ARM® Cortex®-M0

32-bit Microcontroller

NuMicro® Family
Mini58DE 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

Table of Contents

1 GENERAL DESCRIPTION 11
2 FEATURES 12
3 ABBREVIATIONS 15
4 PARTS INFORMATION LIST AND PIN CONFIGURATION 16
4.1 NuMicro® Mini58 Series Naming Rule 16
4.2 NuMicro® Mini58 Series Product Selection Guide 17
4.3 PIN CONFIGURATION 18
4.3.1 LQFP 48-pin 18
4.3.2 QFN 33-pin 20
4.3.3 TSSOP 20-pin 22
4.4 Pin Description 23
5 BLOCK DIAGRAM 28
5.1 NuMicro® Mini58 Block Diagram 28
6 Functional Description 29
6.1 ARM® Cortex®-M0 Core 29
6.1.1 Overview 29
6.1.2 Features 29
6.2 System Manager 31
6.2.1 Overview 31
6.2.2 System Reset 31
6.2.3 Power Modes and Wake-up Sources 37
6.2.4 System Power Architecture 39
6.2.5 System Memory Mapping 41
6.2.6 Register Protection 41
6.2.7 Memory Organization 43
6.2.8 Register Map 45
6.2.9 Register Description 46
6.2.10 System Timer (SysTick) 69
6.2.11 Nested Vectored Interrupt Controller (NVIC) 74
6.2.12 System Control Registers (SCB) 114
6.3 Clock Controller 123
6.3.1 Overview 123
6.3.2 Auto-trim 125

6.3.3 System Clock and SysTick Clock 125
6.3.4 Peripherals Clock Source Selection 126
6.3.5 Power-down Mode Clock 128
6.3.6 Frequency Divider Output 128
6.3.7 Register Map 130
6.3.8 Register Description 131
6.4 Flash Memory Controller (FMC) 146
6.4.1 Overview 146
6.4.2 Features 146
6.4.3 Block Diagram 146
6.4.4 Functional Description 148
6.4.5 Flash Control Register Map 164
6.4.6 Flash Control Register Description 165
6.5 General Purpose I/O (GPIO) 175
6.5.1 Overview 175
6.5.2 Features 175
6.5.3 Block Diagram 176
6.5.4 Basic Configuration 176
6.5.5 Functional Description 177
6.5.6 GPIO Interrupt and Wake-up Function 178
6.5.7 Register Map 179
6.5.8 Register Description 182
6.6 Timer Controller (TMR) 195
6.6.1 Overview 195
6.6.2 Features 195
6.6.3 Block Diagram 195
6.6.4 Basic Configuration 196
6.6.5 Functional Description 197
6.6.6 Register Map 202
6.6.7 Register Description 203
6.7 Enhanced PWM Generator 212
6.7.1 Overview 212
6.7.2 Features 212
6.7.3 Block Diagram 215
6.7.4 Basic Configuration 216
6.7.5 Functional Description 217

6.7.6 Register Map 235
6.7.7 Register Description 237
6.8 Watchdog Timer (WDT) 272
6.8.1 Overview 272
6.8.2 Features 272
6.8.3 Block Diagram 272
6.8.4 Clock Control 272
6.8.5 Basic Configuration 273
6.8.6 Functional Description 273
6.8.7 Registers Map 275
6.8.8 Registers Description 276
6.9 Window Watchdog Timer (WWDT) 279
6.9.1 Overview 279
6.9.2 Features 279
6.9.3 Block Diagram 279
6.9.4 Clock Control 280
6.9.5 Basic Configuration 280
6.9.6 Functional Description 280
6.9.7 Register Map 283
6.9.8 Register Description 284
6.10 UART Controller (UART) 289
6.10.1 Overview 289
6.10.2 Features 289
6.10.3 Block Diagram 290
6.10.4 Basic Configuration 293
6.10.5 Functional Description 293
6.10.6 Register Map 304
6.10.7 Register Description 305
6.11 I^2C Serial Interface Controller (I^2C) 325
6.11.1 Overview 325
6.11.2 Features 325
6.11.3 Basic Configuration 325
6.11.4 Block Diagram 326
6.11.5 Functional Description 326
6.11.6 Register Map 343
6.11.7 Register Description 344

6.12 Serial Peripheral Interface (SPI) 355
6.12.1 Overview 355
6.12.2 Features 355
6.12.3 Block Diagram 355
6.12.4 Basic Configuration 356
6.12.5 Functional Description 356
6.12.6 Timing Diagram 362
6.12.7 Programming Examples 364
6.12.8 Register Map 367
6.12.9 Register Description 368
6.13 Analog-to-Digital Converter (ADC) 382
6.13.1 Overview 382
6.13.2 Features 382
6.13.3 Block Diagram 383
6.13.4 Basic Configuration 383
6.13.5 Functional Description 384
6.13.6 Register Map 390
6.13.7 Register Description 391
6.14 Analog Comparator (ACMP) 405
6.14.1 Overview 405
6.14.2 Features 405
6.14.3 Block Diagram 406
6.14.4 Basic Configuration 406
6.14.5 Functional Description 407
6.14.6 Comparator Reference Voltage (CRV) 408
6.14.7 Register Map 410
6.14.8 Register Description 411
7 APPLICATION CIRCUIT 417
8 ELECTRICAL CHARACTERISTICS 418
9 PACKAGE DIMENSIONS 419
9.1 48-pin LQFP 419
9.2 33-pin QFN (4 mm x 4 mm) 420
9.3 33-pin QFN (5 mm x 5 mm) 421
9.4 20-pin TSSOP 422
10 REVISION HISTORY 423

List of Figures

Figure 4.1-1 NuMicro® Mini58 Series Naming Rule 16
Figure 4.3-1 NuMicro® Mini58 Series LQFP 48-pin Diagram 18
Figure 4.3-2 NuMicro® Mini58 Series LQFP 48-pin Multi-Function Diagram 19
Figure 4.3-3 NuMicro® Mini58 Series QFN 33-pin Diagram 20
Figure 4.3-4 NuMicro® Mini58 Series QFN 33-pin Multi-funciton Diagram 21
Figure 4.3-5 NuMicro® Mini58 Series TSSOP 20-pin Diagram 22
Figure 4.3-6 NuMicro® Mini58 Series TSSOP 20-pin Multi-function Diagram 22
Figure 5.1-1 NuMicro® Mini58 Series Block Diagram 28
Figure 6.1-1 Functional Block Diagram 29
Figure 6.2-1 System Rese Resources 32
Figure 6.2-2 nRESET Reset Waveform 34
Figure 6.2-3 Power-on Reset (POR) Waveform 34
Figure 6.2-4 Low Voltage Reset (LVR) Waveform 35
Figure 6.2-5 Brown-out Detector (BOD) Waveform 36
Figure 6.2-6 Power Mode State Machine 37
Figure 6.2-7 NuMicro® Mini58 Series Power Architecture Diagram 40
Figure 6.3-1 Clock Generator Block Diagram 123
Figure 6.3-2 Clock Generator Global View Diagram 124
Figure 6.3-3 System Clock Block Diagram 125
Figure 6.3-4 SysTick Clock Control Block Diagram 126
Figure 6.3-5 Peripherals Bus Clock Source Selection for PCLK 127
Figure 6.3-6 Clock Source of Frequency Divider 129
Figure 6.3-7 Block Diagram of Frequency Divider 129
Figure 6.4-1 Flash Memory Control Block Diagram 147
Figure 6.4-2 Data Flash Shared with APROM 149
Figure 6.4-3 SPROM Security Mode 154
Figure 6.4-4 Flash Memory Map 155
Figure 6.4-5 System Memory Map with IAP Mode 156
Figure 6.4-6 LDROM with IAP Mode 157
Figure 6.4-7 APROM with IAP Mode 157
Figure 6.4-8 System Memory Map without IAP Mode 158
Figure 6.4-9 Boot Source Selection 159
Figure 6.4-25 ISP Procedure Example 161
Figure 6.4-26 CRC-32 Checksum Calculation 162
Figure 6.4-27 CRC-32 Checksum Calculation Flow 163
Figure 6.5-1 GPIO Controller Block Diagram 176

Figure 6.6-1 Timer Controller Block Diagram 196
Figure 6.6-2 Clock Source of Timer Controller 196
Figure 6.6-3 Continuous Counting Mode 198
Figure 6.6-4 Free-Counting Capture Mode 199
Figure 6.6-5 External Reset Counter Mode 200
Figure 6.7-1 Application Circuit Diagram 214
Figure 6.7-2 PWM Block Diagram 215
Figure 6.7-3 PWM Generator 0 Architecture Diagram 215
Figure 6.7-4 PWM Generator 2 Architecture Diagram 216
Figure 6.7-5 PWM Generator 4 Architecture Diagram 216
Figure 6.7-6 Edge-aligned Type PWM 218
Figure 6.7-7 PWM Edge-aligned Waveform Timing Diagram 218
Figure 6.7-8 Edge-aligned Flow Diagram 219
Figure 6.7-9 Legend of Internal Comparator Output of PWM Counter 220
Figure 6.7-10 PWM Counter Operation Timing 220
Figure 6.7-11 Center-aligned Type PWM 221
Figure 6.7-12 Center-aligned Type Operation Timing 222
Figure 6.7-13 PWM Center-aligned Waveform Timing Diagram 222
Figure 6.7-14 Center-aligned Flow Diagram 223
Figure 6.7-15 Precise Center-aligned Type PWM 224
Figure 6.7-16 PWM Precise Center-aligned Waveform Timing Diagram 225
Figure 6.7-17 Precise Center-aligned Flow Diagram 226
Figure 6.7-18 PWM Center Loading Timing Diagram 227
Figure 6.7-19 PWM Double Buffering Illustration 228
Figure 6.7-20 PWM Controller Output Duty Ratio 228
Figure 6.7-21 Dead-time Insertion 229
Figure 6.7-22 Asymmetric Mode Timing Diagram 230
Figure 6.7-23 Initial State and Polarity Control with Rising Edge Dead-time Insertion 231
Figure 6.7-24 Motor Control PWM Interrupt Architecture 232
Figure 6.7-25 Mask and Mask-aligned Function 233
Figure 6.8-1 Watchdog Timer Block Diagram 272
Figure 6.8-2 Watchdog Timer Clock Control 273
Figure 6.8-3 Watchdog Timer Time-out Interval and Reset Period Timing 274
Figure 6.9-1 WWDT Block Diagram 279
Figure 6.9-2 WWDT Clock Control 280
Figure 6.9-3 WWDT Reset and Reload Behavior 282
Figure 6.10-1 UART0 Controller Clock Control 290

Figure 6.10-2 UART1 Controller Clock Control 290
Figure 6.10-3 UART0 Controller Block Diagram 291
Figure 6.10-4 UART1 Controller Block Diagram 291
Figure 6.10-5 Auto Flow Control Block Diagram 293
Figure 6.10-6 Controller Baud Rate Equation Table 294
Figure 6.10-7 Controller Baud Rate Parameter Setting Table 294
Figure 6.10-8 Controller Baud Rate Register Setting Table 295
Figure 6.10-9 Controller Interrupt Source and Flag 296
Figure 6.10-10 Line Control of Word and Stop Length Setting 297
Figure 6.10-11 Line Control of Parity Bit Setting 297
Figure 6.10-12 CTS Auto Flow Control Enabled 298
Figure 6.10-13 RTS Auto Flow Control Enabled 298
Figure 6.10-14 RTS Flow with Software Control 299
Figure 6.10-15 IrDA Control Block Diagram 299
Figure 6.10-16 IrDA TX/RX Timing Diagram 300
Figure 6.10-17 RS-485 RTS Driving Level in Auto Direction Mode 302
Figure 6.10-18 RS-485 RTS Driving Level with Software Control 302
Figure 6.10-19 Structure of RS-485 Frame 303
Figure 6.11-1 I^2C Controller Block Diagram 326
Figure 6.11-2 I^2C Bus Timing 326
Figure 6.11-3 I^2C Protocol 327
Figure 6.11-4 START and STOP Condition 328
Figure 6.11-5 Bit Transfer on I^2C Bus 328
Figure 6.11-6 Acknowledge on I^2C Bus 329
Figure 6.11-7 Master Transmits Data to Slave 329
Figure 6.11-8 Master Reads Data from Slave 329
Figure 6.11-9 Timing of Two-level Buffer Transmission in Master Write 330
Figure 6.11-10 Timing of Two-level Buffer Transmission in Slave Read 330
Figure 6.11-11 Control I^2C Bus according to Current I^2C Status 331
Figure 6.11-12 Master Transmitter Mode Control Flow 332
Figure 6.11-13 Master Receiver Mode Control Flow 333
Figure 6.11-14 Slave Mode Control Flow 334
Figure 6.11-15 GC Mode 336
Figure 6.11-16 I^2C Data Shifting Direction 338
Figure 6.11-17 I^2C Time-out Count Block Diagram 340
Figure 6.11-18 EEPROM Random Read 341
Figure 6.11-19 Protocol of EEPROM Random Read 341

Figure 6.12-1 SPI Block Diagram 355
Figure 6.12-2 SPI Master Mode Application Block Diagram 356
Figure 6.12-3 SPI Slave Mode Application Block Diagram 357
Figure 6.12-4 32-Bit in One Transaction 357
Figure 6.12-5 Byte Reorder 359
Figure 6.12-6 Timing Waveform for Byte Suspend 360
Figure 6.12-7 FIFO Mode Block Diagram 361
Figure 6.12-8 SPI Timing in Master Mode 363
Figure 6.12-9 SPI Timing in Master Mode (Alternate Phase of SPI_CLK) 363
Figure 6.12-10 SPI Timing in Slave Mode 364
Figure 6.12-11 SPI Timing in Slave Mode (Alternate Phase of SPI_CLK) 364
Figure 6.13-1 AD Controller Block Diagram 383
Figure 6.13-2 ADC Peripheral Clock Control 384
Figure 6.13-3 Single Mode Conversion Timing Diagram 385
Figure 6.13-4 V_{BG} for Measuring AV_{DD} Application Block Diagram 386
Figure 6.13-5 ADC Start Conversion Conditions 387
Figure 6.13-6 A/D Conversion Result Monitor Logics Diagram 387
Figure 6.13-7 A/D Controller Interrupt 388
Figure 6.13-8 Conversion Result Mapping Diagram of ADC Single-end Input 388
Figure 6.13-9 ADC Sequential Mode Type is 2/3-shunt 389
Figure 6.13-10 ADC Sequential Mode Type is 1-shunt 389
Figure 6.14-1 Analog Comparator Block Diagram 406
Figure 6.14-2 Analog Comparator Controller Interrupt Sources 407
Figure 6.14-3 Comparator Hysteresis Function 408
Figure 6.14-4 Comparator Reference Voltage Block Diagram 409

List of Tables

Table 3-1 List of Abbreviations 15
Table 4.2-1 NuMicro® Mini58 Series Product Selection Guide 17
Table 4.4-1 NuMicro® Mini58 Series Pin Description 27
Table 6.2-1 Reset Value of Registers 33
Table 6.2-2 Power Mode Difference Table 37
Table 6.2-3 Clocks in Power Modes 38
Table 6.2-4 Condition of Entering Power-down Mode Again 39
Table 6.2-5 Memory Mapping Table 41
Table 6.2-6 Address Space Assignments for On-Chip Modules 44
Table 6.2-7 Exception Model 75
Table 6.2-8 System Interrupt Map Vector Table 76
Table 6.2-9 Vector Table Format 76
Table 6.3-1 Peripheral Clock Source Selection Table 128
Table 6.3-2 Power-down Mode Control 133
Table 6.6-1 Input Capture Mode Operation 201
Table 6.8-1 Watchdog Timer Time-out Interval Period Selection 274
Table 6.9-1 WWDT Prescaler Value Selection 281
Table 6.9-2 CMPDAT Setting Limitation 282
Table 6.11-1 I^2C Status Code Description 339

1 GENERAL DESCRIPTION

The NuMicro® Mini58 series is pin-to-pin and function compatible with the NuMicro® Mini51 series, the 32-bit microcontroller (MCU) embedded with the ARM® Cortex®-M0 core. The Mini58 series can bridge the gap and replace the cost equivalent to traditional 8- and 16-bit microcontroller by 32-bit performance and rich functions. The Mini58 series supports a wide range of applications from low-end, price sensitive designs to computing-intensive ones and provides advanced high-end features in economical products.

The Mini58 series can run up to 50 MHz which is faster than 24 MHz in Mini51 series, and operate at a wide voltage range of 2.5V ~ 5.5V and temperature range of -40℃ ~ +105℃. For the Mini58 series, the embedded program flash size upgrades from 16 Kbytes to 32 Kbytes and SRAM upgrades from 2 Kbytes to 4 Kbytes. The Mini58 series also offers size configurable Data Flash (shared with program flash), and 2.5 Kbytes flash for the ISP.

The Mini58 series has many high-performance peripheral functions, such as 22.1184 MHz internal RC oscillator (±1% accuracy), I/O port with up to 30 pins, four 32-bit timers, two UARTs with the RS485 function and IrDA function interface, one SPI interface, two I^2C interfaces, up to three 16-bit PWM generators providing six channels, an 8-channel 10-bit ADC, Watchdog Timer, Window Watchdog Timer, two Analog Comparators and a Brown-out Detector. All these peripherals have been incorporated into the Mini58 series to reduce component count, board space and system cost. Compared to the Mini51 series, the Mini58 series supports additional one UART and one I^2C interface for better and more flexible connectivity applications.

Additionally, the Mini58 series is equipped with ISP (In-System Programming) and ICP (In-Circuit Programming) functions, which allow the user to update the program memory without removing the chip from the actual end product. The Mini58 series also supports In-Application-Programming (IAP) function, user switches the code executing without the chip reset after the embedded flash updated.

2 FEATURES

- Core
  - ARM® Cortex®-M0 core running up to 50 MHz
  - One 24-bit system timer
  - Supports low power Idle mode
  - A single-cycle 32-bit hardware multiplier
  - NVIC for the 32 interrupt inputs, each with 4-level of priority
  - Supports Serial Wire Debug (SWD) interface and two watchpoints/four breakpoints
- Built-in LDO for wide operating voltage: 2.5V to 5.5V
- Memory
  - 32 KB Flash memory for program memory (APROM)
  - Configurable Flash memory for data memory (Data Flash)
  - 2.5 KB Flash for loader (LDROM)
  - 4 KB SRAM for internal scratch-pad RAM (SRAM)
- Clock Control
  - Programmable system clock source
    - Switch clock sources on-the-fly
  - Support 4 ~ 24 MHz external high speed crystal oscillator (HXT) for precise timing operation
  - Support 32.768 kHz external low speed crystal oscillator (LXT) for idle wake-up and system operation clock
  - Built-in 22.1184 MHz internal high speed RC oscillator (HIRC) for system operation (1% accuracy at 25^0C, 5V)
    - Dynamically calibrating the HIRC OSC to 22.1184 MHz ±1% from -40℃ to 105℃ by external 32.768K crystal oscillator (LXT)
  - Built-in 10 kHz internal low speed RC oscillator (LIRC) for Watchdog Timer and wake-up operation
  - PLL allowing CPU operation up to the maximum 50 MHz
- I/O Port
  - Up to 30 general-purpose I/O (GPIO) pins for LQFP-48 package
  - Four I/O modes:
    - Quasi-bidirectional input/output
    - Push-Pull output
    - Open-Drain output
    - Input only with high impendence
  - Optional Schmitt trigger input
- Timer
  - Provides two channel 32-bit Timers; one 8-bit pre-scaler counter with 24-bit up-timer for each timer

- Supports Event Counter mode
- Supports Toggle Output mode
- Supports external trigger in Pulse Width Measurement mode
- Supports external trigger in Pulse Width Capture mode

- WDT (Watchdog Timer)
  - Programmable clock source and time-out period
  - Supports wake-up function in Power-down mode and Idle mode
  - Interrupt or reset selectable on watchdog time-out
- WWDT (Window Watchdog Timer)
  - 6-bit down counter value (CNTDAT) and 6-bit compare value (CMPDAT) to make the WWDT time-out window period flexible
  - Supports 4-bit value (PSCSEL) to programmable maximum 11-bit prescale counter period of WWDT counter
- PWM
  - Up to three built-in 16-bit PWM generators, providing six PWM outputs or three complementary paired PWM outputs
  - Individual clock source, clock divider, 8-bit pre-scalar and dead-time generator for each PWM generator
  - PWM interrupt synchronized to PWM period
  - Supports edge-alignment or center-alignment
  - Supports fault detection
- UART (Universal Asynchronous Receiver/Transmitters)
  - Two UART devices
  - Buffered receiver and transmitter, each with 16-byte FIFO
  - Optional flow control function (CTSn and RTSn)
  - Supports IrDA (SIR) function
  - Programmable baud-rate generator up to 1/16 system clock
  - Supports RS-485 function
- SPI (Serial Peripheral Interface)
  - One SPI device
  - Master up to 25 MHz, and Slave up to 10 MHz
  - Supports Master/Slave mode
  - Full-duplex synchronous serial data transfer
  - Variable length of transfer data from 1 to 32 bits
  - MSB or LSB first data transfer
  - RX latching data can be either at rising edge or at falling edge of serial clock
  - TX sending data can be either at rising edge or at falling edge of serial clock
  - Supports Byte Suspend mode in 32-bit transmission
- I^2C

- Two I^2C devices
- Supports 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 allowing 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
- Programmable clocks allow for versatile rate control
- Supports multiple address recognition (four slave addresses with mask option)

- ADC (Analog-to-Digital Converter)
  - 10-bit SAR ADC with 250 kSPS
  - Up to 8-ch single-end input and one internal input from band-gap
  - Conversion started either by software trigger or external pin trigger
- Analog Comparator
  - Two analog comparators with programmable 16-level internal reference voltage
  - Built-in CRV (comparator reference voltage)
- ISP (In-System Programming), ICP (In-Circuit Programming), and IAP (In-Application-Programming) update
- BOD (Brown-out Detector)
  - With 4 programmable threshold levels: 4.4V/3.7V/2.7V/2.2V
  - Supports Brown-out interrupt and reset option
- 96-bit unique ID
- LVR (Low Voltage Reset)
  - Threshold voltage level: 2.0V
- Operating Temperature: -40℃~105℃
- Reliability: EFT > ± 4KV, ESD HBM pass 4KV
- Packages:
  - Green package (RoHS)
  - 48-pin LQFP (7x7), 33-pin QFN (5x5) , 33-pin QFN (4x4), 20-pin TSSOP

3 ABBREVIATIONS

Acronym	Description
ACMP	Analog Comparator Controller
ADC	Analog-to-Digital Converter
AHB	Advanced High-Performance Bus
APB	Advanced Peripheral Bus
BOD	Brown-out Detection
DAP	Debug Access Port
FIFO	First In, First Out
FMC	Flash Memory Controller
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
ICP	In Circuit Programming
ISP	In System Programming
ISR	Interrupt Service Routine
LDO	Low Dropout Regulator
LIRC	10 kHz internal low speed RC oscillator (LIRC)
LXT	32.768 kHz External Low Speed Crystal Oscillator
NVIC	Nested Vectored Interrupt Controller
PCLK	The Clock of Advanced Peripheral Bus
PLL	Phase-Locked Loop
PWM	Pulse Width Modulation
SPI	Serial Peripheral Interface
SPS	Samples per Second
TMR	Timer Controller
UART	Universal Asynchronous Receiver/Transmitter
UCID	Unique Customer ID
WDT	Watchdog Timer
WWDT	Window Watchdog Timer

Table 3-1 List of Abbreviations

4 PARTS INFORMATION LIST AND PIN CONFIGURATION

4.1 NuMicro® Mini58 Series Naming Rule

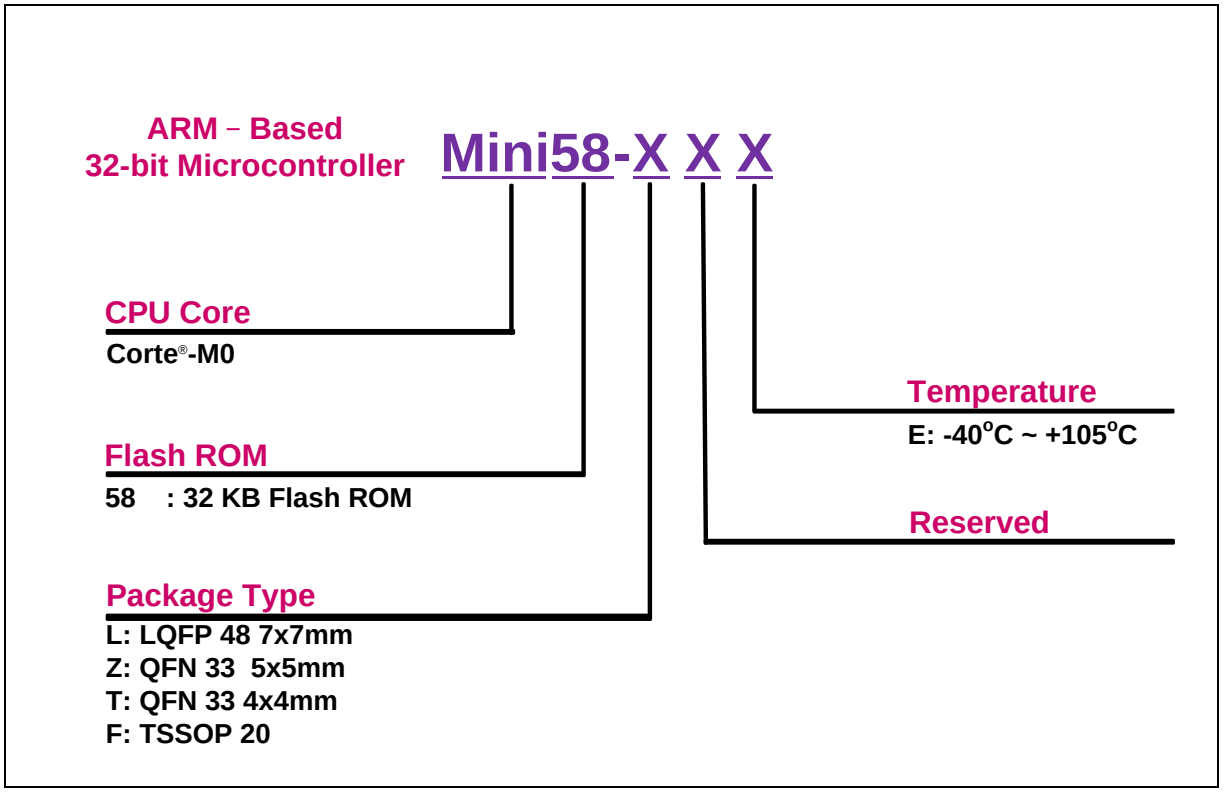


Figure 4.1-1 NuMicro® Mini58 Series Naming Rule

4.2 NuMicro® Mini58 Series Product Selection Guide

Part Number	APROM	RAM	Data Flash	ISP Loader ROM	I/O	Timer	Connectivity			Comp.	PWM	ADC	ISP ICP	IRC 22.1184 MHz	Package
							UART	SPI	I^2C						
MINI58LDE	32 KB	4 KB	Configurable	2.5 KB	up to 30	2x32-bit	2	1	2	2	6	8x10-bit	v	v	LQFP48
MINI58ZDE	32 KB	4 KB	Configurable	2.5 KB	up to 29	2x32-bit	2	1	2	2	6	8x10-bit	v	v	QFN33(5x5)
MINI58TDE	32 KB	4 KB	Configurable	2.5 KB	up to 29	2x32-bit	2	1	2	2	6	8x10-bit	v	v	QFN33(4x4)
MINI58FDE	32 KB	4 KB	Configurable	2.5 KB	up to 17	2x32-bit	2	1	2	-	6	4x10-bit	v	v	TSSOP20

Table 4.2-1 NuMicro® Mini58 Series Product Selection Guide

4.3 PIN CONFIGURATION

4.3.1 LQFP 48-pin

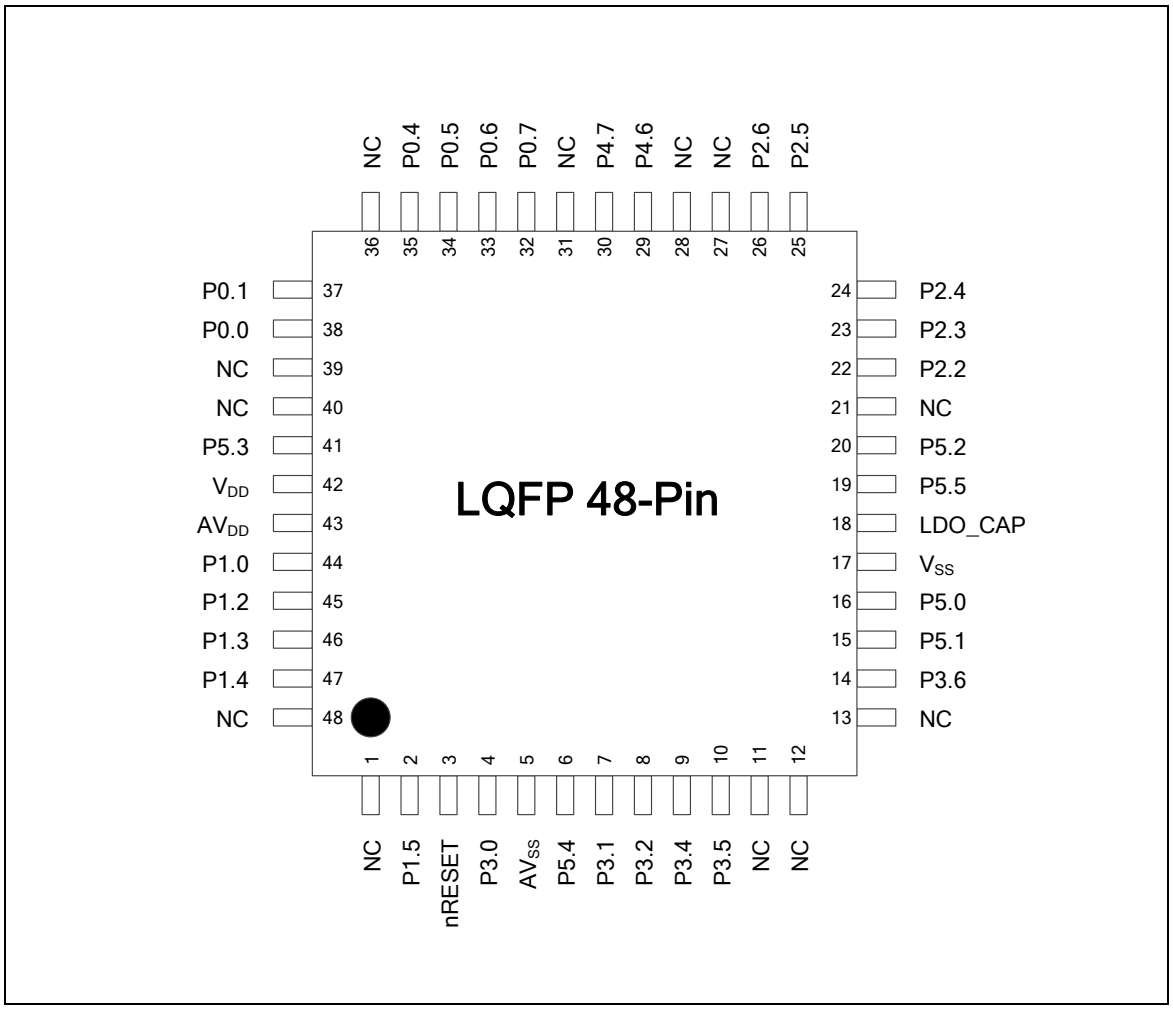


Figure 4.3-1 NuMicro® Mini58 Series LQFP 48-pin Diagram

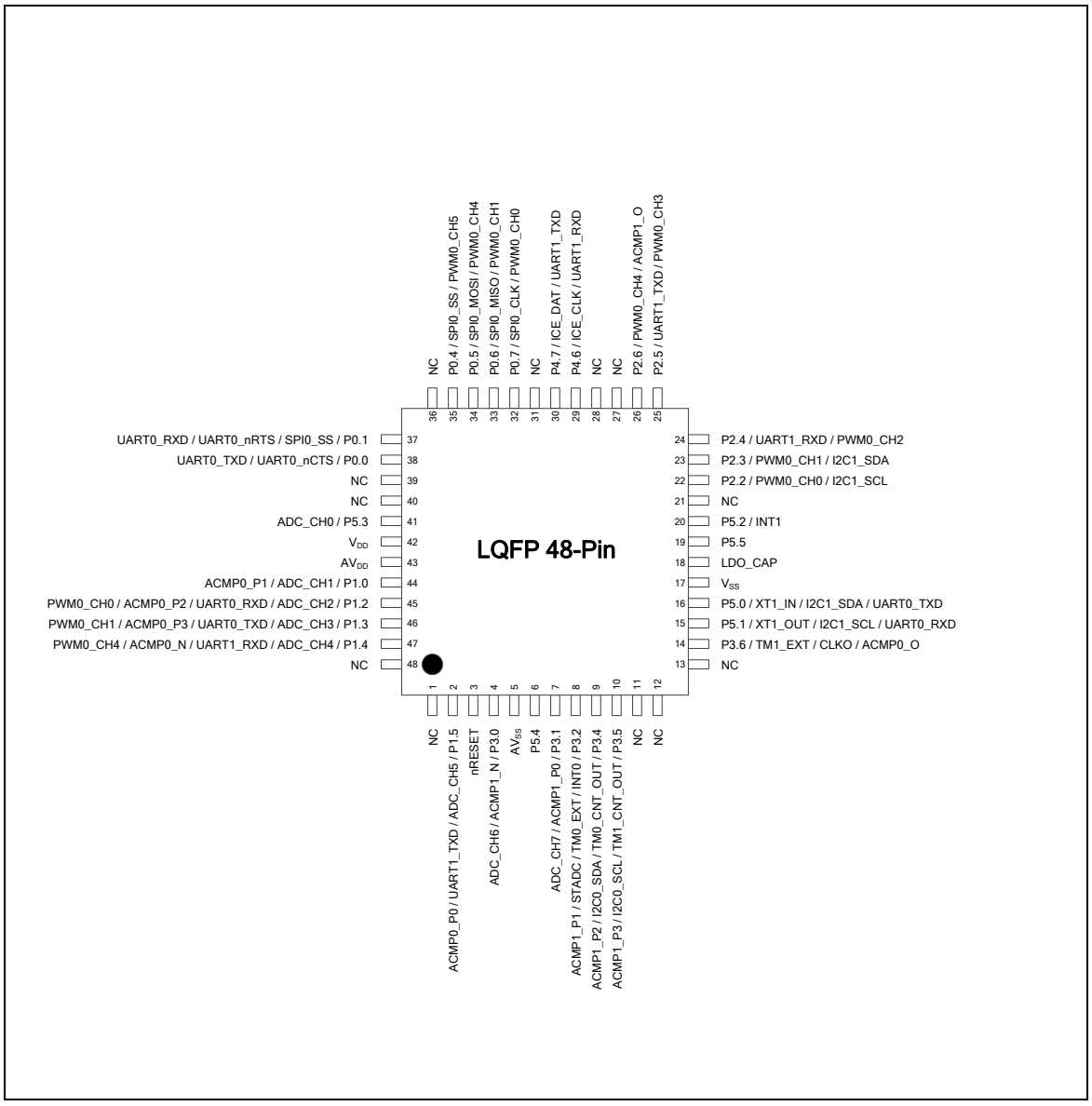


Figure 4.3-2 NuMicro® Mini58 Series LQFP 48-pin Multi-Function Diagram

4.3.2 QFN 33-pin

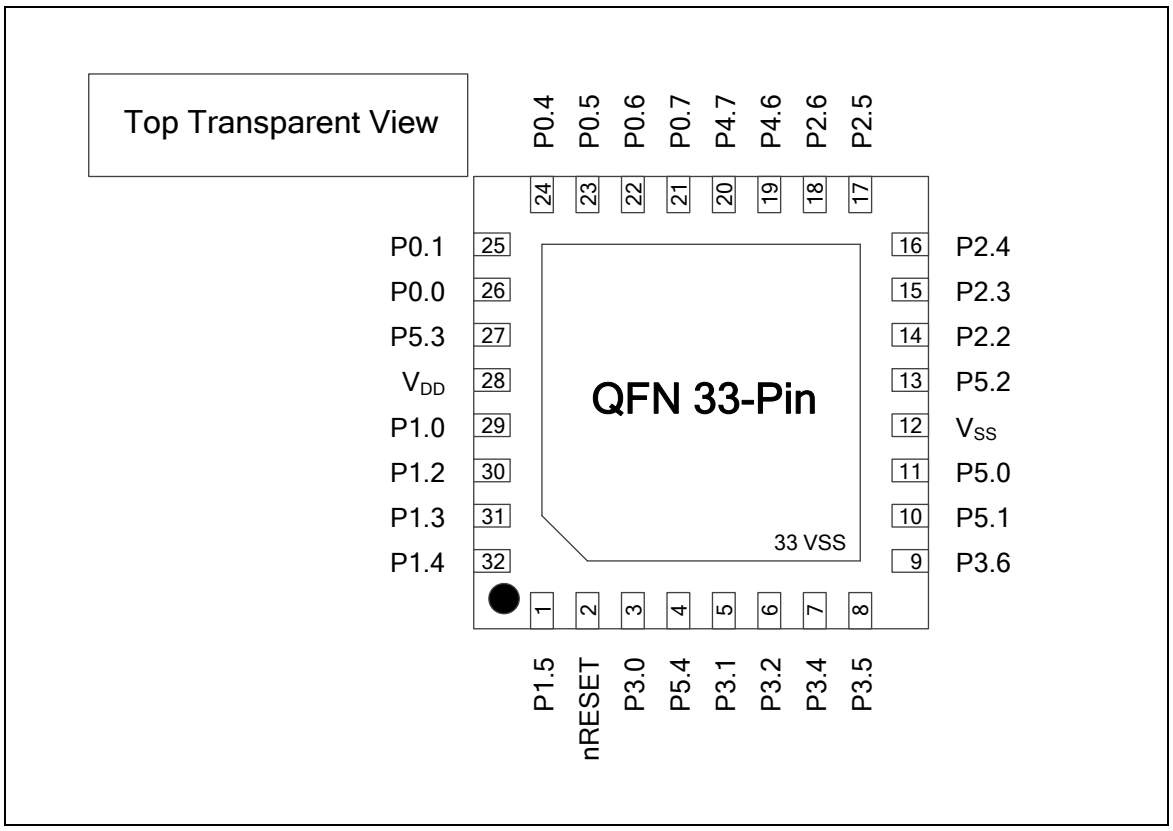


Figure 4.3-3 NuMicro® Mini58 Series QFN 33-pin Diagram

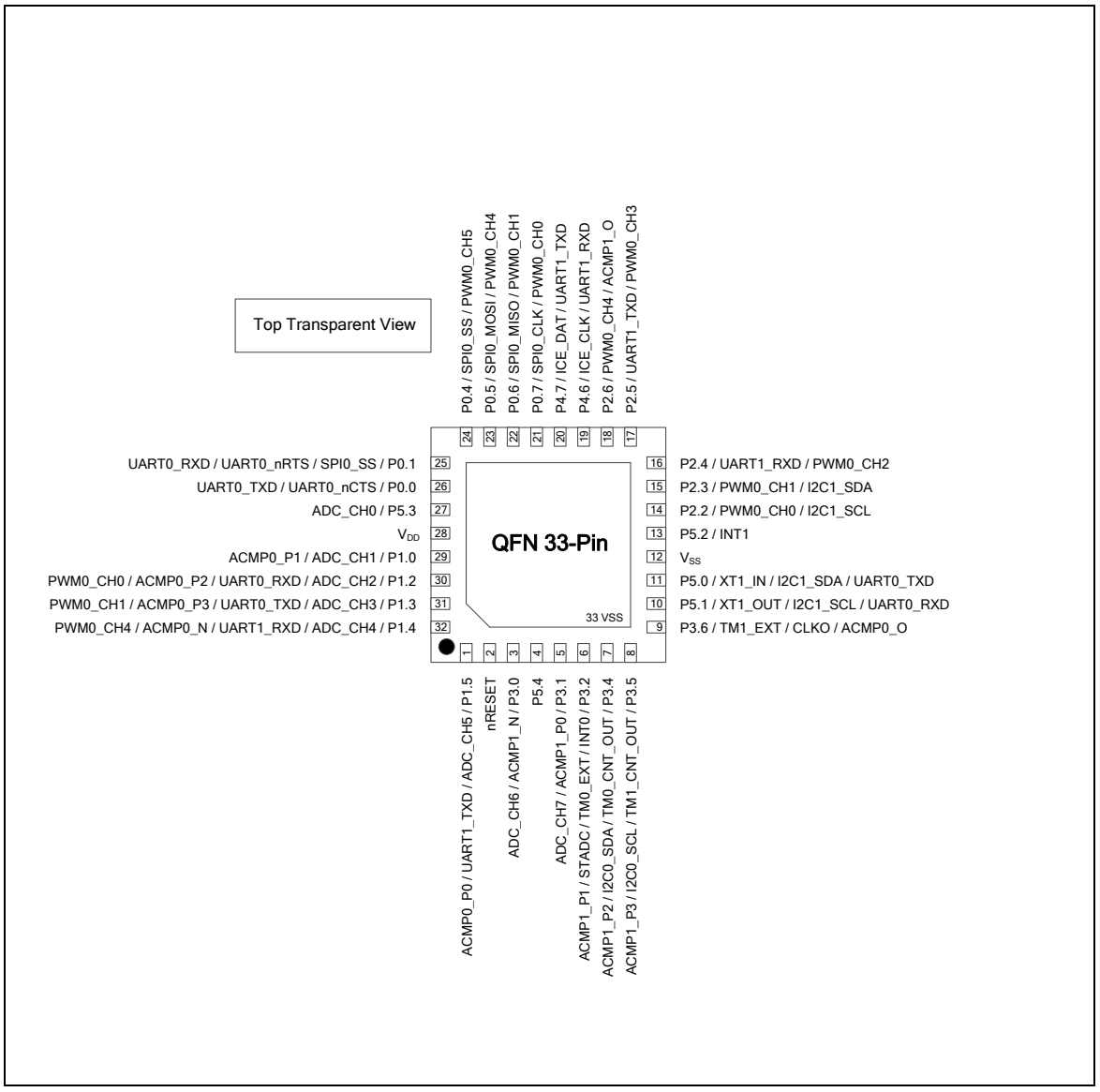


Figure 4.3-4 NuMicro® Mini58 Series QFN 33-pin Multi-funciton Diagram

4.3.3 TSSOP 20-pin

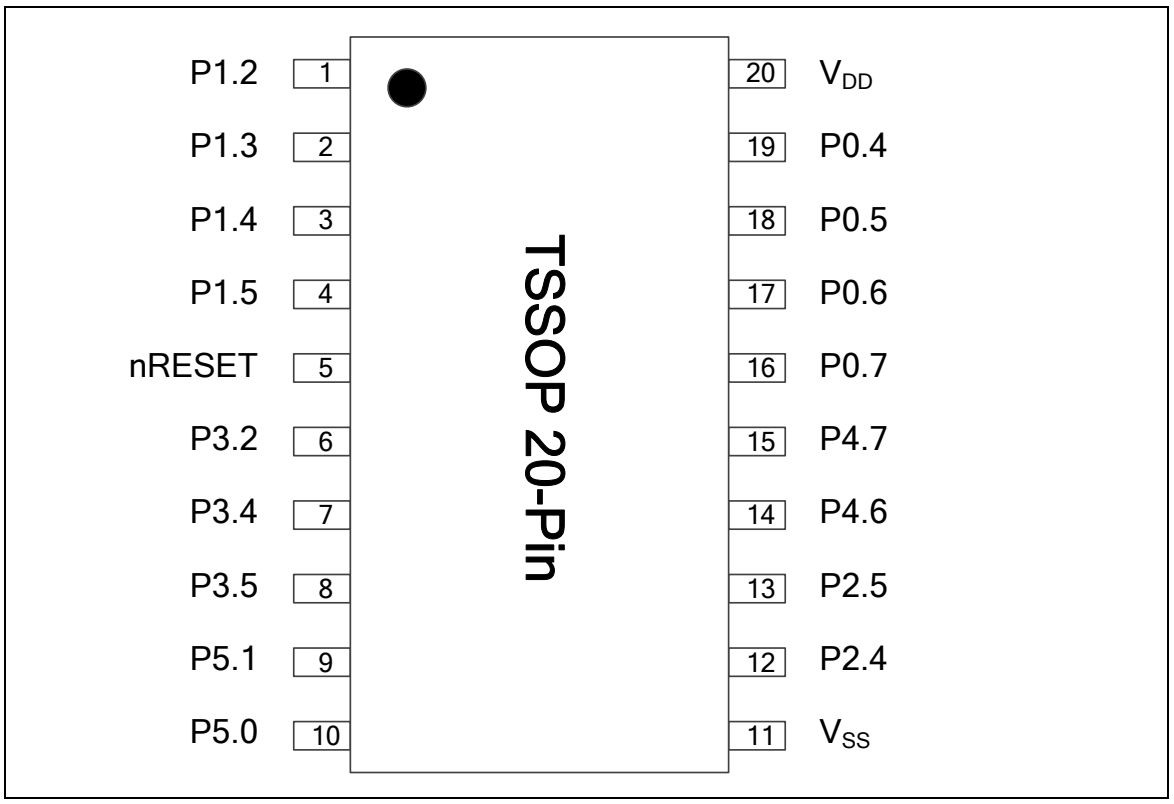


Figure 4.3-5 NuMicro® Mini58 Series TSSOP 20-pin Diagram

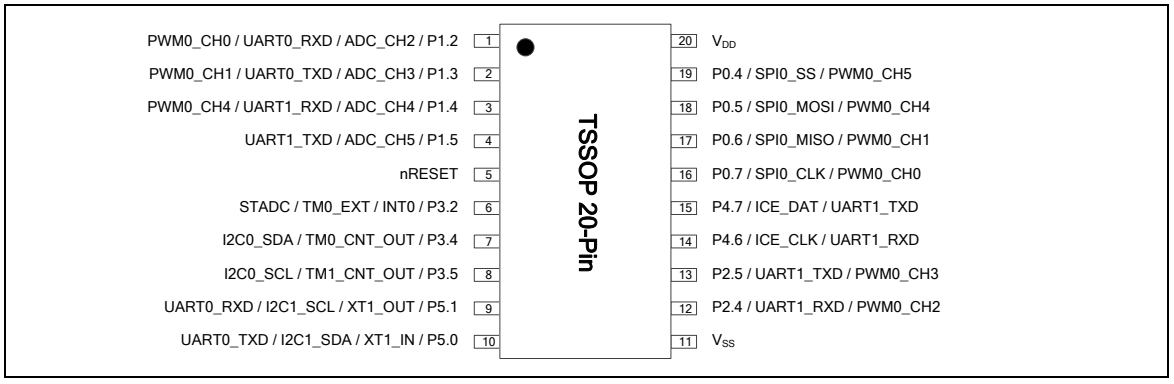


Figure 4.3-6 NuMicro® Mini58 Series TSSOP 20-pin Multi-function Diagram

4.4 Pin Description

Pin Number			Pin Name	Pin Type	Description
LQFP 48-pin	QFN 33-pin	TSSOP 20-pin			
1	---	---	NC	**---**	Not connected
2	1	4	P1.5	**I/O**	General purpose digital I/O pin
			ADC_CH5	**AI**	ADC analog input pin
			UART1_TXD	**O**	UART1 transmitter output pin
			ACMP0_P0	**AI**	Analog comparator positive input pin
3	2	5	nRESET	**I(ST)**	The Schmitt trigger input pin for hardware device reset. A "**Low**" on this pin for 768 clock counter of Internal RC 22.1184 MHz while the system clock is running will reset the device. nRESET pin has an internal pull-up resistor allowing power-on reset by simply connecting an external capacitor to GND. **Note:** It is recommended to use 10 kΩ pull-up resistor and 10 uF capacitor on nRESET pin.
4	3	---	P3.0	**I/O**	General purpose digital I/O pin
			ADC_CH6	**AI**	ADC analog input pin
			ACMP1_N	**AI**	Analog comparator negative input pin
5	---	---	AV_{SS}	**AP**	Ground pin for analog circuit
6	4	---	P5.4	**I/O**	General purpose digital I/O pin
7	5	---	P3.1	**I/O**	General purpose digital I/O pin
			ADC_CH7	**AI**	ADC analog input pin
			ACMP1_P0	**AI**	Analog comparator positive input pin
8	6	6	P3.2	**I/O**	General purpose digital I/O pin
			INT0	**I**	External interrupt 0 input pin
			STADC	**I**	ADC external trigger input pin
			TM0_EXT	**I/O**	Timer 0 external capture / reset trigger input pin / toggle output pin
			ACMP1_P1	**AI**	Analog comparator positive input pin (not support in TSSOP20 package)
9	7	7	P3.4	**I/O**	General purpose digital I/O pin
			TM0_CNT_OUT	**I/O**	Timer 0 external event counter input pin / toggle output pin
			I2C0_SDA	**I/O**	I^2C0 data I/O pin
			ACMP1_P2	**AI**	Analog comparator positive input pin
10	8	8	P3.5	**I/O**	General purpose digital I/O pin
			TM1_CNT_OUT	**I/O**	Timer 1 external event counter input pin / toggle output pin
			I2C0_SCL	**I/O**	I^2C0 clock I/O pin
			ACMP1_P3	**AI**	Analog comparator positive input pin

Pin Number			Pin Name	Pin Type	Description
LQFP 48-pin	QFN 33-pin	TSSOP 20-pin			
11	---	---	NC	**---**	Not connected
12	---	---	NC	**---**	Not connected
13	---	--	NC	**---**	Not connected
14	9	---	P3.6	**I/O**	General purpose digital I/O pin
			ACMP0_O	**O**	Analog comparator output pin
			CLKO	**O**	Frequency divider output pin
			TM1_EXT	**I/O**	Timer 1 external capture / reset trigger input pin / toggle output pin
15	10	9	P5.1	**I/O**	General purpose digital I/O pin
			XT1_OUT	**O**	The output pin from the internal inverting amplifier. It emits the inverted signal of XT1_IN.
			I2C1_SCL	**I/O**	I^2C1 clock I/O pin
			UART0_RXD	**I**	UART0 data receiver input pin
16	11	10	P5.0	**I/O**	General purpose digital I/O pin
			XT1_IN	**I**	The input pin to the internal inverting amplifier. The system clock could be from external crystal or resonator.
			I2C1_SDA	**I/O**	I^2C1 data I/O pin
			UART0_TXD	**O**	UART0 transmitter output pin
17	12	11	V_{SS}	**P**	Ground pin for digital circuit
	33				
18	---	---	LDO_CAP	**P**	LDO output pin
19	---	---	P5.5	**I/O**	General purpose digital I/O pin User program must enable pull-up resistor in the QFN-33 package.
20	13	---	P5.2	**I/O**	General purpose digital I/O pin
			INT1	**I**	External interrupt 1 input pin
21	---	---	NC	**---**	Not connected
22	14	---	P2.2	**I/O**	General purpose digital I/O pin
			PWM0_CH0	**O**	PWM0 output of PWM unit
			I2C1_SCL	**I/O**	I^2C1 clock I/O pin
23	15	---	P2.3	**I/O**	General purpose digital I/O pin
			PWM0_CH1	**O**	PWM1 output of PWM unit
			I2C1_SDA	**I/O**	I^2C1 data I/O pin
24	16	12	P2.4	**I/O**	General purpose input/output digital pin
			UART1_RXD	**I**	UART1 data receiver input pin

Pin Number			Pin Name	Pin Type	Description
LQFP 48-pin	QFN 33-pin	TSSOP 20-pin			
			PWM0_CH2	**O**	PWM2 output of PWM unit
25	17	13	P2.5	**I/O**	General purpose digital I/O pin
			UART1_TXD	**O**	UART1 transmitter output pin
			PWM0_CH3	**O**	PWM3 output of PWM unit
26	18	---	P2.6	**I/O**	General purpose digital I/O pin
			PWM0_CH4	**O**	PWM4 output of PWM unit
			ACMP1_O	**O**	Analog comparator output pin
27	---	---	NC	**---**	Not connected
28	---	---	NC	**---**	Not connected
29	19	14	P4.6	**I/O**	General purpose digital I/O pin
			ICE_CLK	**I**	Serial wired debugger clock pin **Note:** It is recommended to use 100 kΩ pull-up resistor on ICE_CLK pin
			UART1_RXD	**I**	UART1 data receiver input pin
30	20	15	P4.7	**I/O**	General purpose digital I/O pin
			ICE_DAT	**I/O**	Serial wired debugger data pin **Note:** It is recommended to use 100 kΩ pull-up resistor on ICE_DAT pin
			UART1_TXD	**O**	UART1 transmitter output pin
31	---	---	NC	**---**	Not connected
32	21	16	P0.7	**I/O**	General purpose digital I/O pin
			SPI0_CLK	**I/O**	SPI serial clock pin
			PWM0_CH0	**O**	PWM0 output of PWM unit
33	22	17	P0.6	**I/O**	General purpose digital I/O pin
			SPI0_MISO	**I/O**	SPI MISO (master in/slave out) pin
			PWM0_CH1	**O**	PWM1 output of PWM unit
34	23	18	P0.5	**I/O**	General purpose digital I/O pin
			SPI0_MOSI	**O**	SPI MOSI (master out/slave in) pin
			PWM0_CH4	**O**	PWM4 output of PWM unit
35	24	19	P0.4	**I/O**	General purpose digital I/O pin
			SPI0_SS	**I/O**	SPI slave select pin
			PWM0_CH5	**O**	PWM5 output of PWM unit
36	---	---	NC	**---**	Not connected
37	25	---	P0.1	**I/O**	General purpose digital I/O pin

Pin Number			Pin Name	Pin Type	Description
LQFP 48-pin	QFN 33-pin	TSSOP 20-pin			
			UART0_nRTS	**O**	UART0 RTS pin
			UART0_RXD	**I**	UART0 data receiver input pin
			SPI0_SS	**I/O**	SPI slave select pin
38	26	---	P0.0	**I/O**	General purpose digital I/O pin
			UART0_nCTS	**I**	UART0 CTS pin
			UAR0_TXD	**O**	UART0 transmitter output pin
39	---	---	NC	**---**	Not connected
40	---	---	NC	**---**	Not connected
41	27	---	P5.3	**I/O**	General purpose digital I/O pin
			ADC_CH0	**AI**	ADC analog input pin
42	28	20	V_{DD}	**P**	Power supply for digital circuit
43			AV_{DD}	**P**	Power supply for analog circuit
44	29	---	P1.0	**I/O**	General purpose digital I/O pin
			ADC_CH1	**AI**	ADC analog input pin
			ACMP0_P1	**AI**	Analog comparator positive input pin
45	30	1	P1.2	**I/O**	General purpose digital I/O pin
			ADC_CH2	**AI**	ADC analog input pin
			UART0_RXD	**I**	UART0 data receiver input pin
			ACMP0_P2	**AI**	Analog comparator positive input pin (not support in TSSOP20 package)
			PWM0_CH0	**O**	PWM0 output of PWM unit
46	31	2	P1.3	**I/O**	General purpose digital I/O pin
			ADC_CH3	**AI**	ADC analog input pin
			UART0_TXD	**O**	UART0 transmitter output pin
			ACMP0_P3	**AI**	Analog comparator positive input pin (not support in TSSOP20 package)
			PWM0_CH1	**O**	PWM1 output of PWM unit
47	32	3	P1.4	**I/O**	General purpose digital I/O pin
			ADC_CH4	**I/O**	ADC analog input pin
			UART1_RXD	**I**	UART1 data receiver input pin
			ACMP0_N	**AI**	Analog comparator negative input pin (not support in TSSOP20 package)
			PWM0_CH4	**O**	PWM4 output of PWM unit

Pin Number			Pin Name	Pin Type	Description
LQFP 48-pin	QFN 33-pin	TSSOP 20-pin			
48	---	--	NC	**---**	Not connected

Table 4.4-1 NuMicro® Mini58 Series Pin Description

[1] I/O type description. I: input, O: output, I/O: quasi bi-direction, D: open-drain, P: power pin, ST: Schmitt trigger, A: Analog input.

5 BLOCK DIAGRAM

5.1 NuMicro® Mini58 Block Diagram

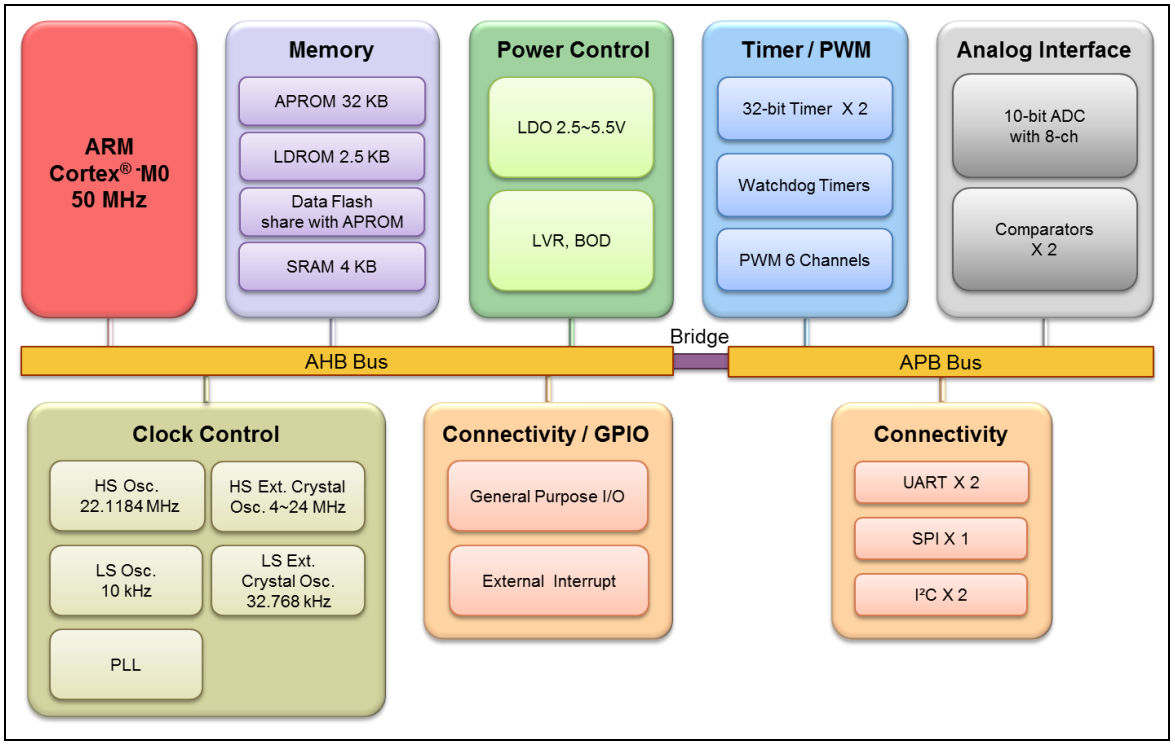


Figure 5.1-1 NuMicro® Mini58 Series Block Diagram

6 FUNCTIONAL DESCRIPTION

6.1 ARM® Cortex®-M0 Core

6.1.1 Overview

The Cortex®-M0 processor, a configurable, multistage, 32-bit RISC processor, 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 processors. 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 the processor.

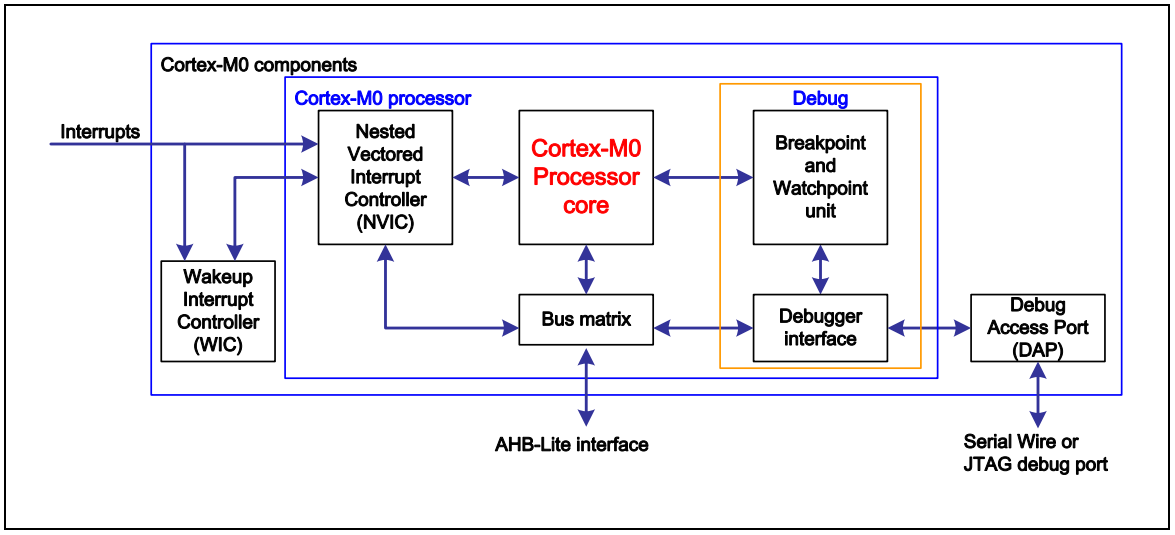


Figure 6.1-1 Functional Block Diagram

6.1.2 Features

- 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 Idle 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 Idle mode
- Debug support
  - Four hardware breakpoints
  - Two watch points
  - 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

System management includes the following sections:

- System Reset
- System Power Architecture
- System Memory Map
- System management registers for Part Number ID, chip reset and on-chip controllers reset, and multi-functional pin control
- System Timer (SysTick)
- Nested Vectored Interrupt Controller (NVIC)
- System Control registers

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 can reset chip through peripheral reset signals. Software reset can trigger reset through 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 (SCS_AIRCR[2])
  - CPU Reset for Cortex®-M0 core Only by writing 1 to CPURST (SYS_IPRST0[1])

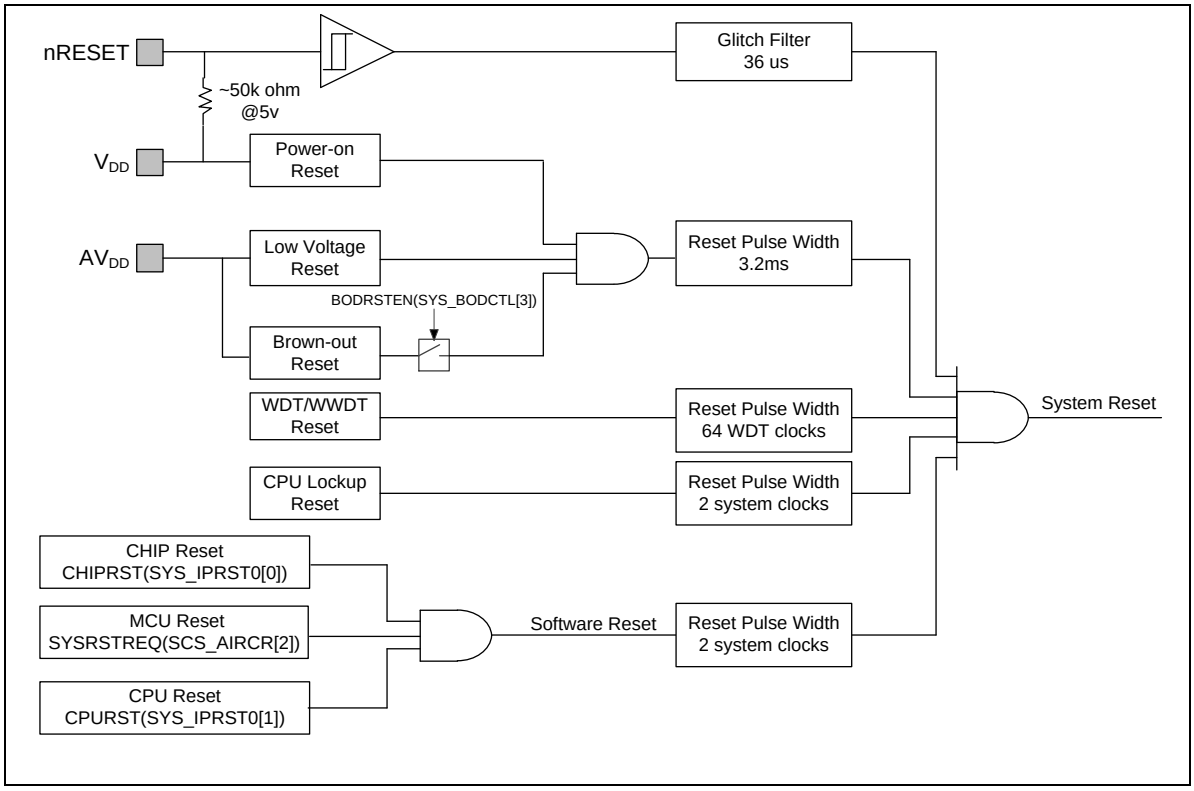


Figure 6.2-1 System Rese Resources

There are a total of 9 reset sources in the NuMicro® family. In general, CPU reset is used to reset Cortex®-M0 only; the other reset sources will reset Cortex®-M0 and all peripherals. However, there are small differences between each reset source and they are listed in Table 6.2-5.

Reset Sources / Register	POR	nRESET	WDT	LVR	BOD	Lockup	CHIP	MCU	CPU
SYS_RSTSTS	0x001	Bit 1 = 1	Bit 2 = 1	0x001	Bit 4 = 1	Bit 8 = 1	Bit 0 = 1	Bit 5 = 1	Bit 7 = 1
CHIPRST (SYS_IPRST0[0])	0x0	-	-	-	-	-	-	-	-
BODEN (SYS_BODCTL[0])	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	-	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	-
BODVL (SYS_BODCTL[2:1])									
BODRSTEN (SYS_BODCTL[3])									
XTLEN (CLK_PWRCTL[1:0])	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	
WDTCKEN (CLK_APBCLK0[0])	0x1	-	0x1	-	-	-	0x1	-	-
HCLKSEL (CLK_CLKSEL0[2:0])	0x8	0x8	0x8	0x8	0x8	0x8	0x8	0x8	-

WDTSEL (CLK_CLKSEL1[1:0])	0x3	0x3	-	-	-	-	-	-	-
XLTSTB (CLK_STATUS[0])	0x0	-	-	-	-	-	-	-	-
PLLSTB (CLK_STATUS[2])	0x0	-	-	-	-	-	-	-	-
LIRCSTB (CLK_STATUS[3])	0x0								
HIRCSTB (CLK_STATUS[4])	0x0	-	-	-	-	-	-	-	-
CLKSFAIL (CLK_STATUS[7])	0x0	0x0	-	-	-	-	-	-	-
WDT_CTL	0x0700	0x0700	0x0700	0x0700	0x0700	-	0x0700	-	-
WDT_ALTCTL	0x0000	0x0000	0x0000	0x0000	0x0000	-	0x0000	-	-
WWDT_RLDCNT	0x0000	0x0000	0x0000	0x0000	0x0000	-	0x0000	-	-
WWDT_CTL	0x3F0800	0x3F0800	0x3F0800	0x3F0800	0x3F0800	-	0x3F0800	-	-
WWDT_STATUS	0x0000	0x0000	0x0000	0x0000	0x0000	-	0x0000	-	-
WWDT_CNT	0x3F	0x3F	0x3F	0x3F	0x3F	-	0x3F	-	-
BS (FMC_ISPCTL[1])	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	-	Reload from CONFIG0	-	-
ISPEN (FMC_ISPCTL[16])									
FMC_DFBA	Reload from CONFIG1	Reload from CONFIG1	Reload from CONFIG1	Reload from CONFIG1	Reload from CONFIG1	-	Reload from CONFIG1	-	-
CBS (FMC_ISPSTS[2:1])	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	-	Reload from CONFIG0	-	-
VECMAP (FMC_ISPSTS[20:9])	Reload base on CONFIG0	Reload base on CONFIG0	Reload base on CONFIG0	Reload base on CONFIG0	Reload base on CONFIG0	-	Reload base on CONFIG0	-	-
Other Peripheral Registers	Reset Value								-
FMC Registers	Reset Value								
Note: '-' means that the value of register keeps original setting.									

Table 6.2-1 Reset Value of Registers

6.2.2.1 nRESET Reset

The nRESET reset means to generate a reset signal by pulling low nRESET pin, which is an asynchronous reset input pin and can be used to reset system at any time. When the nRESET voltage is lower than 0.2 V_{DD} and the state keeps longer than 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_{DD} 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.

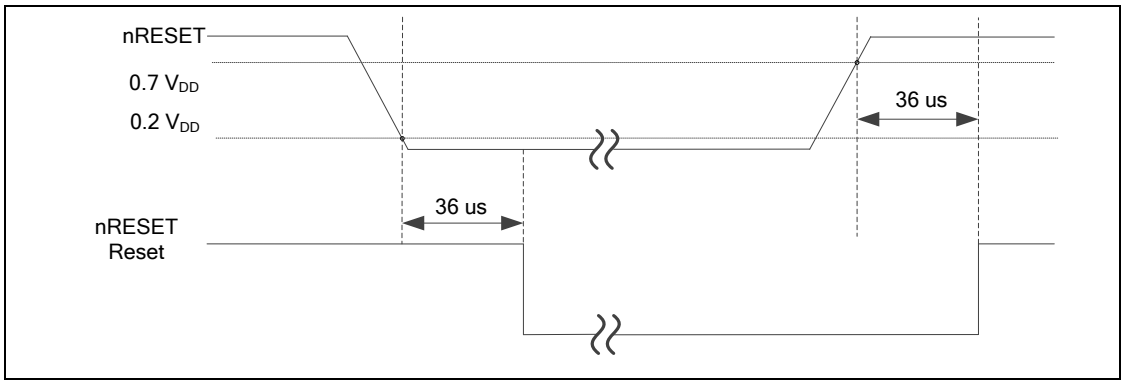


Figure 6.2-2 nRESET Reset Waveform

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 waveform of Power-On reset.

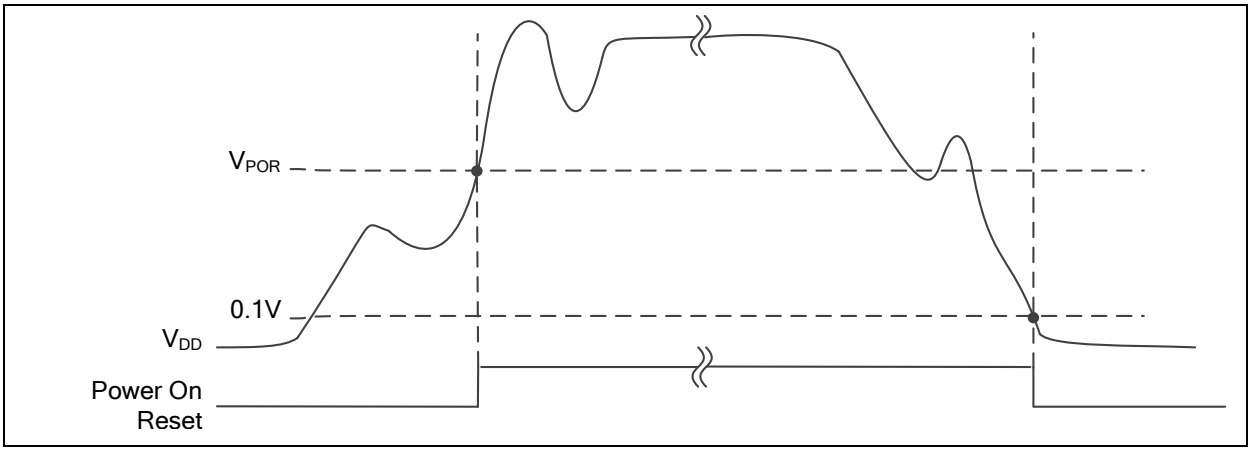


Figure 6.2-3 Power-on Reset (POR) Waveform

6.2.2.3 Low Voltage Reset (LVR)

Low Voltage Reset detects AV_{DD} during system operation. When the AV_{DD} voltage is lower than V_{LVR} and the state keeps longer than De-glitch time (16*HCLK cycles), chip will be reset. The LVR reset will control the chip in reset state until the AV_{DD} voltage rises above V_{LVR} and the state keeps longer than De-glitch time. The PINRF (SYS_RSTSTS[1]) will be set to 1 if the previous reset source is nRESET reset. Figure 6.2-4 shows the Low Voltage Reset waveform.

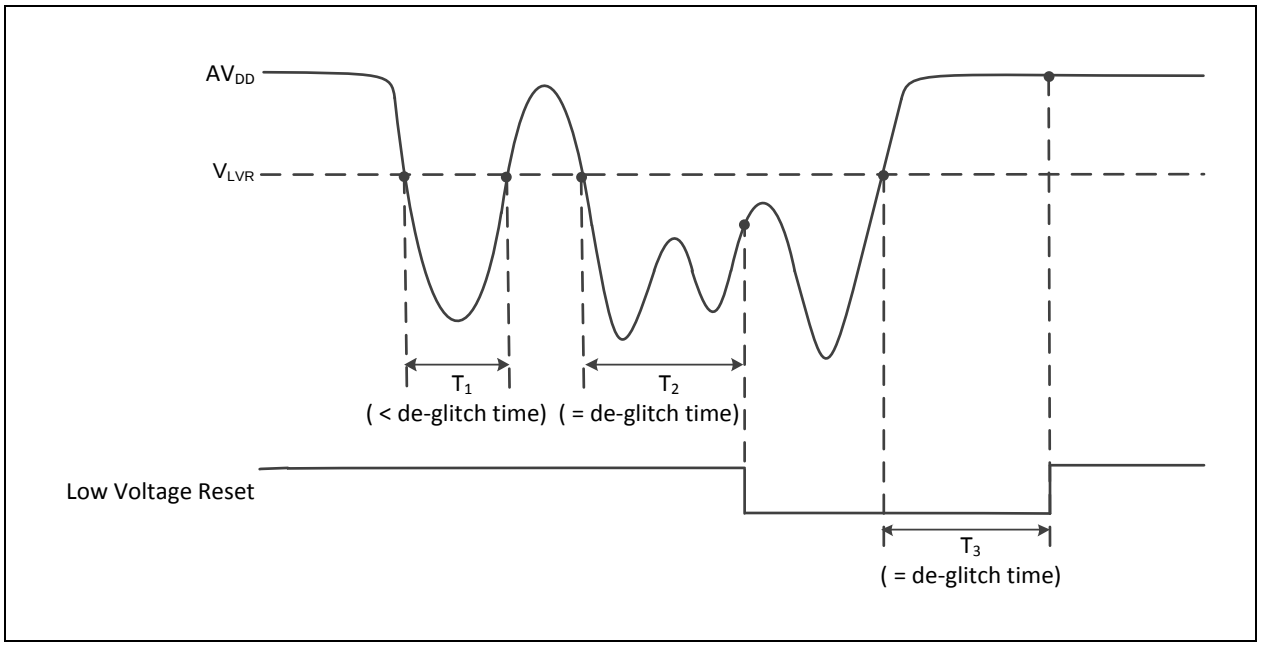


Figure 6.2-4 Low Voltage Reset (LVR) Waveform

6.2.2.4 Brown-out Detector Reset (BOD Reset)

If the Brown-out Detector (BOD) function is enabled by setting the Brown-out Detector Enable Bit BODEN (SYS_BODCTL[0]), Brown-Out Detector function will detect AV_{DD} during system operation. When the AV_{DD} voltage is lower than V_{BOD} which is decided by BODEN (SYS_BODCTL[0]) and BODVL (SYS_BODCTL[2:1]) and the state keeps longer than De-glitch time (Max(20*HCLK cycles, 1*LIRC cycle)), chip will be reset. The BOD reset will control the chip in reset state until the AV_{DD} voltage rises above V_{BOD} and the state keeps longer than De-glitch time. The default value of BODEN, BODVL and BODRSTEN is set by flash controller user configuration register CBOVEXT (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.

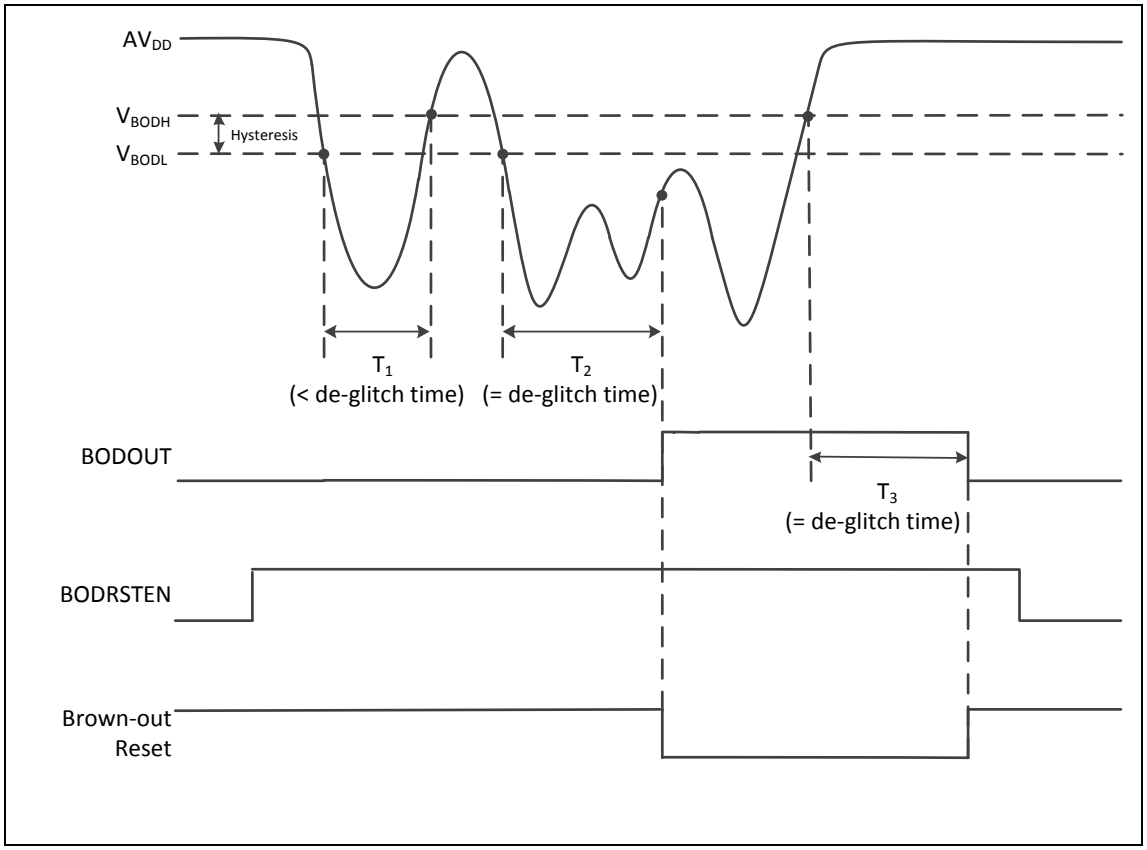


Figure 6.2-5 Brown-out Detector (BOD) Waveform

6.2.2.5 Watch Dog Timer Reset

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 watch dog 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 watch dog time-out. User may decide to enable system reset during watch dog 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 watch dog time-out to indicate the previous reset is a watch dog reset and handle the failure of MCU after watch dog 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 SYSTEM Reset

The CPU Reset means only Cortex®-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 CONFIG setting. User can set the CHIPRST (SYS_IPRST0[0]) 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 CONFIG setting and keep its original software setting for booting from APROM or LDROM. User can set the SYSRESETREQ (SCS_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
Definition	CPU is in active state	CPU is in sleep state	CPU is in sleep state and all clocks stop except LXT and LIRC. SRAM content retended.
Entry Condition	Chip is in normal mode after system reset released	CPU executes WFI instruction.	CPU sets sleep mode enable and power down enable and executes WFI instruction.
Wake-up Sources	N/A	All interrupts	WDT, I²C, Timer, UART, BOD and GPIO
Available Clocks	All	All except CPU clock	LXT and LIRC
After Wake-up	N/A	CPU back to normal mode	CPU back to normal mode

Table 6.2-2 Power Mode Difference Table

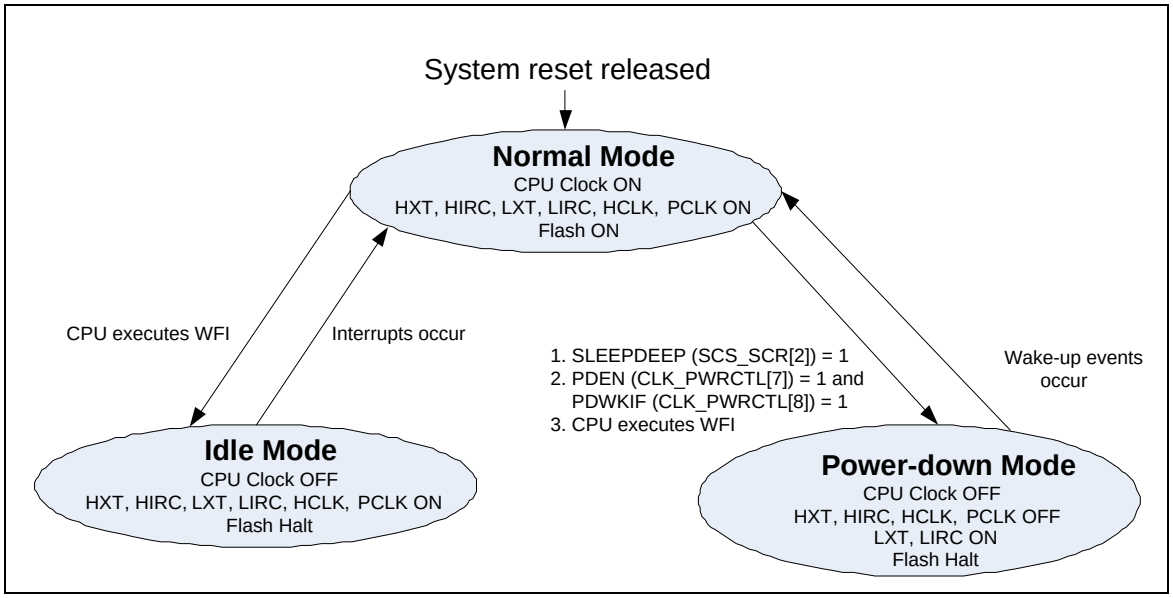


Figure 6.2-6 Power Mode State Machine

1. LXT (32768 Hz XTL) ON or OFF depends on SW setting in run mode.
2. LIRC (10 kHz OSC) ON or OFF depends on S/W setting in run 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.

	Normal Mode	Idle Mode	Power-down Mode
HXT (4~20 MHz XTL)	ON	ON	Halt
HIRC (12/16 MHz OSC)	ON	ON	Halt
LXT (32768 Hz XTL)	ON	ON	ON/OFF[1]
LIRC (10 kHz OSC)	ON	ON	ON/OFF[2]
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
TIMER	ON	ON	ON/OFF[3]
PWM	ON	ON	Halt
WDT	ON	ON	ON/OFF[4]
WWDT	ON	ON	Halt
UART	ON	ON	Halt
I^2C	ON	ON	Halt
SPI	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:

WDT, I²C, Timer, UART, BOD and GPIO

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 SYS_BODCTL[BODIF].
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).
UART	nCTS wake-up	After software writes 1 to clear CTSWKIF (UARTx_INTSTS[16]).
I^2C	Falling edge in the I2C_SDA or I2C_CLK	After software writes 1 to clear WKIF (I2C_STATUS1[0]).

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

6.2.4 System Power Architecture

In this chip, the power distribution is divided into three segments.

- Analog power from AV_{DD} and AV_{SS} provides the power for analog components operation. AV_{DD} must be equal to V_{DD} to avoid leakage current.
- Digital power from V_{DD} and V_{SS} supplies power to the I/O pins and internal regulator which provides a fixed 1.8V power for digital operation.
- Built-in a capacitor for internal voltage regulator

The output of internal voltage regulator, LDO_CAP, requires an external capacitor which should be located close to the corresponding pin. Analog power (AV_{DD}) should be the same voltage level as the digital power (V_{DD}). Figure 6.2-7 shows the power distribution of the Mini58 series.

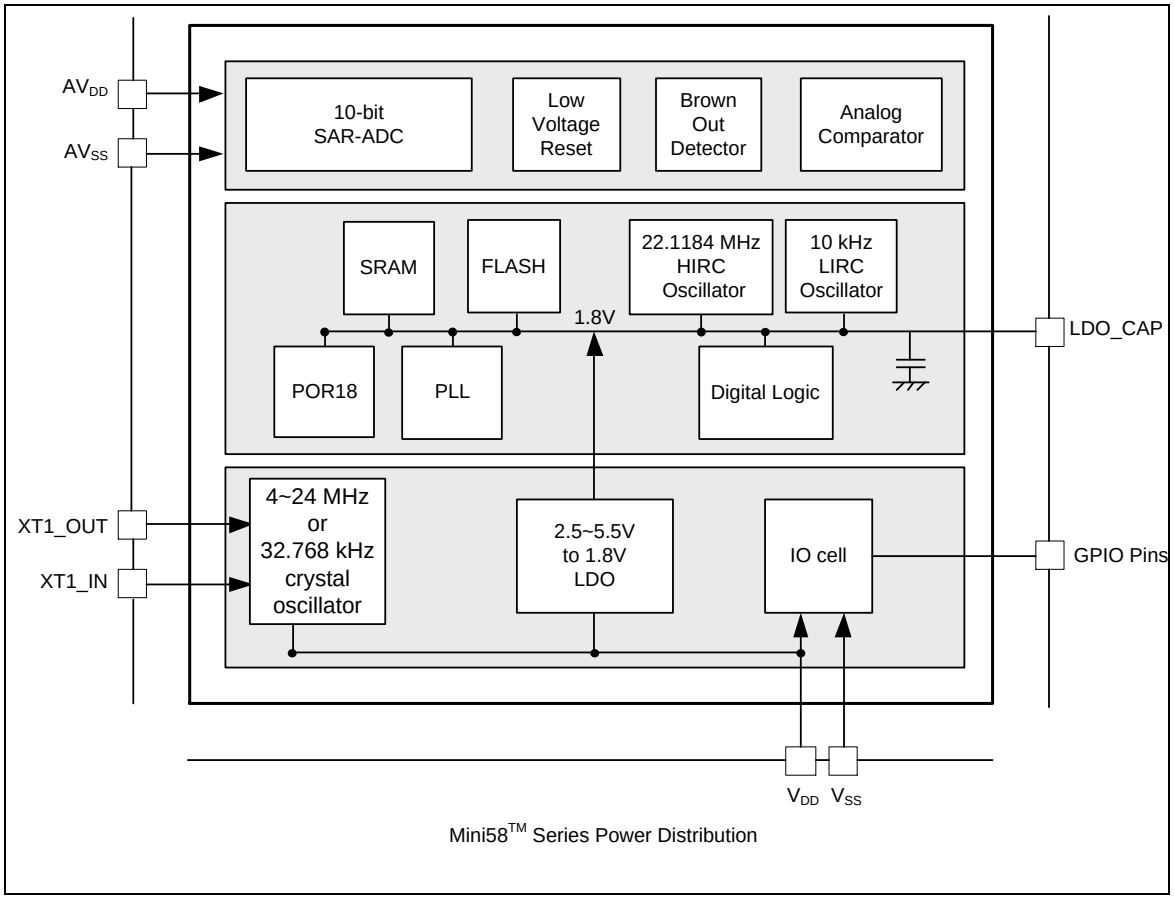


Figure 6.2-7 NuMicro® Mini58 Series Power Architecture Diagram

6.2.5 System Memory Mapping

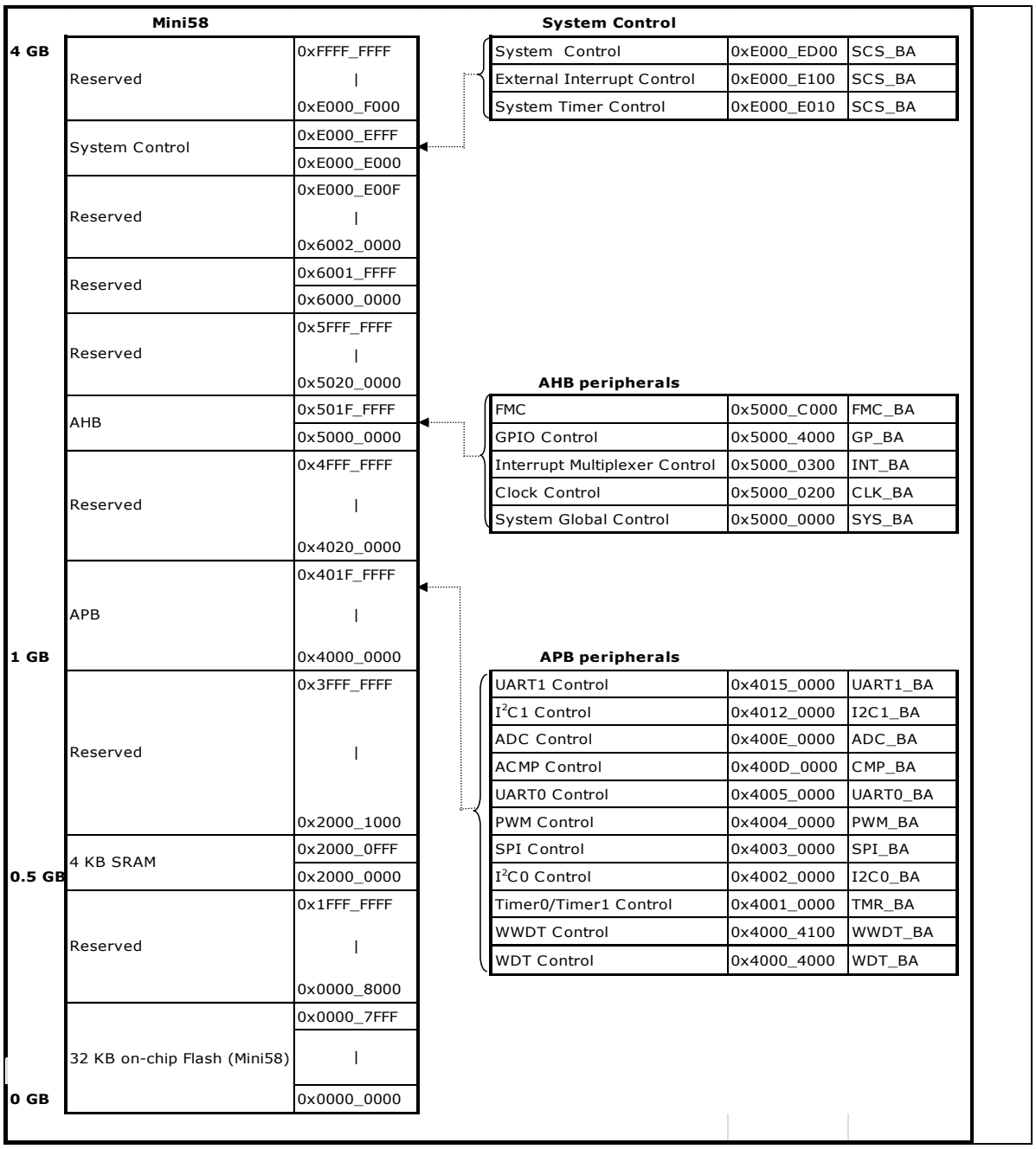


Table 6.2-5 Memory Mapping Table

6.2.6 Register Protection

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 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 REGLCTL (SYS_REGLCTL [0]), “1” is protection disable, “0” is protection enable. Then user can update the target protected register value and then write any data to SYS_REGLCTL to enable register protection.

The protected registers are listed as following table.

Register	Bit	Description
SYS_IPRST0	[1] CPURST	Processor Core One-shot Reset (Write Protect)
	[0] CHIPRST	CHIP One-shot Reset (Write Protect)
SYS_BODCTL	[5] BODLPM	Brown-out Detector Low Power Mode (Write Protect)
	[3] BODRSTEN	Brown-out Reset Enable Bit (Write Protect)
	[2:1] BODVL	Brown-out Detector Threshold Voltage Selection (Write Protect)
NMI_CON	[8] NMI_SEL_EN	NMI Interrupt Enable Bit (Write Protected)
CLK_PWRCTL	[7] PDEN	System Power-down Enable Bit (Write Protect)
	[5] PDWKIEN	Power-down Mode Wake-up Interrupt Enable Bit (Write Protect)
	[4] PDWKDLY	Wake-up Delay Counter Enable Bit (Write Protect)
	[3] LIRCEN	LIRC Enable Bit (Write Protect)
	[2] HIRCEN	HIRC Enable Bit (Write Protect)
	[1:0] XTLEN	External HXT Or LXT Crystal Oscillator Enable Bit (Write Protect)
CLK_APBCLK	[0] WDTCKEN	Watchdog Timer Clock Enable Bit (Write Protect)
CLK_CLKSEL0	[5:3] STCLKSEL	Cortex®-M0 SysTick Clock Source Selection (Write Protect)
	[2:0] HCLKSEL	HCLK Clock Source Selection (Write Protect)
CLK_CLKSEL1	[1:0] WDTSEL	WDT CLK Clock Source Selection (Write Protect)
FMC_ISPCTL	[6] ISPFF	ISP Fail Flag (Write Protect)
	[5] LDUEN	LDROM Update Enable (Write Protect)
	[4] CFGUEN	CONFIG Update Enable Bit (Write Protect)
	[3] APUEN	APROM Update Enable Bit (Write Protect)
	[2] SPUEN	SPROM Update Enable Bit (Write Protect)
	[1] BS	Boot Select (Write Protect)
	[0] ISPEN	ISP Enable Bit (Write Protect)
FMC_ISPTRG	[0] ISPGO	ISP Start Trigger (Write Protect)
FMC_FATCTL	[6:4] FOM	Frequency Optimization Mode (Write Protect)
FMC_ISPSTS	[6] ISPFF	ISP Fail Flag (Write Protect)
TIMER0_CTL	[31] ICEDEBUG	ICE Debug Mode Acknowledge Disable Bit (Write Protect)
TIMER1_CTL	[31] ICEDEBUG	ICE Debug Mode Acknowledge Disable Bit (Write Protect)
WDT_CTL	[31] ICEDEBUG	ICE Debug Mode Acknowledge Disable Bit (Write Protect)

	[10:8] TOUTSEL	WDT Time-out Interval Selection (Write Protect)
	[7] WDTEN	WDT Enable Bit (Write Protect)
	[6] INTEN	WDT Time-out Interrupt Enable Bit (Write Protect)
	[5] WKF	WDT Time-out Wake-up Flag (Write Protect)
	[4] WKEN	WDT Time-out Wake-up Function Control (Write Protect)
	[1] RSTEN	WDT Time-out Reset Enable Bit (Write Protect)
WDT_ALTCTL	[1:0] RSTDSEL	WDT Reset Delay Selection (Write Protect)

6.2.7 Memory Organization

6.2.7.1 Overview

The NuMicro® Mini58 series provides 4G-byte addressing space. The addressing space assigned to each on-chip controllers is shown the following table. The detailed register definition, addressing space, and programming details will be described in the following sections for each on-chip peripheral. The Mini58 series only supports little-endian data format.

6.2.7.2 System Memory Map

The memory locations assigned to each on-chip controllers are shown in the following table.

Addressing Space	Token	Modules
Flash and SRAM Memory Space		
0x0000_0000 – 0x0000_7FFF	FLASH_BA	Flash Memory Space (32 KB)
0x2000_0000 – 0x2000_0FFF	SRAM_BA	SRAM Memory Space (4 KB)
AHB Modules Space (0x5000_0000 – 0x501F_FFFF)		
0x5000_0000 – 0x5000_01FF	SYS_BA	System Global 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	GP_BA	GPIO (P0~P5) Control Registers
0x5000_C000 – 0x5000_FFFF	FMC_BA	Flash Memory Control Registers
APB Modules Space (0x4000_0000 – 0x401F_FFFF)		
0x4000_4000 – 0x4000_00FF	WDT_BA	Watchdog Timer Control Registers
0x4000_4100 – 0x4000_47FF	WWDT_BA	Window Watchdog Timer Control Registers
0x4001_0000 – 0x4001_3FFF	TMR_BA	Timer0/Timer1 Control Registers
0x4002_0000 – 0x4002_3FFF	I2C0_BA	I^2C0 Interface Control Registers
0x4003_0000 – 0x4003_3FFF	SPI_BA	SPI with Master/slave Function Control Registers
0x4004_0000 – 0x4004_3FFF	PWM_BA	PWM Control Registers
0x4005_0000 – 0x4005_3FFF	UART0_BA	UART0 Control Registers
0x400D_0000 – 0x400D_3FFF	ACMP_BA	Analog Comparator Control Registers

Addressing Space	Token	Modules
0x400E_0000 – 0x400E_3FFF	ADC_BA	Analog-Digital-Converter (ADC) Control Registers
0x4012_0000 – 0x4012_3FFF	I2C1_BA	I^2C1 Interface Control Registers
0x4015_0000 – 0x4015_3FFF	UART1_BA	UART1 Control Registers
System Control Space (0xE000_E000 – 0xE000_EFFF)		
0xE000_E010 – 0xE000_E0FF	SCS_BA	System Timer Control Registers
0xE000_E100 – 0xE000_ECFF	SCS_BA	Nested Vectored Interrupt Control Registers
0xE000_ED00 – 0xE000_ED8F	SCS_BA	System Control Block Registers

Table 6.2-6 Address Space Assignments for On-Chip Modules

6.2.8 Register Map

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

Register	Offset	R/W	Description	Reset Value
SYS Base Address: **SYS_BA = 0x5000_0000**				
SYS_PDID	SYS_BA+0x00	R	Part Device Identification Number Register	0xXXXX_XXXX[1]
SYS_RSTSTS	SYS_BA+0x04	R/W	System Reset Status Register	0x0000_00XX
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_BODCTL	SYS_BA+0x18	R/W	Brown-out Detector Control Register	0x0000_000X
SYS_P0_MFP	SYS_BA+0x30	R/W	P0 Multiple Function and Input Type Control Register	0x0000_0000
SYS_P1_MFP	SYS_BA+0x34	R/W	P1 Multiple Function and Input Type Control Register	0x0000_0000
SYS_P2_MFP	SYS_BA+0x38	R/W	P2 Multiple Function and Input Type Control Register	0x0000_0000
SYS_P3_MFP	SYS_BA+0x3C	R/W	P3 Multiple Function and Input Type Control Register	0x0000_0000
SYS_P4_MFP	SYS_BA+0x40	R/W	P4 Multiple Function and Input Type Control Register	0x0000_00C0
SYS_P5_MFP	SYS_BA+0x44	R/W	P5 Multiple Function and Input Type Control Register	0x0000_0000
SYS_IRCTCTL	SYS_BA+0x80	R/W	HIRC 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
SYS_REGLCTL	SYS_BA+0x100	R/W	Register Write-Protection Control Register	0x0000_0000

6.2.9 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							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PDID							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PDID							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PDID							

Bits	Description	
[31:0]	**PDID**	**Product Device Identification Number (Read Only)** This register reflects the device part number code. Software can read this register to identify which device is used. For example, the MINI58LDE PDID code is “0x00A05800”.

NuMicro® Mini58 Series	Part Device Identification Number
MINI58LDE	0x00A05800
MINI58ZDE	0x00A05803
MINI58TDE	0x00A05804
MINI58FDE	0x00A05805

System Reset Status Register (SYS_RSTSTS)

This register provides specific information for software to identify the 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_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							CPULKRF
7	6	5	4	3	2	1	0
CPURF	Reserved	SYSRF	BODRF	Reserved	WDTRF	PINRF	PORF

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8]	**CPULKRF**	**The Cortex®-M0 LOCKUP Flag** 0 = No reset from Cortex®-M0 LOCKUP happened. 1 = The Cortex®-M0 LOCKUP happened and chip is reset. **Note:** Software can write 1 to clear this bit to zero.
[7]	**CPURF**	**CPU Reset Flag** 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). 0 = No reset from CPU. 1 = The Cortex®-M0 Core and FMC are reset by software setting CPURST to 1. **Note:** Software can write 1 to clear this bit to zero.
[6]	**Reserved**	Reserved.
[5]	**SYSRF**	**System Reset Flag** The system reset flag is set by the "Reset Signal" from the Cortex®-M0 Core to indicate the previous reset source. 0 = No reset from Cortex®-M0. 1 = The Cortex®-M0 had issued the reset signal to reset the system by writing 1 to the bit SYSRESETREQ (SCS_AIRCR[2]), Application Interrupt and Reset Control Register, address = 0xE000ED0C) in system control registers of Cortex®-M0 core. **Note:** Software can write 1 to clear this bit to zero.
[4]	**BODRF**	**BOD Reset Flag** The BOD reset flag is set by the "Reset Signal" from the Brown-out Detector to indicate the previous reset source. 0 = No reset from BOD. 1 = The BOD had issued the reset signal to reset the system. **Note:** Software can write 1 to clear this bit to zero.

Bits	Description	
[3]	**Reserved**	Reserved.
[2]	**WDTRF**	**WDT Reset Flag** The WDT reset flag is set by the “Reset Signal” from the Watchdog Timer or Window Watchdog Timer to indicate the previous reset source. 0 = No reset from watchdog timer or window watchdog timer. 1 = The watchdog timer or window watchdog timer had issued the reset signal to reset the system. **Note:** Software can write 1 to clear this bit to zero.
[1]	**PINRF**	**NRESET Pin Reset Flag** The nRESET pin reset flag is set by the “Reset Signal” from the nRESET pin to indicate the previous reset source. 0 = No reset from nRESET pin. 1 = Pin nRESET had issued the reset signal to reset the system. **Note:** Software can write 1 to clear this bit to zero.
[0]	**PORF**	**POR Reset Flag** The POR reset flag is set by the “Reset Signal” from the Power-on Reset (POR) Controller or bit CHIPRST (SYS_IPRST0[0]) to indicate the previous reset source. 0 = No reset from POR or CHIPRST. 1 = Power-on-Reset (POR) or CHIPRST had issued the reset signal to reset the system. **Note:** Software can write 1 to clear this bit to zero.

