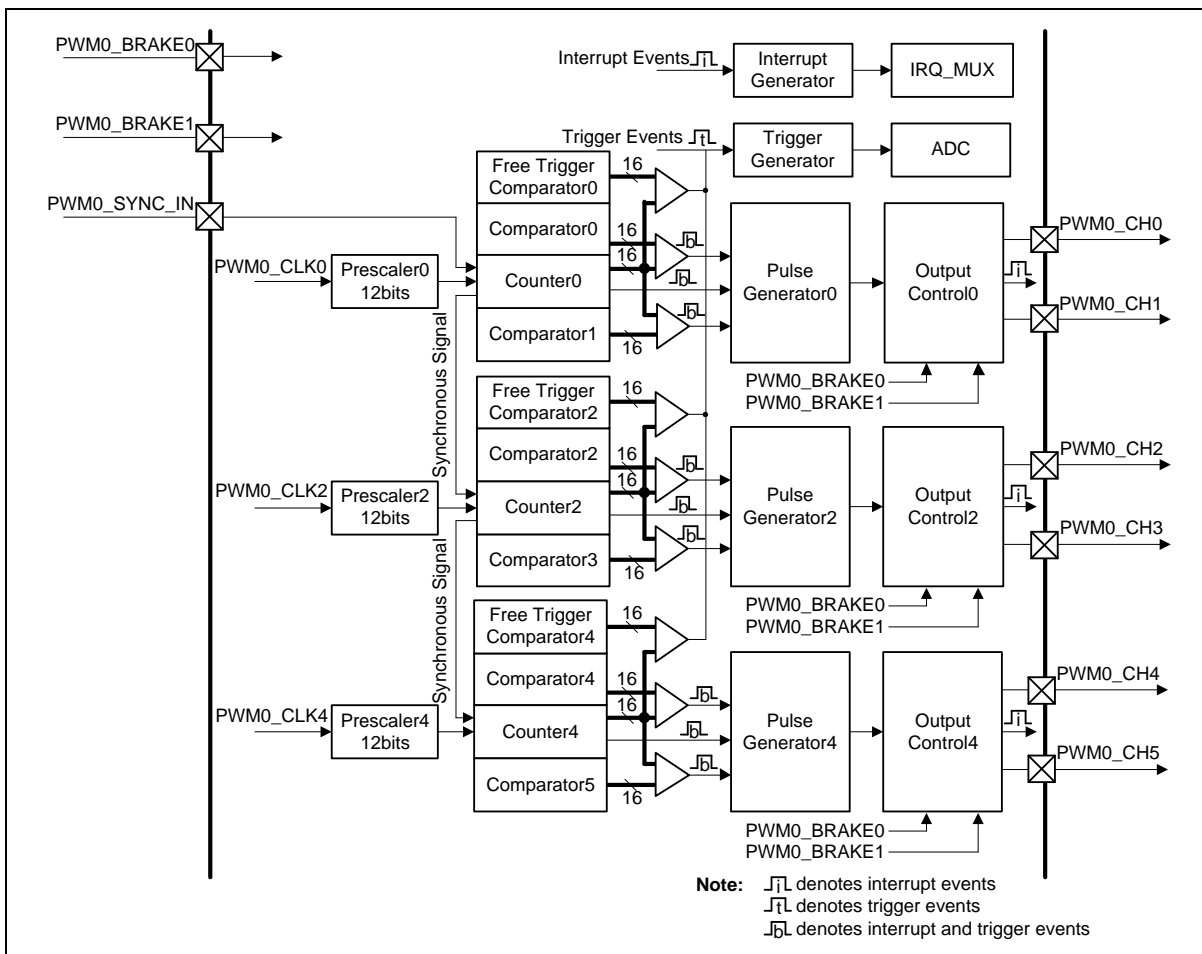


Figure 6.13-5 PWM Complementary Mode Architecture Diagram

### 6.13.4 Basic Configuration

#### 6.13.4.1 Basic Configuration of PWM0

- Clock source Configuration
  - Select the source of PWM0 peripheral clock on PWM0SEL (CLK\_CLKSEL1[28]).
  - Enable PWM0 peripheral clock in PWM0CKEN(CLK\_APBCLK0[20]).
- Reset Configuration
  - Reset PWM0 peripheral in PWM0RST (SYS\_IPRST1[20]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
PWM0	PWM0_BRAKE0	PD.4	MFP5
		PD.2	MFP6
		PA.8	MFP7
	PWM0_BRAKE1	PD.5	MFP5
		PD.3	MFP6

	PWM0_CH0	PC.0, PE.0	MFP6
	PWM0_CH1	PC.1, PE.1	MFP6
	PWM0_CH2	PB.8, PC.2, PE.2	MFP6
	PWM0_CH3	PC.3, PE.3	MFP6
	PWM0_CH4	PC.4	MFP6
	PWM0_CH5	PC.5, PD.6, PD.7	MFP6
	PWM0_SYNC_IN	PD.1	MFP2
		PD.7	MFP3
PWM0_SYNC_OUT	PB.1	MFP6	

6.13.4.2 Basic Configuration of PWM1

- Clock source Configuration
  - Select the source of PWM1 peripheral clock on PWM1SEL (CLK\_CLKSEL1[29]).
  - Enable PWM1 peripheral clock in PWM1CKEN(CLK\_APBCLK0[21]).
- Reset Configuration
  - Reset PWM1 peripheral in PWM1RST (SYS\_IPRST1[21]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
PWM1	PWM1_BRAKE0	PE.4, PF.1	MFP6
	PWM1_BRAKE1	PE.5, PF.2	MFP6
		PA.9	MFP7
	PWM1_CH0	PC.6, PC.9, PC.15, PD.12	MFP6
	PWM1_CH1	PB.12, PC.7, PC.10, PD.13	MFP6
	PWM1_CH2	PA.3, PC.11, PD.14	MFP6
	PWM1_CH3	PA.2, PC.12, PD.15	MFP6
	PWM1_CH4	PA.1, PC.13	MFP6
PWM1_CH5	PA.0, PC.14	MFP6	

6.13.5 Functional Description

6.13.5.1 PWM Prescaler

PWM prescaler is used to divide clock source, prescaler counting CLKPSC +1 times, PWM counter only count once. The prescale is setting by CLKPSC (PWM\_CLKPSCn[11:0], n = 0, 2, 4) bits. Figure 6.13-6 is an example of PWM channel 0 prescale waveform. CLKPSC has the same loading timing as PERIOD, CMPDAT and FTCMPDAT registers.

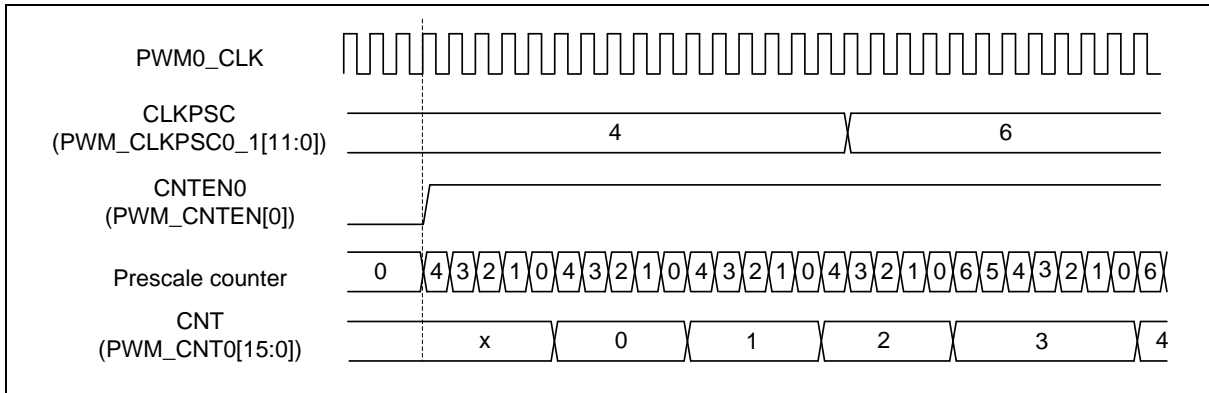


Figure 6.13-6 PWM0\_CH0 Prescaler Waveform in Up Counter Type

6.13.5.2 PWM Counter

PWM supports 3 counter types operation: Up Counter, Down Counter and Up-Down Counter types.

6.13.5.3 Up Counter Type

When PWM counter is set to up counter type, CNTTYPE<sub>n</sub> (PWM\_CTL1[2n+1:2n], n = 0,1..5) is 0x0, it starts up-counting from zero to PERIOD (PWM\_PERIOD<sub>n</sub>[15:0], where n denotes channel number) to complete a PWM period. The current counter value can be read from CNT (PWM\_CNT<sub>n</sub>[15:0]) bits. PWM generates zero point event when the counter counts to 0 and prescale counts to 0. PWM generates period point event when the counter counts to PERIOD and prescale counts to 0. Figure 6.13-7 shows an example of up counter, wherein PWM period time = (PERIOD+1) \* (CLKPSC+1) \* PWM<sub>x</sub>\_CLK.

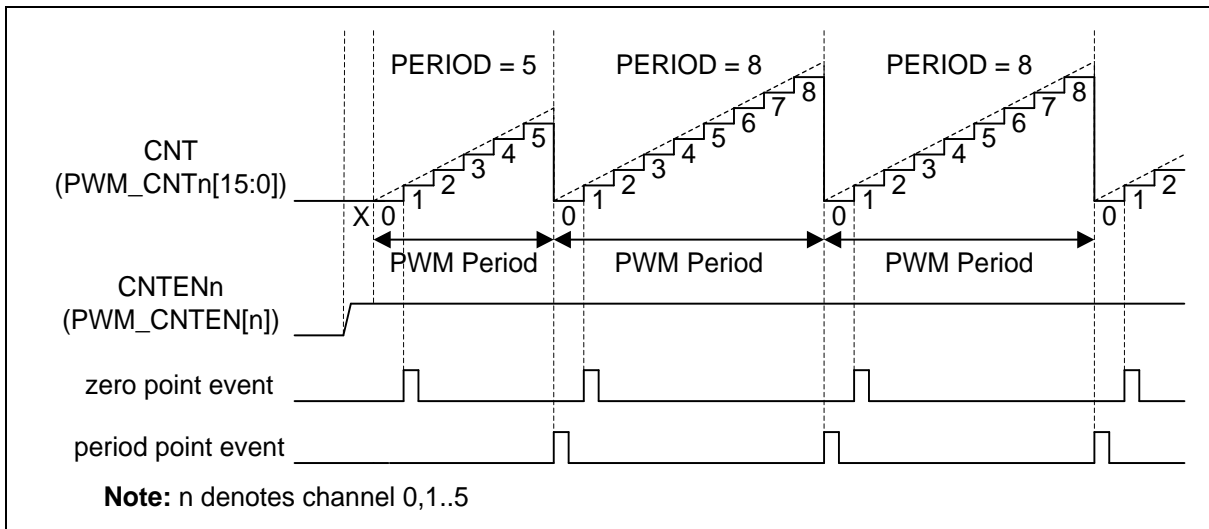


Figure 6.13-7 PWM Up Counter Type

6.13.5.4 Down Counter Type

When PWM counter is set to down counter type, CNTTYPE<sub>n</sub> (PWM\_CTL1[2n+1:2n], n = 0,1..5) is 0x1, it starts down-counting from PERIOD to zero to complete a PWM period. The current counter value can be read from CNT (PWM\_CNT<sub>n</sub>[15:0]) bits. PWM generates zero point event when the counter counts to 0 and prescale counts to 0. PWM generates period point event when the counter counts to PERIOD and prescale counts to 0. Figure 6.13-8 shows an example of down counter, wherein PWM period time = (PERIOD+1) \* (CLKPSC+1) \* PWM<sub>x</sub>\_CLK.



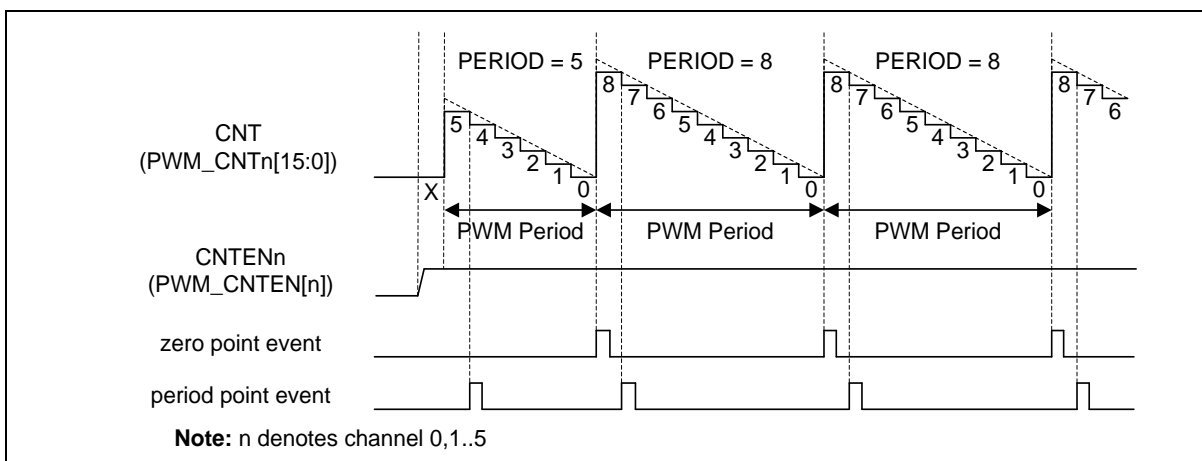


Figure 6.13-8 PWM Down Counter Type

6.13.5.5 Up-Down Counter Type

When PWM counter is set to up-down count type, CNTTYPE<sub>n</sub> (PWM\_CTL1[2n+1:2n], n = 0,1..5) is 0x2, it starts counting-up from zero to PERIOD and then starts counting down to zero to complete a PWM period. The current counter value can be read from CNT (PWM\_CNTn[15:0]) bits. PWM generates zero point event when the counter counts to 0 and prescale counts to 0. PWM generates center point event when the counter counts to PERIOD. Figure 6.13-9 shows an example of up-down counter, wherein PWM period time = (2 \* PERIOD) \* (CLKPSC+1) \* PWMx\_CLK. The DIRF (PWM\_CNTn[16]) bit is counter direction indicator flag, where high is up counting, and low is down counting.

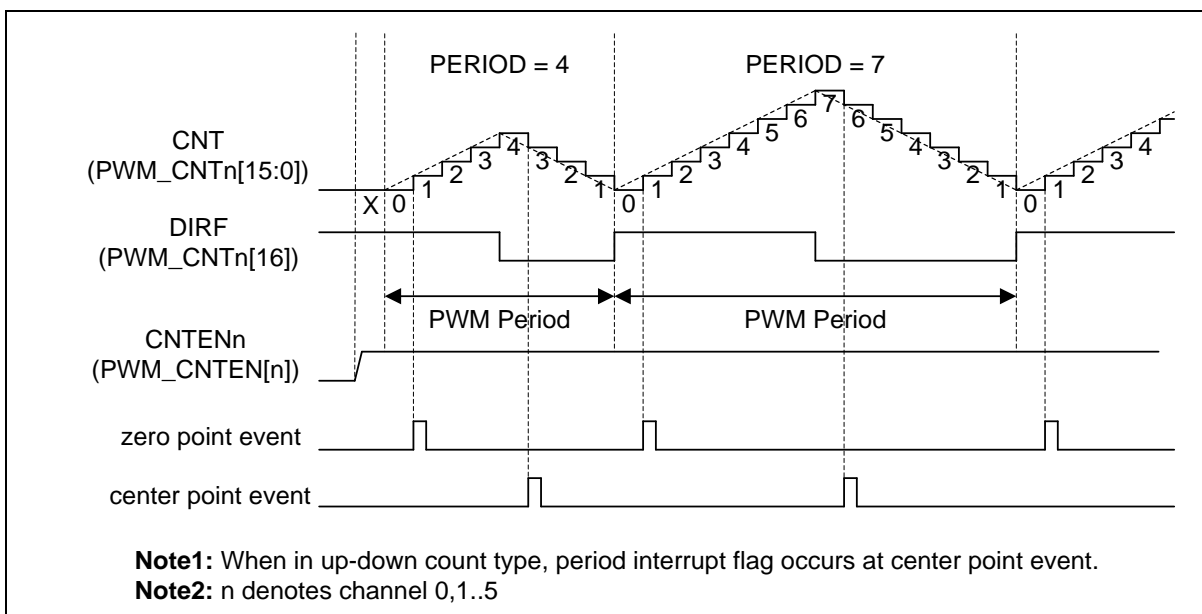


Figure 6.13-9 PWM Up-Down Counter Type

6.13.5.6 PWM Comparator

There are two kinds of comparator registers: one is CMPDAT<sub>n</sub> (n = 0,1..5), and the other is FTCMPDAT<sub>n-m</sub> (n = 0,2,4, m = 1,3,5) register. CMPDAT<sub>n</sub> is a basic comparator register of PWM channel n; In Independent mode each channel only has one comparator, the value of CMPDAT<sub>n</sub> register is continuously compared to the corresponding channel's counter value. In Complementary mode each paired channels has two comparators, and the value of CMPDAT<sub>n</sub> and CMPDAT<sub>m</sub> (n =

0,2,4, m = 1,3,5) registers are continuously compared to the complementary even channel's counter value, because of odd channel's counter is useless. For example, channel 0 and channel 1 are complementary channels, in Complementary mode, channel 1's comparator is continuously compared to channel 0's counter, but not channel 1's. When the counter is equal to value of CMPDAT0 register, PWM generates a compared point event and uses the event to generate PWM pulse, interrupt or use to trigger ADC. In up-down counter type, two events will be generated in a PWM period as shown in Figure 6.13-10. The CMPU is up count compared point event and CMPD is down count compared point event.

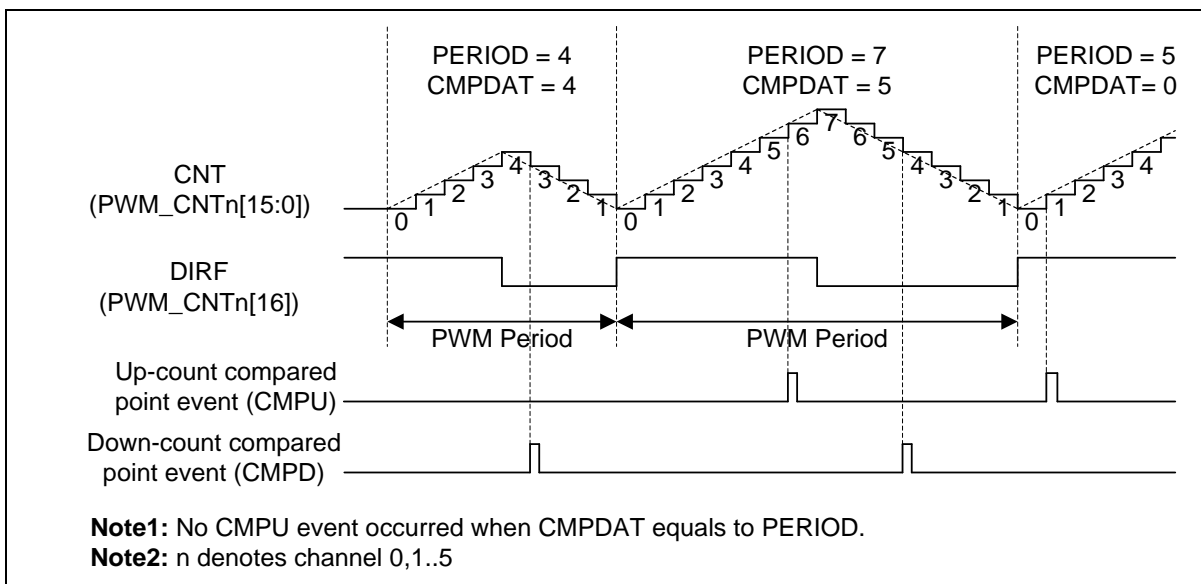


Figure 6.13-10 PWM Compared point Events in Up-Down Counter Type

FTCMPDAT is a free trigger comparator register. Each complementary paired channel only supports one free trigger comparator. The value of FTCMPDATn\_m (n = 0,2,4, m = 1,3,5) register is continuously compared to even channel's counter value. When counter is equal to value of FTCMPDAT register, PWM generates an event and only uses to trigger ADC.

### 6.13.5.7 PWM Double Buffering

The double buffering uses double buffers to separate software writing and hardware action operation timing. There are four loading modes for loading values to buffer: period loading mode, immediately loading mode, window loading mode and center loading mode. After registers are modified through software, hardware will load register value to the buffer register according to the loading mode timing. The hardware action is based on the buffer value. This can prevent asynchronously operation problem due to software and hardware asynchronism.

The PWM provides PBUF (PWM\_PBUFn[15:0]) as the active PERIOD buffer register, CMPBUF (PWM\_CMPBUFn[15:0]) as the active CMPDAT buffer register, FTCMPBUF (PWM\_FTCMPBUFn\_m[15:0]) as the active FTCMPDAT buffer register and CPSCBUF (PWM\_CPSCBUFn\_m[15:0]) as the active CLKPSC buffer register. The concept of double buffering is used in loading modes, which are described in the following sections. For example, as shown in Figure 6.13-11, in period loading mode, writing PERIOD, CMPDAT and FTCMPDAT registers through software, PWM will load new values to their buffer PBUF (PWM\_PBUFn[15:0]), CMPBUF (PWM\_CMPBUFn[15:0]) and FTCMPBUF (PWM\_FTCMPBUFn[15:0]) at start of the next period without affecting the current period counter operation. FTCMPU denotes up-count free trigger compared point event and FTCMPD denotes down-count free trigger compared event.

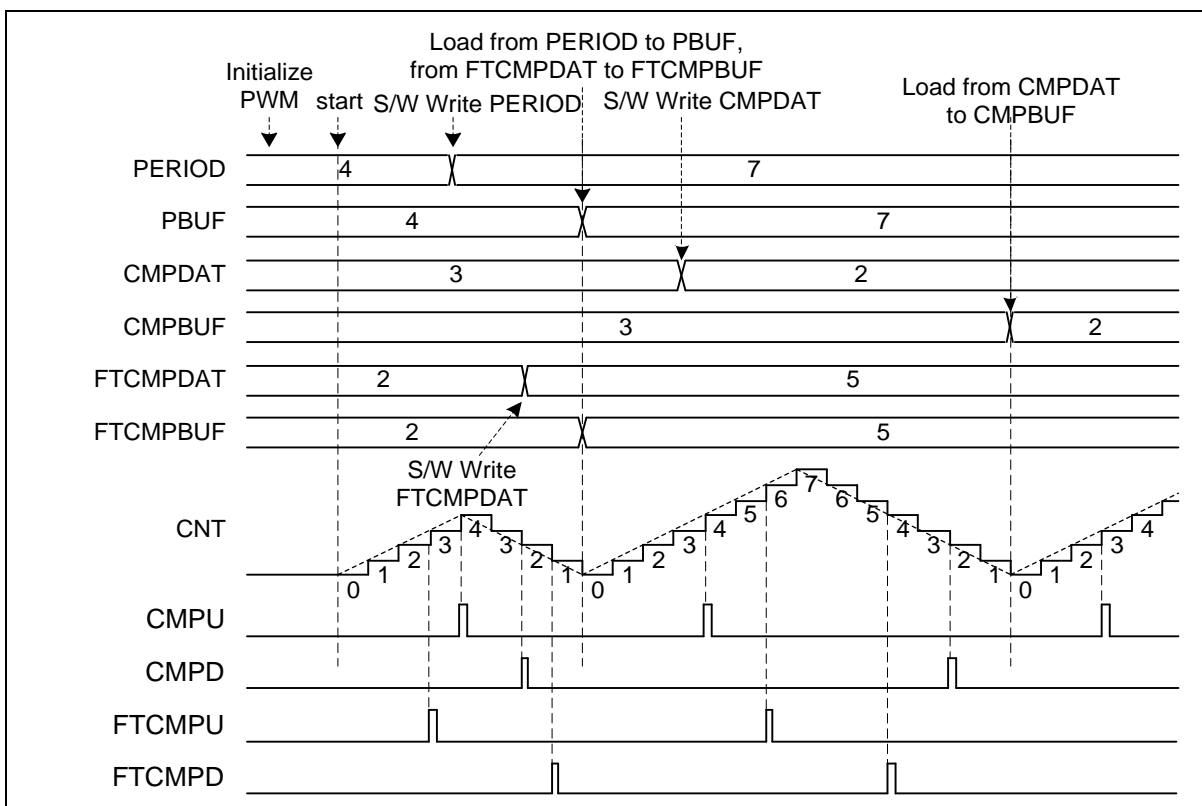


Figure 6.13-11 PWM Double Buffering Illustration

### 6.13.5.8 Period Loading Mode

When immediately loading mode, window loading mode and center loading mode are disabled that IMMLDENn bits, WINLDENn bits and CTRLDN bits of PWM\_CTL0 register are set to 0, PWM operates at period Loading mode. In period Loading mode, CLKPSC(PWM\_CLKPSCn\_m[11:0]), PERIOD(PWM\_PERIODn[15:0]) and CMP(PWM\_CMPDATn[15:0]) will all load to their active CPSCBUF, PBUF and CMPBUF registers while each period is completed. For example, after PWM counter up counts from zero to PERIOD in the up-counter operation or down counts from PERIOD to zero in the down-counter operation or up counts from zero to PERIOD and then down counts to zero in the up-down counter operation.

Figure 6.13-12 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 6.13-12. User can know the PERIOD and CMPDAT update condition, by watching PWM 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.

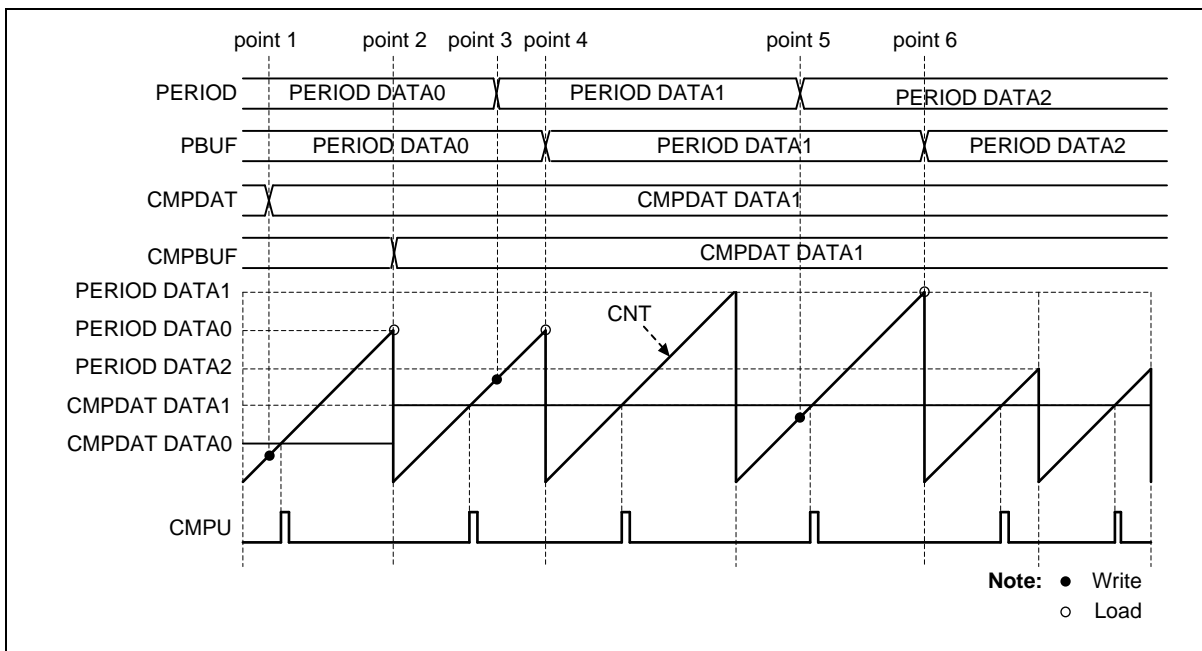


Figure 6.13-12 Period Loading in Up-Count Mode

### 6.13.5.9 Immediately Loading Mode

If the IMMLDENn (PWM\_CTL0[21:16]) bit is set to 1, PWM operates at immediately loading mode. In immediately loading mode, when user update CLKPSC(PWM\_CLKPSCn\_m[11:0]), PERIOD(PWM\_PERIODn[15:0]) or CMP(PWM\_CMPDATn[15:0]), PERIOD or CMPDAT will be load to active CPSCBUF (PWM\_CPSCBUFn\_m[15:0]), PBUF (PWM\_PBUFn[15:0]) or CMPBUF (PWM\_CMPBUFn[15:0]) after current counter count is completed. If the updated PERIOD value is less than current counter value, 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 6.13-13 shows an example and its steps sequence is described below.

1. Software writes CMPDAT DATA1 and hardware immediately loading CMPDAT DATA1 to CMPBUF at point 1.
2. Software writes PERIOD DATA1 which is greater than current counter value at point 2; counter will continue counting until 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.

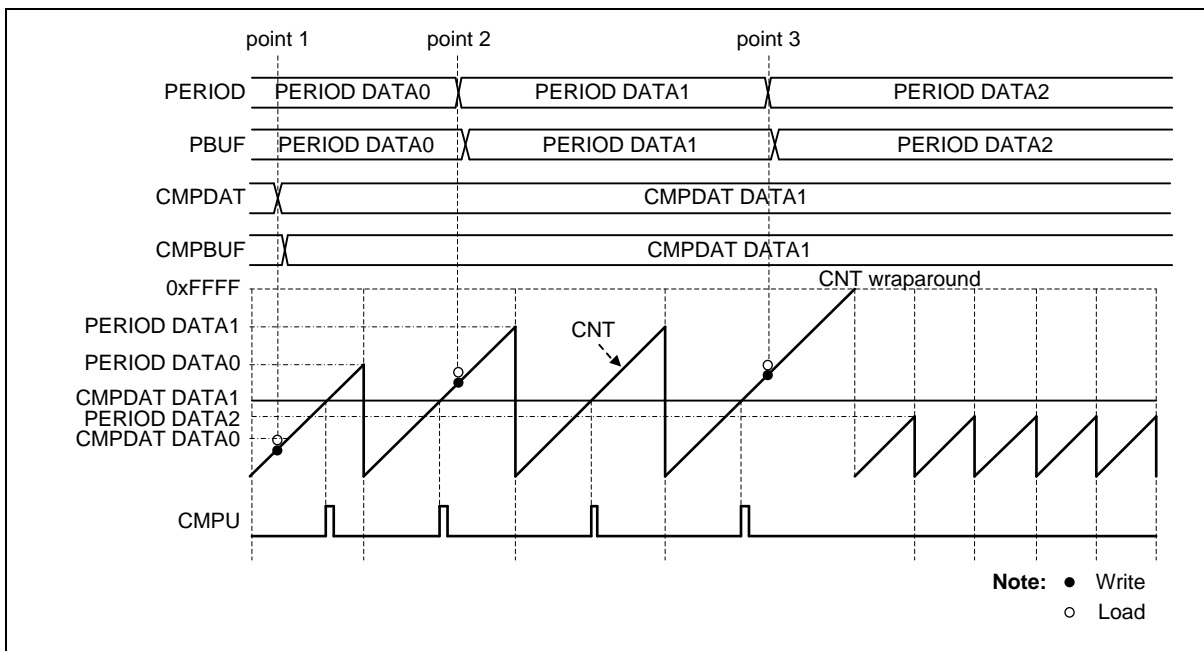


Figure 6.13-13 Immediately Loading in Up-Count Mode

#### 6.13.5.10 Window Loading Mode

When the WINLDENn (PWM\_CTL0[13:8]) bit is set to 1, PWM operates at window loading mode. In Window loading mode, CLKPSC(PWM\_CLKPSCn\_m[11:0]), PERIOD(PWM\_PERIODn[15:0]) and CMP(PWM\_CMPDATn[15:0]) will all load to their active CPSCBUF, PBUF and CMPBUF registers while each period is completed, but all loading are valid only when load window is opened. Every channel n's load window is opened by setting the corresponding LOADn (PWM\_LOAD[5:0]) to 1, and hardware will close the window at the end of PWM period. Figure 6.13-14 shows an example and its steps sequence is described below.

1. Software writes CMPDAT DATA1 at point 1, and the load window is not opened at this period so CMPDAT will not load to CMPBUF.
2. Zero point accumulate interrupt occurs to open the load window at point2.
3. Software writes PERIOD DATA1 at point 3.
4. At point 4, load window has been opened, hardware loads PERIOD DATA1 and CMPDAT DATA1 to their buffer and closes the load window at the end of PWM period.
5. Software writes PERIOD DATA2 at point 5.
6. Hardware loads PERIOD DATA2 to PBUF at the end of PWM period at point 6.
7. Software writes PERIOD DATA3 at point 7.
8. Software writes LOAD to open the load window at point8.
9. Hardware loads PERIOD DATA3 to PBUF and closes the load window at the end of PWM period at point 9.

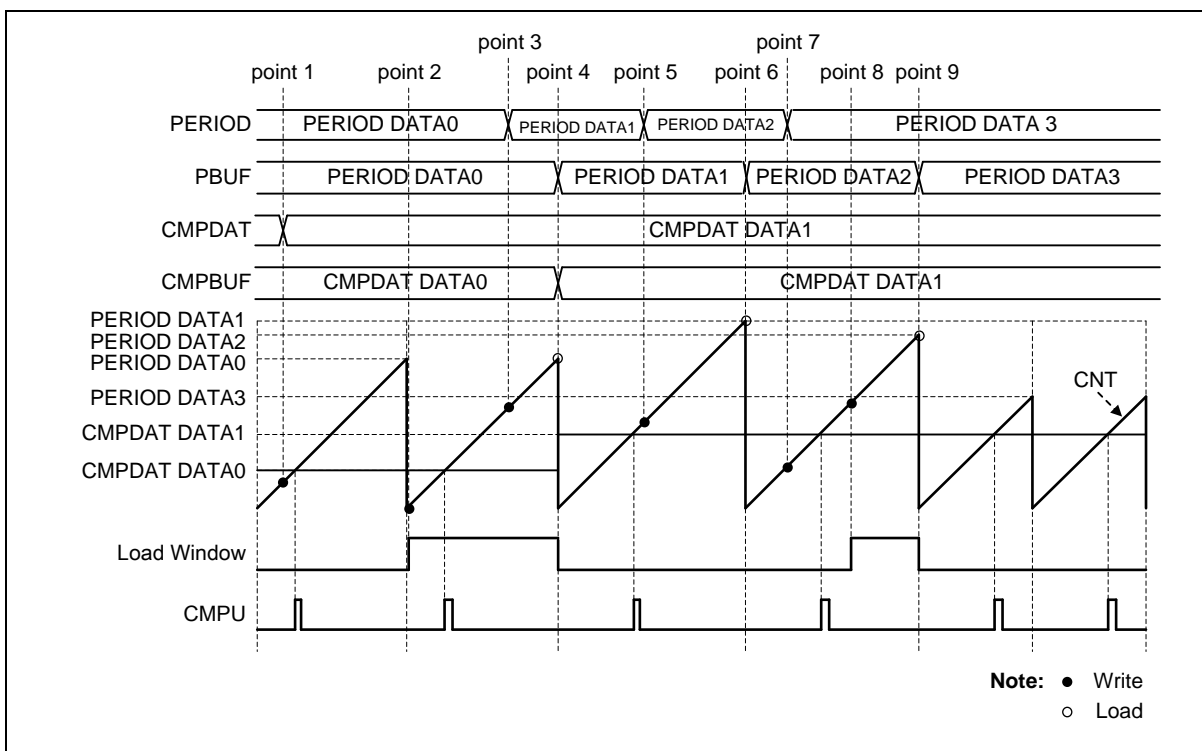


Figure 6.13-14 Window Loading in Up-Count Mode

### 6.13.5.11 Center Loading Mode

When the CTRLDn (PWM\_CTL0[5:0]) bit is set to 1 and PWM counter is set to up-down count type, CNTTYPE<sub>n</sub> (PWM\_CTL1[2n+1:2n], n = 0,1..5) is 0x2, PWM operates at center loading mode. In center loading mode, CMP(PWM\_CMPDAT<sub>n</sub>[15:0]) will load to active CMPBUF register in center of each period, that is, counter counts to PERIOD. CLKPSC(PWM\_CLKPSC<sub>n</sub>\_m[11:0]) and PERIOD(PWM\_PERIOD<sub>n</sub>[15:0]) will all load to their active CPSCBUF and PBUF registers while each period is completed.

Center loading mode can work with window loading mode, the CMP(PWM\_CMPDAT<sub>n</sub>[15:0]) will load to active CMPBUF register in center of each period, but it is valid only at the interval of load window. Figure 6.13-15 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.

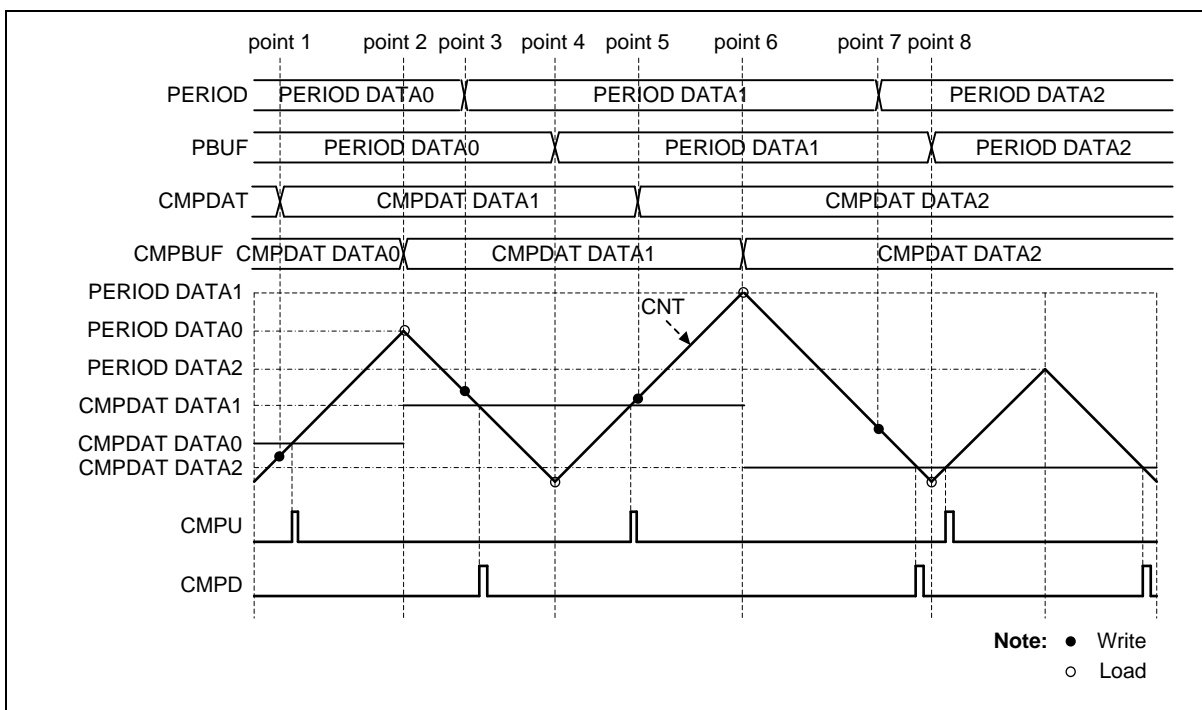


Figure 6.13-15 Center Loading in Up-Down-Count Mode

#### 6.13.5.12 PWM Counter Operation mode

The PWM counter supports two operation modes: One-shot mode and Auto-reload mode. PWM counter will operate in One-shot mode if CNTMODEN (PWM\_CTL1[21:16]) bit is set to 1, and operate in Auto-reload mode if set to 0.

In One-shot mode, CMPDAT and PERIOD registers should be written first and then set CNTENn (PWM\_CNTEN[5:0]) bit as 1 to enable PWM prescaler and counter start running. After PWM counter counted a period, counter value will keep in zero. User can re-start next one-shot by writing new value to CMP(PWM\_CMPDATn[15:0]) bits. If one-shot counter still running, to update CMPDAT register will cause next one-shot as continuous one-shot. Besides, to write CMPDAT register twice under continuous one-shot operation, latest value in CMPDAT register is valid at next one-shot period and only generate one-shot pulse once. Figure 6.13-16 is an example and following is steps sequence.

1. Software writes PERIOD DATA1 and hardware immediately loading PERIOD DATA1 to PBUF at point 1.
2. Software writes CMPDAT DATA1 which is equal to CMPDAT DATA0 at point 2 and hardware immediately loading CMPDAT DATA1 to CMPBUF, this event also trigger one-shot.
3. Software writes CMPDAT DATA2 and re-trigger next one-shot (continuous one-shot) at point 3.
4. Software writes CMPDAT DATA3 to cover CMPDAT DATA2 and re-trigger next one-shot at point 4.
5. Period loading CMPDAT DATA3 to CMPBUF at point 5.
6. There are no new CMPDAT write in the previous period, and the counter value is kept as zero at point 6.

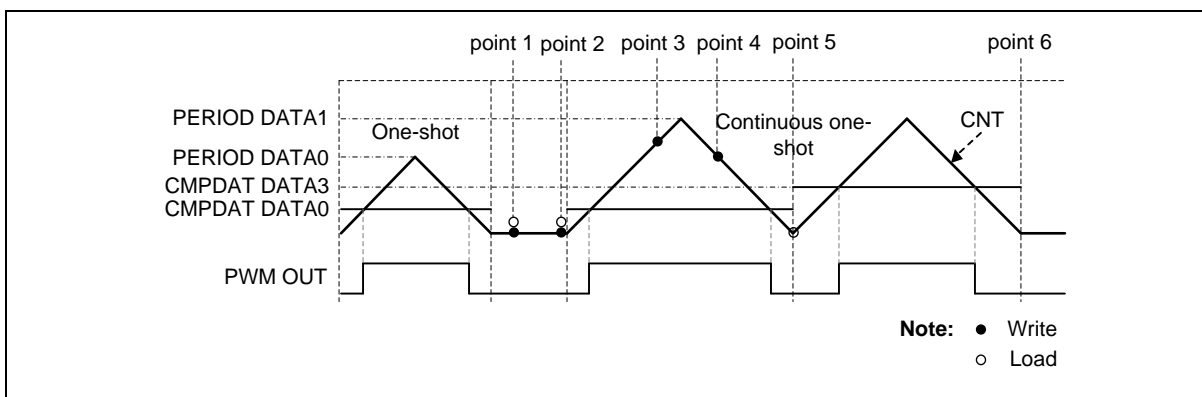


Figure 6.13-16 PWM One-shot Mode Output Waveform

In Auto-reload mode, CMPDAT and PERIOD registers should be written first and then the CNTENn(PWM\_CNTEN[n]) bit is set to 1 to enable PWM prescaler and start to run counter. The value of CLKPSC(PWM\_CLKPSCn\_m[11:0]), PERIOD(PWM\_PERIODn[15:0]) and CMP(PWM\_CMPDATn[15:0]) will auto reload to their active buffer according different loading mode. If PERIOD(PWM\_PERIODn[15:0]) is set to zero, PWM counter will be set to zero.

#### 6.13.5.13 PWM Pulse Generator

The PWM pulse generator uses counter and comparator events to generate PWM 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 and the other at down count. Besides, Complementary mode has two comparators compared with counter, and thus comparing equal points will become four in up-down counter type and two for up or down counter type.

Each event point can decide PWM waveform to do nothing (X), set Low (L), set High (H) or toggle (T) by setting the PWM\_WGCTL0 and PWM\_WGCTL1 registers. Using these points can easily generate asymmetric PWM pulse or variant waveform as shown in Figure 6.13-17. In the figure, PWM is in complementary mode, there are two comparators n and m to generate PWM pulse. n denotes even channel number 0, 2, or 4, and m denotes odd channel number 1, 3, or 5. n channel and m channel are complementary paired. Complementary mode uses two channels (CH0 and CH1, CH2 and CH3, or CH4 and CH5) as a pair of PWM outputs to generate complement paired waveforms. CMPU denotes CNT(PWM\_CNTn[15:0]) is equal to CMP(PWM\_CMPDATn[15:0]) when counting up. CMPD denotes CNT bits is equal to CMP bits when counting down.



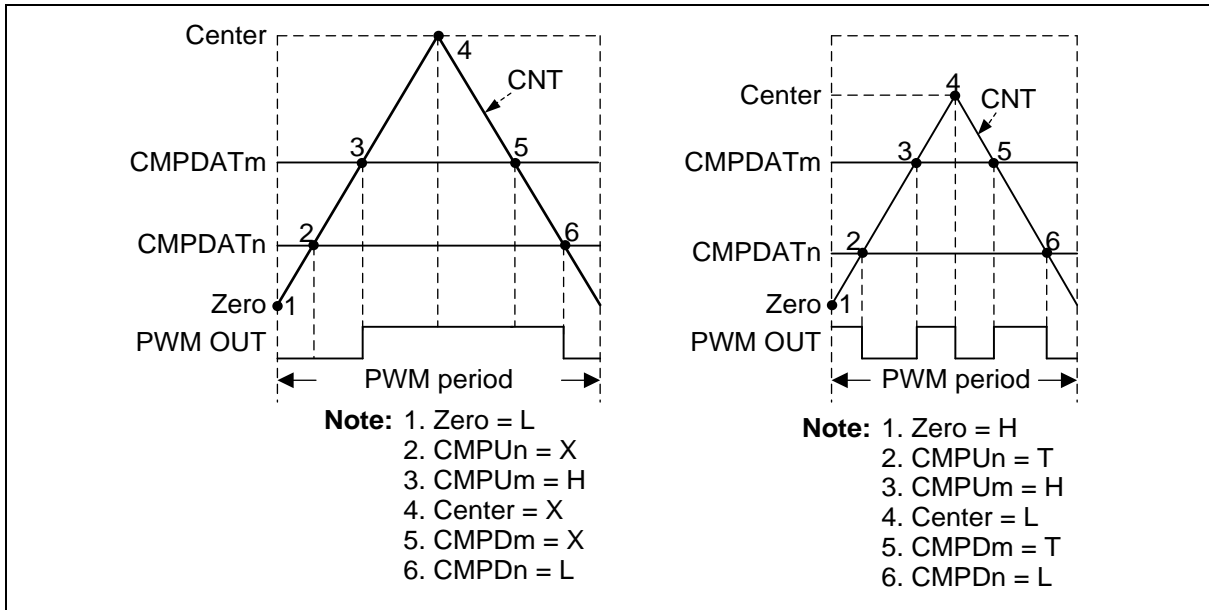


Figure 6.13-17 PWM Pulse Generation

The generation events may sometimes set to the same value, as the reason, events priority between different counter types are list below, up counter type (Table 6.13-2), down counter type (Table 6.13-3) and up-down counter type (Table 6.13-4). By using event priority, user can easily generate 0% to 100% duty pulse as shown in Figure 6.13-18.

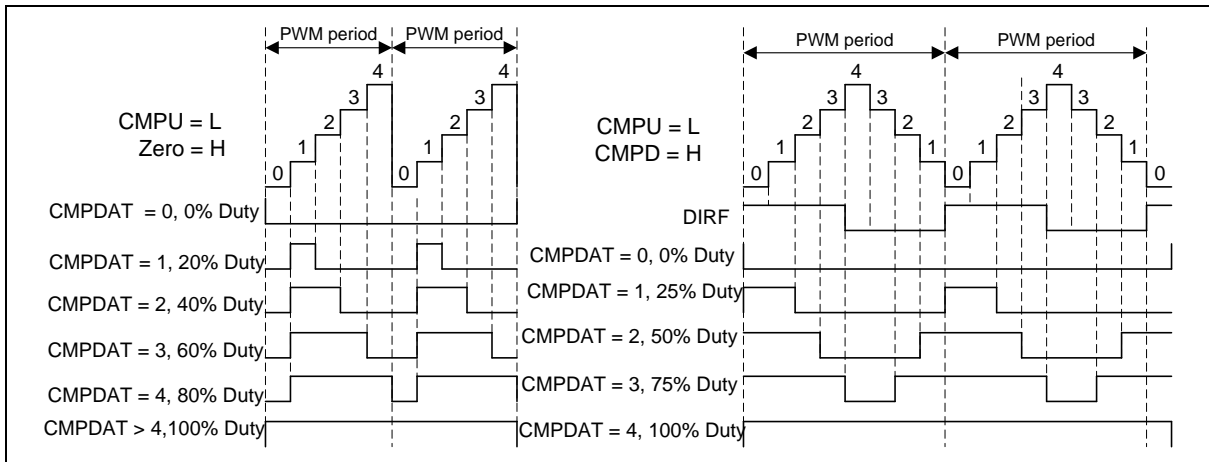


Figure 6.13-18 PWM 0% to 100% Pulse Generation

Priority	Up Event
1 (Highest)	Period event (CNT = PERIOD)
2	Compare up event of odd channel (CNT = CMPUm)
3	Compare up event of even channel (CNT = CMPUn)
4 (Lowest)	Zero event (CNT = zero)

Table 6.13-2 PWM Pulse Generation Event Priority for Up-Counter

Priority	Down Event
1 (Highest)	Zero event (CNT = zero)
2	Compare down event of odd channel (CNT = CMPDm)
3	Compare down event of even channel (CNT = CMPDn)
4 (Lowest)	Period event (CNT = PERIOD)

Table 6.13-3 PWM Pulse Generation Event Priority for Down-Counter

Priority	Up Event	Down Event
1 (Highest)	Compare up event of odd channel (CNT = CMPUm)	Compare down event of odd channel (CNT = CMPDm)
2	Compare up event of even channel (CNT = CMPUn)	Compare down event of even channel (CNT = CMPDn)
3	Zero event (CNT = zero)	Period (center) event (CNT = PERIOD)
4	Compare down event of odd channel (CNT = CMPDm)	Compare up event of odd channel (CNT = CMPUm)
5 (Lowest)	Compare down event of even channel (CNT = CMPDn)	Compare up event of even channel (CNT = CMPUn)

Table 6.13-4 PWM Pulse Generation Event Priority for Up-Down-Counter

6.13.5.14 PWM Output Mode

The PWM supports two output modes: Independent mode which may be applied to DC motor system, Complementary mode with dead-time insertion which may be used in the application of AC induction motor and permanent magnet synchronous motor.

6.13.5.15 Independent mode

By default, the PWM is operating in independent mode, independent mode is enabled when channel n corresponding PWMMODEn (PWM\_CTL1[26:24]) bit is set to 0. In this mode six PWM channels: PWM\_CH0, PWM\_CH1, PWM\_CH2, PWM\_CH3, PWM\_CH4 and PWM\_CH5 are running off its own period and duty as shown in Figure 6.13-19.

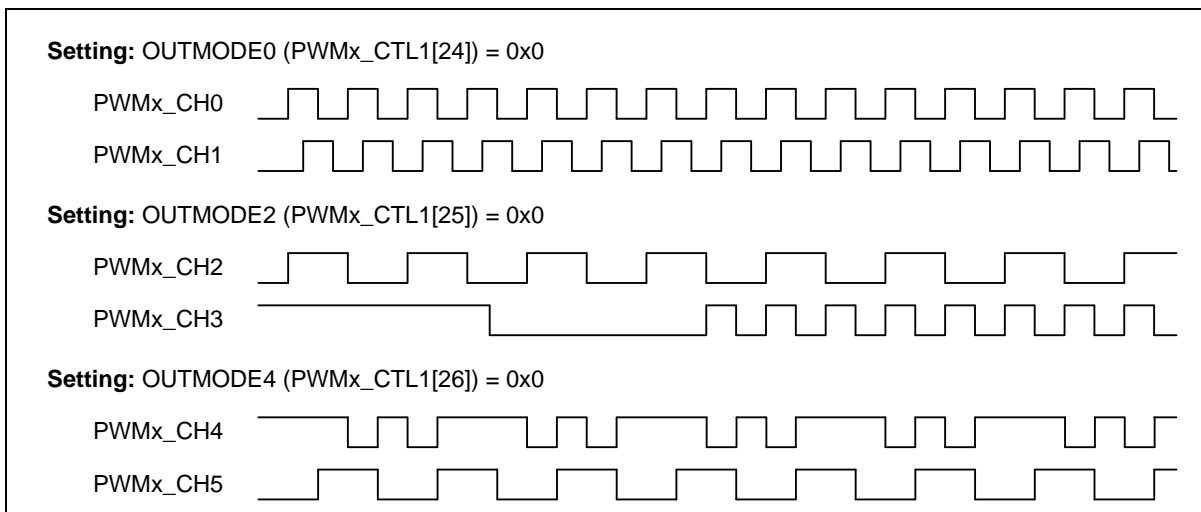


Figure 6.13-19 PWM Independent Mode Waveform

6.13.5.16 Complementary Mode

Complementary mode is enabled when the pair channel corresponding PWMMODEN (PWM\_CTL1[26:24]) bit set to 1. In this mode there are 3 PWM generators utilized for complementary mode, with total of 3 PWM output paired pins in this module. In Complimentary modes, the internal odd PWM signal must always be the complement of the corresponding even PWM signal. PWM\_CH1 will be the complement of PWM\_CH0. PWM\_CH3 will be the complement of PWM\_CH2 and PWM\_CH5 will be the complement of PWM\_CH4 as shown in Figure 6.13-20.

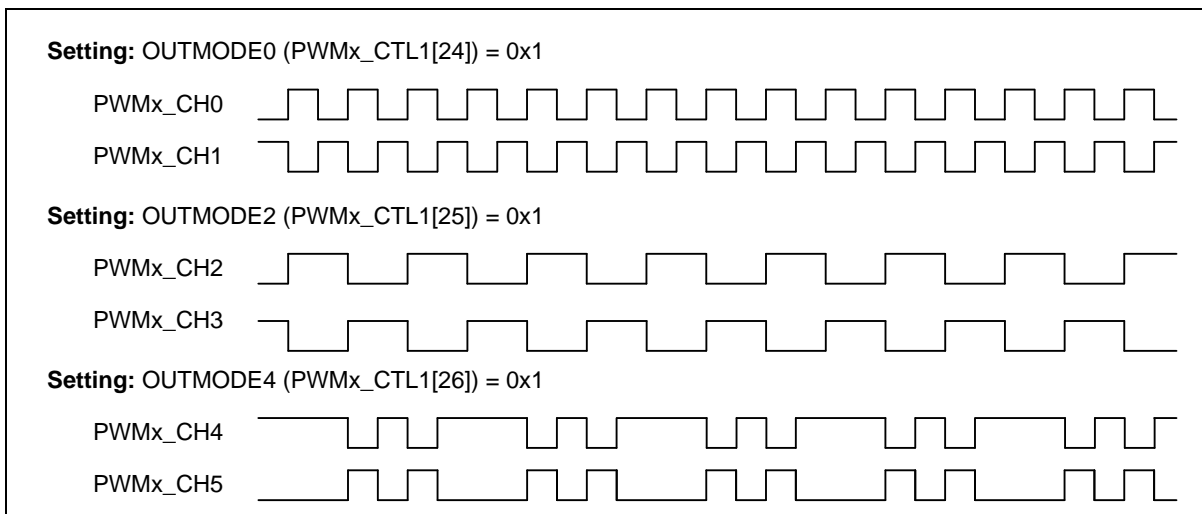


Figure 6.13-20 PWM Complementary Mode Waveform

6.13.5.17 PWM Output Function

Based on the output mode, there are two output functions: group and synchronous functions for advanced output control. Group function, forces the PWM\_CH2 and PWM\_CH4 synchronous with PWM\_CH0 generator and forces the PWM\_CH3 and PWM\_CH5 synchronous with PWM\_CH1, may simplify updating duty control in DC and BLDC motor applications. Besides, Synchronous function makes any channel of PWM0 in phase, user can control phase value and direction.

6.13.5.18 Group function

Group function is enabled when GROUPEN (PWM\_CTL0[24]) is set to 1, no matter in independent or complementary mode. This control allows all even PWM channels output to be controllable by PWM\_PERIOD0 and PWM\_CMPDAT0 registers and all odd PWM channels output to be controllable by PWM\_PERIOD1 and PWM\_CMPDAT1 registers. That is, user only needs to set PWM\_CH0 to get PWM\_CH0, PWM\_CH2 and PWM\_CH4 output the same pulse, and set PWM\_CH1 to get PWM\_CH1, PWM\_CH3 and PWM\_CH5 output the same pulse, as shown in Figure 6.13-21. When operating group function, PWMMODE0, PWMMODE2 and PWMMODE4 bits of CTL1 register must all set to 0 for independent mode or all set to 1 for complementary mode.

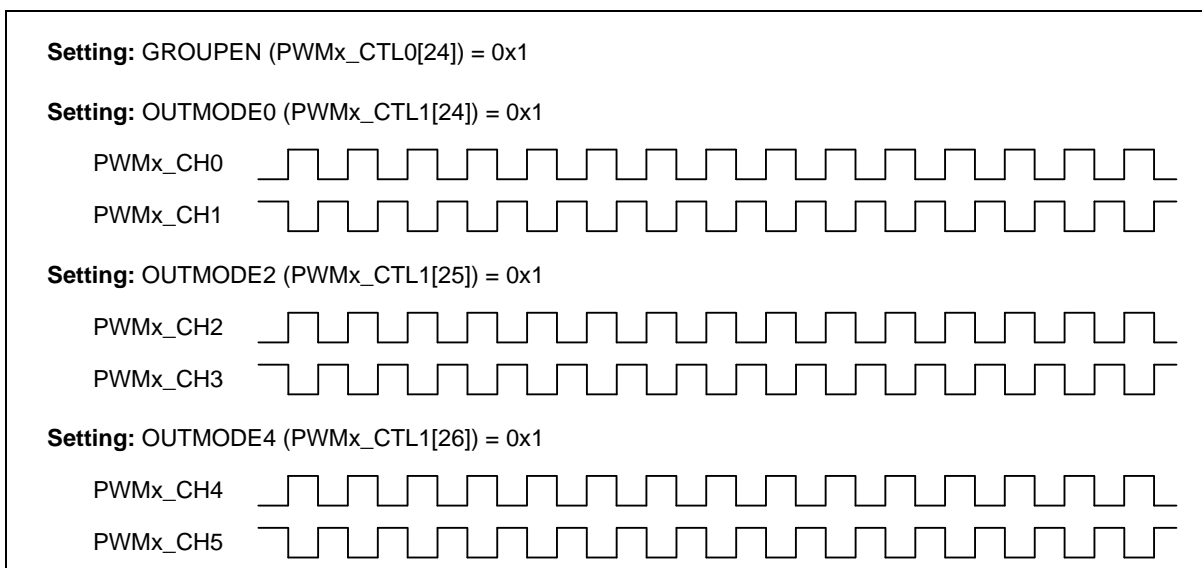


Figure 6.13-21 PWM Group Function Waveform

6.13.5.19 Synchronous function

Synchronous function can only be enabled when complementary mode is enabled. Figure 6.13-23 is counter synchronous function block diagram. Every counter of PWM pairs has a SYNC\_IN and a SYNC\_OUT signals. The SYNC\_IN signal for the first PWM0 pair counter comes from PWM0\_SYNC\_IN pin, and the others come from the SYNC\_OUT signal of previous PWM pair counter. The input signal from PWM0\_SYNC\_IN pin will be filtered by a 3-bit noise filter as Figure 6.13-22. In addition, it can be inverted by setting the bit SINPINV (PWM\_SYNC[23]) to realize the polarity setup for the input signal. The noise filter sampling clock can be selected by setting bits SFLTCSEL (PWM\_SYNC[19:17]) to fit different noise properties. Moreover, by setting the bits SFLTCNT (PWM\_SYNC[22:20]), user can define by how many sampling clock cycles a filter will recognize the effective edge of the SYNC\_IN signal. Configuring the SNFLTEN (PWM\_SYNC[16]) will enable the noise filter function. By default, it is disabled.

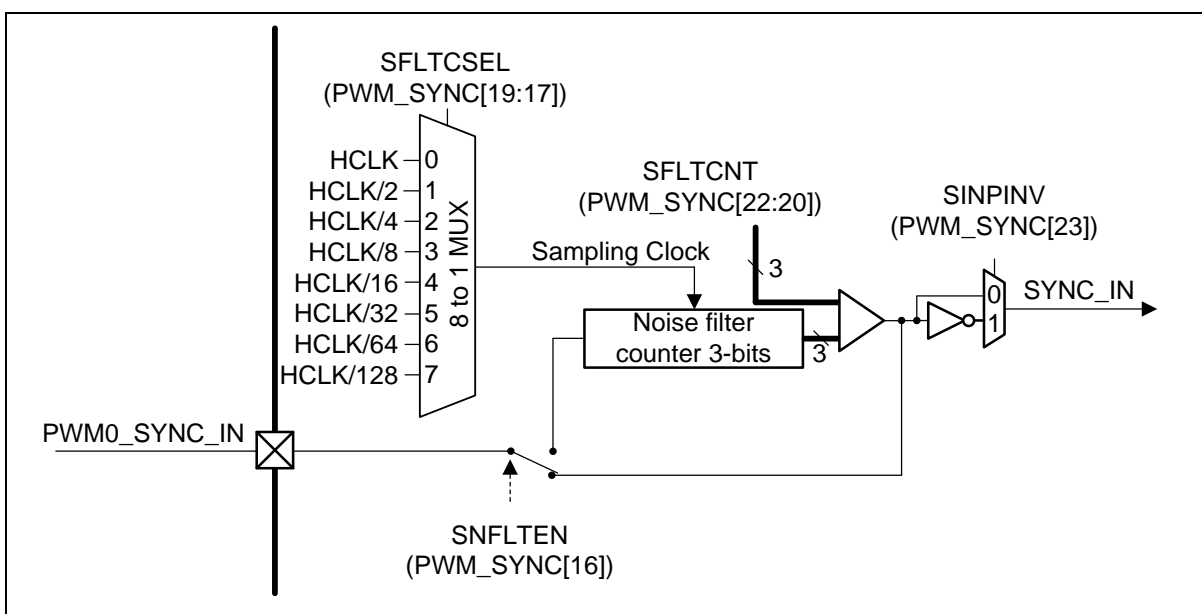


Figure 6.13-22 PWM SYNC\_IN Noise Filter Block Diagram

The SINSRCn (PWM\_SYNC[13:8]) bits can be used to select the synchronize source. When SINSRCn bits is set to 0, PWM0\_SYNC\_IN pin “OR” set SWSYNCn (PWM\_SWSYNC[2:0]) to 1 can generate SYNC\_IN signal for the next counter’s synchronization . Synchronizing source can also be selected as CNT = 0 or CNT = PWM\_CMPDATm register (if being the up-down counter type, it will synchronize twice in a PWM period) to trigger a sync event or to disable SYNC\_OUT signal.

When the PHSEn (PWM\_SYNC[2:0]) is enabled and the synchronous source has a happening event, the counter will load a value from the PHS (PWM\_PHSn\_m[15:0]) register. This method synchronizes counters to different phase in the same time. In the up-down counter type, user can set the value in PHSDIRn (PWM\_SYNC[26:24]) to control the counter direction after synchronization. Although the Synchronous function can synchronize channels in phase, it can’t work from the beginning of PWM enable.

To start PWM counters in the same time, user have to set the PWM Synchronous Start Control Register (PWM\_SSCTL[5:0]) to enable the channel counters which are planned to start counting together, followed by setting the PWM Synchronous Start Trigger Register CNTSEN (PWM\_SSTRG[0]).

For applications, please do not use Group and Synchronous function simultaneously because the Synchronous function will be inactive.

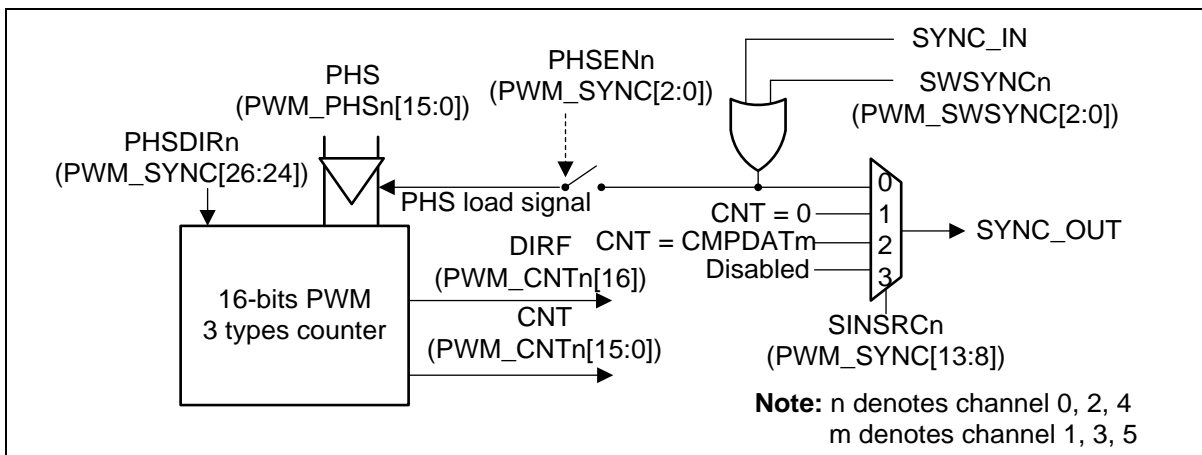


Figure 6.13-23 PWM Counter Synchronous Function Block Diagram

Figure 6.13-24 is an example of the synchronous function in the up-down counter type. In the example, synchronizing source comes from the external PWM SYNC\_IN signal. At the beginning, the output waveform of PWM\_CH0, PWM\_CH2 and PWM\_CH4 are in the same phase. Then at Point A, the PWM SYNC input signal comes as a sync event, resulting in phase shifts and counting direction changes for all of the counters. To realize the altered counter behaviors before the sync event coming, user has to setup the corresponding phase value in the PHS (PWM\_PHSn\_m[15:0]) as well as the counting direction in the PHSDIRn (PWM\_SYNC[26:24]). In this case, one third of phase shifts are made by setting the corresponding channel n’s counter counting direction after synchronizing, as illustrated around the left side of Figure 6.13-24.

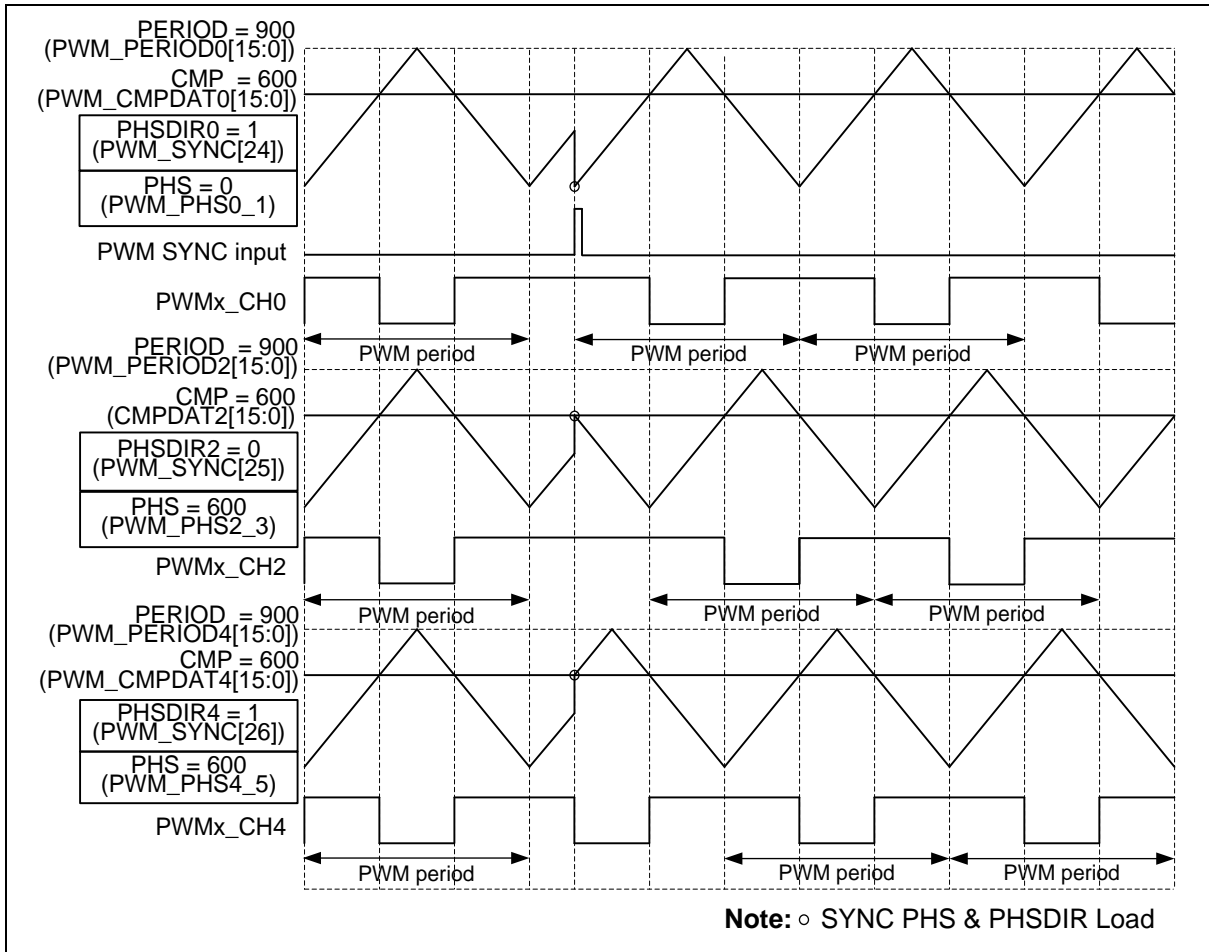


Figure 6.13-24 PWM Synchronous Function with Synchronize source from SYNC\_IN Signal

6.13.5.20 PWM Output Control

After PWM pulse generation, there are four to six steps to control the output of PWM channels. In independent mode, there are Mask, Brake, Pin Polarity and Output Enable four steps as shown in Figure 6.13-25. In complementary mode, it needs two more steps to precede these four steps, Complementary channels and Dead-Time Insertion as shown in Figure 6.13-26.

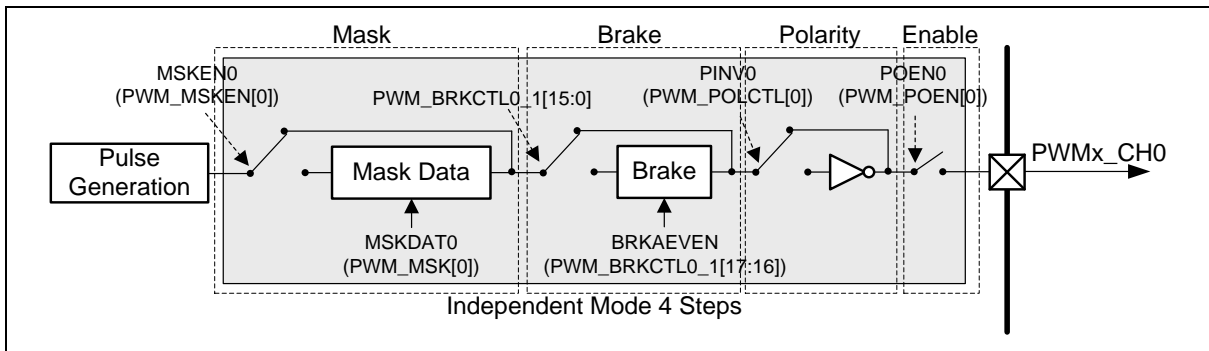


Figure 6.13-25 PWMx\_CH0 Output Control in Independent Mode

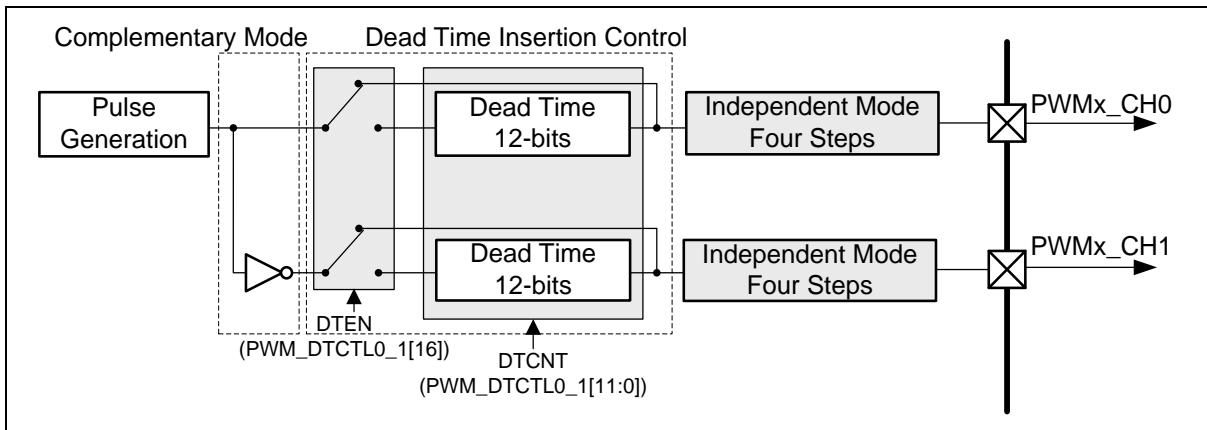


Figure 6.13-26 PWMx\_CH0 and PWMx\_CH1 Output Control in Complementary Mode

6.13.5.21 Dead-Time Insertion

In the complementary application, the complement channels may drive the external devices like power switches. The dead-time generator inserts a low level period called “dead-time” between complementary outputs to drive these devices safely and to prevent system or devices from the burn-out damage. Hence the dead-time control is a crucial mechanism to the proper operation of the complementary system. By setting corresponding channel n DTEN (PWM\_DTCTLn\_m[16]) bit to enable dead-time function and DTCNT (PWM\_DTCTLn\_m[11:0]) to control dead-time period, the dead-time can be calculated from the following formula:

$$\text{Dead-time} = (\text{DTCNT (PWM\_DTCTLn[11:0])} + 1) * \text{PWMx\_CLK period}$$

Dead-time insertion clock source can be selected from prescaler output by setting DTCKSEL (PWM\_DTCTLn\_m[24]) to 1. By default, clock source is come from PWM\_CLK, which is prescaler input. Then the dead-time can be calculated from the following formula:

$$\text{Dead-time} = (\text{DTCNT (PWM\_DTCTLn[11:0])} + 1) * (\text{CLKPSC (PWM\_CLKPSCn [11:0])} + 1) * \text{PWMx\_CLK period}$$

Please note that the PWM\_DTCTLn\_m are write-protected registers.

Figure 6.13-27 indicates the dead-time insertion for one pair of PWM signals.

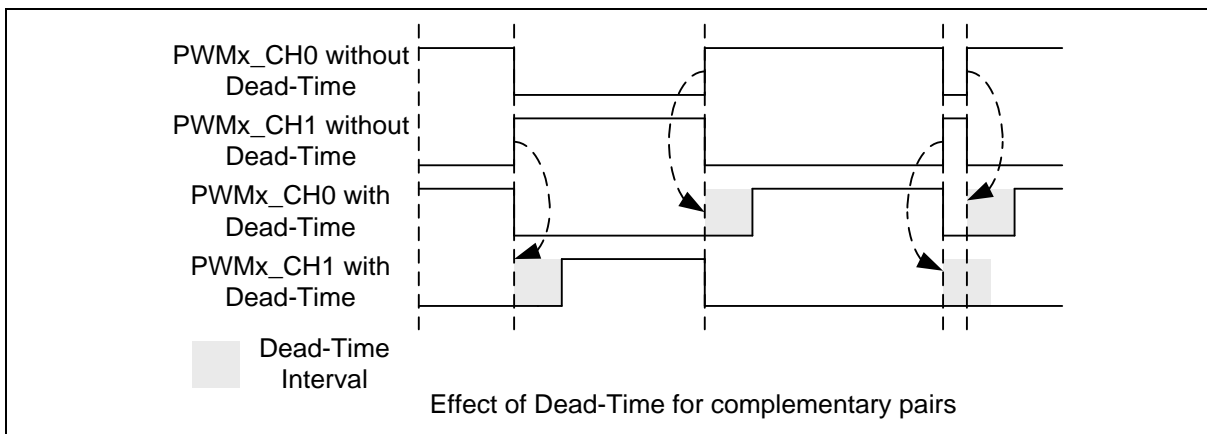


Figure 6.13-27 Dead-Time Insertion

6.13.5.22 PWM Mask Output Function

Each of the PWM channel output value can be manually overridden with the settings in the PWM Mask Enable Control Register (PWM\_MSKEN) and the PWM Masked Data Register (PWM\_MSK). With

these settings, the PWM channel outputs can be assigned to specified logic states independent of the duty cycle comparison units. The PWM mask bits are useful when controlling various types of Electrically Commutated Motor (ECM) like a BLDC motor. The PWM\_MSKEN register contains six bits, MSKENn(PWM\_MSKEN[5:0]). If the MASKENn is set to active-high, the PWM channel n output will be overridden. The PWM\_MSK register contains six bits, MSKDATn(PWM\_MSK[5:0]). The bit value of the MSKDATn determines the state value of the PWM channel n output when the channel is overridden. Figure 6.13-28 shows an example of how PWM mask control can be used for the override feature.

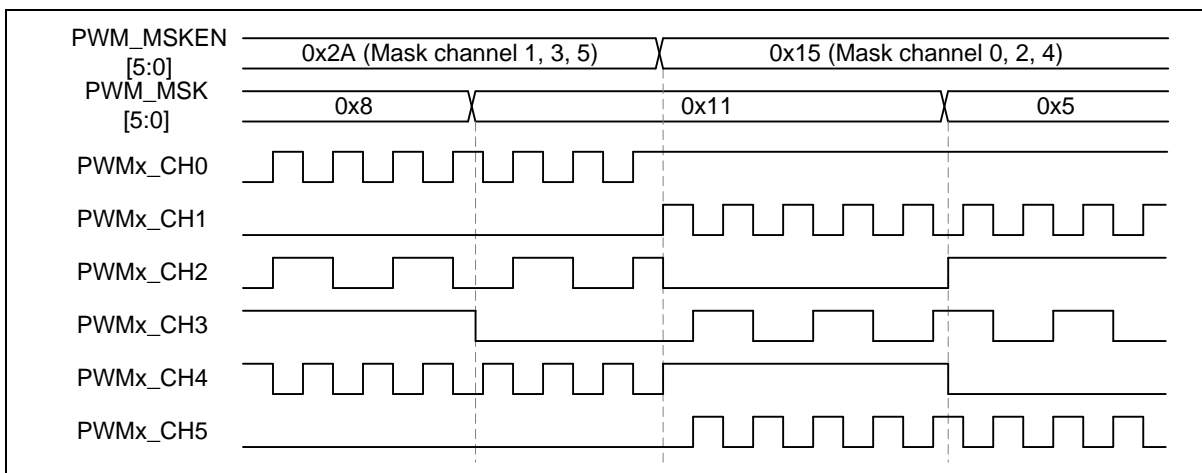


Figure 6.13-28 Illustration of Mask Control Waveform

### 6.13.5.23 PWM Brake

Each PWM module has two external input brake control signals. User can select active brake pin source is from PWMx\_BRAKEy pin by BKxSRC bits of BNF register(x=0,1, y=0,1). The external signals will be filtered by a 3-bits noise filter. User can enable the noise filter function by BRKxNFEN bits of BNF register, and noise filter sampling clock can be selected by setting BRKxNFSEL bits of BNF register to fit different noise properties. Moreover, by setting the BRKxFCNT bits, user can define by how many sampling clock cycles a filter will recognize the effective edge of the brake signal.

In addition, it can be inverted by setting the BRKxPINV (x denotes input external pin 0 or 1) bits of BNF register to realize the polarity setup for the brake control signals. Set BRKxPINV bit to 0, brake event will occurred when PWMx\_BRAKEy(x=0,1, y=0,1) pin status is from low to high; set BRKxPINV to 1, brake event will occurred when PWMx\_BRAKEy pin status is from high to low.



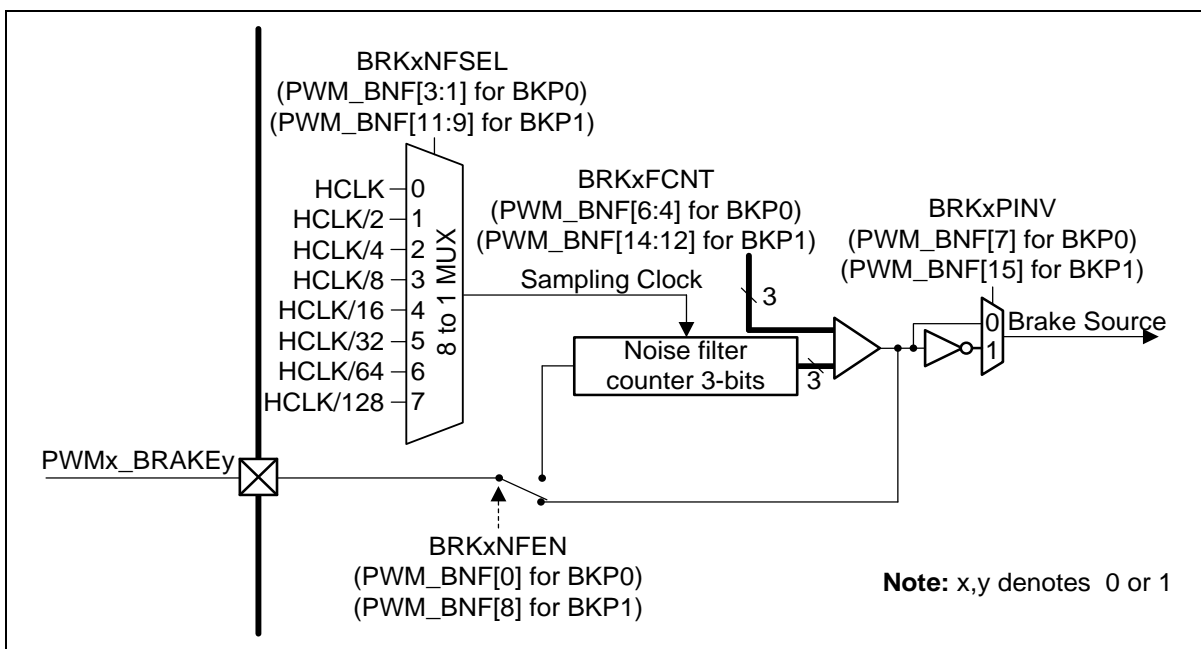


Figure 6.13-29 Brake Noise Filter Block Diagram

For Complementary mode, it is often necessary to set a safe output state to the complementary output pairs once the brake event occurs.

Each complementary channel pair shares a PWM brake function, as shown Figure 6.13-30. To control paired channels to output safety state, user can setup BRKAEVEN (PWM\_BRKCTL0\_1[17:16]) for even channels and BRKAODD (PWM\_BRKCTL0\_1[19:18]) for odd channels when the fault brake event happens. There are two brake detectors: Edge detector and Level detector. When the edge detector detects the brake signal and BRKEIEN<sub>n,m</sub> (PWM\_INTEN1[2:0]) is enabled, the brake function generates BRK\_INT. This interrupt needs software to clear, and the BRKESTS<sub>n</sub> (PWM\_INTSTS1[21:16]) brake state will keep until the next PWM period starts after the interrupt cleared. The brake function can also operate in another way through the level detector. Once the level detector detects the brake signal and the BRKLIEN<sub>n,m</sub> (PWM\_INTEN1[10:8]) is also enabled, the brake function will generate BRK\_INT, but BRKLSTS<sub>n</sub> (PWM\_INTSTS1[29:24]) brake state will auto recovery to normal output while level brake source recovery to high level and pass through “Low Level Detection” at the PWM waveform period when brake condition removed without clear interrupt.

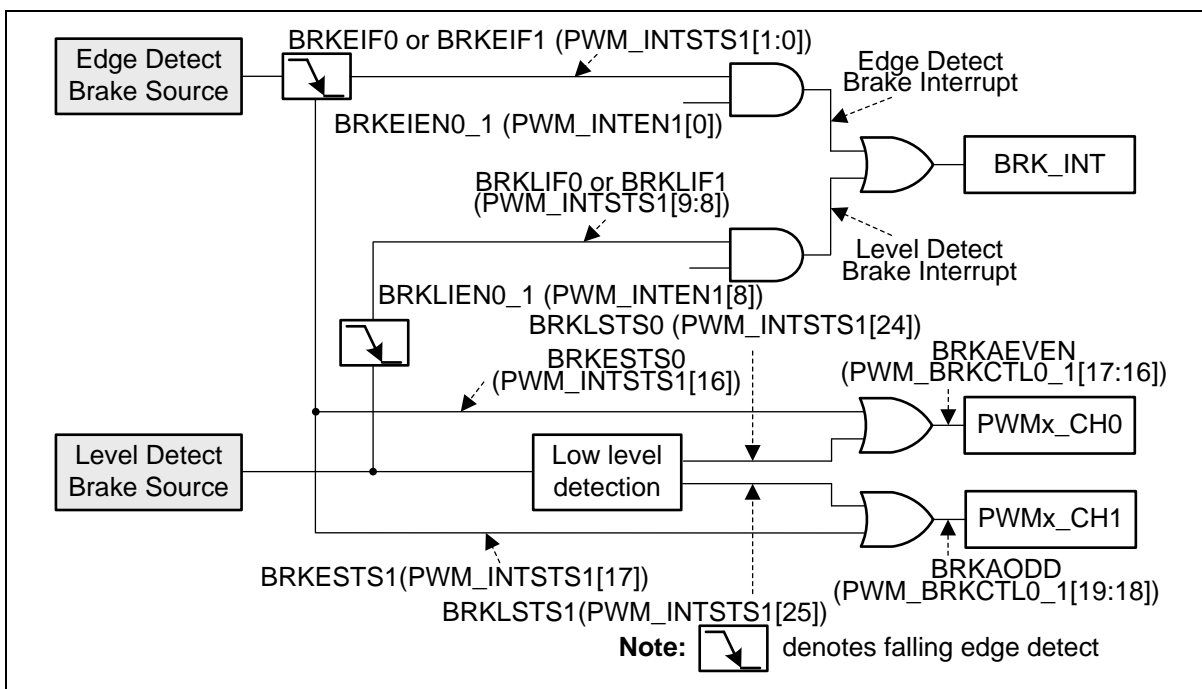


Figure 6.13-30 Brake Block Diagram for PWMx\_CH0 and PWMx\_CH1 Pair

Figure 6.13-31 illustrates the edge detector waveform for PWMx\_CH0 and PWMx\_CH1 pair. In this case, the edge detect brake source has occurred twice for the brake events. When the event occurs, both of the BRKEIF0 and BRKEIF1 flags are set and BRKESTS0 and BRKESTS1 bits are also set to indicate brake state of PWMx\_CH0 and PWMx\_CH1. For the first occurring event, software writes 1 to clear the BRKEIF0 flag. After that, the BRKESTS0 bit is cleared by hardware at the next start of the PWM period. At the same moment, the PWMx\_CH0 outputs the normal waveform even though the brake event is still occurring. The second event also triggers the same flags, but at this time, software writes 1 to clear the BRKEIF1 flag. Afterward, PWMx\_CH1 outputs normally at the next start of the PWM period.

As a contrast to the edge detector example, Figure 6.13-32 illustrates the level detector waveform for PWMx\_CH0 and PWMx\_CH1 pair. In this case, the BRKLIF0 and BRKLIF1 flags can only indicate the brake event having occurred. The BRKLSTS0 and BRKLSTS1 brake states will automatically recover at the start of the next PWM period no matter at what states the BRKLIF0 and BRKLIF1 flags are at that moment.

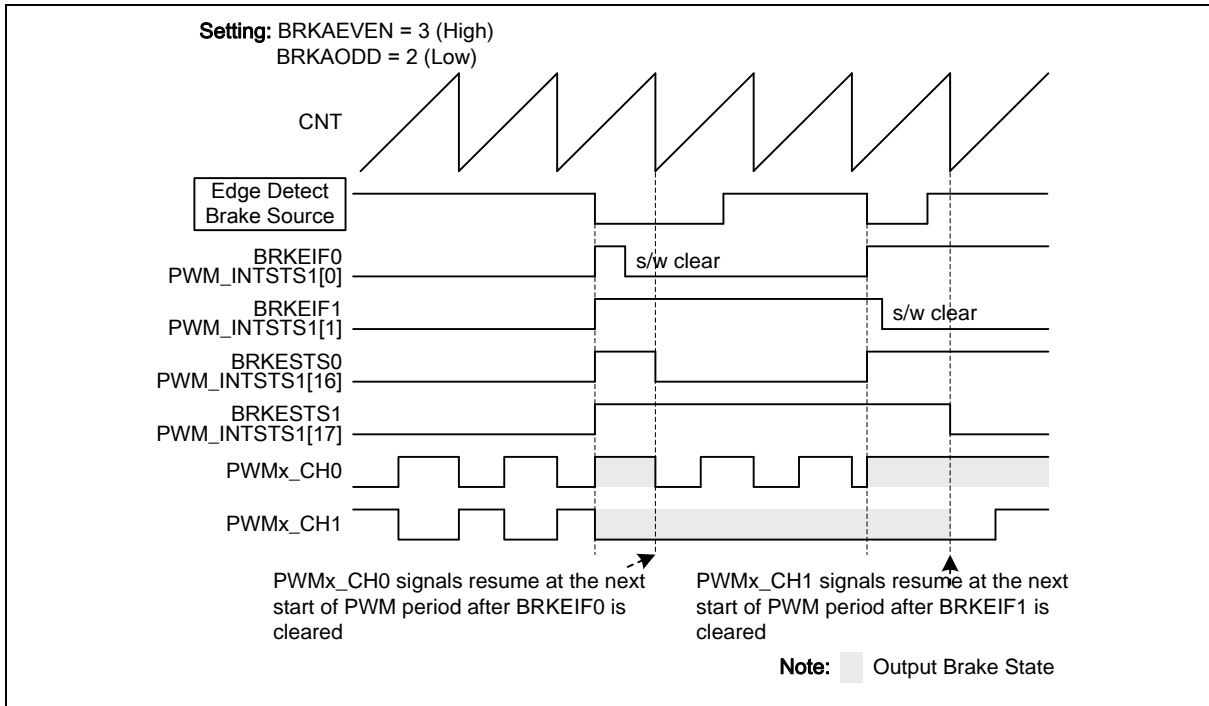


Figure 6.13-31 Edge Detector Waveform for PWMx\_CH0 and PWMx\_CH1 Pair

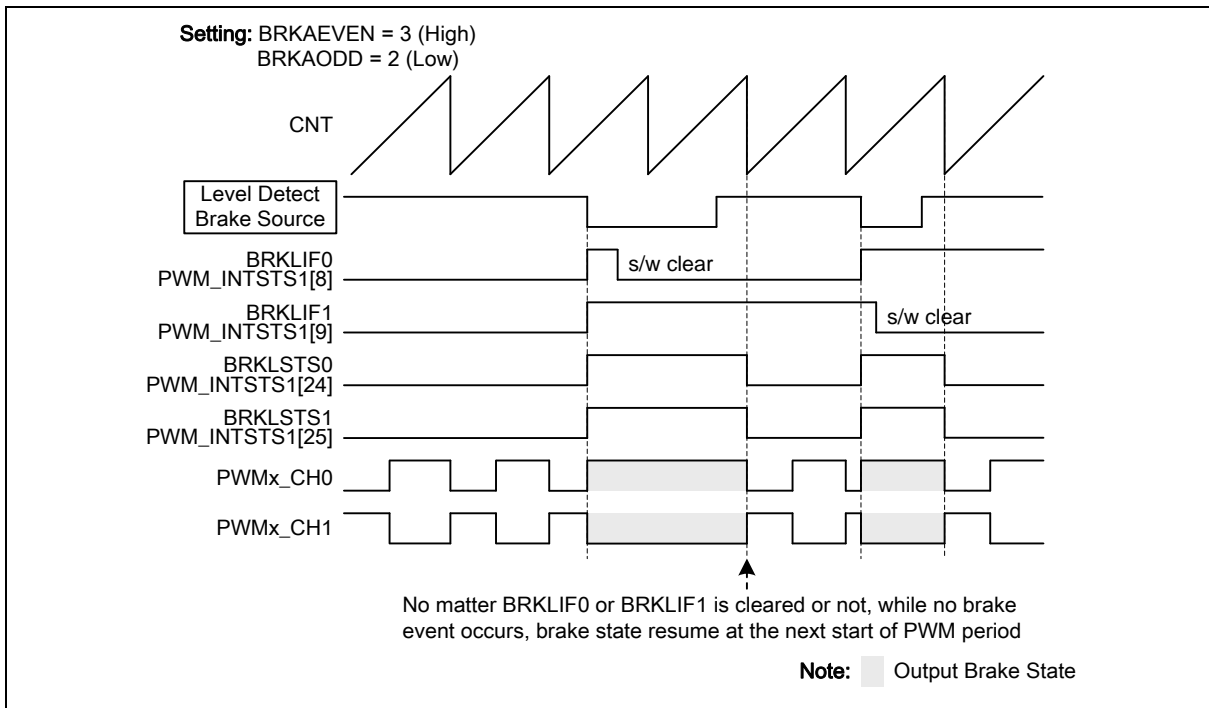


Figure 6.13-32 Level Detector Waveform for PWMx\_CH0 and PWMx\_CH1 Pair

The two kinds of detectors detect the same seven brake sources: two from external input signals, two from analog comparators(ACMP), one from ADC result monitor (ADCRM), one from system fail and one from software triggered, that are shown in Figure 6.13-33. ACMP brake sources will be detected only when internal ACMP0\_O or ACMP1\_O signal from low to high.

Among the above described brake sources, the brake source coming from system fail can still be

specified to several different system fail conditions. These conditions include clock fail, Brown-out detect and Cortex®-M0 lockup. Figure 6.13-34 shows that by setting corresponding enable bits, the enabled system fail condition can be one of the sources to issue the Brake system fail to the PWM brake.

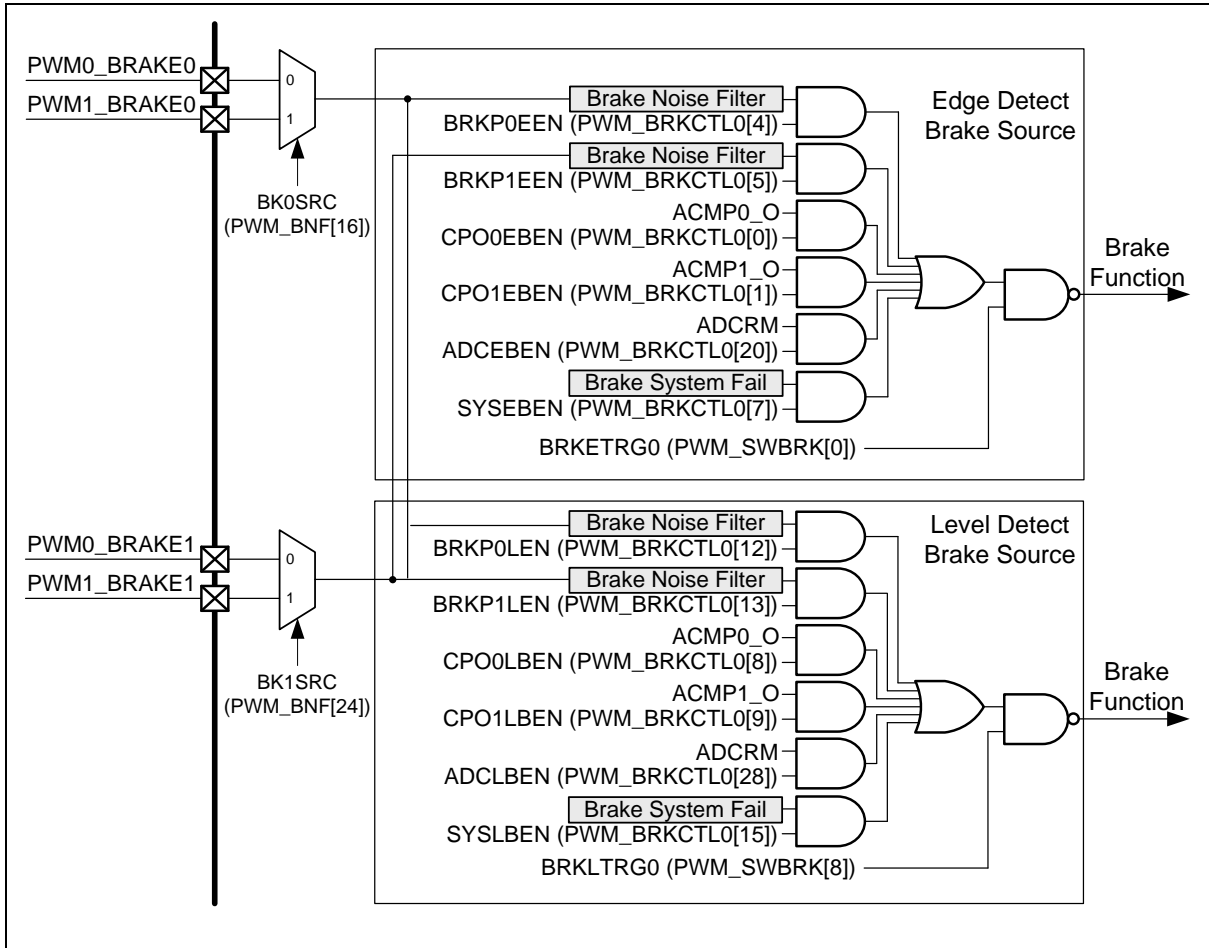


Figure 6.13-33 Brake Source Block Diagram

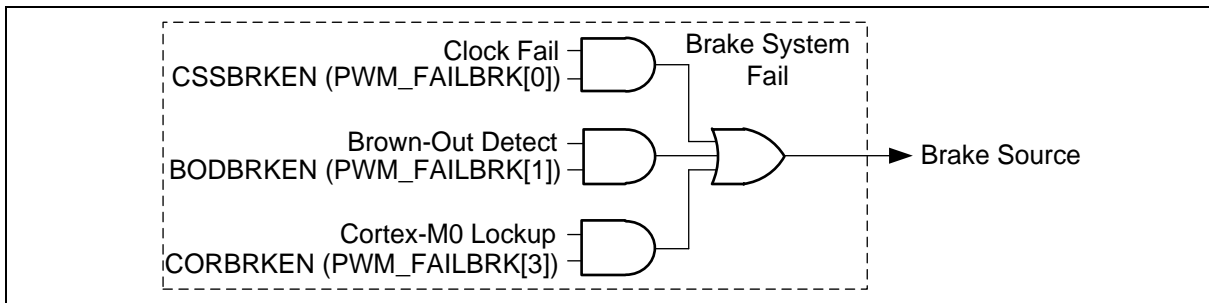


Figure 6.13-34 Brake System Fail Block Diagram

6.13.5.24 LEB Function

Leading edge blanking (LEB) function is use to blank the false trigger from ACMP brake source which may caused by PWM output transition. Set LEBEN (PWM\_LEBCTL[0]) bit to 1 can enable this function. LEB source is come from PWMx\_CH0, PWMx\_CH2 and PWMx\_CH4, use SRCENn (PWM\_LEBCTL[10:8]) bits as input source enable bits. LEB function blanking time is decide by LEBCNT (PWM\_LEBCNT[8:0]) bits, when LEB detected trigger edge, then blanking time will count

from LEBCNT+1 to 0, the counter clock base is ECLK. If new trigger event occur, blanking counter will reset to LEBCNT and down count again. LEB trigger edge can be rising, falling or both rising and falling edge by setting TRGTYPE (PWM\_LEBCTL[17:16]) bits. Figure 6.13-35 shows that LEB will blank leading edge caused by PWMx\_CH0 and PWMx\_CH4.

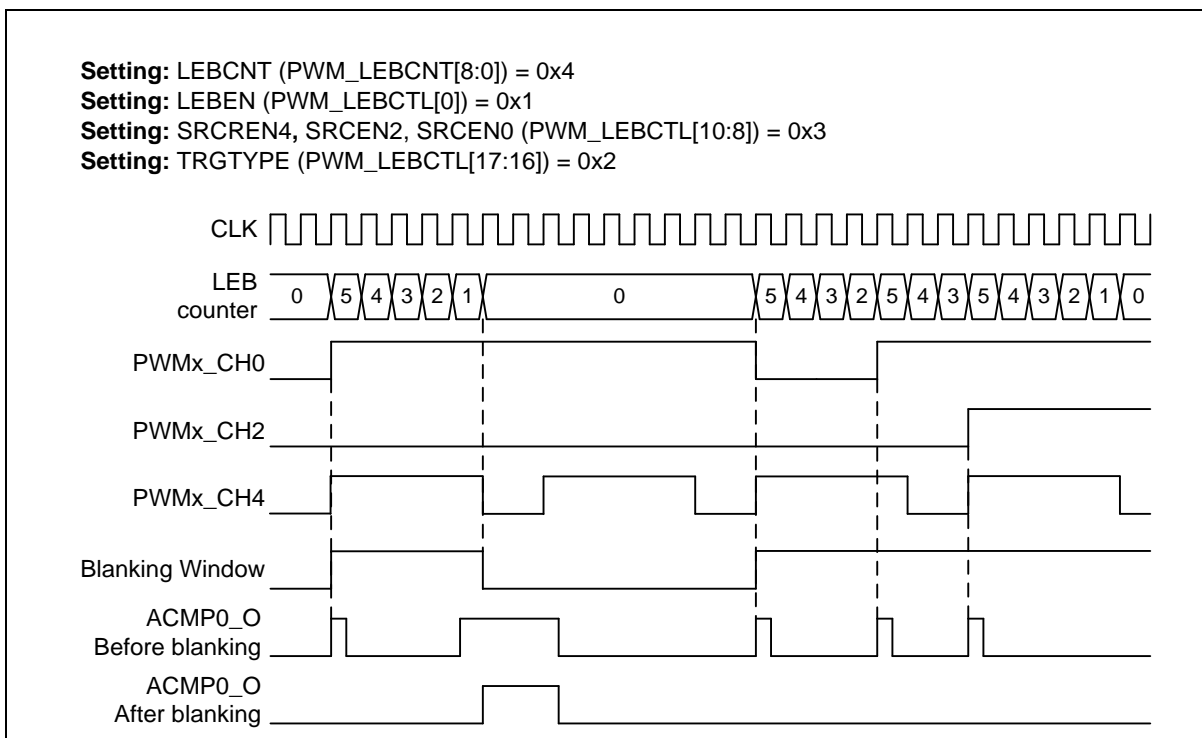


Figure 6.13-35 PWM LEB Function Waveform

### 6.13.5.25 Polarity Control

Each PWM channel, from PWMx\_CH0 to PWMx\_CH5, has an independent polarity control module to configure the polarity of the active state of the PWM output. By default, the PWM output is active high. This implies the PWM OFF state is low and ON state is high. This definition is variable through setting the PINVn (POLCTL[n], n=0,1..5) bits, for each individual PWM channel. Figure 6.13-36 shows the initial state before PWM starting with different polarity settings.

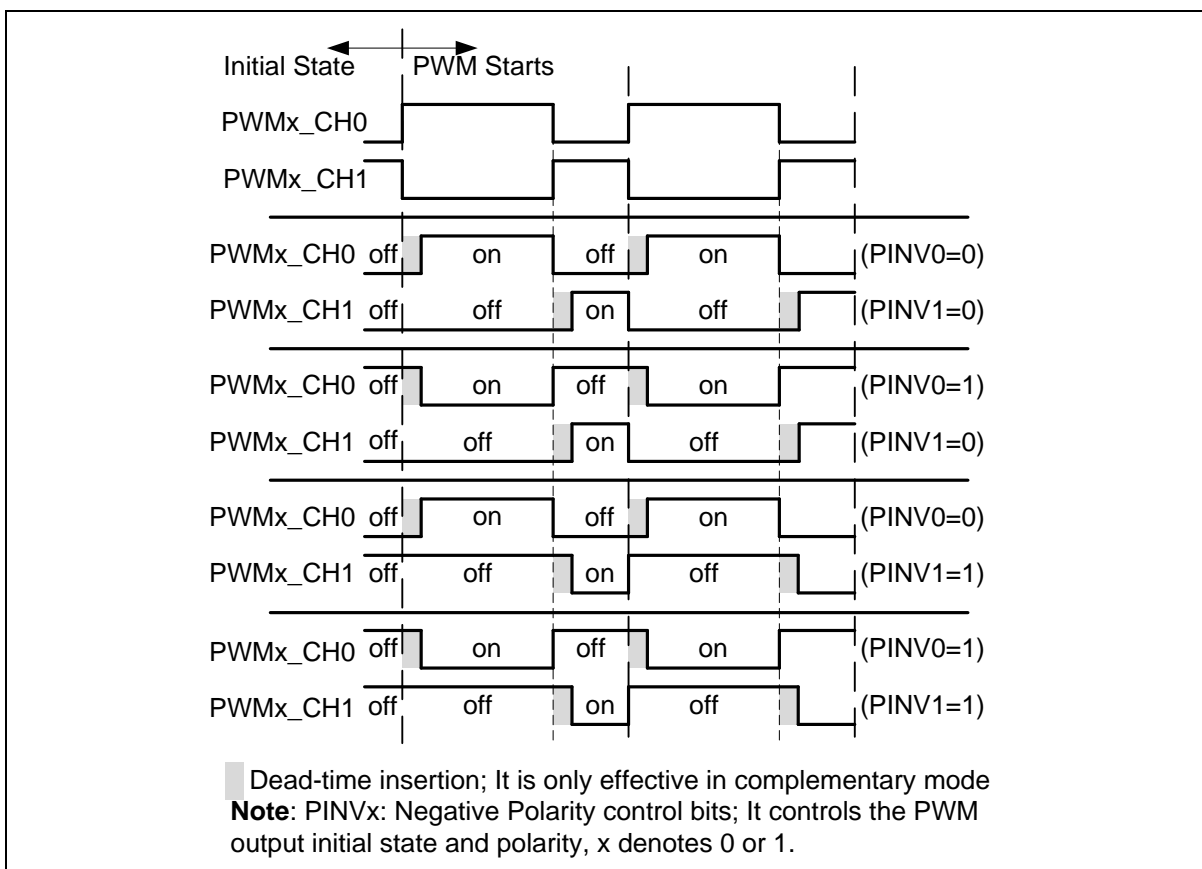


Figure 6.13-36 Initial State and Polarity Control with Rising Edge Dead-Time Insertion

### 6.13.5.26 PWM Interrupt Generator

There are three independent interrupts for each PWM as shown in Figure 6.13-38.

The 1<sup>st</sup> PWM interrupt (PWM\_INT) comes from PWM complementary pair events. The counter can generate the Zero point Interrupt Flag ZIFn (PWM\_INTSTS0[5:0], n=0,1..5) and the Period point Interrupt Flag PIFn (PWM\_INTSTS0[13:8], n=0,1..5). When PWM channel n's counter equals to the comparator value stored in PWM\_CMPDATn register, 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 (PWM\_INTSTS0[21:16]) is set and if matching at the opposite direction, the Down Interrupt Flag CMPDIFn (PWM\_INTSTS0[29:24]) is set. If the corresponding interrupt enable bits are set, the trigger events will generate interrupt signals.

PWM\_INT can use the PWM\_IFA register to accumulate the number of times that the interrupt flags have been triggered. By setting IFAENn\_m (IFAEN0\_1 (PWM\_IFA[7]), IFAEN2\_3 (PWM\_IFA[15]) and IFAEN4\_5 (PWM\_IFA[23])) bits to 1 to enable accumulator. When the accumulator is enabled, PWM\_INT will switch interrupt source from every event trigger interrupt to trigger interrupt once every accumulate times.

By setting the IFSELn\_m (IFSEL0\_1 (PWM\_IFA[6:4]), IFSEL2\_3 (PWM\_IFA[14:12]) and IFSEL4\_5 (PWM\_IFA[22:20])) bits, user can select one of the 8 interrupt sources to accumulate, and compare with IFCNTn\_m (IFCNT0\_1 (PWM\_IFA[3:0]), IFCNT2\_3 (PWM\_IFA[11:8]) and IFCNT4\_5 (PWM\_IFA[19:16])) bits, when interrupt accumulator equals IFCNTn\_m then set IFAIFn\_m (IFAIF0\_1 (PWM\_INTSTS0[7]), IFAIF2\_3 (PWM\_INTSTS0[15]) and IFAIF4\_5 (PWM\_INTSTS0[23])) bits as PWM\_INT signal when enable IFAIENn\_m (IFAIEN0\_1 (PWM\_INTEN0[7]), IFAIEN2\_3 (PWM\_INTEN0[15]) and IFAIEN4\_5 (PWM\_INTEN0[23])) bits. Figure 6.13-37 is an example of channel 0 and channel 1 pair using PWM\_IFA register to output PWM\_INT once every IFCNT0\_1+1 times interrupt events occurred.

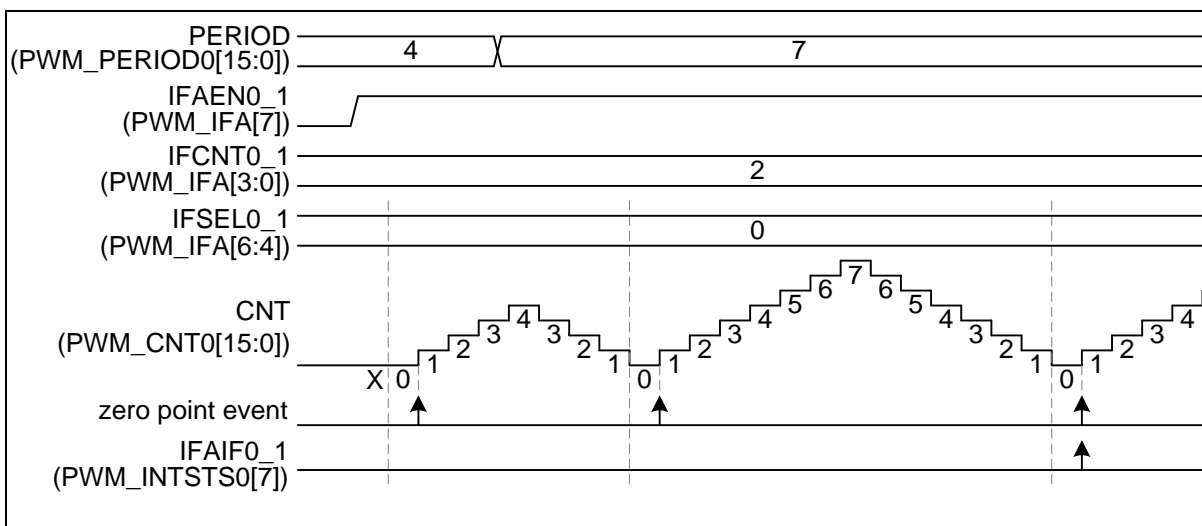


Figure 6.13-37 PWMx\_CH0 and PWMx\_CH1 Pair Accumulate Interrupt Waveform

The 2<sup>nd</sup> interrupt is the capture interrupt (CAP\_INT). It shares the PWM\_INT vector in NVIC. The CAP\_INT can be generated when the CRLIFn (PWM\_CAPIF[5:0]) flag is triggered and the Capture Rising Interrupt Enable bit CAPRIENn (PWM\_CAPIEN[5:0]) is set to 1. Or in the falling edge condition, the CFLIFn (PWM\_CAPIF[13:8]) flag can be triggered when the Capture Falling Interrupt Enable bit CAPFIENn (PWM\_CAPIEN[13:8]) is set to 1.

The last one is the brake interrupt (BRK\_INT). The details of the BRK\_INT is described in the PWM Brake section.

Figure 6.13-38 demonstrates the architecture of the PWM interrupts.

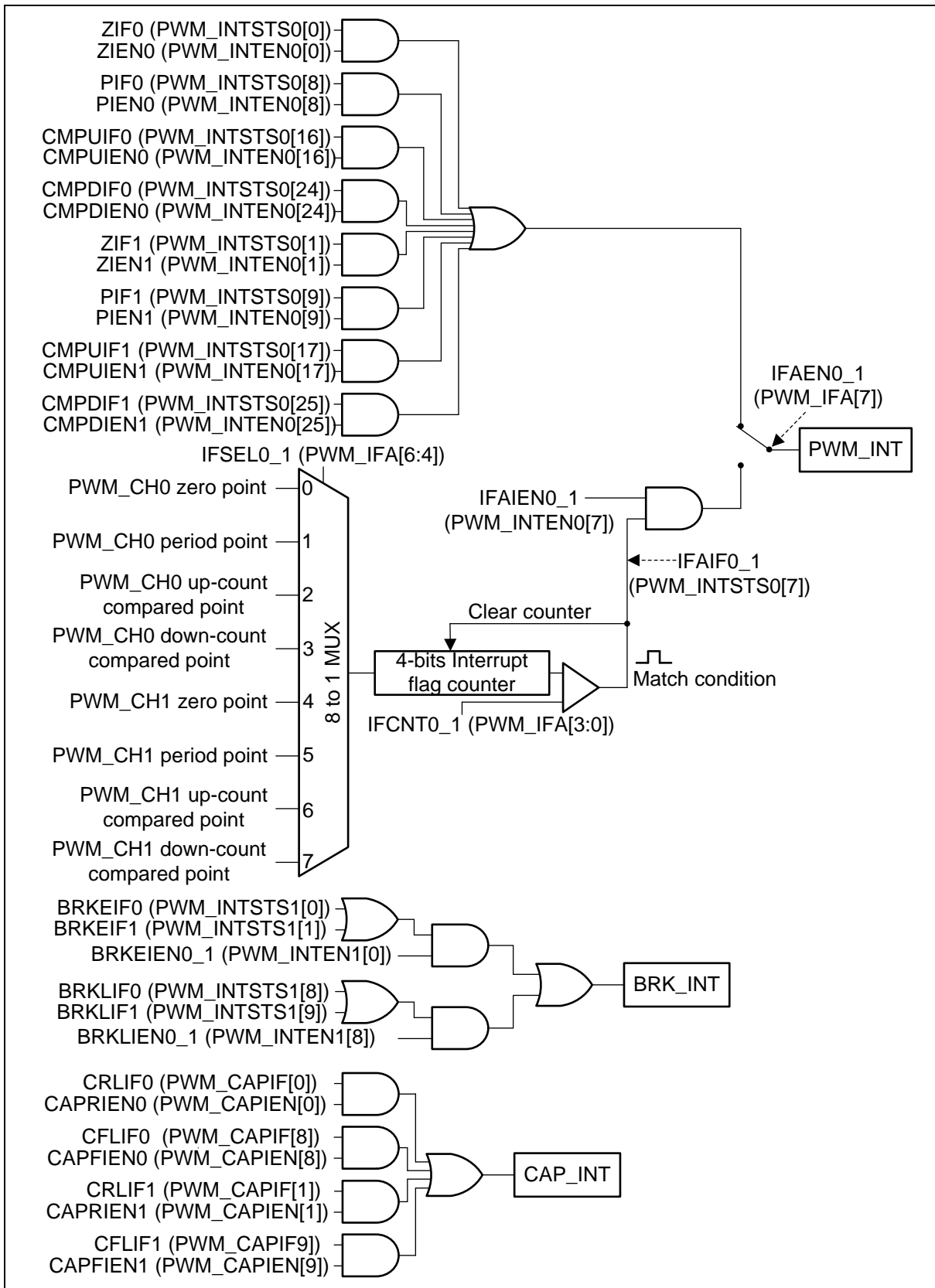


Figure 6.13-38 PWMx\_CH0 and PWMx\_CH1 Pair Interrupt Architecture Diagram



6.13.5.27 PWM Trigger ADC Generator

PWM can be one of the ADC conversion trigger source. Each PWM pair channels share the same trigger source. Setting TRGSELn bit of PWM\_ADCTS0 and PWM\_ADCTS1 registers is to select the trigger sources, where TRGSELn bit is TRGSEL0, TRGSEL1, ..., and TRGSEL5, which are located in PWM\_ADCTS0[3:0], PWM\_ADCTS0[11:8], PWM\_ADCTS0[19:16], PWM\_ADCTS0[27:24], PWM\_ADCTS1[3:0] and PWM\_ADCTS1[11:8], respectively. Setting TRGENn bit of PWM\_ADCTS0 and PWM\_ADCTS1 registers is to enable the trigger output to ADC, where TRGENn bit is TRGEN0, TRGEN1, ..., TRGEN5, which are located in PWM\_ADCTS0[7], PWM\_ADCTS0[15], PWM\_ADCTS0[23], PWM\_ADCTS0[31], PWM\_ADCTS1[7] and PWM\_ADCTS1[15], respectively. The number n (n = 0,1..5) denotes PWM channel number.

There are 16 PWM events can be selected as the trigger source for one pair of channels which shown in Figure 6.13-39. Figure 6.13-40 is the trigger ADC timing waveform in the up-down counter type.

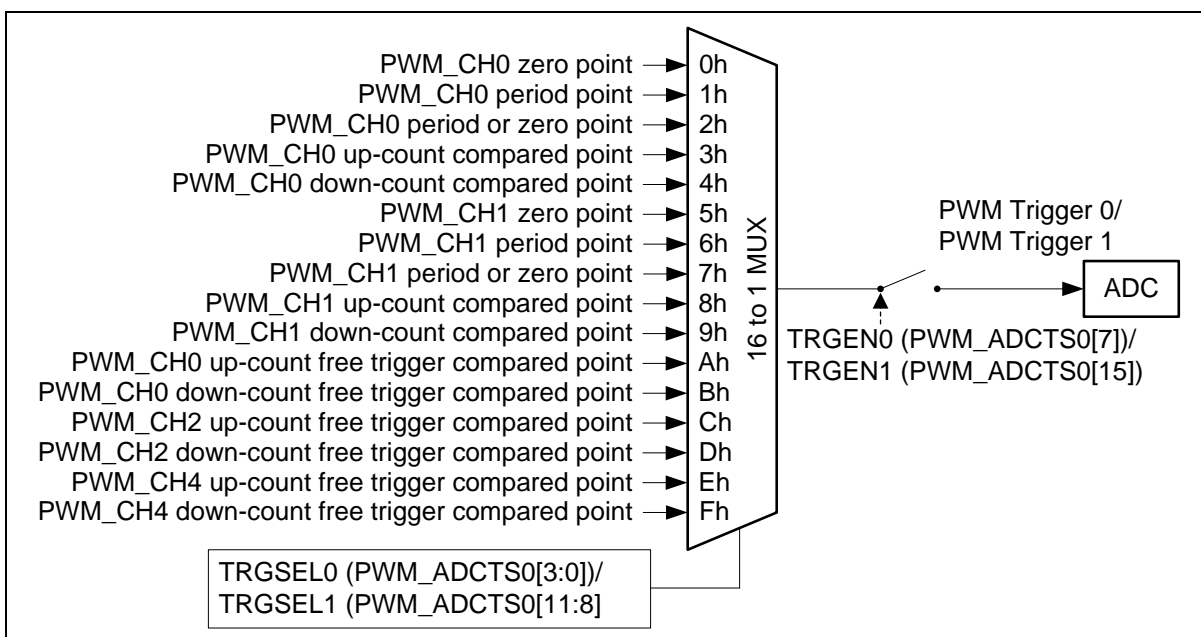


Figure 6.13-39 PWMx\_CH0 and PWMx\_CH1 Pair Trigger ADC Block Diagram

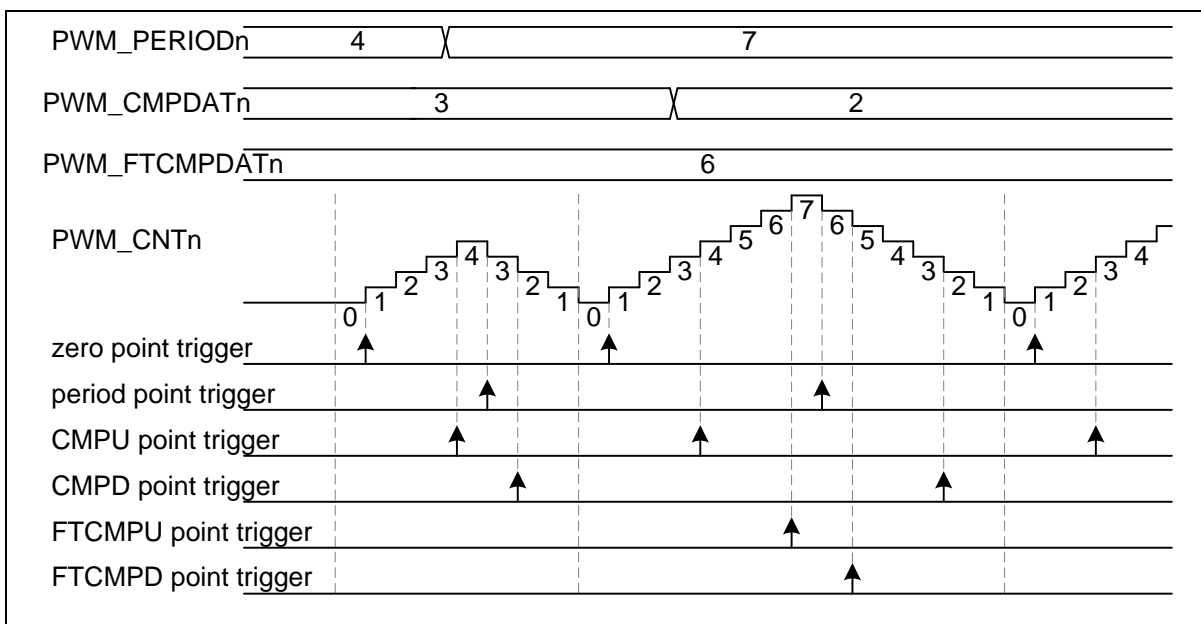


Figure 6.13-40 PWM Trigger ADC in Up-Down Counter Type Timing Waveform

### 6.13.5.28 Capture Operation

The channels of the capture input and the PWM output share the same pin and counter. The counter can operate in up or down counter type. The capture function will always latch the PWM counter to the RCAPDATn (PWM\_RCAPDATn[15:0]) bits or the FCAPDATn (PWM\_FCAPDATn[15:0]) bits, if the input channel has a rising transition or a falling transition, respectively. The capture function will also generate an interrupt CAP\_INT (using PWM\_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 (PWM\_CAPIEN[5:0]) bit is for the rising edge and the CAPFIENn (PWM\_CAPIEN[13:8]) bit is for the falling edge. When rising or falling latch occurs, the corresponding PWM counter may be reloaded with the value of PWM\_PERIODn register, depending on the setting of RCRLDENn or FCRLDENn bits (where RCRLDENn and FCRLDENn are located at PWM\_CAPCTL[21:16] and PWM\_CAPCTL[29:24], respectively). Note that the corresponding GPIO pins must be configured as the capture function by enable the CAPINENn (PWM\_CAPINEN[5:0]) bits for the corresponding capture channel n. Set GPIO mode as Quasi-bidirectional mode can assist these pins in input drive high by internal pull-up resistor, if there are no external pull-up resistor on these pins. Figure 6.13-41 is the capture block diagram of channel 0.



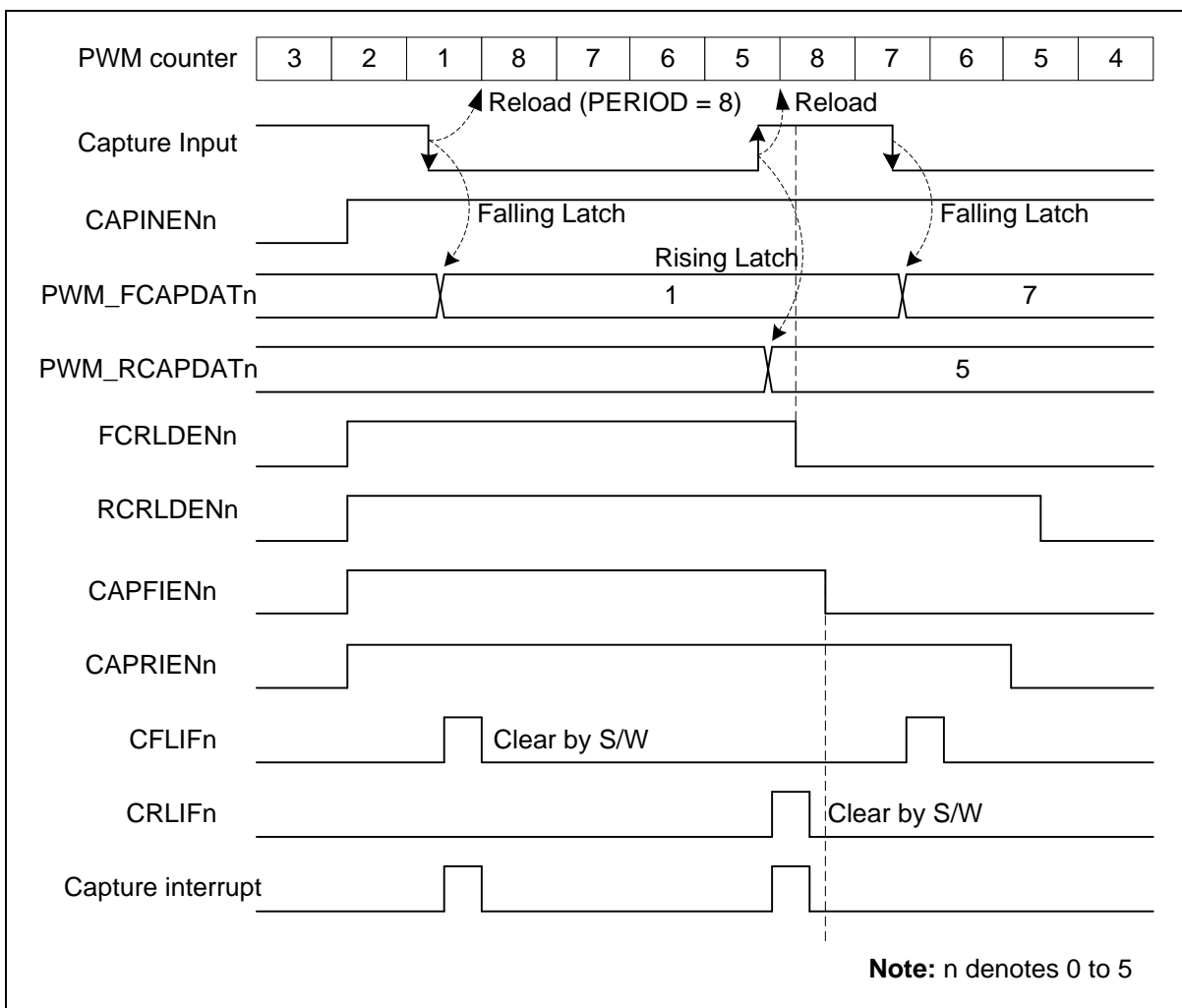


Figure 6.13-42 Capture Operation Waveform

The capture pulse width can be calculated according to the following formula:

For the negative pulse case, the channel low pulse width is calculated as  $(PWM\_PERIODn + 1 - PWM\_RCAPDATn)$ . In Figure 6.13-42, the low pulse width is  $8+1-5 = 4$

For the positive pulse case, the channel high pulse width is calculated as  $(PWM\_PERIODn + 1 - PWM\_FCAPDATn)$ . In Figure 6.13-42, the high pulse width is  $8+1-7 = 2$

#### 6.13.5.29 Capture PDMA Function

The PWM module supports the PDMA transfer function when operating in the capture mode. When the corresponding PDMA enable bit CHENn\_m (CHEN0\_1 at PWM\_PDMACTL[0], CHEN2\_3 at PWM\_PDMACTL[8] and CHEN4\_5 at PWM\_PDMACTL[16], where n and m denote complement pair channels) is set, the capture module will issue a request to PDMA controller when the preceding capture event has happened. The PDMA controller will issue an acknowledgement to the capture module after it has read back the CAPBUF (PWM\_PDMACAPn\_m[15:0], n, m denotes complement pair channels) register in the capture module and has sent the register value to the memory. By setting CAPMODn\_m (CAPMOD0\_1 at PWM\_PDMACTL[2:1], CAPMOD2\_3 at PWM\_PDMACTL[10:9] and CAPMOD4\_5 at PWM\_PDMACTL[18:17]) bits, the PDMA can transfer the rising edge captured data or falling edge captured data or both of them to the memory. When using the PDMA to transfer both of the falling and rising edge data, remember to set CAPORDn\_m (CAPORD0\_1 at PWM\_PDMACTL[3], CAPORD2\_3 at PWM\_PDMACTL[11] and CAPORD4\_5 at PWM\_PDMACTL[19]) bit to decide the

order of the transferred data (falling edge captured is first or rising edge captured first). The complement pair channels share a PDMA channel. Therefore, a selection bit CHSEL<sub>n\_m</sub> (CHSEL0\_1 (PWM\_PDMACTL[4]), CHSEL2\_3 (PWM\_PDMACTL[12]) and CHSEL4\_5 (PWM\_PDMACTL[20])) bit is used to decide either channel n or channel m can be serviced by the PDMA channel.

Figure 6.13-43 is capture PDMA waveform. In this case, the CHSEL0\_1 (PWM\_PDMACTL[4]) bit is set to 0. Hence the PDMA will service channel 0 for the capture data transfer. CAPMOD0\_1 (PWM\_PDMACTL[2:1]) bits are set to 3. That means both of the rising and falling edge captured data will be transferred to the memory. The CAPORD0\_1 (PWM\_PDMACTL[1]) bit is set to 1, so the rising edge data will be the first data to transfer and following is the falling edge data to transfer. As shown in Figure 6.13-43, the last assertions of the CRLIF0 and CFLIF0 signal have some overlap. The value of PWM\_RCAPDAT0 register is 11 will be loaded to PWM\_PDMACAP0\_1 register to wait for transfer but not the PWM\_FCAPDAT0 value. The PWM\_PDMACAP0\_1 register saves the data which will be transferred to the memory by PDMA. The HWDATA in this figure denotes the data which are being transferred by PDMA.