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**
Reserved						**CPURST**	**CHIPRST**

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**CPURST**	**Processor Core One-shot Reset (Write Protect)** Setting this bit will only reset the processor core and Flash Memory Controller (FMC), and this bit will automatically return to 0 after the 2 clock cycles. 0 = Processor core normal operation. 1 = Processor core one-shot reset. **Note:** This bit is write protected. Refer to the SYS_REGLCTL register.
[0]	**CHIPRST**	**CHIP One-shot Reset (Write Protect)** Setting this bit will reset the whole chip, including Processor core and all peripherals, and this bit will automatically return to 0 after the 2 clock cycles. The CHIPRST is the same as the POR reset, all the chip controllers is reset and the chip settings from flash are also reload. 0 = Chip normal operation. 1 = CHIP one-shot reset. **Note:** This bit is write protected. Refer to the SYS_REGLCTL register.

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	Reserved			
23	22	21	20	19	18	17	16
Reserved	ACMPRST	Reserved	PWM0RST	Reserved		UART1RST	UART0RST
15	14	13	12	11	10	9	8
Reserved			SPI0RST	Reserved		I2C1RST	I2C0RST
7	6	5	4	3	2	1	0
Reserved				TMR1RST	TMR0RST	GPIORST	Reserved

Bits	Description	
[31:29]	**Reserved**	Reserved.
[28]	**ADCRST**	**ADC Controller Reset** 0 = ADC controller normal operation. 1 = ADC controller reset.
[27:23]	**Reserved**	Reserved.
[22]	**ACMPRST**	**ACMP Controller Reset** 0 = ACMP controller normal operation. 1 = ACMP controller reset.
[21]	**Reserved**	Reserved.
[20]	**PWM0RST**	**PWM0 Controller Reset** 0 = PWM0 controller normal operation. 1 = PWM0 controller reset.
[19:18]	**Reserved**	Reserved.
[17]	**UART1RST**	**UART1 Controller Reset** 0 = UART1 controller normal operation. 1 = UART1 controller reset.
[16]	**UART0RST**	**UART0 Controller Reset** 0 = UART0 controller normal operation. 1 = UART0 controller reset.
[15:13]	**Reserved**	Reserved.
[12]	**SPI0RST**	**SPI0 Controller Reset** 0 = SPI controller normal operation. 1 = SPI controller reset.

Bits	Description	
[11:10]	**Reserved**	Reserved.
[9]	**I2C1RST**	**I²C1 Controller Reset** 0 = I²C1 controller normal operation. 1 = I²C1 controller reset.
[8]	**I2C0RST**	**I²C0 Controller Reset** 0 = I²C0 controller normal operation. 1 = I²C0 controller reset.
[7:4]	**Reserved**	Reserved.
[3]	**TMR1RST**	**Timer1 Controller Reset** 0 = Timer1 controller normal operation. 1 = Timer1 controller reset.
[2]	**TMR0RST**	**Timer0 Controller Reset** 0 = Timer0 controller normal operation. 1 = Timer0 controller reset.
[1]	**GPIORST**	**GPIO (P0~P5) Controller Reset** 0 = GPIO controller normal operation. 1 = GPIO controller reset.
[0]	**Reserved**	Reserved.

Brown-out Detector Control Register (SYS_BODCTL)

Partial of the SYS_BODCTL control register values are initiated by the flash configuration and write-protected by the lock function. If user needs to program the write-protected content, an unlocked sequence is needed. The unlocked sequence is to continuously write the data 0x59, 0x16, 0x88 to the key controller address 0x5000_0100. A different data value or any other write during the three data program aborts the whole sequence.

After the unlocked sequence, user can check the lock bit at address 0x5000_0100 bit 0, where 1 is unlocked and 0 is locked. Then user can update the write-protected registers. Write any data to the address 0x5000_0100 to re-lock the write-protected register again.

Register	Offset	R/W	Description	Reset Value
SYS_BODCTL	SYS_BA+0x18	R/W	Brown-out Detector Control Register	0x0000_000X

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

Bits	Description	
[31:7]	**Reserved**	Reserved.
[6]	**BODOUT**	**Brown-out Detector Output Status** 0 = Brown-out Detector status output is 0, the detected voltage is higher than BODVL setting. 1 = Brown-out Detector status output is 1, the detected voltage is lower than BODVL setting.
[5]	**BODLPM**	**Brown-out Detector Low Power Mode (Write Protect)** 0 = BOD operate in normal mode (default). 1 = BOD Low Power mode Enabled. **Note:** 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.
[4]	**BODIF**	**Brown-out Detector Interrupt Flag** 0 = Brown-out Detector does not detect any voltage draft at V_{DD} down through or up through the voltage of BODVL setting. 1 = When Brown-out Detector detects the V_{DD} is dropped through the voltage of BODVL setting or the V_{DD} is raised up through the voltage of BODVL setting, this bit is set to 1 and the Brown-out interrupt is requested if Brown-out interrupt is enabled.

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[3]</td><td>BODRSTEN</td><td>Brown-out Reset Enable Bit (Write Protect)
The default value is set by flash controller user configuration register CBORST(CONFIG0[20]) bit.
0 = Brown-out “INTERRUPT” function Enabled; when the Brown-out Detector function is enable and the detected voltage is lower than the threshold, then assert a signal to interrupt the Cortex®-M0 CPU.
1 = Brown-out “RESET” function Enabled; when the Brown-out Detector function is enable and the detected voltage is lower than the threshold then assert a signal to reset the chip.
Note: When the BOD_EN is enabled and the interrupt is asserted, the interrupt will be kept till the BOD_EN is set to 0. The interrupt for CPU can be blocked by disabling the NVIC in CPU for BOD interrupt or disable the interrupt source by disabling the BOD_EN and then re-enabling the BOD_EN function if the BOD function is required.</td></tr>
<tr><td>[2:1]</td><td>BODVL</td><td>Brown-out Detector Threshold Voltage Selection (Write Protect)
The default value is set by flash controller user configuration register CBOV (CONFIG0[22:21]).
00 = Reserved.
01 = Brown-out Detector threshold voltage is 2.7V.
10 = Brown-out Detector threshold voltage is 3.7V.
11 = Brown-out Detector function Disabled.</td></tr>
<tr><td>[0]</td><td>BODEN</td><td>Brown-out Detector Selection Extension (Initiated & Write-protected Bit)
The default value is set by flash controller user configuration register config0 bit[23].
If config0 bit[23] is set to 1, default value of BODEN is 0.
If config0 bit[23] is set to 0, default value of BODEN is 1.
0 = Brown-out detector threshold voltage is selected by the table defined in BODVL.
1 = Brown-out detector threshold voltage is selected by the table defined as below.
<table>
<tr><td>BODVL[1]</td><td>BODVL[0]</td><td>Brown-out voltage</td></tr>
<tr><td>1</td><td>1</td><td>4.4V</td></tr>
<tr><td>1</td><td>0</td><td>3.7V</td></tr>
<tr><td>0</td><td>1</td><td>2.7V</td></tr>
<tr><td>0</td><td>0</td><td>2.2V</td></tr>
</table></td></tr>
</table>

Multiple Function Port0 Control Register (SYS_P0_MFP)

Register	Offset	R/W	Description	Reset Value
SYS_P0_MFP	SYS_BA+0x30	R/W	P0 Multiple Function and Input Type Control Register	0x0000_0000

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

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:16]	**TYPE[n]**	**P0[7:0] Input Schmitt Trigger Function Enable Bits** 0 = P0[7:0] I/O input Schmitt Trigger function Disabled. 1 = P0[7:0] I/O input Schmitt Trigger function Enabled.
[15]	**ALT[7]**	**P0.7 Alternate Function Select Bit** Bits ALT[7] (SYS_P0_MFP[15]), and MFP[7] (SYS_P0_MFP[7]) determine the P0.7 function. (0, 0) = GPIO function is selected. (0, 1) = Reserved. (1, 0) = SPI0_CLK function is selected. (1, 1) = PWM0_CH0 function is selected.
[14]	**ALT[6]**	**P0.6 Alternate Function Select Bit** Bits ALT[6] (SYS_P0_MFP[14]), and MFP[6] (SYS_P0_MFP[6]) determine the P0.6 function. (0, 0) = GPIO function is selected. (0, 1) = Reserved. (1, 0) = SPI0_MISO function is selected. (1, 1) = PWM0_CH1 function is selected.
[13]	**ALT[5]**	**P0.5 Alternate Function Select Bit** Bits ALT[5] (SYS_P0_MFP[13]), and MFP[5] (SYS_P0_MFP[5]) determine the P0.5 function. (0, 0) = GPIO function is selected. (0, 1) = Reserved. (1, 0) = SPI0_MOSI function is selected. (1, 1) = PWM0_CH4 function is selected.

Bits	Description	
[12]	**ALT[4]**	**P0.4 Alternate Function Select Bit** Bits ALT[4] (SYS_P0_MFP[12]), and MFP[4] (SYS_P0_MFP[4]) determine the P0.4 function. (0, 0) = GPIO function is selected. (0, 1) = Reserved. (1, 0) = SPI0_SS function is selected. (1, 1) = PWM0_CH5 function is selected.
[11:10]	**Reserved**	Reserved.
[9]	**ALT[1]**	**P0.1 Alternate Function Select Bit** Bits ALT[1] (SYS_P0_MFP[9]), and MFP[1] (SYS_P0_MFP[1]) determine the P0.1 function. (0, 0) = GPIO function is selected. (0, 1) = SPI0_SS function is selected. (1, 0) = UART0_nRTS function is selected. (1, 1) = UART0_RXD function is selected.
[8]	**ALT[0]**	**P0.0 Alternate Function Select Bit** Bits ALT[0] (SYS_P0_MFP[8]), and MFP[0] (SYS_P0_MFP[0]) determine the P0.0 function. (0, 0) = GPIO function is selected. (0, 1) = Reserved. (1, 0) = UART0_nCTS function is selected. (1, 1) = UART0_TXD function is selected.
[7:0]	**MFP[7:0]**	**P0 Multiple Function Select Bit** The pin function of P0 depends on MFP and ALT. Refer to ALT Description for details.

Multiple Function Port1 Control Register (SYS_P1_MFP)

Register	Offset	R/W	Description	Reset Value
SYS_P1_MFP	SYS_BA+0x34	R/W	P1 Multiple Function and Input Type Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved			P14EXT	P13EXT	P12EXT	Reserved	
23	22	21	20	19	18	17	16
TYPE							
15	14	13	12	11	10	9	8
ALT							
7	6	5	4	3	2	1	0
MFP							

Bits	Description	
[31:29]	**Reserved**	Reserved.
[28]	**P14EXT**	**P1.4 Alternate Function Selection Extension** Bits P14EXT (SYS_P1_MFP[28]), ALT[4] (SYS_P1_MFP[12]), and MFP[4] (SYS_P1_MFP[4]) determine the P1.4 function. (0, 0, 0) = GPIO function is selected. (0, 0, 1) = ADC_CH4 function is selected. (0, 1, 0) = UART1_RXD function is selected. (0, 1, 1) = ACMP0_N function is selected. (1, 0, 0) = PWM0_CH4 function is selected.
[27]	**P13EXT**	**P1.3 Alternate Function Selection Extension** Bits P13EXT (SYS_P1_MFP[27]), ALT[3] (SYS_P1_MFP[11]), and MFP[3] (SYS_P1_MFP[3]) determine the P1.3 function. (0, 0, 0) = GPIO function is selected. (0, 0, 1) = ADC_CH3 function is selected. (0, 1, 0) = UART0_TXD function is selected. (0, 1, 1) = ACMP0_P3 function is selected. (1, 0, 0) = PWM0_CH1 function is selected.
[26]	**P12EXT**	**P1.2 Alternate Function Selection Extension** Bits P12EXT (SYS_P1_MFP[26]), ALT[2] (SYS_P1_MFP[10]), and MFP[2] (SYS_P1_MFP[2]) determine the P1.2 function. (0, 0, 0) = GPIO function is selected. (0, 0, 1) = ADC_CH2 function is selected. (0, 1, 0) = UART0_RXD function is selected. (0, 1, 1) = ACMP0_P2 function is selected. (1, 0, 0) = PWM0_CH0 function is selected.
[25:24]	**Reserved**	Reserved.

Bits	Description	
[23:16]	**TYPE[n]**	**P1[7:0] Input Schmitt Trigger Function Enable Bit** 0 = P1[7:0] I/O input Schmitt Trigger function Disabled. 1 = P1[7:0] I/O input Schmitt Trigger function Enabled.
[15:14]	**Reserved**	Reserved.
[13]	**ALT[5]**	**P1.5 Alternate Function Select Bit** Bits ALT[5] (SYS_P1_MFP[13]), and MFP[5] (SYS_P1_MFP[5]) determine the P1.5 function. (0, 0) = GPIO function is selected. (0, 1) = ADC_CH5 function is selected. (1, 0) = UART1_TXD function is selected. (1, 1) = ACMP0_P0 function is selected.
[12]	**ALT[4]**	**P1.4 Alternate Function Select Bit** Bits P14EXT (SYS_P1_MFP[28]), ALT[4] (SYS_P1_MFP[12]), and MFP[4] (SYS_P1_MFP[4]) determine the P1.4 function. (0, 0, 0) = GPIO function is selected. (0, 0, 1) = ADC_CH4 function is selected. (0, 1, 0) = UART1_RXD function is selected. (0, 1, 1) = ACMP0_N function is selected. (1, 0, 0) = PWM0_CH4 function is selected.
[11]	**ALT[3]**	**P1.3 Alternate Function Select Bit** Bits P13EXT (SYS_P1_MFP[27]), ALT[3] (SYS_P1_MFP[11]), and MFP[3] (SYS_P1_MFP[3]) determine the P1.3 function. (0, 0, 0) = GPIO function is selected. (0, 0, 1) = ADC_CH3 function is selected. (0, 1, 0) = UART0_TXD function is selected. (0, 1, 1) = ACMP0_P3 function is selected. (1, 0, 0) = PWM0_CH1 function is selected.
[10]	**ALT[2]**	**P1.2 Alternate Function Select Bit** Bits P12EXT (SYS_P1_MFP[26]), ALT[2] (SYS_P1_MFP[10]), and MFP[2] (SYS_P1_MFP[2]) determine the P1.2 function. (0, 0, 0) = GPIO function is selected. (0, 0, 1) = ADC_CH2 function is selected. (0, 1, 0) = UART0_RXD function is selected. (0, 1, 1) = ACMP0_P2 function is selected. (1, 0, 0) = PWM0_CH0 function is selected.
[9]	**Reserved**	Reserved.
[8]	**ALT[0]**	**P1.0 Alternate Function Select Bit** Bits ALT[0] (SYS_P1_MFP[8]), and MFP[0] (SYS_P1_MFP[0]) determine the P1.0 function. (0, 0) = GPIO function is selected. (0, 1) = ADC_CH1 function is selected. (1, 0) = Reserved. (1, 1) = ACMP0_P1 function is selected.
[7:0]	**MFP[7:0]**	**P1 Multiple Function Select Bit** The pin function of P1 depends on MFP and ALT. Refer to ALT Description for details.

Multiple Function Port2 Control Register (SYS_P2_MFP)

Register	Offset	R/W	Description	Reset Value
SYS_P2_MFP	SYS_BA+0x38	R/W	P2 Multiple Function and Input Type Control Register	0x0000_0000

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

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:16]	**TYPE[n]**	**P2[7:0] Input Schmitt Trigger Function Enable Bits** 0 = P2[7:0] I/O input Schmitt Trigger function Disabled. 1 = P2[7:0] I/O input Schmitt Trigger function Enabled.
[15]	**Reserved**	Reserved.
[14]	**ALT[6]**	**P2.6 Alternate Function Select Bit** Bits ALT[6] (SYS_P2_MFP[14]), and MFP[6] (SYS_P2_MFP[6]) determine the P2.6 function. (0, 0) = GPIO function is selected. (0, 1) = Reserved. (1, 0) = PWM0_CH4 function is selected. (1, 1) = ACMP1_O function is selected.
[13]	**ALT[5]**	**P2.5 Alternate Function Select Bit** Bits ALT[5] (SYS_P2_MFP[13]), and MFP[5] (SYS_P2_MFP[5]) determine the P2.5 function. (0, 0) = GPIO function is selected. (0, 1) = UART1_TXD function is selected. (1, 0) = PWM0_CH3 function is selected. (1, 1) = Reserved.
[12]	**ALT[4]**	**P2.4 Alternate Function Select Bit** Bits ALT[4] (SYS_P2_MFP[12]), and MFP[4] (SYS_P2_MFP[4]) determine the P2.4 function. (0, 0) = GPIO function is selected. (0, 1) = UART1_RXD function is selected. (1, 0) = PWM0_CH2 function is selected. (1, 1) = Reserved.

Bits	Description	
[11]	**ALT[3]**	**P2.3 Alternate Function Select Bit** Bits ALT[3] (SYS_P2_MFP[11]), and MFP[3] (SYS_P2_MFP[3]) determine the P2.3 function. (0, 0) = GPIO function is selected. (0, 1) = Reserved. (1, 0) = PWM0_CH1 function is selected. (1, 1) = I2C1_SDA1 function is selected.
[10]	**ALT[2]**	**P2.2 Alternate Function Select Bit** Bits ALT[2] (SYS_P2_MFP[10]), and MFP[2] (SYS_P2_MFP[2]) determine the P2.2 function. (0, 0) = GPIO function is selected. (0, 1) = Reserved. (1, 0) = PWM0_CH0 function is selected. (1, 1) = I2C1_SCL function is selected.
[9:8]	**Reserved**	Reserved.
[7:0]	**MFP[7:0]**	**P2 Multiple Function Select Bit** The pin function of P2 depends on MFP and ALT. Refer to ALT Description for details.

Multiple Function Port3 Control Register (SYS_P3_MFP)

Register	Offset	R/W	Description	Reset Value
SYS_P3_MFP	SYS_BA+0x3C	R/W	P3 Multiple Function and Input Type Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved					P32EXT	Reserved	
23	22	21	20	19	18	17	16
TYPE							
15	14	13	12	11	10	9	8
ALT							
7	6	5	4	3	2	1	0
MFP							

Bits	Description	
[31:27]	**Reserved**	Reserved.
[26]	**P32EXT**	**P3.2 Alternate Function Selection Extension** Bits P32EXT (SYS_P3_MFP[26]), ALT[2] (SYS_P3_MFP[10]), and MFP[2] (SYS_P3_MFP[2]) determine the P3.2 function. (0, 0, 0) = GPIO function is selected. (0, 0, 1) = INT0 function is selected. (0, 1, 0) = TM0_EXT function is selected. (0, 1, 1) = STADC function is selected. (1, 0, 0) = ACMP1_P1 function is selected.
[25:24]	**Reserved**	Reserved.
[23:16]	**TYPE[n]**	**P3[7:0] Input Schmitt Trigger Function Enable Bits** 0 = P3[7:0] I/O input Schmitt Trigger function Disabled. 1 = P3[7:0] I/O input Schmitt Trigger function Enabled.
[15]	**Reserved**	Reserved.
[14]	**ALT[6]**	**P3.6 Alternate Function Select Bit** Bits ALT[6] (SYS_P3_MFP[14]), and MFP[6] (SYS_P3_MFP[6]) determine the P3.6 function. (0, 0) = GPIO function is selected. (0, 1) = TM1_EXT function is selected. (1, 0) = CLKO function is selected. (1, 1) = ACMP0_O function is selected.

Bits	Description	
[13]	**ALT[5]**	**P3.5 Alternate Function Select Bit** Bits ALT[5] (SYS_P3_MFP[13]), and MFP[5] (SYS_P3_MFP[5]) determine the P3.5 function. (0, 0) = GPIO function is selected. (0, 1) = TM1_CNT_OUTfunction is selected. (1, 0) = I2C0_SCL function is selected. (1, 1) = ACMP1_P3 function is selected.
[12]	**ALT[4]**	**P3.4 Alternate Function Select Bit** Bits ALT[4] (SYS_P3_MFP[12]), and MFP[4] (SYS_P3_MFP[4]) determine the P3.4 function. (0, 0) = GPIO function is selected. (0, 1) = TM0_CNT_OUT function is selected. (1, 0) = I2C0_SDA function is selected. (1, 1) = ACMP1_P2 function is selected.
[11]	**Reserved**	Reserved.
[10]	**ALT[2]**	**P3.2 Alternate Function Select Bit** Bits P32EXT (SYS_P3_MFP[26]), ALT[2] (SYS_P3_MFP[10]), and MFP[2] (SYS_P3_MFP[2]) determine the P3.2 function. (0, 0, 0) = GPIO function is selected. (0, 0, 1) = INT0 function is selected. (0, 1, 0) = TM0_EXT function is selected. (0, 1, 1) = STADC function is selected. (1, 0, 0) = ACMP1_P1 function is selected.
[9]	**ALT[1]**	**P3.1 Alternate Function Select Bit** The pin function of P3.1 depends on P3_MFP[1] and P3_ALT[1]. Bits ALT[1] (SYS_P3_MFP[9]), and MFP[1] (SYS_P3_MFP[1]) determine the P3.1 function. (0, 0) = GPIO function is selected. (0, 1) = Reserved. (1, 0) = ACMP1_P0 function is selected. (1, 1) = ADC_CH7 function is selected.
[8]	**ALT[0]**	**P3.0 Alternate Function Select Bit** Bits ALT[0] (SYS_P3_MFP[8]), and MFP[0] (SYS_P3_MFP[0]) determine the P3.0 function. (0, 0) = GPIO function is selected. (0, 1) = Reserved. (1, 0) = ACMP1_N function is selected. (1, 1) = ADC_CH6 function is selected.
[7:0]	**MFP[7:0]**	**P3 Multiple Function Select Bits** The pin function of P3 depends on MFP and ALT. Refer to ALT Description for details.

Multiple Function Port4 Control Register (SYS_P4_MFP)

Register	Offset	R/W	Description	Reset Value
SYS_P4_MFP	SYS_BA+0x40	R/W	P4 Multiple Function and Input Type Control Register	0x0000_00C0

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

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:16]	**TYPE[n]**	**P4[7:0] Input Schmitt Trigger Function Enable Bits** 0 = P4[7:0] I/O input Schmitt Trigger function Disabled. 1 = P4[7:0] I/O input Schmitt Trigger function Enabled.
[15]	**ALT[7]**	**P4.7 Alternate Function Select Bit** Bits ALT[7] (SYS_P4_MFP[15]), and MFP[7] (SYS_P4_MFP[7]) determine the P4.7 function. (0, 0) = GPIO function is selected. (0, 1) = ICE_DAT function is selected. (1, 0) = UART1_TXD function is selected. (1, 1) = Reserved.
[14]	**ALT[6]**	**P4.6 Alternate Function Select Bit** Bits ALT[6] (SYS_P4_MFP[14]), and MFP[6] (SYS_P4_MFP[6]) determine the P4.6 function. (0, 0) = GPIO function is selected. (0, 1) = ICE_CLK function is selected. (1, 0) = UART1_RXD function is selected. (1, 1) = Reserved.
[13:8]	**Reserved**	Reserved.
[7:0]	**MFP[7:0]**	**P4 Multiple Function Select Bits** The pin function of P4 depends on MFP and ALT. Refer to ALT Description for details.

Multiple Function Port5 Control Register (SYS_P5_MFP)

Register	Offset	R/W	Description	Reset Value
SYS_P5_MFP	SYS_BA+0x44	R/W	P5 Multiple Function and Input Type Control Register	0x0000_0000

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

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:16]	**TYPE[n]**	**P5[7:0] Input Schmitt Trigger Function Enable Bits** 0 = P5[7:0] I/O input Schmitt Trigger function Disabled. 1 = P5[7:0] I/O input Schmitt Trigger function Enabled.
[15:14]	**Reserved**	Reserved.
[13]	**ALT[5]**	**P5.5 Alternate Function Select Bit** Bits ALT[5] (SYS_P5_MFP[13]), and MFP[5] (SYS_P5_MFP[5]) determine the P5.5 function. (0, 0) = GPIO function is selected. Others = Reserved.
[12]	**ALT[4]**	**P5.4 Alternate Function Select Bit** Bits ALT[4] (SYS_P5_MFP[12]), and MFP[4] (SYS_P5_MFP[4]) determine the P5.4 function. (0, 0) = GPIO function is selected. Others = Reserved.
[11]	**ALT[3]**	**P5.3 Alternate Function Select Bit** Bits ALT[3] (SYS_P5_MFP[11]), and MFP[3] (SYS_P5_MFP[3]) determine the P5.3 function. (0, 0) = GPIO function is selected. (0, 1) = ADC_CH0 function is selected. Others = Reserved.
[10]	**ALT[2]**	**P5.2 Alternate Function Select Bit** Bits ALT[2] (SYS_P5_MFP[10]), and MFP[2] (SYS_P5_MFP[2]) determine the P5.2 function. (0, 0) = GPIO function is selected. (0, 1) = INT1 function is selected. Others = Reserved.

Bits	Description	
[9]	**ALT[1]**	**P5.1 Alternate Function Select Bit** Bits ALT[1] (SYS_P5_MFP[9]), and MFP[1] (SYS_P5_MFP[1]) determine the P5.1 function. (0, 0) = GPIO function is selected. (0, 1) = XT1_OUT function is selected. (1, 0) = I2C1_SCL1 function is selected. (1, 1) = UART0_RXD function is selected. **Note:** To enable external XTAL function, the CLK_PWRCTL bit [1:0] (XTLEN), external HXT or LXT crystal oscillator control register must also be set.
[8]	**ALT[0]**	**P5.0 Alternate Function Select Bit** The pin function of P5.0 depends on MFP[0] and ALT[0]. Bits ALT[0] (SYS_P5_MFP[8]), and MFP[0] (SYS_P5_MFP[0]) determine the P5.0 function. (0, 0) = GPIO function is selected. (0, 1) = XT1_IN function is selected. (1, 0) = I2C1_SDA1 function is selected. (1, 1) = UART0_TXD function is selected. **Note:** To enable external XTAL function, the CLK_PWRCTL bit [1:0] (XTLEN), external HXT or LXT crystal oscillator control register must also be set.
[7:0]	**MFP[7:0]**	**P5 Multiple Function Select Bits** The pin function of P5 depends on MFP and ALT. Refer to ALT Description for details.

HIRC Trim Control Register (SYS_IRCTCTL)

Register	Offset	R/W	Description	Reset Value
SYS_IRCTCTL	SYS_BA+0x80	R/W	HIRC 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							
7	6	5	4	3	2	1	0
Reserved		LOOPSEL		Reserved			FREQSEL

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5:4]	**LOOPSEL**	**Trim Calculation Loop** This field defines trim value calculation based on the number of LXT clock. For example, if LOOPSEL is set as “00”, auto trim circuit will calculate trim value based on the average frequency difference in 4 LXT clocks. This field also defines how many times the auto trim circuit will try to update the HIRC trim value before the frequency of HIRC is locked. Once the HIRC is locked, the internal trim value update counter will be reset. If the trim value update counter reaches this limitation value and frequency of HIRC is still not locked, the auto trim operation will be disabled and FREQSEL will be cleared to 0. 00 = Trim value calculation is based on average difference in 4 LXT clock and trim retry count limitation is 64. 01 = Trim value calculation is based on average difference in 8 LXT clock and trim retry count limitation is 128. 10 = Trim value calculation is based on average difference in 16 LXT clock and trim retry count limitation is 256. 11 = Trim value calculation is based on average difference in 32 LXT clock and trim retry count limitation is 512.
[3:1]	**Reserved**	Reserved.
[0]	**FREQSEL**	**Trim Frequency Select Bit** This bit is to enable the HIRC auto trim. When setting this bit to 1, the HIRC auto trim function will trim HIRC to 22.1184 MHz automatically based on the LXT reference clock. During auto trim operation, if LXT clock error is detected or trim retry limitation count reached, this field will be cleared to 0 automatically. 0 = HIRC auto trim function Disabled. 1 = HIRC auto trim function Enabled and HIRC trimmed to 22.1184 MHz.

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							
7	6	5	4	3	2	1	0
Reserved					CLKEIEN	TFAILIEN	Reserved

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2]	**CLKEIEN**	**LXT Clock Error Interrupt Enable Bit** This bit controls if CPU could get an interrupt while LXT clock is inaccurate during auto trim operation. If this bit is high, and CLKERRIF (SYS_IRCTISTS[2]) is set during auto trim operation, an interrupt will be triggered to notify the LXT clock frequency is inaccurate. 0 = CLKERRIF (SYS_IRCTISTS[2]) status Disabled to trigger an interrupt to CPU. 1 = CLKERRIF (SYS_IRCTISTS[2]) status Enabled to trigger an interrupt to CPU.
[1]	**TFAILIEN**	**Trim Failure Interrupt Enable Bit** This bit controls if an interrupt will be triggered while HIRC trim value update limitation count is reached and HIRC frequency is still not locked on target frequency set by FREQSEL (SYS_IRCTCTL[1:0]). If this bit is high and TFAILIF (SYS_IRCTISTS[1]) is set during auto trim operation, an interrupt will be triggered to notify that HIRC trim value update limitation count is reached. 0 = TFAILIF (SYS_IRCTISTS[1]) status Disabled to trigger an interrupt to CPU. 1 = TFAILIF (SYS_IRCTISTS[1]) status Enabled to trigger an interrupt to CPU.
[0]	**Reserved**	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							
7	6	5	4	3	2	1	0
Reserved					CLKERRIF	TFAILIF	FREQLOCK

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2]	**CLKERRIF**	**LXT Clock Error Interrupt Status** This bit indicates that LXT clock frequency is inaccuracy. Once this bit is set, the auto trim operation stopped and FREQSEL (SYS_IRCTCTL[0]) will be cleared to 0 by hardware automatically. If this bit is set and CLKEIEN (SYS_IRCTIEN[2]) is high, an interrupt will be triggered to notify the LXT clock frequency is inaccuracy. Software can write 1 to clear this bit to 0. 0 = LXT clock frequency is accuracy. 1 = LXT clock frequency is inaccuracy.
[1]	**TFAILIF**	**Trim Failure Interrupt Status** This bit indicates that HIRC trim value update limitation count reached and HIRC clock frequency still doesn't lock. Once this bit is set, the auto trim operation stopped and FREQSEL (SYS_IRCTCTL[1:0]) will be cleared to 0 by hardware automatically. If this bit is set and TFAILIEN (SYS_IRCTIEN[1]) is high, an interrupt will be triggered to notify that HIRC trim value update limitation count was reached. Software can write 1 to clear this bit to 0. 0 = Trim value update limitation count is not reached. 1 = Trim value update limitation count is reached and HIRC frequency is still not locked.
[0]	**FREQLOCK**	**HIRC Frequency Lock Status** This bit indicates the HIRC frequency locked in 22.1184 MHz. This is a read only status bit and doesn't trigger any interrupt.

Register Write-Protection Control Register (SYS_REGLCTL)

Some of the system control registers need to be protected to avoid inadvertent write and disturb the chip operation. These system control registers are protected after the power on reset till user to disable register protection. For user to programs 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 0x59, 0x16, 0x88 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 bit 0, 1 is protection disable, 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 the register protection.

Write this register to disable/enable register protection, and reading it to get the REGLCTL status.

Register	Offset	R/W	Description	Reset Value
SYS_REGLCTL	SYS_BA+0x100	R/W	Register Write-Protection Control Register	0x0000_0000

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

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**REGLCTL**	**Register Write-protection Code (Write Only)** Some registers have write-protection function. Writing these registers have to disable the protected function by writing the sequence value 0x59, 0x16, 0x88 to this field. After this sequence is completed, the REGLCTL bit will be set to 1 and write-protection registers can be normal write.
[0]	**REGLCTL**	**Register Write-protection Disable Index (Read Only)** 0 = Write-protection Enabled for writing protected registers. Any write to the protected register is ignored. 1 = Write-protection Disabled for writing protected registers. Please refer to 6.2.6 Register Protection.**Note:** The bits which are write-protected will be noted as" **(Write Protect)**" beside the description.

6.2.10 System Timer (SysTick)

The Cortex®-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 zero, and reload (wrap) to the value in the SysTick Reload Value Register (SYST_RVR) on the next clock edge, and then decrement on subsequent clocks. When the counter transitions to zero, the COUNTFLAG status bit is set. The COUNTFLAG bit clears on reads.

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

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

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

6.2.10.1 System Timer Control Register Map

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

Register	Offset	R/W	Description	Reset Value
SCS Base Address: **SCS_BA = 0xE000_E000**				
SYST_CSR	SCS_BA+0x10	R/W	SysTick Control and Status Register	0x0000_0000
SYST_RVR	SCS_BA+0x14	R/W	SysTick Reload Value Register	0x00XX_XXXX
SYST_CVR	SCS_BA+0x18	R/W	SysTick Current Value Register	0x00XX_XXXX

6.2.10.2 System Timer Control Register

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**	**System Tick Counter Flag** Returns 1 if timer counted to 0 since last time this register was read. COUNTFLAG is set by a count transition from 1 to 0. COUNTFLAG is cleared on read or by a write to the Current Value register.
[15:3]	**Reserved**	Reserved.
[2]	**CLKSRC**	**System Tick Clock Source Select Bit** 0 = Clock source is optional, refer to STCLKSEL. 1 = Core clock used for SysTick timer.
[1]	**TICKINT**	**System Tick Interrupt Enable Bit** 0 = Counting down to 0 does not cause the SysTick exception to be pended. Software can use COUNTFLAG to determine if a count to 0 has occurred. 1 = Counting down to 0 will cause the SysTick exception to be pended. Clearing the SysTick Current Value register by a register write in software will not cause SysTick to be pended.
[0]	**ENABLE**	**System Tick Counter Enable Bit** 0 = Counter Disabled. 1 = Counter Enabled and will operate in a multi-shot manner.

SysTick Reload Value Register (SYST_RVR)

Register	Offset	R/W	Description	Reset Value
SYST_RVR	SCS_BA+0x14	R/W	SysTick Reload Value Register	0x00XX_XXXX

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

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:0]	**RELOAD**	**System Tick Reload Value** 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	0x00XX_XXXX

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

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

6.2.11 Nested Vectored Interrupt Controller (NVIC)

6.2.11.1 Overview

The Cortex®-M0 CPU 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 core and provides following features.

6.2.11.2 Features

- Nested and Vectored interrupt support
- Automatic processor state saving and restoration
- Dynamic priority change
- 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® Cortex®-M0 Technical Reference Manual” and “ARM® v6-M Architecture Reference Manual”.

6.2.11.3 Exception Model and System Interrupt Map

The following table lists the exception model supported by NuMicro® Mini58 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 the priority 0 is treated as the fourth priority on the system, after three system exceptions “Reset”, “NMI” and “Hard Fault”.

Exception Name	Vector Number	Priority
Reset	1	-3
NMI	2	-2
Hard Fault	3	-1
Reserved	4 ~ 10	Reserved
SVCall	11	Configurable
Reserved	12 ~ 13	Reserved
PendSV	14	Configurable
SysTick	15	Configurable
Interrupt (IRQ0 ~ IRQ31)	16 ~ 47	Configurable

Table 6.2-7 Exception Model

Exception Number	Interrupt Number (Bit In Interrupt Registers)	Interrupt Name	Source Module	Interrupt Description	Power-Down Wake-Up
1 ~ 15	-	**-**	-	System exceptions	-
16	0	**BODOUT**	Brown-out	Brown-out low voltage detected interrupt	Yes
17	1	**WDT_INT**	WDT	Watchdog Timer interrupt	Yes
18	2	**EINT0**	GPIO	External signal interrupt from P3.2 pin	Yes
19	3	**EINT1**	GPIO	External signal interrupt from P5.2 pin	Yes
20	4	**GP0/1_INT**	GPIO	External signal interrupt from GPIO group P0~P1	Yes
21	5	**GP2/3/4_INT**	GPIO	External signal interrupt from GPIO group P2~P4 except P3.2	Yes
22	6	**PWM_INT**	PWM	PWM interrupt	No
23	7	**BRAKE_INT**	PWM	PWM Brake interrupt	No
24	8	**TMR0_INT**	TMR0	Timer 0 interrupt	Yes
25	9	**TMR1_INT**	TMR1	Timer 1 interrupt	Yes
26 ~ 27	10 ~ 11	**-**	-	-	
28	12	**UART0_INT**	UART0	UART0 interrupt	Yes
29	13	**UART1_INT**	UART1	UART1 interrupt	Yes
30	14	**SPI_INT**	SPI	SPI interrupt	No
31	15	**-**	-	-	
32	16	**GP5_INT**	GPIO	External signal interrupt from GPIO group P5 except P5.2	Yes
33	17	**HIRC_TRIM_INT**	HIRC	HIRC trim interrupt	No
34	18	**I2C0_INT**	I^2C0	I^2C0 interrupt	Yes

Exception Number	Interrupt Number (Bit In Interrupt Registers)	Interrupt Name	Source Module	Interrupt Description	Power-Down Wake-Up
35	19	**I2C1_INT**	I^2C1	I^2C1 interrupt	No
36 ~ 40	20 ~ 24	**-**	-	-	
41	25	**ACMP_INT**	ACMP	Analog Comparator 0 or Comparator 1 interrupt	Yes
42 ~ 43	26 ~ 27	**-**	-	-	
44	28	**PWRWU_INT**	CLKC	Clock controller interrupt for chip wake-up from Power-down state	Yes
45	29	**ADC_INT**	ADC	ADC interrupt	No
46 ~ 47	30 ~ 31	**-**	-	-	

Table 6.2-8 System Interrupt Map Vector Table

6.2.11.4 Vector Table

When an interrupt is accepted, the processor will automatically fetch the starting address of the interrupt service routine (ISR) from a vector table in memory. For ARMv6-M, the vector table based address is fixed at 0x00000000. The vector table contains the initialization value for the stack pointer on reset, and the entry point addresses for all exception handlers. The vector number on previous page defines the order of entries in the vector table associated with the exception handler entry as illustrated in previous section.

Vector Table Word Offset (Bytes)	Description
0x00	Initial Stack Pointer Value
Exception Number * 0x04	Exception Entry Pointer using that Exception Number

Table 6.2-9 Vector Table Format

6.2.11.5 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 be activated. 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.

6.2.11.6 NVIC Control Register Map

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

Register	Offset	R/W	Description	Reset Value
SCS Base Address: **SCS_BA = 0xE000_E000**				
NVIC_ISER	SCS_BA+0x100	R/W	IRQ0 ~ IRQ31 Set-Enable Control Register	0x0000_0000
NVIC_ICER	SCS_BA+0x180	R/W	IRQ0 ~ IRQ31 Clear-Enable Control Register	0x0000_0000
NVIC_ISPR	SCS_BA+0x200	R/W	IRQ0 ~ IRQ31 Set-Pending Control Register	0x0000_0000
NVIC_ICPR	SCS_BA+0x280	R/W	IRQ0 ~ IRQ31 Clear-Pending Control Register	0x0000_0000
NVIC_IPR0	SCS_BA+0x400	R/W	IRQ0 ~ IRQ3 Interrupt Priority Control Register	0x0000_0000
NVIC_IPR1	SCS_BA+0x404	R/W	IRQ4 ~ IRQ7 Interrupt Priority Control Register	0x0000_0000
NVIC_IPR2	SCS_BA+0x408	R/W	IRQ8 ~ IRQ11 Interrupt Priority Control Register	0x0000_0000
NVIC_IPR3	SCS_BA+0x40C	R/W	IRQ12 ~ IRQ15 Interrupt Priority Control Register	0x0000_0000
NVIC_IPR4	SCS_BA+0x410	R/W	IRQ16 ~ IRQ19 Interrupt Priority Control Register	0x0000_0000
NVIC_IPR5	SCS_BA+0x414	R/W	IRQ20 ~ IRQ23 Interrupt Priority Control Register	0x0000_0000
NVIC_IPR6	SCS_BA+0x418	R/W	IRQ24 ~ IRQ27 Interrupt Priority Control Register	0x0000_0000
NVIC_IPR7	SCS_BA+0x41C	R/W	IRQ28 ~ IRQ31 Interrupt 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]	**SETENA**	**Interrupt Enable Bits** 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. **Note:** Read value indicates the current enable status.

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]	**CLRENA**	**Interrupt Disable Bits** 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 is Disabled. 1 = Associated interrupt status is Enabled. **Note:** Read value indicates the current enable status.

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
SETPEND							
15	14	13	12	11	10	9	8
SETPEND							
7	6	5	4	3	2	1	0
SETPEND							

Bits	Description	
[31:0]	**SETPEND**	**Set Interrupt Pending Bits** 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. **Note:** Read value indicates the current pending status.

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]	**CLRPEND**	**Clear Interrupt Pending Bits** 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 is not in pending status. 1 = Associated interrupt is in pending status. **Note:** Read value indicates the current pending status.

IRQ0 ~ IRQ3 Interrupt Priority Register (NVIC_IPR0)

Register	Offset	R/W	Description	Reset Value
NVIC_IPR0	SCS_BA+0x400	R/W	IRQ0 ~ IRQ3 Interrupt 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**	**Priority of IRQ3** 0 denotes the highest priority and 3 denotes the lowest priority.
[29:24]	**Reserved**	Reserved.
[23:22]	**PRI_2**	**Priority of IRQ2** 0 denotes the highest priority and 3 denotes the lowest priority.
[21:16]	**Reserved**	Reserved.
[15:14]	**PRI_1**	**Priority of IRQ1** 0 denotes the highest priority and 3 denotes the lowest priority.
[13:8]	**Reserved**	Reserved.
[7:6]	**PRI_0**	**Priority of IRQ0** 0 denotes the highest priority and 3 denotes the lowest priority.
[5:0]	**Reserved**	Reserved.

IRQ4 ~ IRQ7 Interrupt Priority Register (NVIC_IPR1)

Register	Offset	R/W	Description	Reset Value
NVIC_IPR1	SCS_BA+0x404	R/W	IRQ4 ~ IRQ7 Interrupt 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**	**Priority of IRQ7** 0 denotes the highest priority and 3 denotes the lowest priority.
[29:24]	**Reserved**	Reserved.
[23:22]	**PRI_6**	**Priority of IRQ6** 0 denotes the highest priority and 3 denotes the lowest priority.
[21:16]	**Reserved**	Reserved.
[15:14]	**PRI_5**	**Priority of IRQ5** 0 denotes the highest priority and 3 denotes the lowest priority.
[13:8]	**Reserved**	Reserved.
[7:6]	**PRI_4**	**Priority of IRQ4** 0 denotes the highest priority and 3 denotes the lowest priority.
[5:0]	**Reserved**	Reserved.

IRQ8 ~ IRQ11 Interrupt Priority Register (NVIC_IPR2)

Register	Offset	R/W	Description	Reset Value
NVIC_IPR2	SCS_BA+0x408	R/W	IRQ8 ~ IRQ11 Interrupt 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**	**Priority of IRQ11** 0 denotes the highest priority and 3 denotes the lowest priority.
[29:24]	**Reserved**	Reserved.
[23:22]	**PRI_10**	**Priority of IRQ10** 0 denotes the highest priority and 3 denotes the lowest priority.
[21:16]	**Reserved**	Reserved.
[15:14]	**PRI_9**	**Priority of IRQ9** 0 denotes the highest priority and 3 denotes the lowest priority.
[13:8]	**Reserved**	Reserved.
[7:6]	**PRI_8**	**Priority of IRQ8** 0 denotes the highest priority and 3 denotes the lowest priority.
[5:0]	**Reserved**	Reserved.

IRQ12 ~ IRQ15 Interrupt Priority Register (NVIC_IPR3)

Register	Offset	R/W	Description	Reset Value
NVIC_IPR3	SCS_BA+0x40C	R/W	IRQ12 ~ IRQ15 Interrupt 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]	**PRI_15**	**Priority of IRQ15** 0 denotes the highest priority and 3 denotes the lowest priority.
[29:24]	**Reserved**	Reserved.
[23:22]	**PRI_14**	**Priority of IRQ14** 0 denotes the highest priority and 3 denotes the lowest priority.
[21:16]	**Reserved**	Reserved.
[15:14]	**PRI_13**	**Priority of IRQ13** 0 denotes the highest priority and 3 denotes the lowest priority.
[13:8]	**Reserved**	Reserved.
[7:6]	**PRI_12**	**Priority of IRQ12** 0 denotes the highest priority and 3 denotes the lowest priority.
[5:0]	**Reserved**	Reserved.

IRQ16 ~ IRQ19 Interrupt Priority Register (NVIC_IPR4)

Register	Offset	R/W	Description	Reset Value
NVIC_IPR4	SCS_BA+0x410	R/W	IRQ16 ~ IRQ19 Interrupt 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]	**PRI_19**	**Priority of IRQ19** 0 denotes the highest priority and 3 denotes the lowest priority.
[29:24]	**Reserved**	Reserved.
[23:22]	**PRI_18**	**Priority of IRQ18** 0 denotes the highest priority and 3 denotes the lowest priority.
[21:16]	**Reserved**	Reserved.
[15:14]	**PRI_17**	**Priority of IRQ17** 0 denotes the highest priority and 3 denotes the lowest priority.
[13:8]	**Reserved**	Reserved.
[7:6]	**PRI_16**	**Priority of IRQ16** 0 denotes the highest priority and 3 denotes the lowest priority.
[5:0]	**Reserved**	Reserved.

IRQ20 ~ IRQ23 Interrupt Priority Register (NVIC_IPR5)

Register	Offset	R/W	Description	Reset Value
NVIC_IPR5	SCS_BA+0x414	R/W	IRQ20 ~ IRQ23 Interrupt 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]	**PRI_23**	**Priority of IRQ23** 0 denotes the highest priority and 3 denotes the lowest priority.
[29:24]	**Reserved**	Reserved.
[23:22]	**PRI_22**	**Priority of IRQ22** 0 denotes the highest priority and 3 denotes the lowest priority.
[21:16]	**Reserved**	Reserved.
[15:14]	**PRI_21**	**Priority of IRQ21** 0 denotes the highest priority and 3 denotes the lowest priority.
[13:8]	**Reserved**	Reserved.
[7:6]	**PRI_20**	**Priority of IRQ20** 0 denotes the highest priority and 3 denotes the lowest priority.
[5:0]	**Reserved**	Reserved.

IRQ24 ~ IRQ27 Interrupt Priority Register (NVIC_IPR6)

Register	Offset	R/W	Description	Reset Value
NVIC_IPR6	SCS_BA+0x418	R/W	IRQ24 ~ IRQ27 Interrupt 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**	**Priority of IRQ27** 0 denotes the highest priority and 3 denotes the lowest priority.
[29:24]	**Reserved**	Reserved.
[23:22]	**PRI_26**	**Priority of IRQ26** 0 denotes the highest priority and 3 denotes the lowest priority.
[21:16]	**Reserved**	Reserved.
[15:14]	**PRI_25**	**Priority of IRQ25** 0 denotes the highest priority and 3 denotes the lowest priority.
[13:8]	**Reserved**	Reserved.
[7:6]	**PRI_24**	**Priority of IRQ24** 0 denotes the highest priority and 3 denotes the lowest priority.
[5:0]	**Reserved**	Reserved.

IRQ28 ~ IRQ31 Interrupt Priority Register (NVIC_IPR7)

Register	Offset	R/W	Description	Reset Value
NVIC_IPR7	SCS_BA+0x41C	R/W	IRQ28 ~ IRQ31 Interrupt 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]	**PRI_31**	**Priority of IRQ31** 0 denotes the highest priority and 3 denotes the lowest priority.
[29:24]	**Reserved**	Reserved.
[23:22]	**PRI_30**	**Priority of IRQ30** 0 denotes the highest priority and 3 denotes the lowest priority.
[21:16]	**Reserved**	Reserved.
[15:14]	**PRI_29**	**Priority of IRQ29** 0 denotes the highest priority and 3 denotes the lowest priority.
[13:8]	**Reserved**	Reserved.
[7:6]	**PRI_28**	**Priority of IRQ28** 0 denotes the highest priority and 3 denotes the lowest priority.
[5:0]	**Reserved**	Reserved.

6.2.11.7 Interrupt Source Control Registers

Besides the interrupt control registers associated with the NVIC, the NuMicro® Mini58 series also implements some specific control registers to facilitate the interrupt functions, including “interrupt source identify”, ”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
INT Base Address: **INT_BA = 0x5000_0300**				
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) Interrupt Source Identity	0xXXXX_XXXX
IRQ3_SRC	INT_BA+0x0C	R	IRQ3 (EINT1) Interrupt Source Identity	0xXXXX_XXXX
IRQ4_SRC	INT_BA+0x10	R	IRQ4 (GP0/1) Interrupt Source Identity	0xXXXX_XXXX
IRQ5_SRC	INT_BA+0x14	R	IRQ5 (GP2/3/4) Interrupt Source Identity	0xXXXX_XXXX
IRQ6_SRC	INT_BA+0x18	R	IRQ6 (PWM) Interrupt Source Identity	0xXXXX_XXXX
IRQ7_SRC	INT_BA+0x1C	R	IRQ7 (BRAKE) 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	-	Reserved	0xXXXX_XXXX
IRQ11_SRC	INT_BA+0x2C	-	Reserved	0xXXXX_XXXX
IRQ12_SRC	INT_BA+0x30	R	IRQ12 (UART0) 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 (SPI) Interrupt Source Identity	0xXXXX_XXXX
IRQ15_SRC	INT_BA+0x3C	-	Reserved	0xXXXX_XXXX
IRQ16_SRC	INT_BA+0x40	R	IRQ16 (GP5) Interrupt Source Identity	0xXXXX_XXXX
IRQ17_SRC	INT_BA+0x44	R	IRQ17 (HIRC trim) Interrupt Source Identity	0xXXXX_XXXX
IRQ18_SRC	INT_BA+0x48	R	IRQ18 (I^2C0) Interrupt Source Identity	0xXXXX_XXXX
IRQ19_SRC	INT_BA+0x4C	R	IRQ19 (I^2C1) Interrupt Source Identity	0xXXXX_XXXX
IRQ20_SRC	INT_BA+0x50	-	Reserved	0xXXXX_XXXX
IRQ21_SRC	INT_BA+0x54	-	Reserved	0xXXXX_XXXX
IRQ22_SRC	INT_BA+0x58	-	Reserved	0xXXXX_XXXX
IRQ23_SRC	INT_BA+0x5C	-	Reserved	0xXXXX_XXXX
IRQ24_SRC	INT_BA+0x60	-	Reserved	0xXXXX_XXXX

Register	Offset	R/W	Description	Reset Value
INT Base Address: **INT_BA = 0x5000_0300**				
IRQ25_SRC	INT_BA+0x64	R	IRQ25 (ACMP) Interrupt Source Identity	0xXXXX_XXXX
IRQ26_SRC	INT_BA+0x68	-	Reserved	0xXXXX_XXXX
IRQ27_SRC	INT_BA+0x6C	-	Reserved	0xXXXX_XXXX
IRQ28_SRC	INT_BA+0x70	R	IRQ28 (PWRWU) Interrupt Source Identity	0xXXXX_XXXX
IRQ29_SRC	INT_BA+0x74	R	IRQ29 (ADC) Interrupt Source Identity	0xXXXX_XXXX
IRQ30_SRC	INT_BA+0x78	-	Reserved	0xXXXX_XXXX
IRQ31_SRC	INT_BA+0x7C	-	Reserved	0xXXXX_XXXX
NMI_CON	INT_BA+0x80	R/W	NMI Source Interrupt Select Control Register	0x0000_0000
MCU_IRQ	INT_BA+0x84	R/W	MCU IRQ Number Identity Register	0x0000_0000

IRQ0 (BOD) Interrupt Source Identity (IRQ0_SRC)

Register	Offset	R/W	Description	Reset Value
IRQ0_SRC	INT_BA+0x00	R	IRQ0 (BOD) Interrupt Source Identity	0xXXXX_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							**BOD_INT**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**BOD_INT**	**IRQ0 Source Identity** 0 = IRQ0 source is not from BOD interrupt (BOD_INT). 1 = IRQ0 source is from BOD interrupt (BOD_INT).

IRQ1 (WDT) Interrupt Source Identity (IRQ1_SRC)

Register	Offset	R/W	Description	Reset Value
IRQ1_SRC	INT_BA+0x04	R	IRQ1 (WDT) Interrupt Source Identity	0xXXXX_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						**WWDT_INT**	**WDT_INT**

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**WWDT_INT**	**IRQ1 Source Identity** 0 = IRQ1 source is not from window watchdog interrupt (WWDT _INT). 1 = IRQ1 source is from window watchdog interrupt (WWDT_INT).
[0]	**WDT_INT**	**IRQ1 Source Identity** 0 = IRQ1 source is not from watchdog interrupt (WDT _INT). 1 = IRQ1 source is from watchdog interrupt (WDT_INT).

IRQ2 (EINT0) Interrupt Source Identity (IRQ2_SRC)

Register	Offset	R/W	Description	Reset Value
IRQ2_SRC	INT_BA+0x08	R	IRQ2 (EINT0) Interrupt Source Identity	0xXXXX_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							**EINT0**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**EINT0**	**IRQ2 Source Identity** 0 = IRQ2 source is not from external signal interrupt 0 from P3.2 (EINT0). 1 = IRQ2 source is from external signal interrupt 0 from P3.2 (EINT0).

IRQ3 (EINT1) Interrupt Source Identity (IRQ3_SRC)

Register	Offset	R/W	Description	Reset Value
IRQ3_SRC	INT_BA+0x0C	R	IRQ3 (EINT1) Interrupt Source Identity	0xXXXX_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							**EINT1**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**EINT1**	**IRQ3 Source Identity** 0 = IRQ3 source is not from external signal interrupt 1 from P5.2 (EINT1). 1 = IRQ3 source is from external signal interrupt 1 from P5.2 (EINT1).

IRQ4 (GPA/B) Interrupt Source Identity (IRQ4_SRC)

Register	Offset	R/W	Description	Reset Value
IRQ4_SRC	INT_BA+0x10	R	IRQ4 (GP0/1) Interrupt Source Identity	0xXXXX_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						**GP1_INT**	**GP0_INT**

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**GP1_INT**	**IRQ4 Source Identity** 0 = IRQ4 source is not from GP1 interrupt (GP1_INT). 1 = IRQ4 source is from GP1 interrupt (GP1_INT).
[0]	**GP0_INT**	**IRQ4 Source Identity** 0 = IRQ4 source is not from GP0 interrupt (GP0_INT). 1 = IRQ4 source is from GP0 interrupt (GP0_INT).

IRQ5 (GPC/D/F) Interrupt Source Identity (IRQ5_SRC)

Register	Offset	R/W	Description	Reset Value
IRQ5_SRC	INT_BA+0x14	R	IRQ5 (GP2/3/4) Interrupt Source Identity	0xXXXX_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					**GP4_INT**	**GP3_INT**	**GP2_INT**

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2]	**GP4_INT**	**IRQ5 Source Identity** 0 = IRQ5 source is not from GP4 interrupt (GP4_INT). 1 = IRQ5 source is from GP4 interrupt (GP4_INT).
[1]	**GP3_INT**	**IRQ5 Source Identity** 0 = IRQ5 source is not from GP3 interrupt (GP3_INT). 1 = IRQ5 source is from GP3 interrupt (GP3_INT).
[0]	**GP2_INT**	**IRQ5 Source Identity** 0 = IRQ5 source is not from GP2 interrupt (GP2_INT). 1 = IRQ5 source is from GP2 interrupt (GP2_INT).

IRQ6 (PWM) Interrupt Source Identity (IRQ6_SRC)

Register	Offset	R/W	Description	Reset Value
IRQ6_SRC	INT_BA+0x18	R	IRQ6 (PWM) Interrupt Source Identity	0xXXXX_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							**PWM_INT**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**PWM_INT**	**IRQ6 Source Identity** 0 = IRQ6 source is not from PWM interrupt (PWM_INT). 1 = IRQ6 source is from PWM interrupt (PWM_INT).

IRQ7 (BRAKE) Interrupt Source Identity (IRQ7_SRC)

Register	Offset	R/W	Description	Reset Value
IRQ7_SRC	INT_BA+0x1C	R	IRQ7 (BRAKE) Interrupt Source Identity	0xXXXX_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							**BRAKE_INT**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**BRAKE_INT**	**IRQ7 Source Identity** 0 = IRQ7 source is not from Brake interrupt (BRAKE_INT). 1 = IRQ7 source is from Brake interrupt (BRAKE_INT).

IRQ8 (TMR0) Interrupt Source Identity (IRQ8_SRC)

Register	Offset	R/W	Description	Reset Value
IRQ8_SRC	INT_BA+0x20	R	IRQ8 (TMR0) Interrupt Source Identity	0xXXXX_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							**TMR0_INT**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**TMR0_INT**	**IRQ8 Source Identity** 0 = IRQ8 source is not from Timer0 interrupt (TMR0_INT). 1 = IRQ8 source is from Timer0 interrupt (TMR0_INT).

IRQ9 (TMR1) Interrupt Source Identity (IRQ9_SRC)

Register	Offset	R/W	Description	Reset Value
IRQ9_SRC	INT_BA+0x24	R	IRQ9 (TMR1) Interrupt Source Identity	0xXXXX_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							**TMR1_INT**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**TMR1_INT**	**IRQ9 Source Identity** 0 = IRQ9 source is not from Timer1 interrupt (TMR1_INT). 1 = IRQ9 source is from Timer1 interrupt (TMR1_INT).

IRQ12 (UART0) Interrupt Source Identity (IRQ12_SRC)

Register	Offset	R/W	Description	Reset Value
IRQ12_SRC	INT_BA+0x30	R	IRQ12 (UART0) Interrupt Source Identity	0xXXXX_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							**UART0_INT**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**UART0_INT**	**IRQ12 Source Identity** 0 = IRQ12 source is not from UART0 interrupt (UART0_INT). 1 = IRQ12 source is from UART0 interrupt (UART0_INT).

IRQ13 (UART1) Interrupt Source Identity (IRQ13_SRC)

Register	Offset	R/W	Description	Reset Value
IRQ13_SRC	INT_BA+0x34	R	IRQ13 (UART1) Interrupt Source Identity	0xXXXX_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							**UART1_INT**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**UART1_INT**	**IRQ13 Source Identity** 0 = IRQ13 source is not from UART1 interrupt (UART1_INT). 1 = IRQ13 source is from UART1 interrupt (UART1_INT).

IRQ14 (SPI) Interrupt Source Identity (IRQ14_SRC)

Register	Offset	R/W	Description	Reset Value
IRQ14_SRC	INT_BA+0x38	R	IRQ14 (SPI) Interrupt Source Identity	0xXXXX_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							**SPI_INT**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**SPI_INT**	**IRQ14 Source Identity** 0 = IRQ14 source is not from SPI interrupt (SPI_INT). 1 = IRQ14 source is from SPI interrupt (SPI_INT).

IRQ16 (GP5) Interrupt Source Identity (IRQ16_SRC)

Register	Offset	R/W	Description	Reset Value
IRQ16_SRC	INT_BA+0x40	R	IRQ16 (GP5) Interrupt Source Identity	0xXXXX_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							**GP5_INT**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**GP5_INT**	**IRQ16 Source Identity** 0 = IRQ16 source is not from GP5 interrupt (GP5_INT). 1 = IRQ16 source is from GP5 interrupt (GP5_INT).

IRQ17 (HIRC trim) Interrupt Source Identity (IRQ17_SRC)

Register	Offset	R/W	Description	Reset Value
IRQ17_SRC	INT_BA+0x44	R	IRQ17 (HIRC trim) Interrupt Source Identity	0xXXXX_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							**HIRC_TRIM_INT**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**HIRC_TRIM_INT**	**IRQ17 Source Identity** 0 = IRQ17 source is not from HIRC trim interrupt (HIRC_TRIM_INT). 1 = IRQ17 source is from HIRC trim interrupt (HIRC_TRIM_INT).

IRQ18 (I^2C0) Interrupt Source Identity (IRQ18_SRC)

Register	Offset	R/W	Description	Reset Value
IRQ18_SRC	INT_BA+0x48	R	IRQ18 (I^2C0) Interrupt Source Identity	0xXXXX_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							**I2C0_INT**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**I2C0_INT**	**IRQ18 Source Identity** 0 = IRQ18 source is not from I^2C0 interrupt (I2C0_INT). 1 = IRQ18 source is from I^2C0 interrupt (I2C0_INT).

IRQ19 (I^2C1) Interrupt Source Identity (IRQ19_SRC)

Register	Offset	R/W	Description	Reset Value
IRQ19_SRC	INT_BA+0x4C	R	IRQ19 (I^2C1) Interrupt Source Identity	0xXXXX_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							**I2C1_INT**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**I2C1_INT**	**IRQ19 Source Identity** 0 = IRQ19 source is not from I^2C1 interrupt (I2C1_INT). 1 = IRQ19 source is from I^2C1 interrupt (I2C1_INT).

IRQ25 (ACMP) Interrupt Source Identity (IRQ25_SRC)

Register	Offset	R/W	Description	Reset Value
IRQ25_SRC	INT_BA+0x64	R	IRQ25 (ACMP) Interrupt Source Identity	0xXXXX_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							**ACMP_INT**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**ACMP_INT**	**IRQ25 Source Identity** 0 = IRQ25 source is not from ACMP interrupt (ACMP_INT). 1 = IRQ25 source is from ACMP interrupt (ACMP_INT).

IRQ28 (PWRWU) Interrupt Source Identity (IRQ28_SRC)

Register	Offset	R/W	Description	Reset Value
IRQ28_SRC	INT_BA+0x70	R	IRQ28 (PWRWU) Interrupt Source Identity	0xXXXX_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							**PWRWU_INT**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**PWRWU_INT**	**IRQ28 Source Identity** 0 = IRQ28 source is not from PWRWU interrupt (PWRWU_INT). 1 = IRQ28 source is from PWREU interrupt (PWRWU_INT).

IRQ29 (ADC) Interrupt Source Identity (IRQ29_SRC)

Register	Offset	R/W	Description	Reset Value
IRQ29_SRC	INT_BA+0x74	R	IRQ29 (ADC) Interrupt Source Identity	0xXXXX_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							**ADC_INT**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**ADC_INT**	**IRQ29 Source Identity** 0 = IRQ29 source is not from ADC interrupt (ADC_INT). 1 = IRQ29 source is from ADC interrupt (ADC_INT).

NMI Interrupt Source Select Control Register (NMI_SEL)

Register	Offset	R/W	Description	Reset Value
NMI_CON	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_SEL_EN**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			**NMI_SEL**				

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8]	**NMI_SEL_EN**	**NMI Interrupt Enable Bit (Write Protected)** 0 = NMI interrupt Disabled. 1 = NMI interrupt Enabled. **Note:** This bit is the protected bit, and programming it needs to write 0x59, 0x16, and 0x88 to address 0x5000_0100 to disable register protection. Refer to the register SYS_REGLCTL at address SYS_BA+0x100.
[7:5]	**Reserved**	Reserved.
[4:0]	**NMI_SEL**	**NMI Interrupt Source Select Bit** The NMI interrupt to Cortex®-M0 can be selected from one of the peripheral interrupt by setting NMI_SEL.

MCU Interrupt Request Source Register (MCU_IRQ)

Register	Offset	R/W	Description	Reset Value
MCU_IRQ	INT_BA+0x84	R/W	MCU IRQ Number Identity Register	0x0000_0000

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

Bits	Description	
[31:0]	**MCU_IRQ**	**MCU IRQ Source Bits** The MCU_IRQ collects all the interrupts from the peripherals and generates the synchronous interrupt to Cortex®-M0 core. There are two modes to generate interrupt to Cortex®-M0 - the normal mode and test mode. The MCU_IRQ collects all interrupts from each peripheral and synchronizes them then interrupts the Cortex®-M0. When the MCU_IRQ[n] is 0, setting MCU_IRQ[n] to 1 will generate an interrupt to Cortex®-M0 NVIC[n]. When the MCU_IRQ[n] is 1 (mean an interrupt is assert), setting 1 to the MCU_bit[n] will clear the interrupt and setting MCU_IRQ[n] 0 has no effect.

6.2.12 System Control Registers (SCB)

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

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

6.2.12.1 System Control Block Register Map

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

Register	Offset	R/W	Description	Reset Value
SCS Base Address: **SCS_BA = 0xE000_E000**				
SCS_CPUID	SCS_BA+0xD00	R	CPUID Base Register	0x410C_C200
SCS_ICSR	SCS_BA+0xD04	R/W	Interrupt Control State Register	0x0000_0000
SCS_AIRCR	SCS_BA+0xD0C	R/W	Application Interrupt and Reset Control Register	0xFA05_0000
SCS_SCR	SCS_BA+0xD10	R/W	System Control Register	0x0000_0000
SCS_SHPR2	SCS_BA+0xD1C	R/W	System Handler Priority Register 2	0x0000_0000
SCS_SHPR3	SCS_BA+0xD20	R/W	System Handler Priority Register 3	0x0000_0000

6.2.12.2 System Control Register

CPUID Base Register (SCS_CPUID)

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

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

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

Interrupt Control State Register (SCS_ICSR)

Register	Offset	R/W	Description	Reset Value
SCS_ICSR	SCS_BA+0xD04	R/W	Interrupt Control 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				
15	14	13	12	11	10	9	8
VECTPENDING				Reserved			VECTACTIVE
7	6	5	4	3	2	1	0
VECTACTIVE							

Bits	Description	
[31]	**NMIPENDSET**	**NMI Set-pending Bit** Write Operation: 0 = No effect. 1 = Changes NMI exception state to pending. Read Operation: 0 = NMI exception is not pending. 1 = NMI exception is pending. **Note:** 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.
[30:29]	**Reserved**	Reserved.
[28]	**PENDSVSET**	**PendSV Set-pending Bit** Write Operation: 0 = No effect. 1 = Changes PendSV exception state to pending. Read Operation: 0 = PendSV exception is not pending. 1 = PendSV exception is pending. **Note:** Writing 1 to this bit is the only way to set the PendSV exception state to pending.
[27]	**PENDSVCLR**	**PendSV Clear-pending Bit** Write Operation: 0 = No effect. 1 = Removes the pending state from the PendSV exception. **Note:** This bit is write-only. To clear the PENDSV bit, you must “write 0 to PENDSVSET and write 1 to PENDSVCLR” at the same time.

Bits	Description	
[26]	**PENDSTSET**	**SysTick Exception Set-pending Bit** Write Operation: 0 = No effect. 1 = Changes SysTick exception state to pending. Read Operation: 0 = SysTick exception is not pending. 1 = SysTick exception is pending.
[25]	**PENDSTCLR**	**SysTick Exception Clear-pending Bit** Write Operation: 0 = No effect. 1 = Removes the pending state from the SysTick exception. **Note:** This bit is write-only. When you want to clear PENDST bit, you must “write 0 to PENDSTSET and write 1 to PENDSTCLR” at the same time.
[24]	**Reserved**	Reserved.
[23]	**ISRPREEMPT**	**Interrupt Preemption Bit** If set, a pending exception will be serviced on exit from the debug halt state. **Note:** This bit is read only.
[22]	**ISRPENDING**	**Interrupt Pending Flag, Excluding NMI and Faults** 0 = Interrupt not pending. 1 = Interrupt pending. **Note:** This bit is read only.
[21]	**Reserved**	Reserved.
[20:12]	**VECTPENDING**	**Exception Number of the Highest Priority Pending Enabled Exception** 0 = No pending exceptions. Non-zero = Exception number of the highest priority pending enabled exception. **Note:** These bits are read only.
[11:9]	**Reserved**	Reserved.
[8:0]	**VECTACTIVE**	**Contains the Active Exception Number** 0 = Thread mode. Non-zero = Exception number of the currently active exception. **Note:** These bits are read only.

Application Interrupt and Reset Control Register (SCS_AIRCR)

Register	Offset	R/W	Description	Reset Value
SCS_AIRCR	SCS_BA+0xD0C	R/W	Application Interrupt and Reset Control Register	0xFA05_0000

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

Bits	Description	
[31:16]	**VECTORKEY**	**Register Access Key** 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 filed is used to prevent accidental write to this register from resetting the system or clearing of the exception status. Read Operation: Read as 0xFA05.
[15:3]	**Reserved**	Reserved.
[2]	**SYSRESETREQ**	**System Reset Request** Writing this bit 1 will cause a reset signal to be asserted to the chip to indicate a reset is requested. The bit is a write only bit and self-clears as part of the reset sequence.
[1]	**VECTCLRACTIVE**	**Exception Active Status Clear Bit** Reserved for debug use. When writing to the register, user must write 0 to this bit, otherwise behavior is unpredictable.
[0]	**Reserved**	Reserved.

System Control Register (SCS_SCR)

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

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

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

System Handler Priority Register 2 (SCS_SHPR2)

Register	Offset	R/W	Description	Reset Value
SCS_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 (SCS_SHPR3)

Register	Offset	R/W	Description	Reset Value
SCS_SHPR3	SCS_BA+0xD20	R/W	System Handler Priority Register 3	0x0000_0000

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

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

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 clock divider. The chip enters Power-down mode when Cortex®-M0 core executes the WFI instruction only if the PDEN (CLK_PWRCTL[7]) bit is set to 1. After that, chip enters Power-down mode and waits for wake-up interrupt source triggered to exit Power-down mode. In Power-down mode, the clock controller turns off the 4~24 MHz external high speed crystal (HXT) and 22.1184 MHz internal high speed RC oscillator (HIRC) to reduce the overall system power consumption. The following figures show the clock generator and the overview of the clock source control.

The clock generator consists of 3 sources as listed below:

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

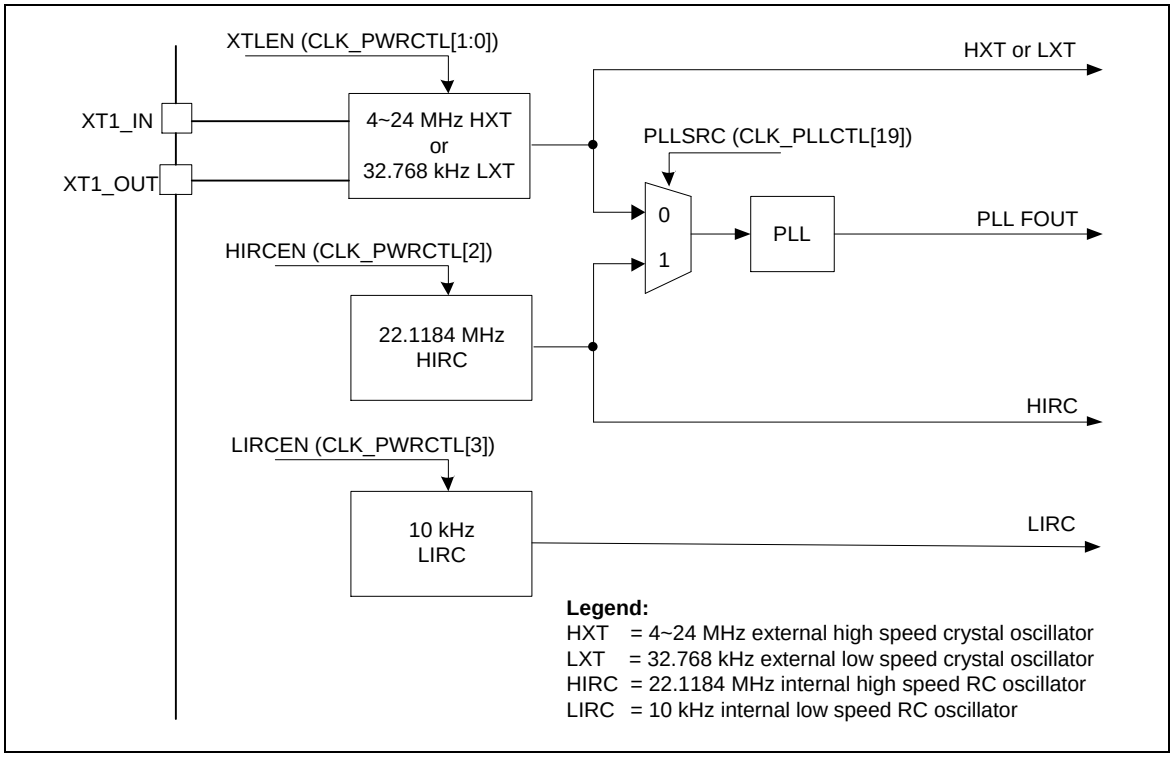


Figure 6.3-1 Clock Generator Block Diagram

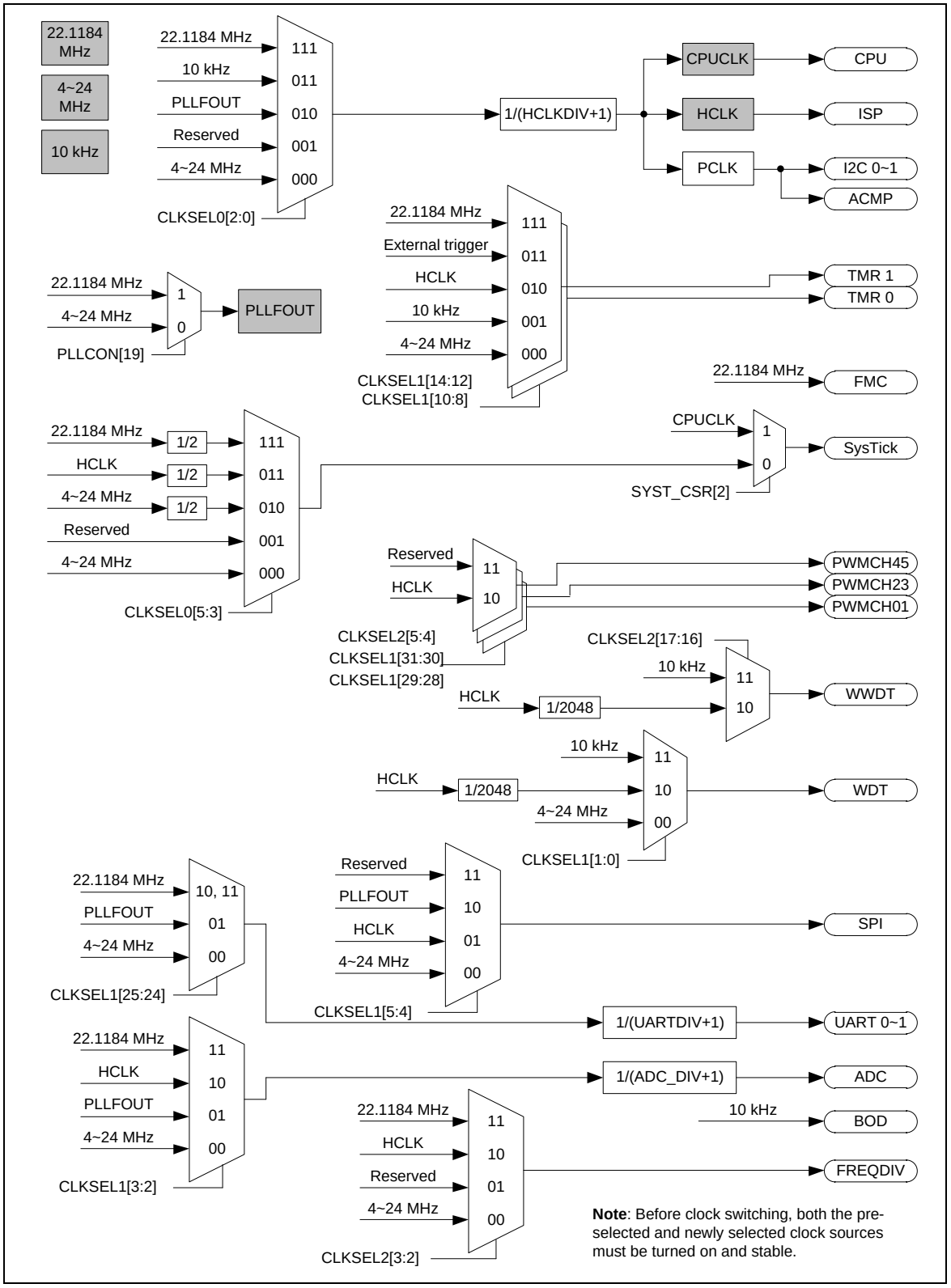


Figure 6.3-2 Clock Generator Global View Diagram

6.3.2 Auto-trim

This chip supports auto-trim function: the HIRC trim (22.1184 MHz internal RC oscillator), according to the accurate LXT (32.768 kHz crystal oscillator), automatically gets accurate HIRC output frequency, 1 % 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 22.1184 MHz HXT as the system clock source, they need to solder 32.768 kHz crystal in system, and set FREQSEL (SYS_IRCTCTL[0] trim frequency selection) to “1”, and the auto-trim function will be enabled. Interrupt status bit FREQLOCK (SYS_IRCTISTS[0] HIRC frequency lock status) high indicates the HIRC output frequency is accurate within 1% 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”.

6.3.3 System Clock and SysTick Clock

The system clock has 4 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.

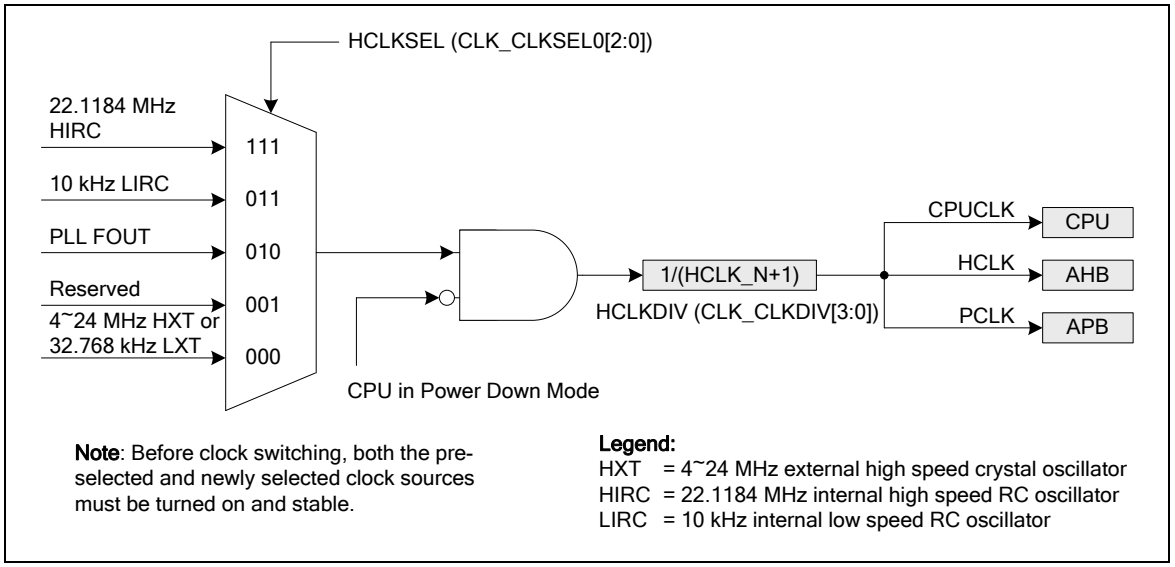


Figure 6.3-3 System Clock Block Diagram

The source of PCLK is equal to HCLK in system clock architecture.

The clock source of SysTick in Cortex®-M0 core can use CPU clock or external clock CLKSRC(SYST_CSR[2]). If using external clock, the SysTick clock (STCLK) has 4 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 6.3-4.

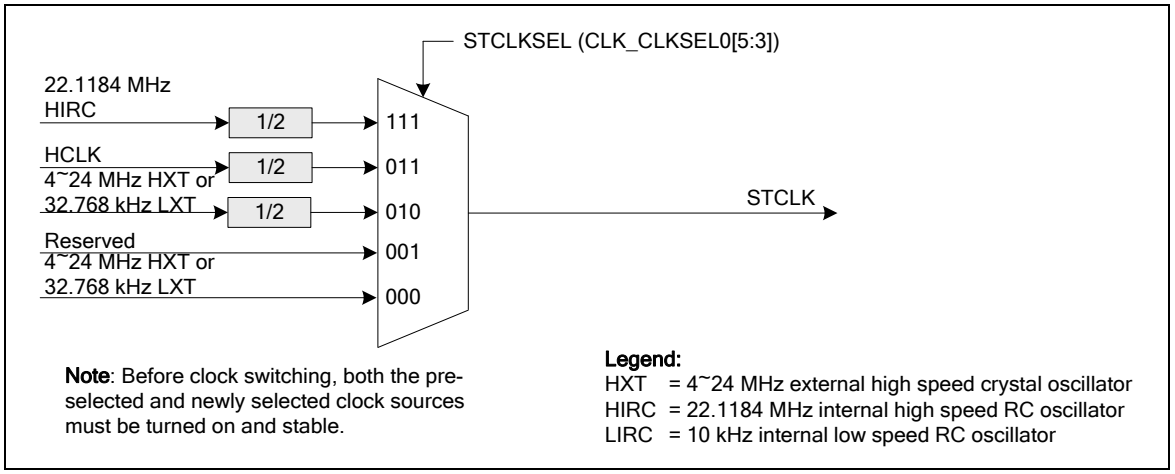


Figure 6.3-4 SysTick Clock Control Block Diagram

6.3.4 Peripherals Clock Source Selection

The peripheral clock has different clock source switch settings depending on different peripherals. Please refer to the CLK_CLKSEL1 and CLK_APBCLK register description in section 6.3.8. Please to note that, while switching clock source from one to another, user must wait until both clock sources are running stabled.

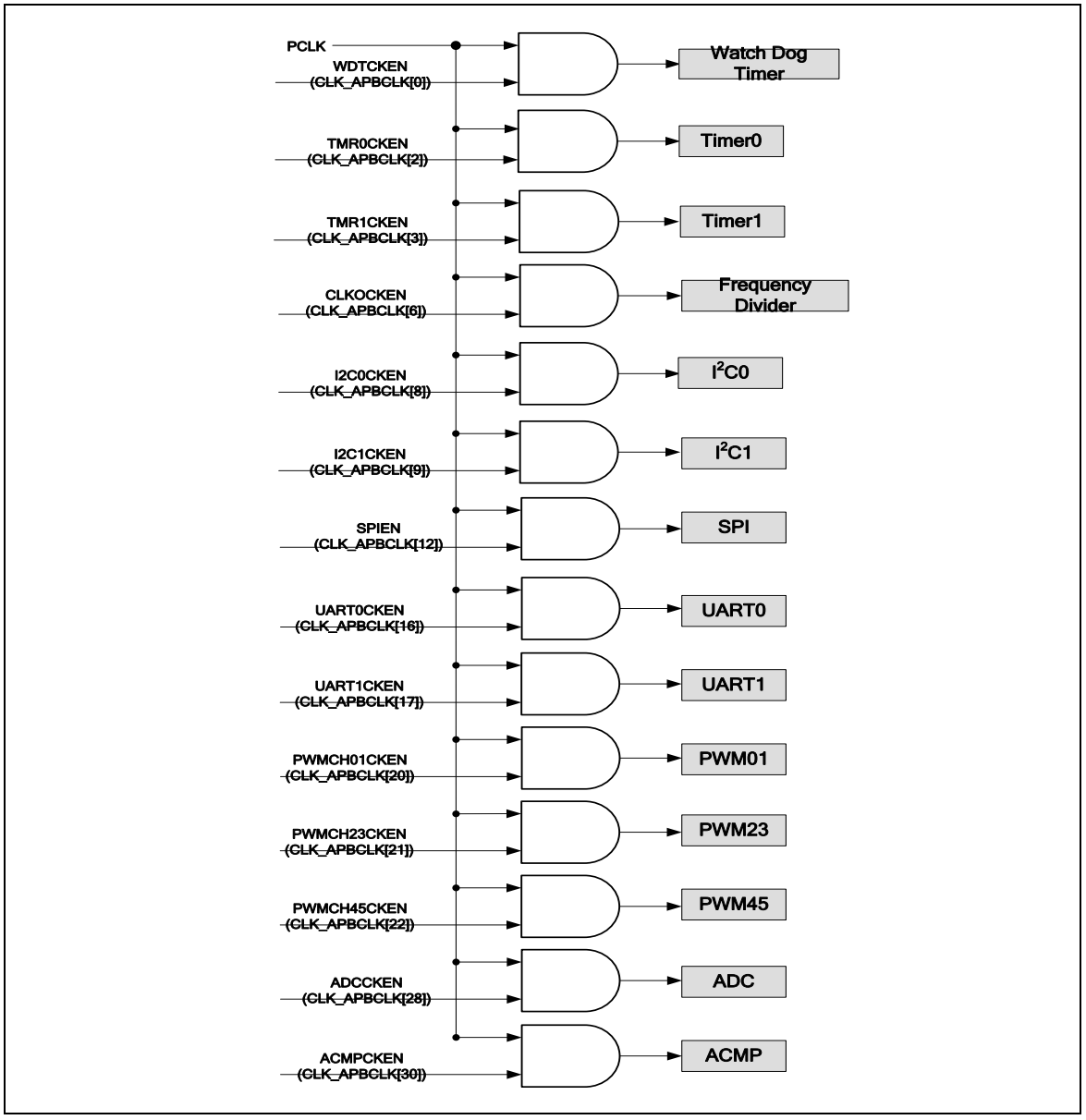


Figure 6.3-5 Peripherals Bus Clock Source Selection for PCLK

	Peripheral Clok Selectable	Ext. CLK (HXT Or LXT)	HIRC	LIRC	HCLK	PLL
WDT	Yes	Yes	No	Yes	Yes	No
WWDT	Yes	Yes	No	Yes	Yes	No
Timer0	Yes	Yes	Yes	Yes	Yes	No
Timer1	Yes	Yes	Yes	Yes	Yes	No
I^2C0	No	-	-	-	-	-
I^2C1	No	-	-	-	-	-
SPI	Yes	Yes	No	No	Yes	Yes
UART0	Yes	Yes	Yes	No	No	Yes
UART1	Yes	Yes	Yes	No	No	Yes
PWM	No	-	-	-	-	-
ADC	Yes	Yes	Yes	No	Yes	Yes
ACMP	No	-	-	-	-	-

Table 6.3-1 Peripheral Clock Source Selection Table

Note: For the peripherals those peripheral clock are not selectable, its clock source is fixed to PCLK.

6.3.5 Power-down Mode Clock

When chip enters Power-down mode, system clocks, some clock sources, and some peripheral clocks will be disabled. Some clock sources and peripheral clocks are still active in Power-down mode.

The clocks still kept active are listed below:

- Clock Generator
- 10 kHz internal low speed oscillator (LIRC) clock
- 32.768 kHz external low speed crystal oscillator (LXT) clock (If PDLXT = 1 and XTLEN[1:0] = 10)
- Peripherals Clock (When 10 kHz low speed oscillator is adopted as clock source)
  - Watchdog Clock
  - Timer 0/1 Clock

6.3.6 Frequency Divider Output

This device is equipped with a power-of-2 frequency divider which is composed of 16 chained divide-by-2 shift registers. One of the 16 shift register outputs selected by a sixteen to one multiplexer is reflected to the CLKO 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 DIV1EN (CLK_CLKOCTL[5]) set to 1, the frequency divider clock (FRQDIV_CLK) will bypass power-of-2 frequency divider. The frequency divider clock will be output to CLKO pin directly.

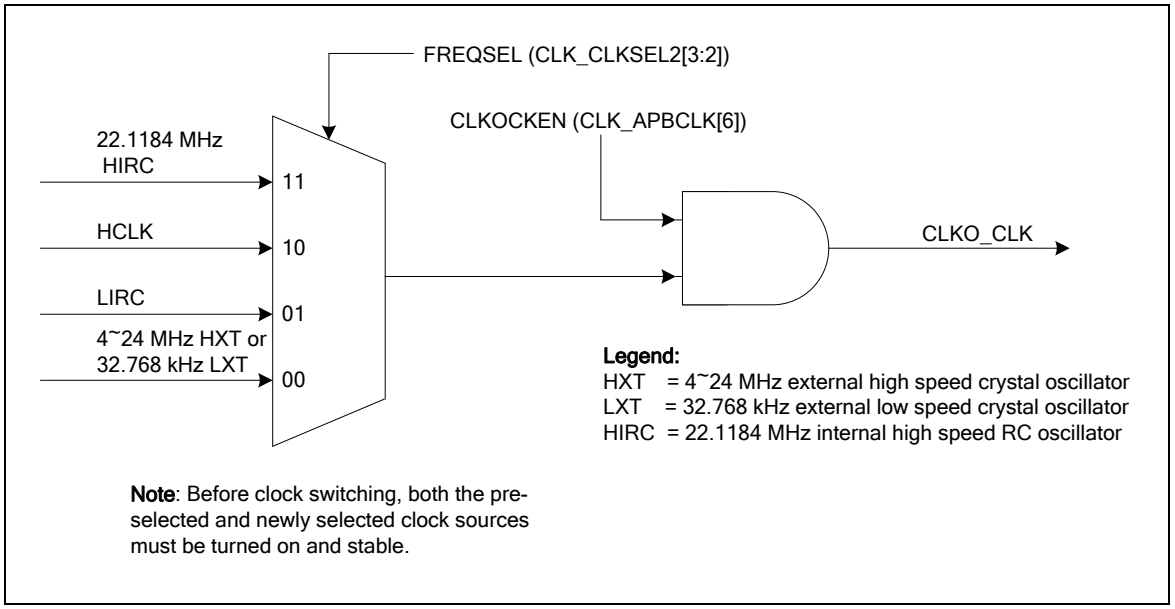


Figure 6.3-6 Clock Source of Frequency Divider

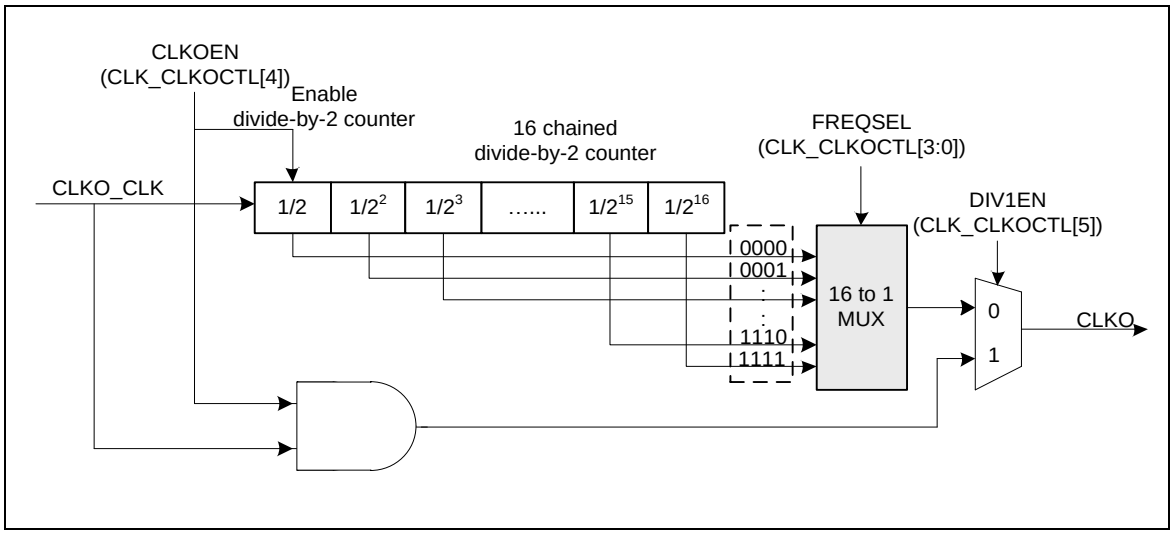


Figure 6.3-7 Block Diagram of Frequency Divider

6.3.7 Register Map

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

Register	Offset	R/W	Description	Reset Value
CLK Base Address: **CLK_BA = 0x5000_0200**				
CLK_PWRCTL	CLK_BA+0x00	R/W	System Power-down Control Register	0x0000_001C
CLK_AHBCLK	CLK_BA+0x04	R/W	AHB Devices Clock Enable Control Register	0x0000_0005
CLK_APBCLK	CLK_BA+0x08	R/W	APB Devices Clock Enable Control Register	0x0000_0001
CLK_STATUS	CLK_BA+0x0C	R/W	Clock Status Monitor Register	0x0000_0018
CLK_CLKSEL0	CLK_BA+0x10	R/W	Clock Source Select Control Register 0	0x0000_003F
CLK_CLKSEL1	CLK_BA+0x14	R/W	Clock Source Select Control Register 1	0xAFFF_FFFF
CLK_CLKDIV	CLK_BA+0x18	R/W	Clock Divider Number Register	0x0000_0000
CLK_CLKSEL2	CLK_BA+0x1C	R/W	Clock Source Select Control Register 2	0x0002_00EF
CLK_PLLCTL	CLK_BA+0x20	R/W	PLL Control Register	0x0005_C230
CLK_CLKOCTL	CLK_BA+0x24	R/W	Frequency Divider Control Register	0x0000_0000

6.3.8 Register Description

Power-down Control Register (CLK_PWRCTL)

Except the BIT[6], all the other bits are protected, and programming these bits need to write 0x59, 0x16, 0x88 to address 0x5000_0100 to disable register protection. Refer to the SYS_REGLCTL register at address SYS_BA + 0x100.

Register	Offset	R/W	Description	Reset Value
CLK_PWRCTL	CLK_BA+0x00	R/W	System Power-down Control Register	0x0000_001C

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

Bits	Description	
[31:10]	**Reserved**	Reserved.
[9]	**PDLXT**	**Enable LXT In Power-down Mode** This bit controls the crystal oscillator active or not in Power-down mode. 0 = No effect to Power-down mode. 1 = If XTLEN[1:0] = 10, LXT is still active in Power-down mode.
[8]	**Reserved**	Reserved.
[7]	**PDEN**	**System Power-down Enable Bit (Write Protect)** When chip wakes up from Power-down mode, this bit is cleared by hardware. User needs to set this bit again for next Power-down. In Power-down mode, 4~24 MHz external high speed crystal oscillator (HXT), 32.768 kHz external low speed crystal oscillator (LXT), and the 22.1184 MHz internal high speed oscillator (HIRC) will be disabled in this mode, and 10 kHz internal low speed RC oscillator (LIRC) are not controlled by Power-down mode. 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 10 kHz internal low speed oscillator. 0 = Chip operating normally or chip in Idle mode because of WFI command. 1 = Chip enters Power-down mode instantly or waits CPU sleep command WFI.

Bits	Description	
[6]	**PDWKIF**	**Power-down Mode Wake-up Interrupt Status** Set by “Power-down wake-up event”, which indicates that resume from Power-down mode” The flag is set if the GPIO, UART, WDT, ACMP, Timer or BOD wake-up occurred. **Note:** This bit works only if PDWKIEN (CLK_PWRCTL[5]) set to 1. Write 1 to clear the bit to 0.
[5]	**PDWKIEN**	**Power-down Mode Wake-up Interrupt Enable Bit (Write Protect)** 0 = Power-down mode wake-up interrupt Disabled. 1 = Power-down mode wake-up interrupt Enabled. **Note:** The interrupt will occur when both PDWKIF and PDWKIEN are high.
[4]	**PDWKDLY**	**Wake-up Delay Counter Enable Bit (Write Protect)** When the chip wakes up from Power-down mode, the clock control will delay certain clock cycles to wait system clock stable. The delayed clock cycle is 4096 clock cycles when chip work at 4~24 MHz external high speed crystal (HXT), 4096 clock cycles for 32.768 kHz external low speed crystal (LXT), and 16 clock cycles when chip works at 22.1184 MHz internal high speed RC oscillator (HIRC). 0 = Clock cycles delay Disabled. 1 = Clock cycles delay Enabled.
[3]	**LIRCEN**	**LIRC Enable Bit (Write Protect)** 0 = 10 kHz internal low speed RC oscillator (LIRC) Disabled. 1 = 10 kHz internal low speed RC oscillator (LIRC) Enabled.
[2]	**HIRCEN**	**HIRC Enable Bit (Write Protect)** 0 = 22.1184 MHz internal high speed RC oscillator (HIRC) Disabled. 1 = 22.1184 MHz internal high speed RC oscillator (HIRC) Enabled. **Note:** The default of HIRCEN bit is 1.
[1:0]	**XTLEN**	**External HXT Or LXT Crystal Oscillator Enable Bit (Write Protect)** The default clock source is from HIRC. These two bits are default set to “00” and the XT1_IN and XT1_OUT pins are GPIO. 00 = XT1_IN and XT1_OUT are GPIO, disable both LXT & HXT (default). 01 = HXT Enabled. 10 = LXT Enabled. 11 = XT1_IN is external clock input pin, XT1_OUT is GPIO. **Note:** To enable external XTAL function, ALT[1:0] and MFP[1:0] bits must also be set in SYS_P5_MFP.

Register Or Instruction Mode	SLEEPDEEP (SCR[2])	PDEN (CLK_PWRCTL[7])	CPU Run WFI Instruction	Clock Disable
Normal operation	0	0	NO	All clocks disabled by control register
Idle mode (CPU entering Sleep mode)	0	0	YES	Only CPU clock disabled
Power-down mode (CPU entering Deep Sleep mode)	1	1	YES	Most clocks are disabled except 10 kHz and only WDT peripheral clock still enable if its peripheral clock source is selected as 10 kHz.

Table 6.3-2 Power-down Mode Control

When chip enters Power-down mode, user can wake-up this chip using some interrupt sources. The related interrupt sources and NVIC IRQ enable bits (NVIC_ISER) should be enabled before setting 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.

Register	Offset	R/W	Description	Reset Value
CLK_AHBCLK	CLK_BA+0x04	R/W	AHB Devices Clock Enable Control Register	0x0000_0005

31	30	29	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					ISPCKEN	Reserved	

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2]	**ISPCKEN**	**Flash ISP Controller Clock Enable Bit** 0 = Flash ISP peripheral clock Disabled. 1 = Flash ISP peripheral clock Enabled.
[1:0]	**Reserved**	Reserved.

APB Devices Clock Enable Control Register (CLK_APBCLK)

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

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

31	30	29	28	27	26	25	24
Reserved	ACMPCKEN	Reserved	ADCCKEN	Reserved			
23	22	21	20	19	18	17	16
Reserved	PWMCH45CKEN	PWMCH23CKEN	PWMCH01CKEN	Reserved		UART1CKEN	UART0CKEN
15	14	13	12	11	10	9	8
Reserved			SPICKEN	Reserved		I2C1CKEN	I2C0CKEN
7	6	5	4	3	2	1	0
Reserved	CLKOCKEN	Reserved		TMR1CKEN	TMR0CKEN	Reserved	WDTCKEN

Bits	Description	
[31]	**Reserved**	Reserved.
[30]	**ACMPCKEN**	**Analog Comparator Clock Enable Bit** 0 = Analog Comparator clock Disabled. 1 = Analog Comparator clock Enabled.
[29]	**Reserved**	Reserved.
[28]	**ADCCKEN**	**Analog-digital-converter (ADC) Clock Enable Bit** 0 = ADC peripheral clock Disabled. 1 = ADC peripheral clock Enabled.
[27:23]	**Reserved**	Reserved.
[22]	**PWMCH45CKEN**	**PWM_45 Clock Enable Bit** 0 = PWM45 clock Disabled. 1 = PWM45 clock Enabled.
[21]	**PWMCH23CKEN**	**PWM_23 Clock Enable Bit** 0 = PWM23 clock Disabled. 1 = PWM23 clock Enabled.
[20]	**PWMCH01CKEN**	**PWM_01 Clock Enable Bit** 0 = PWM01 clock Disabled. 1 = PWM01 clock Enabled.
[19:18]	**Reserved**	Reserved.
[17]	**UART1CKEN**	**UART1 Clock Enable Bit** 0 = UART1 clock Disabled. 1 = UART1 clock Enabled.

Bits	Description	
[16]	**UART0CKEN**	**UART0 Clock Enable Bit** 0 = UART0 clock Disabled. 1 = UART0 clock Enabled.
[15:13]	**Reserved**	Reserved.
[12]	**SPICKEN**	**SPI Clock Enable Bit** 0 = SPI peripheral clock Disabled. 1 = SPI peripheral clock Enabled.
[11:10]	**Reserved**	Reserved.
[9]	**I2C1CKEN**	**I^2C1 Clock Enable Bit** 0 = I^2C1 clock Disabled. 1 = I^2C1 clock Enabled.
[8]	**I2C0CKEN**	**I^2C0 Clock Enable Bit** 0 = I^2C0 clock Disabled. 1 = I^2C0 clock Enabled.
[7]	**Reserved**	Reserved.
[6]	**CLKOCKEN**	**Frequency Divider Output Clock Enable Bit** 0 = CLKO clock Disabled. 1 = CLKO clock Enabled.
[5:4]	**Reserved**	Reserved.
[3]	**TMR1CKEN**	**Timer1 Clock Enable Bit** 0 = Timer1 clock Disabled. 1 = Timer1 clock Enabled.
[2]	**TMR0CKEN**	**Timer0 Clock Enable Bit** 0 = Timer0 clock Disabled. 1 = Timer0 clock Enabled.
[1]	**Reserved**	Reserved.
[0]	**WDTCKEN**	**Watchdog Timer Clock Enable Bit (Write Protect)** 0 = Watchdog Timer clock Disabled. 1 = Watchdog Timer clock Enabled. **Note:** This bit is the protected bit, and programming it needs to write 0x59, 0x16, and 0x88 to address 0x5000_0100 to disable register protection. Refer to the register SYS_REGLCTL at address SYS_BA + 0x100.