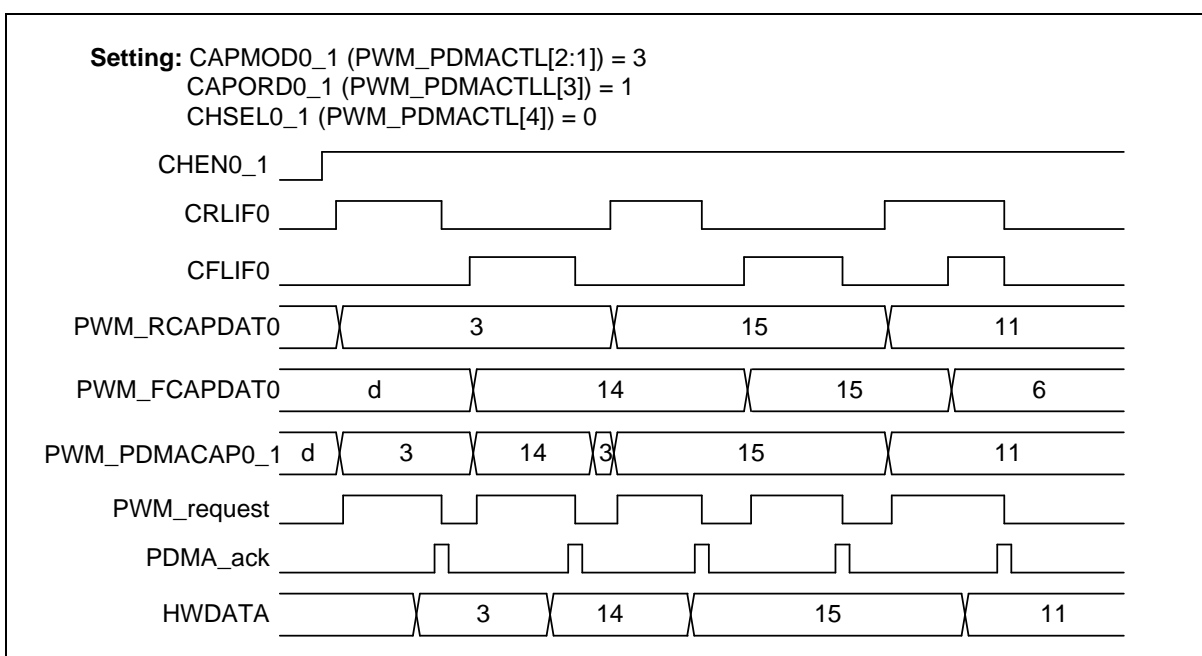


Figure 6.13-43 Capture PDMA Operation Waveform of Channel 0

### 6.13.6 Register Map

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

Register	Offset	R/W	Description	Reset Value
<b>PWM Base Address:</b>				
<b>PWMx_BA = 0x4004_0000 + (0x10_0000 * x)</b>				
x=0,1				
PWM_CTL0	PWMx_BA+0x00	R/W	PWM Control Register 0	0x0000_0000
PWM_CTL1	PWMx_BA+0x04	R/W	PWM Control Register 1	0x0000_0000
PWM_SYNC	PWMx_BA+0x08	R/W	PWM Synchronization Register	0x0000_0000
PWM_SWSYNC	PWMx_BA+0x0C	R/W	PWM Software Control Synchronization Register	0x0000_0000
PWM_CLKSRC	PWMx_BA+0x10	R/W	PWM Clock Source Register	0x0000_0000
PWM_CLKPSC0_1	PWMx_BA+0x14	R/W	PWM Clock Pre-scale Register 0/1	0x0000_0000
PWM_CLKPSC2_3	PWMx_BA+0x18	R/W	PWM Clock Pre-scale Register 2/3	0x0000_0000
PWM_CLKPSC4_5	PWMx_BA+0x1C	R/W	PWM Clock Pre-scale Register 4/5	0x0000_0000
PWM_CNTEN	PWMx_BA+0x20	R/W	PWM Counter Enable Register	0x0000_0000
PWM_CNTCLR	PWMx_BA+0x24	R/W	PWM Clear Counter Register	0x0000_0000
PWM_LOAD	PWMx_BA+0x28	R/W	PWM Load Register	0x0000_0000
PWM_PERIOD0	PWMx_BA+0x30	R/W	PWM Period Register 0	0x0000_0000
PWM_PERIOD1	PWMx_BA+0x34	R/W	PWM Period Register 1	0x0000_0000
PWM_PERIOD2	PWMx_BA+0x38	R/W	PWM Period Register 2	0x0000_0000
PWM_PERIOD3	PWMx_BA+0x3C	R/W	PWM Period Register 3	0x0000_0000
PWM_PERIOD4	PWMx_BA+0x40	R/W	PWM Period Register 4	0x0000_0000
PWM_PERIOD5	PWMx_BA+0x44	R/W	PWM Period Register 5	0x0000_0000
PWM_CMPDAT0	PWMx_BA+0x50	R/W	PWM Comparator Register 0	0x0000_0000
PWM_CMPDAT1	PWMx_BA+0x54	R/W	PWM Comparator Register 1	0x0000_0000
PWM_CMPDAT2	PWMx_BA+0x58	R/W	PWM Comparator Register 2	0x0000_0000
PWM_CMPDAT3	PWMx_BA+0x5C	R/W	PWM Comparator Register 3	0x0000_0000
PWM_CMPDAT4	PWMx_BA+0x60	R/W	PWM Comparator Register 4	0x0000_0000
PWM_CMPDAT5	PWMx_BA+0x64	R/W	PWM Comparator Register 5	0x0000_0000
PWM_DTCTL0_1	PWMx_BA+0x70	R/W	PWM Dead-Time Control Register 0/1	0x0000_0000
PWM_DTCTL2_3	PWMx_BA+0x74	R/W	PWM Dead-Time Control Register 2/3	0x0000_0000
PWM_DTCTL4_5	PWMx_BA+0x78	R/W	PWM Dead-Time Control Register 4/5	0x0000_0000

PWM_PHS0_1	PWMx_BA+0x80	R/W	PWM Counter Phase Register 0/1	0x0000_0000
PWM_PHS2_3	PWMx_BA+0x84	R/W	PWM Counter Phase Register 2/3	0x0000_0000
PWM_PHS4_5	PWMx_BA+0x88	R/W	PWM Counter Phase Register 4/5	0x0000_0000
PWM_CNT0	PWMx_BA+0x90	R	PWM Counter Register 0	0x0000_0000
PWM_CNT1	PWMx_BA+0x94	R	PWM Counter Register 1	0x0000_0000
PWM_CNT2	PWMx_BA+0x98	R	PWM Counter Register 2	0x0000_0000
PWM_CNT3	PWMx_BA+0x9C	R	PWM Counter Register 3	0x0000_0000
PWM_CNT4	PWMx_BA+0xA0	R	PWM Counter Register 4	0x0000_0000
PWM_CNT5	PWMx_BA+0xA4	R	PWM Counter Register 5	0x0000_0000
PWM_WGCTL0	PWMx_BA+0xB0	R/W	PWM Generation Register 0	0x0000_0000
PWM_WGCTL1	PWMx_BA+0xB4	R/W	PWM Generation Register 1	0x0000_0000
PWM_MSKEN	PWMx_BA+0xB8	R/W	PWM Mask Enable Register	0x0000_0000
PWM_MSK	PWMx_BA+0xBC	R/W	PWM Mask Data Register	0x0000_0000
PWM_BNF	PWMx_BA+0xC0	R/W	PWM Brake Noise Filter Register	0x0000_0000
PWM_FAILBRK	PWMx_BA+0xC4	R/W	PWM System Fail Brake Control Register	0x0000_0000
PWM_BRKCTL0_1	PWMx_BA+0xC8	R/W	PWM Brake Edge Detect Control Register 0/1	0x0000_0000
PWM_BRKCTL2_3	PWMx_BA+0xCC	R/W	PWM Brake Edge Detect Control Register 2/3	0x0000_0000
PWM_BRKCTL4_5	PWMx_BA+0xD0	R/W	PWM Brake Edge Detect Control Register 4/5	0x0000_0000
PWM_POLCTL	PWMx_BA+0xD4	R/W	PWM Pin Polar Inverse Register	0x0000_0000
PWM_POEN	PWMx_BA+0xD8	R/W	PWM Output Enable Register	0x0000_0000
PWM_SWBRK	PWMx_BA+0xDC	W	PWM Software Brake Control Register	0x0000_0000
PWM_INTEN0	PWMx_BA+0xE0	R/W	PWM Interrupt Enable Register 0	0x0000_0000
PWM_INTEN1	PWMx_BA+0xE4	R/W	PWM Interrupt Enable Register 1	0x0000_0000
PWM_INTSTS0	PWMx_BA+0xE8	R/W	PWM Interrupt Flag Register 0	0x0000_0000
PWM_INTSTS1	PWMx_BA+0xEC	R/W	PWM Interrupt Flag Register 1	0x0000_0000
PWM_IFA	PWMx_BA+0xF0	R/W	PWM Interrupt Flag Accumulator Register	0x0000_0000
PWM_ADCTS0	PWMx_BA+0xF8	R/W	PWM Trigger ADC Source Select Register 0	0x0000_0000
PWM_ADCTS1	PWMx_BA+0xFC	R/W	PWM Trigger ADC Source Select Register 1	0x0000_0000
PWM_FTCMPDAT0_1	PWMx_BA+0x100	R/W	PWM Free Trigger Compare Register 0/1	0x0000_0000
PWM_FTCMPDAT2_3	PWMx_BA+0x104	R/W	PWM Free Trigger Compare Register 2/3	0x0000_0000
PWM_FTCMPDAT4_5	PWMx_BA+0x108	R/W	PWM Free Trigger Compare Register 4/5	0x0000_0000

PWM_SSCTL	PWMx_BA+0x110	R/W	PWM Synchronous Start Control Register	0x0000_0000
PWM_SSTRG	PWMx_BA+0x114	W	PWM Synchronous Start Trigger Register	0x0000_0000
PWM_LEBCTL	PWMx_BA+0x118	R/W	PWM Leading Edge Blanking Control Register	0x0000_0000
PWM_LEBCNT	PWMx_BA+0x11C	R/W	PWM Leading Edge Blanking Counter Register	0x0000_0000
PWM_STATUS	PWMx_BA+0x120	R/W	PWM Status Register	0x0000_0000
PWM_CAPINEN	PWMx_BA+0x200	R/W	PWM Capture Input Enable Register	0x0000_0000
PWM_CAPCTL	PWMx_BA+0x204	R/W	PWM Capture Control Register	0x0000_0000
PWM_CAPSTS	PWMx_BA+0x208	R	PWM Capture Status Register	0x0000_0000
PWM_RCAPDAT0	PWMx_BA+0x20C	R	PWM Rising Capture Data Register 0	0x0000_0000
PWM_FCAPDAT0	PWMx_BA+0x210	R	PWM Falling Capture Data Register 0	0x0000_0000
PWM_RCAPDAT1	PWMx_BA+0x214	R	PWM Rising Capture Data Register 1	0x0000_0000
PWM_FCAPDAT1	PWMx_BA+0x218	R	PWM Falling Capture Data Register 1	0x0000_0000
PWM_RCAPDAT2	PWMx_BA+0x21C	R	PWM Rising Capture Data Register 2	0x0000_0000
PWM_FCAPDAT2	PWMx_BA+0x220	R	PWM Falling Capture Data Register 2	0x0000_0000
PWM_RCAPDAT3	PWMx_BA+0x224	R	PWM Rising Capture Data Register 3	0x0000_0000
PWM_FCAPDAT3	PWMx_BA+0x228	R	PWM Falling Capture Data Register 3	0x0000_0000
PWM_RCAPDAT4	PWMx_BA+0x22C	R	PWM Rising Capture Data Register 4	0x0000_0000
PWM_FCAPDAT4	PWMx_BA+0x230	R	PWM Falling Capture Data Register 4	0x0000_0000
PWM_RCAPDAT5	PWMx_BA+0x234	R	PWM Rising Capture Data Register 5	0x0000_0000
PWM_FCAPDAT5	PWMx_BA+0x238	R	PWM Falling Capture Data Register 5	0x0000_0000
PWM_PDMACTL	PWMx_BA+0x23C	R/W	PWM PDMA Control Register	0x0000_0000
PWM_PDMACAP0_1	PWMx_BA+0x240	R	PWM Capture Channel 0/1 PDMA Register	0x0000_0000
PWM_PDMACAP2_3	PWMx_BA+0x244	R	PWM Capture Channel 2/3 PDMA Register	0x0000_0000
PWM_PDMACAP4_5	PWMx_BA+0x248	R	PWM Capture Channel 4/5 PDMA Register	0x0000_0000
PWM_CAPIEN	PWMx_BA+0x250	R/W	PWM Capture Interrupt Enable Register	0x0000_0000
PWM_CAPIF	PWMx_BA+0x254	R/W	PWM Capture Interrupt Flag Register	0x0000_0000
PWM_PBUF0	PWMx_BA+0x304	R	PWM PERIOD0 Buffer	0x0000_0000
PWM_PBUF1	PWMx_BA+0x308	R	PWM PERIOD1 Buffer	0x0000_0000
PWM_PBUF2	PWMx_BA+0x30C	R	PWM PERIOD2 Buffer	0x0000_0000
PWM_PBUF3	PWMx_BA+0x310	R	PWM PERIOD3 Buffer	0x0000_0000
PWM_PBUF4	PWMx_BA+0x314	R	PWM PERIOD4 Buffer	0x0000_0000



<b>PWM_PBUF5</b>	PWMx_BA+0x318	R	PWM PERIOD5 Buffer	0x0000_0000
<b>PWM_CMPBUF0</b>	PWMx_BA+0x31C	R	PWM CMPDAT0 Buffer	0x0000_0000
<b>PWM_CMPBUF1</b>	PWMx_BA+0x320	R	PWM CMPDAT1 Buffer	0x0000_0000
<b>PWM_CMPBUF2</b>	PWMx_BA+0x324	R	PWM CMPDAT2 Buffer	0x0000_0000
<b>PWM_CMPBUF3</b>	PWMx_BA+0x328	R	PWM CMPDAT3 Buffer	0x0000_0000
<b>PWM_CMPBUF4</b>	PWMx_BA+0x32C	R	PWM CMPDAT4 Buffer	0x0000_0000
<b>PWM_CMPBUF5</b>	PWMx_BA+0x330	R	PWM CMPDAT5 Buffer	0x0000_0000
<b>PWM_CPSCBUF0_1</b>	PWMx_BA+0x334	R	PWM CLKPSC0_1 Buffer	0x0000_0000
<b>PWM_CPSCBUF2_3</b>	PWMx_BA+0x338	R	PWM CLKPSC2_3 Buffer	0x0000_0000
<b>PWM_CPSCBUF4_5</b>	PWMx_BA+0x33C	R	PWM CLKPSC4_5 Buffer	0x0000_0000
<b>PWM_FTCBUF0_1</b>	PWMx_BA+0x340	R	PWM FTCMPDAT0_1 Buffer	0x0000_0000
<b>PWM_FTCBUF2_3</b>	PWMx_BA+0x344	R	PWM FTCMPDAT2_3 Buffer	0x0000_0000
<b>PWM_FTCBUF4_5</b>	PWMx_BA+0x348	R	PWM FTCMPDAT4_5 Buffer	0x0000_0000
<b>PWM_FTCI</b>	PWMx_BA+0x34C	R/W	PWM FTCMPDAT Indicator Register	0x0000_0000

### 6.13.7 Register Description

#### PWM Control Register 0 (PWM\_CTL0)

Register	Offset	R/W	Description	Reset Value
PWM_CTL0	PWMx_BA+0x00	R/W	PWM Control Register 0	0x0000_0000

31	30	29	28	27	26	25	24
DBGTRIOFF	DBGHALT	Reserved					GROUPEN
23	22	21	20	19	18	17	16
Reserved		IMMLDEN5	IMMLDEN4	IMMLDEN3	IMMLDEN2	IMMLDEN1	IMMLDEN0
15	14	13	12	11	10	9	8
Reserved		WINLDEN5	WINLDEN4	WINLDEN3	WINLDEN2	WINLDEN1	WINLDEN0
7	6	5	4	3	2	1	0
Reserved		CTRLD5	CTRLD4	CTRLD3	CTRLD2	CTRLD1	CTRLD0

Bits	Description
[31]	<p><b>DBGTRIOFF</b></p> <p><b>ICE Debug Mode Acknowledge Disable (Write Protect)</b>                      0 = ICE debug mode acknowledgement effects PWM output.                      PWM pin will be forced as tri-state while ICE debug mode acknowledged.                      1 = ICE debug mode acknowledgement Disabled.                      PWM pin will keep output no matter ICE debug mode acknowledged or not.  <b>Note:</b> This bit is write protected. Refer to SYS_REGLCTL register.</p>
[30]	<p><b>DBGHALT</b></p> <p><b>ICE Debug Mode Counter Halt (Write Protect)</b>                      If counter halt is enabled, PWM 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.  <b>Note:</b> This bit is write protected. Refer to SYS_REGLCTL register.</p>
[29:26]	Reserved.
[24]	<p><b>GROUPEN</b></p> <p><b>Group Function Enable Bit</b>                      0 = The output waveform of each PWM channel are independent.                      1 = Unify the PWMx_CH2 and PWMx_CH4 to output the same waveform as PWMx_CH0 and unify the PWMx_CH3 and PWMx_CH5 to output the same waveform as PWMx_CH1.</p>
[23:22]	Reserved.
[n+16] n=0,1..5	<p><b>IMMLDENn</b></p> <p><b>Immediately Load Enable Bits</b>                      Each bit n controls the corresponding PWM channel n.                      0 = PERIODn register will load to PBUFn register at the end point of each period. CMPDATn register will load to CMPBUFn register at the end point or center point of each period by setting CTRLDN bit.                      1 = PERIODn/CMPDATn registers will load to PBUFn and CMPBUFn register immediately when software update PERIODn/CMPDATn register.  <b>Note:</b> If IMMLDENn bit is enabled, WINLDENn bit and CTRLDN bits will be invalid.</p>

[15:14]	<b>Reserved</b>	Reserved.
[n+8] n=0,1..5	<b>WINLDENn</b>	<p><b>Window Load Enable Bits</b></p> <p>Each bit n controls the corresponding PWM channel n.</p> <p>0 = PERIODn register will load to PBUFn register at the end point of each period. CMPDATn register will load to CMPBUFn register at the end point or center point of each period by setting CTRLDN bit.</p> <p>1 = PERIODn register will load to PBUFn and CMPDATn registers will load to CMPBUFn register at the end point of each period when valid reload window is set. The valid reload window is set by software write 1 to PWM_LOAD register, and cleared by hardware after load success.</p>
[7:6]	<b>Reserved</b>	Reserved.
[n] n=0,1..5	<b>CTRLDn</b>	<p><b>Center Re-load</b></p> <p>Each bit n controls the corresponding PWM channel n.</p> <p>In up-down counter type, PERIODn register will load to PBUFn register at the end point of each period. CMPDATn register will load to CMPBUFn register at the center point of a period.</p>

**PWM Control Register 1 (PWM\_CTL1)**

Register	Offset	R/W	Description	Reset Value
PWM_CTL1	PWMx_BA+0x04	R/W	PWM Control Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved					OUTMODE4	OUTMODE2	OUTMODE0
23	22	21	20	19	18	17	16
Reserved		CNTMODE5	CNTMODE4	CNTMODE3	CNTMODE2	CNTMODE1	CNTMODE0
15	14	13	12	11	10	9	8
Reserved				CNTTYPE5		CNTTYPE4	
7	6	5	4	3	2	1	0
CNTTYPE3		CNTTYPE2		CNTTYPE1		CNTTYPE0	

Bits	Description	
[31:27]	Reserved	Reserved.
[n/2+24] n=0,2,4	OUTMODEn	<p><b>PWM Output Mode</b> Each bit n controls the output mode of corresponding PWM channel n. 0 = PWM independent mode. 1 = PWM complementary mode. <b>Note:</b> When operating in group function, these bits must all set to the same mode.</p>
[23:22]	Reserved	Reserved.
[n+16] n=0,1..5	CNTMODEn	<p><b>PWM Counter Mode</b> Each bit n controls the corresponding PWM channel n. 0 = Auto-reload mode. 1 = One-shot mode.</p>
[15:12]	Reserved	Reserved.
[2n+1:2n] n=0,1..5	CNTTYPEn	<p><b>PWM Counter Behavior Type</b> Each bit n controls corresponding PWM 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.</p>

**PWM Synchronization Register (PWM\_SYNC)**

Register	Offset	R/W	Description	Reset Value
PWM_SYNC	PWMx_BA+0x08	R/W	PWM Synchronization Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved					PHSDIR4	PHSDIR2	PHSDIR0
23	22	21	20	19	18	17	16
SINPINV	SFLTCNT			SFLTCSEL			SNFLTEN
15	14	13	12	11	10	9	8
Reserved		SINSRC4		SINSRC2		SINSRC0	
7	6	5	4	3	2	1	0
Reserved					PHSEN4	PHSEN2	PHSEN0

Bits	Description
[31:27]	<b>Reserved</b> Reserved.
[n/2+24] n=0,2,4	<b>PHSDIRn</b> <b>PWM Phase Direction Control</b> Each bit n controls corresponding PWM channel n. 0 = Control PWM counter count decrement after synchronizing. 1 = Control PWM counter count increment after synchronizing.
[23]	<b>SINPINV</b> <b>SYNC Input Pin Inverse</b> 0 = The state of PWM0_SYNC_IN pin is passed to the negative edge detector. 1 = The inversed state of PWM0_SYNC_IN pin is passed to the negative edge detector.
[22:20]	<b>SFLTCNT</b> <b>SYNC Edge Detector Filter Count</b> The register bits control the counter number of edge detector.
[19:17]	<b>SFLTCSEL</b> <b>SYNC Edge Detector Filter Clock Selection</b> 000 = Filter clock = HCLK. 001 = Filter clock = HCLK/2. 010 = Filter clock = HCLK/4. 011 = Filter clock = HCLK/8. 100 = Filter clock = HCLK/16. 101 = Filter clock = HCLK/32. 110 = Filter clock = HCLK/64. 111 = Filter clock = HCLK/128.
[16]	<b>SNFLTEN</b> <b>PWM0_SYNC_IN Noise Filter Enable Bit</b> 0 = Noise filter of input PWM0_SYNC_IN pin Disabled. 1 = Noise filter of input PWM0_SYNC_IN pin Enabled.
[15:14]	<b>Reserved</b> Reserved.
[n+9:n+8] n=0,2,4	<b>SINSRCn</b> <b>PWM0_SYNC_IN Source Selection</b> Each bit n controls corresponding PWM channel n.

		00 = Synchronize source from SYNC_IN or SWSYNC. 01 = Counter equal to 0. 10 = Counter equal to PWM_CMPDATm, m denotes 1, 3, 5. 11 = SYNC_OUT signal will not be generated.
[7:3]	<b>Reserved</b>	Reserved.
[n/2] n=0,2,4	<b>PHSEn</b>	<b>SYNC Phase Enable Bits</b> n denotes PWM channel 0,2,4 and m denotes channel 1,3,5. 0 = PWM counter disable to load value of PHS(PWM_PHSn_m[15:0]) bits. 1 = PWM counter enable to load value of PHS(PWM_PHSn_m[15:0]) bits.

**PWM Software Control Synchronization Register (PWM\_SWSYNC)**

Register	Offset	R/W	Description	Reset Value
PWM_SWSYNC	PWMx_BA+0x0C	R/W	PWM Software Control Synchronization 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					SWSYNC4	SWSYNC2	SWSYNC0

Bits	Description	
[31:3]	Reserved	Reserved.
[n/2] n=0,2,4	SWSYNCn	<b>Software SYNC Function</b> Each bit n controls corresponding PWM channel n. When SINSRCn (PWM_SYNC[13:8]) is selected to 0, SYNC_OUT source is come from SYNC_IN or this bit.

**PWM Clock Source Register (PWM\_CLKSRC)**

Register	Offset	R/W	Description	Reset Value
PWM_CLKSRC	PWMx_BA+0x10	R/W	PWM Clock Source Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved				ECLKSRC4			
15	14	13	12	11	10	9	8
Reserved				ECLKSRC2			
7	6	5	4	3	2	1	0
Reserved				ECLKSRC0			

Bits	Description	
[31:19]	Reserved	Reserved.
[18:16]	ECLKSRC4	<b>PWMx_CH4/5 External Clock Source Select</b> 000 = PWMx_CLK, x denotes 0 or 1. 001 = TIMER0 time-out event. 010 = TIMER1 time-out event. 011 = TIMER2 time-out event. 100 = TIMER3 time-out event. Others = Reserved.
[15:11]	Reserved	Reserved.
[10:8]	ECLKSRC2	<b>PWMx_CH2/3 External Clock Source Select</b> 000 = PWMx_CLK, x denotes 0 or 1. 001 = TIMER0 time-out event. 010 = TIMER1 time-out event. 011 = TIMER2 time-out event. 100 = TIMER3 time-out event. Others = Reserved.
[7:3]	Reserved	Reserved.
[2:0]	ECLKSRC0	<b>PWMx_CH0/1 External Clock Source Select</b> 000 = PWMx_CLK, x denotes 0 or 1. 001 = TIMER0 time-out event. 010 = TIMER1 time-out event. 011 = TIMER2 time-out event. 100 = TIMER3 time-out event. Others = Reserved.



**PWM Clock Pre-Scale Register 0 1, 2 3, 4 5 (PWM\_CLKPSC0 1, 2 3, 4 5)**

Register	Offset	R/W	Description	Reset Value
PWM_CLKPSC0_1	PWMx_BA+0x14	R/W	PWM Clock Pre-scale Register 0/1	0x0000_0000
PWM_CLKPSC2_3	PWMx_BA+0x18	R/W	PWM Clock Pre-scale Register 2/3	0x0000_0000
PWM_CLKPSC4_5	PWMx_BA+0x1C	R/W	PWM Clock Pre-scale Register 4/5	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
Reserved				CLKPSC			
7	6	5	4	3	2	1	0
CLKPSC							

Bits	Description	
[31:12]	Reserved	Reserved.
[11:0]	CLKPSC	<b>PWM Counter Clock Pre-scale</b> The clock of PWM counter is decided by clock prescaler. Each PWM pair share one PWM counter clock prescaler. The clock of PWM counter is divided by (CLKPSC+ 1).

**PWM Counter Enable Register (PWM\_CNTEN)**

Register	Offset	R/W	Description	Reset Value
PWM_CNTEN	PWMx_BA+0x20	R/W	PWM 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		CNTEN5	CNTEN4	CNTEN3	CNTEN2	CNTEN1	CNTEN0

Bits	Description	
[31:6]	Reserved	Reserved.
[n] n=0,1..5	CNTENn	<b>PWM Counter Enable Bits</b> Each bit n controls the corresponding PWM channel n. 0 = PWM Counter and clock prescaler Stop Running. 1 = PWM Counter and clock prescaler Start Running.

**PWM Clear Counter Register (PWM\_CNTCLR)**

Register	Offset	R/W	Description	Reset Value
PWM_CNTCLR	PWMx_BA+0x24	R/W	PWM 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		CNTCLR5	CNTCLR4	CNTCLR3	CNTCLR2	CNTCLR1	CNTCLR0

Bits	Description	
[31:6]	Reserved	Reserved.
[n] n=0,1..5	CNTCLRn	<p><b>Clear PWM Counter Control Bit</b></p> <p>It is automatically cleared by hardware. Each bit n controls the corresponding PWM channel n.</p> <p>0 = No effect.</p> <p>1 = Clear 16-bit PWM counter to 0x0000(CNT(PWM_CNTn[15:0])).</p>

**PWM Load Register (PWM\_LOAD)**

Register	Offset	R/W	Description	Reset Value
PWM_LOAD	PWMx_BA+0x28	R/W	PWM Load 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		LOAD5	LOAD4	LOAD3	LOAD2	LOAD1	LOAD0

Bits	Description	
[31:6]	Reserved	Reserved.
[n] n=0,1..5	LOADn	<p><b>Re-load PWM Comparator Register (CMPDAT) Control Bit</b> This bit is software write, and hardware clear when current PWM period end. Each bit n controls the corresponding PWM channel n.</p> <p><b>Write Operation:</b> 0 = No effect. 1 = Set load window of window loading mode.</p> <p><b>Read Operation:</b> 0 = No load window is set. 1 = Load window is set.</p> <p><b>Note:</b> This bit is only used in window loading mode; WINLDENn(PWM_CTL0[13:8]) = 1.</p>

**PWM Period Register 0~5 (PWM\_PERIOD0~5)**

Register	Offset	R/W	Description	Reset Value
PWM_PERIOD0	PWMx_BA+0x30	R/W	PWM Period Register 0	0x0000_0000
PWM_PERIOD1	PWMx_BA+0x34	R/W	PWM Period Register 1	0x0000_0000
PWM_PERIOD2	PWMx_BA+0x38	R/W	PWM Period Register 2	0x0000_0000
PWM_PERIOD3	PWMx_BA+0x3C	R/W	PWM Period Register 3	0x0000_0000
PWM_PERIOD4	PWMx_BA+0x40	R/W	PWM Period Register 4	0x0000_0000
PWM_PERIOD5	PWMx_BA+0x44	R/W	PWM Period 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
PERIOD							
7	6	5	4	3	2	1	0
PERIOD							

Bits	Description
[31:16]	<b>Reserved</b> Reserved.
[15:0]	<p><b>PERIOD</b></p> <p><b>PWM Period Register</b> Up-Count mode: In this mode, PWM counter counts from 0 to PERIOD, and restarts from 0. PWM period time = (PERIOD+1) * (CLKPSC+1) * PWMx_CLK.</p> <p><b>Down-Count mode:</b> In this mode, PWM counter counts from PERIOD to 0, and restarts from PERIOD. PWM period time = (PERIOD+1) * (CLKPSC+1) * PWMx_CLK.</p> <p><b>Up-Down-Count mode:</b> In this mode, PWM counter counts from 0 to PERIOD, then decrements to 0 and repeats again. PWM period time = (2 * PERIOD) * (CLKPSC+1) * PWMx_CLK.</p>

**PWM Comparator Register 0~5 (PWM\_CMPDAT0~5)**

Register	Offset	R/W	Description	Reset Value
PWM_CMPDAT0	PWMx_BA+0x50	R/W	PWM Comparator Register 0	0x0000_0000
PWM_CMPDAT1	PWMx_BA+0x54	R/W	PWM Comparator Register 1	0x0000_0000
PWM_CMPDAT2	PWMx_BA+0x58	R/W	PWM Comparator Register 2	0x0000_0000
PWM_CMPDAT3	PWMx_BA+0x5C	R/W	PWM Comparator Register 3	0x0000_0000
PWM_CMPDAT4	PWMx_BA+0x60	R/W	PWM Comparator Register 4	0x0000_0000
PWM_CMPDAT5	PWMx_BA+0x64	R/W	PWM 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
CMP							
7	6	5	4	3	2	1	0
CMP							

Bits	Description
[31:16]	<b>Reserved</b> Reserved.
[15:0]	<b>CMP</b> <b>PWM Comparator Register</b> CMP bits use to compare with CNT(PWM_CNTn[15:0]) bits to generate PWM waveform, interrupt and trigger ADC. In independent mode, CMPDAT0~5 registers denote as 6 independent PWMx_CH0~5 compared point. In complementary mode, CMPDAT0, 2, 4 registers denote as first compared point, and CMPDAT1, 3, 5 register denote as second compared point for the corresponding 3 complementary pairs PWMx_CH0 and PWMx_CH1, PWMx_CH2 and PWMx_CH3, PWMx_CH4 and PWMx_CH5.

**PWM Dead-Time Control Register 0 1, 2 3, 4 5 (PWM\_DTCTL0 1, 2 3, 4 5)**

Register	Offset	R/W	Description	Reset Value
PWM_DTCTL0_1	PWMx_BA+0x70	R/W	PWM Dead-Time Control Register 0/1	0x0000_0000
PWM_DTCTL2_3	PWMx_BA+0x74	R/W	PWM Dead-Time Control Register 2/3	0x0000_0000
PWM_DTCTL4_5	PWMx_BA+0x78	R/W	PWM Dead-Time Control Register 4/5	0x0000_0000

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

Bits	Description	Description
[31:25]	Reserved	Reserved.
[24]	DTCKSEL	<b>Dead-time Clock Select (Write Protect)</b> 0 = Dead-time clock source from PWMx_CLK without counter clock prescale. 1 = Dead-time clock source from prescaler output with counter clock prescale. <b>Note:</b> This register is write protected. Refer to SYS_REGLCTL register.
[23:17]	Reserved	Reserved.
[16]	DTEN	<b>Enable Dead-time Insertion for PWM Pair (PWMx_CH0, PWMx_CH1) (PWMx_CH2, PWMx_CH3) (PWMx_CH4, PWMx_CH5) (Write Protect)</b> Dead-time insertion is only active when this pair of complementary PWM is enabled. If dead-time insertion is inactive, the outputs of pin pair are complementary without any delay. 0 = Dead-time insertion Disabled on the pin pair. 1 = Dead-time insertion Enabled on the pin pair. <b>Note:</b> This register is write protected. Refer to SYS_REGLCTL register.
[15:12]	Reserved	Reserved.
[11:0]	DTCNT	<b>Dead-time Counter (Write Protect)</b> The dead-time can be calculated from the following formula: Dead-time = (DTCNT+1) * PWMx_CLK period, if DTCKSEL bit is 0. Dead-time = (DTCNT+1) * (CLKPSC (PWM_CLKPSCn [11:0])+1)*PWMx_CLK period, if DTCKSEL bit is 1. <b>Note:</b> This register is write protected. Refer to SYS_REGLCTL register.

**PWM Counter Phase Register 0 1, 2 3, 4 5 (PWM\_PHS0 1, 2 3, 4 5)**

Register	Offset	R/W	Description	Reset Value
PWM_PHS0_1	PWMx_BA+0x80	R/W	PWM Counter Phase Register 0/1	0x0000_0000
PWM_PHS2_3	PWMx_BA+0x84	R/W	PWM Counter Phase Register 2/3	0x0000_0000
PWM_PHS4_5	PWMx_BA+0x88	R/W	PWM Counter Phase Register 4/5	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
PHS							
7	6	5	4	3	2	1	0
PHS							

Bits	Description	
[31:16]	Reserved	Reserved.
[15:0]	PHS	<b>PWM Synchronous Start Phase Bits</b> PHS bits determines the PWM synchronous start phase value. These bits only use in synchronous function.



**PWM Counter Register 0~5 (PWM\_CNT0~5)**

Register	Offset	R/W	Description	Reset Value
PWM_CNT0	PWMx_BA+0x90	R	PWM Counter Register 0	0x0000_0000
PWM_CNT1	PWMx_BA+0x94	R	PWM Counter Register 1	0x0000_0000
PWM_CNT2	PWMx_BA+0x98	R	PWM Counter Register 2	0x0000_0000
PWM_CNT3	PWMx_BA+0x9C	R	PWM Counter Register 3	0x0000_0000
PWM_CNT4	PWMx_BA+0xA0	R	PWM Counter Register 4	0x0000_0000
PWM_CNT5	PWMx_BA+0xA4	R	PWM Counter Register 5	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	<b>PWM Direction Indicator Flag (Read Only)</b> 0 = Counter is Down count. 1 = Counter is UP count.
[15:0]	CNT	<b>PWM Counter Data Bits (Read Only)</b> User can monitor CNT to know the current value in 16-bit period counter.

**PWM Generation Register 0 (PWM\_WGCTL0)**

Register	Offset	R/W	Description	Reset Value
PWM_WGCTL0	PWMx_BA+0xB0	R/W	PWM 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.
[2n+17:2n+16] n=0,1..5	PRDPCTLn	<p><b>PWM Period (Center) Point Control</b></p> <p>PWM can control output level on period(center) point event. Each bit n controls the corresponding PWM channel n.</p> <p>00 = Do nothing.                      01 = PWM period (center) point output Low.                      10 = PWM period (center) point output High.                      11 = PWM period (center) point output Toggle.</p> <p><b>Note:</b> This bit is center point control when PWM counter operating in up-down counter type.</p>
[15:12]	Reserved	Reserved.
[2n+1:2n] n=0,1..5	ZPCTLn	<p><b>PWM Zero Point Control</b></p> <p>PWM can control output level on zero point event. Each bit n controls the corresponding PWM channel n.</p> <p>00 = Do nothing.                      01 = PWM zero point output Low.                      10 = PWM zero point output High.                      11 = PWM zero point output Toggle.</p>

**PWM Generation Register 1 (PWM\_WGCTL1)**

Register	Offset	R/W	Description	Reset Value
PWM_WGCTL1	PWMx_BA+0xB4	R/W	PWM 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.
[2n+17:2n+16] n=0,1..5	CMPDCTLn	<p><b>PWM Compare Down Point Control</b></p> <p>PWM can control output level on compare down point event. Each bit n controls the corresponding PWM channel n.</p> <p>00 = Do nothing. 01 = PWM compare down point output Low. 10 = PWM compare down point output High. 11 = PWM compare down point output Toggle.</p> <p><b>Note:</b> In complementary mode, CMPDCTL1, 3, 5 is used as another CMPDCTL for channel 0, 2, 4.</p>
[15:12]	Reserved	Reserved.
[2n+1:2n] n=0,1..5	CMPUCTLn	<p><b>PWM Compare Up Point Control</b></p> <p>PWM can control output level on compare up point event. Each bit n controls the corresponding PWM channel n.</p> <p>00 = Do nothing. 01 = PWM compare up point output Low. 10 = PWM compare up point output High. 11 = PWM compare up point output Toggle.</p> <p><b>Note:</b> In complementary mode, CMPUCTL1, 3, 5 is used as another CMPUCTL for channel 0, 2, 4.</p>

**PWM Mask Enable Register (PWM\_MSKEN)**

Register	Offset	R/W	Description	Reset Value
PWM_MSKEN	PWMx_BA+0xB8	R/W	PWM 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	<p><b>PWM Mask Enable Bits</b></p> <p>The PWM output signal will be masked when this bit is enabled. The corresponding PWM channel n will output MSKDATn (PWM_MSK[5:0]) data. Each bit n controls the corresponding PWM channel n.</p> <p>0 = PWM output signal is non-masked.</p> <p>1 = PWM output signal is masked and output MSKDATn data.</p>

**PWM Mask DATA Register (PWM\_MSK)**

Register	Offset	R/W	Description	Reset Value
PWM_MSK	PWMx_BA+0xBC	R/W	PWM 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	<p><b>PWM Mask Data Bit</b></p> <p>This data bit control the state of PWMx_CHn output pin, if corresponding mask function is enabled. Each bit n controls the corresponding PWM channel n.</p> <p>0 = Output logic low to PWMx_CHn.</p> <p>1 = Output logic high to PWMx_CHn.</p>

**PWM Brake Noise Filter Register (PWM\_BNF)**

Register	Offset	R/W	Description	Reset Value
PWM_BNF	PWMx_BA+0xC0	R/W	PWM Brake Noise Filter Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							BK1SRC
23	22	21	20	19	18	17	16
Reserved							BK0SRC
15	14	13	12	11	10	9	8
BRK1PINV	BRK1FCNT			BRK1NFSEL			BRK1NFEN
7	6	5	4	3	2	1	0
BRK0PINV	BRK0FCNT			BRK0NFSEL			BRK0NFEN

Bits	Description	Description
[31:25]	Reserved	Reserved.
[24]	BK1SRC	<p><b>Brake 1 Pin Source Select</b>                      For PWM0 setting:                      0 = Brake 1 pin source come from PWM0_BRAKE1.                      1 = Brake 1 pin source come from PWM1_BRAKE1.                      For PWM1 setting:                      0 = Brake 1 pin source come from PWM1_BRAKE1.                      1 = Brake 1 pin source come from PWM0_BRAKE1.</p>
[23:17]	Reserved	Reserved.
[16]	BK0SRC	<p><b>Brake 0 Pin Source Select</b>                      For PWM0 setting:                      0 = Brake 0 pin source come from PWM0_BRAKE0.                      1 = Brake 0 pin source come from PWM1_BRAKE0.                      For PWM1 setting:                      0 = Brake 0 pin source come from PWM1_BRAKE0.                      1 = Brake 0 pin source come from PWM0_BRAKE0.</p>
[15]	BRK1PINV	<p><b>Brake 1 Pin Inverse</b>                      0 = Brake pin event will be detected if PWM1_BRAKE<sub>x</sub> pin status transfer from low to high in edge-detect, or pin status is high in level-detect.                      1 = Brake pin event will be detected if PWM1_BRAKE<sub>x</sub> pin status transfer from high to low in edge-detect, or pin status is low in level-detect.</p>
[14:12]	BRK1FCNT	<p><b>Brake 1 Edge Detector Filter Count</b>                      The register bits control the Brake1 filter counter to count from 0 to BRK1FCNT.</p>
[11:9]	BRK1NFSEL	<p><b>Brake 1 Edge Detector Filter Clock Selection</b>                      000 = Filter clock = HCLK.                      001 = Filter clock = HCLK/2.</p>

		<p>010 = Filter clock = HCLK/4.                  011 = Filter clock = HCLK/8.                  100 = Filter clock = HCLK/16.                  101 = Filter clock = HCLK/32.                  110 = Filter clock = HCLK/64.                  111 = Filter clock = HCLK/128.</p>
[8]	<b>BRK1NFEN</b>	<p><b>PWM Brake 1 Noise Filter Enable Bit</b>                  0 = Noise filter of PWM Brake 1 Disabled.                  1 = Noise filter of PWM Brake 1 Enabled.</p>
[7]	<b>BRK0PINV</b>	<p><b>Brake 0 Pin Inverse</b>                  0 = Brake pin event will be detected if PWM0_BRAKE<sub>x</sub> pin status transfer from low to high in edge-detect, or pin status is high in level-detect.                  1 = Brake pin event will be detected if PWM0_BRAKE<sub>x</sub> pin status transfer from high to low in edge-detect, or pin status is low in level-detect.</p>
[6:4]	<b>BRK0FCNT</b>	<p><b>Brake 0 Edge Detector Filter Count</b>                  The register bits control the Brake0 filter counter to count from 0 to BRK0FCNT.</p>
[3:1]	<b>BRK0NFSEL</b>	<p><b>Brake 0 Edge Detector Filter Clock Selection</b>                  000 = Filter clock = HCLK.                  001 = Filter clock = HCLK/2.                  010 = Filter clock = HCLK/4.                  011 = Filter clock = HCLK/8.                  100 = Filter clock = HCLK/16.                  101 = Filter clock = HCLK/32.                  110 = Filter clock = HCLK/64.                  111 = Filter clock = HCLK/128.</p>
[0]	<b>BRK0NFEN</b>	<p><b>PWM Brake 0 Noise Filter Enable Bit</b>                  0 = Noise filter of PWM Brake 0 Disabled.                  1 = Noise filter of PWM Brake 0 Enabled.</p>

**PWM System Fail Brake Control Register (PWM\_FAILBRK)**

Register	Offset	R/W	Description	Reset Value
PWM_FAILBRK	PWMx_BA+0xC4	R/W	PWM System Fail Brake 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				CORBRKEN	Reserved	BODBRKEN	CSSBRKEN

Bits	Description	
[31:4]	Reserved	Reserved.
[3]	CORBRKEN	<b>Core Lockup Detection Trigger PWM Brake Function Enable Bit</b> 0 = Brake Function triggered by Core lockup event Disabled. 1 = Brake Function triggered by Core lockup event Enabled.
[2]	Reserved	Reserved.
[1]	BODBRKEN	<b>Brown-out Detection Trigger PWM Brake Function Enable Bit</b> 0 = Brake Function triggered by BOD event Disabled. 1 = Brake Function triggered by BOD event Enabled.
[0]	CSSBRKEN	<b>Clock Security System Detection Trigger PWM Brake Function Enable Bit</b> 0 = Brake Function triggered by clock fail detection Disabled. 1 = Brake Function triggered by clock fail detection Enabled.



**PWM Brake Edge Detect Control Register 0 1, 2 3, 4 5 (PWM\_BRKCTL0 1, 2 3, 4 5)**

Register	Offset	R/W	Description	Reset Value
PWM_BRKCTL0_1	PWMx_BA+0xC8	R/W	PWM Brake Edge Detect Control Register 0/1	0x0000_0000
PWM_BRKCTL2_3	PWMx_BA+0xCC	R/W	PWM Brake Edge Detect Control Register 2/3	0x0000_0000
PWM_BRKCTL4_5	PWMx_BA+0xD0	R/W	PWM Brake Edge Detect Control Register 4/5	0x0000_0000

31	30	29	28	27	26	25	24
Reserved			ADCLBEN	Reserved			
23	22	21	20	19	18	17	16
Reserved			ADCEBEN	BRKAODD		BRKAEVEN	
15	14	13	12	11	10	9	8
SYSLBEN	Reserved	BRKP1LEN	BRKP0LEN	Reserved		CPO1LBEN	CPO0LBEN
7	6	5	4	3	2	1	0
SYSEBEN	Reserved	BRKP1EEN	BRKP0EEN	Reserved		CPO1EBEN	CPO0EBEN

Bits	Description	
[31:29]	Reserved	Reserved.
[28]	ADCLBEN	<p><b>Enable ADC Result Monitor (ADCRM) As Level-detect Brake Source (Write Protect)</b></p> <p>0 = ADCRM as level-detect brake source Disabled.                      1 = ADCRM as level-detect brake source Enabled.</p> <p><b>Note:</b> This register is write protected. Refer to SYS_REGLCTL register.</p>
[27:21]	Reserved	Reserved.
[20]	ADCEBEN	<p><b>Enable ADC Result Monitor (ADCRM) As Edge-detect Brake Source (Write Protect)</b></p> <p>0 = ADCRM as edge-detect brake source Disabled.                      1 = ADCRM as edge-detect brake source Enabled.</p> <p><b>Note:</b> This register is write protected. Refer to SYS_REGLCTL register.</p>
[19:18]	BRKAODD	<p><b>PWM Brake Action Select for Odd Channel (Write Protect)</b></p> <p>00 = PWMx brake event will not affect odd channels output.                      01 = PWM odd channel output tri-state when PWMx brake event happened.                      10 = PWM odd channel output low level when PWMx brake event happened.                      11 = PWM odd channel output high level when PWMx brake event happened.</p> <p><b>Note:</b> This register is write protected. Refer to SYS_REGLCTL register.</p>
[17:16]	BRKAEVEN	<p><b>PWM Brake Action Select for Even Channel (Write Protect)</b></p> <p>00 = PWMx brake event will not affect even channels output.                      01 = PWM even channel output tri-state when PWMx brake event happened.                      10 = PWM even channel output low level when PWMx brake event happened.</p>

		11 = PWM even channel output high level when PWMx brake event happened. <b>Note:</b> This register is write protected. Refer to SYS_REGLCTL register.
[15]	<b>SYSLBEN</b>	<b>Enable System Fail As Level-detect Brake Source (Write Protect)</b> 0 = System Fail condition as level-detect brake source Disabled. 1 = System Fail condition as level-detect brake source Enabled. <b>Note:</b> This register is write protected. Refer to SYS_REGLCTL register.
[14]	<b>Reserved</b>	Reserved.
[13]	<b>BRKP1LEN</b>	<b>Enable BKP1 Pin As Level-detect Brake Source (Write Protect)</b> 0 = PWMx_BRAKE1 pin as level-detect brake source Disabled. 1 = PWMx_BRAKE1 pin as level-detect brake source Enabled. <b>Note:</b> This register is write protected. Refer to SYS_REGLCTL register.
[12]	<b>BRKP0LEN</b>	<b>Enable BKP0 Pin As Level-detect Brake Source (Write Protect)</b> 0 = PWMx_BRAKE0 pin as level-detect brake source Disabled. 1 = PWMx_BRAKE0 pin as level-detect brake source Enabled. <b>Note:</b> This register is write protected. Refer to SYS_REGLCTL register.
[11:10]	<b>Reserved</b>	Reserved.
[9]	<b>CPO1LBEN</b>	<b>Enable ACMP1_O Digital Output As Level-detect Brake Source (Write Protect)</b> 0 = ACMP1_O as level-detect brake source Disabled. 1 = ACMP1_O as level-detect brake source Enabled. <b>Note:</b> This register is write protected. Refer to SYS_REGLCTL register.
[8]	<b>CPO0LBEN</b>	<b>Enable ACMP0_O Digital Output As Level-detect Brake Source (Write Protect)</b> 0 = ACMP0_O as level-detect brake source Disabled. 1 = ACMP0_O as level-detect brake source Enabled. <b>Note:</b> This register is write protected. Refer to SYS_REGLCTL register.
[7]	<b>SYSEBEN</b>	<b>Enable System Fail As Edge-detect Brake Source (Write Protect)</b> 0 = System Fail condition as edge-detect brake source Disabled. 1 = System Fail condition as edge-detect brake source Enabled. <b>Note:</b> This register is write protected. Refer to SYS_REGLCTL register.
[6]	<b>Reserved</b>	Reserved.
[5]	<b>BRKP1EEN</b>	<b>Enable PWMx_BRAKE1 Pin As Edge-detect Brake Source (Write Protect)</b> 0 = PWMx_BRAKE1 pin as edge-detect brake source Disabled. 1 = PWMx_BRAKE1 pin as edge-detect brake source Enabled. <b>Note:</b> This register is write protected. Refer to SYS_REGLCTL register.
[4]	<b>BRKP0EEN</b>	<b>Enable PWMx_BRAKE0 Pin As Edge-detect Brake Source (Write Protect)</b> 0 = PWMx_BRAKE0 pin as edge-detect brake source Disabled. 1 = PWMx_BRAKE0 pin as edge-detect brake source Enabled. <b>Note:</b> This register is write protected. Refer to SYS_REGLCTL register.
[3:2]	<b>Reserved</b>	Reserved.
[1]	<b>CPO1EBEN</b>	<b>Enable ACMP1_O Digital Output As Edge-detect Brake Source (Write Protect)</b> 0 = ACMP1_O as edge-detect brake source Disabled. 1 = ACMP1_O as edge-detect brake source Enabled. <b>Note:</b> This register is write protected. Refer to SYS_REGLCTL register.
[0]	<b>CPO0EBEN</b>	<b>Enable ACMP0_O Digital Output As Edge-detect Brake Source (Write Protect)</b> 0 = ACMP0_O as edge-detect brake source Disabled.

		<p>1 = ACMP0_O as edge-detect brake source Enabled.  <b>Note:</b> This register is write protected. Refer to SYS_REGLCTL register.</p>
--	--	--

**PWM Pin Polar Inverse Control (PWM\_POLCTL)**

Register	Offset	R/W	Description	Reset Value
PWM_POLCTL	PWMx_BA+0xD4	R/W	PWM 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	<p><b>PWM PIN Polar Inverse Control</b></p> <p>The register controls polarity state of PWMx_CHn output pin. Each bit n controls the corresponding PWM channel n.</p> <p>0 = PWMx_CHn output pin polar inverse Disabled.</p> <p>1 = PWMx_CHn output pin polar inverse Enabled.</p>

**PWM Output Enable Register (PWM\_POEN)**

Register	Offset	R/W	Description	Reset Value
PWM_POEN	PWMx_BA+0xD8	R/W	PWM 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	<b>PWMx_CHn Pin Output Enable Bits</b> Each bit n controls the corresponding PWM channel n. 0 = PWMx_CHn pin at tri-state. 1 = PWMx_CHn pin in output mode.

**PWM Software Brake Control Register (PWM\_SWBRK)**

Register	Offset	R/W	Description	Reset Value
PWM_SWBRK	PWMx_BA+0xDC	W	PWM Software Brake 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					BRKLTRG4	BRKLTRG2	BRKLTRG0
7	6	5	4	3	2	1	0
Reserved					BRKETRG4	BRKETRG2	BRKETRG0

Bits	Description	
[31:11]	Reserved	Reserved.
[n/2+8] n=0,2,4	BRKLTRGn	<b>PWM Level Brake Software Trigger (Write Only) (Write Protect)</b> Write 1 to this bit will trigger level brake, and set BRKLIFn bits to 1 in PWM_INTSTS1 register. Each bit n controls the corresponding PWM pair n. <b>Note:</b> This register is write protected. Refer to SYS_REGLCTL register.
[7:3]	Reserved	Reserved.
[n/2] n=0,2,4	BRKETRGn	<b>PWM Edge Brake Software Trigger (Write Only) (Write Protect)</b> Write 1 to this bit will trigger Edge brake, and set BRKEIFn bits to 1 in PWM_INTSTS1 register. Each bit n controls the corresponding PWM pair n. <b>Note:</b> This register is write protected. Refer to REGWRPROT register.

**PWM Interrupt Enable Register 0 (PWM\_INTEN0)**

Register	Offset	R/W	Description	Reset Value
PWM_INTEN0	PWMx_BA+0xE0	R/W	PWM Interrupt Enable Register 0	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
IFAIEN4_5	Reserved	CMPUIEN5	CMPUIEN4	CMPUIEN3	CMPUIEN2	CMPUIEN1	CMPUIEN0
15	14	13	12	11	10	9	8
IFAIEN2_3	Reserved	PIEN5	PIEN4	PIEN3	PIEN2	PIEN1	PIEN0
7	6	5	4	3	2	1	0
IFAIEN0_1	Reserved	ZIEN5	ZIEN4	ZIEN3	ZIEN2	ZIEN1	ZIEN0

Bits	Description
[31:30]	<b>Reserved</b> Reserved.
[n+24] n=0,1..5	<b>CMPDIENn</b> <b>PWM Compare Down Count Interrupt Enable Bits</b> Each bit n controls the corresponding PWM channel n. 0 = Compare down count interrupt Disabled. 1 = Compare down count interrupt Enabled. <b>Note:</b> In complementary mode, CMPDIEN1, 3, 5 is used as another CMPDIEN for channel 0, 2, 4.
[23]	<b>IFAIEN4_5</b> <b>PWM Channel 4/5 Interrupt Flag Accumulator Interrupt Enable Bit</b> 0 = Interrupt Flag accumulator interrupt Disabled. 1 = Interrupt Flag accumulator interrupt Enabled.
[22]	<b>Reserved</b> Reserved.
[n+16] n=0,1..5	<b>CMPUIENn</b> <b>PWM Compare Up Count Interrupt Enable Bits</b> Each bit n controls the corresponding PWM channel n. 0 = Compare up count interrupt Disabled. 1 = Compare up count interrupt Enabled. <b>Note:</b> In complementary mode, CMPUIEN1, 3, 5 is used as another CMPUIEN for channel 0, 2, 4.
[15]	<b>IFAIEN2_3</b> <b>PWM Channel 2/3 Interrupt Flag Accumulator Interrupt Enable Bit</b> 0 = Interrupt Flag accumulator interrupt Disabled. 1 = Interrupt Flag accumulator interrupt Enabled.
[14]	<b>Reserved</b> Reserved.
[n+8] n=0,1..5	<b>PIENn</b> <b>PWM Period Point Interrupt Enable Bits</b> Each bit n controls the corresponding PWM channel n. 0 = Period point interrupt Disabled. 1 = Period point interrupt Enabled. <b>Note1:</b> When up-down counter type period point means center point.

		<b>Note2:</b> Odd channels will read always 0 at complementary mode.
[7]	<b>IFAIEN0_1</b>	<b>PWM Channel 0/1 Interrupt Flag Accumulator Interrupt Enable Bit</b> 0 = Interrupt Flag accumulator interrupt Disabled. 1 = Interrupt Flag accumulator interrupt Enabled.
[6]	<b>Reserved</b>	Reserved.
[n] n=0,1..5	<b>ZIENn</b>	<b>PWM Zero Point Interrupt Enable Bits</b> Each bit n controls the corresponding PWM channel n. 0 = Zero point interrupt Disabled. 1 = Zero point interrupt Enabled. <b>Note:</b> Odd channels will read always 0 at complementary mode.



**PWM Interrupt Enable Register 1 (PWM\_INTEN1)**

Register	Offset	R/W	Description	Reset Value
PWM_INTEN1	PWMx_BA+0xE4	R/W	PWM Interrupt Enable 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					BRKLIEN4_5	BRKLIEN2_3	BRKLIEN0_1
7	6	5	4	3	2	1	0
Reserved					BRKEIEN4_5	BRKEIEN2_3	BRKEIEN0_1

Bits	Description	
[31:11]	Reserved	Reserved.
[10]	BRKLIEN4_5	<p><b>PWM Level-detect Brake Interrupt Enable for Channel4/5 (Write Protect)</b></p> <p>0 = Level-detect Brake interrupt for channel4/5 Disabled. 1 = Level-detect Brake interrupt for channel4/5 Enabled.</p> <p><b>Note:</b> This register is write protected. Refer to SYS_REGLCTL register.</p>
[9]	BRKLIEN2_3	<p><b>PWM Level-detect Brake Interrupt Enable for Channel2/3 (Write Protect)</b></p> <p>0 = Level-detect Brake interrupt for channel2/3 Disabled. 1 = Level-detect Brake interrupt for channel2/3 Enabled.</p> <p><b>Note:</b> This register is write protected. Refer to SYS_REGLCTL register.</p>
[8]	BRKLIEN0_1	<p><b>PWM Level-detect Brake Interrupt Enable for Channel0/1 (Write Protect)</b></p> <p>0 = Level-detect Brake interrupt for channel0/1 Disabled. 1 = Level-detect Brake interrupt for channel0/1 Enabled.</p> <p><b>Note:</b> This register is write protected. Refer to SYS_REGLCTL register.</p>
[7:3]	Reserved	Reserved.
[2]	BRKEIEN4_5	<p><b>PWM Edge-detect Brake Interrupt Enable for Channel4/5 (Write Protect)</b></p> <p>0 = Edge-detect Brake interrupt for channel4/5 Disabled. 1 = Edge-detect Brake interrupt for channel4/5 Enabled.</p> <p><b>Note:</b> This register is write protected. Refer to SYS_REGLCTL register.</p>
[1]	BRKEIEN2_3	<p><b>PWM Edge-detect Brake Interrupt Enable for Channel2/3 (Write Protect)</b></p> <p>0 = Edge-detect Brake interrupt for channel2/3 Disabled. 1 = Edge-detect Brake interrupt for channel2/3 Enabled.</p> <p><b>Note:</b> This register is write protected. Refer to SYS_REGLCTL register.</p>
[0]	BRKEIEN0_1	<p><b>PWM Edge-detect Brake Interrupt Enable for Channel0/1 (Write Protect)</b></p> <p>0 = Edge-detect Brake interrupt for channel0/1 Disabled. 1 = Edge-detect Brake interrupt for channel0/1 Enabled.</p>

		<b>Note:</b> This register is write protected. Refer to SYS_REGLCTL register.
--	--	---

**PWM Interrupt Flag Register 0 (PWM\_INTSTS0)**

Register	Offset	R/W	Description	Reset Value
PWM_INTSTS0	PWMx_BA+0xE8	R/W	PWM Interrupt Flag Register 0	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
IFAI4_5	Reserved	CMPUIF5	CMPUIF4	CMPUIF3	CMPUIF2	CMPUIF1	CMPUIF0
15	14	13	12	11	10	9	8
IFAI2_3	Reserved	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0
7	6	5	4	3	2	1	0
IFAI0_1	Reserved	ZIF5	ZIF4	ZIF3	ZIF2	ZIF1	ZIF0

Bits	Description	Description
[31:30]	Reserved	Reserved.
[n+24] n=0,1..5	CMPDIFn	<p><b>PWM Compare Down Count Interrupt Flag</b></p> <p>Flag is set by hardware when PWM counter down count and reaches CMP(PWM_CMPDATn[15:0]), software can clear this bit by writing 1 to it. Each bit n controls the corresponding PWM channel n.</p> <p><b>Note1:</b> If CMP equal to PERIOD, this flag is not working in down counter type selection.</p> <p><b>Note2:</b> In complementary mode, CMPDIF1, 3, 5 is used as another CMPDIF for channel 0, 2, 4.</p>
[23]	IFAI4_5	<p><b>PWM Channel 4/5 Interrupt Flag Accumulator Interrupt Flag</b></p> <p>Flag is set by hardware when condition match IFSEL4_5 bits in PWM_IFA register, software can clear this bit by writing 1 to it.</p>
[22]	Reserved	Reserved.
[n+16] n=0,1..5	CMPUIFn	<p><b>PWM Compare Up Count Interrupt Flag</b></p> <p>Flag is set by hardware when PWM counter up count and reaches CMP(PWM_CMPDATn[15:0]), software can clear this bit by writing 1 to it. Each bit n controls the corresponding PWM channel n.</p> <p><b>Note1:</b> If CMP equal to PERIOD, this flag is not working in up counter type selection.</p> <p><b>Note2:</b> In complementary mode, CMPUIF1, 3, 5 is used as another CMPUIF for channel 0, 2, 4.</p>
[15]	IFAI2_3	<p><b>PWM Channel 2/3 Interrupt Flag Accumulator Interrupt Flag</b></p> <p>Flag is set by hardware when condition match IFSEL2_3 bits in PWM_IFA register, software can clear this bit by writing 1 to it.</p>
[14]	Reserved	Reserved.
[n+8] n=0,1..5	PIFn	<p><b>PWM Period Point Interrupt Flag</b></p> <p>This bit is set by hardware when PWM counter reaches PERIOD(PWM_PERIODn[15:0]), software can write 1 to clear this bit to zero. Each bit n controls the corresponding PWM channel n.</p>
[7]	IFAI0_1	<p><b>PWM Channel 0/1 Interrupt Flag Accumulator Interrupt Flag</b></p>

		Flag is set by hardware when condition match IFSEL0_1 bits in PWM_IFA register, software can clear this bit by writing 1 to it.
[6]	<b>Reserved</b>	Reserved.
[n] n=0,1..5	<b>ZIFn</b>	<b>PWM Zero Point Interrupt Flag</b> Each bit n controls the corresponding PWM channel n. This bit is set by hardware when PWM counter reaches zero, software can write 1 to clear this bit to zero.

**PWM Interrupt Flag Register 1 (PWM\_INTSTS1)**

Register	Offset	R/W	Description	Reset Value
PWM_INTSTS1	PWMx_BA+0xEC	R/W	PWM Interrupt Flag Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		BRKLSTS5	BRKLSTS4	BRKLSTS3	BRKLSTS2	BRKLSTS1	BRKLSTS0
23	22	21	20	19	18	17	16
Reserved		BRKESTS5	BRKESTS4	BRKESTS3	BRKESTS2	BRKESTS1	BRKESTS0
15	14	13	12	11	10	9	8
Reserved		BRKLIF5	BRKLIF4	BRKLIF3	BRKLIF2	BRKLIF1	BRKLIF0
7	6	5	4	3	2	1	0
Reserved		BRKEIF5	BRKEIF4	BRKEIF3	BRKEIF2	BRKEIF1	BRKEIF0

Bits	Description
[31:30]	<b>Reserved</b> Reserved.
[n+24] n=0,1..5	<b>BRKLSTSn</b> <b>PWM Channel N Level-detect Brake Status (Read Only)</b> 0 = PWM channel n level-detect brake state is released. 1 = When PWM channel n level-detect brake detects a falling edge of any enabled brake source; this flag will be set to indicate the PWM channel n at brake state. <b>Note:</b> This bit is read only and auto cleared by hardware. When enabled brake source return to high level, PWM will release brake state until current PWM period finished. The PWM waveform will start output from next full PWM period.
[23:22]	<b>Reserved</b> Reserved.
[n+16] n=0,1..5	<b>BRKESTSn</b> <b>PWM Channel N Edge-detect Brake Status</b> 0 = PWM channel n edge-detect brake state is released. 1 = When PWM channel n edge-detect brake detects a falling edge of any enabled brake source; this flag will be set to indicate the PWM channel n at brake state, writing 1 to clear.
[15:14]	<b>Reserved</b> Reserved.
[n+8]	<b>BRKLIFn</b> <b>PWM Channel N Level-detect Brake Interrupt Flag (Write Protect)</b> 0 = PWM channel n level-detect brake event do not happened. 1 = When PWM channel n level-detect brake event happened, this bit is set to 1, writing 1 to clear. <b>Note:</b> This bit is write protected. Refer to SYS_REGLCTL register.
[7:6]	<b>Reserved</b> Reserved.
[n] n=0,1..5	<b>BRKEIFn</b> <b>PWM Channel N Edge-detect Brake Interrupt Flag (Write Protect)</b> 0 = PWM channel n edge-detect brake event do not happened. 1 = When PWM channel n edge-detect brake event happened, this bit is set to 1, writing 1 to clear. <b>Note:</b> This bit is write protected. Refer to SYS_REGLCTL register.

**PWM Interrupt Flag Accumulator Register (PWM\_IFA)**

Register	Offset	R/W	Description	Reset Value
PWM_IFA	PWMx_BA+0xF0	R/W	PWM Interrupt Flag Accumulator Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
IFAEN4_5	IFSEL4_5			IFCNT4_5			
15	14	13	12	11	10	9	8
IFAEN2_3	IFSEL2_3			IFCNT2_3			
7	6	5	4	3	2	1	0
IFAEN0_1	IFSEL0_1			IFCNT0_1			

Bits	Description	
[31:24]	Reserved	Reserved.
[23]	IFAEN4_5	<b>PWM Channel 4/5 Interrupt Flag Accumulator Enable Bit</b> 0 = PWM Channel 4/5 interrupt flag accumulator Disabled. 1 = PWM Channel 4/5 interrupt flag accumulator Enabled.
[22:20]	IFSEL4_5	<b>PWM Channel 4/5 Interrupt Flag Accumulator Source Select</b> 000 = CNT equal to Zero in channel 4. 001 = CNT equal to PERIOD in channel 4. 010 = CNT equal to CMPU in channel 4. 011 = CNT equal to CMPD in channel 4. 100 = CNT equal to Zero in channel 5. 101 = CNT equal to PERIOD in channel 5. 110 = CNT equal to CMPU in channel 5. 111 = CNT equal to CMPD in channel 5.
[19:16]	IFCNT4_5	<b>PWM Channel 4/5 Interrupt Flag Counter</b> The register sets the count number which defines how many times of PWM Channel 4/5 period occurs to set IFAIF4_5 bit to request the PWM period interrupt. PWM flag will be set in every IFCNT4_5[3:0] times of PWM period.
[15]	IFAEN2_3	<b>PWM Channel 2/3 Interrupt Flag Accumulator Enable Bit</b> 0 = PWM Channel 2/3 interrupt flag accumulator Disabled. 1 = PWM Channel 2/3 interrupt flag accumulator Enabled.
[14:12]	IFSEL2_3	<b>PWM Channel 2/3 Interrupt Flag Accumulator Source Select</b> 000 = CNT equal to Zero in channel 2. 001 = CNT equal to PERIOD in channel 2. 010 = CNT equal to CMPU in channel 2. 011 = CNT equal to CMPD in channel 2. 100 = CNT equal to Zero in channel 3. 101 = CNT equal to PERIOD in channel 3.

		110 = CNT equal to CMPU in channel 3. 111 = CNT equal to CMPD in channel 3.
[11:8]	<b>IFCNT2_3</b>	<b>PWM Channel 2/3 Interrupt Flag Counter</b> The register sets the count number which defines how many times of PWM Channel 2/3 period occurs to set IFAIF2_3 bit to request the PWM period interrupt. PWM flag will be set in every IFCNT2_3[3:0] times of PWM period.
[7]	<b>IFAENO_1</b>	<b>PWM Channel 0/1 Interrupt Flag Accumulator Enable Bit</b> 0 = PWM Channel 0/1 interrupt flag accumulator Disabled. 1 = PWM Channel 0/1 interrupt flag accumulator Enabled.
[6:4]	<b>IFSELO_1</b>	<b>PWM Channel 0/1 Interrupt Flag Accumulator Source Select</b> 000 = CNT equal to Zero in channel 0. 001 = CNT equal to PERIOD in channel 0. 010 = CNT equal to CMPU in channel 0. 011 = CNT equal to CMPD in channel 0. 100 = CNT equal to Zero in channel 1. 101 = CNT equal to PERIOD in channel 1. 110 = CNT equal to CMPU in channel 1. 111 = CNT equal to CMPD in channel 1.
[3:0]	<b>IFCNT0_1</b>	<b>PWM Channel 0/1 Interrupt Flag Counter</b> The register sets the count number which defines how many times of PWM Channel 0/1 period occurs to set IFAIF0_1 bit to request the PWM period interrupt. PWM flag will be set in every IFCNT0_1 [3:0] times of PWM period.

**PWM Trigger ADC Source Select Register 0 (PWM\_ADCTS0)**

Register	Offset	R/W	Description	Reset Value
PWM_ADCTS0	PWMx_BA+0xF8	R/W	PWM Trigger ADC 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	PWM_CH3 Trigger ADC enable bit
[30:28]	Reserved	Reserved.
[27:24]	TRGSEL3	<b>PWM_CH3 Trigger ADC Source Select</b> 0000 = PWM_CH2 zero point. 0001 = PWM_CH2 period point. 0010 = PWM_CH2 zero or period point. 0011 = PWM_CH2 up-count compared point. 0100 = PWM_CH2 down-count compared point. 0101 = PWM_CH3 zero point. 0110 = PWM_CH3 period point. 0111 = PWM_CH3 zero or period point. 1000 = PWM_CH3 up-count compared point. 1001 = PWM_CH3 down-count compared point. 1010 = PWM_CH0 up-count free trigger compared point. 1011 = PWM_CH0 down-count free trigger compared point. 1100 = PWM_CH2 up-count free trigger compared point. 1101 = PWM_CH2 down-count free trigger compared point. 1110 = PWM_CH4 up-count free trigger compared point. 1111 = PWM_CH4 down-count free trigger compared point.
[23]	TRGEN2	PWM_CH2 Trigger ADC enable bit
[22:20]	Reserved	Reserved.
[19:16]	TRGSEL2	<b>PWM_CH2 Trigger ADC Source Select</b> 0000 = PWM_CH2 zero point. 0001 = PWM_CH2 period point. 0010 = PWM_CH2 zero or period point. 0011 = PWM_CH2 up-count compared point.



		<p>0100 = PWM_CH2 down-count compared point.                  0101 = PWM_CH3 zero point.                  0110 = PWM_CH3 period point.                  0111 = PWM_CH3 zero or period point.                  1000 = PWM_CH3 up-count compared point.                  1001 = PWM_CH3 down-count compared point.                  1010 = PWM_CH0 up-count free trigger compared point.                  1011 = PWM_CH0 down-count free trigger compared point.                  1100 = PWM_CH2 up-count free trigger compared point.                  1101 = PWM_CH2 down-count free trigger compared point.                  1110 = PWM_CH4 up-count free trigger compared point.                  1111 = PWM_CH4 down-count free trigger compared point.</p>
[15]	<b>TRGEN1</b>	PWM_CH1 Trigger ADC enable bit
[14:12]	<b>Reserved</b>	Reserved.
[11:8]	<b>TRGSEL1</b>	<p><b>PWM_CH1 Trigger ADC Source Select</b>                  0000 = PWM_CH0 zero point.                  0001 = PWM_CH0 period point.                  0010 = PWM_CH0 zero or period point.                  0011 = PWM_CH0 up-count compared point.                  0100 = PWM_CH0 down-count compared point.                  0101 = PWM_CH1 zero point.                  0110 = PWM_CH1 period point.                  0111 = PWM_CH1 zero or period point.                  1000 = PWM_CH1 up-count compared point.                  1001 = PWM_CH1 down-count compared point.                  1010 = PWM_CH0 up-count free trigger compared point.                  1011 = PWM_CH0 down-count free trigger compared point.                  1100 = PWM_CH2 up-count free trigger compared point.                  1101 = PWM_CH2 down-count free trigger compared point.                  1110 = PWM_CH4 up-count free trigger compared point.                  1111 = PWM_CH4 down-count free trigger compared point.</p>
[7]	<b>TRGEN0</b>	PWM_CH0 Trigger ADC enable bit
[6:4]	<b>Reserved</b>	Reserved.
[3:0]	<b>TRGSEL0</b>	<p><b>PWM_CH0 Trigger ADC Source Select</b>                  0000 = PWM_CH0 zero point.                  0001 = PWM_CH0 period point.                  0010 = PWM_CH0 zero or period point.                  0011 = PWM_CH0 up-count compared point.                  0100 = PWM_CH0 down-count compared point.                  0101 = PWM_CH1 zero point.                  0110 = PWM_CH1 period point.                  0111 = PWM_CH1 zero or period point.                  1000 = PWM_CH1 up-count compared point.                  1001 = PWM_CH1 down-count compared point.                  1010 = PWM_CH0 up-count free trigger compared point.                  1011 = PWM_CH0 down-count free trigger compared point.                  1100 = PWM_CH2 up-count free trigger compared point.</p>

		<p>1101 = PWM_CH2 down-count free trigger compared point.</p> <p>1110 = PWM_CH4 up-count free trigger compared point.</p> <p>1111 = PWM_CH4 down-count free trigger compared point.</p>
--	--	---

**PWM Trigger ADC Source Select Register 1 (PWM\_ADCTS1)**

Register	Offset	R/W	Description	Reset Value
PWM_ADCTS1	PWMx_BA+0xFC	R/W	PWM Trigger ADC 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]	<b>Reserved</b> Reserved.
[15]	<b>TRGEN5</b> PWM_CH5 Trigger ADC enable bit
[14:12]	<b>Reserved</b> Reserved.
[11:8]	<b>TRGSEL5</b> <b>PWM_CH5 Trigger ADC Source Select</b> 0000 = PWM_CH4 zero point. 0001 = PWM_CH4 period point. 0010 = PWM_CH4 zero or period point. 0011 = PWM_CH4 up-count compared point. 0100 = PWM_CH4 down-count compared point. 0101 = PWM_CH5 zero point. 0110 = PWM_CH5 period point. 0111 = PWM_CH5 zero or period point. 1000 = PWM_CH5 up-count compared point. 1001 = PWM_CH5 down-count compared point. 1010 = PWM_CH0 up-count free trigger compared point. 1011 = PWM_CH0 down-count free trigger compared point. 1100 = PWM_CH2 up-count free trigger compared point. 1101 = PWM_CH2 down-count free trigger compared point. 1110 = PWM_CH4 up-count free trigger compared point. 1111 = PWM_CH4 down-count free trigger compared point.
[7]	<b>TRGEN4</b> PWM_CH4 Trigger ADC enable bit
[6:4]	<b>Reserved</b> Reserved.
[3:0]	<b>TRGSEL4</b> <b>PWM_CH4 Trigger ADC Source Select</b> 0000 = PWM_CH4 zero point. 0001 = PWM_CH4 period point.

		<p>0010 = PWM_CH4 zero or period point.          0011 = PWM_CH4 up-count compared point.          0100 = PWM_CH4 down-count compared point.          0101 = PWM_CH5 zero point.          0110 = PWM_CH5 period point.          0111 = PWM_CH5 zero or period point.          1000 = PWM_CH5 up-count compared point.          1001 = PWM_CH5 down-count compared point.          1010 = PWM_CH0 up-count free trigger compared point.          1011 = PWM_CH0 down-count free trigger compared point.          1100 = PWM_CH2 up-count free trigger compared point.          1101 = PWM_CH2 down-count free trigger compared point.          1110 = PWM_CH4 up-count free trigger compared point.          1111 = PWM_CH4 down-count free trigger compared point.</p>
--	--	---

**PWM Free Trigger Compare Register 0 1, 2 3, 4 5 (PWM FTCMPDAT0 1, 2 3, 4 5)**

Register	Offset	R/W	Description	Reset Value
PWM_FTCMPDAT0_1	PWMx_BA+0x100	R/W	PWM Free Trigger Compare Register 0/1	0x0000_0000
PWM_FTCMPDAT2_3	PWMx_BA+0x104	R/W	PWM Free Trigger Compare Register 2/3	0x0000_0000
PWM_FTCMPDAT4_5	PWMx_BA+0x108	R/W	PWM Free Trigger Compare Register 4/5	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
FTCMP							
7	6	5	4	3	2	1	0
FTCMP							

Bits	Description	
[31:16]	Reserved	Reserved.
[15:0]	FTCMP	<b>PWM Free Trigger Compare Register</b> FTCMP use to compare with even CNT(PWM_CNTm[15:0], m=0,2,4) to trigger ADC. FTCMPDAT0, 2, 4 corresponding complementary pairs PWMx_CH0 and PWMx_CH1, PWMx_CH2 and PWMx_CH3, PWMx_CH4 and PWMx_CH5.

**PWM Synchronous Start Control Register (PWM\_SSCTL)**

Register	Offset	R/W	Description	Reset Value
PWM_SSCTL	PWMx_BA+0x110	R/W	PWM Synchronous Start 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							SSRC
7	6	5	4	3	2	1	0
Reserved		SSEN5	SSEN4	SSEN3	SSEN2	SSEN1	SSEN0

Bits	Description	
[31:9]	Reserved	Reserved.
[8]	SSRC	<b>PWM Synchronous Start Source Select Bit</b> 0 = Synchronous start source come from PWM0. 1 = Synchronous start source come from PWM1.
[7:6]	Reserved	Reserved.
[n] n=0,1..5	SSEn	<b>PWM Synchronous Start Function Enable Bits</b> When synchronous start function is enabled, the PWM counter enable register (PWM_CNTEN) can be enabled by writing PWM synchronous start trigger bit (CNTSEN). Each bit n controls the corresponding PWM channel n. 0 = PWM synchronous start function Disabled. 1 = PWM synchronous start function Enabled.

**PWM Synchronous Start Trigger Register (PWM\_SSTRG)**

Register	Offset	R/W	Description	Reset Value
PWM_SSTRG	PWMx_BA+0x11 4	W	PWM Synchronous Start Trigger 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							CNTSEN

Bits	Description	
[31:1]	Reserved	Reserved.
[0]	CNTSEN	<p><b>PWM Counter Synchronous Start Enable (Write Only)</b></p> <p>PMW counter synchronous enable function is used to make selected PWM channels (PWMx_CHn) start counting at the same time.</p> <p>Writing this bit to 1 will also set the counter enable bit (CNTENn, n denotes channel 0 to 5) if correlated PWM channel counter synchronous start function is enabled.</p>

**PWM Leading Edge Blanking Control Register (PWM\_LEBCTL)**

Register	Offset	R/W	Description	Reset Value
PWM_LEBCTL	PWMx_BA+0x118	R/W	PWM Leading Edge Blanking Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved						TRGTYPE	
15	14	13	12	11	10	9	8
Reserved					SRCEN4	SRCEN2	SRCEN0
7	6	5	4	3	2	1	0
Reserved							LEBEN

Bits	Description	
[31:18]	Reserved	Reserved.
[17:16]	TRGTYPE	<b>PWM Leading Edge Blanking Trigger Type</b> 00 = When detect leading edge blanking source rising edge, blanking counter start counting. 01 = When detect leading edge blanking source falling edge, blanking counter start counting. 10 = When detect leading edge blanking source rising or falling edge, blanking counter start counting. 11 = Reserved.
[15:11]	Reserved	Reserved.
[10]	SRCEN4	<b>PWM Leading Edge Blanking Source From PWMx_CH4 Enable Bit</b> 0 = PWM Leading Edge Blanking Source from PWMx_CH4 Disabled. 1 = PWM Leading Edge Blanking Source from PWMx_CH4 Enabled.
[9]	SRCEN2	<b>PWM Leading Edge Blanking Source From PWMx_CH2 Enable Bit</b> 0 = PWM Leading Edge Blanking Source from PWMx_CH2 Disabled. 1 = PWM Leading Edge Blanking Source from PWMx_CH2 Enabled.
[8]	SRCEN0	<b>PWM Leading Edge Blanking Source From PWMx_CH0 Enable Bit</b> 0 = PWM Leading Edge Blanking Source from PWMx_CH0 Disabled. 1 = PWM Leading Edge Blanking Source from PWMx_CH0 Enabled.
[7:1]	Reserved	Reserved.
[0]	LEBEN	<b>PWM Leading Edge Blanking Enable Bit</b> 0 = PWM Leading Edge Blanking Disabled. 1 = PWM Leading Edge Blanking Enabled.



**PWM Leading Edge Blanking Counter Register (PWM\_LEBCNT)**

Register	Offset	R/W	Description	Reset Value
PWM_LEBCNT	PWMx_BA+0x11C	R/W	PWM Leading Edge Blanking 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							LEBCNT
7	6	5	4	3	2	1	0
LEBCNT							

Bits	Description	
[31:9]	Reserved	Reserved.
[8:0]	LEBCNT	<b>PWM Leading Edge Blanking Counter</b> This counter value decides leading edge blanking window size. Blanking window size = LEBCNT+1, and LEB counter clock base is ECLK.

**PWM Status Register (PWM\_STATUS)**

Register	Offset	R/W	Description	Reset Value
PWM_STATUS	PWMx_BA+0x120	R/W	PWM Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved		ADCTRGF5	ADCTRGF4	ADCTRGF3	ADCTRGF2	ADCTRGF1	ADCTRGF0
15	14	13	12	11	10	9	8
Reserved					SYNCINF4	SYNCINF2	SYNCINF0
7	6	5	4	3	2	1	0
Reserved		CNTMAXF5	CNTMAXF4	CNTMAXF3	CNTMAXF2	CNTMAXF1	CNTMAXF0

Bits	Description	
[31:22]	Reserved	Reserved.
[n+16] n=0,1..5	ADCTRGFn	<p><b>ADC Start of Conversion Flag</b></p> <p>Each bit n controls the corresponding PWM channel n.</p> <p>0 = Indicates no ADC start of conversion trigger event has occurred.</p> <p>1 = Indicates an ADC start of conversion trigger event has occurred, software can write 1 to clear this bit.</p>
[15:11]	Reserved	Reserved.
[n/2+8] n=0,2,4	SYNCINFn	<p><b>Input Synchronization Latched Flag</b></p> <p>Each bit n controls the corresponding PWM channel n.</p> <p>0 = Indicates no SYNC_IN event has occurred.</p> <p>1 = Indicates an SYNC_IN event has occurred, software can write 1 to clear this bit.</p>
[7:6]	Reserved	Reserved.
[n] n=0,1..5	CNTMAXFn	<p><b>Time-base Counter Equal to 0xFFFF Latched Flag</b></p> <p>Each bit n controls the corresponding PWM channel n.</p> <p>0 = indicates the time-base counter(PWM_CNTn[15:0]) never reached its maximum value 0xFFFF.</p> <p>1 = indicates the time-base counter(PWM_CNTn[15:0]) reached its maximum value, software can write 1 to clear this bit.</p>

**PWM Capture Input Enable Register (PWM\_CAPINEN)**

Register	Offset	R/W	Description	Reset Value
PWM_CAPINEN	PWMx_BA+0x200	R/W	PWM 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	<p><b>Capture Input Enable Bits</b></p> <p>Each bit n controls the corresponding PWM channel n.</p> <p>0 = PWM Channel capture input path Disabled. The input of PWM channel capture function is always regarded as 0.</p> <p>1 = PWM Channel capture input path Enabled. The input of PWM channel capture function comes from correlative multifunction pin.</p>

**PWM Capture Control Register (PWM\_CAPCTL)**

Register	Offset	R/W	Description	Reset Value
PWM_CAPCTL	PWMx_BA+0x204	R/W	PWM 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.
[n+24] n=0,1..5	FCRLDENn	<b>Falling Capture Reload Enable Bits</b> Each bit n controls the corresponding PWM channel n. 0 = Falling capture reload counter Disabled. 1 = Falling capture reload counter Enabled.
[23:22]	Reserved	Reserved.
[n+16] n=0,1..5	RCRLDENn	<b>Rising Capture Reload Enable Bits</b> Each bit n controls the corresponding PWM channel n. 0 = Rising capture reload counter Disabled. 1 = Rising capture reload counter Enabled.
[15:14]	Reserved	Reserved.
[n+8] n=0,1..5	CAPINVn	<b>Capture Inverter Enable Bits</b> Each bit n controls the corresponding PWM 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	<b>Capture Function Enable Bits</b> Each bit n controls the corresponding PWM channel n. 0 = Capture function Disabled. RCAPDAT/FCAPDAT registers will not be updated. 1 = Capture function Enabled. Capture latched the PWM counter value when detected rising or falling edge of input signal and saved to RCAPDAT (Rising latch) and FCAPDAT (Falling latch) registers.

**PWM Capture Status Register (PWM\_CAPSTS)**

Register	Offset	R/W	Description	Reset Value
PWM_CAPSTS	PWMx_BA+0x208	R	PWM 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		CFLIFOV5	CFLIFOV4	CFLIFOV3	CFLIFOV2	CFLIFOV1	CFLIFOV0
7	6	5	4	3	2	1	0
Reserved		CRLIFOV5	CRLIFOV4	CRLIFOV3	CRLIFOV2	CRLIFOV1	CRLIFOV0

Bits	Description	
[31:14]	Reserved	Reserved.
[n+8] n=0,1..5	CFLIFOVn	<p><b>Capture Falling Latch Interrupt Flag Overrun Status (Read Only)</b></p> <p>This flag indicates if falling latch happened when the corresponding CFLIFn(PWM_CAPIF[13:8]) bit is 1. Each bit n controls the corresponding PWM channel n.</p> <p><b>Note:</b> This bit will be cleared automatically when user clear corresponding CFLIFn bit.</p>
[7:6]	Reserved	Reserved.
[n] n=0,1..5	CRLIFOVn	<p><b>Capture Rising Latch Interrupt Flag Overrun Status (Read Only)</b></p> <p>This flag indicates if rising latch happened when the corresponding CRLIFn(PWM_CAPIF[5:0]) bit is 1. Each bit n controls the corresponding PWM channel n.</p> <p><b>Note:</b> This bit will be cleared automatically when user clear corresponding CRLIFn bit.</p>

**PWM Rising Capture Data Register 0~5 (PWM\_RCAPDAT 0~5)**

Register	Offset	R/W	Description	Reset Value
PWM_RCAPDAT0	PWMx_BA+0x20C	R	PWM Rising Capture Data Register 0	0x0000_0000
PWM_RCAPDAT1	PWMx_BA+0x214	R	PWM Rising Capture Data Register 1	0x0000_0000
PWM_RCAPDAT2	PWMx_BA+0x21C	R	PWM Rising Capture Data Register 2	0x0000_0000
PWM_RCAPDAT3	PWMx_BA+0x224	R	PWM Rising Capture Data Register 3	0x0000_0000
PWM_RCAPDAT4	PWMx_BA+0x22C	R	PWM Rising Capture Data Register 4	0x0000_0000
PWM_RCAPDAT5	PWMx_BA+0x234	R	PWM 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]	<b>RCAPDAT</b> <b>PWM Rising Capture Data Register (Read Only)</b> When rising capture condition happened, the PWM counter value will be saved in this register.

**PWM Falling Capture Data Register 0~5 (PWM\_FCAPDAT 0~5)**

Register	Offset	R/W	Description	Reset Value
PWM_FCAPDAT0	PWMx_BA+0x210	R	PWM Falling Capture Data Register 0	0x0000_0000
PWM_FCAPDAT1	PWMx_BA+0x218	R	PWM Falling Capture Data Register 1	0x0000_0000
PWM_FCAPDAT2	PWMx_BA+0x220	R	PWM Falling Capture Data Register 2	0x0000_0000
PWM_FCAPDAT3	PWMx_BA+0x228	R	PWM Falling Capture Data Register 3	0x0000_0000
PWM_FCAPDAT4	PWMx_BA+0x230	R	PWM Falling Capture Data Register 4	0x0000_0000
PWM_FCAPDAT5	PWMx_BA+0x238	R	PWM 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]	<b>FCAPDAT</b> <b>PWM Falling Capture Data Register (Read Only)</b> When falling capture condition happened, the PWM counter value will be saved in this register.

**PWM PDMA Control Register (PWM\_PDMACTL)**

Register	Offset	R/W	Description	Reset Value
PWM_PDMACTL	PWMx_BA+0x23C	R/W	PWM PDMA Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved			CHSEL4_5	CAPORD4_5	CAPMOD4_5		CHEN4_5
15	14	13	12	11	10	9	8
Reserved			CHSEL2_3	CAPORD2_3	CAPMOD2_3		CHEN2_3
7	6	5	4	3	2	1	0
Reserved			CHSEL0_1	CAPORD0_1	CAPMOD0_1		CHEN0_1

Bits	Description	
[31:21]	Reserved	Reserved.
[20]	CHSEL4_5	<b>Select Channel 4/5 to Do PDMA Transfer</b> 0 = Channel4. 1 = Channel5.
[19]	CAPORD4_5	<b>Capture Channel 4/5 Rising/Falling Order</b> Set this bit to determine whether the PWM_RCAPDAT4/5 or PWM_FCAPDAT4/5 register is the first captured data transferred to memory through PDMA when CAPMOD4_5 bits are set to 0x3. 0 = PWM_FCAPDAT4/5 register is the first captured data to memory. 1 = PWM_RCAPDAT4/5 register is the first captured data to memory.
[18:17]	CAPMOD4_5	<b>Select PWM_RCAPDAT4/5 or PWM_FCAPDAT4/5 to Do PDMA Transfer</b> 00 = Reserved. 01 = PWM_RCAPDAT4/5 register. 10 = PWM_FCAPDAT4/5 register. 11 = Both PWM_RCAPDAT4/5 and PWM_FCAPDAT4/5 registers.
[16]	CHEN4_5	<b>Channel 4/5 PDMA Enable</b> 0 = Channel 4/5 PDMA function Disabled. 1 = Channel 4/5 PDMA function Enabled for the channel 4/5 captured data and transfer to memory.
[15:13]	Reserved	Reserved.
[12]	CHSEL2_3	<b>Select Channel 2/3 to Do PDMA Transfer</b> 0 = Channel2. 1 = Channel3.
[11]	CAPORD2_3	<b>Capture Channel 2/3 Rising/Falling Order</b> Set this bit to determine whether the PWM_RCAPDAT2/3 or PWM_FCAPDAT2/3 register is the first captured data transferred to memory through PDMA when



		<p>CAPMOD2_3 bits are set to 0x3.</p> <p>0 = PWM_FCAPDAT2/3 register is the first captured data to memory.</p> <p>1 = PWM_RCAPDAT2/3 register is the first captured data to memory.</p>
[10:9]	<b>CAPMOD2_3</b>	<p><b>Select PWM_RCAPDAT2/3 or PWM_FCAODAT2/3 to Do PDMA Transfer</b></p> <p>00 = Reserved.</p> <p>01 = PWM_RCAPDAT2/3 register.</p> <p>10 = PWM_FCAPDAT2/3 register.</p> <p>11 = Both PWM_RCAPDAT2/3 and PWM_FCAPDAT2/3 registers.</p>
[8]	<b>CHEN2_3</b>	<p><b>Channel 2/3 PDMA Enable</b></p> <p>0 = Channel 2/3 PDMA function Disabled.</p> <p>1 = Channel 2/3 PDMA function Enabled for the channel 2/3 captured data and transfer to memory.</p>
[7:5]	<b>Reserved</b>	Reserved.
[4]	<b>CHSEL0_1</b>	<p><b>Select Channel 0/1 to Do PDMA Transfer</b></p> <p>0 = Channel0.</p> <p>1 = Channel1.</p>
[3]	<b>CAPORD0_1</b>	<p><b>Capture Channel 0/1 Rising/Falling Order</b></p> <p>Set this bit to determine whether the PWM_RCAPDAT0/1 or PWM_FCAPDAT0/1 register is the first captured data transferred to memory through PDMA when CAPMOD0_1 bits are set to 0x3.</p> <p>0 = PWM_FCAPDAT0/1 register is the first captured data to memory.</p> <p>1 = PWM_RCAPDAT0/1 register is the first captured data to memory.</p>
[2:1]	<b>CAPMOD0_1</b>	<p><b>Select PWM_RCAPDAT0/1 or PWM_FCAPDAT0/1 to Do PDMA Transfer</b></p> <p>00 = Reserved.</p> <p>01 = PWM_RCAPDAT0/1 register.</p> <p>10 = PWM_FCAPDAT0/1 register.</p> <p>11 = Both PWM_RCAPDAT0/1 and PWM_FCAPDAT0/1 registers.</p>
[0]	<b>CHEN0_1</b>	<p><b>Channel 0/1 PDMA Enable Bit</b></p> <p>0 = Channel 0/1 PDMA function Disabled.</p> <p>1 = Channel 0/1 PDMA function Enabled for the channel 0/1 captured data and transfer to memory.</p>

**PWM Capture Channel 0 1, 2 3, 4 5 PDMA Register (PWM\_PDMACAP 0 1, 2 3, 4 5)**

Register	Offset	R/W	Description	Reset Value
PWM_PDMACAP0_1	PWMx_BA+0x240	R	PWM Capture Channel 0/1 PDMA Register	0x0000_0000
PWM_PDMACAP2_3	PWMx_BA+0x244	R	PWM Capture Channel 2/3 PDMA Register	0x0000_0000
PWM_PDMACAP4_5	PWMx_BA+0x248	R	PWM Capture Channel 4/5 PDMA 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
CAPBUF							
7	6	5	4	3	2	1	0
CAPBUF							

Bits	Description	
[31:16]	Reserved	Reserved.
[15:0]	CAPBUF	<b>PWM Capture PDMA Register (Read Only)</b> This register is used as a buffer to transfer PWM capture rising or falling data to memory by PDMA.

**PWM Capture Interrupt Enable Register (PWM\_CAPIEN)**

Register	Offset	R/W	Description	Reset Value
PWM_CAPIEN	PWMx_BA+0x250	R/W	PWM 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.
[n+8] n=0,1..5	CAPFIENn	<p><b>PWM Capture Falling Latch Interrupt Enable Bits</b> Each bit n controls the corresponding PWM channel n. 0 = Capture falling edge latch interrupt Disabled. 1 = Capture falling edge latch interrupt Enabled. <b>Note:</b> When Capture with PDMA operating, corresponding channel CAPFIENn bit must be disabled.</p>
[7:6]	Reserved	Reserved.
[n] n=0,1..5	CAPRIENn	<p><b>PWM Capture Rising Latch Interrupt Enable Bits</b> Each bit n controls the corresponding PWM channel n. 0 = Capture rising edge latch interrupt Disabled. 1 = Capture rising edge latch interrupt Enabled. <b>Note:</b> When Capture with PDMA operating, corresponding channel CAPRIENn bit must be disabled.</p>

**PWM Capture Interrupt Flag Register (PWM\_CAPIF)**

Register	Offset	R/W	Description	Reset Value
PWM_CAPIF	PWMx_BA+0x254	R/W	PWM 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		CFLIF5	CFLIF4	CFLIF3	CFLIF2	CFLIF1	CFLIF0
7	6	5	4	3	2	1	0
Reserved		CRLIF5	CRLIF4	CRLIF3	CRLIF2	CRLIF1	CRLIF0

Bits	Description	
[31:14]	Reserved	Reserved.
[n+8] n=0,1..5	CFLIFn	<p><b>PWM Capture Falling Latch Interrupt Flag</b></p> <p>This bit is writing 1 to clear. Each bit n controls the corresponding PWM channel n.</p> <p>0 = No capture falling latch condition happened.</p> <p>1 = Capture falling latch condition happened, this flag will be set to high.</p> <p><b>Note:</b> When Capture with PDMA operating, corresponding channel CFLIFn bit will cleared by hardware after PDMA transfer data.</p>
[7:6]	Reserved	Reserved.
[n] n=0,1..5	CRLIFn	<p><b>PWM Capture Rising Latch Interrupt Flag</b></p> <p>This bit is writing 1 to clear. Each bit n controls the corresponding PWM channel n.</p> <p>0 = No capture rising latch condition happened.</p> <p>1 = Capture rising latch condition happened, this flag will be set to high.</p> <p><b>Note:</b> When Capture with PDMA operating, corresponding channel CRLIFn bit will cleared by hardware after PDMA transfer data.</p>

**PWM Period Register Buffer 0~5 (PWM\_PBUF0~5)**

Register	Offset	R/W	Description	Reset Value
PWM_PBUF0	PWMx_BA+0x304	R	PWM PERIOD0 Buffer	0x0000_0000
PWM_PBUF1	PWMx_BA+0x308	R	PWM PERIOD1 Buffer	0x0000_0000
PWM_PBUF2	PWMx_BA+0x30C	R	PWM PERIOD2 Buffer	0x0000_0000
PWM_PBUF3	PWMx_BA+0x310	R	PWM PERIOD3 Buffer	0x0000_0000
PWM_PBUF4	PWMx_BA+0x314	R	PWM PERIOD4 Buffer	0x0000_0000
PWM_PBUF5	PWMx_BA+0x318	R	PWM PERIOD5 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	<b>PWM Period Register Buffer (Read Only)</b> Used as PERIOD active register.

**PWM Comparator Register Buffer 0~5 (PWM\_CMPBUF0~5)**

Register	Offset	R/W	Description	Reset Value
PWM_CMPBUF0	PWMx_BA+0x31C	R	PWM CMPDAT0 Buffer	0x0000_0000
PWM_CMPBUF1	PWMx_BA+0x320	R	PWM CMPDAT1 Buffer	0x0000_0000
PWM_CMPBUF2	PWMx_BA+0x324	R	PWM CMPDAT2 Buffer	0x0000_0000
PWM_CMPBUF3	PWMx_BA+0x328	R	PWM CMPDAT3 Buffer	0x0000_0000
PWM_CMPBUF4	PWMx_BA+0x32C	R	PWM CMPDAT4 Buffer	0x0000_0000
PWM_CMPBUF5	PWMx_BA+0x330	R	PWM CMPDAT5 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]	<b>Reserved</b> Reserved.
[15:0]	<b>CMPBUF</b> <b>PWM Comparator Register Buffer (Read Only)</b> Used as CMPDAT active register.

**PWM CLKPSC Buffer 0 1, 2 3, 4 5 (PWM CPSCBUF0 1, 2 3, 4 5)**

Register	Offset	R/W	Description	Reset Value
PWM_CPSCBUF0_1	PWMx_BA+0x334	R	PWM CLKPSC0_1 Buffer	0x0000_0000
PWM_CPSCBUF2_3	PWMx_BA+0x338	R	PWM CLKPSC2_3 Buffer	0x0000_0000
PWM_CPSCBUF4_5	PWMx_BA+0x33C	R	PWM CLKPSC4_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
Reserved				CPSCBUF			
7	6	5	4	3	2	1	0
CPSCBUF							

Bits	Description	
[31:12]	Reserved	Reserved.
[11:0]	CPSCBUF	<b>PWM Counter Clock Pre-scale Buffer</b> Used as PWM counter clock pre-scale active register.

**PWM FTCMPDAT Buffer 0 1, 2 3, 4 5 (PWM FTCBUF0 1,2 3,4 5)**

Register	Offset	R/W	Description	Reset Value
PWM_FTCBU F0_1	PWMx_BA+0x340	R	PWM FTCMPDAT0_1 Buffer	0x0000_0000
PWM_FTCBU F2_3	PWMx_BA+0x344	R	PWM FTCMPDAT2_3 Buffer	0x0000_0000
PWM_FTCBU F4_5	PWMx_BA+0x348	R	PWM FTCMPDAT4_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
FTCMPBUF							
7	6	5	4	3	2	1	0
FTCMPBUF							

Bits	Description	
[31:16]	Reserved	Reserved.
[15:0]	FTCMPBUF	<b>PWM FTCMPDAT Buffer (Read Only)</b> Used as FTCMPDAT active register.



**PWM FTCMPDAT Indicator Register (PWM\_FTCI)**

Register	Offset	R/W	Description	Reset Value
PWM_FTCI	PWMx_BA+0x34C	R/W	PWM FTCMPDAT 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					FTCMD4	FTCMD2	FTCMD0
7	6	5	4	3	2	1	0
Reserved					FTCMU4	FTCMU2	FTCMU0

Bits	Description
[31:11]	<b>Reserved</b> Reserved.
[n/2+8] n=0,2,4	<b>FTCMDn</b> <b>PWM FTCMPDAT Down Indicator</b> Indicator will be set to high when FT_CMP(PWM_FTCMPDATn[15:0]) bits equal to PERIOD(PWM_PERIODn[15:0]) bits and DIRF(PWM_CNTn[16]) bit is 0, software can write 1 to clear this bit. Each bit n controls the corresponding PWM channel n.
[7:3]	<b>Reserved</b> Reserved.
[n/2] n=0,2,4	<b>FTCMUn</b> <b>PWM FTCMPDAT Up Indicator</b> Indicator will be set to high when FT_CMP(PWM_FTCMPDATn[15:0]) bits equal to PERIOD(PWM_PERIODn[15:0]) bits and DIRF(PWM_CNTn[16]) bit is 1, software can write 1 to clear this bit. Each bit n controls the corresponding PWM channel n.

## 6.14 Real Time Clock (RTC)

### 6.14.1 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.

### 6.14.2 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 chip wake-up from Idle or Power-down mode while an RTC interrupt signal is generated
- Supports Daylight Saving Time backup control in RTC\_DSTCTL

### 6.14.3 Block Diagram

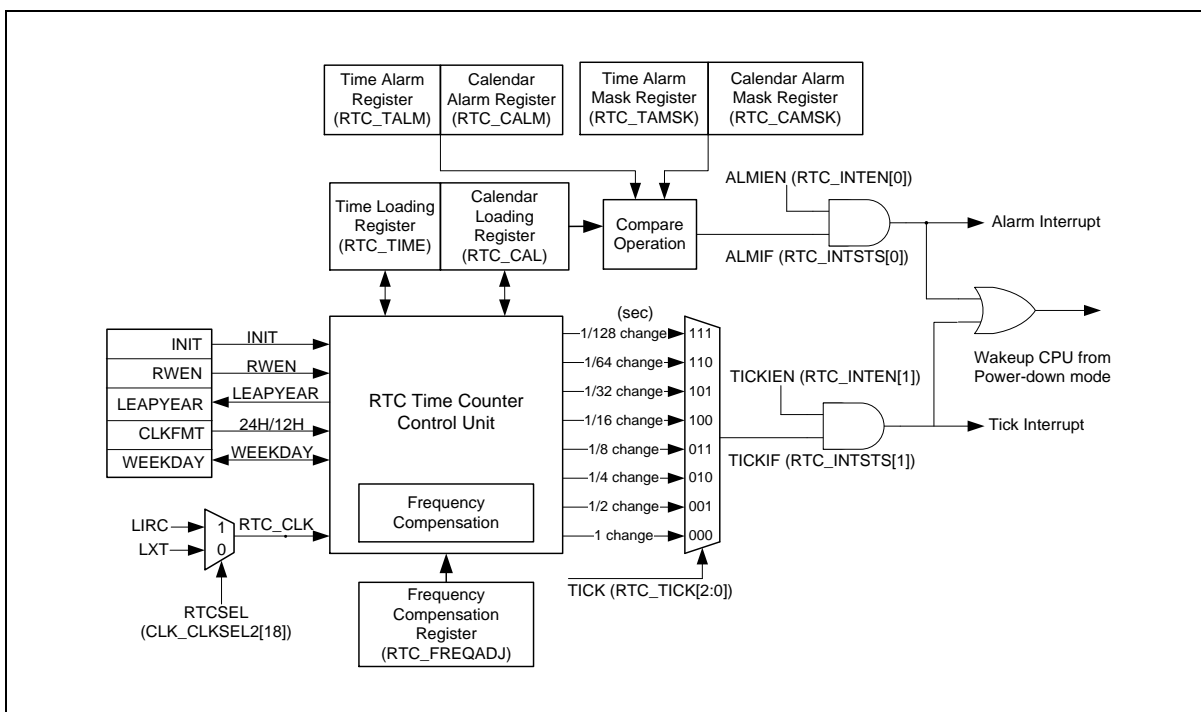


Figure 6.14-1 RTC Block Diagram

### 6.14.4 Basic Configuration

- Clock Source Configuration
  - Select the source of RTC on RTCSEL (CLK\_CLKSEL2[18]).

Group	Pin Name	GPIO	MFP
X32	X32_IN	PF.1	MFP1
	X32_OUT	PF.0	MFP1

### 6.14.5 Functional Description

#### 6.14.5.1 RTC Initiation

When an RTC block is powered on, RTC is at reset state. User has to write a special number 0xA5EB1357 to INIT (RTC\_INIT[31:0]) to make RTC leaving reset state. Once INIT has written as 0xA5EB1357, the RTC will be in normal active state permanently. User can read ACTIVE (RTC\_INIT[0]) to check the RTC is at normal active state or reset state.

#### 6.14.5.2 RTC Read/Write Enable

The RWEN (RTC\_RWEN[15:0]) is served as unlock write-protection function of RTC registers. If RWEN is written to 0xA965, user can read RWENF (RTC\_RWEN[16]) to check the RTC registers are write accessible or locked. Once RWENF bit has been set, RTC write accessible period will keep effect at least 1024 RTC clocks (about 30ms, if RTC clock source is LXT) and RWENF will be cleared automatically after 1024 RTC clocks expired. If user writes RWEN 0xA965 during RWENF flag high, the write accessible period will be auto extended from 768 ~1024 RTC clocks. Table 6.14-1 shows the read/write access attribute of RTC registers.

Register	INIR = 0	RWENF = 1	RWENF = 0 Or RTCBUSY = 1
----------	----------	-----------	--------------------------

RTC_INIT	available	R/W	R/W
RTC_RWEN	available	R/W	R/W
RTC_FREQADJ	available	R/W	R
RTC_TIME	Not available	R/W	R
RTC_CAL	Not available	R/W	R
RTC_CLKFMT	Not available	R/W	R
RTC_WEEKDAY	Not available	R/W	R
RTC_TALM	Not available	R/W	R
RTC_CALM	Not available	R/W	R
RTC_LEAPYEAR	Not available	R	R
RTC_INTEN	available	R/W	R/W
RTC_INTSTS	available	R/W	R/W
RTC_TICK	Not available	R/W	R
RTC_TAMSK	Not available	R/W	R
RTC_CAMSK	Not available	R/W	R
RTC_LXTCTL	available	R/W	R/W
RTC_LXTOCTL	available	R/W	R/W
RTC_LXTOICTL	available	R/W	R/W
RTC_DSTCTL	Not available	R/W	R

Table 6.14-1 RTC Read/Write Access Attribute

6.14.5.3 Frequency Compensation

The FREQADJ (RTC\_FREQADJ[21:0]) allows user to make digital compensation to a clock input for RTC application more accurate. The FREQADJ default value is 0x00200000 if RTC clock source is LXT and 0x00080000 if RTC clock source is LIRC, means no frequency compensate for actual RTC clock frequency. Follow the example and formula below to write compensation value of actual RTC clock frequency to FREQADJ. Example 1 and example 2 below are the compensation examples for higher or lower than standard 32 kHz, if RTC clock source is selected from LXT.

**Example 1:**

Actual frequency measurement: 32773.65 Hz (> 32768 Hz)

$$FREQADJ = (32768 * 0x200000) / 32773.65 = 0x1FFE96$$

**Example 2:**

Actual frequency measurement: 32763.25 Hz (< 32768 Hz)

$$FREQADJ = (32768 * 0x200000) / 32763.25 = 0x200130$$

User can enable CLK1HZEN (CLK\_CLKOCTL[6]) function and relative clock output pin to measure 1

Hz clock output frequency based on compensated RTC clock source.

6.14.5.4 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.

6.14.5.5 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. In 12-hour time scale mode, PM hour digits in RTC\_TIME[21:16] is PM hour time + 0x20, that is if current hour is PM01 and the PM hour digits should be 0x21 (0x01 + 0x20) . The RTC\_TIME[21:16] value in 12-hour and 24-hour time scale mode are all listed in Table 6.14-2.

Note: The Hour Value Write Into RTC_TIME[21:16] And Expressed In BCD Format.			
24-Hour Time Scale (24HEN = 1)		12-Hour Time Scale (24HEN = 0)	
0x00	0x12	0x12 (AM12)	0x32 (PM12)
0x01	0x13	0x01 (AM01)	0x21 (PM01)
0x02	0x14	0x02 (AM02)	0x22 (PM02)
0x03	0x15	0x03 (AM03)	0x23 (PM03)
0x04	0x16	0x04 (AM04)	0x24 (PM04)
0x05	0x17	0x05 (AM05)	0x25 (PM05)
0x06	0x18	0x06 (AM06)	0x26 (PM06)
0x07	0x19	0x07 (AM07)	0x27 (PM07)
0x08	0x20	0x08 (AM08)	0x28 (PM08)
0x09	0x21	0x09 (AM09)	0x29 (PM09)
0x10	0x22	0x10 (AM10)	0x30 (PM10)
0x11	0x23	0x11 (AM11)	0x31 (PM11)

Table 6.14-212/24-Hour Time Scale Selection

6.14.5.6 Day of the Week Counter

The RTC controller provides day of week in WEEKDAY (RTC\_WEEKDAY[2:0]). The value is defined from 0 to 6 to represent from Sunday to Saturday respectively.

6.14.5.7 Periodic Time Tick Interrupt

The RTC periodic time tick 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 (RTC\_TICK[2:0]). When specified time tick event generated, TICKIF (RTC\_INTSTS[1]) will be set to indicate time tick event occurred. If TICKIEN (RTC\_INTEN[1]) is enabled also, the time tick interrupt signal will be generated and inform to CPU periodically.

6.14.5.8 Alarm Interrupt

When the RTC real time and calendar in RTC\_TIME and RTC\_CAL registers are equal to alarm time and calendar in RTC\_TALM and RTC\_CALM registers, the RTC alarm interrupt flag ALMIF

(RTC\_INTSTS[0]) will be set to 1 and the RTC alarm interrupt signal assert if the alarm interrupt enable ALMIEN (RTC\_INTEN[0]) is enabled.

The RTC controller also provides alarm mask function which controlled in RTC\_TAMSK and RTC\_CAMSK registers. User can 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.

#### 6.14.5.9 Daylight Saving Time

The RTC controller also provides RTC\_DSTCTL register to store the control settings of daylight saving time application. User can read RTC\_DSTCTL value to check current RTC date/time counter runs in daylight saving time mode or normal mode.

#### 6.14.5.10 Application Note

1. All data in RTC\_TIME, RTC\_CAL, RTC\_TALM and RTC\_CALM registers are expressed in BCD format.
2. User has to make sure that the loaded values are reasonable in specified fields. For example, load RTC\_CAL as 201a (year), 13 (month), 00 (day), or RTC\_CAL does not match with RTC\_WEEKDAY, etc.
3. Registers value after core power and RTC battery power, both first powered on are shown in Table 6.14-3.

Register	Default Value
RTC_INIT	0
RTC_RWEN	0
RTC_FREQADJ	0x00200000
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 6.14-3 Registers Default Value After Powered On

4. In RTC\_CAL and RTC\_CALM registers, only 2 BCD digits are used to express “year”. The 2 BCD digits of xy means 20xy, rather than 19xy or 21xy.
5. Example of 12-Hour Time Setting  
If current RTC time is PM12:59:30 in 12-hour time scale mode, the RTC\_TIME setting should be:  
RTC\_TIME[21:16]: 0x32 (0x12+0x20)

TENHR (RTC\_TIME[21:20]) is 0x3, HR (RTC\_TIME[19:16]) is 0x2.  
 RTC\_TIME[14:8]: 0x59  
 TENMIN (RTC\_TIME[14:12]) is 0x5, MIN (RTC\_TIME[11:8]) is 0x9.  
 RTC\_TIME[6:0]: 0x30  
 TENSEC (RTC\_TIME[6:4]) is 0x3, SEC (RTC\_TIME[3:0]) is 0x0.

6. Registers in RTC battery power domain and registers in core power domain are shown in Table 6.14-4.

Register	Power Domain
RTC_INIT	Battery Power Domain
RTC_RWEN	Core Power Domain
RTC_FREQADJ	Battery Power Domain
RTC_TIME	Battery Power Domain
RTC_CAL	Battery Power Domain
RTC_CLKFMT	Battery Power Domain
RTC_WEEKDAY	Battery Power Domain
RTC_TALM	Battery Power Domain
RTC_CALM	Battery Power Domain
RTC_LEAPYEAR	Battery Power Domain
RTC_INTEN	Core Power Domain
RTC_INTSTS	Core Power Domain
RTC_TICK	Battery Power Domain
RTC_TAMSK	Battery Power Domain
RTC_CAMSK	Battery Power Domain
RTC_LXTCTL	Battery Power Domain
RTC_LXTOCTL	Battery Power Domain
RTC_LXTICTL	Battery Power Domain
RTC_PF2CTL	Battery Power Domain
RTC_DSTCTL	Battery Power Domain

Table 6.14-4 Registers Power Domain

6.14.5.11 Backup Domain GPIO Function

When PF.0/X32O and PF.1/X32I pins are not used for low speed 32K oscillator function, these pins can be used as GPIO pin function. The CTLSEL in RTC\_LXTOCTL[3] is used to select PF.0/X32O pin is controlled by RTC or GPIO module and PF.1/X32I pin is controlled by CTLSEL in RTC\_LXTICTL[3]. CTLSEL in RTC\_PF2CTL[3] is also use to select PF.2 pin is controlled by RTC or GPIO module.

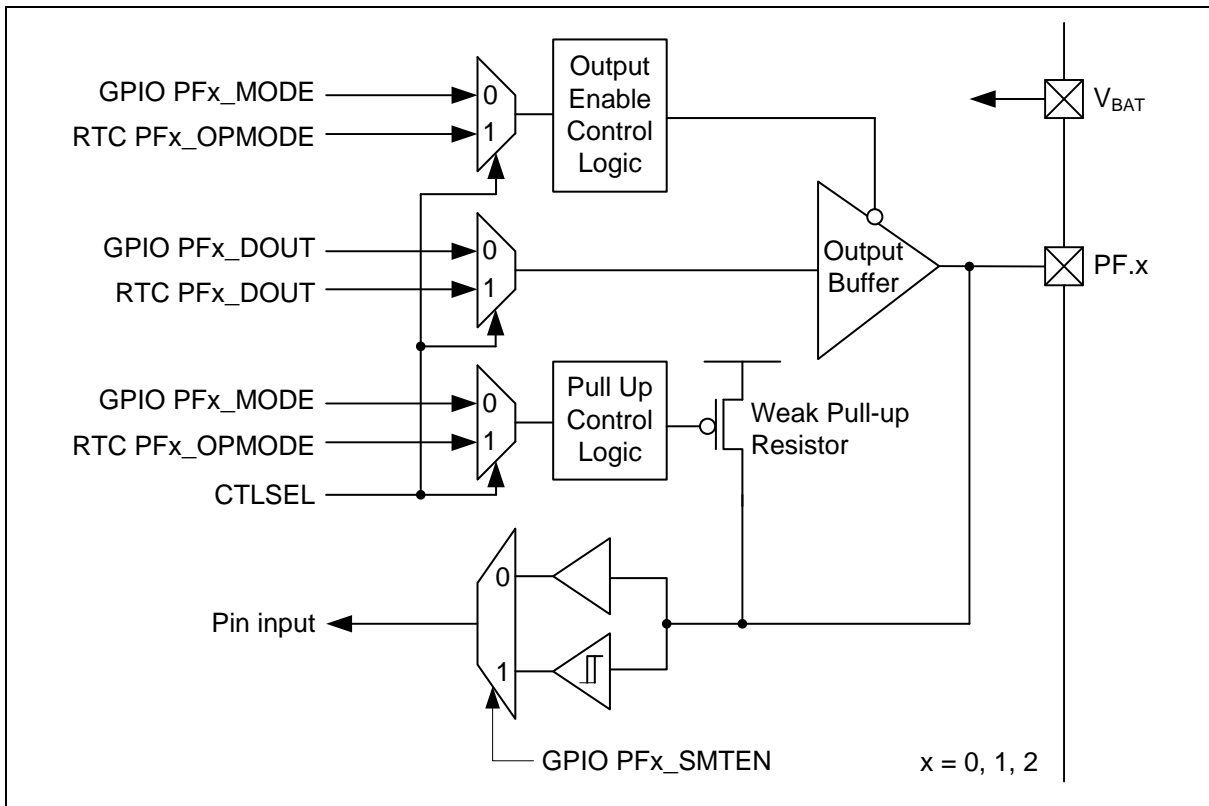


Figure 6.14-2 Backup I/O Control Diagram



6.14.6 Register Map

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

Register	Offset	R/W	Description	Reset Value
<b>RTC Base Address:</b> RTC_BA = 0x4000_8000				
RTC_INIT	RTC_BA+0x00	R/W	RTC Initiation Register	0x0000_0000
RTC_RWEN	RTC_BA+0x04	R/W	RTC Access Enable Register	0x0000_0000
RTC_FREQADJ	RTC_BA+0x08	R/W	RTC Frequency Compensation Register	0x0020_0000
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 Indicaton 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 kHz Oscillator Control Register	0x0000_000E
RTC_LXTOCTL	RTC_BA+0x104	R/W	RTC X32KO Pin Control Register	0x0000_0000
RTC_LXTICTL	RTC_BA+0x108	R/W	RTC X32KI Pin Control Register	0x0000_0000
RTC_PF2CTL	RTC_BA+0x10C	R/W	RTC PF.2 Pin Control Register	0x0000_0000
RTC_DSTCTL	RTC_BA+0x110	R/W	RTC Daylight Saving Time Control Register	0x0000_0000

6.14.7 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							INIT[0]/ACTIVE

Bits	Description	
[31:1]	INIT	<p><b>RTC Initiation</b></p> <p>When RTC block is first powered on, RTC is at reset state. User has to write a special number 0xA5EB1357 to INIT to make RTC leaving reset state. Once the INIT is written as 0xA5EB1357 the RTC will be at normal active state permanently.</p> <p>The INIT[31:1] is a write-only field and read value will be always 0.</p>
[0]	INIT[0]/ACTIVE	<p><b>RTC Active Status (Read Only)</b></p> <p>0 = RTC is at reset state.</p> <p>1 = RTC is at normal active state.</p>

**RTC Access Enable Register (RTC\_RWEN)**

Register	Offset	R/W	Description	Reset Value
RTC_RWEN	RTC_BA+0x04	R/W	RTC Access Enable Register	0x0000_0000

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

Bits	Description	
[31:25]	Reserved	Reserved.
[24]	RTCBUSY	<p><b>RTC Write Busy Flag</b></p> <p>This bit indicates RTC registers are writable or not.</p> <p>0 = RTC register write Disabled.</p> <p>1 = RTC register write Enabled.</p> <p><b>Note:</b> RTCBUSY flag will be set when execute write RTC register command exceed 6 times within 1120 PCLK cycles.</p>
[23:17]	Reserved	Reserved.
[16]	RWENF	<p><b>RTC Register Access Enable Bit (Read Only)</b></p> <p>0 = RTC register read/write Disabled.</p> <p>1 = RTC register read/write Enabled.</p> <p><b>Note1:</b> This bit will be set after RWEN is load a 0xA965, and be cleared automatically after 1024 RTC clocks expired.</p> <p><b>Note2:</b> RWENF will be mask to 0 during RTCBUSY is 1, and first turn on RTCCKEN (CLK_APBCLK[1]) also.</p>
[15:0]	RWEN	<p><b>RTC Register Access Enable Password (Write Only)</b></p> <p>Writing 0xA965 to this field will enable RTC register access period and keep 1024 RTC clocks.</p> <p><b>Note:</b> Writing others vaule will clear RWENF and disable RTC register access function immediately.</p>

**RTC Frequency Compensation Register (RTC\_FREQADJ)**

Register	Offset	R/W	Description	Reset Value
RTC_FREQADJ	RTC_BA+0x08	R/W	RTC Frequency Compensation Register	0x0020_0000

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

Bits	Description	
[31:22]	Reserved	Reserved.
[21:0]	FREQADJ	<p><b>Frequency Compensation Value</b>                      User has to get actual clock frequency of LXT, LXT frequency.  <math>FCR = 0x200000 * (32768 / \text{LXT frequency})</math>.</p> <p><b>Note:</b> This formula is suitable only when RTCSEL (CLK_CLKSEL2[18]) is 0, RTC clock source is from LXT.</p>

**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	
[31:24]	Reserved	Reserved.
[21:20]	TENHR	<b>10-hour Time Digit (0~2)</b> <b>Note:</b> When RTC runs as 12-hour time scale mode, RTC_TIME[21] (the high bit of TENHR[1:0]) means AM/PM indication, RTC_TIME[21] is 0 means AM hour and RTC_TIME[21] is 1 means PM hour.
[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.

**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.

**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							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved							24HEN

Bits	Description	
[31:1]	Reserved	Reserved.
[0]	24HEN	<p><b>24-hour / 12-hour Time Scale Selection</b></p> <p>Indicates that RTC_TIME and RTC_TALM register are in 24-hour time scale or 12-hour time scale</p> <p>0 = 12-hour time scale with AM and PM indication selected.</p> <p>1 = 24-hour time scale selected.</p>

**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	<p><b>Day of the Week Register</b></p> <p>000 = Sunday.                      001 = Monday.                      010 = Tuesday.                      011 = Wednesday.                      100 = Thursday.                      101 = Friday.                      110 = Saturday.                      111 = Reserved.</p> <p><b>Note:</b> RTC will not check WEEKDAY setting with RTC_CAL is reasonable or not.</p>



**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	
[31:24]	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 and RTC will not check loaded data.
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 and RTC will not check loaded data.
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 Indication 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	<b>Leap Year Indication Register (Read Only)</b> 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	<b>Time Tick Interrupt Enable Bit</b> 0 = RTC time tick interrupt Disabled. 1 = RTC time tick interrupt Enabled.
[0]	ALMIEN	<b>Alarm Interrupt Enable Bit</b> 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	<p><b>RTC Time Tick Interrupt Flag</b></p> <p>When RTC time tick event happened, TICKIF will be set to 1 and a time tick interrupt signal will be generated if TICKIEN (RTC_INTEN[1]) is enabled. Chip will also be waken up when time tick interrupt signal occurred if chip is running at Power-down mode.</p> <p>0 = Tick condition did not occur. 1 = Tick condition occurred.</p> <p><b>Note:</b> Writing 1 to clear this bit.</p>
[0]	ALMIF	<p><b>RTC Alarm Interrupt Flag</b></p> <p>When current RTC counter in RTC_TIME and RTC_CAL are matched RTC alarm settings in RTC_TALM and RTC_CALM, ALMIF will be set to 1 and an alarm interrupt signal will be generated if ALMIEN (RTC_INTEN[0]) is enabled. Chip will also be waken up when alarm interrupt signal occurred if chip is running at Power-down mode.</p> <p>0 = Alarm condition is not matched. 1 = Alarm condition is matched.</p> <p><b>Note:</b> Writing 1 to clear this bit.</p>

**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	<p><b>Time Tick Register</b></p> <p>These bits are used to select RTC time tick period for periodic time tick interrupt request.</p> <p>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.</p>

**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	<b>Mask 10-hour Time Digit of Alarm Setting (0~2)</b> <b>Note:</b> MTENHR function is only for 24-hour time scale mode.
[4]	MHR	<b>Mask 1-hour Time Digit of Alarm Setting (0~9)</b> <b>Note:</b> MHR function is only for 24-hour time scale mode.
[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)

**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	Reserved.
[5]	MTENYEAR	Mask 10-Year Calendar Digit of alarm setting (0~9)
[4]	MYEAR	Mask 1-Year Calendar Digit of alarm setting (0~9)
[3]	MTENMON	Mask 10-Month Calendar Digit of alarm setting (0~1)
[2]	MMON	Mask 1-Month Calendar Digit of alarm setting (0~9)
[1]	MTENDAY	Mask 10-Day Calendar Digit of alarm setting (0~3)
[0]	MDAY	Mask 1-Day Calendar Digit of alarm setting (0~9)



**RTC 32 kHz Oscillator Control Register (RTC\_LXTCTL)**

Register	Offset	R/W	Description	Reset Value
RTC_LXTCTL	RTC_BA+0x100	R/W	RTC 32 kHz Oscillator Control Register	0x0000_000E

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				GAIN			Reserved

Bits	Description	
[31:4]	Reserved	Reserved.
[3:1]	GAIN	<p><b>Oscillator Gain Option</b></p> <p>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.</p> <p>000 = L0 mode.                      001 = L1 mode.                      010 = L2 mode.                      011 = L3 mode.                      100 = L4 mode.                      101 = L5 mode.                      110 = L6 mode.                      111 = L7 mode (Default).</p>
[0]	Reserved	Reserved.

**RTC X32KO Pin Control Register (RTC\_LXTOCTL)**

Register	Offset	R/W	Description	Reset Value
RTC_LXTOCTL	RTC_BA+0x104	R/W	RTC X32KO Pin 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				CTLSEL	DOUT	OPMODE	

Bits	Description	
[31:4]	Reserved	Reserved.
[3]	CTLSEL	<p><b>I/O Pin State Backup Selection</b></p> <p>When low speed 32 kHz oscillator (LXT) is disabled, X32KO pin can be used as GPIO PF.0 function. User can program CTLSEL to decide X32KO (PF.0) I/O function is controlled by system power domain GPIO module or V<sub>BAT</sub> power domain RTC_LXTOCTL register.</p> <p>0 = X32KO (PF.0) pin I/O function is controlled by GPIO module.</p> <p>1 = X32KO (PF.0) pin I/O function is controlled by OPMODE and DOUT in RTC_LXTOCTL at V<sub>BAT</sub> power domain.</p> <p><b>Note:</b> CTLSEL will be set to 1 automatically by hardware when system power is turned off and RTC is at normal active state, ACTIVE (RTC_INIT[0]) is 1.</p>
[2]	DOUT	<p><b>I/O Pin Output Data</b></p> <p>0 = X32KO (PF.0) will drive low in output mode.</p> <p>1 = X32KO (PF.0) will drive high in output mode.</p>
[1:0]	OPMODE	<p><b>I/O Pin Operation Mode</b></p> <p>00 = X32KO (PF.0) is in Input mode without pull-up resistor.</p> <p>01 = X32KO (PF.0) is in Push-pull output mode.</p> <p>10 = X32KO (PF.0) is in Open-drain output mode.</p> <p>11 = X32KO (PF.0) is in Input mode with pull-up resistor.</p>

**RTC X32KI Pin Control Register (RTC\_LXTICTL)**

Register	Offset	R/W	Description	Reset Value
RTC_LXTICTL	RTC_BA+0x108	R/W	RTC X32KI Pin 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				CTLSEL	DOUT	OPMODE	

Bits	Description	
[31:4]	Reserved	Reserved.
[3]	CTLSEL	<p><b>I/O Pin State Backup Selection</b></p> <p>When low speed 32 kHz oscillator (LXT) is disabled, X32KO pin can be used as GPIO PF.1 function. User can program CTLSEL to decide X32KI (PF.1) I/O function is controlled by system power domain GPIO module or V<sub>BAT</sub> power domain RTC_LXTICTL register.</p> <p>0 = X32KI (PF.1) pin I/O function is controlled by GPIO module.</p> <p>1 = X32KI (PF.1) pin I/O function is controlled by OPMODE and DOUT in RTC_LXTICTL at V<sub>BAT</sub> power domain.</p> <p><b>Note:</b> CTLSEL will be set to 1 automatically by hardware when system power is turned off and RTC is at normal active state, ACTIVE (RTC_INIT[0]) is 1.</p>
[2]	DOUT	<p><b>IO Pin Output Data</b></p> <p>0 = X32KI (PF.1) will drive low in output mode.</p> <p>1 = X32KI (PF.1) will drive high in output mode.</p>
[1:0]	OPMODE	<p><b>I/O Pin Operation Mode</b></p> <p>00 = X32KI (PF.1) is in Input mode without pull-up resistor.</p> <p>01 = X32KI (PF.1) is in Push-pull output mode.</p> <p>10 = X32KI (PF.1) is in Open-drain output mode.</p> <p>11 = X32KI (PF.1) is in Input mode with pull-up resistor.</p>

**RTC PF2 Pin Control Register (RTC\_PF2CTL)**

Register	Offset	R/W	Description	Reset Value
RTC_PF2CTL	RTC_BA+0x10C	R/W	RTC PF.2 Pin 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				CTLSEL	DOUT	OPMODE	

Bits	Description	
[31:4]	Reserved	Reserved.
[3]	CTLSEL	<p><b>I/O Pin State Backup Selection</b></p> <p>User can program CTLSEL to decide GPIO PF.2 I/O function is controlled by system power domain GPIO module or V<sub>BAT</sub> power domain RTC_LXTICTL register.</p> <p>0 = GPIO PF.2 pin I/O function is controlled by GPIO module.</p> <p>1 = GPIO PF.2 pin I/O function is controlled by OPMODE and DOUT in RTC_PF2CTL at V<sub>BAT</sub> power domain.</p> <p><b>Note:</b> CTLSEL will be set to 1 automatically by hardware when system power is turned off and RTC is at normal active state, ACTIVE (RTC_INIT[0]) is 1.</p>
[2]	DOUT	<p><b>I/O Pin Output Data</b></p> <p>0 = GPIO PF.2 will drive low in output mode.</p> <p>1 = GPIO PF.2 will drive high in output mode.</p>
[1:0]	OPMODE	<p><b>I/O Pin Operation Mode</b></p> <p>00 = PF.2 is in Input mode without pull-up resistor.</p> <p>01 = PF.2 is in Push-pull output mode.</p> <p>10 = PF.2 is in Open-drain output mode.</p> <p>11 = PF.2 is in Input mode with pull-up resistor.</p>

**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	<b>Daylight Saving Back</b> 0= Daylight Saving Time function is not performed. 1= Daylight Saving Time function is performed.
[1]	SUBHR	<b>Subtract 1 Hour</b> 0 = No effect. 1 = Indicates RTC hour digit has been subtracted one hour for winter time change.
[0]	ADDHR	<b>Add 1 Hour</b> 0 = No effect. 1 = Indicates RTC hour digit has been added one hour for summer time change.

## 6.15 Smart Card Host Interface (SC)

### 6.15.1 Overview

The Smart Card Interface controller (SC controller) is based on ISO/IEC 7816-3 standard and fully compliant with PC/SC Specifications. It also provides status of card insertion/removal.

### 6.15.2 Features

- ISO-7816-3 T = 0, T = 1 compliant
- EMV2000 compliant
- Two ISO-7816-3 ports
- Separates receive/transmit 4 byte entry FIFO for data payloads
- Programmable transmission clock frequency
- Programmable receiver buffer trigger level
- Programmable guard time selection (11 ETU ~ 267 ETU)
- One 24-bit timer and two 8-bit timers for Answer to Request (ATR) and waiting times processing
- Supports auto direct / inverse convention function
- Supports transmitter and receiver error retry and error number limiting function
- Supports hardware activation sequence process, and the interval between PWR on and CLK start is configurable
- Supports hardware warm reset sequence process
- Supports hardware deactivation sequence process
- Supports hardware auto deactivation sequence when detected the card removal
- Supports UART mode
  - Full duplex, asynchronous communications
  - Separates receiving/transmitting 4 bytes entry FIFO for data payloads
  - Supports programmable baud rate generator
  - Supports programmable receiver buffer trigger level
  - Programmable transmitting data delay time between the last stop bit leaving the TX-FIFO and the de-assertion by setting EGT (SC\_EGT[7:0])
  - Programmable even, odd or no parity bit generation and detection
  - Programmable stop bit, 1- or 2- stop bit generation

### 6.15.3 Block Diagram

The SC clock control and block diagram are shown in Figure 6.15-1 and Figure 6.15-2. The SC controller is completely asynchronous design with two clock domains, PCLK and engine clock. Note that the PCLK should be higher than or equal to the frequency of engine clock.

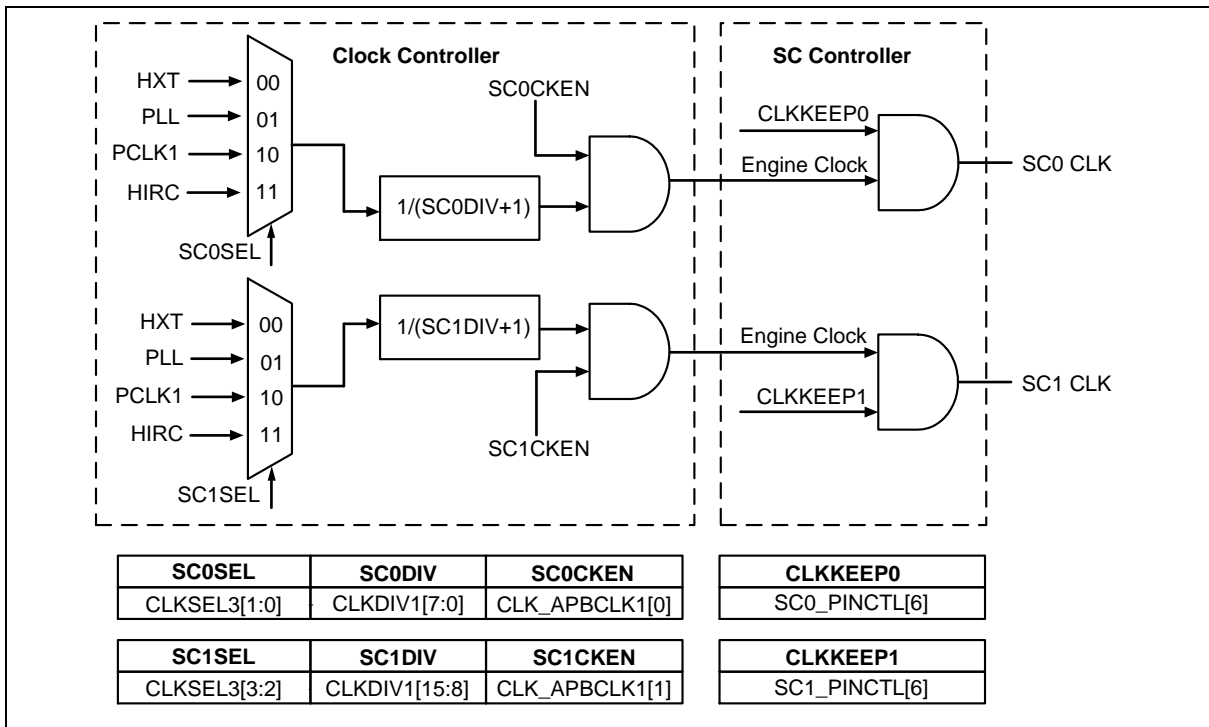


Figure 6.15-1 SC Clock Control Diagram

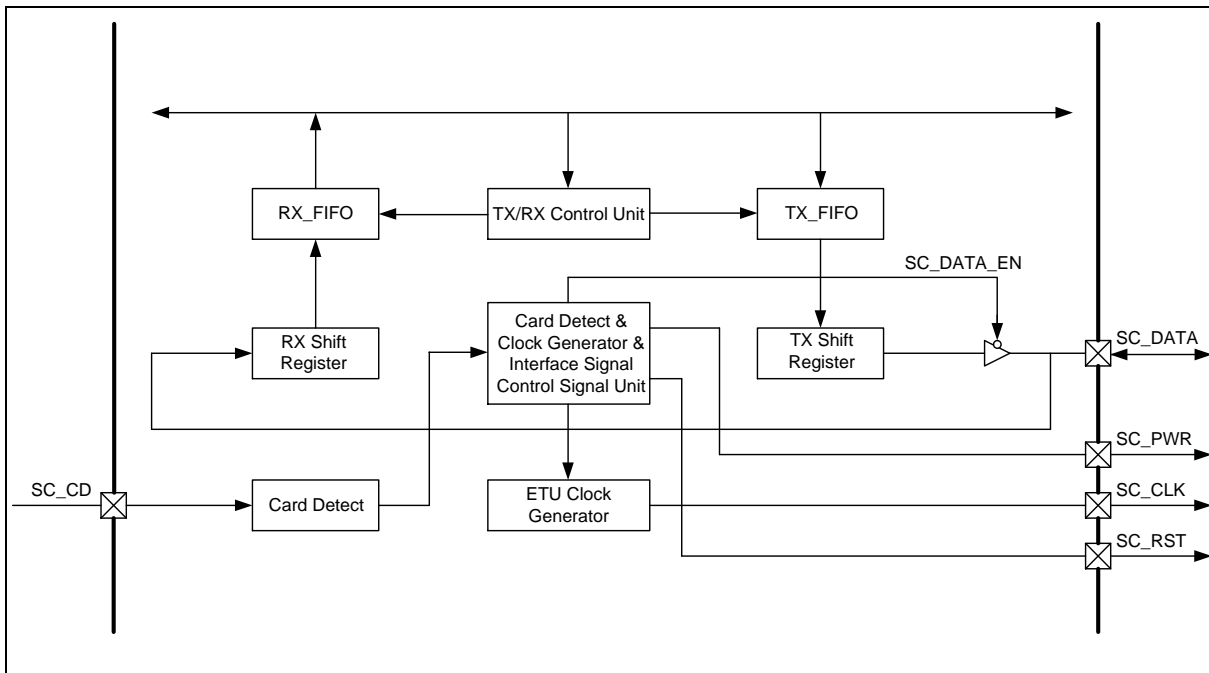


Figure 6.15-2 SC Controller Block Diagram

### 6.15.4 Basic Configuration

The SC Host Controller Pin description is shown in Table 6.15-1.

Pin	Type	Description
SC_DATA	Bi-direction	SC Host Controller DATA
SC_CD	Input	SC Host Controller Card Detect
SC_PWR	Output	SC Host Controller Power ON/OFF
SC_CLK	Output	SC Host Controller Clock
SC_RST	Output	SC Host Controller Reset

Table 6.15-1 SC Host Controller Pin Description

The UART Mode Pin description is shown in Table 6.15-2.

Pin	Type	Description
SC_DATA	Input	UART Receive Data
SC_CLK	Output	UART Transmit Data

Table 6.15-2 UART Pin Description

#### 6.15.4.1 Basic Configuration of SC0

- Clock Source Configuration
  - Select the source of SC0 peripheral clock on SC0SEL (CLK\_CLKSEL3[1:0]).
  - Select the clock divider number of SC0 peripheral clock on SC0DIV(CLK\_CLKDIV1[7:0]).
  - Enable SC0 peripheral clock in SC0CKEN (CLK\_APBCLK1[0]).
- Reset Configuration
  - Reset SC0 controller in SC0RST (SYS\_IPRST2[0]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
SC0	SC0_CLK	PC.1	MFP2
		PA.0, PE.11	MFP5
	SC0_DAT	PC.0	MFP1
		PA.1, PE.10	MFP5
	SC0_PWR	PC.3	MFP1
		PA.8	MFP4
		PA.3, PE.4, PE.8	MFP5
	SC0_RST	PC.2	MFP1
		PA.9	MFP4
PA.2, PB.1, PE.5, PE.9		MFP5	



	SC0_nCD	PC.4	MFP1
		PB.2, PE.0, PE.1	MFP5

6.15.4.2 Basic Configuration of SC1

- Clock Source Configuration
  - Select the source of SC1 peripheral clock on SC1SEL (CLK\_CLKSEL3[3:2]).
  - Select the clock divider number of SC1 peripheral clock on SC1DIV(CLK\_CLKDIV1[15:8]).
  - Enable SC1 peripheral clock in SC1CKEN (CLK\_APBCLK1[1]).
- Reset Configuration
  - Reset SC1 controller in SC1RST (SYS\_IPRST2[1]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
SC1	SC1_CLK	PA.11	MFP5
		PD.0	MFP6
	SC1_DAT	PA.10	MFP5
		PB.7	MFP6
	SC1_PWR	PA.9	MFP5
		PB.6	MFP6
	SC1_RST	PA.8	MFP5
		PB.5	MFP6
SC1_nCD	PF.5	MFP5	
	PB.4	MFP6	

6.15.5 Functional Description

Basically, the smart card interface acts as a half-duplex asynchronous communication port and its data format is composed of ten consecutive bits which is shown in Figure 6.15-3.