Clock Status Register (CLK_STATUS)

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

Register	Offset	R/W	Description	Reset Value
CLK_STATUS	CLK_BA+0x0C	R/W	Clock Status Monitor Register	0x0000_0018

31	30	29	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		HIRCSTB	LIRCSTB	PLLSTB	Reserved	XTLSTB

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**CLKSFAIL**	**Clock Switch Fail Flag (Read Only)** This bit is updated when software switches system clock source. If switch target clock is stable, this bit will be set to 0. If switch target clock is not stable, this bit will be set to 1. 0 = Clock switching success. 1 = Clock switching failure. **Note:** This bit is read only. After selected clock source is stable, hardware will switch system clock to selected clock automatically, and CLKSFAIL will be cleared automatically by hardware.
[6:5]	**Reserved**	Reserved.
[4]	**HIRCSTB**	**HIRC Clock Source Stable Flag (Read Only)** 0 = HIRC clock is not stable or disabled. 1 = HIRC clock is stable and enabled.
[3]	**LIRCSTB**	**LIRC Clock Source Stable Flag (Read Only)** 0 = LIRC clock is not stable or disabled. 1 = LIRC clock is stable and enabled.
[2]	**PLLSTB**	**Internal PLL Clock Source Stable Flag (Read Only)** 0 = Internal PLL clock is not stable or disabled. 1 = Internal PLL clock is stable and enabled.
[1]	**Reserved**	Reserved.
[0]	**XTLSTB**	**HXT Or LXT Clock Source Stable Flag** 0 = HXT or LXT clock is not stable or disabled. 1 = HXT or LXT clock is stable and 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_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		STCLKSEL			HCLKSEL		

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5:3]	**STCLKSEL**	**Cortex®-M0 SysTick Clock Source Selection (Write Protect)** If CLKSRC (SYST_CSR[2]) = 1, SysTick clock source is from HCLK. If CLKSRC (SYST_CSR[2]) = 0, SysTick clock source is defined by below settings. 000 = Clock source is from HXT or LXT. 001 = Reserved. 010 = Clock source is from HXT/2 or LXT/2. 011 = Clock source is from HCLK/2. 111 = Clock source is from HIRC/2. Others = Reserved. **Note1:** These bits are protected bit, and programming them needs to write 0x59, 0x16, and 0x88 to address 0x5000_0100 to disable register protection. Refer to the register SYS_REGLCTL at address SYS_BA + 0x100. **Note2:** If the SysTick clock source is not from HCLK (i.e. CLKSRC(SYST_CSR[2]) = 0), SysTick clock source must less than or equal to HCLK/2. **Note3:** To set CLK_PWRCTL[1:0], select HXT or LXT crystal clock.
[2:0]	**HCLKSEL**	**HCLK Clock Source Selection (Write Protect)** 000 = Clock source is from HXT or LXT. 001 = Reserved. 010 = Clock source is from PLL. 011 = Clock source is from LIRC. 111 = Clock source is from HIRC. Others = Reserved. **Note1:** Before clock switching, the related clock sources (both pre-select and new-select) must be turn-on and stable. **Note2:** These bits are protected bit, and programming them needs to write 0x59, 0x16, and 0x88 to address 0x5000_0100 to disable register protection. Refer to the register SYS_REGLCTL at address SYS_BA + 0x100. **Note3:** To set CLK_PWRCTL[1:0] to select HXT or LXT crystal clock.

Clock Source Select Control Register 1 (CLK_CLKSEL1)

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

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

31	30	29	28	27	26	25	24
PWMCH23SEL		PWMCH01SEL		Reserved		UARTSEL	
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved	TMR1SEL			Reserved	TMR0SEL		
7	6	5	4	3	2	1	0
Reserved		SPISEL		ADCSEL		WDTSEL	

Bits	Description	
[31:30]	**PWMCH23SEL**	**PWM2 And PWM3 Clock Source Selection** PWM2 and PWM3 use the same peripheral clock source; Both of them use the same prescaler. 00 = Reserved. 01 = Reserved. 10 = Clock source is from HCLK. 11 = Reserved.
[29:28]	**PWMCH01SEL**	**PWM0 And PWM1 Clock Source Selection** PWM0 and PWM1 use the same peripheral clock source. Both of them use the same prescaler. 00 = Reserved. 01 = Reserved. 10 = Clock source is from HCLK. 11 = Reserved.
[27:26]	**Reserved**	Reserved.
[25:24]	**UARTSEL**	**UART Clock Source Selection** 00 = Clock source is from HXT or LXT. 01 = Clock source is from PLL. 10 = Clock source is from HIRC. 11 = Clock source is from HIRC. **Note:** To set CLK_PWRCTL[1:0], select HXT or LXT crystal clock.
[23:15]	**Reserved**	Reserved.

Bits	Description	
[14:12]	**TMR1SEL**	**TIMER1 Clock Source Selection** 000 = Clock source is from HXT or LXT. 001 = Clock source is from LIRC. 010 = Clock source is from HCLK. 011 = Clock source is from external trigger. 111 = Clock source is from HIRC. Others = Reserved. **Note:** To set CLK_PWRCTL[1:0], select HXT or LXT crystal clock.
[11]	**Reserved**	Reserved.
[10:8]	**TMR0SEL**	**TIMER0 Clock Source Selection** 000 = Clock source is from HXT or LXT. 001 = Clock source is from LIRC. 010 = Clock source is from HCLK. 011 = Clock source is from external trigger. 111 = Clock source is from HIRC. Others = Reserved. **Note:** To set CLK_PWRCTL[1:0], select HXT or LXT crystal clock.
[7:6]	**Reserved**	Reserved.
[5:4]	**SPISEL**	**SPI Clock Source Selection** 00 = Clock source is from HXT or LXT. 01 = Clock source is from HCLK. 10 = Clock source is from PLL. 11 = Reserved. **Note:** To set CLK_PWRCTL[1:0], select HXT or LXT crystal clock.
[3:2]	**ADCSEL**	**ADC Peripheral Clock Source Selection** 00 = Clock source is from HXT or LXT. 01 = Clock source is from PLL. 10 = Clock source is from HCLK. 11 = Clock source is from HIRC. **Note:** To set CLK_PWRCTL[1:0], select HXT or LXT crystal clock.
[1:0]	**WDTSEL**	**WDT CLK Clock Source Selection (Write Protect)** 00 = Clock source is from HXT or LXT. 01 = Reserved. 10 = Clock source is from HCLK/2048 clock. 11 = Clock source is from LIRC. **Note1:** These bits are the protected bit, and programming them needs to write 0x59, 0x16, and 0x88 to address 0x5000_0100 to disable register protection. Refer to the register SYS_REGLCTL at address SYS_BA + 0x100. **Note2:** To set CLK_PWRCTL[1:0], select HXT or LXT crystal clock.

Clock Divider Register (CLK_CLKDIV)

Register	Offset	R/W	Description	Reset Value
CLK_CLKDIV	CLK_BA+0x18	R/W	Clock Divider Number Register	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
Reserved				HCLKDIV			

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:16]	**ADCDIV**	**ADC Peripheral Clock Divide Number From ADC Peripheral Clock Source** ADC peripheral clock frequency = (ADC peripheral clock source frequency) / (ADCDIV + 1).
[15:12]	**Reserved**	Reserved.
[11:8]	**UARTDIV**	**UART Clock Divide Number From UART Clock Source** UART clock frequency = (UART clock source frequency) / (UARTDIV + 1).
[7:4]	**Reserved**	Reserved.
[3:0]	**HCLKDIV**	**HCLK Clock Divide Number From HCLK Clock Source** HCLK clock frequency = (HCLK clock source frequency) / (HCLKDIV + 1).

Clock Source Select Control Register (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_00EF

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved						WWDTSEL	
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved		PWMCH45SEL		FREQSEL		Reserved	

Bits	Description	
[31:18]	**Reserved**	Reserved.
[17:16]	**WWDTSEL**	**Window Watchdog Timer Clock Source Selection** 00 = Reserved. 01 = Reserved. 10 = Clock source from HCLK/2048 clock. 11 = Clock source from 10 kHz internal low speed RC oscillator (LIRC).
[15:6]	**Reserved**	Reserved.
[5:4]	**PWMCH45SEL**	**PWM4 And PWM5 Clock Source Selection** PWM4 and PWM5 use the same peripheral clock source; Both of them use the same prescaler. 00 = Reserved. 01 = Reserved. 10 = Clock source is from HCLK. 11 = Reserved.
[3:2]	**FREQSEL**	**Clock Divider Clock Source Selection** 00 = Clock source is from HXT or LXT. 01 = Clock source is from LIRC. 10 = Clock source is from HCLK. 11 = Clock source is from HIRC. **Note:** To set CLK_PWRCTL[1:0], select HXT or LXT crystal clock.
[1:0]	**Reserved**	Reserved.

PLL Control Register (CLK_PLLCTL)

The PLL reference clock input is from the 4~24 MHz external high speed crystal (HXT) clock input or from the 22.1184 MHz internal high speed RC oscillator (HIRC). These registers are 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	0x0005_C230

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
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	
[19]	PLLSRC	**PLL Source Clock Selection** 0 = PLL source clock from HXT. 1 = PLL source clock from HIRC.
[18]	OE	**PLL OE (FOUT Enable) Pin Control** 0 = PLL FOUT Enabled. 1 = PLL FOUT is fixed low.
[17]	BP	**PLL Bypass Control** 0 = PLL is in Normal mode (default). 1 = PLL clock output is same as PLL source clock input.
[16]	PD	**Power-down Mode** If the PDEN bit is set to 1 in CLK_PWRCTL register, 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	**PLL Output Divider Control** Refer to the formulas below the table.
[13:9]	INDIV	**PLL Input Divider Control** Refer to the formulas below the table.
[8:0]	FBDIV	**PLL Feedback Divider Control** Refer to the formulas below the table.

PLL Output Clock Frequency Setting: $$FOUT = FIN \times \frac{NF}{NR} \times \frac{1}{NO}$$

Constraint:

1. $4MHz < FIN < 24MHz$
2. $800KHz < Fref = \frac{FIN}{2 \times NR} < 7.5MHz$
3. $100MHz < FCO = Fref \times 2 \times NF = FIN \times \frac{NF}{NR} < 200MHz$
   $120MHz < FCO \ is \ preferred$

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

Default Frequency Setting

The default value: 0xC230
FIN = 12 MHz
NR = (1+2) = 3
NF = (48+2) = 50
NO = 4
FOUT = 12/4 * 50 * 1/3 = 50 MHz

Frequency Divider Control Register (CLK_CLKOCTL)

Register	Offset	R/W	Description	Reset Value
CLK_CLKOCTL	CLK_BA+0x24	R/W	Frequency Divider 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		DIV1EN	CLKOEN	FREQSEL			

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5]	**DIV1EN**	**Frequency Divider One Enable Bit** 0 = Divider output frequency is depended on FREQSEL value. 1 = Divider output frequency is the same as input clock frequency.
[4]	**CLKOEN**	**Frequency Divider Enable Bit** 0 = Frequency Divider Disabled. 1 = Frequency Divider Enabled.
[3:0]	**FREQSEL**	**Divider Output Frequency Selection** The formula of output frequency is $\mathbf{F_{out} = F_{in}/2^{(N+1)}}$. F_{in} is the input clock frequency. F_{out} is the frequency of divider output clock. N is the 4-bit value of FREQSEL(CLK_CLKOCTL[3:0]).

6.4 Flash Memory Controller (FMC)

6.4.1 Overview

The NuMicro® Mini58 series is equipped with 32 Kbytes on-chip embedded flash for application and Data Flash to store some application dependent data. A User Configuration block provides for system initialization. A 2.5 Kbytes loader ROM (LDROM) is used for In-System-Programming (ISP) function. A 512 bytes security protection ROM (SPROM) can conceal user program. 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 32 Kbytes application ROM (APROM).
- Supports 2.5 Kbytes loader ROM (LDROM).
- Supports configurable Data Flash size to share with APROM.
- Supports 512 bytes security protection ROM (SPROM) to conceal user program.
- Supports 12 bytes User Configuration block to control system initialization.
- Supports 512 bytes page erase for all embedded flash.
- Supports CRC-32 checksum calculation function.
- Supports In-System-Programming (ISP) / In-Application-Programming (IAP) to update embedded flash memory.

6.4.3 Block Diagram

The flash memory controller (FMC) consists of AHB slave interface, flash control registers, flash initialization controller, flash operation control and embedded flash memory. Figure 6.4-1 shows the block diagram of flash memory controller.

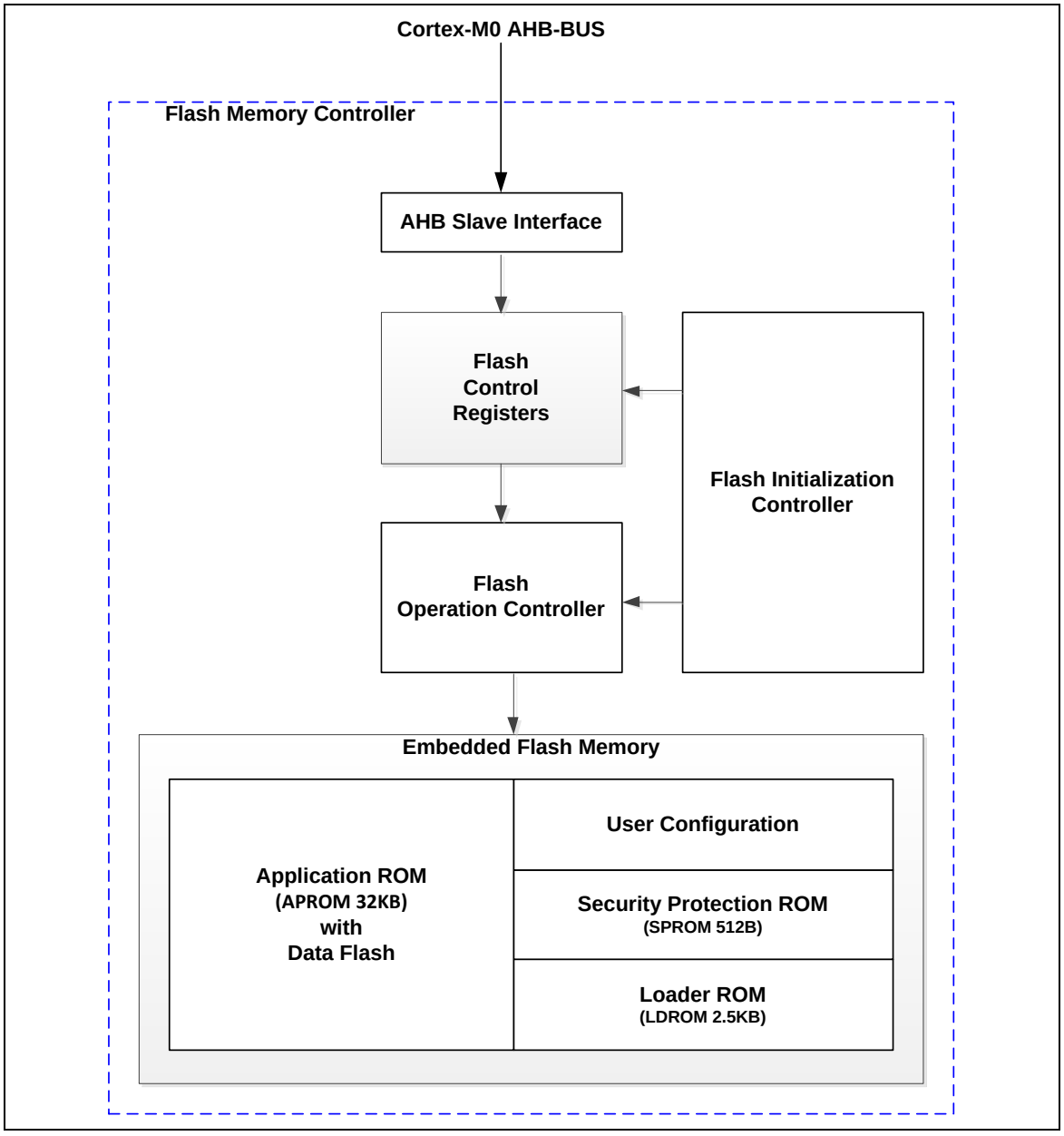


Figure 6.4-1 Flash Memory Control Block Diagram

AHB Slave Interface

There is single AHB slave interfaces in flash memory controller for Cortex®-M0 to perform the instruction, data fetch and ISP control 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 power 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 initialization.

Flash Operation Controller

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

Embedded Flash Memory

The embedded flash memory is the main memory for user application code and parameters. It is consists of the user configuration block, 2.5 KB LDROM, 512B SPROM and 32 KB APROM with Data Flash. The page erase flash size is 512B, and program bit width is 32 bits.

6.4.4 Functional Description

The 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. The embedded flash memory 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.

6.4.4.2 LDROM, APROM and Data Flash

LDROM is designed for a loader to implement In-System-Programming (ISP) function by user. LDROM is a 2.5 KB embedded flash memory, the flash address range is from 0x0010_0000 to 0x0010_09FF. APROM is main memory for user applications. APROM size is 32 Kbytes. Data Flash is used to store application parameters (not instruction). Data Flash is shared with APROM and size is configurable. The base address of Data Flash is determined by DFBA (CONFIG1[19:0]). All of embedded flash memory is 512 bytes page erased.

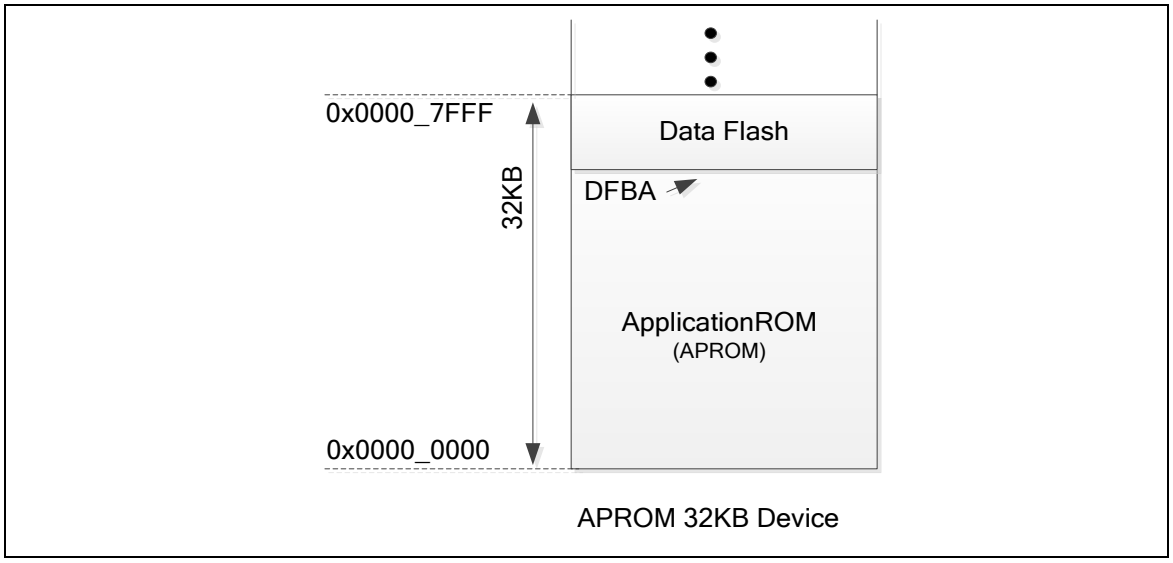


Figure 6.4-2 Data Flash Shared with APROM

6.4.4.3 User Configuration Block

User Configuration block is internal programmable configuration area for boot options, such as flash security lock, boot select, 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 requests. User Configuration block can be updated by ISP function and located at 0x0030_0000 with three 32 bits words (CONFIG0, CONFIG1 and CONFIG2). Any change on User Configuration block will take effect after system reboot.

CONFIG0 (Address = 0x0030_0000)

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

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23]	**CBOVEXT**	**Brown-out Voltage Select Extension** 0 = Brown-out voltage selection includes 2.2V/2.7V/3.7V/4.4V. 1 = Brown-out voltage selection includes 2.7V/3.7V/Disable mode. **Note:** See the table in CBOV.
[22:21]	**CBOV**	**Brown-out Voltage Selection** CBOVEXT / CBOV[1:0] / Brown-out voltage 0 / 00 / 2.2V 0 / 01 / 2.7V 0 / 10 / 3.7V 0 / 11 / 4.4V 1 / 00 / 2.7V 1 / 01 / 2.7V 1 / 10 / 3.7V 1 / 11 / Disable
[20]	**CBORST**	**Brown-out Reset Enable Bit** 0 = Brown-out reset Enabled after powered on or active from reset pin. 1 = Brown-out reset Disabled after powered on or active from reset pin.
[19:11]	**Reserved**	Reserved.
[10]	**CIOINI**	**I/O Initial State Selection** 0 = All GPIO set as Quasi-bidirectional mode after chip powered on or active from reset pin. 1 = All GPIO set as input tri-state mode after powered on or active from reset pin. **Note:** It is recommended to use 100 kΩ pull-up resistor on both ICE_DAT and ICE_CLK pin.
[9:8]	**Reserved**	Reserved.

[7:6]	**CBS**	**Chip Booting Selection** When CBS[0] = 0 with IAP mode, 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, the code in LDROM and APROM can be called by each other. CBS value is valid. 00 = Boot from LDROM with IAP mode. 01 = Boot from LDROM without IAP mode. 10 = Boot from APROM with IAP mode. 11 = Boot from APROM without IAP mode. **Note:** BS (FMC_ISPCTL[1]) is only be used to control boot switching when CBS[0] = 1. VECMAP (FMC_ISPSTS[23:9]) is only be used to remap 0x0~0x1ff when CBS[0] = 0.
[5:2]	**Reserved**	Reserved.
[1]	**LOCK**	**Security Lock Control** 0 = Flash memory content is locked. 1 = Flash memory content is locked except ALOCK (CONFIG2[7:0]) is 0x5A.
[0]	**DFEN**	**Data Flash Enable Bit** 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. 0 = Data Flash Enabled. 1 = Data Flash Disabled.

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	Description	
[31:20]	**Reserved**	Reserved.
[19:0]	**DFBA**	**Data Flash Base Address** This register works only when DFEN (CONFIG0[0]) is 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 512 bytes, it is mandatory to keep bit 8-0 as 0.

CONFIG2 (Address = 0x0030_0008)

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

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

6.4.4.4 Security Protection Memory (SPROM)

The security protection memory (SPROM) is used to store instructions for security application. The SPROM includes 512 bytes at location address 0x20_0000 ~ 0x20_01FF and doesn't support "whole chip erase command". Figure 6.4-3 shows that the last byte of SPROM (address: 0x0020_01FF) is used to identify the SPROM code is non-secured, debug secured or secured mode.

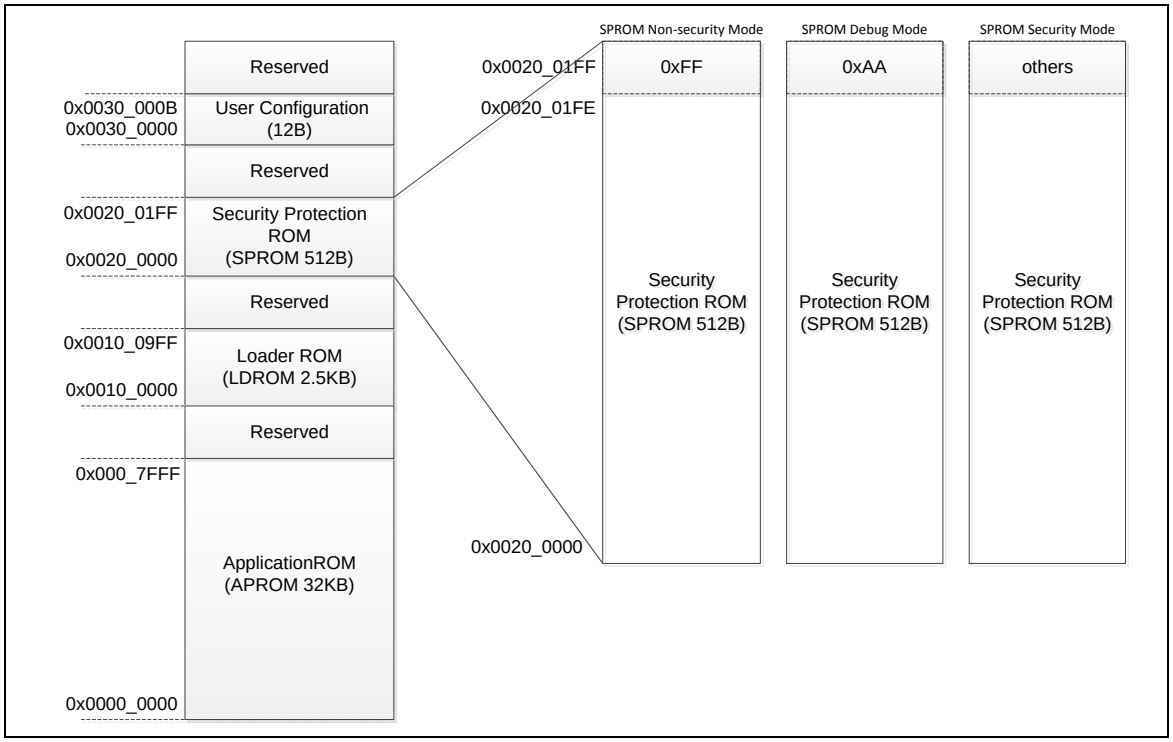


Figure 6.4-3 SPROM Security Mode

(1) SPROM non-secured mode (the last byte is 0xFF). The access behavior of SPROM is the same with APROM and LDROM. All area can be read by CPU or ISP command, and can be erased and programmed by ISP command.

(2) SPROM debug secured mode (the last byte is 0xAA). In order to debug easily, FMC controller accepts to execute program of SPROM when Cortex®-M0 ICE (In-Circuit-Emulator) port is connected. Other behaviors of SPROM are the same with SPROM secured mode.

(3) SPROM secured mode (the last byte is not 0xFF or 0xAA). In order to conceal SPROM code in secured mode, CPU only can perform instruction fetch and get data from SPROM when CPU is run at SPROM area. Otherwise, CPU will get all zero (0x0000_0000) for data access. In order to protect SPROM, the CPU instruction fetch will also get zero value when Cortex®-M0 ICE (In-Circuit-Emulator) port is connected in secured code. At this mode, SPROM doesn't support ISP program and read flash command and only supports page erase command with 0x0055AA03 data.

The SCODE (FMC_ISPSTS[31]) is SPROM secured flag to indicate that SPROM keeps secured mode, debug secured mode or not. It is set to 1 at flash initialization if the last byte of SPROM isn't 0xFF, and can be cleared after the SPROM page erase operation complete. In order to easily test SPROM secured mode at normal run, it also can be set to 1 by user if the last byte of SPROM is 0xFF.

6.4.4.5 Flash Memory Map

In the Mini58 series, the flash memory map is different to system memory map. The system memory map is used by CPU fetch code or data from FMC memory. The flash memory map is used for ISP function to read, program or erase FMC memory. Figure 6.4-4 shows the flash memory map.

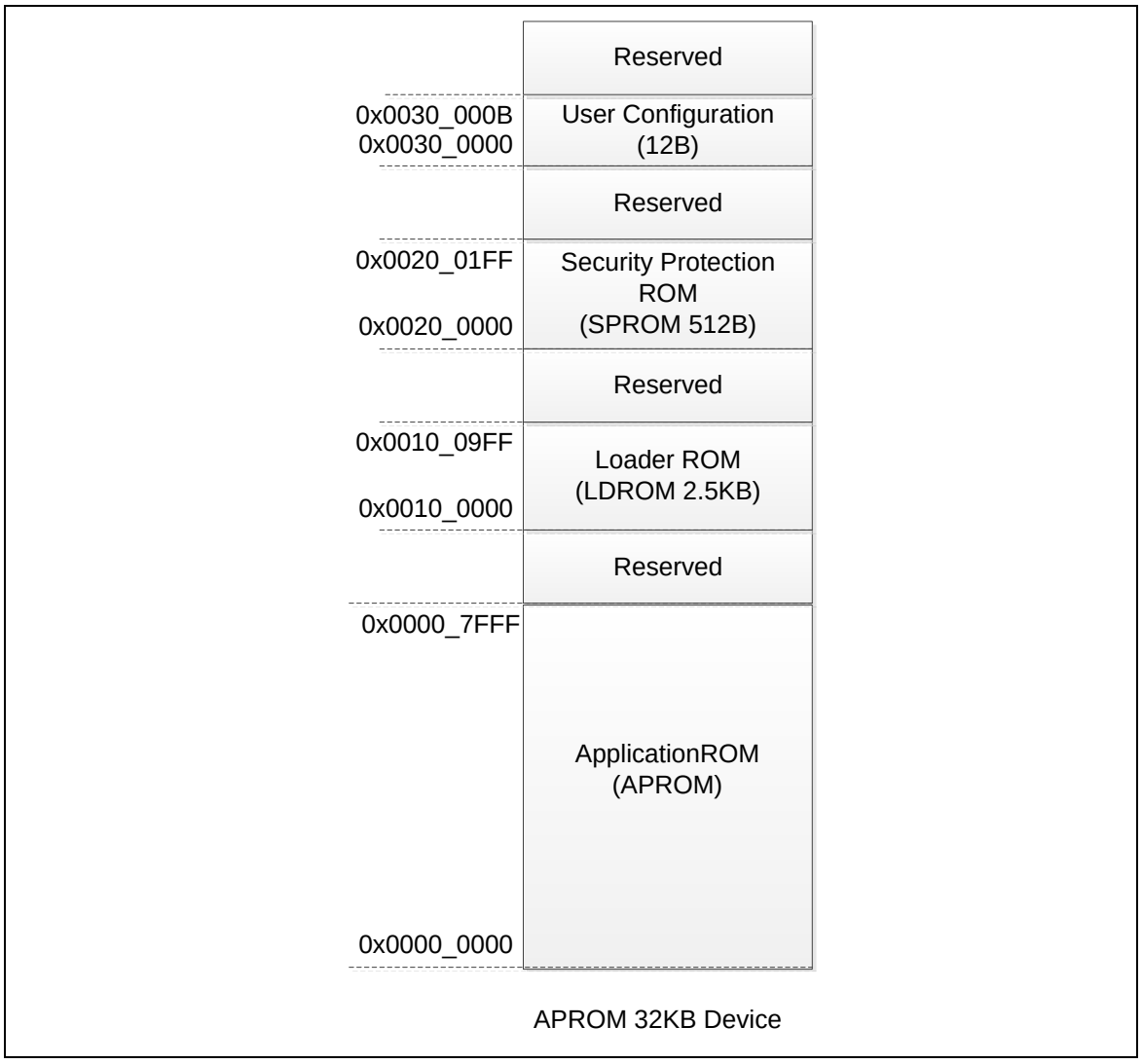


Figure 6.4-4 Flash Memory Map

6.4.4.6 System Memory Map with IAP Mode

The system memory map is used by CPU to fetch code or data from FMC memory. SPROM(0x0020_0000~0x0020_01FF) and LDROM(0x0010_0000~0x0010_09FF) address map are the same as in the flash memory map. 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 initialization. The DFBA~0x0000_7FFF is the Data Flash region for Cortex®-M0 data access, and 0x0000_0200~(DFBA-1) is APROM region for Cortex®-M0 instruction access.

The address from 0x0000_0000 to 0x0000_01FF is called system memory vector. APROM and LDROM can map to the system memory vector for CPU start up. There are two kinds of system memory map with IAP mode when chip booting: (1) LDROM with IAP, and (2) APROM with IAP.

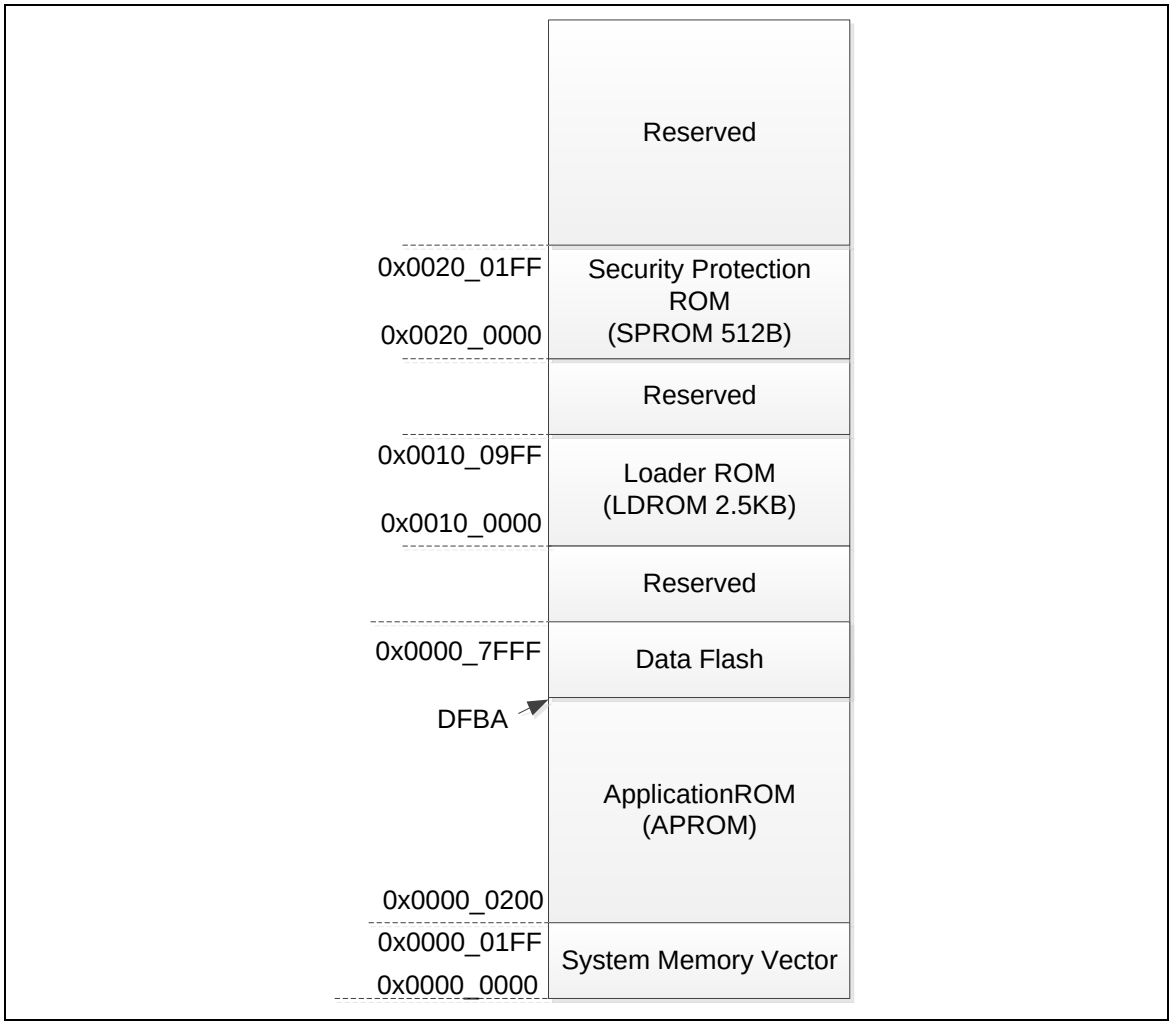


Figure 6.4-5 System Memory Map with IAP Mode

In LDROM with IAP mode, the default value of {VECMAP[11:0], 9'h000} is 0x100000 and first page of LDROM (0x0010_0000 ~ 0x0010_01FF) is mapping to the system memory vector for Cortex®-M0 instruction or data access.

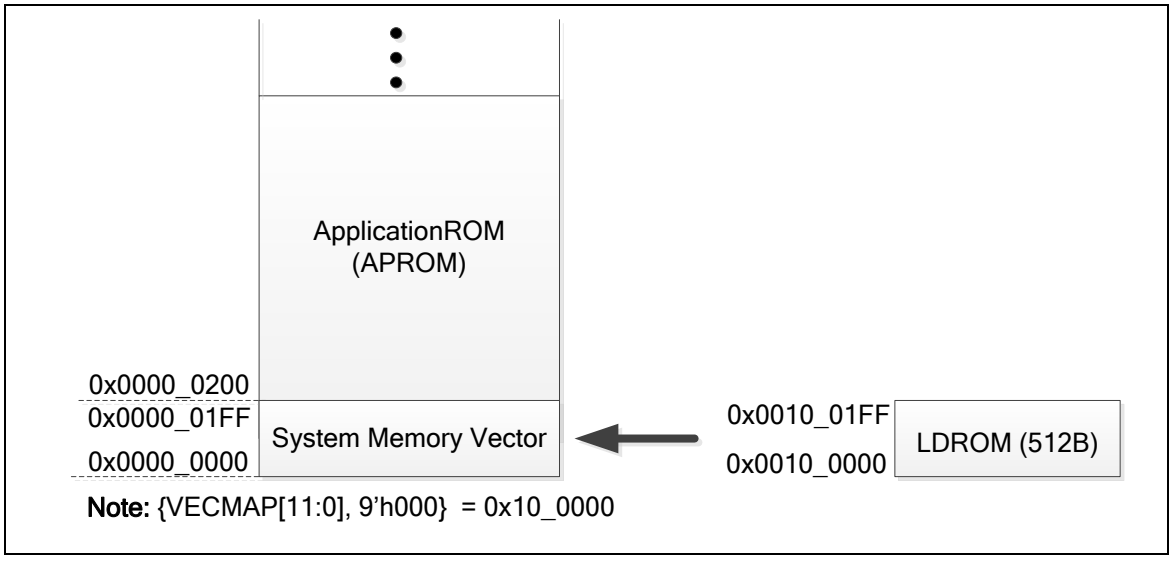


Figure 6.4-6 LDROM with IAP Mode

In APROM with IAP mode, the default value of {VECMAP[11:0], 9'h000} is 0x000000 and first page of APROM (0x0000_0000~0x0000_01FF) is mapping to the system memory vector for Cortex®-M0 instruction or data access.

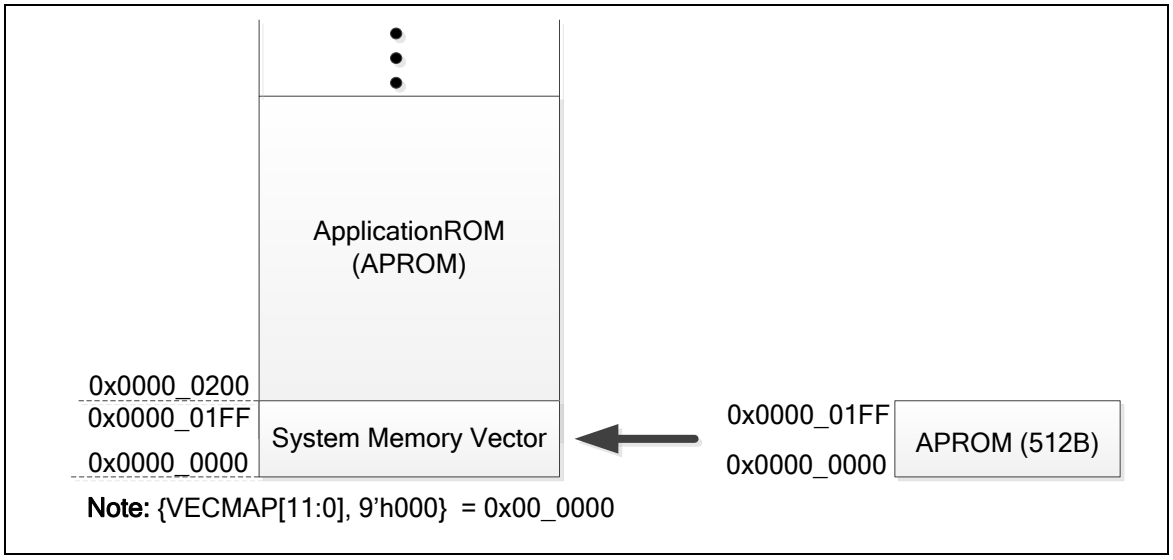


Figure 6.4-7 APROM with IAP Mode

In system memory map with IAP mode, APROM and LDROM can remap to the system memory vector when CPU running. User can write the target remap address to FMC_ISPADDR register and then trigger ISP procedure with the "Vector Page Remap" command (0x2E). In VECMAP (FMC_ISPSTS[23:9]), shows the finial system memory vector mapping address.

6.4.4.7 System Memory Map without IAP mode

In system memory map without IAP mode, CPU still can access

SPROM(0x0020_0000~0x0020_01FF), but the system memory vector mapping is not supported. There are two kinds of system memory map without IAP mode when chip booting: (1) LDROM without IAP, (2) APROM without IAP. In LDROM without IAP mode, LDROM base is mapping to 0x0000_0000. CPU program cannot run to access APROM. In APROM without IAP mode, APROM base is mapping to 0x0000_0000. CPU program cannot run to access LDROM. 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 initialization. The DFBA0x0000_7FFF is the Data Flash region for Cortex®-M0 data access, and 0x0000_0000~(DFBA-1) is APROM region for Cortex®-M0 instruction access.

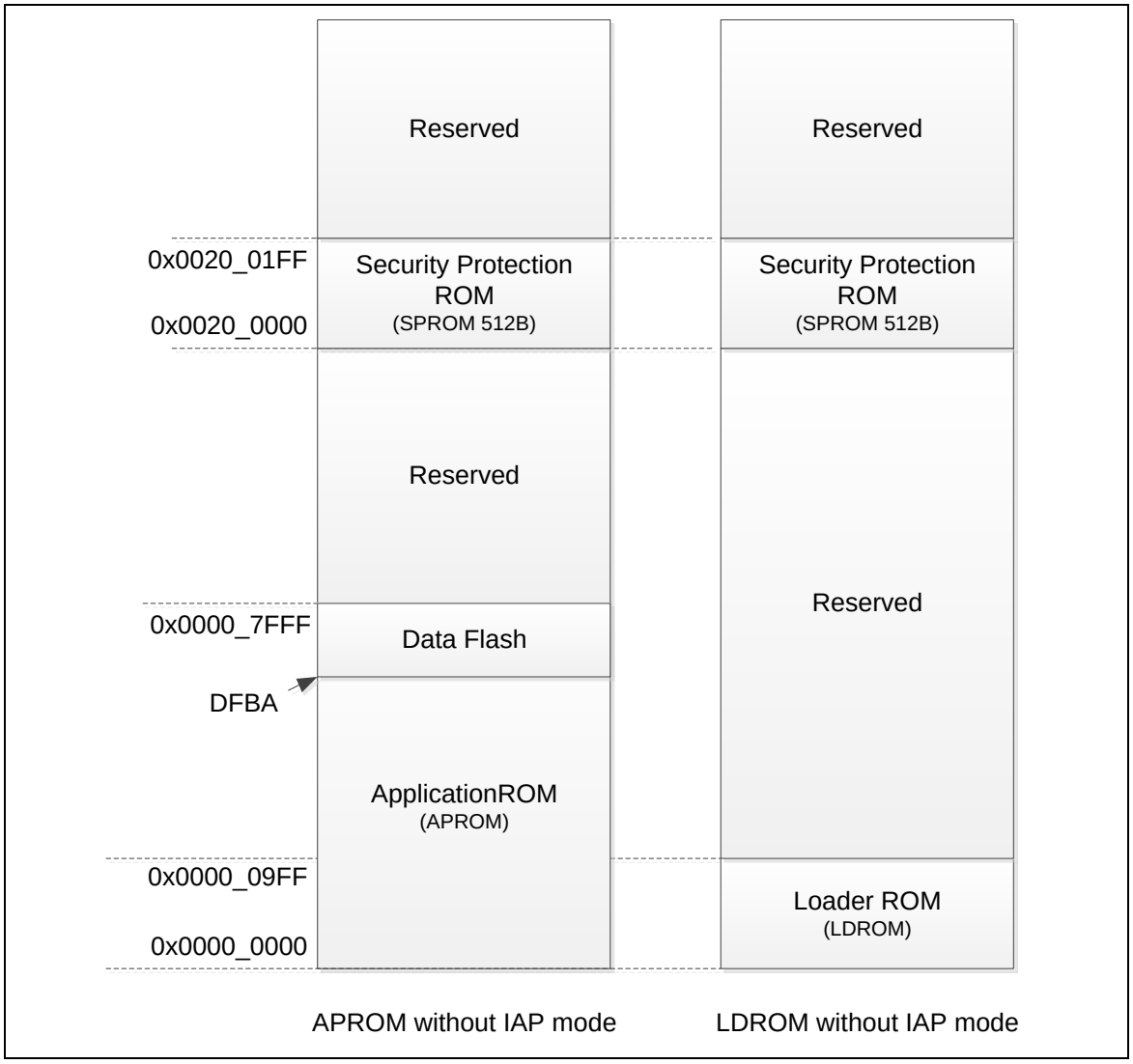


Figure 6.4-8 System Memory Map without IAP Mode

6.4.4.8 Boot Selection

The Mini58 provides four booting sources for user select. They are 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]).

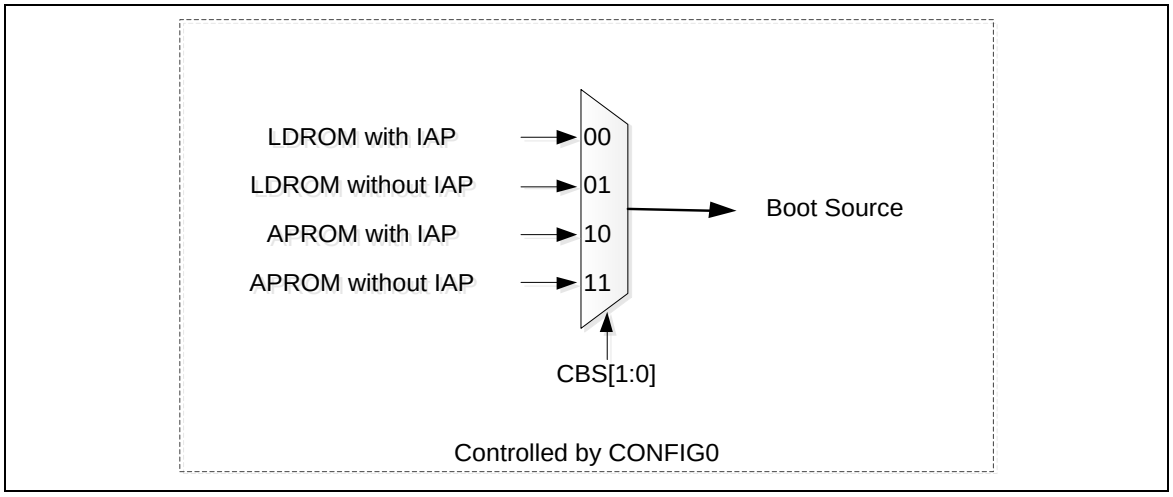


Figure 6.4-9 Boot Source Selection

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

Table 6.4-1 Vector Mapping Support Table

6.4.4.9 In-Application-Programming (IAP)

The Mini58 Series provides In-Application-Programming (IAP) function for user to switch the code executing between APROM, LDROM and SPROM. User can enable the IAP function by booting chip and setting the chip boot selection bits in CBS (CONFIG0[7:6]) as 10 or 00.

When chip boots with IAP function enabled, any executable code (align to 512 bytes) is allowed to map to the system memory vector(0x0000_0000~0x0000_01FF) any time. User can change the remap address to FMC_ISPADDR and then trigger ISP procedure with the “Vector Page Remap” command.

6.4.4.10 In-System-Programming (ISP)

The Mini58 series supports In-System-Programming (ISP) function allowing the embedded flash memory to be reprogrammed under software control. ISP is performed without removing the microcontroller from the system through the firmware and on-chip connectivity interface, such as UART, I^2C, and SPI.

The Mini58 ISP provides the following functions for embedded flash memory.

- Supports flash page erase function
- Supports flash data program function
- Supports flash data read function
- Supports company ID read function
- Supports device ID read function

- Supports unique ID read function
- Supports memory checksum calculation function
- Supports system memory vector remap function

ISP Commands

ISP Command	FMC_ISPCMD	FMC_ISPADDR	FMC_ISPDAT
FLASH Page Erase	0x22	Valid address of flash memory origination. It must be 512 bytes page alignment.	N/A
SPROM Page Erase	0x22	0x0020_0000	0x0055_AA03
FLASH 32-bit Program	0x21	Valid address of flash memory origination	FMC_ISPDAT: Programming Data
FLASH Read	0x00	Valid address of flash memory origination	FMC_ISPDAT: Return Data
Read Company ID	0x0B	0x0000_0000	FMC_ISPDAT: 0x0000_00DA
Read Checksum	0x0D	Keep address of “Run Checksum Calculation”	FMC_ISPDAT: Return Checksum
Run Checksum Calculation	0x2D	Valid start address of memory origination 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 or LDROM. It must be 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. Set FMC_ISPADDR for target flash memory based on flash memory origination. FMC_ISPDAT can be used to set the data to program or used to return the read data according to FMC_ISPCMD.

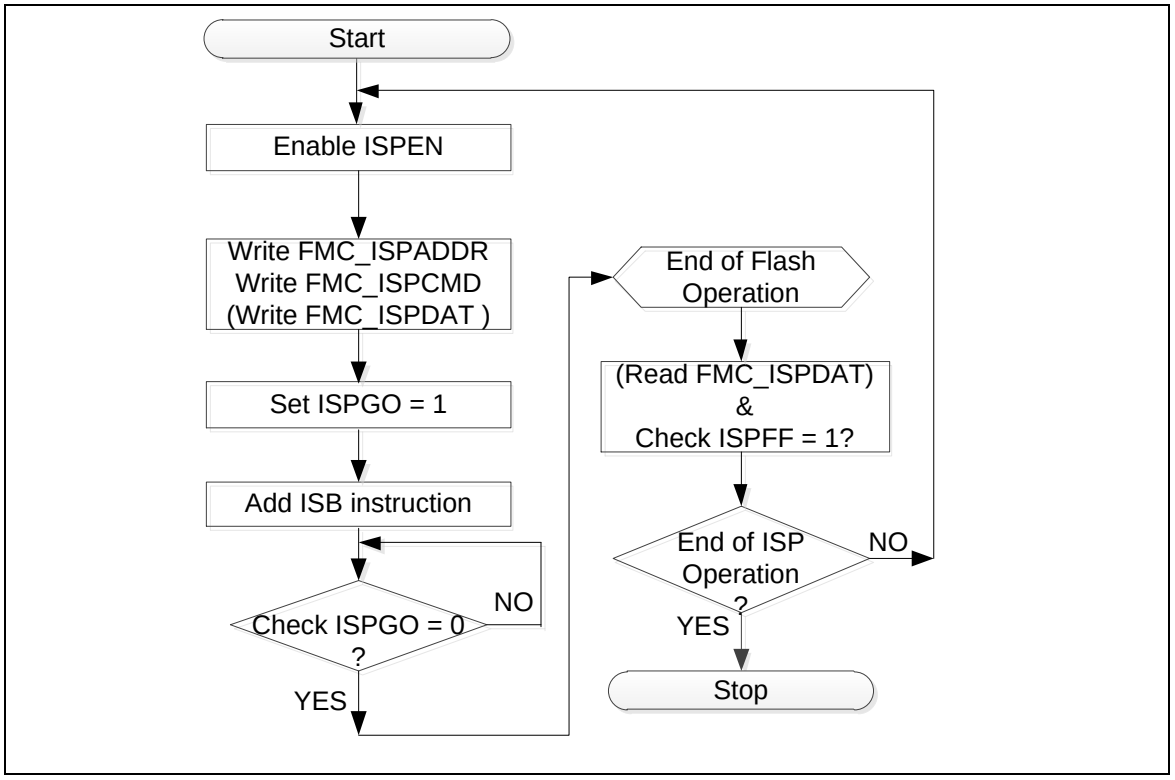


Figure 6.4-10 ISP Procedure Example

Finally, set ISPGO (FMC_ISPTRG[0]) register to perform 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.

When the ISPGO (FMC_ISPTRG[0]) bit is set, CPU will wait for ISP operation to finish during this period; the peripheral still keeps working as usual. If any interrupt request occurs, CPU will not service it till ISP operation is finished. 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.

6.4.4.11 CRC32 Checksum Calculation

The NuMicro® Mini58 series supports the Cyclic Redundancy Check (CRC-32) checksum calculation function to help user quickly check the memory content includes APROM, LDROM and SPROM. The CRC-32 polynomial is below

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

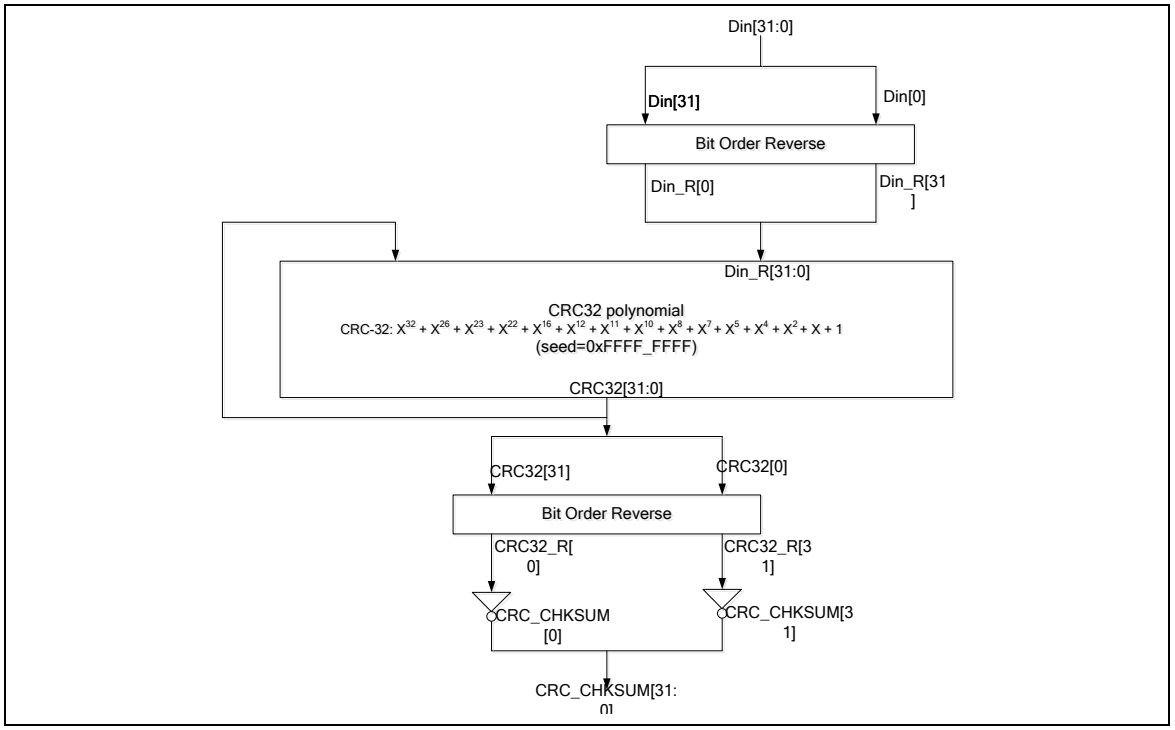


Figure 6.4-11 CRC-32 Checksum Calculation

The following three steps complete the CRC-32 checksum calculation.

1. Perform ISP “Run Memory Checksum” operation: 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 must be multiples of 512 bytes and the starting address includes APROM, LDROM and SPROM.
2. Perform ISP “Read Memory Checksum” operation: the FMC_ISPADDR should be kept as the same as step 1.
3. Read FMC_ISPDAT to get checksum: The checksum is read from FMC_ISPDAT. If the checksum is 0x0000_0000, It must be one of two conditions (1) Checksum calculation is in-progress, (2) Address and size is over device limitation.

When SPROM is set to security mode, CPU and ISP read command cannot read the SPROM content directly but user can use this checksum function to verify SPROM content correction.

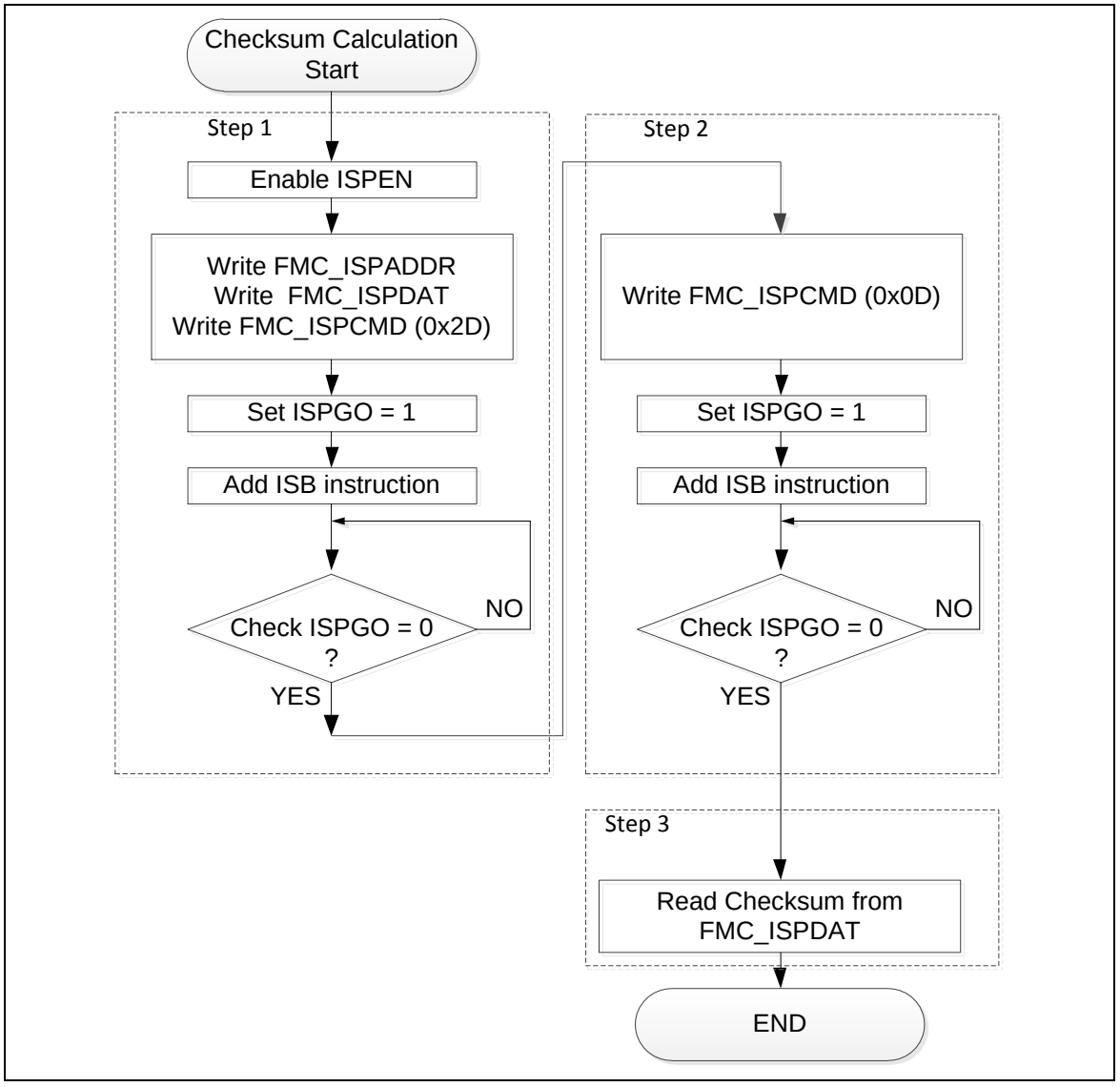


Figure 6.4-12 CRC-32 Checksum Calculation Flow

6.4.5 Flash Control Register Map

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

Register	Offset	R/W	Description	Reset Value
FMC Base Address: **FMC_BA = 0x5000_C000**				
FMC_ISPCTL	FMC_BA+0x00	R/W	ISP Control Register	0x0000_000X
FMC_ISPADDR	FMC_BA+0x04	R/W	ISP Address Register	0x0000_0000
FMC_ISPDAT	FMC_BA+0x08	R/W	ISP Data Register	0x0000_0000
FMC_ISPCMD	FMC_BA+0x0C	R/W	ISP Command Register	0x0000_0000
FMC_ISPTRG	FMC_BA+0x10	R/W	ISP Trigger Control Register	0x0000_0000
FMC_DFBA	FMC_BA+0x14	R	Data Flash Base Address	0xXXXX_XXXX
FMC_FATCTL	FMC_BA+0x18	R/W	Flash Access Time Control Register	0x0000_0000
FMC_ISPSTS	FMC_BA+0x40	R/W	ISP Status Register	0xX0X0_000X

6.4.6 Flash Control 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_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	ISPFF	LDUEN	CFGUEN	APUEN	SPUEN	BS	ISPEN

Bits	Description	
[31:7]	**Reserved**	Reserved.
[6]	**ISPFF**	**ISP Fail Flag (Write Protect)** This bit is set by hardware when a triggered ISP meets any of the following conditions: This bit needs to be cleared by writing 1 to it. (1) APROM writes to itself if APUEN is set to 0. (2) LDROM writes to itself if LDUEN is set to 0. (3) CONFIG is erased/programmed if CFGUEN is set to 0. (4) SPROM is erased/programmed if SPUEN is set to 0. (5) SPROM is programmed at SPROM secured mode. (6) Page Erase command at LOCK mode with ICE connection. (7) Erase or Program command at brown-out detected. (8) Destination address is illegal, such as over an available range. (9) Invalid ISP commands.
[5]	**LDUEN**	**LDROM Update Enable (Write Protect)** LDROM update enable bit. 0 = LDROM cannot be updated. 1 = LDROM can be updated.
[4]	**CFGUEN**	**CONFIG Update Enable Bit (Write Protect)** 0 = CONFIG cannot be updated. 1 = CONFIG can be updated.
[3]	**APUEN**	**APROM Update Enable Bit (Write Protect)** 0 = APROM cannot be updated when the chip runs in APROM. 1 = APROM can be updated when the chip runs in APROM.

[2]	**SPUEN**	**SPROM Update Enable Bit (Write Protect)** 0 = SPROM cannot be updated. 1 = SPROM can be updated.
[1]	**BS**	**Boot Select (Write Protect)** 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 (RSTS_CPU is 1) or system reset (RSTS_SYS) is happened 0 = Booting from APROM. 1 = Booting from LDROM.
[0]	**ISPEN**	**ISP Enable Bit (Write Protect)** Set this bit to enable the ISP function. 0 = ISP function Disabled. 1 = ISP function Enabled.

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** The Mini58 series is equipped with embedded flash. ISPADDR[1:0] must be kept 00 for ISP 32-bit operation. and ISPADR[8:0] must be kept all 0 for Vector Page Re-map Command For CRC32 Checksum Calculation command, this field is the flash starting address for checksum calculation, 512 bytes alignment is necessary for checksum calculation.

ISP Data Register (FMC_ISPDAT)

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

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

Bits	Description	
[31:0]	**ISPDAT**	**ISP Data** Write data to this register before ISP program operation. Read data from this register after ISP read operation. For Run CRC32 Checksum Calculation command, ISPDAT is the memory size (byte) and 512 bytes alignment. For ISP Read Checksum command, ISPDAT is the checksum result. If ISPDAT = 0x0000_0000, it means that (1) the checksum calculation is in progress, or (2) the memory range for checksum calculation is incorrect.

ISP Command (FMC_ISPCMD)

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

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

Bits	Description	
[31:7]	**Reserved**	Reserved.
[6:0]	**CMD**	**ISP CMD** ISP command table is shown below: 0x00= FLASH Read. 0x04= Read Unique ID. 0x0B= Read Company ID. 0x0C= Read Device ID. 0x0D= Read CRC32 Checksum. 0x21= FLASH 32-bit Program. 0x22= FLASH Page Erase. 0x2D= Run CRC32 Checksum Calculation. 0x2E= Vector Remap. The other commands are invalid.

ISP Trigger Control Register (FMC_ISPTRG)

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

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

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**ISPGO**	**ISP Start Trigger (Write Protect)** Write 1 to start ISP operation and this bit will be cleared to 0 by hardware automatically when ISP operation is finished. 0 = ISP operation is finished. 1 = ISP is progressed.

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** This register indicates Data Flash start address. It is a read only register. The Data Flash is shared with APROM. the content of this register is loaded from CONFIG1 This register is valid when DFEN (CONFIG0[0]) =0 .

Flash Access Time Control Register (FMC_FATCTL)

Register	Offset	R/W	Description	Reset Value
FMC_FATCTL	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							
7	6	5	4	3	2	1	0
Reserved	FOM			Reserved			

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8:7]	**Reserved**	Reserved.
[6:4]	**FOM**	**Frequency Optimization Mode (Write Protect)** The Mini58 series supports adjustable flash access timing to optimize the flash access cycles in different working frequency. 0x1 = Frequency ≤ 24MHz. Others = Frequency ≤ 50MHz.
[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	0xX0X0_000X

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

Bits	Description	
[31]	**SCODE**	**Security Code Active Flag** This bit is set to 1 by hardware when detecting SPROM secured code is active at flash initialization, or software writes 1 to this bit to make secured code active; this bit is only cleared by SPROM page erase operation. 0 = SPROM secured code is inactive. 1 = SPROM secured code is active.
[30:21]	**Reserved**	Reserved.
[20:9]	**VECMAP**	**Vector Page Mapping Address (Read Only)** All access to 0x0000_0000~0x0000_01FF is remapped to the flash memory address {VECMAP[11:0], 9'h000} ~ {VECMAP[11:0], 9'h1FF}
[8:7]	**Reserved**	Reserved.
[6]	**ISPFF**	**ISP Fail Flag (Write Protect)** This bit is the mirror of ISPFF (FMC_ISPCTL[6]), it needs to be cleared by writing 1 to FMC_ISPCTL[6] or FMC_ISPSTS[6]. This bit is set by hardware when a triggered ISP meets any of the following conditions: (1) APROM writes to itself if APUEN is set to 0. (2) LDROM writes to itself if LDUEN is set to 0. (3) CONFIG is erased/programmed if CFGUEN is set to 0. (4) SPROM is erased/programmed if SPUEN is set to 0. (5) SPROM is programmed at SPROM secured mode. (6) Page Erase command at LOCK mode with ICE connection. (7) Erase or Program command at brown-out detected. (8) Destination address is illegal, such as over an available range. (9) Invalid ISP commands.
[5:3]	**Reserved**	Reserved.

[2:1]	**CBS**	**Boot Selection Of CONFIG (Read Only)** This bit is initiated with the CBS (CONFIG0[7:6]) after any reset is happened except CPU reset (RSTS_CPU is 1) or system reset (RSTS_SYS) is happened. 00 = LDROM with IAP mode. 01 = LDROM without IAP mode. 10 = APROM with IAP mode. 11 = APROM without IAP mode.
[0]	**ISPBUSY**	**ISP BUSY (Read Only)** 0 = ISP operation is finished. 1 = ISP operation is busy.

6.5 General Purpose I/O (GPIO)

6.5.1 Overview

The NuMicro® Mini58 series has up to 30 General Purpose I/O pins to be shared with other function pins depending on the chip configuration. These 30 pins are arranged in 6 ports named as P0, P1, P2, P3, P4 and P5. Each of the 30 pins is independent and has the corresponding register bits to control the pin mode function and data.

The I/O type of each pin 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 is stay in input mode and each port data register Px_DOUT[n] resets to 1. For Quasi-bidirectional mode, each I/O pin is equipped with a very weak individual pull-up resistor about 110 kΩ ~ 300 kΩ for V_{DD} is from 5.0 V to 2.5 V.

6.5.2 Features

- Four I/O modes:
  - Quasi-bidirectional mode
  - Push-pull output
  - Open-drain output
  - Input-only with high impendence
- Quasi-bidirectional TTL/Schmitt trigger input mode selected by SYS_Px_MFP[23:16]
- I/O pin configured as interrupt source with edge/level setting
- I/O pin internal pull-up resistor enabled only in Quasi-bidirectional I/O mode
- Enabling the pin interrupt function will also enable the pin wake-up function
- High driver and high sink I/O mode support
- Configurable default I/O mode of all pins after reset by CIOINI (Config0[10]) setting
  - CIOINI = 0, all GPIO pins in Quasi-bidirectional mode after chip reset
  - CIOINI = 1, all GPIO pins in Input tri-state mode after chip reset

6.5.3 Block Diagram

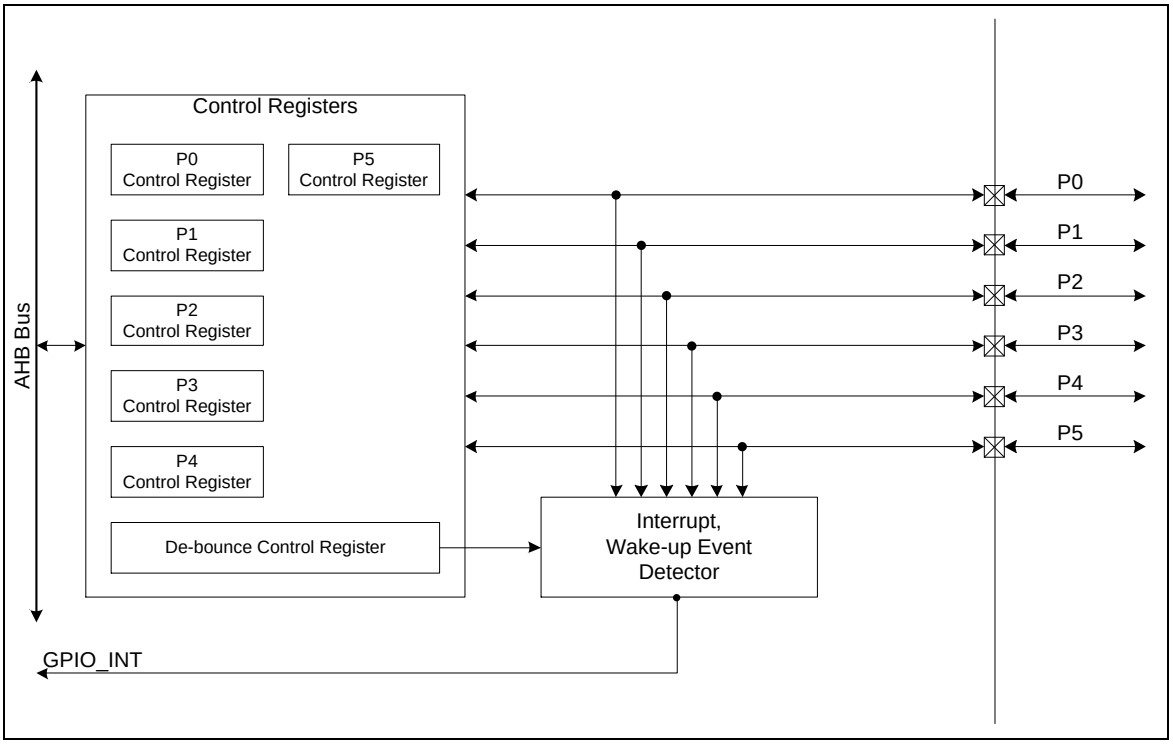


Figure 6.5-1 GPIO Controller Block Diagram

6.5.4 Basic Configuration

The GPIO pin functions are configured in SYS_P0_MFP, SYS_P1_MFP, SYS_P2_MFP, SYS_P3_MFP, SYS_P4_MFP and SYS_P5_MFP registers.

6.5.5 Functional Description

6.5.5.1 Input Mode

Set MODEn (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.

6.5.5.2 Push-pull Output Mode

Set MODEn (Px_MODE[2n+1:2n]) to 01 as Px.n pin is in Push-pull Output mode and the I/O pin supports digital output function with source/sink current capability. The bit value in the corresponding DOUT (Px_DOUT[n]) is driven on the pin.

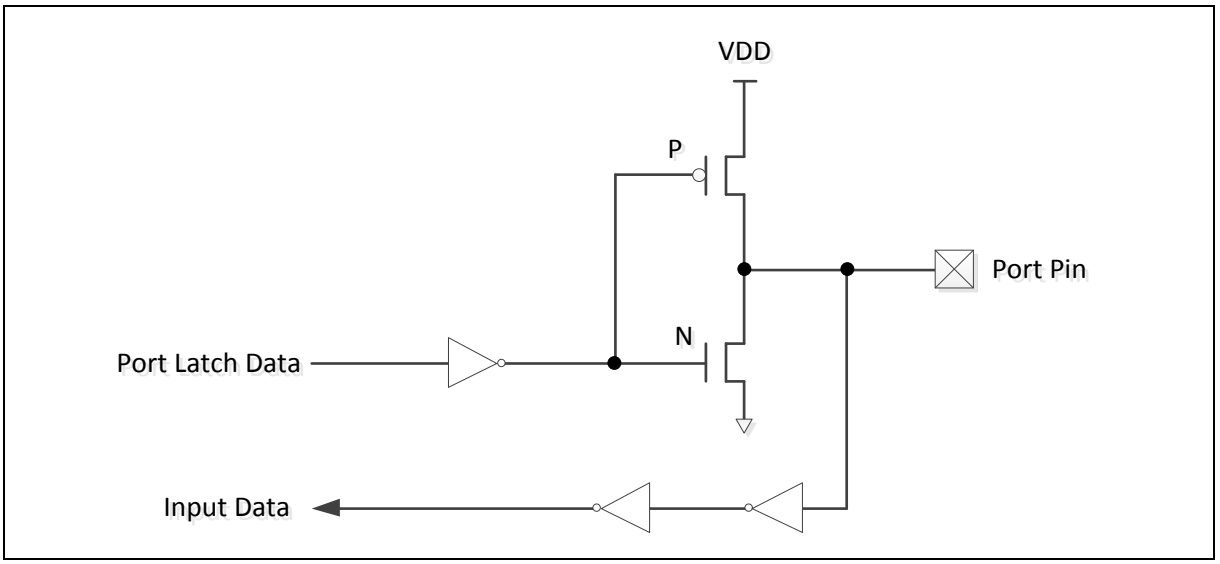


Figure 6.5-2 Push-Pull Output

6.5.5.3 Open-drain Output Mode

Set MODEn (Px_MODE[2n+1:2n]) to 10 as Px.n pin is in Open-drain mode and the digital output function of I/O pin supports only sink current capability, an external pull-up register is needed for driving high state. If the bit value in the corresponding DOUT (Px_DOUT[n]) bit is 0, the pin 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.

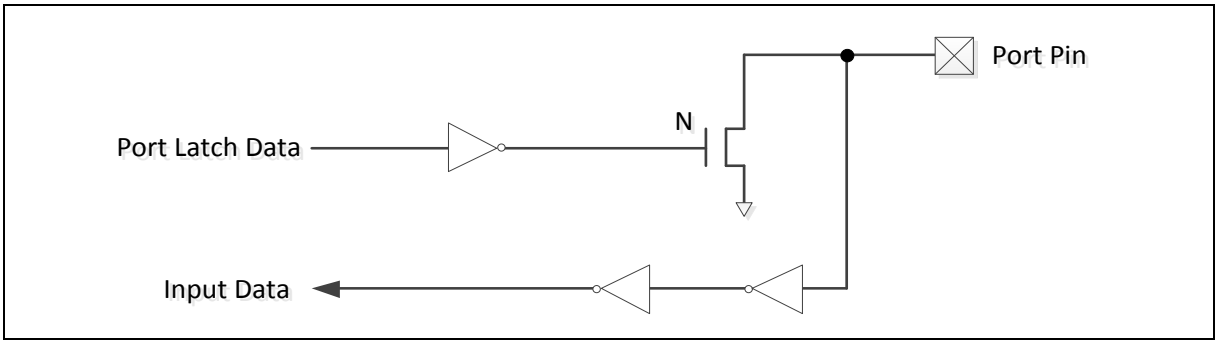


Figure 6.5-3 Open-Drain Output

6.5.5.4 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.

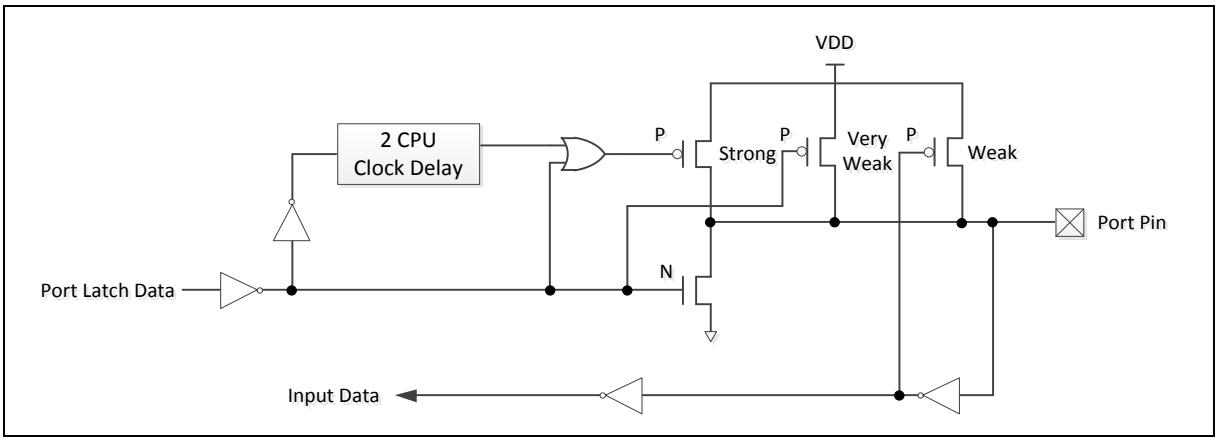


Figure 6.5-4 Quasi-Bidirectional I/O Mode

6.5.6 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.

6.5.7 Register Map

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

Register	Offset	R/W	Description	Reset Value
GPIO Base Address: **GP_BA = 0x5000_4000**				
P0_MODE	GP_BA+0x000	R/W	P0 I/O Mode Control	0x0000_XXXX
P0_DINOFF	GP_BA+0x004	R/W	P0 Digital Input Path Disable Control	0x0000_0000
P0_DOUT	GP_BA+0x008	R/W	P0 Data Output Value	0x0000_00F3
P0_DATMSK	GP_BA+0x00C	R/W	P0 Data Output Write Mask	0x0000_0000
P0_PIN	GP_BA+0x010	R	P0 Pin Value	0x0000_00XX
P0_DBEN	GP_BA+0x014	R/W	P0 De-bounce Enable Control	0x0000_0000
P0_INTTYPE	GP_BA+0x018	R/W	P0 Interrupt Mode Control	0x0000_0000
P0_INTEN	GP_BA+0x01C	R/W	P0 Interrupt Enable Control	0x0000_0000
P0_INTSRC	GP_BA+0x020	R/W	P0 Interrupt Source Flag	0x0000_0000
P1_MODE	GP_BA+0x040	R/W	P1 I/O Mode Control	0x0000_XXXX
P1_DINOFF	GP_BA+0x044	R/W	P1 Digital Input Path Disable Control	0x0000_0000
P1_DOUT	GP_BA+0x048	R/W	P1 Data Output Value	0x0000_003D
P1_DATMSK	GP_BA+0x04C	R/W	P1 Data Output Write Mask	0x0000_0000
P1_PIN	GP_BA+0x050	R	P1 Pin Value	0x0000_00XX
P1_DBEN	GP_BA+0x054	R/W	P1 De-bounce Enable Control	0x0000_0000
P1_INTTYPE	GP_BA+0x058	R/W	P1 Interrupt Mode Control	0x0000_0000
P1_INTEN	GP_BA+0x05C	R/W	P1 Interrupt Enable Control	0x0000_0000
P1_INTSRC	GP_BA+0x060	R/W	P1 Interrupt Source Flag	0x0000_0000
P2_MODE	GP_BA+0x080	R/W	P2 I/O Mode Control	0x0000_XXXX
P2_DINOFF	GP_BA+0x084	R/W	P2 Digital Input Path Disable Control	0x0000_0000
P2_DOUT	GP_BA+0x088	R/W	P2 Data Output Value	0x0000_007C
P2_DATMSK	GP_BA+0x08C	R/W	P2 Data Output Write Mask	0x0000_0000
P2_PIN	GP_BA+0x090	R	P2 Pin Value	0x0000_00XX
P2_DBEN	GP_BA+0x094	R/W	P2 De-bounce Enable Control	0x0000_0000
P2_INTTYPE	GP_BA+0x098	R/W	P2 Interrupt Mode Control	0x0000_0000
P2_INTEN	GP_BA+0x09C	R/W	P2 Interrupt Enable Control	0x0000_0000

P2_INTSRC	GP_BA+0x0A0	R/W	P2 Interrupt Source Flag	0x0000_0000
P3_MODE	GP_BA+0x0C0	R/W	P3 I/O Mode Control	0x0000_XXXX
P3_DINOFF	GP_BA+0x0C4	R/W	P3 Digital Input Path Disable Control	0x0000_0000
P3_DOUT	GP_BA+0x0C8	R/W	P3 Data Output Value	0x0000_0077
P3_DATMSK	GP_BA+0x0CC	R/W	P3 Data Output Write Mask	0x0000_0000
P3_PIN	GP_BA+0x0D0	R	P3 Pin Value	0x0000_00XX
P3_DBEN	GP_BA+0x0D4	R/W	P3 De-bounce Enable Control	0x0000_0000
P3_INTTYPE	GP_BA+0x0D8	R/W	P3 Interrupt Mode Control	0x0000_0000
P3_INTEN	GP_BA+0x0DC	R/W	P3 Interrupt Enable Control	0x0000_0000
P3_INTSRC	GP_BA+0x0E0	R/W	P3 Interrupt Source Flag	0x0000_0000
P4_MODE	GP_BA+0x100	R/W	P4 I/O Mode Control	0x0000_XXXX
P4_DINOFF	GP_BA+0x104	R/W	P4 Digital Input Path Disable Control	0x0000_0000
P4_DOUT	GP_BA+0x108	R/W	P4 Data Output Value	0x0000_00C0
P4_DATMSK	GP_BA+0x10C	R/W	P4 Data Output Write Mask	0x0000_0000
P4_PIN	GP_BA+0x110	R	P4 Pin Value	0x0000_00XX
P4_DBEN	GP_BA+0x114	R/W	P4 De-bounce Enable Control	0x0000_0000
P4_INTTYPE	GP_BA+0x118	R/W	P4 Interrupt Mode Control	0x0000_0000
P4_INTEN	GP_BA+0x11C	R/W	P4 Interrupt Enable Control	0x0000_0000
P4_INTSRC	GP_BA+0x120	R/W	P4 Interrupt Source Flag	0x0000_0000
P5_MODE	GP_BA+0x140	R/W	P5 I/O Mode Control	0x0000_XXXX
P5_DINOFF	GP_BA+0x144	R/W	P5 Digital Input Path Disable Control	0x0000_0000
P5_DOUT	GP_BA+0x148	R/W	P5 Data Output Value	0x0000_003F
P5_DATMSK	GP_BA+0x14C	R/W	P5 Data Output Write Mask	0x0000_0000
P5_PIN	GP_BA+0x150	R	P5 Pin Value	0x0000_00XX
P5_DBEN	GP_BA+0x154	R/W	P5 De-bounce Enable Control	0x0000_0000
P5_INTTYPE	GP_BA+0x158	R/W	P5 Interrupt Mode Control	0x0000_0000
P5_INTEN	GP_BA+0x15C	R/W	P5 Interrupt Enable Control	0x0000_0000
P5_INTSRC	GP_BA+0x160	R/W	P5 Interrupt Source Flag	0x0000_0000
GPIO_DBCTL	GP_BA+0x180	R/W	De-bounce Cycle Control	0x0000_0020
P00_PDIO	GP_BA+0x200	R/W	GPIO P0.0 Pin Data Input/Output	0x0000_0001
P01_PDIO	GP_BA+0x204	R/W	GPIO P0.1 Pin Data Input/Output	0x0000_0001
P04_PDIO	GP_BA+0x210	R/W	GPIO P0.4 Pin Data Input/Output	0x0000_0001

P05_PDIO	GP_BA+0x214	R/W	GPIO P0.5 Pin Data Input/Output	0x0000_0001
P06_PDIO	GP_BA+0x218	R/W	GPIO P0.6 Pin Data Input/Output	0x0000_0001
P07_PDIO	GP_BA+0x21C	R/W	GPIO P0.7 Pin Data Input/Output	0x0000_0001
P10_PDIO	GP_BA+0x220	R/W	GPIO P1.0 Pin Data Input/Output	0x0000_0001
P12_PDIO	GP_BA+0x228	R/W	GPIO P1.2 Pin Data Input/Output	0x0000_0001
P13_PDIO	GP_BA+0x22C	R/W	GPIO P1.3 Pin Data Input/Output	0x0000_0001
P14_PDIO	GP_BA+0x230	R/W	GPIO P1.4 Pin Data Input/Output	0x0000_0001
P15_PDIO	GP_BA+0x234	R/W	GPIO P1.5 Pin Data Input/Output	0x0000_0001
P22_PDIO	GP_BA+0x248	R/W	GPIO P2.2 Pin Data Input/Output	0x0000_0001
P23_PDIO	GP_BA+0x24C	R/W	GPIO P2.3 Pin Data Input/Output	0x0000_0001
P24_PDIO	GP_BA+0x250	R/W	GPIO P2.4 Pin Data Input/Output	0x0000_0001
P25_PDIO	GP_BA+0x254	R/W	GPIO P2.5 Pin Data Input/Output	0x0000_0001
P26_PDIO	GP_BA+0x258	R/W	GPIO P2.6 Pin Data Input/Output	0x0000_0001
P30_PDIO	GP_BA+0x260	R/W	GPIO P3.0 Pin Data Input/Output	0x0000_0001
P31_PDIO	GP_BA+0x264	R/W	GPIO P3.1 Pin Data Input/Output	0x0000_0001
P32_PDIO	GP_BA+0x268	R/W	GPIO P3.2 Pin Data Input/Output	0x0000_0001
P34_PDIO	GP_BA+0x270	R/W	GPIO P3.4 Pin Data Input/Output	0x0000_0001
P35_PDIO	GP_BA+0x274	R/W	GPIO P3.5 Pin Data Input/Output	0x0000_0001
P36_PDIO	GP_BA+0x278	R/W	GPIO P3.6 Pin Data Input/Output	0x0000_0001
P46_PDIO	GP_BA+0x298	R/W	GPIO P4.6 Pin Data Input/Output	0x0000_0001
P47_PDIO	GP_BA+0x29C	R/W	GPIO P4.7 Pin Data Input/Output	0x0000_0001
P50_PDIO	GP_BA+0x2A0	R/W	GPIO P5.0 Pin Data Input/Output	0x0000_0001
P51_PDIO	GP_BA+0x2A4	R/W	GPIO P5.1 Pin Data Input/Output	0x0000_0001
P52_PDIO	GP_BA+0x2A8	R/W	GPIO P5.2 Pin Data Input/Output	0x0000_0001
P53_PDIO	GP_BA+0x2AC	R/W	GPIO P5.3 Pin Data Input/Output	0x0000_0001
P54_PDIO	GP_BA+0x2B0	R/W	GPIO P5.4 Pin Data Input/Output	0x0000_0001
P55_PDIO	GP_BA+0x2B4	R/W	GPIO P5.5 Pin Data Input/Output	0x0000_0001

Note: The un-bonding out pin P5.5 must be set to output mode via software when using QFN-33 package IC for minimize the power-down consumption. User does not need to configure the un-bond pins in TSSOP20 to save power consumption.

6.5.8 Register Description

Port 0-5 I/O Mode Control (Px_MODE)

Register	Offset	R/W	Description	Reset Value
P0_MODE	GP_BA+0x000	R/W	P0 I/O Mode Control	0x0000_XXXX
P1_MODE	GP_BA+0x040	R/W	P1 I/O Mode Control	0x0000_XXXX
P2_MODE	GP_BA+0x080	R/W	P2 I/O Mode Control	0x0000_XXXX
P3_MODE	GP_BA+0x0C0	R/W	P3 I/O Mode Control	0x0000_XXXX
P4_MODE	GP_BA+0x100	R/W	P4 I/O Mode Control	0x0000_XXXX
P5_MODE	GP_BA+0x140	R/W	P5 I/O Mode Control	0x0000_XXXX

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

Bits	Description	
[31:16]	**Reserved**	Reserved.
[2n+1:2n] n=0,1..7	**MODEn**	**Port 0-5 I/O Pin[N] Mode Control** Determine each I/O mode of Px.n pins. 00 = Px.n is in Input mode. 01 = Px.n is in Push-pull Output mode. 10 = Px.n is in Open-drain Output mode. 11 = Px.n is in Quasi-bidirectional mode. **Note1:** The initial value of this field is defined by CIOINI (Config0[10]). If CIOINI is set to 0, the default value is 0x0000_FFFF and all pins will be quasi-bidirectional mode after chip powered on. If CIOINI is set to 1, the default value is 0x0000_0000 and all pins will be input mode after chip powered on. **Note2:** Max. n=7 for port 0, n=2, n=3 are reserved. Max. n=7 for port 1, n=1, n=6, n=7 are reserved. Max. n=7 for port 2, n=0, n=1, n=7 are reserved. Max. n=7 for port 3, n=3, n=7 are reserved. Max. n=7 for port 4, n=0,1..5 are reserved. Max. n=7 for port 5, n=6, n=7 are reserved.

Port 0-5 Digital Input Path Disable Control (Px_DINOFF)

Register	Offset	R/W	Description	Reset Value
P0_DINOFF	GP_BA+0x004	R/W	P0 Digital Input Path Disable Control	0x0000_0000
P1_DINOFF	GP_BA+0x044	R/W	P1 Digital Input Path Disable Control	0x0000_0000
P2_DINOFF	GP_BA+0x084	R/W	P2 Digital Input Path Disable Control	0x0000_0000
P3_DINOFF	GP_BA+0x0C4	R/W	P3 Digital Input Path Disable Control	0x0000_0000
P4_DINOFF	GP_BA+0x104	R/W	P4 Digital Input Path Disable Control	0x0000_0000
P5_DINOFF	GP_BA+0x144	R/W	P5 Digital Input Path Disable Control	0x0000_0000

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

Bits	Description	
[31:24]	**Reserved**	Reserved.
[n+16] n=0,1..7	**DINOFF[n]**	**Port 0-5 Pin[N] Digital Input Path Disable Control** Each of these bits is used to control if the digital input path of corresponding Px.n pin is disabled. If input is analog signal, users can disable Px.n digital input path to avoid input current leakage. 0 = Px.n digital input path Enabled. 1 = Px.n digital input path Disabled (digital input tied to low). **Note:** Max. n=7 for port 0, n=2, n=3 are reserved. Max. n=7 for port 1, n=1, n=6, n=7 are reserved. Max. n=7 for port 2, n=0, n=1, n=7 are reserved. Max. n=7 for port 3, n=3, n=7 are reserved. Max. n=7 for port 4, n=0,1..5 are reserved. Max. n=7 for port 5, n=6, n=7 are reserved.
[15:0]	**Reserved**	Reserved.

Port 0-5 Data Output Value (Px_DOUT)

Register	Offset	R/W	Description	Reset Value
P0_DOUT	GP_BA+0x008	R/W	P0 Data Output Value	0x0000_00F3
P1_DOUT	GP_BA+0x048	R/W	P1 Data Output Value	0x0000_003D
P2_DOUT	GP_BA+0x088	R/W	P2 Data Output Value	0x0000_007C
P3_DOUT	GP_BA+0x0C8	R/W	P3 Data Output Value	0x0000_0077
P4_DOUT	GP_BA+0x108	R/W	P4 Data Output Value	0x0000_00C0
P5_DOUT	GP_BA+0x148	R/W	P5 Data Output Value	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**
DOUT							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[n] n=0,1..7	**DOUT[n]**	**Port 0-5 Pin[N] Output Value** Each of these bits controls the status of a Px.n pin when the Px.n is configured as Push-pull output, Open-drain output or Quasi-bidirectional mode. 0 = Px.n will drive Low if the Px.n pin is configured as Push-pull output, Open-drain output or Quasi-bidirectional mode. 1 = Px.n will drive High if the Px.n pin is configured as Push-pull output or Quasi-bidirectional mode. **Note:** Max. n=7 for port 0, n=2, n=3 are reserved. Max. n=7 for port 1, n=1, n=6, n=7 are reserved. Max. n=7 for port 2, n=0, n=1, n=7 are reserved. Max. n=7 for port 3, n=3, n=7 are reserved. Max. n=7 for port 4, n=0,1..5 are reserved. Max. n=7 for port 5, n=6, n=7 are reserved.

Port 0-5 Data Output Write Mask (Px_DATMSK)

Register	Offset	R/W	Description	Reset Value
P0_DATMSK	GP_BA+0x00C	R/W	P0 Data Output Write Mask	0x0000_0000
P1_DATMSK	GP_BA+0x04C	R/W	P1 Data Output Write Mask	0x0000_0000
P2_DATMSK	GP_BA+0x08C	R/W	P2 Data Output Write Mask	0x0000_0000
P3_DATMSK	GP_BA+0x0CC	R/W	P3 Data Output Write Mask	0x0000_0000
P4_DATMSK	GP_BA+0x10C	R/W	P4 Data Output Write Mask	0x0000_0000
P5_DATMSK	GP_BA+0x14C	R/W	P5 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**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DATMSK							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[n] n=0,1..7	**DATMSK[n]**	**Port 0-5 Pin[N] Data Output Write Mask** These bits are used to protect the corresponding DOUT (Px_DOUT[n]) bit. When the DATMSK (Px_DATMSK[n]) bit is set to 1, the corresponding DOUT (Px_DOUT[n]) bit is protected. If the write signal is masked, writing data to the protect bit is ignored. 0 = Corresponding DOUT (Px_DOUT[n]) bit can be updated. 1 = Corresponding DOUT (Px_DOUT[n]) bit protected. **Note1:** This function only protects the corresponding DOUT (Px_DOUT[n]) bit, and will not protect the corresponding PDIO (Pxn_PDIO[0]) bit. **Note2:** Max. n=7 for port 0, n=2, n=3 are reserved. Max. n=7 for port 1, n=1, n=6, n=7 are reserved. Max. n=7 for port 2, n=0, n=1, n=7 are reserved. Max. n=7 for port 3, n=3, n=7 are reserved. Max. n=7 for port 4, n=0,1..5 are reserved. Max. n=7 for port 5, n=6, n=7 are reserved.

Port 0-5 Pin Value (Px_PIN)

Register	Offset	R/W	Description	Reset Value
P0_PIN	GP_BA+0x010	R	P0 Pin Value	0x0000_00XX
P1_PIN	GP_BA+0x050	R	P1 Pin Value	0x0000_00XX
P2_PIN	GP_BA+0x090	R	P2 Pin Value	0x0000_00XX
P3_PIN	GP_BA+0x0D0	R	P3 Pin Value	0x0000_00XX
P4_PIN	GP_BA+0x110	R	P4 Pin Value	0x0000_00XX
P5_PIN	GP_BA+0x150	R	P5 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**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PIN							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[n] n=0,1..7	**PIN[n]**	**Port 0-5 Pin[N] Pin Value** 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. **Note:** Max. n=7 for port 0, n=2, n=3 are reserved. Max. n=7 for port 1, n=1, n=6, n=7 are reserved. Max. n=7 for port 2, n=0, n=1, n=7 are reserved. Max. n=7 for port 3, n=3, n=7 are reserved. Max. n=7 for port 4, n=0,1..5 are reserved. Max. n=7 for port 5, n=6, n=7 are reserved.

Port 0-5 De-bounce Enable Control (Px_DBEN)

Register	Offset	R/W	Description	Reset Value
P0_DBEN	GP_BA+0x014	R/W	P0 De-bounce Enable Control	0x0000_0000
P1_DBEN	GP_BA+0x054	R/W	P1 De-bounce Enable Control	0x0000_0000
P2_DBEN	GP_BA+0x094	R/W	P2 De-bounce Enable Control	0x0000_0000
P3_DBEN	GP_BA+0x0D4	R/W	P3 De-bounce Enable Control	0x0000_0000
P4_DBEN	GP_BA+0x114	R/W	P4 De-bounce Enable Control	0x0000_0000
P5_DBEN	GP_BA+0x154	R/W	P5 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**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DBEN							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[n] n=0,1..7	**DBEN[n]**	**Port 0-5 Pin[N] Input Signal De-bounce Enable Bit** The DBEN[n] bit is used to enable the de-bounce function for each corresponding bit. If the input signal pulse width cannot be sampled by continuous two de-bounce sample cycle, the input signal transition is seen as the signal bounce and will not trigger the interrupt. The de-bounce clock source is controlled by DBCLKSRC (GPIO_DBCTL [4]), one de-bounce sample cycle period is controlled by DBCLKSEL (GPIO_DBCTL [3:0]). 0 = Px.n de-bounce function Disabled. 1 = Px.n de-bounce function Enabled. The de-bounce function is valid only for edge triggered interrupt. If the interrupt mode is level triggered, the de-bounce enable bit is ignored. **Note1:** If Px.n pin is chosen as Power-down wake-up source, user should be disable the de-bounce function before entering Power-down mode to avoid the second interrupt event occurred after system waken up which caused by Px.n de-bounce function. **Note2:** Max. n=7 for port 0, n=2, n=3 are reserved. Max. n=7 for port 1, n=1, n=6, n=7 are reserved. Max. n=7 for port 2, n=0, n=1, n=7 are reserved. Max. n=7 for port 3, n=3, n=7 are reserved. Max. n=7 for port 4, n=0,1..5 are reserved. Max. n=7 for port 5, n=6, n=7 are reserved.

Port 0-5 Interrupt Mode Control (Px_ INTTYPE)

Register	Offset	R/W	Description	Reset Value
P0_INTTYPE	GP_BA+0x018	R/W	P0 Interrupt Mode Control	0x0000_0000
P1_INTTYPE	GP_BA+0x058	R/W	P1 Interrupt Mode Control	0x0000_0000
P2_INTTYPE	GP_BA+0x098	R/W	P2 Interrupt Mode Control	0x0000_0000
P3_INTTYPE	GP_BA+0x0D8	R/W	P3 Interrupt Mode Control	0x0000_0000
P4_INTTYPE	GP_BA+0x118	R/W	P4 Interrupt Mode Control	0x0000_0000
P5_INTTYPE	GP_BA+0x158	R/W	P5 Interrupt Mode 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							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TYPE							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[n] n=0,1..7	**TYPE[n]**	**Port 0-5 Pin[N] Edge Or Level Detection Interrupt Trigger Type Control** TYPE (Px_INTTYPE[n]) bit is used to control the triggered interrupt is by level trigger or by edge trigger. If the interrupt is by edge trigger, the trigger source can be controlled by de-bounce. If the interrupt is by level trigger, the input source is sampled by one HCLK clock and generates the interrupt. 0 = Edge trigger interrupt. 1 = Level trigger interrupt. If the pin is set as the level trigger interrupt, only one level can be set on the registers RHIEN (Px_INTEN[n+16])/FLIEN (Px_INTEN[n]). If both levels to trigger interrupt are set, the setting is ignored and no interrupt will occur. The de-bounce function is valid only for edge triggered interrupt. If the interrupt mode is level triggered, the de-bounce enable bit is ignored. **Note:** Max. n=7 for port 0, n=2, n=3 are reserved. Max. n=7 for port 1, n=1, n=6, n=7 are reserved. Max. n=7 for port 2, n=0, n=1, n=7 are reserved. Max. n=7 for port 3, n=3, n=7 are reserved. Max. n=7 for port 4, n=0,1..5 are reserved. Max. n=7 for port 5, n=6, n=7 are reserved.

Port 0-5 Interrupt Enable Control (Px_INTEN)

Register	Offset	R/W	Description	Reset Value
P0_INTEN	GP_BA+0x01C	R/W	P0 Interrupt Enable Control	0x0000_0000
P1_INTEN	GP_BA+0x05C	R/W	P1 Interrupt Enable Control	0x0000_0000
P2_INTEN	GP_BA+0x09C	R/W	P2 Interrupt Enable Control	0x0000_0000
P3_INTEN	GP_BA+0x0DC	R/W	P3 Interrupt Enable Control	0x0000_0000
P4_INTEN	GP_BA+0x11C	R/W	P4 Interrupt Enable Control	0x0000_0000
P5_INTEN	GP_BA+0x15C	R/W	P5 Interrupt Enable Control	0x0000_0000

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

Bits	Description	
[31:24]	**Reserved**	Reserved.
[n+16] n=0,1..7	**RHIEN[n]**	**Port 0-5 Pin[N] Rising Edge Or High Level Interrupt Trigger Type Enable Bit** The RHIEN (Px_INTEN[n+16]) bit is used to enable the interrupt for each of the corresponding input Px.n pin. Set bit to 1 also enable the pin wake-up function. When setting the RHIEN (Px_INTEN[n+16]) bit to 1: If the interrupt is level trigger (TYPE (Px_INTTYPE[n]) bit is set to 1), the input Px.n pin will generate the interrupt while this pin state is at high level. If the interrupt is edge trigger (TYPE (Px_INTTYPE[n]) bit is set to 0), the input Px.n pin will generate the interrupt while this pin state changed from low to high. 0 = Px.n level high or low to high interrupt Disabled. 1 = Px.n level high or low to high interrupt Enabled. **Note:** Max. n=7 for port 0, n=2, n=3 are reserved. Max. n=7 for port 1, n=1, n=6, n=7 are reserved. Max. n=7 for port 2, n=0, n=1, n=7 are reserved. Max. n=7 for port 3, n=3, n=7 are reserved. Max. n=7 for port 4, n=0,1..5 are reserved. Max. n=7 for port 5, n=6, n=7 are reserved.
[15:8]	**Reserved**	Reserved.

[n] n=0,1..7	**FLIEN[n]**	**Port 0-5 Pin[N] Falling Edge Or Low Level Interrupt Trigger Type Enable Bit** The FLIEN (Px_INTEN[n]) bit is used to enable the interrupt for each of the corresponding input Px.n pin. Set bit to 1 also enable the pin wake-up function. When setting the FLIEN (Px_INTEN[n]) bit to 1: If the interrupt is level trigger (TYPE (Px_INTTYPE[n]) bit is set to 1), the input Px.n pin will generate the interrupt while this pin state is at low level. If the interrupt is edge trigger (TYPE (Px_INTTYPE[n]) bit is set to 0), the input Px.n pin will generate the interrupt while this pin state changed from high to low. 0 = Px.n level low or high to low interrupt Disabled. 1 = Px.n level low or high to low interrupt Enabled. **Note:** Max. n=7 for port 0, n=2, n=3 are reserved. Max. n=7 for port 1, n=1, n=6, n=7 are reserved. Max. n=7 for port 2, n=0, n=1, n=7 are reserved. Max. n=7 for port 3, n=3, n=7 are reserved. Max. n=7 for port 4, n=0,1..5 are reserved. Max. n=7 for port 5, n=6, n=7 are reserved.

Port 0-5 Interrupt Source Flag (Px_INTSRC)

Register	Offset	R/W	Description	Reset Value
P0_INTSRC	GP_BA+0x020	R/W	P0 Interrupt Source Flag	0x0000_0000
P1_INTSRC	GP_BA+0x060	R/W	P1 Interrupt Source Flag	0x0000_0000
P2_INTSRC	GP_BA+0x0A0	R/W	P2 Interrupt Source Flag	0x0000_0000
P3_INTSRC	GP_BA+0x0E0	R/W	P3 Interrupt Source Flag	0x0000_0000
P4_INTSRC	GP_BA+0x120	R/W	P4 Interrupt Source Flag	0x0000_0000
P5_INTSRC	GP_BA+0x160	R/W	P5 Interrupt Source Flag	0x0000_0000

31	30	29	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**
INTSRC							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[n] n=0,1..15	**INTSRC[n]**	**Port 0-5 Pin[N] Interrupt Source Flag** Write Operation: 0 = No action. 1 = Clear the corresponding pending interrupt. Read Operation: 0 = No interrupt at Px.n. 1 = Px.n generates an interrupt. **Note:** Max. n=7 for port 0, n=2, n=3 are reserved. Max. n=7 for port 1, n=1, n=6, n=7 are reserved. Max. n=7 for port 2, n=0, n=1, n=7 are reserved. Max. n=7 for port 3, n=3, n=7 are reserved. Max. n=7 for port 4, n=0,1..5 are reserved. Max. n=7 for port 5, n=6, n=7 are reserved.

Interrupt De-bounce Cycle Control (GPIO_DBCTL)

Register	Offset	R/W	Description	Reset Value
GPIO_DBCTL	GP_BA+0x180	R/W	De-bounce Cycle 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**	**Interrupt Clock On Mode** 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. 1 = All I/O pins edge detection circuit is always active after reset. **Note:** It is recommended to disable this bit to save system power if no special application concern.
[4]	**DBCLKSRC**	**De-bounce Counter Clock Source Selection** 0 = De-bounce counter clock source is HCLK. 1 = De-bounce counter clock source is 10 kHz internal low speed RC oscillator (LIRC).
[3:0]	**DBCLKSEL**	**De-bounce Sampling Cycle Selection** 0000 = Sample interrupt input once per 1 clock. 0001 = Sample interrupt input once per 2 clocks. 0010 = Sample interrupt input once per 4 clocks. 0011 = Sample interrupt input once per 8 clocks. 0100 = Sample interrupt input once per 16 clocks. 0101 = Sample interrupt input once per 32 clocks. 0110 = Sample interrupt input once per 64 clocks. 0111 = Sample interrupt input once per 128 clocks. 1000 = Sample interrupt input once per 256 clocks. 1001 = Sample interrupt input once per 2*256 clocks. 1010 = Sample interrupt input once per 4*256 clocks. 1011 = Sample interrupt input once per 8*256 clocks. 1100 = Sample interrupt input once per 16*256 clocks. 1101 = Sample interrupt input once per 32*256 clocks. 1110 = Sample interrupt input once per 64*256 clocks. 1111 = Sample interrupt input once per 128*256 clocks.

GPIO Px.n Data Input/Output (Pxn_PDIO)

Register	Offset	R/W	Description	Reset Value
P00_PDIO	GP_BA+0x200	R/W	GPIO P0.0 Pin Data Input/Output	0x0000_0001
P01_PDIO	GP_BA+0x204	R/W	GPIO P0.1 Pin Data Input/Output	0x0000_0001
P04_PDIO	GP_BA+0x210	R/W	GPIO P0.4 Pin Data Input/Output	0x0000_0001
P05_PDIO	GP_BA+0x214	R/W	GPIO P0.5 Pin Data Input/Output	0x0000_0001
P06_PDIO	GP_BA+0x218	R/W	GPIO P0.6 Pin Data Input/Output	0x0000_0001
P07_PDIO	GP_BA+0x21C	R/W	GPIO P0.7 Pin Data Input/Output	0x0000_0001
P10_PDIO	GP_BA+0x220	R/W	GPIO P1.0 Pin Data Input/Output	0x0000_0001
P12_PDIO	GP_BA+0x228	R/W	GPIO P1.2 Pin Data Input/Output	0x0000_0001
P13_PDIO	GP_BA+0x22C	R/W	GPIO P1.3 Pin Data Input/Output	0x0000_0001
P14_PDIO	GP_BA+0x230	R/W	GPIO P1.4 Pin Data Input/Output	0x0000_0001
P15_PDIO	GP_BA+0x234	R/W	GPIO P1.5 Pin Data Input/Output	0x0000_0001
P22_PDIO	GP_BA+0x248	R/W	GPIO P2.2 Pin Data Input/Output	0x0000_0001
P23_PDIO	GP_BA+0x24C	R/W	GPIO P2.3 Pin Data Input/Output	0x0000_0001
P24_PDIO	GP_BA+0x250	R/W	GPIO P2.4 Pin Data Input/Output	0x0000_0001
P25_PDIO	GP_BA+0x254	R/W	GPIO P2.5 Pin Data Input/Output	0x0000_0001
P26_PDIO	GP_BA+0x258	R/W	GPIO P2.6 Pin Data Input/Output	0x0000_0001
P30_PDIO	GP_BA+0x260	R/W	GPIO P3.0 Pin Data Input/Output	0x0000_0001
P31_PDIO	GP_BA+0x264	R/W	GPIO P3.1 Pin Data Input/Output	0x0000_0001
P32_PDIO	GP_BA+0x268	R/W	GPIO P3.2 Pin Data Input/Output	0x0000_0001
P34_PDIO	GP_BA+0x270	R/W	GPIO P3.4 Pin Data Input/Output	0x0000_0001
P35_PDIO	GP_BA+0x274	R/W	GPIO P3.5 Pin Data Input/Output	0x0000_0001
P36_PDIO	GP_BA+0x278	R/W	GPIO P3.6 Pin Data Input/Output	0x0000_0001
P46_PDIO	GP_BA+0x298	R/W	GPIO P4.6 Pin Data Input/Output	0x0000_0001
P47_PDIO	GP_BA+0x29C	R/W	GPIO P4.7 Pin Data Input/Output	0x0000_0001
P50_PDIO	GP_BA+0x2A0	R/W	GPIO P5.0 Pin Data Input/Output	0x0000_0001
P51_PDIO	GP_BA+0x2A4	R/W	GPIO P5.1 Pin Data Input/Output	0x0000_0001
P52_PDIO	GP_BA+0x2A8	R/W	GPIO P5.2 Pin Data Input/Output	0x0000_0001
P53_PDIO	GP_BA+0x2AC	R/W	GPIO P5.3 Pin Data Input/Output	0x0000_0001
P54_PDIO	GP_BA+0x2B0	R/W	GPIO P5.4 Pin Data Input/Output	0x0000_0001

P55_PDIO	GP_BA+0x2B4	R/W	GPIO P5.5 Pin Data Input/Output	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							**PDIO**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**PDIO**	**GPIO Px.N Pin Data Input/Output** Writing this bit can control one GPIO pin output value. 0 = Corresponding GPIO pin set to low. 1 = Corresponding GPIO pin set to high. Read this register to get GPIO pin status. For example, writing P00_PDIO will reflect the written value to bit DOUT (P0_DOUT[0]), reading P00_PDIO will return the value of PIN (P0_PIN[0]). **Note1:** The writing operation will not be affected by register DATMSK (Px_DATMSK[n]). **Note2:** Max. n=7 for port 0, n=2, n=3 are reserved. Max. n=7 for port 1, n=1, n=6, n=7 are reserved. Max. n=7 for port 2, n=0, n=1, n=7 are reserved. Max. n=7 for port 3, n=3, n=7 are reserved. Max. n=7 for port 4, n=0,1..5 are reserved. Max. n=7 for port 5, n=6, n=7 are reserved.

6.6 Timer Controller (TMR)

6.6.1 Overview

The Timer Controller includes two 32-bit timers, TMR0 and TMR1, allowing user to easily implement a timer control for applications. The timer can perform functions, such as frequency measurement, delay timing, clock generation, event counting by external input pins, and interval measurement by external capture pins.

6.6.2 Features

- Two sets of 32-bit timer with 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 (TIMRTx_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
- Supports internal capture triggered while internal ACMP output signal transition

6.6.3 Block Diagram

The timer controller block diagram and clock control are shown in Figure 6.6-1 and Figure 6.6-2.

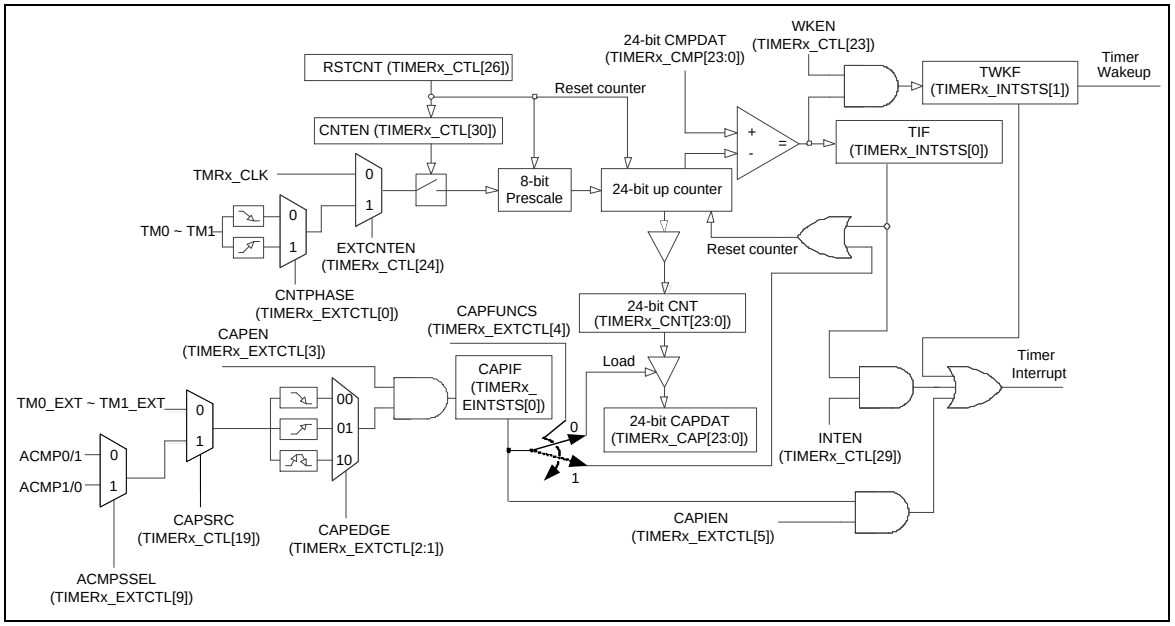


Figure 6.6-1 Timer Controller Block Diagram

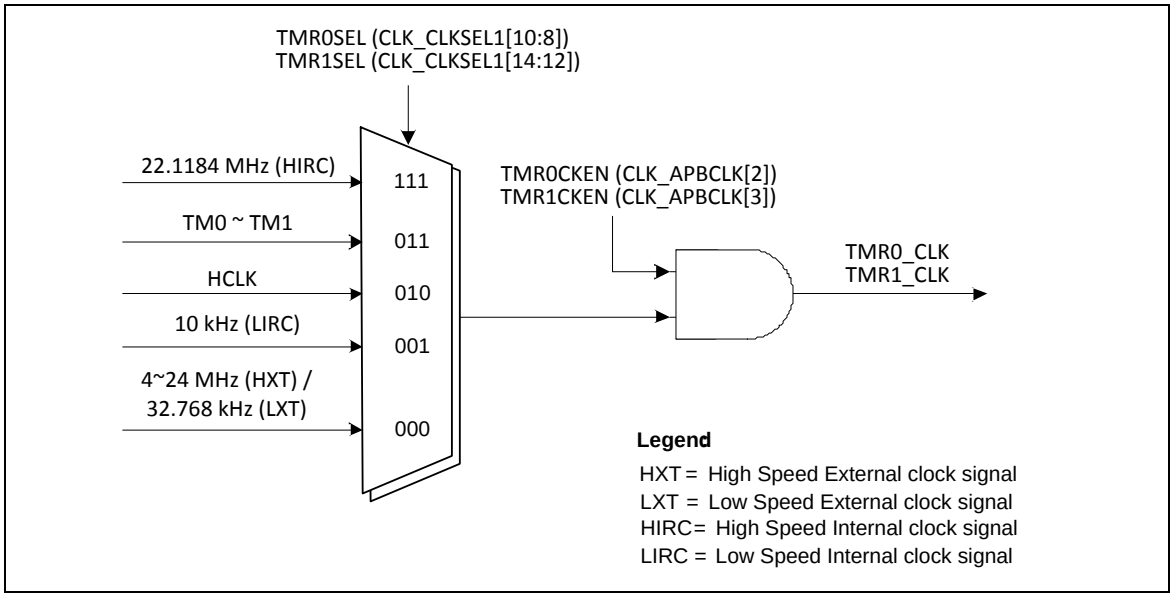


Figure 6.6-2 Clock Source of Timer Controller

6.6.4 Basic Configuration

The peripheral clock source of TMR0 and TMR1 can be enabled in TMRxCKEN (CLK_APBCLK[3:2]) and selected as clock source in TMR0SEL (CLK_CLKSEL1[10:8]) for TMR0, TMR1SEL (CLK_CLKSEL1[14:12]) for TMR1.

6.6.5 Functional Description

6.6.5.1 Timer Interrupt Flag

The timer controller supports two interrupt flags; one is TIF (TIMERx_INTSTS[0]) which is set while timer counter value CNT (TIMERx_CNT[23:0]) matches the timer compared value CMPDAT (TIMERx_CMP[23:0]), and the other is CAPIF (TIMERx_EINTSTS[0]) which is set when the transition on the TMx_EXT pin associated CAPEDGE (TIMERx_EXTCTL[2:1]) setting.

6.6.5.2 Timer Counting Operation Mode

The Timer controller provides four timer counting modes: One-shot, Periodic, Toggle-output and Continuous Counting operation modes as described below.

6.6.5.3 One-shot Mode

If the timer controller is configured at one-shot mode (TIMERx_CTL[28:27] is 2'b00) 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.

6.6.5.4 Periodic Mode

If the timer controller is configured at periodic mode (TIMERx_CTL[28:27] is 2'b01) and CNTEN (TIMERx_CTL[30]) is set, the timer counter starts up counting. Once the CNT (TIMERx_CNT[23:0]) value reaches CMPDAT (TIMERx_CMP[23:0]) value, the TIF (TIMERx_INTSTS[0]) will be set to 1, CNT value will be cleared automatically by timer controller and timer counter operates counting again. In the meantime, if the INTEN (TIMERx_CTL[29]) bit is enabled, the timer interrupt signal is generated and sent to NVIC to inform CPU also. In this mode, the timer controller operates counting and compares with CMPDAT value periodically until the CNTEN bit is cleared by user.

6.6.5.5 Toggle-output Mode

If the timer controller is configured at toggle-output mode (TIMERx_CTL[28:27] is 2'b10) and CNTEN (TIMERx_CTL[30]) is set, the timer counter starts up counting. The counting operation of toggle-output mode is almost the same as periodic mode, except toggle-output mode has associated TM0 ~ TM1 pin to output signal while specify TIF (TIMERx_INTSTS[0]) is set. Thus, the toggle-output signal on TM0 ~ TM1 pin is high and changing back and forth with 50% duty cycle. The output pin could be either TM0 ~ TM1 or TM0_EXT ~ TM1_EXT depending on the TGLPINSEL (TIMERx_CTL[18]).

6.6.5.6 Continuous Counting Mode

If the timer controller is configured at continuous counting mode (TIMERx_CTL[28:27] is 2'b11) and CNTEN (TIMERx_CTL[30]) is set, the timer counter starts up counting. Once the CNT (TIMERx_CNT[23:0]) value reaches the CMPDAT (TIMERx_CMP[23:0]) value, the TIF (TIMERx_INTSTS[0]) will be set to 1 and the 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. 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, the timer counter is kept counting and CNT value will not go 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 the CMPDAT value as 200 and clears TIF, the TIF will be set to 1 again when CNT value reaches 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.

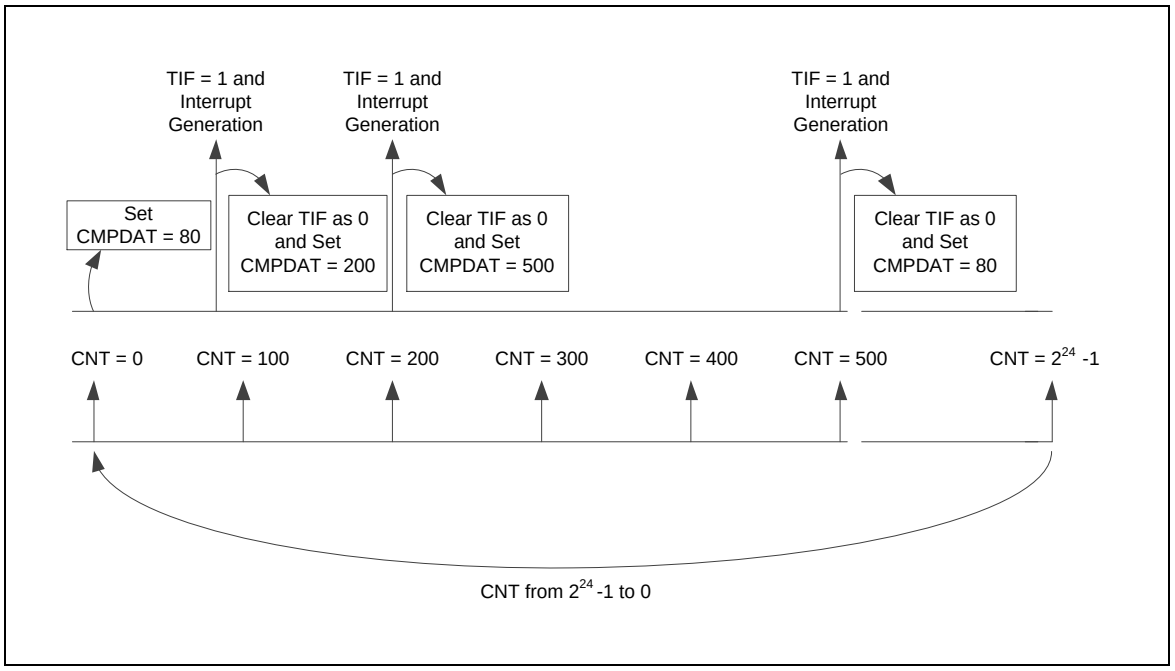


Figure 6.6-3 Continuous Counting Mode

6.6.5.7 Event Counting Mode

The timer controller also provides an application which can count the input event from TMx (x= 0~1) 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 HCLK.

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

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

6.6.5.8 Input Capture Function

The input capture or reset function is provided to capture or reset timer counter value. The capture function with free-counting capture mode and trigger-counting capture mode are configured by CAPSEL (TIMERx_EXTCTL[8]). The free-counting capture mode, reset mode, trigger-counting capture mode are described as follows.

6.6.5.9 *Free-Counting Capture Mode*

The event capture function is used to load CNT (TIMERx_CNT[23:0]) value to CAPDAT (TIMERx_CAP[23:0]) value while edge transition detected on TMx_EXT (x= 0~1) pin. In this mode, CAPSEL (TIMERx_EXTCTL[8]) and CAPFUNCS (TIMERx_EXTCTL[4]) should be as 0 for select TMx_EXT transition is using to trigger event capture function and the timer peripheral clock source should be set as HCLK.

User can enable or disable TMx_EXT pin de-bounce circuit by setting CAPDBEN (TIMERx_EXTCTL[6]). The transition frequency of TMx_EXT pin should be less than 1/3 HCLK if TMx_EXT pin de-bounce disabled or less than 1/8 HCLK if TMx_EXT pin de-bounce enabled to assure the capture function can be work normally, and user can also select edge transition detection of TMx_EXT pin by setting CAPEDGE (TIMERx_EXTCTL[2:1]).

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

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 (TIMERx_EINTSTS[0]) status. The operation method is described in Table 6.6-1.

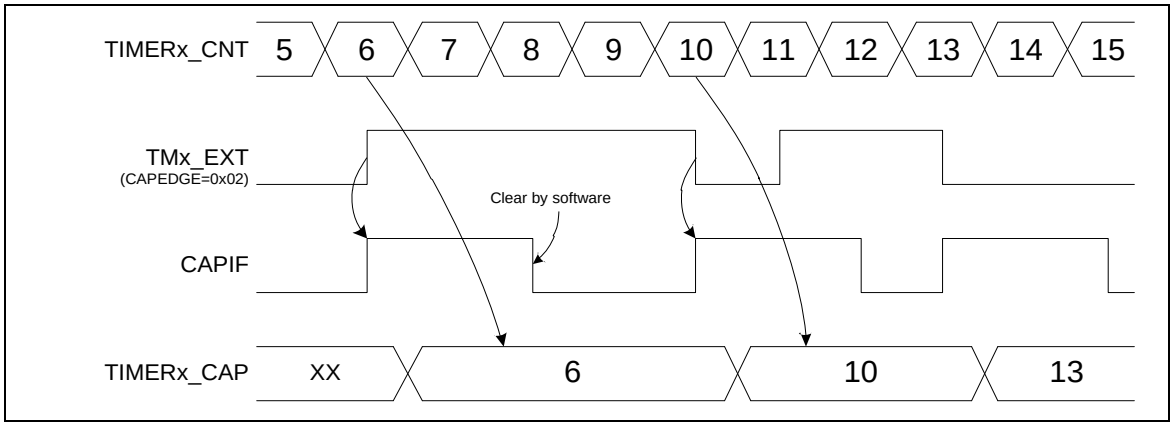


Figure 6.6-4 Free-Counting Capture Mode

6.6.5.10 *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 TMx_EXT (x= 0~1). In this mode, most the settings are the same as event capture mode except CAPFUNCS (TIMERx_EXTCTL[4]) should be as 1 for select TMx_EXT transition is using to trigger reset counter value. The operation method is also described in Table 6.6-1.

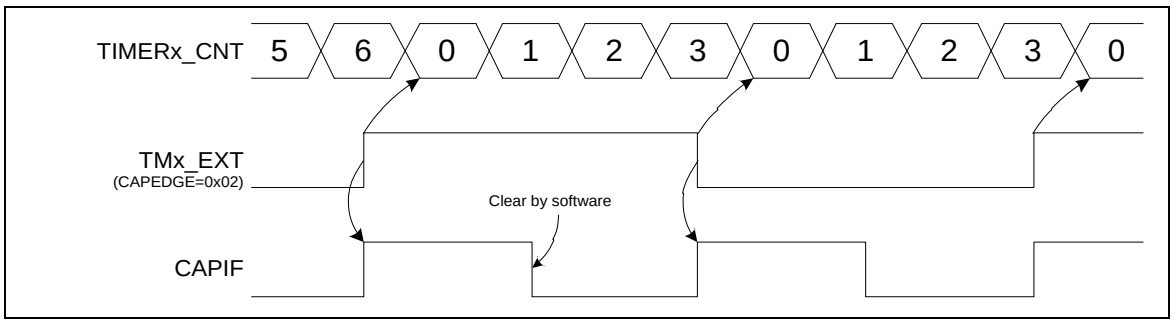


Figure 6.6-5 External Reset Counter Mode

6.6.5.11 Trigger-Counting Capture Mode

If CAPSEL (TIMERx_EXTCTL[8]) is set to 1, CAPEN (TIMERx_EXTCTL[3]) is set to 1 and CAPFUNCS (TIMERx_EXTCTL[4]) is set to 0, the CNT will be reset to 0 then captured into CAPDAT register when TMx_EXT (x= 0~1) pin trigger condition occurred. The TMx_EXT trigger edge can be chosen by CAPEDGE (TIMERx_EXTCTL[2:1]). The detailed operation method is described in Table 6.6-1. When TMx_EXT trigger occurred, CAPIF (TIMERx_EINTSTS[0]) is set to 1, and the interrupt signal is generated, then sent to NVIC to inform CPU if CAPIEN (TIMERx_EXTCTL[5]) is 1. And, the TMx_EXT source operating frequency should be less than 1/3 HCLK frequency if disable TMx_EXT de-bounce or less than 1/8 HCLK frequency if enabling TMx_EXT de-bounce. It also provides TMx_EXT enabled or disabled capture de-bounce function by CAPDBEN (TIMERx_EXTCTL[6]).

Function	CAPSEL (TIMERx_EXTCTL[8])	CAPFUNCS TIMERx_EXTCTL[4])	CAPEDGE TIMERx_EXTCTL[2:1])	Operation Description
Free-counting Capture Mode	0	0	00	A 1 to 0 transition on TMx_EXT (x= 0~1) pin is detected. CNT is captured to CAPDAT.
	0	0	01	A 0 to 1 transition on TMx_EXT (x= 0~1) pin is detected. CNT is captured to CAPDAT.
	0	0	10	Either 1 to 0 or 0 to 1 transition on TMx_EXT (x= 0~1) pin is detected. CNT is captured to CAPDAT.
	0	0	11	Reserved
External Reset Counter Mode	0	1	00	An 1 to 0 transition on TMx_EXT (x= 0~1) pin is detected. CNT is reset to 0.
	0	1	01	A 0 to 1 transition on TMx_EXT (x= 0~1) pin is detected. CNT is reset to 0.
	0	1	10	Either 1 to 0 or 0 to 1 transition on TMx_EXT (x= 0~1) pin is detected. CNT is reset to 0.
	0	1	11	Reserved
Trigger-Counting Capture Mode	1	0	00	Falling Edge Trigger: The 1st 1 to 0 transition on TMx_EXT (x= 0~1) pin is detected to reset CNT as 0 and then starts counting, while the 2nd 1 to 0 transition stops counting.
	1	0	01	Rising Edge Trigger: The 1st 0 to 1 transition to on TMx_EXT (x= 0~1) pin is detected to reset CNT as 0 and then starts counting, while the 2nd 0 to 1 transition stops counting.
	1	0	10	Level Change Trigger: An 1 to 0 transition on TMx_EXT (x= 0~1) pin is detected to reset CNT as 0 and then starts counting, while 0 to 1 transition stops counting.
	1	0	11	Level Change Trigger: A 0 to 1 transition on TMx_EXT (x= 0~1) pin is detected to reset CNT as 0 and then starts

				counting, while 1 to 0 transition stops counting.

Table 6.6-1 Input Capture Mode Operation

6.6.5.12 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 ACMPSSEL (TIMERx_EXTCTL[9]) to decide which ACMP output signal as TMx_EXT (x= 0~1) capture source. The detailed setting of capture function is the same as the previous description in External Capture Mode.

6.6.6 Register Map

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

Register	Offset	R/W	Description	Reset Value
TMR Base Address **TMR_BA = 0x4001_0000**				
TIMER0_CTL	TMR_BA+0x00	R/W	Timer0 Control and Status Register	0x0000_0005
TIMER0_CMP	TMR_BA+0x04	R/W	Timer0 Compare Register	0x0000_0000
TIMER0_INTSTS	TMR_BA+0x08	R/W	Timer0 Interrupt Status Register	0x0000_0000
TIMER0_CNT	TMR_BA+0x0C	R	Timer0 Data Register	0x0000_0000
TIMER0_CAP	TMR_BA+0x10	R	Timer0 Capture Data Register	0x0000_0000
TIMER0_EXTCTL	TMR_BA+0x14	R/W	Timer0 External Control Register	0x0000_0000
TIMER0_EINTSTS	TMR_BA+0x18	R/W	Timer0 External Interrupt Status Register	0x0000_0000
TIMER1_CTL	TMR_BA+0x20	R/W	Timer1 Control and Status Register	0x0000_0005
TIMER1_CMP	TMR_BA+0x24	R/W	Timer1 Compare Register	0x0000_0000
TIMER1_INTSTS	TMR_BA+0x28	R/W	Timer1 Interrupt Status Register	0x0000_0000
TIMER1_CNT	TMR_BA+0x2C	R	Timer1 Data Register	0x0000_0000
TIMER1_CAP	TMR_BA+0x30	R	Timer1 Capture Data Register	0x0000_0000
TIMER1_EXTCTL	TMR_BA+0x34	R/W	Timer1 External Control Register	0x0000_0000
TIMER1_EINTSTS	TMR_BA+0x38	R/W	Timer1 External Interrupt Status Register	0x0000_0000

6.6.7 Register Description

Timer Control Register (TIMERx_CTL)

Register	Offset	R/W	Description	Reset Value
TIMER0_CTL	TMR_BA+0x00	R/W	Timer0 Control and Status Register	0x0000_0005
TIMER1_CTL	TMR_BA+0x20	R/W	Timer1 Control and Status Register	0x0000_0005

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

Bits	Description	
[31]	**ICEDEBUG**	**ICE Debug Mode Acknowledge Disable Bit (Write Protect)** 0 = ICE debug mode acknowledgement effects TIMER counting. TIMER counter will be held while CPU is held by ICE. 1 = ICE debug mode acknowledgement Disabled. TIMER counter will keep going no matter CPU is held by ICE or not. **Note:** This bit is write protected. Refer to the SYS_REGLCTL register.
[30]	**CNTEN**	**Timer Counting Enable Bit** 0 = Stops/Suspends counting. 1 = Starts counting. **Note1:** In stop status, and then setting CNTEN to 1 will enable the 24-bit up counter to keep counting from the last stop counting value. **Note2:** This bit is auto-cleared by hardware in one-shot mode (TIMERx_CTL[28:27] = 2'b00) when the timer interrupt flag TIF (TIMERx_INTSTS[0]) is generated.
[29]	**INTEN**	**Timer Interrupt Enable Bit** 0 = Timer Interrupt Disabled. 1 = Timer Interrupt Enabled. **Note:** If this bit is enabled, when the timer interrupt flag TIF is set to 1, the timer interrupt signal will be generated and inform CPU.
[28:27]	**OPMODE**	**Timer Counting Mode Selection** 00 = The Timer controller is operated in one-shot mode. 01 = The Timer controller is operated in periodic mode. 10 = The Timer controller is operated in toggle-output mode. 11 = The Timer controller is operated in continuous counting mode.
[26]	**RSTCNT**	**Timer Counter Reset**

Bits	Description	
		Setting this bit will reset the 24-bit up counter value CNT (TIMERx_CNT[23:0]) and also force CNTEN (TIMERx_CTL[30]) to 0 if ACTSTS (TIMERx_CTL[25]) is 1. 0 = No effect. 1 = Reset internal 8-bit prescale counter, 24-bit up counter value and CNTEN bit.
[25]	**ACTSTS**	**Timer Active Status (Read Only)** This bit indicates the 24-bit up counter status. 0 = 24-bit up counter is not active. 1 = 24-bit up counter is active.
[24]	**EXTCNTEN**	**Event Counter Mode Enable Bit** This bit is for external counting pin function enabled. 0 = Event counter mode Disabled. 1 = Event counter mode Enabled. **Note:** When timer is used as an event counter, this bit should be set to 1 and select HCLK as timer clock source.
[23]	**WKEN**	**Wake-up Function Enable Bit** If this bit is set to 1, while the 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.
[22:20]	**Reserved**	Reserved.
[19]	**CAPSRC**	**Capture Pin Source Select Bit** 0 = Capture Function source is from TMx_EXT (x= 0~1) pin. 1 = Capture Function source is from internal ACMP output signal. User can set CAPSRCMP (TIMERx_EXTCTL[9]) to decide which ACMP output signal as timer capture source.
[18]	**TGLPINSEL**	**Toggle-output Pin Select Bit** 0 = Toggle mode output to TMx (Timer Event Counter Pin). 1 = Toggle mode output to TMx_EXT (Timer External Capture Pin).
[17]	**CMPCTL**	**Timer Compared Mode Select Bit** 0 = The behavior selection in one-shot or periodic mode Disabled. When user updates CMPDAT while timer is running in one-shot or periodic mode, CNT will be reset to default value. 1 = The behavior selection in one-shot or periodic mode Enabled. When user updates CMPDAT while timer is running in one-shot or periodic mode, the limitations as bellows list, If updated CMPDAT value > CNT, CMPDAT will be updated and CNT keep running continually. If updated CMPDAT value = CNT, timer time-out interrupt will be asserted immediately. If updated CMPDAT value < CNT, CNT will be reset to default value.
[16:8]	**Reserved**	Reserved.
[7:0]	**PSC**	**Prescale Counter** Timer input clock or event source is divided by (PSC+1) before it is fed to the timer up counter. If this field is 0 (PSC = 0), then there is no scaling.

Timer Compare Register (TIMERx_CMP)

Register	Offset	R/W	Description	Reset Value
TIMER0_CMP	TMR_BA+0x04	R/W	Timer0 Compare Register	0x0000_0000
TIMER1_CMP	TMR_BA+0x24	R/W	Timer1 Compare 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**	**Timer Compared Value** CMPDAT is a 24-bit compared value register. When the internal 24-bit up counter value is equal to CMPDAT value, the TIF (TIMERx_INTSTS[0] Timer Interrupt Flag) will set to 1. Time-out period = (Period of timer clock input) * (8-bit PSC + 1) * (24-bit CMPDAT). **Note1:** Never write 0x0 or 0x1 in CMPDAT field, or the core will run into unknown state. **Note2:** 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 the CMPDAT field.

Timer Interrupt Status Register (TIMERx_INTSTS)

Register	Offset	R/W	Description	Reset Value
TIMER0_INTSTS	TMR_BA+0x08	R/W	Timer0 Interrupt Status Register	0x0000_0000
TIMER1_INTSTS	TMR_BA+0x28	R/W	Timer1 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**	**Timer Wake-up Flag** This bit indicates the interrupt wake-up flag status of timer. 0 = Timer does not cause CPU wake-up. 1 = CPU wake-up from Idle or Power-down mode if timer time-out interrupt signal generated. **Note:** This bit is cleared by writing 1 to it.
[0]	**TIF**	**Timer Interrupt Flag** This bit indicates the interrupt flag status of Timer while 24-bit timer up counter CNT (TIMERx_CNT[23:0]) value reaches to CMPDAT (TIMERx_CMP[23:0]) value. 0 = No effect. 1 = CNT value matches the CMPDAT value. **Note:** This bit is cleared by writing 1 to it.

Timer Data Register (TIMERx_CNT)

Register	Offset	R/W	Description	Reset Value
TIMER0_CNT	TMR_BA+0x0C	R	Timer0 Data Register	0x0000_0000
TIMER1_CNT	TMR_BA+0x2C	R	Timer1 Data Register	0x0000_0000

31	30	29	28	27	26	25	24
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	
[31:24]	**Reserved**	Reserved.
[23:0]	**CNT**	**Timer Data Register** This field can be reflected the internal 24-bit timer counter value or external event input counter value from TMx (x=0~1) pin. If EXTCNTEN (TIMERx_CTL[24]) is 0, user can read CNT value for getting current 24- bit counter value. If EXTCNTEN (TIMERx_CTL[24]) is 1, user can read CNT value for getting current 24- bit event input counter value.

Timer Capture Data Register (TIMERx_CAP)

Register	Offset	R/W	Description	Reset Value
TIMER0_CAP	TMR_BA+0x10	R	Timer0 Capture Data Register	0x0000_0000
TIMER1_CAP	TMR_BA+0x30	R	Timer1 Capture Data Register	0x0000_0000

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

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:0]	**CAPDAT**	**Timer Capture Data Register** When CAPEN (TIMERx_EXTCTL[3]) bit is set, CAPFUNCS (TIMERx_EXTCTL[4]) bit is 0, and a transition on TMx_EXT pin matched the CAPEDGE (TIMERx_EXTCTL[2:1]) 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.

Timer External Control Register (TIMERx_EXTCTL)

Register	Offset	R/W	Description	Reset Value
TIMER0_EXTCTL	TMR_BA+0x14	R/W	Timer0 External Control Register	0x0000_0000
TIMER1_EXTCTL	TMR_BA+0x34	R/W	Timer1 External 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						ACMPSSEL	CAPSEL
7	6	5	4	3	2	1	0
CNTDBEN	CAPDBEN	CAPIEN	CAPFUNCS	CAPEN	CAPEDGE		CNTPHASE

Bits	Description	
[31:10]	**Reserved**	Reserved.
[9]	**ACMPSSEL**	**ACMP Source Selection to Trigger Capture Function** For Timer 0: 0 = Capture Function source is from ACMP0 output signal for TIMER0. 1 = Capture Function source is from ACMP1 output signal for TIMER0. For Timer 1: 0 = Capture Function source is from ACMP1 output signal for TIMER1. 1 = Capture Function source is from ACMP0 output signal for TIMER1.
[8]	**CAPSEL**	**Capture Mode Select Bit** 0 = Timer counter reset function or free-counting mode of timer capture function. 1 = Trigger-counting mode of timer capture function.
[7]	**CNTDBEN**	**Timer Counter Pin De-bounce Enable Bit** 0 = TMx (x= 0~1) pin de-bounce Disabled. 1 = TMx (x= 0~1) pin de-bounce Enabled. **Note:** If this bit is enabled, the edge detection of TMx pin is detected with de-bounce circuit.
[6]	**CAPDBEN**	**Timer External Capture Pin De-bounce Enable Bit** 0 = TMx_EXT (x= 0~1) pin de-bounce Disabled. 1 = TMx_EXT (x= 0~1) pin de-bounce Enabled. **Note1:** If this bit is enabled, the edge detection of TMx_EXT pin is detected with de-bounce circuit. **Note2:** The de-bounce circuit doesn't support ACMP output.
[5]	**CAPIEN**	**Timer External Capture Interrupt Enable Bit** 0 = TMx_EXT (x= 0~1) pin detection Interrupt Disabled.

Bits	Description	
		1 = TMx_EXT (x= 0~1) pin detection Interrupt Enabled. **Note:** CAPIEN is used to enable timer external interrupt. If CAPIEN enabled, timer will generate an interrupt when CAPIF (TIMERx_EINTSTS[0]) is 1. For example, while CAPIEN = 1, CAPEN = 1, and CAPEDGE = 00, an 1 to 0 transition on the TMx_EXT pin will cause the CAPIF to be set then the interrupt signal is generated and sent to NVIC to inform CPU.
[4]	**CAPFUNCS**	**Capture Function Select Bit** 0 = External Capture Mode Enabled. 1 = External Reset Mode Enabled. **Note1:** When CAPFUNCS is 0, transition on TMx_EXT (x= 0~1) pin is using to save the 24-bit timer counter value to CAPDAT register. **Note2:** When CAPFUNCS is 1, transition on TMx_EXT (x= 0~1) pin is using to reset the 24-bit timer counter value.
[3]	**CAPEN**	**Timer External Capture Pin Enable Bit** This bit enables the TMx_EXT pin. 0 = TMx_EXT (x= 0~1) pin Disabled. 1 = TMx_EXT (x= 0~1) pin Enabled.
[2:1]	**CAPEDGE**	**Timer External Capture Pin Edge Detection** 00 = A falling edge on TMx_EXT (x= 0~1) pin will be detected. 01 = A rising edge on TMx_EXT (x= 0~1) pin will be detected. 10 = Either rising or falling edge on TMx_EXT (x= 0~1) pin will be detected. 11 = Reserved.
[0]	**CNTPHASE**	**Timer External Count Phase** This bit indicates the detection phase of external counting pin TMx (x= 0~1). 0 = A falling edge of external counting pin will be counted. 1 = A rising edge of external counting pin will be counted.

Timer External Interrupt Status Register (TIMERx_EINTSTS)

Register	Offset	R/W	Description	Reset Value
TIMER0_EINTSTS	TMR_BA+0x18	R/W	Timer0 External Interrupt Status Register	0x0000_0000
TIMER1_EINTSTS	TMR_BA+0x38	R/W	Timer1 External Interrupt Status Register	0x0000_0000

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

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**CAPIF**	**Timer External Capture Interrupt Flag** This bit indicates the timer external capture interrupt flag status. 0 = TMx_EXT (x= 0~1) pin interrupt did not occur. 1 = TMx_EXT (x= 0~1) pin interrupt occurred. **Note1:** This bit is cleared by writing 1 to it. **Note2:** When CAPEN (TIMERx_EXTCTL[3]) bit is set, CAPFUNCS (TIMERx_EXTCTL[4]) bit is 0, and a transition on TMx_EXT (x= 0~1) pin matched the CAPEDGE (TIMERx_EXTCTL[2:1]) setting, this bit will set to 1 by hardware. **Note3:** 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.

6.7 Enhanced PWM Generator

6.7.1 Overview

The NuMicro® Mini58 series has built in one PWM unit (PWM0) which is specially designed for motor driving control applications. The PWM0 supports six PWM generators which can be configured as six independent PWM outputs, PWM0_CH0~PWM0_CH5, or as three complementary PWM pairs, (PWM0_CH0, PWM0_CH1), (PWM0_CH2, PWM0_CH3) and (PWM0_CH4, PWM0_CH5) with three programmable dead-time generators.

Every complementary PWM pairs share one 8-bit prescaler. There are six clock dividers providing five divided frequencies (1, 1/2, 1/4, 1/8, 1/16) for each channel. Each PWM output has independent 16-bit counter for PWM period control, and 16-bit comparators for PWM duty control. The six PWM generators provide twelve independent PWM interrupt flags which are set by hardware when the corresponding PWM period counter comparison matched period and duty. Each PWM interrupt source with its corresponding enable bit can request PWM interrupt. The PWM generators can be configured as One-shot mode to produce only one PWM cycle signal or Auto-reload mode to output PWM waveform continuously.

To prevent PWM driving output pin with unsteady waveform, the 16-bit period down counter and 16-bit comparator are implemented with double buffer. When user writes data to counter/comparator buffer registers, the updated value will be loaded into the 16-bit down counter/ comparator at the end of current period. The double buffering feature avoids glitch at PWM outputs.

Besides PWM, Motor controlling also need Timer, ACMP and ADC to work together. In order to control motor more precisely, we provide some registers that not only configure PWM but also Timer, ADC and ACMP, by doing so, it can save more CPU time and control motor with ease especially in BLDC.

6.7.2 Features

The PWM0 supports the following features:

- Six independent 16-bit PWM duty control units with maximum six port pins:
  - Six independent PWM outputs – PWM0_CH0, PWM0_CH1, PWM0_CH2, PWM0_CH3, PWM0_CH4, and PWM0_CH5
  - Three complementary PWM pairs, with each pin in a pair mutually complement to each other and capable of programmable dead-time insertion – (PWM0_CH0, PWM0_CH1), (PWM0_CH2, PWM0_CH3) and (PWM0_CH4, PWM0_CH5)
  - Three synchronous PWM pairs, with each pin in a pair in-phase – (PWM0_CH0, PWM0_CH1), (PWM0_CH2, PWM0_CH3) and (PWM0_CH4, PWM0_CH5)
- Group control bit – PWM0_CH2 and PWM0_CH4 are synchronized with PWM0_CH0, PWM0_CH3 and PWM0_CH5 are synchronized with PWM0_CH1
- One-shot (only support edge-aligned type) or Auto-reload mode PWM
- Up to 16-bit resolution
- Supports edge-aligned, center-aligned and precise center-aligned mode
- Supports asymmetric PWM generating in center-aligned and precise center-aligned mode
- Supports center loading in center-aligned and precise center-aligned mode
- Programmable dead-time insertion between complementary paired PWMs

- Each pin of PWM0_CH0 to PWM0_CH5 has independent polarity setting control
- Hardware fault brake protections
  - Supports software trigger
  - Two Interrupt source types:
    - Synchronously requested at PWM frequency when down counter comparison matched (edge- and center-aligned type) or underflow (edge-aligned type)
    - Requested when external fault brake asserted
      - BKP0: EINT0 or CPO1
      - BKP1: EINT1 or CPO0
- The PWM signals before polarity control stage are defined in the view of positive logic. The PWM ports is active high or active low are controlled by polarity control register
- Supports mask aligned function
- Supports independently rising CMP matching, PERIOD matching, falling CMP matching (in Center-aligned type), period matching to trigger ADC conversion
- Timer comparing matching event trigger PWM to do phase change in BLDC application
- Supports ACMP output event trigger PWM to force PWM output at most one period low, this feature is usually for step motor control
- Provides interrupt accumulation function

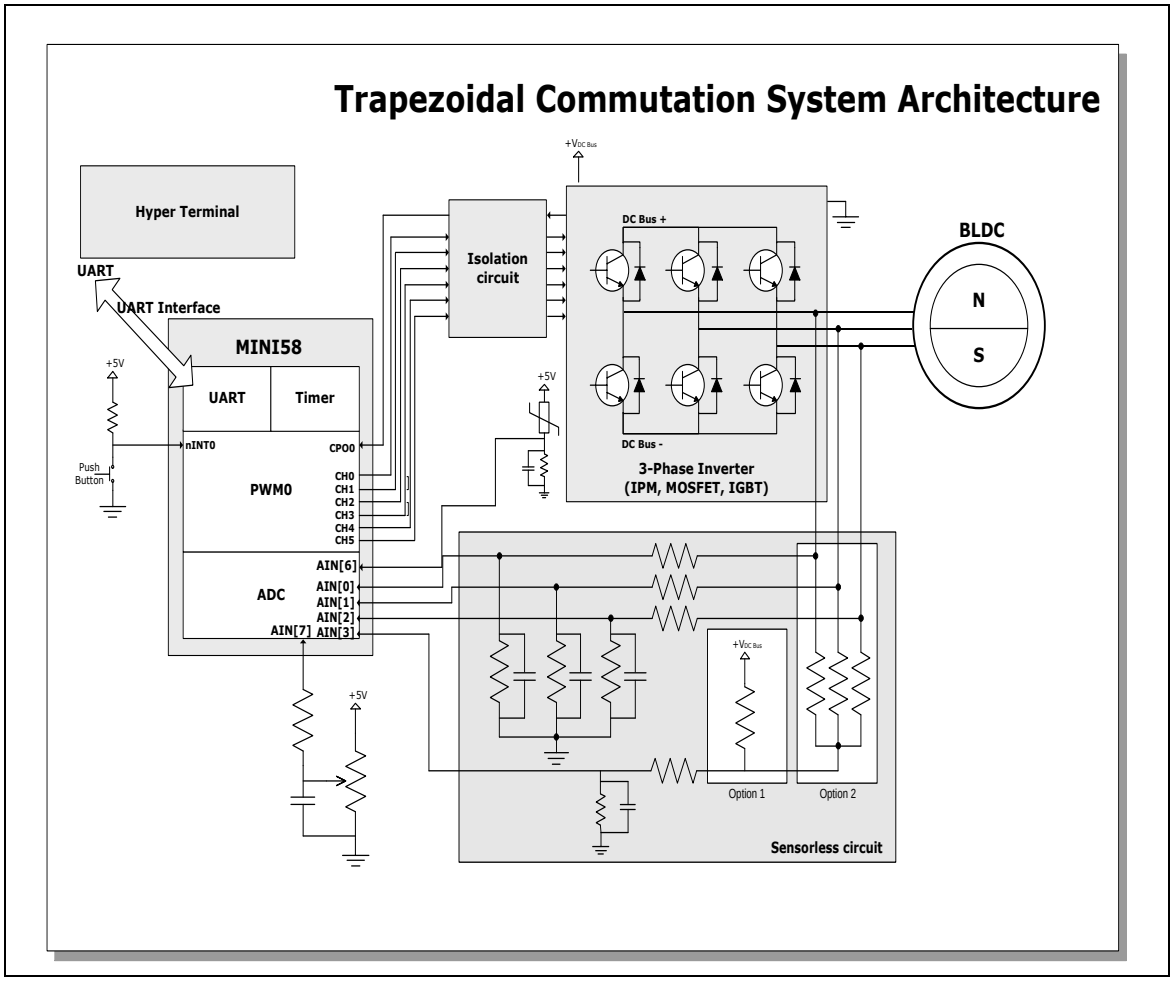


Figure 6.7-1 Application Circuit Diagram

6.7.3 Block Diagram

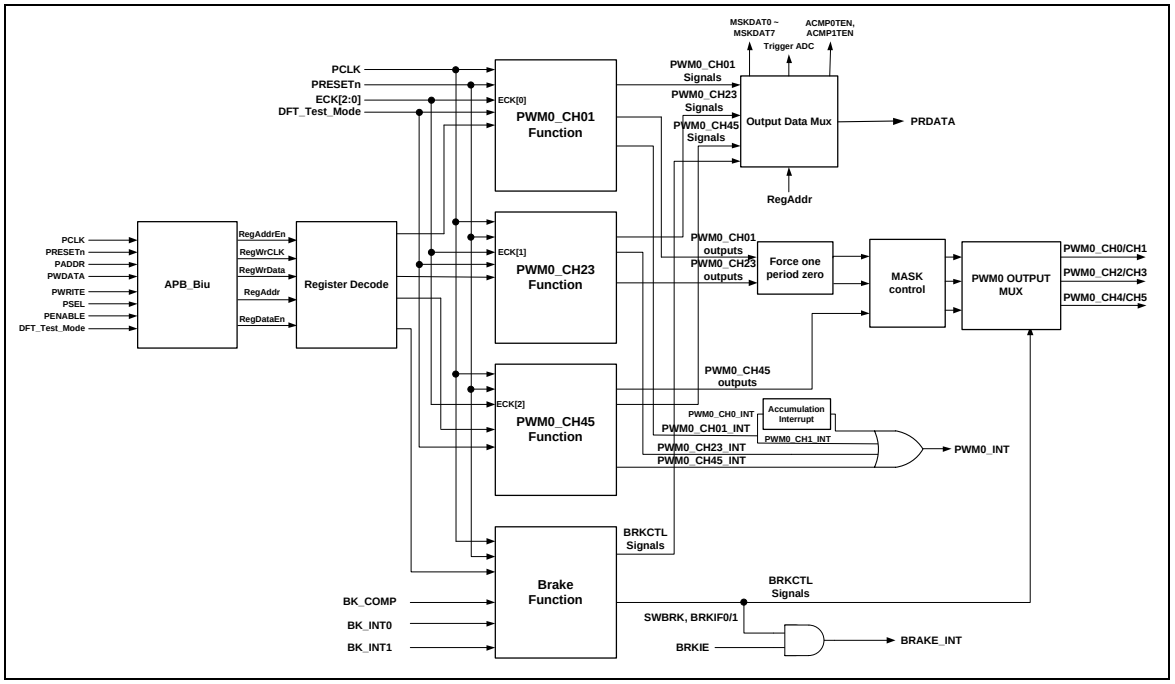


Figure 6.7-2 PWM Block Diagram

Figure 6.7-3 shows the architecture of PWM in pair (e.g. PWM Counter 0/1 are in one pair and PWM Counter 2/3 are in another one, and so on).

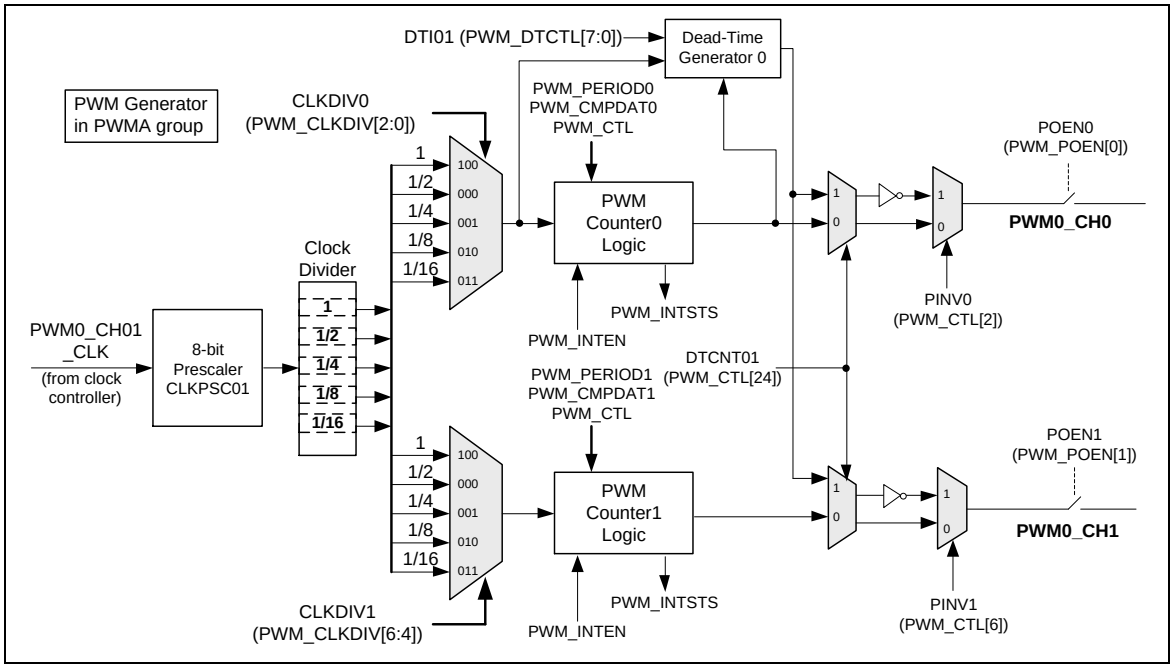


Figure 6.7-3 PWM Generator 0 Architecture Diagram

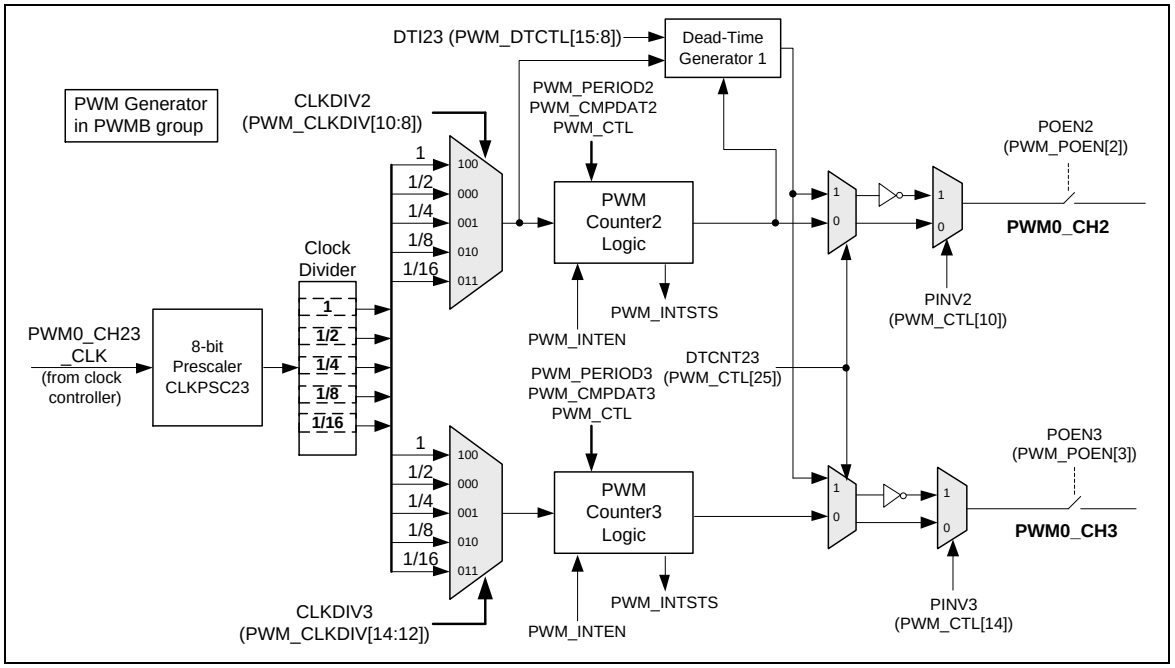


Figure 6.7-4 PWM Generator 2 Architecture Diagram

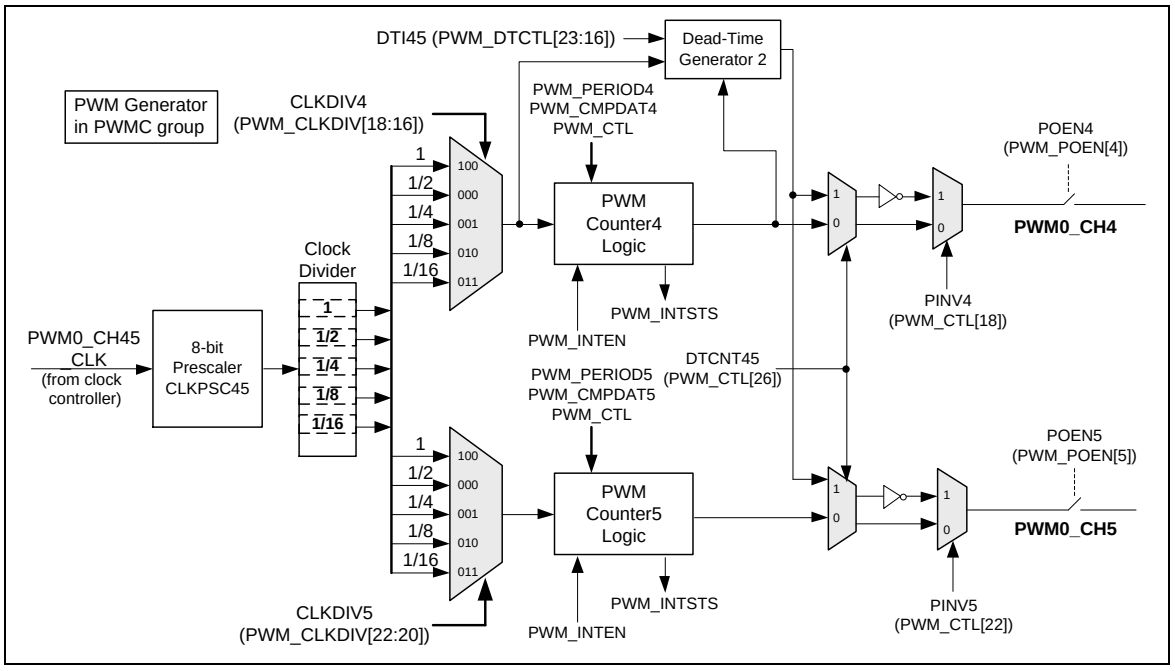


Figure 6.7-5 PWM Generator 4 Architecture Diagram

6.7.4 Basic Configuration

The PWM0 pin functions are configured in SYS_P0_MFP and SYS_P2_MFP registers.

The PWM0 clock can be enabled in CLK_APBCLK[22:20]. The PWM0 clock source must be PCLK.

6.7.5 Functional Description

6.7.5.1 PWM Counter Operation

This device supports three operation types: Edge-aligned, Center-aligned and Precise center-aligned type.

Following equations show the formula for period and duty for each PWM counter operation type:

Edge-aligned (Down counter)

Duty ratio = (CMP + 1) / (PERIOD + 1)

Duty = (CMP + 1) * (clock period)

Period = (PERIOD + 1) * (clock period)

Center-aligned (Up and Down Counter):

Duty ratio = (PERIOD - CMP) / (PERIOD + 1)

Duty = (PERIOD - CMP) * 2 * (clock period)

Period = (PERIOD + 1) * 2 * (clock period)

Precise Center-aligned (Up and Down Counter):

Duty ratio = (PERIOD - (CMP + 1) * 2) / PERIOD

Duty = (PERIOD - (CMP + 1) * 2) * (clock period)

Period = (PERIOD) * (clock period)

Edge-aligned PWM (Down-counter)

In Edge-aligned PWM Output type, the 16-bit PWM counter will start counting-down from PERIODn to match with the value of the duty cycle CMPn (old); when this happens it will toggle the PWM0_CHn output to high and set up CMPDIF compare down match interrupt flag. The counter will continue counting-down to zero; at this moment, it toggles the PWM0_CHn output to low and CMPn (new) and PERIODn (new) are updated with CNTMODEn=1 and set PIF period interrupt flag.

Figure 6.7-6, Figure 6.7-7 and Figure 6.7-8 show the Edge-aligned PWM timing and operation flow.

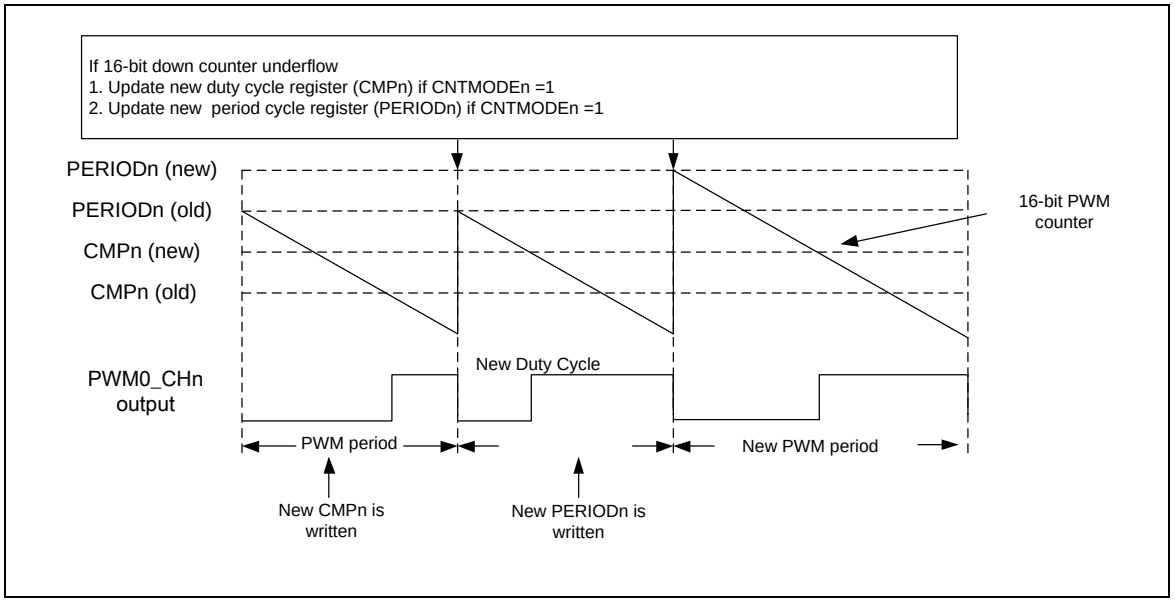


Figure 6.7-6 Edge-aligned Type PWM

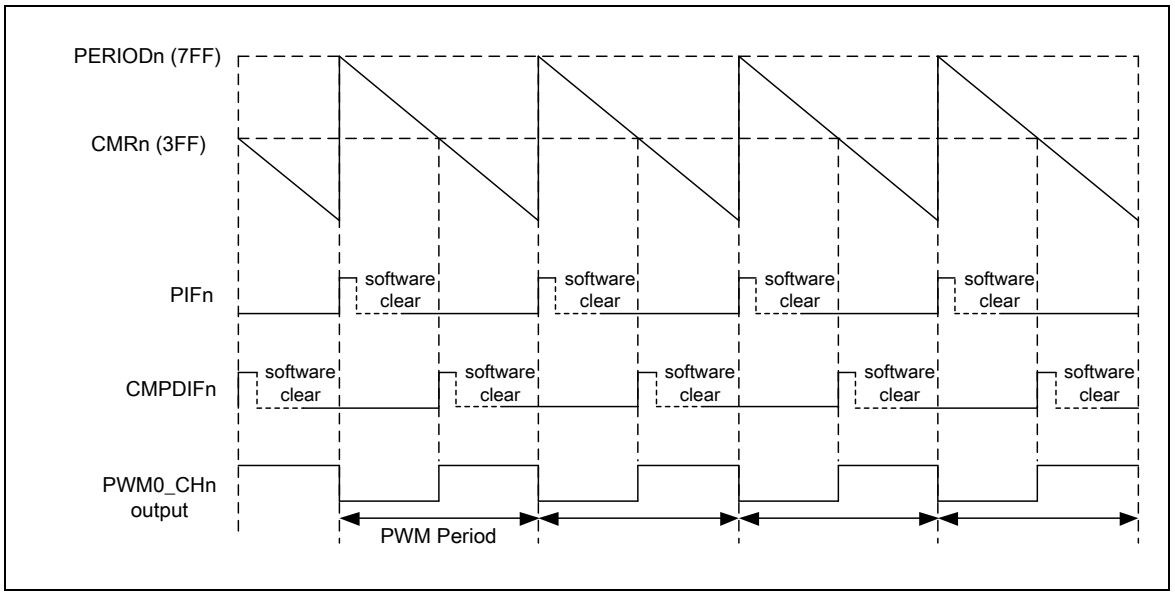


Figure 6.7-7 PWM Edge-aligned Waveform Timing Diagram

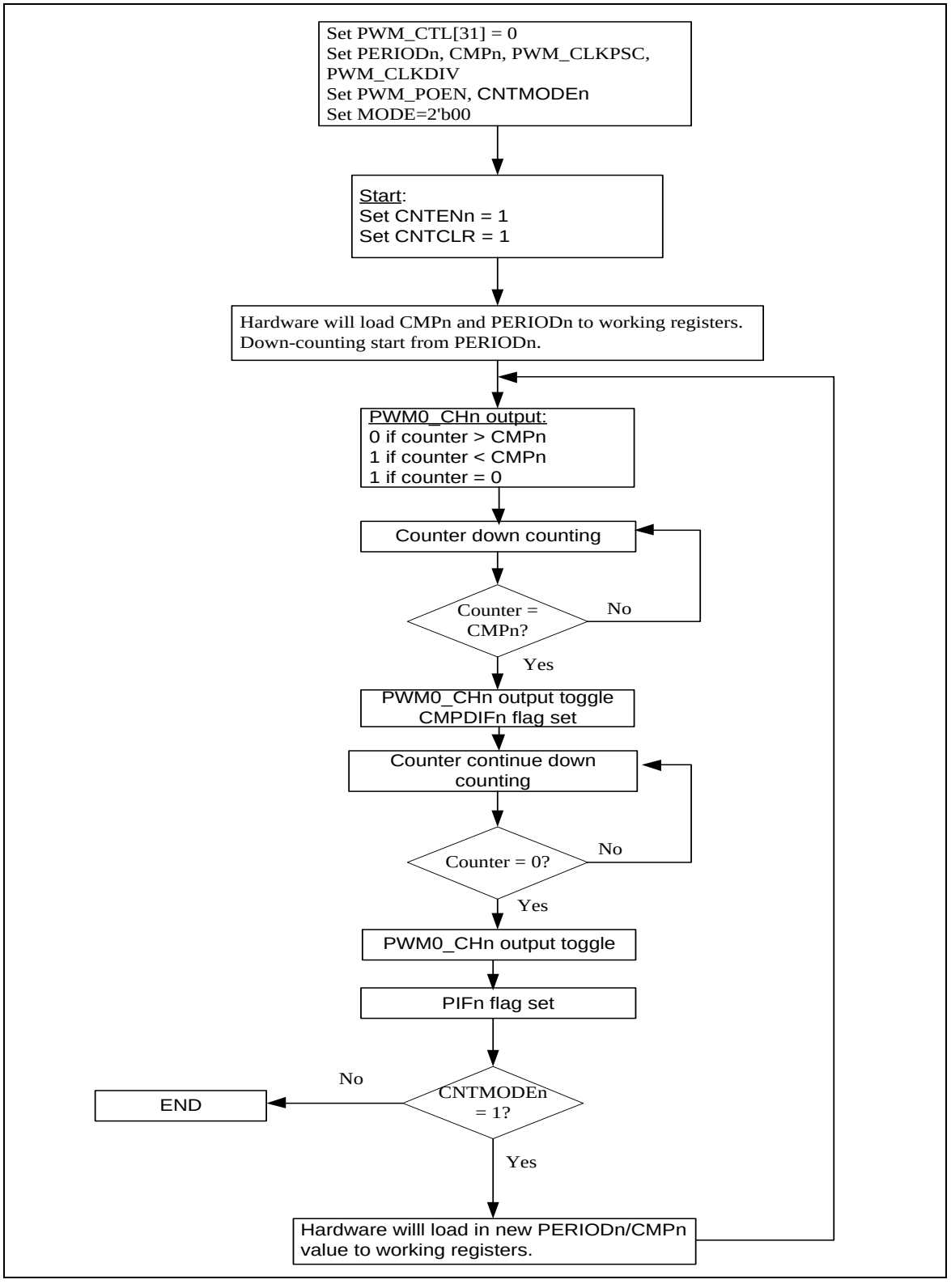


Figure 6.7-8 Edge-aligned Flow Diagram

The PWM period and duty control are decided by PWM down-counter period register (PERIODn) and PWM comparator register (CMPn). The PWM counter timing operation is shown in Figure

6.7-10. The pulse width modulation follows the formula below and the legend of PWM counter Comparator is shown in Figure 6.7-9. Note that the corresponding GPIO pins must be configured as PWM function (enable PWM_POEN) for the corresponding PWM channel.

PWM frequency = HCLK / ((CLKPSCnm + 1) * (clock divider)) / (PERIOD + 1); where nm, could be 01, 23 or 45 depending on the selected PWM channel

Duty ratio = (CMP + 1) / (PERIOD + 1)

CMP >= PERIOD: PWM output is always high

CMP < PERIOD: PWM low width = (PERIOD - CMP) unit[1]; PWM high width = (CMP+1) unit

CMP = 0: PWM low width = (PERIOD) unit; PWM high width = 1 unit

Note: 1. Unit = one PWM clock cycle.

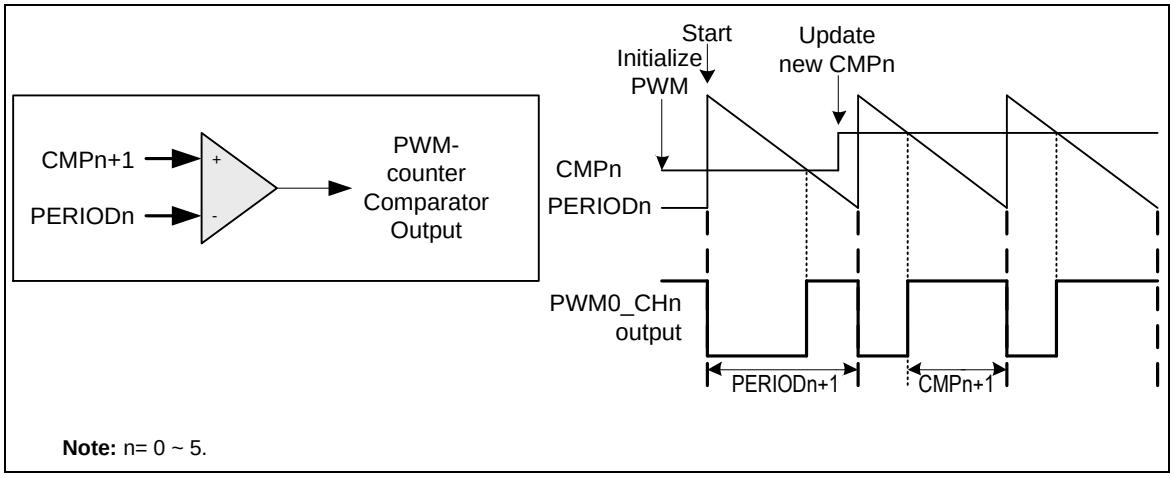


Figure 6.7-9 Legend of Internal Comparator Output of PWM Counter

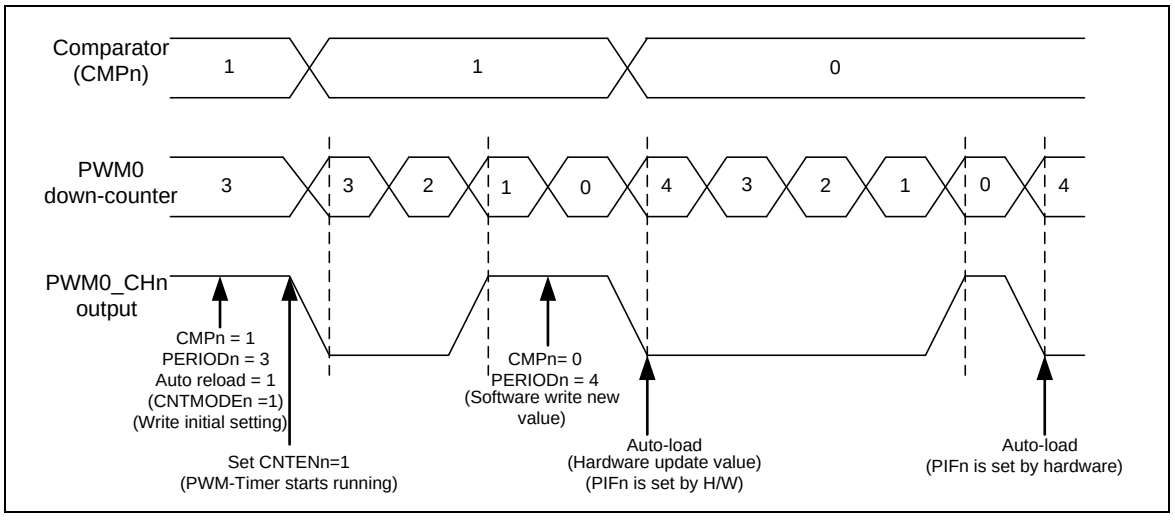


Figure 6.7-10 PWM Counter Operation Timing

Center-Aligned PWM (up/down counter)

The center-aligned PWM signals are produced by the module when the PWM time base is configured in an Up/Down Counting type. The PWM counter will start counting-up from 0 to match the value of CMPn (old); this will cause the toggling of the PWM0_CHn generator output to high and set up CMPUIF compare up match interrupt flag. The counter will continue counting to match with the PERIODn (old). Upon reaching this state counter is configured automatically to down counting and set up PIF period interrupt flag, when PWM counter matches the CMPn (old) value again the PWM0_CHn generator output toggles to low and set up CMPDIF compare down match interrupt flag. Once the PWM counter underflows it will update the PWM period register PERIODn (new) and duty cycle register CMPn (new) with CNTMODEn = 1.

In Center-aligned type, the PWM period interrupt can also be requested at down-counter underflow if PINTTYPE (PWM_INTEN[17]) =0, i.e. at start (end) of each PWM cycle or at up-counter matching with PERIODn if PINTTYPE (PWM_INTEN[17]) =1, i.e. at center point of PWM cycle.

PWM frequency = HCLK / ((CLKPSCnm + 1) * (clock divider)) / (PERIOD + 1); where nm, could be 01, 23 or 45 depending on the selected PWM channel

Duty ratio = (PERIOD - CMP) / (PERIOD + 1)

CMP >= PERIOD: PWM output is always low

CMP < PERIOD: PWM low width = (CMP + 1) * 2 units; PWM high width= (PERIOD - CMP) * 2 units[1]

CMP = 0: PWM low width = 2 units; PWM high width = PERIOD * 2 units

Note: 1. Unit = one PWM clock cycle.

Figure 6.7-11 ~ Figure 6.7-13 and Figure 6.7-14 show the Center-aligned PWM timing and operation flow.

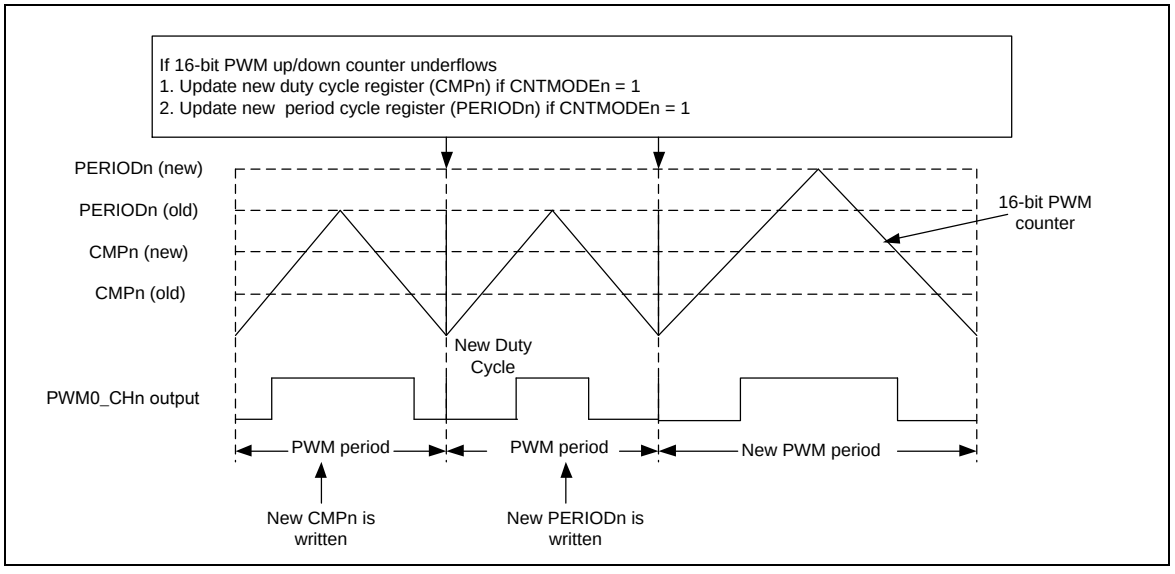


Figure 6.7-11 Center-aligned Type PWM

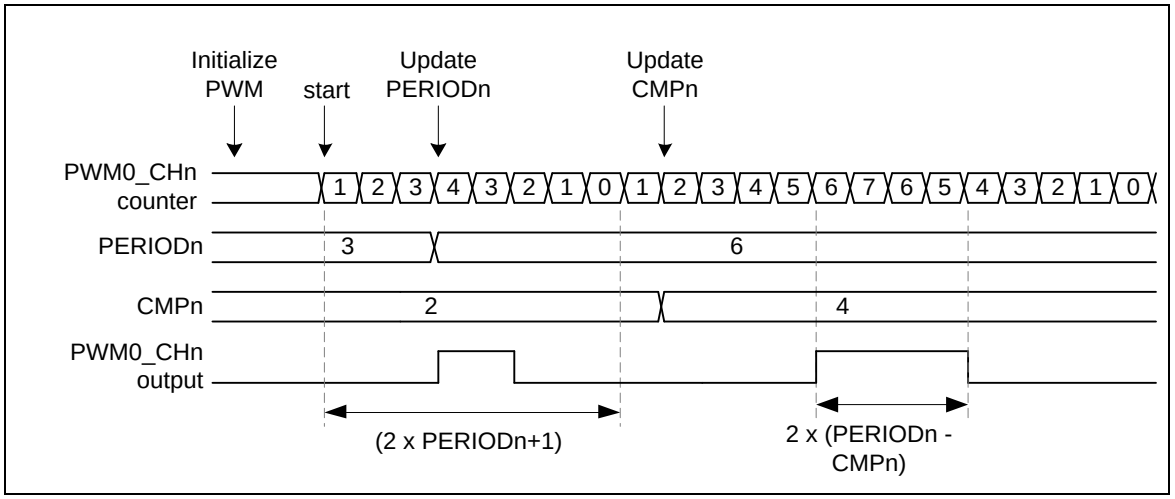


Figure 6.7-12 Center-aligned Type Operation Timing

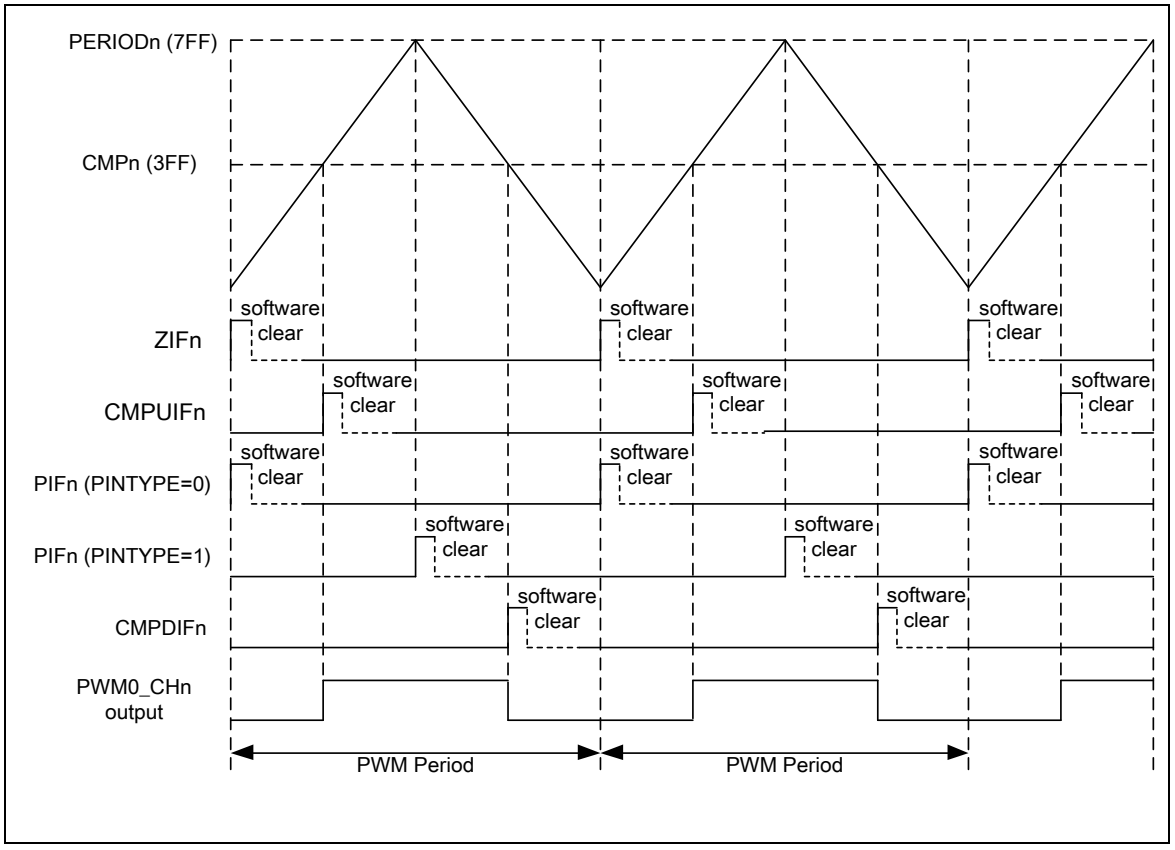


Figure 6.7-13 PWM Center-aligned Waveform Timing Diagram

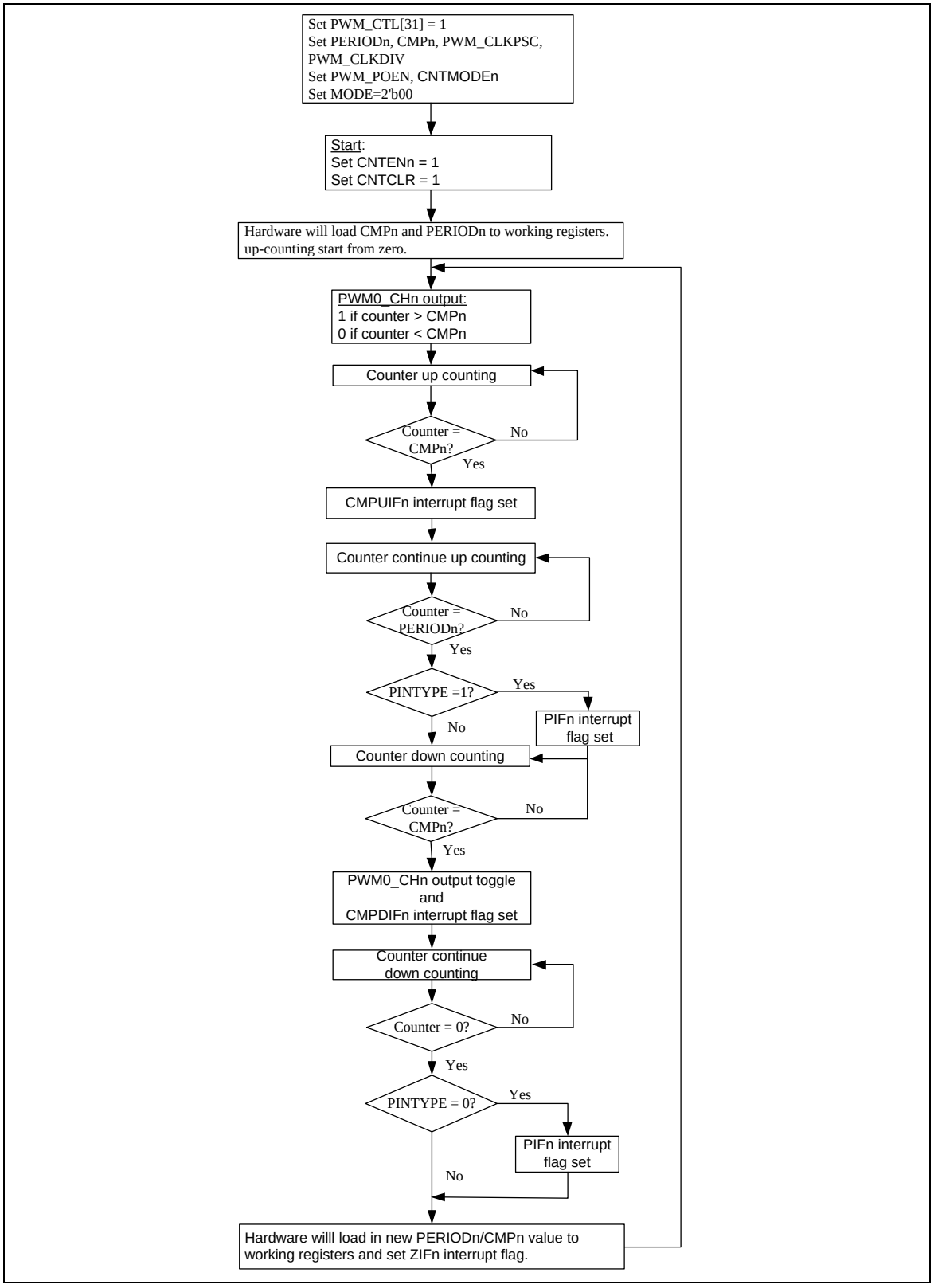


Figure 6.7-14 Center-aligned Flow Diagram

Precise Center-Aligned PWM (Up/Down Counter)

The precise center-aligned PWM signals are produced by the module when the PWM time base is configured in an Up/Down Counting type and enable PCAEN (PWM_PCACTL[0]). The PWM counter will start counting-up from 0 to match the value of CMPn (old); this will cause the toggling of the PWM0_CHn output to high. The counter will continue counting to match with the half of the PERIODn (old). If PERIODn is an odd number, the counter will continue counting to match the integer of lower boundary of the half of the PERIODn (old) and keep the counter value for two clock cycles. Upon reaching this state counter is configured automatically to down counting, when PWM counter matches the CMPn (old) value again the PWM0_CHn output toggles to low. Once the PWM counter underflows it will update the PWM period register PERIODn (new) and duty cycle register CMPn (new) with CNTMODEn = 1.

In Precise Center-aligned type, the PWM period interrupt can also be requested at down-counter underflow if PINTTYPE (PWM_INTEN[17]) =0, i.e. at start (end) of each PWM cycle or at up-counter matching with PERIODn if PINTTYPE (PWM_INTEN[17]) =1, i.e. at center point of PWM cycle.

PWM frequency = HCLK / ((CLKPSCnm + 1) * (clock divider)) / (PERIOD + 1); where nm, could be 01, 23 or 45 depending on the selected PWM channel

Duty ratio = (PERIOD – (CMP+1)*2) / PERIOD

CMP >= PERIOD: PWM output is always low

CMP < PERIOD: PWM low width = (CMP + 1) * 2 units; PWM high width= (PERIOD – (CMP+1)*2) units[1]

CMP = 0: PWM low width = 2 units; PWM high width = PERIOD - 2 units

Note: 1. Unit = one PWM clock cycle.

Figure 6.7-15, Figure 6.7-16 and Figure 6.7-17 show the Precise Center-aligned PWM timing and operation flow.

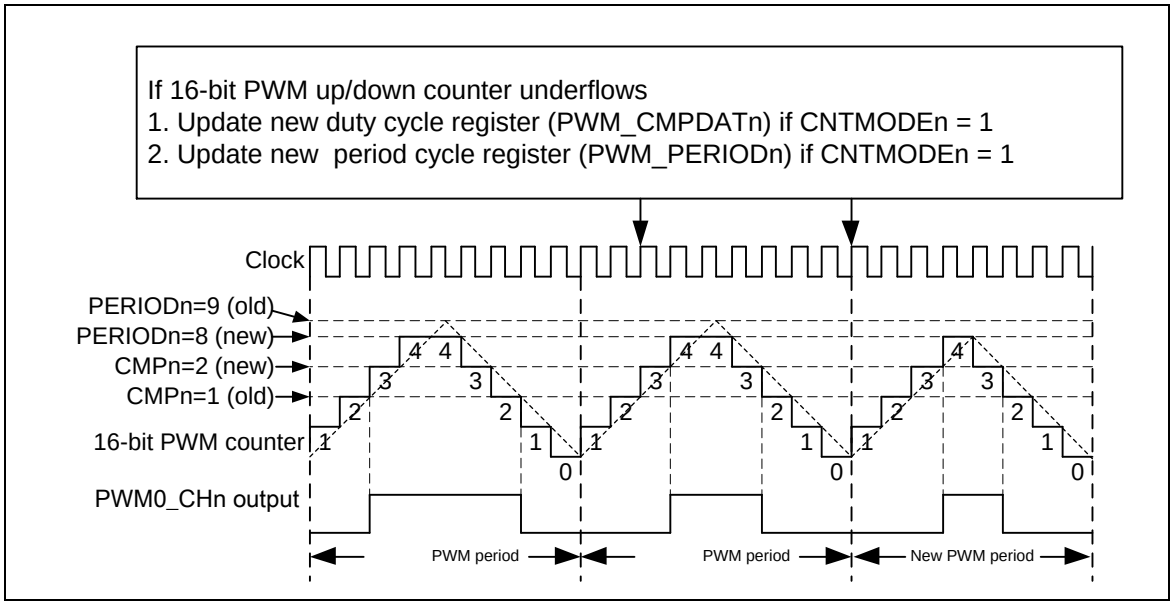


Figure 6.7-15 Precise Center-aligned Type PWM

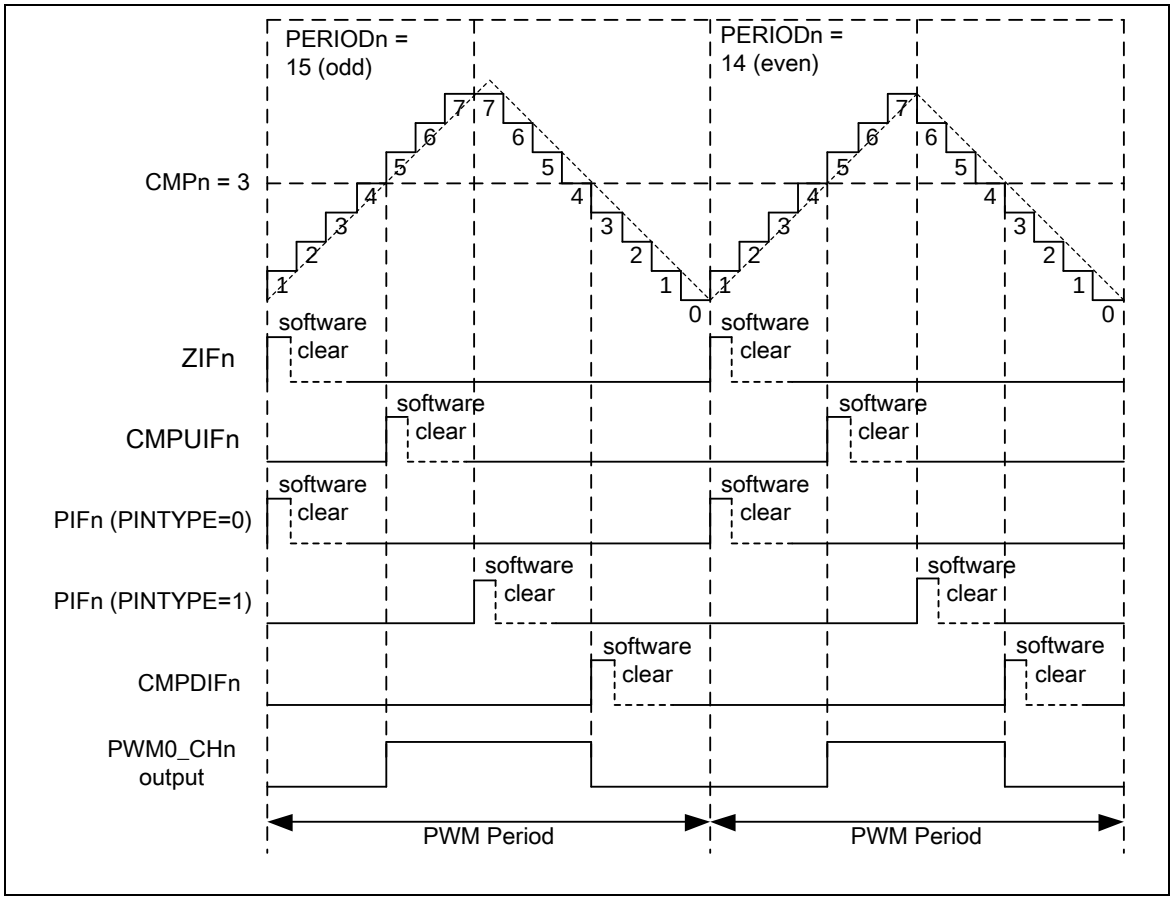


Figure 6.7-16 PWM Precise Center-aligned Waveform Timing Diagram

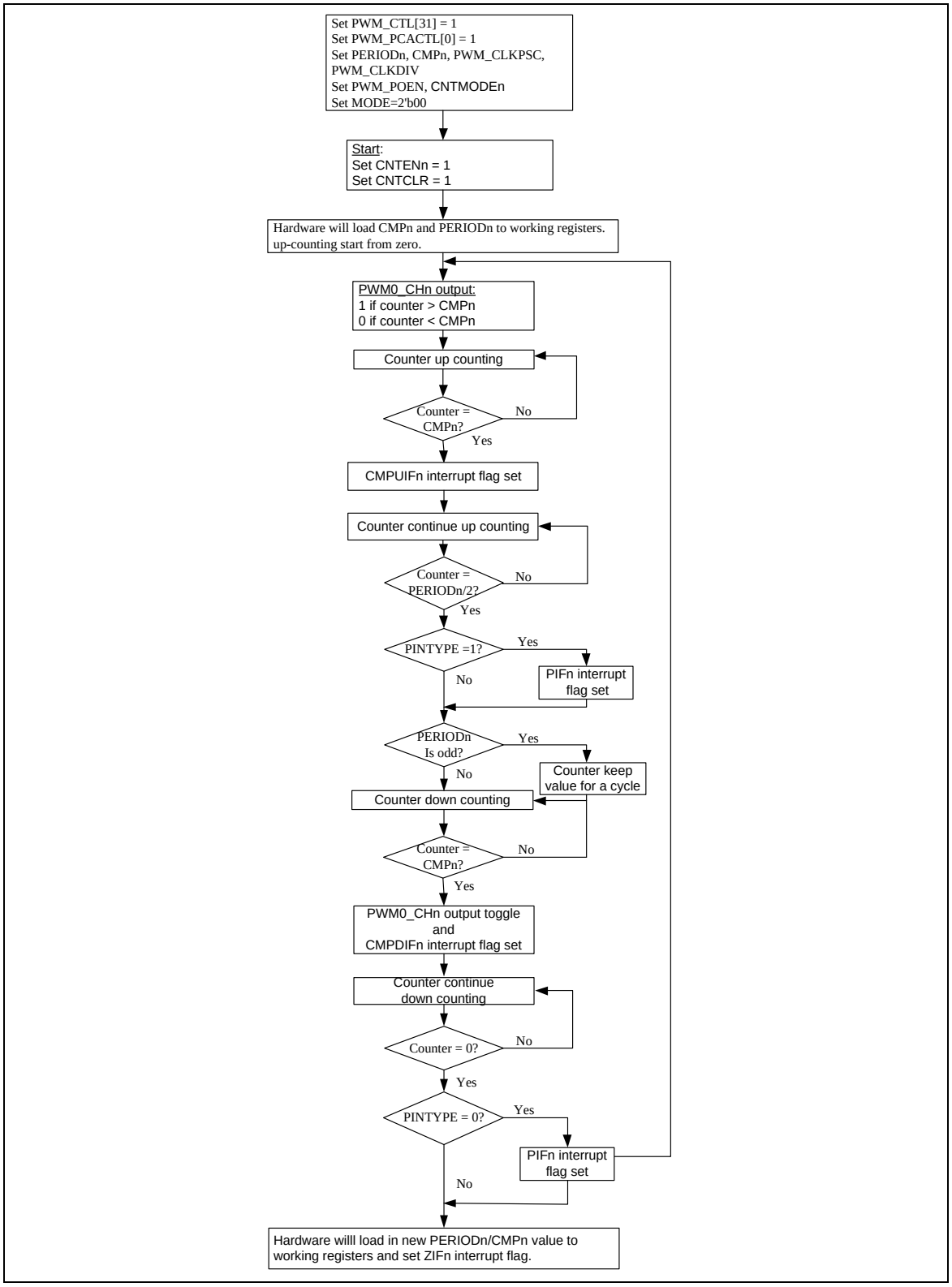


Figure 6.7-17 Precise Center-aligned Flow Diagram

6.7.5.2 PWM Center Loading Operation

In center-aligned or precise center-aligned type, PWM also supports loading new PERIODn, CMPn. If operating in asymmetric mode, CMPDn will also supports center loading operation.

When counter counting to center of PWM period. By setting HCUPDT (PWM_CTL[5]) to enable this function. Figure 6.7-18 shows an example of center loading operation, when counter counts to original center 4; it updates its value to PERIODn 6 then continues counting down.

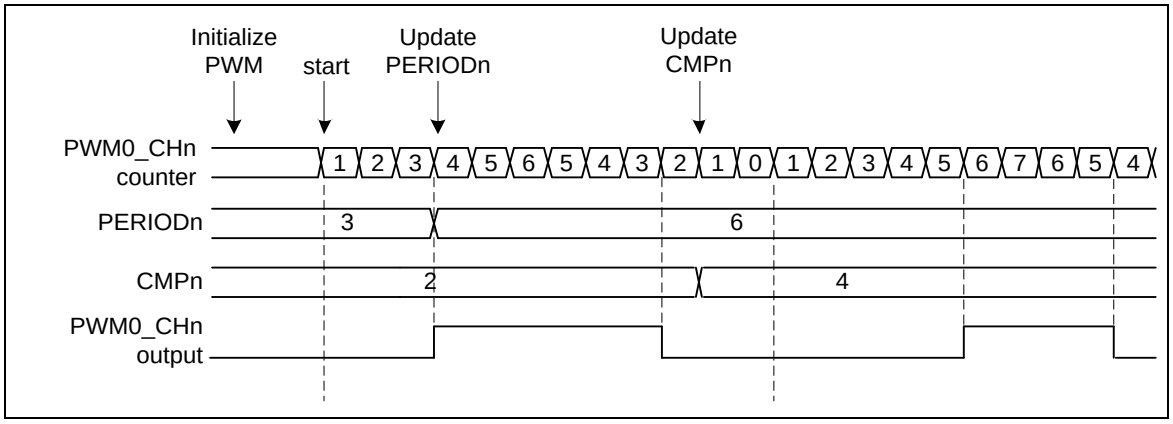


Figure 6.7-18 PWM Center Loading Timing Diagram

6.7.5.3 PWM Double Buffering, Auto-reload and One-shot Operation

The NuMicro® Mini58 series PWM have double buffering function, the reload value is updated at the start of next period without affecting current counter operation. The PWM counter value can be written into PERIODn.

PWM0_CH0 will operate in One-shot mode if CNTMODE0 bit is set to 0, and operate in Auto-reload mode if CNTMODE0 bit is set to 1. It is recommended that switch PWM0_CH0 operating mode before set CNTEN0 bit to 1 to enable PWM0_CH0 counter to start running because the content of PERIOD0 and CMP0 will be cleared to 0 to reset the PWM0_CH0 period and duty setting when PWM0_CH0 operating mode is changed. As PWM0_CH0 operates in One-shot mode, CMP0 and PERIOD0 should be written first and then the CNTEN0 bit set to 1 to enable PWM0_CH0 counter to start running. After the PWM0_CH0 counter down counts from PERIOD0 value to 0, PERIOD0 and CMP0 will be cleared to 0 by hardware and counter will be held. New CMP0 and PERIOD0 value needs to be written by software to set the next one-shot period and duty. When re-starting next one-shot operation, the CMP0 should be written first because PWM0_CH0 counter will automatically re-start counting when PERIOD0 is written a non-zero value. As PWM0_CH0 operates at auto-reload mode, CMP0 and PERIOD0 should be written first and then set CNTEN0 bit to 1 to enable PWM0_CH0 counter to start running. The PERIOD0 value will be reloaded to PWM0_CH0 counter when the down counting reaches 0. If the PERIOD0 is set to 0, PWM0_CH0 counter will be held. PWM0_CH1~PWM0_CH5 performs the same function as PWM0_CH0

Note: One-shot operation only supports edge-aligned type.

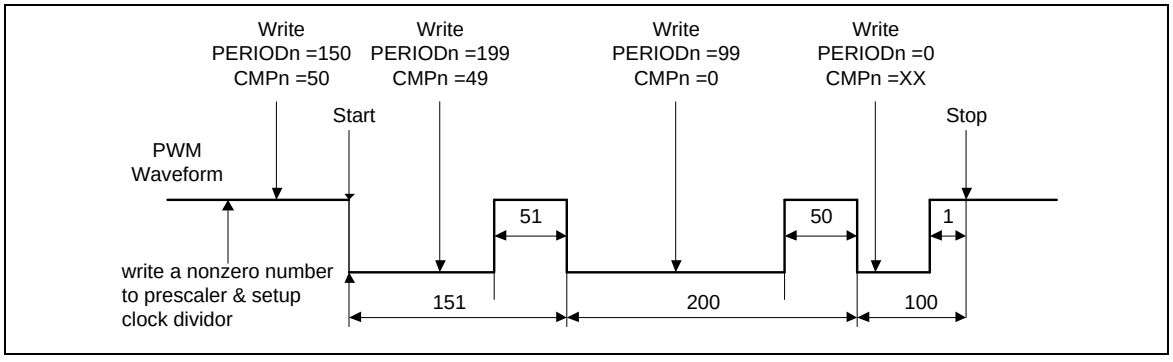


Figure 6.7-19 PWM Double Buffering Illustration

6.7.5.4 *Modulate Duty Ratio*

The double buffering function allows CMPn to be written at any point in the current cycle. The loaded value will take effect from the next cycle.