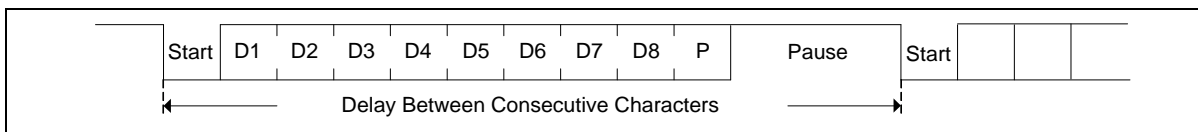


Figure 6.15-3 SC Data Character

The Smart Card Interface controller supports hardware activation, warm reset and deactivation sequence. The activation, warm reset and deactivation sequence are shown as follows.

6.15.5.1 Activation, Warm Reset and Deactivation Sequence

**Activation**

The activation sequence is shown in Figure 6.15-4.

1. Set SC\_RST to low by programming SCRST (SC\_PINCTL[1]) to '0'.
2. Set SC\_PWR at high level by programming PWREN (SC\_PINCTL[0]) to '1' and SC\_DATA at high level (reception mode) by programming SCDATA (SC\_PINCTL[9]) to '1'.
3. Enable SC\_CLK clock by programming CLKKEEP (SC\_PINCTL[6]) to '1'.
4. De-assert SC\_RST to high by programming SCRST (SC\_PINCTL[1]) to '1'.

The activation sequence can be controlled in two ways. The procedure is shown as follows:

**Software Timing Control:**

Set SC\_PINCTL and SC\_TMRCTLx (x=0, 1, 2) to process the activation sequence. SC\_PWR, SC\_CLK, SC\_RST and SC\_DATA pin state can be programmed by SC\_PINCTL. The programming method is shown in activation sequence. The activation sequence timing can be controlled by setting SC\_TMRCTLx (x=0, 1, 2). This programming procedure provides user with a flexible timing setting for activation sequence.

**Hardware Timing Control:**

Set ACTEN (SC\_ALTCTL[3]) to '1' and the interface will perform the activation sequence by hardware. The SC\_PWR to SC\_CLK start (T1) and SC\_CLK start to SC\_RST assert (T2) can be selected by programming INITSEL (SC\_ALTCTL[9:8]). This programming procedure provides user with a simple setting for activation sequence.

The following describes the activation control sequence in hardware activation mode:

1. Set activation timing by setting INITSEL (SC\_ALTCTL[9:8]).
2. Timer0 can be selected by setting TMRSEL (SC\_CTL[14:13]) is 11.
3. Set operation mode OPMODE (SC\_TMRCTL0[27:24]) to 0011 and give an Answer to Request (ATR) value by setting CNT (SC\_TMRCTL0[23:0]) register.
4. When hardware de-asserts SC\_RST to high, hardware will generate an initial end interrupt flag INITIF (SC\_INTSTS[8]) and inform to CPU at the same time if INITIEN (SC\_INTEN[8]) is 1. If the Timer0 decreases the counter to "0" (started from SC\_RST de-assert) and the card does not response ATR before that time, hardware will generate an interrupt flag TMR0IF (SC\_INTSTS[3]).

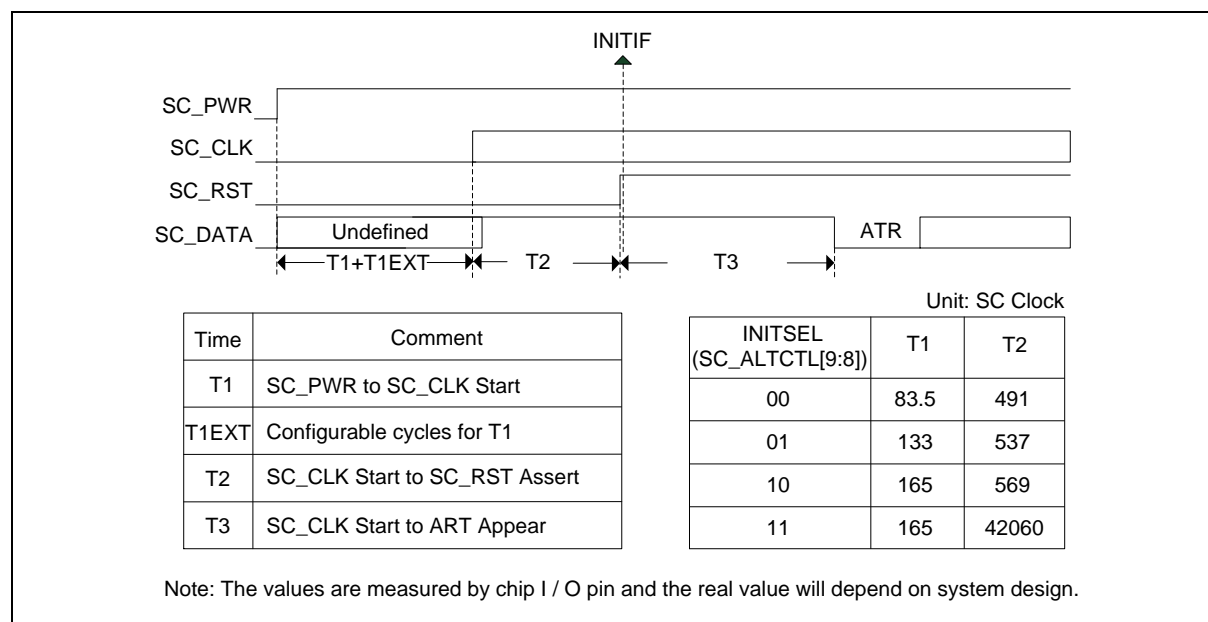


Figure 6.15-4 SC Activation Sequence

### Warm Reset

The warm reset sequence is shown in Figure 6.15-5.

1. Set SC\_RST to low by programming SCRST (SC\_PINCTL[1]) to '0'.
2. Set SC\_DATA to high by programming SCDATA (SC\_PINCTL[9]) to '1'.
3. Set SC\_RST to high by programming SCRST (SC\_PINCTL[1]) to '1'.

The warm reset sequence can be controlled in two ways. The procedure is shown as follows.

#### Software Timing Control

Set SC\_PINCTL and SC\_TMRCTLx (x=0, 1, 2) to process the warm reset sequence. The SC\_RST and SC\_DATA pin state can be programmed by SC\_PINCTL. The warm reset sequence timing can be controlled by setting SC\_TMRCTLx (x=0, 1, 2). This programming procedure provides user with a flexible timing setting for warm reset sequence.

#### Hardware Timing Control

Set WARSTEN (SC\_ALTCTL[4]) to '1' and the interface will perform the warm reset sequence by hardware. The SC\_RST to SC\_DATA reception mode (T4) and SC\_DATA reception mode to SC\_RST assert (T5) can be selected by programming INITSEL (SC\_ALTCTL[9:8]). This programming procedure provides user with a simple setting for warm reset sequence.

The following is the warm reset control sequence by hardware:

1. Set warm reset timing by setting INITSEL (SC\_ALTCTL[9:8]).
2. Select Timer0 by setting TMRSEL (SC\_CTL[14:13]) to 11.
3. Set operation mode OPMODE (SC\_TMRCTL0[27:24]) to 0011 and give an Answer to Request (ATR) value by setting CNT (SC\_TMRCTL0[23:0]) register.
4. Set CNTEN0 (SC\_ALTCTL[5]) and WARSTEN (SC\_ALTCTL[4]) to start counting.
5. When hardware de-asserts SC\_RST to high, hardware will generate an initial end interrupt flag INITIF (SC\_INTSTS[8]) and inform to CPU at the same time if INITIEN (SC\_INTEN[8]) is 1.
6. If the Timer0 decreases the counter to '0' (start from SC\_RST) and the card does not response ATR before that time, hardware will generate an interrupt flag TMR0IF (SC\_INTSTS[3]).

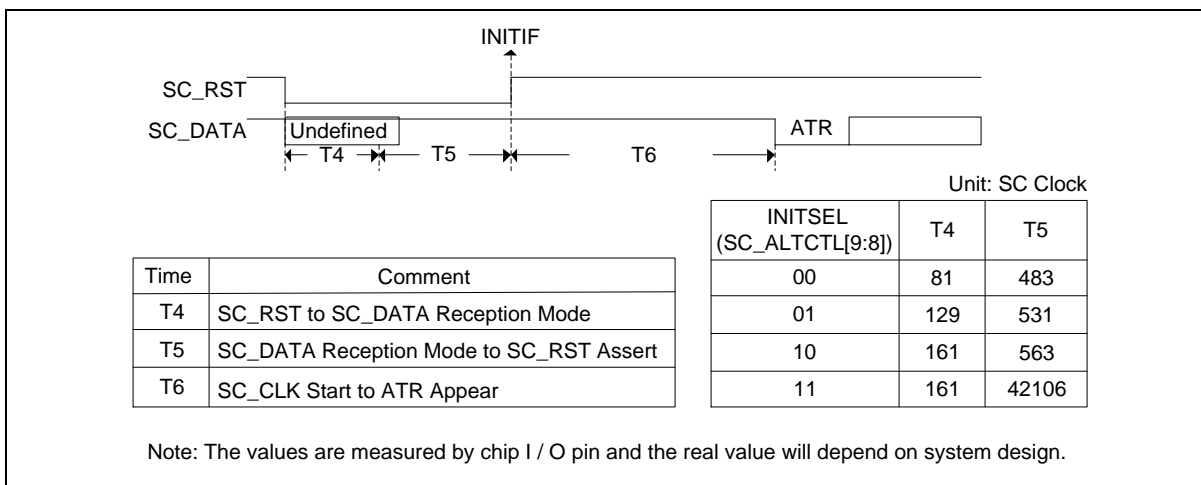


Figure 6.15-5 SC Warm Reset Sequence

### Deactivation

The deactivation sequence is showed in Figure 6.15-6.

1. Set SC\_RST to low by programming SCRST (SC\_PINCTL[1]) to '0'.
2. Stop SC\_CLK by programming CLKKEEP (SC\_PINCTL[6]) to '0'.
3. Set SC\_DATA to low by programming SCDATA (SC\_PINCTL[9]) to '0'.
4. Deactivate SC\_PWR by programming PWREN (SC\_PINCTL[0]) to '0'.

The deactivation sequence can be controlled in two ways. The procedure is shown as follows.

### Software Timing Control

Set SC\_PINCTL and SC\_TMRCTL0 to process the deactivation sequence. SC\_PWR, SC\_CLK, SC\_RST and SC\_DATA pin state can be programmed by SC\_PINCTL. The deactivation sequence timing can be controlled by setting SC\_TMRCTL0. This programming procedure provides user with a flexible timing setting for deactivation sequence.

### Hardware Timing Control

DACTEN (SC\_ALTCTL[2]) to '1' and the interface will perform the deactivation sequence by hardware. The Deactivation trigger to SC\_RST low (T7), SC\_RST low to SC\_CLK stop (T8) and SC\_CLK stop to SC\_PWR de-assert (T9) can be selected by programming INITSEL (SC\_ALTCTL[9:8]). This programming procedure provides user with a simple setting for deactivation sequence.

When hardware de-asserts SC\_PWR to low, the SC controller will generate an interrupt flag INITIF (SC\_INTSTS[8]) and inform to CPU at the same time if INITIEN (SC\_INTEN[8]) is 1.

The SC controller also supports auto deactivation sequence when the card removal detection is enabled by setting ADACEN (SC\_ALTCTL[11]).

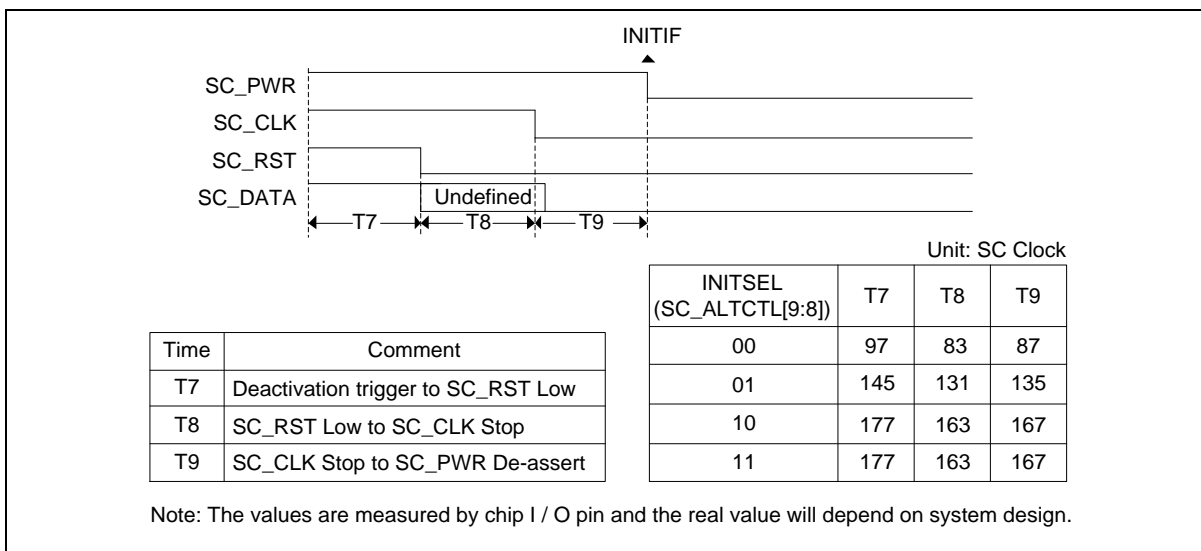


Figure 6.15-6 SC Deactivation Sequence

#### 6.15.5.2 Basic Operation Flow

Basic operation flow of smartcard can be referenced from ISO 7816-3 & EMV.

The program sequence flow is shown in Figure 6.15-7.

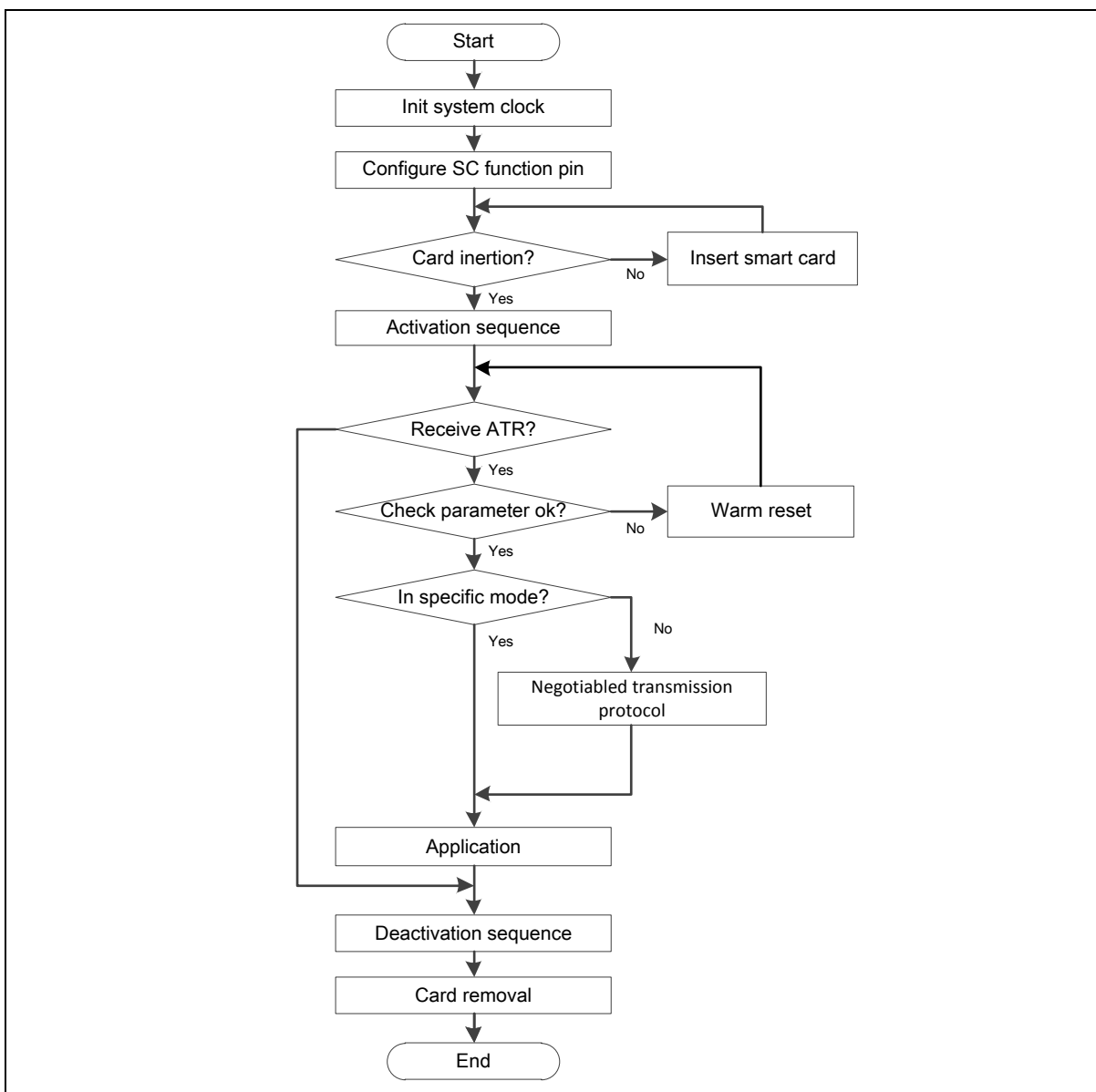


Figure 6.15-7 Basic Operation Flow

### 6.15.5.3 Initial Character TS

According to ISO 7816-3, the initial character TS has two possible patterns shown in Figure 6.15-8. If the TS pattern is 1100\_0000, it is inverse convention. When decoded by inverse convention, the conveyed byte is equal to 0x3F. If the TS pattern is 1101\_1100, it is direct convention. When decoded by direct convention, the conveyed byte is equal to 0x3B. User can set AUTOCEM (SC\_CTL[3]) and then the operating convention will be decided by hardware. User can also set CONSEL (SC\_CTL[5:4]) (set '00' or '11') to change the operating convention after SC received TS of Answer to Request (ATR).

If auto convention function is enabled by setting AUTOCEM (SC\_CTL[3]), the setting step must be done before Answer to Request (ATR) state and the received first data must be 0x3B or 0x3F. After hardware received first data and stored it at DAT (SC\_DAT[7:0]), the hardware will decide the convention mode and change the CONSEL (SC\_CTL[5:4]) automatically. If the received first data is neither 0x3B nor 0x3F, ACERRIF (SC\_INTSTS[10]) Auto Convention Error Interrupt Status Flag, will be set and the hardware will generate an interrupt signal to inform CPU if ACERRIEN (SC\_INTEN[10]) is 1.

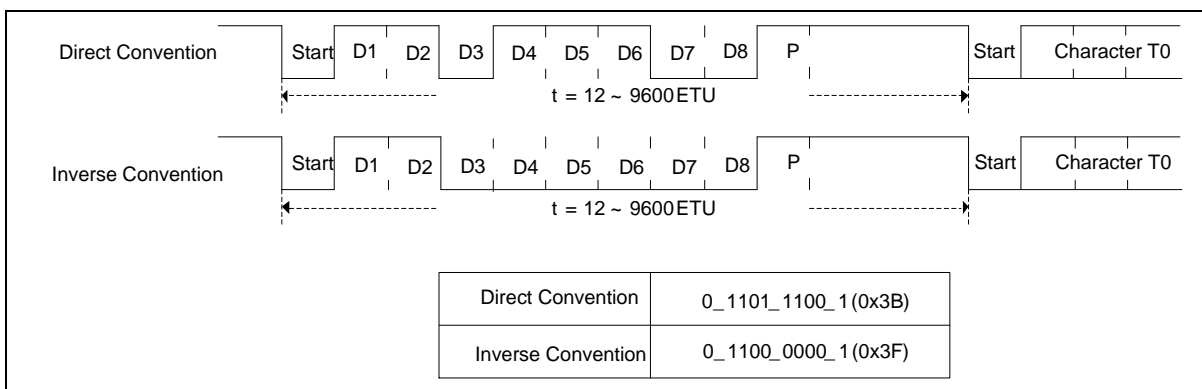


Figure 6.15-8 Initial Character TS

#### 6.15.5.4 Error Signal and Character Repetition

According to ISO7816-3 T=0 mode description, as shown in Figure 6.15-8, if the receiver receives a wrong parity bit, it will pull the SC\_DAT to low by 1.5 bit period to inform the transmitter parity error. Then the transmitter will retransmit the character. The SC interface controller supports hardware error detection function in receiver and supports hardware re-transmit function in transmitter. User can enable re-transmit function by setting TXRTYEN (SC\_CTL[23]). User can also define the retry (re-transmit) number limitation in TXRTY (SC\_CTL[22:20]). The re-transmit number is up to (TXRTY + 1) and if the re-transmit number is equal to (TXRTY + 1), TXOVERR (SC\_STATUS[30]) flag will be set by hardware and if TERRIEN (SC\_INTEN[2]) is enabled, SC controller will generate a transfer error interrupt signal to CPU.

User can also enable received retry function by setting RXRTYEN (SC\_CTL[19]) and define the received retry number limitation in RXRTY (SC\_CTL[18:16]). The receiver retry number is up to (RXRTY + 1), if the number of received errors by receiver is equal to (RXRTY + 1), receiver will receive this error data to buffer and RXOVERR (SC\_STATUS[22]) flag will be set by hardware. If TERRIEN (SC\_INTEN[2]) is enabled, SC controller will also generate a transfer error interrupt signal to CPU. Figure 6.15-9 shows the SC error signal timing of receiver and transmitter.

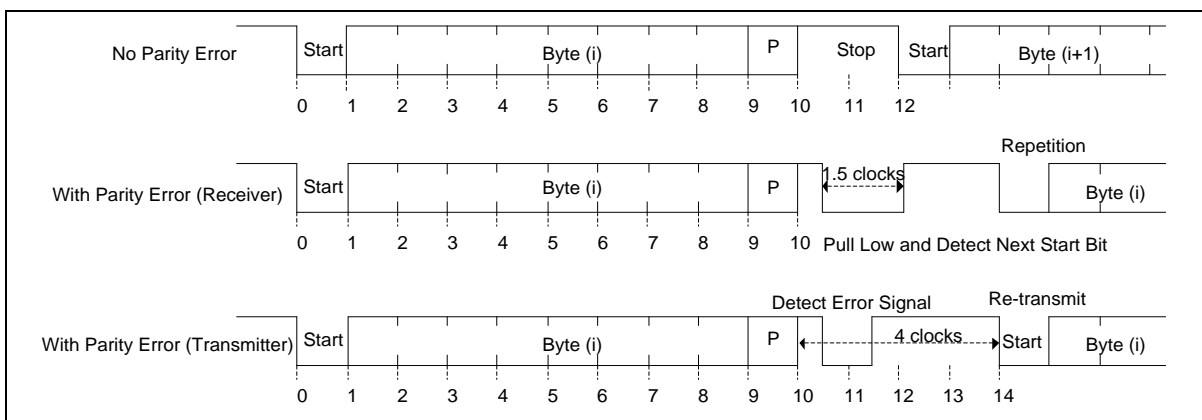


Figure 6.15-9 SC Error Signal

#### 6.15.5.5 Internal Timer Operation Mode

The SC controller includes a 24-bit down counter and two 8-bit down counters. These counters help the controller in processing different real-time interval. Each counter can be set to start counting when the start enable bit CNTENx, x=0, 1, 2 (SC\_ALTCTL[7:5]) has been enabled or a START bit signal has been detected in receiver or transmitter.

The following is the programming flow:

1. Enable counter by setting TMRSEL (SC\_CTL[14:13]) to 11.
2. Select operation mode OPMODE (SC\_TMRCTLx[27:24], x=0, 1, 2).
3. Give a count value CNT for Timer0, Timer1 and Timer2 by setting SC\_TMRCTL0[23:0], SC\_TMRCTL1[7:0] and SC\_TMRCTL2[7:0].
4. Set CNTEN0 (SC\_ALTCTL[5]), CNTEN1 (SC\_ALTCTL[6]) and CNTEN2 (SC\_ALTCTL[7]) to start counting.

The SC\_TMRCTL0, SC\_TMRCTL1 and SC\_TMRCTL2 timer operation mode are listed in Table 6.15-3.

**Note1:** Only SC\_TMRCTL0 of Timer0 supports mode 0011.

**Note2:** START bit can only be detected when Tx or Rx is idle or finish the last transmission.

OPMODE (SC_TMRCTLx[27:24]), X=0, 1, 2)	Operation Mode Description	
0000	The down counter is started when CNTENx (SC_ALTCTL[7:5]) enabled and ended when counter time-out. The time-out counter value will be CNT (SC_TMRCTL0[23:0], SC_TMRCTL1[7:0], SCn_TMRCTL2[7:0]) + 1.	
	Start	Start counting when CNTENx (SC_ALTCTL[7:5]) enabled.
	End	When the down counter equals 0, hardware will set TMRxIF (SC_INTSTS[5:3]) and clear CNTENx (SC_ALTCTL[7:5]) automatically.
0001	The down counter is started when the first START bit (reception or transmission) detected and ended when counter time-out. It takes 2 ETU to detect first START bit after writing data to Tx or receiving data from Rx. The time-out counter value will be CNT (SC_TMRCTL0[23:0], SC_TMRCTL1[7:0], SC_TMRCTL2[7:0]) + 1.	
	Start	Start counting when the first START bit (reception or transmission) detected after CNTENx (SC_ALTCTL[7:5]) set to 1.
	End	When the down counter equals 0, hardware will set TMRxIF (SC_INTSTS[5:3]) and clear CNTENx (SC_ALTCTL[7:5]) automatically.
0010	The down counter is started when the first START bit (reception) detected and ended when counter time-out. It takes 2 ETU to detect first START bit after receiving data from Rx. The time-out value will be CNT (SC_TMRCTL0[23:0], SC_TMRCTL1[7:0], SC_TMRCTL2[7:0]) + 1.	
	Start	Start counting when the first START bit (reception) detected bit after CNTENx (SC_ALTCTL[7:5]) set to 1.
	End	When the down counter equals 0, hardware will set TMRxIF (SC_INTSTS[5:3]) and clear CNTENx (SC_ALTCTL[7:5]) automatically.
0011	The down counter is only used for hardware activation, warm reset sequence to measure ATR timing. The timing starts when SC_RST de-assertion and ends when ATR response received or time-out. If the counter decreases to 0 before ATR response received, hardware will set TMR0IF (SC_INTSTS[3]) and generate an interrupt signal to inform CPU if TMR0IEN (SC_INTEN[3]) enabled. The time-out value will be CNT (SC_TMRCTL0[23:0]) + 1.	
	Start	Start counting when SC_RST de-assertion after CNTEN0 (SC_ALTCTL[5]) set to 1. It is only used for hardware activation, warm reset mode.
	End	When the down counter equals 0 before ATR response received, hardware will set TMR0IF and clear CNTEN0 (SC_ALTCTL[5]) automatically. When ATR received and down counter does not equal to 0, hardware will clear CNTEN0 (SC_ALTCTL[5]) automatically.

0100	Start	Start down counter counting when CNTENx (SC_ALTCTL[7:5]) enabled.
	Recount & reload	When ACTSTSx (SC_ALTCTL[15:13]) is 1, user can change CNT (SC_TMRCTL0[23:0], SC_TMRCTL1[7:0], SC_TMRCTL2[7:0]) value at any time. It will reload the last value which is filled into the CNT(SC_TMRCTL0[23:0], SC_TMRCTL1[7:0], SC_TMRCTL2[7:0]) before the counter count to 0. Only when the down counter equals 0, counter reload the CNT (SC_TMRCTL0[23:0], SC_TMRCTL1[7:0], SC_TMRCTL2[7:0]) value and start to recount.
	Interrupt	If the counter decreases to 0, hardware will set TMRxIF (SC_INTSTS[5:3]) and generate an interrupt signal to inform CPU if TMRxIEN (SC_INTEN[5:3]) enabled. The time-out value will be CNT (SC_TMRCTL0[23:0], SC_TMRCTL1[7:0], SC_TMRCTL2[7:0]) + 1.
	End	The down counter stopped when user clears CNTENx (SC_ALTCTL[7:5]) bit.
0101	Start	The down counter is started when the first START bit (reception or transmission) detected after CNTENx (SC_ALTCTL[7:5]) set to 1. It takes 2 ETU to detect START bit after writing data to Tx or receiving data from Rx.
	Reload	When ACTSTSx (SC_ALTCTL[15:13]) is 1, user can change CNT (SC_TMRCTL0[23:0], SC_TMRCTL1[7:0], SC_TMRCTL2[7:0]) value at any time. It will reload the last value which is filled into the CNT(SC_TMRCTL0[23:0], SC_TMRCTL1[7:0], SC_TMRCTL2[7:0]) before the counter count to 0. Only when the down counter equals 0, counter will reload the CNT (SC_TMRCTL0[23:0], SC_TMRCTL1[7:0], SC_TMRCTL2[7:0]) value.
	Recount	After down counter reloads the CNT value, timer counter starts to recount only when the next START bit is detected.
	Interrupt	If the counter decreases to 0, hardware will set TMRxIF (SC_INTSTS[5:3]) and generate an interrupt signal to inform CPU if TMRxIEN (SC_INTEN[5:3]) enabled. The time-out value will be CNT (SC_TMRCTL0[23:0], SC_TMRCTL1[7:0], SC_TMRCTL2[7:0]) + 1.
	End	The down counter stopped when user clears CNTENx (SC_ALTCTL[7:5]) bit.
0110	Start	The down counter is started when the first START bit (reception) detected after CNTENx (SC_ALTCTL[7:5]) set to 1. It takes 2 ETU to detect START bit after writing data to Tx or receiving data from Rx.
	Reload	When ACTSTSx (SC_ALTCTL[15:13]) is 1, user can change CNT (SC_TMRCTL0[23:0], SC_TMRCTL1[7:0], SC_TMRCTL2[7:0]) value at any time. It will reload the last value which is filled into the CNT(SC_TMRCTL0[23:0], SC_TMRCTL1[7:0], SC_TMRCTL2[7:0]) before the counter counts to 0. Only when the down counter equals 0, counter will reload the CNT (SC_TMRCTL0[23:0], SC_TMRCTL1[7:0], SC_TMRCTL2[7:0]) value.
	Recount	After the down counter reloads the CNT value, timer counter starts to recount only when the next START bit is detected.
	Interrupt	If the counter decreases to 0, hardware will set TMRxIF (SC_INTSTS[5:3]) and generate an interrupt signal to inform CPU if TMRxIEN (SC_INTEN[5:3]) enabled. The time-out value will be CNT (SC_TMRCTL0[23:0], SC_TMRCTL1[7:0], SC_TMRCTL2[7:0]) + 1.
	End	The down counter stopped when user clears CNTENx (SC_ALTCTL[7:5]) bit.
0111	Start	The down counter is started when the first START bit (reception or transmission) detected after CNTENx (SC_ALTCTL[7:5]) set to 1. It takes 2 ETU to detect START bit after writing data to Tx or receiving data from Rx.
	Reload &recount	Only when the next START bit is detected, counter will reload the new value of CNT (SC_TMRCTL0[23:0], SC_TMRCTL1[7:0], SC_TMRCTL2[7:0]) and recount.
	Interrupt	If the counter decreases to 0 before the next START bit detected, hardware will set TMRxIF (SC_INTSTS[5:3]) and generate an interrupt signal to inform CPU if TMRxIEN



		(SC_INTEN[5:3]) enabled. The time-out value will be CNT (SC_TMRCTL0[23:0], SC_TMRCTL1[7:0], SC_TMRCTL2[7:0]) + 1.
	End	The down counter stopped when user clears CNTENx (SC_ALTCTL[7:5]) bit.
1111	Start	The down counter starts counting when user sets CNTENx (SC_ALTCTL[7:5]) bit and it will count to time-out.
	Reload &recount	Only when the next START bit is detected, counter will reload the new value of CNT (SC_TMRCTL0[23:0], SC_TMRCTL1[7:0], SC_TMRCTL2[7:0]) and recount.
	Interrupt	If the counter decreases to 0 before the next START bit detected, hardware will generate time-out interrupt flag TMRxIF (SC_INTSTS[5:3]). The time-out value will be CNTx (SC_TMRCTL0[23:0], SC_TMRCTL1[7:0], SC_TMRCTL2[7:0]) + 1.
	End	The down counter stopped when user clears CNTENx (SC_ALTCTL[7:5]) bit.

Table 6.15-3 Timer0/Timer1/Timer2 Operation Mode

6.15.5.6 Block Guard Time and Extra Guard Time

Block Guard Time (BGT) means the minimum interval between the leading edges of two consecutive characters between different transfer directions which are shown in Figure 6.15-10 and Figure 6.15-11. This field indicates the counter for the bit length of block guard time. According to ISO7816-3, in T = 0 mode, user must fill 15 (real block guard time = 16.5) to this field; in T = 1 mode, user must fill 21 (real block guard time = 22.5) to it.

In transmit direction, the smart card sends data to smart card host controller, first. After the period is greater than BGT (SC\_CTL[12:8]), the smart card host controller begin to send the data.

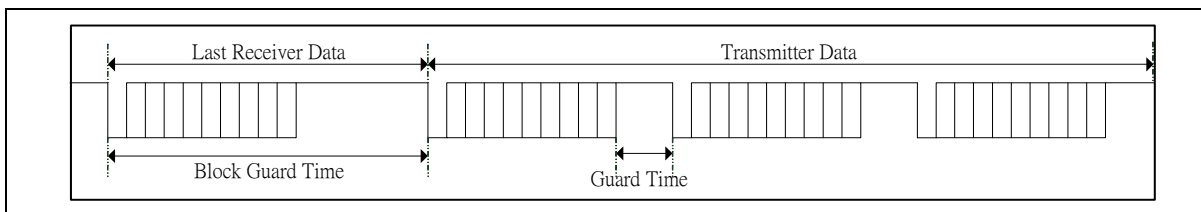


Figure 6.15-10 Transmit Direction Block Guard Time Operation

In receive direction, the smart card host controller sends data to smart card, first. If the smart card responses data to smart card host controller at the time which is less than BGT (SC\_CTL[12:8]), the block guard time interrupt BGTIF (SC\_INTSTS[6]) is generated when RXBGTEN (SC\_ALTCTL[12]) is enabled.

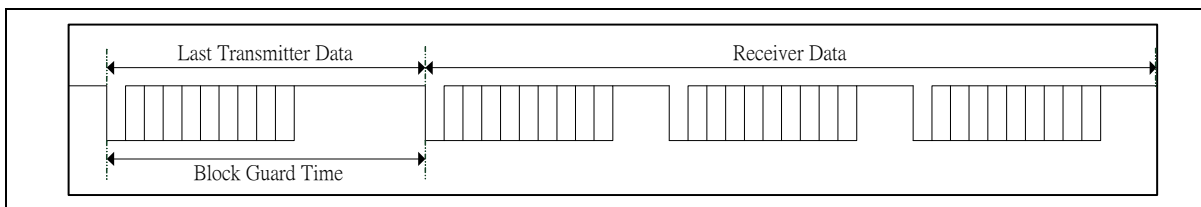


Figure 6.15-11 Receive Direction Block Guard Time Operation

Extra Guard Time (EGT) is adjustable in EGT (SC\_EGT[7:0]), it only affects the data transmitted by smart card interface, the format is shown as Figure 6.15-12.

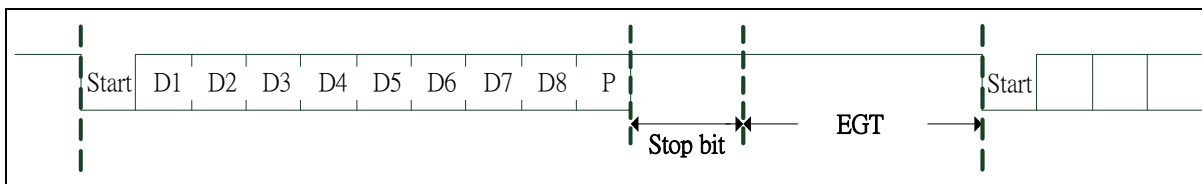


Figure 6.15-12 Extra Guard Time Operation

### 6.15.5.7 UART Mode

When the UARTEN (SC\_UARTCTL[0]) bit is set, the Smart Card Interface controller can also be used as basic UART function. The following is the program example for UART mode.

#### Programming example

1. Set UARTEN (SC\_UARTCTL[0]) bit to enter UART mode.
2. Do user reset by setting RXRST (SC\_ALTCTL[1]) and TXRST(SC\_ALTCTL[0]) bit to ensure that all state machine return idle state.
3. In UART mode, CONSEL (SC\_CTL[5:4]) and AUTOZEN (SC\_CTL[3]) fields must be set to 0.
4. Select the UART baud rate by setting ETURDIV (SC\_ETUCTL[11:0]) fields. For example, if smartcard module clock is 12 MHz and target baud rate is 115200 bps, ETURDIV should fill with  $((12000000 / 115200) - 1)$ .
5. Select the data format include data length by setting WLS (SC\_UARTCTL[5:4]), parity format by setting OPE (SC\_UARTCTL[7]) and PBOFF (SC\_UARTCTL[6]), and stop bit length by setting NSB (SC\_CTL[15]).
6. Select the receiver buffer number trigger level by setting RXTRGLV (SC\_CTL[7:6]) field and the receiver buffer time-out interval by setting RFTM (SC\_RXTOUT[8:0]) field.
7. Write the SC\_DAT (SC\_DAT[7:0]) (Tx send) or read the SC\_DAT (SC\_DAT[7:0]) (Rx receive) can perform UART function.

### 6.15.6 Register Map

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

Register	Offset	R/W	Description	Reset Value
<b>SC Base Address:</b>				
<b>SCx_BA = 0x4019_0000 + (0x4000 * x)</b>				
<b>x= 0, 1</b>				
<b>SC_DAT</b>	SCx_BA+0x00	R/W	SC Receive/Transmit Holding Buffer Register	Undefined
<b>SC_CTL</b>	SCx_BA+0x04	R/W	SC Control Register	0x0000_0000
<b>SC_ALTCTL</b>	SCx_BA+0x08	R/W	SC Alternate Control Register	0x0000_0000
<b>SC_EGT</b>	SCx_BA+0x0C	R/W	SC Extra Guard Time Register	0x0000_0000
<b>SC_RXTOUT</b>	SCx_BA+0x10	R/W	SC Receive Buffer Time-out Counter Register	0x0000_0000
<b>SC_ETUCTL</b>	SCx_BA+0x14	R/W	SC Element Time Unit Control Register	0x0000_0173
<b>SC_INTEN</b>	SCx_BA+0x18	R/W	SC Interrupt Enable Control Register	0x0000_0000
<b>SC_INTSTS</b>	SCx_BA+0x1C	R/W	SC Interrupt Status Register	0x0000_0002
<b>SC_STATUS</b>	SCx_BA+0x20	R/W	SC Transfer Status Register	0x0000_x202
<b>SC_PINCTL</b>	SCx_BA+0x24	R/W	SC Pin Control State Register	0x0000_00x0
<b>SC_TMRCTL0</b>	SCx_BA+0x28	R/W	SC Timer0 Control Register	0x0000_0000
<b>SC_TMRCTL1</b>	SCx_BA+0x2C	R/W	SC Timer1 Control Register	0x0000_0000
<b>SC_TMRCTL2</b>	SCx_BA+0x30	R/W	SC Timer2 Control Register	0x0000_0000
<b>SC_UARTCTL</b>	SCx_BA+0x34	R/W	SC UART Mode Control Register	0x0000_0000
<b>SC_TMRDAT0</b>	SCx_BA+0x38	R	SC Timer0 Current Data Register	0x0000_07FF
<b>SC_TMRDAT12</b>	SCx_BA+0x3C	R	SC Timer1/2 Current Data Register	0x0000_7F7F
<b>SC_ACTCTL</b>	SCx_BA+0x4C	R/W	SC Activation Control Register	0x0000_0000

### 6.15.7 Register Description

#### SC Receive/Transmit Holding Buffer Register (SC\_DAT)

Register	Offset	R/W	Description	Reset Value
SC_DAT	SCx_BA+0x00	R/W	SC Receive/Transmit Holding 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							
7	6	5	4	3	2	1	0
DAT							

Bits	Description	
[31:8]	Reserved	Reserved.
[7:0]	DAT	<p><b>Receive/Transmit Holding Buffer</b></p> <p>Write Operation: By writing data to DAT, the SC will send out an 8-bit data.</p> <p>Read Operation: By reading DAT, the SC will return an 8-bit received data.</p> <p><b>Note:</b> If SCEN (SC_CTL[0]) is not enabled, DAT cannot be programmed.</p>

**SC Control Register (SC\_CTL)**

Register	Offset	R/W	Description	Reset Value
SC_CTL	SCx_BA+0x04	R/W	SC Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	SYNC	Reserved			CDLV	CDBSEL	
23	22	21	20	19	18	17	16
TXRTYEN	TXRTY			RXRTYEN	RXRTY		
15	14	13	12	11	10	9	8
NSB	TMRSEL		BGT				
7	6	5	4	3	2	1	0
RXTRGLV		CONSEL		AUTOCEN	TXOFF	RXOFF	SCEN

Bits	Description	
[31]	Reserved	Reserved.
[30]	SYNC	<p><b>SYNC Flag Indicator (Read Only)</b></p> <p>Due to synchronization, user should check this bit before writing a new value to RXRTY and TXRTY fields.</p> <p>0 = Synchronizing is completion, user can write new data to RXRTY and TXRTY.</p> <p>1 = Last value is synchronizing.</p>
[29:27]	Reserved	Reserved.
[26]	CDLV	<p><b>Card Detect Level Selection</b></p> <p>0 = When hardware detects the card detect pin (SC_CD) from high to low, it indicates a card is detected.</p> <p>1 = When hardware detects the card detect pin (SC_CD) from low to high, it indicates a card is detected.</p> <p><b>Note:</b> User must select card detect level before Smart Card controller enabled.</p>
[25:24]	CDBSEL	<p><b>Card Detect De-bounce Selection</b></p> <p>This field indicates the card detect de-bounce selection.</p> <p>00 = De-bounce sample card insert once per 384 (128 * 3) SC modue clocks and de-bounce sample card removal once per 128 SC modue clocks.</p> <p>Other configurations are reserved.</p>
[23]	TXRTYEN	<p><b>TX Error Retry Enable Bit</b></p> <p>This bit enables transmitter retry function when parity error has occurred.</p> <p>0 = TX error retry function Disabled.</p> <p>1 = TX error retry function Enabled.</p>
[22:20]	TXRTY	<p><b>TX Error Retry Count Number</b></p> <p>This field indicates the maximum number of transmitter retries that are allowed when parity error has occurred.</p> <p><b>Note1:</b> The real retry number is TXRTY + 1, so 8 is the maximum retry number.</p> <p><b>Note2:</b> This field cannot be changed when TXRTYEN enabled. The change flow is to disable TXRTYEN first and then fill in new retry value.</p>

[19]	RXRTYEN	<p><b>RX Error Retry Enable Bit</b></p> <p>This bit enables receiver retry function when parity error has occurred.</p> <p>0 = RX error retry function Disabled.</p> <p>1 = RX error retry function Enabled.</p> <p><b>Note:</b> User must fill in the RXRTY value before enabling this bit.</p>
[18:16]	RXRTY	<p><b>RX Error Retry Count Number</b></p> <p>This field indicates the maximum number of receiver retries that are allowed when parity error has occurred</p> <p><b>Note1:</b> The real retry number is RXRTY + 1, so 8 is the maximum retry number.</p> <p><b>Note2:</b> This field cannot be changed when RXRTYEN enabled. The change flow is to disable RXRTYEN first and then fill in new retry value.</p>
[15]	NSB	<p><b>Stop Bit Length</b></p> <p>This field indicates the length of stop bit.</p> <p>0 = The stop bit length is 2 ETU.</p> <p>1 = The stop bit length is 1 ETU.</p> <p><b>Note1:</b> The default stop bit length is 2. SC and UART adopts NSB to program the stop bit length.</p> <p><b>Note2:</b> In UART mode, RX can receive the data sequence in 1 stop bit or 2 stop bits with NSB is set to 0.</p>
[14:13]	TMRSEL	<p><b>Timer Channel Selection</b></p> <p>00 = All internal timer function Disabled.</p> <p>.</p> <p>11 = Internal 24-bit Timer0 and two 8-bit Timer0 and Timer1 are enabled. User can configure them by setting SC_TMRCTL0[23:0], SC_TMRCTL1[7:0] and SC_TMRCTL2[7:0].</p> <p>Other configurations are reserved</p>
[12:8]	BGT	<p><b>Block Guard Time (BGT)</b></p> <p>Block guard time means the minimum interval between the leading edges of two consecutive characters between different transfer directions. This field indicates the counter for the bit length of block guard time. According to ISO7816-3, in T = 0 mode, user must fill 15 (real block guard time = 16.5) to this field; in T = 1 mode, user must fill 21 (real block guard time = 22.5) to it.</p> <p><b>Note:</b> The real block guard time is BGT + 1.</p>
[7:6]	RXTRGLV	<p><b>Rx Buffer Trigger Level</b></p> <p>When the number of bytes in the receiving buffer equals the RXTRGLV, the RDAIF (SC_INTSTS[0]) will be set. If RDAIEN (SC_INTEN[0]) is enabled, an interrupt signal will be generated to inform CPU.</p> <p>00 = Rx Buffer Trigger Level with 01 bytes.</p> <p>01 = Rx Buffer Trigger Level with 02 bytes.</p> <p>10 = Rx Buffer Trigger Level with 03 bytes.</p> <p>11 = Reserved.</p>
[5:4]	CONSEL	<p><b>Convention Selection</b></p> <p>00 = Direct convention.</p> <p>01 = Reserved.</p> <p>10 = Reserved.</p> <p>11 = Inverse convention.</p> <p><b>Note:</b> If AUTOGEN (SC_CTL[3]) is enabled, this field is ignored.</p>
[3]	AUTOGEN	<p><b>Auto Convention Enable Bit</b></p> <p>This bit is used to enable auto convention function.</p>

		<p>0 = Auto-convention Disabled. 1 = Auto-convention Enabled.</p> <p><b>Note1:</b> If user enables auto convention function, the setting step must be done before Answer to Reset (ATR) state and the first data must be 0x3B or 0x3F. After hardware received first data and stored it at buffer, hardware will decided the convention and change the CONSEL (SC_CTL[5:4]) bits automatically when received first data is 0x3B or 0x3F. If received first byte is 0x3B, TS is direct convention, CONSEL (SC_CTL[5:4]) will be set to 00 automatically, otherwise the TS is inverse convention, and CONSEL (SC_CTL[5:4]) will be set to 11.</p> <p><b>Note2:</b> If the first data is not 0x3B or 0x3F, hardware will set ACERRIF (SC_INTSTS[10]) and generate an interrupt signal to inform CPU when ACERRIEN (SC_INTEN[10]) is enabled.</p>
[2]	<b>TXOFF</b>	<p><b>TX Transition Disable Bit</b> This bit is used to disable Tx transmit function. 0 = The transceiver Enabled. 1 = The transceiver Disabled.</p>
[1]	<b>RXOFF</b>	<p><b>RX Transition Disable Bit</b> This bit is used to disable Rx receive function. 0 = The receiver Enabled. 1 = The receiver Disabled. <b>Note:</b> If AUTOcen (SC_CTL[3]) is enabled, this field is ignored.</p>
[0]	<b>SCEN</b>	<p><b>SC Controller Enable Bit</b> Set this bit to 1 to enable SC operation function. 0 = SC will force all transition to IDLE state. 1 = SC controller is enabled and all function can work correctly. <b>Note:</b> SCEN must be set to 1 before filling in other SC registers, or smart card will not work properly.</p>

**SC Alternate Control Register (SC\_ALTCTL)**

Register	Offset	R/W	Description	Reset Value
SC_ALTCTL	SCx_BA+0x08	R/W	SC Alternate Control Register	0x0000_0000

31	30	29	28	27	26	25	24
SYNC		Reserved					
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
ACTSTS2	ACTSTS1	ACTSTS0	RXBGTEN	ADACEN	Reserved	INITSEL	
7	6	5	4	3	2	1	0
CNTEN2	CNTEN1	CNTEN0	WARSTEN	ACTEN	DACTEN	RXRST	TXRST

Bits	Description
[31]	<p><b>SYNC</b></p> <p><b>SYNC Flag Indicator (Read Only)</b>                      Due to synchronization, user should check this bit when writing a new value to SC_ALTCTL register.                      0 = Synchronizing is completion, user can write new data to SC_ALTCTL register.                      1 = Last value is synchronizing.</p>
[30:16]	<p><b>Reserved</b></p> <p>Reserved.</p>
[15]	<p><b>ACTSTS2</b></p> <p><b>Internal Timer2 Active Status (Read Only)</b>                      This bit indicates the timer counter status of timer2.                      0 = Timer2 is not active.                      1 = Timer2 is active.  <b>Note:</b> Timer2 is active does not always mean timer2 is counting the CNT (SC_TMRCTL2[7:0]).</p>
[14]	<p><b>ACTSTS1</b></p> <p><b>Internal Timer1 Active Status (Read Only)</b>                      This bit indicates the timer counter status of timer1.                      0 = Timer1 is not active.                      1 = Timer1 is active.  <b>Note:</b> Timer1 is active does not always mean timer1 is counting the CNT (SC_TMRCTL1[7:0]).</p>
[13]	<p><b>ACTSTS0</b></p> <p><b>Internal Timer0 Active Status (Read Only)</b>                      This bit indicates the timer counter status of timer0.                      0 = Timer0 is not active.                      1 = Timer0 is active.  <b>Note:</b> Timer0 is active does not always mean timer0 is counting the CNT (SC_TMRCTL0[23:0]).</p>
[12]	<p><b>RXBGTEN</b></p> <p><b>Receiver Block Guard Time Function Enable Bit</b>                      This bit enables the receiver block guard time function.                      0 = Receiver block guard time function Disabled.                      1 = Receiver block guard time function Enabled.</p>



[11]	ADACEN	<p><b>Auto Deactivation When Card Removal</b></p> <p>This bit is used for enable hardware auto deactivation when smart card is removed.</p> <p>0 = Auto deactivation Disabled.</p> <p>1 = Auto deactivation Enabled.</p> <p><b>Note:</b> When the card is removed, hardware will stop any process and then do deactivation sequence if this bit is set. If auto deactivation process completes, hardware will set INITIF (SC_INTSTS[8]) also.</p>
[10]	Reserved	Reserved.
[9:8]	INITSEL	<p><b>Initial Timing Selection</b></p> <p>This field indicates the initial timing of hardware activation, warm-reset or deactivation. The unit of initial timing is SC module clock.</p> <p>Activation: refer to SC Activation Sequence in Figure 6.15-4.</p> <p>Warm-reset: refer to Warm-Reset Sequence in Figure 6.15-5.</p> <p>Deactivation: refer to Deactivation Sequence in Figure 6.15-6.</p> <p><b>Note:</b> When setting activation and warm reset in Timer0 operation mode 0011, it may have deviation at most 128 SC module clock cycles.</p>
[7]	CNTEN2	<p><b>Internal Timer2 Start Enable Bit</b></p> <p>This bit enables Timer 2 to start counting. User can fill 0 to stop count and set 1 to reload and start count. The counter unit is ETU base.</p> <p>0 = Stop counting.</p> <p>1 = Start counting.</p> <p><b>Note1:</b> This field is used for internal 8-bit timer when TMRSEL (SC_CTL[14:13]) is 11 only. Do not fill in CNTEN2 when TMRSEL (SC_CTL[14:13]) is not equal to 11.</p> <p><b>Note2:</b> If the operation mode is not in auto-reload mode (SC_TMRCTL2[26] = 0), this bit will be auto-cleared by hardware.</p> <p><b>Note3:</b> If SCEN (SC_CTL[0]) is not enabled, this field cannot be programmed.</p>
[6]	CNTEN1	<p><b>Internal Timer1 Start Enable Bit</b></p> <p>This bit enables Timer 1 to start counting. User can fill 0 to stop count and set 1 to reload and start count. The counter unit is ETU base.</p> <p>0 = Stop counting.</p> <p>1 = Start counting.</p> <p><b>Note1:</b> This field is used for internal 8-bit timer when TMRSEL (SC_CTL[14:13]) is 11 only. Do not fill CNTEN1 when TMRSEL (SC_CTL[14:13]) is not equal to 11.</p> <p><b>Note2:</b> If the operation mode is not in auto-reload mode (SC_TMRCTL1[26] = 0), this bit will be auto-cleared by hardware.</p> <p><b>Note3:</b> If SCEN (SC_CTL[0]) is not enabled, this field cannot be programmed.</p>
[5]	CNTEN0	<p><b>Internal Timer0 Start Enable Bit</b></p> <p>This bit enables Timer 0 to start counting. User can fill 0 to stop count and set 1 to reload and start count. The counter unit is ETU base.</p> <p>0 = Stop counting.</p> <p>1 = Start counting.</p> <p><b>Note1:</b> This field is used for internal 24-bit timer when TMRSEL (SC_CTL[14:13]) is 11 only. Do not fill CNTEN0 when TMRSEL (SC_CTL[14:13]) is not equal to 11.</p> <p><b>Note2:</b> If the operation mode is not in auto-reload mode (SC_TMRCTL0[26] = 0), this bit will be auto-cleared by hardware.</p> <p><b>Note3:</b> If SCEN (SC_CTL[0]) is not enabled, this field cannot be programmed.</p>
[4]	WARSTEN	<p><b>Warm Reset Sequence Generator Enable Bit</b></p> <p>This bit enables SC controller to initiate the card by warm reset sequence.</p> <p>0 = No effect.</p> <p>1 = Warm reset sequence generator Enabled.</p>

		<p><b>Note1:</b> When the warm reset sequence completed, this bit will be cleared automatically and the INITIF (SC_INTSTS[8]) will be set to 1.</p> <p><b>Note2:</b> This field will be cleared by set TXRST (SC_ALTCTL[0]) or RXRST (SC_ALTCTL[1]). Thus, do not fill in WARSTEN, TXRST or RXRST at the same time.</p> <p><b>Note3:</b> If SCEN (SC_CTL[0]) is not enabled, this filed cannot be programmed.</p>
[3]	ACTEN	<p><b>Activation Sequence Generator Enable Bit</b></p> <p>This bit enables SC controller to initiate the card by activation sequence.</p> <p>0 = No effect.</p> <p>1 = Activation sequence generator Enabled.</p> <p><b>Note1:</b> When the activation sequence completed, this bit will be cleared automatically and the INITIF (SC_INTSTS[8]) will be set to 1.</p> <p><b>Note2:</b> This field will be cleared by set TXRST (SC_ALTCTL[0]) or RXRST (SC_ALTCTL[1]). Thus, do not fill in ACTEN, TXRST or RXRST at the same time.</p> <p><b>Note3:</b> If SCEN (SC_CTL[0]) is not enabled, this filed cannot be programmed.</p>
[2]	DACTEN	<p><b>Deactivation Sequence Generator Enable Bit</b></p> <p>This bit enables SC controller to initiate the card by deactivation sequence.</p> <p>0 = No effect.</p> <p>1 = Deactivation sequence generator Enabled.</p> <p><b>Note1:</b> When the deactivation sequence completed, this bit will be cleared automatically and the INITIF (SC_INTSTS[8]) will be set to 1.</p> <p><b>Note2:</b> This field will be cleared by set TXRST (SC_ALTCTL[0]) or RXRST (SC_ALTCTL[1]). Thus, do not fill in DACTEN, TXRST or RXRST at the same time.</p> <p><b>Note3:</b> If SCEN (SC_CTL[0]) is not enabled, this filed cannot be programmed.</p>
[1]	RXRST	<p><b>RX Software Reset</b></p> <p>When RXRST is set, all the bytes in the receive buffer and Rx internal state machine will be cleared.</p> <p>0 = No effect.</p> <p>1 = Reset the Rx internal state machine and pointers.</p> <p><b>Note:</b> This bit will be auto cleared after reset is complete.</p>
[0]	TXRST	<p><b>TX Software Reset</b></p> <p>When TXRST is set, all the bytes in the transmit buffer and Tx internal state machine will be cleared.</p> <p>0 = No effect.</p> <p>1 = Reset the Tx internal state machine and pointers.</p> <p><b>Note:</b> This bit will be auto cleared after reset is complete.</p>

**SC Extra Guard Time Register (SC\_EGT)**

Register	Offset	R/W	Description	Reset Value
SC_EGT	SCx_BA+0x0C	R/W	SC Extra Guard Time 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
EGT							

Bits	Description	
[31:8]	Reserved	Reserved.
[7:0]	EGT	<b>Extra Guard Time</b> This field indicates the extra guard time value. <b>Note:</b> The extra guard time unit is ETU base.

**SC Receiver buffer Time-out Register (SC\_RXTOUT)**

Register	Offset	R/W	Description	Reset Value
SC_RXTOUT	SCx_BA+0x10	R/W	SC Receive Buffer Time-out 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							RFTM
7	6	5	4	3	2	1	0
RFTM							

Bits	Description	
[31:9]	Reserved	Reserved.
[8:0]	RFTM	<p><b>SC Receiver FIFO Time-out Counter</b></p> <p>The time-out down counter resets and starts counting whenever the Rx buffer received a new data. Once the counter decrease to 1 and no new data is received or CPU does not read data by reading DAT (SC_DAT[7:0]), a receiver time-out flag RXTOIF (SC_INTSTS[9]) will be set, and hardware will generate an interrupt signal to inform CPU when RXTOIEN (SC_INTEN[9]) is enabled.</p> <p><b>Note1:</b> The counter unit is ETU based and the interval of time-out is (RFTM + 0.5) ETU time.</p> <p><b>Note2:</b> Filling in all 0 to this field will disable this function.</p>

**SC Element Time Unit Control Register (SC\_ETUCTL)**

Register	Offset	R/W	Description	Reset Value
SC_ETUCTL	SCx_BA+0x14	R/W	SC Element Time Unit Control Register	0x0000_0173

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				ETURDIV			
7	6	5	4	3	2	1	0
ETURDIV							

Bits	Description	
[31:12]	Reserved	Reserved.
[11:0]	ETURDIV	<p><b>ETU Rate Divider</b></p> <p>The field is used for define ETU time unit. The real ETU time unit is (ETURDIV + 1) * SC clock time.</p> <p><b>Note:</b> User can configure this field, but this field must be greater than 0x004.</p>

**SC Interrupt Enable Control Register (SC\_INTEN)**

Register	Offset	R/W	Description	Reset Value
SC_INTEN	SCx_BA+0x18	R/W	SC Interrupt 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					ACERRIEN	RXTOIEN	INITIEN
7	6	5	4	3	2	1	0
CDIEN	BGTIEN	TMR2IEN	TMR1IEN	TMR0IEN	TERRIEN	TXEIEIEN	RDAIEN

Bits	Description	
[31:11]	Reserved	Reserved.
[10]	ACERRIEN	<b>Auto Convention Error Interrupt Enable Bit</b> This field is used to enable auto-convention error interrupt. 0 = Auto-convention error interrupt Disabled. 1 = Auto-convention error interrupt Enabled.
[9]	RXTOIEN	<b>Receiver Buffer Time-out Interrupt Enable Bit</b> This field is used to enable receiver buffer time-out interrupt. 0 = Receiver buffer time-out interrupt Disabled. 1 = Receiver buffer time-out interrupt Enabled.
[8]	INITIEN	<b>Initial End Interrupt Enable Bit</b> This field is used to enable activation (ACTEN (SC_ALTCTL[3] = 1)), deactivation (DACTEN (SC_ALTCTL[2] = 1)) and warm reset (WARSTEN (SC_ALTCTL [4])) sequence complete interrupt. 0 = Initial end interrupt Disabled. 1 = Initial end interrupt Enabled.
[7]	CDIEN	<b>Card Detect Interrupt Enable Bit</b> This field is used to enable card detect interrupt. The card detect status is CDPINSTS (SC_STATUS[13]). 0 = Card detect interrupt Disabled. 1 = Card detect interrupt Enabled. <b>Note:</b> Either cared insert or card remove event will generate crad detect event.
[6]	BGTIEN	<b>Block Guard Time Interrupt Enable Bit</b> This field is used to enable block guard time interrupt in receive direction. 0 = Block guard time interrupt Disabled. 1 = Block guard time interrupt Enabled. <b>Note:</b> This bit is valid only for receive direction block guard time.
[5]	TMR2IEN	<b>Timer2 Interrupt Enable Bit</b>

		This field is used to enable Timer2 interrupt function. 0 = Timer2 interrupt Disabled. 1 = Timer2 interrupt Enabled.
[4]	<b>TMR1IEN</b>	<b>Timer1 Interrupt Enable Bit</b> This field is used to enable the Timer1 interrupt function. 0 = Timer1 interrupt Disabled. 1 = Timer1 interrupt Enabled.
[3]	<b>TMR0IEN</b>	<b>Timer0 Interrupt Enable Bit</b> This field is used to enable Timer0 interrupt function. 0 = Timer0 interrupt Disabled. 1 = Timer0 interrupt Enabled.
[2]	<b>TERRIEN</b>	<b>Transfer Error Interrupt Enable Bit</b> This field is used to enable transfer error interrupt. The transfer error status is at SC_STATUS which includes receiver break error BEF (SC_STATUS[6]), frame error FEF (SC_STATUS[5]), parity error PEF (SC_STATUS[4]), receive buffer overflow error RXOV (SC_STATUS[0]), transmit buffer overflow error TXOV (SC_STATUS[8]), receiver retry over limit error RXOVERR (SC_STATUS[22]) or transmitter retry over limit error TXOVERR (SC_STATUS[30]). 0 = Transfer error interrupt Disabled. 1 = Transfer error interrupt Enabled.
[1]	<b>TXEIEIEN</b>	<b>Transmit Buffer Empty Interrupt Enable Bit</b> This field is used to enable transmit buffer empty interrupt. 0 = Transmit buffer empty interrupt Disabled. 1 = Transmit buffer empty interrupt Enabled.
[0]	<b>RDAIEN</b>	<b>Receive Data Reach Interrupt Enable Bit</b> This field is used to enable received data bytes in Rx buffer reaching trigger level RXTRGLV (SC_CTL[7:6]) interrupt. 0 = Received data bytes in Rx buffer reach trigger level interrupt Disabled. 1 = Received data bytes in Rx buffer reach trigger level interrupt Enabled.

**SC Interrupt Status Register (SC\_INTSTS)**

Register	Offset	R/W	Description	Reset Value
SC_INTSTS	SCx_BA+0x1C	R/W	SC Interrupt Status Register	0x0000_0002

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					ACERRIF	RXTOIF	INITIF
7	6	5	4	3	2	1	0
CDIF	BGTIF	TMR2IF	TMR1IF	TMR0IF	TERRIF	TXEIF	RDAIF

Bits	Description	
[31:11]	Reserved	Reserved.
[10]	ACERRIF	<p><b>Auto Convention Error Interrupt Status Flag</b>                      This field indicates auto convention sequence error.                      0 = Received TS at ATR state is 0x3B or 0x3F.                      1 = Received TS at ATR state is neither 0x3B nor 0x3F.  <b>Note:</b> This bit can be cleared by writing 1 to it.</p>
[9]	RXTOIF	<p><b>Receive Buffer Time-out Interrupt Status Flag (Read Only)</b>                      This field is used for indicate receive buffer time-out interrupt status flag.                      0 = Receive buffer time-out interrupt did not occur.                      1 = Receive buffer time-out interrupt occurred.  <b>Note:</b> This bit is read only, user must read all receive buffer remaining data by reading DAT (SC_DAT[7:0]) to clear it.</p>
[8]	INITIF	<p><b>Initial End Interrupt Status Flag</b>                      This field is used for activation (ACTEN (SC_ALTCTL[3])), deactivation (DACTEN (SC_ALTCTL[2])) and warm reset (WARSTEN (SC_ALTCTL[4])) sequence interrupt status flag.                      0 = Initial sequence is not complete.                      1 = Initial sequence is completed.  <b>Note:</b> This bit can be cleared by writing 1 to it.</p>
[7]	CDIF	<p><b>Card Detect Interrupt Status Flag (Read Only)</b>                      This field is used for card detect interrupt status flag. The actual card detect status is in CINSERT (SC_STATUS[12]) and CREMOVE (SCn_STATUS[11]).                      0 = Card detect event did not occur.                      1 = Card detect event occurred.  <b>Note1:</b> This bit is read only, and will be cleared after CINSERT or CREMOVE status has been cleared.  <b>Note2:</b> Either cared insert or card remove event will generate crad detect event.</p>
[6]	BGTIF	<b>Block Guard Time Interrupt Status Flag</b>



		<p>This field is used for indicate block guard time interrupt status flag in receive direction.</p> <p>0 = Block guard time interrupt did not occur. 1 = Block guard time interrupt occurred.</p> <p><b>Note1:</b> This bit is valid only when RXBGTEN (SC_ALTCTL[12]) is enabled. <b>Note2:</b> This bit can be cleared by writing 1 to it.</p>
[5]	TMR2IF	<p><b>Timer2 Interrupt Status Flag</b></p> <p>This field is used for Timer2 interrupt status flag.</p> <p>0 = Timer2 interrupt did not occur. 1 = Timer2 interrupt occurred.</p> <p><b>Note:</b> This bit can be cleared by writing 1 to it.</p>
[4]	TMR1IF	<p><b>Timer1 Interrupt Status Flag</b></p> <p>This field is used for Timer1 interrupt status flag.</p> <p>0 = Timer1 interrupt did not occur. 1 = Timer1 interrupt occurred.</p> <p><b>Note:</b> This bit can be cleared by writing 1 to it.</p>
[3]	TMR0IF	<p><b>Timer0 Interrupt Status Flag</b></p> <p>This field is used for Timer0 interrupt status flag.</p> <p>0 = Timer0 interrupt did not occur. 1 = Timer0 interrupt occurred.</p> <p><b>Note:</b> This bit can be cleared by writing 1 to it.</p>
[2]	TERRIF	<p><b>Transfer Error Interrupt Status Flag</b></p> <p>This field is used for indicate transfer error interrupt status flag. The transfer error status is at SC_STATUS which includes receiver break error BEF (SC_STATUS[6]), frame error FEF (SC_STATUS[5]), parity error PEF (SC_STATUS[4]), receive buffer overflow error RXOV (SC_STATUS[0]), transmit buffer overflow error TXOV (SC_STATUS[8]), receiver retry over limit error RXOVERR (SC_STATUS[22]) or transmitter retry over limit error TXOVERR (SC_STATUS[30]).</p> <p>0 = Transfer error interrupt did not occur. 1 = Transfer error interrupt occurred.</p> <p><b>Note1:</b> This field is the status flag of BEF, FEF, PEF, RXOV, TXOV, RXOVERR or TXOVERR. <b>Note2:</b> This bit can be cleared by writing 1 to it.</p>
[1]	TXEIF	<p><b>Transmit Buffer Empty Interrupt Status Flag (Read Only)</b></p> <p>This field is used for transmit buffer empty interrupt status flag.</p> <p>0 = Transmit buffer is not empty. 1 = Transmit buffer is empty.</p> <p><b>Note:</b> This bit is read only. If user wants to clear this bit, user must write data to DAT (SC_DAT[7:0]) and then this bit will be cleared automatically.</p>
[0]	RDAIF	<p><b>Receive Data Reach Interrupt Status Flag (Read Only)</b></p> <p>This field is used for received data bytes in Rx buffer reaching trigger level RXTRGLV (SC_CTL[7:6]) interrupt status flag.</p> <p>0 = Number of receive buffer is less than RXTRGLV setting. 1 = Number of receive buffer data equals the RXTRGLV setting.</p> <p><b>Note:</b> This bit is read only. If user reads data from DAT (SC_DAT[7:0]) and remains receiver buffer data byte number is less than RXTRGLV, this bit will be cleared automatically.</p>

**SC Transfer Status Register (SC\_STATUS)**

Register	Offset	R/W	Description	Reset Value
SC_STATUS	SCx_BA+0x20	R/W	SC Transfer Status Register	0x0000_x202

31	30	29	28	27	26	25	24
TXACT	TXOVERR	TXRTYERR	Reserved		TXPOINT		
23	22	21	20	19	18	17	16
RXACT	RXOVERR	RXRTYERR	Reserved		RXPOINT		
15	14	13	12	11	10	9	8
Reserved		CDPINSTS	CINSERT	CREMOVE	TXFULL	TXEMPTY	TXOV
7	6	5	4	3	2	1	0
Reserved	BEF	FEF	PEF	Reserved	RXFULL	RXEMPTY	RXOV

Bits	Description	
[31]	TXACT	<p><b>Transmit in Active Status Flag (Read Only)</b></p> <p>This bit indicates Tx transmit status.</p> <p>0 = This bit is cleared automatically when Tx transfer is finished or the last byte transmission has completed.</p> <p>1 = Transmit is active or the STOP bit of last byte has not been transmitted when Tx transfer is in active.</p>
[30]	TXOVERR	<p><b>Transmitter over Retry Error</b></p> <p>This bit is used for transmitter retry counts over than retry number limitation.</p> <p>0 = Transmitter retries counts is not over than TXRTY (SC_CTL[22:20]) + 1.</p> <p>1 = Transmitter retries counts over than TXRTY (SC_CTL[22:20]) + 1.</p> <p><b>Note:</b> This bit can be cleared by writing 1 to it.</p>
[29]	TXRTYERR	<p><b>Transmitter Retry Error</b></p> <p>This bit is used for indicate transmitter error retry and set by hardware..</p> <p>0 = No Tx retry transfer.</p> <p>1 = Tx has any error and retries transfer.</p> <p><b>Note1:</b> This bit can be cleared by writing 1 to it.</p> <p><b>Note2:</b> This bit is a flag and cannot generate any interrupt signal to CPU.</p>
[28:27]	Reserved	Reserved.
[26:24]	TXPOINT	<p><b>Transmit Buffer Pointer Status (Read Only)</b></p> <p>This field indicates the Tx buffer pointer status. When CPU writes data into DAT (SC_DAT[7:0]), TXPOINT increases one. When one byte of Tx buffer is transferred to Transmitter Shift Register, TXPOINT decreases one.</p>
[23]	RXACT	<p><b>Receiver in Active Status Flag (Read Only)</b></p> <p>This bit indicates Rx transfer status.</p> <p>0 = This bit is cleared automatically when Rx transfer is finished.</p> <p>1 = This bit is set by hardware when Rx transfer is in active.</p>
[22]	RXOVERR	<p><b>Receiver over Retry Error</b></p>

		<p>This bit is used for indicate receiver retry counts over than retry number limitation.</p> <p>0 = Receiver retries counts is not over than RXRTY (SC_CTL[18:16]) + 1.</p> <p>1 = Receiver retries counts over than RXRTY (SC_CTL[18:16]) + 1.</p> <p><b>Note1:</b> This bit can be cleared by writing 1 to it.</p> <p><b>Note2:</b> If user enables receiver retries function by setting RXRTYEN (SC_CTL[19]), the PEF (SC_STATUS[4]) bit will not set.</p>
[21]	<b>RXRTYERR</b>	<p><b>Receiver Retry Error</b></p> <p>This bit is used for indicate receiver error retry and set by hardware.</p> <p>0 = No Rx retry transfer.</p> <p>1 = Rx has any error and retries transfer.</p> <p><b>Note1:</b> This bit can be cleared by writing 1 to it.</p> <p><b>Note2:</b> This bit is a flag and cannot generate any interrupt signal to CPU.</p> <p><b>Note3:</b> If user enables receiver retries function by setting RXRTYEN (SC_CTL[19]), the PEF (SC_STATUS[4]) bit will not set.</p>
[20:19]	<b>Reserved</b>	Reserved.
[18:16]	<b>RXPOINT</b>	<p><b>Receive Buffer Pointer Status (Read Only)</b></p> <p>This field indicates the Rx buffer pointer status. When SC controller receives one byte from external device, RXPOINT increases one. When one byte in Rx buffer is read by reading DAT (SC_DAT[7:0]), RXPOINT decreases one.</p>
[15:14]	<b>Reserved</b>	Reserved.
[13]	<b>CDPINSTS</b>	<p><b>Card Detect Pin Status (Read Only)</b></p> <p>This bit is the pin status of SC_CD.</p> <p>0 = The SC_CD pin state at low.</p> <p>1 = The SC_CD pin state at high.</p>
[12]	<b>CINSERT</b>	<p><b>Card Insert Status of SC_CD Pin</b></p> <p>This bit is set whenever card has been inserted.</p> <p>0 = No effect.</p> <p>1 = Card insert.</p> <p><b>Note1:</b> This bit can be cleared by writing 1 to it.</p> <p><b>Note2:</b> The card detect function will start after SCEN (SC_CTL[0]) is set.</p>
[11]	<b>CREMOVE</b>	<p><b>Card Removal Status of SCn_CD Pin</b></p> <p>This bit is set whenever card has been removal.</p> <p>0 = No effect.</p> <p>1 = Card removed.</p> <p><b>Note1:</b> This bit can be cleared by writing 1 to it.</p> <p><b>Note2:</b> Card detect function will start after SCEN (SC_CTL[0]) is set.</p>
[10]	<b>TXFULL</b>	<p><b>Transmit Buffer Full Status Flag (Read Only)</b></p> <p>This bit indicates Tx buffer is full or not.</p> <p>0 = Tx buffer count is less than 4.</p> <p>1 = Tx buffer count equals to 4.</p>
[9]	<b>TXEMPTY</b>	<p><b>Transmit Buffer Empty Status Flag (Read Only)</b></p> <p>This bit indicates TX buffer is empty or not.</p> <p>0 = Tx buffer is not empty.</p> <p>1 = Tx buffer is empty, it means the last byte of Tx buffer has been transferred to Transmitter Shift Register.</p> <p><b>Note:</b> This bit will be cleared when writing data into DAT (SC_DAT[7:0]).</p>
[8]	<b>TXOV</b>	<b>Transmit Overflow Error Interrupt Status Flag</b>

		<p>This bit is set when Tx buffer overflow.</p> <p>0 = Tx buffer is not overflow.</p> <p>1 = Tx buffer is overflow, it means an additional write operation to DAT (SC_DAT[7:0]) when Tx buffer is already full.</p> <p><b>Note:</b> This bit can be cleared by writing 1 to it.</p>
[7]	<b>Reserved</b>	Reserved.
[6]	<b>BEF</b>	<p><b>Receiver Break Error Status Flag</b></p> <p>This bit is set to logic 1 whenever the received data input (Rx) held in the “spacing state” (logic 0) is longer than a full word transmission time (that is, the total time of “start bit” + “data bits” + “parity bit” + “stop bits”).</p> <p>0 = Receiver break error flag did not occur.</p> <p>1 = Receiver break error flag occurred.</p> <p><b>Note1:</b> This bit can be cleared by writing 1 to it.</p> <p><b>Note2:</b> If user sets receiver retries function by setting RXRTYEN (SCn_CTL[19]), hardware will not set this flag.</p>
[5]	<b>FEF</b>	<p><b>Receiver Frame Error Status Flag</b></p> <p>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).</p> <p>0 = Receiver frame error flag did not occur.</p> <p>1 = Receiver frame error flag occurred.</p> <p><b>Note1:</b> This bit can be cleared by writing 1 to it.</p> <p><b>Note2:</b> If user sets receiver retries function by setting RXRTYEN (SC_CTL[19]), hardware will not set this flag.</p>
[4]	<b>PEF</b>	<p><b>Receiver Parity Error Status Flag</b></p> <p>This bit is set to logic 1 whenever the received character does not have a valid “parity bit”.</p> <p>0 = Receiver parity error flag did not occur.</p> <p>1 = Receiver parity error flag occurred.</p> <p><b>Note1:</b> This bit can be cleared by writing 1 to it.</p> <p><b>Note2:</b> If user sets receiver retries function by setting RXRTYEN (SC_CTL[19]), hardware will not set this flag.</p>
[3]	<b>Reserved</b>	Reserved.
[2]	<b>RXFULL</b>	<p><b>Receive Buffer Full Status Flag (Read Only)</b></p> <p>This bit indicates Rx buffer is full or not.</p> <p>0 = Rx buffer count is less than 4.</p> <p>1 = Rx buffer count equals to 4.</p>
[1]	<b>RXEMPTY</b>	<p><b>Receive Buffer Empty Status Flag (Read Only)</b></p> <p>This bit indicates Rx buffer is empty or not.</p> <p>0 = Rx buffer is not empty.</p> <p>1 = Rx buffer is empty, it means the last byte in Rx buffer has been read from DAT (SC_DAT[7:0]).</p>
[0]	<b>RXOV</b>	<p><b>Receive Overflow Error Status Flag</b></p> <p>This bit is set when Rx buffer overflow.</p> <p>0 = Rx buffer is not overflow.</p> <p>1 = Rx buffer is overflow when the number of received bytes is greater than Rx buffer size (4 bytes).</p> <p><b>Note:</b> This bit can be cleared by writing 1 to it.</p>

**SC Pin Control State Register (SC\_PINCTL)**

Register	Offset	R/W	Description	Reset Value
SC_PINCTL	SCx_BA+0x24	R/W	SC Pin Control State Register	0x0000_00x0

31	30	29	28	27	26	25	24	
Reserved	SYNC	Reserved						
23	22	21	20	19	18	17	16	
Reserved						RSTSTS	PWRSTS	DATSTS
15	14	13	12	11	10	9	8	
Reserved				PWRINV	Reserved	SCDATA	Reserved	
7	6	5	4	3	2	1	0	
Reserved	CLKKEEP	Reserved				SCRST	PWREN	

Bits	Description
[31]	<b>Reserved</b> Reserved.
[30]	<b>SYNC</b> <b>SYNC Flag Indicator (Read Only)</b> Due to synchronization, user should check this bit when writing a new value to SC_PINCTL register. 0 = Synchronizing is completion, user can write new data to SC_PINCTL register. 1 = Last value is synchronizing.
[29:19]	<b>Reserved</b> Reserved.
[18]	<b>RSTSTS</b> <b>SC_RST Pin Status (Read Only)</b> This bit is the pin status of SC_RST. 0 = SC_RST pin is low. 1 = SC_RST pin is high.
[17]	<b>PWRSTS</b> <b>SC_PWR Pin Status (Read Only)</b> This bit is the pin status of SC_PWR. 0 = SC_PWR pin to low. 1 = SC_PWR pin to high.
[16]	<b>DATSTS</b> <b>SC_DATA Pin Status (Read Only)</b> This bit is the pin status of SC_DATA. 0 = The SC_DATA pin status is low. 1 = The SC_DATA pin status is high.
[15:12]	<b>Reserved</b> Reserved.
[11]	<b>PWRINV</b> <b>SC_PWR Pin Inverse</b> This bit is used for inverse the SC_PWR pin. There are four kinds of combination for SC_PWR pin setting by PWRINV (SC_PINCTL[11]) and PWREN (SC_PINCTL[0]). And all conditions as below list.

		<p>00 = SC_PWR pin is 0.                  01 = SC_PWR pin is 1.                  10 = SC_PWR pin is 1.                  11 = SC_PWR pin is 0.  <b>Note:</b> User must select PWRINV (SC_PINCTL[11]) before smart card is enabled by SCEN (SC_CTL[0]).</p>
[10]	<b>Reserved</b>	Reserved.
[9]	<b>SCDATA</b>	<p><b>SC_DATA Pin Signal</b>                  This bit is the signal status of SC_DATA but user can also drive SC_DATA pin to high or low by control this bit.                  Write this bit can drive SC_RST pin.                  0 = Drive SC_DATA pin to low.                  1 = Drive SC_DATA pin to high.                  Read this bit to get SC_DATA signal status.                  0 = SC_DATA signal status is low.                  1 = SC_DATA signal status is high.  <b>Note:</b> When SC is at activation, warm reset or deactivation mode, this bit will be changed automatically. Thus, do not fill in this field when SC is in these modes.</p>
[8:7]	<b>Reserved</b>	Reserved.
[6]	<b>CLKKEEP</b>	<p><b>SC Clock Enable Bit</b>                  0 = SC clock generation Disabled.                  1 = SC clock always keeps free running.  <b>Note:</b> When operating in activation, warm reset or deactivation mode, this bit will be changed automatically. Thus, do not fill in this field when operating in these modes.</p>
[5:2]	<b>Reserved</b>	Reserved.
[1]	<b>SCRST</b>	<p><b>SC_RST Pin Signal</b>                  This bit is the signal status of SC_RST but user can drive SC_RST pin to high or low by control this bit.                  Write this bit can drive SC_RST pin.                  0 = Drive SC_RST pin to low.                  1 = Drive SC_RST pin to high.                  Read this bit to get SC_RST signal status.                  0 = SC_RST signal status is low.                  1 = SC_RST signal status is high.  <b>Note:</b> When operating at activation, warm reset or deactivation mode, this bit will be changed automatically. Thus, do not fill in this field when operating in these modes.</p>
[0]	<b>PWREN</b>	<p><b>SC_PWR Pin Signal</b>                  User can set PWRINV (SC_PINCTL[11]) and PWREN (SC_PINCTL[0]) to decide SC_PWR pin is in high or low level.                  Write this bit can drive SC_PWR pin                  Refer PWRINV (SC_PINCTL[11]) description for programming SC_PWR pin voltage level.                  Read this bit to get SC_PWR signal status.                  0 = SC_PWR signal status is low.                  1 = SC_PWR signal status is high.  <b>Note:</b> When operating at activation, warm reset or deactivation mode, this bit will be changed automatically. Thus, do not fill in this field when operating in these modes.</p>

**SC Timer0 Control Register (SC\_TMRCTL0)**

Register	Offset	R/W	Description	Reset Value
SC_TMRCTL0	SCx_BA+0x28	R/W	SC Timer0 Control Register	0x0000_0000

31	30	29	28	27	26	25	24	
SYNC		Reserved			OPMODE			
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]	SYNC	<p><b>SYNC Flag Indicator (Read Only)</b></p> <p>Due to synchronization, user should check this bit when writing a new value to SC_TMRCTL0 register.</p> <p>0 = Synchronizing is completion, user can write new data to SC_TMRCTL0 register.</p> <p>1 = Last value is synchronizing.</p>
[30:28]	Reserved	Reserved.
[27:24]	OPMODE	<p><b>Timer 0 Operation Mode Selection</b></p> <p>This field indicates the internal 24-bit Timer0 operation selection. Refer to Table 6.15-3 for programming Timer0.</p>
[23:0]	CNT	<p><b>Timer 0 Counter Value</b></p> <p>This field indicates the internal Timer0 counter values.</p> <p><b>Note:</b> Unit of Timer0 counter is ETU base.</p>

**SC Timer1 Control Register (SC\_TMRCTL1)**

Register	Offset	R/W	Description	Reset Value
SC_TMRCTL1	SCx_BA+0x2C	R/W	SC Timer1 Control Register	0x0000_0000

31	30	29	28	27	26	25	24	
SYNC		Reserved			OPMODE			
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	
CNT								

Bits	Description	
[31]	SYNC	<p><b>SYNC Flag Indicator (Read Only)</b></p> <p>Due to synchronization, user should check this bit when writing a new value to SC_TMRCTL1 register.</p> <p>0 = Synchronizing is completion, user can write new data to SC_TMRCTL1 register.</p> <p>1 = Last value is synchronizing.</p>
[30:28]	Reserved	Reserved.
[27:24]	OPMODE	<p><b>Timer 1 Operation Mode Selection</b></p> <p>This field indicates the internal 8-bit Timer1 operation selection. Refer to Table 6.15-3 for programming Timer1.</p>
[23:8]	Reserved	Reserved.
[7:0]	CNT	<p><b>Timer 1 Counter Value</b></p> <p>This field indicates the internal Timer1 counter values.</p> <p><b>Note:</b> Unit of Timer1 counter is ETU base.</p>



**SC Timer2 Control Register (SC\_TMRCTL2)**

Register	Offset	R/W	Description	Reset Value
SC_TMRCTL2	SCx_BA+0x30	R/W	SC Timer2 Control Register	0x0000_0000

31	30	29	28	27	26	25	24	
SYNC		Reserved			OPMODE			
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	
CNT								

Bits	Description	
[31]	SYNC	<p><b>SYNC Flag Indicator (Read Only)</b></p> <p>Due to synchronization, user should check this bit when writing a new value to SC_TMRCTL2 register.</p> <p>0 = Synchronizing is completion, user can write new data to SC_TMRCTL2 register.</p> <p>1 = Last value is synchronizing.</p>
[30:28]	Reserved	Reserved.
[27:24]	OPMODE	<p><b>Timer 2 Operation Mode Selection</b></p> <p>This field indicates the internal 8-bit Timer2 operation selection</p> <p>Refer to Table 6.15-3 for programming Timer2.</p>
[23:8]	Reserved	Reserved.
[7:0]	CNT	<p><b>Timer 2 Counter Value</b></p> <p>This field indicates the internal Timer2 counter values.</p> <p><b>Note:</b> Unit of Timer2 counter is ETU base.</p>

**SC UART Mode Control Register (SC\_UARTCTL)**

Register	Offset	R/W	Description	Reset Value
SC_UARTCTL	SCx_BA+0x34	R/W	SC UART Mode Control Register	0x0000_0000

Bits	Description	
[31:8]	<b>Reserved</b>	Reserved.
[7]	<b>OPE</b>	<p><b>Odd Parity Enable Bit</b> This is used for odd/even parity selection. 0 = Even number of logic 1's are transmitted or check the data word and parity bits in receiving mode. 1 = Odd number of logic 1's are transmitted or check the data word and parity bits in receiving mode. <b>Note:</b> This bit has effect only when PBOFF bit is 0.</p>
[6]	<b>PBOFF</b>	<p><b>Parity Bit Disable Control</b> This bit is used to disable parity check function. 0 = Parity bit is generated or checked between the "last data word bit" and "stop bit" of the serial data. 1 = Parity bit is not generated (transmitting data) or checked (receiving data) during transfer. <b>Note:</b> In smart card mode, this bit must be 0 (default setting is with parity bit).</p>
[5:4]	<b>WLS</b>	<p><b>Word Length Selection</b> This field is used to select UART data transfer length. 00 = Word length is 8 bits. 01 = Word length is 7 bits. 10 = Word length is 6 bits. 11 = Word length is 5 bits. <b>Note:</b> In smart card mode, this field must be 00.</p>
[3:1]	<b>Reserved</b>	Reserved.
[0]	<b>UARTEN</b>	<p><b>UART Mode Enable Bit</b> Set this bit to enable UART mode function. 0 = Smart Card mode. 1 = UART mode. <b>Note1:</b> When operating in UART mode, user must set CONSEL (SC_CTL[5:4]) = 00 and AUTOCEN (SC_CTL[3]) = 0. <b>Note2:</b> When operating in Smart Card mode, user must set UARTEN (SC_UARTCTL[0]) = 0. <b>Note3:</b> When UART mode is enabled, hardware will generate a reset SC event to reset FIFO and internal state machine.</p>

**SC Timer0 Current Data Register (SC\_TMRDAT0)**

Register	Offset	R/W	Description	Reset Value
SC_TMRDAT0	SCx_BA+0x38	R	SC Timer0 Current Data Register	0x0000_07FF

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
CNT0							
15	14	13	12	11	10	9	8
CNT0							
7	6	5	4	3	2	1	0
CNT0							

Bits	Description	
[31:24]	Reserved	Reserved.
[23:0]	CNT0	<b>Timer0 Current Data Value (Read Only)</b> This field indicates the current counter values of Timer0.

**SC Timer1/2 Current Data Register (SC\_TMRDAT12)**

Register	Offset	R/W	Description	Reset Value
SC_TMRDAT12	SCx_BA+0x3C	R	SC Timer1/2 Current Data Register	0x0000_7F7F

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
CNT2							
7	6	5	4	3	2	1	0
CNT1							

Bits	Description	
[31:16]	Reserved	Reserved.
[15:8]	CNT2	<b>Timer2 Current Data Value (Read Only)</b> This field indicates the current counter values of Timer2.
[7:0]	CNT1	<b>Timer1 Current Data Value (Read Only)</b> This field indicates the current counter values of Timer1.

**SC Activation Control Register (SC\_ACTCTL)**

Register	Offset	R/W	Description	Reset Value
SC_ACTCTL	SCx_BA+0x4C	R/W	SC Activation 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				T1EXT			

Bits	Description

[31:5]	Reserved	Reserved.
[4:0]	T1EXT	<p><b>T1 Extend Time of Hardware Activation</b></p> <p>This field provide the configurable cycles to extend the activation time T1 period. Please refer to SC activation sequence in Figure 6.15-4.</p> <p>The cycle scaling factor is 2048 and</p> <p>Extend cycles = (T1EXT * 2048) cycles.</p> <p>For example:</p> <p>If SCLK is 4MHz, each clock cycle is 0.25us.</p> <p>Filled 20 to T1EXT, then</p> <p>Extend time = 20 * 2048 * 0.25us = 10.24 ms.</p> <p><b>Note:</b> Setting 0 to this field conforms to the protocol ISO/IEC 7816-3.</p>

## 6.16 Serial Peripheral Interface (SPI)

### 6.16.1 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 NUC126 series contains up to two sets of SPI controllers 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. Each SPI controller can be configured as a master or a slave device.

This controller also supports the PDMA function to access the data buffer. The SPI controller also support I<sup>2</sup>S mode to connect external audio CODEC.

### 6.16.2 Features

- SPI Mode
  - Up to two sets of SPI controllers
  - 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 PDMA transfer
  - Supports one data channel half-duplex transfer
  - Support receive-only mode
- I<sup>2</sup>S Mode
  - Supports Master or Slave
  - Capable of handling 8-, 16-, 24- and 32-bit word sizes
  - Provides separate 4-level depth transmit and receive FIFO buffers
  - Supports monaural and stereo audio data
  - Supports PCM mode A, PCM mode B, I<sup>2</sup>S and MSB justified data format
  - Supports PDMA transfer

### 6.16.3 Block Diagram

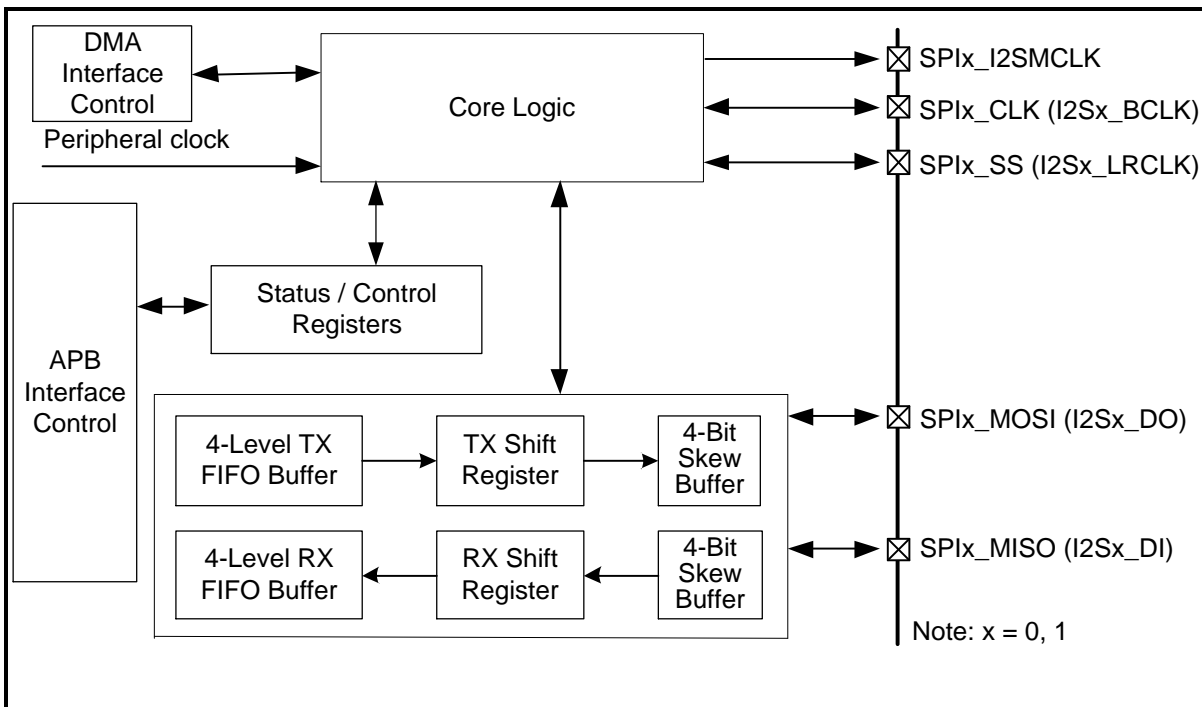


Figure 6.16-1 SPI Block Diagram (SPI0/1)

#### 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.

#### RX FIFO Buffer:

The received FIFO buffer is also a 4-level depth, 32-bit wide, first-in, first-out register buffer. The receive control logic will store the received data to this buffer. The FIFO buffer data can be read from SPIx\_RX register by software.

#### 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.

### 6.16.4 Basic Configuration

#### 6.16.4.1 Basic Configuration of SPI0

- Clock source Configuration
  - Select the source of SPI0 peripheral clock on SPI0SEL (CLK\_CLKSEL2[25:24]).

- Enable SPI0 peripheral clock in SPI0CKEN (CLK\_APBCLK0[12]).
- Reset Configuration
  - Reset SPI0 controller in SPI0RST (SYS\_IPRST1[12]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
SPI0	SPI0_CLK	PB.2, PB.7, PC.0, PC.12, PE.0, PE.13	MFP2
	SPI0_I2SMCLK	PD.0	MFP1
		PC.5, PC.9, PD.3, PD.7	MFP2
	SPI0_MISO	PB.3, PB.6, PC.4, PC.11, PE.10	MFP2
	SPI0_MOSI	PB.5, PC.3, PC.10, PE.11	MFP2
	SPI0_SS	PB.4, PC.2, PC.13, PE.12	MFP2

6.16.4.2 Basic Configuration of SPI1

- Clock source Configuration
  - Select the source of SPI1 peripheral clock on SPI1SEL (CLK\_CLKSEL2[27:26]).
  - Enable SPI1 peripheral clock in SPI1CKEN (CLK\_APBCLK0[13]).
- Reset Configuration
  - Reset SPI1 controller in SPI1RST (SYS\_IPRST1[13]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
SPI1	SPI1_CLK	PE.13	MFP1
		PA.7, PD.4, PD.15	MFP2
		PB.2, PB.7	MFP3
		PE.10	MFP6
	SPI1_I2SMCLK	PA.9	MFP1
		PA.12, PD.0	MFP2
	SPI1_MISO	PE.10	MFP1
		PA.6, PD.5, PD.14	MFP2
		PB.3, PB.6	MFP3
		PE.11	MFP6
	SPI1_MOSI	PE.11	MFP1
		PA.5, PD.13, PE.3	MFP2
		PB.5	MFP3
		PE.12	MFP6



SPI1_SS	PE.12	MFP1
	PA.4, PD.6, PD.12	MFP2
	PB.4	MFP3
	PE.13	MFP6

SPI/I<sup>2</sup>S Interface Controller Pin description is shown as follows:

Pin	SPI Mode	I <sup>2</sup> S Mode
SPIx_SS	SPI slave selection pin	I <sup>2</sup> S left/right channel synchronization clock pin (I2Sx_LRCLK)
SPIx_CLK	SPI clock pin	I <sup>2</sup> S bit clock pin (I2Sx_BCLK)
SPIx_MISO	SPI master input or slave output pin	I <sup>2</sup> S data input pin (I2Sx_DI)
SPIx_MOSI	SPI master output or slave input pin	I <sup>2</sup> S data output pin (I2Sx_DO)
SPIx_I2SMCLK	Not available	I <sup>2</sup> S Master clock output pin

Table 6.16-1 SPI/I<sup>2</sup>S Interface Controller Pin

### 6.16.5 Functional Description

#### 6.16.5.1 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, PLL, PCLK0 or HIRC48. SPIxSEL of CLK\_CLKSEL2 register determines the clock source of the peripheral clock. The DIVIDER (SPIx\_CLKDIV[7:0]) setting determines the divisor of the clock rate calculation.

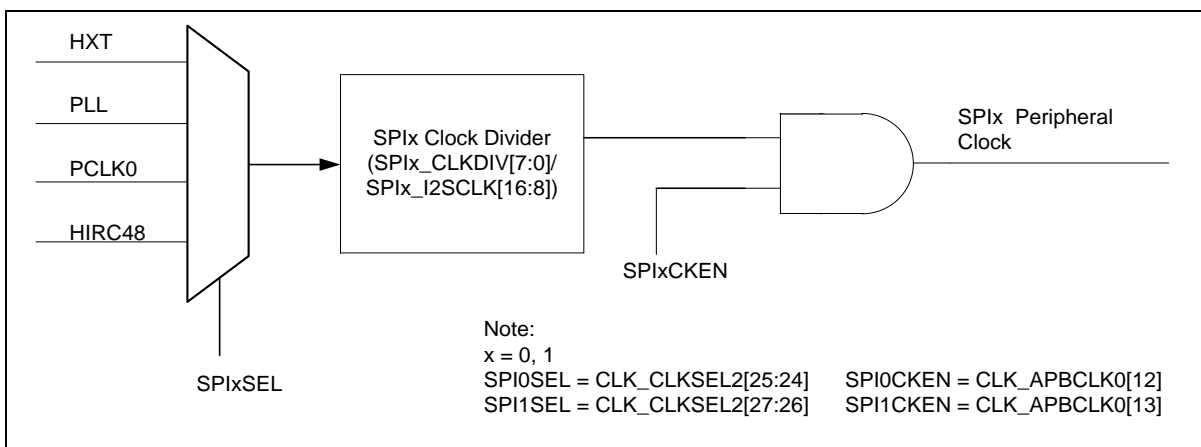


Figure 6.16-2 SPI Peripheral Clock

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. If the clock source of peripheral clock is not system clock, the frequency of SPI peripheral clock shall be slower than the system clock frequency regardless of Master or Slave mode.

In I<sup>2</sup>S mode, the peripheral clock rate is equal to I<sup>2</sup>S bit clock rate determined by SPIx\_I2SCLK

register.

**Master/Slave Mode**

This SPI controller 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.

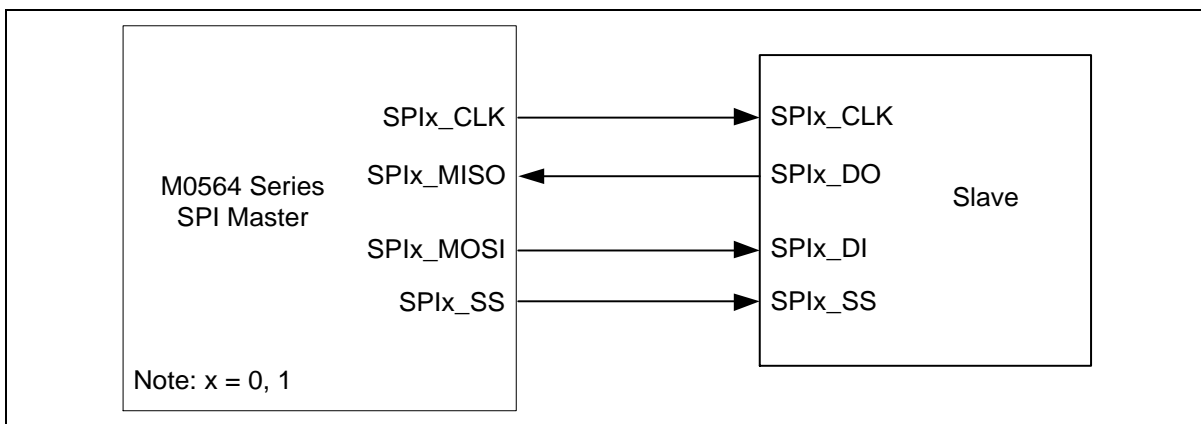


Figure 6.16-3 SPI Full-Duplex Master Mode Application Block Diagram

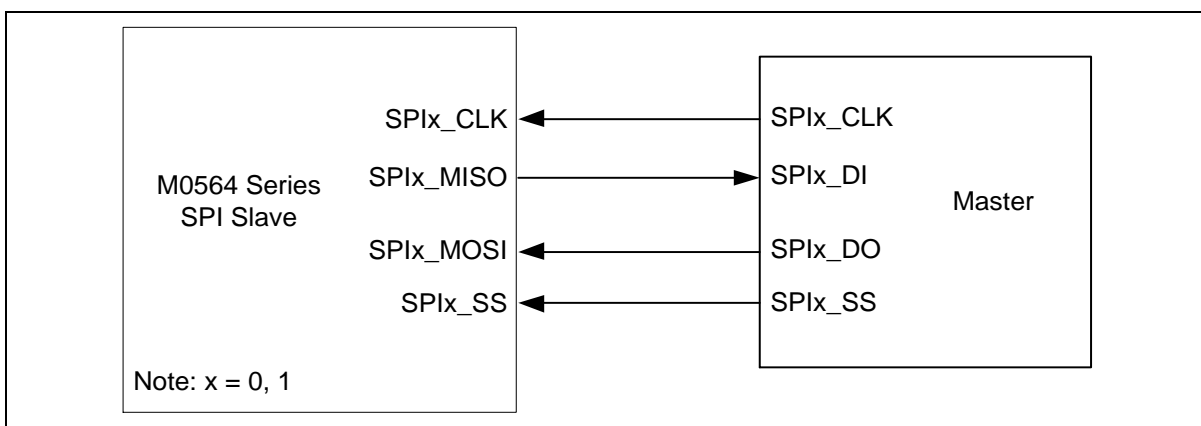


Figure 6.16-4 SPI Full-Duplex Slave Mode Application Block Diagram

**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.

**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 will be set to 1.

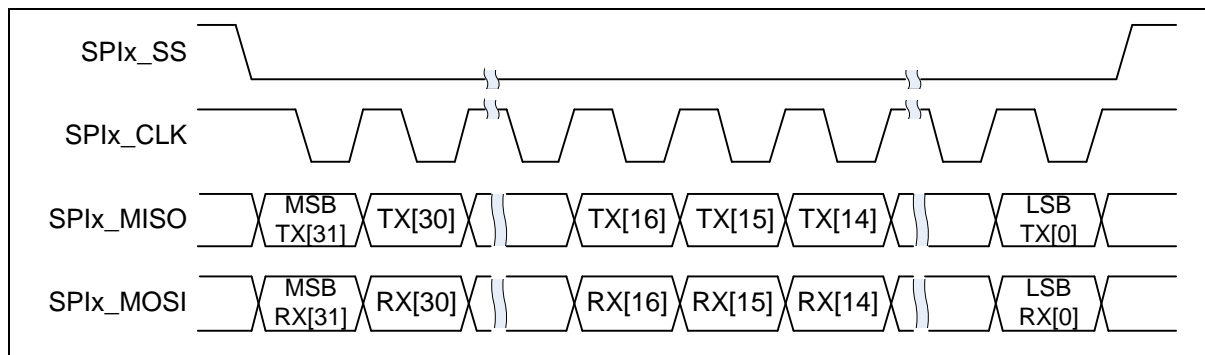


Figure 6.16-532-Bit in One Transaction

**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).

**6.16.5.2 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.

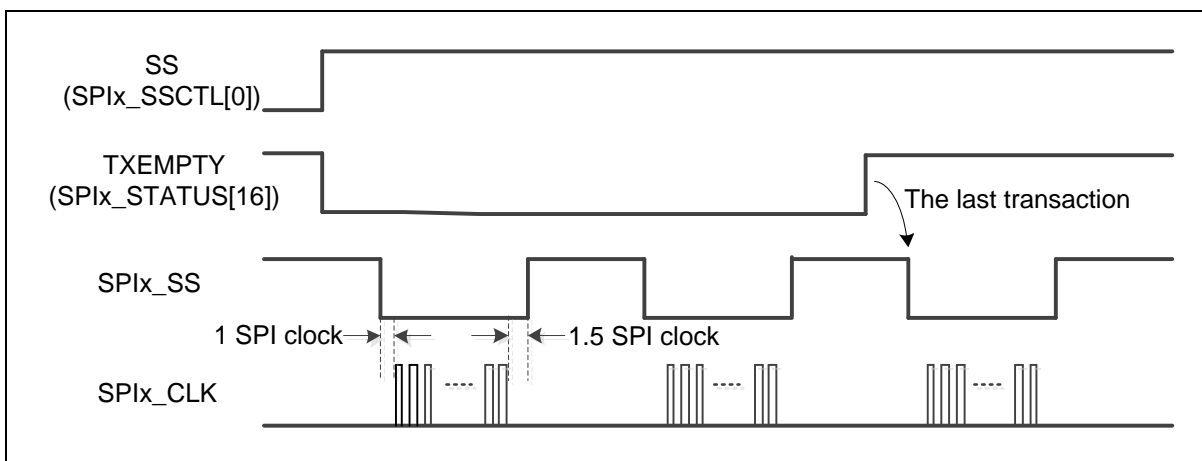


Figure 6.16-6 Automatic Slave Selection (SSACTPOL = 0, SUSPITV > 0x2)

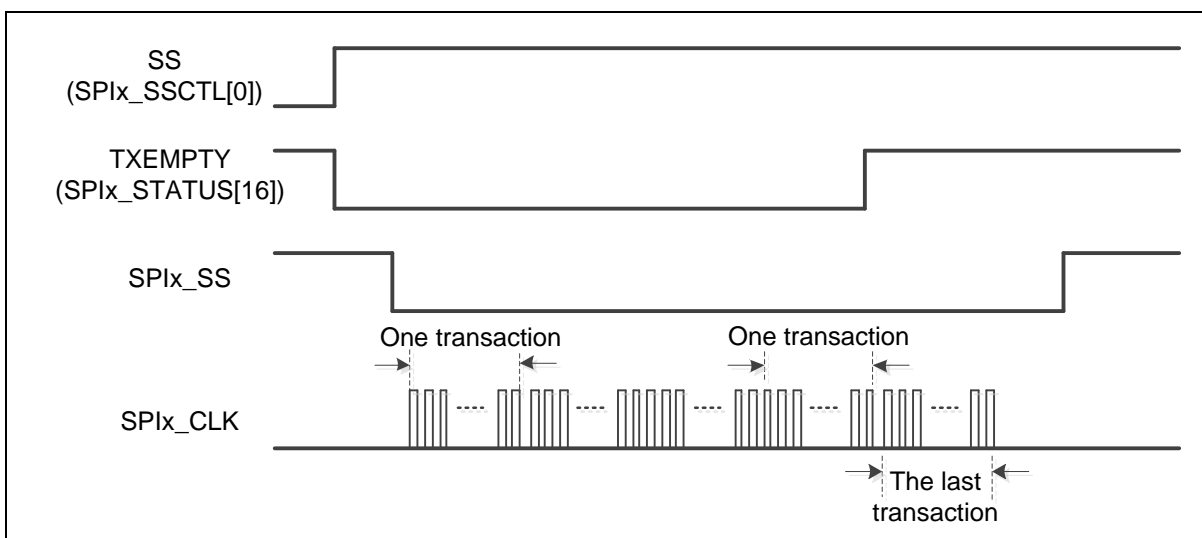


Figure 6.16-7 Automatic Slave Selection (SSACTPOL = 0, SUSPITV < 0x3)

### 6.16.5.3 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.

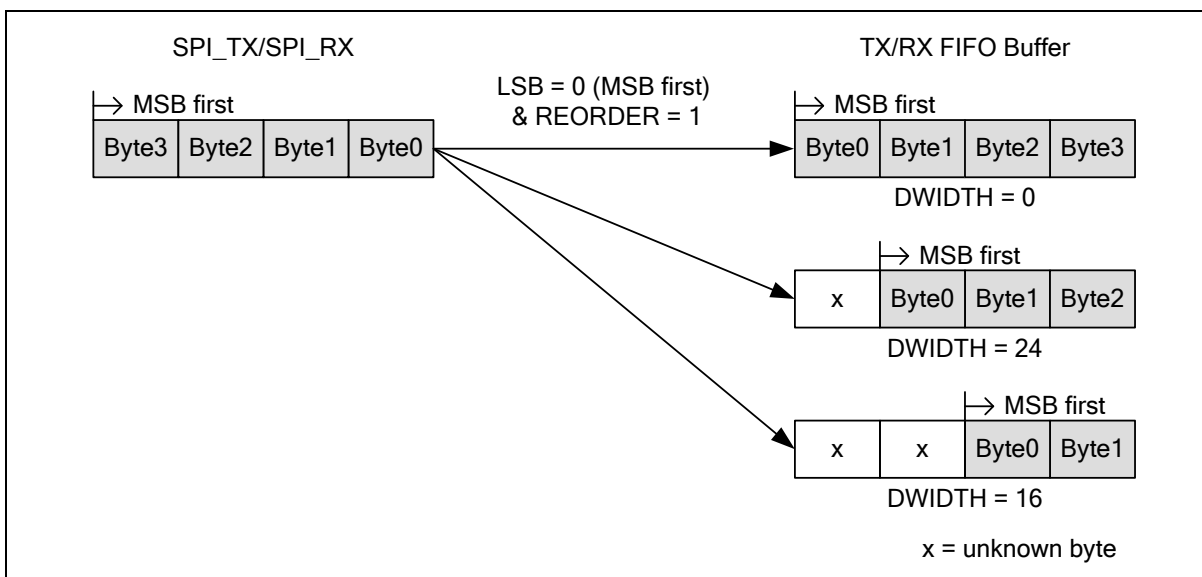


Figure 6.16-8 Byte Reorder Function

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]).

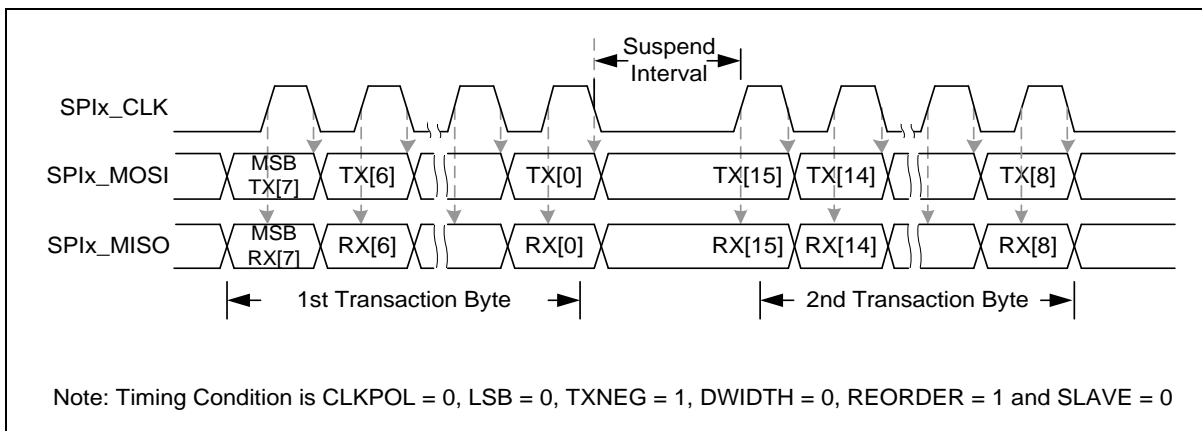


Figure 6.16-9 Timing Waveform for Byte Suspend

#### 6.16.5.4 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 TXRST (SPIx\_FIFOCTL[1]) and RXRST (SPIx\_FIFOCTL[0]) at the same time automatically.

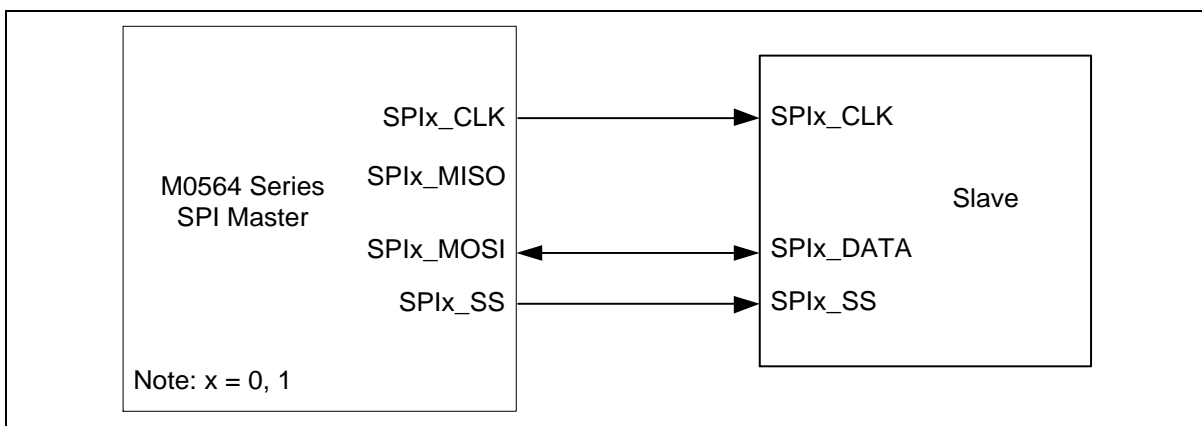


Figure 6.16-10 SPI Half-Duplex Master Mode Application Block Diagram

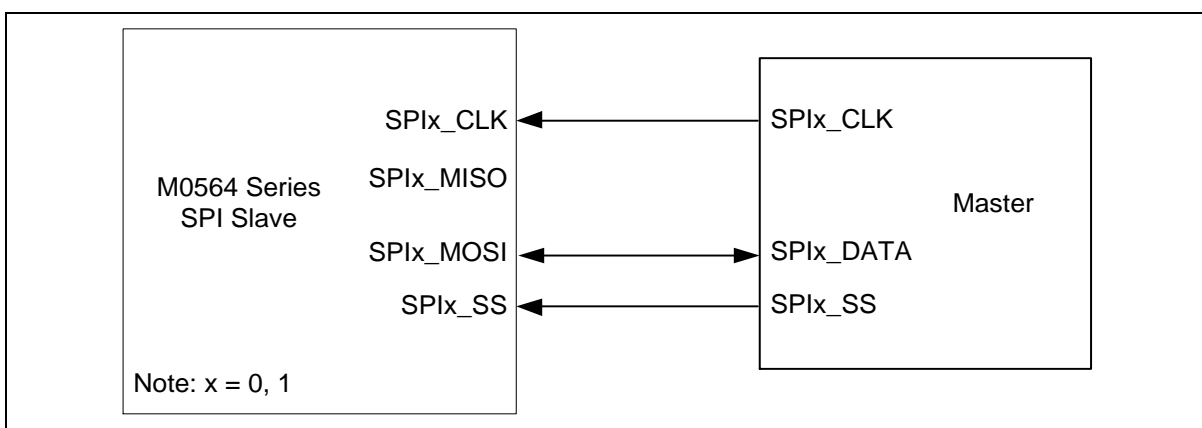


Figure 6.16-11 SPI Half-Duplex Slave Mode Application Block Diagram

#### 6.16.5.5 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 TXRST (SPIx\_FIFCTL[1]) and RXRST (SPIx\_FIFCTL[0]) at the same time automatically. After enabling this mode, the output SPI bus clock will be sent out in 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.

#### 6.16.5.6 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.

**Note:** SPI supports single request PDMA (Read/Write) only, burst request PDMA is not supported.

6.16.5.7 FIFO Buffer Operation

The SPI controllers are equipped 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[29:28]) and RXTH (SPIx\_FIFCTL[25:24]). When the count of valid data stored in transmit FIFO buffer is less than or equal to TXTH (SPIx\_FIFCTL[29:28]) 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[25:24]) setting, RXTHIF (SPIx\_STATUS[10]) will be set to 1.

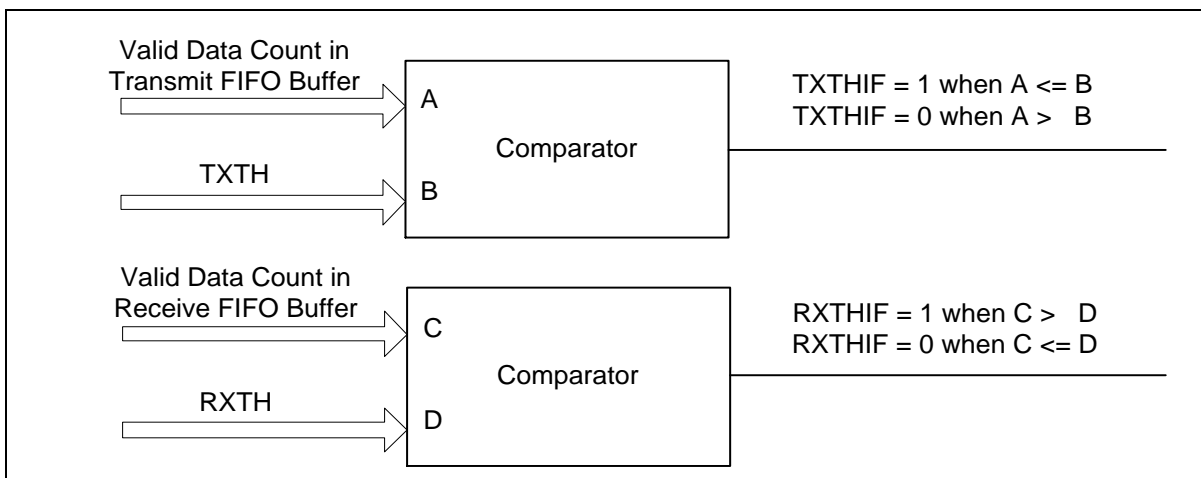


Figure 6.16-12 FIFO Threshold Comparator

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 APB 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.

In the example 1 of the following Figure, it indicates the updated condition of TXEMPTY (SPIx\_STATUS[16]) and the relationship among the FIFO buffer, shift register and the skew buffer. 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.

In the Example 2, it indicates the updated condition of TXFULL (SPIx\_STATUS[17]) when there are 4 data in the FIFO buffer and the next data of Data5 does not be written into the FIFO buffer when the

TXFULL = 1.

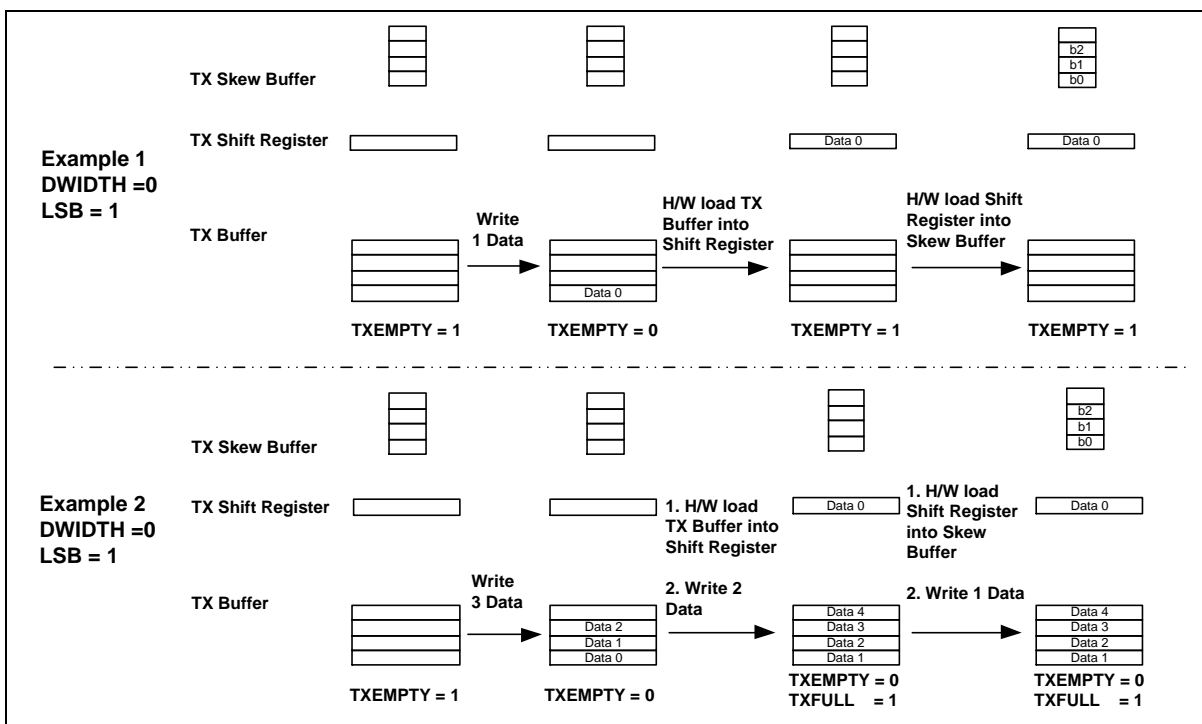


Figure 6.16-13 Transmit FIFO Buffer Example

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 transfer 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 reach 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). 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).



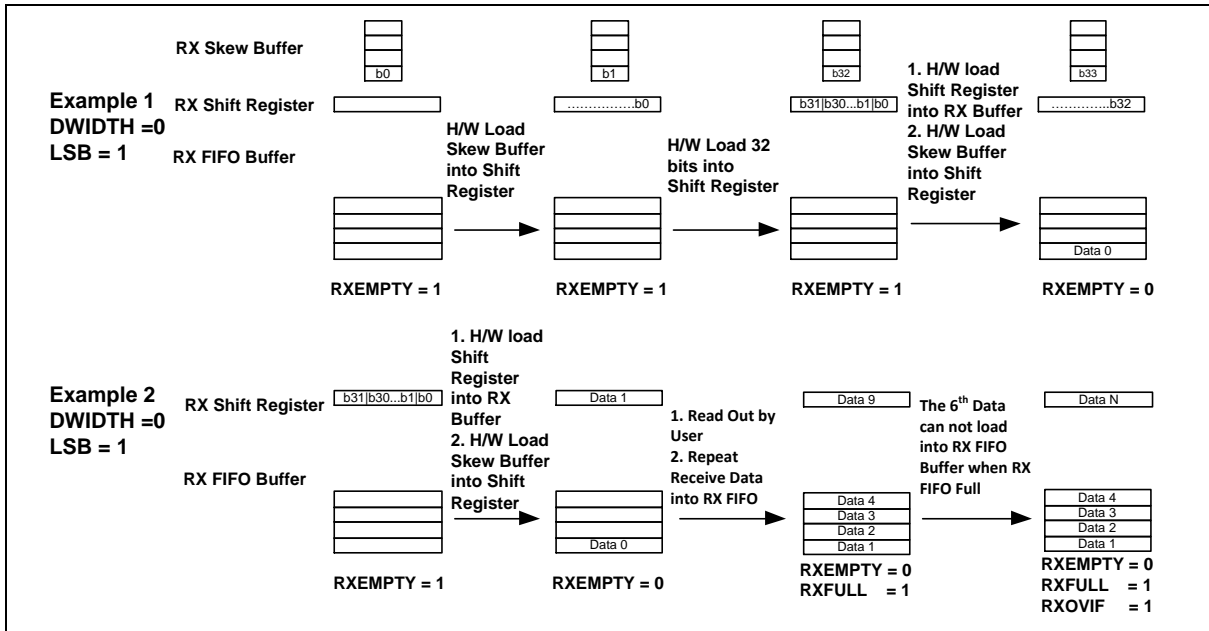


Figure 6.16-14 Receive FIFO Buffer Example

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.

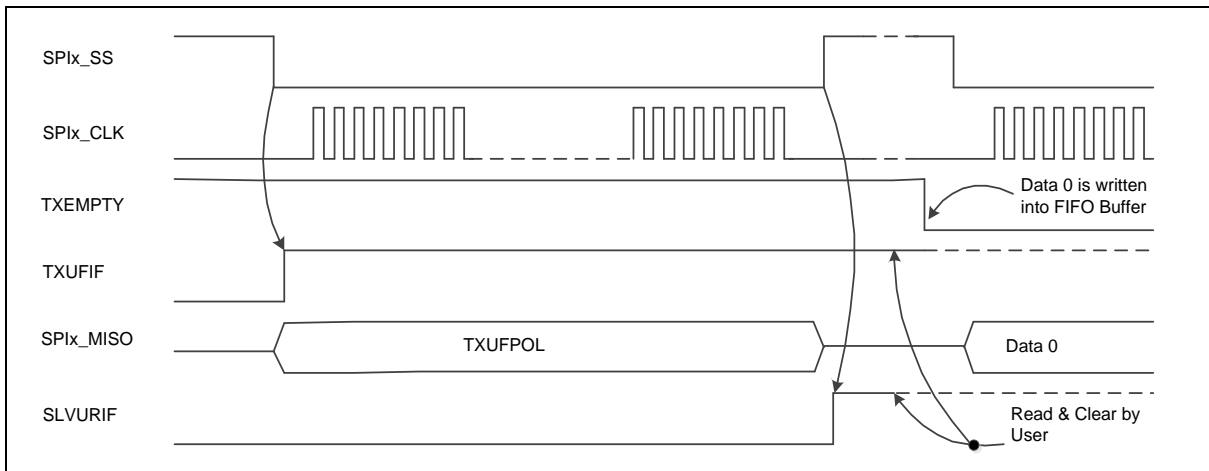


Figure 6.16-15 TX Underflow Event and Slave Under Run Event

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 Example figure). If the receive bit count mismatch 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.

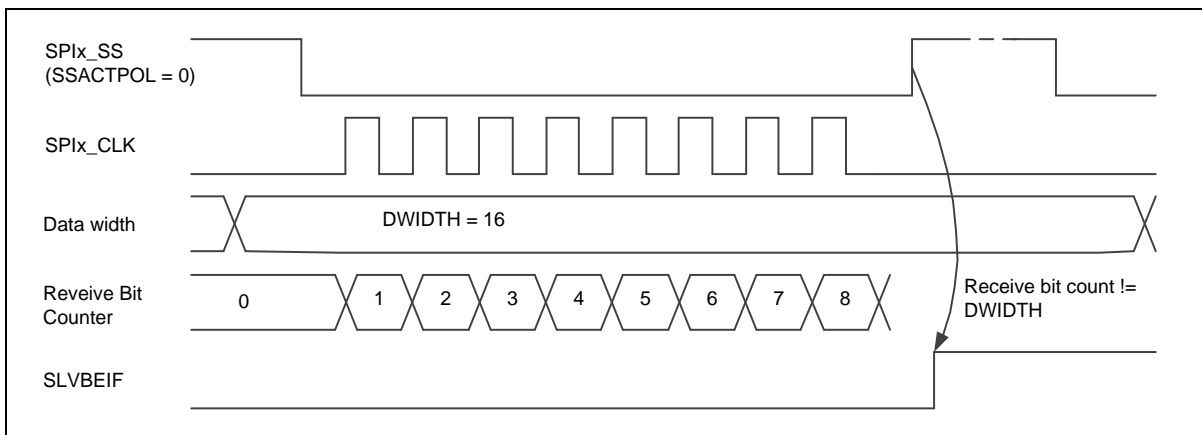


Figure 6.16-16 Slave Mode Bit Count Error

A receive time-out function is built-in in this controller. 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.

#### 6.16.5.8 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 SSINA IEN (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 mismatch 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 Underrun 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, 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 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[29:28]), 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[25:24]), 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.

6.16.5.9 I<sup>2</sup>S Mode

The SPI0 and SPI1 controllers support I<sup>2</sup>S mode with PCM mode A, PCM mode B, MSB justified and I<sup>2</sup>S 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 I<sup>2</sup>S 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.

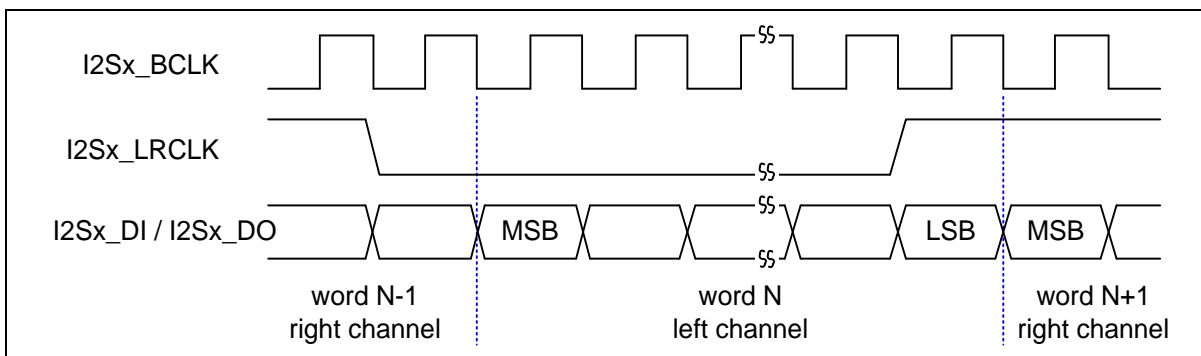


Figure 6.16-17 I<sup>2</sup>S Data Format Timing Diagram

In MSB justified data format, the MSB is sent and latched on the first clock of an audio channel.

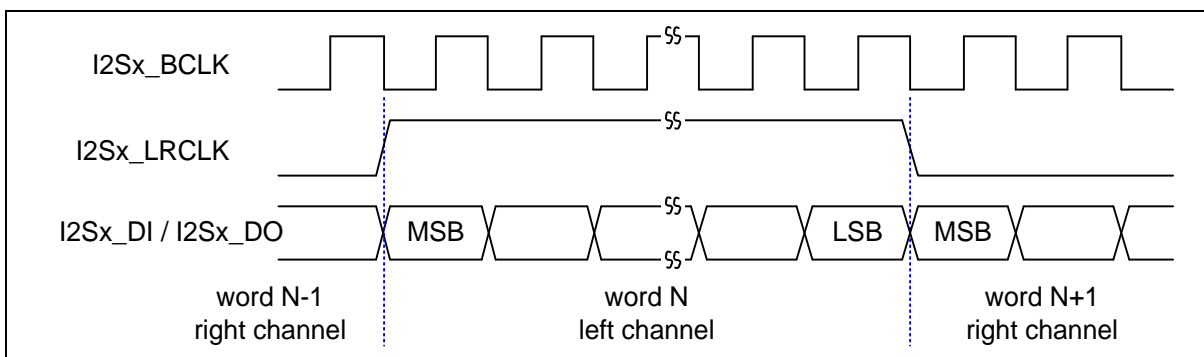


Figure 6.16-18 MSB Justified Data Format Timing Diagram

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.

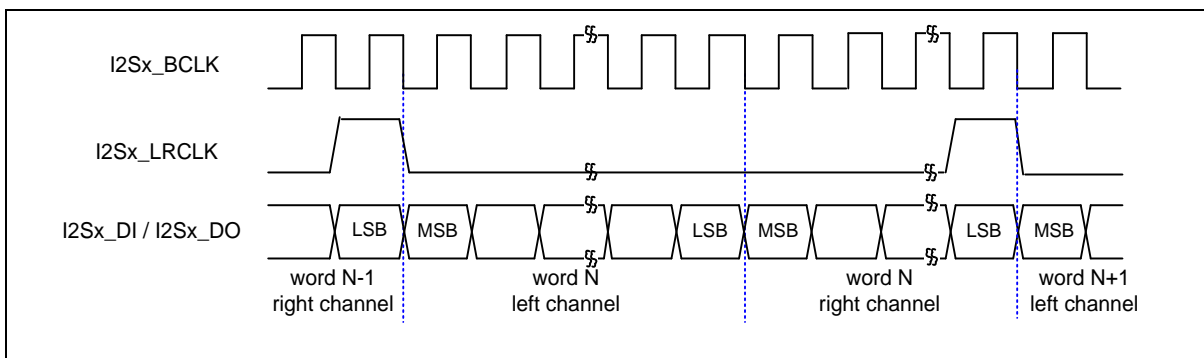


Figure 6.16-19 PCM Mode A Timing Diagram

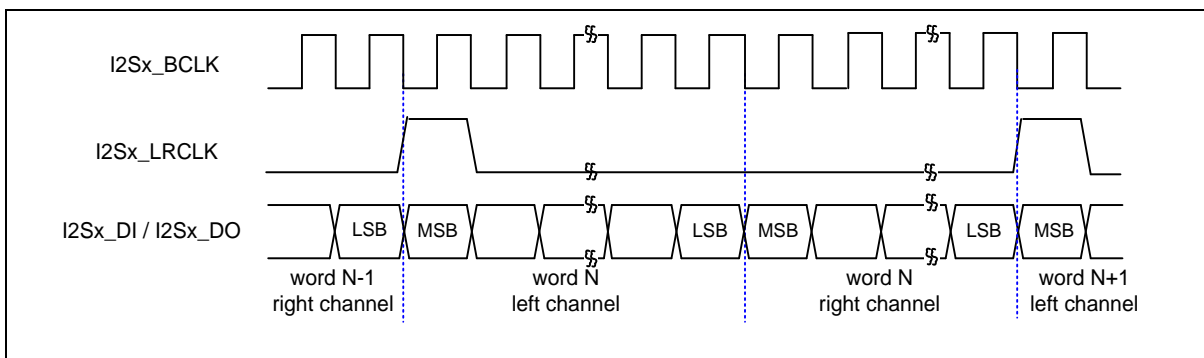


Figure 6.16-20 PCM Mode B Timing Diagram

6.16.5.10 I<sup>2</sup>S Mode FIFO operation

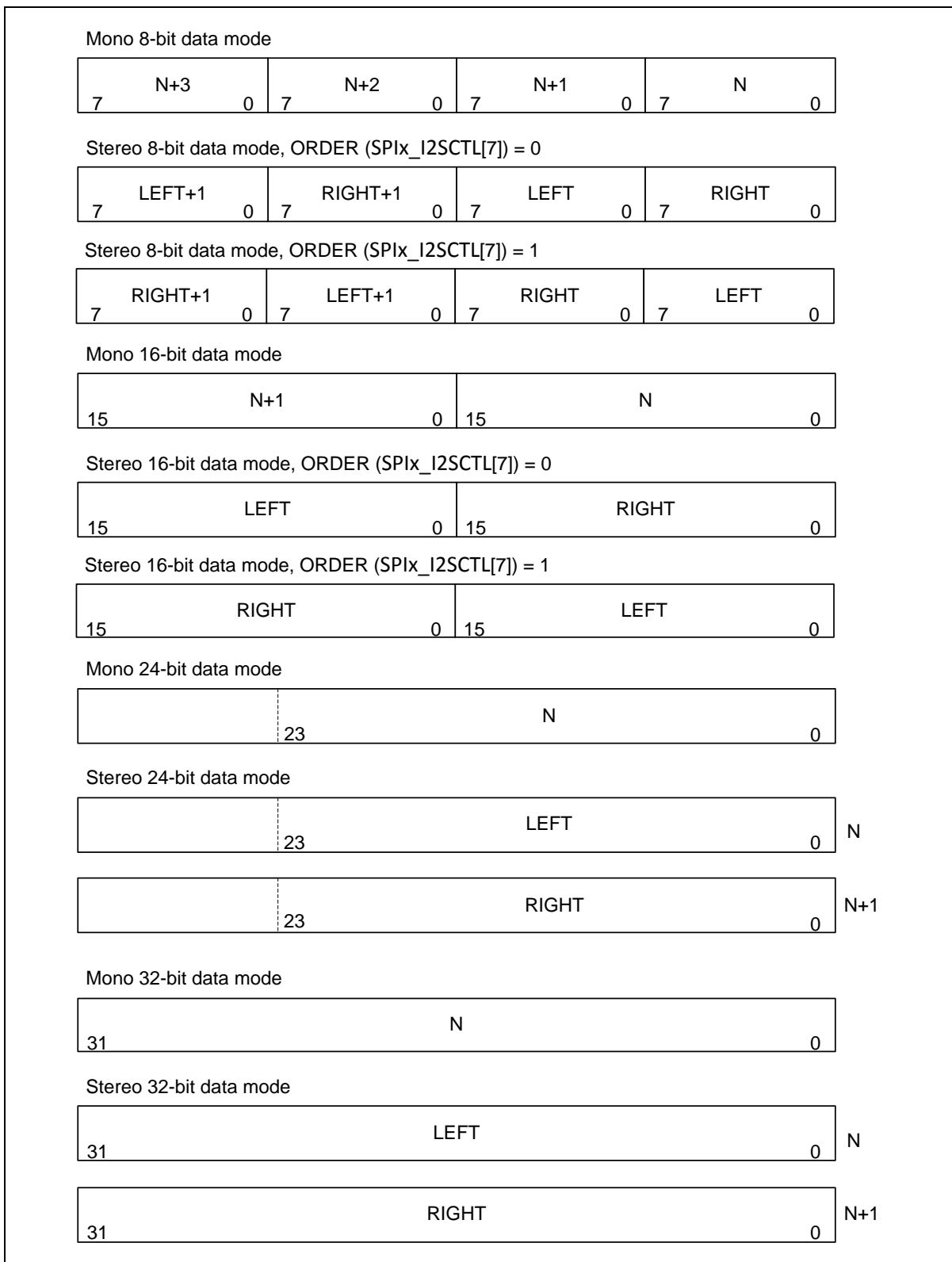


Figure 6.16-21 FIFO Contents for Various I<sup>2</sup>S Modes

### 6.16.6 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.

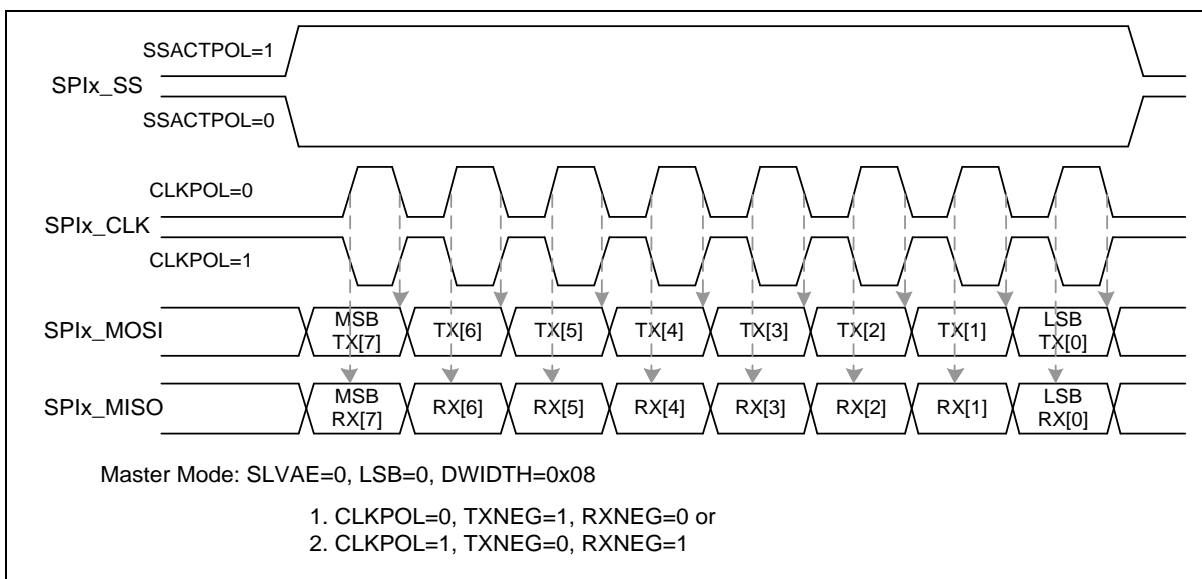


Figure 6.16-22 SPI Timing in Master Mode

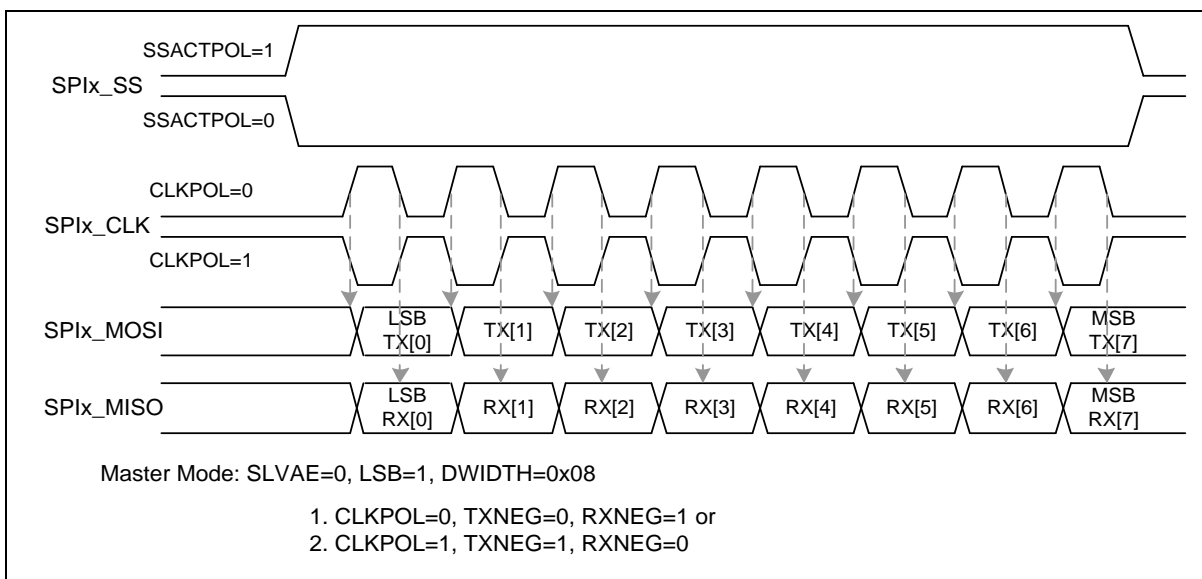


Figure 6.16-23 SPI Timing in Master Mode (Alternate Phase of SPIx\_CLK)

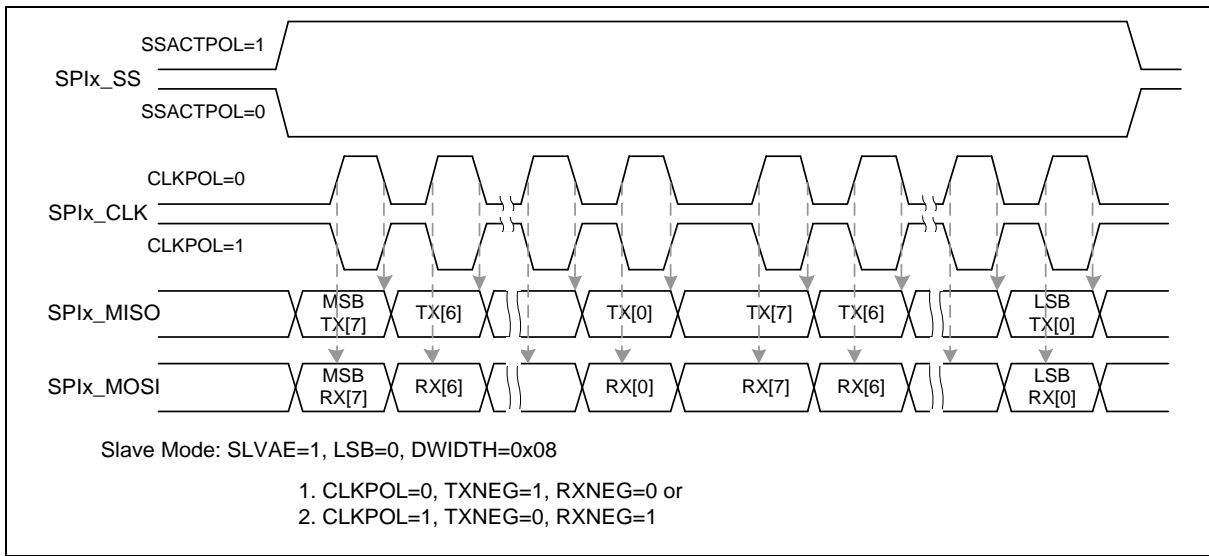


Figure 6.16-24 SPI Timing in Slave Mode

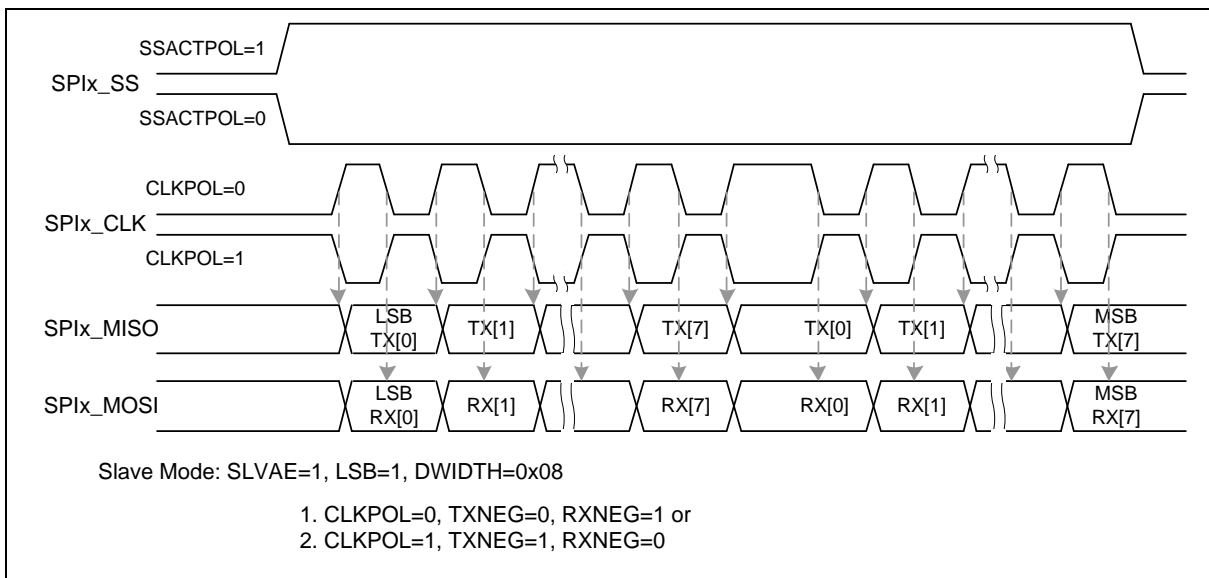


Figure 6.16-25 SPI Timing in Slave Mode (Alternate Phase of SPIx\_CLK)

### 6.16.7 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.
- Use 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 [7:0]) 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.  
Select high level for the input of slave selection signal by setting SSACTPOL (SPIx\_SSCTL[2]) to 1.
2. 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).
  - 6) Set LSB transfer first by setting LSB (SPIx\_CTL[13]) to 1.
3. 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.
4. 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.
5. 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.
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 4) to continue another data transfer or stop data transfer.

### 6.16.8 Register Map

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

Register	Offset	R/W	Description	Reset Value
<b>SPI Base Address:</b>				
<b>SPIx_BA = 0x4003_0000 + (0x0000_4000 * x)</b>				
x=0, 1				
<b>SPIx_CTL</b>	SPIx_BA+0x00	R/W	SPI Control Register	0x0000_0034
<b>SPIx_CLKDIV</b>	SPIx_BA+0x04	R/W	SPI Clock Divider Register	0x0000_0000
<b>SPIx_SSCTL</b>	SPIx_BA+0x08	R/W	SPI Slave Select Control Register	0x0000_0000
<b>SPIx_PDMACTL</b>	SPIx_BA+0x0C	R/W	SPI PDMA Control Register	0x0000_0000
<b>SPIx_FIFOCTL</b>	SPIx_BA+0x10	R/W	SPI FIFO Control Register	0x2200_0000
<b>SPIx_STATUS</b>	SPIx_BA+0x14	R/W	SPI Status Register	0x0005_0110
<b>SPIx_TX</b>	SPIx_BA+0x20	W	SPI Data Transmit Register	0x0000_0000
<b>SPIx_RX</b>	SPIx_BA+0x30	R	SPI Data Receive Register	0x0000_0000
<b>SPIx_I2SCTL</b>	SPIx_BA+0x60	R/W	I <sup>2</sup> S Control Register	0x0000_0000
<b>SPIx_I2SCLK</b>	SPIx_BA+0x64	R/W	I <sup>2</sup> S Clock Divider Control Register	0x0000_0000
<b>SPIx_I2SSTS</b>	SPIx_BA+0x68	R/W	I <sup>2</sup> S Status Register	0x0005_0100

### 6.16.9 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 I<sup>2</sup>S 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	<p><b>Data Port Direction Control</b></p> <p>This bit is used to select the data input/output direction in half-duplex transfer.</p> <p>0 = SPI data is input direction.</p> <p>1 = SPI data is output direction.</p>
[19]	REORDER	<p><b>Byte Reorder Function Enable Bit</b></p> <p>0 = Byte Reorder function Disabled.</p> <p>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.</p> <p><b>Note:</b> Byte Reorder function is only available if DWIDTH is defined as 16, 24, and 32 bits.</p>
[18]	SLAVE	<p><b>Slave Mode Control</b></p> <p>0 = Master mode.</p> <p>1 = Slave mode.</p>
[17]	UNITIEN	<p><b>Unit Transfer Interrupt Enable Bit</b></p> <p>0 = SPI unit transfer interrupt Disabled.</p> <p>1 = SPI unit transfer interrupt Enabled.</p>
[16]	Reserved	Reserved.
[15]	RXONLY	<p><b>Receive-only Mode Enable Bit (Master Only)</b></p> <p>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.</p> <p>0 = Receive-only mode Disabled.</p> <p>1 = Receive-only mode Enabled.</p>
[14]	HALFDPX	<p><b>SPI Half-duplex Transfer Enable Bit</b></p> <p>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.</p>

		0 = SPI operates in full-duplex transfer. 1 = SPI operates in half-duplex transfer.
[13]	LSB	<b>Send LSB First</b> 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	<b>Data Width</b> 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.
[7:4]	SUSPITV	<b>Suspend Interval (Master Only)</b> 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. $(SUSPITV[3:0] + 0.5) * \text{period of SPICLK clock cycle}$ 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	<b>Clock Polarity</b> 0 = SPI bus clock is idle low. 1 = SPI bus clock is idle high.
[2]	TXNEG	<b>Transmit on Negative Edge</b> 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	<b>Receive on Negative Edge</b> 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	<b>SPI Transfer Control Enable Bit</b> 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. <b>Note:</b> 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							
7	6	5	4	3	2	1	0
DIVIDER							

Bits	Description	
[31:8]	Reserved	Reserved.
[7:0]	DIVIDER	<p><b>Clock Divider</b></p> <p>The value in this field is the frequency divider for generating the peripheral clock, <math>f_{spi\_eclk}</math>, and the SPI bus clock of SPI Master. The frequency is obtained according to the following equation.</p> $f_{spi\_eclk} = \frac{f_{spi\_clock\_src}}{(DIVIDER + 1)}$ <p>where</p> <p><math>f_{spi\_clock\_src}</math> is the peripheral clock source, which is defined in the clock control register, CLK_CLKSEL2.</p> <p><b>Note:</b> Not supported in I<sup>2</sup>S mode.</p>

**Note:** User should set **DIVIDER** 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 I<sup>2</sup>S 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	<b>Slave Select Inactive Interrupt Enable Bit</b> 0 = Slave select inactive interrupt Disabled. 1 = Slave select inactive interrupt Enabled.
[12]	SSACTIEN	<b>Slave Select Active Interrupt Enable Bit</b> 0 = Slave select active interrupt Disabled. 1 = Slave select active interrupt Enabled.
[11:10]	Reserved	Reserved.
[9]	SLVURIEN	<b>Slave Mode TX Under Run Interrupt Enable Bit</b> 0 = Slave mode TX under run interrupt Disabled. 1 = Slave mode TX under run interrupt Enabled.
[8]	SLVBEIEN	<b>Slave Mode Bit Count Error Interrupt Enable Bit</b> 0 = Slave mode bit count error interrupt Disabled. 1 = Slave mode bit count error interrupt Enabled.
[7:4]	Reserved	Reserved.
[3]	AUTOSS	<b>Automatic Slave Selection Function Enable Bit (Master Only)</b> 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	<b>Slave Selection Active Polarity</b> 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]	<b>SS</b>	<p><b>Slave Selection Control (Master Only)</b></p> <p>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.</p> <p>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]).</p>
-----	-----------	--

**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	<b>PDMA Reset</b> 0 = No effect. 1 = Reset the PDMA control logic of the SPI controller. This bit will be automatically cleared to 0.
[1]	RXPDMAEN	<b>Receive PDMA Enable Bit</b> 0 = Receive PDMA function Disabled. 1 = Receive PDMA function Enabled.
[0]	TXPDMAEN	<b>Transmit PDMA Enable Bit</b> 0 = Transmit PDMA function Disabled. 1 = Transmit PDMA function Enabled. <b>Note:</b> 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\_FIFOCTL)**

Register	Offset	R/W	Description	Reset Value
SPIx_FIFOCTL	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
TXUFIE	TXUFPOL	RXOVIE	RXTOIE	TXTHIE	RXTHIE	TXRST	RXRST

Bits	Description	
[31:30]	Reserved	Reserved.
[29:28]	TXTH	<b>Transmit FIFO Threshold</b> 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.
[27:26]	Reserved	Reserved.
[25:24]	RXTH	<b>Receive FIFO Threshold</b> 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.
[23:10]	Reserved	Reserved.
[9]	TXFBCLR	<b>Transmit FIFO Buffer Clear</b> 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. <b>Note:</b> The TX shift register will not be cleared.
[8]	RXFBCLR	<b>Receive FIFO Buffer Clear</b> 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. <b>Note:</b> The RX shift register will not be cleared.
[7]	TXUFIE	<b>TX Underflow Interrupt Enable Bit</b> 0 = Slave TX underflow interrupt Disabled. 1 = Slave TX underflow interrupt Enabled.
[6]	TXUFPOL	<b>TX Underflow Data Polarity</b> 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.

		<p><b>Note:</b></p> <p>1. The TX underflow event occurs if there is no any data in TX FIFO when the slave selection signal is active.</p> <p>2. This bit should be set as 0 in I<sup>2</sup>S mode.</p> <p>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.</p>
[5]	<b>RXOVIEN</b>	<p><b>Receive FIFO Overrun Interrupt Enable Bit</b></p> <p>0 = Receive FIFO overrun interrupt Disabled.</p> <p>1 = Receive FIFO overrun interrupt Enabled.</p>
[4]	<b>RXTOIEN</b>	<p><b>Slave Receive Time-out Interrupt Enable Bit</b></p> <p>0 = Receive time-out interrupt Disabled.</p> <p>1 = Receive time-out interrupt Enabled.</p>
[3]	<b>TXTHIEN</b>	<p><b>Transmit FIFO Threshold Interrupt Enable Bit</b></p> <p>0 = TX FIFO threshold interrupt Disabled.</p> <p>1 = TX FIFO threshold interrupt Enabled.</p>
[2]	<b>RXTHIEN</b>	<p><b>Receive FIFO Threshold Interrupt Enable Bit</b></p> <p>0 = RX FIFO threshold interrupt Disabled.</p> <p>1 = RX FIFO threshold interrupt Enabled.</p>
[1]	<b>TXRST</b>	<p><b>Transmit Reset</b></p> <p>0 = No effect.</p> <p>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.</p> <p><b>Note:</b> If TX underflow event occurs in SPI Slave mode, this bit can be used to make SPI return to idle state.</p>
[0]	<b>RXRST</b>	<p><b>Receive Reset</b></p> <p>0 = No effect.</p> <p>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.</p>

**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 I<sup>2</sup>S mode.

31	30	29	28	27	26	25	24
TXCNT				RXCNT			
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	RXEMPTY
7	6	5	4	3	2	1	0
SLVURIF	SLVBEIF	Reserved	SSLINE	SSINAIF	SSACTIF	UNITIF	BUSY

Bits	Description	
[31:28]	TXCNT	<b>Transmit FIFO Data Count (Read Only)</b> This bit field indicates the valid data count of transmit FIFO buffer.
[27:24]	RXCNT	<b>Receive FIFO Data Count (Read Only)</b> This bit field indicates the valid data count of receive FIFO buffer.
[23]	TXRXRST	<b>TX or RX Reset Status (Read Only)</b> 0 = The reset function of TXRST or RXRST is done. 1 = Doing the reset function of TXRST or RXRST. <b>Note:</b> 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	<b>TX Underflow Interrupt Flag</b> 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. <b>Note1:</b> This bit will be cleared by writing 1 to it. <b>Note2:</b> 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	<b>Transmit FIFO Threshold Interrupt Flag (Read Only)</b> 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	<b>Transmit FIFO Buffer Full Indicator (Read Only)</b> 0 = Transmit FIFO buffer is not full.

		1 = Transmit FIFO buffer is full.
[16]	<b>TXEMPTY</b>	<b>Transmit FIFO Buffer Empty Indicator (Read Only)</b> 0 = Transmit FIFO buffer is not empty. 1 = Transmit FIFO buffer is empty.
[15]	<b>SPIENSTS</b>	<b>SPI Enable Status (Read Only)</b> 0 = SPI controller Disabled. 1 = SPI controller Enabled. <b>Note:</b> 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]	<b>Reserved</b>	Reserved.
[12]	<b>RXTOIF</b>	<b>Receive Time-out Interrupt Flag</b> 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. <b>Note:</b> This bit will be cleared by writing 1 to it.
[11]	<b>RXOVIF</b>	<b>Receive FIFO Overrun Interrupt Flag</b> 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. <b>Note:</b> This bit will be cleared by writing 1 to it.
[10]	<b>RXTHIF</b>	<b>Receive FIFO Threshold Interrupt Flag (Read Only)</b> 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]	<b>RXFULL</b>	<b>Receive FIFO Buffer Full Indicator (Read Only)</b> 0 = Receive FIFO buffer is not full. 1 = Receive FIFO buffer is full.
[8]	<b>RXEMPTY</b>	<b>Receive FIFO Buffer Empty Indicator (Read Only)</b> 0 = Receive FIFO buffer is not empty. 1 = Receive FIFO buffer is empty.
[7]	<b>SLVURIF</b>	<b>Slave Mode TX Under Run Interrupt Flag</b> 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 occurs. <b>Note:</b> This bit will be cleared by writing 1 to it.
[6]	<b>SLVBEIF</b>	<b>Slave Mode Bit Count Error Interrupt Flag</b> In Slave mode, when the slave select line goes to inactive state, if bit counter is mismatch 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 occurs. <b>Note:</b> 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]	<b>Reserved</b>	Reserved.

[4]	SSLINE	<p><b>Slave Select Line Bus Status (Read Only)</b></p> <p>0 = The slave select line status is 0. 1 = The slave select line status is 1.</p> <p><b>Note:</b> 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.</p>
[3]	SSINAIF	<p><b>Slave Select Inactive Interrupt Flag</b></p> <p>0 = Slave select inactive interrupt was cleared or not occurred. 1 = Slave select inactive interrupt event occurred.</p> <p><b>Note:</b> Only available in Slave mode. This bit will be cleared by writing 1 to it.</p>
[2]	SSACTIF	<p><b>Slave Select Active Interrupt Flag</b></p> <p>0 = Slave select active interrupt was cleared or not occurred. 1 = Slave select active interrupt event occurred.</p> <p><b>Note:</b> Only available in Slave mode. This bit will be cleared by writing 1 to it.</p>
[1]	UNITIF	<p><b>Unit Transfer Interrupt Flag</b></p> <p>0 = No transaction has been finished since this bit was cleared to 0. 1 = SPI controller has finished one unit transfer.</p> <p><b>Note:</b> This bit will be cleared by writing 1 to it.</p>
[0]	BUSY	<p><b>Busy Status (Read Only)</b></p> <p>0 = SPI controller is in idle state. 1 = SPI controller is in busy state.</p> <p>The following listing are the bus busy conditions:</p> <ul style="list-style-type: none"> <li>a. SPIx_CTL[0] = 1 and TXEMPTY = 0.</li> <li>b. For SPI Master mode, SPIx_CTL[0] = 1 and TXEMPTY = 1 but the current transaction is not finished yet.</li> <li>c. For SPI Master mode, SPIx_CTL[0] = 1 and RXONLY = 1.</li> <li>d. For SPI Slave mode, the SPIx_CTL[0] = 1 and there is serial clock input into the SPI core logic when slave select is active.</li> </ul> <p>For SPI Slave mode, the SPIx_CTL[0] = 1 and the transmit buffer or transmit shift register is not empty even if the slave select is inactive.</p>

**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]	<p><b>TX</b></p> <p><b>Data Transmit Register</b>                      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 I<sup>2</sup>S mode.</p> <p>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.</p> <p>In I<sup>2</sup>S 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 I<sup>2</sup>S mode FIFO operation section</p> <p><b>Note:</b> 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.</p>

**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]	<p><b>RX</b></p> <p><b>Data Receive Register</b>                      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. This is a read only register.</p>

I<sup>2</sup>S Control Register (SPIx\_I2SCTL)

Register	Offset	R/W	Description	Reset Value
SPIx_I2SCTL	SPIx_BA+0x60	R/W	I <sup>2</sup> S Control Register	0x0000_0000

Note: Not supported in SPI mode.

31	30	29	28	27	26	25	24
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	
[31:30]	Reserved	Reserved.
[29:28]	FORMAT	<b>Data Format Selection</b> 00 = I <sup>2</sup> S data format. 01 = MSB justified data format. 10 = PCM mode A. 11 = PCM mode B.
[27:26]	Reserved	Reserved.
[25]	LZCIEN	<b>Left Channel Zero Cross Interrupt Enable Bit</b> Interrupt occurs if this bit is set to 1 and left channel zero cross event occurs. 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[24]	RZCIEN	<b>Right Channel Zero Cross Interrupt Enable Bit</b> Interrupt occurs if this bit is set to 1 and right channel zero cross event occurs. 0 = Interrupt Disabled. 1 = Interrupt Enabled.
[23]	RXLCH	<b>Receive Left Channel Enable Bit</b> When monaural format is selected (MONO = 1), I <sup>2</sup> 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	<b>Left Channel Zero Cross Detection Enable Bit</b> 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	<p><b>Right Channel Zero Cross Detection Enable Bit</b></p> <p>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.</p> <p>0 = Right channel zero cross detection Disabled. 1 = Right channel zero cross detection Enabled.</p>
[15]	MCLKEN	<p><b>Master Clock Enable Bit</b></p> <p>If MCLKEN is set to 1, I<sup>2</sup>S controller will generate master clock on SPIx_I2SMCLK pin for external audio devices.</p> <p>0 = Master clock Disabled. 1 = Master clock Enabled.</p>
[14:9]	Reserved	Reserved.
[8]	SLAVE	<p><b>Slave Mode</b></p> <p>I<sup>2</sup>S can operate as master or slave. For Master mode, I2Sx_BCLK and I2Sx_LRCLK pins are output mode and send bit clock from the NUC126 series 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.</p> <p>0 = Master mode. 1 = Slave mode.</p>
[7]	ORDER	<p><b>Stereo Data Order in FIFO</b></p> <p>0 = Left channel data at high byte. 1 = Left channel data at low byte.</p>
[6]	MONO	<p><b>Monaural Data</b></p> <p>0 = Data is stereo format. 1 = Data is monaural format.</p>
[5:4]	WDWIDTH	<p><b>Word Width</b></p> <p>00 = data size is 8-bit. 01 = data size is 16-bit. 10 = data size is 24-bit. 11 = data size is 32-bit.</p>
[3]	MUTE	<p><b>Transmit Mute Enable Bit</b></p> <p>0 = Transmit data is shifted from buffer. 1 = Transmit channel zero.</p>
[2]	RXEN	<p><b>Receive Enable Bit</b></p> <p>0 = Data receive Disabled. 1 = Data receive Enabled.</p>
[1]	TXEN	<p><b>Transmit Enable Bit</b></p> <p>0 = Data transmit Disabled. 1 = Data transmit Enabled.</p>
[0]	I2SEN	<p><b>I<sup>2</sup>S Controller Enable Bit</b></p> <p>0 = I<sup>2</sup>S mode Disabled. 1 = I<sup>2</sup>S mode Enabled.</p> <p><b>Note:</b></p> <ol style="list-style-type: none"> <li>If this bit is enabled, I2Sx_BCLK will start to output in Master mode.</li> <li>Before changing the configurations of SPIx_I2SCTL, SPIx_I2SCLK, and SPIx_FIFCTL registers, user shall clear the I2SEN (SPIx_I2SCTL[0]) and confirm the I2SENSTS (SPIx_I2SSTS[15]) is 0.</li> </ol>

**I<sup>2</sup>S Clock Divider Control Register (SPIx\_I2SCLK)**

Register	Offset	R/W	Description	Reset Value
SPIx_I2SCLK	SPIx_BA+0x64	R/W	I <sup>2</sup> S Clock Divider Control Register	0x0000_0000

Note: Not supported in SPI mode.

31	30	29	28	27	26	25	24
Reserved							
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	Description
[31:17]	Reserved	Reserved.
[16:8]	BCLKDIV	<p><b>Bit Clock Divider</b></p> <p>The I<sup>2</sup>S controller will generate bit clock in Master mode. The clock frequency of bit clock, <math>f_{BCLK}</math>, is determined by the following expression:</p> $f_{BCLK} = \frac{f_{i2s\_clock\_src}}{2 \times (BCLKDIV + 1)}$ <p>where</p> <p><math>f_{i2s\_clock\_src}</math> is the frequency of I<sup>2</sup>S peripheral clock source, which is defined in the clock control register CLK_CLKSEL2.</p> <p>In I<sup>2</sup>S Slave mode, this field is used to define the frequency of peripheral clock and it's determined by <math>f_{i2s\_clock\_src} \div \left( \frac{BCLKDIV}{2} + 1 \right)</math>.</p> <p>The peripheral clock frequency in I<sup>2</sup>S Slave mode must be equal to or faster than 6 times of input bit clock.</p>
[7:6]	Reserved	Reserved.
[5:0]	MCLKDIV	<p><b>Master Clock Divider</b></p> <p>If MCLKEN is set to 1, I<sup>2</sup>S controller will generate master clock for external audio devices. The frequency of master clock, <math>f_{MCLK}</math>, is determined by the following expressions:</p> <p>If MCLKDIV &gt;= 1, <math>f_{MCLK} = \frac{f_{i2s\_clock\_src}}{2 \times MCLKDIV}</math></p> <p>If MCLKDIV = 0, <math>f_{MCLK} = f_{i2s\_clock\_src}</math></p> <p>where</p> <p><math>f_{i2s\_clock\_src}</math> is the frequency of I<sup>2</sup>S 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.</p>

**Note:** User should set BCLKDIV carefully because the peripheral clock frequency must be slower than or equal to system frequency

**I<sup>2</sup>S Status Register (SPIx\_I2SSTS)**

Register	Offset	R/W	Description	Reset Value
SPIx_I2SSTS	SPIx_BA+0x68	R/W	I2S 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	Reserved	LZCIF	RZCIF	TXUFIF	TXTHIF	TXFULL	TXEMPTY
15	14	13	12	11	10	9	8
I2SENSTS	Reserved		RXTOIF	RXOVIF	RXTHIF	RXFULL	RXEMPTY
7	6	5	4	3	2	1	0
Reserved			RIGHT	Reserved			

Bits	Description
[31]	Reserved. Reserved.
[30:28]	<b>TXCNT</b> Transmit FIFO Data Count (Read Only) This bit field indicates the valid data count of transmit FIFO buffer.
[27]	Reserved. Reserved.
[26:24]	<b>RXCNT</b> Receive FIFO Data Count (Read Only) This bit field indicates the valid data count of receive FIFO buffer.
[23]	<b>TXRXRST</b> <b>TX or RX Reset Status (Read Only)</b> 0 = The reset function of TXRST or RXRST is done. 1 = Doing the reset function of TXRST or RXRST. <b>Note:</b> 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]	Reserved. Reserved.
[21]	<b>LZCIF</b> <b>Left Channel Zero Cross Interrupt Flag</b> 0 = No zero cross event occurred on left channel. 1 = Zero cross event occurred on left channel.
[20]	<b>RZCIF</b> <b>Right Channel Zero Cross Interrupt Flag</b> 0 = No zero cross event occurred on right channel. 1 = Zero cross event occurred on right channel.
[19]	<b>TXUFIF</b> <b>Transmit FIFO Underflow Interrupt Flag</b> 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. <b>Note:</b> This bit will be cleared by writing 1 to it.
[18]	<b>TXTHIF</b> <b>Transmit FIFO Threshold Interrupt Flag (Read Only)</b> 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. <b>Note:</b> If TXTHIEN = 1 and TXTHIF = 1, the SPI/I <sup>2</sup> S controller will generate a SPI interrupt request.
[17]	TXFULL	<b>Transmit FIFO Buffer Full Indicator (Read Only)</b> 0 = Transmit FIFO buffer is not full. 1 = Transmit FIFO buffer is full.
[16]	TXEMPTY	<b>Transmit FIFO Buffer Empty Indicator (Read Only)</b> 0 = Transmit FIFO buffer is not empty. 1 = Transmit FIFO buffer is empty.
[15]	I2SENSTS	<b>I<sup>2</sup>S Enable Status (Read Only)</b> 0 = SPI/I <sup>2</sup> S control logic Disabled. 1 = SPI/I <sup>2</sup> S control logic Enabled. <b>Note:</b> The SPI peripheral clock is asynchronous with the system clock. In order to make sure the SPI/I <sup>2</sup> S control logic is disabled, this bit indicates the real status of SPI/I <sup>2</sup> S control logic for user.
[14:13]	Reserved	Reserved.
[12]	RXTOIF	<b>Receive Time-out Interrupt Flag</b> 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. <b>Note:</b> This bit will be cleared by writing 1 to it.
[11]	RXOVIF	<b>Receive FIFO Overrun Interrupt Flag</b> When the receive FIFO buffer is full, the follow-up data will be dropped and this bit will be set to 1. <b>Note:</b> This bit will be cleared by writing 1 to it.
[10]	RXTHIF	<b>Receive FIFO Threshold Interrupt Flag (Read Only)</b> 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. <b>Note:</b> If RXTHIEN = 1 and RXTHIF = 1, the SPI/I <sup>2</sup> S controller will generate a SPI interrupt request.
[9]	RXFULL	<b>Receive FIFO Buffer Full Indicator (Read Only)</b> 0 = Receive FIFO buffer is not full. 1 = Receive FIFO buffer is full.
[8]	RXEMPTY	<b>Receive FIFO Buffer Empty Indicator (Read Only)</b> 0 = Receive FIFO buffer is not empty. 1 = Receive FIFO buffer is empty.
[7:5]	Reserved	Reserved.
[4]	RIGHT	<b>Right Channel (Read Only)</b> This bit indicates the current transmit data is belong to which channel. 0 = Left channel. 1 = Right channel.
[3:0]	Reserved	Reserved.

## 6.17 Timer Controller (TMR)

### 6.17.1 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.

The Timer controller also provides four PWM generators. Each PWM generator supports two PWM output channels in independent mode and complementary mode. The output state of PWM output pin can be control by pin mask, polarity and break control, and dead-time generator.

### 6.17.2 Features

#### 6.17.2.1 Timer Function Features

- Four sets of 32-bit timers, each timer equips 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 event for interval measurement
- Supports external capture pin event to reset 24-bit up counter
- 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 PWM, ADC and PDMA function
- Supports internal capture triggered while internal ACMP output signal transition
- Supports Inter-Timer trigger mode
- Supports event counting source from internal USB SOF signal

#### 6.17.2.2 PWM Function Features

- Supports maximum clock frequency up to 72MHz
- Supports independent mode for PWM generator with two output channels
- Supports complementary mode for PWM generator with paired PWM output channel
  - 12-bit dead-time insertion with 12-bit prescale
- Supports 12-bit prescale from 1 to 4096
- Supports 16-bit PWM counter
  - Up, down and up-down count operation type
  - One-shot or auto-reload counter operation mode
- Supports mask function and tri-state enable for each PWM output pin
- Supports brake function
  - Brake source from pin, analog comparator and system safety events (clock failed,

- Brown-out detection and CPU lockup)
- Brake pin noise filter control for brake source
- Edge detect brake source to control brake state until brake interrupt cleared
- Level detect brake source to auto recover function after brake condition removed
- Supports interrupt on the following events:
  - PWM zero point, period point, up-count compared or down-count compared point events
  - Brake condition happened
- Supports trigger ADC on the following events:
  - PWM zero point, period, zero or period point, up-count compared or down-count compared point events

### 6.17.3 Block Diagram

The Timer Controller block diagram and clock control are shown as follows.

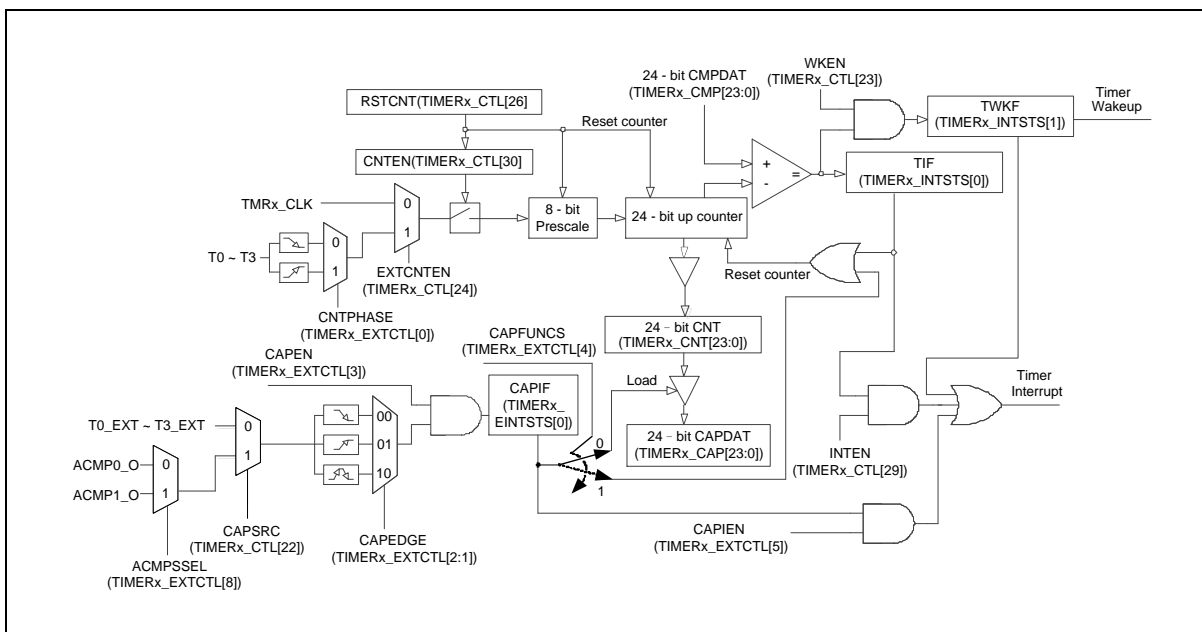


Figure 6.17-1 Timer Controller Block Diagram

Set FUNMODE (TIMERx\_ALTCTL[0]) 0 to enable timer mode. 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 6.17-2.



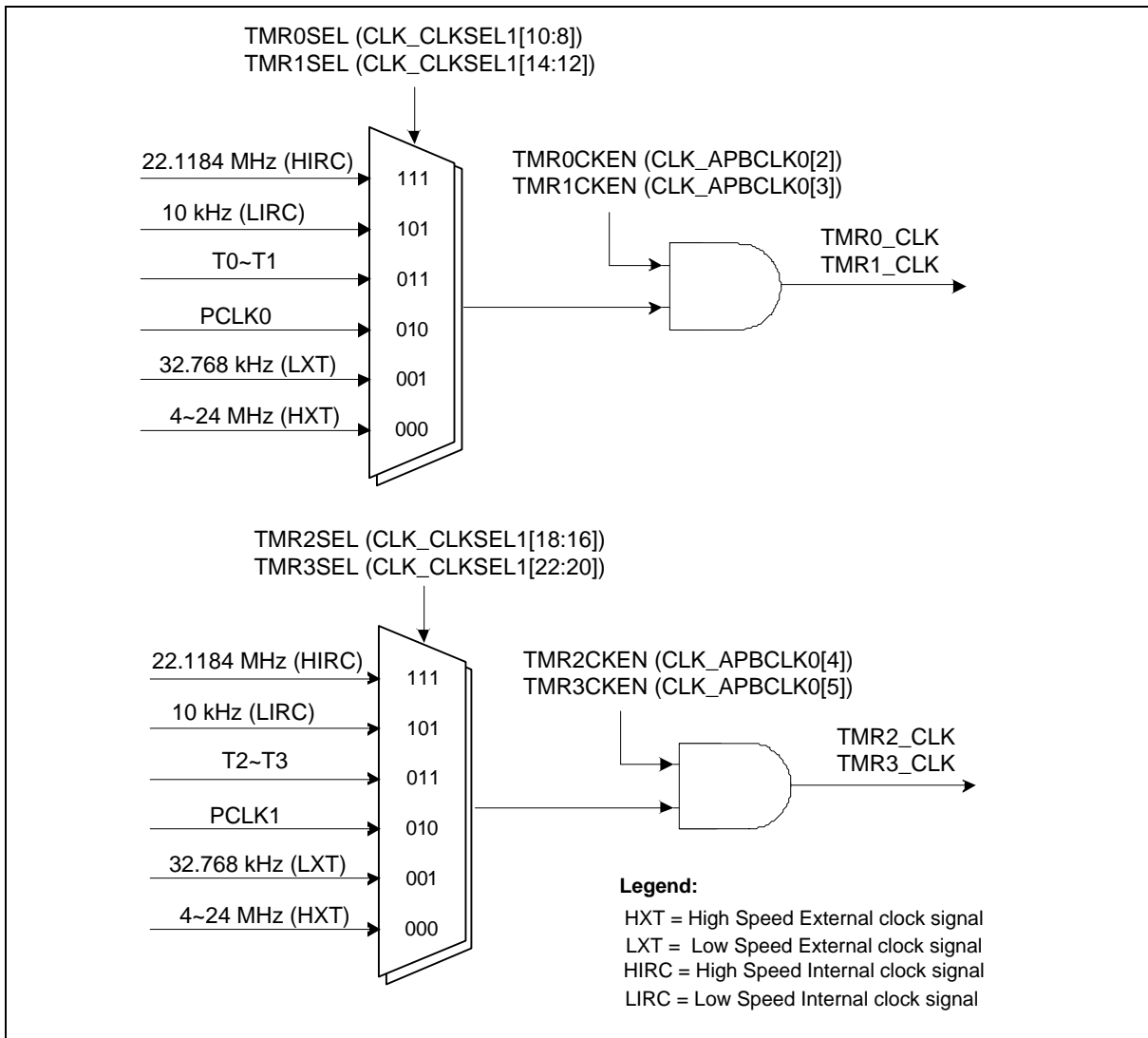


Figure 6.17-2 Clock Source of Timer Controller

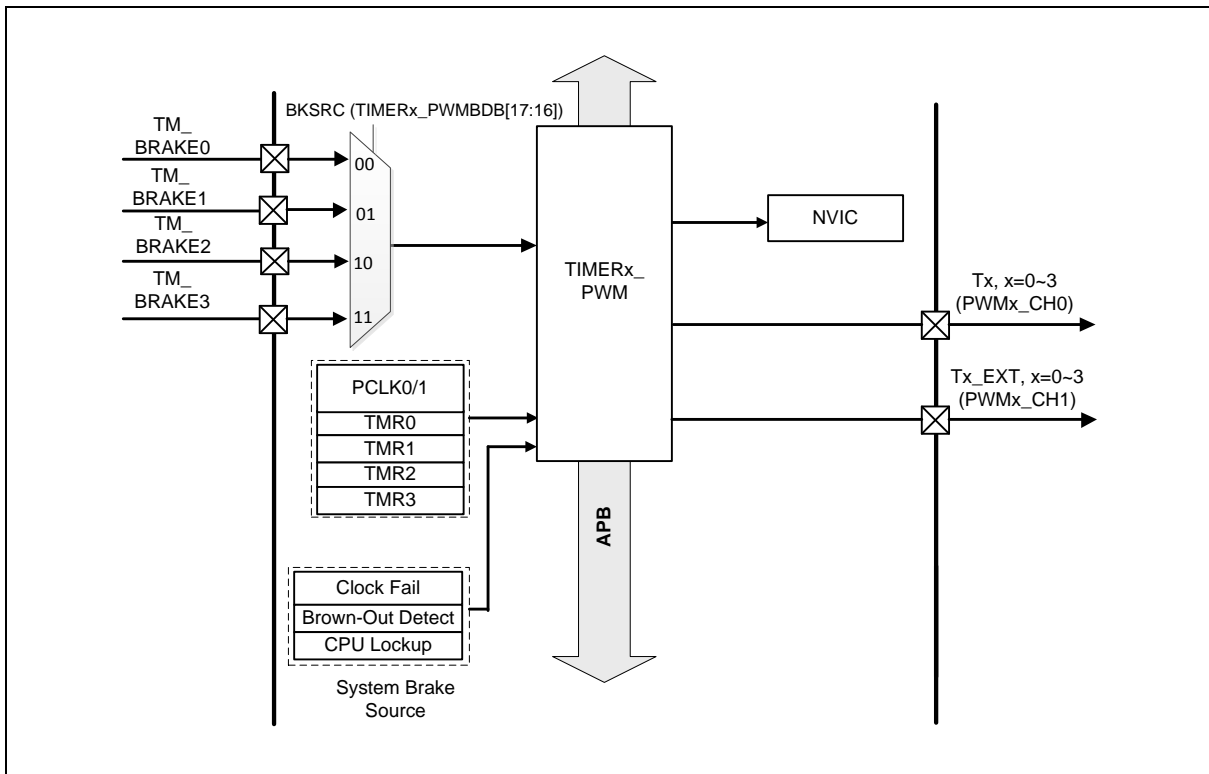


Figure 6.17-3 PWM Generator Overview Block Diagram

Set FUNMODE (TIMERx\_ALTCTL[0]) 1 to enable PWM mode. The clock source of Timer0 ~ Timer3 in PWM mode can be enabled in TMRxCKEN (CLK\_APBCLK0[5:2]). TMR0\_CLK and TMR1\_CLK clock sources are fixed to PCLK0. TMR2\_CLK and TMR3\_CLK clock sources are fixed to PCLK1. PWM system clock frequency will be PCLKx frequency as Figure 6.17-4 .

The clock source of PWM counter (TIMERx\_PWMCLK) can be selected from PWM system clock (TMRx\_CLK) or Timer interrupt events (TMRx\_INT) as Figure 6.17-5.

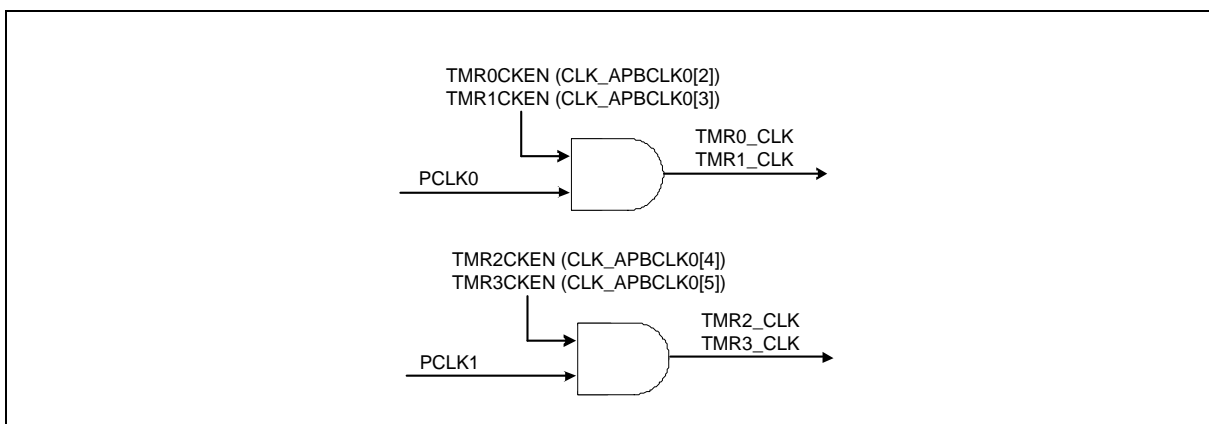


Figure 6.17-4 PWM System Clock Source Control

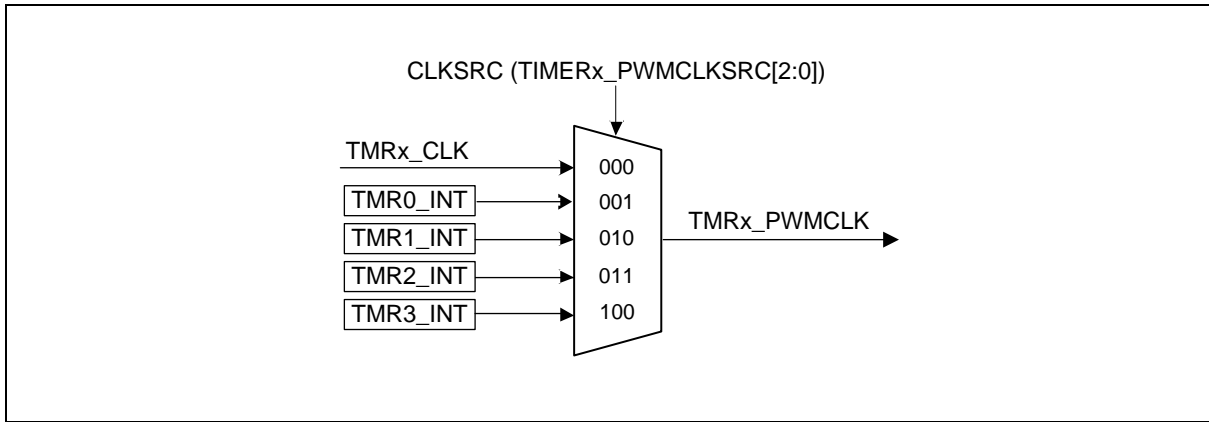


Figure 6.17-5 PWM Counter Clock Source Control

Figure 6.17-6 and Figure 6.17-7 illustrate the architecture of PWM independent mode and complementary mode. Both independent mode and complementary mode supports PWMx\_CH0 and PWMx\_CH1 output channels in each PWM generator.

When counter count to 0, PERIOD (TIMERx\_PWMPERIOD[15:0]) or equal to CMP (TIMERx\_PWMCMPDAT[15:0]), relative events will be generated. These events are passed to corresponding generators to generate PWM pulse (Pulse Generator), interrupt signal (Interrupt Generator) and trigger signal (Trigger Generator) for ADC to start conversion. Output Control block is used to decide PWM pulse output; brake function in Output Control block also generates interrupt events. And Dead-Time Control is available only in PWM complementary mode.

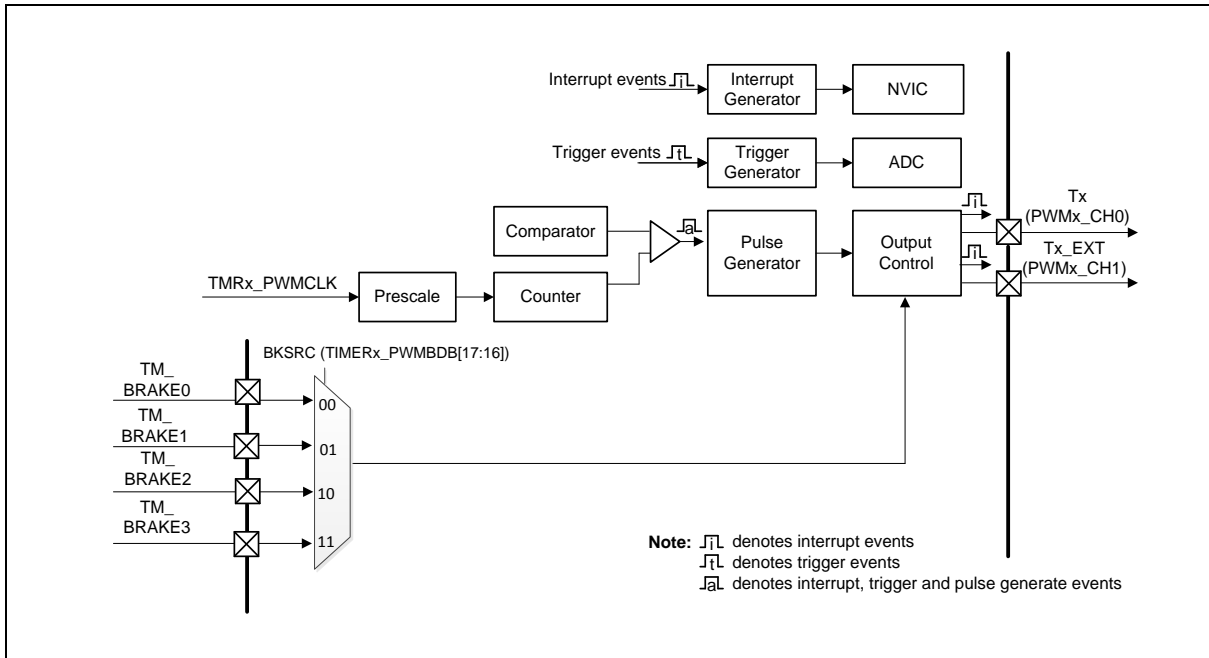


Figure 6.17-6 PWM Independent Mode Architecture Diagram

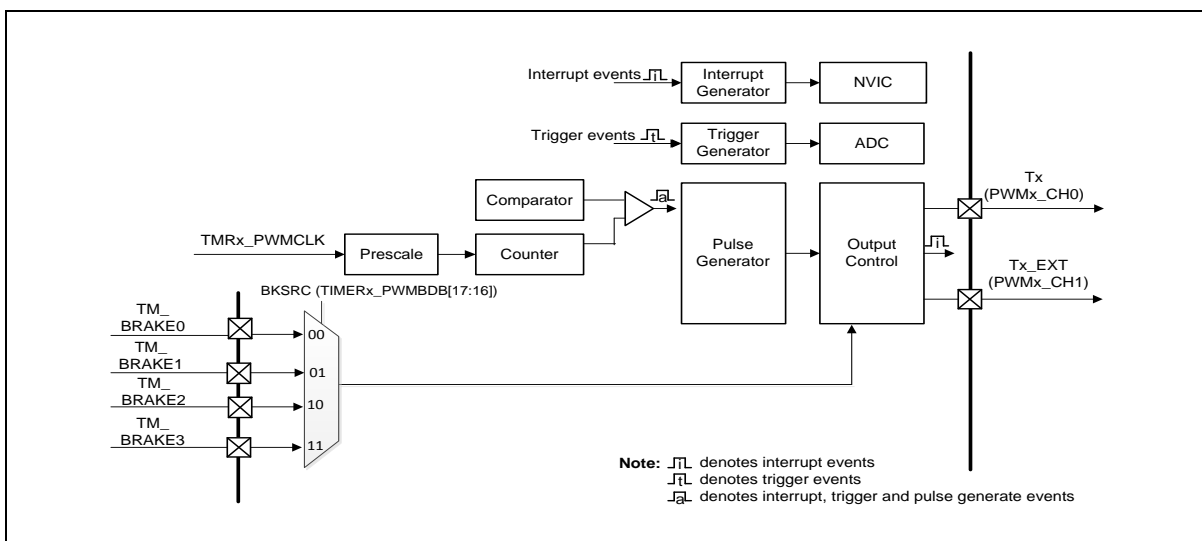


Figure 6.17-7 PWM Complementary Mode Architecture Diagram

### 6.17.4 Basic Configuration

Set FUNMODE (TIMERx\_ALTCTL[0]) 0 to enable timer mode. 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.

Set FUNMODE (TIMERx\_ALTCTL[0]) 1 to enable PWM mode. The clock source of Timer0 ~ Timer3 in PWM mode can be enabled in TMRxCKEN (CLK\_APBCLK0[5:2]). TMR0\_CLK and TMR1\_CLK clock sources are fixed to PCLK0. TMR2\_CLK and TMR3\_CLK clock sources are fixed to PCLK1.

#### 6.17.4.1 Basic Configuration of TIMER01

- Clock Source Configuration
  - Select the source of TIMER01 peripheral clock on TMR0SEL (CLK\_CLKSEL1[10:8]) for Timer0 and TMR1SEL (CLK\_CLKSEL1[14:12]) for Timer1.
  - Enable TIMER01 peripheral clock in TMR0CKEN (CLK\_APBCLK0[2]) and TMR1CKEN (CLK\_APBCLK0[3]).
- Reset Configuration
  - Reset TIMER01 controller in TMR0RST (SYS\_IPRST1[2]) and TMR1RST (SYS\_IPRST1[3]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
TM0	TM0	PE.8	MFP3
		PD.1, PD.4	MFP6
	TM0_EXT	PA.7, PD.2	MFP3
		PE.10	MFP8
		PB.3	MFP10
TM1	TM1	PE.9	MFP3

		PD.7	MFP4
		PD.5	MFP6
		PA.8	MFP8
	TM1_EXT	PA.6, PD.3	MFP3
		PE.11	MFP8
		PB.0, PB.4	MFP10

6.17.4.2 Basic Configuration of TIMER23

- Clock Source Configuration
  - Select the source of TIMER23 peripheral clock on TMR2SEL (CLK\_CLKSEL1[18:16]) for Timer2 and TMR3SEL (CLK\_CLKSEL1[22:20]) for Timer3.
  - Enable TIMER23 peripheral clock in TMR2CKEN (CLK\_APBCLK0[4]) and TMR3CKEN (CLK\_APBCLK0[5]).
- Reset Configuration
  - Reset TIMER23 controller in TMR2RST (SYS\_IPRST1[4]) and TMR3RST (SYS\_IPRST1[5]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
TM2	TM2	PD.3	MFP1
		PB.0, PD.10	MFP4
		PA.14, PD.8	MFP6
		PA.9	MFP8
	TM2_EXT	PA.5	MFP3
		PE.0	MFP4
		PE.12	MFP8
TM3	TM3	PB.1, PD.11	MFP4
		PA.15, PD.9	MFP6
	TM3_EXT	PA.4, PE.1, PF.5	MFP3
		PE.13	MFP8

6.17.5 Timer Functional Description

6.17.5.1 Timer Interrupt Flag

The timer controller supports two interrupt flags; one is TIF (TIMERx\_INTSTS[0]) and its set while timer counter value CNT (TIMERx\_CNT[23:0]) matches the timer compared value CMPDAT (TIMERx\_CMP[23:0]), the other is CAPIF (TIMERx\_EINTSTS[0]) and its set when the transition on the Tx\_EXT pin associated CAPEDGE (TIMERx\_EXTCTL[14:12]) setting.

#### 6.17.5.2 Timer Counting Mode

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

#### 6.17.5.3 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.

#### 6.17.5.4 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, timer controller operates counting and compares with CMPDAT value periodically until the CNTEN bit is cleared by user.

#### 6.17.5.5 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 T0 ~ T3 pin to output signal while specify TIF (TIMERx\_INTSTS[0]) is set. Thus, the toggle-output signal on T0 ~ T3 pin is high and changing back and forth with 50% duty cycle.

#### 6.17.5.6 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.

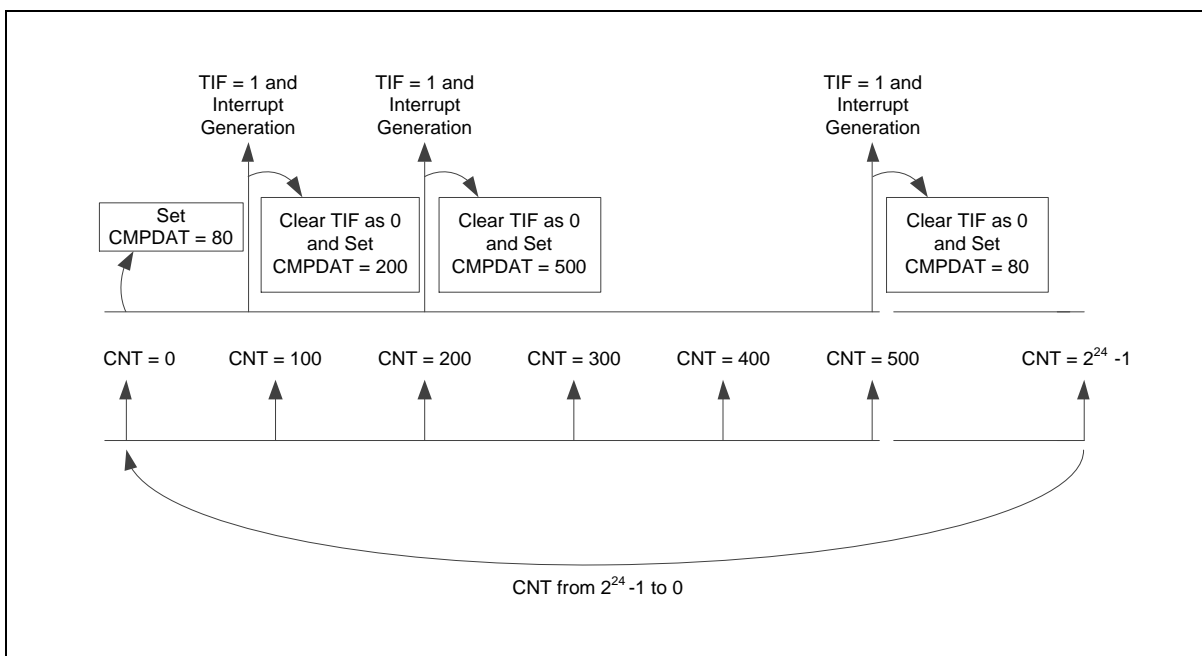


Figure 6.17-8 Continuous Counting Mode

#### 6.17.5.7 Event Counting Mode

Timer controller also provides an application which can count the input event from Tx (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.

User can enable or disable Tx pin de-bounce circuit by setting CNTDBEN (TIMERx\_EXTCTL[7]). The input event frequency should be less than 1/3 PCLK if Tx pin de-bounce disabled or less than 1/8 PCLK if Tx pin de-bounce enabled to assure the returned CNT value is correct, and user can also select edge detection phase of Tx 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 Tx pin.

#### 6.17.5.8 External 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 Tx\_EXT (x= 0~3) pin. In this mode, CAPFUNCS (TIMERx\_EXTCTL[4]) should be as 0 for select Tx\_EXT transition is using to trigger event capture function and the timer peripheral clock source should be set as PCLK.

User can enable or disable Tx\_EXT pin de-bounce circuit by setting CAPDBEN (TIMERx\_EXTCTL[6]). The transition frequency of Tx\_EXT pin should be less than 1/3 PCLK if Tx\_EXT pin de-bounce disabled or less than 1/8 PCLK if Tx\_EXT pin de-bounce enabled to assure the capture function can be work normally, and user can also select edge transition detection of Tx\_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 Tx\_EXT pin is detected.

Users 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.

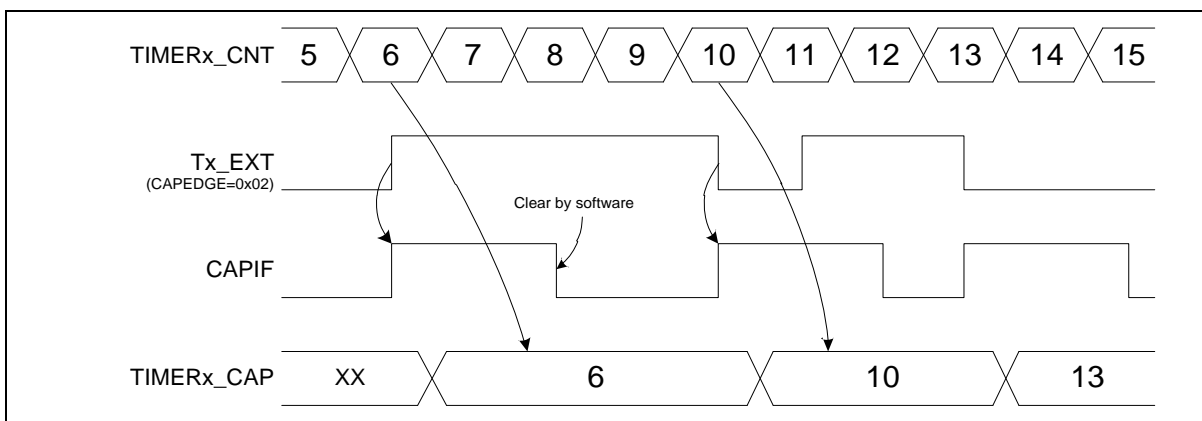


Figure 6.17-9 External Capture Mode

6.17.5.9 External Reset Counter Mode

The timer controller also provides reset counter function to reset CNT (TIMERx\_CNT[23:0]) value while edge transition detected on Tx\_EXT (x= 0~3). In this mode, most the settings are the same as event capture mode except CAPFUNCS (TIMERx\_EXTCTL[4]) should be as 1 for select Tx\_EXT transition is using to trigger reset counter value.

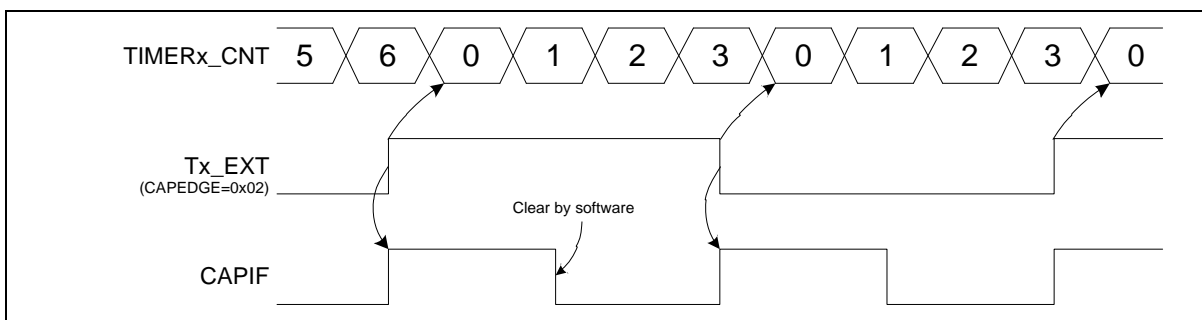


Figure 6.17-10 External Reset Counter Mode

6.17.5.10 Timer Trigger Function

The timer controller provides timer time-out interrupt or capture interrupt internal trigger event to generate PWM counter counting once, start ADC convert and trigger PDMA transfer. If TRGSSEL (TIMERx\_TRGCTL[0]) is 0, time-out interrupt signal is used to trigger PWM, ADC and PDMA. If TRGSSEL (TIMERx\_TRGCTL[0]) is 1, capture interrupt signal is used to trigger PWM, ADC and PDMA.

When the TRGPWM (TIMERx\_TRGCTL[1]) is set, if the timer interrupt signal is generated, the timer controller will generate a trigger one pulse as PWM counter clock source.

When the TRGADC (TIMERx\_TRGCTL[2]) is set, if the timer interrupt signal is generated, the timer controller will trigger ADC to start convert.

When the TRGPDMA (TIMERx\_TRGCTL[4]) is set, if the timer interrupt signal is generated, the timer controller will trigger PDMA to start transfer.



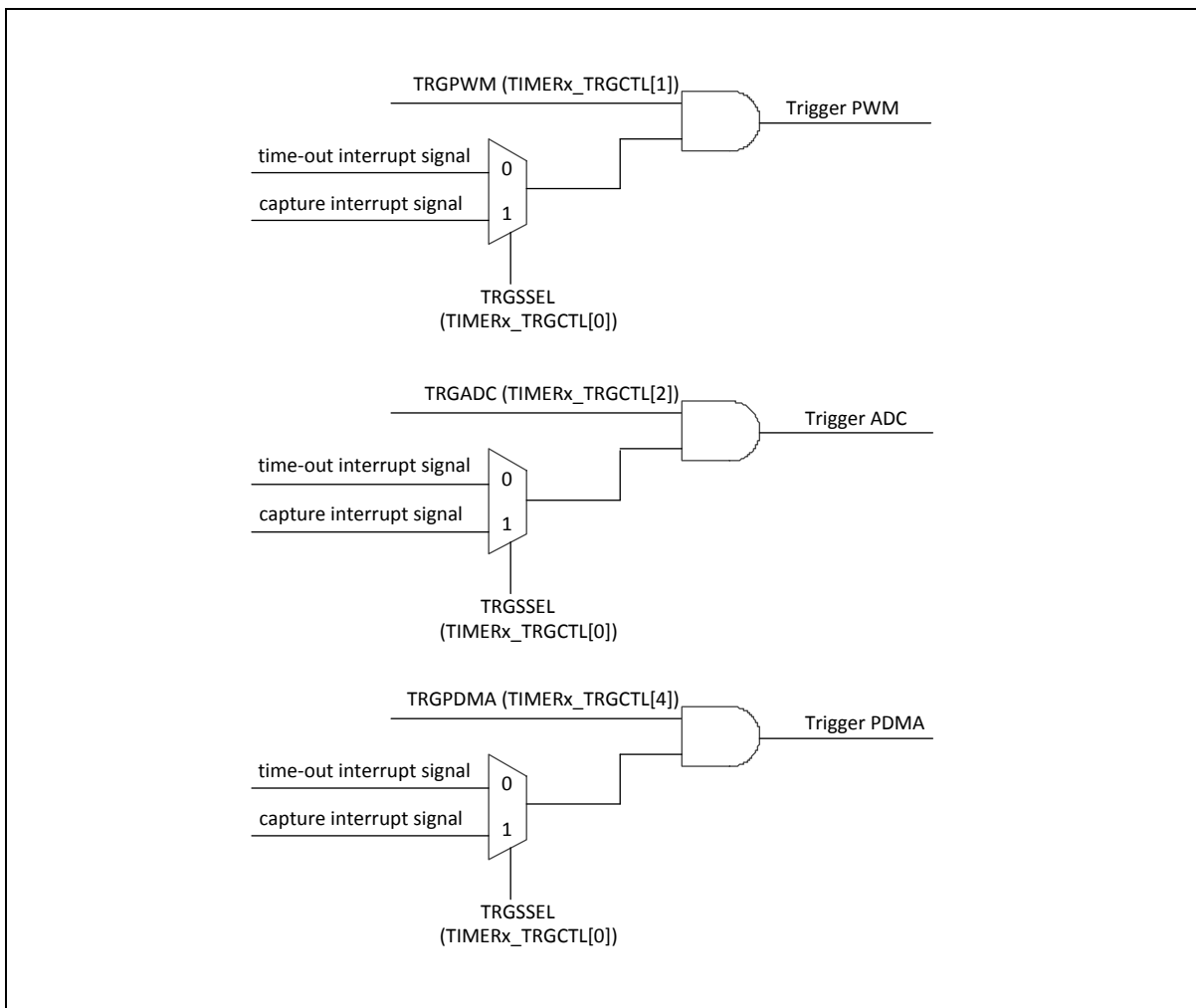


Figure 6.17-11 Internal Timer Trigger

6.17.5.11 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 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 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 (Tx) more precisely. Figure 6.17-12 shows the sample flow of Inter-Timer Trigger Capture Mode for Timer0 as event

counting mode and Timer1 as trigger-counting capture mode.

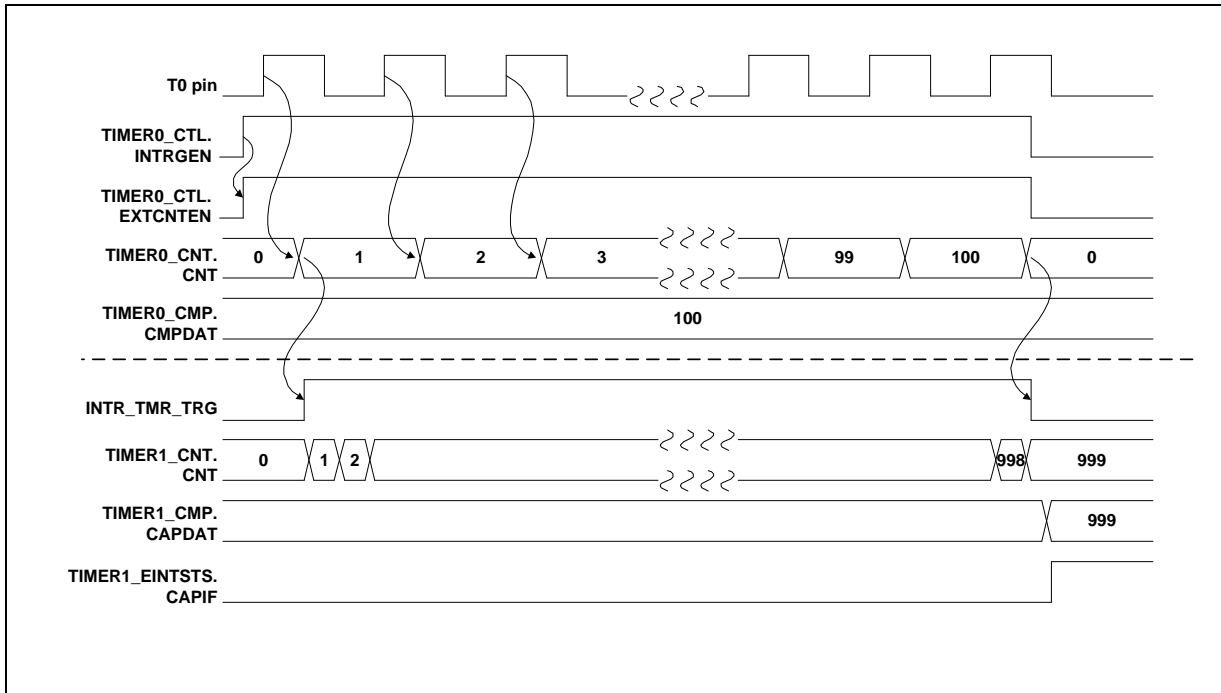


Figure 6.17-12 Inter-Timer Trigger Capture Timing

6.17.5.12 Internal Capture Trigger from ACMP

The external capture function can also be triggered by internal output signal transition on ACMP0, or ACMP1 output. User can set ACMPSEL (TIMERx\_EXTCTL[9:8]) to decide which ACMP output signal as Timerx (x= 0~3) capture source. The detail setting of capture function is the same as previous descriptions in External Capture Mode.

6.17.6 PWM Functional Description

6.17.6.1 PWM Prescale

The PWM prescale is used to divide clock source, and the clock of PWM counter is divided by (CLKPSC+ 1). The prescale is set by CLKPSC (TIMERx\_PWMCLKPSC[11:0]). Figure 6.17-13 shows an example of PWM prescale waveform in up count type.

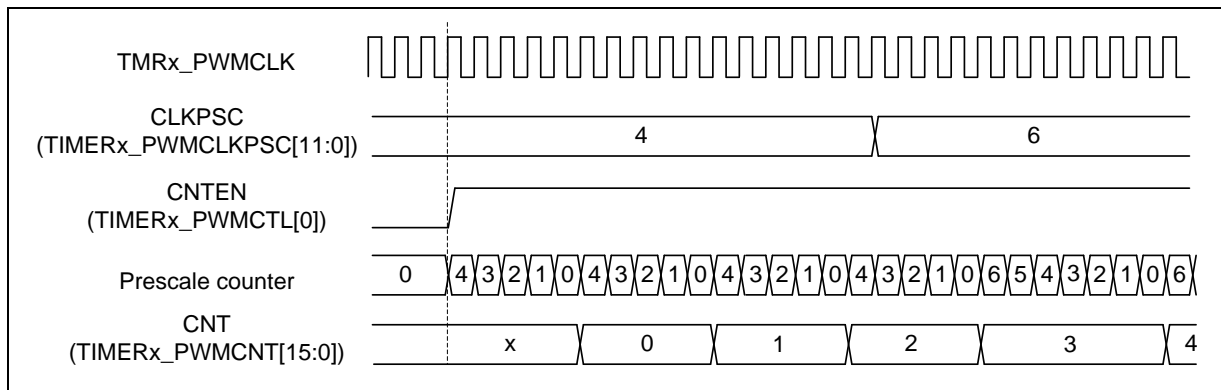


Figure 6.17-13 PWM Prescale Waveform in Up Count Type

6.17.6.2 PWM Counter

PWM supports three counter types operation: up count, down count and up-down count types.

6.17.6.3 Up Count Type

When the PWM counter is set to up count type, CNTTYPE (TIMERx\_PWMCTL[2:1]) is 0x0, it starts up-counting from zero to PERIOD (TIMERx\_PWMPERIOD[15:0]). The current counter value can be read from the CNT (TIMERx\_PWMCNT[15:0]). PWM generates a zero point event when both counter and prescale counts to 0. PWM generates a period point event when the counter counts to PERIOD and prescale counts to 0. Figure 6.17-14 shows an example of PWM up count type, where PWM period time is  $(PERIOD+1) * (CLKPSC+1) * TMRx\_PWMCLK$ .

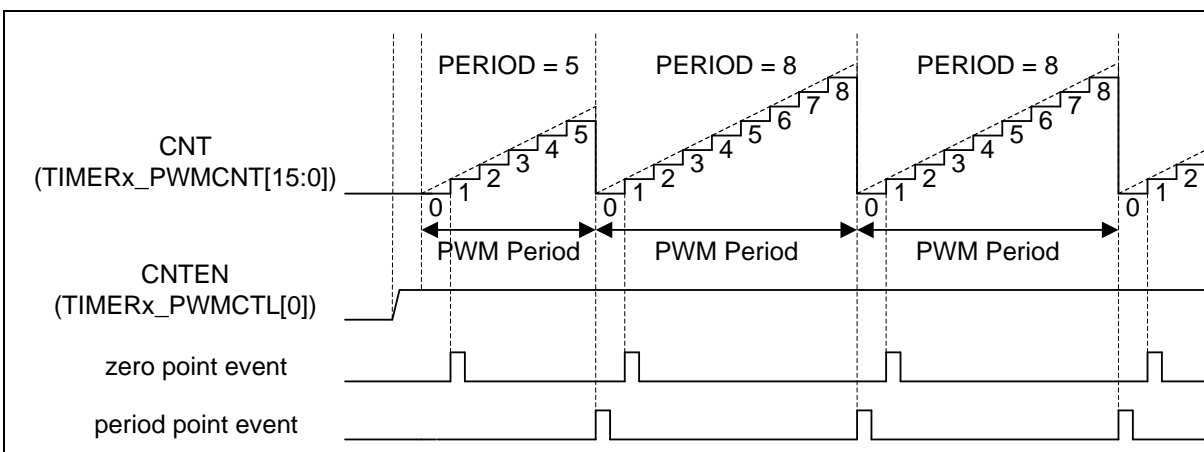


Figure 6.17-14 PWM Up Count Type

6.17.6.4 Down Count Type

When the PWM counter is set to down count type, CNTTYPE (TIMERx\_PWMCTL[2:1]) is 0x1, it starts down-counting from PERIOD to zero, current counter value can be read from CNT (TIMERx\_PWMCNT[15:0]). PWM generates a zero point event when both counter and prescale counts to 0. PWM generates a period point event when the counter counts to PERIOD and prescale counts to 0. Figure 6.17-15 is an example of PWM down count type, where PWM period time is  $(PERIOD+1) * (CLKPSC+1) * TMRx\_PWMCLK$ .

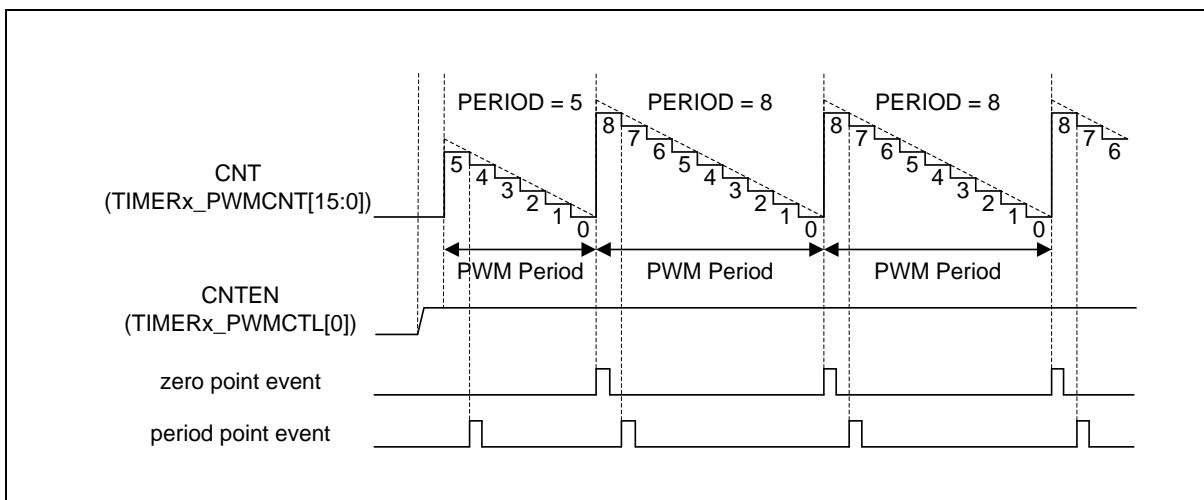


Figure 6.17-15 PWM Down Count Type

### 6.17.6.5 Up-Down Count Type

When the PWM counter is set to up-down count type, CNTTYPE (TIMERx\_PWMCTL[2:1]) is 0x2, it starts counting up from zero to PERIOD and then starts counting down to zero. The current counter value can be read from CNT (TIMERx\_PWMCNT[15:0]). PWM generates a zero point event when both counter and prescale counts to 0. PWM generates a center point event when the counter counts to PERIOD and prescale counts to 0. Figure 6.17-16 shows an example of PWM up-down count type, where PWM period time is  $(2 * PERIOD) * (CLKPSC+1) * TMRx\_PWMCLK$ . The DIRF (TIMERx\_PWMCNT[16]) is counter direction indicator flag, where 1 is up counting, and 0 is down counting.

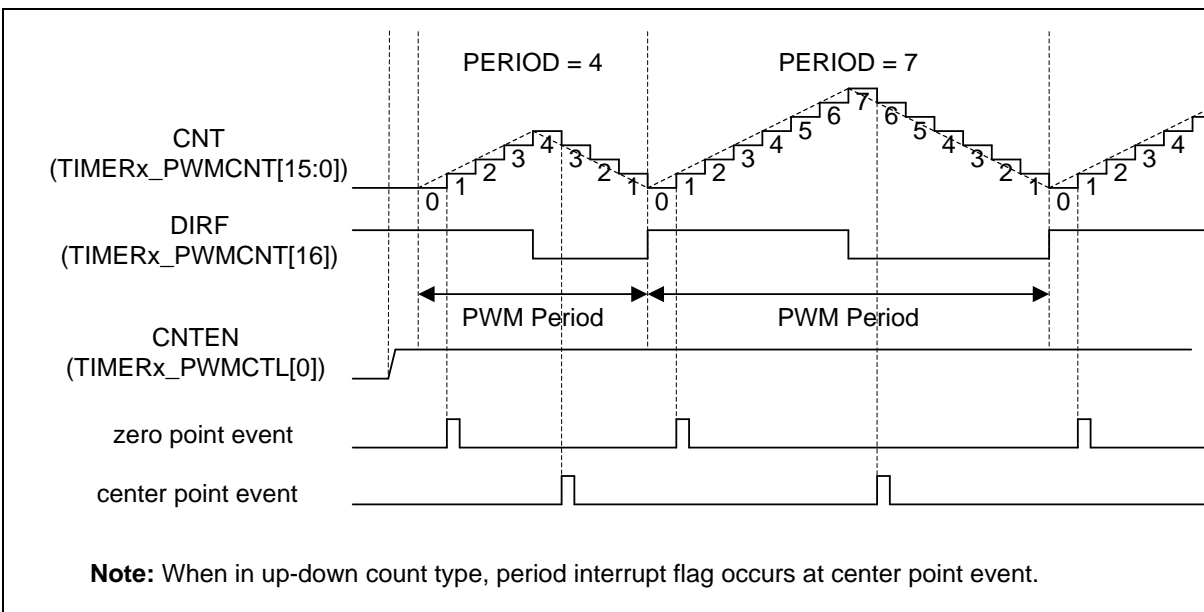


Figure 6.17-16 PWM Up-Down Count Type

### 6.17.6.6 PWM Counter Operation mode

The PWM counter supports two operation modes: one-shot mode and auto-reload mode. PWM counter will operate in one-shot mode if CNTMODE (TIMERx\_PWMCTL[3]) bit is set to 1, and operate in auto-reload mode if CNTMODE bit is set to 0.

In both modes, CMP (TIMERx\_PWMCMPDAT[15:0]) and PERIOD (TIMERx\_PWMPERIOD[15:0]) should be written first and then set CNTEN (TIMERx\_PWMCTL[0]) bit to 1 to start counter running.

In one-shot mode, PWM counter value will reload to default value according count type after one PWM period is completed. User can write CMP to continuous one-shot operation to generate next one-shot pulse once no matter current one-shot counter is running or completed.

In auto-reload mode, PWM counter is continuous running with current active PERIOD and CMP. If user set PERIOD to zero in auto-reload mode, PWM counter value will reload to default value according count type after one PWM period is completed.

### 6.17.6.7 PWM Comparator

The CMP (TIMERx\_PWMCMPDAT[15:0]) is comparator register of PWM. The CMP value is continuously compared to the corresponding counter value. When the counter is equal to CMP, PWM generates a compared point event. This event will generate PWM output pulse, interrupt signal or trigger ADC start convert. In up-down count type, two events will be generated in a PWM period as shown in Figure 6.17-17. The CMPU is up count compared point event and CMPD is down count compared point event.

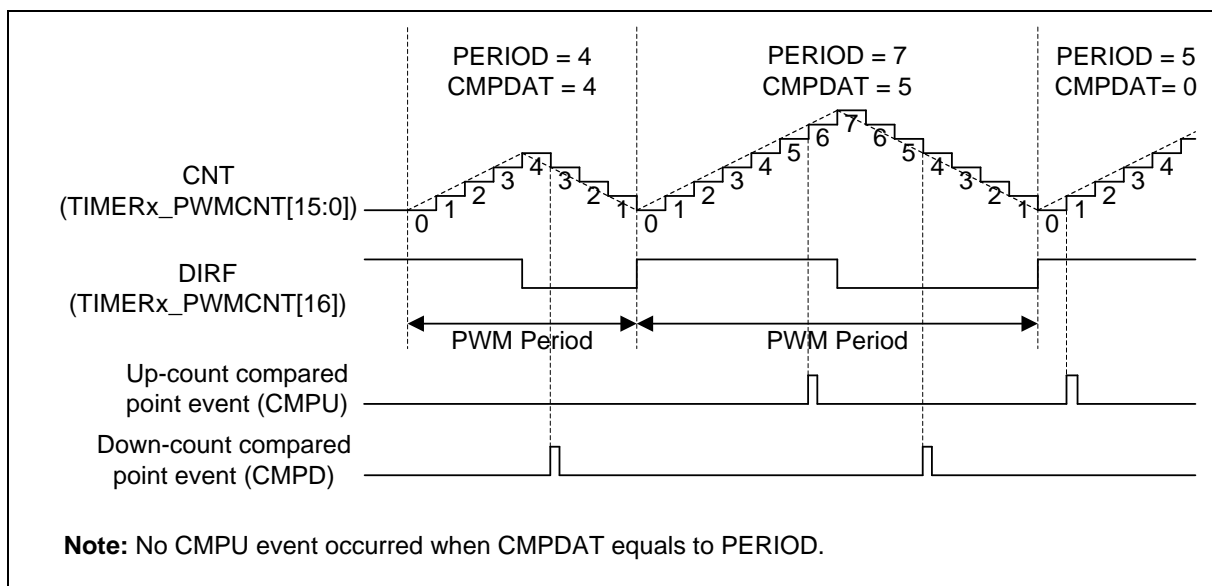


Figure 6.17-17 PWM Comparator Events in Up-Down Count Type

#### 6.17.6.8 Period Loading Mode

When the IMMLDEN (TIMERx\_PWMCTL[9]) bit set to 0, PWM operates at period loading mode. The PWM provides PBUF (TIMERx\_PWMPBUF[15:0]) is the active PERIOD buffer register and CMPBUF (TIMERx\_PWMCMPBUF[15:0]) is the active CMP buffer register. In period loading mode, both PERIOD (TIMERx\_PWMPERIOD[15:0]) and CMP (TIMERx\_PWMCMPDAT[15:0]) will load to their active PBUF and CMPBU register while each PWM period is completed. Figure 6.17-18 shows period loading timing of up count type, where PERIOD DATA0 denotes the initial data of PERIOD, PERIOD DATA1 denotes the first updated PERIOD data by user and so on, CMP also follows this rule. The following steps are the sequence of Figure 6.17-18.

1. User writes CMP DATA1 to CMPDAT at point 1.
2. Period loading CMP DATA1 to CMPBUF at the end of PWM period at point 2.
3. User writes PERIOD DATA1 to PERIOD at point 3.
4. Period loading PERIOD DATA1 to PBUF at the end of PWM period at point 4.
5. User writes PERIOD DATA2 to PERIOD at point 5.
6. Period loading PERIOD DATA2 to PBUF at the end of PWM period at point 6.

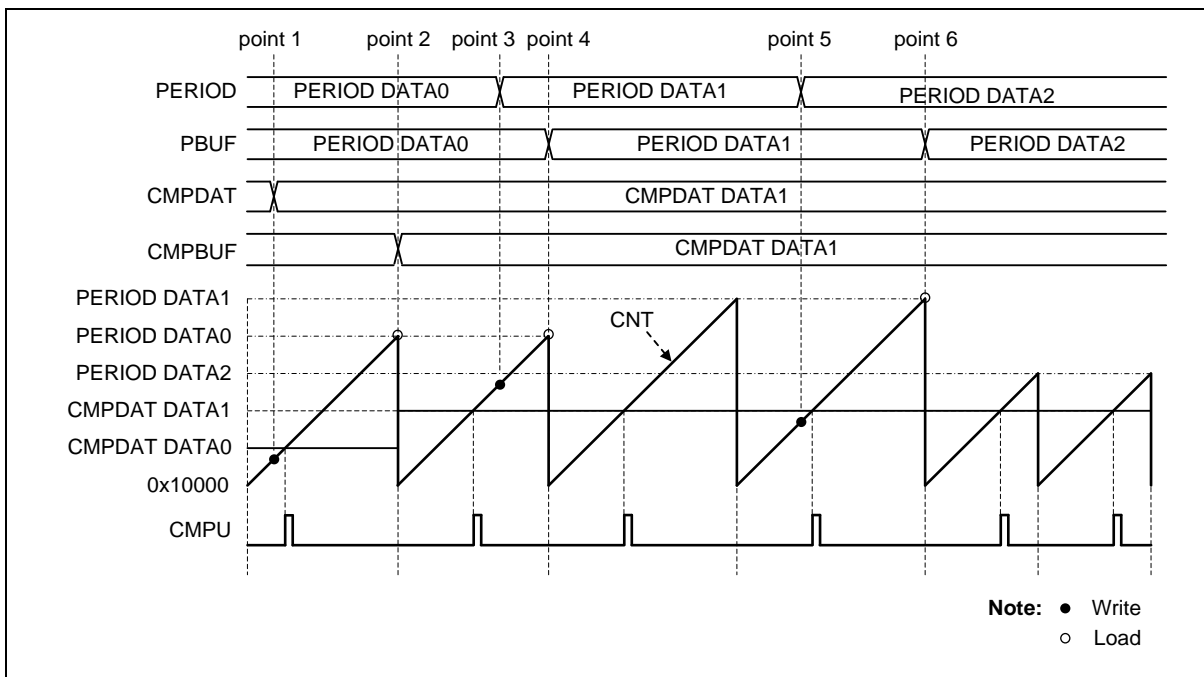


Figure 6.17-18 Period Loading Mode with Up Count Type

#### 6.17.6.9 Immediately Loading Mode

When the IMMLDEN (TIMERx\_PWMCTL[9]) bit set to 1, PWM operates at immediately loading mode. In immediately loading mode, when user update PERIOD (TIMERx\_PWMPERIOD[15:0]) or CMP (TIMERx\_PWMCMPDAT[15:0]), PERIOD or CMP will be load to active PBUF (TIMERx\_PWMPBUF[15:0]) or CMPBUF (TIMERx\_PWMCMPBUF[15:0]) after current counter count is completed. If the update PERIOD value is less than current counter value, counter will count wraparound. The following steps are the sequence of Figure 6.17-19.

1. User writes CMP DATA1 at point 1 and hardware will load CMP DATA1 to CMPBUF after current counter count is completed.
2. User writes PERIOD DATA1 at point 2 and PERIOD DATA1 is greater than current counter value, PWM counter will continuously count until equal to PERIOD DATA1 to complete one PWM period.
3. User writes PERIOD DATA2 at point 3 and PERIOD DATA2 is less than the current counter value, PWM counter will continuously count to maximum counter value 0x1FFFF and wraparound from 0x10000 to PERIOD DATA2 to complete one PWM period.

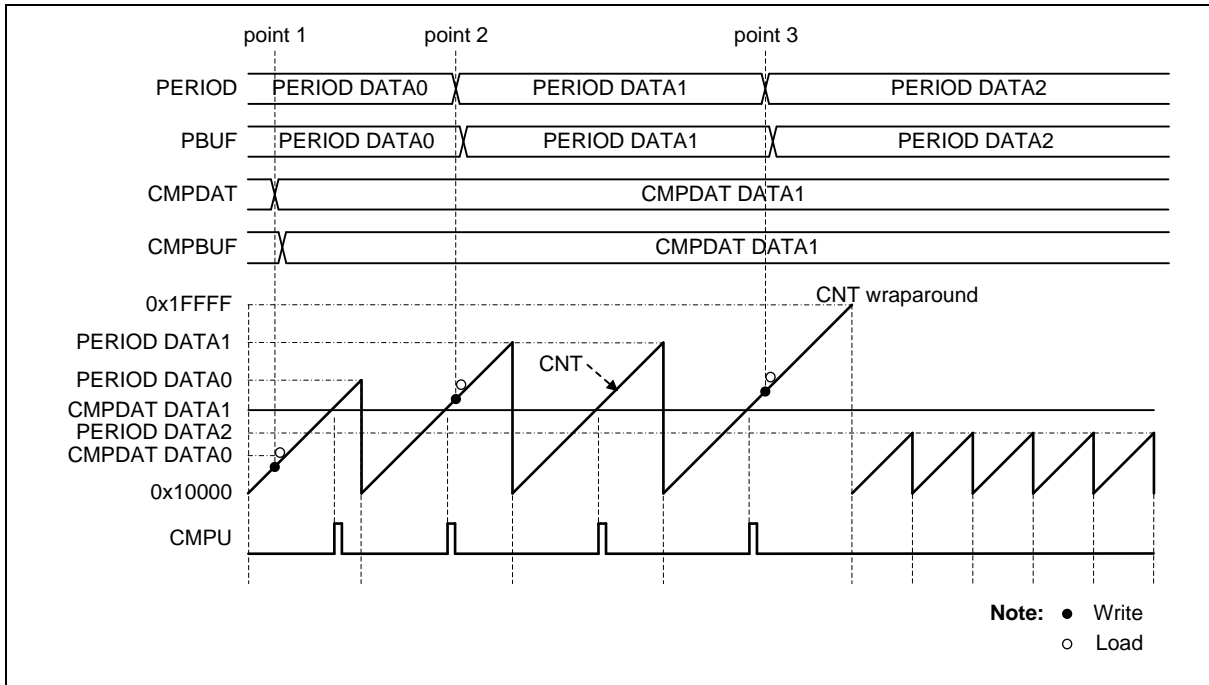


Figure 6.17-19 Immediately Loading Mode with Up Count Type

6.17.6.10 PWM Pulse Generator

PWM pulse generator uses counter and comparator events to generate PWM output pulse. The events are zero point and period point in up count type and down count type, center point in up-down count type and counter equal to comparator point in three count types.

Each event point can generate PWM output waveform in different count type as shown in Figure 6.17-20.

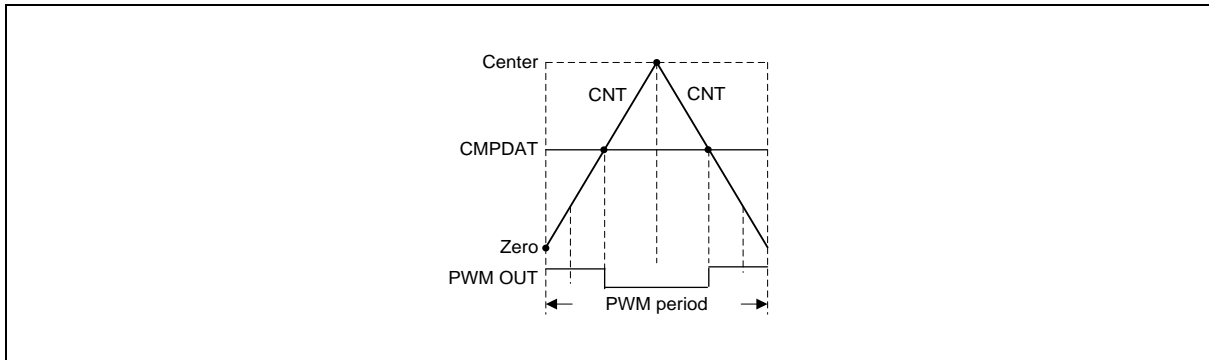


Figure 6.17-20 PWM Pulse Generation in Up-Down Count Type

The PWM generation events may sometimes generated at the same time, as the reason, events priority between different counter types should be take care are list in below tables. Table 6.17-1 is for event priority in up count type. Table 6.17-2 is for event priority in down count type and Table 6.17-3 is for event priority in up-down count type.

Priority	Zero And CMPU Point Event (CMP = 0)	PWM Output
1 (High)	Compare up event	Low
2 (Low)	Zero event	High

Table 6.17-1 PWM Pulse Generation Event Priority in Up Count Type

Priority	Zero And CMPD Point Event (CMP = 0)	PWM Output
1 (High)	Zero event	Low
2 (Low)	Compare down event	High
Priority	Period and CMPD point event (CMP = PERIOD)	PWM output
1 (High)	Compare down event	High
2 (Low)	Period event	Low

Table 6.17-2 PWM Pulse Generation Event Priority in Down Count Type

Priority	CMPU And CMPU Point Event (CMP = PERIOD)	PWM Output
1 (High)	Compare down event	High
2 (Low)	Compare up event	Low

Table 6.17-3 PWM Pulse Generation Event Priority in Up-Down Count Type

According to event priority limitation, PWM generator can support 0% and 100% duty cycle PWM output waveform only in up count and up-down count type. Figure 6.17-21 is an example about PWM duty cycle from 0% to 100% in up count type and up-down count type where PERIOD is 4 with different CMP value.

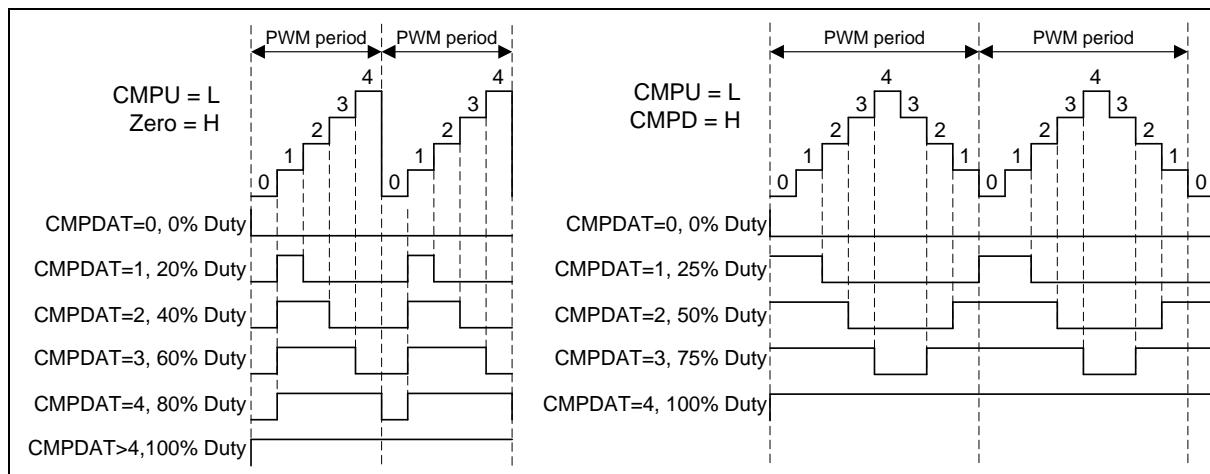


Figure 6.17-21 PWM 0% to 100% Duty Cycle in Up Count Type and Up-Down Count Type



6.17.6.11 PWM Output Mode

The PWM supports two output modes: independent mode which may be applied to DC motor system, complementary mode with dead-time insertion which may be used in the application of AC induction motor and permanent magnet synchronous motor.

6.17.6.12 Independent mode

When OUTMODE (TIMERx\_PWMCTL[16]) bit is set to 0, PWM output operates in independent mode. In this mode, both PWMx\_CH0 and PWMx\_CH1 can output the same waveform as shown in Figure 6.17-22.

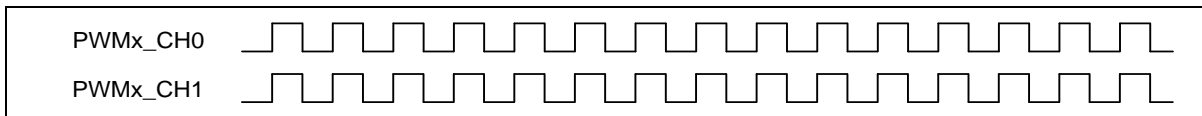


Figure 6.17-22 PWM Independent Mode Output Waveform

6.17.6.13 Complementary mode

When OUTMODE (TIMERx\_PWMCTL[16]) bit is set to 1, PWM output operates in complementary mode. In this mode, both PWMx\_CH0 and PWMx\_CH1 can output waveform and PWMx\_CH1 must always be the complement of PWMx\_CH0 as shown in Figure 6.17-23.

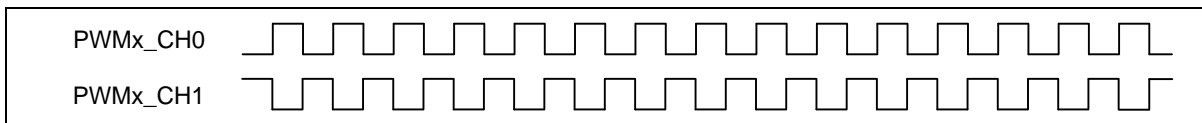


Figure 6.17-23 PWM Complementary Mode Output Waveform

6.17.6.14 PWM Output Control

After PWM pulse generator, there are four steps to control output waveform in independent output mode and five control steps in complementary output mode. User can set POEN0 (TIMERx\_PWMPOEN[0]) and POEN1 (TIMERx\_PWMPOEN[1]) 1 to enable PWMx\_CH0 and PWMx\_CH1 output waveform.

In Independent mode, there are mask control, brake control, polarity control and output enable control to control output waveform as shown in Figure 6.17-24.

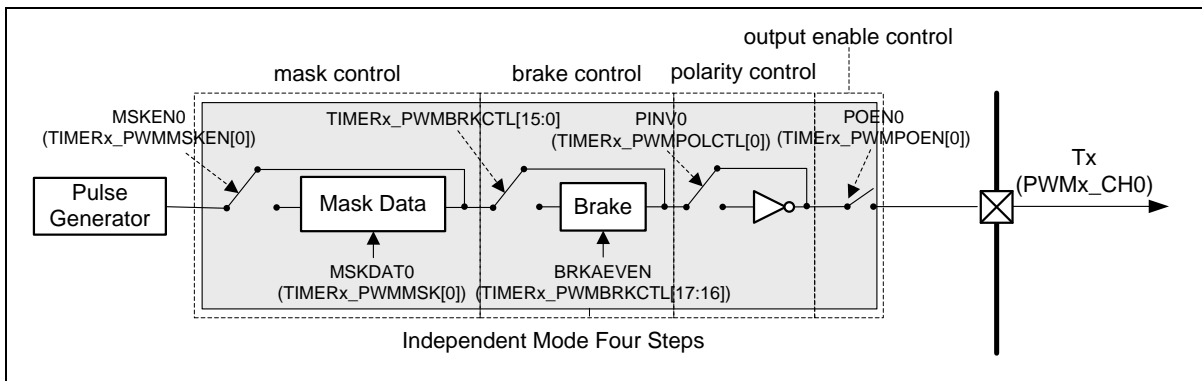


Figure 6.17-24 PWMx\_CH0 Output Control in Independent Mode

In complementary mode, there are dead-time insertion control and four control steps the same as independent mode to control PWMx\_CH0 and PWMx\_CH1 outputs as shown in Figure 6.17-25.

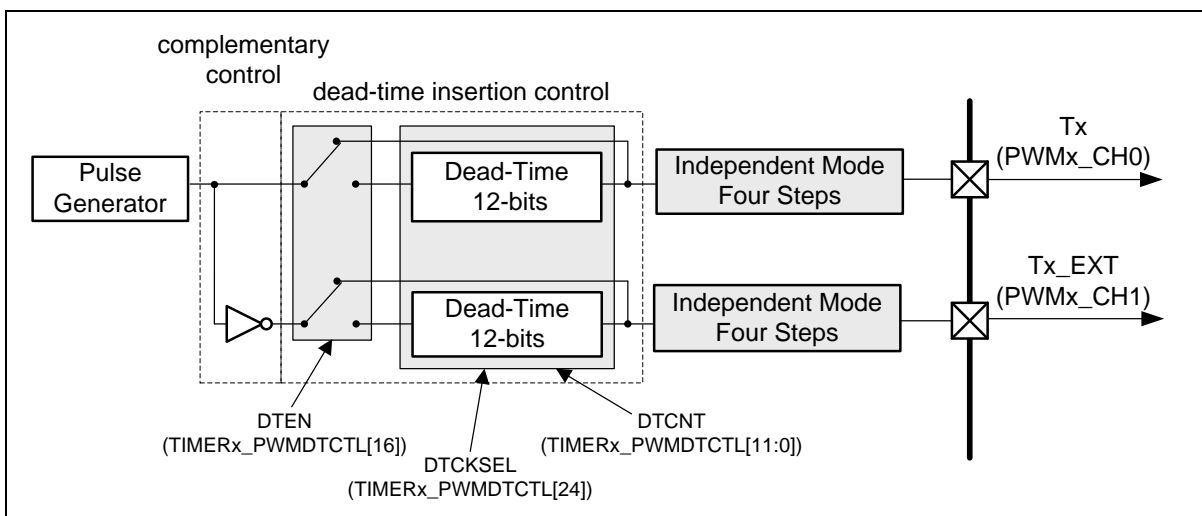


Figure 6.17-25 PWMx\_CH0 and PWMx\_CH1 Output Control in Complementary Mode

6.17.6.15 Dead-Time Insertion Control

In the complementary application, the complement channels may drive the external devices like power switches. The dead-time generator inserts a low level interval between complementary outputs PWMx\_CH0 and PWMx\_CH1 as shown in Figure 6.17-26. User sets DTEN (TIMERx\_PWMDTCTL[16]) bit to enable dead-time control function, DTCNT (TIMERx\_PWMDTCTL[11:0]) and DTCKSEL (TIMERx\_PWMDTCTL[24]) to control dead-time interval. The dead-time interval can be calculated from the following formula:

Dead-time interval = (DTCNT + 1) \* TMRx\_PWMCLK period, if DTCKSEL is 0

Dead-time interval = (DTCNT + 1) \* TMRx\_PWMCLK \* (CLKPSC + 1) period, if DTCKSEL is 1

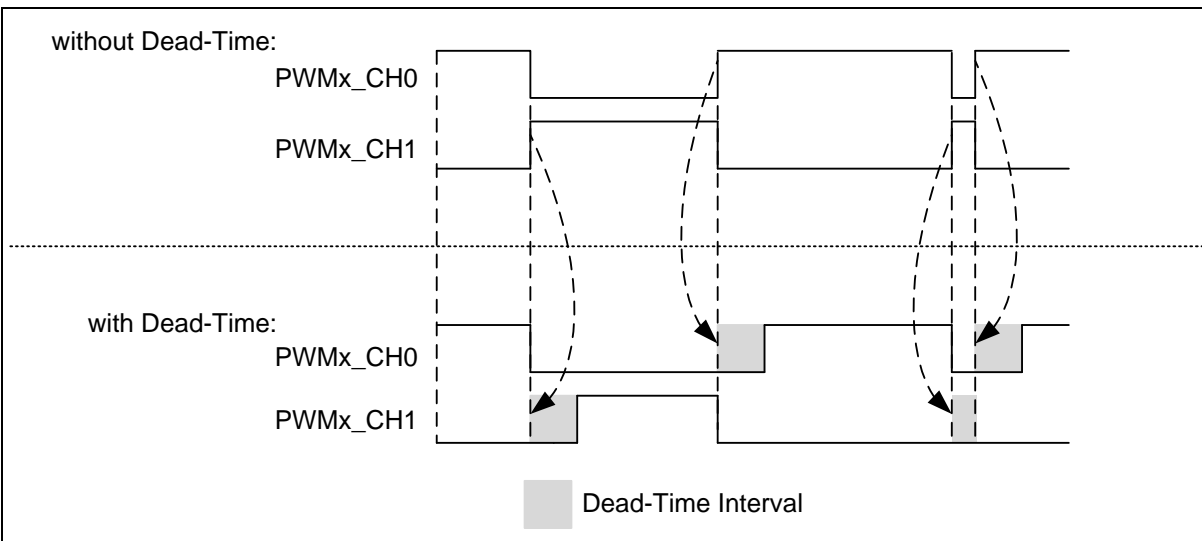


Figure 6.17-26 Dead-Time Insertion

6.17.6.16 PWM Mask Output Control

PWMx\_CH0/CH1 output value can be masked to specified logic states by setting MSKEN0/1 (TIMERx\_PWMMSKEN[1:0]) and MSKDAT0/1 (TIMERx\_PWMMSK[1:0]). The PWM output mask function is useful when controlling various types of Electrically Commutated Motor (ECM) like a BLDC motor. Figure 6.17-27 shows an example of PWM output mask control in PWMx\_CH0 and

PWMx\_CH1.

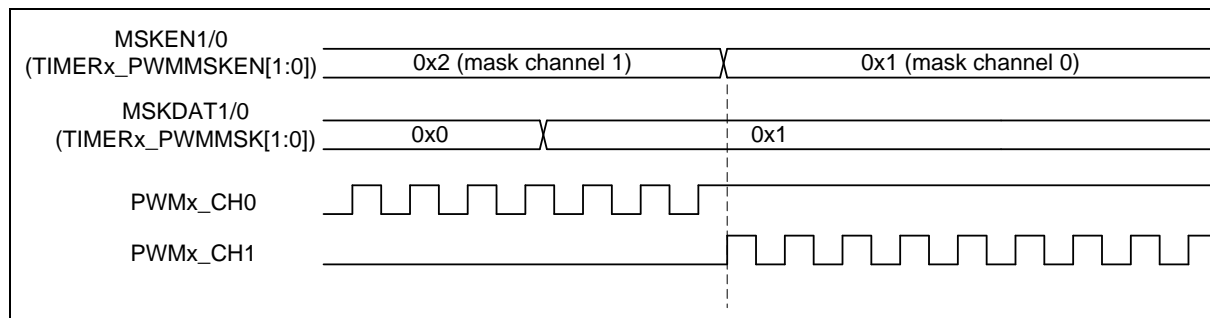


Figure 6.17-27 PWM Output Mask Control Waveform

### 6.17.6.17 PWM Brake Control

Each PWM generator supports one external input brake pin as PWM brake event source. User can select active brake pin source in BKPINSRC (TIMERx\_PWMBNF[17:16]), TM\_BRAKE<sub>x</sub> (x=0~3). There is a 3-bit noise filter counter to filter the external brake pin signal. User can enable BRKNFEN (TIMERx\_PWMBNF[0]) to enable the brake pin noise filter function and the noise filter sampling clock can be selected by setting BRKNFSEL (TIMERx\_PWMBNF[3:1]) to fit different noise properties. Moreover, by setting BRKFCNT (TIMERx\_PWMBNF[6:4]), user can define by how many sampling clock cycles a filter will recognize the effective edge of the brake pin signal. In addition, brake pin polar can be inverted by setting BRKPINV (TIMERx\_PWMBNF[7]) to realize the polarity setup for the brake control signals. Set BRKPINV to 0, brake event will occurred when TM\_BRAKE<sub>x</sub> (x=0~3) pin status from low to high; set BRKPINV to 1, brake event will occurred when TM\_BRAKE<sub>x</sub> (x=0~3) pin status from high to low.

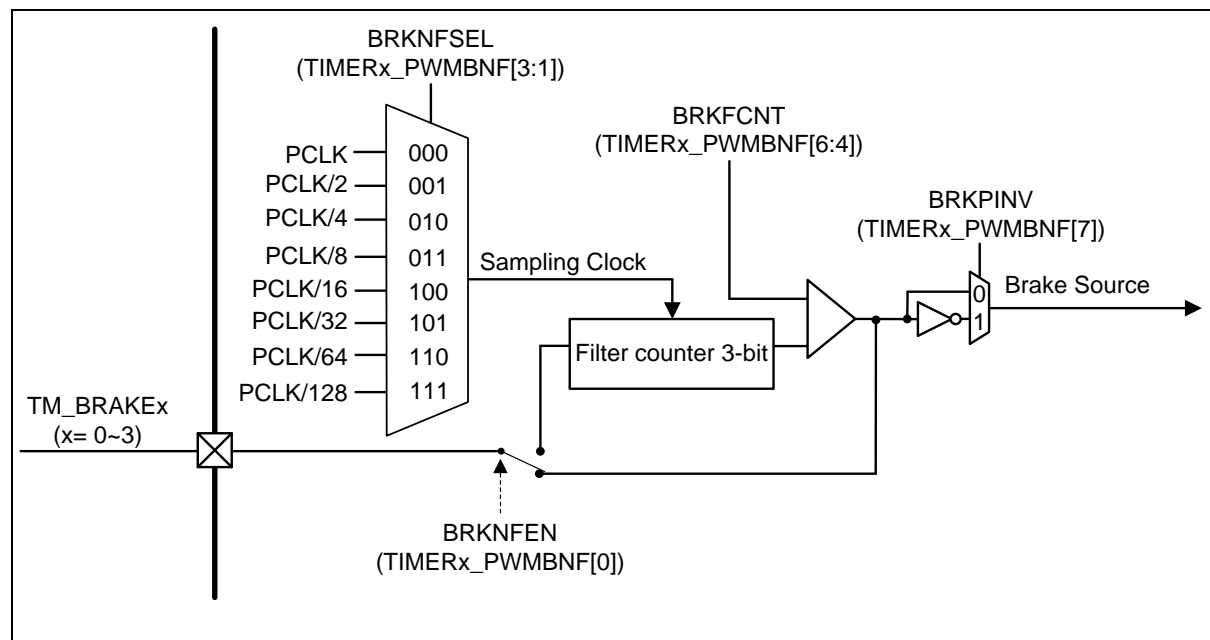


Figure 6.17-28 Brake Pin Noise Filter Block Diagram

User can set BRKAEVEN (TIMERx\_PWMBRKCTL[17:16]) for PWMx\_CH0 output state and BRKAODD (TIMERx\_PWMBRKCTL[19:18]) for PWMx\_CH1 output state when PWM brake event happened. There are two brake detector sources, edge detect brake source and level detect brake source when brake event happened. Figure 6.17-29 shows the brake event block diagram for PWMx\_CH0 and PWMx\_CH1.

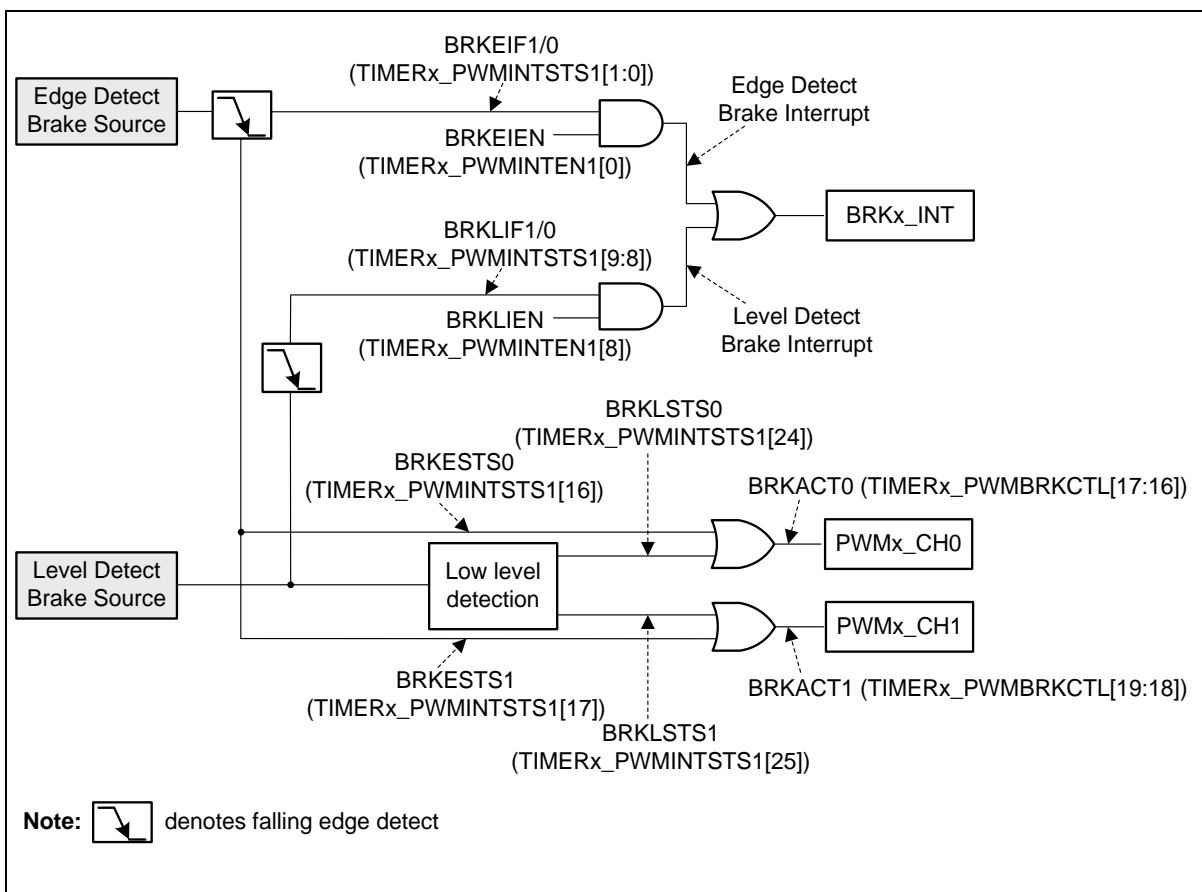


Figure 6.17-29 Brake Event Block Diagram for PWMx\_CH0 and PWMx\_CH1

When the edge detector detects the brake signal, the brake function generates interrupt status for PWMx\_CH1/0 is BRKEIF1/0 (TIMERx\_PWMINTSTS1[1:0]) and brake event status for PWMx\_CH1/0 is BRKESTS1/0 (TIMERx\_PWMINTSTS1[17:16]). The interrupt status BRKEIF1/0 can be cleared by writing 1 to it, and the brake event status BRKESTS1/0 will keep until the next PWM period starts when corresponding BRKEIF1/0 flag has been cleared and PWM generator can resume normal output.

Figure 6.17-30 shows an example of edge detector brake waveform for PWMx\_CH0 and PWMx\_CH1. In this case, the edge detect brake source has occurred twice for the brake events. When the first brake event occurs, both of the BRKEIF0 and BRKEIF1 flags are set and BRKESTS0 and BRKESTS1 status are also set to indicate brake state of PWMx\_CH0 and PWMx\_CH1. For the first occurring event, user writes 1 to clear the BRKEIF0. After that, the BRKESTS0 is cleared by hardware at the next start of the PWM period and the PWMx\_CH0 outputs the normal waveform even though the edge brake event is still occurring. At the same time, BRKESTS1 keep 1 and PWMx\_CH1 keep outputs low in brake state. The second event also triggers the same flags, but at this time, user writes 1 to clear the BRKEIF1. Afterward, PWMx\_CH1 outputs normally at the next start of the PWM period.

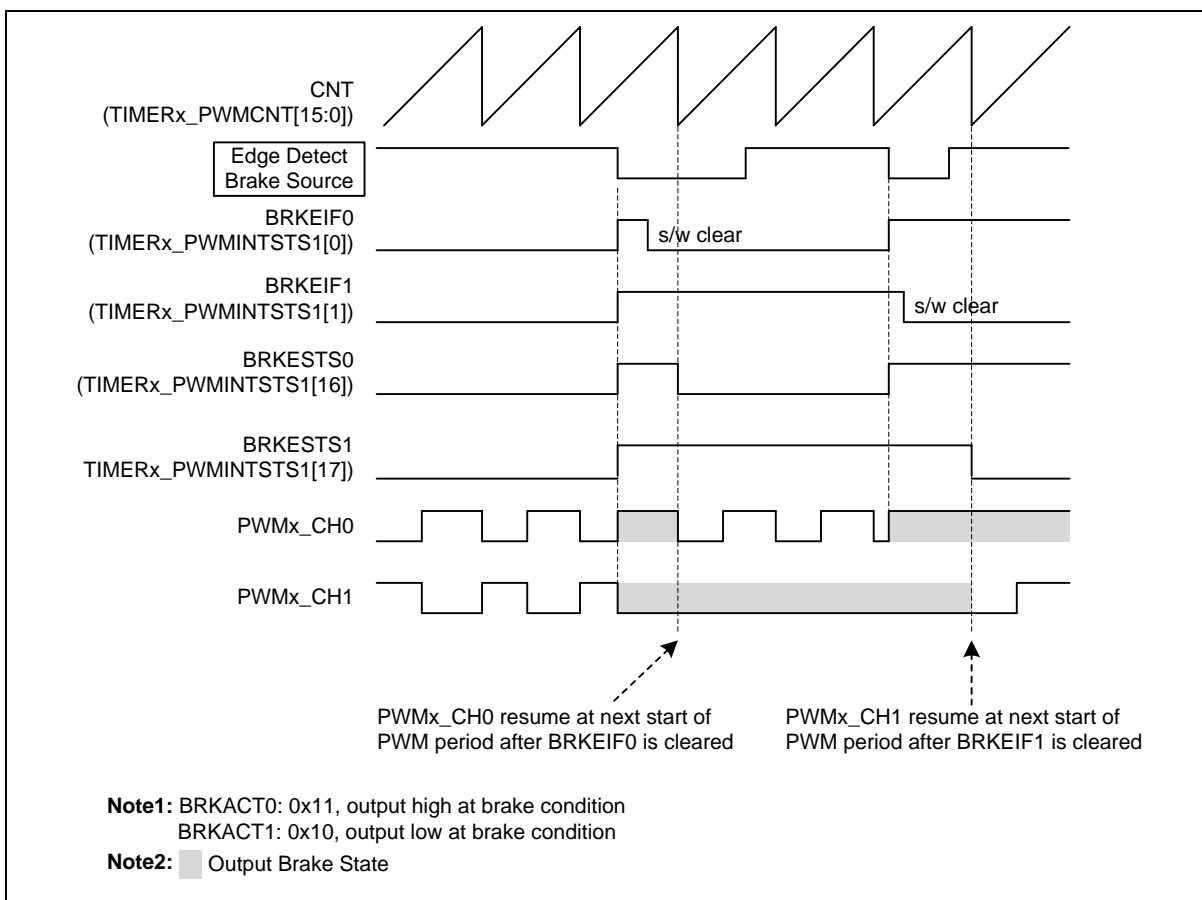


Figure 6.17-30 Edge Detector Brake Waveform for PWMx\_CH0 and PWMx\_CH1

When the level detector detects the brake signal, the brake function generates interrupt status for PWMx\_CH1/0 is BRKLIF1/0 (TIMERx\_PWMINTSTS1[9:8]) and brake event status for PWMx\_CH1/0 is BRKLSTS1/0 (TIMERx\_PWMINTSTS1[25:24]). The interrupt status BRKLIF1/0 can be cleared by writing 1 to it, and the brake event status BRKLSTS1/0 will be cleared only when current period is completed and brake condition removed, then PWM generator can resume normal output when next PWM period starts.

Figure 6.17-31 shows an example of level detector brake waveform for PWMx\_CH0 and PWMx\_CH1. In this case, the BRKLIF0 and BRKLIF1 can only indicate the brake event has occurred, writes 1 to clear this flags will not affect BRKLSTS0 and BRKLSTS1 brake event status. Both BRKLSTS0 and BRKLSTS1 brake states will automatically cleared at the start of the next PWM period when level brake condition has released no matter BRKLIF0 and BRKLIF1 status.

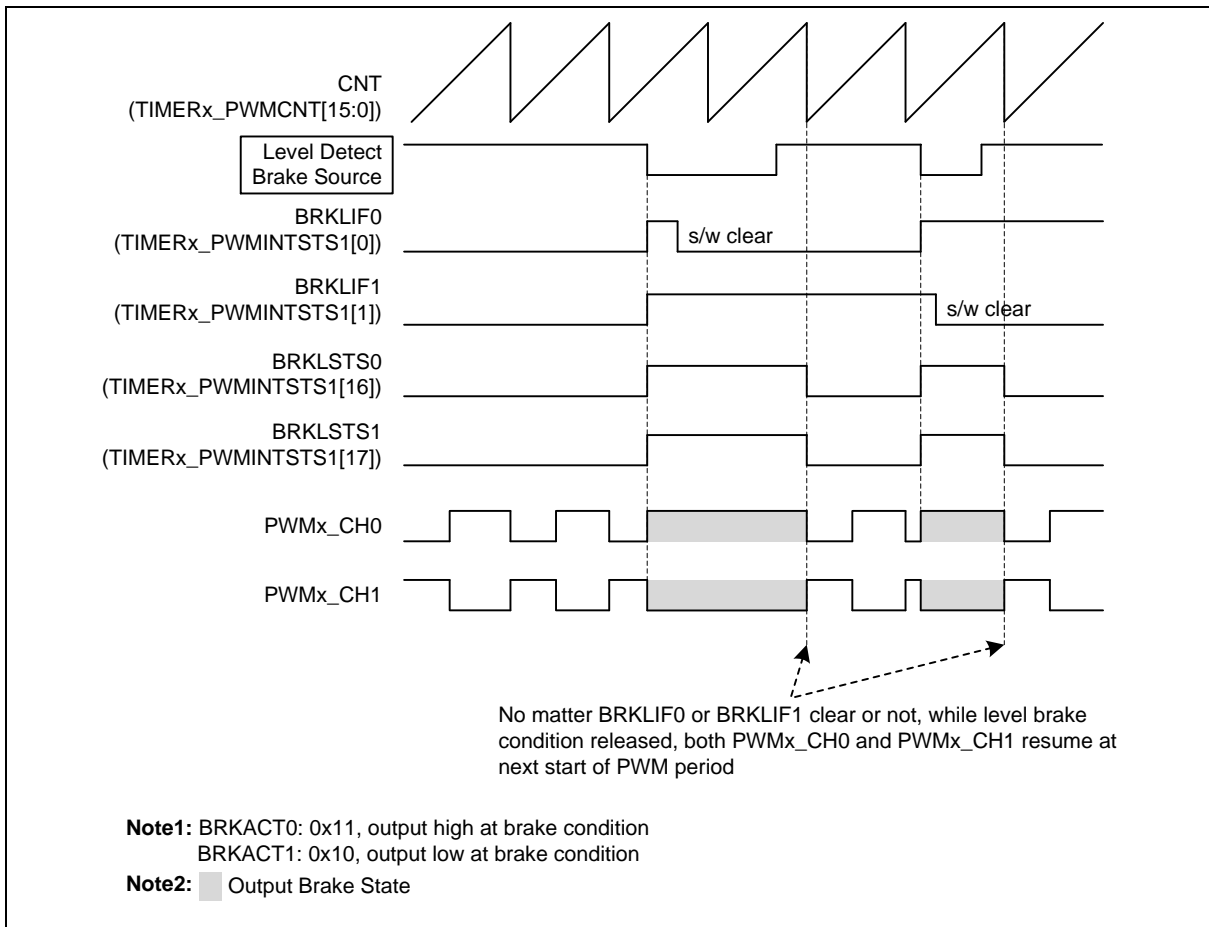


Figure 6.17-31 Level Detector Brake Waveform for PWMx\_CH0 and PWMx\_CH1

The two kinds of detectors detect the same five brake sources as shown in Figure 6.17-32: one from TM\_BRAKE<sub>x</sub> (x=0~3) external input signals, two from internal ACMP comparator signals, one from system fail events and one from software trigger brake event. ACMP brake sources will be detected only when internal ACMP0\_O or ACMP1\_O signal from low to high.

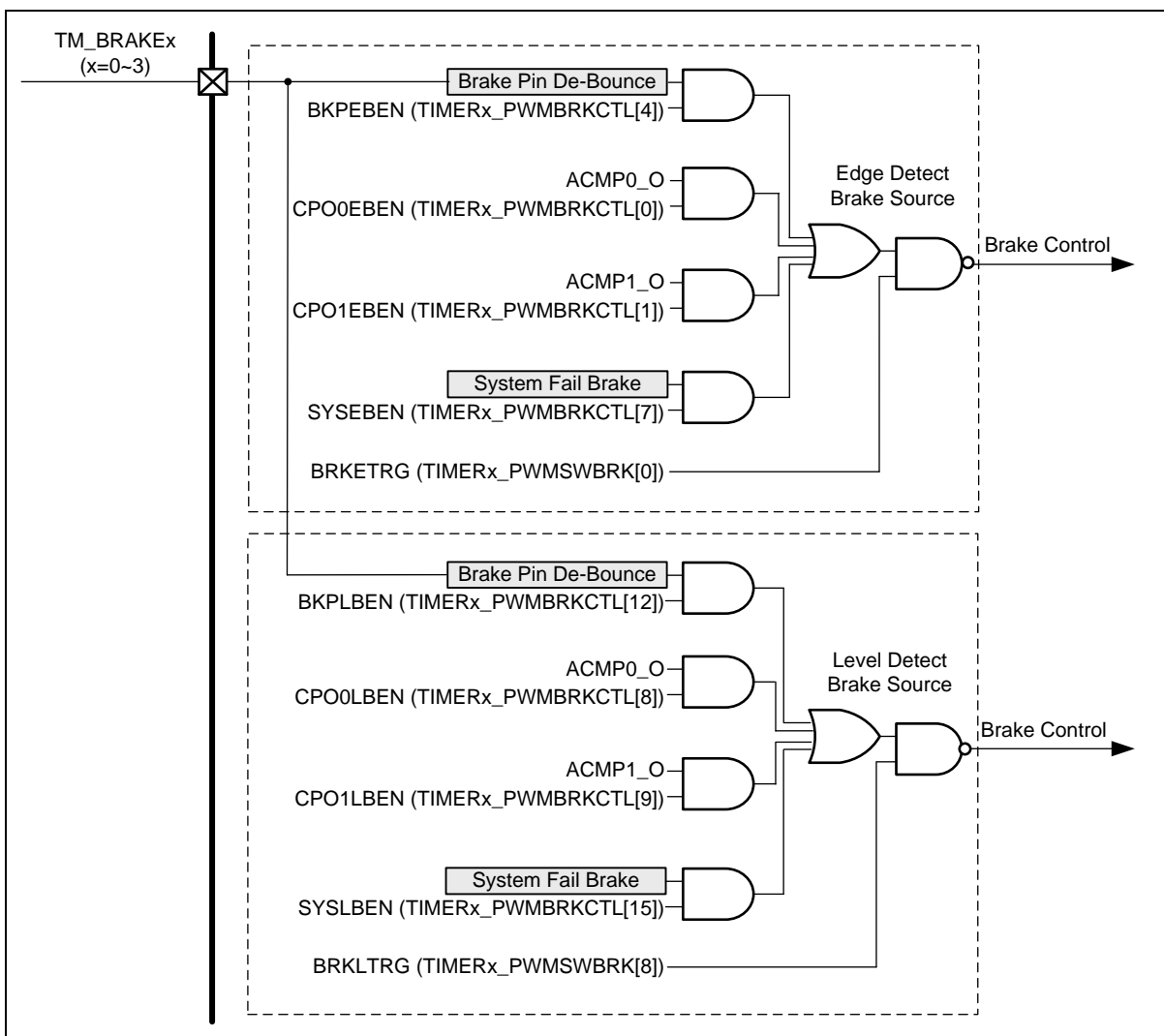


Figure 6.17-32 Brake Source Block Diagram

Among the above described brake sources, the brake source coming from system fail event can be specified to one of the different system fail conditions, these conditions include clock fail, BOD detect and CPU lockup as shown in Figure 6.17-33.