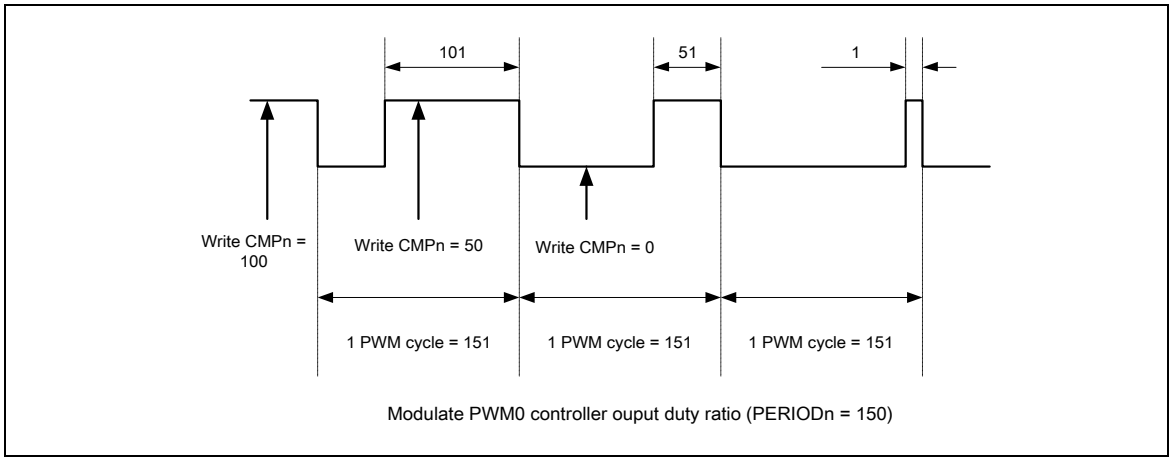


Figure 6.7-20 PWM Controller Output Duty Ratio

6.7.5.5 *PWM Operation Modes*

This powerful PWM unit supports independent mode which may be applied to DC or BLDC motor system, Complementary mode with dead-time insertion which may be used in the application of AC induction motor and synchronous motor, and Synchronous mode that makes both pins of each pair are in phase. Besides, the Group mode, which forces the PWM0_CH2 and PWM0_CH4 synchronous with PWM0_CH0 generator, may simplify updating duty control in DC and BLDC motor applications and Asymmetric mode, to generate asymmetric PWM waveform and interrupt timing.

6.7.5.6 *Independent Mode*

Independent mode is enabled when MODE (PWM_CTL[29:28]) = 00.

By default, the PWM is operated in independent mode, with six PWM channels outputs. Each channel is running off its own duty-cycle generator module.

6.7.5.7 *Complementary Mode*

Complementary mode is enabled when MODE (PWM_CTL[29:28]) = 01.

In this module there are three duty-cycle generators utilized for complementary mode, with total of three PWM output pair pins in this module. The total six PWM outputs are grouped into output pairs of even and odd numbered outputs. In complimentary modes, the internal odd PWM signal PWM0_CHn, always be the complement of the corresponding even PWM signal. For example, PWM0_CH1 will be the complement of PWM0_CH0. PWM0_CH3 will be the complement of PWM0_CH2 and PWM0_CH5 will be the complement of PWM0_CH4. The time base for the PWM module is provided by its own 16-bit counter, which also incorporates selectable pre-scalar options.

6.7.5.8 Dead-time Insertion

The dead-time generator inserts an “off” period called “dead-time” between the turnings off of one pin to the turning on of the complementary pin of the paired pins. This is to prevent damage to the power switching devices that will be connected to the PWM output pins. The complementary output pair mode has an 8-bit down counter used to produce the dead-time insertion. The complementary outputs are delayed until the counter counts down to zero.

The dead-time can be calculated from the following formula:

dead-time = PWM_CLK * (DTInm+1). where nm, could be 01, 23, 45

The timing diagram as shown below indicates the dead-time insertion for one pair of PWM signals.

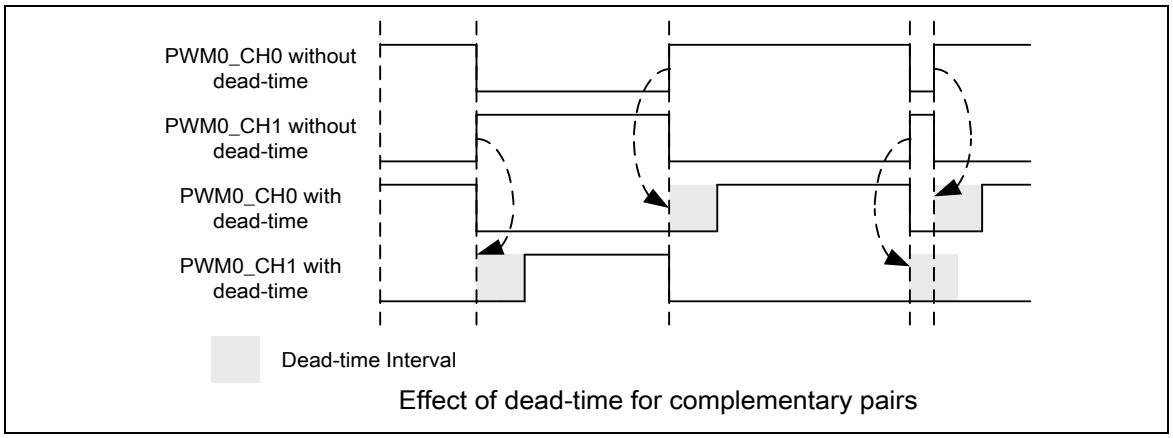


Figure 6.7-21 Dead-time Insertion

In Power inverter applications, a dead-time insertion avoids the upper and lower switches of the half bridge from being active at the same time. Hence the dead-time control is crucial to proper operation of a system. Some amount of time must be provided between turning off of one PWM output in a complementary pair and turning on the other transistor as the power output devices cannot switch instantaneously.

6.7.5.9 Synchronous Mode

Synchronous mode is enabled when MODE (PWM_CTL[29:28]) = 10.

In the synchronization mode the PWM pair signals from PWM Generator are in-phase.

PWM0_CH1=PWM0_CH0, PWM0_CH3=PWM0_CH2 and PWM0_CH5=PWM0_CH4.

6.7.5.10 Group Mode

Group mode is enabled when GROUPEN (PWM_CTL[30]) = 1.

This device supports Group mode control which allows all even PWM channels output to be duty controllable by PWM0_CH0 duty register.

If GROUPEN = 1, both (PWM0_CH2, PWM0_CH3) and (PWM0_CH4, PWM0_CH5) pairs will follow (PWM0_CH0, PWM0_CH1), which imply;

PWM0_CH4 = PWM0_CH2 = PWM0_CH0;

PWM0_CH5 = PWM0_CH3 = PWM0_CH1 = invert (PWM0_CH0) if Complementary mode is enabled when MODE (PWM_CTL[29:28]) = 01.

Note: For applications, please do not use Group and Synchronous mode simultaneously because

the Synchronous mode will be inactive.

6.7.5.11 Asymmetric Mode

Asymmetric mode only works under Center-aligned type. Asymmetric mode is enabled when ASYMEN (PWM_CTL[21]) = 1. In this mode PWM counter will compare with another compared value CMPDn (PWM_CMPDATn[31:16]) when counting down. If CMRDn is not equal to the CMRn, the PWM will generate asymmetric waveform and set CMPDIFn (PWM_INTSTS[13:8]) of the corresponding channel n.

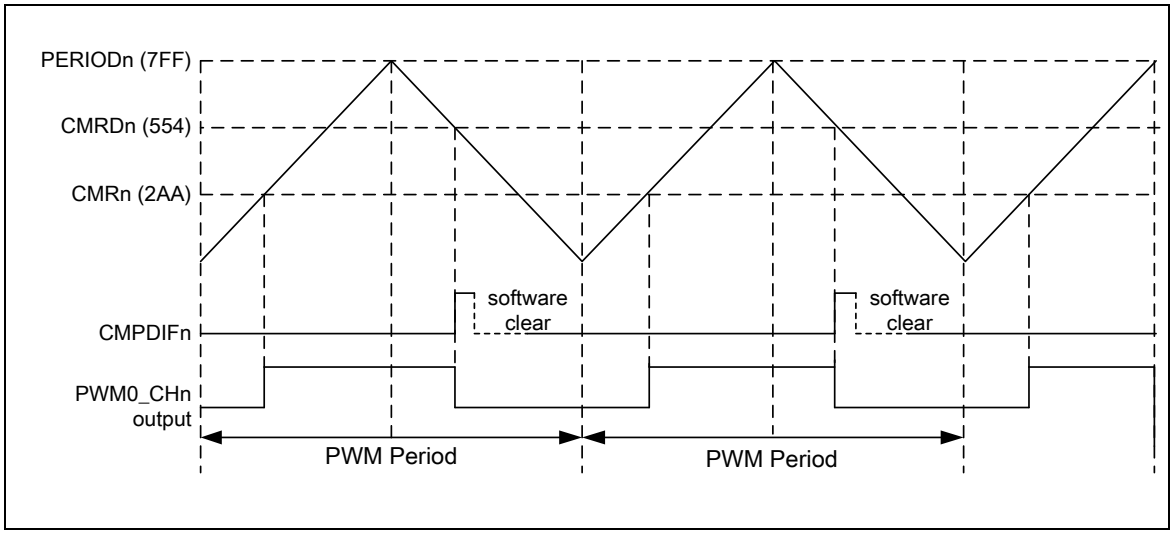


Figure 6.7-22 Asymmetric Mode Timing Diagram

6.7.5.12 Polarity Control

Each PWM port from PWM0_CH0 to PWM0_CH5 has independent polarity control to configure the polarity of active state of PWM output. By default, the PWM output is active high.

Figure 6.7-23 shows the initial state before PWM starts with different polarity settings.

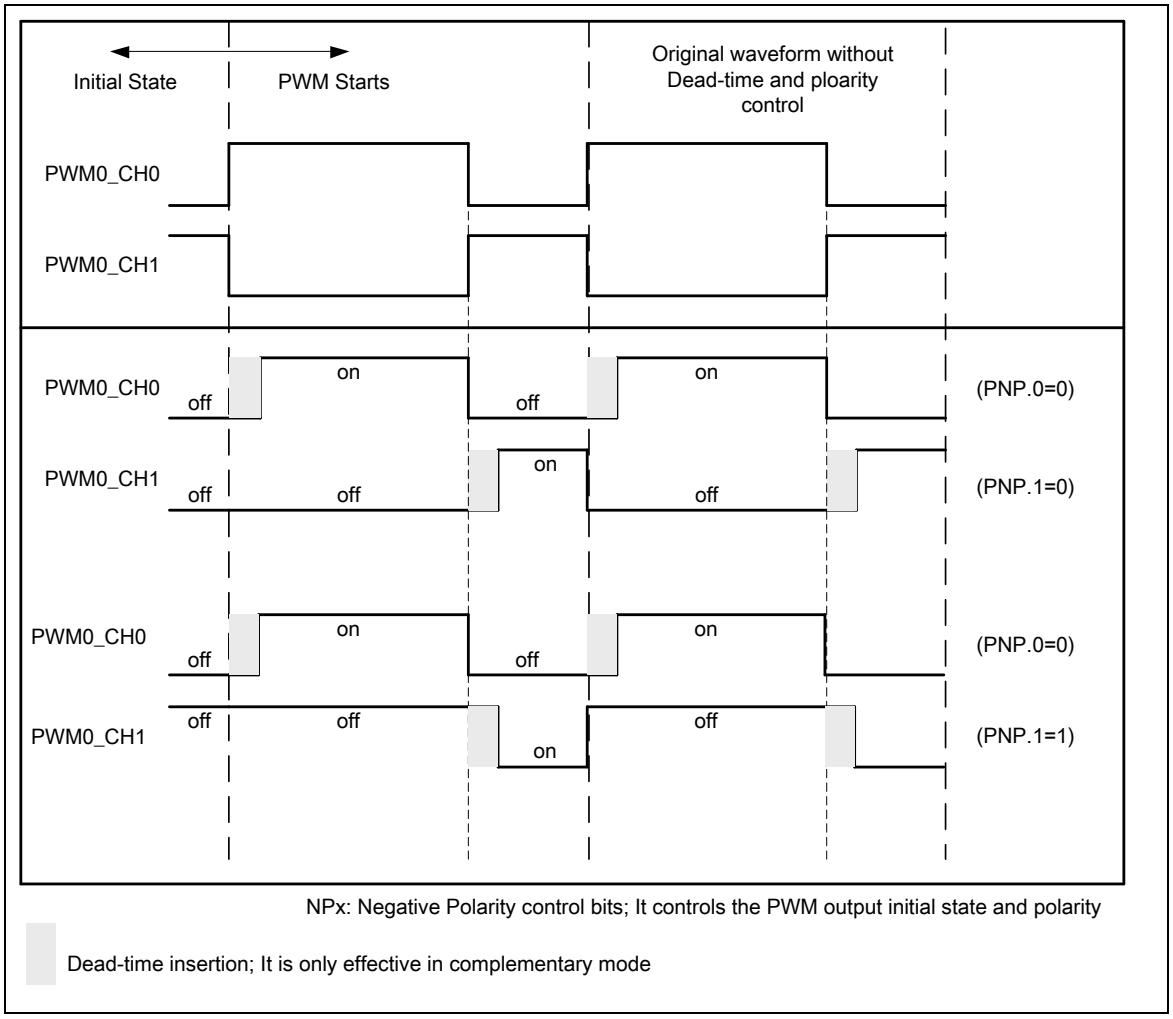


Figure 6.7-23 Initial State and Polarity Control with Rising Edge Dead-time Insertion

6.7.5.13 PWM for Motor Control Interrupt Architecture

There are seven interrupt sources for PWM unit, which are ZIFn (PWM_INTSTS[5:0]) PWM counter count to zero interrupt flag; CMPUIFn (PWM_INTSTS[29:24]) PWM counter up-counts to CMPn (PWM_CMPDATn[15:0]) interrupt flag; PIFn (PWM_INTSTS[23:18]) PWM counter counts to period of edge-aligned type or counts to center of center-aligned type interrupt flag; CMPDIFn (PWM_INTSTS[13:8]) PWM counter down-counts to CMPn (PWM_CMPDATn[15:0]) interrupt flag, if operating in asymmetric type it down count to CMPDn (PWM_CMPDATn[31:16]); BRKIF0 (PWM_INTSTS[16]) Brake0 interrupt flag, BRKIF1 (PWM_INTSTS[17]) Brake1 interrupt flag (BRKIF1) and SWBRK (PWM_BRKCTL[9]) software trigger brake interrupt.

The bit BRKIEN (PWM_INTEN[16]) controls the brake interrupt enable; the bits ZIENn (PWM_INTEN[5:0]) control the ZIFn interrupt enable; the bits CMPUIENn (PWM_INTEN[29:24]) control the CMPUIFn interrupt enable; the bits PIENn (PWM_INTEN[23:18]) control the PIFn interrupt enable; and the bits CMPDIENn (PWM_INTEN[13:8]) control the CMPDIFn interrupt enable. Note that all the interrupt flags are set by hardware and must be cleared by software.

Figure 6.7-24 shows the architecture of Motor Control PWM interrupts.

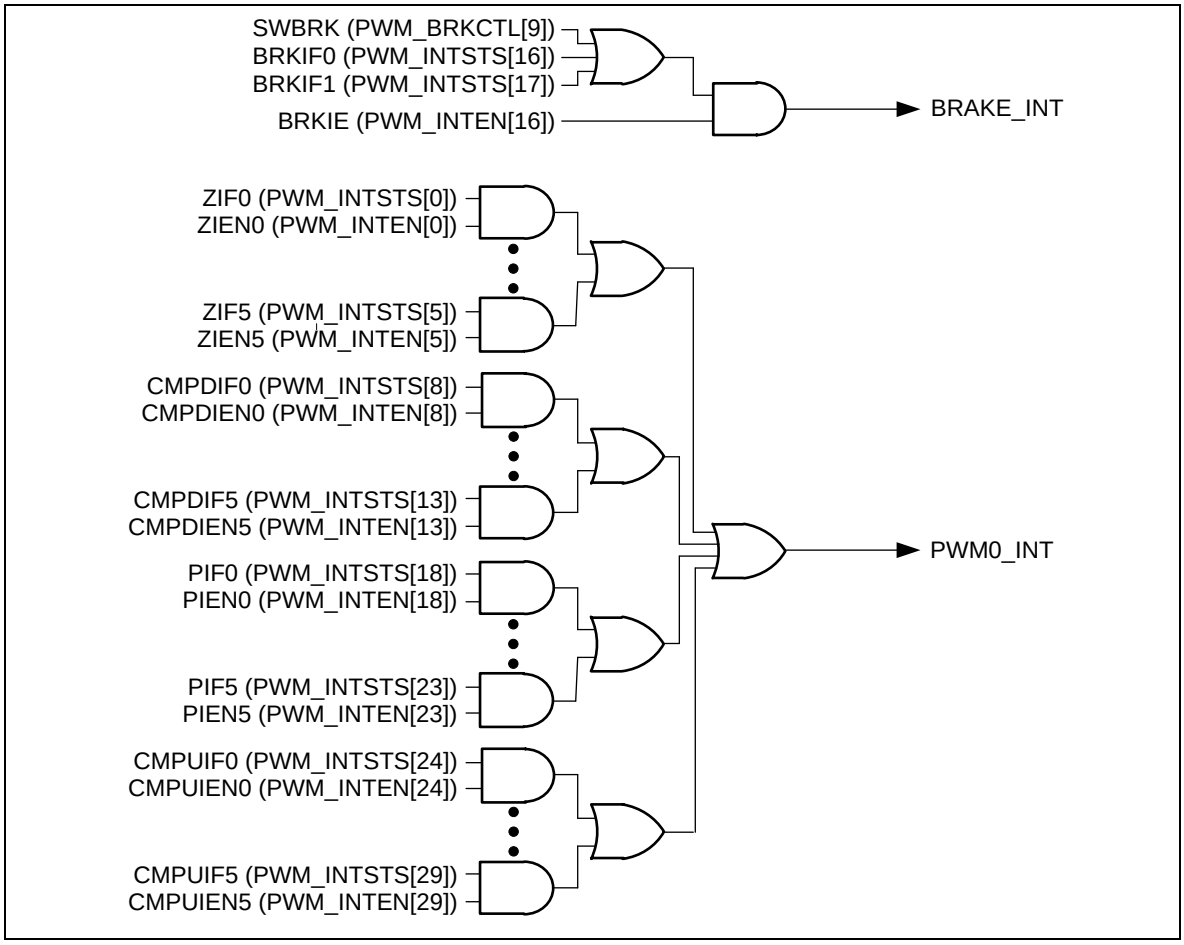


Figure 6.7-24 Motor Control PWM Interrupt Architecture

6.7.5.14 PWM Brake Function

PWM supports fault brake function to control PWM's 6 channels to output level of BKODn (PWM_BRKCTL[29:24]); channel 7 (GPIO P0.0) and channel 6 (GPIO P0.1) to output level of D7BKOD and D6BKOD while brake condition occurs.

This device supports two external brake pins as brake condition: BKP0 and BKP1 pins. It also supports software to trigger the brake function. Brake function is controlled by the contents of the PWM_BRKCTL registers.

Since both the brake conditions being asserted will automatically raise the BRKSTS flag, user program can poll these brake flag bits or enable PWM's brake interrupt to determine which condition causes a brake to occur.

Note: When a brake happens, the PWM0_CH0 ~ PWM0_CH5 enable bits will be disabled by hardware. User program must clear fault brake event flag (BRKSTS) first, then write the PWM0_CHn enable bits again to release PWM0_CHn Brake state.

6.7.5.15 PWM Phase Change Function

The phase change function can be used to trigger PWM by TIMER module with ACMP by enabled TMR0TEN and TMR1TEN (PWM_PHCHG[30] and PWM_PHCHG[22]) registers. To use Timer trigger PWM, by configuring both PWM_PHCHG and PWM_PHCHGNXT register. Each time when time-out event coming, PWM_PHCHG's value will be updated by

PWM_PHCHG_NXT's value automatically, PWM_PHCHG's bit field is identical with PWM_PHCHGNXT's, each time when PWM_PHCHG updated, the related function will also change, It includes enable ACMP trigger PWM enable, ACMP0TEN and ACMP1TEN (PWM_PHCHG[31] and PWM_PHCHG[23]).

There are two steps to select source of ACMP. First, set POSCTL[1:0] (PWM_PHCHGMSK[9:8]) to select ACMP input control source. If control source is PWM_PHCHG register (POSCTL[1:0] = 11), then setting A0POSSEL (PWM_PHCHG[29:28]) and A1POSSEL (PWM_PHCHG[21:20]) to select input source of ACMP0 and ACMP1. As long as ACMP0 or ACMP1 trigger PWM, setting AOFFENn0 (PWM_PHCHG[27:24]) corresponding to ACMP0 or AOFFENn1 (PWM_PHCHG[19:16]) corresponding to ACMP1 will force PWM0_CHn (n denotes PWM channel 0~3) to output low lasting for a period cycle; it is useful in step motor application.

Besides trigger PWM, phase change register also with mask control bits to change the phase of PWM output. By setting 0 to corresponding channel's MSKENn (PWM_PHCHG[13:8]) to enable channel's mask function, then corresponding channel will output level of MSKDATn (PWM_PHCHG[5:0]). MSKDAT7 and MSKDAT6 (PWM_PHCHG[7:6]) are independent output data to GPIO P0.0 and P0.1 when setting 1 to MASKEND7 and MASKEND6 (PWM_PHCHGMSK[7:6]).

6.7.5.16 PWM Phase Change Mask Aligned Function

The PWM supports phase change mask aligned function to aligned mask data to each PWM period. By setting corresponding channel's ALIGNn (PWM_MSKALIGN[21:16]), PWM channel n will not change output level to MSKDATn (PWM_MSKALIGN[5:0]) immediately, it will changes until next start of PWM period. Please to note that, PWM_MSKALIGN and PWM_PHCHG registers MSKENn and MSKDATn both setting to the same registers.

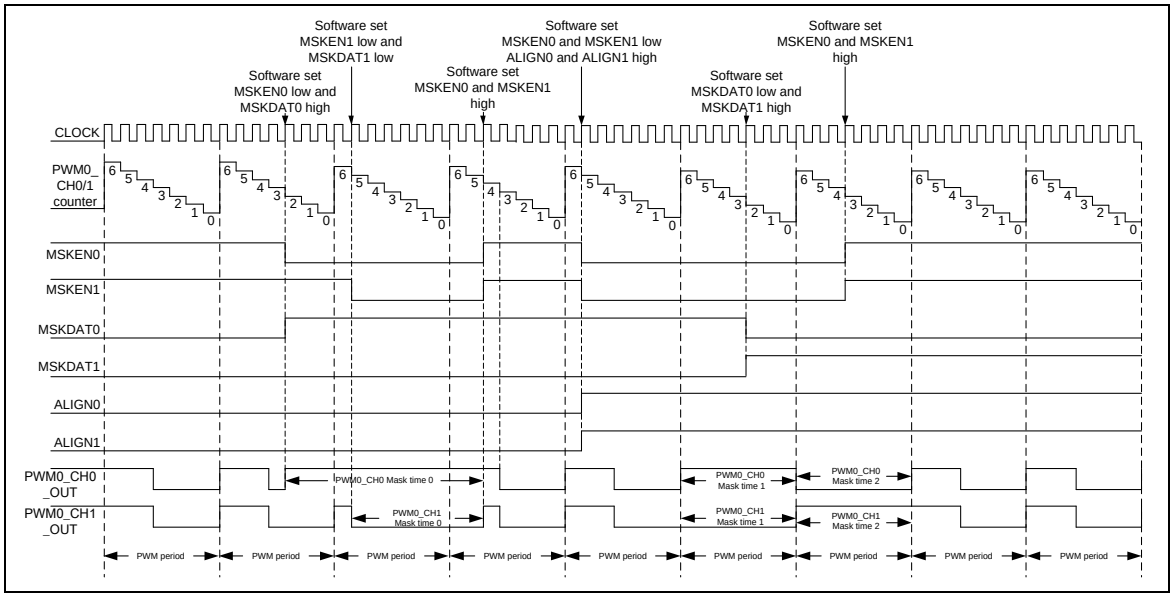


Figure 6.7-25 Mask and Mask-aligned Function

6.7.5.17 PWM Counter Start Procedure

The following procedure is recommended for PWM counter start

1. Configure prescaler register (PWM_CLKPSC) for setting clock prescaler (CLKPSCnm).
2. Configure clock select register (PWM_CLKDIV) for setting clock source select (CLKDIVn).

3. Configure PWM control register (PWM_CTL) for setting auto-reload mode (CNTMODEn = 1), PWM counter aligned type (CNTTYPE) and DISABLE PWM counter (CNTENn = 0).
4. Configure PWM control register (PWM_CTL) for setting inverter on/off (PINVn), and Dead-time generator on/off (DTCNTnm). (Optional)
5. Configure PWM_DTCTL register to set dead-time interval. (Optional)
6. Configure comparator register (PWM_CMPDATn) for setting PWM duty (CMPn).
7. Configure PWM counter register (PWM_PERIODn) for setting PWM counter loaded value (PERIODn).
8. Configure PWM interrupt enable register (PWM_INTEN) for setting PWM period interrupt type (INTTYPE), PWM zero interrupt enable bit (ZIENn), PWM compare up match interrupt enable bit (CMPUIENn), PWM period interrupt enable bit (PIENn), PWM compare down match interrupt enable bit (CMPDIENn). (Optional)
9. Configure PWM output enable register (PWM_POEN) to enable PWM output channel
10. Configure PWM control register (PWM_CTL) to enable PWM counter (CNTENn = 1)

6.7.5.18 PWM Counter Stop Procedure

Method 1:

Set 16-bit counter register (PERIODn) to 0. When interrupt request happened, disable PWM counter (CNTENn in PWM_CTL). (Recommended)

Method 2:

Disable PWM Counter directly (CNTENn in PWM_CTL) (Not recommended)

The reason why this method is not recommended is that disabling CNTENn will immediately stop PWM output signal and lead to change the duty of the PWM output, this may cause damage to the motor control circuit.

6.7.6 Register Map

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

Register	Offset	R/W	Description	Reset Value
PWM0 Base Address: **PWM0_BA = 0x4004_0000**				
PWM_CLKPSC	PWM0_BA+0x00	R/W	PWM Clock Pre-scale Register	0x0000_0000
PWM_CLKDIV	PWM0_BA+0x04	R/W	PWM Clock Select Register	0x0000_0000
PWM_CTL	PWM0_BA+0x08	R/W	PWM Control Register	0x0000_0000
PWM_PERIOD0	PWM0_BA+0x0C	R/W	PWM Counter Period Register 0	0x0000_0000
PWM_PERIOD1	PWM0_BA+0x10	R/W	PWM Counter Period Register 1	0x0000_0000
PWM_PERIOD2	PWM0_BA+0x14	R/W	PWM Counter Period Register 2	0x0000_0000
PWM_PERIOD3	PWM0_BA+0x18	R/W	PWM Counter Period Register 3	0x0000_0000
PWM_PERIOD4	PWM0_BA+0x1C	R/W	PWM Counter Period Register 4	0x0000_0000
PWM_PERIOD5	PWM0_BA+0x20	R/W	PWM Counter Period Register 5	0x0000_0000
PWM_CMPDAT0	PWM0_BA+0x24	R/W	PWM Comparator Register 0	0x0000_0000
PWM_CMPDAT1	PWM0_BA+0x28	R/W	PWM Comparator Register 1	0x0000_0000
PWM_CMPDAT2	PWM0_BA+0x2C	R/W	PWM Comparator Register 2	0x0000_0000
PWM_CMPDAT3	PWM0_BA+0x30	R/W	PWM Comparator Register 3	0x0000_0000
PWM_CMPDAT4	PWM0_BA+0x34	R/W	PWM Comparator Register 4	0x0000_0000
PWM_CMPDAT5	PWM0_BA+0x38	R/W	PWM Comparator Register 5	0x0000_0000
PWM_INTEN	PWM0_BA+0x54	R/W	PWM Interrupt Enable Register	0x0000_0000
PWM_INTSTS	PWM0_BA+0x58	R/W	PWM Interrupt Status Register	0x0000_0000
PWM_POEN	PWM0_BA+0x5C	R/W	PWM Output Enable Register	0x0000_0000
PWM_BRKCTL	PWM0_BA+0x60	R/W	PWM Fault Brake Control Register	0x0000_0000
PWM_DTCTL	PWM0_BA+0x64	R/W	PWM Dead-time Control Register	0x0000_0000
PWM_ADCTCTL0	PWM0_BA+0x68	R/W	PWM Trigger Control Register 0	0x0000_0000
PWM_ADCTCTL1	PWM0_BA+0x6C	R/W	PWM Trigger Control Register 1	0x0000_0000

PWM_ADCTSTS0	PWM0_BA+0x70	R/W	PWM Trigger Status Register 0	0x0000_0000
PWM_ADCTSTS1	PWM0_BA+0x74	R/W	PWM Trigger Status Register 1	0x0000_0000
PWM_PHCHG	PWM0_BA+0x78	R/W	PWM Phase Changed Register	0x0000_3F00
PWM_PHCHGNXT	PWM0_BA+0x7C	R/W	PWM Next Phase Change Register	0x0000_3F00
PWM_PHCHGMSK	PWM0_BA+0x80	R/W	PWM Phase Change Mask Register	0x0000_0000
PWM_IFA	PWM0_BA+0x84	R/W	PWM Period Interrupt Accumulation Control Register	0x0000_00F0
PWM_PCACTL	PWM0_BA+0x88	R/W	PWM Precise Center-Aligned Type Control Register	0x0000_0000
PWM_MSKALIGN	PWM0_BA+0x8C	R/W	PWM Phase Change Mask Aligned Register	0x0000_3F00

6.7.7 Register Description

PWM Pre-Scale Register (PWM_CLKPSC)

Register	Offset	R/W	Description	Reset Value
PWM_CLKPSC	PWM0_BA+0x00	R/W	PWM Clock Pre-scale Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
CLKPSC45							
15	14	13	12	11	10	9	8
CLKPSC23							
7	6	5	4	3	2	1	0
CLKPSC01							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:16]	**CLKPSC45**	**Clock Prescaler 4 For PWM Counter 4 And 5** Clock input is divided by (CLKPSC45 + 1) before it is fed to the corresponding PWM counter. If CLKPSC45 = 0, the clock prescaler 4 output clock will be stopped. So the corresponding PWM counter will also be stopped.
[15:8]	**CLKPSC23**	**Clock Prescaler 2 For PWM Counter 2 And 3** Clock input is divided by (CLKPSC23 + 1) before it is fed to the corresponding PWM counter. If CLKPSC23 = 0, the clock prescaler 2 output clock will be stopped. So the corresponding PWM counter will also be stopped.
[7:0]	**CLKPSC01**	**Clock Prescaler 0 For PWM Counter 0 And 1** Clock input is divided by (CLKPSC01 + 1) before it is fed to the corresponding PWM counter. If CLKPSC01 = 0, the clock prescaler 0 output clock will be stopped. So the corresponding PWM counter will also be stopped.

PWM Clock Selector Register (PWM_CLKDIV)

Register	Offset	R/W	Description	Reset Value
PWM_CLKDIV	PWM0_BA+0x04	R/W	PWM Clock Select Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved	**CLKDIV5**			**Reserved**	**CLKDIV4**		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	**CLKDIV3**			**Reserved**	**CLKDIV2**		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**CLKDIV1**			**Reserved**	**CLKDIV0**		

Bits	Description	
[31:23]	**Reserved**	Reserved.
[22:20]	**CLKDIV5**	**Counter 5 Clock Divider Selection** Select clock input for PWM counter. 000 = Clock input / (CLKPSC45/2). 001 = Clock input / (CLKPSC45/4). 010 = Clock input / (CLKPSC45/8). 011 = Clock input / (CLKPSC45/16). 100 = Clock input / CLKPSC45. Others = Clock input.
[19]	**Reserved**	Reserved.
[18:16]	**CLKDIV4**	**Counter 4 Clock Divider Selection** Select clock input for PWM counter. 000 = Clock input / (CLKPSC45/2). 001 = Clock input / (CLKPSC45/4). 010 = Clock input / (CLKPSC45/8). 011 = Clock input / (CLKPSC45/16). 100 = Clock input / CLKPSC45. Others = Clock input.
[15]	**Reserved**	Reserved.

Bits	Description	
[14:12]	**CLKDIV3**	**Counter 3 Clock Divider Selection** Select clock input for PWM counter. 000 = Clock input / (CLKPSC23/2). 001 = Clock input / (CLKPSC23/4). 010 = Clock input / (CLKPSC23/8). 011 = Clock input / (CLKPSC23/16). 100 = Clock input / CLKPSC23. Others = Clock input.
[11]	**Reserved**	Reserved.
[10:8]	**CLKDIV2**	**Counter 2 Clock Divider Selection** Select clock input for PWM counter. 000 = Clock input / (CLKPSC23/2). 001 = Clock input / (CLKPSC23/4). 010 = Clock input / (CLKPSC23/8). 011 = Clock input / (CLKPSC23/16). 100 = Clock input / CLKPSC23. Others = Clock input.
[7]	**Reserved**	Reserved.
[6:4]	**CLKDIV1**	**Counter 1 Clock Divider Selection** Select clock input for PWM counter. 000 = Clock input / (CLKPSC01/2). 001 = Clock input / (CLKPSC01/4). 010 = Clock input / (CLKPSC01/8). 011 = Clock input / (CLKPSC01/16). 100 = Clock input / CLKPSC01. Others = Clock input.
[3]	**Reserved**	Reserved.
[2:0]	**CLKDIV0**	**Counter 0 Clock Divider Selection** Select clock input for PWM counter. 000 = Clock input / (CLKPSC01/2). 001 = Clock input / (CLKPSC01/4). 010 = Clock input / (CLKPSC01/8). 011 = Clock input / (CLKPSC01/16). 100 = Clock input / CLKPSC01. Others = Clock input.

PWM Control Register (PWM_CTL)

Register	Offset	R/W	Description	Reset Value
PWM_CTL	PWM0_BA+0x08	R/W	PWM Control Register	0x0000_0000

31	30	29	28	27	26	25	24
CNTTYPE	GROUPEN	MODE		CNTCLR	DTCNT45	DTCNT23	DTCNT01
23	22	21	20	19	18	17	16
CNTMODE5	PINV5	ASYMEN	CNTEN5	CNTMODE4	PINV4	Reserved	CNTEN4
15	14	13	12	11	10	9	8
CNTMODE3	PINV3	Reserved	CNTEN3	CNTMODE2	PINV2	Reserved	CNTEN2
7	6	5	4	3	2	1	0
CNTMODE1	PINV1	HCUPDT	CNTEN1	CNTMODE0	PINV0	DBGTRIOFF	CNTEN0

Bits	Description	
[31]	**CNTTYPE**	**PWM Counter-aligned Type Select Bit** 0 = Edge-aligned type. 1 = Center-aligned type.
[30]	**GROUPEN**	**Group Function Enable Bit** 0 = The signals timing of all PWM channels are independent. 1 = Unify the signals timing of PWM0_CH0, PWM0_CH2 and PWM0_CH4 in the same phase which is controlled by PWM0_CH0 and also unify the signals timing of PWM0_CH1, PWM0_CH3 and PWM0_CH5 in the same phase which is controlled by PWM0_CH1.
[29:28]	**MODE**	**PWM Operating Mode Select Bit** 00 = Independent mode. 01 = Complementary mode. 10 = Synchronized mode. 11 = Reserved.
[27]	**CNTCLR**	**Clear PWM Counter Control Bit** 0 = Do not clear PWM counter. 1 = All 16-bit PWM counters cleared to 0x0000. **Note:** It is automatically cleared by hardware.
[26]	**DTCNT45**	**Dead-time 4 Counter Enable Bit (PWM0_CH4 And PWM0_CH5 Pair For PWMC Group)** 0 = Dead-time 4 generator Disabled. 1 = Dead-time 4 generator Enabled. **Note:** When the dead-time generator is enabled, the pair of PWM0_CH4 and PWM0_CH5 becomes a complementary pair for PWMC group.
[25]	**DTCNT23**	**Dead-time 2 Counter Enable Bit (PWM0_CH2 And PWM0_CH3 Pair For PWMB Group)** 0 = Dead-time 2 generator Disabled. 1 = Dead-time 2 generator Enabled. **Note:** When the dead-time generator is enabled, the pair of PWM0_CH2 and PWM0_CH3 becomes a complementary pair for PWMB group.

Bits	Description	
[24]	**DTCNT01**	**Dead-time 0 Counter Enable Bit (PWM0_CH0 And PWM0_CH1 Pair For PWMA Group)** 0 = Dead-time 0 generator Disabled. 1 = Dead-time 0 generator Enabled. **Note:** When the dead-time generator is enabled, the pair of PWM0_CH0 and PWM0_CH1 becomes a complementary pair for PWMA group.
[23]	**CNTMODE5**	**PWM Counter 5 Auto-reload/One-shot Mode** 0 = One-shot mode. 1 = Auto-reload mode. **Note:** If there is a rising transition at this bit, it will cause PERIOD5 and CMP5 cleared.
[22]	**PINV5**	**PWM0_CH5 Output Inverter Enable Bit** 0 = PWM0_CH5 output inverter Disabled. 1 = PWM0_CH5 output inverter Enabled.
[21]	**ASYMEN**	**Asymmetric Mode In Center-aligned Type** 0 = Symmetric mode in center-aligned type. 1 = Asymmetric mode in center-aligned type.
[20]	**CNTEN5**	**PWM Counter 5 Enable Start Run** 0 = Corresponding PWM counter running Stopped. 1 = Corresponding PWM counter start run Enabled.
[19]	**CNTMODE4**	**PWM Counter 4 Auto-reload/One-shot Mode** 0 = One-shot mode. 1 = Auto-reload mode. **Note:** If there is a rising transition at this bit, it will cause PERIOD4 and CMP4 cleared.
[18]	**PINV4**	**PWM0_CH4 Output Inverter Enable Bit** 0 = PWM0_CH4 output inverter Disabled. 1 = PWM0_CH4 output inverter Enabled.
[17]	**Reserved**	Reserved.
[16]	**CNTEN4**	**PWM Counter 4 Enable Start Run** 0 = Corresponding PWM counter running Stopped. 1 = Corresponding PWM counter start run Enabled.
[15]	**CNTMODE3**	**PWM Counter 3 Auto-reload/One-shot Mode** 0 = One-shot mode. 1 = Auto-reload mode. **Note:** If there is a rising transition at this bit, it will cause PERIOD3 and CMP3 cleared.
[14]	**PINV3**	**PWM0_CH 3 Output Inverter Enable Bit** 0 = PWM0_CH3 output inverter Disabled. 1 = PWM0_CH3 output inverter Enabled.
[13]	**Reserved**	Reserved.
[12]	**CNTEN3**	**PWM Counter 3 Enable Start Run** 0 = Corresponding PWM counter running Stopped. 1 = Corresponding PWM counter start run Enabled.

Bits	Description	
[11]	**CNTMODE2**	**PWM Counter 2 Auto-reload/One-shot Mode** 0 = One-shot mode. 1 = Auto-reload mode. **Note:** If there is a rising transition at this bit, it will cause PERIOD2 and CMP2 cleared.
[10]	**PINV2**	**PWM0_CH2 Output Inverter Enable Bit** 0 = PWM0_CH2 output inverter Disabled. 1 = PWM0_CH2 output inverter Enabled.
[9]	**Reserved**	Reserved.
[8]	**CNTEN2**	**PWM Counter 2 Enable Start Run** 0 = Corresponding PWM counter running Stopped. 1 = Corresponding PWM counter start run Enabled.
[7]	**CNTMODE1**	**PWM Counter 1 Auto-reload/One-shot Mode** 0 = One-shot mode. 1 = Auto-reload mode. **Note:** If there is a rising transition at this bit, it will cause PERIOD1 and CMP1 cleared.
[6]	**PINV1**	**PWM0_CH1 Output Inverter Enable Bit** 0 = PWM0_CH1 output inverter Disable. 1 = PWM0_CH1 output inverter Enable.
[5]	**HCUPDT**	**Half Cycle Update Enable for Center-aligned Type** 0 = Disable half cycle update PERIOD & CMP. 1 = Enable half cycle update PERIOD & CMP.
[4]	**CNTEN1**	**PWM Counter 1 Enable/Disable Start Run** 0 = Corresponding PWM counter running Stopped. 1 = Corresponding PWM counter start run Enabled.
[3]	**CNTMODE0**	**PWM Counter 0 Auto-reload/One-shot Mode** 0 = One-shot mode. 1 = Auto-reload mode. **Note:** If there is a rising transition at this bit, it will cause PERIOD0 and CMP0 cleared.
[2]	**PINV0**	**PWM0_CH0 Output Inverter Enable Bit** 0 = PWM0_CH0 output inverter Disabled. 1 = PWM0_CH0 output inverter Enabled.
[1]	**DBGTRIOFF**	**Disable PWM Output Tri-state Under Debug Mode (Available In DEBUG Mode Only)** 0 = Safe mode: The counter is frozen and PWM outputs are shut down Safe state for the inverter. The counter can still be re-started from where it stops. 1 = Normal mode: The counter continues to operate normally May be dangerous in some cases since a constant duty cycle is applied to the inverter (no more interrupts serviced).
[0]	**CNTEN0**	**PWM Counter 0 Enable Start Run** 0 = Corresponding PWM counter running Stopped. 1 = Corresponding PWM counter start run Enabled.

PWM Counter Register 0-5 (PWM_PERIOD0-5)

Register	Offset	R/W	Description	Reset Value
PWM_PERIOD0	PWM0_BA+0x0C	R/W	PWM Counter Period Register 0	0x0000_0000
PWM_PERIOD1	PWM0_BA+0x10	R/W	PWM Counter Period Register 1	0x0000_0000
PWM_PERIOD2	PWM0_BA+0x14	R/W	PWM Counter Period Register 2	0x0000_0000
PWM_PERIOD3	PWM0_BA+0x18	R/W	PWM Counter Period Register 3	0x0000_0000
PWM_PERIOD4	PWM0_BA+0x1C	R/W	PWM Counter Period Register 4	0x0000_0000
PWM_PERIOD5	PWM0_BA+0x20	R/W	PWM Counter 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
PERIODn							
7	6	5	4	3	2	1	0
PERIODn							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0] n=0,1..5	**PERIODn**	**PWM Counter Period Value** PERIODn determines the PWM counter period. Edge-aligned type: PWM frequency = HCLK/((prescale+1)*(clock divider))/(PERIODn+1); where xy, could be 01, 23, 45 depending on the selected PWM channel. Duty ratio = (CMPn+1)/(PERIODn+1). CMPn >= PERIODn: PWM output is always high. CMPn < PERIODn: PWM low width = (PERIODn-CMPn) unit; PWM high width = (CMP+1) unit. CMPn = 0: PWM low width = (PERIODn) unit; PWM high width = 1 unit. Center-aligned type: PWM frequency = HCLK/((prescale+1)*(clock divider))/ (2*PERIODn+1); where xy, could be 01, 23, 45 depending on the selected PWM channel. Duty ratio = (PERIODn - CMPn)/(PERIODn+1). CMPn >= PERIODn: PWM output is always low. CMPn < PERIODn: PWM low width = (CMPn + 1) * 2 unit; PWM high width = (PERIODn - CMPn) * 2 unit. CMPn = 0: PWM low width = 2 unit; PWM high width = (PERIODn) * 2 unit.

Bits	Description	
		(Unit = One PWM clock cycle). **Note:** Any write to PERIODn will take effect in the next PWM cycle.

PWM Comparator Register 0-5 (PWM_CMPDAT0-5)

Register	Offset	R/W	Description	Reset Value
PWM_CMPDAT0	PWM0_BA+0x24	R/W	PWM Comparator Register 0	0x0000_0000
PWM_CMPDAT1	PWM0_BA+0x28	R/W	PWM Comparator Register 1	0x0000_0000
PWM_CMPDAT2	PWM0_BA+0x2C	R/W	PWM Comparator Register 2	0x0000_0000
PWM_CMPDAT3	PWM0_BA+0x30	R/W	PWM Comparator Register 3	0x0000_0000
PWM_CMPDAT4	PWM0_BA+0x34	R/W	PWM Comparator Register 4	0x0000_0000
PWM_CMPDAT5	PWM0_BA+0x38	R/W	PWM Comparator Register 5	0x0000_0000

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

Bits	Description	
[31:16] n=0,1..5	**CMPDn**	**PWM Comparator Register For Down Counter In Asymmetric Mode** CMPn >= PERIODn: up counter PWM output is always low. CMPDn >= PERIODn: down counter PWM output is always low. Others: PWM output is always high.
[15:0] n=0,1..5	**CMPn**	**PWM Comparator Register** CMP determines the PWM duty. Edge-aligned type: PWM frequency = HCLK/((CLKPSCnm+1)*(clock divider))/(PERIODn+1); where nm, could be 01, 23, 45 depending on the selected PWM channel. Duty ratio = (CMPn+1)/(PERIODn+1). CMPn >= PERIODn: PWM output is always high. CMPn < PERIODn: PWM low width = (PERIODn-CMPn) unit; PWM high width = (CMP+1) unit. CMPn = 0: PWM low width = (PERIODn) unit; PWM high width = 1 unit. Center-aligned type: PWM frequency = HCLK/((CLKPSCnm+1)*(clock divider)) /(2*PERIODn+1); where nm, could be 01, 23, 45 depending on the selected PWM channel. Duty ratio = (PERIODn - CMPn)/(PERIODn+1).

Bits	Description	
		CMPn >= PERIODn: PWM output is always low. CMPn < PERIODn: PWM low width = (CMPn + 1) * 2 unit; PWM high width = (PERIODn - CMPn) * 2 unit. CMPn = 0: PWM low width = 2 unit; PWM high width = (PERIODn) * 2 unit. (Unit = One PWM clock cycle). **Note:** Any write to CMPn will take effect in the next PWM cycle.

PWM Interrupt Enable Register (PWM_INTEN)

Register	Offset	R/W	Description	Reset Value
PWM_INTEN	PWM0_BA+0x54	R/W	PWM Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		CMPUIEN5	CMPUIEN4	CMPUIEN3	CMPUIEN2	CMPUIEN1	CMPUIEN0
23	22	21	20	19	18	17	16
PIEN5	PIEN4	PIEN3	PIEN2	PIEN1	PIEN0	PINTTYPE	BRKIEN
15	14	13	12	11	10	9	8
Reserved		CMPDIEN5	CMPDIEN4	CMPDIEN3	CMPDIEN2	CMPDIEN1	CMPDIEN0
7	6	5	4	3	2	1	0
Reserved		ZIEN5	ZIEN4	ZIEN3	ZIEN2	ZIEN1	ZIEN0

Bits	Description	
[31:30]	**Reserved**	Reserved.
[29:24] n=0,1..5	**CMPUIENn**	**PWM Compare Up Interrupt Enable Bit** 0 = PWM0_CHn compare up interrupt Disabled. 1 = PWM0_CHn compare up interrupt Enabled.
[23:18] n=0,1..5	**PIENn**	**PWM Period Interrupt Enable Bit** 0 = PWM0_CHn period interrupt Disabled. 1 = PWM0_CHn period interrupt Enabled.
[17]	**PINTTYPE**	**PWM Interrupt Type Selection** 0 = ZIFn will be set if PWM counter underflows. 1 = ZIFn will be set if PWM counter matches PERIODn register. **Note:** This bit is effective when PWM is in center-aligned type only.
[16]	**BRKIEN**	**Fault Brake0 And Fault Brake1 Interrupt Enable Bit** 0 = BRKIF0 and BRKIF1 trigger PWM interrupt Disabled. 1 = BRKIF0 and BRKIF1 trigger PWM interrupt Enabled.
[15:14]	**Reserved**	Reserved.
[13:8] n=0,1..5	**CMPDIENn**	**PWM Compare Down Interrupt Enable Bit** 0 = PWM0_CHn compare down interrupt Disabled. 1 = PWM0_CHn compare down interrupt Enabled.
[7:6]	**Reserved**	Reserved.
[5:0] n=0,1..5	**ZIENn**	**PWM Zero Point Interrupt Enable Bit** 0 = PWM0_CHn zero point interrupt Disabled. 1 = PWM0_CHn zero point interrupt Enabled.

PWM Interrupt Indication Register (PWM_INTSTS)

Register	Offset	R/W	Description	Reset Value
PWM_INTSTS	PWM0_BA+0x58	R/W	PWM Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		CMPUIF5	CMPUIF4	CMPUIF3	CMPUIF2	CMPUIF1	CMPUIF0
23	22	21	20	19	18	17	16
PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	BRKIF1	BRKIF0
15	14	13	12	11	10	9	8
Reserved		CMPDIF5	CMPDIF4	CMPDIF3	CMPDIF2	CMPDIF1	CMPDIF0
7	6	5	4	3	2	1	0
Reserved		ZIF5	ZIF4	ZIF3	ZIF2	ZIF1	ZIF0

Bits	Description	
[31:30]	**Reserved**	Reserved.
[29:24] n=0,1..5	**CMPUIFn**	**PWM Compare Up Interrupt Flag** Flag is set by hardware when PWM0_CHn counter up count reaches CMPn. **Note:** This bit can be cleared by software writing 1.
[23:18] n=0,1..5	**PIFn**	**PWM Period Interrupt Flag** Flag is set by hardware when PWM0_CHn counter reaches PERIODn. **Note:** This bit can be cleared by software writing 1.
[17]	**BRKIF1**	**PWM Brake1 Flag** 0 = PWM Brake does not recognize a falling signal at BKP1. 1 = When PWM Brake detects a falling signal at pin BKP1; this flag will be set to high. **Note:** This bit can be cleared by software writing 1.
[16]	**BRKIF0**	**PWM Brake0 Flag** 0 = PWM Brake does not recognize a falling signal at BKP0. 1 = When PWM Brake detects a falling signal at pin BKP0; this flag will be set to high. **Note:** This bit can be cleared by software writing 1.
[15:14]	**Reserved**	Reserved.
[13:8] n=0,1..5	**CMPDIFn**	**PWM Compare Down Interrupt Flag** Flag is set by hardware when PWMn counter downcount reaches CMPn. **Note:** This bit can be cleared by software writing 1.
[7:6]	**Reserved**	Reserved.
[5:0] n=0,1..5	**ZIFn**	**PWM Zero Point Interrupt Flag** Flag is set by hardware when PWMn counter downcount reaches zero point. **Note:** This bit can be cleared by software writing 1.

Note: User can clear each interrupt flag by writing 1 to the corresponding bit in PWM_INTSTS.

PWM Output Control Register (PWM_POEN)

Register	Offset	R/W	Description	Reset Value
PWM_POEN	PWM0_BA+0x5C	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.
[5:0] n=0,1..5	**POENn**	**PWM Output Enable Bits** 0 = PWM channel n output to pin Disabled. 1 = PWM channel n output to pin Enabled. **Note:** The corresponding GPIO pin must be switched to PWM function.

PWM Fault Brake Control Register (PWM_BRKCTL)

Register	Offset	R/W	Description	Reset Value
PWM_BRKCTL	PWM0_BA+0x60	R/W	PWM Fault Brake Control Register	0x0000_0000

31	30	29	28	27	26	25	24
D7BKOD	D6BKOD	BKODD5	BKOD4	BKOD3	BKOD2	BKOD1	BKOD0
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved						SWBRK	BRKACT
7	6	5	4	3	2	1	0
BRKSTS	Reserved			BRK1SEL	BRK0SEL	BRK1EN	BRK0EN

Bits	Description	
[31]	**D7BKOD**	**Channel 7 Brake Output Data Select Bit** 0 = Channel 7 output low when fault brake conditions asserted. 1 = Channel 7 output high when fault brake conditions asserted.
[30]	**D6BKOD**	**Channel 6 Brake Output Data Select Bit** 0 = Channel 6 output low when fault brake conditions asserted. 1 = Channel 6 output high when fault brake conditions asserted.
[29:24] n=0,1..5	**BKODn**	**PWM Brake Output Data Select Bits** 0 = PWM channel n output low when fault brake conditions asserted. 1 = PWM channel n output high when fault brake conditions asserted.
[23:10]	**Reserved**	Reserved.
[9]	**SWBRK**	**Software Brake** 0 = Disable PWM Software brake and back to normal PWM function. 1 = Assert PWM Brake immediately.
[8]	**BRKACT**	**PWM Brake Action Type** 0 = PWM counter stop when brake is asserted. 1 = PWM counter keep going when brake is asserted.
[7]	**BRKSTS**	**PWM Fault Brake Event Status Flag** 0 = PWM output initial state when fault brake conditions asserted. 1 = PWM output fault brake state when fault brake conditions asserted. **Note:** This bit can be cleared by software writing 1 and must be cleared before restarting the PWM counter.
[6:4]	**Reserved**	Reserved.
[3]	**BRK1SEL**	**BKP0 Fault Brake Function Source Select Bit** 0 = EINT0 as one brake source in BKP0. 1 = CPO1 as one brake source in BKP0.

Bits	Description	
[2]	**BRK0SEL**	**BKP1 Fault Brake Function Source Select Bit** 0 = EINT1 as one brake source in BKP1. 1 = CPO0 as one brake source in BKP1.
[1]	**BRK1EN**	**Enable BKP1 Pin Trigger Fault Brake Function 1** 0 = Disabling BKP1 pin can trigger brake function 1 (EINT1 or CPO0). 1 = Enabling a falling at BKP1 pin can trigger brake function 1.
[0]	**BRK0EN**	**Enable BKP0 Pin Trigger Fault Brake Function 0** 0 = Disabling BKP0 pin can trigger brake function 0 (EINT0 or CPO1). 1 = Enabling a falling at BKP0 pin can trigger brake function 0.

PWM Dead-time Interval Register (PWM_DTCTL)

Register	Offset	R/W	Description	Reset Value
PWM_DTCTL	PWM0_BA+0x64	R/W	PWM Dead-time Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DTI45							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DTI23							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DTI01							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:16]	**DTI45**	**Dead-time Interval Register For Pair Of Channel4 And Channel5 (PWM0_CH4 And PWM0_CH5 Pair)** These 8 bits determine dead-time length. The unit time of dead-time length is received from corresponding PWM_CLKDIV bits.
[15:8]	**DTI23**	**Dead-time Interval Register For Pair Of Channel2 And Channel3 (PWM0_CH2 And PWM0_CH3 Pair)** These 8 bits determine dead-time length. The unit time of dead-time length is received from corresponding PWM_CLKDIV bits.
[7:0]	**DTI01**	**Dead-time Interval Register For Pair Of Channel0 And Channel1 (PWM0_CH0 And PWM0_CH1 Pair)** These 8 bits determine dead-time length. The unit time of dead-time length is received from corresponding PWM_CLKDIV bits.

PWM Trigger ADC Control Register (PWM_ADCTCTL0)

Register	Offset	R/W	Description	Reset Value
PWM_ADCTCTL0	PWM0_BA+0x68	R/W	PWM Trigger Control Register 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				ZPTRGEN3	CDTRGEN3	CPTRGEN3	CUTRGEN3
23	22	21	20	19	18	17	16
Reserved				ZPTRGEN2	CDTRGEN2	CPTRGEN2	CUTRGEN2
15	14	13	12	11	10	9	8
Reserved				ZPTRGEN1	CDTRGEN1	CPTRGEN1	CUTRGEN1
7	6	5	4	3	2	1	0
Reserved				ZPTRGEN0	CDTRGEN0	CPTRGEN0	CUTRGEN0

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27]	**ZPTRGEN3**	**Channel 3 Zero Point Trigger ADC Enable Bit** Enable PWM trigger ADC function while channel3's counter matching 0 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note:** This bit is valid for both center-aligned type and edged aligned type.
[26]	**CDTRGEN3**	**Channel 3 Compare Down Trigger ADC Enable Bit** Enable PWM trigger ADC function while channel3's counter matching CMP3 in down-count direction 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note:** This bit is valid for both center-aligned type and edged aligned type.
[25]	**CPTRGEN3**	**Channel 3 Center Point Trigger ADC Enable Bit** Enable PWM trigger ADC function while channel3's counter matching PERIOD3 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note:** This bit is only valid for PWM in center-aligned type. When PWM is in edged aligned type, setting this bit is meaningless and will not take any effect.
[24]	**CUTRGEN3**	**Channel 3 Compare Up Trigger ADC Enable Bit** Enable PWM trigger ADC function while channel3's counter matching CMP3 in up-count direction 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note:** This bit is only valid for PWM in center-aligned type. When PWM is in edged aligned type, setting this bit is meaningless and will not take any effect.
[23:20]	**Reserved**	Reserved.

Bits	Description	
[19]	**ZPTRGEN2**	**Channel 2 Zero Point Trigger ADC Enable Bit** Enable PWM trigger ADC function while channel2's counter matching 0 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note:** This bit is valid for both center-aligned type and edged-aligned type.
[18]	**CDTRGEN2**	**Channel 2 Compare Down Trigger ADC Enable Bit** Enable PWM trigger ADC function while channel2's counter matching CMP2 in down-count direction 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note:** This bit is valid for both center-aligned type and edged-aligned type.
[17]	**CPTRGEN2**	**Channel 2 Center Point Trigger ADC Enable Bit** Enable PWM trigger ADC function while channel2's counter matching PERIOD2 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note:** This bit is only valid for PWM in center-aligned type. When PWM is in edged-aligned type, setting this bit is meaningless and will not take any effect.
[16]	**CUTRGEN2**	**Channel 2 Compare Up Trigger ADC Enable Bit** Enable PWM trigger ADC function while channel2's counter matching CMP2 in up-count direction 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note:** This bit is only valid for PWM in center-aligned type. When PWM is in edged-aligned type, setting this bit is meaningless and will not take any effect.
[15:12]	**Reserved**	Reserved.
[11]	**ZPTRGEN1**	**Channel 1 Zero Point Trigger ADC Enable Bit** Enable PWM trigger ADC function While channel1's Counter Matching 0 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note:** This bit is valid for both center-aligned type and edged-aligned type.
[10]	**CDTRGEN1**	**Channel 1 Compare Down Trigger ADC Enable Bit** Enable PWM trigger ADC function while channel1's counter matching CMP1 in down-count direction 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note:** This bit is valid for both center-aligned type and edged-aligned type.
[9]	**CPTRGEN1**	**Channel 1 Center Point Trigger ADC Enable Bit** Enable PWM trigger ADC function while channel1's counter matching PERIOD1 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note:** This bit is only valid for PWM in center-aligned type. When PWM is in edged-aligned type, setting this bit is meaningless and will not take any effect.

Bits	Description	
[8]	**CUTRGEN1**	**Channel 1 Compare Up Trigger ADC Enable Bit** Enable PWM trigger ADC function while channel1's counter matching CMP1 in up-count direction 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note:** This bit is only valid for PWM in center-aligned type. When PWM is in edged-aligned type, setting this bit is meaningless and will not take any effect.
[7:4]	**Reserved**	Reserved.
[3]	**ZPTRGEN0**	**Channel 0 Zero Point Trigger ADC Enable Bit** Enable PWM trigger ADC function while channel0's counter matching 0 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note:** This bit is valid for both center-aligned type and edged-aligned type.
[2]	**CDTRGEN0**	**Channel 0 Compare Down Trigger ADC Enable Bit** Enable PWM trigger ADC function while channel0's counter matching CMP0 in down-count direction 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note:** This bit is valid for both center-aligned type and edged-aligned type.
[1]	**CPTRGEN0**	**Channel 0 Center Point Trigger ADC Enable Bit** Enable PWM Trigger ADC Function While channel0's Counter Matching PERIOD0 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note:** This bit is only valid for PWM in center-aligned type. When PWM is in edged-aligned type, setting this bit is meaningless and will not take any effect.
[0]	**CUTRGEN0**	**Channel 0 Compare Up Trigger ADC Enable Bit** Enable PWM trigger ADC function while channel0's counter matching CMP0 in up-count direction 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note:** This bit is only valid for PWM in center-aligned type. When PWM is in edged-aligned type, setting this bit is meaningless and will not take any effect.

PWM Trigger ADC Control Register (PWM_ADCTCTL1)

Register	Offset	R/W	Description	Reset Value
PWM_ADCTCTL1	PWM0_BA+0x6C	R/W	PWM Trigger 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				ZPTRGEN5	CDTRGEN5	CPTRGEN5	CUTRGEN5
7	6	5	4	3	2	1	0
Reserved				ZPTRGEN4	CDTRGEN4	CPTRGEN4	CUTRGEN4

Bits	Description	
[31:12]	**Reserved**	Reserved.
[11]	**ZPTRGEN5**	**Channel 5 Zero Point Trigger ADC Enable Bit** Enable PWM trigger ADC function while channel5's counter matching 0 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note:** This bit is valid for both center-aligned type and edged-aligned type.
[10]	**CDTRGEN5**	**Channel 5 Compare Down Trigger ADC Enable Bit** Enable PWM trigger ADC function while channel5's counter matching CMP5 in down-count direction 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note:** This bit is valid for both center-aligned type and edged-aligned type.
[9]	**CPTRGEN5**	**Channel 5 Center Point Trigger ADC Enable Bit** Enable PWM trigger ADC function while channel5's counter matching PERIOD5 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note:** This bit is only valid for PWM in center-aligned type. When PWM is in edged-aligned type, setting this bit is meaningless and will not take any effect.
[8]	**CUTRGEN5**	**Channel 5 Compare Up Trigger ADC Enable Bit** Enable PWM trigger ADC function while channel5's counter matching CMP5 in up-count direction 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note:** This bit is only valid for PWM in center-aligned type. When PWM is in edged-aligned type, setting this bit is meaningless and will not take any effect.
[7:4]	**Reserved**	Reserved.

Bits	Description	
[3]	**ZPTRGEN4**	**Channel 4 Zero Point Trigger ADC Enable Bit** Enable PWM trigger ADC function while channel4's counter matching 0 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note:** This bit is valid for both center-aligned type and edged-aligned type.
[2]	**CDTRGEN4**	**Channel 4 Compare Down Trigger ADC Enable Bit** Enable PWM trigger ADC function while channel4's counter matching CMP4 in down-count direction 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note:** This bit is valid for both center-aligned type and edged-aligned type.
[1]	**CPTRGEN4**	**Channel 4 Center Point Trigger ADC Enable Bit** Enable PWM trigger ADC function while channel4's counter matching PERIOD4 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note:** This bit is only valid for PWM in center-aligned type. When PWM is in edged-aligned type, setting this bit is meaningless and will not take any effect.
[0]	**CUTRGEN4**	**Channel 4 Compare Up Trigger ADC Enable Bit** Enable PWM trigger ADC function while channel4's counter matching CMP4 in up-count direction 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note:** This bit is only valid for PWM in center-aligned type. When PWM is in edged-aligned type, setting this bit is meaningless and will not take any effect.

PWM Trigger Status Register (PWM_ADCTSTS0)

Register	Offset	R/W	Description	Reset Value
PWM_ADCTSTS0	PWM0_BA+0x70	R/W	PWM Trigger Status Register 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				ZPTRGF3	CDTRGF3	CPTRGF3	CUTRGF3
23	22	21	20	19	18	17	16
Reserved				ZPTRGF2	CDTRGF2	CPTRGF2	CUTRGF2
15	14	13	12	11	10	9	8
Reserved				ZPTRGF1	CDTRGF1	CPTRGF1	CUTRGF1
7	6	5	4	3	2	1	0
Reserved				ZPTRGF0	CDTRGF0	CPTRGF0	CUTRGF0

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27]	**ZPTRGF3**	**Channel 3 Zero Point Trigger ADC Flag** When the channel3's counter is counting to zero point, this bit will be set for trigger ADC. **Note:** This bit can be cleared by software writing 1.
[26]	**CDTRGF3**	**Channel 3 Compare Down Trigger ADC Flag** When the channel3's counter is counting down to CMP3, this bit will be set for trigger ADC. **Note:** This bit can be cleared by software writing 1.
[25]	**CPTRGF3**	**Channel 3 Center Point Trigger ADC Flag** When the channel3's counter is counting to PERIOD3, this bit will be set for trigger ADC. **Note:** This bit can be cleared by software writing 1.
[24]	**CUTRGF3**	**Channel 3 Compare Up Trigger ADC Flag** When the channel3's counter is counting up to CMP3, this bit will be set for trigger ADC. **Note:** This bit can be cleared by software writing 1.
[23:20]	**Reserved**	Reserved.
[19]	**ZPTRGF2**	**Channel 2 Zero Point Trigger ADC Enable Bit** When the channel2's counter is counting to zero point, this bit will be set for trigger ADC. **Note:** This bit can be cleared by software writing 1.
[18]	**CDTRGF2**	**Channel 2 Compare Down Trigger ADC Flag** When the channel2's counter is counting down to CMP2, this bit will be set for trigger ADC. **Note:** This bit can be cleared by software writing 1.
[17]	**CPTRGF2**	**Channel 2 Center Point Trigger ADC Flag** When the channel2's counter is counting to PERIOD2, this bit will be set for trigger ADC. **Note:** This bit can be cleared by software writing 1.

Bits	Description	
[16]	**CUTRGF2**	**Channel 2 Compare Up Trigger ADC Flag** When the channel2's counter is counting up to CMP2, this bit will be set for trigger ADC. **Note:** This bit can be cleared by software writing 1.
[15:12]	**Reserved**	Reserved.
[11]	**ZPTRGF1**	**Channel 1 Zero Point Trigger ADC Flag** When the channel1's counter is counting to zero point, this bit will be set for trigger ADC. **Note:** This bit can be cleared by software writing 1.
[10]	**CDTRGF1**	**Channel 1 Compare Down Trigger ADC Flag** When the channel1's counter is counting down to CMP1, this bit will be set for trigger ADC. **Note:** This bit can be cleared by software writing 1.
[9]	**CPTRGF1**	**Channel 1 Center Point Trigger ADC Flag** When the channel1's counter is counting to PERIOD1, this bit will be set for trigger ADC. **Note:** This bit can be cleared by software writing 1.
[8]	**CUTRGF1**	**Channel 1 Compare Up Trigger ADC Flag** When the channel1's counter is counting up to CMP1, this bit will be set for trigger ADC. **Note:** This bit can be cleared by software writing 1.
[7:4]	**Reserved**	Reserved.
[3]	**ZPTRGF0**	**Channel 0 Zero Point Trigger ADC Flag** When the channel0's counter is counting to zero point, this bit will be set for trigger ADC. **Note:** This bit can be cleared by software writing 1.
[2]	**CDTRGF0**	**Channel 0 Compare Down Trigger ADC Flag** When the channel0's counter is counting down to CMP0, this bit will be set for trigger ADC. **Note:** This bit can be cleared by software writing 1.
[1]	**CPTRGF0**	**Channel 0 Center Point Trigger ADC Flag** When the channel0's counter is counting to PERIOD0, this bit will be set for trigger ADC. **Note:** This bit can be cleared by software writing 1.
[0]	**CUTRGF0**	**Channel 0 Compare Up Trigger ADC Flag** When the channel0's counter is counting up to CMP0, this bit will be set for trigger ADC. **Note:** This bit can be cleared by software writing 1.

PWM Trigger Status Register (PWM_ADCTSTS1)

Register	Offset	R/W	Description	Reset Value
PWM_ADCTSTS1	PWM0_BA+0x74	R/W	PWM Trigger 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				ZPTRGF5	CDTRGF5	CPTRGF5	CUTRGF5
7	6	5	4	3	2	1	0
Reserved				ZPTRGF4	CDTRGF4	CPTRGF4	CUTRGF4

Bits	Description	
[31:12]	**Reserved**	Reserved.
[11]	**ZPTRGF5**	**Channel 5 Zero Point Trigger ADC Flag** When the channel5's counter is counting to zero point, this bit will be set for trigger ADC. **Note:** This bit can be cleared by software writing 1.
[10]	**CDTRGF5**	**Channel 5 Compare Down Trigger ADC Flag** When the channel5's counter is counting down to CMP5, this bit will be set for trigger ADC. **Note:** This bit can be cleared by software writing 1.
[9]	**CPTRGF5**	**Channel 5 Center Point Trigger ADC Flag** When the channel5's counter is counting to PERIOD5, this bit will be set for trigger ADC. **Note:** This bit can be cleared by software writing 1.
[8]	**CUTRGF5**	**Channel 5 Compare Up Trigger ADC Flag** When the channel5's counter is counting up to CMP5, this bit will be set for trigger ADC. **Note:** This bit can be cleared by software writing 1.
[7:4]	**Reserved**	Reserved.
[3]	**ZPTRGF4**	**Channel 4 Zero Point Trigger ADC Flag** When the channel4's counter is counting to zero point, this bit will be set for trigger ADC. **Note:** This bit can be cleared by software writing 1.
[2]	**CDTRGF4**	**Channel 4 Compare Down Trigger ADC Flag** When the channel4's counter is counting down to CMP4, this bit will be set for trigger ADC. **Note:** This bit can be cleared by software writing 1.
[1]	**CPTRGF4**	**Channel 4 Center Point Trigger ADC Flag** When the channel4's counter is counting to PERIOD4, this bit will be set for trigger ADC. **Note:** This bit can be cleared by software writing 1.

Bits	Description	
[0]	**CUTRGF4**	**Channel 4 Compare Up Trigger ADC Flag** When the channel4's counter is counting up to CMP4, this bit will be set for trigger ADC. **Note:** This bit can be cleared by software writing 1.

Phase Change Register (PWM_PHCHG)

Register	Offset	R/W	Description	Reset Value
PWM_PHCHG	PWM0_BA+0x78	R/W	PWM Phase Changed Register	0x0000_3F00

31	30	29	28	27	26	25	24
ACMP0TEN	TMR0TEN	A0POSSEL		AOFFEN30	AOFFEN20	AOFFEN10	AOFFEN00
23	22	21	20	19	18	17	16
ACMP1TEN	TMR1TEN	A1POSSEL		AOFFEN31	AOFFEN21	AOFFEN11	AOFFEN01
15	14	13	12	11	10	9	8
AUTOCLR1	AUTOCLR0	MSKEN5	MSKEN4	MSKEN3	MSKEN2	MSKEN1	MSKEN0
7	6	5	4	3	2	1	0
MSKDAT7	MSKDAT6	MSKDAT5	MSKDAT4	MSKDAT3	MSKDAT2	MSKDAT1	MSKDAT0

Bits	Description	
[31]	**ACMP0TEN**	**ACMP0 Trigger PWM Function Enable Bit** 0 = ACMP0 trigger PWM function Disabled. 1 = ACMP0 trigger PWM function Enabled. **Note:** This bit will be auto cleared when ACMP0 trigger PWM if AUTOCLR0 is set.
[30]	**TMR0TEN**	**TIMER0 Trigger PWM Function Enable Bit** When this bit is set, TIMER0 time-out event will update PWM_PHCHG with PHCHG_NXT register. 0 = TIMER0 trigger PWM function Disabled. 1 = TIMER0 trigger PWM function Enabled.
[29:28]	**A0POSSEL**	**ACMP0 Positive Input Source Select Bits** 00 = Select P1.5 as the input of ACMP0. 01 = Select P1.0 as the input of ACMP0. 10 = Select P1.2 as the input of ACMP0. 11 = Select P1.3 as the input of ACMP0.
[27]	**AOFFEN30**	**ACMP0 Trigger Channel 3 One Cycle Output Off Enable Bit** Setting this bit will force PWM channel 3 to output low lasting for at most one period cycle as long as ACMP0 trigger It; This feature is usually in step motor application. 0 = PWM0_CH3 one period cycle output low Disabled. 1 = PWM0_CH3 one period cycle output low Enabled. **Note:** This function is only available for PWM0_CH0~ PWM0_CH3.
[26]	**AOFFEN20**	**ACMP0 Trigger Channel 2 One Cycle Output Off Enable Bit** Setting this bit will force PWM channel 2 to output low lasting for at most one period cycle as long as ACMP0 trigger It; This feature is usually in step motor application. 0 = PWM0_CH2 one period cycle output low Disabled. 1 = PWM0_CH2 one period cycle output low Enabled. **Note:** This function is only available for PWM0_CH0~PWM0_CH3.

Bits	Description	
[25]	**AOFFEN10**	**ACMP0 Trigger Channel 1 One Cycle Output Off Enable Bit** Setting this bit will force PWM channel 1 to output low lasting for at most one period cycle as long as ACMP0 trigger It; This feature is usually in step motor application. 0 = PWM0_CH1 one period cycle output low Disabled. 1 = PWM0_CH1 one period cycle output low Enabled. **Note:** This function is only available for PWM0_CH0~PWM0_CH3.
[24]	**AOFFEN00**	**ACMP0 Trigger Channel 0 One Cycle Output Off Enable Bit** Setting this bit will force PWM channel 0 to output low lasting for at most one period cycle as long as ACMP0 trigger It; This feature is usually in step motor application. 0 = PWM0_CH0 one period cycle output low Disabled. 1 = PWM0_CH0 one period cycle output low Enabled. **Note:** This function is only available for PWM0_CH0~PWM0_CH3.
[23]	**ACMP1TEN**	**ACMP1 Trigger Function Enable Bit** 0 = ACMP1 trigger PWM function Disabled. 1 = ACMP1 trigger PWM function Enabled. **Note:** This bit will be auto cleared when ACMP1 trigger PWM if AUTOCLR1 is set.
[22]	**TMR1TEN**	**TIMER1 Trigger PWM Function Enable Bit** When this bit is set, TIMER1 time-out event will update PWM_PHCHG with PHCHG_NXT register. 0 = TIMER1 trigger PWM function Disabled. 1 = TIMER1 trigger PWM function Enabled.
[21:20]	**A1POSSEL**	**ACMP1 Positive Input Source Select Bits** 00 = Select P3.1 as the input of ACMP1. 01 = Select P3.2 as the input of ACMP1. 10 = Select P3.3 as the input of ACMP1. 11 = Select P3.4 as the input of ACMP1.
[19]	**AOFFEN31**	**ACMP1 Trigger Channel 3 One Cycle Output Off Enable Bit** Setting this bit will force PWM channel 3 to output low lasting for at most one period cycle as long as ACMP1 trigger It; This feature is usually in step motor application. 0 = PWM0_CH3 one period cycle output low Disabled. 1 = PWM0_CH3 one period cycle output low Enabled. **Note:** This function is only available for PWM0_CH0~PWM0_CH3.
[18]	**AOFFEN21**	**ACMP1 Trigger Channel 2 One Cycle Output Off Enable Bit** Setting this bit will force PWM channel 2 to output low lasting for at most one period cycle as long as ACMP1 trigger It; This feature is usually in step motor application. 0 = PWM0_CH2 one period cycle output low Disabled. 1 = PWM0_CH2 one period cycle output low Enabled. **Note:** This function is only available for PWM0_CH0~PWM0_CH 3.
[17]	**AOFFEN11**	**ACMP1 Trigger Channel 1 One Cycle Output Off Enable Bit** Setting this bit will force PWM channel 1 to output low lasting for at most one period cycle as long as ACMP1 trigger It; This feature is usually in step motor application. 0 = PWM0_CH1 one period cycle output low Disabled. 1 = PWM0_CH1 one period cycle output low Enabled. **Note:** This function is only available for PWM0_CH0~PWM0_CH3.

Bits	Description	
[16]	**AOFFEN01**	**ACMP1 Trigger Channel 0 One Cycle Output Off Enable Bit** Setting this bit will force PWM channel 0 to output low lasting for at most one period cycle as long as ACMP1 trigger It; This feature is usually in step motor application. 0 = PWM0_CH0 one period cycle output low Disabled. 1 = PWM0_CH0 one period cycle output low Enabled. **Note:** This function is only available for PWM0_CH0~PWM0_CH3.
[15]	**AUTOCLR1**	**Hardware Auto Clear ACMP1TEN** 0 = Hardware will auto clear ACMP1TEN when ACMP1 trigger PWM. 1 = Hardware will not auto clear ACMP1TEN when ACMP1 trigger PWM.
[14]	**AUTOCLR0**	**Hardware Auto Clear ACMP0TEN** 0 = Hardware will auto clear ACMP0TEN when ACMP0 trigger PWM. 1 = Hardware will not auto clear ACMP0TEN when ACMP0 trigger PWM.
[13:8] n=0,1..5	**MSKENn**	**PWMn Output Mask Enable Bits** 0 = Output MSKDATn specified in bit n of PWM_PHCHG register. 1 = Output the original channel n waveform.
[7]	**MSKDAT7**	**PWM0_CH7 (GPIO P0.0) Mask Data** When MASKEND7 Is 1, channel 7's output level is MSKDAT7. 0 = PWM0_CH7 output low level. 1 = PWM0_CH7 output high level.
[6]	**MSKDAT6**	**PWM0_CH6 (GPIO P0.1) Mask Data** When MASKEND6 Is 1, channel 6's output level is MSKDAT6. 0 = PWM0_CH6 output low level. 1 = PWM0_CH6 output high level.
[5:0] n=0,1..5	**MSKDATn**	**PWM0_CHn Mask Data** When MSKENn is 0, channel n's output level is MSKDATn. 0 = PWM0_CHn output low level. 1 = PWM0_CHn output high level.

Phase Change Register (PWM_PHCHGNXT)

Register	Offset	R/W	Description	Reset Value
PWM_PHCHGNXT	PWM0_BA+0x7C	R/W	PWM Next Phase Change Register	0x0000_3F00

31	30	29	28	27	26	25	24
ACMP0TEN	TMR0TEN	A0POSSEL		AOFFEN30	AOFFEN20	AOFFEN10	AOFFEN00
23	22	21	20	19	18	17	16
ACMP1TEN	TMR1TEN	A1POSSEL		AOFFEN31	AOFFEN21	AOFFEN11	AOFFEN01
15	14	13	12	11	10	9	8
AUTOCLR1	AUTOCLR0	MSKEN5	MSKEN4	MSKEN3	MSKEN2	MSKEN1	MSKEN0
7	6	5	4	3	2	1	0
MSKDAT7	MSKDAT6	MSKDAT5	MSKDAT4	MSKDAT3	MSKDAT2	MSKDAT1	MSKDAT0

Bits	Description	
[31]	**ACMP0TEN**	**ACMP0 Trigger Function Enable Bit** 0 = ACMP0 trigger PWM function Disabled. 1 = ACMP0 trigger PWM function Enabled. **Note:** This bit will be auto cleared when ACMP0 trigger PWM if AUTOCLR0 is set.
[30]	**TMR0TEN**	**TMR0 Trigger PWM Function Enable Bit** When this bit is set, TMR0 time-out event will update PWM_PHCHG with PHCHG_NXT register. 0 = TMR0 trigger PWM function Disabled. 1 = TMR0 trigger PWM function Enabled.
[29:28]	**A0POSSEL**	**ACMP0 Positive Input Source Select Bits** 00 = Select P1.5 as the input of ACMP0. 01 = Select P1.0 as the input of ACMP0. 10 = Select P1.2 as the input of ACMP0. 11 = Select P1.3 as the input of ACMP0.
[27]	**AOFFEN30**	**ACMP0 Trigger Channel 3 One Cycle Output Off Enable Bit** Setting this bit will force PWM channel 3 to output low lasting for at most one period cycle as long as ACMP0 trigger It; This feature is usually in step motor application. 0 = PWM0_CH3 one period cycle output low Disabled. 1 = PWM0_CH3 one period cycle output low Enabled. **Note:** This function is only available for PWM0_CH0~PWM0_CH3.
[26]	**AOFFEN20**	**ACMP0 Trigger Channel 2 One Cycle Output Off Enable Bit** Setting this bit will force PWM channel 2 to output low lasting for at most one period cycle as long as ACMP0 trigger It; This feature is usually in step motor application. 0 = PWM0_CH2 one period cycle output low Disabled. 1 = PWM0_CH2 one period cycle output low Enabled. **Note:** This function is only available for PWM0_CH0~PWM0_CH3.

Bits	Description	
[25]	**AOFFEN10**	**ACMP0 Trigger Channel 1 One Cycle Output Off Enable Bit** Setting this bit will force PWM channel 1 to output low lasting for at most one period cycle as long as ACMP0 trigger It; This feature is usually in step motor application. 0 = PWM0_CH1 one period cycle output low Disabled. 1 = PWM0_CH1 one period cycle output low Enabled. **Note:** This function is only available for PWM0_CH0~PWM0_CH3.
[24]	**AOFFEN00**	**ACMP0 Trigger Channel 0 One Cycle Output Off Enable Bit** Setting this bit will force PWM channel 0 to output low lasting for at most one period cycle as long as ACMP0 trigger It; This feature is usually in step motor application. 0 = PWM0_CH0 one period cycle output low Disabled. 1 = PWM0_CH0 one period cycle output low Enabled. **Note:** This function is only available for PWM0_CH0~PWM0_CH3.
[23]	**ACMP1TEN**	**ACMP1 Trigger Function Enable Bit** 0 = ACMP1 trigger PWM function Disabled. 1 = ACMP1 trigger PWM function Enabled. **Note:** This bit will be auto cleared when ACMP1 trigger PWM if AUTOCLR1 is set.
[22]	**TMR1TEN**	**TMR1 Trigger PWM Function Enable Bit** When this bit is set, TMR1 time-out event will update PWM_PHCHG with PHCHG_NXT register. 0 = TMR1 trigger PWM function Disabled. 1 = TMR1 trigger PWM function Enabled.
[21:20]	**A1POSSEL**	**ACMP1 Positive Input Source Select Bits** 00 = Select P3.1 as the input of ACMP1. 01 = Select P3.2 as the input of ACMP1. 10 = Select P3.3 as the input of ACMP1. 11 = Select P3.4 as the input of ACMP1.
[19]	**AOFFEN31**	**ACMP1 Trigger Channel 3 One Cycle Output Off Enable Bit** Setting this bit will force PWM channel 3 to output low lasting for at most one period cycle as long as ACMP1 trigger It; This feature is usually in step motor application. 0 = PWM0_CH3 one period cycle output low Disabled. 1 = PWM0_CH3 one period cycle output low Enabled. **Note:** This function is only available for PWM0_CH0~PWM0_CH3.
[18]	**AOFFEN21**	**ACMP1 Trigger Channel 2 One Cycle Output Off Enable Bit** Setting this bit will force PWM channel 2 to output low lasting for at most one period cycle as long as ACMP1 trigger It; This feature is usually in step motor application. 0 = PWM0_CH2 one period cycle output low Disabled. 1 = PWM0_CH2 one period cycle output low Enabled. **Note:** This function is only available for PWM0_CH0~PWM0_CH3.
[17]	**AOFFEN11**	**ACMP1 Trigger Channel 1 One Cycle Output Off Enable Bit** Setting this bit will force PWM channel 1 to output low lasting for at most one period cycle as long as ACMP1 trigger It; This feature is usually in step motor application. 0 = PWM0_CH1 one period cycle output low Disabled. 1 = PWM0_CH1 one period cycle output low Enabled. **Note:** This function is only available for PWM0_CH0~PWM0_CH3.

Bits	Description	
[16]	**AOFFEN01**	**ACMP1 Trigger Channel 0 One Cycle Output Off Enable Bit** Setting this bit will force PWM channel 0 to output low lasting for at most one period cycle as long as ACMP1 trigger It; This feature is usually in step motor application. 0 = PWM0_CH0 one period cycle output low Disabled. 1 = PWM0_CH0 one period cycle output low Enabled. **Note:** This function is only available for PWM0_CH0~PWM0_CH3.
[15]	**AUTOCLR1**	**Hardware Auto Clear ACMP1TEN** 0 = Hardware will auto clear ACMP1TEN when ACMP1 trigger PWM. 1 = Hardware will not auto clear ACMP1TEN when ACMP1 trigger PWM.
[14]	**AUTOCLR0**	**Hardware Auto Clear ACMP0TEN** 0 = Hardware will auto clear ACMP0TEN when ACMP0 trigger PWM. 1 = Hardware will not auto clear ACMP0TEN when ACMP0 trigger PWM.
[13:8] n=0,1..5	**MSKENn**	**PWM Output Mask Enable Bits** 0 = Output MSKDATn specified in bit n of PWM_PHCHG register. 1 = Output the original channel n waveform.
[7]	**MSKDAT7**	**PWM0_CH7 (GPIO P0.0) Mask Data** When MASKEND7 Is 1, channel 7's output level is MSKDAT7. 0 = PWM0_CH7 output low level. 1 = PWM0_CH7 output high level.
[6]	**MSKDAT6**	**PWM0_CH6 (GPIO P0.1) Mask Data** When MASKEND6 Is 1, channel 6's output level is MSKDAT6. 0 = PWM0_CH6 output low level. 1 = PWM0_CH6 output high level.
[5:0] n=0,1..5	**MSKDATn**	**PWM0_CHn Mask Data** When MSKENn is 0, channel n's output level is MSKDATn. 0 = PWM0_CHn output low level. 1 = PWM0_CHn output high level.

Phase Change Mask Register (PWM_PHCHGMSK)

Register	Offset	R/W	Description	Reset Value
PWM_PHCHGMSK	PWM0_BA+0x80	R/W	PWM Phase Change 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						POSCTL	
7	6	5	4	3	2	1	0
MASKEND7	MASKEND6	Reserved					

Bits	Description	
[31:10]	**Reserved**	Reserved.
[9]	**POSCTL[1]**	**Positive Input Control For ACMP1** 0 = The input of ACMP is controlled by CMP1CR. 1 = The input of ACMP is controlled by CMP1SEL of PWM_PHCHG register. **Note:** Register CMP1CR is described in Comparator Controller chapter.
[8]	**POSCTL[0]**	**Positive Input Control For ACMP0** 0 = The input of ACMP is controlled by CMP0CR. 1 = The input of ACMP is controlled by CMP0SEL of PWM_PHCHG register. **Note:** Register CMP0CR is described in Comparator Controller chapter.
[7]	**MASKEND7**	**PWM0_CH7 (GPIO P0.0) Output Mask Enable Bit** 0 = Output the original GPIO P0.0. 1 = Output MSKDAT7 specified in bit 7 of PWM_PHCHG register.
[6]	**MASKEND6**	**PWM0_CH6 (GPIO P0.1) Output Mask Enable Bit** 0 = Output the original GPIO P0.1. 1 = Output MSKDAT6 specified in bit 6 of PWM_PHCHG register.
[5:0]	**Reserved**	Reserved.

Interrupt Accumulation Control Register (PWM_IFA)

Register	Offset	R/W	Description	Reset Value
PWM_IFA	PWM0_BA+0x84	R/W	PWM Period Interrupt Accumulation Control Register	0x0000_00F0

31	30	29	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**
IFCNT				**Reserved**			**IFAEN**

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:4]	**IFCNT**	**Interrupt Accumulation Counter** When IFAEN is set, IFCNT will decrease when every ZIFn flag is set and when IFCNT reach to zero, the PWMn interrupt will occurred and IFCNT will reload itself.
[3:1]	**Reserved**	Reserved.
[0]	**IFAEN**	**Interrupt Accumulation Function Enable Bit** 0 = Interrupt accumulation function Disabled. 1 = Interrupt accumulation function Enabled.

Precise PWM Center-Aligned Type Control Register (PWM_PCACTL)

Register	Offset	R/W	Description	Reset Value
PWM_PCACTL	PWM0_BA+0x88	R/W	PWM Precise Center-Aligned Type 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							**PCAEN**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**PCAEN**	**PWM Precise Center-aligned Type Enable Bit** 0 = Precise center-aligned type Disabled. 1 = Precise center-aligned type Enabled.

Phase Change Mask Aligned Register (PWM_MSKALIGN)

Register	Offset	R/W	Description	Reset Value
PWM_MSKALIGN	PWM0_BA+0x8C	R/W	PWM Phase Change Mask Aligned Register	0x0000_3F00

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved		ALIGN5	ALIGN4	ALIGN3	ALIGN2	ALIGN1	ALIGN0
15	14	13	12	11	10	9	8
Reserved		MSKEN5	MSKEN4	MSKEN3	MSKEN2	MSKEN1	MSKEN0
7	6	5	4	3	2	1	0
Reserved		MSKDAT5	MSKDAT4	MSKDAT3	MSKDAT2	MSKDAT1	MSKDAT0

Bits	Description	
[31:22]	**Reserved**	Reserved.
[21:16]	**ALIGNn**	**PWM0_CHn Output Mask Aligned Enable Bit** 0 = PWM0_CHn output will mask immediately when mask function enabled. 1 = PWM0_CHn output will mask when output aligned to PWM period.
[15:14]	**Reserved**	Reserved.
[13:8] n=0,1..5	**MSKENn**	**PWM Output Mask Enable Bits** 0 = Output MSKDATn specified in bit n of PWM_PHCHG register. 1 = Output the original channel n waveform.
[7:6]	**Reserved**	Reserved.
[5:0] n=0,1..5	**MSKDATn**	**PWM0_CHn Mask Data** When MSKENn is 0, channel n's output level is MSKDATn. 0 = PWM0_CHn output low level. 1 = PWM0_CHn output high level.

Note: MSKENn and MSKDATn of PWM_PHCHG and PWM_MSKALIGN both control the same bits.

6.8 Watchdog Timer (WDT)

6.8.1 Overview

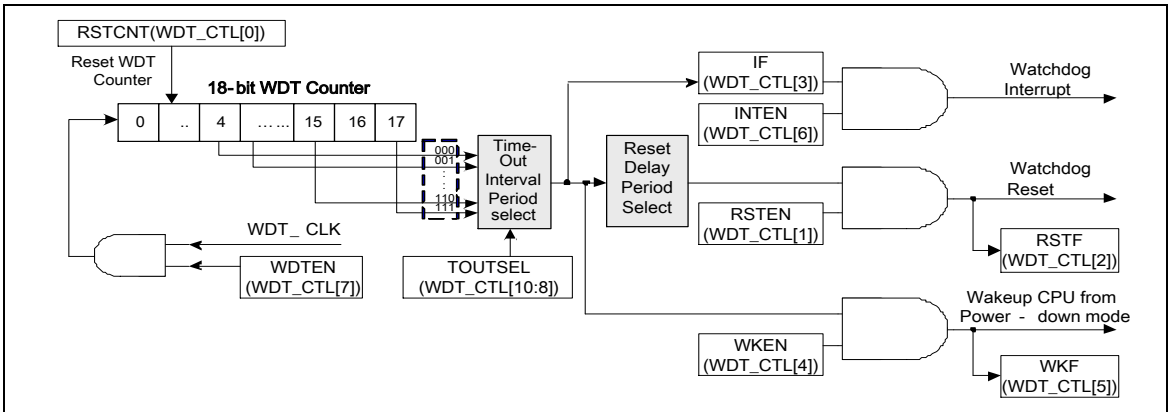
The Watchdog Timer is used to perform a system reset when system runs into an unknown state. This prevents system from hanging for an infinite period of time. Besides, the Watchdog Timer supports the function to wake-up system from Idle/Power-down mode.

6.8.2 Features

- 18-bit free running up counter for WDT time-out interval
- Selectable time-out interval (2^4 ~ 2^{18}) WDT_CLK cycles and the time-out interval is 1.6 ms ~ 26.214s if WDT_CLK = 10 kHz
- System kept in reset state for a period of (1 / WDT_CLK) * 63
- Supports selectable WDT reset delay period, including 1026、130、18 or 3 WDT_CLK reset delay period
- Supports to force WDT enabled after chip powered on or reset by setting CWDTEN[2:0] in Config0 register
- Supports WDT time-out wake-up function only if WDT clock source is selected as LIRC or LXT

6.8.3 Block Diagram

The WDT block diagram is shown in Figure 6.8-1.



Note1: WDT resets CPU and lasts 63 WDT_CLK.

Note2: The WDT reset delay period can be selected as 3/18/130/1026 WDT_CLK.

Figure 6.8-1 Watchdog Timer Block Diagram

6.8.4 Clock Control

The WDT clock control is shown in Figure 6.8-2.

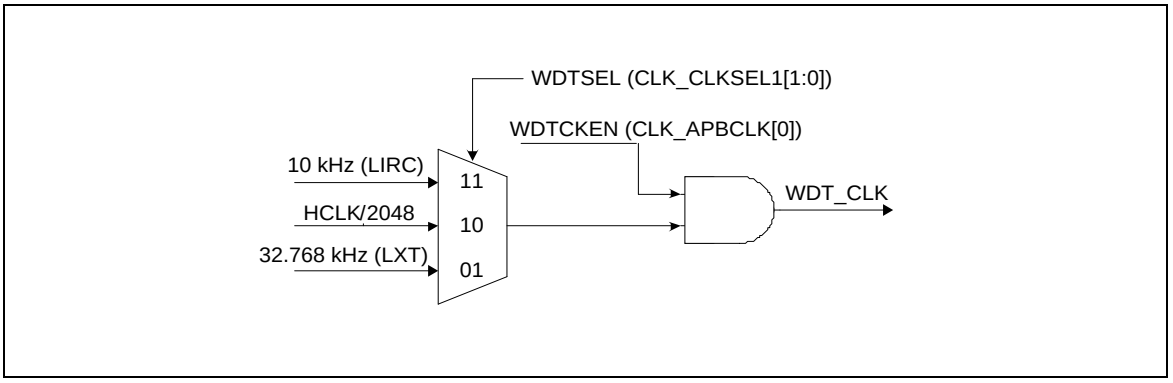


Figure 6.8-2 Watchdog Timer Clock Control

6.8.5 Basic Configuration

The WDT peripheral clock is enabled in WDTCKEN (CLK_APBCLK[0]) and clock source can be selected in WDTSEL (CLK_CLKSEL1[1:0]).

The WDT controller can also be forced enabled and active in 10 kHz after chip is powered on or reset while CWDTEN[2:0] (i.e. CWDTEN[2] is Config0[31], CWDTEN[1:0] is Config0[4:3]) is not configured to 111.

6.8.6 Functional Description

The WDT includes an 18-bit free running up counter with programmable time-out intervals. Table 6.8-1 shows the WDT time-out interval period selection and Figure 6.8-3 shows the WDT time-out interval and reset period timing.

6.8.6.1 WDT Time-out Interrupt

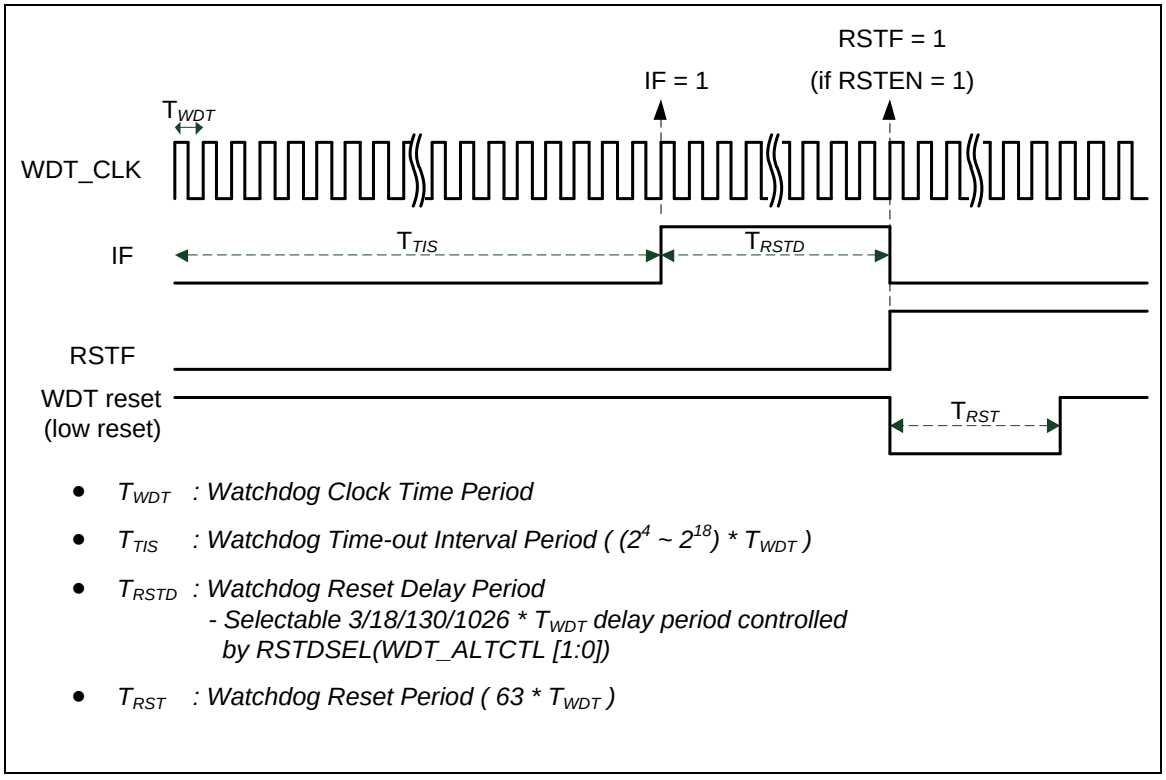
Setting WDTEN (WDT_CTL[7]) to 1 will enable the WDT function and the WDT counter to start counting up. There are eight time-out interval period can be selected by setting TOUTSEL (WDT_CTL[10:8]). When the WDT up counter reaches the TOUTSEL (WDT_CTL[10:8]) setting, the WDT time-out interrupt will occur and then WDT time-out interrupt flag IF (WDT_CTL[3]) will be set to 1 immediately.

6.8.6.2 WDT Reset Delay Period and Reset System

A specified T_{RSTD} reset delay period occurs when the IF (WDT_CTL[3]) is set to 1. User should set RSTCNT (WDT_CTL[0]) to reset the 18-bit WDT up counter value to avoid generating the WDT time-out reset signal before the T_{RSTD} reset delay period expires. Moreover, user should set RSTDSEL (WDT_ALTCTL [1:0]) to select reset delay period to clear WDT counter. If the WDT up counter value has not been cleared after the specified T_{RSTD} delay period expires, the WDT control will set RSTF (WDT_CTL[2]) to 1 if RSTEN (WDT_CTL[1]) bit is enabled, and then chip enters reset state immediately. Refer to Figure 6.8-3 Watchdog Timer Time-out Interval and Reset Period Timing. The T_{RST} reset period will keep the last 63 WDT clocks and then chip restart executing program from reset vector (0x0000_0000). The RSTF (WDT_CTL[2]) will keep 1 after WDT time-out resets the chip. User can check RSTF (WDT_CTL[2]) via software to recognize if the system has been reset by WDT time-out reset or not.

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 6.8-1 Watchdog Timer Time-out Interval Period Selection



- T_{WDT} : *Watchdog Clock Time Period*
- T_{TIS} : *Watchdog Time-out Interval Period ((2^4 ~ 2^{18}) * T_{WDT})*
- T_{RSTD} : *Watchdog Reset Delay Period*
  *- Selectable 3/18/130/1026 * T_{WDT} delay period controlled by RSTDSEL(WDT_ALTCTL [1:0])*
- T_{RST} : *Watchdog Reset Period (63 * T_{WDT})*

Figure 6.8-3 Watchdog Timer Time-out Interval and Reset Period Timing

6.8.6.3 WDT Wake-up

If WDT clock source is selected to LIRC or LXT, system can be woken up from Power-down mode while WDT time-out interrupt signal is generated and WKEN (WDT_CTL[4]) enabled. In the meanwhile, the WKF (WDT_CTL[5]) will be set to 1 automatically. User can check WKF (WDT_CTL[5]) status via software to recognize if the system has been woken up by WDT time-out interrupt or not.

6.8.7 Registers Map

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

Register	Offset	R/W	Description	Reset Value
WDT Base Address: **WDT_BA = 0x4000_4000**				
WDT_CTL	WDT_BA+0x00	R/W	WDT Control Register	0x0000_0700
WDT_ALTCTL	WDT_BA+0x04	R/W	WDT Alternative Control Register	0x0000_0000

6.8.8 Registers 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	**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**	**RSTCNT**

Bits	Description	
[31]	**ICEDEBUG**	**ICE Debug Mode Acknowledge Disable Bit (Write Protect)** 0 = ICE debug mode acknowledgement affects WDT counting. WDT up counter will be held while CPU is held by ICE. 1 = ICE debug mode acknowledgement Disabled. WDT up counter will keep going no matter CPU is held by ICE or not. **Note:** This bit is write protected. Refer to the SYS_REGLCTL register.
[30:11]	**Reserved**	Reserved.
[10:8]	**TOUTSEL**	**WDT Time-out Interval Selection (Write Protect)** These three bits select the time-out interval period for the WDT. 000 = 2^4 * WDT_CLK. 001 = 2^6 * WDT_CLK. 010 = 2^8 * WDT_CLK. 011 = 2^{10} * WDT_CLK. 100 = 2^{12} * WDT_CLK. 101 = 2^{14} * WDT_CLK. 110 = 2^{16} * WDT_CLK. 111 = 2^{18} * WDT_CLK. **Note:** This bit is write protected. Refer to the SYS_REGLCTL register.
[7]	**WDTEN**	**WDT Enable Bit (Write Protect)** 0 = WDT Disabled (This action will reset the internal up counter value). 1 = WDT Enabled. **Note:** This bit is write protected. Refer to the SYS_REGLCTL register.

Bits	Description	
[6]	**INTEN**	**WDT Time-out Interrupt Enable Bit (Write Protect)** If this bit is enabled, the WDT time-out interrupt signal is generated and inform to CPU. 0 = WDT time-out interrupt Disabled. 1 = WDT time-out interrupt Enabled. **Note:** This bit is write protected. Refer to the SYS_REGLCTL register.
[5]	**WKF**	**WDT Time-out Wake-up Flag (Write Protect)** This bit indicates the interrupt wake-up flag status of WDT 0 = WDT does not cause chip wake-up. 1 = Chip wake-up from Idle or Power-down mode if WDT time-out interrupt signal generated. **Note1:** This bit is write protected. Refer to the SYS_REGLCTL register. **Note2:** This bit is cleared by writing 1 to it.
[4]	**WKEN**	**WDT Time-out Wake-up Function Control (Write Protect)** If this bit is set to 1, while WDT time-out interrupt flag IF (WDT_CTL[3]) is generated to 1 and interrupt enable bit INTEN (WDT_CTL[6]) is enabled, the WDT time-out interrupt signal will generate a wake-up trigger event to chip. 0 = Wake-up trigger event Disabled if WDT time-out interrupt signal generated. 1 = Wake-up trigger event Enabled if WDT time-out interrupt signal generated. **Note1:** This bit is write protected. Refer to the SYS_REGLCTL register. **Note2:** Chip can be woken-up by WDT time-out interrupt signal generated only if WDT clock source is selected to LIRC or LXT.
[3]	**IF**	**WDT Time-out Interrupt Flag** This bit will be set to 1 while WDT up counter value reaches the selected WDT time-out interval 0 = WDT time-out interrupt did not occur. 1 = WDT time-out interrupt occurred. **Note:** This bit is cleared by writing 1 to it.
[2]	**RSTF**	**WDT Time-out Reset Flag** This bit indicates the system has been reset by WDT time-out reset or not. 0 = WDT time-out reset did not occur. 1 = WDT time-out reset occurred. **Note:** This bit is cleared by writing 1 to it.
[1]	**RSTEN**	**WDT Time-out Reset Enable Bit (Write Protect)** Setting this bit will enable the WDT time-out reset function If the WDT up counter value has not been cleared after the specific WDT reset delay period expires. 0 = WDT time-out reset function Disabled. 1 = WDT time-out reset function Enabled. **Note:** This bit is write-protected. Refer to the SYS_REGLCTL register.
[0]	**RSTCNT**	**Reset WDT Up Counter (Write Protect)** 0 = No effect. 1 = Reset the internal 18-bit WDT up counter value. **Note1:** This bit is write protected. Refer to the SYS_REGLCTL register. **Note2:** This bit will be automatically cleared by hardware.

WDT Alternative Control Register (WDT_ALTCTL)

Register	Offset	R/W	Description	Reset Value
WDT_ALTCTL	WDT_BA+0x04	R/W	WDT Alternative Control Register	0x0000_0000

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

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1:0]	**RSTDSEL**	**WDT Reset Delay Selection (Write Protect)** When WDT time-out happened, user has a time named WDT Reset Delay Period to clear WDT counter by setting RSTCNT (WDT_CTL[0]) to prevent WDT time-out reset happened. User can select a suitable setting of RSTDSEL for different WDT Reset Delay Period. 00 = WDT Reset Delay Period is 1026 * WDT_CLK. 01 = WDT Reset Delay Period is 130 * WDT_CLK. 10 = WDT Reset Delay Period is 18 * WDT_CLK. 11 = WDT Reset Delay Period is 3 * WDT_CLK. **Note1:** This bit is write protected. Refer to the SYS_REGLCTL register. **Note2:** This register will be reset to 0 if WDT time-out reset happened.

6.9 Window Watchdog Timer (WWDT)

6.9.1 Overview

The Window Watchdog Timer (WWDT) is used to perform a system reset within a specified window period to prevent software run to uncontrollable status by any unpredictable condition.

6.9.2 Features

- 6-bit down counter value (CNTDAT) and 6-bit compare value (CMPDAT) to make the WWDT time-out window period flexible
- Supports 4-bit value (PSCSEL) to programmable maximum 11-bit prescale counter period of WWDT counter

6.9.3 Block Diagram

The WWDT block diagram is shown in Figure 6.9-1

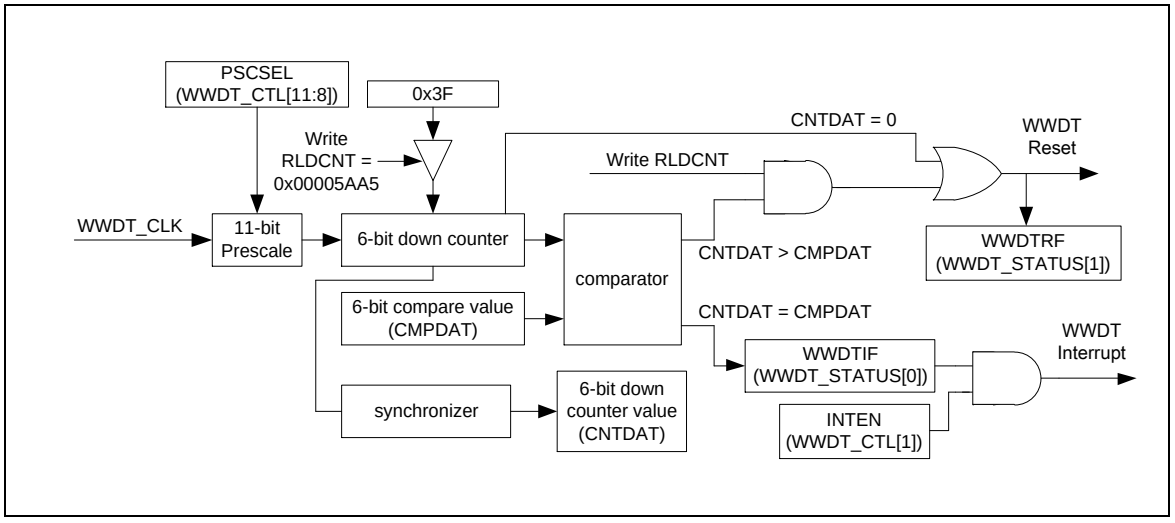


Figure 6.9-1 WWDT Block Diagram

6.9.4 Clock Control

The WWDT clock control is shown in Figure 6.9-2

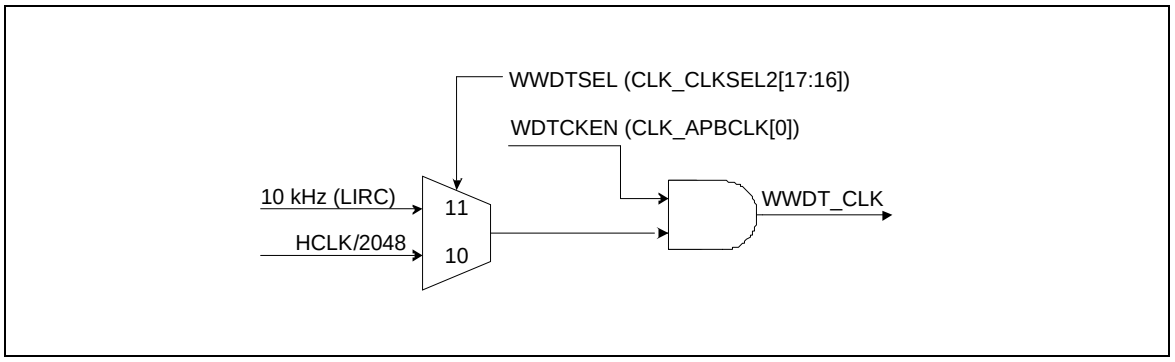


Figure 6.9-2 WWDT Clock Control

6.9.5 Basic Configuration

The WWDT peripheral clock is enabled in WDTCKEN (CLK_APBCLK[0]) and clock source can be selected in WWDTSEL[1:0] (CLK_CLKSEL2[17:16]).

6.9.6 Functional Description

The WWDT includes a 6-bit down counter with programmable prescale value to define different WWDT time-out intervals. The clock source of 6-bit WWDT is based on system clock divide 2048 (HCLK/2048) or 10 kHz internal low speed RC oscillator (LIRC) with a programmable 11-bit prescale counter value which controlled by PSCSEL (WWDT_CTL[11:8]). Also, the correlate of PSCSEL (WWDT_CTL[11:8]) and prescale value are listed in Table 6.9-1.

PSCSEL	Prescaler Value	Max. Time-Out Period	Max. Time-Out Interval (WWDT_CLK=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 6.9-1 WWDT Prescaler Value Selection

6.9.6.1 WWDT Counting

When the WWDTEN (WWDT_CTL[0]) is set, WWDT down counter will start counting from 0x3F to 0. To prevent program runs to disable WWDT counter counting unexpected, the WWDT_CTL register can only be written once after chip is powered on or reset. User cannot disable WWDT counter counting (WWDTEN), change counter prescale period (PSCSEL) or change window compare value (CMPDAT) while WWDTEN (WWDT_CTL[0]) has been enabled by user unless chip is reset.

To avoid the system is reset while CPU clock is disabled, the WWDT counter will stop counting when CPU enters Idle/Power-down mode. After CPU enters normal mode, the WWDT counter will start down counting.

6.9.6.2 WWDT Compare Match Interrupt

During down counting by the WWDT counter, the WWDTIF (WWDT_STATUS[0]) is set to 1 while the WWDT counter value (CNTDAT) is equal to window compare value (CMPDAT) and WWDTIF can be cleared by user; if INTEN (WWDT_CTL[1]) is also set to 1 by user, the WWDT compare match interrupt signal is generated also while WWDTIF is set to 1 by hardware.

6.9.6.3 WWDT Reset System

When WWDTIF (WWDT_STATUS[0]) is generated, user must reload WWDT counter value to 0x3F by writing 0x00005AA5 to WWDT_RLDCNT register, and also to prevent WWDT counter value reached to 0 and generate WWDT reset system signal to inform system reset. If current CNTDAT (WWDT_CNT[5:0]) is larger than CMPDAT (WWDT_CTL[21:16]) and user writes 0x00005AA5 to the WWDT_RLDCNT register, the WWDT reset system signal will be generated immediately to cause chip reset also.

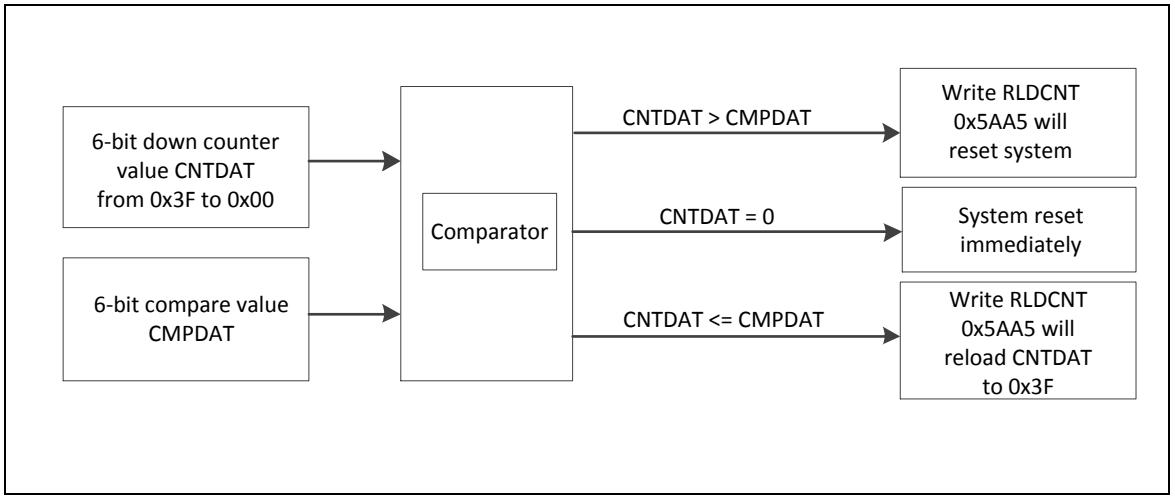


Figure 6.9-3 WWDT Reset and Reload Behavior

6.9.6.4 WWDT Window Setting Limitation

When user writes 0x00005AA5 to WWDT_RLDCNT register to reload WWDT counter value to 0x3F, it needs 3 WWDT clocks to sync the reload command to actually perform reload action. Notice that if user set PSCSEL (WWDT_CTL[11:8]) to 0000, the counter prescale value should be as 1, and the CMPDAT (WWDT_CTL[21:16]) must be larger than 2. Otherwise, writing WWDT_RLDCNT register to reload WWDT counter value to 0x3F is unavailable, WWDTIF (WWDT_STATUS[0]) is generated, and WWDT reset system event always happened.

PSCSEL	Prescale Value	Valid CMPDAT Value
0000	1	0x3 ~ 0x3F
0001	2	0x2 ~ 0x3F
Others	Others	0x0 ~ 0x3F

Table 6.9-2 CMPDAT Setting Limitation

6.9.7 Register Map

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

Register	Offset	R/W	Description	Reset Value
WWDT Base Address: **WWDT_BA = 0x4000_4100**				
WWDT_RLDCNT	WWDT_BA+0x00	W	WWDT Reload Counter Register	0x0000_0000
WWDT_CTL	WWDT_BA+0x04	R/W	WWDT Control Register	0x003F_0800
WWDT_STATUS	WWDT_BA+0x08	R/W	WWDT Status Register	0x0000_0000
WWDT_CNT	WWDT_BA+0x0C	R	WWDT Counter Value Register	0x0000_003F

6.9.8 Register Description

WWDT Reload Counter Register (WWDT_RLDCNT)

Register	Offset	R/W	Description	Reset Value
WWDT_RLDCNT	WWDT_BA+0x00	W	WWDT Reload Counter Register	0x0000_0000

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

Bits	Description	
[31:0]	**RLDCNT**	**WWDT Reload Counter Register** Writing 0x00005AA5 to this register will reload the WWDT counter value to 0x3F. **Note:** User can only write WWDT_RLDCNT register to reload WWDT counter value when current WWDT counter value between 0 and CMPDAT (WWDT_CTL[21:16]). If user writes WWDT_RLDCNT when current WWDT counter value is larger than CMPDAT, WWDT reset signal will generate immediately.

WWDT Control Register (WWDT_CTL)

Register	Offset	R/W	Description	Reset Value
WWDT_CTL	WWDT_BA+0x04	R/W	WWDT Control Register	0x003F_0800

Note: This register can be write only one time after chip is powered on or reset.

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

Bits	Description	
[31]	**ICEDEBUG**	**ICE Debug Mode Acknowledge Disable Bit** 0 = ICE debug mode acknowledgement effects WWDT counting. WWDT down counter will be held while CPU is held by ICE. 1 = ICE debug mode acknowledgement Disabled. WWDT down counter will keep going no matter CPU is held by ICE or not.
[30:22]	**Reserved**	Reserved.
[21:16]	**CMPDAT**	**WWDT Window Compare Bits** Set this register to adjust the valid reload window. **Note:** User can only write WWDT_RLDCNT register to reload WWDT counter value when current WWDT counter value between 0 and CMPDAT. If user writes WWDT_RLDCNT register when current WWDT counter value larger than CMPDAT, WWDT reset signal will generate immediately.
[15:12]	**Reserved**	Reserved.
[11:8]	**PSCSEL**	**WWDT Counter Prescale Period Select Bits** 0000 = Pre-scale is 1; Max time-out period is 1 * 64 * WWDT_CLK. 0001 = Pre-scale is 2; Max time-out period is 2 * 64 * WWDT_CLK. 0010 = Pre-scale is 4; Max time-out period is 4 * 64 * WWDT_CLK. 0011 = Pre-scale is 8; Max time-out period is 8 * 64 * WWDT_CLK. 0100 = Pre-scale is 16; Max time-out period is 16 * 64 * WWDT_CLK. 0101 = Pre-scale is 32; Max time-out period is 32 * 64 * WWDT_CLK. 0110 = Pre-scale is 64; Max time-out period is 64 * 64 * WWDT_CLK. 0111 = Pre-scale is 128; Max time-out period is 128 * 64 * WWDT_CLK. 1000 = Pre-scale is 192; Max time-out period is 192 * 64 * WWDT_CLK. 1001 = Pre-scale is 256; Max time-out period is 256 * 64 * WWDT_CLK. 1010 = Pre-scale is 384; Max time-out period is 384 * 64 * WWDT_CLK. 1011 = Pre-scale is 512; Max time-out period is 512 * 64 * WWDT_CLK. 1100 = Pre-scale is 768; Max time-out period is 768 * 64 * WWDT_CLK.

		1101 = Pre-scale is 1024; Max time-out period is 1024 * 64 * WWDT_CLK. 1110 = Pre-scale is 1536; Max time-out period is 1536 * 64 * WWDT_CLK. 1111 = Pre-scale is 2048; Max time-out period is 2048 * 64 * WWDT_CLK.
[7:2]	**Reserved**	Reserved.
[1]	**INTEN**	**WWDT Interrupt Enable Bit** If this bit is enabled, the WWDT counter compare match interrupt signal is generated and inform to CPU. 0 = WWDT counter compare match interrupt Disabled. 1 = WWDT counter compare match interrupt Enabled.
[0]	**WWDTEN**	**WWDT Enable Bit** Set this bit to enable 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**	**WWDT Timer-out Reset Flag** This bit indicates the system has been reset by WWDT time-out reset or not. 0 = WWDT time-out reset did not occur. 1 = WWDT time-out reset occurred. **Note:** This bit is cleared by writing 1 to it.
[0]	**WWDTIF**	**WWDT Compare Match Interrupt Flag** This bit indicates the interrupt flag status of WWDT while WWDT counter value matches CMPDAT (WWDT_CTL[21:16]). 0 = No effect. 1 = WWDT counter value matches CMPDAT. **Note:** This bit is cleared by writing 1 to it.

WWDT Counter Value Register (WWDT_CNT)

Register	Offset	R/W	Description	Reset Value
WWDT_CNT	WWDT_BA+0x0C	R	WWDT Counter Value Register	0x0000_003F

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

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5:0]	**CNTDAT**	**WWDT Counter Value** CNTDAT will be updated continuously to monitor 6-bit WWDT down counter value.

6.10 UART Controller (UART)

6.10.1 Overview

The NuMicro® Mini58 series provides two channels of Universal Asynchronous Receiver/Transmitters (UART). The UART0 performs supports flow control function. The UART0 performs a serial-to-parallel conversion on data received from the peripheral, and a parallel-to-serial conversion on data transmitted from the CPU. The UART0 controller also supports IrDA SIR Function, and RS-485 function mode. The UART0 channel supports six types of interrupts. The UART1 channel supports five types of interrupts. The UART1 only performs a serial-to-parallel conversion on data received from the peripheral, and a parallel-to-serial conversion on data transmitted from the CPU. The UART0 has 16 bytes Receiver/Transmitter FIFO. The UART1 only has one Receiver/Transmitter buffer.

6.10.2 Features

- Full duplex, asynchronous communications
- Separates receive/transmit 16/16 bytes entry FIFO for data payloads (Only Available in UART0)
- Separates receive/transmit 1/1 byte buffer for data payloads (Only Available in UART1)
- Supports hardware auto flow control/flow control function (CTS, RTS) and programmable RTS flow control trigger level (Only Available in UART0)
- Programmable receiver buffer trigger level (Only Available in UART0)
- Supports programmable baud-rate generator for each channel individually
- Supports CTS wake-up function (Only Available in UART0)
- 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 UART_TOUT[15:8] register
- 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 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 (Only Available in UART0)
  - Supports 3/16-bit duration for normal mode
- Supports RS-485 function mode (Only Available in UART0)
  - Supports RS-485 9-bit mode
  - Supports hardware or software enable to program RTS pin to control RS-485 transmission direction directly

6.10.3 Block Diagram

The UART0 clock control and block diagram are shown in Figure 6.10-1 and Figure 6.10-3. The UART1 clock control and block diagram are shown in Figure 6.10-2 and Figure 6.10-4.

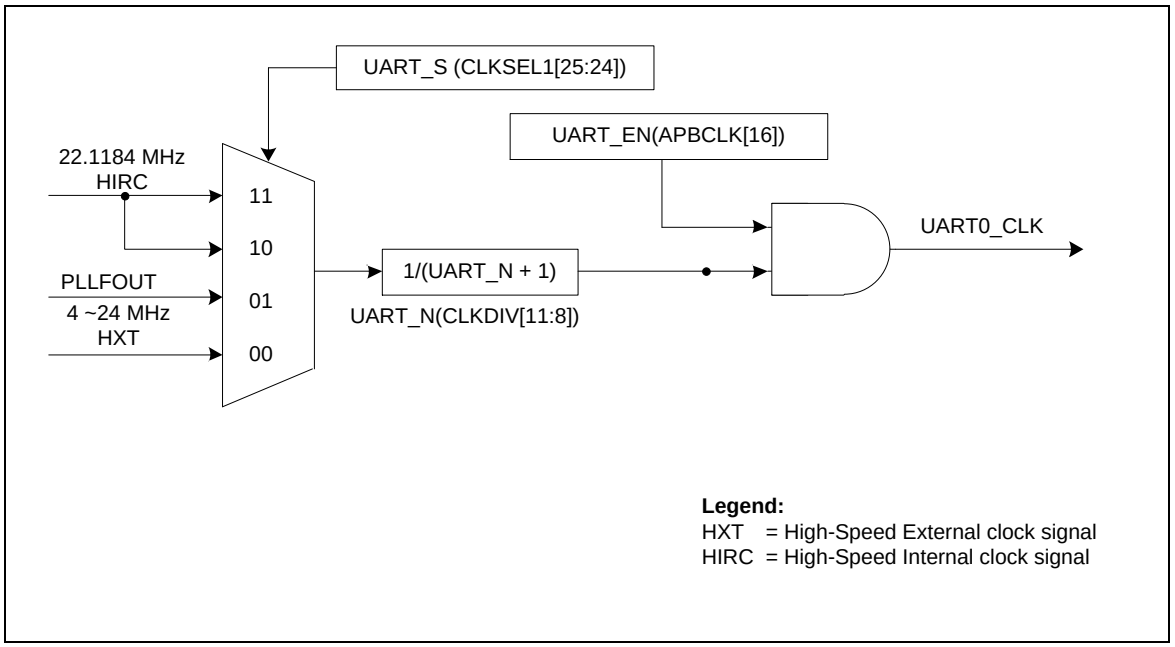


Figure 6.10-1 UART0 Controller Clock Control

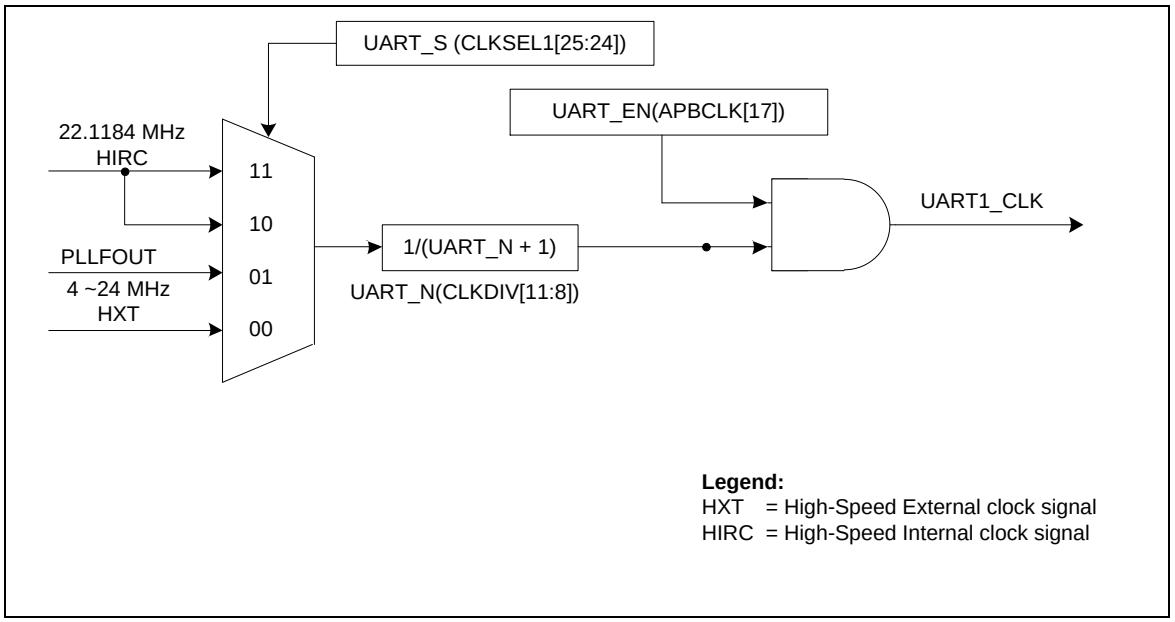


Figure 6.10-2 UART1 Controller Clock Control

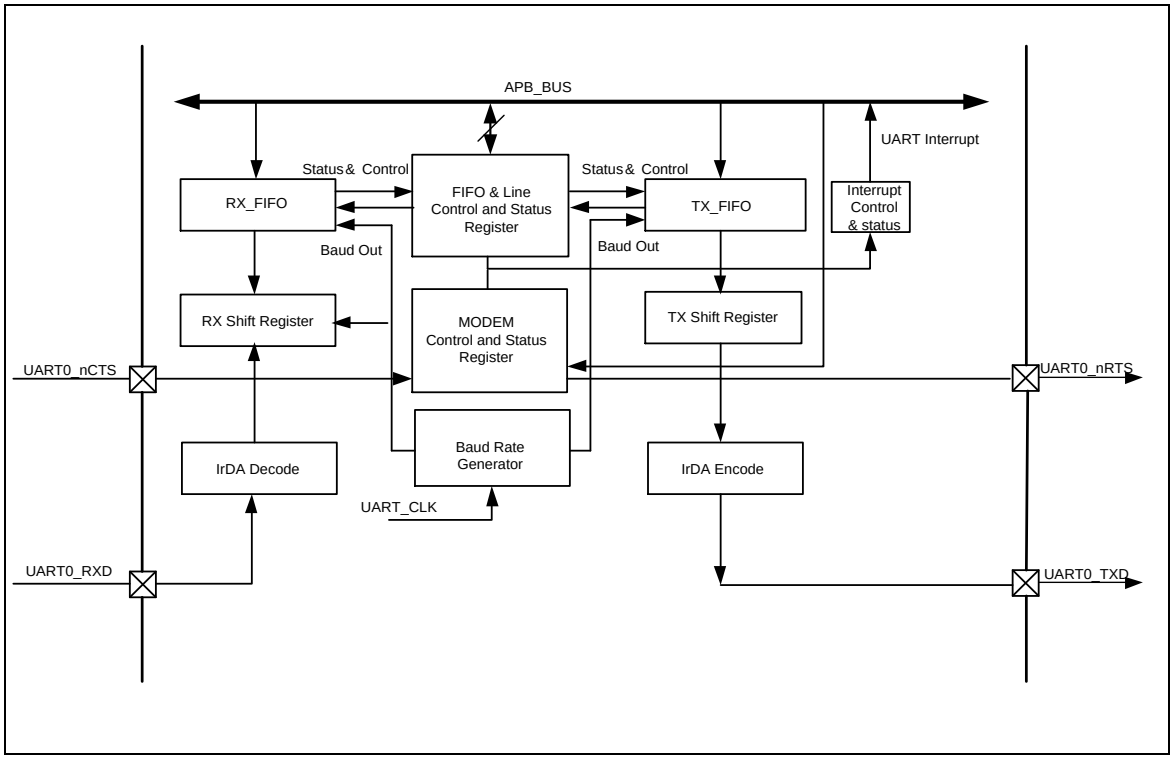


Figure 6.10-3 UART0 Controller Block Diagram

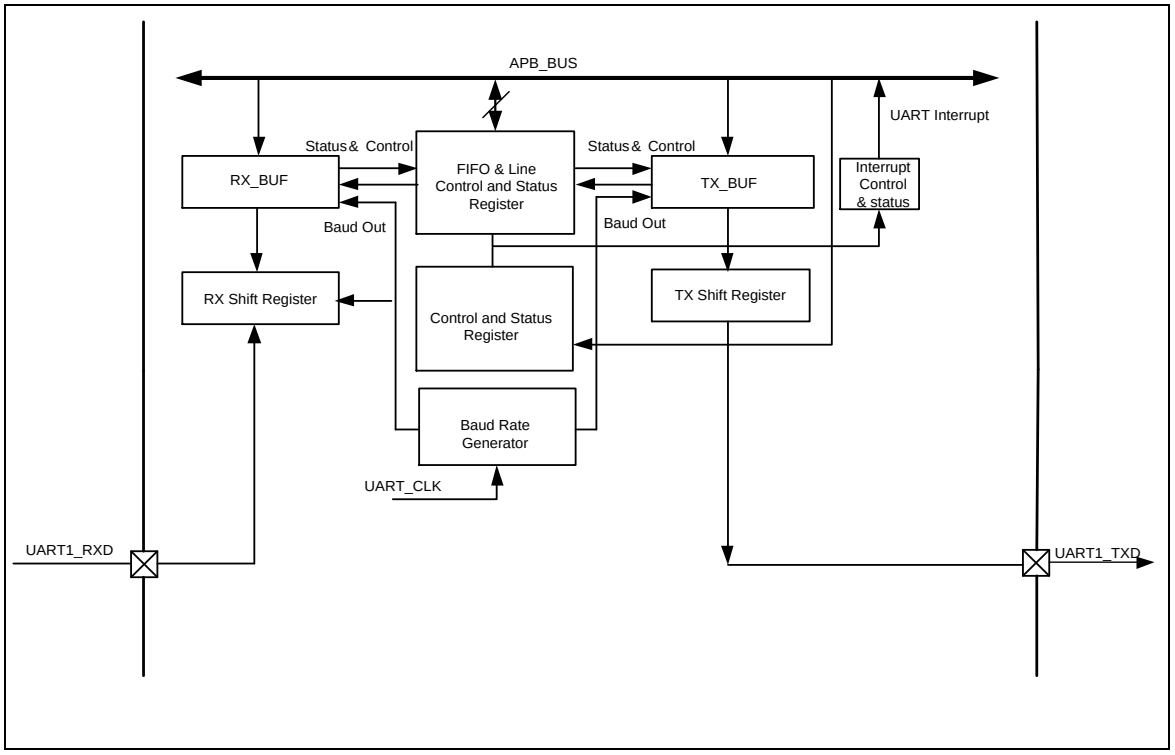


Figure 6.10-4 UART1 Controller Block Diagram

Each block is described in detail as follows:

TX_FIFO/TX_BUF

The UART0 transmitter is buffered with a 16-byte FIFO to reduce the number of interrupts presented to the CPU. The UART1 transmitter is buffered with only a 1-byte.

RX_FIFO/RX_BUF

The UART0 receiver is buffered with a 16-byte FIFO (plus three error bits per byte) to reduce the number of interrupts presented to the CPU. The UART1 receiver is buffered with a 1-byte buffer (plus three error bits per byte)

TX shift Register

This block is shifting the transmitting data out serial control block.

RX Shift Register

This block is shifting the receiving data in serial control block.

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. (Only Available in UART0)

IrDA Decode

This block is IrDA decoding control block. (Only Available in UART0)

FIFO & Line Control and Status Register

This field is register set that including the FIFO control registers (UART_FIFO), FIFO status registers (UART_FIFOSTS), and line control register (UART_LINE) for transmitter and receiver. The time-out control register (UART_TOUT) identifies the condition of time-out interrupt.

Interrupt Control and Status Register

There are six types of interrupts, transmitter FIFO empty interrupt (THERIF), receiver data available interrupt (RDAIF), receive line status interrupt (parity error or framing error or break interrupt) (RLSIF), time-out interrupt (RXTOINT), Buffer error interrupt (BUFERRINT), nCTS wake-up interrupt (Only available in UART0). Interrupt enabling register (UART_INTEN) enable or disable the responding interrupt and interrupt status register (UART_INTSTS) identifying the occurrence of the responding interrupt.

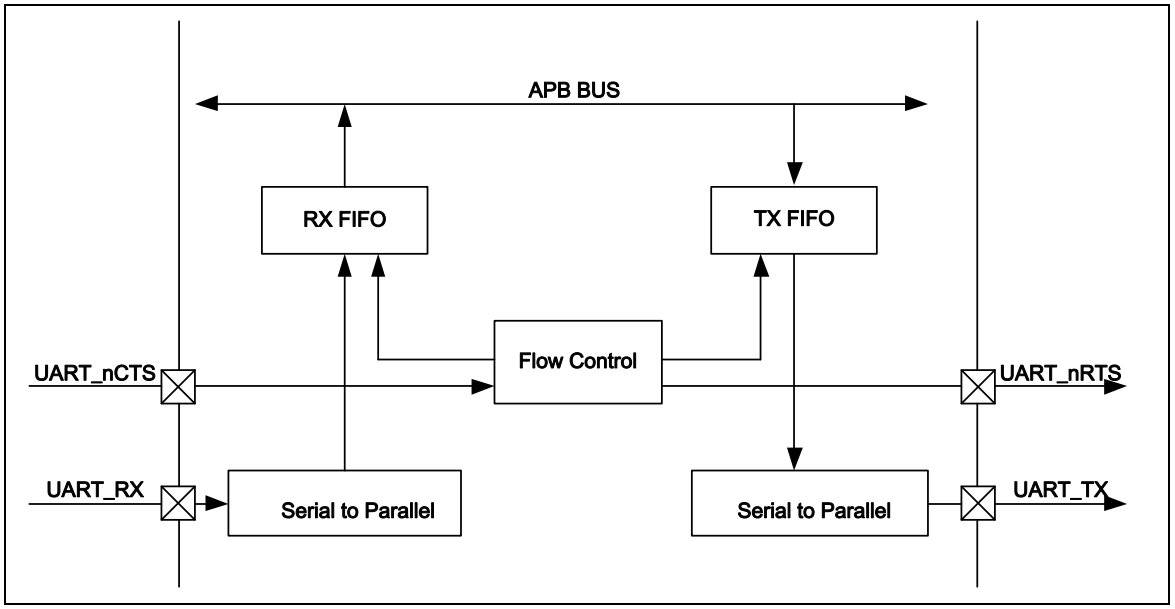


Figure 6.10-5 Auto Flow Control Block Diagram

6.10.4 Basic Configuration

The UART Controller function pins are configured in SYS_P0_MFP/SYS_P1_MFP registers for UART0 and configured in SYS_P1_MFP/ SYS_P2_MFP/ SYS_P4_MFP registers for UART1.

The UART Controller clock are enabled in UART_EN (CLK_APBCLK[16]) for UART.

The UART Controller clock source is selected by UARTSEL (CLKSEL[25:24]).

The UART Controller clock pre-scale is determined by UARTDIV (CLK_CLKDIV[11:8]).

6.10.5 Functional Description

The UART0 Controller supports three function modes including UART, IrDA, and RS-485 mode. User can select a function by setting the UART_FUNSEL register. The UART1 Controller only supports UART data transmit and receive mode.

6.10.5.1 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. The baud rate equation is Baud Rate = UART_CLK / M * [BRD + 2], where M and BRD are defined in Baud Rate Divider Register (UART_BAUD). Figure 6.10-6 lists the UART baud rate equations in the various conditions and UART baud rate parameter settings. There is no error for the baud rate results calculated through the baud rate parameter and register setting below. In IrDA function mode, the baud rate generator must be set in Mode 0. (Only Available in UART0)

Mode	BAUDM1	BAUDM0	Divider X	BRD	M	Baud Rate Equation
Mode 0	0	0	B	A	16	UART_CLK / [16 * (A+2)]

Mode 1	1	0	B	A	B+1	UART_CLK / [(B+1) * (A+2)], B must >= 8
Mode 2	1	1	Don't care	A	1	UART_CLK / (A+2), A must >=8

Figure 6.10-6 Controller Baud Rate Equation Table

UART Peripheral Clock = 22.1184 MHz			
Baud Rate	**Mode 0**	**Mode 1**	**Mode 2**
921600	Not support	A=0, B=11	A=22
460800	A=1	A=1, B=15 A=2, B=11	A=46
230400	A=4	A=4, B=15 A=6, B=11	A=94
115200	A=10	A=10, B=15 A=14, B=11	A=190
57600	A=22	A=22, B=15 A=30, B=11	A=382
38400	A=34	A=62, B=8 A=46, B=11 A=34, B=15	A=574
19200	A=70	A=126, B=8 A=94, B=11 A=70, B=15	A=1150
9600	A=142	A=254, B=8 A=190, B=11 A=142, B=15	A=2302
4800	A=286	A=510, B=8 A=382, B=11 A=286, B=15	A=4606

Figure 6.10-7 Controller Baud Rate Parameter Setting Table

UART Peripheral Clock = 22.1184 MHz			
Baud Rate	**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

Figure 6.10-8 Controller Baud Rate Register Setting Table

6.10.5.2 UART Controller FIFO Control and Status

The UART0 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 UART0 at any time during operation. The reported status information includes the type and condition of the transfer operations being performed by the UART0, as well as 3 error conditions (parity error, framing error, break interrupt) probably occur while receiving data. This FIFO control and status also support all of UART data transmit, receive, IrDA, and RS-485 function mode.

The UART1 Controller is built-in with a 1 byte transmitter buffer and a 1 byte receiver buffer. The CPU can read the status of the UART1 at any time during operation. The reported status information includes the type and condition of the transfer operations being performed by the UART1, as well as 3 error conditions (parity error, framing error, break interrupt) probably occur while receiving data.

6.10.5.3 UART Controller Wake-up Function

Only UART0 support CTS Wakeup function. When the chip is in Power-down mode, an external CTS change will wake-up chip from Power-down mode. This wake-up function is available in every function mode. User must enable the WKCTSIEN (Wake-up CPU Interrupt Function Enable Control) to use the wake-up function. If WKCTSIEN (UART_INTEN[9]) is enabled and CTS is activate, chip can be wake-up from Power-down mode. The CTSWKIF (UART_INTSTS[16]) is set to 1 to indicate chip wakeup by external CTS trigger. Once the software enters Interrupt Service Routine, the CTSWKIF (UART_INTSTS[16]) should be cleared. The UART1 has no wake-up function.

6.10.5.4 UART Controller Interrupt and Status

UART0 Controller supports 6 types of interrupts and UART1 Controller supports 5 types of interrupts. The interrupts are listed as following:

- Receiver threshold level reached interrupt (RDAINT)
- Transmitter FIFO empty interrupt (THREINT)

- Line status interrupt (parity error, frame error or break interrupt) (RLSINT)
- MODEM/Wake-up status interrupt (MODEMINT) (Only Available in UART0)
- Receiver buffer time-out interrupt (RXTOINT)
- Buffer error interrupt (BUFERRINT)

Figure 6.10-9 describes the interrupt sources and flags. The interrupt is generated when the interrupt flag is generated and the interrupt enable bit is set. User must clear the interrupt flag after the interrupt is generated.

Interrupt Source	Interrupt Indicator	Interrupt Enable Bit	Interrupt Flag	Flag Cleared By
Receive Data Available Interrupt	RDAINT	RDAIEN	**RDAIF**	Read UART_DAT
Transmit Holding Register Empty Interrupt	THREINT	THREIEN	**THREIF**	Write UART_DAT
Receive Line Status Interrupt	RLSINT	RLSIEN	**RLSIF** = (BIF or FEF or PEF)	**Writing '1' to** BIF/FEF/ PEF
			RLSIF = ADDRDETF	**Writing '1' to** ADDRDETF
Modem Status Interrupt (Only Available in UART0)	MODEMINT	MODEMIEN	**MODEMIF** = CTSDETF	**Write '1' to** CTSDETF
RX Time-out Interrupt	RXTOINT	RXTOIEN	**RXTOIF**	Read UART_DAT
Buffer Error Interrupt	BUFERRINT	BUFERRIEN	**BUFERRIF** = (TXOVIF or RXOVIF)	**Writing '1' to** TXOVIF / RXOVIF

Figure 6.10-9 Controller Interrupt Source and Flag

6.10.5.5 UART Function Mode

The UART0/UART1 Controller provides UART function (user must set UART_FUNSEL [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 UART0 transmitter and receiver contain 16 bytes FIFO for payloads (Only Available in UART0). User can program receiver buffer trigger level and receiver buffer time-out detection for receiver (Only Available in UART0). The transmitting data delay time between the last stop and the next start bit can be programed by setting DLY (UART_TOUT [15:8]) register (Available in UART0/UART1). The UART0 supports hardware auto-flow control and flow control function (CTS, RTS), programmable RTS flow control trigger level and fully programmable serial-interface characteristics. (Only Available in UART0)

UART Line Control Function

The UART Controller supports fully programmable serial-interface characteristics by setting the UART_LINE register. Software can use the UART_LINE register to program the word length, stop bit and parity bit. Figure 6.10-10 lists the UART word and stop bit length settings and the UART 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

Figure 6.10-10 Line Control of Word and Stop Length Setting

Parity Type	SPE (UART_LINE[5])	EPE (UART_LINE[4])	PBE (UART_LINE[3])	Description
No Parity	x	x	0	No parity bit output.
Odd Parity	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	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	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	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).

Figure 6.10-11 Line Control of Parity Bit Setting

UART Auto-Flow Control Function (Only Available in UART0)

The UART0 supports auto-flow control function that uses two signals, CTS (clear-to-send) and RTS (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 RTS to external device. When the number of bytes in the RX FIFO equals the value of RTSTRGLV (UART_FIFO [19:16]), the RTS is de-asserted. The UART0 sends data out when UART0 detects CTS is asserted from external device. If the valid asserted CTS is not detected, the UART0 will not send data out.

Figure 6.10-12 demonstrates the CTS auto flow control of UART0 function mode. User must set ATOCTSEN (UART_INTEN [13]) to enable CTS auto flow control function. The CTSACTLV (UART_MODEM [8]) can set CTS pin input active state. The CTSDETF (UART_MODEMSTS [0]) is set when any state change of CTS pin input has occurred, and then TX data will be automatically transmitted from TX FIFO.

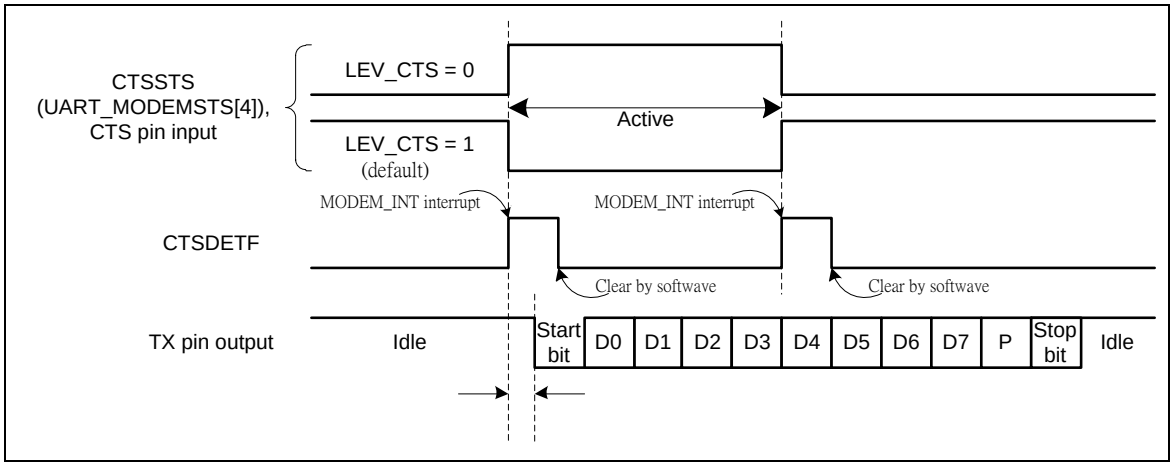


Figure 6.10-12 CTS Auto Flow Control Enabled

As shown in Figure 6.10-13, in UART0 RTS Auto Flow control mode (ATORTSEN(UART_INTEN[12])=1), the nRTS internal signal is controlled by UART0 FIFO controller with RTS_RTI_LEV(UART_FIFO[19:16]) trigger level.

Setting RTSACTLV(UART_MODEM[9]) can control the RTS pin output is inverse or non-inverse from nRTS signal. User can read the RTSSTS(UART_MODEM[13]) bit to get real RTS pin output voltage logic status.

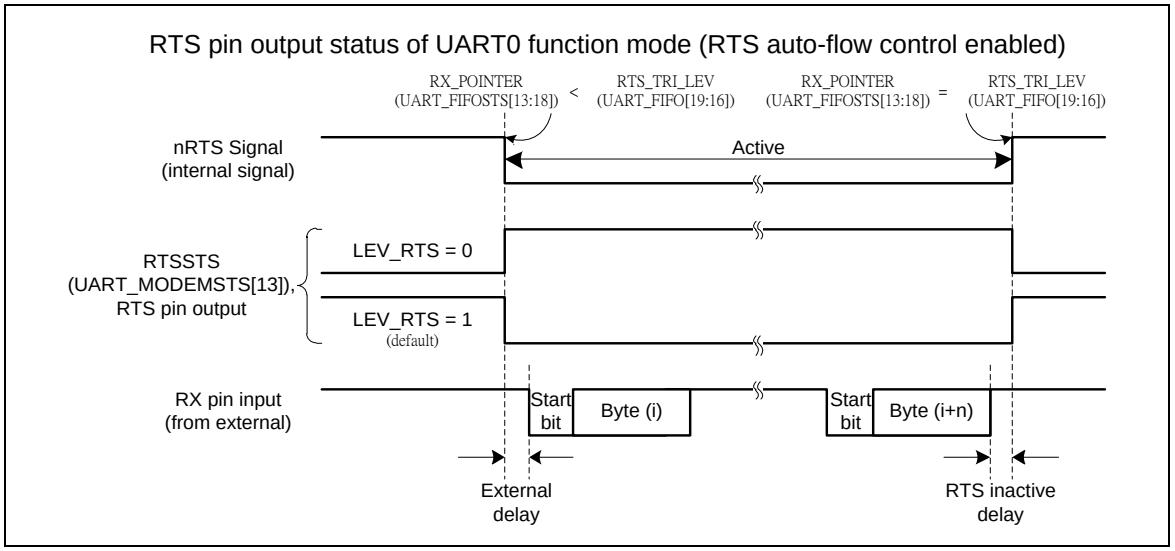


Figure 6.10-13 RTS Auto Flow Control Enabled

As shown in Figure 6.10-14, in software mode (ATORTSEN(UART_INTEN[12])=0) the RTS flow is directly controlled by software programming of RTS(UART_MODEM[1]) control bit.

Setting RTSACTLV(UART_MODEM[9]) can control the RTS 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 RTS pin output voltage logic status.

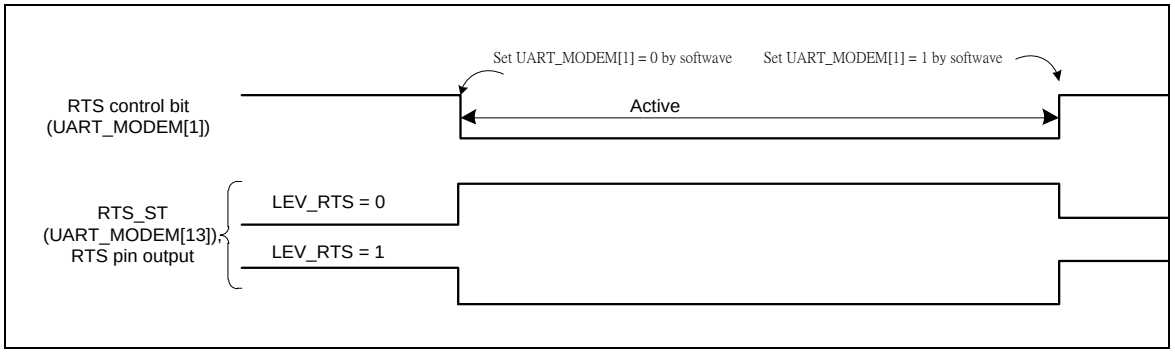


Figure 6.10-14 RTS Flow with Software Control

6.10.5.6 IrDA Function (Only Available in UART0)

The UART0 controller also provides Serial IrDA (SIR, Serial Infrared) function (user must set IrDA_EN (UART_FUNSEL[1:0]) to '10' to enable 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 (half duplex). The IrDA SIR block contains an IrDA SIR Protocol encoder/decoder. The IrDA SIR protocol is half-duplex only. Thus it cannot transmit and receive data at the same time. The IrDA SIR physical layer specifies a minimum 10ms transfer delay between transmission and reception. This delay feature must be implemented by software.

In IrDA mode, the BAUDM1 (UART_BAUD [29]) register must be disabled.

Baud Rate = Clock / (16 * BRD), where BRD is Baud Rate Divider in UART_BAUD register.

Figure 6.10-15 demonstrates the IrDA control block diagram.

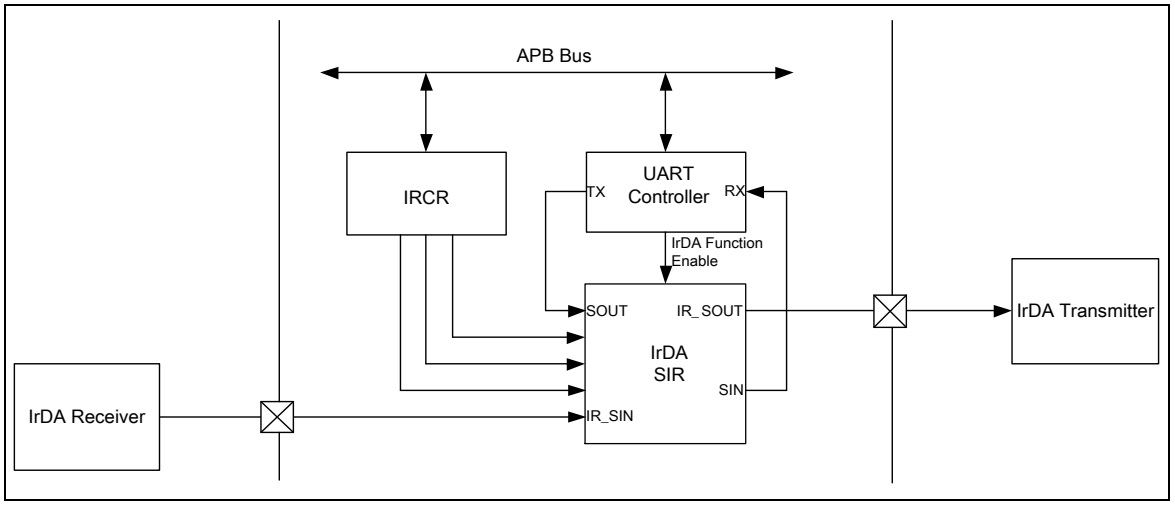


Figure 6.10-15 IrDA Control Block Diagram

IrDA SIR Transmit Encoder

The IrDA SIR Transmit Encoder modulate Non-Return-to Zero (NRZ) transmit bit stream output from UART0. The IrDA SIR physical layer specifies use of Return-to-Zero, Inverted (RZI) modulation scheme which represent logic 0 as an infra light pulse. The modulated output pulse stream is transmitted to an external output driver and infrared light emitting diode.

In Normal mode, 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 UART0 received data input. The decoder input is normally high in the idle state. (Because of this, RXINV (UART_IRCR[6]) should be set as 1 by default.)

A start bit is detected when the decoder input is LOW.

IrDA SIR Operation

The IrDA SIR Encoder/decoder provides functionality which converts between UART data stream and half duplex serial SIR interface. Figure 6.10-16 is IrDA encoder/decoder waveform:

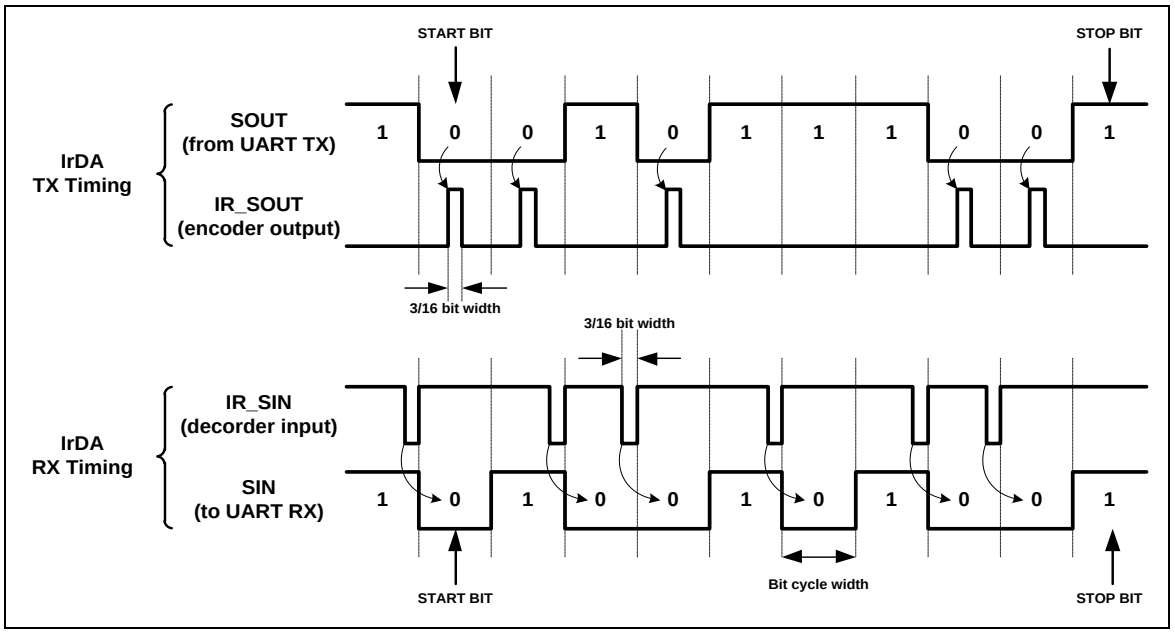


Figure 6.10-16 IrDA TX/RX Timing Diagram

6.10.5.7 RS-485 Function Mode (Only Available in UART0)

Another alternate function of UART Controller is RS-485 function (user must set UART_FUNSEL [1:0] to '11' to enable RS-485 function), and direction control provided by RTS pin or can program GPIO (P0.3 for RTS0 and P0.1 for RTS1) to implement the function by software. The RS-485 transceiver control is implemented by using the RTS control signal from an asynchronous serial port to enable the RS-485 driver. Many characteristics of the RX and TX are same as UART in RS-485 mode

The controller can configuration of it 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 Multi-drop 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 Multi-drop Operation Mode (NMM)

In RS-485 Normal Multi-drop Operation Mode, at 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 RS485_NMM (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 RS485_NMM (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)

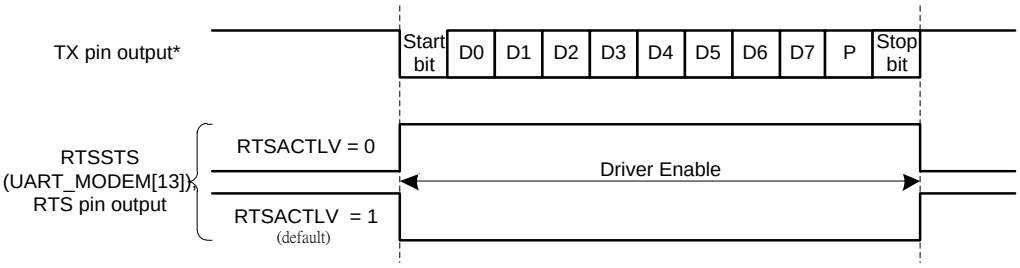
In RS-485 Auto Address Detection Operation Mode, the receiver will ignore any data until an address byte is detected (bit 9 = 1) and the address byte data matches the ADDRMV (UART_ALTCTL[31:24]) value. The address byte data will be stored in the RX FIFO. The all received byte data will be accepted and stored in the RX FIFO until and address byte data not match the ADDRMV (UART_ALTCTL[31:24]) value.

RS-485 Auto Direction Mode (AUD)

Another option function of RS-485 controllers is RS-485 auto direction control function. User must set RS485_AUD(UART_ALTCTL[10]) to 1 to enabled RS-485 auto direction mode. The RS-485 transceiver control is implemented using the RTS control signal from an asynchronous serial port. The RTS line is connected to the RS-485 transceiver enable pin such that setting the RTS line to high (logic 1) enables the RS-485 transceiver. Setting the RTS 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 RTS driving level.

Figure 6.10-17 demonstrates the RS-485 RTS driving level in AUD mode. The RTS pin will be automatically driven during TX data transmission.

Setting RTSACTLV(UART_MODEM[9]) can control RTS pin output driving level. User can read the RTSSTS(UART_MODEM[13]) bit to get real RTS pin output voltage logic status.

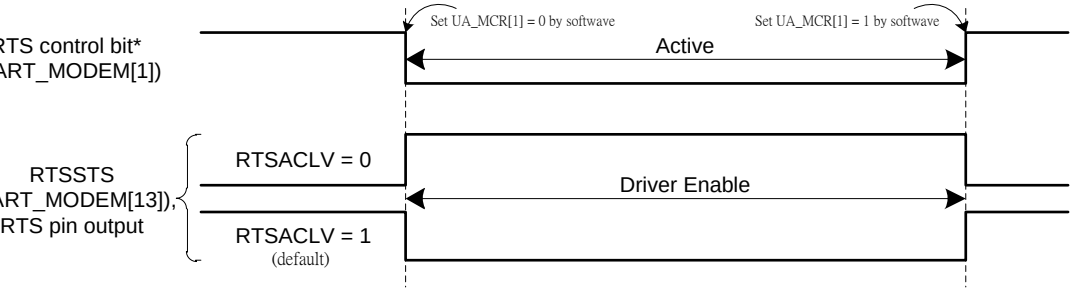


Note: RS485_AUD(UART_ALTCTL[10]) = 1, the RTS pin output by hardware control only.

Figure 6.10-17 RS-485 RTS Driving Level in Auto Direction Mode

Figure 6.10-18 demonstrates the RS-485 RTS driving level in software control (RS485_AUD(UART_ALTCTL[10])=0). The RTS driving level is controlled by programing the RTS(UART_MODEM[1]) control bit.

Setting RTSACTLV(UART_MODEM[9]) can control the RTS 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 RTS pin output voltage logic status.



Note: RS485_AUD(UART_ALTCTL[10]) = 0, support RTS control bit by software control only.

Figure 6.10-18 RS-485 RTS Driving Level with Software Control

Program Sequence Example:

1. Program UART_FUNSEL (UART_FUNSEL [1:0]) to select RS-485 function.
2. Program the RXOFF (UART_FIFO[8]) bit to determine enable or disable RS-485 receiver.
3. Program the RS-485_NMM (UART_ALTCTL[8])or RS-485_AAD mode.
4. If the RS-485_AAD (UART_ALTCTL[9]) mode is selected, the ADDRMV(UART_ALTCTL[31:24]) is programmed for auto address match value.
5. Determine auto direction control by programming RS-485_AUD (UART_ALTCTL[10]).

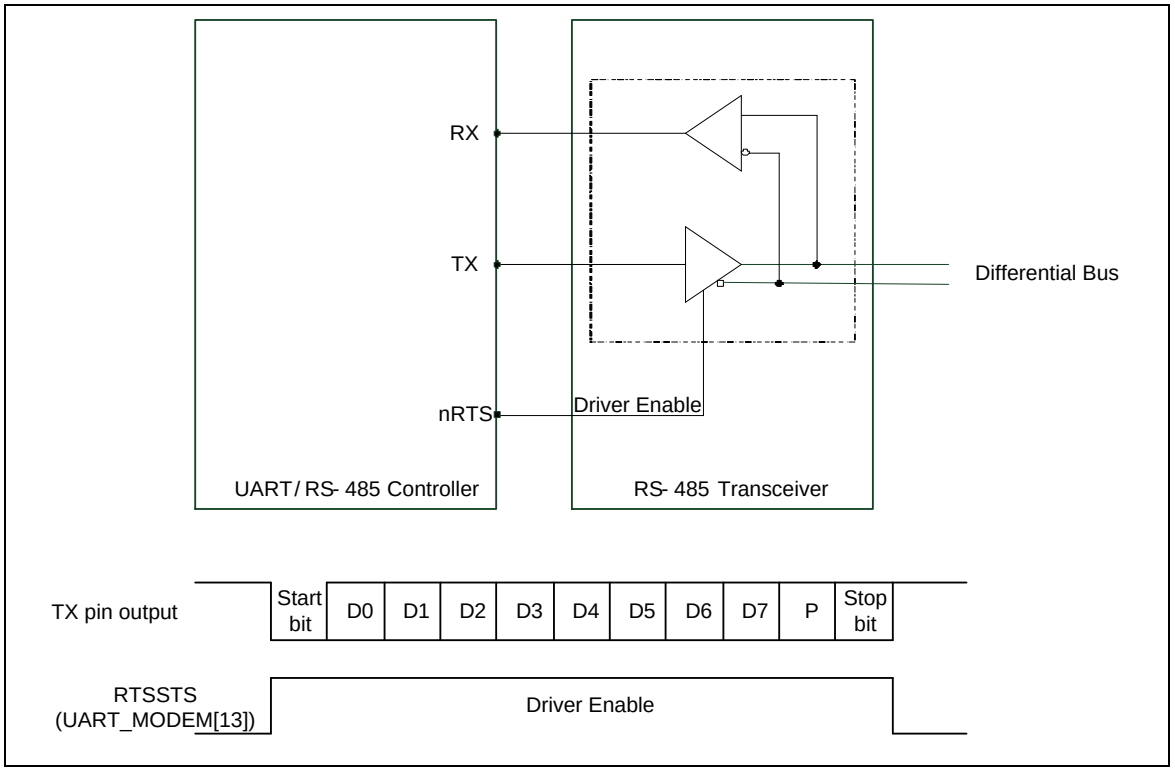


Figure 6.10-19 Structure of RS-485 Frame

6.10.6 Register Map

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

Register	Offset	R/W	Description	Reset Value
UART Base Address: **UART0_BA = 0x4005_0000** **UART1_BA = 0x4015_0000**				
UART_DAT **x = 0, 1**	UARTx_BA+0x00	R/W	UART Receive/Transmit Buffer Register	Undefined
UART_INTEN **x = 0, 1**	UARTx_BA+0x04	R/W	UART Interrupt Enable Register	0x0000_0000
UART_FIFO **x = 0, 1**	UARTx_BA+0x08	R/W	UART FIFO Control Register	0x0000_0101
UART_LINE **x = 0, 1**	UARTx_BA+0x0C	R/W	UART Line Control Register	0x0000_0000
UART_MODEM **x = 0**	UARTx_BA+0x10	R/W	UART Modem Control Register	0x0000_0200
UART_MODEMSTS **x = 0**	UARTx_BA+0x14	R/W	UART Modem Status Register	0x0000_0110
UART_FIFOSTS **x = 0, 1**	UARTx_BA+0x18	R/W	UART FIFO Status Register	0x1040_4000
UART_INTSTS **x = 0, 1**	UARTx_BA+0x1C	R/W	UART Interrupt Status Register	0x0000_0002
UART_TOUT **x = 0, 1**	UARTx_BA+0x20	R/W	UART Time-out Register	0x0000_0000
UART_BAUD **x = 0, 1**	UARTx_BA+0x24	R/W	UART Baud Rate Divisor Register	0x0F00_0000
UART_IRDA **x = 0**	UARTx_BA+0x28	R/W	UART IrDA Control Register	0x0000_0040
UART_ALTCTL **x = 0**	UARTx_BA+0x2C	R/W	UART Alternate Control/Status Register	0x0000_0000
UART_FUNSEL **x = 0, 1**	UARTx_BA+0x30	R/W	UART Function Select Register	0x0000_0000

6.10.7 Register Description

Receive/Transmit Buffer Register (UART_DAT)

Register	Offset	R/W	Description	Reset Value
UART_DAT **x = 0, 1**	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							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DAT							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**DAT**	**Receiving/Transmit Buffer** Write Operation: By writing one byte to this register, the data byte will be stored in transmitter FIFO. The UART Controller will send out the data stored in transmitter FIFO top location through the UART_TXD. Read Operation: By reading this register, the UART will return an 8-bit data received from receiving FIFO.

Interrupt Enable Register (UART_INTEN)

Register	Offset	R/W	Description	Reset Value
UART_INTEN **x = 0, 1**	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							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		**ATOCTSEN**	**ATORTSEN**	**TOCNTEN**	**Reserved**	**WKCTSIEN**	**Reserved**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**BUFERRIEN**	**RXTOIEN**	**MODEMIEN**	**RLSIEN**	**THREIEN**	**RDAIEN**

Bits	Description	
[31:14]	**Reserved**	Reserved.
[13]	**ATOCTSEN**	**CTS Auto Flow Control Enable Bit (Only Available In UART0)** 0 = CTS auto flow control Disabled. 1 = CTS auto flow control Enabled. **Note:** When CTS auto-flow is enabled, the UART will send data to external device when CTS input assert (UART will not send data to device until CTS is asserted).
[12]	**ATORTSEN**	**RTS Auto Flow Control Enable Bit (Only Available In UART0)** 0 = RTS auto flow control Disabled. 1 = RTS auto flow control Enabled. **Note:** When RTS 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 RTS signal.
[11]	**TOCNTEN**	**Time-out Counter Enable Bit** 0 = Time-out counter Disabled. 1 = Time-out counter Enabled.
[10]	**Reserved**	Reserved.
[9]	**WKCTSIEN**	**Wake-up CPU Function Interrupt Enable Bit (Only Available In UART0)** 0 = UART wake-up function Disabled. 1 = UART Wake-up function Enabled. **Note:** When the chip is in Power-down mode, an external CTS change will wake-up chip from Power-down mode.
[8:6]	**Reserved**	Reserved.
[5]	**BUFERRIEN**	**Buffer Error Interrupt Enable Bit** 0 = Buffer Error Interrupt Masked Disabled. 1 = Buffer Error Interrupt Masked Enabled.
[4]	**RXTOIEN**	**RX Time-out Interrupt Enable Bit** 0 = RXTOINT Masked off.

Bits	Description	
		1 = RXTOINT Enabled.
[3]	**MODEMIEN**	**Modem Status Interrupt Enable Bit** 0 = MODEMINT Masked off. 1 = MODEMINT Enabled.
[2]	**RLSIEN**	**Receive Line Status Interrupt Enable Bit** 0 = RLSINT Masked off. 1 = RLSINT Enabled.
[1]	**THREIEN**	**Transmit Holding Register Empty Interrupt Enable Bit** 0 = THREINT Masked off. 1 = THREINT Enabled.
[0]	**RDAIEN**	**Receive Data Available Interrupt Enable Bit** 0 = RDAINT Masked off. 1 = RDAINT Enabled.

FIFO Control Register (UART_FIFO)

Register	Offset	R/W	Description	Reset Value
UART_FIFO **x = 0, 1**	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**	**RTS Trigger Level (For Auto-flow Control Use) (Only Available In UART0)** 0000 = RTS Trigger Level is 1 byte. 0001 = RTS Trigger Level is 4 bytes. 0010 = RTS Trigger Level is 8 bytes. 0011 = RTS Trigger Level is 14 bytes. Other = Reserved. **Note:** This field is used for RTS auto-flow control.
[15:9]	**Reserved**	Reserved.
[8]	**RXOFF**	**Receiver Disable Register (Only Available In UART0)** The receiver is disabled or not (setting 1 to disable the receiver). 0 = Receiver Enabled. 1 = Receiver Disabled. **Note:** This field is used for RS-485 Normal Multi-drop mode. It should be programmed before RS-485_NMM (UART_ALTCTL [8]) is programmed.
[7:4]	**RFITL**	**RX FIFO Interrupt (RDAINT) Trigger Level (Only Available In UART0)** When the number of bytes in the receive FIFO equals the RFITL then the RDAIF will be set (if RDAIEN in UART_INTEN register is enable, an interrupt will generated). 0000 = RX FIFO Interrupt Trigger Level is 1 byte. 0001 = RX FIFO Interrupt Trigger Level is 4 bytes. 0010 = RX FIFO Interrupt Trigger Level is 8 bytes. 0011 = RX FIFO Interrupt Trigger Level is 14 bytes. Other = Reserved.
[3]	**Reserved**	Reserved.
[2]	**TXRST**	**TX Field Software Reset** When TX_RST is set, all the byte in the transmit FIFO and TX internal state machine are

Bits	Description	
		cleared. 0 = No effect. 1 = Reset TX internal state machine and pointers reset. **Note:** This bit will auto clear needs at least 3 UART Controller peripheral clock cycles.
[1]	**RXRST**	**RX Field Software Reset** When RX_RST is set, all the byte in the receiver FIFO and RX internal state machine are cleared. 0 = No effect. 1 = Reset RX internal state machine and pointers reset. **Note:** This bit will auto clear needs at least 3 UART Controller peripheral clock cycles.
[0]	**Reserved**	Reserved.

Line Control Register (UART_LINE)

Register	Offset	R/W	Description	Reset Value
UART_LINE **x = 0, 1**	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							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**BCB**	**SPE**	**EPE**	**PBE**	**NSB**	**WLS**	

Bits	Description	
[31:7]	**Reserved**	Reserved.
[6]	**BCB**	**Break Control Bit** 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 and has no effect on the transmitter logic. 0 = Break control Disabled. 1 = Break control Enabled.
[5]	**SPE**	**Stick Parity Enable Bit** 0 = Stick parity Disabled. 1 = If PBE (UART_LINE[3]) and EPE (UART_LINE[4]) are logic 1, the parity bit is transmitted and checked as logic 0. If PBE (UART_LINE[3]) is 1 and EPE (UART_LINE[4]) is 0 then the parity bit is transmitted and checked as 1.
[4]	**EPE**	**Even Parity Enable Bit** 0 = Odd number of logic 1's is transmitted and checked in each word. 1 = Even number of logic 1's is transmitted and checked in each word. This bit has effect only when PBE (UART_LINE[3]) is set.
[3]	**PBE**	**Parity Bit Enable Bit** 0 = No parity bit. 1 = Parity bit is generated on each outgoing character and is checked on each incoming data.
[2]	**NSB**	**Number of "STOP Bit"** 0 = One "STOP bit" is generated in the transmitted data. 1 = When select 5-bit word length, 1.5 "STOP bit" is generated in the transmitted data. When select 6-, 7- and 8-bti word length, 2 "STOP bit" is generated in the transmitted data.
[1:0]	**WLS**	**Word Length Select Bit** 00 = Word length is 5-bit. 01 = Word length is 6-bit.

Bits	Description	
		10 = Word length is 7-bit. 11 = Word length is 8-bit.

MODEM Control Register (UART_MODEM) (Only Available in UART0)

Register	Offset	R/W	Description	Reset Value
UART_MODEM x = 0	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**	**RTS Pin State (Read Only)** This bit mirror from RTS pin output of voltage logic status. 0 = RTS pin output is low level voltage logic state. 1 = RTS pin output is high level voltage logic state.
[12:10]	**Reserved**	Reserved.
[9]	**RTSACTLV**	**RTS Pin Active Level** This bit defines the active level state of RTS pin output. 0 = RTS pin output is high level active. 1 = RTS pin output is low level active. (Default) **Note1:** Refer to Figure 6.10-13 and Figure 6.10-14 UART function mode. **Note2:** Refer to Figure 6.10-17 and Figure 6.10-18 for RS-485 function mode.
[8:2]	**Reserved**	Reserved.
[1]	**RTS**	**RTS (Request-to-send) Signal Control** This bit is direct control internal RTS signal active or not, and then drive the RTS pin output with RTSACTLV bit configuration. 0 = RTS signal is active. 1 = RTS signal is inactive. **Note1:** This RTS signal control bit is not effective when RTS auto-flow control is enabled in UART function mode. **Note2:** This RTS signal control bit is not effective when RS-485 auto direction mode (AUD) is enabled in RS-485 function mode.
[0]	**Reserved**	Reserved.

Modem Status Register (UART_MODEMSTS) (Only Available in UART0)

Register	Offset	R/W	Description	Reset Value
UART_MODEMSTS **x = 0**	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**	**CTS Pin Active Level** This bit defines the active level state of CTS pin input. 0 = CTS pin input is high level active. 1 = CTS pin input is low level active. (Default)
[7:5]	**Reserved**	Reserved.
[4]	**CTSSTS**	**CTS Pin Status (Read Only)** This bit mirror from CTS pin input of voltage logic status. 0 = CTS pin input is low level voltage logic state. 1 = CTS pin input is high level voltage logic state. **Note:** This bit echoes when UART Controller peripheral clock is enabled, and CTS multi-function port is selected.
[3:1]	**Reserved**	Reserved.
[0]	**CTSDETF**	**Detect CTS State Change Flag** This bit is set whenever CTS input has change state, and it will generate Modem interrupt to CPU when MODEMIEN (UART_INTEN [3]) is set to 1. 0 = CTS input has not change state. 1 = CTS input has change state. **Note:** This bit is cleared by writing 1 to it.

FIFO Status Register (UART_FIFOSTS)

Register	Offset	R/W	Description	Reset Value
UART_FIFOSTS x = 0, 1	UARTx_BA+0x18	R/W	UART FIFO Status Register	0x1040_4000

31	30	29	28	27	26	25	24
Reserved			TXEMPTYF	Reserved			TXOVIF
23	22	21	20	19	18	17	16
TXFULL	TXEMPTY	TXPTR					
15	14	13	12	11	10	9	8
RXFULL	RXEMPTY	RXPTR					
7	6	5	4	3	2	1	0
Reserved	BIF	FEF	PEF	ADDRDETF	Reserved		RXOVIF

Bits	Description	
[31:29]	**Reserved**	Reserved.
[28]	**TXEMPTYF**	**Transmitter Empty Flag (Read Only)** This bit is set by hardware when TX FIFO (UART_DAT) is empty and the STOP bit of the last byte has been transmitted. 0 = TX FIFO is not empty. 1 = TX FIFO is empty. **Note:** This bit is cleared automatically when TX FIFO is not empty or the last byte transmission has not completed.
[27:25]	**Reserved**	Reserved.
[24]	**TXOVIF**	**TX Overflow Error Interrupt Flag** If TX FIFO (UART_DAT) is full, an additional write to UART_DAT will cause this bit to logic 1. 0 = TX FIFO did not overflow. 1 = TX FIFO overflowed. **Note:** This bit is cleared by writing 1 to it.
[23]	**TXFULL**	**Transmitter FIFO Full (Read Only)** This bit indicates TX FIFO full or not. 0 = TX FIFO is not full. 1 = TX FIFO is full. **Note:** This bit is set when the number of usage in TX FIFO Buffer is equal to 16, otherwise is cleared by hardware.
[22]	**TXEMPTY**	**Transmitter FIFO Empty (Read Only)** This bit indicates TX FIFO empty or not. 0 = TX FIFO is not empty. 1 = TX FIFO is empty. **Note:** When the last byte of TX FIFO has been transferred to Transmitter Shift Register, hardware sets this bit high. It will be cleared when writing data into THR (TX FIFO not

Bits	Description	
		empty).
[21:16]	**TXPTR**	**TX FIFO Pointer (Read Only)** This field indicates the TX FIFO Buffer Pointer. When CPU writes one byte into UART_DAT, TXPTR increases one. When one byte of TX FIFO is transferred to Transmitter Shift Register, TXPTR decreases one. The Maximum value shown in TXPTR is 15. When the using level of TX FIFO Buffer equal to 16, the TXFULL bit is set to 1 and TXPTR will show 0. As one byte of TX FIFO is transferred to Transmitter Shift Register, the TXFULL bit is cleared to 0 and TXPTR will show 15.
[15]	**RXFULL**	**Receiver FIFO Full (Read Only)** This bit initiates RX FIFO full or not. 0 = RX FIFO is not full. 1 = RX FIFO is full. **Note:** This bit is set when the number of usage in RX FIFO Buffer is equal to 16, otherwise is cleared by hardware.
[14]	**RXEMPTY**	**Receiver FIFO Empty (Read Only)** This bit initiate RX FIFO empty or not. 0 = RX FIFO is not empty. 1 = RX FIFO is empty. **Note:** When the last byte of RX FIFO has been read by CPU, hardware sets this bit high. It will be cleared when UART receives any new data.
[13:8]	**RXPTR**	**RX FIFO Pointer (Read Only)** This field indicates the RX FIFO Buffer Pointer. When UART receives one byte from external device, RXPTR increases one. When one byte of RX FIFO is read by CPU, RXPTR decreases one. The Maximum value shown in RXPTR is 15. When the using level of RX FIFO Buffer equal to 16, the RXFULL bit is set to 1 and RXPTR will show 0. As one byte of RX FIFO is read by CPU, the RXFULL bit is cleared to 0 and RXPTR will show 15.
[7]	**Reserved**	Reserved.
[6]	**BIF**	**Break Interrupt Flag (Read Only)** This bit is set to logic 1 whenever the received data input (RX) is held in the “spacing state” (logic 0) for longer than a full word transmission time (that is, the total time of “start bit” + data bits + parity + stop bits). 0 = No Break interrupt is generated. 1 = Break interrupt is generated. **Note:** This bit is read only, but software can write 1 to clear it.
[5]	**FEF**	**Framing Error Flag (Read Only)** This bit is set to logic 1 whenever the received character does not have a valid “stop bit” (that is, the stop bit follows the last data bit or parity bit is detected as logic 0). 0 = No framing error is generated. 1 = Framing error is generated. **Note:** This bit is read only, but can be cleared by writing ‘1’ to it .
[4]	**PEF**	**Parity Error Flag (Read Only)** This bit is set to logic 1 whenever the received character does not have a valid “parity bit”. 0 = No parity error is generated. 1 = Parity error is generated.. **Note:** This bit is read only, but can be cleared by writing ‘1’ to it.
[3]	**ADDRDETF**	**RS-485 Address Byte Detection Flag (Only Available In UART0)** This bit is set to 1 while ADDRDEN (UART_ALTCTL[15]) is set to 1 to enable Address

Bits	Description	
		detection mode and receive detect a data with an address bit (bit 9 = 1). **Note1:** This field is used for RS-485 function mode. **Note2:** This bit is cleared by writing 1 to it.
[2:1]	**Reserved**	Reserved.
[0]	**RXOVIF**	**RX Overflow Error Interrupt Flag** This bit is set when RX FIFO overflow. If the number of bytes of received data is greater than RX_FIFO (UART_DAT) size, 16 bytes this bit will be set. 0 = RX FIFO did not overflow. 1 = RX FIFO overflowed. **Note:** This bit is cleared by writing 1 to it.

Interrupt Status Control Register (UART_INTSTS)

Register	Offset	R/W	Description	Reset Value
UART_INTSTS **x = 0, 1**	UARTx_BA+0x1C	R/W	UART Interrupt Status Register	0x0000_0002

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							**CTSWKIF**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		**BUFERRINT**	**RXTOINT**	**MODEMINT**	**RLSINT**	**THREINT**	**RDAINT**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**BUFERRIF**	**RXTOIF**	**MODEMIF**	**RLSIF**	**THREIF**	**RDAIF**

Bits	Description	
[31:17]	**Reserved**	Reserved.
[16]	**CTSWKIF**	**NCTS Wake-up Interrupt Flag (Read Only) (Only Available In UART0)** 0 = Chip stays in power-down state. 1 = Chip wake-up from power-down state by nCTS wake-up. **Note1:** If WKCTSIEN (UART_IER[9])is enabled, the wake-up interrupt is generated. **Note2:** This bit is read only, but can be cleared by writing '1' to it.
[15:14]	**Reserved**	Reserved.
[13]	**BUFERRINT**	**Buffer Error Interrupt Indicator (Read Only)** This bit is set if BUFERRIEN (UART_INTEN[5]) and BUFERRIF (UART_INTSTS[5]) are both set to 1. 0 = No buffer error interrupt is generated. 1 = buffer error interrupt is generated.
[12]	**RXTOINT**	**Time-out Interrupt Indicator (Read Only)** This bit is set if RXTOIEN (UART_INTEN[4]) and RXTOIF (UART_INTSTS[4]) are both set to 1. 0 = No Time-out interrupt is generated. 1 = Time-out interrupt is generated.
[11]	**MODEMINT**	**MODEM Status Interrupt Indicator (Read Only)** This bit is set if MODEMIEN (UART_INTEN[3]) and MODENIF (UART_INTSTS[3]) are both set to 1. 0 = No Modem interrupt is generated. 1 = Modem interrupt is generated.
[10]	**RLSINT**	**Receive Line Status Interrupt (Read Only)** This bit is set if RLSIEN (UART_INTEN[2]) and RLSIF (UART_INTSTS[2]) are both set to 1. 0 = No RLS interrupt is generated. 1 = RLS interrupt is generated.

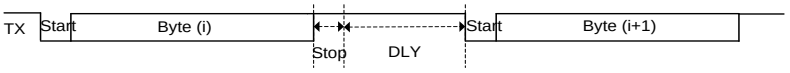
Bits	Description	
[9]	**THREINT**	**Transmit Holding Register Empty Interrupt Indicator (Read Only)** This bit is set if THREIEN (UART_INTEN[1]) and THREIF (UART_INTSTS[1]) are both set to 1. 0 = No THRE interrupt is generated. 1 = THRE interrupt is generated.
[8]	**RDAINT**	**Receive Data Available Interrupt Indicator (Read Only)** This bit is set if RDAIEN (UART_INTEN[0]) and RDAIF (UART_INTSTS[0]) are both set to 1. 0 = No RDA interrupt is generated. 1 = RDA interrupt is generated.
[7:6]	**Reserved**	Reserved.
[5]	**BUFERRIF**	**Buffer Error Interrupt Flag (Read Only)** This bit is set when the TX/RX FIFO overflow flag (TXOVIF (UART_FIFOSTS[24] or RXOVIF(UART_FIFOSTS[0])) is set. When BUFERRIF (UART_INTSTS[5]) is set, the transfer is not correct. If BUFERRIEN (UART_INTEN[5]) is enabled, the buffer error interrupt will be generated. 0 = No buffer error interrupt flag is generated. 1 = Buffer error interrupt flag is generated. **Note:** This bit is read only and reset to 0 when all bits of TXOVIF (UART_FIFOSTS[24]) and RXOVIF (UART_FIFOSTS[0]) are cleared.
[4]	**RXTOIF**	**Time-out Interrupt Flag (Read Only)** This bit is set when the RX FIFO is not empty and no activities occurred in the RX FIFO and the time-out counter equal to TOIC (UARTTOUT[7:0]). If RXTOIEN (UART_INTEN [4]) is enabled, the Tout interrupt will be generated. 0 = No Time-out interrupt flag is generated. 1 = Time-out interrupt flag is generated. **Note:** This bit is read only and user can read UART_DAT (RX is in active) to clear it.
[3]	**MODEMIF**	**MODEM Interrupt Flag (Read Only) (Only Available In UART0)** This bit is set when the CTS pin has state change (CTSDETF (UART_MODEMSTS[0]) = 1). If MODEMIEN (UART_INTEN[3]) is enabled, the Modem interrupt will be generated. 0 = No Modem interrupt flag is generated. 1 = Modem interrupt flag is generated. **Note:** This bit is read only and reset to 0 when bit CTSDETF (UART_MODEMSTS[0]) is cleared by a write 1 on CTSDETF (UART_MODEMSTS[0]).
[2]	**RLSIF**	**Receive Line Interrupt Flag (Read Only)** This bit is set when the RX receive data have parity error, framing error or break error (at least one of 3 bits, BIF, FEF and PEF, is set). If RLSIEN (UART_INTEN [2]) is enabled, the RLS interrupt will be generated. 0 = No RLS interrupt flag is generated. 1 = RLS interrupt flag is generated. **Note1:** In RS-485 function mode, this field is set including “receiver detects and receives address byte character (bit 9 = 1) bit". At the same time, the bit of ADDRDETF (UART_FIFOSTS[3]) is also set. **(Only Available in UART0)** **Note2:** 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. **(Only Available in UART0)**
[1]	**THREIF**	**Transmit Holding Register Empty Interrupt Flag (Read Only)** 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.

Bits	Description	
		0 = No THRE interrupt flag is generated. 1 = THRE interrupt flag is generated. **Note:** This bit is read only and it will be cleared when writing data into UART_DAT (TX FIFO not empty).
[0]	**RDAIF**	**Receive Data Available Interrupt Flag (Read Only)** When the number of bytes in the RX FIFO equals the RFITL(UART_FIFO[7:4]) then the RDAIF(UART_INTSTS[0]) will be set. If RDAIEN (UART_INTEN [0]) is enabled, the RDA interrupt will be generated. 0 = No RDA interrupt flag is generated. 1 = RDA interrupt flag is generated. **Note:** This bit is read only and it will be cleared when the number of unread bytes of RX FIFO drops below the threshold level (RFITL (UART_FIFO[7:4])).

Time-out Register (UART_TOUT)

Register	Offset	R/W	Description	Reset Value
UART_TOUT **x = 0, 1**	UARTx_BA+0x20	R/W	UART Time-out Register	0x0000_0000

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

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:8]	**DLY**	**TX Delay Time Value** This field is used to program the transfer delay time between the last stop bit and next start bit. The Unit is bit time. 
[7:0]	**TOIC**	**Time-out Interrupt Comparator** The time-out counter resets and starts counting (the counting clock = baud rate) whenever the RX FIFO receives a new data word. Once the content of time-out counter (TOUT_CNT) is equal to that of time-out interrupt comparator (TOIC), a receiver time-out interrupt (RXTOINT) is generated if RXTOIEN (UART_INTEN[4]) is enabled. A new incoming data word or RX FIFO empty clears RXTOINT. 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.

Baud Rate Divider Register (UART_BAUD)

Register	Offset	R/W	Description	Reset Value
UART_BAUD **x = 0, 1**	UARTx_BA+0x24	R/W	UART Baud Rate Divisor 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**	**Divider X Enable Bit** The BRD = Baud Rate Divider, and the baud rate equation is: Baud Rate = Clock / [M * (BRD + 2)], The default value of M is 16. 0 = Divider X Disabled (the equation of M = 16). 1 = Divider X Enabled (the equation of M = X+1, but EDIVM1 [27:24] must >= 8). **Note:** In IrDA mode, this bit must be disabled.
[28]	**BAUDM0**	**Divider X Equal 1** 0 = Divider M = X (the equation of M = X+1, but EDIVM1[27:24] must >= 8). 1 = Divider M = 1 (the equation of M = 1, but BRD [15:0] must >= 8). **Note:** Refer to section “UART Controller Baud Rate Generator” for more information.
[27:24]	**EDIVM1**	**Divider X** The baud rate divider M = X+1.
[23:16]	**Reserved**	Reserved.
[15:0]	**BRD**	**Baud Rate Divider** The field indicates the baud rate divider.

IrDA Control Register (UART_IRDA) (Only Available in UART0)

Register	Offset	R/W	Description	Reset Value
UART_IRDA **x = 0**	UARTx_BA+0x28	R/W	UART IrDA Control Register	0x0000_0040

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

Bits	Description	
[31:7]	**Reserved**	Reserved.
[6]	**RXINV**	**IrDA Inverse Receive Input Signal** 0 = None inverse receiving input signal. 1 = Inverse receiving input signal. (Default)
[5]	**TXINV**	**IrDA Inverse Transmitting Output Signal** 0 = None inverse transmitting signal. (Default) 1 = Inverse transmitting output signal.
[4:2]	**Reserved**	Reserved.
[1]	**TXEN**	**IrDA Receiver/Transmitter Selection Enable Bit** 0 = IrDA Transmitter Disabled and Receiver Enabled. (Default) 1 = IrDA Transmitter Enabled and Receiver Disabled.
[0]	**Reserved**	Reserved.

Note: In IrDA mode, the BAUDM1 (UART_BAUD[29]) egister must be disabled, the baud equation must be Clock/16*(BRD+2).

UART Alternate Control/Status Register (UART_ALTCTL) (Only Available in UART0)

Register	Offset	R/W	Description	Reset Value
UART_ALTCTL x = 0	UARTx_BA+0x2C	R/W	UART Alternate Control/Status Register	0x0000_0000

31	30	29	28	27	26	25	24
ADDRMV							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
ADDRDEN	Reserved				RS485_AUD	RS485_AAD	RS485_NMM
7	6	5	4	3	2	1	0
Reserved							

Bits	Description	
[31:24]	**ADDRMV**	**Address Match Values** This field contains the RS-485 address match values. **Note:** This field is used for RS-485 auto address detection mode.
[23:16]	**Reserved**	Reserved.
[15]	**ADDRDEN**	**RS-485 Address Detection Enable Bit** This bit is used to enable RS-485 Address Detection mode. 0 = RS-485 address detection mode Disabled. 1 = RS-485 address detection mode Enabled. **Note:** This field is used for RS-485 any operation mode.
[14:11]	**Reserved**	Reserved.
[10]	**RS485_AUD**	**RS-485 Auto Direction Mode (AUD) Control** 0 = RS-485 Auto Direction Function Mode (AUD) Disabled. 1 = RS-485 Auto Direction Function Mode (AUD) Enabled. **Note:** It cannot be active with RS485_NMM operation mode.
[9]	**RS485_AAD**	**RS-485 Auto Address Detection Operation Mode (AAD)** 0 = RS-485 Auto Address Detection Operation Mode (AAD) Disabled. 1 = RS-485 Auto Address Detection Operation Mode (AAD) Enabled. **Note:** It cannot be active with RS485_NMM operation mode.
[8]	**RS485_NMM**	**RS-485 Normal Multi-drop Operation Mode (NMM) Control** 0 = RS-485 Normal Multi-drop Operation Mode (NMM) Disabled. 1 = RS-485 Normal Multi-drop Operation Mode (NMM) Enabled. **Note:** It cannot be active with RS485_AAD operation mode.
[7:0]	**Reserved**	Reserved.

UART Function Select Register (UART_FUNSEL)

Register	Offset	R/W	Description	Reset Value
UART_FUNSEL **x = 0, 1**	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						**FUN_SEL**	

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1:0]	**FUN_SEL**	**Function Selection** 00 = UART function mode. 01 = Reserved. 10 = IrDA function mode. (Only Available in UART0) 11 = RS-485 function mode. (Only Available in UART0)

6.11 I²C Serial Interface Controller (I²C)

6.11.1 Overview

I²C is a two-wire, bi-directional serial bus that provides a simple and efficient method for data exchange between devices. The I²C standard is a true multi-master bus including collision detection and arbitration that prevents data corruption if two or more masters attempt to control the bus simultaneously. There are two sets of I²C controller and only I²C0 supports Power-down wake-up function.

6.11.2 Features

The I²C bus uses two wires (SDA and SCL) to transfer information between devices connected to the bus. The main features of the bus include:

- Supports up to two I²C ports
- Master/Slave mode
- Bi-directional data transfer between masters and slaves
- Multi-master bus
- Arbitration between simultaneously transmitting masters without corruption of serial data on the bus
- Serial clock synchronization allowing 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
- Built-in 14-bit time-out counter that requests the I²C interrupt if the I²C bus hangs up and timer-out counter overflows
- Programmable clocks allowing for versatile rate control
- Supports 7-bit addressing mode
- Supports multiple address recognition (four slave address registers with mask option)
- Supports Power-down wake-up function (Only I²C0 channel support this function)
- Supports two-level buffer function

6.11.3 Basic Configuration

The basic configurations of I²C0 are as follows:

- I²C pins are configured on SYS_P3_MFP [13:12] register.
- Enable I²C clock (I2CCKEN) on CLK_APBCLK [8] register.
- Reset I²C controller (I2CRST) on SYS_IPRST1 [8] register.

The basic configurations of I2C1 are as follows:

- I²C pins are configured on SYS_P2_MFP [11:10] or SYS_P5_MFP [9:8] register.

- Enable I²C clock (I2CCKEN) on CLK_APBCLK [9] register.
- Reset I²C controller (I2CRST) on SYS_IPRST1 [9] register.

6.11.4 Block Diagram

The block diagram of I²C controller is shown in Figure 6.11-1.

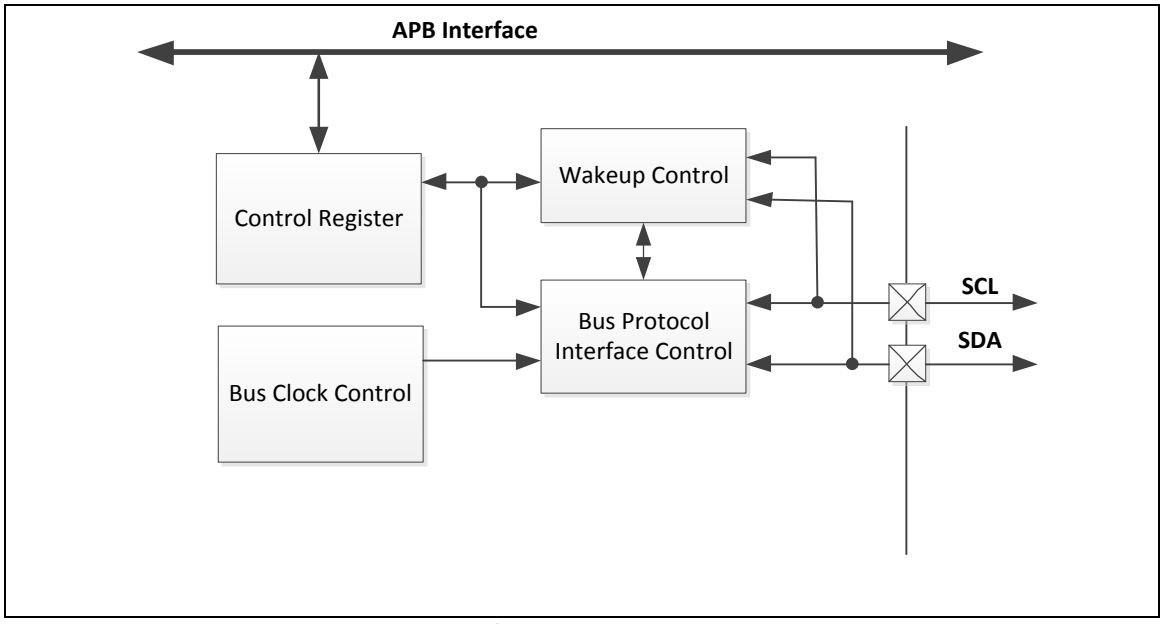


Figure 6.11-1 I²C Controller Block Diagram

6.11.5 Functional Description

On I²C bus, data is transferred between a Master and a Slave. Data bits transfer on the SCL and SDA lines are synchronous 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 I²C Bus Timing.

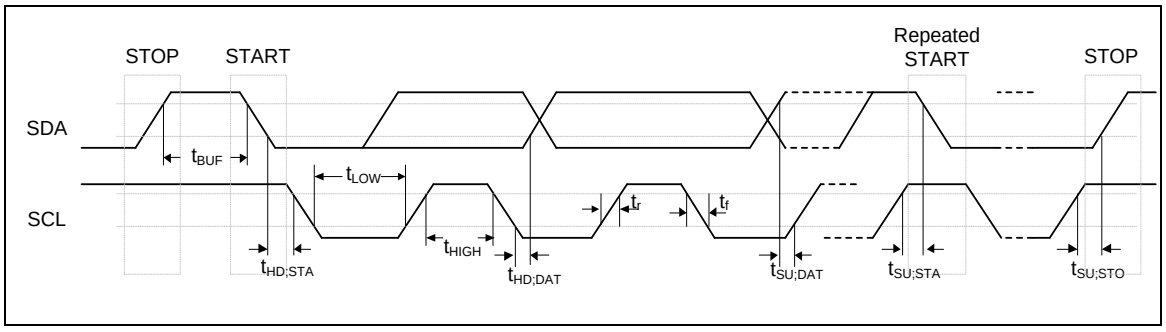


Figure 6.11-2 I²C Bus Timing

The device's on-chip I²C provides the serial interface that meets the I²C bus standard mode specification. The I²C port handles byte transfers autonomously. To enable this port, the bit I2CEN in I2C_CTL should be set to '1'. The I²C hardware interfaces to the I²C bus via two pins: SDA and SCL. When I/O pins are used as I²C ports, user must set the pins function to I²C in advance.

There is a two-level buffer to improve the performance of I²C 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.

The I²C SCL bus is stretched low when there is SI event. The NSTRETCH control bit is used to force the I²C SCL bus is no stretched under the SI event.

There are under run or overrun interrupt when the two-level buffer mode is enabled and the interrupt event enable is set.

Note: A pull-up resistor is needed for I²C operation as the SDA and SCL are open-drain pins.

6.11.5.1 I²C Protocol

Figure 6.11-3 shows the typical I²C protocol. Normally, a standard communication consists of four parts:

1) START or Repeated START signal generation
2) Slave address and R/W bit transfer
3) Data transfer
4) STOP signal generation

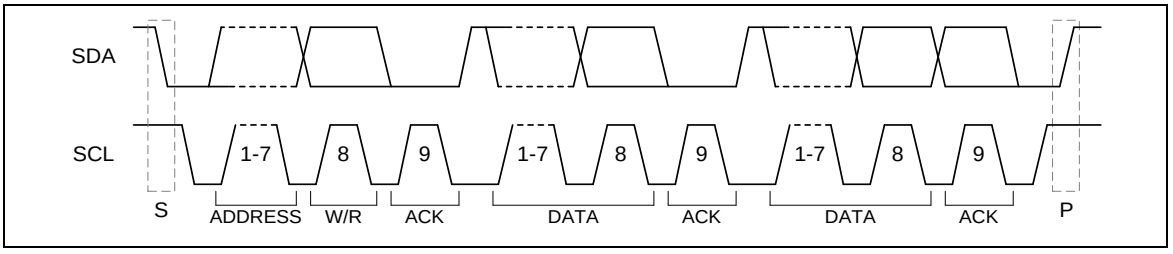


Figure 6.11-3 I²C Protocol

6.11.5.1.1 START or Repeated START signal

When the bus is free/idle, which means that 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 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 transfer.

A Repeated START is not a STOP signal between two START signals and usually referred to as the "Sr" bit. The master uses this method to communicate with another slave or the same slave in a different transfer direction (e.g. from writing to a device to reading from a device) without releasing the bus.

6.11.5.1.2 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.

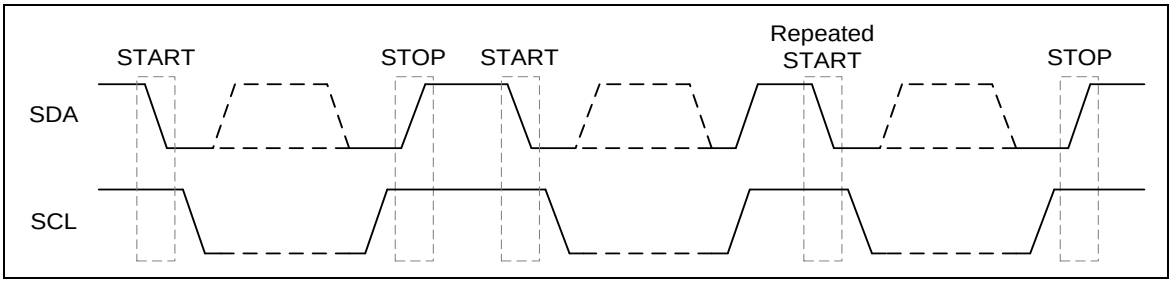


Figure 6.11-4 START and STOP Condition

6.11.5.1.3 Slave Address Transfer

The first byte of data transferred by the master immediately after the START signal is the Slave address (SLA). This is a 7-bit calling address followed by a Read/Write (RW) bit. The R/W bit signals of the slave indicate the data transfer direction. No two slaves in the system can have the same address. Only the slave with an address that matches the one transmitted by the master will respond by returning an acknowledge bit by pulling the SDA low at the 9th SCL clock cycle.

6.11.5.1.4 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.

If the master, as the 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.

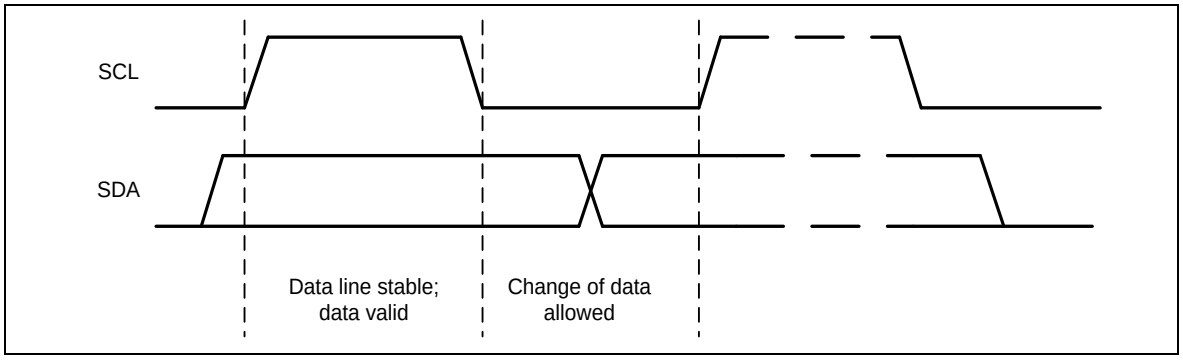


Figure 6.11-5 Bit Transfer on I^{2}C Bus

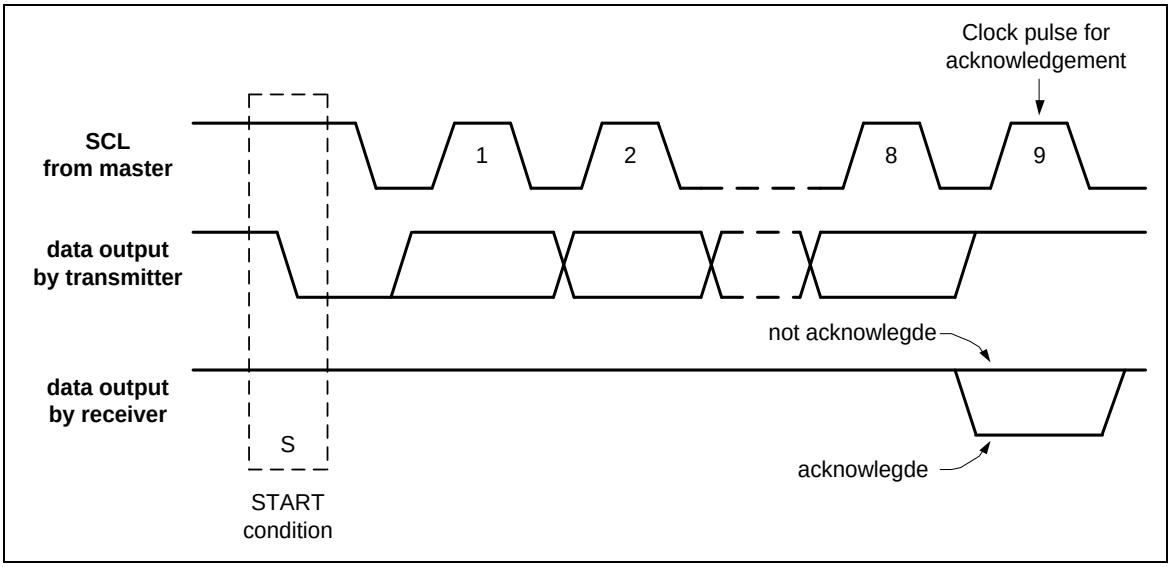


Figure 6.11-6 Acknowledge on I²C Bus

6.11.5.1.5 Data transfer on I²C bus

Figure 6.11-7 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.