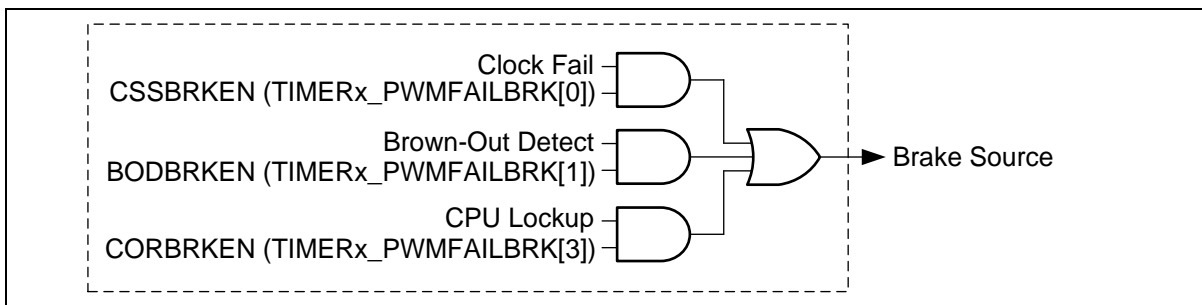


Figure 6.17-33 System Fail Brake Block Diagram

6.17.6.18 Polarity Control

Each PWMx\_CH0 and PWMx\_CH1 has an independent polarity control to configure the polarity of the active state of PWM output. User can control polarity state of PWMx\_CH0 on PINV0

(TIMERx\_PWMPOLCTL[0]) and PWMx\_CH1 on PINV1 (TIMERx\_PWMPOLCTL[1]). Figure 6.17-34 shows the PWMx\_CH0 and PWMx\_CH1 output with polarity control and dead-time insertion.

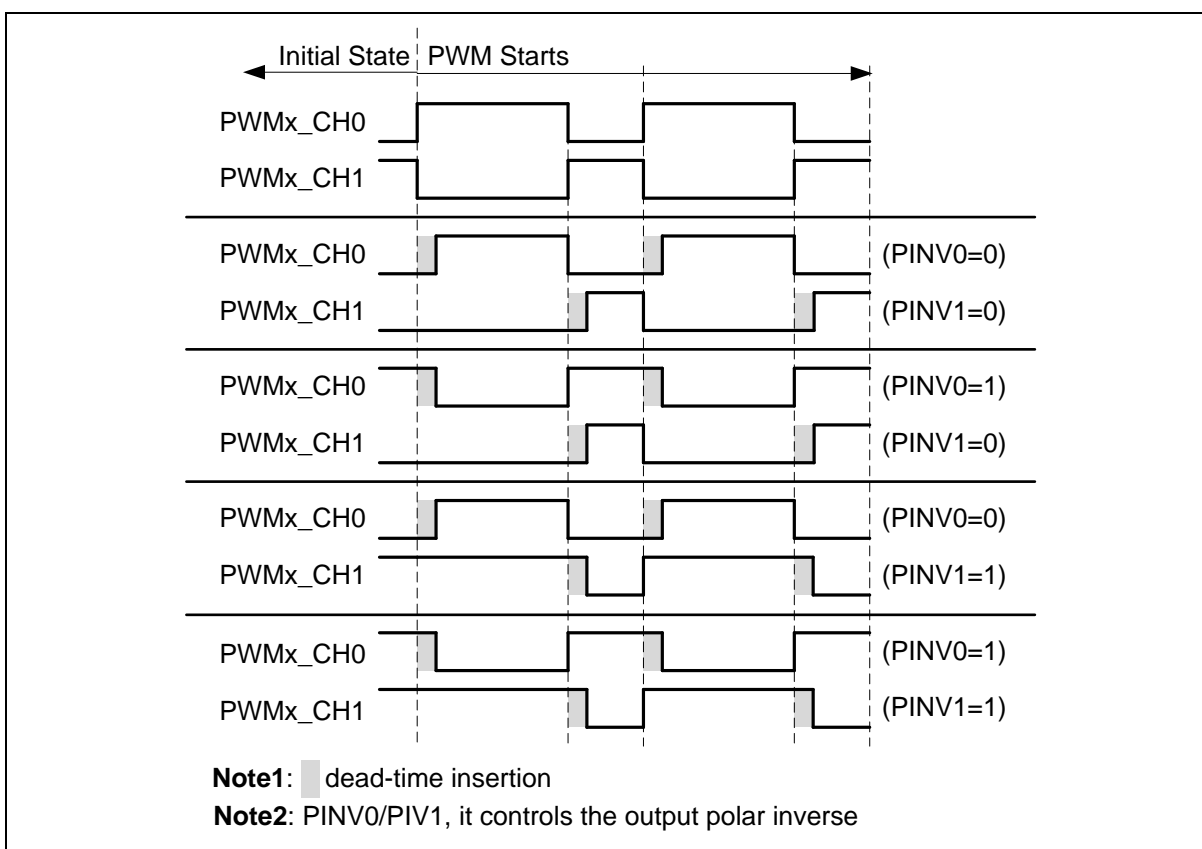


Figure 6.17-34 PWMx\_CH0 and PWMx\_CH1 Polarity Control with Dead-Time Insertion

### 6.17.6.19 PWM Interrupt Generator

There are independent interrupts for each PWM as shown in Figure 6.17-35.

The PWM interrupt (PWMx\_INT) comes from PWM complementary pair events. The counter can generate the zero point interrupt flag ZIF (TIMERx\_PWMINTSTS0[0]) and the period point interrupt flag PIF (TIMERx\_PWMINTSTS0[1]). When counter equals to the comparator value stored in CMP (TIMERx\_PWMCMPDAT[15:0]), the different interrupt flags will be triggered depending on the counting direction. If counter and CMP matched occurs at up-count direction, the comparator up interrupt flag CMPUIF (TIMERx\_PWMINTSTS0[2]) is set and if matched at down-count direction, the comparator down interrupt flag CMPDIF (TIMERx\_PWMINTSTS0[3]) is set. If the corresponding interrupt enable bits are set, the interrupt trigger events will also generates interrupt signals. When PWM brake event occurred, the relatives interrupt event will be triggered according to PWM brake settings.



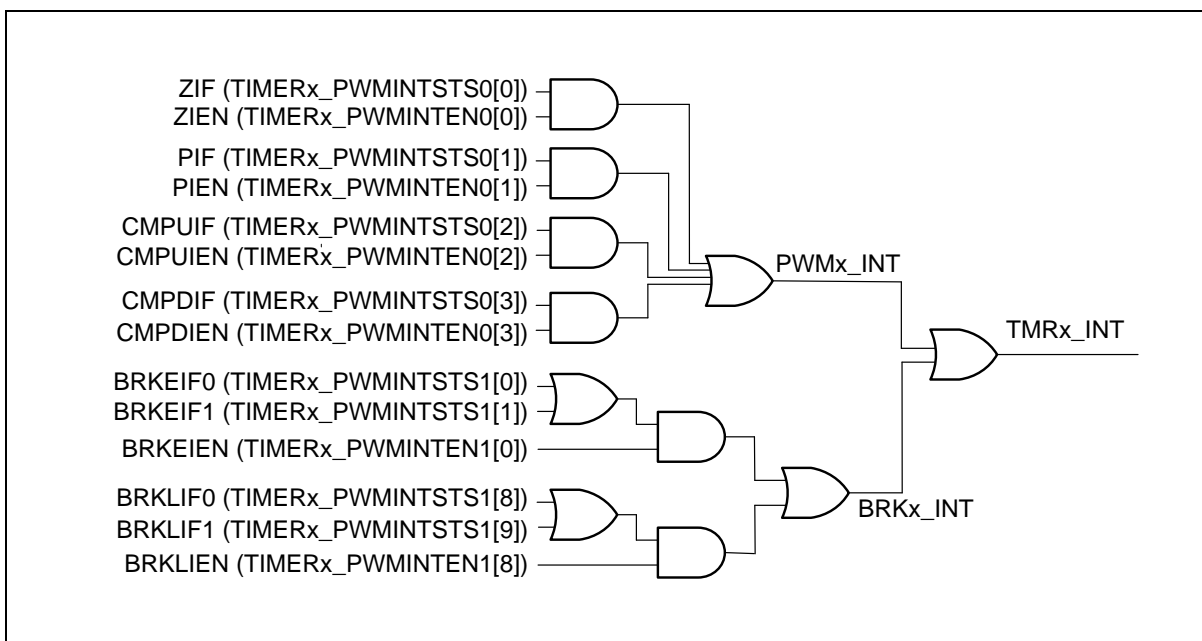


Figure 6.17-35 PWM Interrupt Architecture Diagram

6.17.6.20 PWM Trigger ADC Generator

The PWM counter event can be one of the ADC conversion trigger source. User sets TRGSEL (TIMERx\_PWMADCTS[3:0]) to select which PWM counter event can trigger ADC conversion after TRGEN (TIMERx\_PWMADCTS [7]) is enabled.

There are five PWM counter events can be selected as the trigger source to start ADC conversion which shown in Figure 6.17-36.

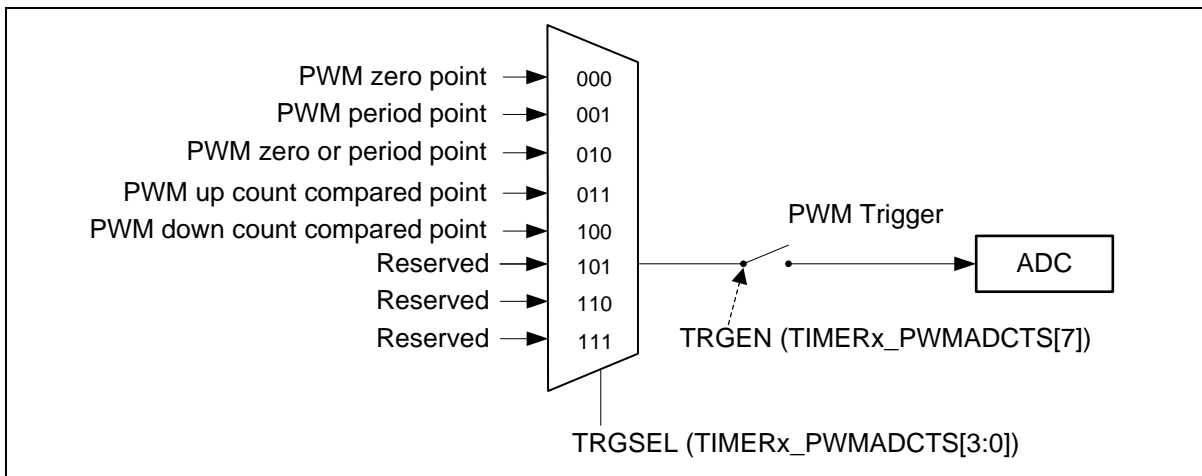


Figure 6.17-36 PWM Trigger ADC Block Diagram

### 6.17.7 Register Map

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

Register	Offset	R/W	Description	Reset Value
<b>TIMER Base Address:</b>				
TMR01_BA = 0x4001_0000				
TMR23_BA = 0x4011_0000				
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
TIMER0_ALTCTL	TMR01_BA+0x20	R/W	Timer0 Alternative Control Register	0x0000_0000
TIMER0_PWMCTL	TMR01_BA+0x40	R/W	Timer0 PWM Control Register	0x0000_0000
TIMER0_PWMCLKSRC	TMR01_BA+0x44	R/W	Timer0 PWM Counter Clock Source Register	0x0000_0000
TIMER0_PWMCLKPSC	TMR01_BA+0x48	R/W	Timer0 PWM Counter Clock Pre-scale Register	0x0000_0000
TIMER0_PWMCNTCLR	TMR01_BA+0x4C	R/W	Timer0 PWM Clear Counter Register	0x0000_0000
TIMER0_PWMPERIOD	TMR01_BA+0x50	R/W	Timer0 PWM Period Register	0x0000_0000
TIMER0_PWMCMPDAT	TMR01_BA+0x54	R/W	Timer0 PWM Comparator Register	0x0000_0000
TIMER0_PWMDTCTL	TMR01_BA+0x58	R/W	Timer0 PWM Dead-Time Control Register	0x0000_0000
TIMER0_PWMCNT	TMR01_BA+0x5C	R	Timer0 PWM Counter Register	0x0000_0000
TIMER0_PWMSKEN	TMR01_BA+0x60	R/W	Timer0 PWM Output Mask Enable Register	0x0000_0000
TIMER0_PWMMSK	TMR01_BA+0x64	R/W	Timer0 PWM Output Mask Data Control Register	0x0000_0000
TIMER0_PWMBNF	TMR01_BA+0x68	R/W	Timer0 PWM Brake Pin Noise Filter Register	0x0000_0000
TIMER0_PWMFAILBRK	TMR01_BA+0x6C	R/W	Timer0 PWM System Fail Brake Control Register	0x0000_0000

TIMER0_PWM BRKCTL	TMR01_BA+0x70	R/W	Timer0 PWM Brake Control Register	0x0000_0000
TIMER0_PWM POLCTL	TMR01_BA+0x74	R/W	Timer0 PWM Pin Output Polar Control Register	0x0000_0000
TIMER0_PWM POEN	TMR01_BA+0x78	R/W	Timer0 PWM Pin Output Enable Register	0x0000_0000
TIMER0_PWM SWBRK	TMR01_BA+0x7C	W	Timer0 PWM Software Trigger Brake Control Register	0x0000_0000
TIMER0_PWM INTEN0	TMR01_BA+0x80	R/W	Timer0 PWM Interrupt Enable Register 0	0x0000_0000
TIMER0_PWM INTEN1	TMR01_BA+0x84	R/W	Timer0 PWM Interrupt Enable Register 1	0x0000_0000
TIMER0_PWM INTSTS0	TMR01_BA+0x88	R/W	Timer0 PWM Interrupt Status Register 0	0x0000_0000
TIMER0_PWM INTSTS1	TMR01_BA+0x8C	R/W	Timer0 PWM Interrupt Status Register 1	0x0000_0000
TIMER0_PWM ADCTS	TMR01_BA+0x90	R/W	Timer0 PWM ADC Trigger Source Select Register	0x0000_0000
TIMER0_PWM SCTL	TMR01_BA+0x94	R/W	Timer0 PWM Synchronous Control Register	0x0000_0000
TIMER0_PWM STRG	TMR01_BA+0x98	W	Timer0 PWM Synchronous Trigger Register	0x0000_0000
TIMER0_PWM STATUS	TMR01_BA+0x9C	R/W	Timer0 PWM Status Register	0x0000_0000
TIMER0_PWM PBUF	TMR01_BA+0xA0	R	Timer0 PWM Period Buffer Register	0x0000_0000
TIMER0_PWM CMPBUF	TMR01_BA+0xA4	R	Timer0 PWM Comparator Buffer 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_INTS TS	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_EXT CTL	TMR01_BA+0x114	R/W	Timer1 External Control Register	0x0000_0000
TIMER1_EINT STS	TMR01_BA+0x118	R/W	Timer1 External Interrupt Status Register	0x0000_0000
TIMER1_TRG CTL	TMR01_BA+0x11C	R/W	Timer1 Trigger Control Register	0x0000_0000
TIMER1_ALT CTL	TMR01_BA+0x120	R/W	Timer1 Alternative Control Register	0x0000_0000
TIMER1_PWM CTL	TMR01_BA+0x140	R/W	Timer1 PWM Control Register	0x0000_0000
TIMER1_PWM	TMR01_BA+0x144	R/W	Timer1 PWM Counter Clock Source Register	0x0000_0000

CLKSRC				
TIMER1_PWM_CLKPSC	TMR01_BA+0x148	R/W	Timer1 PWM Counter Clock Pre-scale Register	0x0000_0000
TIMER1_PWM_CNTCLR	TMR01_BA+0x14C	R/W	Timer1 PWM Clear Counter Register	0x0000_0000
TIMER1_PWM_PERIOD	TMR01_BA+0x150	R/W	Timer1 PWM Period Register	0x0000_0000
TIMER1_PWM_CMPDAT	TMR01_BA+0x154	R/W	Timer1 PWM Comparator Register	0x0000_0000
TIMER1_PWM_DTCTL	TMR01_BA+0x158	R/W	Timer1 PWM Dead-Time Control Register	0x0000_0000
TIMER1_PWM_CNT	TMR01_BA+0x15C	R	Timer1 PWM Counter Register	0x0000_0000
TIMER1_PWM_MSKEN	TMR01_BA+0x160	R/W	Timer1 PWM Output Mask Enable Register	0x0000_0000
TIMER1_PWM_MSK	TMR01_BA+0x164	R/W	Timer1 PWM Output Mask Data Control Register	0x0000_0000
TIMER1_PWM_BNF	TMR01_BA+0x168	R/W	Timer1 PWM Brake Pin Noise Filter Register	0x0000_0000
TIMER1_PWM_FAILBRK	TMR01_BA+0x16C	R/W	Timer1 PWM System Fail Brake Control Register	0x0000_0000
TIMER1_PWM_BRKCTL	TMR01_BA+0x170	R/W	Timer1 PWM Brake Control Register	0x0000_0000
TIMER1_PWM_POLCTL	TMR01_BA+0x174	R/W	Timer1 PWM Pin Output Polar Control Register	0x0000_0000
TIMER1_PWM_POEN	TMR01_BA+0x178	R/W	Timer1 PWM Pin Output Enable Register	0x0000_0000
TIMER1_PWM_SWBRK	TMR01_BA+0x17C	W	Timer1 PWM Software Trigger Brake Control Register	0x0000_0000
TIMER1_PWM_INTEN0	TMR01_BA+0x180	R/W	Timer1 PWM Interrupt Enable Register 0	0x0000_0000
TIMER1_PWM_INTEN1	TMR01_BA+0x184	R/W	Timer1 PWM Interrupt Enable Register 1	0x0000_0000
TIMER1_PWM_INTSTS0	TMR01_BA+0x188	R/W	Timer1 PWM Interrupt Status Register 0	0x0000_0000
TIMER1_PWM_INTSTS1	TMR01_BA+0x18C	R/W	Timer1 PWM Interrupt Status Register 1	0x0000_0000
TIMER1_PWM_ADCTS	TMR01_BA+0x190	R/W	Timer1 PWM ADC Trigger Source Select Register	0x0000_0000
TIMER1_PWM_SCTL	TMR01_BA+0x194	R/W	Timer1 PWM Synchronous Control Register	0x0000_0000
TIMER1_PWM_SSTRG	TMR01_BA+0x198	W	Timer1 PWM Synchronous Start Trigger Register	0x0000_0000
TIMER1_PWM_STATUS	TMR01_BA+0x19C	R/W	Timer1 PWM Status Register	0x0000_0000
TIMER1_PWM_PBUF	TMR01_BA+0x1A0	R	Timer1 PWM Period Buffer Register	0x0000_0000

TIMER1_PWM_CMPBUF	TMR01_BA+0x1A4	R	Timer1 PWM Comparator Buffer 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
TIMER2_ALTCTL	TMR23_BA+0x20	R/W	Timer2 Alternative Control Register	0x0000_0000
TIMER2_PWM_CTL	TMR23_BA+0x40	R/W	Timer2 PWM Control Register	0x0000_0000
TIMER2_PWM_CLKSRC	TMR23_BA+0x44	R/W	Timer2 PWM Counter Clock Source Register	0x0000_0000
TIMER2_PWM_CLKPSC	TMR23_BA+0x48	R/W	Timer2 PWM Counter Clock Pre-scale Register	0x0000_0000
TIMER2_PWM_CNTCLR	TMR23_BA+0x4C	R/W	Timer2 PWM Clear Counter Register	0x0000_0000
TIMER2_PWM_PERIOD	TMR23_BA+0x50	R/W	Timer2 PWM Period Register	0x0000_0000
TIMER2_PWM_CMPDAT	TMR23_BA+0x54	R/W	Timer2 PWM Comparator Register	0x0000_0000
TIMER2_PWM_DTCTL	TMR23_BA+0x58	R/W	Timer2 PWM Dead-Time Control Register	0x0000_0000
TIMER2_PWM_CNT	TMR23_BA+0x5C	R	Timer2 PWM Counter Register	0x0000_0000
TIMER2_PWM_MSKEN	TMR23_BA+0x60	R/W	Timer2 PWM Output Mask Enable Register	0x0000_0000
TIMER2_PWM_MSK	TMR23_BA+0x64	R/W	Timer2 PWM Output Mask Data Control Register	0x0000_0000
TIMER2_PWM_BNF	TMR23_BA+0x68	R/W	Timer2 PWM Brake Pin Noise Filter Register	0x0000_0000
TIMER2_PWM_FAILBRK	TMR23_BA+0x6C	R/W	Timer2 PWM System Fail Brake Control Register	0x0000_0000
TIMER2_PWM_BRKCTL	TMR23_BA+0x70	R/W	Timer2 PWM Brake Control Register	0x0000_0000
TIMER2_PWM_POLCTL	TMR23_BA+0x74	R/W	Timer2 PWM Pin Output Polar Control Register	0x0000_0000
TIMER2_PWM	TMR23_BA+0x78	R/W	Timer2 PWM Pin Output Enable Register	0x0000_0000

POEN				
TIMER2_PWM SWBRK	TMR23_BA+0x7C	W	Timer2 PWM Software Trigger Brake Control Register	0x0000_0000
TIMER2_PWM INTEN0	TMR23_BA+0x80	R/W	Timer2 PWM Interrupt Enable Register 0	0x0000_0000
TIMER2_PWM INTEN1	TMR23_BA+0x84	R/W	Timer2 PWM Interrupt Enable Register 1	0x0000_0000
TIMER2_PWM INTSTS0	TMR23_BA+0x88	R/W	Timer2 PWM Interrupt Status Register 0	0x0000_0000
TIMER2_PWM INTSTS1	TMR23_BA+0x8C	R/W	Timer2 PWM Interrupt Status Register 1	0x0000_0000
TIMER2_PWM ADCTS	TMR23_BA+0x90	R/W	Timer2 PWM ADC Trigger Source Select Register	0x0000_0000
TIMER2_PWM SCTL	TMR23_BA+0x94	R/W	Timer2 PWM Synchronous Control Register	0x0000_0000
TIMER2_PWM STRG	TMR23_BA+0x98	W	Timer2 PWM Synchronous Trigger Register	0x0000_0000
TIMER2_PWM STATUS	TMR23_BA+0x9C	R/W	Timer2 PWM Status Register	0x0000_0000
TIMER2_PWM PBUF	TMR23_BA+0xA0	R	Timer2 PWM Period Buffer Register	0x0000_0000
TIMER2_PWM CMPBUF	TMR23_BA+0xA4	R	Timer2 PWM Comparator Buffer 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_EXT CTL	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_TRG CTL	TMR23_BA+0x11C	R/W	Timer3 Trigger Control Register	0x0000_0000
TIMER3_ALT CTL	TMR23_BA+0x120	R/W	Timer3 Alternative Control Register	0x0000_0000
TIMER3_PWM CTL	TMR23_BA+0x140	R/W	Timer3 PWM Control Register	0x0000_0000
TIMER3_PWM CLKSRC	TMR23_BA+0x144	R/W	Timer3 PWM Counter Clock Source Register	0x0000_0000
TIMER3_PWM CLKPSC	TMR23_BA+0x148	R/W	Timer3 PWM Counter Clock Pre-scale Register	0x0000_0000
TIMER3_PWM CNTCLR	TMR23_BA+0x14C	R/W	Timer3 PWM Clear Counter Register	0x0000_0000

TIMER3_PWM PERIOD	TMR23_BA+0x150	R/W	Timer3 PWM Period Register	0x0000_0000
TIMER3_PWM CMPDAT	TMR23_BA+0x154	R/W	Timer3 PWM Comparator Register	0x0000_0000
TIMER3_PWM DTCTL	TMR23_BA+0x158	R/W	Timer3 PWM Dead-Time Control Register	0x0000_0000
TIMER3_PWM CNT	TMR23_BA+0x15C	R	Timer3 PWM Counter Register	0x0000_0000
TIMER3_PWM MSKEN	TMR23_BA+0x160	R/W	Timer3 PWM Output Mask Enable Register	0x0000_0000
TIMER3_PWM MSK	TMR23_BA+0x164	R/W	Timer3 PWM Output Mask Data Control Register	0x0000_0000
TIMER3_PWM BNF	TMR23_BA+0x168	R/W	Timer3 PWM Brake Pin Noise Filter Register	0x0000_0000
TIMER3_PWM FAILBRK	TMR23_BA+0x16C	R/W	Timer3 PWM System Fail Brake Control Register	0x0000_0000
TIMER3_PWM BRKCTL	TMR23_BA+0x170	R/W	Timer3 PWM Brake Control Register	0x0000_0000
TIMER3_PWM POLCTL	TMR23_BA+0x174	R/W	Timer3 PWM Pin Output Polar Control Register	0x0000_0000
TIMER3_PWM POEN	TMR23_BA+0x178	R/W	Timer3 PWM Pin Output Enable Register	0x0000_0000
TIMER3_PWM SWBRK	TMR23_BA+0x17C	W	Timer3 PWM Software Trigger Brake Control Register	0x0000_0000
TIMER3_PWM INTEN0	TMR23_BA+0x180	R/W	Timer3 PWM Interrupt Enable Register 0	0x0000_0000
TIMER3_PWM INTEN1	TMR23_BA+0x184	R/W	Timer3 PWM Interrupt Enable Register 1	0x0000_0000
TIMER3_PWM INTSTS0	TMR23_BA+0x188	R/W	Timer3 PWM Interrupt Status Register 0	0x0000_0000
TIMER3_PWM INTSTS1	TMR23_BA+0x18C	R/W	Timer3 PWM Interrupt Status Register 1	0x0000_0000
TIMER3_PWM ADCTS	TMR23_BA+0x190	R/W	Timer3 PWM ADC Trigger Source Select Register	0x0000_0000
TIMER3_PWM SCTL	TMR23_BA+0x194	R/W	Timer3 PWM Synchronous Control Register	0x0000_0000
TIMER3_PWM SSTRG	TMR23_BA+0x198	W	Timer3 PWM Synchronous Start Trigger Register	0x0000_0000
TIMER3_PWM STATUS	TMR23_BA+0x19C	R/W	Timer3 PWM Status Register	0x0000_0000
TIMER3_PWM PBUF	TMR23_BA+0x1A0	R	Timer3 PWM Period Buffer Register	0x0000_0000
TIMER3_PWM CMPBUF	TMR23_BA+0x1A4	R	Timer3 PWM Comparator Buffer Register	0x0000_0000

6.17.8 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	<p><b>ICE Debug Mode Acknowledge Disable Bit (Write Protect)</b></p> <p>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.</p> <p><b>Note:</b> This bit is write protected. Refer to the SYS_REGLCTL register.</p>
[30]	CNTEN	<p><b>Timer Counting Enable Bit</b></p> <p>0 = Stop/Suspend counting.                      1 = Start counting.</p> <p><b>Note1:</b> 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.</p> <p><b>Note2:</b> 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.</p> <p><b>Note3:</b> 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.</p>
[29]	INTEN	<p><b>Timer Interrupt Enable Bit</b></p> <p>0 = Timer time-out interrupt Disabled.                      1 = Timer time-out interrupt Enabled.</p> <p><b>Note:</b> 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.</p>
[28:27]	OPMODE	<p><b>Timer Counting Mode Select</b></p> <p>00 = The Timer controller is operated in One-shot mode.                      01 = The Timer controller is operated in Periodic mode.</p>



		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	<b>Timer Active Status Bit (Read Only)</b> This bit indicates the 24-bit up counter status. 0 = 24-bit up counter is not active. 1 = 24-bit up counter is active.
[24]	EXTCNTEN	<b>Event Counter Mode Enable Bit</b> This bit is for external counting pin function enabled. 0 = Event counter mode Disabled. 1 = Event counter mode Enabled. <b>Note:</b> When timer is used as an event counter, this bit should be set to 1 and select PCLK as timer clock source.
[23]	WKEN	<b>Wake-up Function Enable Bit</b> 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	<b>Capture Pin Source Selection</b> 0 = Capture Function source is from Tx_EXT (x= 0~3) pin. 1 = Capture Function source is from internal ACMP output signal. User can set ACMPSEL (TIMERx_EXTCTL[8]) to decide which internal ACMP output signal as timer capture source.
[21]	TGLPINSEL	<b>Toggle-output Pin Select</b> 0 = Toggle mode output to Tx (Timer Event Counter Pin). 1 = Toggle mode output to Tx_EXT (Timer External Capture Pin).
[20]	PERIOSEL	<b>Periodic Mode Behavior Selection Enable Bit</b> 0 = The behavior selection in periodic mode 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 Enabled. When user updates CMPDAT while timer is running in periodic mode, the limitations as bellows list, <b>Note:</b> If the updated CMPDAT value > CNT, CMPDAT will be updated and CNT keep running continually. If the updated CMPDAT value = CNT, timer time-out interrupt will be asserted immediately. If the updated CMPDAT value < CNT, CNT will be reset to default value.
[19]	INTRGEN	<b>Inter-timer Trigger Mode Enable Bit</b> 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. <b>Note:</b> For Timer1/3, this bit is ignored and the read back value is always 0.
[18:8]	Reserved	Reserved.

[7:0]	<b>PSC</b>	<p><b>Prescale Counter</b></p> <p>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.</p> <p><b>Note:</b> Update prescale counter value will reset internal 8-bit prescale counter and 24-bit up counter value.</p>
-------	------------	---

**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]	CMPDAT	<p><b>Timer Comparator Value</b></p> <p>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.</p> <p>Time-out period = (Period of timer clock input) * (8-bit PSC + 1) * (24-bit CMPDAT).</p> <p><b>Note1:</b> Never write 0x0 or 0x1 in CMPDAT field, or the core will run into unknown state.</p> <p><b>Note2:</b> 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.</p>

**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]	TWKF	<p><b>Timer Wake-up Flag</b></p> <p>This bit indicates the interrupt wake-up flag status of timer.</p> <p>0 = Timer does not cause CPU wake-up.</p> <p>1 = CPU wake-up from Idle or Power-down mode if timer time-out interrupt signal generated.</p> <p><b>Note:</b> This bit is cleared by writing 1 to it.</p>
[0]	TIF	<p><b>Timer Interrupt Flag</b></p> <p>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.</p> <p>0 = No effect.</p> <p>1 = CNT value matches the CMPDAT value.</p> <p><b>Note:</b> This bit is cleared by writing 1 to it.</p>

**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	<p><b>Timer Data Register Reset Active (Read Only)</b></p> <p>This bit indicates if the counter reset operation active.</p> <p>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.</p> <p>0 = Reset operation is done. 1 = Reset operation triggered by writing TIMERx_CNT is in progress.</p>
[30:24]	Reserved	Reserved.
[23:0]	CNT	<p><b>Timer Data Register</b></p> <p><b>Read operation:</b></p> <p>Read this register to get CNT value. For example:</p> <p>If EXTCNTEN (TIMERx_CTL[24] ) is 0, user can read CNT value for getting current 24-bit counter value.</p> <p>If EXTCNTEN (TIMERx_CTL[24] ) is 1, user can read CNT value for getting current 24-bit event input counter value.</p> <p><b>Write operation:</b></p> <p>Writing any value to this register will reset current CNT value to 0 and reload internal 8-bit prescale counter.</p>

**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	<p><b>Timer Capture Data Register</b></p> <p>When CAPEN (TIMERx_EXTCTL[3]) bit is set, CAPFUNCS (TIMERx_EXTCTL[4]) bit is 0, and a transition on Tx_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.</p>

**Timer External Control Register (TIMERx\_EXTCTL)**

Register	Offset	R/W	Description	Reset Value
TIMER0_EXT CTL	TMR01_BA+0x14	R/W	Timer0 External Control Register	0x0000_0000
TIMER1_EXT CTL	TMR01_BA+0x114	R/W	Timer1 External Control Register	0x0000_0000
TIMER2_EXT CTL	TMR23_BA+0x14	R/W	Timer2 External Control Register	0x0000_0000
TIMER3_EXT CTL	TMR23_BA+0x114	R/W	Timer3 External Control Register	0x0000_0000

31	30	29	28	27	26	25	24	
Reserved								
23	22	21	20	19	18	17	16	
Reserved							ECNTSSEL	
15	14	13	12	11	10	9	8	
Reserved		CAPEDGE			Reserved			ACMPSEL
7	6	5	4	3	2	1	0	
CNTDBEN	CAPDBEN	CAPIEN	CAPFUNCS	CAPEN	Reserved		CNTPHASE	

Bits	Description
[31:17]	<b>Reserved</b> Reserved.
[16]	<b>ECNTSSEL</b> <b>Event Counter Source Selection to Trigger Event Counter Function</b> 0 = Event Counter input source is from Tx (x= 0~3) pin. 1 = Event Counter input source is from USB internal SOF output signal.
[15]	<b>Reserved</b> Reserved.
[14:12]	<b>Timer External Capture Pin Edge Detect</b> 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 Tx_EXT (x= 0~3) pin. 001 = Capture event occurred when detect rising edge transfer on Tx_EXT (x= 0~3) pin. 010 = Capture event occurred when detect both falling and rising edge transfer on Tx_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 Tx_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 Tx_EXT (x= 0~3) pin. 111 = First capture event occurred at rising edge, follows capture events are at falling edge transfer on Tx_EXT (x= 0~3) pin. 100, 101 = Reserved.
[11:9]	<b>Reserved</b> Reserved.

[8]	ACMPSEL	<p><b>ACMP Source Selection to Trigger Capture Function</b></p> <p>0 = Capture Function source is from internal ACMP0 output signal.          1 = Capture Function source is from internal ACMP1 output signal.</p> <p><b>Note:</b> these bits only available when CAPSRC (TIMERx_CTL[22]) is 1.</p>
[7]	CNTDBEN	<p><b>Timer Counter Pin De-bounce Enable Bit</b></p> <p>0 = Tx (x= 0~3) pin de-bounce Disabled.          1 = Tx (x= 0~3) pin de-bounce Enabled.</p> <p><b>Note:</b> If this bit is enabled, the edge detection of Tx pin is detected with de-bounce circuit.</p>
[6]	CAPDBEN	<p><b>Timer External Capture Pin De-bounce Enable Bit</b></p> <p>0 = Tx_EXT (x= 0~3) pin de-bounce or ACMP output de-bounce Disabled.          1 = Tx_EXT (x= 0~3) pin de-bounce or ACMP output de-bounce Enabled.</p> <p><b>Note:</b> If this bit is enabled, the edge detection of Tx_EXT pin or ACMP output is detected with de-bounce circuit.</p>
[5]	CAPIEN	<p><b>Timer External Capture Interrupt Enable Bit</b></p> <p>0 = Tx_EXT (x= 0~3) pin detection Interrupt Disabled.          1 = Tx_EXT (x= 0~3) pin detection Interrupt Enabled.</p> <p><b>Note:</b> CAPIEN is used to enable timer external interrupt. If CAPIEN enabled, timer will rise an interrupt when CAPIF (TIMERx_EINTSTS[0]) is 1.</p> <p>For example, while CAPIEN = 1, CAPEN = 1, and CAPEDGE = 00, a 1 to 0 transition on the Tx_EXT pin will cause the CAPIF to be set then the interrupt signal is generated and sent to NVIC to inform CPU.</p>
[4]	CAPFUNCS	<p><b>Capture Function Selection</b></p> <p>0 = External Capture Mode Enabled.          1 = External Reset Mode Enabled.</p> <p><b>Note1:</b> When CAPFUNCS is 0, transition on Tx_EXT (x= 0~3) pin is using to save current 24-bit timer counter value (CNT value) to CAPDAT field.</p> <p><b>Note2:</b> When CAPFUNCS is 1, transition on Tx_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.</p>
[3]	CAPEN	<p><b>Timer External Capture Pin Enable Bit</b></p> <p>This bit enables the Tx_EXT capture pin input function.</p> <p>0 =Tx_EXT (x= 0~3) pin Disabled.          1 =Tx_EXT (x= 0~3) pin Enabled.</p>
[2:1]	Reserved	Reserved.
[0]	CNTPHASE	<p><b>Timer External Count Phase</b></p> <p>This bit indicates the detection phase of external counting pin Tx (x= 0~3).</p> <p>0 = A falling edge of external counting pin will be counted.          1 = A rising edge of external counting pin will be counted.</p>



**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
[0]	<p><b>CAPIF</b></p> <p><b>Timer External Capture Interrupt Flag</b>                      This bit indicates the timer external capture interrupt flag status.                      0 = Tx_EXT (x= 0~3) pin interrupt did not occur.                      1 = Tx_EXT (x= 0~3) pin interrupt occurred.</p> <p><b>Note1:</b> This bit is cleared by writing 1 to it.</p> <p><b>Note2:</b> When CAPEN (TIMERx_EXTCTL[3]) bit is set, CAPFUNCS (TIMERx_EXTCTL[4]) bit is 0, and a transition on Tx_EXT (x= 0~3) pin matched the CAPEDGE (TIMERx_EXTCTL[2:1]) setting, this bit will set to 1 by hardware.</p> <p><b>Note3:</b> 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.</p>

**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	Reserved	TRGADC	TRGPWM	TRGSSEL

Bits	Description
[31:5]	<b>Reserved</b> Reserved.
[4]	<b>TRGPDMA</b> <b>Trigger PDMA Enable Bit</b> 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. <b>Note:</b> 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]	<b>Reserved</b> Reserved.
[2]	<b>TRGADC</b> <b>Trigger ADC Enable Bit</b> If this bit is set to 1, each timer time-out event or capture event can be triggered ADC conversion. 0 = Timer interrupt trigger ADC Disabled. 1 = Timer interrupt trigger ADC Enabled. <b>Note:</b> If TRGSSEL (TIMERx_TRGCTL[0]) = 0, time-out interrupt signal will trigger ADC conversion. If TRGSSEL (TIMERx_TRGCTL[0]) = 1, capture interrupt signal will trigger ADC conversion.
[1]	<b>TRGPWM</b> <b>Trigger PWM Enable Bit</b> If this bit is set to 1, each timer time-out event or capture event can be as PWM counter

		<p>clock source.</p> <p>0 = Timer interrupt trigger PWM Disabled.</p> <p>1 = Timer interrupt trigger PWM Enabled.</p> <p><b>Note:</b> If TRGSSEL (TIMERx_TRGCTL[0]) = 0, time-out interrupt signal as PWM counter clock source.</p> <p>If TRGSSEL (TIMERx_TRGCTL[0]) = 1, capture interrupt signal as PWM counter clock source.</p>
[0]	TRGSSEL	<p><b>Trigger Source Select Bit</b></p> <p>This bit is used to select internal trigger source is form timer time-out interrupt signal or capture interrupt signal.</p> <p>0 = Time-out interrupt signal is used to internal trigger PWM, PDMA, and ADC.</p> <p>1 = Capture interrupt signal is used to internal trigger PWM, PDMA, and ADC.</p>

**Timer Alternative Control Register (TIMERx\_ALTCTL)**

Register	Offset	R/W	Description	Reset Value
TIMER0_ALTCTL	TMR01_BA+0x20	R/W	Timer0 Alternative Control Register	0x0000_0000
TIMER1_ALTCTL	TMR01_BA+0x120	R/W	Timer1 Alternative Control Register	0x0000_0000
TIMER2_ALTCTL	TMR23_BA+0x20	R/W	Timer2 Alternative Control Register	0x0000_0000
TIMER3_ALTCTL	TMR23_BA+0x120	R/W	Timer3 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							FUNCSEL

Bits	Description
[31:1]	<b>Reserved</b> Reserved.
[0]	<b>FUNCSEL</b> <b>Function Selection</b> 0 = Timer controller is used as timer function. 1 = Timer controller is used as PWM function. <b>Note:</b> When timer is used as PWM, the clock source of time controller will be forced to PCLKx automatically.

**Timer PWM Control Register (TIMERx\_PWMCTL)**

Register	Offset	R/W	Description	Reset Value
TIMER0_PWMCTL	TMR01_BA+0x40	R/W	Timer0 PWM Control Register	0x0000_0000
TIMER1_PWMCTL	TMR01_BA+0x140	R/W	Timer1 PWM Control Register	0x0000_0000
TIMER2_PWMCTL	TMR23_BA+0x40	R/W	Timer2 PWM Control Register	0x0000_0000
TIMER3_PWMCTL	TMR23_BA+0x140	R/W	Timer3 PWM Control Register	0x0000_0000

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

Bits	Description	
[31]	DBGTRIOFF	<p><b>ICE Debug Mode Acknowledge Disable Bit (Write Protect)</b></p> <p>0 = ICE debug mode acknowledgement effects PWM output.                      PWM output pin will be forced as tri-state while ICE debug mode acknowledged.                      1 = ICE debug mode acknowledgement Disabled.                      PWM output pin will keep output no matter ICE debug mode acknowledged or not.  <b>Note:</b> This register is write protected. Refer to SYS_REGLCTL register.</p>
[30]	DBGHALT	<p><b>ICE Debug Mode Counter Halt (Write Protect)</b></p> <p>If debug mode counter halt is enabled, PWM counter will keep current value until exit ICE debug mode.                      0 = ICE debug mode counter halt Disabled.                      1 = ICE debug mode counter halt Enabled.  <b>Note:</b> This register is write protected. Refer to SYS_REGLCTL register.</p>
[29:17]	Reserved	Reserved.
[16]	OUTMODE	<p><b>PWM Output Mode</b></p> <p>This bit controls the output mode of corresponding PWM channel.                      0 = PWM independent mode.                      1 = PWM complementary mode.</p>
[15:10]	Reserved	Reserved.
[9]	IMMLDEN	<p><b>Immediately Load Enable Bit</b></p> <p>0 = PERIOD will load to PBUF when current PWM period is completed no matter CTRLD</p>

		<p>is enabled/disabled. If CTRLD is disabled, CMP will load to CMPBUF when current PWM period is completed; if CTRLD is enabled in up-down count type, CMP will load to CMPBUF at the center point of current period.</p> <p>1 = PERIOD/CMP will load to PBUF/CMPBUF immediately when user update PERIOD/CMP.</p> <p><b>Note:</b> If IMMLDEN is enabled, CTRLD will be invalid.</p>
[8]	<b>CTRLD</b>	<p><b>Center Re-load</b></p> <p>In up-down count type, PERIOD will load to PBUF when current PWM period is completed always and CMP will load to CMPBUF at the center point of current period.</p>
[7:4]	<b>Reserved</b>	Reserved.
[3]	<b>CNTMODE</b>	<p><b>PWM Counter Mode</b></p> <p>0 = Auto-reload mode. 1 = One-shot mode.</p>
[2:1]	<b>CNTTYPE</b>	<p><b>PWM Counter Behavior Type</b></p> <p>00 = Up count type. 01 = Down count type. 10 = Up-down count type. 11 = Reserved.</p>
[0]	<b>CNTEN</b>	<p><b>PWM Counter Enable Bit</b></p> <p>0 = PWM counter and clock prescale Stop Running. 1 = PWM counter and clock prescale Start Running.</p>

**Timer PWM Counter Clock Source Register (TIMERx PWMCLKSRC)**

Register	Offset	R/W	Description	Reset Value
TIMER0_PWM_CLKSRC	TMR01_BA+0x44	R/W	Timer0 PWM Counter Clock Source Register	0x0000_0000
TIMER1_PWM_CLKSRC	TMR01_BA+0x144	R/W	Timer1 PWM Counter Clock Source Register	0x0000_0000
TIMER2_PWM_CLKSRC	TMR23_BA+0x44	R/W	Timer2 PWM Counter Clock Source Register	0x0000_0000
TIMER3_PWM_CLKSRC	TMR23_BA+0x144	R/W	Timer3 PWM Counter 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					CLKSRC		

Bits	Description	
[31:3]	Reserved	Reserved.
[2:0]	CLKSRC	<p><b>PWM Counter Clock Source Select</b></p> <p>The PWM counter clock source can be selected from TMRx_CLK or internal timer time-out or capture event.</p> <p>000 = TMRx_CLK.</p> <p>001 = Internal TIMER0 time-out or capture event.</p> <p>010 = Internal TIMER1 time-out or capture event.</p> <p>011 = Internal TIMER2 time-out or capture event.</p> <p>100 = Internal TIMER3 time-out or capture event.</p> <p>Others = Reserved.</p> <p><b>Note:</b> If TIMER0 PWM function is enabled, the PWM counter clock source can be selected from TMR0_CLK, TIMER1 interrupt events, TIMER2 interrupt events, or TIMER3 interrupt events.</p>

**Timer PWM Counter Clock Pre-scale Register (TIMERx PWMCLKPSC)**

Register	Offset	R/W	Description	Reset Value
TIMER0_PWM_CLKPSC	TMR01_BA+0x48	R/W	Timer0 PWM Counter Clock Pre-scale Register	0x0000_0000
TIMER1_PWM_CLKPSC	TMR01_BA+0x148	R/W	Timer1 PWM Counter Clock Pre-scale Register	0x0000_0000
TIMER2_PWM_CLKPSC	TMR23_BA+0x48	R/W	Timer2 PWM Counter Clock Pre-scale Register	0x0000_0000
TIMER3_PWM_CLKPSC	TMR23_BA+0x148	R/W	Timer3 PWM Counter Clock Pre-scale 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	<b>PWM Counter Clock Pre-scale</b> The active clock of PWM counter is decided by counter clock prescale and divided by (CLKPSC + 1). If CLKPSC is 0, then there is no scaling in PWM counter clock source.



**Timer PWM Clear Counter Register (TIMERx PWMCNTCLR)**

Register	Offset	R/W	Description	Reset Value
TIMER0_PWM CNTCLR	TMR01_BA+0x4C	R/W	Timer0 PWM Clear Counter Register	0x0000_0000
TIMER1_PWM CNTCLR	TMR01_BA+0x14C	R/W	Timer1 PWM Clear Counter Register	0x0000_0000
TIMER2_PWM CNTCLR	TMR23_BA+0x4C	R/W	Timer2 PWM Clear Counter Register	0x0000_0000
TIMER3_PWM CNTCLR	TMR23_BA+0x14C	R/W	Timer3 PWM 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							CNTCLR

Bits	Description	
[31:1]	Reserved	Reserved.
[0]	CNTCLR	<p><b>Clear PWM Counter Control Bit</b></p> <p>It is automatically cleared by hardware.</p> <p>0 = No effect.</p> <p>1 = Clear 16-bit PWM counter to 0x10000 in up and up-down count type and reset counter value to PERIOD in down count type.</p>

**Timer PWM Period Register (TIMERx PWMPERIOD)**

Register	Offset	R/W	Description	Reset Value
TIMER0_PWM PERIOD	TMR01_BA+0x50	R/W	Timer0 PWM Period Register	0x0000_0000
TIMER1_PWM PERIOD	TMR01_BA+0x150	R/W	Timer1 PWM Period Register	0x0000_0000
TIMER2_PWM PERIOD	TMR23_BA+0x50	R/W	Timer2 PWM Period Register	0x0000_0000
TIMER3_PWM PERIOD	TMR23_BA+0x150	R/W	Timer3 PWM 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]	<b>Reserved</b> Reserved.
[15:0]	<p><b>PERIOD</b></p> <p><b>PWM Period Register</b>                      In up count type: PWM counter counts from 0 to PERIOD, and restarts from 0.                      In down count type: PWM counter counts from PERIOD to 0, and restarts from PERIOD.                      In up-down count type: PWM counter counts from 0 to PERIOD, then decrements to 0 and repeats again.                      In up and down count type:                      PWM period time = (PERIOD + 1) * (CLKPSC + 1) * TMRx_PWMCLK.                      In up-down count type:                      PWM period time = 2 * PERIOD * (CLKPSC + 1) * TMRx_PWMCLK.  <b>Note:</b> User should take care DIRF (TIMERx_PWMCNT[16]) bit in up/down/up-down count type to monitor current counter direction in each count type.</p>

**Timer PWM Comparator Register (TIMERx PWMCPDAT)**

Register	Offset	R/W	Description	Reset Value
TIMER0_PWMCPDAT	TMR01_BA+0x54	R/W	Timer0 PWM Comparator Register	0x0000_0000
TIMER1_PWMCPDAT	TMR01_BA+0x154	R/W	Timer1 PWM Comparator Register	0x0000_0000
TIMER2_PWMCPDAT	TMR23_BA+0x54	R/W	Timer2 PWM Comparator Register	0x0000_0000
TIMER3_PWMCPDAT	TMR23_BA+0x154	R/W	Timer3 PWM Comparator 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
CMP							
7	6	5	4	3	2	1	0
CMP							

Bits	Description	
[31:16]	Reserved	Reserved.
[15:0]	CMP	<b>PWM Comparator Register</b> PWM CMP is used to compare with PWM CNT to generate PWM output waveform, interrupt events and trigger ADC to start convert.

**Timer PWM Dead-time Control Register (TIMERx PWMDTCTL)**

Register	Offset	R/W	Description	Reset Value
TIMER0_PWM DTCTL	TMR01_BA+0x58	R/W	Timer0 PWM Dead-Time Control Register	0x0000_0000
TIMER1_PWM DTCTL	TMR01_BA+0x158	R/W	Timer1 PWM Dead-Time Control Register	0x0000_0000
TIMER2_PWM DTCTL	TMR23_BA+0x58	R/W	Timer2 PWM Dead-Time Control Register	0x0000_0000
TIMER3_PWM DTCTL	TMR23_BA+0x158	R/W	Timer3 PWM Dead-Time Control Register	0x0000_0000

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

Bits	Description
[31:25]	<b>Reserved</b> Reserved.
[24]	<b>DTCKSEL</b> <b>Dead-time Clock Select (Write Protect)</b> 0 = Dead-time clock source from TMRx_PWMCLK without counter clock prescale. 1 = Dead-time clock source from TMRx_PWMCLK with counter clock prescale. <b>Note:</b> This register is write protected. Refer to SYS_REGLCTL register.
[23:17]	<b>Reserved</b> Reserved.
[16]	<b>DTEN</b> <b>Enable Dead-time Insertion for PWMx_CH0 and PWMx_CH1 (Write Protect)</b> Dead-time insertion function is only active when PWM complementary mode is enabled. If dead-time insertion is inactive, the outputs of PWMx_CH0 and PWMx_CH1 are complementary without any delay. 0 = Dead-time insertion Disabled on the pin pair. 1 = Dead-time insertion Enabled on the pin pair. <b>Note:</b> This register is write protected. Refer to SYS_REGLCTL register.
[15:12]	<b>Reserved</b> Reserved.
[11:0]	<b>DTCNT</b> <b>Dead-time Counter (Write Protect)</b> The dead-time can be calculated from the following two formulas: Dead-time = (DTCNT[11:0] + 1) * TMRx_PWMCLK, if DTCKSEL is 0. Dead-time = (DTCNT[11:0] + 1) * TMRx_PWMCLK * (CLKPSC + 1), if DTCKSEL is 1. <b>Note:</b> This register is write protected. Refer to SYS_REGLCTL register.

**Timer PWM Counter Register (TIMERx PWMCNT)**

Register	Offset	R/W	Description	Reset Value
TIMER0_PWM CNT	TMR01_BA+0x5C	R	Timer0 PWM Counter Register	0x0000_0000
TIMER1_PWM CNT	TMR01_BA+0x15C	R	Timer1 PWM Counter Register	0x0000_0000
TIMER2_PWM CNT	TMR23_BA+0x5C	R	Timer2 PWM Counter Register	0x0000_0000
TIMER3_PWM CNT	TMR23_BA+0x15C	R	Timer3 PWM 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	<b>PWM Counter Direction Indicator Flag (Read Only)</b> 0 = Counter is active in down counting. 1 = Counter is active in up counting.
[15:0]	CNT	<b>PWM Counter Value Register (Read Only)</b> User can monitor CNT to know the current counter value in 16-bit period counter.

**Timer PWM Output Mask Enable Register (TIMERx PWMMSKEN)**

Register	Offset	R/W	Description	Reset Value
<b>TIMER0_PWM MSKEN</b>	TMR01_BA+0x60	R/W	Timer0 PWM Output Mask Enable Register	0x0000_0000
<b>TIMER1_PWM MSKEN</b>	TMR01_BA+0x160	R/W	Timer1 PWM Output Mask Enable Register	0x0000_0000
<b>TIMER2_PWM MSKEN</b>	TMR23_BA+0x60	R/W	Timer2 PWM Output Mask Enable Register	0x0000_0000
<b>TIMER3_PWM MSKEN</b>	TMR23_BA+0x160	R/W	Timer3 PWM Output 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						MSKEN1	MSKEN0

Bits	Description	
[31:2]	Reserved	Reserved.
[1]	<b>MSKEN1</b>	<p><b>PWMx_CH1 Output Mask Enable Bit</b></p> <p>The PWMx_CH1 output signal will be masked when this bit is enabled. The PWMx_CH1 will output MSKDAT1 (TIMER_PWMMSK[1]) data.</p> <p>0 = PWMx_CH1 output signal is non-masked.</p> <p>1 = PWMx_CH1 output signal is masked and output MSKDAT1 data.</p>
[0]	<b>MSKEN0</b>	<p><b>PWMx_CH0 Output Mask Enable Bit</b></p> <p>The PWMx_CH0 output signal will be masked when this bit is enabled. The PWMx_CH0 will output MSKDAT0 (TIMER_PWMMSK[0]) data.</p> <p>0 = PWMx_CH0 output signal is non-masked.</p> <p>1 = PWMx_CH0 output signal is masked and output MSKDAT0 data.</p>

**Timer PWM Output Mask Data Control Register (TIMERx PWMMSK)**

Register	Offset	R/W	Description	Reset Value
<b>TIMER0_PWMMSK</b>	TMR01_BA+0x64	R/W	Timer0 PWM Output Mask Data Control Register	0x0000_0000
<b>TIMER1_PWMMSK</b>	TMR01_BA+0x164	R/W	Timer1 PWM Output Mask Data Control Register	0x0000_0000
<b>TIMER2_PWMMSK</b>	TMR23_BA+0x64	R/W	Timer2 PWM Output Mask Data Control Register	0x0000_0000
<b>TIMER3_PWMMSK</b>	TMR23_BA+0x164	R/W	Timer3 PWM Output Mask Data 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						MSKDAT1	MSKDAT0

Bits	Description	
[31:2]	<b>Reserved</b>	Reserved.
[1]	<b>MSKDAT1</b>	<p><b>PWMx_CH1 Output Mask Data Control Bit</b></p> <p>This bit is used to control the output state of PWMx_CH1 pin when PWMx_CH1 output mask function is enabled (MSKEN1 = 1).</p> <p>0 = Output logic Low to PWMx_CH1.</p> <p>1 = Output logic High to PWMx_CH1.</p>
[0]	<b>MSKDAT0</b>	<p><b>PWMx_CH0 Output Mask Data Control Bit</b></p> <p>This bit is used to control the output state of PWMx_CH0 pin when PWMx_CH0 output mask function is enabled (MSKEN0 = 1).</p> <p>0 = Output logic Low to PWMx_CH0.</p> <p>1 = Output logic High to PWMx_CH0.</p>

**Timer PWM Brake Pin Noise Filter Register (TIMERx PWMBNF)**

Register	Offset	R/W	Description	Reset Value
TIMER0_PWM_BNF	TMR01_BA+0x68	R/W	Timer0 PWM Brake Pin Noise Filter Register	0x0000_0000
TIMER1_PWM_BNF	TMR01_BA+0x168	R/W	Timer1 PWM Brake Pin Noise Filter Register	0x0000_0000
TIMER2_PWM_BNF	TMR23_BA+0x68	R/W	Timer2 PWM Brake Pin Noise Filter Register	0x0000_0000
TIMER3_PWM_BNF	TMR23_BA+0x168	R/W	Timer3 PWM Brake Pin Noise Filter Register	0x0000_0000

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

Bits	Description	
[31:18]	Reserved	Reserved.
[17:16]	BKPINSRC	<b>Brake Pin Source Select</b> 00 = Brake pin source comes from TM_BRAKE0. 01 = Brake pin source comes from TM_BRAKE1. 10 = Brake pin source comes from TM_BRAKE2. 11 = Brake pin source comes from TM_BRAKE3.
[15:8]	Reserved	Reserved.
[7]	BRKPINV	<b>Brake Pin Detection Control Bit</b> 0 = Brake pin event will be detected if TM_BRAKE <sub>x</sub> pin status transfer from low to high in edge-detect, or pin status is high in level-detect. 1 = Brake pin event will be detected if TM_BRAKE <sub>x</sub> pin status transfer from high to low in edge-detect, or pin status is low in level-detect .
[6:4]	BRKFCNT	<b>Brake Pin Noise Filter Count</b> The fields is used to control the active noise filter sample time. Once noise filter sample time = (Period time of BRKDBCS) * BRKDBCNT.
[3:1]	BRKNFSEL	<b>Brake Pin Noise Filter Clock Selection</b> 000 = Noise filter clock is PCLK <sub>x</sub> . 001 = Noise filter clock is PCLK <sub>x</sub> /2. 010 = Noise filter clock is PCLK <sub>x</sub> /4. 011 = Noise filter clock is PCLK <sub>x</sub> /8. 100 = Noise filter clock is PCLK <sub>x</sub> /16.



		<p>101 = Noise filter clock is PCLKx/32.                  110 = Noise filter clock is PCLKx/64.                  111 = Noise filter clock is PCLKx/128.</p>
[0]	<b>BRKNFEN</b>	<p><b>Brake Pin Noise Filter Enable Bit</b>                  0 = Pin noise filter detect of TM_BRAKE<sub>x</sub> Disabled.                  1 = Pin noise filter detect of TM_BRAKE<sub>x</sub> Enabled.</p>

**Timer PWM System Fail Brake Control Register (TIMERx PWMFAILBRK)**

Register	Offset	R/W	Description	Reset Value
<b>TIMER0_PWMFAILBRK</b>	TMR01_BA+0x6C	R/W	Timer0 PWM System Fail Brake Control Register	0x0000_0000
<b>TIMER1_PWMFAILBRK</b>	TMR01_BA+0x16C	R/W	Timer1 PWM System Fail Brake Control Register	0x0000_0000
<b>TIMER2_PWMFAILBRK</b>	TMR23_BA+0x6C	R/W	Timer2 PWM System Fail Brake Control Register	0x0000_0000
<b>TIMER3_PWMFAILBRK</b>	TMR23_BA+0x16C	R/W	Timer3 PWM System Fail Brake 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				CORBRKEN	Reserved	BODBRKEN	CSSBRKEN

Bits	Description
[31:4]	<b>Reserved</b> Reserved.
[3]	<b>CORBRKEN</b> <b>Core Lockup Detection Trigger PWM Brake Function Enable Bit</b> 0 = Brake Function triggered by core lockup event Disabled. 1 = Brake Function triggered by core lockup event Enabled.
[2]	<b>Reserved</b> Reserved.
[1]	<b>BODBRKEN</b> <b>Brown-out Detection Trigger PWM Brake Function Enable Bit</b> 0 = Brake Function triggered by BOD event Disabled. 1 = Brake Function triggered by BOD event Enabled.
[0]	<b>CSSBRKEN</b> <b>Clock Security System Detection Trigger PWM Brake Function Enable Bit</b> 0 = Brake Function triggered by clock fail detection Disabled. 1 = Brake Function triggered by clock fail detection Enabled.

**Timer PWM Brake Control Register (TIMERx PWMBRKCTL)**

Register	Offset	R/W	Description	Reset Value
TIMER0_PWMBRKCTL	TMR01_BA+0x70	R/W	Timer0 PWM Brake Control Register	0x0000_0000
TIMER1_PWMBRKCTL	TMR01_BA+0x170	R/W	Timer1 PWM Brake Control Register	0x0000_0000
TIMER2_PWMBRKCTL	TMR23_BA+0x70	R/W	Timer2 PWM Brake Control Register	0x0000_0000
TIMER3_PWMBRKCTL	TMR23_BA+0x170	R/W	Timer3 PWM Brake Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved				BRKAODD		BRKAEVEN	
15	14	13	12	11	10	9	8
SYSLBEN	Reserved		BRKPLEN	Reserved		CPO1LBEN	CPO0LBEN
7	6	5	4	3	2	1	0
SYSEBEN	Reserved		BRKPEEN	Reserved		CPO1EBEN	CPO0EBEN

Bits	Description	
[31:20]	Reserved	Reserved.
[19:18]	BRKAODD	<p><b>PWM Brake Action Select for PWMx_CH1 (Write Protect)</b></p> <p>00 = TIMERx_PWM brake event will not affect PWMx_CH1 output.                      01 = PWMx_CH1 output tri-state when TIMERx_PWM brake event happened.                      10 = PWMx_CH1 output low level when TIMERx_PWM brake event happened.                      11 = PWMx_CH1 output high level when TIMERx_PWM brake event happened.</p> <p><b>Note:</b> This register is write protected. Refer to SYS_REGLCTL register.</p>
[17:16]	BRKAEVEN	<p><b>PWM Brake Action Select for PWMx_CH0 (Write Protect)</b></p> <p>00 = TIMERx_PWM brake event will not affect PWMx_CH0 output.                      01 = PWMx_CH0 output tri-state when TIMERx_PWM brake event happened.                      10 = PWMx_CH0 output low level when TIMERx_PWM brake event happened.                      11 = PWMx_CH0 output high level when TIMERx_PWM brake event happened.</p> <p><b>Note:</b> This register is write protected. Refer to SYS_REGLCTL register.</p>
[15]	SYSLBEN	<p><b>Enable System Fail As Level-detect Brake Source (Write Protect)</b></p> <p>0 = System fail condition as level-detect brake source Disabled.                      1 = System fail condition as level-detect brake source Enabled.</p> <p><b>Note:</b> This register is write protected. Refer to SYS_REGLCTL register.</p>
[14:13]	Reserved	Reserved.
[12]	BRKPLEN	<p><b>Enable TM_BRAKEx Pin As Level-detect Brake Source (Write Protect)</b></p> <p>0 = TM_BRAKEx pin event as level-detect brake source Disabled.</p>

		1 = TM_BRAKE <sub>x</sub> pin event as level-detect brake source Enabled. <b>Note:</b> This register is write protected. Refer to SYS_REGLCTL register.
[11:10]	Reserved	Reserved.
[9]	CPO1LBEN	<b>Enable Internal ACMP1_O Digital Output As Level-detect Brake Source (Write Protect)</b> 0 = Internal ACMP1_O signal as level-detect brake source Disabled. 1 = Internal ACMP1_O signal as level-detect brake source Enabled. <b>Note1:</b> Only internal ACMP1_O signal from low to high will be detected as brake event. <b>Note:</b> This register is write protected. Refer to SYS_REGLCTL register.
[8]	CPO0LBEN	<b>Enable Internal ACMP0_O Digital Output As Level-detect Brake Source (Write Protect)</b> 0 = Internal ACMP0_O signal as level-detect brake source Disabled. 1 = Internal ACMP0_O signal as level-detect brake source Enabled. <b>Note1:</b> Only internal ACMP0_O signal from low to high will be detected as brake event. <b>Note2:</b> This register is write protected. Refer to SYS_REGLCTL register.
[7]	SYSEBEN	<b>Enable System Fail As Edge-detect Brake Source (Write Protect)</b> 0 = System fail condition as edge-detect brake source Disabled. 1 = System fail condition as edge-detect brake source Enabled. <b>Note:</b> This register is write protected. Refer to SYS_REGLCTL register.
[6:5]	Reserved	Reserved.
[4]	BRKPEEN	<b>Enable TM_BRAKE<sub>x</sub> Pin As Edge-detect Brake Source (Write Protect)</b> 0 = TM_BRAKE <sub>x</sub> pin event as edge-detect brake source Disabled. 1 = TM_BRAKE <sub>x</sub> pin event as edge-detect brake source Enabled. <b>Note:</b> This register is write protected. Refer to SYS_REGLCTL register.
[3:2]	Reserved	Reserved.
[1]	CPO1EBEN	<b>Enable Internal ACMP1_O Digital Output As Edge-detect Brake Source (Write Protect)</b> 0 = Internal ACMP1_O signal as edge-detect brake source Disabled. 1 = Internal ACMP1_O signal as edge-detect brake source Enabled. <b>Note1:</b> Only internal ACMP1_O signal from low to high will be detected as brake event. <b>Note2:</b> This register is write protected. Refer to SYS_REGLCTL register.
[0]	CPO0EBEN	<b>Enable Internal ACMP0_O Digital Output As Edge-detect Brake Source (Write Protect)</b> 0 = Internal ACMP0_O signal as edge-detect brake source Disabled. 1 = Internal ACMP0_O signal as edge-detect brake source Enabled. <b>Note1:</b> Only internal ACMP0_O signal from low to high will be detected as brake event. <b>Note2:</b> This register is write protected. Refer to SYS_REGLCTL register.

**Timer PWM Pin Output Polar Control Register (TIMERx PWMPOLCTL)**

Register	Offset	R/W	Description	Reset Value
TIMER0_PWM POLCTL	TMR01_BA+0x74	R/W	Timer0 PWM Pin Output Polar Control Register	0x0000_0000
TIMER1_PWM POLCTL	TMR01_BA+0x174	R/W	Timer1 PWM Pin Output Polar Control Register	0x0000_0000
TIMER2_PWM POLCTL	TMR23_BA+0x74	R/W	Timer2 PWM Pin Output Polar Control Register	0x0000_0000
TIMER3_PWM POLCTL	TMR23_BA+0x174	R/W	Timer3 PWM Pin Output Polar 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						PINV1	PINV0

Bits	Description	
[31:2]	Reserved	Reserved.
[1]	PINV1	<b>PWMx_CH1 Output Pin Polar Control Bit</b> The bit is used to control polarity state of PWMx_CH1 output pin. 0 = PWMx_CH1 output pin polar inverse Disabled. 1 = PWMx_CH1 output pin polar inverse Enabled.
[0]	PINV0	<b>PWMx_CH0 Output Pin Polar Control Bit</b> The bit is used to control polarity state of PWMx_CH0 output pin. 0 = PWMx_CH0 output pin polar inverse Disabled. 1 = PWMx_CH0 output pin polar inverse Enabled.

**Timer PWM Pin Output Enable Register (TIMERx PWMPOEN)**

Register	Offset	R/W	Description	Reset Value
TIMER0_PWMPOEN	TMR01_BA+0x78	R/W	Timer0 PWM Pin Output Enable Register	0x0000_0000
TIMER1_PWMPOEN	TMR01_BA+0x178	R/W	Timer1 PWM Pin Output Enable Register	0x0000_0000
TIMER2_PWMPOEN	TMR23_BA+0x78	R/W	Timer2 PWM Pin Output Enable Register	0x0000_0000
TIMER3_PWMPOEN	TMR23_BA+0x178	R/W	Timer3 PWM Pin 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						POEN1	POEN0

Bits	Description	
[31:2]	Reserved	Reserved.
[1]	POEN1	<b>PWMx_CH1 Output Pin Enable Bit</b> 0 = PWMx_CH1 pin at tri-state mode. 1 = PWMx_CH1 pin in output mode.
[0]	POEN0	<b>PWMx_CH0 Output Pin Enable Bit</b> 0 = PWMx_CH0 pin at tri-state mode. 1 = PWMx_CH0 pin in output mode.

**Timer PWM Software Trigger Brake Control Register (TIMERx PWMSWBRK)**

Register	Offset	R/W	Description	Reset Value
TIMER0_PWM SWBRK	TMR01_BA+0x7C	W	Timer0 PWM Software Trigger Brake Control Register	0x0000_0000
TIMER1_PWM SWBRK	TMR01_BA+0x17C	W	Timer1 PWM Software Trigger Brake Control Register	0x0000_0000
TIMER2_PWM SWBRK	TMR23_BA+0x7C	W	Timer2 PWM Software Trigger Brake Control Register	0x0000_0000
TIMER3_PWM SWBRK	TMR23_BA+0x17C	W	Timer3 PWM Software Trigger Brake 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							BRKLTRG
7	6	5	4	3	2	1	0
Reserved							BRKETRG

Bits	Description	
[31:9]	Reserved	Reserved.
[8]	BRKLTRG	<p><b>Software Trigger Level-detect Brake Source (Write Only) (Write Protect)</b></p> <p>Write 1 to this bit will trigger PWM level-detect brake source, then BRKLIF0 and BRKLIF1 will set to 1 automatically in TIMERx_PWMINTSTS1 register.</p> <p><b>Note:</b> This register is write protected. Refer to SYS_REGLCTL register.</p>
[7:1]	Reserved	Reserved.
[0]	BRKETRG	<p><b>Software Trigger Edge-detect Brake Source (Write Only) (Write Protect)</b></p> <p>Write 1 to this bit will trigger PWM edge-detect brake source, then BRKEIF0 and BRKEIF1 will set to 1 automatically in TIMERx_PWMINTSTS1 register.</p> <p><b>Note:</b> This register is write protected. Refer to SYS_REGLCTL register.</p>

**Timer PWM Interrupt Enable Register 0 (TIMERx PWMINTEN0)**

Register	Offset	R/W	Description	Reset Value
TIMER0_PWMINTEN0	TMR01_BA+0x80	R/W	Timer0 PWM Interrupt Enable Register 0	0x0000_0000
TIMER1_PWMINTEN0	TMR01_BA+0x180	R/W	Timer1 PWM Interrupt Enable Register 0	0x0000_0000
TIMER2_PWMINTEN0	TMR23_BA+0x80	R/W	Timer2 PWM Interrupt Enable Register 0	0x0000_0000
TIMER3_PWMINTEN0	TMR23_BA+0x180	R/W	Timer3 PWM Interrupt Enable 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				CMPDIEN	CMPUIEN	PIEN	ZIEN

Bits	Description	
[31:4]	Reserved	Reserved.
[3]	CMPDIEN	<b>PWM Compare Down Count Interrupt Enable Bit</b> 0 = Compare down count interrupt Disabled. 1 = Compare down count interrupt Enabled.
[2]	CMPUIEN	<b>PWM Compare Up Count Interrupt Enable Bit</b> 0 = Compare up count interrupt Disabled. 1 = Compare up count interrupt Enabled.
[1]	PIEN	<b>PWM Period Point Interrupt Enable Bit</b> 0 = Period point interrupt Disabled. 1 = Period point interrupt Enabled. <b>Note:</b> In up-down count type, period point means the center point of current PWM period.
[0]	ZIEN	<b>PWM Zero Point Interrupt Enable Bit</b> 0 = Zero point interrupt Disabled. 1 = Zero point interrupt Enabled.



**Timer PWM Interrupt Enable Register 1 (TIMERx PWMINTEN1)**

Register	Offset	R/W	Description	Reset Value
TIMER0_PWMINTEN1	TMR01_BA+0x84	R/W	Timer0 PWM Interrupt Enable Register 1	0x0000_0000
TIMER1_PWMINTEN1	TMR01_BA+0x184	R/W	Timer1 PWM Interrupt Enable Register 1	0x0000_0000
TIMER2_PWMINTEN1	TMR23_BA+0x84	R/W	Timer2 PWM Interrupt Enable Register 1	0x0000_0000
TIMER3_PWMINTEN1	TMR23_BA+0x184	R/W	Timer3 PWM Interrupt Enable 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							BRKLIEN
7	6	5	4	3	2	1	0
Reserved							BRKEIEN

Bits	Description	
[31:9]	Reserved	Reserved.
[8]	BRKLIEN	<b>PWM Level-detect Brake Interrupt Enable Bit (Write Protect)</b> 0 = PWM level-detect brake interrupt Disabled. 1 = PWM level-detect brake interrupt Enabled. <b>Note:</b> This bit is write protected. Refer to SYS_REGLCTL register.
[7:1]	Reserved	Reserved.
[0]	BRKEIEN	<b>PWM Edge-detect Brake Interrupt Enable Bit (Write Protect)</b> 0 = PWM edge-detect brake interrupt Disabled. 1 = PWM edge-detect brake interrupt Enabled. <b>Note:</b> This bit is write protected. Refer to SYS_REGLCTL register.

**Timer PWM Interrupt Status Register 0 (TIMERx PWMINTSTS0)**

Register	Offset	R/W	Description	Reset Value
TIMER0_PWMINTSTS0	TMR01_BA+0x88	R/W	Timer0 PWM Interrupt Status Register 0	0x0000_0000
TIMER1_PWMINTSTS0	TMR01_BA+0x188	R/W	Timer1 PWM Interrupt Status Register 0	0x0000_0000
TIMER2_PWMINTSTS0	TMR23_BA+0x88	R/W	Timer2 PWM Interrupt Status Register 0	0x0000_0000
TIMER3_PWMINTSTS0	TMR23_BA+0x188	R/W	Timer3 PWM Interrupt Status 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				CMPDIF	CMPUIF	PIF	ZIF

Bits	Description
[31:4]	<b>Reserved</b> Reserved.
[3]	<b>CMPDIF</b> <b>PWM Compare Down Count Interrupt Flag</b> This bit is set by hardware when TIMERx_PWM counter in down count direction and reaches CMP. <b>Note1:</b> If CMP equal to PERIOD, there is no CMPDIF flag in down count type. <b>Note2:</b> This bit is cleared by writing 1 to it.
[2]	<b>CMPUIF</b> <b>PWM Compare Up Count Interrupt Flag</b> This bit is set by hardware when TIMERx_PWM counter in up count direction and reaches CMP. <b>Note1:</b> If CMP equal to PERIOD, there is no CMPUIF flag in up count type and up-down count type.. <b>Note2:</b> This bit is cleared by writing 1 to it.
[1]	<b>PIF</b> <b>PWM Period Point Interrupt Flag</b> This bit is set by hardware when TIMERx_PWM counter reaches PERIOD. <b>Note1:</b> When in up-down count type, PIF flag means the center point flag of current PWM period. <b>Note2:</b> This bit is cleared by writing 1 to it.
[0]	<b>ZIF</b> <b>PWM Zero Point Interrupt Flag</b> This bit is set by hardware when TIMERx_PWM counter reaches zero. <b>Note:</b> This bit is cleared by writing 1 to it.

**Timer PWM Interrupt Status Register 1 (TIMERx PWMINTSTS1)**

Register	Offset	R/W	Description	Reset Value
TIMER0_PWMINTSTS1	TMR01_BA+0x8C	R/W	Timer0 PWM Interrupt Status Register 1	0x0000_0000
TIMER1_PWMINTSTS1	TMR01_BA+0x18C	R/W	Timer1 PWM Interrupt Status Register 1	0x0000_0000
TIMER2_PWMINTSTS1	TMR23_BA+0x8C	R/W	Timer2 PWM Interrupt Status Register 1	0x0000_0000
TIMER3_PWMINTSTS1	TMR23_BA+0x18C	R/W	Timer3 PWM Interrupt Status Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						BRKLSTS1	BRKLSTS0
23	22	21	20	19	18	17	16
Reserved						BRKESTS1	BRKESTS0
15	14	13	12	11	10	9	8
Reserved						BRKLIF1	BRKLIF0
7	6	5	4	3	2	1	0
Reserved						BRKEIF1	BRKEIF0

Bits	Description	
[31:26]	Reserved	Reserved.
[25]	BRKLSTS1	<p><b>Level-detect Brake Status of PWMx_CH1 (Read Only)</b></p> <p>0 = PWMx_CH1 level-detect brake state is released. 1 = PWMx_CH1 at level-detect brake state.</p> <p><b>Note:</b> If TIMERx_PWM level-detect brake source has released, both PWMx_CH0 and PWMx_CH1 will release brake state when current PWM period finished and resume PWMx_CH0 and PWMx_CH1 output waveform start from next full PWM period.</p>
[24]	BRKLSTS0	<p><b>Level-detect Brake Status of PWMx_CH0 (Read Only)</b></p> <p>0 = PWMx_CH0 level-detect brake state is released. 1 = PWMx_CH0 at level-detect brake state.</p> <p><b>Note:</b> If TIMERx_PWM level-detect brake source has released, both PWMx_CH0 and PWMx_CH1 will release brake state when current PWM period finished and resume PWMx_CH0 and PWMx_CH1 output waveform start from next full PWM period.</p>
[23:18]	Reserved	Reserved.
[17]	BRKESTS1	<p><b>Edge-detect Brake Status of PWMx_CH1 (Read Only)</b></p> <p>0 = PWMx_CH1 edge-detect brake state is released. 1 = PWMx_CH1 at edge-detect brake state.</p> <p><b>Note:</b> User can set BRKEIF1 1 to clear BRKEIF1 flag and PWMx_CH1 will release brake state when current PWM period finished and resume PWMx_CH1 output waveform start from next full PWM period.</p>
[16]	BRKESTS0	<p><b>Edge -detect Brake Status of PWMx_CH0 (Read Only)</b></p> <p>0 = PWMx_CH0 edge-detect brake state is released.</p>

		1 = PWMx_CH0 at edge-detect brake state. <b>Note:</b> User can set BRKEIF0 1 to clear BRKEIF0 flag and PWMx_CH0 will release brake state when current PWM period finished and resume PWMx_CH0 output waveform start from next full PWM period.
[15:10]	<b>Reserved</b>	Reserved.
[9]	<b>BRKLIF1</b>	<b>Level-detect Brake Interrupt Flag on PWMx_CH1 (Write Protect)</b> 0 = PWMx_CH1 level-detect brake event did not happen. 1 = PWMx_CH1 level-detect brake event happened. <b>Note1:</b> This bit is cleared by writing 1 to it. <b>Note2:</b> This register is write protected. Refer to SYS_REGLCTL register.
[8]	<b>BRKLIF0</b>	<b>Level-detect Brake Interrupt Flag on PWMx_CH0 (Write Protect)</b> 0 = PWMx_CH0 level-detect brake event did not happen. 1 = PWMx_CH0 level-detect brake event happened. <b>Note1:</b> This bit is cleared by writing 1 to it. <b>Note2:</b> This register is write protected. Refer to SYS_REGLCTL register.
[7:2]	<b>Reserved</b>	Reserved.
[1]	<b>BRKEIF1</b>	<b>Edge-detect Brake Interrupt Flag PWMx_CH1 (Write Protect)</b> 0 = PWMx_CH1 edge-detect brake event did not happen. 1 = PWMx_CH1 edge-detect brake event happened. <b>Note1:</b> This bit is cleared by writing 1 to it. <b>Note2:</b> This register is write protected. Refer to SYS_REGLCTL register.
[0]	<b>BRKEIF0</b>	<b>Edge-detect Brake Interrupt Flag on PWMx_CH0 (Write Protect)</b> 0 = PWMx_CH0 edge-detect brake event did not happen. 1 = PWMx_CH0 edge-detect brake event happened. <b>Note1:</b> This bit is cleared by writing 1 to it. <b>Note2:</b> This register is write protected. Refer to SYS_REGLCTL register.

**Timer PWM ADC Trigger Control Register (TIMERx PWMADCTS)**

Register	Offset	R/W	Description	Reset Value
TIMER0_PWMADCTS	TMR01_BA+0x90	R/W	Timer0 PWM ADC Trigger Source Select Register	0x0000_0000
TIMER1_PWMADCTS	TMR01_BA+0x190	R/W	Timer1 PWM ADC Trigger Source Select Register	0x0000_0000
TIMER2_PWMADCTS	TMR23_BA+0x90	R/W	Timer2 PWM ADC Trigger Source Select Register	0x0000_0000
TIMER3_PWMADCTS	TMR23_BA+0x190	R/W	Timer3 PWM ADC Trigger Source 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
TRGEN	Reserved				TRGSEL		

Bits	Description	
[31:8]	Reserved	Reserved.
[7]	TRGEN	<b>PWM Counter Event Trigger ADC Conversion Enable Bit</b> 0 = PWM counter event trigger ADC conversion Disabled. 1 = PWM counter event trigger ADC conversion Enabled.
[6:3]	Reserved	Reserved.
[2:0]	TRGSEL	<b>PWM Counter Event Source Select to Trigger ADC Conversion</b> 000 = Trigger ADC conversion at zero point (ZIF). 001 = Trigger ADC conversion at period point (PIF). 010 = Trigger ADC conversion at zero or period point (ZIF or PIF). 011 = Trigger ADC conversion at compare up count point (CMPUIF). 100 = Trigger ADC conversion at compare down count point (CMPDIF). Others = Reserved.

**Timer PWM Synchronous Control Register (TIMERx PWMSCTL)**

Register	Offset	R/W	Description	Reset Value
TIMER0_PWMSCTL	TMR01_BA+0x94	R/W	Timer0 PWM Synchronous Control Register	0x0000_0000
TIMER1_PWMSCTL	TMR01_BA+0x194	R/W	Timer1 PWM Synchronous Control Register	0x0000_0000
TIMER2_PWMSCTL	TMR23_BA+0x94	R/W	Timer2 PWM Synchronous Control Register	0x0000_0000
TIMER3_PWMSCTL	TMR23_BA+0x194	R/W	Timer3 PWM Synchronous 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							SYNCSRC
7	6	5	4	3	2	1	0
Reserved						SYNCSRC	

Bits	Description	
[31:9]	Reserved	Reserved.
[8]	SYNCSRC	<p><b>PWM Synchronous Counter Start/Clear Source Select</b></p> <p>0 = Counter synchronous start/clear by trigger TIMER0_PWMSTRG STRGEN.                      1 = Counter synchronous start/clear by trigger TIMER2_PWMSTRG STRGEN.</p> <p><b>Note1:</b> If TIMER0/1/2/3 PWM counter synchronous source are from TIMER0, TIME0_PWMSCTL[8], TIME1_PWMSCTL[8], TIME2_PWMSCTL[8] and TIME3_PWMSCTL[8] should be 0.</p> <p><b>Note2:</b> If TIMER0/1/ PWM counter synchronous source are from TIMER0, TIME0_PWMSCTL[8] and TIME1_PWMSCTL[8] should be set 0, and TIMER2/3/ PWM counter synchronous source are from TIMER2, TIME2_PWMSCTL[8] and TIME3_PWMSCTL[8] should be set 1.</p>
[7:2]	Reserved	Reserved.
[1:0]	SYNCSRC	<p><b>PWM Synchronous Mode Enable Select</b></p> <p>00 = PWM synchronous function Disabled.                      01 = PWM synchronous counter start function Enabled.                      10 = Reserved.                      11 = PWM synchronous counter clear function Enabled.</p>

**Timer PWM Synchronous Trigger Register (TIMERx PWMSTRG)**

Register	Offset	R/W	Description	Reset Value
TIMER0_PWMSTRG	TMR01_BA+0x98	W	Timer0 PWM Synchronous Trigger Register	0x0000_0000
TIMER2_PWMSTRG	TMR23_BA+0x98	W	Timer2 PWM Synchronous Trigger 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							STRGEN

Bits	Description	
[31:1]	Reserved	Reserved.
[0]	STRGEN	<p><b>PWM Counter Synchronous Trigger Enable Bit (Write Only)</b></p> <p>PMW counter synchronous function is used to make selected PWM channels (include TIMER0/1/2/3 PWM, TIMER0/1 PWM and TIMER2/3 PWM) start counting or clear counter at the same time according to TIMERx_PWMCTL setting.</p> <p><b>Note:</b> This bit is only available in TIMER0 and TIMER2.</p>

**Timer PWM Status Register (TIMERx PWMSTATUS)**

Register	Offset	R/W	Description	Reset Value
<b>TIMER0_PWM STATUS</b>	TMR01_BA+0x9C	R/W	Timer0 PWM Status Register	0x0000_0000
<b>TIMER1_PWM STATUS</b>	TMR01_BA+0x19C	R/W	Timer1 PWM Status Register	0x0000_0000
<b>TIMER2_PWM STATUS</b>	TMR23_BA+0x9C	R/W	Timer2 PWM Status Register	0x0000_0000
<b>TIMER3_PWM STATUS</b>	TMR23_BA+0x19C	R/W	Timer3 PWM Status Register	0x0000_0000

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

Bits	Description
[31:17]	<b>Reserved</b> Reserved.
[16]	<b>ADCTRGF</b> <b>Trigger ADC Start Conversion Flag</b> 0 = PWM counter event trigger ADC start conversion has not occurred. 1 = PWM counter event trigger ADC start conversion has occurred. <b>Note:</b> This bit is cleared by writing 1 to it.
[15:1]	<b>Reserved</b> Reserved.
[0]	<b>CNTMAXF</b> <b>PWM Counter Equal to 0xFFFF Flag</b> 0 = PWM counter value never reached its maximum value 0xFFFF. 1 = PWM counter value has reached its maximum value. <b>Note:</b> This bit is cleared by writing 1 to it.



**Timer PWM Period Buffer Register (TIMERx PWMPBUF)**

Register	Offset	R/W	Description	Reset Value
TIMER0_PWM PBUF	TMR01_BA+0xA0	R	Timer0 PWM Period Buffer Register	0x0000_0000
TIMER1_PWM PBUF	TMR01_BA+0x1A0	R	Timer1 PWM Period Buffer Register	0x0000_0000
TIMER2_PWM PBUF	TMR23_BA+0xA0	R	Timer2 PWM Period Buffer Register	0x0000_0000
TIMER3_PWM PBUF	TMR23_BA+0x1A0	R	Timer3 PWM Period Buffer 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
PBUF							
7	6	5	4	3	2	1	0
PBUF							

Bits	Description	
[31:16]	Reserved	Reserved.
[15:0]	PBUF	<b>PWM Period Buffer Register (Read Only)</b> Used as PERIOD active register.

**Timer PWM Comparator Buffer Register (TIMERx PWMCMPBUF)**

Register	Offset	R/W	Description	Reset Value
TIMER0_PWM_CMPBUF	TMR01_BA+0xA4	R	Timer0 PWM Comparator Buffer Register	0x0000_0000
TIMER1_PWM_CMPBUF	TMR01_BA+0x1A4	R	Timer1 PWM Comparator Buffer Register	0x0000_0000
TIMER2_PWM_CMPBUF	TMR23_BA+0xA4	R	Timer2 PWM Comparator Buffer Register	0x0000_0000
TIMER3_PWM_CMPBUF	TMR23_BA+0x1A4	R	Timer3 PWM Comparator Buffer 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
CMPBUF							
7	6	5	4	3	2	1	0
CMPBUF							

Bits	Description
[31:16]	<b>Reserved</b> Reserved.
[15:0]	<b>CMPBUF</b> <b>PWM Comparator Buffer Register (Read Only)</b> Used as CMP active register.

## 6.18 USB Device Controller (USB\_D)

### 6.18.1 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. It implements a full-speed (12 Mbit/s) function interface with added support for USB 2.0 Link Power Management.

In this device controller, there are two main interfaces: the 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 512 bytes 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 (USB\_D\_BUFSEGx).

There are 8 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 four different interrupt events in this controller. They are the wake-up idle event, device plug-in or plug-out event, USB events, like IN ACK, OUT ACK etc, and BUS events, like 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 (USB\_D\_INTSTS) to acknowledge what kind of interrupt occurring, and then check the related USB Endpoint Status Register (USB\_D\_EPSTS) to acknowledge what kind of event occurring in this endpoint.

A software-disconnect function is also supported for this USB controller. It is used to simulate the disconnection of this device from the host. If user enables SE0 bit (USB\_D\_SE0), the USB controller will force the output of USB\_D+ and USB\_D- to level low. It will casue host detect disconnect after user enable SE0 bit for a while. Finally, user can disable the SE0 bit, 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*.

### 6.18.2 Features

- Compliant with USB 2.0 Full-Speed specification
- Provides 1 interrupt vector with 4 different interrupt events (WKIDLE, VBUSDET, USB and BUS)
- Supports Control/Bulk/Interrupt/Isochronous transfer type
- Supports suspend function when no bus activity existing for 3 ms
- Supports 8 endpoints for configurable Control/Bulk/Interrupt/Isochronous transfer types and maximum 512 bytes buffer size
- Provides remote wake-up capability
- Supports Start of Frame (SOF) interrupt and USB frame number monitor.
- Supports USB 2.0 Link Power Management

### 6.18.3 Block Diagram

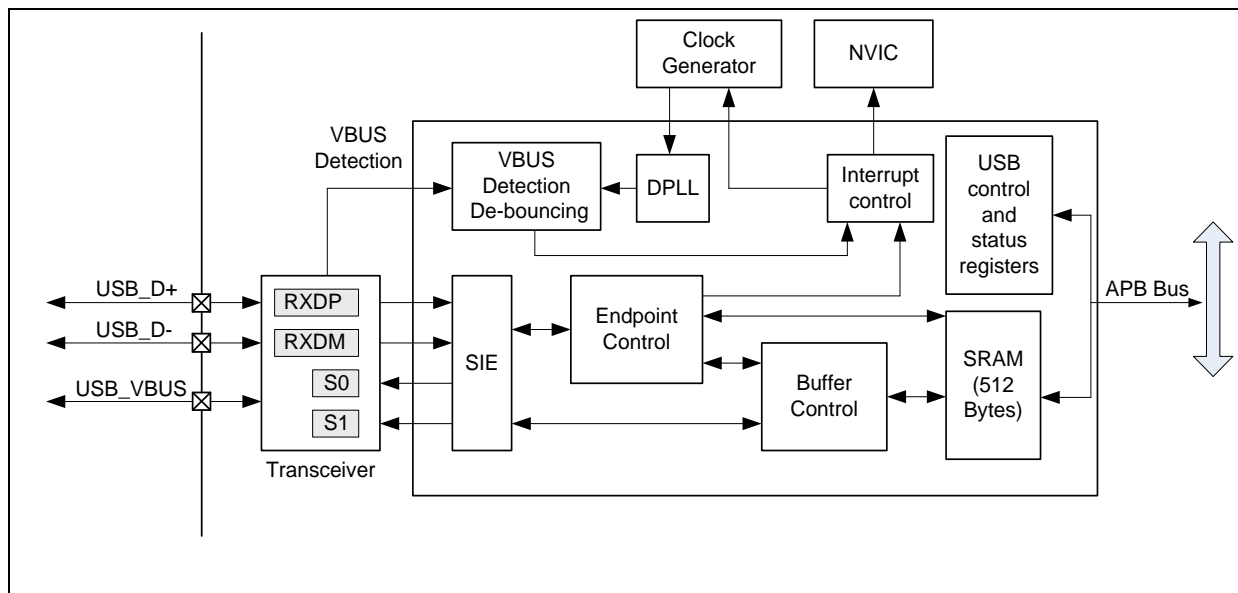


Figure 6.18-1 USB Block Diagram

### 6.18.4 Basic Configuration

User has to set the PLL related configurations before USB device controller is enabled. Set the USBCKEN (CLK\_APBCLK0[27]) bit to enable USB D clock and 4-bit pre-scaler USBDIV (CLK\_CLKDIV0[7:4]) to generate the proper USB D clock rate.

#### 6.18.4.1 Basic Configuration of USB Device 1.1

- Clock Source Configuration
  - Set the PLL controller (CLK\_PLLCTL).
  - Select the clock divider number of USB D peripheral clock on USBDIV(CLK\_CLKDIV0[7:4])
  - Enable USB D peripheral clock in USBCKEN (CLK\_APBCLK0[27]).
- Reset Configuration
  - Reset USB D controller in USBDRST (SYS\_IPRST1[27]).

### 6.18.5 Functional Description

#### 6.18.5.1 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
- SOF, 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

#### 6.18.5.2 Endpoint Control

This controller supports 8 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.

#### 6.18.5.3 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.

#### 6.18.5.4 VBUS 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 VBUS detection interrupt to avoid bounce problems on USB plug-in or unplug. VBUS detection interrupt appears about 10 ms 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.

#### 6.18.5.5 Interrupt Control

The USB provides 1 interrupt vector with 4 interrupt events (WKIDLE, VBUSDET, USB and BUS). The WKIDLE event is used to detect unexpected wakeup. 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.

WKIDLE interrupt is only presented when no the other USB interrupt events happened more than 20ms after the chip is waked up from Power-down mode. After the chip enters Power-down mode, any change on USB\_VBUS, USB\_D+ and USB\_D- can wake-up this chip if USB wake-up function is enabled. If the system is wake-up by unexpected situation, e.g., noise on USB\_D+, the system will wake-up but cannot receive any USB event. In such situation, the system may consume more power than in Power-down mode. To avoid it, use wake-up idle event to re-enter Power-down mode. Figure 6.17-2 is the control flow of wake-up interrupt.

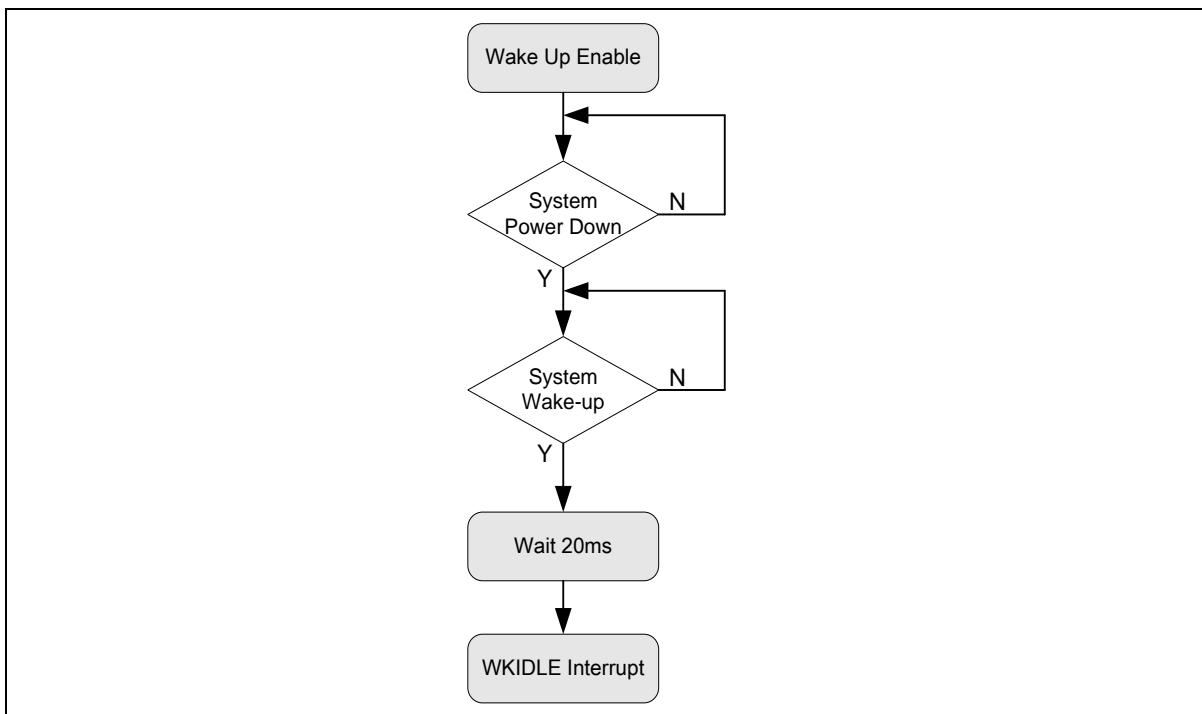


Figure 6.18-2 WKIDLE Interrupt Operation Flow

The USB interrupt is used to notify users of any USB event on the bus, and user can read EPSTS (USBD\_EPSTS[31:8]) and EPEVT7~0 (USBD\_INTSTS[23:16]) 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.

6.18.5.6 Power Saving

User can write 0 to USBD\_ATTR[4] to disable PHY under special circumstances, like suspend, to conserve power.

6.18.5.7 Buffer Control

There is 512 bytes SRAM in the controller and the 8 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\_MXPLDx register.

Figure 6.18-3 depicts the starting address for each endpoint according the content of USBD\_BUFSEGx and USBD\_MXPLDx registers. If the USBD\_BUFSEG0 is programmed as 0x08 and USBD\_MXPLD0 is set as 0x40, the SRAM size of endpoint 0 is start from USBD\_BA+0x108 and end in USBD\_BA+0x147. (Note: The USBD SRAM base is USBD\_BA+0x100).

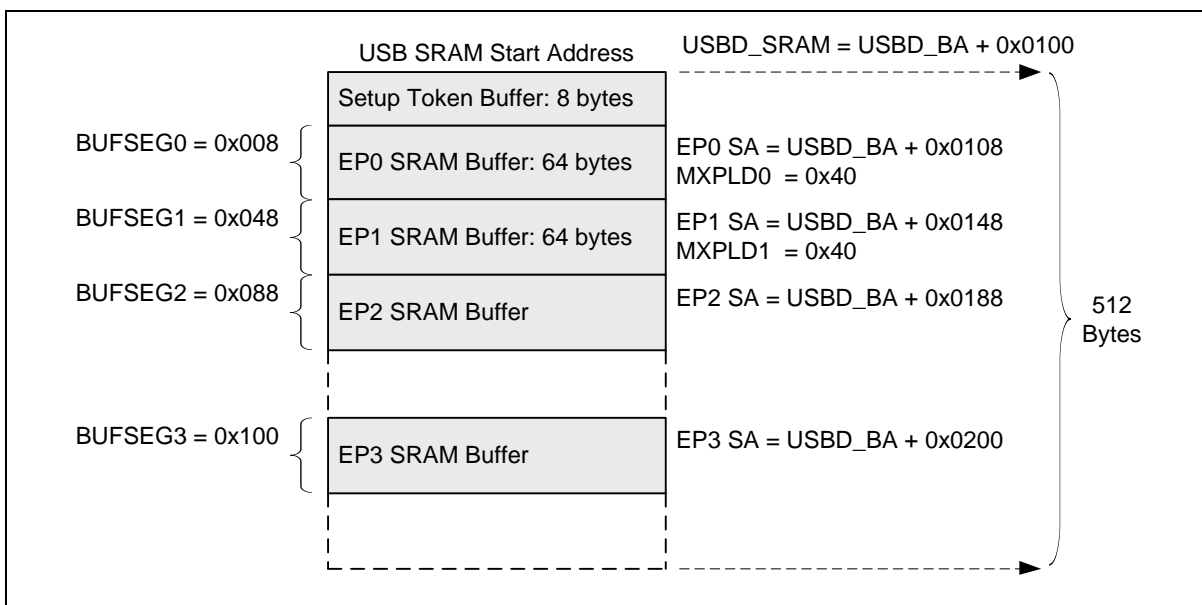


Figure 6.18-3 Endpoint SRAM Structure

6.18.5.8 Handling Transactions with USB Device Peripheral

The interrupt or polling USBD\_INTSTS can be used 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\_MXPLDx 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.

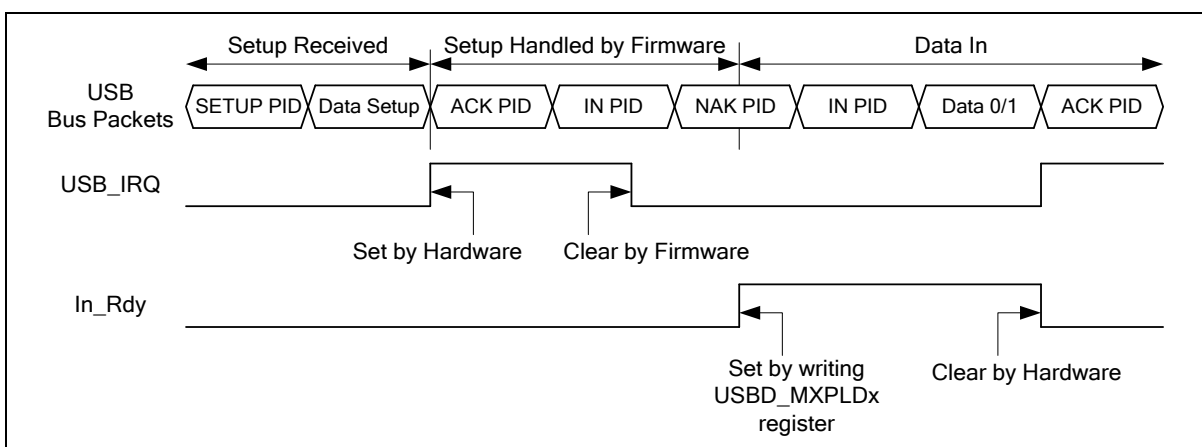


Figure 6.18-4 Setup Transaction Followed by Data IN Transaction

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\_MXPLDx 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

USB\_MXPLDx register needs to be written by firmware to assert the signal “Out\_Rdy” again to accept the next transaction.

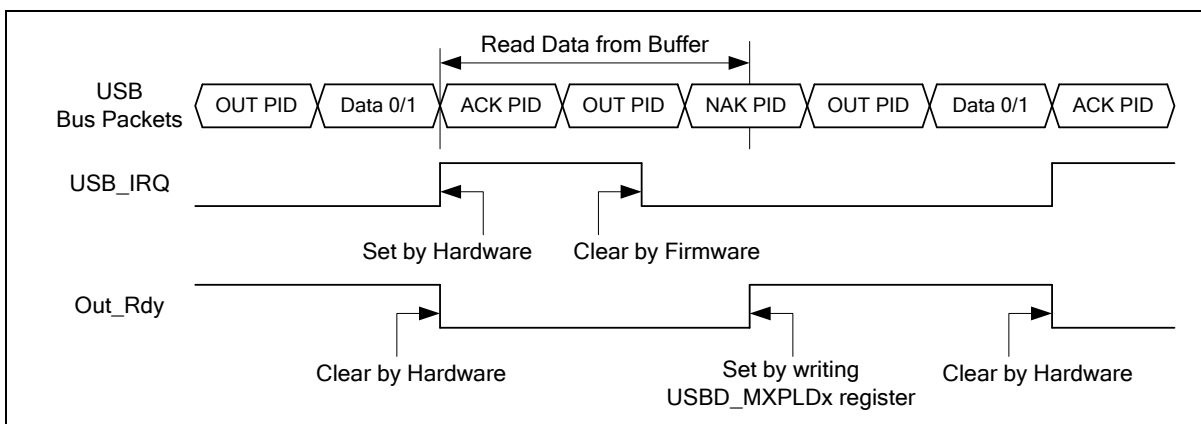


Figure 6.18-5 Data Out Transfer

### 6.18.5.9 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)

New fast mechanism for transitioning the bus on a root port from an enable state (called L0), to a new Sleep state(called L1), detail define for L0 and L1 state see table 6.18-1, the register USBD\_ATTR & USBD\_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 VBUS) 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 VBUS) 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 6.18-1 USB Link Power Manager (Lx) States

The state transaction process please refer to Figure 6.18-6, and for more information on the USB Link Power Manager(LPM), please refer to USB2.0 Link Power Mangement ECN.



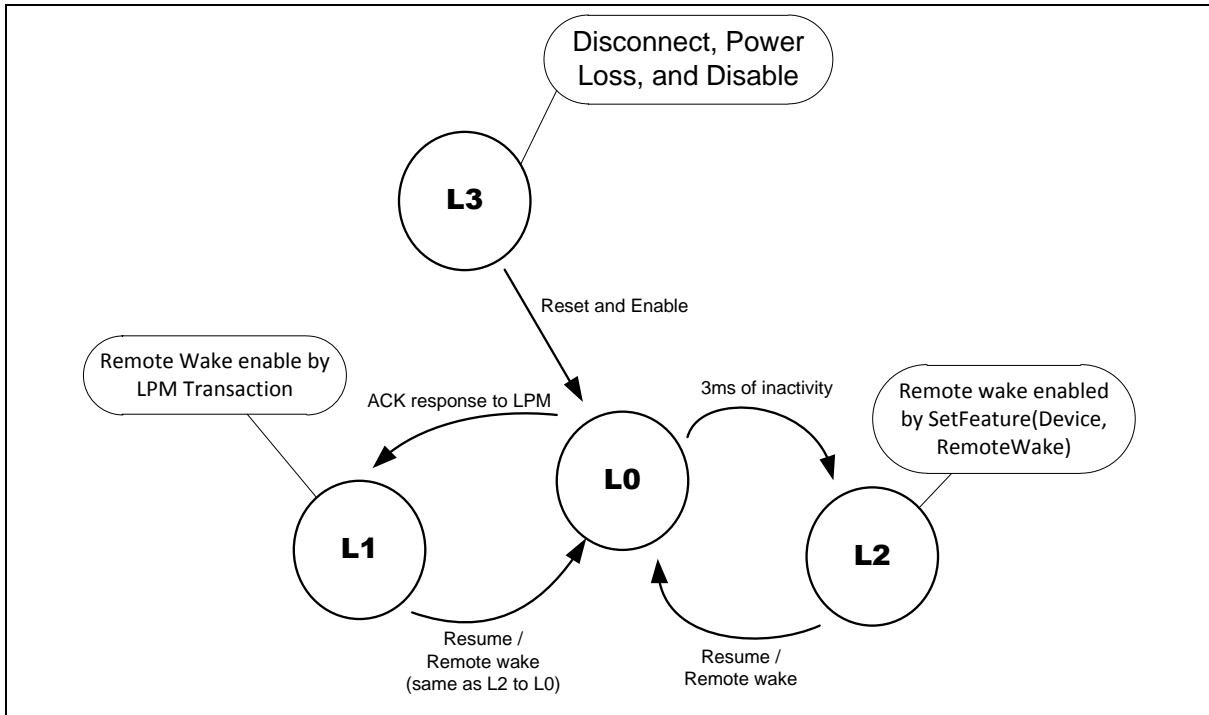


Figure 6.18-6 LPM State Transition Diagram

### 6.18.6 Register Map

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

Register	Offset	R/W	Description	Reset Value
<b>USB Base Address:</b>				
<b>USB_BA = 0x4006_0000</b>				
USB_D_INTEN	USB_BA+0x000	R/W	USB Device Interrupt Enable Register	0x0000_0000
USB_D_INTSTS	USB_BA+0x004	R/W	USB Device Interrupt Event Status Register	0x0000_0000
USB_D_FADDR	USB_BA+0x008	R/W	USB Device Function Address Register	0x0000_0000
USB_D_EPSTS	USB_BA+0x00C	R	USB Device Endpoint Status Register	0x0000_0000
USB_D_ATTR	USB_BA+0x010	R/W	USB Device Bus Status and Attribution Register	0x0000_0040
USB_D_VBUSDET	USB_BA+0x014	R	USB Device VBUS Detection Register	0x0000_0000
USB_D_STBUFSEG	USB_BA+0x018	R/W	Setup Token Buffer Segmentation Register	0x0000_0000
USB_D_LPMATTR	USB_BA+0x088	R	USB LPM Attribution Register	0x0000_0000
USB_D_FN	USB_BA+0x08C	R	USB Frame number Register	0x0000_0XXX
USB_D_SE0	USB_BA+0x090	R/W	USB Device Drive SE0 Control Register	0x0000_0001
USB_D_BUFSEG0	USB_BA+0x500	R/W	Endpoint 0 Buffer Segmentation Register	0x0000_0000
USB_D_MXPLD0	USB_BA+0x504	R/W	Endpoint 0 Maximal Payload Register	0x0000_0000
USB_D_CFG0	USB_BA+0x508	R/W	Endpoint 0 Configuration Register	0x0000_0000
USB_D_CFGP0	USB_BA+0x50C	R/W	Endpoint 0 Set Stall and Clear In/Out Ready Control Register	0x0000_0000
USB_D_BUFSEG1	USB_BA+0x510	R/W	Endpoint 1 Buffer Segmentation Register	0x0000_0000
USB_D_MXPLD1	USB_BA+0x514	R/W	Endpoint 1 Maximal Payload Register	0x0000_0000
USB_D_CFG1	USB_BA+0x518	R/W	Endpoint 1 Configuration Register	0x0000_0000
USB_D_CFGP1	USB_BA+0x51C	R/W	Endpoint 1 Set Stall and Clear In/Out Ready Control Register	0x0000_0000
USB_D_BUFSEG2	USB_BA+0x520	R/W	Endpoint 2 Buffer Segmentation Register	0x0000_0000
USB_D_MXPLD2	USB_BA+0x524	R/W	Endpoint 2 Maximal Payload Register	0x0000_0000
USB_D_CFG2	USB_BA+0x528	R/W	Endpoint 2 Configuration Register	0x0000_0000
USB_D_CFGP2	USB_BA+0x52C	R/W	Endpoint 2 Set Stall and Clear In/Out Ready Control Register	0x0000_0000
USB_D_BUFSEG3	USB_BA+0x530	R/W	Endpoint 3 Buffer Segmentation Register	0x0000_0000
USB_D_MXPLD3	USB_BA+0x534	R/W	Endpoint 3 Maximal Payload Register	0x0000_0000
USB_D_CFG3	USB_BA+0x538	R/W	Endpoint 3 Configuration Register	0x0000_0000
USB_D_CFGP3	USB_BA+0x53C	R/W	Endpoint 3 Set Stall and Clear In/Out Ready Control Register	0x0000_0000
USB_D_BUFSEG4	USB_BA+0x540	R/W	Endpoint 4 Buffer Segmentation Register	0x0000_0000

<b>USB_D_MXPLD4</b>	USB_D_BA+0x544	R/W	Endpoint 4 Maximal Payload Register	0x0000_0000
<b>USB_D_CFG4</b>	USB_D_BA+0x548	R/W	Endpoint 4 Configuration Register	0x0000_0000
<b>USB_D_CFGP4</b>	USB_D_BA+0x54C	R/W	Endpoint 4 Set Stall and Clear In/Out Ready Control Register	0x0000_0000
<b>USB_D_BUFSEG5</b>	USB_D_BA+0x550	R/W	Endpoint 5 Buffer Segmentation Register	0x0000_0000
<b>USB_D_MXPLD5</b>	USB_D_BA+0x554	R/W	Endpoint 5 Maximal Payload Register	0x0000_0000
<b>USB_D_CFG5</b>	USB_D_BA+0x558	R/W	Endpoint 5 Configuration Register	0x0000_0000
<b>USB_D_CFGP5</b>	USB_D_BA+0x55C	R/W	Endpoint 5 Set Stall and Clear In/Out Ready Control Register	0x0000_0000
<b>USB_D_BUFSEG6</b>	USB_D_BA+0x560	R/W	Endpoint 6 Buffer Segmentation Register	0x0000_0000
<b>USB_D_MXPLD6</b>	USB_D_BA+0x564	R/W	Endpoint 6 Maximal Payload Register	0x0000_0000
<b>USB_D_CFG6</b>	USB_D_BA+0x568	R/W	Endpoint 6 Configuration Register	0x0000_0000
<b>USB_D_CFGP6</b>	USB_D_BA+0x56C	R/W	Endpoint 6 Set Stall and Clear In/Out Ready Control Register	0x0000_0000
<b>USB_D_BUFSEG7</b>	USB_D_BA+0x570	R/W	Endpoint 7 Buffer Segmentation Register	0x0000_0000
<b>USB_D_MXPLD7</b>	USB_D_BA+0x574	R/W	Endpoint 7 Maximal Payload Register	0x0000_0000
<b>USB_D_CFG7</b>	USB_D_BA+0x578	R/W	Endpoint 7 Configuration Register	0x0000_0000
<b>USB_D_CFGP7</b>	USB_D_BA+0x57C	R/W	Endpoint 7 Set Stall and Clear In/Out Ready Control Register	0x0000_0000

Memory Type	Address	Size	Description
<b>USB_D_BA = 0x4006_0000</b>			
<b>USB_D_SRAM</b>	USB_D_BA+0x100 ~ USB_D_BA+0x2FF	512 Bytes	The SRAM is used for the entire endpoints buffer. Refer to section Figure 6.18-3 for the endpoint SRAM structure and its description.

6.18.7 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	WKIDLEIEN	VBDETIEN	USBIEN	BUSIEN

Bits	Description	
[31:16]	Reserved	Reserved.
[15]	INNAKEN	<p><b>Active NAK Function and Its Status in IN Token</b></p> <p>0 = When device responds NAK after receiving IN token, IN NAK status will not be updated to USBD_EPSTS register, so that the USB interrupt event will not be asserted.</p> <p>1 = IN NAK status will be updated to USBD_EPSTS register and the USB interrupt event will be asserted, when the device responds NAK after receiving IN token.</p>
[14:9]	Reserved	Reserved.
[8]	WKEN	<p><b>Wake-up Function Enable Bit</b></p> <p>0 = USB Wake-up Function Disabled.</p> <p>1 = USB Wake-up Function Enabled.</p>
[7:5]	Reserved	Reserved.
[4]	SOFIEN	<p><b>Start of Frame Interrupt Enable Bit</b></p> <p>0 = SOF Interrupt Disabled.</p> <p>1 = SOF Interrupt Enabled.</p>
[3]	WKIDLEIEN	<p><b>USB Wake-up Idle Interrupt Enable Bit</b></p> <p>0 = Wake-up Idle Interrupt Disabled.</p> <p>1 = Wake-up Idle Interrupt Enabled.</p>
[2]	VBDETIEN	<p><b>VBUS Detection Interrupt Enable Bit</b></p> <p>0 = VBUS Detection Interrupt Disabled.</p> <p>1 = VBUS Detection Interrupt Enabled.</p>
[1]	USBIEN	<p><b>USB Event Interrupt Enable Bit</b></p> <p>0 = USB Event Interrupt Disabled.</p> <p>1 = USB Event Interrupt Enabled.</p>

[0]	<b>BUSIEN</b>	<b>Bus Event Interrupt Enable Bit</b> 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		Reserved					
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							
7	6	5	4	3	2	1	0
Reserved			SOFIF	WKIDLEIF	VBDDETIF	USBIF	BUSIF

Bits	Description	
[31]	SETUP	<b>Setup Event Status</b> 0 = No Setup event. 1 = Setup event occurred, cleared by write 1 to USB_D_INTSTS[31].
[30:24]	Reserved	Reserved.
[23]	EPEVT7	<b>Endpoint 7's USB Event Status</b> 0 = No event occurred in endpoint 7. 1 = USB event occurred on Endpoint 7, check USB_D_EPSTS[31:29] to know which kind of USB event was occurred, cleared by write 1 to USB_D_INTSTS[23] or USB_D_INTSTS[1].
[22]	EPEVT6	<b>Endpoint 6's USB Event Status</b> 0 = No event occurred in endpoint 6. 1 = USB event occurred on Endpoint 6, check USB_D_EPSTS[28:26] to know which kind of USB event was occurred, cleared by write 1 to USB_D_INTSTS[22] or USB_D_INTSTS[1].
[21]	EPEVT5	<b>Endpoint 5's USB Event Status</b> 0 = No event occurred in endpoint 5. 1 = USB event occurred on Endpoint 5, check USB_D_EPSTS[25:23] to know which kind of USB event was occurred, cleared by write 1 to USB_D_INTSTS[21] or USB_D_INTSTS[1].
[20]	EPEVT4	<b>Endpoint 4's USB Event Status</b> 0 = No event occurred in endpoint 4. 1 = USB event occurred on Endpoint 4, check USB_D_EPSTS[22:20] to know which kind of USB event was occurred, cleared by write 1 to USB_D_INTSTS[20] or USB_D_INTSTS[1].
[19]	EPEVT3	<b>Endpoint 3's USB Event Status</b> 0 = No event occurred in endpoint 3. 1 = USB event occurred on Endpoint 3, check USB_D_EPSTS[19:17] to know which kind of USB event was occurred, cleared by write 1 to USB_D_INTSTS[19] or USB_D_INTSTS[1].

[18]	EPEVT2	<p><b>Endpoint 2's USB Event Status</b></p> <p>0 = No event occurred in endpoint 2.</p> <p>1 = USB event occurred on Endpoint 2, check USBD_EPSTS[16:14] to know which kind of USB event was occurred, cleared by write 1 to USBD_INTSTS[18] or USBD_INTSTS[1].</p>
[17]	EPEVT1	<p><b>Endpoint 1's USB Event Status</b></p> <p>0 = No event occurred in endpoint 1.</p> <p>1 = USB event occurred on Endpoint 1, check USBD_EPSTS[13:11] to know which kind of USB event was occurred, cleared by write 1 to USBD_INTSTS[17] or USBD_INTSTS[1].</p>
[16]	EPEVT0	<p><b>Endpoint 0's USB Event Status</b></p> <p>0 = No event occurred in endpoint 0.</p> <p>1 = USB event occurred on Endpoint 0, check USBD_EPSTS[10:8] to know which kind of USB event was occurred, cleared by write 1 to USBD_INTSTS[16] or USBD_INTSTS[1].</p>
[15:5]	Reserved	Reserved.
[4]	SOFIF	<p><b>Start of Frame Interrupt Status</b></p> <p>0 = SOF event does not occur.</p> <p>1 = SOF event occurred, cleared by write 1 to USBD_INTSTS[4].</p>
[3]	WKIDLEIF	<p><b>No-event-wake-up Interrupt Status</b></p> <p>0 = WKIDLE event does not occur.</p> <p>1 = No-event-wake-up event occurred, cleared by write 1 to USBD_INTSTS[3].</p>
[2]	VBDETIF	<p><b>VBUS Detection Interrupt Status</b></p> <p>0 = There is not attached/detached event in the USB.</p> <p>1 = There is attached/detached event in the USB bus and it is cleared by write 1 to USBD_INTSTS[2].</p>
[1]	USBIF	<p><b>USB Event Interrupt Status</b></p> <p>The USB event includes the SETUP Token, IN Token, OUT ACK, ISO IN, or ISO OUT events in the bus.</p> <p>0 = No USB event occurred.</p> <p>1 = USB event occurred, check EPSTS0~5[2:0] to know which kind of USB event was occurred, cleared by write 1 to USBD_INTSTS[1] or EPSTS0~7 and SETUP (USBD_INTSTS[31]).</p>
[0]	BUSIF	<p><b>BUS Interrupt Status</b></p> <p>The BUS event means that there is one of the suspense or the resume function in the bus.</p> <p>0 = No BUS event occurred.</p> <p>1 = Bus event occurred; check USBD_ATTR[3:0] and USBD_ATTR[13:12] to know which kind of bus event was occurred, cleared by write 1 to USBD_INTSTS[0].</p>

**USB Device Function Address Register (USB\_D\_FADDR)**

A 7-bit value is used as the address of a device on the USB BUS.

Register	Offset	R/W	Description	Reset Value
USB_D_FADDR	USB_D_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 (USB\_D\_EPSTS)**

Register	Offset	R/W	Description	Reset Value
USB_D_EPSTS	USB_D_BA+0x00C	R	USB Device Endpoint Status Register	0x0000_0000

31	30	29	28	27	26	25	24
EPSTS7			EPSTS6			EPSTS5	
23	22	21	20	19	18	17	16
EPSTS5	EPSTS4			EPSTS3			EPSTS2
15	14	13	12	11	10	9	8
EPSTS2		EPSTS1			EPSTS0		
7	6	5	4	3	2	1	0
OV	Reserved						

Bits	Description
[31:29]	<p><b>EPSTS7</b></p> <p><b>Endpoint 7 Status</b>                      These bits are used to indicate the current status of this endpoint                      000 = In ACK.                      001 = In NAK.                      010 = Out Packet Data0 ACK.                      011 = Setup ACK.                      110 = Out Packet Data1 ACK.                      111 = Isochronous transfer end.</p>
[28:26]	<p><b>EPSTS6</b></p> <p><b>Endpoint 6 Status</b>                      These bits are used to indicate the current status of this endpoint                      000 = In ACK.                      001 = In NAK.                      011 = Setup ACK.                      010 = Out Packet Data0 ACK.                      110 = Out Packet Data1 ACK.                      111 = Isochronous transfer end.</p>
[25:23]	<p><b>EPSTS5</b></p> <p><b>Endpoint 5 Status</b>                      These bits are used to indicate the current status of this endpoint                      000 = In ACK.                      001 = In NAK.                      010 = Out Packet Data0 ACK.                      011 = Setup ACK.                      110 = Out Packet Data1 ACK.                      111 = Isochronous transfer end.</p>
[22:20]	<p><b>EPSTS4</b></p> <p><b>Endpoint 4 Status</b>                      These bits are used to indicate the current status of this endpoint                      000 = In ACK.</p>

		<p>001 = In NAK.                  010 = Out Packet Data0 ACK.                  011 = Setup ACK.                  110 = Out Packet Data1 ACK.                  111 = Isochronous transfer end.</p>
[19:17]	<b>EPSTS3</b>	<p><b>Endpoint 3 Status</b>                  These bits are used to indicate the current status of this endpoint                  000 = In ACK.                  001 = In NAK.                  010 = Out Packet Data0 ACK.                  011 = Setup ACK.                  110 = Out Packet Data1 ACK.                  111 = Isochronous transfer end.</p>
[16:14]	<b>EPSTS2</b>	<p><b>Endpoint 2 Status</b>                  These bits are used to indicate the current status of this endpoint                  000 = In ACK.                  001 = In NAK.                  010 = Out Packet Data0 ACK.                  011 = Setup ACK.                  110 = Out Packet Data1 ACK.                  111 = Isochronous transfer end.</p>
[13:11]	<b>EPSTS1</b>	<p><b>Endpoint 1 Status</b>                  These bits are used to indicate the current status of this endpoint                  000 = In ACK.                  001 = In NAK.                  010 = Out Packet Data0 ACK.                  011 = Setup ACK.                  110 = Out Packet Data1 ACK.                  111 = Isochronous transfer end.</p>
[10:8]	<b>EPSTS0</b>	<p><b>Endpoint 0 Status</b>                  These bits are used to indicate the current status of this endpoint                  000 = In ACK.                  001 = In NAK.                  010 = Out Packet Data0 ACK.                  011 = Setup ACK.                  110 = Out Packet Data1 ACK.                  111 = Isochronous transfer end.</p>
[7]	<b>OV</b>	<p><b>Overrun</b>                  It indicates that the received data is over 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.</p>
[6:0]	<b>Reserved</b>	Reserved.

**USB Bus Status and Attribution Register (USB\_D\_ATTR)**

Register	Offset	R/W	Description	Reset Value
USB_D_ATTR	USB_D_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	Reserved	DPPUEN
7	6	5	4	3	2	1	0
USBEN	Reserved	RWAKEUP	PHYEN	TOUT	RESUME	SUSPEND	USBRST

Bits	Description	
[31:14]	Reserved	Reserved.
[13]	L1RESUME	<b>LPM L1 Resume (Read Only)</b> 0 = Bus no LPM L1 state resume. 1 = LPM L1 state Resume from LPM L1 state suspend.
[12]	L1SUSPEND	<b>LPM L1 Suspend (Read Only)</b> 0 = Bus 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	<b>LPM Token Acknowledge Enable Bit</b> 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	<b>CPU Access USB SRAM Size Mode Selection</b> 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]	Reserved	Reserved.
[8]	DPPUEN	<b>Pull-up Resistor on USB_DP Enable Bit</b> 0 = Pull-up resistor in USB_D+ bus Disabled. 1 = Pull-up resistor in USB_D+ bus Active.
[7]	USBEN	<b>USB Controller Enable Bit</b> 0 = USB Controller Disabled. 1 = USB Controller Enabled.
[6]	Reserved	Reserved.
[5]	RWAKEUP	<b>Remote Wake-up</b>

		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]	<b>PHYEN</b>	<b>PHY Transceiver Function Enable Bit</b> 0 = PHY transceiver function Disabled. 1 = PHY transceiver function Enabled.
[3]	<b>TOUT</b>	<b>Time-out Status (Read Only)</b> 0 = No time-out. 1 = No Bus response more than 18 bits time.
[2]	<b>RESUME</b>	<b>Resume Status (Read Only)</b> 0 = No bus resume. 1 = Resume from suspend.
[1]	<b>SUSPEND</b>	<b>Suspend Status (Read Only)</b> 0 = No Bus suspend. 1 = Bus idle more than 3ms, either cable is plugged off or host is sleeping.
[0]	<b>USBRST</b>	<b>USB Reset Status (Read Only)</b> 0 = No Bus reset. 1 = Bus reset when SE0 (single-ended 0) more than 2.5us.

**USB Device VBUS Detection Register (USB\_D\_VBUSDET)**

Register	Offset	R/W	Description	Reset Value
USB_D_VBUSDET	USB_D_BA+0x014	R	USB Device VBUS 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	<b>Device VBUS Detection</b> 0 = Controller is not attached to the USB host. 1 = Controller is attached to the USB host.

**USB SETUP Token Buffer Segmentation Register (USB\_D\_STBUFSEG)**

Register	Offset	R/W	Description	Reset Value
USB_D_STBUFSEG	USB_D_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	<p><b>SETUP Token Buffer Segmentation</b></p> <p>It is used to indicate the offset address for the SETUP token with the USB Device SRAM starting address. The effective starting address is                      USB_D_SRAM address + {STBUFSEG[8:3], 3'b000}</p> <p>Where the USB_D_SRAM address = USB_D_BA+0x100h.</p> <p><b>Note:</b> It is used for SETUP token only.</p>
[2:0]	Reserved	Reserved.

**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	<b>LPM Remote Wakeup</b> This bit contains the bRemoteWake value received with last ACK LPM Token
[7:4]	LPMBESL	<b>LPM Best Effort Service Latency</b> These bits contain the BESL value received with last ACK LPM Token 0000 = 125us. 0001 = 150us. 0010 = 200us. 0011 = 300us. 0100 = 400us. 0101 = 500us. 0110 = 1000us. 0111 = 2000us. 1000 = 3000us. 1001 = 4000us. 1010 = 5000us. 1011 = 6000us. 1100 = 7000us. 1101 = 8000us. 1110 = 9000us. 1111 = 10000us.
[3:0]	LPMLINKSTS	<b>LPM Link State</b> These bits contain the bLinkState received with last ACK LPM Token 0000 = Reserve. 0001 = L1 (Sleep). 0010 – 1111 = Reserve.

**USB Frame Number Register (USB\_D\_FN)**

Register	Offset	R/W	Description	Reset Value
USB_D_FN	USB_D_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	<b>Frame Number</b> These bits contain the 11-bits frame number in the last received SOF packet.



**USB Drive SE0 Register (USB\_D\_SE0)**

Register	Offset	R/W	Description	Reset Value
USB_D_SE0	USB_D_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	<p><b>Drive Single Ended Zero in USB Bus</b></p> <p>The Single Ended Zero (SE0) is when both lines (USB_D+ and USB_D-) are being pulled low.</p> <p>0 = Normal operation.</p> <p>1 = Force USB PHY transceiver to drive SE0.</p>

**USB Buffer Segmentation Register (USB\_BUFSEGx)**

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

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	<p><b>Endpoint Buffer Segmentation</b></p> <p>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 <math>USB\_SRAM\ address + \{ BUFSEG[8:3], 3'b000\}</math></p> <p>Where the <math>USBD\_SRAM\ address = USBD\_BA + 0x100h</math>.</p> <p>Refer to the section Figure 6.18-3 for the endpoint SRAM structure and its description.</p>
[2:0]	Reserved	Reserved.

**USB Maximal Payload Register (USB\_MXPLDx)**

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

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	<p><b>Maximal Payload</b></p> <p>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.</p> <p>(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.</p> <p>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.</p> <p>(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.</p> <p><b>Note:</b> Once MXPLD is written, the data packets will be transmitted/received immediately after IN/OUT token arrived.</p>

**USB Configuration Register (USB\_CFGx)**

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

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	<b>Clear STALL Response</b> 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	<b>Data Sequence Synchronization</b> 0 = DATA0 PID. 1 = DATA1 PID. <b>Note:</b> 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 the bit.
[6:5]	STATE	<b>Endpoint State</b> 00 = Endpoint Disabled. 01 = Out endpoint. 10 = IN endpoint. 11 = Undefined.
[4]	ISOCH	<b>Isochronous Endpoint</b>

		This bit is used to set the endpoint as Isochronous endpoint, no handshake. 0 = No Isochronous endpoint. 1 = Isochronous endpoint.
[3:0]	<b>EPNUM</b>	<b>Endpoint Number</b> These bits are used to define the endpoint number of the current endpoint

**USB Extra Configuration Register (USB\_CFGPx)**

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

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	<p><b>Set STALL</b></p> <p>0 = Disable the device to response STALL. 1 = Set the device to respond STALL automatically.</p>
[0]	CLRRDY	<p><b>Clear Ready</b></p> <p>When the USBD_MXPLDx 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.</p> <p>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.</p>

## 6.19 USCI - Universal Serial Control Interface Controller

### 6.19.1 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 I<sup>2</sup>C functional protocol.

### 6.19.2 Features

The controller can be individually configured to match the application needs. The following protocols are supported:

- UART
- SPI
- I<sup>2</sup>C

### 6.19.3 Block Diagram

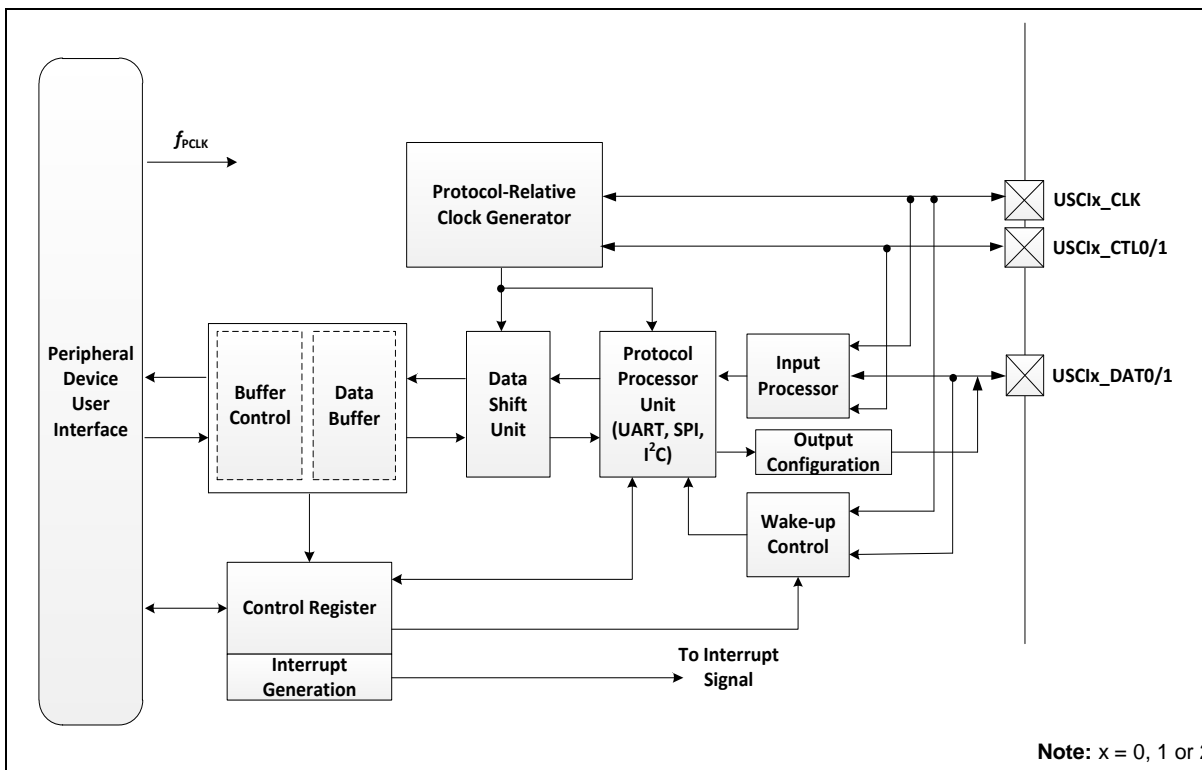


Figure 6.19-1 USCI Block Diagram

### 6.19.4 Functional Description

The structure of the Universal Serial Control Interface (USCI) controller is shown in Figure 6.19-1 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 I<sup>2</sup>C. They can be selected by FUNMODE (USCI\_CTL [2:0]). Note that the FUNMODE must be set 0 before changing protocol.

6.19.4.1 I/O Processor

**Input Signal**

All input stages offer the similar feature set. They are used for all protocols.

Table 6.19-1 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 <sup>2</sup> 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 6.19-1 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).

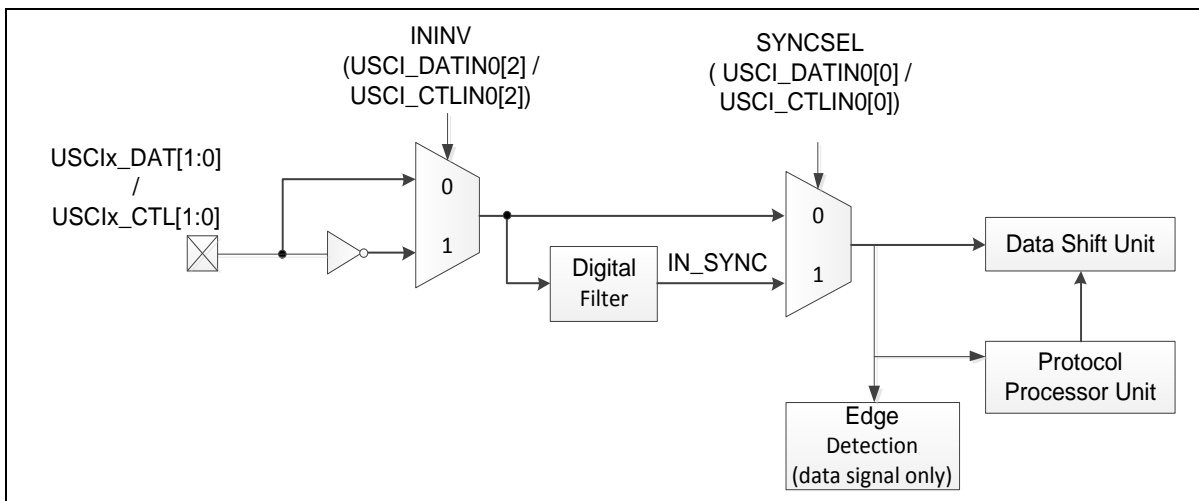


Figure 6.19-2 Input Conditioning for USCIX\_DAT[1:0] and USCIX\_CTL[1:0]

The input structure of USCIX\_CLK is similar to USCIX\_CTL[1:0] input structure, except it does not support inverse function.