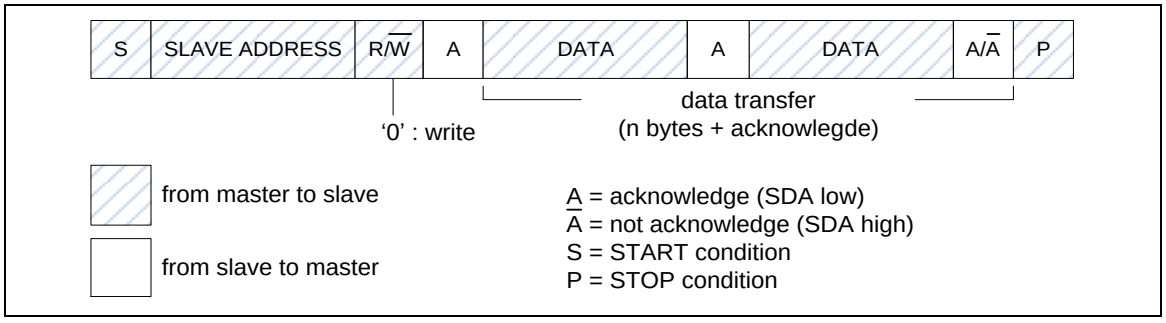


Figure 6.11-7 Master Transmits Data to Slave

Figure 6.11-8 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.

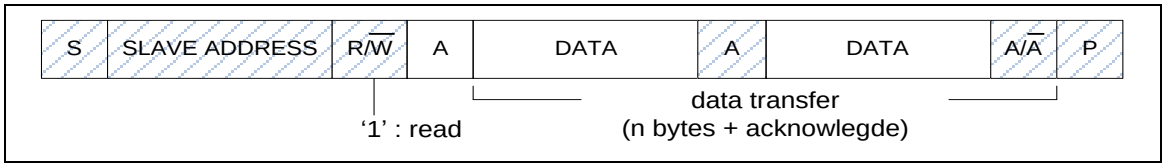


Figure 6.11-8 Master Reads Data from Slave

6.11.5.1.6 Two-level Buffer Mode on I²C bus

Set to enable the two-level buffer for I²C transmitted or received buffer. It is used to improve the

performance of the I²C bus. If this TWOFF_EN 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 3rd data's SI event is cleared. In this time, the 4th data can be transmitted and the I²C stop after the 4th data transmission done.

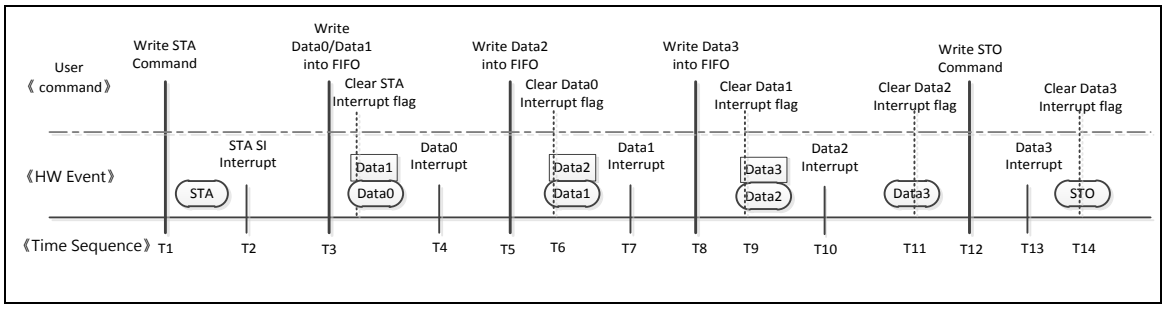


Figure 6.11-9 Timing of Two-level Buffer Transmission in Master Write

For example, if there are 4 data shall be received in Slave mode. The controller can receives the 2nd data in the I²C bus after the 1st data had been loaded into the received buffer and the user can read the 1st data after the 1st interrupt status is cleared.

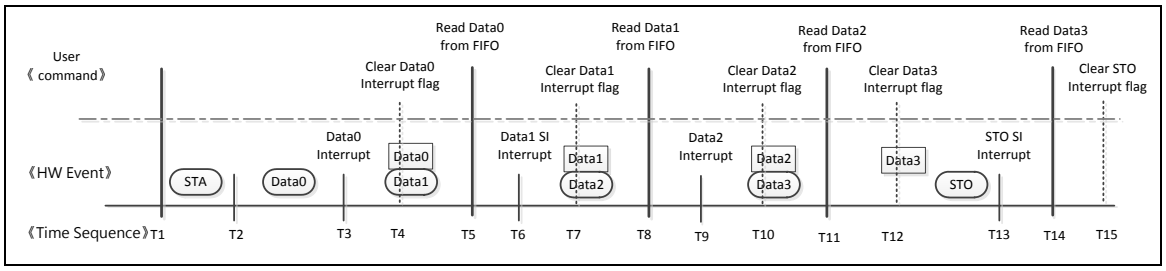


Figure 6.11-10 Timing of Two-level Buffer Transmission in Slave Read

If the buffer is not empty after the bus stop (STO), the user can set the TWOFF_EN bit to '0' to make the buffer control state machine enter idle state and the buffer status will be set to the default value.

6.11.5.2 Operation Modes

The on-chip I²C ports support three operation modes, Master, Slave, and General Call Mode.

In a given application, an I²C port may operate as a master or as a slave. In Slave mode, the I²C port hardware looks for its own slave address and the general call address. If one of these addresses is detected, and if the slave is willing to receive or transmit data from/to master(by setting the AA bit), acknowledge pulse will be transmitted out on the 9th clock, hence an interrupt is requested on both master and slave devices if interrupt is enabled. When the microcontroller wishes to become the bus master, hardware waits until the bus is free before entering Master mode so that a possible slave action is not be interrupted. If bus arbitration is lost in Master mode, I²C port switches to Slave mode immediately and can detect its own slave address in the same serial transfer.

To control the I²C bus transfer in each mode, user needs to set I2C_CTL, I2C_DAT registers

according to current status code of I2C_STATUS register. In other words, for each I^{2}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^{2}C hardware after the SI flag of 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. If the I^{2}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 6.11-11 shows the current I^{2}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^{2}C bus. If a slave on the bus matches the address and response ACK, the I2C_STATUS will be updated by status code 0x18.

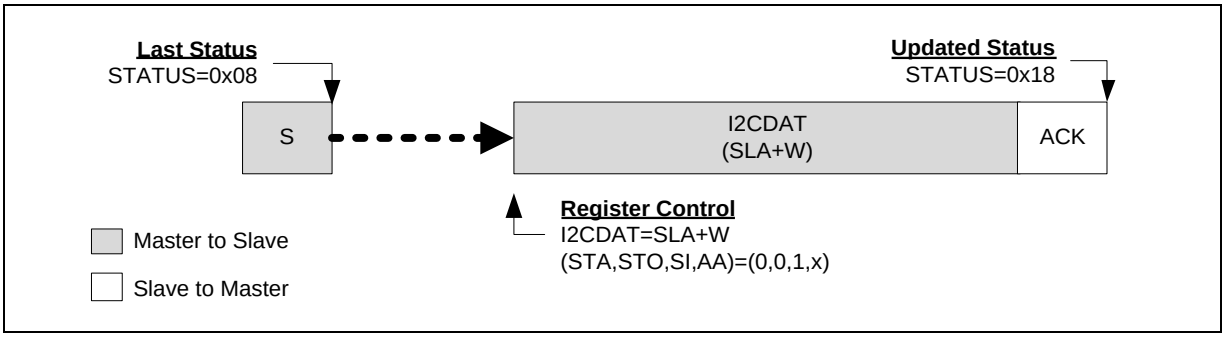


Figure 6.11-11 Control I^{2}C Bus according to Current I^{2}C Status

6.11.5.2.1 Master Mode

All possible protocols for I^{2}C master are shown in Figure 6.11-15 and Figure 6.11-13. User needs to follow proper path of the flow to implement required I^{2}C protocol.

In other words, user can send a START signal to bus and I^{2}C will be in Master Transmitter mode (as shown in Figure 6.11-15) or Master receiver mode (as shown in Figure 6.11-13) after START signal has been sent successfully and new status code would be 0x08. Followed by START signal, user can send slave address, read/write bit, data and Repeat START, STOP to perform I^{2}C protocol.

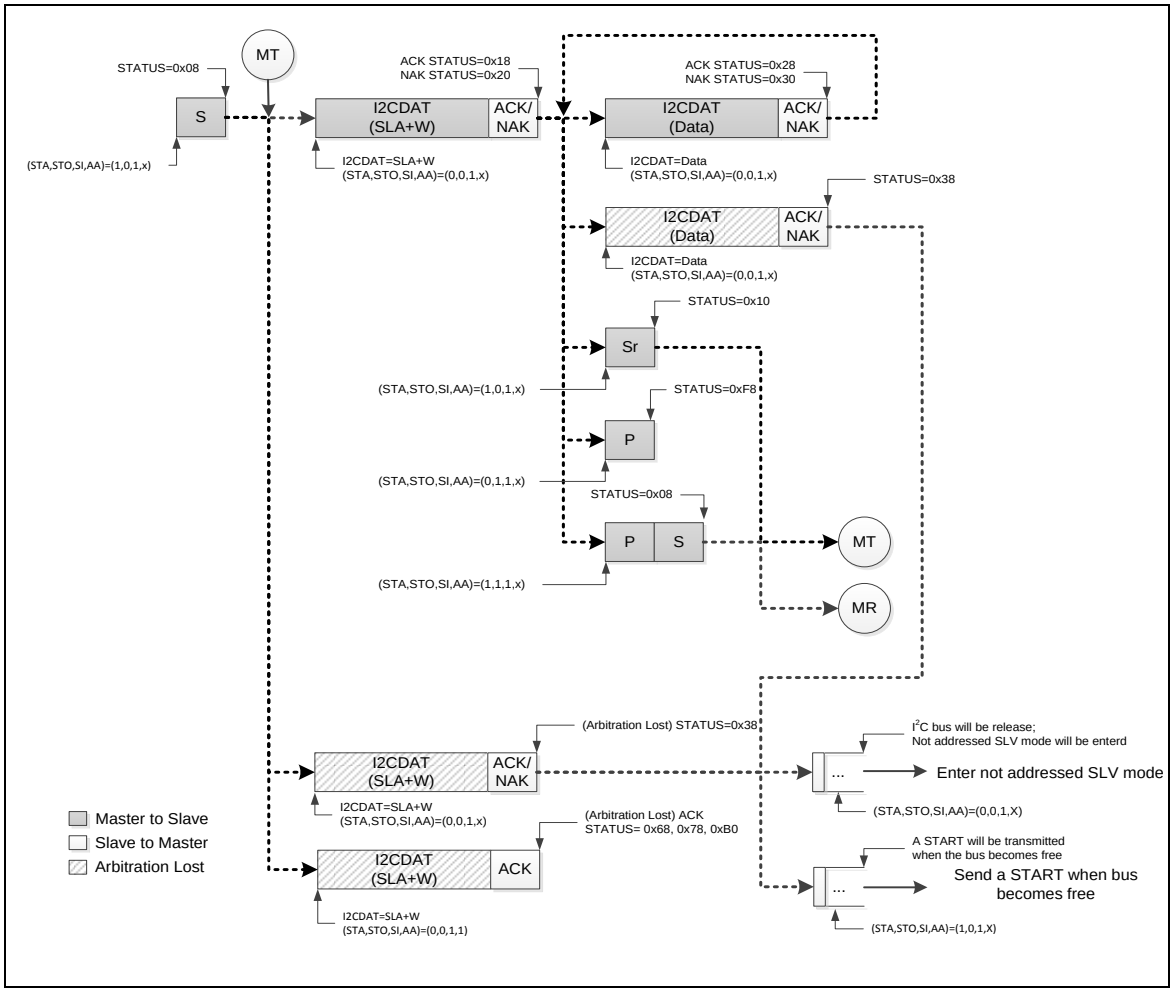


Figure 6.11-12 Master Transmitter Mode Control Flow

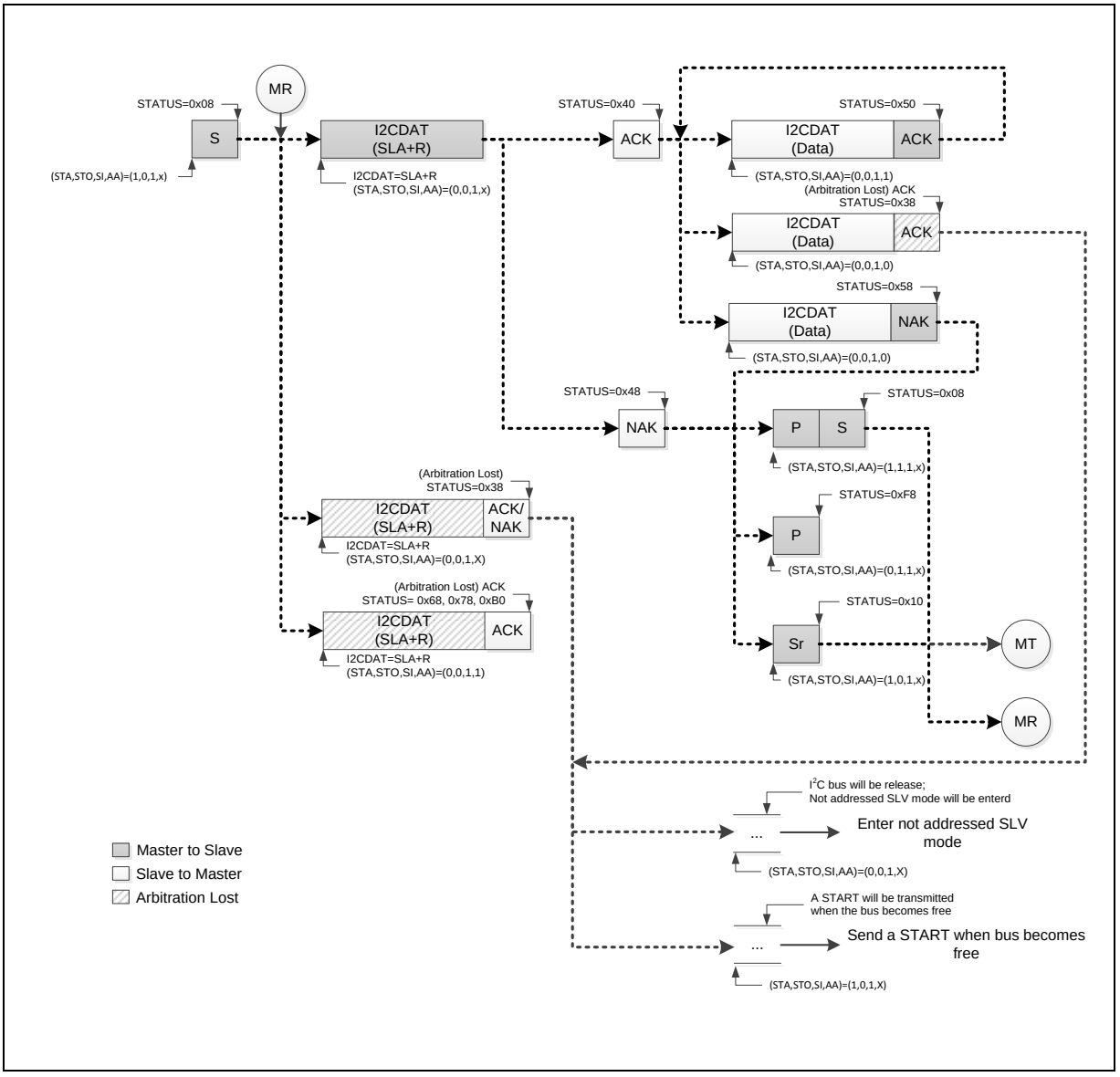


Figure 6.11-13 Master Receiver Mode Control Flow

If the I²C is in Master mode and gets arbitration lost, the status code will be 0x38. In status 0x38, user may set (STA, STO, SI, AA) = (1, 0, 1, X) to send START to re-start Master operation when bus becomes free. Otherwise, user may set (STA, STO, SI, AA) = (0, 0, 1, X) to release I²C bus and enter not addressed Slave mode.

6.11.5.2.2 Slave Mode

When reset as default, I²C is not addressed and will not recognize the address on I²C bus. User can set slave address by I2C_ADDRx and set (STA, STO, SI, AA) = (0, 0, 1, 1) to let I²C recognize the address sent by master. Figure 6.11-15 shows all the possible flow for I²C in Slave mode. Users need to follow a proper flow (as shown in Figure 6.11-15) to implement their own I²C protocol.

If bus arbitration is lost in Master mode, the I²C port switches to Slave mode immediately and can detect its own slave address in the same serial transfer. If the detected address is SLA+W (Master want to write data to Slave) after arbitration lost, the status code is 0x68. If the detected address is SLA+R (Master want to read data from Slave) after arbitration lost, the status code is 0xB0.

Note: During I²C communication, the SCL clock will be released when writing '1' to clear the SI flag in Slave mode.

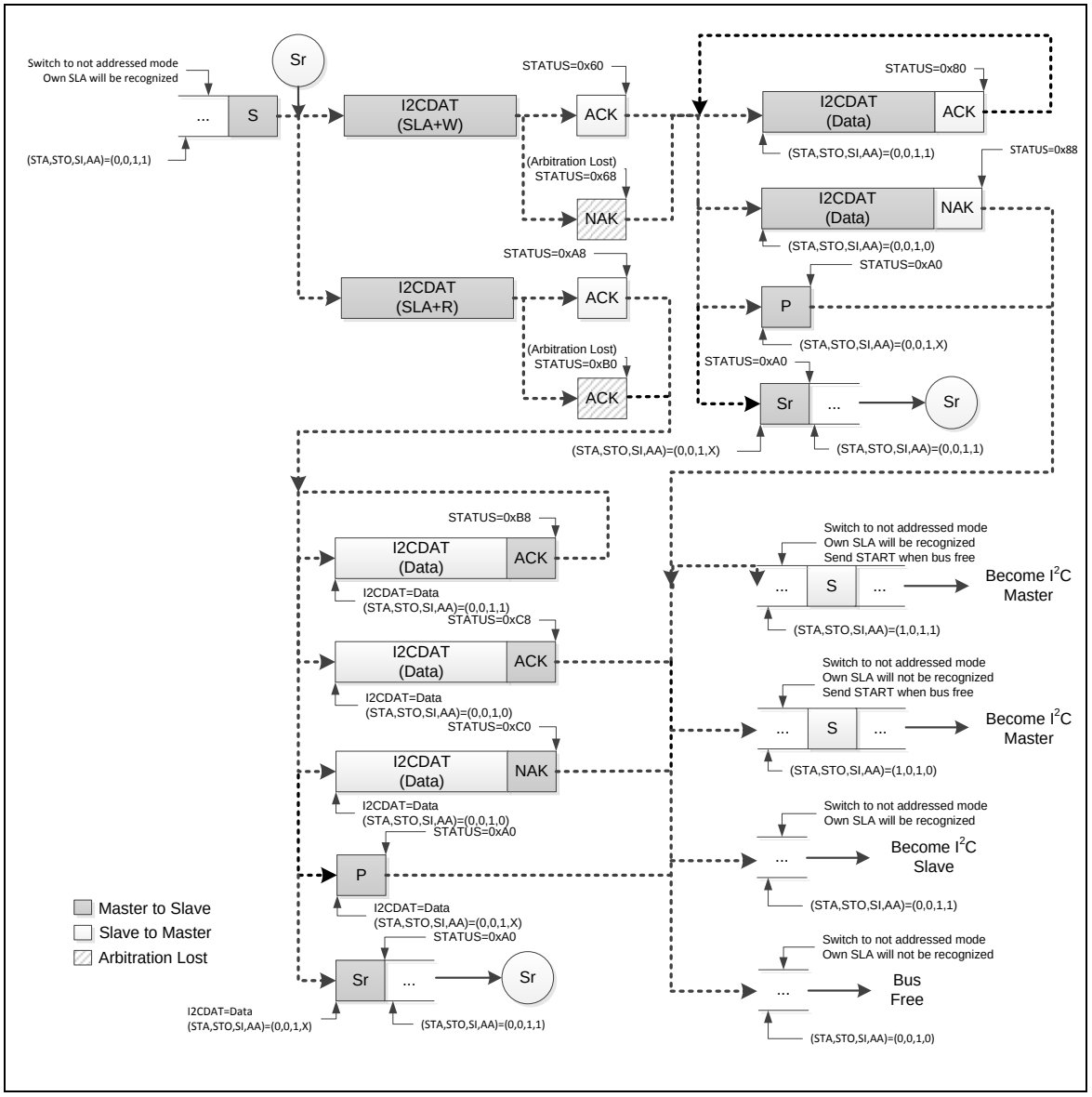


Figure 6.11-14 Slave Mode Control Flow

If I²C is still receiving data in addressed Slave mode but got a STOP or Repeat START, the status code will be 0xA0. User could follow the action for status code 0x88 as shown in Figure 6.11-15 when getting 0xA0 status.

If I²C is still transmitting data in addressed Slave mode but got a STOP or Repeat START, the status code will be 0xA0. User could follow the action for status code 0xC8 as shown in Figure

6.11-15 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^2C signal or address from master. At this status, I^2C should be reset to leave this status.

6.11.5.2.3 General Call (GC) Mode

If the GC bit (I2C_ADDRx [0]) is set, the I^2C 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^2C is in Slave mode, it can receive the general call address by 0x00 after master send general call address to I^2C bus, then it will follow status of GC mode.

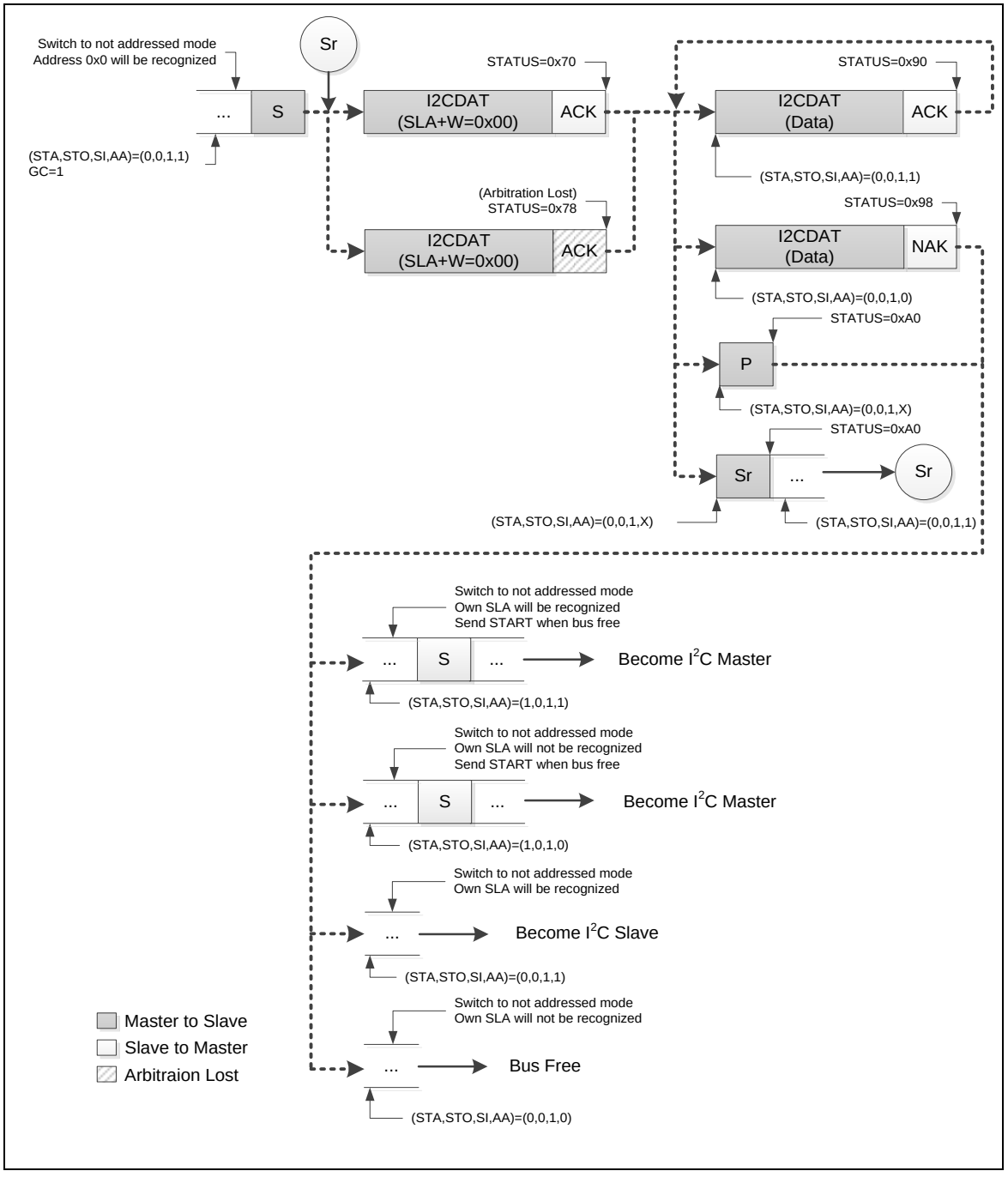


Figure 6.11-15 GC Mode

If I²C 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 6.11-15 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 I²C signal or address from master. At this time, the I²C controller should be reset to exit this status.

6.11.5.2.4 Multi-Master

In some applications, there are two or more masters on the same I^2C bus to access slaves, and the masters may transmit data simultaneously. The I^2C supports multi-master by including collision detection and arbitration to prevent data corruption.

- 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 send STOP to back to not addressed Slave mode.
- When I2C_STATUS = 0x00, a “Bus Error” is received. To recover I^2C 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 I^2C Protocol Registers

To control the I^2C 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 rate register) and I2C_TOCTL (time-out counter register), I2C_CTL1 (control register 1) and I2C_STATUS1 (status register 1).

6.11.5.3.1 Address Registers (I2C_ADDR)

The I^2C port is equipped with four slave address registers I2C_ADDRn (n=0~3). The contents of the register are irrelevant when I^2C is in Master mode. In Slave mode, the bit field I2C_ADDRn[7:1] must be loaded with the chip’s own slave address. The I^2C hardware will react if the contents of I2C_ADDRn are matched with the received slave address.

The I^2C ports support the “General Call” function. If the GC bit (I2C_ADDRn[0]) is set, the I^2C port hardware will respond to General Call address (00H). Clearing GC bit will disable general call function.

When GC bit is set and the I^2C is in Slave mode, it can receive the general call address by 00H after Master send general call address to I^2C bus, then it will follow status of GC mode.

6.11.5.3.2 Slave Address Mask Registers (I2C_ADDRMSK)

The I^2C bus controller supports multiple address recognition with four address mask registers I2C_ADDRMSKx (x=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.

6.11.5.3.3 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 read from or write to this 8-bit (I2C_DAT[7:0]) directly while it is not in the process of shifting a byte. When I^2C 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 present on the bus.

The acknowledge bit is controlled by the I²C hardware and cannot be accessed by the CPU. Serial data is shifted through 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 date will be shifted to I2C_DATA[7:0] when sending I2C_DATA[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.

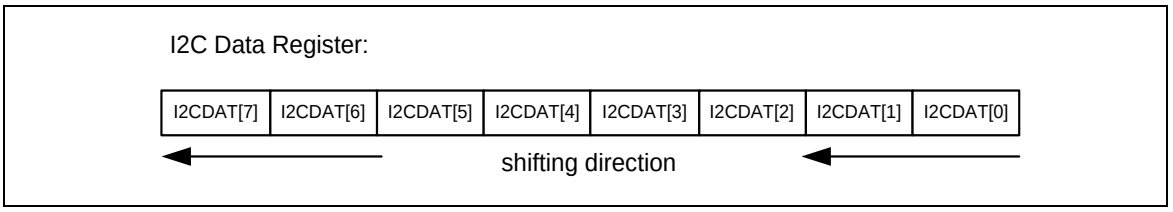


Figure 6.11-16 I²C Data Shifting Direction

6.11.5.3.4 Control Register (I2C_CTL)

The CPU can read from and write to I2C_CTL[7:0] directly. When the I²C port is enabled by setting I2CEN (I2C_CTL [6]) to high, the internal states will be controlled by I2C_CTL and I²C logic hardware.

There are two bits are affected by hardware: the SI bit is set when the I²C hardware requests a serial interrupt, and the STO bit is cleared when a STOP condition is present on the bus. The STO bit is also cleared when I2CEN = 0.

Once a new status code is generated and stored in I2C_STATUS, the I²C 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 I²C 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.

6.11.5.3.5 Status Register (I2C_STATUS)

I2C_STATUS[7:0] is an 8-bit read-only register. The bit field I2C_STATUS[7:0] contain the status code. There are 26 possible status codes. All states are listed in section Table 6.11-1. When I2C_STATUS[7:0] is F8H, no serial interrupt is requested. All other I2C_STATUS[7:0] values correspond to defined I²C states. When each of these states is entered, a status interrupt is requested (SI = 1). A valid status code is present in I2C_STATUS[7:0] one cycle after SI is set by hardware and is still present one cycle after SI has been reset by software.

In addition, the state 00H stands for a Bus Error, which occurs when a START or STOP condition is present at an incorrect position in the I²C format frame. A Bus Error may occur during the serial transfer of an address byte, a data byte or an acknowledge bit. To recover I²C from bus error, STO should be set and SI should be cleared to enter Not Addressed Slave mode. Then STO is cleared to release bus and to wait for a new communication. I²C bus cannot recognize stop condition during this action when bus error occurs.

Master Mode		**Slave Mode**	
STATUS	**Description**	**STATUS**	**Description**
0x08	Start	0xA0	Slave Transmit Repeat Start or Stop
0x10	Master Repeat Start	0xA8	Slave Transmit Address ACK
0x18	Master Transmit Address ACK	0xB0	Slave Transmit Arbitration Lost

0x20	Master Transmit Address NACK	0xB8	Slave Transmit Data ACK
0x28	Master Transmit Data ACK	0xC0	Slave Transmit Data NACK
0x30	Master Transmit Data NACK	0xC8	Slave Transmit Last Data ACK
0x38	Master Arbitration Lost	0x60	Slave Receive Address ACK
0x40	Master Receive Address ACK	0x68	Slave Receive Arbitration Lost
0x48	Master Receive Address NACK	0x80	Slave Receive Data ACK
0x50	Master Receive Data ACK	0x88	Slave Receive Data NACK
0x58	Master Receive Data NACK	0x70	GC mode Address ACK
0x00	Bus error	0x78	GC mode Arbitration Lost
		0x90	GC mode Data ACK
		0x98	GC mode Data NACK
0xF8	Bus Released **Note:** The status “0xF8” exists in both master/slave modes, and it won’t raise interrupt.		

Table 6.11-1 I^{2}C Status Code Description

6.11.5.3.6 I^{2}C Clock Baud Rate Bits (I2C_CLKDIV)

The data baud rate of I^{2}C is determined by the I2C_CLKDIV[7:0] register when I^{2}C is in Master mode. It is not necessary in a Slave mode. In Slave mode, I^{2}C will automatically synchronize with any clock frequency from master I^{2}C device.

The data baud rate of I^{2}C setting is Data Baud Rate of I^{2}C = (system clock) / (4 * (I2C_CLKDIV[7:0] +1)). If system clock = 16 MHz, the I2C_CLKDIV[7:0] = 40 (28H), the data baud rate of I^{2}C = 16 MHz / (4 * (40 +1)) = 97.5K bits/sec.

6.11.5.3.7 I^{2}C Time-out Counter Register (I2C_TOCTL)

There is a 14-bit time-out counter which can be used to deal with the I^{2}C bus hang-up. If the time-out counter is enabled, the counter starts up counting until it overflows (TOIF=1) and generates I^{2}C interrupt to CPU or stops counting by clearing I2CEN to 0. When time-out counter is enabled, setting flag SI to high will reset counter and re-start up counting after SI is cleared. If I^{2}C bus is hung 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^{2}C interrupt. Refer to Figure 6.11-17 for the 14-bit time-out counter. User may write 1 to clear TOIF to 0.

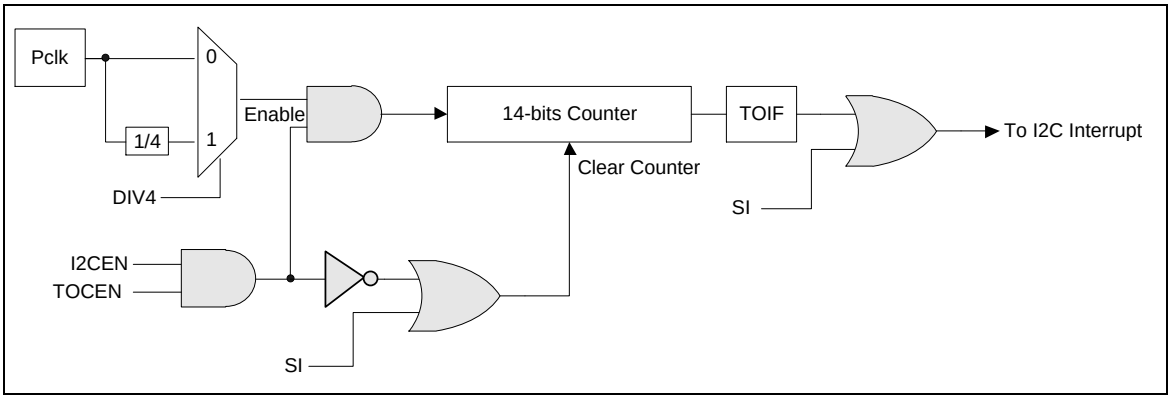


Figure 6.11-17 I²C Time-out Count Block Diagram

6.11.5.3.8 I²C Control Register 1 (I2C_CTL1)

The NSTRETCH (I2C_CTL1[2]) bit is used to no stretch the bus clock when this bit is set to 1.

For the TWOLVFIFO (I2C_CTL1[1]) bit , it is used to enable the two-level buffer for I²C transmitted or received buffer. It is used to improve the performance of the I²C bus. If this bit is set to 1, the control bit of START (I2C_CTL[5]) for repeat start or STOP (I2C_CTL[4]) bit should be set after the current interrupt is cleared. For example, if there are 4 data to be transmitted and then stopped. The STOP bit shall be set after the 3rd data's interrupt event is cleared. At this time, the 4th data can be transmitted and the I²C stops after the 4th data transmission done.

The two-level buffer status interrupt enable can be enabled by I2C_CTL1[4:3] to generate the under run or overrun event.

When entering Power-down mode, other I²C master can wake-up the chip by addressing the I²C device, and user must configure the related setting before entering Sleep mode. When the chip is woken-up by address match with one of the four address register, the following data will be abandoned at this time. Note that only I²C0 channel supports wake-up function.

6.11.5.3.9 I²C Status Register 1 (I2C_STATUS1)

The two-level buffer status, busy free information, buffer empty, buffer full and overrun or under run are also listed in this register.

When system is woken up by other I²C master device, WKIF is set to indicate this event. User needs to write "1" to clear this bit. The other status bits are used to indicate the current buffer status when the TWOLVFIFO (I2C_CTL1[1]) is set. Note that only I²C0 channel supports wake-up function.

6.11.5.4 Example of Random Read on EEPROM

The following steps are used to configure the I²C related registers when using I²C to read data from EEPROM.

1. Set the multi-function pin in the "SYS_P3_MFP" registers as SCL and SDA pins.
2. Enable I²C APB clock, I2CCKEN=1 in the "CLK_APBCLK" register.
3. Set I2C_RST=1 to reset I²C controller then set I²C controller to normal operation, I2CRST=0 in the "SYS_IPRST1" register.

4. Set I2CEN=1 to enable I²C controller in the “I2C_CTL” register.
5. Give I²C clock a divided register value for I²C clock rate in the “I2C_CLKDIV”.
6. Set SETENA=0x00040000 in the “NVIC_ISER” register to set I²C IRQ.
7. Set EI=1 to enable I²C Interrupt in the “I2C_CTL” register.
8. Set I²C address registers which are “I2C_ADDR0~I2C_ADDR3”.

Random read operation is one of the methods toaccess EEPROM. The method allows the master to access any address of EEPROM space. Figure 6.11-18 shows the EEPROM random read operation.

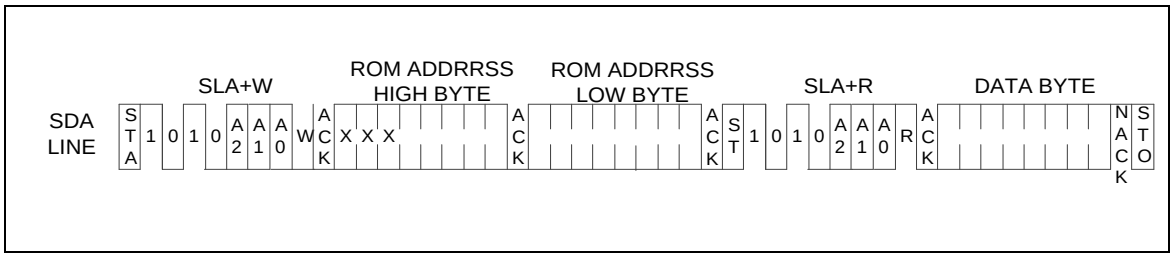


Figure 6.11-18 EEPROM Random Read

Figure 6.11-19 shows how to use I²C controller to implement the protocol of EEPROM random read.

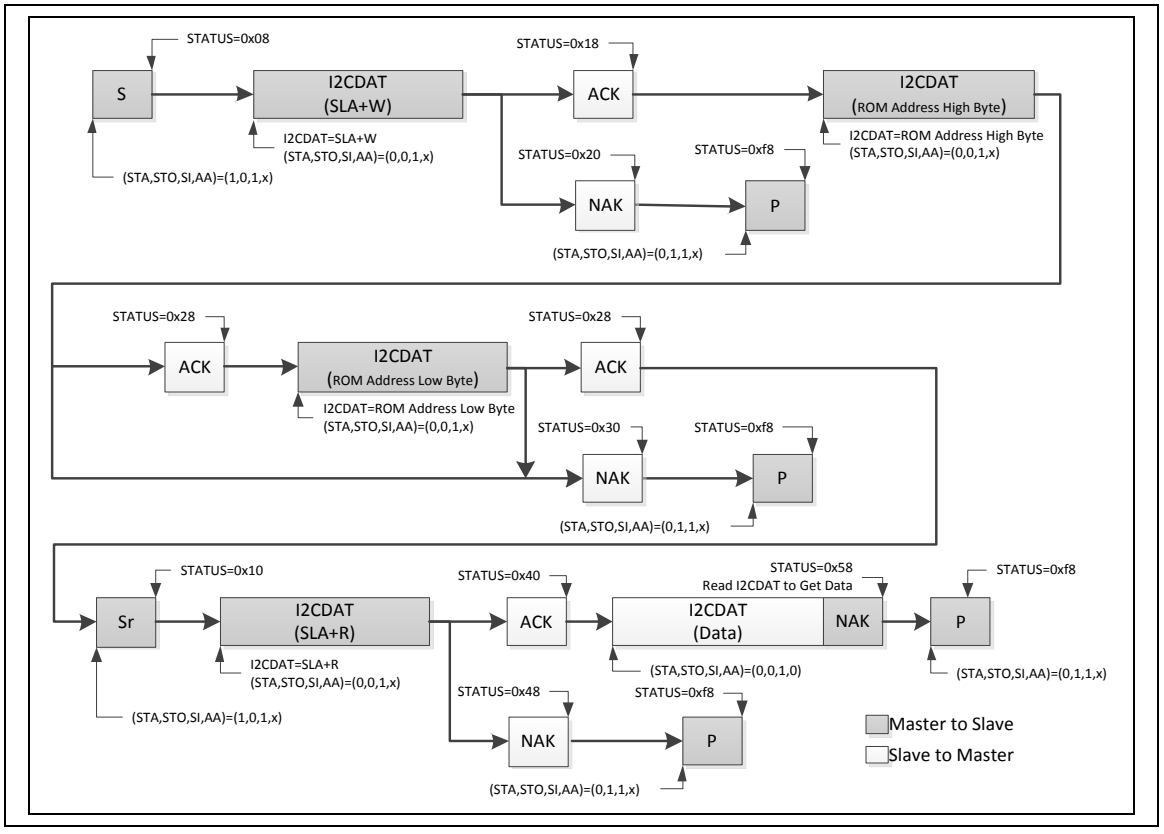


Figure 6.11-19 Protocol of EEPROM Random Read

The I^2C controller sends START to bus to be a master. Then it sends a SLA+W (Slave address + Write bit) to EERPOM 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
I²C Base Address: **I2C0_BA = 0x4002_0000** **I2C1_BA = 0x4012_0000**				
I2C_CTL **x= 0, 1**	I2Cx_BA+0x00	R/W	I²C Control Register	0x0000_0000
I2C_DAT **x= 0, 1**	I2Cx_BA+0x08	R/W	I²C DATA Register	0x0000_0000
I2C_STATUS **x= 0, 1**	I2Cx_BA+0x0C	R	I²C Status Register	0x0000_00F8
I2C_CLKDIV **x= 0, 1**	I2Cx_BA+0x10	R/W	I²C Clock Divided Register	0x0000_0000
I2C_TOCTL **x= 0, 1**	I2Cx_BA+0x14	R/W	I²C Time-out Control Register	0x0000_0000
I2C_ADDR0 **x= 0, 1**	I2Cx_BA+0x04	R/W	I²C Slave Address Register 0	0x0000_0000
I2C_ADDR1 **x= 0, 1**	I2Cx_BA+0x18	R/W	I²C Slave Address Register 1	0x0000_0000
I2C_ADDR2 **x= 0, 1**	I2Cx_BA+0x1C	R/W	I²C Slave Address Register 2	0x0000_0000
I2C_ADDR3 **x= 0, 1**	I2Cx_BA+0x20	R/W	I²C Slave Address Register 3	0x0000_0000
I2C_ADDRMSK0 **x= 0, 1**	I2Cx_BA+0x24	R/W	I²C Slave Address Mask Register 0	0x0000_0000
I2C_ADDRMSK1 **x= 0, 1**	I2Cx_BA+0x28	R/W	I²C Slave Address Mask Register 1	0x0000_0000
I2C_ADDRMSK2 **x= 0, 1**	I2Cx_BA+0x2C	R/W	I²C Slave Address Mask Register 2	0x0000_0000
I2C_ADDRMSK3 **x= 0, 1**	I2Cx_BA+0x30	R/W	I²C Slave Address Mask Register 3	0x0000_0000
I2C_CTL1 **x= 0, 1**	I2Cx_BA+0x3C	R/W	I²C Control Register 1	0x0000_0000
I2C_STATUS1 **x= 0, 1**	I2Cx_BA+0x40	R/W	I²C Status Register 1	0x0000_0000

6.11.7 Register Description

I²C Control Register (I2C_CTL)

Register	Offset	R/W	Description	Reset Value
I2C_CTL **x= 0, 1**	I2Cx_BA+0x00	R/W	I²C 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**
INTEN	**I2CEN**	**STA**	**STO**	**SI**	**AA**	**Reserved**	

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**INTEN**	**Interrupt Enable Bit** 0 = I²C interrupt Disabled. 1 = I²C interrupt Enabled.
[6]	**I2CEN**	**I²C Controller Enable Bit** 0 = I²C Controller Disabled. 1 = I²C Controller Enabled. Set to enable I²C serial function controller. When I2CEN=1 the I²C serial function enables. The function of multi-function pin must be set to I²C first.
[5]	**STA**	**I²C START Control Bit** Set STA to logic 1 to enter Master mode. I²C hardware sends a START or repeats the START condition to bus when the bus is free.
[4]	**STO**	**I²C STOP Control Bit** In Master mode, setting STO to transmit a STOP condition to bus then I²C hardware will check the bus condition if a STOP condition is detected this bit will be cleared by hardware automatically. In Slave mode, setting STO resets I²C hardware to the defined “not addressed” Slave mode. This means it is NO LONGER in the Slave receiver mode to receive data from the master transmit device.
[3]	**SI**	**I²C Interrupt Flag** When a new I²C state is present in the I2C_STATUS register, the SI flag is set by hardware, and if bit INTEN (I2C_CTL[7]) is set, the I²C interrupt is requested. SI must be cleared by software. This bit can be cleared by software writing ‘1’.

Bits	Description	
[2]	**AA**	**Assert Acknowledge Control Bit** When AA=1 is 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.
[1:0]	**Reserved**	Reserved.

I²C Data Register (I2C_DAT)

Register	Offset	R/W	Description	Reset Value
I2C_DAT **x= 0, 1**	I2Cx_BA+0x08	R/W	I²C 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[7:0]**	**I²C Data Register** Bit [7:0] is located with the 8-bit transferred data of the I²C serial port.

I²C Status Register (I2C_STATUS)

Register	Offset	R/W	Description	Reset Value
I2C_STATUS x= 0, 1	I2Cx_BA+0x0C	R	I²C Status Register	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[7:0]**	**I²C Status Register** The three least significant bits are always 0. The five most significant bits contain the status code. There are 26 possible status codes. When I2C_STATUS contains F8H, no serial interrupt is requested. All the other I2C_STATUS values correspond to defined I²C states. When each of these states is entered, a status interrupt is requested (SI = 1). A valid status code is present in I2C_STATUS one cycle after SI is set by hardware and is still present one cycle after SI has been reset by software. In addition, the 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. Examples of illegal position are during the serial transfer of an address byte, a data byte or an acknowledge bit.

I²C Clock Divided Register (I2C_CLKDIV)

Register	Offset	R/W	Description	Reset Value
I2C_CLKDIV **x= 0, 1**	I2Cx_BA+0x10	R/W	I²C 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[7:0]**	**I²C Clock Divided Register** The I²C clock rate bits: Data Baud Rate of I²C = (system clock) / (4 * (I2C_CLKDIV+1)). **Note:** The minimum value of I2C_CLKDIV is 4.

I²C Time-out Control Register (I2C_TOCTL)

Register	Offset	R/W	Description	Reset Value
I2C_TOCTL x= 0, 1	I2Cx_BA+0x14	R/W	I²C 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	TOCURIEN	TOIF

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2]	**TOCEN**	**Time-out Counter Enable Bit** 0 = Time-out counter Disabled. 1 = Time-out counter Enabled. **Note:** When the 14-bit time-out counter is enabled, it will start counting when SI is cleared. Setting 1 to the SI flag will reset counter and re-start up counting after SI is cleared.
[1]	**TOCURIEN**	**Time-out Counter Input Clock Divided By 4** 0 = Time-out counter input clock divided by 4 Disabled. 1 = Time-out counter input clock divided by 4 Enabled. **Note:** When enabled, the time-out period is extended 4 times.
[0]	**TOIF**	**Time-out Flag** This bit is set by hardware when I²C time-out happened and it can interrupt CPU if I²C interrupt enable bit (INTEN) is set to 1. **Note:** This bit can be cleared by software writing '1'.

I²C Slave Address Register (I2C_ADDRx)

Register	Offset	R/W	Description	Reset Value
I2C_ADDR0 **x= 0, 1**	I2Cx_BA+0x04	R/W	I²C Slave Address Register 0	0x0000_0000
I2C_ADDR1 **x= 0, 1**	I2Cx_BA+0x18	R/W	I²C Slave Address Register 1	0x0000_0000
I2C_ADDR2 **x= 0, 1**	I2Cx_BA+0x1C	R/W	I²C Slave Address Register 2	0x0000_0000
I2C_ADDR3 **x= 0, 1**	I2Cx_BA+0x20	R/W	I²C Slave Address 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							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ADDR							**GC**

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:1]	**ADDR[7:1]**	**I²C Address** The content of this register is irrelevant when I²C is in Master mode. In Slave mode, the seven most significant bits must be loaded with the MCU's own address. The I²C hardware will react if either of the address is matched.
[0]	**GC**	**General Call Function Control** 0 = General Call Function Disabled. 1 = General Call Function Enabled.

I²C Slave Address Mask Register (I2C_ADDRMSKx)

Register	Offset	R/W	Description	Reset Value
I2C_ADDRMSK0 **x= 0, 1**	I2Cx_BA+0x24	R/W	I²C Slave Address Mask Register 0	0x0000_0000
I2C_ADDRMSK1 **x= 0, 1**	I2Cx_BA+0x28	R/W	I²C Slave Address Mask Register 1	0x0000_0000
I2C_ADDRMSK2 **x= 0, 1**	I2Cx_BA+0x2C	R/W	I²C Slave Address Mask Register 2	0x0000_0000
I2C_ADDRMSK3 **x= 0, 1**	I2Cx_BA+0x30	R/W	I²C Slave Address Mask 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							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ADDRMSK							**Reserved**

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:1]	**ADDRMSK[7:1]**	**I²C Address Mask Bits** 0 = I²C address mask Disabled (the received corresponding register bit should be exactly the same as address register). 1 = I²C address mask Enabled (the received corresponding address bit is “Don’t care”).
[0]	**Reserved**	Reserved.

I²C Control Register 1 (I2C_CTL1)

Register	Offset	R/W	Description	Reset Value
I2C_CTL1 x= 0, 1	I2Cx_BA+0x3C	R/W	I²C Control Register 1	0x0000_0000

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

Bits	Description	
[31:5]	**Reserved**	Reserved.
[4]	**URIEN**	**I²C Under Run Interrupt Control** Setting URIEN to 1 will send a interrupt to system when the TWOLVFF bit is enabled and there is under run event happened in transmitted buffer. 0 = Under run Interrupt Disabled. 1 = Under run Interrupt Enabled.
[3]	**OVIEN**	**I²C Overrun Interrupt Control** Setting OVIEN to 1 will send a interrupt to system when the TWOLVFF bit is enabled and there is overrun event in received buffer. 0 = Overrun Interrupt Disabled. 1 = Overrun Interrupt Enabled.
[2]	**NSTRETCH**	**No Stretch On The I²C Bus** 0 = The I²C SCL bus is stretched by hardware if the SI is not cleared in master mode. 1 = The I²C SCL bus is not stretched by hardware if the SI is not cleared in master mode.
[1]	**TWOLVFIFO**	**Two-level Buffer Enable Bit** 0 = Two-level buffer Disabled. 1 = Two-level buffer Enabled. Set to enable the two-level buffer for I²C transmitted or received buffer. It is used to improve the performance of the I²C 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 3rd data's SI event being clear. In this time, the 4th data can be transmitted and the I²C stop after the 4th data transmission done.

Bits	Description	
[0]	**WKEN**	**Wake-up Enable Bit** 0 = I^2C wake-up function Disabled. 1 = I^2C wake-up function Enabled. The system can be woken up by I^2C bus when the system is set into Power mode and the received data matched one of the addresses in Address Register. **Note:** Only I^2C0 channel supports wake-up function. This bit is not valid on I^2C1 channel.

I²C Status Register 1 (I2C_STATUS1)

Register	Offset	R/W	Description	Reset Value
I2C_STATUS1 x= 0, 1	I2Cx_BA+0x40	R/W	I²C 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							
7	6	5	4	3	2	1	0
Reserved			URIF	OVIF	EMPTY	FULL	WKIF

Bits	Description	
[31:5]	**Reserved**	Reserved.
[4]	**URIF**	**I²C Under Run Status** This bit indicates the transmitted buffer is under run when the TWOLVFIFO = 1.
[3]	**OVIF**	**I²C Overrun Status** This bit indicates the received buffer is overrun when the TWOLVFIFO = 1.
[2]	**EMPTY**	**I²C Two-level Buffer Empty** This bit indicates if the RX buffer is empty or not when the TWOLVFIFO = 1.
[1]	**FULL**	**I²C Two-level Buffer Full** This bit indicates if the TX buffer is full or not when the TWOLVFIFO = 1.
[0]	**WKIF**	**I²C Wake-up Interrupt Flag** When chip is woken up from Power-down mode by I²C, this bit is set to 1. This bit can be cleared by software writing “1”. **Note:** Only I²C0 channel supports wake-up function. This bit is not valid on I²C1 channel.

6.12 Serial Peripheral Interface (SPI)

6.12.1 Overview

The Serial Peripheral Interface (SPI) applies to synchronous serial data communication and allows full duplex transfer. Devices communicate in Master/Slave mode with 4-wire bi-direction interface. The SPI controller performing a serial-to-parallel conversion on data received from a peripheral device, and a parallel-to-serial conversion on data transmitted to a peripheral device. SPI controller can be configured as a master or a slave device.

6.12.2 Features

- Supports Master or Slave mode operation
- Configurable transfer bit length
- Provides four 32-bit FIFO buffers
- Supports MSB first or LSB first transfer
- Supports byte reorder function
- Supports byte or word suspend mode
- Supports Slave 3-wire mode

6.12.3 Block Diagram

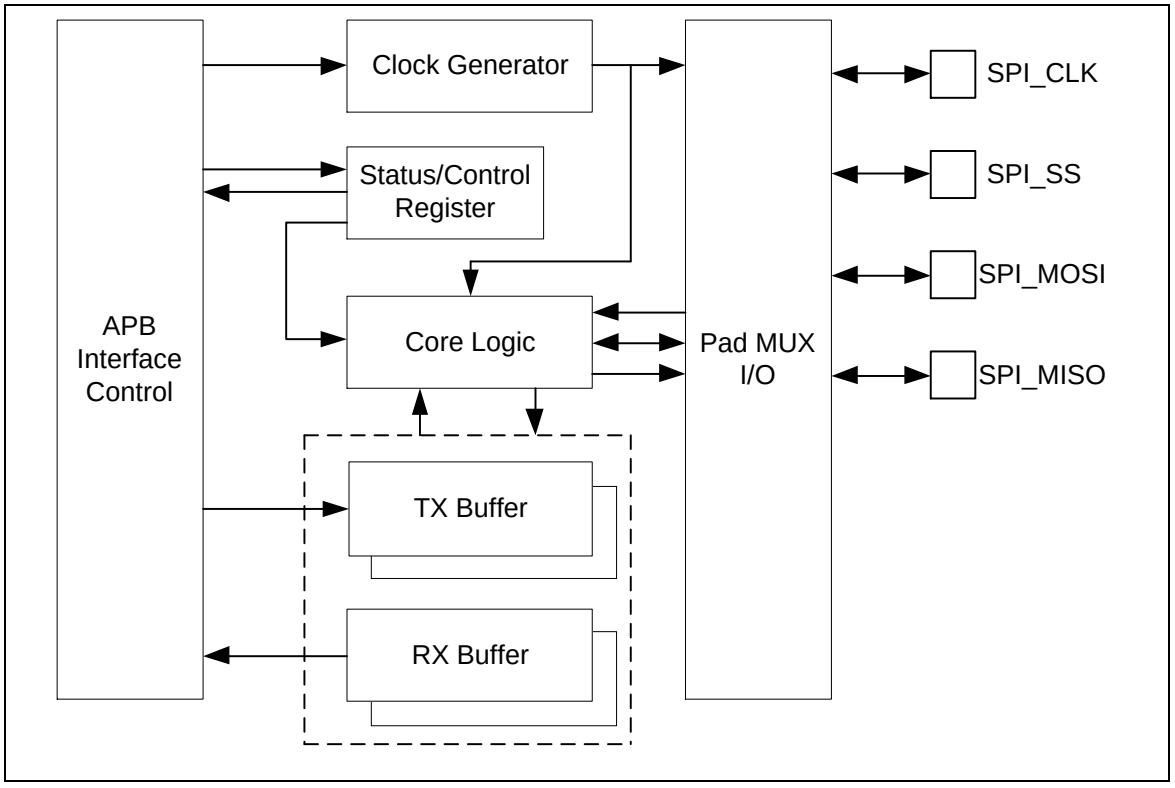


Figure 6.12-1 SPI Block Diagram

6.12.4 Basic Configuration

The SPI pin functions are configured in SYS_P0_MFP register. The SPI peripheral clock can be enabled in CLK_APBCLK[12] and its source can be selected in CLK_CLKSEL1[5:4].

6.12.5 Functional Description

6.12.5.1 Terminology

SPI Peripheral Clock and Serial Bus Clock

The SPI controller needs the SPI peripheral clock to drive the SPI logic unit to perform the data transfer. The SPI bus clock is the clock presented on SPI_CLK pin. The SPI peripheral clock frequency is determined by the settings of clock source, DIVMOD (SPI_SLVCTL[31]) option and clock divisor (SPI_CLKDIV[7:0]). The SPISEL bit of CLK_CLKSEL1 register determines the clock source of the SPI peripheral clock. Set the DIVMOD bit to 0 for the compatible SPI clock frequency calculation of previous products. The DIVIDER setting of SPI_CLKDIV register determines the divisor of the clock rate calculation.

In SPI Master mode, the SPI peripheral clock is equal to the SPI bus clock.

In SPI Slave mode, the SPI bus clock is provided by an off-chip master device. The SPI peripheral clock frequency of slave device must be faster than the bus clock frequency of the master device connected together. The frequency of SPI peripheral clock cannot be faster than the APB clock frequency regardless of Master mode or Slave mode

Master/Slave Mode

The SPI controller can be set as Master or Slave mode by setting the SLAVE bit (SPI_CTL[18]) to communicate with the off-chip SPI slave or master device. The application block diagrams in Master and Slave mode are shown in Figure 6.12-2.

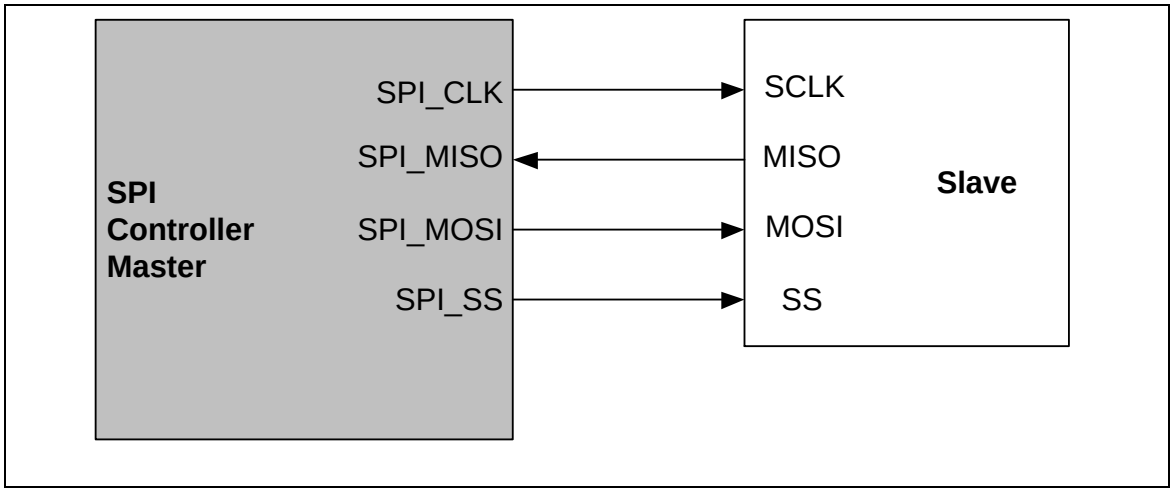


Figure 6.12-2 SPI Master Mode Application Block Diagram

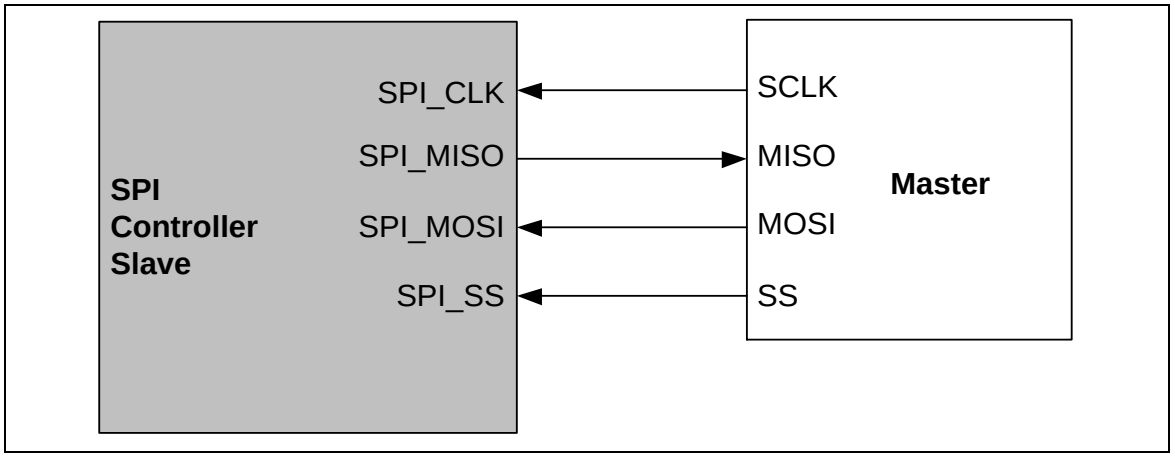


Figure 6.12-3 SPI Slave Mode Application Block Diagram

Clock Polarity

The CLKPOL bit (SPI_CTL[11]) defines the bus clock idle state. If CLKPOL = 1, the output SPI_CLK is high at idle state; otherwise it is at low if CLKPOL = 0.

Transmit/Receive Bit Length

The bit length of a transaction word is defined in DWIDTH bit field (SPI_CTL[7:3]). It can be configured up to 32-bit length in a transaction word to transmitting and receiving.

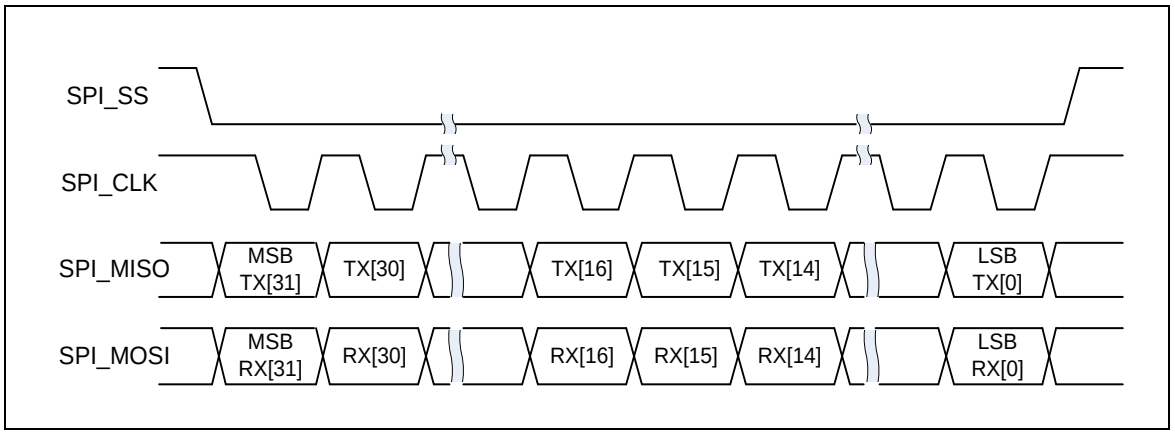


Figure 6.12-4 32-Bit in One Transaction

LSB/MSB First

The LSB bit (SPI_CTL[10]) defines the data transmission either from LSB or MSB firstly to start to transmit/receive data.

Transmit Edge

The TXNEG bit (SPI_CTL[2]) defines the data transmitted out either on negative-edge or on positive-edge of serial clock SPI_CLK.

Receive Edge

The RXNEG bit (SPI_CTL[1]) defines the data received in either on negative-edge or on positive-edge of serial clock SPI_CLK.

Note: the settings of TXNEG and RXNEG are mutual exclusive. In other words, do not transmit and receive data on the same clock edge.

Word Suspend

The four bits field of SUSPITV (SPI_CTL[15:12]) provide a configurable suspend interval 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 serial clock cycles). This SUSPITV setting will not take effect to the word suspend interval if the software disables the FIFO mode.

Slave Selection

In Master mode, the SPI controller can drive one off-chip slave device through the slave select output pin SPI_SS. In Slave mode, the off-chip master device drives the slave select signal from the SPI_SS input pin to the SPI controller. In Master/Slave mode, the active state of slave selected signal can be programmed to low active or high active in SSACTPOL bit (SPI_SSCTL[2]), and the SSLTEN bit (SPI_SSCTL[4]) defines the slave select signal SPI_SS is level-trigger or edge-trigger. The selection of trigger condition depends on what type of peripheral slave/master device is connected.

In Slave mode, if the SSLTEN bit (SPI_SSCTL[4]) is configured as level trigger, the LTF bit (SPI_SSCTL[5]) is used to indicate if both the number of received data and the number of received bits meet the requirement which defined in DWIDTH (SPI_CTL[7:3]) among one transaction done (the transaction done means the unit transfer interrupt flag is set to 1 when the slave select signal is inactivated or the SPI controller finishes one data transfer).

Level-trigger / Edge-trigger

In Slave mode, the slave select signal can be configured as level-trigger or edge-trigger. In edge-trigger, the data transfer starts from an active edge of the slave select signal and ends on an inactive edge of the slave select signal. The unit-transfer interrupt flag UNITIF (SPI_CTL[16]) will be set to 1 as an inactive edge is detected. If master does not send an inactive edge to slave, the transfer procedure will not be completed and the unit-transfer interrupt flag of slave will not be set. In level-trigger, the unit-transfer interrupt flag of slave will be set when one of the following two conditions occurs. The first condition is that if the number of transferred bit matches the settings of DWIDTH (SPI_CTL[7:3]), the unit-transfer interrupt flag of slave will be set. The second condition, if master set the slave select pin to inactive level during the transfer in progress, it will force slave device to terminate the current transfer no matter how many bits have been transferred and the unit-transfer interrupt flag will be set. User can read the status of LTF bit to check if the data has been completely transferred.

6.12.5.2 Automatic Slave Select

In Master mode, if the bit AUTOSS (SPI_SSCTL[3]) is set, the slave select signal will be generated automatically and output to SPI_SS pin according to SS (SPI_SSCTL[0]) whether enabled or not. It means that the slave select signal, which is enabled in SS (SPI_SSCTL[0], is asserted by the SPI controller when transmit/receive is started by setting the SPIEN bit

(SPI_CTL[0]) and is de-asserted after the data transfer is finished. If the AUTOSS bit is cleared, the slave select output signal will be asserted and de-asserted by manual setting and clearing the related bit in SS (SPI_SSCTL[0]). The active level of the slave select output signal is specified in SSACTPOL bit (SPI_SSCTL[2]).

6.12.5.3 Byte Reorder Function

When the transfer is set as MSB first (LSB bit (SPI_CTL[10]) = 0) and the byte reorder function is enabled, the data stored in the TX buffer and RX buffer will be rearranged in the order as [Byte0, Byte1, Byte2, Byte3] when the bit length is configured as 32-bit (DWIDTH (SPI_CTL[7:3]) = 0). The sequence of transmitted/received data will be Byte0, Byte1, Byte2, and then Byte3. If the DWIDTH (SPI_CTL[7:3])) is set to 24-bit, 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 (SPI_CTL[7:3]) is configured as 16, 24, or 32 bits.

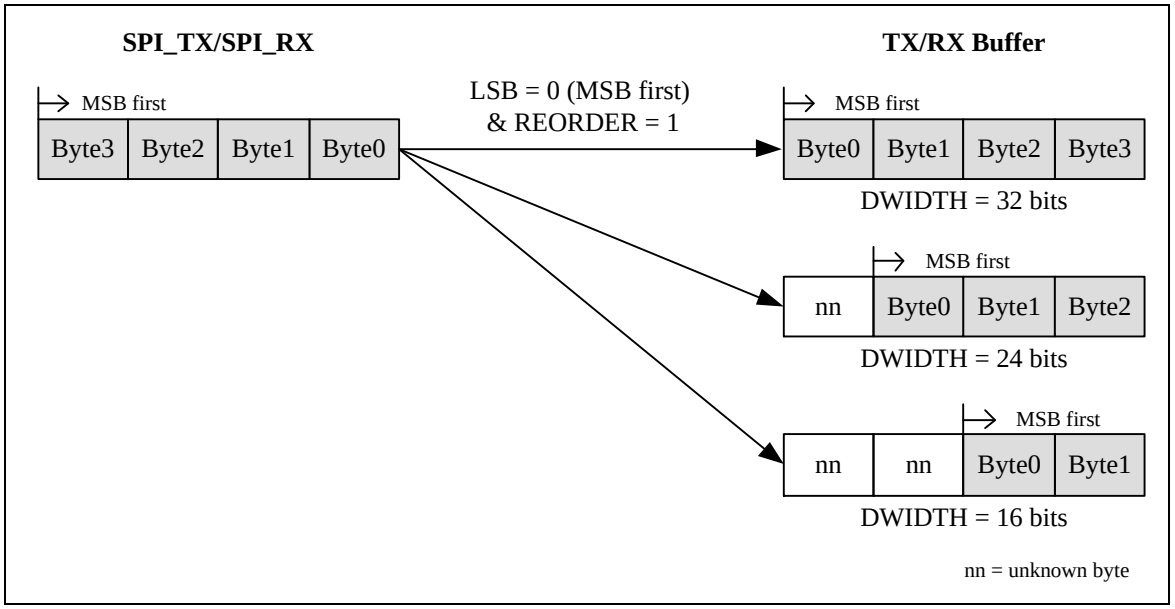


Figure 6.12-5 Byte Reorder

6.12.5.4 Byte Suspend Function

Both settings of byte suspend interval and word suspend interval are configured in SUSPITV (SPI_CTL[15:12]). In Master mode, if byte reorder function is enabled by setting REORDER (SPI_CTL[19]) to 1, the hardware will insert a suspend interval of 0.5 ~ 15.5 serial clock periods between two successive bytes in a transaction word. The setting of DWIDTH (SPI_CTL[7:3]) can be configured as 16, 24 or 32 bits.

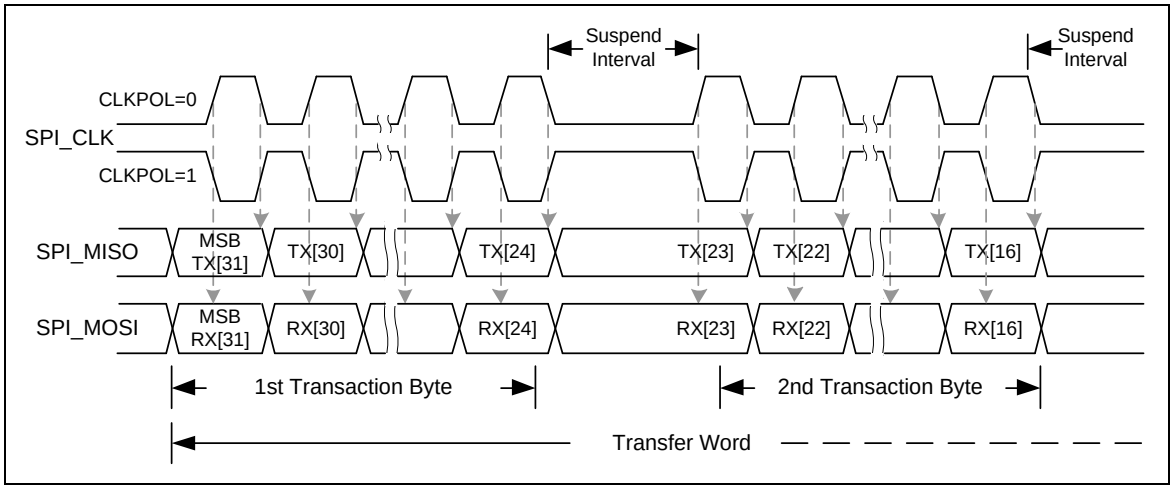


Figure 6.12-6 Timing Waveform for Byte Suspend

6.12.5.5 Slave 3-Wire Mode

When the SLV3WIRE bit (SPI_SLVCTL[8]) is set by software to enable the Slave 3-wire mode, the SPI controller can work with no slave select signal in Slave mode. The SLV3WIRE bit only takes effect in Slave mode. Only three pins, SPI_CLK, SPI_MISO, and SPI_MOSI, are required to communicate with a SPI master. The SPI_SS pin can be configured as a GPIO. When the SLV3WIRE bit is set to 1, the SPI slave will be ready to transmit/receive data after the SPIEN bit is set to 1. As the number of received bits meets the requirement which defined in DWIDTH (SPI_CTL[7:3]), the unit-transfer interrupt flag UNITIF (SPI_CTL[16]) will be set to 1.

Note: In Slave 3-wire mode, the SSLTEN bit (SPI_SSCTL[4]) should be set as 1.

6.12.5.6 FIFO Mode

The SPI controller supports FIFO mode when the FIFOEN bit, SPI_CTL[21], is set as 1. The SPI controller equips with four 32-bit wide transmit and receive FIFO buffers.

The transmit FIFO buffer is a 4-layer depth, 32-bit wide, first-in and first-out register buffer. The software can write data to the transmit FIFO buffer by writing the SPI_TX register. The data stored in the transmit FIFO buffer will be read and sent out by the transmission control logic. If the 4-layer transmit FIFO buffer is full, the TXFULL (SPI_CTL[27]) bit will be set to 1. When the SPI transmission logic unit draws out the last datum of the transmit FIFO buffer, so that the 4-layer transmit FIFO buffer is empty, the TXEMPTY (SPI_CTL[26]) bit will be set to 1. Notice that the TXEMPTY flag (SPI_CTL[26]) is set to 1 while the last transaction is still in progress.

The received FIFO buffer is also a 4-layer depth, 32-bit wide, first-in and first-out register buffer. The receive control logic will store the received data to this buffer. The software can read the FIFO buffer data from SPI_RX register. There are FIFO related status bits, like RXEMPTY (SPI_CTL[24]) and RXFULL (SPI_CTL[25]), to indicate the current status of received FIFO buffer.

In FIFO mode, the software can set the transmitting and receiving threshold by setting the TXTH (SPI_FIFOCTL[29:28]) and RXTH (SPI_FIFOCTL[25:24]) settings. When the count of valid data stored in transmit FIFO buffer is less than or equal to TXTH setting, the TXTHIF bit (SPI_STATUS[4]) will be set to 1. When the count of valid data stored in receive FIFO buffer is larger than RXTH setting, the RXTHIF bit (SPI_STATUS[0]) will be set to 1.

In FIFO mode, the software can write 4 datum to the SPI transmit FIFO buffer in advance. When the SPI controller operates with FIFO mode, the SPIEN bit of SPI_CTL register will be controlled by hardware, software should not modify the content of SPI_CTL register unless clearing the FIFOEN bit to disable the FIFO mode.

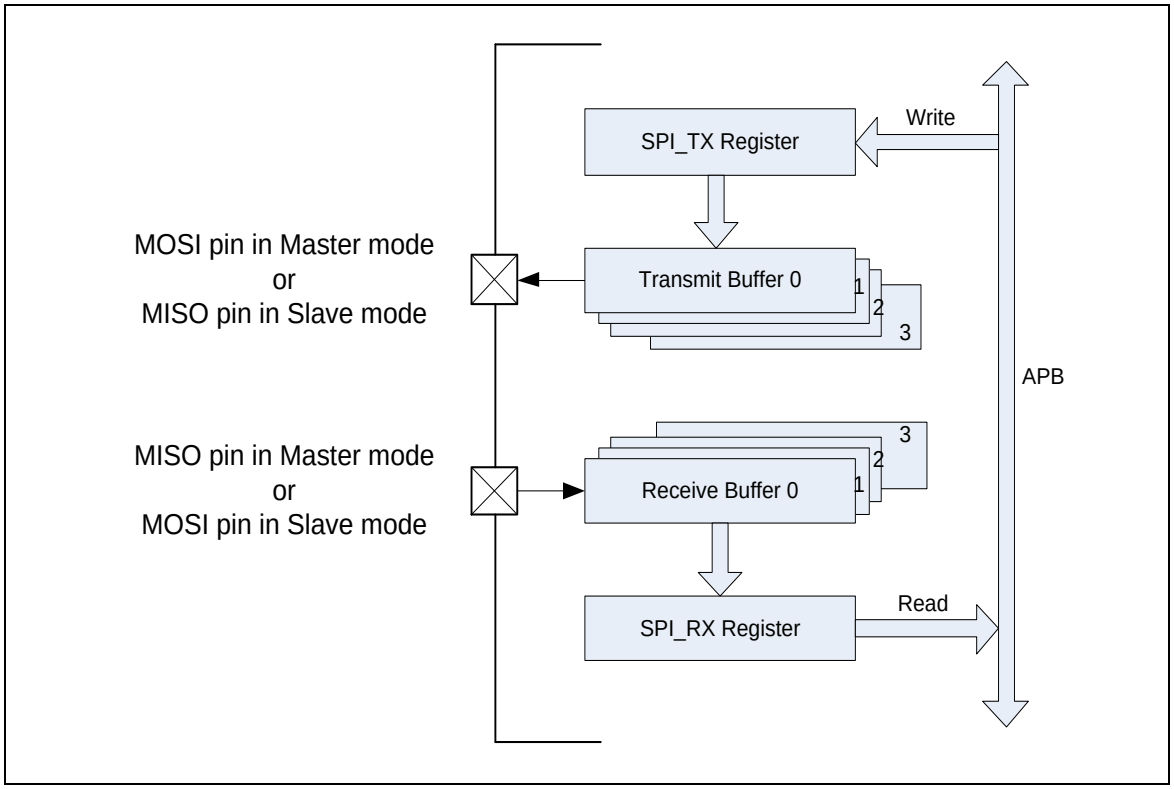


Figure 6.12-7 FIFO Mode Block Diagram

SPI Master mode and FIFO mode Enabled

In Master mode transmission operation, the TXEMPTY (SPI_CTL[26]) flag will be cleared to 0 when the FIFOEN bit (SPI_CTL[21]) is set to 1 and the software writes the first data to the SPI_TX register. The transmission starts immediately as long as the transmit FIFO buffer is not empty. User can write the next data into SPI_TX register immediately. The SPI controller will insert a suspend interval between two successive transactions in FIFO mode and the period of suspend interval is decided by the setting of SUSPITV (SPI_CTL [15:12]). User can write data into SPI_TX register as long as the TXFULL (SPI_CTL[27]) flag is 0.

The subsequent transactions will be triggered automatically if the transmitted data are updated in time. If the SPI_TX register does not be updated after all data transfer are done, the transfer will stop.

In Master mode reception operation, the serial data are received from SPI_MISO pin and stored to the receive FIFO buffer. The RXEMPTY (SPI_CTL[24]) flag will be cleared to 0 while the receive FIFO buffer contain unread data. The software can read the received data from SPI_RX register as long as the RXEMPTY (SPI_CTL[24]) flag is 0. If the receive FIFO buffer contain 4 unread data, the RXFULL (SPI_CTL[25]) flag will be set to 1. The SPI controller will stop receiving data until the software read the SPI_RX register.

SPI Slave mode and FIFO mode Enabled

In Slave mode, when the FIFOEN bit (SPI_CTL[21]) is set as 1, the SPIEN bit (SPI_CTL[0]) will be set as 1 by hardware automatically. If user wants to stop the slave mode SPI data transfer, both the FIFOEN bit and SPIEN bit must be cleared to 0 by software.

In Slave mode transmission operation, when the software writes data to SPI_TX register, the data

will be loaded into transmit FIFO buffer and the TXEMPTY flag (SPI_CTL[26]) will be set to 0. The transmission will start when the slave device receives clock signal from master. The software can write data to SPI_TX register as long as TXFULL flag (SPI_CTL[27]) is 0. After all data have been drawn out by the SPI transmission logic unit and the software does not update the SPI_TX register, the TXEMPTY flag (SPI_CTL[26]) will be set to 1.

In Slave mode reception operation, the serial data is received from SPI_MOSI pin and stored to receive FIFO buffer. The reception mechanism is similar to master mode reception operation.

6.12.5.7 Interrupt

SPI unit-transfer interrupt

As the SPI controller finishes a unit transfer, the unit transfer interrupt flag UNITIF (SPI_CTL[16]) will be set to 1. The unit transfer interrupt event will generate an interrupt to CPU if the unit transfer interrupt enable bit UNITIEN (SPI_CTL[17]) is set. The unit transfer interrupt flag can be cleared only by writing 1 to it.

SPI slave 3-wire mode start interrupt

In 3-wire mode, the Slave 3-wire mode start interrupt flag, SLVSTIF (SPI_SLVCTL[11]), will be set to 1 when the slave senses the SPI bus clock signal. The SPI controller will issue an interrupt if the SLVSTIEN (SPI_SLVCTL[10]) is set to 1. If the count of the received bits is less than the setting of DWIDTH (SPI_CTL[7:3]) and there is no more SPI bus clock input over the expected time period which is defined by the user, the user can set the SLVABT bit (SPI_SLVCTL[9]) to abort the current transfer. The unit-transfer interrupt flag, UNITIF, will be set to 1 if the SLVABT bit is set to 1 by software.

Receive FIFO time-out interrupt

In FIFO mode, there is time-out function to inform user. If there is a received data in the FIFO and it does not be 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 time-out interrupt to the system if the time-out interrupt enable bit RXTOIEN (SPI_FIFOCTL[21]) 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 (SPI_FIFOCTL[29:28]), the transmit FIFO interrupt flag 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 (SPI_FIFOCTL[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 (SPI_FIFOCTL[25:24]), the receive FIFO interrupt flag 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 (SPI_FIFOCTL[2]), is set to 1.

6.12.6 Timing Diagram

The active state of slave select signal can be defined by the settings of SSACTPOL bit (SPI_SSCTL[2]) and SSLTEN bit (SPI_SSCTL[4]). The serial clock (SPI_CLK) idle state can be

configured as high state or low state by setting the CLKPOL bit (SPI_CTL[11]). It also provides the bit length of a transaction word in DWIDTH (SPI_CTL[7:3]), and transmit/receive data from MSB or LSB first in LSB bit (SPI_CTL[10]). User also can select which edge of bus clock to transmit/receive data in TXNEG/RXNEG (SPI_CTL[2:1]). Four SPI timing diagrams for master/slave operations and the related settings are shown below.

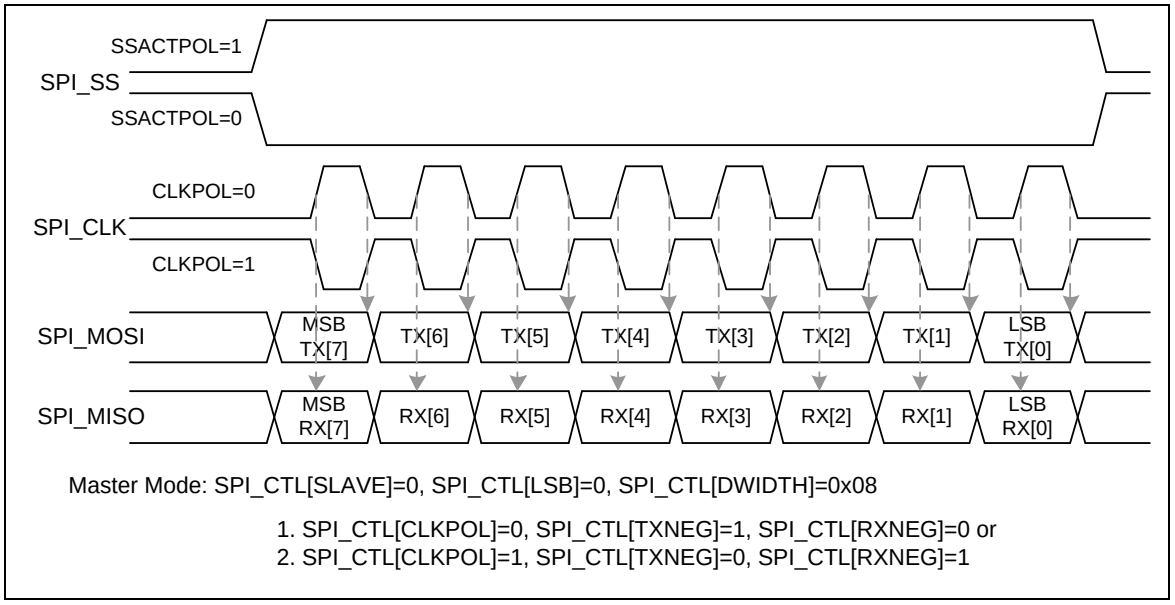


Figure 6.12-8 SPI Timing in Master Mode

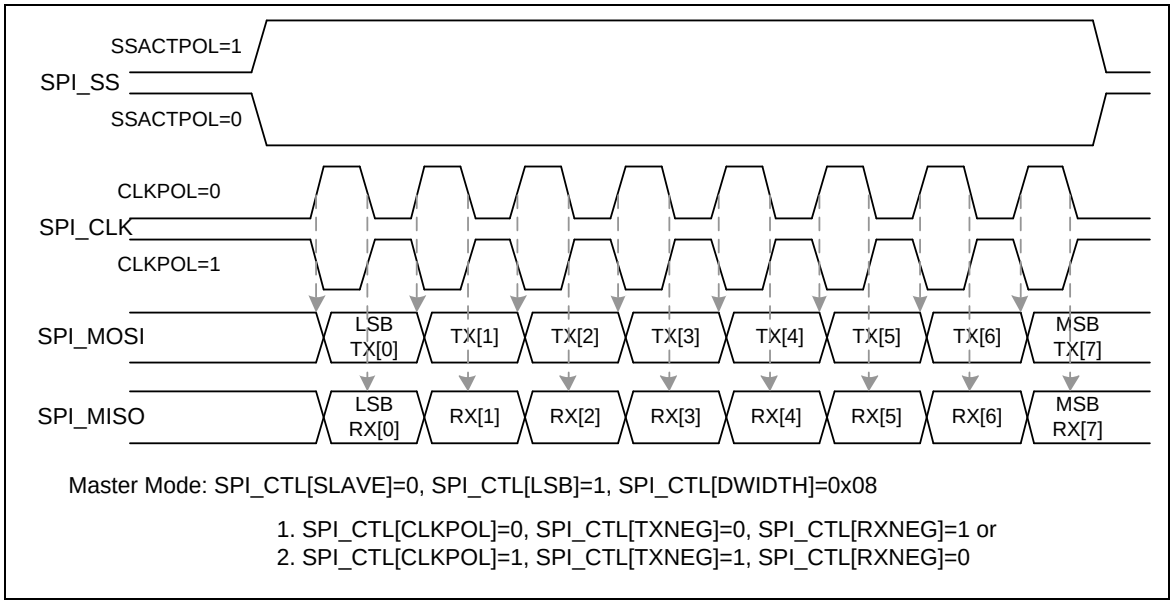


Figure 6.12-9 SPI Timing in Master Mode (Alternate Phase of SPI_CLK)

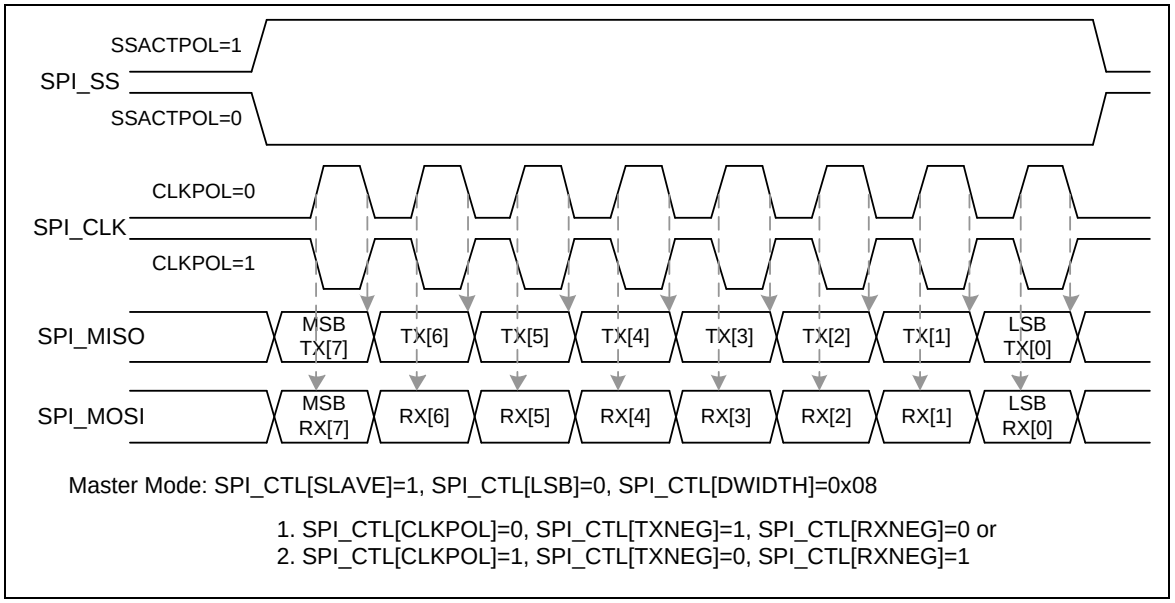


Figure 6.12-10 SPI Timing in Slave Mode

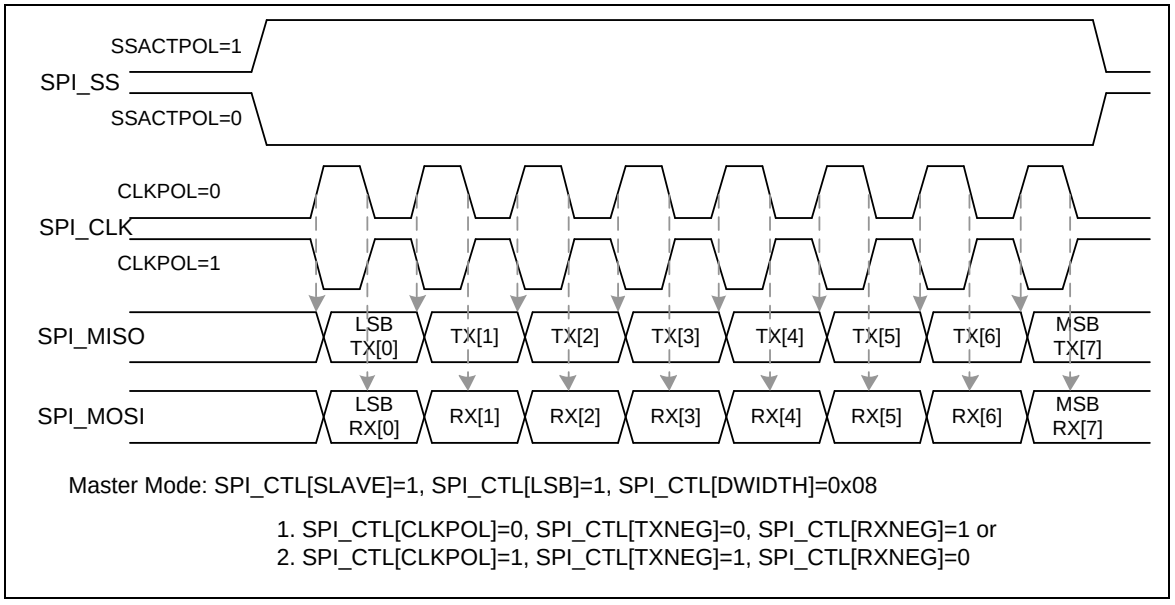


Figure 6.12-11 SPI Timing in Slave Mode (Alternate Phase of SPI_CLK)

6.12.7 Programming Examples

Example 1: SPI controller is set as a master to access an off-chip slave device with following specifications:

- Data bit is latched at positive-edge of bus clock
- Data bit is driven at negative-edge of bus clock
- Data bit is transferred from MSB first
- Data width is 8 bits

- SPI_CLK is low at idle state
- Only one byte data is transmitted/received in a transaction
- Connect with an off-chip slave device. Slave select signal is active low

The operation flow is described follows.

1) Set the DIVIDER (SPI_CLKDIV[7:0]) to determine the output frequency of bus clock.
2) Write the related settings into the SPI_CTL register to control this SPI master actions
    1. Set this SPI controller as master device in SLAVE bit (SPI_CTL[18] = 0)
    2. Force the serial clock low at idle state in CLKPOL bit (SPI_CTL[11] = 0)
    3. Select data transmitted at negative edge of bus clock in TXNEG bit (SPI_CTL[2] = 1)
    4. Select data latched at positive edge of bus clock in RXNEG bit (SPI_CTL[1] = 0)
    5. Set the bit length of word transfer as 8-bit in DWIDTH bit field (SPI_CTL[7:3] = 0x08)
    6. Set MSB transfer first in LSB bit (SPI_CTL[10] = 0), and does not need to care the SUSPITV bit field (SPI_CTL[15:12]) due to it is not in FIFO mode in this case.
3) Write the SPI_SSCTL register a proper value for the related settings of Master mode.
    1. Clear the Automatic Slave Select bit AUTOSS(SPI_SSCTL[3] = 0).

       Select low level trigger output of slave select signal in the Slave Select Active Level control bit, SSACTPOL (SPI_SSCTL[2] = 0).
    2. Set the slave select signal to be active by setting the Slave Select control bit SS (SPI_SSCTL[0]) to active the off-chip slave device.
4) 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 SPI_TX register.
5) If this SPI master just only attempts to receive (read) one byte data from the off-chip slave device and does not care what data will be transmitted, the software does not need to update the SPI_TX register.
6) Set the SPIEN bit (SPI_CTL[0] = 1) to start the data transfer on the SPI interface.
7) Waiting for SPI interrupt (if the Unit-Transfer Interrupt Enable UNITIEN bit (SPI_CTL[17]) is set) or just polling the SPIEN bit (SPI_CTL[0]) till it is cleared to 0 by hardware automatically.
8) Read out the received one byte data from SPI_RX[7:0].
9) Go to 4) to continue another data transfer or set SS bit (SPI_SSCTL [0]) to 0 to inactivate the off-chip slave device.

Example 2: The SPI controller is set as a slave device that is connected by 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 at positive-edge of bus clock
- Data bit is driven at negative-edge of bus clock
- Data bit is transferred from LSB first
- Data width is 8 bits
- SPI_CLK is high at idles state
- Only one byte data is transmitted/received in a transaction

- Slave select signal is high level trigger

The operation flow is as follows.

1) Set the DIVIDER (SPI_CLKDIV[7:0]) to determine the slave peripheral clock frequency. The slave peripheral clock frequency must be larger than the SPI bus clock frequency.
2) Write the SPI_SSCTL register a proper value for the related settings of Slave mode.

   Select high level and level trigger for the input of slave select signal by setting the Slave Select Active Level control bit SSACTPOL (SPI_SSCTL[2] = 1) and the Slave Select Level Trigger Enable bit SSLTEN (SPI_SSCTL[4] = 1).
3) Write the related settings into the SPI_CTL register to control this SPI slave actions
   1. Set this SPI controller as slave device in SLAVE bit (SPI_CTL[18] = 1)
   2. Select the serial clock as high at idle state in CLKPOL bit (SPI_CTL[11] = 1)
   3. Select data transmitted at negative-edge of serial clock in TXNEG bit (SPI_CTL[2] = 1)
   4. Select data latched at positive-edge of serial clock in RXNEG bit (SPI_CTL[1] = 0)
   5. Set the bit length of word transfer as 8 bits in DWIDTH bit field (SPI_CTL[7:3] = 0x08)
   6. Set LSB transfer first in LSB bit (SPI_CTL[10] = 1)
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 SPI_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 SPI_TX does not need to be updated by software.
6) Set the SPIEN bit (SPI_CTL[0] = 1) to wait for the slave select trigger input and bus clock input from the off-chip master device to start the data transfer on the SPI interface.
7) Waiting for SPI interrupt (if the Unit-Transfer Interrupt Enable UNITIEN bit (SPI_CTL[17]) is set) or just polling the SPIEN bit (SPI_CTL[0]) until it is cleared to 0 by hardware automatically.
8) Read out the received one byte data from SPI_RX[7:0].
9) Go to 4) to continue another data transfer or clear the SPIEN bit (SPI_CTL[0]) to stop data transfer.

6.12.8 Register Map

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

Register	Offset	R/W	Description	Reset Value
SPI Base Address: **SPI_BA = 0x4003_0000**				
SPI_CTL	SPI_BA+0x00	R/W	SPI Control and Status Register	0x0500_3004
SPI_CLKDIV	SPI_BA+0x04	R/W	SPI Clock Divider Register	0x0000_0000
SPI_SSCTL	SPI_BA+0x08	R/W	SPI Slave Select Register	0x0000_0000
SPI_RX	SPI_BA+0x10	R	SPI Data Receive Register	0x0000_0000
SPI_TX	SPI_BA+0x20	W	SPI Data Transmit Register	0x0000_0000
SPI_SLVCTL	SPI_BA+0x3C	R/W	SPI Slave Control and Status Register	0x0000_0000
SPI_FIFOCTL	SPI_BA+0x40	R/W	SPI FIFO Control Register	0x2200_0000
SPI_STATUS	SPI_BA+0x44	R/W	SPI Status Register	0x0500_0000

6.12.9 Register Description

SPI Control and Status Register (SPI_CTL)

Register	Offset	R/W	Description	Reset Value
SPI_CTL	SPI_BA+0x00	R/W	SPI Control and Status Register	0x0500_3004

31	30	29	28	27	26	25	24
Reserved				TXFULL	TXEMPTY	RXFULL	RXEMPTY
23	22	21	20	19	18	17	16
Reserved		FIFOEN	Reserved	REORDER	SLAVE	UNITIEN	UNITIF
15	14	13	12	11	10	9	8
SUSPITV				CLKPOL	LSB	Reserved	
7	6	5	4	3	2	1	0
DWIDTH					TXNEG	RXNEG	SPIEN

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27]	**TXFULL**	**Transmit FIFO Buffer Full Indicator (Read Only)** 0 = The transmit FIFO buffer is not full. 1 = The transmit FIFO buffer is full. **Note:** It's a mutual mirror bit of SPI_STATUS[27].
[26]	**TXEMPTY**	**Transmit FIFO Buffer Empty Indicator (Read Only)** 0 = The transmit FIFO buffer is not empty. 1 = The transmit FIFO buffer is empty. **Note:** It's a mutual mirror bit of SPI_STAUTS[26].
[25]	**RXFULL**	**Receive FIFO Buffer Full Indicator (Read Only)** 0 = The receive FIFO buffer is not full. 1 = The receive FIFO buffer is full. **Note:** It's a mutual mirror bit of SPI_STATUS[25]
[24]	**RXEMPTY**	**Receive FIFO Buffer Empty Indicator (Read Only)** 0 = The receive FIFO buffer is not empty. 1 = The receive FIFO buffer is empty. **Note:** It's a mutual mirror bit of SPI_STATUS[24].
[23:22]	**Reserved**	Reserved.
[21]	**FIFOEN**	**FIFO Mode Enable Bit** 0 = FIFO Mode Disabled. 1 = FIFO Mode Enabled. **Note 1:** Before enabling FIFO mode, the other related settings should be set in advance. **Note 2:** In Master mode, if the FIFO mode is enabled, the SPIEN bit will be set to 1 automatically after writing data into the 4-layer depth transmit FIFO. When all data stored

		in transmit FIFO buffer are transferred, the SPIEN bit will back to 0.
[20]	**Reserved**	Reserved.
[19]	**REORDER**	**Byte Reorder Function** 0 = Byte reorder function Disabled. 1 = Byte reorder function Enabled. **Note:** This setting is only available if DWIDTH is defined as 16, 24, or 32 bits.
[18]	**SLAVE**	**Slave Mode Control** 0 = Master mode. 1 = Slave mode.
[17]	**UNITIEN**	**Unit-transfer Interrupt Enable Bit** 0 = SPI unit-transfer interrupt Disabled. 1 = SPI unit-transfer interrupt Enabled.
[16]	**UNITIF**	**Unit-transfer Interrupt Flag** 0 = The transfer does not finish yet. 1 = The SPI controller has finished one unit transfer. **Note 1:** This bit will be cleared by writing 1 to itself. **Note 2:** It's a mutual mirror bit of SPI_STATUS[16].
[15:12]	**SUSPITV**	**Suspend Interval (Master Only)** The four bits provide configurable suspend interval between two successive transactions 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) * period of SPI_CLK clock cycle Example: SUSPITV = 0x0 …. 0.5 SPI_CLK clock cycle. SUSPITV = 0x1 …. 1.5 SPI_CLK clock cycle. …… SUSPITV = 0xE …. 14.5 SPI_CLK clock cycle. SUSPITV = 0xF …. 15.5 SPI_CLK clock cycle.
[11]	**CLKPOL**	**Clock Polarity** 0 = SPI_CLK idle low. 1 = SPI_CLK idle high.
[10]	**LSB**	**LSB First** 0 = The MSB is transmitted/received first. 1 = The LSB is transmitted/received first.
[9:8]	**Reserved**	Reserved.
[7:3]	**DWIDTH**	**Transmit Bit Length** This field specifies how many bits are transmitted in one transmit/receive. The minimum bit length is 8 bits and can up to 32 bits. DWIDTH = 0x01~0x07 …. reserved (can't use). DWIDTH = 0x08 …. 8 bits. DWIDTH = 0x09 …. 9 bits. …… DWIDTH = 0x1F …. 31 bits. DWIDTH = 0x00 …. 32 bits.

[2]	**TXNEG**	**Transmit On Negative Edge** 0 = The transmitted data output signal is driven on the rising-edge of SPI_CLK. 1 = The transmitted data output signal is driven on the falling-edge of SPI_CLK.
[1]	**RXNEG**	**Receive On Negative Edge** 0 = The received data input signal latched on the rising-edge of SPI_CLK. 1 = The received data input signal latched on the falling-edge of SPI_CLK.
[0]	**SPIEN**	**SPI Transfer Control Bit And Busy Status** If FIFO mode is enabled, this bit will be controlled by hardware and it's read only. If FIFO mode is disabled, during the data transfer, this bit keeps the value of 1. As the transfer is finished, this bit will be cleared automatically. 0 = Writing 0 to this bit to stop data transfer if SPI is transferring. 1 = In Master mode, writing 1 to this bit to start the SPI data transfer; in Slave mode, writing 1 to this bit indicates that the slave is ready to communicate with a master. **Note 1:** When FIFO mode is disabled, all configurations should be ready before writing 1 to the SPIEN bit. **Note 2:** In SPI Slave mode, if FIFO mode is disabled and the SPI bus clock is kept at idle state during a data transfer, the SPIEN bit will not be cleared to 0 when slave select signal goes to inactive state.