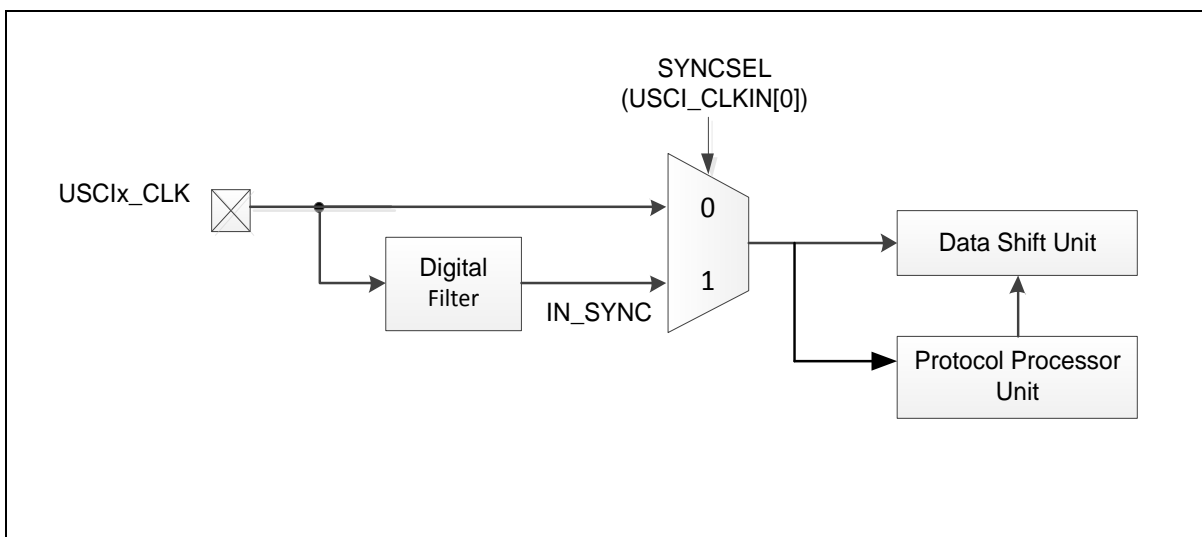


Figure 6.19-3 Input Conditioning for USC1x\_CLK

All configurations of control, clock and data input structures are in USC1\_CTLIN0, USC1\_CLKIN and USC1\_DATIN0 registers respectively. EDGEDET (USC1\_DATIN0[4:3]) is used to select the edge detection condition. Note that the EDGEDET for USC1\_DATIN0 must be set 2'b10 on UART mode. The programmable edge detection indicates that the desired event has occurred by activating the trigger signal.

ININV (USC1\_DATIN0[2] / USC1\_CTLIN0[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 (USC1\_DATIN0[0] / USC1\_CTLIN0[0] / USC1\_CLKIN[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, there is the possibility to synchronize the input signal (signal IN\_SYNC is synchronized to  $f_{PCLK}$ ). The synchronized input signal is taken into account by SYNCSEL = 1. The synchronization leads to a delay in the signal path of 2-3 times the period of  $f_{PCLK}$ .

**Output Signals**

Table 6.19-2 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 <sup>2</sup> C
Serial Bus Clock Output	USC1x_CLK	-	SPI_CLK	SCL
Control Output	USC1x_CTL0	-	SPI_SS	-
	USC1x_CTL1	nRTS	-	-
Data Output	USC1x_DAT0	-	SPI_MOSI_0	SDA
	USC1x_DAT1	TX	SPI_MISO_0	-

Table 6.19-2 Output Signals for Different Protocols

The description of protocol-specific items are given in the related protocol chapters.

6.19.4.2 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 USC1x\_DAT0 pin or USC1x\_DAT1 pin depends on what protocol is selected.

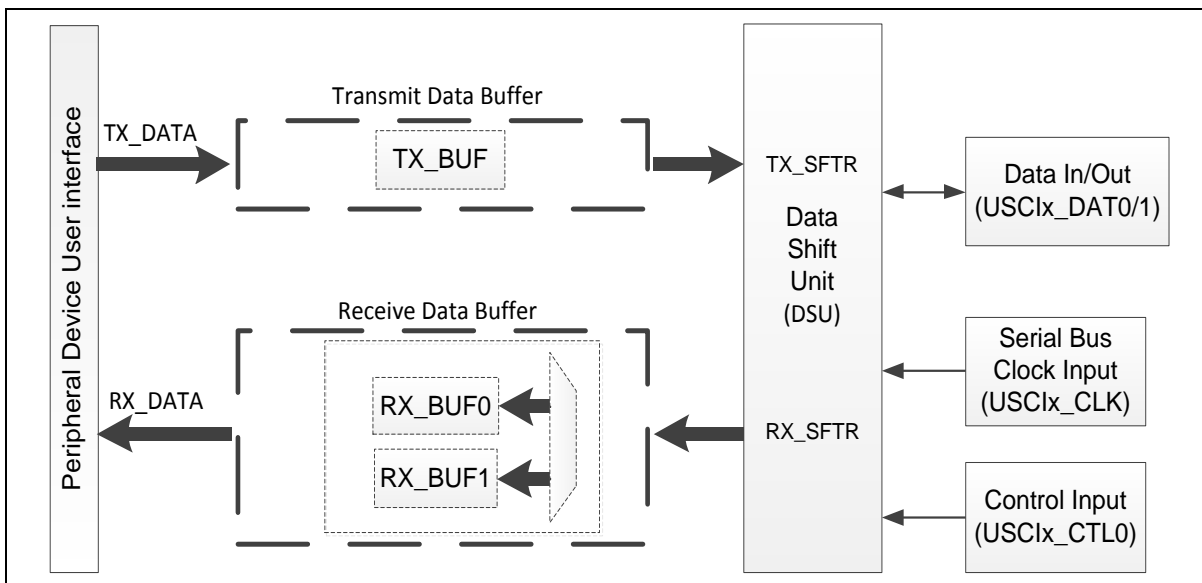


Figure 6.19-4 Block Diagram of Data Buffering

The operation of data handling includes:

- The peripheral device user interface (APB) is used to handle data, interrupts, 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 USC1\_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 USC1\_RXDAT register. The first received data is read out first and the next received data becomes visible in USC1\_RXDAT and can be read out next.

Transmit data can be loaded to TX\_BUF by writing to the transmit register USC1\_TXDAT.

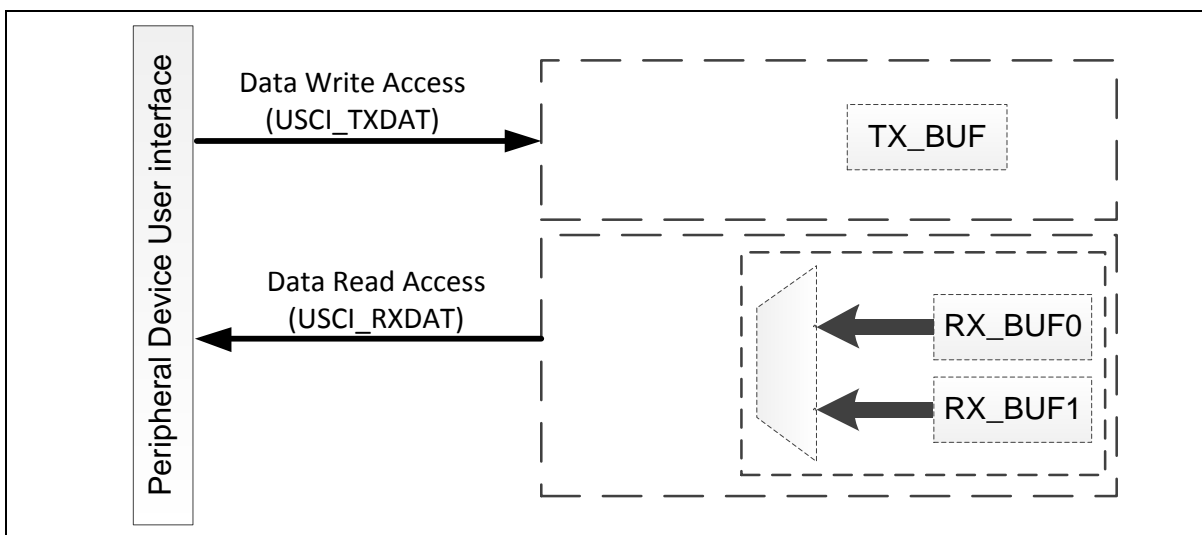


Figure 6.19-5 Data Access Structure

### 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.

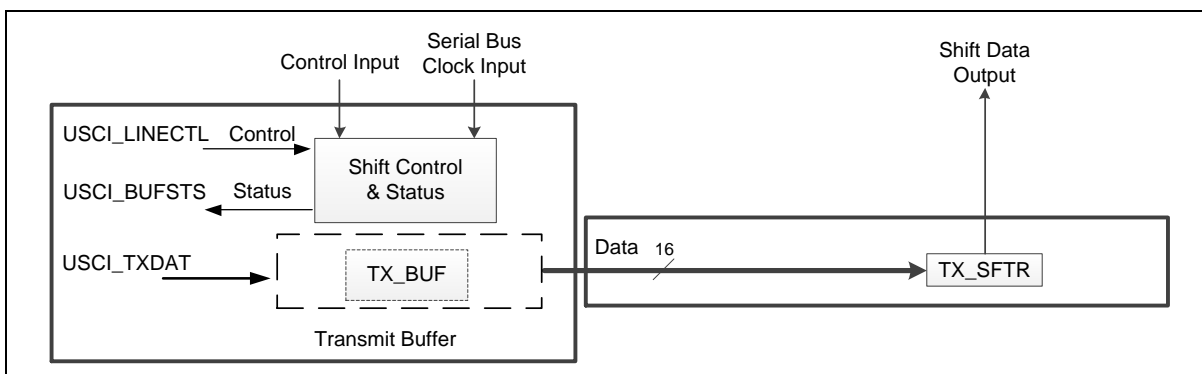


Figure 6.19-6 Transmit Data Path

### 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 (USCI\_BUFSTS[8]) 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.

**I<sup>2</sup>C:** 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.

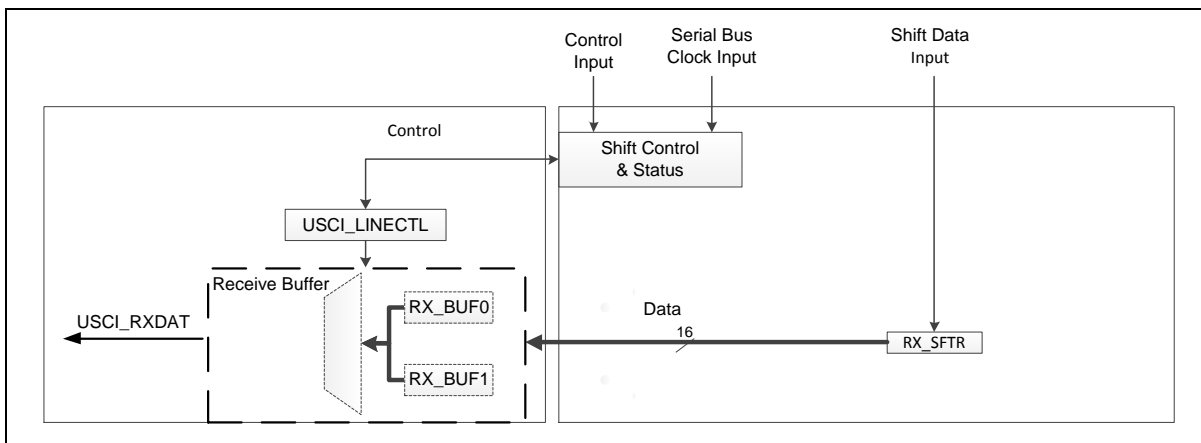


Figure 6.19-7 Receive Data Path

#### 6.19.4.3 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.

6.19.4.4 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.

6.19.4.5 Protocol-Relative Clock Generator

The USCI 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 6.19-8.

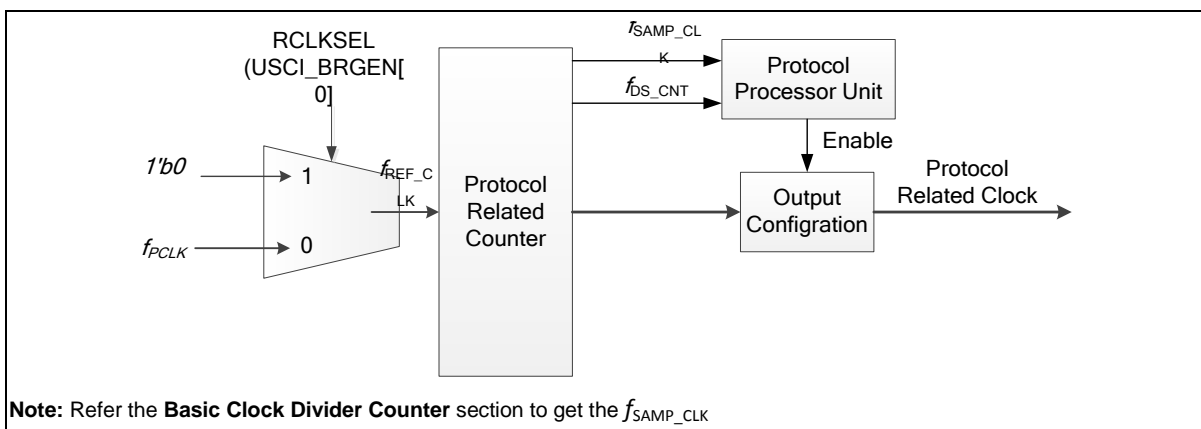


Figure 6.19-8 Protocol-Relative Clock Generator

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.

$$f_{DIV\_CLK} = f_{REF\_CLK} \times \frac{1}{CLKDIV + 1} \text{ if PTCLKSEL} = 0$$

$$f_{DIV\_CLK} = f_{REF\_CLK} \times \frac{1}{(CLKDIV + 1) \times 2} \text{ if PTCLKSEL} = 1$$

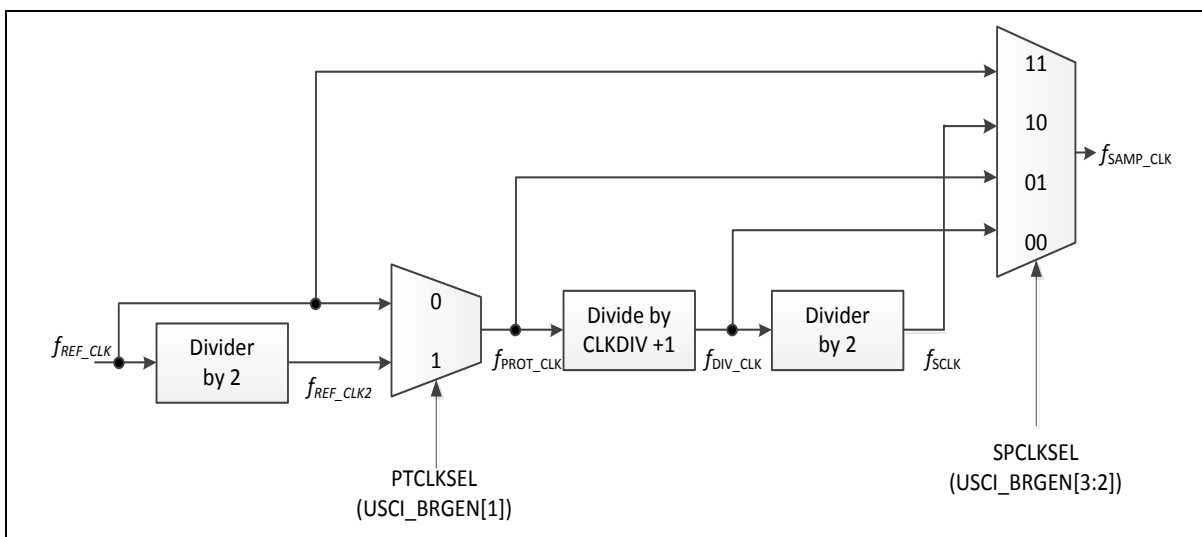


Figure 6.19-9 Basic Clock Divider Counter

### 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.

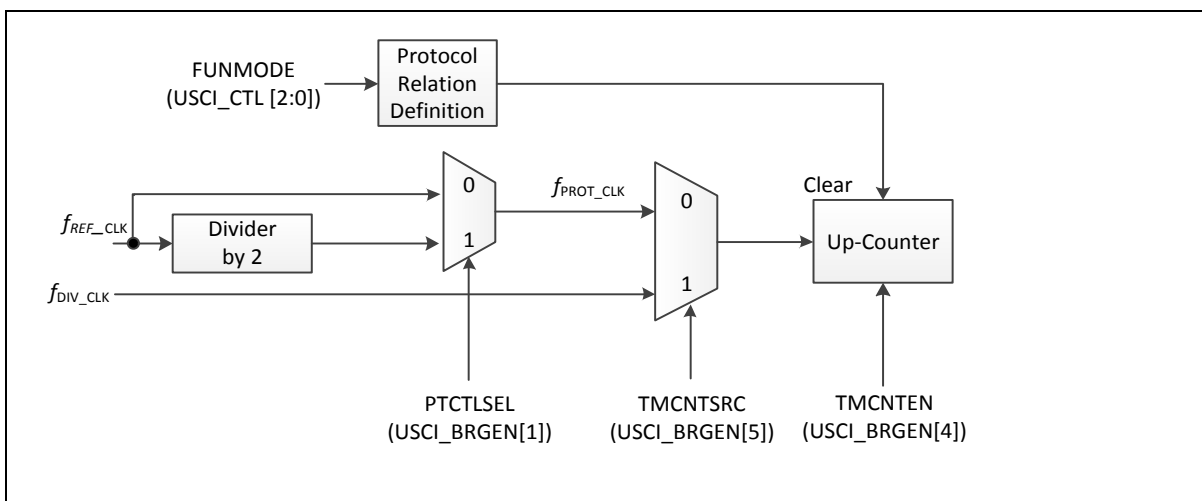


Figure 6.19-10 Block of Timing Measurement Counter

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.
- I<sup>2</sup>C: 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  $f_{SAMP\_CLK}$ . The sample time

counter allows generating time intervals for protocol-specific purposes. The period of a sample frequency  $f_{PDS\_CNT}$  is given by the selected input frequency  $f_{SAMP\_CLK}$  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.

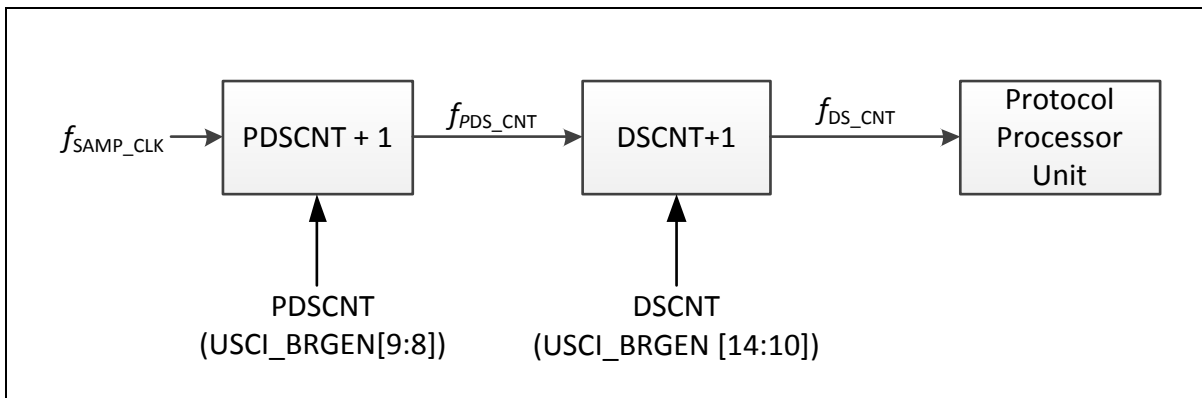


Figure 6.19-11 Sample Time Counter

#### 6.19.4.6 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 enabled, 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 finished. It is indicated by flag TXENDIF (USCI\_PROTSTS [2]) and, if enabled, 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 I<sup>2</sup>C mode, the transmit data valid is 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 enabled, 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 enabled, 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 enabled, leads to a protocol interrupt.

The general event and interrupt structure is shown in below.

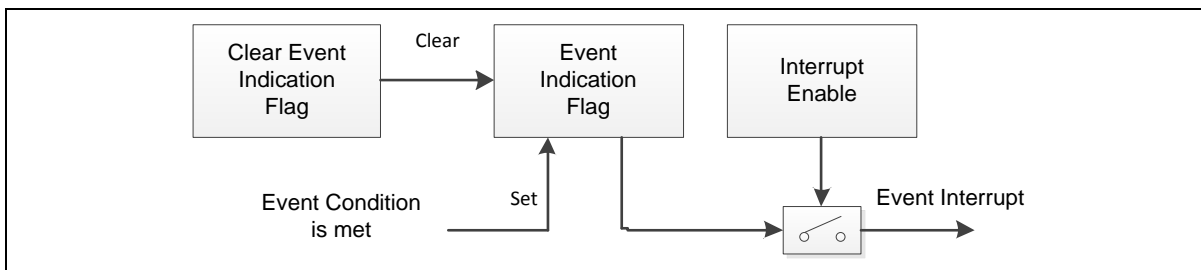


Figure 6.19-12 Event and Interrupt Structure

Each general interrupt enable can set by RXENDIEN, RXSTIEN, TXENDIEN, and TXSTIEN of USCI\_INTEN [4:1]. The events are including 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 6.19-3.

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 6.19-3 Data Transfer Events and Interrupt Handling

#### 6.19.4.7 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 I <sup>2</sup> C mode	USCI_PROTSTS [13:8], USCI_PROTSTS [5]		USCI_PROTIEN [6:0]
---	--	--	--------------------

Table 6.19-4 Protocol-specific Events and Interrupt Handling

6.19.4.8 *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.

## 6.20 USCI – UART Mode

### 6.20.1 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.

### 6.20.2 Features

- Supports one transmit buffer and two receive buffer for data payload
- Supports hardware auto flow control function
- Supports programmable baud-rate generator
- Support 9-Bit Data Transfer (Support 9-Bit RS-485)
- Baud rate detection possible by built-in capture event of baud rate generator
- 
- Supports Wake-up function (Data and nCTS Wakeup Only)

### 6.20.3 Block Diagram

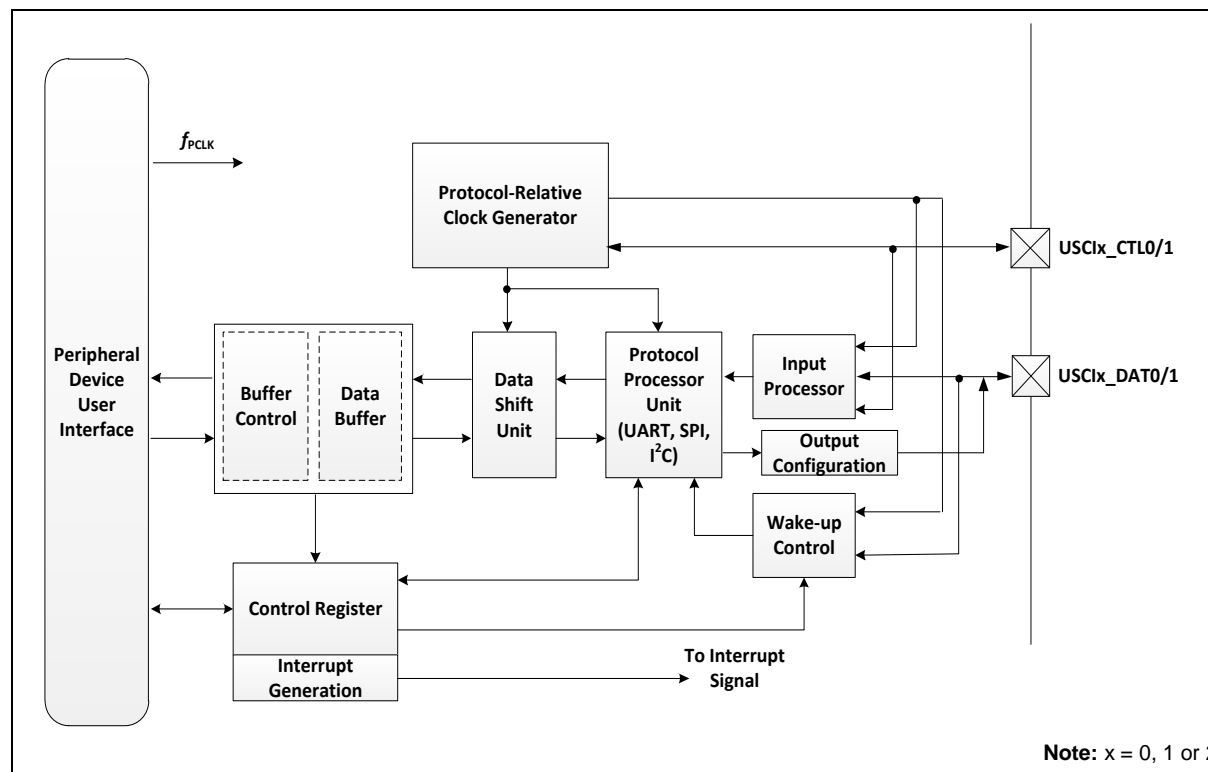


Figure 6.20-1 USCI UART Mode Block Diagram

## 6.20.4 Basic Configuration

### 6.20.4.1 Basic Configuration of USCI0 UART

- Clock Source Configuration
  - Enable USCI0 peripheral clock in USCI0CKEN (CLK\_APBCLK1[8])
- Reset Configuration
  - Reset USCI0 controller in USCI0RST (SYS\_IPRST2[8])
- Pin Configuration

USCI0 pins are configured in SYS\_GPA\_MFPL, SYS\_GPB\_MFPH, SYS\_GPC\_MFPL and SYS\_GPE\_MFPL registers.

Group	Pin Name	GPIO	MFP
USCI0	USCI0_CLK	PE.5	MFP4
		PC.4	MFP5
		PB.9	MFP8
	USCI0_CTL0	PE.4	MFP4
		PC.3	MFP5
		PB.8, PE.2	MFP8
	USCI0_CTL1	PC.2, PC.7	MFP4
		PB.4	MFP8
	USCI0_DAT0	PC.0, PC.5	MFP4
		PB.2	MFP8
	USCI0_DAT1	PC.1, PC.6	MFP4
		PB.3	MFP8

### 6.20.4.2 Basic Configuration of USCI1 UART

- Clock Source Configuration
  - Enable USCI1 peripheral clock in USCI1CKEN (CLK\_APBCLK1[9])
- Reset Configuration
  - Reset USCI1 controller in USCI0RST (SYS\_IPRST2[9])
- Pin Configuration

USCI1 pins are configured in SYS\_GPA\_MFPL, SYS\_GPA\_MFPH, SYS\_GPB\_MFPL, SYS\_GPC\_MFPL and SYS\_GPC\_MFPH registers

Group	Pin Name	GPIO	MFP
USCI1	USCI1_CLK	PD.15	MFP1
		PA.15	MFP4
		PA.3	MFP8

	USCI1_CTL0	PD.12	MFP1
		PA.0	MFP4
		PA.2	MFP8
	USCI1_CTL1	PD.7	MFP1
		PA.1, PA.14	MFP4
	USCI1_DAT0	PD.14	MFP1
		PB.0	MFP6
	USCI1_DAT1	PD.13	MFP1
PB.1		MFP8	

6.20.4.3 Basic Configuration of USCI2 UART

- Clock Source Configuration
  - Enable USCI2 peripheral clock in USCI2CKEN (CLK\_APBCLK1[10])
- Reset Configuration
  - Reset USCI2 controller in USCI0RST (SYS\_IPRST2[10])
- Pin Configuration

USCI2 pins are configured in SYS\_GPB\_MFPL, SYS\_GPC\_MFPH, SYS\_GPD\_MFPL, SYS\_GPD\_MFPH and SYS\_GPF\_MFPL registers

Group	Pin Name	GPIO	MFP
USCI2	USCI2_CLK	PC.11, PD.1	MFP4
		PF.2	MFP5
	USCI2_CTL0	PC.12, PD.0, PD.9	MFP4
		PF.1	MFP5
	USCI2_CTL1	PB.7, PC.9, PD.8	MFP4
		PF.0	MFP5
	USCI2_DAT0	PC.13, PD.2	MFP4
		PD.10	MFP5
	USCI2_DAT1	PC.10, PD.3	MFP4
		PD.11	MFP5

6.20.5 Functional Description

6.20.5.1 USCI Common Function Description

Please refer to section 6.19.4 for detailed information.

6.20.5.2 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 USC1x\_DAT0 and the transmit output (TXD) signal is handled by the output stage of USC1x\_DAT1.

For full-duplex communication, an independent communication line is needed for each transfer direction. The following figure shows an example with a point-to-point full-duplex connection between two communication partners UART module A and UART module B.

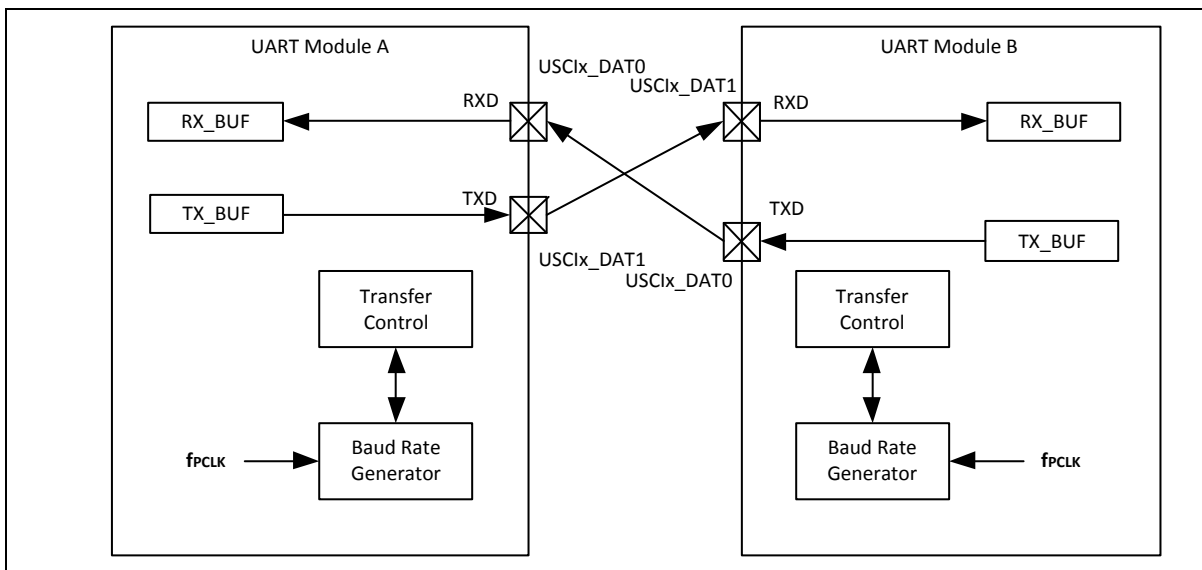


Figure 6.20-2 UART Signal Connection for Full-Duplex Communication

### Input Signal

For UART protocol, the number of input signals be figured in Table 6.20-1. 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 the following table.

Selected Protocol		UART
Control Input	USC1x_CTL0	nCTS
	USC1x_CTL1	X
Data Input(s)	USC1x_DAT0	RX
	USC1x_DAT1	X

Table 6.20-1 Input Signals for UART Protocols

### 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	USC1_CTL0	X
	USC1_CTL1	nRTS

Data Output (s)	USCI_DAT0	X
	USCI_DAT1	TX

Table 6.20-2 Output Signals for Different Protocols

### 6.20.5.3 Frame Format

A standard UART frame is shown in following figure. 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.

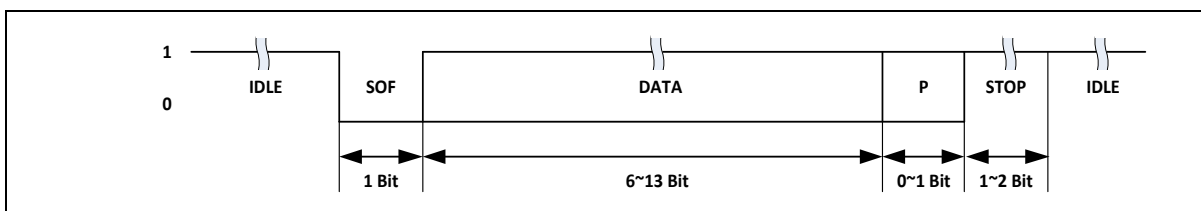


Figure 6.20-3 UART Standard Frame Format

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 USC1x\_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 (UUART\_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 (UUART\_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 (UUART\_PROTCTL[1]) and EVENPARITY (UUART\_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 (UUART\_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.

*6.20.5.4 Operating Mode*

In order 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 010B 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 USCI\_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 USCI\_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 USCI\_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 USCI\_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.

*6.20.5.5 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 6.20-4, 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.

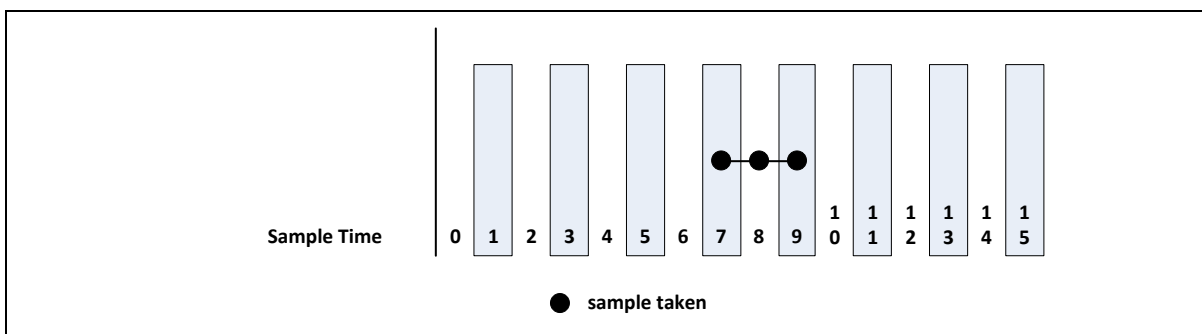


Figure 6.20-4 UART Bit Timing (data sample time)

### 6.20.5.6 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 (UUART\_BRGEN [0])  
to define the input frequency  $f_{REF\_CLK}$
- SPCLKSEL (UUART\_BRGEN[3:2])  
to define the multiple source of the sample clock  $f_{SAMP\_CLK}$
- PDSCNT (UUART\_BRGEN [9:8])  
to define the length of a data sample time (division of  $f_{REF\_CLK}$  by 1, 2, 3, or 4)
- DSCNT (UUART\_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:

$$f_{UART} = f_{REF\_CLK} \times \frac{1}{CLKDIV + 1} \times \frac{1}{PDSCNT + 1} \times \frac{1}{DSCNT + 1}$$

To generate slower frequencies, additional divide-by-2 stages can be selected by PTCLKSEL = 1 ( $f_{PROT\_CLK} = f_{REF\_CLK/2}$ ), leading to:

$$f_{UART} = \frac{f_{REF\_CLK}}{2} \times \frac{1}{CLKDIV + 1} \times \frac{1}{PDSCNT + 1} \times \frac{1}{DSCNT + 1}$$

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:

$$f_{UART} = f_{REF\_CLK} \times \frac{1}{CLKDIV + 1} \times \frac{1}{2} \times \frac{1}{PDSCNT + 1} \times \frac{1}{DSCNT + 1}$$

### 6.20.5.7 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 (USCIx\_DAT0) and then revised the baud rate clock divider CLKDIV (UUART\_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 (USCIx\_DAT0) and the actual timer value is captured into bit field BRDETITV (UUART\_PROTCTL [24:16]) in each falling edge of the detected signal.

When the ABREN (UUART\_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 BRDETIV (UART\_PROTCTL [24:16]) in the next falling edge. It is suggested to use the  $f_{DIV\_CLK}$  (TMCNTSRC (UART\_BRGENC[5]) = 1) as the counter source.

The CLKDIV (UART\_BRGEN[25:16]) will be revised by BRDETIV (UART\_PROTCTL [25:16]) after the auto baud rate function done (the time of 4<sup>th</sup> 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 (UART\_BRGEN[25:16]) and DSCNT (UART\_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 PDESCNT (UART\_BRGEN[9:8]) is 0x0.

During the auto baud rate detection, the ABRDETIF (UART\_PROTSTS[9]) and the BRDETIV (UART\_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 4<sup>th</sup> 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 BRDETIV 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 (UART\_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 BRDETIV.

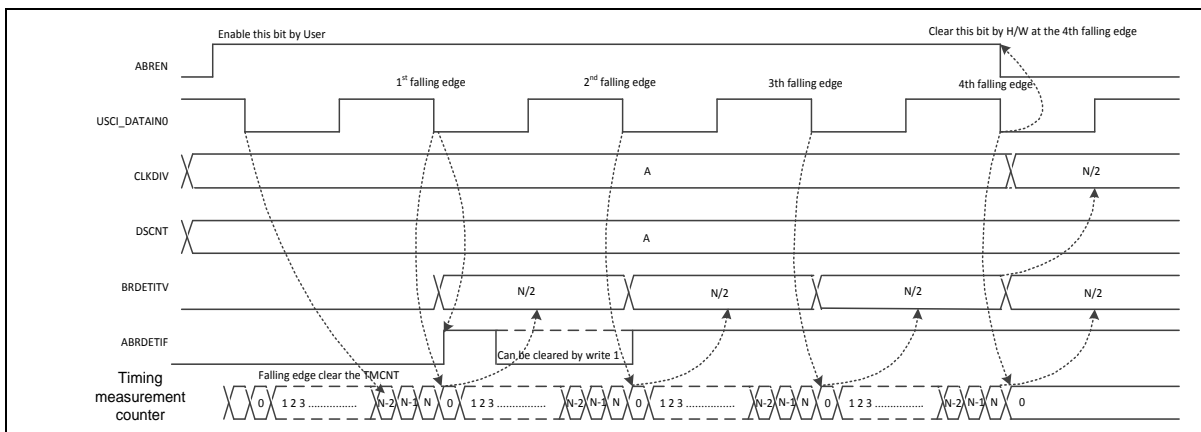


Figure 6.20-5 UART Auto Baud Rate Control

### 6.20.5.8 Auto Flow Control

The UART supports hardware auto-flow control that provides nRTS flow control on receiver buffer indicator RXFULL (UART\_BUFSTS[1]). 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.

### 6.20.5.9 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 (UART\_PROTCTL[1]) be set) when the STICKEN (UART\_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 also can 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.

### 6.20.5.10 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 is described as follows:

#### (a) Incoming data wake-up

When system is in power-down and both of the WKEN (UUART\_WKCTL [0]) and DATWKEN (UUART\_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 (UUART\_PROTCTL[14:11]) shall be set. These bits field of WAKECNT (UUART\_PROTCTL [14:11]) indicate how many clock cycle selected by  $f_{PDS\_CLK}$  do the controller can get the 1<sup>st</sup> bit (start bit) when the device is wakeup from Power-down mode.

**Note1:** By the WAKECNT is loaded into the hardware counter at the time of WKF (UUART\_WKSTS[0]) is clear so that the user shall clear the wakeup flag first to make sure the time period of WAKECNT is closed to the wake time of system.

**Note2:** In order to receive the incoming data, the relation between the selected clock stable and the baud rate shall be take care. (for example: The stable time of HXT is 4096 clock period.

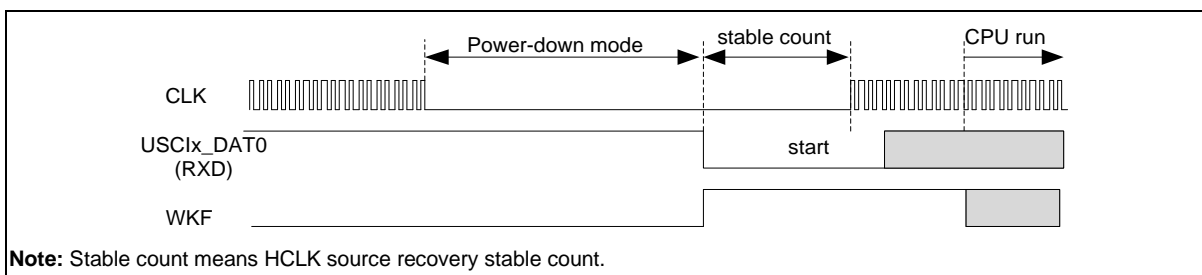


Figure 6.20-6 Incoming Data Wake-Up

#### (b) nCTS pin wake-up

When system is in power-down and both of the WKEN (UUART\_WKCTL [0]) and CTSWKEN (UUART\_PROTCTL[10]) are set , the toggle of nCTS pin can wake-up the system.

##### Case 1(nCTS transition from low to high):

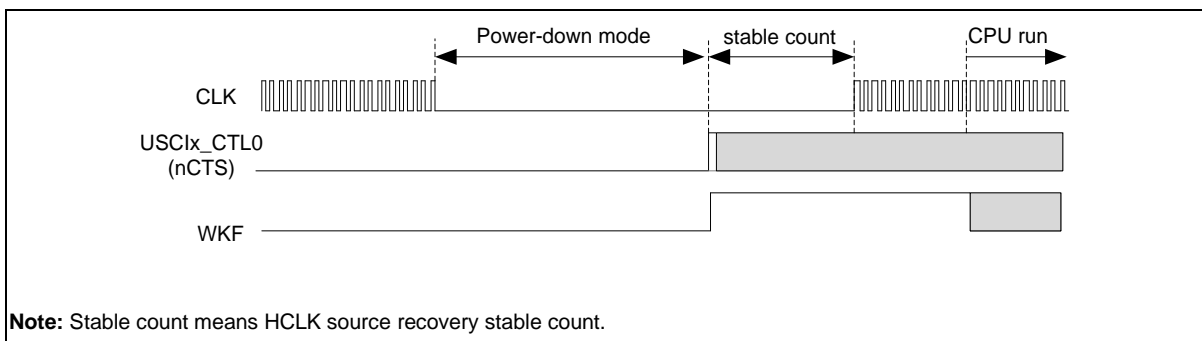


Figure 6.20-7 nCTS Wake-Up Case 1

**Case 2 (nCTS transition from high to low):**

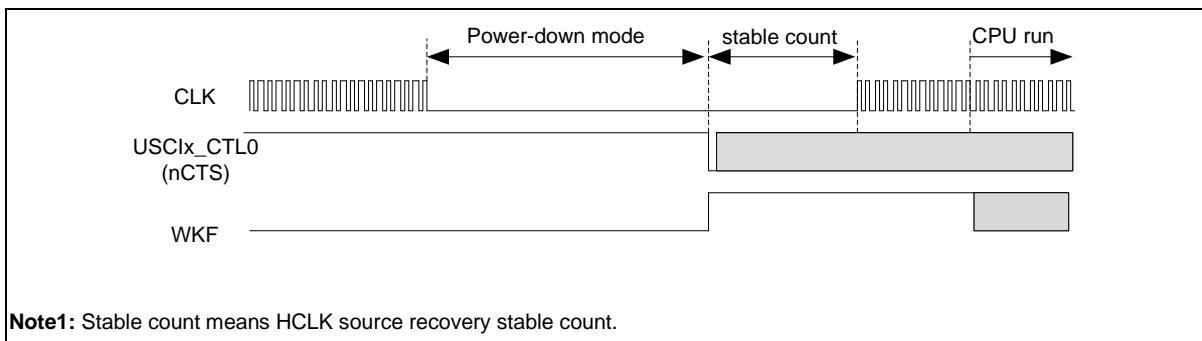


Figure 6.20-8 nCTS Wake-Up Case 2

6.20.5.11 Interrupt Events

**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 (0 = received parity bit equal to calculated parity value), and the result of frame check by the protocol-related error indication (0 = received stop bit equal to the format value '1'). This information is elaborated for each data frame.

The break error flag 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]), indicate 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 BRDETITV (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 (UUART\_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 (UUART\_PROTSTS [4]) becomes set at the end of the last receive bit.

#### 6.20.5.12 Programming Example

The following steps are used to configure the UART protocol setting and the data transmission.

1. Set FUNMODE (UUART\_CTL[2:0]) to 0x2 to select UART protocol.
2. Write baud rate generator register UUART\_BRGEN to select desired baud rate.
  - Set SPCLKSEL (UUART\_BRGEN[3:2]), PTCLKSEL (UUART\_BRGEN[1]) and RCLKSEL (UUART\_BRGEN[0]) to select the clock source.
  - Configure CLKDIV (UUART\_BRGEN[25:16]), DSCNT (UUART\_BRGEN[14:10]) and PDSCNT (UUART\_BRGEN[9:8]) to determine the baud rate divider.
3. Write line control register UUART\_LINECTL and protocol control register UUART\_PROTCTL to configure the transmission data format and UART protocol setting.
  - Program data field length in DWIDTH (UUART\_LINECTL[11:8]).
  - Enable parity bit and determine the parity bit type by setting EVENPARITY (UUART\_PROTCTL[2]) and PARITYEN (UUART\_PROTCTL[1]).
  - Configure stop bit length by setting STOPB (UUART\_PROTCTL[0]).
  - Enable LSB (UUART\_LINECTL[0]) to select LSB first transmission for UART protocol.
  - Set EDGEDET (UUART\_DATIN0[4:3]) to “10” to select the detected edge as falling edge for receiver start bit detection.
4. Set PROTEN (UUART\_PROTCTL[31]) to 1 to enable UART protocol.
5. Transmit and receive data.
  - Write transmit data register UUART\_TXDAT to transmit data.
  - Wait until TXSTIF(UUART\_PROTSTS[1]) is set and then user can write the next data in UUART\_TXDAT.
  - When TXENDIF(UUART\_PROTSTS[2]) is set, the transmit buffer is empty and the stop bit of the last data has been transmitted.
  - If RXENDIF(UUART\_PROTSTS[4]) is set, the receiver has finished a data frame completely. User can get the data by reading receive data register UUART\_RXDAT.

### 6.20.6 Register Map

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

Register	Offset	R/W	Description	Reset Value
<b>UART Base Address:</b> <b>UARTx_BA = 0x4007_0000 + (0x10_0000 * x)</b> <b>x = 0, 1</b> <b>UART2_BA = 0x4007_4000</b>				
<b>UART_CTL</b>	UARTx_BA+0x00	R/W	USCI Control Register	0x0000_0000
<b>UART_INTEN</b>	UARTx_BA+0x04	R/W	USCI Interrupt Enable Register	0x0000_0000
<b>UART_BRGEN</b>	UARTx_BA+0x08	R/W	USCI Baud Rate Generator Register	0x0000_3C00
<b>UART_DATIN0</b>	UARTx_BA+0x10	R/W	USCI Input Data Signal Configuration Register 0	0x0000_0000
<b>UART_CTLIN0</b>	UARTx_BA+0x20	R/W	USCI Input Control Signal Configuration Register 0	0x0000_0000
<b>UART_CLKIN</b>	UARTx_BA+0x28	R/W	USCI Input Clock Signal Configuration Register	0x0000_0000
<b>UART_LINECTL</b>	UARTx_BA+0x2C	R/W	USCI Line Control Register	0x0000_0000
<b>UART_TXDAT</b>	UARTx_BA+0x30	W	USCI Transmit Data Register	0x0000_0000
<b>UART_RXDAT</b>	UARTx_BA+0x34	R	USCI Receive Data Register	0x0000_0000
<b>UART_BUFCTL</b>	UARTx_BA+0x38	R/W	USCI Transmit/Receive Buffer Control Register	0x0000_0000
<b>UART_BUFSTS</b>	UARTx_BA+0x3C	R	USCI Transmit/Receive Buffer Status Register	0x0000_0101
<b>UART_WKCTL</b>	UARTx_BA+0x54	R/W	USCI Wake-up Control Register	0x0000_0000
<b>UART_WKSTS</b>	UARTx_BA+0x58	R/W	USCI Wake-up Status Register	0x0000_0000
<b>UART_PROTCTL</b>	UARTx_BA+0x5C	R/W	USCI Protocol Control Register	0x0000_0000
<b>UART_PROTIEN</b>	UARTx_BA+0x60	R/W	USCI Protocol Interrupt Enable Register	0x0000_0000
<b>UART_PROTSTS</b>	UARTx_BA+0x64	R/W	USCI Protocol Status Register	0x0000_0000

6.20.7 Register Description

USCI Control Register (UART\_CTL)

Register	Offset	R/W	Description	Reset Value
UART_CTL	UARTx_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]	<b>Reserved</b> Reserved.
[2:0]	<p><b>FUNMODE</b></p> <p><b>Function Mode</b> 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.</p> <p>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<sup>2</sup>C protocol is selected.</p> <p><b>Note:</b> Other bit combinations are reserved.</p>

**USCI Interrupt Enable Register (UART\_INTEN)**

Register	Offset	R/W	Description	Reset Value
UART_INTEN	UARTx_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]	<b>Reserved</b> Reserved.
[4]	<b>RXENDIEN</b> <b>Receive End Interrupt Enable Bit</b> 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]	<b>RXSTIEN</b> <b>Receive Start Interrupt Enable Bit</b> 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]	<b>TXENDIEN</b> <b>Transmit End Interrupt Enable Bit</b> 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]	<b>TXSTIEN</b> <b>Transmit Start Interrupt Enable Bit</b> 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]	<b>Reserved</b> Reserved.

**USCI Baud Rate Generator Register (UART\_BRGEN)**

Register	Offset	R/W	Description	Reset Value
UART_BRGEN	UARTx_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	<p><b>Clock Divider</b> This bit field defines the ratio between the protocol clock frequency <math>f_{PROT\_CLK}</math> and the clock divider frequency <math>f_{DIV\_CLK}</math> (<math>f_{DIV\_CLK} = f_{PROT\_CLK} / (CLKDIV+1)</math> ).</p> <p><b>Note:</b> In UART function, it can be updated by hardware in the 4<sup>th</sup> 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 BRDETITV (UART_PROTCTL[24:16]) to calculate the precise baud rate.</p>
[15]	Reserved	Reserved.
[14:10]	DSCNT	<p><b>Denominator for Sample Counter</b> This bit field defines the divide ratio of the sample clock <math>f_{SAMP\_CLK}</math>. The divided frequency <math>f_{DS\_CNT} = f_{PDS\_CNT} / (DSCNT+1)</math>.</p> <p><b>Note:</b> 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.</p>
[9:8]	PDSCNT	<p><b>Pre-divider for Sample Counter</b> This bit field defines the divide ratio of the clock division from sample clock <math>f_{SAMP\_CLK}</math>. The divided frequency <math>f_{PDS\_CNT} = f_{SAMP\_CLK} / (PDSCNT+1)</math>.</p>
[7:6]	Reserved	Reserved.
[5]	TMCNTSRC	<p><b>Timing Measurement Counter Clock Source Selection</b> 0 = Timing measurement counter with <math>f_{PROT\_CLK}</math>. 1 = Timing measurement counter with <math>f_{DIV\_CLK}</math>.</p>
[4]	TMCNTEN	<p><b>Timing Measurement Counter Enable Bit</b> This bit enables the 10-bit timing measurement counter. 0 = Timing measurement counter Disabled. 1 = Timing measurement counter Enabled.</p>
[3:2]	SPCLKSEL	<p><b>Sample Clock Source Selection</b> This bit field used for the clock source selection of a sample clock (<math>f_{SAMP\_CLK}</math>) for the</p>



		<p>protocol processor.</p> <p>00 = <math>f_{SAMP\_CLK} = f_{DIV\_CLK}</math>.</p> <p>01 = <math>f_{SAMP\_CLK} = f_{PROT\_CLK}</math>.</p> <p>10 = <math>f_{SAMP\_CLK} = f_{SCLK}</math>.</p> <p>11 = <math>f_{SAMP\_CLK} = f_{REF\_CLK}</math>.</p>
[1]	<b>PTCLKSEL</b>	<p><b>Protocol Clock Source Selection</b></p> <p>This bit selects the source signal of protocol clock (<math>f_{PROT\_CLK}</math>).</p> <p>0 = Reference clock <math>f_{REF\_CLK}</math>.</p> <p>1 = <math>f_{REF\_CLK2}</math> (its frequency is half of <math>f_{REF\_CLK}</math>).</p>
[0]	<b>RCLKSEL</b>	<p><b>Reference Clock Source Selection</b></p> <p>This bit selects the source signal of reference clock (<math>f_{REF\_CLK}</math>).</p> <p>0 = Peripheral device clock <math>f_{PCLK}</math>.</p> <p>1 = Reserved.</p>

**USCI Input Data Signal Configuration (UART\_DATIN0)**

Register	Offset	R/W	Description	Reset Value
UART_DATIN0	UARTx_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	<p><b>Input Signal Edge Detection Mode</b></p> <p>This bit field selects which edge activates the trigger event of input data signal.</p> <p>00 = The trigger event activation is disabled.</p> <p>01 = A rising edge activates the trigger event of input data signal.</p> <p>10 = A falling edge activates the trigger event of input data signal.</p> <p>11 = Both edges activate the trigger event of input data signal.</p> <p><b>Note:</b> In UART function mode, it is suggested to set this bit field as 10.</p>
[2]	ININV	<p><b>Input Signal Inverse Selection</b></p> <p>This bit defines the inverter enable of the input asynchronous signal.</p> <p>0 = The un-synchronized input signal will not be inverted.</p> <p>1 = The un-synchronized input signal will be inverted.</p>
[1]	Reserved	Reserved.
[0]	SYNCSEL	<p><b>Input Signal Synchronization Selection</b></p> <p>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.</p> <p>0 = The un-synchronized signal can be taken as input for the data shift unit.</p> <p>1 = The synchronized signal can be taken as input for the data shift unit.</p>

**USCI Input Control Signal Configuration (UART\_CTLIN0)**

Register	Offset	R/W	Description	Reset Value
UART_CTLIN0	UARTx_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	<b>Input Signal Inverse Selection</b> 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	<b>Input Synchronization Signal Selection</b> 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 (UART\_CLKIN)**

Register	Offset	R/W	Description	Reset Value
UART_CLKIN	UARTx_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	<p><b>Input Synchronization Signal Selection</b></p> <p>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.</p> <p>0 = The un-synchronized signal can be taken as input for the data shift unit.</p> <p>1 = The synchronized signal can be taken as input for the data shift unit.</p>

**USCI Line Control Register (UART\_LINECTL)**

Register	Offset	R/W	Description	Reset Value
UART_LINECTL	UARTx_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	<p><b>Word Length of Transmission</b></p> <p>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.</p> <p>0x0: The data word contains 16 bits located at bit positions [15:0].</p> <p>0x1: Reserved.</p> <p>0x2: Reserved.</p> <p>0x3: Reserved.</p> <p>0x4: The data word contains 4 bits located at bit positions [3:0].</p> <p>0x5: The data word contains 5 bits located at bit positions [4:0].</p> <p>...</p> <p>0xF: The data word contains 15 bits located at bit positions [14:0].</p> <p><b>Note:</b> In UART protocol, the length can be configured as 6~13 bits.</p>
[7]	CTLOINV	<p><b>Control Signal Output Inverse Selection</b></p> <p>This bit defines the relation between the internal control signal and the output control signal.</p> <p>0 = No effect.</p> <p>1 = The control signal will be inverted before its output.</p> <p><b>Note:</b> In UART protocol, the control signal means nRTS signal.</p>
[6]	Reserved	Reserved.
[5]	DATOINV	<p><b>Data Output Inverse Selection</b></p> <p>This bit defines the relation between the internal shift data value and the output data signal of USC1x_DAT1 pin.</p> <p>0 = The value of USC1x_DAT1 is equal to the data shift register.</p> <p>1 = The value of USC1x_DAT1 is the inversion of data shift register.</p>
[4:1]	Reserved	Reserved.

[0]	LSB	<p><b>LSB First Transmission Selection</b></p> <p>0 = The MSB, which bit of transmit/receive data buffer depends on the setting of DWIDTH, is transmitted/received first.</p> <p>1 = The LSB, the bit 0 of data buffer, will be transmitted/received first.</p>
-----	-----	---

**USCI Transmit Data Register (UART\_TXDAT)**

Register	Offset	R/W	Description	Reset Value
UART_TXDAT	UARTx_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	<b>Transmit Data</b> 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	UARTx_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	<p><b>Received Data</b> This bit field monitors the received data which stored in receive data buffer.</p> <p><b>Note:</b> RXDAT[15:13] indicate the same frame status of BREAK, FRMERR and PARITYERR (UART_PROTSTS[7:5]).</p>



**USCI Transmitter/Receive Buffer Control Register (UART\_BUFCTL)**

Register	Offset	R/W	Description	Reset Value
UART_BUFCTL	UARTx_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	Description
[31:18]	Reserved	Reserved.
[17]	RXRST	<p><b>Receive Reset</b>                      0 = No effect.                      1 = Reset the receive-related counters, state machine, and the content of receive shift register and data buffer.  <b>Note1:</b> It is cleared automatically after one PCLK cycle.  <b>Note2:</b> It is suggest to check the RXBUSY (UART_PROTSTS[10]) before this bit will be set to 1.</p>
[16]	TXRST	<p><b>Transmit Reset</b>                      0 = No effect.                      1 = Reset the transmit-related counters, state machine, and the content of transmit shift register and data buffer.  <b>Note:</b> It is cleared automatically after one PCLK cycle.</p>
[15]	RXCLR	<p><b>Clear Receive Buffer</b>                      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.  <b>Note:</b> It is cleared automatically after one PCLK cycle.</p>
[14]	RXOVLEN	<p><b>Receive Buffer Overrun Error Interrupt Enable Bit</b>                      0 = Receive overrun interrupt Disabled.                      1 = Receive overrun interrupt Enabled.</p>
[13:8]	Reserved	Reserved.
[7]	TXCLR	<p><b>Clear Transmit Buffer</b>                      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.  <b>Note:</b> It is cleared automatically after one PCLK cycle.</p>

[6:0]	Reserved	Reserved.
-------	----------	-----------

**USCI Transmit/Receive Buffer Status Register (UUART\_BUFSTS)**

Register	Offset	R/W	Description	Reset Value
UUART_BUFSTS	UUARTx_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						TXFULL	TXEMPTY
7	6	5	4	3	2	1	0
Reserved				RXOVIF	Reserved	RXFULL	RXEMPTY

Bits	Description
[31:10]	<b>Reserved</b> Reserved.
[9]	<b>TXFULL</b> <b>Transmit Buffer Full Indicator</b> 0 = Transmit buffer is not full. 1 = Transmit buffer is full.
[8]	<b>TXEMPTY</b> <b>Transmit Buffer Empty Indicator</b> 0 = Transmit buffer is not empty. 1 = Transmit buffer is empty.
[7:4]	<b>Reserved</b> Reserved.
[3]	<b>RXOVIF</b> <b>Receive Buffer Over-run Error Interrupt Status</b> This bit indicates that a receive buffer overrun error event has been detected. If RXOVIEN (UUART_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 error event has not been detected. 1 = A receive buffer overrun error event has been detected.
[2]	<b>Reserved</b> Reserved.
[1]	<b>RXFULL</b> <b>Receive Buffer Full Indicator</b> 0 = Receive buffer is not full. 1 = Receive buffer is full.
[0]	<b>RXEMPTY</b> <b>Receive Buffer Empty Indicator</b> 0 = Receive buffer is not empty. 1 = Receive buffer is empty.

**USCI Wake-up Control Register (UART\_WKCTL)**

Register	Offset	R/W	Description	Reset Value
UART_WKCTL	UARTx_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	<p><b>Power Down Blocking Option</b></p> <p>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.</p> <p>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.</p>
[1]	Reserved	Reserved.
[0]	WKEN	<p><b>Wake-up Enable Bit</b></p> <p>0 = Wake-up function Disabled.</p> <p>1 = Wake-up function Enabled.</p>

**USCI Wake-up Status Register (UART\_WKSTS)**

Register	Offset	R/W	Description	Reset Value
UART_WKSTS	UARTx_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	<b>Wake-up Flag</b> 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 (UART\_PROTCTL)**

Register	Offset	R/W	Description	Reset Value
UUART_PROTCTL	UUARTx_BA+0x5C	R/W	USCI Protocol Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PROTEN	Reserved	BCEN	Reserved	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	<b>UART Protocol Enable Bit</b> 0 = UART Protocol Disabled. 1 = UART Protocol Enabled.
[30]	Reserved	Reserved.
[29]	BCEN	<b>Transmit Break Control Enable Bit</b> 0 = Transmit Break Control Disabled. 1 = Transmit Break Control Enabled. <b>Note:</b> 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 transmitter logic.
[27]	Reserved	Reserved.
[26]	STICKEN	<b>Stick Parity Enable Bit</b> 0 = Stick parity Disabled. 1 = Stick parity Enabled. <b>Note:</b> Refer to RS-485 Support section for detail information.
[25]	Reserved	Reserved.
[24:16]	BRDETITV	<b>Baud Rate Detection Interval</b> 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. <b>Note:</b> This bit can be cleared to 0 by software writing '0' to the BRDETITV.
[15]	Reserved	Reserved.
[14:11]	WAKECNT	<b>Wake-up Counter</b> These bits field indicate how many clock cycle selected by f <sub>PDS_CNT</sub> do the slave can get the 1 <sup>st</sup> bit (start bit) when the device is wake-up from Power-down mode.
[10]	CTSWKEN	<b>nCTS Wake-up Mode Enable Bit</b>

		0 = nCTS wake-up mode Disabled. 1 = nCTS wake-up mode Enabled.
[9]	<b>DATWKEN</b>	<b>Data Wake-up Mode Enable Bit</b> 0 = Data wake-up mode Disabled. 1 = Data wake-up mode Enabled.
[6]	<b>ABREN</b>	<b>Auto-baud Rate Detect Enable Bit</b> 0 = Auto-baud rate detect function Disabled. 1 = Auto-baud rate detect function Enabled. <b>Note:</b> When the auto - baud rate detect operation finishes, hardware will clear this bit. The associated interrupt ABRDETIF (USCI_PROTST[9]) will be generated (If ARBIEN (UUART_PROTIEN [1]) is enabled).
[5]	<b>RTSAUDIREN</b>	<b>nRTS Auto Direction Enable Bit</b> When nRTS auto direction is enabled, if the transmitted bytes in the TX buffer is empty, the UART will reassert nRTS signal. 0 = nRTS auto direction control Disabled. 1 = nRTS auto direction control Enabled. <b>Note1:</b> This bit is used for nRTS auto direction control for RS485. <b>Note2:</b> This bit has effect only when the RTSAUTOEN is not set.
[4]	<b>CTSAUTOEN</b>	<b>nCTS Auto-flow Control Enable Bit</b> 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]	<b>RTSAUTOEN</b>	<b>nRTS Auto-flow Control Enable Bit</b> When nRTS auto-flow is enabled, if the receiver buffer is full (RXFULL (UUART_BUFSTS[1] = 1'b1)), the UART will de-assert nRTS signal. 0 = nRTS auto-flow control Disabled. 1 = nRTS auto-flow control Enabled. <b>Note:</b> This bit has effect only when the RTSAUDIREN is not set.
[2]	<b>EVENPARITY</b>	<b>Even Parity Enable Bit</b> 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. <b>Note:</b> This bit has effect only when PARITYEN is set.
[1]	<b>PARITYEN</b>	<b>Parity Enable Bit</b> This bit defines the parity bit is enabled in an UART frame. 0 = The parity bit Disabled. 1 = The parity bit Enabled.
[0]	<b>STOPB</b>	<b>Stop Bits</b> 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 (UUART\_PROTIEN)**

Register	Offset	R/W	Description	Reset Value
UUART_PROTIEN	UUARTx_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]	<b>Reserved</b> Reserved.
[2]	<b>RLSIEN</b> <b>Receive Line Status Interrupt Enable Bit</b> 0 = Receive line status interrupt Disabled. 1 = Receive line status interrupt Enabled. <b>Note:</b> UUART_PROTSTS[7:5] indicates the current interrupt event for receive line status interrupt.
[1]	<b>ABRIEN</b> <b>Auto-baud Rate Interrupt Enable Bit</b> 0 = Auto-baud rate interrupt Disabled. 1 = Auto-baud rate interrupt Enabled.
[0]	<b>Reserved</b> Reserved.



**USCI Protocol Status Register – UART (UUART\_PROTSTS)**

Register	Offset	R/W	Description	Reset Value
UUART_PROTSTS	UUARTx_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	Description
[31:18]	Reserved	Reserved.
[17]	CTSLV	<b>nCTS Pin Status (Read Only)</b> This bit is 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	<b>nCTS Synchronized Level Status (Read Only)</b> This bit is 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	<b>Auto-baud Rate Error Status</b> 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. <b>Note1:</b> This bit is set at the same time of ABRDETIF. <b>Note2:</b> This bit can be cleared by writing "1" to ABRDETIF or ABERRSTS.
[10]	RXBUSY	<b>RX Bus Status Flag (Read Only)</b> This bit indicates the busy status of the receiver. 0 = The receiver is Idle. 1 = The receiver is BUSY.
[9]	ABRDETIF	<b>Auto-baud Rate Interrupt Flag</b> This bit is set when auto-baud rate detection is done among the falling edge of the input

		<p>data. If the ABRIEN (UART_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.</p> <p>0 = Auto-baud rate detect function is not done. 1 = One Bit auto-baud rate detect function is done.</p> <p><b>Note:</b> This bit can be cleared by writing "1" to it.</p>
[8]	<b>Reserved</b>	Reserved.
[7]	<b>BREAK</b>	<p><b>Break Flag</b></p> <p>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).</p> <p>0 = No Break is generated. 1 = Break is generated in the receiver bus.</p> <p><b>Note:</b> This bit can be cleared by write "1" among the BREAK, FRMERR and PARITYERR bits.</p>
[6]	<b>FRMERR</b>	<p><b>Framing Error Flag</b></p> <p>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).</p> <p>0 = No framing error is generated. 1 = Framing error is generated.</p> <p><b>Note:</b> This bit can be cleared by write "1" among the BREAK, FRMERR and PARITYERR bits.</p>
[5]	<b>PARITYERR</b>	<p><b>Parity Error Flag</b></p> <p>This bit is set to logic 1 whenever the received character does not have a valid "parity bit".</p> <p>0 = No parity error is generated. 1 = Parity error is generated.</p> <p><b>Note:</b> This bit can be cleared by write "1" among the BREAK, FRMERR and PARITYERR bits.</p>
[4]	<b>RXENDIF</b>	<p><b>Receive End Interrupt Flag</b></p> <p>0 = A receive finish interrupt status has not occurred. 1 = A receive finish interrupt status has occurred.</p> <p><b>Note:</b> It is cleared by software writing 1 into this bit.</p>
[3]	<b>RXSTIF</b>	<p><b>Receive Start Interrupt Flag</b></p> <p>0 = A receive start interrupt status has not occurred. 1 = A receive start interrupt status has occurred.</p> <p><b>Note:</b> It is cleared by software writing 1 into this bit.</p>
[2]	<b>TXENDIF</b>	<p><b>Transmit End Interrupt Flag</b></p> <p>0 = A transmit end interrupt status has not occurred. 1 = A transmit end interrupt status has occurred.</p> <p><b>Note:</b> It is cleared by software writing 1 into this bit.</p>
[1]	<b>TXSTIF</b>	<p><b>Transmit Start Interrupt Flag</b></p> <p>0 = A transmit start interrupt status has not occurred. 1 = A transmit start interrupt status has occurred.</p> <p><b>Note1:</b> It is cleared by software writing 1 into this bit. <b>Note2:</b> Used for user to load next transmit data when there is no data in transmit buffer.</p>
[0]	<b>Reserved</b>	Reserved.

## 6.21 USCI - SPI Mode

### 6.21.1 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.

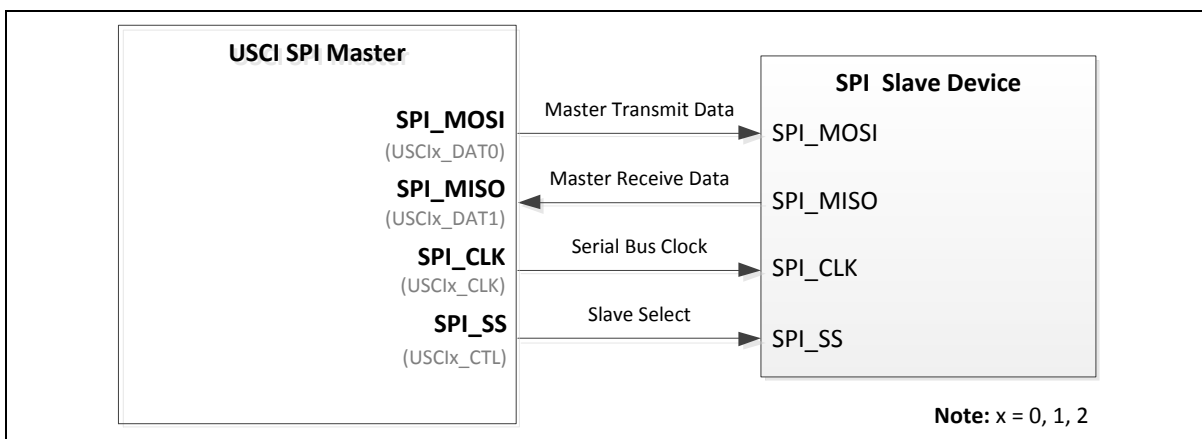


Figure 6.21-1 SPI Master Mode Application Block Diagram

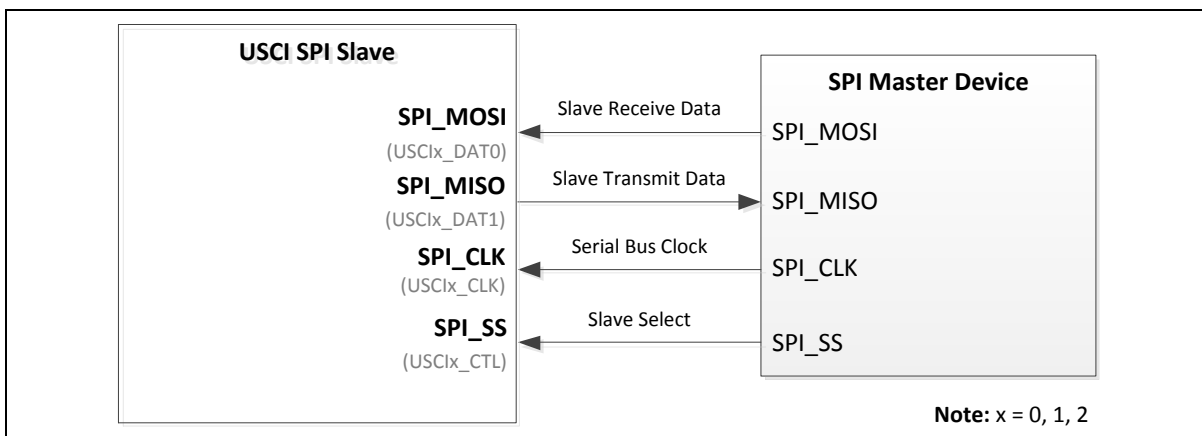


Figure 6.21-2 SPI Slave Mode Application Block Diagram

### 6.21.2 Features

- Supports Master or Slave mode operation (the maximum frequency -- Master =  $f_{PCLK} / 2$ , Slave <  $f_{PCLK} / 5$ )
- 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 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

### 6.21.3 Block Diagram

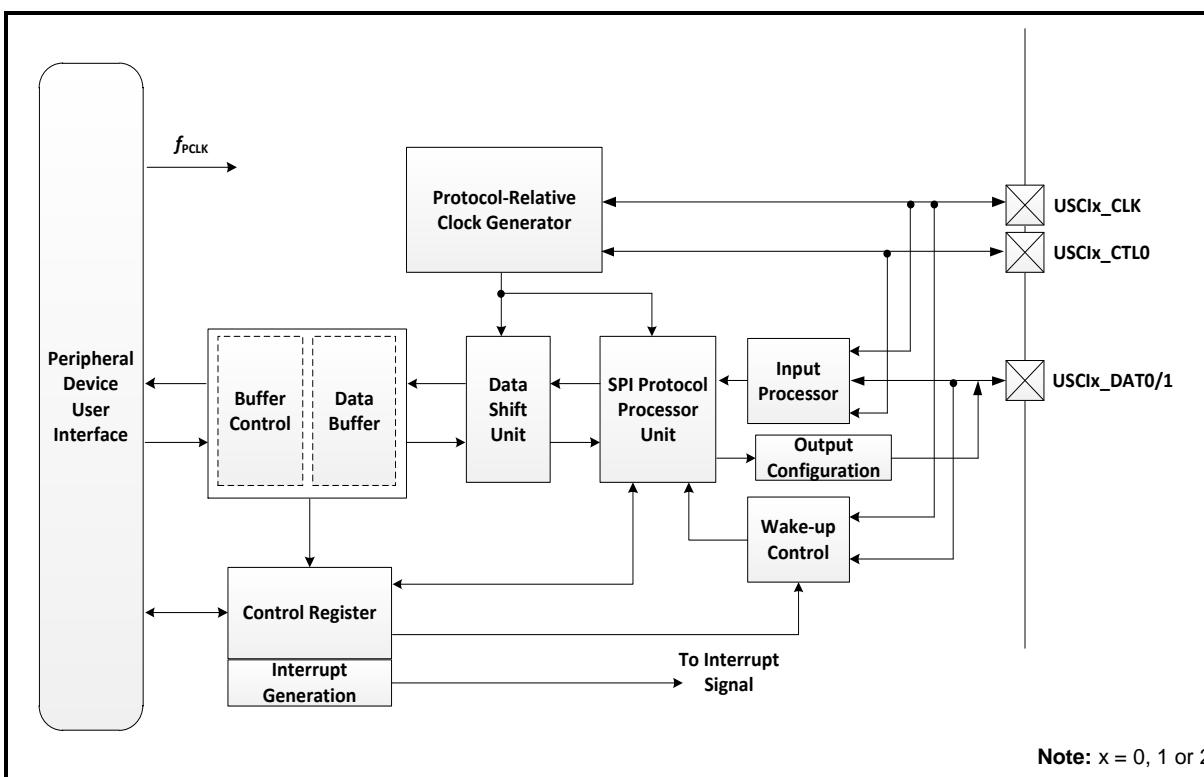


Figure 6.21-3 USCI SPI Mode Block Diagram

### 6.21.4 Basic Configuration

#### 6.21.4.1 Basic Configuration of USCI0 SPI

- Clock Source Configuration
  - Enable USCI0 peripheral clock in USCI0CKEN (CLK\_APBCLK1[8]).
  - Enable USCI0\_SPI functi on USCI\_CTL[2:0] register, USCI\_CTL[2:0]=3'b001
- Reset Configuration
  - Reset USCI0 controller in USCI0RST (SYS\_IPRST2[8]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
USCI0	USCI0_CLK	PE.5	MFP4
		PC.4	MFP5
		PB.9	MFP8

	USCI0_CTL0	PE.4	MFP4
		PC.3	MFP5
		PB.8, PE.2	MFP8
	USCI0_CTL1	PC.2, PC.7	MFP4
		PB.4	MFP8
	USCI0_DAT0	PC.0, PC.5	MFP4
		PB.2	MFP8
	USCI0_DAT1	PC.1, PC.6	MFP4
PB.3		MFP8	

6.21.4.2 Basic Configuration of USCI1 SPI

- Clock Source Configuration
  - Enable USCI1 peripheral clock in USCI1CKEN (CLK\_APBCLK1[9]).
  - Enable USCI1\_SPI functi on USCI\_CTL[2:0] register, USCI\_CTL[2:0]=3'b001
- Reset Configuration
  - Reset USCI1 controller in USCI1RST (SYS\_IPRST2[9]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
USCI1	USCI1_CLK	PD.15	MFP1
		PA.15	MFP4
		PA.3	MFP8
	USCI1_CTL0	PD.12	MFP1
		PA.0	MFP4
		PA.2	MFP8
	USCI1_CTL1	PD.7	MFP1
		PA.1, PA.14	MFP4
	USCI1_DAT0	PD.14	MFP1
		PB.0	MFP6
	USCI1_DAT1	PD.13	MFP1
		PB.1	MFP8

6.21.4.3 Basic Configuration of USCI2 SPI

- Clock Source Configuration
  - Enable USCI2 peripheral clock in USCI2CKEN (CLK\_APBCLK1[10]).
  - Enable USCI2\_SPI functi on USCI\_CTL[2:0] register, USCI\_CTL[2:0]=3'b001

- Reset Configuration
  - Reset USCI2 controller in USCI2RST (SYS\_IPRST2[10]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
USCI2	USCI2_CLK	PC.11, PD.1	MFP4
		PF.2	MFP5
	USCI2_CTL0	PC.12, PD.0, PD.9	MFP4
		PF.1	MFP5
	USCI2_CTL1	PB.7, PC.9, PD.8	MFP4
		PF.0	MFP5
	USCI2_DAT0	PC.13, PD.2	MFP4
		PD.10	MFP5
	USCI2_DAT1	PC.10, PD.3	MFP4
		PD.11	MFP5

### 6.21.5 Functional Description

#### 6.21.5.1 USCI Common Function Description

Please refer to section 6.19.4 for detailed information.

#### 6.21.5.2 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 below.

SPI Mode		Receive Data	Transmit Data	Serial Bus Clock	Slave Select
Full-duplex Master	SPI	SPI_MISO (USCIx_DAT1)	SPI_MOSI (USCIx_DAT0)	SPI_CLK (USCIx_CLK)	SPI_SS (USCIx_CTL0)
Full-duplex Slave	SPI	SPI_MOSI (USCIx_DAT0)	SPI_MISO (USCIx_DAT1)	SPI_CLK (USCIx_CLK)	SPI_SS (USCIx_CTL0)
Half-duplex Master/Slave	SPI	SPI_MOSI (USCIx_DAT0)	SPI_MOSI (USCIx_DAT0)	SPI_CLK (USCIx_CLK)	SPI_SS (USCIx_CTL0)

Table 6.21-1 SPI Communication Signals

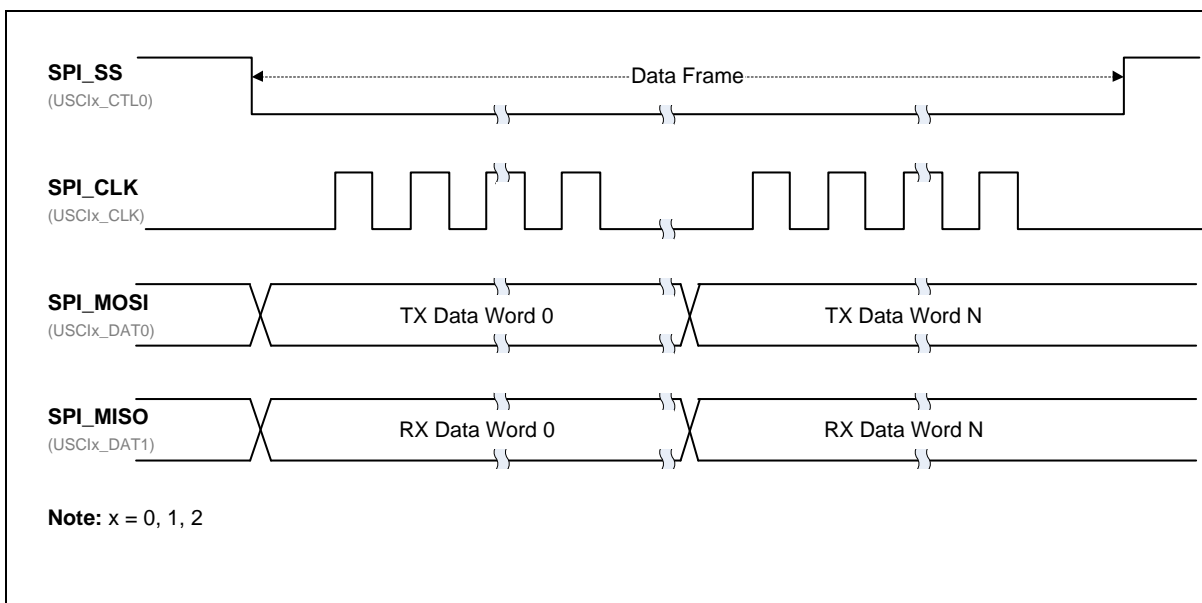


Figure 6.21-4 Wire Full-Duplex SPI Communication Signals (Master Mode)

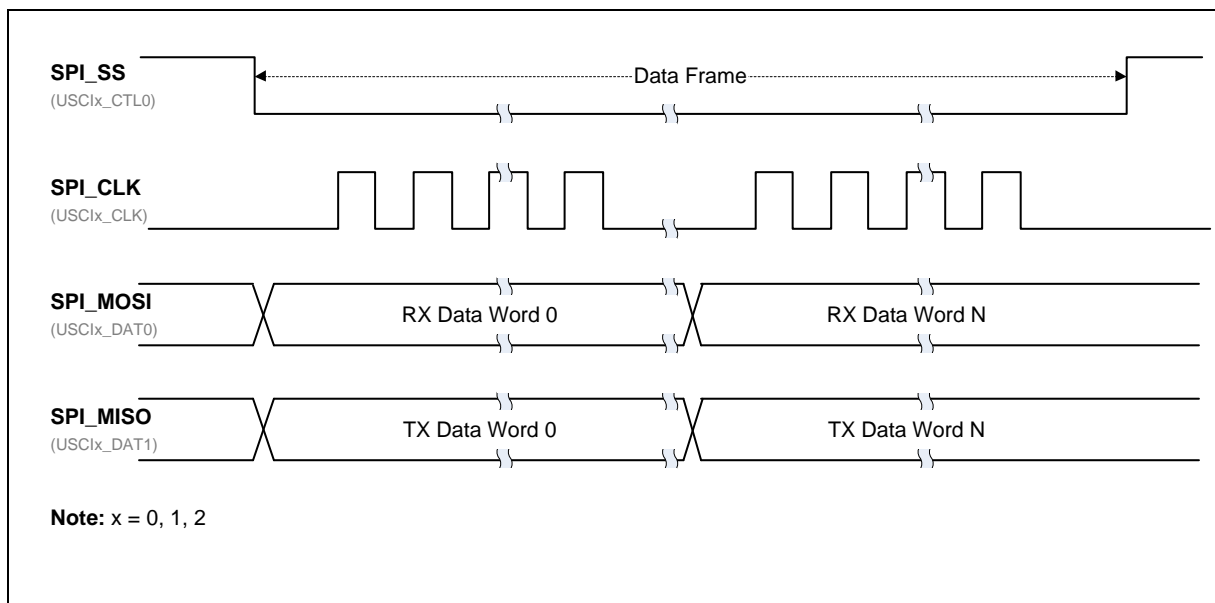


Figure 6.21-5 Wire Full-Duplex SPI Communication Signals (Slave Mode)

### 6.21.5.3 Serial Bus Clock Configuration

The USCI controller needs the peripheral clock to drive the USCI logic unit to perform the data transfer. The peripheral clock frequency is equal to PCLK frequency.

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]). Refer to section 6.19.4.5 for details of protocol-relative clock generator.

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 6.21-2 Serial Bus Clock Configuration

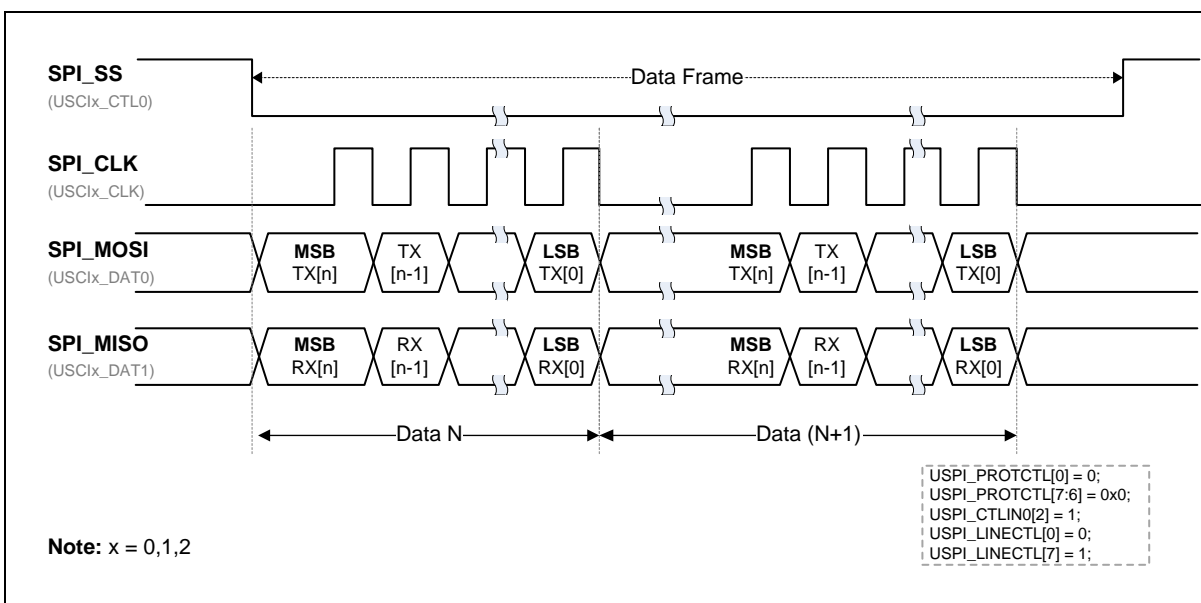


Figure 6.21-6 SPI Communication with Different SPI Clock Configuration (SCLKMODE=0x0)



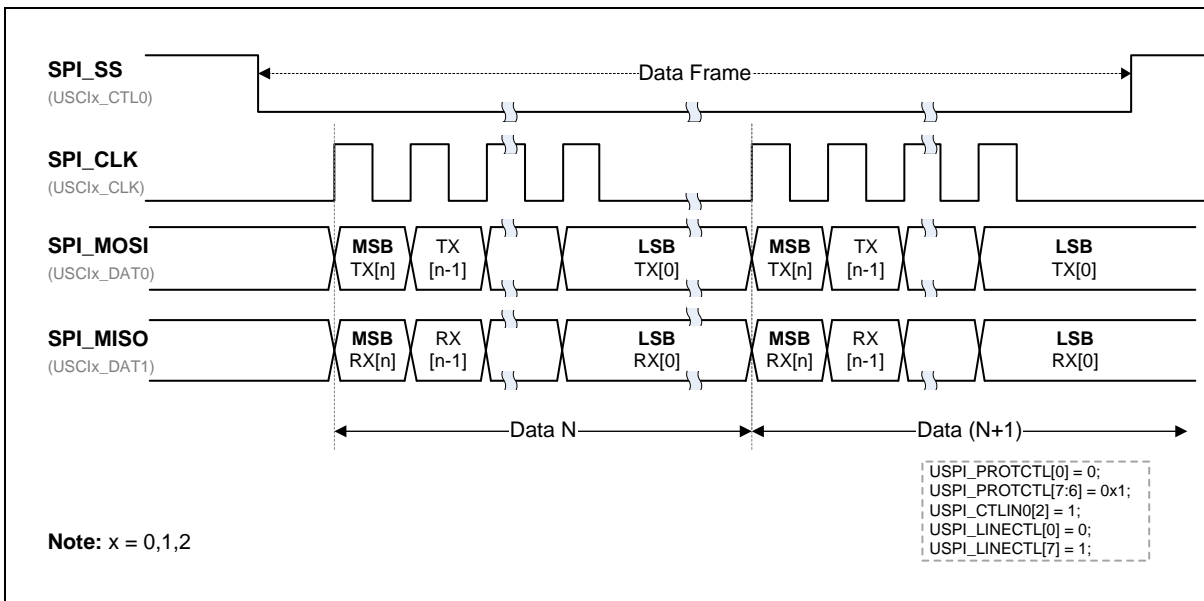


Figure 6.21-7 SPI Communication with Different SPI Clock Configuration (SCLKMODE=0x1)

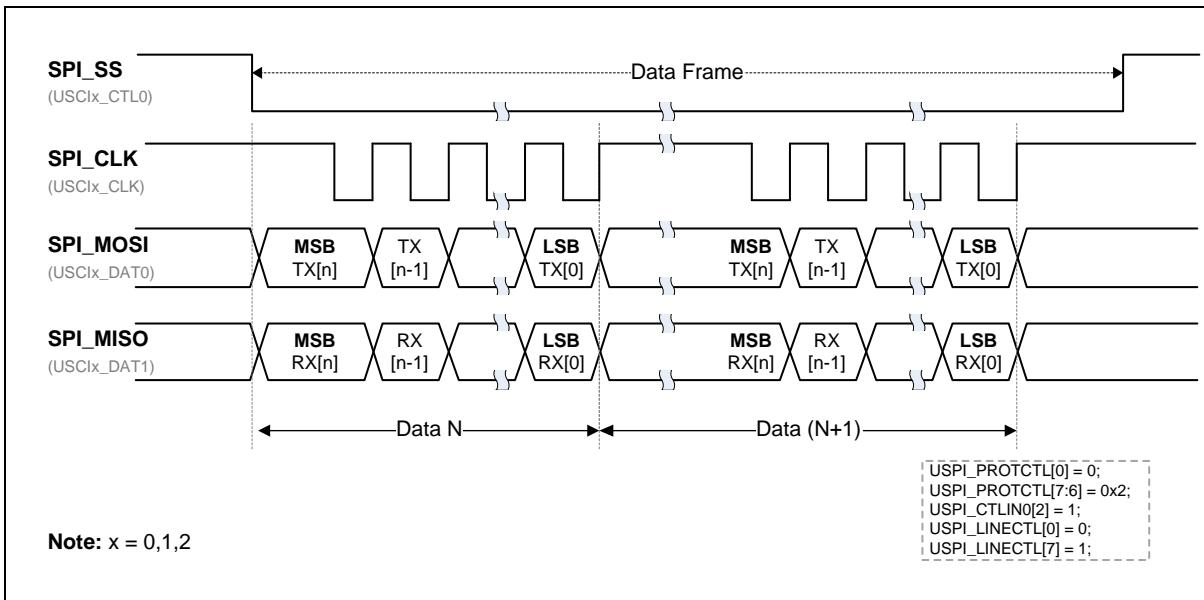


Figure 6.21-8 SPI Communication with Different SPI Clock Configuration (SCLKMODE=0x2)

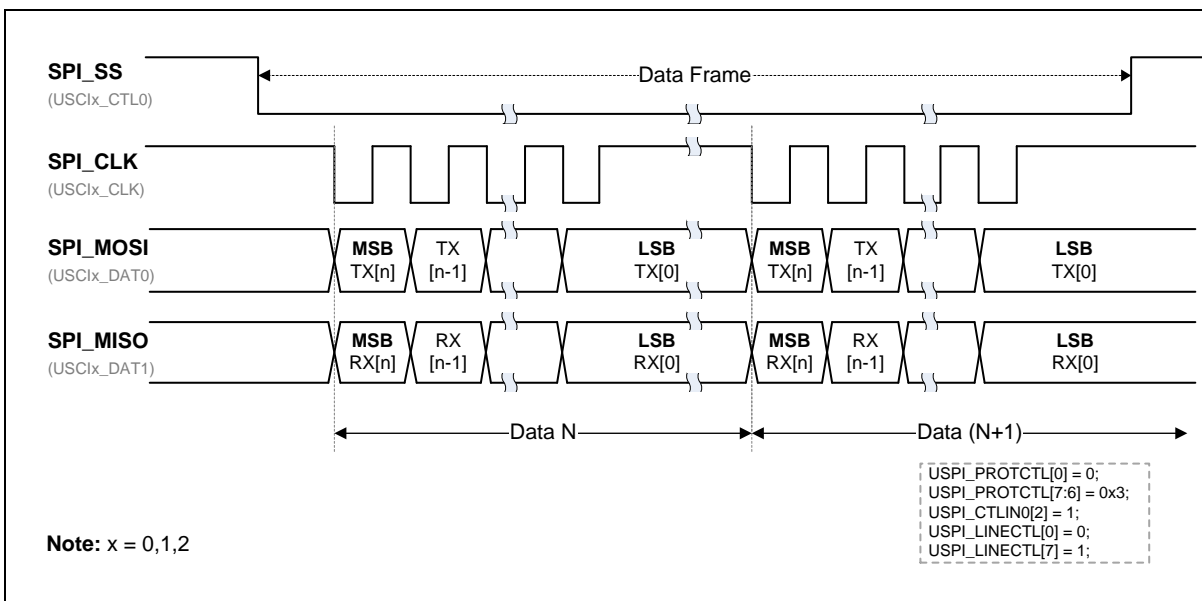


Figure 6.21-9 SPI Communication with Different SPI Clock Configuration (SCLKMODE=0x3)

#### 6.21.5.4 Slave Select Signal

The slave selection signal of SPI protocol is active high by default. In SPI Master mode, the USCI 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 USCI 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 USCI peripheral clock cycles.

The input slave select signal of SPI Slave has to be keep inactive for at least 2 USCI peripheral clock cycles between two consecutive frames in order to correctly detect the end of a frame.

#### 6.21.5.5 Transmit and Receive Data

The bit length of a transmit/receive data word in SPI protocol of USCI 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.

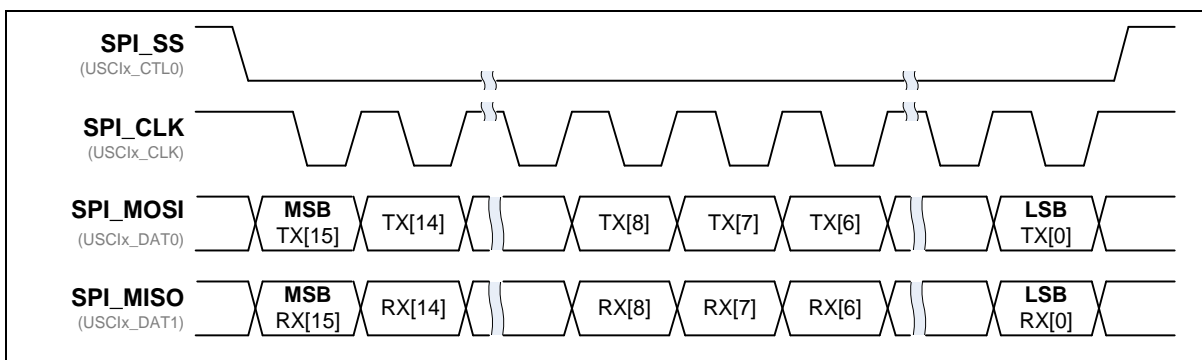


Figure 6.21-10 16-bit data Length in One Word Transaction with MSB First Format

### 6.21.5.6 Word Suspend

SUSPITV (USPI\_PROTCTL[11:8]) provide 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).

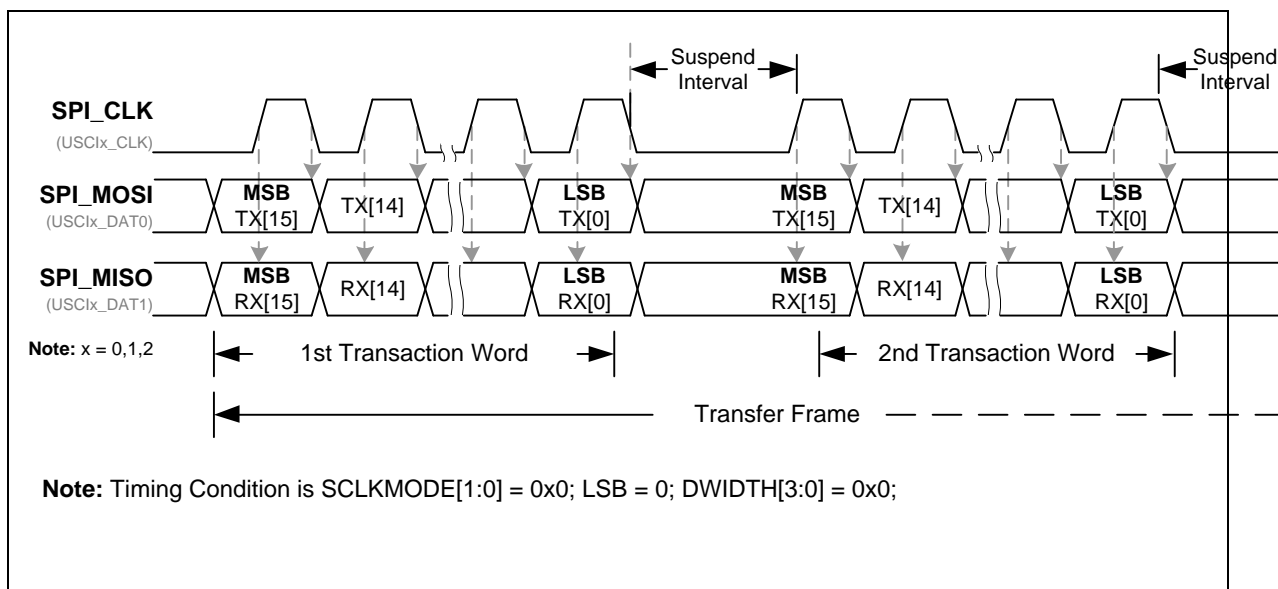


Figure 6.21-11 Word Suspend Interval between Two Transaction Words

### 6.21.5.7 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 USCI 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 great 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.

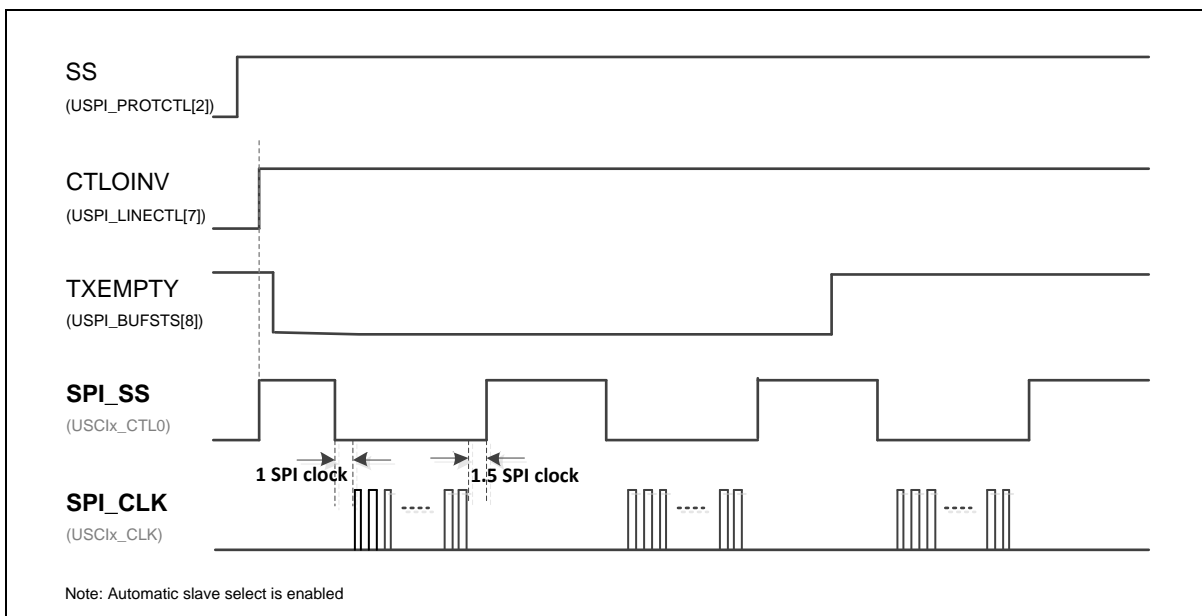


Figure 6.21-12 Auto Slave Select (SUSPITV ≥ 0x3)

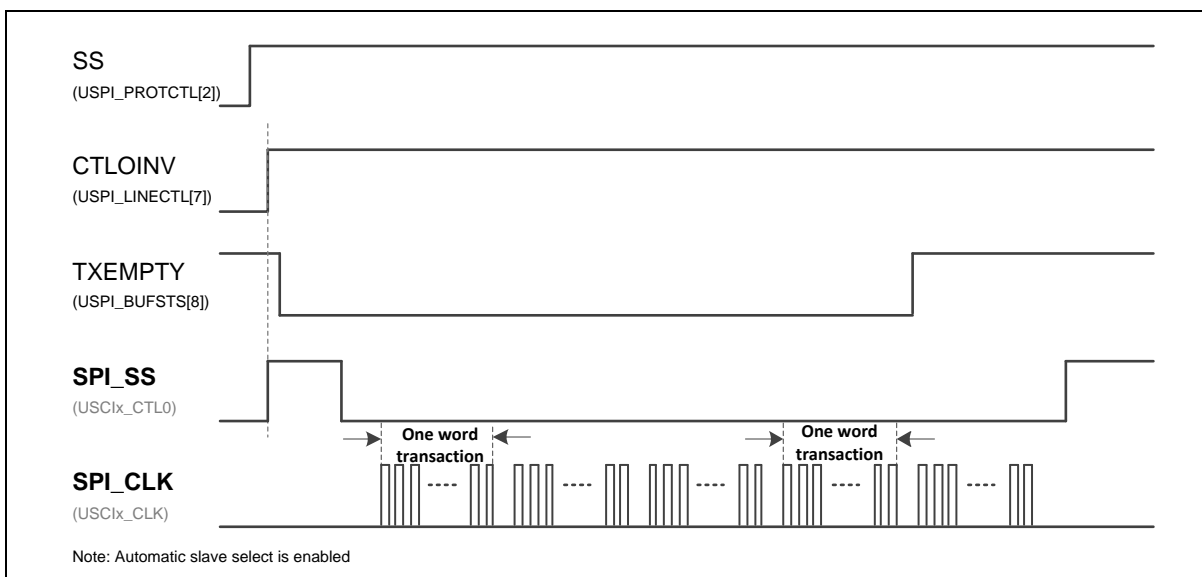


Figure 6.21-13 Auto Slave Select (SUSPITV < 0x3)

### 6.21.5.8 Slave 3-wire Mode

When the SLV3WIRE (USPI\_PROTCTL[1]) is set by software to enable the Slave 3-wire mode, the USC1 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 USC1x\_CLK line), SPI\_MOSI (through USC1x\_DAT0 line), and SPI\_MISO (through USC1x\_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.

#### 6.21.5.9 Data Transfer Mode

The USCI 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 detail 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 6.21-14 and Figure 6.21-15 are the one output data channel and one input data channel half-duplex transfer diagrams with the external device.

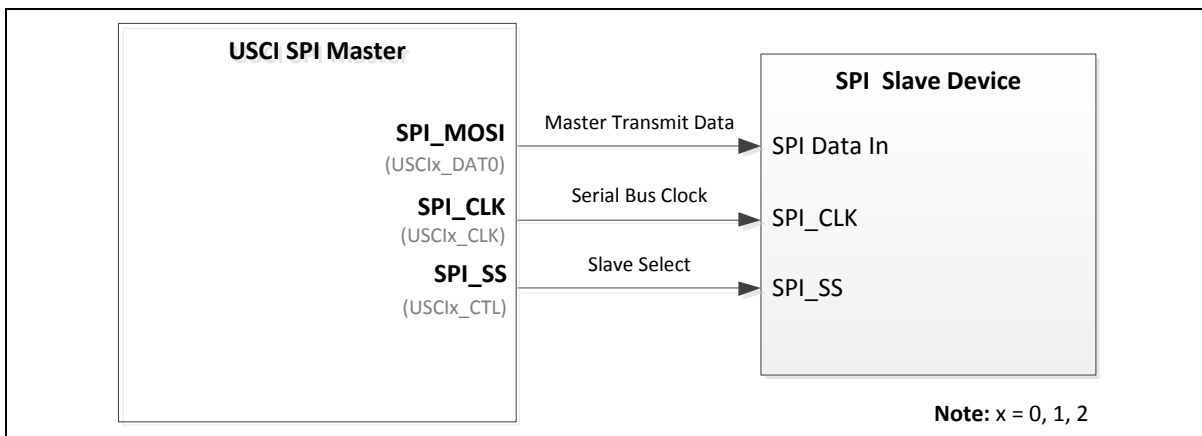


Figure 6.21-14 One Output Data Channel Half-duplex (SPI Master Mode)

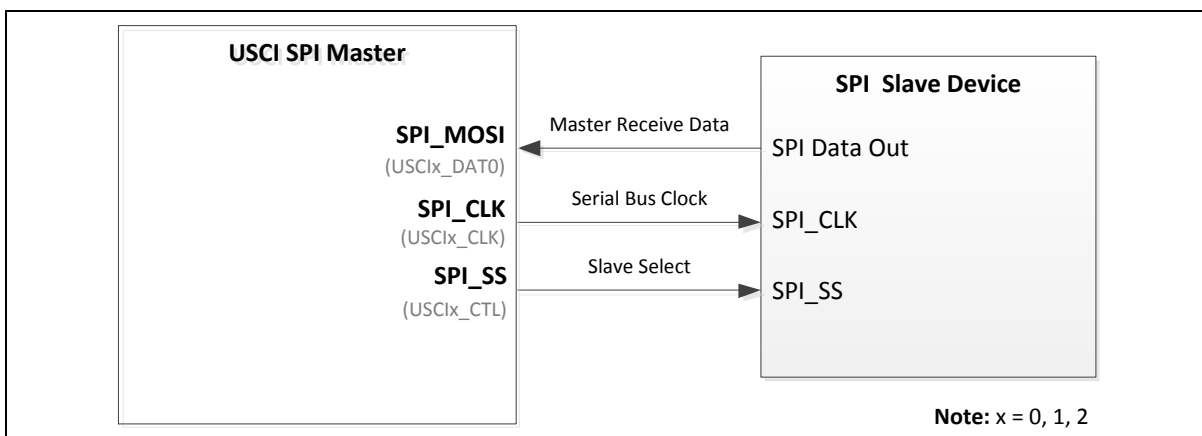


Figure 6.21-15 One Input Data Channel Half-duplex (SPI Master Mode)

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 USCI 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.

#### 6.21.5.10 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 SSINAIEN (USPI\_PROTIEN[0]) or SSACTIEN (USPI\_PROTIEN[1]), 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 USCI 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.

#### 6.21.5.11 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.

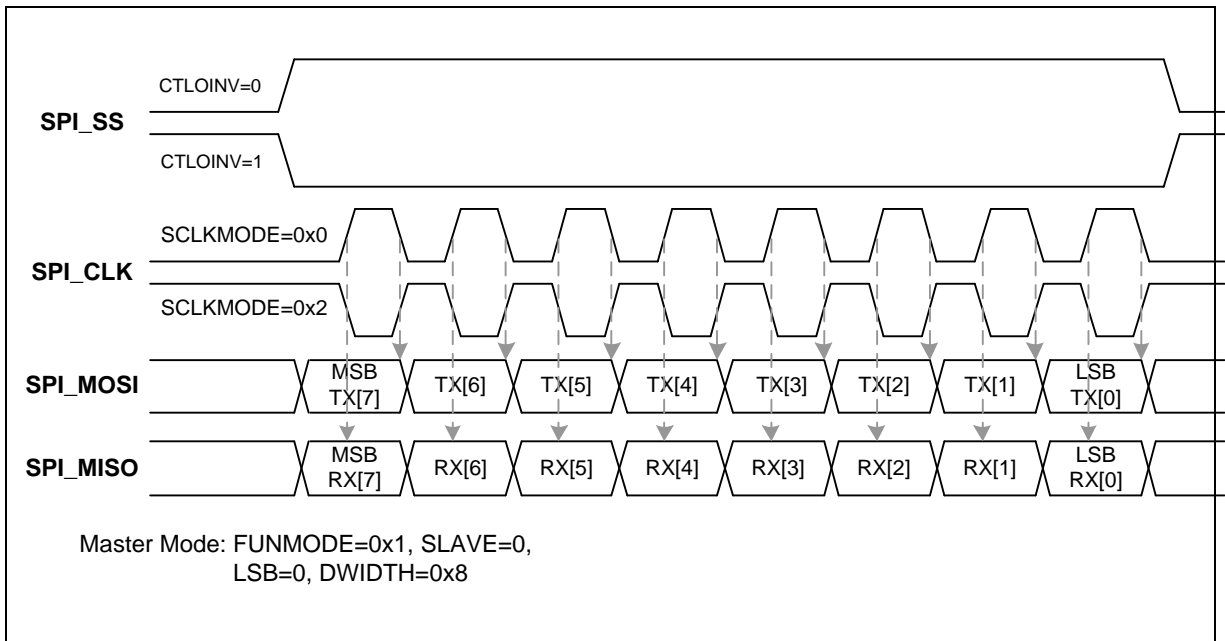


Figure 6.21-16 SPI Timing in Master Mode

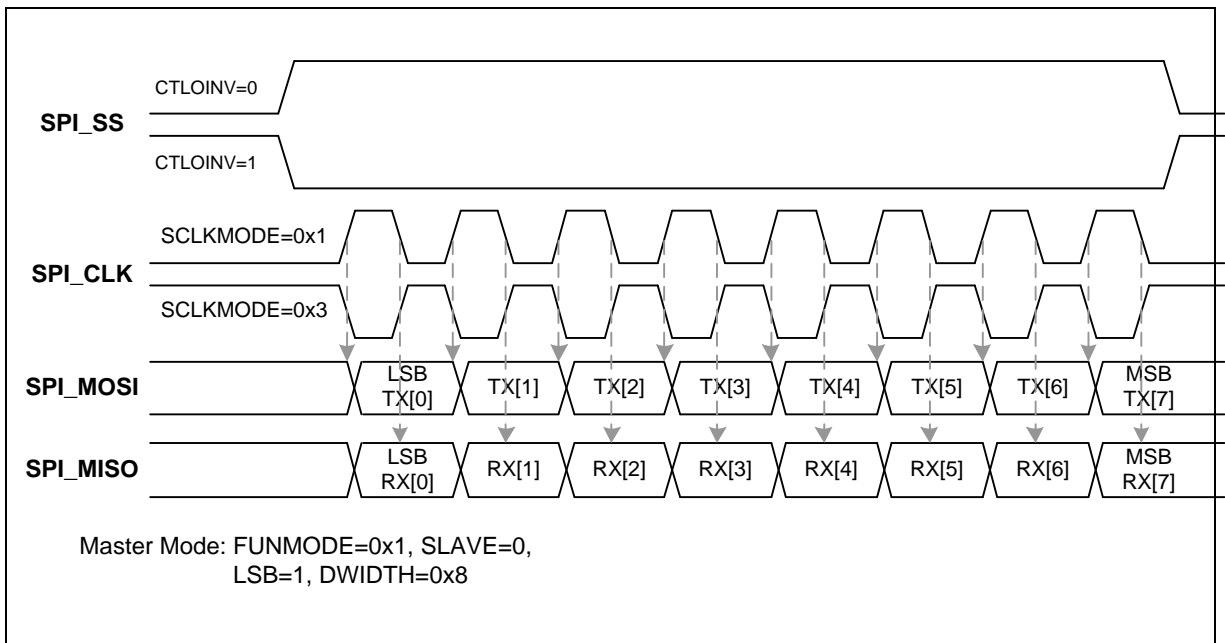


Figure 6.21-17 SPI Timing in Master Mode (Alternate Phase of Serial Bus Clock)



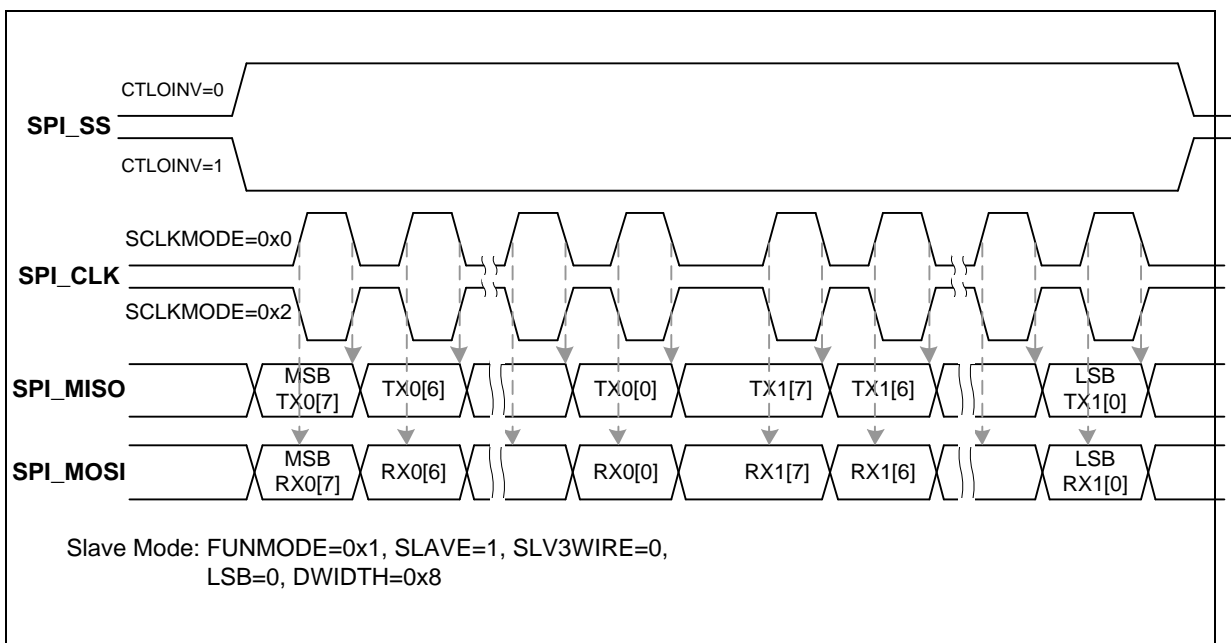


Figure 6.21-18 SPI Timing in Slave Mode

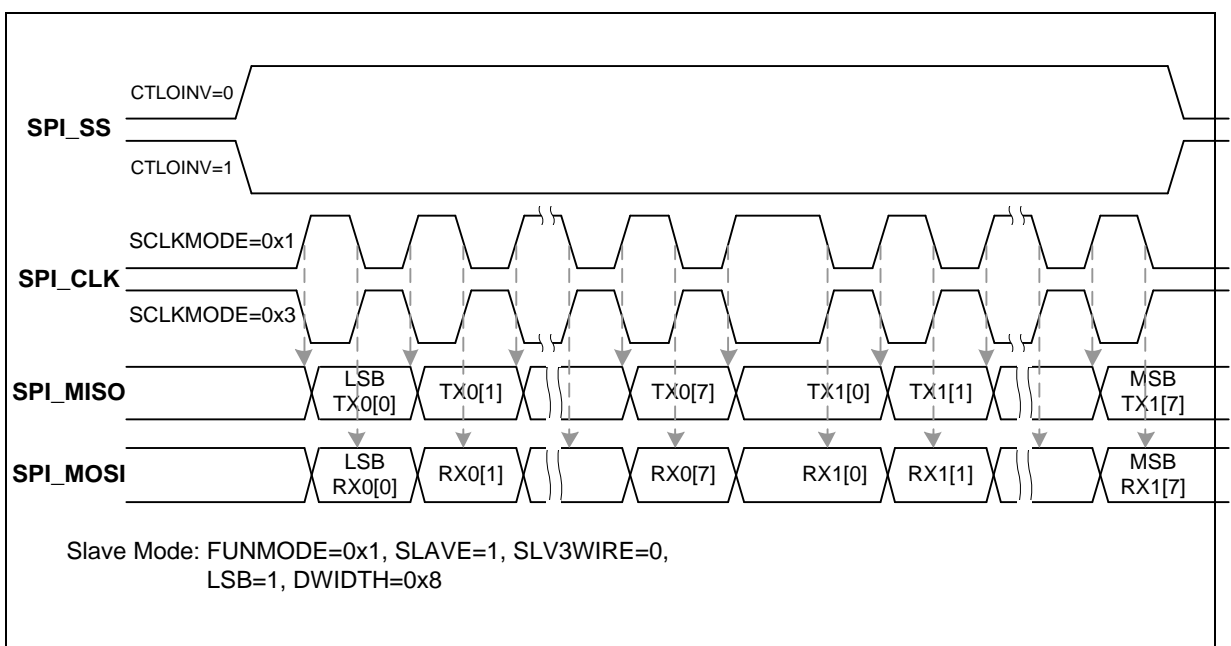


Figure 6.21-19 SPI Timing in Slave Mode (Alternate Phase of Serial Bus Clock)

### 6.21.5.12 Programming Flow

This section describes the programming flow for USCI SPI data transfer.

For Master mode,

1. Enable USCI peripheral clock by setting CLK\_APBCLK1 register.
2. Configure user-specified pins as USCI 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, configured 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.
6. Set PROTEN (USPI\_PROTCTL[31]) to 1 to enable SPI protocol.
7. 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.
8. 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.
9. 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 USCI peripheral clock by setting CLK\_APBCLK1 register.
2. Configure user-specified pins as USCI 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.

#### 6.21.5.13 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.

### 6.21.6 Register Map

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

Register	Offset	R/W	Description	Reset Value
<b>USPI Base Address:</b> <b>USPIx_BA = 0x4007_0000 + (0x10_0000 * x)</b> <b>x = 0, 1</b> <b>USPI2_BA = 0x4007_4000</b>				
USPI_CTL	USPIx_BA+0x00	R/W	USCI Control Register	0x0000_0000
USPI_INTEN	USPIx_BA+0x04	R/W	USCI Interrupt Enable Register	0x0000_0000
USPI_BRGEN	USPIx_BA+0x08	R/W	USCI Baud Rate Generator Register	0x0000_3C00
USPI_DATIN0	USPIx_BA+0x10	R/W	USCI Input Data Signal Configuration Register 0	0x0000_0000
USPI_CTLIN0	USPIx_BA+0x20	R/W	USCI Input Control Signal Configuration Register 0	0x0000_0000
USPI_CLKIN	USPIx_BA+0x28	R/W	USCI Input Clock Signal Configuration Register	0x0000_0000
USPI_LINECTL	USPIx_BA+0x2C	R/W	USCI Line Control Register	0x0000_0000
USPI_TXDAT	USPIx_BA+0x30	W	USCI Transmit Data Register	0x0000_0000
USPI_RXDAT	USPIx_BA+0x34	R	USCI Receive Data Register	0x0000_0000
USPI_BUFCTL	USPIx_BA+0x38	R/W	USCI Transmit/Receive Buffer Control Register	0x0000_0000
USPI_BUFSTS	USPIx_BA+0x3C	R	USCI Transmit/Receive Buffer Status Register	0x0000_0101
USPI_WKCTL	USPIx_BA+0x54	R/W	USCI Wake-up Control Register	0x0000_0000
USPI_WKSTS	USPIx_BA+0x58	R/W	USCI Wake-up Status Register	0x0000_0000
USPI_PROTCTL	USPIx_BA+0x5C	R/W	USCI Protocol Control Register	0x0000_0300
USPI_PROTIEN	USPIx_BA+0x60	R/W	USCI Protocol Interrupt Enable Register	0x0000_0000
USPI_PROTSTS	USPIx_BA+0x64	R/W	USCI Protocol Status Register	0x0000_0000

6.21.7 Register Description

USCI Control Register (USPI\_CTL)

Register	Offset	R/W	Description	Reset Value
USPI_CTL	USPIx_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]	<b>Reserved</b> Reserved.
[2:0]	<p><b>FUNMODE</b></p> <p><b>Function Mode</b> 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.</p> <p>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<sup>2</sup>C protocol is selected.</p> <p><b>Note:</b> Other bit combinations are reserved.</p>

**USCI Interrupt Enable Register (USPI\_INTEN)**

Register	Offset	R/W	Description	Reset Value
USPI_INTEN	USPIx_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	<b>Receive End Interrupt Enable Bit</b> 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	<b>Receive Start Interrupt Enable Bit</b> 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	<b>Transmit End Interrupt Enable Bit</b> 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	<b>Transmit Start Interrupt Enable Bit</b> 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 (USPI\_BRGEN)**

Register	Offset	R/W	Description	Reset Value
USPI_BRGEN	USPIx_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	<b>Clock Divider</b> This bit field defines the ratio between the protocol clock frequency $f_{PROT\_CLK}$ and the clock divider frequency $f_{DIV\_CLK}$ ( $f_{DIV\_CLK} = f_{PROT\_CLK} / (CLKDIV+1)$ ). <b>Note:</b> I <sup>2</sup> C function, the minimum value of CLKDIV is 8.
[15:6]	Reserved	Reserved.
[5]	TMCNTSRC	<b>Time Measurement Counter Clock Source Selection</b> 0 = Time measurement counter with $f_{PROT\_CLK}$ . 1 = Time measurement counter with $f_{DIV\_CLK}$ .
[4]	TMCNTEN	<b>Time Measurement Counter Enable Bit</b> This bit enables the 10-bit timing measurement counter. 0 = Time measurement counter Disabled. 1 = Time measurement counter Enabled.
[3:2]	SPCLKSEL	<b>Sample Clock Source Selection</b> This bit field used for the clock source selection of sample clock ( $f_{SAMP\_CLK}$ ) for the protocol processor. 00 = $f_{DIV\_CLK}$ . 01 = $f_{PROT\_CLK}$ . 10 = $f_{SCLK}$ . 11 = $f_{REF\_CLK}$ .
[1]	PTCLKSEL	<b>Protocol Clock Source Selection</b> This bit selects the source of protocol clock ( $f_{PROT\_CLK}$ ). 0 = Reference clock $f_{REF\_CLK}$ . 1 = $f_{REF\_CLK2}$ (its frequency is half of $f_{REF\_CLK}$ ).
[0]	RCLKSEL	<b>Reference Clock Source Selection</b> This bit selects the source of reference clock ( $f_{REF\_CLK}$ ).

		0 = Peripheral device clock $f_{PCLK}$ . 1 = Reserved.
--	--	---



**USCI Input Data Signal Configuration (USPI\_DATIN0)**

Register	Offset	R/W	Description	Reset Value
USPI_DATIN0	USPIx_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	<p><b>Input Signal Inverse Selection</b></p> <p>This bit defines the inverter enable of the input asynchronous signal.</p> <p>0 = The un-synchronized input signal will not be inverted.</p> <p>1 = The un-synchronized input signal will be inverted.</p> <p><b>Note:</b> In SPI protocol, it is suggested this bit should be set as 0.</p>
[1]	Reserved	Reserved.
[0]	SYNCSEL	<p><b>Input Signal Synchronization Selection</b></p> <p>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.</p> <p>0 = The un-synchronized signal can be taken as input for the data shift unit.</p> <p>1 = The synchronized signal can be taken as input for the data shift unit.</p> <p><b>Note:</b> In SPI protocol, it is suggested this bit should be set as 0.</p>

**USCI Input Control Signal Configuration (USPI\_CTLIN0)**

Register	Offset	R/W	Description	Reset Value
USPI_CTLIN0	USPIx_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	<p><b>Input Signal Inverse Selection</b></p> <p>This bit defines the inverter enable of the input asynchronous signal.</p> <p>0 = The un-synchronized input signal will not be inverted.</p> <p>1 = The un-synchronized input signal will be inverted.</p>
[1]	Reserved	Reserved.
[0]	SYNCSEL	<p><b>Input Synchronization Signal Selection</b></p> <p>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.</p> <p>0 = The un-synchronized signal can be taken as input for the data shift unit.</p> <p>1 = The synchronized signal can be taken as input for the data shift unit.</p> <p><b>Note:</b> In SPI protocol, it is suggested this bit should be set as 0.</p>

**USCI Input Clock Signal Configuration (USPI\_CLKIN)**

Register	Offset	R/W	Description	Reset Value
USPI_CLKIN	USPIx_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	<p><b>Input Synchronization Signal Selection</b></p> <p>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.</p> <p>0 = The un-synchronized signal can be taken as input for the data shift unit.</p> <p>1 = The synchronized signal can be taken as input for the data shift unit.</p> <p><b>Note:</b> In SPI protocol, it is suggested this bit should be set as 0.</p>

**USCI Line Control Register (USPI\_LINECTL)**

Register	Offset	R/W	Description	Reset Value
USPI_LINECTL	USPIx_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	<p><b>Word Length of Transmission</b>                      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.</p> <p>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].</p>
[7]	CTLOINV	<p><b>Control Signal Output Inverse Selection</b>                      This bit defines the relation between the internal control signal and the output control signal.</p> <p>0 = No effect.                      1 = The control signal will be inverted before its output.</p> <p><b>Note:</b> The control signal has different definitions in different protocol. In SPI protocol, the control signal means slave select signal.</p>
[6]	Reserved	Reserved.
[5]	DATOINV	<p><b>Data Output Inverse Selection</b>                      This bit defines the relation between the internal shift data value and the output data signal of USCIX_DAT0/1 pin.</p> <p>0 = Data output level is not inverted.                      1 = Data output level is inverted.</p>
[4:1]	Reserved	Reserved.
[0]	LSB	<b>LSB First Transmission Selection</b>

		<p>0 = The MSB, which bit of transmit/receive data buffer depends on the setting of DWIDTH, is transmitted/received first.</p> <p>1 = The LSB, the bit 0 of data buffer, will be transmitted/received first.</p>
--	--	--

**USCI Transmit Data Register (USPI\_TXDAT)**

Register	Offset	R/W	Description	Reset Value
USPI_TXDAT	USPIx_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	<p><b>Port Direction Control</b></p> <p>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.</p> <p>0 = The data pin is configured as output mode. 1 = The data pin is configured as input mode.</p>
[15:0]	TXDAT	<p><b>Transmit Data</b></p> <p>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.</p>

**USCI Receive Data Register (USPI\_RXDAT)**

Register	Offset	R/W	Description	Reset Value
USPI_RXDAT	USPIx_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	<b>Received Data</b> 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	USPIx_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	RXOVIEN	Reserved					
7	6	5	4	3	2	1	0
TXCLR	TXUDRIEN	Reserved					

Bits	Description	
[31:18]	Reserved	Reserved.
[17]	RXRST	<p><b>Receive Reset</b>                      0 = No effect.                      1 = Reset the receive-related counters, state machine, and the content of receive shift register and data buffer.  <b>Note:</b> It is cleared automatically after one PCLK cycle.</p>
[16]	TXRST	<p><b>Transmit Reset</b>                      0 = No effect.                      1 = Reset the transmit-related counters, state machine, and the content of transmit shift register and data buffer.  <b>Note:</b> It is cleared automatically after one PCLK cycle.</p>
[15]	RXCLR	<p><b>Clear Receive Buffer</b>                      0 = No effect.                      1 = The receive buffer is cleared. Should only be used while the buffer is not taking part in data traffic.  <b>Note:</b> It is cleared automatically after one PCLK cycle.</p>
[14]	RXOVIEN	<p><b>Receive Buffer Overrun Interrupt Enable Bit</b>                      0 = Receive overrun interrupt Disabled.                      1 = Receive overrun interrupt Enabled.</p>
[13:8]	Reserved	Reserved.
[7]	TXCLR	<p><b>Clear Transmit Buffer</b>                      0 = No effect.                      1 = The transmit buffer is cleared. Should only be used while the buffer is not taking part in data traffic.  <b>Note:</b> It is cleared automatically after one PCLK cycle.</p>
[6]	TXUDRIEN	<p><b>Slave Transmit Under-run Interrupt Enable Bit</b>                      0 = Transmit under-run interrupt Disabled.</p>



		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	USPIx_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	<p><b>Transmit Buffer Under-run Interrupt Status</b></p> <p>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</p> <p>0 = A transmit buffer under-run event has not been detected. 1 = A transmit buffer under-run event has been detected.</p>
[10]	Reserved	Reserved.
[9]	TXFULL	<p><b>Transmit Buffer Full Indicator</b></p> <p>0 = Transmit buffer is not full. 1 = Transmit buffer is full.</p>
[8]	TXEMPTY	<p><b>Transmit Buffer Empty Indicator</b></p> <p>0 = Transmit buffer is not empty. 1 = Transmit buffer is empty and available for the next transmission datum.</p>
[7:4]	Reserved	Reserved.
[3]	RXOVIF	<p><b>Receive Buffer Overrun Interrupt Status</b></p> <p>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.</p> <p>0 = A receive buffer overrun event has not been detected. 1 = A receive buffer overrun event has been detected.</p>
[2]	Reserved	Reserved.
[1]	RXFULL	<p><b>Receive Buffer Full Indicator</b></p> <p>0 = Receive buffer is not full. 1 = Receive buffer is full.</p>
[0]	RXEMPTY	<p><b>Receive Buffer Empty Indicator</b></p>

		0 = Receive buffer is not empty. 1 = Receive buffer is empty.
--	--	--

**USCI Wake-up Control Register (USPI\_WKCTL)**

Register	Offset	R/W	Description	Reset Value
USPI_WKCTL	USPIx_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	WKADDREN	WKEN

Bits	Description	
[31:3]	Reserved	Reserved.
[2]	PDBOPT	<b>Power Down Blocking Option</b> 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]	WKADDREN	<b>Wake-up Address Match Enable Bit</b> 0 = The chip is woken up according data toggle. 1 = The chip is woken up according address match.
[0]	WKEN	<b>Wake-up Enable Bit</b> 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	USPIx_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	<b>Wake-up Flag</b> 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 – SPI (USPI\_PROTCTL)**

Register	Offset	R/W	Description	Reset Value
USPI_PROTCTL	USPIx_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]	<p><b>PROTEN</b></p> <p><b>SPI Protocol Enable Bit</b> 0 = SPI Protocol Disabled. 1 = SPI Protocol Enabled.</p>
[30:29]	Reserved.
[28]	<p><b>TXUDRPOL</b></p> <p><b>Transmit Under-run Data Polarity (for Slave)</b> This bit defines the transmitting data level 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.</p>
[27:26]	Reserved.
[25:16]	<p><b>SLVTOCNT</b></p> <p><b>Slave Mode Time-out Period (Slave Only)</b> 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<sub>DIV_CLK</sub>.</p>
[15]	Reserved.
[14:12]	<p><b>TSMSEL</b></p> <p><b>Transmit Data Mode Selection</b> This bit field describes how receive and transmit data is shifted in and out. TSMSEL = 000b: Full-duplex SPI. TSMSEL = 100b: Half-duplex SPI. Other values are reserved. <b>Note:</b> Changing the value of this bit field will produce the TXRST and RXRST to clear the TX/RX data buffer automatically.</p>
[11:8]	<p><b>SUSPITV</b></p> <p><b>Suspend Interval (Master Only)</b> 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</p>

		<p>edge of the following transaction word. The default value is 0x3. The period of the suspend interval is obtained according to the following equation.</p> $(SUSPITV[3:0] + 0.5) * \text{period of SPI\_CLK clock cycle}$ <p>Example:</p> <p>SUSPITV = 0x0 .... 0.5 SPI_CLK clock cycle.</p> <p>SUSPITV = 0x1 .... 1.5 SPI_CLK clock cycle.</p> <p>.....</p> <p>SUSPITV = 0xE .... 14.5 SPI_CLK clock cycle.</p> <p>SUSPITV = 0xF .... 15.5 SPI_CLK clock cycle.</p>
[7:6]	<b>SCLKMODE</b>	<p><b>Serial Bus Clock Mode</b></p> <p>This bit field defines the SCLK idle status, data transmit, and data receive edge.</p> <p>MODE0 = The idle state of SPI clock is low level. Data is transmitted with falling edge and received with rising edge.</p> <p>MODE1 = The idle state of SPI clock is low level. Data is transmitted with rising edge and received with falling edge.</p> <p>MODE2 = The idle state of SPI clock is high level. Data is transmitted with rising edge and received with falling edge.</p> <p>MODE3 = The idle state of SPI clock is high level. Data is transmitted with falling edge and received with rising edge.</p>
[5:4]	<b>Reserved</b>	Reserved.
[3]	<b>AUTOSS</b>	<p><b>Automatic Slave Select Function Enable (Master Only)</b></p> <p>0 = Slave select signal will be controlled by the setting value of SS (USPI_PROTCTL[2]) bit.</p> <p>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.</p>
[2]	<b>SS</b>	<p><b>Slave Select Control (Master Only)</b></p> <p>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.</p> <p>If the AUTOSS function is enabled (AUTOSS = 1), the setting value of this bit will not affect the current state of slave select signal.</p> <p><b>Note:</b> In SPI protocol, the internal slave select signal is active high.</p>
[1]	<b>SLV3WIRE</b>	<p><b>Slave 3-wire Mode Selection (Slave Only)</b></p> <p>The SPI protocol can work with 3-wire interface (without slave select signal) in Slave mode.</p> <p>0 = 4-wire bi-direction interface.</p> <p>1 = 3-wire bi-direction interface.</p>
[0]	<b>SLAVE</b>	<p><b>Slave Mode Selection</b></p> <p>0 = Master mode.</p> <p>1 = Slave mode.</p>

**USCI Protocol Interrupt Enable Register – SPI (USPI\_PROTIEN)**

Register	Offset	R/W	Description	Reset Value
USPI_PROTIEN	USPIx_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	<p><b>Slave Mode Bit Count Error Interrupt Enable Bit</b></p> <p>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.</p> <p>0 = The Slave mode bit count error interrupt Disabled.</p> <p>1 = The Slave mode bit count error interrupt Enabled.</p>
[2]	SLVTOIEN	<p><b>Slave Time-out Interrupt Enable Bit</b></p> <p>In SPI protocol, this bit enables the interrupt generation in case of a Slave time-out event.</p> <p>0 = The Slave time-out interrupt Disabled.</p> <p>1 = The Slave time-out interrupt Enabled.</p>
[1]	SSACTIEN	<p><b>Slave Select Active Interrupt Enable Bit</b></p> <p>This bit enables/disables the generation of a slave select interrupt if the slave select changes to active.</p> <p>0 = Slave select active interrupt generation Disabled.</p> <p>1 = Slave select active interrupt generation Enabled.</p>
[0]	SSINAIEN	<p><b>Slave Select Inactive Interrupt Enable Bit</b></p> <p>This bit enables/disables the generation of a slave select interrupt if the slave select changes to inactive.</p> <p>0 = Slave select inactive interrupt generation Disabled.</p> <p>1 = Slave select inactive interrupt generation Enabled.</p>



**USCI Protocol Status Register – SPI (USPI\_PROTSTS)**

Register	Offset	R/W	Description	Reset Value
USPI_PROTSTS	USPIx_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]	<b>Reserved</b> Reserved.
[18]	<p><b>SLVUDR</b></p> <p><b>Slave Mode Transmit Under-run Status (Read Only)</b> 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.</p>
[17]	<p><b>BUSY</b></p> <p><b>Busy Status (Read Only)</b> 0 = SPI is in idle state. 1 = SPI is in busy state. The following lists the bus busy conditions: a. USPI_PROTCTL[31] = 1 and the TXEMPTY = 0. b. For SPI Master mode, the TXEMPTY = 1 but the current transaction is not finished yet. c. 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. d. 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.</p>
[16]	<p><b>SSLINE</b></p> <p><b>Slave Select Line Bus Status (Read Only)</b> 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.</p>
[15:10]	<b>Reserved</b> Reserved.
[9]	<p><b>SSACTIF</b></p> <p><b>Slave Select Active Interrupt Flag (for Slave Only)</b> 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.</p>

		1 = The slave select signal has changed to active. <b>Note:</b> The internal slave select signal is active high.
[8]	<b>SSINAIF</b>	<b>Slave Select Inactive Interrupt Flag (for Slave Only)</b> 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. <b>Note:</b> The internal slave select signal is active high.
[7]	<b>Reserved</b>	Reserved.
[6]	<b>SLVBEIF</b>	<b>Slave Bit Count Error Interrupt Flag (for Slave Only)</b> 0 = Slave bit count error event does not occur. 1 = Slave bit count error event occurs. <b>Note:</b> It is cleared by software write 1 to this bit.
[5]	<b>SLVTOIF</b>	<b>Slave Time-out Interrupt Flag (for Slave Only)</b> 0 = Slave time-out event did not occur. 1 = Slave time-out event occurred. <b>Note:</b> This bit is cleared by software writing 1 to it.
[4]	<b>RXENDIF</b>	<b>Receive End Interrupt Flag</b> 0 = Receive end event does not occur. 1 = Receive end event occurred. <b>Note:</b> This bit is cleared by software writing 1 to it.
[3]	<b>RXSTIF</b>	<b>Receive Start Interrupt Flag</b> 0 = Receive start event did not occur. 1 = Receive start event occurred. <b>Note:</b> This bit is cleared by software writing 1 to it.
[2]	<b>TXENDIF</b>	<b>Transmit End Interrupt Flag</b> 0 = Transmit end event did not occur. 1 = Transmit end event occurred. <b>Note:</b> This bit is cleared by software writing 1 to it.
[1]	<b>TXSTIF</b>	<b>Transmit Start Interrupt Flag</b> 0 = Transmit start event did not occur. 1 = Transmit start event occurred. <b>Note:</b> This bit is cleared by software writing 1 to it.
[0]	<b>Reserved</b>	Reserved.

## 6.22 USCI - I<sup>2</sup>C Mode

### 6.22.1 Overview

On I<sup>2</sup>C 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 6.22-1 for more detailed I<sup>2</sup>C BUS Timing.

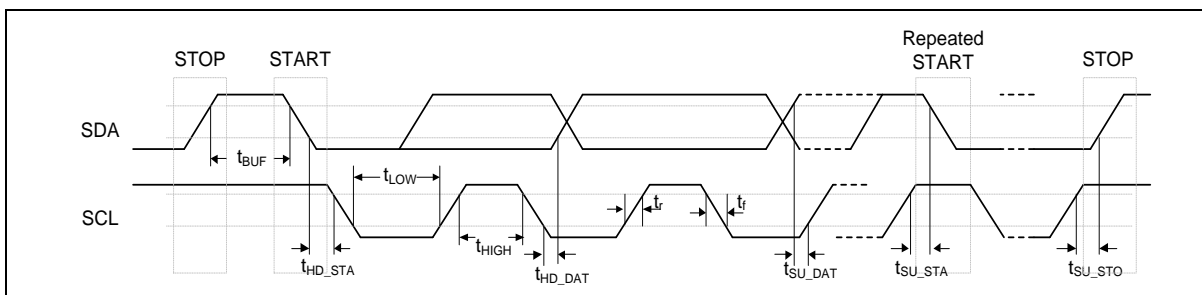


Figure 6.22-1 I<sup>2</sup>C Bus Timing

The device's on-chip I<sup>2</sup>C provides the serial interface that meets the I<sup>2</sup>C bus standard mode specification. The I<sup>2</sup>C port handles byte transfers autonomously. The I<sup>2</sup>C mode is selected by FUNMODE (UI2C\_CTL [2:0]) = 100b. When enable this port, the USCI interfaces to the I<sup>2</sup>C bus via two pins: SDA and SCL. When I/O pins are used as I<sup>2</sup>C ports, user must set the pins function to I<sup>2</sup>C in advance.

**Note:** Pull-up resistor is needed for I<sup>2</sup>C operation because the SDA and SCL are set to open-drain pins when USCI is selected to I<sup>2</sup>C operation mode .

### 6.22.2 Features

- Full master and slave device capability
- Supports of 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 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)

6.22.3 Block Diagram

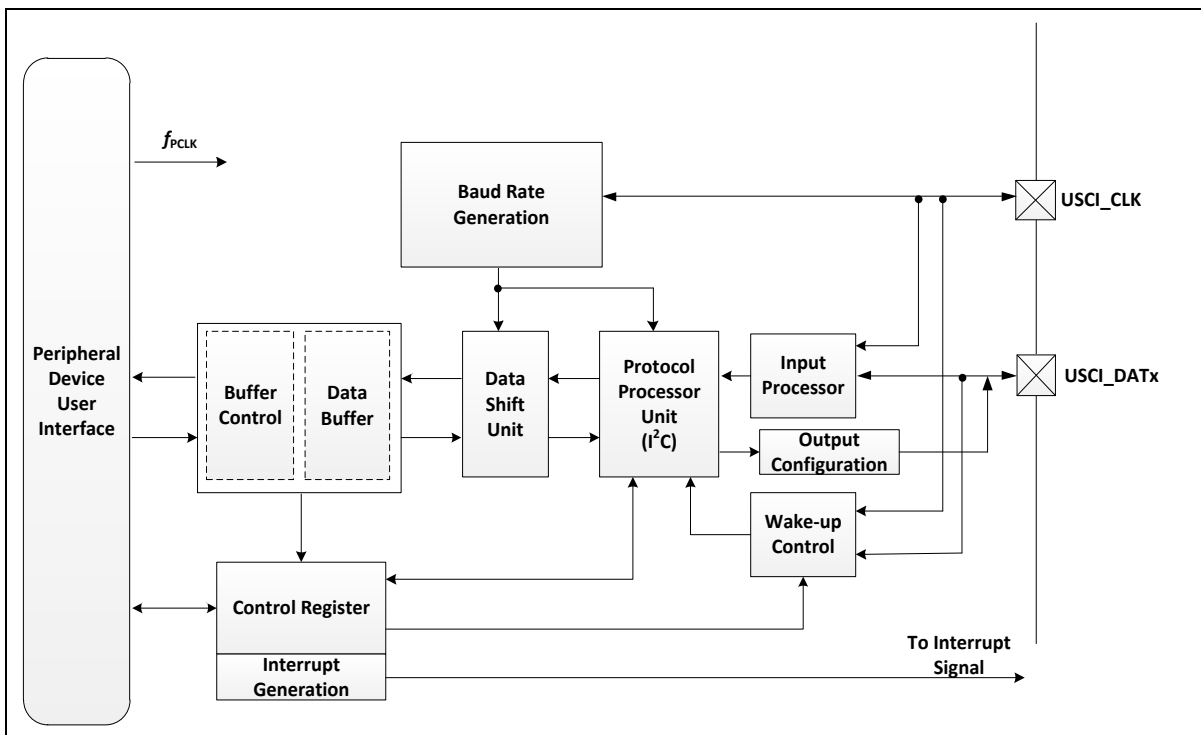


Figure 6.22-2 USCI I2C Mode Block Diagram

6.22.4 Basic Configuration

6.22.4.1 Basic Configuration of USCI0-I2C

- Clock Source Configuration
  - Enable USCI0-I2C peripheral clock in USCI0CKEN (CLK\_APBCLK1[8]).
- Reset Configuration
  - Reset USCI0-I2C controller in USCI0RST (SYS\_IPRST2[8]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
USCI0	USCI0_CLK	PE.5	MFP4
		PC.4	MFP5
		PB.9	MFP8
	USCI0_CTL0	PE.4	MFP4
		PC.3	MFP5
		PB.8, PE.2	MFP8
	USCI0_CTL1	PC.2, PC.7	MFP4
		PB.4	MFP8
	USCI0_DAT0	PC.0, PC.5	MFP4

		PB.2	MFP8
	USCI0_DAT1	PC.1, PC.6	MFP4
		PB.3	MFP8

6.22.4.2 Basic Configuration of USCI1-I2C

- Clock Source Configuration
  - Enable USCI1-I2C peripheral clock in USCI1CKEN (CLK\_APBCLK1[9]).
- Reset Configuration
  - Reset USCI1-I2C controller in USCI1RST (SYS\_IPRST2[9]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
USCI1	USCI1_CLK	PD.15	MFP1
		PA.15	MFP4
		PA.3	MFP8
	USCI1_CTL0	PD.12	MFP1
		PA.0	MFP4
		PA.2	MFP8
	USCI1_CTL1	PD.7	MFP1
		PA.1, PA.14	MFP4
	USCI1_DAT0	PD.14	MFP1
		PB.0	MFP6
USCI1_DAT1	PD.13	MFP1	
	PB.1	MFP8	

6.22.4.3 Basic Configuration of USCI2-I2C

- Clock Source Configuration
  - Enable USCI2-I2C peripheral clock in USCI2CKEN (CLK\_APBCLK1[10]).
- Reset Configuration
  - Reset USCI2-I2C controller in USCI2RST (SYS\_IPRST2[10]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
USCI2	USCI2_CLK	PC.11, PD.1	MFP4
		PF.2	MFP5
	USCI2_CTL0	PC.12, PD.0, PD.9	MFP4
		PF.1	MFP5

	USCI2_CTL1	PB.7, PC.9, PD.8	MFP4
		PF.0	MFP5
	USCI2_DAT0	PC.13, PD.2	MFP4
		PD.10	MFP5
	USCI2_DAT1	PC.10, PD.3	MFP4
		PD.11	MFP5

### 6.22.5 Functional Description

#### 6.22.5.1 START or Repeated START Signal

Figure 6.22-2 shows the typical I<sup>2</sup>C 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

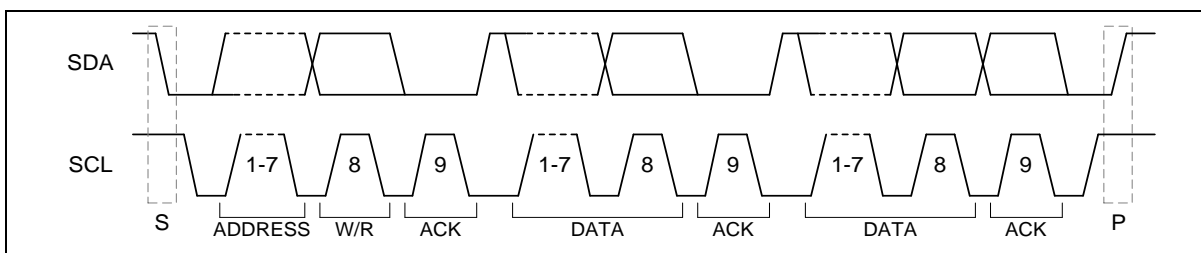


Figure 6.22-3 I<sup>2</sup>C Protocol

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 a START signal between START signal and STOP signal 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.

#### 6.22.5.2 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 that we called bus free. Figure 6.22-4 shows the waveform of START, Repeat START and STOP.

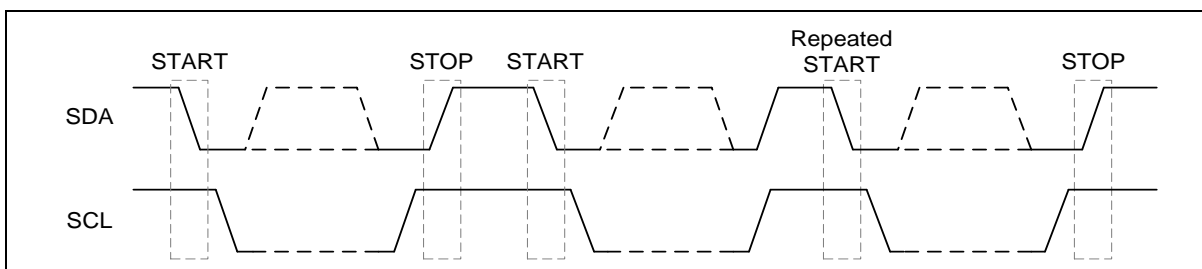


Figure 6.22-4 START and STOP Conditions

### 6.22.5.3 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.

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 I<sup>2</sup>C port hardware will respond to General Call address (00H). Clear GC bit to disable general call function.
- The I<sup>2</sup>C 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.
- In addition to this, 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 I<sup>2</sup>C 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 I<sup>2</sup>C port is equipped 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.

### 6.22.5.4 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.

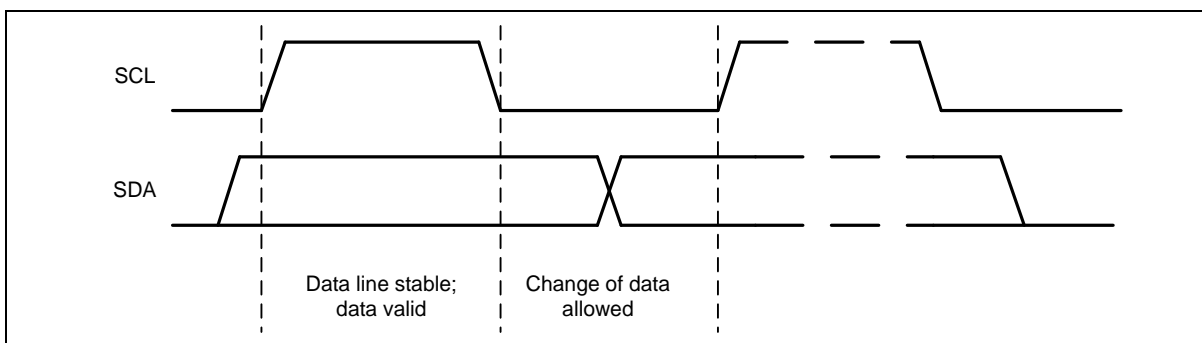


Figure 6.22-5 Bit Transfer on the I<sup>2</sup>C Bus

If the master received data, does Not Acknowledge (NACK) the slave, the slave releases the SDA line for the master to generate a STOP or Repeated START signal.

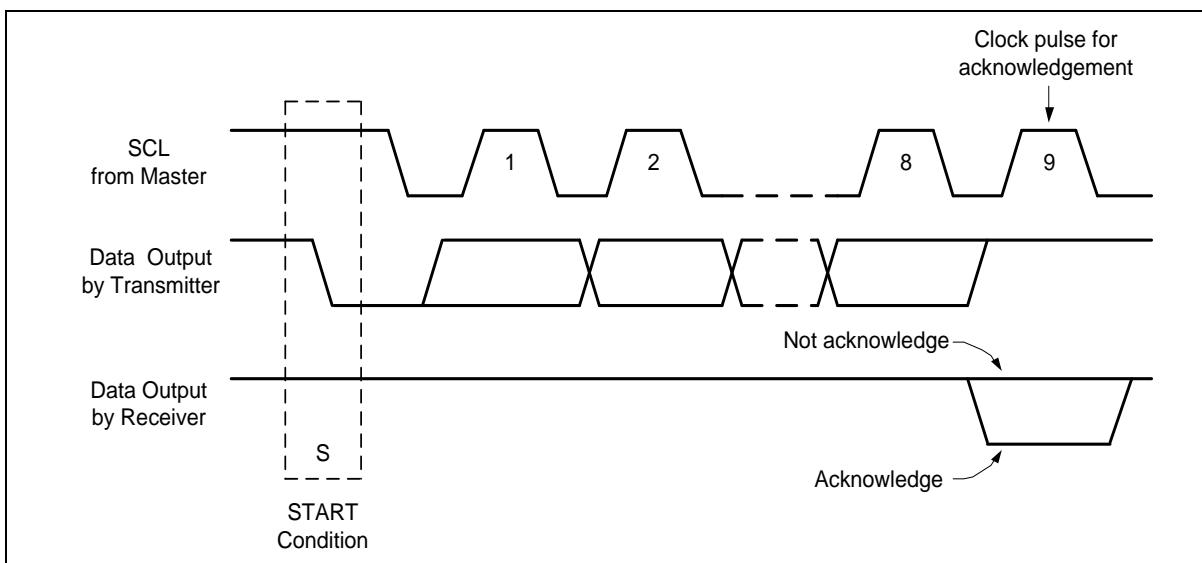


Figure 6.22-6 Acknowledge on the I<sup>2</sup>C Bus

#### 6.22.5.5 Clock Baud Rate Bits

For this section, please refer to Figure 6.19-9 basic clock divider counter. The data baud rate of I<sup>2</sup>C is determined by UI2C\_BRGEN register when I<sup>2</sup>C is in Master Mode, and it is not necessary in a Slave mode. In the Slave mode, I<sup>2</sup>C will automatically synchronize it with any clock frequency from master I<sup>2</sup>C 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:

$$f_{I2C} = \frac{f_{REF\_CLK}}{2} \times \frac{1}{CLKDIV + 1} \times \frac{1}{PDSCNT + 1} \times \frac{1}{DSCNT + 1}$$

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:

$$f_{I2C} = \frac{f_{REF\_CLK}}{4} \times \frac{1}{CLKDIV + 1} \times \frac{1}{PDSCNT + 1} \times \frac{1}{DSCNT + 1}$$

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:

$$f_{I2C} = \frac{f_{REF\_CLK}}{2} \times \frac{1}{CLKDIV + 1} \times \frac{1}{2} \times \frac{1}{PDSCNT + 1} \times \frac{1}{DSCNT + 1}$$

### 6.22.5.6 Byte Stretching

If a device is selected as transceiver 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]).

### 6.22.5.7 Master Arbitration

In some applications, there are two or more masters on the same I<sup>2</sup>C bus to access slaves, and the masters may transmit data simultaneously. The I<sup>2</sup>C supports multi-master by including collision detection and arbitration to prevent data corruption.

If two masters sometimes initiate I<sup>2</sup>C 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 I<sup>2</sup>C 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 I<sup>2</sup>C master must monitor the I<sup>2</sup>C bus for collisions and act accordingly. Figure 6.22-7 describe DATA1 and DATA2 are compete arbitration.

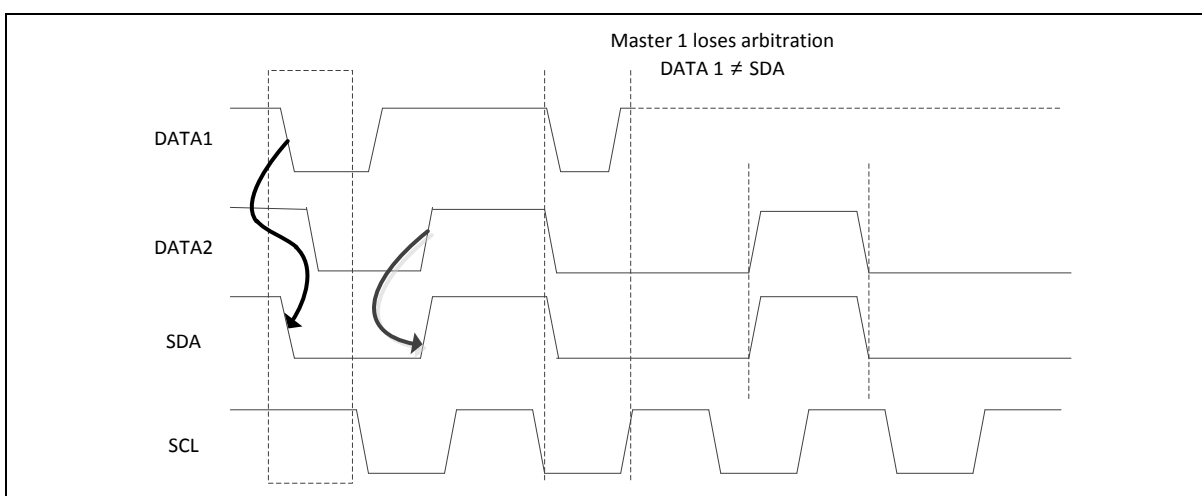


Figure 6.22-7 Arbitration Lost

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, the next data bit values can be 0. 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.

#### 6.22.5.8 Transmission Chain

The I<sup>2</sup>C 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  $f_{PROT\_CLK}$ . We suggest user adopt  $f_{PCLK}$ .

#### 6.22.5.9 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.

#### 6.22.5.10 Interrupt

### I<sup>2</sup>C Protocol Interrupt Events

The following protocol-related events are generated in I<sup>2</sup>C 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]))

#### 6.22.5.11 Operating the I<sup>2</sup>C

In order to operate the I<sup>2</sup>C protocol, the following issues have to be considered:

##### Select I<sup>2</sup>C mode:

It is recommended to configure all parameters of the I<sup>2</sup>C that do not change during run time while FUNMODE (UI2C\_CTL [2:0]) = 000b. The I<sup>2</sup>C control flow has to be done while FUNMODE (UI2C\_CTL [2:0]) = 000b to avoid unintended edges of the input signals and the I<sup>2</sup>C 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 I<sup>2</sup>C bus lines.

**Note:** *The step to enable the alternate output port functions should only be done after the I<sup>2</sup>C 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  $1/f_{PROT\_CLK}$  if another I<sup>2</sup>C participant lengthens the SCL low phase on the bus. Note that the SCL maximum frequency is  $SAMP\_CLK/2$  and the SPCLKSEL (UI2C\_BRGEN [3:2]) must be set 0 for selecting  $f_{SAMP\_CLK} = f_{DIV\_CLK}$ .

**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 I<sup>2</sup>C ports support three operation modes, Master, Slave, and General Call Mode.

In a given application, I<sup>2</sup>C port may operate as a master or as a slave. In Slave mode, the I<sup>2</sup>C 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 interrupted. If address arbitration is lost in Master mode, I<sup>2</sup>C port switches to Slave mode immediately and can detect its own slave address in the same serial transfer.

To control the I<sup>2</sup>C 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 I<sup>2</sup>C 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 I<sup>2</sup>C 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 I<sup>2</sup>C 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 6.22-8 shows the current I<sup>2</sup>C 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 I<sup>2</sup>C 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.

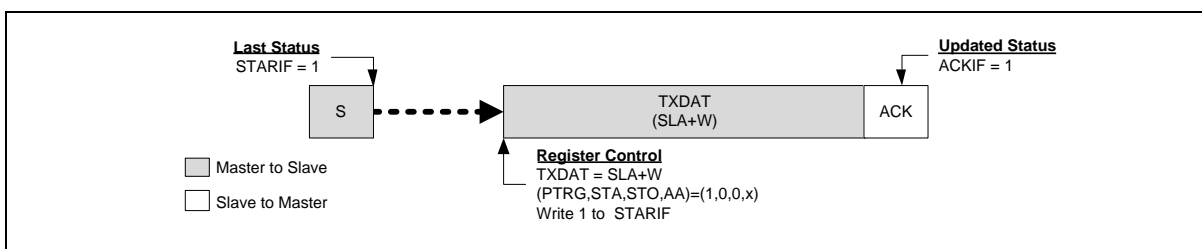


Figure 6.22-8 Control I<sup>2</sup>C Bus according to Current I<sup>2</sup>C Status

### Data Transfer on the I<sup>2</sup>C Bus

Figure 6.22-9 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.

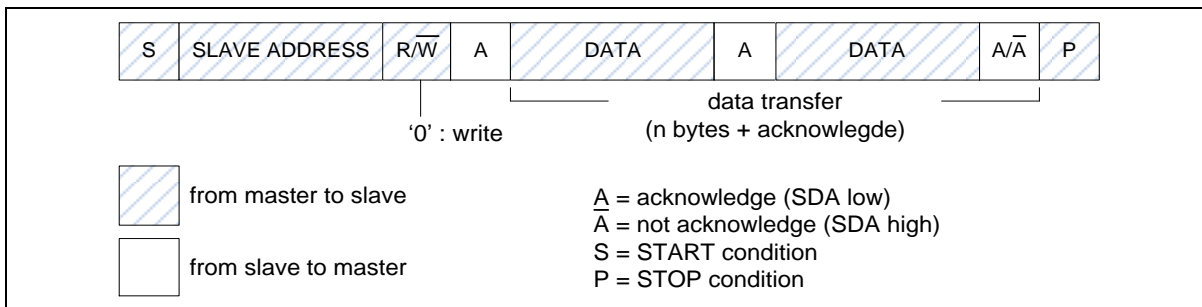


Figure 6.22-9 Master Transmits Data to Slave with a 7-bit address

Figure 6.22-10 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.

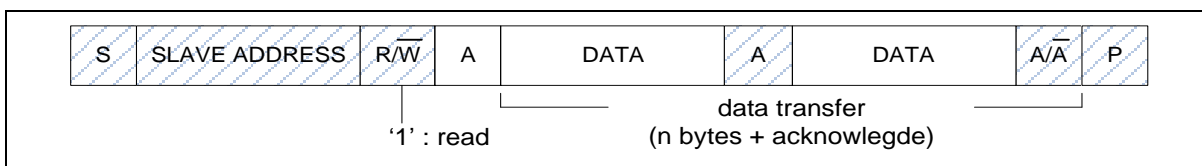


Figure 6.22-10 Master Reads Data from Slave with a 7-bit address

Figure 6.22-11 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.

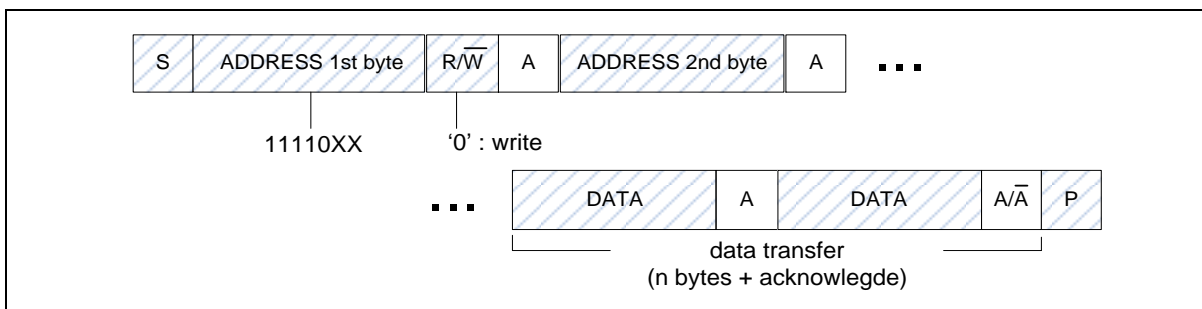


Figure 6.22-11 Master Transmits Data to Slave by 10-bit address

Figure 6.22-12 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.

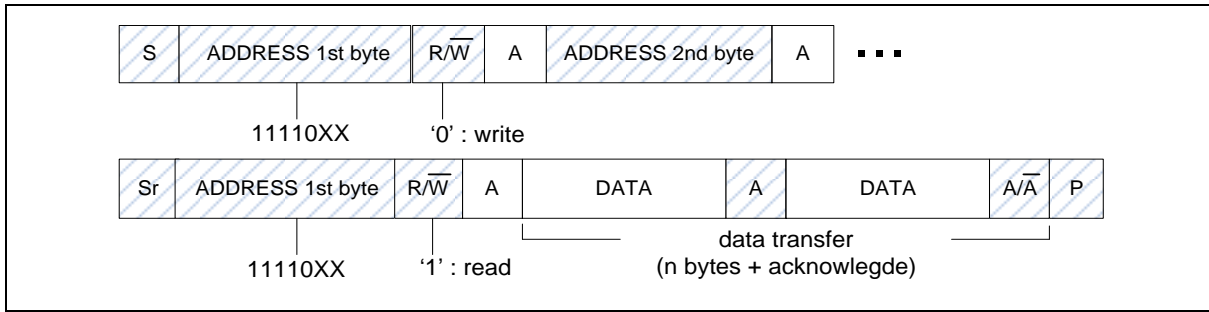


Figure 6.22-12 Master Reads Data from Slave by 10-bit address

**Master Mode**

In Figure 6.22-13, all possible protocols for I<sup>2</sup>C master are shown. User needs to follow proper path of the flow to implement required I<sup>2</sup>C protocol.

In other words, user can send a START signal to bus and I<sup>2</sup>C will be in Master Transmitter mode (Figure 6.22-13) or Master receiver mode (Figure 6.22-14) 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 I<sup>2</sup>C protocol.

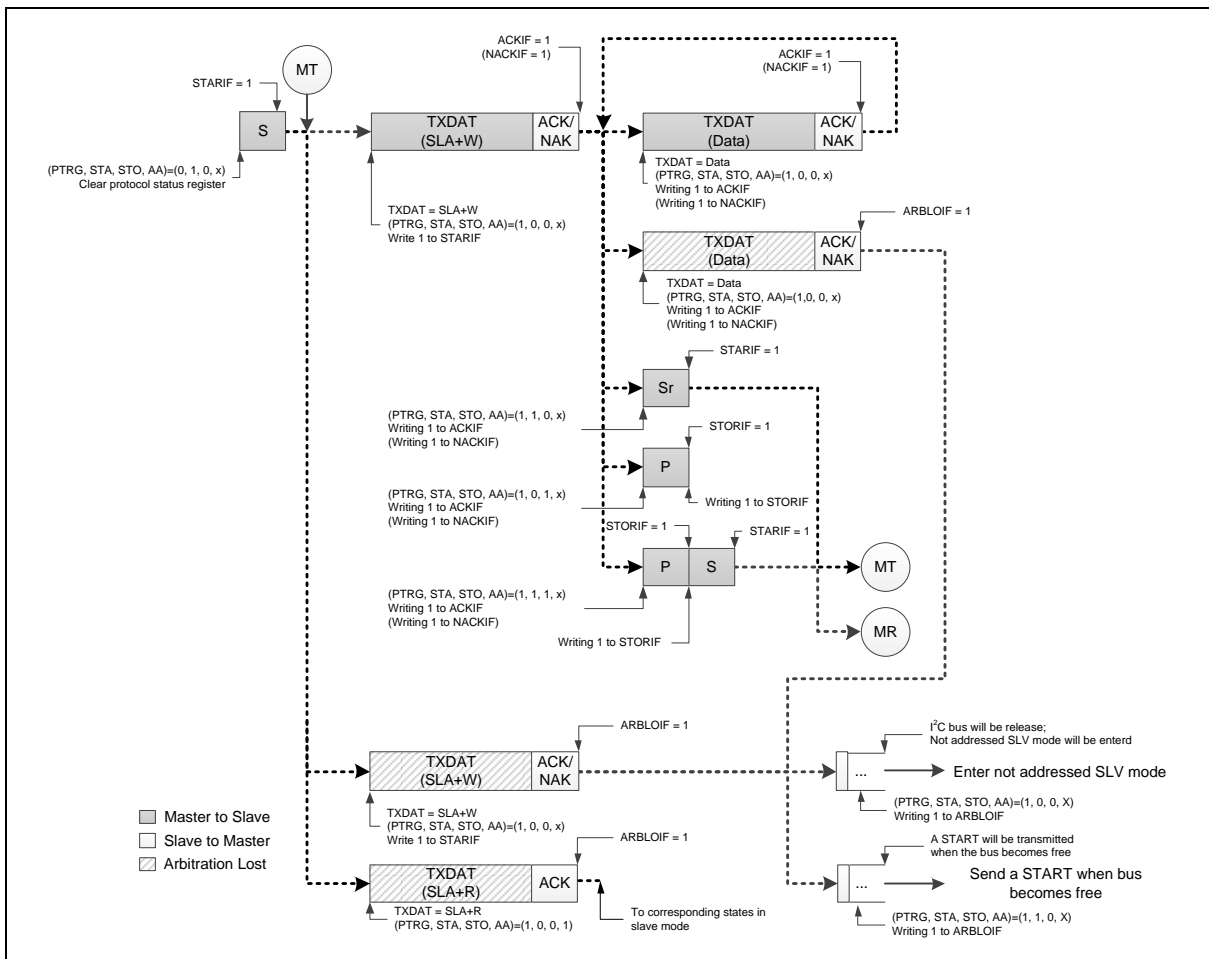


Figure 6.22-13 Master Transmitter Mode Control Flow with 7-bit Address

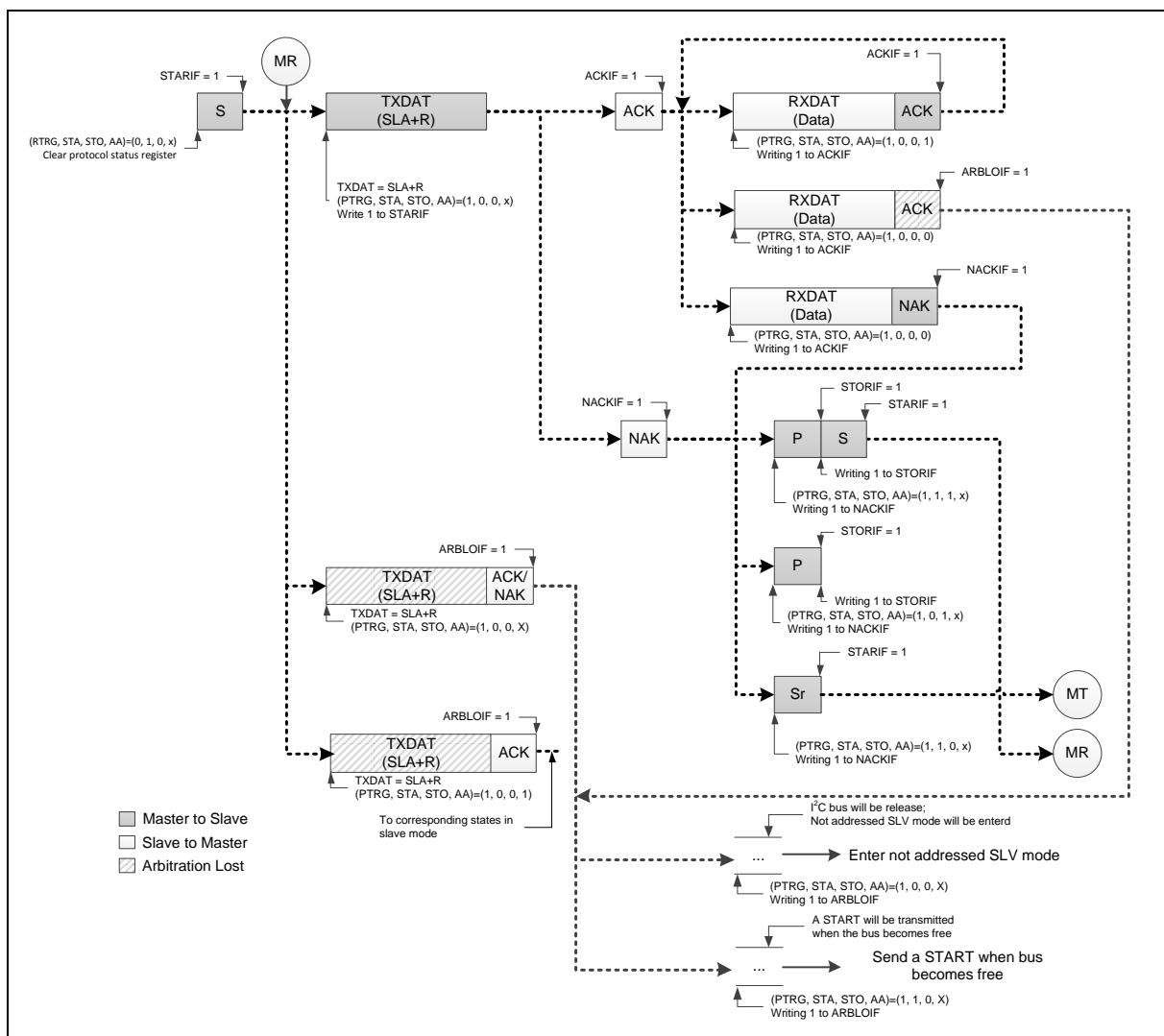


Figure 6.22-14 Master Receiver Mode Control Flow with 7-bit Address

If the I<sup>2</sup>C 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 become free. Otherwise, user may writing 1 to ARBLOIF (UI2C\_PROTSTS [11]) and set (PTRG, STA, STO, AA) = (1, 0, 0, X) to release I<sup>2</sup>C bus and enter not addressed Slave mode.

### Slave Mode

When reset, I<sup>2</sup>C is not addressed and will not recognize the address on I<sup>2</sup>C bus. User can set device address by UI2C\_DEVADDRn and set (PTRG, STA, STO, AA) = (1, 0, 0, 1) to let I<sup>2</sup>C recognize the address sent by master. Figure 6.22-15 shows all the possible flow for I<sup>2</sup>C in Slave mode. Users need to follow a proper flow (as shown in Figure 6.22-15 to implement their own I<sup>2</sup>C protocol.

If bus arbitration is lost in Master mode, I<sup>2</sup>C 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) or SLA+R (Master want to read data from Slave) after arbitration lost, the ARBLOIF will be set to 1.

The I<sup>2</sup>C controller supports two slave address match flags, are ADMAT0 and ADMAT1 on UI2C\_ADMAT[1:0] register. Every control register represent which address is used and set 1 to inform software.

**Note:** During I<sup>2</sup>C communication, the SCL clock will be released when writing '1' to PTRG (UI2C\_PROTCTL [5]) in Slave mode.

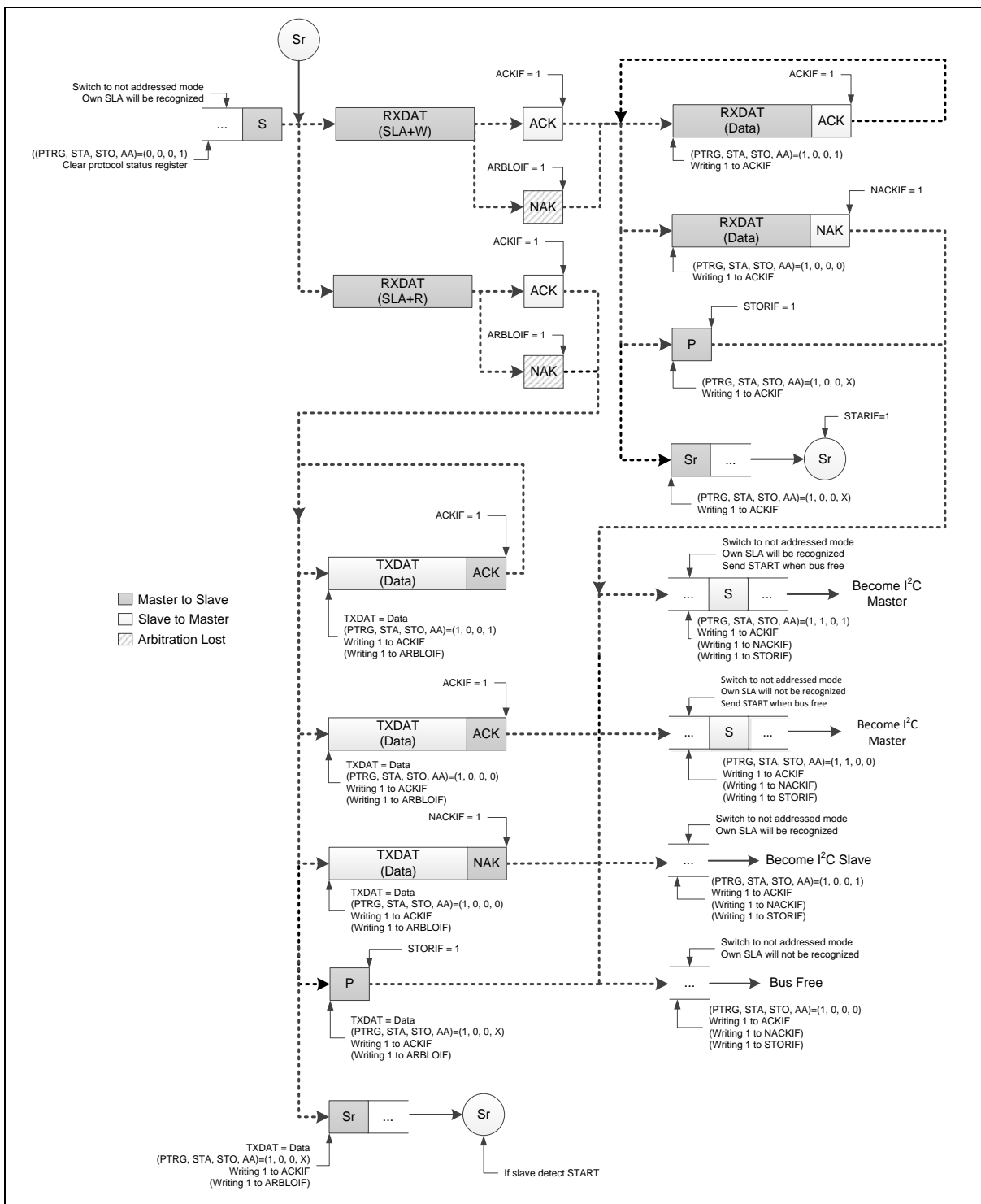


Figure 6.22-15 Save Mode Control Flow with 7-bit address

If I<sup>2</sup>C 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 6.22-15 when got



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 I<sup>2</sup>C signal or address from master. At this status, I<sup>2</sup>C 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 I<sup>2</sup>C 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 I<sup>2</sup>C in slave mode, it can receive the general call address by 0x00 after master send general call address to I<sup>2</sup>C bus, and then it also will follow protocol status register.

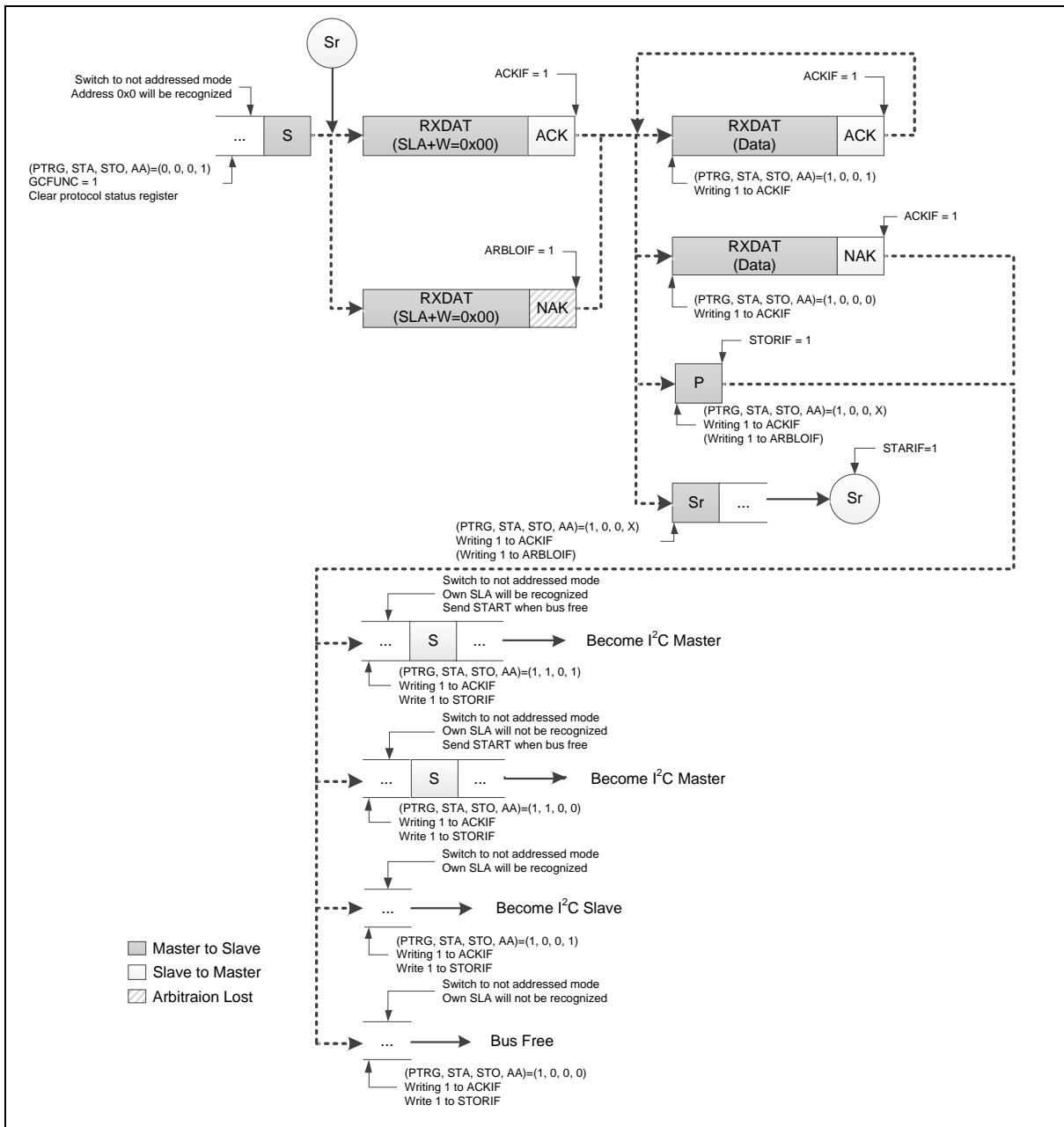


Figure 6.22-16 GC Mode with 7-bit address

If I<sup>2</sup>C 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 6.22-16 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 I<sup>2</sup>C signal or address from master. At this time, I<sup>2</sup>C controller should be reset by setting FUNMODE (UI2C\_CTL [2:0]) = 000b to leave this status.

### Protocol Functional Description

#### Monitor Mode

When I<sup>2</sup>C enter 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. Note that monitor mode not support 10-bit mode.

**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 I<sup>2</sup>C 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 I<sup>2</sup>C 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 [5:0] (UI2C\_TMCTL[11:6]) to configure hold time and STCTL [5:0] (UI2C\_TMCTL[5:0]) 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 on the limitation of setup and hold time configuration, the timing setting must follow I<sup>2</sup>C protocol. Once setup time configuration greater than design limitation, that means if setup time setting make SCL output less than three PCLKs, I<sup>2</sup>C controller can't work normally due to SCL must sample three times. And once hold time configuration greater than I<sup>2</sup>C clock limitation, I<sup>2</sup>C will occur bus error. Suggest that user calculate suitable timing with baud rate and protocol before setting timing. Table 6.22-1 shows the relationship between I<sup>2</sup>C 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 our design, but user should follow I<sup>2</sup>C protocol standard.

PCLK	I <sup>2</sup> C Baud Rate	100k	200k	400k		
	12MHz		120	60	30	
24MHz		240	120	60		
48MHz		480	240	120		

Table 6.22-1 Relationship between I<sup>2</sup>C Baud Rate and PCLK

For setup time wrong adjustment example, we assume one SCL cycle contains ten PCLKs and set STCTL [5:0] (UI2C\_TMCTL[5:0]) to 3 that stretch three PCLKs for setup time setting. The setup time setting limitation:  $ST_{limit} = (UI2C\_BRGEN[25:16]+1) - 6$ . For example, if user decide PCLK = 12MHz and baud rate =100k, the UI2C\_BRGEN[25:16] must set 59, and the STCTL [5:0] maximum value is 54.

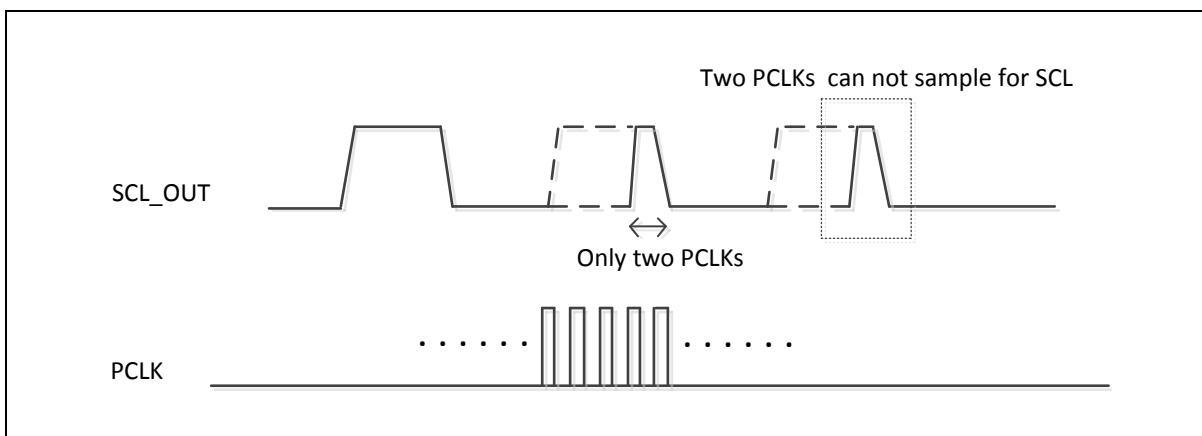


Figure 6.22-17 Setup Time Wrong Adjustment

For hold time wrong adjustment example, we use I<sup>2</sup>C Baud Rate = 400k and PCLK = 48MHz, the SCL high/low duty = 60 PCLK. When we set HTCTL [5:0] (UI2C\_TMCTL[11:6]) to 63 and STCTL [5:0] (UI2C\_TMCTL[5:0]) to 0, then SDA output delay will over SCL high duty and cause bus error. The hold time setting limitation:  $HT_{limit} = (UI2C\_BRGEN[25:16]+1) - 9$ . For example, if user decide PCLK = 12MHz and baud rate =100k, the UI2C\_BRGEN[25:16] must set 59, and the HTCTL [5:0] maximum value is 51.

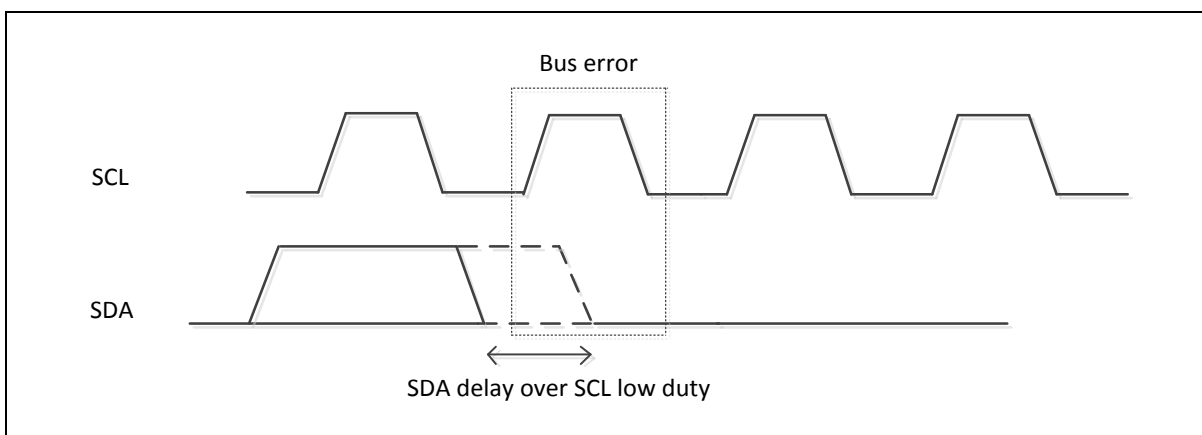


Figure 6.22-18 Hold Time Wrong Adjustment

### I<sup>2</sup>C Time-out Function

There is a time-out counter TOCNT (UI2C\_PROTCTL [25:16]) which can be used to deal with the I<sup>2</sup>C 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 I<sup>2</sup>C interrupt to CPU or stops counting by clearing TOIEN (UI2C\_PROTIEN [0]) to 0 or setting all I<sup>2</sup>C 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 6.22-19 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 (USCI\_BRGEN [5]) must be set zero.

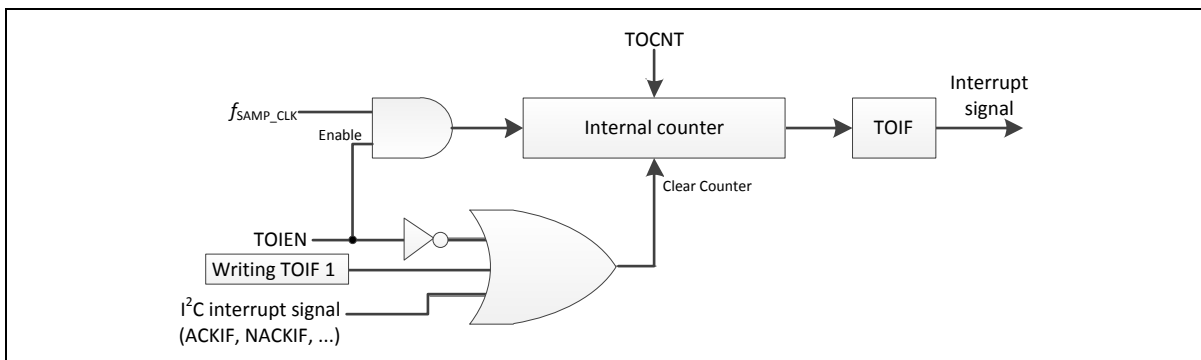


Figure 6.22-19 I<sup>2</sup>C Time-out Count Block Diagram

### Wake-up Function

When chip enters Power-down mode and set WKEN (WKCTL[0]) to 1, other I<sup>2</sup>C master can wake-up our chip by addressing our I<sup>2</sup>C 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 wakeup 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 the address wakeup frame has 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 wakeup procedure. Note that user must clear WKIF 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 wake-up by the address match frame.

When system is woken up by other I<sup>2</sup>C master device, WKIF is set to indicate this event. User needs write "1" to clear this bit.

I<sup>2</sup>C also support data toggle mode. When system is in power-down and the WKEN (UI2C\_WKCTL [0]) set to 1 and WKADDREN (UI2C\_WKCTL[1]) set to 0, the toggle of incoming data pin can wake-up the system.

### Example for Random Read on EEPROM

The following steps are used to configure the USCI0\_I<sup>2</sup>C related registers when using I<sup>2</sup>C protocol to read data from EEPROM.

1. Set USCI0\_I<sup>2</sup>C the multi-function pin in the SYS\_GPB\_MFPL or SYS\_GPBMPFH or SYS\_GPC\_MFPL or SYS\_GPE\_MFPL registers as SCL and SDA pins.
2. Enable USCI0 APB clock, USCI0CKEN =1 in the "CLK\_APBCLK[8]" register.
3. Set USCI0RST=1 to reset USCI controller then set USCI0RST=0 let USCI controller to normal operation, in the "SYS\_IPRST2[8]" register.
4. Set FUNMODE =100b to enable USCI0\_I<sup>2</sup>C controller in the "UI2C\_CTL" register.
5. Give USCI0\_I<sup>2</sup>C clock a divided register value for USCI0\_I<sup>2</sup>C clock rate in the "UI2C\_BRGEN".
6. Set SETENA =0x00400000 in the "NVIC\_ISR" register to set USCI\_IRQ.

7. Set ACKIEN, ERRIEN, ARBLOIEN, NACKIEN, STORIEN, STARIEN, and TOIEN to enable I<sup>2</sup>C Interrupt in the "UI2C\_PROTIEN" register.
8. Set USCI address registers "USCI\_ADDR0 ~ USCI\_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 6.22-21 shows the EEPROM random read operation.

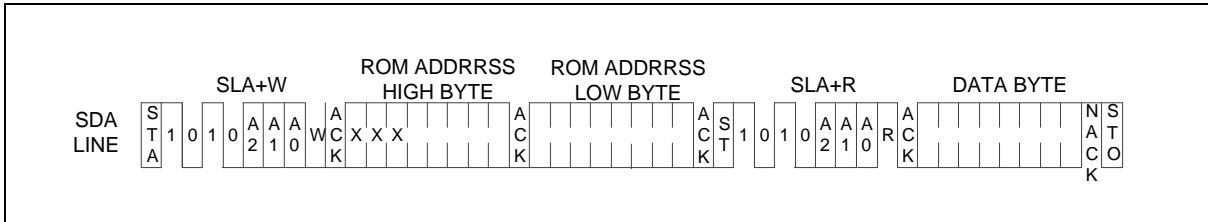


Figure 6.22-20 EEPROM Random Read

Figure 6.22-21 shows how to use I<sup>2</sup>C controller to implement the protocol of EEPROM random read.

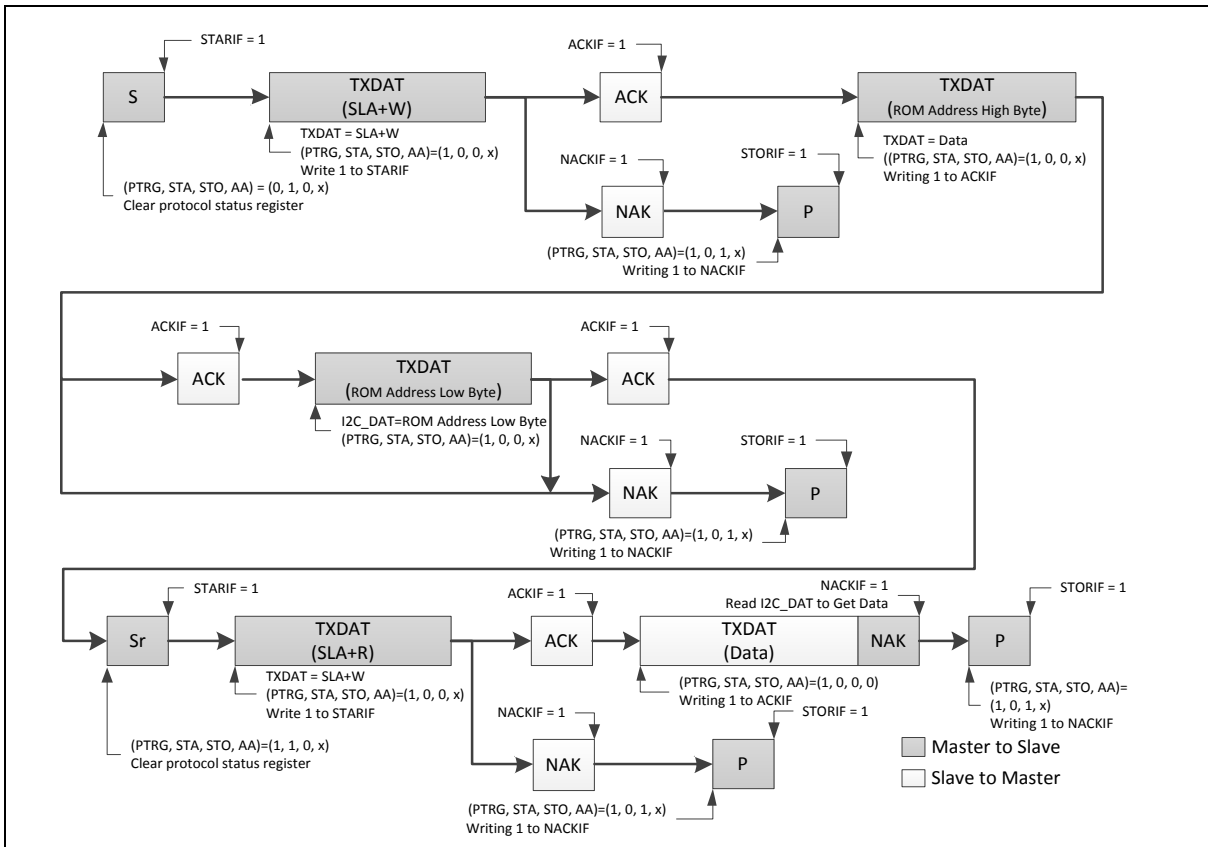


Figure 6.22-21 Protocol of EEPROM Random Read

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

### 6.22.6 Register Map

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

Register	Offset	R/W	Description	Reset Value
<b>UI2C Base Address:</b> $UI2Cx\_BA = 0x4007\_0000 + (0x10\_0000 * x)$ $x = 0, 1$ $UI2C2\_BA = 0x4007\_4000$				
UI2C_CTL	UI2Cx_BA+0x00	R/W	USCI Control Register	0x0000_0000
UI2C_BRGEN	UI2Cx_BA+0x08	R/W	USCI Baud Rate Generator Register	0x0000_3C00
UI2C_LINECTL	UI2Cx_BA+0x2C	R/W	USCI Line Control Register	0x0000_0000
UI2C_TXDAT	UI2Cx_BA+0x30	W	USCI Transmit Data Register	0x0000_0000
UI2C_RXDAT	UI2Cx_BA+0x34	R	USCI Receive Data Register	0x0000_0000
UI2C_DEVADDR0	UI2Cx_BA+0x44	R/W	USCI Device Address Register 0	0x0000_0000
UI2C_DEVADDR1	UI2Cx_BA+0x48	R/W	USCI Device Address Register 1	0x0000_0000
UI2C_ADDRMSK0	UI2Cx_BA+0x4C	R/W	USCI Device Address Mask Register 0	0x0000_0000
UI2C_ADDRMSK1	UI2Cx_BA+0x50	R/W	USCI Device Address Mask Register 1	0x0000_0000
UI2C_WKCTL	UI2Cx_BA+0x54	R/W	USCI Wake-up Control Register	0x0000_0000
UI2C_WKSTS	UI2Cx_BA+0x58	R/W	USCI Wake-up Status Register	0x0000_0000
UI2C_PROTCTL	UI2Cx_BA+0x5C	R/W	USCI Protocol Control Register	0x0000_0000
UI2C_PROTIEN	UI2Cx_BA+0x60	R/W	USCI Protocol Interrupt Enable Register	0x0000_0000
UI2C_PROTSTS	UI2Cx_BA+0x64	R/W	USCI Protocol Status Register	0x0000_0000
UI2C_ADMAT	UI2Cx_BA+0x88	R/W	I <sup>2</sup> C Slave Match Address Register	0x0000_0000
UI2C_TMCTL	UI2Cx_BA+0x8C	R/W	I <sup>2</sup> C Timing Configure Control Register	0x0000_0000

6.22.7 Register Description

USCI Control Register (UI2C\_CTL)

Register	Offset	R/W	Description	Reset Value
UI2C_CTL	UI2Cx_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]	<b>Reserved</b> Reserved.
[2:0]	<p><b>FUNMODE</b></p> <p><b>Function Mode</b> 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.</p> <p>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<sup>2</sup>C protocol is selected.</p> <p><b>Note:</b> Other bit combinations are reserved.</p>



**USCI Baud Rate Generator Register (UI2C\_BRGEN)**

Register	Offset	R/W	Description	Reset Value
UI2C_BRGEN	UI2Cx_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	<p><b>Clock Divider</b></p> <p>This bit field defines the ratio between the protocol clock frequency <math>f_{PROT\_CLK}</math> and the clock divider frequency <math>f_{DIV\_CLK}</math> (<math>f_{DIV\_CLK} = f_{PROT\_CLK} / (CLKDIV+1)</math>).</p> <p><b>Note:</b> In UART function, it can be updated by hardware in the 4<sup>th</sup> falling edge of the input data 0x55 when the auto baud rate function (ABREN(UI2C_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 (UI2C_PROTCTL[24:16]) to calculate the precise baud rate.</p>
[15]	Reserved	Reserved.
[14:10]	DSCNT	<p><b>Denominator for Sample Counter</b></p> <p>This bit field defines the divide ratio of the sample clock <math>f_{SAMP\_CLK}</math>.</p> <p>The divided frequency <math>f_{DS\_CNT} = f_{PDS\_CNT} / (DSCNT+1)</math>.</p> <p><b>Note:</b> 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.</p>
[9:8]	PDSCNT	<p><b>Pre-divider for Sample Counter</b></p> <p>This bit field defines the divide ratio of the clock division from sample clock <math>f_{SAMP\_CLK}</math>. The divided frequency <math>f_{PDS\_CNT} = f_{SAMP\_CLK} / (PDSCNT+1)</math>.</p>
[7:6]	Reserved	Reserved.
[5]	TMCNTSRC	<p><b>Time Measurement Counter Clock Source Selection</b></p> <p>0 = Time measurement counter with <math>f_{PROT\_CLK}</math>.</p> <p>1 = Time measurement counter with <math>f_{DIV\_CLK}</math>.</p>
[4]	TMCNTEN	<p><b>Time Measurement Counter Enable Bit</b></p> <p>This bit enables the 10-bit timing measurement counter.</p> <p>0 = Time measurement counter Disabled.</p> <p>1 = Time measurement counter Enabled.</p>
[3:2]	SPCLKSEL	<p><b>Sample Clock Source Selection</b></p> <p>This bit field used for the clock source selection of a sample clock (<math>f_{SAMP\_CLK}</math>) for the protocol processor.</p>

		<p>00 = <math>f_{\text{SAMP\_CLK}} = f_{\text{DIV\_CLK}}</math>.                      01 = <math>f_{\text{SAMP\_CLK}} = f_{\text{PROT\_CLK}}</math>.                      10 = <math>f_{\text{SAMP\_CLK}} = f_{\text{SCLK}}</math>.                      11 = <math>f_{\text{SAMP\_CLK}} = f_{\text{REF\_CLK}}</math>.</p>
[1]	<b>PTCLKSEL</b>	<p><b>Protocol Clock Source Selection</b>                      This bit selects the source signal of protocol clock (<math>f_{\text{PROT\_CLK}}</math>).                      0 = Reference clock <math>f_{\text{REF\_CLK}}</math>.                      1 = <math>f_{\text{REF\_CLK}2}</math> (its frequency is half of <math>f_{\text{REF\_CLK}}</math>).</p>
[0]	<b>RCLKSEL</b>	<p><b>Reference Clock Source Selection</b>                      This bit selects the source signal of reference clock (<math>f_{\text{REF\_CLK}}</math>).                      0 = Peripheral device clock <math>f_{\text{PCLK}}</math>.                      1 = Reserved.</p>

**USCI Line Control Register (UI2C\_LINECTL)**

Register	Offset	R/W	Description	Reset Value
UI2C_LINECTL	UI2Cx_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	<p><b>Word Length of Transmission</b></p> <p>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.</p> <p>0x0: The data word contains 16 bits located at bit positions [15:0].</p> <p>0x1: Reserved.</p> <p>0x2: Reserved.</p> <p>0x3: Reserved.</p> <p>0x4: The data word contains 4 bits located at bit positions [3:0].</p> <p>0x5: The data word contains 5 bits located at bit positions [4:0].</p> <p>...</p> <p>0xF: The data word contains 15 bits located at bit positions [14:0].</p> <p><b>Note:</b> In I<sup>2</sup>C protocol, the length must be configured as 8 bits.</p>
[7:1]	Reserved	Reserved.
[0]	LSB	<p><b>LSB First Transmission Selection</b></p> <p>0 = The MSB, which bit of transmit/receive data buffer depends on the setting of DWIDTH, is transmitted/received first.</p> <p>1 = The LSB, the bit 0 of data buffer, will be transmitted/received first.</p>

**USCI Transmit Data Register (UI2C\_TXDAT)**

Register	Offset	R/W	Description	Reset Value
UI2C_TXDAT	UI2Cx_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	<b>Transmit Data</b> Software can use this bit field to write 8-bit transmit data for transmission.

**USCI Receive Data Register (UI2C\_RXDAT)**

Register	Offset	R/W	Description	Reset Value
UI2C_RXDAT	UI2Cx_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	<p><b>Received Data</b> This bit field monitors the received data which stored in receive data buffer.</p> <p><b>Note1:</b> In I<sup>2</sup>C protocol, only use RXDAT[7:0].</p>

**USCI Device Address Register (UI2C\_DEVADDR)**

Register	Offset	R/W	Description	Reset Value
UI2C_DEVADDR0	UI2Cx_BA+0x44	R/W	USCI Device Address Register 0	0x0000_0000
UI2C_DEVADDR1	UI2Cx_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	<p><b>Device Address</b></p> <p>In I<sup>2</sup>C protocol, this bit field contains the programmed slave address. If the first received address byte is b1111 0AAX, 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].</p> <p><b>Note:</b> When I<sup>2</sup>C operating in 7-bit address mode, only use DEVADDR[6:0]</p>

**USCI Device Address Mask Register (UI2C\_ADDRMSK) – for I<sup>2</sup>C Only**

Register	Offset	R/W	Description	Reset Value
UI2C_ADDRMSK0	UI2Cx_BA+0x4C	R/W	USCI Device Address Mask Register 0	0x0000_0000
UI2C_ADDRMSK1	UI2Cx_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	<p><b>USCI Device Address Mask</b></p> <p>0 = Mask Disabled (the received corresponding register bit should be exact the same as address register.).</p> <p>1 = Mask Enabled (the received corresponding address bit is don't care.).</p> <p>USCI support multiple address recognition with two 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.</p>

**USCI Wake-up Control Register (UI2C\_WKCTL)**

Register	Offset	R/W	Description	Reset Value
UI2C_WKCTL	UI2Cx_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	<b>Wake-up Address Match Enable Bit</b> 0 = The chip is woken up according data toggle. 1 = The chip is woken up according address match.
[0]	WKEN	<b>Wake-up Enable Bit</b> 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	UI2Cx_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	<b>Wake-up Flag</b> 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 – I<sup>2</sup>C (UI2C\_PROTCTL)**

Register	Offset	R/W	Description	Reset Value
UI2C_PROTCTL	UI2Cx_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	<b>I<sup>2</sup>C Protocol Enable Bit</b> 0 = I <sup>2</sup> C Protocol Disabled. 1 = I <sup>2</sup> C Protocol Enabled.
[30:26]	Reserved	Reserved.
[25:16]	TOCNT	<b>Time-out Clock Cycle</b> 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. <b>Note:</b> The TMCNTSRC (UI2C_BRGEN [5]) must be set zero on I <sup>2</sup> C mode.
[15:10]	Reserved	Reserved.
[9]	MONEN	<b>Monitor Mode Enable Bit</b> This bit enables monitor mode. In monitor mode the SDA output will be put in high impedance mode. This prevents the I <sup>2</sup> C module from outputting data of any kind (including ACK) onto the I <sup>2</sup> C data bus. 0 = Monitor mode Disabled. 1 = Monitor mode Enabled. <b>Note:</b> 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 <sup>2</sup> C clock line.
[8]	SCLOUTEN	<b>SCL Output Enable Bit</b> 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 <sup>2</sup> C interrupt. 0 = SCL output will be forced high due to open drain mechanism. 1 = I <sup>2</sup> C module may act as a slave peripheral just like in normal operation, the I <sup>2</sup> C holds the clock line low until it has had time to clear I <sup>2</sup> C interrupt.
[7:6]	Reserved	Reserved.

[5]	<b>PTRG</b>	<p><b>I<sup>2</sup>C Protocol Trigger</b></p> <p>When a new state is present in the UI2C_PROTSTS register, if the related interrupt enable bits are set, the I<sup>2</sup>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<sup>2</sup>C protocol function will go ahead until the STOP is active or the PROTEN is disabled.</p> <p>0 = I<sup>2</sup>C's stretch disabled and the I<sup>2</sup>C protocol function will go ahead. 1 = I<sup>2</sup>C's stretch active.</p>
[4]	<b>ADDR10EN</b>	<p><b>Address 10-bit Function Enable Bit</b></p> <p>0 = Address match 10 bit function Disabled. 1 = Address match 10 bit function Enabled.</p>
[3]	<b>STA</b>	<p><b>I<sup>2</sup>C START Control</b></p> <p>Setting STA to logic 1 to enter Master mode, the I<sup>2</sup>C hardware sends a START or repeat START condition to bus when the bus is free.</p>
[2]	<b>STO</b>	<p><b>I<sup>2</sup>C STOP Control</b></p> <p>In Master mode, setting STO to transmit a STOP condition to bus then I<sup>2</sup>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<sup>2</sup>C hardware to the defined "not addressed" slave mode when bus error (UI2C_PROTSTS.ERRIF = 1).</p>
[1]	<b>AA</b>	<p><b>Assert Acknowledge Control</b></p> <p>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.</p>
[0]	<b>GCFUNC</b>	<p><b>General Call Function</b></p> <p>0 = General Call Function Disabled. 1 = General Call Function Enabled.</p>

**USCI Protocol Interrupt Enable Register – I<sup>2</sup>C (UI2C\_PROTIEN)**

Register	Offset	R/W	Description	Reset Value
UI2C_PROTIEN	UI2Cx_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	<p><b>Acknowledge Interrupt Enable Bit</b></p> <p>This bit enables the generation of a protocol interrupt if an acknowledge is detected by a master.</p> <p>0 = The acknowledge interrupt Disabled.</p> <p>1 = The acknowledge interrupt Enabled.</p>
[5]	ERRIEN	<p><b>Error Interrupt Enable Bit</b></p> <p>This bit enables the generation of a protocol interrupt if an I<sup>2</sup>C error condition is detected (indicated by ERR (UI2C_PROTSTS [16])).</p> <p>0 = The error interrupt Disabled.</p> <p>1 = The error interrupt Enabled.</p>
[4]	ARBLOIEN	<p><b>Arbitration Lost Interrupt Enable Bit</b></p> <p>This bit enables the generation of a protocol interrupt if an arbitration lost event is detected.</p> <p>0 = The arbitration lost interrupt Disabled.</p> <p>1 = The arbitration lost interrupt Enabled.</p>
[3]	NACKIEN	<p><b>Non - Acknowledge Interrupt Enable Bit</b></p> <p>This bit enables the generation of a protocol interrupt if a non - acknowledge is detected by a master.</p> <p>0 = The non - acknowledge interrupt Disabled.</p> <p>1 = The non - acknowledge interrupt Enabled.</p>
[2]	STORIEN	<p><b>Stop Condition Received Interrupt Enable Bit</b></p> <p>This bit enables the generation of a protocol interrupt if a stop condition is detected.</p> <p>0 = The stop condition interrupt Disabled.</p> <p>1 = The stop condition interrupt Enabled.</p>
[1]	STARIEN	<p><b>Start Condition Received Interrupt Enable Bit</b></p> <p>This bit enables the generation of a protocol interrupt if a start condition is detected.</p>

		<p>0 = The start condition interrupt Disabled.                  1 = The start condition interrupt Enabled.</p>
[0]	<b>TOIEN</b>	<p><b>Time-out Interrupt Enable Bit</b>                  In I<sup>2</sup>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.</p>

**USCI Protocol Status Register – I<sup>2</sup>C (UI2C\_PROTSTS)**

Register	Offset	R/W	Description	Reset Value
UI2C_PROTSTS	UI2Cx_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]	<b>Reserved</b> Reserved.
[19]	<p><b>ERRARBLO</b></p> <p><b>Error Arbitration Lost</b> This bit indicates bus arbitration lost due to bigger noise which is can't be filtered by input processor. The I<sup>2</sup>C can send start condition when ERRARBLO is set. Thus this bit doesn't be cared on slave mode.</p> <p>0 = The bus is normal status for transmission. 1 = The bus is error arbitration lost status for transmission.</p> <p><b>Note:</b> This bit has no interrupt signal, and it will be cleared automatically by hardware.</p>
[18]	<b>Reserved</b> Reserved.
[17]	<p><b>WRSTSWK</b></p> <p><b>Read/Write Status Bit in Address Wakeup Frame</b> 0 = Write command be record on the address match wakeup frame. 1 = Read command be record on the address match wakeup frame.</p>
[16]	<p><b>WKAKDONE</b></p> <p><b>Wakeup Address Frame Acknowledge Bit Done</b> 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.</p> <p><b>Note:</b> This bit can't release when WKIF is set.</p>
[15]	<p><b>SLAREAD</b></p> <p><b>Slave Read Request Status</b> This bit indicates that a slave read request has been detected. 0 = A slave read request has not been detected. 1 = A slave read request has been detected.</p> <p><b>Note:</b> This bit has no interrupt signal, and it will be cleared automatically by hardware.</p>
[14]	<p><b>SLASEL</b></p> <p><b>Slave Select Status</b> 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.</p> <p><b>Note:</b> This bit has no interrupt signal, and it will be cleared automatically by hardware.</p>

[13]	ACKIF	<p><b>Acknowledge Received Interrupt Flag</b></p> <p>This bit indicates that an acknowledge has been received in master mode. A protocol interrupt can be generated if UI2C_PROTCTL.ACKIEN = 1.</p> <p>0 = An acknowledge has not been received. 1 = An acknowledge has been received.</p> <p>It is cleared by software writing 1 into this bit</p>
[12]	ERRIF	<p><b>Error Interrupt Flag</b></p> <p>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_PROTCTL.ERRIEN = 1.</p> <p>0 = An I<sup>2</sup>C error has not been detected. 1 = An I<sup>2</sup>C error has been detected.</p> <p>It is cleared by software writing 1 into this bit</p> <p><b>Note:</b> This bit is set when slave mode, user must write one into STO register to the defined “not addressed” slave mode.</p>
[11]	ARBLOIF	<p><b>Arbitration Lost Interrupt Flag</b></p> <p>This bit indicates that an arbitration has been lost. A protocol interrupt can be generated if UI2C_PROTCTL.ARBLOIEN = 1.</p> <p>0 = An arbitration has not been lost. 1 = An arbitration has been lost.</p> <p><b>Note:</b> This bit is cleared by software writing 1 to it.</p>
[10]	NACKIF	<p><b>Non - Acknowledge Received Interrupt Flag</b></p> <p>This bit indicates that a non - acknowledge has been received in master mode. A protocol interrupt can be generated if UI2C_PROTCTL.NACKIEN = 1.</p> <p>0 = A non - acknowledge has not been received. 1 = A non - acknowledge has been received.</p> <p><b>Note:</b> This bit is cleared by software writing 1 to it.</p>
[9]	STORIF	<p><b>Stop Condition Received Interrupt Flag</b></p> <p>This bit indicates that a stop condition has been detected on the I<sup>2</sup>C bus lines. A protocol interrupt can be generated if UI2C_PROTCTL.STORIEN = 1.</p> <p>0 = A stop condition has not yet been detected. 1 = A stop condition has been detected.</p> <p><b>Note:</b> This bit is cleared by software writing 1 to it.</p>
[8]	STARIF	<p><b>Start Condition Received Interrupt Flag</b></p> <p>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.</p> <p>A protocol interrupt can be generated if UI2C_PROTCTL.STARIEN = 1.</p> <p>0 = A start condition has not yet been detected. 1 = A start condition has been detected.</p> <p><b>Note:</b> This bit is cleared by software writing 1 to it.</p>
[7]	Reserved	Reserved.
[6]	ONBUSY	<p><b>On Bus Busy</b></p> <p>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</p> <p>0 = The bus is IDLE (both SCLK and SDA High). 1 = The bus is busy.</p>
[5]	TOIF	<p><b>Time-out Interrupt Flag</b></p>

		<p>0 = A time-out interrupt status has not occurred.                  1 = A time-out interrupt status has occurred.  <b>Note:</b> This bit is cleared by software writing 1 to it.</p>
[0]	<b>Reserved</b>	Reserved.



**USCI Slave match Address Register (UI2C\_ADMAT)**

Register	Offset	R/W	Description	Reset Value
UI2C_ADMAT	UI2Cx_BA+0x88	R/W	I <sup>2</sup> 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	<b>USCI Address 1 Match Status</b> 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	<b>USCI Address 0 Match Status</b> 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	UI2Cx_BA+0x8C	R/W	I <sup>2</sup> C Timing Configure 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				HTCTL			
7	6	5	4	3	2	1	0
HTCTL		STCTL					

Bits	Description	
[31:8]	Reserved	Reserved.
[11:6]	HTCTL	<p><b>Hold Time Configure Control</b></p> <p>This field is used to generate the delay timing between SCL falling edge SDA edge in transmission mode.</p> <p>The delay hold time is numbers of peripheral clock = HTCTL x f<sub>PCLK</sub>.</p>
[5:0]	STCTL	<p><b>Setup Time Configure Control</b></p> <p>This field is used to generate a delay timing between SDA edge and SCL rising edge in transmission mode..</p> <p>The delay setup time is numbers of peripheral clock = STCTL x f<sub>PCLK</sub>.</p>

## 6.23 UART Interface Controller (UART)

### 6.23.1 Overview

The NUC126 series provides three channels of Universal Asynchronous Receiver/Transmitters (UART). The UART controller performs Normal Speed UART and supports flow control function. The UART controller performs a serial-to-parallel conversion on data received from the peripheral and a parallel-to-serial conversion on data transmitted from the CPU. Each UART controller channel supports ten types of interrupts. The UART controller also supports IrDA SIR, LIN and RS-485 function modes and auto-baud rate measuring function.

### 6.23.2 Features

- Full-duplex asynchronous communications
- Separates receive and transmit 16/16 bytes 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
- 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
  - Support for 3/16 bit duration for normal mode
- Supports LIN function mode
  - Supports LIN master/slave mode
  - Supports programmable break generation function for transmitter
  - Supports break detection function for receiver
- 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
- Support PDMA transfer function

6.23.3 Block Diagram

The UART clock control and block diagram are shown in Figure 6.23-1 and Figure 6.23-2 respectively.

**Note:** The frequency of UARTx\_CLK should not be greater than 30 times HCLK.

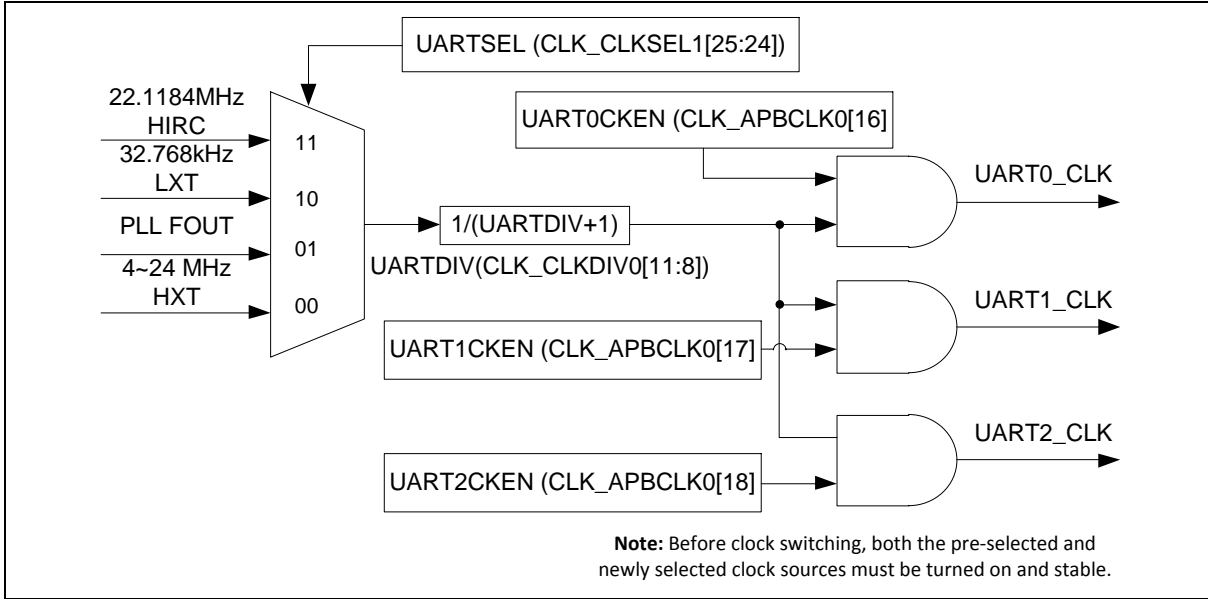


Figure 6.23-1 UART Clock Control Diagram

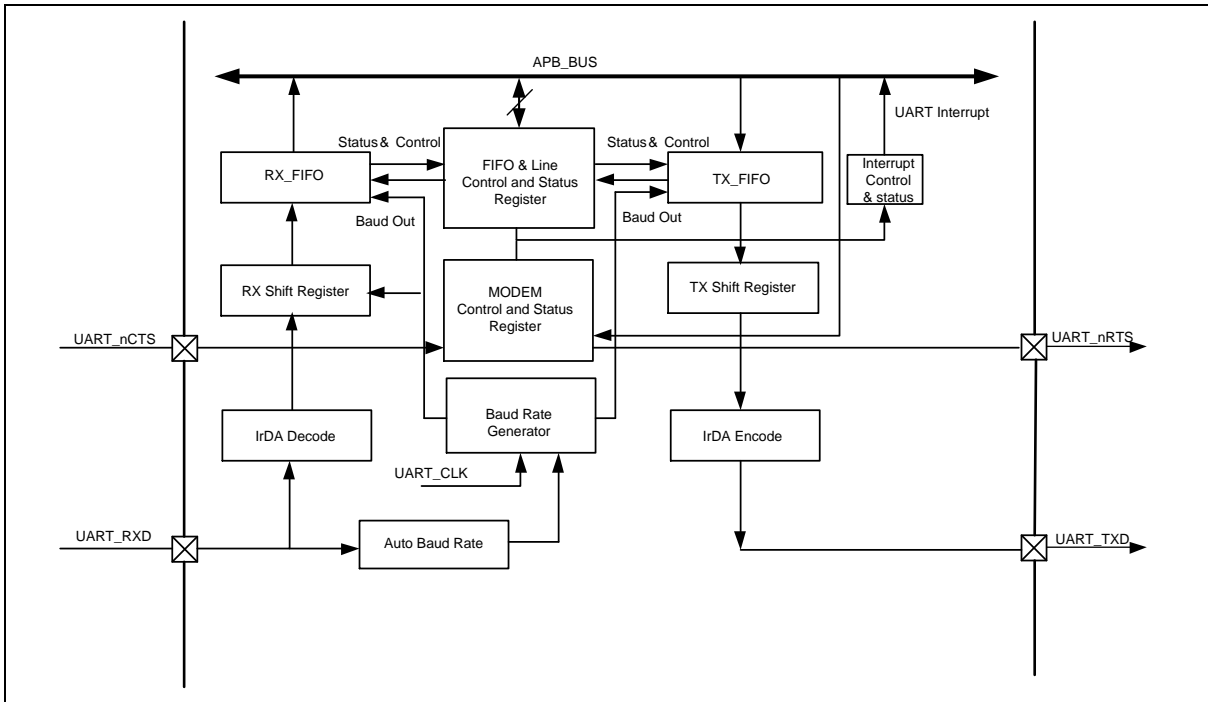


Figure 6.23-2 UART Block Diagram

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, Receive Data Available Interrupt (RDAINT), Transmit Holding Register Empty Interrupt (THERINT), Transmitter Empty Interrupt (TXENDINT), Receive Line Status Interrupt (parity error or framing error or break interrupt) (RLSINT), MODEM Status Interrupt (MODEMINT), Receiver Buffer Time-out Interrupt (RXTOINT), Buffer Error Interrupt (BUFERRINT), LIN Bus Interrupt (LININT), Wake-up Interrupt (WKINT) and Auto-Baud Rate Interrupt (ABRINT). 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.
THERINT	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.

LININT	LIN Bus Interrupt.
WKINT	Wake-up Interrupt.
ABRINT	Auto-Baud Rate Interrupt.

Table 6.23-1 UART Interrupt

### 6.23.4 Basic Configuration

The UART Interface Controller Pin description is shown in Table 6.23-2:

Pin	Type	Description
UART_TXD	Output	UART transmit
UART_RXD	Input	UART receive
UART_nCTS	Input	UART modem clear to send
UART_nRTS	Output	UART modem request to send

Table 6.23-2 UART Interface Controller Pin

#### 6.23.4.1 Basic Configurations of UART0

- Clock Source Configuration
  - Select the source of UART0 peripheral clock on UARTSEL (CLK\_CLKSEL1[25:24]).
  - Select the clock divider number of UART0 peripheral clock on UARTDIV (CLK\_CLKDIV0[11:8]).
  - Enable UART0 peripheral clock in UART0CKEN (CLK\_APBCLK0[16]).
- Reset Configuration
  - Reset UART0 controller in UART0RST (SYS\_IPRST1[16]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
UART0	UART0_RXD	PA.3	MFP2
		PD.0, PD.6, PD.9, PD.13, PE.6	MFP3
	UART0_TXD	PA.2	MFP2
		PD.1, PD.12, PE.7	MFP3
	UART0_nCTS	PA.2, PD.8, PD.14	MFP3
UART0_nRTS	PA.3, PC.8, PD.15	MFP3	

#### 6.23.4.2 Basic Configurations of UART1

- Clock Source Configuration
  - Select the source of UART1 peripheral clock on UARTSEL (CLK\_CLKSEL1[25:24]).
  - Select the clock divider number of UART1 peripheral clock on UARTDIV (CLK\_CLKDIV0[11:8]).
  - Enable UART1 peripheral clock in UART1CKEN (CLK\_APBCLK0[17]).

- Reset Configuration  
Reset UART1 controller in UART1RST (SYS\_IPRST1[17]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
UART1	UART1_RXD	PE.9	MFP1
		PA.1, PA.9, PE.13	MFP3
		PA.12, PB.2	MFP4
	UART1_TXD	PE.8	MFP1
		PA.0, PA.8, PE.12	MFP3
		PA.13, PB.3	MFP4
	UART1_nCTS	PA.0	MFP1
		PA.10, PE.10	MFP3
		PB.4	MFP4
	UART1_nRTS	PA.1	MFP1
		PA.11, PE.11	MFP3
		PB.8, PE.2	MFP4

6.23.4.3 Basic Configurations of UART2

- Clock Source Configuration
  - Select the source of UART2 peripheral clock on UARTSEL (CLK\_CLKSEL1[25:24]).
  - Select the clock divider number of UART2 peripheral clock on UARTDIV (CLK\_CLKDIV0[11:8]).
  - Enable UART2 peripheral clock in UART2CKEN (CLK\_APBCLK0[18]).
- Reset Configuration  
Reset UART2 controller in UART2RST (SYS\_IPRST1[18]).
- Pin Configuration

Group	Pin Name	GPIO	MFP
UART2	UART2_RXD	PA.12, PB.0, PC.3	MFP3
		PE.3	MFP4
		PB.4, PB.5	MFP9
	UART2_TXD	PA.13, PB.1, PC.2	MFP3
		PD.6	MFP4
	UART2_nCTS	PA.14, PC.0	MFP3
		PD.5	MFP4
	UART2_nRTS	PA.15, PC.1	MFP3

		PD.4	MFP4
--	--	------	------

### 6.23.5 Functional Description

The UART controller supports four function modes including UART, IrDA, LIN and RS-485 mode. User can select a function by setting the UART\_FUNCSEL register. The four function modes will be described in following section.

#### 6.23.5.1 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 6.23-3 list the UART baud rate equations in the various conditions. Table 6.23-4 and Table 6.23-5 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 detail register description is shown in UART\_BAUD register. There are three setting mode. 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 $\geq 8$ .
Mode 2	1	1	$UART\_CLK / (BRD+2)$ . If $UART\_CLK \leq 3 * HCLK$ , BRD must $\geq 9$ . 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 $\geq 11$ . if $4 * HCLK < UART\_CLK \leq 5 * HCLK$ , BRD must $\geq 14$ . (If the UART_CLK is selected from LXT, BRD can be greater than or equal to 1)

Table 6.23-3 UART Controller Baud Rate Equation Table

UART Peripheral Clock = 22.1184 MHz			
Baud Rate	Mode 0	Mode 1	Mode 2
921600	Not support	BRD=0, EDIVM1=11	BRD=22
460800	BRD=1	BRD=1, EDIVM1 =15 BRD=2, EDIVM1 =11	BRD=46
230400	BRD =4	BRD =4, EDIVM1 =15 BRD =6, EDIVM1 =11	BRD =94
115200	BRD =10	BRD =10, EDIVM1 =15 BRD =14, EDIVM1 =11	BRD =190
57600	BRD =22	BRD =22, EDIVM1 =15 BRD =30, EDIVM1 =11	BRD =382
38400	BRD =34	BRD =62, EDIVM1 =8 BRD =46, EDIVM1 =11 BRD =34, E EDIVM1 =15	BRD =574
19200	BRD =70	BRD =126, EDIVM1 =8 BRD =94, EDIVM1 =11 BRD =70, EDIVM1 =15	BRD =1150



9600	BRD =142	BRD =254, EDIVM1 =8 BRD =190, EDIVM1 =11 BRD =142, EDIVM1 =15	BRD =2302
4800	BRD =286	BRD =510, EDIVM1 =8 BRD =382, EDIVM1 =11 BRD =286, EDIVM1 =15	BRD =4606

Table 6.23-4 UART Controller Baud Rate Parameter Setting Example Table

UART Peripheral Clock = 22.1184 MHz			
Baud Rate	UART_BAUD Value		
	Mode 0	Mode 1	Mode 2
921600	Not support	0x2B00_0000	0x3000_0016
460800	0x0000_0001	0x2F00_0001 0x2B00_0002	0x3000_002E
230400	0x0000_0004	0x2F00_0004 0x2B00_0006	0x3000_005E
115200	0x0000_000A	0x2F00_000A 0x2B00_000E	0x3000_00BE
57600	0x0000_0016	0x2F00_0016 0x2B00_001E	0x3000_017E
38400	0x0000_0022	0x2800_003E 0x2B00_002E 0x2F00_0022	0x3000_023E
19200	0x0000_0046	0x2800_007E 0x2B00_005E 0x2F00_0046	0x3000_047E
9600	0x0000_008E	0x2800_00FE 0x2B00_00BE 0x2F00_008E	0x3000_08FE
4800	0x0000_011E	0x2800_01FE 0x2B00_017E 0x2F00_011E	0x3000_11FE

Table 6.23-5 UART Controller Baud Rate Register Setting Example Table

6.23.5.2 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 BRCOMDEC bit (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 UART\_DAT[7:0] and BRCOMP[8] is used to define the parity bit.

**Example:**

(1). UART's peripheral clock = 32.768K and baud rate is 9600

Baud rate is 9600, UART peripheral clock is 32.768K → 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 =0,

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 6.23-6 Baud Rate Compensation Example Table 1

the BRCOMP (UART\_BRCOMP[8:0]) can be set as 9'b010100101 = 0xa5.

(2). UART's peripheral clock = 32.768K and baud rate is 4800

Baud rate is 4800, UART peripheral clock is 32.768K → 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 =1,

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 6.23-7 Baud Rate Compensation Example Table 2

the BRCOMP (UART\_BRCOMP[8:0]) can be set as 9'b010000010 = 0x82.

### 6.23.5.3 UART Controller Auto-Baud Rate Function Mode

The 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 1<sup>st</sup> 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 1<sup>st</sup> rising edge is calculated by setting ABRDBITS (UART\_ALTCTL[20:19]). Setting ABRDEN (UART\_ALTCTL[18]) is 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 1<sup>st</sup> 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. Once the auto-baud rate measurement is finished, the ABRDIF (UART\_FIFOSTS[1]) is set. When auto-baud rate counter is overflow, ABRDIF (UART\_FIFOSTS[1]) or ABRDIF (UART\_FIFOSTS[2]) is set. ABRDIF (UART\_FIFOSTS[1]) or ABRDIF (UART\_FIFOSTS[2]) cause the auto-baud rate flag ABRIF (UART\_ALTCTL[17]) is generated. If the ABRIEN (UART\_INTEN[18]) is enabled, ABRIF (UART\_ALTCTL[17]) cause the auto-baud rate interrupt ABRINT (UART\_INTSTS[31]) is generated.

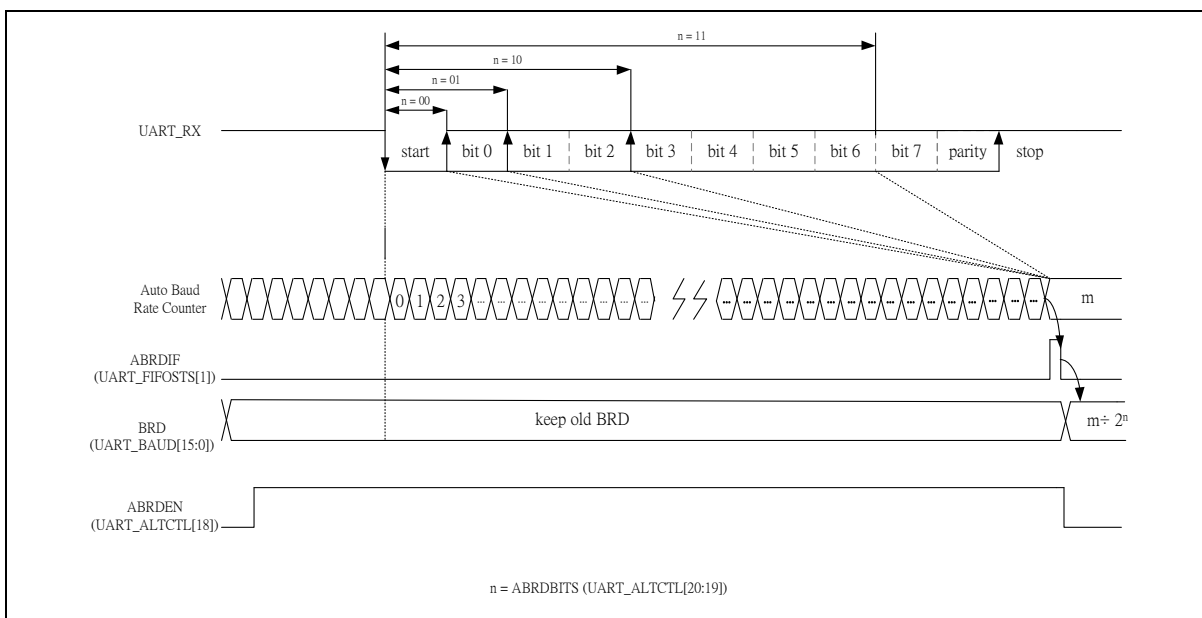


Figure 6.23-3 Auto-Baud Rate Measurement

#### Programming Sequence Example:

1. Program ABRDBITS (UART\_ALTCTL[20:19]) to determine UART RX data 1<sup>st</sup> 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. After ABRDIF (UART\_FIFOSTS[1]) is set, the auto-baud rate measurement is finished.
5. Operate UART transmit and receive action.
6. Set ABRDIF (UART\_FIFOSTS[2]) if auto-baud rate counter is overflow.
7. Go to Step 3.

#### 6.23.5.4 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 6.23-4.

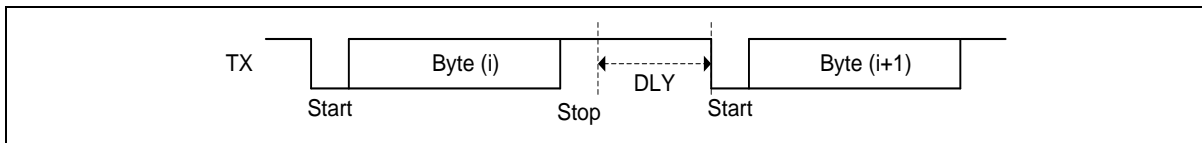


Figure 6.23-4 Transmit Delay Time Operation

#### 6.23.5.5 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, LIN and RS-485 mode support FIFO control and status function.

#### 6.23.5.6 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]) is 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]) is 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]) is generated.

##### nCTS Wake-up Case 1 (nCTS Transition from Low to High)

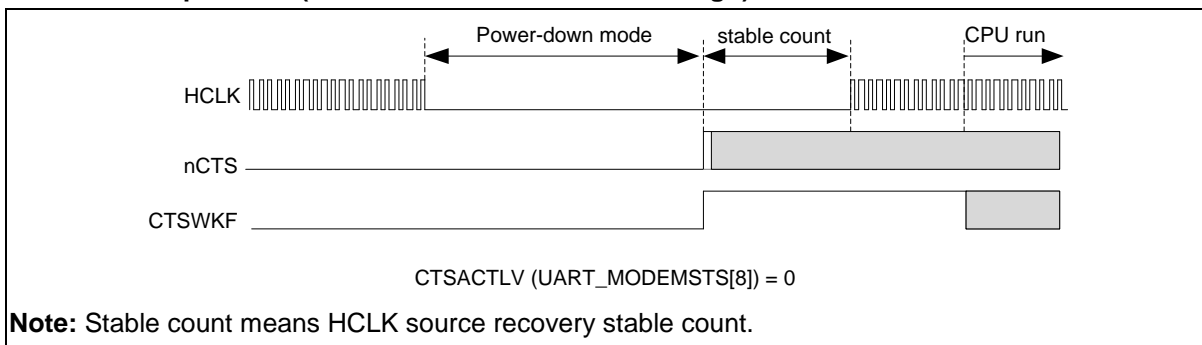


Figure 6.23-5 UART nCTS Wake-up Case1

##### nCTS Wake-up Case 2 (nCTS Transition from High to Low)

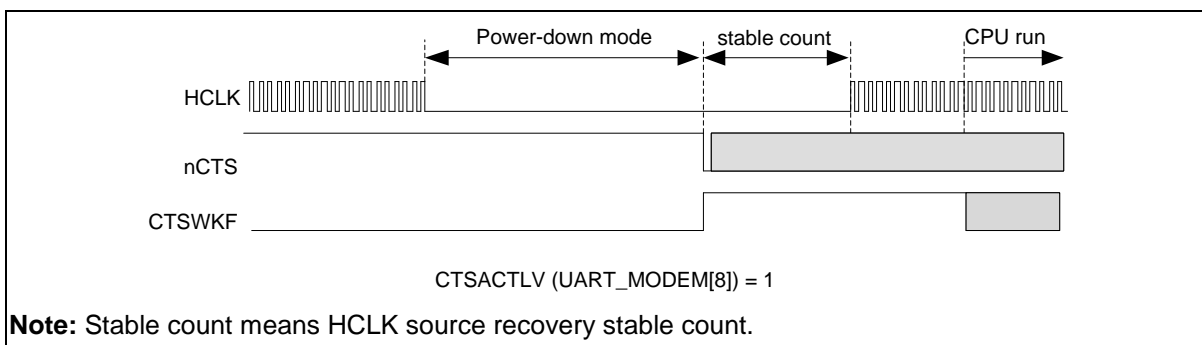


Figure 6.23-6 UART nCTS Wake-up Case2

### 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 1<sup>st</sup> bit (start bit) when the system is wakeup 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]) is generated.

**Note1:** The UART controller clock source should be selected as HIRC and the compensation time for start bit is about 35us. It means that the value of STCOMP (UART\_DWKCOMP[15:0]) can be set as 774.

**Note2:** The value of BRD(UART\_BAUD[15:0]) should be greater than STCOMP (UART\_DWKCOMP[15:0]).

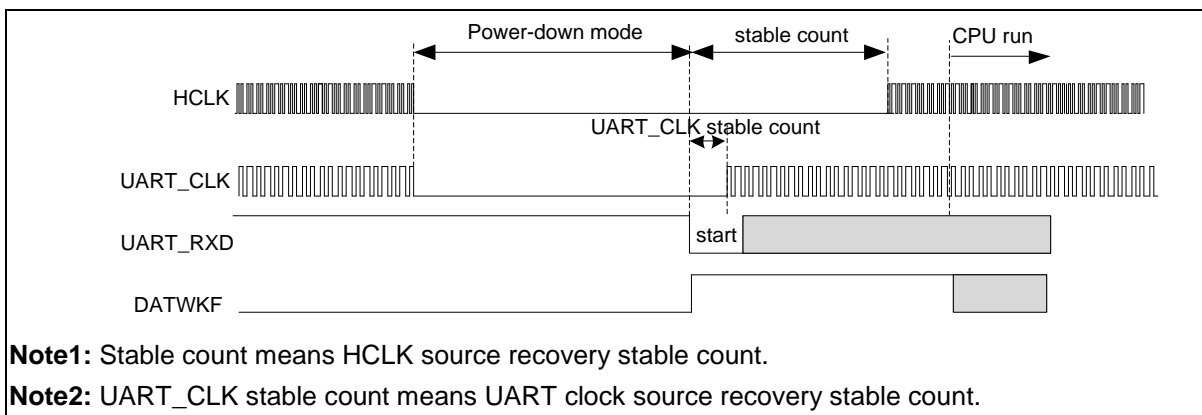


Figure 6.23-7 UART Data Wake-up

### 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 RFRTWKF (UART\_WKSTS[2]) is generated.

**Note:** The UART controller clock source should be selected as LXT in Power-down mode to receive data.

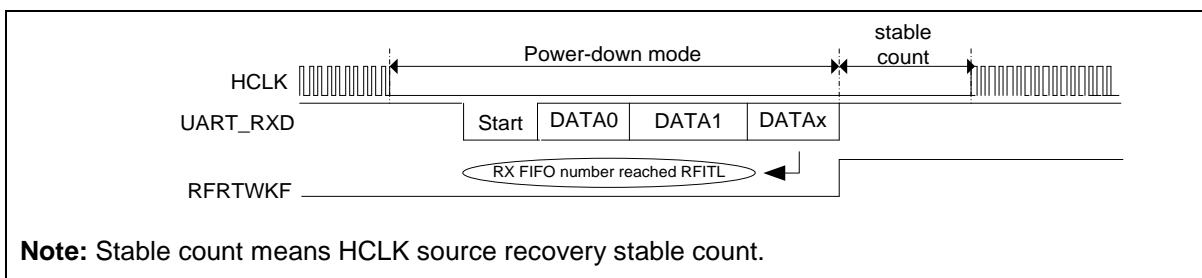


Figure 6.23-8 UART Received Data FIFO Reached Threshold Wake-up

### 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 ADDR MV (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 ADDR MV (UART\_ALTCTL[31:24]) that cause the RS485 address match (AAD mode) wake-up flag RS485WKF (UART\_WKSTS[3]) is generated.

**Note:** The UART controller clock source should be selected as LXT in Power-down mode to receive data.

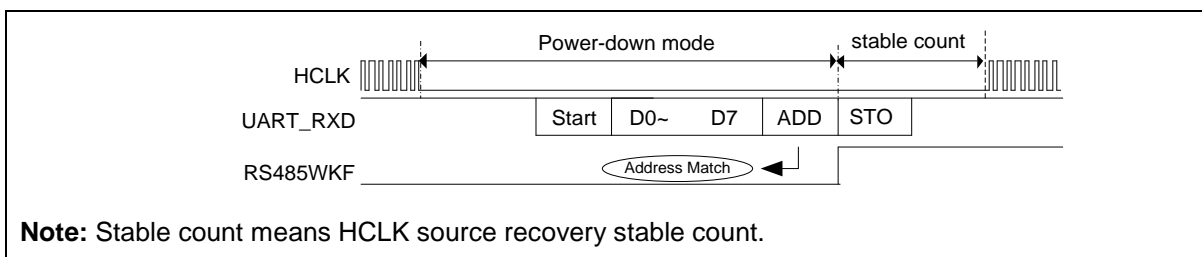


Figure 6.23-9 UART RS-485 AAD Mode Address Match Wake-up

### 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 WKTOUEN (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 to the time-out value TOIC (UART\_TOUT[7:0]), it can wake-up the system. If the WKTOUEN (UART\_WKCTL[4]) is enabled, when the time-out counter equals to the time-out value TOIC (UART\_TOUT[7:0]) that cause the Received Data FIFO threshold time-out wake-up wake-up flag TOUTWKF (UART\_WKSTS[4]) is generated.

**Note:** The UART controller clock source should be selected as LXT in Power-down mode to receive data.

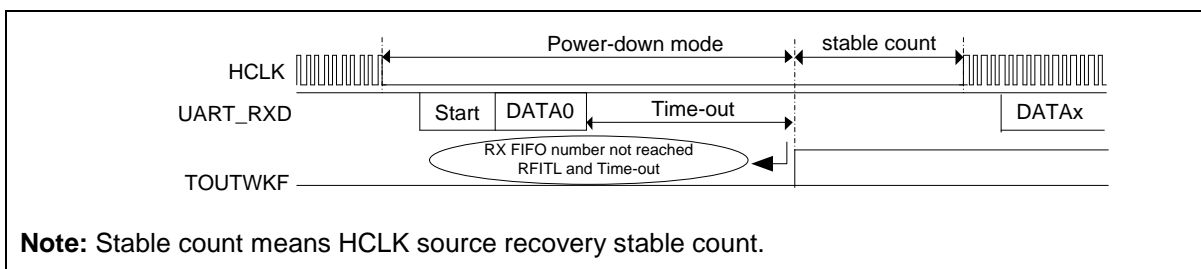


Figure 6.23-10 UART Received Data FIFO Threshold Time-out Wake-up

### 6.23.5.7 UART Controller Interrupt and Status

Each UART controller supports ten types of interrupts including:

- Receive Data Available Interrupt (RDAINTE)
- Transmit Holding Register Empty Interrupt (THERINT)
- 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)
- LIN Bus Interrupt (LININT)
  - LIN Break Detection Flag (BRKDETF)
  - Bit Error Detect Status Flag (BITEF)
  - LIN Slave ID Parity Error Flag (SLVIDPEF)
  - LIN Slave Header Error Flag (SLVHEF)
  - LIN Slave Header Detection Flag (SLVHDETF)
- 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)

Table 6.23-8 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	<b>RDAIF</b>	N/A	Read UART_DAT
Transmit Register Holding Empty Interrupt	THERINT	THREIEN	<b>THREIF</b>	N/A	Write UART_DAT
Transmitter Empty Interrupt	TXENDINT	TXENDIEN	<b>TXENDIF</b>	N/A	Write UART_DAT
Receive Line Status Interrupt	RLSINT	RLSIEN	<b>RLSIF</b>	RLSIF = BIF	Write '1' to BIF
				RLSIF = FEF	Write '1' to FEF
				RLSIF = PEF	Write '1' to PEF
				RLSIF ADDRDETf	Write '1' to ADDRDETf
Modem Status Interrupt	MODEMINT	MODEMIEN	<b>MODEMIF</b>	MODEMIF CTSDETF	Write '1' to CTSDETF
Receiver Buffer Time-out Interrupt	RXTOINT	RXTOIEN	<b>RXTOIF</b>	N/A	Read UART_DAT
Buffer Error Interrupt	BUFERRINT	BUFERRIEN	<b>BUFERRIF</b>	BUFERRIF TXOVIF	Write '1' to TXOVIF
				BUFERRIF RXOVIF	Write '1' to RXOVIF
LIN Bus Interrupt	LININT	LINIEN	<b>LINIF</b>	LINIF = BRKDETf	Write '1' to LINIF and Write '1' to BRKDETf
				LINIF = BITEF	Write '1' to LINIF and Write '1' to BITEF
				LINIF = SLVIDPEF	Write '1' to LINIF and Write '1' to SLVIDPEF
				LINIF = SLVHEF	Write '1' to LINIF and Write '1' to SLVHEF
				LINIF = SLVHDETf	Write '1' to LINIF and Write '1' to SLVHDETf
Wake-up Interrupt	WKINT	WKIEN	<b>WKIF</b>	WKIF = CTSWKF	Write '1' to CTSWKF
				WKIF = DATWKF	Write '1' to DATWKF



				<b>WKIF = RFRTWKF</b>	<b>Write '1' to RFRTWKF</b>
				<b>WKIF = RS485WKF</b>	<b>Write '1' to RS485WKF</b>
				<b>WKIF = TOUTWKF</b>	<b>Write '1' to TOUTWKF</b>
Auto-Baud Interrupt	Rate	ABRINT	ABRIEN	<b>ABRIF = ABRDIF</b>	<b>Write '1' to ABRDIF</b>
				<b>ABRIF = ABRDIOIF</b>	<b>Write '1' to ABRDIOIF</b>

Table 6.23-8 UART Controller Interrupt Source and Flag List

6.23.5.8 UART Function Mode

The UART controller provides UART function (Setting FUNCSEL (UART\_FUNCSEL [1:0]) to '00' to enable UART function mode). The UART baud rate is up to 1 Mbps.

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 deasserted.

**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 6.23-9 and Table 6.23-10 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 6.23-9 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 6.23-10 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.

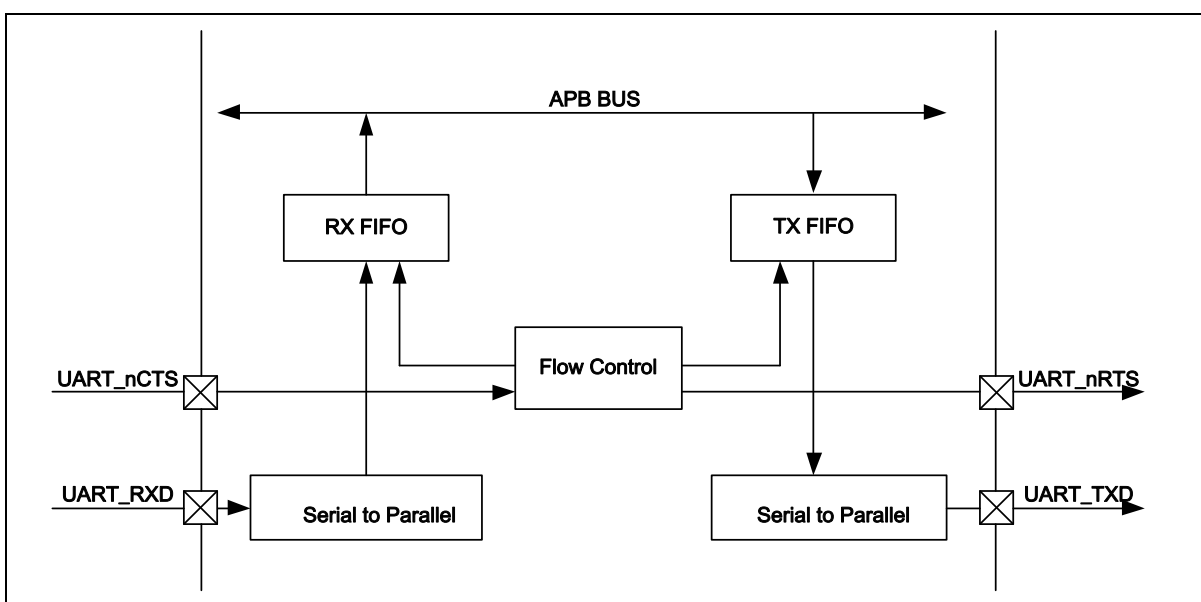


Figure 6.23-11 Auto-Flow Control Block Diagram

Figure 6.23-12 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.

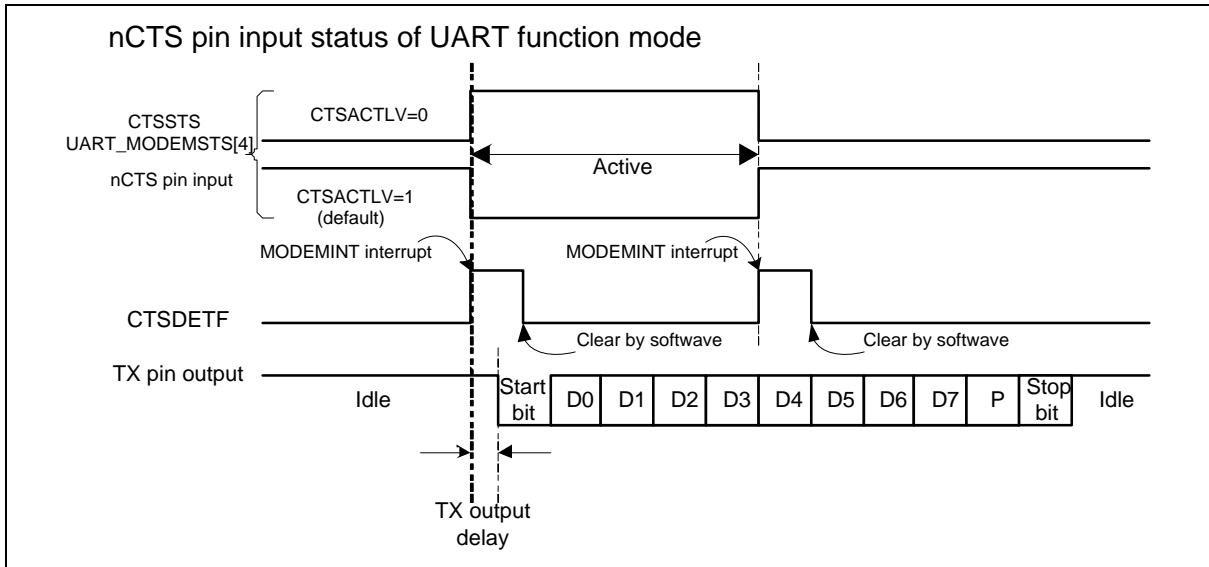


Figure 6.23-12 UART nCTS Auto-Flow Control Enabled

As shown in Figure 6.23-13, 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.

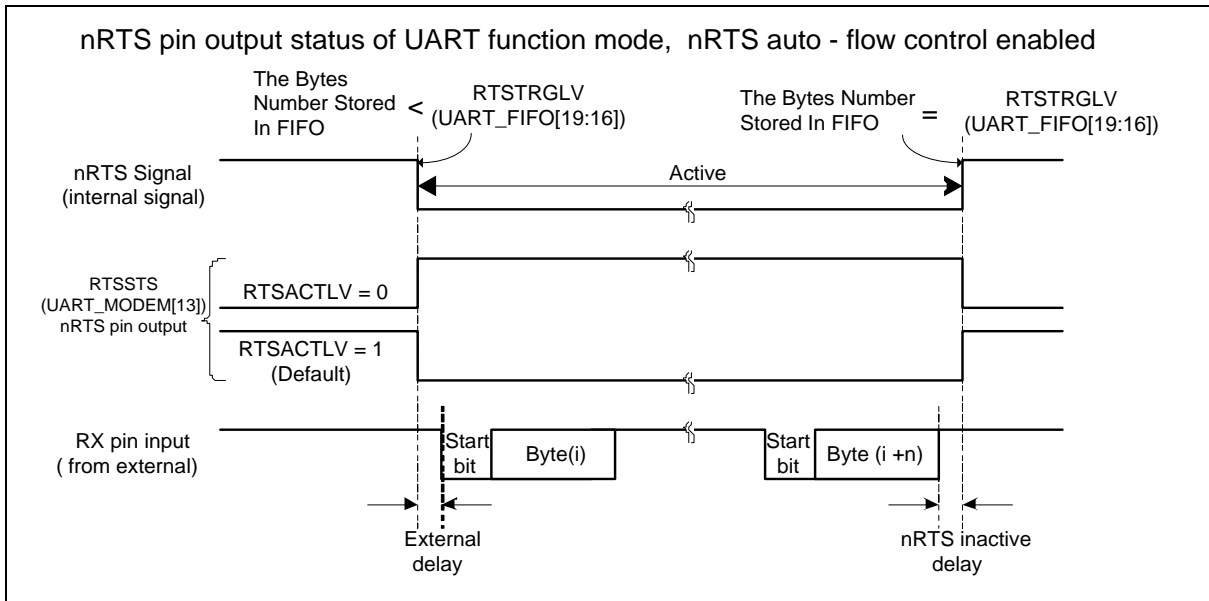


Figure 6.23-13 UART nRTS Auto-Flow Control Enabled

As shown in Figure 6.23-14, 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.

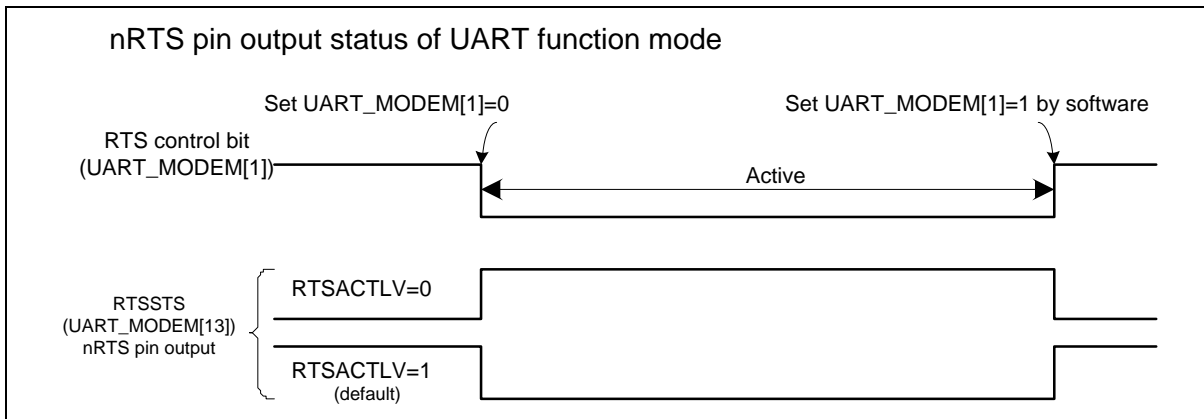


Figure 6.23-14 UART nRTS Auto-Flow with Software Control

### 6.23.5.9 IrDA Function Mode

The UART controller also provides Serial IrDA (SIR, Serial Infrared) function (Setting UART\_FUNCSEL [1:0] to '10' 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 6.23-15.

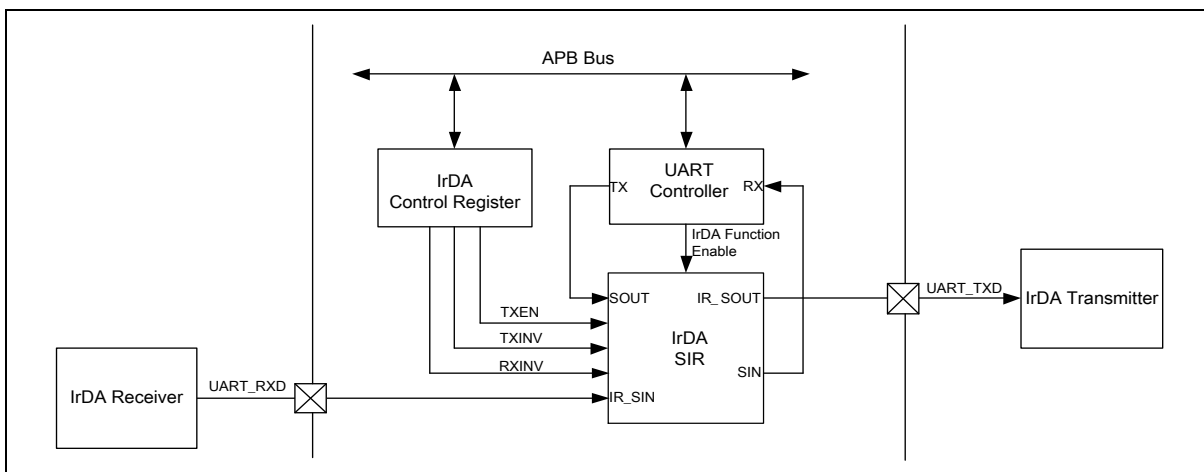


Figure 6.23-15 IrDA Control Block Diagram

### 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 6.23-16 is IrDA encoder/decoder waveform.

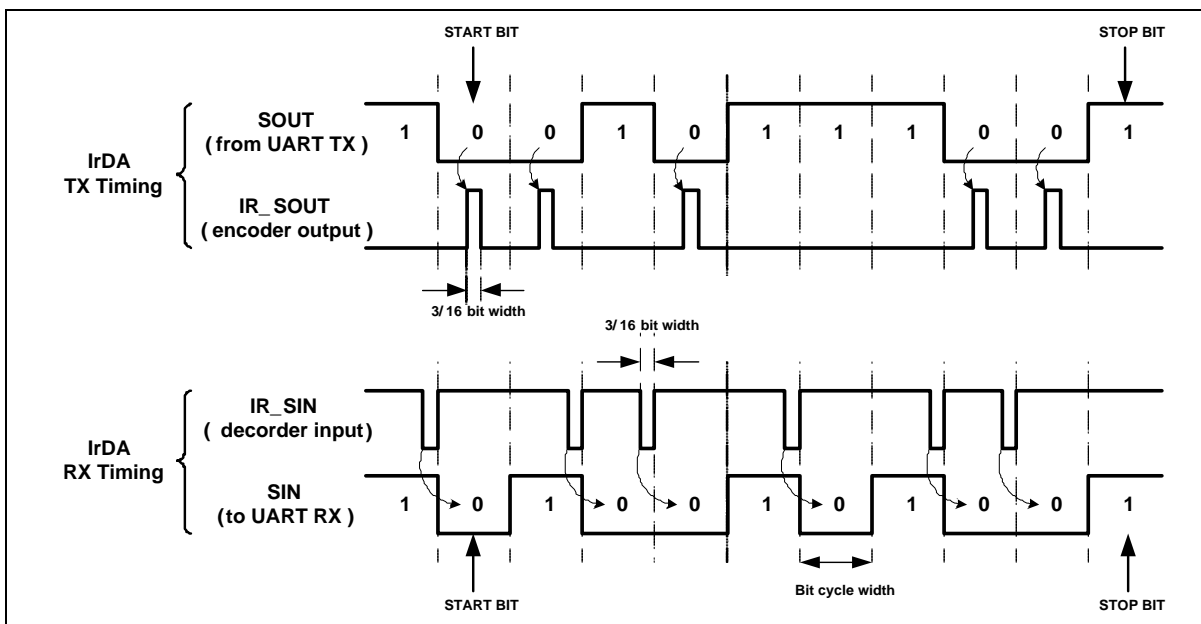


Figure 6.23-16 IrDA TX/RX Timing Diagram

**6.23.5.10 LIN Function Mode (Local Interconnection Network)**

The UART Controller supports LIN function. Setting FUNCSEL (UART\_FUNCSEL[1:0]) to '01' to select LIN mode operation. The UART Controller supports LIN break/delimiter generation and break/delimiter detection in LIN master mode, and supports header detection and automatic resynchronization in LIN Slave mode.

**Structure of LIN Frame**

According to the LIN protocol, all information transmitted is packed as frames; a frame consists of a header (provided by the master task) and a response (provided by a slave task). The header (provided by the master task) consists of a break field and a sync field followed by a frame identifier (frame ID). The frame identifier uniquely defines the purpose of the frame. The slave task is appointed for providing the response associated with the frame ID. The response consists of a data field and a checksum field. Figure 6.23-17 is the structure of LIN Frame.

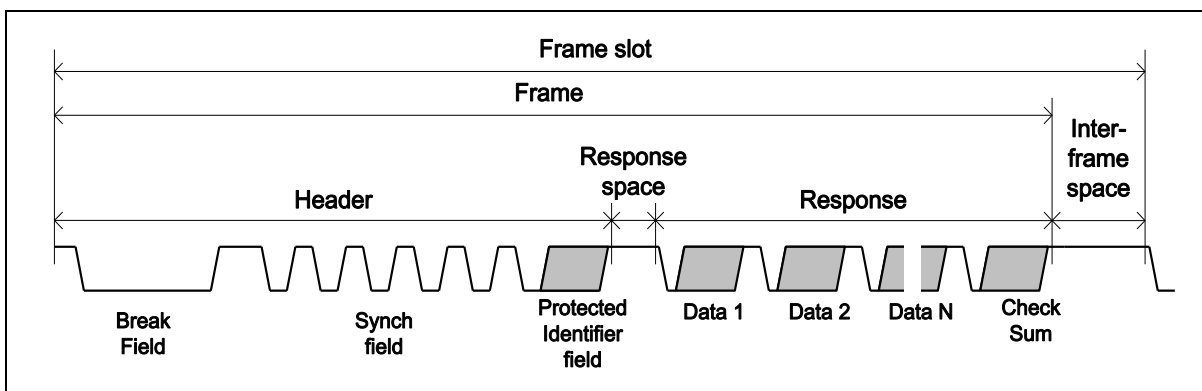


Figure 6.23-17 Structure of LIN Frame

### Structure of LIN Byte

In LIN mode, each byte field is initiated by a START bit with value 0 (dominant), followed by 8 data bits and no parity bit, LSB is first and ended by 1 stop bit with value 1 (recessive) in accordance with the LIN standard. The structure of Byte is shown in Figure 6.23-18.

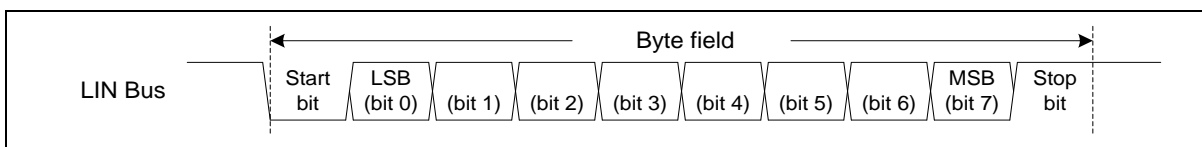


Figure 6.23-18 Structure of LIN Byte

### LIN Master Mode

The UART Controller supports LIN Master mode. To enable and initialize the LIN Master mode, the following steps are necessary:

1. Setting the UART\_BAUD register to select the desired baud rate.
2. Setting WLS (UART\_LINE[1:0]) to '11' to configure the word length with 8 bits, clearing PBE (UART\_LINE[3]) bit to disable parity check and clearing NSB (UART\_LINE[2]) bit to configure with one stop bit.
3. Setting FUNCSEL (UART\_FUNCSEL[1:0]) to '01' to select LIN function mode operation.

A complete header consists of a break field and sync field followed by a frame identifier (frame ID). The UART controller can be selected header sending by three header selected modes. The header selected mode can be "break field" or "break field and sync field" or "break field, sync field and frame ID field" by setting HSEL (UART\_LINCTL[23:22]). If the selected header is "break field", software must handle the following sequence to send a complete header to bus by filling sync data (0x55) and frame ID data to the UART\_DAT register. If the selected header is "break field and sync field", software must handle the sequence to send a complete header to bus by filling the frame ID data to UART\_DAT register, and if the selected header is "break field, sync field and frame ID field", hardware will control the header sending sequence automatically but software must filled frame ID data to PID (UART\_LINCTL [31:24]). When operating in header selected mode in which the selected header is "break field, sync field and frame ID field", the frame ID parity bit can be calculated by software or hardware depending whether the IDPEN (UART\_LINCTL[9]) bit is set or not.

HSEL	Break Field	Sync Field	ID Field
0	Generated by Hardware	Handled by Software	Handled by Software
1	Generated by Hardware	Generated by Hardware	Handled by Software

2	Generated by Hardware	Generated by Hardware	Generated by Hardware (But Software needs to fill ID to PID (UART_LINCTL[31:24]) first)
---	-----------------------	-----------------------	---

Table 6.23-11 LIN Header Selection in Master Mode

When UART is operated in LIN data transmission, LIN bus transfer state can be monitored by hardware or software. User can enable hardware monitoring by setting BITERREN (UART\_LINCTL [12]) to “1”, if the input pin (UART\_RX) state is not equal to the output pin (UART\_TX) state in LIN transmitter state that hardware will generate an interrupt to CPU. Software can also monitor the LIN bus transfer state by checking the read back data in UART\_DAT register. The following sequence is a program sequence example.

The procedure without software error monitoring in Master mode:

1. Fill Protected Identifier to PID (UART\_LINCTL[31:24]).
2. Select the hardware transmission header field including “break field + sync field + protected identifier field” by setting HSEL (UART\_LINCTL [23:22]) to “10”.
3. Set SENDH (UART\_LINCTL[8]) bit to 1 for requesting header transmission.
4. Wait until SENDH (UART\_LINCTL[8]) bit cleared by hardware.
5. Wait until TXEMPTYF (UART\_FIFOSTS[28]) set to 1 by hardware.

**Note1:** The default setting of break field is 12 dominant bits (break field) and 1 recessive bit break/sync delimiter. Setting BRKFL (UART\_LINCTL [19:16]) and BSL (UART\_LINCTL[21:20]) to change the LIN break field length and break/sync delimiter length.

**Note2:** The default setting of break/sync delimiter length is 1-bit time and the inter-byte spaces default setting is also 1-bit time. Setting BSL (UART\_LINCTL[21:20]) and DLY(UART\_TOUT[15:8]) can change break/sync delimiter length and inter-byte spaces.

**Note3:** If the header includes the “break field, sync field and frame ID field”, software must fill frame ID to PID (UART\_LINCTL[31:24]) before trigger header transmission (setting the SENDH (UART\_LINCTL[8])). The frame ID parity can be generated by software or hardware depending on IDPEN (UART\_LINCTL[9]) setting. If the parity generated by software with IDPEN (UART\_LINCTL[9]) is set to ‘0’, software must fill 8 bit data (include 2 bit parity) in this field. If the parity generated by hardware with IDPEN (UART\_LINCTL[9]) is set to ‘1’, software fill ID0~ID5 and hardware calculates P0 and P1.

The procedure with software error monitoring in Master mode:

1. Choose the hardware transmission header field to only include “break field” by setting HSEL (UART\_LINCTL [23:22]) to ‘00’.
2. Enable break detection function by setting BRKDETEN (UART\_LINCTL[10]).
3. Request break + break/sync delimiter transmission by setting the SENDH (UART\_LINCTL[8]).
4. Wait until the BRKDETF (UART\_LINSTS[8]) flag is set to “1” by hardware.
5. Request sync field transmission by writing 0x55 into UART\_DAT register.
6. Wait until the RDAIF (UART\_INTSTS[0]) is set to “1” by hardware and then read back the UART\_DAT register.
7. Request header frame ID transmission by writing the protected identifier value to UART\_DAT register.
8. Wait until the RDAIF (UART\_INTSTS[0]) is set to “1” by hardware and then read back the UART\_DAT register.

**LIN Break and Delimiter Detection**

When software enables the break detection function by setting BRKDETEN (UART\_LINCTL[10]), the break detection circuit is activated. The break detection circuit is totally independent from the UART receiver.

When the break detection function is enabled, the circuit looks at the input UART\_RX pin for a start signal. If UART LIN controller detects consecutive dominant is greater than 11 bits dominant followed by a recessive bit (delimiter), the BRKDETF (UART\_LINSTS[8]) flag is set at the end of break field. If the LINIEN (UART\_INTEN[8]) bit is set to 1, an interrupt LININT (UART\_INTSTS[15]) will be generated. The behavior of the break detection and break flag are shown in Figure 6.23-19.