SPI Clock Divider Register (SPI_CLKDIV)

Register	Offset	R/W	Description	Reset Value
SPI_CLKDIV	SPI_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**	**Clock Divider Register (Master Only)** The value in this field is the frequency divider to determine the SPI peripheral clock frequency f_{spi}, and the SPI master's bus clock frequency on the SPI_CLK output pin. The frequency is obtained according to the following equation: If the bit of DIVMOD, SPI_SLVCTL[31], is set to 0. $f_{spi} = \frac{f_{SPI_clock_src}}{(DIVIDER+1)*2}$ else if DIVMOD is set to 1, $f_{spi} = \frac{f_{SPI_clock_src}}{(DIVIDER+1)}$ where $f_{SPI_clock_src}$ is the SPI peripheral clock source which is defined in the CLK_CLKSEL1 register.

SPI Slave Select Register (SPI_SSCTL)

Register	Offset	R/W	Description	Reset Value
SPI_SSCTL	SPI_BA+0x08	R/W	SPI Slave 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		LTF	SSLTEN	AUTOSS	SSACTPOL	Reserved	SS

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5]	**LTF**	**Level Trigger Flag (Read Only, Slave Only)** When the SSLTEN bit is set in Slave mode, this bit can be read to indicate the received bit number is met the requirement or not. 0 = The transaction number or the transferred bit length of one transaction does not meet the specified requirements. 1 = The transaction number and the transferred bit length meet the specified requirements which defined in DWIDTH.
[4]	**SSLTEN**	**Slave Select Level Trigger Enable Bit (Slave Only)** 0 = The input slave select signal is edge-trigger. 1 = The input slave select signal is level-trigger.
[3]	**AUTOSS**	**Automatic Slave Selection Function Enable Bit (Master Only)** 0 = SPI_SS pin signal will be asserted/de-asserted by setting /clearing SS bit. 1 = SPI_SS pin signal will be generated automatically by hardware, which means that slave select signal will be asserted by the SPI controller when transmit/receive is started by setting SPIEN, and will be de-asserted after each transmit/receive is finished.
[2]	**SSACTPOL**	**Slave Select Active Level (Slave Only)** It defines the active status of slave select signal (SPI_SS). If SSLTEN bit is 1: 0 = The slave select signal SPI_SS is active at low-level. 1 = The slave select signal SPI_SS is active at high-level. If SSLTEN bit is 0: 0 = The slave select signal SPI_SS is active at falling-edge. 1 = The slave select signal SPI_SS is active at rising-edge.
[1]	**Reserved**	Reserved.
[0]	**SS**	**Slave Select Control Bits (Master Only)** If AUTOSS bit is 0, 0 = Set the SPI_SS line to inactive state.

		1 = Set the SPI_SS line to active state. If AUTOSS bit is 1, 0 = Keep the SPI_SS line at inactive state. 1 = Select the SPI_SS line to be automatically driven to active state for the duration of transmission/reception, and will be driven to inactive state for the rest of the time. The active state of SPI_SS is specified in SSACTPOL bit.

SPI Data Receive Register (SPI_RX)

Register	Offset	R/W	Description	Reset Value
SPI_RX	SPI_BA+0x10	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]	**RX**	**Data Receive Register (Read Only)** The Data Receive Register holds the value of received data of the last executed transfer. Valid bits depend on the transmit bit length field DWIDTH in the SPI_CTL register. For example, if DWIDTH is set to 0x08, the bit field RX[7:0] holds the received data. The values of the other bits are unknown. The Data Receive Register is read-only register.

SPI Data Transmit Register (SPI_TX)

Register	Offset	R/W	Description	Reset Value
SPI_TX	SPI_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]	**TX**	**Data Transmit Register** The Data Transmit Register holds the data to be transmitted in the next transfer. Valid bits depend on the transmit bit length field DWIDTH in the SPI_CTL register. For example, if DWIDTH is set to 0x08, the bit filed TX[7:0] will be transmitted in next transfer.

SPI Slave Control and Status Register (SPI_SLVCTL)

Register	Offset	R/W	Description	Reset Value
SPI_SLVCTL	SPI_BA+0x3C	R/W	SPI Slave Control and Status Register	0x0000_0000

31	30	29	28	27	26	25	24
DIVMOD	Reserved						
23	22	21	20	19	18	17	16
Reserved							SSINAIEN
15	14	13	12	11	10	9	8
Reserved				SLVSTIF	SLVSTIEN	SLVABT	SLV3WIRE
7	6	5	4	3	2	1	0
Reserved							

Bits	Description	
[31]	**DIVMOD**	**Clock Configuration Backward Compatible Option** 0 = The clock configuration is backward compatible. 1 = The clock configuration is not backward compatible. **Note:** Refer to the description of SPI_CLKDIV register for details.
[30:17]	**Reserved**	Reserved.
[16]	**SSINAIEN**	**Slave Select Inactive Interrupt Option (Slave Only)** 0 = As the slave select signal goes to inactive level, the IF bit will NOT be set to 1. 1 = As the slave select signal goes to inactive level, the IF bit will be set to 1. **Note:** This setting is only available if the SPI controller is configured as level trigger in slave device.
[15:12]	**Reserved**	Reserved.
[11]	**SLVSTIF**	**Slave 3-wire Mode Start Interrupt Status (Slave Only)** This bit dedicates if a transaction has started in Slave 3-wire mode. 0 = Slave does not detect any SPI bus clock transfer since the SLVSTIEN bit was set to 1. 1 = The transfer has started in Slave 3-wire mode. **Note 1:** It will be cleared automatically when a transaction is done or by writing 1 to this bit. **Note 2:** It is a mutual mirror bit of SPI_STATUS[11].
[10]	**SLVSTIEN**	**Slave 3-wire Mode Start Interrupt Enable (Slave Only)** It is used to enable interrupt when the transfer has started in Slave 3-wire mode. If there is no transfer done interrupt over the time period which is defined by user after the transfer start, user can set the SLVABT bit to force the transfer done. 0 = Transaction start interrupt Disabled. 1 = Transaction start interrupt Enabled. **Note:** It will be cleared to 0 as the current transfer is done or the SLVSTIF bit is cleared to 0.
[9]	**SLVABT**	**Slave 3-wire Mode Abort Control Bit (Slave Only)**

		In normal operation, there is an interrupt event when the number of received bits meets the requirement which defined in DWIDTH. If the number of received bits is less than the requirement and there is no more bus clock input over one transfer time in Slave 3-wire mode, user can set this bit to force the current transfer done and then user can get a unit transfer interrupt event. 0 = No force the transfer done when the SLV3WIRE bit is set to 1. 1 = Force the transfer done when the SLV3WIRE bit is set to 1. **Note:** This bit will be cleared to 0 automatically by hardware after it is set to 1 by software.
[8]	**SLV3WIRE**	**Slave 3-wire Mode Enable Bit (Slave Only)** The SPI controller work with 3-wire interface including SPI_CLK, SPI_MISO, and SPI_MOSI. 0 = The controller is 4-wire bi-direction interface in Slave mode. 1 = The controller is 3-wire bi-direction interface in Slave mode. The controller will be ready to transmit/receive data after the SPIEN bit is set to 1. **Note:** In Slave 3-wire mode, the SSLTEN bit (SPI_SSCTL[4]) shall be set as 1.
[7:0]	**Reserved**	Reserved.

SPI FIFO Control Register (SPI_FIFOCTL)

Register	Offset	R/W	Description	Reset Value
SPI_FIFOCTL	SPI_BA+0x40	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		RXTOIEN	Reserved				
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved	RXOVIEN	Reserved		TXTHIEN	RXTHIEN	TXRST	RXRST

Bits	Description	
[31:30]	**Reserved**	Reserved.
[29:28]	**TXTH**	**Transmit FIFO Threshold** If the valid data count of the transmit FIFO buffer is less than or equal to the TXTH setting, the TXTHIF bit will be set to 1, else the TXTHIF bit will be cleared to 0.
[27:26]	**Reserved**	Reserved.
[25:24]	**RXTH**	**Receive FIFO Threshold** If the valid data count of the receive FIFO buffer is larger than the RXTH setting, the RXTHIF bit will be set to 1, else the RXTHIF bit will be cleared to 0.
[23:22]	**Reserved**	Reserved.
[21]	**RXTOIEN**	**Receive FIFO Time-out Interrupt Enable Bit** 0 = Time-out interrupt Disabled. 1 = Time-out interrupt Enabled.
[20:7]	**Reserved**	Reserved.
[6]	**RXOVIEN**	**Receive FIFO Overrun Interrupt Enable Bit** 0 = Receive FIFO overrun interrupt Disabled. 1 = Receive FIFO overrun interrupt Enabled.
[5:4]	**Reserved**	Reserved.
[3]	**TXTHIEN**	**Transmit Threshold Interrupt Enable Bit** 0 = Transmit threshold interrupt Disabled. 1 = Transmit threshold interrupt Enabled.
[2]	**RXTHIEN**	**Receive Threshold Interrupt Enable Bit** 0 = Receive threshold interrupt Disabled. 1 = Receive threshold interrupt Enabled.
[1]	**TXRST**	**Clear Transmit FIFO Buffer** 0 = No effect.

		1 = Clear transmit FIFO buffer. **Note:** This bit will be cleared to 0 by hardware after software sets it to 1 and the transmit FIFO is cleared.
[0]	**RXRST**	**Clear Receive FIFO Buffer** 0 = No effect. 1 = Clear receive FIFO buffer. **Note:** This bit will be cleared to 0 by hardware after software sets it to 1 and the receive FIFO is cleared.

SPI Status Register (SPI_STATUS)

Register	Offset	R/W	Description	Reset Value
SPI_STATUS	SPI_BA+0x44	R/W	SPI Status Register	0x0500_0000

31	30	29	28	27	26	25	24
TXCNT				TXFULL	TXEMPTY	RXFULL	RXEMPTY
23	22	21	20	19	18	17	16
Reserved			SLVTOIF	Reserved			UNITIF
15	14	13	12	11	10	9	8
RXCNT				SLVSTIF	Reserved		
7	6	5	4	3	2	1	0
Reserved			TXTHIF	Reserved	RXOVIF	Reserved	RXTHIF

Bits	Description	
[31:28]	**TXCNT**	**Transmit FIFO Data Count (Read Only)** Indicates the valid data count of transmit FIFO buffer.
[27]	**TXFULL**	**Transmit FIFO Buffer Full Indicator (Read Only)** 0 = The transmit FIFO buffer is not full. 1 = The transmit FIFO buffer is full. **Note:** It's a mutual mirror bit of SPI_CTL[27].
[26]	**TXEMPTY**	**Transmit FIFO Buffer Empty Indicator (Read Only)** 0 = The transmit FIFO buffer is not empty. 1 = The transmit FIFO buffer is empty. **Note:** It's a mutual mirror bit of SPI_CTL[26].
[25]	**RXFULL**	**Receive FIFO Buffer Full Indicator (Read Only)** 0 = The receive FIFO buffer is not full. 1 = The receive FIFO buffer is full. **Note:** It's a mutual mirror bit of SPI_CTL[25].
[24]	**RXEMPTY**	**Receive FIFO Buffer Empty Indicator (Read Only)** 0 = The receive FIFO buffer is not empty. 1 = The receive FIFO buffer is empty. **Note:** It's a mutual mirror bit of SPI_CTL[24].
[23:21]	**Reserved**	Reserved.
[20]	**SLVTOIF**	**Time-out Interrupt Flag** 0 = No receive FIFO time-out event. 1 = The receive FIFO buffer is not empty and it does not be read over 64 SPI clock periods in Master mode or over 576 SPI peripheral clock periods in Slave mode. When the received FIFO buffer is read by software, the time-out status will be cleared automatically. **Note:** This bit will be cleared by writing 1 to itself.
[19:17]	**Reserved**	Reserved.

[16]	**UNITIF**	**SPI Unit-transfer Interrupt Flag** 0 = The transfer does not finish yet. 1 = The SPI controller has finished one unit transfer. **Note 1:** This bit will be cleared by writing 1 to itself. **Note 2:** It's a mutual mirror bit of SPI_CTL[16].
[15:12]	**RXCNT**	**Receive FIFO Data Count (Read Only)** Indicates the valid data count of receive FIFO buffer.
[11]	**SLVSTIF**	**Slave Start Interrupt Status (Slave Only)** It is used to dedicate that the transfer has started in slave 3-wire mode. 0 = Slave does not detect any SPI bus clock transfer since the SLVSTIEN bit was set to 1. The transfer is not started. 1 = The transfer has started in Slave 3-wire mode. **Note 1:** It will be cleared as transfer done or by writing one to this bit. **Note 2**: It's a mutual mirror bit of SPI_SLVCTL[11].
[10:5]	**Reserved**	Reserved.
[4]	**TXTHIF**	**Transmit FIFO Threshold Interrupt Status (Read Only)** 0 = The valid data count within the transmit FIFO buffer is larger than the setting value of TXTH. 1 = The valid data count within the transmit FIFO buffer is less than or equal to the setting value of TXTH. **Note:** If TXTHIEN = 1 and TXTHIF = 1, the SPI controller will generate a SPI interrupt request.
[3]	**Reserved**	Reserved.
[2]	**RXOVIF**	**Receive FIFO Overrun Status** When the receive FIFO buffer is full, the follow-up data will be dropped and this bit will be set to 1. 0 = No overrun in receive FIFO. 1 = Overrun in receive FIFO. **Note:** This bit will be cleared by writing 1 to itself.
[1]	**Reserved**	Reserved.
[0]	**RXTHIF**	**Receive FIFO Threshold Interrupt Status (Read Only)** 0 = The valid data count within the receive FIFO buffer is less than or equal to the setting value of RXTH. 1 = The valid data count within the receive FIFO buffer is larger than the setting value of RXTH. **Note:** If RXTHIEN = 1 and RXTHIF = 1, the SPI controller will generate a SPI interrupt request.

6.13 Analog-to-Digital Converter (ADC)

6.13.1 Overview

The Mini58 series contains one 10-bit successive approximation analog-to-digital converters (SAR A/D converter) with eight input channels. The A/D converters can be started by software, external pin (STADC/P3.2) or PWM trigger.

6.13.2 Features

- Analog input voltage range: 0 ~ Analog Supply Voltage from AV_{DD}
- 10-bit resolution and 8-bit accuracy is guaranteed
- Up to eight single-end analog input channels
- Maximum ADC clock frequency is 6 MHz, and 14 ADC clocks per sample
- Two operating modes
  - Single mode: A/D conversion is performed one time on a specified channel
  - PWM sequence mode: When PWM trigger, two of three ADC channels from 0 to 2 will automatically convert analog data in the sequence of channel [0,1] or channel[1,2] or channel[0,2] defined by MODESEL (ADC_SEQCTL[3:2])
- An A/D conversion can be started by:
  - Software write 1 to SWTRG bit
  - External pin STADC
  - PWM trigger with optional start delay period
- Each Conversion result is held in data register with valid and overrun indicators
- Conversion results can be compared with specified value and user can select whether to generate an interrupt when conversion result matches the compare register setting
- Channel 7 supports 2 input sources: External analog voltage and internal fixed band-gap voltage

6.13.3 Block Diagram

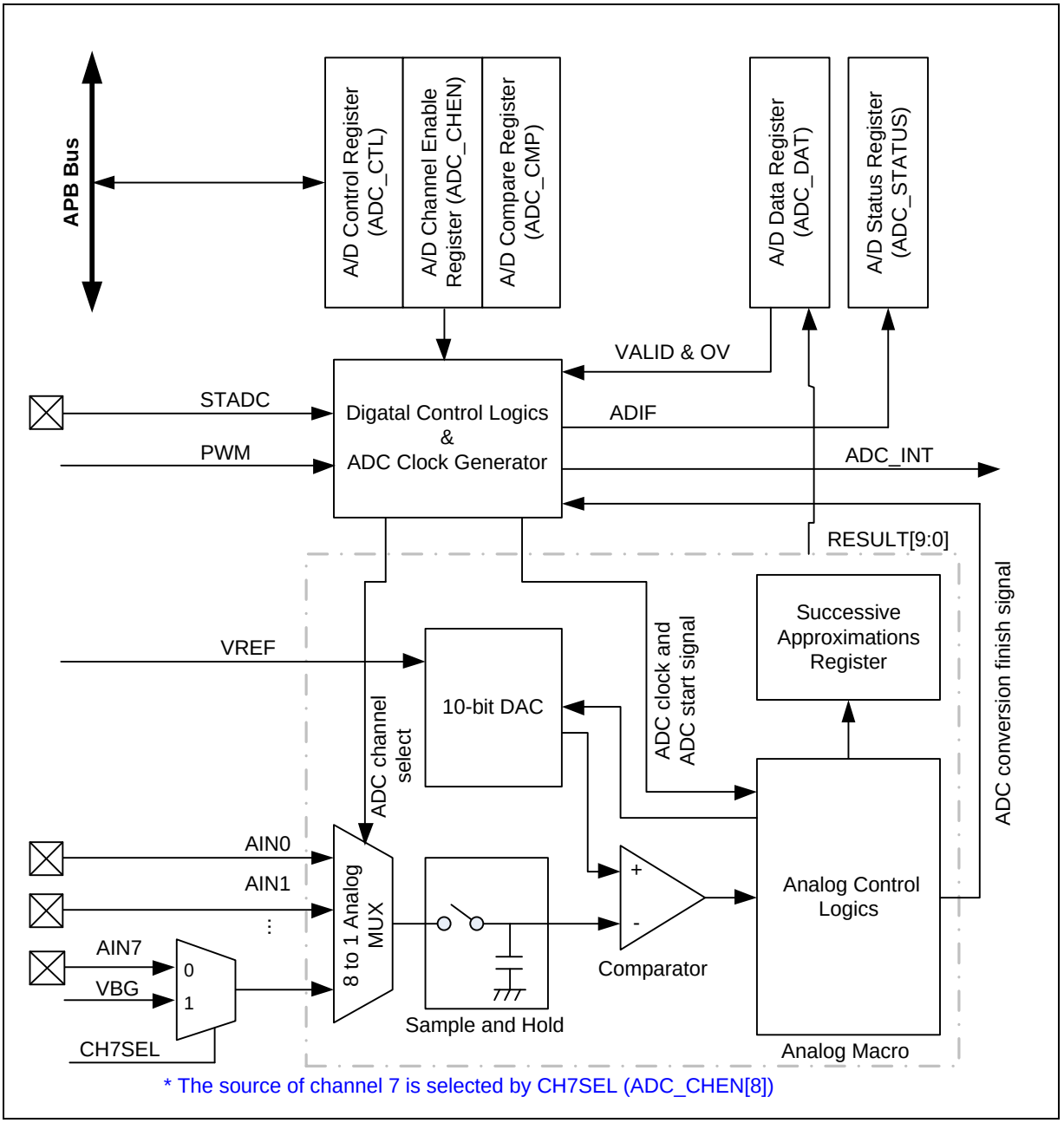


Figure 6.13-1 AD Controller Block Diagram

6.13.4 Basic Configuration

The ADC pin functions are configured in SYS_P1_MFP and SYS_P3_MFP register. It is recommended to disable the digital input path of the analog input pins to avoid the leakage current. User can disable the digital input path by configuring P1_DINOFF and P3_DINOFF register.

The ADC peripheral clock can be enabled in ADCCKEN (CLK_APBCLK[28]). The ADC peripheral clock source is selected by ADCSEL (CLK_CLKSEL1[3:2]). The clock prescalar is determined by

ADCDIN (CLK_CLKDIV[23:16]).

6.13.5 Functional Description

The A/D converter operates by successive approximation with 10-bit resolution. When changing the analog input channel is enabled, in order to prevent incorrect operation, software must clear SWTRG bit to 0 in the ADC_CTL register. The A/D converter discards the current conversion immediately and enters idle state while SWTRG bit is cleared.

6.13.5.1 ADC Peripheral Clock Generator

The ADC engine has four clock sources selected by ADCSEL (CLK_CLKSEL1[3:2]), and selected between HXT and LXT by XTLEN (CLK_PWRCTL[1:0]). The ADC clock peripheral frequency is divided by an 8-bit prescaler with the following formula:

ADC peripheral clock frequency = (ADC peripheral clock source frequency) / (ADCDIV+1);
where the 8-bit ADCDIV is located in register CLK_CLKDIV[23:16].

In general, software can set ADCSEL and ADCDIV to get 6 MHz or slightly less.

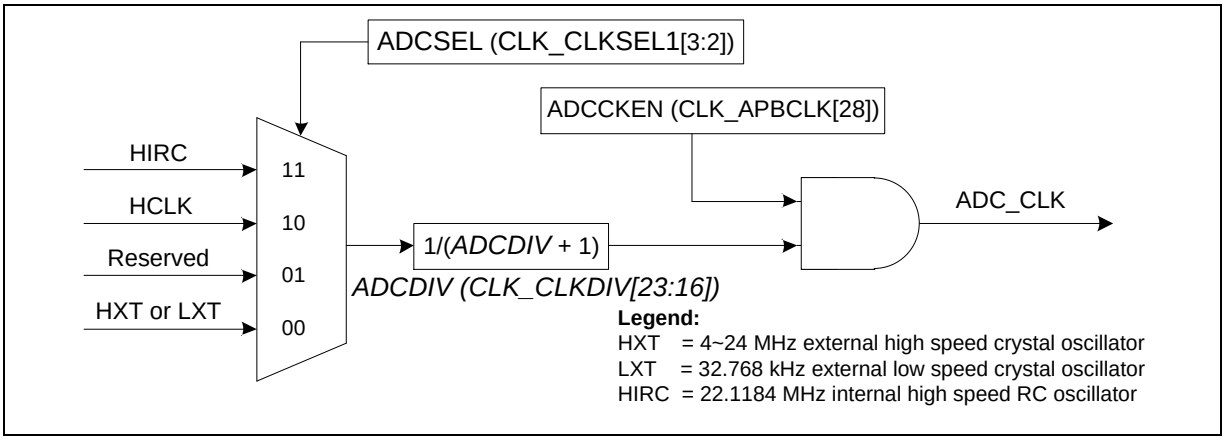


Figure 6.13-2 ADC Peripheral Clock Control

6.13.5.2 ADC Operation

A/D conversion is performed only once on the specified single channel. The operation is as follows:

1. A/D conversion will be started when the SWTRG bit of ADC_CTL is set to 1 by software or external trigger input.
2. When A/D conversion is finished, the result is stored in the A/D data register.
3. The ADIF bit of ADC_STATUS register will be set to 1. If the ADCIEN bit of ADC_CTL register is set to 1, the ADC interrupt will be asserted.
4. The SWTRG bit remains 1 during A/D conversion. When A/D conversion ends, the SWTRG bit is automatically cleared to 0 and the A/D converter enters idle state.
5. Figure 6.13-3 shows an example timing diagram for Single mode.

Note: If software enables more than one channel, the channel with the smallest number will be selected and the other enabled channels will be ignored.

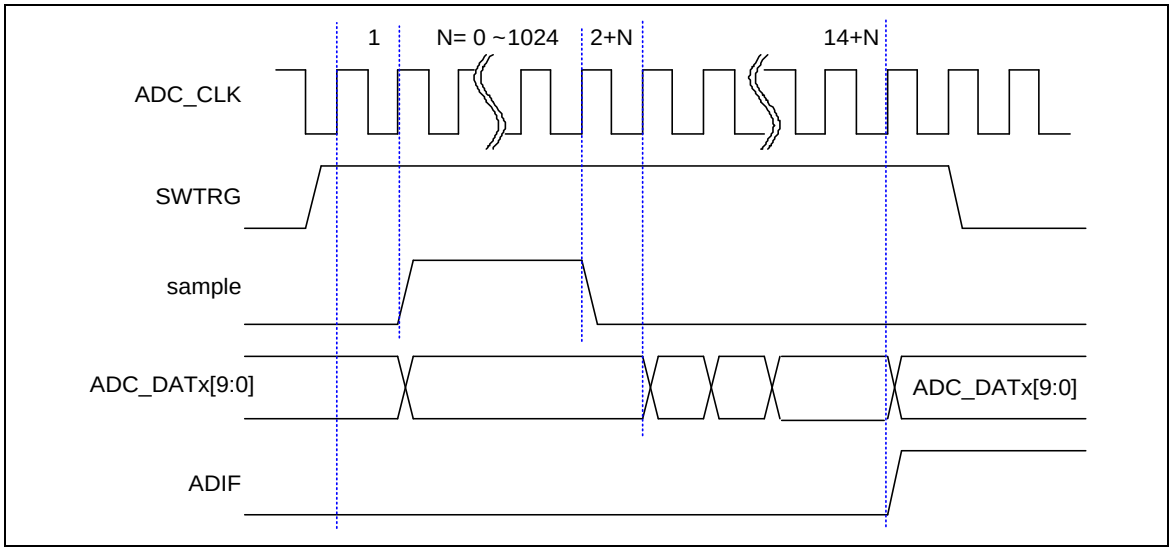


Figure 6.13-3 Single Mode Conversion Timing Diagram

6.13.5.3 External Trigger Input Sampling and A/D Conversion Time

A/D conversion can be triggered by external pin request. When the HWTRGEN (ADC_CTL[8]) bit is set to 1 to enable ADC external trigger function, setting the HWTRGSEL (ADC_CTL[5:4]) bits to 00b is to select external trigger input from the STADC pin. Software can set HWTRGCOND to select trigger condition between falling or rising edge. An 8-bit sampling counter is used to deglitch. 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.

6.13.5.4 Internal Reference Voltage

The band-gap voltage reference (V_{BG}) is an internal fixed reference voltage regardless of power supply variations. The V_{BG} output is internally connected to ADC channel 7 source multiplexer and Analog Comparators's (ACMP) negative input side.

For battery power detection application, user can use the V_{BG} as ADC input channel such that user can convert the A/D conversion result to calculate AV_{DD} with following formula.

$AV_{DD} = ((2 \text{ ^ } N) / R) * V_{BG}$

N: ADC resolution

R: A/D conversion result

V_{BG}: Band-gap voltage

The block diagram is shown as -4.

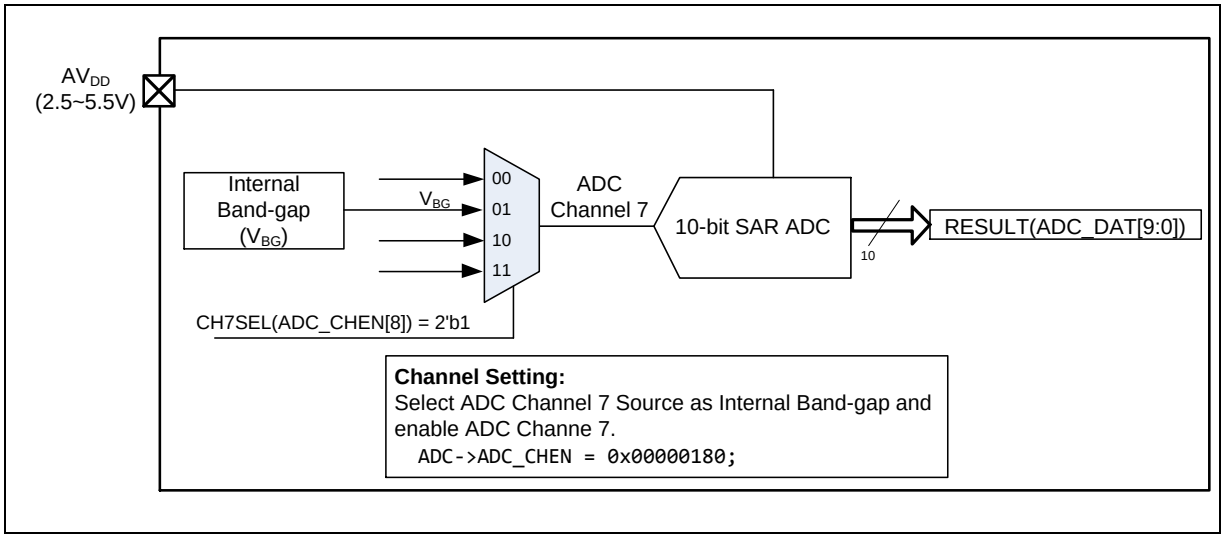


Figure 6.13-4 V_{BG} for Measuring AV_{DD} Application Block Diagram

For example, the V_{BG} typical value is 1.24 V, the ADC is 10-bit resolution, select V_{BG} as ADC channel 7 input source, and enable ADC channel 7. Then trigger ADC to converse.

If the A/D conversion result is 423:

N = 10

R = 423

V_{BG} = 1.24 V

AV_{DD} = ((2 ^ 10) / 423) * 1.24 = (1024 / 423) * 1.24 = 3 V

If the A/D conversion result is 512:

AV_{DD} = ((2 ^ 10) / 512) * 1.24 = (1024 / 512) * 1.24 = 2.48 V

6.13.5.5 PWM Trigger

A/D conversion can also be triggered by PWM request. When the HWTRGEN is set to high to enable ADC external hardware trigger function, setting the HWTRGSEL (ADC_CTL[5:4]) bits to 11b is to select external hardware trigger input source from PWM trigger. When PWM trigger is enabled, setting DELAY (ADC_TRGDLY[7:0]) bits can insert a delay time between PWM trigger condition and ADC start conversion.

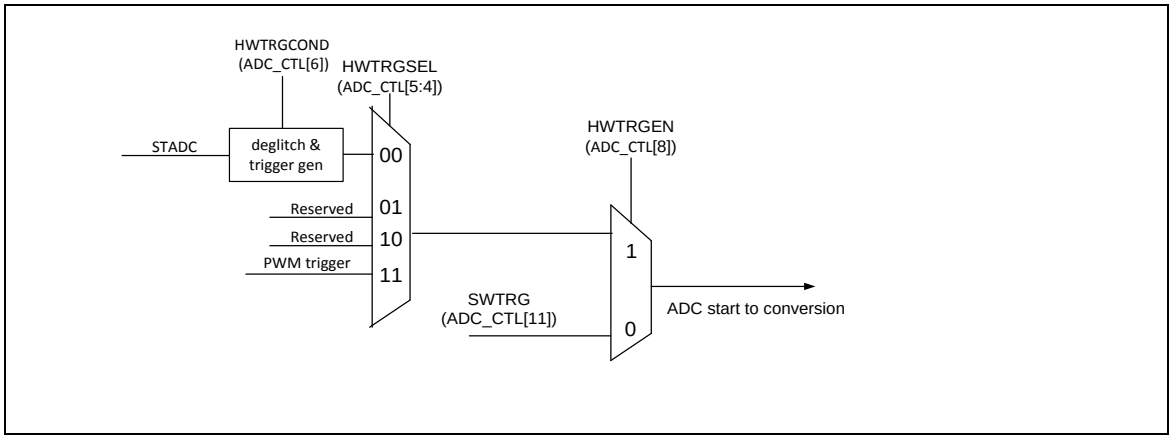


Figure 6.13-5 ADC Start Conversion Conditions

6.13.5.6 Conversion Result Monitor by Compare Mode Function

The Mini58 series ADC controller provides two compare registers, ADC_CMP0 and ADC_CMP1, to monitor maximum two specified channels. Software can select which channel to be monitored by setting CMPCH (ADC_CMPx[5:0]). CMPCOND 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 CMPDAT (ADC_CMPx[25:16]) ; 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 CMPDAT[9:0]. When the conversion of the channel specified by CMPCH 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 clear to 0. When the match counter reaches the setting of (CMPMCNT+1) then ADCMPF bit will be set to 1, if ADCMPIE bit is set then an ADC_INT 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. Figure 6.13-6 show detailed logic diagram.

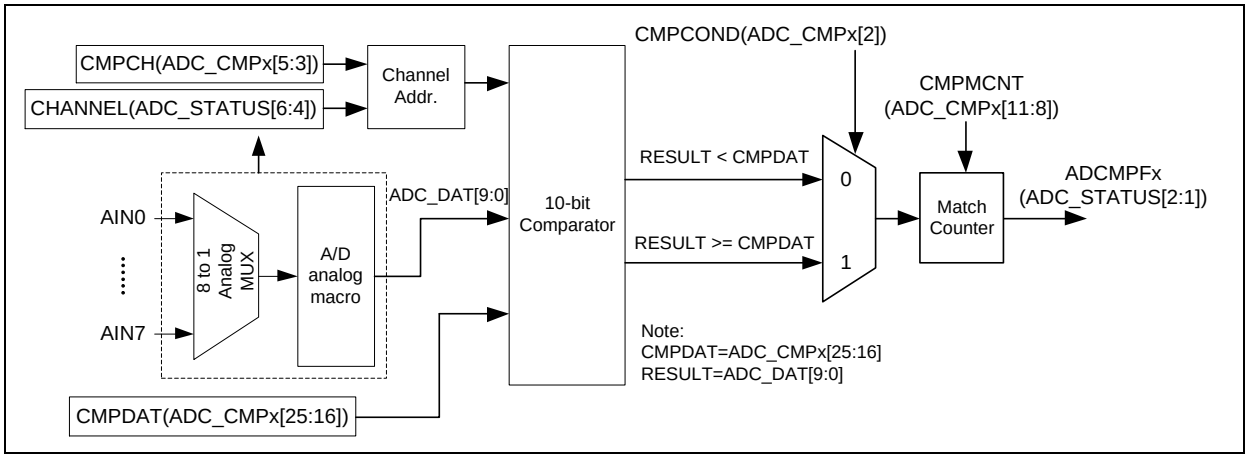


Figure 6.13-6 A/D Conversion Result Monitor Logics Diagram

6.13.5.7 Interrupt Sources

There are three interrupt sources of ADC interrupt. When an ADC operation mode finishes its conversion, the A/D conversion end flag, ADIF, will be set to 1. The ADCMPF0 and ADCMPF1 are the compare flags of compare function. When the conversion result meets the settings of ADC_CMP0/1, the corresponding flag will be set to 1. When one of the flags, ADIF, ADCMPF0

and ADCMPF1, is set to 1 and the corresponding interrupt enable bit, ADCIEN of ADC_CTL and ADCMPIE of ADC_CMP0/1, is set to 1, the ADC interrupt will be asserted. Software can clear these flags to revoke the interrupt request.

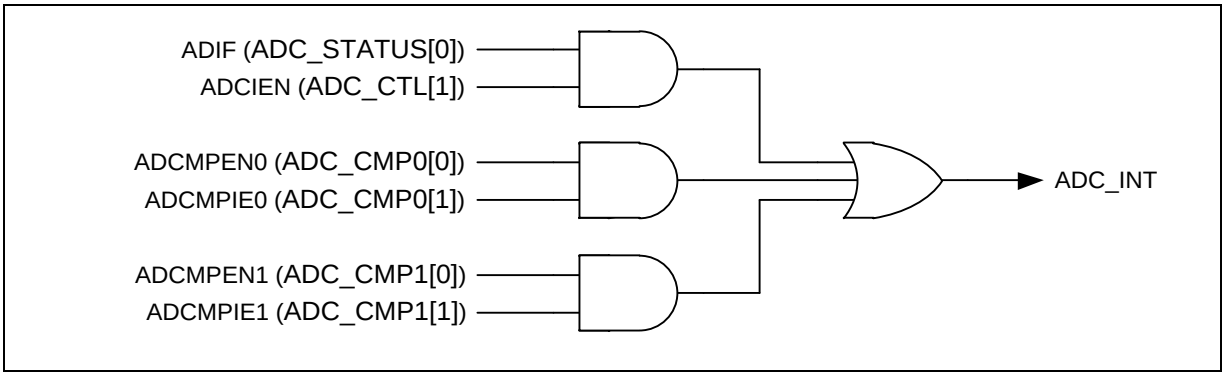


Figure 6.13-7 A/D Controller Interrupt

6.13.5.8 Conversion Result

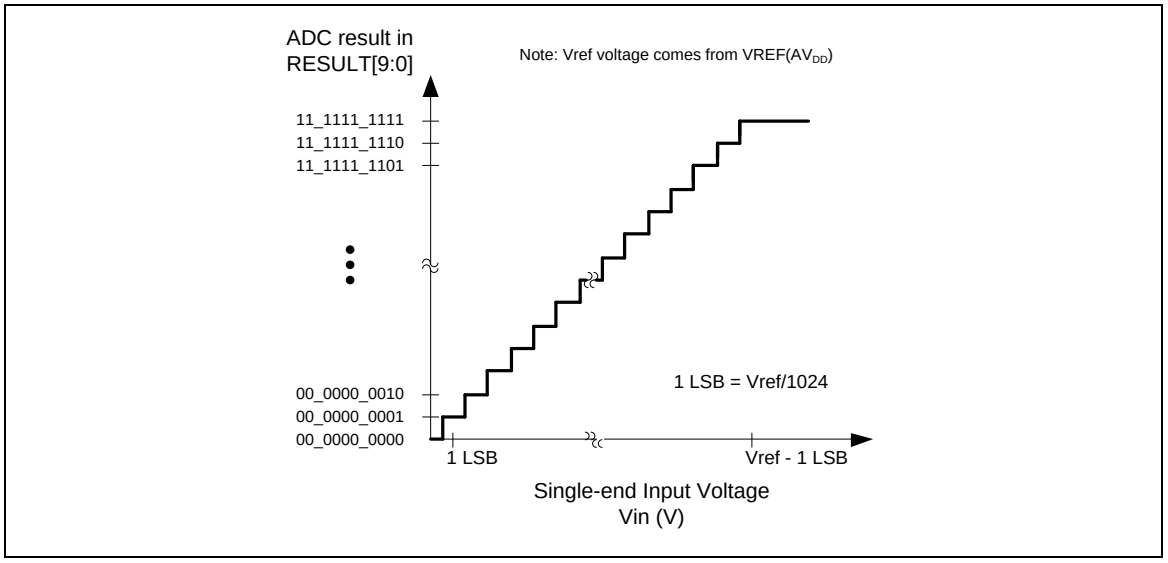


Figure 6.13-8 Conversion Result Mapping Diagram of ADC Single-end Input

6.13.5.9 PWM Sequential

Support sequential mode for 2 channels to reduce half interrupt frequency. When the SEQEN (ADC_SEQCTL[0]) is set to high to enable A/D PWM sequential function, setting the TRG1CTL (ADC_SEQCTL[11:8]) and TRG2CTL (ADC_SEQCTL[19:16]) are to select external trigger input from the PWM channel 0/2/4, type can be rising/center/falling/period, When ADC sequential mode is enabled, two of three ADC channels from 0 to 2 will automatically convert analog data in the sequence of channel [0, 1] or channel[1,2] or channel[0,2] defined by MODESEL (ADC_SEQCTL[3:2]). By the way, if SEQTYPE (ADC_SEQCTL[1]) is set to low, ADC delay time is only inserted before the first conversion. The second conversion starts immediately after the first conversion is completed. (for 2/3-shunt type), if SEQTYPE (ADC_SEQCTL[1]) is set to high, ADC delay time is inserted before each conversion. (for 1-shunt type), By the way, Valid ADC channel are CH0~2 in 2/3-shunt type, Valid ADC channel are CH0~7 in 1-shunt type, Figure 6.13-9 and Figure 6.13-10 show the function diagram.

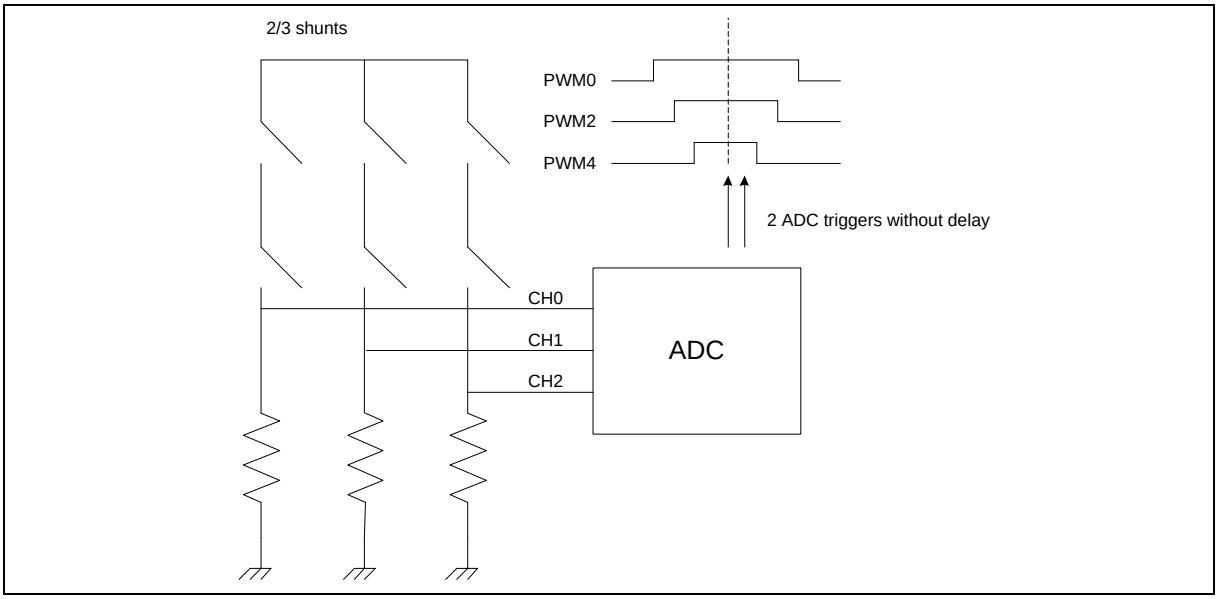


Figure 6.13-9 ADC Sequential Mode Type is 2/3-shunt

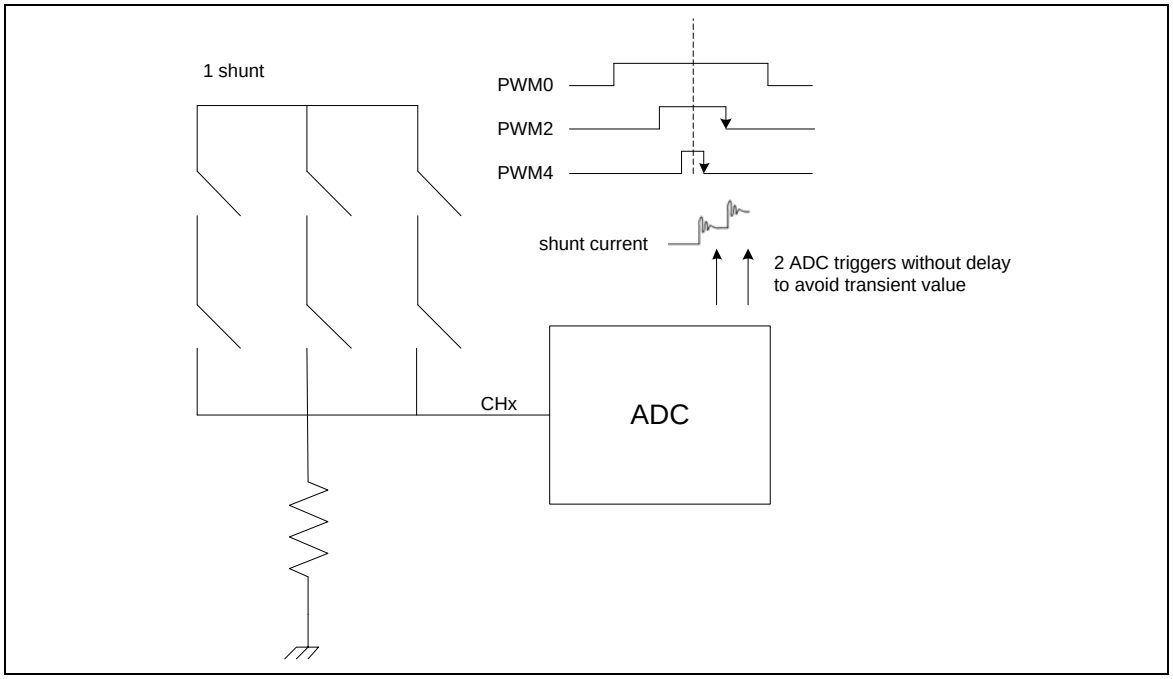


Figure 6.13-10 ADC Sequential Mode Type is 1-shunt

6.13.6 Register Map

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

Register	Offset	R/W	Description	Reset Value
ADC Base Address: **ADC_BA = 0x400E_0000**				
ADC_DAT	ADC_BA+0x00	R	A/D Data Register	0x0000_0000
ADC_CTL	ADC_BA+0x20	R/W	A/D Control Register	0x0000_0000
ADC_CHEN	ADC_BA+0x24	R/W	A/D Channel Enable Register	0x0000_0000
ADC_CMP0	ADC_BA+0x28	R/W	A/D Compare Register 0	0x0000_0000
ADC_CMP1	ADC_BA+0x2C	R/W	A/D Compare Register 1	0x0000_0000
ADC_STATUS	ADC_BA+0x30	R/W	A/D Status Register	0x0000_0000
ADC_TRGDLY	ADC_BA+0x44	R/W	A/D Trigger Delay Control Register	0x0000_0000
ADC_EXTSMPT	ADC_BA+0x48	R/W	A/D Sampling Time Counter Register	0x0000_0000
ADC_SEQCTL	ADC_BA+0x4C	R/W	A/D PWM Sequential Mode Control Register	0x0000_0000
ADC_SEQDAT1	ADC_BA+0x50	R	A/D PWM Sequential Mode First Result Register1	0x0000_0000
ADC_SEQDAT2	ADC_BA+0x54	R	A/D PWM Sequential Mode Second Result Register1	0x0000_0000

6.13.7 Register Description

ADC Data Register (ADC_DAT)

Register	Offset	R/W	Description	Reset Value
ADC_DAT	ADC_BA+0x00	R	A/D Data Register	0x0000_0000

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

Bits	Description	
[31:18]	**Reserved**	Reserved.
[17]	**VALID**	**Valid Flag** This bit is set to 1 when ADC conversion is completed and cleared by hardware after the ADC_DAT register is read. 0 = Data in RESULT[9:0] bits not valid. 1 = Data in RESULT[9:0] bits valid.
[16]	**OV**	**Over Run Flag** If converted data in RESULT[9:0] has not been read before the new conversion result is loaded to this register, OV is set to 1. It is cleared by hardware after the ADC_DAT register is read. 0 = Data in RESULT[9:0] is recent conversion result. 1 = Data in RESULT[9:0] overwrote.
[15:10]	**Reserved**	Reserved.
[9:0]	**RESULT**	**A/D Conversion Result** This field contains conversion result of ADC.

ADC Control Register (ADC_CTL)

Register	Offset	R/W	Description	Reset Value
ADC_CTL	ADC_BA+0x20	R/W	A/D 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				SWTRG	Reserved		HWTRGEN
7	6	5	4	3	2	1	0
Reserved	HWTRGCOND	HWTRGSEL		Reserved		ADCIEN	ADCEN

Bits	Description	
[31:12]	**Reserved**	Reserved.
[11]	**SWTRG**	**Software Trigger A/D Conversion Start** SWTRG bit can be set to 1 from two sources: software and external pin STADC. SWTRG will be cleared to 0 by hardware automatically after conversion complete. 0 = Conversion stopped and A/D converter entered idle state. 1 = Conversion start.
[10:9]	**Reserved**	Reserved.
[8]	**HWTRGEN**	**Hardware External Trigger Enable Bit** Enable or disable triggering of A/D conversion by external STADC pin. If external trigger is enabled, the SWTRG bit can be set to 1 by the selected hardware trigger source. 0= External trigger Disabled. 1= External trigger Enabled.
[7]	**Reserved**	Reserved.
[6]	**HWTRGCOND**	**Hardware External Trigger Condition** This bit decides whether the external pin STADC trigger event is falling or raising edge. The signal must be kept at stable state at least 4 PCLKs at high and low state for edge trigger. 0 = Falling edge. 1 = Raising edge.
[5:4]	**HWTRGSEL**	**Hardware Trigger Source Select Bit** 00 = A/D conversion is started by external STADC pin. 11 = A/D conversion is started by PWM trigger. Others = Reserved. **Note:** Software should disable TRGEN and SWTRG before change TRGS.
[3:2]	**Reserved**	Reserved.
[1]	**ADCIEN**	**A/D Interrupt Enable Bit** A/D conversion end interrupt request is generated if ADCIEN bit is set to 1.

Bits	Description	
		0 = A/D interrupt function Disabled. 1 = A/D interrupt function Enabled.
[0]	**ADCEN**	**A/D Converter Enable Bit** 0 = A/D Converter Disabled. 1 = A/D Converter Enabled. **Note:** 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.

ADC Channel Enable Register (ADC_CHEN)

Register	Offset	R/W	Description	Reset Value
ADC_CHEN	ADC_BA+0x24	R/W	A/D Channel 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							CH7SEL
7	6	5	4	3	2	1	0
CHEN7	CHEN6	CHEN5	CHEN4	CHEN3	CHEN2	CHEN1	CHEN0

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8]	**CH7SEL**	**Analog Input Channel 7 Selection** 0 = External analog input. 1 = Internal band-gap voltage (VBG). **Note:** When software selects the band-gap voltage as the analog input source of ADC channel 7, the ADC clock rate needs to be limited to lower than 300 kHz.
[7]	**CHEN7**	**Analog Input Channel 7 Enable Bit** 0 = Channel 7 Disabled. 1 = Channel 7 Enabled.
[6]	**CHEN6**	**Analog Input Channel 6 Enable Bit** 0 = Channel 6 Disabled. 1 = Channel 6 Enabled.
[5]	**CHEN5**	**Analog Input Channel 5 Enable Bit** 0 = Channel 5 Disabled. 1 = Channel 5 Enabled.
[4]	**CHEN4**	**Analog Input Channel 4 Enable Bit** 0 = Channel 4 Disabled. 1 = Channel 4 Enabled.
[3]	**CHEN3**	**Analog Input Channel 3 Enable Bit** 0 = Channel 3 Disabled. 1 = Channel 3 Enabled.
[2]	**CHEN2**	**Analog Input Channel 2 Enable Bit** 0 = Channel 2 Disabled. 1 = Channel 2 Enabled.
[1]	**CHEN1**	**Analog Input Channel 1 Enable Bit**

Bits	Description	
		0 = Channel 1 Disabled. 1 = Channel 1 Enabled.
[0]	**CHEN0**	**Analog Input Channel 0 Enable Bit** 0 = Channel 0 Disabled. 1 = Channel 0 Enabled. **Note:** If software enables more than one channel, the channel with the smallest number will be selected and the other enabled channels will be ignored.

A/D Compare Register 0/1 (ADC_CMP0/1)

Register	Offset	R/W	Description	Reset Value
ADC_CMP0	ADC_BA+0x28	R/W	A/D Compare Register 0	0x0000_0000
ADC_CMP1	ADC_BA+0x2C	R/W	A/D Compare Register 1	0x0000_0000

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

Bits	Description	
[31:26]	**Reserved**	Reserved.
[25:16]	**CMPDAT**	**Comparison Data** The 10-bit data is used to compare with conversion result of specified channel.
[15:12]	**Reserved**	Reserved.
[11:8]	**CMPMCNT**	**Compare Match Count** When the specified A/D channel analog conversion result matches the compare condition defined by CMPCOND[2], the internal match counter will increase 1. When the internal counter reaches the value to (CMPMCNT+1), the ADCMPFx bit will be set.
[7:6]	**Reserved**	Reserved.
[5:3]	**CMPCH**	**Compare Channel Selection** Set this field to select which channel's result to be compared. **Note:** Valid setting of this field is channel 0~7.
[2]	**CMPCOND**	**Compare Condition** 0 = Set the compare condition as that when a 10-bit A/D conversion result is less than the 10-bit CMPDAT (ADC_CMPx[25:16]), the internal match counter will increase one. 1 = Set the compare condition as that when a 10-bit A/D conversion result is greater or equal to the 10-bit CMPDAT (ADC_CMPx[25:16]), the internal match counter will increase one. **Note:** When the internal counter reaches the value to (CMPMCNT+1), the ADCMPFx bit will be set.
[1]	**ADCMPIE**	**A/D Compare Interrupt Enable Bit** If the compare function is enabled and the compare condition matches the setting of CMPCOND and CMPMCNT, ADCMPIE bit will be asserted, in the meanwhile, if ADCMPIE is set to 1, a compare interrupt request is generated. 0 = Compare function interrupt Disabled. 1 = Compare function interrupt Enabled.
[0]	**ADCMPEN**	**A/D Compare Enable Bit**

Bits	Description	
		Set 1 to this bit to enable comparing CMPDAT (ADC_CMPx[25:16]) with specified channel conversion results when converted data is loaded into the ADC_DAT register. 0 = Compare function Disabled. 1 = Compare function Enabled.

A/D Status Register (ADC_STATUS)

Register	Offset	R/W	Description	Reset Value
ADC_STATUS	ADC_BA+0x30	R/W	A/D Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							OV
15	14	13	12	11	10	9	8
Reserved							VALID
7	6	5	4	3	2	1	0
Reserved	CHANNEL			BUSY	ADCMPF1	ADCMPF0	ADIF

Bits	Description	
[31:17]	**Reserved**	Reserved.
[16]	**OV**	**Overrun Flag (Read Only)** It is a mirror to OV bit in ADC_DAT register.
[15:9]	**Reserved**	Reserved.
[8]	**VALID**	**Data Valid Flag (Read Only)** It is a mirror of VALID bit in ADC_DAT register.
[7]	**Reserved**	Reserved.
[6:4]	**CHANNEL**	**Current Conversion Channel (Read Only)** This filed reflects the current conversion channel when BUSY=1. When BUSY=0, it shows the number of the next converted channel.
[3]	**BUSY**	**BUSY/IDLE (Read Only)** This bit is mirror of as SWTRG bit in ADC_CTL 0 = A/D converter is in idle state. 1 = A/D converter is busy at conversion.
[2]	**ADCMPF1**	**A/D Compare Flag 1** When the selected channel A/D conversion result meets the setting condition in ADC_CMP1, this bit is set to 1. 0 = Conversion result in ADC_DAT does not meet the ADC_CMP1 setting. 1 = Conversion result in ADC_DAT meets the ADC_CMP1 setting. **Note:** This bit can be cleared to 0 by software writing 1.
[1]	**ADCMPF0**	**A/D Compare Flag 0** When the selected channel A/D conversion result meets the setting condition in ADC_CMP0, this bit is set to 1. 0 = Conversion result in ADC_DAT does not meet the ADC_CMP0 setting. 1 = Conversion result in ADC_DAT meets the ADC_CMP0 setting. **Note:** This bit can be cleared to 0 by software writing 1.

Bits	Description	
[0]	**ADIF**	**A/D Conversion End Flag** A status flag that indicates the end of A/D conversion. ADIF is set to 1 When A/D conversion ends. **Note:** This bit can be cleared to 0 by software writing 1.

A/D Trigger Delay Controller Register (ADC_TRGDLY)

Register	Offset	R/W	Description	Reset Value
ADC_TRGDLY	ADC_BA+0x44	R/W	A/D 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**
DELAY							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**DELAY**	**PWM Trigger Delay Timer** Set this field will delay ADC start conversion time after PWM trigger. PWM trigger delay time is (4 * DELAY) * system clock.

A/D Sampling Register (ADC_EXTSMPT)

Register	Offset	R/W	Description	Reset Value
ADC_EXTSMPT	ADC_BA+0x48	R/W	A/D Sampling Time 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				EXTSMPT			

Bits	Description	
[31:4]	Reserved	Reserved.
[3:0]	EXTSMPT	**Additional ADC Sample Clock** If the ADC input is unstable, user can set this register to increase the sampling time to get a stable ADC input signal. The default sampling time is 1 ADC clocks. The additional clock number will be inserted to lengthen the sampling clock. 0 = Number of additional clock cycles is 0. 1 = Number of additional clock cycles is 1. 2 = Number of additional clock cycles is 2. 3 = Number of additional clock cycles is 4. 4 = Number of additional clock cycles is 8. 5 = Number of additional clock cycles is 16. 6 = Number of additional clock cycles is 32. 7 = Number of additional clock cycles is 64. 8 = Number of additional clock cycles is 128. 9 = Number of additional clock cycles is 256. 10 = Number of additional clock cycles is 512. 11 = Number of additional clock cycles is 1024. 12 = Number of additional clock cycles is 1024. 13 = Number of additional clock cycles is 1024. 14 = Number of additional clock cycles is 1024. 15 = Number of additional clock cycles is 1024.

A/D PWM Sequential Register (ADC_SEQCTL)

Register	Offset	R/W	Description	Reset Value
ADC_SEQCTL	ADC_BA+0x4C	R/W	A/D PWM Sequential Mode Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved				TRG2CTL			
15	14	13	12	11	10	9	8
Reserved				TRG1CTL			
7	6	5	4	3	2	1	0
Reserved				MODESEL		SEQTYPE	SEQEN

Bits	Description	
[31:20]	**Reserved**	Reserved.
[19:16]	**TRG2CTL**	**PWM Trigger Source Selection For TRG2CTL[3:2]** 00 = PWM Trigger source is PWM0. 01 = PWM Trigger source is PWM2. 10 = PWM Trigger source is PWM4. 11 = PWM Trigger source is reserved. **PWM Trigger Type Selection for TRG2CTL[1:0]** 00 = Rising of the selected PWM. 01 = Center of the selected PWM. 10 = Falling of the selected PWM. 11 = Period of the selected PWM. **Note:** PWM trigger source is valid for 1-shunt type.
[15:12]	**Reserved**	Reserved.
[11:8]	**TRG1CTL**	**PWM Trigger Source Selection For TRG1CTL[3:2]** 00 = PWM Trigger source is PWM0. 01 = PWM Trigger source is PWM2. 10 = PWM Trigger source is PWM4. 11 = PWM Trigger source is reserved. **PWM Trigger Type Selection for TRG1CTL[1:0]** 00 = Rising of the selected PWM. 01 = Center of the selected PWM. 10 = Falling of the selected PWM. 11 = Period of the selected PWM. **Note:** PWM trigger source is valid for 1-shunt and 2/3-shunt type.
[7:4]	**Reserved**	Reserved.
[3:2]	**MODESEL**	**ADC Sequential Mode Selection**

Bits	Description	
		00 = Issue ADC_INT after Channel 0 then Channel 1 conversion finishes when SEQEN =1. 01 = Issue ADC_INT after Channel 1 then Channel 2 conversion finishes when SEQEN =1. 10 = Issue ADC_INT after Channel 0 then Channel 2 conversion finishes when SEQEN =1. 11 = Reserved.
[1]	**SEQTYPE**	**ADC Sequential Mode Type** 0 = ADC delay time is only inserted before the first conversion. The second conversion starts immediately after the first conversion is completed. (for 2/3-shunt type) 1 = ADC delay time is inserted before each conversion. (for 1-shunt type)
[0]	**SEQEN**	**ADC Sequential Mode Enable Bit** When ADC sequential mode is enabled, two of three ADC channels from 0 to 2 will automatically convert analog data in the sequence of channel [0, 1] or channel[1, 2] or channel[0, 2] defined by MODESEL (ADC_SEQCTL[3:2]). 0 = ADC sequential mode Disabled. 1 = ADC sequential mode Enabled.

A/D PWM Sequential Mode Result Register (ADC_SEQDAT1/2)

Register	Offset	R/W	Description	Reset Value
ADC_SEQDAT1	ADC_BA+0x50	R	A/D PWM Sequential Mode First Result Register1	0x0000_0000
ADC_SEQDAT2	ADC_BA+0x54	R	A/D PWM Sequential Mode Second Result Register1	0x0000_0000

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

Bits	Description	
[31:18]	**Reserved**	Reserved.
[17]	**VALID**	**Valid Flag** This bit is set to 1 when ADC conversion is completed and cleared by hardware after the ADC_DAT register is read. 0 = Data in RESULT[9:0] bits not valid. 1 = Data in RESULT[9:0] bits valid.
[16]	**OV**	**Over Run Flag** If converted data in RESULT[9:0] has not been read before the new conversion result is loaded to this register, OV is set to 1. It is cleared by hardware after the ADC_DAT register is read. 0 = Data in RESULT[9:0] is recent conversion result. 1 = Data in RESULT[9:0] overwritten.
[15:10]	**Reserved**	Reserved.
[9:0]	**RESULT**	**A/D PWM Sequential Mode Conversion Result** This field contains conversion result of ADC.

6.14 Analog Comparator (ACMP)

6.14.1 Overview

The NuMicro® Mini58 series contains two comparators which can be used in a number of different configurations. 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 interrupt when the comparator output value changes.

6.14.2 Features

- Analog input voltage range: 0 ~ AV_{DD}
- Supports Hysteresis function
- Optional internal reference voltage source for each comparator negative input
- ACMP0 supports:
  - Four positive sources
    - P1.5, P1.0, P1.2, or P1.3
  - Three negative sources
    - P1.4
    - Internal Comparator Reference Voltage (CRV)
    - Internal band-gap voltage (V_{BG})
- ACMP1 supports:
  - Four positive sources
    - P3.1, P3.2, P3.4, or P3.5
  - Three negative sources
    - P3.0
    - Internal Comparator Reference Voltage (CRV)
    - Internal band-gap voltage (V_{BG})

6.14.3 Block Diagram

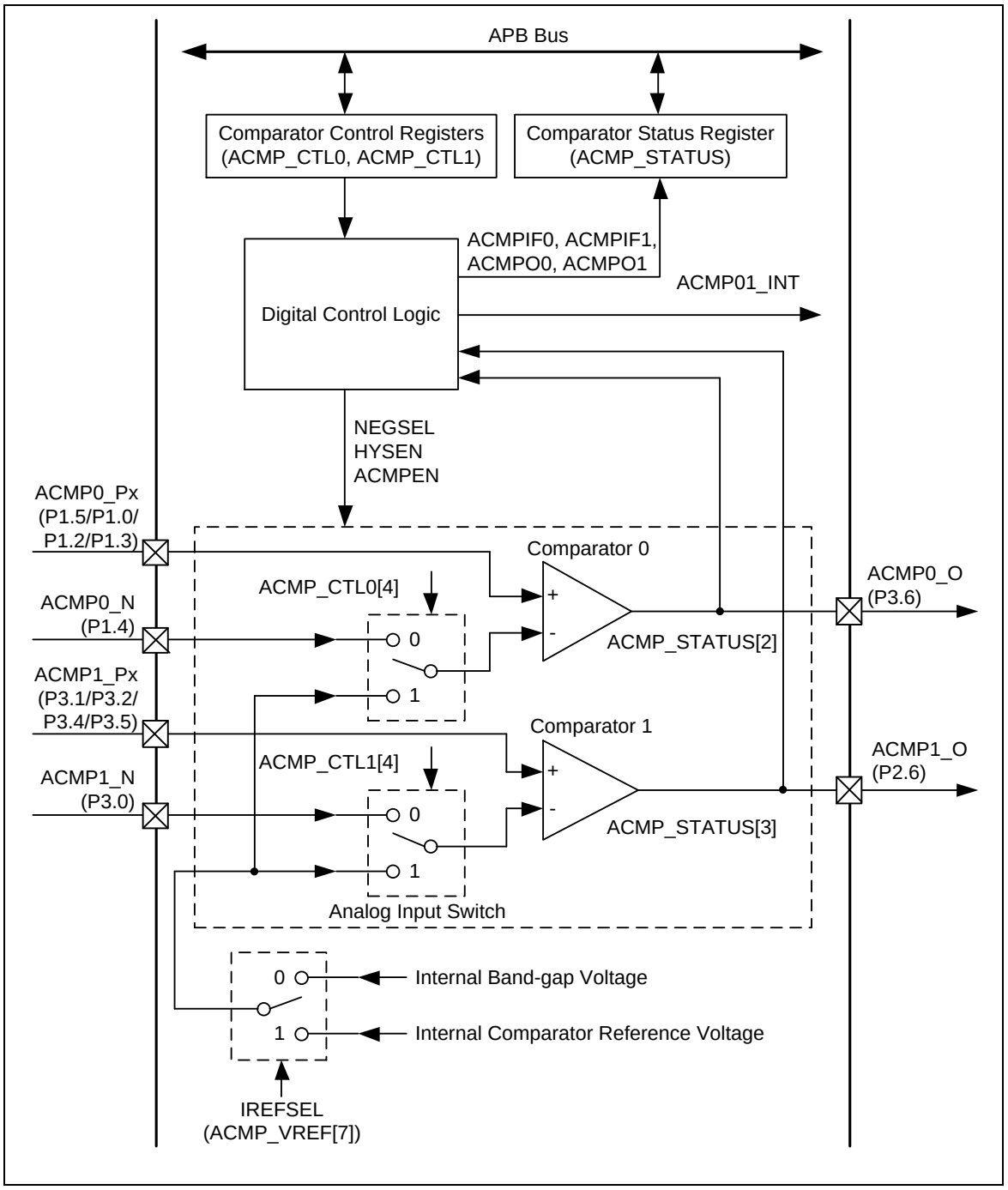


Figure 6.14-1 Analog Comparator Block Diagram

6.14.4 Basic Configuration

The ACMP pin functions are configured in SYS_P1_MFP, SYS_P2_MFP and SYS_P3_MFP

registers. It is recommended to disable the digital input path of the analog input pins to avoid the leakage current. The digital input path can be disabled by configuring the P1_OFFD and P3_OFFD registers. If a GPIO pin is configured as an ACMP input pin, this pin should not be set as Push-pull Output mode in the Px_PMD register. Input mode is the safest configuration. If Open-drain, Output mode or Quasi-bidirectional mode is selected, do not output 0 on this GPIO pin. The default GPIO output value is 1. The default Px_PMD setting is determined by user configuration. It could be configured as Input mode or Quasi-bidirectional mode in user configuration.

The ACMP peripheral clocks can be enabled by setting ACMPCKEN (CLK_APBCLK[30]) to 1.

6.14.5 Functional Description

6.14.5.1 Interrupt Sources

The output of comparators are sampled by PCLK and reflected at ACMPOx (ACMP_STATUS[3] and ACMP_STATUS[2]). If ACMPIE (ACMP_CRx[1]) is set to 1, the comparator interrupt will be enabled. As the output state of comparator is changed, the comparator interrupt will be asserted and the corresponding flag, ACMPFx (ACMP_STATUS[1] and ACMP_STATUS[0]), will be set. This flag can be cleared by software writing 1.

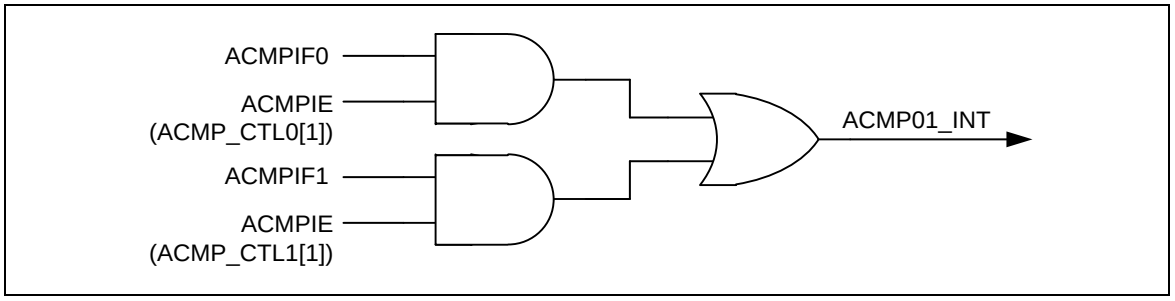


Figure 6.14-2 Analog Comparator Controller Interrupt Sources

6.14.5.2 Hysteresis Function

The analog comparator provides hysteresis function to make the comparator output transition more stable. If comparator output is 0, it will not change to 1 until the positive input voltage exceeds the negative input voltage by a positive hysteresis voltage. Similarly, if comparator output is 1, it will not change to 0 until the positive input voltage drops below the negative input voltage by a negative hysteresis voltage.

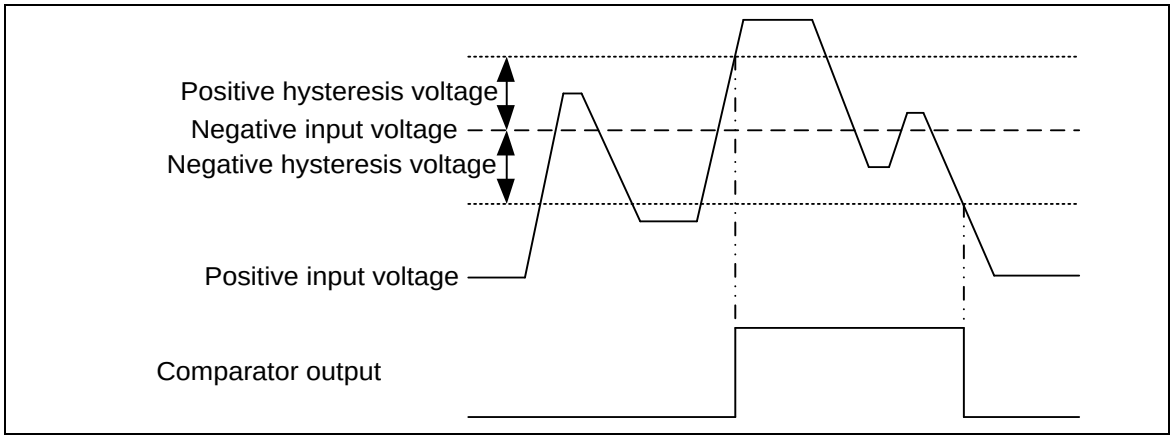


Figure 6.14-3 Comparator Hysteresis Function

6.14.5.3 Filter Function

The analog comparator provides filter function to avoid the unstable state of comparator output. By setting FILTSEL (ACMP_CTL0[23:20], ACMP_CTL1[23:20]), 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. The filter function is very useful to get noise free output. User selects the filter amount is just larger than noise time by setting FILTSEL (ACMP_CTL0[23:20], ACMP_CTL1[23:20]). For example, if the noise time is shorter than 8 PCLK period, user can select FILTSEL (ACMP_CTL0[23:20], ACMP_CTL1[23:20]) as 0100 to get noise free output for ACMP. Thus, the noise time that is shorter than 8 PCLK is ignored. If the noise increase to 13 PCLK period, user can select FILTSEL (ACMP_CTL0[23:20], ACMP_CTL1[23:20]) as 0101 to get noise free output for ACMP.

6.14.6 Comparator Reference Voltage (CRV)

6.14.6.1 Introduction

The comparator reference voltage (CRV) module is responsible for generating reference voltage for comparators. The CRV module consists of resisters ladder and analog switch, and user can set the CRV output voltage using CRVCTL (ACMP_VREF[3:0]) and select the reference voltage to ACMP by setting IREFSEL (ACMP_VREF[7]).

6.14.6.2 Features

- User selectable references voltage by setting CRVCTL (ACMP_VREF[3:0])
- Automatic disable resisters ladder for reducing power consumption when setting IREFSEL (ACMP_VREF[7]) = 0 (selecting Band-gap source voltage)

The block diagram of the CRV module is shown below:

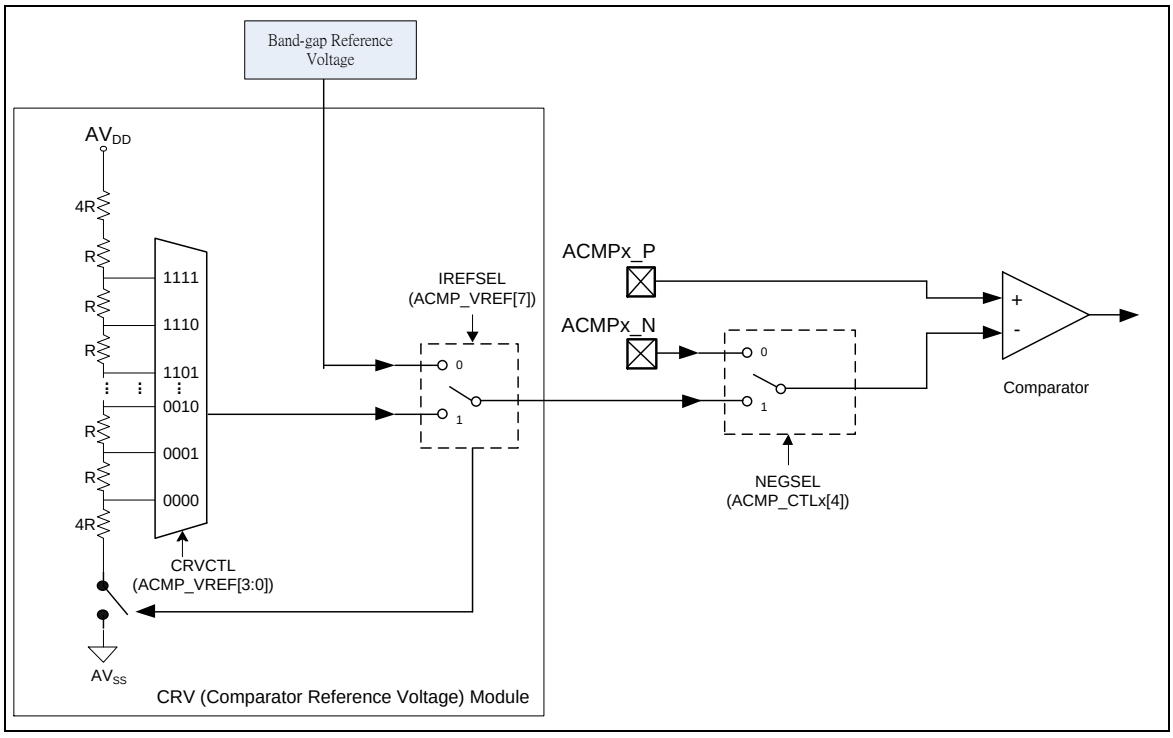


Figure 6.14-4 Comparator Reference Voltage Block Diagram

6.14.7 Register Map

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

Register	Offset	R/W	Description	Reset Value
ACMP Base Address: **ACMP_BA = 0x400D_0000**				
ACMP_CTL0	ACMP_BA+0x00	R/W	Analog Comparator 0 Control Register	0x0000_0000
ACMP_CTL1	ACMP_BA+0x04	R/W	Analog Comparator 1 Control Register	0x0000_0000
ACMP_STATUS	ACMP_BA+0x08	R/W	Analog Comparator 0/1 Status Register	0x0000_0000
ACMP_VREF	ACMP_BA+0x0C	R/W	Analog Comparator Reference Voltage Control Register	0x0000_0000

6.14.8 Register Description

Analog Comparator 0 Control Register (ACMP_CTL0)

Register	Offset	R/W	Description	Reset Value
ACMP_CTL0	ACMP_BA+0x00	R/W	Analog Comparator 0 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	POSSEL		Reserved				
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
FILTSEL				Reserved			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						FTRGEN	RTRGEN
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			NEGSEL	Reserved	HYSSEL	ACMPIE	ACMPEN

Bits	Description	
[31]	**Reserved**	Reserved.
[30:29]	**POSSEL**	**Analog Comparator 0 Positive Input Selection** 00 = ACMP0_Px is from ACMP0_P0 (P1.5) pin. 01 = ACMP0_Px is from ACMP0_P1 (P1.0) pin. 10 = ACMP0_Px is from ACMP0_P2 (P1.2) pin. 11 = ACMP0_Px is from ACMP0_P3 (P1.3) pin.
[28:24]	**Reserved**	Reserved.
[23:20]	**FILTSEL**	**Comparator Output Filter Count Selection** 0000 = Filter function is Disabled. 0001 = ACMP0 output is sampled 1 consecutive PCLK. 0010 = ACMP0 output is sampled 2 consecutive PCLKs. 0011 = ACMP0 output is sampled 4 consecutive PCLKs. 0100 = ACMP0 output is sampled 8 consecutive PCLKs. 0101 = ACMP0 output is sampled 16 consecutive PCLKs. 0110 = ACMP0 output is sampled 32 consecutive PCLKs. 0111 = ACMP0 output is sampled 64 consecutive PCLKs. 1000 = ACMP0 output is sampled 128 consecutive PCLKs. 1001 = ACMP0 output is sampled 256 consecutive PCLKs. 1010 = ACMP0 output is sampled 512 consecutive PCLKs. Others = Reserved.
[19:10]	**Reserved**	Reserved.
[9]	**FTRGEN**	**Analog Comparator 0 Falling Edge Trigger Enable** 0 = Analog comparator 0 falling edge trigger PWM or Timer enabled. 1 = Analog comparator 0 falling edge trigger disabled. **Note:** The bit is only effective while analog comparator 0 triggers PWM or Timer.

Bits	Description	
[8]	**RTRGEN**	**Analog Comparator 0 Rising Edge Trigger Enable Bit** 0 = Analog comparator 0 rising edge trigger PWM or Timer enabled. 1 = Analog comparator 0 rising edge trigger disabled. **Note:** The bit is only effective while analog comparator 0 triggers PWM or Timer.
[7:5]	**Reserved**	Reserved.
[4]	**NEGSEL**	**Analog Comparator 0 Negative Input Select Bit** 0 = The source of the negative comparator input is from CPN0 pin. 1 = The source of the negative comparator input is from internal band-gap voltage or comparator reference voltage.
[3]	**Reserved**	Reserved.
[2]	**HYSSEL**	**Analog Comparator 0 Hysteresis Select Bit** 0 = Hysteresis function Disabled. 1 = Hysteresis function Enabled.
[1]	**ACMPIE**	**Analog Comparator 0 Interrupt Enable Bit** 0 = Interrupt function Disabled. 1 = Interrupt function Enabled.
[0]	**ACMPEN**	**Analog Comparator 0 Enable Bit** 0 = Analog Comparator 0 Disabled. 1 = Analog Comparator 1 Enabled. **Note:** Analog comparator output needs to wait 2 us stable time after this bit is set.

Analog Comparator 1 Control Register (ACMP_CTL1)

Register	Offset	R/W	Description	Reset Value
ACMP_CTL1	ACMP_BA+0x04	R/W	Analog Comparator 1 Control Register	0x0000_0000

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

Bits	Description	
[31]	**Reserved**	Reserved.
[30:29]	**POSSEL**	**Analog Comparator 1 Positive Input Selection** 00 = ACMP1_Px is from ACMP1_P0 (P3.1) pin. 01 = ACMP1_Px is from ACMP1_P1 (P3.2) pin. 10 = ACMP1_Px is from ACMP1_P2 (P3.4) pin. 11 = ACMP1_Px is from ACMP1_P3 (P3.5) pin.
[28:24]	**Reserved**	Reserved.
[23:20]	**FILTSEL**	**Comparator Output Filter Count Selection** 0000 = Filter function is Disabled. 0001 = ACMP1 output is sampled 1 consecutive PCLK. 0010 = ACMP1 output is sampled 2 consecutive PCLKs. 0011 = ACMP1 output is sampled 4 consecutive PCLKs. 0100 = ACMP1 output is sampled 8 consecutive PCLKs. 0101 = ACMP1 output is sampled 16 consecutive PCLKs. 0110 = ACMP1 output is sampled 32 consecutive PCLKs. 0111 = ACMP1 output is sampled 64 consecutive PCLKs. 1000 = ACMP1 output is sampled 128 consecutive PCLKs. 1001 = ACMP1 output is sampled 256 consecutive PCLKs. 1010 = ACMP1 output is sampled 512 consecutive PCLKs. Others = Reserved.
[19:10]	**Reserved**	Reserved.
[9]	**FTRGEN**	**Analog Comparator 1 Falling Edge Trigger Enable Bit** 0 = Analog comparator 1 falling edge trigger PWM or Timer Enabled. 1 = Analog comparator 1 falling edge trigger Disabled. **Note:** The bit is only effective while analog comparator 1 triggers PWM or Timer.
[8]	**RTRGEN**	**Analog Comparator 1 Rising Edge Trigger Enable Bit**

Bits	Description	
		0 = Analog comparator 1 rising edge trigger PWM or Timer Enabled. 1 = Analog comparator 1 rising edge trigger Disabled. **Note:** The bit is only effective while analog comparator 1 triggers PWM or Timer.
[7:5]	**Reserved**	Reserved.
[4]	**NEGSEL**	**Analog Comparator 1 Negative Input Select Bit** 0 = The source of the negative comparator input is from CPN1 pin. 1 = The source of the negative comparator input is from internal band-gap voltage or comparator reference voltage.
[3]	**Reserved**	Reserved.
[2]	**HYSSEL**	**Analog Comparator 1 Hysteresis Select Bit** 0 = Hysteresis function Disabled. 1 = Hysteresis function Enabled.
[1]	**ACMPIE**	**Analog Comparator 1 Interrupt Enable Bit** 0 = Interrupt function Disabled. 1 = Interrupt function Enabled.
[0]	**ACMPEN**	**Analog Comparator 1 Enable Bit** 0 = Analog Comparator 1 Disabled. 1 = Analog Comparator 1 Enabled. **Note:** Analog comparator output needs to wait 2 us stable time after this bit is set.

Analog Comparator 0/1 Status Register (ACMP_STATUS)

Register	Offset	R/W	Description	Reset Value
ACMP_STATUS	ACMP_BA+0x08	R/W	Analog Comparator 0/1 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				ACMPO1	ACMPO0	ACMPIF1	ACMPIF0

Bits	Description	
[31:4]	**Reserved**	Reserved.
[3]	**ACMPO1**	**Analog Comparator 1 Output** Synchronized to the APB clock to allow reading by software. Cleared when the comparator 1 is disabled ACMPEN (ACMP_CTL1[0]) = 0. 0 = Analog comparator 1 outputs 0. 1 = Analog comparator 1 outputs 1.
[2]	**ACMPO0**	**Analog Comparator 0 Output** Synchronized to the APB clock to allow reading by software. Cleared when the comparator 0 is disabled ACMPEN (ACMP_CTL0[0]) = 0. 0 = Analog comparator 0 outputs 0. 1 = Analog comparator 0 outputs 1.
[1]	**ACMPIF1**	**Analog Comparator 1 Flag** This bit is set by hardware whenever the comparator 1 output changes state. This will generate an interrupt if ACMPIE (ACMP_CTL1[1]) = 1. 0 = Analog comparator 1 output does not change. 1 = Analog comparator 1 output changed. **Note:** This bit can be cleared to 0 by software writing 1.
[0]	**ACMPIF0**	**Analog Comparator 0 Flag** This bit is set by hardware whenever the comparator 0 output changes state. This will generate an interrupt if ACMPIE (ACMP_CTL0[1]) = 1. 0 = Analog comparator 0 output does not change. 1 = Analog comparator 0 output changed. **Note:** This bit can be cleared to 0 by software writing 1.

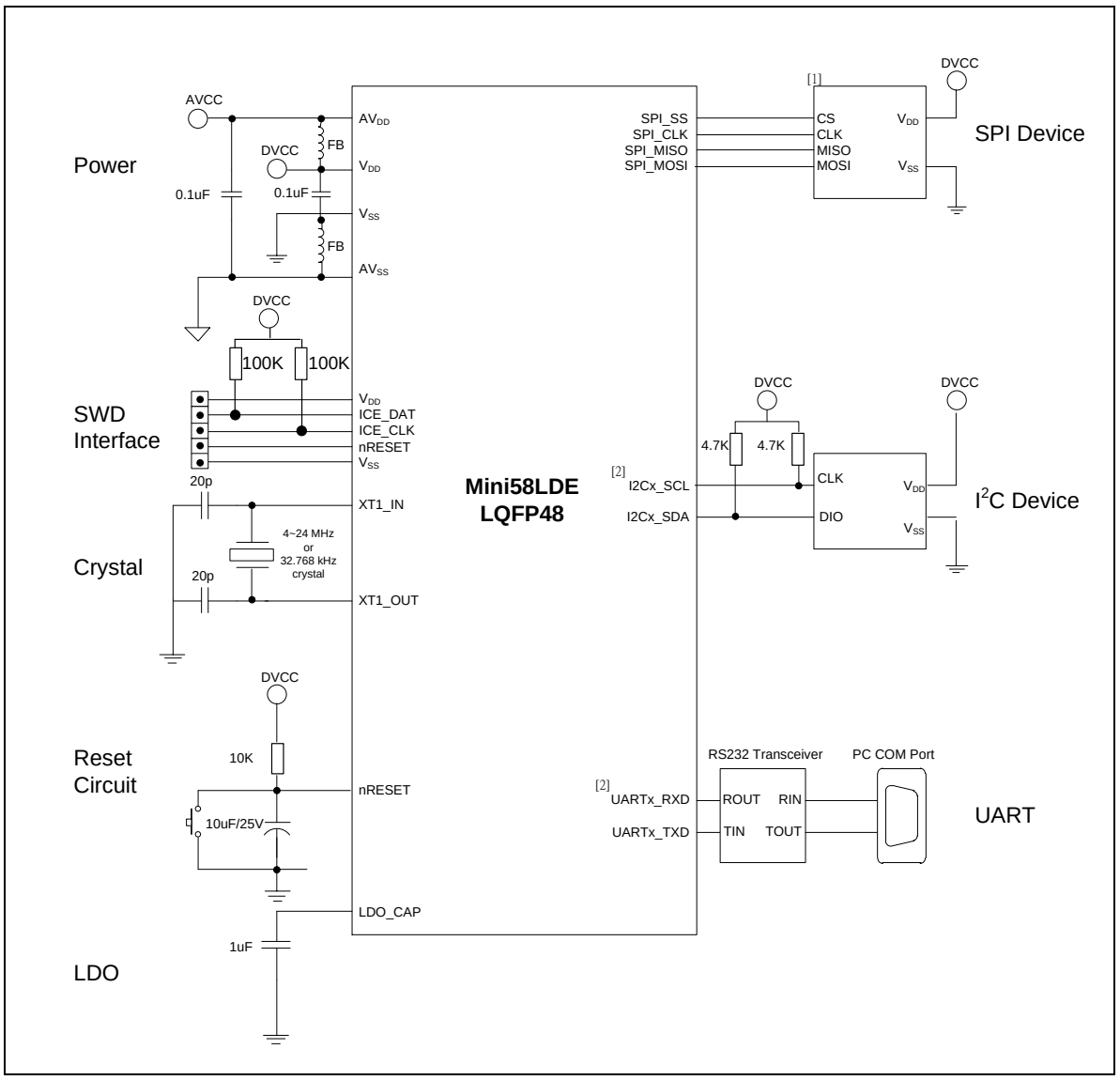
Analog Comparator Reference Voltage Control Register (ACMP_VREF)

Register	Offset	R/W	Description	Reset Value
ACMP_VREF	ACMP_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
IREFSEL	Reserved			CRVCTL			

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**IREFSEL**	**CRV Internal Reference Selection** 0 = Band-gap voltage. 1 = Internal comparator reference voltage.
[6:4]	**Reserved**	Reserved.
[3:0]	**CRVCTL**	**Comparator Reference Voltage Control** Comparator reference voltage = AV_{DD} * (1 / 6 + CRVCTL(ACMP_VREF[3:0]) / 24).

7 APPLICATION CIRCUIT



Note 1: For the SPI device, the Mini58 chip supply voltage must be equal to SPI device working voltage. For example, when the SPI Flash working voltage is 3.3 V, the Mini58 chip supply voltage must also be 3.3V.

Note 2: x denotes 0 or 1.

Note 3: It is recommended to use 100 kΩ pull-up resistor on both ICE_DAT and ICE_CLK pin.

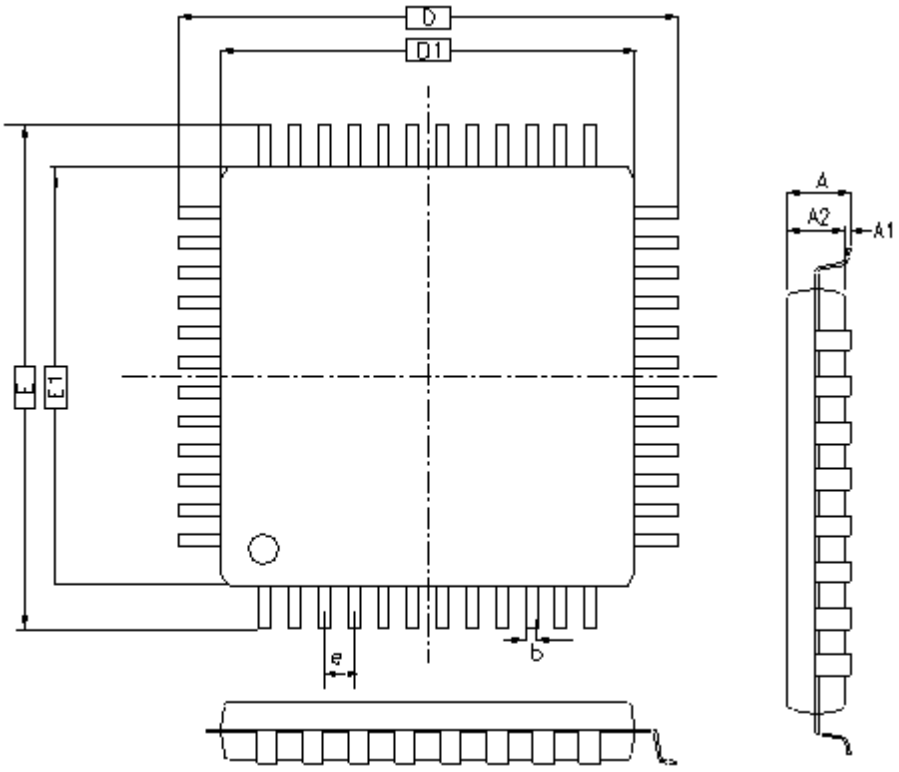
Note 4: It is recommended to use 10 kΩ pull-up resistor and 10 uF capacitor on nRESET pin.

8 ELECTRICAL CHARACTERISTICS

For information on the Mini58 series electrical characteristics, please refer to NuMicro® Mini58 Series Datasheet.

9 PACKAGE DIMENSIONS

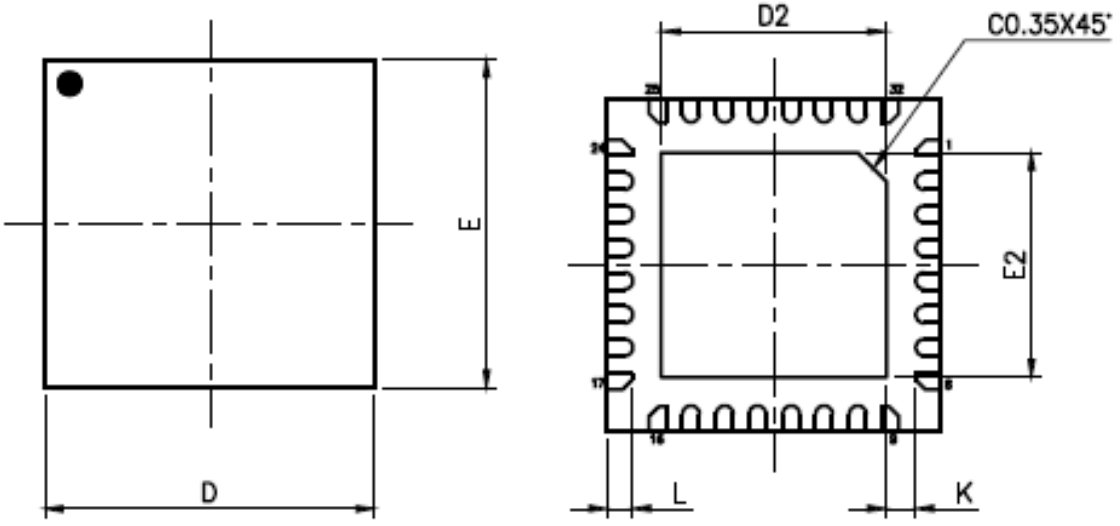
9.1 48-pin LQFP

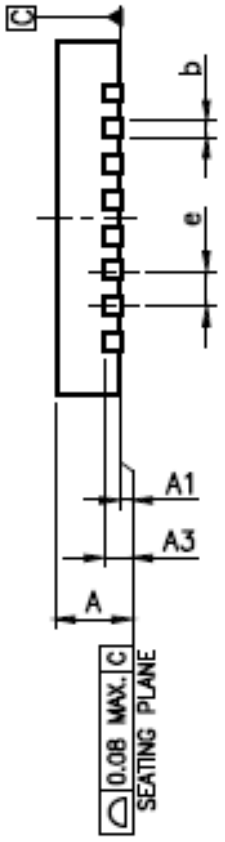


CONTROL DIMENSIONS ARE IN MILLIMETERS

SYMBOL	MILLIMETER			INCH		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	—	—	1.60	—	—	0.063
A1	0.05	0.10	0.15	0.002	0.004	0.006
A2	1.35	1.40	1.45	0.053	0.055	0.057
D1	6.90	7.00	7.10	0.272	0.276	0.280
E1	6.90	7.00	7.10	0.272	0.276	0.280
e	0.35	0.50	0.65	0.014	0.020	0.026
D	8.9	9.00	9.10	0.350	0.354	0.358
E	8.9	9.00	9.10	0.350	0.354	0.358
L	0.45	0.60	0.75	0.018	0.024	0.030
L1	—	1.00	—	—	0.039	—
C	0.09	—	0.20	0.004	—	0.0079
θ	0°	—	7°	0°	—	7°
b	0.17	0.22	0.27	0.007	0.0087	0.011

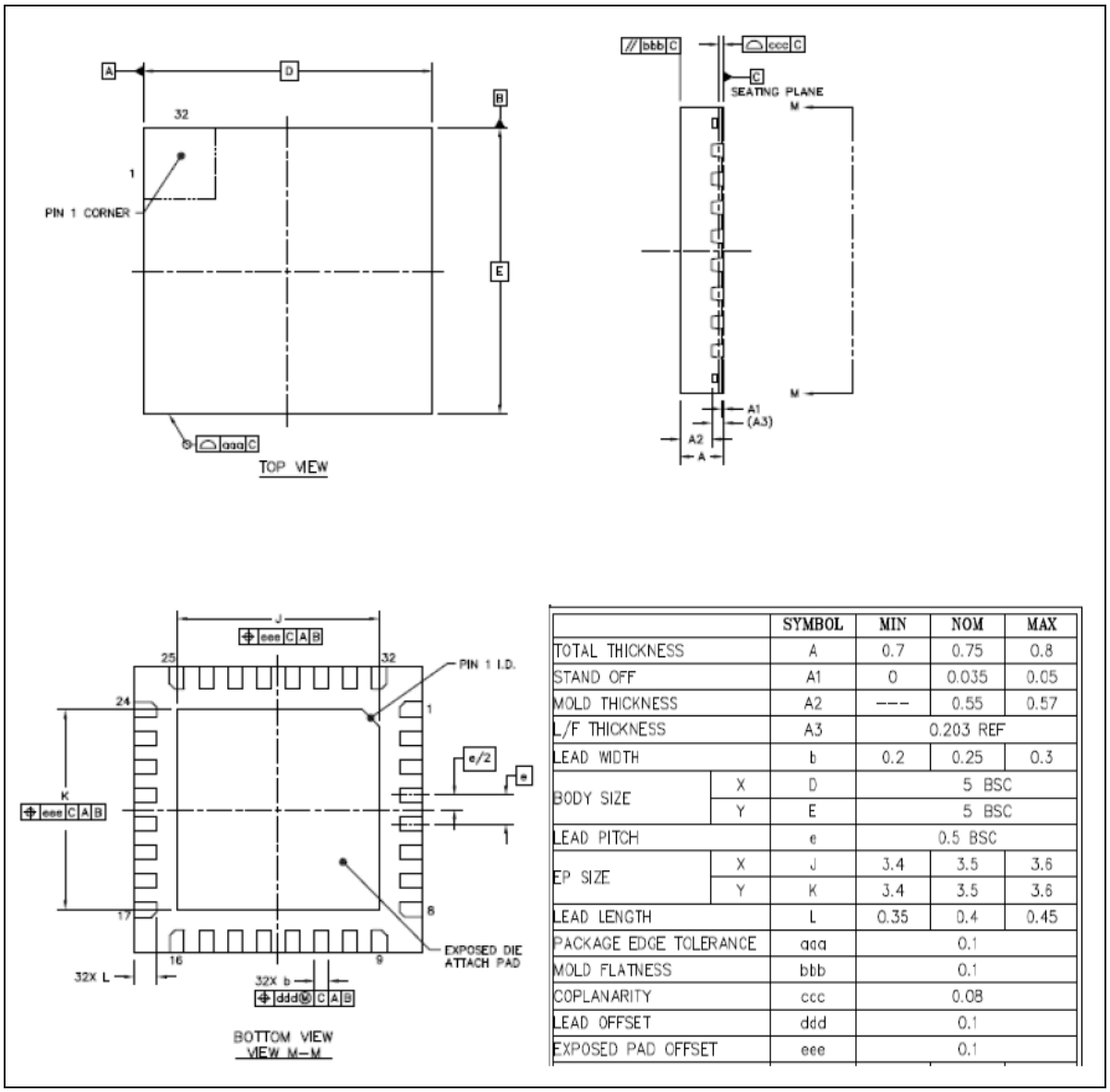
9.2 33-pin QFN (4 mm x 4 mm)





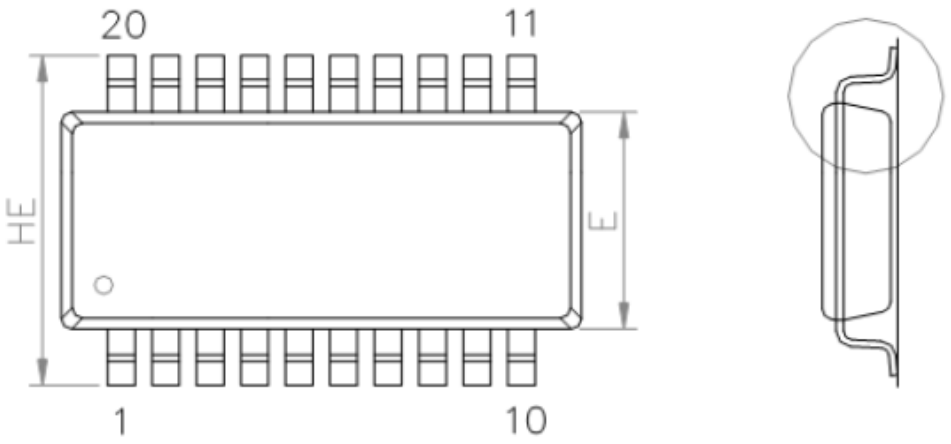
SYMBOLS	MIN.	NOM.	MAX.
A	0.70	0.75	0.80
A1	0.00	0.02	0.05
A3		0.203 REF.	
b	0.15	0.20	0.25
D		4.00 BSC	
E		4.00 BSC	
e		0.40 BSC	
L	0.25	0.30	0.35
D2	2.60	2.70	2.80
E2	2.60	2.70	2.80
K	0.20	—	—

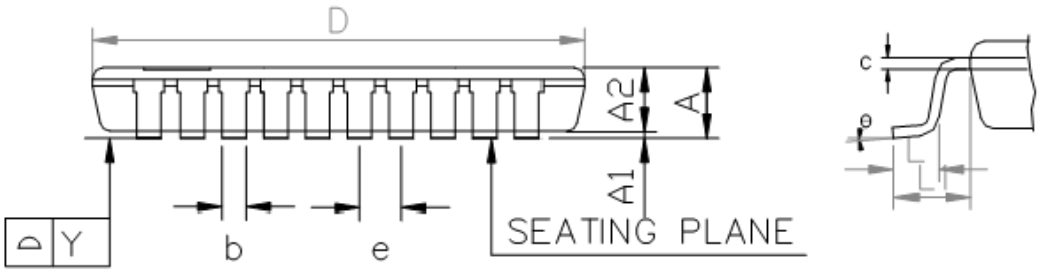
9.3 33-pin QFN (5 mm x 5 mm)



		SYMBOL	MIN	NOM	MAX
TOTAL THICKNESS		A	0.7	0.75	0.8
STAND OFF		A1	0	0.035	0.05
MOLD THICKNESS		A2	---	0.55	0.57
L/F THICKNESS		A3		0.203 REF	
LEAD WIDTH		b	0.2	0.25	0.3
BODY SIZE	X	D		5 BSC	
	Y	E		5 BSC	
LEAD PITCH		e		0.5 BSC	
EP SIZE	X	J	3.4	3.5	3.6
	Y	K	3.4	3.5	3.6
LEAD LENGTH		L	0.35	0.4	0.45
PACKAGE EDGE TOLERANCE		aaa		0.1	
MOLD FLATNESS		bbb		0.1	
COPLANARITY		ccc		0.08	
LEAD OFFSET		ddd		0.1	
EXPOSED PAD OFFSET		eee		0.1	

9.4 20-pin TSSOP





SYMBOL	DIMENSION (MM)			DIMENSION (INCH)		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	-	-	1.20	-	-	0.047
A1	0.05	-	0.15	0.002	-	0.006
A2	0.80	0.90	1.05	0.031	0.035	0.041
E	4.30	4.40	4.50	0.169	0.173	0.177
HE	6.40 BSC			0.252 BSC		
D	6.40	6.50	6.60	0.252	0.256	0.260
L	0.50	0.60	0.75	0.020	0.024	0.030
L1	1.00 REF			0.039 REF		
b	0.19	-	0.30	0.007	-	0.012
e	0.65 BSC			0.026 BSC		
c	0.09	-	0.20	0.004	-	0.008
θ	0°	-	8°	0°	-	8°
Y	0.10 BASIC			0.004 BASIC		

10 REVISION HISTORY

Date	Revision	Description
2015.06.11	1.00	Preliminary version.
2015.10.12	1.01	Updated LDROM size from 2 Kbytes to 2.5 Kbytes.
2016.07.18	1.02	1. Added Section 6.2.6 Register Protection. 2. Added Section 6.13.5.4 Internal Reference Voltage.
2018.05.09	1.03	Added Section 6.9.6.1description.
2020.04.15	1.04	1. Updated 33-pin QFN (4 mm x 4 mm) Package Dimension in section 9.2. 2. Added notes about the hardware reference design for ICE_DAT, ICE_CLK and nRESET pins in section 4.4 and chapter 7.

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.