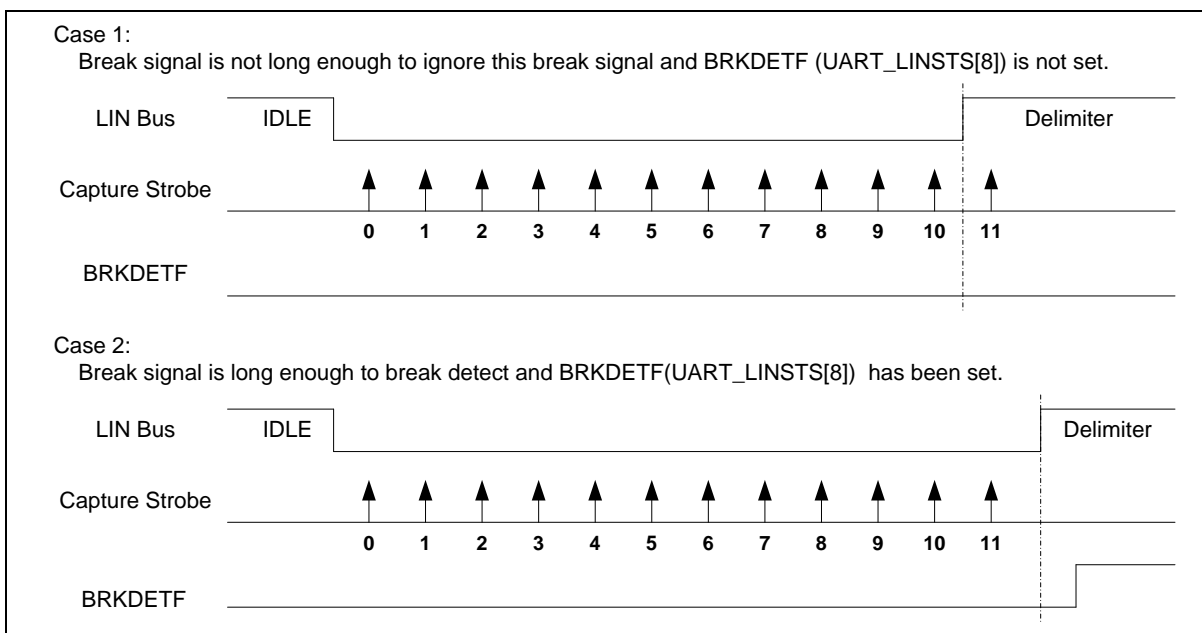


Figure 6.23-19 Break Detection in LIN Mode

### LIN Frame ID and Parity Format

The LIN frame ID value in LIN function mode is shown, the frame ID parity can be generated by software or hardware depends on IDPEN (UART\_LINCTL[9]).

If the parity generated by hardware (IDPEN (UART\_LINCTL[9])=1), user fill ID0~ID5 (UART\_LINCTL [29:24] ) hardware will calculate P0 (UART\_LINCTL[30]) and P1 (UART\_LINCTL[31]) otherwise user must filled frame ID and parity in this field.

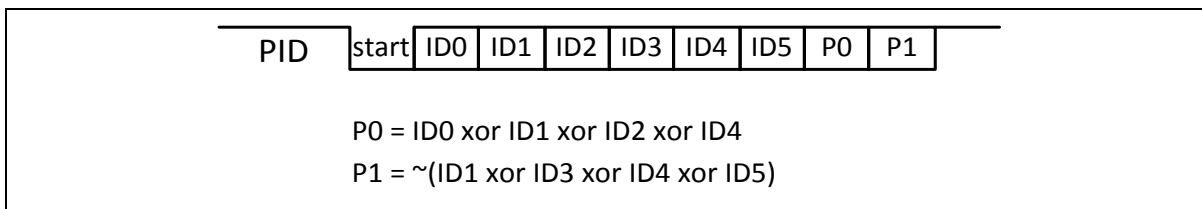


Figure 6.23-20 LIN Frame ID and Parity Format

### LIN Slave Mode

The UART Controller supports LIN Slave mode. To enable and initialize the LIN Slave mode, the following steps are necessary:

1. Set the UART\_BAUD register to select the desired baud rate.
2. Configure the data length to 8 bits by setting WLS (UART\_LINE[1:0]) to '11' and disable parity check by clearing PBE (UART\_LINE[3]) bit and configure with one stop bit by clearing NSB (UART\_LINE[2]) bit.



3. Select LIN function mode by setting FUNCSEL (UART\_FUNCSEL[1:0]) to '01'.
4. Enable LIN slave mode by setting the SLVEN (UART\_LINCTL[0]) to 1.

### LIN header reception

According to the LIN protocol, a slave node must wait for a valid header which comes from the master node. Next the slave task will take one of following actions (depending on the master header frame ID value).

- Receive the response.
- Transmit the response.
- Ignore the response and wait for next header.

In LIN Slave mode, user can enable the slave header detection function by setting the SLVHDEN (UART\_LINCTL[1]) to detect complete frame header (receive “break field”, “sync field” and “frame ID field”). When a LIN header is received, the SLVHDEF (UART\_LINSTS[0]) flag will be set. If the LINIEN (UART\_INTEN[8]) bit is set to 1, an interrupt will be generated. User can enable the frame ID parity check function by setting IDPEN (UART\_LINCTL[9]). If only received frame ID parity is not correct (break and sync field are correct), the SLVIDPEF (UART\_LINSTS[2]) flag is set to '1'. If the LINIEN (UART\_INTEN[8]) is set to 1, an interrupt will be generated and SLVHDEF (UART\_LINSTS[0]) is set to '1'. User can also put LIN in mute mode by setting MUTE (UART\_LINCTL[4]) to '1'. This mode allows detection of headers only (break + sync + frame ID) and prevents the reception of any other characters. In order to avoid bit rate tolerance, the controller supports automatic resynchronization function to avoid clock deviation error, user can enable this feature by setting SLVAREN (UART\_LINCTL[2]).

### LIN Response Transmission

The LIN slave node can transmit response and receive response. When slave node is the publisher of the response, the slave node sends response by filling data to the UART\_DAT register. If the slave node is the subscriber of the response, the slave node receives data from LIN bus.

### LIN Header Time-out Error

The LIN slave controller contains a header time-out counter. If the entire header is not received within the maximum time limit of 57 bit times, the header error flag SLVHEF (UART\_LINSTS [1]) will be set. The time-out counter is enabled at each break detect edge and stopped in the following conditions.

- A LIN frame ID field has been received.
- The header error flag asserts.
- Writing 1 to the SLVSYNCF (UART\_LINSTS[3]) to re-search a new frame header.

### Mute Mode and LIN Exit from Mute Mode Condition

In Mute mode, a LIN slave node will not receive any data until specified condition occurred. It allows header detection only and prevents the reception of any other characters. User can enable Mute mode by setting the MUTE (UART\_LINCTL[4]) and exiting from Mute mode condition can be selected by HSEL (UART\_LINCTL[23:22]).

**Note:** It is recommended to set LIN slave node to Mute mode after checksum transmission.

The LIN slave controller exiting from Mute mode is described as follows. If HSEL (UART\_LINCTL[23:22]) is set to “break field”, when LIN slave controller detects a valid LIN break + delimiter, the controller will enable the receiver (exit from Mute mode) and subsequent data (sync data, frame ID data, response data) are received in RX FIFO.

If HSEL (UART\_LINCTL[23:22]) is set to “break field and sync field”, when the LIN slave controller detects a valid LIN break + delimiter followed by a valid sync field without frame error, the controller will enable the receiver (exit from mute mode) and subsequent data (ID data, response data) are received in RX FIFO. If HSEL (UART\_LINCTL[23:22]) is set to “break field, sync field and ID field”, when the LIN slave controller detects a valid LIN break + delimiter and valid sync field without frame error followed by ID data without frame error and received ID data matched PID

(UART\_LINCTL[31:24]) value. The controller will enable the receiver (exit from mute mode) and subsequent data (response data) are received in RX FIFO.

**Slave Mode Non-automatic Resynchronization (NAR)**

User can disable the automatic resynchronization function to fix the communication baud rate. When operating in Non-Automatic Resynchronization mode, software needs some initial process, and the initialization process flow of Non-Automatic Resynchronization mode is shown as follows:

1. Select the desired baud rate by setting the UART\_BAUD register.
2. Select LIN function mode by setting FUNCSEL (UART\_FUNCSEL[1:0]) to '01'.
3. Disable automatic resynchronization function by setting SLVAREN (UART\_LINCTL[2]) is set to 0.
4. Enable LIN slave mode by setting the SLVEN (UART\_LINCTL[0]) is set to 1.

**Slave Mode with Automatic Resynchronization (AR)**

In Automatic Resynchronization (AR) mode, the controller will adjust the baud rate generator after each sync field reception. The initialization process flow of Automatic Resynchronization mode is shown as follows:

1. Select the desired baud rate by setting the UART\_BAUD register.
2. Select LIN function mode by setting UART\_FUNCSEL (UART\_FUNCSEL[1:0]) to '01'.
3. Enable automatic resynchronization function by setting SLVAREN (UART\_LINCTL[2]) to '1'.
4. Enable LIN slave mode by setting the SLVEN (UART\_LINCTL[0]) is set to '1'.

When the automatic resynchronization function is enabled, after each LIN break field, the time duration between five falling edges is sampled on peripheral clock and the result of this measurement is stored in an internal 13-bit register and the UART\_BAUD register value will be automatically updated at the end of the fifth falling edge. If the measure timer (13-bit) overflows before five falling edges, then the header error flag SLVHEF (UART\_LINSTS [1]) will be set.

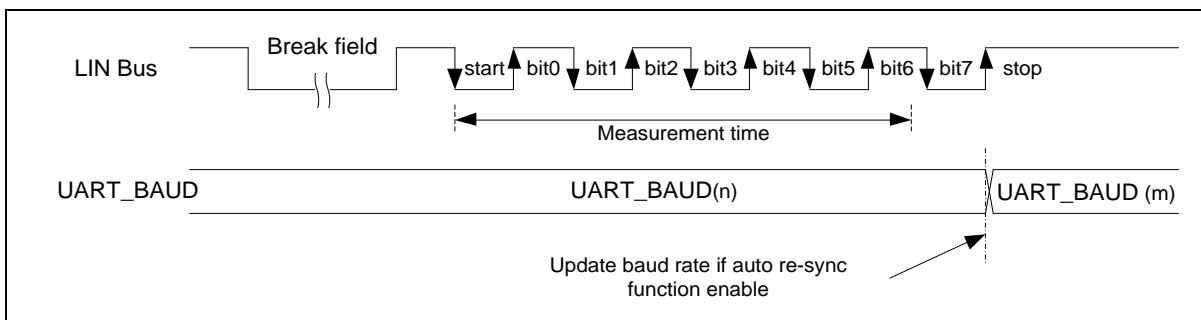


Figure 6.23-21 LIN Sync Field Measurement

When operating in Automatic Resynchronization (AR) mode, software must select the desired baud rate by setting the UART\_BAUD register and hardware will store it at internal TEMP\_REG register, after each LIN break field, the time duration between five falling edges is sampled on peripheral clock and the result of this measurement is stored in an internal 13-bit register BAUD\_LIN and the result will be updated to UART\_BAUD register automatically.

To guarantee the transmission baud rate, the baud rate generator must reload the initial value before each new break reception. The initial value is programmed by the application during initialization (TEMP\_REG). User can set SLVDUEN (UART\_LINCTL [3]) to enable auto reload initial baud rate value function. If the SLVDUEN (UART\_LINCTL [3]) is set, when received the next character, hardware will auto reload the initial value to UART\_BAUD, and when the UART\_BAUD be updated, the SLVDUEN (UART\_LINCTL [3]) will be cleared automatically. The behavior of LIN updated method as shown in Figure 6.23-22.

**Note1:** It is recommended to set the SLVDUEN bit before every checksum reception.

**Note2:** When a header error is detected, user must write 1 to SLVSYNCF (UART\_LINSTS[3]) to re-search new frame header. When writing 1 to it, hardware will reload the initial baud rate TEMP\_REG and re-search new frame header.

**Note3:** When operating in Automatic Resynchronization mode, the baud rate setting must be operated at mode2 (BAUDM1 (UART\_BAUD [29]) and BAUDM0 (UART\_BAUD[28]) must be 1).

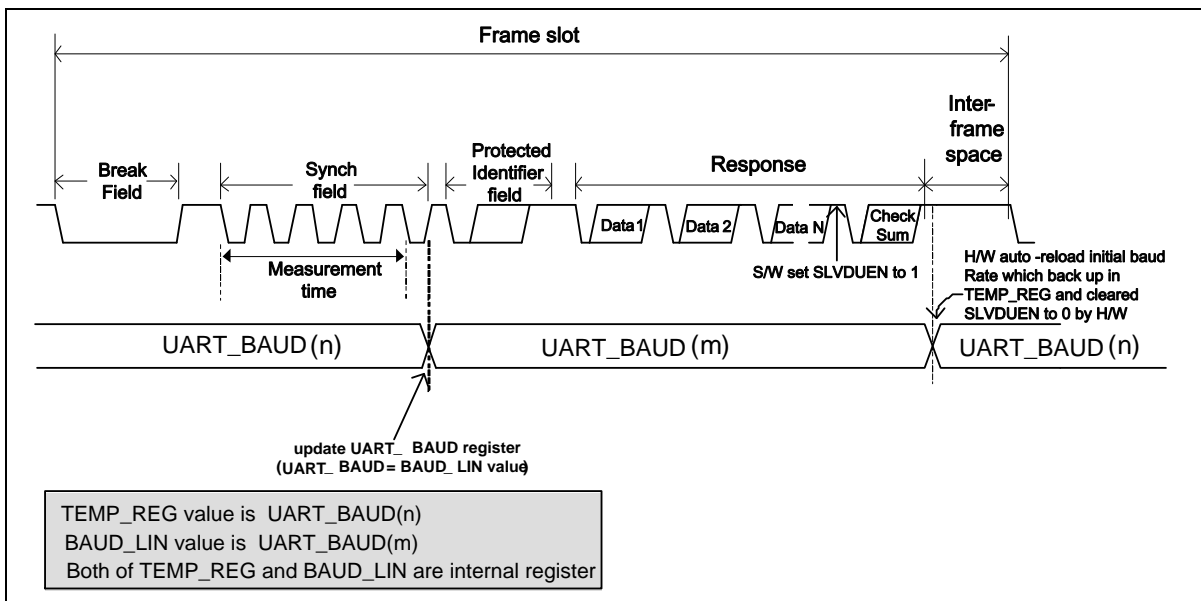


Figure 6.23-22 UART\_BAUD Update Sequence in AR Mode if SLVDUEN is 1

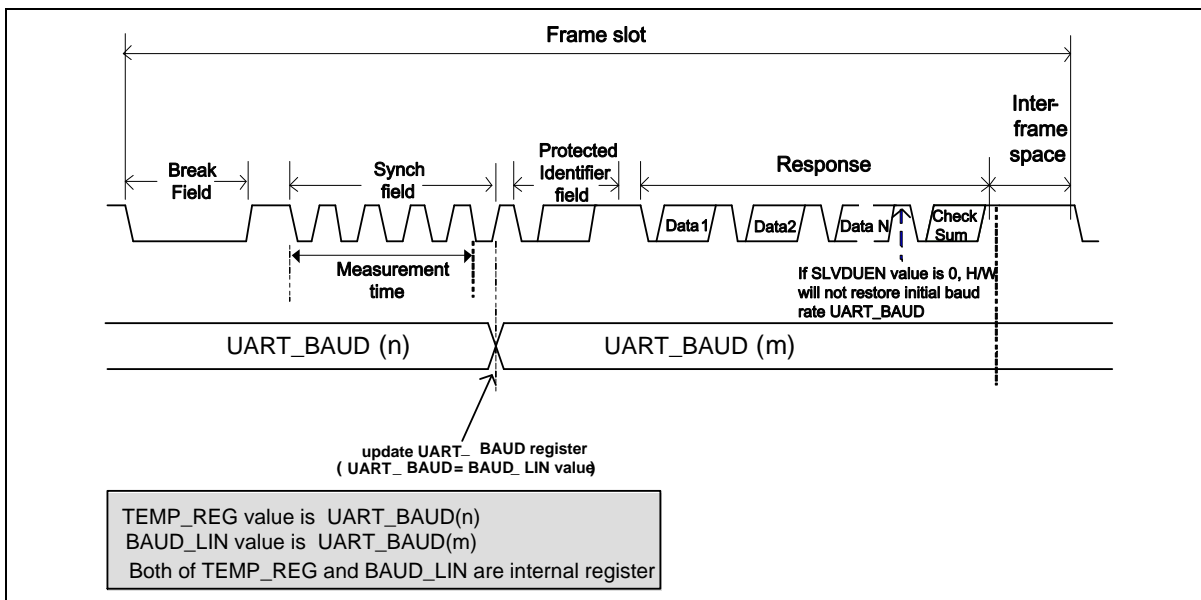


Figure 6.23-23 UART\_BAUD Update Sequence in AR mode if SLVDUEN is 0

**Deviation error on the sync field**

When operating in Automatic Resynchronization mode, the controller will check the deviation error on the sync field. The deviation error is checked by comparing the current baud rate with the received sync field. Two checks are performed in parallel.

Check1: Based on measurement between the first falling edge and the last falling edge of the sync field.

- If the difference is more than 14.84%, the header error flag SLVHEF (UART\_LINSTS[1]) will be set.
- If the difference is less than 14.06%, the header error flag SLVHEF (UART\_LINSTS[1]) will not be set.
- If the difference is between 14.84% and 14.06%, the header error flag SLVHEF (UART\_LINSTS[1]) may either set or not.

Check2: Based on measurement of time between each falling edge of the sync field.

- If the difference is more than 18.75%, the header error flag SLVHEF (UART\_LINSTS[1]) will be set.
- If the difference is less than 15.62%, the header error flag SLVHEF (UART\_LINSTS[1]) will not be set.
- If the difference is between 18.75% and 15.62%, the header error flag SLVHEF (UART\_LINSTS[1]) may either set or not.

**Note:** The deviation check is based on the current baud rate clock. Therefore, in order to guarantee correct deviation checking, the baud rate must reload the nominal value before each new break reception by setting SLVDUEN (UART\_LINCTL[3]) register (It is recommend setting the SLVDUEN (UART\_LINCTL[3]) bit before every checksum reception).

#### LIN Header Error Detection

In LIN Slave function mode, when user enables the header detection function by setting the SLVHDEN (UART\_LINCTL[1]), hardware will handle the header detect flow. If the header has an error, the LIN header error flag SLVHEF (UART\_LINSTS[1]) will be set and an interrupt is generated if the LINIEN (UART\_INTEN[8]) bit is set. When header error is detected, user must reset the detect circuit to re-search a new frame header by writing 1 to SLVSYNCF (UART\_LINSTS[3]) to re-search a new frame header.

The LIN header error flag SLVHEF (UART\_LINSTS[1]) is set if one of the following conditions occurs:

- Break Delimiter is too short (less than 0.5-bit time).
- Frame error in sync field or Identifier field.
- The sync field data is not 0x55 (Non-Automatic Resynchronization mode).
- The sync field deviation error (With Automatic Resynchronization mode).
- The sync field measure time-out (With Automatic Resynchronization mode).
- LIN header reception time-out.

#### 6.23.5.11 RS-485 Function Mode

Another alternate function of UART controller is RS-485 function (user must set UART\_FUNCSEL [1:0] to '11' 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 an RS-485 addressable slave and 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 UART\_LINE register to control the 9-th bit (When the PBE, EPE and SPE are set, the 9-th bit is transmitted 0 and when PBE and SPE are set and EPE is cleared, the 9-th 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)**

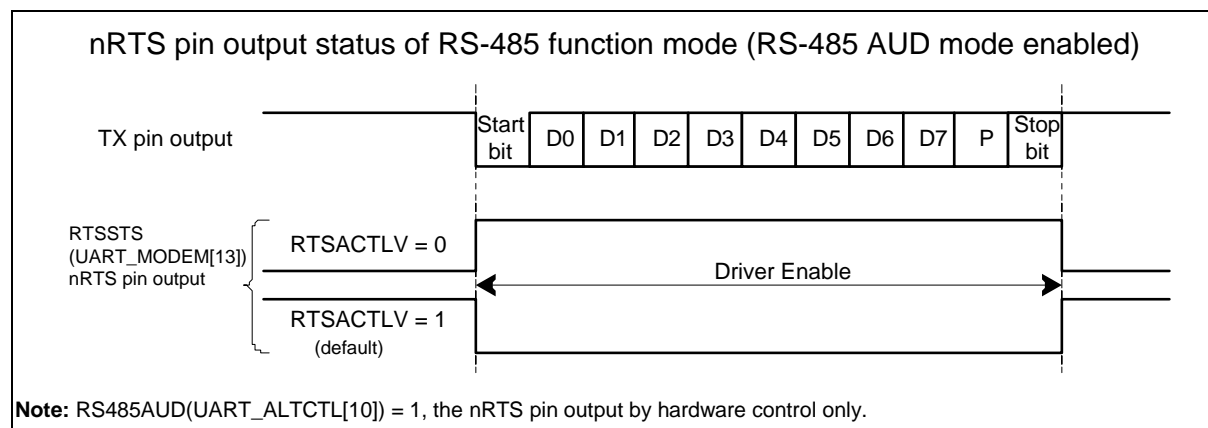
In RS-485 Auto Address Detection Operation Mode (RS485AAD (UART\_ALTCTL[9]) = 1), the receiver will ignore any data until an address byte is detected (bit 9 = 1) and the address byte data matches the ADDR MV (UART\_ALTCTL[31:24]) value. The address byte data will be stored in the RX FIFO. The all received 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 to disabled. User can set RTSACTLV in UART\_MODEM register to change the nRTS driving level.

Figure 6.23-24 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.



**Note:** RS485AUD(UART\_ALTCTL[10]) = 1, the nRTS pin output by hardware control only.

Figure 6.23-24 RS-485 nRTS Driving Level in Auto Direction Mode

Figure 6.23-25 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.

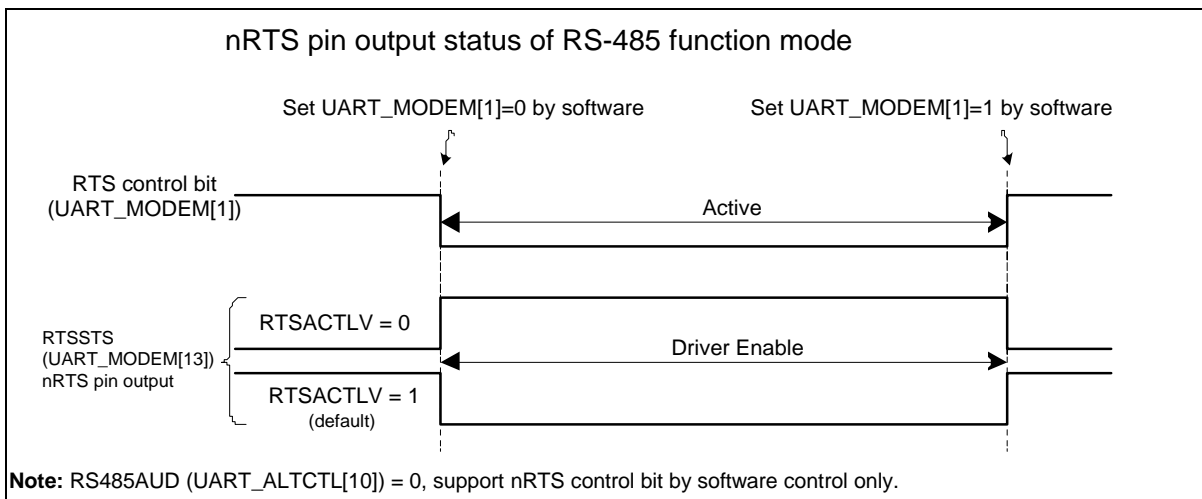


Figure 6.23-25 RS-485 nRTS Driving Level with Software Control

**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 ADDR MV (UART\_ALTCTL[31:24]) is programmed for auto address match value.
5. Determine auto direction control by programming RS485AUD (UART\_ALTCTL[10]).

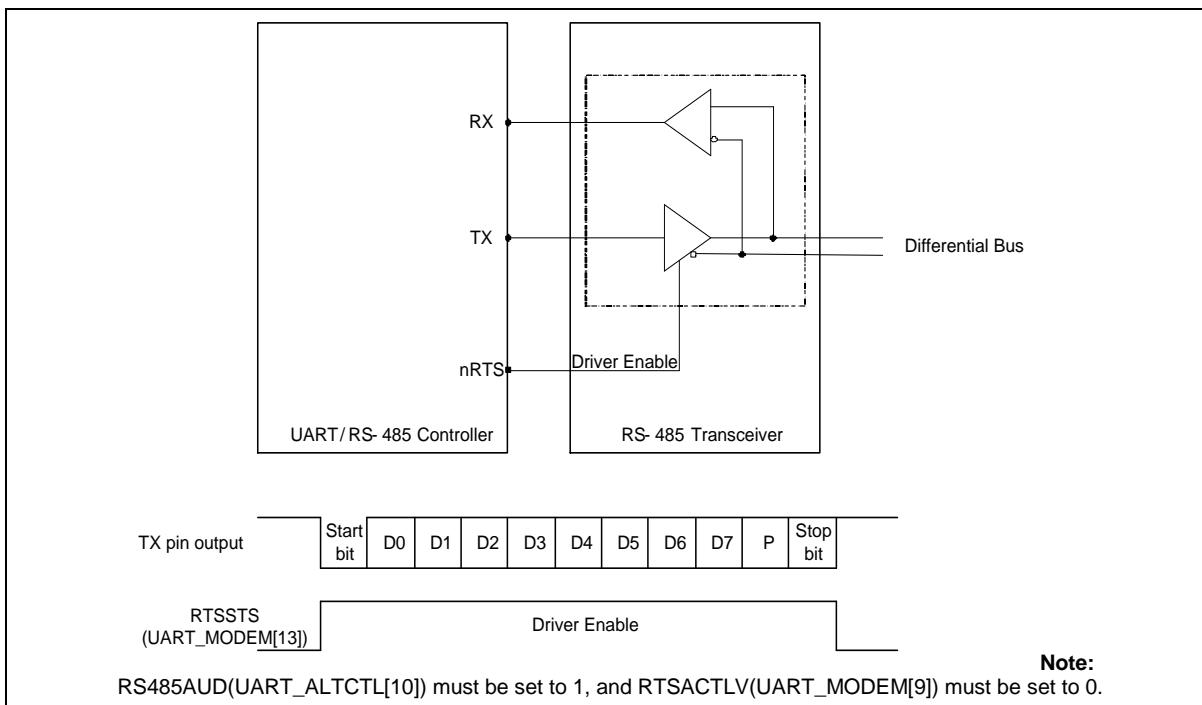


Figure 6.23-26 Structure of RS-485 Frame

#### 6.23.5.12 PDMA Transfer Function

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.

**Note:** If STOPn (PDMA\_STOP[n]) is set to stop UART RXPDMA task and the UART receive is not finish. UART controller will complete the transfer and stored current receive data in receive buffer. By reading RXEMPTY (UART\_FIFOSTS[14]) to check there is valid data in receive buffer or not.



### 6.23.6 Register Map

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

Register	Offset	R/W	Description	Reset Value
<b>UART Base Address:</b> <b>UARTx_BA = 0x4005_0000 + (0x10_0000 * x)</b> x=0, 1 <b>UART2_BA = 0x4015_4000</b>				
UART_DAT	UARTx_BA+0x00	R/W	UART Receive/Transmit Buffer Register	Undefined
UART_INTEN	UARTx_BA+0x04	R/W	UART Interrupt Enable Register	0x0000_0000
UART_FIFO	UARTx_BA+0x08	R/W	UART FIFO Control Register	0x0000_0101
UART_LINE	UARTx_BA+0x0C	R/W	UART Line Control Register	0x0000_0000
UART_MODEM	UARTx_BA+0x10	R/W	UART Modem Control Register	0x0000_0200
UART_MODEM STS	UARTx_BA+0x14	R/W	UART Modem Status Register	0x0000_0110
UART_FIFOSTS	UARTx_BA+0x18	R/W	UART FIFO Status Register	0xB040_4000
UART_INTSTS	UARTx_BA+0x1C	R/W	UART Interrupt Status Register	0x0040_0002
UART_TOUT	UARTx_BA+0x20	R/W	UART Time-out Register	0x0000_0000
UART_BAUD	UARTx_BA+0x24	R/W	UART Baud Rate Divider Register	0x0F00_0000
UART_IRDA	UARTx_BA+0x28	R/W	UART IrDA Control Register	0x0000_0040
UART_ALTCTL	UARTx_BA+0x2C	R/W	UART Alternate Control/Status Register	0x0000_000C
UART_FUNCSEL	UARTx_BA+0x30	R/W	UART Function Select Register	0x0000_0000
UART_LINCTL	UARTx_BA+0x34	R/W	UART LIN Control Register (Only for UART0 and UART1)	0x000C_0000
UART_LINSTS	UARTx_BA+0x38	R/W	UART LIN Status Register (Only for UART0 and UART1)	0x0000_0000



<b>UART_BRCOMP</b>	UARTx_BA+0x3C	R/W	UART Baud Rate Compensation Register	0x0000_0000
<b>UART_WKCTL</b>	UARTx_BA+0x40	R/W	UART Wake-up Control Register	0x0000_0000
<b>UART_WKSTS</b>	UARTx_BA+0x44	R/W	UART Wake-up Status Register	0x0000_0000
<b>UART_DWKCOMP</b>	UARTx_BA+0x48	R/W	UART Incoming Data Wake-up Compensation Register	0x0000_0000

6.23.7 Register Description

UART Receive/Transmit Buffer Register (UART\_DAT)

Register	Offset	R/W	Description	Reset Value
UART_DAT	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	<p><b>Parity Bit Receive/Transmit Buffer</b></p> <p>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.</p> <p>Read Operation: If PBE (UART_LINE[3]) and PSS (UART_LINE[7]) are enabled, the parity bit can be read by this bit.</p> <p><b>Note:</b> This bit has effect only when PBE (UART_LINE[3]) and PSS (UART_LINE[7]) are set.</p>
[7:0]	DAT	<p><b>Data Receive/Transmit Buffer</b></p> <p>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.</p> <p>Read Operation: By reading this register, the UART controller will return an 8-bit data received from receiver FIFO.</p>

**UART Interrupt Enable Register (UART\_INTEN)**

Register	Offset	R/W	Description	Reset Value
UART_INTEN	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	
15	14	13	12	11	10	9	8
RXPDMAEN	TXPDMAEN	ATOCTSEN	ATORTSEN	TOCNTEN	Reserved		LINIEN
7	6	5	4	3	2	1	0
Reserved	WKIEN	BUFERRIEN	RXTOIEN	MODEMIEN	RLSIEN	THREIEN	RDAIEN

Bits	Description	
[31:23]	Reserved	Reserved.
[22]	TXENDIEN	<p><b>Transmitter Empty Interrupt Enable Bit</b></p> <p>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).</p> <p>0 = Transmitter empty interrupt Disabled.</p> <p>1 = Transmitter empty interrupt Enabled.</p>
[21:19]	Reserved	Reserved.
[18]	ABRIEN	<p><b>Auto-baud Rate Interrupt Enable Bit</b></p> <p>0 = Auto-baud rate interrupt Disabled.</p> <p>1 = Auto-baud rate interrupt Enabled.</p>
[17:16]	Reserved	Reserved.
[15]	RXPDMAEN	<p><b>RX PDMA Enable Bit</b></p> <p>This bit can enable or disable RX PDMA service.</p> <p>0 = RX PDMA Disabled.</p> <p>1 = RX PDMA Enabled.</p> <p><b>Note:</b> 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 stop. 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.</p>
[14]	TXPDMAEN	<p><b>TX PDMA Enable Bit</b></p> <p>This bit can enable or disable TX PDMA service.</p> <p>0 = TX PDMA Disabled.</p> <p>1 = TX PDMA Enabled.</p>
[13]	ATOCTSEN	<b>nCTS Auto-flow Control Enable Bit</b>

		<p>0 = nCTS auto-flow control Disabled.                      1 = nCTS auto-flow control Enabled.</p> <p><b>Note:</b> 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).</p>
[12]	<b>ATORTSEN</b>	<p><b>nRTS Auto-flow Control Enable Bit</b>                      0 = nRTS auto-flow control Disabled.                      1 = nRTS auto-flow control Enabled.</p> <p><b>Note:</b> 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.</p>
[11]	<b>TOCNTEN</b>	<p><b>Receive Buffer Time-out Counter Enable Bit</b>                      0 = Receive Buffer Time-out counter Disabled.                      1 = Receive Buffer Time-out counter Enabled.</p>
[10:9]	<b>Reserved</b>	Reserved.
[8]	<b>LINIEN</b>	<p><b>LIN Bus Interrupt Enable Bit</b>                      0 = LIN bus interrupt Disabled.                      1 = LIN bus interrupt Enabled.</p> <p><b>Note:</b> This bit is used for LIN function mode.</p>
[7]	<b>Reserved</b>	Reserved.
[6]	<b>WKIEN</b>	<p><b>Wake-up Interrupt Enable Bit</b>                      0 = Wake-up Interrupt Disabled.                      1 = Wake-up Interrupt Enabled.</p>
[5]	<b>BUFERRIEN</b>	<p><b>Buffer Error Interrupt Enable Bit</b>                      0 = Buffer error interrupt Disabled.                      1 = Buffer error interrupt Enabled.</p>
[4]	<b>RXTOIEN</b>	<p><b>RX Time-out Interrupt Enable Bit</b>                      0 = RX time-out interrupt Disabled.                      1 = RX time-out interrupt Enabled.</p>
[3]	<b>MODEMIEN</b>	<p><b>Modem Status Interrupt Enable Bit</b>                      0 = Modem status interrupt Disabled.                      1 = Modem status interrupt Enabled.</p>
[2]	<b>RLSIEN</b>	<p><b>Receive Line Status Interrupt Enable Bit</b>                      0 = Receive Line Status interrupt Disabled.                      1 = Receive Line Status interrupt Enabled.</p>
[1]	<b>THREIEN</b>	<p><b>Transmit Holding Register Empty Interrupt Enable Bit</b>                      0 = Transmit holding register empty interrupt Disabled.                      1 = Transmit holding register empty interrupt Enabled.</p>
[0]	<b>RDAIEN</b>	<p><b>Receive Data Available Interrupt Enable Bit</b>                      0 = Receive data available interrupt Disabled.                      1 = Receive data available interrupt Enabled.</p>

**UART FIFO Control Register (UART\_FIFO)**

Register	Offset	R/W	Description	Reset Value
UART_FIFO	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	<p><b>nRTS Trigger Level for Auto-flow Control Use</b></p> <p>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.</p> <p><b>Note:</b> This field is used for auto nRTS flow control.</p>
[15:9]	Reserved	Reserved.
[8]	RXOFF	<p><b>Receiver Disable Bit</b></p> <p>The receiver is disabled or not (set 1 to disable receiver).                      0 = Receiver Enabled.                      1 = Receiver Disabled.</p> <p><b>Note:</b> This bit is used for RS-485 Normal Multi-drop mode. It should be programmed before RS485NMM (UART_ALTCTL [8]) is programmed.</p>
[7:4]	RFITL	<p><b>RX FIFO Interrupt Trigger Level</b></p> <p>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).</p> <p>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.</p>
[3]	Reserved	Reserved.
[2]	TXRST	<p><b>TX Field Software Reset</b></p> <p>When TXRST (UART_FIFO[2]) is set, all the byte in the transmit FIFO and TX internal</p>

		<p>state machine are cleared.</p> <p>0 = No effect.</p> <p>1 = Reset the TX internal state machine and pointers.</p> <p><b>Note1:</b> This bit will automatically clear at least 3 UART peripheral clock cycles.</p> <p><b>Note2:</b> Before setting this bit, it should wait for the TXEMPTYF (UART_FIFOSTS[28]) be set.</p>
[1]	<b>RXRST</b>	<p><b>RX Field Software Reset</b></p> <p>When RXRST (UART_FIFO[1]) is set, all the byte in the receiver FIFO and RX internal state machine are cleared.</p> <p>0 = No effect.</p> <p>1 = Reset the RX internal state machine and pointers.</p> <p><b>Note1:</b> This bit will automatically clear at least 3 UART peripheral clock cycles.</p> <p><b>Note2:</b> Before setting this bit, it should wait for the RXIDLE (UART_FIFOSTS[29]) be set.</p>
[0]	<b>Reserved</b>	Reserved.

**UART Line Control Register (UART\_LINE)**

Register	Offset	R/W	Description	Reset Value
UART_LINE	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	<p><b>RX Data Inverted</b> 0 = Received data signal inverted Disabled. 1 = Received data signal inverted Enabled.</p> <p><b>Note1:</b> 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.</p> <p><b>Note2:</b> This bit is valid when FUNCSEL (UART_FUNCSEL[1:0]) is select UART, LIN or RS485 function.</p>
[8]	TXDINV	<p><b>TX Data Inverted</b> 0 = Transmitted data signal inverted Disabled. 1 = Transmitted data signal inverted Enabled.</p> <p><b>Note1:</b> 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.</p> <p><b>Note2:</b> This bit is valid when FUNCSEL (UART_FUNCSEL[1:0]) is select UART, LIN or RS485 function.</p>
[7]	PSS	<p><b>Parity Bit Source Selection</b> 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.</p> <p><b>Note1:</b> This bit has effect only when PBE (UART_LINE[3]) is set.</p> <p><b>Note2:</b> 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]).</p>
[6]	BCB	<p><b>Break Control Bit</b> 0 = Break Control Disabled. 1 = Break Control Enabled.</p> <p><b>Note:</b> When this bit is set to logic 1, the transmitted serial data output (TX) is forced to the</p>

		Spacing State (logic 0). This bit acts only on TX line and has no effect on the transmitter logic.
[5]	<b>SPE</b>	<p><b>Stick Parity Enable Bit</b></p> <p>0 = Stick parity Disabled. 1 = Stick parity Enabled.</p> <p><b>Note:</b> 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.</p>
[4]	<b>EPE</b>	<p><b>Even Parity Enable Bit</b></p> <p>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.</p> <p><b>Note:</b> This bit has effect only when PBE (UART_LINE[3]) is set.</p>
[3]	<b>PBE</b>	<p><b>Parity Bit Enable Bit</b></p> <p>0 = Parity bit generated Disabled. 1 = Parity bit generated Enabled.</p> <p><b>Note:</b> Parity bit is generated on each outgoing character and is checked on each incoming data.</p>
[2]	<b>NSB</b>	<p><b>Number of "STOP Bit"</b></p> <p>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.</p>
[1:0]	<b>WLS</b>	<p><b>Word Length Selection</b></p> <p>This field sets UART word length.</p> <p>00 = 5 bits. 01 = 6 bits. 10 = 7 bits. 11 = 8 bits.</p>



**UART Modem Control Register (UART\_MODEM)**

Register	Offset	R/W	Description	Reset Value
UART_MODEM	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	<p><b>nRTS Pin Status (Read Only)</b>                      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.</p>
[12:10]	Reserved	Reserved.
[9]	RTSACTLV	<p><b>nRTS Pin Active Level</b>                      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)</p> <p><b>Note1:</b> Refer to Figure 6.23-13 and Figure 6.23-14 for UART function mode.  <b>Note2:</b> Refer to Figure 6.23-24 and Figure 6.23-25 for RS-485 function mode.  <b>Note3:</b> 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.</p>
[8:2]	Reserved	Reserved.
[1]	RTS	<p><b>nRTS (Request-to-send) Signal Control</b>                      This bit is direct control internal nRTS 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.</p> <p><b>Note1:</b> This nRTS signal control bit is not effective when nRTS auto-flow control is enabled in UART function mode.  <b>Note2:</b> This nRTS signal control bit is not effective when RS-485 auto direction mode (AUD) is enabled in RS-485 function mode.</p>
[0]	Reserved	Reserved.

**UART Modem Status Register (UART\_MODEMSTS)**

Register	Offset	R/W	Description	Reset Value
UART_MODEMSTS	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	<p><b>nCTS Pin Active Level</b>                      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)</p> <p><b>Note:</b> 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.</p>
[7:5]	Reserved	Reserved.
[4]	CTSSTS	<p><b>nCTS Pin Status (Read Only)</b>                      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.</p> <p><b>Note:</b> This bit echoes when UART controller peripheral clock is enabled, and nCTS multi-function port is selected.</p>
[3:1]	Reserved	Reserved.
[0]	CTSDETF	<p><b>Detect nCTS State Change Flag</b>                      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.</p> <p><b>Note:</b> This bit can be cleared by writing "1" to it.</p>

**UART FIFO Status Register (UART\_FIFOSTS)**

Register	Offset	R/W	Description	Reset Value
UART_FIFOSTS	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	ADDRDEF	ABRDTIOF	ABRDIF	RXOVIF

Bits	Description	
[31]	TXRXACT	<p><b>TX and RX Active Status (Read Only)</b>                      This bit indicates TX and RX are active or inactive.                      0 = TX and RX are inactive.                      1 = TX and RX are active. (Default)</p> <p><b>Note:</b> When TXRXDIS (UART_FUNCSEL[3]) is set and both TX and RX are in idle state, this bit is cleared. The UART controller can not transmit or receive data at this moment. Otherwise this bit is set.</p>
[30]	Reserved	Reserved.
[29]	RXIDLE	<p><b>RX Idle Status (Read Only)</b>                      This bit is set by hardware when RX is idle.                      0 = RX is busy.                      1 = RX is idle. (Default)</p>
[28]	TXEMPTYF	<p><b>Transmitter Empty Flag (Read Only)</b>                      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.</p> <p><b>Note:</b> This bit is cleared automatically when TX FIFO is not empty or the last byte transmission has not completed.</p>
[27:25]	Reserved	Reserved.
[24]	TXOVIF	<p><b>TX Overflow Error Interrupt Flag</b>                      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.</p> <p><b>Note:</b> This bit can be cleared by writing "1" to it.</p>
[23]	TXFULL	<p><b>Transmitter FIFO Full (Read Only)</b>                      This bit indicates TX FIFO full or not.</p>

		<p>0 = TX FIFO is not full. 1 = TX FIFO is full.</p> <p><b>Note:</b> This bit is set when the number of usage in TX FIFO Buffer is equal to 16, otherwise it is cleared by hardware.</p>
[22]	<b>TXEMPTY</b>	<p><b>Transmitter FIFO Empty (Read Only)</b> This bit indicates TX FIFO empty or not.</p> <p>0 = TX FIFO is not empty. 1 = TX FIFO is empty.</p> <p><b>Note:</b> 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).</p>
[21:16]	<b>TXPTR</b>	<p><b>TX FIFO Pointer (Read Only)</b> 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.</p> <p>The Maximum value shown in TXPTR is 15. When the using level of TX FIFO Buffer 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.</p>
[15]	<b>RXFULL</b>	<p><b>Receiver FIFO Full (Read Only)</b> This bit initiates RX FIFO full or not.</p> <p>0 = RX FIFO is not full. 1 = RX FIFO is full.</p> <p><b>Note:</b> This bit is set when the number of usage in RX FIFO Buffer is equal to 16, otherwise it is cleared by hardware.</p>
[14]	<b>RXEMPTY</b>	<p><b>Receiver FIFO Empty (Read Only)</b> This bit initiate RX FIFO empty or not.</p> <p>0 = RX FIFO is not empty. 1 = RX FIFO is empty.</p> <p><b>Note:</b> 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.</p>
[13:8]	<b>RXPTR</b>	<p><b>RX FIFO Pointer (Read Only)</b> 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.</p> <p>The Maximum value shown in RXPTR is 15. When the using level of RX FIFO Buffer 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.</p>
[7]	<b>Reserved</b>	Reserved.
[6]	<b>BIF</b>	<p><b>Break Interrupt Flag</b> 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).</p> <p>0 = No Break interrupt is generated. 1 = Break interrupt is generated.</p> <p><b>Note:</b> This bit can be cleared by writing “1” to it.</p>
[5]	<b>FEF</b>	<p><b>Framing Error Flag</b> 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).</p> <p>0 = No framing error is generated. 1 = Framing error is generated.</p>

		<b>Note:</b> This bit can be cleared by writing “1” to it.
[4]	<b>PEF</b>	<p><b>Parity Error Flag</b></p> <p>This bit is set to logic 1 whenever the received character does not have a valid “parity bit”.</p> <p>0 = No parity error is generated. 1 = Parity error is generated.</p> <p><b>Note:</b> This bit can be cleared by writing “1” to it.</p>
[3]	<b>ADDRDET</b>	<p><b>RS-485 Address Byte Detect Flag</b></p> <p>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').</p> <p><b>Note1:</b> This field is used for RS-485 function mode and ADDRDEN (UART_ALTCTL[15]) is set to 1 to enable Address detection mode.</p> <p><b>Note2:</b> This bit can be cleared by writing “1” to it.</p>
[2]	<b>ABRDTOIF</b>	<p><b>Auto-baud Rate Detect Time-out Interrupt Flag</b></p> <p>This bit is set to logic “1” in Auto-baud Rate Detect mode when the baud rate counter is overflow.</p> <p>0 = Auto-baud rate counter is underflow. 1 = Auto-baud rate counter is overflow.</p> <p><b>Note:</b> This bit can be cleared by writing “1” to it.</p>
[1]	<b>ABRDIF</b>	<p><b>Auto-baud Rate Detect Interrupt Flag</b></p> <p>This bit is set to logic “1” when auto-baud rate detect function is finished.</p> <p>0 = Auto-baud rate detect function is not finished. 1 = Auto-baud rate detect function is finished.</p> <p><b>Note:</b> This bit can be cleared by writing “1” to it.</p>
[0]	<b>RXOVIF</b>	<p><b>RX Overflow Error Interrupt Flag</b></p> <p>This bit is set when RX FIFO overflow.</p> <p>If the number of bytes of received data is greater than RX_FIFO (UART_DAT) size 16 bytes, this bit will be set.</p> <p>0 = RX FIFO is not overflow. 1 = RX FIFO is overflow.</p> <p><b>Note:</b> This bit can be cleared by writing “1” to it.</p>

**UART Interrupt Status Register (UART\_INTSTS)**

Register	Offset	R/W	Description	Reset Value
UART_INTSTS	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	
23	22	21	20	19	18	17	16
Reserved	TXENDIF	HWBUFEIF	HWTOIF	HWMODIF	HWRLSIF	Reserved	
15	14	13	12	11	10	9	8
LININT	WKINT	BUFERRINT	RXTOINT	MODEMINT	RLSINT	THREINT	RDAINT
7	6	5	4	3	2	1	0
LINIF	WKIF	BUFERRIF	RXTOIF	MODEMIF	RLSIF	THREIF	RDAIF

Bits	Description	
[31]	ABRINT	<p><b>Auto-baud Rate Interrupt Indicator (Read Only)</b></p> <p>This bit is set if ABRIEN (UART_INTEN[18]) and ABRIF (UART_ALTCTL[17]) are both set to 1.</p> <p>0 = No Auto-baud Rate interrupt is generated.</p> <p>1 = The Auto-baud Rate interrupt is generated.</p>
[30]	TXENDINT	<p><b>Transmitter Empty Interrupt Indicator (Read Only)</b></p> <p>This bit is set if TXENDIEN (UART_INTEN[22]) and TXENDIF(UART_INTSTS[22]) are both set to 1.</p> <p>0 = No Transmitter Empty interrupt is generated.</p> <p>1 = Transmitter Empty interrupt is generated.</p>
[29]	HWBUFEINT	<p><b>PDMA Mode Buffer Error Interrupt Indicator (Read Only)</b></p> <p>This bit is set if BUFERRIEN (UART_INTEN[5]) and HWBUFEIF (UART_INTSTS[21]) are both set to 1.</p> <p>0 = No buffer error interrupt is generated in PDMA mode.</p> <p>1 = Buffer error interrupt is generated in PDMA mode.</p>
[28]	HWTOINT	<p><b>PDMA Mode RX Time-out Interrupt Indicator (Read Only)</b></p> <p>This bit is set if RXTOIEN (UART_INTEN[4]) and HWTOIF(UART_INTSTS[20]) are both set to 1.</p> <p>0 = No RX time-out interrupt is generated in PDMA mode.</p> <p>1 = RX time-out interrupt is generated in PDMA mode.</p>
[27]	HWMODINT	<p><b>PDMA Mode MODEM Status Interrupt Indicator (Read Only)</b></p> <p>This bit is set if MODEMIEN (UART_INTEN[3]) and HWMODIF(UART_INTSTS[19]) are both set to 1.</p> <p>0 = No Modem interrupt is generated in PDMA mode.</p> <p>1 = Modem interrupt is generated in PDMA mode.</p>
[26]	HWRLSINT	<p><b>PDMA Mode Receive Line Status Interrupt Indicator (Read Only)</b></p> <p>This bit is set if RLSIEN (UART_INTEN[2]) and HWRLSIF(UART_INTSTS[18]) are both set to 1.</p>

		0 = No RLS interrupt is generated in PDMA mode. 1 = RLS interrupt is generated in PDMA mode.
[25:23]	<b>Reserved</b>	Reserved.
[22]	<b>TXENDIF</b>	<b>Transmitter Empty Interrupt Flag (Read Only)</b> 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. <b>Note:</b> This bit is cleared automatically when TX FIFO is not empty or the last byte transmission has not completed.
[21]	<b>HWBUFEIF</b>	<b>PDMA Mode Buffer Error Interrupt Flag (Read Only)</b> 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. <b>Note:</b> This bit is cleared when both TXOVIF (UART_FIFOSTS[24]) and RXOVIF (UART_FIFOSTS[0]) are cleared.
[20]	<b>HWTOIF</b>	<b>PDMA Mode RX Time-out Interrupt Flag (Read Only)</b> 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. <b>Note:</b> This bit is read only and user can read UART_DAT (RX is in active) to clear it.
[19]	<b>HWMODIF</b>	<b>PDMA Mode MODEM Interrupt Flag (Read Only)</b> 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. <b>Note:</b> 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]	<b>HWRLSIF</b>	<b>PDMA Mode Receive Line Status Flag (Read Only)</b> 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. <b>Note1:</b> In RS-485 function mode, this field include "receiver detect any address byte received address byte character (bit9 = '1') bit". <b>Note2:</b> 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. <b>Note3:</b> 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.
[17:16]	<b>Reserved</b>	Reserved.
[15]	<b>LININT</b>	<b>LIN Bus Interrupt Indicator (Read Only)</b>

		<p>This bit is set if LINIEN (UART_INTEN[8]) and LINIF(UART_INTSTS[7]) are both set to 1.</p> <p>0 = No LIN Bus interrupt is generated.</p> <p>1 = The LIN Bus interrupt is generated.</p>
[14]	WKINT	<p><b>UART Wake-up Interrupt Indicator (Read Only)</b></p> <p>This bit is set if WKIEN (UART_INTEN[6]) and WKIF (UART_INTSTS[6]) are both set to 1.</p> <p>0 = No UART wake-up interrupt is generated.</p> <p>1 = UART wake-up interrupt is generated.</p>
[13]	BUFERRINT	<p><b>Buffer Error Interrupt Indicator (Read Only)</b></p> <p>This bit is set if BUFERRIEN(UART_INTEN[5]) and BUFERRIF(UART_INTSTS[5]) are both set to 1.</p> <p>0 = No buffer error interrupt is generated.</p> <p>1 = Buffer error interrupt is generated.</p>
[12]	RXTOINT	<p><b>RX Time-out Interrupt Indicator (Read Only)</b></p> <p>This bit is set if RXTOIEN (UART_INTEN[4]) and RXTOIF(UART_INTSTS[4]) are both set to 1.</p> <p>0 = No RX time-out interrupt is generated.</p> <p>1 = RX time-out interrupt is generated.</p>
[11]	MODEMINT	<p><b>MODEM Status Interrupt Indicator (Read Only)</b></p> <p>This bit is set if MODEMIEN(UART_INTEN[3]) and MODEMIF(UART_INTSTS[3]) are both set to 1</p> <p>0 = No Modem interrupt is generated.</p> <p>1 = Modem interrupt is generated..</p>
[10]	RLSINT	<p><b>Receive Line Status Interrupt Indicator (Read Only)</b></p> <p>This bit is set if RLSIEN (UART_INTEN[2]) and RLSIF(UART_INTSTS[2]) are both set to 1.</p> <p>0 = No RLS interrupt is generated.</p> <p>1 = RLS interrupt is generated.</p>
[9]	THREINT	<p><b>Transmit Holding Register Empty Interrupt Indicator (Read Only)</b></p> <p>This bit is set if THREIEN (UART_INTEN[1]) and THREIF(UART_INTSTS[1]) are both set to 1.</p> <p>0 = No THRE interrupt is generated.</p> <p>1 = THRE interrupt is generated.</p>
[8]	RDAINT	<p><b>Receive Data Available Interrupt Indicator (Read Only)</b></p> <p>This bit is set if RDAIEN (UART_INTEN[0]) and RDAIF (UART_INTSTS[0]) are both set to 1.</p> <p>0 = No RDA interrupt is generated.</p> <p>1 = RDA interrupt is generated.</p>
[7]	LINIF	<p><b>LIN Bus Interrupt Flag</b></p> <p>This bit is set when LIN slave header detect (SLVHDET(UART_LINSTS[0]=1)), LIN break detect (BRKDET(UART_LINSTS[8]=1)), bit error detect (BITEF(UART_LINSTS[9]=1)), LIN slave ID parity error (SLVIDPEF(UART_LINSTS[2]=1)) or LIN slave header error detect (SLVHEF (UART_LINSTS[1])). If LINIEN (UART_INTEN [8]) is enabled the LIN interrupt will be generated.</p> <p>0 = None of SLVHDET, BRKDET, BITEF, SLVIDPEF and SLVHEF is generated.</p> <p>1 = At least one of SLVHDET, BRKDET, BITEF, SLVIDPEF and SLVHEF is generated.</p> <p><b>Note:</b> This bit is cleared when SLVHDET(UART_LINSTS[0]), BRKDET(UART_LINSTS[8]), BITEF(UART_LINSTS[9]), SLVIDPEF (UART_LINSTS[2]) and SLVHEF(UART_LINSTS[1]) all are cleared and software writing '1' to</p>



		LINIF(UART_INTSTS[7]).
[6]	WKIF	<p><b>UART Wake-up Interrupt Flag (Read Only)</b></p> <p>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.</p> <p>0 = No UART wake-up interrupt flag is generated. 1 = UART wake-up interrupt flag is generated.</p> <p><b>Note:</b> 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.</p>
[5]	BUFERRIF	<p><b>Buffer Error Interrupt Flag (Read Only)</b></p> <p>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.</p> <p>0 = No buffer error interrupt flag is generated. 1 = Buffer error interrupt flag is generated.</p> <p><b>Note:</b> 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]).</p>
[4]	RXTOIF	<p><b>RX Time-out Interrupt Flag (Read Only)</b></p> <p>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.</p> <p>0 = No RX time-out interrupt flag is generated. 1 = RX time-out interrupt flag is generated.</p> <p><b>Note:</b> This bit is read only and user can read UART_DAT (RX is in active) to clear it.</p>
[3]	MODEMIF	<p><b>MODEM Interrupt Flag (Read Only)</b></p> <p>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.</p> <p>0 = No Modem interrupt flag is generated. 1 = Modem interrupt flag is generated.</p> <p><b>Note:</b> This bit is read only and reset to 0 when bit CTSDETF is cleared by a write 1 on CTSDETF(UART_MODEMSTS[0]).</p>
[2]	RLSIF	<p><b>Receive Line Interrupt Flag (Read Only)</b></p> <p>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.</p> <p>0 = No RLS interrupt flag is generated. 1 = RLS interrupt flag is generated.</p> <p><b>Note1:</b> 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.</p> <p><b>Note2:</b> 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.</p> <p><b>Note3:</b> 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.</p>
[1]	THREIF	<p><b>Transmit Holding Register Empty Interrupt Flag (Read Only)</b></p> <p>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.</p> <p>0 = No THRE interrupt flag is generated.</p>

		<p>1 = THRE interrupt flag is generated.</p> <p><b>Note:</b> This bit is read only and it will be cleared when writing data into UART_DAT (TX FIFO not empty).</p>
[0]	RDAIF	<p><b>Receive Data Available Interrupt Flag (Read Only)</b></p> <p>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.</p> <p>0 = No RDA interrupt flag is generated.</p> <p>1 = RDA interrupt flag is generated.</p> <p><b>Note:</b> 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])).</p>

**UART Time-out Register (UART\_TOUT)**

Register	Offset	R/W	Description	Reset Value
UART_TOUT	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	<b>TX Delay Time Value</b> This field is used to programming the transfer delay time between the last stop bit and next start bit. The unit is bit time.
[7:0]	TOIC	<b>Time-out Interrupt Comparator</b> 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	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	<p><b>BAUD Rate Mode Selection Bit 1</b></p> <p>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 6.23-3.</p> <p><b>Note:</b> In IrDA mode must be operated in mode 0.</p>
[28]	BAUDM0	<p><b>BAUD Rate Mode Selection Bit 0</b></p> <p>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 6.23-3.</p>
[27:24]	EDIVM1	<p><b>Extra Divider for BAUD Rate Mode 1</b></p> <p>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 6.23-3.</p>
[23:16]	Reserved	Reserved.
[15:0]	BRD	<p><b>Baud Rate Divider</b></p> <p>The field indicates the baud rate divider. This field is used in baud rate calculation.</p> <p><b>Note:</b> The detail description is shown in Table 6.23-3.</p>

**UART IrDA Control Register (UART\_IRDA)**

Register	Offset	R/W	Description	Reset Value
UART_IRDA	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	<p><b>IrDA Inverse Receive Input Signal</b>                      0 = None inverse receiving input signal.                      1 = Inverse receiving input signal. (Default)</p> <p><b>Note1:</b> 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.</p> <p><b>Note2:</b> This bit is valid when FUNCSEL (UART_FUNCSEL[1:0]) is select IrDA function.</p>
[5]	TXINV	<p><b>IrDA Inverse Transmitting Output Signal</b>                      0 = None inverse transmitting signal. (Default).                      1 = Inverse transmitting output signal.</p> <p><b>Note1:</b> 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.</p> <p><b>Note2:</b> This bit is valid when FUNCSEL (UART_FUNCSEL[1:0]) is select IrDA function.</p>
[4:2]	Reserved	Reserved.
[1]	TXEN	<p><b>IrDA Receiver/Transmitter Selection Enable Bit</b>                      0 = IrDA Transmitter Disabled and Receiver Enabled. (Default)                      1 = IrDA Transmitter Enabled and Receiver Disabled.</p> <p><b>Note:</b> In IrDA mode, the BAUDM1 (UART_BAUD [29]) register must be disabled, the baud equation must be <math>\text{Clock} / (16 * (\text{BRD} + 2))</math>.</p>
[0]	Reserved	Reserved.

**UART Alternate Control/Status Register (UART\_ALTCTL)**

Register	Offset	R/W	Description	Reset Value
UART_ALTCTL	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
LINTXEN	LINRXEN	Reserved		BRKFL			

Bits	Description	
[31:24]	ADDRMV	<p><b>Address Match Value</b> This field contains the RS-485 address match values. <b>Note:</b> This field is used for RS-485 auto address detection mode.</p>
[23:21]	Reserved	Reserved.
[20:19]	ABRDBITS	<p><b>Auto-baud Rate Detect Bit Length</b> 00 = 1-bit time from Start bit to the 1st rising edge. The input pattern shall be 0x01. 01 = 2-bit time from Start bit to the 1st rising edge. The input pattern shall be 0x02. 10 = 4-bit time from Start bit to the 1st rising edge. The input pattern shall be 0x08. 11 = 8-bit time from Start bit to the 1st rising edge. The input pattern shall be 0x80. <b>Note:</b> The calculation of bit number includes the START bit.</p>
[18]	ABRDEN	<p><b>Auto-baud Rate Detect Enable Bit</b> 0 = Auto-baud rate detect function Disabled. 1 = Auto-baud rate detect function Enabled. <b>Note:</b> This bit is cleared automatically after auto-baud detection is finished.</p>
[17]	ABRIF	<p><b>Auto-baud Rate Interrupt Flag (Read Only)</b> 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. <b>Note:</b> This bit is read only, but it can be cleared by writing "1" to ABRDIOF(UART_FIFOSTS[2]) and ABRDIF(UART_FIFOSTS[1]).</p>
[16]	Reserved	Reserved.
[15]	ADDRDEN	<p><b>RS-485 Address Detection Enable Bit</b> This bit is used to enable RS-485 Address Detection mode. 0 = Address detection mode Disabled. 1 = Address detection mode Enabled.</p>

		<b>Note:</b> This bit is used for RS-485 any operation mode.
[14:11]	<b>Reserved</b>	Reserved.
[10]	<b>RS485AUD</b>	<b>RS-485 Auto Direction Function (AUD)</b> 0 = RS-485 Auto Direction Operation function (AUD) Disabled. 1 = RS-485 Auto Direction Operation function (AUD) Enabled. <b>Note:</b> It can be active with RS-485_AAD or RS-485_NMM operation mode.
[9]	<b>RS485AAD</b>	<b>RS-485 Auto Address Detection Operation Mode (AAD)</b> 0 = RS-485 Auto Address Detection Operation mode (AAD) Disabled. 1 = RS-485 Auto Address Detection Operation mode (AAD) Enabled. <b>Note:</b> It cannot be active with RS-485_NMM operation mode.
[8]	<b>RS485NMM</b>	<b>RS-485 Normal Multi-drop Operation Mode (NMM)</b> 0 = RS-485 Normal Multi-drop Operation mode (NMM) Disabled. 1 = RS-485 Normal Multi-drop Operation mode (NMM) Enabled. <b>Note:</b> It cannot be active with RS-485_AAD operation mode.
[7]	<b>LINTXEN</b>	<b>LIN TX Break Mode Enable Bit</b> 0 = LIN TX Break mode Disabled. 1 = LIN TX Break mode Enabled. <b>Note:</b> When TX break field transfer operation finished, this bit will be cleared automatically.
[6]	<b>LINRXEN</b>	<b>LIN RX Enable Bit</b> 0 = LIN RX mode Disabled. 1 = LIN RX mode Enabled.
[5:4]	<b>Reserved</b>	Reserved.
[3:0]	<b>BRKFL</b>	<b>UART LIN Break Field Length</b> This field indicates a 4-bit LIN TX break field count. <b>Note1:</b> This break field length is BRKFL + 1. <b>Note2:</b> According to LIN spec, the reset value is 0xC (break field length = 13).

**UART Function Select Register (UART\_FUNCSEL)**

Register	Offset	R/W	Description	Reset Value
UART_FUNCSEL	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				TXRXDIS	Reserved	FUNCSEL	

Bits	Description
[31:4]	<b>Reserved</b> Reserved.
[3]	<b>TXRXDIS</b> <b>TX and RX Disable Bit</b> Setting this bit can disable TX and RX. 0 = TX and RX Enabled. 1 = TX and RX Disabled. <b>Note:</b> The TX and RX will not disable immediately when this bit is set. The TX and RX complete current task before disable TX and RX. When TX and RX disable, the TXRXACT (UART_FIFOSTS[31]) is cleared.
[2]	<b>Reserved</b> Reserved.
[1:0]	<b>FUNCSEL</b> <b>Function Select</b> 00 = UART function. 01 = LIN function. 10 = IrDA function. 11 = RS-485 function.



**UART LIN Control Register (UART\_LINCTL)**

Register	Offset	R/W	Description	Reset Value
UART_LINCTL	UARTx_BA+0x34	R/W	UART LIN Control Register (Only for UART0 and UART1)	0x000C_0000

31	30	29	28	27	26	25	24
PID							
23	22	21	20	19	18	17	16
HSEL		BSL			BRKFL		
15	14	13	12	11	10	9	8
Reserved			BITERREN	LINRXOFF	BRKDETEN	IDPEN	SENDH
7	6	5	4	3	2	1	0
Reserved			MUTE	SLVDUEN	SLVAREN	SLVHDEN	SLVEN

Bits	Description
[31:24]	<p><b>PID</b></p> <p><b>LIN PID Bits</b> This field contains the LIN frame ID value when in LIN function mode, the frame ID parity can be generated by software or hardware depends on IDPEN (UART_LINCTL[9]) = 1. If the parity generated by hardware, user fill ID0~ID5 (PID [29:24] ), hardware will calculate P0 (PID[30]) and P1 (PID[31]), otherwise user must filled frame ID and parity in this field. <b>Note1:</b> User can fill any 8-bit value to this field and the bit 24 indicates ID0 (LSB first). <b>Note2:</b> This field can be used for LIN master mode or slave mode.</p>
[23:22]	<p><b>HSEL</b></p> <p><b>LIN Header Select</b> 00 = The LIN header includes "break field". 01 = The LIN header includes "break field" and "sync field". 10 = The LIN header includes "break field", "sync field" and "frame ID field". 11 = Reserved. <b>Note:</b> This bit is used to master mode for LIN to send header field (SENDH (UART_LINCTL [8]) = 1) or used to slave to indicates exit from mute mode condition (MUTE (UART_LINCTL[4] = 1).</p>
[21:20]	<p><b>BSL</b></p> <p><b>LIN Break/Sync Delimiter Length</b> 00 = The LIN break/sync delimiter length is 1-bit time. 01 = The LIN break/sync delimiter length is 2-bit time. 10 = The LIN break/sync delimiter length is 3-bit time. 11 = The LIN break/sync delimiter length is 4-bit time. <b>Note:</b> This bit used for LIN master to sending header field.</p>
[19:16]	<p><b>BRKFL</b></p> <p><b>LIN Break Field Length</b> This field indicates a 4-bit LIN TX break field count. <b>Note1:</b> These registers are shadow registers of BRKFL (UART_ALTCTL[3:0]), User can read/write it by setting BRKFL (UART_ALTCTL[3:0]) or BRKFL (UART_LINCTL[19:16]). <b>Note2:</b> This break field length is BRKFL + 1. <b>Note3:</b> According to LIN spec, the reset value is 12 (break field length = 13).</p>
[15:13]	<p><b>Reserved</b></p> <p>Reserved.</p>

[12]	<b>BITERREN</b>	<p><b>Bit Error Detect Enable Bit</b></p> <p>0 = Bit error detection function Disabled. 1 = Bit error detection function Enabled.</p> <p><b>Note:</b> In LIN function mode, when occur bit error, the BITEF (UART_LINSTS[9]) flag will be asserted. If the LINIEN (UART_INTEN[8]) = 1, an interrupt will be generated.</p>
[11]	<b>LINRXOFF</b>	<p><b>LIN Receiver Disable Bit</b></p> <p>If the receiver is enabled (LINRXOFF (UART_LINCTL[11]) = 0), all received byte data will be accepted and stored in the RX FIFO, and if the receiver is disabled (LINRXOFF (UART_LINCTL[11]) = 1), all received byte data will be ignore.</p> <p>0 = LIN receiver Enabled. 1 = LIN receiver Disabled.</p> <p><b>Note:</b> This bit is only valid when operating in LIN function mode (FUNCSEL (UART_FUNCSEL[1:0]) = 01).</p>
[10]	<b>BRKDETEN</b>	<p><b>LIN Break Detection Enable Bit</b></p> <p>When detect consecutive dominant greater than 11 bits, and are followed by a delimiter character, the BRKDETF (UART_LINSTS[8]) flag is set at the end of break field. If the LINIEN (UART_INTEN [8])=1, an interrupt will be generated.</p> <p>0 = LIN break detection Disabled . 1 = LIN break detection Enabled.</p>
[9]	<b>IDPEN</b>	<p><b>LIN ID Parity Enable Bit</b></p> <p>0 = LIN frame ID parity Disabled. 1 = LIN frame ID parity Enabled.</p> <p><b>Note1:</b> This bit can be used for LIN master to sending header field (SENDH (UART_LINCTL[8])) = 1 and HSEL (UART_LINCTL[23:22]) = 10 or be used for enable LIN slave received frame ID parity checked.</p> <p><b>Note2:</b> This bit is only used when the operation header transmitter is in HSEL (UART_LINCTL[23:22]) = 10.</p>
[8]	<b>SENDH</b>	<p><b>LIN TX Send Header Enable Bit</b></p> <p>The LIN TX header can be “break field” or “break and sync field” or “break, sync and frame ID field”, it is depend on setting HSEL (UART_LINCTL[23:22]).</p> <p>0 = Send LIN TX header Disabled. 1 = Send LIN TX header Enabled.</p> <p><b>Note1:</b> This bit is shadow bit of LINTXEN (UART_ALTCTL [7]); user can read/write it by setting LINTXEN (UART_ALTCTL [7]) or SENDH (UART_LINCTL [8]).</p> <p><b>Note2:</b> When transmitter header field (it may be “break” or “break + sync” or “break + sync + frame ID” selected by HSEL (UART_LINCTL[23:22]) field) transfer operation finished, this bit will be cleared automatically.</p>
[7:5]	<b>Reserved</b>	Reserved.
[4]	<b>MUTE</b>	<p><b>LIN Mute Mode Enable Bit</b></p> <p>0 = LIN mute mode Disabled. 1 = LIN mute mode Enabled.</p> <p><b>Note:</b> The exit from mute mode condition and each control and interactions of this field are explained in 6.23.5.10 (LIN slave mode).</p>
[3]	<b>SLVDUEN</b>	<p><b>LIN Slave Divider Update Method Enable Bit</b></p> <p>0 = UART_BAUD updated is written by software (if no automatic resynchronization update occurs at the same time). 1 = UART_BAUD is updated at the next received character. User must set the bit before checksum reception.</p> <p><b>Note1:</b> This bit only valid when in LIN slave mode (SLVEN (UART_LINCTL[0]) = 1).</p> <p><b>Note2:</b> This bit used for LIN Slave Automatic Resynchronization mode. (for Non-Automatic Resynchronization mode, this bit should be kept cleared)</p>

		<b>Note3:</b> The control and interactions of this field are explained in 6.23.5.10 (Slave mode with automatic resynchronization).
[2]	<b>SLVAREN</b>	<p><b>LIN Slave Automatic Resynchronization Mode Enable Bit</b></p> <p>0 = LIN automatic resynchronization Disabled. 1 = LIN automatic resynchronization Enabled.</p> <p><b>Note1:</b> This bit only valid when in LIN slave mode (SLVEN (UART_LINCTL[0]) = 1).</p> <p><b>Note2:</b> When operation in Automatic Resynchronization mode, the baud rate setting must be mode2 (BAUDM1 (UART_BAUD [29]) and BAUDM0 (UART_BAUD [28]) must be 1).</p> <p><b>Note3:</b> The control and interactions of this field are explained in 6.23.5.10 (Slave mode with automatic resynchronization).</p>
[1]	<b>SLVHDEN</b>	<p><b>LIN Slave Header Detection Enable Bit</b></p> <p>0 = LIN slave header detection Disabled. 1 = LIN slave header detection Enabled.</p> <p><b>Note1:</b> This bit only valid when in LIN slave mode (SLVEN (UART_LINCTL[0]) = 1).</p> <p><b>Note2:</b> In LIN function mode, when detect header field (break + sync + frame ID), SLVHDETF (UART_LINSTS [0]) flag will be asserted. If the LINIEN (UART_INTEN[8]) = 1, an interrupt will be generated.</p>
[0]	<b>SLVEN</b>	<p><b>LIN Slave Mode Enable Bit</b></p> <p>0 = LIN slave mode Disabled. 1 = LIN slave mode Enabled.</p>

**UART LIN Status Register (UART\_LINSTS)**

Register	Offset	R/W	Description	Reset Value
UART_LINSTS	UARTx_BA+0x38	R/W	UART LIN Status Register (Only for UART0 and UART1)	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						BITEF	BRKDETf
7	6	5	4	3	2	1	0
Reserved				SLVSYNCF	SLVIDPEF	SLVHEF	SLVHDETf

Bits	Description	
[31:10]	Reserved	Reserved.
[9]	BITEF	<p><b>Bit Error Detect Status Flag</b></p> <p>At TX transfer state, hardware will monitor the bus state, if the input pin (UART_RXD) state not equals to the output pin (UART_TXD) state, BITEF (UART_LINSTS[9]) will be set.</p> <p>When occur bit error, if the LINIEN (UART_INTEN[8]) = 1, an interrupt will be generated.</p> <p>0 = Bit error not detected. 1 = Bit error detected.</p> <p><b>Note1:</b> This bit can be cleared by writing 1 to it. <b>Note2:</b> This bit is only valid when enable bit error detection function (BITERREN (UART_LINCTL [12]) = 1).</p>
[8]	BRKDETf	<p><b>LIN Break Detection Flag</b></p> <p>This bit is set by hardware when a break is detected and be cleared by writing 1 to it through software.</p> <p>0 = LIN break not detected. 1 = LIN break detected.</p> <p><b>Note1:</b> This bit can be cleared by writing 1 to it. <b>Note2:</b> This bit is only valid when LIN break detection function is enabled (BRKDETEN (UART_LINCTL[10]) =1).</p>
[7:4]	Reserved	Reserved.
[3]	SLVSYNCF	<p><b>LIN Slave Sync Field</b></p> <p>This bit indicates that the LIN sync field is being analyzed in Automatic Resynchronization mode. When the receiver header have some error been detect, user must reset the internal circuit to re-search new frame header by writing 1 to this bit.</p> <p>0 = The current character is not at LIN sync state. 1 = The current character is at LIN sync state.</p> <p><b>Note1:</b> This bit is only valid when in LIN Slave mode (SLVEN(UART_LINCTL[0]) = 1). <b>Note2:</b> This bit can be cleared by writing 1 to it. <b>Note3:</b> When writing 1 to it, hardware will reload the initial baud rate and re-search a new</p>

		frame header.
[2]	SLVIDPEF	<p><b>LIN Slave ID Parity Error Flag</b></p> <p>This bit is set by hardware when received frame ID parity is not correct.</p> <p>0 = No active.</p> <p>1 = Received frame ID parity is not correct.</p> <p><b>Note1:</b> This bit can be cleared by writing 1 to it.</p> <p><b>Note2:</b> This bit is only valid when in LIN slave mode (SLVEN (UART_LINCTL [0])= 1) and enable LIN frame ID parity check function IDPEN (UART_LINCTL [9]).</p>
[1]	SLVHEF	<p><b>LIN Slave Header Error Flag</b></p> <p>This bit is set by hardware when a LIN header error is detected in LIN slave mode and be cleared by writing 1 to it. The header errors include “break delimiter is too short (less than 0.5 bit time)”, “frame error in sync field or Identifier field”, “sync field data is not 0x55 in Non-Automatic Resynchronization mode”, “sync field deviation error with Automatic Resynchronization mode”, “sync field measure time-out with Automatic Resynchronization mode” and “LIN header reception time-out”.</p> <p>0 = LIN header error not detected.</p> <p>1 = LIN header error detected.</p> <p><b>Note1:</b> This bit can be cleared by writing 1 to it.</p> <p><b>Note2:</b> This bit is only valid when UART is operated in LIN slave mode (SLVEN (UART_LINCTL [0]) = 1) and enables LIN slave header detection function (SLVHDEN (UART_LINCTL [1])).</p>
[0]	SLVHDETF	<p><b>LIN Slave Header Detection Flag</b></p> <p>This bit is set by hardware when a LIN header is detected in LIN slave mode and be cleared by writing 1 to it.</p> <p>0 = LIN header not detected.</p> <p>1 = LIN header detected (break + sync + frame ID).</p> <p><b>Note1:</b> This bit can be cleared by writing 1 to it.</p> <p><b>Note2:</b> This bit is only valid when in LIN slave mode (SLVEN (UART_LINCTL [0]) = 1) and enable LIN slave header detection function (SLVHDEN (UART_LINCTL [1])).</p> <p><b>Note3:</b> When enable ID parity check IDPEN (UART_LINCTL [9]), if hardware detect complete header (“break + sync + frame ID”), the SLVHDETF will be set whether the frame ID correct or not.</p>

**UART Baud Rate Compensation Register (UART\_BRCOMP)**

Register	Offset	R/W	Description	Reset Value
UART_BRCOMP	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	<b>Baud Rate Compensation Decrease</b> 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	<b>Baud Rate Compensation Patten</b> These 9-bits are used to define the relative bit is compensated or not. BRCOMP[7:0] is used to define the compensation of UART_DAT[7:0] and BRCOMP[8] is used to define the parity bit.

**UART Wake-up Control Register (UART\_WKCTL)**

Register	Offset	R/W	Description	Reset Value
UART_WKCTL	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			WKOUTEN	WKRS485EN	WKRFRTEN	WKDATEN	WKCTSEN

Bits	Description	
[31:5]	Reserved	Reserved.
[4]	WKOUTEN	<p><b>Received Data FIFO Reached Threshold Time-out Wake-up Enable Bit</b></p> <p>0 = Received Data FIFO reached threshold time-out wake-up system function Disabled.</p> <p>1 = Received Data FIFO reached threshold time-out wake-up system function Enabled, when the system is in Power-down mode, Received Data FIFO reached threshold time-out will wake-up system from Power-down mode.</p> <p><b>Note:</b> It is suggest the function is enabled when the WKRFRTEN (UART_WKCTL[2]) is set to 1.</p>
[3]	WKRS485EN	<p><b>RS-485 Address Match (AAD Mode) Wake-up Enable Bit</b></p> <p>0 = RS-485 Address Match (AAD mode) wake-up system function Disabled.</p> <p>1 = RS-485 Address Match (AAD mode) wake-up system function Enabled, when the system is in Power-down mode, RS-485 Address Match will wake-up system from Power-down mode.</p> <p><b>Note:</b> This bit is used for RS-485 Auto Address Detection (AAD) mode in RS-485 function mode and ADDRDEN (UART_ALTCTL[15]) is set to 1.</p>
[2]	WKRFRTEN	<p><b>Received Data FIFO Reached Threshold Wake-up Enable Bit</b></p> <p>0 = Received Data FIFO reached threshold wake-up system function Disabled.</p> <p>1 = Received Data FIFO reached threshold wake-up system function Enabled, when the system is in Power-down mode, Received Data FIFO reached threshold will wake-up system from Power-down mode.</p>
[1]	WKDATEN	<p><b>Incoming Data Wake-up Enable Bit</b></p> <p>0 = Incoming data wake-up system function Disabled.</p> <p>1 = Incoming data wake-up system function Enabled, when the system is in Power-down mode, incoming data will wake-up system from Power-down mode.</p>
[0]	WKCTSEN	<p><b>nCTS Wake-up Enable Bit</b></p> <p>0 = nCTS Wake-up system function Disabled.</p> <p>1 = nCTS Wake-up system function Enabled, when the system is in Power-down mode, an external nCTS change will wake-up system from Power-down mode.</p>

**UART Wake-up Status Register (UART\_WKSTS)**

Register	Offset	R/W	Description	Reset Value
UART_WKSTS	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	<p><b>Received Data FIFO Threshold Time-out Wake-up Flag</b>                      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 wake-up.  <b>Note1:</b> If WKROUTEN (UART_WKCTL[4]) is enabled, the Received Data FIFO reached threshold time-out wake-up cause this bit is set to '1'.  <b>Note2:</b> This bit can be cleared by writing '1' to it.</p>
[3]	RS485WKF	<p><b>RS-485 Address Match (AAD Mode) Wake-up Flag</b>                      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.  <b>Note1:</b> If WKRS485EN (UART_WKCTL[3]) is enabled, the RS-485 Address Match (AAD mode) wake-up cause this bit is set to '1'.  <b>Note2:</b> This bit can be cleared by writing '1' to it.</p>
[2]	RFRTWKF	<p><b>Received Data FIFO Reached Threshold Wake-up Flag</b>                      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.  <b>Note1:</b> If WKRFRTEN (UART_WKCTL[2]) is enabled, the Received Data FIFO Reached Threshold wake-up cause this bit is set to '1'.  <b>Note2:</b> This bit can be cleared by writing '1' to it.</p>
[1]	DATWKF	<p><b>Incoming Data Wake-up Flag</b>                      This bit is set if chip wake-up from power-down state by data wake-up.</p>



		<p>0 = Chip stays in power-down state.                  1 = Chip wake-up from power-down state by Incoming Data wake-up.  <b>Note1:</b> If WKDATEN (UART_WKCTL[1]) is enabled, the Incoming Data wake-up cause this bit is set to '1'.  <b>Note2:</b> This bit can be cleared by writing '1' to it.</p>
[0]	CTSWKF	<p><b>nCTS Wake-up Flag</b>                  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.  <b>Note1:</b> If WKCTSEN (UART_WKCTL[0]) is enabled, the nCTS wake-up cause this bit is set to '1'.  <b>Note2:</b> This bit can be cleared by writing '1' to it.</p>

**UART Incoming Data Wake-up Compensation Register (UART\_DWKCOMP)**

Register	Offset	R/W	Description	Reset Value
UART_DWKCOMP	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	<p><b>Start Bit Compensation Value</b></p> <p>These bits field indicate how many clock cycle selected by UART_CLK do the UART controller can get the 1<sup>st</sup> bit (start bit) when the device is wake-up from power-down mode.</p> <p><b>Note:</b> It is valid only when WKDATEN (UART_WKCTL[1]) is set.</p>

## 6.24 Watchdog Timer (WDT)

### 6.24.1 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, the Watchdog Timer supports the function to wake up system from Idle/Power-down mode.

### 6.24.2 Features

- Supports 18-bit free running up counter
- Selectable time-out interval ( $2^4 \sim 2^{18}$ ) and the time-out interval is 1.6 ms ~ 26.214s if WDT\_CLK is 10 kHz Supports selectable WDT reset delay period between WDT time-out event to WDT reset system event, and it includes 1026 · 130 · 18 or 3 \* WDT\_CLK delay period
- System kept in reset state about 63 \* WDT\_CLK period time after system reset event occurred
- Supports to force WDT function enabled after chip powered on or reset by setting CWDTEN[2:0] in Config0 register
- Supports WDT time-out wake-up function only if WDT clock source is selected as LIRC or LXT

### 6.24.3 Block Diagram

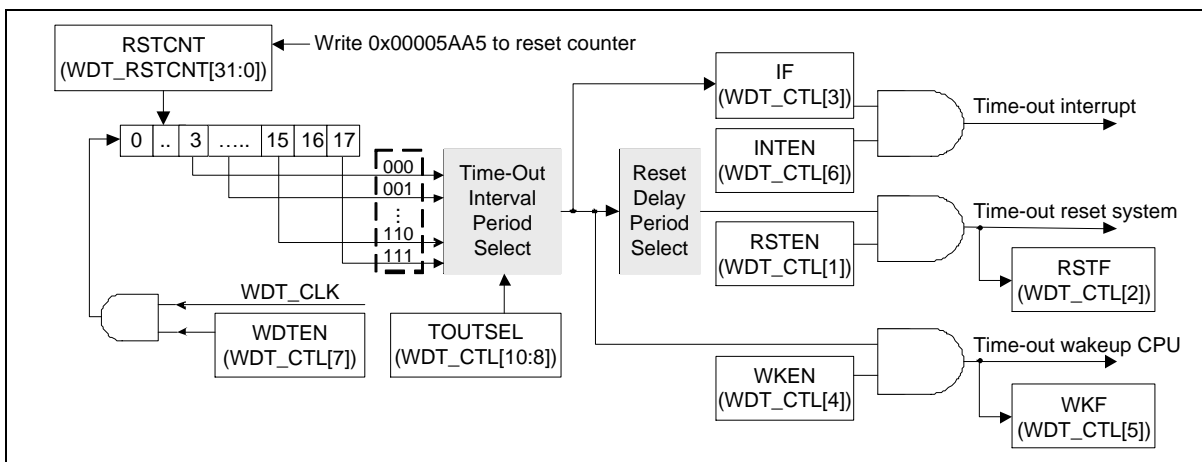


Figure 6.24-1 Watchdog Timer Block Diagram

### 6.24.4 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]).

The WDT clock control is shown in Figure 6.24-2.

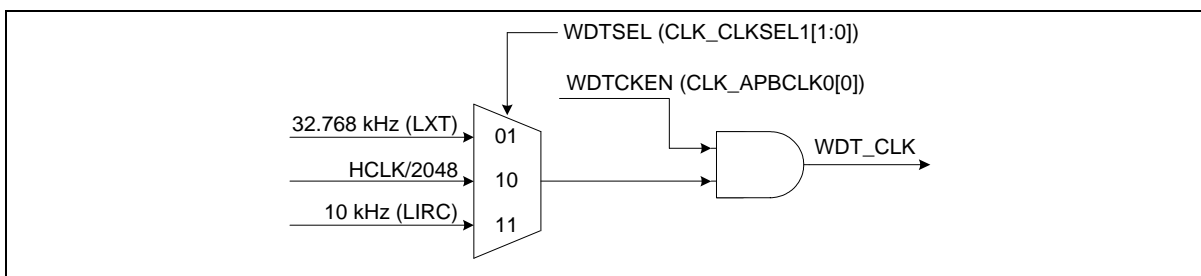


Figure 6.24-2 Watchdog Timer Clock Control

The WDT controller can also be forced enabled and works in 10 kHz after chip powered on or reset while CWDTEN[2:0] is not configured to 0x111, which CWDTEN[2] is defined in Config0[31] and CWDTEN[1:0] is defined in Config0[4:3].

### 6.24.5 Functional Description

The WDT controller includes an 18-bit free running up counter with programmable time-out intervals. Table 6.24-1 is shown the WDT time-out interval period selection and Figure 6.24-3 is shown the WDT time-out interval and reset period.

#### WDT Time-out Interrupt

Setting WDTEN (WDT\_CTL[7]) to 1 will enable the WDT counter to start counting up. There are eight time-out interval period can be selected by setting TOUTSEL (WDT\_CTL [10:8]). When the WDT counter value reaches the TOUTSEL setting, WDT time-out event will occur and WDT time-out interrupt flag IF (WDT\_CTL[3]) will be set to 1 immediately.

#### WDT Reset Delay Period and Reset System

User can set RSTEN (WDT\_CTL[1]) to enable WDT time-out reset system function, and system will be reset after IF (WDT\_CTL[3]) is generated and WDT counter is not reset before  $T_{RSTD}$  reset delay period expired. User can program 0x00005AA5 in WDT\_RSTCNT register to reset the 18-bit WDT up counter value to avoid generate WDT time-out reset system signal. Moreover, user can set RSTDSEL (WDT\_ALTCTL[1:0]) to select reset delay period for adjusting delay period after IF occurred until system reset. If the WDT counter value has not been cleared after the specific  $T_{RSTD}$  reset delay period expires, the RSTF (WDT\_CTL[2]) will set to 1 then chip enters to reset state immediately.

Figure 6.24-3 shows the detail WDT time-out and reset period timing. If system reset event occurred by WDT time-out reset function, the  $T_{RST}$  reset period will keep  $63 * WDT\_CLK$ , after the  $T_{RST}$  reset period, chip will restart executing program and RSTF will keep 1 until writing 1 to clear it. WDTRF (SYS\_RSTSTS[2]) will also be set after system has been reset by WDT time-out reset event.

TOUTSEL	Time-Out Interval Period ( $T_{TIS}$ )	Reset Delay Period ( $T_{RSTD}$ )
000	$2^4 * T_{WDT}$	$(3/18/130/1026) * T_{WDT}$
001	$2^6 * T_{WDT}$	$(3/18/130/1026) * T_{WDT}$
010	$2^8 * T_{WDT}$	$(3/18/130/1026) * T_{WDT}$
011	$2^{10} * T_{WDT}$	$(3/18/130/1026) * T_{WDT}$
100	$2^{12} * T_{WDT}$	$(3/18/130/1026) * T_{WDT}$
101	$2^{14} * T_{WDT}$	$(3/18/130/1026) * T_{WDT}$
110	$2^{16} * T_{WDT}$	$(3/18/130/1026) * T_{WDT}$
111	$2^{18} * T_{WDT}$	$(3/18/130/1026) * T_{WDT}$

Table 6.24-1 Watchdog Timer Time-out Interval Period Selection

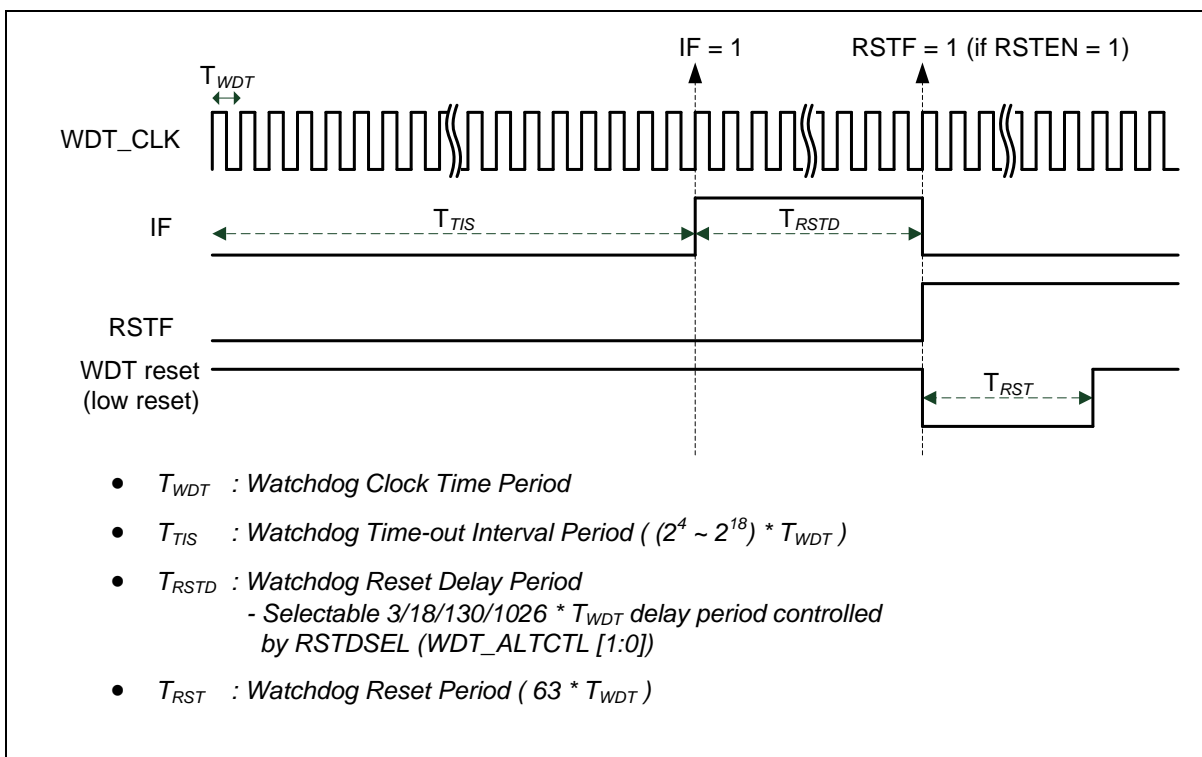


Figure 6.24-3 Watchdog Timer Time-out Interval and Reset Period Timing

**WDT Wake-up**

If WDT clock source is selected to LIRC (10 kHz) or LXT (32 kHz), system can be waken-up from Power-down mode while WDT time-out interrupt signal is generated and both INTEN (WDT\_CTL[6]) and WKEN (WDT\_CTL[4]) are enabled. In the meanwhile, the WKF (WDT\_CTL[5]) will set to 1 automatically, user can check WKF (WDT\_CTL[5]) status by software to recognize the system has been waken-up by WDT time-out interrupt event or not.

### 6.24.6 Register Map

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

Register	Offset	R/W	Description	Reset Value
<b>WDT Base Address:</b>				
<b>WDT_BA = 0x4000_4000</b>				
<b>WDT_CTL</b>	WDT_BA+0x00	R/W	WDT Control Register	0x0000_0700
<b>WDT_ALTCTL</b>	WDT_BA+0x04	R/W	WDT Alternative Control Register	0x0000_0000
<b>WDT_RSTCNT</b>	WDT_BA+0x08	W	WDT Reset Counter Register	0x0000_0000

6.24.7 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_0700

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]	<p><b>ICEDEBUG</b></p> <p><b>ICE Debug Mode Acknowledge Disable Bit (Write Protect)</b>                      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.  <b>Note:</b> This bit is write protected. Refer to the SYS_REGLCTL register.</p>
[30]	<p><b>SYNC</b></p> <p><b>WDT Enable Control SYNC Flag Indicator (Read Only)</b>                      If user execute enable/disable WDTEEN (WDT_CTL[7]), this flag can be indicated enable/disable WDTEEN function is completed or not.                      0 = Set WDTEEN bit is completed.                      1 = Set WDTEEN bit is synchronizing and not become active yet..  <b>Note:</b> Perform enable or disable WDTEEN bit needs 2 * WDT_CLK period to become active.</p>
[29:11]	<p><b>Reserved</b></p> <p>Reserved.</p>
[10:8]	<p><b>TOUTSEL</b></p> <p><b>WDT Time-out Interval Selection (Write Protect)</b>                      These three bits select the time-out interval period after WDT starts counting.                      000 = 2<sup>4</sup> * WDT_CLK.                      001 = 2<sup>6</sup> * WDT_CLK.                      010 = 2<sup>8</sup> * WDT_CLK.                      011 = 2<sup>10</sup> * WDT_CLK.                      100 = 2<sup>12</sup> * WDT_CLK.                      101 = 2<sup>14</sup> * WDT_CLK.                      110 = 2<sup>16</sup> * WDT_CLK.                      111 = 2<sup>18</sup> * WDT_CLK.  <b>Note:</b> This bit is write protected. Refer to the SYS_REGLCTL register.</p>
[7]	<p><b>WDTEN</b></p> <p><b>WDT Enable Bit (Write Protect)</b></p>

		<p>0 = Set WDT counter stop, and internal up counter value will be reset also.                      1 = Set WDT counter start .</p> <p><b>Note1:</b> This bit is write protected. Refer to the SYS_REGLCTL register.</p> <p><b>Note2:</b> Perform enable or disable WDTEN bit needs 2 * WDT_CLK period to become active, user can read SYNC (WDT_CTL[30]) to check enable/disable command is completed or not.</p> <p><b>Note3:</b> If CWDTEN[2:0] (combined with Config0[31] and Config0[4:3]) bits is not configure to 0x111, this bit is forced as 1 and user cannot change this bit to 0.</p>
[6]	INTEN	<p><b>WDT Time-out Interrupt Enable Bit (Write Protect)</b></p> <p>If this bit is enabled, when WDT time-out event occurs, the IF (WDT_CTL[3]) will be set to 1 and WDT time-out interrupt signal is generated and inform to CPU.</p> <p>0 = WDT time-out interrupt Disabled.                      1 = WDT time-out interrupt Enabled.</p> <p><b>Note:</b> This bit is write protected. Refer to the SYS_REGLCTL register.</p>
[5]	WKF	<p><b>WDT Time-out Wake-up Flag (Write Protect)</b></p> <p>This bit indicates the WDT time-out event has triggered chip wake-up or not.</p> <p>0 = WDT does not cause chip wake-up.                      1 = Chip wake-up from Idle or Power-down mode when WDT time-out interrupt signal is generated.</p> <p><b>Note:</b> This bit is cleared by writing 1 to it.</p>
[4]	WKEN	<p><b>WDT Time-out Wake-up Function Control (Write Protect)</b></p> <p>If this bit is set to 1, while WDT time-out interrupt flag IF (WDT_CTL[3]) is generated and interrupt enable bit INTEN (WDT_CTL[6]) is enabled, the WDT time-out interrupt signal will generate a event to trigger CPU wake-up.</p> <p>0 = Trigger wake-up event function Disabled if WDT time-out interrupt signal generated.                      1 = Trigger wake-up event function Enabled if WDT time-out interrupt signal generated.</p> <p><b>Note1:</b> This bit is write protected. Refer to the SYS_REGLCTL register.</p> <p><b>Note2:</b> Chip can be woken-up while WDT time-out interrupt signal generated only if WDT clock source is selected to LIRC (10 kHz) or LXT (32 kHz).</p>
[3]	IF	<p><b>WDT Time-out Interrupt Flag</b></p> <p>This bit will set to 1 while WDT up counter value reaches the selected WDT time-out interval</p> <p>0 = WDT time-out interrupt event did not occur.                      1 = WDT time-out interrupt event occurred.</p> <p><b>Note:</b> This bit is cleared by writing 1 to it.</p>
[2]	RSTF	<p><b>WDT Time-out Reset Flag</b></p> <p>This bit indicates the system has been reset by WDT time-out reset system event or not.</p> <p>0 = WDT time-out reset system event did not occur.                      1 = WDT time-out reset system event has been occurred.</p> <p><b>Note:</b> This bit is cleared by writing 1 to it.</p>
[1]	RSTEN	<p><b>WDT Time-out Reset Enable Bit (Write Protect)</b></p> <p>Setting this bit will enable the WDT time-out reset system function If the WDT up counter value has not been cleared after the specific WDT reset delay period expires.</p> <p>0 = WDT time-out reset system function Disabled.                      1 = WDT time-out reset system function Enabled.</p> <p><b>Note:</b> This bit is write protected. Refer to the SYS_REGLCTL register.</p>
[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
[1:0]	<p><b>RSTDSEL</b></p> <p><b>WDT Reset Delay Period Selection (Write Protect)</b>                      When WDT time-out event happened, user has a time named WDT Reset Delay Period to execute WDT counter reset to prevent WDT time-out reset system occurred. User can select a suitable setting of RSTDSEL for application program.                      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.  <b>Note1:</b> This bit is write protected. Refer to the SYS_REGLCTL register.  <b>Note2:</b> This register will be reset to 0 if WDT time-out reset system event occurred.</p>

**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]	<p><b>RSTCNT</b></p> <p><b>WDT Reset Counter Register</b></p> <p>Writing 0x00005AA5 to this field will reset the internal 18-bit WDT up counter value to 0.</p> <p><b>Note:</b> Perform RSTCNT to reset counter needs 2 * WDT_CLK period to become active.</p>

## 6.25 Window Watchdog Timer (WWDT)

### 6.25.1 Overview

The Window Watchdog Timer (WWDT) is used to perform a system reset while WWDT counter is not reload within a specified window period when application program run to uncontrollable status by any unpredictable condition.

### 6.25.2 Features

- Supports 6-bit down counter value CNTDAT (WWDT\_CNT[5:0]) and maximum 6-bit compare value CMPDAT (WWDT\_CTL[21:16]) to adjust the WWDT compare time-out window period flexible
- Supports PSCSEL (WWDT\_CTL[11:8]) to programmable maximum 11-bit prescale counter period of WWDT counter
- WWDT counter suspends in Idle/Power-down mode
- WWDT counter only can be reloaded within in valid window period to prevent system reset

### 6.25.3 Block Diagram

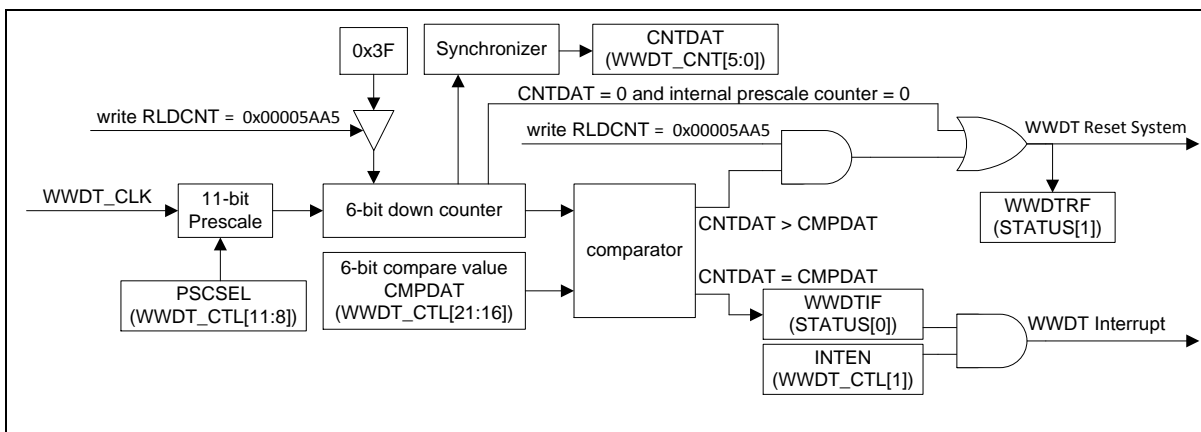


Figure 6.25-1 WWDT Block Diagram

### 6.25.4 Basic Configuration

- Clock Source Configuration
  - Select the source of WWDT peripheral clock on WWDTSEL (CLK\_CLKSEL2[17:16]).
  - Enable WWDT peripheral clock in WDTCKEN (CLK\_APBCLK0[0]).

The WWDT clock control and block diagram are shown in Figure 6.25-2.

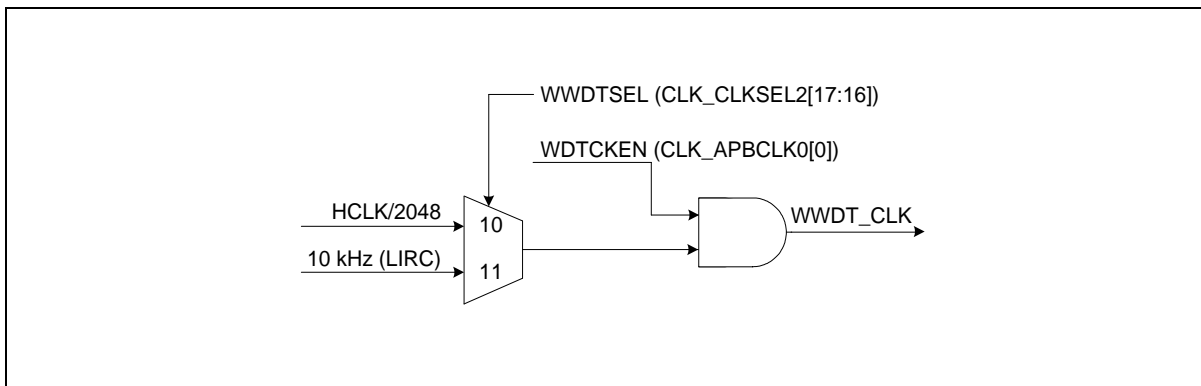


Figure 6.25-2 WWDT Clock Control

### 6.25.5 Functional Description

The WWDT includes a 6-bit down counter with programmable maximum 11-bit prescale value to define different WWDT time-out intervals. The clock source of WWDT counter is based on system clock divide 2048 (PCLK/2048) or 10 kHz internal low speed RC oscillator (LIRC) with programmable maximum 11-bit prescale which controlled by PSCSEL (WWDT\_CTL[11:8]). Also, the maximum WWDT counter compare time-out period and the correlate of PSCSEL (WWDT\_CTL[11:8]) and prescale value are listed in Table 6.25-1.

PSCSEL	Prescale Value	Max. Time-Out Period	Max. Time-Out Interval (WWDT_CLK = 10 KHz)
0000	1	$1 * 64 * T_{WWDT}$	6.4 ms
0001	2	$2 * 64 * T_{WWDT}$	12.8 ms
0010	4	$4 * 64 * T_{WWDT}$	25.6 ms
0011	8	$8 * 64 * T_{WWDT}$	51.2 ms
0100	16	$16 * 64 * T_{WWDT}$	102.4 ms
0101	32	$32 * 64 * T_{WWDT}$	204.8 ms
0110	64	$64 * 64 * T_{WWDT}$	409.6 ms
0111	128	$128 * 64 * T_{WWDT}$	819.2 ms
1000	192	$192 * 64 * T_{WWDT}$	1.2288 s
1001	256	$256 * 64 * T_{WWDT}$	1.6384 s
1010	384	$384 * 64 * T_{WWDT}$	2.4576 s
1011	512	$512 * 64 * T_{WWDT}$	3.2768 s
1100	768	$768 * 64 * T_{WWDT}$	4.9152 s
1101	1024	$1024 * 64 * T_{WWDT}$	6.5536 s

1110	1536	$1536 * 64 * T_{WWDT}$	9.8304 s
1111	2048	$2048 * 64 * T_{WWDT}$	13.1072 s

Table 6.25-1 WWDT Prescale Value Selection and Time-out Period

**WWDT Counting**

When the WWDTEN (WWDT\_CTL[0]) is set, WWDT counter will start down counting from 0x3F to 0 and the interval of each count and WWDT compare time-out period is selected by PSCSEL (WWDT\_CTL[11:8]). Table 6.25-1 shows 4-bit PSCSEL setting correlate with prescale value and maximum compare time-out period when WWDT clock source is selected as 10 kHz. To prevent program runs to disable WWDT counter counting unexpected, all the control bits in WWDT\_CTL register can only be written once after chip is powered on or reset.

**WWDT Compare Match Interrupt**

When WWDT counter value CNTDAT (WWDT\_CNT[5:0]) down counts equal to window compare value CMPDAT (WWDT\_CTL[21:16]) and internal preacale counter counts to 0, the WWDT counter compare match interrupt WWDTIF (WWDT\_STATUS[0]) will be generated and it can be cleared by writing 1. Figure 6.17-13 shows an example of WWDT compare match interrupt when PSCSEL (WWDT\_CTL[11:8]) is 0x2 and prescale value is 4.

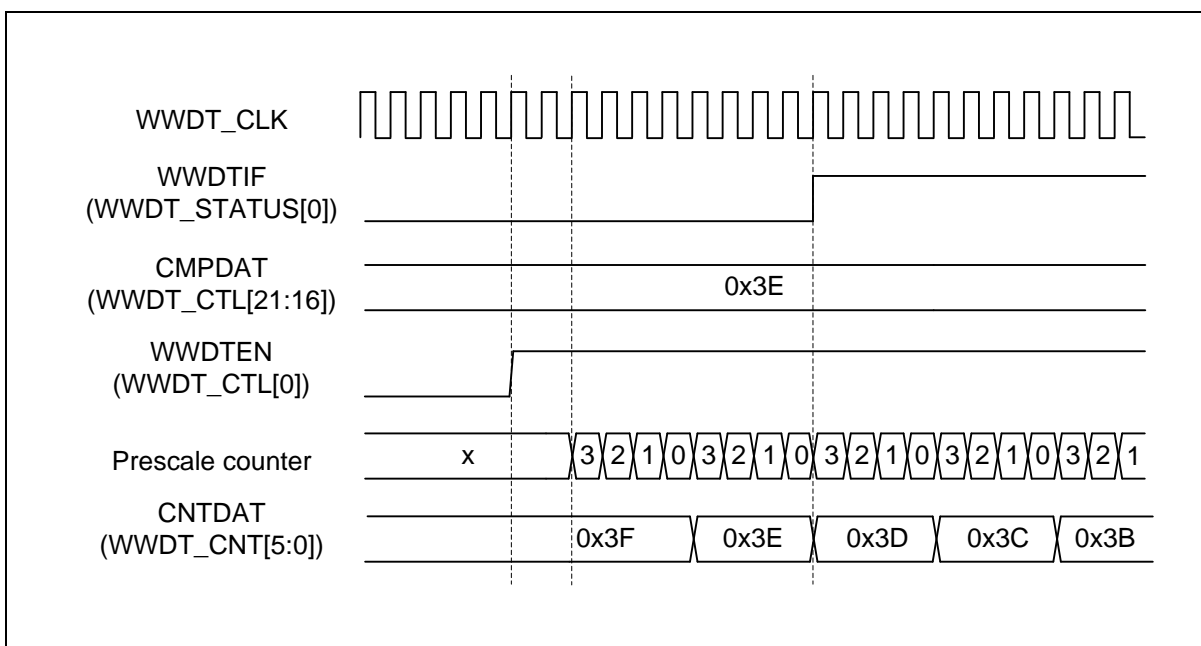


Figure 6.25-3 WWDT Compare Match Interrupt when CMPDAT is 0x3E

**WWDT Counter Reload and Reset System**

While WWDT counter and internal prescale counter both down counts to 0, WWDT reset system signal is occurred and system will be reset. If WWDTIF (WWDT\_STATUS[0]) is generated, user can write 0x00005AA5 in WWDT\_RLDCNT register to reload CNTDAT (WWDT\_CNT[5:0]) to 0x3F, and also to prevent WWDT reset system signal occurred. If current CNTDAT is larger than CMPDAT (WWDT\_CTL[21:16]), it means WWDTIF not occurred yet, when user writes 0x00005AA5 to the WWDT\_RLDCNT register, the WWDT reset system signal will be generated immediately to cause chip reset also. The behavior of counter reload and system reset are shown in Figure 6.25-4.

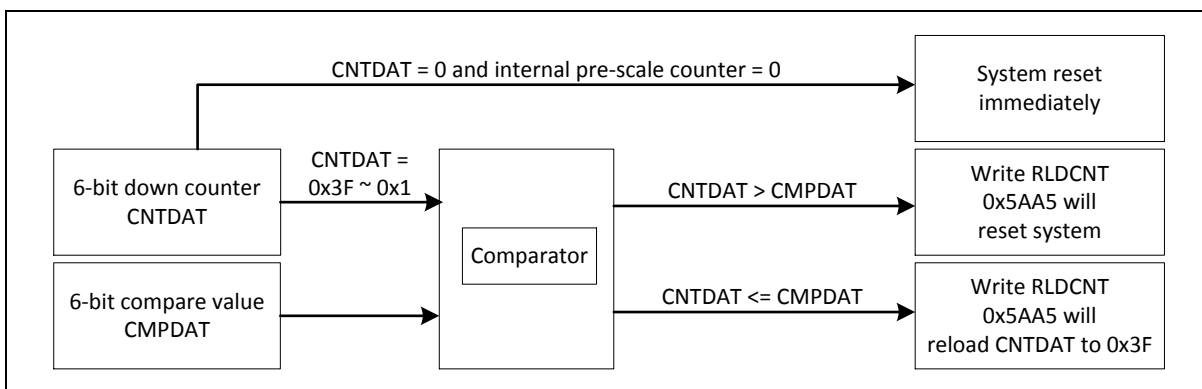


Figure 6.25-4 WWDT Counter Reload and Reset Behavior

### WWDT Window Setting Limitation

If user writes 0x00005AA5 in WWDT\_RLDCNT register to reload WWDT counter value to 0x3F, it needs (WWDT\_CLK \* 3) period to sync the reload command to actually perform reload action. Note that if user sets PSCSEL (WWDT\_CTL[11:8]) to 0x0, the counter prescale value should be as 1, and the CMPDAT (WWDT\_CTL[21:16]) must be larger than 2. Otherwise, when WWDTIF (WWDT\_STATUS[0]) is generated, writing WWDT\_RLDCNT register to reload WWDT counter value to 0x3F is unavailable and WWDT reset system event always happened after CNTDAT (WWDT\_CNT[5:0]) equals to 0. Table 6.25-2 shows the detail relation between valid CMPDAT value and 4-bit PSCSEL setting.

4-Bit PSCSEL	Prescale Value	Valid CMPDAT Value
0000	1	0x3 ~ 0x3F
0001	2	0x2 ~ 0x3F
Others	Others	0x1 ~ 0x3F

Table 6.25-2 CMPDAT Setting Limitation

6.25.6 Register Map

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

Register	Offset	R/W	Description	Reset Value
<b>WWDT Base Address:</b>				
<b>WWDT_BA = 0x4000_4100</b>				
<b>WWDT_RLDCNT</b>	WWDT_BA+0x00	W	WWDT Reload Counter Register	0x0000_0000
<b>WWDT_CTL</b>	WWDT_BA+0x04	R/W	WWDT Control Register	0x003F_0800
<b>WWDT_STATUS</b>	WWDT_BA+0x08	R/W	WWDT Status Register	0x0000_0000
<b>WWDT_CNT</b>	WWDT_BA+0x0C	R	WWDT Counter Value Register	0x0000_003F

6.25.7 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]	<p><b>RLDCNT</b></p> <p><b>WWDT Reload Counter Register</b></p> <p>Writing only 0x00005AA5 to this register will reload the WWDT counter value to 0x3F.</p> <p><b>Note1:</b> User can only execute the reload WWDT counter value command when current CNTDAT (WWDT_CNT[5:0]) is between 1 and CMPDAT (WWDT_CTL[21:16]). If user writes 0x00005AA5 in WWDT_RLDCNT register when current CNTDAT is larger than CMPDAT, WWDT reset system event will be generated immediately.</p> <p><b>Note2:</b> Executing WWDT counter reload always needs (WWDT_CLK *3) period to reload CNTDAT to 0x3F and internal prescale counter will be reset also.</p>



**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]	<p><b>ICEDEBUG</b></p> <p><b>ICE Debug Mode Acknowledge Disable Bit</b>                      0 = ICE debug mode acknowledgement effects WWDT counter counting.                      The WWDT down counter will be held while CPU is held by ICE.                      1 = ICE debug mode acknowledgement Disabled.                      The WWDT down counter will keep counting no matter CPU is held by ICE or not.</p>
[30:22]	<p><b>Reserved</b></p> <p>Reserved.</p>
[21:16]	<p><b>CMPDAT</b></p> <p><b>WWDT Window Compare Value</b>                      Set this field to adjust the valid reload window interval when WWDTIF (WWDT_STATUS[0]) is generated.  <b>Note:</b> User can only write WWDT_RLDCNT register to reload WWDT counter value when current CNTDAT (WWDT_CNT[5:0]) is between 1 and CMPDAT. If user writes 0x00005AA5 in WWDT_RLDCNT register when current CNTDAT is larger than CMPDAT, WWDT reset system event will be generated immediately.</p>
[15:12]	<p><b>Reserved</b></p> <p>Reserved.</p>
[11:8]	<p><b>PSCSEL</b></p> <p><b>WWDT Counter Prescale Period Selection</b>                      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.</p>

		1100 = Pre-scale is 768; Max time-out period is $768 * 64 * \text{WWDT\_CLK}$ . 1101 = Pre-scale is 1024; Max time-out period is $1024 * 64 * \text{WWDT\_CLK}$ . 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]	<b>Reserved</b>	Reserved.
[1]	<b>INTEN</b>	<b>WWDT Interrupt Enable Bit</b> If this bit is enabled, when WWDTIF (WWDT_STATUS[0]) is set to 1, 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]	<b>WWDTEN</b>	<b>WWDT Enable Bit</b> Set this bit to start WWDT counter counting. 0 = WWDT counter is stopped. 1 = WWDT counter is starting 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	<p><b>WWDT Timer-out Reset System Flag</b></p> <p>If this bit is set to 1, it indicates that system has been reset by WWDT counter time-out reset system event.</p> <p>0 = WWDT time-out reset system event did not occur.</p> <p>1 = WWDT time-out reset system event occurred.</p> <p><b>Note:</b> This bit is cleared by writing 1 to it.</p>
[0]	WWDTIF	<p><b>WWDT Compare Match Interrupt Flag</b></p> <p>This bit indicates that current CNTDAT (WWDT_CNT[5:0]) matches the CMPDAT (WWDT_CTL[21:16]).</p> <p>0 = No effect.</p> <p>1 = WWDT CNTDAT matches the CMPDAT.</p> <p><b>Note:</b> This bit is cleared by writing 1 to it.</p>

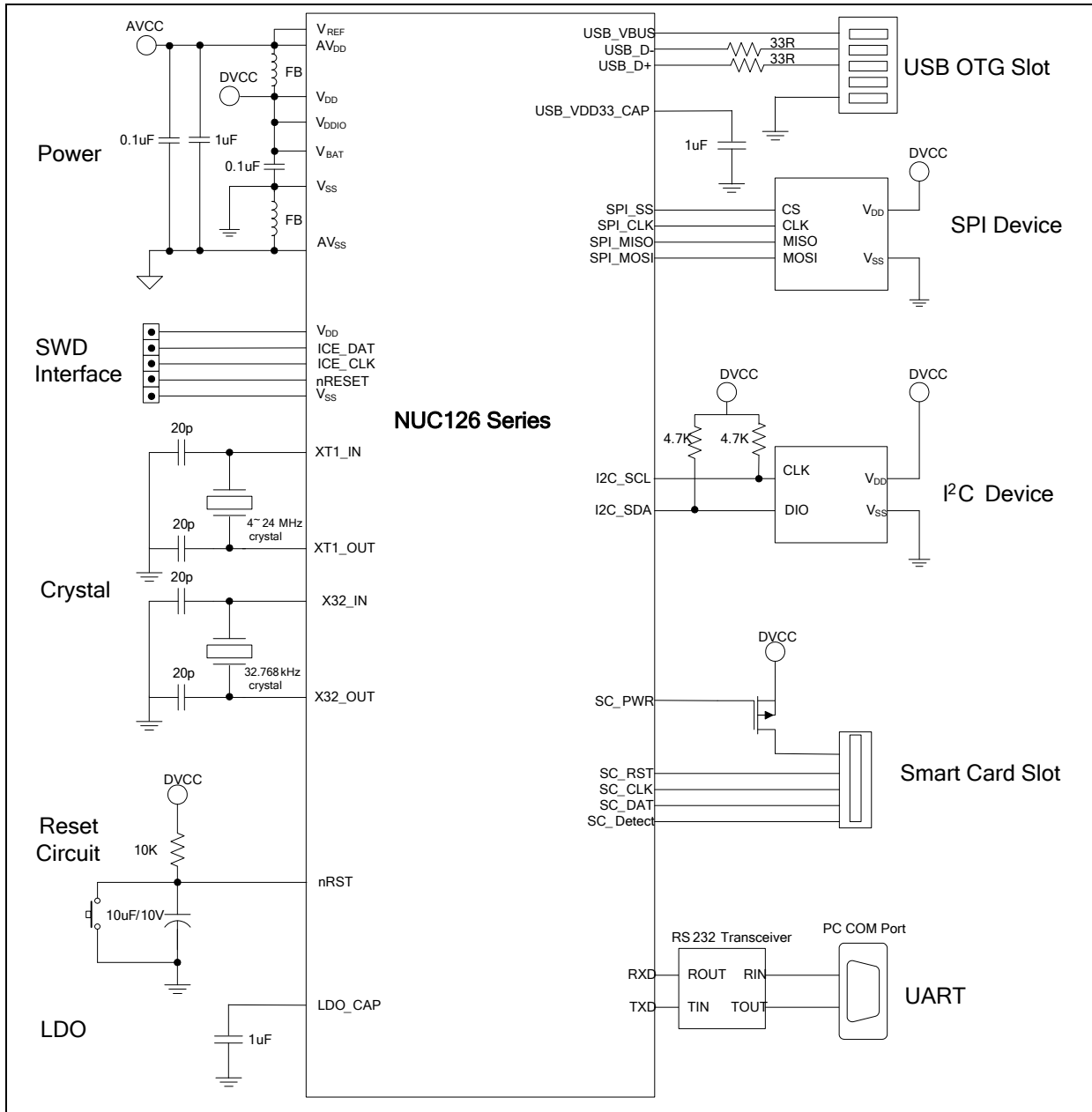
**WWDT Counter Value Register (WWDT\_CNT)**

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					

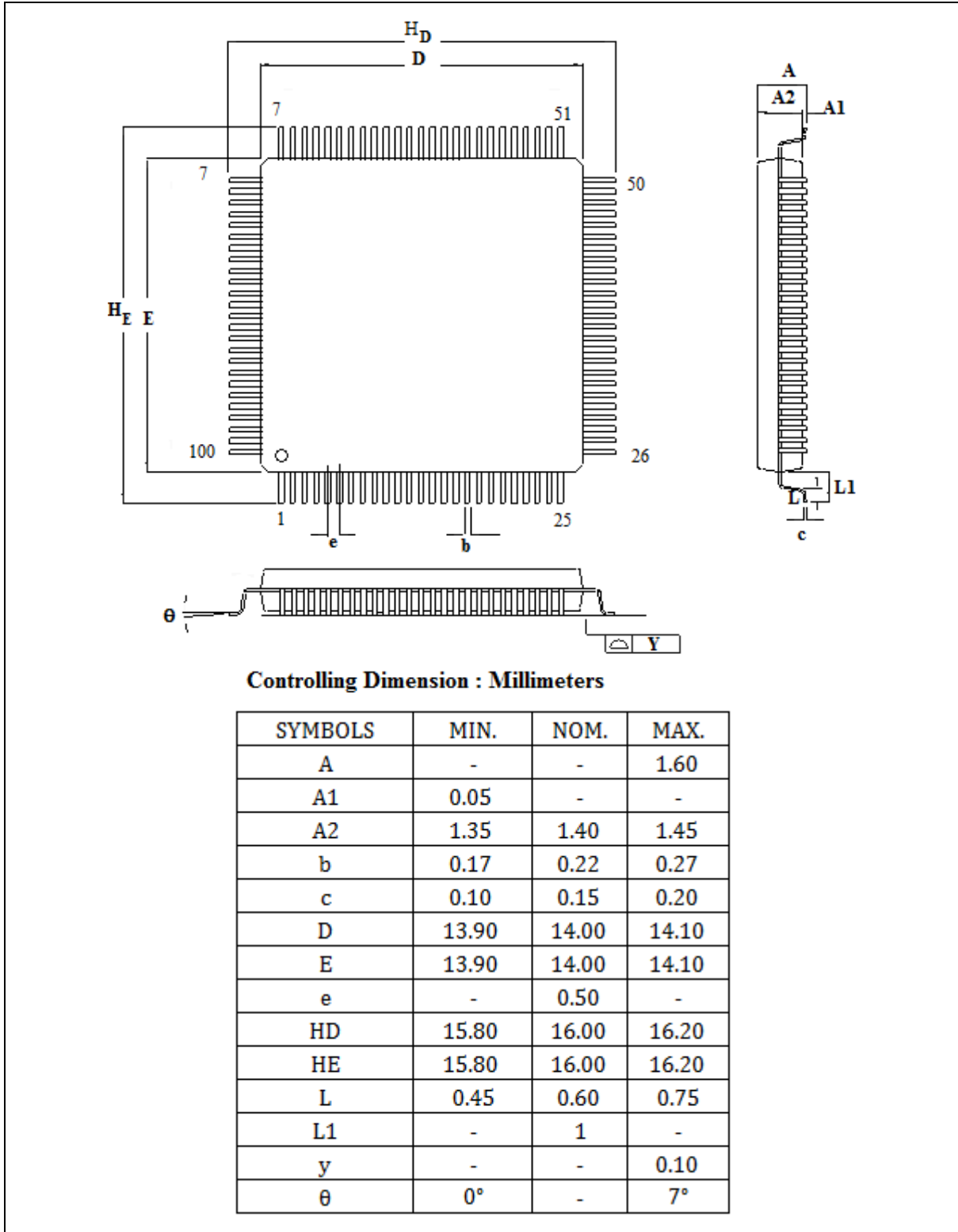
Bits	Description	
[31:6]	Reserved	Reserved.
[5:0]	CNTDAT	<b>WWDT Counter Value</b> CNTDAT will be updated continuously.

7 APPLICATION CIRCUIT

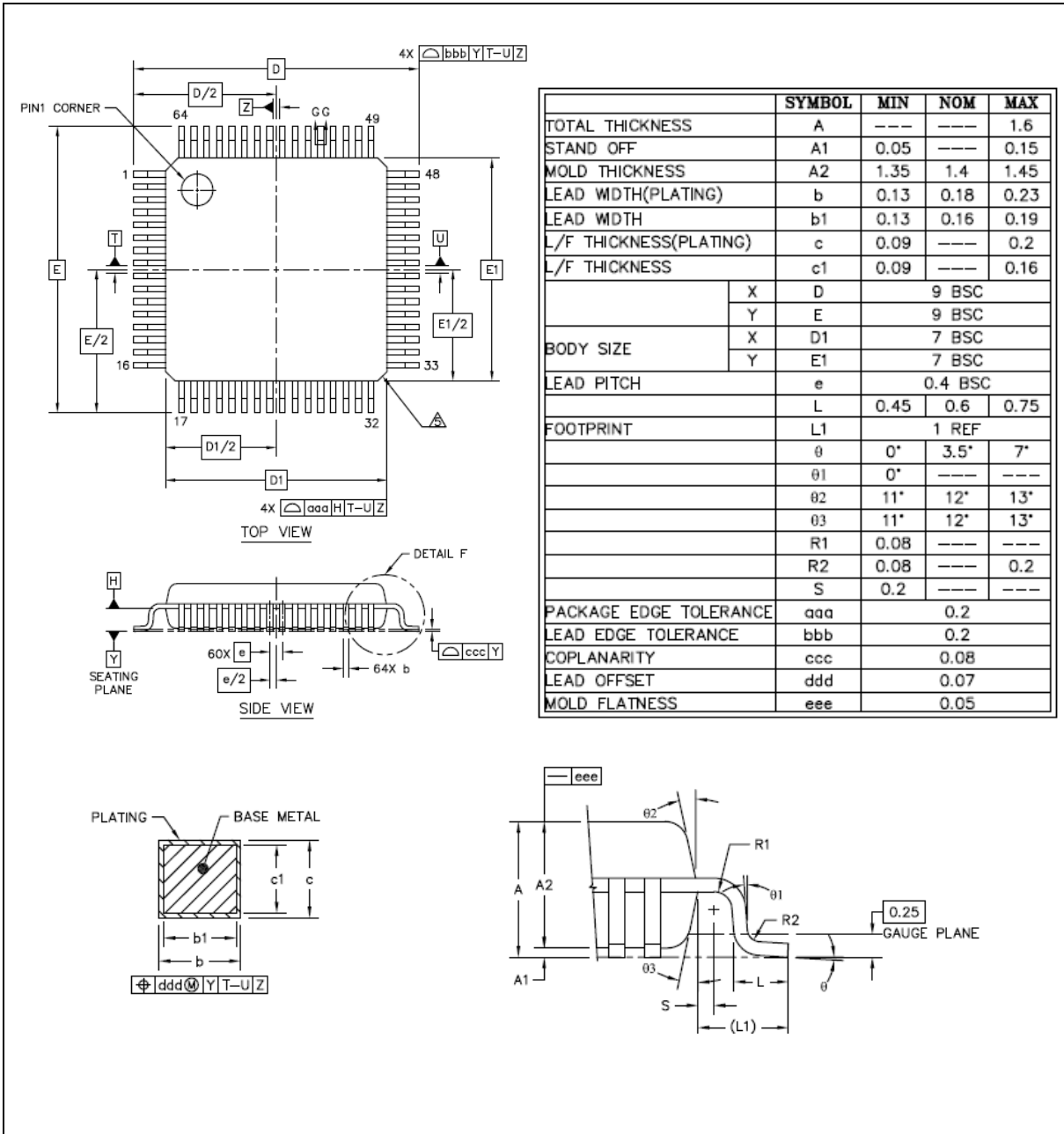


8 PACKAGE DIMENSIONS

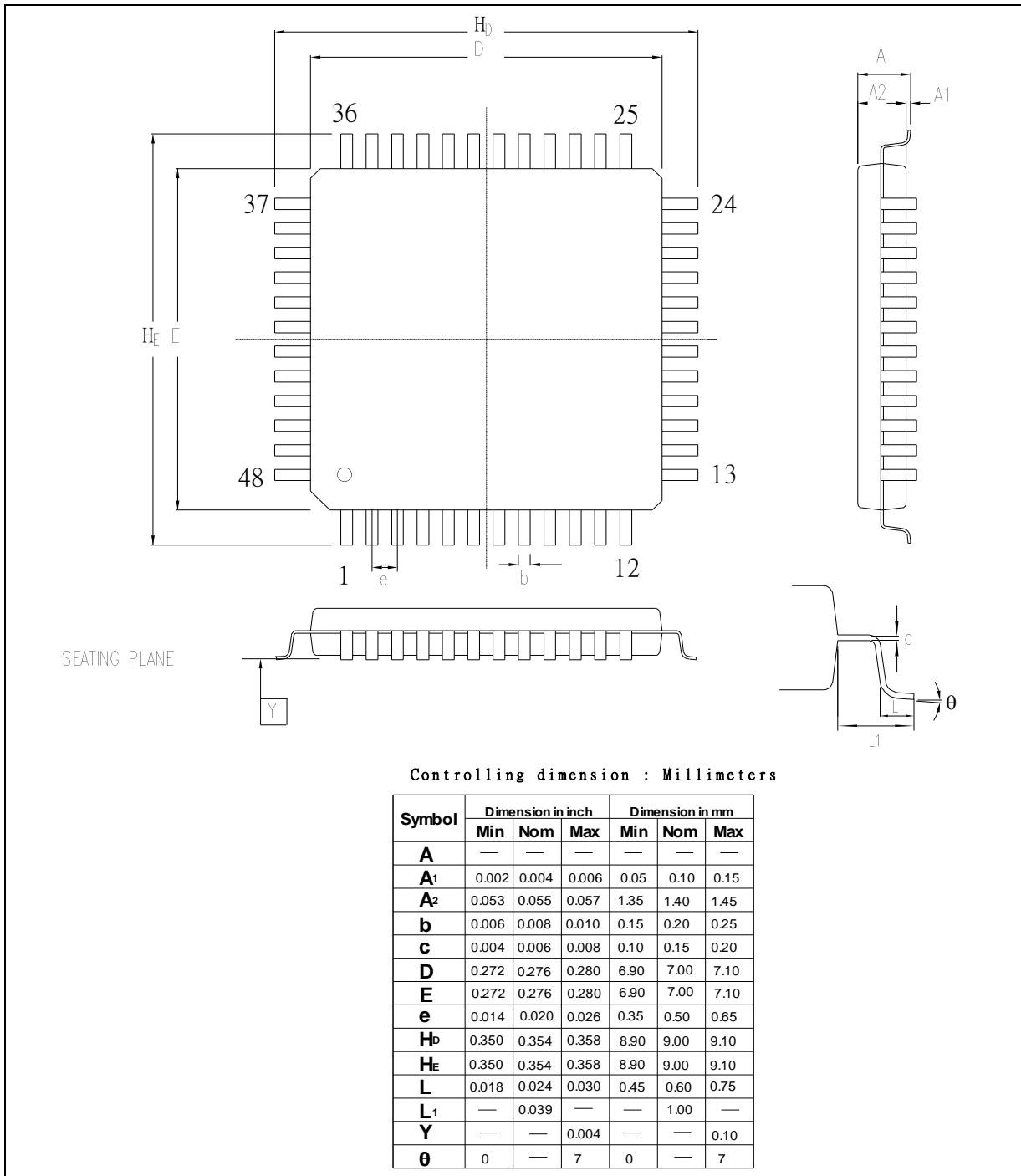
8.1 LQFP 100L (14x14x1.4 mm footprint 2.0 mm)



8.2 LQFP 64L (7x7x1.4 mm footprint 2.0 mm)

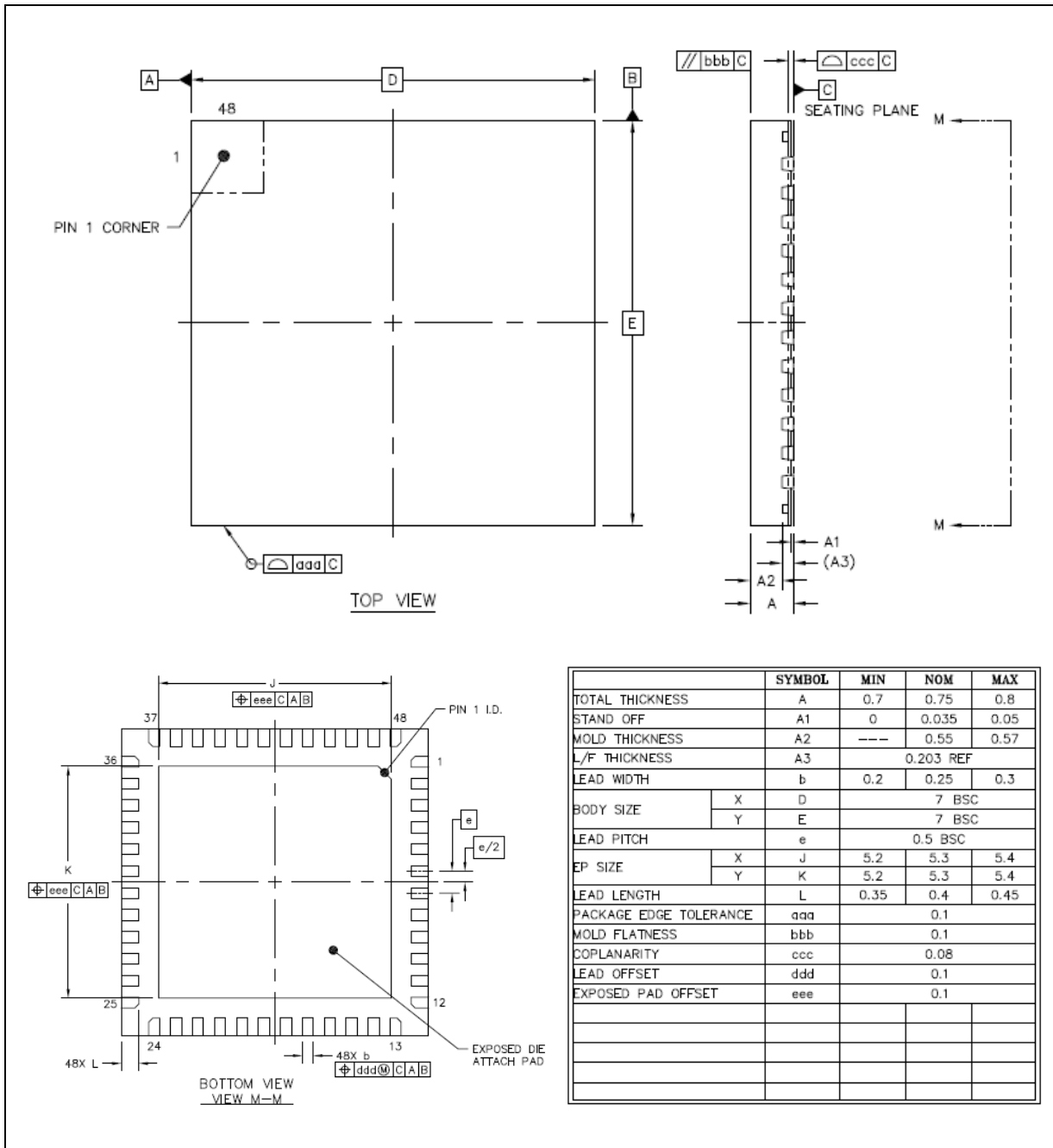


8.3 LQFP 48L (7x7x1.4 mm Footprint 2.0 mm)





8.4 QFN 48L (7x7x0.8 mm)



**9 REVISION HISTORY**

Date	Revision	Description
2016.10.28	1.00	Preliminary version
2017.07.25	1.01	1. Revised part number in section 4.1.2. 2. Updated Basic Configuration sections in Chapter 6.
2017.12.15	1.02	1. Revised HIRC trim description in section 6.2.8 and 6.2.12 2. Revised Clock Output description in section 6.3.5
2018.08.08	1.03	1. Revised $V_{DDIO}$ description in section 1.1 and 4.1.2. 2. Revised Timer/PWM PWM mode description in section 2.1. 3. Revised $V_{BAT}$ pin description in section 4.1.2. 4. Added NUC126 QFN48 information in section 2.1, 4.1, 4.2 and 8.4.

### 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.*